**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

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

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

NGÀNH KỸ THUẬT PHẦN MỀM

**XÂY DỰNG HỆ THỐNG HỌC TRỰC TUYẾN HAPOLEARN**

**CBHD :** ThS. Ngô Thị Bích Thúy

**Sinh viên :** Nguyễn Thọ Duy

**Mã số sinh viên :** 2018600025

Hà Nội – 2023

**LỜI CẢM ƠN**

Đầu tiên, em xin gửi lời cảm ơn chân thành đến Trường Đại Học Công Nghiệp Hà Nội và khoa Công nghệ thông tin đã đưa học phần đồ án tốt nghiệp vào chương trình giảng dạy. Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn – cô Ngô Thị Bích Thúy đã dạy dỗ, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua. Trong thời gian học tập em đã được truyền đạt những kiến thức chuyên ngành hết sức quý báu và bổ ích và đó sẽ là hành trang để em có thể vững bước trên con đường tương lai.

Học phần Đồ án tốt nghiệp là học phần thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Em đã cố gắng nghiên cứu một cách nghiêm túc và kỹ lưỡng để có thể hoàn thành bài nghiên cứu thật chi tiết và hoàn chỉnh. Tuy nhiên do vốn kiến thức còn hạn nên bài nghiên cứu khó có thể tránh khỏi những thiếu sót cũng như chưa chi tiết hoàn chỉnh vậy nên kính mong thầy giáo xem xét và góp ý để bài tiểu luận của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

[Hà Nội – 2023 0](#_Toc132491174)

[**LỜI CẢM ƠN** i](#_Toc132491175)

[MỞ ĐẦU 1](#_Toc132491176)

[1. Lý do chọn đề tài 1](#_Toc132491177)

[2. Mục tiêu đề tài 1](#_Toc132491178)

[3. Đối tượng và phạm vi nghiên cứu 1](#_Toc132491179)

[4. Ý nghĩa khoa học, thực tiễn và kết quả mong muốn của đề tài 2](#_Toc132491180)

[Chương 1 – KHẢO SÁT YÊU CẦU HỆ THỐNG 3](#_Toc132491181)

[1.1. Khảo sát sơ bộ hệ thống 3](#_Toc132491182)

[1.2. Xác định yêu cầu hệ thống 3](#_Toc132491183)

[1.3. Khảo sát chi tiết hệ thống 4](#_Toc132491184)

[Chương 2 – MÔ TẢ CÁC YÊU CẦU CHỨC NĂNG HỆ THỐNG 5](#_Toc132491185)

[2.1 Biểu đồ use case 5](#_Toc132491186)

[2.1.1. Các use case chính 5](#_Toc132491187)

[2.2 Mô tả chi tiết các use case 6](#_Toc132491188)

[2.2.1 Đăng nhập 6](#_Toc132491189)

[2.2.2 Quản lí sinh viên 6](#_Toc132491190)

[2.2.3 Quản lí khóa học 7](#_Toc132491191)

[2.2.4. Quản lí bài học 9](#_Toc132491192)

[2.2.5. Tham gia khóa học 10](#_Toc132491193)

[2.2.6. Xem bài học 11](#_Toc132491194)

[2.2.7. Đánh giá bài học 12](#_Toc132491195)

[2.2.8. Đăng kí 12](#_Toc132491196)

[2.2.9. Quên mật khẩu 13](#_Toc132491197)

[2.2.10. Gửi mail chăm sóc học viên 14](#_Toc132491198)

[Chương 3 THIẾT KẾ CƠ SỞ DỮ LIỆU 15](#_Toc132491199)

[Biểu đồ quan hệ cơ sở dữ liệu 15](#_Toc132491200)

[Thiết kế bảng 16](#_Toc132491201)

[Bảng users 16](#_Toc132491202)

[Bảng courses 16](#_Toc132491203)

[Bảng lessons 17](#_Toc132491204)

[3.2.4. Bảng reviews 17](#_Toc132491205)

[3.2.5. Bảng programs 18](#_Toc132491206)

[Chương 4 PHÂN TÍCH VÀ THIẾT KẾ CHỨC NĂNG 19](#_Toc132491207)

[4.1. Use case Đăng nhập 19](#_Toc132491208)

[4.1.1. Biểu đồ trình tự 19](#_Toc132491209)

[4.1.2. Biểu đồ lớp 20](#_Toc132491210)

[4.2. Use case Quản lí khóa học 21](#_Toc132491211)

[4.2.1. Biểu đồ trình tự 21](#_Toc132491212)

[4.2.2. Biều đồ lớp 22](#_Toc132491213)

[4.3. Use case Quản lí bài học 23](#_Toc132491214)

[4.3.1. Biểu đồ trình tự 23](#_Toc132491215)

[4.3.2. Biểu đồ lớp 24](#_Toc132491216)

[4.4. Use case Quản lí học viên 25](#_Toc132491217)

[4.4.1. Biều đồ trình tự 25](#_Toc132491218)

[4.4.2. Biểu đồ lớp 26](#_Toc132491219)

[4.5. Use case Tham gia khóa học 27](#_Toc132491220)

[4.6. Use case Đánh giá bài học 28](#_Toc132491221)

[4.6.1. Biểu đồ trình tự 28](#_Toc132491222)

[4.7. Use case tạo nội dung bài học 29](#_Toc132491223)

[4.7.1. Biểu đồ trình tự 29](#_Toc132491224)

DANH MỤC HÌNH

# MỞ ĐẦU

## Lý do chọn đề tài

Ngày nay, cuộc sống xã hội ngày càng phát triển, hiện đại và tiến bộ

hơn, đòi hỏi con người phải có đầy đủ kiến thức, kỹ năng để hội nhập, giúp

ích cho bản thân, gia đình và xã hội. Từ đó, vai trò của việc học luôn được đề

cao và chú trọng. Việc học ngày càng quan trọng thì cách tiếp cận việc học

cũng quan trọng không kém. Nhất là trong thời buổi hiện đại, con người có

thể tiếp cận việc học với nhiều cách học khác nhau. Trong đó, không thể

không nhắc đến phương pháp “học online” đang phát triển và phổ biến trên

thế giới. Chỉ cần một chiếc laptop hay điện thoại có kết nối Internet, người

học hoàn toàn có thể học tập bất kì nơi đâu. Đây là một cách học nhanh

chóng, dễ dàng, thuận tiện và tiết kiệm. Chính bởi thấy được tầm quan trọng

và những ưu điểm vượt trội của việc “học online”, em chọn đề tài này để giải quyết những nhu cầu học tập đối với đối tượng học viên

## Mục tiêu đề tài

* Xây dựng các chức năng cơ bản của một hệ thông học trực tuyến.
* Website có khả năng tự tương thích, hiển thị được trên tất cả các thiết bị hiện tại và có thể nâng cấp trong tương lai.
* Website đáp ứng được nhu cầu học online và quản lí.
* Nắm bắt được công nghệ thiết kế web bằng PHP, MySQL, HTML, JAVASCRIPT, CSS, v.v.
* Ứng dụng mô hình MVC vào xây dựng trang web.

1. **Đối tượng và phạm vi nghiên cứu**

Hệ thống website này được xây dựng cho trung tâm giáo dục HapoLearn.

1. **Ý nghĩa khoa học, thực tiễn và kết quả mong muốn của đề tài**

* Ý nghĩa khoa học thực tiễn:
* Website được xây dựng đã áp dụng thành tựu của công nghệ thông tin vào lĩnh vực giáo dục.
* Sự ra đời của website này không chỉ giúp cho việc học trực tuyến một cách dễ dàng nhanh chóng mà còn dễ dàng trong việc quản lý, dễ dàng quảng bá các dịch vụ của mình đến với học viên.
* Kết quả mong muốn của đề tài:
* Xây dựng thành công hệ thống học trực tuyến Hapolearn.
* Nắm bắt được các kiến thức về HTML, CSS, PHP, MySQL, v.v.

