|  |  |
| --- | --- |
|  | **BỘ CÔNG THƯƠNG**  **TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**  **---------------------------------------** |
| NGUYỄN PHI LONG | **ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC** |
|  | **NGÀNH KỸ THUẬT PHẦN MỀM** |
|  |  |
|  | **XÂY DỰNG ỨNG DỤNG WEB HỖ TRỢ GIÁM THỊ XÁC NHẬN**  **THÍ SINH DỰA TRÊN TÍNH NĂNG NHẬN DIỆN KHUÂN MẶT** |
|  |  |
|  |  |
| KỸ THUẬT PHẦN MỀM | **CBHD: ThS. Phạm Thị Kim Phượng**  **Sinh viên: Nguyễn Phi Long**  **Mã sinh viên: 2020605458** |
|  | **Hà Nội – Năm 2024** |

**MỤC LỤC**

[DANH MỤC CÁC BẢNG 5](#_Toc177047689)

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ 6](#_Toc177047690)

[LỜI CẢM ƠN 8](#_Toc177047691)

[MỞ ĐẦU 9](#_Toc177047692)

[CHƯƠNG 1. KHẢO SÁT HỆ THỐNG 12](#_Toc177047693)

[1.1. Giới thiệu chung về công ty/ tổ chức 12](#_Toc177047694)

[1.2. Hoạt động của hệ thống 13](#_Toc177047695)

[1.3. Các yêu cầu chức năng 14](#_Toc177047696)

[1.3.1. Phân tích và đánh giá phương án chức năng 15](#_Toc177047697)

[CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG 19](#_Toc177047698)

[2.1. Biểu đồ use case 19](#_Toc177047699)

[2.1.1. Biểu đồ Use case 19](#_Toc177047700)

[2.1.2. Quan hệ giữa các Use case 19](#_Toc177047701)

[2.2. Mô tả chi tiết các use case 21](#_Toc177047702)

[2.2.1.Usecase Đăng nhập 21](#_Toc177047703)

[2.2.2.Usecase quên mật khẩu 22](#_Toc177047704)

[2.2.3. Usecase quản lý trang cá nhân 22](#_Toc177047705)

[2.2.4. Usecase quản lý tài khoản 23](#_Toc177047706)

[2.2.5. Usecase quản lý sinh viên của lịch thi 24](#_Toc177047707)

[2.2.6. Usecase quản lý lịch thi 25](#_Toc177047708)

[2.2.7. Usecase điểm danh sinh viên 26](#_Toc177047709)

[2.3.Thiết kế cơ sở dữ liệu 27](#_Toc177047710)

[2.3.1.Biểu đồ thực thể liên kết 27](#_Toc177047711)

[2.3.2. Thiết kế bảng 29](#_Toc177047712)

[2.3.3.Quan hệ các bảng 32](#_Toc177047713)

[2.4.Phân tích các use case 32](#_Toc177047714)

[2.4.1.Phân tích use case 32](#_Toc177047715)

[CHƯƠNG 3. THIẾT KẾ GIAO DIỆN, CÀI ĐẶT VÀ KIỂM THỬ 42](#_Toc177047716)

[3.1.Thiết kế giao diện 42](#_Toc177047717)

[3.2.Giới thiệu ngôn ngữ cài đặt 47](#_Toc177047718)

[3.2.1.Giới thiệu về React JS 47](#_Toc177047719)

[3.2.2.NodeJS 50](#_Toc177047720)

[3.2.3.Mô hình 53](#_Toc177047721)

[3.2.4.Cơ sở dữ liệu MySQL 56](#_Toc177047722)

[3.2.5.Amazon Rekognition 58](#_Toc177047723)

[3.2.6.Các dịch vụ liên quan AWS 61](#_Toc177047724)

[3.2.7.Chi tiết chi phí 64](#_Toc177047725)

[3.2.8.Kết luận và Đề xuất 65](#_Toc177047726)

[3.2.9.Cài đặt chương trình 67](#_Toc177047727)

[3.2.10.Kiểm thử 69](#_Toc177047728)

[CHƯƠNG 4. TỔNG KẾT VÀ ĐÁNH GIÁ 74](#_Toc177047729)

[4.1.Kết quả đạt được 74](#_Toc177047730)

[4.2.Định hướng phát triển 75](#_Toc177047731)

[TÀI LIỆU THAM KHẢO 77](#_Toc177047732)

# DANH MỤC CÁC BẢNG

[*Bảng 2. 1.Quan hệ giữa các use case* 18](#_Toc177047568)

[*Bảng 2. 2. Cơ sở dữ liệu của bảng exam\_list* 26](#_Toc177047569)

[*Bảng 2. 3. Cơ sở dữ liệu của exams* 26](#_Toc177047570)

[*Bảng 2. 4. Cơ sở dữ liệu của password\_reset\_tokens* 27](#_Toc177047571)

[*Bảng 2. 5. Cơ sở dữ liệu của bảng user* 28](#_Toc177047572)

[*Bảng 3. 1. Kiểm thử chức năng đăng nhập* 66](#_Toc177047574)

[*Bảng 3. 2.Kiểm thử chức năng đăng ký* 68](#_Toc177047575)

[*Bảng 3. 3.Kiểm thử một số chức năng khác* 70](#_Toc177047576)

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[*Hình 2. 1.Biểu đồ use tổng quát* 17](#_Toc177047633)

[*Hình 2. 2.Quan hệ các bảng dữ liệu* 30](#_Toc177047634)

[*Hình 2. 3. Biểu đồ trình tự đăng nhập* 31](#_Toc177047635)

[*Hình 2. 4. Biểu đồ lớp đăng nhập* 31](#_Toc177047636)

[*Hình 2. 5. Biểu đồ trình tự quên mật khẩu* 32](#_Toc177047637)

[*Hình 2. 6. Biểu đồ lớp quên mật khẩu* 32](#_Toc177047638)

[*Hình 2. 7.Biểu đồ tuần tự quản lý trang cá nhân* 33](#_Toc177047639)

[*Hình 2. 8. Biểu đồ lớp quản lý trang cá nhân* 34](#_Toc177047640)

[*Hình 2. 9.Biểu đồ tuần tự quản lý tài khoản* 34](#_Toc177047641)

[*Hình 2. 10.Biểu đồ lớp quản lý tài khoản* 35](#_Toc177047642)

[*Hình 2. 11.Biểu đồ tuần tự quản lý sinh viên của lịch thi* 36](#_Toc177047643)

[*Hình 2. 12.Biểu đồ lớp quản lý sinh viên của lịch thi* 36](#_Toc177047644)

[*Hình 2. 13.Biểu đồ tuần tự quản lý lịch thi* 37](#_Toc177047645)

[*Hình 2. 14.Biểu đồ lớp quản lý lịch thi* 38](#_Toc177047646)

[*Hình 2. 15.Biểu đồ tuần tự điểm danh sinh viên* 39](#_Toc177047647)

[*Hình 2. 16.Biểu đồ lớp điểm danh sinh viên* 39](#_Toc177047648)

[*Hình 3. 1. Giao diện chọn lịch thi* 41](#_Toc177047649)

[*Hình 3. 2.Giao diện điểm danh* 41](#_Toc177047650)

[*Hình 3. 3.Giao diện đăng nhập cho Client* 42](#_Toc177047651)

[*Hình 3. 4.Giao diện thống kê* 42](#_Toc177047652)

[*Hình 3. 5.Giao diện quản lý tài khoản* 43](#_Toc177047653)

[*Hình 3. 6.Giao diện tạo tài khoản* 43](#_Toc177047654)

[*Hình 3. 7.Giao diện trang cá nhân* 44](#_Toc177047655)

[*Hình 3. 8.Giao diện cập nhật tài khoản* 44](#_Toc177047656)

[*Hình 3. 9.Giao diện quản lý lịch thi* 45](#_Toc177047657)

[*Hình 3. 10.Giao diện tạo lịch thi* 45](#_Toc177047658)

[*Hình 3. 11.Giao diện chi tiết lịch thi và danh sách sinh viên* 46](#_Toc177047659)

[*Hình 3. 12.Giao diện đăng nhập* 46](#_Toc177047660)

[*Hình 3. 13.Mô hình MVC* 52](#_Toc177047661)

# LỜI CẢM ƠN

Lời đầu tiên, em xin chân thành cảm ơn quý thầy cô trong **Khoa Công Nghệ Thông Tin** đã luôn đồng hành và hỗ trợ em suốt chặng đường học tập và nghiên cứu tại trường. Đặc biệt, em muốn gửi lời tri ân đặc biệt đến cô **ThS. Phạm Thị Kim Phượng**, người đã dành thời gian và tâm huyết để hướng dẫn, giúp đỡ và chia sẻ kiến thức quý báu trong suốt thời gian thực hiện đồ án tốt nghiệp. Em không quên bày tỏ lòng biết ơn sâu sắc đến tất cả các thầy cô ở **Trường Đại học Công nghiệp Hà Nội**, những người đã chia sẻ tình cảm, kinh nghiệm và kỹ thuật giáo dục quý báu. Điều này đã giúp em hiểu rõ hơn về cách xây dựng và phát triển đề tài “**Xây dựng ứng dụng web hỗ trợ giám thị xác nhận thí sinh dựa trên tính năng nhận dạng khuôn mặt** “của mình. Mặc dù thời gian hạn chế và những thách thức không tránh khỏi, em đã cố gắng hết sức để áp dụng những kiến thức vào đề tài của mình. Tuy nhiên, em nhận thức rằng còn nhiều điều để hoàn thiện và phát triển trong tương lai. Do đó, em mong nhận được sự đóng góp, góp ý và hỗ trợ chân thành từ thầy cô cùng các bạn. Những đóng góp này sẽ là nguồn động viên quý báu giúp em vượt qua những hạn chế và phát triển hơn trong sự nghiệp nghiên cứu của mình.

Lời cuối cùng em xin kính chúc thầy cô và các bạn sức khỏe và gặt hái thêm nhiều thành công trong sự nghiệp.

Em xin chân thành cảm ơn!

Người thực hiện

Nguyễn Phi Long

# MỞ ĐẦU

**Lý do chọn đề tài**

Lý do chọn đề tài "Xây dựng ứng dụng web hỗ trợ giám thị xác nhận thí sinh dựa trên tính năng nhận dạng khuôn mặt" là nhằm giải quyết các vấn đề hiện tại trong việc xác thực danh tính thí sinh tại các kỳ thi. Với sự phát triển nhanh chóng của công nghệ nhận dạng khuôn mặt, ứng dụng này có thể cung cấp một giải pháp chính xác và hiệu quả để ngăn chặn gian lận và đảm bảo tính công bằng. Công nghệ nhận dạng khuôn mặt không chỉ giúp nâng cao độ tin cậy trong việc xác thực thí sinh, mà còn tối ưu hóa quy trình kiểm tra, giảm thiểu sự can thiệp thủ công của giám thị, và tiết kiệm thời gian. Việc áp dụng công nghệ hiện đại này vào quản lý thi cử không chỉ giúp tăng cường bảo mật mà còn đáp ứng nhu cầu ngày càng cao về một hệ thống thi cử hiện đại và minh bạch.

**Mục tiêu nghiên cứu**

1. Phát triển hệ thống nhận dạng khuôn mặt chính xác và hiệu quả: Thiết kế và triển khai một thuật toán nhận dạng khuôn mặt có độ chính xác cao, khả năng nhận diện nhanh chóng, và khả năng hoạt động trong các điều kiện ánh sáng và góc nhìn khác nhau.
2. Tạo giao diện người dùng thân thiện và dễ sử dụng: Xây dựng giao diện web cho giám thị dễ dàng thao tác, bao gồm các chức năng như quét khuôn mặt, xác thực danh tính thí sinh và quản lý dữ liệu liên quan một cách trực quan và hiệu quả.
3. Tích hợp hệ thống quản lý dữ liệu thí sinh: Phát triển các chức năng để lưu trữ và quản lý thông tin thí sinh, bao gồm dữ liệu cá nhân và hình ảnh khuôn mặt, đồng thời đảm bảo an toàn và bảo mật thông tin theo các tiêu chuẩn bảo mật hiện hành.
4. Đảm bảo khả năng mở rộng và tích hợp với các hệ thống khác: Thiết kế ứng dụng với khả năng mở rộng để dễ dàng tích hợp với các hệ thống quản lý thi cử khác hoặc các phần mềm giám sát khác, nhằm nâng cao tính linh hoạt và khả năng thích ứng của hệ thống.
5. Đánh giá và cải tiến hiệu suất hệ thống: Tiến hành các thử nghiệm và đánh giá hiệu suất của hệ thống trong điều kiện thực tế để xác định các điểm mạnh và điểm yếu, từ đó đưa ra các giải pháp cải tiến nhằm nâng cao hiệu quả hoạt động và độ tin cậy của ứng dụng.

**Đối tượng nghiên cứu**

* Công ty Education
* Công nghệ sử dụng: react js, node js,python, amazon service.
* Hệ quản trị cơ sở dữ liệu: MySQL.

**Phạm vi nghiên cứu**

* Thời gian: Thực hiện trong khoảng thời gian 9 tuần.
* Khảo sát về thực trạng tại các trường học , đại học. Tìm hiểu những yếu tố và điều kiện cần thiết phải có để có thể hỗ trợ cho dự án một cách tối ưu nhất cũng như tiện lợi cho người sử dụng.
* Nghiên cứu qua Internet: đánh giá và đưa ra các ý kiến khách quan về vấn đề.
* Thực hiện phân tích các chức năng chính cần thiết cho hệ thống.
* Lựa chọn công nghệ phù hợp để phát triển cũng như các công cụ cần thiết để xây dựng.

**Kết quả mong muốn đạt được của đề tài**

1. Chất lượng nhận dạng khuôn mặt cao: Ứng dụng có khả năng nhận diện khuôn mặt với độ chính xác trên 95%, ngay cả trong điều kiện ánh sáng khác nhau và các góc nhìn khác nhau. Điều này giúp đảm bảo thí sinh được xác thực một cách chính xác và nhanh chóng.
2. Giao diện người dùng dễ sử dụng: Giao diện web được thiết kế thân thiện, trực quan, giúp giám thị dễ dàng thực hiện các thao tác như quét khuôn mặt, xác nhận danh tính thí sinh và quản lý thông tin. Giao diện này cũng cần được tối ưu hóa cho cả máy tính và thiết bị di động.
3. Quản lý và bảo mật dữ liệu hiệu quả: Hệ thống có khả năng lưu trữ, quản lý và bảo mật thông tin thí sinh một cách an toàn, tuân thủ các tiêu chuẩn bảo mật dữ liệu hiện hành để bảo vệ thông tin cá nhân và ngăn chặn truy cập trái phép.
4. Khả năng tích hợp và mở rộng linh hoạt: Ứng dụng có khả năng tích hợp dễ dàng với các hệ thống quản lý thi cử khác và có khả năng mở rộng để đáp ứng nhu cầu tương lai. Điều này giúp ứng dụng dễ dàng thích ứng với các yêu cầu mới và tích hợp với các phần mềm khác nếu cần.
5. Hiệu suất hệ thống ổn định và tin cậy: Hệ thống hoạt động ổn định với thời gian phản hồi nhanh, xử lý hiệu quả trong môi trường thực tế. Các thử nghiệm cho thấy ứng dụng có thể duy trì hiệu suất tốt khi xử lý một lượng lớn thí sinh cùng một lúc.

**Cấu trúc báo cáo**

Báo cáo này được tổ chức thành 4 chương chính:

Chương 1. Khảo sát hệ thống: Nội dung của chương trình bày chi tiết về các bước khảo sát hệ thống nhận diện khuân mặt với các phần giới thiệu về công ty, hoạt động của hệ thống, các yêu cầu chức năng, tài liệu thu thập được.

Chương 2. Phân tích hệ thống: Tập trung vào việc phân tích yếu tố quan trọng như biểu đồ use case, mô tả use case, thiết kế cơ sở dữ liệu, phân tích use case.

Chương 3. Thiết kế giao diện, cài đặt và kiểm thử: Trình bày về sơ đồ điều hướng giữa các màn hình, thiết kế chi tiết màn hình, giới thiệu ngôn ngữ cài đặt, cài đặt chương trình và quá trình kiểm thử để đảm bảo tính ổn định và an toàn.

Chương 4. Tổng kết và đánh giá: Trình bày về kết luận chung về kết quả đạt được so với mục đích, các kiến thức tiếp thu được, đề xuất hướng phát triển trong tương lai.

Những kết quả này sẽ đảm bảo ứng dụng không chỉ đáp ứng yêu cầu về bảo mật và hiệu quả trong việc xác thực thí sinh, mà còn nâng cao trải nghiệm người dùng và khả năng mở rộng của hệ thống.

# CHƯƠNG 1. KHẢO SÁT HỆ THỐNG

## 1.1. Giới thiệu chung về công ty/ tổ chức

Trong bối cảnh giáo dục ngày càng phát triển, việc giám sát và đảm bảo tính trung thực trong các kỳ thi trở thành một yếu tố quan trọng. Truyền thống, việc coi thi thường dựa vào sự hiện diện vật lý của giám thị trong phòng thi để theo dõi và kiểm tra thí sinh. Tuy nhiên, phương pháp này không chỉ tốn nhiều nguồn lực mà còn gặp nhiều hạn chế trong việc phát hiện các hành vi gian lận tinh vi. Với sự tiến bộ của công nghệ, đặc biệt là trong lĩnh vực trí tuệ nhân tạo và nhận dạng khuôn mặt, có thể cải thiện hiệu quả và tính chính xác của quá trình giám sát thi cử.

Mục đích của đề tài "Xây dựng ứng dụng web hỗ trợ giám thị xác nhận thí sinh dựa trên tính năng nhận dạng khuôn mặt" là nghiên cứu và phát triển một ứng dụng web sử dụng dịch vụ nhận dạng khuôn mặt của Amazon Web Services (AWS) để hỗ trợ quá trình coi thi. Ứng dụng này không chỉ giúp giám thị dễ dàng theo dõi và quản lý thí sinh trong thời gian thực, mà còn tự động phát hiện các hành vi đáng ngờ như việc thay đổi thí sinh, hay hành vi gian lận trong phòng thi. Bằng cách ứng dụng công nghệ nhận dạng khuôn mặt, chúng tôi mong muốn nâng cao tính chính xác, hiệu quả và giảm thiểu sự phụ thuộc vào nguồn lực con người trong quá trình giám sát thi cử.

Hơn nữa, việc sử dụng dịch vụ nhận dạng khuôn mặt của AWS mang lại nhiều lợi ích về mặt công nghệ như khả năng mở rộng, bảo mật và độ tin cậy cao. AWS cung cấp một nền tảng mạnh mẽ và linh hoạt, cho phép tích hợp dễ dàng với các hệ thống hiện có và hỗ trợ việc phát triển các tính năng nâng cao trong tương lai. Qua đó, ứng dụng này không chỉ có ý nghĩa trong việc nâng cao chất lượng và sự công bằng trong giáo dục, mà còn góp phần vào xu hướng số hóa và tự động hóa trong quản lý giáo dục hiện đại.

## 1.2. Hoạt động của hệ thống

Đồ án "Xây dựng ứng dụng web hỗ trợ giám thị xác nhận thí sinh dựa trên tính năng nhận dạng khuôn mặt" sẽ tập trung vào việc phát triển và triển khai một hệ thống web giúp giám sát và quản lý kỳ thi một cách hiệu quả, tích hợp công nghệ nhận dạng khuôn mặt tiên tiến. Nội dung đồ án được chia thành hai phần chính, bao gồm các chức năng dành cho người quản trị (Admin) và người dùng (Client).

Phần đầu tiên của đồ án sẽ chi tiết hóa các chức năng dành cho Admin, bao gồm việc đăng ký, đăng nhập và quản lý tài khoản. Admin có thể thực hiện các thao tác như quên mật khẩu, thay đổi mật khẩu và cập nhật thông tin cá nhân để đảm bảo rằng tài khoản luôn được bảo mật và cập nhật. Ngoài ra, Admin còn có khả năng quản lý tài khoản của các người dùng khác, giúp duy trì hệ thống và điều chỉnh quyền truy cập một cách hiệu quả. Một khía cạnh quan trọng khác là việc quản lý môn học, từ việc thêm mới, cập nhật thông tin đến việc quản lý học viên của từng môn học. Đặc biệt, Admin có thể tạo lịch thi cho các môn học, đảm bảo rằng kỳ thi được tổ chức một cách suôn sẻ và có tổ chức.

Phần tiếp theo của đồ án sẽ tập trung vào các chức năng dành cho Client, chủ yếu là giám thị và thí sinh. Một trong những chức năng quan trọng là việc điểm danh sinh viên cho các môn học sử dụng công nghệ nhận dạng khuôn mặt của AWS. Thông qua ứng dụng, giám thị có thể dễ dàng xác minh danh tính của sinh viên tham gia thi cử, đảm bảo rằng chỉ những sinh viên đã được xác thực mới có thể tham gia kỳ thi. Sau khi điểm danh, hệ thống sẽ tự động cập nhật danh sách sinh viên đã điểm danh trên giúp giám thị có thể theo dõi và quản lý thông tin một cách thuận tiện và chính xác.

Để hiện thực hóa các chức năng này, đồ án sẽ đi qua các giai đoạn từ phân tích yêu cầu hệ thống, thiết kế kiến trúc và giao diện, cho đến triển khai và kiểm thử hệ thống. Cụ thể, phần thiết kế hệ thống sẽ mô tả chi tiết cấu trúc của ứng dụng web, bao gồm các thành phần frontend và backend, và cách chúng tích hợp với các dịch vụ của AWS như Amazon Rekognition cho nhận dạng khuôn mặt. Việc triển khai sẽ bao gồm cài đặt và cấu hình các dịch vụ cần thiết, đồng thời đảm bảo rằng hệ thống có thể hoạt động ổn định và bảo mật.

## 1.3. Các yêu cầu chức năng

Trong bối cảnh phát triển và hiện đại hóa lĩnh vực giáo dục, việc quản lý thi cử và điểm danh sinh viên một cách hiệu quả và chính xác là vô cùng quan trọng. Đồ án "Xây dựng ứng dụng web hỗ trợ giám thị xác nhận thí sinh dựa trên tính năng nhận dạng khuôn mặt" được thiết kế nhằm giải quyết các thách thức trong việc giám sát và quản lý kỳ thi, đồng thời tối ưu hóa quy trình điểm danh sinh viên cho các môn học. Hệ thống này sẽ bao gồm hai nhóm chức năng chính, phục vụ cho các đối tượng người dùng khác nhau: Admin và Client.

