

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**XÂY DỰNG PHẦN MỀM THI
TRẮC NGHIỆM**

Người hướng dẫn: **THS. NGUYỄN THỊ VŨ THẢO**
Sinh viên thực hiện: **NGUYỄN ĐỨC DƯƠNG NAM**
Mã số sinh viên: **2051220083**
Lớp: **20CT2**

Đà Nẵng,/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG PHẦN MỀM THI
TRẮC NGHIỆM

Người hướng dẫn: **THS. NGUYỄN THỊ VŨ THẢO**
Sinh viên thực hiện: **NGUYỄN ĐỨC DƯƠNG NAM**
Mã số sinh viên: **2051220083**
Lớp: **20CT2**

Đà Nẵng,/2025

TÓM TẮT

Tên đề tài: Phần mềm thi trắc nghiệm

Sinh viên thực hiện: Nguyễn Đức Dương Nam

Mã số sinh viên: 2051220083

Lớp: 20CT2

Phần mềm thi trắc nghiệm được viết bằng C# và Windows Form . Xây dựng hệ thống với 4 tác nhân chính là admin , giảng viên , trưởng khoa và người dùng. Phần mềm có các chức năng cơ bản như: chỉnh sửa sinh viên, xoá sinh viên, tạo ngân hàng câu hỏi, tạo đề thi, tạo môn thi, tự chấm điểm đề thi, duyệt đề thi, thống kê và xuất file .

LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc đến tất cả các Thầy Cô giáo trong bộ môn, những người đã luôn tận tâm truyền đạt tri thức, khơi dậy cảm hứng học tập và nghiên cứu trong chúng em. Sự nhiệt huyết, tâm huyết và những bài giảng bổ ích của Thầy Cô đã trang bị cho chúng em không chỉ là kiến thức mà còn là tinh thần trách nhiệm và khát vọng chinh phục những đỉnh cao tri thức mới.

Đặc biệt, em xin dành lời cảm ơn chân thành nhất đến cô ThS. Nguyễn Thị Vũ Thảo, người đã tận tình hướng dẫn và đồng hành cùng em trong suốt quá trình thực hiện đề tài này. Sự quan tâm, động viên của cô đã giúp em vượt qua những khó khăn và thử thách trong học tập và nghiên cứu. Em vô cùng biết ơn cô vì đã dành nhiều thời gian quý báu để hướng dẫn, sửa chữa, và cung cấp những lời khuyên sâu sắc, giúp em hoàn thiện đề tài một cách tốt nhất.

Những đóng góp của cô không chỉ đơn thuần là những kiến thức học thuật mà còn là những bài học quý báu về sự kiên nhẫn, sáng tạo và tinh thần không ngừng học hỏi. Sự tận tụy và lòng nhiệt thành của cô đã để lại trong em niềm cảm kích lớn lao, là động lực để em cố gắng hơn nữa trên con đường học tập và sự nghiệp sau này.

Em xin chân thành cảm ơn và kính chúc Quý Thầy Cô dồi dào sức khỏe, hạnh phúc và tiếp tục thành công trong sự nghiệp trồng người cao quý!

Xin chân thành cảm ơn!

CAM ĐOAN

Tôi xin cam đoan rằng đề án tốt nghiệp này là công trình nghiên cứu của cá nhân tôi, được thực hiện một cách trung thực, nghiêm túc và tuân thủ đúng các quy định về liên chính học thuật. Toàn bộ nội dung, dữ liệu, kết quả, và tài liệu tham khảo được trình bày trong đề án này đều được tôi tìm hiểu, nghiên cứu và tổng hợp một cách khách quan, minh bạch.

Tôi xin khẳng định rằng không có bất kỳ hành vi sao chép, vi phạm bản quyền, hay gian lận học thuật nào được thực hiện trong quá trình hoàn thành đề án này. Nếu có bất kỳ sự sai sót hay vi phạm nào, tôi xin hoàn toàn chịu trách nhiệm trước quy định của nhà trường và pháp luật.

Tôi cũng xin gửi lời cảm ơn chân thành đến các Thầy Cô và tất cả những người đã hỗ trợ, góp ý và tạo điều kiện cho tôi trong suốt quá trình thực hiện đề án tốt nghiệp này.

Xin trân trọng cảm ơn!

Sinh viên thực hiện

MỤC LỤC

LỜI CẢM ƠN	i
CAM ĐOAN	ii
MỤC LỤC	iii
DANH SÁCH CÁC HÌNH VẼ	vi
DANH SÁCH KÝ HIỆU VÀ CHỮ VIẾT TẮT	viii
MỞ ĐẦU	ix
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ XÂY DỰNG HỆ THỐNG	1
1.1 Ngôn ngữ lập trình C#.....	1
1.2 Windows Forms	5
1.3 Cơ sở dữ liệu SQL.....	7
1.4 Công cụ lập trình	9
1.4.1 Visual Studio	9
1.4.2 SQL SeverManagement Studio.....	10
CHƯƠNG 2: PHÂN TÍCH CHỨC NĂNG VÀ THIẾT KẾ HỆ THỐNG	12
2.1 Mô tả bài toán	12
2.2 Phân tích chức năng	12
2.2.1 Chức năng của học sinh	12
2.2.2 Chức năng của trưởng khoa.....	13
2.2.3 Chức năng của giáo viên	13
2.2.4 Chức năng của quản trị viên	14
2.3 Biểu đồ UseCase	14
2.3.1 Biểu đồ UseCase của sinh viên	14
2.3.2 Biểu đồ UseCase của người quản trị.....	16
2.3.3 Biểu đồ UseCase của giáo viên.....	19
2.4.4 Biểu đồ UseCase của trưởng khoa	21

2.5 Sơ đồ tuần tự	23
2.5.1 Biểu đồ tuần tự đăng nhập	23
2.5.2 Biểu đồ tuần tự quản lý người dùng	24
2.5.3 Biểu đồ tuần tự quản lý câu hỏi	25
2.5.4 Biểu đồ tuần tự quản lý bài thi	26
2.5.5 Biểu đồ tuần tự thi	27
2.5.6 Biểu đồ tuần tự quản lý môn học	28
2.6 Thiết kế cơ sở dữ liệu.....	29
2.6.1 Sơ đồ quan hệ khóa.....	29
2.6.2 Danh mục các bảng	29
CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM ĐỀ TÀI	34
3.1 Môi trường	34
3.2 Kết quả giao diện	34
3.1.1 Giao diện trang chủ	34
3.1.2 Giao diện trang đăng nhập.....	35
3.1.3 Giao diện trang quên mật khẩu	35
3.1.4 Giao diện trang làm đề thi.....	36
3.1.5 Giao diện trang chọn môn thi.....	36
3.1.6 Giao diện trang kết quả thi	37
3.1.7 Giao diện trang quản lý đề thi	37
3.1.8 Giao diện trang tạo đề thi.....	38
3.1.9 Giao diện trang quản lý người dùng	38
3.1.10 Giao diện trang quản lý môn học	39
3.1.11 Giao diện trang thống kê	39
3.1.12 Giao diện trang tạo đề thi giáo viên	40
3.1.13 Giao diện trang quản lý đề thi giáo viên	40
3.1.14 Giao diện trang quản lý duyệt đề trưởng khoa	41
3.1.15 Giao diện trang quản lý phân công trưởng khoa	41

3.3 Cấu trúc tổ chức code	42
3.2.1 Cấu trúc tổ chức Front – End	42
3.2.2 Cấu trúc tổ chức Back - End	42
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	43
1.Kết quả đạt được.....	43
2.Những mặt còn hạn chế.....	43
3.Đề xuất giải pháp khắc phục	44
TÀI LIỆU THAM KHẢO.....	45

DANH SÁCH CÁC HÌNH VẼ

Hình 2. 1 Biểu đồ Use Case quản lý thông tin	14
Hình 2. 2 Biểu đồ Use Case bài thi	15
Hình 2. 3 Biểu đồ Use Case quản lý đề thi.....	16
Hình 2. 4 Biểu đồ Use Case quản lý người dùng	17
Hình 2. 5 Biểu đồ Use Case quản lý môn học.....	18
Hình 2. 6 Biểu đồ Use Case tạo đề thi.....	19
Hình 2. 7 Biểu đồ Use Case quản lý đề thi.....	20
Hình 2. 8 Biểu đồ Use Case Duyệt đề	21
Hình 2. 9 Biểu đồ Use Case quản lý phân công	22
Hình 2. 10 Sơ đồ tuần tự đăng nhập	23
Hình 2. 11 Sơ đồ tuần tự quản lý người dùng	24
Hình 2. 12 Sơ đồ tuần tự quản lý câu hỏi	25
Hình 2. 13 Sơ đồ tuần tự quản lý bài thi.....	26
Hình 2. 14 Sơ đồ tuần tự thi	27
Hình 2. 15 Sơ đồ tuần tự quản lý môn học.....	28
Hình 2. 16 Sơ đồ quan hệ thực thể	29
Hình 3. 1 Giao diện trang chủ	34
Hình 3. 2 Giao diện trang đăng nhập.....	35
Hình 3. 3 giao diện trang quên mật khẩu.....	35
Hình 3. 4 Giao diện trang làm đề thi	36
Hình 3. 5 Giao diện trang chọn môn thi	36
Hình 3. 6 Giao diện trang kết quả thi	37
Hình 3. 7 Giao diện trang quản lý đề thi	37
Hình 3. 8 Giao diện trang tạo đề thi	38
Hình 3. 9 Giao diện trang quản lý người dùng	38
Hình 3. 10 Giao diện trang quản lý môn học	39
Hình 3. 11 Giao diện trang thống kê	39
Hình 3. 12 Giao diện trang tạo đề thi giáo viên	40
Hình 3. 13 Giao diện trang quản lý đề thi giáo viên	40
Hình 3. 14 Giao diện trang quản lý duyệt đề trường khoa	41
Hình 3. 15 Giao diện trang quản lý phân công trường khoa	41

DANH SÁCH CÁC BẢNG

Bảng 2. 1 Đặc tả Use Case quản lý thông tin	15
Bảng 2. 2 Đặc tả Use Case bài thi	15
Bảng 2. 3 Use case Quản lý đề thi.....	16
Bảng 2. 4 Use case Quản lý đề thi.....	17
Bảng 2. 5 Use Case quản lý môn học	18
Bảng 2. 6 Use Case tạo đề thi.....	19
Bảng 2. 7 Use case quản lý đề thi.....	20
Bảng 2. 8 Use case duyệt đề.....	21
Bảng 2. 9 Use case quản lý phân công	22
Bảng 2. 10 Bảng dữ liệu Admin	29
Bảng 2. 11 Bảng dữ liệu User	30
Bảng 2. 12 Bảng dữ liệu Department_Heads	30
Bảng 2. 13 Bảng dữ liệu Teachers.....	30
Bảng 2. 14 Bảng dữ liệu Exam_Appovals	31
Bảng 2. 15 Bảng dữ liệu Ranking	31
Bảng 2. 16 Bảng dữ liệu History_Test	31
Bảng 2. 17 Bảng dữ liệu Question.....	32
Bảng 2. 18 Bảng dữ liệu List_exam	32
Bảng 2. 19 Bảng dữ liệu Teacher_Subjects.....	32

DANH SÁCH KÝ HIỆU VÀ CHỮ VIẾT TẮT

Ký hiệu	Tiếng Anh	Tiếng Việt
C#	C Sharp	Ngôn ngữ lập trình C#
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
SSMS	SQL Server Management Studio	Công cụ quản lý SQL Server
IDE	Integrated Development Environment	Môi trường phát triển tích hợp
GUI	Graphical User Interface	Giao diện đồ họa người dùng
ADO.NET	ActiveX Data Objects .NET	Công nghệ truy cập dữ liệu .NET

MỞ ĐẦU

1. Tổng quan đề tài

Trong bối cảnh sự phát triển mạnh mẽ của công nghệ thông tin, việc áp dụng các nền tảng trực tuyến vào giáo dục đã trở thành xu hướng tất yếu. Một trong những giải pháp nổi bật là hệ thống thi trắc nghiệm trực tuyến, giúp nâng cao tính hiệu quả và linh hoạt trong kiểm tra, đánh giá kiến thức của học sinh, sinh viên.

Đề tài “Xây dựng phần mềm thi trắc nghiệm” hướng đến việc phát triển một nền tảng giúp tổ chức các bài kiểm tra một cách dễ dàng, nhanh chóng và chính xác. Phần mềm sẽ cung cấp giao diện trực quan, hỗ trợ người dùng (giáo viên, học sinh, trưởng khoa) trong việc tạo và tham gia các kỳ thi với nhiều tính năng như quản lý câu hỏi, thời gian làm bài, chấm điểm tự động, và thống kê kết quả.

2. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài này bao gồm.

- Các phương pháp và kỹ thuật phát triển hệ thống thi trắc nghiệm trực tuyến.
- Nhu cầu sử dụng công cụ trực tuyến trong giảng dạy và kiểm tra của giáo viên và học sinh.
- Các công nghệ phù hợp để xây dựng hệ thống như C#, Windows Form, SQL
- Quy trình tổ chức và quản lý các kỳ thi trực tuyến, đảm bảo tính bảo mật và công bằng trong khi thi.

3. Phạm vi nghiên cứu

Nội dung: tập trung phát triển chức năng chính của hệ thống, tạo đề thi, làm bài kiểm tra, quản lý câu hỏi và chấm điểm tự động. Hệ thống sẽ không mở rộng đến các chức năng phức tạp như tích hợp mạng xã hội như tích hợp mạng xã hội hoặc hỗ trợ học tập toàn diện.

Không gian: Triển khai nghiên cứu tại Trường Đại Học Kiến Trúc Đà Nẵng

Thời gian: Thực hiện trong khoản thời gian từ ngày 05/09/2025 đến ngày 30/11/2025

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ XÂY DỰNG HỆ THỐNG

1.1 Ngôn ngữ lập trình C#

- ❖ C# (C Sharp) là ngôn ngữ lập trình bậc cao do Microsoft phát triển trong khuôn khổ nền tảng .NET. Ngay từ khi ra đời, C# được định hướng trở thành một ngôn ngữ mạnh mẽ, hiện đại và an toàn, đáp ứng tốt các yêu cầu của công nghiệp phần mềm. Nhờ thừa hưởng ưu điểm của nhiều ngôn ngữ như Java, C++, VB.NET, C# đã nhanh chóng trở thành lựa chọn phổ biến trong lập trình ứng dụng doanh nghiệp, từ hệ thống web, desktop đến mobile và game.
- ❖ Khả năng tích hợp sâu với hệ sinh thái .NET giúp C# trở thành công nghệ chủ lực trong phát triển ứng dụng trên nền tảng Windows – đặc biệt phù hợp cho các hệ thống quản lý và phần mềm nghiệp vụ nội bộ.
- ❖ C# hỗ trợ lập trình hướng đối tượng (OOP) toàn diện, bao gồm các đặc tính như đóng gói, kế thừa, đa hình và trừu tượng. Điều này giúp cấu trúc chương trình rõ ràng, dễ mở rộng và dễ bảo trì – rất cần thiết cho các dự án quy mô lớn.
- ❖ Bộ thư viện phong phú của .NET cho phép lập trình viên xây dựng nhanh các tính năng phức tạp như thao tác dữ liệu, kết nối cơ sở dữ liệu, xử lý mạng, mã hóa – giải mã, giao diện người dùng (UI) và nhiều chức năng hệ thống khác. Nhờ đó, thời gian phát triển phần mềm được rút ngắn đáng kể.
- ❖ C# có khả năng chạy đa nền tảng thông qua .NET Core (nay là .NET 5/6/7/8). Điều này giúp các ứng dụng viết bằng C# không chỉ hoạt động trên Windows mà còn chạy hiệu quả trên Linux, macOS và thậm chí triển khai trên Docker hoặc các dịch vụ cloud như Azure, AWS, GCP.
- ❖ Tính an toàn và ổn định cao: C# quản lý bộ nhớ tự động bằng Garbage Collector, hạn chế lỗi tràn bộ nhớ, rò rỉ tài nguyên – những lỗi phổ biến trong các ngôn ngữ như C/C++.
- ❖ Hỗ trợ lập trình bất đồng bộ (async/await) giúp tăng hiệu năng xử lý, tối ưu hiệu quả của ứng dụng có nhiều tác vụ chạy song song, đặc biệt quan trọng trong lập trình web và dịch vụ API.

Nhờ sự ổn định, cú pháp rõ ràng và cộng đồng lớn mạnh, C# luôn nằm trong nhóm ngôn ngữ lập trình được sử dụng nhiều nhất trên thế giới và tiếp tục phát triển

manh mẽ cùng hệ sinh thái .NET.

Đặc điểm chính của C#:

- Lập trình hướng đối tượng (OOP): C# được xây dựng dựa trên mô hình lập trình hướng đối tượng, hỗ trợ đầy đủ bốn tính chất quan trọng gồm kế thừa, đa hình, trừu tượng hóa và đóng gói. Điều này giúp cấu trúc chương trình rõ ràng, dễ quản lý và mở rộng. Các đối tượng trong chương trình có thể tái sử dụng, giảm thiểu trùng lặp mã nguồn và nâng cao khả năng bảo trì, rất phù hợp với các hệ thống lớn như phần mềm quản lý, ứng dụng doanh nghiệp và hệ thống thi trắc nghiệm.
- Kiểu dữ liệu an toàn: C# là một ngôn ngữ kiểu dữ liệu an toàn, điều này có nghĩa là mỗi biến phải được gán một kiểu dữ liệu cụ thể và kiểm tra tại thời điểm biên dịch.
- Quản lý bộ nhớ tự động: C# sử dụng garbage collector để tự động giải phóng bộ nhớ không sử dụng, giúp người lập trình tránh được các vấn đề liên quan đến quản lý bộ nhớ.
- Phong cách lập trình đồng bộ và bất đồng bộ: C# hỗ trợ cả lập trình đồng bộ và bất đồng bộ thông qua từ khóa async và await, cho phép xử lý các tác vụ đồng thời một cách hiệu quả.
- Đa nền tảng (cross-platform): Với .NET Core và sau này là .NET 5 và .NET 6, C# đã trở thành một ngôn ngữ đa nền tảng, cho phép viết ứng dụng chạy trên nhiều hệ điều hành như Windows, Linux và macOS.
- Phát triển ứng dụng web: C# được sử dụng rộng rãi trong phát triển các ứng dụng web thông qua ASP.NET Core, một framework mạnh mẽ cho việc xây dựng các ứng dụng web hiện đại.
- Ứng dụng di động: C# có thể được sử dụng để phát triển ứng dụng di động trên các nền tảng như iOS và Android thông qua Xamarin.
- Game development: C# cũng được sử dụng rộng rãi trong phát triển trò chơi, đặc biệt là trên nền tảng Unity, một môi trường phát triển trò chơi đa nền tảng mạnh mẽ.
- Cộng đồng lớn: C# có một cộng đồng lập trình viên lớn, với nhiều tài liệu, thư viện mã nguồn mở và hỗ trợ trực tuyến.
- Khả năng mở rộng bảo trì cao: Nhờ cấu trúc hướng đối tượng, C# cho phép chia chương trình thành nhiều lớp, module và namespace riêng biệt. Điều này giúp quản lý mã nguồn khoa học, dễ chỉnh sửa hoặc mở rộng thêm chức năng mà không ảnh hưởng đến toàn bộ hệ thống. Đây là lợi thế lớn khi phát triển các dự

án dài hạn như hệ thống quản lý sinh viên, phần mềm thi trắc nghiệm, phần mềm kế toán,...

