

# CHƯƠNG 1: TỔNG QUAN

## 1.1. LÝ DO CHỌN ĐỀ TÀI

- Hiện nay, với sự phát triển của giáo dục đại học và sự gia tăng nhanh chóng về số lượng sinh viên, việc quản lý thông tin sinh viên trở thành một thách thức lớn đối với các trường đại học. Các phương pháp quản lý truyền thống như sử dụng giấy tờ hoặc bảng tính Excel không còn phù hợp, vì:
  - Dễ xảy ra sai sót khi nhập liệu, thống kê hoặc tra cứu dữ liệu.
  - Khó khăn trong việc quản lý số lượng lớn sinh viên, lớp học và môn học.
  - Hạn chế khả năng tích hợp và mở rộng hệ thống.
- Một hệ thống quản lý sinh viên hiện đại có thể:
  - Đơn giản hóa quy trình quản lý thông tin.
  - Hỗ trợ tra cứu nhanh chóng, giảm thiểu sai sót.
  - Tăng cường khả năng phân tích và thống kê dữ liệu về tình hình học tập của sinh viên.
  - Tích hợp công nghệ thông tin vào quản lý giáo dục, góp phần hiện đại hóa nền giáo dục.
- Với những lý do trên, việc xây dựng một hệ thống quản lý sinh viên hiệu quả và dễ sử dụng là một nhu cầu cấp thiết và cũng là mục tiêu chính của đề tài này.

## 1.2. XÁC ĐỊNH ĐỀ TÀI

### 1.2.1. Tên đề tài

#### Hệ thống quản lý sinh viên cấp đại học.

### 1.2.2. Mục tiêu của đề tài

- Đề tài hướng đến các mục tiêu cụ thể sau:
  - Xây dựng một hệ thống quản lý toàn diện về thông tin sinh viên, lớp học, môn học và điểm số.
  - Hỗ trợ các chức năng thêm, sửa, xóa, tìm kiếm thông tin dễ dàng và chính xác.
  - Cung cấp báo cáo, thống kê về kết quả học tập và phân loại sinh viên.
  - Đảm bảo tính linh hoạt, dễ mở rộng và bảo mật thông tin.

### 1.2.3. Phạm vi hệ thống

- Hệ thống tập trung vào các chức năng chính:
  - Quản lý sinh viên:**
    - Lưu trữ thông tin cá nhân như mã sinh viên, họ tên, ngày sinh, giới tính, địa chỉ, số điện thoại.
  - Quản lý lớp học:**
    - Theo dõi thông tin lớp học, danh sách sinh viên thuộc từng lớp.
  - Quản lý môn học:**
    - Quản lý danh sách các môn học, số tín chỉ, thông tin giảng viên phụ trách.
  - Quản lý điểm số:**
    - Lưu trữ điểm số từng môn học, tính toán điểm trung bình, xếp loại học lực.

#### 1.2.4. Phạm vi nghiên cứu

- Đề tài tập trung nghiên cứu và triển khai các nội dung sau:
    - Phân tích nghiệp vụ và thiết kế hệ thống quản lý sinh viên cấp đại học.
    - Ứng dụng phương pháp phân tích thiết kế hướng đối tượng (OOAD) với công cụ UML.
    - Sử dụng cơ sở dữ liệu MySQL để lưu trữ và quản lý thông tin.
    - Xây dựng giao diện thân thiện và dễ sử dụng cho người dùng.
- 

### 1.3. CẤU TRÚC BÁO CÁO

- Báo cáo được chia thành các chương sau:
    - Chương 1: Tổng quan**
      - Trình bày lý do chọn đề tài, mục tiêu, phạm vi nghiên cứu, và cấu trúc báo cáo.
    - Chương 2: Kiến thức cơ sở**
      - Trình bày các lý thuyết liên quan đến cơ sở dữ liệu, phân tích thiết kế hệ thống và ngôn ngữ mô hình hóa UML.
    - Chương 3: Nội dung thực hiện**
      - Mô tả chi tiết quá trình khảo sát, phân tích nghiệp vụ, thiết kế hệ thống và xây dựng phần mềm.
    - Chương 4: Kết quả nghiên cứu, hướng phát triển và kết luận**
      - Tổng kết kết quả đạt được, đánh giá ưu nhược điểm và đề xuất cải tiến hệ thống trong tương lai.
- 

### 1.4. PHƯƠNG PHÁP NGHIÊN CỨU

- Để thực hiện đề tài, các phương pháp nghiên cứu sau được áp dụng:

#### 1.4.1. Phương pháp khảo sát thực tế

- Tìm hiểu thực trạng và các vấn đề trong quản lý sinh viên tại các trường đại học.
- Phân tích nhu cầu của người dùng như nhân viên quản lý, giảng viên và sinh viên.

#### 1.4.2. Phương pháp phân tích hệ thống

- Ứng dụng phương pháp phân tích hướng đối tượng để xây dựng sơ đồ Use Case, Class Diagram, Sequence Diagram, và Activity Diagram.
- Phân tích quy trình quản lý sinh viên hiện tại và đề xuất các cải tiến.

#### 1.4.3. Phương pháp thực nghiệm

- Lập trình và triển khai hệ thống trên nền tảng công nghệ phù hợp.
- Chạy thử nghiệm hệ thống với các tình huống thực tế để đánh giá hiệu quả.

#### 1.4.4. Phương pháp tài liệu

- Nghiên cứu các tài liệu về cơ sở dữ liệu, ngôn ngữ lập trình và mô hình UML.
  - Tìm hiểu các hệ thống quản lý sinh viên hiện có để rút ra bài học kinh nghiệm và ý tưởng cải tiến.
-

## CHƯƠNG 2: KIẾN THỨC CƠ SỞ

### 2.1. LÝ THUYẾT CƠ SỞ DỮ LIỆU

#### 2.1.1. Định nghĩa

- Cơ sở dữ liệu (Database) là tập hợp có tổ chức các dữ liệu, được lưu trữ và quản lý để dễ dàng truy xuất, sửa đổi, phân tích và chia sẻ. Trong hệ thống quản lý, cơ sở dữ liệu là một thành phần cốt lõi giúp lưu trữ các thông tin liên quan một cách khoa học và logic.

#### 2.1.2. Các thành phần chính

##### 1. Thực thể (Entity):

Là đối tượng cụ thể hoặc trừu tượng mà ta cần lưu trữ thông tin. Ví dụ, trong hệ thống quản lý sinh viên, các thực thể bao gồm Sinh viên, Lớp học, Môn học, Điểm.

##### 2. Thuộc tính (Attribute):

Là các đặc điểm hoặc thông tin cụ thể của thực thể. Ví dụ, thuộc tính của thực thể Sinh viên có thể là họ tên, mã sinh viên, ngày sinh, và giới tính.

##### 3. Mối quan hệ (Relationship):

Là sự liên kết giữa các thực thể. Ví dụ, mỗi sinh viên sẽ thuộc một lớp học và học nhiều môn học.

#### 2.1.3. Ưu điểm của cơ sở dữ liệu

- Giảm trùng lặp thông tin:** Dữ liệu được tổ chức logic, tránh dư thừa.
- Đảm bảo toàn vẹn dữ liệu:** Các quy tắc ràng buộc giúp dữ liệu luôn chính xác và hợp lệ.
- Dễ dàng truy xuất:** Cơ sở dữ liệu cung cấp công cụ tìm kiếm, truy vấn nhanh chóng.
- Tính bảo mật:** Có thể phân quyền người dùng và bảo vệ dữ liệu khỏi truy cập trái phép.

#### 2.1.4. Nhược điểm của cơ sở dữ liệu

- Phức tạp trong thiết kế:** Đòi hỏi kỹ năng chuyên sâu khi xây dựng mô hình cơ sở dữ liệu.
- Chi phí đầu tư:** Cần các hệ quản trị cơ sở dữ liệu và phần cứng phù hợp.
- Khả năng lỗi hệ thống:** Rủi ro mất mát dữ liệu khi gặp sự cố nếu không có sao lưu.

#### 2.1.5. Phân loại cơ sở dữ liệu

##### 1. Cơ sở dữ liệu quan hệ (Relational Database):

Lưu trữ dữ liệu dưới dạng bảng, mỗi bảng tương ứng với một thực thể. Ví dụ: MySQL, PostgreSQL.

##### 2. Cơ sở dữ liệu phi quan hệ (NoSQL):

Dữ liệu được tổ chức theo dạng key-value, document, hoặc graph. Ví dụ: MongoDB, Firebase.

##### 3. Cơ sở dữ liệu phân tán:

Dữ liệu được lưu trữ và xử lý trên nhiều máy chủ. Ví dụ: Apache Cassandra.

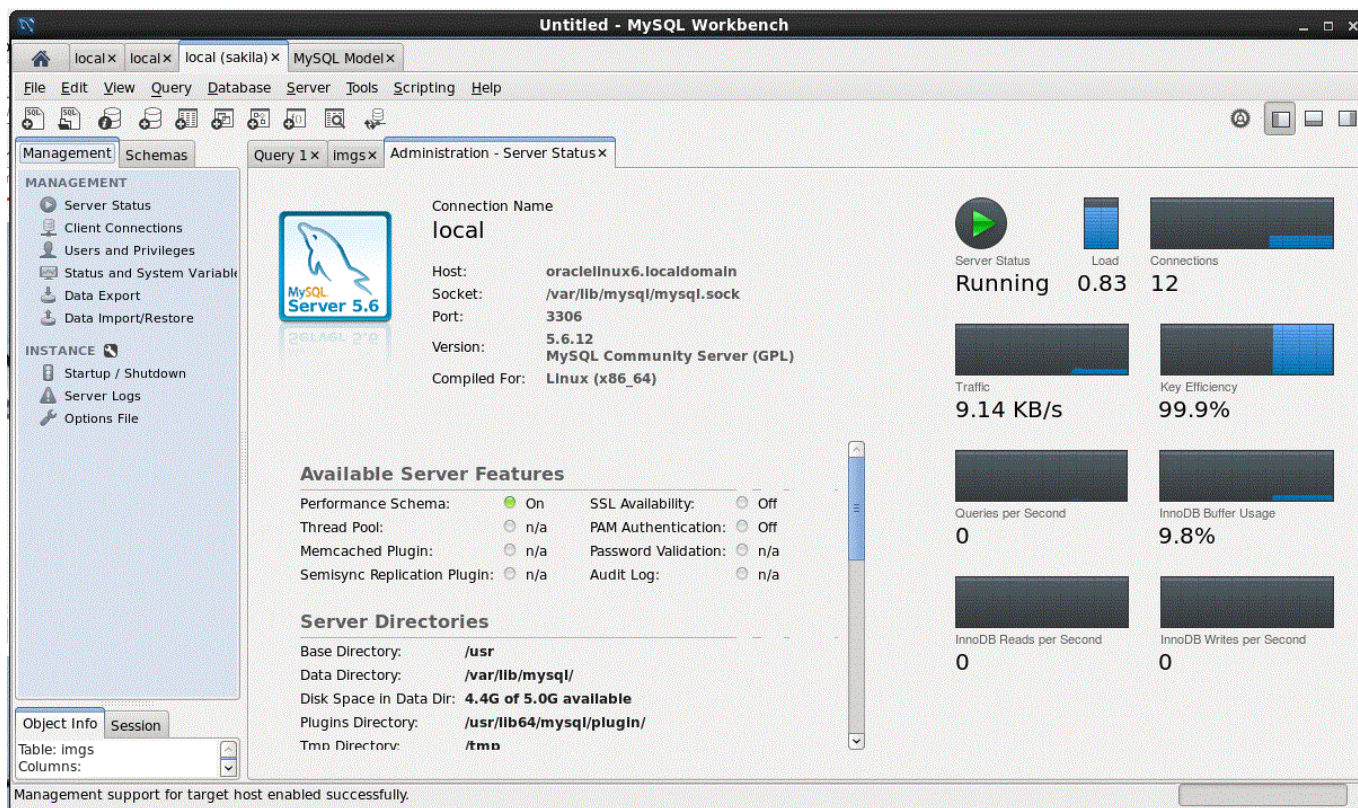
##### 4. Cơ sở dữ liệu nhúng:

Được tích hợp trực tiếp vào ứng dụng. Ví dụ: SQLite.

### 2.2. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

### 2.2.1. Giới thiệu

- MySQL là một hệ quản trị cơ sở dữ liệu mã nguồn mở, sử dụng mô hình quan hệ (RDBMS). Nó cho phép lưu trữ, truy vấn và quản lý dữ liệu một cách hiệu quả. MySQL là một trong những công cụ được sử dụng phổ biến nhất, đặc biệt trong các hệ thống web.



### 2.2.2. Tính năng chính của MySQL

- Hỗ trợ ngôn ngữ SQL:** Ngôn ngữ tiêu chuẩn để thao tác dữ liệu.
- Tương thích đa nền tảng:** Hoạt động tốt trên Windows, Linux, macOS.
- Bảo mật cao:** Hỗ trợ phân quyền người dùng, mã hóa dữ liệu.
- Hiệu năng tốt:** Xử lý khối lượng dữ liệu lớn với tốc độ cao.

### 2.2.3. Ưu điểm của MySQL

- Dễ sử dụng:** Giao diện thân thiện, cú pháp SQL đơn giản.
- Hiệu suất cao:** Tối ưu hóa cho các ứng dụng web và dữ liệu lớn.
- Miễn phí:** Phiên bản cộng đồng cung cấp đầy đủ tính năng cơ bản.

### 2.2.4. Nhược điểm của MySQL

- Hạn chế tính năng cao cấp:** Không mạnh mẽ bằng Oracle hoặc SQL Server trong các tác vụ phức tạp.
- Phụ thuộc vào cấu hình:** Bảo mật và hiệu năng phụ thuộc vào khả năng cấu hình của người dùng.

## 2.3. PHÂN TÍCH THIẾT KẾ HƯỚNG ĐỐI TƯỢNG – UML

### 2.3.1. Khái niệm OOAD

- Phân tích và thiết kế hướng đối tượng (Object-Oriented Analysis and Design - OOAD) là phương pháp tiếp cận sử dụng các đối tượng để phân tích và thiết kế hệ thống. OOAD giúp biểu diễn các thành phần hệ thống theo cách gần gũi với thế giới thực, dễ dàng quản lý và mở rộng.

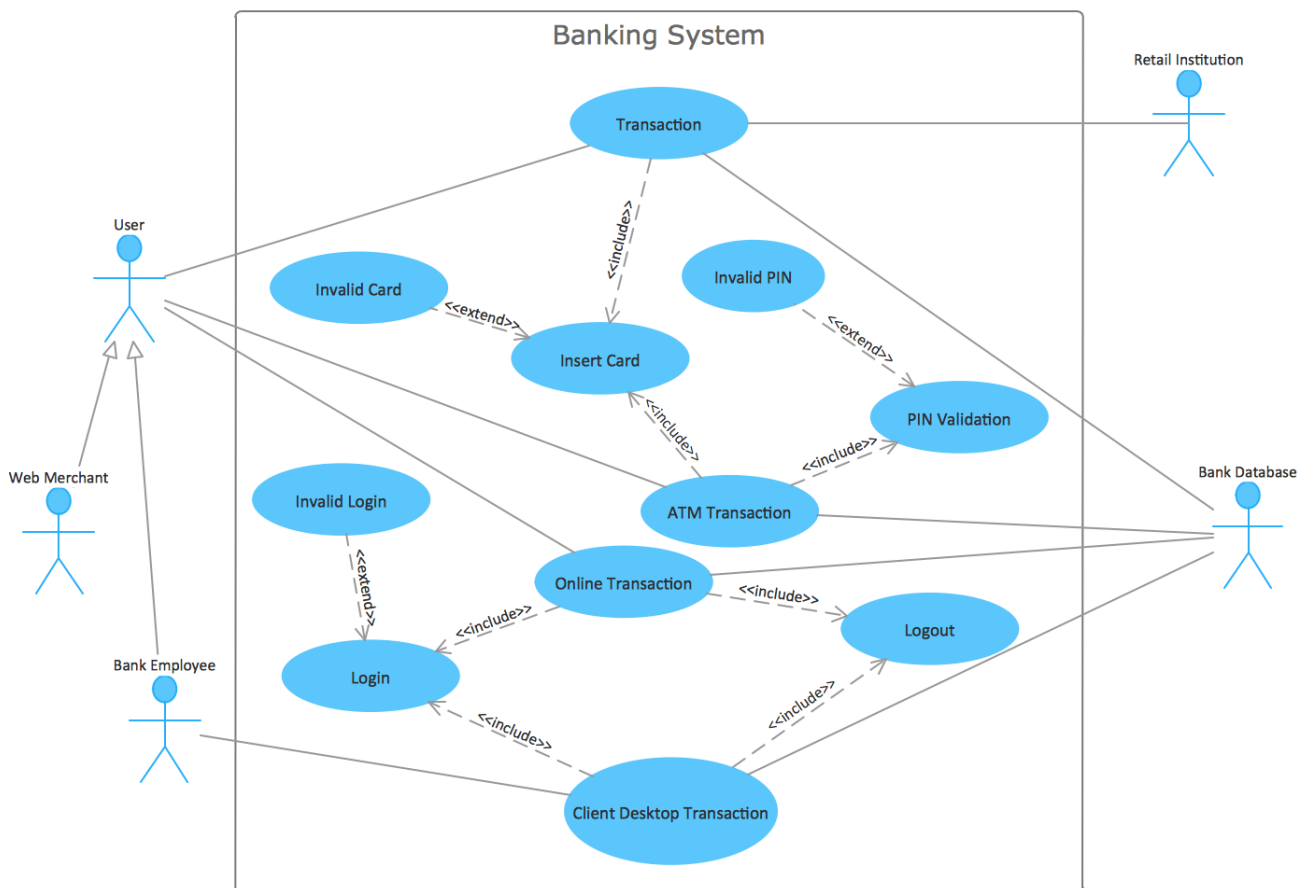
### 2.3.2. Ngôn ngữ UML

- UML (Unified Modeling Language)** là một ngôn ngữ mô hình hóa tiêu chuẩn, được sử dụng để trực quan hóa và tài liệu hóa các hệ thống phần mềm. UML cung cấp các sơ đồ giúp phân tích và thiết kế hệ thống dễ dàng hơn.

### 2.3.3. Các sơ đồ UML chính

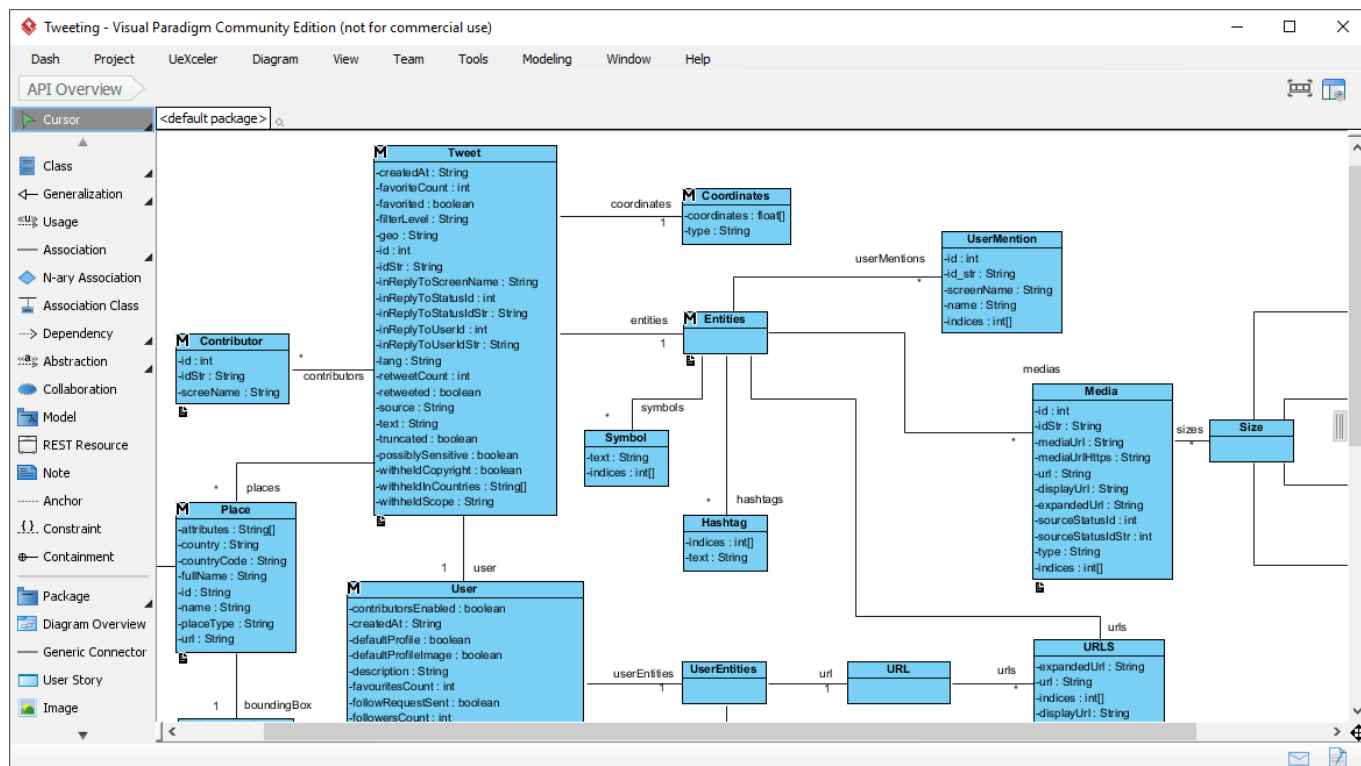
#### 1. Use Case Diagram:

- Biểu diễn các chức năng chính của hệ thống và mối quan hệ với người dùng.
- Ví dụ: Chức năng "Quản lý sinh viên" bao gồm thêm, sửa, xóa sinh viên.