**Chức Năng Của Admin**

**Đăng Ký, Đăng Nhập:**

* Hệ thống cung cấp giao diện cho Admin đăng ký tài khoản mới và đăng nhập vào hệ thống. Việc này đảm bảo rằng chỉ những người có quyền hạn mới có thể truy cập vào các chức năng quản trị của hệ thống.

**Quên Mật Khẩu, Thay Đổi Mật Khẩu:**

* Admin có thể sử dụng chức năng quên mật khẩu để khôi phục lại quyền truy cập trong trường hợp mất mật khẩu. Ngoài ra, hệ thống cho phép Admin thay đổi mật khẩu định kỳ hoặc khi cần thiết để tăng cường bảo mật.

**Thay Đổi Thông Tin Cá Nhân:**

* Admin có thể cập nhật các thông tin cá nhân của mình như tên, email, hoặc số điện thoại. Điều này giúp đảm bảo rằng các thông tin liên lạc của Admin luôn được cập nhật và chính xác.

**Quản Lý Tài Khoản:**

* Admin có khả năng quản lý các tài khoản người dùng khác trong hệ thống. Điều này bao gồm việc thêm mới, xóa bỏ hoặc cập nhật thông tin của các tài khoản này, giúp duy trì kiểm soát chặt chẽ và tổ chức các quyền truy cập.

**Quản Lý Học Viên Của Môn Học:**

* Admin có thể quản lý danh sách học viên tham gia từng môn học, bao gồm việc thêm mới, hoặc loại bỏ học viên. Điều này giúp đảm bảo rằng thông tin về học viên luôn được duy trì một cách chính xác và kịp thời.

**Tạo Lịch Thi Cho Môn Học:**

* Hệ thống cho phép Admin lên lịch thi cho các môn học, bao gồm việc xác định ngày, giờ và địa điểm thi. Chức năng này giúp tổ chức kỳ thi một cách hiệu quả và thông báo kịp thời cho sinh viên.

**Chức Năng Của Client**

**Điểm Danh Sinh Viên Cho Môn Học:**

* Giám thị có thể sử dụng hệ thống để điểm danh sinh viên bằng công nghệ nhận dạng khuôn mặt của AWS. Việc này giúp xác minh danh tính của sinh viên một cách tự động và nhanh chóng, đảm bảo rằng chỉ những sinh viên đã đăng ký mới được tham gia vào lớp học hoặc kỳ thi.

**Xem Danh Sách Sinh Viên Đã Điểm Danh:**

* Hệ thống sẽ tự động cập nhật danh sách sinh viên đã được điểm. Giám thị có thể dễ dàng truy cập và xem danh sách này để theo dõi và quản lý sự có mặt của sinh viên trong lớp học hoặc kỳ thi.

### 1.3.1. Phân tích và đánh giá phương án chức năng

* **Chức Năng Của Admin**

**Đăng Ký, Đăng Nhập**

Phân Tích:

Mô Tả: Chức năng này cho phép Admin tạo tài khoản mới và đăng nhập vào hệ thống để truy cập vào các chức năng quản trị. Đây là bước đầu tiên để đảm bảo tính bảo mật và quyền hạn truy cập.

Yêu Cầu: Giao diện người dùng cần đơn giản và dễ sử dụng, yêu cầu xác thực thông tin (email, mật khẩu), và hỗ trợ bảo mật cao (như mã hóa mật khẩu).

Đánh Giá: Đây là chức năng cơ bản và cần thiết cho bất kỳ hệ thống quản trị nào. Cung cấp khả năng kiểm soát truy cập cho phép quản lý người dùng và bảo vệ dữ liệu.

**Quên Mật Khẩu, Thay Đổi Mật Khẩu**

Phân Tích:

Mô Tả: Cung cấp khả năng khôi phục mật khẩu và thay đổi mật khẩu giúp tăng cường bảo mật và thuận tiện cho Admin khi quên mật khẩu hoặc cần cập nhật mật khẩu thường xuyên.

Yêu Cầu: Cần có hệ thống gửi email khôi phục mật khẩu, yêu cầu người dùng xác thực danh tính trước khi thay đổi mật khẩu mới.

Đánh Giá: Đây là tính năng quan trọng cho bảo mật và dễ dàng sử dụng. Tuy nhiên, cần đảm bảo rằng các phương pháp khôi phục và thay đổi mật khẩu không bị lạm dụng.

**Thay Đổi Thông Tin Cá Nhân**

Phân Tích:

Mô Tả: Admin có thể cập nhật thông tin cá nhân như tên, email, số điện thoại. Điều này đảm bảo rằng các thông tin liên lạc luôn được cập nhật và chính xác.

Yêu Cầu: Giao diện dễ sử dụng, cho phép Admin chỉnh sửa thông tin và lưu thay đổi. Cần có xác thực để bảo vệ thông tin cá nhân.

Đánh Giá: Đây là chức năng hữu ích cho việc duy trì thông tin chính xác. Tuy nhiên, cần đảm bảo rằng các thay đổi thông tin không làm mất tính toàn vẹn của dữ liệu.

**Quản Lý Tài Khoản**

Phân Tích:

Mô Tả: Admin có thể thêm mới, xóa bỏ hoặc cập nhật thông tin các tài khoản người dùng khác. Điều này giúp quản lý người dùng hiệu quả và duy trì quyền truy cập.

Yêu Cầu: Cần có giao diện quản lý tài khoản, cùng với các chức năng tìm kiếm, phân loại, và phân quyền.

Đánh Giá: Đây là chức năng quan trọng cho việc quản lý và duy trì tổ chức. Tuy nhiên, cần cẩn thận với quyền hạn phân phối để tránh việc lạm dụng.

**Quản Lý Học Viên Của Môn Học**

Phân Tích:

Mô Tả: Admin có thể quản lý danh sách học viên của từng môn học, bao gồm việc thêm mới hoặc loại bỏ học viên.

Yêu Cầu: Cần có chức năng để cập nhật danh sách học viên dễ dàng, cùng với khả năng đồng bộ dữ liệu với hệ thống quản lý học tập.

Đánh Giá: Chức năng này đảm bảo rằng thông tin học viên luôn chính xác và kịp thời. Cần đảm bảo việc đồng bộ dữ liệu không gây xung đột.

**Tạo Lịch Thi Cho Môn Học**

Phân Tích:

Mô Tả: Admin có thể lên lịch thi cho các môn học, bao gồm việc xác định ngày, giờ và địa điểm thi. Chức năng này hỗ trợ tổ chức kỳ thi một cách hiệu quả.

Yêu Cầu: Cần có giao diện để lên lịch và thông báo cho sinh viên. Hệ thống cần tích hợp với các lịch trình và thông báo kịp thời.

Đánh Giá: Đây là chức năng quan trọng cho việc tổ chức thi cử. Tuy nhiên, cần đảm bảo rằng lịch thi được quản lý chặt chẽ để tránh xung đột.

* **Chức Năng Của Client**

**Điểm Danh Sinh Viên Cho Môn Học**

Phân Tích:

Mô Tả: Giám thị có thể sử dụng hệ thống để điểm danh sinh viên bằng công nghệ nhận diện khuôn mặt của AWS. Chức năng này giúp xác minh danh tính sinh viên tự động và nhanh chóng.

Yêu Cầu: Hệ thống cần tích hợp công nghệ nhận diện khuôn mặt của AWS, cùng với giao diện để chụp và xử lý hình ảnh. Cần có cơ chế kiểm tra và xác nhận kết quả.

Đánh Giá: Chức năng này cải thiện hiệu quả điểm danh và giảm thiểu lỗi so với các phương pháp truyền thống. Tuy nhiên, cần đảm bảo rằng hệ thống nhận diện hoạt động chính xác trong các điều kiện ánh sáng khác nhau và với các thay đổi trong khuôn mặt.

**Xem Danh Sách Sinh Viên Đã Điểm Danh**

Phân Tích:

Mô Tả: Giám thị có thể xem danh sách sinh viên đã được điểm danh. Hệ thống tự động cập nhật danh sách này để theo dõi và quản lý sự có mặt của sinh viên.

Yêu Cầu: Cần có giao diện để xem và quản lý danh sách điểm danh, cùng với chức năng lọc và tìm kiếm dữ liệu.

Đánh Giá: Chức năng này giúp giám thị theo dõi sự có mặt của sinh viên dễ dàng và chính xác. Tuy nhiên, cần đảm bảo rằng danh sách được cập nhật kịp thời và không có lỗi dữ liệu.

# CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG

## 2.1. Biểu đồ use case

### 2.1.1. Biểu đồ Use case



*Hình 2. 1.Biểu đồ use tổng quát*

### 2.1.2. Quan hệ giữa các Use case

Hệ thống website tôi hướng đến 2 đối tượng là: Admin, Client

|  |  |
| --- | --- |
| Đăng Ký | Admin có thể tạo tài khoản mới bằng cách điền các thông tin cá nhân cần thiết và gửi yêu cầu đăng ký. |
| Đăng Nhập | Admin nhập tên đăng nhập và mật khẩu để truy cập vào hệ thống quản trị. |
| Quên Mật Khẩu | Cung cấp tùy chọn cho Admin để yêu cầu đặt lại mật khẩu thông qua email trong trường hợp quên mật khẩu. |
| Thay Đổi Mật Khẩu | Admin có thể cập nhật mật khẩu mới cho tài khoản sau khi đăng nhập thành công hoặc khôi phục mật khẩu. |
| Thay Đổi Thông Tin Cá Nhân | Admin có thể cập nhật các thông tin cá nhân như tên, email, và số điện thoại thông qua trang hồ sơ cá nhân. |
| Quản Lý Tài Khoản | Admin có thể xem danh sách, tạo mới, chỉnh sửa hoặc xóa các tài khoản người dùng khác trong hệ thống. |
| Quản Lý Môn Học | Admin có thể thêm mới, chỉnh sửa hoặc xóa các thông tin chi tiết về các môn học trong hệ thống. |
| Quản Lý Học Viên Của Lịch Thi | Admin quản lý danh sách học viên cho từng lịch thi, bao gồm việc thêm hoặc xóa học viên. |
| Tạo Lịch Thi Cho Môn Học | Admin có thể thiết lập ngày, giờ, và địa điểm thi cho các môn học. |
| Điểm Danh Sinh Viên Cho Môn Học | Giám thị sử dụng công nghệ nhận dạng khuôn mặt để tự động điểm danh sinh viên tham dự môn học hoặc kỳ thi. |
| Xem Danh Sách Sinh Viên | Giám thị có thể truy cập và xem danh sách sinh viên đã được điểm |

*Bảng 2. 1.Quan hệ giữa các use case*

## 2.2. Mô tả chi tiết các use case

### 2.2.1.Usecase Đăng nhập

* **Mô tả vắn tắt:** Người dùng có thể đăng nhập vào hệ thống bằng tên người dùng và mật khẩu của mình.
* **Luồng sự kiện:** 
  + **Luồng cơ bản**

1. Use case này bắt đầu khi người dùng muốn đăng nhập vào hệ thống. Hệ thống sẽ hiển thị màn hình yêu cầu nhập email và mật khẩu.
2. Người dung nhập emailvà mậtkhẩusau đó click vào nút Đăng nhập. Hệ thống kiểm tra tàikhoản đăng nhập, mật khẩu và quyền truy cập trong bảng USERS và hiển thị màn hìnhtrangchủ hệ thống. Use case kết thúc.

* **Luồng rẽ nhánh**

1. Tại bước 2 trong luồng cơ bản nếu người dùng không nhập tên hoặc không nhập mật khẩu hệ thống sẽ hiển thị thông báo “Cần phải nhập đủ thông tin!”. Người dùng cần nhập đủ thông tin để tiếp tục.
2. Tại bước 2 trong luồng cơ bản nếu người dùng nhập sai tên hoặc/và mật khẩu thì hệ thống sẽ hiển thị thông báo lỗi “Thông tin đăng nhập không đúng!” và use case kết thúc.
3. Tại bất kỳ thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị thông báo “Lỗi kết nối!” và use case kết thúc.

* **Các yêu cầu đặc biệt:** Cần phân quyền truy cập khi đăng nhập hệ thống.
* **Tiền điều kiện:** Người dùng cần có một tài khoản trước khi có thể đăng nhập
* **Hậu điều kiện:** Không có.
* **Điểm mở rộng:** Không có.

### 2.2.2.Usecase quên mật khẩu

* **Mô tả vắn tắt:** Use case cho phép người dùng có thể yêu cầu đặt lại mật khẩu bằng cách nhập địa chỉ email đã đăng ký.
* **Luồng sự kiện:**

1. Use case bắt đầu khi khách hàng kích vào nút “Quên mật khẩu ” ở trang chủ. Hệ thống sẽ hiển thị form nhập email lên màn hình.
2. Người dùng cung cấp địa chỉ email đã đăng ký.
3. Hệ thống gửi một liên kết đặc biệt đến email người dùng để reset mật khẩu.
4. Người dùng truy cập vào liên kết và đặt lại mật khẩu mới cho tài khoản của mình.

* **Điểm mở rộng:** Không có.
* **Tiền điều kiện:** Người dùng đã có tài khoản trong bảng user
* **Hậu điều kiện:** Không có.
* **Yêu cầu đặc biệt:** Không có.

### Usecase quản lý trang cá nhân

* **Mô tả vắn tắt:** Use case cho phép người dùng có thể quản lý thông tin cá nhân, cập nhật mật khẩu, đổi hình đại diện, và thực hiện các hành động khác liên quan đến quản lý tài khoản.
* **Luồng sự kiện:**

1. Usecae bắt đầu khi người dùng truy cập vào trang cá nhân và có các tùy chọn chỉnh sửa thông tin cá nhân và thay đổi mật khẩu.
2. Sau khi thực hiện, thông tin cá nhân và mật khẩu được cập nhật và hiển thị đúng cách.

* **Điểm mở rộng:** Không có.
* **Tiền điều kiện:** Người dung phải đăng nhập trước khi chỉnh sửa.
* **Hậu điều kiện:** Không có.
* **Yêu cầu đặc biệt:** Không có.

### 2.2.4. Usecase quản lý tài khoản

* **Mô tả vắn tắt:** Usecase này cho phép admin có thể xem danh sách, tạo mới, chỉnh sửa hoặc xóa các tài khoản người dùng khác trong hệ thống.
* **Luồng sự kiện:**

1. Usecase bắt đầu khi Admin truy cập trang quản lý tài khoản và thực hiện các thao tác: xem danh sách, tạo mới, chỉnh sửa, xóa.
2. Xem danh sách: Admin kich vào nút “Quản lý tài khoản” trên màn hình trang chủ. Hệ thống sẽ hiển thị màn hình thông tin chi tiết gồm:ID, Ảnh đại diện, Họ và tên, Email,Số điện thoại, Mã số người dung, Phân quyền, Trạng thái.
3. Tạo mới : Admin kích nào nút “Tạo tài khoản” trên cửa sổ quản lý tài khoản . Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết gồm: tên, email, mật khẩu, số điện thoại, phân quyền, mã số người dung và upload ảnh. Admin nhập đầy đủ thông tin và nhấn nút “Hoàn thành”. Hệ thống tự tạo tài khoản và lưu thông tin sản phẩm vào bảng USER và hiển thị thông báo “Tạo tài khoản thành công!” và danh sách đã được cập nhật.
4. Sửa:Admin kích vào nút “Sửa” trên một dòng tài khoản muốn sửa. Hệ thống lấy thông tin cũ của tài khoản và hiển thị lên màn hình trong Textbox. Admin nhập thông tin cần thay đổi và nhấn nút “Hoàn thành”. Hệ thống lưu thông tin sản phẩm mới trong bảng USER và EXAMS sau đó hiển thị thông báo “Đã sửa tài khoàn” và danh sách các tài khoản đã được cập nhật.
5. Chặn tài khoản : Admin kích vào nút “Chặn tài khoản” trên một dòng tài khoản muốn chặn. Sau đó xác nhận yes or no.Hệ thống sẽ lưu thông tin mới trong bảng USER sau đó hiển thị thông báo và các tài khoản đã được cập nhật.
6. Thay đổi quyền : Admin kích vào nút “Thay đổi quyền” trên một dòng tài khoản muốn sửa. Hệ thống lấy thông tin cũ của tài khoản và hiển thị lên màn hình trong Textbox. Admin chọn quyền mới cần thay đổi và nhấn nút “OK”. Hệ thống lưu thông tin sản phẩm mới trong bảng USER và EXAMS sau đó hiển thị thông báo “Đã sửa tài khoàn” và danh sách các tài khoản đã được cập nhật.
7. Xóa: Người dung kích vào nút “Xóa” trên một dòng tài khoản bất kỳ. Hệ thống hiển thị xác nhận “Yes or No?”. Người dùng nhấn nút “Yes”. Hệ thống cập nhật dữ liệu sản phẩm trong bảng USERvà hiển thị thông báo “Xóa thành công” và danh sách các tài khoản đã được cập nhật. Use case kết thúc.

* **Điểm mở rộng:** Không có.
* **Tiền điều kiện:** Người dung phải là Admin.
* **Hậu điều kiện:** Không có.
* **Yêu cầu đặc biệt:** Không có.

### 2.2.5. Usecase quản lý sinh viên của lịch thi

* **Mô tả vắn tắt:** Usecase này cho phép admin quản lý danh sách học viên cho từng lịch thi, bao gồm việc thêm hoặc xóa học viên..
* **Luồng sự kiện:**

1. Usecase bắt đầu khi Admin truy cập trang quản lý lịch thi, sau đó chọn môn học và thực hiện các thao tác: xem danh sách, tạo mới, xóa.
2. Xem danh sách: Admin kich vào nút “Quản lý lịch thi” trên màn hình trang chủ sau đó chọn môn học muốn xem lịch thi. Hệ thống sẽ hiển thị màn hình thông tin chi tiết gồm:ID, Họ và tên, Môn thi, Phòng thi, Điểm danh và các thao tác.
3. Thêm sinh viên: Admin kích nào nút “Thêm sinh viên vào lịch thi” trên cửa sổ quản lý lịch thi . Hệ thống hiển thị màn hình yêu cầu chọn sinh viên cần thêm và nhấn hoàn thành sau đó trang sẽ hiển thị danh sách sinh viên cập nhật vừa thêm.
4. Xóa sinh viên : Người dung kích vào nút “Xóa” trên một dòng tài khoản bất kỳ. Hệ thống hiển thị xác nhận “Yes or No?”. Người dùng nhấn nút “Yes”. Hệ thống cập nhật dữ liệu sản phẩm trong bảng EXAMS và hiển thị thông báo “Xóa thành công” và danh sách các sinh viên đã được cập nhật. Use case kết thúc.

* **Điểm mở rộng:** Không có.
* **Tiền điều kiện:** Người dung phải là Admin.
* **Hậu điều kiện:** Không có.
* **Yêu cầu đặc biệt:** Không có.

### 2.2.6. Usecase quản lý lịch thi

* **Mô tả vắn tắt:** Usecase này cho phép admin có thể thêm mới, chỉnh sửa hoặc xóa các thông tin chi tiết về các lịch thi trong hệ thống..
* **Luồng sự kiện:**

1. Usecase bắt đầu khi Admin truy cập trang quản lý lịch thi và thực hiện các thao tác: xem danh sách, tạo mới, chỉnh sửa, xóa.
2. Xem danh sách: Admin kich vào nút “Xem chi tiết” trên dòng lịch thi muốn xem .Hệ thống sẽ hiển thị màn hình thông tin chi tiết gồm:ID, , Họ và tên, Mã số người dùng, môn thi, phòng thi và trạng thái điểm danh.
3. Tạo mới : Admin kích nào nút “Tạo lịch thi” trên cửa sổ quản lý lịch thi . Hệ thống hiển thị màn hình yêu cầu nhập thông tin chi tiết gồm:môn thi, phòng thi, ngày thi, giám thị . Admin nhập đầy đủ thông tin và nhấn nút “Hoàn thành”. Hệ thống tự tạo tài khoản và lưu thông tin sản phẩm vào bảng EXAMS và hiển thị thông báo “Tạo lịch thi thành công!” và danh sách đã được cập nhật.
4. Sửa:Admin kích vào nút “Chỉnh sửa” trên một dòng tài khoản muốn sửa. Hệ thống lấy thông tin cũ của lịch thi và hiển thị lên màn hình trong Textbox. Admin nhập thông tin cần thay đổi và nhấn nút “Hoàn thành”. Hệ thống lưu thông tin sản phẩm mới trong bảng EXAMS sau đó hiển thị thông báo “Đã sửa lịch thi” và danh sách các lịch thi đã được cập nhật.
5. Xóa: Người dung kích vào nút “Xóa” trên một dòng lịch thi bất kỳ. Hệ thống hiển thị xác nhận “Yes or No?”. Người dùng nhấn nút “Yes”. Hệ thống cập nhật dữ liệu sản phẩm trong bảng EXAMS và hiển thị thông báo “Xóa thành công” và danh sách các lịch thi đã được cập nhật. Use case kết thúc.

* **Điểm mở rộng:** Không có.
* **Tiền điều kiện:** Người dung phải là Admin.
* **Hậu điều kiện:** Không có.
* **Yêu cầu đặc biệt:** Không có.

### 2.2.7. Usecase điểm danh sinh viên

* **Mô tả vắn tắt:** Usecase này cho phép giám thị sử dụng công nghệ nhận dạng khuôn mặt để tự động điểm danh sinh viên tham dự môn học hoặc kỳ thi.
* **Luồng sự kiện:**

1. Usecase bắt đầu khi giám thị truy cập trang điểm danh và thực hiện các thao tác chọn lịch thi và điểm danh
2. Giám thị bấm vào nút “ sử dụng camera” và bắt đầu điểm danh. Hệ thống sẽ nhận diện khuân mặt và đưa lên service để bắt đầu so sánh và đưa ra kết quả.Nếu khớp với khuân mặt dự thi sẽ hiển thị điểm danh thành công và ngược lại sẽ hiển thị là không có sinh viên được điểm danh.

* **Điểm mở rộng:** Không có.
* **Tiền điều kiện:** Người dung phải là giám thị hoặc admin.
* **Hậu điều kiện:** Không có.
* **Yêu cầu đặc biệt:** Không có.

