

**Học Viện Công Nghệ Bưu Chính Viễn Thông**

**Viện Khoa Học Kỹ Thuật Bưu Điện**

**Môn Lập Trình Hướng Đối Tượng**



**Báo Cáo Bài Tập Lớn Đề Tài:**

**Ngân Hàng Câu Hỏi**

**Giảng viên : Nguyễn Việt Dũng**

**Sinh viên thực hiện: Mai Đức Anh**

**Nguyễn Anh Tuấn**

**Vũ Minh Hoàn**

**Lớp: INT11332-16**

**Mục Lục**

[**CHƯƠNG 1: MỞ ĐẦU** 4](#_Toc185463992)

[**1.1** **Lý do chọn đề tài** 4](#_Toc185463993)

[**1.2** **Nội dung và phạm vi đề tài** 5](#_Toc185464002)

[**1.2.1** **Nội dung đề tài** 5](#_Toc185464003)

[**1.2.2 Phạm vi đề tài** 6](#_Toc185464004)

[**1.3 Yêu cầu phi chức năng** 7](#_Toc185464005)

[**CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG** 8](#_Toc185464006)

[**2.1 Ngôn ngữ lập trình JAVA** 8](#_Toc185464007)

[**2.2. Spring Boot framework** 10](#_Toc185464008)

[**2.3 Cơ sở dữ liệu Mysql** 11](#_Toc185464009)

[**2.4 IntelliJ** 13](#_Toc185464010)

[**2.5 Rest API.** 14](#_Toc185464011)

[**2.6 Postman** 15](#_Toc185464012)

[**CHƯƠNG 3: PHÂN TÍCH** 16](#_Toc185464013)

[**3.1 Mô tả hệ thống** 16](#_Toc185464014)

[**3.2 Sơ đồ phân rã** 17](#_Toc185464034)

[**3.3 Biểu đồ ERD** 18](#_Toc185464037)

[**3.5 Đặc tả một số Use case chính** 34](#_Toc185464123)

[**3.5.1 Use case đăng nhập** 34](#_Toc185464124)

[**3.5.2 Use case quản lý câu hỏi** 35](#_Toc185464144)

[**3.5.3 Use Case quản lý môn học** 36](#_Toc185464166)

[**3.5.4 Use Case Tạo đề thi** 37](#_Toc185464188)

[**3.5.5 Use Case quản lý đề thi** 38](#_Toc185464210)

[**Chương 4: THIẾT KẾ GIAO DIỆN** 39](#_Toc185464230)

[**4.1 Mục tiêu và nghiệp vụ** 39](#_Toc185464231)

[**4.2 Mô tả các giao diện chính** 40](#_Toc185464244)

[**CHƯƠNG 5: KẾT LUẬN** 44](#_Toc185464274)

[**TÀI LIỆU THAM KHẢO** 45](#_Toc185464279)

[**1.** **https://www.npmjs.com/ - Tra cứu tài liệu thư viện** 45](#_Toc185464280)

[**2.** **https://chatgpt.com/ - Tra cứu thắc mắc cần giải đáp** 45](#_Toc185464281)

**Lời Mở Đầu**

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các hệ thống quản lý thông minh vào giáo dục và đào tạo ngày càng trở nên phổ biến và cần thiết. Việc xây dựng ngân hàng câu hỏi và tạo đề thi đã và đang đóng vai trò quan trọng trong việc hỗ trợ học sinh rèn luyện, trau dồi kiến thức, cũng như nâng cao kỹ năng và tinh thần trong các kỳ thi. Tuy nhiên, việc quản lý số lượng lớn câu hỏi và tổ chức chúng một cách khoa học, hiệu quả luôn là một thách thức không nhỏ đối với các cơ sở giáo dục.

Xuất phát từ thực tiễn đó, nhóm chúng em đã lựa chọn đề tài **"Hệ thống quản lý ngân hàng câu hỏi"** nhằm xây dựng một hệ thống thông minh, có khả năng quản lý toàn diện thông tin về các câu hỏi theo từng môn học, chủ đề, và mức độ khó. Hệ thống này sẽ giúp tự động hóa quá trình tạo đề thi theo cấu trúc định trước, đồng thời hỗ trợ học sinh tự luyện tập một cách có hệ thống và hiệu quả.

Báo cáo này trình bày chi tiết quá trình phân tích, thiết kế và triển khai hệ thống quản lý ngân hàng câu hỏi, từ việc xác định yêu cầu đến các giải pháp kỹ thuật được áp dụng trong quá trình xây dựng. Chúng em hy vọng rằng hệ thống không chỉ góp phần nâng cao chất lượng giảng dạy và học tập mà còn là cơ sở để mở rộng thêm nhiều tính năng hữu ích trong tương lai, đáp ứng nhu cầu đổi mới và phát triển của ngành giáo dục.

**Hoạt Động Thành Viên**

|  |  |
| --- | --- |
| Họ Và Tên | Vai Trò |
| Mai Đức Anh | slide , báo cáo |
| Nguyễn Anh Tuấn | Frontennd, Backend |
| Vũ Minh Hoàn | Backend |

# 

# **CHƯƠNG 1: MỞ ĐẦU**

* 1. **Lý do chọn đề tài**

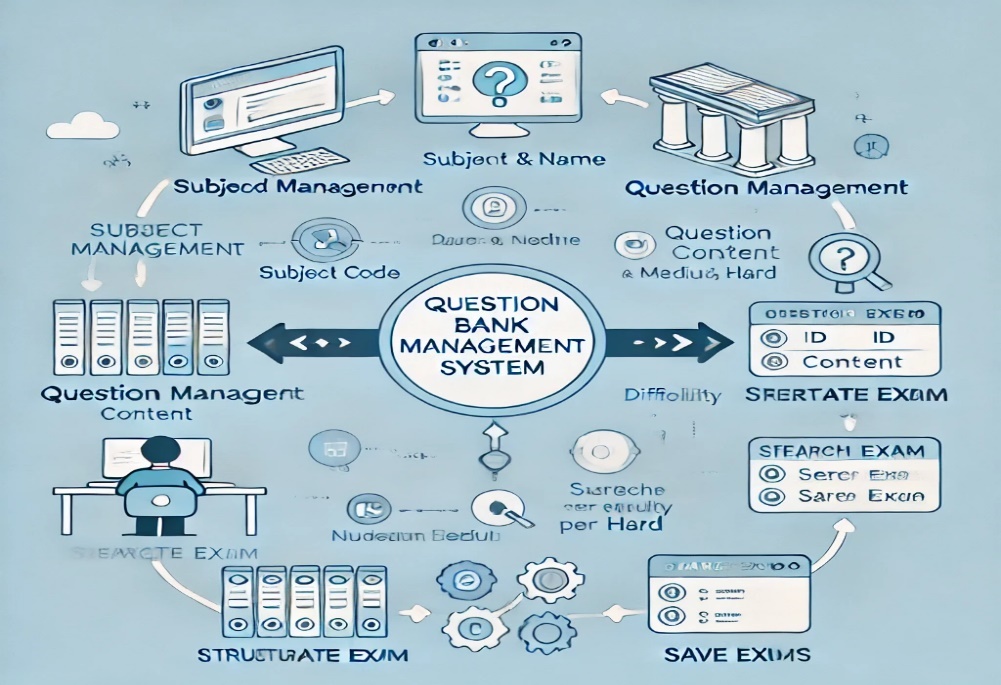
Trong bối cảnh giáo dục hiện nay, việc soạn thảo và quản lý câu hỏi tự luận đóng vai trò quan trọng trong quá trình giảng dạy và học tập. Tuy nhiên, thực tế cho thấy cả giáo viên và học sinh đều đang gặp nhiều khó khăn trong việc tìm kiếm và sử dụng tài liệu phù hợp.