# Chương 1 – KHẢO SÁT YÊU CẦU HỆ THỐNG

## Khảo sát sơ bộ hệ thống

* + 1. **Mục đích**
* Tìm hiểu cách thức vận hành của trung tâm:
* Tìm hiểu được các điểm thuận lợi, khó khăn trong giảng dạy.
* Tìm hiểu được các điểm thuận lợi, khó khăn trong quản lí.
* Tìm hiểu được thông tin về trung tâm, học viên:
* Các khóa học của trung tâm.
* Số lượng học viên của trung tâm.
* Mong muốn của trung tâm trong quá trình giảng dạy.
* Hoạt động của website:
* Lượng truy cập vào website.
* Hiệu suất của website.
  + 1. **Phương pháp**
* Điều tra quá trình giảng dạy và học tập của giáo viên và học viên.
* Phỏng vấn, nắm bắt yêu cầu của trung tâm, nắm bắt khó khăn của giáo viên và học viên.
  + 1. **Đối tượng khảo sát**
* Quản lí trung tâm giáo dục Hapo
* Giáo viên và học viên trung tâm Hapo
  + 1. **Kết quả khảo sát sơ bộ**
* Trung tâm mong muốn học viên ngoài việc học ở trên lớp thì cần bổ sung thêm quá trình học ở nhà và cung cấp thêm các khóa học và tài liệu hữu ích dành cho học viên.

## Xác định yêu cầu hệ thống

* + 1. Yêu cầu chức năng.
* Đối với quản lí/giáo viên:
  + Xem thông tin học viên.
  + Xem quá trình học của học viên.
  + Thêm mới học viên.
  + Chỉnh sửa, bổ sung thông tin cho học viên.
  + Tạo các khóa học mới.
  + Tạo các bài học mới.
  + Chỉnh sửa và bổ sung bài học.
  + Đăng nhập, đăng kí.
  + Gửi mail chăm sóc học viên.
* Đối với học viên:
  + Đăng nhập, đăng kí, quên mật khẩu.
  + Xem khóa học, sử dụng khóa học.
  + Xem bài học, tài liệu học.
  + Xem quá trình học tập.
  + Bình luận, khảo sát về khóa học.
    1. Yêu cầu phi chức năng.
* Hiệu năng tốt, thời gian phản hồi nhanh.
* Chịu tải tốt, đáp ứng nhiều lượt truy cập.
* Giao diện đẹp, dễ dùng.
* Khả năng bảo trì tốt, dễ dàng quản lí.

## Khảo sát chi tiết hệ thống

* + 1. **Cách thức vận hành trung tâm.**

Học viên sau khi học trực tiếp tại trung tâm sẽ được giáo viên giao bài tập về nhà cùng với tài liệu. Giáo viên sau khi dạy xong sẽ phải ghi chép thông tin đánh giá học viên và gửi lên trung tâm.

* + 1. **Báo cáo, thống kê**

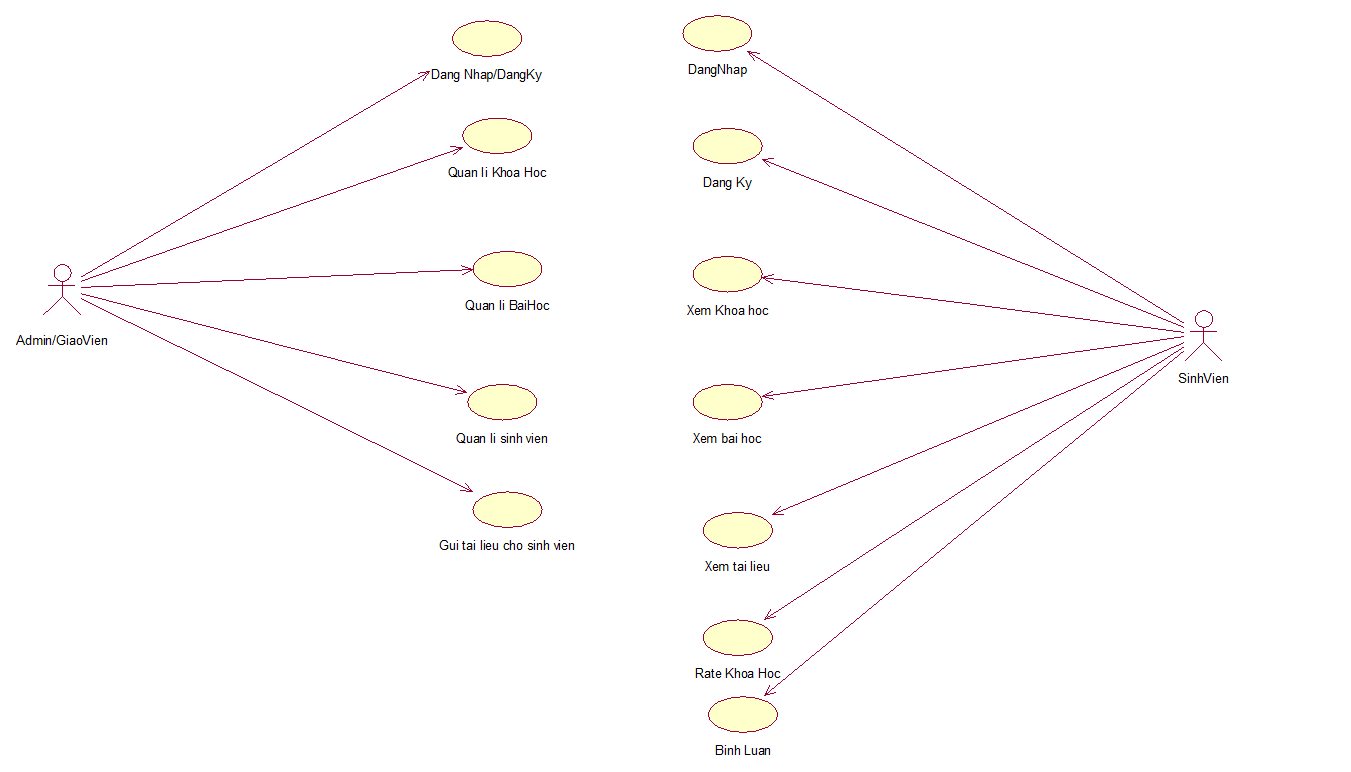
Trung tâm sẽ thống kê số lượng học viên tham gia khóa học.

Ngoài ra sẽ thống kê được quá trình học tập của học viên, có tham gia học đầy đủ và hoàn thành bài tập trước, sau khi lên lớp không. Từ đó trung tâm sẽ có thêm thông tin để hỗ trợ học viên khi gặp vấn đề.

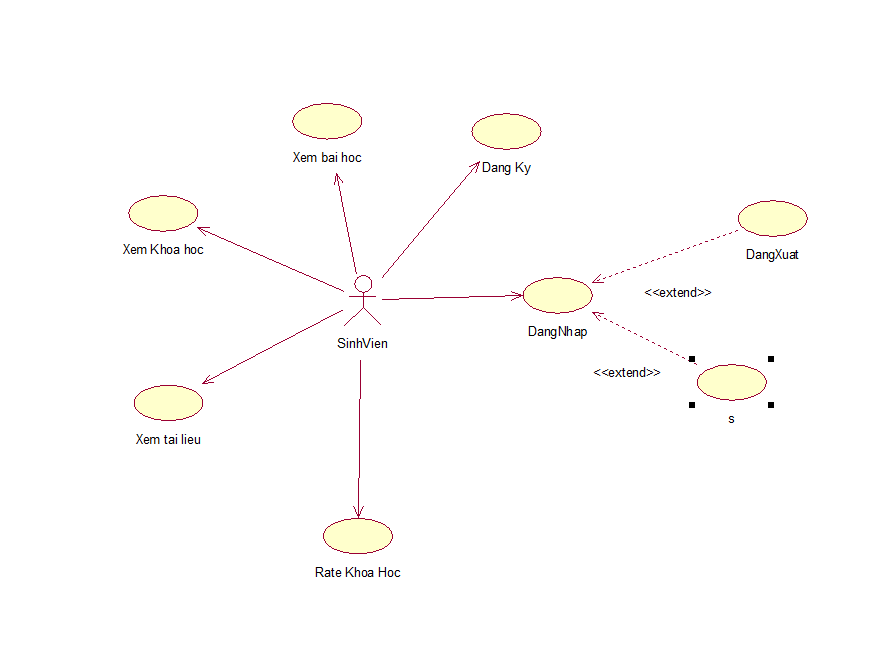
# Chương 2 – MÔ TẢ CÁC YÊU CẦU CHỨC NĂNG HỆ THỐNG

