

**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**BÁO CÁO**

**Kỹ Thuật Lập Trình Hướng Đối Tượng**

**Đề tài:**

**Xây Dựng Ứng Dụng Quản Lý Thi Trắc Nghiệm**

Người hướng dẫn : ThS. Huỳnh Trung Trụ

Sinh viên thực hiện: Nguyễn Đăng Việt

Lê Quang Lộc

Lớp: D18CQCN03-N

Khóa : 2018-2023

Hệ : Đại Học Chính Quy

***Thành phố Hồ Chí Minh, tháng 11 năm 2023***



**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**BÁO CÁO**

**Kỹ Thuật Lập Trình Hướng Đối Tượng**

**Đề tài:**

**Xây Dựng Ứng Dụng Quản Lý Thi Trắc Nghiệm**

Người hướng dẫn : ThS. Huỳnh Trung Trụ

Sinh viên thực hiện: Nguyễn Đăng Việt

Lê Quang Lộc

Lớp: D18CQCN03-N

Khóa : 2018-2023

Hệ : Đại Học Chính Quy

***Thành phố Hồ Chí Minh, tháng 11 năm 2023***

**LỜI CẢM ƠN**

Đề tài “Ứng dụng quản lý thi trắc nghiệm” là nội dung chúng tôi chọn để nghiên cứu và làm báo cáo đồ án môn học chuyên ngành Công nghệ thông tin tại trường Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở TP. HCM.

Để hoàn thành quá trình nghiên cứu và hoàn thiện báo cáo này, lời đầu tiên chúng tôi xin chân thành gửi lời cảm ơn sâu sắc đến thầy Huỳnh Trung Trụ là giảng viên trực tiếp giảng dạy và hướng dẫn môn học Kỹ thuật lập trình hướng đối tượng. Thầy đã chỉ bảo, chia sẻ kiến thức và hướng dẫn chúng tôi trong suốt quá trình nghiên cứu và thực hiện báo cáo này.

Thành phố Hồ Chí Minh, tháng 11 năm 2023

Sinh viên thực hiện

*Nguyễn Đăng Việt – Lê Quang Lộc*

**NHẬN XÉT CỦA GIẢNG VIÊN**

**MỤC LỤC**

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 1](#_Toc150288005)

[1. Tên đề tài 1](#_Toc150288006)

[2. Sự cần thiết của đề tài 1](#_Toc150288007)

[3. Mục tiêu của đề tài 1](#_Toc150288008)

[4. Kết quả phải đạt 1](#_Toc150288009)

[5. Ý nghĩa thực tiễn 2](#_Toc150288010)

[CHƯƠNG II: PHÂN TÍCH ĐỀ TÀI 2](#_Toc150288011)

[1. Hiện trạng thực tế 2](#_Toc150288012)

[2. Sơ đồ tổng quan hệ thống 4](#_Toc150288013)

[3. Yêu cầu chức năng 5](#_Toc150288014)

[4. Yêu cầu phi chức năng 6](#_Toc150288015)

[5. Sơ đồ mô hình công nghệ ứng dụng 7](#_Toc150288016)

[6. Phác thảo sơ đồ mô hình công nghệ 8](#_Toc150288017)

[*6.1* *ReactJS* 9](#_Toc150288018)

[*6.2* *Spring Boot* 10](#_Toc150288019)

[*6.3* *MySQL* 11](#_Toc150288020)

[*6.4* *Jenkins* 12](#_Toc150288021)

[*6.5* *AWS* 13](#_Toc150288022)

[CHƯƠNG III: THIẾT KẾ PHẦN MỀM 14](#_Toc150288023)

[1. Mô hình Use Case 14](#_Toc150288024)

[*1.1* *Use Case chung dành cho tất cả người dùng* 14](#_Toc150288025)

[*1.2* *Use Case Quản lý câu hỏi* 17](#_Toc150288026)

[*1.3* *Use Case Quản lý đề thi* 20](#_Toc150288027)

[*1.4* *Use Case Quản lý môn học* 23](#_Toc150288028)

[*1.5* *Use Case Quản lý ca thi* 27](#_Toc150288029)

[*1.6* *Use Case Quản lý lớp tín chỉ* 30](#_Toc150288030)

[*1.7* *Use Case Quản lý đăng ký* 33](#_Toc150288031)

[*1.8* *Use Case Quản lý vai trò* 36](#_Toc150288032)

[*1.9* *Use Case Quản lý người dùng* 38](#_Toc150288033)

[*1.10* *Use Case Xem bảng xếp hạng* 40](#_Toc150288034)

[*1.11* *Use Case của sinh viên* 41](#_Toc150288035)

[2. Sơ đồ tuần tự 43](#_Toc150288036)

[*2.1* *Sơ đồ tuần tự chức năng thêm* 43](#_Toc150288037)

[*2.2* *Sơ đồ tuần tự chức năng sửa* 44](#_Toc150288038)

[*2.3* *Sơ đồ tuần tự chức năng xóa* 45](#_Toc150288039)

[*2.4* *Sơ đồ tuần tự chức năng hủy* 45](#_Toc150288040)

[*2.5* *Sơ đồ tuần tự chức năng làm bài thi* 46](#_Toc150288041)

[3. Thiết kế sơ đồ quan hệ thực thể (ERD) 46](#_Toc150288042)

[4. Thiết kế chi tiết thực thể 47](#_Toc150288043)

[5. Thiết kế sơ đồ lớp cơ sở dữ liệu (Class Diagrams) 48](#_Toc150288044)

[6. Từ điển dữ liệu 49](#_Toc150288045)

[CHƯƠNG IV: PHÁT TRIỂN ỨNG DỤNG 55](#_Toc150288046)

[1. Đăng nhập 55](#_Toc150288047)

[2. Quản lý câu hỏi 55](#_Toc150288048)

[3. Quản lý đề thi 59](#_Toc150288049)

[4. Quản lý môn học 60](#_Toc150288050)

[5. Quản lý ca thi 61](#_Toc150288051)

[6. Quản lý lớp tín chỉ 64](#_Toc150288052)

[7. Quản lý đăng ký 65](#_Toc150288053)

[8. Quản lý vai trò 66](#_Toc150288054)

[9. Quản lý người xếp hạng 67](#_Toc150288055)

[10. Quản lý người dùng 67](#_Toc150288056)

[11. Quản lý thi trắc nghiệm 68](#_Toc150288057)

[KẾT LUẬN 71](#_Toc150288058)

[1. Kết quả đạt được 71](#_Toc150288059)

[2. Hạn chế 71](#_Toc150288060)

[3. Hướng phát triển 71](#_Toc150288061)

[TÀI LIỆU THAM KHẢO 72](#_Toc150288062)

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

* 1. **Tên đề tài**

Ứng dụng Quản lý thi trắc nghiệm

* 1. **Sự cần thiết của đề tài**

Việc thực hiện quản lý và tổ chức thi trắc nghiệm trực tuyến mang lại nhiều lợi ích quan trọng, đặc biệt là trong ngữ cảnh của giáo dục và đào tạo hiện đại. Trong đó có lợi ích về hiệu suất, tiện ích và tiết kiệm chi phí, đồng thời tăng cường tính tương tác và an toàn của quá trình thi cũng như quản lý giáo dục. Đây được coi là mục đích quan trọng của việc triển khai hệ thống quản lý và tổ chức thi trắc nghiệm trực tuyến.

* 1. **Mục tiêu của đề tài**

Hệ thống quản lý và tổ chức thi trắc nghiệm trực tuyến mang lại nhiều hiệu quả cho nhiều đối tượng khác nhau, bao gồm sinh viên, giáo viên, người quản lý và tổ chức thi; mang lại hiệu quả cao bằng cách tối ưu hóa quy trình, cung cấp phản hồi ngay lập tức, tăng cường bảo mật, đảm bảo an toàn linh hoạt và khả năng tích hợp cho người dùng trong cả quá trình học tập và thi cử. Dưới đây là một số hiệu quả tổng quan của việc triển khai hệ thống này:

* Tăng cường an toàn và bảo mật:

Việc tích hợp các biện pháp bảo mật như chụp hình webcam giúp tăng cường an toàn cho cả sinh viên và hệ thống.

* Tạo cơ hội cho học tập trực tuyến:

Hệ thống có thể kết hợp với các nền tảng học trực tuyến để tạo cơ hội cho học tập trực tuyến và tương tác học thuật.

* Phản hồi và điều chỉnh linh hoạt:

Dữ liệu và kết quả thi có thể được sử dụng để đưa ra phản hồi và điều chỉnh chương trình học tập một cách linh hoạt.

* 1. **Kết quả phải đạt**

**−** Phân tích và thiết kế cơ sở dữ liệu.

− Thiết kế giao diện người dùng.

− Xây dựng các modules dành cho từng đối tượng người dùng.

* 1. **Ý nghĩa thực tiễn**
* *Đối với người quản lý và giảng viên*

− Tối ưu hóa hiệu suất hệ thống:

Dữ liệu thống kê có thể được sử dụng để tối ưu hóa hiệu suất hệ thống, từ việc cải thiện chất lượng đề thi đến việc tối ưu hóa quá trình chấm điểm tự động.

− Phát triển chiến lược quản lý giáo dục:

Dữ liệu thu thập từ các kết quả thi có thể hỗ trợ quyết định chiến lược quản lý và phát triển chính sách giáo dục.

− Quản lý phòng thi và đề thi:

Hệ thống giúp tổ chức thi dễ dàng quản lý thông tin về lịch thi, phòng thi và đề thi.

− Chấm điểm tự động và kết quả nhanh chóng:

Chấm điểm tự động giúp tăng tốc quy trình chấm điểm và cung cấp kết quả nhanh chóng.

* *Đối với người dự thi*

− Tiếp cận nội dung học tập:

Hệ thống có thể cung cấp sinh viên một cách tiếp cận linh hoạt đối với nội dung học tập, câu hỏi mẫu và tài liệu tham khảo.

− Tăng cường trải nghiệm học tập:

Phương tiện tương tác như video, âm thanh, và đồ họa có thể được tích hợp để tăng cường trải nghiệm học tập của sinh viên

**CHƯƠNG II: PHÂN TÍCH ĐỀ TÀI**

1. **Hiện trạng thực tế**

* Trường có nhiều khoa, mỗi khoa có mã khoa để phân biệt các khoa với nhau và tên khoa.
* Mỗi khoa quản lý nhiều lớp, mỗi lớp chỉ thuộc một khoa. Mỗi lớp có mã lớp để phân biệt các lớp với nhau và tên lớp, năm nhập học.
* *Thông tin về sinh viên:*
* Mỗi sinh viên sẽ có mã sinh viên để phân biệt các sinh viên, họ, tên, ngày sinh, giới tính, địa chỉ, email, mật khẩu để đăng nhập vào phần mềm thi trắc nghiệm. Giá trị mặc định của mật khẩu là giống với mã sinh viên và được viết thường tất cả các chữ cái. Ví dụ đối với mã sinh viên là N18DCCN001 thì mật khẩu mặc định của sinh viên sẽ là n18dccn001.
* Mỗi sinh viên chỉ thuộc một lớp.
* Mỗi sinh viên có thể đăng ký các lớp tín chỉ của khoa mà sinh đang theo học và một lớp tín chỉ có thể có nhiều sinh viên đăng ký. Ứng với mỗi đăng ký lớp tín chỉ, sinh viên được thi tối đa 2 lần là giữa kỳ và cuối kỳ.
* *Thông tin về giảng viên:*
  + Mỗi giảng viên sẽ có các thuộc tính như sinh viên.
  + Giảng viên là người chịu trách nhiệm soạn thảo câu hỏi của các môn học để đưa vào bộ đề nguồn.
  + Một câu hỏi chỉ có thể được tạo ra bởi một giảng viên. Một giảng viên có thể soạn ra nhiều câu hỏi.
  + Một đề thi chỉ có thể được tạo ra bởi một giảng viên. Một giảng viên có thể soạn ra nhiều đề thi. Khi soạn đề thi cho phép giảng viên sử dụng các tiêu chí chọn câu hỏi như tên chương, mức độ và số lượng câu hỏi.
  + Một giảng viên có thể dạy nhiều lớp tín chỉ, một lớp tín chỉ chỉ được giảng dạy bởi một giảng viên.
  + Một giảng viên có thể tạo ra nhiều ca thi, một ca thi chỉ được tạo ra bởi một giảng viên.
* *Thông tin về đề thi:*
* Một đề thi sẽ bao gồm nhiều câu hỏi, một câu hỏi có thể thuộc nhiều đề thi khác nhau. Một câu hỏi có thể thuộc các loại sau đây: một lựa chọn, nhiều lựa chọn, điền khuyết và mức độ phân loại của câu hỏi.
* Một đề thi chỉ được sử dụng duy nhất cho một ca thi và không thể tái sử dụng lại.
* Một đề thi phải thuộc về một môn học cụ thể và một môn học có thể có nhiều đề thi.
* *Thông tin về ca thi:* 
  + Một ca thi có thể sử dụng nhiều đề thi khác nhau.
  + Một ca thi sẽ bao gồm nhiều đăng ký lớp tín chỉ và tương ứng một đăng ký lớp tín chỉ sẽ có thể thi tối đa 2 lần(tương ứng với giữa kỳ và cuối kỳ).
  + Một lần thi của sinh viên sẽ có nhiều chi tiết thi, chi tiết thi này sẽ bao gồm các câu hỏi của đề thi và câu trả lời của sinh viên.

Trong một trường đại học, đề tài này nhắm đến ba đối tượng sử dụng chính: sinh viên, giảng viên, quản trị viên. Quản trị viên có thể là các giảng viên, hoặc các thầy cô có quyền cao hơn trong một hệ thống trường đại học. Đối với mỗi nhóm người dùng chính này, hệ thốn sẽ đảm bảo các chức năng cơ bản phục vụ việc quản lý kết quả thi và cho thi một cách tự động trên máy.

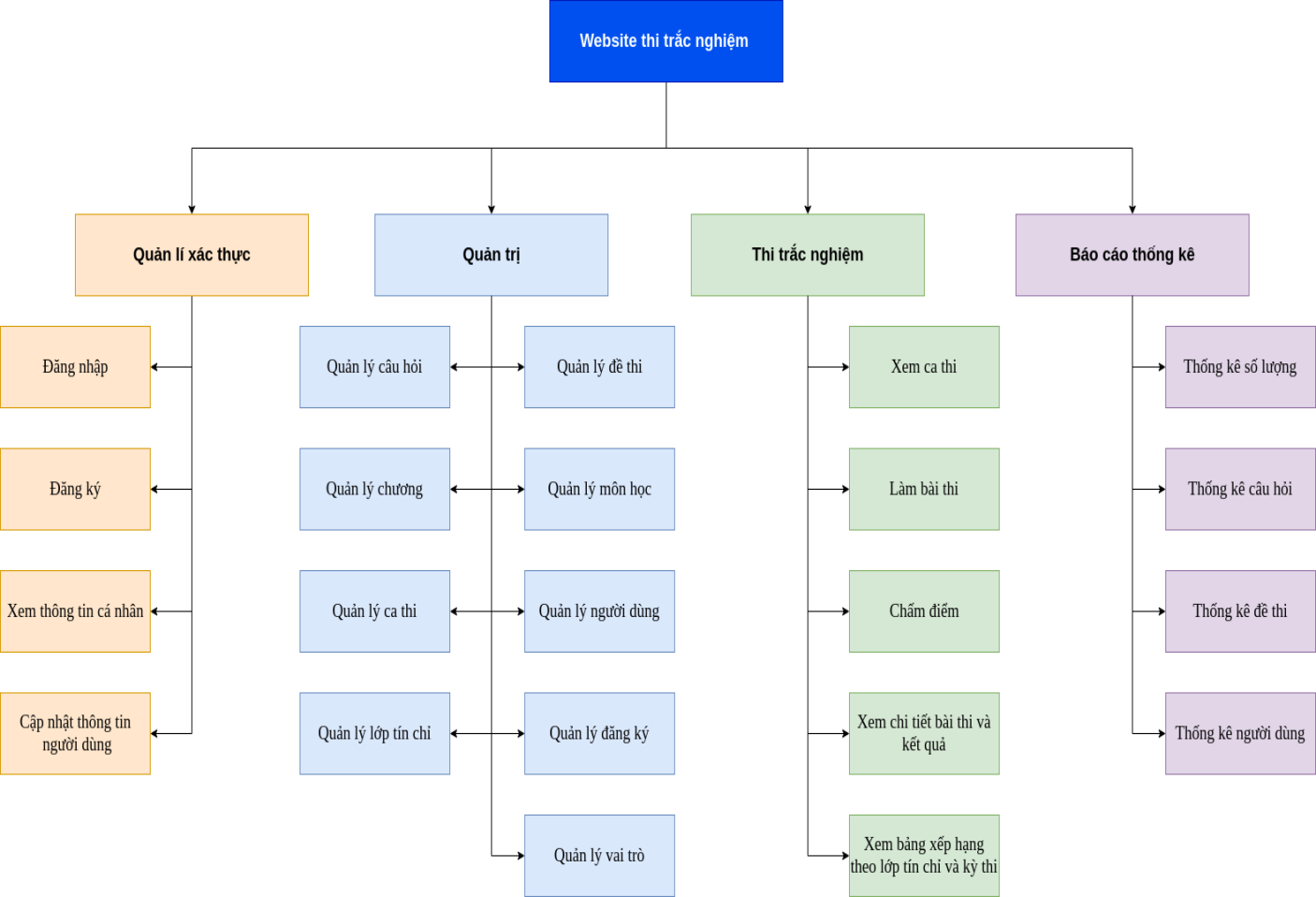
Sau khi đăng nhập thành công vào hệ thống (dùng MASV), sinh viên chọn ra ca thi để thi, chương trình sẽ tải ra các câu hỏi của đề thi đó và thời gian thi. Nếu như sinh viên đã thi rồi thì không cho thi nữa. Các câu hỏi trong đề thi phải là ngẫu nhiên và không trùng nhau.

Sinh viên có thể chọn bất kỳ câu hỏi nào để trả lời mà không cần phải trả lời theo thứ tự câu hỏi.

Sau khi hết giờ làm bài (hoặc chọn kết thúc bài thi) thì chương trình sẽ thông báo kết quả cho sinh viên ngay. Bao gồm các thông tin quan trọng như mã sinh viên, họ tên sinh viên, lớp, môn thi, ca thi, ngày thi, điểm số. Sinh viên cũng có thể xem lại kết quả của bài vừa thi.

Sinh viên có thể bảng xếp hạng theo ca thi và lớp tín chỉ.

1. **Sơ đồ tổng quan hệ thống**



*Hình 2.1: Sơ đồ Tổng quan hệ thống*

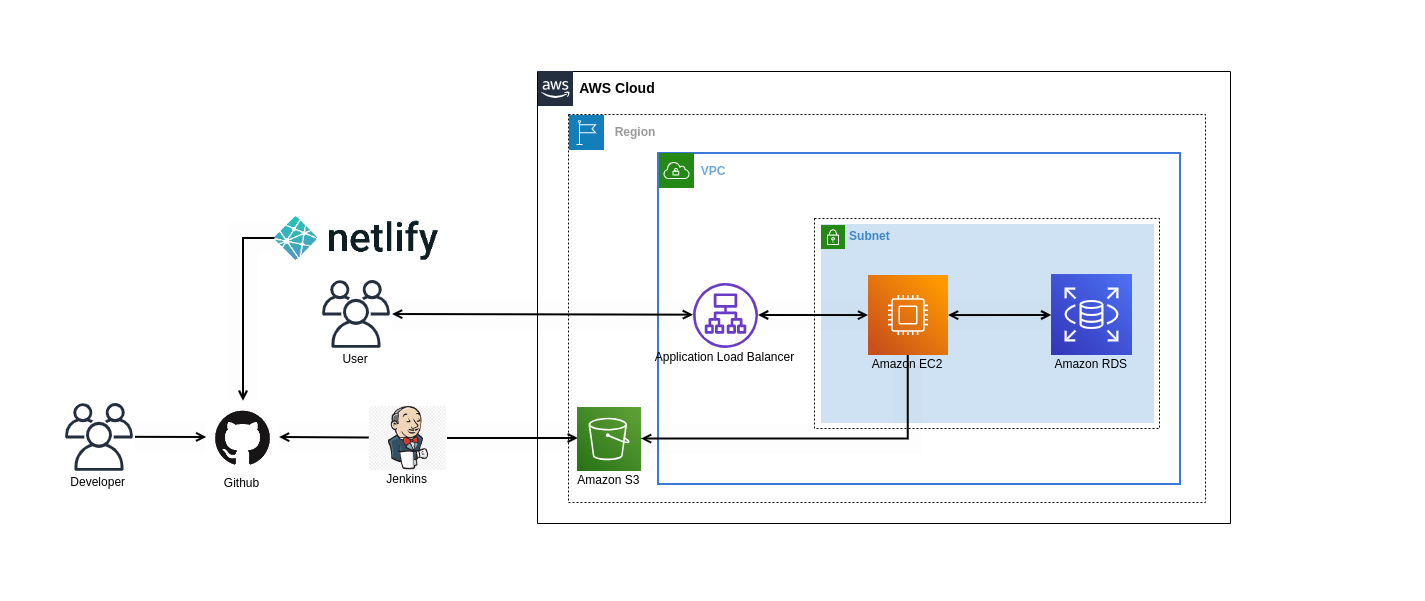
1. **Yêu cầu chức năng**

* Quản trị viên:
  + Quản lý toàn bộ hệ thống:
* Câu hỏi.
* Đề thi.
* Môn học.
* Chương.
* Ca thi.
* Lớp tín chỉ.
* Đăng ký.
* Người dùng.
* Xem bảng xếp hạng.
* Xem kết quả thi của sinh viên.
  + Thống kê thông số cơ bản của hệ thống.
* Giảng viên:
  + Quản lý câu hỏi.
  + Quản lý đề thi.
  + Xem môn học, quản lý chương.
  + Tạo ca thi.
  + Xem lớp tín chỉ.
  + Xem bảng xếp hạng.
  + Xem kết quả thi.
* Sinh viên:
  + Xem danh sách ca thi.
  + Làm bài thi.
  + Xem bảng xếp hạng.
  + Xem kết quả thi.
  + Xem lại các bài thi đã thi.

