

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC TẬP DOANH NGHIỆP

**Xây dựng ứng dụng web quản lý khóa học ngắn hạn trường
SICT-HaUI**

GVHD: ThS. Vũ Thị Dương

Sinh viên: Nguyễn Điền Sơn

Lớp: 2021KTPM01 ***Khóa:*** 16

Hà Nội – Năm 2025

MỤC LỤC

LỜI CẢM ƠN.....	5
PHẦN I. MỞ ĐẦU	6
1. Tên đề tài.....	6
2. Lý do chọn đề tài	6
3. Mục đích của đề tài	6
4. Mục tiêu của đề tài	6
5. Bố cục đề tài	7
6. Phương pháp nghiên cứu.....	7
PHẦN II. NỘI DUNG.....	8
Chương 1. Cơ sở lý thuyết.....	8
1.1 Cơ sở lý luận.....	8
1.1.1 Khái niệm về Spring Framework	8
1.1.2 Mô hình MVC trong Spring.....	8
1.1.3 DI, IoC trong Spring	9
1.2 Cơ sở thực tiễn	9
1.2.1 Nhu cầu thực tế sử dụng Spring.....	9
1.2.2 Lợi ích của Spring.....	9
1.2.3 Ứng dụng của Spring trong các dự án thực tế.....	10
1.3 Tổng quan các công nghệ sử dụng	10
1.3.1 Hibernate - ORM cho Java.....	10
1.3.2 JDBC - Kết nối CSDL	11
1.3.3 Spring Security - Bảo mật ứng dụng.....	11
2.1 Giới thiệu	12
2.1.1 Mục tiêu hệ thống	12

2.1.2 Người dùng của hệ thống	12
2.2 Khảo sát sơ bộ hệ thống	13
2.2.1 Yêu cầu chức năng.....	13
2.2.2 Yêu cầu phi chức năng	14
2.3 Phân tích hệ thống	14
2.3.1 Kiến trúc hệ thống	14
2.3.2 Sơ đồ luồng dữ liệu (Data Flow Diagram).....	15
2.3.3 Mô hình thực thể - quan hệ.....	15
2.4 Phân tích use case	16
2.4.1 Use case “Đăng ký”	16
2.4.2 Use case “Đăng nhập”	18
2.4.3 Use case “Quản lý người dùng”	20
2.4.4 Use case “Quản lý khóa học”	23
2.4.5 Use case “Đăng ký khóa học”	26
2.4.6 Use case “Quản lý bài tập”	28
2.4.7 Use case “Nộp bài tập”	31
2.4.8 Use case “Chấm điểm”	32
2.5 Thiết kế giao diện	35
2.5.1 Giao diện Đăng ký.....	35
2.5.2 Giao diện Đăng nhập	35
2.5.3 Giao diện Thêm khóa học	36
2.5.4 Giao diện Chỉnh sửa khóa học.....	36
2.5.5 Giao diện Quản lý tài khoản	36
2.5.6 Giao diện Thêm bài tập	37

2.5.7 Giao diện Nhập điểm	37
Chương 3. Cài đặt và thử nghiệm	38
3.1 Cài đặt JDK	38
3.2 Thiết lập biến môi trường cho Java	39
3.3 Cài đặt Eclipse	41
3.4 Chạy dự án	43
PHẦN III. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	45
1. Nội dung đã thực hiện được	45
2. Hướng phát triển	46
PHẦN IV. TÀI LIỆU THAM KHẢO	48

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 Thực tập doanh nghiệp.

Đặc biệt, em xin tỏ lòng biết ơn sâu sắc tới cô ThS. Vũ Thị Dương đã 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 Thực tập doanh 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 quản lý khóa học ngắn hạn trường SICT-HaUI

2. Lý do chọn đề tài

Lựa chọn đề tài "Xây dựng ứng dụng web quản lý khóa học ngắn hạn trường SICT-HaUI" 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.

3. 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.

4. 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

5. 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 sở 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.

6. Phương pháp nghiên cứu

a. 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

b. 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

c. 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

1.1 Cơ sở lý luận

1.1.1 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.

1.1.2 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.

1.1.3 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 Giới thiệu

Ứng dụng Quản lý khóa học là một hệ thống web được xây dựng nhằm giúp các cơ sở giáo dục, trung tâm đào tạo dễ dàng quản lý thông tin liên quan đến khóa học, học viên và giảng viên. Hệ thống này giúp tự động hóa các quy trình quản lý, từ việc tổ chức các khóa học đến theo dõi tiến độ học tập của học viên và hiệu quả giảng dạy của giảng viên.

2.1.1 Mục tiêu hệ thống

Hệ thống hướng đến các mục tiêu sau:

- **Quản lý thông tin khóa học:** Thêm, chỉnh sửa, xóa thông tin về các khóa học.
- **Quản lý học viên:** Đăng ký học viên vào các khóa học, theo dõi tiến độ học tập và kết quả của học viên.
- **Quản lý giảng viên:** Quản lý thông tin giảng viên và phân công giảng viên cho các khóa học.
- **Theo dõi và đánh giá:** Cung cấp các thống kê về số lượng học viên, giảng viên và hiệu quả hoạt động của hệ thống.

2.1.2 Người dùng của hệ thống

Hệ thống được thiết kế với ba loại người dùng chính, mỗi loại có quyền hạn và chức năng khác nhau:

- **Admin (Quản trị viên):**
 - Quản lý toàn bộ hệ thống.
 - Thêm, chỉnh sửa, xóa thông tin về học viên, giảng viên và khóa học.
 - Phân công học viên vào các khóa học.
 - Theo dõi và xuất báo cáo thống kê.
- **Giảng viên (Teacher):**
 - Xem danh sách học viên trong các khóa học mà mình phụ trách.

- Tạo bài tập, chấm điểm và theo dõi tình trạng hoàn thành bài tập của học viên.
- Cập nhật điểm thi và theo dõi tiến độ học tập của học viên.
- **Học viên (Student):**
 - Xem danh sách các khóa học đã đăng ký.
 - Xem chi tiết khóa học, điểm số và bài tập.
 - Đánh dấu trạng thái hoàn thành bài tập.

2.2 Khảo sát sơ bộ hệ thống

2.2.1 Yêu cầu chức năng

Hệ thống cần đáp ứng các yêu cầu chức năng sau:

- **Quản lý thông tin khóa học:**
 - Thêm, sửa, xóa khóa học.
 - Hiện thị danh sách các khóa học đang hoạt động.
- **Quản lý học viên:**
 - Thêm, sửa, xóa thông tin học viên.
 - Đăng ký hoặc hủy đăng ký học viên vào các khóa học.
 - Theo dõi tiến độ và kết quả học tập của học viên.
- **Quản lý giảng viên:**
 - Thêm, sửa, xóa thông tin giảng viên.
 - Phân công giảng viên cho các khóa học.
 - Quản lý bài tập và điểm số.
- **Quản lý bài tập và điểm:**
 - Tạo, chỉnh sửa, xóa bài tập cho các khóa học.
 - Chấm điểm và cập nhật kết quả cho học viên.

2.2.2 Yêu cầu phi chức năng

- **Hiệu năng:** Hệ thống phải đáp ứng tốc độ truy xuất dữ liệu nhanh ngay cả khi có nhiều người dùng.
- **Tính bảo mật:** Bảo vệ thông tin người dùng và dữ liệu quan trọng khỏi các truy cập trái phép.
- **Khả năng mở rộng:** Dễ dàng nâng cấp và tích hợp thêm các chức năng mới khi cần thiết.
- **Tính thân thiện với người dùng:** Giao diện đơn giản, dễ sử dụng, phù hợp với cả học viên, giảng viên và quản trị viên.

2.3 Phân tích hệ thống

2.3.1 Kiến trúc hệ thống

Hệ thống được thiết kế theo kiến trúc 3 lớp (Three-Tier Architecture), bao gồm:

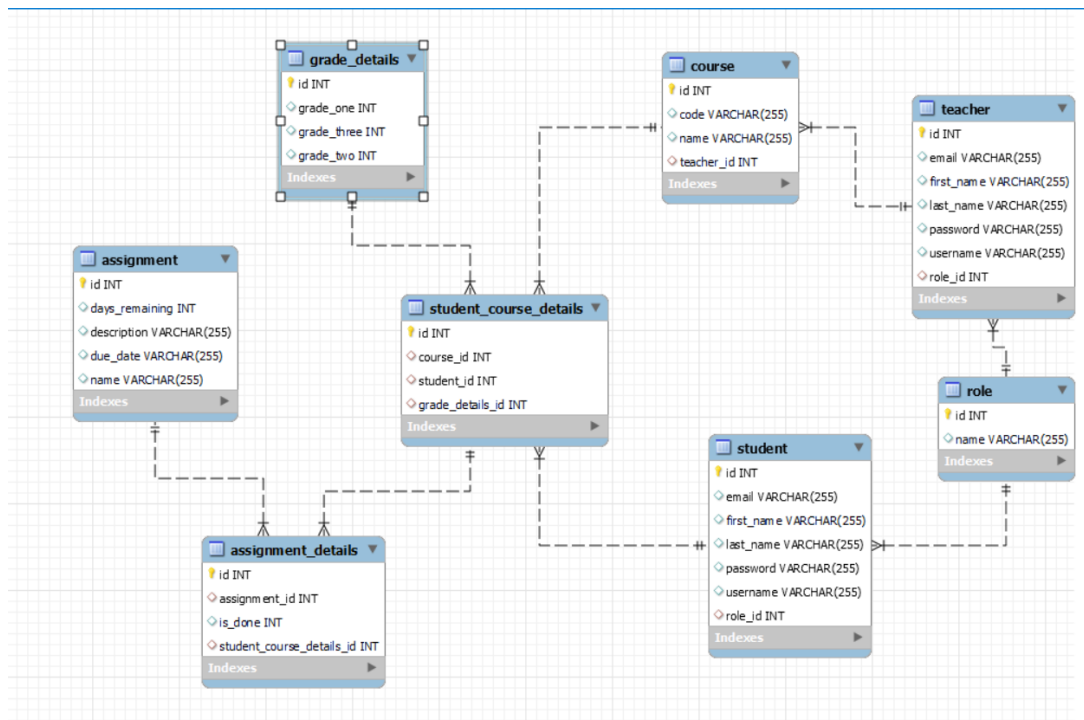
- **Lớp Presentation (Giao diện):**
 - Chịu trách nhiệm hiển thị giao diện người dùng và nhận dữ liệu đầu vào.
 - Được xây dựng bằng HTML, CSS, JavaScript và jQuery.
 - Kết nối với API RESTful để giao tiếp với lớp logic nghiệp vụ.
- **Lớp Business Logic (Xử lý nghiệp vụ):**
 - Xử lý các logic nghiệp vụ chính như quản lý khóa học, học viên, giảng viên và bài tập.
 - Được phát triển bằng Spring Boot và Spring MVC.
 - Đảm bảo tính toàn vẹn dữ liệu và thực hiện các quy tắc nghiệp vụ.
- **Lớp Data Access (Truy cập dữ liệu):**
 - Quản lý việc lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu.
 - Sử dụng Hibernate để kết nối và thao tác với cơ sở dữ liệu MySQL.