## 2.1 Biểu đồ use case

### 2.1.1. Các use case chính



Hình 2.1 Use case tổng quát



Hình 2.2 Biểu đồ phân rã (Sinh viên)

## 2.2 Mô tả chi tiết các use case

### 2.2.1 Đăng nhập

* **Tên use case:** Đăng nhập.
* **Mô tả vắn tắt**:Use case này cho phép admin đăng nhập tài khoản.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi Admin bấm vào “Đăng nhập” trên thanh menu. Hệ thống sẽ truy xuất vào danh sách các tài khoản và lấy thông tin (email, password) từ bảng **users** để hiển thị thông tin lên màn hình.
2. Người dùng nhập **email** và **password** sau đó kích vào nút **Đăng nhập.**Hệthống kiểm tra tên đăng nhập, mật khẩu và quyền truy nhập trong bảng**users** và hiển thị màn hình hệ thống với các menu tương ứng với  
   quyền. Use case kết thúc

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.
2. Tại bước 2 nếu không nhập đầy đủ thông tin theo yêu cầu hệ thống sẽ đưa ra thông báo yêu cầu nhập đầy đủ thông tin.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép admin/giáo viên thực hiện
* Tiền điều kiện: không có.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

### 2.2.2 Quản lí sinh viên

* **Tên use case:** Quản lý sinh viên.
* **Mô tả vắn tắt**:Use case này cho phép admin quản lí sinh viên trong hệ thống.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi Admin bấm vào “Quản lí sinh viên” trên thanh menu. Hệ thống sẽ truy xuất vào danh sách các tài khoản và lấy thông tin sinh viên từ bảng **students** để hiển thị thông tin lên màn hình.
2. Sửa tài khoản:
3. Admin bấm vào biểu tượng sửa bên phải trên dòng tài khoản muốn sửa. hệ thống sẽ hiển thị các thông tin của sinh viên.
4. Khi admin thay đổi bất kỳ thông tin xong bấm vào nút “Save”. Hệ thống sẽ kiểm tra và cập nhật lại thông tin của bản ghi vào bảng **students** và hiển thị thông tin đã được cập nhật.
5. Xóa tài khoản:
   1. Admin bấm vào biểu tượng xóa bên phải trên dòng sinh viên muốn xóa. Hệ thống sẽ hiển thị popup confirm có muốn xóa không.
   2. Sau khi kiểm tra thông tin admin bấm nút “Xác nhận” hệ thống sẽ xóa tài khoản theo yêu cầu và hiển thị danh sách sinh viên sau khi xóa.

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.
2. Tại bước 2 nếu không nhập đầy đủ thông tin theo yêu cầu hệ thống sẽ đưa ra thông báo yêu cầu nhập đầy đủ thông tin.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép admin thực hiện
* Tiền điều kiện: không có.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

### 2.2.3 Quản lí khóa học

* **Tên use case:** Quản lý khóa học.
* **Mô tả vắn tắt**:Use case này cho phép giáo viên quản lí các khóa học có trong hệ thống.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi Admin bấm vào “Quản lí khóa học” trên thanh menu. Hệ thống sẽ truy xuất vào danh sách khóa học và lấy thông tin khóa học từ bảng **courses** để hiển thị thông tin lên màn hình.
2. Tạo khóa học:

a. Admin bấm vào nút “Thêm khóa học” bên phải danh sách khóa học. Hệ thống sẽ hiển thị các thông tin cần thiết để thêm khóa học

b. Khi admin đã nhập đầy đủ các thông tin cần thiết của khóa học, nhấn nút “Save”. Hệ thống sẽ kiểm tra và tạo mới thông tin khóa học vào bảng **courses**.

1. Sửa khóa học:

a. Admin bấm vào biểu tượng sửa bên phải trên dòng khóa học muốn sửa. hệ thống sẽ hiển thị các thông tin của khóa học.

b.Khi admin thay đổi bất kỳ thông tin xong bấm vào nút “Save”. Hệ thống sẽ kiểm tra và cập nhật lại thông tin của bản ghi vào bảng **course** và hiển thị thông tin đã được cập nhật.

1. Xóa khóa học:
   1. Admin bấm vào biểu tượng xóa bên phải trên dòng khóa học muốn xóa. Hệ thống sẽ hiển thị popup confirm có muốn xóa không.
   2. Sau khi kiểm tra thông tin admin bấm nút “Xác nhận” hệ thống sẽ xóa tài khoản theo yêu cầu và hiển thị danh sách khóa học sau khi xóa.

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.
2. Tại bước 2 và 3 nếu không nhập đầy đủ thông tin theo yêu cầu hệ thống sẽ đưa ra thông báo yêu cầu nhập đầy đủ thông tin.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép admin thực hiện
* Tiền điều kiện: không có.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

### 2.2.4. Quản lí bài học

* **Tên use case:** Quản lý bàihọc.
* **Mô tả vắn tắt**:Use case này cho phép giáo viên quản lí các bàihọc có trong hệ thống.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi Admin bấm vào “Quản lí bàihọc” trên thanh menu. Hệ thống sẽ truy xuất vào danh sách bàihọc và lấy thông tin bàihọc từ bảng **lessons** để hiển thị thông tin lên màn hình.
2. Tạo bài học:

a. Admin bấm vào nút “Thêm bàihọc” bên phải danh sách bàihọc. Hệ thống sẽ hiển thị các thông tin cần thiết để thêm bàihọc

b. Khi admin đã nhập đầy đủ các thông tin cần thiết của bàihọc, nhấn nút “Save”. Hệ thống sẽ kiểm tra và tạo mới thông tin bàihọc vào bảng **lessons**.

1. Sửa bài học:

a. Admin bấm vào biểu tượng sửa bên phải trên dòng bàihọc muốn sửa. hệ thống sẽ hiển thị các thông tin của bàihọc.

b.Khi admin thay đổi bất kỳ thông tin xong bấm vào nút “Save”. Hệ thống sẽ kiểm tra và cập nhật lại thông tin của bản ghi vào bảng **lessons** và hiển thị thông tin đã được cập nhật.

1. Xóa bài học:
   1. Admin bấm vào biểu tượng xóa bên phải trên dòng bài học muốn xóa. Hệ thống sẽ hiển thị popup confirm có muốn xóa không.
   2. Sau khi kiểm tra thông tin admin bấm nút “Xác nhận” hệ thống sẽ xóa tài khoản theo yêu cầu và hiển thị danh sách bài học sau khi xóa.

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.
2. Tại bước 2 và 3 nếu không nhập đầy đủ thông tin theo yêu cầu hệ thống sẽ đưa ra thông báo yêu cầu nhập đầy đủ thông tin.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép admin thực hiện
* Tiền điều kiện: không có.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

**2.2.5. Tham gia khóa học**

* **Tên use case:** Tham gia khóa học.
* **Mô tả vắn tắt**:Use case này cho phép học viên tham gia các khóa học trong hệ thống.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi học viên bấm vào “Xem thông tin khóa học” trong danh sách khóa học. Hệ thống sẽ truy xuất vào danh sách khóahọc và lấy thông tin khóahọc từ bảng **course** để hiển thị thông tin khóa học lên màn hình.
2. Tham gia khóa học:

Sinh viên chọn tham gia khóa học tại trang thông tin chi tiết khóa học. Hệ thống sẽ truy xuất thông tin sinh viên và khóa học và tạo quan hệ giữa bảng **students** và **courses.**

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép học viên thực hiện
* Tiền điều kiện: Học viên cần đăng nhập vào hệ thống.
* Hậu điều kiện: không có.
* Điểm mở rộng:Học viên có thể xem các bài học có trong khóa.