#### 2. Class Diagram:

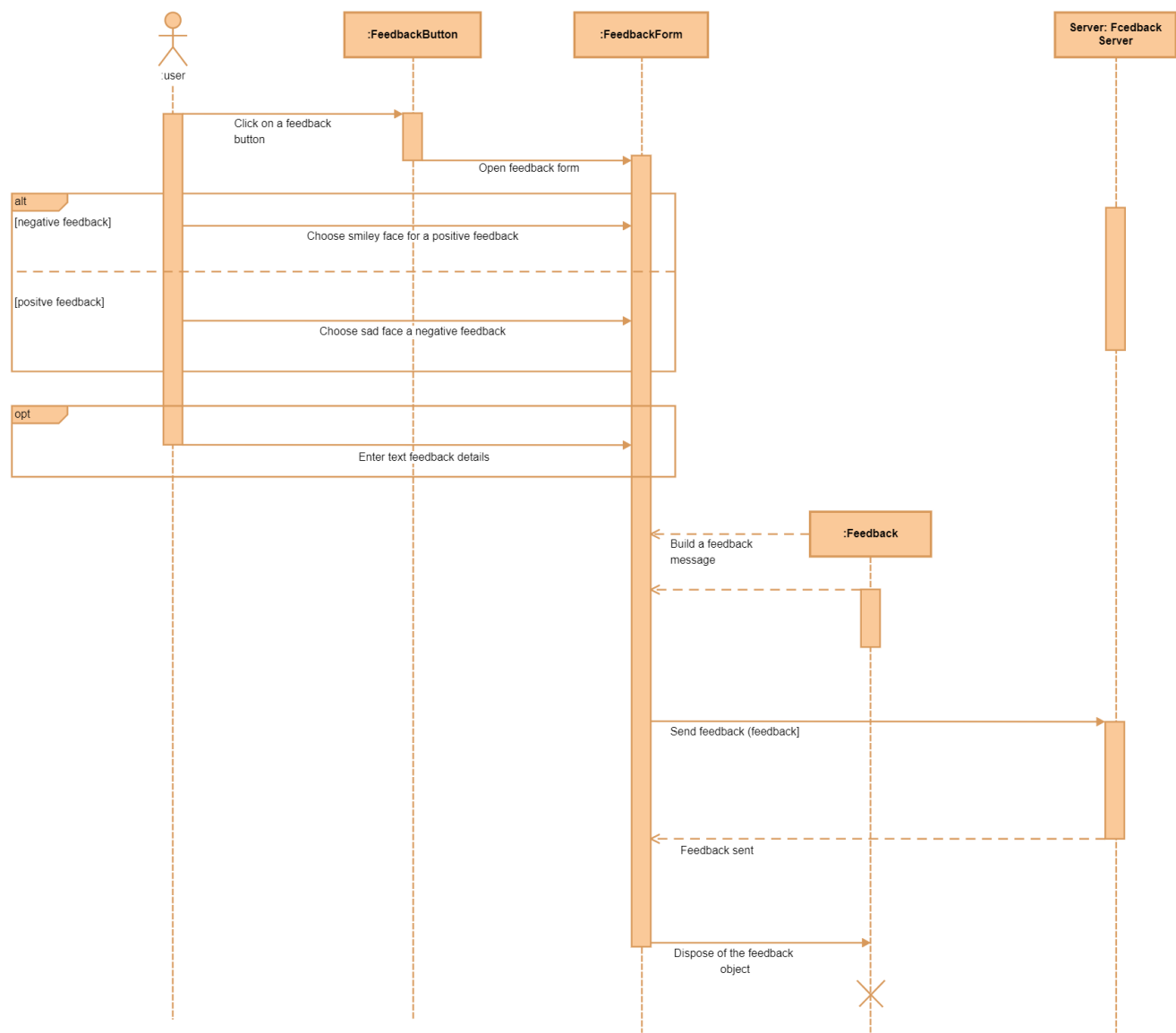
- Biểu diễn các lớp, thuộc tính, phương thức và mối quan hệ giữa các lớp.
- Ví dụ: Lớp Sinh viên có các thuộc tính (họ tên, mã số) và phương thức (đăng ký môn học).



### 3. Sequence Diagram:

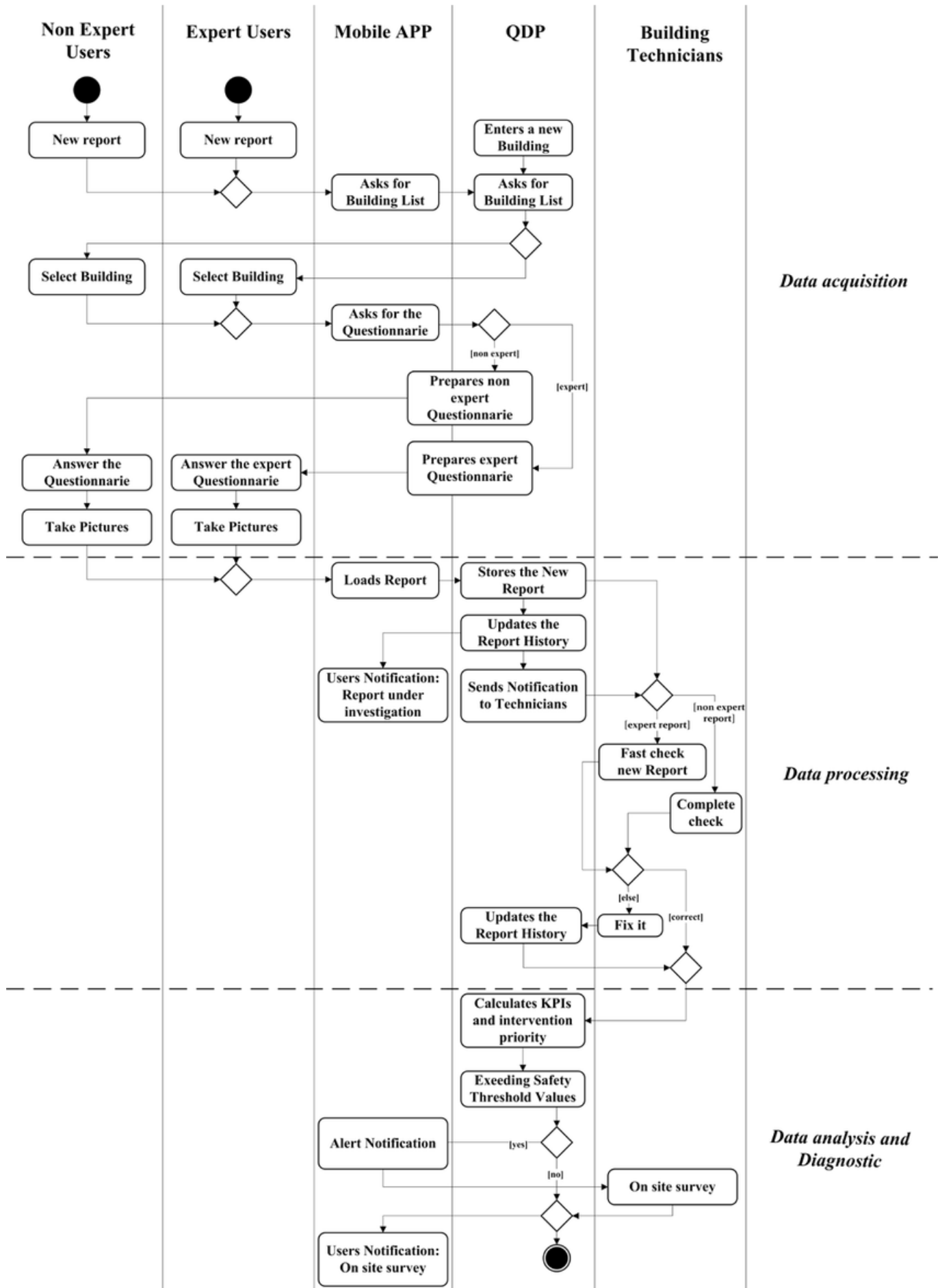
- Minh họa trình tự các tương tác giữa các đối tượng trong một chức năng cụ thể.
- Ví dụ: Quá trình đăng ký môn học của sinh viên.





4. Activity Diagram:

- Biểu diễn các luồng hoạt động hoặc quy trình nghiệp vụ.
- Ví dụ: Quy trình quản lý điểm thi.



## 2.4. MỤC ĐÍCH VÀ YÊU CẦU PHÂN TÍCH HỆ THỐNG



### 2.4.1. Mục đích

- **Tăng hiệu quả quản lý:** Hệ thống giúp tối ưu hóa việc lưu trữ và xử lý dữ liệu sinh viên.
- **Đảm bảo tính logic và chính xác:** Dữ liệu được tổ chức chặt chẽ, tránh sai sót.
- **Hỗ trợ mở rộng và bảo trì:** Thiết kế hệ thống linh hoạt, dễ dàng nâng cấp trong tương lai.

### 2.4.2. Yêu cầu

#### 1. Tính chính xác:

- Đảm bảo dữ liệu không trùng lặp, sai lệch.
- Xử lý chính xác các nghiệp vụ như thêm, sửa, xóa sinh viên.

#### 2. Tính thân thiện:

- Giao diện trực quan, dễ sử dụng cho người quản trị và sinh viên.

#### 3. Tính bảo mật:

- Phân quyền truy cập, đảm bảo dữ liệu an toàn khỏi truy cập trái phép.

#### 4. Tính hiệu quả:

- Hệ thống có khả năng xử lý nhanh chóng với lượng lớn dữ liệu.

## 2.5. Công cụ sử dụng

- Công cụ tiếp thông tin : **GOOGLE**.
- Công cụ uml: **plantuml**

### 2.5.1. PlantUML

- PlantUML là một công cụ hỗ trợ vẽ các loại biểu đồ UML, hoạt động dựa trên việc mô tả cấu trúc của biểu đồ bằng mã nguồn văn bản. Thay vì sử dụng giao diện đồ họa, người dùng viết mã để định nghĩa các thành phần và mối quan hệ giữa chúng.

### 2.5.2. Lý do mình dùng Plantuml

- Dễ học và sử dụng: Chỉ cần hiểu cú pháp cơ bản, bạn có thể tạo ra các biểu đồ chuyên nghiệp.
- Hỗ trợ nhiều loại biểu đồ: Bao gồm sơ đồ lớp (Class Diagram), sơ đồ trình tự (Sequence Diagram), sơ đồ trạng thái (State Diagram), sơ đồ hoạt động (Activity Diagram), và nhiều loại biểu đồ khác.
- Tích hợp tốt: Có thể tích hợp với các IDE như IntelliJ, Eclipse, VS Code hoặc các công cụ như Doxygen, Markdown.
- Mã nguồn có thể dễ dàng chia sẻ và chỉnh sửa: Vì biểu đồ được định nghĩa bằng văn bản, bạn có thể lưu trữ mã trong Git hoặc các hệ thống kiểm soát phiên bản khác.
- Miễn phí và mã nguồn mở.

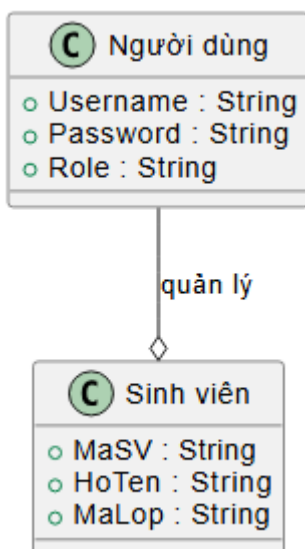
### 2.5.3. Các loại biểu đồ được hỗ trợ

- PlantUML hỗ trợ tạo nhiều loại biểu đồ UML và phi UML, bao gồm:
  - Biểu đồ UML:

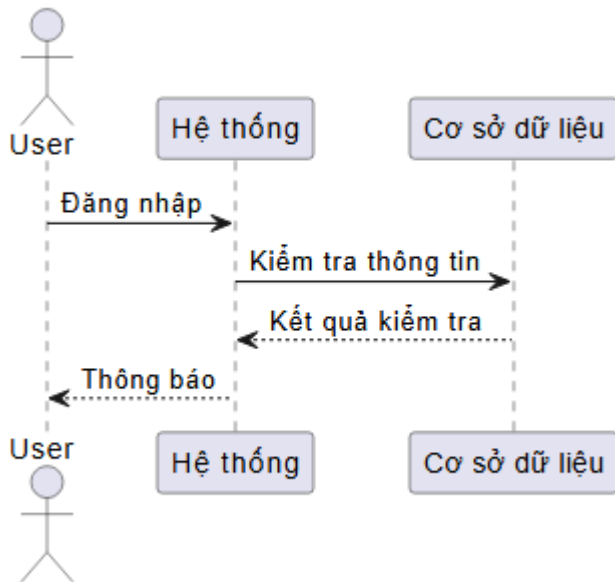
- Class Diagram (Sơ đồ lớp)
- Sequence Diagram (Sơ đồ trình tự)
- Activity Diagram (Sơ đồ hoạt động)
- State Diagram (Sơ đồ trạng thái)
- Use Case Diagram (Sơ đồ trường hợp sử dụng)
- Component Diagram (Sơ đồ thành phần)
- Deployment Diagram (Sơ đồ triển khai)
- Biểu đồ phi UML:
  - Biểu đồ Gantt
  - Biểu đồ WBS (Work Breakdown Structure)
  - Biểu đồ dòng chảy dữ liệu (Data Flow Diagram)
  - Wireframe Diagram (Biểu đồ giao diện người dùng)

#### 2.5.4. Cú pháp cơ bản

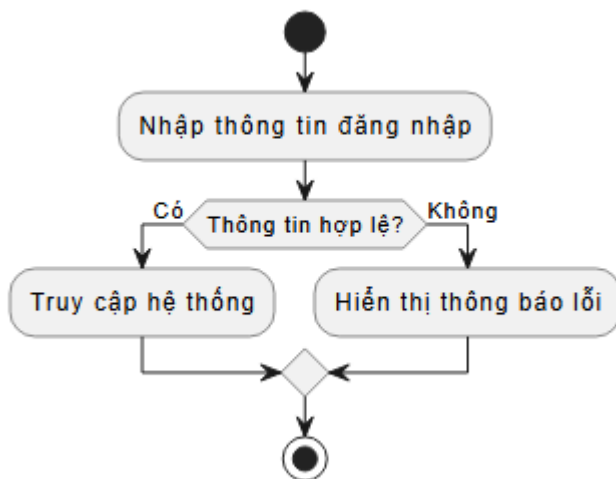
- PlantUML sử dụng một ngôn ngữ có cú pháp đơn giản. Dưới đây là một số ví dụ:
- Ví dụ 1: Sơ đồ lớp



- Ví dụ 2: Sơ đồ trình tự



- Ví dụ 3: Sơ đồ hoạt động



### 2.5.5. Công cụ hỗ trợ

- Bạn có thể sử dụng PlantUML trên các nền tảng và công cụ sau:
  - Trực tuyến:
    - **PlantText**
    - **PlantUML Editor**
  - Ngoại tuyến:
    - **Tích hợp IDE:** Sử dụng plugin PlantUML cho IntelliJ, Eclipse, hoặc VS Code.
    - **Command Line:** Cài đặt PlantUML trên máy tính và chạy bằng dòng lệnh.
    - **Với Graphviz:** Để hỗ trợ kết xuất đồ họa phức tạp hơn.

### 2.5.6. Ưu điểm và nhược điểm

- Ưu điểm:
  - Dễ dàng tích hợp với nhiều công cụ và quy trình phát triển phần mềm.
  - Hỗ trợ nhiều loại biểu đồ và cú pháp dễ đọc.

- Nhẹ, không yêu cầu giao diện đồ họa phức tạp.
- Hoàn toàn miễn phí và có cộng đồng hỗ trợ lớn.
- Nhược điểm:
  - Cần phải học cú pháp nếu bạn chưa quen với mã hóa.
  - Các biểu đồ phức tạp có thể trở nên khó đọc trong phần mã nguồn.

### 2.5.7. Cài đặt

- Yêu cầu:
  - Java Runtime Environment (JRE).
  - Graphviz (nếu cần biểu đồ phức tạp).
- Cách cài đặt:
  - Tải PlantUML:
  - Tải file .jar từ trang chính thức.
- Chạy PlantUML:
  - Sử dụng dòng lệnh:

```
java -jar plantuml.jar example.txt
```

- Cài đặt plugin:
  - Tích hợp với IDE như IntelliJ, Eclipse hoặc VS Code bằng cách cài đặt plugin tương ứng.

---

## CHƯƠNG 3: NỘI DUNG THỰC HIỆN

---

### 3.1. KHẢO SÁT THỰC TẾ

#### 3.1.1. Mục tiêu khảo sát

- Nhằm xác định các yêu cầu cần thiết cho hệ thống quản lý sinh viên, chúng tôi đã tiến hành khảo sát thực tế tại một số trường đại học. Mục tiêu chính bao gồm:
  - Hiểu rõ quy trình quản lý sinh viên hiện tại.
  - Xác định các khó khăn, hạn chế trong quản lý.
  - Thu thập yêu cầu từ người dùng cuối (cán bộ quản lý, giảng viên, sinh viên).

#### 3.1.2. Kết quả khảo sát

- Quy trình quản lý sinh viên hiện tại chủ yếu bao gồm các hoạt động:

##### 1. Quản lý thông tin sinh viên:

- Cập nhật thông tin cá nhân: mã sinh viên, họ tên, ngày sinh, địa chỉ.
- Lưu trữ hồ sơ học tập (điểm số, xếp loại, kết quả tốt nghiệp).

##### 2. Quản lý lớp học:

- Phân bổ sinh viên vào các lớp học.

- Theo dõi danh sách sinh viên từng lớp.

### 3. Quản lý môn học:

- Tạo danh sách môn học, số tín chỉ.
- Đăng ký môn học cho sinh viên.

### 4. Quản lý điểm số:

- Ghi nhận và lưu trữ điểm thi từng môn học.
- Tính điểm trung bình, phân loại học lực.

#### Những khó khăn hiện tại:

- Quản lý dữ liệu phân tán, thiếu sự đồng bộ.
- Khó khăn trong việc tìm kiếm, thống kê dữ liệu.
- Dễ xảy ra sai sót khi nhập liệu thủ công.

