|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  **KHOA CÔNG NGHỆ THÔNG TIN**  A blue and yellow logo  Description automatically generated  **ĐỒ ÁN TỐT NGHIỆP**  ĐỀ TÀI  **QUẢN LÝ ĐIỂM SINH VIÊN**   |  |  | | --- | --- | | Giảng viên hướng dẫn | : TS. Nguyễn Trọng Phúc | | Sinh viên thực hiện | : Bùi Ngọc Tiến | | Lớp | : CNTT3 – K61 | | Mã sinh viên | : 201207598 |   **Hà Nội – 2024** |

**LỜI CẢM ƠN**

*Để đạt được kết quả của ngày hôm nay là một quá trình kéo dài trong suốt hơn bốn năm em học tập và rèn luyện tại nhà trường, bên cạnh sự cố gắng nỗ lực của bản thân, em xin bày tỏ lòng biết ơn sâu sắc đến nhà trường, quý thầy cô và tất cả những người đã ân cần hướng dẫn động viên và hỗ trợ em trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp này.*

*Đặc biệt, em xin gửi lời cảm ơn chân thành tới thầy Nguyễn Trọng Phúc, người đã trực tiếp hướng dẫn em trong quá trình thực hiện đồ án. Những kinh nghiệm, những kiến thức sâu rộng và lòng nhiệt huyết của thầy là nguồn động lực vô cùng quý báu giúp em vượt qua những thách thức và phát triển không ngừng trong quá trình nghiên cứu và hoàn thiện đồ án.*

*Cuối cùng, em muốn gửi lời cảm ơn đến gia đình, bạn bè và tất cả những người thân yêu đã luôn đồng hành cùng em trong suốt quá trình này. Đó như là một sức mạnh to lớn giúp em vững tin hơn trong công việc cũng như trong cuộc sống.*

*Em xin kính chúc nhà trường, quý thầy cô và tất cả mọi người sức khỏe dồi dào, hạnh phúc và thành công.*

*Trân trọng,*

|  |  |
| --- | --- |
|  | *Hà Nội, ngày … tháng … năm 2024* |
|  | *Sinh viên thực hiện* |
|  | *Bùi Ngọc Tiến* |

**DANH MỤC CÁC TỪ VIẾT TẮT**

1. API - Application Programming Interface: là giao diện lập trình ứng dụng, nơi trao đổi dữ liệu giữa các thành phần trong hệ thống và giữa các hệ thống với nhau.

2. HTTP - Hyper Text Transfer Protocol: là giao thức truyền tải siêu văn bản được sử dụng rộng rãi, giao thức này cho phép tìm và nạp các tài nguyên cho ứng dụng.

3. RESTful API: là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web Services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

4. SOLID: là viết tắt của 5 chữ cái đầu trong 5 nguyên tắc thiết kế hướng đối tượng. Giúp cho lập trình viên viết ra những đoạn code dễ đọc, dễ hiểu, dễ maintain, 5 nguyên tắc đó bao gồm:

- **S**ingle responsibility priciple (SRP)

- **O**pen/Closed principle (OCP)

- **L**iskov substitution principe (LSP)

- **I**nterface segregation principle (ISP)

- **D**ependency inversion principle (DIP)

5. Domain: là một địa chỉ duy nhất trên internet được sử dụng để xác định và truy cập vào một trang web hoặc tài nguyên trực tuyến khác

6. Website: là tập hợp các trang chứa thông tin bao gồm văn bản, hình ảnh, video, dữ liệu… nằm trên một domain, được lưu trữ trên máy chủ web. Website có thể được người dùng truy cập từ xa thông qua mạng internet.

7. UTC - University of Transport and Communications: là từ viết tắt của Trường Đại học Giao thông Vận tải.

**DANH MỤC BẢNG BIỂU**

**DANH MỤC HÌNH ẢNH**

**LỜI NÓI ĐẦU**

Trong hệ thống giáo dục hiện đại, việc quản lý và theo dõi điểm số sinh viên đóng một vai trò quan trọng trong việc đảm bảo chất lượng giáo dục và phát triển bền vững của các cơ sở giáo dục. Chính vì lẽ đó, việc chọn đề tài “Quản lý điểm sinh viên” cho đồ án tốt nghiệp của em không chỉ là một quyết định tự nhiên mà còn là sự cam kết đối với sứ mệnh nâng cao chất lượng giáo dục trong cộng đồng.

Một trong những lí do quan trọng mà em chọn đề tài này là để giải quyết một trong những thách thức chính của các cơ sở giáo dục hiện nay: cải thiện quy trình quản lý điểm số sinh viên một cách hiệu quả và minh bạch. Quản lý điểm số không chỉ là việc ghi chép và tính toán điểm số của sinh viên, mà còn liên quan đến việc phân phối công bằng các tài nguyên giáo dục và cung cấp phản hồi xây dựng cho sinh viên. Hiểu rõ những thách thức và cơ hội mà việc quản lý điểm số mang lại, em hy vọng rằng đồ án của mình sẽ cung cấp các giải pháp và công cụ hữu ích cho việc tối ưu hóa quy trình này.

Ngoài ra, đề tài “Quản lý điểm sinh viên” cũng phản ánh sự quan tâm của em đối với việc áp dụng công nghệ vào giáo dục. Trong thời đại số hóa ngày nay, việc sử dụng các công nghệ thông tin và truyền thông tiên tiến có thể giúp tối ưu hóa quy trình quản lý điểm số, từ việc nhập điểm, xử lý dữ liệu đến việc phân phối thông tin điểm số cho sinh viên và giảng viên. Em tin rằng việc phát triển một hệ thống quản lý điểm số thông minh và dễ sử dụng sẽ giúp nâng cao trải nghiệm học tập và giảng dạy trong cộng đồng giáo dục.

Cuối cùng, đồ án này cũng mang lại cho em cơ hội để phát triển kỹ năng lập trình và quản lý dự án. Việc thiết kế và triển khai một hệ thống quản lý điểm số đòi hỏi sự kết hợp giữa kiến thức về lập trình và kỹ năng quản lý dự án để đảm bảo tính hiệu quả và tính linh hoạt của hệ thống. Em rất háo hức với cơ hội này để áp dụng những kiến thức và kỹ năng của mình vào một dự án thực tế và có ý nghĩa.

**CHƯƠNG 1: GIỚI THIỆU VÀ KHẢO SÁT**

1.1. Giới thiệu chung về bài toán quản lý:

Quản lý là một khái niệm rộng và quan trọng trong mọi lĩnh vực của đời sống, từ kinh tế, xã hội đến giáo dục. Bài toán quản lý bao gồm việc tổ chức, điều hành và kiểm soát các nguồn lực để đạt được mục tiêu đã đề ra một cách hiệu quả và hiệu suất nhất. Các nguồn lực này có thể là con người, tài chính, thông tin, thời gian và công nghệ. Quản lý tốt không chỉ giúp các tổ chức và cá nhân hoạt động một cách trơn tru và có hiệu quả mà còn thúc đẩy sự phát triển bền vững và lâu dài.

Trong bất kỳ hệ thống nào, quản lý hiệu quả đòi hỏi sự kết hợp của nhiều yếu tố như kế hoạch chiến lược, cơ cấu tổ chức, quy trình làm việc, và công cụ hỗ trợ. Sự phát triển của công nghệ thông tin đã mang lại những thay đổi mạnh mẽ trong cách thức quản lý, giúp tự động hóa và tối ưu hóa nhiều quy trình phức tạp, từ đó nâng cao năng suất và giảm thiểu sai sót.

1.2. Bài toán quản lý điểm sinh viên:

Quản lý điểm sinh viên là một phần quan trọng của quản lý trong hệ thống giáo dục. Điểm số không chỉ đánh giá hiệu suất học tập của sinh viên mà còn là một phương tiện để cung cấp phản hồi cho sinh viên và giáo viên về tiến độ và chất lượng của quá trình học tập. Tuy nhiên, bài toán quản lý điểm sinh viên đặt ra nhiều thách thức do tính phức tạp và đa dạng của quá trình đánh giá học tập.

Đầu tiên, quản lý điểm sinh viên đòi hỏi sự minh bạch và công bằng. Hệ thống điểm số phải đảm bảo rằng mỗi sinh viên được đánh giá dựa trên năng lực và thành tích riêng của họ, đồng thời cung cấp phản hồi rõ ràng và công bằng về kết quả đạt được.

Thứ hai, quản lý điểm cần phải linh hoạt và thích ứng. Hệ thống điểm số phải có khả năng điều chỉnh và thích nghi với các yêu cầu đánh giá khác nhau của từng khoa, từng môn học và từng loại hình kiểm tra. Nó cũng cần có khả năng theo dõi tiến trình học tập của sinh viên theo thời gian và cung cấp phản hồi liên tục.

Cuối cùng, quản lý điểm sinh viên đặt ra thách thức về tính bảo mật và quản lý dữ liệu. Bảo mật thông tin cá nhân của sinh viên và đảm bảo tính toàn vẹn của dữ liệu điểm số là yếu tố không thể thiếu trong mọi hệ thống quản lý điểm.

1.3. Tên đề tài:

- Tên đề tài: Quản lý điểm sinh viên.

- Tên ứng dụng: Quản lý điểm sinh viên Trường đại học Kevin.

1.4. Quy mô:

- Quy mô: Một Trường đại học ở Việt Nam.