**2.2.6. Xem bài học**

* **Tên use case:** Xem bài học.
* **Mô tả vắn tắt**: Usecase này cho phép học viên xem bài học trong khóa học.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi học viên bấm vào “Xem thông tin khóa học” trong danh sách khóa học. Hệ thống sẽ truy xuất vào danh sách khóahọc và lấy thông tin khóahọc từ bảng **course** để hiển thị thông tin khóa học lên màn hình.
2. Xem bài học:

Sau khi đã “Tham gia khóa học”, sinh viên có thể xem các bài học trong khóa. Hệ thống sẽ lưu các thông tin xem khóa học của sinh viên để lưu tiến độ học tập của sinh viên.

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép học viên thực hiện
* Tiền điều kiện:
* Học viên cần đăng nhập vào hệ thống.
* Học viên cần tham gia khóa học của bài học đó.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

**2.2.7. Đánh giá bài học**

* **Tên use case:** Đánh giá bài học.
* **Mô tả vắn tắt**: Use case này cho phép học viên đánh giá bài học sau khi đã tham gia “Xem bài học”.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi học viên bấm vào xem bài học trong danh sách bài học. Hệ thống sẽ hiển thị thông tin bài học trong bảng **lessons** và hiển thị các đánh giá khóa học trước đó trong bảng **reviews**.
2. Đánh giá bài học:

Học viên tiến hành đánh giá bài học bằng cách rate số sao và thêm bình luận với mỗi bài học. Sau khi nhấn nút đánh giá, sẽ hiển thị đánh giá của sinh viên phía dưới mỗi bài học.

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép học viên thực hiện
* Tiền điều kiện:
* Học viên cần đăng nhập vào hệ thống.
* Học viên cần tham gia khóa học của bài học đó.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

**2.2.8. Đăng kí**

* **Tên use case:** Đăng kí tài khoản.
* **Mô tả vắn tắt**: Use case này cho phép học viên đăng kí tài khoản trên hệ thống.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi học viên bấm vào đăng kí trên thanh topbar. Hệ thống sẽ hiển thị các thông tin bao gồm thông tin học viên email, số điện thoại và mật khẩu. Sau khi nhập đủ các thông tin trên, hệ thống sẽ mã hóa mật khẩu và đăng kí thông tin học viên vào bảng **students.**

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép học viên thực hiện
* Tiền điều kiện: không có.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

**2.2.9. Quên mật khẩu**

* **Tên use case:** Quên mật khẩu.
* **Mô tả vắn tắt**: Use case cho phép sinh viên lấy lại mật khẩu khi quên.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi học viên nhấn vào nút “Quên mật khẩu” tại màn hình đăng nhập. Hệ thống sẽ yêu cầu học viên nhập địa chỉ email. Sau khi nhập và ấn “Submit” hệ thống sẽ gửi mail bao gồm mật khẩu mới cho sinh viên.

* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép học viên thực hiện
* Tiền điều kiện: Học viên cần có tài khoản trước đó.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

**2.2.10. Gửi mail chăm sóc học viên**

* **Tên use case:** Gửi mail chăm sóc học viên.
* **Mô tả vắn tắt**: Hệ thống sẽ định kì gửi mail báo cáo tiến độ cho học viên.
* **Luồng sự kiện:**
* **Luồng cơ bản:**

1. Use case này bắt đầu khi học viên bấm nút “Đăng kí nhận thông tin học tập”. Hệ thống sẽ lưu lại mong muốn và sẽ gửi mail định kì hàng tuần cho học viên

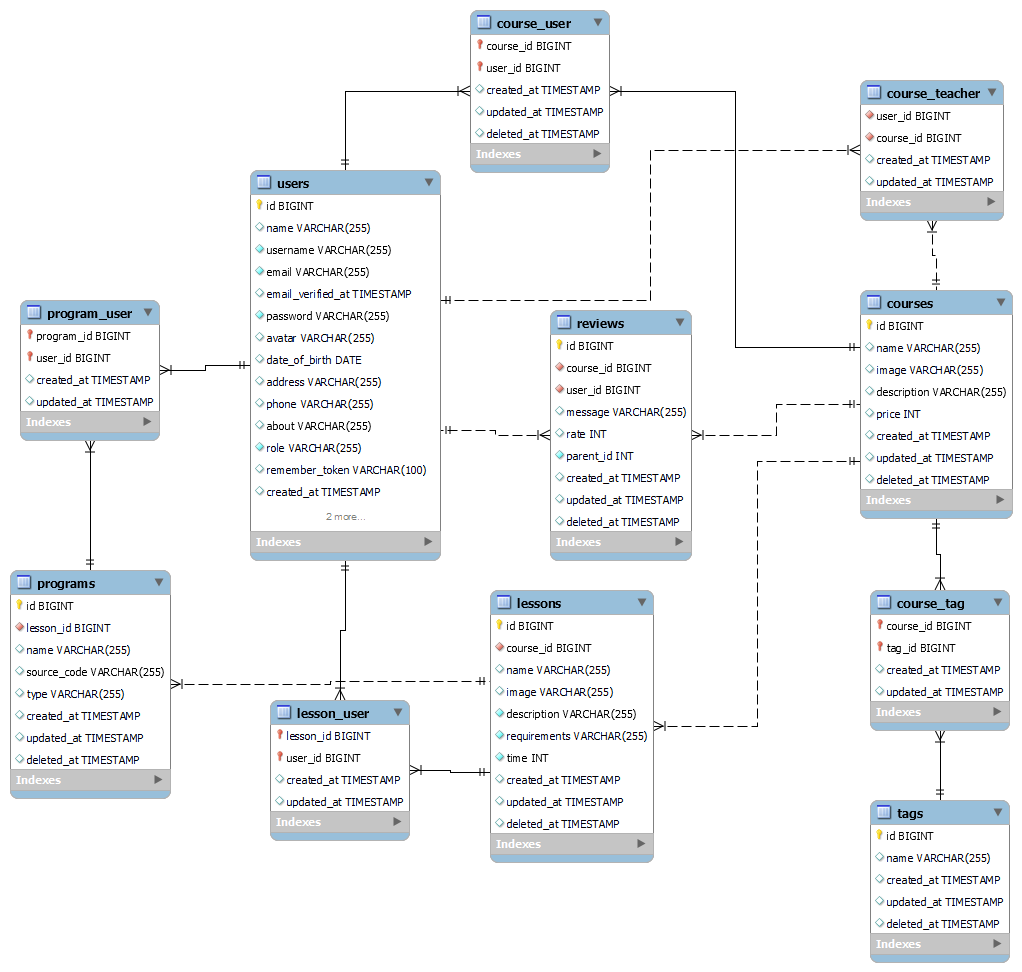
* **Luồng rẽ nhánh:**

1. Tại bất kì thời điểm nào trong quá trình thực hiện use case nếu không kết nối được với cơ sở dữ liệu thì hệ thống sẽ hiển thị một dòng thông báo lỗi và use case kết thúc.

* Các yêu cầu đặc biệt: Use case này chỉ cho phép học viên thực hiện
* Tiền điều kiện:
* Học viên cần đăng nhập vào hệ thống.
* Hậu điều kiện: không có.
* Điểm mở rộng: không có.