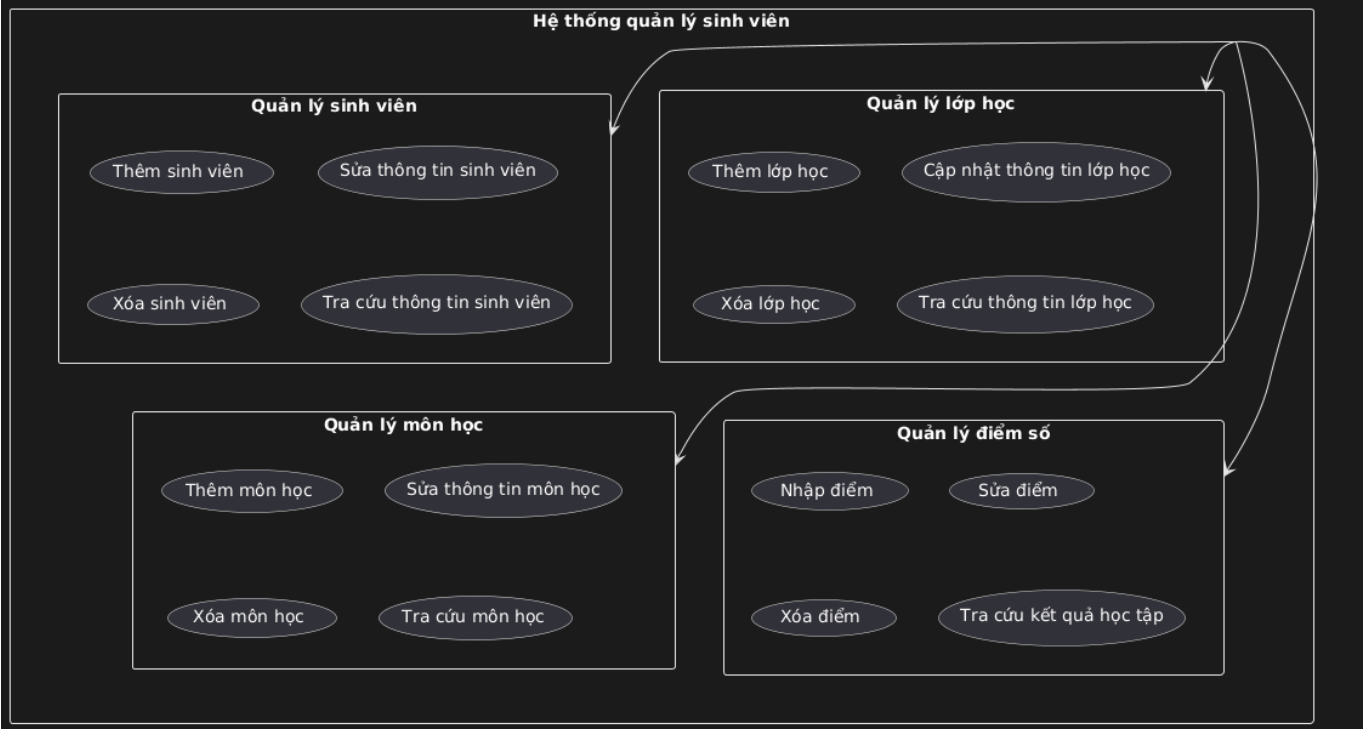
---

## 3.2. MÔ TẢ NGHIỆP VỤ

### 3.2.1. Sơ đồ chức năng tổng quát

- Hệ thống quản lý sinh viên được chia thành 4 chức năng chính:
  1. **Quản lý sinh viên:** Quản lý thông tin cá nhân và hồ sơ học tập của sinh viên.
  2. **Quản lý lớp học:** Tổ chức và theo dõi thông tin các lớp học.
  3. **Quản lý môn học:** Xây dựng và cập nhật thông tin các môn học.
  4. **Quản lý điểm số:** Lưu trữ và xử lý kết quả học tập.

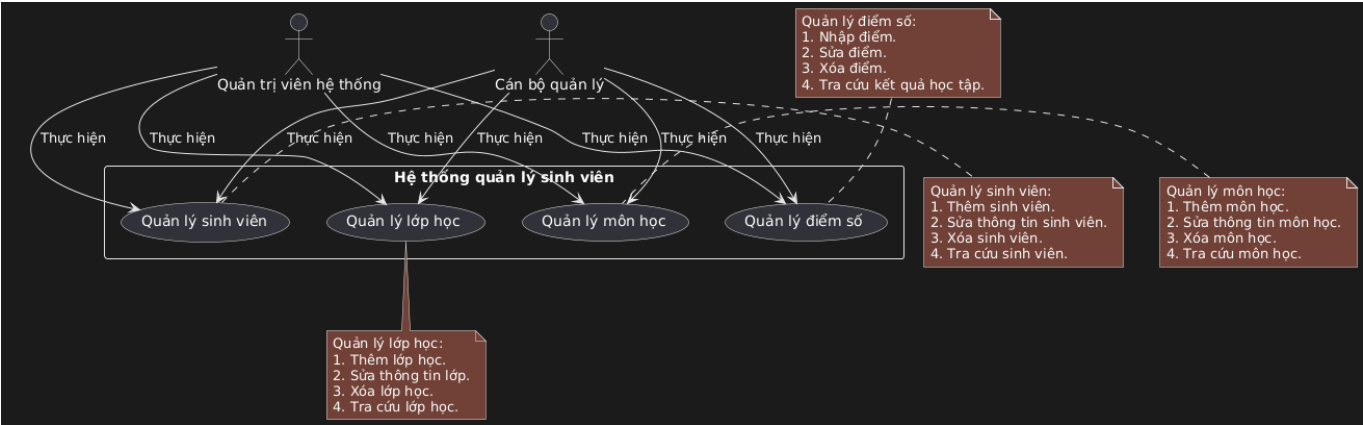




3.3. SƠ ĐỒ USE CASE

3.3.1. Use Case tổng quát

- Use Case tổng quát thể hiện các chức năng chính của hệ thống và mối quan hệ giữa người dùng và hệ thống.



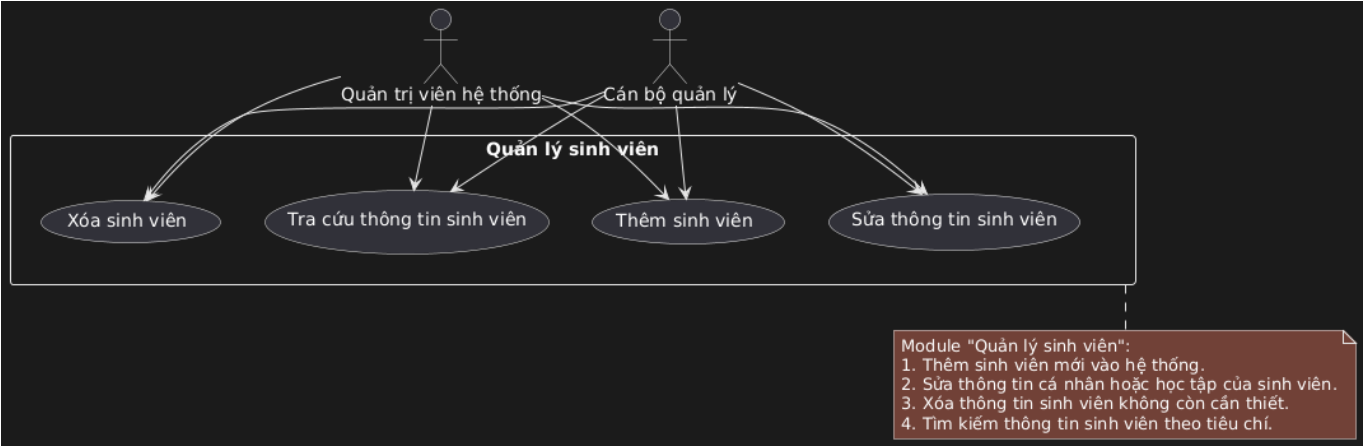
- **Bảng mô tả Use Case tổng quát:**

| Tên Use Case        | Quản lý sinh viên   |
|---------------------|---|
| Mô tả               | Người dùng thực hiện các chức năng liên quan đến quản lý sinh viên.   |
| Tác nhân chính      | Quản trị viên hệ thống, cán bộ quản lý.   |
| Luồng sự kiện chính | 1. Người dùng đăng nhập vào hệ thống.<br>2. Truy cập module quản lý sinh viên.<br>3. Thêm, sửa, xóa hoặc tra cứu thông tin sinh viên. |

3.3.2. Use Case quản lý sinh viên



- Sơ đồ use case:

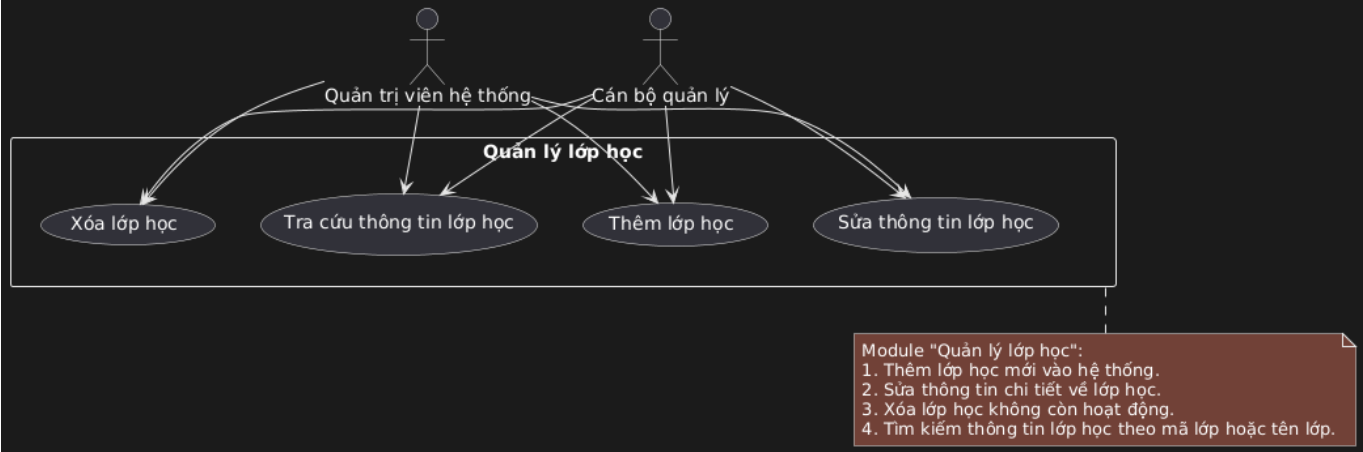


Bảng mô tả chi tiết Use Case "Quản lý sinh viên":

| Tên Use Case        | Quản lý sinh viên   |
|---------------------|---|
| Mô tả               | Quản trị viên có thể thêm, sửa, xóa và tra cứu thông tin sinh viên.   |
| Tác nhân chính      | Quản trị viên hệ thống.   |
| Luồng sự kiện chính | <div>1. Quản trị viên đăng nhập vào hệ thống.</div> <div>2. Chọn module quản lý sinh viên.</div> <div>3. Thực hiện thao tác (thêm/sửa/xóa/tra cứu).</div> <div>4. Xác nhận và lưu thay đổi.</div> |

3.3.3. Use Case quản lý lớp học

- Sơ đồ use case:

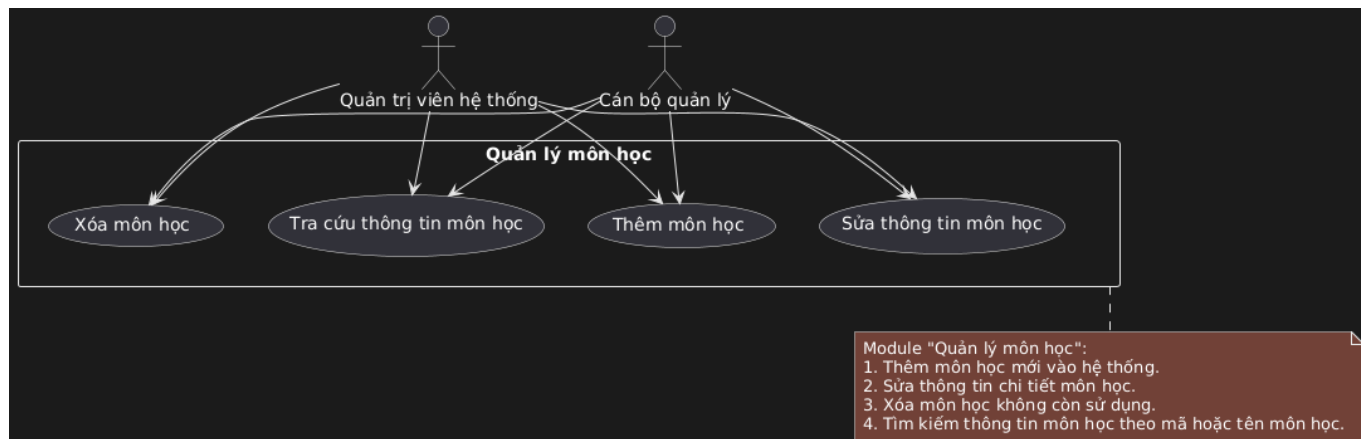


- Mô tả sơ đồ:
  - Tác nhân (Actors):
    - Quản trị viên hệ thống (Admin): Người quản lý toàn bộ dữ liệu lớp học.
    - Cán bộ quản lý (Manager): Người thực hiện các thao tác trên module quản lý lớp học.
- Các Use Case (Chức năng):

- Thêm lớp học: Tạo mới thông tin lớp học, bao gồm mã lớp, tên lớp, và danh sách sinh viên.
- Sửa thông tin lớp học: Cập nhật thông tin về lớp học (như tên lớp hoặc danh sách sinh viên).
- Xóa lớp học: Loại bỏ lớp học không còn cần thiết trong hệ thống.
- Tra cứu thông tin lớp học: Tìm kiếm lớp học dựa trên các tiêu chí như mã lớp hoặc tên lớp.

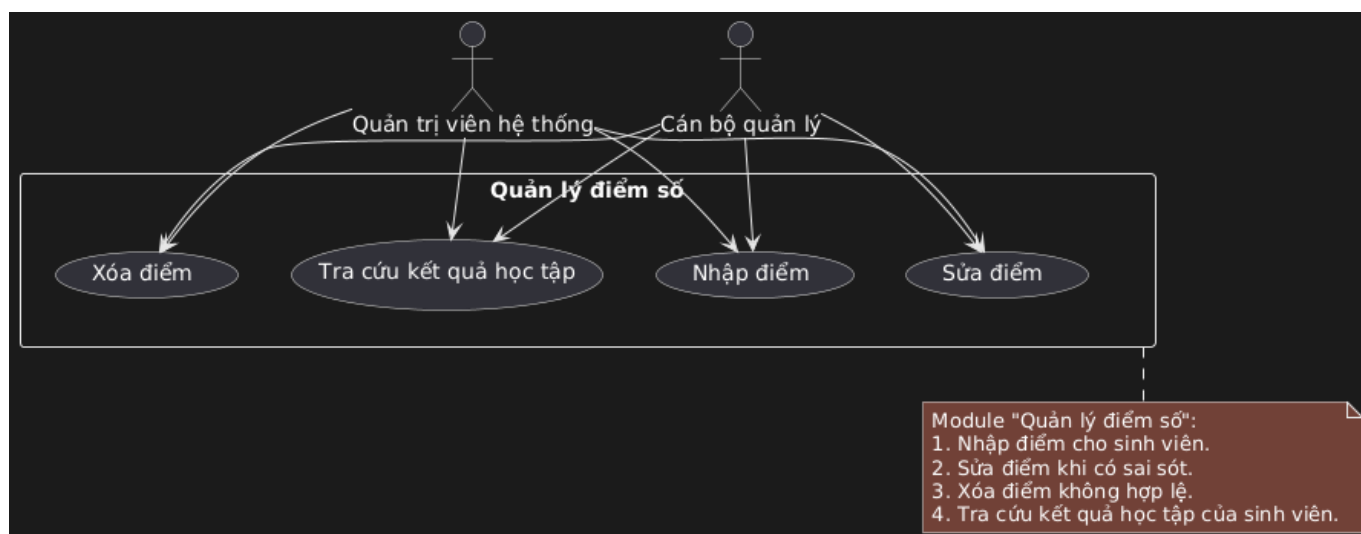
### 3.3.4. Use Case quản lý môn học

- Sơ đồ use case:



### 3.3.5. Use Case quản lý điểm số

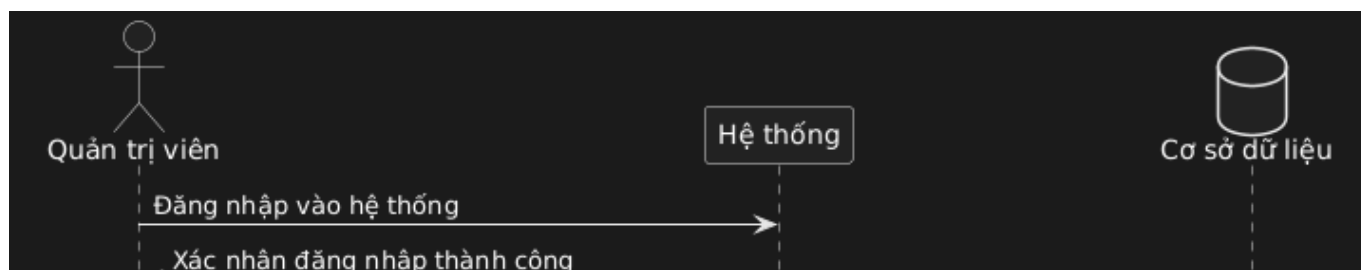
- Sơ đồ use case:

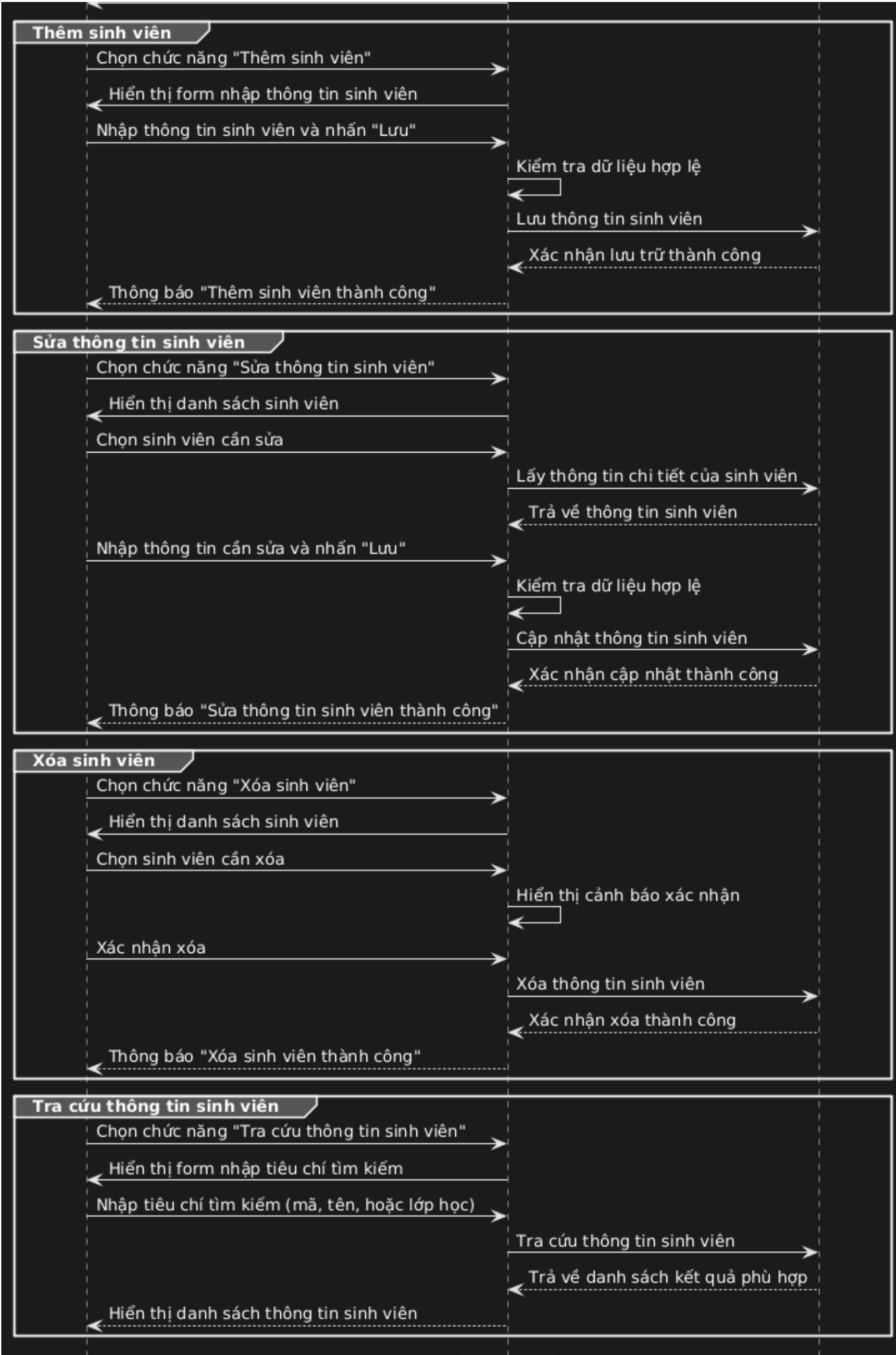


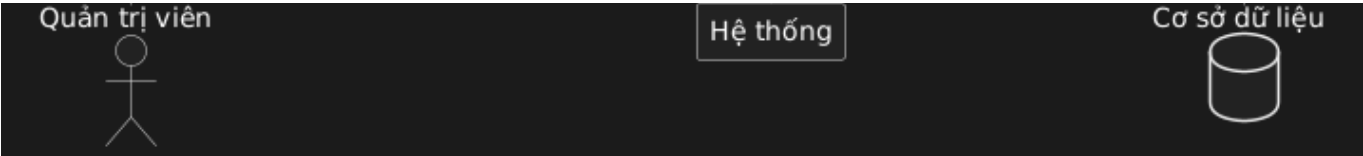
## 3.4. SƠ ĐỒ SEQUENCE (TUẦN TỰ)