## 2.3.Thiết kế cơ sở dữ liệu

### 2.3.1.Biểu đồ thực thể liên kết

1. **Thực Thể User:**

* Thuộc tính: id, email, phone, username, password, role, status, image, created\_at, updated\_at, class, student\_id.
* Khóa chính: id.

1. **Thực Thể Exam:**

* Thuộc tính: exam\_id, date, subject, room, invigilator\_1, invigilator\_2, invigilator\_3, invigilator\_4.
* Khóa chính: exam\_id.
* Khóa ngoại: invigilator\_1, invigilator\_2, invigilator\_3, invigilator\_4 (liên kết với id của thực thể User).

1. **Thực Thể Exam List:**

* Thuộc tính: exam\_id, user\_id, attendance.
* Khóa chính: exam\_id, user\_id.
* Khóa ngoại: exam\_id (liên kết với exam\_id của thực thể Exam), user\_id (liên kết với id của thực thể User).

1. **Thực Thể Password Reset Token:**

* Thuộc tính: id, user\_id, token, expires\_at, created\_at.
* Khóa chính: id.
* Khóa ngoại: user\_id (liên kết với id của thực thể User).

#### 2.3.1.1.Quan hệ giữa các thực thể

* User và Exam:
  + Một User có thể là giám sát viên (invigilator) của nhiều Exam.
  + Mối quan hệ: 1-N (Một User có thể giám sát nhiều Exam).
* Exam và Exam List:
  + Một Exam có nhiều Exam List (danh sách sinh viên tham gia).
  + Mối quan hệ: 1-N (Một Exam có nhiều Exam List).
* User và Exam List:
  + Một User có thể tham gia nhiều Exam và có nhiều Exam List.
  + Mối quan hệ: 1-N (Một User có nhiều Exam List).
* User và Password Reset Token:
  + Một User có thể có nhiều yêu cầu đặt lại mật khẩu.
  + Mối quan hệ: 1-N (Một User có nhiều Password Reset Token).

### 2.3.2. Thiết kế bảng

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Khóa** |
| exam\_id | int | NO | PRI, FK |
| user\_id | int | NO | PRI, FK |
| attendance | tinyint(1) | YES |  |

*Bảng 2. 2. Cơ sở dữ liệu của bảng exam\_list*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Khóa** |
| exam\_id | int | NO | PRI, AI |
| date | date | NO |  |
| subject | varchar(100) | NO |  |
| room | varchar(50) | NO |  |
| invigilator\_1 | int | YES | FK |
| invigilator\_2 | int | YES | FK |
| invigilator\_3 | int | YES | FK |
| invigilator\_4 | int | YES | FK |

*Bảng 2. 3. Cơ sở dữ liệu của exams*

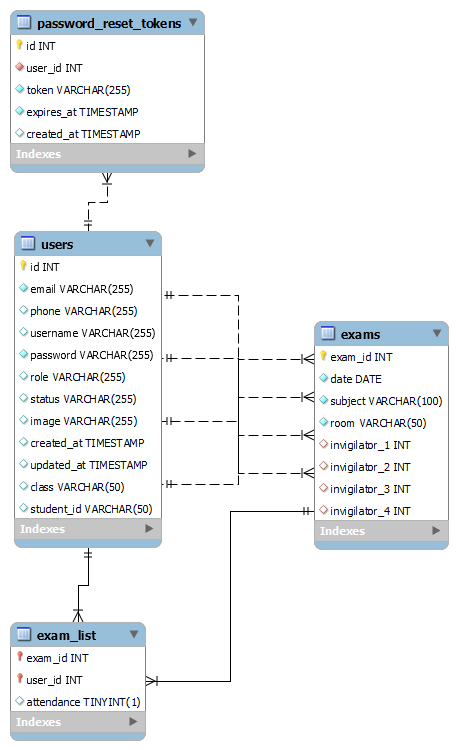
|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Khóa** |
| id | int | NO | PRI, AI |
| user\_id | int | NO | FK |
| token | varchar(255) | NO |  |
| expires\_at | timestamp | NO |  |
| created\_at | timestamp | YES |  |

*Bảng 2. 4. Cơ sở dữ liệu của password\_reset\_tokens*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu dữ liệu** | **Null** | **Khóa** |
| id | int | NO | PRI, AI |
| email | varchar(255) | NO | UNI |
| phone | varchar(255) | YES |  |
| username | varchar(255) | YES |  |
| password | varchar(255) | NO |  |
| role | varchar(255) | YES |  |
| status | varchar(255) | YES |  |
| image | varchar(255) | YES |  |
| created\_at | timestamp | YES |  |
| updated\_at | timestamp | YES |  |
| class | varchar(50) | YES |  |
| student\_id | varchar(50) | YES |  |

*Bảng 2. 5. Cơ sở dữ liệu của bảng user*

### 2.3.3.Quan hệ các bảng



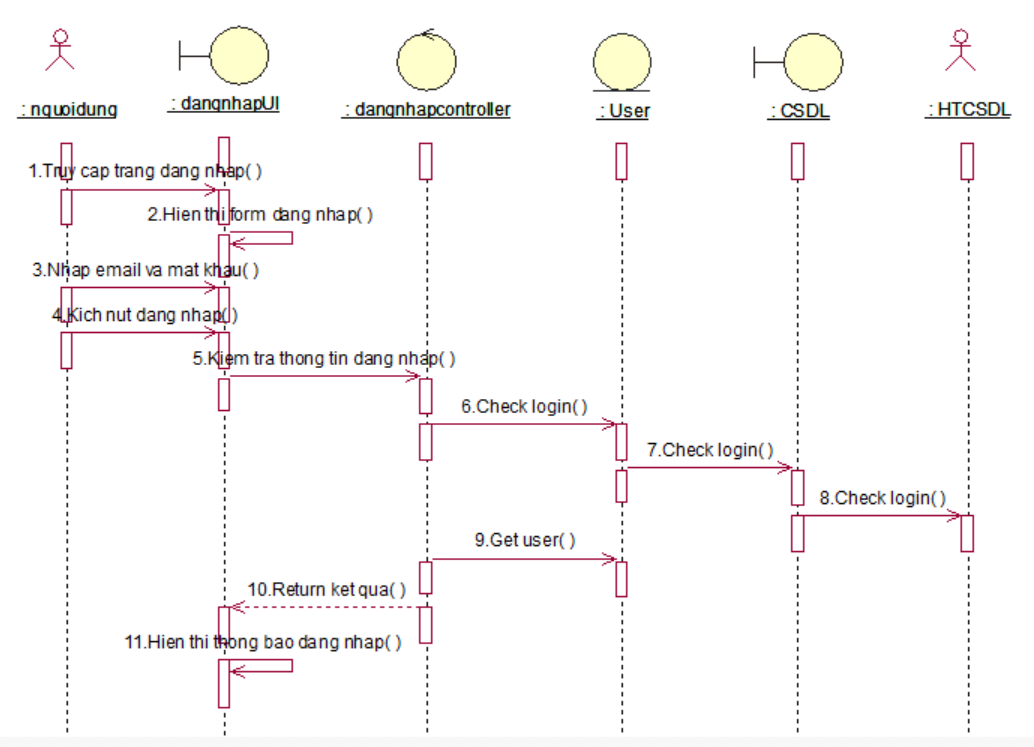
*Hình 2. 2.Quan hệ các bảng dữ liệu*

## 2.4.Phân tích các use case

### 2.4.1.Phân tích use case

#### 2.4.1.1.Đăng nhập

Biểu đồ trình tự:

****

*Hình 2. 3. Biểu đồ trình tự đăng nhập*

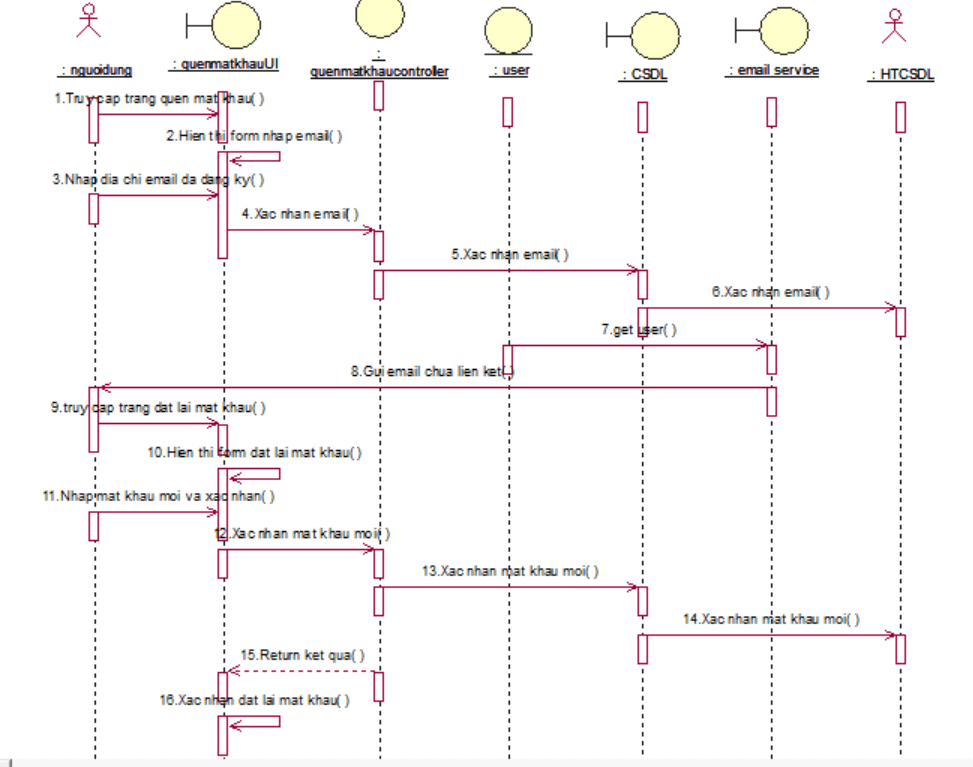
Biểu đồ lớp



*Hình 2. 4. Biểu đồ lớp đăng nhập*

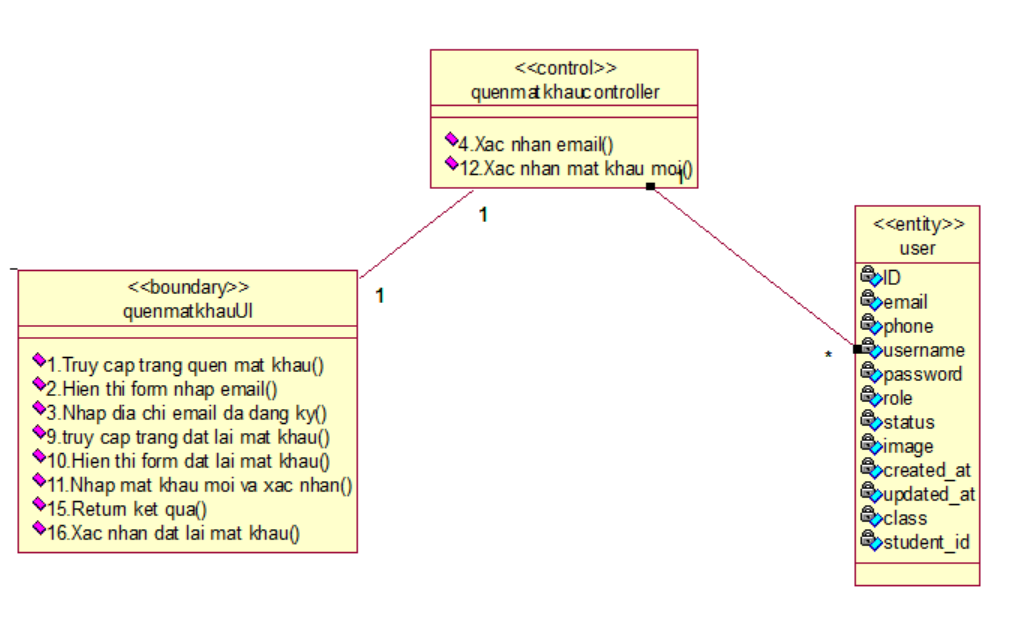
#### 2.4.1.2.Quên mật khẩu

Biểu đồ trình tự



*Hình 2. 5. Biểu đồ trình tự quên mật khẩu*

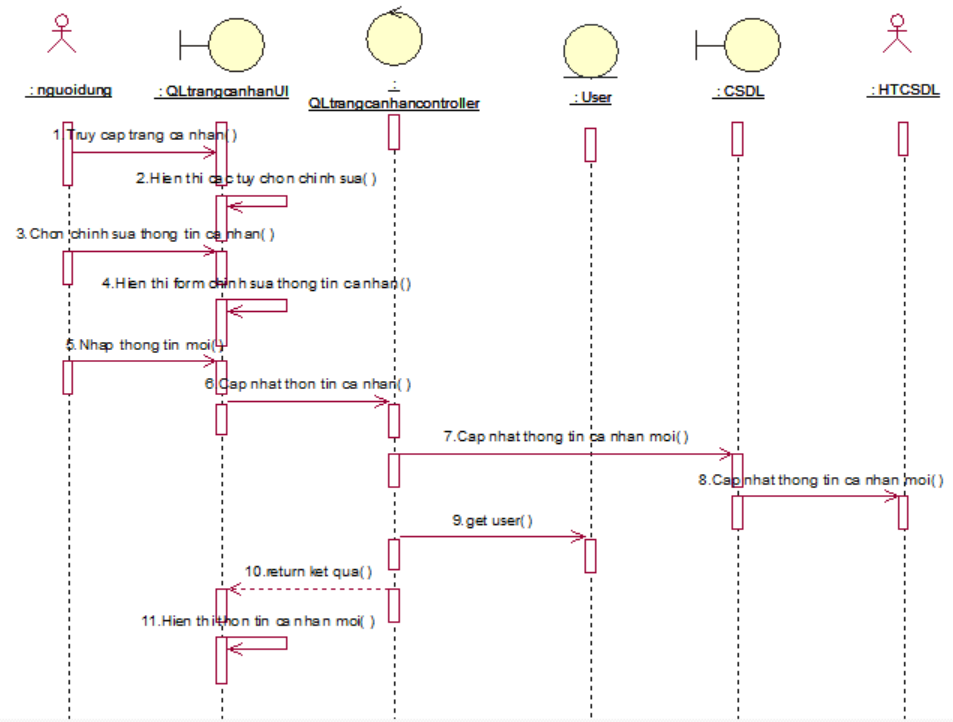
Biểu đồ lớp



*Hình 2. 6. Biểu đồ lớp quên mật khẩu*

#### 2.4.1.3.Quản lý trang cá nhân

Biểu đồ tuần tự



*Hình 2. 7.Biểu đồ tuần tự quản lý trang cá nhân*

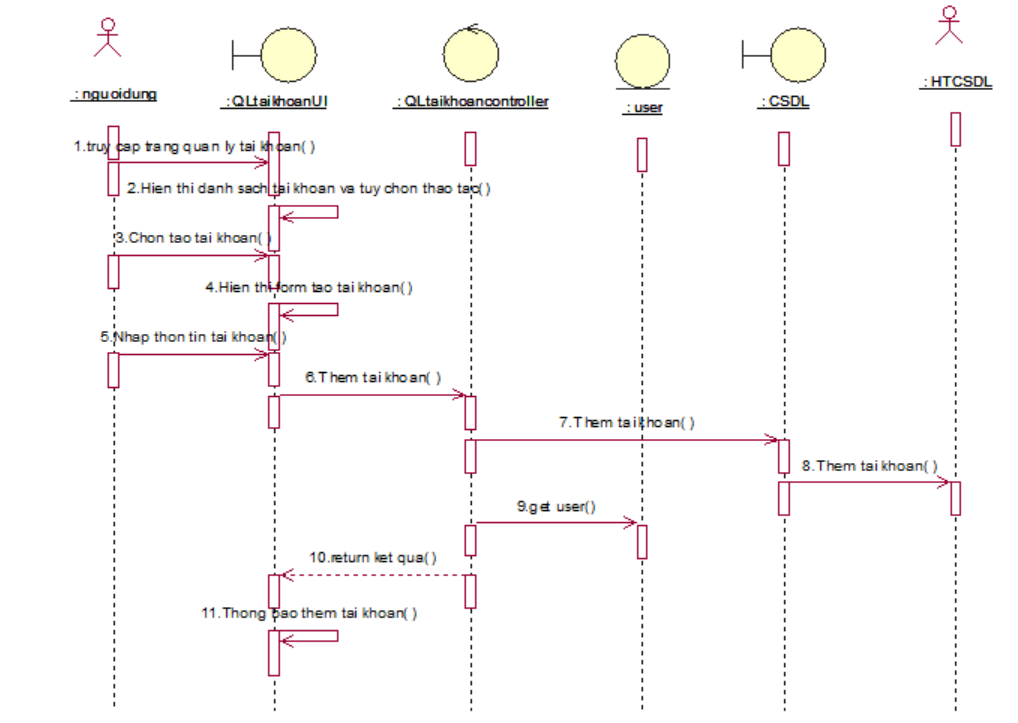
Biểu đồ lớp



*Hình 2. 8. Biểu đồ lớp quản lý trang cá nhân*

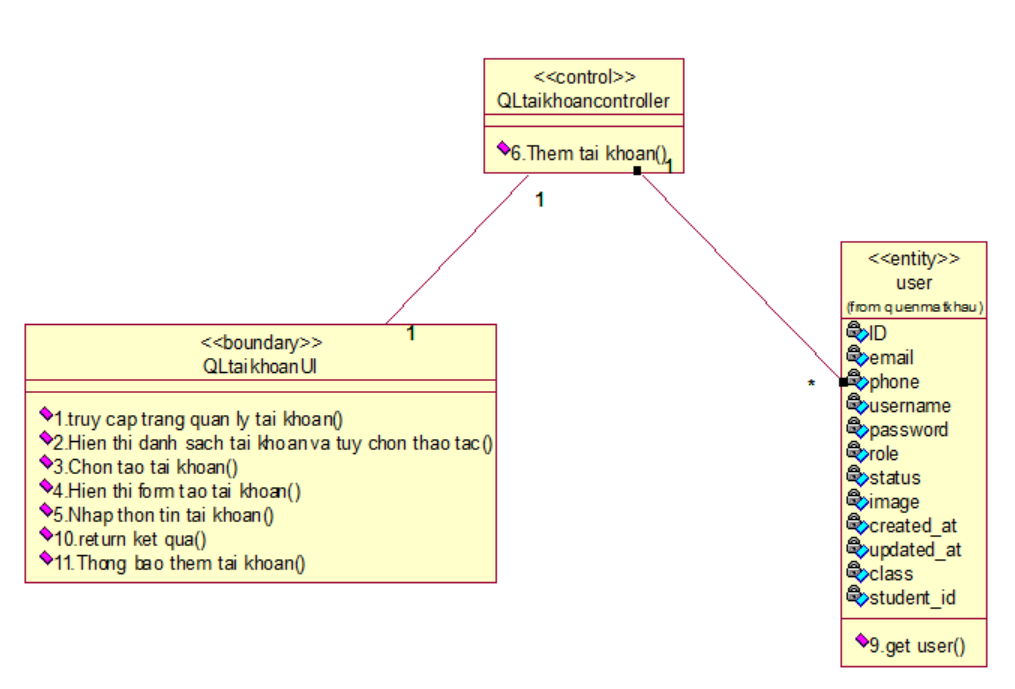
#### 2.4.1.4.Quản lý tài khoản

Biểu đồ tuần tự



*Hình 2. 9.Biểu đồ tuần tự quản lý tài khoản*

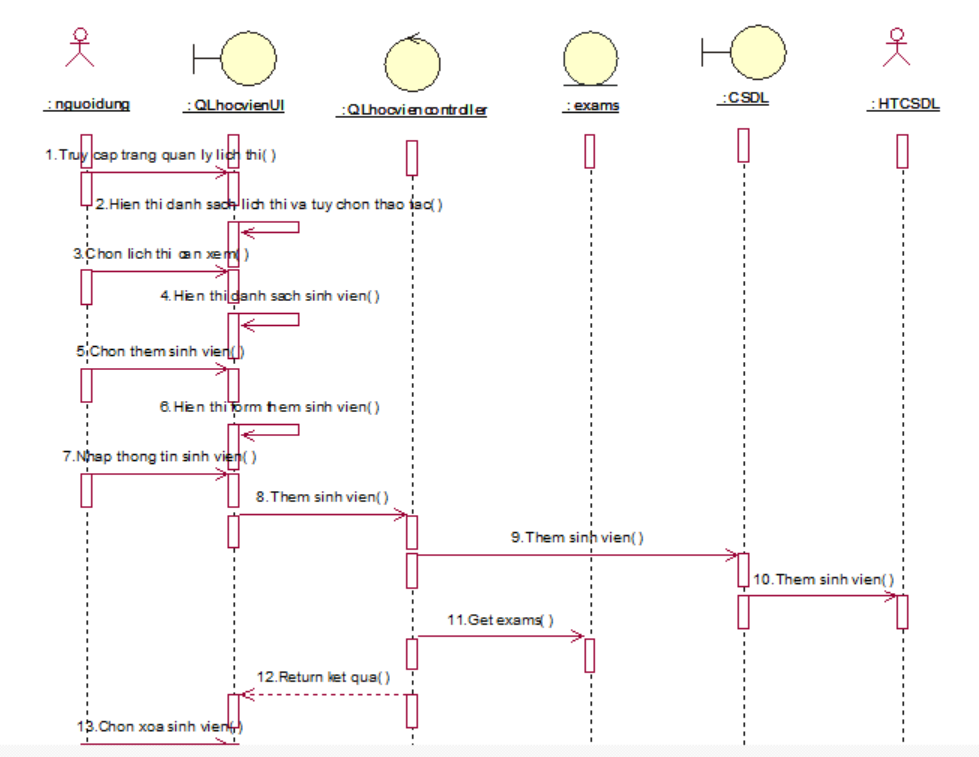
Biểu đồ lớp



*Hình 2. 10.Biểu đồ lớp quản lý tài khoản*

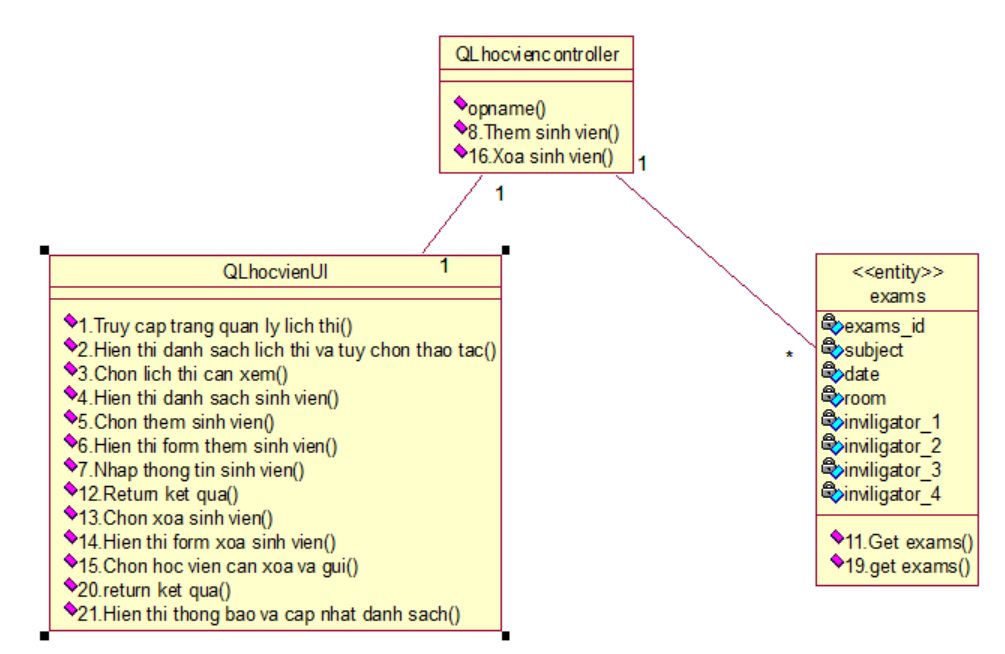
#### 2.4.1.5.Quản lý sinh viên của lịch thi