1. **Yêu cầu phi chức năng**

* Hiệu năng hoạt động: Đảm bảo thời gian đáp ứng các yêu cầu chức năng của người dùng một cách nhanh nhất có thể.
  + - Giới hạn thông tin trả về vừa đủ để đáp ứng yêu cầu người dùng, không trả về các thông tin không cần thiết.
    - Tối ưu hóa truy vấn cơ sở dữ liệu (chiếu, chọn trước, kết sau).
* Tính sẵn có: Đảm bảo hệ thống có thể hoạt động bất cứ lúc nào người dùng có nhu cầu truy cập.
* Sức chứa: Đảm bảo thông tin về hệ thống được lưu trữ vào cơ sở dữ liệu một cách đầy đủ và không bị giới hạn.
* Độ khả dụng: Đảm bảo các chức năng có thể được sử dụng một cách dễ dàng và thoải mái.
* Độ bảo mật: Thực hiện các biện pháp xác thực người dùng, phân quyền. Đảm bảo không thể truy cập các nguồn tài nguyên ngoài quyền hạn của người dùng.
* Khả năng phục hồi: Thực hiện sao lưu cơ sở dữ liệu và source code để có thể phục hồi trong các trường hợp bất ngờ xảy ra.
* Toàn vẹn dữ liệu: Đảm bảo dữ liệu hay thông tin không bị thay đổi, mất mát trong khi lưu trữ hay truyền tải.
* Khả năng sử dụng: Đảm bảo các chức năng không thể thiếu của hệ thống phải có như quản lý ca thi
* Khả năng tương tác: Đảm bảo người dùng có khả năng tương tác với hệ thống thông qua giao diện dễ sử dụng và hiệu quả.

1. **Sơ đồ mô hình công nghệ ứng dụng**



*Hình 2.2: Sơ đồ Mô hình công nghệ ứng dụng*

Hệ thống được xây dựng dựa trên mô hình client server theo kiến trúc REST API. Hệ thống hoạt động theo mô hình client-server, trong đó server là tập hợp các service nhỏ lắng nghe các request từ client. Với từng request khác nhau thì có thể một hoặc nhiều service xử lý.

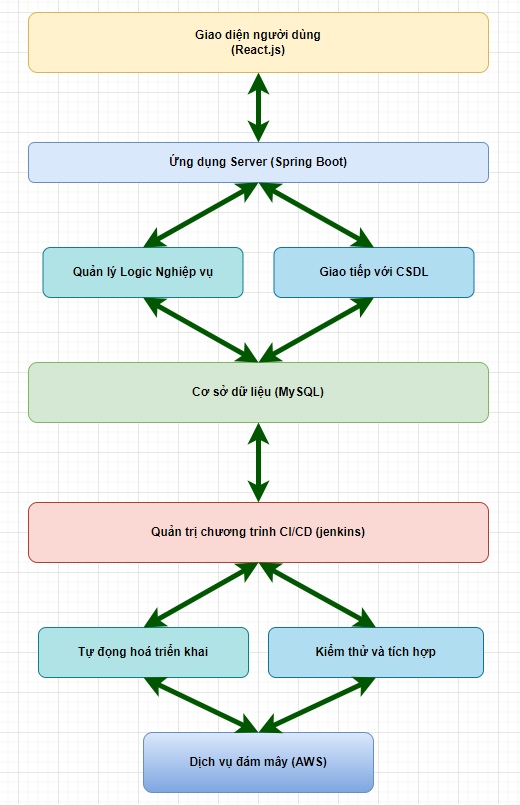
Server và client không lưu trạng thái của nhau, mỗi request lên server thì client phải đóng gói thông tin đầy đủ để server hiểu được. Điều này giúp hệ thống dễ phát triển,bảo trì, mở rộng vì không cần tốn công CRUD trạng thái của client . Hệ thống phát triển theo hướng này có ưu điểm nhưng cũng có khuyết điểm là gia tăng lượng thông tin cần truyền tải giữa client và server.

Chuẩn hóa các interface : Đây là một trong những đặc tính quan trọng của hệ thống REST. Bằng cách tạo ra các quy ước chuẩn để giao tiếp giữa các thành phần trong hệ thống, đơn giản hóa việc client có thể tương tác với server. Các quy ước này áp dụng cho toàn bộ các service giúp cho người sử dụng hệ thống dễ sử dụng hơn.

Phân lớp hệ thống : trong hệ thống REST chia tách các thành phần hệ thống theo từng lớp, mỗi lớp chỉ sử dụng lớp ở dưới nó và giao tiếp với lớp ở ngay trên nó mà thôi. Điều này giúp giảm độ phức tạp của hệ thống, giúp các thành phần tách biệt nhau từ đó dễ dàng mở rộng từng thành phần.

Giúp cho ứng dụng trở nên rõ ràng hơn. REST URL đại diện cho resource chứ không phải là hành động. Dữ liệu được trả về với định dạng json. Code đơn giản và ngắn gọn. Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

1. **Phác thảo sơ đồ mô hình công nghệ**

****

*Hình 2.3: Phác thảo Sơ đồ mô hình công nghệ*

* 1. *ReactJS*

Các thư viện ReactJS tiêu biểu sử dụng trong đồ án

* @reduxjs/toolkit: Quản lý trạng thái cho React component và thực hiện các asynchronous task.
* axios: Xây dựng các API.
* styled-components: Tạo ra các customized component tái sử dụng.
* @material-table: Dùng cho việc tạo ra các table nhanh chóng.
* alertifyjs: Thông báo.
* @material-ui: Các component phổ biến được phát triển bởi Google.
* chart.js: Biểu đồ.
* react-router-dom: Điều hướng trang web.
* material-ui/icons: Thư viện icons.
* Flowbite-react: Thư viện UI.
  1. *Spring Boot*

Spring Boot chính là một Java framework siêu to và khổng lồ và có nhiều khả năng hữu ích vì nó có thể giúp lập trình viên giải quyết rất nhiều vấn đề. So với framework Spring thông thường, Spring Boot tỏ ra những lợi thế vượt trội. Khi sử dụng Spring Boot, rất nhiều thứ được cải tiến hỗ trợ lập trình viên như:

• Auto config: tự động cấu hình.

• Dựa trên các Annotation để tạo lập các bean thay vì XML.

• Server Tomcat có thể được nhúng ngay trong file JAR build ra và có thể chạy ở bất kì đâu mà java chạy được.

***Ưu điểm nổi bật của Spring Boot:***

1. Phát triển web một cách đơn giản và nhanh chóng.

2. Config an toàn.

3. Có thể hỗ trợ YAML.

4. Có tính quản trị cao.

5. Dễ dàng ứng dụng Spring và các sự kiện.

6. Có thể cấu hình ở bên ngoài và tạo ra những tệp thuộc tính.

7. Tính bảo mật cao.

8. Ghi log.

9. Externalized Configuration: Spring Boot cho phép người dùng có thể sử dụng cấu hình (config) từ bên ngoài. Chính vì vậy mà mỗi ứng dụng được tạo ra có thể chạy được trên nhiều môi trường khác nhau.

10. Profiles: Dùng để phân chia các loại cấu hình cho các môi trường khác nhau.

11. Loggin: Sử dụng phục vụ cho toàn bộ chức năng log trong phạm vi nội bộ. Những logging này sẽ được quản lý một cách mặc định.

12. Ngoài ra còn rất nhiều các tính năng tương tự khác của Spring Boot như: Developing web Applications, Working with SQL Technologies, Security, Messaging, Caching, Calling rest Services with RestTemplate/WebClient, Sending Email, Validation,….

* 1. *MySQL*
* *Các tính năng cốt lõi của MySQL:*
* MySQL cho phép dữ liệu được lưu trữ và truy cập trên nhiều công cụ lưu trữ, bao gồm InnoDB, CSV và NDB. MySQL cũng có khả năng sao chép dữ liệu và phân vùng bảng để có hiệu suất và độ bền tốt hơn. Người dùng MySQL không bắt buộc phải học các lệnh mới; họ có thể truy cập dữ liệu của mình bằng các lệnh SQL tiêu chuẩn.
* MySQL được viết bằng C và C++ và có thể truy cập và có sẵn trên hơn 20 nền tảng, bao gồm Mac, Windows, Linux và Unix. RDBMS hỗ trợ cơ sở dữ liệu lớn với hàng triệu bản ghi và hỗ trợ nhiều loại dữ liệu bao gồm các số nguyên có chữ ký hoặc không dấu có độ dài 1, 2, 3, 4 và 8 byte(s); FLOAT; DOUBLE; CHAR; VARCHAR; BINARY; VARBINARY; TEXT; BLOB; DATE; TIME; DATETIME; TIMESTAMP; YEAR; SET; ENUM; và các kiểu OpenGIS. Các loại chuỗi có độ dài cố định và biến đổi cũng được hỗ trợ.
* Để bảo mật, MySQL sử dụng một đặc quyền truy cập và hệ thống mật khẩu được mã hóa cho phép xác minh dựa trên máy chủ. Các máy khách MySQL có thể kết nối với Máy chủ MySQL bằng một số giao thức, bao gồm cả giao thức TCP/IP trên bất kỳ nền tảng nào. MySQL cũng hỗ trợ một số chương trình máy khách và tiện ích, chương trình dòng lệnh và công cụ quản trị như MySQL Workbench.
  1. *Jenkins*
* Jenkins là một dự án mã nguồn mở được viết chủ yếu bởi ngôn ngữ Java dùng để tự động hóa tất cả mọi tác vụ liên quan đến xây dựng, kiểm thử, luôn tục tích hợp hoặc triển khai sản phẩm đến người dùng cuối. Nó được sử dụng để thực hiện các workflow CI/CD (gọi là các pipelines).
* Các pipelines tự động hóa quá trình test, công bố những báo cáo những thay đổi độc lập trong code base lớn. Ngoài ra, nó còn hỗ trợ việc tích hợp các nhánh riêng biệt của code thành một nhánh chính. Pipelines còn liên tục phát hiện các lỗi, build, tự động test, chuẩn bị code base cho việc deploy. Cuối cùng, các code được deploy đến container và máy ảo, cũng như bare metal và các server cloud.
* Jenkins được cài đặt dễ dàng trên các thiết bị có sẵn Java và Tomcat. Hỗ trợ xây dựng các phần mềm build tool phổ biến hiện nay như Maven, Gradle, Ant,...Cung cấp hơn 1700 tiện ích tích hợp với hầu hết các công cụ và open source khác như Docker, Cloud service, Ansible,…
* *Ứng dụng Jenkins trong đề tài*

Một pipeline đơn giản được triển khai với thứ tự như sau:

* + Build: Tải source code từ github và build project dùng maven
  + Send Build Artifact To AWS S3: Tải file .war vừa build được từ bước trước lên AWS S3.
  + Deploy Application Production Stage: Dùng tiện ích SSH Publisher của Jenkins để chạy đoạn mã cập nhật file .war mới nhất cho ứng dụng.
  1. *AWS*

Amazon web services hoạt động dựa trên nền tảng cloud computing (điện toán đám mây) vậy điện toán đám mây là gì ? Điện toán đám mây thường được gọi là "đám mây", là việc phân phối các tài nguyên máy tính theo yêu cầu - tất cả mọi thứ từ các ứng dụng đến các trung tâm dữ liệu-qua internet trên cơ sở trả cho sử dụng.

* Tài nguyên đàn hồi - Quy mô lên hoặc xuống nhanh và dễ dàng để đáp ứng nhu cầu.
* Dịch vụ đo đếm, do đó bạn chỉ phải trả cho những gì bạn sử dụng.
* Tự phục vụ - Tất cả tài nguyên CNTT cần có quyền truy cập tự phục vụ.
* *Ứng dụng AWS trong đề tài*

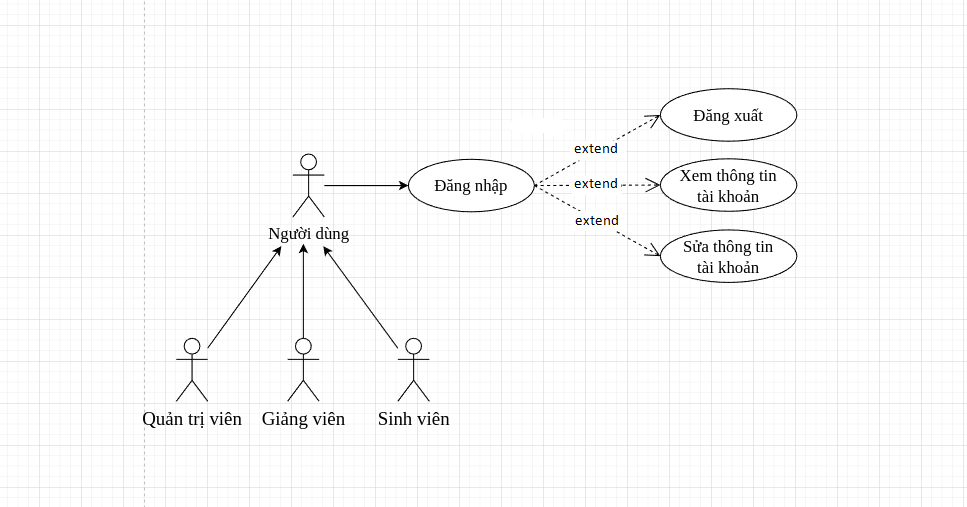
Trong đề tài này, sử dụng 2 dịch vụ của AWS:

* RDS: Relational Database Service là một dịch vụ web cho phép dễ dàng cấu hình một cơ sở dữ liệu quan hệ trên cloud. Đề tài sử dụng DB engine MySQL cho cơ sở dữ liệu của ứng dụng. Có thể dễ dàng truy cập từ EC2 instance.
* EC2: Elastic Compute Cloud là một dịch vụ web cung cấp năng lực điện toán bảo mật và có kích cỡ linh hoạt trên đám mây. Dịch vụ này được dùng để host web server cho ứng dụng, công bố API ra Internet.

**CHƯƠNG III: THIẾT KẾ PHẦN MỀM**

## Mô hình Use Case

### *Use Case chung dành cho tất cả người dùng*



Hình 3.1.1 Use case chức năng chung cho tất cả người dùng

#### Use Case đăng nhập

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Use Case đăng nhập. |
| Actor | Quản trị viên, giảng viên, sinh viên gọi chung là người dùng. |
| Điều kiện kích hoạt | Người dùng chọn chức năng đăng nhập. |
| Điều kiện trước | Để đăng nhập vào hệ thống tài khoản người dùng phải được tạo từ trước, mật khẩu phải chính xác, đã được phân quyền trước và dựa trên phân quyền để xác định tài nguyên có thể truy cập của người dùng. |
| Điều kiện sau | Sau khi đăng nhập thành công, người dùng sẽ có thể sử dụng các chức năng theo phân quyền của người dùng. |
| Luồng sự kiện chính | 1. Người dùng chọn chức năng đăng xuất, trang sẽ chuyển hướng người dùng về chức năng đăng nhập.  2. Hệ thống yêu cầu người dùng nhập mã người dùng và mật khẩu.  3. Kiểm tra thông tin người dùng như người dùng tồn tại và mật khẩu có chính xác.  4. Nếu các thông tin là chính xác, người dùng sẽ được đăng nhập vào hệ thống và chuyển hướng đến trang chủ của ứng dụng.  5. Kết thúc Use Case. |
| Luồng sự kiện phụ | Dữ liệu đầu vào không thỏa các điều kiện sau hệ thống sẽ báo lỗi.  1. Thông tin đăng nhập rỗng.  2. Người dùng không tồn tại.  3. Mật khẩu không chính xác. |

#### Use Case đăng xuất

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Use case đăng xuất. |
| Actor | Quản trị viên, giảng viên, sinh viên gọi chung là người dùng. |
| Điều kiện kích hoạt | Người dùng chọn chức năng xuất. |
| Điều kiện trước | Khi người dùng đang ở trong hệ thống, người dùng có thể chọn chức năng đăng xuất để thoát ra khỏi hệ thống. |
| Điều kiện sau | Sau khi đăng xuất thành công người dùng sẽ được chuyển hướng về trang đăng nhập. |
| Luồng sự kiện chính | 1. Người dùng chọn chức năng đăng xuất.  2. Hệ thống sẽ xóa thông tin đăng nhập của người dùng.  3. Chuyển hướng về trang đăng nhập.  4. Kết thúc Use Case. |

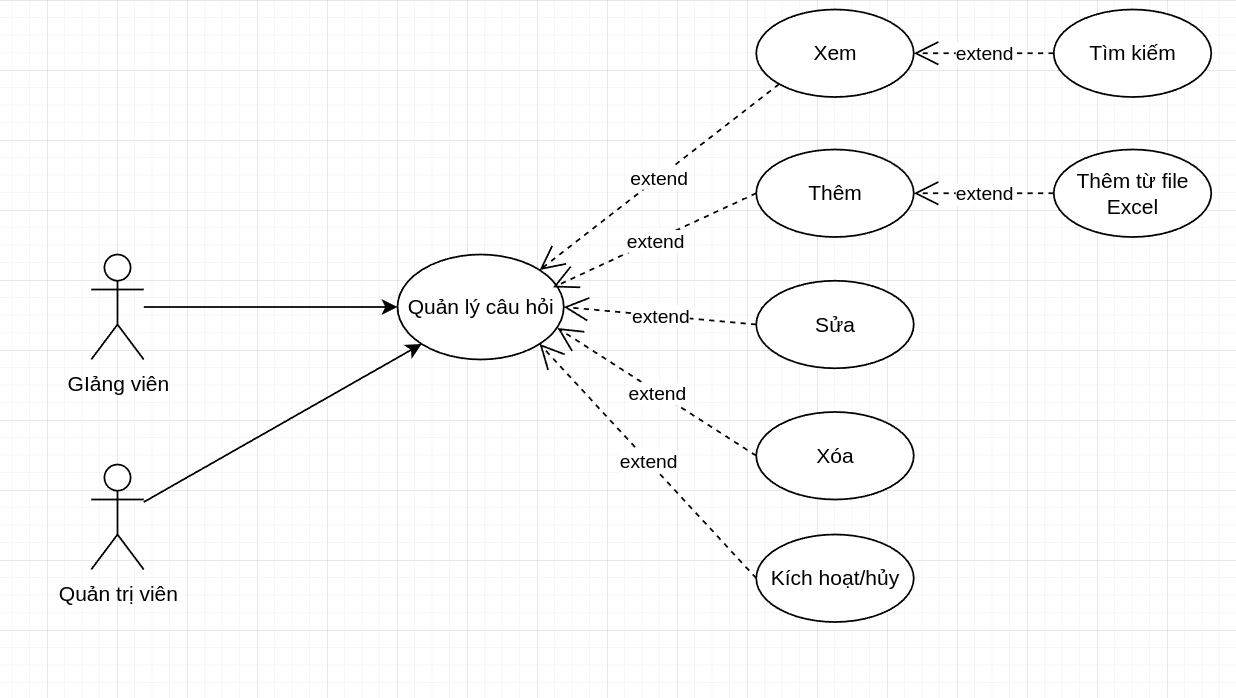
#### Use Case xem thông tin tài khoản

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Use Case xem thông tin tài khoản |
| Actor | Quản trị viên, giảng viên, sinh viên gọi chung là người dùng. |
| Điều kiện kích hoạt | Người dùng chọn chức năng xem thông tin tài khoản. |
| Điều kiện trước | Người dùng phải đăng nhập vào hệ thống. |
| Điều kiện sau | Hiển thị thông tin cá nhân của người dùng. |

#### Use Case sửa thông tin cá nhân

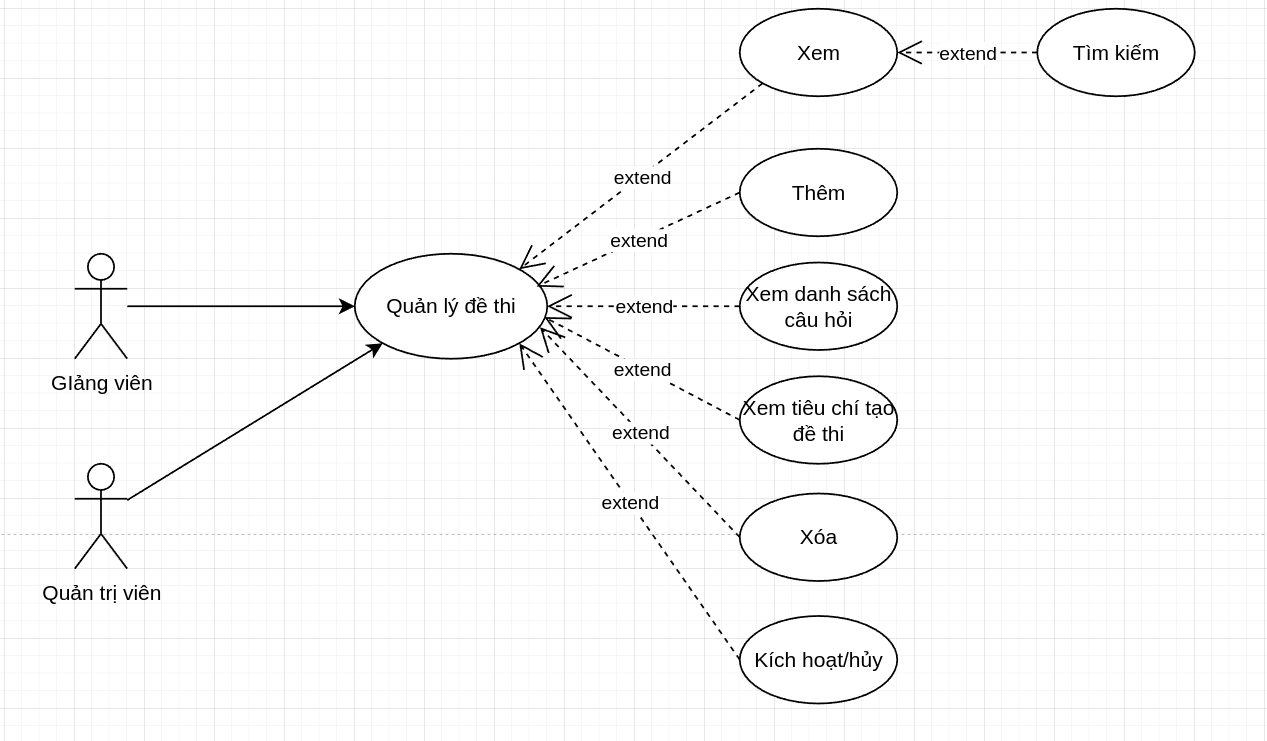
|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Use Case sửa thông tin cá nhân. |
| Actor | Quản trị viên, giảng viên, sinh viên gọi chung là người dùng. |
| Điều kiện kích hoạt | Người dùng chọn chức năng sửa thông tin cá nhân. |
| Điều kiện trước | Người dùng đăng nhập vào hệ thống và chọn chức năng sửa thông tin cá nhân. |
| Điều kiện sau | Người dùng thay đổi thông tin cần chỉnh sửa, hệ thống sẽ lưu lại thông tin mới cho người dùng. |
| Luồng sự kiện chính | 1. Chọn chức năng sửa thông tin cá nhân.  2. Hiển thị form sửa thông tin.  3. Thay đổi trường dữ liệu cần sửa.  4. Kiểm tra tính hợp lệ của dữ liệu.  5. Cập nhật cơ sở dữ liệu.  6. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  1. Thông tin để trống ở các trường không được phép trống.  2. Địa chỉ email đã tồn tại.  3. Ngày sinh không đủ 18 tuổi.  4. Không tìm thấy người dùng. |

