BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

A blue diamond with a torch and a yellow circle with a red flame

Description automatically generated

**Niên luận**

**Ngành mạng máy tính và truyền thông dữ liệu**

**Đề tài**

**HỆ THỐNG HỖ TRỢ QUẢN LÝ TIẾN ĐỘ HỌC TẬP CỦA SINH VIÊN**

**Sinh viên: Cao Nguyễn Tường Vi**

**Mã số : B2110955**

**Khóa: 47**

Cần Thơ, 04/2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU**



**Niên luận**

**Ngành mạng máy tính và truyền thông dữ liệu**

**Đề tài**

**HỆ THỐNG HỖ TRỢ QUẢN LÝ TIẾN ĐỘ HỌC TẬP CỦA SINH VIÊN**

|  |  |
| --- | --- |
| **NGƯỜI HƯỚNG DẪN**  **TS. Lâm Chí Nguyện** | **SINH VIÊN THỰC HIỆN**  **Cao Nguyễn Tường Vi**  **Mã số: B2110955**  **Khóa: 47** |

Cần Thơ, 04/2025

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------**

|  |  |
| --- | --- |
|  | Cần Thơ, ngày … tháng 04 năm 2025  **Giảng viên hướng dẫn**  **TS. Lâm Chí Nguyện** |

# **Mục lục**

**[Chương 1 Tổng quan đề tài 5](#_Toc27623)**

**[1.1. Đặt vấn đề 5](#_Toc552)**

[1.1.1 Bối cảnh xuất hiện vấn đề 5](#_Toc31399)

[1.1.2 Nội dung và yêu cầu bài toán 5](#_Toc15064)

**[1.2. Ý nghĩa và mục tiêu của hệ thống 6](#_Toc10981)**

[1.2. Lịch sử giải quyết vấn đề: 6](#_Toc32716)

[1.3. Phạm vi của đề tài: 8](#_Toc11554)

[1.4. Phương pháp nghiên cứu/ hướng giải quyết vấn đề: 9](#_Toc3957)

[1.4.2. Các yêu cầu quản lý tiến độ học tập 10](#_Toc7722)

**[Chương 2 Cơ sở lý thuyết: 15](#_Toc13038)**

**[2.1. Cơ sở lý thuyết 15](#_Toc19503)**

[2.1.1. Khái niệm về hệ thống thông tin quản lý. 15](#_Toc27441)

[2.1.2. Tiến độ học tập của sinh viên. 16](#_Toc14726)

[2.1.3. Cấu trúc dữ liệu trong hệ thống. 17](#_Toc18791)

[2.1.4. Tính toán GPA và quản lý học phần. 19](#_Toc29267)

[2.1.5. Công nghệ và giao diện người dùng. 19](#_Toc21076)

[2.1.6. Vai trò và ý nghĩa với sinh viên. 21](#_Toc13492)

**[2.2. Cơ sở kỹ thuật 22](#_Toc3679)**

**[2.3. Ngôn ngữ lập trình 25](#_Toc10024)**

**[2.4. Đề xuất công thức, Lý thuyết của tác giả 26](#_Toc13020)**

**[Chương 3 Nội dung và kết quả nghiên cứu: 30](#_Toc30533)**

**[3.1. Mô hình phân rã chức năng 30](#_Toc30318)**

**[3.2. Mô hình hoạt động nghiệp vụ 31](#_Toc19189)**

[1. Đăng ký / Đăng nhập 31](#_Toc29125)

[2. Cập nhật thông tin cá nhân 32](#_Toc218)

[3. Quản lý học phần (Course) 33](#_Toc14082)

[4. Quản lý điểm (Score) 34](#_Toc24706)

[5. Quản lý học kỳ (Semester) 34](#_Toc23943)

[6. Theo dõi tiến độ học tập (Progress) 35](#_Toc8598)

**[3.3. Mô hình thực thể quan hệ 35](#_Toc27905)**

[1. Thực thể: 35](#_Toc10539)

[2. Mối quan hệ giữa các thực thể: 35](#_Toc10721)

**[3.4. Mô hình tổ chức hệ thống 36](#_Toc5843)**

**[Chương 4 Kết luận và đề nghị 38](#_Toc5807)**

**[4.1. Kết luận: 38](#_Toc18226)**

**[4.2. Một số giao diện của hệ thống: 38](#_Toc22741)**

**[4.3. Đề nghị và hướng phát triển: 39](#_Toc24596)**

**[Phụ lục 41](#_Toc25809)**

**[1. Phụ lục hình ảnh: 41](#_Toc23130)**

**[2. Phụ lục bảng: 41](#_Toc13556)**

**[Tài liệu tham khảo. 42](#_Toc24635)**

# **Tổng quan đề tài**

## **Đặt vấn đề**

### 1.1.1 Bối cảnh xuất hiện vấn đề

Trong bối cảnh giáo dục đại học ngày càng phát triển mạnh mẽ, số lượng sinh viên tại các trường đại học, cao đẳng không ngừng gia tăng. Đồng thời, sự đa dạng trong chương trình đào tạo, các quy định học vụ và yêu cầu học tập đặt ra nhiều thách thức cho công tác quản lý, giám sát và hỗ trợ sinh viên trong quá trình học tập.

Thông thường, việc theo dõi tiến độ học tập của sinh viên vẫn còn phụ thuộc nhiều vào các phương pháp truyền thống như bảng điểm giấy, sổ tay sinh viên, sổ liên lạc nội bộ, hay hệ thống quản lý học vụ đơn giản. Những cách làm này tuy vẫn có giá trị trong một số tình huống, nhưng dần trở nên không hiệu quả và thiếu tính toàn diện khi áp dụng trong môi trường giáo dục hiện đại với quy mô lớn, yêu cầu cao về tự động hóa và cá nhân hóa.

Giáo dục hiện nay không chỉ dừng lại ở việc truyền đạt kiến thức, mà còn đòi hỏi sự tương tác cao giữa sinh viên và nhà trường. Việc nắm bắt và theo dõi sát sao tiến độ học tập của từng sinh viên đóng vai trò quan trọng trong việc định hướng, cố vấn học tập, phát hiện sớm các vấn đề trong quá trình học, từ đó có biện pháp hỗ trợ kịp thời và hiệu quả.

Trước nhu cầu đó, việc xây dựng một **hệ thống quản lý tiến độ học tập hiện đại, tích hợp và linh hoạt** là hoàn toàn cần thiết. Đây không chỉ là một xu hướng tất yếu trong chuyển đổi số giáo dục, mà còn là giải pháp then chốt nhằm nâng cao chất lượng đào tạo, đảm bảo tính minh bạch, công bằng và hiệu quả trong công tác quản lý sinh viên.

### 1.1.2 Nội dung và yêu cầu bài toán

Trên cơ sở những khó khăn và bất cập nêu trên, bài toán đặt ra là cần thiết kế và xây dựng một hệ thống có khả năng:

1. **Theo dõi tiến độ học tập cá nhân hóa cho từng sinh viên:**

* Quản lý thông tin sinh viên: Bao gồm thông tin cá nhân, lớp học, chuyên ngành và chương trình đào tạo.
* Theo dõi điểm số: Lưu trữ kết quả học tập theo từng học kỳ, từng học phần; hiển thị điểm trung bình học kỳ, điểm tích lũy, học lực.
* Giám sát quá trình học tập: Xác định số lượng học phần đã hoàn thành, học phần còn nợ, điều kiện để xét tốt nghiệp hoặc học lại.
* Lên lịch học: Hỗ trợ sinh viên lựa chọn lịch học theo học kỳ, sắp xếp thời gian phù hợp với tiến độ và kế hoạch học tập cá nhân.

1. **Tối ưu trải nghiệm người dùng:**

* Giao diện thân thiện, trực quan, dễ sử dụng cho cả sinh viên và cán bộ quản lý.
* Hỗ trợ truy cập đa nền tảng: Ứng dụng web tương thích trên máy tính, điện thoại thông minh.
* Tích hợp các tính năng hỗ trợ như tìm kiếm, lọc học kỳ, thông báo, thống kê.

1. **Hỗ trợ quản lý và cố vấn học tập:**

* Giúp giảng viên dễ dàng theo dõi tiến độ của sinh viên từng lớp, từng môn học.
* Hỗ trợ cố vấn học tập đưa ra lời khuyên, điều chỉnh kế hoạch học tập phù hợp.
* Cung cấp báo cáo thống kê giúp nhà trường đánh giá chất lượng đào tạo, hiệu quả học tập theo từng khóa, từng ngành.

## **1.2. Ý nghĩa và mục tiêu của hệ thống**

### 1.2.1 Đối với sinh viên:

* Chủ động trong việc theo dõi tiến độ học tập, nắm bắt chính xác điểm số, số tín chỉ đã hoàn thành và cần hoàn thành.
* Lập kế hoạch học tập cá nhân một cách khoa học, phù hợp với khả năng, lịch trình và mục tiêu học tập.
* Nhận cảnh báo sớm khi có nguy cơ học lại, học bù, hoặc chưa đủ điều kiện tốt nghiệp.
* Tăng cường khả năng tự học, tự quản lý bản thân và rèn luyện kỹ năng lên kế hoạch – một kỹ năng quan trọng trong thời đại số.

### 1.2.2 Tóm lại:

Hệ thống quản lý tiến độ học tập sinh viên không chỉ đơn thuần là một công cụ công nghệ phục vụ quản lý học vụ, mà còn là một “**trợ lý học tập số**” đồng hành cùng sinh viên trong suốt quá trình học đại học. Việc phát triển hệ thống này mang lại giá trị lâu dài trong nâng cao chất lượng giáo dục, cá nhân hóa trải nghiệm học tập và tối ưu hóa công tác quản lý đào tạo tại các cơ sở giáo dục đại học.

## **Lịch sử giải quyết vấn đề:**

### 1.2.1. Lịch sử giải quyết vấn đề:

Vấn đề quản lý tiến độ học tập của sinh viên đã được nghiên cứu và giải quyết ở nhiều quốc gia và cơ sở giáo dục trên thế giới, đặt biệt với sự phát triển của công nghệ thông tin và các hệ thống quản lý học tập (LMS - Learning Management System). Sự phát triển của hệ thống quản lý học tập (LMS) là một hành trình dài gắn liền với sự thay đổi của công nghệ và nhu cầu học tập ngày càng cao của cá nhân cũng như tổ chức. Từ những bước đi đầu tiên còn đơn giản và hạn chế, LMS đã trải qua những giai đoạn chuyển mình mạnh mẽ để trở thành một công cụ trung tâm trong việc quản lý, tổ chức và triển khai đào tạo trực tuyến [[1](https://oes.vn/su-phat-trien-cua-he-thong-quan-ly-hoc-tap/)]:

1. **Giai đoạn khởi đầu:**

Vào những năm cuối thế kỷ 20, khái niệm hệ thống quản lý học tập bắt đầu hình thành từ những công cụ giao tiếp trực tuyến sơ khai như Bulletin Board Systems (BBS). Cùng thời gian này, các hệ thống quản lý nội dung (CMS) cũng được tận dụng để phục vụ mục đích lưu trữ và quản lý nội dung học tập, từ đó tạo tiền đề cho LMS sơ khai với những chức năng cơ bản như đăng tải bài giảng, quản lý danh sách người dùng và theo dõi tiến độ học tập một cách đơn giản.

Cuối những năm 1990, các hệ thống LMS chuyên dụng đầu tiên bắt đầu xuất hiện, đánh dấu một bước ngoặt quan trọng trong việc quản lý học tập trực tuyến.

1. **Giai đoạn phát triển mạnh mẽ:**

Bước vào đầu thế kỷ 21, sự phát triển của công nghệ Internet băng rộng và đặc biệt là sự xuất hiện của công nghệ web 2.0 đã thúc đẩy LMS tiến thêm một bước dài.

Nhiều nền tảng LMS nổi bật như Moodle, Blackboard và Canvas ra đời và nhanh chóng trở thành tiêu chuẩn trong lĩnh vực giáo dục và đào tạo.

Đây là thời kỳ LMS khẳng định vai trò quan trọng trong việc xây dựng môi trường học tập linh hoạt, phù hợp với quy mô lớn và đa dạng đối tượng học viên.

1. **Giai đoạn hiện đại:**

Trong kỷ nguyên số ngày nay, hệ thống LMS đã phát triển thành các nền tảng học tập toàn diện, được tích hợp với những công nghệ tiên tiến như trí tuệ nhân tạo (AI), phân tích dữ liệu lớn (big data) và các công cụ tự động hóa.

Các hệ thống quản lý học tập đã được phát triển thêm với các tính năng mạnh mẽ, hệ thống nhắc nhở tự động, và tích hợp với các công cụ học tập trực tuyến như Google Classroom, Canvas,...

### 1.2.2. Kết quả và tồn tại:

Với sự tiện ích và phát triển mạnh mẽ, hệ thống ngày càng được triển khai rộng rãi, chứng minh được hiệu quả trong việc quản lý và theo dõi quá trình học tập của sinh viên, cung cấp cho sinh viên công cụ hỗ trợ học tập trực tuyến. Mặt khác, hệ thống quản lý hiện đại đã giúp giảng viên có cái nhìn tổng quan về tình trạng học tập của sinh viên từ đó đưa ra giải pháp thích hợp cho công tác giảng dạy.

Tuy nhiên, đi đôi với sự phát triển và tin dùng hiện đại, vẫn còn tồn tại một số vấn đề trong quá trình triển khai:

* **Chi phí và khả năng triển khai**: Đối với một số trường đại học nhỏ hoặc ở quốc gia đang phát triển, chi phí triển khai và duy trì các hệ thống LMS hiện đại vẫn là một vấn đề lớn.
* **Tốc độ truy cập vào hệ thống:** Một số hệ thống vẫn chưa đáp ứng được tốc độ phản hồi khi có nhiều sinh viên truy cập cùng một thời điểm, gây tình trạng “đóng băng” hệ thống.
* **Giao diện thiếu thu hút:** Giao diện hệ thống chưa thật sự bắt mắt và chưa tạo được hứng thú sử dụng thúc đẩy sinh viên sử dụng với tần suất thường xuyên.