Biểu đồ tuần tự



*Hình 2. 11.Biểu đồ tuần tự quản lý sinh viên của lịch thi*

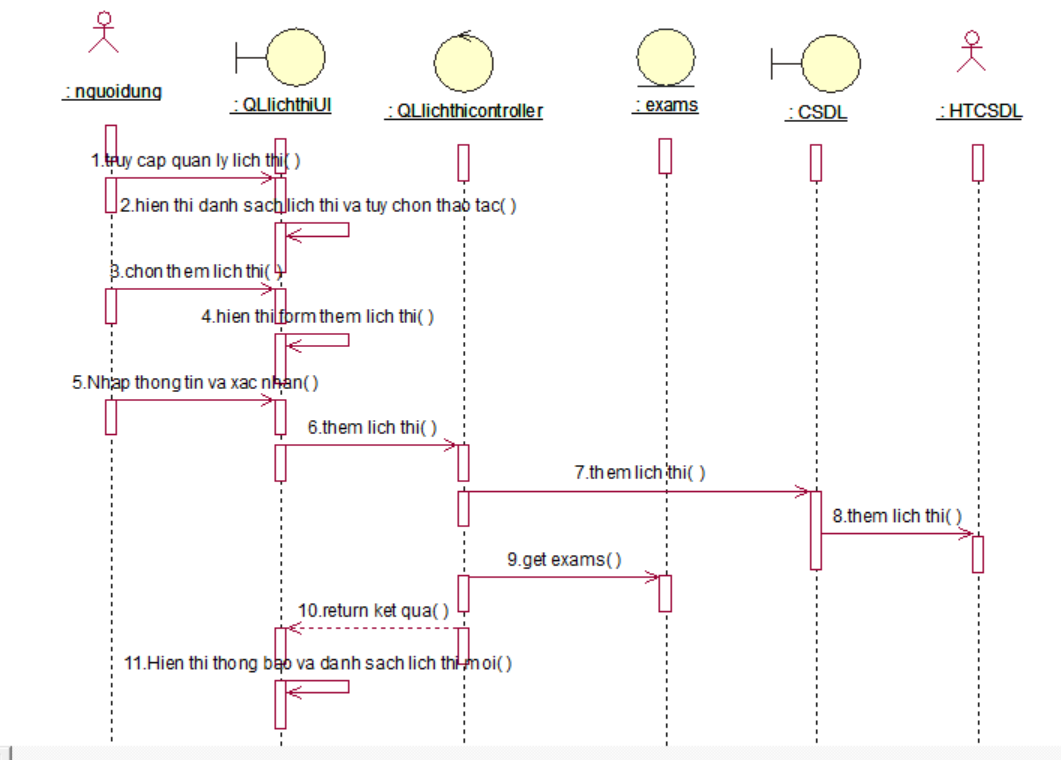
Biểu đồ lớp



*Hình 2. 12.Biểu đồ lớp quản lý sinh viên của lịch thi*

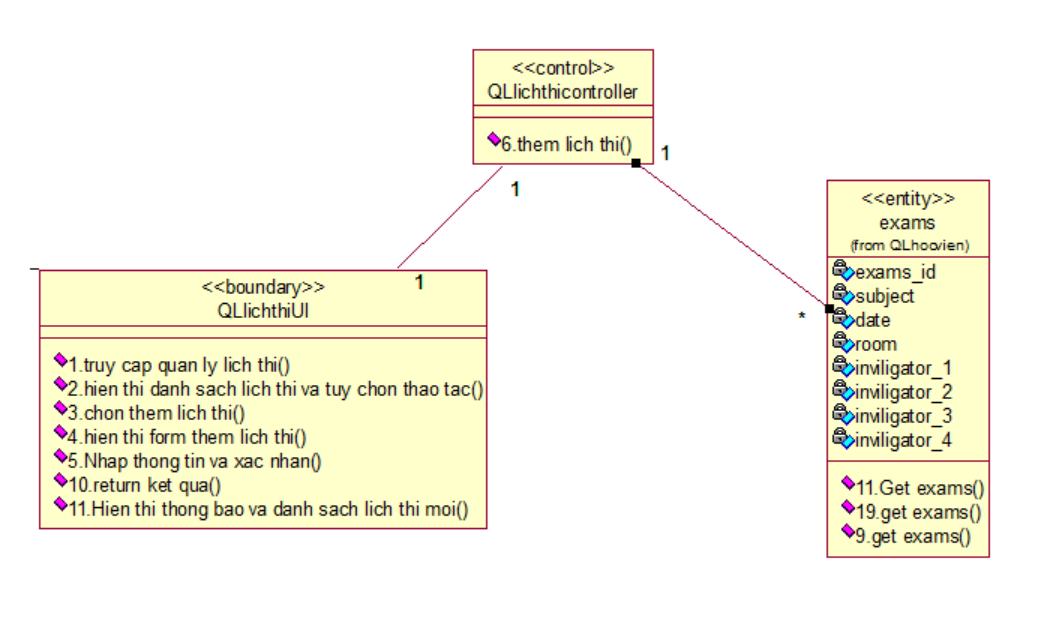
#### 2.4.1.6.Quản lý lịch thi

Biểu đồ tuần tự



*Hình 2. 13.Biểu đồ tuần tự quản lý lịch thi*

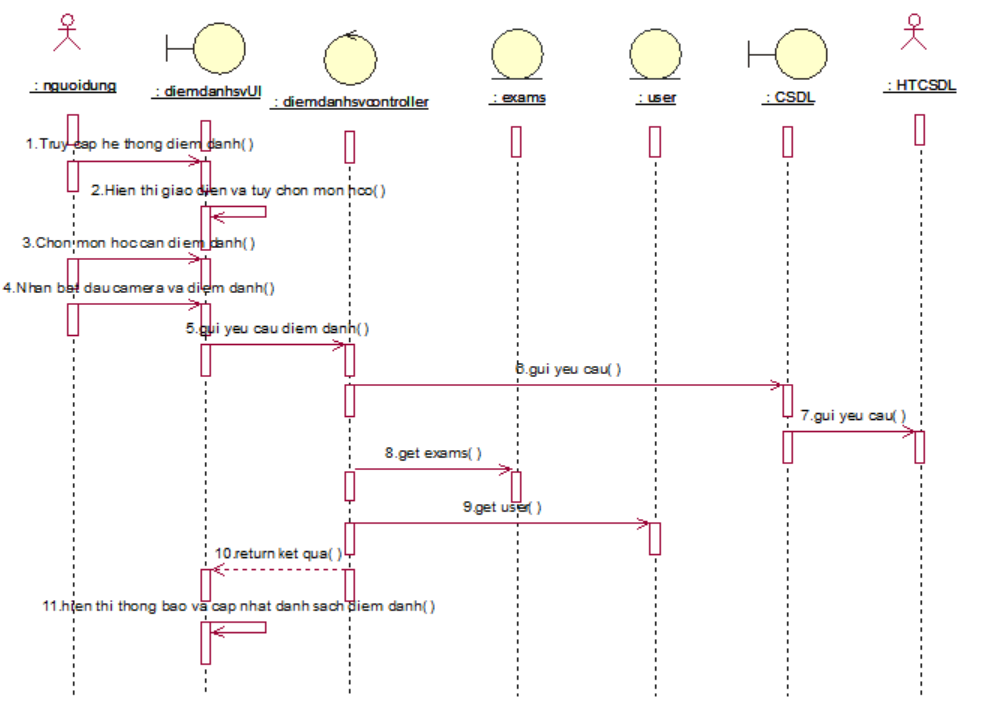
Biểu đồ lớp



*Hình 2. 14.Biểu đồ lớp quản lý lịch thi*

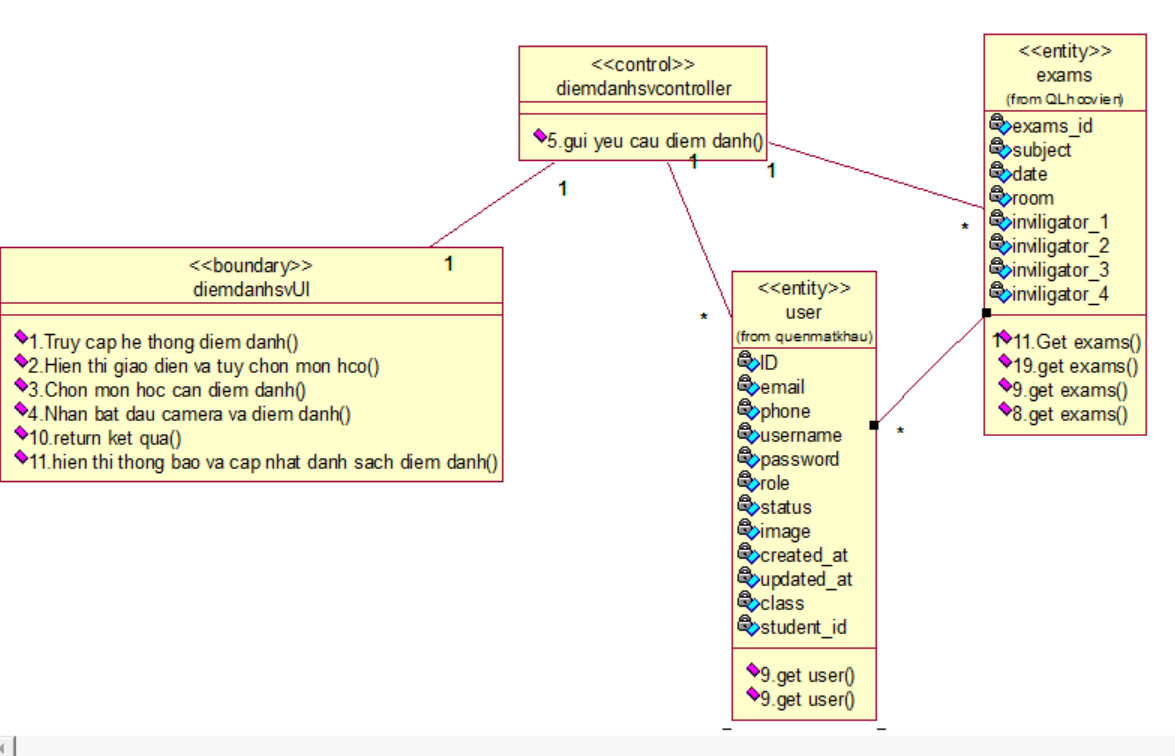
#### 2.4.1.7.Điểm danh sinh viên

Biểu đồ tuần tự



*Hình 2. 15.Biểu đồ tuần tự điểm danh sinh viên*

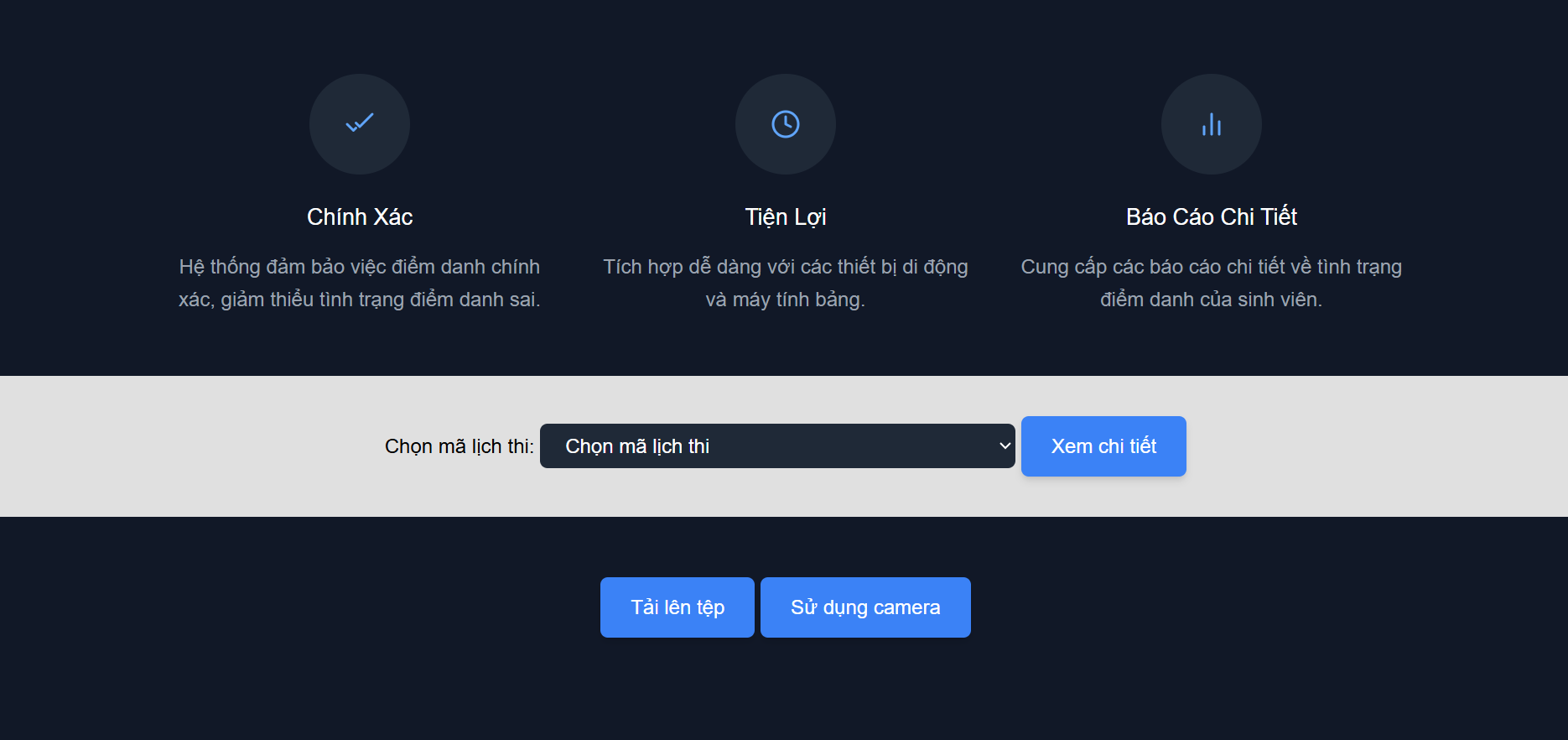
Biểu đồ lớp



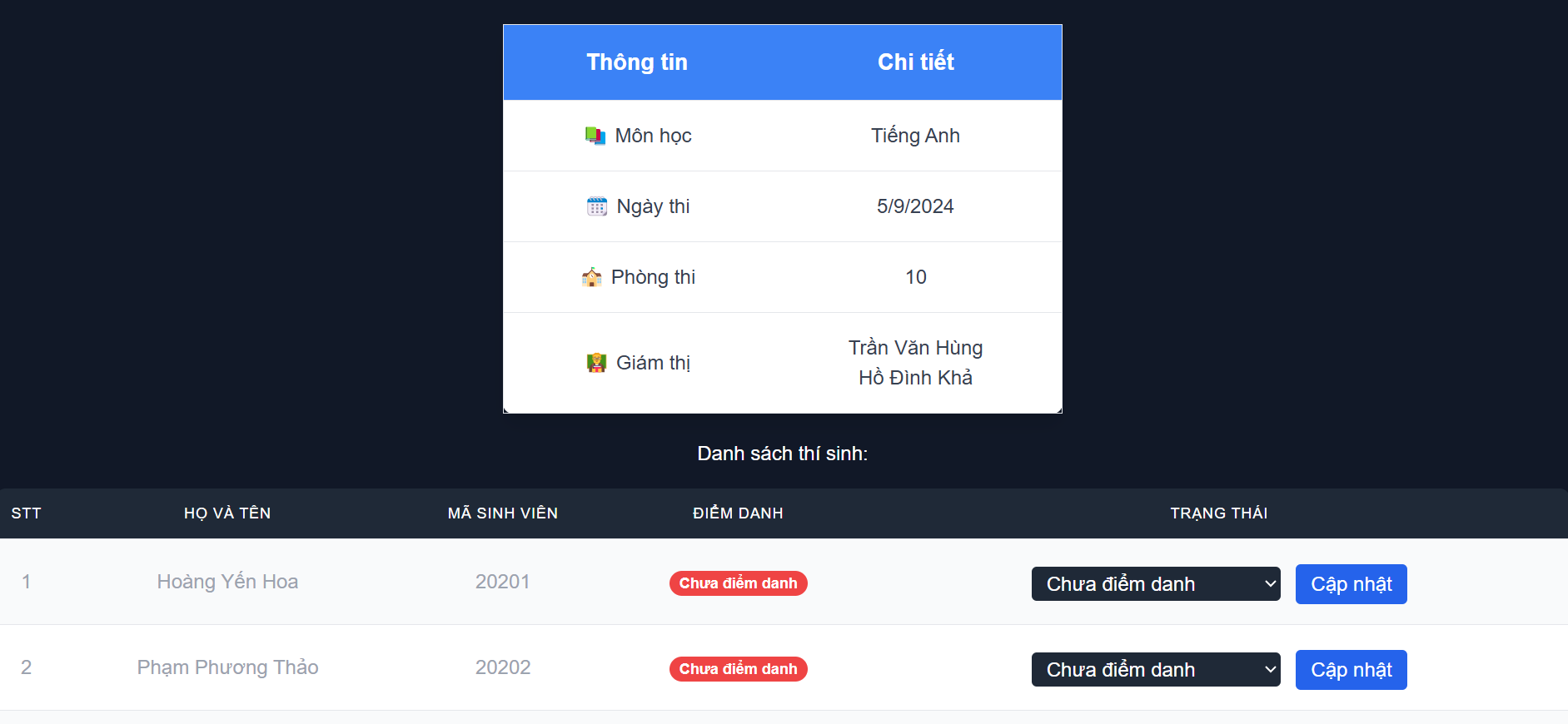
*Hình 2. 16.Biểu đồ lớp điểm danh sinh viên*