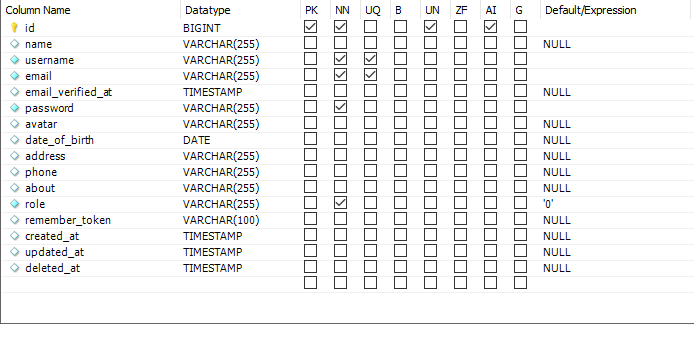
# Chương 3 THIẾT KẾ CƠ SỞ DỮ LIỆU

## 3.1. Biểu đồ quan hệ cơ sở dữ liệu

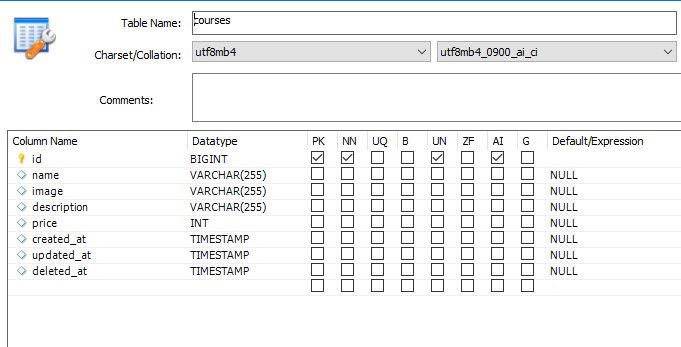


## 3.2. Thiết kế bảng

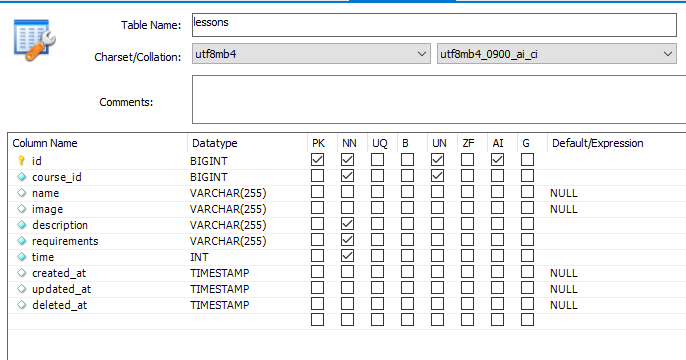
### 3.2.1. Bảng users



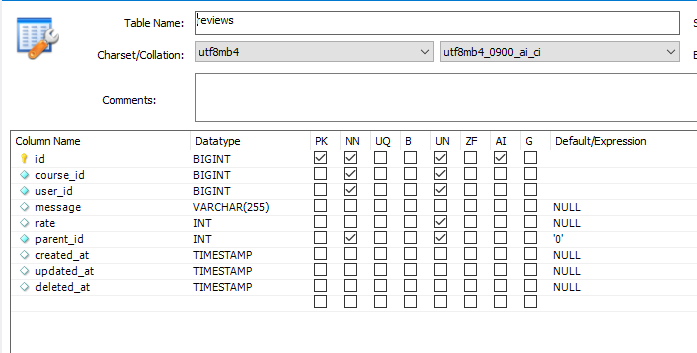
### 3.2.2. Bảng courses



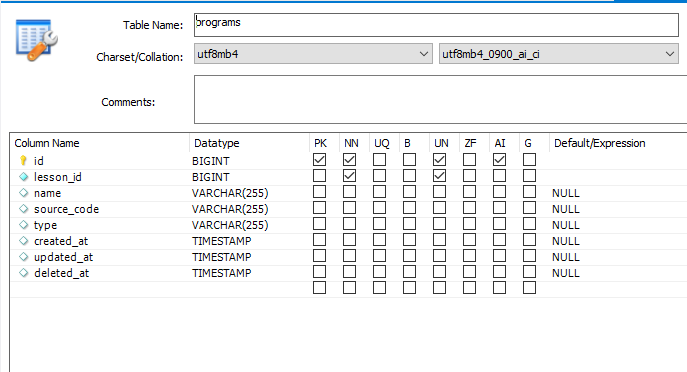
### 3.2.3. Bảng lessons



### 3.2.4. Bảng reviews



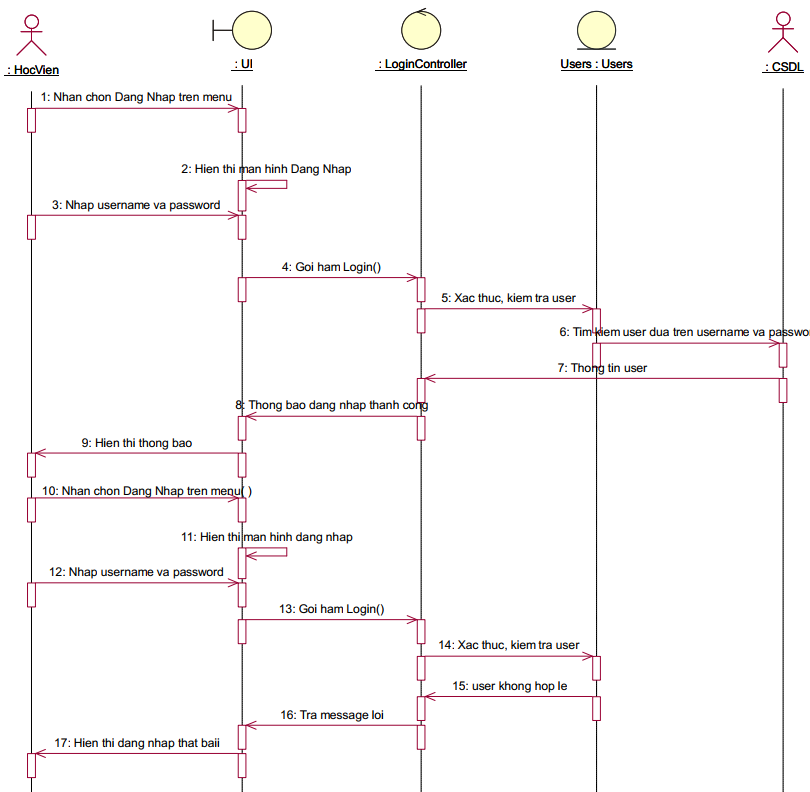
### 3.2.5. Bảng programs



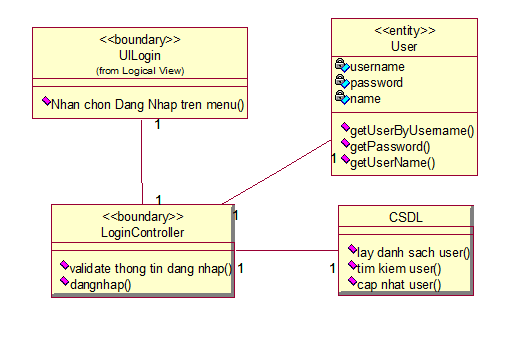
# Chương 4 PHÂN TÍCH VÀ THIẾT KẾ CHỨC NĂNG

