

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN 1



**ĐỒ ÁN TỐT NGHIỆP**

**Đề tài**

Xây dựng website học và thi lập trình trực tuyến

**Giảng viên hướng dẫn: Bùi Văn Kiên**

**Sinh viên thực hiện:**

Kiều Văn Hiếu – B21DCCN052

Nguyễn Thiên Hóa – B21DCCN376

*Hà Nội – 2025*

## LỜI CẢM ƠN

Chúng em xin bày tỏ lòng biết ơn sâu sắc tới các thầy giáo, cô giáo tại Học viện Công nghệ Bưu chính Viễn thông, đặc biệt là các thầy cô trong Khoa Công nghệ thông tin 1, đã tận tình giảng dạy, truyền đạt kiến thức và kinh nghiệm quý báu cho chúng em trong suốt bốn năm học vừa qua. Những kiến thức và sự định hướng của thầy cô không chỉ giúp chúng em hoàn thành tốt chương trình học mà còn là nền tảng vững chắc cho con đường học tập và làm việc sau này.

Chúng em xin gửi lời cảm ơn chân thành tới thầy Bùi Văn Kiên đã trực tiếp hướng dẫn, góp ý và hỗ trợ chúng em trong suốt quá trình thực hiện đề tài. Sự tận tâm và những ý kiến đóng góp quý báu của thầy đã giúp chúng em hoàn thiện đồ án một cách tốt nhất.

Cuối cùng, chúng em xin cảm ơn gia đình, bạn bè đã luôn động viên, giúp đỡ và tạo điều kiện thuận lợi để chúng em có thể hoàn thành đồ án này. Do thời gian và kiến thức còn hạn chế, đồ án không tránh khỏi những thiếu sót, chúng em rất mong nhận được sự góp ý của thầy cô để có thể hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

# MỤC LỤC

DANH MỤC BẢNG.....	5
DANH MỤC HÌNH ẢNH .....	7
DANH MỤC THUẬT NGỮ VÀ CÁC TỪ VIẾT TẮT .....	9
LỜI MỞ ĐẦU.....	10
CHƯƠNG 1: GIỚI THIỆU ĐỒ ÁN VÀ CÁC CÔNG NGHỆ SỬ DỤNG.....	11
1.1. Giới thiệu đề tài .....	11
1.2. Mô tả đề tài.....	11
1.2.1. Tổng quan hệ thống .....	11
1.2.2. Đối tượng sử dụng và phạm vi ứng dụng .....	11
1.2.3. Các chức năng chính của hệ thống .....	12
1.3. Công cụ và công nghệ sử dụng .....	13
1.3.1. Ngôn ngữ và nền tảng phát triển .....	13
1.3.2. Framework backend – Django.....	14
1.3.3. Framework frontend – React .....	15
1.3.4. Cơ sở dữ liệu – MySQL .....	15
1.3.5. Công nghệ hỗ trợ chấm bài lập trình DOMjudge .....	16
1.3.6. Công cụ phát triển và triển khai.....	16
1.4. Tổng kết.....	17
CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....	18
2.1. Phân tích hệ thống .....	18
2.1.1. Xác định tác nhân và chức năng .....	18
2.1.2. Biểu đồ Usecase Tổng quát .....	19
2.1.3. Kịch bản các chức năng chính của hệ thống .....	20
2.2. Thiết kế hệ thống .....	91
2.2.1. Biểu đồ lớp .....	91
2.2.2. Thiết kế cơ sở dữ liệu .....	93
2.2.3. Biểu đồ tuần tự cho từng chức năng .....	93
CHƯƠNG 3: HỆ THỐNG GỌI Ý SỬ DỤNG MÔ HÌNH .....	102
3.1. Tổng quan về bài toán gợi ý trong môi trường học lập trình .....	102

3.1.1. Đặt vấn đề.....	102
3.1.2. Phân tích đặc thù dữ liệu .....	102
3.1.3. Kết luận.....	102
3.2. Cơ sở lý thuyết .....	103
3.2.1. Lọc theo nội dung (Content-based Filtering) .....	103
3.2.2. Lọc cộng tác dựa trên mô hình (Model-based Collaborative Filtering) .....	104
3.2.3. Mô hình gợi ý lai ghép (Hybrid Recommendation System).....	106
3.2.4. Lý thuyết đánh giá độ khó động .....	107
3.3. Thiết kế chi tiết và thực hiện cài đặt .....	108
3.3.1. Dữ liệu và tiền xử lý .....	108
3.3.2. Cài đặt thuật toán và chiến lược lai ghép .....	109
3.3.3. Kết quả thực nghiệm.....	110
<b>CHƯƠNG 4: TRIỂN KHAI HỆ THỐNG .....</b>	<b>112</b>
4.1. Backend .....	112
4.1.1. Cấu trúc thư mục .....	112
4.1.2. Mô hình kiến trúc backend django .....	113
4.1.3. Tích hợp với hệ thống bên ngoài .....	113
4.1.4. Triển khai.....	114
4.2. Frontend .....	114
4.2.1. Tổng quan .....	114
4.2.2. Giao diện chức năng chính .....	116
<b>KẾT LUẬN.....</b>	<b>133</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>133</b>