Đối với giáo viên, việc soạn thảo bài tập và đề thi yêu cầu rất nhiều thời gian và công sức. Các thầy cô phải vất vả tìm kiếm tài liệu từ nhiều nguồn khác nhau để lựa chọn câu hỏi phù hợp. Việc này không chỉ gây áp lực lớn mà còn khó đảm bảo tính khoa học, đồng bộ về mức độ khó và cấu trúc đề thi. Bên cạnh đó, việc tự biên soạn đề thi thủ công thường thiếu tính hệ thống và dễ xảy ra tình trạng trùng lặp câu hỏi, chưa đánh giá đúng năng lực học sinh.

Đối với học sinh, việc tìm kiếm tài liệu ôn tập và luyện tập cũng không hề dễ dàng. Nhiều em gặp khó khăn khi không có nguồn tài liệu chính thống hoặc các đề thi mẫu phù hợp để luyện tập. Điều này khiến các em thiếu tự tin khi bước vào các kỳ thi thật, đặc biệt khi đề thi không bám sát cấu trúc và mức độ khó của các kỳ thi chính thức.

Chính vì vậy, việc xây dựng hệ thống quản lý ngân hàng câu hỏi tự luận là giải pháp hữu hiệu giúp:

* Hỗ trợ giáo viên: Tạo ra một ngân hàng câu hỏi phong phú, có sẵn và phân loại rõ ràng theo từng môn học, khối kiến thức và mức độ khó. Điều này giúp tiết kiệm thời gian, công sức cho giáo viên khi soạn đề thi, đồng thời đảm bảo tính đồng bộ và khoa học của đề thi.
* Giúp học sinh ôn tập hiệu quả: Học sinh có thể luyện tập với các câu hỏi bám sát cấu trúc đề thi thật, từ đó chuẩn bị tốt hơn cho các kỳ thi và đánh giá chính xác năng lực của bản thân.
* Nâng cao chất lượng đề thi: Hệ thống tự động tạo đề thi dựa trên yêu cầu cụ thể sẽ đảm bảo tính công bằng, cân bằng giữa các mức độ khó và khối kiến thức, đáp ứng mục tiêu kiểm tra, đánh giá năng lực học sinh/sinh viên.



**Hình 1. Quy Trình Ngân Hàng Câu Hỏi**

Với những lợi ích thiết thực này, hệ thống không chỉ hỗ trợ công tác giảng dạy của giáo viên mà còn giúp học sinh có được nguồn tài liệu ôn tập chất lượng, từ đó nâng cao hiệu quả học tập và chuẩn bị tốt nhất cho các kỳ thi quan trọng. Đây chính là lý do đề tài này có ý nghĩa thực tiễn cao và rất cần thiết trong lĩnh vực giáo dục hiện nay.

* 1. **Nội dung và phạm vi đề tài**
     1. **Nội dung đề tài**

Đề tài tập trung xây dựng hệ thống **quản lý ngân hàng câu hỏi tự luận** phục vụ công tác giảng dạy, kiểm tra và đánh giá trong môi trường giáo dục. Tìm hiểu, nghiên cứu về ngôn ngữ lập trình Java, Spring Boot framework và ReactJS nhằm áp dụng vào trong quá trình xây dựng ứng dụng web. Nội dung chính của đề tài bao gồm:

* **Quản lý môn học**:
  + Xây dựng chức năng lưu trữ và quản lý thông tin môn học như mã môn, tên môn và số tín chỉ.
  + Phân chia môn học thành các **khối kiến thức** khác nhau để thuận tiện trong việc phân loại và lựa chọn câu hỏi.
* **Quản lý câu hỏi tự luận**:
* Hệ thống lưu trữ đầy đủ thông tin về câu hỏi bao gồm: mã câu hỏi, nội dung câu hỏi, môn học, khối kiến thức và mức độ khó (Dễ, Trung bình, Khó, Rất khó).
* Cho phép thêm, chỉnh sửa và xóa câu hỏi trong ngân hàng.
* **Quản lý đề thi**:
* Xây dựng chức năng tạo đề thi tự động theo yêu cầu cụ thể như số lượng câu hỏi, mức độ khó và khối kiến thức.
* Đảm bảo đề thi tạo ra tuân thủ đúng cấu trúc và phân bổ hợp lý giữa các mức độ khó.
* **Tìm kiếm và lọc câu hỏi**:
* Xây dựng tính năng tìm kiếm câu hỏi nhanh chóng dựa trên các tiêu chí như mã câu hỏi, môn học, mức độ khó và khối kiến thức.
* **Tạo đề thi theo cấu trúc**:
* Cho phép người dùng tạo đề thi với cấu trúc tùy chỉnh bao gồm số lượng câu hỏi, tỷ lệ mức độ khó và chủ đề phù hợp với mục tiêu kiểm tra đánh giá.
* **Lưu trữ và tái sử dụng**:
* Hệ thống lưu trữ đề thi và ngân hàng câu hỏi một cách khoa học, thuận tiện cho việc chỉnh sửa và sử dụng lại trong tương lai.

**1.2.2 Phạm vi đề tài**

Đề tài được triển khai với phạm vi cụ thể như sau:

**Đối tượng sử dụng**:

* + Giáo viên, giảng viên và các cán bộ giáo dục chịu trách nhiệm soạn đề thi, ra đề kiểm tra.
  + Học sinh, sinh viên có nhu cầu luyện tập và ôn thi với các đề thi mẫu.

**Phạm vi dữ liệu**:

* + Dữ liệu tập trung vào **câu hỏi tự luận** thuộc các môn học trong hệ thống giáo dục trung học và đại học.
  + Các câu hỏi được phân loại theo:
    - **Môn học**: Ví dụ như Toán học, Ngữ văn, Lịch sử, Vật lý,...
    - **Khối kiến thức**: Các chủ đề, chương hoặc phần kiến thức cụ thể trong mỗi môn học.
    - **Mức độ khó**: Dễ, Trung bình, Khó và Rất khó.

**Phạm vi kỹ thuật**:

* + Hệ thống sẽ được phát triển dưới dạng phần mềm hoặc ứng dụng web có giao diện thân thiện, dễ sử dụng.
  + Chức năng chính: quản lý câu hỏi, tạo đề thi tự động, tìm kiếm và lọc câu hỏi.
  + Lưu trữ dữ liệu: Hệ thống sẽ lưu trữ dữ liệu ngân hàng câu hỏi và đề thi trên cơ sở dữ liệu có tính bảo mật và khả năng mở rộng.

**Giới hạn**:

* + Đề tài chỉ tập trung vào câu hỏi tự luận, chưa mở rộng sang câu hỏi trắc nghiệm hoặc các hình thức kiểm tra khác.

**1.3 Yêu cầu phi chức năng**

**+** Độ tin cậy: Hệ thống phải đảm bảo độ chính xác .

+ Tính khả dụng: Giao diện người dùng thân thiện, dễ sử dụng.

+ Tính mở rộng: Dễ dàng bổ sung tính năng trong tương lai.

+ Nội dung: Phải dễ hiểu và thân thiện với người sử dụng. Không có lỗi chính tả, không khó để đọc chữ, hình ảnh được sắp xếp gọn gàng, hợp lý. Chương trình cần bắt lỗi đầy đủ khi dữ liệu không đúng.

**CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG**

**2.1 Ngôn ngữ lập trình JAVA**

Ngôn ngữ lập trình *Java* ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995 (Java 1.0 [J2SE]). Tính đến thời điểm này (tháng 5/2017) phiên bản mới nhất của Java Standard Edition (JSE) là 8. Với ưu thế về đa nền tảng (multi platform) Java càng lúc càng được ứng dụng rộng rãi trên nhiều thiết bị từ máy tính đến mobile và nhiều thiết bị phần cứng khác...

A logo of a coffee cup

Description automatically generated

**Hình 2. Logo ngôn ngữ lập trình Java**

- *Java* là ngôn ngữ lập trình hướng đối tượng nên nó cũng có 4 đặc điểm chung của các ngôn ngữ hướng đối tượng:

• Tính trừu tượng (Abstraction): là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.

• Tính đa hình (Polymorphism): cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.

• Tính kế thừa (Inheritance): Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

• Tính đóng gói (Encapsulation): là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.