### *Use Case Quản lý câu hỏi*

Hình 3.1.2 Use Case Quản lý câu hỏi

|  |  |  |
| --- | --- | --- |
| **Use Case** | **Nội dung** | |
| Mô tả | Cho phép quản trị viên hoặc giảng viên thực hiện các chức năng liên quan đến câu hỏi. | |
| Actor | Quản trị viên, giảng viên. | |
| Điều kiện kích hoạt | Quản trị viên hoặc giảng viên chọn chức năng quản lý câu hỏi. | |
| Điều kiện trước | Quản trị viên hoặc giảng viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên” hoặc “Giảng viên”. | |
| Điều kiện sau | Hệ thống sẽ tải danh sách các câu hỏi (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Tìm kiếm câu hỏi theo môn học, người biên soạn.  2. Thêm câu hỏi.  3. Sửa câu hỏi.  4. Xóa câu hỏi.  5. Kích hoạt/hủy câu hỏi. | |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm câu hỏi:**  1. Chọn một môn học hoặc 1 người biên soạn từ Select.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách câu hỏi dựa trên tiêu chí lọc của người dùng.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có câu hỏi nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm câu hỏi:**  1. Chọn chức năng thêm câu hỏi thông thường.  2. Điền thông tin câu hỏi và nhấn nút thêm.  3. Hệ thống sẽ kiểm tra các dữ liệu đầu vào.  4. Lưu vào cơ sở dữ liệu.  5. Thông báo thêm thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm câu hỏi từ file Excel:**  1. Chọn chức năng thêm câu hỏi từ file Excel.  2. Form chọn file Excel hiển thị ra.  3. Người dùng chọn file Excel với đuôi kết thúc bằng .xlsx và nhấn nút Thêm tất cả.  4. Hệ thống kiểm tra các câu hỏi thỏa mãn các ràng buộc dữ liệu.  5. Hệ thống sẽ lưu vào cơ sở dữ liệu các câu hỏi chưa được thêm thông qua kiểm tra nội dung câu hỏi.  6. Thêm mới chương và môn học nếu chưa tồn tại.  7. Thông báo thêm được bao nhiêu trên tổng số câu hỏi trong file Excel thành công.  8. Kết thúc Use Case.  **- Luồng sự kiện chính khi sửa câu hỏi:**  1. Chọn chức năng sửa câu hỏi ở phần thao tác.  2. Hệ thống hiện ra form với các trường dữ liệu được set giá trị mặc định là các thuộc tính của câu hỏi.  2. Thay đổi dữ liệu trường cần sửa và nhấn nút sửa.  3. Hệ thống sẽ kiểm tra các dữ liệu đầu vào.  4. Lưu vào cơ sở dữ liệu.  5. Thông báo sửa thành công.  6.Kết thúc Use Case.  **- Luồng sự kiện chính khi xóa câu hỏi:**  1. Chọn chức năng xóa câu hỏi ở phần thao tác.  2. Hệ thống thực hiện thao tác xóa câu hỏi.  3. Thông báo lỗi nếu câu hỏi không thể xóa.  4. Thông báo xóa thành công nếu không vi phạm ràng buộc dữ liệu.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi kích hoạt/hủy câu hỏi:**  1. Chọn chức năng kích hoạt/hủy câu hỏi ở phần thao tác.  2. Hệ thống kiểm tra trạng thái hiện tại của câu hỏi và gửi yêu cầu thay đổi trạng thái câu hỏi, khác với trạng thái hiện tại của câu hỏi.  3. Cập nhật cơ sở dữ liệu.  4. Thông báo kích hoạt/huỷ thành công.  5. Kết thúc Use Case. | |
| Luồng sự kiện phụ | | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  **- Khi thêm câu hỏi:**  1. Thông tin rỗng ở các trường dữ liệu NOT NULL.  2. Nội dung câu hỏi đã tồn tại.  3. Không tìm thấy chương.  **- Khi thêm câu hỏi bằng file Excel:**  1.Không thể lấy được giá trị từ cell trong file Excel.  **- Khi sửa câu hỏi:**  1. Thông tin rỗng ở các trường dữ liệu NOT NULL.  2. Nội dung câu hỏi đã tồn tại.  3. Không tìm thấy câu hỏi.  4. Không tìm thấy chương.  **- Khi xóa câu hỏi:**  1.Vi phạm ràng buộc dữ liệu. | |

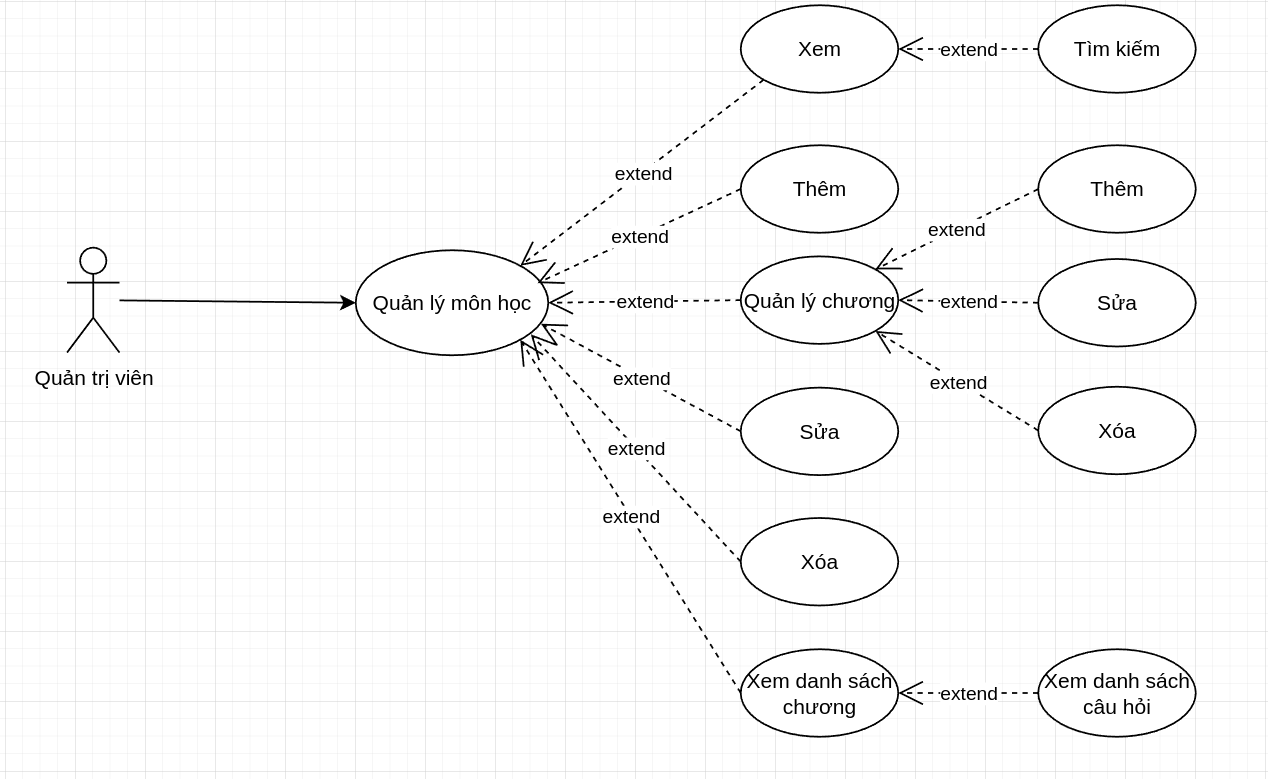
### *Use Case Quản lý đề thi*

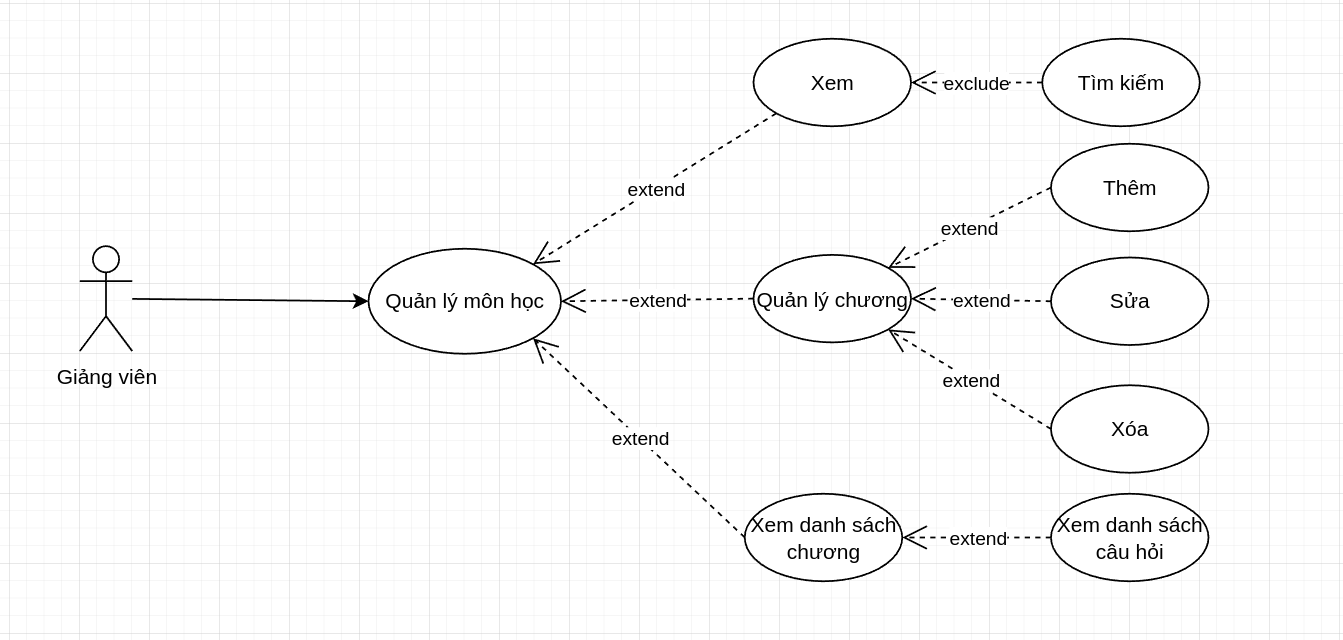
Hình 3.1.3: Use Case Quản lý đề thi

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên hoặc giảng viên thực hiện các chức năng liên quan đến đề thi. |
| Actor | Quản trị viên, giảng viên. |
| Điều kiện kích hoạt | Quản trị viên hoặc giảng viên chọn chức năng quản lý đề thi. |
| Điều kiện trước | Quản trị viên hoặc giảng viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên” hoặc “Giảng viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách các đề thi (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Tìm kiếm đề thi theo môn học.  2. Thêm đề thi.  3. Xem danh sách câu hỏi.  4. Xem danh sách tiêu chí tạo đề thi.  5. Xóa đề thi.  6. Kích hoạt/hủy đề thi. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm đề thi:**  1. Chọn một môn học từ Select.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách đề thi dựa trên tiêu chí lọc của người dùng.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có đề thi nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm đề thi:**  1. Chọn chức năng thêm đề thi.  2. Điền thông tin đề thi, điền số lượng câu hỏi rồi bấm tải câu hỏi hoặc là điền tiêu chí lọc câu hỏi và bấm tải câu hỏi.  3. Hệ thống sẽ tải ra danh sách các câu hỏi ngẫu nhiên thõa mãn các tiêu chí lọc để người dùng xem trước.  4. Bấm nút thêm, lưu thông tin đề thi vào cơ sở dữ liệu.  5. Thông báo thêm thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi xóa đề thi:**  1. Chọn chức năng xóa đề thi ở phần thao tác. Chỉ có thể xóa đề thi chưa được sử dụng bởi ca thi nào.  2. Hệ thống thực hiện thao tác xóa đề thi.  3. Thông báo lỗi nếu đề thi không thể xóa.  4. Thông báo xóa thành công nếu không vi phạm ràng buộc dữ liệu.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi kích hoạt/hủy đề thi:**  1. Chọn chức năng kích hoạt/hủy đề thi ở phần thao tác. Chỉ có thể hủy đề thi chưa được sử dụng.  2. Hệ thống kiểm tra trạng thái hiện tại của đề thi và gửi yêu cầu thay đổi trạng thái đề thi, khác với trạng thái hiện tại của đề thi.  3. Cập nhật cơ sở dữ liệu.  4. Thông báo kích hoạt/huỷ thành công.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi xem danh sách câu hỏi:**  1. Chọn chức năng xem danh sách câu hỏi ở phần thao tác.  2. Hệ thống hiển thị ra danh sách các câu hỏi của đề thi đó.  3. Kết thúc Use Case.  **- Luồng sự kiện chính khi xem danh sách tiêu chí tạo đề thi:**  1. Di chuột(hover) vào chức năng xem danh sách tiêu chí tạo đề thi.  2. Tooltip hiện ra danh sách các tiêu chí bao gồm: chương, độ khó và số lượng câu hỏi.  3. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  **- Khi thêm đề thi:**  1. Tên đề thi đã tồn tại.  2. Không có danh sách câu hỏi.  **- Khi xóa đề thi:**  1. Vi phạm ràng buộc dữ liệu. |

### *Use Case Quản lý môn học*

Hình 3.1.4.1: Use Case Quản lý môn học của quản trị viên

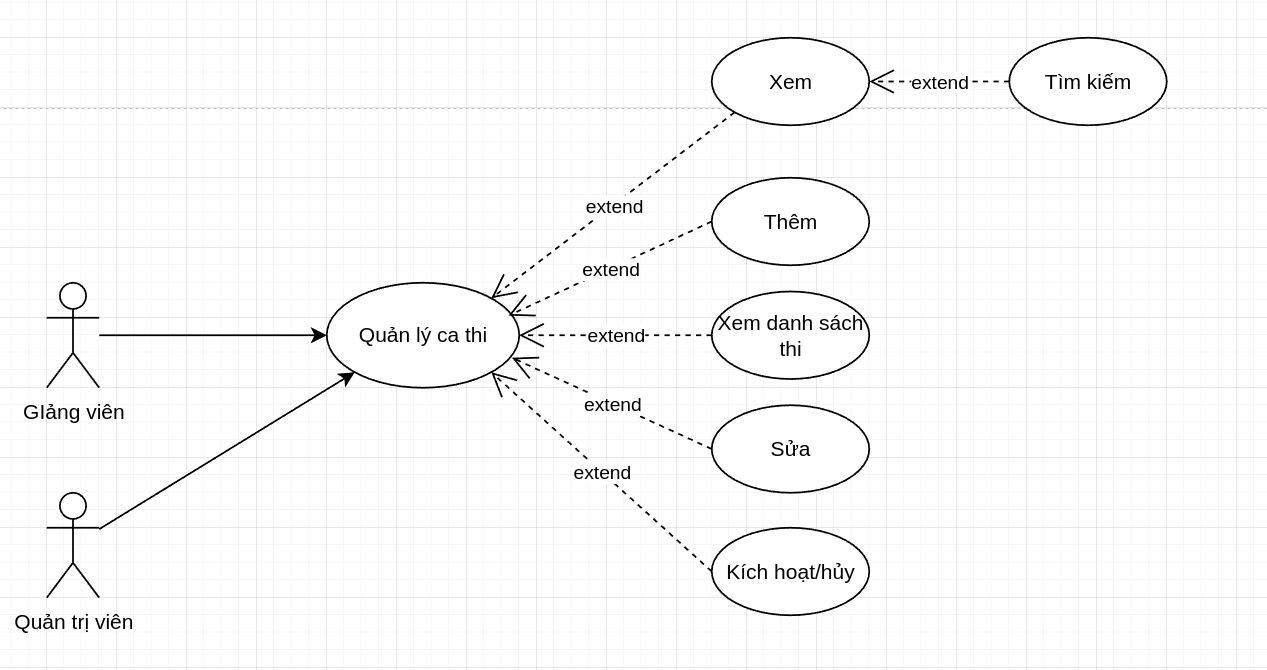




Hình 3.1.4.2: Use Case Quản lý môn học của giảng viên

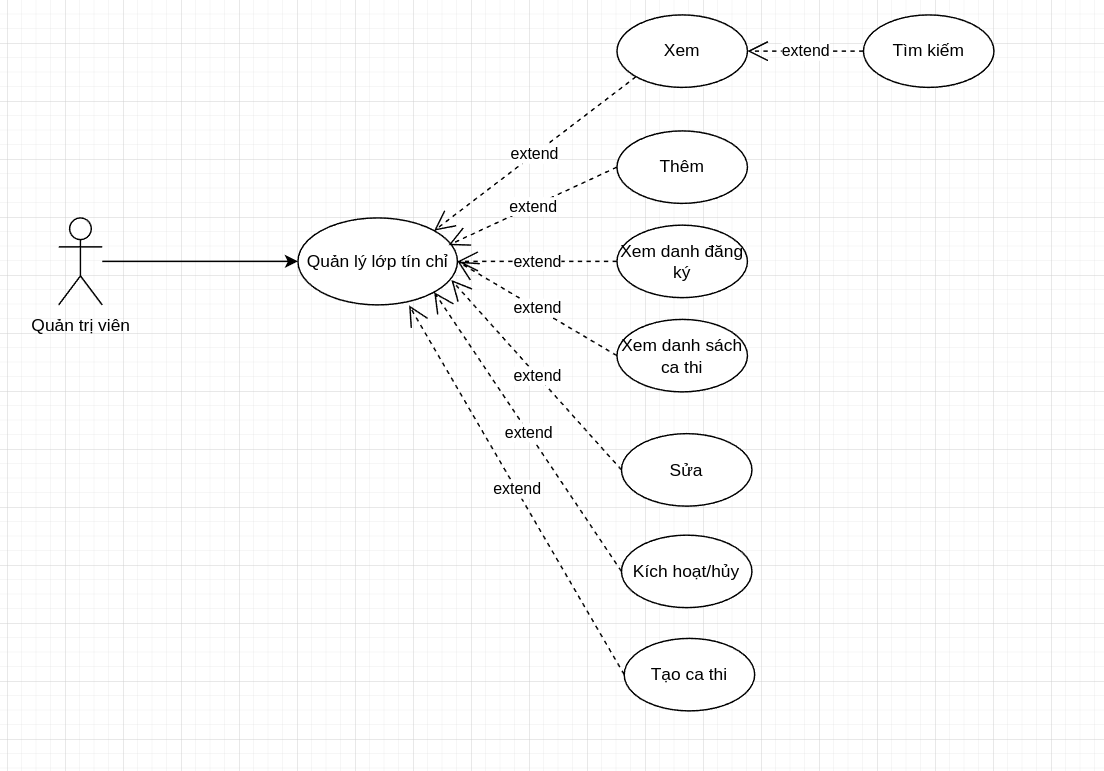
|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên, giảng viên thực hiện các chức năng liên quan đến môn học. |
| Actor | Quản trị viên, giảng viên. |
| Điều kiện kích hoạt | Quản trị viên, giảng viên chọn chức năng quản lý môn học. |
| Điều kiện trước | Quản trị viên hoặc giảng viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên” hoặc “Giảng viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách các môn học (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  **- Đối với quản trị viên:**  1. Tìm kiếm môn học theo tên và mã môn học.  2. Thêm môn học.  3. Quản lý chương theo môn học.  4. Xem danh sách chương.  5. Xóa môn học.  6. Sửa môn học.  - Đối với giảng viên:  1. Tìm kiếm môn học theo tên và mã môn học.  2. Quản lý chương theo môn học.  3. Xem danh sách chương. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm môn học theo tên và mã môn học:**  1. Nhập mã môn hoặc tên môn học vào thanh kiếm tìm.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách môn học dựa trên input từ người dùng.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có môn học nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm môn học:**  1. Chọn chức năng thêm môn học.  2. Điền thông tin môn học, có thể điền thông tin chương và bấm nút thêm.  3. Kiểm tra các ràng buộc dữ liệu.  4. Lưu thông tin môn học vào cơ sở dữ liệu.  5. Thông báo thêm thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi quản lý chương:**  1. Chọn chức năng sửa môn học, quản trị viên có thể thêm chương, xóa chương hoặc cập nhật lại tên chương.  2. Hệ thống sẽ lưu thông tin chương của môn học đó cùng với môn học.  3. Thông báo thao tác thành công.  4. Kết thúc Use Case.  **- Luồng sự kiện chính khi sửa môn học:**  1. Chọn chức năng sửa môn học ở phần thao tác.  2. Hệ thống hiện ra form với các trường dữ liệu được set giá trị mặc định là các thuộc tính của môn học.  2. Thay đổi dữ liệu trường cần sửa và nhấn nút sửa.  3. Hệ thống sẽ kiểm tra các dữ liệu đầu vào.  4. Lưu vào cơ sở dữ liệu.  5. Thông báo sửa thành công.  6.Kết thúc Use Case.  **- Luồng sự kiện chính khi xóa môn học:**  1. Chọn chức năng xem danh sách câu hỏi ở phần thao tác.  2. Hệ thống hiển thị ra danh sách các câu hỏi của đề thi đó.  3. Kết thúc Use Case.  **- Luồng sự kiện chính khi xem danh sách chương:**  1. Chọn chức năng xem danh sách chương ở phần thao tác. Chỉ có thể xem nếu môn học đó có chương.  2. Hệ thống hiện ra danh sách các chương và danh sách câu hỏi tương ứng với các chương đó.  3. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  **- Khi thêm môn học:**  1. Kiểm tra thông tin rỗng ở các trường NOT NULL.  2. Mã môn học đã tồn tại.  3. Số tiết lý thuyết + thực hành không phải là bội số của ba.  **- Khi sửa môn học:**  1. Kiểm tra thông tin rỗng ở các trường NOT NULL.  2. Số tiết lý thuyết + thực hành không phải là bội số của ba.  3. Không tìm thấy môn học.  **- Khi xóa môn học:**  1. Vi phạm ràng buộc dữ liệu. |

