|  |  |  |
| --- | --- | --- |
|  | **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT** | |
|  | |
| **Diagram  Description automatically generated** | |
|  |  | |
|  | |
|  | |
|  | |
| **ĐỒ ÁN TỐT NGHIỆP**  **SINH VIÊN : PHÍ THỊ QUỲNH** | |
| **NGÀNH CÔNG NGHỆ THÔNG TIN** | |
|  | |
|  | |
| **TÊN ĐỀ TÀI: XÂY DỰNG WEBSITE HỖ TRỢ QUẢN LÍ ĐỒ ÁN TÔT NGHIỆP KHOA CÔNG NGHỆ THÔNG TIN : PHÂN HỆ CHỨC NĂNG BỘ MÔN VÀ GIẢNG VIÊN** | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  | |
|  |  | |
|  | **Hà Nội - Năm 2023** | |

# LỜI CẢM ƠN

Trong vài thập kỉ trở lại đây, với sự bùng nổ của ngành công nghệ thông tin, ngành công nghệ thông tin đã mang lại cho chúng ta những thành tựu công nghệ mới cũng như ứng dụng trong mọi lĩnh vực đời sống, việc này tạo điều kiện cho sự phát triển và ra đời của thương mại điện tử, quản lí . Hơn nữa, với thời kỳ cách mạng 4.0 ngày nay bằng việc sử dụng mã nguồn mở, các nhà thiết kế, các lập trình viên đã và đang xây dựng những website quản lí, bán hàng… Với mục đích giải quyết rất nhiều những vấn đề nan giải cần xử lý của hình thức mua bán, trao đổi hàng hóa như thời gian, định mức, số lượng, quản lý…

Với sự mong muốn được tìm hiểu sâu và góp phần làm phát triển thêm hệ thống quản lí đồ án tôt nghiệp khoa công nghệ thông tin, nên em đã chọn Đồ án tốt nghiệp Công nghệ thông tin đề tài “Xây dựng website hỗ trợ quản lí đồ án tốt nghiệp khoa công nghệ thông tin : Phân hệ chức năng Bộ môn và Giảng viên”.

Trong quá trình làm đồ án tốt nghiệp em xin chân thành cảm ơn thầy Lê Hồng Anh đã hướng dẫn chỉ bảo và hướng dẫn em trong suốt quá trình thực hiện đề tài tốt nghiệp.

Trong quá trình làm đồ án tốt nghiệp do còn hạn chế vè thời gian và kinh nghiệm thực tế nên còn nhiều sai sót, em mog nhận được những góp ý chân thành từ thầy cô và các bạn

# CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI QUẢN LÝ ĐỒ ÁN SINH VIÊN KHOA CÔNG NGHỆ THÔNG TIN

Trong chương này, em trình bày tổng quan về đề tài: Mục đích và yêu cầu đặt ra khi xây dựng phần mềm quản lý đồ án sinh viên khoa Công nghệ thông tin. Quá trình khảo sát quản lý đồ án của khoa hiện nay, từ đó đưa ra phương pháp giải quyết bài toán quản lý đồ án sinh viên.

## 1.1 Mục đích, yêu cầu đặt ra khi xây dựng phần mềm quản lý đồ án.

### 1.1.1 Mục đích:

- Mục đích chính là cung cấp một hệ thống quản lý đồ án hiệu quả để lưu trữ và quản lý thông tin liên quan đến các đồ án của sinh viên, giảng viên và các bộ môn trong khoa Công nghệ thông tin. Hệ thống này giúp quản lý và tìm kiếm thông tin đồ án một cách dễ dàng và nhanh chóng.

* **Quản lí tiến độ và thời gian :**

Hệ thống quản lý đồ án có thể giúp theo dõi tiến độ và thời gian thực hiện các đồ án. Nó có thể cung cấp các công cụ để ghi nhận và theo dõi tiến độ, đánh giá và định kỳ báo cáo về tiến trình của các đồ án.

* **Quản lí tài nguyên :**

Hệ thống có thể cung cấp một nền tảng để lưu trữ và quản lí các thông tin giảng viên, sinh viên, đề tài và các thông tin liên quan đến đồ án…Điều này giúp đảm bảo tính toàn vẹn và sẵn sàng truy cập đến tài liệu liên quan đến đồ án.

* **Đánh giá và phân công :**

Hệ thống có thể hôc trọ quá trình phân công giảng viên hướng dẫn, phản biện các công việc liên quan đến đồ án. Nó có thể thống kê đánh giá kết quả của sinh viên, định rõ nhiệm vụ và phân công từng công việc cho sinh vien và giảng viên.

**1.1.2 Yêu cầu đặt ra:**

* **Giao diện người dùng than thiện:**

Hệ thống cần có giao diện người dùng than thiện, dễ sử dụng với người dùng và được tối ưu hóa để tương tác và truy cập thông tin một cách dễ dàng.

* **Tính bảo mật:**

Hệ thống cần đảm bảo tính bảo mật của thông tin đồ án, bao gồm quản lý quyền truy cập và bảo vệ dữ liệu.

## Tính mở rộng và linh hoạt:

Hệ tống cần có khả năng mở rộng và linh hoạt để có thể mở rộng các chức năng và tính năng trong tương laim đáp ứng nhu cầu thay đổi của công nghệ thông tin.

## Tính tin cậy và khả năng sao lưu:

## Hệ thống cần đảm bảo tính tin cậy và khả năng sao lưu dữ liệu quan trọng của đồ án. Điều này bao gồm việc thực hiện các biện pháp bảo vệ dữ liệu, sao lưu định kỳ và khôi phục dữ liệu khi cần thiết. Tránh việc làm mất dữ liệu.

## Tương Thích Đa Nền Tảng:

## Hỗ trợ truy cập từ nhiều thiết bị và trình duyệt khác nhau, đảm bảo tính linh hoạt và tiện ích.

## Mục đích và yeu cầu đặt ra khi xây dụng phần mềm quản lí đồ án của khoa Công nghệ thông tin là tạo ra một hệ thống hiệu quả, linh hoạt và bảo mật để quản lí thông tin, tiến độ, tài liệu và các tương tác liên quan đến đồ án trong khoa.

## 1.2 Khảo sát quy trình quản lý đồ án sinh viên khoa Công nghệ thông tin.

Xác định đề tài đồ án: Sinh viên được yêu cầu chọn hoặc đề xuất một đề tài đồ án. Thông thường, khoa sẽ cung cấp danh sách các đề tài hoặc hướng dẫn để giúp sinh viên lựa chọn.

Đăng ký đồ án: Sinh viên cần đăng ký đồ án và cung cấp thông tin chi tiết về đề tài, giảng viên hướng dẫn và nhóm nếu áp dụng.

Gửi yêu cầu xác nhận giảng viê hướng dẫn:Sing viên gửi lời mời hướng dẫn đến giảng viên để giảng viên xác nhận có đồng ý hướng dẫn hay không

+.. Giảng viên hướng dẫn có nhiệm vụ hỗ trợ sinh viên trong quá trình thực hiện đồ án, đưa ra hướng dẫn, gợi ý và theo dõi tiến độ của sinh viên.

Lập kế hoạch và thiết kế đồ án: Sinh viên cùng với giảng viên hướng dẫn sẽ lập kế hoạch chi tiết cho đồ án, bao gồm các công việc, mục tiêu, phạm vi, lịch trình và tài nguyên cần thiết. Sinh viên cũng sẽ thiết kế kiến trúc và các yếu tố kỹ thuật của đồ án.

Thực hiện đồ án: Sinh viên thực hiện các công việc theo kế hoạch đã lập. Điều này có thể bao gồm nghiên cứu, thu thập dữ liệu, phân tích, thiết kế, lập trình, kiểm thử và tài liệu hóa.

Theo dõi tiến độ: Sinh viên cần theo dõi tiến độ của đồ án và báo cáo cho giảng viên hướng dẫn. Các cuộc họp định kỳ có thể được tổ chức để đánh giá tiến trình và giải quyết các vấn đề phát sinh.

Kiểm tra và đánh giá: Sau khi hoàn thành đồ án, sinh viên sẽ tiến hành kiểm tra và đánh giá đồ án. Đánh giá có thể bao gồm việc thuyết trình, báo cáo, đánh giá kỹ thuật và đánh giá từ giảng viên hướng dẫn.

Chấm điểm phản biện : Sau khi được giáo viên hướng dẫn thông qua sinh viên sẽ nộp báo cáo đồ án lên bộ môn và được phân thầy cô phản biện để sinh viên phản biện đề tài chấm điểm cho sinh viên đó.

Chấm điểm và bảo vệ: Khoa sẽ tiến hành chấm điểm và xếp loại đồ án dựa trên các tiêu chí đã định. Nếu cần, sinh viên có thể tham gia buổi bảo vệ để giải thích và thảo luận về đồ án trước một ban giám khảo.

Lưu trữ và công bố: Đồ án hoàn thành có thể được lưu trữ trong cơ sở dữ liệu của khoa hoặc công bố công khai theo quy định của trường đại học hoặc tổ chức.

### 1.2.1 Quy trình nghiệp vụ:

* **Đăng nhập và phân quyền: (bộ môn, giảng viên)**
* **Quản lí đăng kí đợt bảo vệ :**

Bộ môn sẽ thêm mới đợt bảo vệ mới với thời gian bắt đầu và thời gian kết thúc

* **Quản lý đăng kí đồ án (Bộ môn):**

Bộ môn sẽ tạo mới sinh viên thêm sinh viên vào đợt đồ án , cập nhật đề tài hoặc thông tin sinh viên cùng giảng viên hướng dẫn

* **Phân công phản biện (Bộ môn)**

Bộ môn sẽ phân công giảng viên bất kì trong bộ môn làm giảng viên phản biện cho sinh viên và chấm điểm cho sinh viên đó.

* **Quản lý hội đồng (Bộ môn):**

Thành lập hội đồng, thêm giảng viên vào gội đồng. Cập nhập thông tin sinh viên thêm sinh viên vào hội đồng bảo vệ đồ án tốt nghiệp.

* **Thống kê (Bộ môn):**

## Danh sách hướng dẫn và phản biện theo đợt

## Danh sách sinh viên không đạt

## Thống kê thông tin giảng viên và hướng nghiên cứu

## Số lượng giảng viên hướng dẫn và phản biện của cả bộ môn

* **Tạo tài khoản đăng nhập cho giảng viên**
* **Cập nhật thông tin cá nhân, hướng nghiên cứu (Giảng viên)**
* **Xác nhận cho sinh viên HD (Giảng viên)**

Giảng viên sẽ xác nhận có đồng ý làm giảng viên hướng dẫn cho sinh viên hay không. Nếu không đồng ý thì sinh viên phải gửi lại yêu cầu đến giảng viên khác.

* **Chấm điểm PB cho sinh viên (Giảng viên):**

Giảng viên sẽ cập nhập điểm phản biện của sinh viên mình phản biện

* **Thống kê (Giảng viên)**

## Thông tin đề tài đã thực hiện theo: năm học/ nhiều năm học

## Tìm kiếm tên đề tài theo từ khóa

## Sinh viên HD/Phản biện theo năm học

### Quản lý giao đề tài (Khoa):

## Khoa sẽ duyệt hoặc không duyệt đề tài từng đợt cho từng sinh viên.

* **Lưu trữ và quản lý thông tin:**

Kết quả đồ án và tài liệu liên quan được lưu trữ một cách cẩn thận để dễ dàng truy cập và tra cứu trong tương lai.

### 1.2.2 Những khó khăn và thuận lợi trong quá trình Quản lý đồ án cho khoa Công nghệ thông tin

* **Khó khăn:**
* Hiểu rõ yêu cầu: Một trong những khó khăn ban đầu có thể là hiểu rõ yêu cầu của đề tài. Đôi khi, yêu cầu có thể mơ hồ hoặc thiếu rõ ràng, và bạn phải tự mình làm rõ để thực hiện đúng.
* Tìm hiểu kiến thức: Để thực hiện đồ án về quản lý đồ án, bạn cần hiểu về quy trình quản lý dự án, các phương pháp và công cụ quản lý dự án. Nếu bạn chưa có kiến thức sẵn có, việc tìm hiểu và áp dụng kiến thức mới có thể gặp khó khăn.
* Phân tích và thiết kế hệ thống: Để xây dựng hệ thống quản lý đồ án, bạn cần phân tích yêu cầu, thiết kế cấu trúc dữ liệu và xác định các chức năng. Đây có thể là một quá trình phức tạp và đòi hỏi kiến thức về phân tích thiết kế hệ thống.
* Triển khai và kiểm thử: Việc triển khai hệ thống và kiểm thử để đảm bảo tính ổn định và chính xác có thể gặp phải khó khăn. Có thể xảy ra các lỗi không mong muốn hoặc khó khăn trong việc tích hợp các phần mềm hoặc công nghệ khác nhau.
* **Thuận lợi:**
* Hỗ trợ từ giảng: Trong quá trình làm đồ án, bạn có thể nhờ sự hỗ trợ từ giảng viên . Thầy đã cung cấp gợi ý, chia sẻ kinh nghiệm và giúp đỡ bạn vượt qua những khó khăn trong quá trình nghiên cứu và thực hiện đồ án.
* Áp dụng thực tế: Quản lý đồ án là một lĩnh vực rất thực tế và có thể áp dụng vào nhiều ngữ cảnh khác nhau. Điều này cho phép thấy rõ ứng dụng thực tế của kiến thức và kỹ năng mà bạn đạt được trong quá trình làm đồ án.
* Quy trình đánh giá rõ ràng: Sự minh bạch trong quy trình đánh giá giúp sinh viên hiểu rõ tiêu chí và đạt đến kỳ vọng của giảng viên.
* Cơ hội học hỏi tốt: Việc thực hiện đồ án mang lại cơ hội cho sinh viên học hỏi, nâng cao kỹ năng làm việc nhóm và tự quản lý.
* Hệ thống lưu trữ đồ án hiệu quả: Sự tổ chức tốt và hệ thống lưu trữ đồ án hiệu quả giúp bảo vệ thông tin và kết quả của sinh viên.