2.3.2 Sơ đồ luồng dữ liệu (Data Flow Diagram)

Luồng dữ liệu trong hệ thống được chia thành các bước chính sau:

1. Người dùng (Admin, Giảng viên, Học viên) gửi yêu cầu thông qua giao diện web.
2. Yêu cầu được chuyển đến lớp xử lý nghiệp vụ để thực hiện các logic liên quan.
3. Hệ thống truy vấn hoặc cập nhật dữ liệu trong cơ sở dữ liệu.
4. Kết quả trả về giao diện người dùng để hiển thị.

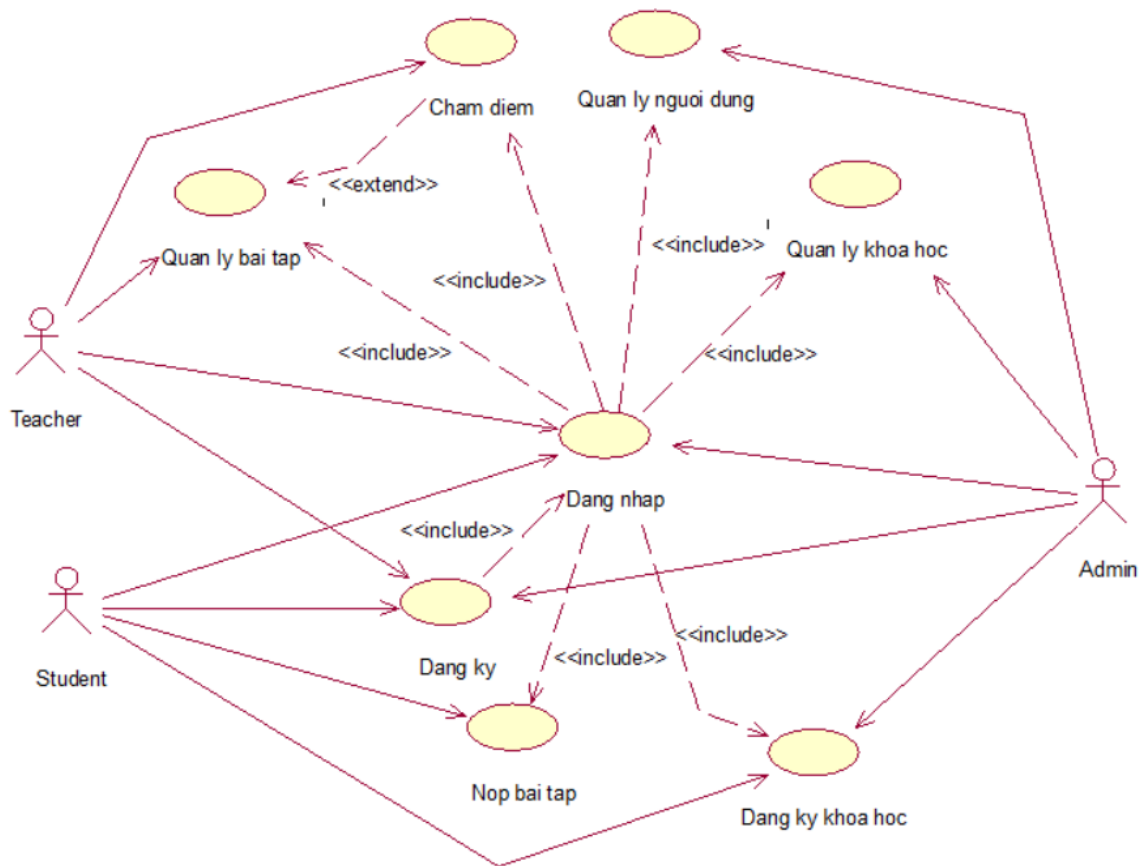
2.3.3 Mô hình thực thể - quan hệ



Hình 2.1 Biểu đồ lớp thực thể liên kết

2.4 Phân tích use case

Biểu đồ use case tổng quát



Hình 2.2 Sơ đồ use case tổng quát

2.4.1 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:

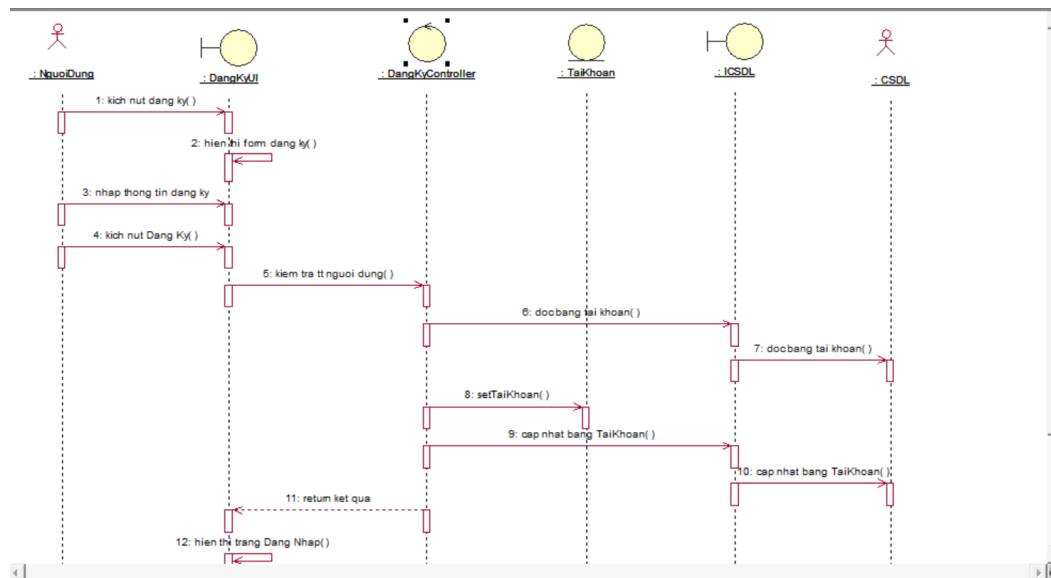
- Người dùng chưa có tài khoản trong hệ thống.

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 2.3 Biểu đồ trình tự use case “Đăng ký”

2.4.2 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:**

- Người dùng truy cập vào trang đăng nhập của hệ thống.
- 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:

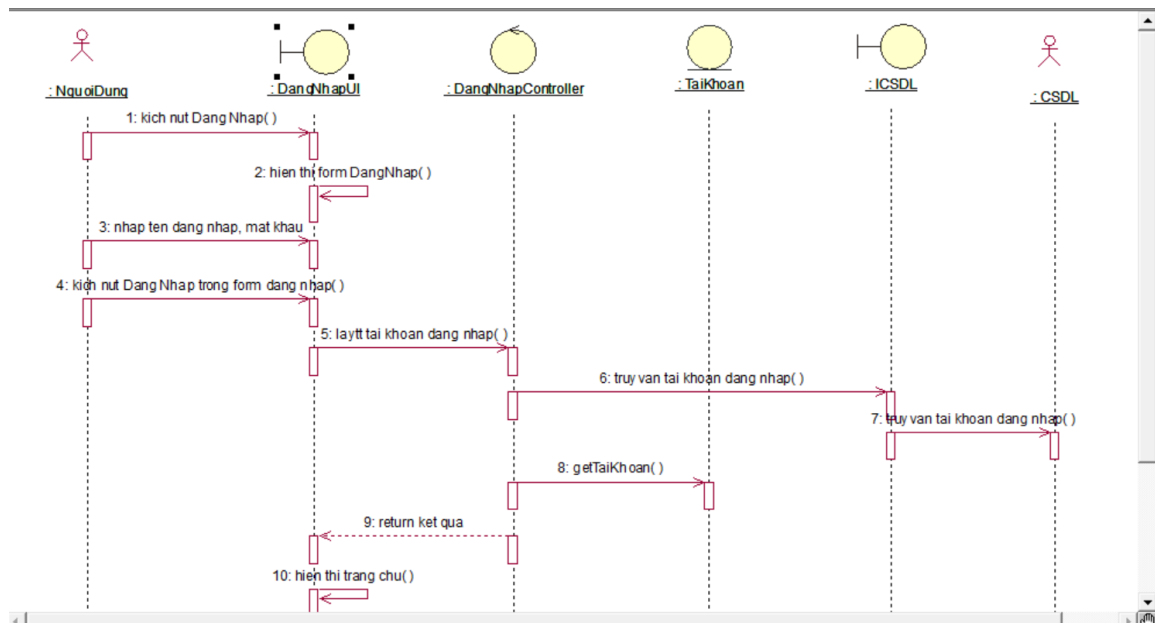
- Người dùng đã có tài khoản và đã đăng ký thành công trong hệ thống.