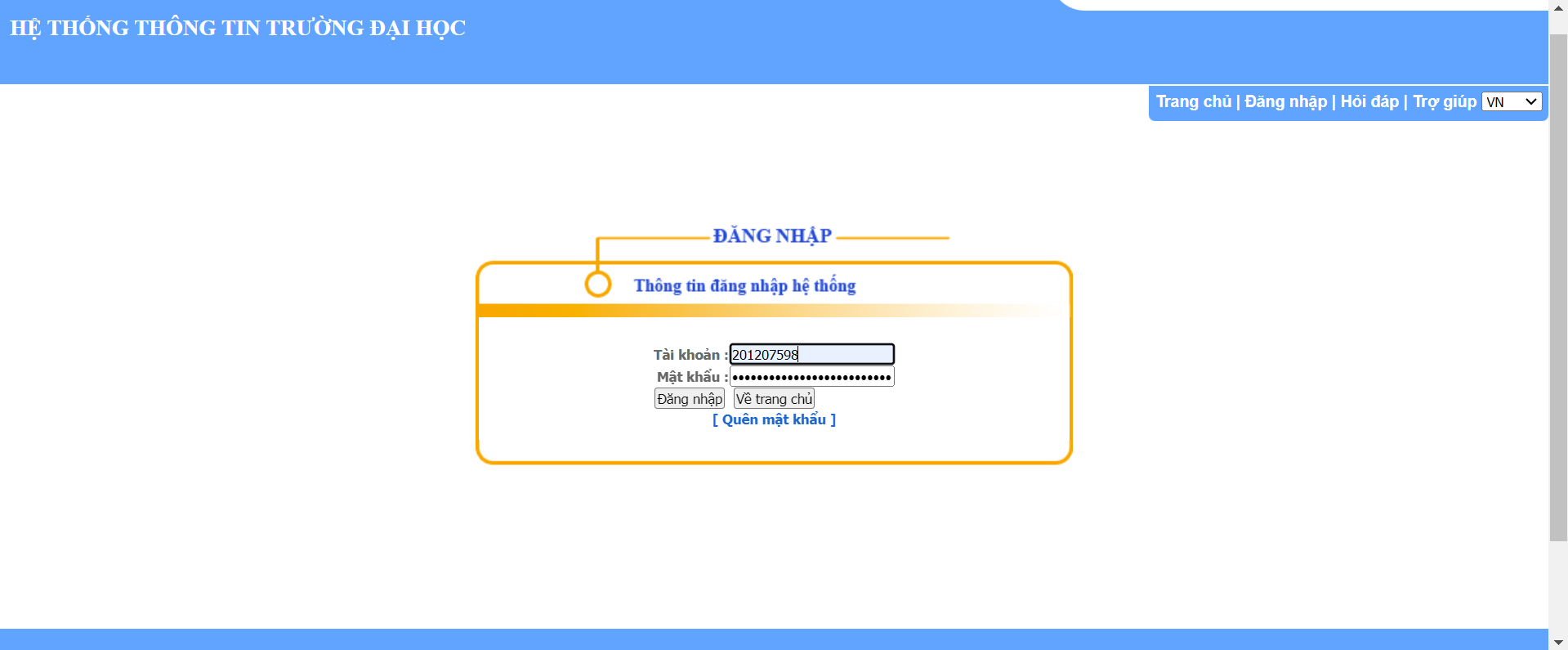
1.5. Khảo sát:

1.5.1. Đối tượng khảo sát:

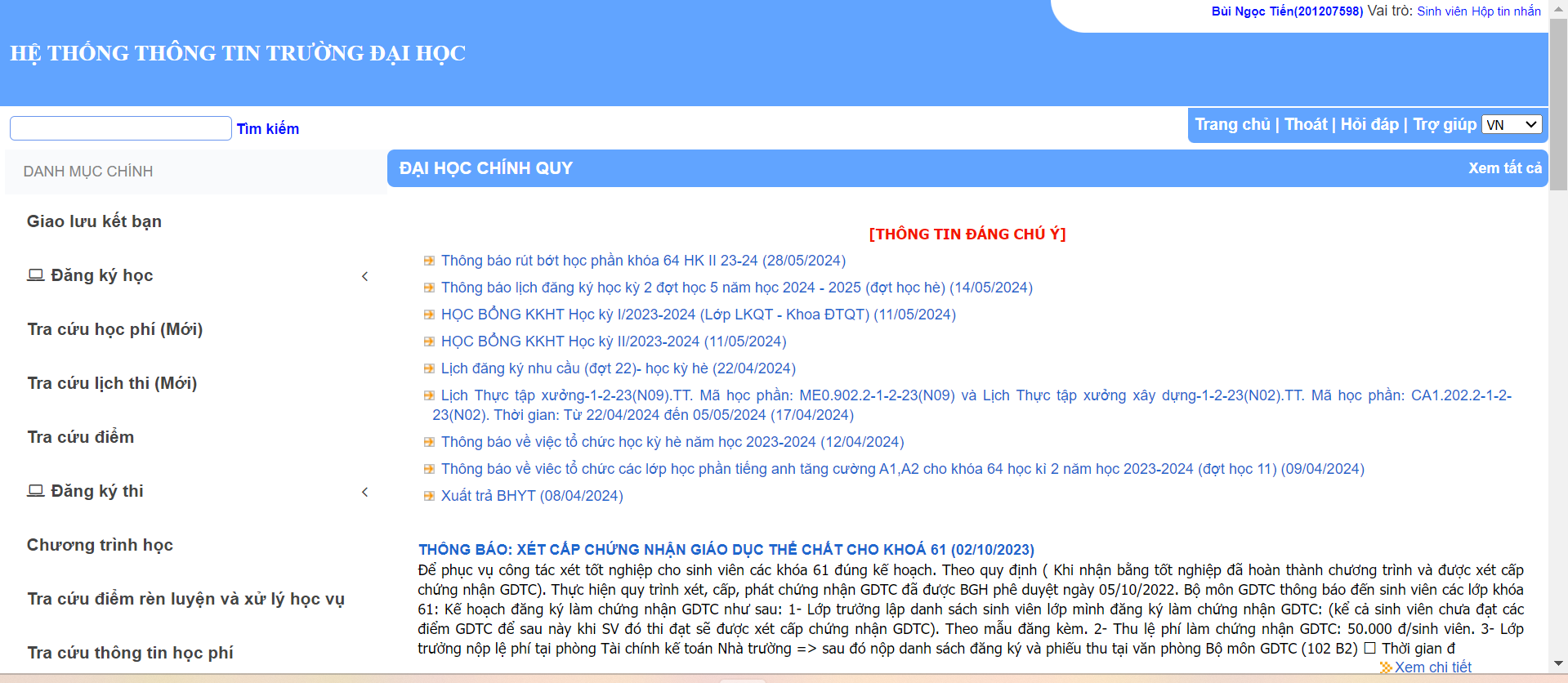
- Khảo sát về các ứng dụng quản lý đào tạo, quản lý điểm của các Trường đại học ở Việt Nam.

1.5.2. Kết quả khảo sát:

- Website quản lý đào tạo Trường đại học Giao thông vận tải (https://qldt.utc.edu.vn):



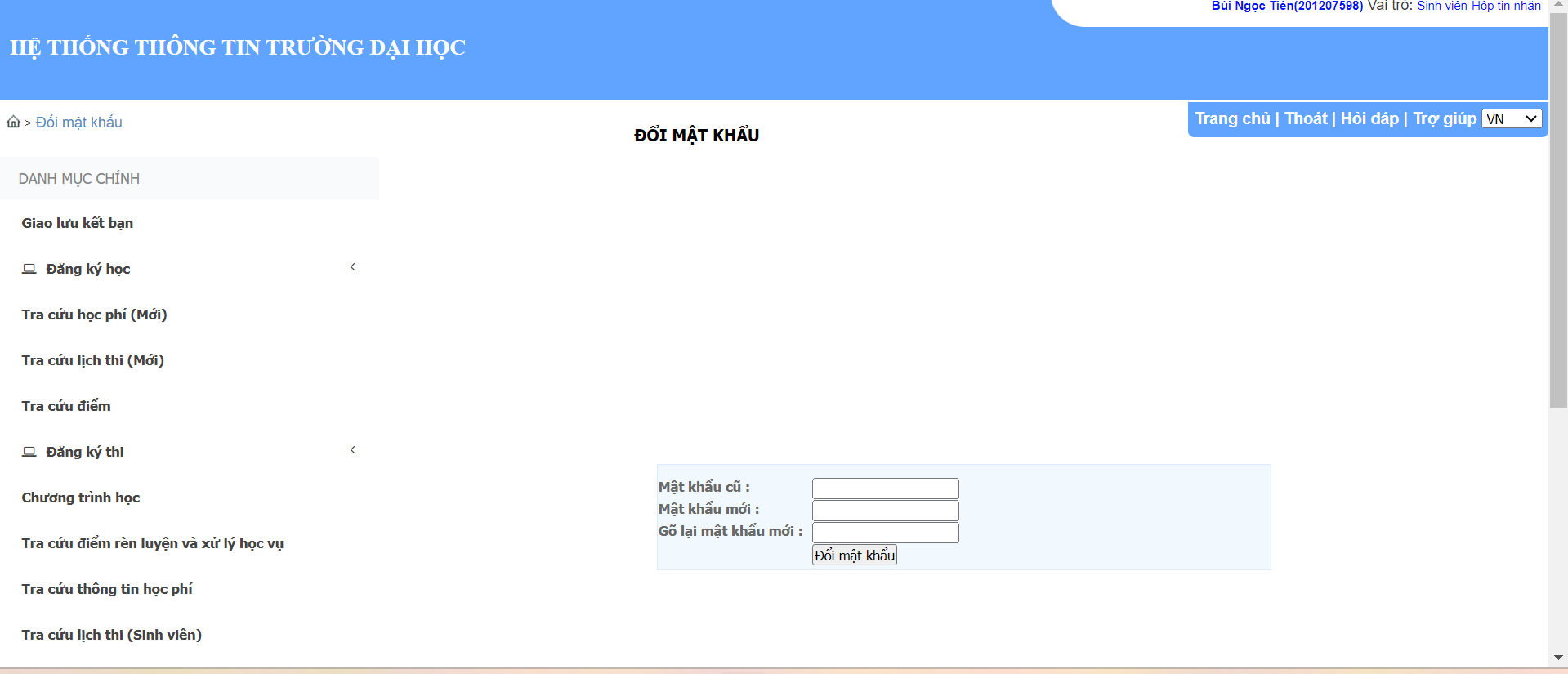
**Hình 1.1** Giao diện đăng nhập website quản lý đào tạo UTC