- Hỗ trợ lập trình tổng quát (Generics): C# hỗ trợ Generics để xây dựng các lớp, phương thức và kiểu dữ liệu có thể tái sử dụng với nhiều dạng dữ liệu khác nhau. Generics giúp tăng hiệu năng, giảm lỗi ép kiểu và cải thiện chất lượng mã nguồn. Các cấu trúc như List<T>, Dictionary<TKey, TValue> là ví dụ điển hình trong thư viện chuẩn của C#.
- Tính bảo mật cao: C# tích hợp nhiều cơ chế bảo mật ở mức ngôn ngữ và mức framework như kiểm soát quyền truy cập, mã hóa dữ liệu, xác thực người dùng và cơ chế sandbox. Điều này rất quan trọng khi xây dựng các hệ thống đòi hỏi tính bảo mật cao như phần mềm ngân hàng, quản lý thông tin khách hàng hoặc hệ thống thi trực tuyến.
- Hỗ trợ đa luồng (Multithreading): C# cung cấp các lớp như Thread, Task, Parallel giúp xử lý nhiều công việc cùng lúc trên các luồng khác nhau. Điều này nâng cao hiệu suất, đặc biệt đối với các chương trình cần xử lý khối lượng dữ liệu lớn, tính toán nặng hoặc thao tác nhiều kết nối mạng đồng thời.
- Tích hợp tốt với cơ sở dữ liệu: C# làm việc rất hiệu quả với nhiều hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, PostgreSQL,... nhờ các công cụ mạnh mẽ như Entity Framework Core. Lập trình viên có thể thao tác dữ liệu thông qua mô hình ORM, giúp code sạch hơn, dễ bảo trì và giảm lỗi truy vấn.

Tổng thể, C# là một ngôn ngữ linh hoạt và mạnh mẽ, được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau từ phát triển ứng dụng desktop đến web và trò chơi.

❖ **Lợi ích của C# :**

- Hiệu năng cao: C# được biên dịch thành mã trung gian (IL – Intermediate Language) và sau đó được tối ưu bởi bộ JIT (Just-In-Time Compiler) của .NET. Nhờ vậy, ứng dụng C# có tốc độ xử lý nhanh, ổn định và tận dụng tốt tài nguyên hệ thống. Với các bản .NET mới, hiệu năng của C# ngày càng được cải thiện, phù hợp cho cả ứng dụng doanh nghiệp, dịch vụ web và xử lý dữ liệu lớn.
- Dễ học và sử dụng: Cú pháp của C# tương đối rõ ràng, dễ đọc và nhất quán. Ngôn ngữ được thiết kế hướng tới sự đơn giản nhưng mạnh mẽ, giúp người mới học lập trình dễ nắm bắt. Các khái niệm như lớp, đối tượng, hàm, biến, điều kiện và vòng lặp đều rất trực quan, gần gũi với Java hoặc C++, giúp giảm thời gian làm quen.
- Bảo mật tốt: C# tích hợp nhiều cơ chế bảo mật ở cả mức ngôn ngữ và mức nền tảng .NET, chẳng hạn như kiểm tra kiểu dữ liệu, kiểm soát quyền truy cập, mã hóa – giải mã và quản lý bộ nhớ tự động bằng Garbage Collector. Điều này

giúp giảm thiểu rủi ro như rò rỉ bộ nhớ, lỗi tràn bộ nhớ hay lỗi truy cập vùng nhớ trái phép – những lỗi nguy hiểm thường gặp trong các ngôn ngữ cấp thấp như C/C++.

- Đa nền tảng: Nhờ .NET Core và hiện tại là .NET 5/6/7/8, C# có khả năng chạy trên nhiều hệ điều hành như Windows, Linux và macOS. Có thể dùng để phát triển ứng dụng trên Windows, web, di động và game.
- Tái sử dụng mã: C# hỗ trợ lập trình hướng đối tượng (OOP), generics, inheritance, interface,... giúp xây dựng các module có khả năng tái sử dụng cao. Các lớp, phương thức và thư viện có thể dùng lại trong nhiều dự án khác nhau, giảm thời gian phát triển và tăng độ ổn định của phần mềm. Điều này đặc biệt hữu ích trong các dự án lớn hoặc khi phát triển theo mô hình nhiều tầng (multi-layer architecture).
- Cộng đồng lớn: C# được Microsoft phát triển và liên tục cập nhật, cùng với đó là cộng đồng lập trình viên đông đảo trên toàn thế giới. Người học và người làm việc với C# có thể dễ dàng tìm tài liệu, khóa học, ví dụ code và các thư viện mã nguồn mở hỗ trợ. Ngoài ra, các nền tảng như GitHub, StackOverflow, Microsoft Docs... đều cung cấp lượng tài nguyên vô cùng phong phú cho C# và .NET.

❖ **Ứng dụng thực tế:**

C# được sử dụng rộng rãi trong nhiều lĩnh vực phát triển phần mềm nhờ tính linh hoạt, hiệu suất cao và khả năng tích hợp mạnh mẽ với nền tảng .NET. Một số ứng dụng thực tế phổ biến của C# bao gồm:

- Phần mềm máy tính để bàn: Dùng để xây dựng các ứng dụng quản lý, kế toán, bán hàng, hoặc hệ thống điều khiển nội bộ thông qua WinForms và WPF.
- Ứng dụng web: Sử dụng cùng ASP.NET để phát triển các website, hệ thống quản lý nội dung, và ứng dụng web động cho doanh nghiệp.
- Phát triển game: Là ngôn ngữ chính của Unity, C# được dùng để tạo các trò chơi 2D, 3D và thực tế ảo (VR/AR) trên nhiều nền tảng khác nhau.
- Ứng dụng di động: Với Xamarin hoặc .NET MAUI, C# cho phép phát triển ứng dụng di động chạy trên cả Android và iOS chỉ với một mã nguồn chung.
- Ứng dụng doanh nghiệp: C# thường được dùng để xây dựng các hệ thống lớn như ERP, CRM, và các giải pháp quản lý dữ liệu nội bộ.
- Công nghệ mới: C# còn được áp dụng trong lĩnh vực IoT, trí tuệ nhân tạo (AI) và điện toán đám mây (Cloud) thông qua nền tảng Microsoft Azure.

Nhờ sự linh hoạt và hệ sinh thái mạnh mẽ, C# không chỉ là ngôn ngữ quan trọng trong phát triển phần mềm doanh nghiệp mà còn là nền tảng vững chắc cho nhiều lĩnh vực công nghệ hiện đại.

1.2 Windows Forms

Windows Forms (WinForms) là một phần quan trọng của .NET Framework, cung cấp bộ công cụ mạnh mẽ giúp lập trình viên xây dựng các ứng dụng Windows Desktop với giao diện đồ họa trực quan. WinForms hoạt động dựa trên mô hình lập trình hướng sự kiện (Event-Driven Programming), trong đó mỗi hành động của người dùng như nhấp chuột, nhập dữ liệu, hay di chuyển chuột đều được xử lý thông qua các sự kiện (events). Điều này giúp lập trình viên dễ dàng kiểm soát và phản hồi lại các tương tác từ người dùng.

WinForms cung cấp sẵn một tập hợp lớn các control như:

- Button, TextBox, Label, CheckBox, RadioButton
- DataGridView để hiển thị bảng dữ liệu
- ListView, TreeView để quản lý dữ liệu phân cấp
- Panel, GroupBox để bố cục giao diện Các control này có thể được kéo – thả trực tiếp trên giao diện thiết kế của Visual Studio, giúp quá trình xây dựng UI diễn ra nhanh chóng và trực quan.

Bên cạnh đó, Windows Forms còn hỗ trợ tùy chỉnh giao diện thông qua thuộc tính (properties), phương thức (methods) và sự kiện (events). Lập trình viên có thể thay đổi màu sắc, kích thước, phông chữ, hình ảnh, cách bố trí,... tạo nên những giao diện thân thiện và phù hợp với nhu cầu của từng ứng dụng.

❖ Đặc điểm nổi bật của Windows Forms:

- Giao diện người dùng đồ họa: Windows Forms cung cấp một bộ công cụ đồ họa phong phú cho phép lập trình viên thiết kế giao diện người dùng một cách trực quan thông qua thao tác kéo – thả trên Visual Studio. Các điều khiển như Button, TextBox, CheckBox, RadioButton, ComboBox, DataGridView... được cung cấp sẵn, giúp xây dựng giao diện nhanh chóng mà không cần viết nhiều mã. Điều này đặc biệt phù hợp với các ứng dụng quản lý, hệ thống nội bộ hoặc chương trình phục vụ nhu cầu doanh nghiệp.
- Lập trình dựa trên sự kiện (Event-driven programming): Windows Forms hoạt động theo mô hình hướng sự kiện, trong đó mọi hành động của người dùng như nhấn nút, nhập dữ liệu, di chuyển chuột hoặc thay đổi giá trị đều kích hoạt các sự kiện tương ứng. Lập trình viên chỉ cần xử lý các sự kiện này thông qua hàm

xử lý (event handler). Điều này giúp cho quá trình phát triển ứng dụng trở nên rõ ràng, logic và dễ mở rộng.

- Tích hợp dữ liệu (Data Binding): WinForms hỗ trợ cơ chế Data Binding cho phép liên kết trực tiếp các control với dữ liệu từ nhiều nguồn như cơ sở dữ liệu, DataSet, danh sách đối tượng hoặc các API. Điều này giúp giảm việc viết mã xử lý thủ công và tăng tính đồng nhất trên giao diện. Đặc biệt, các control như DataGridView hay ListBox hoạt động rất hiệu quả với Data Binding, phù hợp cho các ứng dụng quản lý dữ liệu.
- Tính năng điều khiển và định dạng: Windows Forms cho phép tùy chỉnh và định dạng các điều khiển giao diện người dùng một cách linh hoạt, bao gồm kích thước, màu sắc, phông chữ và hình ảnh.
- Hỗ trợ cho nhiều loại ứng dụng: Windows Forms không chỉ dành riêng cho ứng dụng desktop truyền thống, mà còn có thể được sử dụng trong các ứng dụng nhúng và các ứng dụng khác trên nền tảng Windows.
- Hỗ trợ cho các kiểu dự án khác nhau: Windows Forms có thể được sử dụng trong các kiểu dự án khác nhau như ứng dụng cửa sổ đơn, ứng dụng MDI (Multiple Document Interface), hoặc ứng dụng đa cửa sổ.

❖ ***Lợi ích sử dụng Windows Forms :***

- Dễ sử dụng: Windows Forms hỗ trợ giao diện thiết kế kéo – thả trực quan trong Visual Studio, giúp lập trình viên dễ dàng bố trí các control như nút bấm, ô nhập liệu hay bảng dữ liệu chỉ bằng thao tác chuột. Điều này giúp giảm đáng kể độ phức tạp trong việc xây dựng giao diện và rất phù hợp cho sinh viên, người mới học lập trình hoặc những dự án cần phát triển nhanh.
- Phát triển nhanh: WinForms hỗ trợ nhiều công cụ tích hợp trong Visual Studio như trình thiết kế giao diện (Designer), cửa sổ thuộc tính (Properties Panel), bộ xử lý sự kiện tự động và trình gỡ lỗi (Debugger). Ngoài ra, .NET cung cấp sẵn nhiều thư viện mạnh mẽ phục vụ xử lý dữ liệu, tính toán, đồ họa và kết nối cơ sở dữ liệu, giúp lập trình viên giảm thời gian code và tăng tốc độ hoàn thành sản phẩm.
- Tích hợp tốt với .NET: Windows Forms hoạt động ổn định trong môi trường .NET, có thể dễ dàng kết nối với cơ sở dữ liệu, API và các dịch vụ khác của Microsoft.
- Giao diện thân thiện: Các ứng dụng được xây dựng bằng WinForms thường có giao diện đơn giản, trực quan và gần gũi với người dùng Windows. Nhờ đó, người dùng không cần nhiều thời gian để làm quen. WinForms cũng hỗ trợ tùy chỉnh giao diện như màu sắc, kích thước, hình ảnh, biểu tượng và phong cách

hiển thị giúp ứng dụng mang tính chuyên nghiệp và dễ sử dụng trong môi trường doanh nghiệp

- Hỗ trợ mở rộng: Có thể kết hợp với các công nghệ khác như Entity Framework, LINQ, hoặc thư viện bên thứ ba để tăng tính năng và hiệu quả của ứng dụng.
- Độ ổn định cao và phù hợp cho phần mềm doanh nghiệp: Windows Forms đã được sử dụng hơn 20 năm và chứng minh độ ổn định trong vô số hệ thống nội bộ của doanh nghiệp, trường học và tổ chức. Dù Microsoft ra mắt nhiều công nghệ mới, WinForms vẫn được duy trì và hỗ trợ, là lựa chọn đáng tin cậy cho các ứng dụng quản lý, ứng dụng hành chính và công cụ nội bộ.

❖ **Ứng dụng thực tế:**

- Windows Forms được sử dụng phổ biến trong việc phát triển các ứng dụng máy tính để bàn, đặc biệt là trong môi trường doanh nghiệp và quản lý nội bộ, chẳng hạn như:
- Phần mềm quản lý: Dùng để xây dựng các ứng dụng quản lý bán hàng, kho hàng, nhân sự, tài chính, hay điểm danh.
- Công cụ nội bộ doanh nghiệp: Tạo ra các ứng dụng hỗ trợ công việc văn phòng như nhập liệu, thống kê, hoặc quản lý dữ liệu.
- Phần mềm tiện ích: Các công cụ nhỏ phục vụ cho xử lý file, cấu hình hệ thống hoặc hỗ trợ kỹ thuật.
- Ứng dụng giáo dục: Dùng để phát triển phần mềm học tập, thi trắc nghiệm hoặc quản lý điểm học sinh.
- Ứng dụng kết nối cơ sở dữ liệu: Windows Forms giúp dễ dàng xây dựng ứng dụng truy xuất, thêm, sửa, xóa dữ liệu từ SQL Server hoặc các hệ quản trị khác

1.3 Cơ sở dữ liệu SQL

SQL (Structured Query Language) là một ngôn ngữ lập trình được thiết kế để quản lý và tương tác với các hệ thống quản lý cơ sở dữ liệu (Database Management Systems - DBMS).

SQL là thành phần quan trọng trong mọi ứng dụng quản lý dữ liệu. Trong các hệ thống phần mềm doanh nghiệp như quản lý bán hàng, nhân sự, kho, tài chính..., SQL dùng để lưu trữ và truy xuất dữ liệu thông qua ứng dụng được viết bằng C#, Java, PHP hoặc các ngôn ngữ khác. Dưới đây là một giới thiệu về cơ sở dữ liệu SQL:

❖ **Đặc điểm nổi bật của SQL**

- Cấu trúc dữ liệu có tổ chức (Structured Data): Dữ liệu trong SQL được lưu trữ trong các bảng (tables) có hàng (rows) và cột (columns), giúp việc quản lý, truy xuất và phân tích trở nên rõ ràng và nhất quán.

- Quan hệ giữa các bảng (Relational Model): SQL sử dụng mô hình quan hệ để liên kết dữ liệu giữa các bảng. Các khóa như Primary Key và Foreign Key giúp đảm bảo dữ liệu không bị trùng lặp, đúng logic và luôn có sự liên kết chặt chẽ. Đây là nền tảng quan trọng trong các hệ thống quản lý phức tạp như quản lý sinh viên, bán hàng, ngân hàng...
- Ngôn ngữ truy vấn mạnh mẽ: SQL cung cấp các lệnh như SELECT, INSERT, UPDATE, DELETE, cùng nhiều câu lệnh nâng cao như JOIN, GROUP BY, HAVING giúp truy vấn dữ liệu linh hoạt.
- Tính nhất quán và toàn vẹn dữ liệu (Data Integrity): Hệ thống SQL hỗ trợ nhiều loại ràng buộc như NOT NULL, UNIQUE, CHECK, PRIMARY KEY, FOREIGN KEY giúp dữ liệu luôn chính xác. Cùng với cơ chế giao dịch (Transactions), SQL đảm bảo dữ liệu không bị lỗi hoặc mất mát khi xảy ra sự cố trong quá trình xử lý.
- Bảo mật cao: SQL cung cấp nhiều cơ chế bảo mật tích hợp như phân quyền truy cập (GRANT/REVOKE), xác thực người dùng, mã hóa dữ liệu và kiểm soát truy cập theo mức độ. Điều này giúp SQL được ứng dụng mạnh mẽ trong các hệ thống cần bảo mật cao như y tế, tài chính, giáo dục và chính phủ.
- Khả năng mở rộng (Scalability): Các hệ quản trị SQL hiện đại hỗ trợ mở rộng theo chiều dọc (Vertical Scaling) và chiều ngang (Horizontal Scaling), cho phép xử lý lượng dữ liệu lớn mà vẫn đảm bảo hiệu suất tốt.

❖ **Lợi ích khi sử dụng SQL**

- Dễ học và phổ biến: SQL có cú pháp gần gũi, dễ hiểu, dễ tiếp cận cho cả người mới bắt đầu lẫn người dùng chuyên nghiệp. SQL cũng là tiêu chuẩn chung cho hầu hết các hệ quản trị cơ sở dữ liệu hiện nay, giúp việc học và áp dụng trở nên thuận tiện.
- Tương thích cao: SQL được hỗ trợ bởi nhiều hệ quản trị nổi tiếng như MySQL, SQL Server, PostgreSQL, Oracle, MariaDB, giúp lập trình viên dễ dàng chuyển đổi hoặc triển khai hệ thống trên các nền tảng khác nhau mà không cần viết lại toàn bộ cấu trúc dữ liệu.
- Quản lý dữ liệu hiệu quả: SQL được tối ưu để xử lý các tập dữ liệu lớn, hỗ trợ tìm kiếm, lọc, thống kê và tổng hợp dữ liệu chỉ bằng vài câu lệnh. Điều này đặc biệt hữu ích trong các hệ thống báo cáo, phân tích doanh nghiệp (Business Intelligence) hoặc các hệ thống lưu trữ dữ liệu lớn.
- Tính ổn định và đáng tin cậy: Các hệ thống cơ sở dữ liệu dựa trên SQL được thiết kế lâu đời và đã chứng minh độ bền vững. Doanh nghiệp tin dùng SQL vì

khả năng xử lý ổn định, độ tin cậy cao, hỗ trợ backup – restore tốt và đảm bảo dữ liệu an toàn trước lỗi hệ thống.