## **Phạm vi của đề tài:**

### 1.3.1. Phạm vi đề tài:

* **Quản lý thông tin sinh viên:** Đây là phần cơ bản, liên quan đến việc quản lý thông tin cá nhân, bao gồm như tên, thông tin tài khoản, …
* **Quản lý môn học:**
* **Quản lý học phần:** Cung cấp danh sách học phần bao gồm: thông tin môn học, bao gồm tên môn học, mã môn học, thời gian học, và kết quả học tập.
* **Quản lý năm học:**  Cho phép tạo thêm học kỳ mới với các học phần đã được thêm từ danh sách học phần.
* **Quản lý tiến độ học tập:** Hệ thống sẽ chủ yếu tập trung vào theo dõi quá trình học tập của sinh viên, bao gồm những môn học sinh viên đã hoàn thành, số tính chỉ tích lũy, kết quả học tập. Cung cấp các thông báo về tiến độ học tập, thay đổi về lịch học, các môn học điểm thấp.

### 1.3.2. Mức độ giải quyết đề tài:

Trong khuôn khổ đề tài này, hệ thống được xây dựng ở **mức độ trung bình**, đảm bảo đáp ứng các chức năng cốt lõi trong việc quản lý và hỗ trợ sinh viên theo dõi tiến độ học tập, đồng thời bổ sung các tiện ích nâng cao giúp tối ưu hóa trải nghiệm người dùng và hiệu quả quản lý.

### **Quản lý thông tin học tập cơ bản**

* Hệ thống cho phép lưu trữ và xử lý thông tin sinh viên như tên người dùng, mật khẩu, số điện thoại,...
* Thống kê các học phần đã hoàn thành, điểm số theo từng học kỳ, và số tín chỉ tích lũy.
* Hiển thị tiến độ học tập một cách trực quan để sinh viên dễ dàng nắm bắt vị trí hiện tại trong chương trình đào tạo.

### **Cảnh báo và nhắc nhở học tập**

* Hệ thống tự động phát hiện và đưa ra cảnh báo khi sinh viên có học phần chưa hoàn thành, hoặc không đạt yêu cầu tích lũy.
* Thông báo lịch học theo từng học kỳ.

### **Báo cáo tiến độ học tập**

* Hệ thống cung cấp thông báo về kết quả học tập và tiến độ học phần.
* Các báo cáo được thể hiện rõ ràng giúp người học dễ dàng theo dõi và đánh giá.

### **Tối ưu trải nghiệm người dùng**

* Giao diện được thiết kế thân thiện, dễ thao tác, phù hợp với cả sinh viên và giảng viên.
* Trải nghiệm người dùng được đặt lên hàng đầu nhằm đảm bảo hiệu quả sử dụng cao, giảm thiểu thao tác dư thừa.

## **Phương pháp nghiên cứu/ hướng giải quyết vấn đề:**

### Xác định khái niệm cơ bản:

Tiến độ học tập của sinh viên là quá trình theo dõi và đánh giá sự phát triển của sinh viên trong suốt thời gian tham gia chương trình đào tạo.

Tiến độ này được phản ánh thông qua nhiều chỉ số, bao gồm:

* Số lượng môn học đã đăng ký và hoàn thành.
* Tổng số tín chỉ đã tích lũy so với yêu cầu tốt nghiệp.
* Kết quả học tập của từng học phần, bao gồm điểm số theo thang điểm 10, điểm chữ (A, B, C, D, F) và điểm hệ 4.
* Thành tích tại các kỳ thi, bài kiểm tra định kỳ và bài tập lớn.
* Tham gia các hoạt động học tập bổ trợ, như seminar, workshop, thực tập.

Việc nắm rõ tiến độ học tập giúp sinh viên:

* Đánh giá chính xác vị trí hiện tại trong lộ trình học tập.
* Lập kế hoạch học tập hiệu quả hơn cho các học kỳ tiếp theo.
* Đảm bảo hoàn thành chương trình học đúng hạn hoặc sớm hơn dự kiến.
* Quản lý tiến độ học tập là quá trình theo dõi, thu thập, lưu trữ và phân tích dữ liệu liên quan đến sự phát triển học tập của sinh viên nhằm hỗ trợ việc ra quyết định học tập hiệu quả.

Nhiệm vụ chính của quản lý tiến độ học tập bao gồm:

* Ghi nhận tất cả thông tin học tập như: điểm số, kết quả học phần, số tín chỉ đạt được, số học phần chưa hoàn thành hoặc bị nợ.
* Theo dõi sự thay đổi kết quả học tập qua các học kỳ để phát hiện sớm những vấn đề bất thường (suy giảm thành tích, không đạt tiến độ...).
* Đánh giá năng lực học tập tổng thể thông qua chỉ số như GPA học kỳ, GPA tích lũy, số lượng môn nợ, mức xếp loại học lực.
* Cảnh báo và đề xuất giải pháp nếu sinh viên có nguy cơ không đạt yêu cầu đầu ra, ví dụ như thiếu tín chỉ tốt nghiệp hoặc GPA dưới mức tối thiểu.

Hệ thống quản lý tiến độ học tập phải đảm bảo:

* Dễ sử dụng: Sinh viên và giảng viên có thể truy cập thông tin một cách nhanh chóng và rõ ràng.
* Chính xác: Dữ liệu luôn được cập nhật và tính toán đúng quy định.
* Minh bạch: Thông tin phản ánh trung thực quá trình học tập của từng cá nhân.
* Bảo mật: Bảo vệ dữ liệu cá nhân của sinh viên khỏi các truy cập trái phép.

Quản lý tiến độ học tập là quá trình theo dõi, thu thập, lưu trữ và phân tích dữ liệu liên quan đến sự phát triển học tập của sinh viên nhằm hỗ trợ việc ra quyết định học tập hiệu quả.

Nhiệm vụ chính của quản lý tiến độ học tập bao gồm:

* Ghi nhận tất cả thông tin học tập như: điểm số, kết quả học phần, số tín chỉ đạt được, số học phần chưa hoàn thành hoặc bị nợ.
* Theo dõi sự thay đổi kết quả học tập qua các học kỳ để phát hiện sớm những vấn đề bất thường (suy giảm thành tích, không đạt tiến độ...).
* Đánh giá năng lực học tập tổng thể thông qua chỉ số như GPA học kỳ, GPA tích lũy, số lượng môn nợ, mức xếp loại học lực.
* Cảnh báo và đề xuất giải pháp nếu sinh viên có nguy cơ không đạt yêu cầu đầu ra, ví dụ như thiếu tín chỉ tốt nghiệp hoặc GPA dưới mức tối thiểu.

Hệ thống quản lý tiến độ học tập phải đảm bảo:

* Dễ sử dụng: Sinh viên và giảng viên có thể truy cập thông tin một cách nhanh chóng và rõ ràng.
* Chính xác: Dữ liệu luôn được cập nhật và tính toán đúng quy định.
* Minh bạch: Thông tin phản ánh trung thực quá trình học tập của từng cá nhân.
* Bảo mật: Bảo vệ dữ liệu cá nhân của sinh viên khỏi các truy cập trái phép.

## Các yêu cầu quản lý tiến độ học tập

**1. Yêu cầu về chức năng**

* Quản lý sinh viên:
* Lưu trữ thông tin cá nhân: họ tên, email, số điện thoại, tài khoản đăng nhập.
* Cho phép cập nhật thông tin cá nhân khi cần thiết.
* Quản lý học phần:
* Lưu trữ danh sách học phần (mã học phần, tên học phần, số tín chỉ).
* Cho phép tìm kiếm, lọc học phần theo mã hoặc tên.
* Quản lý điểm số:
* Ghi nhận điểm số cho từng học phần (bao gồm: điểm số, điểm chữ, điểm hệ 4).
* Cho phép cập nhật hoặc chỉnh sửa điểm số nếu cần thiết.
* Quản lý học kỳ:
* Ghi nhận năm học, học kỳ và danh sách các học phần trong học kỳ đó.
* Cho phép thêm mới, cập nhật thông tin học kỳ.
* Tính toán tự động:
* Tính GPA học kỳ và GPA tích lũy dựa trên điểm số và số tín chỉ.
* Tính tổng số tín chỉ tích lũy và cảnh báo nếu sinh viên nợ môn hoặc vượt ngưỡng tín chỉ.
* Theo dõi tiến độ:
* Hiển thị trực quan lộ trình học tập: số tín chỉ đã tích lũy, số môn học còn thiếu, tổng số tín chỉ yêu cầu để tốt nghiệp.
* Cảnh báo khi sinh viên chưa đạt đủ yêu cầu (ví dụ: thiếu tín chỉ, GPA quá thấp, môn học chưa đạt).

**2. Yêu cầu về phi chức năng**

* Hiệu suất:

Tốc độ truy xuất dữ liệu nhanh, đặc biệt khi thao tác với danh sách học phần và điểm số nhiều.

* Bảo mật:
* Đảm bảo mật khẩu người dùng được mã hóa khi lưu trữ.
* Kiểm soát quyền truy cập, chỉ cho phép sinh viên xem và chỉnh sửa dữ liệu của chính mình.
* Tính mở rộng:

Hệ thống dễ dàng mở rộng khi có thêm yêu cầu mới như thêm tiêu chí xếp loại, thay đổi cơ cấu tính GPA,...

* Tính khả dụng:

Hệ thống hoạt động ổn định, hạn chế tối đa lỗi xảy ra trong quá trình nhập liệu và tính toán.

* Giao diện thân thiện:

Dễ sử dụng, giao diện rõ ràng, có hướng dẫn thao tác, phù hợp với cả người dùng không am hiểu kỹ thuật.

### Phân tích các yếu tố ảnh hưởng đến tiến độ học tập:

Trong quá trình học tập, tiến độ của sinh viên chịu tác động bởi nhiều yếu tố khác nhau. Việc phân tích các yếu tố này giúp hệ thống quản lý tiến độ học tập đề xuất các giải pháp hỗ trợ sinh viên hiệu quả hơn. Các yếu tố chủ yếu bao gồm:

**1. Động lực học tập**

* **Khái niệm**: Động lực học tập là nguồn năng lượng nội tại thúc đẩy sinh viên nỗ lực đạt được mục tiêu học tập của mình.
* Tác động: Sinh viên có động lực cao thường hoàn thành tốt các học phần, đạt điểm số cao, tiến độ học tập ổn định hoặc nhanh hơn so với kế hoạch.
* Giải pháp hệ thống:
* Cung cấp thống kê trực quan về tiến độ hiện tại (ví dụ: số tín chỉ đã đạt, số môn còn thiếu).
* Hiển thị thành tích, khen thưởng khi đạt mốc điểm cao hoặc hoàn thành học kỳ tốt.
* Gửi thông báo động viên, nhắc nhở học tập định kỳ.

1. **Quản lý thời gian**

* Khái niệm: Quản lý thời gian hiệu quả là khả năng lên kế hoạch và phân bổ hợp lý thời gian cho việc học, sinh hoạt và giải trí.
* Tác động: Sinh viên thiếu kỹ năng quản lý thời gian thường bị trễ hạn bài tập, thi lại nhiều môn, dẫn đến kéo dài thời gian học tập.
* Giải pháp hệ thống:
* Tích hợp lịch nhắc nhở cho các kỳ thi, hạn nộp bài tập.
* Đề xuất kế hoạch học tập ngắn hạn (theo tuần, tháng) dựa trên danh sách môn học.
* Cảnh báo học phần tồn đọng hoặc môn học sắp đến kỳ thi.

**3. Khả năng tự học**

* Khái niệm: Khả năng tự học phản ánh sự chủ động, sáng tạo của sinh viên trong việc tiếp cận và lĩnh hội kiến thức.
* Tác động: Sinh viên tự học tốt thường tiến bộ nhanh, chủ động giải quyết các môn khó, giảm thiểu tình trạng học lại.
* Giải pháp hệ thống:
* Cung cấp tài liệu tham khảo, bài tập tự luyện liên quan đến các môn học.
* Gợi ý khóa học bổ trợ nếu hệ thống phát hiện điểm số thấp ở một số học phần.

**4. Mức độ phù hợp của chương trình đào tạo**

* Tác động: Chương trình quá tải hoặc thiếu linh hoạt sẽ gây áp lực lớn lên tiến độ học tập.
* Giải pháp hệ thống:
* Hỗ trợ sinh viên lên kế hoạch cá nhân hóa, chọn môn học phù hợp theo khả năng từng học kỳ.
* Theo dõi và báo cáo mức độ hoàn thành từng giai đoạn chương trình.

**5. Sức khỏe thể chất và tinh thần**

* Tác động: Vấn đề sức khỏe ảnh hưởng trực tiếp đến khả năng học tập và duy trì tiến độ.
* Giải pháp hệ thống:
* Gửi nhắc nhở về việc cân bằng học tập và nghỉ ngơi.
* Cho phép sinh viên tạm hoãn học phần nếu có lý do chính đáng và ghi nhận trong hệ thống.

### 1.4.4. Xây dựng mô hình lý thuyết cho hệ thống:

Dựa trên các kiến thức và lý thuyết thu thập được, xây dựng mô hình lý thuyết cho hệ thống. Mô hình này gồm các yếu tố cơ bản:

* **Cấu trúc cơ sở dữ liệu:** xây dựng cơ sở dữ liệu để lưu trữ thông tin sinh viên, môn học, điểm số, tiến độ học tập.
* **Chức năng theo dõi tiến độ học tập:** Tạo các công cụ cho sinh viên giảng viên theo dõi chế độ học tập, như các tính năng như bảng điểm, thông báo môn học không đạt yêu cầu,...
* **Hệ thống báo cáo và thông báo:** thiết kế hệ thống báo cáo và thông báo để giúp sinh viên nhận được các cảnh báo và nhắc nhở kịp thời về tiến độ học tập.