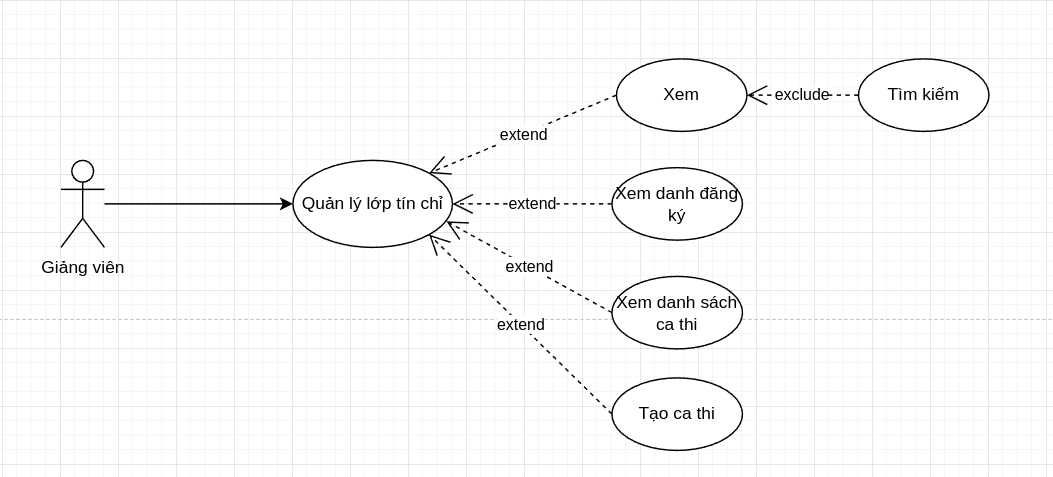
### *Use Case Quản lý ca thi*

Hình 3.1.5: Use Case quản lý ca thi

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên hoặc giảng viên thực hiện các chức năng liên quan đến ca thi. |
| Actor | Quản trị viên, giảng viên. |
| Điều kiện kích hoạt | Quản trị viên hoặc giảng viên chọn chức năng quản lý ca thi. |
| Điều kiện trước | Quản trị viên hoặc giảng viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên” hoặc “Giảng viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách các ca thi (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Tìm kiếm ca thi theo lớp tín chỉ.  2. Thêm ca thi.  3. Xem danh sách thi.  4. Sửa ca thi.  5. Kích hoạt/hủy ca thi. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm ca thi:**  1. Chọn một lớp tín chỉ từ Select.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách ca thi dựa trên tiêu chí lọc của người dùng.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có ca thi nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm ca thi:**  1. Chọn chức năng thêm ca thi. Không thể thêm ca thi cho môn học của lớp tín chỉ không có đề thi.  2. Điền thông tin của ca thi và bấm nút thêm ca thi.  3. Hệ thống kiểm tra các thông tin hợp lệ của ca thi.  4. Lưu thông tin ca thi vào cơ sở dữ liệu.  5. Thông báo thêm thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi sửa ca thi:**  1. Chọn chức năng sửa ca thi ở phần thao tác. Chỉ có thể được sửa ca thi chưa được thi.  2. Hệ thống hiện ra form chỉnh sửa ca thi với các trường được gắn giá trị mặc định là thông tin của ca thi.  3. Thay đổi thông tin ca thi và bấm nút Chỉnh sửa.  4. Thay đổi thông tin ca thi trong cơ sở dữ liệu.  5. Thông báo sửa ca thi thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi kích hoạt/hủy ca thi:**  1. Chọn chức năng kích hoạt/hủy ca thi ở phần thao tác. Chỉ có thể hủy ca thi khi ca thi đó chưa thi.  2. Hệ thống kiểm tra trạng thái hiện tại của ca thi và gửi yêu cầu thay đổi trạng thái ca thi, khác với trạng thái hiện tại của ca thi.  3. Cập nhật cơ sở dữ liệu.  4. Thông báo kích hoạt/huỷ thành công.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi xem danh thi:**  1. Chọn chức năng xem danh sách thi ở phần thao tác.  2. Hệ thống hiển thi ra danh sách các đăng ký của lớp tín chỉ trong ca thi đó.  3. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  **- Khi thêm ca thi:**  1. Kiểm tra thông tin rỗng ở các trường NOT NULL.  2. Ngày thi phải là ngày lớn hơn hiện tại.  3. Thời gian làm bài ít hơn hoặc bằng 120 phút.  4. Không có danh sách đề thi.  **- Khi sửa ca thi:**  1. Kiểm tra thông tin rỗng ở các trường NOT NULL.  2. Ngày thi phải là ngày lớn hơn hiện tại.  3. Thời gian làm bài ít hơn hoặc bằng 120 phút.  4. Không có danh sách đề thi.  5. Không tìm thấy ca thi. |

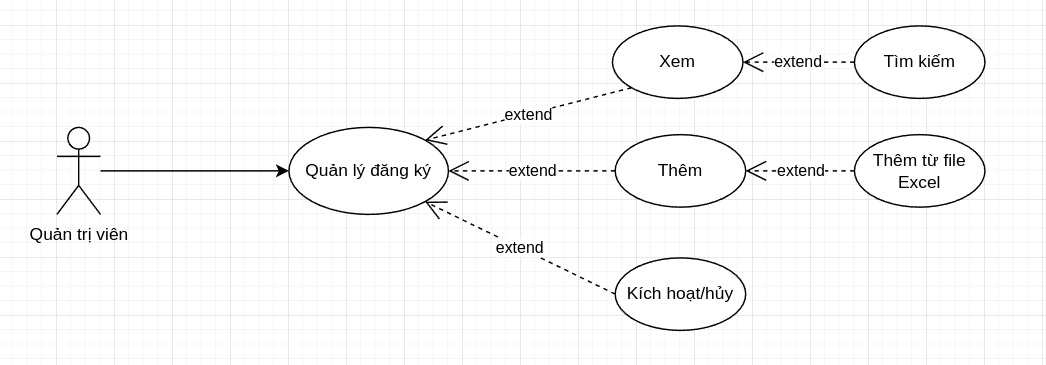
### *Use Case Quản lý lớp tín chỉ*

Hình 3.1.6.1 Use Case Quản lý lớp tín chỉ của quản trị viên

Hình 3.1.6.2 Use Case Quản lý lớp tín chỉ của giảng viên

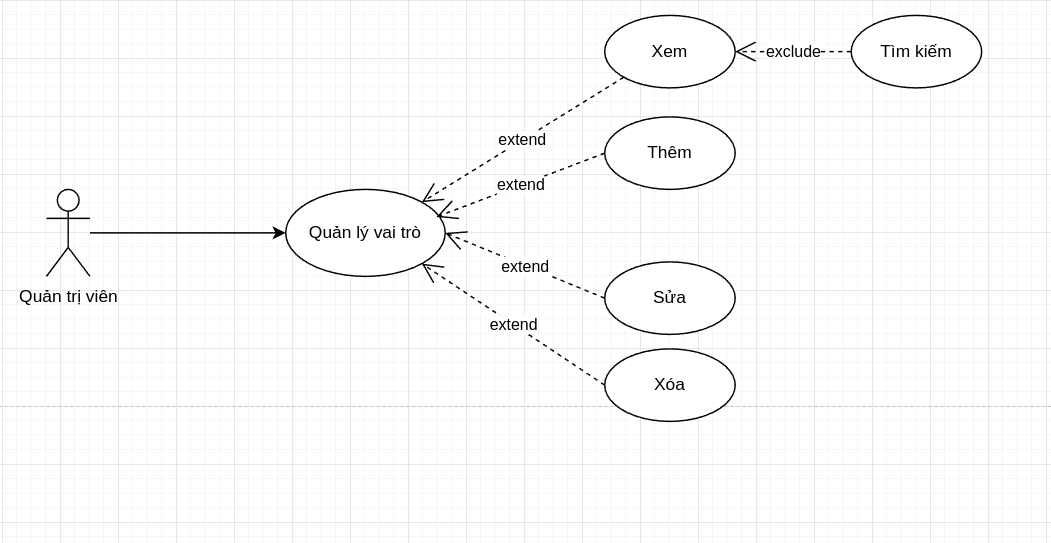
|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên hoặc giảng viên thực hiện các chức năng liên quan đến lớp tín chỉ. |
| Actor | Quản trị viên, giảng viên. |
| Điều kiện kích hoạt | Quản trị viên hoặc giảng viên chọn chức năng quản lý lớp tín chỉ. |
| Điều kiện trước | Quản trị viên hoặc giảng viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên” hoặc “Giảng viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách các lớp tín chỉ (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  - Đối với quản trị viên:  1. Xem, tìm kiếm lớp tín chỉ theo môn học, giảng viên.  2. Thêm lớp tín chỉ.  3. Xem danh sách đăng ký.  4. Xem danh sách ca thi.  5. Sửa lớp tín chỉ.  6. Kích hoạt/hủy lớp tín chỉ.  7. Tạo ca thi cho lớp tín chỉ.  - Đối với giảng viên:  1. Xem, tìm kiếm lớp tín chỉ theo môn học, giảng viên.  2. Xem danh sách đăng ký.  3. Xem danh sách ca thi.  4. Tạo ca thi cho lớp tín chỉ. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm lớp tín chỉ:**  1. Chọn một môn học hoặc giảng viên từ Select.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách lớp tín chỉ dựa trên tiêu chí lọc của người dùng.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có lớp tín chỉ nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm lớp tín chỉ:**  1. Chọn chức năng thêm lớp tín chỉ.  2. Điền thông tin lớp tín chỉ và bấm thêm.  3. Hệ thống kiểm tra các ràng buộc dữ liệu.  4. Lưu thông tin lớp tín chỉ vào cơ sở dữ liệu.  5. Thông báo thêm thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi sửa lớp tín chỉ:**  1. Chọn chức năng sửa lớp tín chỉ ở phần thao tác.  2. Hệ thống hiện ra form sửa lớp tín chỉ và gắn giá trị mặc định của form là các thông tin của lớp tín chỉ được chỉnh sửa.  3. Quản trị viên thay đổi thông tin và bấm nút chỉnh sửa.  4. Hệ thống kiểm tra các ràng buộc dữ liệu.  5. Lưu thông tin lớp tín chỉ vào cơ sở dữ liệu.  6. Thông báo sửa thành công.  7. Kết thúc Use Case.  **- Luồng sự kiện chính khi kích hoạt/hủy lớp tín chỉ:**  1. Chọn chức năng kích hoạt/hủy lớp tín chỉ ở phần thao tác.  2. Hệ thống kiểm tra trạng thái hiện tại của lớp tín chỉ và gửi yêu cầu thay đổi trạng thái lớp tín chỉ, khác với trạng thái hiện tại của lớp tín chỉ.  3. Cập nhật cơ sở dữ liệu.  4. Thông báo kích hoạt/huỷ thành công.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi xem danh sách đăng ký:**  1. Chọn chức năng xem danh sách đăng ký ở phần thao tác.  2. Hệ thống hiển thị ra danh sách các đăng ký của lớp tín chỉ đó.  3. Kết thúc Use Case.  **- Luồng sự kiện chính khi xem danh sách ca thi:**  1. Chọn chức năng xem danh sách ca thi ở phần thao tác.  2. Hệ thống hiển thị ra danh sách các ca thi của lớp tín chỉ đó.  3. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  **- Khi thêm lớp tín chỉ:**  1. Kiểm tra thông tin rỗng ở các trường NOT NULL.  2. Niên khóa bé hơn hiện tại.  3. Giảng viên không có vai trò là giảng viên.  **- Khi sửa lớp tín chỉ:**  1. Kiểm tra thông tin rỗng ở các trường NOT NULL.  2. Niên khóa bé hơn hiện tại.  3. Giảng viên không có vai trò là giảng viên.  4. Không tìm thấy lớp tín chỉ. |

### *Use Case Quản lý đăng ký*

Hình 3.1.7 Use Case Quản lý đăng ký

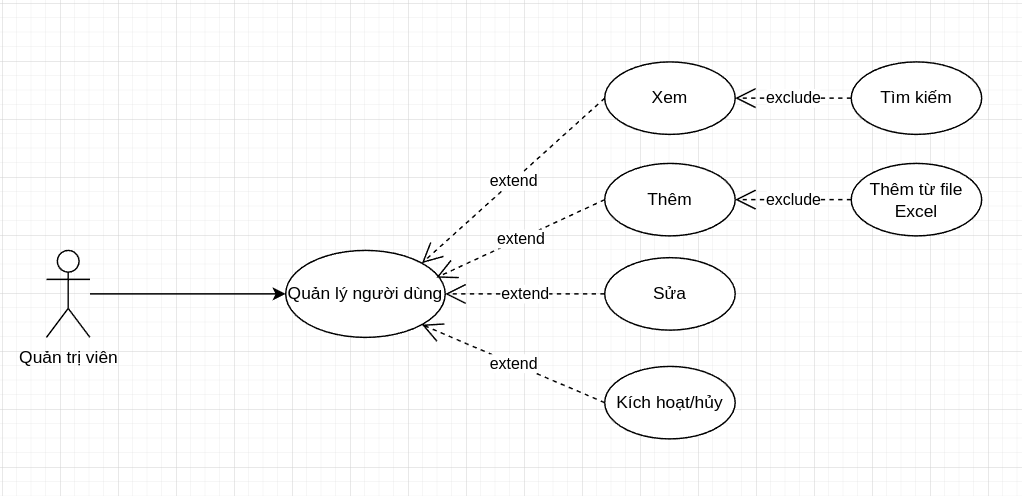
|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên thực hiện các chức năng liên quan đến đăng ký. |
| Actor | Quản trị viên. |
| Điều kiện kích hoạt | Quản trị viên chọn chức năng quản lý đăng ký. |
| Điều kiện trước | Quản trị viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách các đăng ký (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Tìm kiếm đăng ký theo lớp tín chỉ.  2. Thêm đăng ký.  3. Thêm đăng ký từ file Excel.  4. Kích hoạt/hủy đăng ký. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm đăng ký:**  1. Chọn một lớp tín chỉ từ Select.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách lớp tín chỉ dựa trên tiêu chí lọc của người dùng.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có đăng ký nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi kích hoạt/hủy đăng ký:**  1. Chọn chức năng kích hoạt/hủy đăng ký ở phần thao tác.  2. Hệ thống kiểm tra trạng thái hiện tại của đăng ký và gửi yêu cầu thay đổi trạng thái đăng ký, khác với trạng thái hiện tại của đăng ký.  3. Cập nhật cơ sở dữ liệu.  4. Thông báo kích hoạt/huỷ thành công.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm đăng ký.**  1. Chọn chức năng thêm đăng ký.  2. Form thêm đăng ký hiện ra.  3. Chọn thông tin đăng ký và bấm thêm đăng ký  4. Lưu vào cơ sở dữ liệu.  5. Thông báo thêm đăng ký thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm đăng ký từ file Excel:**  1. Chọn chức năng thêm đăng ký từ file Excel.  2. Form chọn file Excel hiển thị ra.  3. Người dùng chọn file Excel với đuôi kết thúc bằng .xlsx và nhấn nút Thêm tất cả.  4. Hệ thống kiểm tra các đăng ký thỏa mãn các ràng buộc dữ liệu như kiểm tra NULL: mã sinh viên, mã môn học, niên khóa.  5. Thêm mới sinh viên, giảng viên, lớp tín chỉ, môn học nếu chưa tồn tại.  6. Thông báo thêm được bao nhiêu trên tổng số đăng ký trong file Excel thành công.  7. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  **- Khi thêm đề thi:**  1. Tên đề thi đã tồn tại.  2. Không có danh sách câu hỏi.  **- Khi xóa đề thi:**  1. Vi phạm ràng buộc dữ liệu. |

### *Use Case Quản lý vai trò*

Hình 3.1.8 Use Case Quản lý vai trò

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên thực hiện các chức năng liên quan đến đề thi. |
| Actor | Quản trị viên. |
| Điều kiện kích hoạt | Quản trị viên chọn chức năng quản lý vai trò. |
| Điều kiện trước | Quản trị viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách các vai trò (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Xem, tìm kiếm vai trò theo mã và tên vai trò.  2. Thêm vai trò.  3. Sửa vai trò.  4. Xóa vai trò. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm vai trò:**  1. Điền mã vai trò hoặc tên vai trò vào thanh tìm kiếm.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách vai trò dựa trên tiêu chí lọc của người dùng.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có vai trò nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm vai trò:**  1. Chọn chức năng thêm vai trò.  2. Điền thông tin vai trò và bấm thêm vai trò.  3. Hệ thống kiểm tra tên vai trò đã tồn tại.  4. Lưu vai trò vào cơ sở dữ liệu.  5. Thông báo thêm thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi xóa vai trò:**  1. Chọn chức năng xóa vai trò ở phần thao tác.  2. Hệ thống thực hiện thao tác xóa vai trò.  3. Thông báo lỗi nếu vai trò không thể xóa.  4. Thông báo xóa thành công nếu không vi phạm ràng buộc dữ liệu.  5. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ thông báo lỗi trong các trường hợp sau:  **- Khi thêm vai trò:**  1. Tên vai trò đã tồn tại.  **- Khi xóa vai trò:**  1. Vi phạm ràng buộc dữ liệu. |

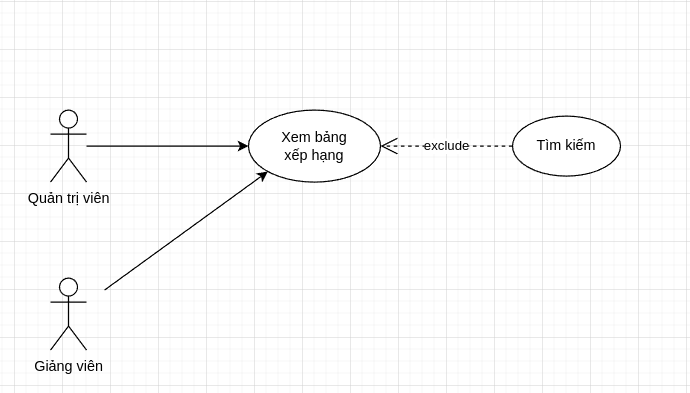
### *Use Case Quản lý người dùng*



Hình 3.1.9 Use Case Quản lý người dùng

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên thực hiện các chức năng liên quan đến người dùng. |
| Actor | Quản trị viên. |
| Điều kiện kích hoạt | Quản trị viên chọn chức năng quản lý người dùng. |
| Điều kiện trước | Quản trị viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách các người dùng (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Tìm kiếm người dùng theo tên và mã người dùng.  2. Thêm người dùng.  3. Sửa người dùng.  4. Kích hoạt/hủy người dùng. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm người dùng:**  1. Điền mã hoặc tên người dùng vào thanh tìm kiếm.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách người dùng dựa trên tiêu chí lọc của quản trị viên.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có người dùng nào.  5. Kết thúc Use Case.  **- Luồng sự kiện chính khi thêm người dùng:**  1. Chọn chức năng thêm người dùng.  2. Điền thông tin người dùng và bấm thêm.  3. Hệ thống kiểm tra dữ liệu.  4. Lưu thông tin vào cơ sở dữ liệu.  5. Thông báo thêm thành công.  6. Kết thúc Use Case.  **- Luồng sự kiện chính khi sửa người dùng:**  1. Chọn chức năng sửa người dùng ở phần thao tác.  2. Hệ thống hiển thị form chỉnh sửa người dùng và set các giá trị mặc định của form là thông tin của người dùng.  3. Thay đổi thông tin cần cập nhật và bấm nút sửa.  4. Hệ thống kiểm tra dữ liệu đầu vào.  5. Lưu thông tin vào cơ sở dữ liệu.  6. Thông báo sửa thành công.  7. Kết thúc Use Case.  **- Luồng sự kiện chính khi kích hoạt/hủy người dùng:**  1. Chọn chức năng kích hoạt/hủy đề thi ở phần thao tác.  2. Hệ thống kiểm tra trạng thái hiện tại của người dùng và gửi yêu cầu thay đổi trạng thái người dùng, khác với trạng thái hiện tại của người dùng.  3. Cập nhật cơ sở dữ liệu.  4. Thông báo kích hoạt/huỷ thành công.  5. Kết thúc Use Case. |
| Luồng sự kiện phụ | Hệ thống sẽ báo lỗi trong các trường hợp sau:  **- Luồng sự kiện khi thêm, sửa người dùng.**  1. Người dùng không tồn tại.  2. Mật khẩu mới không đủ 8 ký tự.  3. Mã đổi mật khẩu không đúng.  4. Đã hết thời gian đổi mật khẩu. |

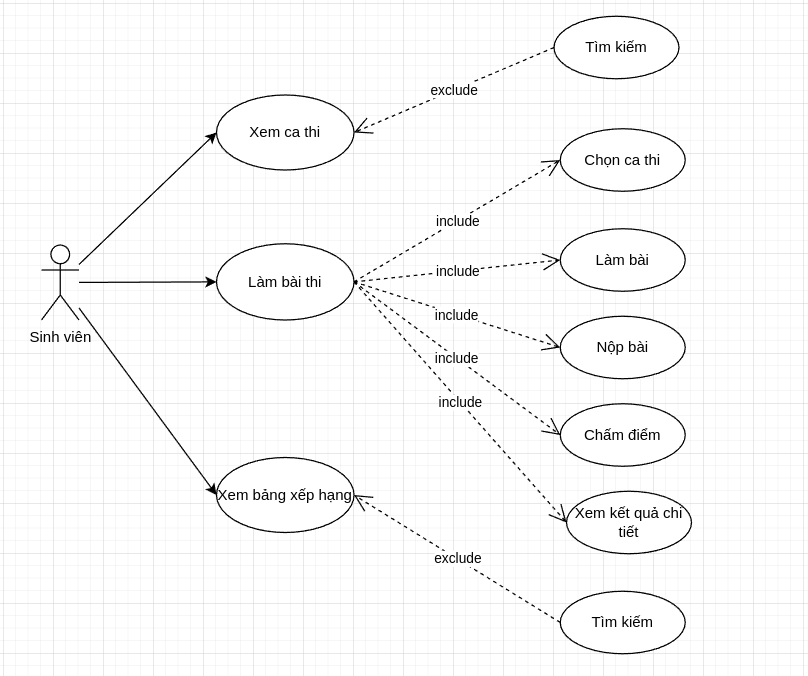
### *Use Case Xem bảng xếp hạng*

****

Hình 3.1.10 Use Case Xem bảng xếp hạng

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Cho phép quản trị viên hoặc giảng viên xem bảng xếp hạng. |
| Actor | Quản trị viên, giảng viên. |
| Điều kiện kích hoạt | Quản trị viên hoặc giảng viên chọn chức năng xem bảng xếp hạng. |
| Điều kiện trước | Quản trị viên hoặc giảng viên đăng nhập vào hệ thống và phải có vai trò là “Quản trị viên” hoặc “Giảng viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách bảng xếp hạng (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Lọc theo niên khóa, học kỳ và lớp tín chỉ. |