## **DANH MỤC BẢNG**

Bảng 2.1 Kịch bản chuẩn chức năng đăng ký tài khoản.....	22
Bảng 2.2 Kịch bản chuẩn chức năng đăng nhập .....	23
Bảng 2.3 Kịch bản chuẩn chức năng cập nhật hồ sơ cá nhân .....	25
Bảng 2.4 Kịch bản chuẩn chức năng xem danh sách khóa học .....	26
Bảng 2.5 Kịch bản chuẩn chức năng tìm kiếm khóa học.....	27
Bảng 2.6 Kịch bản chuẩn chức năng lọc khóa học .....	28
Bảng 2.7 Kịch bản chuẩn chức năng xem chi tiết khóa học, bài học .....	29
Bảng 2.8 Kịch bản chuẩn chức năng đăng ký khóa học .....	31
Bảng 2.9 Kịch bản chuẩn chức năng làm bài quiz trắc nghiệm.....	33
Bảng 2.10 Kịch bản chuẩn chức năng nộp bài luyện tập lập trình .....	34
Bảng 2.11 Kịch bản chuẩn chức năng tham gia contest .....	35
Bảng 2.12 Kịch bản chuẩn chức năng nộp bài trong contest.....	36
Bảng 2.13 Kịch bản chuẩn chức năng xem bảng xếp hạng contest.....	38
Bảng 2.14 Kịch bản chuẩn chức năng xem bảng xếp hạng global .....	40
Bảng 2.15 Kịch bản chuẩn chức năng xem chi tiết người dùng .....	42
Bảng 2.16 Kịch bản chuẩn chức năng thêm người dùng mới.....	43
Bảng 2.17 Kịch bản chuẩn chức năng chỉnh sửa người dùng.....	44
Bảng 2.18 Kịch bản chuẩn chức năng xóa người dùng .....	45
Bảng 2.19 Kịch bản chuẩn chức năng tạo khóa học mới.....	46
Bảng 2.20 Kịch bản chuẩn chức năng chỉnh sửa khóa học.....	47
Bảng 2.21 Kịch bản chuẩn chức năng xem chi tiết khóa học .....	49
Bảng 2.22 Kịch bản chuẩn chức năng xóa khóa học .....	50
Bảng 2.23 Kịch bản chuẩn chức năng xem chi tiết bài học .....	51
Bảng 2.24 Kịch bản chuẩn chức năng thêm bài học .....	53
Bảng 2.25 Kịch bản chuẩn chức năng sửa bài bài học .....	54
Bảng 2.26 Kịch bản chuẩn chức năng xóa bài bài học .....	55
Bảng 2.27 Kịch bản chuẩn chức năng xem bài bài quiz.....	57
Bảng 2.28 Kịch bản chuẩn chức năng tạo bài bài quiz.....	58
Bảng 2.29 Kịch bản chuẩn chức năng sửa bài bài quiz .....	60
Bảng 2.30 Kịch bản chuẩn chức năng xóa bài bài quiz .....	60
Bảng 2.31 Kịch bản chuẩn chức năng bài tập lập trình (problem) .....	62
Bảng 2.32 Kịch bản chuẩn chức năng tạo bài lập trình .....	64
Bảng 2.33 Kịch bản chuẩn chức năng chỉnh sửa bài lập trình.....	65
Bảng 2.34 Kịch bản chuẩn chức năng xóa bài lập trình .....	66
Bảng 2.35 Kịch bản chuẩn chức năng xem chi tiết contest .....	67
Bảng 2.36 Kịch bản chuẩn chức năng tạo contest .....	69
Bảng 2.37 Kịch bản chuẩn chức năng chỉnh sửa contest .....	70
Bảng 2.38 Kịch bản chuẩn chức năng xoá contest .....	71
Bảng 2.39 Kịch bản chuẩn chức năng thêm problem vào contest .....	73
Bảng 2.40 Kịch bản chuẩn chức năng xem thống kê học viên .....	74

Bảng 2.41 Kịch bản chuẩn chức năng xem thông kê khóa học .....	75
Bảng 2.42 Kịch bản chuẩn chức năng xem báo cáo doanh thu .....	77
Bảng 2.43 UC chi tiết chức năng xem thông kê contest.....	79
Bảng 2.44 Kịch bản chuẩn chức năng thêm vai trò .....	81
Bảng 2.45 Kịch bản chuẩn chức năng sửa vai trò.....	83
Bảng 2.46 Kịch bản chuẩn chức năng xóa vai trò .....	84
Bảng 2.47 Kịch bản chuẩn chức năng xem danh sách quyền.....	85
Bảng 2.48 Kịch bản chuẩn chức năng xem danh sách loại nhóm quyền.....	86
Bảng 2.49 Kịch bản chuẩn chức năng xem danh sách ngôn ngữ.....	87
Bảng 2.50 Kịch bản chuẩn chức năng xem danh sách thẻ.....	88
Bảng 2.51 Kịch bản chuẩn chức năng thêm thẻ.....	89
Bảng 2.52 Kịch bản chuẩn chức năng sửa thẻ .....	90
Bảng 2.53 Kịch bản chuẩn chức năng xóa thẻ.....	91

## DANH MỤC HÌNH ẢNH

Hình 2.1 Biểu đồ Usecase tổng quan toàn hệ thống .....	20
Hình 2.2 Biểu đồ Usecase đăng ký tài khoản .....	21
Hình 2.3 Biểu đồ UC đăng nhập .....	22
Hình 2.4 UC chi tiết chức năng cập nhật hồ sơ cá nhân .....	23
Hình 2.5 Biểu đồ UC xem danh sách khóa học .....	25
Hình 2.6 Biểu đồ UC tìm kiếm khóa học .....	26
Hình 2.7 Biểu đồ UC lọc khóa học .....	27
Hình 2.8 UC chi tiết xem chi tiết khóa học, bài học .....	28
Hình 2.9 Biểu đồ UC đăng ký khóa học .....	30
Hình 2.10 Biểu đồ UC làm bài quiz trắc nghiệm .....	32
Hình 2.11 UC chi tiết chức năng nộp bài luyện tập lập trình .....	33
Hình 2.12 UC chi tiết chức năng tham gia contest .....	34
Hình 2.13 UC chi tiết chức năng nộp bài trong contest .....	35
Hình 2.14 UC chi tiết chức năng xem bảng xếp hạng contest .....	37
Hình 2.15 Biểu đồ UC xem bảng xếp hạng tổng .....	38
Hình 2.16 Biểu đồ UC quản lý người dùng .....	41
Hình 2.17 Biểu đồ UC quản lý khóa học .....	45
Hình 2.18 UC chi tiết chức năng quản lý bài học .....	50
Hình 2.19 Biểu đồ UC quản lý bài Quiz .....	56
Hình 2.20 Biểu đồ UC quản lý bài tập lập trình .....	61
Hình 2.21 UC chi tiết chức năng quản lý contest .....	66
Hình 2.22 UC chi tiết chức năng thêm problem vào contest .....	71
Hình 2.23 UC chi tiết chức năng xem thống kê học viên .....	73
Hình 2.24 UC chi tiết chức năng xem thống kê khóa học .....	74
Hình 2.25 UC chi tiết chức năng xem báo cáo doanh thu .....	76
Hình 2.26 UC chi tiết chức năng xem thống kê contest .....	77
Hình 2.27 Biểu đồ UC quản lý vai trò .....	79
Hình 2.28 Biểu đồ UC xem danh sách quyền .....	84
Hình 2.29 Biểu đồ UC xem danh sách loại nhóm quyền .....	85
Hình 2.30 UC chi tiết chức năng xem danh sách ngôn ngữ lập trình .....	86
Hình 2.31 UC chi tiết chức năng quản lý thẻ .....	87
Hình 2.32 Biểu đồ lớp toàn hệ thống .....	92
Hình 2.33 Cơ sở dữ liệu .....	93
Hình 2.34 Sơ đồ tuần tự chức năng xem khóa học .....	94
Hình 2.35 Sơ đồ tuần tự chức năng đăng ký khóa học .....	94
Hình 2.36 Sơ đồ tuần tự chức năng nộp bài tập luyện tập .....	95
Hình 2.37 Sơ đồ tuần tự chức năng Xem cuộc thi lập trình .....	95
Hình 2.38 Sơ đồ tuần tự chức năng đăng ký thi lập trình .....	96
Hình 2.39 Sơ đồ tuần tự chức năng nộp bài tập trong cuộc thi lập trình .....	96
Hình 2.40 Sơ đồ tuần tự chức năng xem bảng xếp hạng cuộc thi .....	97
Hình 2.41 Sơ đồ tuần tự chức năng xem bảng xếp hạng tổng .....	97
Hình 2.42 Sơ đồ tuần tự chức năng tạo khóa học .....	98