A diagram of a diagram

Description automatically generated

**Hình 3. Bốn tính chất OOP**

- Bên cạnh đó Java còn có một số đặc tính khác:

• Độc lập nền (Write Once, Run Anywhere): Không giống như nhiều ngôn ngữ lập trình khác như C và C ++, khi Java được biên dịch, nó không được biên dịch sang mã máy cụ thể, mà thay vào đó là mã byte code chạy trên máy ảo Java (JVM). Điều này đồng nghĩa với việc bất cứ thiết bị nào có cài đặt JVM sẽ có thể thực thi được các chương trình Java.

• Đơn giản: học Java thật sự dễ hơn nhiều so với C/C++, nếu bạn đã quen với các ngôn ngữ lập trình hướng đối tượng thì việc học Java sẽ dễ dàng hơn. Java trở nên đơn giản hơn so với C/C++ do đã loại bỏ tính đa kế thừa và phép toán con trỏ từ C/C++. 7

• Bảo mật: Java hỗ trợ bảo mật rất tốt bởi các thuật toán mã hóa như mã hóa một chiều (one way hashing) hoặc mã hóa công cộng (public key)...

• Đa luồng: Với tính năng đa luồng Java có thể viết chương trình có thể thực thi nhiều task cùng một lúc. Tính năng này thường được xử dụng rất nhiều trong lập trình game.

• Hiệu suất cao: nhờ vào trình thu gom rác (garbage collection), giải phóng bộ nhớ đối với các đối tượng không được dùng đến.

• Linh hoạt: Java được xem là linh hoạt hơn C/C ++ vì nó được thiết kế để thích ứng với nhiều môi trường phát triển.

**2.2. Spring Boot framework**

Spring Boot là một framework Java được sử dụng để xây dựng các ứng dụng và dịch vụ web dễ dàng và nhanh chóng. Nền tảng cung cấp các cấu hình mặc định cho một số thư viện và bộ công cụ hỗ trợ xây dựng, triển khai, quản lý ứng dụng Spring-based.



**Hình 4. Spring Boot là gì?**

Cách Spring Boot hoạt động nhằm tối ưu hóa quy trình phát triển ứng dụng Java. Điều này sẽ giúp nhà phát triển tập trung vào việc xây dựng tính năng chính của ứng dụng mà không cần phải lo lắng về cấu hình phức tạp.

-Một số ưu điểm dưới đây đã giúp Spring Boot trở thành sự lựa chọn phổ biến cho việc xây dựng các ứng dụng và dịch vụ web trong hệ sinh thái Java:



**Hình 5. Ưu điểm của Spring Boot**

+ Tối ưu hóa quá trình phát triển: Spring Boot cung cấp cấu hình mặc định thông minh và tự động, giúp giảm thiểu việc cấu hình thủ công và tối ưu quá trình phát triển ứng dụng Java.

+ Tích hợp tốt: Spring Boot tích hợp tốt với nhiều công nghệ và thư viện khác trong hệ sinh thái Spring Framework. Nền tảng cho phép hệ thống dễ dàng tích hợp các module và dịch vụ khác nhau mà không cần phải lo lắng về cấu hình phức tạp.

+ Embedded server: Spring Boot đi kèm với các máy chủ nhúng như Tomcat, Jetty, hoặc Undertow. Đây là công cụ không thể thiếu trong việc triển khai ứng dụng một cách đơn giản mà không cần cấu hình thêm bất kỳ máy chủ nào khác.

+ Tự động cấu hình: Spring Boot sử dụng cơ chế cấu hình tự động thông minh, cho phép ứng dụng tự cấu hình dựa trên các thư viện và module được sử dụng.

+ Quản lý phụ thuộc: Spring Boot cung cấp các công cụ quản lý phụ thuộc mạnh mẽ như Maven hoặc Gradle, giúp quản lý các phụ thuộc của ứng dụng một cách hiệu quả.

+ Monitoring và quản lý: Spring Boot cung cấp các công cụ hỗ trợ giám sát và quản lý ứng dụng dễ dàng, bao gồm Spring Boot Actuator cho việc giám sát và quản lý ứng dụng.

**2.3 Cơ sở dữ liệu Mysql**

**MySQL** là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được phát triển và duy trì bởi Oracle Corporation. Nó sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để quản lý và thao tác dữ liệu. MySQL được sử dụng rộng rãi trong các ứng dụng web và các dự án có yêu cầu hiệu suất cao vì tính linh hoạt và khả năng mở rộng của nó.

 **Hình6. Mysql**

**Ưu điểm của MySQL:**

1. **Mã nguồn mở**: MySQL là phần mềm miễn phí và mã nguồn mở, cho phép người dùng tự do sử dụng, thay đổi và phân phối lại.
2. **Hiệu suất cao**: MySQL có khả năng xử lý lượng lớn dữ liệu với hiệu suất tốt, nhờ vào khả năng tối ưu hóa các truy vấn và hỗ trợ các chỉ mục (index).
3. **Dễ sử dụng**: MySQL có cú pháp đơn giản và dễ học, là sự lựa chọn phổ biến cho các lập trình viên mới bắt đầu.
4. **Tính linh hoạt**: Hỗ trợ nhiều hệ điều hành như Windows, Linux, macOS, giúp dễ dàng triển khai trên các nền tảng khác nhau.
5. **Cộng đồng lớn**: Là một công cụ phổ biến, MySQL có một cộng đồng hỗ trợ rộng lớn và tài liệu phong phú.

**Nhược điểm của MySQL:**

1. **Hạn chế tính năng**: Mặc dù MySQL phát triển nhanh chóng, nhưng so với các hệ quản trị cơ sở dữ liệu như **PostgreSQL** hoặc **Oracle**, MySQL vẫn thiếu một số tính năng nâng cao như hỗ trợ các loại dữ liệu phức tạp và tính toàn vẹn dữ liệu cao cấp.
2. **Khả năng mở rộng hạn chế**: Dù MySQL có khả năng mở rộng tốt trong nhiều trường hợp, nhưng khi làm việc với các ứng dụng rất lớn và phức tạp, nó có thể gặp khó khăn trong việc xử lý tải nặng hoặc yêu cầu cao về khả năng phân tán dữ liệu.
3. **Quản lý sao lưu và phục hồi**: Một số người dùng cho rằng khả năng sao lưu và phục hồi của MySQL không linh hoạt và dễ sử dụng bằng một số hệ quản trị cơ sở dữ liệu khác.

Nhìn chung, MySQL là một lựa chọn tuyệt vời cho nhiều ứng dụng, đặc biệt là các ứng dụng web, nhờ vào tính linh hoạt, hiệu suất và cộng đồng hỗ trợ mạnh mẽ. Tuy nhiên, trong các tình huống yêu cầu tính năng phức tạp hoặc khả năng mở rộng vượt trội, người dùng có thể cần xem xét các giải pháp khác.

**2.4 IntelliJ**

**IntelliJ IDEA** là một **môi trường phát triển tích hợp (IDE)** mạnh mẽ được phát triển bởi **JetBrains**. Nó được thiết kế đặc biệt để hỗ trợ phát triển phần mềm bằng **Java** và các ngôn ngữ lập trình khác như **Kotlin**, **Groovy**, **Scala**, **Python**, **JavaScript**, v.v.

a. Hỗ trợ thông minh (Smart Coding Assistance)

* **Code Completion**: Gợi ý và tự động hoàn thành mã thông minh.
* **Code Analysis**: Phát hiện lỗi trong thời gian thực và gợi ý cách sửa.
* **Live Templates**: Cung cấp các đoạn mã mẫu giúp viết code nhanh hơn.

b. Refactoring mạnh mẽ

* Cho phép **đổi tên biến**, **tách/ghép hàm**, và **tái cấu trúc mã nguồn** dễ dàng mà không làm hỏng chương trình.

c. Hỗ trợ đa ngôn ngữ

* IntelliJ không chỉ hỗ trợ Java mà còn các ngôn ngữ như: **Kotlin**, **Scala**, **Python**, **JavaScript**, **TypeScript**, và nhiều ngôn ngữ khác.

d. Tích hợp công cụ xây dựng dự án

* Hỗ trợ các công cụ như **Maven**, **Gradle**, **Ant** giúp quản lý dependencies và build project dễ dàng.

e. Debugging và Testing

* Hỗ trợ Debug mạnh mẽ với khả năng **breakpoint**, theo dõi giá trị biến theo thời gian thực.
* Tích hợp các framework kiểm thử như **JUnit**, **TestNG**.

f. Hỗ trợ Framework hiện đại

* IntelliJ tích hợp các framework phổ biến như:
  + **Spring Boot** và **Spring MVC**
  + **Hibernate**
  + **Angular**, **React**, **Vue.js**
  + **Docker**, **Kubernetes**
  + **Microservices Architecture**

g. Quản lý mã nguồn

* Tích hợp với các hệ thống quản lý mã nguồn như **Git**, **GitHub**, **SVN**, **Mercurial**, v.v.

h. Hiệu suất và tùy chỉnh

* IntelliJ được tối ưu hóa để hoạt động nhanh chóng và mượt mà.
* Cho phép người dùng tùy chỉnh giao diện và cài đặt plugin để mở rộng tính năng.