- Hỗ trợ giao dịch (Transaction): SQL đảm bảo tính ACID (Atomicity, Consistency, Isolation, Durability) giúp các thao tác dữ liệu an toàn và không bị lỗi trong quá trình xử lý.

❖ *Ứng dụng thực tế của SQL*

SQL được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là nơi cần lưu trữ, truy xuất và quản lý dữ liệu có cấu trúc, chẳng hạn như:

- Ứng dụng web và thương mại điện tử: Quản lý tài khoản người dùng, đơn hàng, sản phẩm và thanh toán.
- Hệ thống quản lý doanh nghiệp (ERP, CRM): Lưu trữ và xử lý dữ liệu nhân sự, khách hàng, kế toán và báo cáo.
- Ngân hàng và tài chính: Quản lý giao dịch, thông tin tài khoản và báo cáo tài chính.
- Giáo dục: Xây dựng hệ thống quản lý sinh viên, điểm số, và chương trình đào tạo.
- Phân tích dữ liệu và báo cáo: Dùng SQL để trích xuất và tổng hợp dữ liệu phục vụ phân tích kinh doanh (Business Intelligence)

1.4 Công cụ lập trình

1.4.1 Visual Studio

Visual Studio là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) mạnh mẽ được phát triển bởi Microsoft. Nó cung cấp một loạt các công cụ và tính năng để hỗ trợ việc phát triển các ứng dụng trên nhiều nền tảng và ngôn ngữ lập trình khác nhau. Dưới đây là một số điểm nổi bật về Visual Studio :

- **Hỗ trợ nhiều ngôn ngữ lập trình:** Visual Studio hỗ trợ một loạt các ngôn ngữ lập trình như C#, C++, Visual Basic, F#, Python, JavaScript, TypeScript và nhiều ngôn ngữ khác nữa.
- **Công cụ thiết kế giao diện người dùng (GUI):** Visual Studio cung cấp các công cụ mạnh mẽ để thiết kế giao diện người dùng cho các ứng dụng desktop, web và di động. Các công cụ này cho phép bạn kéo và thả các điều khiển, xây dựng giao diện người dùng một cách trực quan.
- **Debugging và Profiling:** Visual Studio cung cấp một loạt các công cụ để debug và optimize mã nguồn của bạn. Điều này bao gồm việc breakpoint, watch windows, và các công cụ phân tích hiệu suất để tìm ra và sửa lỗi hiệu suất trong mã của bạn.

- **Hỗ trợ Source Control:** Visual Studio tích hợp tốt với các hệ thống quản lý mã nguồn như Git, SVN và TFVC, cho phép bạn làm việc với dự án của mình trong môi trường phát triển phân tán.
- **Công cụ IntelliSense:** Visual Studio cung cấp IntelliSense, một tính năng thông minh giúp dự đoán và hoàn thiện mã của bạn khi bạn gõ. Điều này giúp tăng tốc độ lập trình và giảm thời gian tìm lỗi.
- **Mở rộng với Extensions:** Visual Studio cho phép bạn mở rộng chức năng của nó thông qua các extensions. Cộng đồng Visual Studio đã tạo ra một loạt các extension cho mọi thứ từ quản lý package đến công cụ kiểm tra chất lượng mã.
- **Hỗ trợ đa nền tảng:** Visual Studio hỗ trợ phát triển ứng dụng cho nhiều nền tảng như Windows, macOS, iOS, Android và Linux.

Visual Studio được coi là một trong những môi trường phát triển phổ biến nhất và mạnh mẽ nhất trên thị trường, được sử dụng rộng rãi trong cộng đồng phát triển phần mềm.

1.4.2 SQL Server Management Studio

SQL Server Management Studio (SSMS) là một ứng dụng được Microsoft cung cấp miễn phí để quản lý và tương tác với các cơ sở dữ liệu SQL Server. Đây là một công cụ quan trọng cho các nhà quản trị cơ sở dữ liệu và nhà phát triển ứng dụng SQL Server. Dưới đây là một số điểm nổi bật về SQL Server Management Studio:

- **Giao diện người dùng đồ họa (GUI):** SSMS cung cấp một giao diện người dùng đồ họa trực quan giúp người dùng tương tác với các đối tượng trong cơ sở dữ liệu SQL Server như bảng, thủ tục lưu trữ, chế độ xem và chế độ cảnh báo.
- **Truy vấn dữ liệu:** SSMS cho phép người dùng thực thi và chỉnh sửa các câu lệnh SQL trực tiếp từ giao diện người dùng. Điều này cho phép người dùng thực hiện các thao tác như truy vấn dữ liệu, cập nhật, xóa và thêm mới dữ liệu.
- **Quản lý cơ sở dữ liệu:** SSMS cho phép quản trị viên tạo, xóa và chỉnh sửa cơ sở dữ liệu, bảng, chỉ mục, quan hệ, chế độ xem, thủ tục lưu trữ và các đối tượng khác của cơ sở dữ liệu SQL Server.
- **Xem và quản lý cảnh báo:** SSMS cho phép người dùng xem và quản lý các cảnh báo trong cơ sở dữ liệu, cũng như thiết lập các cảnh báo mới để theo dõi các sự kiện quan trọng.
- **Debugging và Profiling:** SSMS cung cấp các công cụ để debug các thủ tục lưu trữ và xem hồ sơ thực hiện (execution profile) của các truy vấn SQL để tối ưu hiệu suất của chúng.

- **Tích hợp với Visual Studio:** SSMS tích hợp chặt chẽ với Visual Studio, cho phép nhà phát triển và quản trị viên dễ dàng làm việc với cả hai môi trường mà không cần chuyển đổi giữa chúng.
- **Hỗ trợ mở rộng:** SSMS hỗ trợ các công cụ mở rộng và tiện ích bổ sung từ cộng đồng, giúp tùy chỉnh và mở rộng chức năng của ứng dụng theo nhu cầu cụ thể.

Tóm lại, SQL Server Management Studio là một công cụ quan trọng và mạnh mẽ cho việc quản lý cơ sở dữ liệu SQL Server và phát triển ứng dụng liên quan đến dữ liệu trong môi trường Microsoft SQL Server.

CHƯƠNG 2: PHÂN TÍCH CHỨC NĂNG VÀ THIẾT KẾ HỆ THỐNG

2.1 Mô tả bài toán

Bài toán đặt ra là xây dựng một hệ thống phần mềm thi trắc nghiệm dành cho học sinh nhằm hỗ trợ việc tổ chức và tham gia các kỳ thi theo hình thức trắc nghiệm một cách hiệu quả và tiện lợi. phần mềm sẽ cung cấp các chức năng cần thiết cho học sinh, giáo viên, trưởng khoa và người quản trị hệ thống, bao gồm việc tạo đề thi, làm bài thi, chấm điểm tự động, duyệt đề thi, thống kê và quản lý người dùng.

Hệ thống cần đảm bảo khả năng xử lý nhiều người dùng truy cập đồng thời, lưu trữ và bảo mật dữ liệu một cách an toàn, và cung cấp trải nghiệm người dùng mượt mà với giao diện trực quan. Ngoài ra, việc tính toán điểm thi và cung cấp kết quả nhanh chóng là một trong những yêu cầu quan trọng của hệ thống.

2.2 Phân tích chức năng

2.2.1 Chức năng của học sinh

- **Đăng nhập**

Người dùng đăng nhập theo tài khoản đã tạo. Nếu đăng nhập sai, yêu cầu nhập lại.

Trong trường hợp người dùng quên mật khẩu, có thể sử dụng tính năng quên mật khẩu để có thể được cấp lại mật khẩu thông qua việc xác minh số điện thoại.

- **Kiểm tra thông tin**

Người dùng có thể kiểm tra các thông tin cá nhân như : Tên tài khoản, họ tên, email, số điện thoại.

- **Thay đổi mật khẩu**

Người dùng có thể nhập mật khẩu cũ và đổi mật khẩu mới nếu muốn đảm bảo an toàn hơn cho tài khoản của mình.

- **Lựa chọn đề thi.**

Người dùng có thể lựa chọn các đề thi của từng môn thi đã có để tham gia làm bài thi trắc nghiệm.

- **Xem kết quả bài thi**

Người dùng có thể xem được điểm số, số câu đúng, thời gian làm bài của bài thi sau khi ấn nút nộp bài.

- **Xem lại bài làm**

Người dùng có thể xem lại bài làm của mình bao gồm cả đáp án đúng và đáp án mình đã chọn.

- Kiểm tra lịch sử làm bài.

Người dùng có thể kiểm tra lại lịch sử các bài đã làm về: thời gian làm bài, số điểm, số câu làm đúng, mã đề.

- Xem bảng xếp hạng

Người dùng có thể xem được bảng xếp hạng các môn học, hoặc lựa chọn từng môn học riêng biệt để xem kết quả của những người dùng khác.

2.2.2 Chức năng của trưởng khoa

- Đăng nhập

Người dùng đăng nhập theo tài khoản đã tạo. Nếu đăng nhập sai, yêu cầu nhập lại.

- Duyệt đề thi

Người dùng đăng nhập tài khoản trưởng khoa để có thể thực hiện duyệt đề thi hoặc từ chối duyệt đề được gửi từ phía giáo viên.

- Quản lý phân công

Người dùng có thể phân công giáo viên dạy một hoặc nhiều môn, một môn có thể phân được nhiều giáo viên dạy.

- Quản lý đề thi

Người dùng đăng nhập tài khoản trưởng khoa có thể thực hiện thao tác thêm, sửa, xóa, tìm kiếm các đề thi trắc nghiệm đã được lưu vào chương trình.

- Quản lý môn học

Người dùng đăng nhập tài khoản trưởng khoa có thể thực hiện thao tác thêm, xóa các môn học đã được lưu vào chương trình.

- Thống kê và xuất báo cáo

Thống kê được số lượng môn học, điểm trung bình môn học, số giáo viên dạy môn học đó và xuất file excel.

2.2.3 Chức năng của giáo viên

- Đăng nhập

Người dùng đăng nhập theo tài khoản đã tạo. Nếu đăng nhập sai, yêu cầu nhập lại.

- Tạo đề thi

Người dùng đăng nhập tài khoản Teacher có thể thực hiện thao tác thêm đề thi trắc nghiệm được hoàn thành hoặc lưu tạm thời .

- Quản lý đề thi

Người dùng có thể tìm kiếm , cập nhật , xóa , chỉnh sửa đề thi trắc nghiệm được lưu vào chương trình .

Người dùng có thể hoàn thành tiếp đề thi đã được lưu tạm thời trước khi được duyệt đề thi.

➤ **Thống kê**

Thống kê được số lượng thí sinh thi, thời gian thi, môn thi, điểm thi của từng thí sinh.

2.2.4 Chức năng của quản trị viên

➤ **Đăng nhập**

Người dùng đăng nhập theo tài khoản đã tạo. Nếu đăng nhập sai, yêu cầu nhập lại.

➤ **Quản lý tài khoản người dùng**

Người dùng đăng nhập tài khoản admin có thể xem thông tin cá nhân của các tài khoản người dùng. Có thể chỉnh sửa, tìm kiếm và xóa thông tin của các tài khoản người dùng.

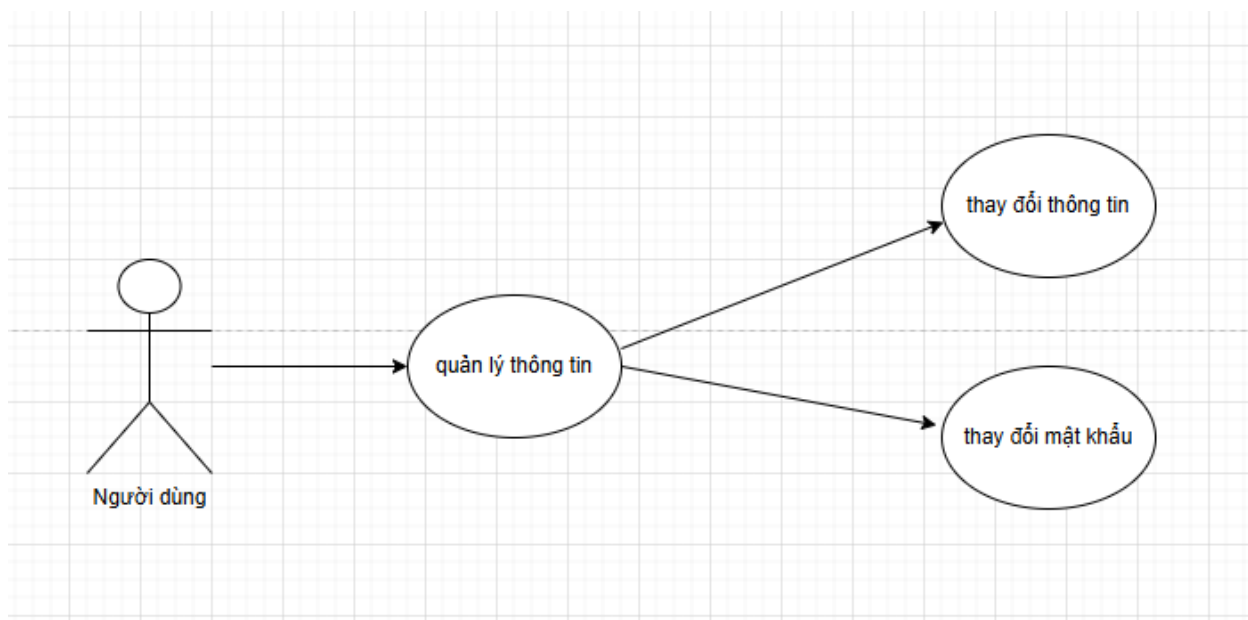
➤ **Thống kê và xuất báo cáo**

Người dùng đăng nhập tài khoản admin có thể thực hiện thao tác xem báo cáo về số lượng đề thi có trong hệ thống của từng môn học, số lượng người dùng và số lượt làm bài thi của từng môn học.

2.3 Biểu đồ UseCase

2.3.1 Biểu đồ UseCase của sinh viên

➤ **User case quản lý thông tin**



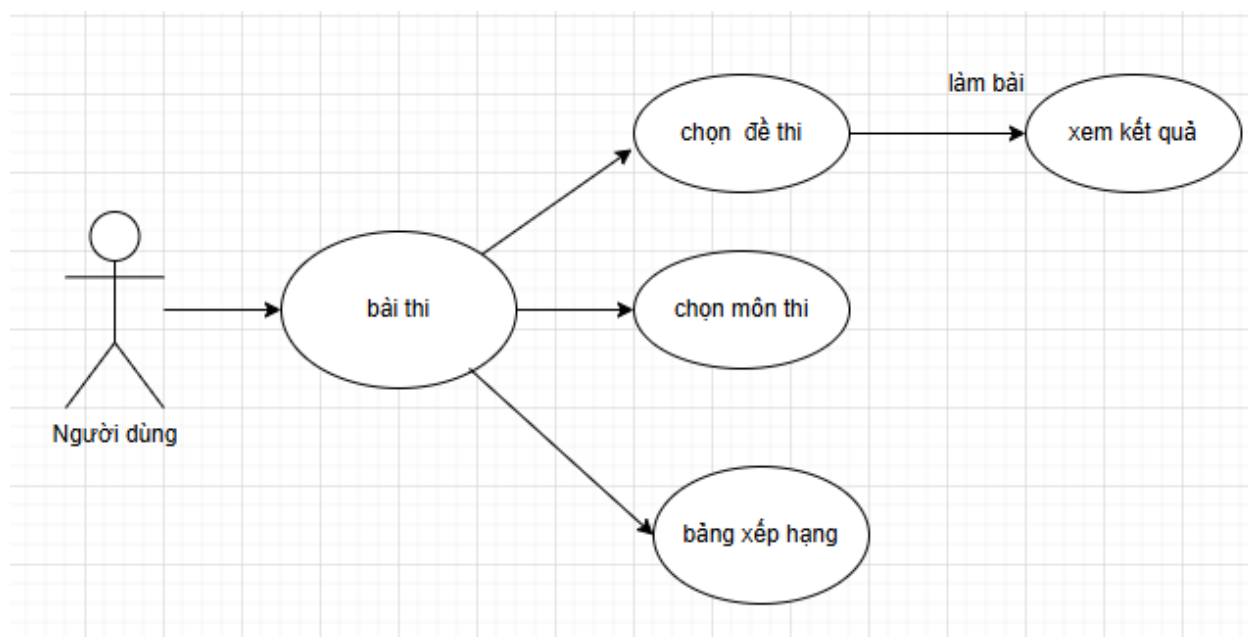
Hình 2. 1 Biểu đồ Use Case quản lý thông tin

Mô tả Use Case quản lý thông tin :

Bảng 2. 1 Đặc tả Use Case quản lý thông tin

Tác nhân	Học sinh
Mô tả	Cho phép xem thông tin cá nhân
Tiền điều kiện	Sinh viên đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Sinh viên chọn mục ‘ Thông tin ‘ trên hệ thống 2. Hệ thống hiển thị hết thông tin người dùng 3. Nếu muốn thay đổi thông tin , hệ thống sẽ kiểm tra và cập nhật dữ liệu mới cho sinh viên

➤ User case bài thi



Hình 2. 2 Biểu đồ Use Case bài thi

Mô tả Use Case bài thi :

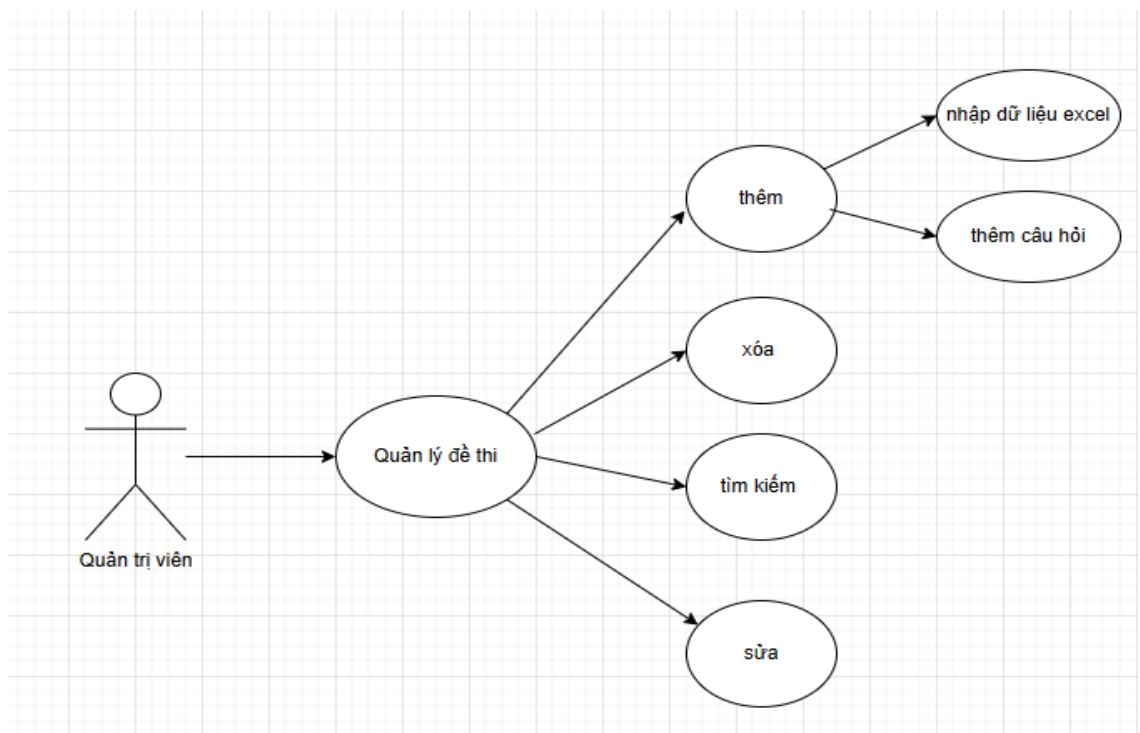
Bảng 2. 2 Đặc tả Use Case bài thi

Tác nhân	Học sinh
Mô tả	Cho phép làm bài thi
Tiền điều kiện	Sinh viên đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Sinh viên chọn mục ‘ Vào thi ‘ trên hệ thống