Hình 2.43 Sơ đồ tuần tự chức năng tạo bài học .....	98
Hình 2.44 Sơ đồ tuần tự chức năng tạo quiz.....	99
Hình 2.45 Sơ đồ tuần tự chức năng tạo bài tập lập trình .....	100
Hình 2.46 Sơ đồ tuần tự chức năng tạo cuộc thi lập trình .....	100
Hình 2.47 Sơ đồ tuần tự chức năng xem thống kê khóa học .....	101
Hình 2.48 Sơ đồ tuần tự chức năng xem thống kê cuộc thi lập trình.....	101
Hình 3.1 Model ALS.....	110
Hình 3.2 Model KNN .....	110
Hình 3.3 Kết quả chạy mô hình .....	111
Hình 3.4 Kết quả gợi ý trên giao diện.....	111
Hình 4.1 Giao diện chức năng đăng ký tài khoản.....	117
Hình 4.2 Giao diện chức năng đăng nhập.....	117
Hình 4.3 Giao diện chức năng hồ sơ người dùng .....	118
Hình 4.4 Giao diện chức năng danh sách khóa học .....	118
Hình 4.5 Giao diện chức năng đăng ký khóa học .....	119
Hình 4.6 Giao diện chức năng vào học .....	119
Hình 4.7 Giao diện chức năng luyện tập.....	120
Hình 4.8 Giao diện chức năng làm bài lập trình .....	120
Hình 4.9 Giao diện chức năng bảng xếp hạng luyện tập .....	121
Hình 4.10 Giao diện chức năng cuộc thi lập trình .....	121
Hình 4.11 Giao diện chức năng bảng xếp hạng cuộc thi .....	122
Hình 4.12 Giao diện chức năng bảng xếp hạng toàn cầu.....	122
Hình 4.13 Giao diện trang dashboard .....	123
Hình 4.14 Giao diện quản lý khóa học .....	123
Hình 4.15 Giao diện tạo mới khóa học .....	124
Hình 4.16 Giao diện quản lý bài học .....	124
Hình 4.17 Giao diện tạo mới bài học .....	125
Hình 4.18 Giao diện quản lý bài tập lập trình.....	125
Hình 4.19 Giao diện tạo bài tập lập trình.....	126
Hình 4.20 Giao diện quản lý Quiz .....	126
Hình 4.21 Giao diện tạo mới bài Quiz.....	127
Hình 4.22 Giao diện quản lý Contest.....	127
Hình 4.23 Giao diện tạo mới Contest .....	128
Hình 4.24 Giao diện quản lý người dùng.....	128
Hình 4.25 Giao diện tạo mới tài khoản người dùng .....	129
Hình 4.26 Giao diện quản lý vai trò.....	129
Hình 4.27 Giao diện tạo mới vai trò .....	130
Hình 4.28 Giao diện thống kê người dùng.....	130
Hình 4.29 Giao diện thống kê khóa học .....	131
Hình 4.30 Giao diện thống kê Contest.....	131
Hình 4.31 Giao diện thống kê doanh thu .....	132

## **DANH MỤC THUẬT NGỮ VÀ CÁC TỪ VIẾT TẮT**

Thuật ngữ hoặc từ viết tắt	Miêu tả
UC	Use case
Admin	Quản trị viên
User	Người dùng
Problems	Bài tập lập trình
Contest	Cuộc thi
Submissions	Bài nộp của người dùng
CB	Content-based Filtering
CF	Colabulary Filtering

## LỜI MỞ ĐẦU

Trong bối cảnh công nghệ thông tin ngày càng phát triển mạnh mẽ, lập trình đã trở thành một kỹ năng quan trọng và thiết yếu đối với sinh viên ngành Công nghệ thông tin cũng như người học trong nhiều lĩnh vực khác nhau. Việc học lập trình hiệu quả không chỉ yêu cầu nắm vững kiến thức lý thuyết mà còn cần có môi trường thực hành thường xuyên và cơ chế đánh giá chính xác năng lực của người học.

Xuất phát từ nhu cầu đó, đề tài “**Xây dựng website học và thi lập trình trực tuyến**” được thực hiện nhằm xây dựng một hệ thống hỗ trợ học tập và đánh giá kỹ năng lập trình trên nền tảng web. Hệ thống tích hợp các chức năng học tập, luyện tập và thi đấu lập trình, giúp người học tiếp cận kiến thức một cách linh hoạt, chủ động và hiệu quả.

Thông qua việc thực hiện đề tài, nhóm chúng em mong muốn vận dụng các kiến thức đã học trong quá trình đào tạo để xây dựng một sản phẩm có tính ứng dụng thực tiễn cao. Đây cũng là cơ hội để nhóm rèn luyện kỹ năng phân tích, thiết kế và triển khai một hệ thống phần mềm hoàn chỉnh.

# CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI VÀ CÁC CÔNG NGHỆ SỬ DỤNG

## 1.1. Giới thiệu đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 và sự phát triển mạnh mẽ của công nghệ thông tin, lập trình ngày càng trở thành một kỹ năng quan trọng và thiết yếu đối với sinh viên cũng như người đi làm trong nhiều lĩnh vực khác nhau. Nhu cầu học lập trình không chỉ dừng lại ở việc tiếp thu kiến thức lý thuyết mà còn đòi hỏi khả năng thực hành thường xuyên, được đánh giá chính xác và có môi trường trao đổi, học hỏi hiệu quả.

Hiện nay, đã có nhiều nền tảng học lập trình trực tuyến trên thế giới như Coursera, Udemy, LeetCode, Codeforces,... Tuy nhiên, đa số các hệ thống này chỉ tập trung vào một khía cạnh riêng lẻ như học lý thuyết hoặc luyện tập thuật toán, chưa tích hợp đầy đủ các chức năng học tập, thực hành, thi đấu và xây dựng cộng đồng trong một hệ thống thống nhất.