**Hình 1.2** Giao diện trang chủ website quản lý đào tạo UTC

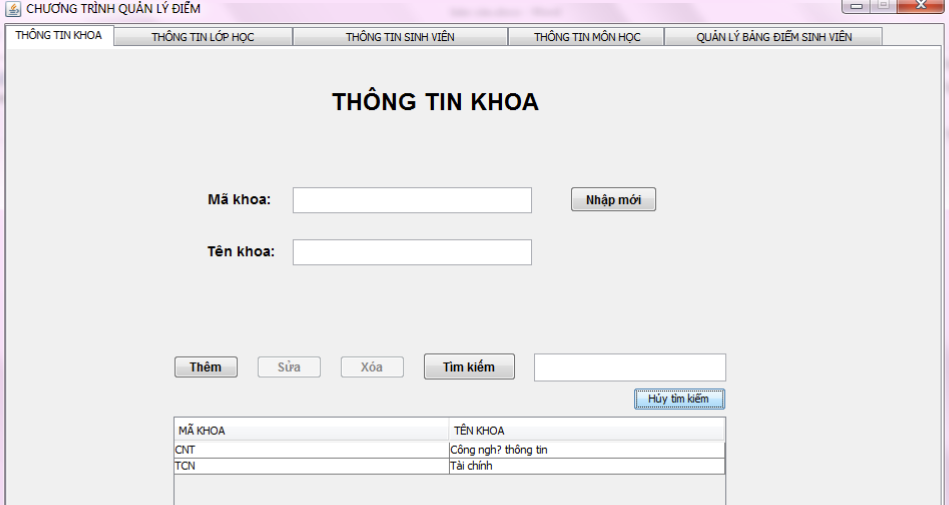


**Hình 1.3** Màn hình tra cứu điểm website quản lý đào tạo UTC (quyền sinh viên)

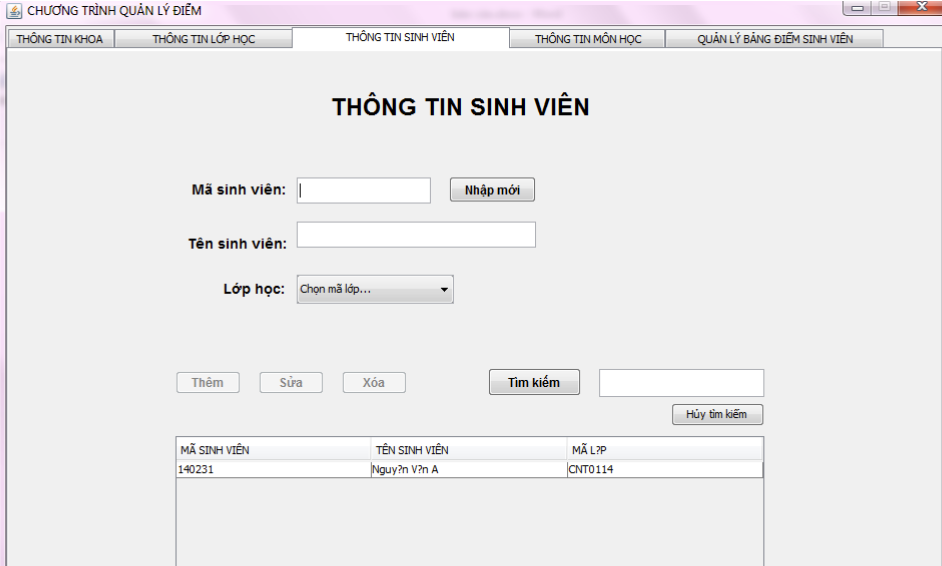


**Hình 1.4** Màn hình chức năng đổi mật khẩu website quản lý đào tạo UTC

- Hệ thống quản lý điểm sinh viên - Aptech Việt Nam (https://aptechvietnam.com.vn):



**Hình 1.5** Màn hình quản lý thông tin khoa của hệ thống Aptech



**Hình 1.6** Màn hình quản lý thông tin sinh viên của hệ thống Aptech



**Hình 1.7** Màn hình tính điểm tổng kết cho sinh viên theo môn của hệ thống Aptech

1.6. Kết luận - Đưa ra bài toán:

- Thiết kế ứng dụng có giao diện đơn giản, màu sắc không cần quá nổi bật, màn hình danh sách, chi tiết các chức năng đồng bộ với nhau giúp người dùng dễ dàng sử dụng.

- Tập trung chủ yếu vào việc xử lí logic, tính toán liên quan đến điểm số để đảm bảo tính minh bạch, tin cậy của ứng dụng.

- Xử lí đăng nhập và phân quyền cho người dùng, đảm bảo với các quyền hạn khác nhau sẽ được sử dụng các chức năng tương ứng của chương trình.

- Các chức năng cơ bản mà ứng dụng cần đáp ứng được:

+ Đăng nhập và phân quyền cho người dùng.

+ Quản lý thông tin sinh viên, giảng viên.

+ Quản lý khoa, môn học, lớp học, lớp học phần.

+ Quản lý điểm, tính toán điểm số, xếp loại sinh viên.

+ Vẽ biểu đồ báo cáo, thống kê về sinh viên và điểm số.

+ Quản lý tài khoản đăng nhập của người dùng.

**CHƯƠNG 2: GIỚI THIỆU VỀ CÔNG NGHỆ**

2.1. Frontend:

2.1.1. HTML - HyperText Markup Language:

- HTML là ngôn ngữ đánh dấu chuẩn được sử dụng để tạo ra và cấu trúc các trang web trên Internet. Được phát triển bởi Tim Berners-Lee vào cuối những năm 1980 và đầu những năm 1990, HTML là một trong những công nghệ cốt lõi của World Wide Web, cùng với CSS (Cascading Style Sheets) và JavaScript.

- HTML hoạt động bằng cách sử dụng các thẻ (tags) để đánh dấu các thành phần khác nhau của một trang web. Mỗi thẻ HTML được đặt trong dấu ngoặc nhọn và thường đi theo cặp, bao gồm một thẻ mở và một thẻ đóng. Các thẻ này được sử dụng để xác định các phần tử như đoạn văn (paragraph), tiêu đề (heading), liên kết (link), hình ảnh (image), bảng biểu (table), và nhiều thành phần khác.

- Một số đặc điểm nổi bật:

+ Cấu trúc trang web: HTML cung cấp một cách thức để tổ chức nội dung của trang web theo một cấu trúc rõ ràng và hợp lý, giúp trình duyệt web hiểu và hiển thị nội dung một cách chính xác.

+ Liên kết tài liệu: HTML cho phép tạo các liên kết đến các tài liệu khác, giúp dễ dàng điều hướng giữa các trang web.

+ Hỗ trợ đa phương tiện: HTML hỗ trợ nhúng hình ảnh, video, âm thanh và các tệp đa phương tiện khác vào trang web.

+ Tương thích với CSS và JavaScript: HTML thường được sử dụng kết hợp với CSS để định dạng và bố trí trang web, cũng như với JavaScript để tạo ra các trang web động và tương tác.

- HTML đã trải qua nhiều phiên bản phát triển, với HTML5 là phiên bản mới nhất, cung cấp nhiều tính năng cải tiến và hỗ trợ tốt hơn cho các ứng dụng web hiện đại. HTML5 giới thiệu các thẻ và thuộc tính mới, cải thiện hỗ trợ đa phương tiện, và cung cấp các API mạnh mẽ để xử lý đồ họa, lưu trữ cục bộ, và nhiều tính năng khác.

2.1.2. CSS - Cascading Style Sheets:

- CSS là ngôn ngữ dùng để mô tả cách thức trình bày và bố cục của các phần tử HTML trên trang web. CSS được phát triển bởi W3C (World Wide Web Consortium) và đã trở thành một phần không thể thiếu trong việc thiết kế và xây dựng các trang web hiện đại.

- CSS cho phép các nhà phát triển web tách biệt nội dung và hình thức của trang web, giúp việc quản lý và bảo trì trang web trở nên dễ dàng và hiệu quả hơn. Thay vì phải định dạng từng phần tử HTML một cách thủ công, CSS cung cấp các quy tắc (rules) để áp dụng định dạng đồng nhất cho các phần tử tương tự.

- Một số đặc điểm nổi bật:

+ Tách biệt nội dung và trình bày: CSS cho phép tách biệt nội dung HTML khỏi phần trình bày, giúp cải thiện cấu trúc và tính nhất quán của trang web.

+ Bố trí và định dạng: CSS cung cấp các công cụ mạnh mẽ để bố trí và định dạng trang web, từ việc điều chỉnh khoảng cách giữa các phần tử đến việc tạo các layout phức tạp.

+ Khả năng tái sử dụng: Các quy tắc CSS có thể được tái sử dụng trên nhiều trang web, giúp tiết kiệm thời gian và công sức khi phát triển và duy trì trang web.

+ Hỗ trợ đa nền tảng: CSS cho phép trang web hiển thị tốt trên nhiều thiết bị và kích thước màn hình khác nhau, từ máy tính để bàn đến điện thoại di động, thông qua các kỹ thuật như responsive design.

- CSS được tổ chức theo ba cách chính:

+ Inline CSS: Các quy tắc CSS được viết trực tiếp trong các thẻ HTML, sử dụng thuộc tính style.

+ Internal CSS: Các quy tắc CSS được viết trong phần <style> của tài liệu HTML.

+ External CSS: Các quy tắc CSS được viết trong một tệp riêng biệt và được liên kết với tài liệu HTML thông qua thẻ <link>.

- CSS3 là phiên bản mới nhất của CSS, mang lại nhiều tính năng cải tiến và hỗ trợ các hiệu ứng phức tạp như chuyển động (animations), bóng (shadows), gradient, và nhiều hơn nữa. CSS3 giúp tạo ra các trang web hiện đại và trực quan, mang lại trải nghiệm người dùng tốt hơn.

2.1.3. JS - JavaScript:

- JS là một ngôn ngữ lập trình phổ biến được sử dụng rộng rãi trong phát triển web để tạo ra các trang web tương tác và động. Được phát triển bởi Brendan Eich tại Netscape vào năm 1995, JavaScript đã trở thành một trong ba công nghệ cốt lõi của World Wide Web, cùng với HTML và CSS.

- JavaScript hoạt động trên phía client, có nghĩa là mã JavaScript được thực thi trên trình duyệt của người dùng, giúp tạo ra các hiệu ứng tương tác và phản hồi ngay lập tức mà không cần phải tải lại toàn bộ trang web. Điều này mang lại trải nghiệm người dùng mượt mà và hiệu quả hơn.

- Một số đặc điểm nổi bật của JavaScript:

+ Tương tác động: JavaScript cho phép tạo ra các trang web tương tác động, như xác thực form, xử lý sự kiện (event handling), và cập nhật nội dung mà không cần tải lại trang.