**2.5 Rest API.**

2.5.1 Khái niệm Rest API

- Rest API (Representational State Transfer Application Programming Interface) là một giao diện lập trình ứng dụng dựa trên kiến trúc REST, cho phép các hệ thống giao tiếp với nhau thông qua giao thức HTTP. Rest API tập trung vào việc định nghĩa và thao tác với tài nguyên (resource) bằng cách sử dụng các phương thức HTTP chuẩn như GET, POST, PUT, PATCH, DELETE

- Rest API tuân theo nguyên tắc cốt lõi như client – server, stateless (không lưu trạng thái giữa các request), giúp đảm bảo tính linh hoạt, dễ bảo trì và mở rộng. Rest API thường dùng định dạng JSON hoặc XML để trao đổi dữ liệu giữa client và server.

2.5.2 Chuẩn Restful API

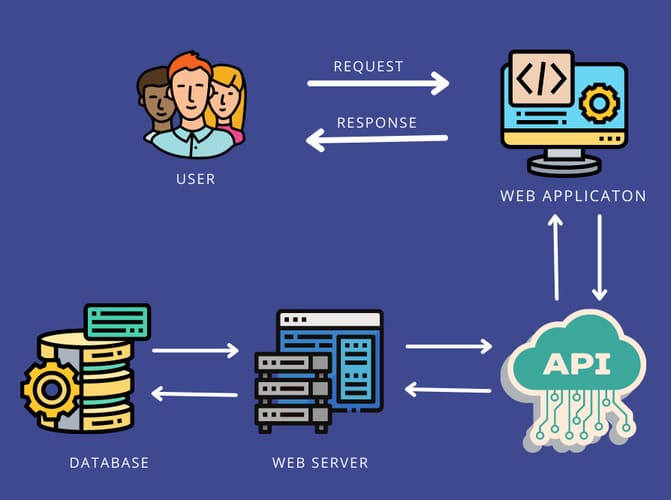
- Restful API là một dạng chuẩn của kiến trúc Rest API kèm theo các yêu cầu :

+ Tên endpoint mô tả đúng tài nguyên ví dụ:

* + /users thay vì /getuser
  + /orders thay vì /getorder

+ Sử dụng đúng phương thức HTTP

* + GET để lấy dữ liệu
  + POST để gửi dữ liệu đi
  + PUT/PATCH để cập nhật dữ liệu
  + DELETE để xóa dữ liệu



**2.6 Postman**

**Postman** là một công cụ mạnh mẽ và phổ biến dùng để **thiết kế**, **kiểm thử** và **quản lý API** trong quá trình phát triển phần mềm. Với giao diện thân thiện, Postman giúp các nhà phát triển và kiểm thử viên dễ dàng làm việc với API, kiểm tra dữ liệu, đảm bảo API hoạt động chính xác và hiệu quả.

* Postman ban đầu là một tiện ích mở rộng trên trình duyệt **Chrome**. Hiện tại, nó đã phát triển thành một ứng dụng độc lập hỗ trợ **Windows**, **macOS**, và **Linux**.
* Postman giúp người dùng gửi các yêu cầu HTTP/HTTPS đến **API endpoints** và nhận phản hồi một cách trực quan.
* Công cụ này được sử dụng rộng rãi trong quá trình phát triển và kiểm thử các ứng dụng dựa trên **RESTful API** và **GraphQL API**.

**CHƯƠNG 3: PHÂN TÍCH**

**3.1 Mô tả hệ thống**

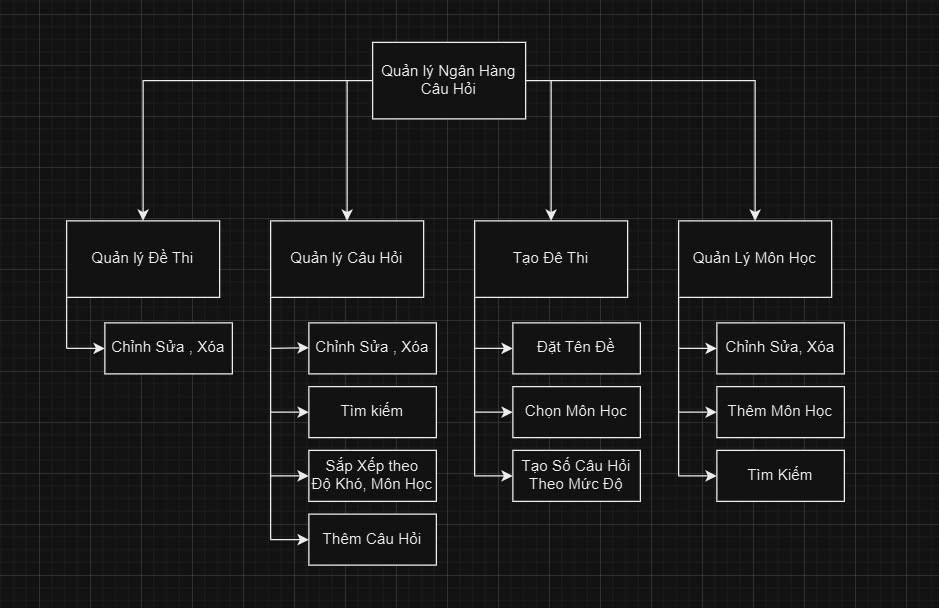
Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng các hệ thống quản lý thông minh vào giáo dục và đào tạo ngày càng trở nên phổ biến và cần thiết. Việc xây dựng ngân hàng câu hỏi và tạo đề thi đã và đang đóng vai trò quan trọng trong việc hỗ trợ học sinh rèn luyện, trau dồi kiến thức, cũng như nâng cao kỹ năng và tinh thần trong các kỳ thi. Tuy nhiên, việc quản lý số lượng lớn câu hỏi và tổ chức chúng một cách khoa học, hiệu quả luôn là một thách thức không nhỏ đối với các cơ sở giáo dục.

Đề tài "Hệ thống quản lý ngân hàng câu hỏi" nhằm xây dựng một hệ thống thông minh, có khả năng quản lý toàn diện thông tin về các câu hỏi theo từng môn học, chủ đề, và mức độ khó. Hệ thống này sẽ giúp tự động hóa quá trình tạo đề thi theo cấu trúc định trước, đồng thời hỗ trợ học sinh tự luyện tập một cách có hệ thống và hiệu quả.