## 4.1. Use case Đăng nhập

## 4.1.1. Biểu đồ trình tự

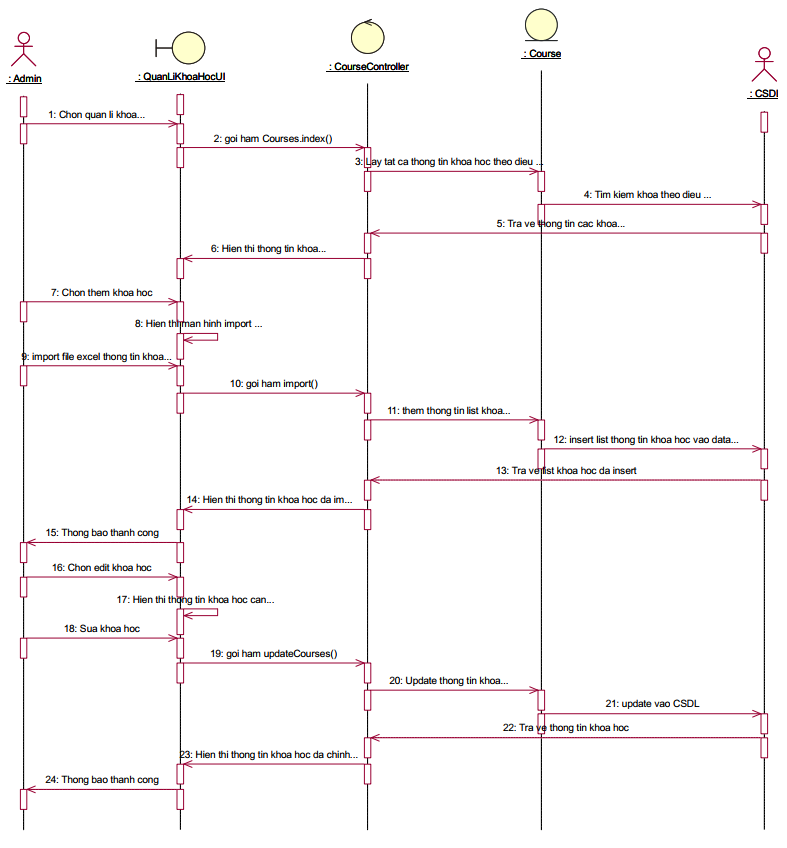


### 4.1.2. Biểu đồ lớp

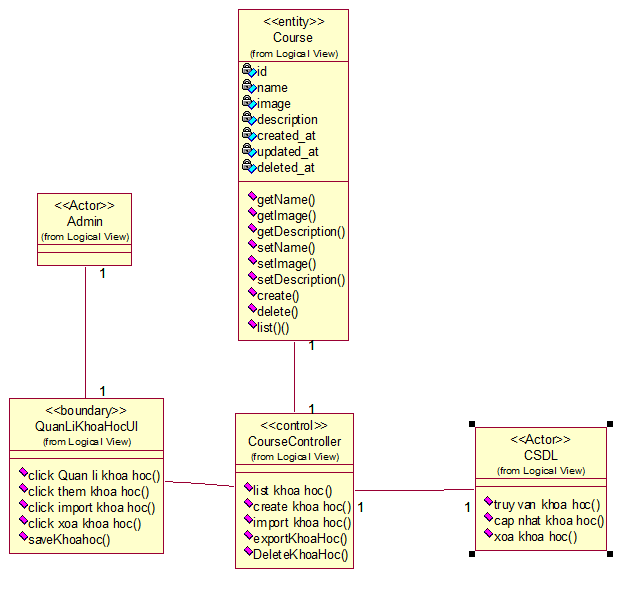


## 4.2. Use case Quản lí khóa học

### 4.2.1. Biểu đồ trình tự

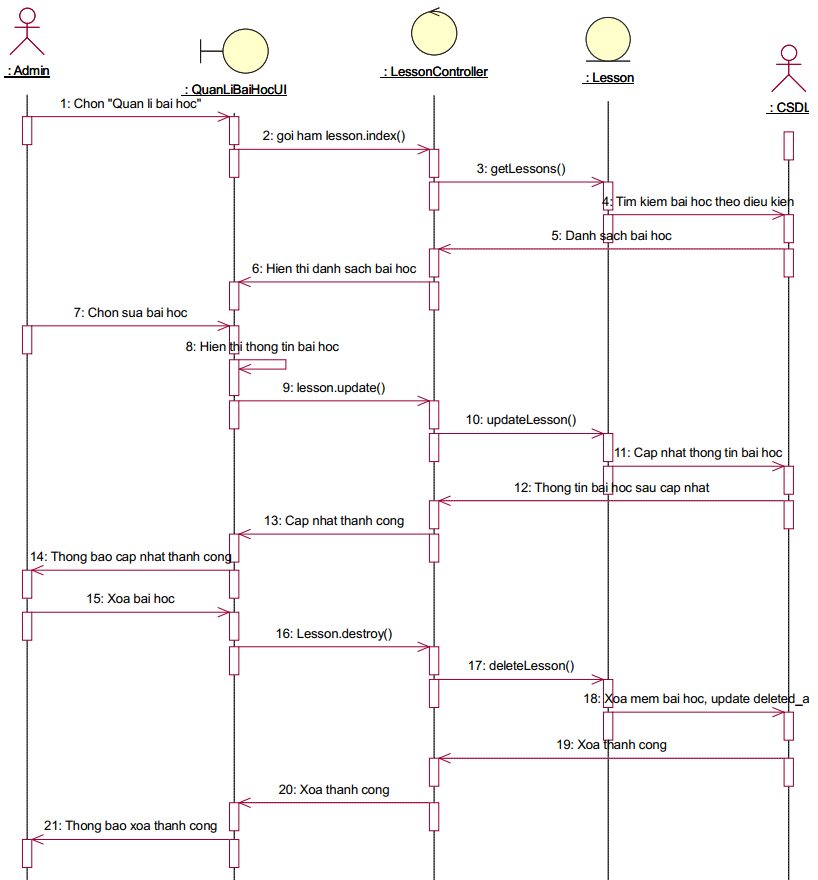


### 4.2.2. Biều đồ lớp

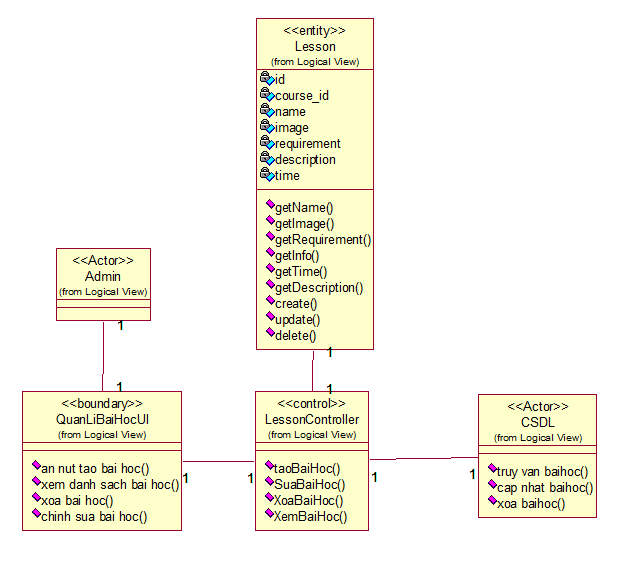


## 4.3. Use case Quản lí bài học

### 4.3.1. Biểu đồ trình tự

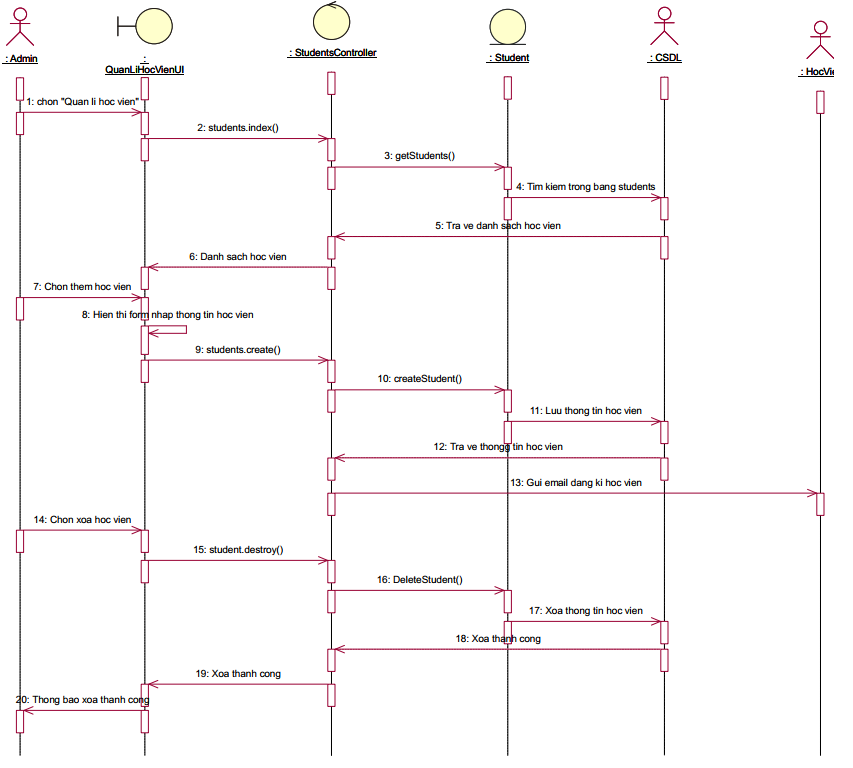


### 4.3.2. Biểu đồ lớp

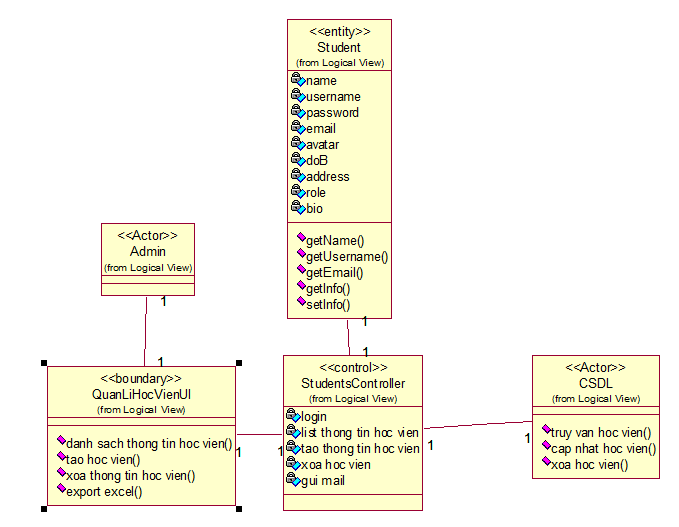


## 4.4. Use case Quản lí học viên

### 4.4.1. Biều đồ trình tự

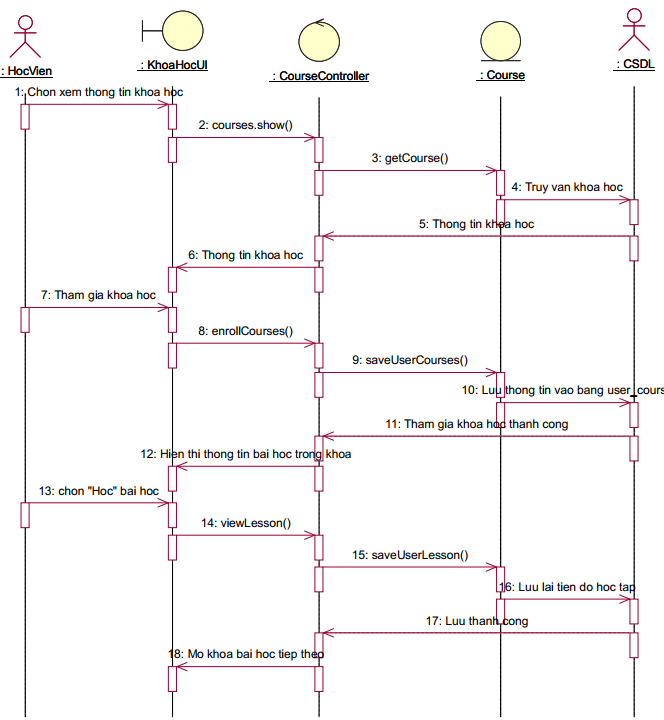


### 4.4.2. Biểu đồ lớp



## 4.5. Use case Tham gia khóa học

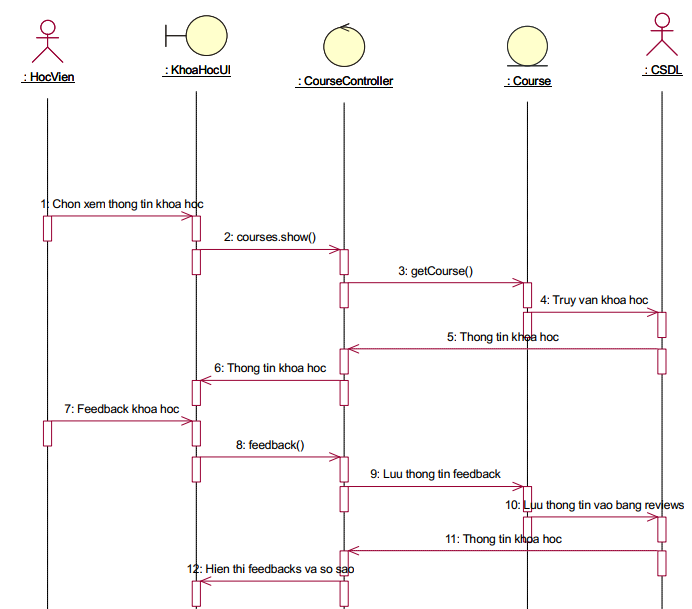
### 4.5.1. Biểu đồ trình tự



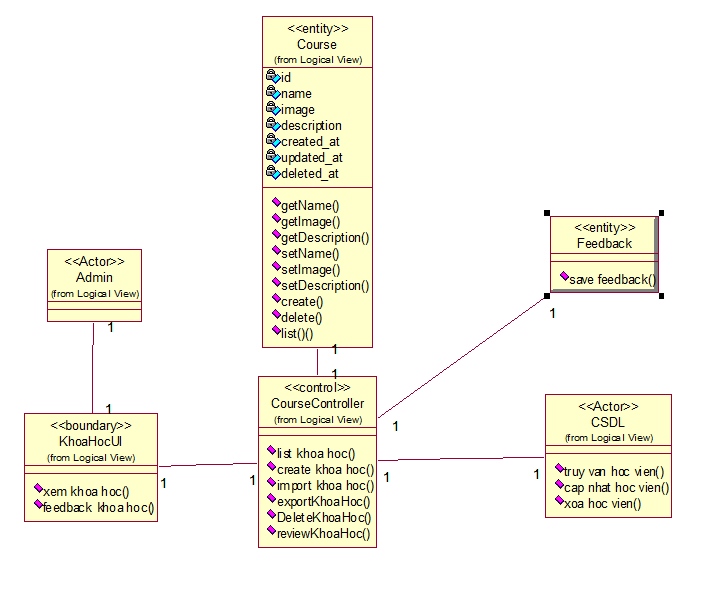