## 1.3 Bài toán Quản lý đồ án tốt nghiệp khoa Công nghệ thông tin Trường Đại học Mỏ địa chất.

* **Bài toán:**

Tích hợp hệ thống quản lý đồ án tốt nghiệp khoa Công nghệ thông tin.

* **Mô tả:**
* Mục Tiêu:

Xây dựng một hệ thống quản lý đồ án tốt nghiệp cho Khoa Công nghệ Thông tin Trường Đại học Mỏ địa chất nhằm tối ưu hóa quá trình theo dõi, đánh giá và quản lý đồ án của sinh viên.

* Chức năng cơ bản:

- Quản lý đợt đồ án:

Thêm, sửa, xóa thông tin đợt đồ án.

- Quản lý thông tin sinh viên:

Thêm, sửa, xóa thông tin sinh viên.

- Quản lí đồ án:

Bộ môn thêm mới đồ án.

Xem và quản lý kết quả đồ án.

Thêm sửa xóa thông tin đồ án và thông tin sinh viên

Duyệt đồ án theo từng đợt.

- Quản lý thông tin giảng viên:

Thêm, sửa, xóa thông tin giảng viên.

- Giảng viên xác nhận cho sinh viên hướng dẫn

- Phân công giảng viên phản biện

- Chấm điểm phản biện cho sinh viên

- Thành lập hội đồng, nhập điểm hội đồng cho sinh viên, duyệt hội đồng

* Tính năng mở rộng :

- Tích hợp tìm kiêm và thống kê:

- Cung cấp chức năng tìm kiếm thông tin sinh viên và đồ án.

- Tạo báo cáo thống kê về tiến độ, chất lượng, và đánh giá đồ án.

* Minh bạch và theo dõi :

- Tăng cường minh bạch và theo dõi tiến độ đồ án từng sinh viên.

* Quản lí hiệu quả:

- Hỗ trợ quản lý hiệu quả các hoạt động đồ án, giảm gánh nặng công việc cho giảng viên, bộ môn, khoa và quản trị viên.

* Tiện lợi và linh hoạt:

- Cung cấp sự tiện lợi và linh hoạt trong việc quản lý thông tin và tương tác trực tuyến.

**1.4 Những ưu nhược điểm của phần mềm**

* Ưu điểm:
* Tăng hiệu quả quản lý: Phần mềm giúp tối ưu hóa quá trình quản lý thông tin đồ án, giúp tiết kiệm thời gian và công sức cho việc tổ chức và tìm kiếm thông tin.
* Tra cứu dễ dàng: Khả năng tra cứu nhanh chóng và dễ dàng giúp người dùng tìm kiếm thông tin cần thiết một cách nhanh chóng và hiệu quả.
* Theo dõi tiến độ: Phần mềm cung cấp chức năng theo dõi tiến độ thực hiện các đồ án, giúp người quản lý nắm bắt được tình hình và đưa ra quyết định kịp thời.
* Hỗ trợ hợp tác: Khả năng tương tác trực tuyến và chia sẻ thông tin giữa các thành viên đội ngũ nghiên cứu giúp tăng cường khả năng hợp tác và giao tiếp.
* Báo cáo thống kê: Phần mềm cung cấp khả năng tạo ra các báo cáo thống kê về hoạt động nghiên cứu, giúp quản lý đưa ra quyết định dựa trên dữ liệu.
* Nhược điểm:
* Phụ thuộc vào công nghệ: Phần mềm cần dựa vào công nghệ, do đó sự cố kỹ thuật hoặc sự thay đổi trong công nghệ có thể ảnh hưởng đến khả năng hoạt động của phần mềm.
* Thời gian phát triển: Xây dựng và triển khai phần mềm mất khá nhiều thời gian để phát triển và đưa vào sử dụng.
* Khả năng tương thích: Cần đảm bảo phần mềm tương thích với các hệ thống và thiết bị khác nhau để đảm bảo sự liên kết thông tin.
* Bảo mật và quyền truy cập: Đảm bảo an toàn thông tin và quản lý quyền truy cập có thể là một thách thức khá lớn.
* Khả năng mở rộng và tuỳ chỉnh: Hiện tài đề tài đang giới hạn ở một mức nhỏ nên khả năng mở rộng và tùy chỉnh sau khi đưa vào chạy thử nghiệm sẽ không tránh khỏi phát sinh ra những lỗi không mong muốn.

## 1.5 Kết luận chương 1

Chương 1 đã mở đầu cho dự án "Xây dựng website hỗ trợ quản lí đồ án tôt nhiệp khoa Công nghệ thông tin:Phân hệ chức năng Bộ môn và Giảng viên" với mục tiêu xây dựng một hệ thống hiệu quả và minh bạch, hỗ trợ quản lý đồ án sinh viên. Bước đầu tiên trong quá trình nghiên cứu này là tìm hiểu sâu rộng về các thách thức hiện tại trong quản lý đồ án và những lợi ích mà một hệ thống quản lý có thể mang lại.

# CHƯƠNG 2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Phân tích hệ thống là một công việc quan trọng và không thể thiếu được trong lập trình giải quyết bất kỳ mô hình cụ thể nào. Nó giúp người lập trình nắm được toàn bộ quy trình hoạt động của hệ thống, các mối liên quan ràng buộc, tìm ra tác nhân chính tác động (Trung tâm) của hệ thống.

Phân tích hệ thống còn giúp cho người lập trình biết được giới hạn các công việc cần thực hiện, các tác nhân tác động vào hệ thống. Để tiếp cận làm rõ từng hoạt động cụ thể ta sẽ đi vào phân tích từng hoạt động của hệ thống, các mối quan hệ của các hoạt động đó với nhau và toàn bộ hệ thống, từ đó xác định được đầu ra, đầu vào và đưa ra được mô hình thực thể chi tiết nhất.

## Phân tích hệ thống:

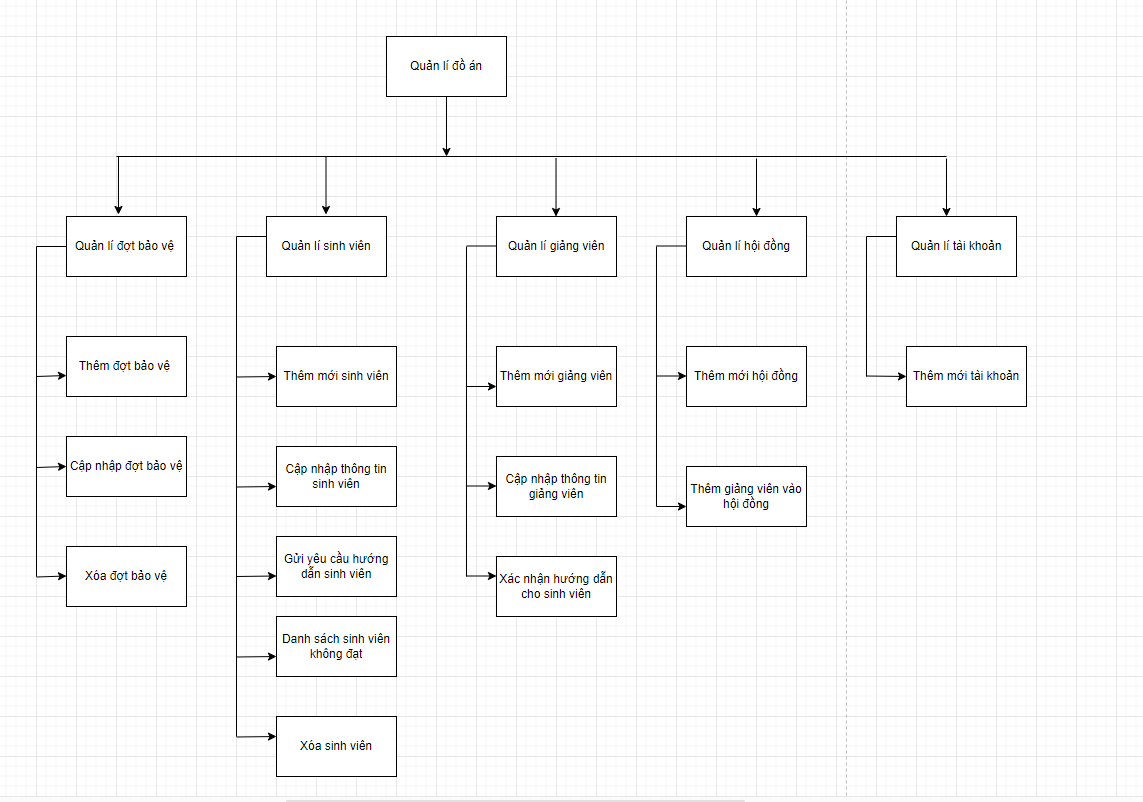
Hệ thống gồm hai nhóm chức năng chính: nhóm chức năng quản trị hệ thống bộ môn và nhóm chức năng của giảng viên. Tương ứng với hai nhóm chức năng trên hệ thống gồm hai nhóm tác nhân chính.

* **Hai nhóm tác nhân chính của hệ thống:**
* Giảng viên: Giảng viên là nhóm tác nhân chính trong hệ thống quản lý đồ án cho khoa Công nghệ thông tin. Họ là những người có trách nhiệm giảng dạy và hướng dẫn sinh viên trong quá trình thực hiện đồ án.. Đăc biệt là tác nhân này có thể duyệt danh sách sinh viên đăng ký đề tài của họ và nhận phản hồi đề xuất đề tài của sinh viên.
* Bộ môn: Bộ môn đại diện cho nhóm tác nhân chính khác trong hệ thống quản lý đồ án. Bộ môn bao gồm các thành viên của khoa Công nghệ thông tin có trách nhiệm quản lý và tổ chức các hoạt động liên quan đến quản lý đồ án trong lĩnh vực của mình. Bộ môn có thể sử dụng hệ thống để quản lý và theo dõi các đề tài nghiên cứu của sinh viên trong chuyên ngành của mình. Bộ môn có thể cung cấp hỗ trợ và hướng dẫn cho sinh viên thông qua hệ thống.

### Phân tích là thiết kế hệ thống

### Sơ đồ phân cấp chức năng của hệ thống

#### **Sơ đồ phân cấp chức năng của hệ thống:**



Hình 1: Sơ đồ phân cấp chức năng của hệ thống.

#### **2.1.1.2 Sơ đồ Giảng viên**

A diagram of a person with text

Description automatically generated

Hình 2: Sơ đồ use case giảng viên

#### **Sơ đồ Bộ môn :**

A diagram of a person with text

Description automatically generated

Hình 3 : Sơ đồ Use case bộ môn.

### Phân tích các mô hình Use Case chi tiết

### 2.2.1 Sơ đồ Use Case Chức năng đăng nhập

A diagram of a diagram

Description automatically generated

Hình 4 : Sơ đồ chức năng đăng nhập

|  |  |
| --- | --- |
| **Use case** | Đăng nhập |
| **Actor** | Giảng Viên, bộ môn |
| **Mô tả** | Mô tả việc đăng nhập vào hệ thống |
| **Điều kiện** | Actor phải có tài khoản |
| **Các bước cơ bản** | 1. Hệ thống yêu cầu actor nhập tên đăng nhập, mật khẩu.  2. Actor nhập tên đăng nhập, mật khẩu của mình và nhấn nút đăng nhập;  3. Hệ thống kiểm tra thông tin đăng nhập.  4. Hệ thống thông báo thành công và cho actor đăng nhập vào hệ thống, đồng thời phân quyền tùy theo loại |
| **Trường hợp khác** | 4.1 Tên đăng nhập hoặc mật khẩu không đúng, hệ thống hiện thông báo cho người dung và yêu cầu đăng nhập lại |
| **Trường hợp đặc biệt** | Không có |
|  |  |

2.2.1 Sơ đồ Use Case Chức năng Đăng kí

A diagram of a diagram

Description automatically generated

Hình 5: Use case cho chức năng đăng kí

|  |  |
| --- | --- |
| **Use case** | Đăng kí |
| **Actor** | Giảng Viên, bộ môn |
| **Mô tả** | Mô tả việc đăng kí vào hệ thống |
| **Điều kiện** | Actor phải tạo tài khoản |
| **Các bước cơ bản** | 1. Hệ thống yêu cầu actor nhập tên đăng nhập, mật khẩu.  2. Actor nhập tên đăng nhập, mật khẩu và nhấn nút đăng kí;  3. Hệ thống kiểm tra thông tin đăng kí.  4. Hệ thống thông báo thành công và cho actor đăng kí thành công vào hệ thống, đồng thời phân quyền tùy theo loại |
| **Trường hợp khác** | 4.1 Tên đăng nhập hoặc mật khẩu không đúng, hệ thống hiện thông báo cho người dung và yêu cầu nhập lại tên đăng nhập và mật khẩu theo đúng yêu cầu |
| **Trường hợp đặc biệt** | Không có |
|  |  |

