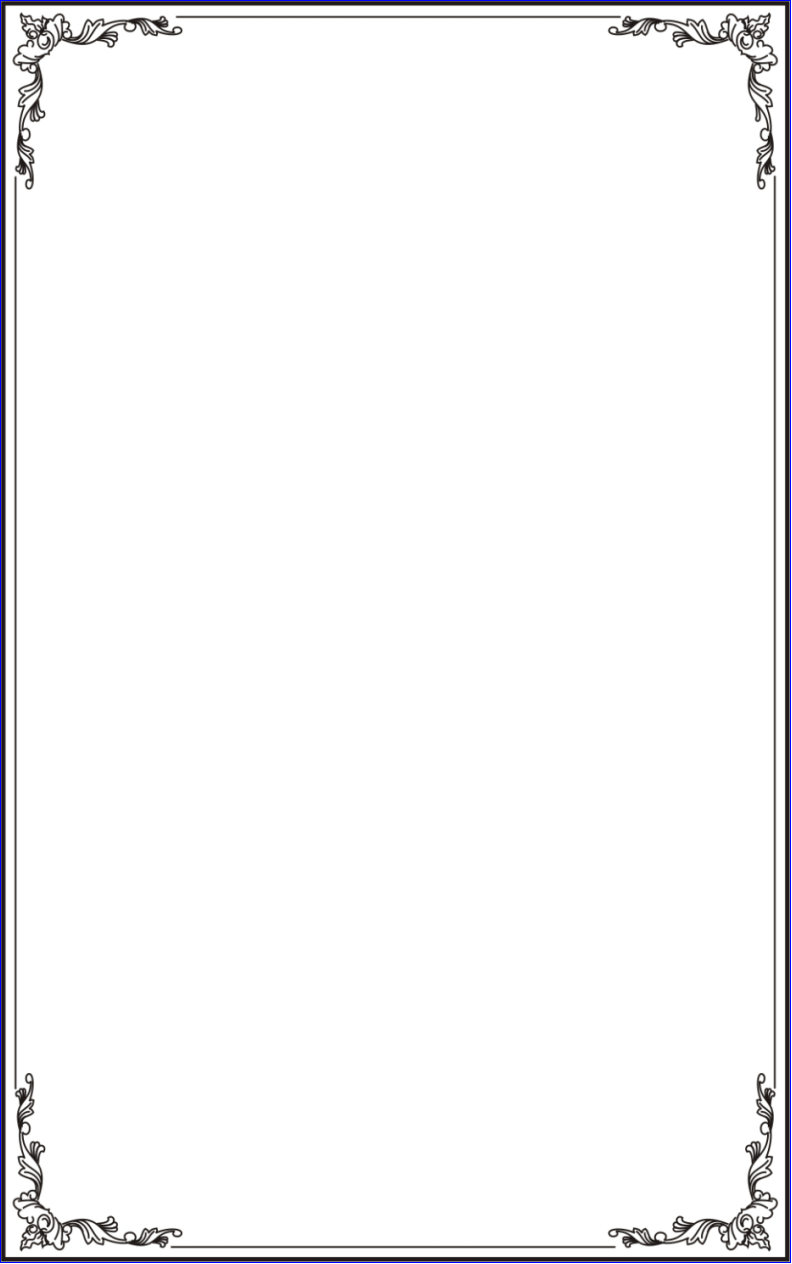
****

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

---------------------------------------

****

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG ỨNG DỤNG WEB CHO HỆ THỐNG HỌC TRỰC TUYẾN SỬ DỤNG SPRING BOOT VỚI AI**

**NGÀNH KỸ THUẬT PHẦN MỀM**

**GVHD:** **TS. Hà Mạnh Đào**

**Sinh viên:**  Nguyễn Điền Sơn

**Lớp:** 2021KTPM01 **Khóa:** 16

**Hà Nội – Năm 2025**

**MỤC LỤC**

[**LỜI CẢM ƠN** 5](#_Toc192694355)

[**PHẦN I. MỞ ĐẦU** 6](#_Toc192694356)

[**1.** **Tên đề tài** 6](#_Toc192694357)

[**2.** **Lý do chọn đề tài** 6](#_Toc192694358)

[**3.** **Mục đích của đề tài** 6](#_Toc192694359)

[**4.** **Mục tiêu của đề tài** 6](#_Toc192694360)

[**5.** **Bố cục đề tài** 7](#_Toc192694361)

[**6.** **Phương pháp nghiên cứu** 7](#_Toc192694362)

[**PHẦN II. NỘI DUNG** 8](#_Toc192694363)

[**Chương 1. Cơ sở lý thuyết** 8](#_Toc192694364)

[**1.1** **Cơ sở lý luận** 8](#_Toc192694365)

[**1.1.1** **Khái niệm về Spring Framework** 8](#_Toc192694366)

[**1.1.2** **Mô hình MVC trong Spring** 8](#_Toc192694367)

[**1.1.3** **DI, IoC trong Spring** 9](#_Toc192694368)

[**1.2 Cơ sở thực tiễn** 9](#_Toc192694369)

[**1.2.1 Nhu cầu thực tế sử dụng Spring** 9](#_Toc192694370)

[**1.2.2 Lợi ích của Spring** 9](#_Toc192694371)

[**1.2.3 Ứng dụng của Spring trong các dự án thực tế** 10](#_Toc192694372)

[**1.3 Tổng quan các công nghệ sử dụng** 10](#_Toc192694373)

[**1.3.1 Hibernate - ORM cho Java** 10](#_Toc192694374)

[**1.3.2 JDBC - Kết nối CSDL** 11](#_Toc192694375)

[**1.3.3 Spring Security - Bảo mật ứng dụng** 11](#_Toc192694376)

[**Chương 2. Phân tích thiết kế hệ thống** 12](#_Toc192694377)

[**2.1 Khảo sát và phân tích hiện trạng** 12](#_Toc192694378)

[**2.1.1 Giới thiệu đơn vị khảo sát** 12](#_Toc192694379)

[**2.1.2 Hiện trạng hoạt động và các vấn đề tồn tại** 12](#_Toc192694380)

[**2.1.3 Nhu cầu xây dựng hệ thống mới** 13](#_Toc192694381)

[**2.1.4 Phương pháp khảo sát** 14](#_Toc192694382)

[**2.1.5 Kết quả khảo sát** 15](#_Toc192694383)

[**2.1.6 Định hướng xây dựng hệ thống** 15](#_Toc192694384)

[**2.2 Đặc tả yêu cầu hệ thống phần mềm** 16](#_Toc192694385)

[**2.2.1 Yêu cầu chức năng** 16](#_Toc192694386)

[**2.2.2 Yêu cầu phi chức năng** 17](#_Toc192694387)

[**2.3 Phân tích và thiết kế hệ thống** 18](#_Toc192694388)

[**2.3.1 Phân tích và mô hình hóa chức năng của hệ thống** 18](#_Toc192694389)

[**2.3.2 Phân tích và mô hình hóa chức năng** 19](#_Toc192694390)

[***Biểu đồ use case tổng quát*** 19](#_Toc192694391)

[**2.3.2.1 Use case “Đăng ký”** 19](#_Toc192694392)

[**2.3.2.2 Use case “Đăng nhập”** 21](#_Toc192694393)

[**2.3.2.3 Use case “Quản lý người dùng”** 23](#_Toc192694394)

[**2.3.2.4 Use case “Quản lý khóa học”** 26](#_Toc192694395)

[**2.3.2.5 Use case “Đăng ký khóa học”** 29](#_Toc192694396)

[**2.3.2.6 Use case “Quản lý bài tập”** 31](#_Toc192694397)

[**2.3.2.7 Use case “Nộp bài tập”** 34](#_Toc192694398)

[**2.3.2.8 Use case “Chấm điểm”** 36](#_Toc192694399)

[**2.4 Phân tích và mô hình hóa dữ liệu** 39](#_Toc192694400)

[**2.4.1 Các thực thể chính và thuộc tính** 39](#_Toc192694401)

[**2.4.2 Biểu đồ lớp thực thể liên kết** 40](#_Toc192694402)

[**2.5 Thiết kế giao diện** 41](#_Toc192694403)

[**2.5.1 Giao diện Đăng ký** 41](#_Toc192694404)

[**2.5.2 Giao diện Đăng nhập** 41](#_Toc192694405)

[**2.5.3 Giao diện Thêm khóa học** 42](#_Toc192694406)

[**2.5.4 Giao diện Chỉnh sửa khóa học** 42](#_Toc192694407)

[**2.5.5 Giao diện Quản lý tài khoản** 42](#_Toc192694408)

[**2.5.6 Giao diện Thêm bài tập** 43](#_Toc192694409)

[**2.5.7 Giao diện Nhập điểm** 43](#_Toc192694410)

[**Chương 3. Cài đặt và thử nghiệm** 44](#_Toc192694411)

[**3.1 Cài đặt JDK** 44](#_Toc192694412)

[**3.2 Thiết lập biến môi trường cho Java** 45](#_Toc192694413)

[**3.3 Cài đặt Eclipse** 47](#_Toc192694414)

[**3.4 Chạy dự án** 49](#_Toc192694415)

[**PHẦN III. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 51](#_Toc192694416)

[**1. Nội dung đã thực hiện được** 51](#_Toc192694417)

[**2. Hướng phát triển** 52](#_Toc192694418)

[**PHẦN IV. TÀI LIỆU THAM KHẢO** 54](#_Toc192694419)

# **LỜI CẢM ƠN**

Lời đầu tiên, em xin chân thành cảm ơn khoa Công nghệ thông tin, trường Đại học Công nghiệp Hà Nội đã tạo điều kiện thuận lợi cho em thực hiện môn Đồ án tốt nghiệp.

Đặc biệt, em xin tỏ lòng biết ơn sâu sắc tới thầy TS. Hà Mạnh Đàođã tận tâm hướng dẫn, hỗ trợ em trong quá trình thực hiện đồ án. Cô đã đưa ra những góp ý quý báu, giúp em định hướng rõ ràng và hoàn thiện bài làm của mình một cách tốt nhất.

Trong suốt thời gian thực hiện môn Đồ án tốt nghiệp, em đã có cơ hội tiếp cận với những vấn đề thực tế, rèn luyện kỹ năng làm việc và nâng cao kiến thức chuyên môn. Tuy nhiên, do thời gian tìm hiểu còn hạn chế và kinh nghiệm thực tế chưa nhiều, chắc chắn bài làm vẫn còn nhiều thiếu sót. Kính mong cô góp ý để em có thể hoàn thiện đồ án một cách tốt nhất.

*Em* *xin* *chân* *thành* *cảm* *ơn!*

***Sinh viên thực hiện***

*Nguyễn Điền Sơn*

# **PHẦN I. MỞ ĐẦU**

1. **Tên đề tài**

Xây dựng ứng dụng web cho hệ thống học trực tiếp sử dụng Spring Boot với AI.

1. **Lý do chọn đề tài**

Lựa chọn đề tài "Xây dựng ứng dụng web cho hệ thống học trực tiếp sử dụng Spring Boot với AI" là một quyết định xuất phát từ nhận thức sâu sắc về tính thực tiễn và sự ứng dụng cao trong thực tế, đặc biệt trong bối cảnh hiện nay khi nhu cầu học tập trực tuyến ngày càng gia tăng.

Trong thời đại số, việc học qua mạng đã trở thành xu hướng, không chỉ giúp người học tiết kiệm thời gian, chi phí mà còn tạo cơ hội học hỏi cho tất cả mọi người ở mọi lứa tuổi. Các khóa học trực tuyến đa dạng về chủ đề, từ kỹ năng mềm đến các môn học chuyên sâu. Bên cạnh đó, xu hướng học tập trực tuyến còn mở ra cơ hội học hỏi cho cả những người đã đi làm và những người muốn nâng cao kiến thức chuyên môn của mình.

Việc xây dựng một website quản lý khóa học không chỉ giải quyết một vấn đề thực tiễn mà còn là cơ hội để phát triển và kiểm thử nhiều kỹ năng kỹ thuật. Quá trình này yêu cầu chúng ta phải thiết kế giao diện người dùng dễ sử dụng, quản lý cơ sở dữ liệu hiệu quả. Điều này không chỉ giúp nâng cao hiểu biết về phần mềm dịch vụ và công nghệ sử dụng mà còn giúp chúng ta trở thành những lập trình viên có khả năng giải quyết các vấn đề kỹ thuật phức tạp, đồng thời tạo ra một sản phẩm có giá trị thực tiễn cao.

1. **Mục đích của đề tài**

Tìm hiểu về công nghệ Spring và vận dụng những kỹ thuật của Spring để xây dựng ứng dụng web quản lý khóa học.

1. **Mục tiêu của đề tài**

* Tìm hiểu về công nghệ Spring
* Tìm hiểu về ngôn ngữ java, IDE Intellij, cơ sở dữ liệu MySQL
* Ứng dụng các kiến thức trên vào việc xây dựng ứng dụng web quản lý khóa học

1. **Bố cục đề tài**

***Chương 1. Cơ sở lý thuyết***

Trình bày khái niệm về Spring Framework, mô hình MVC trong Spring, Dependency Injection(DI), Inversion of Control(IoC) trong Spring

***Chương 2. Phân tích thiết kế hệ thống***

Nêu ra được các yêu cầu của người dùng, liệt kê các chức năng chính của hệ thống và thiết kế cơ sỏ dữ liệu. Phân tích đặc tả, thiết kế các biểu đồ lớp, use case, tuần tự.

***Chương 3. Cài đặt chương trình***

Xây dựng chương trình dựa vào các phân tích ở trên, chỉ ra cách cài đặt chương trình và kết quả của sản phẩm.

1. **Phương pháp nghiên cứu**
   1. **Phương pháp 1**
   * Tên phương pháp: Nghiên cứu thực tế
   * Cách thực hiện: Khảo sát hệ thống quản lý khóa học trực tuyến hiện có
   1. **Phương pháp 2**
   * Tên phương pháp: Nghiên cứu qua sách báo
   * Cách thực hiện: Mượn sách, báo từ thư viện, nhà sách về nghiên cứu
   1. **Phương pháp 3**
   * Tên phương pháp: Nghiên cứu qua Internet
   * Cách thực hiện: Tìm kiếm và nghiên cứu thông tin, xem mã nguồn các hệ thống quản lý khóa học qua các tài liệu trực tuyến và mã nguồn mở.