Web gồm các mô-đun quản lý cơ bản như sau:

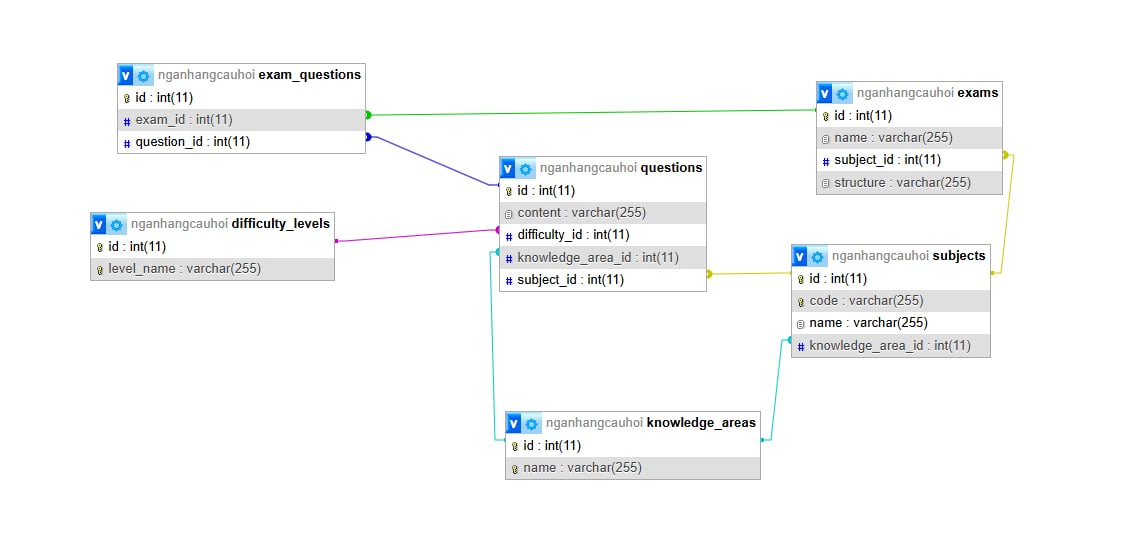
* Mô-đun quản lý câu hỏi:
  + Thêm câu hỏi:Cho phép người dùng có thể thêm câu hỏi bằng cách nhập nội dung và chọn môn học.
  + Chỉnh sửa và xóa: Người dùng có thể chỉnh sửa và xóa câu hỏi .
* Mô-đun Quản lý môn học :
* Thêm môn học: Cho phép thêm mới môn học vào hệ thống.
* Chỉnh sửa và xóa : Người dùng có thể chỉnh sửa hoặc xóa câu hỏi.
* Mô-đun Quản lý đề thi:
* Lưu trữ đề thi: Sau khi tạo đề thi của người dùng, đề thi sẽ tự động lưu trữ vào kho quản lý để người dùng có thể sử dụng lại.
* Chỉnh sửa và xóa : Sau khi đề được tạo người dùng có thể chỉnh sửa đề và xóa đề trong kho quản lý đề thi .
* Mô-đun Tạo đề thi:
* Đặt tên đề: Đặt tên theo yêu cầu người dùng .
* Tạo đề theo môn học: Người dùng tạo đề thi theo môn học muốn tạo.
* Tạo đề theo số câu hỏi, theo độ khó của đề: Người dùng có thể tạo số câu hỏi và độ khó của từng câu theo bản thân .
* Mô-đun tìm kiếm:
* Tìm kiếm Câu hỏi: Dễ dàng tìm kiếm câu hỏi qua nội dung câu hỏi.
* Tìm kiếm Môn Học: Cho phép tìm kiếm môn học theo tên môn học.

**3.2 Sơ đồ phân rã**

****

**Hình 7. Sơ đồ phân ra chức năng**

**3.3 Biểu đồ ERD**

****

**Hình 8. Biểu đồ EER**

- Hệ thống quản lý ngân hàng câu hỏi đã được thiết kế với các bảng liên quan như sau:

**1. Bảng exam\_questions**

* **Mô tả**: Đây là bảng trung gian để thể hiện mối quan hệ **nhiều - nhiều** giữa bảng exams và bảng questions.
* **Cột chính**:
  + id (PK): Khóa chính xác định duy nhất mỗi bản ghi.
* **Cột liên kết**:
  + exam\_id: Khóa ngoại liên kết đến bảng exams.
  + question\_id: Khóa ngoại liên kết đến bảng questions.

**2. Bảng exams**

* **Mô tả**: Chứa thông tin về các bài thi.
* **Cột chính**:
  + id (PK): Khóa chính, xác định duy nhất một bài thi.
* **Cột dữ liệu**:
  + name: Tên bài thi.
  + structure: Cấu trúc của bài thi.
* **Cột liên kết**:
  + subject\_id: Khóa ngoại liên kết đến bảng subjects.

**3. Bảng questions**

* **Mô tả**: Chứa thông tin về các câu hỏi trong ngân hàng câu hỏi.
* **Cột chính**:
  + id (PK): Khóa chính, xác định duy nhất mỗi câu hỏi.
* **Cột dữ liệu**:
  + content: Nội dung câu hỏi.
* **Cột liên kết**:
  + difficulty\_id: Khóa ngoại liên kết đến bảng difficulty\_levels (độ khó của câu hỏi).
  + knowledge\_area\_id: Khóa ngoại liên kết đến bảng knowledge\_areas (lĩnh vực kiến thức).
  + subject\_id: Khóa ngoại liên kết đến bảng subjects (môn học).

**4. Bảng difficulty\_levels**

* **Mô tả**: Lưu trữ thông tin về các mức độ khó của câu hỏi.
* **Cột chính**:
  + id (PK): Khóa chính, xác định duy nhất một mức độ khó.
* **Cột dữ liệu**:
  + level\_name: Tên mức độ khó (ví dụ: dễ, trung bình, khó).

**5. Bảng subjects**

* **Mô tả**: Chứa thông tin về các môn học.
* **Cột chính**:
  + id (PK): Khóa chính, xác định duy nhất một môn học.
* **Cột dữ liệu**:
  + code: Mã của môn học.
  + name: Tên của môn học.
* **Cột liên kết**:
  + knowledge\_area\_id: Khóa ngoại liên kết đến bảng knowledge\_areas.

**6. Bảng knowledge\_areas**

* **Mô tả**: Lưu trữ thông tin về các lĩnh vực kiến thức.
* **Cột chính**:
  + id (PK): Khóa chính, xác định duy nhất một lĩnh vực kiến thức.
* **Cột dữ liệu**:
  + name: Tên lĩnh vực kiến thức.

**Mối Quan Hệ Giữa Các Bảng**

1. **Exam – Questions**:
   * Mối quan hệ **nhiều - nhiều** thông qua bảng trung gian exam\_questions.
   * Một bài thi (exams) có thể chứa nhiều câu hỏi (questions) và một câu hỏi có thể thuộc nhiều bài thi.
2. **Questions – Difficulty Levels**:
   * Mối quan hệ **1 - nhiều**.
   * Một mức độ khó (difficulty\_levels) có thể áp dụng cho nhiều câu hỏi, nhưng mỗi câu hỏi chỉ có một mức độ khó.
3. **Questions – Knowledge Areas**:
   * Mối quan hệ **1 - nhiều**.
   * Một lĩnh vực kiến thức (knowledge\_areas) có thể có nhiều câu hỏi (questions).
4. **Subjects – Knowledge Areas**:
   * Mối quan hệ **1 - nhiều**.
   * Một lĩnh vực kiến thức (knowledge\_areas) có thể bao gồm nhiều môn học (subjects).
5. **Exams – Subjects**:
   * Mối quan hệ **1 - nhiều**.
   * Một môn học (subjects) có thể chứa nhiều bài thi (exams).
6. **Questions – Subjects**:
   * Mối quan hệ **1 - nhiều**.
   * Một môn học (subjects) có thể chứa nhiều câu hỏi (questions).