+ Tương tác với HTML và CSS: JavaScript có thể thay đổi nội dung và kiểu dáng của trang web bằng cách tương tác với các phần tử HTML và CSS, giúp tạo ra các hiệu ứng trực quan và các thay đổi động.

+ Hỗ trợ đa nền tảng: JavaScript hoạt động trên hầu hết các trình duyệt web hiện đại, bao gồm Chrome, Firefox, Safari, và Edge, đảm bảo tính tương thích cao.

+ Khả năng mở rộng: JavaScript có thể được mở rộng thông qua các thư viện và framework, như jQuery, Angular, React, và Vue.js, giúp phát triển các ứng dụng web phức tạp một cách dễ dàng hơn.

2.1.4. VueJS:

- VueJS là một framework JavaScript hiện đại và tiến bộ, được thiết kế để xây dựng các giao diện người dùng tương tác và ứng dụng web đơn trang (Single Page Applications - SPA). VueJS được tạo ra bởi Evan You và phát hành lần đầu vào năm 2014. Kể từ đó, VueJS đã nhanh chóng trở thành một trong những framework phổ biến nhất nhờ vào sự đơn giản, linh hoạt và khả năng mở rộng.

- Một số đặc điểm nổi bật của VueJS:

+ Dễ học và sử dụng: được thiết kế để dễ dàng học và áp dụng, đặc biệt là đối với những người đã quen thuộc với HTML, CSS và JavaScript. Cú pháp đơn giản và tài liệu phong phú giúp người mới bắt đầu nhanh chóng nắm bắt.

+ Tiến bộ và linh hoạt: VueJS có thể được sử dụng cho các dự án nhỏ, cũng như mở rộng để xây dựng các ứng dụng phức tạp lớn. Các thành phần của VueJS có thể được tích hợp dần dần vào các dự án hiện tại mà không cần phải thay đổi toàn bộ cấu trúc.

+ Hệ thống component mạnh mẽ: cho phép xây dựng các ứng dụng từ các component tái sử dụng, giúp quản lý mã nguồn dễ dàng hơn và cải thiện khả năng bảo trì.

+ Reactivity: VueJS có một hệ thống reactivity mạnh mẽ, cho phép tự động cập nhật giao diện người dùng khi dữ liệu thay đổi. Điều này giúp phát triển ứng dụng động một cách dễ dàng và hiệu quả.

+ Ecosystem phong phú: VueJS có một hệ sinh thái phong phú với các thư viện và công cụ bổ sung, như Vue Router (quản lý routing), Vuex (quản lý state), và Nuxt.js (framework phía server). Điều này giúp đơn giản hóa quá trình phát triển và mở rộng ứng dụng.

2.1.5. Giới thiệu về công cụ lập trình Visual Studio Code:

- Visual Studio Code (VS Code) là một công cụ lập trình mã nguồn mở và miễn phí, được phát triển bởi Microsoft. Được phát hành lần đầu vào năm 2015, VS Code nhanh chóng trở thành một trong những trình soạn thảo mã phổ biến nhất hiện nay nhờ vào tính năng mạnh mẽ, hiệu suất cao, và khả năng tùy chỉnh linh hoạt. VS Code hỗ trợ nhiều ngôn ngữ lập trình và framework khác nhau, từ JavaScript, Python, C++, đến HTML và CSS.

- Một số đặc điểm nổi bật của VS Code:

+ Giao diện người dùng thân thiện: VS Code có giao diện người dùng trực quan và dễ sử dụng, với các tính năng như thanh điều hướng bên, thanh trạng thái, và các tab mã nguồn, giúp quản lý và điều hướng giữa các tệp và thư mục dễ dàng.

+ Tính năng IntelliSense: Tính năng IntelliSense của VS Code cung cấp gợi ý mã thông minh, tự động hoàn thành và thông tin về cú pháp, giúp tăng tốc độ lập trình và giảm thiểu lỗi cú pháp.

+ Debugging mạnh mẽ: VS Code tích hợp sẵn công cụ debugging mạnh mẽ, cho phép thiết lập breakpoint, kiểm tra biến, và theo dõi luồng thực thi mã một cách hiệu quả.

+ Hỗ trợ Git tích hợp: VS Code cung cấp hỗ trợ Git tích hợp, cho phép quản lý mã nguồn, thực hiện các thao tác commit, push, pull, và theo dõi sự thay đổi mã trực tiếp từ trình soạn thảo mà không cần chuyển đổi sang giao diện dòng lệnh.

+ Tùy chỉnh linh hoạt: Các extensions cho phép tùy chỉnh môi trường phát triển để phù hợp với nhu cầu cụ thể của dự án, từ kiểm tra mã đến triển khai tự động.

2.2. Backend:

2.2.1. ASP.NET Core Web API:

- ASP.NET Core Web API là một framework phát triển ứng dụng web API được Microsoft phát triển dựa trên nền tảng ASP.NET Core. Được giới thiệu lần đầu tiên vào năm 2016, ASP.NET Core Web API đã trở thành một công cụ quan trọng cho việc xây dựng và triển khai các dịch vụ web API linh hoạt, hiệu quả và mở rộng.

- Một số đặc điểm nổi bật của ASP.NET Core Web API:

+ Cross-platform và Open-source: ASP.NET Core là một framework mã nguồn mở và đa nền tảng, cho phép phát triển và triển khai ứng dụng trên Windows, macOS và Linux.

+ Hiệu suất cao: ASP.NET Core được tối ưu hóa để đạt được hiệu suất tốt nhất, với khả năng xử lý các yêu cầu HTTP hàng ngàn trên mỗi giây.

+ Tích hợp sâu sắc với công nghệ Microsoft: ASP.NET Core Web API tích hợp tốt với các công nghệ khác của Microsoft như Entity Framework Core (ORM), SignalR (real-time web), và Azure (cloud computing), tạo ra một hệ sinh thái mạnh mẽ cho việc phát triển ứng dụng web.

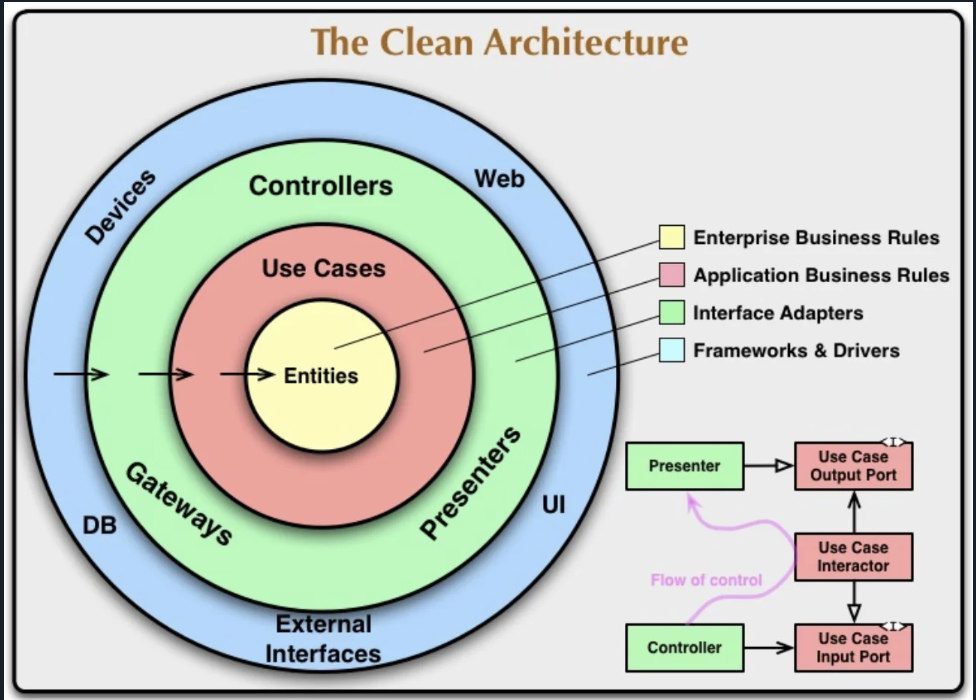
+ Hỗ trợ Dependency Injection (DI): ASP.NET Core cung cấp một container DI tích hợp sẵn, giúp quản lý các thành phần của ứng dụng và tăng tính linh hoạt và dễ bảo trì.

+ Authentication và Authorization: ASP.NET Core Web API hỗ trợ các cơ chế xác thực và phân quyền linh hoạt, cho phép bảo mật dữ liệu và điều khiển quyền truy cập vào các tài nguyên.

+ Testing dễ dàng: ASP.NET Core hỗ trợ việc viết unit test và integration test dễ dàng, giúp đảm bảo chất lượng của mã nguồn và ứng dụng.

+ Tích hợp với công cụ Swagger: ASP.NET Core Web API tích hợp sẵn Swagger UI, cho phép tạo và xem tài liệu API một cách dễ dàng và tự động.

2.2.2. Mô hình Clean Architecture:



**Hình 2.1** Mô hình Clean Architecture

- Mô hình Clean Architecture là một phương pháp thiết kế phần mềm được đề xuất bởi Robert C. Martin (hay còn được gọi là Uncle Bob). Mục tiêu của mô hình này là tạo ra các hệ thống phần mềm linh hoạt, dễ bảo trì và dễ mở rộng bằng cách tách biệt các thành phần của hệ thống và đảm bảo rằng các thành phần này không phụ thuộc vào nhau.

- Mô hình này đặt sự tập trung vào sự phân chia rõ ràng giữa các lớp và các thành phần của hệ thống. Nó đề xuất một cấu trúc chia thành các lớp (layers) và các vòng (circles) với các quy tắc rõ ràng về sự phụ thuộc giữa chúng.

- Cấu trúc của Mô hình Clean Architecture:

+ Entities: Đây là các đối tượng cơ bản của ứng dụng, biểu diễn các khái niệm cốt lõi và logic của doanh nghiệp. Các entities không biết về bất kỳ chi tiết triển khai nào, chúng chỉ đơn giản là các cấu trúc dữ liệu và phương thức đơn giản.

+ Use Cases: Lớp này chứa các use case của hệ thống, tức là các nhiệm vụ cụ thể mà hệ thống cần thực hiện để đáp ứng các yêu cầu từ người dùng. Mỗi use case biểu diễn một tác vụ cụ thể, như thêm một sinh viên mới vào hệ thống hoặc tính toán điểm trung bình của một sinh viên.