### 1.4.5. Khảo sát các hệ thống quản lý học tập hiện có:

**Moodle, Blackboard** và **Canvas** là ba hệ thống quản lý học tập *(LMS - Learning Management System)* phổ biến, được sử dụng rộng rãi ở các trường học và đại học. Mỗi hệ thống cung cấp các tính năng và công cụ hỗ trợ giảng dạy, học tập và quản lý học vụ:

1. **Moodle:**

Là một hệ thống mã nguồn mở miễn phí, một trong những hệ thống LMS phổ biến nhất và được sử dụng rộng rãi.[[4](https://moodle.org/mod/glossary/showentry.php?eid=7922&displayformat=dictionary)]

* **Tính năng chính:**
* Quản lý khóa học.
* Diễn đàn và thảo luận.
* Kiểm tra và đánh giá.
* Báo cáo và phân tích tiến độ học tập.
* Tính mở rộng và tùy chỉnh.
* **Ưu điểm:**
* Miễn phí và có mã nguồn mở, dễ dàng tùy chỉnh và mở rộng.
* Hỗ trợ nhiều công cụ học tập như bài kiểm tra, bài tập,...
* Phù hợp với nhiều cấp độ giáo dục, từ phổ thông đến đại học.
* **Nhược điểm:**
* Giao diện người dùng có thể khó sử dụng với những người không quen với các hệ thống công nghệ.
* Cần kỹ thuật và kỹ năng để tùy chỉnh và triển khai.

1. **Blackboard:**

Là hệ thống LMS thương mại, hệ thống quản lý học tập bảng đen ảo [[5](https://classin.vn/top-he-thong-lms-tot-nhat-hien-nay/)] một trong những hệ thống phổ biến nhất trong các trường đại học và cao đẳng, đặc biệt là ở Bắc Mỹ.

* **Tính năng chính:**
* Quản lý khóa học.
* Tính năng đánh giá.
* Diễn đàn thảo luận.
* Hỗ trợ đa phương tiện.
* Cộng tác và nhóm.
* **Ưu điểm:**
* Giao diện thân thiện, dễ dùng.
* Cung cấp nhiều công cụ đánh giá và theo dõi tiến độ học tập.
* Hỗ trợ mạnh mẽ cho các tính năng đa phương tiện và hợp tác.
* **Nhược điểm:**
* Trả phí, có thể khá tốn kém cho các tổ chức giáo dục nhỏ và ngân sách hạn chế.
* Dù dễ sử dụng, song có thể gặp khó khăn trong tùy chỉnh và mở rộng theo yêu cầu đặt biệt.

1. **Canvas:**

Canvas LMS là Hệ thống quản lý học tập nguồn mở đóng vai trò là trung tâm cho các lớp học trực tuyến, kết hợp và trực tiếp [[6]( https:/www.instructure.com/en-au/lms-learning-management-system)].

* **Tính năng chính:**
* Quản lý khóa học và bài tập.
* Công cụ giao tiếp.
* Chấm điểm và đánh giá.
* Tích hợp các công cụ khác.
* Dễ dàng tích hợp các công nghệ mới và dễ sử dụng.
* **Ưu điểm:**
* Giao diện người dùng trực quan và dễ sử dụng.
* Cung cấp và hỗ trợ tích hợp các công cụ linh hoạt của bên thứ ba và dễ dàng tùy chỉnh.
* Có phiên bảng mã nguồn mở miễn phí cho các tổ chức ngân sách hạn chế.
* **Nhược điểm:**
* Phiên bản thương mại có chi phí khá cao đối với các tổ chức giáo dục lớn.
* Mặc dù dễ sử dụng, nhưng có thể thiếu một số tính năng tùy chỉnh khi so với các hệ thống khác như Moodle.

# Cơ sở lý thuyết:

## **Cơ sở lý thuyết**

### 2.1.1. Khái niệm về hệ thống thông tin quản lý.

Hệ thống thông tin quản lý (Management Information System - MIS) là một tổ hợp các thành phần có sự liên kết chặt chẽ với nhau, bao gồm phần cứng, phần mềm, cơ sở dữ liệu, quy trình xử lý và con người, nhằm hỗ trợ cho việc thu thập, lưu trữ, xử lý và phân phối thông tin một cách hiệu quả trong tổ chức. MIS không chỉ đơn thuần cung cấp thông tin thô mà còn tổng hợp, phân tích và trình bày thông tin dưới dạng hữu ích, phục vụ cho các cấp quản lý trong việc ra quyết định, lập kế hoạch, điều phối và kiểm soát hoạt động.

Trong lĩnh vực giáo dục, đặc biệt là tại các trường đại học và cao đẳng, hệ thống thông tin quản lý đóng vai trò thiết yếu. Nó hỗ trợ việc quản lý hồ sơ sinh viên, theo dõi kết quả học tập, xây dựng và quản lý kế hoạch đào tạo, phân bổ nguồn lực giảng dạy, tổ chức thi cử, và nhiều hoạt động học thuật khác. Hệ thống này giúp các cơ sở giáo dục nâng cao hiệu quả quản lý, đảm bảo sự chính xác, minh bạch và kịp thời trong việc xử lý thông tin sinh viên, đồng thời hỗ trợ quá trình cải tiến chất lượng đào tạo.

Một phân hệ quan trọng trong MIS giáo dục là **hệ thống quản lý tiến độ học tập**. Đây là hệ thống chuyên biệt nhằm theo dõi, đánh giá, lưu trữ và báo cáo toàn bộ quá trình học tập của sinh viên từ lúc nhập học đến khi tốt nghiệp.

Hệ thống quản lý tiến độ học tập cho phép:

* **Ghi nhận kết quả học tập** của từng học kỳ, bao gồm điểm số, xếp loại học lực, số tín chỉ tích lũy.
* **Theo dõi quá trình học tập**: Hệ thống lưu trữ lịch sử học phần đã học, số tín chỉ còn thiếu, học phần nợ, số lần học lại, thi lại nếu có.
* **Đánh giá toàn diện**: Tính toán GPA (Grade Point Average), xếp loại học lực (Xuất sắc, Giỏi, Khá, Trung bình...), và đưa ra các thống kê học tập chi tiết.
* **Cảnh báo sớm**: Phát hiện các trường hợp sinh viên có kết quả học tập yếu kém, đưa ra cảnh báo học vụ hoặc đề xuất hỗ trợ kịp thời.
* **Lập kế hoạch cá nhân hóa**: Cung cấp dữ liệu cho việc tư vấn học tập, hướng dẫn sinh viên xây dựng lộ trình học phù hợp với khả năng và mục tiêu.

Hệ thống được xây dựng nhằm đạt được các mục tiêu cụ thể như:

* **Cung cấp công cụ minh bạch** cho sinh viên theo dõi tiến độ học tập cá nhân theo từng kỳ học và toàn bộ chương trình đào tạo.
* **Hỗ trợ quản lý** trong việc đánh giá năng lực học tập của sinh viên một cách nhanh chóng và chính xác.
* **Nâng cao hiệu quả quản lý** học vụ, giảm thiểu sai sót trong việc cập nhật điểm số, tính toán kết quả.
* **Đảm bảo tính liên tục** và hệ thống trong công tác quản lý.

### 2.1.2. Tiến độ học tập của sinh viên.

Tiến độ học tập của sinh viên là chỉ số phản ánh mức độ hoàn thành chương trình đào tạo dựa trên các học phần đã đăng ký và kết quả đạt được trong từng học kỳ. Đây là căn cứ quan trọng để đánh giá khả năng tiếp thu kiến thức, năng lực học tập và sự tuân thủ kế hoạch đào tạo của mỗi sinh viên.

Tiến độ học tập thường được thể hiện thông qua các yếu tố sau:

* **Số tín chỉ đã hoàn thành:** Là tổng số tín chỉ mà sinh viên đã tích lũy sau khi hoàn thành và đạt yêu cầu các học phần, so với tổng số tín chỉ yêu cầu để tốt nghiệp.
* **Kết quả học tập của từng học phần:**
* Điểm số: Kết quả thi hoặc đánh giá kết thúc học phần dưới dạng điểm số thang 10.
* Điểm chữ: Quy đổi điểm số sang thang điểm chữ (A, B+, B, C+, C, D+, D, F) theo quy chế đào tạo.
* Điểm hệ 4: Quy đổi điểm số theo thang điểm 4 để tính GPA.
* **Số học phần chưa đạt hoặc bị nợ:**
* Bao gồm những môn học sinh viên chưa hoàn thành, bị điểm dưới mức đạt yêu cầu, hoặc buộc phải học lại/thi lại.
* Đây là yếu tố quan trọng ảnh hưởng đến khả năng hoàn thành chương trình đào tạo đúng hạn.
* **GPA học kỳ và GPA tích lũy:**
* GPA học kỳ: Trung bình điểm hệ 4 của các học phần trong một học kỳ.
* GPA tích lũy: Trung bình điểm hệ 4 của tất cả các học phần đã hoàn thành kể từ khi bắt đầu chương trình học.
* Hai chỉ số này dùng để đánh giá học lực theo từng thời kỳ cũng như toàn bộ quá trình học.

Bên cạnh đó, việc quản lý tiến độ học tập có vai trò cực kỳ quan trọng đối với sinh viên và nhà trường:

* Giúp sinh viên chủ động kiểm soát lộ trình học:
* Sinh viên có thể theo dõi được những học phần đã hoàn thành, học phần còn thiếu, cũng như kết quả học tập hiện tại.
* Từ đó, sinh viên dễ dàng lên kế hoạch đăng ký môn học phù hợp cho các học kỳ tiếp theo nhằm hoàn thành đúng tiến độ.
* Phát hiện sớm các học phần chưa đạt yêu cầu:
* Hệ thống sẽ ghi nhận các học phần có kết quả dưới chuẩn để sinh viên kịp thời học lại, thi lại hoặc cải thiện điểm số.
* Tránh tình trạng dồn nợ quá nhiều tín chỉ, ảnh hưởng đến tiến độ ra trường.
* Hỗ trợ lập kế hoạch học lại hoặc cải thiện học lực:
* Dựa trên các chỉ số GPA và số tín chỉ tích lũy, sinh viên có thể điều chỉnh chiến lược học tập, tập trung vào những môn trọng điểm nhằm cải thiện thành tích chung.
* Đáp ứng điều kiện tốt nghiệp:
* Hệ thống theo dõi các tiêu chí cần thiết như tổng số tín chỉ, học phần bắt buộc, chuẩn đầu ra điểm số để hỗ trợ xét tốt nghiệp chính xác và kịp thời.

2.1.3. Cấu trúc dữ liệu trong hệ thống.

Hệ thống quản lý tiến độ học tập sử dụng các mô hình dữ liệu liên kết chặt chẽ với nhau, bao gồm:

* **Người dùng (User)**: Lưu trữ thông tin cá nhân của sinh viên, bao gồm tên đăng nhập, mật khẩu, email, số điện thoại, ảnh đại diện và ngày tạo tài khoản.
* **Học phần (Course)**: Quản lý thông tin về các môn học trong chương trình đào tạo, bao gồm mã học phần, tên học phần và số tín chỉ tương ứng.
* **Điểm số (Score)**: Liên kết giữa sinh viên và học phần, lưu trữ kết quả học tập gồm điểm số (theo thang 10), điểm chữ (theo hệ chữ cái) và điểm hệ 4 (dùng để tính GPA).
* **Học kỳ (Semester)**: Ghi nhận thông tin về từng học kỳ cụ thể trong năm học, bao gồm tên học kỳ (ví dụ: "Học kỳ 1", "Học kỳ 2") và năm học tương ứng (ví dụ: "2021-2022").

Các mô hình dữ liệu này được thiết kế theo kiểu liên kết khóa ngoại thông qua các ObjectId trong MongoDB, đảm bảo khả năng mở rộng linh hoạt và truy xuất dữ liệu hiệu quả.

Hệ thống sử dụng MongoDB làm hệ quản trị cơ sở dữ liệu chính. MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL hướng tài liệu (Document-Oriented Database), rất phù hợp với các hệ thống yêu cầu khả năng linh hoạt cao trong thiết kế và quản lý dữ liệu.

Một số đặc điểm nổi bật của MongoDB:

* **Cấu trúc dữ liệu linh hoạt:**
* Khác với các cơ sở dữ liệu quan hệ (RDBMS) truyền thống, MongoDB không sử dụng bảng (table) mà lưu trữ dữ liệu trong các collection chứa các document dạng BSON (Binary JSON).
* Các document trong cùng một collection có thể có cấu trúc khác nhau, cho phép thay đổi schema mà không ảnh hưởng đến toàn bộ cơ sở dữ liệu.
* **Hiệu suất cao:**
* MongoDB hỗ trợ cơ chế caching (ghi đệm) trên RAM, giảm thiểu truy cập ổ đĩa vật lý, từ đó tăng tốc độ đọc và ghi dữ liệu.
* **Khả năng mở rộng mạnh mẽ:**
* MongoDB hỗ trợ sharding (phân mảnh dữ liệu) giúp hệ thống dễ dàng mở rộng theo chiều ngang (horizontal scaling) khi khối lượng dữ liệu tăng cao.
* **Tính sẵn sàng cao:**
* Với mô hình replica set (sao lưu tự động giữa các node), MongoDB đảm bảo dữ liệu có tính toàn vẹn và hệ thống có thể tự động phục hồi trong trường hợp lỗi xảy ra.