### 2.2.2 Sơ đồ Use Case Chức năng đăng xuất.

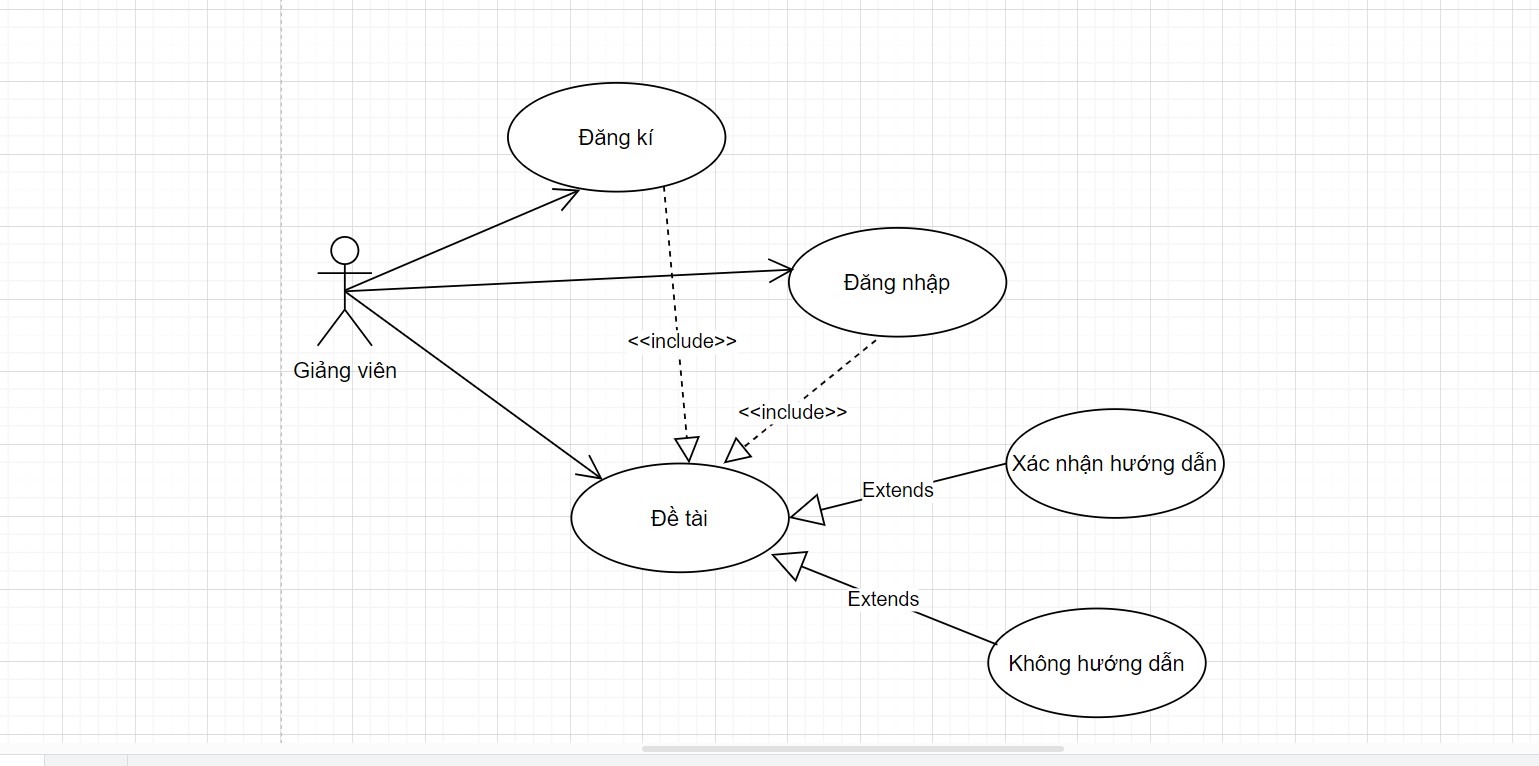
A diagram of a person with a diagram

Description automatically generated

Hình 6: Use case cho chức năng đăng xuất

|  |  |
| --- | --- |
| **Use case** | Đăng xuất |
| **Actor** | Giảng viên, bộ môn |
| **Mô tả** | Mô tả việc đăng xuất khỏi hệ thống |
| **Điều kiện trước** | Actor phải đăng nhập thành công |
| **Các bước cơ bản** | 1.Actor nhập tên đăng nhập, mật khẩu của mình và nhấn nút đăng nhập.  2.Actor chọn chức năng đăng xuất khỏi hệ thống  3.Hệ thống hiển thị yêu cầu xác nhận từ actor  4.Actor dùng xác nhận đăng xuất5.Hệ thống đăng actor khỏi hệ thống. |
| **Trường hợp khác** | 4.1. Actor không xác nhận đăng xuất thì hệ thống sẽ giữnguyên hiện trạng. |
| **Hậu điều kiện** | Đăng xuất actor khỏi hệ thống và bỏ quyển sử dụng hệthống của actor |
| **Trường hợp đặc biệt** | Không có |

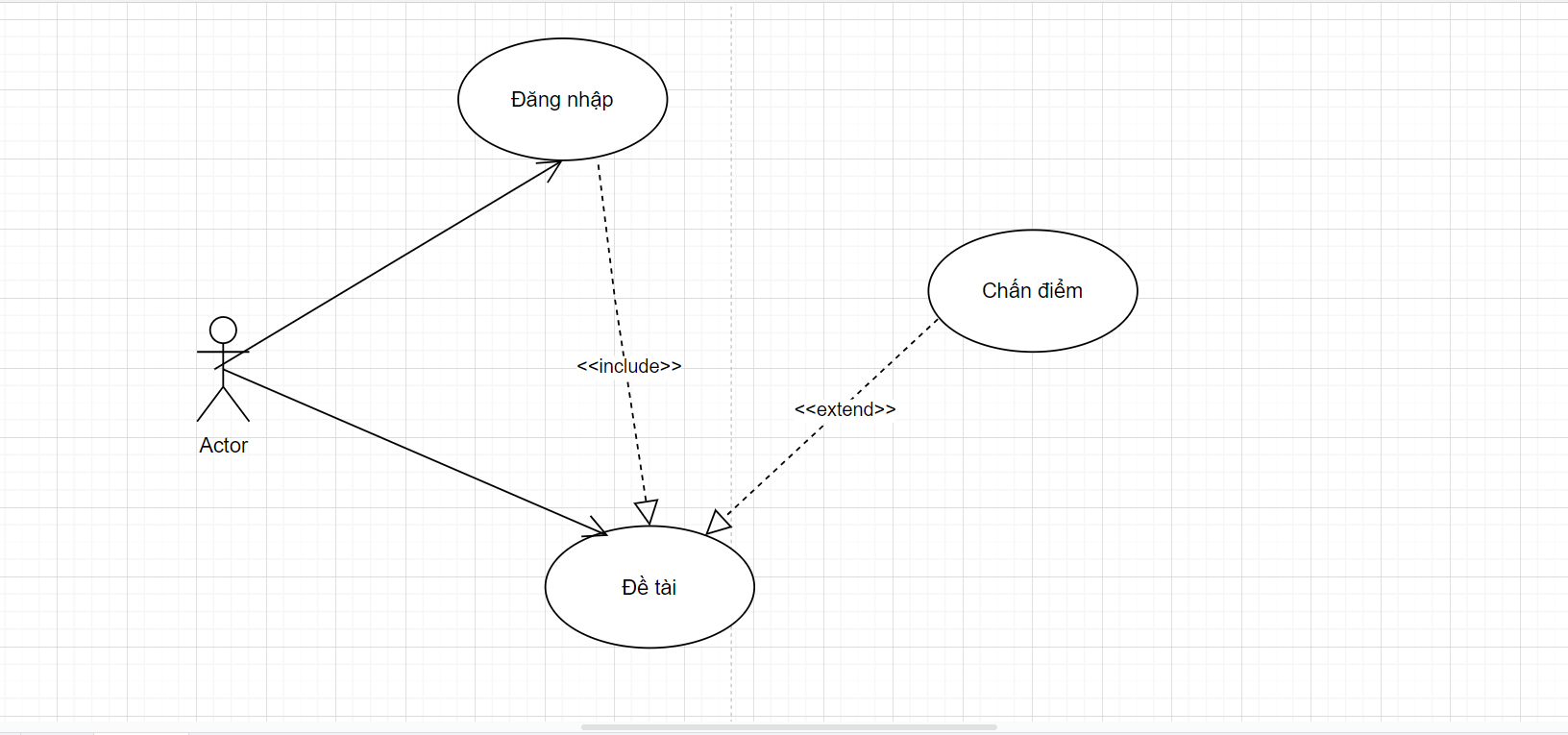
### 2.2.2 Sơ đồ Use Case Chúc năng xác nhận giảng viên hướng dẫn.



Hình 7: Use case xác nhận làm giảng viên hướng dẫn

|  |  |
| --- | --- |
| **Use case** | Xác nhận giảng viên hướng dẫn |
| **Actor** | Giảng viên |
| **Mô tả** | Use case này cho phép giảng viên xác nhận thông tin sinh viên và đề tài sinh viên đăng kí làm đồ án và đồng ý hướng dẫn sinh viên làm đồ án. |
| **Điều kiện trước** | Giảng viên phải đăng nhập vào hệ thống |
| **Các bước cơ bản** | Use case được thực hiện khi sinh viên có yêu cầu đăng ký đề tài làm đồ án và gửi yêu cầu mong muốn được giảng viên hướng dẫn làm đề tài.  1. Giảng chọn chức năng “Xác nhận hướng dẫn” sau khi đăng nhập vào hệ thống.  2. Hệ thống hiện thị bảng thông tin danh sách sinh viên gửi yêu cầu hướng dẫn.  Thông tin bao gồm:   * Số thứ tự * Tên sinh viên * Lớp * Tên đề tài * Hành động   3. Giảng viên Đăng nhập vào hệ thống và ấn button “Xác nhận hướng dẫn ”để đồng ý hướng dẫn sinh viên.Nếu giảng viên không đồng ý hướng dẫn thì ấn vào button “Hủy bỏ” để xác nhận không hướng dẫn sinh viên.  4. Hệ thống kiểm tra thông tin của sinh viên mà giảng viên vừa xác nhận và lưu vào database.  5. Kết thúc Use case |
| **Trường hợp khác** | Không có |
| **Hậu điều kiện** | Lưu thông tin xác nhận hướng dẫn sinh viên với đề tài vào hệ thống nếu use case thực hiện thành công. |
| **Trường hợp đặc biệt** | Không có |

2.2.2 Sơ đồ Use Case chức năng chấm điểm phản biện cho sinh viên :



Hình 8 : Use case chức năng chấm điểm phản biện

|  |  |
| --- | --- |
| **Use case** | Chấm điểm phản biện |
| **Actor** | Giảng viên |
| **Mô tả** | Use case này cho phép giảng viên chấm điểm phản biện cho sinh viên và đề tài sinh viên đăng kí làm đồ án . |
| **Điều kiện trước** | Giảng viên phải đăng nhập vào hệ thống |
| **Các bước cơ bản** | Use case được thực hiện khi sinh viên hết thời gian làm đồ án và phản biện với giảng viên để xác nhận có được bảo vệ đồ án không.  1. Giảng chọn chức năng “Chấm điểm phản biện” sau khi đăng nhập vào hệ thống để cho điểm và xác nhận cho sinh viên bảo vệ đồ án.  2. Hệ thống hiện thị bảng thông tin danh sách sinh viên gửi yêu cầu hướng dẫn.  Thông tin bao gồm:   * Số thứ tự * Tên sinh viên * Lớp * Tên đề tài * Điểm * Hành động   3. Giảng viên Đăng nhập vào hệ thống và ấn button “Xác nhận hướng dẫn ”để đồng ý hướng dẫn sinh viên.Nếu giảng viên không đồng ý hướng dẫn thì ấn vào button “Hủy bỏ” để xác nhận không hướng dẫn sinh viên.  4. Hệ thống kiểm tra thông tin của sinh viên mà giảng viên vừa xác nhận và lưu vào database.  5. Kết thúc Use case |
| **Trường hợp khác** | Không có |
| **Hậu điều kiện** | Lưu thông tin xác nhận hướng dẫn sinh viên với đề tài vào hệ thống nếu use case thực hiện thành công. |
| **Trường hợp đặc biệt** | Không có |