### 4.5.2. Biểu đồ lớp

## 4.6. Use case Đánh giá bài học

### 4.6.1. Biểu đồ trình tự

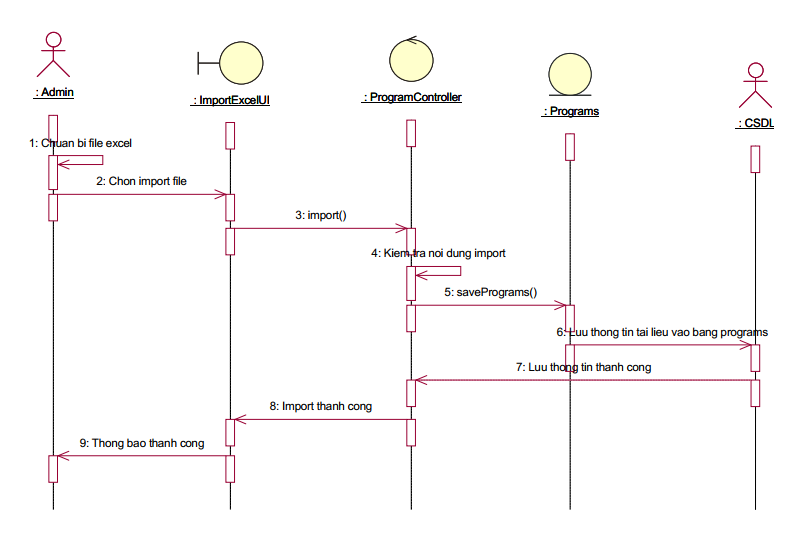


### Biểu đồ lớp

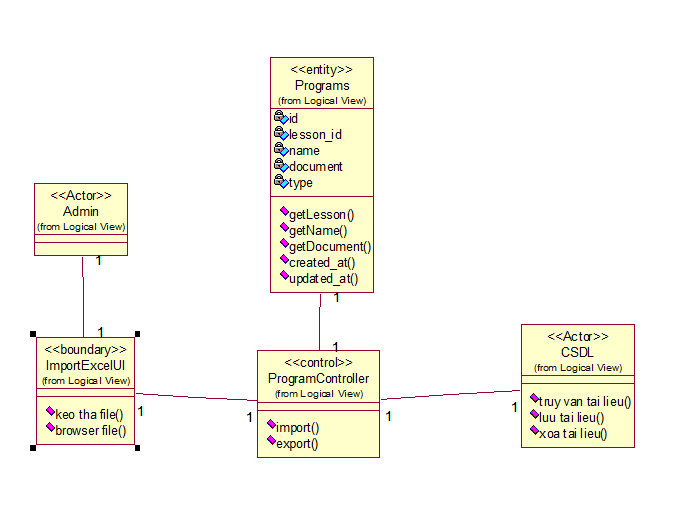


## 4.7. Use case tạo nội dung bài học

### 4.7.1. Biểu đồ trình tự



### 4.7.2. Biểu đồ lớp



# Chương 5 Thiết kế giao diện và cài đặt

## 5.1. Thiết kế chi tiết các màn hình

## 5.2. Giới thiệu ngôn ngữ cài đặt

### 5.2.1. Ngôn ngữ PHP

PHP là từ viết tắt của thuật ngữ Personal Home Page. Đây là một dạng mã lệnh hoặc một chuỗi ngôn ngữ kịch bản. Trong đó, ngôn ngữ PHP chủ yếu được phát triển để dành cho những ứng dụng nằm trên máy chủ. Mỗi khi các lập trình viên PHP viết các chương trình thì các chuỗi lệnh sẽ được chạy ở trên server, từ đó sinh ra mã HTML. Nhờ vậy mà những ứng dụng trên các website có thể chạy được một cách dễ dàng.