**Tính Khả Thi**

1. **Quản lý câu hỏi và bài thi**:
   * Hệ thống cho phép quản lý thông tin về bài thi, câu hỏi và mức độ khó của câu hỏi một cách rõ ràng.
2. **Dễ mở rộng**:
   * Có thể thêm các bảng mới như **classes**, **teachers**, hoặc phân loại chi tiết câu hỏi.
3. **Quản lý mối quan hệ chặt chẽ**:
   * Các khóa ngoại và bảng trung gian đảm bảo dữ liệu liên kết chặt chẽ giữa các thành phần của hệ thống.
4. **Phù hợp thực tế**:
   * Mô hình này phù hợp để xây dựng hệ thống ngân hàng câu hỏi cho các bài thi trực tuyến hoặc hệ thống quản lý câu hỏi học thuật.
5. **Triển khai**
6. **Đầu tiên khởi tạo một database có tên nganhangcauhoi và tạo các bảng**

A computer screen shot of text

Description automatically generated

**Hình 9: tạo các bảng Knowledge\_area, difficulty\_level, subjects**

* Tạo bảng Knowledge\_area để lưu thông tin vùng kiến thức
* Tạo bảng difficulty\_level để chứa thông tin độ khó của câu
* Tạo bảng subjects để lưu trữ môn học

A screenshot of a computer program

Description automatically generated

**Hình 10: tạo bảng questions và exams**

* Tạo bảng questions để chứa thông tin câu hỏi
* Tạo bảng exams để chứa thông tin đề kiểm tra

A close-up of a computer screen

Description automatically generated**Hình 11: tạo bảng exam\_questions**

* Tạo bảng exam\_questions để liên kết 2 bảng exams và questions dùng để truy vấn câu hỏi và tạo câu hỏi cho bài kiểm tra

1. **Xây dựng các lớp entity ánh xạ với cơ sở dữ liệu**

A screenshot of a computer program

Description automatically generated

**Hình 12: tạo class entity Difficulty\_level**

* xây dựng class Difficulty\_level gồm các thuộc tính ID và levelName
* có các phương thức getId, setId. getLevelName, setLevelName
* Phương thức trả về toString

A screen shot of a computer program

Description automatically generated

**Hình 13**: **tạo class entity Knowledge\_areas**

* xây dựng class Knowledge\_areas gồm các thuộc tính ID và Name
* có các phương thức getId, setId. getName, setName
* Phương thức trả về toString
* Tiếp tục tạo thêm các class entity cho subject, questions, exams, exam\_questions. Sao cho ánh xạ đúng với bảng database

1. **Xây dựng class cho package repository để tương tác với cơ sở dữ liệu**

A screen shot of a computer program

Description automatically generated**Hình 14: xây dựng repository cho difficulty**

* **Tương tự xây dựng repository với các đối tượng khác**

A computer screen shot of a program code

Description automatically generated**Hình 15: xây dựng repository cho đối tượng questions**

* viết câu lệnh truy vấn để hỗ trợ truy vấn lấy câu hỏi ngẫu nhiên hỗ trợ cho việc tạo đề thi random câu hỏi

1. **Xây dựng lớp service dùng để xử lý logic**

A screen shot of a computer program

Description automatically generated**Hình 16: xây dựng service xử lý logic cho đối tượng exams**

* Tạo phương thức lấy danh sách đề thi, tạo đề thi, sửa và xóa đề thi

A computer screen shot of a program

Description automatically generatedA computer code on a black background

Description automatically generated**Hình 17, 18: phương thức tạo đề thi**

* phương thức tạo đề thi random câu theo môn và theo độ khó
* **Cũng như vậy tạo service để xử lý logic với các đối tượng còn lại**

1. **Xây dựng comtroller xử lý api**

A screen shot of a computer code

Description automatically generated**Hình 19: tạo class controller cho đối tượng**

* xây dựng enpoint get cho exams để lấy dữ liệu
* xây dựng enpoint post cho exams để thêm dữ liệu
* xây dựng enpoint put cho exams để update dữ liệu
* xây dựng enpoint delete cho exams để xóa dữ liệu
* (Hình 20, 21)

A screen shot of a computer program

Description automatically generatedA screen shot of a computer screen

Description automatically generated **Hình 20,21 xây dựng controller cho exams**

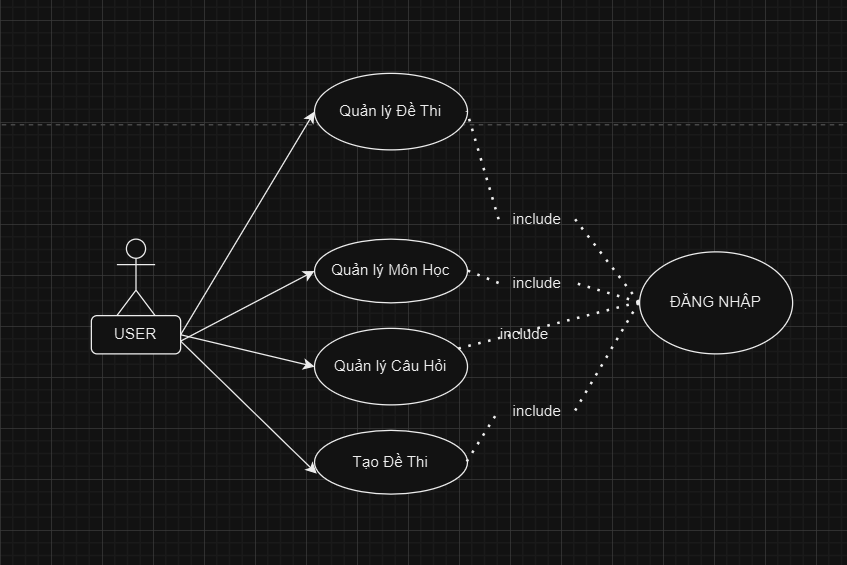
* **Xây dựng controller cho các đối tượng còn lại để xử lý api**

1. **Xây dựng Corsconfig để kết nối với fontend**

A screen shot of a computer program

Description automatically generated**Hình 22: Xây dựng corsconfig kết nối với fontend**