# **PHẦN II. NỘI DUNG**

## **Chương 1. Cơ sở lý thuyết**

* 1. **Cơ sở lý luận**
     1. **Khái niệm về Spring Framework**

**Spring Framework** là một framework Java mã nguồn mở, giúp phát triển ứng dụng hiệu năng cao, dễ kiểm thử và tái sử dụng. Nó dựa trên **Dependency Injection (DI)** và **Aspect-Oriented Programming (AOP)**, gồm các module chính:

* Core: Quản lý IoC, DI
* AOP: Hỗ trợ lập trình hướng khía cạnh
* Data Access: Kết nối database (JDBC, ORM, Transaction)
* Web (Spring MVC): Phát triển ứng dụng web
* Test: Hỗ trợ kiểm thử với JUnit, TestNG

Spring nhẹ (~2MB), linh hoạt và có cộng đồng lớn, giúp đơn giản hóa phát triển Java EE.

* + 1. **Mô hình MVC trong Spring**

**Mô hình MVC trong Spring** là kiến trúc giúp tổ chức ứng dụng theo ba thành phần riêng biệt:

* Model chịu trách nhiệm quản lý dữ liệu và xử lý logic nghiệp vụ. Nó có thể lấy dữ liệu từ cơ sở dữ liệu hoặc từ các nguồn khác, sau đó cung cấp cho View hiển thị.
* View đảm nhiệm việc hiển thị dữ liệu và giao tiếp với người dùng. Spring hỗ trợ nhiều loại View như JSP, Thymeleaf, FreeMarker... giúp tách biệt hoàn toàn giao diện với logic xử lý.
* Controller đóng vai trò trung gian, tiếp nhận yêu cầu từ người dùng, gọi Model để xử lý dữ liệu và trả kết quả về View. Spring MVC sử dụng **DispatcherServlet** làm trung tâm điều phối yêu cầu, giúp quản lý luồng dữ liệu hiệu quả.

Mô hình MVC trong Spring giúp tổ chức mã nguồn rõ ràng, dễ mở rộng, bảo trì và kiểm thử.

* + 1. **DI, IoC trong Spring**

Inversion of Control (IoC) là nguyên tắc giúp tách biệt quản lý phụ thuộc giữa các thành phần trong ứng dụng. Thay vì một đối tượng tự tạo hoặc tìm kiếm phụ thuộc của nó, Spring sẽ chủ động cung cấp các phụ thuộc đó, giúp mã nguồn linh hoạt và dễ mở rộng hơn.

Dependency Injection (DI) là cách hiện thực IoC trong Spring. DI cho phép truyền các phụ thuộc vào một đối tượng thay vì đối tượng đó tự khởi tạo. Có hai cách chính để thực hiện DI:

* Constructor Injection: Truyền phụ thuộc thông qua constructor.
* Setter Injection: Truyền phụ thuộc thông qua phương thức setter.

Spring hỗ trợ DI thông qua XML, annotation (@Autowired), hoặc Java Config. Việc sử dụng DI và IoC giúp giảm sự phụ thuộc chặt chẽ giữa các thành phần, cải thiện khả năng kiểm thử và bảo trì mã nguồn.

### **1.2 Cơ sở thực tiễn**

#### **1.2.1 Nhu cầu thực tế sử dụng Spring**

Spring được sử dụng rộng rãi trong phát triển ứng dụng web nhờ khả năng linh hoạt, hiệu năng cao và hỗ trợ nhiều công nghệ. Nó giúp xây dựng ứng dụng hiện đại, đáp ứng nhu cầu của doanh nghiệp như phát triển hệ thống quản lý, thương mại điện tử, dịch vụ API, microservices...

#### **1.2.2 Lợi ích của Spring**

Spring mang lại nhiều lợi ích quan trọng:

* Kiến trúc linh hoạt: Hỗ trợ mô hình MVC, giúp dễ bảo trì và mở rộng.
* Quản lý phụ thuộc tốt: IoC và DI giúp giảm kết nối chặt chẽ giữa các thành phần.
* Bảo mật mạnh mẽ: Spring Security hỗ trợ xác thực, phân quyền, bảo vệ ứng dụng.
* Tích hợp dễ dàng: Hỗ trợ JDBC, Hibernate, JPA để thao tác cơ sở dữ liệu hiệu quả.
* Hỗ trợ RESTful API: Xây dựng API nhanh chóng, phù hợp với các ứng dụng hiện đại.
* Cộng đồng lớn: Hỗ trợ tốt, cập nhật thường xuyên, dễ dàng mở rộng và nâng cấp.

#### **1.2.3 Ứng dụng của Spring trong các dự án thực tế**

Spring được ứng dụng trong nhiều lĩnh vực:

* Hệ thống quản lý doanh nghiệp: ERP, CRM, quản lý nhân sự.
* Thương mại điện tử: Các website bán hàng, nền tảng thanh toán.
* Dịch vụ API và microservices: Hệ thống backend cho ứng dụng web và mobile.
* Tài chính - ngân hàng: Xử lý giao dịch, hệ thống tín dụng, bảo mật thông tin.

### **1.3 Tổng quan các công nghệ sử dụng**

#### **1.3.1 Hibernate - ORM cho Java**

Hibernate là một framework ORM (Object-Relational Mapping) giúp lập trình viên làm việc với cơ sở dữ liệu một cách dễ dàng bằng cách ánh xạ các đối tượng Java với bảng trong database. Hibernate hỗ trợ truy vấn dữ liệu bằng HQL (Hibernate Query Language), tự động quản lý kết nối và tối ưu hiệu suất truy vấn. Nó giúp giảm bớt sự phụ thuộc vào SQL thuần và tăng tính linh hoạt trong phát triển ứng dụng.

#### **1.3.2 JDBC - Kết nối CSDL**

JDBC (Java Database Connectivity) là API giúp Java kết nối và thao tác với cơ sở dữ liệu thông qua SQL. Nó cung cấp các thành phần như DriverManager, Connection, Statement, PreparedStatement, ResultSet, giúp thực thi truy vấn và xử lý dữ liệu. Spring JDBC hỗ trợ quản lý kết nối, xử lý giao dịch và đơn giản hóa thao tác với cơ sở dữ liệu, giúp lập trình viên giảm thiểu lỗi và tối ưu hiệu suất.

#### **1.3.3 Spring Security - Bảo mật ứng dụng**

Spring Security là một framework bảo mật mạnh mẽ giúp bảo vệ ứng dụng Spring khỏi các mối đe dọa như tấn công XSS, CSRF, SQL Injection... Nó hỗ trợ các cơ chế xác thực (authentication) như username/password, OAuth2, JWT, cũng như phân quyền (authorization) dựa trên vai trò người dùng. Spring Security dễ dàng tích hợp với các hệ thống xác thực bên ngoài như LDAP, SSO, giúp đảm bảo an toàn cho ứng dụng web.

## **Chương 2. Phân tích thiết kế hệ thống**

### **2.1 Khảo sát và phân tích hiện trạng**

#### **2.1.1 Giới thiệu đơn vị khảo sát**

Để xây dựng hệ thống quản lý khóa học ngắn hạn, nhóm chúng tôi đã thực hiện khảo sát tại Trường SICT – Trường Đại học Công nghiệp Hà Nội (HaUI). Trung tâm chuyên tổ chức các khóa học đào tạo ngắn hạn về công nghệ thông tin, điện – điện tử, tự động hóa… phục vụ cho sinh viên và người học bên ngoài trường.

Thông tin đơn vị khảo sát:

* Tên đơn vị: Trung tâm SICT – HaUI
* Địa chỉ: Khu A, Trường Đại học Công nghiệp Hà Nội
* Cơ cấu tổ chức:
  + Ban quản lý trung tâm
  + Bộ phận tuyển sinh và hỗ trợ học viên
  + Bộ phận đào tạo và giảng viên
  + Bộ phận kỹ thuật

#### **2.1.2 Hiện trạng hoạt động và các vấn đề tồn tại**

Hệ thống quản lý khóa học tại Trung tâm SICT hiện tại hoạt động theo quy trình sau:

1. Đăng ký khóa học
   * Học viên đăng ký trực tiếp tại trung tâm hoặc qua biểu mẫu Google Form.
   * Bộ phận tuyển sinh kiểm tra và nhập thông tin vào Excel.
   * Xác nhận đăng ký qua email hoặc điện thoại.
2. Phân công giảng viên
   * Bộ phận đào tạo tra cứu danh sách giảng viên phù hợp từ file lưu trữ.
   * Sắp xếp giảng viên cho từng khóa học, thông báo qua email.
3. Tổ chức lớp học
   * Lịch học được thông báo qua nhóm Zalo hoặc email.
   * Giảng viên tự chuẩn bị tài liệu và bài giảng.
4. Theo dõi tiến độ và đánh giá học viên
   * Học viên nộp bài tập qua Google Drive hoặc email.
   * Điểm số và nhận xét được lưu trên file Excel.
5. Báo cáo và thống kê
   * Mỗi tháng, trung tâm tổng hợp dữ liệu bằng tay để đánh giá hiệu quả đào tạo.

Các vấn đề tồn tại:

* Dữ liệu quản lý phân tán, dễ mất mát.
* Thủ công hóa nhiều bước gây mất thời gian.
* Khó khăn trong việc tra cứu thông tin nhanh chóng.
* Không có hệ thống báo cáo tự động.

#### **2.1.3 Nhu cầu xây dựng hệ thống mới**

Dựa trên khảo sát, trung tâm mong muốn xây dựng một hệ thống quản lý khóa học trực tuyến, đáp ứng các yêu cầu sau:

1. Tin học hóa toàn bộ quy trình: Đăng ký, quản lý học viên, giảng viên, lịch học.
2. Tích hợp hệ thống đăng ký trực tuyến: Cho phép học viên đăng ký và nhận thông tin xác nhận ngay lập tức.
3. Quản lý giảng viên: Lưu trữ thông tin giảng viên, hỗ trợ phân công giảng dạy tự động.
4. Quản lý nội dung khóa học: Lưu trữ tài liệu, bài tập và kết quả học tập của học viên.
5. Thống kê và báo cáo: Cung cấp báo cáo tự động về số lượng học viên, hiệu quả giảng dạy.

#### **2.1.4 Phương pháp khảo sát**

Em đã sử dụng các phương pháp khảo sát sau để thu thập thông tin:

1. Phỏng vấn trực tiếp
   * Đối tượng: Nhân viên trung tâm, giảng viên, học viên.
   * Nội dung:
     + Hiện trạng hoạt động và khó khăn gặp phải.
     + Mong muốn và yêu cầu đối với hệ thống mới.
   * Kết quả thu thập:
     + Nhân viên trung tâm mong muốn hệ thống có tính năng nhập liệu nhanh, dễ theo dõi.
     + Giảng viên cần hệ thống hỗ trợ tải lên bài giảng, chấm điểm dễ dàng.
     + Học viên mong muốn đăng ký nhanh chóng, xem tài liệu thuận tiện.
2. Xây dựng bảng hỏi
   * Mục tiêu: Xác định các tính năng quan trọng cần có trong hệ thống.
   * Nội dung câu hỏi:
     + Bạn gặp khó khăn gì khi quản lý khóa học hiện tại?
     + Bạn mong muốn hệ thống có những tính năng nào?
   * Kết quả: Hơn 80% nhân viên mong muốn tích hợp hệ thống quản lý tập trung thay vì dùng Excel.
3. Thu thập tài liệu hiện tại
   * Biểu mẫu đăng ký học viên.
   * File Excel quản lý giảng viên và học viên.
   * Email xác nhận lịch học.

#### **2.1.5 Kết quả khảo sát**

Sau khi khảo sát, nhóm thu được các kết luận quan trọng:

* Trung tâm SICT cần một hệ thống quản lý khóa học trực tuyến để tối ưu hóa quy trình làm việc.
* Hệ thống cần có các phân quyền rõ ràng: Quản trị viên, Giảng viên, Học viên.
* Dữ liệu cần được tập trung, truy xuất nhanh chóng.
* Hệ thống phải hỗ trợ đăng ký khóa học, phân công giảng viên, quản lý tài liệu, đánh giá học viên.
* Cần có tính năng báo cáo thống kê tự động để đánh giá hiệu quả đào tạo.

#### **2.1.6 Định hướng xây dựng hệ thống**

Dựa trên khảo sát, hệ thống mới sẽ được xây dựng với các chức năng chính:

1. Quản lý khóa học: Thêm, sửa, xóa khóa học, phân loại theo lĩnh vực.
2. Quản lý học viên: Đăng ký trực tuyến, theo dõi tiến độ học tập.
3. Quản lý giảng viên: Lưu trữ thông tin, phân công giảng dạy.
4. Quản lý tài liệu và bài giảng: Hỗ trợ giảng viên đăng tải tài liệu.
5. Đánh giá và chấm điểm: Học viên nộp bài, giảng viên chấm điểm trực tuyến.
6. Thống kê và báo cáo: Tổng hợp dữ liệu về số lượng khóa học, giảng viên, học viên.

### **2.2 Đặc tả yêu cầu hệ thống phần mềm**

#### **2.2.1 Yêu cầu chức năng**

* Quản lý khóa học:
  + Cho phép thêm, chỉnh sửa và xóa thông tin về các khóa học.
  + Hiển thị danh sách khóa học đang hoạt động.
  + Quản lý thông tin chi tiết của từng khóa học như mô tả, thời gian học, giảng viên phụ trách.
* Quản lý học viên:
  + Đăng ký và quản lý thông tin học viên.
  + Ghi danh hoặc hủy ghi danh học viên vào các khóa học.
  + Theo dõi tiến độ học tập và điểm số của học viên.
* Quản lý giảng viên:
  + Quản lý thông tin giảng viên và phân công giảng viên vào các khóa học.
  + Cập nhật thông tin giảng viên, bao gồm chuyên môn và kinh nghiệm giảng dạy.
  + Quản lý bài giảng, bài tập và đánh giá kết quả học tập của học viên.
* Quản lý bài tập và điểm số:
  + Cho phép giảng viên tạo, chỉnh sửa và xóa bài tập trong từng khóa học.
  + Học viên có thể nộp bài tập trực tuyến.
  + Giảng viên có thể chấm điểm, nhận xét và cập nhật kết quả bài tập.
* Thống kê và báo cáo:
  + Cung cấp các báo cáo về số lượng khóa học, số lượng học viên, giảng viên và hiệu quả đào tạo.
  + Hỗ trợ xuất báo cáo thống kê theo từng kỳ học hoặc từng khóa học.