	2. Hệ thống hiển thị môn thi cần làm 3. Chọn một môn để thi hệ thống sẽ hiển thị tất cả bài thi được giao cho sinh viên 4. Sinh viên truy cập vào bài thi đã được giao 5. Hoàn thành bài thi và nộp bài 6. Sinh viên sẽ được nhìn thấy kết quả bài thi đã làm và xem bảng xếp hạng
--	--

2.3.2 Biểu đồ UseCase của người quản trị

➤ Use case Quản lý đề Thi



Hình 2. 3 Biểu đồ Use Case quản lý đề thi

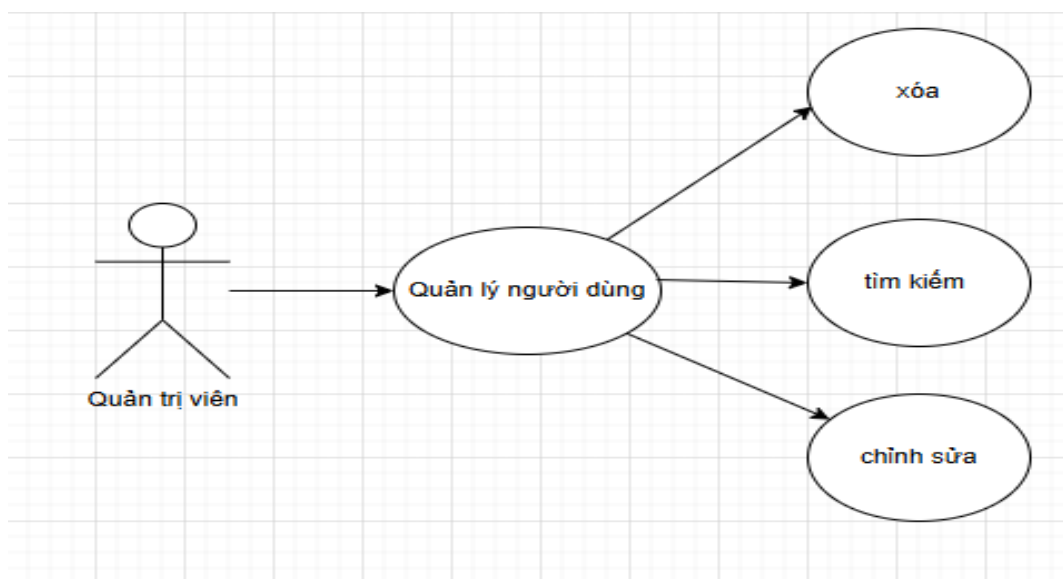
Mô tả Use case quản lý đề thi :

Bảng 2. 3 Use case Quản lý đề thi

Tác nhân	Quản trị viên
Mô tả	Cho phép Quản trị viên chọn các chức năng thêm, xóa ,tìm kiếm,sửa bài thi
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống

Luồng xử lý chính	1. Quản trị viên chọn mục ‘ Quản lý đề thi ‘ trên hệ thống 2. Hệ thống hiển thị thông tin đề thi , danh sách đề thi 3. Nếu chọn thêm đề quản trị viên có thể chọn môn học cần thêm và có thể thêm đề thủ công hoặc import file 4. Nếu chọn xóa đề chọn môn cần xóa 5. Nếu chọn tìm kiếm nhập môn cần tìm
-------------------	--

➤ Use case quản lý người dùng



Hình 2. 4 Biểu đồ Use Case quản lý người dùng

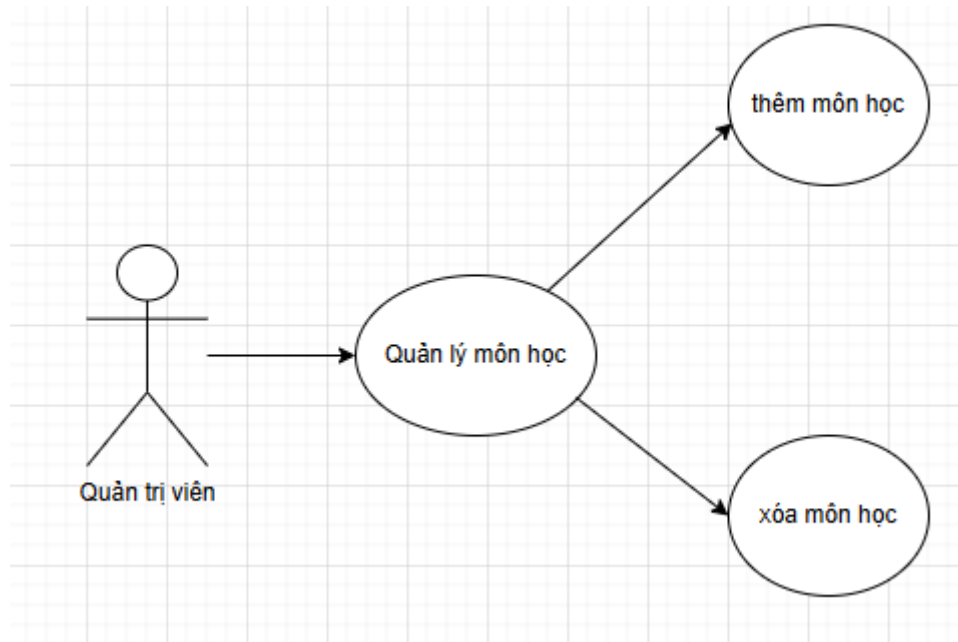
Mô tả Use case quản lý người dùng :

Bảng 2. 4 Use case Quản lý đề thi

Tác nhân	Quản trị viên
Mô tả	Cho phép Quản trị viên chọn các chức năng sửa, xóa ,tìm kiếm
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Quản trị viên chọn mục ‘ Quản lý người dùng ‘ trên hệ thống 2. Hệ thống hiển thị thông tin người dùng 3. Nếu chọn chỉnh sửa quản trị viên có thể sửa thông tin người dùng để cập nhật lại thông tin mới 4. Nếu chọn xóa đề chọn tài khoản người dùng cần xóa

5. Nếu chọn tìm kiếm nhập tên tài khoản cần tìm kiếm

➤ Use case quản lý môn học



Hình 2. 5 Biểu đồ Use Case quản lý môn học

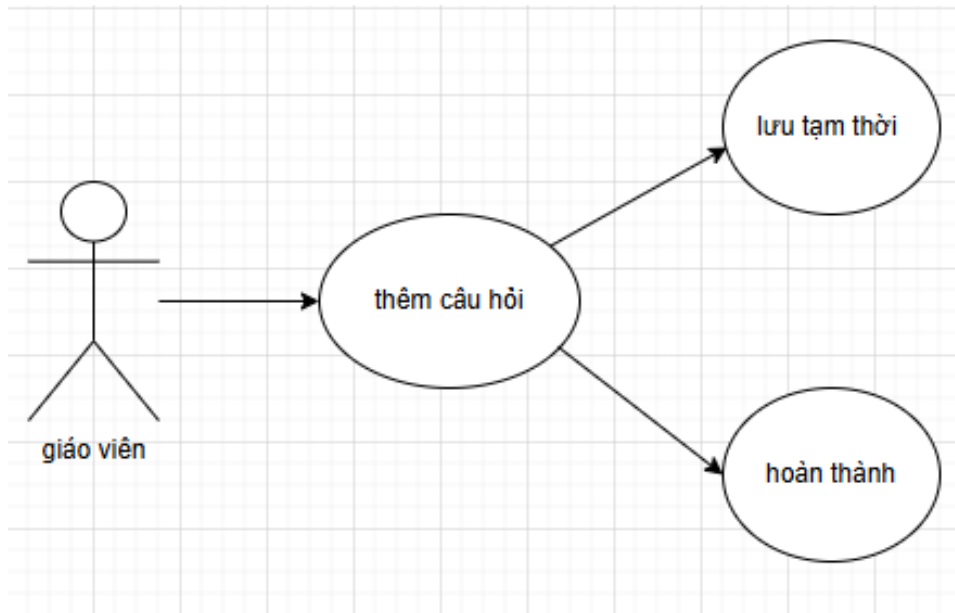
Mô tả Use Case quản lý môn học :

Bảng 2. 5 Use Case quản lý môn học

Tác nhân	Quản trị viên
Mô tả	Cho phép Quản trị viên chọn các xóa , thêm
Tiền điều kiện	Quản trị viên đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Quản trị viên chọn mục ‘ Quản lý môn học ‘ trên hệ thống 2. Hệ thống hiển thị thông tin môn học 3. Nếu chọn thêm sẽ thêm môn học cần thêm 4. Nếu chọn xóa sẽ xóa môn học cần xóa

2.3.3 Biểu đồ UseCase của giáo viên

➤ Use case tạo đề thi



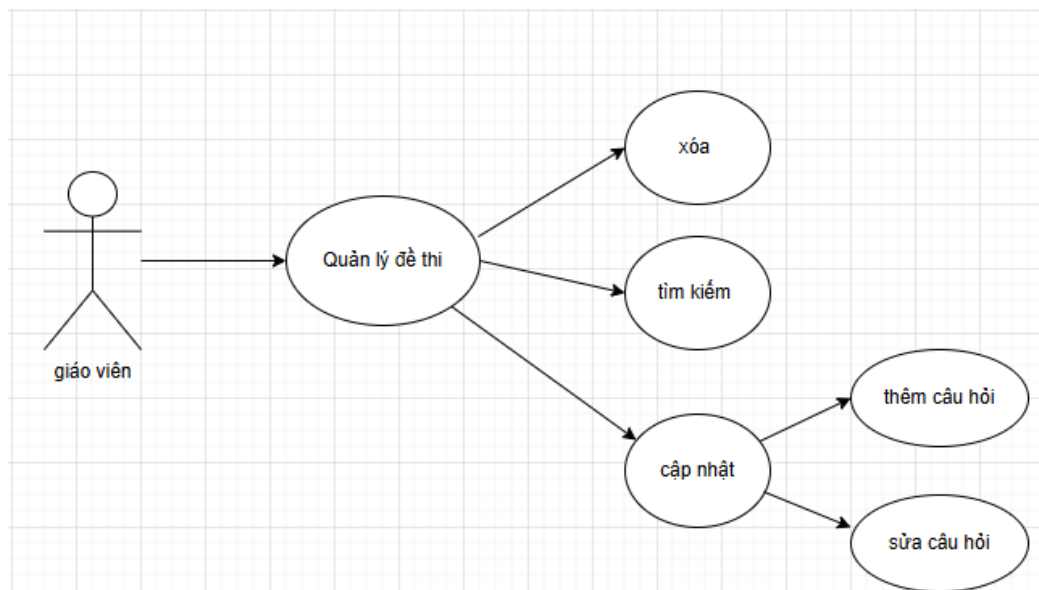
Hình 2. 6 Biểu đồ Use Case tạo đề thi

Mô tả Use Case tạo đề thi :

Bảng 2. 6 Use Case tạo đề thi

Tác nhân	Giáo viên
Mô tả	Cho phép giáo viên chọn thêm câu hỏi
Tiền điều kiện	Giáo viên đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Giáo viên chọn mục ‘ Thêm câu hỏi ‘ trên hệ thống 2. Hệ thống hiển thị mục để thêm câu hỏi 3. Khi câu hỏi được thêm giáo viên có thể lưu tạm thời nếu chưa hoàn thành đề thi 4. Khi hoàn thành đề thi đề thi sẽ được chuyển qua phía trưởng khoa để duyệt đề

➤ Use case quản lý đề thi



Hình 2. 7 Biểu đồ Use Case quản lý đề thi

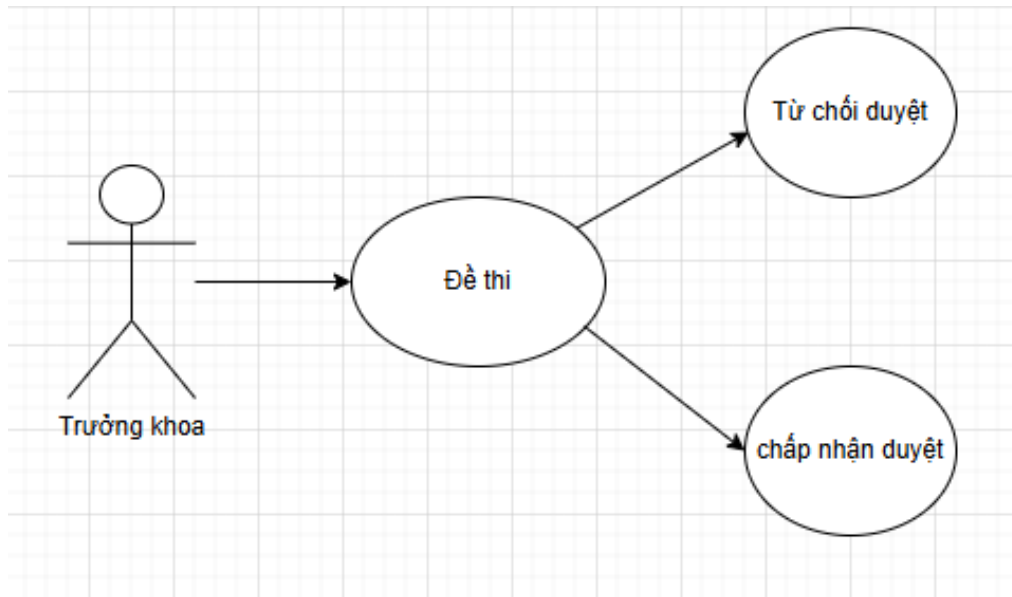
Mô tả Use case quản lý đề thi :

Bảng 2. 7 Use case quản lý đề thi

Tác nhân	Giáo viên
Mô tả	Cho phép giáo viên chọn xóa , tìm kiếm , cập nhật
Tiền điều kiện	Giáo viên đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Giáo viên chọn mục ‘ Quản lý đề thi ‘ trên hệ thống 2. Hệ thống hiển thị các đề thi giáo viên đã làm 3. Nếu chọn xóa giáo viên sẽ chọn đề cần xóa 4. Nếu chọn tìm kiếm giáo viên có thể kiểm tra bằng tên đề 5. Khi cập nhật đề thi nếu hoàn thành đề thi sẽ được chuyển qua phía trưởng khoa để duyệt

2.4.4 Biểu đồ UseCase của trường khoa

➤ Use case Duyệt đề



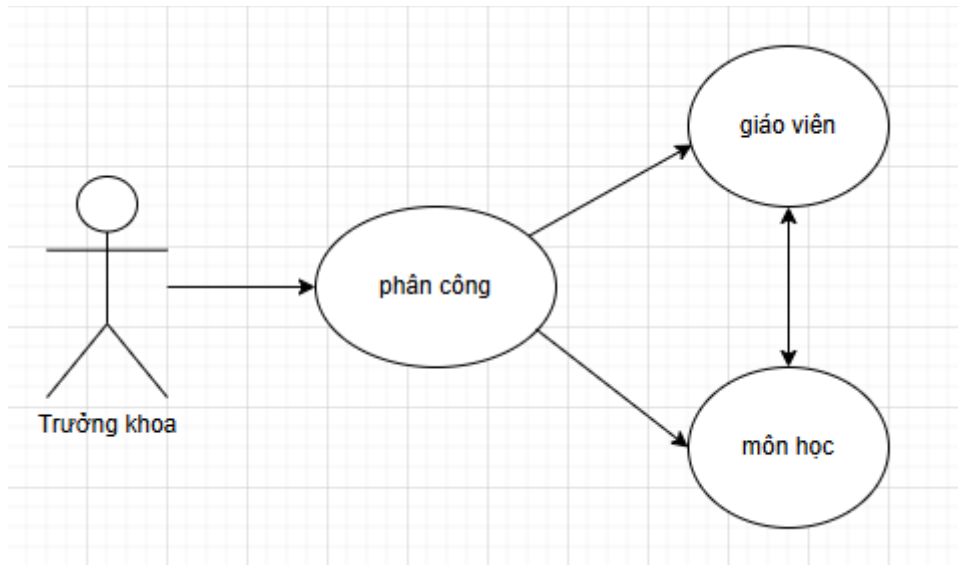
Hình 2. 8 Biểu đồ Use Case Duyệt đề

Mô tả Use case duyệt đề :

Bảng 2. 8 Use case duyệt đề

Tác nhân	Trưởng khoa
Mô tả	Cho phép Trưởng khoa duyệt đề thi
Tiền điều kiện	Trưởng khoa đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Trưởng khoa chọn mục ‘ Duyệt đề thi ‘ trên hệ thống 2. Hệ thống hiển thị các đề thi giáo viên đã gửi qua đề duyệt 3. Nếu chọn duyệt đề hệ thống sẽ ghi nhận đề đã được chấp nhận và chuyển đề lên danh sách đề thi của sinh viên 4. Nếu chọn từ chối hệ thống sẽ ghi nhận từ chối đề của giáo viên gửi và sẽ thông báo lại cho giáo viên