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 2.4 Biểu đồ trình tự use case “Đăng nhập”

2.4.3 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:

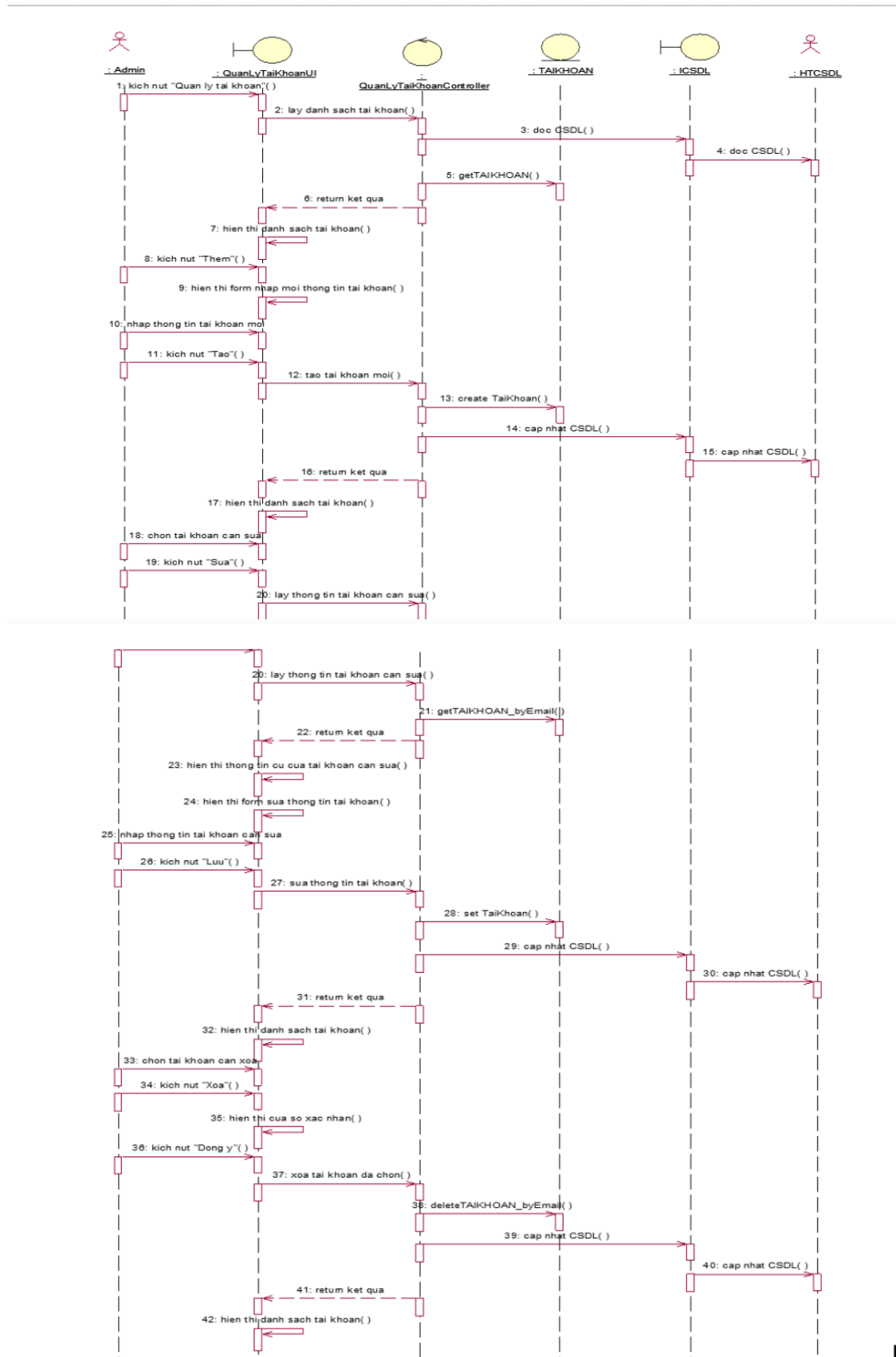
- Người quản lý phải đăng nhập vào hệ thống với vai trò admin hoặc người quản lý.
- Người quản lý có quyền truy cập vào trang quản lý người dùng.

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 2.5 Biểu đồ trình tự use case “Quản lý tài khoản”

2.4.4 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:

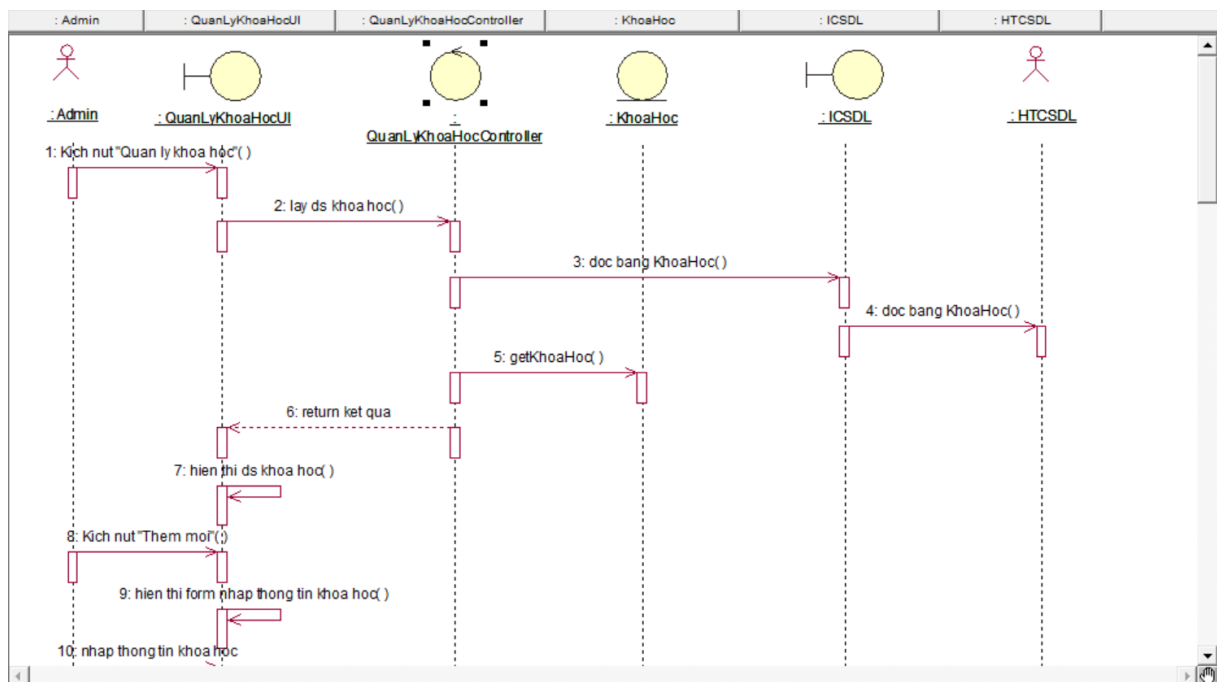
- Người quản lý phải đăng nhập vào hệ thống với vai trò admin hoặc giảng viên.
- Người quản lý có quyền truy cập vào trang quản lý khóa học.