### *Use Case của sinh viên*



Hình 3.1.11 Use Case của sinh viên

#### Use Case Xem ca thi

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Use Case chức năng xem ca thi của sinh viên. |
| Actor | Sinh viên. |
| Điều kiện kích hoạt | Sinh chọn chức năng xem ca thi. |
| Điều kiện trước | Sinh viên đăng nhập vào hệ thống và phải có vai trò là “Sinh viên” |
| Điều kiện sau | Hệ thống sẽ tải danh sách các ca thi (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Tìm kiếm ca thi theo lớp tín chỉ. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi tìm kiếm ca thi:**  1. Chọn một niên khóa, học kỳ hoặc loại kỳ thi từ Select.  2. Hệ thống sẽ gửi yêu cầu làm mới danh sách ca thi dựa trên tiêu chí lọc của sinh viên.  3. Nhận về danh sách mới và hiển thị ra danh sách mới.  4. Nếu danh sách rỗng, hệ thống sẽ báo là không có ca thi nào.  5. Kết thúc Use Case. |

#### Use Case làm bài thi

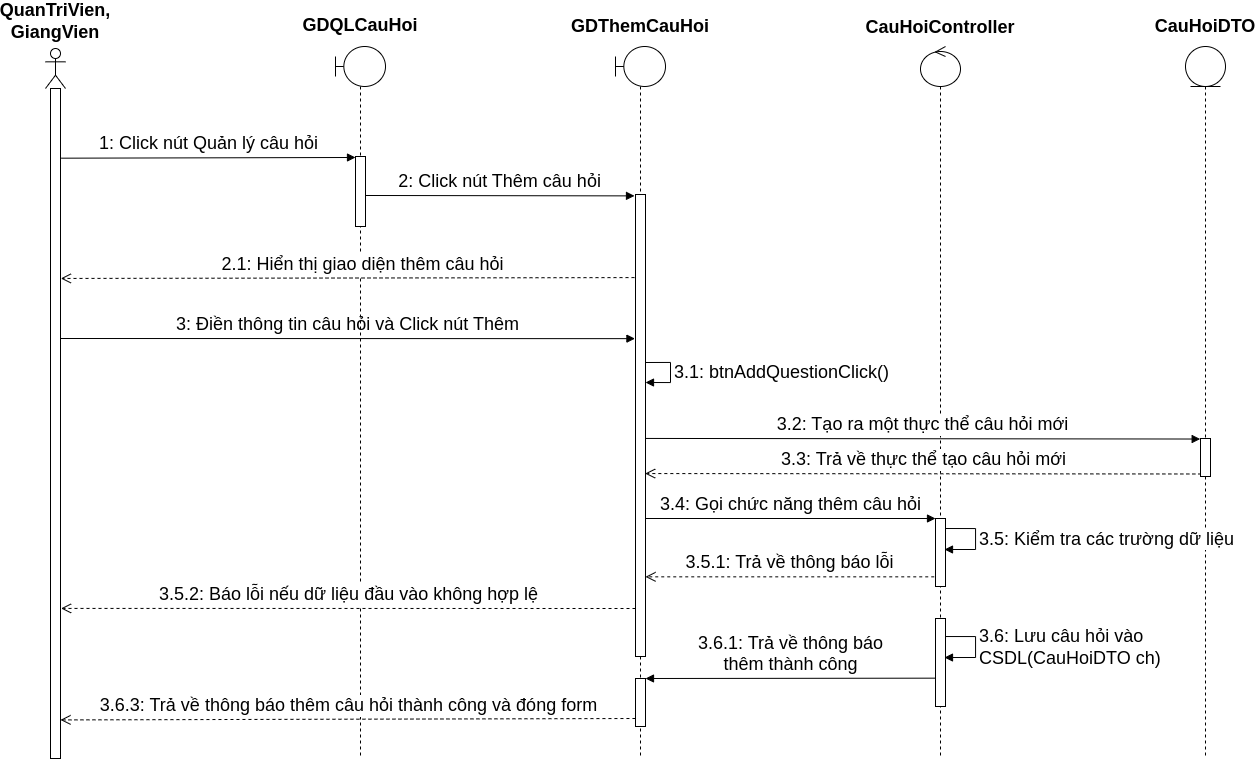
|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Use Case chức năng làm bài thi của sinh viên. |
| Actor | Sinh viên. |
| Điều kiện kích hoạt | Sinh chọn chức năng làm bài thi. |
| Điều kiện trước | Sinh viên đăng nhập vào hệ thống và phải có vai trò là “Sinh viên”. Sinh viên có ca thi chưa thi và thời gian của ca thi phải lớn hơn hiện tại. |
| Điều kiện sau | Hệ thống cho phép sinh viên làm bài thi. |
| Luồng sự kiện chính | **- Luồng sự kiện chính khi làm bài thi:**  1. Chọn 1 ca thi từ danh sách các ca thi.  2. Bấm nút làm bài để lấy danh sách các câu hỏi từ đề thi đã được gán sẵn trong danh sách các đề thi của ca thi.  3. Sinh viên làm bài thi.  4. Sinh viên nhấn nút nộp bài thi hoặc hết thời gian làm bài hệ thống sẽ tự kích hoạt chức năng nộp bài cho sinh viên.  5. Hệ thống chấm điểm bài thi và trả về kết quả thi.  6. Sinh viên có thể xem chi tiết bài thi.  7. Kết thúc Use Case. |

#### Use Case Xem bảng xếp hạng

|  |  |
| --- | --- |
| **Use Case** | **Nội dung** |
| Mô tả | Use Case chức năng xem bảng xếp hạng. |
| Actor | Sinh viên. |
| Điều kiện kích hoạt | Sinh chọn chức năng xem bảng xếp hạng. |
| Điều kiện trước | Sinh viên đăng nhập vào hệ thống và phải có vai trò là “Sinh viên”. |
| Điều kiện sau | Hệ thống sẽ tải danh sách bảng xếp hạng (phân trang), hiển thị ra màn hình và cho phép thực hiện các thao tác như sau:  1. Lọc theo niên khóa, học kỳ và lớp tín chỉ. |

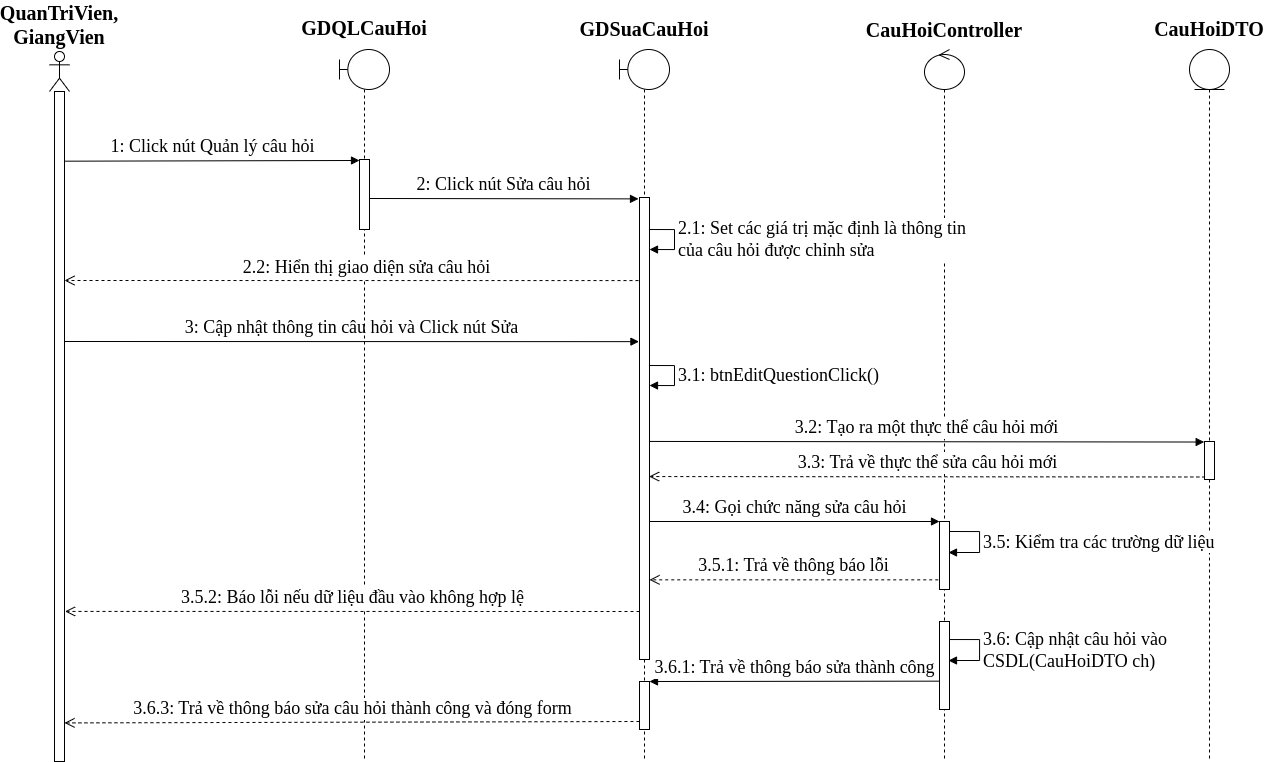
1. **Sơ đồ tuần tự**
   1. *Sơ đồ tuần tự chức năng thêm*

Mô hình tuần tự chức năng thêm tượng trưng cho chức năng thêm của tất cả module khác.

Hình 3.2.1 Sơ đồ tuần tự chức năng thêm

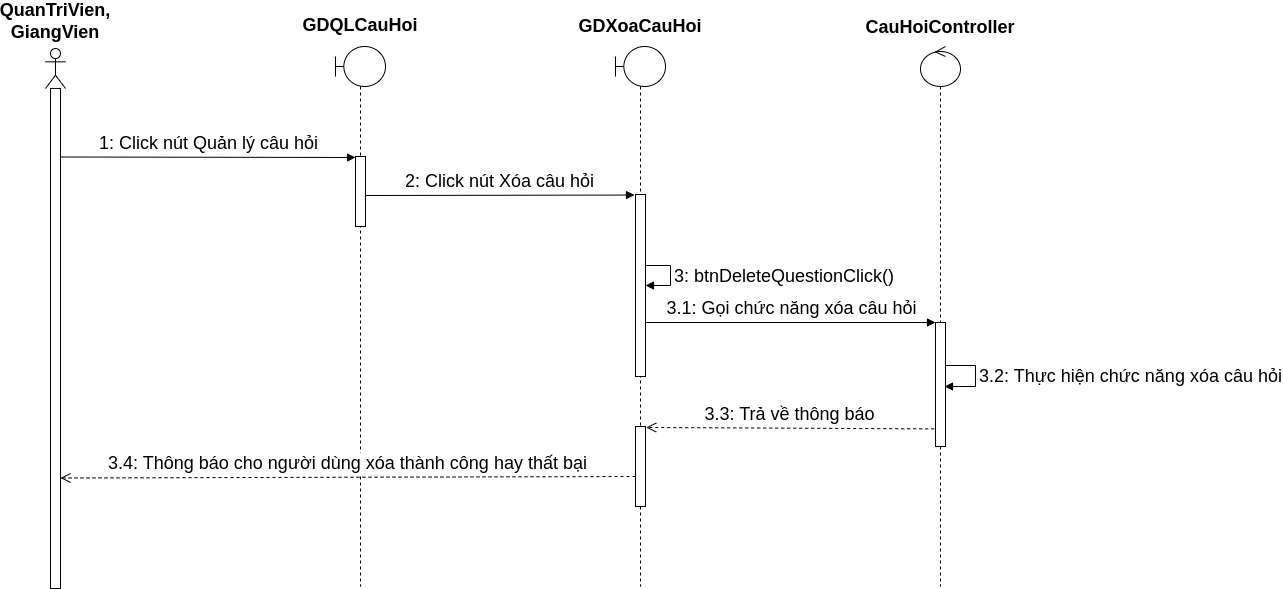
* 1. *Sơ đồ tuần tự chức năng sửa*

Mô hình tuần tự chức năng sửa tượng trưng cho chức năng thêm của tất cả module khác.

Hình 3.2.2 Sơ đồ tuần tự chức năng sửa

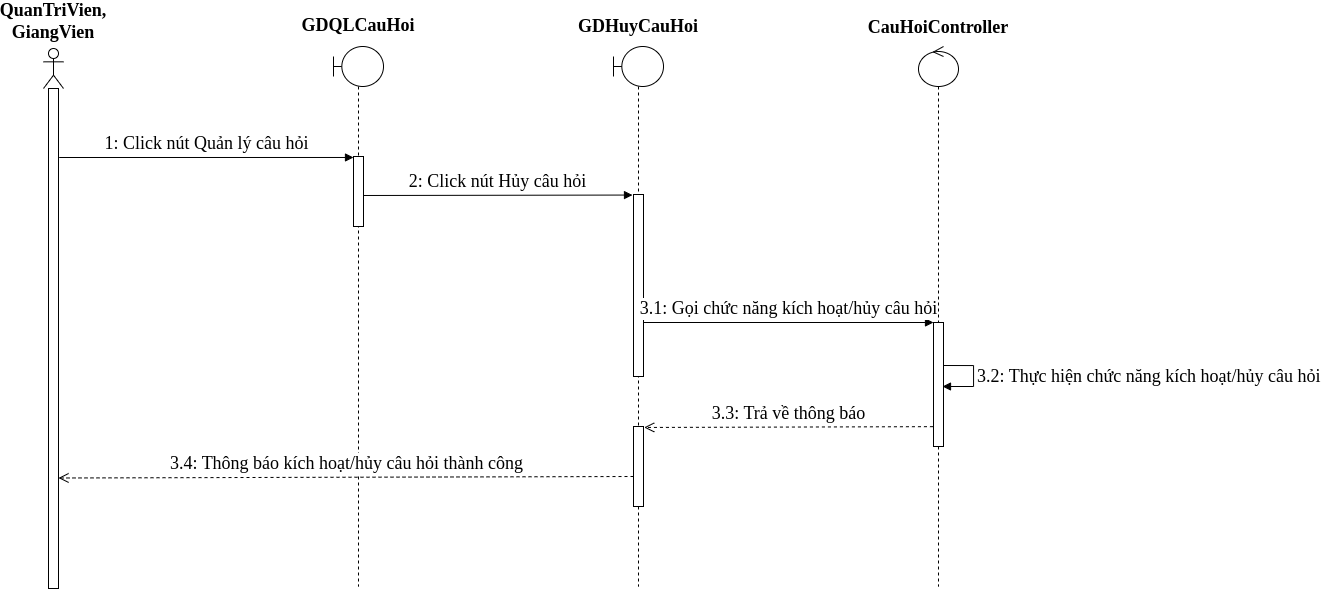
* 1. *Sơ đồ tuần tự chức năng xóa*

Mô hình tuần tự chức năng xóa tượng trưng cho chức năng xóa của tất cả module khác.

Hình 3.2.3 Sơ đồ tuần tự chức năng xóa

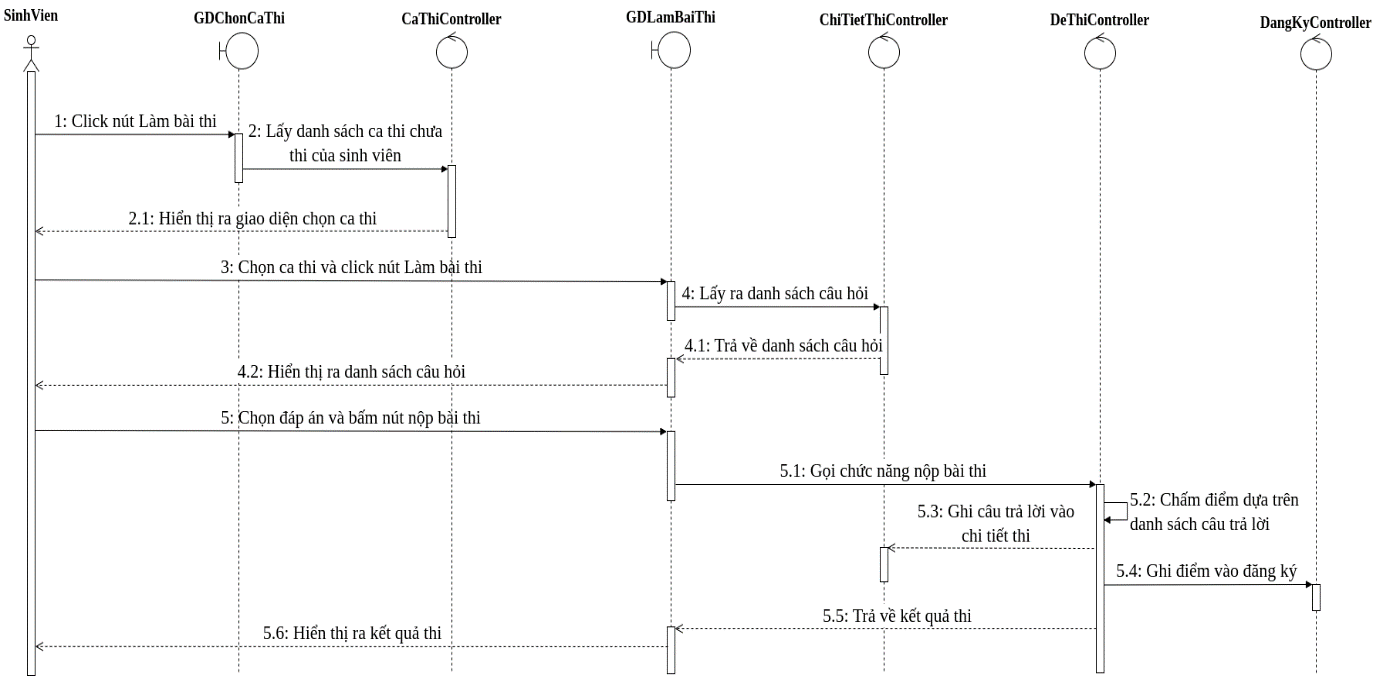
* 1. *Sơ đồ tuần tự chức năng hủy*

Mô hình tuần tự chức năng hủy tượng trưng cho chức năng hủy của tất cả module khác.

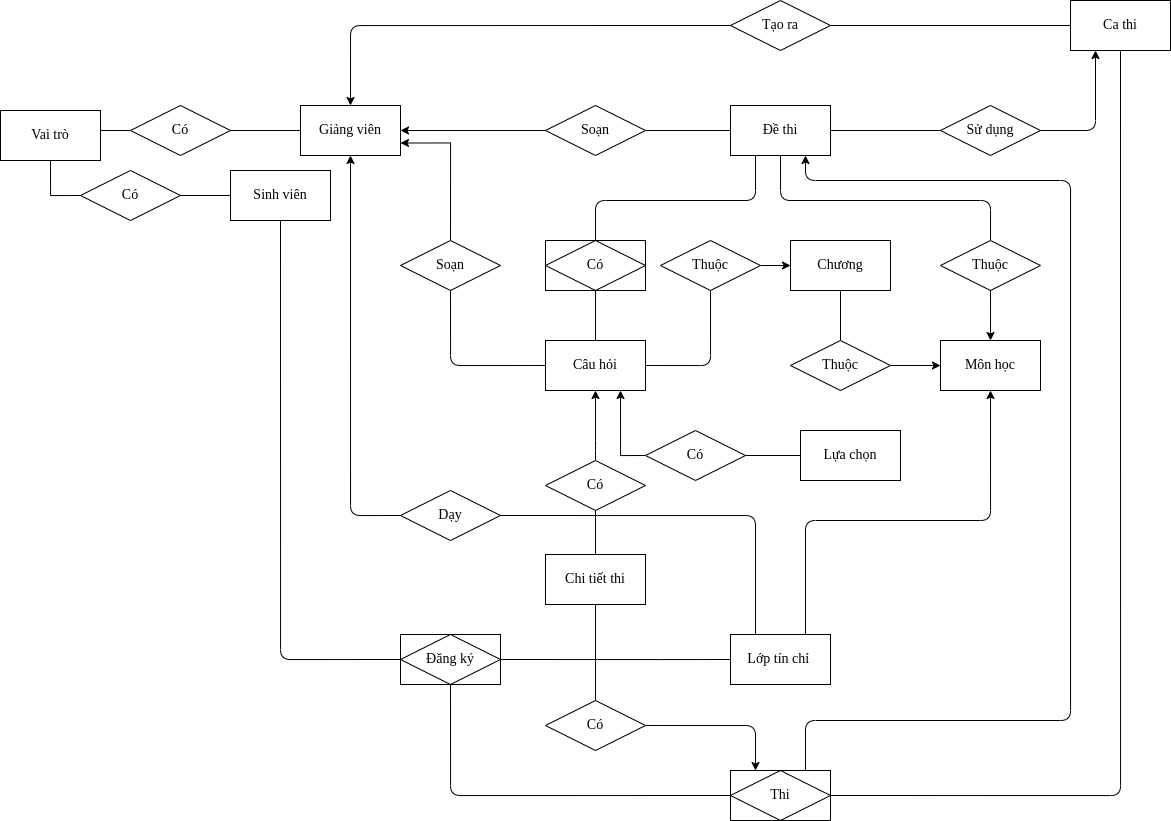
****

Hình 3.2.4 Sơ đồ tuần tự chức năng hủy

* 1. *Sơ đồ tuần tự chức năng làm bài thi*

Hình 3.2.5 Sơ đồ tuần tự chức năng làm bài thi

1. **Thiết kế sơ đồ quan hệ thực thể (ERD)**



*Hình 3.3 Sơ đồ Quan hệ thực thể (ERD)*

1. **Thiết kế chi tiết thực thể**

**− cathi** (macathi, ngaythi, tencathi, tietbaodanh, dahuy, dathi, thoigianlambai, loaikythi, magv).

− **cauhoi** (macauhoi, noidungcauhoi, hinhanh, dokho, loaicauhoi, machuong, magv, consudung).