2.2.2 Sơ đồ Use Case Chức năng cập nhập thông tin cá nhân:

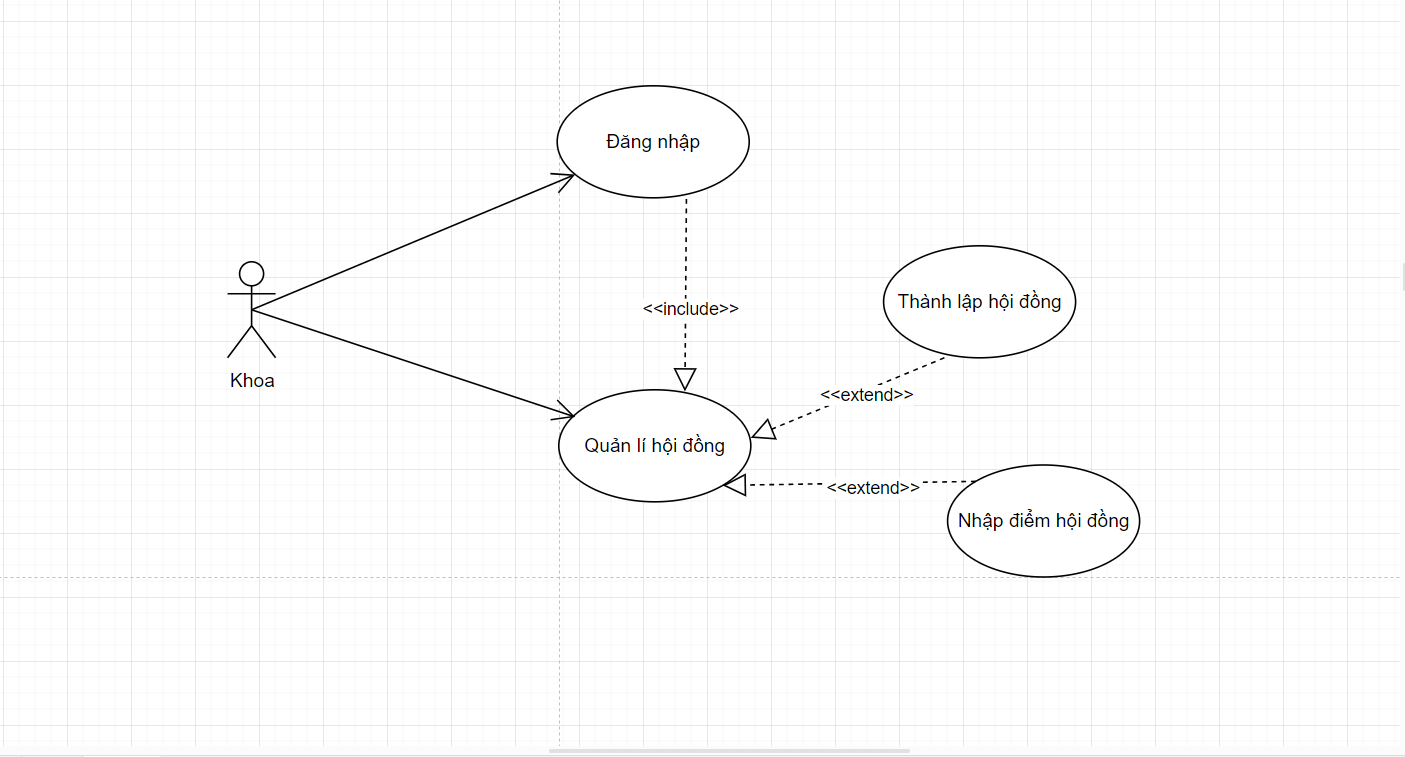
A diagram of a diagram

Description automatically generated

Hình 9 : User case cho chức năng cập nhập hông tin cá nhân giảng viên

|  |  |
| --- | --- |
| **Use case** | Cập nhập thông tin cá nhân |
| **Actor** | Giảng viên |
| **Mô tả** | Use case này cho phép giảng viên caaph nhập thông tin cá nhân và cập nhập hướng nghiên cứu. |
| **Điều kiện trước** | Giảng viên phải đăng nhập vào hệ thống |
| **Các bước cơ bản** | Use case được thực hiện khi giảng viên có như cầu cập nhập thông tin chi tiết và hướng nghiên cứu.  1. Giảng chọn chức năng “Cập nhập thông tin” sau khi đăng nhập vào hệ thống để cập nhập thông tin  2. Hệ thống hiện thị bảng thông tin cần nhập .Thông tin bao gồm:   * Tên giảng viên * Emai * Số điện thoại * Bộ môn * Mô cả hướng nghiên cứu   3. Giảng viên Đăng nhập vào hệ thống và ấn button “Cập nhập thông tin ”để câpkj nhập thông tin.  4. Hệ thống kiểm tra thông tin mà giảng viên vừa xác nhận và lưu vào database.  5. Kết thúc Use case |
| **Trường hợp khác** | Không có |
| **Hậu điều kiện** | Lưu thông tin giảng viên và database nếu thực hiện lưu thông tin thành công |
| **Trường hợp đặc biệt** | Không có |

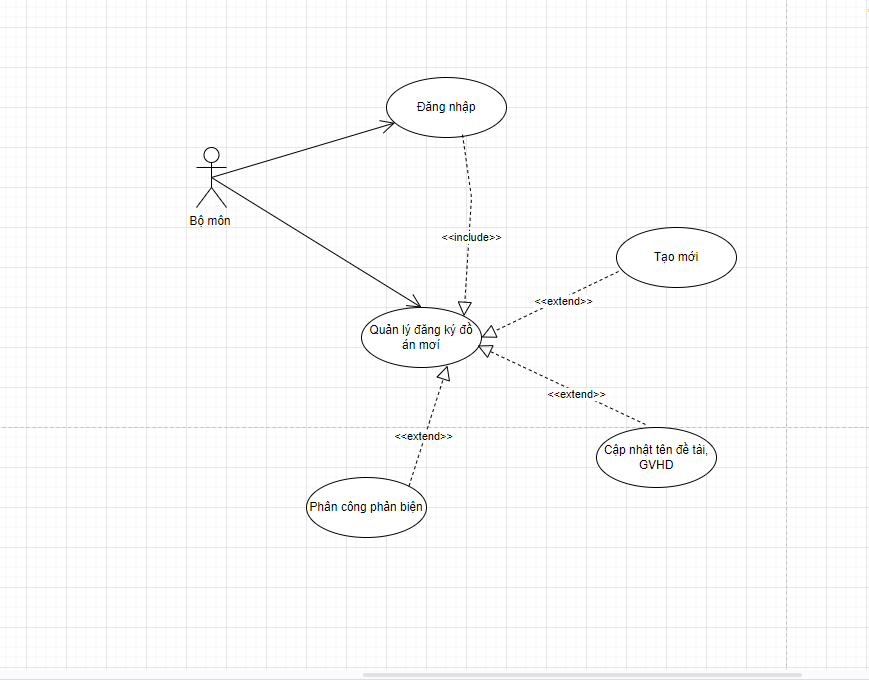
2.2.2 Sơ đồ Use Case Chức năng Quản lí hội đồng:



Hình 10 : User case cho chức năng Quản lí hội đồng

|  |  |
| --- | --- |
| **Use case** | Quản lý hội đồng |
| **Actor** | Bộ môn |
| **Mô tả** | Use case này cho phép “Bộ môn” có thể cho phép thành lập hội đồng và nhập điểm hội đồng |
| **Các bước cơ bản** | Bộ môn phải đăng nhập vào hệ thống |
| **Điều kiện trước** | Use case được thực hiện khi Bộ môn muốn thành lập hội đồng và nhập điểm  1. Bộ môn chọn chức năng “Quản lí hội đồng”   * Thành lâp hội đồng * Nhập điểm hội đồng   2. Hệ thống sẽ hiển thị danh sách các hội đồng đã có sẵn cùng với các thao tác:  3. Khi thao tác “Duyệt hội đồng” hệ thống sẽ hiện ra số lượng hội đồng cần duyệt của các bộ môn, thực hiện ấn button duyệt.  4. Khi thao tác Duyệt thì hệ thống sẽ lưu thông tin hội đồng vào database  5.Khi thao tác không duyệt hội dồng thì hội dồng sẽ không được duyệt  6. Hệ thống sẽ kiểm tra và xử lý các thông tin vừa thao tác.  7. Kết thúc Use case |
| **Trường hợp khác** | Không có |
| **Hậu điều kiện** | Lưu thông tin hội đồng và trả vào hệ thống nếu use casethực hiện thành công. |
| **Trường hợp đặc biệt** | Không có |

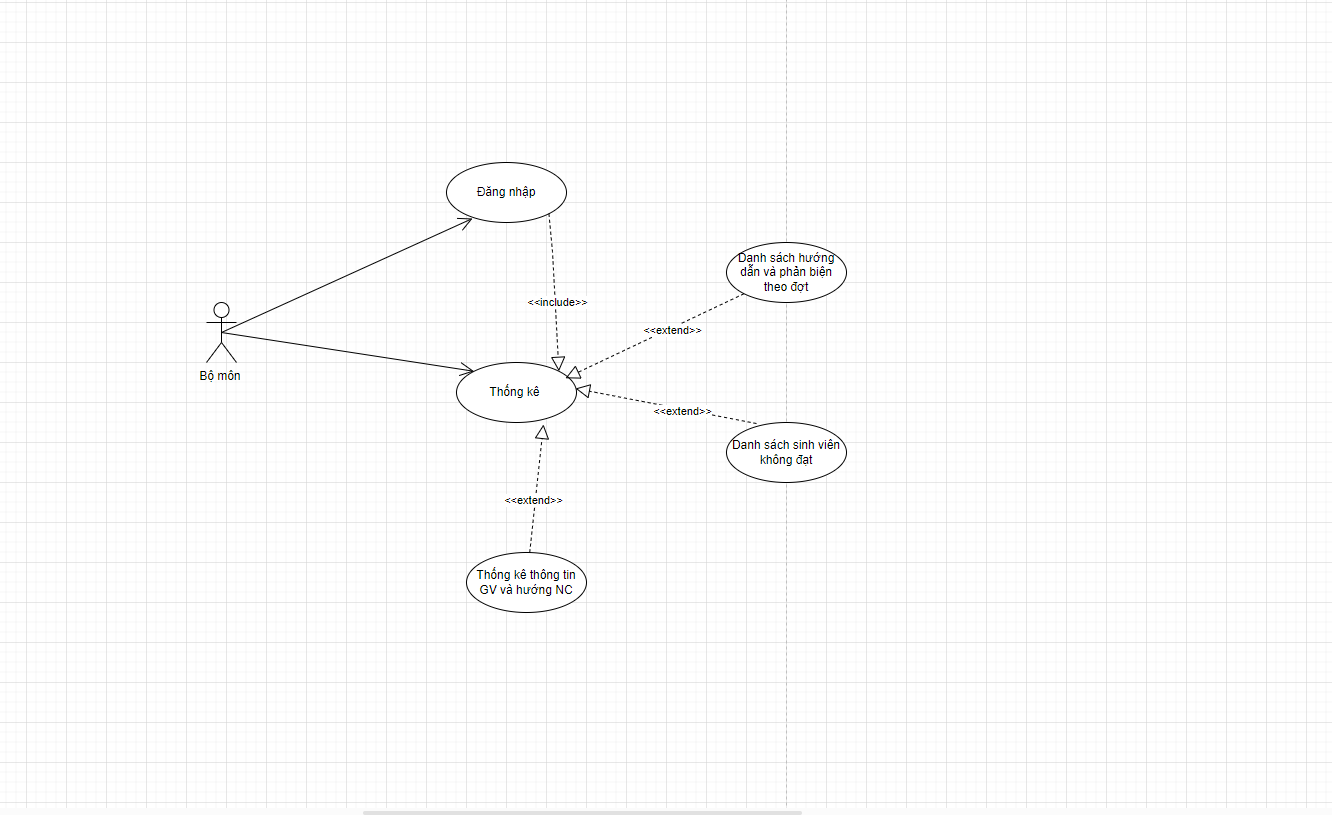
**2.2.2 Sơ đồ Use Case Chức năng Quản lí đăng kí đồ án mới:**

****

Hình 11 : Use case chức năng quản lí đăng kí đồ án mới

|  |  |
| --- | --- |
| **Use case** | Quản lý đăng kí đồ án mới |
| **Actor** | Bộ môn |
| **Mô tả** | Use case này cho phép “Bộ môn” có thể đăng kí đồ án mới |
| **Các bước cơ bản** | Bộ môn phải đăng nhập vào hệ thống |
| **Điều kiện trước** | Use case được thực hiện khi Bộ môn muốn thành lập hội đồng và nhập điểm  1. Bộ môn chọn chức năng “Quản lí đồ án”   * Tạo mới * Cập nhập tên đề tài, GVHD * Phân công phản biện   2. Hệ thống sẽ hiển thị danh sách các hội đồng đã có sẵn cùng với các thao tác:  3. Khi thao tác các chức năng hệ thống sẽ lưu vào database  4. Hệ thống sẽ kiểm tra và xử lý các thông tin vừa thao tác.  5. Kết thúc Use case |
| **Trường hợp khác** | Không có |
| **Hậu điều kiện** | Lưu thông tin đồ án và trả vào hệ thống nếu use case thực hiện thành công. |
| **Trường hợp đặc biệt** | Không có |

### 2.2.11 Sơ đồ Use Case chức năng thống kê:



Hình 13: Use case chức năng thống kê của bộ môn

|  |  |
| --- | --- |
| **Use case** | Thống kê |
| **Actor** | Bộ môn |
| **Mô tả** | Use case này cho phép “Bộ môn” có thể thống kê thông tin. |
| **Các bước cơ bản** | Bộ môn phải đăng nhập vào hệ thống |
| **Điều kiện trước** | Use case được thực hiện khi Bộ môn muốn thống kê thông tin  1. Bộ môn chọn chức năng “Thống kê”   * Danh sách hướng dẫn và phản biện theo đợt * Danh sách sinh viên không đạt. * Thống kê thông tin GV và hướng NC   2. Hệ thống sẽ hiển thị danh sách các hội đồng đã có sẵn cùng với các thao tác  3. Khi thao tác các chức năng hệ thống sẽ lưu vào database  4. Hệ thống sẽ kiểm tra và xử lý các thông tin vừa thao tác.  5. Kết thúc Use case |
| **Trường hợp khác** | Không có |
| **Hậu điều kiện** | Không có |
| **Trường hợp đặc biệt** | Không có |