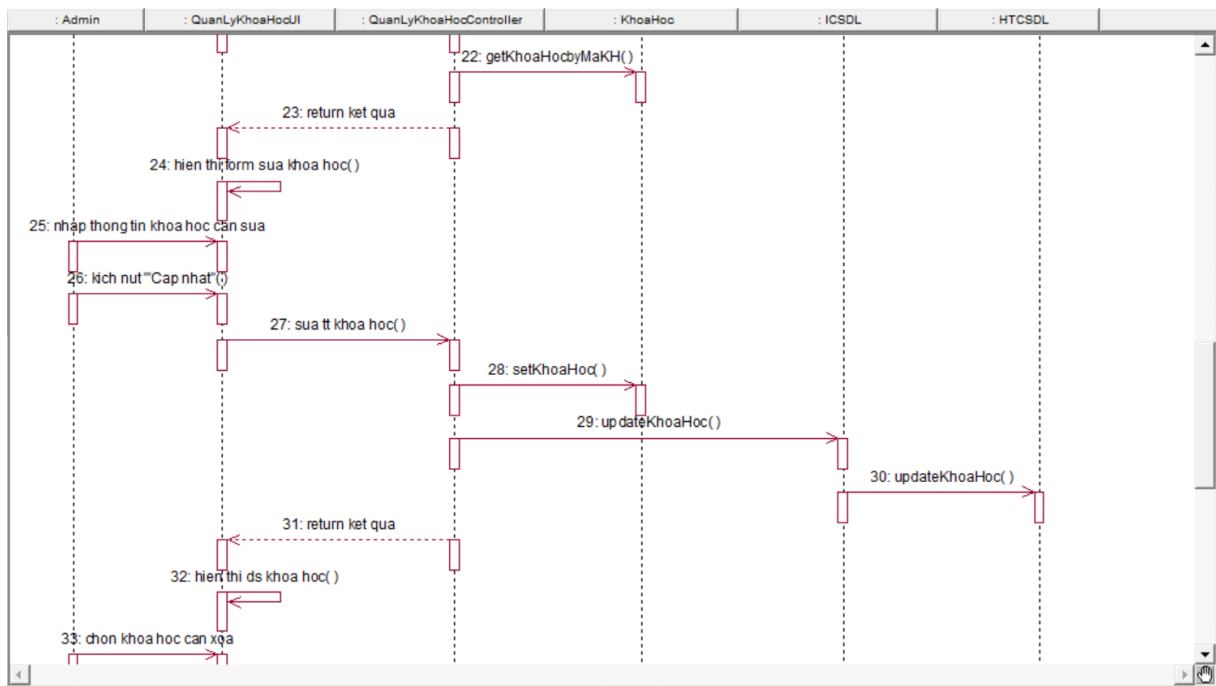
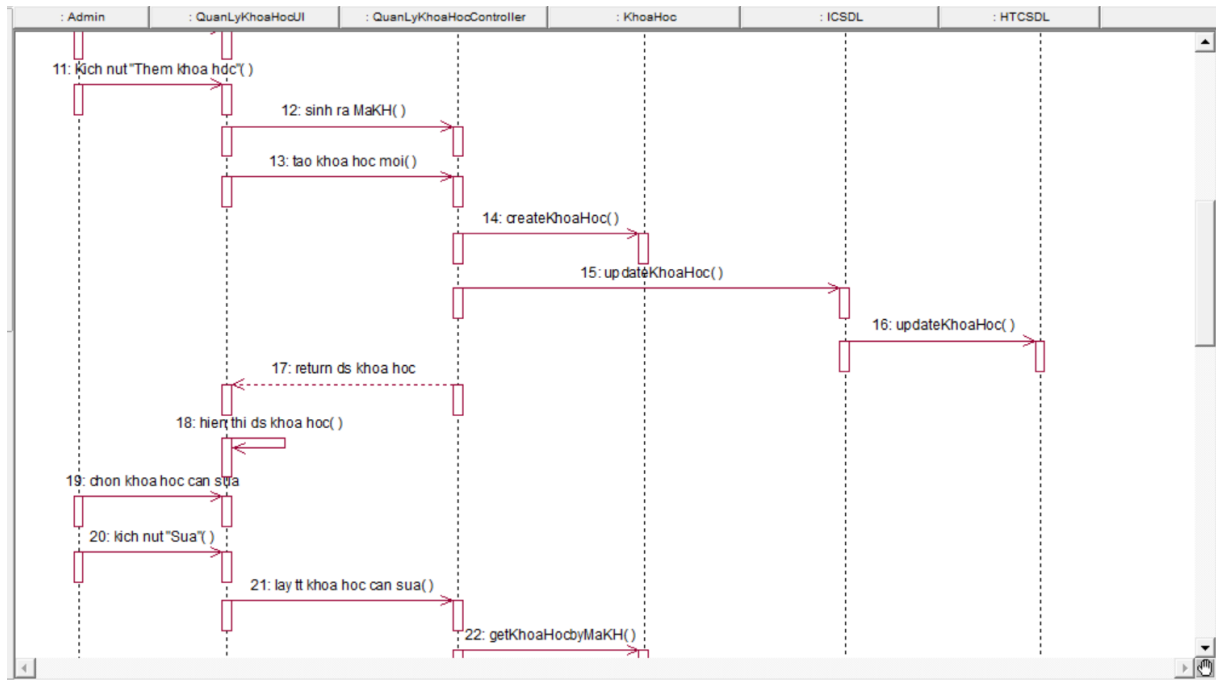
Hậu điều kiện:

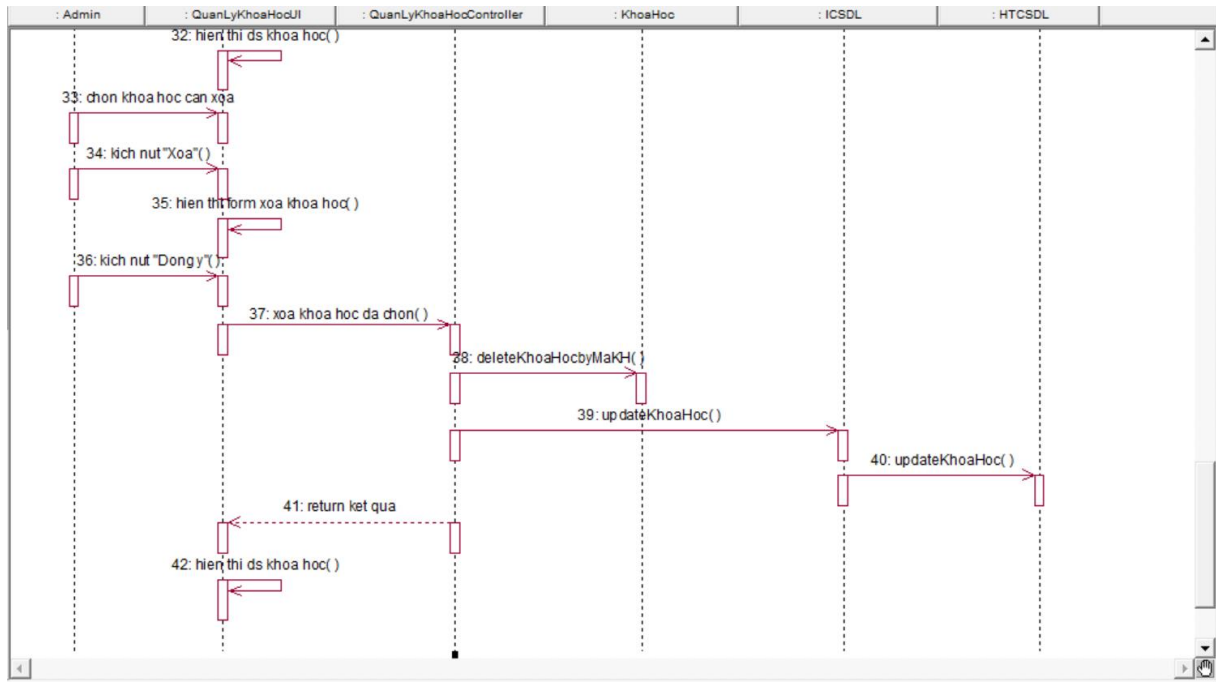
- Khóa học mới được thêm, sửa, hoặc xóa trong hệ thống.

Điểm mở rộng:

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







Hình 2.6 Biểu đồ trình tự use case “Quản lý khóa học”

2.4.5 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:

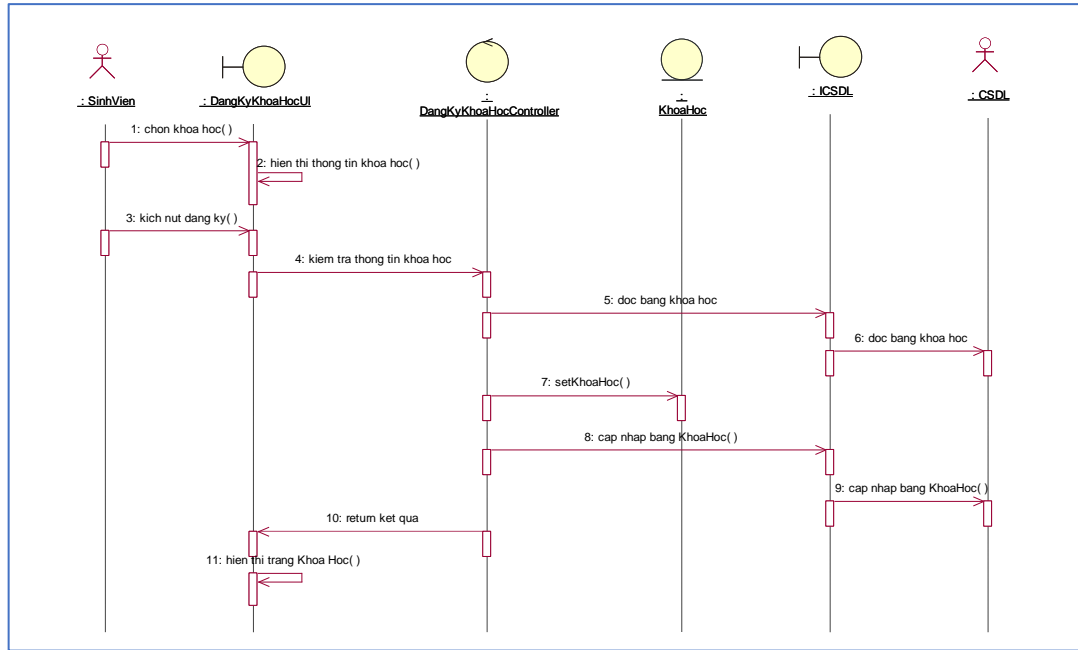
- Sinh viên phải đăng nhập vào hệ thống.
- Sinh viên chưa đăng ký khóa học này.