### 3.4.1. Module quản lý sinh viên

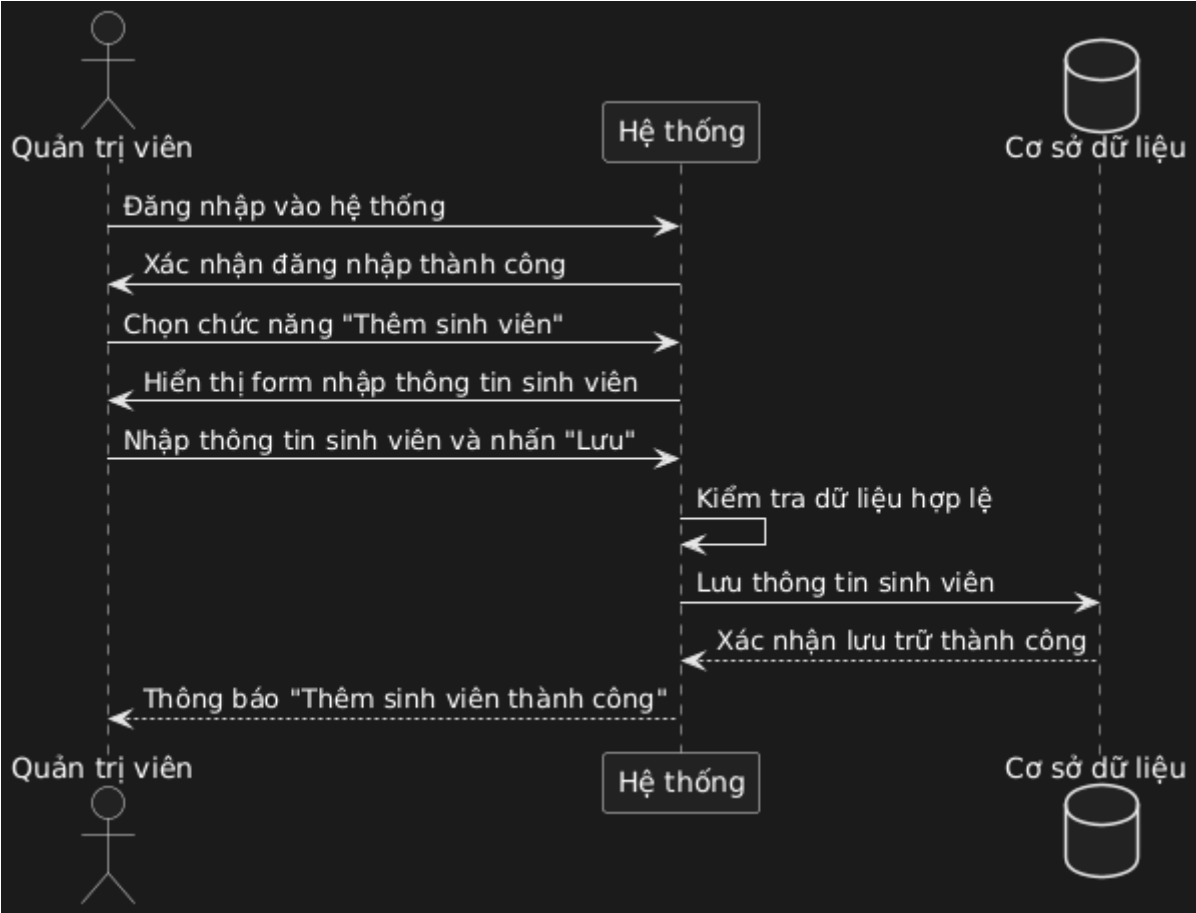
- Tổng quát ta sẽ có sơ đồ sau:





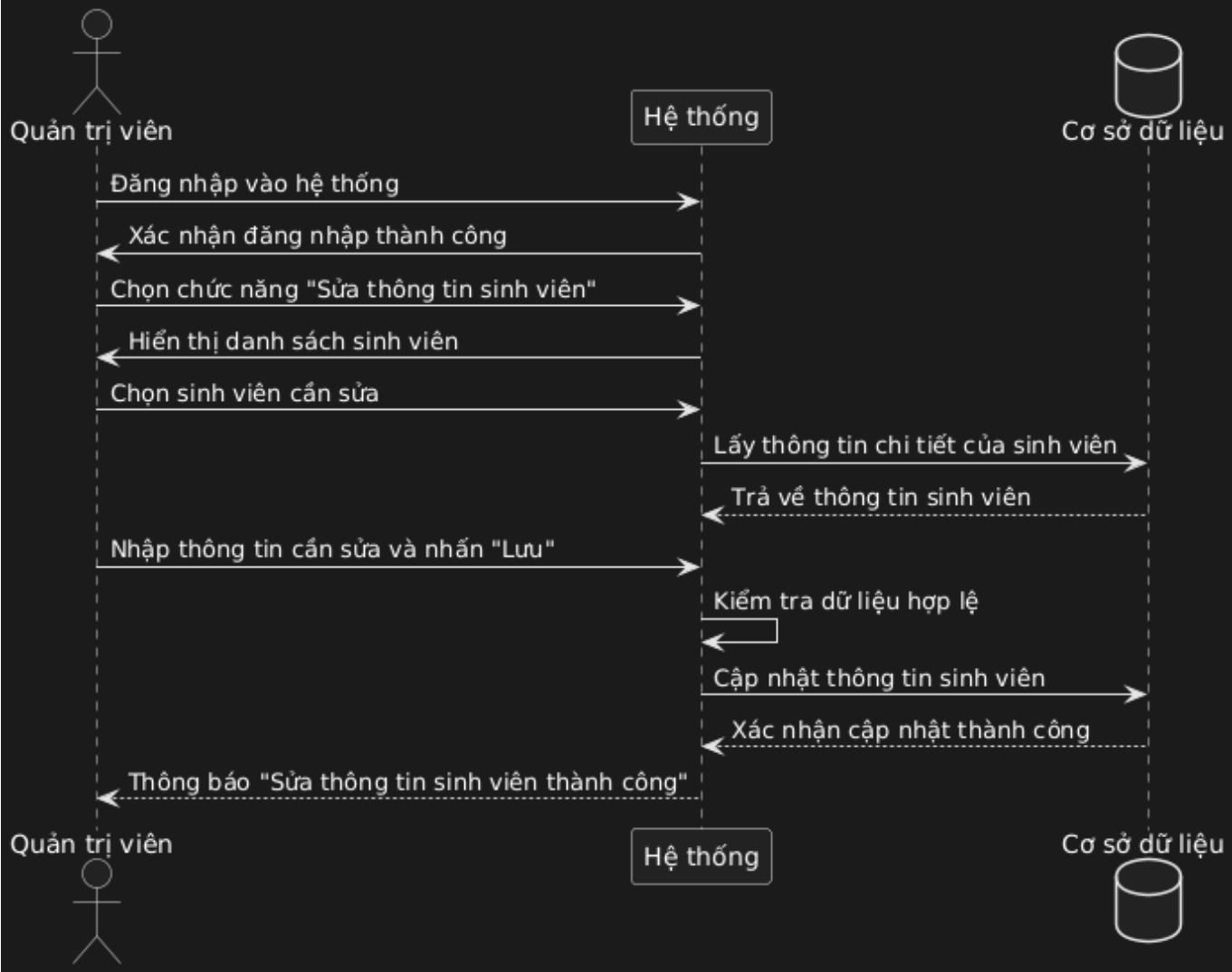


- Chi tiết từng chức năng thì ta có các sơ đồ sau:
  - Sequence Diagram: Thêm sinh viên

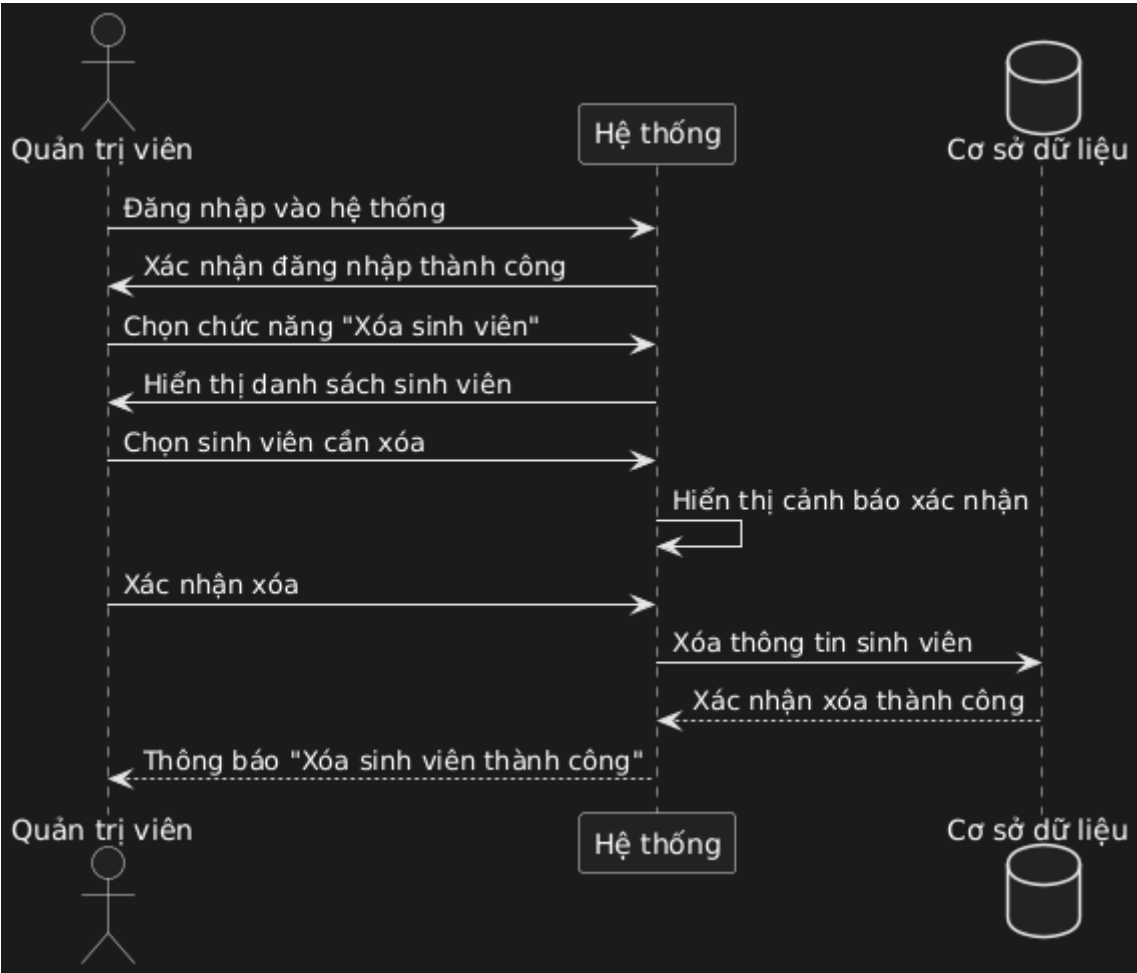


- Sequence

Diagram: Sửa thông tin sinh viên

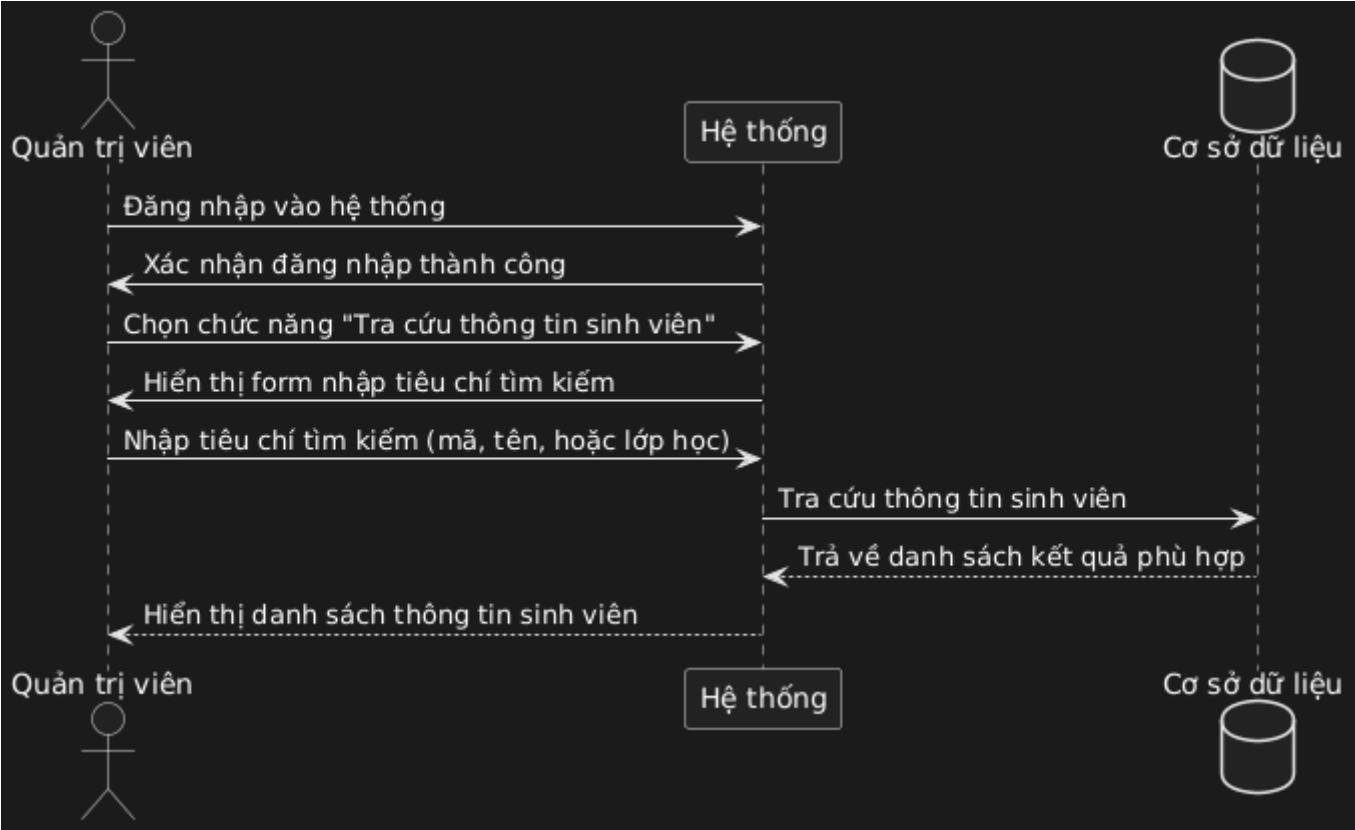


- Sequence Diagram: Xóa sinh viên



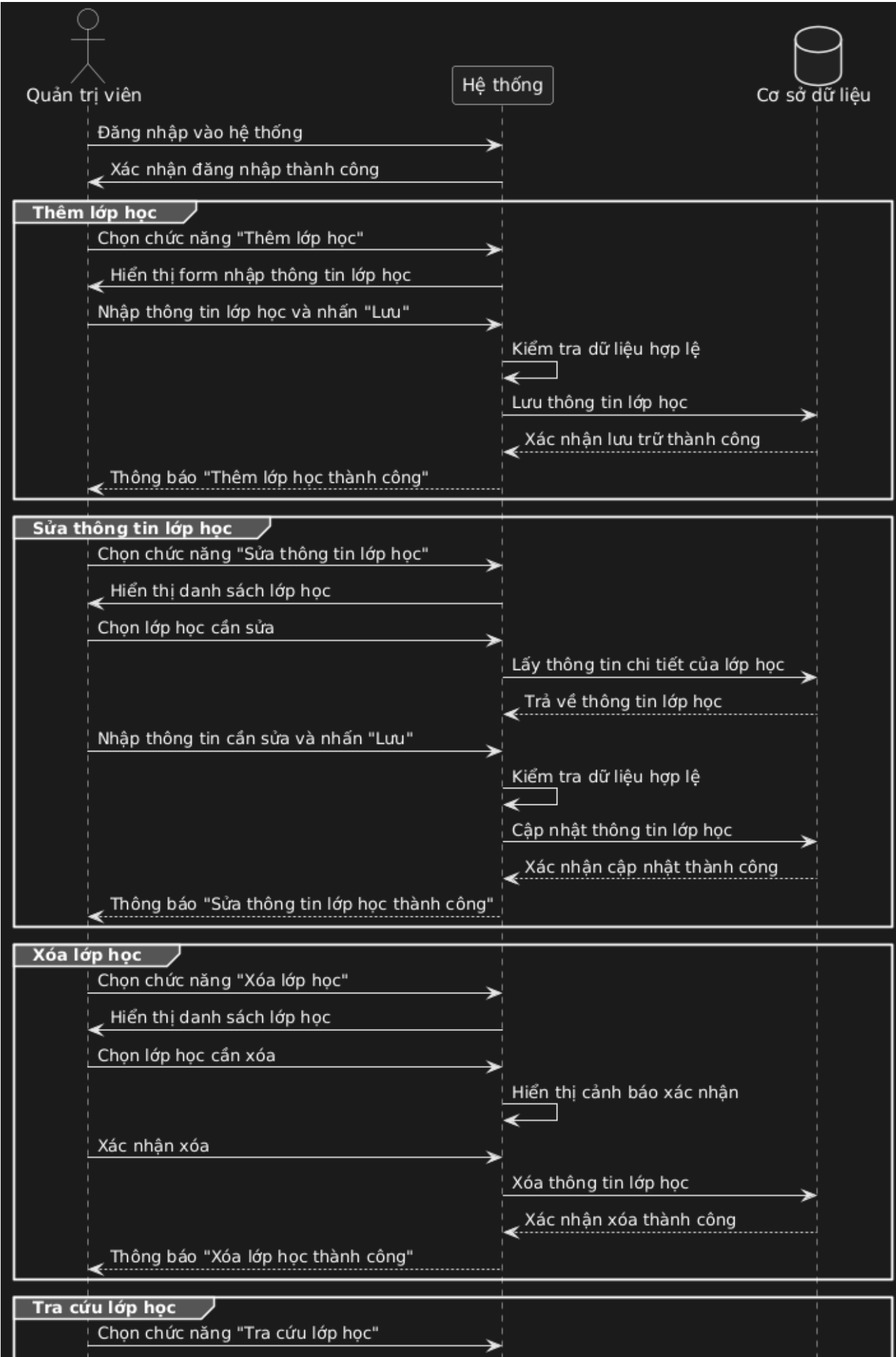
- Sequence

Diagram: Tra cứu thông tin sinh viên

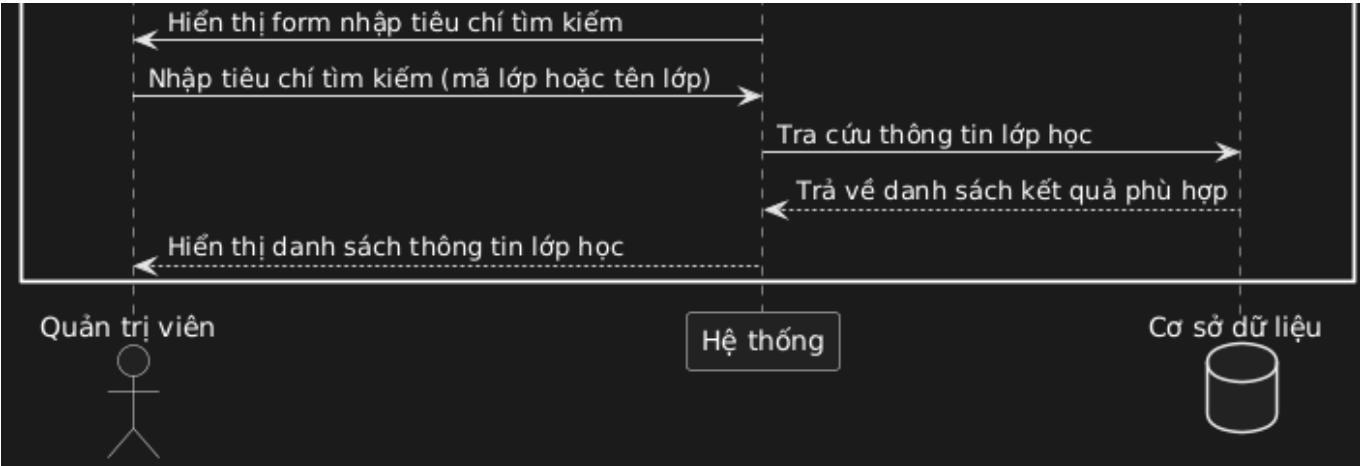


3.4.2. Module quản lý lớp học

- Sơ đồ tuần tự:

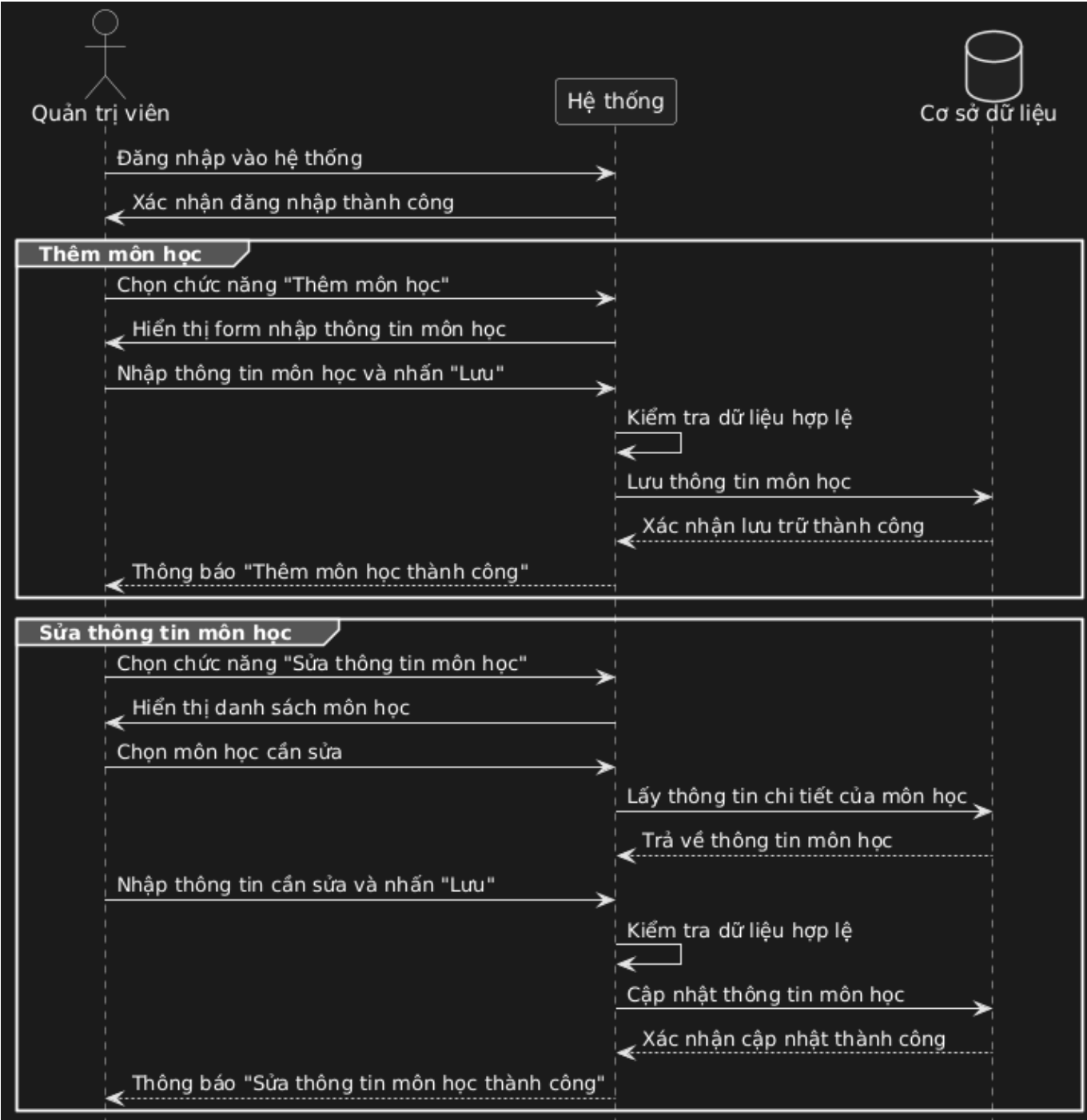


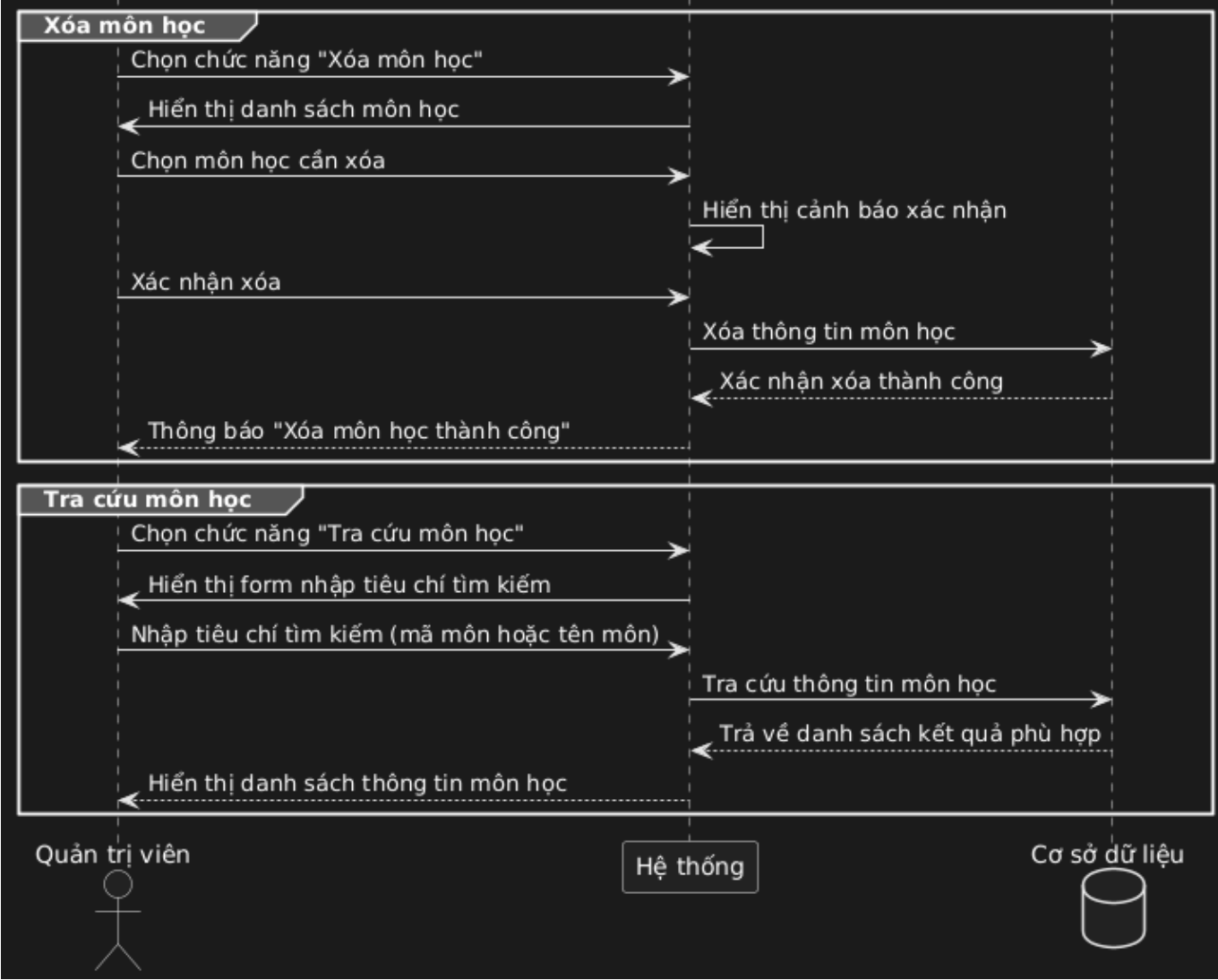




3.4.3. Module quản lý môn học

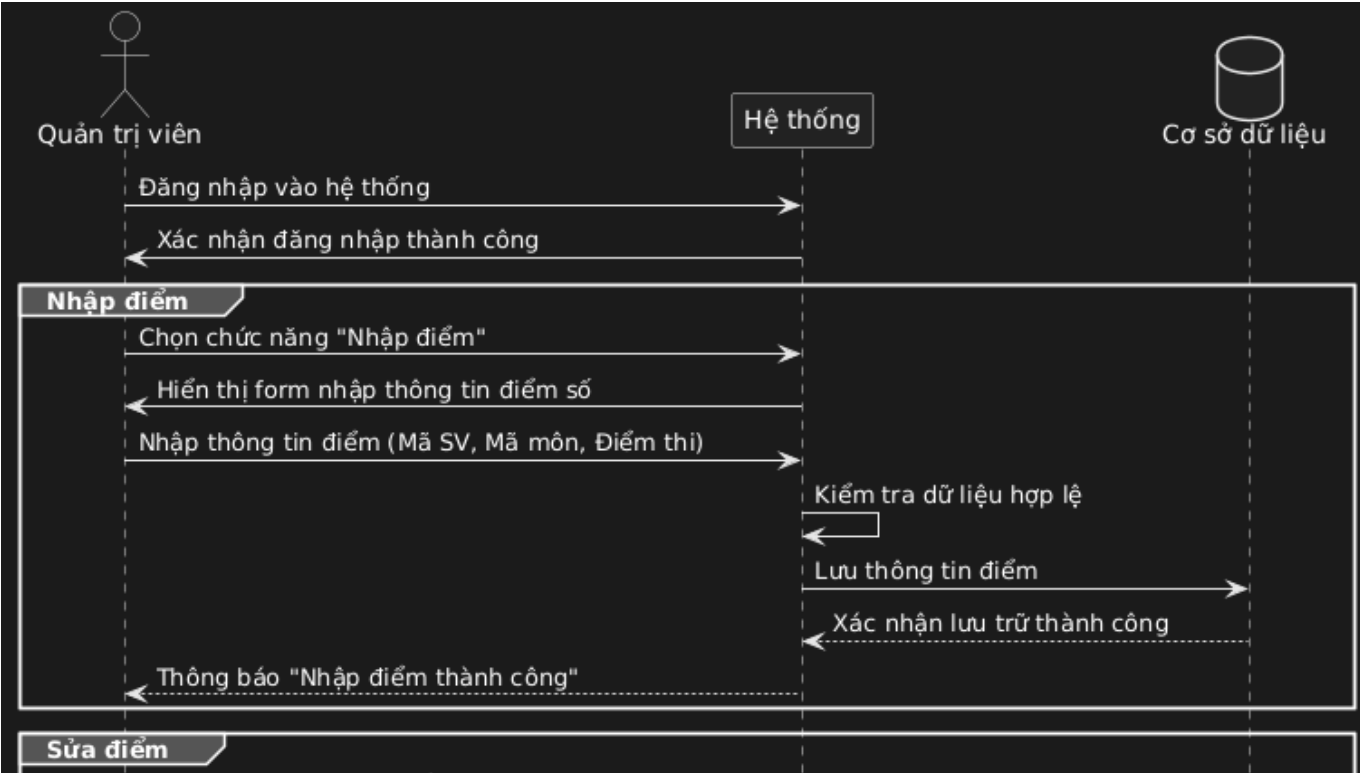
- Sơ đồ tuần tự:





3.4.4. Module quản lý điểm số

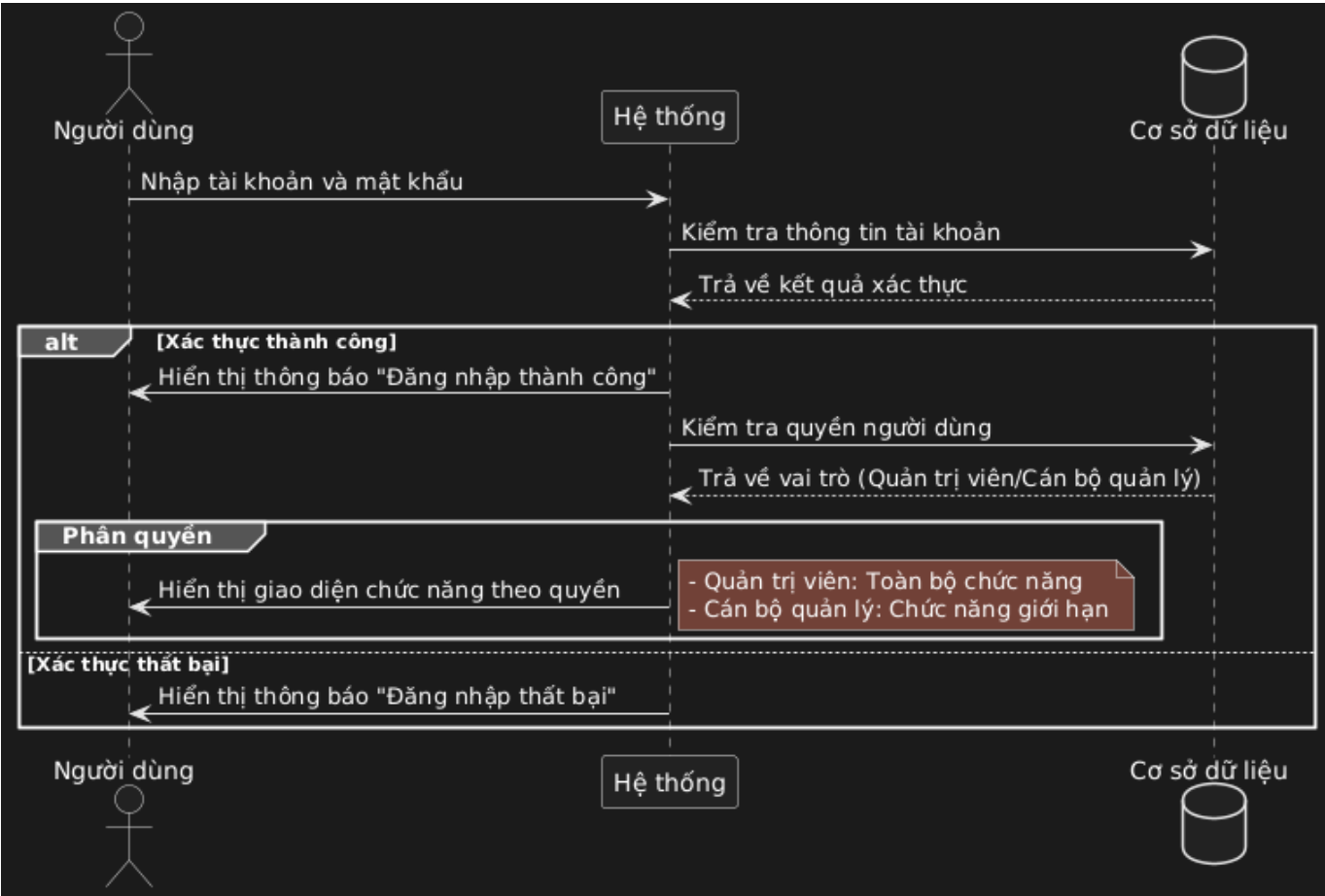
- Sơ đồ tuần tự:





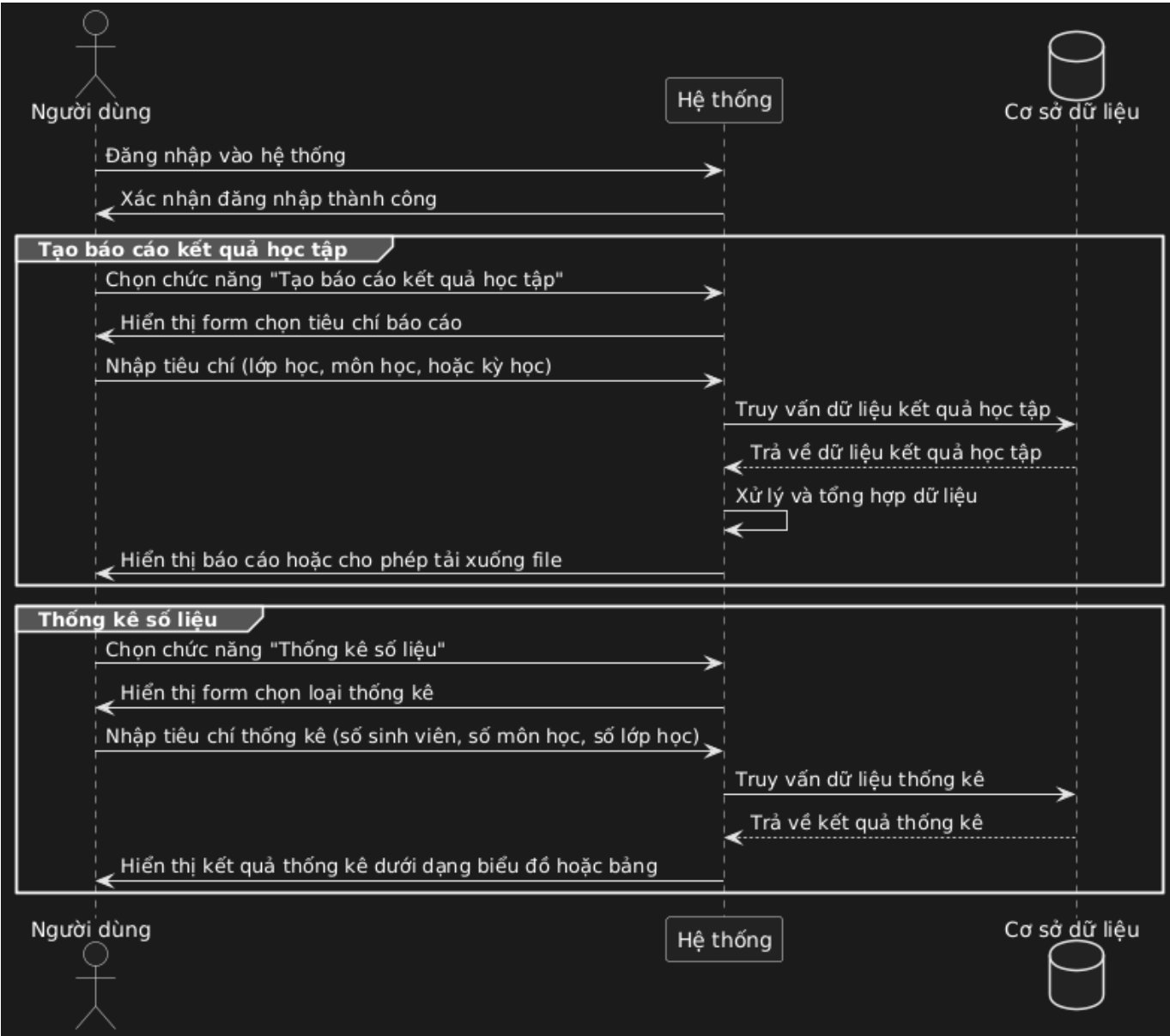
3.4.5. Module đăng nhập và phân quyền

- Sơ đồ tuần tự:



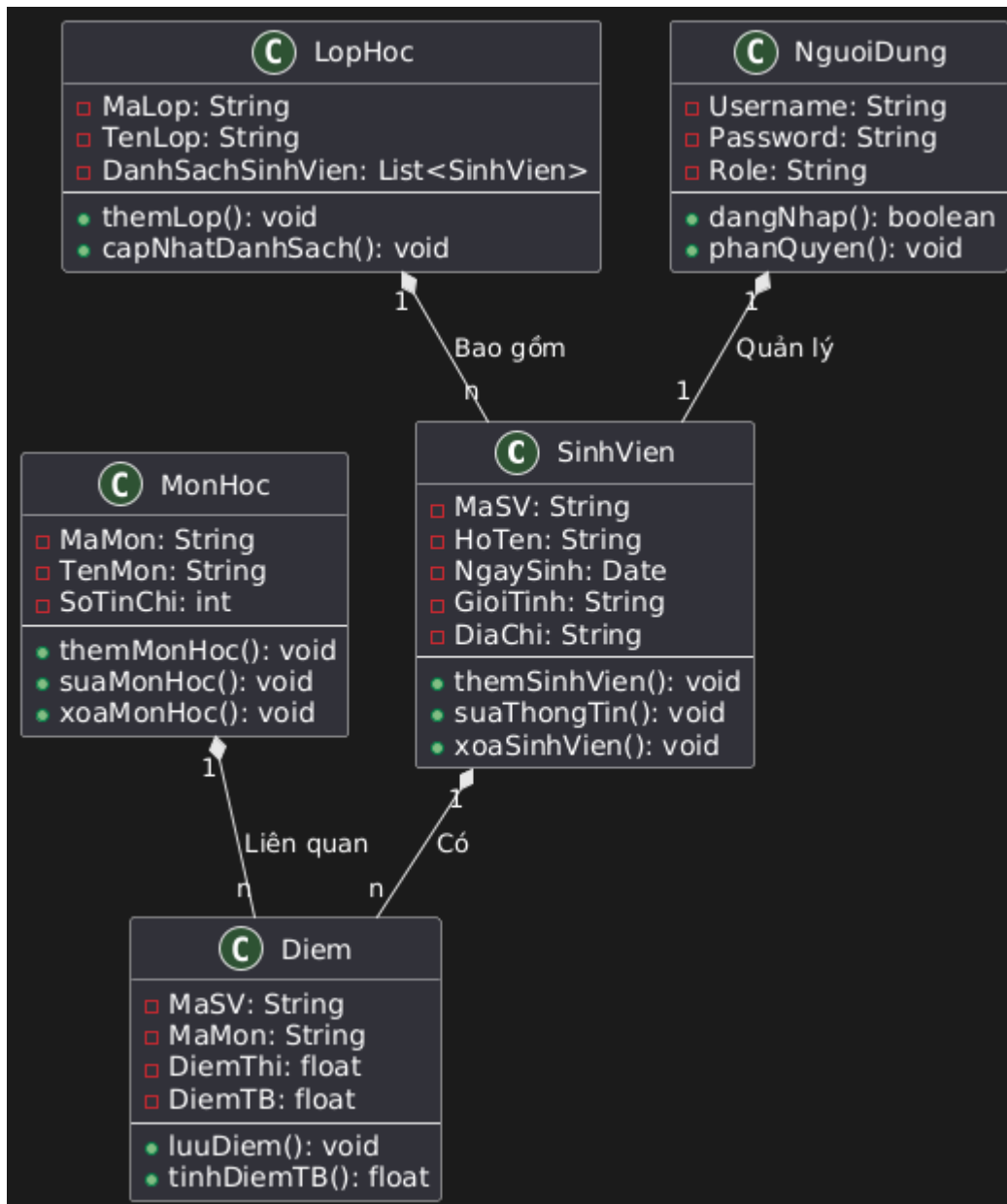
3.4.6. Module báo cáo và thống kê

- Sơ đồ tuần tự:



3.5. CLASS DIAGRAM

- Class Diagram thể hiện cấu trúc của hệ thống, bao gồm các lớp, thuộc tính, phương thức và mối quan hệ giữa chúng.
- Tổng quát ta sẽ có sơ đồ sau:



- Với từng module ta có từng sơ đồ như sau.