➤ Use case quản lý phân công



Hình 2. 9 Biểu đồ Use Case quản lý phân công

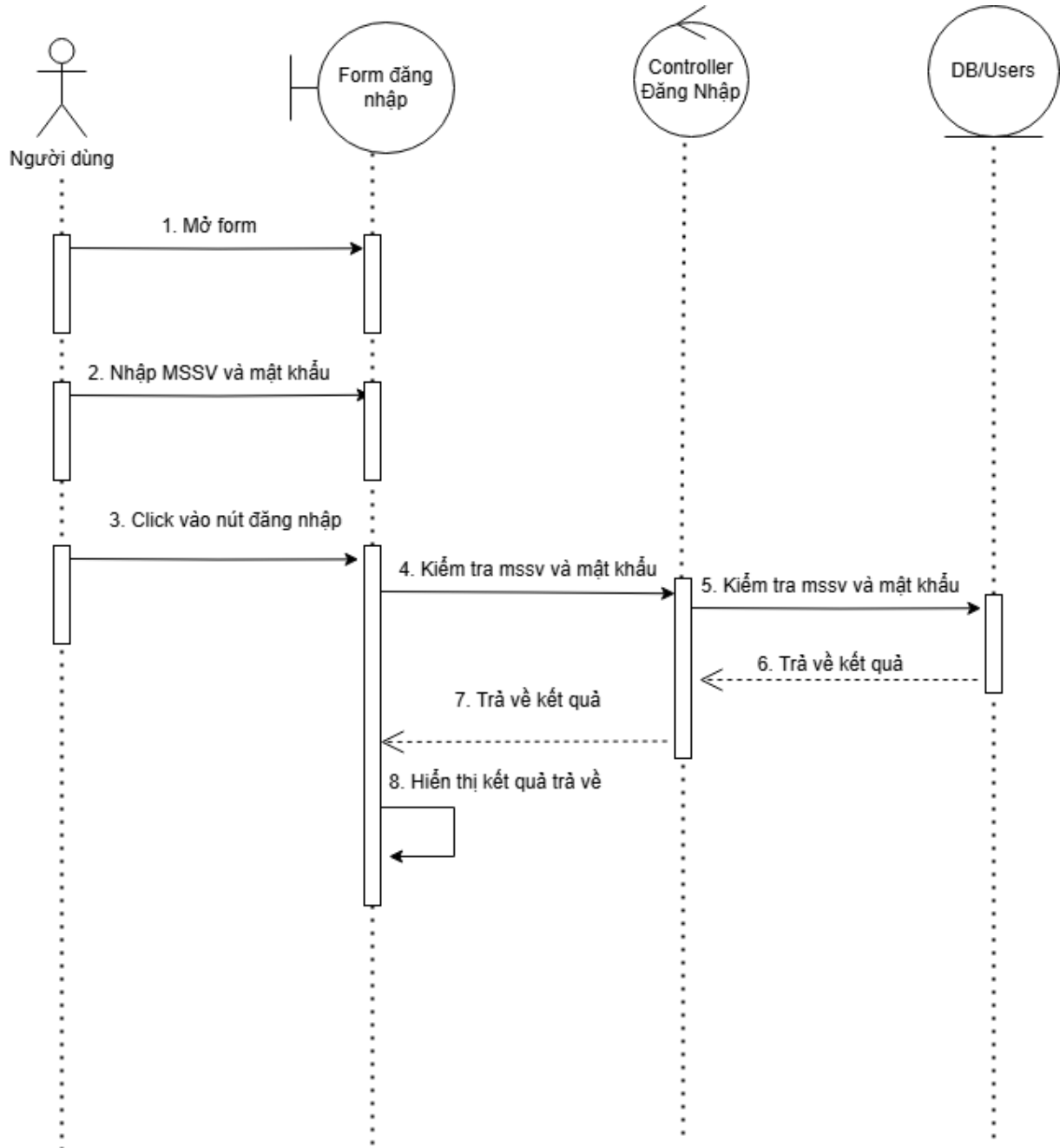
Mô tả Use case quản lý phân công :

Bảng 2. 9 Use case quản lý phân công

Tác nhân	Trưởng khoa
Mô tả	Cho phép Trưởng khoa phân công giáo viên
Tiền điều kiện	Trưởng khoa đã đăng nhập vào hệ thống
Luồng xử lý chính	1. Trưởng khoa chọn mục ‘ quản lý phân công ‘ trên hệ thống 2. Hệ thống hiển thị danh sách giáo viên và môn học 3. Trưởng khoa sẽ chọn một giáo viên được phân công vào một hoặc nhiều môn học cần dạy 4. Thông tin danh sách giáo viên được phân công sẽ được hiển thị lên hệ thống

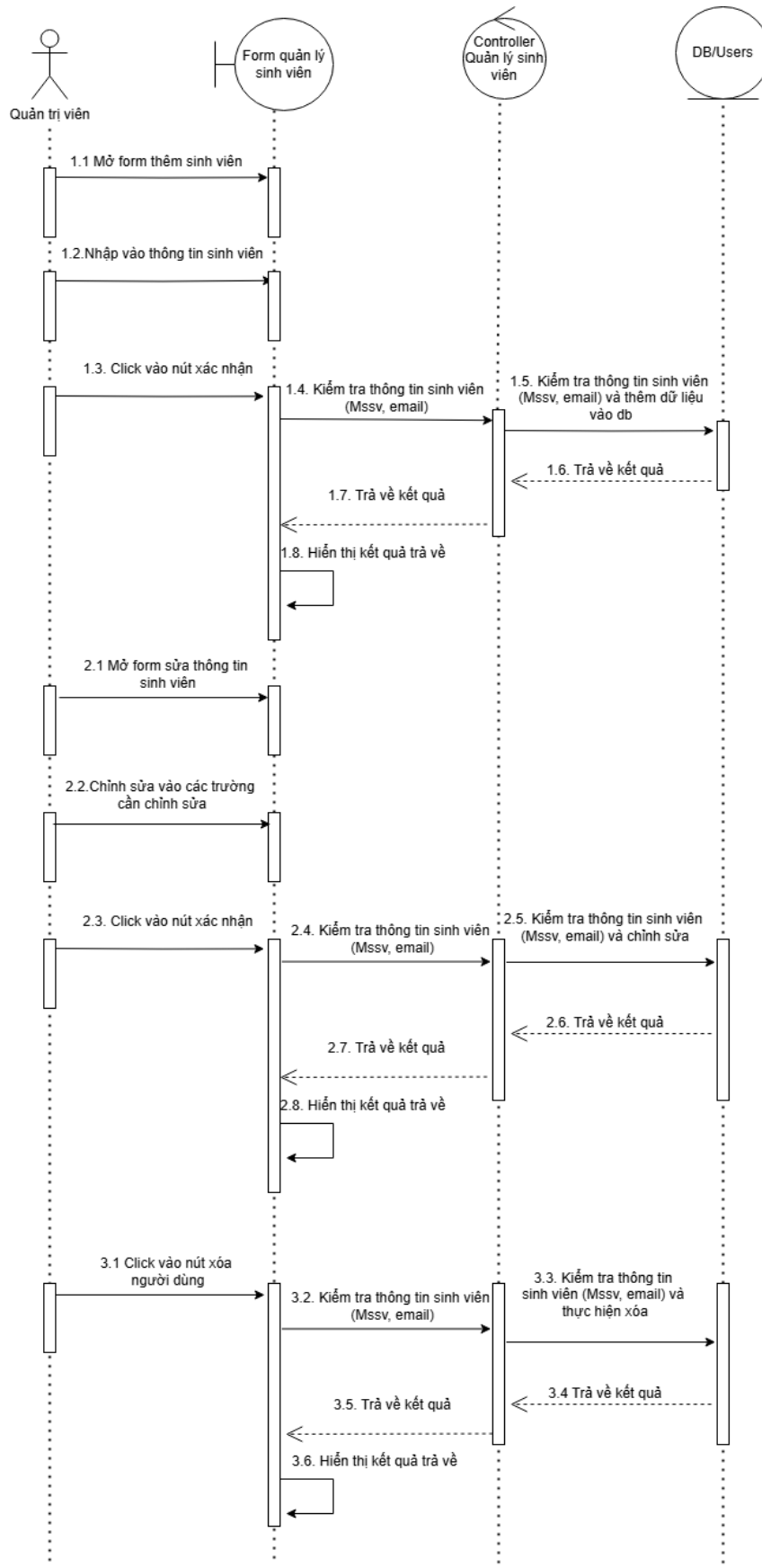
2.5 Sơ đồ tuần tự

2.5.1 Biểu đồ tuần tự đăng nhập



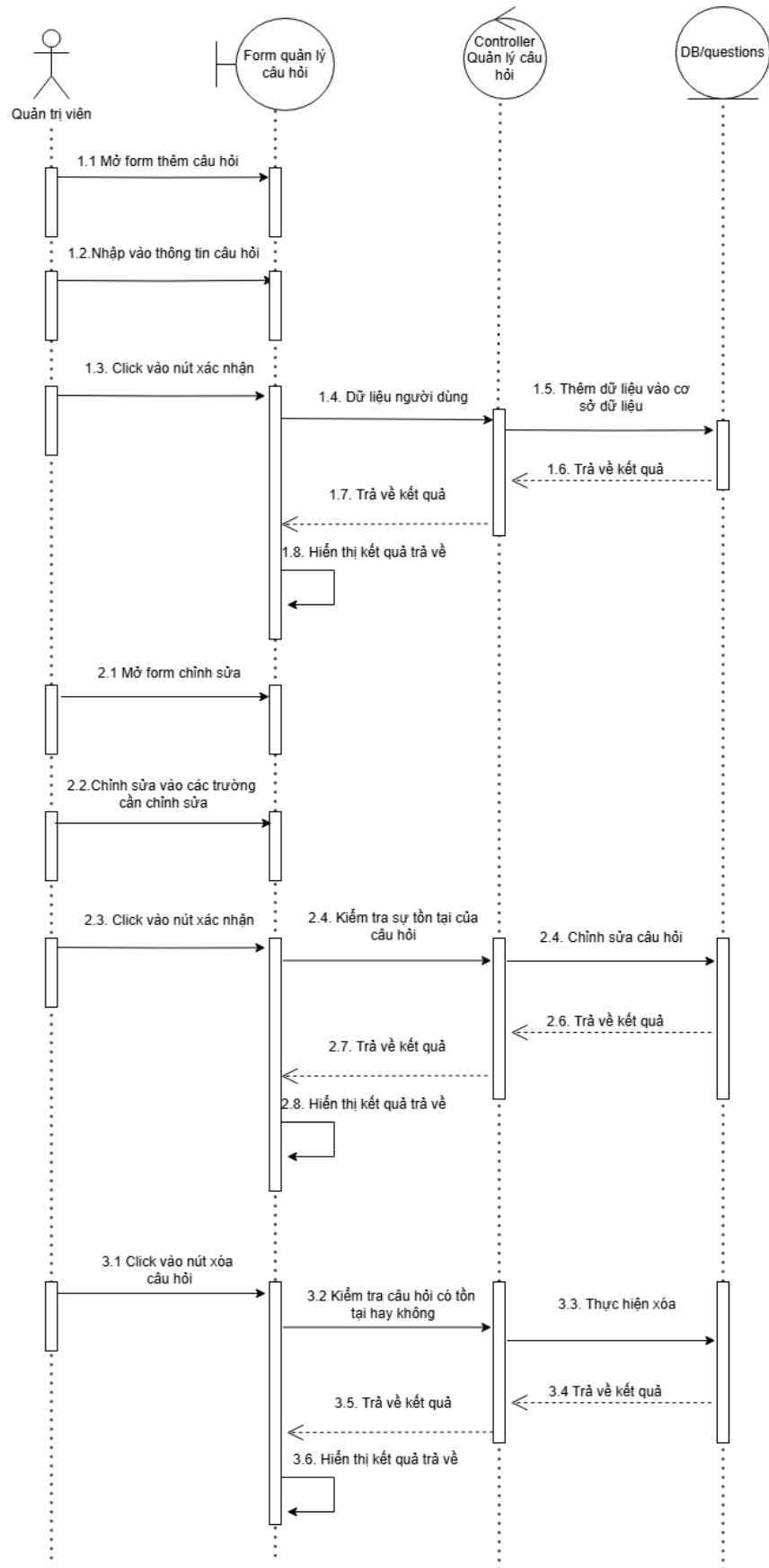
Hình 2. 10 Sơ đồ tuần tự đăng nhập

2.5.2 Biểu đồ tuần tự quản lý người dùng



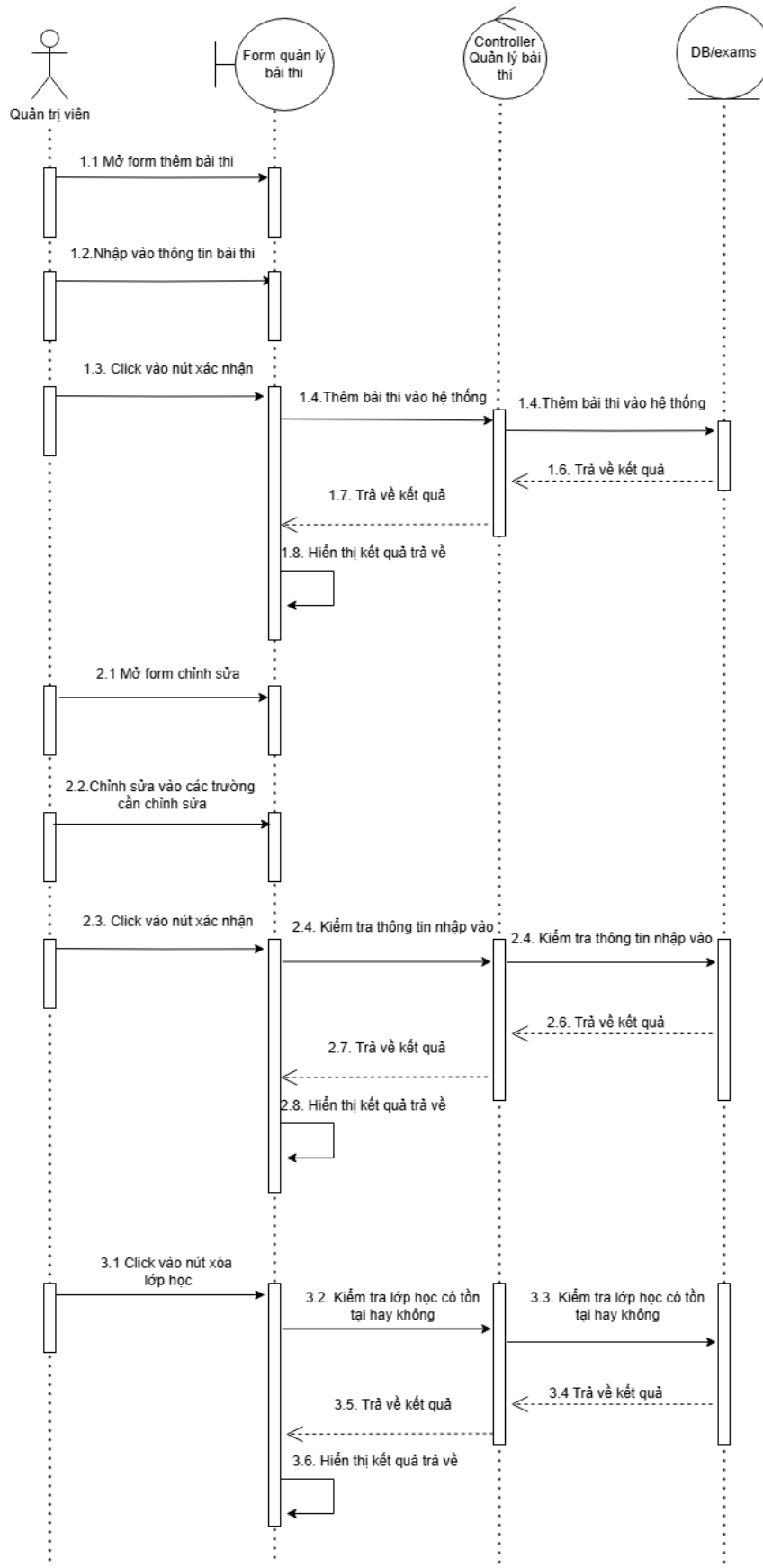
Hình 2. 11 Sơ đồ tuần tự quản lý người dùng

2.5.3 Biểu đồ tuần tự quản lý câu hỏi



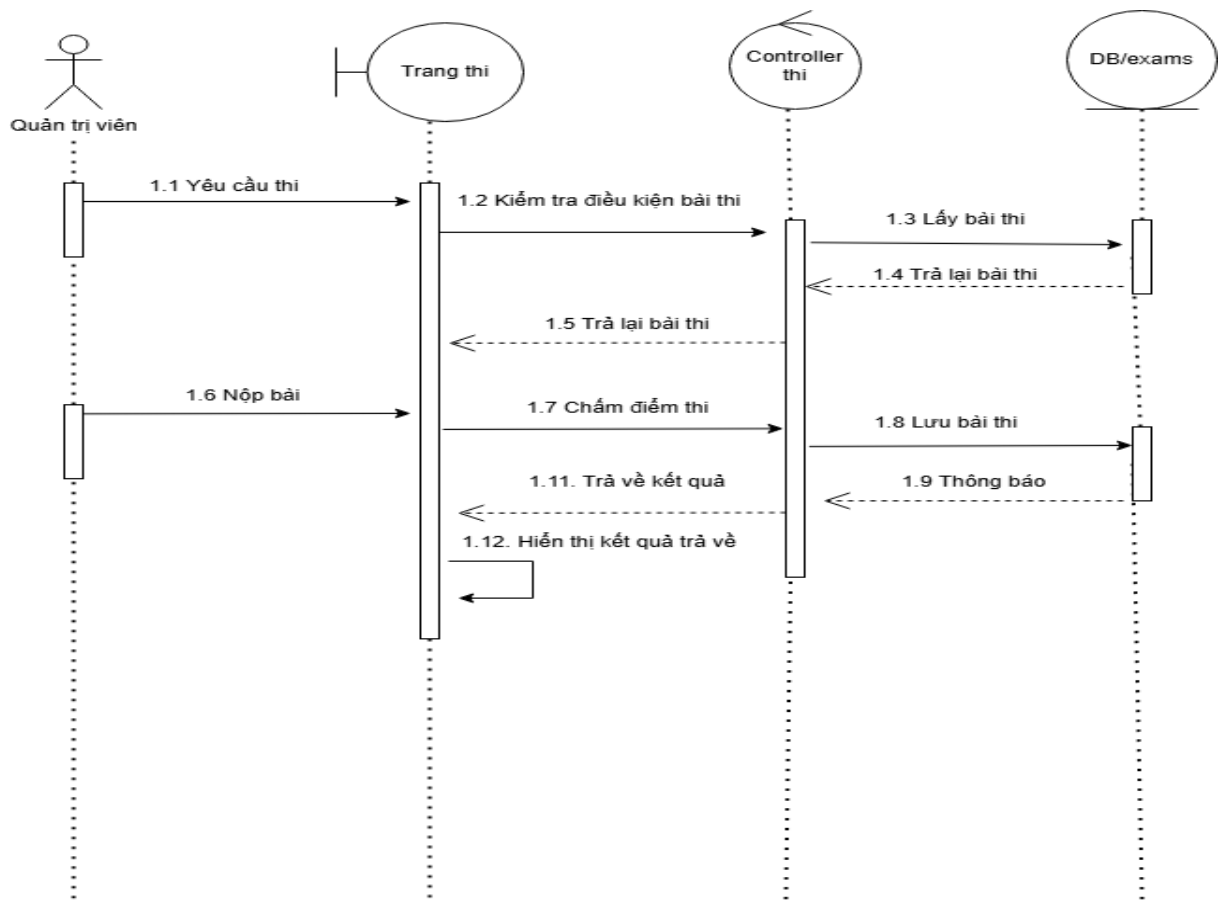
Hình 2. 12 Sơ đồ tuần tự quản lý câu hỏi