Xuất phát từ thực tế đó, đề tài “Hệ thống học và thi lập trình trực tuyến” được lựa chọn nhằm xây dựng một nền tảng học tập toàn diện, hiện đại và dễ mở rộng. Hệ thống cho phép người học tiếp cận kiến thức lập trình mọi lúc, mọi nơi thông qua các khóa học trực tuyến đa dạng, kết hợp giữa lý thuyết, bài tập thực hành và các kỳ thi lập trình. Đặc biệt, hệ thống hỗ trợ trình biên dịch trực tuyến, cơ chế chấm điểm tự động với các test case ẩn và bảng xếp hạng theo thời gian thực, giúp người học nhận được phản hồi nhanh chóng và khách quan về năng lực của bản thân.

## 1.2. Mô tả đề tài

### 1.2.1. Tổng quan hệ thống

Hệ thống học và thi lập trình trực tuyến là một nền tảng web tích hợp được thiết kế nhằm phục vụ đồng thời ba nhu cầu cốt lõi trong quá trình học lập trình: học lý thuyết – thực hành lập trình – đánh giá năng lực thông qua thi đấu. Hệ thống cho phép người học tiếp cận kiến thức lập trình một cách có hệ thống, thực hành trực tuyến trên trình duyệt và được chấm điểm tự động dựa trên bộ test chuẩn, đồng thời tạo ra môi trường cạnh tranh và cộng đồng học tập tích cực.

### 1.2.2. Đối tượng sử dụng và phạm vi ứng dụng

Hệ thống hướng đến các nhóm đối tượng chính sau:

- Sinh viên ngành Công nghệ thông tin: sử dụng hệ thống để học lập trình cơ bản đến nâng cao, luyện tập thuật toán và tham gia các kỳ thi lập trình.

- Người tự học lập trình, học sinh xuất phát sớm: tiếp cận các khóa học trực tuyến, thực hành và thực chiến nâng cao năng lực của bản thân.
- Giảng viên, quản trị viên: xây dựng nội dung khóa học, tạo bài toán, tổ chức kỳ thi và theo dõi tiến độ học tập của học viên

Phạm vi ứng dụng của hệ thống tập trung vào môi trường web, cho phép người dùng truy cập mọi lúc, mọi nơi thông qua trình duyệt, không phụ thuộc vào thiết bị hay hệ điều hành cụ thể.

### **1.2.3. Các chức năng chính của hệ thống**

#### *1.2.3.1. Chức năng dành cho người dùng hệ thống (user)*

- Quản lý tài khoản và hồ sơ cá nhân: Người học có thể đăng ký, đăng nhập hệ thống, cập nhật thông tin cá nhân, theo dõi lịch sử học tập, lịch sử nộp bài và kết quả các kỳ thi đã tham gia.
- Đăng ký và xem các khóa học, bài học: Hệ thống cho phép người học xem danh sách các khóa học, tìm kiếm và lọc theo ngôn ngữ lập trình, mức độ, mức học phí,... Mỗi khóa học được tổ chức theo cấu trúc bài học rõ ràng, giúp người học tiếp cận kiến thức theo trình độ hợp lý. Người dùng có thể đăng ký khóa học và truy cập vào khóa học, bài học sau khi đã đăng ký thành công hoặc được phép truy cập vào chi tiết khóa học, bài học để tiến hành học lập trình.
- Thực hiện các bài quiz: Trong mỗi khóa học sẽ có các bài quiz tổng kết kiến thức để người dùng thực hiện làm bài và đánh giá được năng lực sau khi học xong bài học, kết quả được chấm tự động và sẽ lưu lại để đánh giá quá trình học tập của người dùng.
- Thực hành lập trình trực tuyến: Hệ thống tích hợp trình soạn thảo và trình biên dịch trực tuyến, hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C/C++, Java, Python,... Người học có thể viết, chạy và kiểm tra chương trình ngay trên website hoặc có thể nộp file mã nguồn của mình. Bài làm được đưa vào môi trường sandbox để biên dịch và chạy thử với bộ testcase. Kết quả chấm được trả về dưới các trạng thái chuẩn như: Accepted, Wrong Answer, Time Limit Exceeded, Memory Limit Exceeded, Runtime Error,...
- Bảng xếp hạng và đánh giá năng lực: Hệ thống cung cấp bảng xếp hạng theo từng cuộc thi và bảng xếp hạng toàn hệ thống giúp người học theo dõi được kết quả của mình so với những người dùng khác của hệ thống.

#### *1.2.3.2. Chức năng dành cho quản trị viên*

- Quản lý khóa học và nội dung học tập: Admin có thể tạo, chỉnh sửa, xóa khóa học; sắp xếp bài học; upload video, tài liệu và thiết kế các bài quiz phù hợp với mục tiêu đào tạo.
- Quản lý bài toán lập trình: Hệ thống cho phép admin tạo và quản lý ngân hàng bài toán lập trình, bao gồm mô tả đề bài, ví dụ mẫu, ràng buộc về thời gian và bộ nhớ, cũng như bộ test case ẩn phục vụ chấm điểm tự động.
- Tổ chức các cuộc thi lập trình: Admin có thể tạo contest với thời gian bắt đầu/kết thúc, lựa chọn danh sách bài toán, cấu hình chế độ công khai hoặc riêng tư, phục vụ cho thi học phần hoặc thi đấu nội bộ.
- Quản lý người dùng và phân quyền: Admin quản lý danh sách người dùng, phân quyền (quản trị viên, học viên), đảm bảo an toàn và tính đúng đắn của hệ thống.
- Quản lý các dữ liệu về ngôn ngữ lập trình, các thẻ (tag) dùng cho khóa học, bài tập, cuộc thi
- Thông kê và báo cáo: Hệ thống cung cấp các báo cáo về số lượng người học, tiến độ học tập, kết quả thi, cũng như thông kê các khóa học phổ biến nhằm hỗ trợ việc đánh giá và phát triển hệ thống.

### **1.3. Công cụ và công nghệ sử dụng**

#### **1.3.1. Ngôn ngữ và nền tảng phát triển**

##### **1.3.1.1. Python**

Python được lựa chọn là ngôn ngữ lập trình chính cho phía backend của hệ thống học và thi lập trình trực tuyến. Đây là một ngôn ngữ bậc cao, mã nguồn mở, có cú pháp đơn giản, dễ đọc và dễ bảo trì, rất phù hợp cho việc phát triển các hệ thống web có quy mô vừa và lớn.