### 2.2.11 Sơ đồ Use Case chức năng quản lý tài khoản :

A diagram of a company

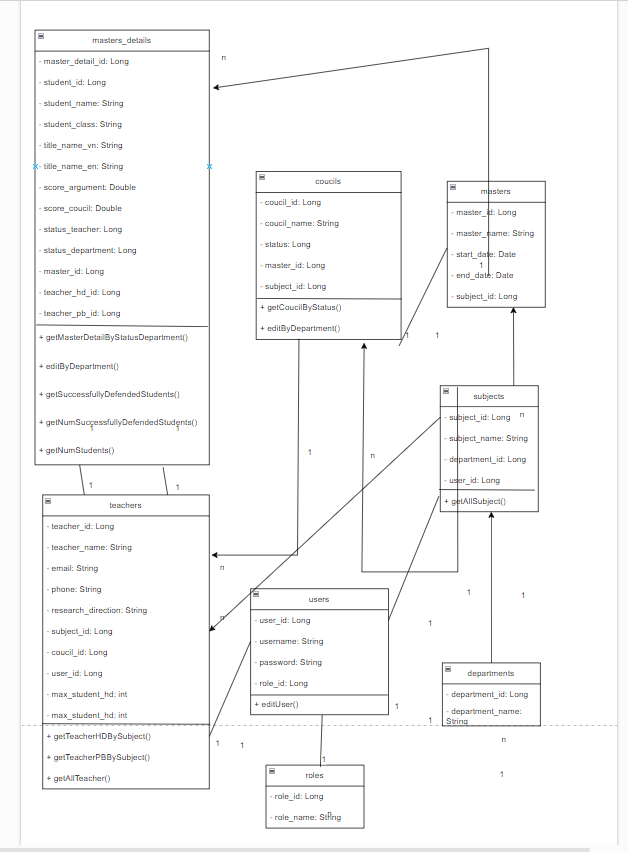
Description automatically generated

Hình 11 : Use case cho chức năng quản li tài khoản

|  |  |
| --- | --- |
| **Use case** | Quản lý tài khoản |
| **Actor** | Admin |
| **Mô tả** | Use case này cho phép Admin quản lý thông tin tài khoản của giảng viên, sinh viên, bộ môn, Khoa. |
| **Các** bước **cơ bản** | Admin phải đăng nhập vào hệ thống |
| **Điều kiện trước** | Use case này thực hiện khi Admin muốn thêm, phân quyền hoặc chỉnh sửa thông tin tài khoản của giảng viên, bộ môn.  1. Admin chọn chức năng “Quản lý Tài khoản”  2. Hệ thống sẽ hiển thị danh sách tài khoản của người dùng  3. Khi thao tác”Thêm tài khoản” sẽ hiển thị form yêu cầu nhập đầy đủ thông tin tài khoản sau đó nhấn “Ok”  4. Khi thao tác “Sửa” sẽ hiển thị form yêu cầu nhập đầyđủ thông tin mới của tài khoản sau đó nhấn “Ok”.  5. Khi thao tác “Xem chi tiết” sẽ hiển thị form chi tiết thông tin tài khoản.  6. Khi thao tác “Phân quyền” sẽ cho phép Admin phân người dùng theo loại (Giảng Viên, Bộ môn, Khoa)  7. Khi thao tác “Xóa” cho phép người dùng xóa tài khoản của giảng viên, sinh viên, bộ môn, khoa ra khỏi hệ thống.  8. Hệ thống sẽ kiểm tra và xử lý các thông tin vừa thao tác  9. Kết thúc Use Case |
| **Trường hợp khác** | 3.1. Khi điền thiếu thông tin tài khoản hệ thống sẽ hiển thị popup hiển thị thông báo lỗi.  4.1. Khi điền thiếu thông tin tài khoản hệ thống sẽ thông báo lỗi để người dùng điền đúng thông tin.  6.1. Nếu không phân quyền tài khoản hệ thống sẽ thông báo lỗi |
| **Hậu điều kiện** | Lưu thông tin vào hệ thống nếu use case thực hiện thành công. |
| **Trường hợp đặc biệt** | Không có |

## Đặc tả tiến trình của hệ thống

### 2.3.1Biểu đồ lớp



### 2.3.2Biểu đồ tuần tự

### 2.3.2.1Biểu đồ tuần tự Đăng nhập :

A blueprint with text and symbols

Description automatically generated

Hình 12 : Biểu đồ tuần tự đăng nhập

### 2.3.2.2Biểu đồ tuần tự Đăng kí:

A diagram of a diagram

Description automatically generated

Hình 13 : Biểu đồ tuần tự cho chức năng đăng kí

### 2.3.2.3 Biểu đồ tuần tự Quản lí hội đồng:

A diagram of a project

Description automatically generated

### 2.3.2.4 Biểu đồ tuần tự Quản lí đề tài:

A diagram of a graph

Description automatically generated with medium confidence

### 2.3.2.4 Biểu đồ tuần tự them mới giảng viên :

A diagram of a diagram

Description automatically generated

### 2.3.2.4 Biểu đồ tuần tự sửa thông tin giảng viên :

A diagram of a diagram

Description automatically generated

### 2.3.2.4 Biểu đồ tuần tự xóa thông tin giảng viên :

A diagram of a project

Description automatically generated with medium confidence

### 2.3.2.4 Biểu đồ tuần tự them mới bộ môn :

A diagram of a diagram

Description automatically generated with medium confidence

### 2.3.2.4 Biểu đồ tuần tự sửa thông tin bộ môn :

A diagram of a diagram

Description automatically generated

**2.3.2.4 Biểu đồ tuần tự xóa thông tin bộ môn :**

A diagram of a flowchart

Description automatically generated

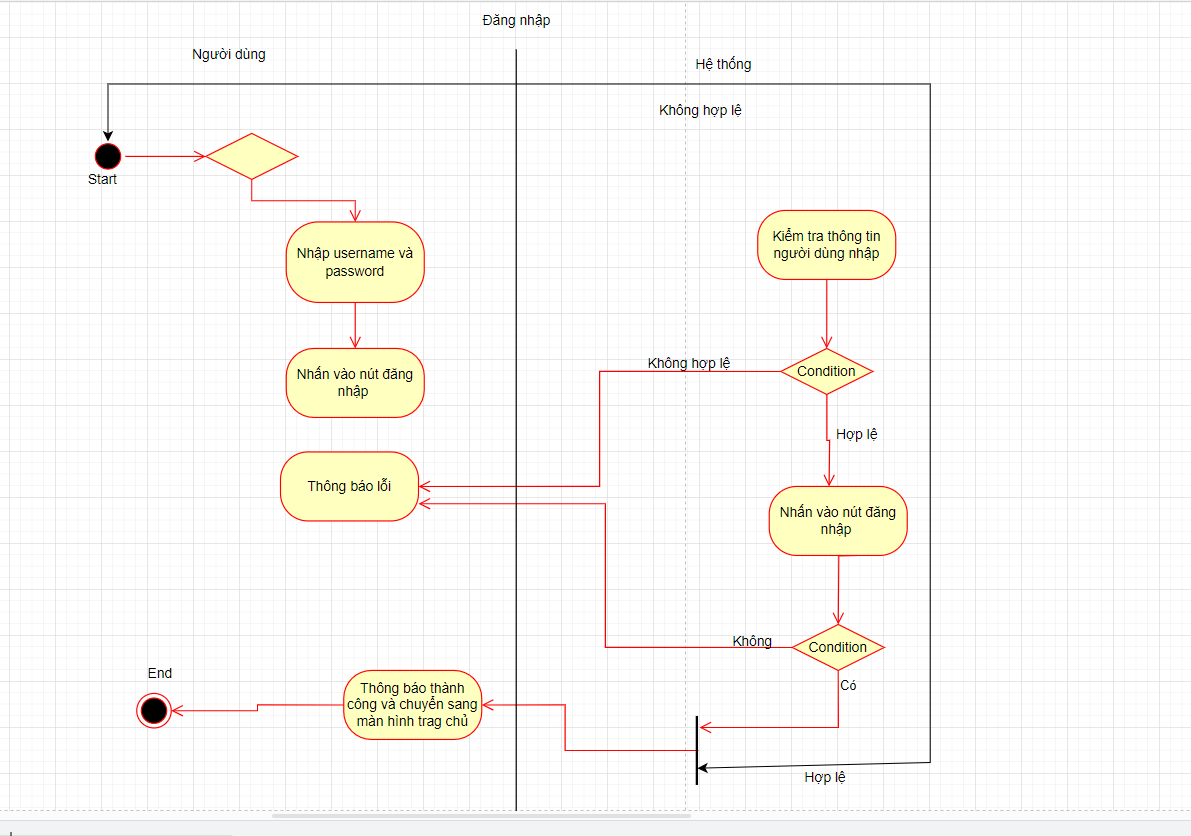
### .3.2.4 Biểu đồ tuần tự duyệt danh sách sinh viên đăng kí giảng viên hướng dẫn:

A diagram of a project

Description automatically generated

## 2.4 Xây dựng sơ đồ hoạt động

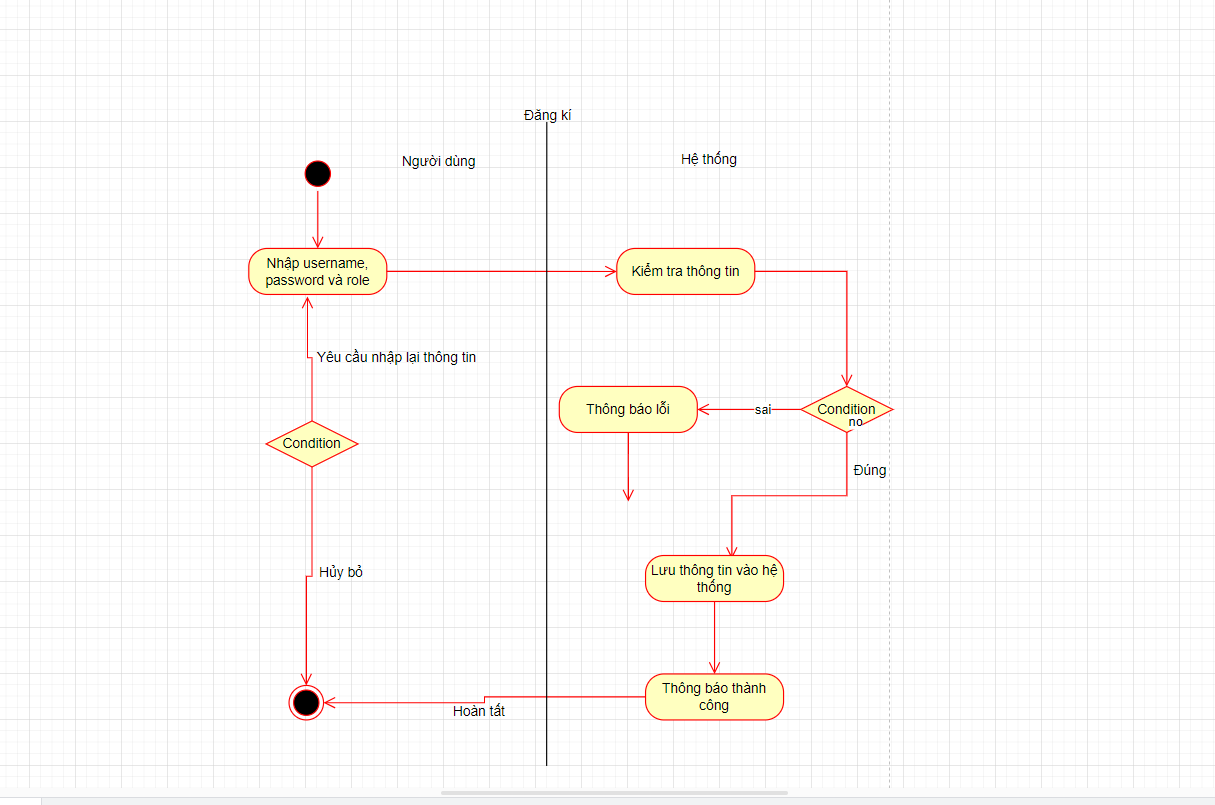
### 2.4.1 Sơ đồ hoạt động chức năng Đăng nhập



Hình 25 : Sơ đồ hoạt động chức năng đăng nhập

Hoạt động : Người dùng Đăng nhập vào hệ thống bắt buộc phải nhập đầy đủ thông tin username và password nhấn action đăng nhập thì hệ thống sẽ kiểm tra thông tin người dùng nếu nhập đúng hợp lệ yêu cầu thì sẽ kiểm tra tài khoản mật khẩu có tồn tại hay không nếu tài khoản, mật khẩu không tồn tại thì hiển thị ra message thông báo lỗi .Nêu người dùng nhập thành công thì chuyển sang màn hình trang chủ.