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 2.7 Biểu đồ trình tự use case “Đăng ký khóa học”

2.4.6 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:

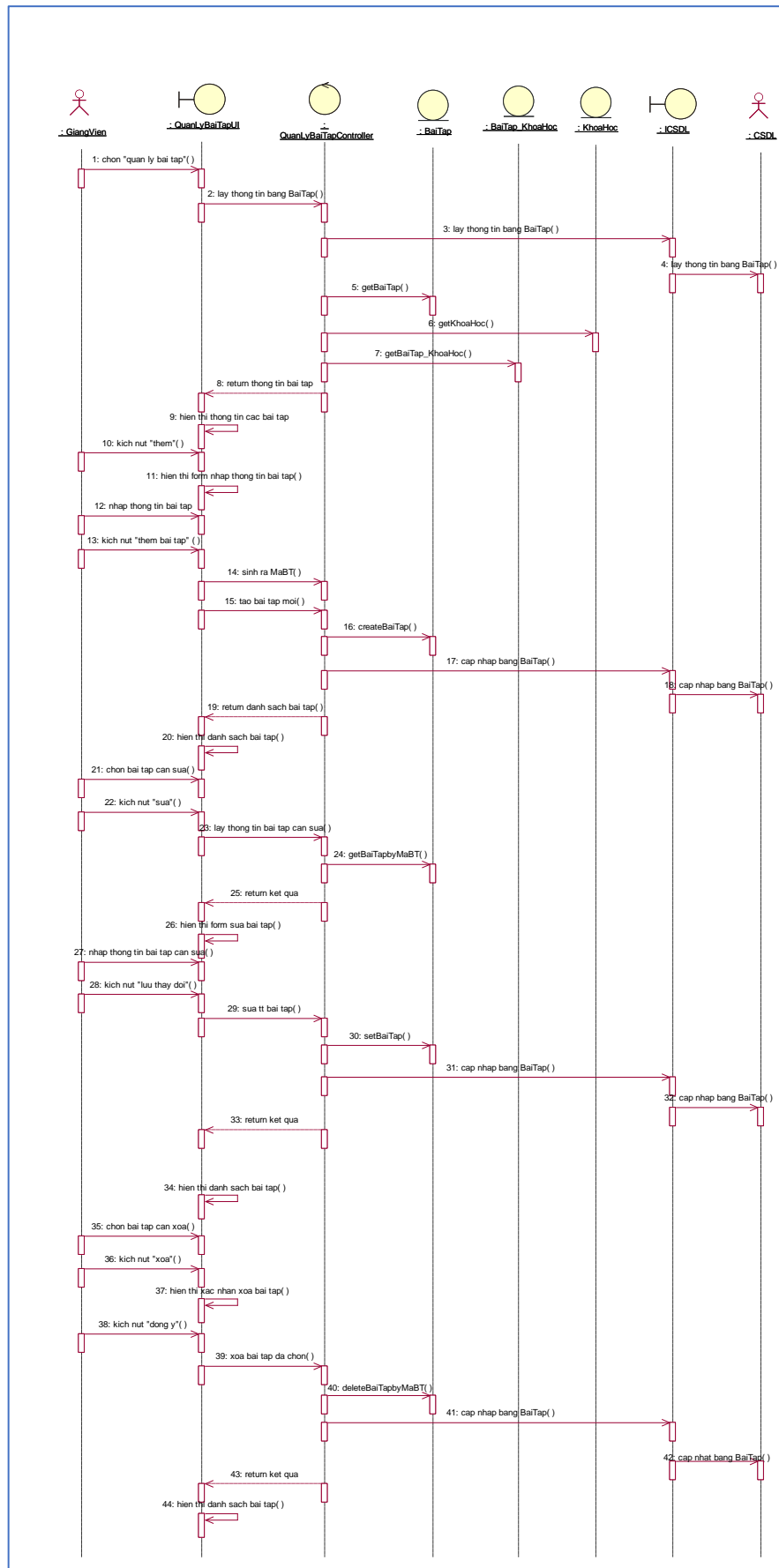
- Giảng viên phải đăng nhập vào hệ thống với vai trò giảng viên.
- Giảng viên có quyền quản lý khóa học và bài tập của khóa học đó.

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 2.8 Biểu đồ trình tự use case “Quản lý bài tập”

2.4.7 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:

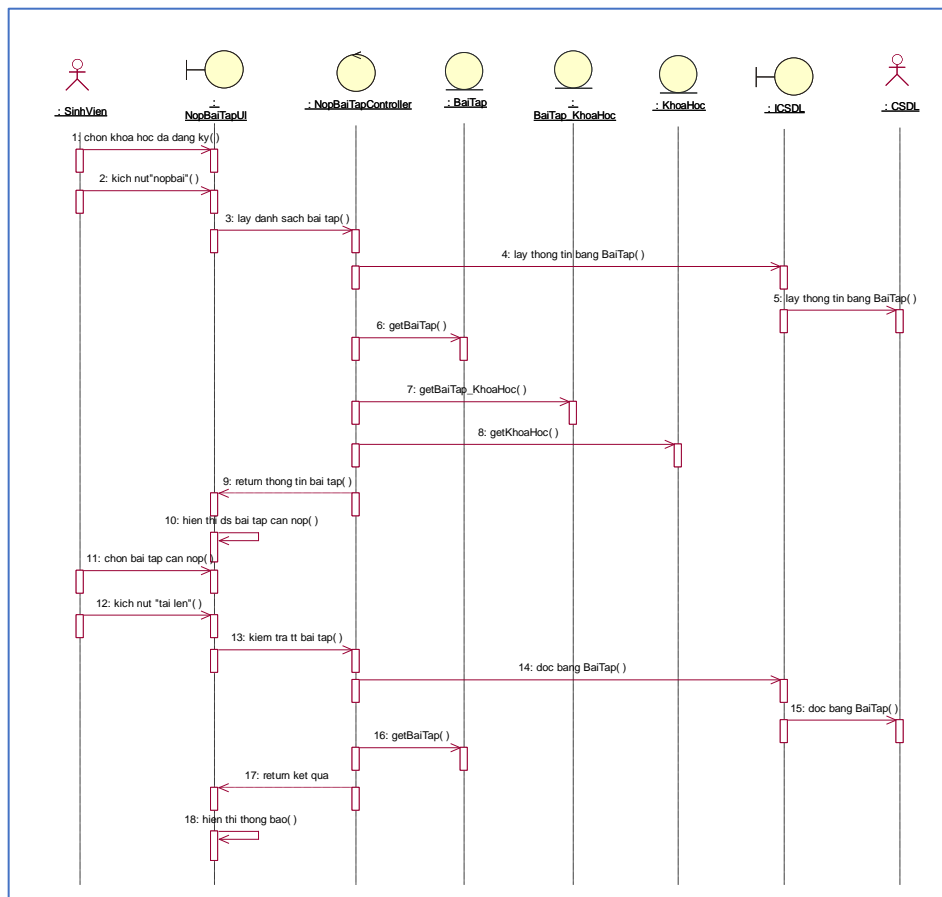
- Sinh viên đã đăng nhập vào hệ thống.
- Sinh viên đã hoàn thành bài tập và có tệp bài tập để nộp.

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 2.9 Biểu đồ trình tự use case “Nộp bài tập”

2.4.8 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:

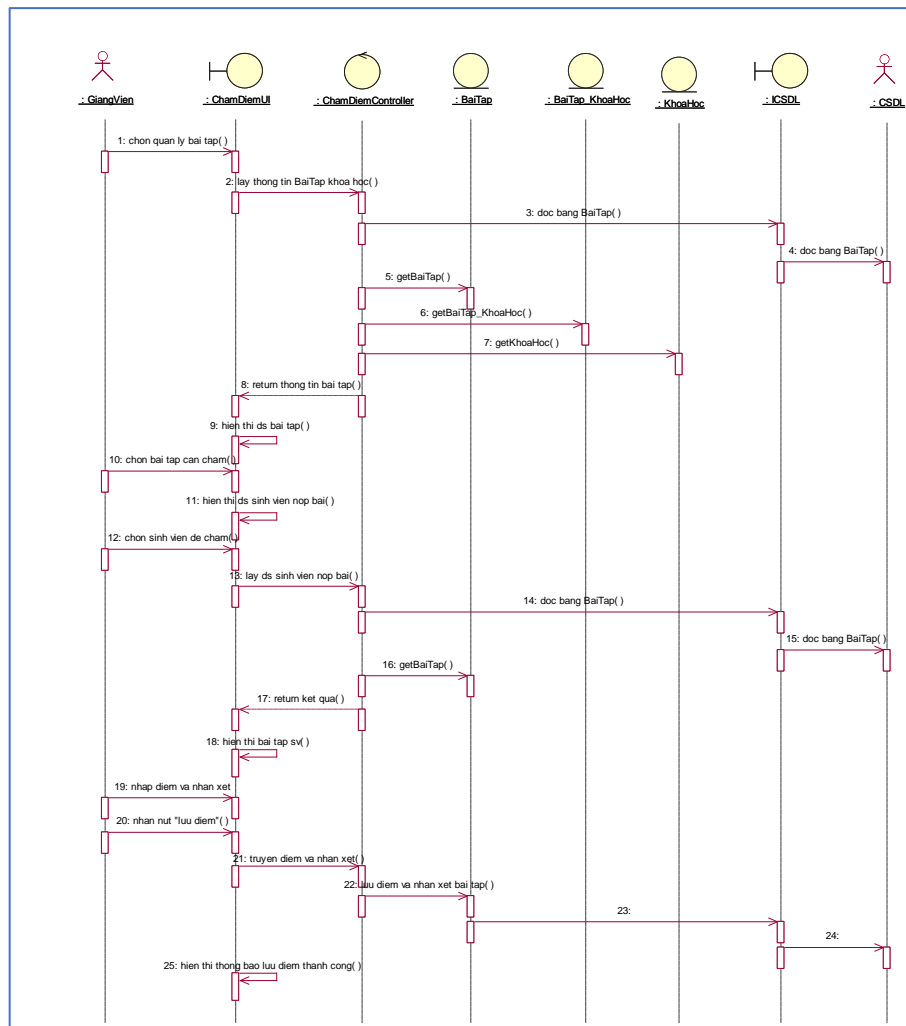
- Giảng viên đã đăng nhập vào hệ thống.
- Sinh viên đã nộp bài tập cho giảng viên chấm điểm.