+ Interface Adapters: Lớp này chứa các adapter để chuyển đổi dữ liệu giữa các định dạng bên ngoài (như giao diện người dùng, cơ sở dữ liệu, hoặc dịch vụ web) và các định dạng nội bộ của ứng dụng. Adapter cung cấp một giao diện để các use cases có thể tương tác với các thành phần bên ngoài mà không cần biết chi tiết triển khai.

+ Frameworks and Drivers: Lớp này chứa các thành phần bên ngoài như framework web, cơ sở dữ liệu, hoặc bất kỳ công nghệ nào mà hệ thống phụ thuộc vào. Đây là nơi triển khai các chi tiết cụ thể của ứng dụng, nhưng không nên có bất kỳ logic doanh nghiệp nào.

- Một số lợi ích của mô hình Clean Architecture:

+ Tính độc lập và dễ kiểm thử: Các lớp được phân chia rõ ràng và không phụ thuộc vào nhau, điều này làm cho việc kiểm thử và bảo trì dễ dàng hơn.

+ Dễ dàng mở rộng và thay đổi: Cấu trúc linh hoạt của mô hình Clean Architecture cho phép thay đổi hoặc mở rộng một phần của hệ thống mà không cần phải lo lắng về các ảnh hưởng đến các phần khác.

+ Tách biệt logic doanh nghiệp và chi tiết triển khai: Mô hình này tách biệt rõ ràng giữa logic doanh nghiệp và các chi tiết triển khai, giúp tạo ra mã nguồn dễ đọc và dễ bảo trì.

2.2.3. Giới thiệu về Visual Studio:

- Visual Studio là một IDE (Integrated Development Environment - Môi trường Phát triển Tích hợp) phổ biến của Microsoft, được sử dụng rộng rãi trong việc phát triển các ứng dụng và dịch vụ. Visual Studio cung cấp một loạt các công cụ và tính năng giúp nhà phát triển tạo ra các ứng dụng chất lượng cao trên nhiều nền tảng khác nhau, bao gồm desktop, web, di động, và đám mây.

- Tương tự như VS Code, Visual Studio cũng có một số đặc điểm nổi bật như hỗ trợ nhiều ngôn ngữ lập trình, giao diện người dùng thân thiện, tích hợp source control, Azure, hỗ trợ phát triển đa nền tảng.

2.3. Database:

2.3.1. Postgresql:

- Postgresql là một hệ quản trị cơ sở dữ liệu mở rộng, có khả năng xử lý các loại dữ liệu khác nhau với tính năng mạnh mẽ và linh hoạt. Nó được phát triển và duy trì bởi một cộng đồng lớn các nhà phát triển trên khắp thế giới và được công nhận là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất hiện nay.

- Một số tính năng nổi bật:

+ Tính mở rộng và phân phối: PostgreSQL hỗ trợ phân phối dữ liệu trên nhiều máy chủ và máy ảo, cho phép mở rộng hệ thống theo nhu cầu. Nó cung cấp các tính năng như Replication, Sharding và Streaming Replication để đảm bảo tính khả dụng cao.

+ Tiêu chuẩn và tuân thủ: PostgreSQL tuân thủ các tiêu chuẩn SQL và hỗ trợ các tính năng nâng cao như ACID (Atomicity, Consistency, Isolation, Durability), MVCC (Multi-Version Concurrency Control), và các loại dữ liệu phong phú như JSON, XML, và các loại dữ liệu địa lý.

+ Tính linh hoạt và mở rộng: PostgreSQL cho phép tạo ra các kiểu dữ liệu tùy chỉnh, các hàm và trình tự (sequence) tùy chỉnh, và cung cấp một số lượng lớn các loại chỉ mục và hàm để tối ưu hóa hiệu suất của cơ sở dữ liệu.

+ Bảo mật cao: PostgreSQL cung cấp các tính năng bảo mật như SSL, GSSAPI, và LDAP để bảo vệ dữ liệu và đảm bảo tính bảo mật trong quá trình truy cập và truy vấn dữ liệu.

+ Cộng đồng lớn và hỗ trợ tốt: PostgreSQL có một cộng đồng lớn các nhà phát triển và người dùng trên khắp thế giới, cung cấp hỗ trợ thông qua diễn đàn, email list, và các kênh truyền thông xã hội.

+ Tích hợp với các ngôn ngữ lập trình và framework: PostgreSQL có thể tích hợp dễ dàng với các ngôn ngữ lập trình phổ biến như Python, Java, Ruby, và PHP, cũng như với các framework như Django, Ruby on Rails, và Spring.

+ Cập nhật và phát triển liên tục: PostgreSQL có một chu kỳ phát triển nhanh chóng và liên tục, với các phiên bản mới được phát hành thường xuyên, mang lại các tính năng mới và cải thiện hiệu suất.

2.3.2. DBeaver:

- DBeaver là một công cụ quản lý cơ sở dữ liệu đa nền tảng và mã nguồn mở, được phát triển bởi một nhóm nhà phát triển độc lập. DBeaver cung cấp một loạt các tính năng hữu ích cho việc kết nối, truy vấn và quản lý các hệ thống cơ sở dữ liệu khác nhau từ một giao diện đơn giản và trực quan.

- Một số đặc điểm nổi bật:

+ Hỗ trợ nhiều cơ sở dữ liệu: DBeaver hỗ trợ nhiều hệ thống quản lý cơ sở dữ liệu (DBMS) phổ biến như PostgreSQL, MySQL, SQLite, Oracle, SQL Server, và nhiều DBMS khác, cho phép bạn làm việc với các cơ sở dữ liệu khác nhau từ cùng một giao diện.

+ Tính năng truy vẫn mạnh mẽ: DBeaver cung cấp các công cụ truy vấn mạnh mẽ như trình chỉnh sửa và thực thi SQL, biểu đồ ERD (Entity-Relationship Diagram), và công cụ giám sát và quản lý cây thư mục cơ sở dữ liệu.

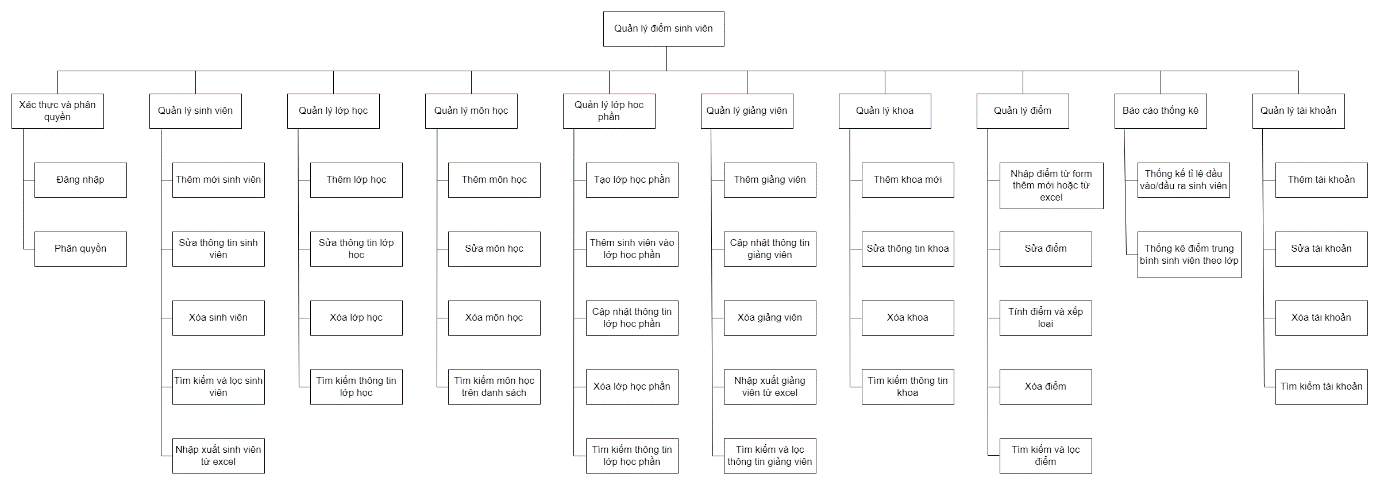
+ Hỗ trợ kết nối bảo mật: DBeaver hỗ trợ kết nối bảo mật với các cơ sở dữ liệu thông qua SSL và SSH, đảm bảo an toàn cho dữ liệu của bạn trong quá trình truyền và truy cập.

+ Mở rộng với các plugin: DBeaver có hệ thống plugin phong phú, cho phép bạn mở rộng chức năng của nó bằng cách cài đặt các plugin từ cộng đồng hoặc phát triển riêng của mình.

+ Mã nguồn mở và miễn phí: DBeaver là một phần mềm mã nguồn mở và miễn phí, cho phép bạn sử dụng và phân phối lại mà không cần phải trả bất kỳ chi phí nào.

**CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ**

3.1. Sơ đồ phân rã chức năng:



**Hình 3.1** Sơ đồ phân rã chức năng

3.2. Chi tiết các chức năng:

3.2.1. Chức năng đăng nhập:

- Chức năng đăng nhập là một trong những chức năng cơ bản và quan trọng nhất của bất kỳ phần mềm quản lý điểm sinh viên nào. Đăng nhập không chỉ giúp bảo vệ dữ liệu cá nhân của người dùng mà còn đảm bảo rằng chỉ những người được ủy quyền mới có thể truy cập vào hệ thống. Việc này ngăn chặn những truy cập trái phép và bảo vệ thông tin nhạy cảm như điểm số, thông tin cá nhân của sinh viên và các tài liệu quan trọng khác.

- Ngoài ra, chức năng đăng nhập còn tạo ra một môi trường làm việc cá nhân hóa cho người dùng. Mỗi giáo viên, sinh viên hoặc quản trị viên sẽ có một tài khoản riêng biệt với các quyền hạn khác nhau, giúp họ thực hiện các nhiệm vụ của mình một cách hiệu quả và an toàn. Chẳng hạn, giáo viên có thể nhập điểm và nhận xét, sinh viên có thể kiểm tra điểm số của mình, còn quản trị viên có thể quản lý và điều chỉnh các thông tin trong hệ thống.