#### **2.2.2 Yêu cầu phi chức năng**

Hệ thống cần đáp ứng các yêu cầu phi chức năng để đảm bảo hiệu suất và trải nghiệm người dùng:

* Hiệu năng:
  + Đảm bảo tốc độ truy xuất dữ liệu nhanh, ngay cả khi có nhiều người dùng truy cập đồng thời.
  + Hỗ trợ phân trang và bộ nhớ đệm (caching) để tối ưu hóa hiệu suất.
* Tính bảo mật:
  + Bảo vệ dữ liệu người dùng bằng cách sử dụng cơ chế xác thực và phân quyền.
  + Mã hóa thông tin nhạy cảm để tránh rò rỉ dữ liệu.
  + Kiểm soát quyền truy cập theo vai trò (Admin, Giảng viên, Học viên).
* Khả năng mở rộng:
  + Hệ thống cần có kiến trúc linh hoạt để dễ dàng mở rộng và bổ sung các chức năng mới.
  + Hỗ trợ tích hợp với các hệ thống bên ngoài như cổng thanh toán hoặc nền tảng học trực tuyến.
* Tính thân thiện với người dùng:
  + Giao diện trực quan, dễ sử dụng cho cả học viên, giảng viên và quản trị viên.
  + Thiết kế tương thích với nhiều thiết bị như máy tính, điện thoại di động và máy tính bảng.
  + Hỗ trợ đa ngôn ngữ để đáp ứng nhu cầu của người dùng đa dạng.

### **2.3 Phân tích và thiết kế hệ thống**

#### **2.3.1 Phân tích và mô hình hóa chức năng của hệ thống**

Các tác nhân (Actors) trong hệ thống quản lý khóa học:

**Quản trị viên:**

* Quản lý danh sách khóa học (thêm, chỉnh sửa, xóa).
* Quản lý thông tin giảng viên.
* Quản lý thông tin học viên.
* Xem và xuất báo cáo thống kê.

**Giảng viên:**

* Xem danh sách khóa học mình phụ trách.
* Cập nhật thông tin giảng dạy (chuyên môn, kinh nghiệm).
* Tạo, chỉnh sửa, xóa bài tập.
* Chấm điểm và nhận xét bài tập.
* Theo dõi tiến độ học tập của học viên.

**Học viên:**

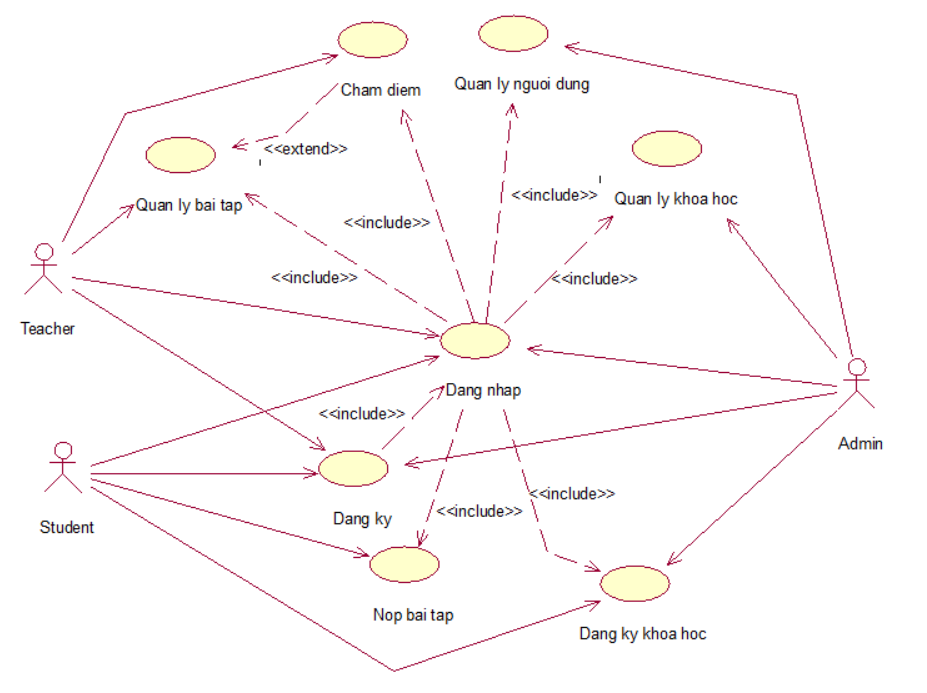
* Đăng ký và cập nhật thông tin cá nhân.
* Xem danh sách khóa học có sẵn.
* Ghi danh và hủy ghi danh khóa học.
* Nộp bài tập trực tuyến.
* Xem điểm số và nhận xét từ giảng viên.

**Hệ thống:**

* Tự động cập nhật báo cáo thống kê.
* Gửi thông báo về tiến độ học tập, bài tập và điểm số.
* Xác thực và phân quyền người dùng.

#### **2.3.2 Phân tích và mô hình hóa chức năng**

##### ***Biểu đồ use case tổng quát***

******

*Hình 2.1 Sơ đồ use case tổng quát*

##### **2.3.2.1 Use case “Đăng ký”**

**Mô tả:**

Use case này cho phép người dùng tạo tài khoản mới trong hệ thống để sử dụng các dịch vụ của hệ thống.

**Tác nhân:**

* Học viên
* Giảng viên
* Admin

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Người dùng truy cập vào trang đăng ký tài khoản của hệ thống.
2. Hệ thống hiển thị giao diện đăng ký với các trường thông tin cần điền (Họ tên, Email, Tên đăng nhập, Mật khẩu).
3. Người dùng điền đầy đủ thông tin vào các trường yêu cầu và nhấn nút "Đăng ký".
4. Hệ thống kiểm tra tính hợp lệ của thông tin nhập vào (kiểm tra định dạng email, tên đăng nhập, mật khẩu).
5. Nếu thông tin hợp lệ:
   * Hệ thống lưu thông tin người dùng vào cơ sở dữ liệu.
   * Hệ thống gửi email xác nhận đến người dùng.
6. Người dùng nhận email và kích hoạt tài khoản bằng cách click vào liên kết trong email.
7. Sau khi tài khoản được kích hoạt, người dùng được chuyển đến trang đăng nhập.

**Luồng rẽ nhánh**:

1. Nếu người dùng nhập thông tin thiếu hoặc thông tin không hợp lệ (ví dụ: email sai định dạng), hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.
2. Nếu email hoặc tên đăng nhập đã tồn tại trong hệ thống, hệ thống sẽ hiển thị thông báo yêu cầu người dùng chọn email hoặc tên đăng nhập khác.

**Yêu cầu đặc biệt:**

* Email đăng ký phải là duy nhất trong hệ thống.
* Mật khẩu cần có ít nhất 8 ký tự và bao gồm cả chữ cái và số.
* Hệ thống phải kiểm tra email hợp lệ và gửi liên kết xác nhận.

**Tiền điều kiện:**

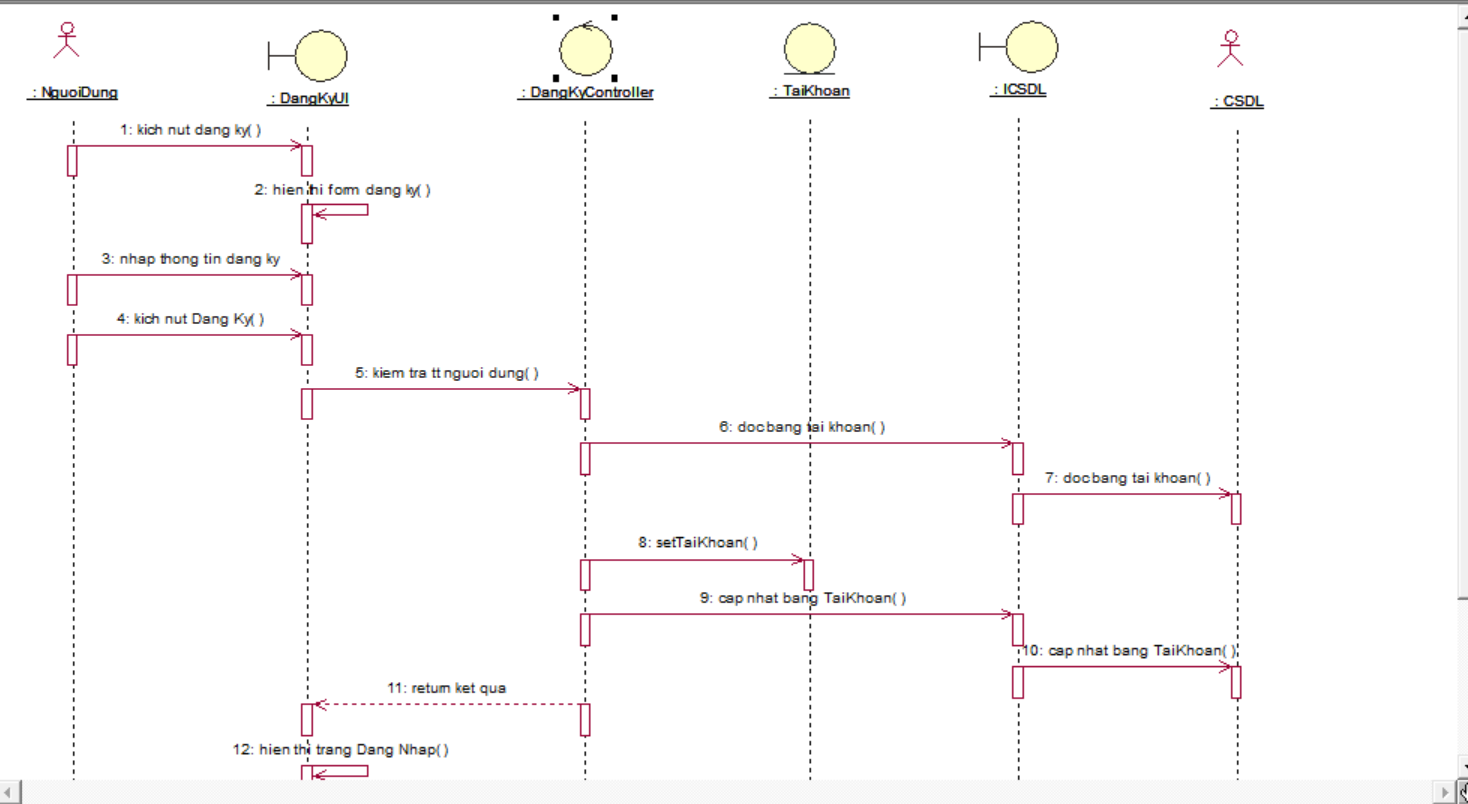
* Người dùng chưa có tài khoản trong hệ thống.

**Hậu điều kiện:**

* Tài khoản người dùng được tạo và lưu vào cơ sở dữ liệu, và tài khoản đã được kích hoạt thành công.

**Điểm mở rộng:**

* Tích hợp chức năng đăng ký qua mạng xã hội như Google, Facebook.



*Hình 2.2 Biểu đồ trình tự use case “Đăng ký”*

##### **2.3.2.2 Use case “Đăng nhập”**

**Mô tả:**

Use case này cho phép người dùng đăng nhập vào hệ thống bằng tài khoản đã đăng ký trước đó.

**Tác nhân:**

* Học viên
* Giảng viên
* Admin

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Người dùng truy cập vào trang đăng nhập của hệ thống.
2. Hệ thống yêu cầu người dùng nhập tên đăng nhập và mật khẩu.
3. Người dùng nhập thông tin đăng nhập và nhấn nút "Đăng nhập".
4. Hệ thống xác thực thông tin đăng nhập với dữ liệu trong cơ sở dữ liệu.
5. Nếu thông tin hợp lệ, hệ thống chuyển người dùng đến trang chủ của hệ thống và hiển thị thông tin cá nhân của người dùng.
6. Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại thông tin.

**Luồng rẽ nhánh**:

1. Nếu người dùng quên mật khẩu, hệ thống cung cấp chức năng "Quên mật khẩu", yêu cầu người dùng nhập email để nhận hướng dẫn đặt lại mật khẩu.
2. Nếu người dùng nhập sai mật khẩu hoặc tên đăng nhập, hệ thống hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.

**Yêu cầu đặc biệt:**

* Mật khẩu phải được mã hóa khi lưu trữ và truyền tải.
* Hệ thống phải sử dụng HTTPS để bảo mật thông tin người dùng.