# CHƯƠNG 3. THIẾT KẾ GIAO DIỆN, CÀI ĐẶT VÀ KIỂM THỬ

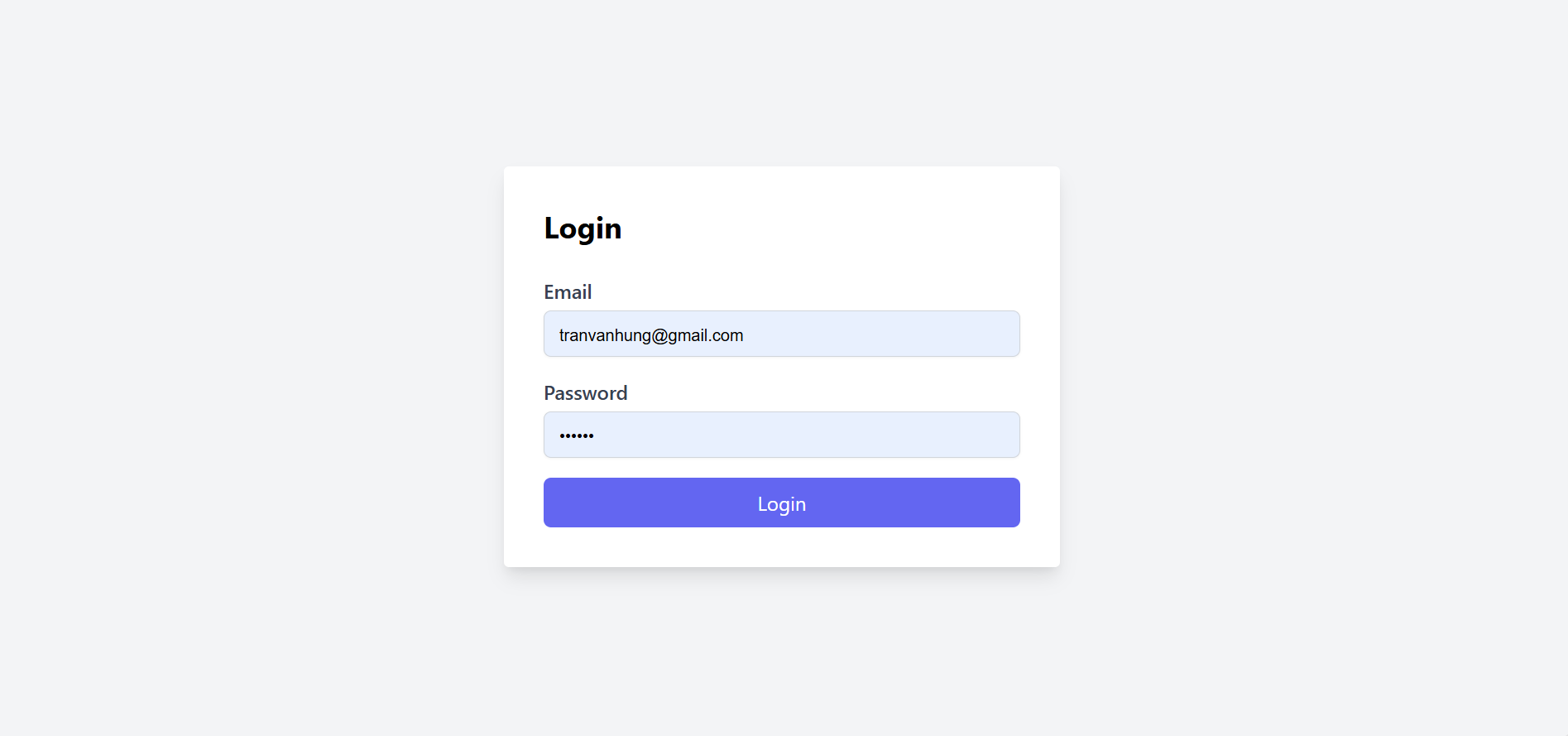
## 3.1.Thiết kế giao diện



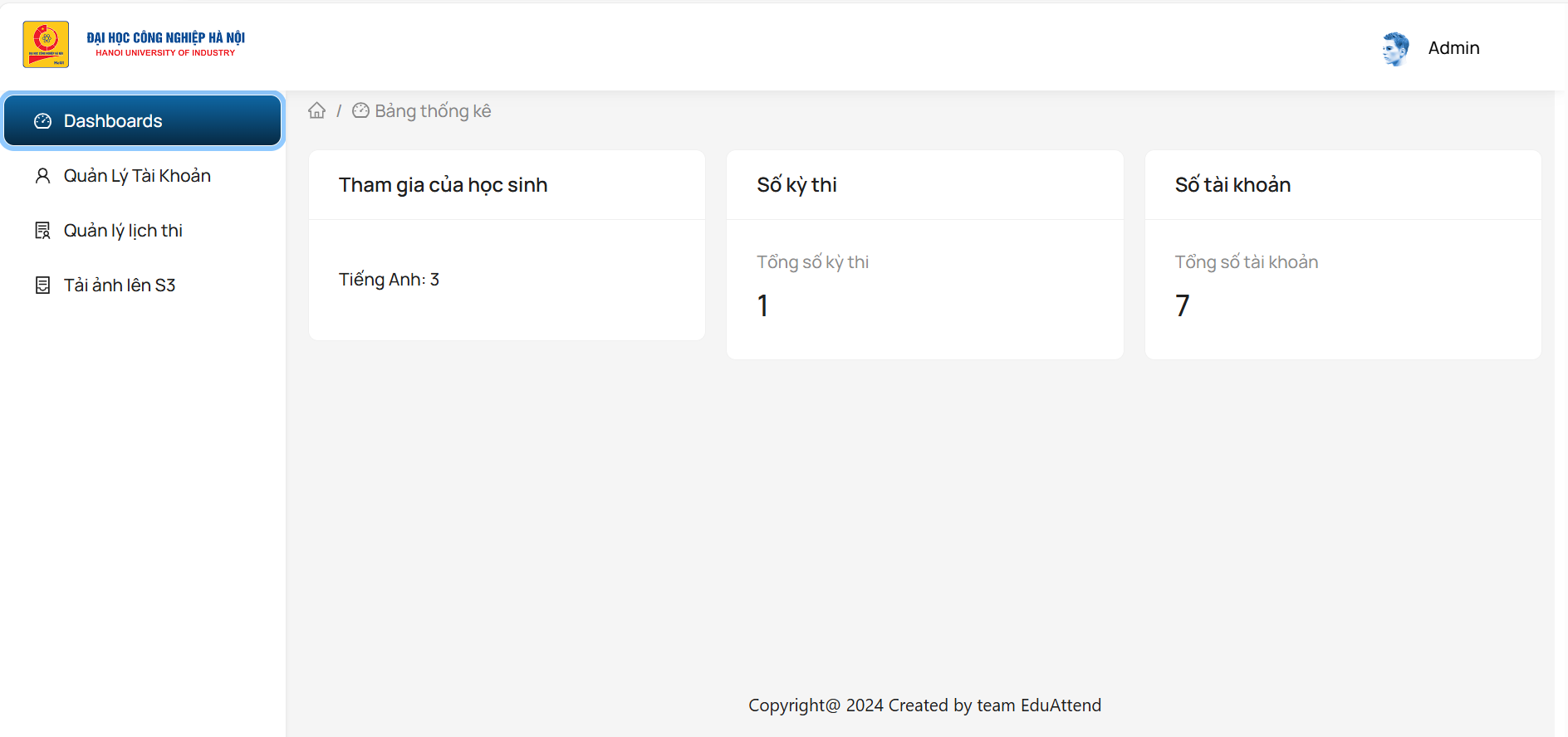
*Hình 3. 1. Giao diện chọn lịch thi*

**

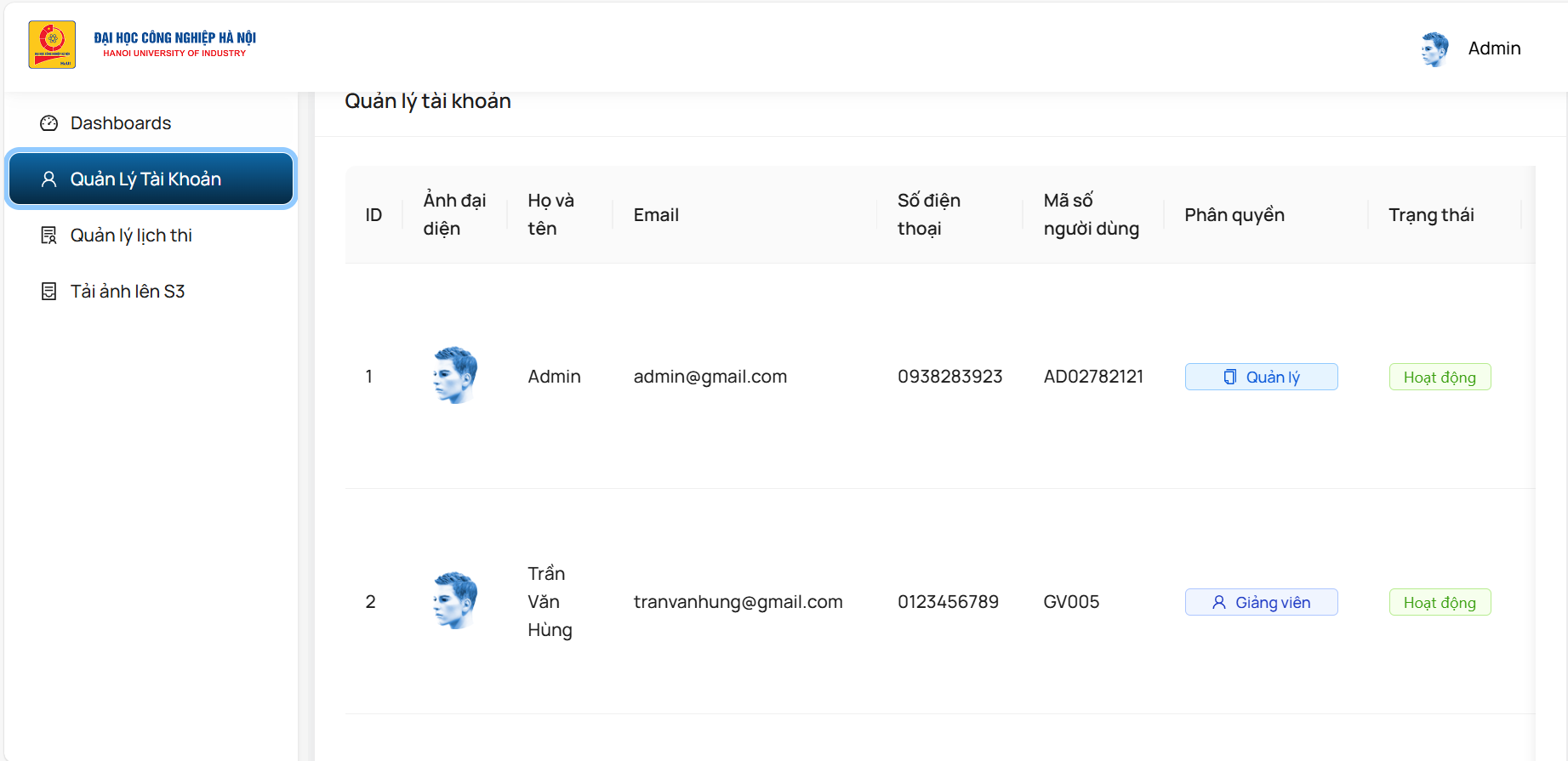
*Hình 3. 2.Giao diện điểm danh*

**

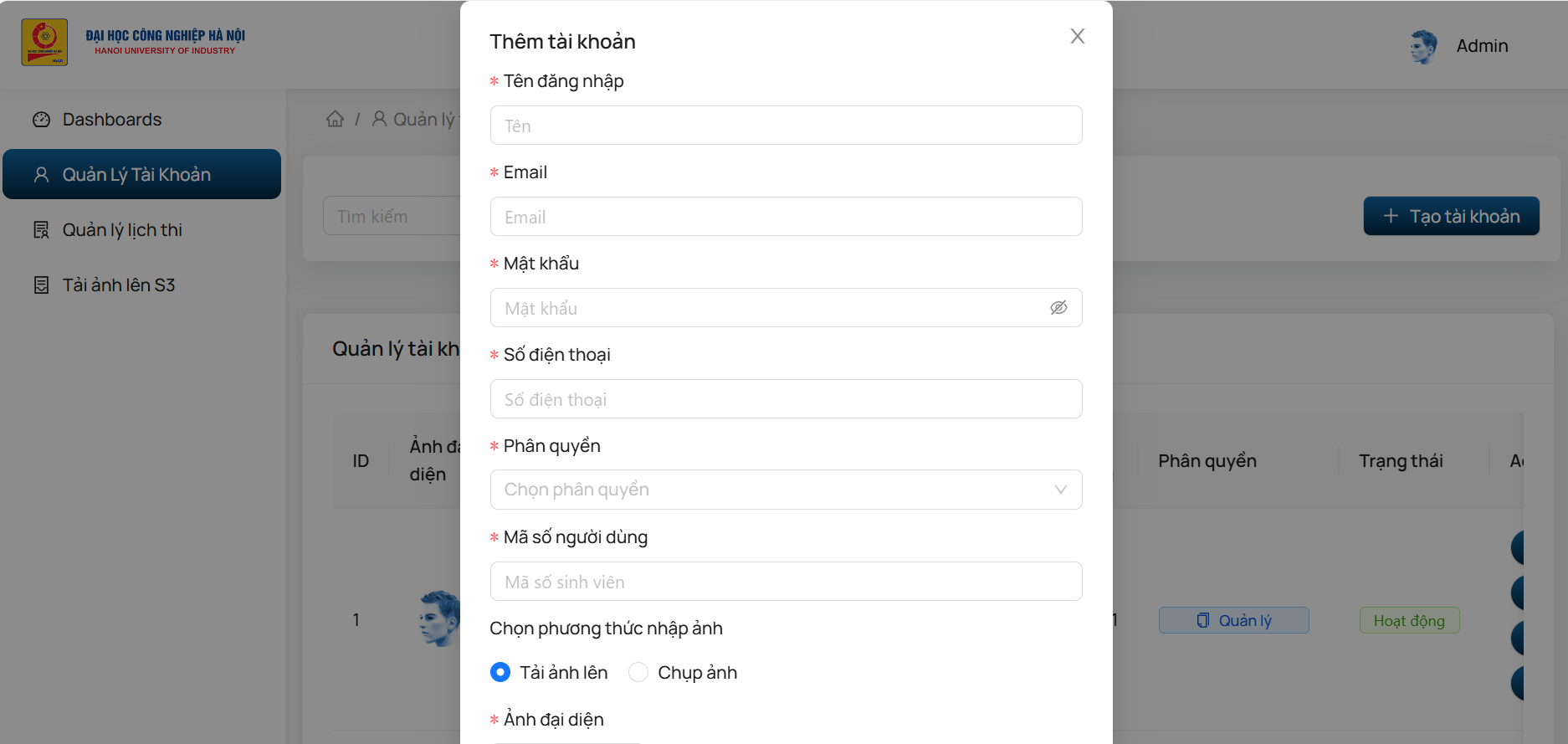
*Hình 3. 3.Giao diện đăng nhập cho Client*