*Hình 1**: Cơ sở dữ liệu MongoDB.*

Tóm tắt ưu điểm khi lựa chọn MongoDB:

* Không yêu cầu thiết kế schema cứng nhắc, phù hợp với những thay đổi trong yêu cầu quản lý đào tạo.
* Tốc độ truy vấn nhanh nhờ lưu trữ tài liệu kiểu JSON.
* Dễ dàng tích hợp với Node.js và các công nghệ hiện đại.
* Hỗ trợ lưu trữ dữ liệu phi cấu trúc, lý tưởng cho hệ thống giáo dục với dữ liệu sinh viên đa dạng.

### 2.1.4. Tính toán GPA và quản lý học phần.

GPA (Grade Point Average) là một chỉ số quan trọng dùng để đánh giá chất lượng học tập của sinh viên trong từng học kỳ cũng như trong toàn bộ quá trình học tập tại trường. Hệ thống quản lý tiến độ học tập cung cấp các chức năng hỗ trợ tự động tính toán và theo dõi các chỉ số liên quan, cụ thể:

* GPA tích lũy (Cumulative GPA):
* Là tung bình cộng có trọng số của tất cả các học phần mà sinh viên đã hoàn thành.
* Việc sử dụng trọng số tín chỉ đảm bảo rằng các học phần có khối lượng kiến thức lớn hơn sẽ ảnh hưởng nhiều hơn đến kết quả chung.
* Số tín chỉ tích lũy:
* Là tổng số tín chỉ của tất cả các học phần mà sinh viên đã đạt yêu cầu (không bao gồm các học phần bị điểm F hoặc chưa đạt chuẩn).
* Thông tin này giúp sinh viên dễ dàng đối chiếu với lộ trình đào tạo và kế hoạch tốt nghiệp.
* Môn nợ:
* Là danh sách các học phần mà sinh viên chưa đạt yêu cầu, được xác định thông qua điểm chữ là "F".
* Hệ thống thống kê tự động số lượng và danh sách các học phần bị nợ, hỗ trợ sinh viên lên kế hoạch học lại hoặc cải thiện kết quả.
* Cảnh báo số tín chỉ vượt giới hạn:
* Hệ thống tự động kiểm tra tổng số tín chỉ tích lũy.
* Nếu sinh viên tích lũy số tín chỉ vượt quá một ngưỡng quy định (ví dụ: 152 tín chỉ đối với chương trình đào tạo tiêu chuẩn), hệ thống sẽ phát sinh cảnh báo.
* Tính năng này giúp sinh viên và cố vấn học tập nắm bắt kịp thời tình trạng học tập, tránh các trường hợp phát sinh không cần thiết hoặc gây ảnh hưởng đến lộ trình tốt nghiệp.

### 2.1.5. Công nghệ và giao diện người dùng.

Hệ thống quản lý tiến độ học tập được xây dựng theo mô hình kiến trúc **MVC (Model - View - Controller)**, đảm bảo sự phân tách rõ ràng giữa các thành phần trong ứng dụng, giúp dễ dàng mở rộng, bảo trì và phát triển trong tương lai.

Công nghệ sử dụng:

* *Backend:*
* **Node.js:** Nền tảng JavaScript phía server giúp xây dựng ứng dụng nhanh và hiệu quả.
* **Express:** Framework đơn giản và mạnh mẽ cho Node.js, hỗ trợ định tuyến (routing), middleware, xử lý yêu cầu HTTP một cách linh hoạt.
* **Mongoose:** Thư viện quản lý mô hình dữ liệu, hỗ trợ thao tác với MongoDB một cách dễ dàng và có tổ chức.
* *Frontend:*
* **Handlebars:** Công cụ template engine giúp tạo ra các trang web động, dễ dàng nhúng dữ liệu từ server vào giao diện.
* **Bootstrap:** Bộ thư viện CSS và JavaScript giúp xây dựng giao diện responsive, đảm bảo tính thân thiện với người dùng trên nhiều thiết bị.
* **JavaScript**: Hỗ trợ xử lý các thao tác động trên giao diện người dùng như lọc dữ liệu, cập nhật thông tin mà không cần reload trang.
* *Cơ sở dữ liệu:*
* **MongoDB**: Cơ sở dữ liệu NoSQL dạng document, lưu trữ dữ liệu linh hoạt dưới dạng BSON, rất phù hợp cho hệ thống quản lý tiến độ học tập cần thường xuyên thay đổi và mở rộng.

Một số thư viện hỗ trợ khác giúp cho trải nghiệm người dùng tối ưu hơn:

* **body-parser:** Middleware hỗ trợ phân tích (parse) dữ liệu từ request body, thường dùng với các form HTML.
* **connect-flash:** Hỗ trợ lưu trữ và truyền tải thông báo giữa các request (ví dụ: thông báo lỗi đăng nhập).
* **express-flash**: Tích hợp connect-flash vào Express dễ dàng hơn.
* **express-session:** Quản lý session người dùng (ví dụ: lưu thông tin đăng nhập sau khi xác thực thành công).
* **jsonwebtoken**: Hỗ trợ tạo và xác thực token JWT trong các chức năng bảo mật nâng cao.
* **morgan**: Middleware ghi log các request HTTP ra console, tiện lợi cho việc kiểm tra và debug trong quá trình phát triển.
* **multer:**  Middleware hỗ trợ upload file (ví dụ: upload avatar người dùng).
* **nodemon:** Công cụ tự động reload server Node.js khi có thay đổi mã nguồn, hỗ trợ quá trình phát triển nhanh chóng hơn.
* **node-sass:** Biên dịch các tập tin .scss thành .css, hỗ trợ viết giao diện frontend dễ dàng hơn, tùy biến cao hơn Bootstrap thuần.

Các tính năng nổi bật của giao diện người dùng:

* Xem danh sách các học phần đã học kèm theo thông tin chi tiết như số tín chỉ, điểm số, điểm chữ.
* Thêm mới học phần vào học kỳ hiện tại.
* Cập nhật, chỉnh sửa điểm số của từng học phần đã đăng ký.
* Lọc và tìm kiếm dữ liệu theo năm học và học kỳ cụ thể.
* Thêm học kỳ mới để theo dõi tiến độ học tập qua từng giai đoạn.
* Tính toán và hiển thị GPA học kỳ và GPA tích lũy tự động theo thời gian thực.
* Giao diện được thiết kế trực quan, dễ thao tác ngay cả với người dùng không chuyên về công nghệ, hỗ trợ tối ưu hóa trải nghiệm sử dụng.

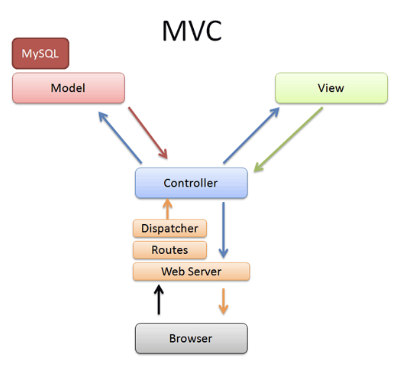
### 2.1.6. Vai trò và ý nghĩa với sinh viên.

Hệ thống quản lý tiến độ học tập đóng vai trò như một “trợ lý học tập số” giúp sinh viên chủ động hơn trong quá trình học đại học. Các lợi ích cụ thể gồm:

* **Theo dõi kết quả học tập**: Sinh viên có thể dễ dàng tra cứu điểm số của từng học phần, điểm trung bình học kỳ và tổng kết toàn khóa. Thông tin được trình bày trực quan, giúp sinh viên nắm bắt kịp thời tình hình học tập của bản thân.
* **Lập kế hoạch học tập**: Hệ thống hỗ trợ sinh viên nhận diện các môn học chưa đạt yêu cầu, từ đó xây dựng kế hoạch học lại hoặc học cải thiện phù hợp với tiến độ cá nhân.
* **Tự đánh giá và điều chỉnh**: Dựa trên các dữ liệu học tập được cung cấp, sinh viên có thể tự đánh giá khả năng của mình theo từng giai đoạn. Điều này giúp điều chỉnh phương pháp học và chiến lược học tập hiệu quả hơn.
* **Tăng tính chủ động**: Khi có cái nhìn tổng thể và liên tục về tiến độ học tập, sinh viên sẽ phát triển được thói quen tự quản lý thời gian, chủ động trong việc học và nâng cao tính kỷ luật cá nhân.

## **Cơ sở kỹ thuật**

### 2.2.1. Kiến trúc hệ thống

****

*Hình 2**: Minh họa mô hình MVC (Model - View - Controller) [[2](https://studycodingblog.wordpress.com/2016/11/30/viet-web-bang-mvc-5-la-gi-va-tai-sao-phai-dung-no/)]*

Hệ thống quản lý tiến độ học tập cho sinh viên được xây dựng theo mô hình **MVC (Model - View - Controller)** nhằm tách biệt các lớp xử lý, giao diện và dữ liệu, giúp tăng tính linh hoạt, dễ bảo trì và mở rộng.

* **Model**: Quản lý dữ liệu và logic nghiệp vụ, tương tác trực tiếp với cơ sở dữ liệu MongoDB thông qua thư viện Mongoose. Trong hệ thống này, các mô hình như Score, Course, Semester, User được xây dựng bằng thư viện Mongoose của Node.js để giao tiếp với MongoDB.
* **View**: Giao diện người dùng được xây dựng bằng **Handlebars (hbs)**, kết hợp với **Bootstrap** để đảm bảo giao diện thân thiện, dễ sử dụng trên nhiều thiết bị.
* **Controller**: Xử lý logic nghiệp vụ, nhận dữ liệu từ người dùng, tương tác, xử lý các yêu cầu từ người dùng, điều phối dữ liệu giữa Model và View, được xây dựng với Node.js và Express.

### 2.2.2. Công nghệ sử dụng

#### **a. Ngôn ngữ lập trình**

* **JavaScript:** Ngôn ngữ lập trình chính, được sử dụng cho cả backend (Node.js) và frontend. JavaScript hỗ trợ lập trình bất đồng bộ hiệu quả, phù hợp với các ứng dụng web hiện đại.
* **HTML/CSS**: Ngôn ngữ đánh dấu và định dạng giao diện người dùng, giúp hiển thị thông tin trực quan và thẩm mỹ.
* **Handlebars:** Công cụ template engine cho phép chèn dữ liệu động từ controller lên giao diện HTML một cách linh hoạt và dễ bảo trì.

#### **b. Nền tảng backend**

* **Node.js:** Môi trường chạy JavaScript phía server, nổi bật với khả năng xử lý bất đồng bộ (non-blocking I/O) và hiệu suất cao, phù hợp cho các ứng dụng thời gian thực và web API.
* **Express.js:** Framework phát triển dựa trên Node.js, hỗ trợ tổ chức code theo mô hình RESTful API, đơn giản hóa việc xây dựng các tuyến (routes), xử lý middleware và quản lý phiên làm việc (session).

#### **c. Cơ sở dữ liệu**

* **MongoDB:** Hệ quản trị cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng tài liệu JSON (BSON). Ưu điểm là khả năng mở rộng linh hoạt, phù hợp với dữ liệu có cấu trúc thay đổi như hồ sơ sinh viên và kết quả học tập.
* **Mongoose:** Thư viện ODM (Object Data Modeling) cho MongoDB, hỗ trợ định nghĩa schema, kiểm tra tính hợp lệ (validation), và cung cấp các phương thức mạnh mẽ để truy vấn và cập nhật dữ liệu.

#### **d. Frontend và giao diện**

#### **Bootstrap:** Bộ công cụ CSS phổ biến hỗ trợ thiết kế giao diện web responsive, hiện đại, dễ dàng tùy chỉnh và mở rộng.

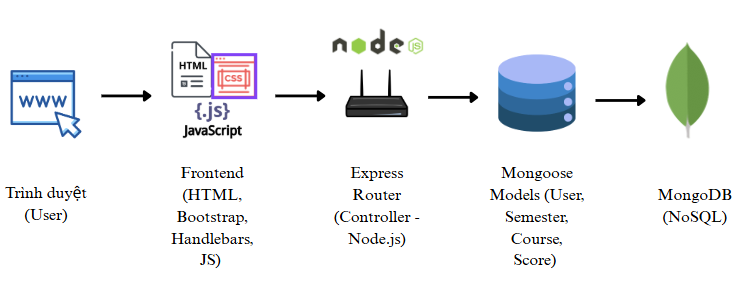
#### **JavaScript DOM Manipulation:** Sử dụng các phương thức DOM để thao tác dữ liệu bảng học phần, hỗ trợ các tính năng như phân trang, tìm kiếm, lọc học phần ngay trên giao diện.

#### **AJAX/Fetch API**: Kỹ thuật gửi yêu cầu bất đồng bộ đến server, cho phép cập nhật dữ liệu giao diện mà không cần tải lại toàn bộ trang web, cải thiện trải nghiệm người dùng.

#### **e. Công cụ hỗ trợ**

* **Visual Studio Code (VS Code):** Môi trường phát triển tích hợp (IDE) chính, hỗ trợ nhiều tiện ích mở rộng phục vụ lập trình Node.js, JavaScript và MongoDB.
* **Git/GitHub:** Hệ thống quản lý phiên bản mã nguồn phân tán, hỗ trợ lưu trữ, chia sẻ và cộng tác phát triển phần mềm hiệu quả.
* **MongoDB Compass:** Giao diện đồ họa (GUI) cho MongoDB, hỗ trợ người phát triển dễ dàng quản lý, truy vấn, chỉnh sửa và phân tích cơ sở dữ liệu.