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:

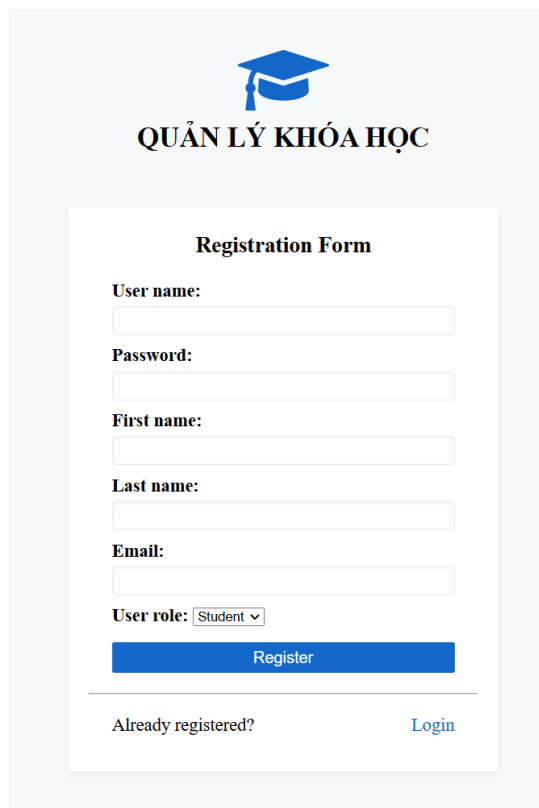
- Hệ thống có thể gửi thông báo cho sinh viên khi bài tập của họ đã được chấm điểm và có điểm mới.



Hình 2.10 Biểu đồ trình tự use case “Chấm điểm”

2.5 Thiết kế giao diện

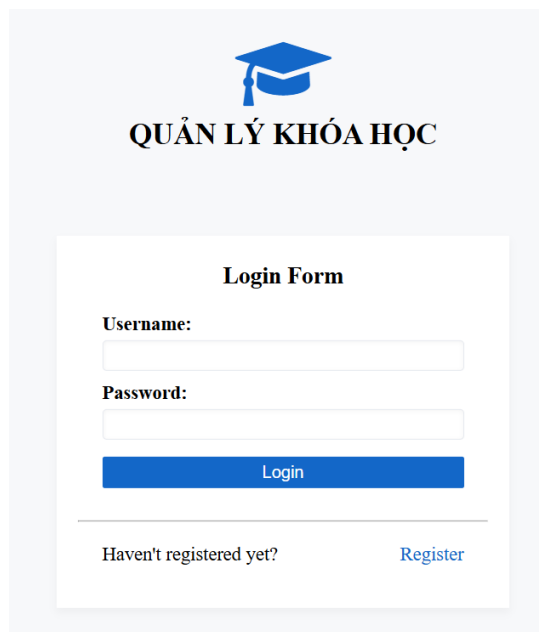
2.5.1 Giao diện Đăng ký



The registration form is titled "QUẢN LÝ KHÓA HỌC" with a graduation cap icon. It contains a "Registration Form" section with the following fields: "User name:", "Password:", "First name:", "Last name:", "Email:", and "User role:" (a dropdown menu with "Student" selected). A blue "Register" button is at the bottom of the form. Below the form, there is a link "Already registered?" followed by a blue "Login" link.

Hình 2.11 Giao diện Đăng ký

2.5.2 Giao diện Đăng nhập



The login form is titled "QUẢN LÝ KHÓA HỌC" with a graduation cap icon. It contains a "Login Form" section with the following fields: "Username:" and "Password:". A blue "Login" button is at the bottom of the form. Below the form, there is a link "Haven't registered yet?" followed by a blue "Register" link.

Hình 2.12 Giao diện Đăng nhập

2.5.3 Giao diện Thêm khóa học

Add New Course

Course code:

Course name:

Teacher:

Phong Nguyễn Anh ▾

[Add](#)

Hình 2.13 Giao diện Thêm khóa học

2.5.4 Giao diện Chỉnh sửa khóa học

Course List			
Code	Name	Teacher	Action
oop	Lập trình hướng đối tượng	Phong Nguyễn Anh	View Delete
dsa	Cấu trúc dữ liệu và giải thuật	Biên Chu Văn	View Delete
java	Lập trình Java	Phong Nguyễn Anh	View Delete
Create New Course			

Hình 2.14 Giao diện Chỉnh sửa khóa học

2.5.5 Giao diện Quản lý tài khoản

Id	First Name	Last Name	Email	Action
8	Son	Nguyễn Điền	email@gmail.com	Courses Delete
9	Lâm	Nguyễn Thế	email@gmail.com	Courses Delete
10	My	Nguyễn Trà	email@gmail.com	Courses Delete
11	Long	Nguyễn Văn	email@gmail.com	Courses Delete
12	Thủy	Phạm Thị	email@gmail.com	Courses Delete
13	Tuấn	Đoàn Hữu	email@gmail.com	Courses Delete
14	Hường	Vũ Thị	email@gmail.com	Courses Delete


Hình 2.15 Giao diện Quản lý tài khoản

2.5.6 Giao diện Thêm bài tập

Add New Assignment

Name:

Description:

Due Date:
 

[Save](#)

Hình 2.16 Giao diện Thêm bài tập

2.5.7 Giao diện Nhập điểm

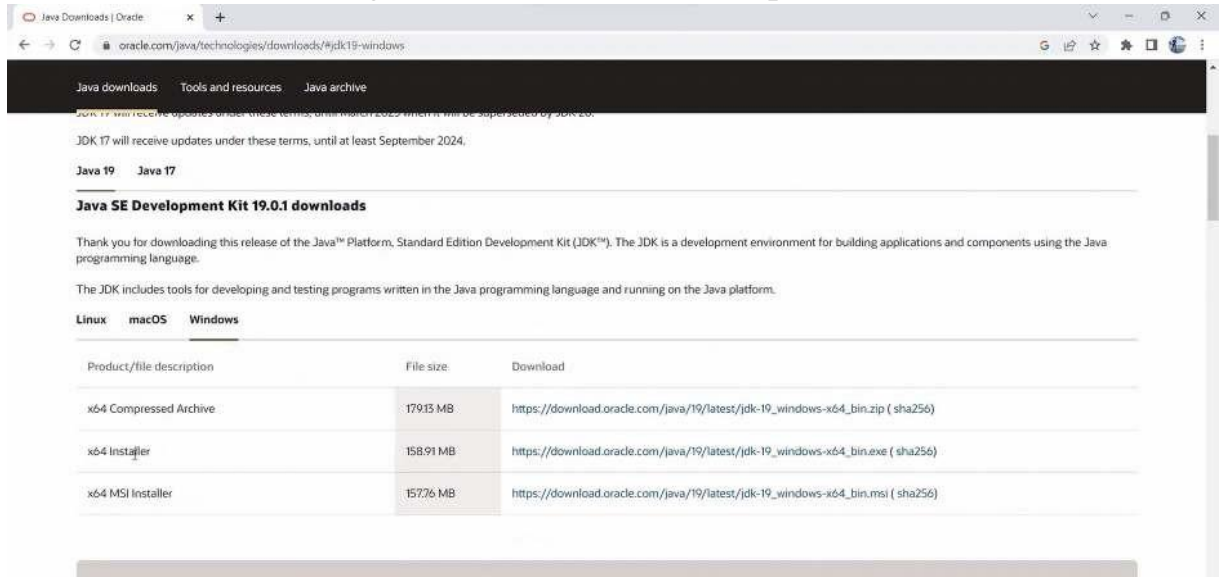
Lập trình hướng đối tượng Edit Grades				
Student	Grade 1	Grade 2	Grade 3	Action
Son Nguyễn Điền	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	Modify
Lâm Nguyễn Thế	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	Modify
My Nguyễn Trà	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	Modify
Long Nguyễn Văn	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	Modify
Thùy Phạm Thị	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	Modify
Tuấn Đoàn Hữu	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	Modify
Hường Vũ Thị	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	Modify

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/> để có thể download bản JDK chính thức từ Oracle. Ở giao diện chính sẽ hiển thị phiên bản Java mới nhất.

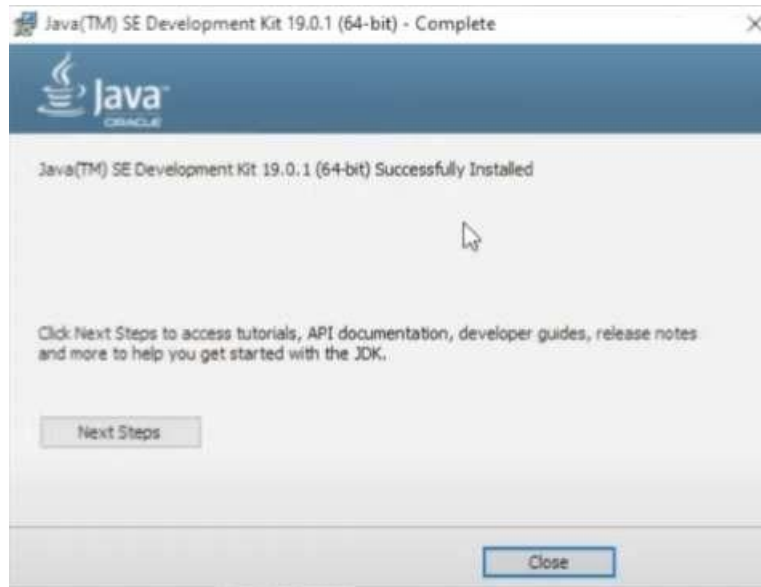


Bước 2: Bạn chọn hệ điều hành đang dùng sau đó bấm vào đường link bên dưới, máy tính sẽ hiển thị giao diện yêu cầu bạn chọn nơi lưu trữ.

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.