### 3.5.1 Module Quản lý Sinh viên

- Sơ đồ lớp **Quản lý Sinh viên**.



3.5.2 Module Quản lý Lớp Học

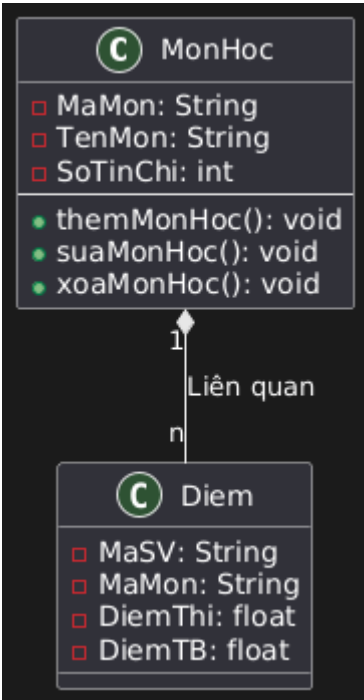
- Sơ đồ lớp **Quản lý Lớp Học**.



3.5.3 Module Quản lý Môn Học

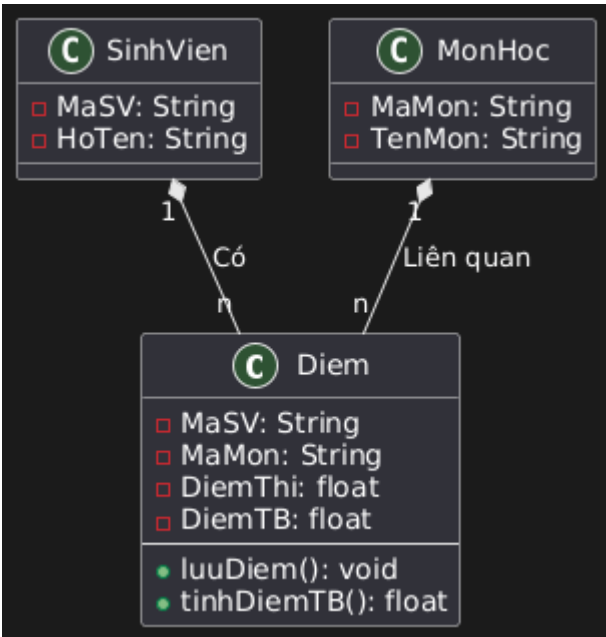
- Sơ đồ lớp **Module Quản lý Môn Học**.





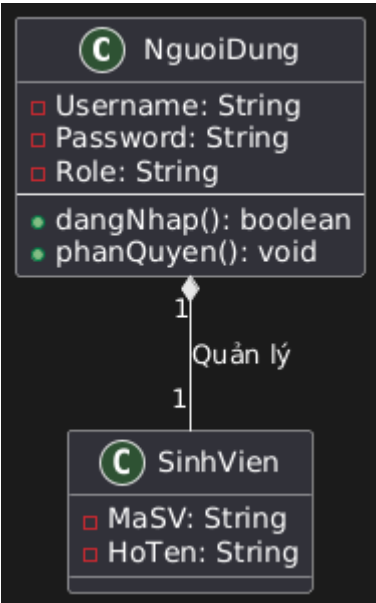
3.5.4 Module Quản lý Điểm Số

- Sơ đồ lớp **Module Quản lý Điểm Số**.



3.5.5 Module Đăng Nhập và Phân Quyền

- Sơ đồ lớp **Module Đăng Nhập và Phân Quyền**.



### 3.6 Đặc tả dữ liệu

- Đặc tả dữ liệu là phần quan trọng trong thiết kế hệ thống, giúp xác định cấu trúc, kiểu dữ liệu, và mối quan hệ giữa các bảng trong cơ sở dữ liệu (CSDL).

#### 3.6.1. Bảng Sinh viên (Student)

**Mục đích:**

- Lưu trữ thông tin cá nhân của sinh viên.

**Cấu trúc bảng:**

| Tên cột  | Kiểu dữ liệu | Mô tả   |
|----------|--------------|---|
| MaSV     | VARCHAR(10)  | Mã sinh viên (khóa chính).                      |
| HoTen    | VARCHAR(100) | Họ và tên sinh viên.                            |
| NgaySinh | DATE         | Ngày sinh.                                      |
| GioiTinh | VARCHAR(10)  | Giới tính (Nam/Nữ).                             |
| DiaChi   | VARCHAR(200) | Địa chỉ liên hệ.                                |
| MaLop    | VARCHAR(10)  | Mã lớp (khóa ngoại, liên kết với bảng Lớp học). |

#### 3.6.2. Bảng Lớp học (Class)

**Mục đích:**

- Quản lý thông tin các lớp học.

**Cấu trúc bảng:**

| Tên cột | Kiểu dữ liệu | Mô tả                    |
|---------|--------------|--------------------------|
| MaLop   | VARCHAR(10)  | Mã lớp học (khóa chính). |
| TenLop  | VARCHAR(50)  | Tên lớp học.             |
| Khoa    | VARCHAR(50)  | Khoa hoặc ngành học.     |

3.6.3. Bảng Môn học (Subject)

Mục đích:

- Lưu thông tin về các môn học.

Cấu trúc bảng:

| Tên cột  | Kiểu dữ liệu | Mô tả                    |
|----------|--------------|--------------------------|
| MaMon    | VARCHAR(10)  | Mã môn học (khóa chính). |
| TenMon   | VARCHAR(100) | Tên môn học.             |
| SoTinChi | INT          | Số tín chỉ của môn học.  |

3.6.4. Bảng Điểm (Grade)

Mục đích:

- Lưu trữ kết quả học tập của sinh viên.

Cấu trúc bảng:

| Tên cột | Kiểu dữ liệu | Mô tả   |
|---------|--------------|---|
| MaSV    | VARCHAR(10)  | Mã sinh viên (khóa ngoại, liên kết với bảng Sinh viên). |
| MaMon   | VARCHAR(10)  | Mã môn học (khóa ngoại, liên kết với bảng Môn học).     |
| DiemThi | FLOAT        | Điểm thi của sinh viên.                                 |
| DiemTB  | FLOAT        | Điểm trung bình môn.                                    |

3.6.5. Bảng Người dùng (User)

Mục đích:

Quản lý thông tin tài khoản đăng nhập.

Cấu trúc bảng:

| Tên cột | Kiểu dữ liệu | Mô tả |
|---------|--------------|-------|
|---------|--------------|-------|

| Tên cột  | Kiểu dữ liệu | Mô tả                                     |
|----------|--------------|---|
| Username | VARCHAR(50)  | Tên đăng nhập (khóa chính).               |
| Password | VARCHAR(100) | Mật khẩu (được mã hóa).                   |
| Role     | VARCHAR(20)  | Quyền hạn (Quản trị viên/Cán bộ quản lý). |

3.6.6. Bảng Thống kê/Báo cáo (Report)

Mục đích:

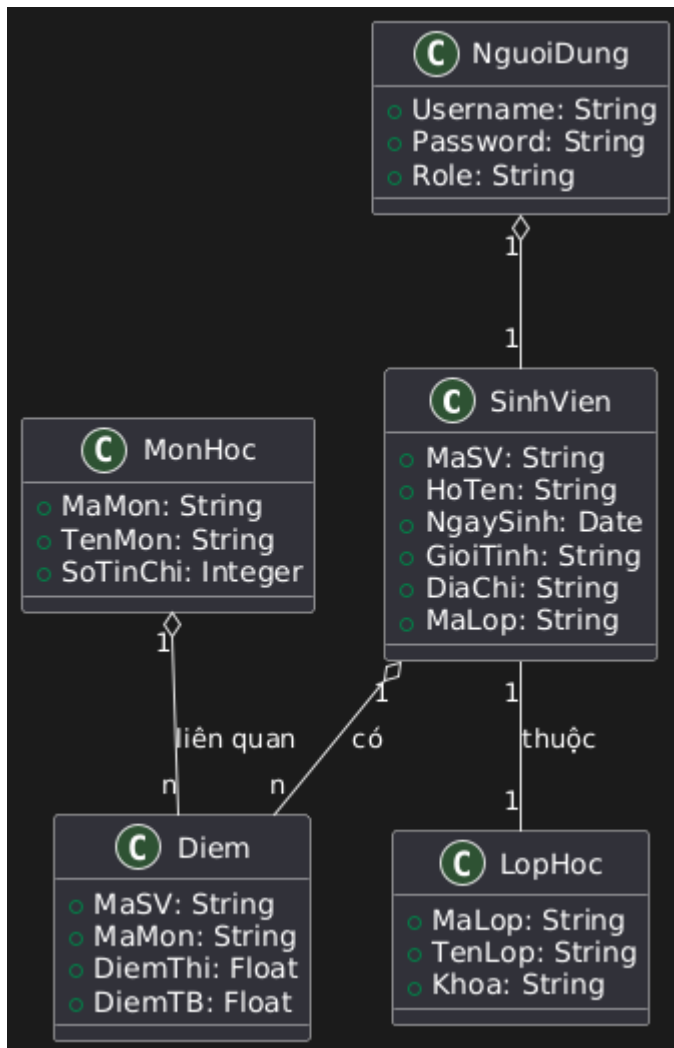
- Lưu dữ liệu tạm thời để tạo báo cáo hoặc thống kê.

Cấu trúc bảng:

| Tên cột | Kiểu dữ liệu | Mô tả                |
|---------|--------------|----------------------|
| MaSV    | VARCHAR(10)  | Mã sinh viên.        |
| MaLop   | VARCHAR(10)  | Mã lớp học.          |
| MaMon   | VARCHAR(10)  | Mã môn học.          |
| DiemThi | FLOAT        | Điểm thi.            |
| DiemTB  | FLOAT        | Điểm trung bình môn. |

Mối quan hệ giữa các bảng

- Dưới đây là sơ đồ mối quan hệ (ERD - Entity Relationship Diagram):

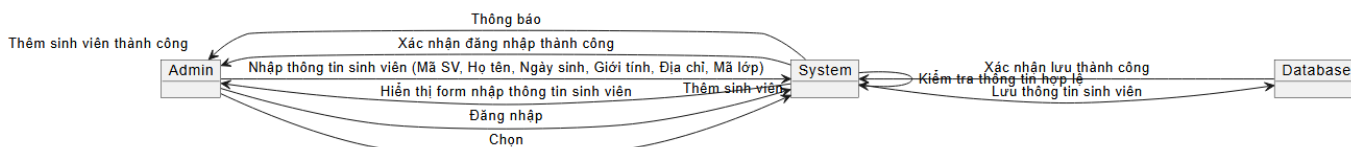


## 3.7 Biểu đồ cộng tác

### 3.7.1. Biểu đồ cộng tác: Quản lý Sinh Viên

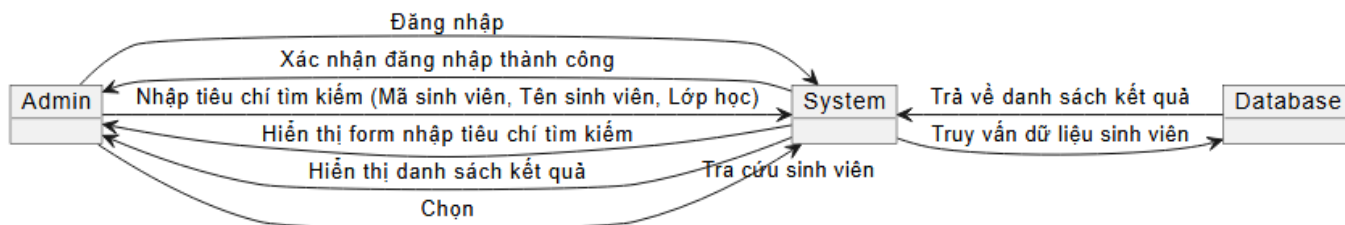
#### a. Thêm Sinh Viên

- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Thêm sinh viên".
  - Hệ thống hiển thị form để nhập thông tin sinh viên.
  - Quản trị viên điền các thông tin (Mã sinh viên, Họ tên, Ngày sinh, Giới tính, Địa chỉ, Mã lớp) và nhấn "Lưu".
  - Hệ thống kiểm tra thông tin hợp lệ:
    - Nếu hợp lệ: Tiến hành lưu dữ liệu vào cơ sở dữ liệu.
    - Nếu không hợp lệ: Hiển thị lỗi và yêu cầu sửa thông tin.
  - Sau khi lưu thành công, hệ thống thông báo hoàn tất.



#### b. Tra cứu Sinh Viên

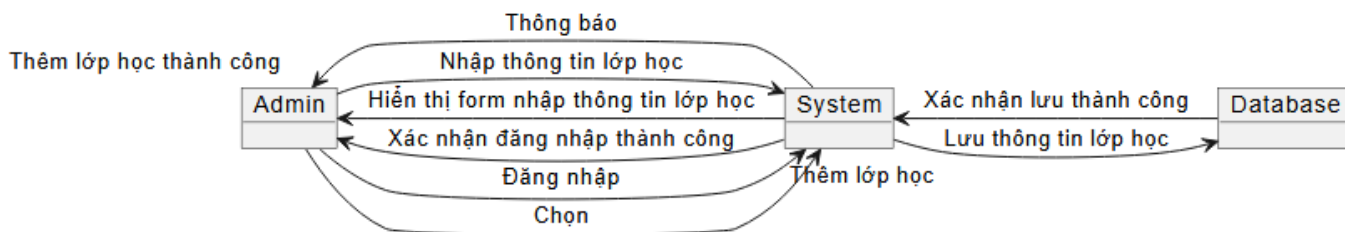
- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Tra cứu sinh viên".
  - Hệ thống hiển thị form để nhập tiêu chí tìm kiếm (Mã sinh viên, Tên sinh viên, Lớp học).
  - Quản trị viên nhập tiêu chí và nhấn "Tìm kiếm".
  - Hệ thống truy vấn dữ liệu trong cơ sở dữ liệu và trả về danh sách kết quả.
  - Kết quả hiển thị cho quản trị viên.



### 3.7.2. Biểu đồ cộng tác: Quản lý Lớp Học

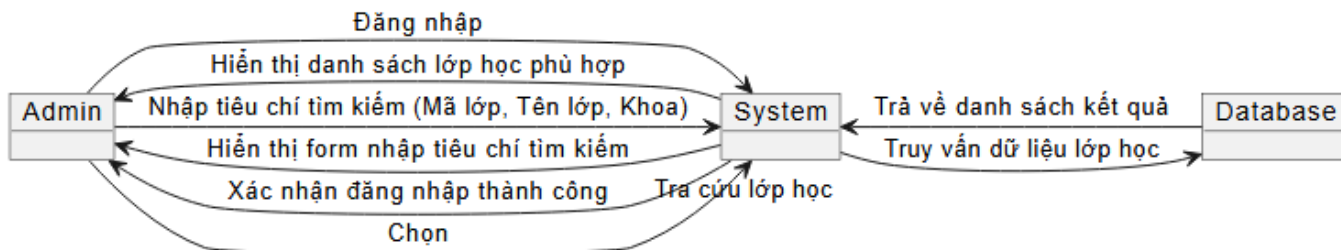
#### a. Thêm Lớp Học

- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Thêm lớp học".
  - Hệ thống hiển thị form để nhập thông tin lớp học (Mã lớp, Tên lớp, Khoa).
  - Quản trị viên điền thông tin và nhấn "Lưu".
  - Hệ thống kiểm tra dữ liệu:
    - Nếu hợp lệ: Tiến hành lưu vào cơ sở dữ liệu.
    - Nếu không hợp lệ: Yêu cầu quản trị viên sửa thông tin.
  - Sau khi lưu thành công, hệ thống thông báo hoàn tất.



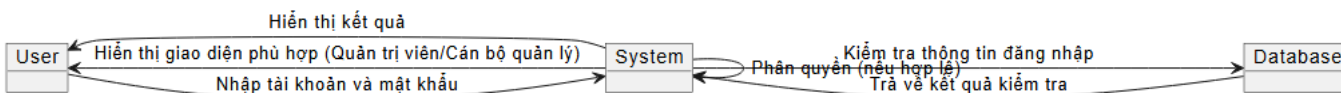
#### b. Tra cứu Lớp Học

- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Tra cứu lớp học".
  - Hệ thống hiển thị form nhập tiêu chí tìm kiếm (Mã lớp, Tên lớp, Khoa).
  - Quản trị viên nhập tiêu chí và nhấn "Tìm kiếm".
  - Hệ thống truy vấn cơ sở dữ liệu và trả về danh sách kết quả.
  - Hệ thống hiển thị danh sách lớp học phù hợp.



### 3.7.3. Biểu đồ cộng tác: Đăng Nhập và Phân Quyền

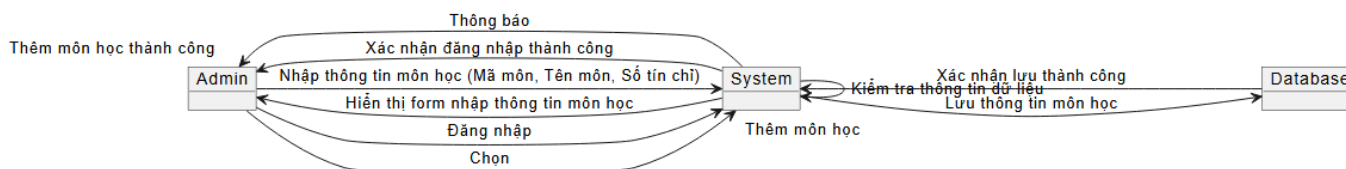
- Mô tả:
  - Người dùng nhập tài khoản và mật khẩu vào hệ thống.
  - Hệ thống kiểm tra thông tin đăng nhập trong cơ sở dữ liệu:
    - Nếu hợp lệ: Xác nhận đăng nhập và phân quyền.
    - Nếu không hợp lệ: Hiển thị thông báo lỗi.
  - Hệ thống hiển thị giao diện phù hợp với quyền hạn (Quản trị viên hoặc Cán bộ quản lý).