**Tiền điều kiện:**

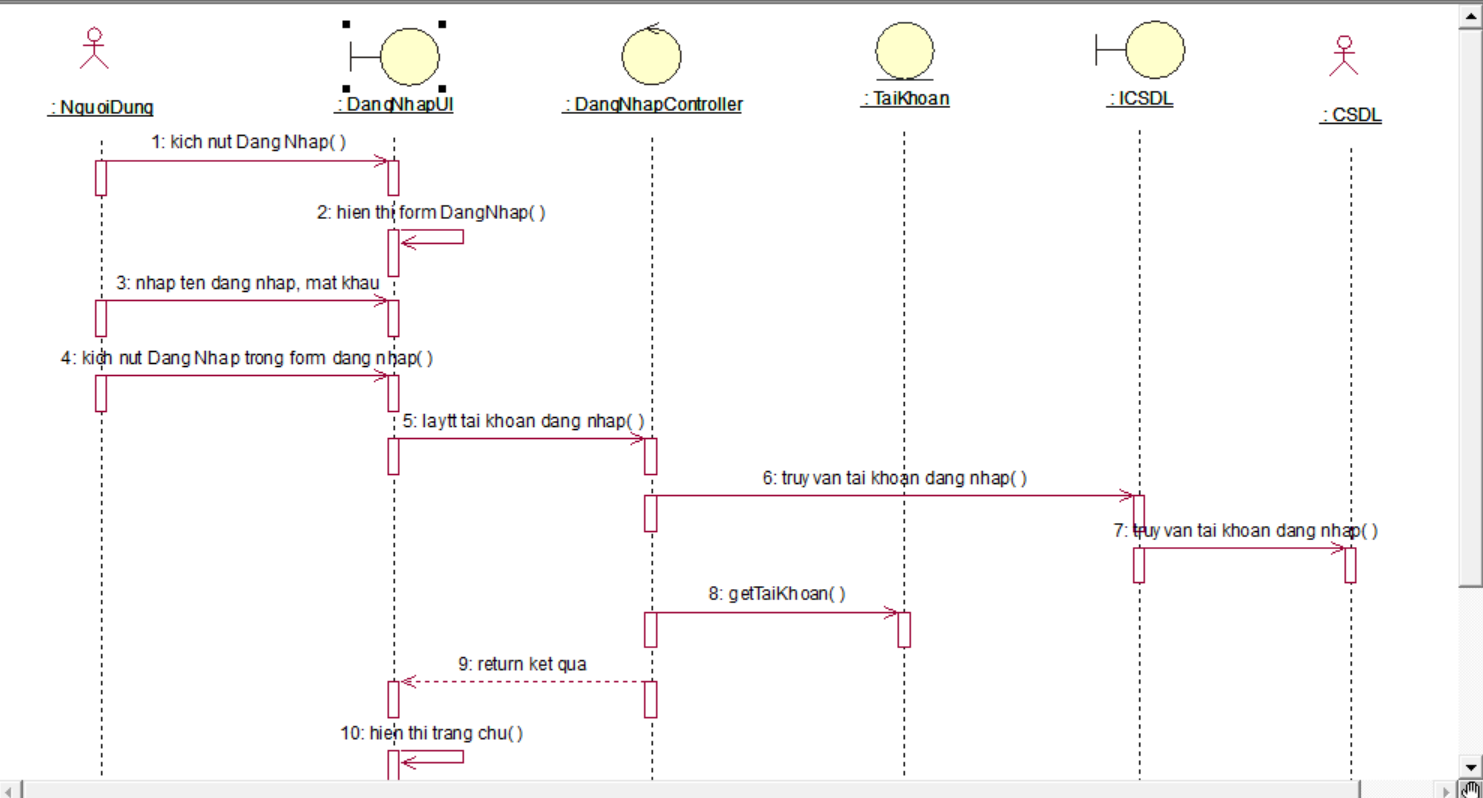
* Người dùng đã có tài khoản và đã đăng ký thành công trong hệ thống.

**Hậu điều kiện:**

* Người dùng được chuyển đến trang chủ và có thể thực hiện các hành động phù hợp với vai trò của mình (Học viên, Giảng viên, Admin).

**Điểm mở rộng:**

* Tích hợp chức năng đăng nhập qua mạng xã hội như Google, Facebook.
* Nếu có kết nối không an toàn, hệ thống sẽ yêu cầu người dùng cập nhật mật khẩu hoặc xác nhận lại kết nối.



*Hình 2.3 Biểu đồ trình tự use case “Đăng nhập”*

##### **2.3.2.3 Use case “Quản lý người dùng”**

**Mô tả:**

Use case này cho phép admin hoặc người quản lý quản lý thông tin người dùng, bao gồm việc thêm, sửa và xóa người dùng trong hệ thống.

**Tác nhân:**

* Admin
* Người quản lý hệ thống

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Người quản lý truy cập vào trang quản lý người dùng.
2. Hệ thống hiển thị danh sách tất cả người dùng trong hệ thống, bao gồm tên, email, vai trò, trạng thái tài khoản, và các hành động (sửa, xóa).
3. Người quản lý có thể chọn hành động thêm người dùng mới.
4. Hệ thống hiển thị một biểu mẫu để điền thông tin người dùng mới (họ tên, email, mật khẩu, vai trò).
5. Người quản lý nhập đầy đủ thông tin và nhấn "Thêm người dùng".
6. Hệ thống kiểm tra tính hợp lệ của thông tin và thêm người dùng mới vào cơ sở dữ liệu.
7. Hệ thống thông báo thành công và hiển thị người dùng mới trong danh sách.
8. Người quản lý có thể chọn hành động "Sửa" để thay đổi thông tin người dùng.
9. Hệ thống hiển thị giao diện sửa đổi thông tin, người quản lý cập nhật thông tin và nhấn "Lưu thay đổi".
10. Hệ thống cập nhật thông tin vào cơ sở dữ liệu và thông báo thành công.
11. Người quản lý có thể chọn hành động "Xóa" để xóa người dùng khỏi hệ thống.
12. Hệ thống yêu cầu xác nhận xóa, nếu xác nhận, người dùng sẽ bị xóa khỏi cơ sở dữ liệu và hệ thống thông báo thành công.

**Luồng rẽ nhánh**:

1. Nếu người quản lý không nhập đầy đủ thông tin khi thêm người dùng mới, hệ thống sẽ yêu cầu điền đủ các trường bắt buộc và không thêm người dùng cho đến khi thông tin hợp lệ.
2. Nếu không thể kết nối với cơ sở dữ liệu khi thao tác, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người quản lý thử lại sau.

**Tiền điều kiện:**

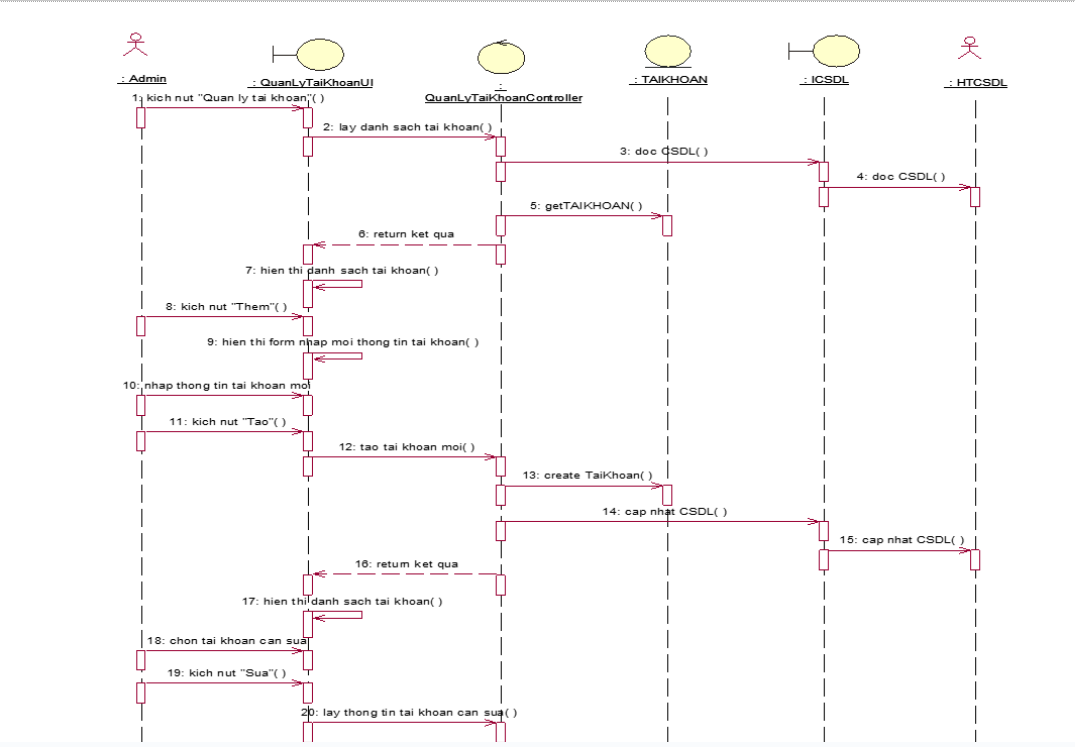
* Người quản lý phải đăng nhập vào hệ thống với vai trò admin hoặc người quản lý.
* Người quản lý có quyền truy cập vào trang quản lý người dùng.

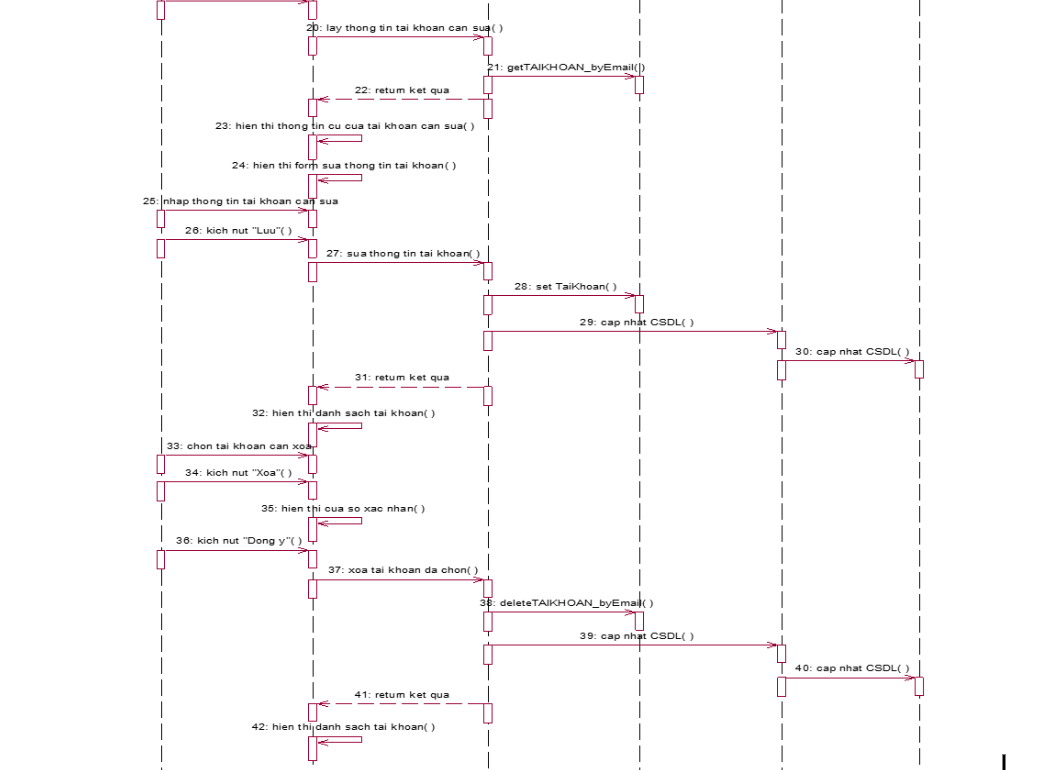
**Hậu điều kiện:**

* Người dùng mới được thêm, sửa, hoặc xóa trong hệ thống.

**Điểm mở rộng:**

* Có thể thêm các chức năng phức tạp hơn như phân quyền người dùng, thay đổi trạng thái tài khoản (kích hoạt/khóa tài khoản).





*Hình 2.4 Biểu đồ trình tự use case “Quản lý tài khoản”*

##### **2.3.2.4 Use case “Quản lý khóa học”**

**Mô tả:**

Use case này cho phép admin hoặc giảng viên quản lý các khóa học trong hệ thống, bao gồm việc thêm, sửa, xóa khóa học.

**Tác nhân:**

* Admin
* Giảng viên

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Người quản lý (admin/giảng viên) truy cập vào trang quản lý khóa học.
2. Hệ thống hiển thị danh sách các khóa học hiện có trong hệ thống.
3. Người quản lý có thể chọn hành động thêm khóa học mới.
4. Hệ thống hiển thị biểu mẫu để nhập thông tin khóa học mới (Tên khóa học, Mã khóa học, Mô tả khóa học, Thời gian bắt đầu, Thời gian kết thúc).
5. Người quản lý nhập đầy đủ thông tin khóa học và nhấn "Thêm khóa học".
6. Hệ thống kiểm tra tính hợp lệ của thông tin và thêm khóa học mới vào cơ sở dữ liệu.
7. Hệ thống thông báo thành công và hiển thị khóa học mới trong danh sách.
8. Người quản lý có thể chọn hành động "Sửa" để thay đổi thông tin khóa học.
9. Hệ thống hiển thị giao diện sửa khóa học, người quản lý cập nhật thông tin và nhấn "Lưu thay đổi".
10. Hệ thống cập nhật thông tin vào cơ sở dữ liệu và thông báo thành công.
11. Người quản lý có thể chọn hành động "Xóa" để xóa khóa học khỏi hệ thống.
12. Hệ thống yêu cầu xác nhận xóa, nếu xác nhận, khóa học sẽ bị xóa khỏi cơ sở dữ liệu và hệ thống thông báo thành công.

**Luồng rẽ nhánh**:

1. Nếu người quản lý không nhập đầy đủ thông tin khi thêm khóa học, hệ thống sẽ yêu cầu điền đầy đủ các trường bắt buộc và không thêm khóa học cho đến khi thông tin hợp lệ.
2. Nếu không thể kết nối với cơ sở dữ liệu khi thao tác, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu người quản lý thử lại sau.