- Hơn nữa, chức năng đăng nhập còn hỗ trợ trong việc theo dõi và ghi lại lịch sử hoạt động của người dùng. Điều này rất quan trọng trong việc xác định trách nhiệm và giải quyết các vấn đề phát sinh. Chẳng hạn, nếu có sự cố về dữ liệu, hệ thống có thể dễ dàng truy xuất xem ai đã thực hiện những thay đổi nào và khi nào chúng xảy ra, từ đó nhanh chóng khắc phục sự cố.

- Do đó, chức năng đăng nhập không chỉ là cửa ngõ bảo mật cho hệ thống mà còn là nền tảng để đảm bảo tính minh bạch, trách nhiệm và hiệu quả trong việc quản lý điểm sinh viên.

3.2.2. Các chức năng thiết yếu: quản lý khoa, quản lý môn học, quản lý lớp học:

- Các chức năng như quản lý khoa, quản lý môn học và quản lý lớp học là những thành phần cốt lõi của phần mềm quản lý điểm sinh viên, đóng vai trò quan trọng trong việc tổ chức và vận hành hệ thống giáo dục một cách hiệu quả.

- Chức năng quản lý khoa giúp theo dõi và quản lý các thông tin liên quan đến từng khoa trong trường, bao gồm thông tin về giảng viên, các chương trình đào tạo, và các hoạt động học thuật. Điều này giúp đảm bảo sự phối hợp chặt chẽ giữa các khoa và nâng cao chất lượng đào tạo.

- Chức năng quản lý lớp học hỗ trợ việc tổ chức và điều phối các lớp học, bao gồm việc phân bổ sinh viên vào các lớp, lập thời khóa biểu, và quản lý danh sách điểm danh. Chức năng này đảm bảo rằng mỗi lớp học được tổ chức một cách hợp lý, giảng viên có thể dễ dàng theo dõi tình hình học tập của sinh viên, và sinh viên có thể nắm rõ lịch học cũng như các hoạt động liên quan.

- Chức năng quản lý môn học cho phép hệ thống lưu trữ và quản lý thông tin chi tiết về từng môn học, từ tên môn học, mã môn học, mô tả, đến số tín chỉ và giáo trình liên quan. Điều này giúp sinh viên và giảng viên dễ dàng tra cứu thông tin, đăng ký môn học và theo dõi tiến trình học tập.

3.2.3. Chức năng quản lý sinh viên, quản lý giảng viên:

- Chức năng quản lý sinh viên và quản lý giảng viên là hai yếu tố không thể thiếu trong phần mềm quản lý điểm sinh viên, giúp hệ thống hoạt động một cách mạch lạc và hiệu quả.

- Chức năng quản lý sinh viên cho phép theo dõi và cập nhật thông tin chi tiết về từng sinh viên, bao gồm thông tin cá nhân, lịch sử học tập, điểm số, và các hoạt động ngoại khóa. Hệ thống này giúp quản trị viên dễ dàng nắm bắt tình hình học tập của sinh viên, đồng thời hỗ trợ sinh viên trong việc tra cứu điểm số, đăng ký môn học, và theo dõi tiến độ học tập. Chức năng này cũng giúp tạo ra các báo cáo thống kê về tình hình học tập của toàn trường, hỗ trợ trong việc ra quyết định và cải thiện chất lượng đào tạo.

- Chức năng quản lý giảng viên cung cấp một công cụ mạnh mẽ để quản lý thông tin của giảng viên, bao gồm hồ sơ cá nhân, quá trình giảng dạy, và lịch sử công tác. Hệ thống này giúp đảm bảo rằng mỗi giảng viên được phân công đúng môn học và lớp học, đồng thời hỗ trợ trong việc lập thời khóa biểu giảng dạy và quản lý khối lượng công việc. Chức năng này cũng giúp giảng viên dễ dàng nhập điểm, nhận xét, và theo dõi tiến độ học tập của sinh viên, từ đó nâng cao chất lượng giảng dạy.

- Vì vậy, chức năng quản lý sinh viên và quản lý giảng viên đóng vai trò quan trọng trong việc tạo ra một môi trường học tập và giảng dạy hiệu quả, đảm bảo rằng mọi thông tin liên quan đến sinh viên và giảng viên đều được tổ chức và quản lý một cách khoa học, thuận tiện cho việc tra cứu và sử dụng.

3.2.4. Chức năng quản lý lớp học phần:

- Chức năng quản lý lớp học phần cho phép nhà trường tổ chức và quản lý các lớp học phần một cách hệ thống và hiệu quả. Chức năng này bao gồm việc lập kế hoạch, phân công giảng viên, quản lý danh sách sinh viên đăng ký, và theo dõi tiến trình học tập của mỗi lớp học phần. Nhờ đó, nhà trường có thể đảm bảo rằng mỗi lớp học phần được sắp xếp hợp lý, giảng viên và sinh viên đều nắm rõ lịch học và yêu cầu của môn học. Điều này cũng giúp giảm thiểu các xung đột lịch học và đảm bảo rằng nguồn lực giảng dạy được sử dụng tối ưu.

3.2.5. Chức năng quản lý điểm:

- Chức năng quản lý điểm là chức năng quan trọng nhất trong phần mềm quản lý điểm sinh viên, đóng vai trò then chốt trong việc đảm bảo tính chính xác, minh bạch và hiệu quả trong quản lý học tập.

- Chức năng này cho phép ghi nhận, lưu trữ và theo dõi điểm số của từng sinh viên một cách chi tiết và chính xác, hỗ trợ giảng viên trong việc nhập điểm, cập nhật kết quả kiểm tra và đánh giá tiến độ học tập của sinh viên. Đối với sinh viên, hệ thống cung cấp một công cụ để theo dõi điểm số của mình, từ đó nắm bắt được tình hình học tập và điều chỉnh phương pháp học tập nếu cần thiết.

3.2.6. Chức năng báo cáo thống kê:

- Chức năng báo cáo thống kê là một chức năng rất hữu ích, nó giúp nhà trường và các cơ sở giáo dục theo dõi và đánh giá kết quả học tập của sinh viên một cách hiệu quả, từ đó cải thiện quy trình quản lý giáo dục. Chức năng này cung cấp nhiều biểu đồ khác nhau như:

+ Thống kê tỉ lệ đầu vào/đầu ra của sinh viên theo từng năm.

+ Thống kê điểm trung bình của sinh viên theo lớp.

3.2.7. Chức năng quản lý tài khoản:

- Chức năng quản lý tài khoản đảm bảo sự an toàn, hiệu quả và tiện lợi trong việc sử dụng hệ thống. Bao gồm các tính năng như:

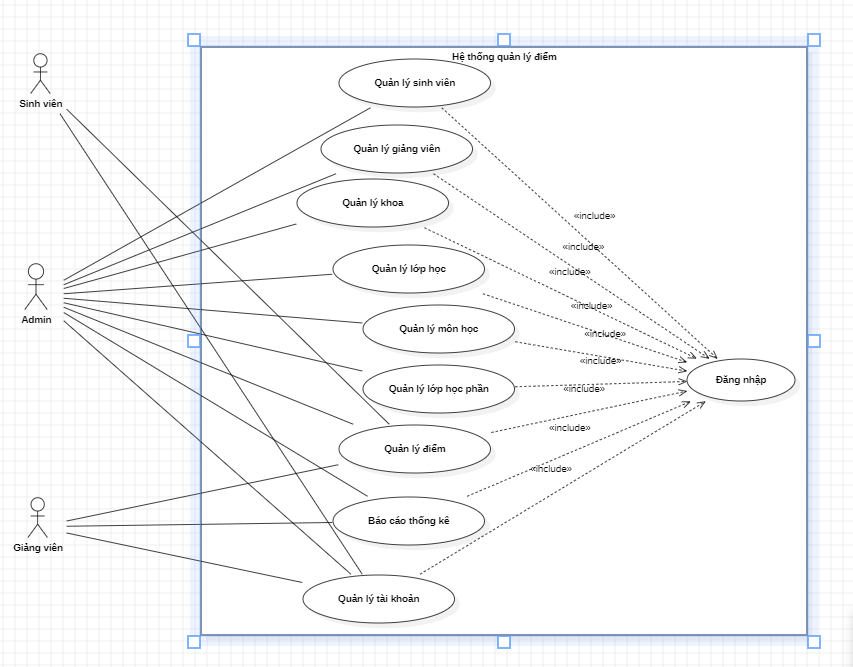
+ Tạo và quản lý tài khoản người dùng

+ Phân quyền người dùng

+ Theo dõi hoạt động người dùng

3.3. Sơ đồ use case:

3.3.1. Sơ đồ use case tổng quát:



**Hình 3.2** Sơ đồ use case tổng quát

3.3.2. Phân tích quy trình sử dụng chương trình:

3.3.2.1. Quy trình sử dụng khi đăng nhập với quyền quản trị viên:

- Quản trị viên thực hiện đăng nhập vào chương trình. Sau khi đăng nhập, quản trị viên có thể xem, quản lý thông tin cá nhân của sinh viên, giảng viên, quản lý các môn học, các khoa, các lớp học phần, quản lý điểm số sinh viên, xem các biểu đồ thống kê liên quan đến điểm của sinh viên, quản lý tài khoản đăng nhập của sinh viên và giảng viên, lấy lại mật khẩu cho họ nếu có yêu cầu.

3.3.2.2. Quy trình sử dụng khi đăng nhập với quyền giảng viên:

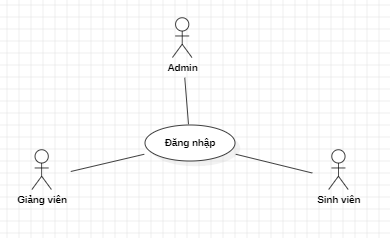
- Giảng viên nhập đúng thông tin đăng nhập được cung cấp để đăng nhập vào chương trình. Sau đó, giảng viên có thể quản lý thông tin tài khoản cá nhân, quản lý điểm sinh viên, nhập điểm các môn mà mình giảng dạy vào hệ thống, ngoài ra, giảng viên cũng có thế xem các biểu đồ thống kê liên quan đến điểm của sinh viên.

3.3.2.3. Quy trình sử dụng khi đăng nhập với quyền sinh viên:

- Sau khi đăng nhập vào chương trình theo tài khoản được nhà trường cung cấp, sinh viên có thể quản lý thông tin cá nhân của mình, xem thông tin về điểm số các môn học đã được nhà trường cập nhật lên hệ thống.