### 3.7.4. Biểu đồ cộng tác: Quản lý Môn Học

#### a. Thêm môn học

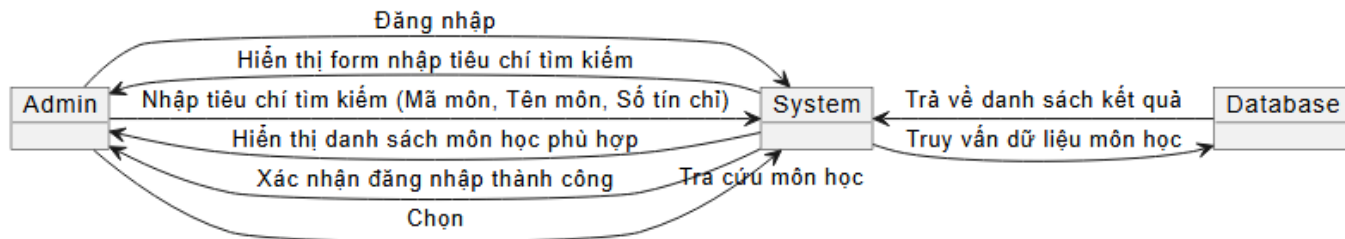
- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Thêm môn học".
  - Hệ thống hiển thị form nhập thông tin môn học (Mã môn, Tên môn, Số tín chỉ).
  - Quản trị viên điền thông tin và nhấn "Lưu".
  - Hệ thống kiểm tra dữ liệu:
    - Nếu hợp lệ: Tiến hành lưu vào cơ sở dữ liệu.
    - Nếu không hợp lệ: Yêu cầu quản trị viên sửa thông tin.
  - Sau khi lưu thành công, hệ thống thông báo hoàn tất.



#### b. Tra cứu môn học

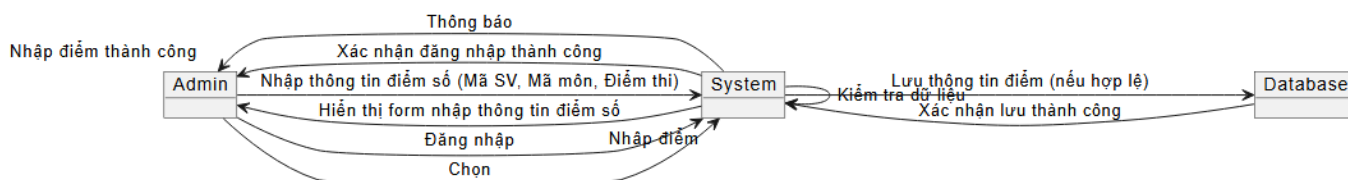
- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Tra cứu môn học".
  - Hệ thống hiển thị form nhập tiêu chí tìm kiếm (Mã môn, Tên môn, Số tín chỉ).
  - Quản trị viên nhập tiêu chí và nhấn "Tìm kiếm".
  - Hệ thống truy vấn cơ sở dữ liệu và trả về danh sách kết quả.
  - Hệ thống hiển thị danh sách môn học phù hợp.





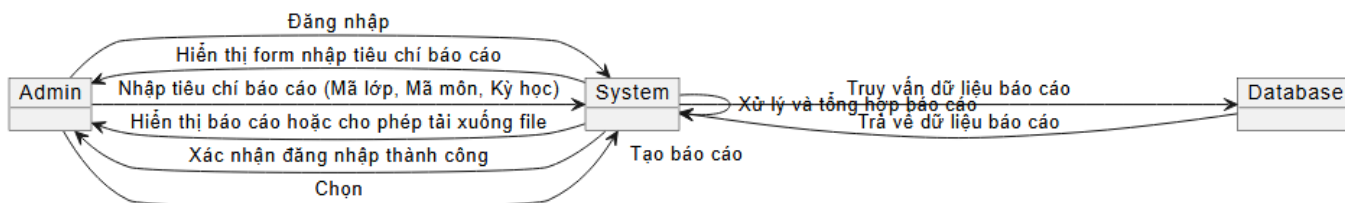
### 3.7.5. Biểu đồ cộng tác: Quản lý Điểm Số

- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Nhập điểm".
  - Hệ thống hiển thị form nhập thông tin điểm số (Mã SV, Mã môn, Điểm thi).
  - Quản trị viên nhập điểm và nhấn "Lưu".
  - Hệ thống kiểm tra dữ liệu:
    - Nếu hợp lệ: Lưu vào cơ sở dữ liệu.
    - Nếu không hợp lệ: Yêu cầu quản trị viên chỉnh sửa.
  - Sau khi lưu thành công, hệ thống thông báo hoàn tất.



### 3.7.6. Biểu đồ cộng tác: Báo Cáo và Thống Kê

- Mô tả:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Tạo báo cáo".
  - Hệ thống hiển thị form nhập tiêu chí báo cáo (Mã lớp, Mã môn, hoặc kỳ học).
  - Quản trị viên nhập tiêu chí và nhấn "Tạo báo cáo".
  - Hệ thống truy vấn dữ liệu và xử lý báo cáo.
  - Hệ thống hiển thị kết quả báo cáo hoặc cho phép tải xuống file.

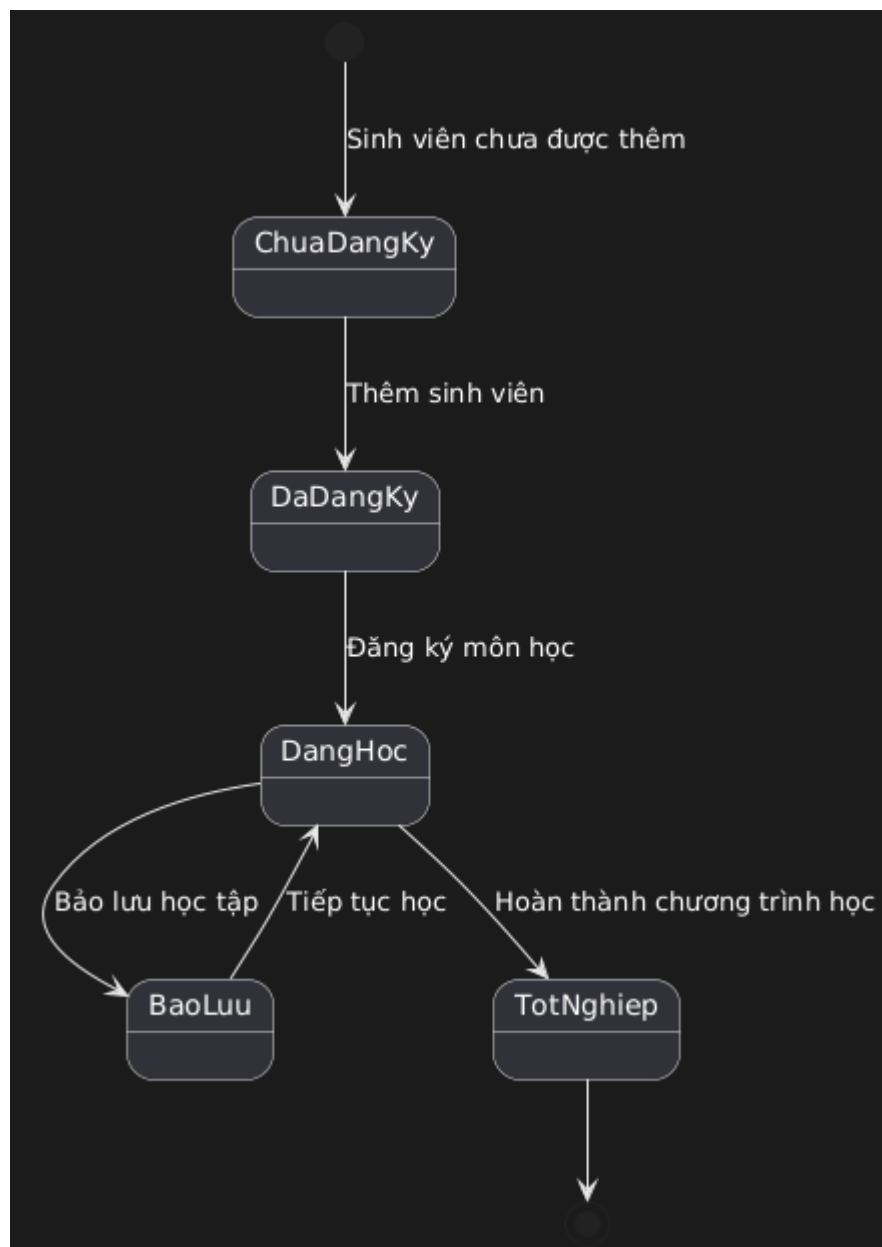


## 3.8 Biểu đồ trạng thái

### 3.8.1. Biểu đồ trạng thái cho Sinh Viên

- Mô tả:
  - Biểu đồ trạng thái này mô tả các giai đoạn trạng thái của một sinh viên trong hệ thống từ lúc được thêm vào đến khi hoàn thành chương trình học hoặc bảo lưu.

- Các trạng thái:
  - Chưa đăng ký: Sinh viên chưa có trong hệ thống.
  - Đã đăng ký: Sinh viên được thêm vào hệ thống và thuộc một lớp học.
  - Đang học: Sinh viên đã đăng ký các môn học.
  - Bảo lưu: Sinh viên tạm dừng chương trình học.
  - Tốt nghiệp: Sinh viên hoàn thành chương trình học.



### 3.8.2. Biểu đồ trạng thái cho Lớp Học

- Mô tả:
  - Mô tả các trạng thái của một lớp học từ khi được khởi tạo đến khi kết thúc.
- Các trạng thái:
  - Chưa khởi tạo: Lớp học chưa được thêm vào hệ thống.
  - Đang hoạt động: Lớp học đang diễn ra và có sinh viên tham gia.
  - Hoàn thành: Lớp học đã kết thúc.



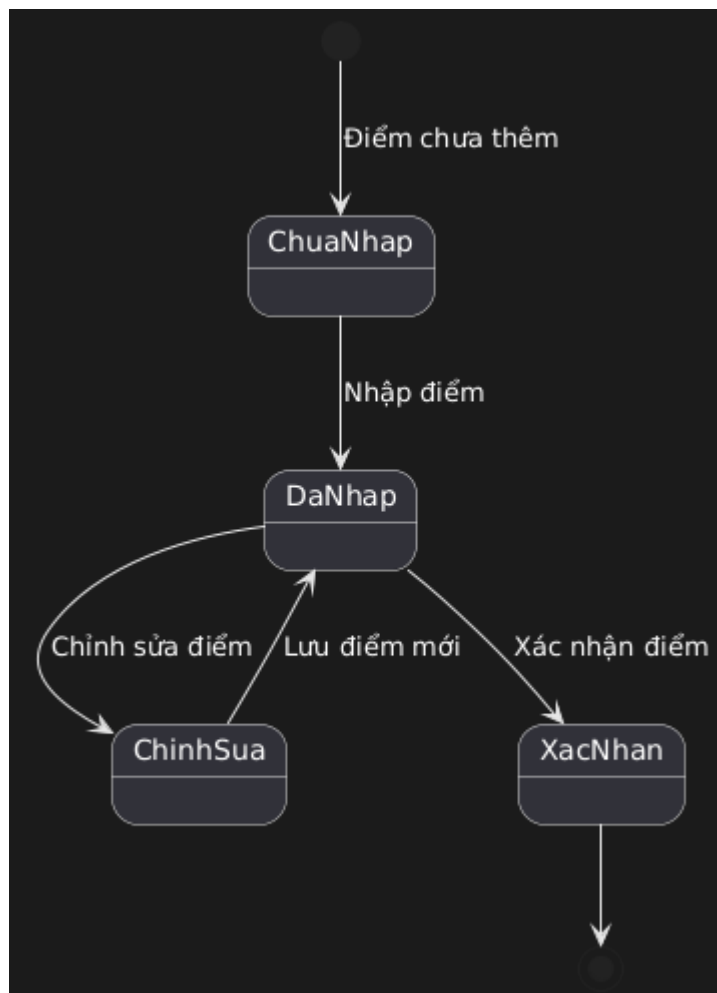
### 3.8.3. Biểu đồ trạng thái cho Môn Học

- Mô tả:
  - Mô tả các trạng thái của một môn học trong hệ thống từ lúc khởi tạo đến khi hoàn thành hoặc bị hủy bỏ.
- Các trạng thái:
  - Chưa khởi tạo: Môn học chưa được thêm vào hệ thống.
  - Đang hoạt động: Môn học đang được giảng dạy.
  - Hoàn thành: Môn học đã kết thúc.
  - Hủy bỏ: Môn học bị hủy bỏ.



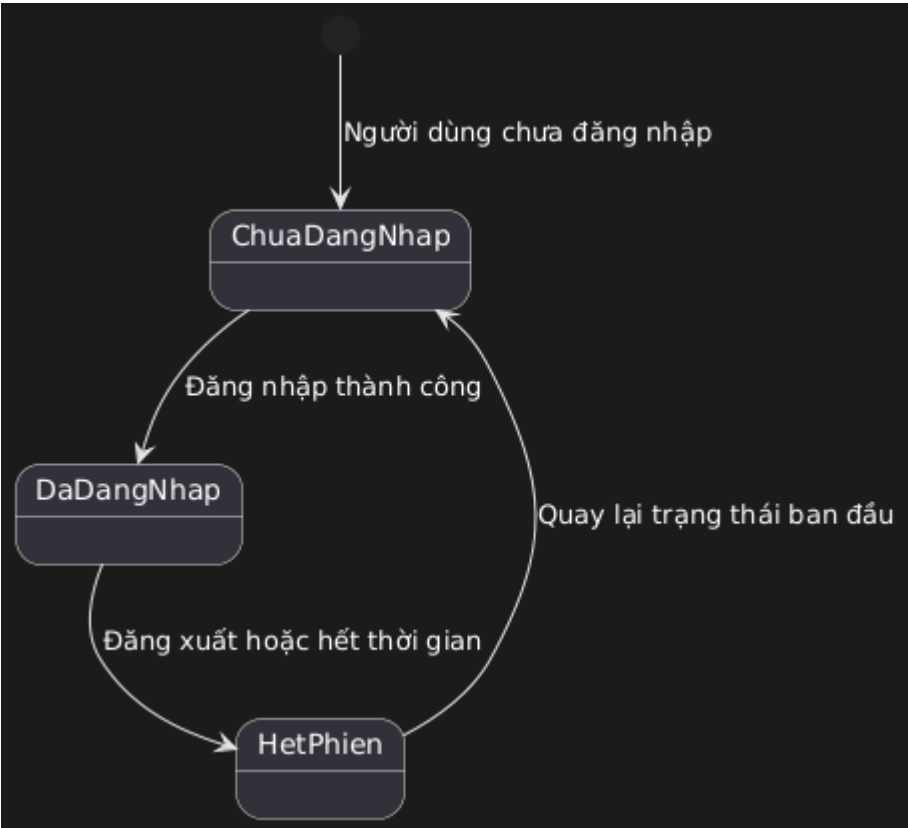
#### 3.8.4. Biểu đồ trạng thái cho Điểm Số

- Mô tả:
  - Mô tả các trạng thái của điểm số từ lúc chưa được nhập vào hệ thống đến khi xác nhận hoàn tất.
- Các trạng thái:
  - Chưa nhập: Điểm số chưa được thêm vào hệ thống.
  - Đã nhập: Điểm số được nhập vào hệ thống nhưng chưa duyệt.
  - Chính sửa: Điểm số được cập nhật.
  - Xác nhận: Điểm số đã được xác nhận và lưu cố định.



### 3.8.5. Biểu đồ trạng thái cho Người Dùng (Đăng nhập và Phân quyền)

- Mô tả:
  - Mô tả các trạng thái của một người dùng khi tương tác với hệ thống, bao gồm các trạng thái từ đăng nhập đến đăng xuất.
- Các trạng thái:
  - Chưa đăng nhập: Người dùng chưa truy cập vào hệ thống.
  - Đã đăng nhập: Người dùng đã được xác thực và phân quyền.
  - Hết phiên: Phiên làm việc của người dùng đã kết thúc do đăng xuất hoặc hết thời gian.



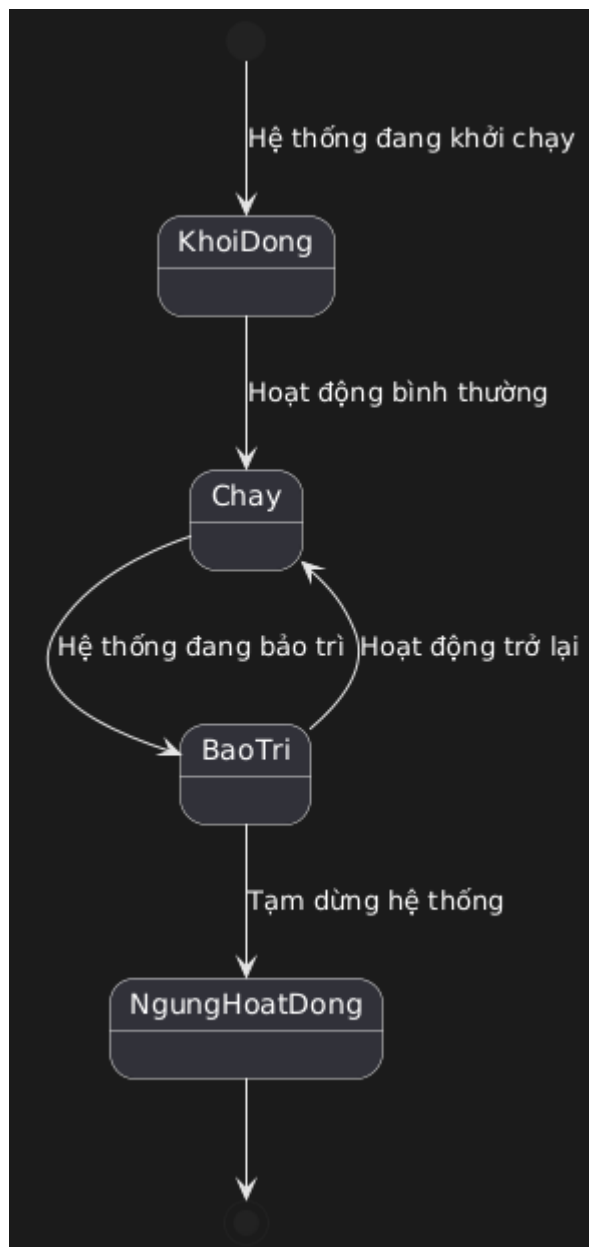
3.8.6. Biểu đồ trạng thái cho Báo Cáo

- Mô tả:
  - Mô tả các trạng thái của một báo cáo từ khi được yêu cầu đến khi hoàn thành hoặc bị hủy.
- Các trạng thái:
  - Chưa tạo: Báo cáo chưa được yêu cầu.
  - Đang tạo: Báo cáo đang được xử lý và tổng hợp dữ liệu.
  - Hoàn thành: Báo cáo đã được tạo xong và hiển thị/tải về.
  - Hủy: Quá trình tạo báo cáo bị hủy do lỗi hoặc yêu cầu từ người dùng.



### 3.8.7. Biểu đồ trạng thái cho Hệ Thống

- Mô tả:
  - Mô tả trạng thái tổng quát của hệ thống từ lúc khởi động đến khi hoạt động hoặc bảo trì.
- Các trạng thái:
  - Khởi động: Hệ thống đang khởi chạy.
  - Chạy: Hệ thống hoạt động bình thường.
  - Đang bảo trì: Hệ thống đang được bảo trì hoặc cập nhật.
  - Ngừng hoạt động: Hệ thống tạm dừng hoạt động.