2.5.4 Biểu đồ tuần tự quản lý bài thi



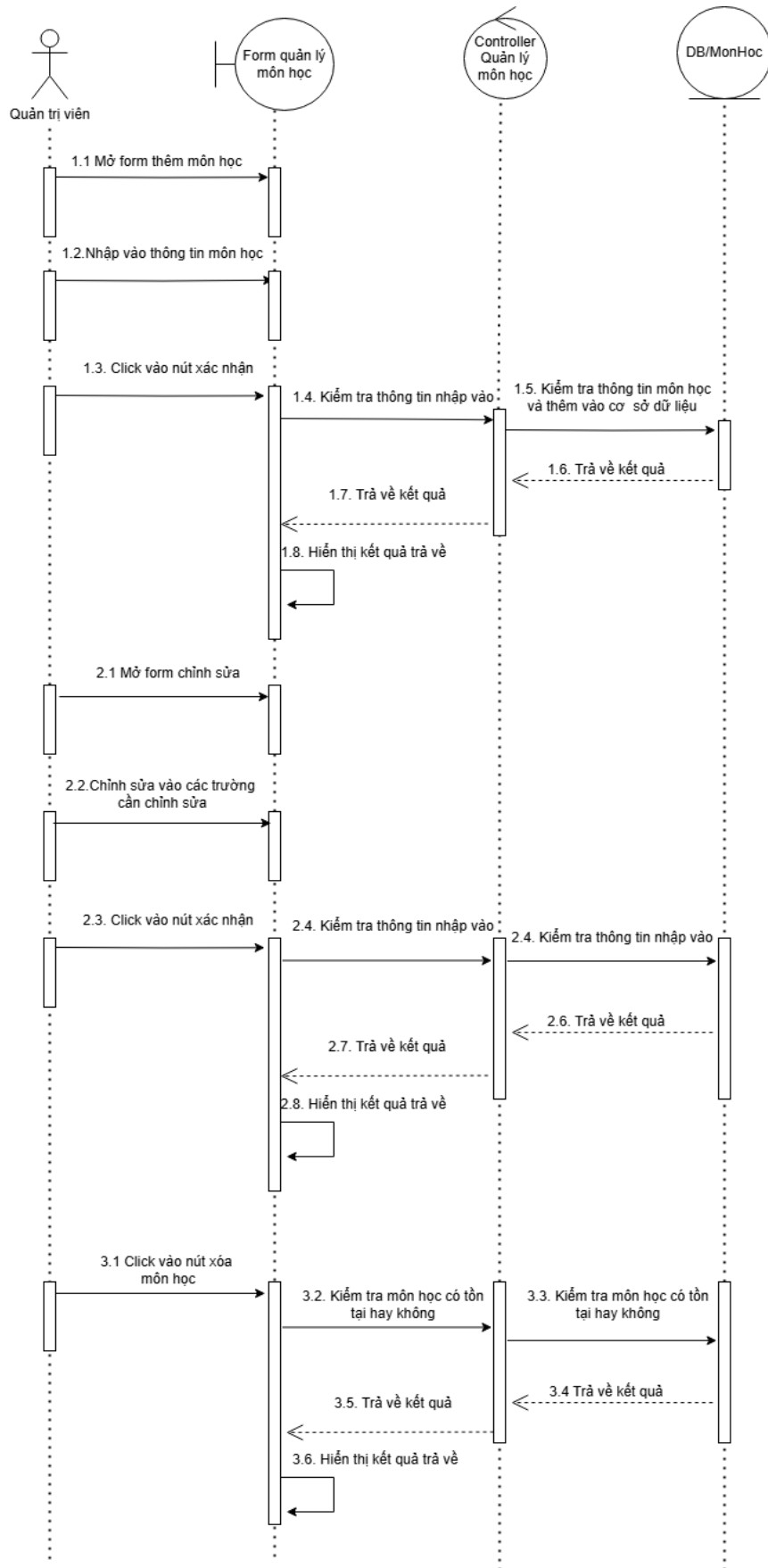
Hình 2. 13 Sơ đồ tuần tự quản lý bài thi

2.5.5 Biểu đồ tuần tự thi



Hình 2. 14 Sơ đồ tuần tự thi

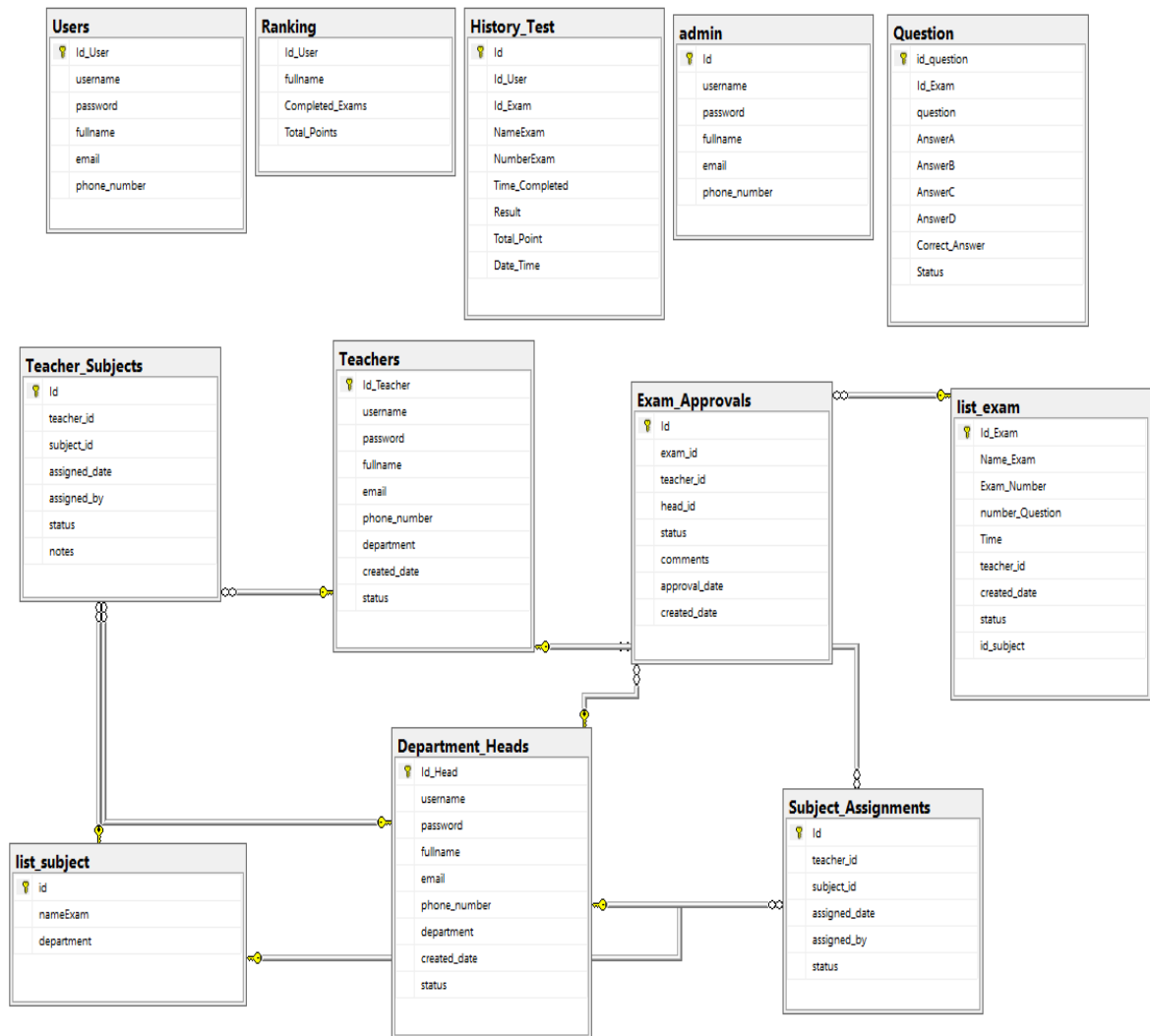
2.5.6 Biểu đồ tuần tự quản lý môn học



Hình 2. 15 Sơ đồ tuần tự quản lý môn học

2.6 Thiết kế cơ sở dữ liệu

2.6.1 Sơ đồ quan hệ khóa



Hình 2. 16 Sơ đồ quan hệ khóa

2.6.2 Danh mục các bảng

Bảng admin bảng này dùng để lưu trữ thông tin chi tiết của quản trị viên

Bảng 2. 10 Bảng dữ liệu Admin

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id_Admin	INTEGER	Mã định danh quản trị viên
2	Username	VARCHAR	Tên đăng nhập
3	Password	VARCHAR	Mật khẩu đã được mã hóa
4	Fullname	VARCHAR	Họ và tên đầy đủ
5	Phone_number	VARCHAR	Số điện thoại

Bảng Users bảng này dùng để lưu trữ thông tin chi tiết của học sinh

Bảng 2. 11 Bảng dữ liệu User

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id_User	INT	Mã định danh học sinh
2	Username	VARCHAR	Tên đăng nhập
3	Password	VARCHAR	Mật khẩu đã được mã hóa
4	Fullname	VARCHAR	Họ và tên đầy đủ
5	Phone_number	VARCHAR	Số điện thoại

Bảng Department_Heads bảng này dùng để lưu trữ thông tin chi tiết của trưởng khoa

Bảng 2. 12 Bảng dữ liệu Department_Heads

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id_Head	INT	Mã định danh trưởng khoa
2	Username	VARCHAR	Tên đăng nhập
3	Password	VARCHAR	Mật khẩu đã được mã hóa
4	Fullname	VARCHAR	Họ và tên đầy đủ
5	Phone_number	VARCHAR	Số điện thoại
6	Department	VARCHAR	Bộ phận
7	created_date	TIMESTAMP	Ngày tạo dữ liệu
8	status	INT	Trạng thái hoạt động

Bảng Teachers bảng này dùng để lưu trữ thông tin chi tiết của giáo viên

Bảng 2. 13 Bảng dữ liệu Teachers

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id_Teachers	INT	Mã định danh giáo viên
2	Username	VARCHAR	Tên đăng nhập
3	Password	VARCHAR	Mật khẩu đã được mã hóa
4	Fullname	VARCHAR	Họ và tên đầy đủ
5	Phone_number	VARCHAR	Số điện thoại
6	Department	VARCHAR	Bộ phận
7	created_date	TIMESTAMP	Ngày tạo dữ liệu
8	status	INT	Trạng thái hoạt động

Bảng Exam_Appovals bảng dùng để yêu cầu duyệt đề thi từ trưởng khoa

Bảng 2. 14 Bảng dữ liệu Exam_Appovals

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id	INT	Số thứ tự
2	Exam_id	INT	Mã đề thi
3	Teacher_id	INT	Mã định danh giáo viên
4	Head_id	INT	Mã định danh trưởng khoa
5	Status	VARCHAR	Trạng thái đề thi
6	Comments	VARCHAR	Nhận xét của trưởng khoa
7	approval_date	TIMESTAMP	Ngày duyệt đề
8	created_date	TIMESTAMP	Ngày gửi duyệt đề

Bảng Ranking bảng này dùng để xếp hạng thí sinh

Bảng 2. 15 Bảng dữ liệu Ranking

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id	INT	Số thứ tự
2	Id_User	INT	Liên kết đến Users, xác định học sinh
3	Fullname	VARCHAR	Tên của học sinh
4	Completed_Exams	INT	Số bài thi đã hoàn thành
5	Total_Points	VARCHAR	Tổng điểm tích lũy

Bảng History_Test bảng này dùng để xem lịch sử làm bài

Bảng 2. 16 Bảng dữ liệu History_Test

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id	INT	Khóa chính của từng lần làm bài
2	Id_User	INT	Mã định danh học sinh
3	Id_Exam	INT	Mã Đề thi
4	NameExam	VARCHAR	Tên bài thi
5	NumberExam	INT	Số thứ tự bài thi
6	Time_Completed	TIMESTAMP	Thời gian hoàn thành
7	Result	VARCHAR	Điểm đạt được
8	Total_Point	VARCHAR	Tổng điểm bài thi
9	Date_Time	TIMESTAMP	Ngày giờ nộp bài

Bảng Question bảng này dùng để lưu trữ ngân hàng câu hỏi

Bảng 2. 17 Bảng dữ liệu Question

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id_question	INT	Mã định danh câu hỏi
2	Id_Exam	INT	Mã đề thi
3	Question	VARCHAR	Nội dung câu hỏi
4	AnswerA	VARCHAR	Đáp án A
5	AnswerB	VARCHAR	Đáp án B
6	AnswerC	VARCHAR	Đáp án C
7	AnswerD	VARCHAR	Đáp án D
8	Correct_Answer	VARCHAR	Đáp án đúng
9	Status	INT	Trạng thái

Bảng List_exam bảng này hiển thị danh sách đề thi

Bảng 2. 18 Bảng dữ liệu List_exam

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id	INT	Khóa chính
2	Exam_id	INT	Mã đề thi để liên kết
3	Teacher_id	INT	Mã giáo viên để gửi đề duyệt
4	Head_id	INT	Mã trưởng khoa để duyệt đề
5	Status	VARCHAR	Trạng thái đề thi
6	Comments	VARCHAR	Nhận xét của trưởng khoa
7	Approval_date	TIMESTAMP	Ngày duyệt đề
8	Created_date	TIMESTAMP	Ngày gửi đề

Bảng Teacher_Subjects bảng này liên kết giáo viên với môn học

Bảng 2. 19 Bảng dữ liệu Teacher_Subjects

TT	Trường	Kiểu dữ liệu	Mô tả
1	Id	INT	Khóa chính
2	Teacher_id	INT	Mã giáo viên được phân công môn học
3	Subject_id	INT	Mã môn học phân công
4	Assigned_date	TIMESTAMP	Ngày phân công

5	Assigned_by	VARCHAR	Trưởng khoa sẽ phân công giáo viên
6	Status	VARCHAR	Trạng thái giáo viên dạy hay đã ngừng
7	notes	VARCHAR	Ghi chú

CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM ĐỀ TÀI

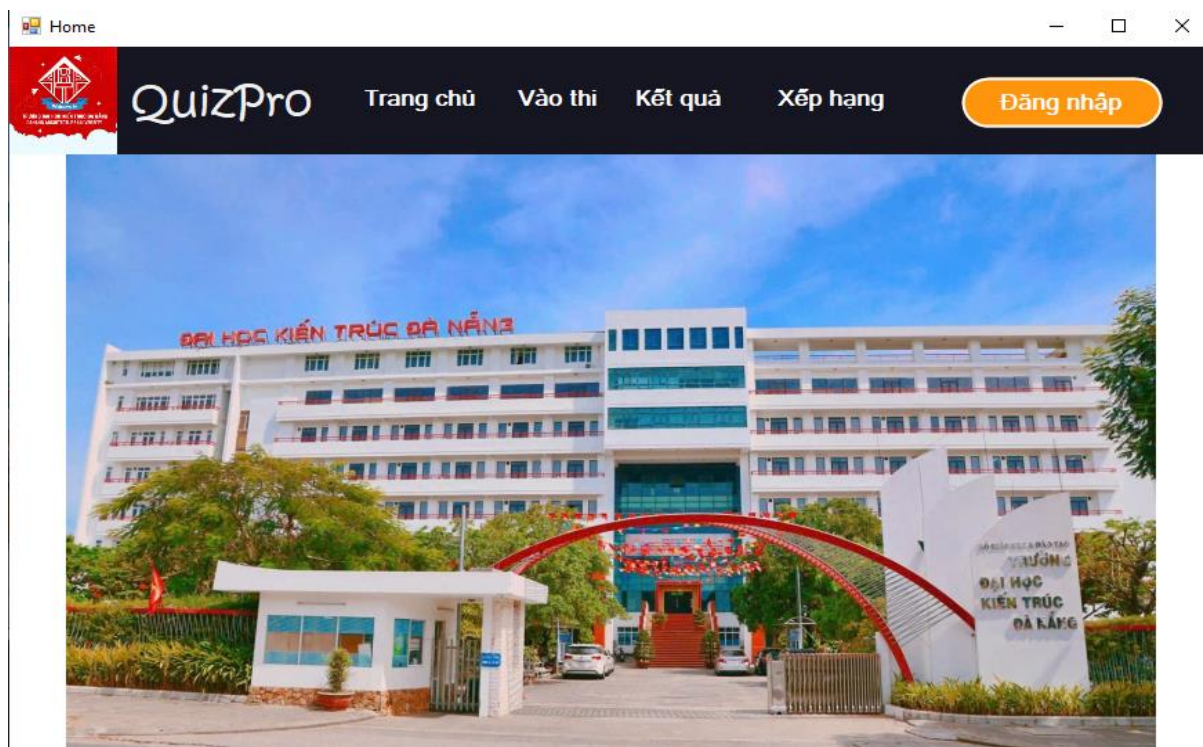
3.1 Môi trường

- ❖ Đề tài được phát triển trên môi trường phát triển:
 - Vi xử lý (processor): Processor 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
 - Ram: 16,0 GB
 - Storage 238 GB SSD Colorful SL500 256GB, 238 GB SSD NVMe Lexar SSD NM620 256GB
 - Graphics Card NVIDIA GeForce MX330 (2 GB), Intel(R) Iris(R) Xe Graphics (128 MB)
- ❖ Công cụ lập trình
 - Phần mềm viết code: Visual Studio
 - Cơ sở dữ liệu: SQL Server Management Studio
 - Ngôn ngữ lập trình: C#(Backend), Windows Forms(FrontEnd)

3.2 Kết quả giao diện

3.1.1 Giao diện trang chủ

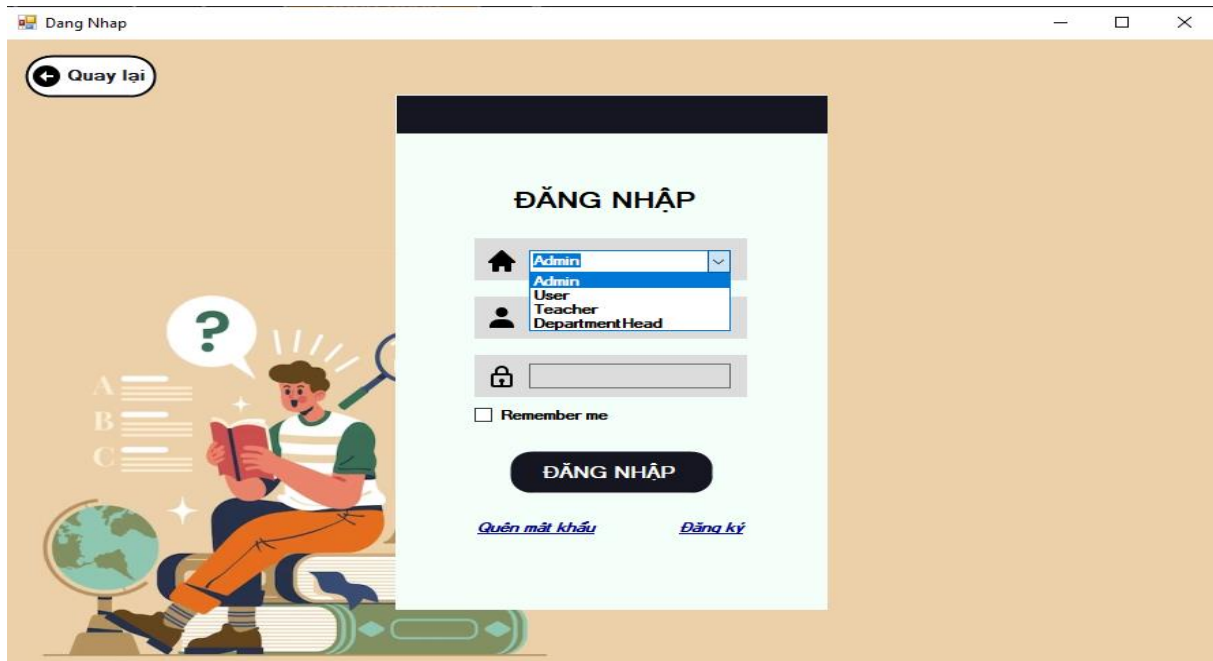
Giao diện tổng quát khi mở phần mềm.