**

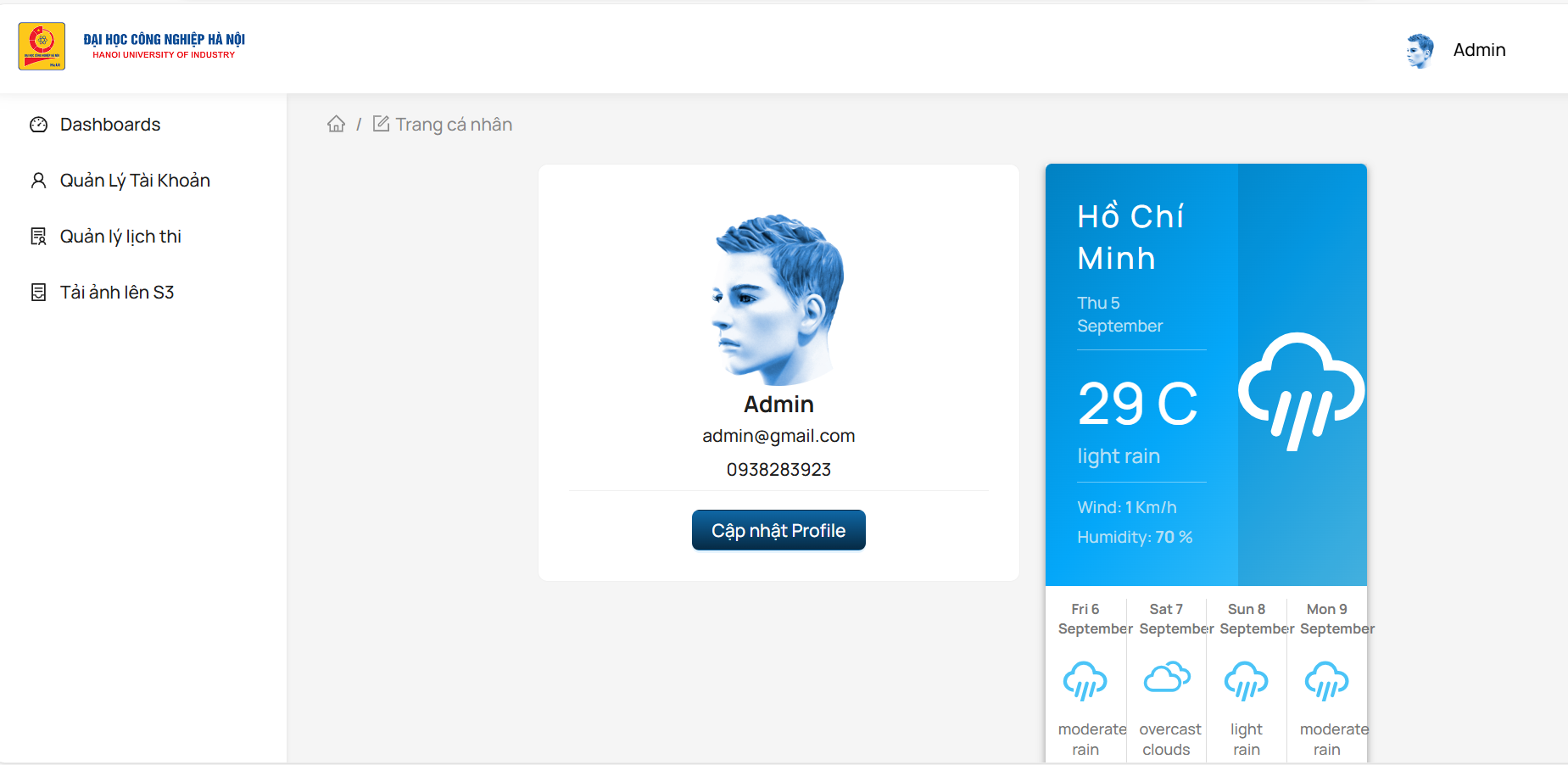
*Hình 3. 4.Giao diện thống kê*

**

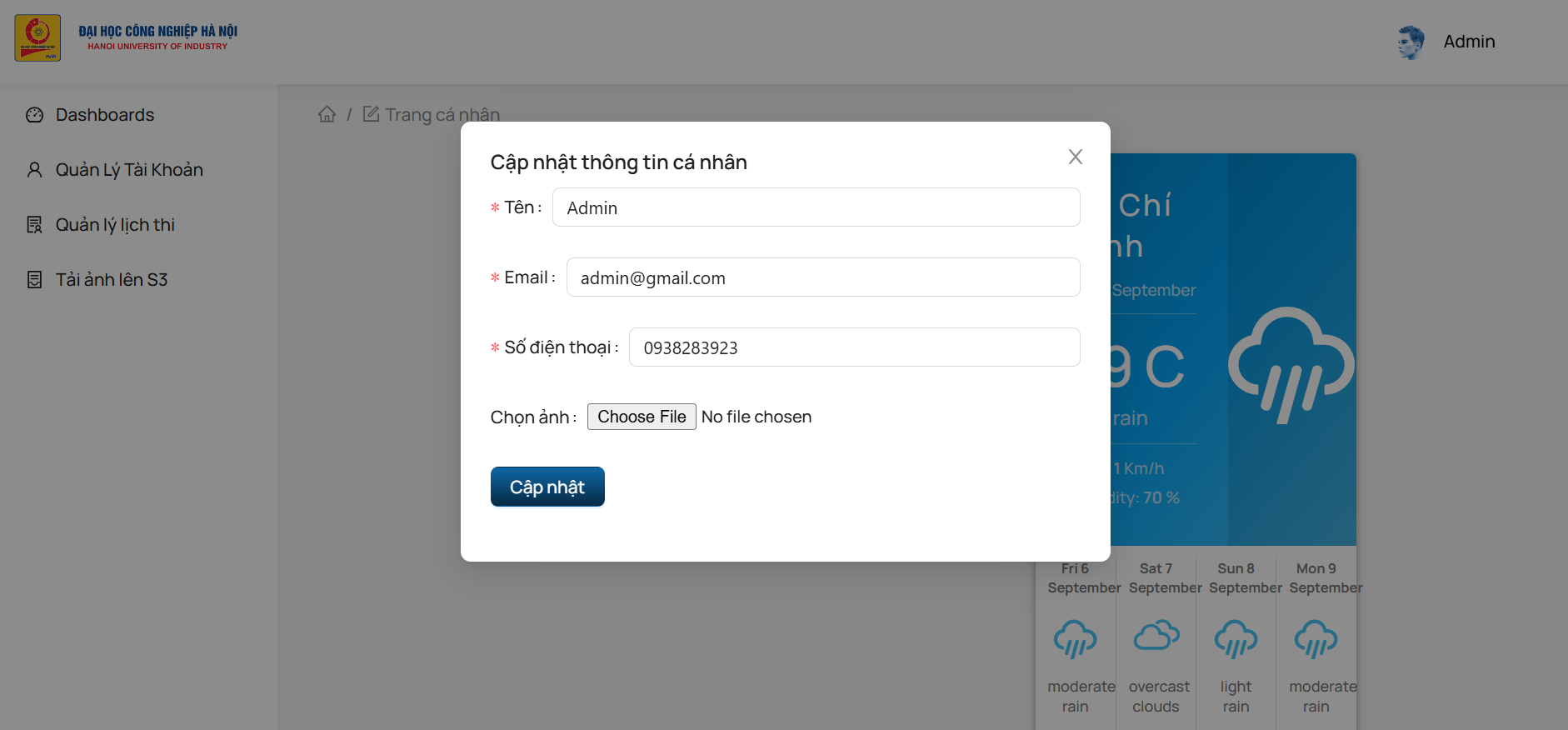
*Hình 3. 5.Giao diện quản lý tài khoản*

**

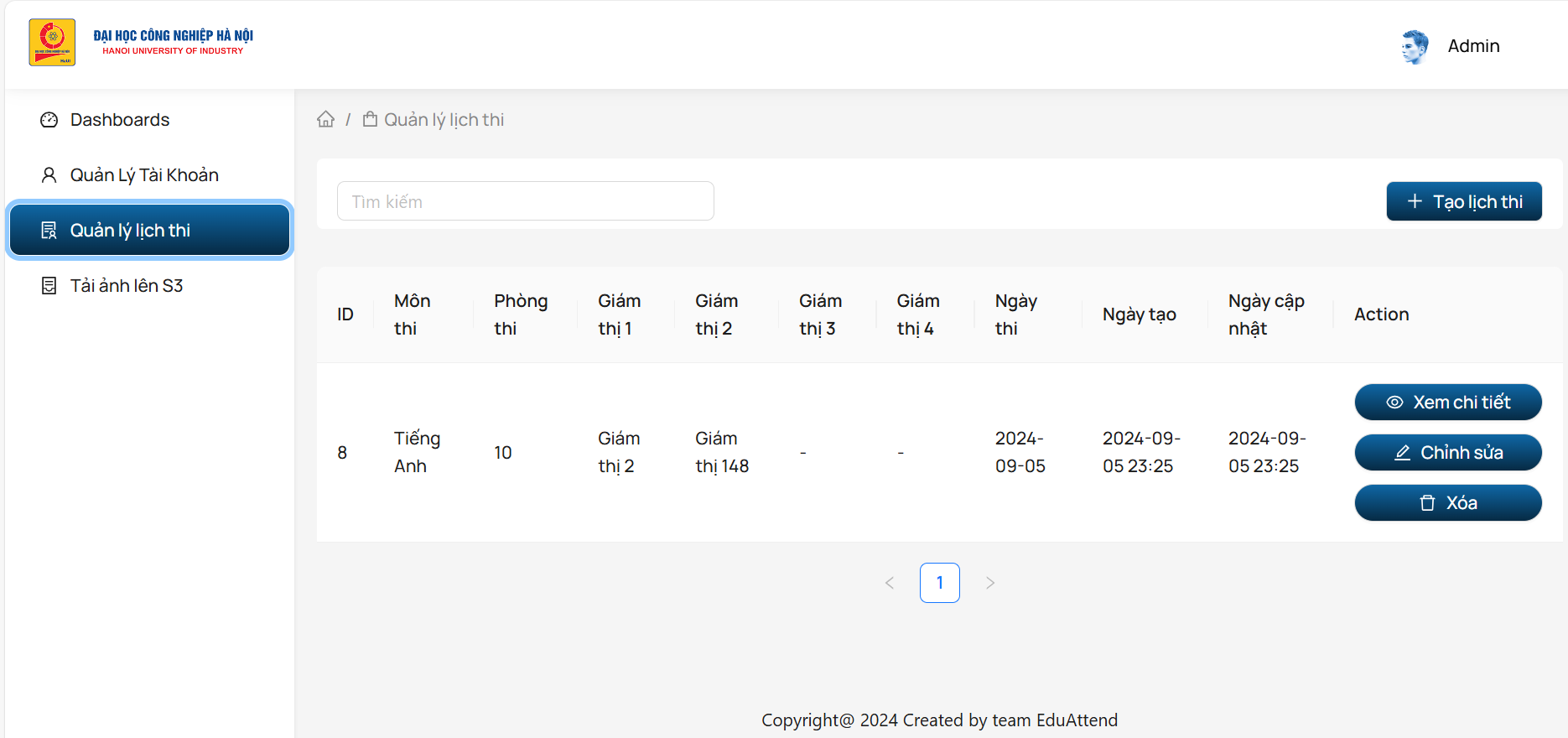
*Hình 3. 6.Giao diện tạo tài khoản*

**

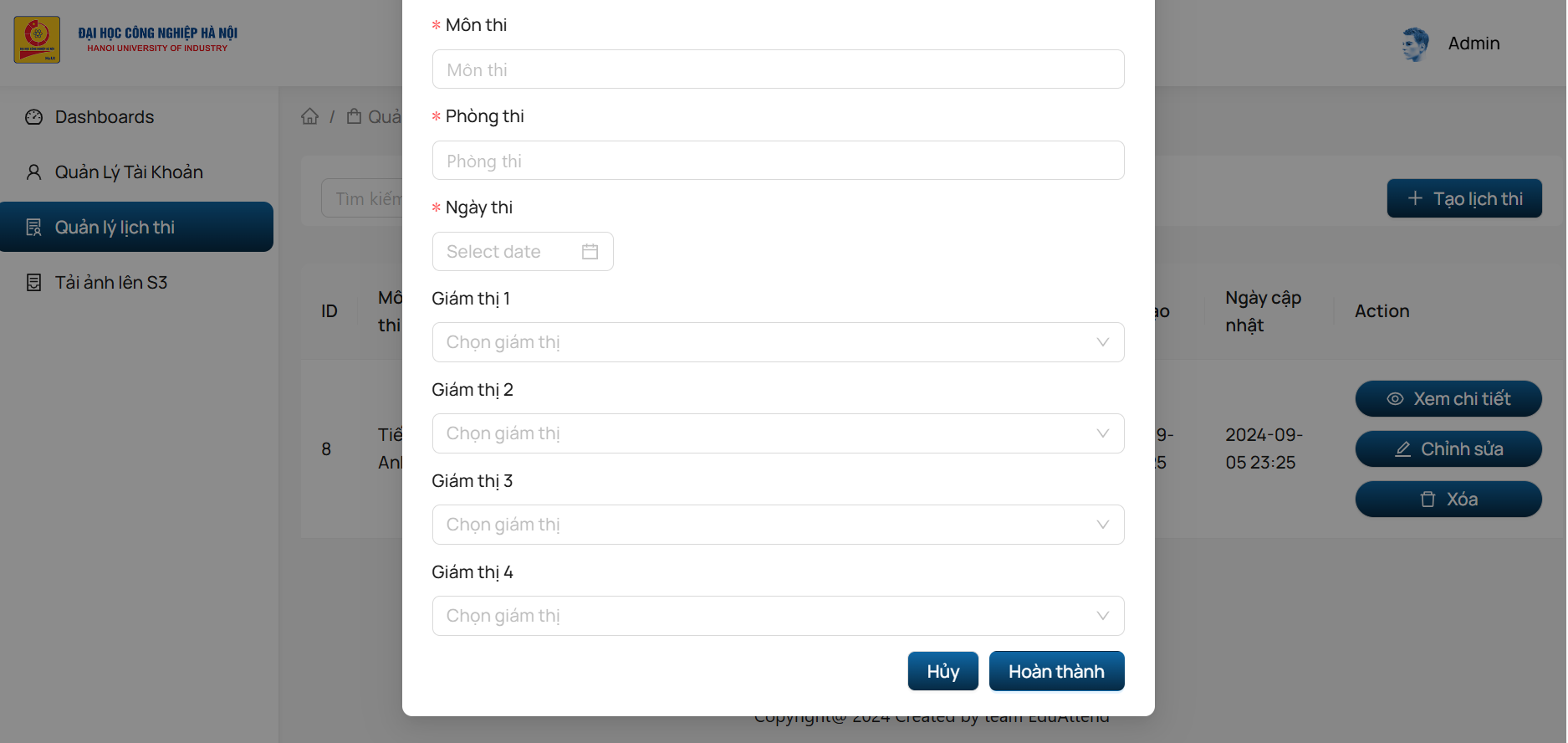
*Hình 3. 7.Giao diện trang cá nhân*

**

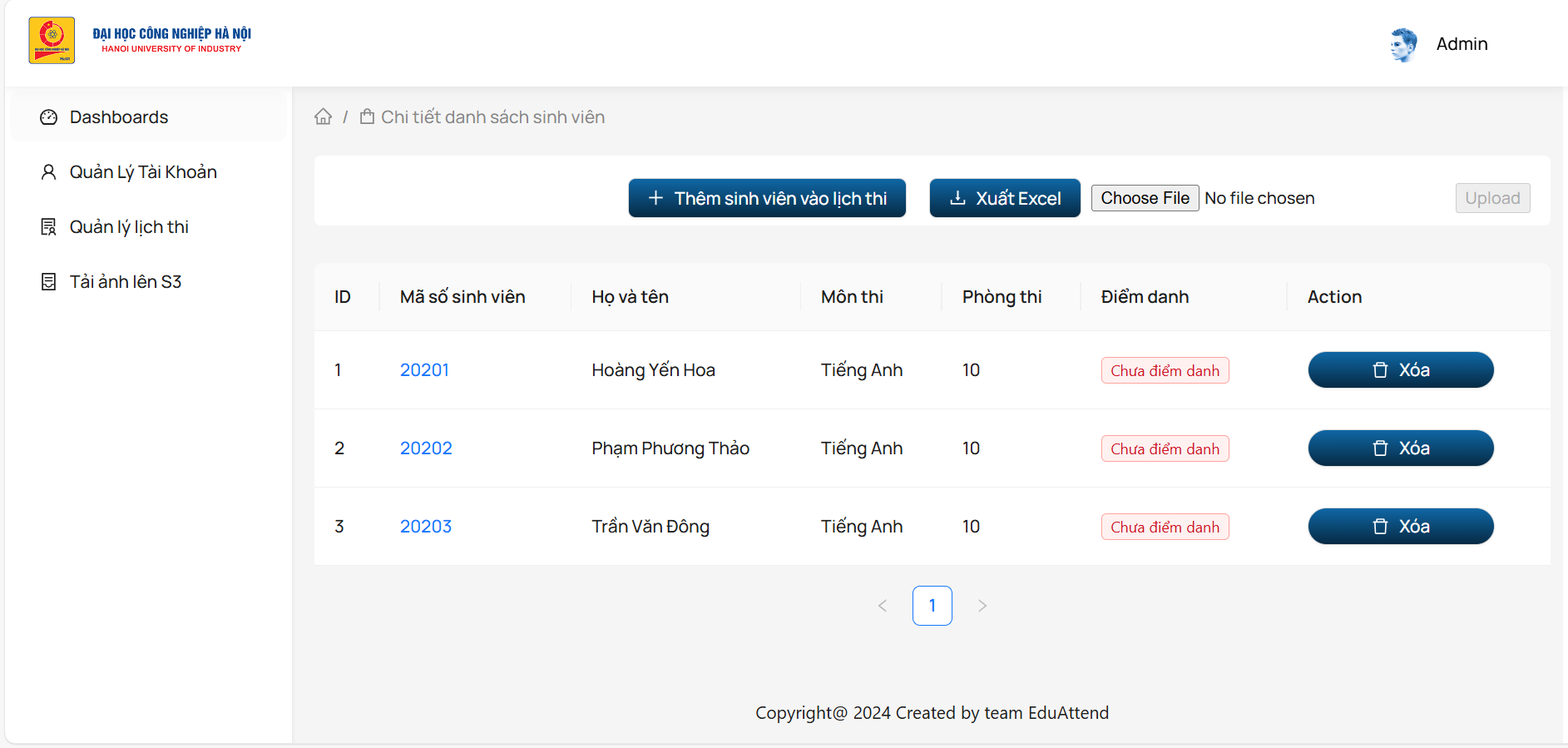
*Hình 3. 8.Giao diện cập nhật tài khoản*

**

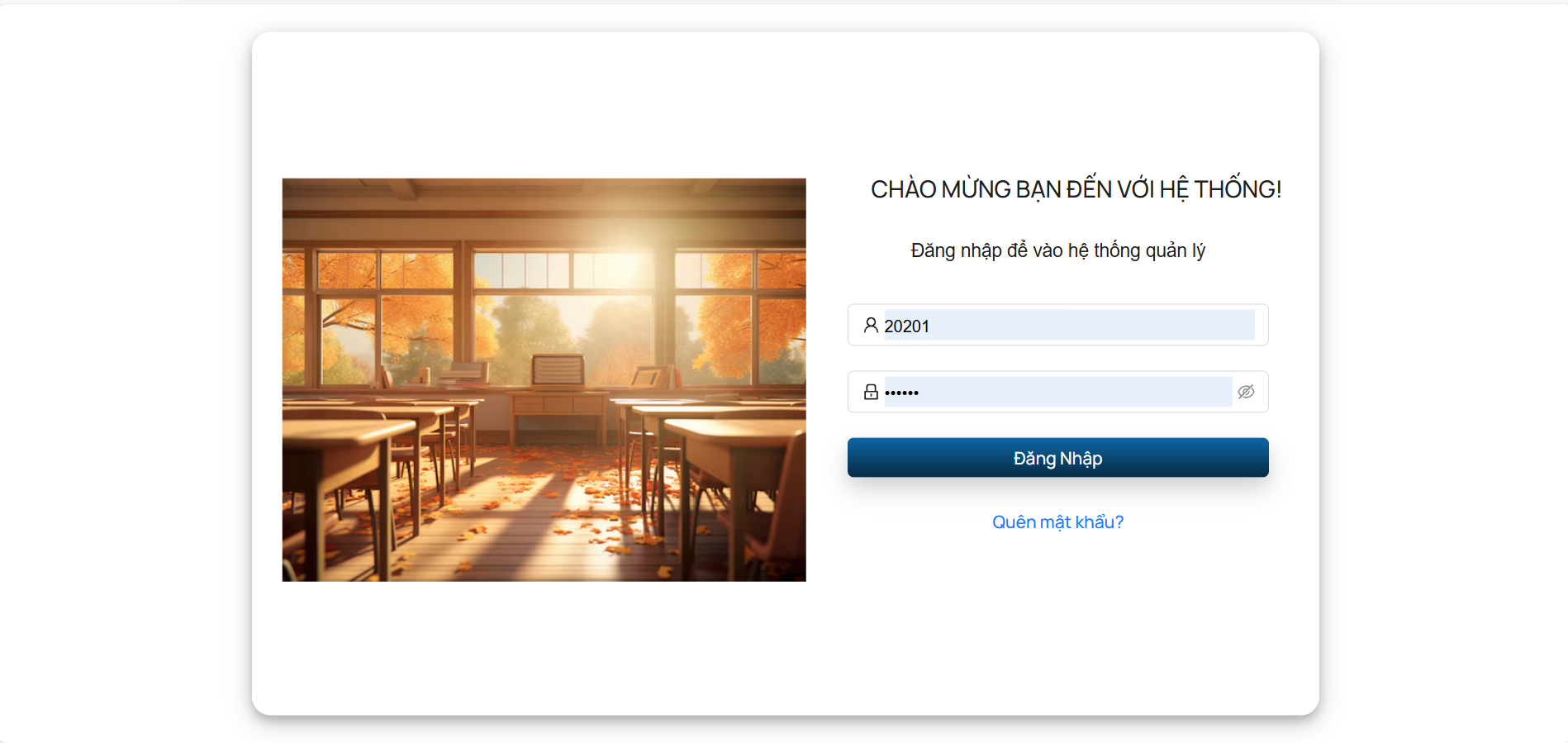
*Hình 3. 9.Giao diện quản lý lịch thi*

**

*Hình 3. 10.Giao diện tạo lịch thi*

**

*Hình 3. 11.Giao diện chi tiết lịch thi và danh sách sinh viên*

**

*Hình 3. 12.Giao diện đăng nhập*

## 3.2.Giới thiệu ngôn ngữ cài đặt

### 3.2.1.Giới thiệu về React JS

React.js, thường gọi tắt là React, là một thư viện JavaScript mạnh mẽ được phát triển bởi Facebook và cộng đồng open-source. Nó được sử dụng rộng rãi trong việc xây dựng giao diện người dùng hiện đại trên web và ứng dụng di động. Đặc trưng nổi bật của React chính là việc quản lý giao diện người dùng một cách hiệu quả và linh hoạt [1].

React tập trung vào việc xây dựng các thành phần (components) riêng lẻ, mỗi thành phần đại diện cho một phần của giao diện người dùng. Điều này giúp tạo ra mã nguồn dễ đọc, tái sử dụng và bảo trì. React sử dụng JSX (JavaScript XML) để định nghĩa giao diện người dùng, cho phép lập trình viên kết hợp mã JavaScript và HTML một cách tự nhiên.

Một trong những điểm đặc biệt của React là sử dụng Virtual DOM (Document Object Model ảo) để cải thiện hiệu suất ứng dụng. Thay vì cập nhật toàn bộ DOM mỗi khi có sự thay đổi, React tạo một bản sao ảo của DOM và so sánh nó với DOM thực tế, chỉ cập nhật những phần thay đổi thực sự. Điều này làm giảm tải cho trình duyệt và tăng tốc độ render, giúp ứng dụng chạy nhanh hơn và mượt mà hơn.

React cung cấp cách quản lý trạng thái (state) của ứng dụng một cách hiệu quả thông qua props và state của các thành phần. Điều này giúp xây dựng các ứng dụng tương tác và phản hồi nhanh chóng.

Ngoài ra, React được hỗ trợ mạnh mẽ từ cộng đồng, với hàng ngàn các thư viện và công cụ bổ sung, cũng như tài liệu phong phú và ví dụ thực tế. Điều này làm cho việc học và sử dụng React trở nên dễ dàng và có thể áp dụng trong nhiều loại dự án khác nhau.

#### 3.2.1.1.Ưu điểm của React JS

React.js, một thư viện JavaScript phổ biến cho việc xây dựng giao diện người dùng, mang đến một loạt ưu điểm đáng chú ý. Dưới đây là một đoạn văn mô tả các ưu điểm của React.js:

React.js được biết đến với tính linh hoạt và hiệu suất cao. Một trong những ưu điểm lớn nhất của React là khả năng tái sử dụng thành phần. Thư viện này cho phép bạn xây dựng các thành phần (components) riêng lẻ có khả năng tái sử dụng cao, từ các button đơn giản đến các thành phần phức tạp như bảng điều khiển. Việc này giúp tối ưu hóa quá trình phát triển và duy trì mã nguồn, giảm thiểu lặp lại và tăng cường tính nhất quán trong giao diện người dùng.

Khả năng làm việc chặt chẽ với Virtual DOM cũng là một ưu điểm nổi bật của React. Virtual DOM giúp cải thiện hiệu suất ứng dụng bằng cách chỉ cập nhật những phần thay đổi thực sự trong giao diện, giảm tải cho trình duyệt và tăng tốc độ render. Điều này đồng nghĩa với việc ứng dụng React có thể chạy nhanh hơn và mượt mà hơn, đặc biệt là trên các ứng dụng web lớn và phức tạp.

React cung cấp sự quản lý trạng thái (state) mạnh mẽ thông qua props và state của các thành phần. Điều này giúp dễ dàng theo dõi và kiểm soát dữ liệu trong ứng dụng, từ đó tạo ra các ứng dụng có khả năng tương tác cao và phản hồi nhanh chóng.

Sự hỗ trợ mạnh mẽ từ cộng đồng là một điểm mạnh khác của React. Cộng đồng React lớn mạnh, và có sẵn hàng ngàn các thư viện và công cụ bổ sung, giúp bạn giải quyết các vấn đề phát triển một cách nhanh chóng và hiệu quả. React cũng có tài liệu đầy đủ và các ví dụ thực tế, giúp bạn nắm bắt nhanh chóng cách sử dụng nó trong dự án của mình.

#### 3.2.1.2.Nhược điểm của React JS

React.js, một thư viện JavaScript phổ biến dùng để xây dựng giao diện người dùng, mang lại nhiều ưu điểm đáng chú ý. Tuy nhiên, với những lợi ích này, React.js cũng đi kèm với một số nhược điểm cần được xem xét.

Trước hết, việc học và làm quen với ReactJjs có thể đầy thách thức, đặc biệt đối với những người mới bắt đầu trong lĩnh vực phát triển web hoặc không có kinh nghiệm với JavaScript. JSX (JavaScript XML) và việc làm việc với các component là những khái niệm mới mẻ có thể gây khó khăn ban đầu.

Một nhược điểm khác là bạn thường cần sử dụng nhiều công cụ và thư viện bổ sung như Webpack, Babel, Redux, hoặc React Router để phát triển ứng dụng ReactJs hoàn chỉnh. Điều này có thể làm tăng độ phức tạp của dự án và đòi hỏi sự hiểu biết rộng rãi về hệ sinh thái của React.

Mặc dù React.js được thiết kế để cải thiện hiệu suất ứng dụng web, nhưng hiệu suất của ứng dụng React có thể không đủ trong các trường hợp của ứng dụng lớn hoặc khi không được tối ưu hóa một cách đúng cách. Điều này có thể dẫn đến tình trạng ứng dụng chạy chậm trên các trình duyệt cũ hoặc khi có nhiều component phức tạp.

Hơn nữa, việc quản lý trạng thái (state) trong các ứng dụng React lớn có thể trở nên phức tạp và khó kiểm soát. Dù có Redux để hỗ trợ quản lý trạng thái, nhưng điều này đòi hỏi sự hiểu biết và kỹ năng sử dụng một cách hiệu quả.

Ngoài ra, SEO (Search Engine Optimization) trong các ứng dụng React cũng đặt ra thách thức. Mặc dù có cải tiến, trình duyệt không thể hiểu các dữ liệu được tạo ra bằng JavaScript, điều này có thể ảnh hưởng đến khả năng tìm kiếm của ứng dụng trên các công cụ tìm kiếm.

### 3.2.2.NodeJS

#### Giới thiệu về NodeJS

NodeJS là một nền tảng được xây dựng dựa trên thời gian chạy JavaScript của Chrome để dễ dàng xây dựng các ứng dụng mạng nhanh và có thể mở rộng. NodeJS sử dụng kiến trúc hướng sự kiện event-driver, không chặn (non-blocking) làm cho nó nhẹ nhàng hơn, hoàn hảo cho các ứng dụng real-time sử dụng nhiều dữ liệu chạy trên các thiết bị phân tán [2].

NodeJS là môi trường chạy mã JavaScript phía máy chủ.Nó được xây dựng dựa trên JavaScript engine V8 của Google Chrome, cho phép chạy mã JavaScript nhanh và hiệu quả trên máy chủ.

Với NodeJS bạn thể xây dựng ứng dụng web và back-end bằng JavaScript, một ngôn ngữ phổ biến và linh hoạt. NodeJS cho phép bạn xử lý các ứng dụng khách yêu cầu và tạo API, kết nối với cơ sở dữ liệu, thao tác với các tệp và thực hiện các tác vụ máy chủ khác một cách đơn giản và hiệu quả.

NodeJS cũng chú trọng đến việc xử lý không đồng bộ (bất đồng bộ), cho phép ứng dụng xử lý nhiều yêu cầu cùng một lúc và không bị chặn bởi các tác vụ đang chờ đợi. Điều này làm cho NodeJS trở thành một lựa chọn phù hợp cho việc xây dựng ứng dụng chạy thời gian thực và có khả năng xử lý tải cao.

NodeJS cũng đi kèm với mô-đun thư viện phong phú, cho phép lập trình viên sử dụng các chức năng có sẵn và không cần viết lại từ đầu. Nó còn có một cộng đồng rộng lớn và hỗ trợ mạnh mẽ từ cộng đồng lập trình viên, chắc chắn rằng bạn có thể tìm thấy các tài liệu hướng dẫn và các mô-đun hữu ích để phát triển ứng dụng NodeJS của riêng bạn.

#### 3.2.2.1.Ưu điểm của NodeJS

NodeJS có nhiều ưu điểm khi sử dụng trong việc phát triển ứng dụng web và back-end. Dưới đây là một số điểm ưu tiên của NodeJS:

**+** **Hiệu suất cao:** NodeJS được xây dựng trên JavaScript engine V8, đảm bảo hiệu suất cao với việc xử lý mã JavaScript nhanh chóng. Nó cũng hỗ trợ xử lý không đồng bộ, cho phép xử lý nhiều yêu cầu cùng một lúc mà không gây gián đoạn.

**+** **Xử lý không đồng bộ:** NodeJS được xây dựng với kiến trúc không đồng bộ, cho phép xử lý nhiều yêu cầu cùng một lúc mà không bị chặn bởi các nhiệm vụ chờ đợi. Điều này làm cho ứng dụng trở nên rất nhạy bén và có thể đáp ứng tải cao.

**+ Sử dụng cùng một ngôn ngữ:** Với NodeJS, bạn có thể sử dụng cùng một ngôn ngữ JavaScript cho phía máy chủ và phía khách hàng. Điều này giúp giảm múc độ phức tạp trong quá trình phát triển và duy trì ứng dụng.

**+ Hệ sinh thái mô-đun phong phú:** NodeJS có một mô-đun hệ sinh thái phông phú, với hàng sát mô-đun có sẵn để tái sử dụng. Bạn có thể tìm thấy các mô-đun hữu ích trong cộng đồng NodeJS và sử dụng chúng để giảm thời gian phát triển và tăng tính linh hoạt của ứng dụng.