**Tiền điều kiện:**

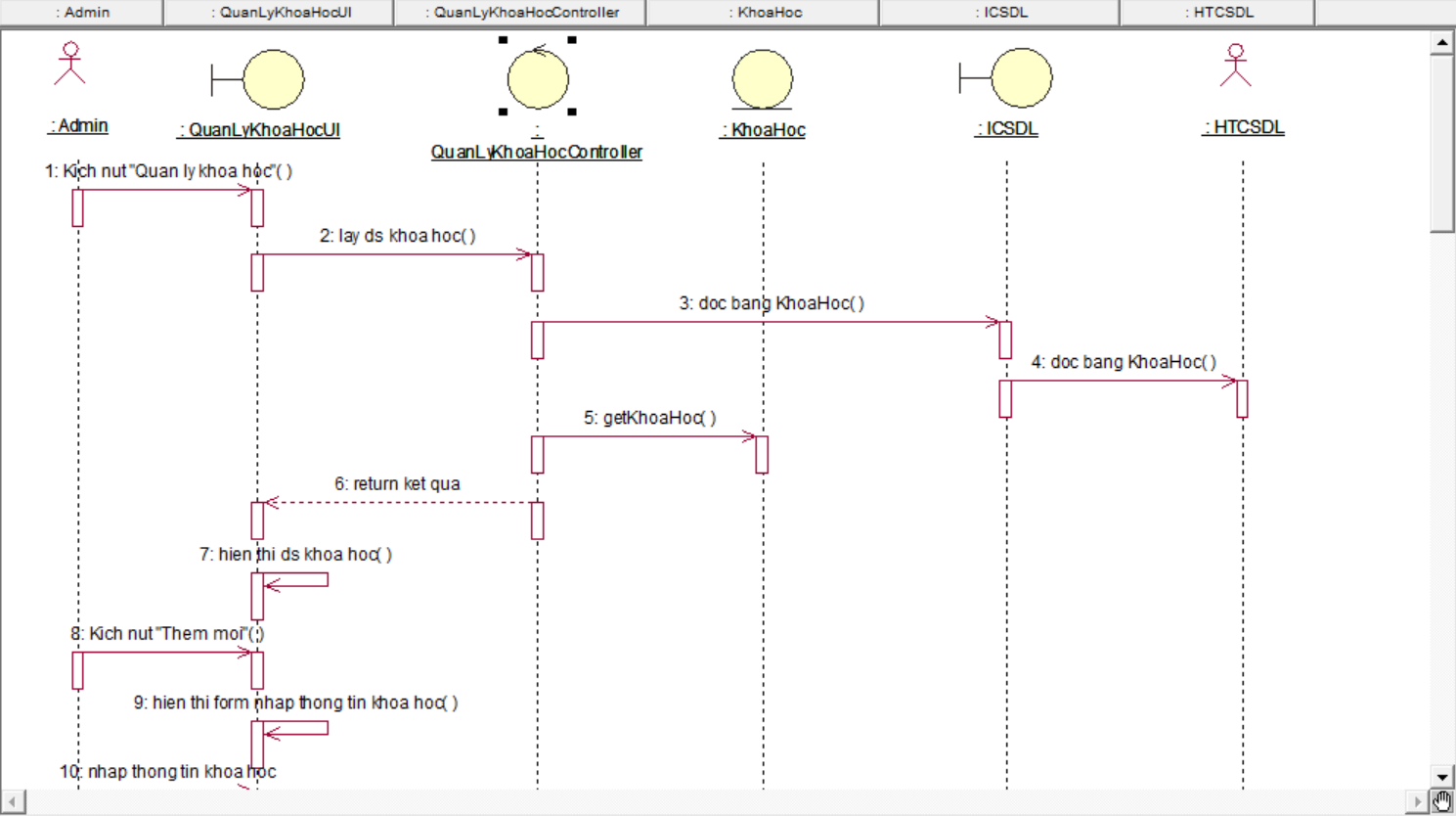
* Người quản lý phải đăng nhập vào hệ thống với vai trò admin hoặc giảng viên.
* Người quản lý có quyền truy cập vào trang quản lý khóa học.

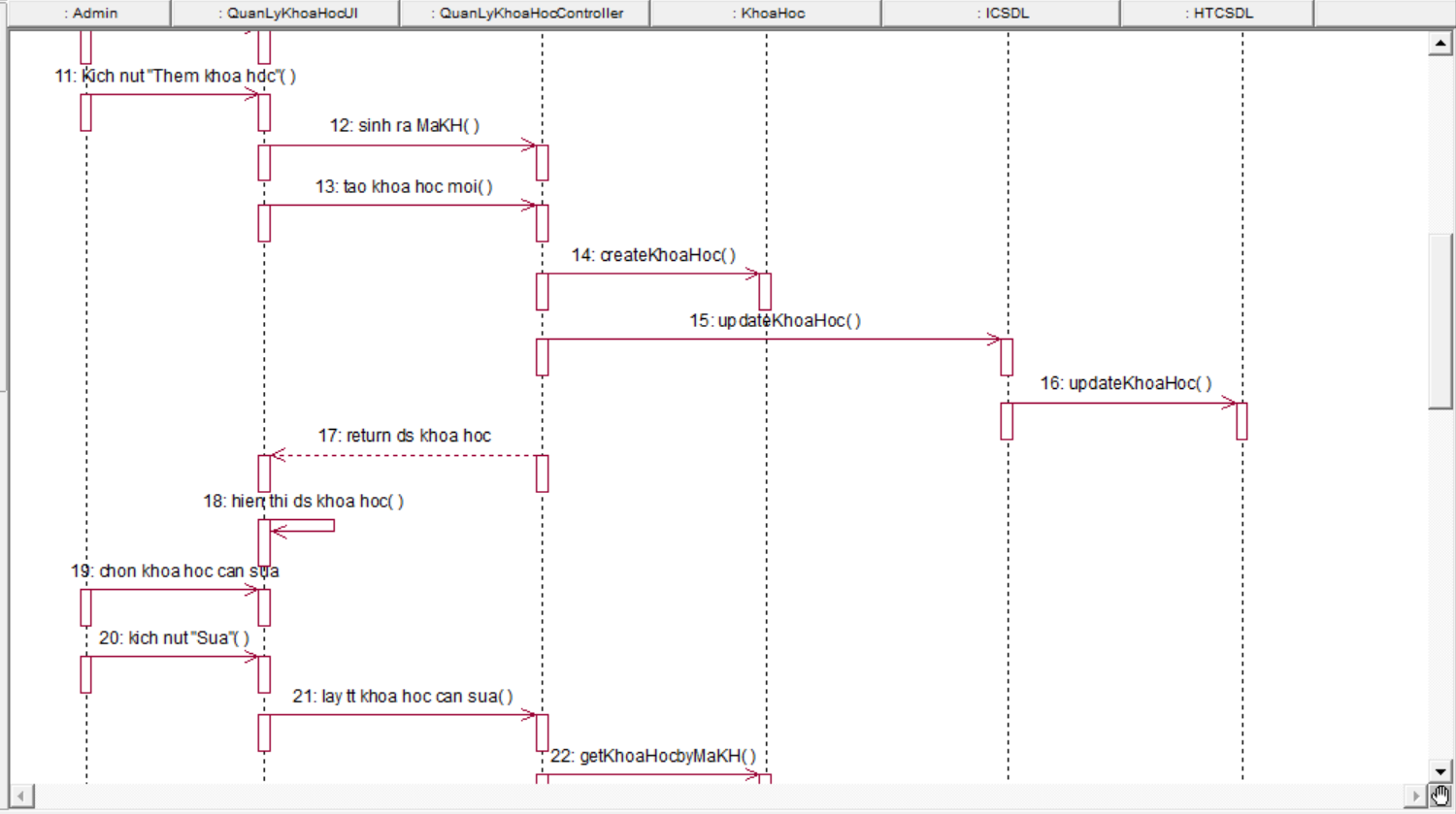
**Hậu điều kiện:**

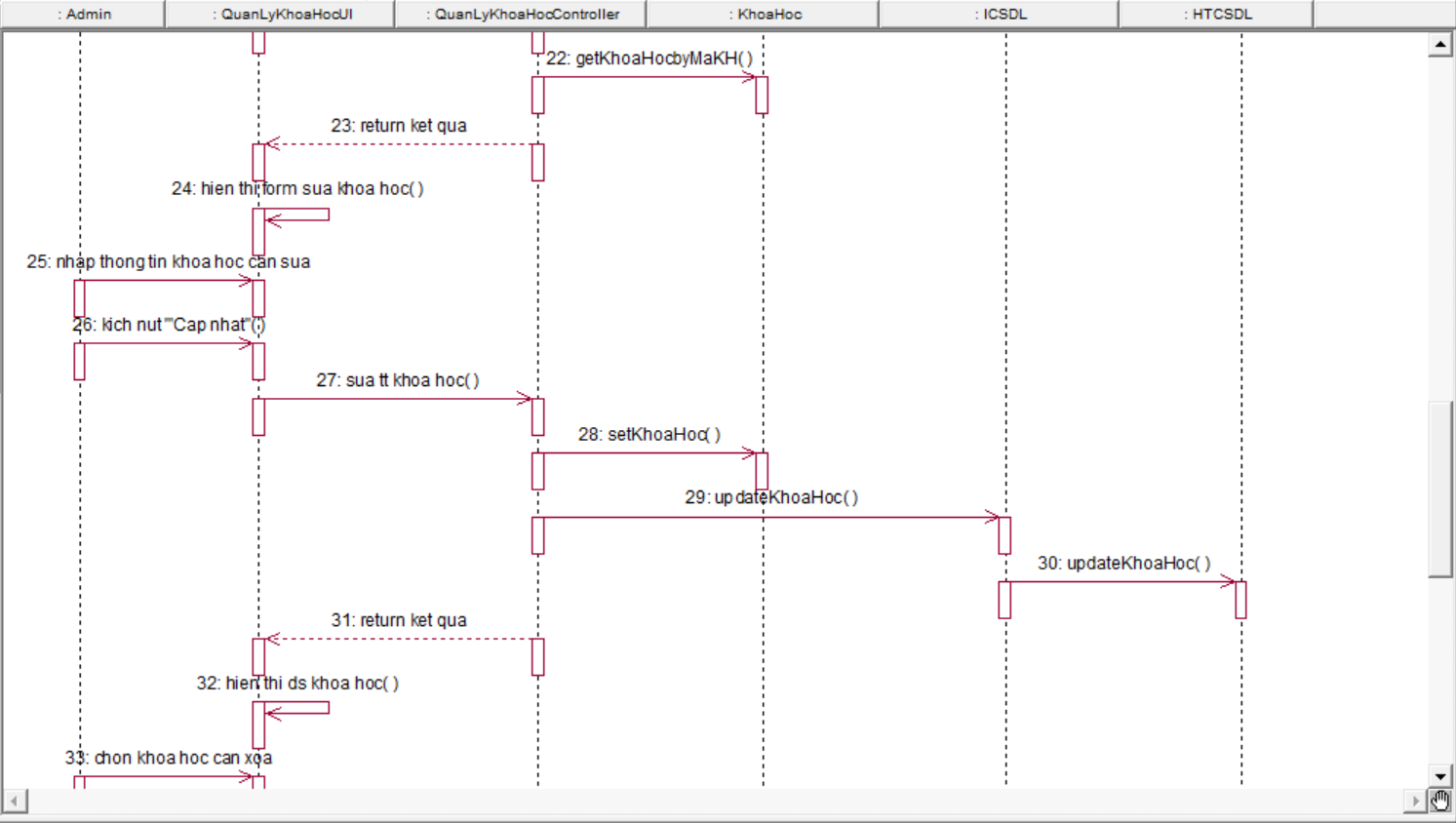
* Khóa học mới được thêm, sửa, hoặc xóa trong hệ thống.

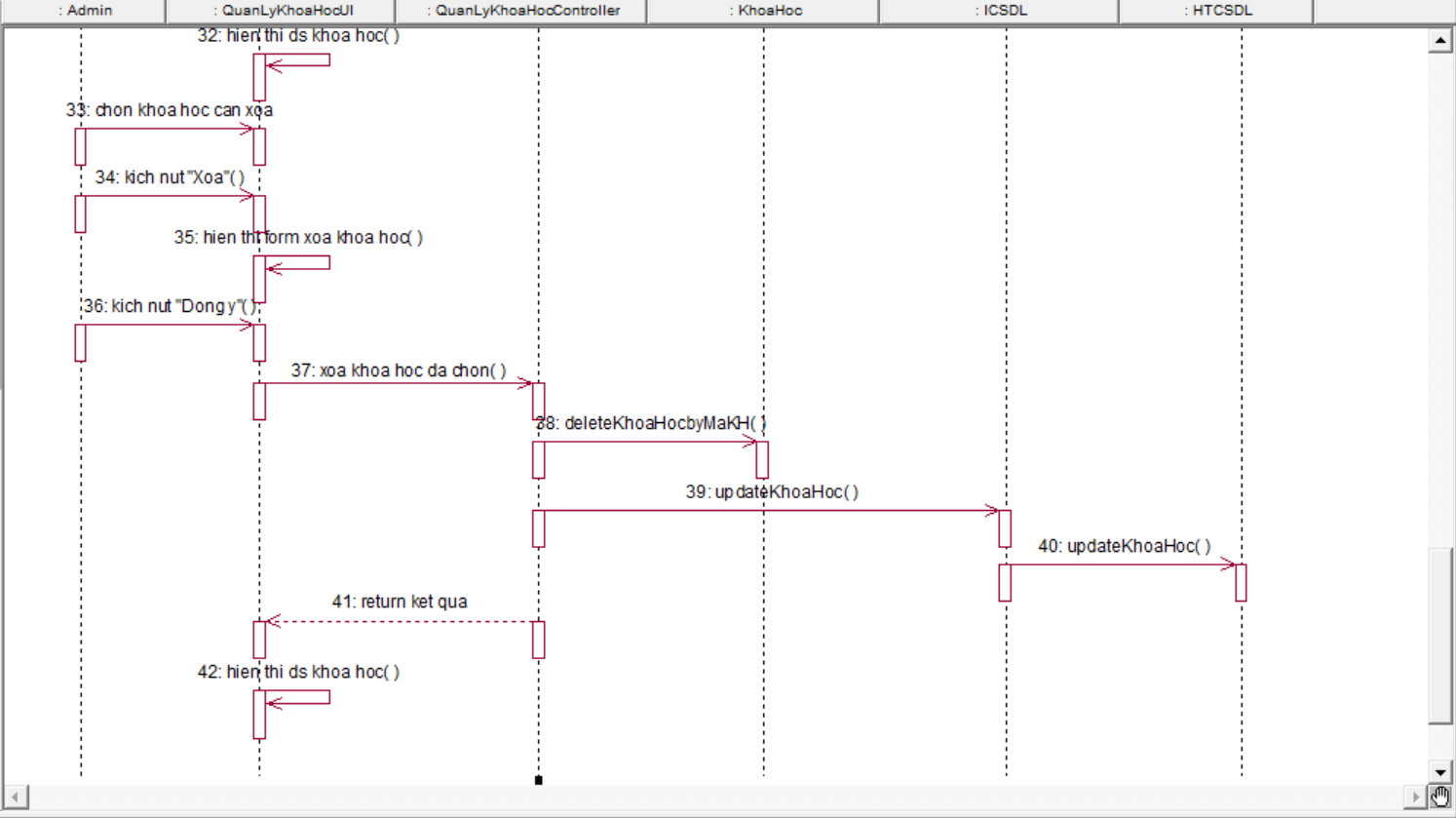
**Điểm mở rộng:**

* Có thể bổ sung thêm tính năng đăng ký khóa học cho sinh viên hoặc quản lý lịch học.

**

**

**

**

*Hình 2.5 Biểu đồ trình tự use case “Quản lý khóa học”*

##### **2.3.2.5 Use case “Đăng ký khóa học”**

**Mô tả:**

Use case này cho phép sinh viên đăng ký vào các khóa học mà họ muốn tham gia trong hệ thống.