− **chitietthi** (id, cautraloi, macauhoi, macathi, maltc, masv, lanthi).

− **chuong** (machuong, sochuong, tenchuong, mamh).

− **dangky** (masv, maltc, diemcc, diemck, diemgk, huydangky).

− **dethi** (madethi, taoluc, tendethi, capnhatluc, dasudung, mamh, magv, consudung).

− **dethi\_cauhoi** (madethi, macauhoi).

− **loptinchi** (maltc, nhom, nienkhoa, hocky, huylop, mamh, magv).

− **luachon** (id, noidung, ladapan, thutu, macauhoi).

− **monhoc** (mamh, tenmh, sotiet\_th, sotiet\_lt).

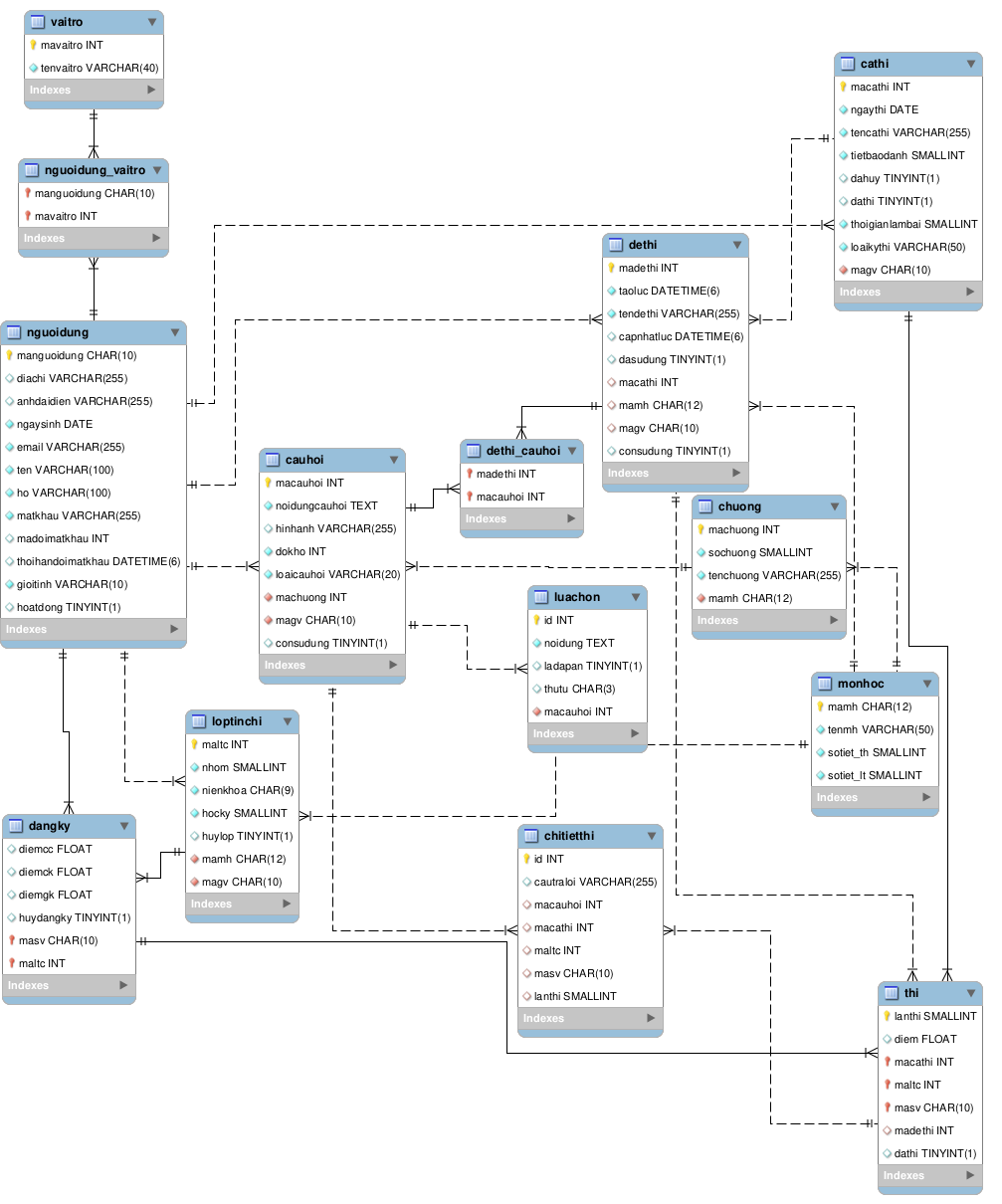
− **nguoidung** (manguoidung, diachi, anhdaidien, ngaysinh, email, ten, ho, matkhau, madoimatkhau, thoihandoimatkhau, gioitinh, hoatdong).

− **nguoidung\_vaitro** (manguoidung, mavaitro).

− **thi** (macathi, maltc, masv, lanthi, diem, madethi, dathi).

− **vaitro** (mavaitro, tenvaitro).

1. **Thiết kế sơ đồ lớp cơ sở dữ liệu (Class Diagrams)**



*Hình 3.5 Sơ đồ Lớp cơ sở dữ liệu (Class Diagrams)*

1. **Từ điển dữ liệu**

* Bảng: **cauhoi**(macauhoi, noidungcauhoi, hinhanh, dokho, loaicauhoi, machuong, magv, consudung).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | macauhoi | int | primary key, not null, ai | Mã câu hỏi |
| **2** | noidungcauhoi | text | not null | Nội dung câu hỏi |
| **3** | dokho | tinyint(2) | not null | Độ khó |
| **4** | loaicauhoi | varchar(20) | not null | Loại câu hỏi |
| **5** | machuong | int | foreign key, not null | Mã chương |
| **6** | magv | char(10) | foreign key, not null | Mã giảng viên |
| **7** | consudung | tinyint(1) | default(1) | Còn sử dụng |
| **8** | hinhanh | varchar(255) |  | Hình ảnh |

**Nhiệm vụ:** Dùng để lưu thông tin câu hỏi

**Mô tả:** Chứa các thông tin của câu hỏi

* Bảng**: luachon**(id, noidung, ladapan, thutu, macauhoi).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | id | int | primary key, not null, ai | Mã lựa chọn |
| **2** | noidung | text | not null | Nội dung |
| **3** | ladapan | tinyint(1) | default(0) | Là đáp án |
| **4** | thutu | char(3) | not null | Thứ tự |
| **5** | macauhoi | int | foreign key, not null | Mã câu hỏi |

**Nhiệm vụ:** Dùng để lưu trữ thông tin lựa chọn đáp án

**Mô tả:** Chứa các thông tin của việc lựa chọn đáp án

* Bảng**: chuong**(machuong, sochuong, tenchuong, mamh).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | machuong | int | primary key, not null, ai | Mã chương |
| **2** | sochuong | smallint | not null | Số chương |
| **3** | tenchuong | varchar(255) | not null, unique | Tên chương |
| **4** | mamh | char(12) | foreign key, not null | Mã môn học |

**Nhiệm vụ:** Dùng để lưu thông tin của chương (trong nội dung môn học)

**Mô tả:** Chứa các thông tin của chương đó

* Bảng**: monhoc**(mamh, tenmh, sotiet\_th, sotiet\_lt).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | mamh | char(12) | primary key, not null | Mã môn học |
| **2** | tenmh | varchar(50) | not null | Tên môn học |
| **3** | sotiet\_th | smallint | not null | Số tiết thực hành |
| **4** | sotiet\_lt | smallint | not null | Số tiết lý thuyết |

**Nhiệm vụ:** Dùng để lưu thông tin của môn học

**Mô tả:** Chứa các thông tin của môn học

* Bảng**: dethi**(madethi, taoluc, tendethi, capnhatluc, dasudung, mamh, magv, consudung).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | madethi | int | primary key, not null, ai | Mã đề thi |
| **2** | taoluc | datetime(6) | not null | Tạo lúc |
| **3** | tendethi | varchar(255) | not null, unique | Tên đề thi |
| **4** | capnhatluc | datetime(6) |  | Cập nhật lúc |
| **5** | dasudung | tinyint(1) | default(0) | Đã sử dụng |
| **6** | mamh | char(12) | foreign key, not null | Mã môn học |
| **7** | magv | char(10) | foreign key, not null | Mã giảng viên |

**Nhiệm vụ:** Dùng để lưu thông tin của đề thi

**Mô tả:** Chứa các thông tin của đề thi

* Bảng**: nguoidung**(manguoidung, diachi, anhdaidien, ngaysinh, email, ten, ho, matkhau, madoimatkhau, thoihandoimatkhau, gioitinh, hoatdong).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | manguoidung | char(10) | primary key, not null | Mã người dùng |
| **2** | diachi | varchar(255) |  | Địa chỉ |
| **3** | anhdaidien | varchar(255) |  | Ảnh đại diện |
| **4** | ngaysinh | date | not null | Ngày sinh |
| **5** | email | varchar(255) | not null, unique | Email |
| **6** | ho | varchar(100) | not null | Họ |
| **7** | ten | varchar(100) | not null | Tên |
| **8** | matkhau | varchar(255) | not null | Mật khẩu |
| **9** | madoimatkhau | int |  | Mã đổi mật khẩu |
| **10** | thoihandoimatkhau | datetime(6) |  | Thời hạn đổi mật khẩu |
| **11** | gioitinh | varchar(10) | not null | Giới tính |
| **12** | hoatdong | tinyint(1) | default(1) | Hoạt động |

**Nhiệm vụ:** Dùng để lưu thông tin của người dùng

**Mô tả:** Chứa các thông tin của người dùng

* Bảng**: cathi**(macathi, ngaythi, tencathi, tietbaodanh, dahuy, dathi, thoigianlambai, loaikythi, magv).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | macathi | int | primary key, not null, ai | Mã ca thi |
| **2** | ngaythi | date | not null | Ngày thi |
| **3** | tencathi | varchar(255) | not null | Tên ca thi |
| **4** | tietbaodanh | smallint | not null | Tiết báo danh |
| **5** | dahuy | tinyint(1) | default(0) | Đã huỷ |
| **6** | dathi | tinyint(1) | default(0) | Đã thi |
| **7** | thoigianlambai | smallint | not null | Thời gian làm bài |
| **8** | loaikythi | varchar(50) | not null | Loại kỳ thi |
| **9** | magv | char(10) | foreign key, not null | Mã giảng viên |

**Nhiệm vụ:** Dùng để lưu thông tin của ca thi

**Mô tả:** Chứa các thông tin của ca thi

* Bảng**: loptinchi**(maltc, nhom, nienkhoa, hocky, huylop, mamh, magv).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | maltc | int | primary key, not null, ai | Mã lớp tín chỉ |
| **2** | nhom | smallint | not null | Nhóm |
| **3** | nienkhoa | char(9) | not null | Niên khoá |
| **4** | hocky | smallint | not null | Học kỳ |
| **5** | huylop | tinyint(1) | default(0) | Huỷ lớp |
| **6** | mamh | char(12) | foreign key, not null | Mã môn học |
| **7** | magv | char(10) | foreign key, not null | Mã giảng viên |

**Nhiệm vụ:** Dùng để lưu thông tin của lớp tín chỉ

**Mô tả:** Chứa các thông tin của lớp tín chỉ

* Bảng**: dethi\_cauhoi**(madethi, macauhoi).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | madethi | int | primary key, not null | Mã đề thi |
| **2** | macauhoi | int | primary key, not null | Mã câu hỏi |

**Nhiệm vụ:** Dùng để tham chiếu các thông tin của đề thi và câu hỏi

* Bảng**: dangky**(masv, maltc, diemcc, diemgk, diemck, huydangky).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | maltc | int | primary key, not null | Mã lớp tín chỉ |
| **2** | masv | int | primary key, not null | Mã sinh viên |
| **3** | diemcc | float |  | Điểm chuyên cần |
| **4** | diemgk | float |  | Điểm giữa kỳ |
| **5** | diemck | float |  | Điểm cuối kỳ |
| **6** | huydangky | tinyint(1) | default(0) | Huỷ đăng ký |

**Nhiệm vụ:** Dùng để lưu thông tin đăng ký

**Mô tả:** Chứa các thông tin đăng ký

* Bảng**: thi**(macathi, maltc, masv, lanthi, diem, madethi, dathi)**.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | macathi | int | primary key, not null | Mã ca thi |
| **2** | maltc | int | primary key, not null | Mã lớp tín chỉ |
| **3** | masv | char(10) | primary key, not null | Mã sinh viên |
| **4** | lanthi | smallint | primary key, not null | Lần thi |
| **5** | diem | float |  | Điểm |
| **6** | madethi | int | default(0) | Mã đề thi |

**Nhiệm vụ:** Dùng để lưu thông tin thi

**Mô tả:** Chứa các thông tin thi

* Bảng**: chitietthi**(id, cautraloi, macauhoi, macathi, maltc, masv, lanthi).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | id | int | primary key, not null, ai | Mã chi tiết thi |
| **2** | cautraloi | varchar(255) |  | Câu trả lời |
| **3** | macauhoi | int |  | Mã câu hỏi |
| **4** | macathi | int | foreign key, not null | Mã ca thi |
| **5** | maltc | int | foreign key, not null | Mã lớp tín chỉ |
| **6** | masv | char(10) | foreign key, not null | Mã sinh viên |
| **7** | lanthi | smallint | foreign key, not null | Lần thi |

**Nhiệm vụ:** Dùng để lưu thông tin chi tiết thi

**Mô tả:** Chứa các thông tin chi tiết thi

* Bảng**: vaitro**(mavaitro, tenvaitro).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | mavaitro | int | primary key, not null | Mã vai trò |
| **2** | tenvaitro | varchar(40) | not null, unique | Tên vai trò |

**Nhiệm vụ:** Dùng để lưu trữ vai trò người dùng

* Bảng**: nguoidung\_vaitro**(manguoidung, mavaitro).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Ràng buộc dữ liệu** | **Mô tả** |
| **1** | manguoidung | int | primary key, not null | Mã người dùng |
| **2** | mavaitro | varchar(40) | not null, unique | Mã vai trò |

**Nhiệm vụ:** Dùng để tham chiếu thông tin người dùng

**CHƯƠNG IV: PHÁT TRIỂN ỨNG DỤNG**

1. **Đăng nhập**



*Hình 4.1 Giao diện Đăng nhập*

***POST: /api/auth/login***

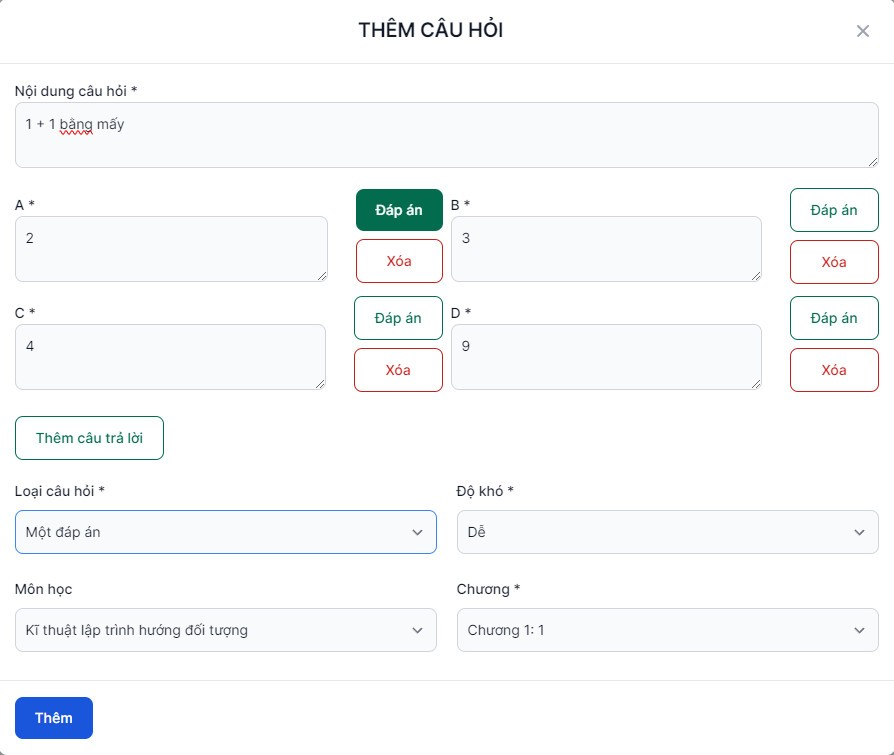
***Success Response:***

* Code: 200 OK
* Body: Trả về thông tin người dùng.

***Error Response:***

*Code 400 Bad Request – Tài khoản hoặc mật khẩu không đúng.*

1. **Quản lý câu hỏi**



*Hình 4.2.1 Giao diện Thêm câu hỏi*

***POST: /api/questions/save***

***Success Response:***

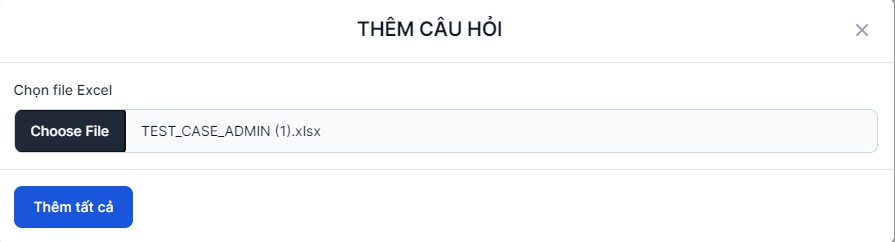
* Code: 200 OK
* Body: <String>: Trả về thông báo thêm câu hỏi thành công

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Nội dung câu hỏi đã tồn tại.*

*Code 400 Bad Request – Không tìm thấy chương.*



*Hình 4.2.2 Giao diện Thêm câu hỏi từ file Excel*

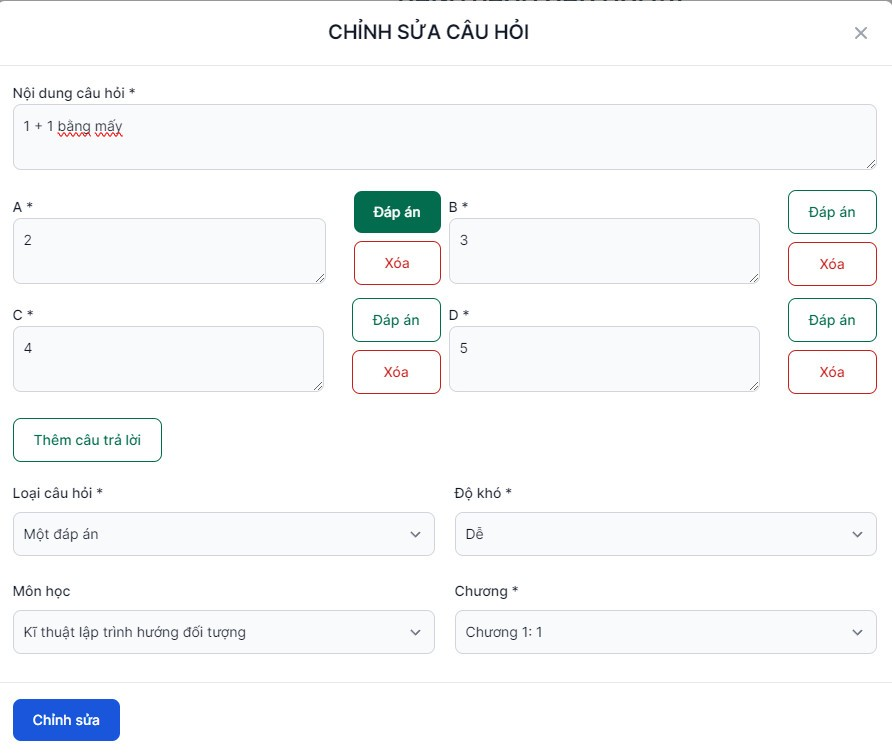
***POST: /api/questions/save/multiple***

***Success Response:***

* Code: 200 OK
* Body: <String>: Trả về thông báo thêm câu hỏi thành công.

***Error Response:***

*Code 400 Bad Request – Không thể đọc được nội dung từ file Excel.*



*Hình 4.2.3 Giao diện Chỉnh sửa câu hỏi*

***POST: /api/questions/save?isEdit=true***

***Success Response:***

* Code: 200 OK
* Body: <String>: Trả về thông báo thêm câu hỏi thành công

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Nội dung câu hỏi đã tồn tại.*

*Code 400 Bad Request – Không tìm thấy chương.*

1. **Quản lý đề thi**



*Hình 4.3.1 Giao diện Thêm đề thi*

***POST: /api/tests/save***

***Success Response:***

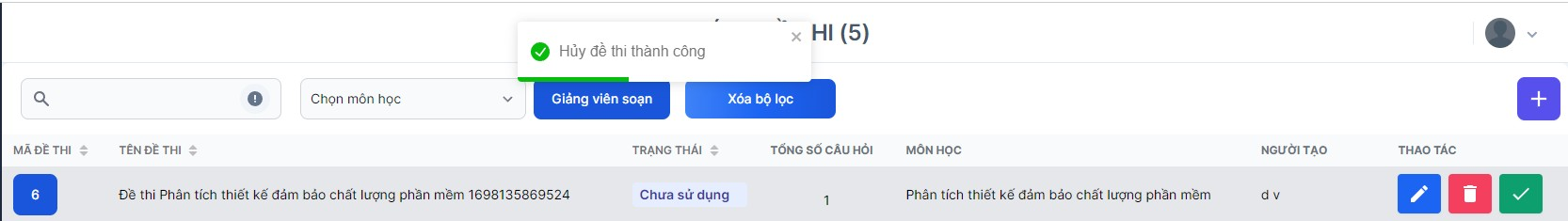
* Code: 200 OK
* Body: <String>: Trả về thông báo thêm đề thi thành công.