### 2.2.3. Sơ đồ triển khai kỹ thuật



*Hình 3**: Sơ đồ triển khai kỹ thuật của hệ thống.*

### 2.2.4. Mô hình cơ sở dữ liệu chính

* **Users (Người dùng)**
  + *username*: String
  + *password*: String
  + *phone*: String
  + *email:* String
  + *avatar:* String
* **Courses (Học phần)**
  + *maHocPhan*: String
  + *tenHocPhan*: String
  + *soTinChi*: Number
* **Scores (Điểm số học phần)**
  + *HocPhan*: ObjectId (ref: Courses)
  + *diemSo*: Number
  + *diemChu*: String
  + *tenHocKy*: String
  + *namHoc*: String
* **Semesters (Học kỳ)**
  + *tenHocKy*: String
  + *namHoc*: String
  + *score*: [ObjectId] (ref: Scores)

### 2.2.5. Tính năng kỹ thuật nổi bật

* Phân trang học phần bằng JavaScript

Hệ thống hỗ trợ phân trang (pagination) danh sách học phần trực tiếp trên giao diện người dùng bằng JavaScript, giúp hiển thị dữ liệu gọn gàng, dễ theo dõi mà không cần tải lại toàn bộ trang (reload).

* Tính toán tự động điểm số

Hệ thống tự động tính toán:

* Điểm trung bình học kỳ (Semester GPA): Dựa trên điểm số và số tín chỉ của các học phần trong từng học kỳ.
* GPA tích lũy (Cumulative GPA): Trung bình cộng có trọng số của tất cả học phần đã hoàn thành.
* Theo dõi số tín chỉ và cảnh báo

Hệ thống kiểm tra tổng số tín chỉ sinh viên đã tích lũy. Nếu số tín chỉ vượt ngưỡng quy định (ví dụ: 152 tín chỉ), hệ thống sẽ tự động đưa ra cảnh báo để sinh viên điều chỉnh kế hoạch học tập hợp lý.

* Giao diện tương tác với Modal

Khi thêm học phần mới, hệ thống sử dụng Modal (cửa sổ pop-up) giúp sinh viên nhập nhanh thông tin mà không cần chuyển trang, nâng cao trải nghiệm người dùng (UX).

* Tìm kiếm học phần thông minh

Hệ thống hỗ trợ tìm kiếm theo tên học phần hoặc mã học phần thông qua bộ lọc JavaScript trên giao diện, giúp sinh viên nhanh chóng tìm thấy môn học cần thao tác.

## **Ngôn ngữ lập trình**

Hệ thống được xây dựng chủ yếu bằng **JavaScript**, một ngôn ngữ lập trình hiện đại, linh hoạt, và phổ biến trong phát triển web. Cụ thể, hệ thống sử dụng JavaScript ở cả phía **client (frontend)** và **server (backend)**, giúp đồng bộ trong quá trình phát triển và giảm độ phức tạp khi làm việc với nhiều ngôn ngữ khác nhau.

1. **JavaScript – Ngôn ngữ lập trình chính**

* **Ở phía client (trình duyệt)**:
* Sử dụng JavaScript để xử lý các thao tác tương tác người dùng như tìm kiếm học phần, phân trang, gửi yêu cầu thêm học phần thông qua Fetch API.
* Kết hợp với Bootstrap và DOM API để cải thiện trải nghiệm người dùng.
* **Ở phía server (backend)**:
* Sử dụng JavaScript chạy trên môi trường **Node.js** để xây dựng ứng dụng web hiệu suất cao.
* Xử lý các request, thao tác với cơ sở dữ liệu, và phản hồi lại client thông qua framework **Express.js**.

1. **HTML/CSS**

* Dùng để xây dựng cấu trúc và định dạng giao diện người dùng.
* Kết hợp với **Bootstrap** để đảm bảo giao diện tương thích với nhiều thiết bị, đẹp mắt và dễ sử dụng.

1. **Handlebars (hbs)**

* Đây là một template engine dựa trên HTML giúp kết nối dữ liệu từ backend lên frontend.
* Dễ viết, dễ hiểu, hỗ trợ logic đơn giản như vòng lặp, điều kiện, phù hợp với hệ thống quản lý điểm và học phần.

1. **Mongoose Schema**

* Dùng để định nghĩa cấu trúc dữ liệu trong MongoDB bằng JavaScript.
* Cung cấp các phương thức mạnh mẽ để thao tác, validate, và bảo vệ dữ liệu.

***Tóm lại:***

|  |  |  |
| --- | --- | --- |
| **Thành phần hệ thống** | **Ngôn ngữ** | **Công nghệ đi kèm** |
| Giao diện người dùng | HTML, CSS, JavaScript | Bootstrap, Handlebars |
| Logic phía client | JavaScript | Fetch API, DOM |
| Logic phía server | JavaScript | Node.js, Express |
| Cơ sở dữ liệu | JavaScript (Schema) | Mongoose, MongoDB |

Bảng 1: Bảng tóm tắt ngôn ngữ lập trình của hệ thống.

## **Đề xuất công thức, Lý thuyết của tác giả**

Trong quá trình nghiên cứu và xây dựng hệ thống quản lý tiến độ học tập cho sinh viên, tác giả đề xuất một số công thức và lý thuyết nhằm phục vụ cho việc đánh giá kết quả học tập, tính toán điểm trung bình tích lũy và phân loại sinh viên. Những công thức này đóng vai trò hỗ trợ cho các chức năng thống kê, đánh giá học lực và đưa ra cảnh báo học vụ nếu cần thiết.

### 2.4.1. Công thức chuyển đổi điểm số sang thang điểm 4

Để đồng bộ với các hệ thống giáo dục quốc tế hoặc đánh giá học lực một cách chuẩn hóa, tác giả đề xuất bảng chuyển đổi điểm số sang thang điểm 4 như sau:

|  |  |  |
| --- | --- | --- |
| **Điểm chữ** | **Điểm số tương ứng** | **Thang điểm 4** |
| A | 9.0 - 10 | 4.0 |
| B+ | 8.0 - 8.9 | 3.5 |
| B | 7.0 - 7.9 | 3.0 |
| C+ | 6.5 - 6.9 | 2.5 |
| C | 5.5 - 6.4 | 2.0 |
| D+ | 5.0 - 5.4 | 1.5 |
| D | 4.0 - 4.9 | 1.0 |
| F | < 4.0 | 0.0 |

*Bảng 2**: Bảng chuyển đổi điểm số sang thang điểm 4*

**Hàm chuyển đổi điểm số sang GPA 4.0 được đề xuất như sau (viết dưới dạng logic):**

function convertTo4Scale(diemSo) {

if (diemSo >= 9.0) return 4.0;

if (diemSo >= 8.0) return 3.5;

if (diemSo >= 7.0) return 3.0;

if (diemSo >= 6.5) return 2.5;

if (diemSo >= 5.5) return 2.0;

if (diemSo >= 5.0) return 1.5;

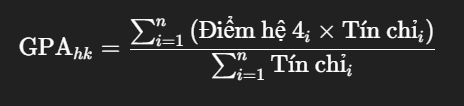
if (diemSo >= 4.0) return 1.0;

return 0.0;

}

### 2.4.2. Công thức tính điểm trung bình học kỳ (GPA học kỳ)

GPA học kỳ được tính dựa trên số tín chỉ và điểm từng môn học của học kỳ đó[[7](https://dsa.ctu.edu.vn/images/upload/vbanply/2021/Vanban/QD1813_QD_ban_hanh_Quy_dinh_cong_tac_hoc_vu_2021.pdf)]:

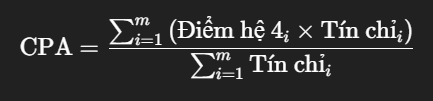


Trong đó:

* *n* là số môn học trong học kỳ
* Điểm hệ *4i*​ là điểm môn học đã được chuyển sang thang 4
* *Tín chỉ i​* là số tín chỉ của môn học đó

2.4.3. Công thức tính điểm trung bình tích lũy (CPA)

Tương tự GPA học kỳ, điểm trung bình tích lũy được tính trên toàn bộ số học phần đã học[[7](https://dsa.ctu.edu.vn/images/upload/vbanply/2021/Vanban/QD1813_QD_ban_hanh_Quy_dinh_cong_tac_hoc_vu_2021.pdf)]:



Trong đó:

* m là số lượng học phần đã học từ đầu khoá đến thời điểm hiện tại.

### 2.4.4. Quy tắc phân loại học lực

Dựa trên điểm trung bình tích lũy (CPA), học lực của sinh viên được phân loại như sau:

|  |  |
| --- | --- |
| **CPA** | **Xếp loại học lực** |
| 3.6 - 4.0 | Xuất sắc |
| 3.2 - 3.59 | Giỏi |
| 2.5 - 3.19 | Khá |
| 2.0 - 2.49 | Trung bình |
| 1.0 - 1.99 | Yếu |
| < 1.0 | Kém |

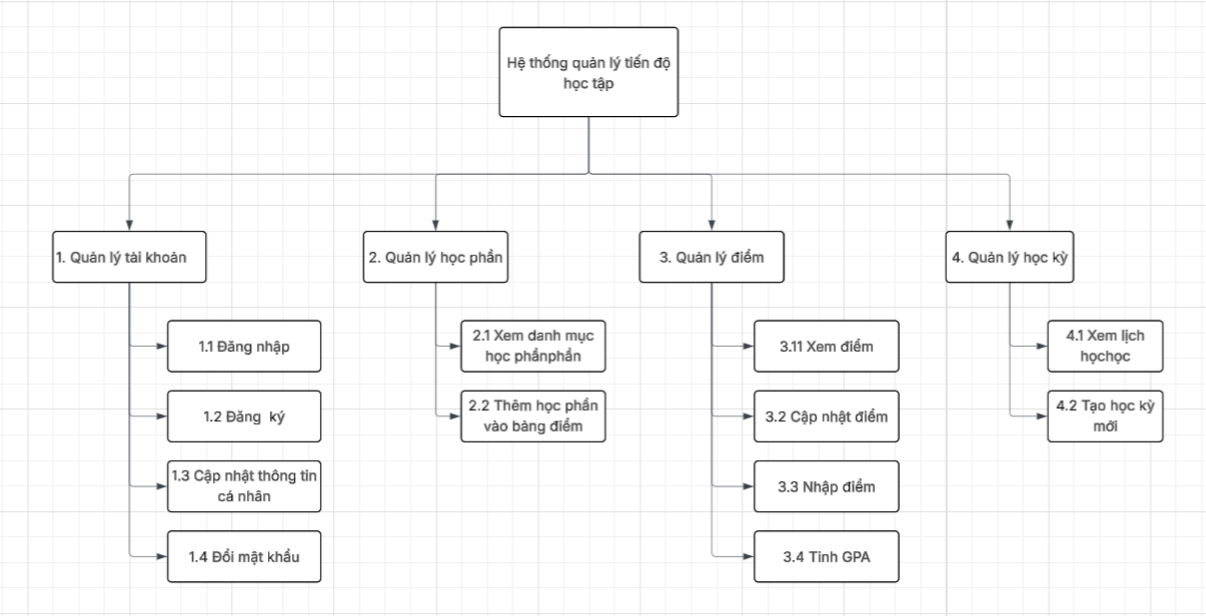
*Bảng 3**: Bảng tính điểm trung bình tích lũy.*

Những công thức và lý thuyết nêu trên được áp dụng trực tiếp vào hệ thống để hỗ trợ các chức năng tính toán tự động, đưa ra phân tích học tập và hỗ trợ quyết định quản lý học vụ. Việc chuẩn hóa quy trình đánh giá giúp hệ thống minh bạch hơn, dễ bảo trì và tương thích tốt với các hệ thống giáo dục hiện đại.

# Nội dung và kết quả nghiên cứu:

## **Mô hình phân rã chức năng**

Hệ thống được tổ chức theo dạng phân cấp từ chức năng tổng quát đến các chức năng chi tiết hơn như sau:

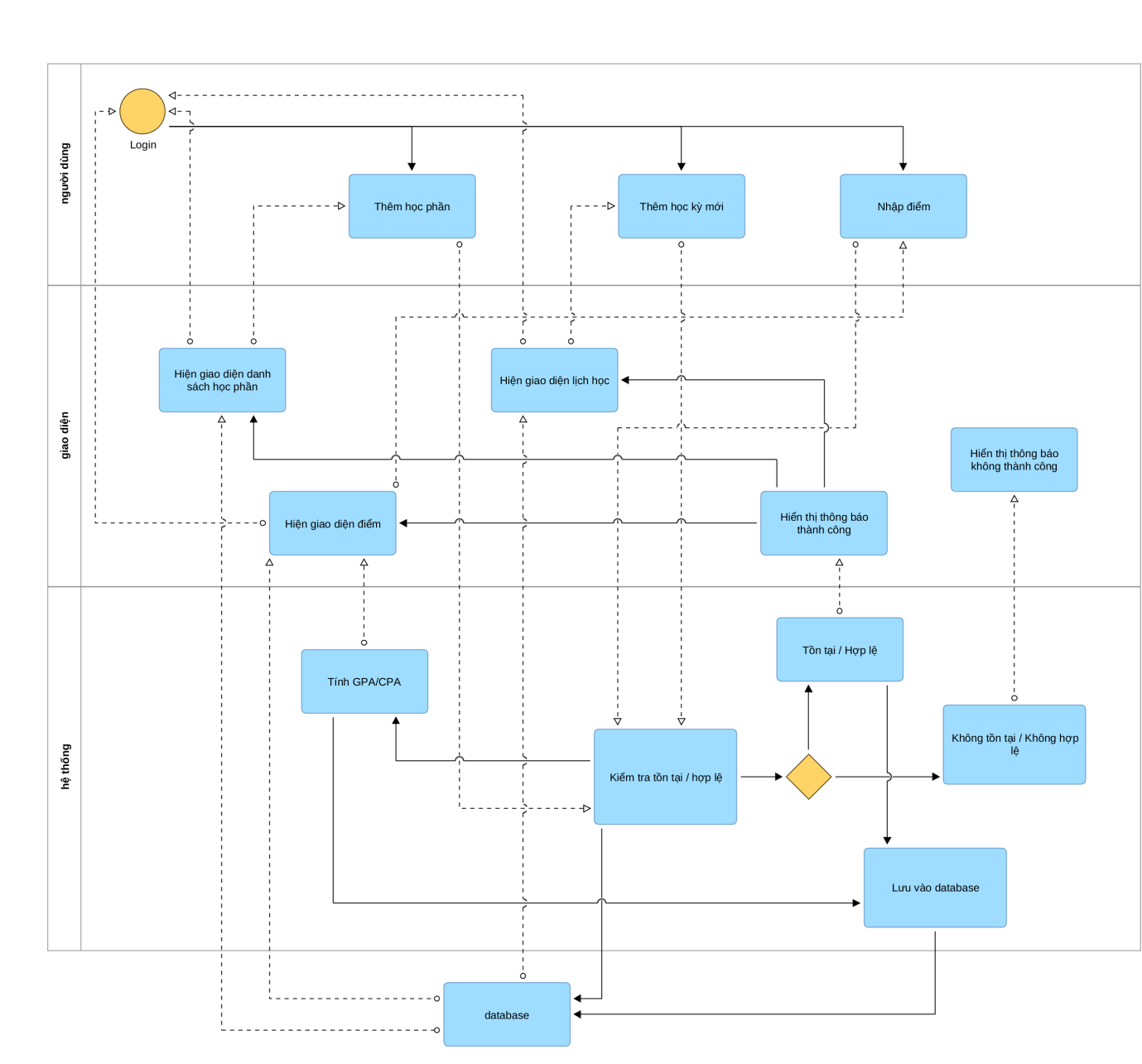


*Hình 4**: Mô hình phân rã chức năng của hệ thống.*

|  |  |  |
| --- | --- | --- |
| **Mã chức năng** | **Tên chức năng** | **Mô tả ngắn gọn** |
| 1.1 | Đăng nhập | Người dùng đăng nhập hệ thống |
| 1.2 | Đăng ký | Tạo tài khoản sinh viên |
| 1.3 | Cập nhật thông tin cá nhân | Chỉnh sửa username, email, SĐT, mật khẩu |
| 1.4 | Đổi mật khẩu | Nhập mật khẩu mới để thay đổi |
| 2.1 | Xem danh sách học phần | Hiển thị tất cả học phần có sẵn |
| 2.2 | Thêm học phần vào bảng điểm | Chọn học phần và thêm vào điểm |
| 3.1 | Xem điểm | Xem bảng điểm theo học kỳ/năm |
| 3.2 | Nhập điểm | Nhập diemSo, diemChu cho học phần |
| 3.3 | Cập nhật điểm | Chỉnh sửa điểm đã nhập |
| 3.4 | Tính GPA, xếp loại | Tự động tính điểm trung bình và phân loại học lực |
| 4.1 | Xem lịch học | Hiển thị các lớp học đã có trong học kỳ |
| 4.2 | Tạo học kỳ mới | Tạo học kỳ và thêm các học phần đã chọn |

*Bảng 4**: Bảng mô tả ngắn gọn chức năng của hệ thống.*

## **Mô hình hoạt động nghiệp vụ**

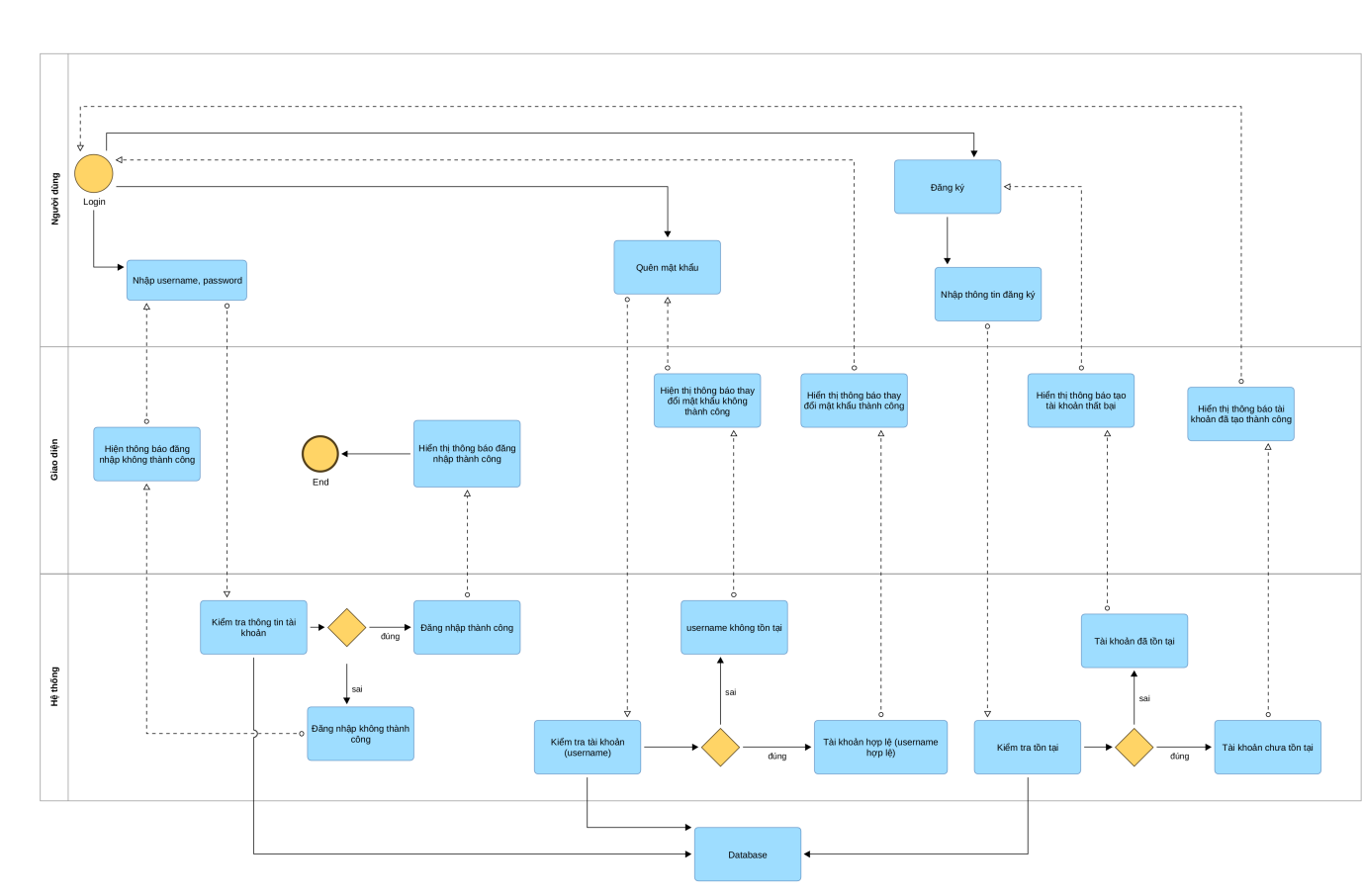


*Hình 5**: Mô hình hoạt động nghiệp vụ tổng quát của hệ thống.*

### **1. Đăng ký / Đăng nhập**

**Luồng xử lý:**

* Người dùng truy cập vào hệ thống và được chuyển đến trang **/login** hoặc **/register**
* Với đăng ký:
  + Người dùng nhập username, email, số điện thoại, mật khẩu.
  + Hệ thống kiểm tra trùng tên đăng nhập:
    - Nếu không trùng, hệ thống tạo mới tài khoản và lưu vào cơ sở dữ liệu.
    - Nếu trùng, thông báo lỗi cho người dùng.
* Với đăng nhập:
  + Người dùng nhập username và mật khẩu.
  + Hệ thống xác thực thông tin:
    - Nếu đúng, tạo **session** và chuyển đến trang tài khoản **/account**.
    - Nếu thông tin không hợp lệ, hiển thị thông báo lỗi.

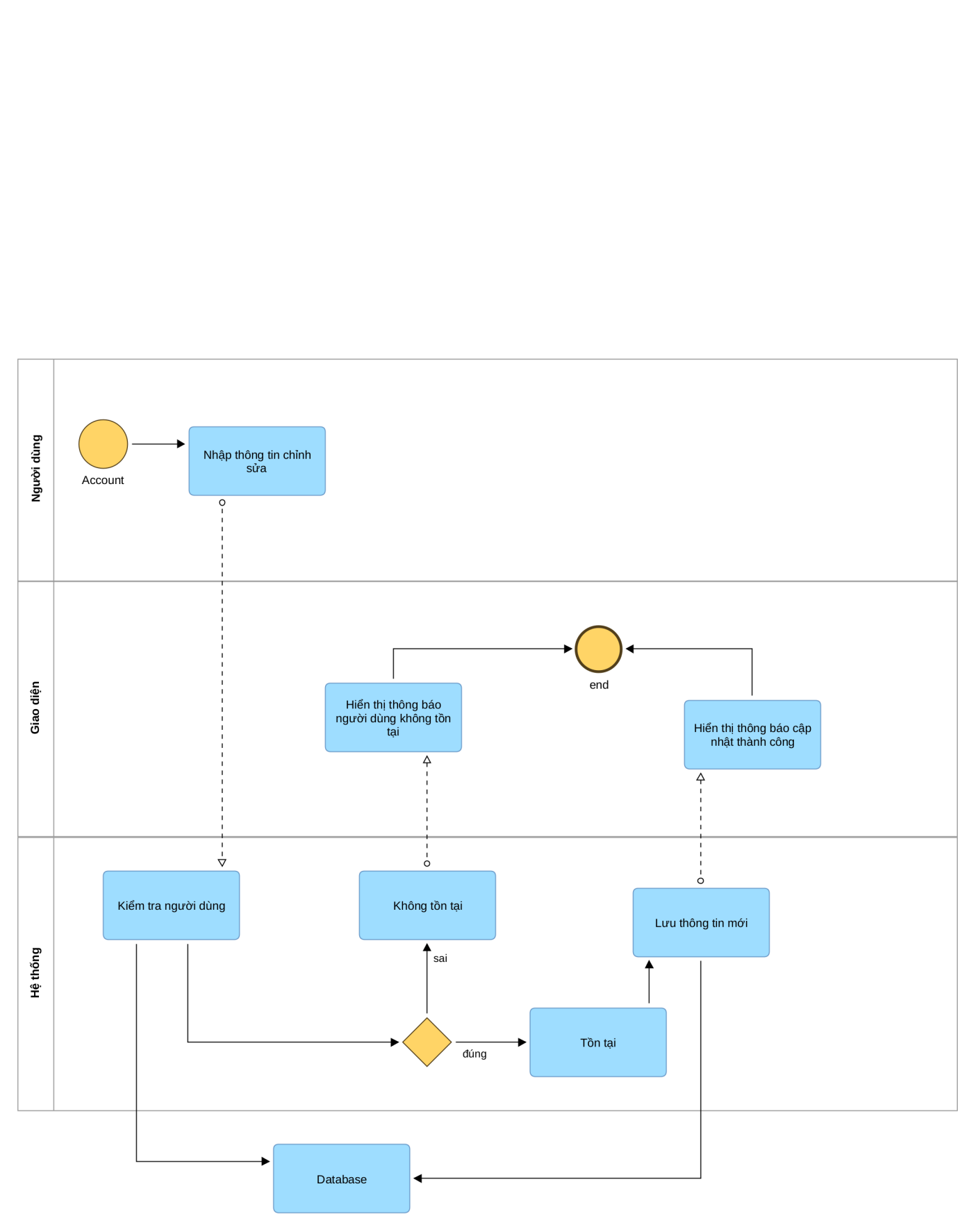


*Hình 6**: Mô hình hoạt động nghiệp vụ của Đăng Nhập/Đăng Ký.*

**2. Cập nhật thông tin cá nhân**

**Luồng xử lý:**

* Người dùng đăng nhập và vào trang **/account**.
* Tại đây, người dùng có thể cập nhật:
  + Tên
  + Email
  + Số điện thoại
  + Mật khẩu.
* Dữ liệu được gửi lên server → cập nhật CSDL.
* Hệ thống đồng thời cập nhật session hiện tại để đảm bảo thông tin mới được hiển thị tức thời.