**Tác nhân:**

* Sinh viên

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Sinh viên đăng nhập vào hệ thống.
2. Sinh viên truy cập vào trang "Khóa học" và xem danh sách các khóa học hiện có.
3. Hệ thống hiển thị danh sách các khóa học với các thông tin như tên khóa học, mô tả, thời gian bắt đầu, thời gian kết thúc.
4. Sinh viên chọn một khóa học muốn tham gia và nhấn vào nút "Đăng ký".
5. Hệ thống yêu cầu sinh viên xác nhận thông tin khóa học mà họ muốn đăng ký.
6. Sinh viên xác nhận đăng ký khóa học và nhấn "Xác nhận".
7. Hệ thống kiểm tra xem sinh viên đã đăng ký khóa học đó hay chưa.
   * Nếu chưa, hệ thống sẽ thêm khóa học vào danh sách khóa học đã đăng ký của sinh viên và hiển thị thông báo thành công.
   * Nếu đã đăng ký, hệ thống sẽ hiển thị thông báo rằng sinh viên đã tham gia khóa học và không thể đăng ký lại.
8. Sinh viên có thể quay lại trang khóa học hoặc tiếp tục duyệt các khóa học khác.

**Luồng rẽ nhánh**:

1. Nếu sinh viên chưa đăng nhập vào hệ thống, họ sẽ được yêu cầu đăng nhập trước khi có thể thực hiện đăng ký khóa học.
2. Nếu không có kết nối với cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu sinh viên thử lại sau.

**Tiền điều kiện:**

* Sinh viên phải đăng nhập vào hệ thống.
* Sinh viên chưa đăng ký khóa học này.

**Hậu điều kiện:**

* Sinh viên sẽ được ghi nhận là đã đăng ký khóa học trong hệ thống.

**Điểm mở rộng:**

* Sinh viên có thể bỏ đăng ký khóa học, hệ thống sẽ yêu cầu xác nhận và cập nhật lại danh sách khóa học của sinh viên.



*Hình 2.6 Biểu đồ trình tự use case “Đăng ký khóa học”*

##### **2.3.2.6 Use case “Quản lý bài tập”**

**Mô tả:**

Use case này cho phép giảng viên quản lý bài tập trong các khóa học, bao gồm việc tạo, chỉnh sửa, và xóa bài tập.

**Tác nhân:**

* Giảng viên

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Giảng viên đăng nhập vào hệ thống với vai trò giảng viên.
2. Giảng viên truy cập vào trang quản lý khóa học của mình và chọn khóa học muốn quản lý bài tập.
3. Hệ thống hiển thị danh sách các bài tập hiện có trong khóa học.
4. Giảng viên có thể chọn thêm bài tập mới bằng cách nhấn nút "Thêm bài tập".
5. Hệ thống hiển thị một biểu mẫu để giảng viên nhập thông tin bài tập mới (tên bài tập, mô tả, hạn nộp, yêu cầu bài làm).
6. Giảng viên nhập đầy đủ thông tin bài tập và nhấn "Lưu bài tập".
7. Hệ thống kiểm tra tính hợp lệ của thông tin và lưu bài tập vào cơ sở dữ liệu.
8. Giảng viên có thể chọn hành động "Sửa" để chỉnh sửa bài tập đã tạo.
9. Hệ thống hiển thị giao diện chỉnh sửa bài tập, giảng viên cập nhật thông tin và nhấn "Lưu thay đổi".
10. Hệ thống cập nhật bài tập trong cơ sở dữ liệu và thông báo thành công.
11. Giảng viên có thể chọn hành động "Xóa" để xóa bài tập khỏi hệ thống.
12. Hệ thống yêu cầu giảng viên xác nhận hành động xóa, nếu xác nhận, bài tập sẽ bị xóa khỏi cơ sở dữ liệu và hệ thống thông báo thành công.

**Luồng rẽ nhánh**:

1. Nếu giảng viên không nhập đầy đủ thông tin khi thêm bài tập, hệ thống sẽ yêu cầu giảng viên điền đầy đủ thông tin và không thêm bài tập cho đến khi thông tin hợp lệ.
2. Nếu không thể kết nối với cơ sở dữ liệu khi thao tác, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu giảng viên thử lại sau.

**Tiền điều kiện:**

* Giảng viên phải đăng nhập vào hệ thống với vai trò giảng viên.
* Giảng viên có quyền quản lý khóa học và bài tập của khóa học đó.

**Hậu điều kiện:**

* Bài tập mới được tạo, sửa hoặc xóa trong hệ thống.

**Điểm mở rộng:**

* Có thể bổ sung chức năng gửi thông báo cho sinh viên khi có bài tập mới hoặc hạn nộp bài tập.



*Hình 2.7 Biểu đồ trình tự use case“Quản lý bài tập”*

##### **2.3.2.7 Use case “Nộp bài tập”**

**Mô tả:**

Use case này cho phép sinh viên nộp bài tập đã hoàn thành cho giảng viên.

**Tác nhân:**

* Sinh viên

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Sinh viên đăng nhập vào hệ thống.
2. Sinh viên truy cập vào trang "Khóa học" và chọn khóa học đã đăng ký.
3. Hệ thống hiển thị danh sách các bài tập trong khóa học.
4. Sinh viên chọn bài tập muốn nộp và nhấn vào tên bài tập đó.
5. Hệ thống hiển thị các thông tin chi tiết của bài tập, bao gồm yêu cầu và hạn nộp.
6. Sinh viên hoàn thành bài tập và chuẩn bị tệp tin nộp.
7. Sinh viên nhấn vào nút "Nộp bài" và hệ thống yêu cầu sinh viên tải lên tệp bài tập của mình.
8. Sinh viên chọn tệp và nhấn "Tải lên".
9. Hệ thống kiểm tra tính hợp lệ của tệp (định dạng, dung lượng, v.v.). Nếu hợp lệ, hệ thống sẽ lưu bài tập của sinh viên vào cơ sở dữ liệu và hiển thị thông báo "Nộp bài thành công".
10. Nếu sinh viên nộp bài muộn, hệ thống sẽ hiển thị cảnh báo về việc nộp bài sau thời hạn.
11. Sinh viên có thể quay lại trang khóa học hoặc tiếp tục với các tác vụ khác.

**Luồng rẽ nhánh**:

1. Nếu sinh viên chưa hoàn thành bài tập và cố gắng nộp, hệ thống sẽ hiển thị thông báo yêu cầu sinh viên hoàn thành bài tập trước khi nộp.
2. Nếu hệ thống không thể kết nối với cơ sở dữ liệu hoặc không thể tải lên tệp, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu sinh viên thử lại sau.

**Tiền điều kiện:**

* Sinh viên đã đăng nhập vào hệ thống.
* Sinh viên đã hoàn thành bài tập và có tệp bài tập để nộp.

**Hậu điều kiện:**

* Bài tập của sinh viên được nộp thành công và lưu vào cơ sở dữ liệu.

**Điểm mở rộng:**

* Sinh viên có thể nộp bài tập nhiều lần trong trường hợp bài tập cho phép nộp lại.



*Hình 2.8 Biểu đồ trình tự use case “Nộp bài tập”*

##### **2.3.2.8 Use case “Chấm điểm”**

**Mô tả:**

Use case này cho phép giảng viên chấm điểm bài tập của sinh viên sau khi sinh viên đã nộp bài.

**Tác nhân:**

* Giảng viên

**Luồng sự kiện:**

**Luồng cơ bản**:

1. Giảng viên đăng nhập vào hệ thống.
2. Giảng viên truy cập vào trang "Khóa học" và chọn khóa học muốn quản lý bài tập.
3. Hệ thống hiển thị danh sách các bài tập trong khóa học cùng với trạng thái nộp bài của sinh viên.
4. Giảng viên chọn bài tập muốn chấm điểm.
5. Hệ thống hiển thị danh sách sinh viên đã nộp bài tập và các tệp bài tập đã nộp.
6. Giảng viên chọn một sinh viên để chấm điểm.
7. Hệ thống hiển thị bài tập của sinh viên đó và các thông tin cần thiết (nếu có) như các yêu cầu, tài liệu tham khảo.
8. Giảng viên chấm điểm và cung cấp nhận xét cho bài tập của sinh viên.
9. Giảng viên nhập điểm vào hệ thống và nhấn "Lưu điểm".
10. Hệ thống cập nhật điểm và nhận xét của giảng viên vào cơ sở dữ liệu, đồng thời thông báo thành công.
11. Giảng viên có thể tiếp tục chấm điểm các bài tập khác hoặc quay lại trang khóa học.

**Luồng rẽ nhánh**:

1. Nếu giảng viên không thể kết nối với cơ sở dữ liệu, hệ thống sẽ hiển thị thông báo lỗi và yêu cầu giảng viên thử lại sau.
2. Nếu giảng viên không nhập điểm hoặc nhận xét đầy đủ, hệ thống sẽ yêu cầu nhập lại thông tin và không lưu điểm cho bài tập đó.

**Tiền điều kiện:**

* Giảng viên đã đăng nhập vào hệ thống.
* Sinh viên đã nộp bài tập cho giảng viên chấm điểm.

**Hậu điều kiện:**

* Điểm và nhận xét của giảng viên được lưu vào cơ sở dữ liệu và cập nhật cho sinh viên.

**Điểm mở rộng:**

* Hệ thống có thể gửi thông báo cho sinh viên khi bài tập của họ đã được chấm điểm và có điểm mới.



*Hình 2.9 Biểu đồ trình tự use case “Chấm điểm”*

### **2.4 Phân tích và mô hình hóa dữ liệu**

#### **2.4.1 Các thực thể chính và thuộc tính**

a. student (Học viên)

* + id: Khóa chính.
  + email, first\_name, last\_name, password, username: Thông tin đăng nhập và nhận diện.
  + role\_id: Liên kết đến bảng role để xác định vai trò của học viên.

b. teacher (Giảng viên)

* + id: Khóa chính.
  + email, first\_name, last\_name, password, username: Thông tin đăng nhập.
  + role\_id: Liên kết đến bảng role.

c. role (Vai trò)

* + id: Khóa chính.
  + name: Tên vai trò.

d. course (Khóa học)

* + id: Khóa chính.
  + code, name: Mã và tên khóa học.
  + teacher\_id: Liên kết đến bảng teacher để xác định giảng viên phụ trách.

e. student\_course\_details (Chi tiết khóa học của học viên)

* + id: Khóa chính.
  + course\_id: Liên kết đến course.
  + student\_id: Liên kết đến student.
  + grade\_details\_id: Liên kết đến grade\_details.

f. grade\_details (Chi tiết điểm số)

* + id: Khóa chính.
  + grade\_one, grade\_two, grade\_three: Các cột điểm của học viên.

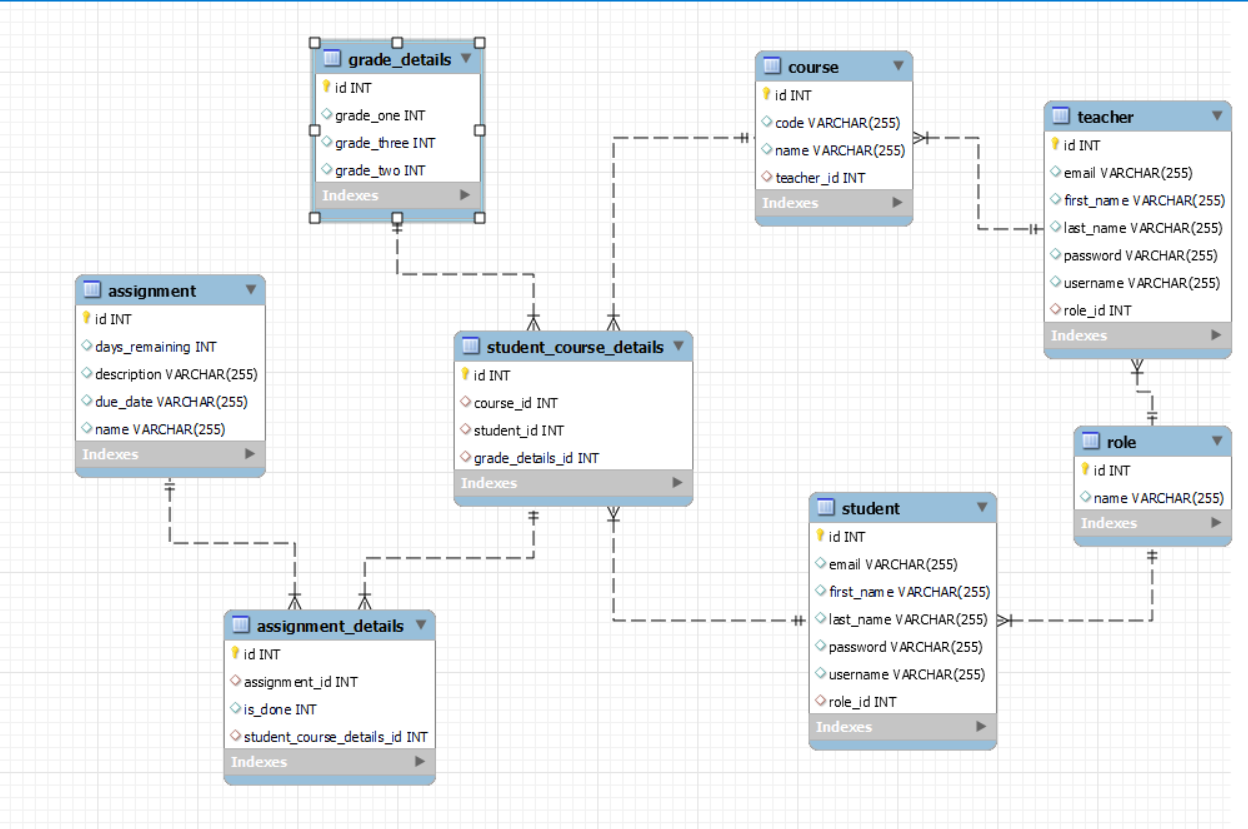
g. assignment (Bài tập)

* + id: Khóa chính.
  + name, description, due\_date, days\_remaining: Thông tin bài tập.

h. assignment\_details (Chi tiết bài tập)

* + id: Khóa chính.
  + assignment\_id: Liên kết đến bảng assignment.
  + student\_course\_details\_id: Liên kết đến student\_course\_details.
  + is\_done: Trạng thái hoàn thành.