Python là ngôn ngữ thông dịch, hỗ trợ nhiều mô hình lập trình khác nhau như lập trình hướng đối tượng, lập trình hàm và lập trình thủ tục. Cú pháp của Python gần với ngôn ngữ tự nhiên, giúp giảm số lượng dòng mã và tăng khả năng bảo trì chương trình. Nhờ đặc điểm này, Python được sử dụng rộng rãi trong giảng dạy và nghiên cứu cũng như trong các dự án phần mềm thực tế.

Một ưu điểm nổi bật của Python là hệ sinh thái thư viện phong phú và cộng đồng người dùng lớn. Python cung cấp nhiều thư viện hỗ trợ phát triển ứng dụng web, xử lý dữ liệu, trí tuệ nhân tạo, học máy và tự động hóa. Điều này giúp Python trở thành một trong những ngôn ngữ lập trình phổ biến nhất hiện nay.

### *1.3.1.2. Javascript*

JavaScript là ngôn ngữ thông dịch, hoạt động chủ yếu trên phía client (trình duyệt), tuy nhiên hiện nay JavaScript cũng có thể được sử dụng ở phía server thông qua nền tảng Node.js. Ngôn ngữ này hỗ trợ nhiều mô hình lập trình như lập trình hướng đối tượng dựa trên prototype, lập trình hàm và lập trình hướng sự kiện, giúp lập trình viên linh hoạt trong việc xây dựng ứng dụng.

Một đặc điểm quan trọng của JavaScript là khả năng thao tác trực tiếp với DOM (Document Object Model), cho phép thay đổi cấu trúc, nội dung và giao diện của trang web theo thời gian thực. Điều này giúp tạo ra các ứng dụng web động, có khả năng phản hồi nhanh với hành động của người dùng, nâng cao trải nghiệm người dùng.

JavaScript có hệ sinh thái thư viện và framework rất phong phú như ReactJS, Angular, VueJS,... giúp đơn giản hóa quá trình phát triển giao diện người dùng và xây dựng các ứng dụng web hiện đại theo mô hình single-page application (SPA). Bên cạnh đó, JavaScript hỗ trợ tốt các kỹ thuật giao tiếp bất đồng bộ như AJAX, Fetch API và WebSocket, cho phép trao đổi dữ liệu hiệu quả giữa client và server.

Nhờ tính linh hoạt, khả năng chạy trên nhiều nền tảng và sự phổ biến rộng rãi trong cộng đồng phát triển web, JavaScript hiện là một trong những ngôn ngữ lập trình quan trọng nhất trong lĩnh vực phát triển ứng dụng web hiện nay.

### **1.3.2. Framework backend – Django**

Django là một framework web mã nguồn mở được phát triển bằng ngôn ngữ Python, ra đời vào năm 2005 bởi Django Software Foundation. Django được thiết kế với mục tiêu giúp lập trình viên xây dựng các ứng dụng web một cách nhanh chóng, an toàn và có khả năng mở rộng cao.

Django tuân theo mô hình kiến trúc MTV (Model – Template – View), trong đó Model chịu trách nhiệm quản lý dữ liệu và tương tác với cơ sở dữ liệu, Template đảm nhiệm việc hiển thị giao diện, còn View xử lý logic nghiệp vụ và điều hướng luồng xử lý của ứng dụng. Kiến trúc này giúp tách biệt rõ ràng giữa dữ liệu, giao diện và xử lý nghiệp vụ, từ đó tăng tính dễ bảo trì và mở rộng của hệ thống.

Một trong những ưu điểm nổi bật của Django là việc cung cấp sẵn nhiều tính năng quan trọng như hệ thống xác thực người dùng, phân quyền, quản lý session, middleware và cơ chế bảo mật. Django hỗ trợ phòng chống các lỗ hổng phổ biến trong ứng dụng web như SQL Injection,

Cross-Site Scripting (XSS) và Cross-Site Request Forgery (CSRF), giúp nâng cao độ an toàn cho hệ thống.

Ngoài ra, Django tích hợp ORM (Object-Relational Mapping), cho phép thao tác với cơ sở dữ liệu thông qua các đối tượng Python thay vì viết trực tiếp câu lệnh SQL. Điều này giúp giảm sự phụ thuộc vào hệ quản trị cơ sở dữ liệu cụ thể và tăng hiệu quả trong quá trình phát triển.

### **1.3.3. Framework frontend – React**

ReactJS là một thư viện JavaScript mã nguồn mở do Facebook (nay là Meta) phát triển và công bố lần đầu vào năm 2013. ReactJS được thiết kế nhằm hỗ trợ xây dựng giao diện người dùng (User Interface – UI) cho các ứng dụng web hiện đại, đặc biệt là các ứng dụng có mức độ tương tác cao.

ReactJS dựa trên tư tưởng xây dựng giao diện theo mô hình component-based, trong đó giao diện được chia thành các thành phần độc lập, có thể tái sử dụng và dễ dàng quản lý. Mỗi component chịu trách nhiệm cho một phần giao diện cụ thể và có thể kết hợp với các component khác để tạo thành một giao diện hoàn chỉnh. Cách tiếp cận này giúp tăng tính linh hoạt, khả năng mở rộng và dễ bảo trì mã nguồn.

Một đặc điểm nổi bật của ReactJS là cơ chế Virtual DOM. Thay vì cập nhật trực tiếp lên DOM thật mỗi khi dữ liệu thay đổi, ReactJS sẽ cập nhật trên Virtual DOM trước, sau đó so sánh và chỉ áp dụng những thay đổi cần thiết lên DOM thật. Cơ chế này giúp cải thiện hiệu năng hiển thị và mang lại trải nghiệm người dùng mượt mà hơn.

ReactJS hỗ trợ tốt việc xây dựng các ứng dụng Single Page Application (SPA), cho phép tải trang một lần và cập nhật nội dung động mà không cần tải lại toàn bộ trang. Ngoài ra, ReactJS dễ dàng tích hợp với các thư viện và framework khác trong hệ sinh thái JavaScript, hỗ trợ các kỹ thuật quản lý trạng thái, định tuyến và giao tiếp bát đồng bộ với server.

### **1.3.4. Cơ sở dữ liệu – MySQL**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được phát triển bởi công ty MySQL AB và hiện nay thuộc sở hữu của Oracle Corporation. MySQL được thiết kế để lưu trữ, quản lý và truy xuất dữ liệu một cách hiệu quả, đặc biệt phù hợp với các ứng dụng web có quy mô vừa và lớn.