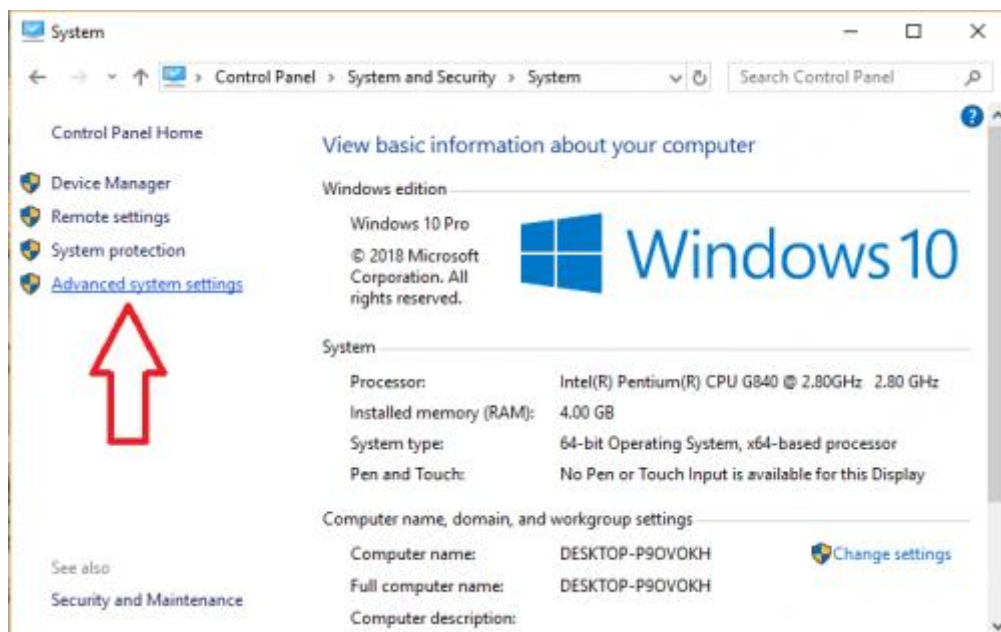


Bước 4: Sau khi quá trình cài đặt kết thúc, bấm Close để đóng cửa sổ lại.



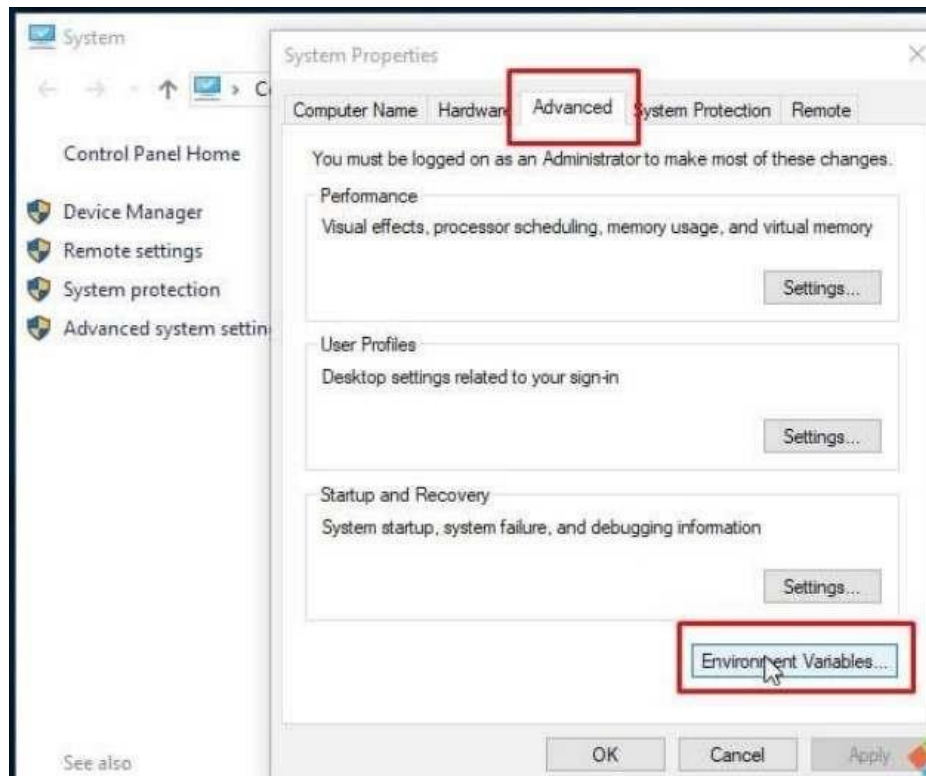
3.2 Thiết lập biến môi trường cho Java

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.

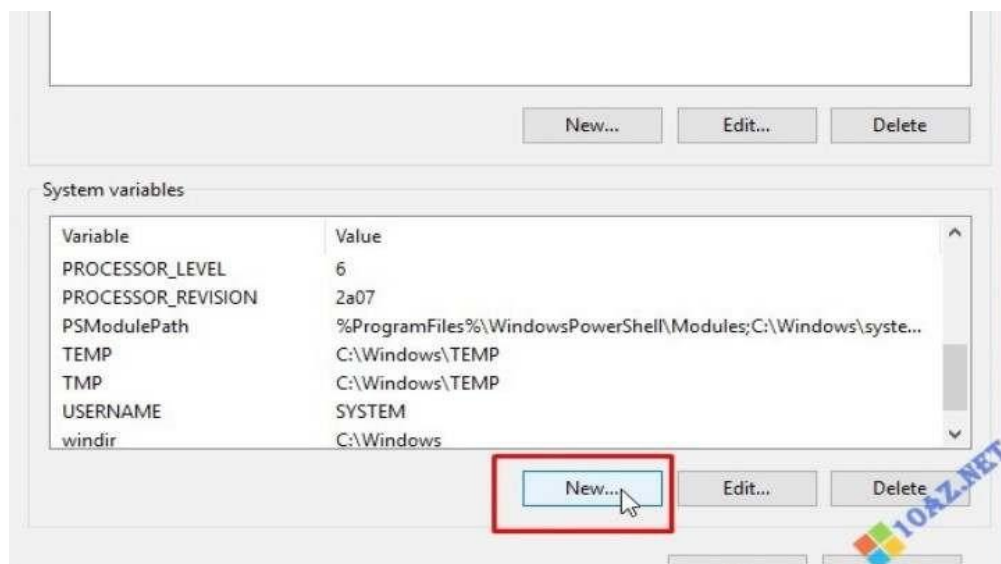


Chọn Advanced system settings

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



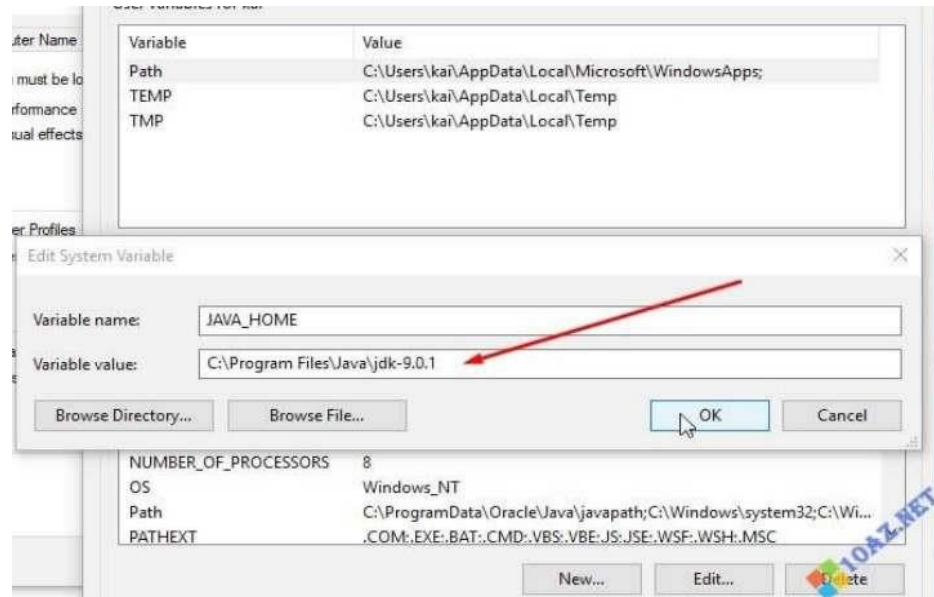
Bước 3: Trong cửa sổ Enviroment Variables, tìm tới mục System variables phía dưới và bấm New để tạo biến môi trường mới.



Tạo biến mới

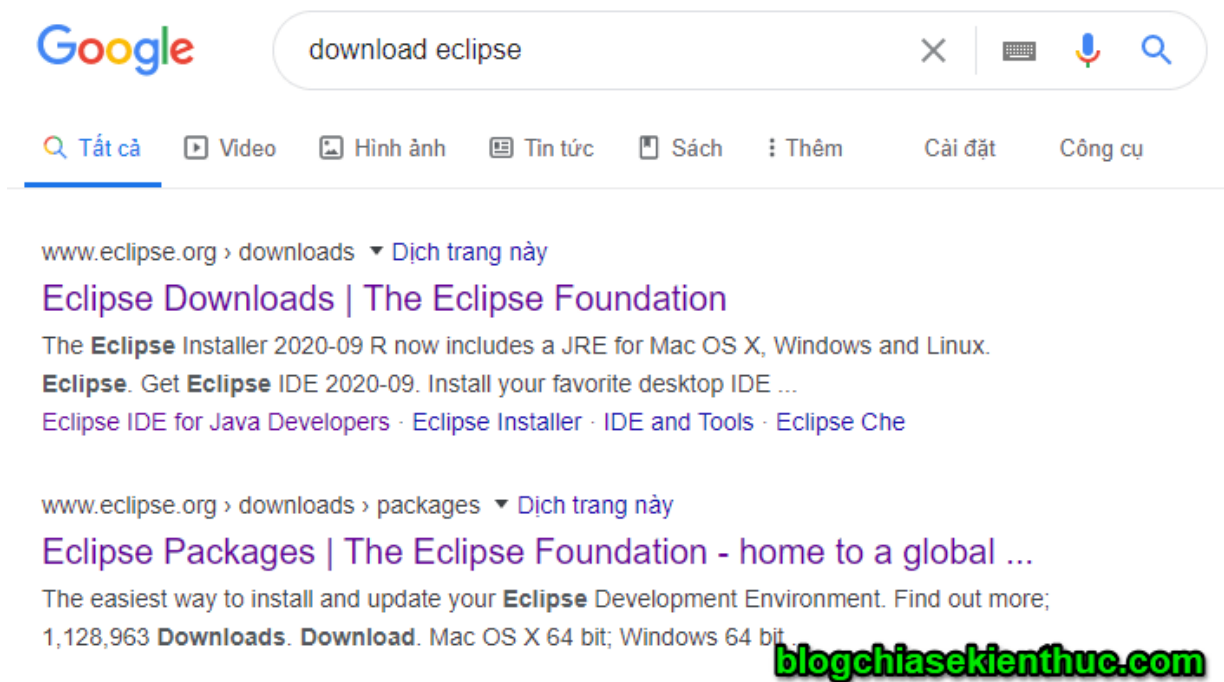
Bước 4: Khung cài đặt mới hiện ra, bạn nhập:

- JAVA_HOME vào ô Variable name.
- Truy cập vào thư mục cài đặt của Java, copy đường dẫn và dán vào ô Variable value (mục này tùy 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.