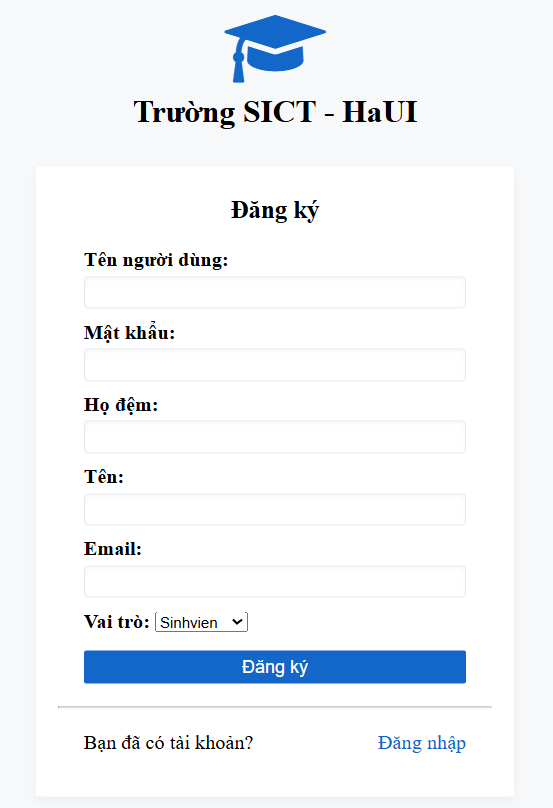
#### **2.4.2 Biểu đồ lớp thực thể liên kết**



*Hình 2.10 Biểu đồ lớp thực thể liên kết*

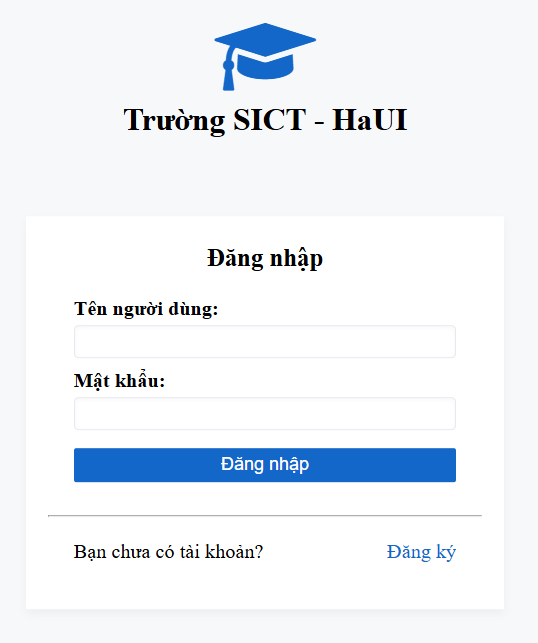
### **2.5 Thiết kế giao diện**

#### **2.5.1 Giao diện Đăng ký**

**

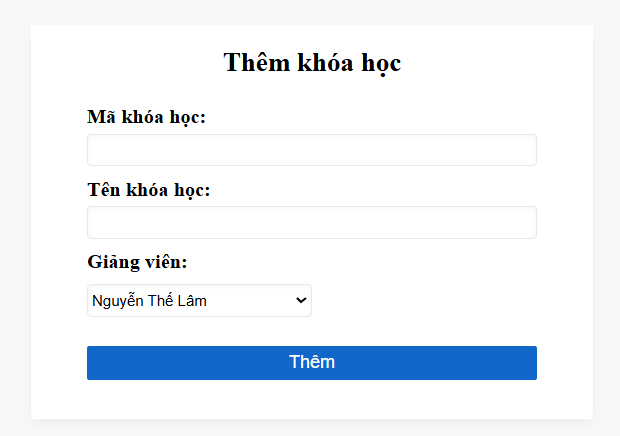
*Hình 2.11 Giao diện Đăng ký*

#### **2.5.2 Giao diện Đăng nhập**

**

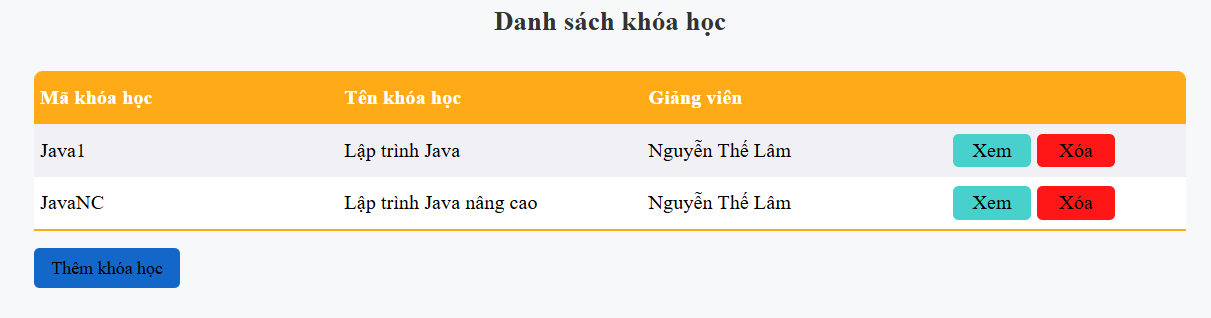
*Hình 2.12 Giao diện Đăng nhập*

#### **2.5.3 Giao diện Thêm khóa học**

**

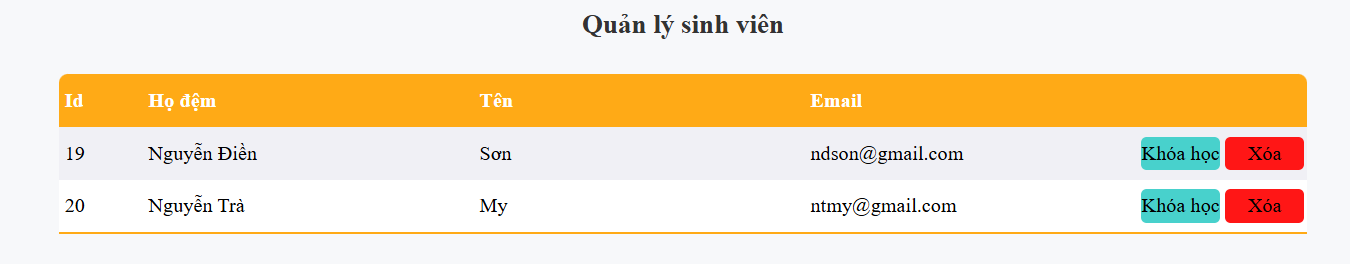
*Hình 2.13 Giao diện Thêm khóa học*

#### **2.5.4 Giao diện Chỉnh sửa khóa học**

**

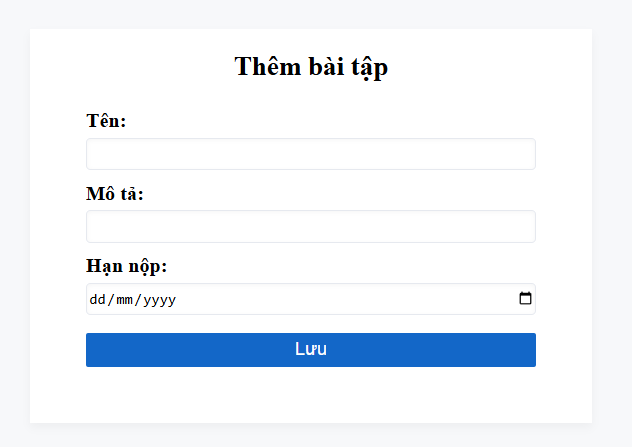
*Hình 2.14 Giao diện Chỉnh sửa khóa học*

#### **2.5.5 Giao diện Quản lý tài khoản**

**

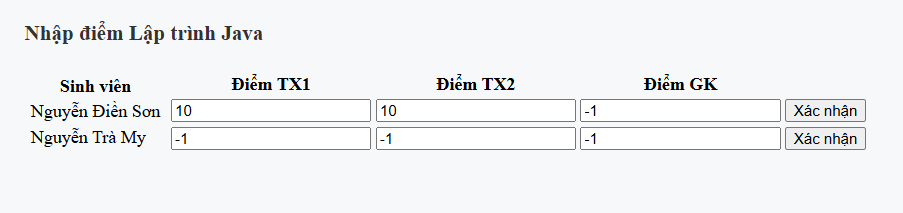
*Hình 2.15 Giao diện Quản lý tài khoản*

#### **2.5.6 Giao diện Thêm bài tập**

**

*Hình 2.16 Giao diện Thêm bài tập*

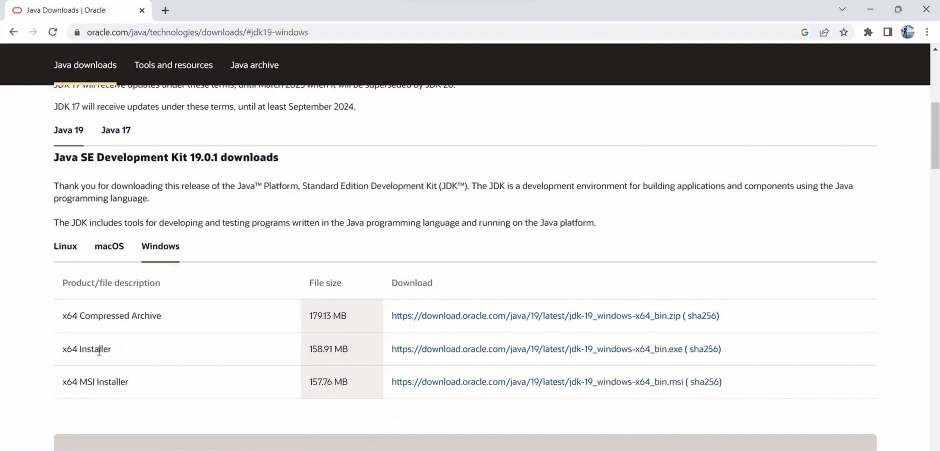
#### **2.5.7 Giao diện Nhập điểm**

**

*Hình 2.17 Giao diện Nhập điểm*

## **Chương 3. Cài đặt và thử nghiệm**

### **3.1 Cài đặt JDK**

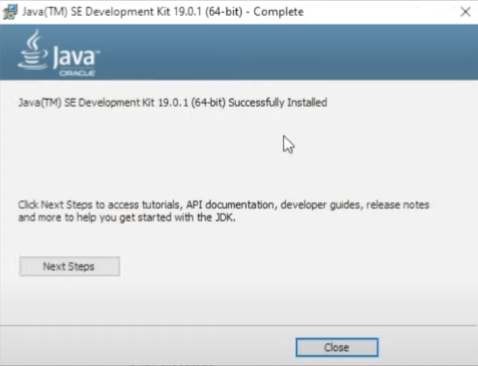
**Bước 1:** Truy cập https://[www.oracle.com/](http://www.oracle.com/) để có thể download bản JDK chính thức từ Oracle. Ở giao diện chính sẽ hiển thị phiên bản Java mới nhất.

**Bước 2:** Bạn chọn hệ điều hành đang dùng sau đó bấm vào đường link bên dưới, máy tính sẽ hiển thị giao diện yêu cầu bạn chọn nơi lưu trữ.

**Bước** **3:** Sau khi quá trình tải về hoàn tất, nhấp đúp vào file .exe để bắt đầu cài đặt. Chọn Next ở tất cả các bước cài đặt.

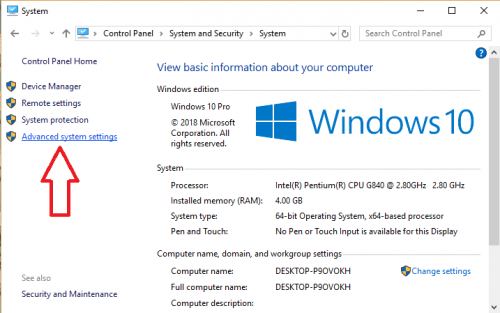


**Bước** **4**: Sau khi quá trình cài đặt kết thúc, bấm Close để đóng cửa sổ lại.

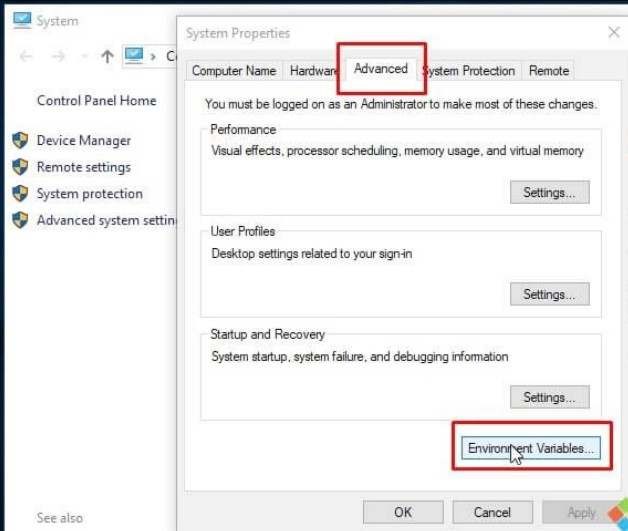


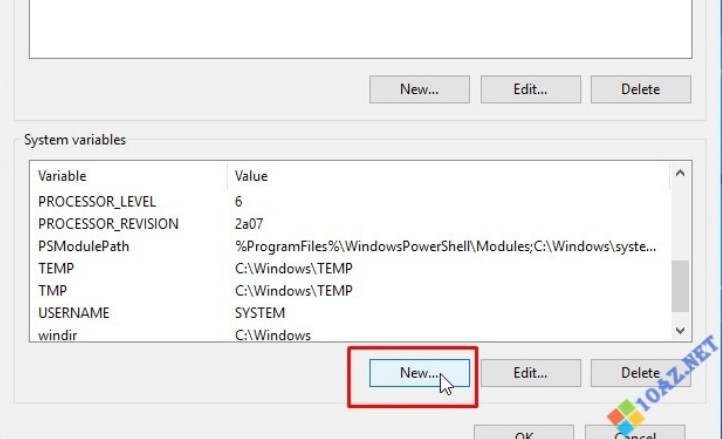
### **3.2 Thiết lập biến môi trường cho Java**

Bước 1: Nhấp chuột phải vào This PC sau đó chọn Properties để truy cập vào System, trong cửa sổ mới bạn nhấp chọn Advanced system settings từ danh mục bên trái.