MySQL sử dụng ngôn ngữ truy vấn có cấu trúc SQL (Structured Query Language) để thao tác dữ liệu, cho phép thực hiện các thao tác như tạo bảng, thêm, sửa, xóa và truy vấn dữ liệu. Hệ

quản trị này hỗ trợ đầy đủ các đặc tính của mô hình quan hệ như khóa chính, khóa ngoại và các ràng buộc toàn vẹn, giúp đảm bảo tính nhất quán và toàn vẹn của dữ liệu trong hệ thống.

Một ưu điểm nổi bật của MySQL là hiệu năng cao và khả năng xử lý đồng thời tốt, cho phép nhiều người dùng truy cập và thao tác trên cơ sở dữ liệu cùng lúc. MySQL cũng hỗ trợ nhiều cơ chế lưu trữ khác nhau (storage engine), trong đó InnoDB là engine phổ biến nhất, cung cấp các tính năng như giao dịch (transaction), khóa dòng (row-level locking) và khôi phục dữ liệu sau sự cố.

### **1.3.5. Công nghệ hỗ trợ chấm bài lập trình DOMjudge**

DOMjudge là một hệ thống chấm bài lập trình tự động (Online Judge) mã nguồn mở, được phát triển nhằm phục vụ cho các cuộc thi lập trình và các hoạt động đào tạo, đánh giá năng lực lập trình. DOMjudge hiện được sử dụng rộng rãi trong các kỳ thi lập trình sinh viên và các hệ thống luyện tập lập trình trực tuyến.

DOMjudge cho phép quản lý bài toán lập trình với đầy đủ thông tin như mô tả đề bài, dữ liệu vào/ra, giới hạn thời gian (Time Limit) và bộ nhớ (Memory Limit). Hệ thống tự động biên dịch và thực thi chương trình của người dùng trên các bộ test case ẩn, từ đó đưa ra kết quả đánh giá một cách khách quan và chính xác.

Một đặc điểm quan trọng của DOMjudge là cơ chế sandbox trong quá trình chấm bài. Mỗi bài nộp được thực thi trong môi trường cách ly, giúp đảm bảo an toàn hệ thống và tính công bằng khi đánh giá. DOMjudge hỗ trợ nhiều ngôn ngữ lập trình phổ biến như C, C++, Java, Python,... và cung cấp các trạng thái chấm bài tiêu chuẩn như Accepted, Wrong Answer, Time Limit Exceeded, Memory Limit Exceeded và Runtime Error.

Nhờ tính ổn định, độ chính xác cao và khả năng mở rộng, DOMjudge là giải pháp phù hợp để tích hợp vào các hệ thống học và thi lập trình trực tuyến.

### **1.3.6. Công cụ phát triển và triển khai**

Trong quá trình xây dựng và triển khai hệ thống học và thi lập trình trực tuyến, đề tài sử dụng nhiều công cụ hỗ trợ nhằm nâng cao hiệu quả phát triển, quản lý mã nguồn và đảm bảo quá trình triển khai hệ thống được ổn định, đồng nhất.

*Visual Studio Code (VS Code)* là môi trường phát triển tích hợp (IDE) chính được sử dụng trong đề tài. VS Code hỗ trợ nhiều ngôn ngữ lập trình như Python và JavaScript, đồng thời cung cấp hệ thống extension phong phú giúp lập trình viên dễ dàng viết mã, gỡ lỗi và quản lý dự án.

Giao diện trực quan và khả năng tùy biến cao của VS Code giúp nâng cao năng suất trong quá trình phát triển backend và frontend.

*Git* là hệ thống quản lý phiên bản được sử dụng để theo dõi lịch sử thay đổi của mã nguồn. Thông qua Git, các thay đổi trong quá trình phát triển được quản lý chặt chẽ, giúp dễ dàng quay lại các phiên bản trước khi cần thiết. GitHub được sử dụng làm nền tảng lưu trữ mã nguồn và hỗ trợ làm việc nhóm, cho phép nhiều thành viên cùng phát triển, kiểm soát xung đột và đánh giá mã nguồn thông qua cơ chế pull request.

*Postman* là công cụ hỗ trợ kiểm thử các API trong quá trình phát triển backend. Postman cho phép gửi các yêu cầu HTTP với nhiều phương thức khác nhau, kiểm tra dữ liệu trả về và phát hiện lỗi sớm trong quá trình xây dựng các dịch vụ web.

*Docker* được sử dụng để đóng gói ứng dụng và các thành phần liên quan thành các container độc lập. Việc sử dụng Docker giúp đảm bảo môi trường chạy của hệ thống nhất quán giữa các máy phát triển và máy chủ triển khai, đồng thời hỗ trợ tốt cho việc triển khai các thành phần như backend Django và hệ thống chấm bài lập trình.

Nhờ việc sử dụng các công cụ phát triển và triển khai phù hợp, quá trình xây dựng hệ thống được thực hiện một cách hiệu quả, dễ quản lý và đảm bảo tính ổn định khi đưa vào vận hành thực tế.

#### 1.4. Tổng kết

Trước sự phát triển mạnh mẽ của công nghệ thông tin và nhu cầu ngày càng cao về việc học lập trình gắn liền với thực hành và đánh giá năng lực, việc xây dựng một nền tảng học tập tích hợp, hiện đại và có khả năng mở rộng là hết sức cần thiết. Hệ thống có thể được triển khai trong môi trường đào tạo thực tế tại các trường đại học, trung tâm đào tạo hoặc phục vụ cộng đồng tự học lập trình, đáp ứng xu hướng chuyển đổi số trong giáo dục hiện nay.

## CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Phân tích hệ thống

#### 2.1.1. Xác định tác nhân và chức năng

##### 2.1.1.1. Các tác nhân chính

- User: là tác nhân đại diện cho người dùng sử dụng hệ thống. User có thể là sinh viên, học viên hoặc bất kỳ người dùng nào có nhu cầu học và luyện tập lập trình trực tuyến.
- Admin: là tác nhân có quyền quản lý và vận hành hệ thống. Admin chịu trách nhiệm đảm bảo hệ thống hoạt động ổn định, nội dung học tập chính xác và phù hợp với mục tiêu đào tạo.