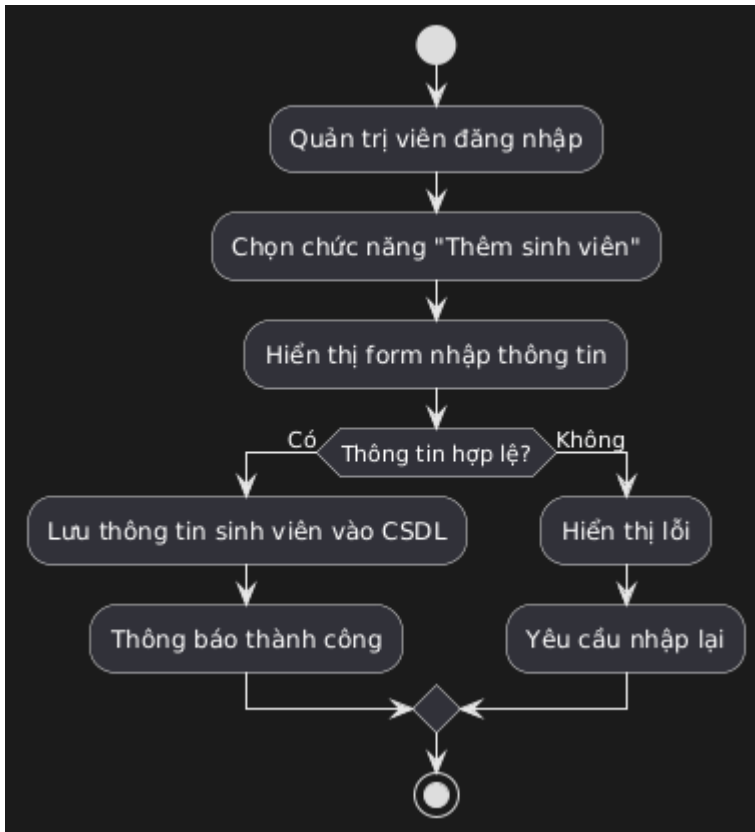


## 3.9 Biểu đồ hoạt động

### 3.9.1. Biểu đồ hoạt động: Thêm Sinh Viên

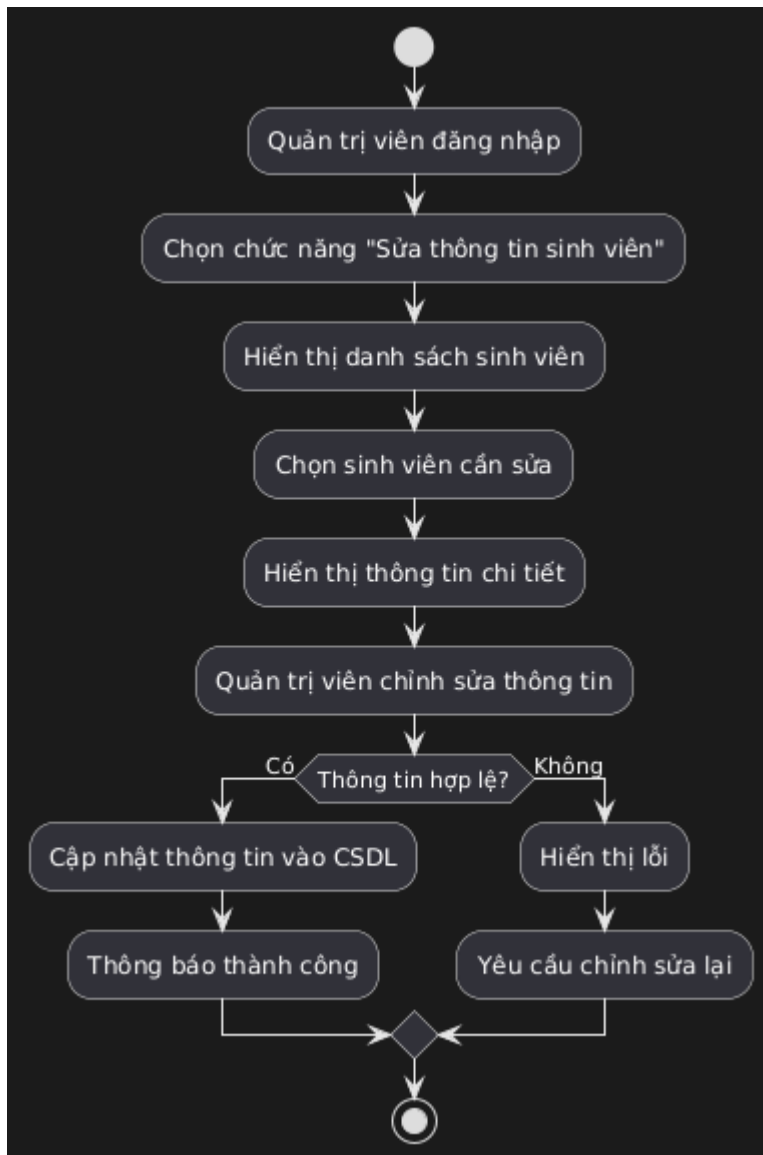
- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Thêm sinh viên".
  - Hệ thống hiển thị form để nhập thông tin sinh viên.
  - Quản trị viên điền thông tin cần thiết và nhấn "Lưu".
  - Hệ thống kiểm tra dữ liệu:
    - Nếu hợp lệ: Thông tin được lưu vào cơ sở dữ liệu và hiển thị thông báo thành công.
    - Nếu không hợp lệ: Hiển thị lỗi và yêu cầu nhập lại.





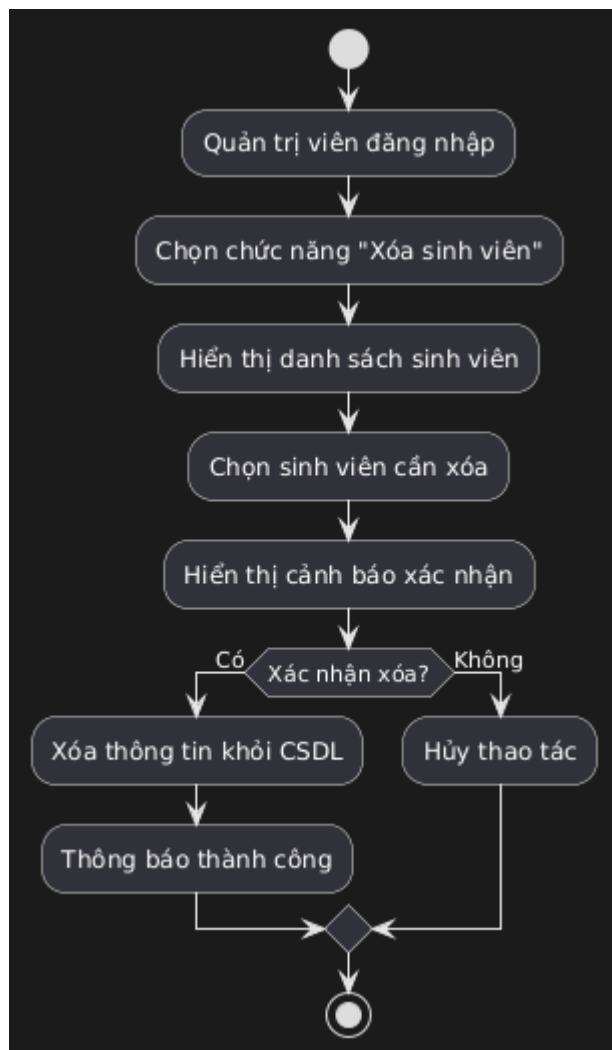
### 3.9.2. Biểu đồ hoạt động: Sửa Thông Tin Sinh Viên

- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Sửa thông tin sinh viên".
  - Hệ thống hiển thị danh sách sinh viên.
  - Quản trị viên chọn sinh viên cần chỉnh sửa.
  - Hệ thống hiển thị thông tin chi tiết của sinh viên.
  - Quản trị viên chỉnh sửa thông tin và nhấn "Lưu".
  - Hệ thống kiểm tra thông tin:
    - Nếu hợp lệ: Cập nhật vào cơ sở dữ liệu và thông báo thành công.
    - Nếu không hợp lệ: Hiển thị lỗi và yêu cầu chỉnh sửa lại.



### 3.9.3. Biểu đồ hoạt động: Xóa Sinh Viên

- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Xóa sinh viên".
  - Hệ thống hiển thị danh sách sinh viên.
  - Quản trị viên chọn sinh viên cần xóa.
  - Hệ thống hiển thị cảnh báo xác nhận xóa.
  - Quản trị viên xác nhận:
    - Nếu đồng ý: Hệ thống xóa thông tin khỏi cơ sở dữ liệu và hiển thị thông báo thành công.
    - Nếu không đồng ý: Hủy thao tác và quay lại danh sách sinh viên.



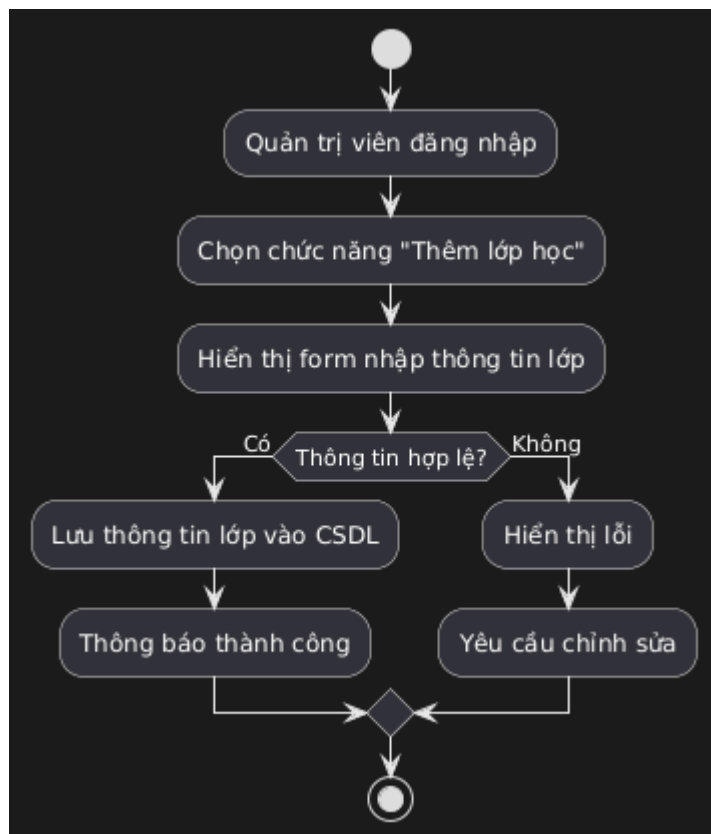
### 3.9.4. Biểu đồ hoạt động: Tra Cứu Sinh Viên

- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Tra cứu sinh viên".
  - Hệ thống hiển thị form nhập tiêu chí tìm kiếm.
  - Quản trị viên nhập tiêu chí (Mã SV, Tên SV, Lớp học, ...).
  - Hệ thống truy vấn cơ sở dữ liệu.
  - Hiển thị danh sách kết quả phù hợp cho quản trị viên.



### 3.9.5. Biểu đồ hoạt động: Thêm Lớp Học

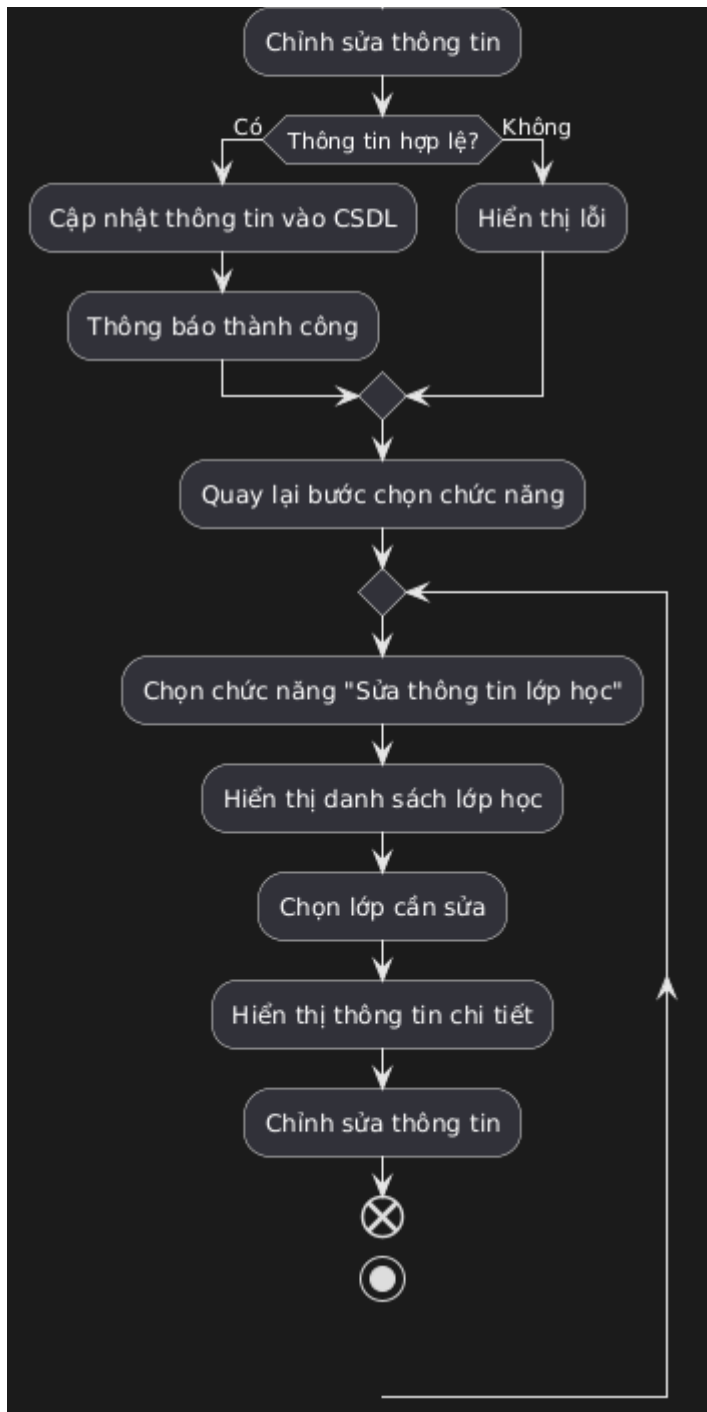
- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Thêm lớp học".
  - Hệ thống hiển thị form nhập thông tin lớp học.
  - Quản trị viên điền thông tin cần thiết (Mã lớp, Tên lớp, Khoa).
  - Hệ thống kiểm tra dữ liệu:
    - Nếu hợp lệ: Lưu thông tin vào cơ sở dữ liệu và thông báo thành công.
    - Nếu không hợp lệ: Hiển thị lỗi và yêu cầu chỉnh sửa.



### 3.9.6. Biểu đồ hoạt động: Sửa Thông Tin Lớp Học

- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Sửa thông tin lớp học".
  - Hệ thống hiển thị danh sách lớp học.
  - Quản trị viên chọn lớp học cần sửa.
  - Hệ thống hiển thị thông tin chi tiết.
  - Quản trị viên chỉnh sửa thông tin và nhấn "Lưu".
  - Hệ thống kiểm tra thông tin:
    - Nếu hợp lệ: Cập nhật vào cơ sở dữ liệu và thông báo thành công.
    - Nếu không hợp lệ: Hiển thị lỗi và yêu cầu sửa lại.





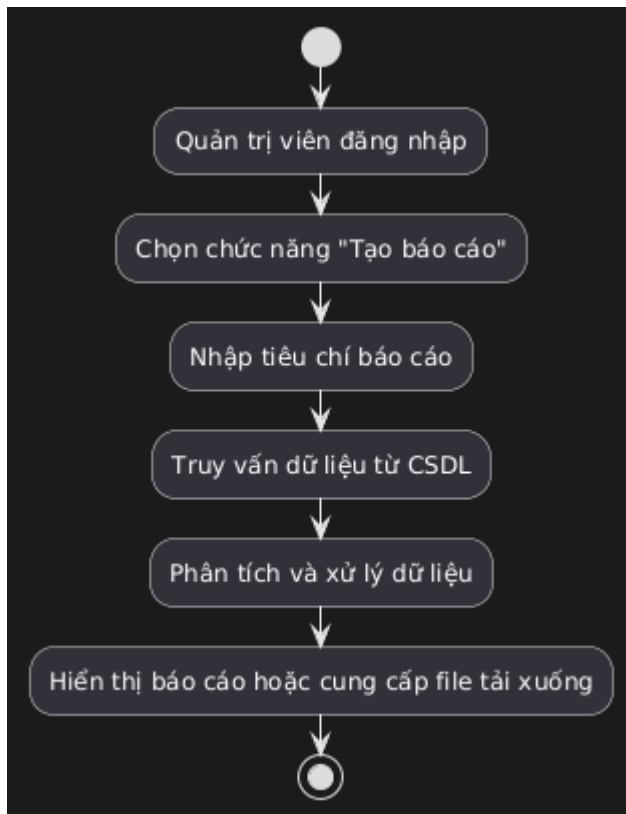
### 3.9.7. Biểu đồ hoạt động: Nhập Điểm

- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Nhập điểm".
  - Hệ thống hiển thị form nhập điểm (Mã sinh viên, Mã môn, Điểm thi).
  - Quản trị viên nhập thông tin và nhấn "Lưu".
  - Hệ thống kiểm tra dữ liệu:
    - Nếu hợp lệ: Lưu thông tin vào cơ sở dữ liệu và thông báo thành công.
    - Nếu không hợp lệ: Hiển thị lỗi và yêu cầu chỉnh sửa.



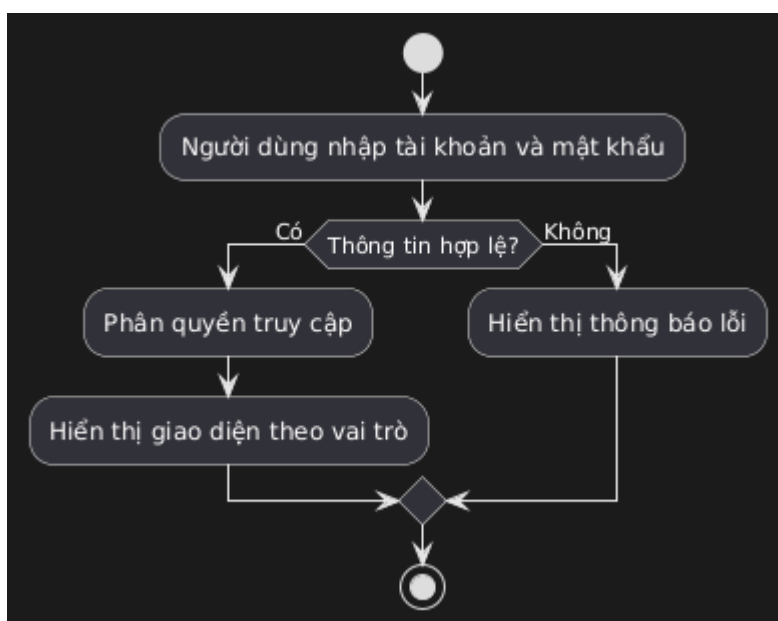
### 3.9.8. Biểu đồ hoạt động: Tạo Báo Cáo

- Mô tả quy trình:
  - Quản trị viên đăng nhập vào hệ thống.
  - Chọn chức năng "Tạo báo cáo".
  - Nhập tiêu chí báo cáo (Lớp học, Môn học, Kỳ học).
  - Hệ thống truy vấn cơ sở dữ liệu và xử lý dữ liệu.
  - Hệ thống tạo báo cáo.
  - Hiển thị báo cáo hoặc cung cấp file tải xuống.



### 3.9.9. Biểu đồ hoạt động: Đăng Nhập và Phân Quyền

- Mô tả quy trình:
  - Người dùng nhập tài khoản và mật khẩu.
  - Hệ thống kiểm tra thông tin đăng nhập:
    - Nếu hợp lệ: Phân quyền truy cập (Quản trị viên hoặc Người dùng thông thường).
    - Nếu không hợp lệ: Hiển thị thông báo lỗi.
  - Hệ thống hiển thị giao diện phù hợp với vai trò.
  -



## 3.10 Lý thuyết về Sơ đồ ERD (Entity-Relationship Diagram)

### 3.10.1. Sơ đồ ERD là gì?



- Sơ đồ ERD (Entity-Relationship Diagram) là một công cụ trực quan dùng để biểu diễn mô hình dữ liệu logic của hệ thống. Nó mô tả các thực thể (entities), thuộc tính (attributes) của các thực thể, và mối quan hệ (relationships) giữa chúng.

### 3.10.2. Thành phần chính của Sơ đồ ERD

- Thực thể (Entity):
  - Là một đối tượng, sự vật, hoặc khái niệm có thể nhận dạng được trong hệ thống.
  - Ví dụ: Sinh viên, Lớp học, Môn học.
- Thuộc tính (Attribute):
  - Là các thông tin mô tả đặc điểm của một thực thể hoặc mối quan hệ.
  - Ví dụ: Mã sinh viên, Họ tên, Ngày sinh là các thuộc tính của thực thể Sinh viên.
- Khóa chính (Primary Key):
  - Là thuộc tính hoặc tập hợp thuộc tính dùng để xác định duy nhất một thực thể.
  - Ví dụ: MaSV là khóa chính của thực thể Sinh viên.
- Mối quan hệ (Relationship):
  - Biểu diễn mối liên kết giữa các thực thể trong hệ thống.
  - Ví dụ: Sinh viên thuộc Lớp học, Môn học liên quan đến Điểm số.
- Bậc của mối quan hệ:
  - 1:1 (One-to-One): Một thực thể trong tập này liên kết với duy nhất một thực thể trong tập kia.
  - 1:n (One-to-Many): Một thực thể trong tập này liên kết với nhiều thực thể trong tập kia.
  - n:n (Many-to-Many): Nhiều thực thể trong tập này liên kết với nhiều thực thể trong tập kia.

### 3.10.3 Sơ đồ ERD cho Hệ thống Quản lý Sinh Viên

- Đây là các thực thể và mối quan hệ cần thiết:
  - Thực thể chính:
    - **Sinh viên (Student):**
      - MaSV (PK), HoTen, NgaySinh, GioiTinh, DiaChi, MaLop.
    - **Lớp học (Class):**
      - MaLop (PK), TenLop, Khoa.
    - **Môn học (Subject):**
      - MaMon (PK), TenMon, SoTinChi.
    - **Điểm (Grade):**
      - MaSV (FK), MaMon (FK), DiemThi, DiemTB.

- **Người dùng (User):**
  - Username (PK), Password, Role.