### 2.4.2 Sơ đồ hoạt động chức năng Đăng kí

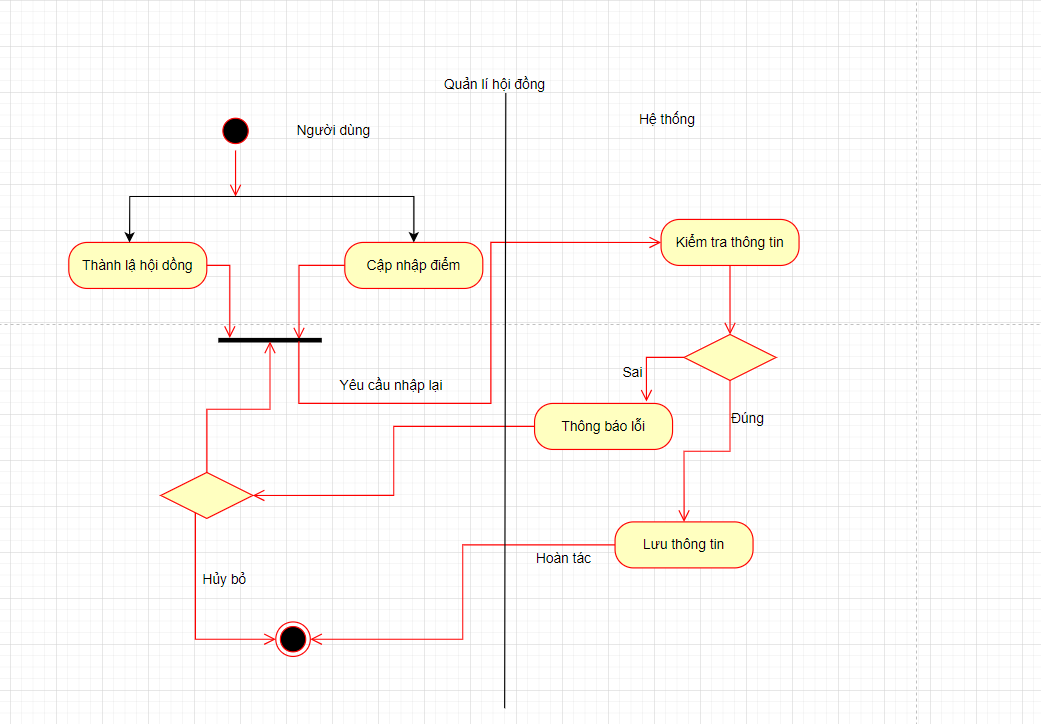


Hình 26: Sơ đồ hoạt động đăng kí

Hoạt động : Người dùng muốnvào hệ thống bắt buộc phải có tài khoản để vào được vậy phải đăng kí tài khoản người dùng cần nhập thông tin đăng kí, nhập xong hệ thống sẽ kiểm tra xem thông tin người dùng đã chính xác chưa nếu chưa chính xác thì yêu cầu nhập lại. Nếu nhập chính xác thì thông tin sẽ được lưu vào hệ thống thông báo thành công và người dùng hoàn tất quá trình đăng kí tài khoản.

Ý nghĩa : Giúp người dùng có thể đăng kí tài khoản truy cập vào hệ thống.

### 2.4.2 Sơ đồ hoạt động chức năng Quản lí hội đồng

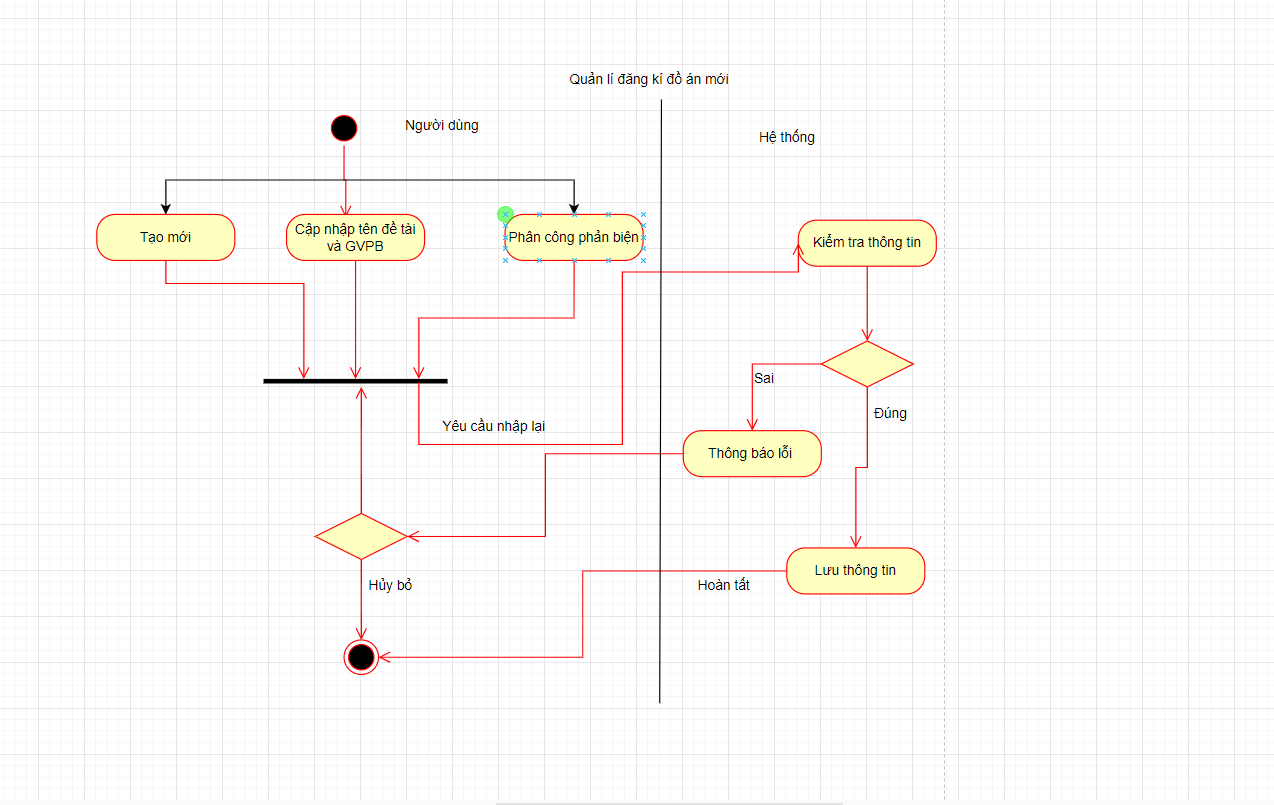


Hình 27 : Sơ đồ hoạt động chức năng Quản lí hội đồng

Hoạt động : Quản lí hội đồng có 2 chức năng chính là thành lập hội đồng và cập nhập điểm . Người dùng thao tác các yêu cầu thì hệ thống kiểm tra xem yêu cầu thêm, cập nhâp đã đúng hay chưa, nếu sai thì yêu cầu nhập lại thông tin, nếu đúng hiển thị thông tin thành công

Ý nghĩa : Giúp thành lập hội đồng và cập nhập thông tin theo đúng yêu cầu tránh sai sót.

### 2.4.2 Sơ đồ hoạt động chức năng Quản lí đăng kí đồ án mới :

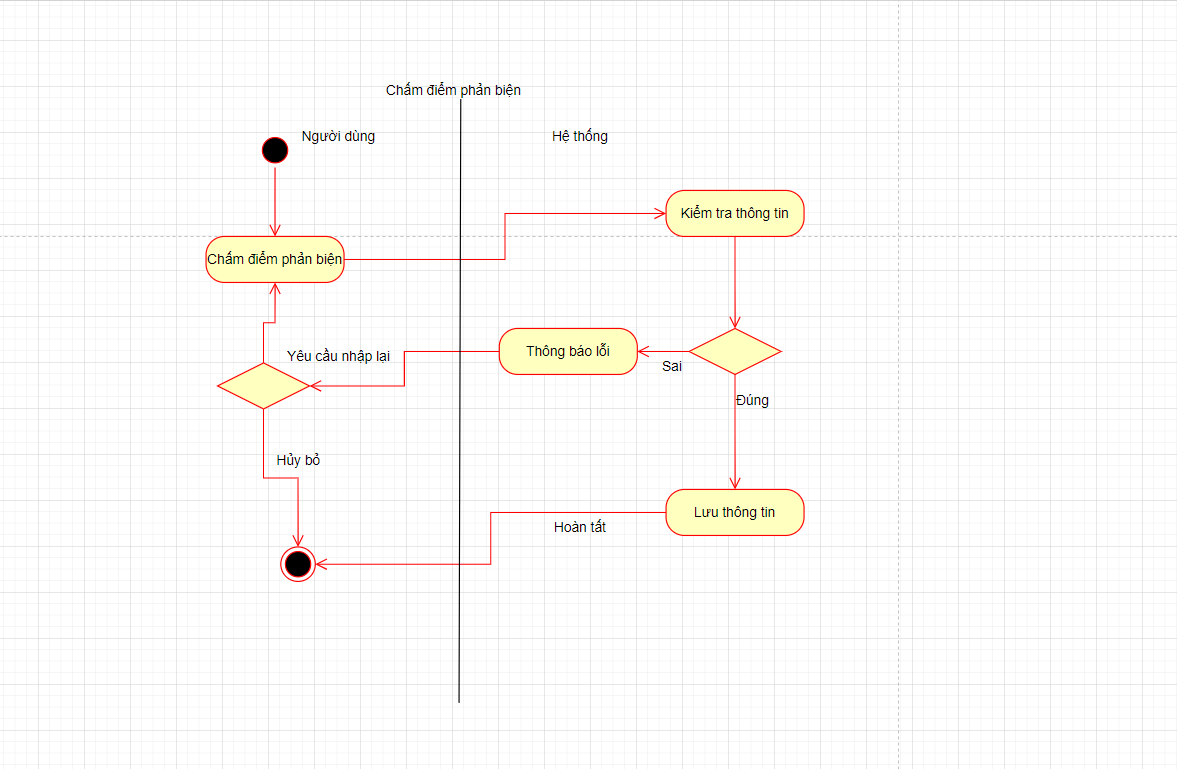


Hình 27 : Sơ đồ hoạt động quán lí đăng kí đồ án mới

Hoạt động : Quản lí đăng kí đồ án mới có 3 chức năng chính là tạo mới, cập nhập tên đề tài và giảng viên hướng dẫn, phân công phản biện. Người dùng thao tác các yêu cầu thì hệ thống kiểm tra xem yêu cầu tạo mới, cập nhập, phân công đã đúng hay chưa, nếu sai thì yêu cầu nhập lại thông tin, nếu đúng hiển thị thông tin thành công

Ý nghĩa : Giúp quản lí đăng kí đồ án tạo mới, phân công phản biện, cập nhập đề tài và GVHD nhanh tránh sai sót.

### 2.4.2 Sơ đồ hoạt động chức năng chấm điểm phản biện:



Hoạt động : Chấm điểm phản biện, Giảng viên được phân công phản biện sẽ kiểm tra phản biện đánh giá đề tài chấm điểm.Hệ thống kiểm tra thông tin nhập điểm, nếu nhập sai hiển thị thông báo lỗi nhập sai điều kiện, nếu nhập đúng hiển thị thông báo thành công

Ý nghĩa : Giúp giảng viên chấm điểm cập nhập điểm trực tiếp lên hệ thống .

## 2.5 Xây dựng cơ sở dữ liệu

### 2.5.1 Các bảng cơ sở dữ liệu :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *master\_details* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | master\_detail\_id | Long | No | PK | Khóa chính |
| 2 | score\_argument | Double | No |  | Điểm phản biện |
| 3 | score\_coucil | Double | No |  | Điểm hội đồng |
| 4 | status\_department | Nvarchar | No |  | Khoa |
| 5 | status\_teacher | Nvarchar (15) | No |  | Giảng viên |
| 6 | Student\_class | Nvarchar (20) | No |  | Lớp |
| 7 | Student\_id | Long | No |  | Má sinh viên |
| 8 | Student\_name | Nvarchar (20) | No |  | Tên sinh viên |
| 9 | Title\_name\_en | Varchar (255) | No |  | Tên đề tài tiếng anh |
| 10 | Title\_name\_vn | Nvarchar (255) | No |  | Tên đề tài tiếng việt |
| 11 | Master\_id | Long | No | FK | Mã bảng master |
| 12 | Teacher\_hd\_id | Long | No | FK | Giảng viên hướng dẫn |
| 13 | Teacher\_pb\_id | Long | No | FK | Giảng viên phản biện |

Bảng 1 : Thông tin bảng *master\_details*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *teacher* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | Teacher\_id | Long | No | PK | Mã giảng viên |
| 2 | Email | Varchar(255) | No |  | Email |
| 3 | phone | Varchar(255) | No |  | Số điện thoại |
| 4 | Research\_direction | Varchar(255) | No |  | Khoa |
| 5 | Teacher\_name | Varchar(255) | No |  | Tên giảng viên |
| 6 | Coucil\_id | Long | No | FK | Mã hội đồng |
| 7 | Subject\_id | Long | No | FK | Mã bộ môn |
| 8 | User\_id | Long | No | FK | Mã đăng nhập |
| 9 | Max\_students\_hd | Interger | No |  | Tối đa số sinh viên hướng dẫn |
| 10 | Max\_students\_pb | Interger | No |  | Tối da số sinh viên phản biện |

Bảng 2 : Thông tin bảng *teacher*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *coucils* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | Coucil\_id | Long | No | PK | Mã phản biện |
| 2 | Coucli\_name | Varchar(255) | No |  | Tên đợt phản biện |
| 3 | status | Varchar(255) | No |  | Mô tả |
| 4 | Master\_id | Long | No | FK | Mã bảng master |
| 5 | Subject\_id | Long | No | FK | Mã bộ môn |

Bảng 3 : Thông tin bảng *coucils*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *masters* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | master\_id | Long | No | PK | Mã thông tin đợt bv |
| 2 | End\_date | Datetime | No |  | Ngày kết thúc |
| 3 | Start\_date | Datetime | No |  | Ngày bắt đầu |
| 4 | Master\_name | Varchar(255) | No |  | Tên đợt |
| 5 | Subject\_id | Long | No | FK | Mã bộ môn |