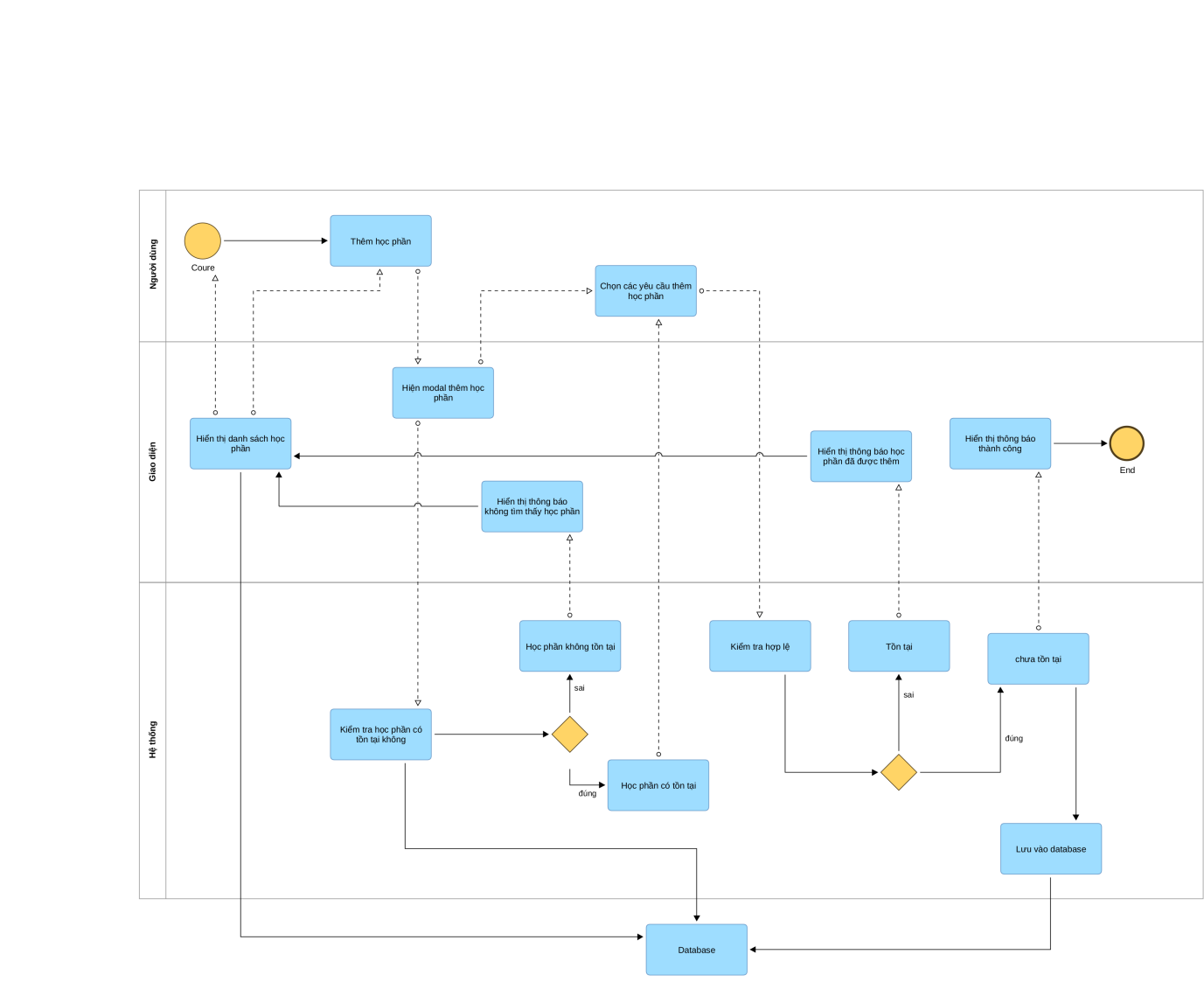


*Hình 7**: Mô hình hoạt động nghiệp vụ của Account.*

### **3. Quản lý học phần (Course)**

**Luồng xử lý:**

* Người dùng có thể:
  + Xem danh sách các học phần có trong hệ thống.
  + Tìm kiếm học phần theo tên hoặc mã học phần.
* Hệ thống kiểm tra:
  + Nếu học phần chưa có trong bảng điểm → Cho phép thêm
  + Nếu học phần đã có → Ẩn hoặc vô hiệu hóa nút thêm.
* Khi thêm học phần mới:
  + Hệ thống tạo bản ghi *Score* chứa thông tin học phần.



Hình 8: *Mô hình hoạt động nghiệp vụ của Course.*

### **4. Quản lý điểm (Score)**

**Luồng xử lý:**

* Hệ thống hiển thị điểm theo học kỳ và năm học.
* Cho phép cập nhật *diemSo, diemChu* cho từng học phần.
* Hệ thống tự động:
  + Tính GPA của từng học kỳ.
  + Trung bình tích lũy của tất cả học kỳ.
  + Xếp loại học lực (Xuất sắc, Giỏi, Khá, Trung bình, Yếu, Kém).
* Các điểm F được tổng hợp thành danh sách "môn nợ".

### **5. Quản lý học kỳ (Semester)**

**Luồng xử lý:**

* Hệ thống hiển thị lịch học theo học kỳ và năm học.
* Người dùng thêm học kỳ mới bằng cách:
  + Nhập tên học kỳ, năm học.
  + Chọn danh sách *Score* muốn gán cho học kỳ đó.
* Hệ thống kiểm tra:
  + Nếu tên học kỳ và năm học chưa tồn tại, cho phép lưu học kỳ mới.
  + Nếu trùng lặp, thông báo lỗi.
* Sau khi thêm, học kỳ sẽ chứa danh sách học phần đã chọn.

### **6. Theo dõi tiến độ học tập (Progress)**

**Luồng xử lý:**

* Hệ thống tổng hợp tất cả học phần từ *Score*:
  + Tính tổng số tín chỉ đã học.
  + Tính số tín chỉ đã rớt (điểm chữ F).
  + So sánh với tổng tín chỉ cần để ra trường.
* Nếu tổng số tín chỉ tích lũy vượt giới hạn cho phép, hệ thống sẽ đưa ra cảnh báo.

## **3.3. Mô hình thực thể quan hệ**

1. **Thực thể:**

* User (Người dùng):
* username
* password
* phone
* avatar
* email
* Course (Học phần):
* tenHocPhan
* maHocPhan
* soTinChi
* Score (Điểm):
* username
* diemSo
* diemChu
* HocPhan
* gioHoc
* Semester (Học kỳ):
* namHoc
* tenHocKy
* score[ ]

1. **Mối quan hệ giữa các thực thể:**

* User - Score:

Một User có nhiều Score (1:N).

→ Mỗi bản ghi Score gắn với 1 sinh viên.

* Course - Score:

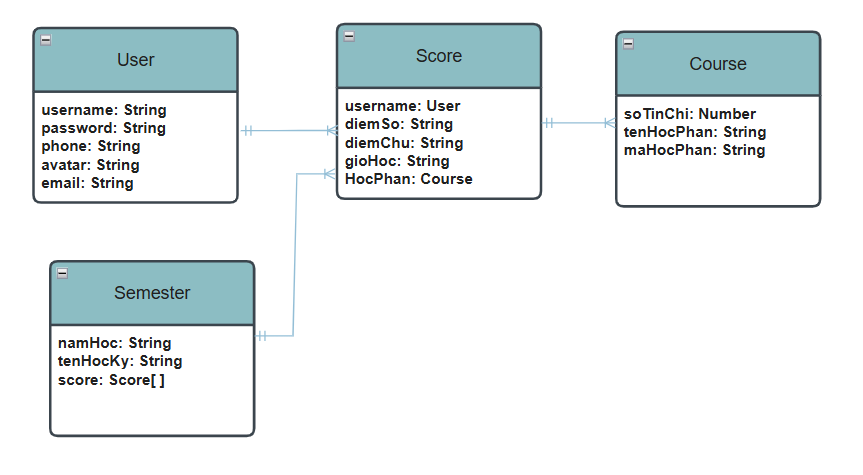
Một Course có thể xuất hiện trong nhiều Score (1:N).

→ Một môn học có nhiều sinh viên học.

* Semester - Score:

Một Semester có nhiều Score (1:N).

→ Một học kỳ chứa nhiều môn học của sinh viên.



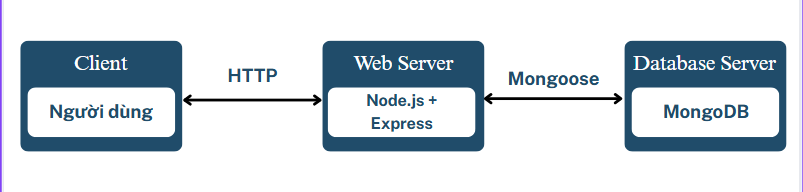
*Hình 9**: Mô hình thực thể quản hệ (dữ liệu) của hệ thống.*

## **3.4. Mô hình tổ chức hệ thống**

Hệ thống được tổ chức theo mô hình Client - Server:

* Client: Người dùng (sinh viên, quản trị viên) truy cập hệ thống qua trình duyệt web.
* Server:
* Web Server: Xử lý yêu cầu, xác thực người dùng, xử lý logic nghiệp vụ.
* Database Server: Lưu trữ dữ liệu người dùng, học phần, điểm số.

→ Các thành phần giao tiếp với nhau qua HTTP và kết nối database.

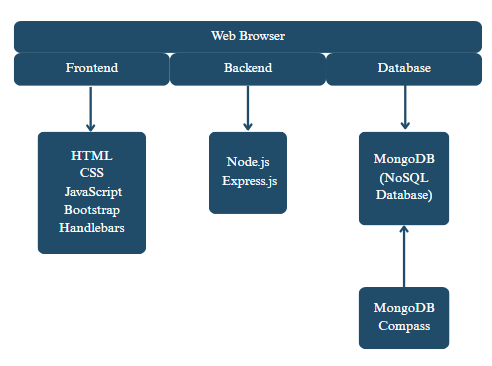


*Hình 10**: Mô hình tổng quan hệ thống.*

Các thành phần cài đặt như sau:

|  |  |
| --- | --- |
| **Thành phần** | **Công nghệ sử dụng** |
| Frontend (Giao diện) | HTML5, CSS3, JavaScript, Bootstrap, Handlebars |
| Backend (Xử lý server) | Node.js + Express.js |
| Cơ sở dữ liệu | MongoDB (NoSQL Database) |

*Bảng 5**:Các thành phần cài đặt của hệ thống.*



*Hình 11**: Mô hình cài đặt hệ thống.*

# Kết luận và đề nghị

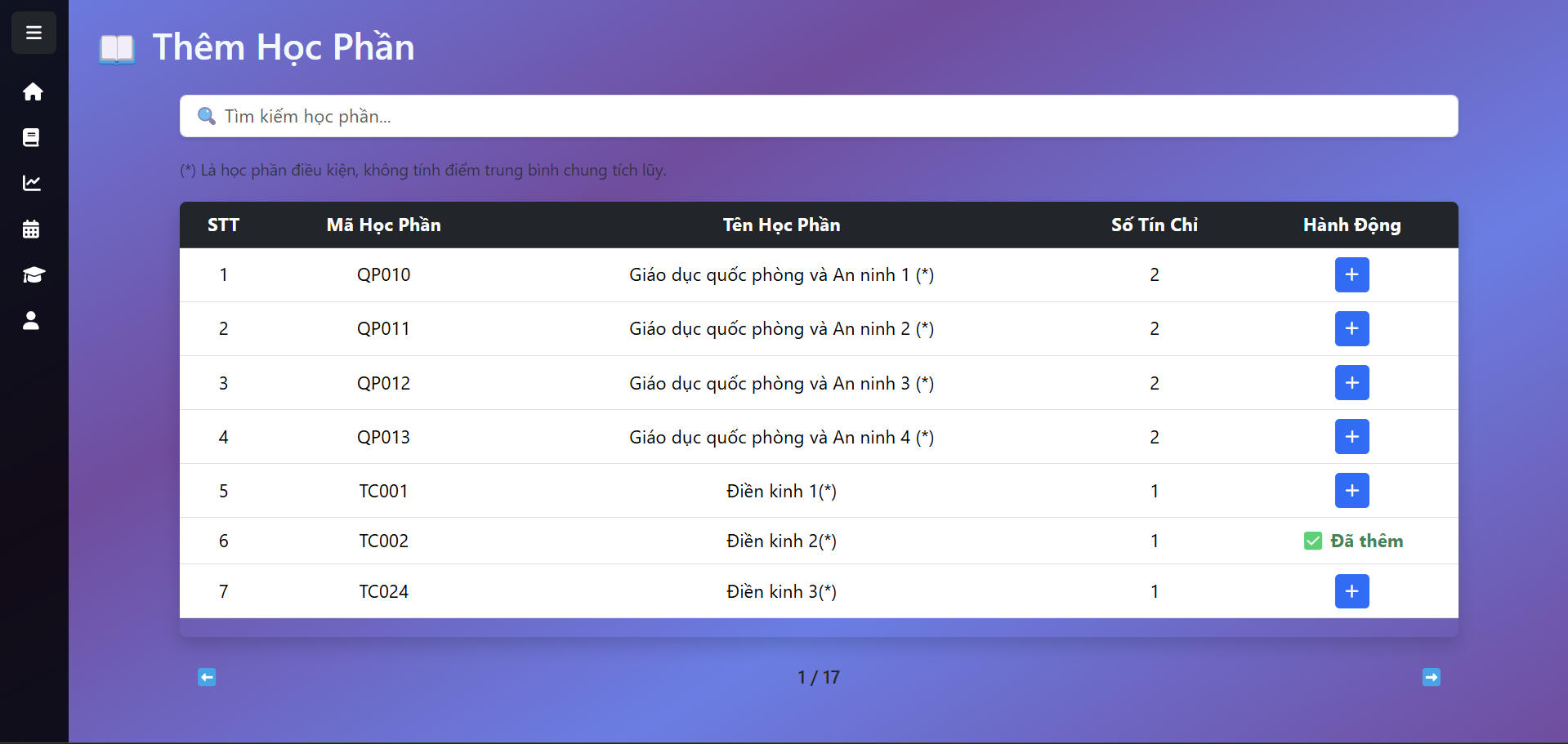
## **4.1. Kết luận:**

Qua quá trình tìm hiểu, phân tích và xây dựng hệ thống quản lý tiến độ học tập cho sinh viên, đề tài đã đạt được những kết quả nhất định, đáp ứng được các yêu cầu cơ bản đề ra ban đầu. Hệ thống đã cho phép người dùng dễ dàng:

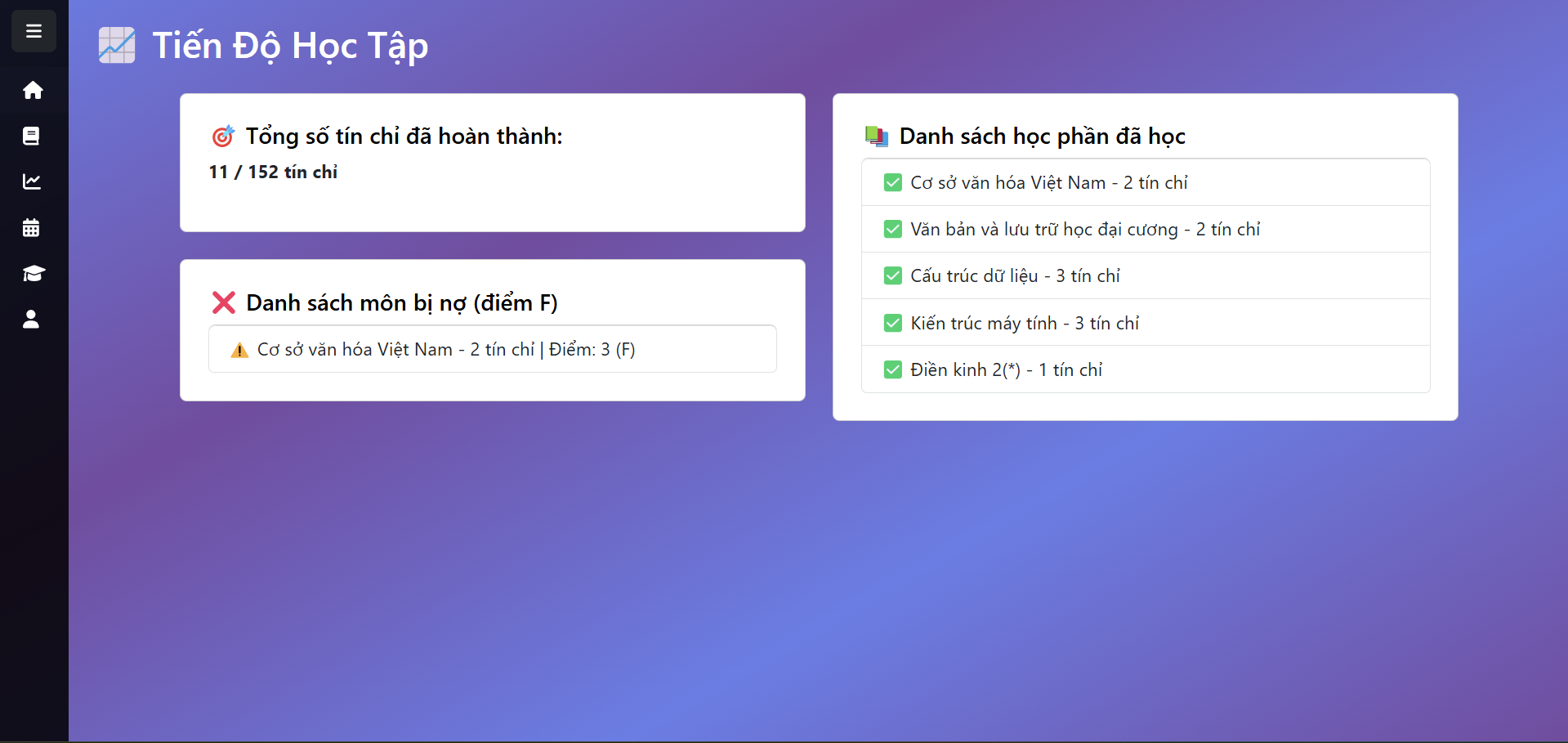
* Theo dõi thông tin cá nhân và kết quả học tập theo từng học kỳ.
* Quản lý danh sách học phần, lịch học và điểm số một cách khoa học.
* Thống kê được tiến độ học tập và cảnh báo những môn học chưa hoàn thành hoặc có kết quả chưa đạt yêu cầu.
* Giao diện thân thiện, dễ sử dụng.

Bên cạnh đó, hệ thống còn góp phần hiện đại hóa công tác quản lý học tập trong môi trường giáo dục, hỗ trợ sinh viên định hướng kế hoạch học tập phù hợp.

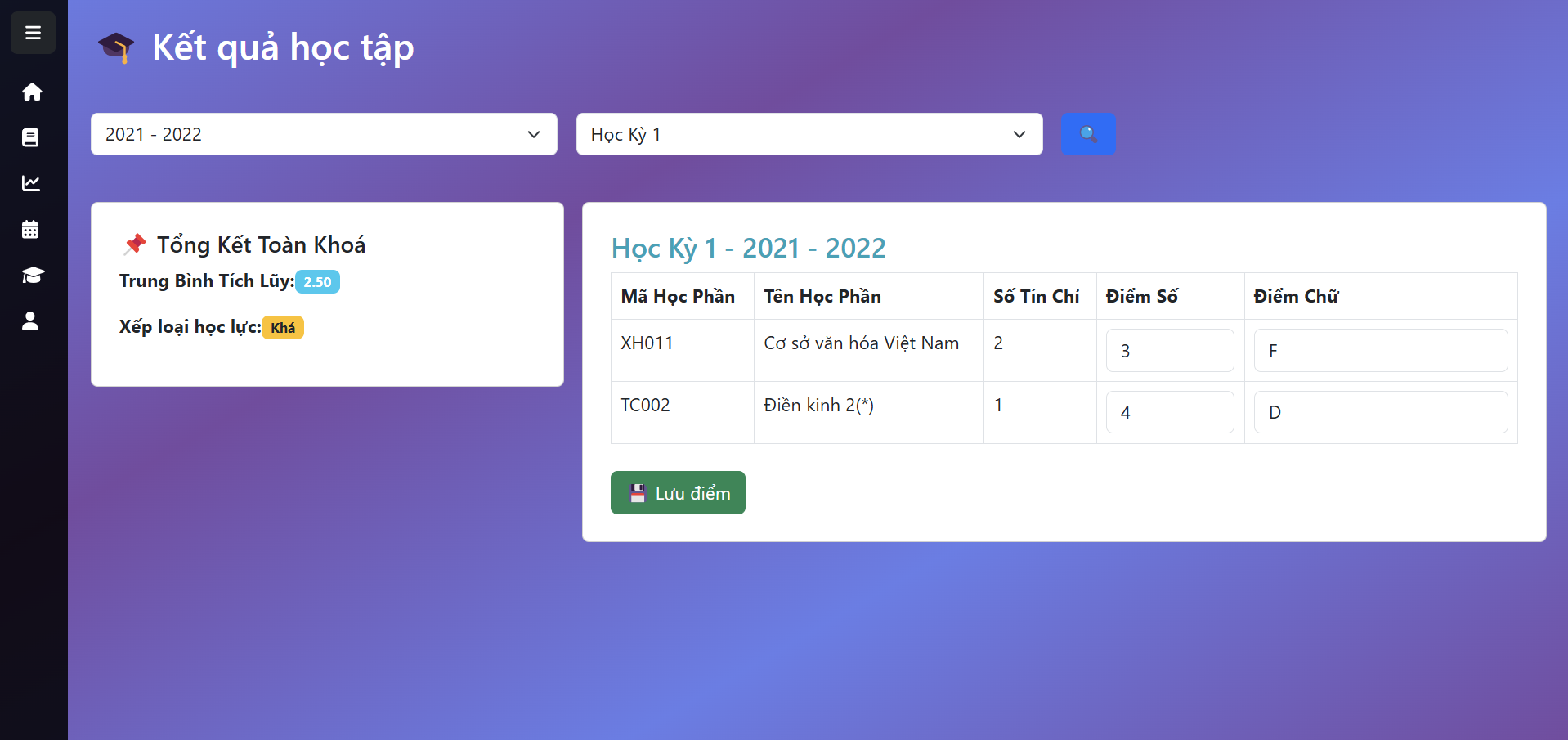
## **4.2. Một số giao diện của hệ thống:**



*Hình 12**: Giao diện “Thêm học phần” vào Score.*



*Hình 13**: Giao diện “Tiến độ học tập”.*



*Hình 14**: Giao diện “Kết quả học tập”.*

## **4.3. Đề nghị và hướng phát triển:**

Từ kết quả nghiên cứu và triển khai hệ thống, thực hiện đề xuất một số hướng phát triển và cải tiến trong tương lai như sau:

* **Nâng cấp chức năng phân tích dữ liệu học tập**: Tích hợp các công cụ phân tích để đưa ra nhận định, dự báo về khả năng tốt nghiệp, gợi ý lộ trình học tập cá nhân hóa.
* **Mở rộng hệ thống theo hướng học tập kết nối (LMS)**: Kết nối với các hệ thống học trực tuyến, tích hợp chấm điểm tự động, ngân hàng câu hỏi, diễn đàn học thuật...
* **Tăng cường tính bảo mật và phân quyền người dùng**: Đảm bảo tính riêng tư và an toàn dữ liệu, cho phép phân quyền chi tiết hơn giữa sinh viên, giảng viên và quản trị viên.
* **Tối ưu giao diện và hiệu năng hệ thống**: Cải thiện trải nghiệm người dùng, nâng cao tốc độ xử lý và khả năng mở rộng cho nhiều người dùng đồng thời.

Hệ thống bước đầu đã chứng minh được tính khả thi và hiệu quả trong môi trường học đường. Tuy nhiên, để phát triển mạnh mẽ và bền vững, rất cần sự phối hợp chặt chẽ giữa người phát triển và các bên liên quan như phòng đào tạo, sinh viên, và giảng viên trong việc góp ý và hoàn thiện sản phẩm.

# Phụ lục

## **Phụ lục hình ảnh:**

*[Hình](#_Toc1777)* [1](#_Toc1777) *[: Cơ sở dữ liệu MongoDB.](#_Toc1777)* [18](#_Toc1777)

*[Hình](#_Toc13314)* [2](#_Toc13314) *[: Minh họa mô hình MVC (Model - View - Controller) [2]](#_Toc13314)* [22](#_Toc13314)

*[Hình](#_Toc24203)* [3](#_Toc24203) *[: Sơ đồ triển khai kỹ thuật của hệ thống.](#_Toc24203)* [24](#_Toc24203)

*[Hình](#_Toc11340)* [4](#_Toc11340) *[: Mô hình phân rã chức năng của hệ thống.](#_Toc11340)* [30](#_Toc11340)

*[Hình](#_Toc21728)* [5](#_Toc21728) *[: Mô hình hoạt động nghiệp vụ tổng quát của hệ thống.](#_Toc21728)* [31](#_Toc21728)

*[Hình](#_Toc28610)* [6](#_Toc28610) *[: Mô hình hoạt động nghiệp vụ của Đăng Nhập/Đăng Ký.](#_Toc28610)* [32](#_Toc28610)

*[Hình](#_Toc11119)* [7](#_Toc11119) *[: Mô hình hoạt động nghiệp vụ của Account.](#_Toc11119)* [33](#_Toc11119)

[Hình 8 :](#_Toc32393) *[Mô hình hoạt động nghiệp vụ của Course.](#_Toc32393)* [34](#_Toc32393)

*[Hình](#_Toc11052)* [9](#_Toc11052) *[: Mô hình thực thể quản hệ (dữ liệu) của hệ thống.](#_Toc11052)* [36](#_Toc11052)

*[Hình](#_Toc28117)* [10](#_Toc28117) *[: Mô hình tổng quan hệ thống.](#_Toc28117)* [36](#_Toc28117)

*[Hình](#_Toc15492)* [11](#_Toc15492) *[: Mô hình cài đặt hệ thống.](#_Toc15492)* [37](#_Toc15492)

*[Hình](#_Toc24585)* [12](#_Toc24585) *[: Giao diện “Thêm học phần” vào Score.](#_Toc24585)* [38](#_Toc24585)

*[Hình](#_Toc16539)* [13](#_Toc16539) *[: Giao diện “Tiến độ học tập”.](#_Toc16539)* [39](#_Toc16539)

*[Hình](#_Toc18464)* [14](#_Toc18464) *[: Giao diện “Kết quả học tập”.](#_Toc18464)* [39](#_Toc18464)

## **Phụ lục bảng:**

[Bảng 1 : Bảng tóm tắt ngôn ngữ lập trình của hệ thống. 26](#_Toc28324)

*[Bảng](#_Toc6439)* [2](#_Toc6439) *[: Bảng chuyển đổi điểm số sang thang điểm 4](#_Toc6439)* [27](#_Toc6439)

*[Bảng](#_Toc12610)* [3](#_Toc12610) *[: Bảng tính điểm trung bình tích lũy.](#_Toc12610)* [28](#_Toc12610)

*[Bảng](#_Toc12304)* [4](#_Toc12304) *[: Bảng mô tả ngắn gọn chức năng của hệ thống.](#_Toc12304)* [31](#_Toc12304)

*[Bảng](#_Toc16040)* [5](#_Toc16040) *[:Các thành phần cài đặt của hệ thống.](#_Toc16040)* [37](#_Toc16040)

# Tài liệu tham khảo.

1. [1] <https://oes.vn/su-phat-trien-cua-he-thong-quan-ly-hoc-tap/>
2. [2]<https://studycodingblog.wordpress.com/2016/11/30/viet-web-bang-mvc-5-la-gi-va-tai-sao-phai-dung-no/>
3. [3] [https://topdev.vn/blog/mongodb-la-gi/](3.https:/topdev.vn/blog/mongodb-la-gi/)
4. [4][https://moodle.org/mod/glossary/showentry.php?eid=7922&displayformat=dictionary](4.https:/moodle.org/mod/glossary/showentry.php?eid=7922&displayformat=dictionary)
5. [5] <https://classin.vn/top-he-thong-lms-tot-nhat-hien-nay/>
6. [6] <https://www.instructure.com/en-au/lms-learning-management-system>
7. [7]<https://dsa.ctu.edu.vn/images/upload/vbanply/2021/Vanban/QD1813_QD_ban_hanh_Quy_dinh_cong_tac_hoc_vu_2021.pdf>