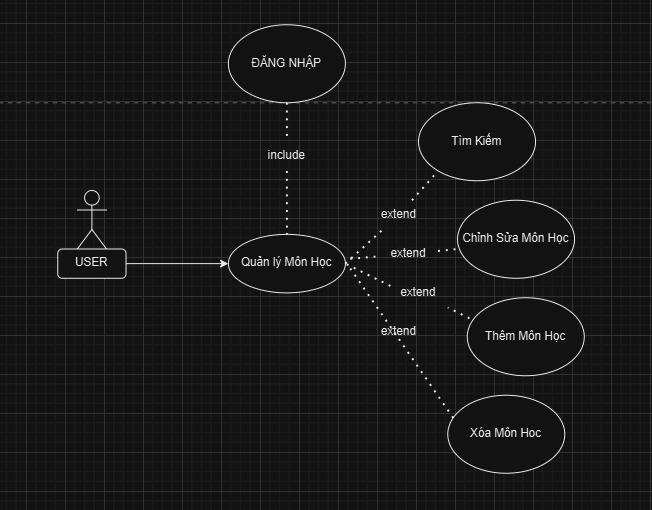
**3.4 Biểu đồ UseCase**

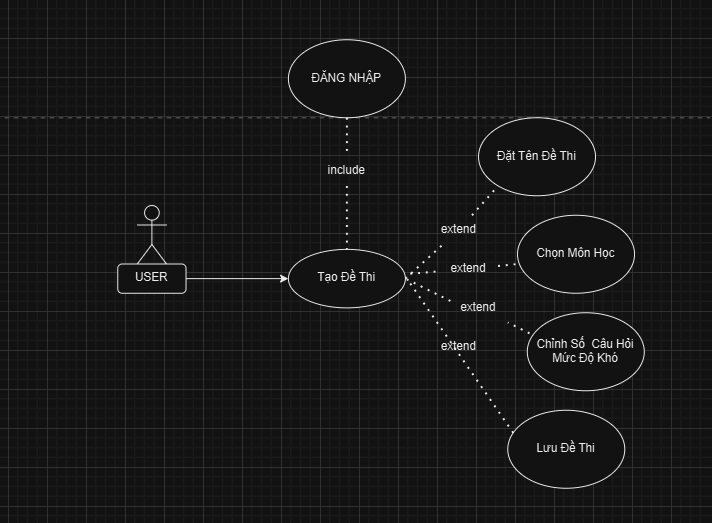
**3.4.1 Use case tồng quát **

**Hình 23. UseCase hệ thống**

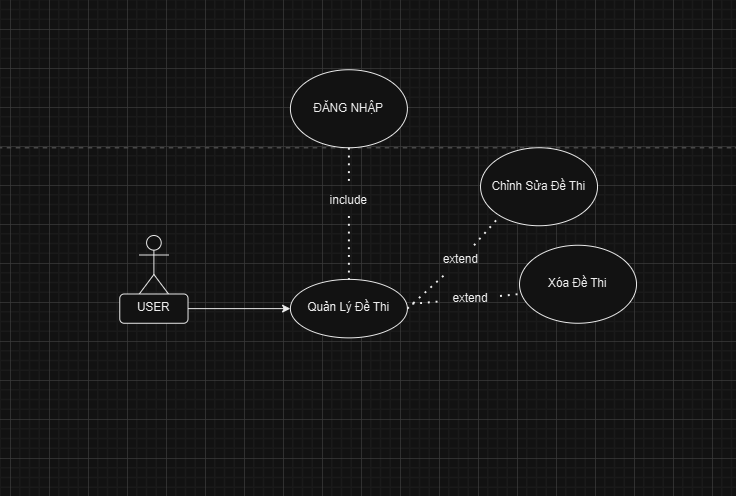
**A diagram of a person

Description automatically generatedHình 24. UseCase Quản lý câu hỏi**

**Hình 25. Use case Quản lý Môn Học**

****

**Hình 26. UseCase Tạo Đề Thi**

****

**Hình 27. UseCase quản Lý Đề Thi**

**3.5 Đặc tả một số Use case chính**

**3.5.1 Use case đăng nhập**

|  |  |
| --- | --- |
| Tên Use case | Đăng nhập |
| Tên tác nhân chính | User |
| Mức | 1 |
| Người chịu trách nhiệm | User |
| Điều kiện tiên quyết | User phải có tài khoản đã đăng kí trong hệ thống |
| Đảm bảo thành công | User đã đăng nhập vào hệ thống |
| Kích hoạt | Truy cập vào hệ thống |
| Chuỗi sự kiện | * Nhập tên tài khoản và mật khẩu * Hệ thống mã hóa dữ liệu truyền vào và đối chiếu với dữ liệu trong cơ sở dữ liệu * Hiện trang quản lý |

**Bảng 3.5.1 Đặc tả Use case Đăng nhập**

**3.5.2 Use case quản lý câu hỏi**

|  |  |
| --- | --- |
| Tên Use case | Quản lý Ngân hàng Câu Hỏi |
| Tên tác nhân chính | User |
| Mức | 1 |
| Người chịu trách nhiệm | User |
| Điều kiện tiên quyết | User đã đăng nhập |
| Đảm bảo tối thiểu | Form quản lý học viên phải trống để thực hiện tác vụ tiếp |
| Đảm bảo thành công | Đưa ra được danh sách câu hỏi sau khi đã thay đổi |
| Kích hoạt | Người dùng đăng nhập và chọn các chức năng trong mục quản lý câu hỏi |
| Chuỗi sự kiện | * Chọn chức năng "Quản lý Câu Hỏi”: hệ thống sẽ hiển thị các chức năng con trong mục quản lý câu hỏi gồm tìm kiếm, thêm, xóa, chỉnh sửa thông tin câu hỏi. * Hiển thị danh sách sắp xếp câu hỏi theo độ khó hoặc môn học. * Người dùng nhập thông tin câu hỏi mới nhập hoặc chỉnh sửa thông tin. |

**Bảng 3.5.2 Đặc tả Use case Quản lý học viên**

**3.5.3 Use Case quản lý môn học**

|  |  |
| --- | --- |
| Tên Use case | Quản lý môn học |
| Tên tác nhân chính | User |
| Mức | 1 |
| Người chịu trách nhiệm | User |
| Điều kiện tiên quyết | User đã đăng nhập |
| Đảm bảo tối thiểu | Form quản lý khóa học phải trống để thực hiện tác vụ tiếp |
| Đảm bảo thành công | Đưa ra được danh sách môn học sau khi đã thay đổi |
| Kích hoạt | Người dùng đăng nhập và chọn các chức năng trong mục quản lý môn học |
| Chuỗi sự kiện | * Chọn chức năng "Quản lý môn học”: hệ thống sẽ hiển thị các chức năng con trong mục quản lý môn học gồm tìm kiếm, thêm, xóa, chỉnh sửa thông tin môn học , thêm môn học vào trang quản lý . * Hiển thị danh sách môn học. * Người dùng nhập thông tin môn học mới nhập hoặc chỉnh sửa thông tin. |

**Bảng 3.5.2 Đặc tả Use case Quản lý môn học**

**3.5.4 Use Case Tạo đề thi**

|  |  |
| --- | --- |
| Tên Use case | Tạo đề thi |
| Tên tác nhân chính | User |
| Mức | 1 |
| Người chịu trách nhiệm | User |
| Điều kiện tiên quyết | User đã đăng nhập |
| Đảm bảo tối thiểu | Form quản lý khóa học phải trống để thực hiện tác vụ tiếp |
| Đảm bảo thành công | Đưa ra được đề thi sau khi đã tạo |
| Kích hoạt | Người dùng đăng nhập và chọn các chức năng trong mục tạo đề thi |
| Chuỗi sự kiện | * Chọn chức năng "Tạo đề thi”: hệ thống sẽ hiển thị các chức năng con trong mục tạo đề thi gồm đặt tên đề, chọn môn học, chỉnh sửa số câu hỏi và độ khó , lưu đề vào trang quản lý đề thi . * Hiển thị đề thi. * Người dùng nhập thông tin đề thi mới nhập . |

**Bảng 3.5.4 Đặc tả Use case Tạo đề thi**

**3.5.5 Use Case quản lý đề thi**

|  |  |
| --- | --- |
| Tên Use case | Quản lý đề thi |
| Tên tác nhân chính | User |
| Mức | 1 |
| Người chịu trách nhiệm | User |
| Điều kiện tiên quyết | User đã đăng nhập |
| Đảm bảo tối thiểu | Lưu được đề thi vào trang quản lý |
| Đảm bảo thành công | Xem chỉnh sửa được đề thi sau khi lưu vào kho |
| Kích hoạt | Người dùng đăng nhập và chọn các chức năng trong mục quản lý đề thi |
| Chuỗi sự kiện | * Chọn chức năng "Quản lý đề thi”: hệ thống sẽ hiển thị các chức năng con trong mục quản lý đề thi gồm chỉnh sửa đề, xóa đề . |

**Bảng 3.5.5 Đặc tả Use case Tạo đề thi**

**Chương 4: THIẾT KẾ GIAO DIỆN**

**4.1 Mục tiêu và nghiệp vụ**

**-** Giao diện của hệ thống quản lý ngân hàng câu hỏi được thiết kế với các mục tiêu chính:

* Thân thiện với người dùng: Đảm bảo giao diện dễ sử dụng, phù hợp với đối tượng sử dụng bao gồm học sinh, giảng viên và các nhân sự khác.
* Tính nhất quán: Đảm bảo tính nhất quán giữa các màn nhằm tránh sự nhầm lẫn
* Đơn giản và trực quan: Các thành phần được bố trí khoa học, ưu tiên những tác vụ thường xuyên sử dụng.
* Màu sắc dễ nhìn, tối giản.
* Thanh điều hướng nằm ở đầu mỗi trang gồm các thành phần chính như logo, các trang, nút đăng xuất người dùng.

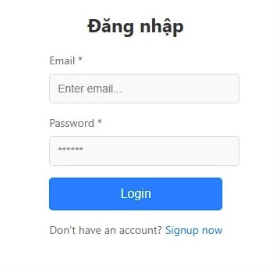
- Hệ thống quản lý ngân hàng câu hỏi phải đáp ứng được đủ các nghiệp vụ:

* Quản lý câu hỏi: Hiển thị thanh tìm kiếm, câu hỏi , thanh sắp xếp, độ khó, môn học.
* Quản lý đề thi : Hiển thị danh sách đề thi
* Quản lý môn học: Hiển thị danh sách về số tên môn, mã môn, phạm vi kiến thức.
* Tạo đề thi : Người dùng có thể tạo đề theo ý muốn
* - Với những mục tiêu trên, nhóm chúng em đã họp và đưa ra những quyết định thiết kế đáp ứng đúng, đủ các tiêu chí cũng như yêu cầu của bài toán. Đảm bảo giao diện thân thiên với người dùng cũng như gọn gàng và dễ nhìn.