3.3.2.4. Phân tích chi tiết một số use case chính:

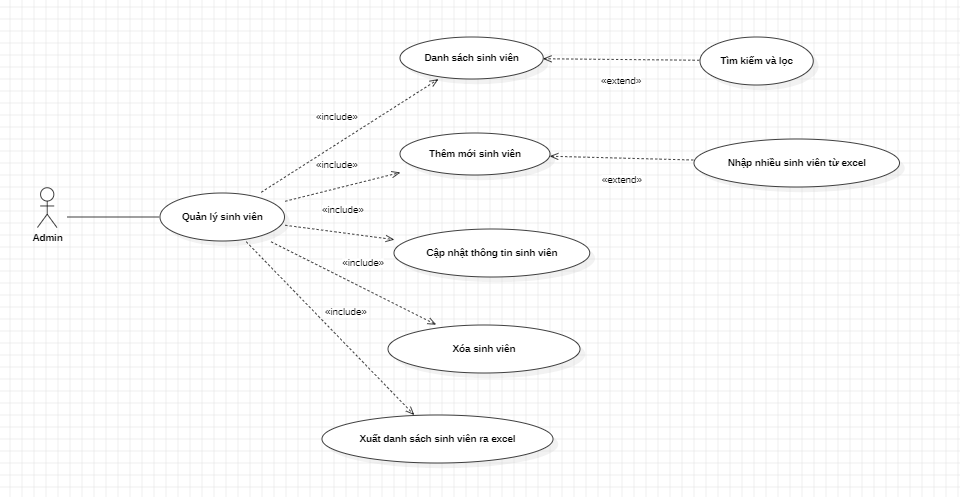
**- Đăng nhập:**

****

**Hình 3.3** Sơ đồ use case đăng nhập

+ Mô tả: Hành động đăng nhập là hành động bắt buộc khi người dùng muốn sử dụng phần mềm quản lý điểm sinh viên. Để đăng nhập thành công, người dùng cần nhập đúng tài khoản và mật khẩu. Đối với quản trị viên, tài khoản đăng nhập sẽ được tạo khi mua ứng dụng, đối với giảng viên và sinh viên, tài khoản sẽ được cấp bởi quản trị viên khi họ đang học tập và giảng dạy ở trường. Trong trường hợp nhập sai tài khoản đăng nhập, hệ thống sẽ show cảnh báo lỗi và yêu cầu nhập lại cho đến khi nhập đúng thông tin đăng nhập. Sau khi đã đăng nhập thành công, chương trình sẽ dựa vào quyền của tài khoản người dùng để chuyển hướng đến các trang tương ứng.

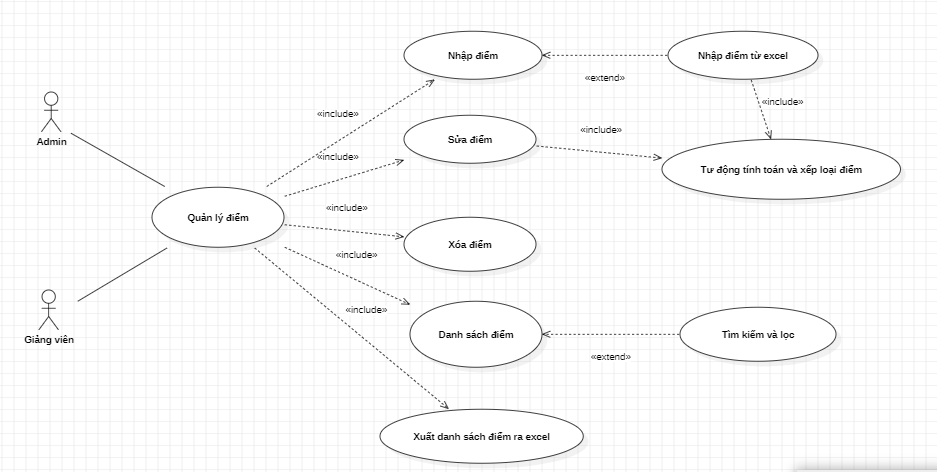
**- Quản lý sinh viên:**

****

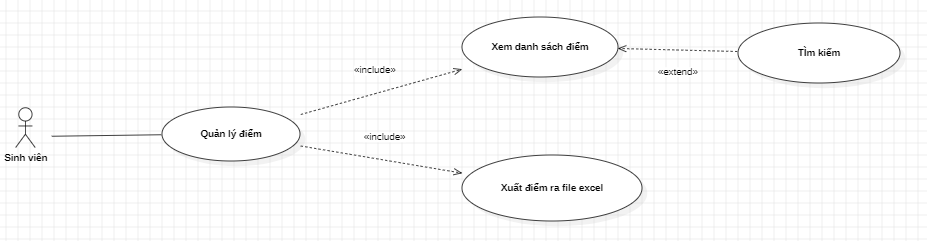
**Hình 3.4** Sơ đồ use case quản lý sinh viên

- Mô tả: Chức năng quản lý sinh viên là chức năng chỉ có người dùng có quyền quản trị viên mới được sử dụng. Khi người dùng chọn vào chức năng này, màn hình danh sách tất cả các sinh viên sẽ được hiển thị theo cơ chế phân trang, có thể là 10, 20, 30, 50 hoặc 100 bản ghi trên 1 trang tùy thuộc vào người dùng lựa chọn. Tại đây, để thuận tiện cho việc tra cứu thông tin sinh viên, chương trình đã cung cấp chức năng tìm kiếm sinh viên theo mã hoặc tên, lọc sinh viên theo lớp, giới tính hoặc địa chỉ. Để thêm sinh viên mới hoặc chỉnh sửa thông tin sinh viên, người dùng cần thực hiện bấm vào chức năng thêm trên giao diện hoặc chức năng sửa ở trên từng dòng của danh sách sinh viên, sau đó chương trình sẽ show form chi tiết để người dùng có thể thực hiện. Trong trường hợp có lỗi xảy ra chẳng hạn như bỏ trống thông tin bắt buộc nhập hoặc mã sinh viên đã tồn tại trong hệ thống … chương trình sẽ show cảnh báo lỗi để hỗ trợ người dùng nhập lại thông tin cho chính xác, sau đó dữ liệu hợp lệ sẽ được lưu vào database. Ngoài ra, chương trình còn hỗ trợ người dùng nhập danh sách sinh viên từ file excel đã chuẩn bị trước để tối ưu quá trình nhập dữ liệu từng bản ghi. Cuối cùng, khi người dùng có nhu cầu xuất các sinh viên theo điều kiện tìm kiếm ra excel, chỉ cần bấm vào biểu tượng excel trên màn hình danh sách để thực hiện.

**- Quản lý điểm:**

****

**Hình 3.5** Sơ đồ use case quản lý điểm với quyền quản trị viên và giảng viên



**Hình 3.6** Sơ đồ use case quản lý điểm với quyền sinh viên

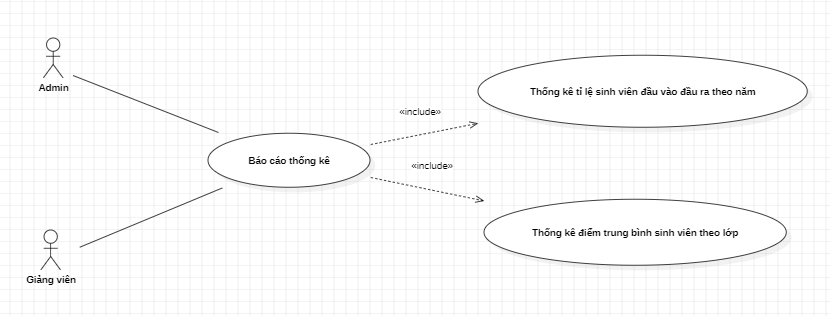
- Mô tả: Chức năng quản lý điểm là chức năng chính của hệ thống quản lý điểm sinh viên, bất kỳ người dùng nào đăng nhập vào chương trình cũng có thể sử dụng chức năng này, tuy nhiên sẽ có sự khác biệt về tính năng sử dụng phụ thuộc vào phân quyền của tài khoản đăng nhập.

+ Đối với quyền quản trị viên, người dùng có thể sử dụng tất cả các tính năng của chức năng quản lý điểm. Để nhập điểm (có thể nhập danh sách điểm từ excel) hoặc sửa thông tin điểm số, người dùng cần chọn các chức năng tương ứng trên giao diện chương trình, sau khi nhập đầy đủ thông tin yêu cầu và bấm lưu, chương trình sẽ thực hiện kiểm tra tính hợp lệ của dữ liệu để đảm bảo rằng các thông tin luôn đầy đủ và chính xác. Nếu dữ liệu đã hợp lệ, hệ thống sẽ tự động tính toán điểm trung bình dựa trên các dữ liệu điểm thành phần được người dùng nhập vào và tự động đánh giá xếp loại điểm cho sinh viên sau đó hiển thị thông tin lên màn hình danh sách. Người dùng có thể dễ dàng tra cứu thông tin điểm số một cách nhanh chóng và tiện lợi dựa vào tính năng lọc và tìm kiếm trên màn hình danh sách. Nếu cần xuất thông tin điểm ra file excel, chỉ cần bấm vào chức năng này trên giao diện, một hộp thoại sẽ hiển thị lên để người dùng chọn nơi lưu trữ dữ liệu khi xuất ra excel và bấm đồng ý để thực hiện. Cuối cùng, khi muốn xóa các bản ghi, người dùng có thể chọn xóa ở cột chức năng trên từng dòng dữ liệu ở danh sách hoặc tích chọn tối thiểu 2 bản ghi (tích vào các checkbox ở đầu các dòng dữ liệu) để sử dụng tính năng thực hiện xóa hàng loạt bản ghi.

+ Đối với quyền giảng viên, sự khác biệt duy nhất so với quyền quản trị viên đó là khi ở màn hình danh sách, chương trình sẽ chỉ cho giảng viên xem thông tin điểm số các môn mà giảng viên đó giảng dạy, còn lại các tính năng khác giảng viên thực hiện tương tự như khi đăng nhập với quyền quản trị viên đã được mô tả ở trên.

+ Còn đối với quyền sinh viên, người dùng chỉ có thể xem danh sách điểm số của mình trên danh sách, chương trình sẽ hộ trợ sinh viên tìm kiếm thông tin điểm số và xuất danh sách điểm ra file excel.