Hình 3. 1 Giao diện trang chủ

3.1.2 Giao diện trang đăng nhập

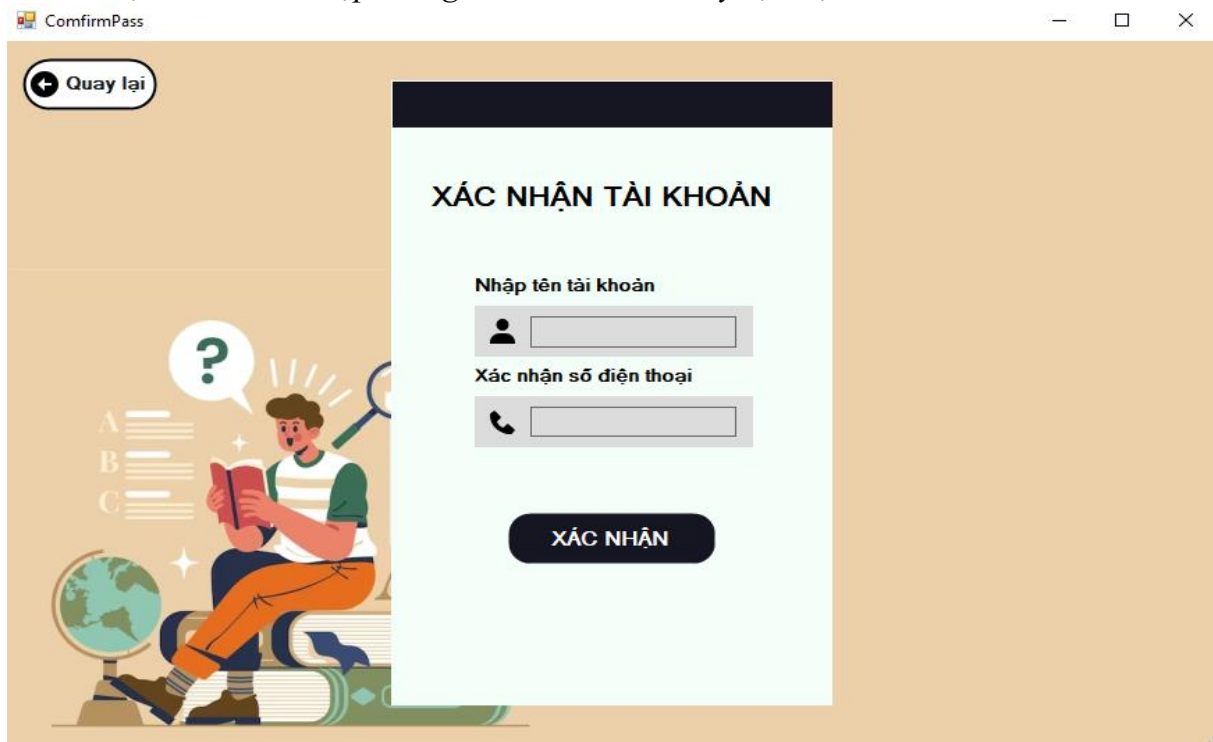
User ,admin , teacher , head cần nhập đủ thông tin trước khi đăng nhập vào hệ thống.



Hình 3. 2 Giao diện trang đăng nhập

3.1.3 Giao diện trang quên mật khẩu

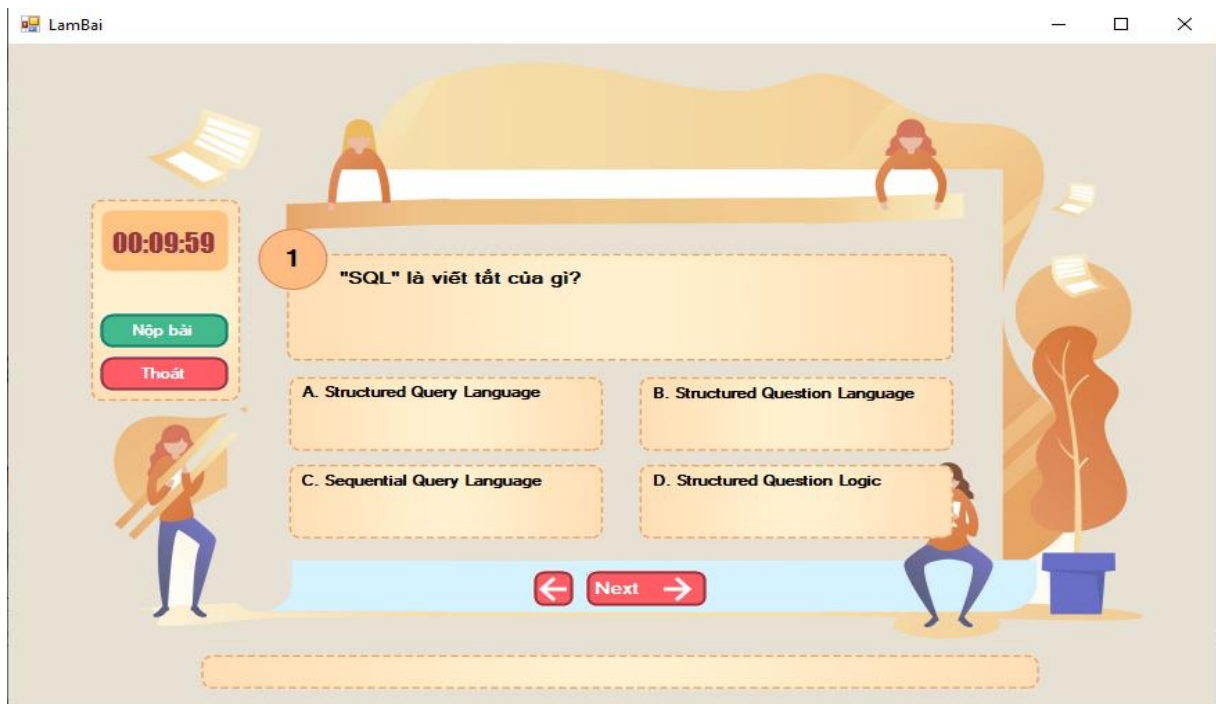
Học sinh cần nhập thông tin tài khoản để lấy lại mật khẩu.



Hình 3. 3 giao diện trang quên mật khẩu

3.1.4 Giao diện trang làm đề thi

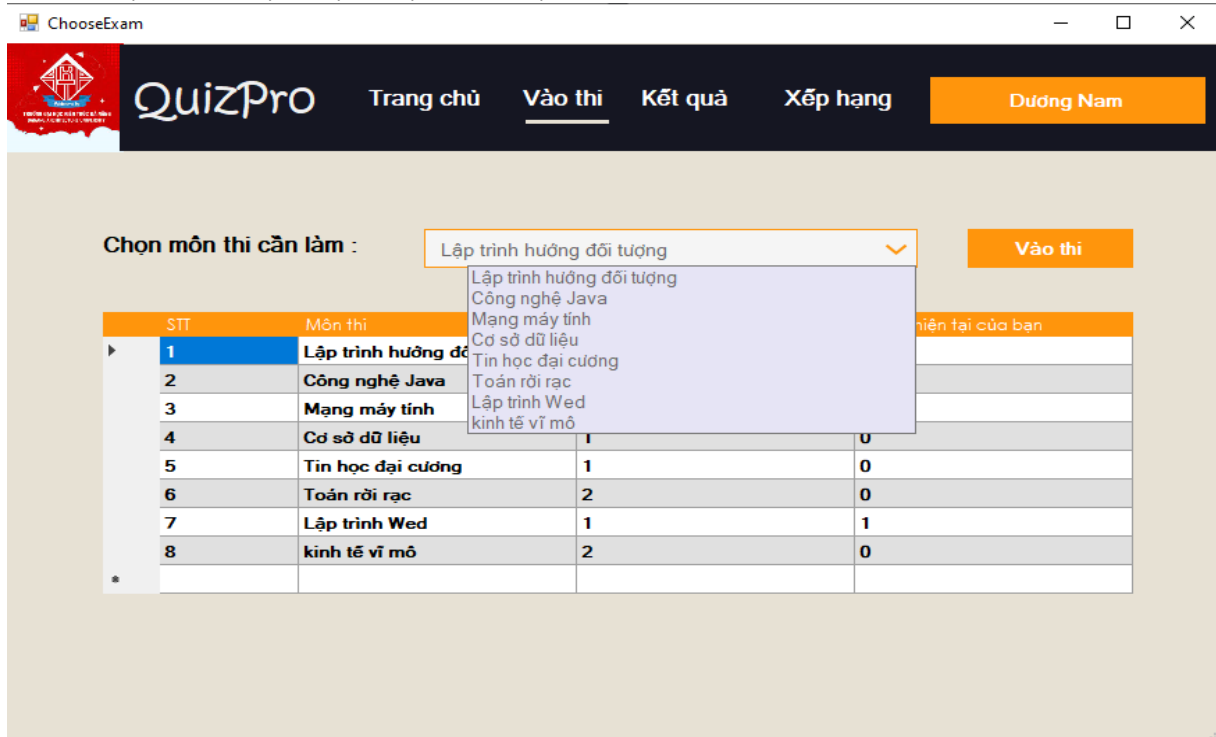
Học sinh thực hiện làm đề thi đã được giao.



Hình 3. 4 Giao diện trang làm đề thi

3.1.5 Giao diện trang chọn môn thi

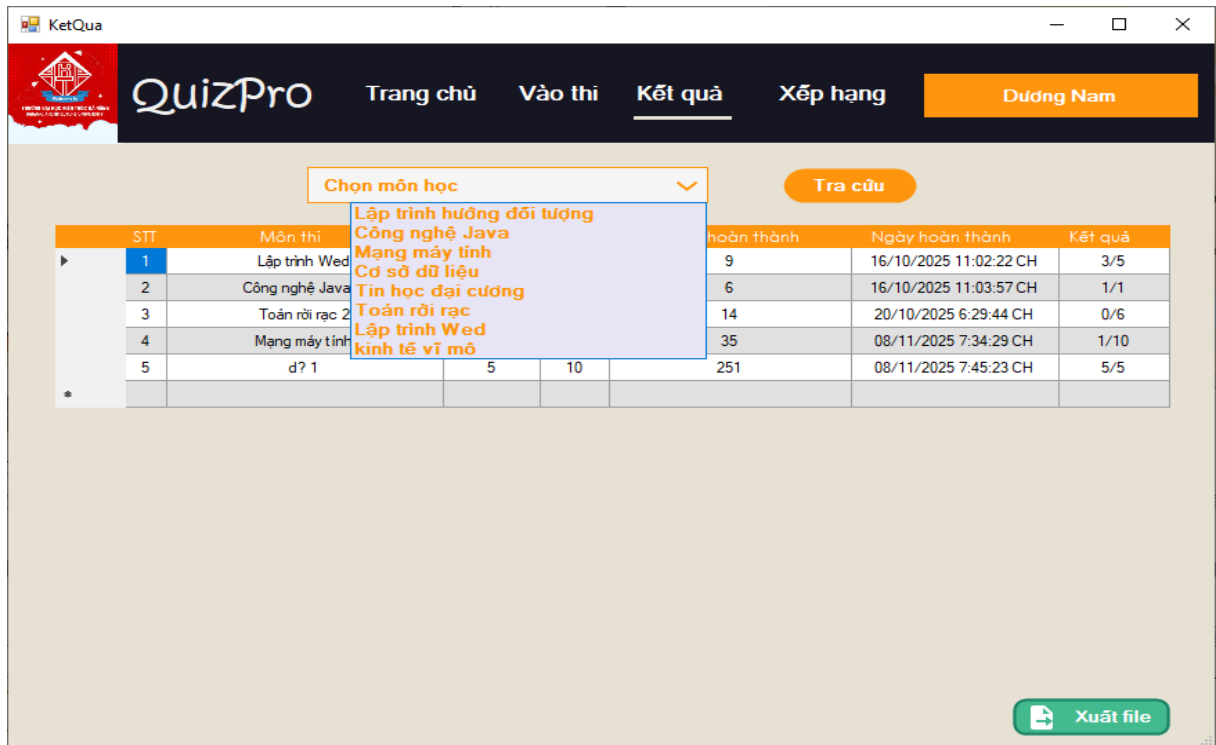
Học sinh thực hiện chọn môn học trước khi bắt đầu thi.



Hình 3. 5 Giao diện trang chọn môn thi

3.1.6 Giao diện trang kết quả thi

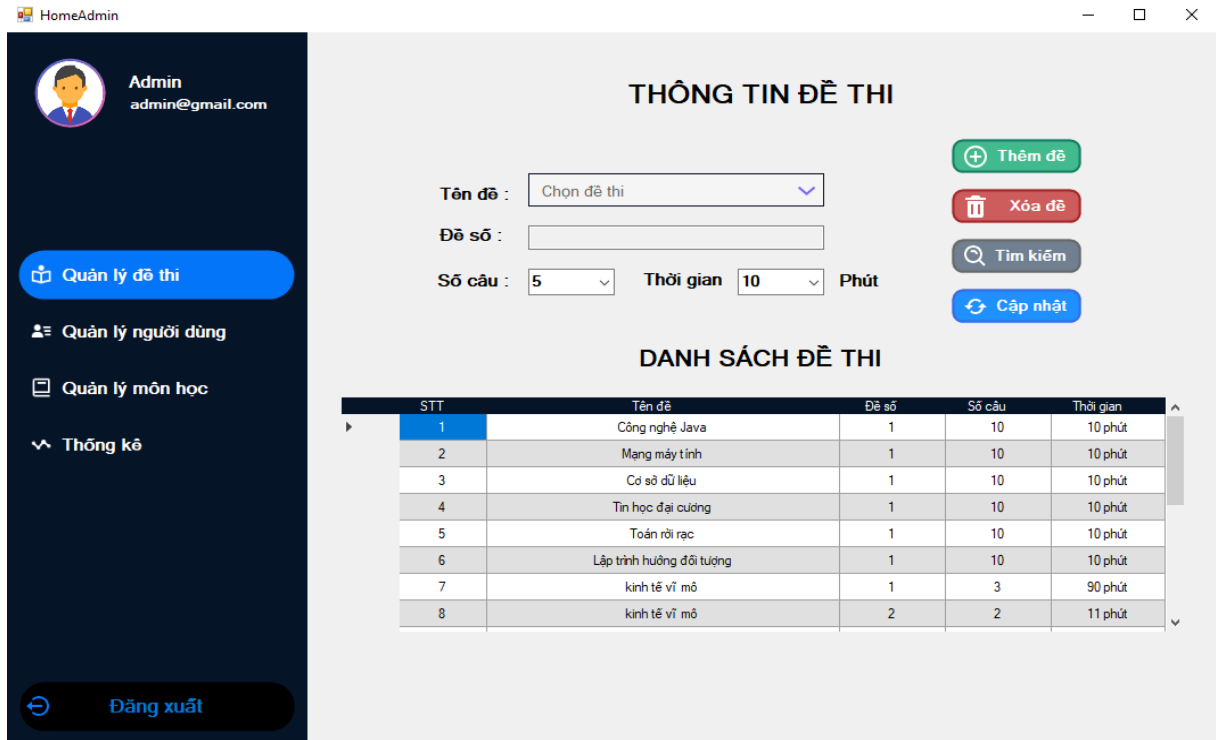
Học sinh có thể xem kết quả, thành tích từng môn học.



Hình 3. 6 Giao diện trang kết quả thi

3.1.7 Giao diện trang quản lý đề thi

Quản trị viên quản lý thêm đề, xóa đề.



Hình 3. 7 Giao diện trang quản lý đề thi

3.1.8 Giao diện trang tạo đề thi

Quản trị viên thêm từng câu hỏi cho đề thi.

The screenshot shows a web application window titled 'AddQuestion'. It features a 'Quay lại' (Back) button in the top left. The main heading is 'Môn học: Công nghệ Java' (Subject: Java Technology) with a star icon, and 'Đề số: 2' (Test number: 2). Below this, there is a section for 'Câu hỏi 1' (Question 1) with a large text input field. Underneath are four rows for multiple-choice answers: 'Đáp án A:', 'Đáp án B:', 'Đáp án C:', and 'Đáp án D:', each followed by a text input field. At the bottom left of this section is a 'ĐÚNG:' (Correct:) label with a dropdown menu. At the bottom of the window are two red buttons: 'Back' with a left arrow and 'Next' with a right arrow. The background has a colorful confetti pattern.

Hình 3. 8 Giao diện trang tạo đề thi

3.1.9 Giao diện trang quản lý người dùng

Quản trị viên sẽ quản lý tài khoản của học sinh.