Home / Downloads / Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the [Eclipse Foundation Software User Agreement](#) unless specified.

 **Download**

Download from: China - Dalian Neusoft University of Information (大连东软信息学院) ([https](https://www.dl.neusoft.edu.cn/))

File: eclipse-inst-jre-win64.exe SHA-512

[>> Select Another Mirror](#)

OR Get It Faster from our Members


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).

Thank you for your download!

If the download doesn't start in a few seconds, please [click here](#) to start the download.

Power the **Eclipse Community** with your donation

If you value Eclipse technologies, please consider making a donation. Contributions from users like you help fund the operations of the Eclipse Foundation. All money donated to the Eclipse Foundation will be used to support the Eclipse Community.



\$5 \$10 \$35 \$50 \$100

USD

Donate

blogchiasekienthuc.com

3.4 Chạy dự án

Chạy script SQL:

```

20  --
21
22  • DROP TABLE IF EXISTS `assignment`;
23  • /*!40101 SET @saved_cs_client      = @@character_set_client */;
24  • /*!50503 SET character_set_client = utf8mb4 */;
25  • CREATE TABLE `assignment` (
26      `id` int NOT NULL AUTO_INCREMENT,
27      `days_remaining` int DEFAULT NULL,
28      `description` varchar(255) DEFAULT NULL,
29      `due_date` varchar(255) DEFAULT NULL,
30      `name` varchar(255) DEFAULT NULL,
31      PRIMARY KEY (`id`)
32  ) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
33  • /*!40101 SET character_set_client = @saved_cs_client */;
34
35  --
36  -- Dumping data for table `assignment`
37  --
38
39  • LOCK TABLES `assignment` WRITE;
40  • /*!40000 ALTER TABLE `assignment` DISABLE KEYS */;
41  • INSERT INTO `assignment` VALUES (1,10,'test','2024-12-14','alo'),(2,4,'Kiểm tra thường xuyên 1','2024-12-19','KTTX01');
42  • /*!40000 ALTER TABLE `assignment` ENABLE KEYS */;
43  • UNLOCK TABLES;
44
45  --
46  -- Table structure for table `assignment_details`
47  --

```

Mở project bằng Eclipse:

```

> quanlykhoahoc [quanlykhoahoc master]
  > src/main/java
    > com.ndson03.quanlykhoahoc
      > QuanLyKhoaHocApplication.java
      > com.ndson03.quanlykhoahoc.config
      > com.ndson03.quanlykhoahoc.controller
      > com.ndson03.quanlykhoahoc.dao
      > com.ndson03.quanlykhoahoc.entity
      > com.ndson03.quanlykhoahoc.service
      > com.ndson03.quanlykhoahoc.user
    > src/main/resources
      > static
      > templates
        > application.properties
        > index.html
    > src/test/java
    > JRE System Library [JavaSE-11]
    > Maven Dependencies
    > > bin
    > src
    > target
      > mvnw
      > mvnw.cmd
      > pom.xml
      > README.md

```

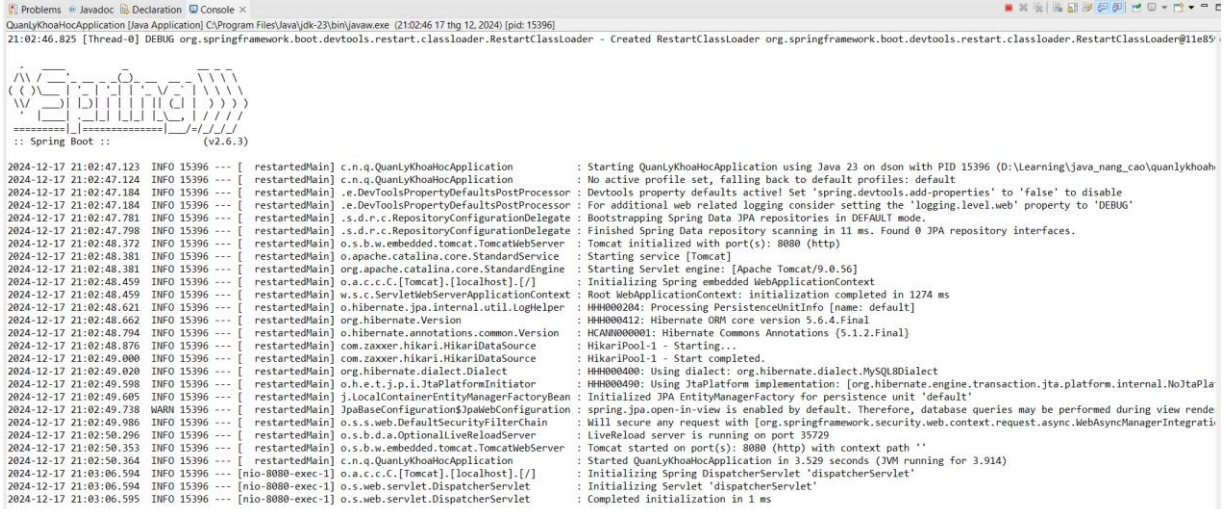
Cấu hình lại application.properties:

```

application.properties
1 # JDBC properties
2 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
3 spring.datasource.url=${DB_URL:jdbc:mysql://localhost:3306/spring_security_custom_user_demo?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true}
4 spring.datasource.username=${DB_USER:springstudent}
5 spring.datasource.password=${DB_PASS:springstudent}
6
7 # Hibernate properties
8 spring.jpa.hibernate.ddl-auto=update
9 spring.jpa.show-sql=true
10
11 # Server port (use default 8080 locally)
12 server.port=${PORT:8080}
13

```

Chạy dự án:



```

Problems Javadoc Declaration Console
QuanLyKhoaHocApplication [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (21:02:46 17 thg 12 2024) [pid: 15396]
21:02:46.825 [Thread-0] DEBUG org.springframework.boot.devtools.restart.classloader.RestartClassLoader - Created RestartClassLoader org.springframework.boot.devtools.restart.classloader.RestartClassLoader@11e85...

:: Spring Boot ::
(v2.6.3)

2024-12-17 21:02:47.123 INFO 15396 --- [ restartedMain] c.n.q.QuanLyKhoaHocApplication : Starting QuanLyKhoaHocApplication using Java 23 on dson with PID 15396 (D:\Learning\java_nang_cao\quanlykhoa
2024-12-17 21:02:47.124 INFO 15396 --- [ restartedMain] c.n.q.QuanLyKhoaHocApplication : No active profile set, falling back to default profiles: default
2024-12-17 21:02:47.184 INFO 15396 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false' to disable
2024-12-17 21:02:47.184 INFO 15396 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property to 'DEBUG'
2024-12-17 21:02:47.781 INFO 15396 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2024-12-17 21:02:47.798 INFO 15396 --- [ restartedMain] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 11 ms. Found 0 JPA repository interfaces.
2024-12-17 21:02:48.372 INFO 15396 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2024-12-17 21:02:48.381 INFO 15396 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-12-17 21:02:48.381 INFO 15396 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.56]
2024-12-17 21:02:48.459 INFO 15396 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2024-12-17 21:02:48.459 INFO 15396 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1274 ms
2024-12-17 21:02:48.621 INFO 15396 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2024-12-17 21:02:48.662 INFO 15396 --- [ restartedMain] org.hibernate.Version : HHH0000412: Hibernate ORM core version 5.6.4.Final
2024-12-17 21:02:48.794 INFO 15396 --- [ restartedMain] o.hibernate.annotations.common.Version : HCA00000001: Hibernate Commons Annotations {5.1.2.Final}
2024-12-17 21:02:48.876 INFO 15396 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2024-12-17 21:02:49.000 INFO 15396 --- [ restartedMain] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2024-12-17 21:02:49.020 INFO 15396 --- [ restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
2024-12-17 21:02:49.598 INFO 15396 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jts.platform.internal.NoJtaPla
2024-12-17 21:02:49.605 INFO 15396 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2024-12-17 21:02:49.738 WARN 15396 --- [ restartedMain] jpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rende
2024-12-17 21:02:49.986 INFO 15396 --- [ restartedMain] o.s.s.web.DefaultSecurityFilterChain : Will secure any request with [org.springframework.security.web.context.request.async.WebAsyncManagerIntegrati
2024-12-17 21:02:50.296 INFO 15396 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2024-12-17 21:02:50.353 INFO 15396 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2024-12-17 21:02:50.364 INFO 15396 --- [ restartedMain] c.n.q.QuanLyKhoaHocApplication : Started QuanLyKhoaHocApplication in 3.529 seconds (JVM running for 3.914)
2024-12-17 21:03:06.594 INFO 15396 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2024-12-17 21:03:06.595 INFO 15396 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2024-12-17 21:03:06.595 INFO 15396 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 1 ms

```

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/Swing.html>
3. <https://www.thymeleaf.org/documentation.html>