Bảng 4 : Thông tin bảng *masters*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *user* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | user\_id | Long | No | PK | Mã người dùng |
| 2 | username | Varchar(255) | No |  | Tên đăng nhập |
| 3 | password | Varchar(255) | No |  | Mật khẩu |
| 4 | Role\_id | Long | No | FK | Mã người dùng |

Bảng 5 : Thông tin bảng *user*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *subjects* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | subject\_id | Long | No | PK | Mã bộ môn |
| 2 | Subject\_name | Varchar(255) | No |  | Tên bộ môn |
| 3 | Department\_id | Long | No | FK | Mã khoa |
| 4 | user\_id | Long | No | FK | Mã người dùng |

Bảng 6 : Thông tin bảng *subjects*

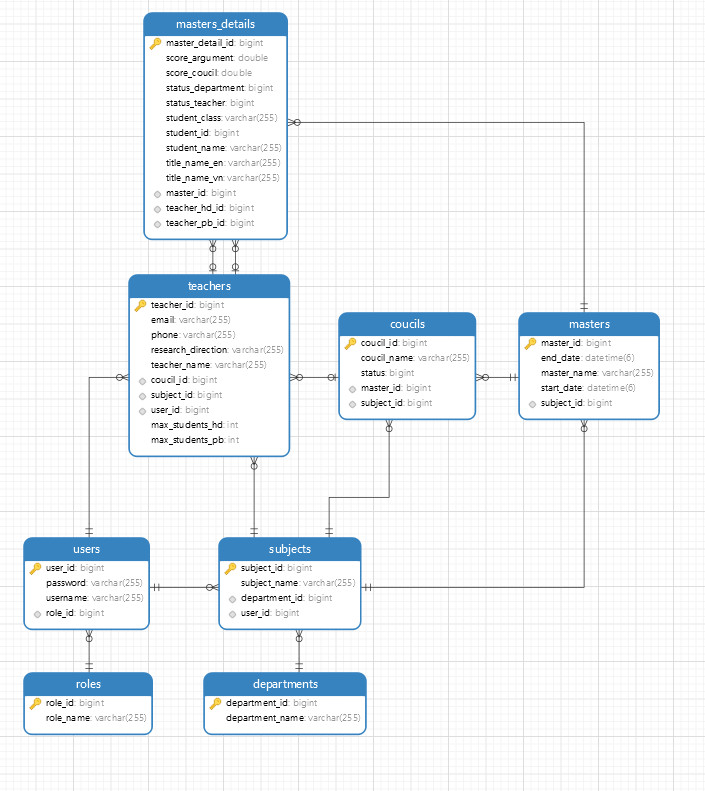
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *roles* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | role\_id | Long | No | PK | Mã phân quyền người dùng |
| 2 | role\_name | Varchar(255) | No |  | Tên phân quyền người dùng |

Bảng 7 : Thông tin bảng *roles*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng:** *departments* | | | | | |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Cho phép null** | **Khóa** | **Mô tả** |
| 1 | department\_id | Long | No | PK | Mã khoa |
| 2 | department\_name | Varchar(255) | No |  | Tên khoa |

Bảng 8 : Thông tin bảng *departments*

### 2.5.2 Lược đồ quan hệ giữa các bảng



Hình 20: Ảnh lược đồ quan hệ giữa các bảng

# CHƯƠNG 3: NGÔN NGỮ PHÁT TRIỂN HỆ THỐNG

## 3.1 Ngôn ngữ lập trình giao diện Angular.



*Hình 1: Hình ảnh ngôn ngữ lập trình Angular.*

* **Khái quát:**

Angular là một framework phát triển web được xây dựng và duy trì bởi đội ngũ của Google. Nó là một công cụ mạnh mẽ cho việc xây dựng ứng dụng web động và đa trang. Angular thường được sử dụng để phát triển các ứng dụng đơn trang (Single Page Applications - SPAs) nhờ vào khả năng tương tác cao và cấu trúc tổ chức linh hoạt.

* **Ngôn Ngữ Lập Trình:**

Angular sử dụng TypeScript, một siêu tập hợp của JavaScript, để viết mã nguồn. TypeScript cung cấp kiểu dữ liệu tĩnh và nhiều tính năng khác giúp giảm lỗi và tăng tính dễ bảo trì.

* **Cấu Trúc MVC:**

Angular tuân theo mô hình kiến trúc MVC (Model-View-Controller) hoặc chính xác hơn là MVVM (Model-View-ViewModel). Điều này giúp tạo ra ứng dụng dễ bảo trì và mở rộng.

* **Ưu điểm:**
* Kiến Trúc Mô-đun và Linh Hoạt:

Angular hỗ trợ kiến trúc mô-đun, giúp tạo ra ứng dụng dễ bảo trì và mở rộng.

* Dependency Injection (DI):

Hệ thống Dependency Injection giúp quản lý và tổ chức mã nguồn một cách hiệu quả, tăng tính linh hoạt và khả năng kiểm thử.

* Angular CLI - Tiện Ích Dòng Lệnh:

CLI giúp tạo và quản lý dự án một cách thuận tiện, tự động hóa nhiều công việc như tạo mới component, module, hoặc chạy test.

* Two-Way Data Binding:

Angular cung cấp two-way data binding giữa Model và View, giúp đồng bộ hóa dữ liệu một cách dễ dàng.

* RxJS và Observables:

Sử dụng RxJS giúp xử lý các luồng sự kiện và làm cho việc xử lý bất đồng bộ trở nên thuận lợi.

* TypeScript:

Sử dụng TypeScript giúp tăng tính rõ ràng và bảo mật trong mã nguồn.

* Hiệu Suất Cao:

Angular tối ưu hóa hiệu suất của ứng dụng thông qua các kỹ thuật như Ahead-of-Time (AOT) compilation.

* Cộng Đồng Lớn và Hỗ Trợ Mạnh Mẽ:

Angular có một cộng đồng lớn và tích cực, cung cấp nhiều tài liệu, hướng dẫn và giải pháp cho các vấn đề phổ biến.

* Bảo Mật Cao:

Angular chú trọng vào bảo mật, với nhiều tính năng như Cross-Site Scripting (XSS) và Cross-Site Request Forgery (CSRF) protection.

* **Nhược điểm:**
* Khả Năng Học Cao Ban Đầu:

Angular có một độ dốc học khá cao, đặc biệt là đối với những người mới vào lập trình web.

* Kích Thước Ứng Dụng Lớn:

Các ứng dụng Angular có thể có kích thước lớn do cần đảm bảo nhiều tính năng và thư viện.

* Phức Tạp Cho Ứng Dụng Nhỏ:

Đối với các ứng dụng nhỏ, sử dụng Angular có thể cảm thấy quá nặng nề và không hiệu quả.

* Kéo Dài Thời Gian Phát Triển Ban Đầu:

Việc triển khai Angular có thể mất nhiều thời gian hơn so với một số framework khác.

* Chưa Hoàn Toàn Tương Thích Ngược:

Các phiên bản Angular mới có thể không hoàn toàn tương thích với các phiên bản cũ, đòi hỏi công sức cập nhật khi nâng cấp.

* Phần Đánh Giá SEO Khó Khăn:

Trong môi trường SEO, Angular có thể đối mặt với một số khó khăn, đặc biệt là đối với ứng dụng động và single-page.

## 3.2 Hệ quản trị cơ sở dữ liệu MySQL.



*Hình 2: Mysql*

* **Lịch Sử và Phát Triển:**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở được phát triển bởi một công ty Thụy Điển có tên là MySQL AB, sau đó được Sun Microsystems mua lại vào năm 2008, và hiện nay là một phần của Oracle Corporation. MySQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới.

* **Mã Nguồn Mở:**

MySQL là một dự án mã nguồn mở, điều này có nghĩa là người dùng có thể sử dụng, sửa đổi và phân phối mã nguồn theo các điều khoản của Giấy phép Công cộng GNU (GPL). Điều này giúp tạo ra cộng đồng sáng tạo và tích cực quanh MySQL.

* **Đặc Điểm Chính:**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ, với sự hỗ trợ đầy đủ cho ngôn ngữ truy vấn SQL (Structured Query Language). Nó hỗ trợ nhiều hình thức lưu trữ dữ liệu như InnoDB, MyISAM, và MEMORY, mang lại sự linh hoạt cho các loại ứng dụng khác nhau.

* **Tính Nhẹ và Hiệu Quả:**

MySQL được thiết kế để làm việc hiệu quả và có hiệu suất cao trong các môi trường khác nhau, từ các dự án nhỏ đến các ứng dụng lớn và công nghiệp.

* **Hỗ Trợ Đa Nền Tảng:**

MySQL có sẵn cho nhiều nền tảng, bao gồm Linux, Windows, macOS và nhiều hệ điều hành khác. Điều này giúp cho việc triển khai và tích hợp dễ dàng trên nhiều môi trường khác nhau.

* **Hệ Thống Quyền Truy Cập:**

MySQL cung cấp một hệ thống quản lý quyền mạnh mẽ, cho phép quản trị viên kiểm soát quyền truy cập và thực hiện các biện pháp bảo mật.

* **Cộng Đồng Lớn và Hỗ Trợ Mạnh Mẽ:**

Có một cộng đồng sáng tạo và tích cực xung quanh MySQL, với nhiều tài liệu, diễn đàn trực tuyến, và nguồn lực giúp đỡ người dùng giải quyết vấn đề và tối ưu hóa sử dụng của họ.

* **Hỗ Trợ Nâng Cao:**

MySQL cung cấp nhiều tính năng nâng cao như truy vấn thời gian thực, replication, partitioning, và clustering để đáp ứng các yêu cầu phức tạp của các ứng dụng doanh nghiệp.

* **Sử Dụng Rộng Rãi:**

MySQL đã được sử dụng rộng rãi trong nhiều loại dự án và ứng dụng, từ các trang web cá nhân cho đến các hệ thống doanh nghiệp quy mô lớn.

* **Tương Thích và Mở Rộng:**

MySQL tương thích với nhiều ngôn ngữ lập trình và cung cấp các API và driver

để tương tác với nhiều loại ứng dụng.

* **Nhược Điểm của MySQL:**
* Khả Năng Mở Rộng Giới Hạn:

Trong môi trường có nhu cầu lưu trữ và truy xuất dữ liệu lớn, MySQL có thể gặp hạn chế về khả năng mở rộng so với một số hệ quản trị cơ sở dữ liệu mới.

* Chưa Hỗ Trợ Nhiều Tính Năng SQL Tiên Tiến:

So với một số hệ quản trị cơ sở dữ liệu khác, MySQL có thể không hỗ trợ một số tính năng SQL tiên tiến.

* Thiếu Các Tính Năng ACID Đầy Đủ:

Mặc dù MySQL hỗ trợ các tính năng ACID (Atomicity, Consistency, Isolation, Durability), nhưng trong một số trường hợp, nó có thể không cung cấp đầy đủ tính năng này so với các giải pháp khác.

* Quản Lý Đồng Thời Chưa Hiệu Quả Ở Một Số Tình Huống:

MySQL có thể gặp khó khăn khi quản lý đồng thời ở một số tình huống có nhiều ghi và đọc cùng lúc.

* Cập Nhật và Phát Triển Chậm Hơn Một Số Hệ Quản Trị Cơ Sở Dữ Liệu Mới:

Trong một số trường hợp, quá trình cập nhật và phát triển của MySQL có thể diễn ra chậm hơn so với một số hệ quản trị cơ sở dữ liệu mới hơn.

## 3.3 Ngôn ngữ lập trình back-end Spring boot



*Hình 3: Spring boot*

* Spring Framework:

Spring Boot là một dự án con của Spring Framework, một framework lập trình chính cho Java. Spring Framework cung cấp một cơ sở hạ tầng mạnh mẽ cho việc xây dựng các ứng dụng Java doanh nghiệp.

* Mục Tiêu Chính:

Spring Boot được phát triển với mục tiêu đơn giản hóa và tăng tốc quá trình phát triển ứng dụng Spring. Nó giúp giảm bớt cấu hình phức tạp và các tác vụ mặc định, đồng thời cung cấp các giải pháp mặc định để các vấn đề phổ biến.

* Ưu Điểm:
* Tiết Kiệm Thời Gian và Ghi Công:

Spring Boot giảm đáng kể việc cấu hình thông qua các giả mạo tự động và cung cấp các giải pháp mặc định cho nhiều vấn đề phổ biến.

* Tích Hợp Mạnh Mẽ:

Spring Boot tích hợp nhanh chóng với nhiều công nghệ và framework khác nhau, giúp giảm thời gian tích hợp và tăng tính linh hoạt.

* Tích Hợp Tốt với Spring Ecosystem:

Nó tích hợp tốt với các dự án khác của Spring như Spring Data, Spring Security, Spring Cloud, giúp xây dựng ứng dụng phức tạp một cách dễ dàng.

* Cộng Đồng Mạnh Mẽ:

Với sự hỗ trợ của cộng đồng lớn, Spring Boot có nhiều tài liệu, hướng dẫn và nguồn lực trực tuyến.

* Embedded Server:

Hỗ trợ các máy chủ nhúng như Tomcat, Jetty, Undertow, giúp triển khai ứng dụng mà không cần một máy chủ ngoại vi.

* Tự Động Cấu Hình:

Spring Boot sử dụng các giả mạo tự động để cấu hình ứng dụng, giảm độ phức tạp của các file cấu hình XML.

* Tính Năng:
* Auto-Configuration:

Tự động cấu hình dựa trên các thư viện và framework được thêm vào classpath.