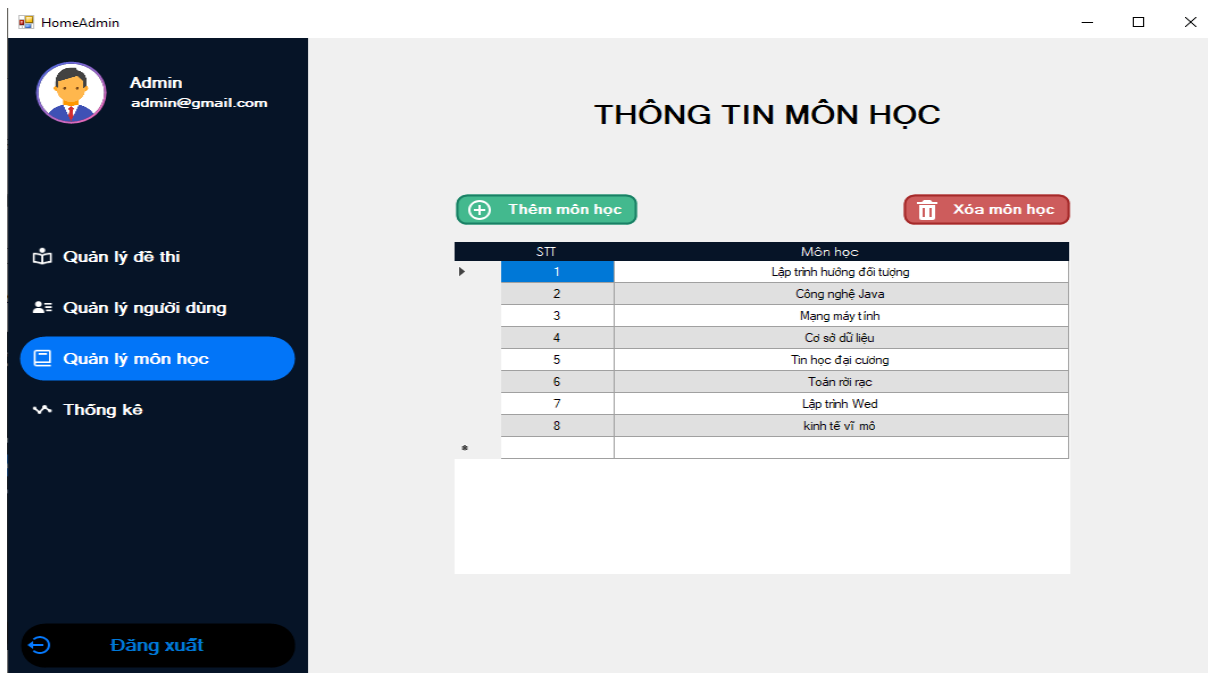
The screenshot shows a web application window titled 'HomeAdmin'. On the left is a dark sidebar with a user profile (Admin, admin@gmail.com) and a menu with options: 'Quản lý đề thi' (Manage tests), 'Quản lý người dùng' (Manage users - highlighted in blue), 'Quản lý môn học' (Manage subjects), and 'Thống kê' (Statistics). At the bottom of the sidebar is a 'Đăng xuất' (Logout) button. The main content area is titled 'THÔNG TIN NGƯỜI DÙNG' (User Information) and contains four input fields: 'Tên đăng nhập:' (Username), 'Họ tên:' (Full name), 'Email:', and 'Số điện thoại:' (Phone number). To the right of these fields are four buttons: 'Cập nhật' (Update), 'Tìm kiếm' (Search), 'Xóa' (Delete), and 'Tải lại' (Refresh). Below this is a section titled 'DANH SÁCH NGƯỜI DÙNG' (User List) containing a table with 5 columns: STT, Tên đăng nhập, Họ tên, Email, and Số điện thoại. The table has 5 rows of data. Below the table is a large empty white box.

STT	Tên đăng nhập	Họ tên	Email	Số điện thoại
1	john_doe	John Doe	john.doe@example.com	1234567890
2	jane_smith	Jane Smith	jane.smith@example.com	9876543210
3	user1	Dương Nam	user1@gmail.com	0942909123
4	user2	Dương Nam A	2	2
5	user3	Dương Nam B	4	4

Hình 3. 9 Giao diện trang quản lý người dùng

3.1.10 Giao diện trang quản lý môn học

Quản trị viên sẽ quản lý thêm bớt môn học.



Hình 3. 10 Giao diện trang quản lý môn học

3.1.11 Giao diện trang thống kê

Quản trị viên sẽ thống kê toàn bộ đề thi , số người dùng hiện có , tổng số lượt thi và xuất file báo cáo.



Hình 3. 11 Giao diện trang thống kê

3.1.12 Giao diện trang tạo đề thi giáo viên

Giáo viên cũng có thể tạo đề thi để gửi lên trưởng khoa.

Hình 3. 12 Giao diện trang tạo đề thi giáo viên

3.1.13 Giao diện trang quản lý đề thi giáo viên

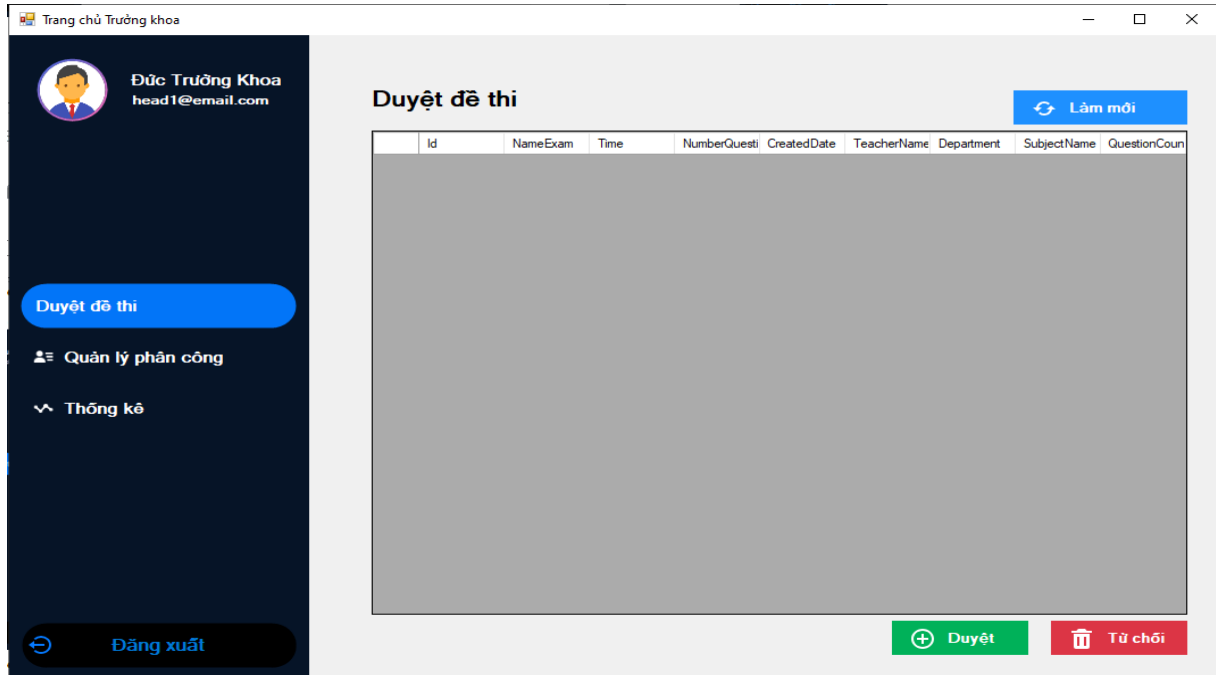
Giáo viên sẽ quản lý đề thi của mình tạo ra và có thêm thêm câu hỏi hoặc chỉnh sửa câu hỏi.

ID	Tên đề	Môn học	Thời gian	Số câu	Trạng thái	Ngày tạo	Điểm TB
1183	đ? 1	Toán rời rạc	15	5	Approved	08/11/2025 7:...	10
1182	Toán r?i r?c 2	Toán rời rạc	45	5	Approved	20/10/2025 6:...	0
1173	Lập trình hướng đối t...	Lập trình hướng ...	3	2	Approved	26/09/2025 9:...	0
171	Công nghệ Java 2	Công nghệ Java	5	1	Approved	22/09/2025 1:...	10
170	kinh tế vĩ mô	kinh tế vĩ mô	11	2	Approved	22/09/2025 1:...	0
169	kinh tế vĩ mô	kinh tế vĩ mô	90	3	Approved	22/09/2025 1:...	0

Hình 3. 13 Giao diện trang quản lý đề thi giáo viên

3.1.14 Giao diện trang quản lý duyệt đề trưởng khoa

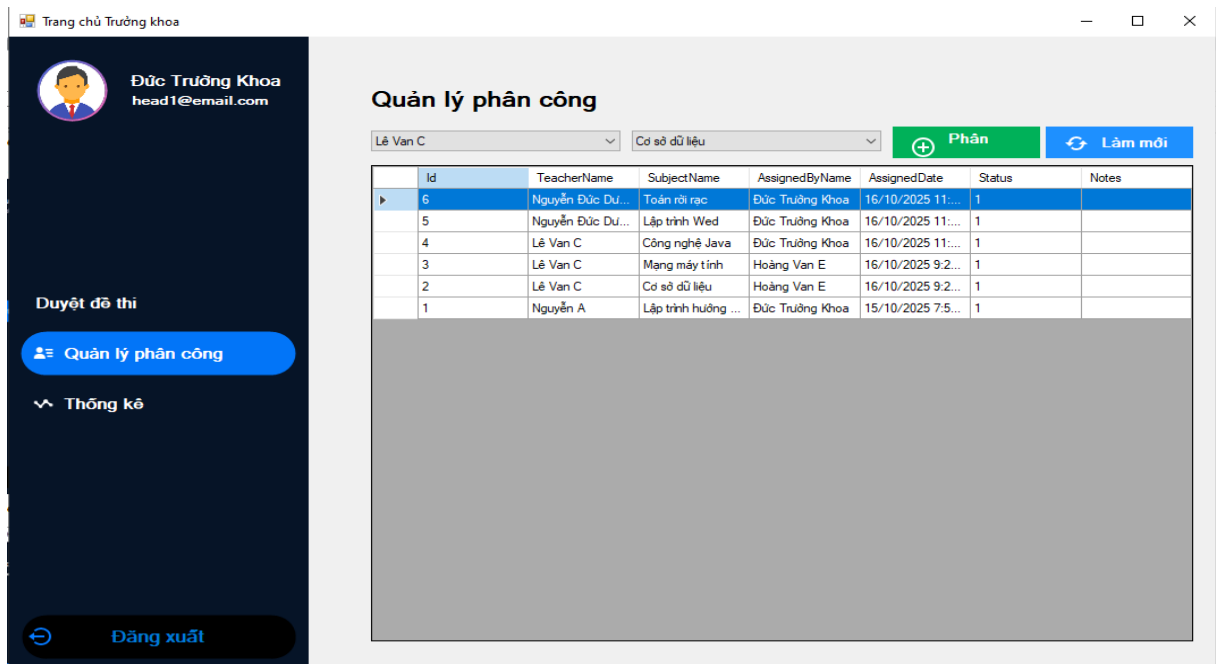
Khi giáo viên tạo đề thi cần thông qua trưởng khoa để xem xét chấp nhận đề thi trước khi cho học sinh làm.



Hình 3. 14 Giao diện trang quản lý duyệt đề trưởng khoa

3.1.15 Giao diện trang quản lý phân công trưởng khoa

Trưởng khoa sẽ phân công giáo viên dạy môn học bất khi , mỗi môn học có thể được nhiều giáo viên dạy.



Hình 3. 15 Giao diện trang quản lý phân công trưởng khoa

3.3 Cấu trúc tổ chức code

3.2.1 Cấu trúc tổ chức Front – End

Phần giao diện được xây dựng bằng Windows Forms theo mô hình lập trình điều khiển sự kiện. Mỗi chức năng của người dùng tương ứng với một Form riêng biệt, giúp chương trình rõ ràng và dễ phát triển. Ví dụ: Form quản lý người dùng, Form tạo đề thi, Form làm bài thi của học sinh, Form duyệt đề thi của trưởng khoa,... Tất cả đều được thiết kế theo nguyên tắc “một Form – một chức năng” nên đảm bảo tính tách biệt và hạn chế lỗi phát sinh.

Trong từng Form, các sự kiện như nhấn nút, chọn dữ liệu hoặc nhập thông tin sẽ được xử lý thông qua các phương thức sự kiện (event handlers). Giao diện người dùng chỉ giữ vai trò hiển thị, nhận thao tác của người dùng và gửi yêu cầu xuống tầng nghiệp vụ. Điều này tránh việc viết logic xử lý trực tiếp trong Form – vốn là nguyên nhân khiến hệ thống dễ rối và khó bảo trì.

Ngoài ra, để tăng tính thống nhất trong giao diện, các control tùy chỉnh như nút bấm, bảng hiển thị hoặc hộp thoại thông báo được chuẩn hóa dưới dạng component riêng và tái sử dụng trong toàn hệ thống.

3.2.2 Cấu trúc tổ chức Back - End

Phần Back-End được chia thành ba tầng: tầng đối tượng dữ liệu (DTO), tầng xử lý nghiệp vụ (BUS) và tầng truy cập dữ liệu (DAL). Việc chia thành ba tầng như vậy giúp toàn bộ hệ thống hoạt động có cấu trúc và tách biệt rõ ràng.

Tầng DTO chịu trách nhiệm lưu trữ dữ liệu tạm thời dưới dạng đối tượng, ví dụ đối tượng người dùng, đối tượng môn học, đối tượng câu hỏi, đối tượng đề thi, kết quả thi,... Các đối tượng này giúp việc truyền tải dữ liệu giữa các tầng trở nên thống nhất và an toàn.

Tầng DAL đảm nhận việc giao tiếp trực tiếp với cơ sở dữ liệu SQL Server. Mọi câu lệnh truy vấn như thêm, sửa, xóa, lấy danh sách đều được thực thi tại tầng này. DAL sử dụng ADO.NET để kết nối và tương tác với dữ liệu. Nhờ tách riêng DAL, việc thay đổi cấu trúc cơ sở dữ liệu hoặc bổ sung truy vấn mới không ảnh hưởng đến giao diện và logic xử lý phía trên.

Tầng BUS là nơi thực hiện toàn bộ nghiệp vụ của hệ thống. Tầng này kiểm tra tính hợp lệ của dữ liệu, xử lý logic, tính điểm bài thi, kiểm tra quyền truy cập, duyệt đề thi, phân công giáo viên và gọi đến DAL để lấy hoặc cập nhật dữ liệu. Tầng BUS giúp đảm bảo rằng mọi quy trình của hệ thống đều được xử lý đúng theo yêu cầu bài toán và tách biệt hoàn toàn với giao diện.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

Sau quá trình nghiên cứu và triển khai đề tài “Xây dựng phần mềm thi trắc nghiệm”, hệ thống đã hoàn thành đầy đủ các chức năng cơ bản phục vụ nhu cầu tổ chức các kỳ thi trắc nghiệm trong môi trường học tập. Các kết quả chính đạt được bao gồm:

- Xây dựng được phần mềm thi trắc nghiệm hoạt động ổn định trên nền Windows, giao diện trực quan, dễ sử dụng.
- Hệ thống hỗ trợ đầy đủ bốn vai trò: Học sinh, Giáo viên, Trưởng khoa và Quản trị viên, mỗi vai trò có bộ chức năng riêng.
- Chức năng thi trắc nghiệm được triển khai hoàn chỉnh: chọn môn thi, chọn bài thi, làm bài, tính thời gian, nộp bài và tự động chấm điểm.
- Hoàn thiện các chức năng quản lý như: quản lý người dùng, quản lý môn học, tạo đề thi, duyệt đề thi, phân công giáo viên.
- Xây dựng được cơ sở dữ liệu thống nhất, đảm bảo toàn vẹn dữ liệu và lưu trữ lịch sử làm bài.
- Hoàn thiện các chức năng thống kê và xuất báo cáo, hỗ trợ công tác theo dõi và quản lý.

Hệ thống đảm bảo hoạt động ổn định, xử lý dữ liệu chính xác và đáp ứng được các yêu cầu nghiệp vụ của bài toán đặt ra.

2. Những mặt còn hạn chế

Mặc dù hệ thống đã đáp ứng phần lớn yêu cầu, tuy nhiên vẫn tồn tại một số hạn chế cần được cải thiện trong tương lai:

- Giao diện Windows Forms còn mang tính truyền thống, chưa hiện đại và chưa hỗ trợ đa nền tảng.
- Chưa tích hợp tính năng chống gian lận nâng cao như giám sát từ xa, chống chuyển tab hoặc cảnh báo hành vi bất thường.
- Bộ câu hỏi và đề thi chưa hỗ trợ import/export với nhiều định dạng phong phú.
- Hệ thống chưa hỗ trợ thi trực tuyến thông qua mạng nội bộ hoặc internet.
- Chức năng phân quyền ở mức cơ bản, chưa có hệ thống phân quyền chi tiết theo từng thao tác nhỏ.
- Việc xử lý đề thi chưa hỗ trợ tạo ngẫu nhiên dạng “bốc câu hỏi” theo nhóm chủ đề hoặc ma trận đề.

- Chưa hỗ trợ báo cáo nâng cao dưới dạng biểu đồ và phân tích thống kê sâu.

3. Đề xuất giải pháp khắc phục

Để hoàn thiện hệ thống hơn nữa và tăng tính ứng dụng thực tế, một số hướng cải thiện và phát triển trong tương lai bao gồm:

Nâng cấp giao diện

- Chuyển sang WPF hoặc nền tảng web (ASP.NET Core) để hiện đại hóa giao diện và hỗ trợ đa nền tảng.

Phát triển phiên bản trực tuyến

- Xây dựng hệ thống thi online cho phép truy cập qua mạng LAN hoặc Internet.

Tăng cường tính năng chống gian lận

- Theo dõi hành vi người dùng, cảnh báo khi chuyển cửa sổ.
- Tự động khóa bài thi nếu phát hiện gian lận.

Mở rộng chức năng tạo đề

- Tạo đề theo ma trận kiến thức.
- Tự động sinh đề từ ngân hàng câu hỏi lớn.
- Hỗ trợ nhiều kiểu câu hỏi (ảnh, âm thanh, kéo thả...).

Phát triển hệ thống báo cáo nâng cao

- Biểu đồ trực quan (cột, tròn, đường).
- Phân tích thống kê điểm theo lớp, môn, giáo viên.

Cải thiện hiệu năng và bảo mật

- Sử dụng ORM như Entity Framework.
- Mã hóa mật khẩu bằng thuật toán mạnh (SHA256/Bcrypt).
- Tối ưu truy vấn và cache dữ liệu.

Hỗ trợ đa ngôn ngữ

- Giúp phần mềm có thể sử dụng trong nhiều môi trường đào tạo khác nhau.

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Văn Vinh, Giáo trình Lập trình C# cơ bản và nâng cao, NXB Khoa Học Kỹ Thuật, 2019.
- [2] Phạm Hữu Khang, Hệ quản trị cơ sở dữ liệu SQL Server, NXB Đại học Quốc Gia TP.HCM, 2018.
- [3] Trần Ngọc Quang, Lập trình Windows Forms với C#, Đại học Công nghệ Thông tin, 2020.
- [4] Nguyễn Đức Thuận, Phân tích và thiết kế hệ thống thông tin, NXB Thống Kê, 2021.
- [5] Nguyễn Văn Hào, Giáo trình Công nghệ phần mềm, NXB Giáo Dục, 2017.