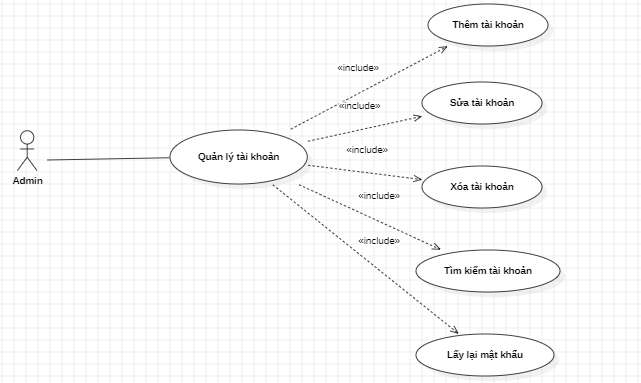
**- Báo cáo thống kê:**

****

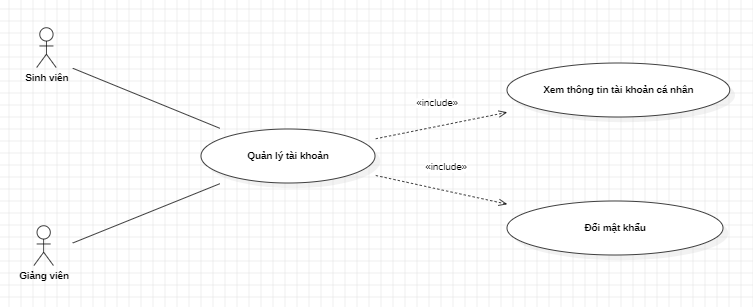
**Hình 3.7** Sơ đồ use case báo cáo thống kê

- Mô tả: Để xem được các báo cáo thống kê về hệ thống quản lý điểm sinh viên, người dùng cần đăng nhập vào chương trình với tài khoản có quyền quản trị viên hoặc giảng viên sau đó chọn vào biểu đồ mà mình muốn xem ở trên giao diện. Hiện tại chương trình đã có các biểu đồ rất hữu ích như: thống kê tỉ lệ sinh viên đầu vào/đầu ra trong vòng 5 nam trở lại đây, thống kê điểm trung bình của sinh viên theo lớp học phần.

**- Quản lý tài khoản:**

****

**Hình 3.8** Sơ đồ use case quản lý tài khoản với quyền quản trị viên



**Hình 3.9** Sơ đồ use case quản lý tài khoản với quyền giảng viên, sinh viên

- Mô tả: Với quyền quản trị viên, người dùng có thể quản lý tất cả các tài khoản đăng nhập của người dùng, bao gồm thêm, xóa, sửa và tìm kiếm tài khoản tương tự như các chức năng quản lý khác. Ngoài ra, ở phần quản lý tài khoản này còn có tính năng lấy lại mật khẩu, khi bấm thực hiện chức năng này ở cột chức năng trên mỗi dòng danh sách, chương trình sẽ thực hiện cập nhật lại mật khẩu cho tài khoản này là 1. Sau đó, người dùng có thể đăng nhập vào để đổi lại mật khẩu của mình bằng mật khẩu khác an toàn hơn.

3.4. Sơ đồ Activity:

3.4.1. Đăng nhập:

3.5. Mô hình thực thể liên kết:

3.6. Thiết kế cơ sở dữ liệu:

3.6.1. Tên các bảng và mục đích sử dụng:

**Bảng 3.1** Thông tin chung các bảng trong database

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Mục đích sử dụng** |
| 1 | role | Lưu thông tin các quyền của chương trình |
| 2 | user | Lưu thông tin tài khoản đăng nhập |
| 3 | faculty | Lưu thông tin khoa |
| 4 | classes | Lưu thông tin lớp học |
| 5 | student | Lưu thông tin sinh viên |
| 6 | teacher | Lưu thông tin giảng viên |
| 7 | semester | Lưu thông tin học kỳ |
| 8 | subject | Lưu thông tin môn học |
| 9 | class\_registration | Lưu thông tin lớp học phần |
| 10 | class\_registration\_detail | Lưu thông tin chi tiết lớp học phần |
| 11 | score | Lưu thông tin điểm số của sinh viên |

3.6.2. Thông tin chi tiết các bảng:

**Bảng 3.2** Bảng thông tin quyền người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | role\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | role\_code | int4 | Mã quyền |
| 3 | role\_name | text | Tên quyền |
| 4 | created\_by | text | Người tạo bản ghi |
| 5 | created\_date | date | Ngày tạo bản ghi |
| 6 | modified\_by | text | Người sửa bản ghi |
| 7 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.3** Bảng thông tin tài khoản đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | user\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | user\_name | text | Tên đăng nhập |
| 3 | pass\_word | text | Mật khẩu |
| 4 | role\_id | uuid | Khóa ngoại, trỏ tới bảng role |
| 5 | status | int4 | Trạng thái sử dụng |
| 6 | created\_by | text | Người tạo bản ghi |
| 7 | created\_date | date | Ngày tạo bản ghi |
| 8 | modified\_by | text | Người sửa bản ghi |
| 9 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.4** Bảng thông tin khoa

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | faculty\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | faculty\_code | text | Mã khoa |
| 3 | faculty\_name | text | Tên khoa |
| 4 | created\_by | text | Người tạo bản ghi |
| 5 | created\_date | date | Ngày tạo bản ghi |
| 6 | modified\_by | text | Người sửa bản ghi |
| 7 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.5** Bảng thông tin lớp học

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | classes\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | classes\_code | text | Mã lớp học |
| 3 | classes\_name | text | Tên lớp học |
| 4 | faculty\_id | uuid | Khóa ngoại, trỏ tới bảng faculty |
| 5 | created\_by | text | Người tạo bản ghi |
| 6 | created\_date | date | Ngày tạo bản ghi |
| 7 | modified\_by | text | Người sửa bản ghi |
| 8 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.6** Bảng thông tin sinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | student\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | student\_code | text | Mã sinh viên |
| 3 | student\_name | text | Tên sinh viên |
| 4 | birthday | date | Ngày sinh |
| 5 | gender | text | Giới tính |
| 6 | address | text | Địa chỉ |
| 7 | phone\_number | text | Số điện thoại |
| 8 | email | text | Địa chỉ email |
| 9 | image | text | Ảnh |
| 10 | admission\_year | date | Năm vào trường |
| 11 | graduation\_year | date | Năm ra trường |
| 12 | classes\_id | uuid | Khóa ngoại, trỏ tới bảng classes |
| 13 | created\_by | text | Người tạo bản ghi |
| 14 | created\_date | date | Ngày tạo bản ghi |
| 15 | modified\_by | text | Người sửa bản ghi |
| 16 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.7** Bảng thông tin giảng viên

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | teacher\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | teacher\_code | text | Mã giảng viên |
| 3 | teacher\_name | text | Tên giảng viên |
| 4 | birthday | date | Ngày sinh |
| 5 | gender | text | Giới tính |
| 6 | address | text | Địa chỉ |
| 7 | phone\_number | text | Số điện thoại |
| 8 | email | text | Địa chỉ email |
| 9 | image | text | Ảnh |
| 10 | faculty\_id | uuid | Khóa ngoại, trỏ tới bảng faculty |
| 11 | created\_by | text | Người tạo bản ghi |
| 12 | created\_date | date | Ngày tạo bản ghi |
| 13 | modified\_by | text | Người sửa bản ghi |
| 14 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.8** Bảng thông tin học kỳ

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | semester\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | semester\_code | text | Mã học kỳ |
| 3 | semester\_name | text | Tên học kỳ |
| 4 | created\_by | text | Người tạo bản ghi |
| 5 | created\_date | date | Ngày tạo bản ghi |
| 6 | modified\_by | text | Người sửa bản ghi |
| 7 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.9** Bảng thông tin môn học

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | subject\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | subject\_code | text | Mã môn học |
| 3 | subject\_name | text | Tên môn học |
| 4 | number\_tc | int4 | Số tín chỉ |
| 5 | semester\_id | uuid | Khóa ngoại, trở tới bảng semester |
| 6 | created\_by | text | Người tạo bản ghi |
| 7 | created\_date | date | Ngày tạo bản ghi |
| 8 | modified\_by | text | Người sửa bản ghi |
| 9 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.10** Bảng thông tin lớp học phần

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | class\_registration\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | class\_registration\_code | text | Mã lớp học phần |
| 3 | class\_registration\_name | text | Tên lớp học phần |
| 4 | subject\_id | uuid | Khóa ngoại, trỏ tới bảng subject |
| 5 | teacher\_id | uuid | Khóa ngoại, trở tới bảng teacher |
| 6 | created\_by | text | Người tạo bản ghi |
| 7 | created\_date | date | Ngày tạo bản ghi |
| 8 | modified\_by | text | Người sửa bản ghi |
| 9 | modified\_date | date | Ngày sửa bản ghi |

**Bảng 3.11** Bảng thông tin chi tiết lớp học phần

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | class\_registration\_detail\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | class\_registration\_id | uuid | Khóa ngoại, trở tới bảng class\_registration |
| 3 | student\_id | uuid | id của sinh viên tham gia lớp học phần |
| 4 | student\_code | text | Mã sinh viên tham gia lớp học phần |
| 5 | student\_name | text | Tên sinh viên tham gia lớp học phần |

**Bảng 3.12** Bảng thông tin điểm sinh viên

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên cột** | **Kiểu dữ liệu** | **Mô tả** |
| 1 | score\_id | uuid | Khóa chính, xác định bản ghi |
| 2 | score\_attendance | float8 | Điểm chuyên cần |
| 3 | score\_test | float8 | Điểm kiểm tra |
| 4 | score\_exam | float8 | Điểm thi |
| 5 | score\_average | float8 | Điểm trung bình |
| 6 | student\_id | uuid | id của sinh viên |
| 7 | teacher\_id | uuid | id của giảng viên |
| 8 | class\_registration\_id | uuid | Khóa ngoại, trỏ tới bảng class\_registration |
| 9 | created\_by | text | Người tạo bản ghi |
| 10 | created\_date | date | Ngày tạo bản ghi |
| 11 | modified\_by | text | Người sửa bản ghi |
| 12 | modified\_date | date | Ngày sửa bản ghi |