##### 2.1.1.2. Các chức năng chính và phạm vi hệ thống

- Chức năng của hệ thống cho User:
  - Đăng ký tài khoản: Cho phép người dùng tạo tài khoản mới để sử dụng hệ thống.
  - Đăng nhập hệ thống: Cho phép người dùng đăng nhập vào hệ thống bằng thông tin đã đăng ký.
  - Cập nhật hồ sơ cá nhân: Người dùng có thể cập nhật thông tin cá nhân và theo dõi lịch sử học tập.
  - Xem danh sách khóa học: Hiển thị danh sách các khóa học hiện có trên hệ thống.
  - Tìm kiếm khóa học: Cho phép người dùng tìm kiếm khóa học theo từ khóa.
  - Lọc khóa học: Cho phép người dùng lọc khóa học theo ngôn ngữ lập trình, trình độ và mức giá.
  - Xem chi tiết khóa học và bài học: Hiển thị thông tin chi tiết về khóa học và các bài học trong khóa học.
  - Đăng ký tham gia khóa học: Cho phép người dùng đăng ký tham gia các khóa học mong muốn.
  - Xem chi tiết bài học: Cho phép người dùng xem nội dung chi tiết của từng bài học.
  - Làm bài quiz trắc nghiệm: Cho phép người dùng thực hiện các bài kiểm tra trắc nghiệm trực tuyến.
  - Nộp bài luyện tập lập trình: Người dùng nộp bài lập trình để hệ thống chấm điểm tự động.
  - Tham gia contest: Cho phép người dùng tham gia các cuộc thi lập trình.
  - Nộp bài trong contest: Người dùng nộp bài giải trong quá trình tham gia cuộc thi.
  - Xem bảng xếp hạng contest: Hiển thị bảng xếp hạng của các cuộc thi lập trình.
  - Xem bảng xếp hạng global: Hiển thị bảng xếp hạng tổng hợp của người dùng trên toàn hệ thống.
- Chức năng của hệ thống cho Admin:
  - Quản lý người dùng: Thực hiện xem chi tiết, thêm mới, chỉnh sửa và xóa người dùng trong hệ thống.
  - Quản lý khóa học: Tạo mới, chỉnh sửa, xem chi tiết và xóa các khóa học.
  - Quản lý bài học: Thêm, chỉnh sửa, xem chi tiết và xóa các bài học thuộc khóa học.
  - Quản lý bài quiz: Tạo, chỉnh sửa, xem và xóa các bài quiz trắc nghiệm.

- Quản lý bài tập lập trình (Problem): Tạo mới, chỉnh sửa, xem chi tiết và xóa các bài tập lập trình.
- Quản lý contest: Tạo, chỉnh sửa, xem chi tiết và xóa các cuộc thi lập trình.
- Thêm bài tập lập trình vào contest: Cấu hình danh sách bài tập cho từng contest.
- Xem thống kê học viên: Theo dõi tình hình và số lượng học viên trong hệ thống.
- Xem thống kê khóa học: Theo dõi mức độ phổ biến và số lượng đăng ký khóa học.
- Xem báo cáo doanh thu: Theo dõi doanh thu từ các khóa học.
- Xem thống kê contest: Theo dõi số lượng người tham gia và kết quả các contest.
- Quản lý vai trò (Role): Thêm, chỉnh sửa, xem danh sách và xóa các vai trò.
- Xem danh sách quyền: Hiển thị danh sách các quyền trong hệ thống.
- Xem danh sách loại quyền: Hiển thị các nhóm quyền trong hệ thống.
- Xem danh sách ngôn ngữ lập trình: Hiển thị các ngôn ngữ lập trình được hỗ trợ.
- Quản lý thẻ (Tag): Thêm, chỉnh sửa, xem danh sách và xóa các thẻ phân loại.

### **2.1.2. Biểu đồ Usecase Tổng quát**

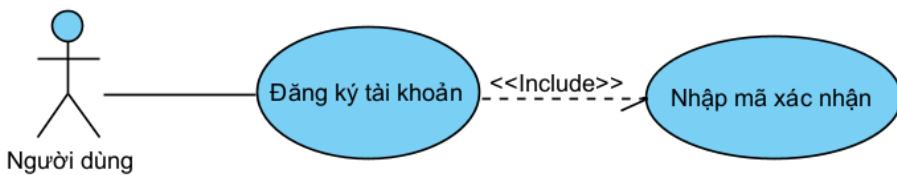


Hình 2.1 Biểu đồ Usecase tổng quan toàn hệ thống.

### 2.1.3. Kịch bản các chức năng chính của hệ thống

#### 2.1.3.1. Chức năng của User

##### a. Đăng ký tài khoản



Hình 2.2 Biểu đồ Usecase đăng ký tài khoản

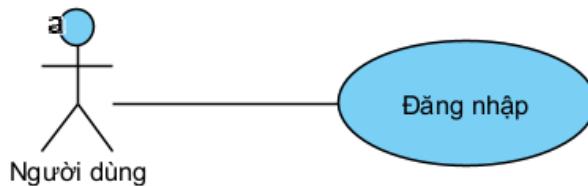
- Kịch bản chức năng đăng ký tài khoản

Use case	Đăng ký tài khoản
Actor	Người dùng
Tiền điều kiện	Người dùng có mong muốn tạo tài khoản mới
Hậu điều kiện	Người dùng tạo tài khoản thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng truy cập trang “Đăng ký tài khoản”.</li> <li>2. Hệ thống hiển thị form đăng ký, bao gồm các trường: Họ tên, Email, Mật khẩu, Xác nhận mật khẩu, Số điện thoại, Địa chỉ và nút “Đăng ký”.</li> <li>3. Người dùng nhập thông tin đăng ký, ví dụ:           <ul style="list-style-type: none"> <li>- Họ tên: Nguyễn Văn A</li> <li>- Email: <a href="mailto:nguyenvana@gmail.com">nguyenvana@gmail.com</a></li> <li>- Mật khẩu: VanA@12345</li> <li>- Xác nhận mật khẩu: VanA@12345</li> </ul>           Sau đó nhấn nút “Đăng ký”.         </li> <li>4. Hệ thống kiểm tra tính hợp lệ của dữ liệu và gửi mã xác nhận đến email người dùng đã đăng ký. Hệ thống hiển thị màn hình nhập mã xác nhận email.</li> <li>5. Người dùng nhập mã xác nhận, ví dụ: 839421, và nhấn nút “Xác nhận”.</li> </ol>