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Tên đề thi đã tồn tại.*

*Code 400 Bad Request – Không tìm thấy môn học.*



*Hình 4.3.2 Giao diện Huỷ/ Kích hoạt đề thi*

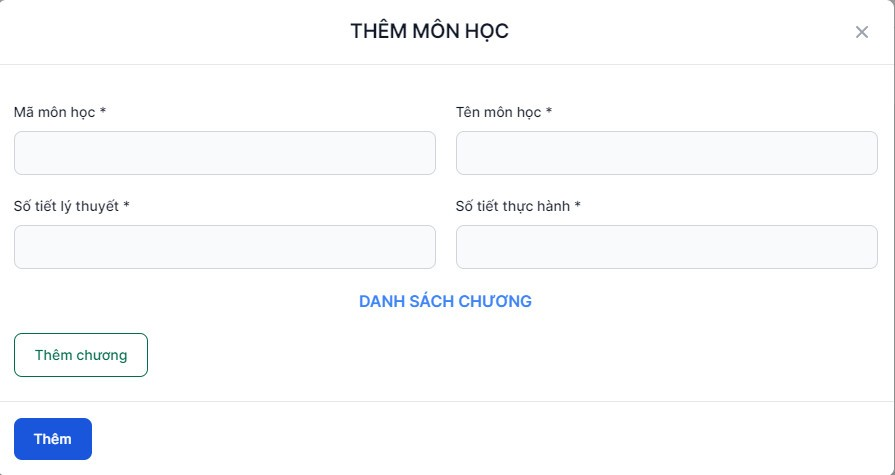
***PUT: /api/tests/${id}?action=${action}***

*Ví dụ: /api/tests/10?action=disable.*

***Success Response:***

* Code: 200 OK
* Body: <String> Trả về thông tin kích hoạt/hủy đề thi thành công.

1. **Quản lý môn học**



*Hình 4.4.1 Giao diện Thêm môn học*

***POST: /api/subjects/save***

***Success Response:***

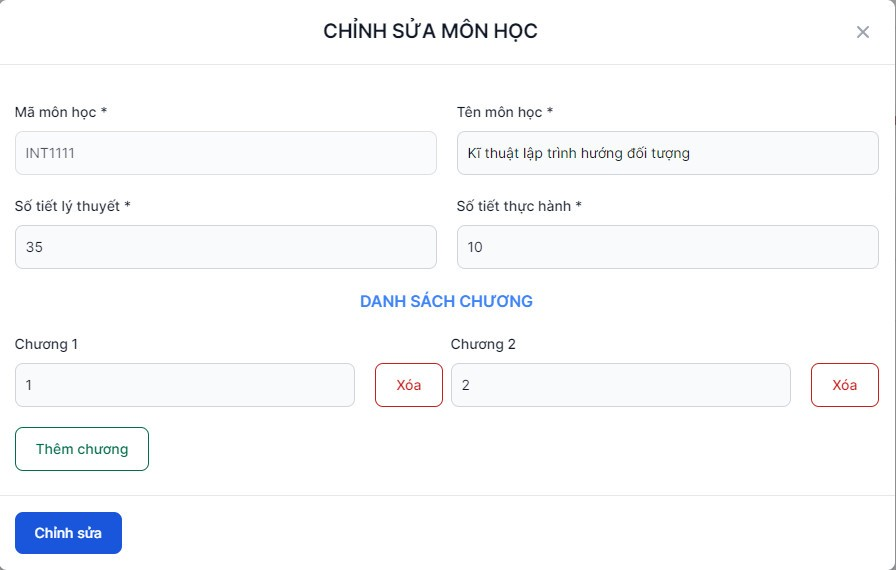
* Code: 200 OK
* Body: <String>: Trả về thông báo thêm môn học thành công

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Mã môn học đã tồn tại.*

*Code 400 Bad Request – Số tiết lý thuyết + thực hành không phải bội số của 3.*



*Hình 4.4.2 Giao diện Chỉnh sửa môn học*

***POST: /api/subjects/save?isEdit=true***

***Success Response:***

* Code: 200 OK
* Body: <String>: Trả về thông báo sửa môn học thành công

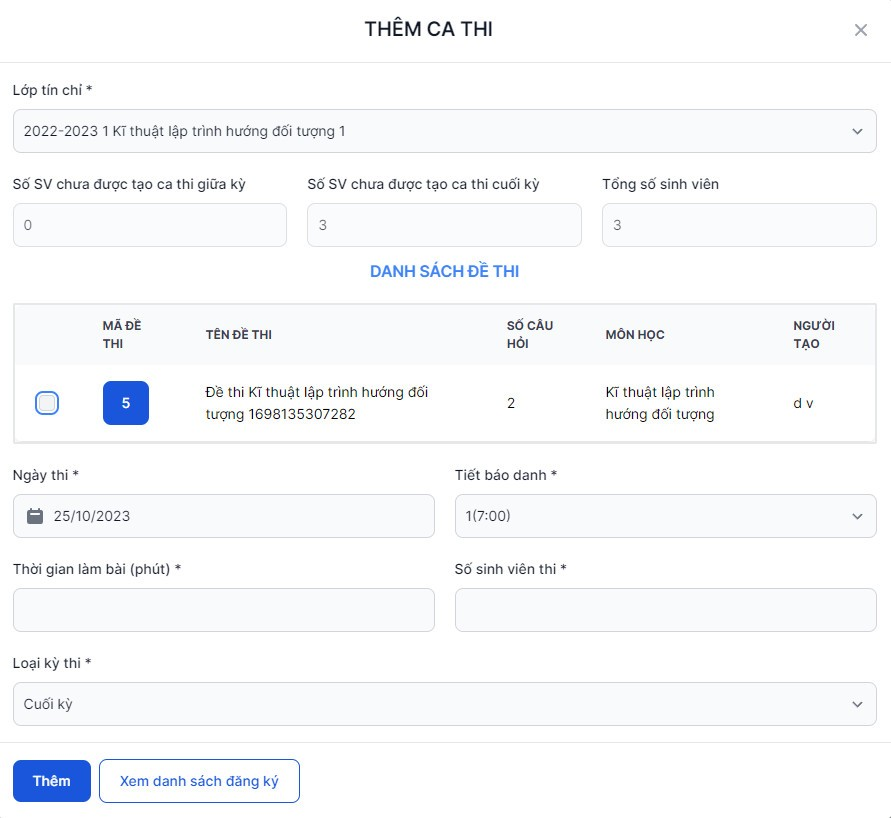
***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Số tiết lý thuyết + thực hành không phải bội số của 3.*

*Code 400 Bad Request – Không tìm thấy môn học.*

1. **Quản lý ca thi**



*Hình 4.5.1 Giao diện Thêm ca thi*

***POST: /api/exams/save***

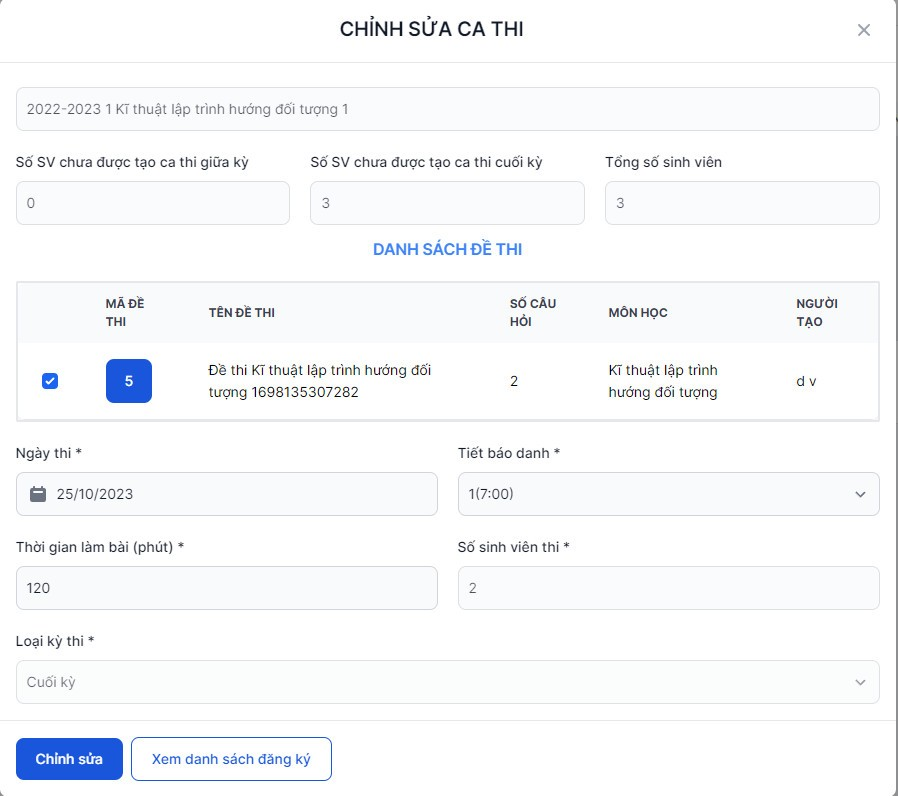
***Success Response:***

* Code: 200 OK
* Body: <String>: Trả về thông báo tạo ca thi thành công

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Không tìm thấy lớp tín chỉ.*



*Hình 4.5.2 Giao diện Chỉnh sửa ca thi*

***POST: /api/tests/save?isEdit=true***

***Success Response:***

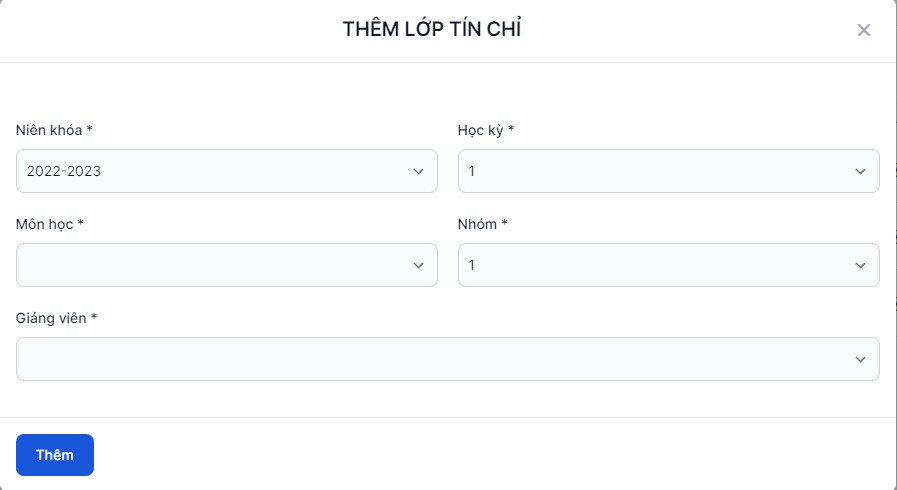
* Code: 200 OK
* Body: <String>: Trả về thông báo sửa ca thi thành công

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Không tìm thấy lớp tín chỉ.*

1. **Quản lý lớp tín chỉ**



*Hình 4.6.1 Giao diện Thêm lớp tín chỉ*

***POST: /api/creditClasses/save***

***Success Response:***

* Code: 200 OK
* Body: <String>: Trả về thông báo thêm lớp tín chỉ thành công.

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Niên khóa + học kỳ + môn học + nhóm phải là duy nhất.*

*Code 400 Bad Request – Không tìm thấy môn học.*

*Code 400 Bad Request – Không tìm thấy giảng viên.*



*Hình 4.6.2 Giao diện Chỉnh sửa lớp tín chỉ*

***POST: /api/creditClasses/save?isEdit=true***

***Success Response:***

* Code: 200 OK
* Body: <String>: Trả về thông báo sửa lớp tín chỉ thành công.

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

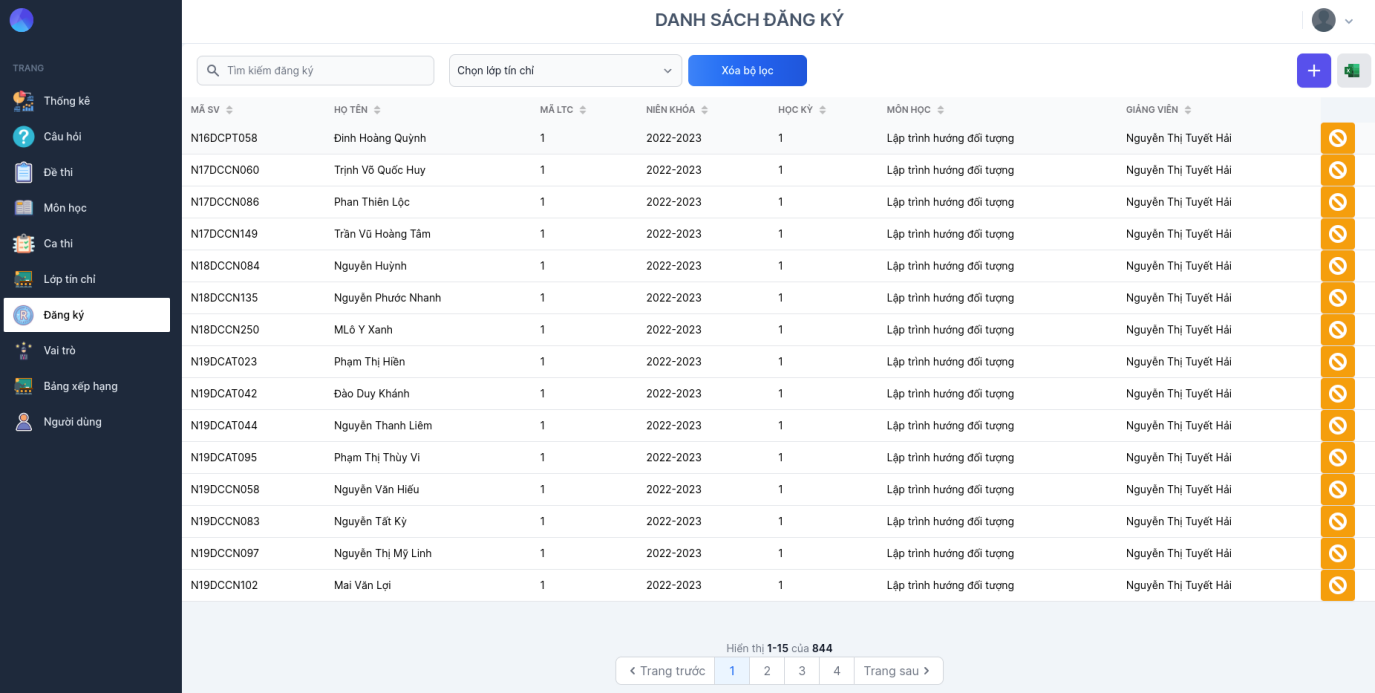
*Code 400 Bad Request – Niên khóa + học kỳ + môn học + nhóm phải là duy nhất.*

*Code 400 Bad Request – Không tìm thấy môn học.*

*Code 400 Bad Request – Không tìm thấy giảng viên.*

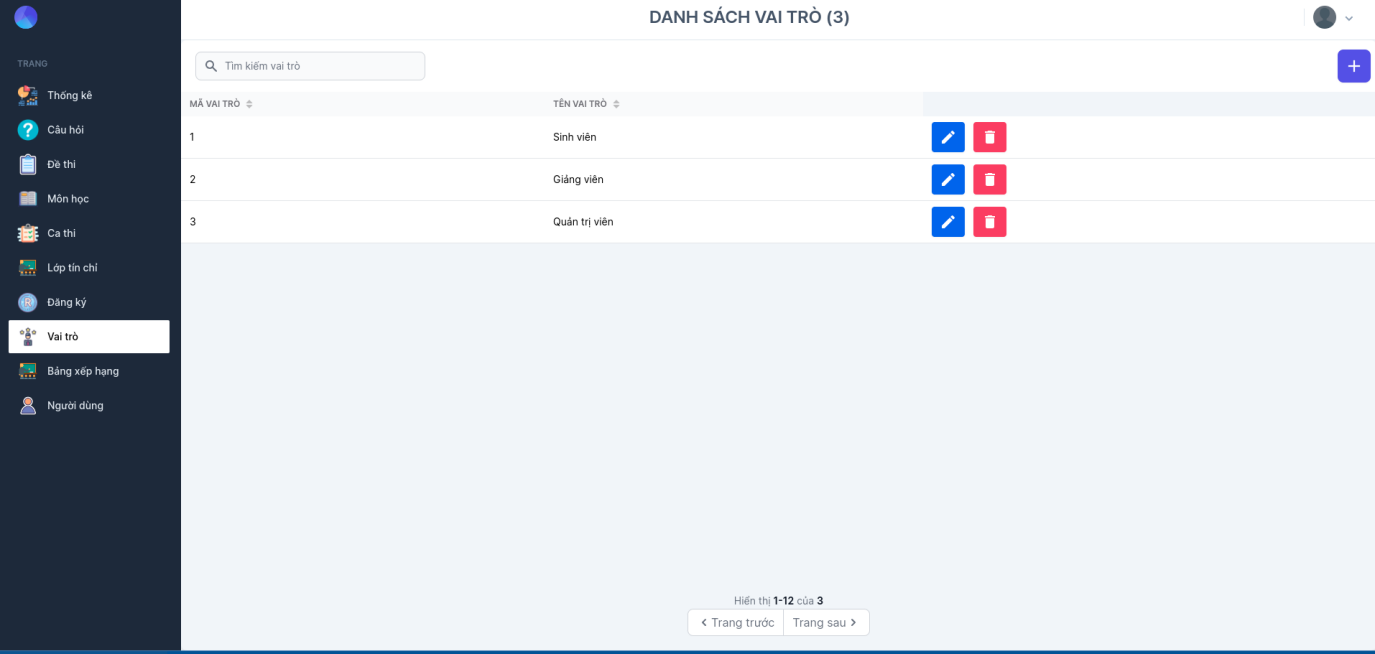
*Code 400 Bad Request – Không tìm thấy lớp tín chỉ.*

1. **Quản lý đăng ký**



*Hình 4.7 Giao diện Quản lý đăng ký*

1. **Quản lý vai trò**

 Hình 4.8 Giao diện quản lý vai trò

***POST: /api/admin/roles/save***

***Success Response:***

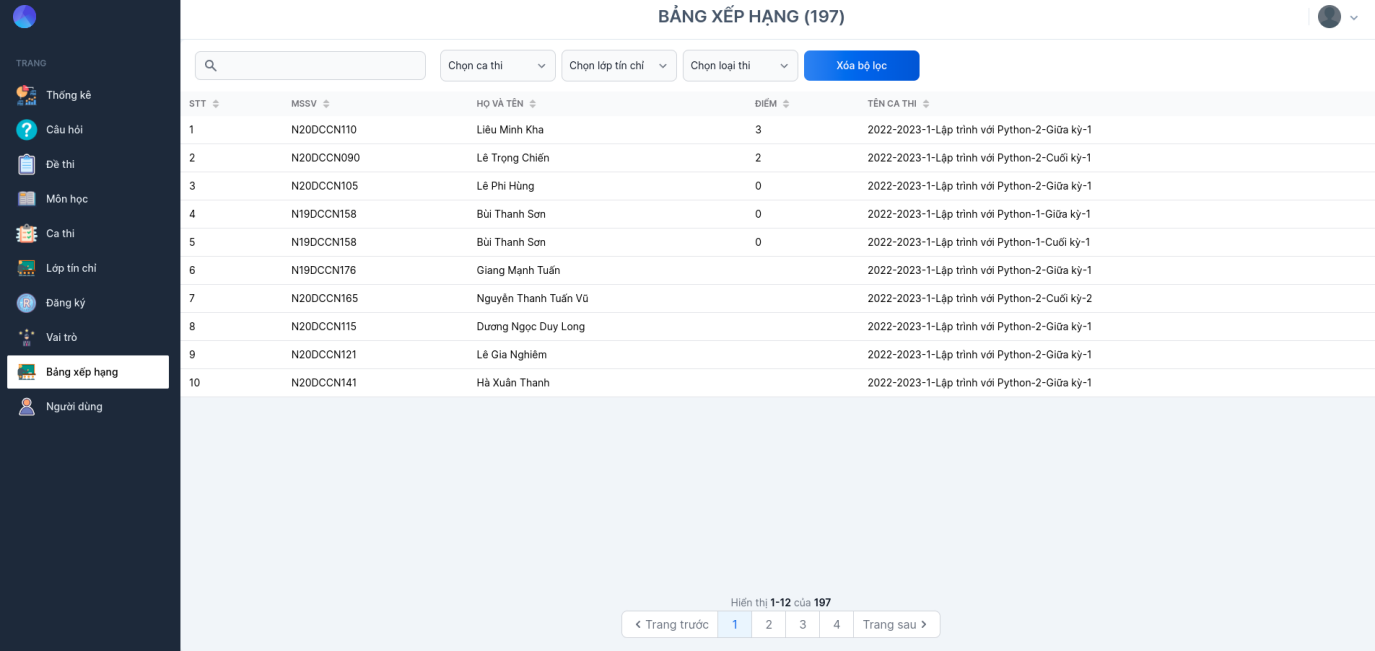
* Code: 200 OK
* Body: <String>: Trả về thông báo thêm vai trò thành công.