**4.2 Mô tả các giao diện chính**

1. **Trang Đăng Nhập Xác Thực Người Dùng**

Trang đăng nhập là nơi người dùng nhập thông tin đăng nhập để truy cập vào hệ thống.

****

**Hình 27. Trang đặng nhập**

**2. Trang Quản Lý Môn Học**

Đây là giao diện quản lý các môn học được người dùng tạo trê hệ thống.

Tích hợp chức năng tìm kiếm và phân trang để dễ dàng quản lý khi số lượng môn học lớn.

**A screenshot of a computer

Description automatically generated**

**Hình 29. Trang quản lý môn học**

**3. Trang Quản Lý Câu hỏi**

Là nơi người dùng có thể theo dõi và quản lý danh sách câu hỏi trong hệ thống.

Bảng dữ liệu hiển thị các thông tin chính:Nội dung, độ khó, hành động , id và các nút hành động: chỉnh sửa và xoá câu hỏi.

Chức năng chính: Thêm mới câu hổi, tìm kiếm theo nội dung câu hỏi, xoá câu hỏi, chỉnh sửa câu hỏi.

**A screenshot of a computer

Description automatically generated**

**Hình 30. Trang quản lý Câu hỏi**

**5.Trang quản lý đề thi**

**Cung cấp thông tin về đề thi:**

+ Tên đề

+Môn học

+Nút chỉnh sửa

+ Nút xóa

**A screenshot of a computer

Description automatically generated**

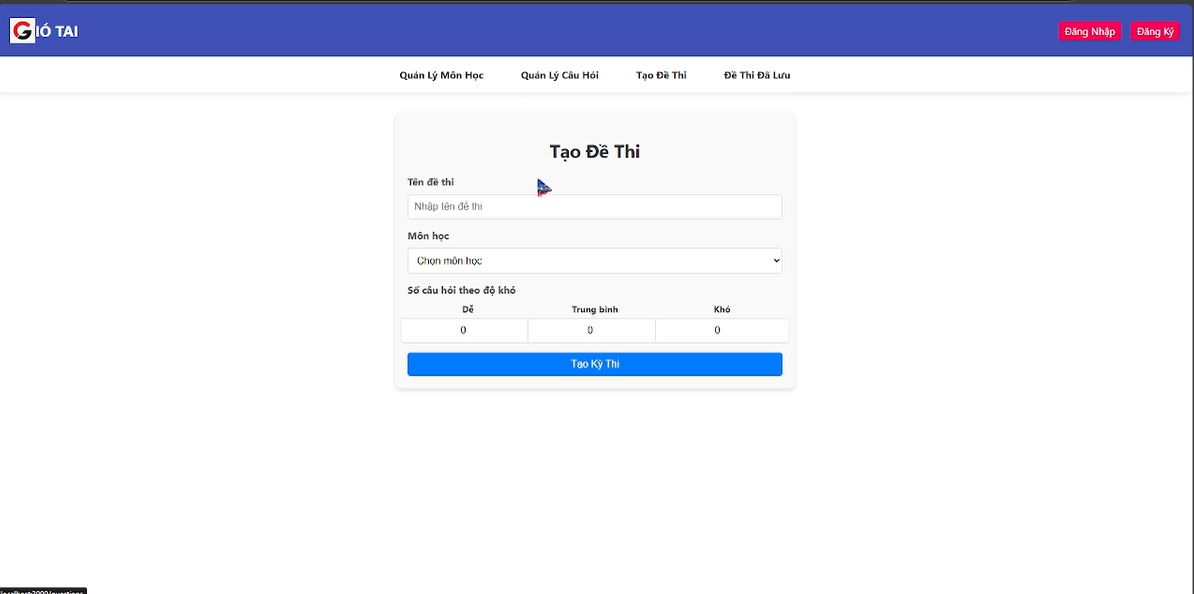
**Hình 31. Trang quản lý đề thi**

**5. Trang Tạo đề thi**

+ Cho phép tạo đề thi theo form ở bảng tạo đề như:

+ Tên đề thi , môn học , số câu hỏi, và mức độ khó.

+ Chức năng chính là tạo đề thi theo yêu cầu.

****

**Hình 32. Trang Tạo đề thi**

**CHƯƠNG 5: KẾT LUẬN**

Hệ thống quản lý ngân hàng câu hỏi là một giải pháp hiệu quả hỗ trợ các cơ sở giáo dục quản lý thông tin môn học, câu hỏi và đề thi một cách dễ dàng và chính xác. Bằng cách tự động hóa các quy trình quản lý, hệ thống giúp giảm thiểu thời gian, công sức, và các thủ tục thủ công phức tạp, từ đó nâng cao hiệu suất làm việc và chất lượng giảng dạy.

Hệ thống không chỉ cung cấp các chức năng cơ bản như quản lý môn học, câu hỏi, đề thi mà còn hỗ trợ tìm kiếm, thống kê và báo cáo thông tin một cách nhanh chóng và chính xác. Việc ứng dụng công nghệ vào quản lý câu hỏi và tạo đề thi sẽ giúp cải thiện đáng kể hiệu quả luyện tập và đánh giá, tạo ra giá trị lớn hơn cho người dạy và người học.

Web đã đáp ứng các chức năng chính như:

* **Quản lý môn học**
* **Quản lý câu hỏi**
* **Quản lý đề thi**
* **Tạo đề thi tự động**

Website vận hành tốt, đáp ứng đầy đủ các yêu cầu đã đề ra. Các module quản lý đều hỗ trợ đầy đủ các chức năng như thêm, sửa, xóa và tìm kiếm dữ liệu.

**Sản phẩm đạt được:**

* Một website quản lý ngân hàng câu hỏi với đầy đủ các tính năng cần thiết.
* Hỗ trợ tự động hóa quá trình tạo đề thi với cấu trúc định trước.
* Cung cấp báo cáo tổng quan về hoạt động quản lý.

**Ưu điểm:**

* Giao diện thân thiện, dễ sử dụng.
* Đơn giản nhưng đáp ứng đầy đủ nhu cầu quản lý ngân hàng câu hỏi.
* Có tính ứng dụng thực tế cao.
* Áp dụng các công nghệ hiện đại.
* Tính năng xác thực và phân quyền giúp bảo mật dữ liệu.
* Sử dụng Validate để kiểm tra tính hợp lệ của dữ liệu, tránh xử lý các yêu cầu không hợp lệ.

**Khuyết điểm:**

* Chưa tối ưu hóa cơ sở dữ liệu.
* Chưa hỗ trợ report với nhiều loại file khác nhau hoặc cấu trúc file phức tạp.

**Hướng phát triển:**

Hiện tại, website đã hỗ trợ tốt cho việc quản lý ngân hàng câu hỏi, và trong tương lai có thể phát triển thêm:

* Mở rộng thêm các tính năng hỗ trợ việc quản lý hiệu quả hơn.
* Nâng cao tính bảo mật cho hệ thống.
* Tối ưu hóa cơ sở dữ liệu.
* Cải thiện tính năng report, hỗ trợ nhiều loại file với cấu trúc đơn giản hơn.

Hệ thống quản lý ngân hàng câu hỏi không chỉ đáp ứng nhu cầu hiện tại mà còn là nền tảng để phát triển thêm nhiều tính năng hữu ích, hỗ trợ tối đa cho quá trình học tập và giảng dạy trong tương lai

**TÀI LIỆU THAM KHẢO**

1. [**https://www.npmjs.com/**](https://www.npmjs.com/) **- Tra cứu tài liệu thư viện**
2. [**https://chatgpt.com/**](https://chatgpt.com/) **- Tra cứu thắc mắc cần giải đáp**