Ngôn ngữ PHP thông thường sẽ phù hợp với việc lập trình website bởi nó có thể dễ dàng kết nối với các website khác có sử dụng HTML để chạy trên các trình duyệt web. Vì vậy, đây là ngôn ngữ lập trình được người dùng đánh giá là khá dễ đọc. Ngôn ngữ PHP cũng trở thành ngôn ngữ web dev phổ biến mà các lập trình viên phải học trước khi bắt đầu vào nghề.

### 5.2.2. Framework Laravel

Laravel [3] là một framework mã nguồn mở dành cho việc phát triển các ứng dụng web bằng ngôn ngữ PHP2 . Laravel được phát triển bởi Taylor Otwell và được phát hành lần đầu vào năm 2011. Laravel được thiết kế để cung cấp một cách tiếp cận đơn giản và linh hoạt cho việc phát triển các ứng dụng web với tính năng đầy đủ.

Laravel bao gồm nhiều tính năng quan trọng, bao gồm:

- Cú pháp đơn giản và dễ hiểu: Laravel cung cấp cú pháp đơn giản và dễ hiểu, giúp cho các lập trình viên mới bắt đầu với framework có thể dễ dàng tiếp cận.

- Tính năng routing: Laravel cung cấp tính năng routing để định tuyến các yêu cầu HTTP 3 đến các phương thức của ứng dụng.

- Hỗ trợ tạo các bảng cơ sở dữ liệu: Laravel cung cấp tính năng tạo các bảng cơ sở dữ liệu, tạo và sử dụng các model để tương tác với cơ sở dữ liệu một cách dễ dàng.

- Hỗ trợ giao diện người dùng: Laravel cung cấp tính năng hỗ trợ tạo giao diện người dùng, với các tính năng như Blade template engine, tạo các form và kiểm tra dữ liệu nhập vào.

- Hỗ trợ xử lý các yêu cầu AJAX : Laravel cung cấp tính năng hỗ trợ xử lý các yêu cầu AJAX để tạo ra các ứng dụng động.

- Hỗ trợ các chức năng bảo mật: Laravel cung cấp các tính năng bảo mật như bảo vệ các truy vấn SQL , giám sát các hoạt động người dùng và xác thực người dùng. Laravel là một trong những framework PHP phổ biến nhất và được sử dụng rộng rãi trong cộng đồng phát triển ứng dụng web

### 5.2.3. MySQL

MySQL [4] là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS6) mã nguồn mở phổ biến, được phát triển bởi công ty Oracle Corporation. Nó là một phần mềm miễn phí và được sử dụng rộng rãi trên các hệ thống web và ứng dụng doanh nghiệp.

MySQL cung cấp các tính năng quản lý cơ sở dữ liệu, bao gồm tạo, sửa đổi và xóacác bảng, truy vấn dữ liệu, quản lý người dùng và phân quyền, cũng như sao lưu và phục hồi cơ sở dữ liệu. Nó cũng hỗ trợ nhiều ngôn ngữ lập trình khác nhau như PHP, Java, Python và các công nghệ web như Apache 7 và Nginx8.

MySQL có thể chạy trên nhiều nền tảng khác nhau, bao gồm Windows, Linux, macOS và nhiều hệ thống khác. Nó được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp nhỏ đến lớn vì tính linh hoạt, độ tin cậy và hiệu suất cao.

Ưu điểm:

- Hỗ trợ đa luồng, nhanh chóng đối với cả các tập dữ liệu lớn

- Độ bảo mật cao, thích hợp cho những ứng dụng truy cập CSDL thông qua internet

khi sở hữu rất nhiều tính năng về bảo mật.

- MySQL hoạt động trên nhiều hệ điều hành và với nhiều ngôn ngữ bao gồm PHP,

PERL9

, C10, JAVA11, v.v.

- Cung cấp nhiều kiểu dữ liệu khác nhau: số, kí tự, đoạn văn, nhị phân …

- Kết nối giữa các bảng tốc độ cao.

- Có cơ chế phân quyền và mật khẩu linh hoạt để bảo mật và bảo vệ. Bạn có thể kiểm soát quyền và truy cập dựa trên máy được kết nối. Mật khẩu luôn được bảo mật khi được gửi qua mạng và luôn được lưu trữ mã hóa.

Nhược điểm:

- MySQL bị hạn chế dung lượng, cụ thể, khi số bản ghi của người dùng lớn dần, sẽ

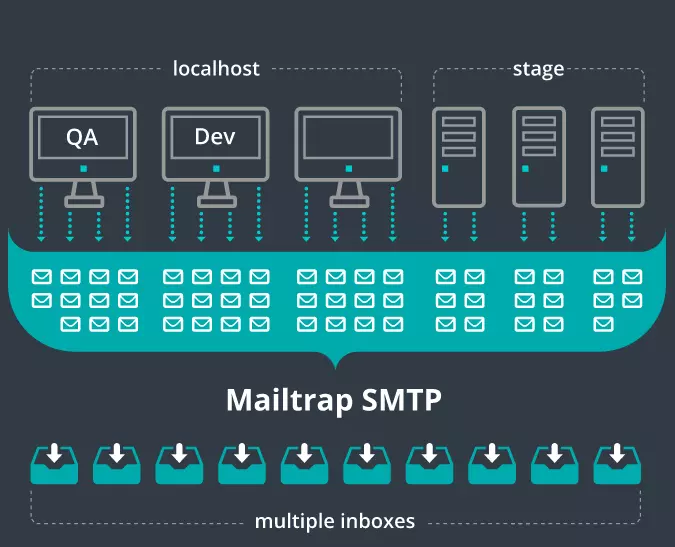
gây khó khăn cho việc truy xuất dữ liệu.

- Hiện MySQL không hỗ trợ truy vấn con, chế độ xem, thủ tục được lưu trữ, trình kích hoạt và thực thi khóa ngoại.

### 5.2.4. MailTrap

Mailtrap cung cấp một máy chủ SMTP giả (fake SMTP server) dành cho đội ngũ phát triển. Được sử dụng để test các emails được gửi từ các môi trường pre-production và môi trường test với dữ liệu thực tế mà vẫn tránh được nguy cơ gửi thư rác (test email) tới các khách hàng thật của hệ thống cần test. Nó còn cho phép bạn tự động forward mail đến hộp thư của bạn thường xuyên, chia sẻ với team của bạn và nhiều hơn nữa!

Về cơ bản, bạn chỉ cần đăng ký một tài khoản sử dụng Mailtrap, setting đồng bộ với project code của app cần test, và gửi tất cả các email cần test của môi trường pre-production tới các địa chỉ email cần nhận mail thông báo ^^. Tất cả các email đó sẽ được gửi tới fake SMTP Mailtrap server thay vì gửi đến các email mà bạn đã chỉ định.



Các tính năng cơ bản của MailTrap

1. Không bao giờ xảy ra việc spam mail các khách hàng thật bởi những sơ suất của đội phát triển
2. Phân tích email cho spam và blacklist
3. Test HTML cho việc hỗ trợ bởi các common email clients
4. Mặc định bảo mật (HTTPS & TLS)
5. Cài đặt nhanh chóng và dễ dàng bằng cách dùng config code
6. Cải thiện RESTful & POP3 API autotests
7. Tự động Forward emails tới những người cần chúng
8. Sắp xếp các Email test vào hộp thư đặc biệt
9. Chia sẻ inboxes với team của bạn