* Spring Boot Starter:

Các Starter cung cấp dependencies và cấu hình mặc định cho các loại ứng dụng cụ thể.

* Spring Boot CLI:

Dòng lệnh CLI giúp phát triển Spring Boot dễ dàng hơn thông qua việc viết script và thực thi nhanh chóng.

* Spring Boot Actuator:

Cung cấp các tính năng giám sát và quản lý ứng dụng một cách dễ dàng.

* Spring Initializr:

Web-based tool giúp tạo và tùy chỉnh dự án Spring Boot một cách nhanh chóng.

* Sử Dụng Phổ Biến:

Spring Boot được sử dụng rộng rãi cho việc phát triển ứng dụng web, microservices, và các dự án Java khác.

* Nhược Điểm:
* Độ Phức Tạp Tăng Lên Với Ứng Dụng Lớn:

Dù Spring Boot giảm độ phức tạp cấu hình, nhưng với các ứng dụng lớn và phức tạp, có thể gặp phải sự khó khăn trong việc duy trì và quản lý.

* Kích Thước Ứng Dụng Lớn:

Ứng dụng Spring Boot có thể có kích thước lớn hơn so với một số framework nhẹ hơn, đặc biệt là khi sử dụng nhiều thư viện và dependencies.

* Thời Gian Khởi Động:

Việc khởi động ứng dụng Spring Boot có thể mất một khoảng thời gian, đặc biệt là khi có nhiều dependencies và phức tạp.

* Chưa Hỗ Trợ Hoàn Toàn Reactive Programming:

Mặc dù Spring Framework 5 đã hỗ trợ Reactive Programming, nhưng Spring Boot có thể không hoàn toàn tận dụng được các tính năng của nó trong mọi tình huống.

* Yêu Cầu Kiến Thức Spring Core:

Đối với những người mới bắt đầu với Spring Boot, kiến thức về Spring Core có thể là một ngưỡng khó khăn.

* Quá Nhiều Tính Năng Mặc Định:

Một số người phát triển có thể cảm thấy không thoải mái với việc sử dụng quá nhiều tính năng mặc định của Spring Boot, và họ muốn có sự linh hoạt lớn hơn trong việc tùy chỉnh.

* Kéo Dài Thời Gian Học:

Dù có nhiều tính năng giúp giảm thời gian phát triển, nhưng với những người mới học, quá trình học về Spring Boot có thể mất một khoảng thời gian đáng kể.

* Quản Lý Phiên Bản:

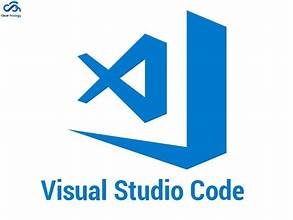
Việc quản lý phiên bản của Spring Boot và các dependencies có thể là một thách thức khi cần nâng cấp phiên bản.

## 3.4 Kết luận chương 3

Mục tiêu chính của chương giới thiệu là xác định ngữ cảnh và lý do nghiên cứu các ngôn ngữ lập trình. Chúng ta đã thấy rằng các ngôn ngữ lập trình không chỉ là một công cụ lập trình mạnh mẽ, mà còn mang đến sự đa dạng và linh hoạt trong việc giải quyết các vấn đề lập trình và phát triển ứng dụng.

# CHƯƠNG 4: CÔNG CỤ PHÁT TRIỂN HỆ THỐNG

## 4.1 Visual Studio Code



*Hình 4: VS Code*

* Visual Studio Code - Một Sự Nhìn Nhận Toàn Diện:

Visual Studio Code, thường được gọi là VS Code, là một trình soạn thảo mã nguồn mã nguồn mở và miễn phí, phát triển bởi Microsoft. Nó được thiết kế để đáp ứng nhu cầu phát triển phần mềm hiện đại và cung cấp một trải nghiệm phát triển mạnh mẽ và linh hoạt.

* Tính Năng Nổi Bật:
* Dễ Dàng Sử Dụng:

VS Code có giao diện người dùng thân thiện và dễ sử dụng, giúp những người mới bắt đầu và chuyên gia đều cảm thấy thoải mái.

* Hỗ Trợ Nhiều Ngôn Ngữ Lập Trình:

Với hỗ trợ đa dạng cho nhiều ngôn ngữ lập trình, từ JavaScript, Python, đến C# và Java, VS Code là sự lựa chọn đa năng cho các lập trình viên.

* Mở Rộng Bằng Extensions:

VS Code hỗ trợ nhiều tiện ích mở rộng, giúp tùy chỉnh và mở rộng chức năng của nó theo nhu cầu cụ thể của người dùng.

* Debugger Tích Hợp:

Debugger tích hợp mạnh mẽ giúp người phát triển dễ dàng theo dõi và sửa lỗi trong mã nguồn.

* Source Control Integration:

VS Code tích hợp chặt chẽ với các hệ thống quản lý phiên bản như Git, giúp quản lý mã nguồn dễ dàng.

* Hỗ Trợ Terminal Tích Hợp:

Terminal tích hợp giúp người phát triển thực hiện các lệnh hệ thống và quản lý dự án mà không cần chuyển sang ứng dụng bên ngoài.

* Hiệu Suất Cao và Nhẹ Nhàng:

VS Code được xây dựng với sự tập trung vào hiệu suất và tối ưu hóa tài nguyên, giúp chạy mượt mà trên nhiều hệ điều hành và cấu hình máy tính.

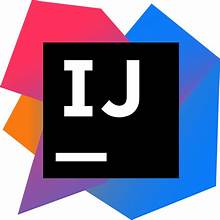
* Hỗ Trợ Cộng Đồng:

Với một cộng đồng lớn và tích cực, VS Code liên tục nhận được cập nhật, bổ sung tính năng mới và sửa lỗi, tạo nên một môi trường phát triển động và linh hoạt.

* Sử Dụng Rộng Rãi:

Visual Studio Code đã trở thành một trong những trình soạn thảo mã nguồn phổ biến nhất và được sử dụng rộng rãi trong cộng đồng phát triển phần mềm.

## 4.2 IntelliJ IDEA



*Hình 5: IntelliJ IDEA*

* IntelliJ IDEA - Môi Trường Phát Triển Toàn Diện:

IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) được phát triển bởi JetBrains. Nó nổi tiếng với sự linh hoạt và tính năng phong phú, hỗ trợ nhiều ngôn ngữ lập trình và công nghệ.

* Tính Năng Nổi Bật:
* Hỗ Trợ Nhiều Ngôn Ngữ:

IntelliJ IDEA không chỉ hỗ trợ Java, mà còn cung cấp tích hợp tốt với nhiều ngôn ngữ khác như Kotlin, Scala, Groovy, JavaScript, và nhiều ngôn ngữ khác.

* Refactoring Mạnh Mẽ:

IDE cung cấp nhiều công cụ tái cấu trúc mã nguồn (refactoring) mạnh mẽ giúp cải thiện cấu trúc và hiệu suất mã nguồn.

* Debugger và Profiler:

IntelliJ IDEA tích hợp debugger và profiler chất lượng cao, giúp dễ dàng theo dõi và tối ưu hóa mã nguồn.

* Hỗ Trợ Maven và Gradle:

Hỗ trợ tích hợp mạnh mẽ cho các công cụ quản lý dự án như Maven và Gradle, giúp quản lý các phụ thuộc và cấu hình dự án dễ dàng.

* Hỗ Trợ Version Control:

IntelliJ IDEA tích hợp tốt với hệ thống quản lý phiên bản như Git, SVN, Mercurial, cung cấp khả năng làm việc nhóm dễ dàng.

* Đồ Họa Giao Diện (GUI) Builder:

Cung cấp công cụ để xây dựng giao diện người dùng một cách nhanh chóng thông qua GUI Builder.

* Hiệu Suất Cao và Thân Thiện với Người Dùng:

IntelliJ IDEA được tối ưu hóa để chạy mượt mà và tiết kiệm tài nguyên, đồng thời có giao diện người dùng thân thiện, giúp người phát triển làm việc một cách hiệu quả.

* Hỗ Trợ Cộng Đồng:

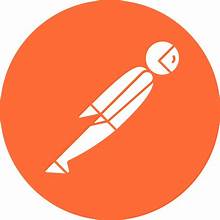
Cộng đồng IntelliJ IDEA rộng lớn, với nhiều plugin và hỗ trợ trực tuyến. JetBrains liên tục cập nhật IDE để đáp ứng nhu cầu mới và sửa lỗi.

* Sử Dụng Rộng Rãi:

IntelliJ IDEA là một trong những IDE phổ biến nhất và được sử dụng rộng rãi trong cộng đồng phát triển phần mềm, đặc biệt là trong lĩnh vực phát triển Java.

IntelliJ IDEA không chỉ là một công cụ phát triển mà còn là một nền tảng linh hoạt và mạnh mẽ, mang lại trải nghiệm lập trình chuyên nghiệp cho những người phát triển.

## 4.3 Postman



*Hình 8: Postman*

* Postman - Công Cụ Kiểm Thử và Phát Triển API:

Postman là một ứng dụng giúp kiểm thử, phát triển và quản lý API (Application Programming Interface). Nó cung cấp một giao diện người dùng đồ họa để tạo, gửi, và kiểm thử các yêu cầu HTTP và HTTPS đối với các API.

* Tính Năng Nổi Bật:
* Xây Dựng và Gửi Yêu Cầu HTTP:

Postman cho phép người sử dụng dễ dàng xây dựng và gửi các yêu cầu HTTP và HTTPS với các phương thức khác nhau như GET, POST, PUT, DELETE.

* Quản Lý Biến Môi Trường và Collection:

Hỗ trợ quản lý biến môi trường và sưu tập (collection) để tổ chức và chia sẻ các yêu cầu liên quan.

* Kiểm Thử API:

Cung cấp các tính năng kiểm thử API mạnh mẽ như kiểm thử nhanh, kiểm thử đơn vị, và kiểm thử tích hợp.

* Automate và Kiểm Thử Tự Động:

Postman hỗ trợ tự động hóa các bước kiểm thử và cho phép tạo các bộ kiểm thử tự động.

* Giao Diện Người Dùng Thân Thiện:

Giao diện đồ họa thân thiện và dễ sử dụng giúp người phát triển và kiểm thử viên tận dụng hết các tính năng.

* Đồ Họa Hóa Dữ Liệu:

Cho phép đồ họa hóa dữ liệu trả về từ các yêu cầu API, giúp người sử dụng dễ dàng hiểu và xử lý dữ liệu.

* Chia Sẻ và Hợp Tác:

Hỗ trợ chia sẻ các bộ kiểm thử và yêu cầu thông qua link chia sẻ, giúp tăng cường hợp tác trong đội ngũ phát triển.

* Sử Dụng Đa Nền Tảng:

Postman có phiên bản trực tuyến và ứng dụng desktop, hỗ trợ nhiều nền tảng như Windows, macOS và Linux.

* Hỗ Trợ Đa Ngôn Ngữ:

Giao diện của Postman được hỗ trợ trong nhiều ngôn ngữ khác nhau, giúp người dùng từ các quốc gia và khu vực khác nhau sử dụng dễ dàng.

## 4.6 MySQL Workbench

## IMG_256

*Hình 9: MySQL Workbench*

* Thiết Kế Cơ Sở Dữ Liệu:

ER Modeling: Cho phép bạn tạo và quản lý mô hình thực thể-quan hệ (ER) một cách trực quan.

Tạo Bảng và Quan hệ: Dễ dàng tạo và quản lý các bảng, quan hệ và các ràng buộc khóa.

* Thực Thi Truy Vấn SQL:

Query Editor: Cung cấp một trình soạn thảo SQL mạnh mẽ với các tính năng như kiểm tra cú pháp, hoàn thành từ khóa và định dạng mã nguồn.

Execution Plan: Hiển thị kế hoạch thực hiện truy vấn để tối ưu hóa hiệu suất.

* Quản Lý Người Dùng và Quyền Truy Cập:

User Administration: Cho phép tạo, xóa và quản lý người dùng và quyền truy cập một cách dễ dàng.

Role Management: Hỗ trợ quản lý vai trò để đơn giản hóa việc quản lý quyền truy cập.

* Định Dạng và Xuất Dữ Liệu:

Export/Import Data: Cung cấp chức năng xuất và nhập dữ liệu từ và vào nhiều định dạng khác nhau như CSV, SQL, JSON.

Data Dump: Tạo các bản sao dự phòng hoặc dump dữ liệu một cách nhanh chóng.

* Quản Lý Kết Nối và Server:

Connection Management: Dễ dàng quản lý và kết nối đến các máy chủ MySQL khác nhau.

Server Administration: Cung cấp các công cụ quản lý máy chủ như kiểm tra trạng thái, quản lý thiết lập, và kiểm tra logs.

* Visualization Tools:

Database Relationship Graphs: Hiển thị mối quan hệ giữa các bảng trong một biểu đồ.

Performance Dashboard: Đánh giá và theo dõi hiệu suất của cơ sở dữ liệu.

* SQL Code Snippets:

Cung cấp một thư viện các đoạn mã SQL mẫu để tăng tốc quá trình phát triển.

* Công Cụ Điều Khiển Phiên Bản:

Hỗ trợ tích hợp với các hệ thống quản lý phiên bản như Git để theo dõi các thay đổi trong cơ sở dữ liệu.

**CHƯƠNG 5 : GIAO DIỆN WEBSITE QUẢN LÍ ĐỒ ÁN TỐT NGHIỆP KHOA CÔNG NGHỆ THÔNG TIN**