***Error Response:***

*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

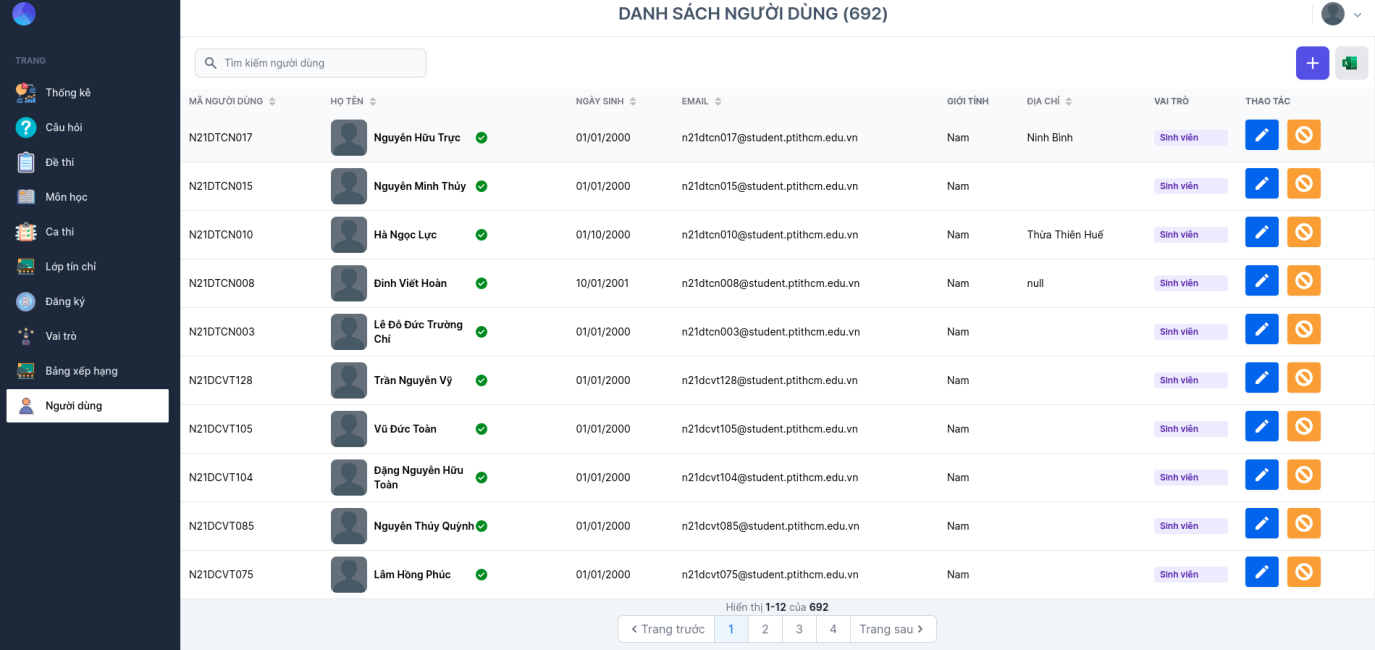
*Code 400 Bad Request – Tên vai trò đã tồn tại.*

1. **Quản lý người xếp hạng**



*Hình 4.9 Giao diện Bảng xếp hạng*

1. **Quản lý người dùng**



*Hình 4.10 Giao diện Quản lý người dùng*

***POST: /api/auth/register***

***Success Response:***

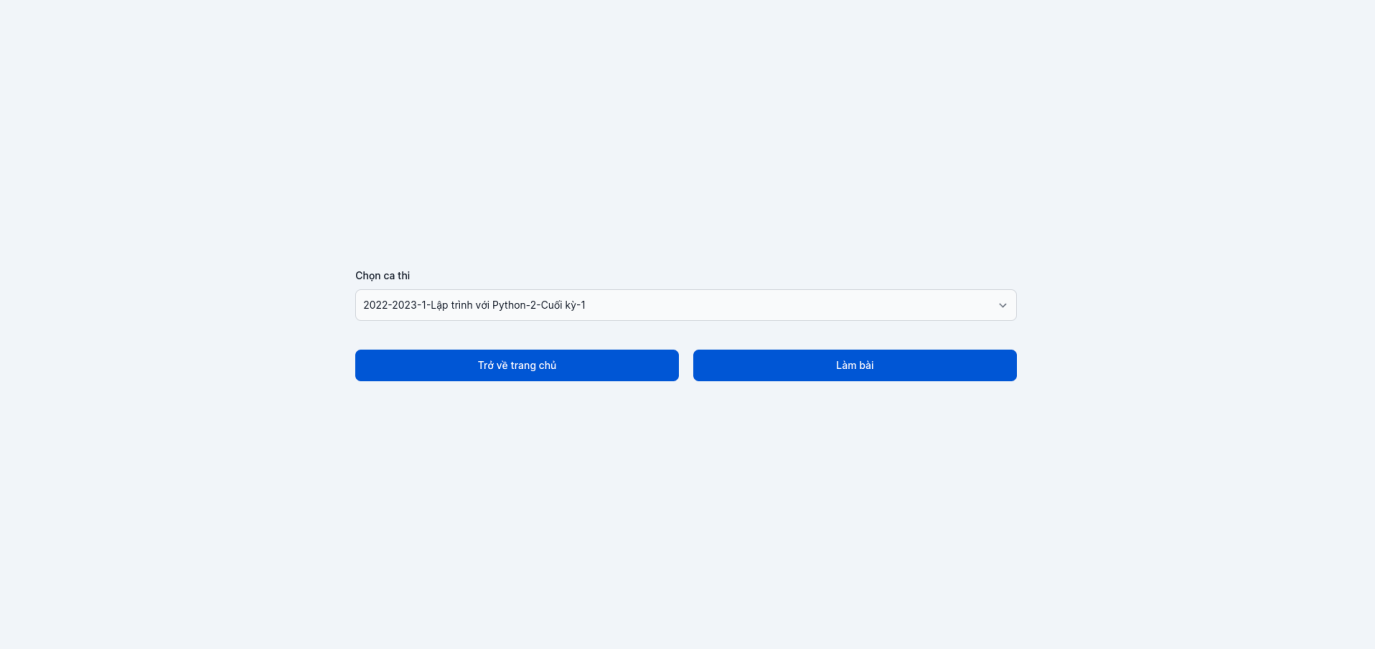
* Code: 200 OK
* Body: <String>: Trả về thông báo thêm người dùng thành công.

***Error Response:***

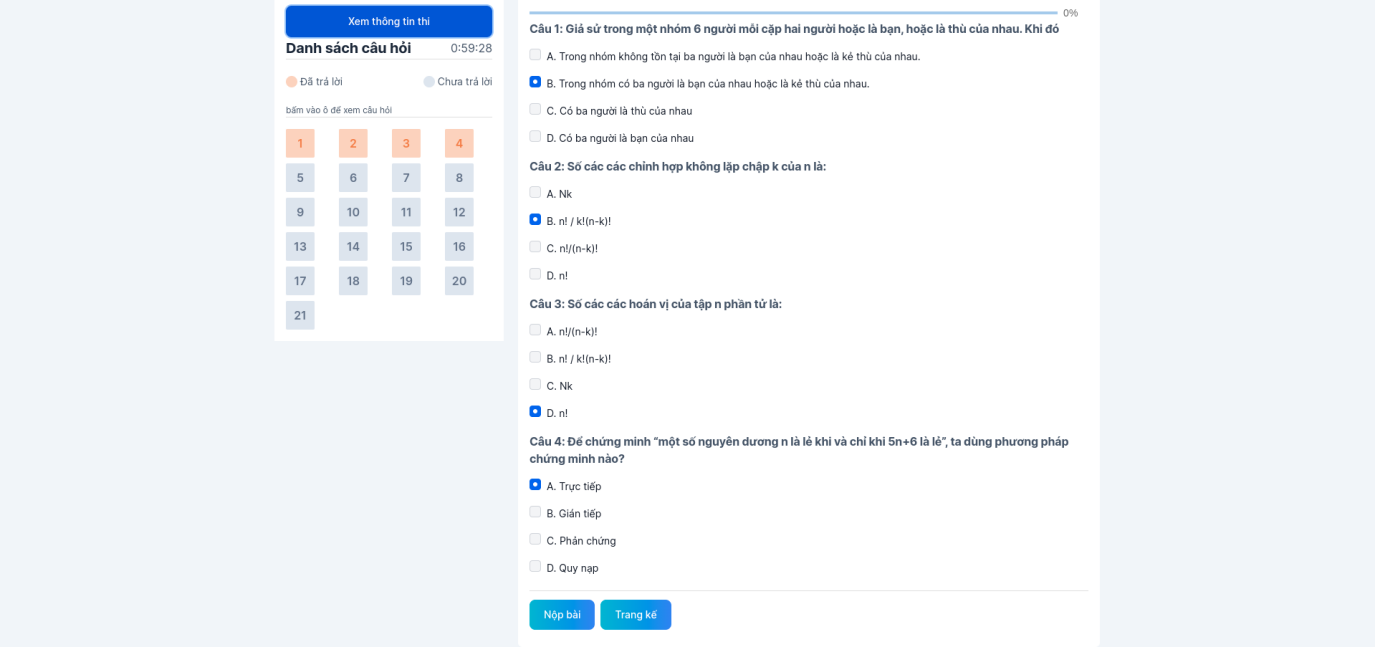
*Code 400 Bad Request – Kiểm tra NOT NULL các trường dữ liệu không được phép NULL.*

*Code 400 Bad Request – Tuổi phải lớn hơn mười tám, địa chỉ email đã được sử dụng.*

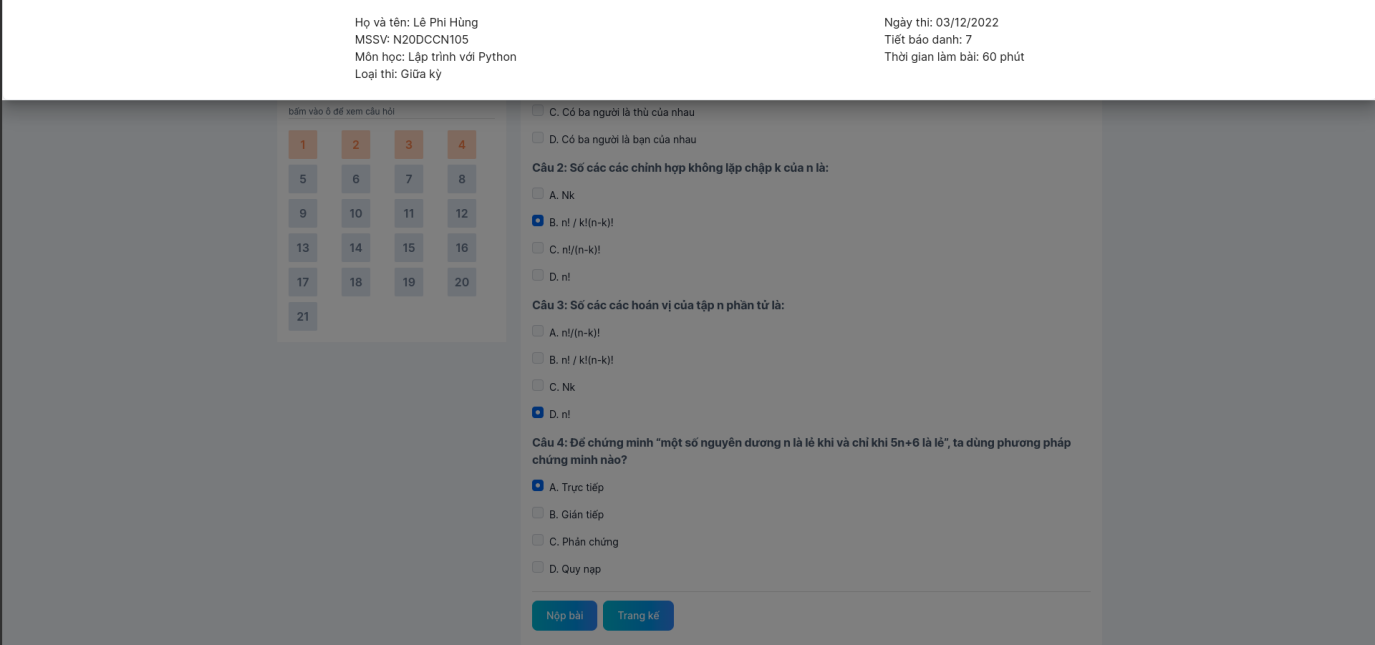
1. **Quản lý thi trắc nghiệm**



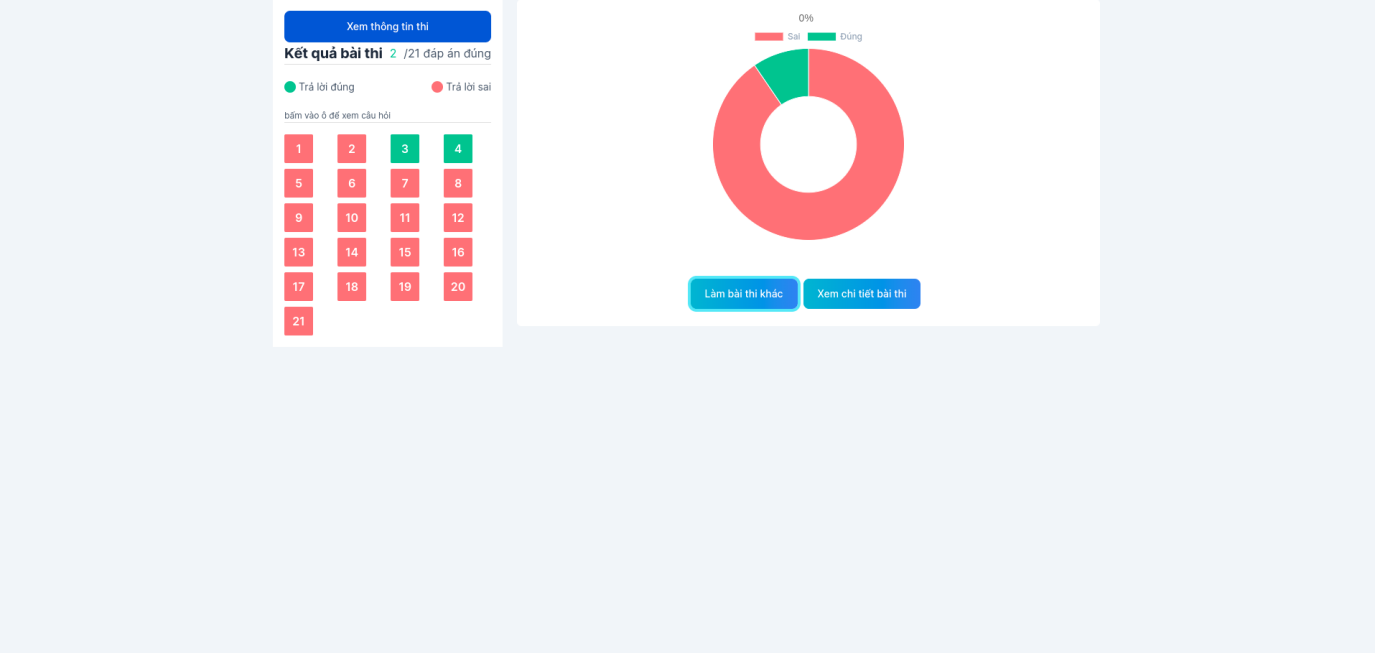
*Hình 4.11.1 Giao diện Chọn ca thi*



*Hình 4.11.2 Giao diện Làm bài thi*



*Hình 4.11.3 Giao diện Xem thông tin sinh viên đang thi*

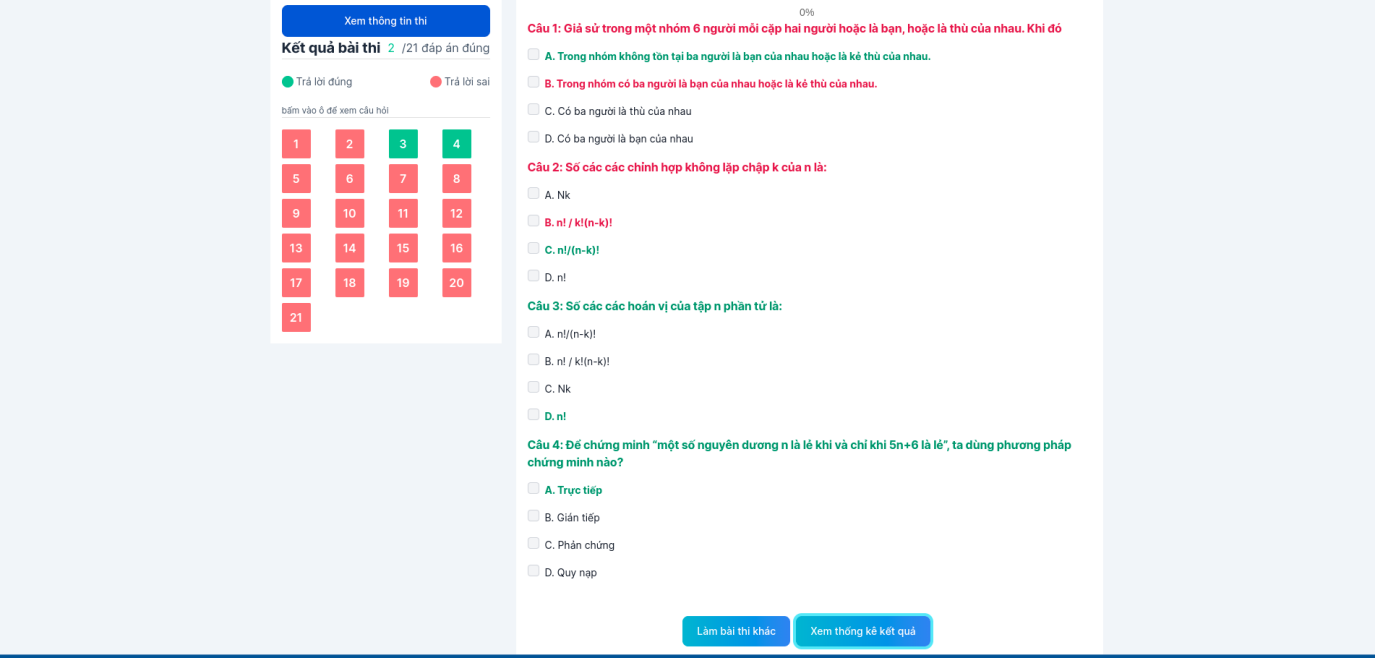


*Hình 4.11.4 Giao diện Nộp bài thi*

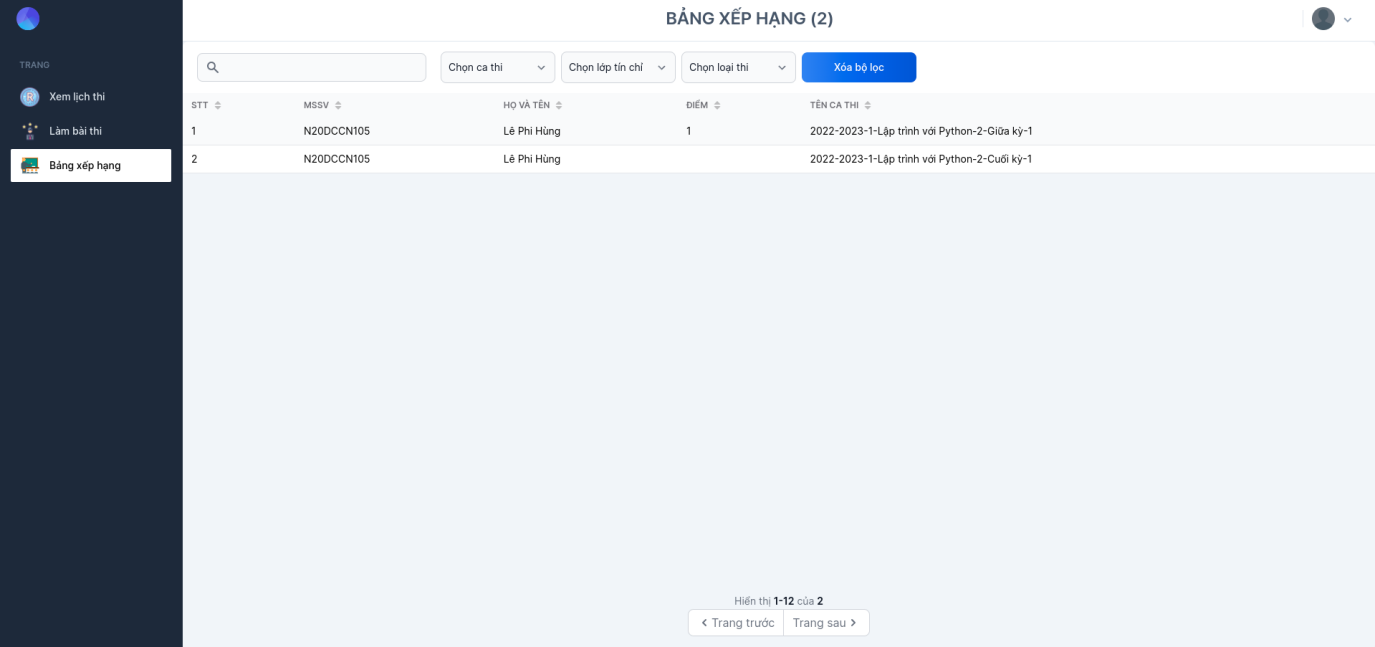
***POST: /api/tests/${examId}/handIn***

***Success Response:***

* Code: 200 OK
* Body: <TestDTO>: Trả về kết quả thi và chi tiết bài thi của sinh viên.



*Hình 4.11.5 Giao diện Xem chi tiết thi*



*Hình 4.11.6 Giao diện Xem bảng xếp hạng của sinh viên*

# KẾT LUẬN

* 1. **Kết quả đạt được**

Đề tài đã hoàn thành được các tiêu chí của đề cương đưa ra, xây dựng được một Website thi trắc nghiệm tương đối hoàn thiện về mặt giao diện, đảm bảo đầy đủ chức năng về cho thi cũng như quản lý thi. Phần mềm có giao diện dễ sử dụng, đơn giản nhưng hiệu quả. Bên cạnh đó, đề tài cũng còn những điểm thiếu sót như sau và cần có định hướng phát triển thêm trong tương lai: Cho phép các đề thi được tạo ra không trùng quá 20% số lượng câu hỏi, đảo thứ tự đáp án của cùng một câu hỏi, cho phép phục hồi bài thi đang làm nếu có sự cố bất ngờ xảy ra.

* 1. **Hạn chế**
* Chương trình chỉ phân quyền ở mức độ ứng dụng, chưa đảm bảo tín bảo mật cơ sở dữ liệu.
* Thiếu sự đa dạng về nội dung và hình thức của câu hỏi (câu hỏi bằng âm thanh, bằng hình ảnh, video…)
* Dữ liệu cơ sở dữ liệu còn thụ động khó cập nhật thông tin khi dữ liệu thực tế bị thay đổi, chưa có chức năng cập nhật lại dữ liệu địa chỉ thực tế.
  1. **Hướng phát triển**

Tiếp tục hoàn thiện và phát triển thêm đề tài để phục vụ tốt hơn nhu cầu quản lý và cho thi trắc nghiệm của trường đại học, bổ sung các chức năng còn thiếu để giúp đề tài nâng cao hơn về tính hoàn thiện.

**TÀI LIỆU THAM KHẢO**

1. Giao diện làm bài thi trắc nghiệm: https://tracnghiemhay.com/.
2. Khái niệm về ReactJS: https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7.
3. Khái niệm về MySQL: https://www.hostinger.vn/huong-dan/mysql-la-gi/.
4. Khái niệm về Spring Boot: https://topdev.vn/blog/gioi-thieu-ve-spring-boot-spring-boot-la-gi/.
5. ReactJS: https://reactjs.org/.
6. Spring Boot: https://spring.io/projects/spring-bootboot.