**+ Cộng đồng lập trình thành viên mạnh mẽ:** NodeJS có một cộng đồng lập trình thành viên lớn và nổi. Điều này đảm bảo rằng bạn co thể tìm thấy các tài nguyên, hướng dẫn và hổ trợ từ những người khác trong cộng đồng để giải quyết vấn đề và nâng cao kỹ năng lập trình của mình.

**+ Thích hợp cho ứng dụng thời gian thực:** Với việc hỗ trợ xử lý không đồng bộ và hiệu suất cao. NodeJS là một lựa chọn phù hợp cho việc xây dựng ứng dụng thời gian thực như ứng dụng trò chuyện trực tuyến,máy chủ trò chơi và ứng dụng theo thời gian thực khác.

#### 3.2.2.2.Nhược điểm của NodeJS

Mặc dù NodeJS có nhiều ưu điểm, nhưng cũng có một số nhược điểm bạn cần lưu ý:

**+ Đơn luồng:** NodeJS dựa trên mô hình đơn luồng, điều này có nghĩa là có một luồng xử lý trong Node.js. Điều này có thể gây ra vấn đề khi có các tác vụ tốn nhiều thời gian xử lý, làm cho ứng dụng hoạt động chậm.

**+ Khả năng xử lý CPU hạn chế:** NodeJS phù hợp cho việc xử lý các tác vụ I/O (ví dụ: yêu cầu mạng, truy vấn cơ sở dữ liệu), nhưng không phải là lựa chọn tốt cho các tác vụ tính toán mất nhiều thời gian. Do đó, nếu ứng dụng của bạn có yêu cầu xử lý CPU nặng, Node.js có thể không phải là lựa chọn tốt.

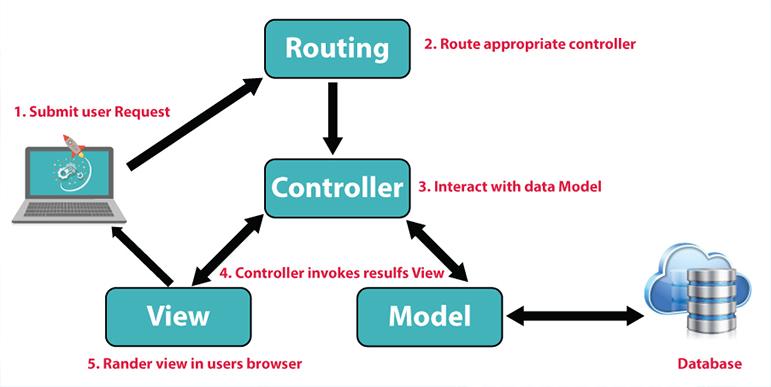
**+ Thư viện thứ ba không ổn định:** Mặc dù NodeJS có một hệ sinh thái module phong phú, nhung không phải tất cả các module bên thứ ba đều đảm bảo ổn định và chất lượng. Có thể gặp phải các vấn đề như sự không tương thích, lỗi hoặc sự thiếu hỗ trợ từ các module bên thứ ba.

**+ Quản lý phiên bản module:** Khi sử dụng NodeJS, việc quản lý phiên bản module có thể trở nên phức tạp. Vì NodeJS sử dụng mô hình xử lý module phiên bản, việc kiểm soát và đồng bộ các phiên bản module có thể trở nên rắc rối, đặc biệt khi có nhiều module sử dụng các phiên bản khác nhiêu.

**+ Scalability (tính mở rộng):** Scalability là một thách thức khi sử dụng NodeJS trong một môi trường có tải cao hoặc cần xây dựng ứng dụng lớn. Điều này đòi hỏi việc kiến trúc ứng dụng và các giải pháp phân tán được quản lý cẩn thận để đảm bảo tính mở rộng và hiệu xuất ổn định.

### 3.2.3.Mô hình

#### 3.2.3.1.Giới thiệu về mô hình MVC



*Hình 3. 13.Mô hình MVC*

MVC là viết tắt của cụm từ "Model-View-Controller," là một mô hình kiến trúc phần mềm được sử dụng rộng rãi trong lĩnh vực công nghệ phần mềm. Mô hình này được thiết kế để quản lý và xây dựng giao diện người dùng trên các ứng dụng máy tính. MVC phân chia ứng dụng thành ba thành phần chính, và mỗi thành phần có vai trò và nhiệm vụ riêng biệt và độc lập với nhau.

Trong mô hình MVC, có ba thành phần không thể thiếu:

**+ Model:** Đây là thành phần chịu trách nhiệm lưu trữ toàn bộ dữ liệu của ứng dụng và đóng vai trò là giao diện giữa hai thành phần khác là View và Controller. Model thường đại diện cho dữ liệu sử dụng bởi ứng dụng, có thể là cơ sở dữ liệu, các file dữ liệu như XML, hoặc thậm chí là các đối tượng đơn giản như biểu tượng hoặc nhân vật trong một trò chơi.

**+ View:** Đây là thành phần giao diện dành cho người sử dụng, thể hiện thông tin và tương tác giữa ứng dụng và người dùng. View bao gồm các thành phần như cửa sổ, nút bấm, văn bản, và hiển thị dữ liệu để người dùng có thể nhìn thấy và tương tác với chúng.

+ **Controller:** Thành phần này có nhiệm vụ xử lý các yêu cầu được người dùng gửi qua View. Controller bao gồm cả Model và View, và nó nhận các dữ liệu đầu vào từ người dùng và thực hiện các cập nhật và xử lý tương ứng. Controller là bộ não của ứng dụng, điều hướng và quản lý luồng dữ liệu giữa Model và View.

#### 3.2.3.2.Ưu điểm của mô hình MVC

Mô hình MVC (Model-View-Controller) mang đến nhiều ưu điểm quan trọng trong phát triển ứng dụng. Trước hết, mô hình này giúp tiết kiệm băng thông bởi vì không sử dụng viewstate, từ đó giúp giảm tải cho server và cải thiện tính ổn định của website. Khả năng này không chỉ giúp giảm khả năng gặp sự cố kết nối mạng mà còn cải thiện trải nghiệm người dùng.

Việc kiểm tra ứng dụng cũng trở nên dễ dàng hơn trong mô hình MVC. Nhờ sự phân chia rõ ràng giữa các thành phần Model, Controller và View, việc kiểm tra lỗi phần mềm trước khi ứng dụng được bàn giao cho người dùng trở nên đơn giản và hiệu quả hơn. Điều này giúp đảm bảo rằng ứng dụng làm việc ổn định và cung cấp chất lượng tốt.

Mô hình MVC tách biệt các thành phần trong ứng dụng, bao gồm Model (mô hình dữ liệu), Controller (bộ điều khiển), và View (giao diện). Sự tách biệt này giúp quản lý mã nguồn dễ dàng hơn, làm cho mã trở nên sạch sẽ và dễ đọc. Điều này giúp tôi phát triển làm việc hiệu quả hơn và dễ dàng theo dõi và duy trì mã nguồn.

Controller, một phần quan trọng của mô hình MVC, được tối ưu hóa cho nhiều nền tảng ngôn ngữ khác nhau, từ đó tạo ra tính linh hoạt cho ứng dụng. Nó giúp ứng dụng hoạt động tốt trên nhiều môi trường và thiết bị khác nhau.

Duy trì ứng dụng trong mô hình MVC trở nên dễ dàng hơn do sự tách biệt giữa các thành phần. Bạn có thể chỉnh sửa hoặc nâng cấp một phần mà không ảnh hưởng đến các phần khác.

#### 3.2.3.3.Nhược điểm của mô hình MVC

Mô hình MVC (Model-View-Controller) là một kiến trúc phát triển phần mềm mạnh mẽ, nhưng cũng đi kèm với một số nhược điểm cần được xem xét. Ban đầu, việc hiểu và triển khai mô hình này có thể phức tạp, đặc biệt đối với người mới bắt đầu. Sự tách biệt chặt chẽ giữa Model (mô hình), View (giao diện) và Controller (điều khiển) có thể dẫn đến hiệu suất kém trong trường hợp các ứng dụng yêu cầu nhiều tương tác giữa các thành phần này.

Một trong những thách thức khác là việc quản lý mã trở nên phức tạp, đặc biệt là trong các dự án lớn. Cấu trúc MVC có thể tạo ra nhiều lớp và tầng lớp mã, làm cho mã trở nên khó quản lý và khó hiểu. Thời gian phát triển dự án cũng có thể kéo dài hơn so với các mô hình khác, đặc biệt khi yêu cầu của dự án thay đổi thường xuyên.

Đối với các dự án nhỏ và đơn giản, triển khai mô hình MVC có thể cảm thấy quá phức tạp và không cần thiết, làm cho mã trở nên quá nặng nề và khó quản lý. Cuối cùng, việc sử dụng mô hình MVC đòi hỏi kiến thức và kỹ năng cao cho phát triển và bảo trì hiệu quả, và có thể không phù hợp cho những dự án với đội ngũ phát triển không có kinh nghiệm trong việc làm việc với kiến trúc này.

### 3.2.4.Cơ sở dữ liệu MySQL

#### 3.2.4.1.Giới thiệu Mysql

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến nhất hiện nay. Được phát triển vào giữa những năm 1990 bởi công ty Thụy Điển MySQL AB, MySQL nhanh chóng trở thành lựa chọn hàng đầu cho các ứng dụng web, nhờ vào tính linh hoạt, hiệu suất cao và khả năng mở rộng. Tên gọi "MySQL" xuất phát từ sự kết hợp của "My" - tên con gái của một trong những người sáng lập, và "SQL" - ngôn ngữ truy vấn có cấu trúc (Structured Query Language), là ngôn ngữ chuẩn được sử dụng để quản lý và thao tác với cơ sở dữ liệu.

MySQL được thiết kế với mục tiêu cung cấp một hệ quản trị cơ sở dữ liệu dễ sử dụng nhưng mạnh mẽ, phù hợp cho các dự án từ nhỏ đến lớn. Với khả năng hỗ trợ nhiều hệ điều hành như Linux, Windows và macOS, MySQL đã trở thành nền tảng dữ liệu của nhiều ứng dụng web phổ biến như WordPress, Joomla, và các hệ thống thương mại điện tử lớn. Sự phổ biến của MySQL còn được củng cố bởi sự hỗ trợ của một cộng đồng phát triển mạnh mẽ và sự tài trợ từ các công ty công nghệ lớn, bao gồm Oracle Corporation, công ty hiện đang sở hữu và phát triển MySQL.

#### 3.2.4.2.Ưu điểm của Mysql

MySQL nổi bật như một trong những hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến nhất thế giới nhờ vào nhiều ưu điểm vượt trội. Trước hết, MySQL được đánh giá cao về hiệu suất và tốc độ xử lý, đặc biệt là trong các ứng dụng web. Với khả năng xử lý hàng triệu truy vấn mỗi ngày, MySQL trở thành nền tảng lý tưởng cho các trang web có lượng truy cập lớn, giúp duy trì hiệu quả và sự ổn định ngay cả dưới áp lực cao. Là phần mềm mã nguồn mở, MySQL cung cấp sự linh hoạt tối đa cho người dùng và các nhà phát triển, cho phép họ tự do tùy chỉnh và mở rộng theo nhu cầu cụ thể của dự án mà không gặp rào cản về chi phí bản quyền.

Một trong những lý do chính khiến MySQL được ưa chuộng là tính linh hoạt cao trong việc hỗ trợ nhiều hệ điều hành và ngôn ngữ lập trình khác nhau, bao gồm PHP, Java, Python, và .NET. Điều này làm cho MySQL dễ dàng tích hợp vào các môi trường phát triển đa dạng, đáp ứng tốt cho cả các dự án nhỏ và lớn. Về mặt bảo mật, MySQL không hề thua kém bất kỳ hệ quản trị cơ sở dữ liệu nào khác. Nó cung cấp các tính năng bảo mật mạnh mẽ như xác thực người dùng, quản lý quyền truy cập chi tiết, và mã hóa dữ liệu, đảm bảo rằng thông tin của người dùng luôn được bảo vệ an toàn trước các mối đe dọa.

Khả năng mở rộng của MySQL cũng là một điểm mạnh đáng chú ý. MySQL có thể dễ dàng xử lý các cơ sở dữ liệu từ nhỏ đến rất lớn nhờ vào các tính năng như replication (sao chép dữ liệu) và clustering (cụm máy chủ), giúp hệ thống luôn sẵn sàng đáp ứng nhu cầu ngày càng tăng của các tổ chức lớn. Hơn nữa, sự hỗ trợ từ một cộng đồng phát triển toàn cầu mạnh mẽ là một tài sản quý giá của MySQL. Với hàng ngàn nhà phát triển và người dùng đóng góp vào việc cải thiện và hỗ trợ, người dùng MySQL luôn có sẵn nhiều tài liệu, công cụ, và giải pháp để giúp họ giải quyết mọi thách thức mà họ có thể gặp phải.

Những ưu điểm này giúp MySQL không chỉ là một lựa chọn phổ biến cho các nhà phát triển web và các doanh nghiệp mà còn là một nền tảng tin cậy để xây dựng các hệ thống dữ liệu mạnh mẽ, ổn định và linh hoạt.

#### 3.2.4.3.Nhược điểm của Mysql

Mặc dù MySQL nổi bật với nhiều ưu điểm và sự phổ biến rộng rãi, nhưng nó cũng tồn tại một số nhược điểm cần lưu ý khi lựa chọn hệ quản trị cơ sở dữ liệu cho các dự án phức tạp. Trước hết, MySQL có những giới hạn nhất định về tính năng so với các hệ quản trị cơ sở dữ liệu tiên tiến khác như PostgreSQL hay Oracle. MySQL thiếu một số tính năng cao cấp như hỗ trợ đầy đủ cho Common Table Expressions (CTEs), khả năng xử lý các truy vấn phức tạp và các giao dịch có độ phức tạp cao. Điều này có thể làm giảm hiệu quả khi cần thao tác với các dữ liệu lớn hoặc thực hiện các phép tính toán và phân tích phức tạp.

Một vấn đề khác là hiệu suất của MySQL có thể giảm đáng kể khi xử lý các cơ sở dữ liệu rất lớn hoặc trong các ứng dụng yêu cầu tốc độ truy cập rất cao. Mặc dù MySQL có khả năng mở rộng, nhưng việc tối ưu hóa hiệu suất cho các ứng dụng có lượng dữ liệu khổng lồ hoặc yêu cầu truy vấn thời gian thực phức tạp có thể trở thành thách thức lớn, đòi hỏi sự hiểu biết sâu rộng về cơ sở dữ liệu và các phương pháp tối ưu hóa cụ thể của MySQL.

Bên cạnh đó, mặc dù MySQL là một hệ quản trị cơ sở dữ liệu tuân thủ chuẩn SQL, nó không hoàn toàn tương thích với tất cả các tiêu chuẩn SQL, và có thể dẫn đến các vấn đề tương thích khi di chuyển dữ liệu hoặc tích hợp với các hệ quản trị cơ sở dữ liệu khác. Điều này đặc biệt quan trọng khi phải làm việc trong môi trường có sự tương tác đa dạng với nhiều loại cơ sở dữ liệu khác nhau.

Việc phát triển và cập nhật của MySQL, dù có một cộng đồng phát triển mạnh mẽ, đôi khi cũng diễn ra chậm hơn so với một số hệ quản trị cơ sở dữ liệu thương mại khác. Điều này có thể ảnh hưởng đến khả năng tiếp cận các tính năng mới hoặc các cải tiến về hiệu suất kịp thời, làm cho MySQL đôi khi trở thành một lựa chọn kém hấp dẫn hơn đối với các dự án cần các giải pháp hiện đại và nhanh chóng cập nhật.

### 3.2.5.Amazon Rekognition

#### 3.2.5.1.Giới thiệu Amazon Rekognition

Amazon Rekognition là một dịch vụ mạnh mẽ do Amazon Web Services (AWS) cung cấp, chuyên cung cấp các tính năng nhận dạng hình ảnh và video bằng trí tuệ nhân tạo (AI). Dịch vụ này cho phép các nhà phát triển tích hợp dễ dàng các khả năng nhận dạng khuôn mặt, phát hiện vật thể và hành vi, phân tích video và ảnh động, cũng như đánh giá nội dung không phù hợp.

Amazon Rekognition cung cấp một loạt các tính năng nhận dạng khuôn mặt và phân tích hành vi, từ việc nhận dạng các đặc điểm khuôn mặt cụ thể đến phân tích hành vi trong video. Nó cũng hỗ trợ các tính năng phát hiện vật thể và đánh giá nội dung, giúp các tổ chức xây dựng và triển khai các ứng dụng thông minh và an toàn.

Đặc biệt, Amazon Rekognition tích hợp sâu vào hạ tầng AWS, cho phép tích hợp mượt mà với các dịch vụ khác như AWS Lambda, Amazon S3 và Amazon Kinesis Video Streams. Nhờ vào việc cung cấp API đơn giản và SDK hỗ trợ nhiều ngôn ngữ lập trình, Amazon Rekognition giúp các nhà phát triển nhanh chóng phát triển và triển khai các ứng dụng AI phức tạp mà không cần có sự chuyên sâu về machine learning.

Với những khả năng nổi bật và tích hợp linh hoạt, Amazon Rekognition là một lựa chọn lý tưởng cho các tổ chức muốn tận dụng sức mạnh của trí tuệ nhân tạo để tăng cường an ninh, quản lý dữ liệu và cải thiện trải nghiệm người dùng trong các ứng dụng của họ.

#### 3.2.5.2.Ưu điểm của Amazon Rekognition

Amazon Rekognition là một dịch vụ trí tuệ nhân tạo (AI) do Amazon Web Services (AWS) cung cấp, chuyên cung cấp các tính năng mạnh mẽ cho việc nhận dạng hình ảnh và video. Với khả năng phát hiện khuôn mặt, vật thể và hành vi, Amazon Rekognition đem lại cho các tổ chức và nhà phát triển những lợi ích đáng kể trong việc tự động hóa các quy trình phân tích dữ liệu và nâng cao trải nghiệm người dùng.

Một trong những ưu điểm nổi bật của Amazon Rekognition là tính linh hoạt và dễ sử dụng. Dịch vụ này cung cấp API đơn giản và SDK hỗ trợ nhiều ngôn ngữ lập trình, giúp các nhà phát triển có thể tích hợp các tính năng AI vào ứng dụng của mình một cách nhanh chóng và hiệu quả. Điều này không chỉ giảm thiểu thời gian phát triển mà còn giúp tăng tốc độ triển khai các giải pháp trí tuệ nhân tạo.

Amazon Rekognition cũng được đánh giá cao về hiệu suất và khả năng mở rộng. Với hạ tầng mạnh mẽ của AWS, dịch vụ này có thể xử lý các tình huống phức tạp và đồng thời đáp ứng được nhu cầu của các ứng dụng có lượng người dùng lớn một cách mượt mà. Điều này đặc biệt quan trọng đối với các ứng dụng yêu cầu độ chính xác cao và thời gian phản hồi nhanh.

Thêm vào đó, Amazon Rekognition còn cam kết đảm bảo bảo mật dữ liệu và tin cậy của người dùng. Dịch vụ này tuân thủ các tiêu chuẩn bảo mật nghiêm ngặt của ngành công nghiệp, đảm bảo rằng thông tin cá nhân và dữ liệu quan trọng được bảo vệ an toàn.

#### 3.2.5.3.Nhược điểm của Amazon Rekognition

Tuy nhiên, như mọi công nghệ và dịch vụ, Amazon Rekognition cũng có một số nhược điểm cần được xem xét trước khi triển khai:

Một trong những vấn đề đối mặt của Amazon Rekognition là độ chính xác và độ tin cậy của các tính năng nhận dạng. Dù đã được cải thiện đáng kể, các thuật toán AI vẫn có thể gặp phải các trường hợp nhận dạng sai lầm hoặc không chính xác, đặc biệt khi đối mặt với điều kiện ánh sáng yếu, góc chụp khó hoặc các tình huống phức tạp.

Bên cạnh đó, chi phí sử dụng Amazon Rekognition cũng là một yếu tố cần xem xét. Dịch vụ này tính phí theo số lượng yêu cầu xử lý và lưu trữ dữ liệu, do đó việc tính toán chi phí và tối ưu hóa tài nguyên là điều cần thiết để tránh gánh nặng tài chính cho các tổ chức và doanh nghiệp.

Thêm vào đó, việc bảo mật và quản lý dữ liệu cũng là một vấn đề quan trọng. Mặc dù Amazon Rekognition cam kết đảm bảo bảo mật thông tin, việc lưu trữ và xử lý dữ liệu nhạy cảm vẫn đòi hỏi các biện pháp bảo mật chặt chẽ để đảm bảo tuân thủ các quy định pháp luật và sự tin tưởng của người dùng.

### 3.2.6.Các dịch vụ liên quan AWS

#### **3.2.6.1.Dịch Vụ Lưu Trữ**

**Amazon S3 (Simple Storage Service)**:

* **Giới thiệu**: Amazon S3 là dịch vụ lưu trữ đối tượng mạnh mẽ, cho phép lưu trữ và truy xuất dữ liệu từ bất kỳ đâu trên web. Nó cung cấp lưu trữ dữ liệu có thể mở rộng với khả năng bảo mật cao.
* **Vai trò**: Lưu trữ các hình ảnh và video đầu vào, cũng như lưu trữ các kết quả phân tích và nhận diện từ Amazon Rekognition. Nó cũng hỗ trợ khả năng sao lưu và phục hồi dữ liệu.

#### **3.2.6.2.Xử Lý Dữ Liệu**

**AWS Lambda**:

* **Giới thiệu**: AWS Lambda là dịch vụ điện toán không máy chủ, cho phép bạn chạy mã mà không cần quản lý máy chủ. Mã được thực thi dựa trên sự kiện kích hoạt và tự động mở rộng quy mô.
* **Vai trò**: Xử lý các sự kiện từ Amazon S3 khi các hình ảnh hoặc video được tải lên, và thực hiện các phân tích hoặc thao tác xử lý dựa trên các kết quả từ Amazon Rekognition.