Chọn Advanced system settings

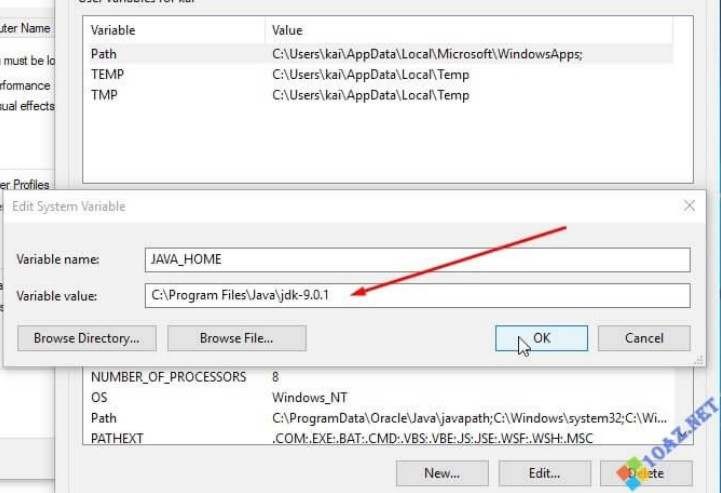
**Bước 2:** Cửa sổ System Properties hiện ra, chuyển qua tab Advanced và bấm vào nút Enviroment Variables ở phía dưới cửa sổ.

Bấm Enviroment Variables Bước 3: Trong cửa sổ Enviroment Variables, tìm tới mục System variables phía dưới và bấm New để tạo biến môi trường mới.

Tạo biến mới

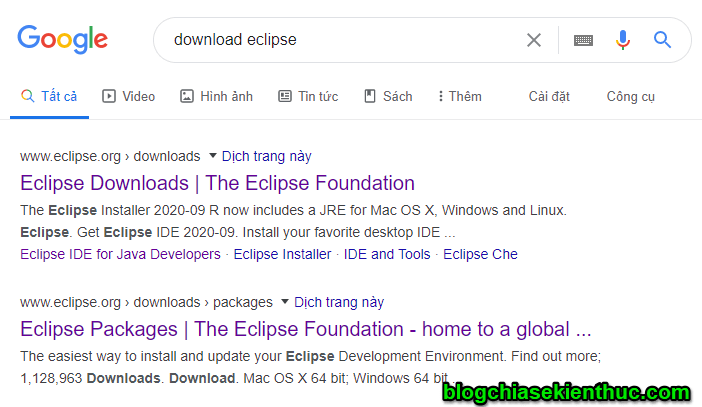
**Bước 4:** Khung cài đặt mới hiện ra, bạn nhập:

* JAVA\_HOME vào ô Variable name.
* Truy cập vào thư mục cài đặt của Java, copy đường dẫn và dán vào ô Varialbe value (mục này tuỳ hệ điều hành và phiên bản Java nhưng thường có dạng C:\Program Files\Java\xxxx).

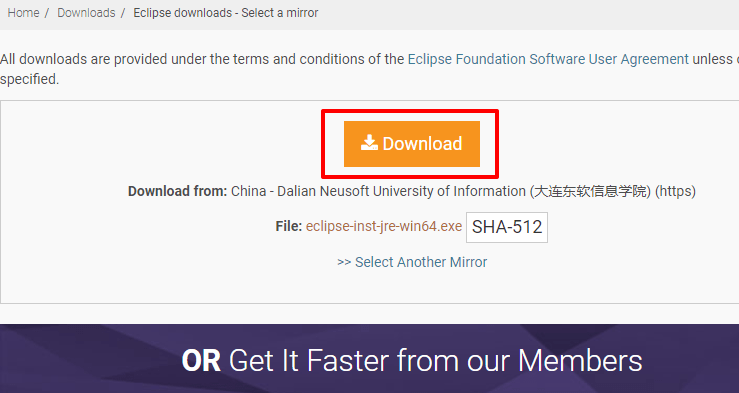


### **3.3 Cài đặt Eclipse**

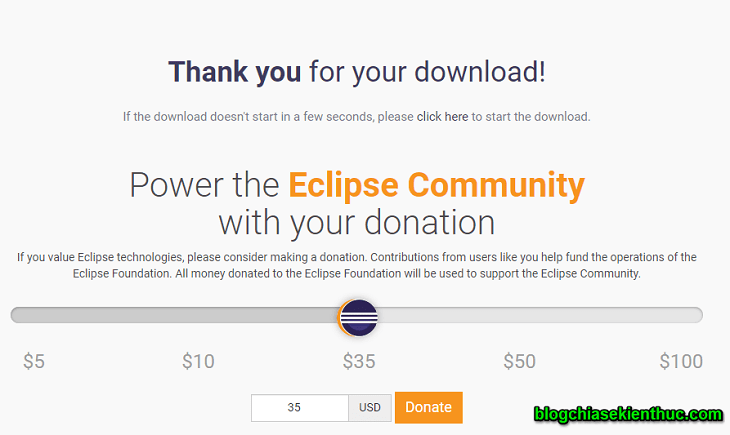
**Bước 1:** Bước 1: Các bạn Search Google với từ khóa “download eclipse”. Hoặc follow theo link này để đến trang download.



**Bước 2:** Bấm vào Download để tải file cài đặt về.

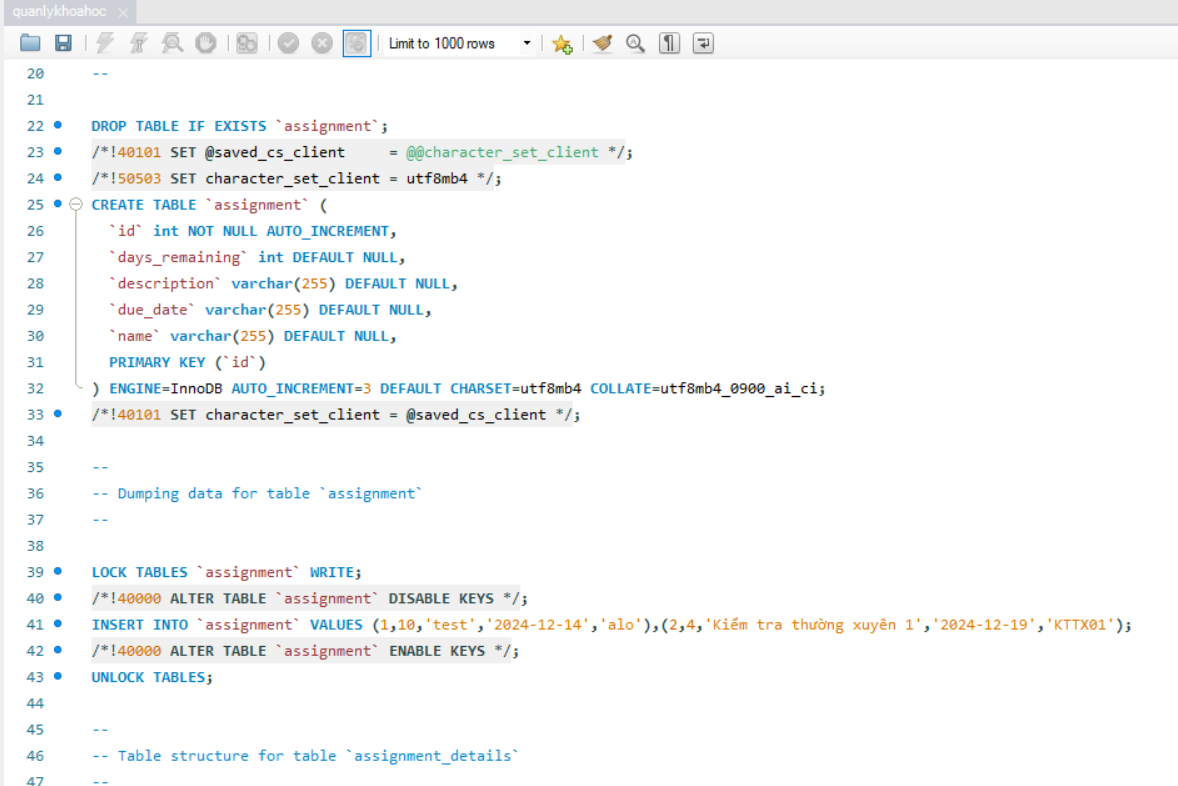


**Bước 3:** Sau khi download về xong thì màn hình sẽ hiển thị như thế này (nếu bạn có tiền thì có thể donate để cộng đồng phát triển công cụ này tốt hơn).

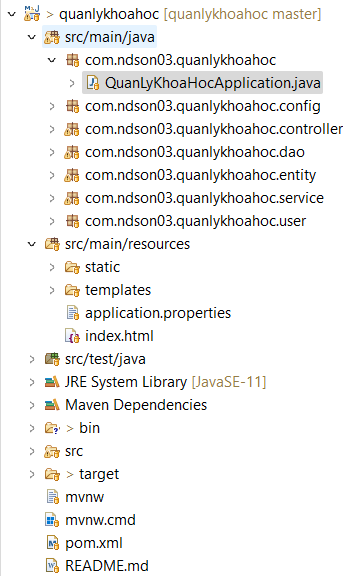


### **3.4 Chạy dự án**

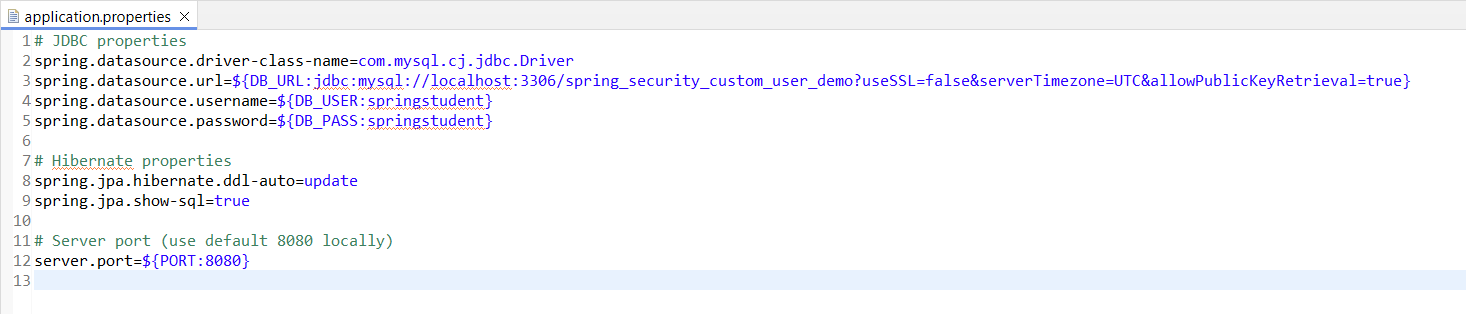
Chạy script SQL:

****

Mở project bằng Eclipse:

****

Cấu hình lại application.properties:



Chạy dự án:



# **PHẦN III. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **1. Nội dung đã thực hiện được**

Trong quá trình xây dựng dự án web quản lý khóa học, các nội dung và chức năng chính đã được hoàn thành bao gồm:

* Chức năng Đăng nhập/Đăng xuất:
  + Phân quyền người dùng: Admin, Giảng viên và Học viên.
* Quản lý khóa học:
  + Thêm, sửa, xóa và xem thông tin các khóa học.
  + Hiển thị danh sách khóa học với các tính năng tìm kiếm và lọc dữ liệu.
* Quản lý học viên:
  + Thêm mới, chỉnh sửa và xóa thông tin học viên.
  + Hiển thị danh sách học viên theo từng khóa học.
* Quản lý giảng viên:
  + Thêm và quản lý thông tin giảng viên phụ trách khóa học.
* Chức năng đăng ký khóa học:
  + Cho phép học viên xem danh sách và đăng ký các khóa học phù hợp.
* Báo cáo và thống kê:
  + Hiển thị số lượng khóa học, học viên và các biểu đồ thống kê trực quan.
* Giao diện người dùng thân thiện:
  + Giao diện trực quan, dễ sử dụng với các tính năng điều hướng rõ ràng.
* Xử lý lỗi và thông báo hệ thống:
  + Cung cấp các thông báo khi thực hiện thao tác và xử lý các tình huống lỗi (404, lỗi truy cập).

Dự án đã hoàn thành các yêu cầu đề ra ban đầu và đảm bảo hệ thống hoạt động ổn định trên môi trường local.

**2. Hướng phát triển**

Dựa trên kết quả hiện tại, hệ thống có thể được phát triển thêm các tính năng và cải tiến trong tương lai, cụ thể như sau:

* Triển khai hệ thống online:
  + Đưa ứng dụng lên môi trường cloud hoặc server thực tế để người dùng có thể truy cập từ bất kỳ đâu.
* Tích hợp hệ thống thanh toán:
  + Xây dựng chức năng thanh toán trực tuyến cho học viên khi đăng ký các khóa học.
* Cải thiện chức năng báo cáo và thống kê:
  + Thêm các biểu đồ nâng cao, báo cáo chi tiết theo khoảng thời gian và từng giảng viên.
  + Xuất dữ liệu báo cáo ra các định dạng như PDF hoặc Excel.
* Xây dựng tính năng gửi thông báo qua email:
  + Gửi thông báo tự động qua email khi học viên đăng ký khóa học hoặc khi có cập nhật từ giảng viên.
* Phát triển tính năng học trực tuyến:
  + Tích hợp các công cụ hỗ trợ học trực tuyến như video bài giảng, tài liệu PDF và tính năng tương tác giữa học viên và giảng viên.
* Bổ sung API cho Mobile App:
  + Phát triển các API cho phép xây dựng ứng dụng mobile để hỗ trợ người dùng trên các thiết bị di động.
* Tăng cường bảo mật hệ thống:
  + Tích hợp các công nghệ bảo mật như JWT hoặc OAuth2 để bảo vệ dữ liệu người dùng.
  + Cải thiện hệ thống quản lý mật khẩu với tính năng mã hóa và phục hồi mật khẩu an toàn.
* Tối ưu hiệu suất hệ thống:
  + Tối ưu cơ sở dữ liệu và cải thiện tốc độ tải trang khi hệ thống mở rộng với lượng dữ liệu lớn.

# **PHẦN IV. TÀI LIỆU THAM KHẢO**

1. https://docs.spring.io/spring-framework/reference/index.html
2. <https://docs.oracle.com/javase/8/docs/api/javax/swing/Spring.html>
3. https://www.thymeleaf.org/documentation.html