#### **3.2.6.3.Dịch Vụ Dữ Liệu Thời Gian Thực**

**Amazon Kinesis Video Streams**:

* **Giới thiệu**: Amazon Kinesis Video Streams cho phép bạn dễ dàng thu thập, lưu trữ và phân tích dữ liệu video từ các nguồn khác nhau.
* **Vai trò**: Cung cấp khả năng truyền video trực tiếp đến Amazon Rekognition cho các phân tích thời gian thực, chẳng hạn như phân tích video từ camera an ninh.

#### **3.2.6.4.Phân Tích và Xử Lý Dữ Liệu**

**Amazon SageMaker**:

* **Giới thiệu**: Amazon SageMaker là dịch vụ máy học được quản lý, giúp bạn xây dựng, đào tạo và triển khai các mô hình học máy với khả năng mở rộng cao.
* **Vai trò**: Cung cấp khả năng xây dựng và triển khai các mô hình học máy tùy chỉnh để bổ sung các khả năng nhận diện hình ảnh của Amazon Rekognition.

#### **3.2.6.5.Quản Lý và Phân Tích Dữ Liệu**

**Amazon DynamoDB**:

* **Giới thiệu**: Amazon DynamoDB là dịch vụ cơ sở dữ liệu NoSQL có khả năng mở rộng tự động, cung cấp hiệu suất thấp latencies và độ tin cậy cao.
* **Vai trò**: Lưu trữ và quản lý dữ liệu phân tích và các kết quả nhận diện từ Amazon Rekognition, đồng thời hỗ trợ các ứng dụng cần truy cập dữ liệu nhanh chóng và liên tục.

#### **3.2.6.6.Bảo Mật và Quản Lý Quyền Truy Cập**

**AWS Identity and Access Management (IAM)**:

* **Giới thiệu**: AWS IAM cho phép bạn kiểm soát quyền truy cập đến các dịch vụ và tài nguyên AWS một cách an toàn.
* **Vai trò**: Quản lý quyền truy cập cho các dịch vụ khác nhau trong hệ thống, đảm bảo chỉ những người hoặc dịch vụ có quyền mới có thể thực hiện các thao tác nhất định.

#### **3.2.6.7.Giao Diện Người Dùng**

**Amazon API Gateway**:

* **Giới thiệu**: Amazon API Gateway là dịch vụ giúp dễ dàng tạo, triển khai và quản lý các API.

**Vai trò**: Cung cấp một giao diện API cho các ứng dụng khách để tương tác với các dịch vụ AWS như Amazon Rekognition và Amazon Lambda

**Bảng So Sánh Chi Phí và Thời Gian Sử Dụng Dịch Vụ AWS Rekognition Chi Tiết:**

|  |  |  |
| --- | --- | --- |
| Tiêu chí | Chụp ảnh từng người | Chụp ảnh nhóm (10 người/lần) |
| Số lượng sinh viên | 8,000 sinh viên | 8,000 sinh viên |
| Số ảnh cần xử lý | 8,000 ảnh | 800 ảnh |
| Chi phí xử lý ảnh |  |  |
| - Gói miễn phí (5,000 ảnh/tháng) | Miễn phí | Miễn phí |
| - 3,000 ảnh còn lại |  |  |
| - 1,000 ảnh tiếp theo | 1 USD | 0.8 USD |
| - 2,000 ảnh tiếp theo | 1.6 USD | - |
| Tổng chi phí | 2.6 USD | 0.8 USD |
| Thời gian xử lý mỗi ảnh | < 1 giây | < 1 giây |
| Tổng thời gian xử lý | < 8,000 giây (≈ 2.22 giờ) | < 800 giây (≈ 13.33 phút) |
| Ưu điểm | Đơn giản, dễ triển khai | Tiết kiệm chi phí, nhanh chóng |
| Nhược điểm | Chi phí cao hơn, thời gian lâu hơn | Cần tổ chức, sắp xếp sinh viên |

### 3.2.7.Chi tiết chi phí

#### 1. Chụp ảnh từng người

* **Gói miễn phí AWS Rekognition**: Xử lý miễn phí 5,000 ảnh đầu tiên mỗi tháng.
* **Số ảnh cần xử lý**: 8,000 ảnh.
* **Chi phí cho 8,000 ảnh**:
  + 5,000 ảnh đầu tiên: Miễn phí.
  + 3,000 ảnh tiếp theo:
    - 1,000 ảnh đầu tiên: 1 USD.
    - 2,000 ảnh tiếp theo: 2,000 x 0.80 USD = 1.6 USD.
  + **Tổng chi phí**: 1 USD + 1.6 USD = 2.6 USD.

#### 2. Chụp ảnh nhóm (ví dụ 10 người/lần)

* **Gói miễn phí AWS Rekognition**: Xử lý miễn phí 5,000 ảnh đầu tiên mỗi tháng.
* **Số lượng nhóm ảnh**: 8,000/10=800 nhóm.
* **Chi phí cho 800 ảnh**:
  + 800 ảnh đầu tiên: Miễn phí.
  + **Tổng chi phí**: 0 USD.

#### **3** .Quy trinh hoạt động tổng quát

1. **Đăng ký khuôn mặt sinh viên**:
   * Hình ảnh của sinh viên được tải lên S3: Employee Pics.
   * Registration Lambda lấy hình ảnh từ S3 và sử dụng Amazon Rekognition để phân tích và lưu trữ dữ liệu khuôn mặt trong DynamoDB.
2. **Nhận diện khuôn mặt**:
   * Sinh viên hoặc sinh viên tải lên hình ảnh của mình thông qua ứng dụng React Frontend.
   * Hình ảnh được gửi tới API Gateway, từ đó truyền tiếp tới Auth Lambda.
   * Auth Lambda sẽ sử dụng Amazon Rekognition để so sánh hình ảnh này với dữ liệu đã lưu trữ trong DynamoDB. Nếu khớp, hệ thống sẽ xác thực danh tính của người đó.

### 3.2.8.Kết luận và Đề xuất

#### 3.2.8.1.Kết luận

* **Chi phí**:
  + Chụp ảnh từng người: 2.6 USD.
  + Chụp ảnh nhóm: 0.8 USD (dưới gói miễn phí).
* **Thời gian**:
  + Chụp ảnh từng người: ≈ 2.22 giờ.
  + Chụp ảnh nhóm: ≈ 13.33 phút.

#### 3.2.8.2.Đề xuất

* Áp dụng phương pháp chụp ảnh nhóm 10 người một lần để tối ưu chi phí và thời gian.
* Lập kế hoạch và tổ chức việc chụp ảnh nhóm để đảm bảo hiệu quả và chính xác trong quá trình nhận dạng khuôn mặt.

#### 3.2.8.3. Chức năng của hệ thống

**Đăng ký sinh viên trước kỳ thi:**

* + Mỗi sinh viên sẽ chụp ảnh khuôn mặt và lưu trữ trong cơ sở dữ liệu của hệ thống.
  + Ảnh được xử lý và lưu trữ bởi dịch vụ AWS Rekognition.

**Xác thực danh tính sinh viên trong kỳ thi:**

* + Sinh viên khi vào phòng thi sẽ được chụp ảnh ngay tại chỗ.
  + Ảnh chụp sẽ được gửi đến AWS Rekognition để so sánh với ảnh đã lưu trong cơ sở dữ liệu.
  + Kết quả nhận dạng sẽ trả về cho giám thị để xác nhận danh tính sinh viên.

**Giám sát trong suốt quá trình thi:**

* + Ứng dụng có thể được sử dụng để chụp ảnh

#### 3.2.8.4. Lợi ích khi sử dụng AWS Rekognition

**Tăng cường hiệu quả xác thực danh tính:**

* + Xác thực nhanh chóng và chính xác, giảm thời gian so với việc kiểm tra thủ công.
  + Giảm sai sót do con người gây ra trong quá trình xác thực.

**Giảm thiểu gian lận:**

* + Phát hiện và ngăn chặn việc sử dụng danh tính giả hoặc việc thay người thi.

**Tối ưu hóa chi phí và thời gian:**

* + Sử dụng AWS Rekognition giúp tiết kiệm chi phí so với các giải pháp tự xây dựng hoặc thuê ngoài.
  + Thời gian xử lý nhanh chóng, giúp hệ thống hoạt động hiệu quả ngay cả khi có số lượng sinh viên lớn.

### 3.2.9.Cài đặt chương trình

#### 3.2.9.1.Chức năng nhận diện



#### 3.2.9.2. Các API của backend



### 3.2.10.Kiểm thử

#### 3.2.10.1.Kiểm thử chức năng đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **TC** | **Các bước thực hiện** | **Kết quả dự kiến** | **Kết quả đạt được** |
| TC1 | 1. Nhập sai tài khoản  2. Nhập mật khẩu  3. Bấm đăng nhập | Thông báo tài khoản không tồn tại. | Đạt |
| TC2 | 1. Nhập tài khoản đúng  2. Nhập mật khẩu sai  3. Bấm đăng nhập | Mật khẩu không chính xác. | Đạt |
| TC3 | 1. Bỏ trống tài khoản  2. Bỏ trống mật khẩu  3. Bấm đăng nhập | Tại view hiển thị, yêu cầu người dùng nhập đầy đủ tài khoản, mật khẩu. | Đạt |
| TC4 | 1. Nhập sai định dạng tài khoản  2. Nhập mật khẩu đúng | 1. Thông báo mật khẩu sai.  2. Dữ liệu được mã hóa đạt chuẩn về vấn đề bảo mật. | Đạt |
| TC5 | 1. Nhập tài khoản đúng  2. Nhập mật khẩu dúng  3. Bấm đăng nhập | Chuyển hướng đến trang chủ . | Đạt |

*Bảng 3. 1. Kiểm thử chức năng đăng nhập*

#### 3.2.10.2.Kiểm thử chức năng đăng ký

|  |  |  |
| --- | --- | --- |
| **Các bước thực hiện** | **Kết quả dự kiến** | **Kết quả đạt được** |
| 1. Nhập sai định dạng email.  2. Nhập sai định dạng mật khẩu.  2. Nhập lại mật khẩu.  3. Nhập đúng định dạng tên đăng nhập.  4. Bấm đăng ký. | Thông báo nhập sai định dạng email, nhập lại mật khẩu sai. | Đạt |
| 1. Nhập đúng email.  2. Nhập đúng mật khẩu.  3. Nhập lại mật khẩu sai.  4. Nhập đúng định dạng tên đăng nhập.  5. Bấm đăng ký. | Thông báo nhập mật khẩu không khớp. | Đạt |
| 1. Nhập đúng email.  2. Nhập mật khẩu đúng.  3. Nhập lại mật khẩu đúng.  4. Nhập sai định dạng tên đăng nhập.  5. Bấm đăng ký. | Thông báo nhập sai định dạng tên đăng ký. | Đạt |
| 1. Bỏ trống email.  2. Bỏ trống mật khẩu.  3. Bỏ trống xác thực lại mật khẩu.  4. Bỏ trống tên đăng nhập.  5. Bấm đăng ký . | Thông báo không được bỏ trống các trường. | Đạt |
| 1. Nhập tên đăng nhập đúng định dạng.  2. Nhập đúng định dạng mật khẩu.  3. Nhập đúng xác thực mật khẩu.  4. Nhập đúng email nhưng đã tồn tại.  5. Bấm đăng ký. | Thông báo email đã tồn tại. | Đạt |
| 1. Nhập tên đăng nhập đúng định dạng.  2. Nhập đúng định dạng mật khẩu.  2. Nhập khớp mật khẩu lại.  3. Nhập đúng email.  3. Bấm đăng ký. | Đăng ký thành công. | Đạt |

*Bảng 3. 2.Kiểm thử chức năng đăng ký*

#### 3.2.10.3.Kiểm thử một số chức năng khác

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Các bước thực hiện** | **Kết quả dự kiến** | **Kết quả đạt được** |
| |  | | --- | | **Thêm mới học viên** |  |  | | --- | |  | | 1. Admin đăng nhập vào hệ thống. 2. Truy cập danh sách học viên của môn học. 3. Thêm mới học viên với thông tin cần thiết. | |  | | --- | | Học viên mới được thêm vào danh sách học viên. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Loại bỏ học viên** |  |  | | --- | |  | | 1. Admin đăng nhập. 2. Truy cập danh sách học viên. 3. Chọn một học viên và bấm nút loại bỏ. | |  | | --- | | Học viên được xóa khỏi danh sách. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Tạo lịch thi mới cho môn học** |  |  | | --- | |  | | 1. Admin đăng nhập. 2. Truy cập phần quản lý lịch thi. 3. Tạo lịch thi với ngày, giờ, và địa điểm cụ thể cho môn học. | |  | | --- | | Lịch thi mới được tạo thành công và hiển thị trong hệ thống. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Cập nhật lịch thi hiện tại** |  |  | | --- | |  | | 1. Admin đăng nhập. 2. Truy cập phần quản lý lịch thi. 3. Chỉnh sửa lịch thi đã tạo trước đó và lưu thay đổi. | |  | | --- | | Lịch thi được cập nhật thành công với thông tin mới. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Điểm danh sinh viên bằng nhận dạng** |  |  | | --- | |  | | 1. Giám thị đăng nhập. 2. Truy cập chức năng điểm danh. 3. Dùng công nghệ nhận dạng khuôn mặt để điểm danh sinh viên trong lớp học. | |  | | --- | | Hệ thống tự động xác minh danh tính sinh viên và điểm danh thành công. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Xem danh sách sinh viên đã điểm danh** |  |  | | --- | |  | | 1. Giám thị đăng nhập. 2. Truy cập phần danh sách sinh viên đã điểm danh. 3. Xem danh sách các sinh viên đã được xác minh và điểm danh. | |  | | --- | | Hiển thị danh sách đầy đủ các sinh viên đã điểm danh thành công. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Sinh viên không thể điểm danh sai** |  |  | | --- | |  | | 1. Sinh viên không thuộc môn học đăng nhập. 2. Thử sử dụng tính năng điểm danh nhận dạng khuôn mặt. | |  | | --- | | Hệ thống từ chối điểm danh sinh viên không thuộc môn học. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Thông báo lỗi khi không nhận diện được** |  |  | | --- | |  | | 1. Giám thị sử dụng hệ thống điểm danh. 2. Sinh viên có vấn đề về hình ảnh (không rõ mặt hoặc sai ánh sáng). 3. Hệ thống cố gắng nhận diện. | |  | | --- | | Hệ thống thông báo lỗi nhận diện không thành công. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Xem lịch thi đã tạo** |  |  | | --- | |  | | 1. Sinh viên đăng nhập. 2. Truy cập vào phần xem lịch thi của môn học. 3. Xem thông tin lịch thi. | |  | | --- | | Sinh viên có thể xem chi tiết ngày, giờ, và địa điểm của lịch thi. |  |  | | --- | |  | | Đạt |
| |  | | --- | | **Tự động cập nhật danh sách điểm danh** |  |  | | --- | |  | | 1. Giám thị thực hiện điểm danh cho lớp học. 2. Kết thúc điểm danh. 3. Truy cập vào danh sách sinh viên đã điểm danh. | |  | | --- | | Danh sách sinh viên đã điểm danh tự động được cập nhật. |  |  | | --- | |  | | Đạt |

*Bảng 3. 3.Kiểm thử một số chức năng khác*

# CHƯƠNG 4. TỔNG KẾT VÀ ĐÁNH GIÁ

## 4.1.Kết quả đạt được

Dự án " Xây dựng ứng dụng web hỗ trợ giám thị xác nhận thí sinh dựa trên tính năng nhận dạng khuôn mặt " đã đạt được những kết quả ấn tượng và vượt ngoài mong đợi. Bằng việc tích hợp hiệu quả giữa Flask API, MySQL và AWS Rekognition, nhóm phát triển đã tạo ra một hệ thống giám sát thi cử tự động, đáng tin cậy và chính xác.

Ứng dụng này đã chứng minh được khả năng nhận dạng và xác thực khuôn mặt của thí sinh một cách nhanh chóng và chính xác, từ đó hỗ trợ giám thị trong việc quản lý và theo dõi quá trình thi cử một cách hiệu quả. AWS Rekognition đã cho thấy sức mạnh vượt trội trong việc xử lý và phân tích hình ảnh, giúp phát hiện các hành vi gian lận, như việc sử dụng người thay thế hoặc các thiết bị gian lận khác. Nhờ đó, tính toàn vẹn và công bằng của các kỳ thi được đảm bảo ở mức cao nhất.

Cơ sở dữ liệu MySQL được thiết kế một cách khoa học và tối ưu, giúp lưu trữ và truy xuất thông tin một cách nhanh chóng và hiệu quả. Các thông tin quan trọng về thí sinh, bao gồm dữ liệu khuôn mặt, thông tin cá nhân và kết quả nhận dạng, đều được quản lý và bảo mật chặt chẽ. Việc sử dụng Flask API làm nền tảng cho hệ thống đã mang lại sự linh hoạt và dễ dàng trong việc triển khai, mở rộng và bảo trì ứng dụng.

Ngoài ra, ứng dụng còn cung cấp giao diện người dùng thân thiện và dễ sử dụng, giúp giám thị có thể dễ dàng thao tác và quản lý thông tin. Các báo cáo và thống kê được tạo ra một cách chi tiết và rõ ràng, cung cấp cho giám thị và các nhà quản lý giáo dục cái nhìn tổng quan và chính xác về tình hình thi cử.

Tổng thể, dự án đã mang lại một giải pháp công nghệ tiên tiến, hiện đại và hiệu quả, góp phần nâng cao chất lượng và sự công bằng trong các kỳ thi. Ứng dụng không chỉ giúp giảm thiểu tối đa các hành vi gian lận mà còn tiết kiệm thời gian và công sức cho giám thị, tạo điều kiện thuận lợi cho việc tổ chức các kỳ thi trong môi trường giáo dục hiện đại.

## 4.2.Định hướng phát triển

Trong tương lai, dự án " Xây dựng ứng dụng web hỗ trợ giám thị xác nhận thí sinh dựa trên tính năng nhận dạng khuôn mặt" sẽ tiếp tục phát triển và mở rộng để đáp ứng nhu cầu ngày càng cao của các tổ chức giáo dục và cơ quan quản lý thi cử. Một trong những định hướng chính là nâng cao độ chính xác và tốc độ nhận dạng khuôn mặt thông qua việc cải thiện thuật toán và tối ưu hóa mô hình nhận dạng. Điều này sẽ được thực hiện bằng cách sử dụng các kỹ thuật học sâu và mạng nơ-ron tiên tiến, giúp hệ thống có thể nhận diện khuôn mặt nhanh hơn và chính xác hơn, ngay cả trong điều kiện ánh sáng yếu hoặc khi khuôn mặt bị che khuất một phần.

Để đảm bảo an toàn và bảo mật cho thông tin thí sinh, dự án sẽ tích hợp các biện pháp bảo mật nâng cao như mã hóa dữ liệu đầu cuối và xác thực đa yếu tố. Việc giám sát an ninh liên tục và tuân thủ các tiêu chuẩn bảo mật quốc tế cũng sẽ được đảm bảo, nhằm bảo vệ dữ liệu khỏi các mối đe dọa tiềm tàng.

Dự án cũng sẽ tập trung vào việc mở rộng quy mô và khả năng tích hợp với các hệ thống quản lý thi cử hiện có. Hệ thống sẽ được thiết kế để hỗ trợ quy mô lớn hơn, có khả năng xử lý hàng nghìn thí sinh trong mỗi kỳ thi. Bên cạnh đó, việc tích hợp với các hệ thống quản lý học sinh, hệ thống quản lý thi trắc nghiệm, và các cổng thông tin giáo dục sẽ được triển khai, tạo ra một giải pháp tổng thể và toàn diện, giúp các tổ chức giáo dục dễ dàng quản lý và theo dõi quá trình thi cử.

Phát triển ứng dụng di động cũng là một định hướng quan trọng, nhằm tăng tính tiện lợi và linh hoạt cho giám thị và thí sinh. Với phiên bản di động của ứng dụng, giám thị và thí sinh có thể sử dụng điện thoại thông minh hoặc máy tính bảng để truy cập vào hệ thống, giúp cải thiện trải nghiệm người dùng và dễ dàng quản lý thông tin.

Để đáp ứng nhu cầu của các tổ chức giáo dục trên toàn cầu, hệ thống sẽ được phát triển để hỗ trợ nhiều ngôn ngữ và điều chỉnh cho phù hợp với các quy định và yêu cầu địa phương. Điều này sẽ giúp mở rộng phạm vi ứng dụng và tăng cường khả năng cạnh tranh trên thị trường quốc tế.

Ngoài ra, dự án sẽ tận dụng sức mạnh của trí tuệ nhân tạo và phân tích dữ liệu lớn để cung cấp các báo cáo và phân tích chi tiết về quá trình thi cử. Điều này sẽ giúp các tổ chức giáo dục và cơ quan quản lý thi cử có cái nhìn sâu sắc và đưa ra các quyết định chính xác và kịp thời. Các báo cáo và phân tích sẽ được tạo ra một cách chi tiết và rõ ràng, cung cấp cho giám thị và các nhà quản lý giáo dục cái nhìn tổng quan và chính xác về tình hình thi cử.

Cuối cùng, dự án sẽ tìm kiếm cơ hội hợp tác với các tổ chức giáo dục, cơ quan quản lý thi cử, và cộng đồng nghiên cứu để thúc đẩy sự phát triển và cải tiến liên tục. Các chương trình đào tạo và hỗ trợ kỹ thuật cũng sẽ được triển khai để giúp người dùng tận dụng tối đa các tính năng của hệ thống. Với những định hướng phát triển này, dự án sẽ tiếp tục tiên phong trong việc ứng dụng công nghệ nhận dạng khuôn mặt vào giám sát thi cử, mang lại lợi ích thiết thực cho các tổ chức giáo dục và góp phần nâng cao chất lượng và công bằng trong các kỳ thi. Ứng dụng không chỉ giúp giảm thiểu tối đa các hành vi gian lận mà còn tiết kiệm thời gian và công sức cho giám thị, tạo điều kiện thuận lợi cho việc tổ chức các kỳ thi trong môi trường giáo dục hiện đại.

# TÀI LIỆU THAM KHẢO

1. <https://react.dev/>
2. <https://nodejs.org/en>
3. <https://aws.amazon.com/vi/rekognition/>
4. <https://techie.vn/giai-dap-tat-tan-tat-ve-cong-nghe-nhan-dang-khuon-mat/>
5. How to code in React.js (Tác giả: Joe Morgan)
6. Exploring cross-platform development with Flutter, React native, and Xamarin (Tác giả: Eric Windmill)