	<p>6. Hệ thống xác thực mã xác nhận, kích hoạt tài khoản và hiển thị thông báo “Đăng ký tài khoản thành công”.</p>
Ngoại lệ	<p>4. Thông báo nhập sai hoặc thiếu thông tin</p> <p>4.1. Hệ thống hiển thị thông báo “Vui lòng nhập đầy đủ thông tin và đảm bảo thông tin chính xác”</p> <p>4.2. Người dùng nhập lại thông tin, bổ sung những thông tin còn thiếu</p> <p>4.3. Hệ thống kiểm tra hợp lệ.</p> <p>4. Tài khoản email đã có trong hệ thống</p> <p>4.1. Hệ thống thông báo email đã được sử dụng</p> <p>4.2. Người dùng nhập lại email mới</p> <p>4.3. Hệ thống kiểm tra hợp lệ</p>

Bảng 2.1 Kịch bản chuẩn chức năng đăng ký tài khoản

**b. Đăng nhập**



Hình 2.3 Biểu đồ UC đăng nhập

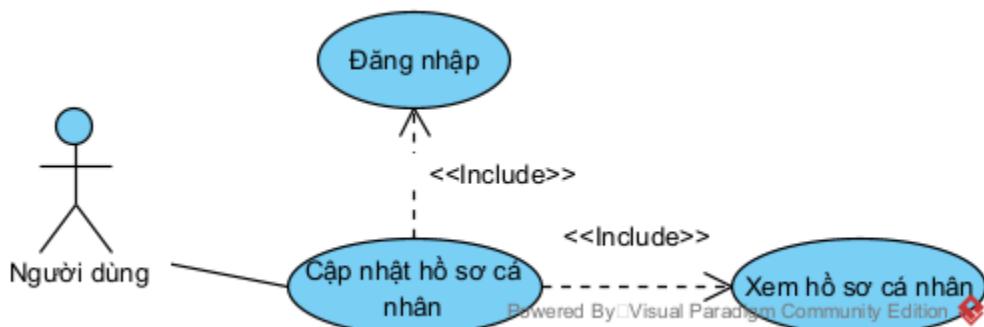
- Kịch bản chức năng đăng nhập

Use case	Đăng nhập tài khoản
Actor	Người dùng
Tiền điều kiện	Người dùng đã có tài khoản
Hậu điều kiện	Người dùng đăng nhập tài khoản thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>Người dùng truy cập trang “Đăng nhập tài khoản”</li> </ol>

	<ol style="list-style-type: none"> <li>2. Hệ thống hiển thị form đăng nhập (gồm: username, password).</li> <li>3. Người dùng nhập đầy đủ thông tin cần thiết và bấm nút “Đăng nhập”.</li> <li>4. Hệ thống kiểm tra dữ liệu hợp lệ và hiển thị thông báo đăng nhập thành công.</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>4. Thông báo nhập sai hoặc thiếu thông tin đăng nhập             <ol style="list-style-type: none"> <li>4.1. Hệ thống hiển thị thông báo “Vui lòng nhập đầy đủ thông tin và đảm bảo thông tin chính xác”</li> <li>4.2. Người dùng nhập lại thông tin, bổ sung những thông tin còn thiếu</li> <li>4.3. Hệ thống kiểm tra hợp lệ.</li> </ol> </li> </ol>

Bảng 2.2 Kịch bản chuẩn chức năng đăng nhập

### c. Cập nhật hồ sơ cá nhân



Hình 2.4 UC chi tiết chức năng cập nhật hồ sơ cá nhân

- Kịch bản chức năng cập nhật hồ sơ cá nhân

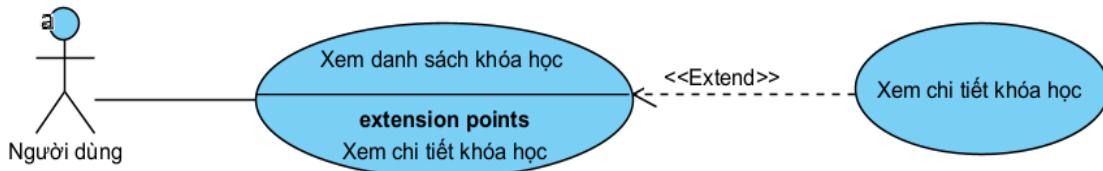
Use case	Cập nhật hồ sơ cá nhân
Actor	Học viên
Tiền điều kiện	Học viên đã đăng nhập thành công vào hệ thống
Hậu điều kiện	Học viên thay đổi được hồ sơ cá nhân của bản thân

Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập thành công, từ giao diện chính, học viên ấn vào avatar hoặc tên của học viên ở góc trên bên phải giao diện</li> <li>2. Hệ thống hiển thị dropdown danh sách chức năng trong đó có chức năng “Hồ sơ”</li> <li>3. Học viên ấn vào chức năng Hồ sơ</li> <li>4. Giao diện hồ sơ cá nhân của học viên hiện lên trong đó có thông tin của học viên cùng nút “Chỉnh sửa thông tin”</li> <li>5. Học viên ấn vào nút “Chỉnh sửa thông tin”</li> <li>6. Hệ thống hiển thị form chỉnh sửa các trường thông tin gồm avatar, các thông tin được phép chỉnh sửa cùng nút “Lưu” và “Hủy”, các thông tin được phép chỉnh sửa gồm: <ul style="list-style-type: none"> <li>- Họ và tên</li> <li>- Email</li> <li>- Mô tả</li> <li>- Giới tính</li> <li>- Ngày sinh</li> <li>- Số điện thoại</li> <li>- Địa chỉ</li> </ul> </li> <li>7. Học viên thay đổi thông tin và ấn nút “Lưu”</li> <li>8. Hệ thống cập nhật thông tin học viên vừa thay đổi, thông báo cập nhật thành công và trả lại giao diện hồ sơ cá nhân của học viên với thông tin mới do học viên đã chỉnh sửa</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>7. Học viên nhập thiếu các thông tin bắt buộc và ấn “Lưu” <ol style="list-style-type: none"> <li>7.1. Hệ thống báo thiếu dữ liệu bắt buộc</li> <li>7.2. Học viên nhập lại đầy đủ thông tin bắt buộc và ấn “Lưu”</li> </ol> <p>Quay về bước 8</p> </li> <li>7. Học viên nhập email trùng với email đã tồn tại trong hệ thống và ấn “Lưu” <ol style="list-style-type: none"> <li>7.1. Hệ thống báo đã có email tồn tại</li> <li>7.2. Học viên nhập lại email khác không trùng và ấn “Lưu”</li> </ol> </li> </ol>

	<p>Quay về bước 8</p> <p>7. Học viên ấn nút “Hủy”</p> <p>7.1 Hệ thống quay trở lại giao diện hồ sơ cá nhân của học viên</p>
--	---

Bảng 2.3 Kịch bản chuẩn chức năng cập nhật hồ sơ cá nhân

d. Xem danh sách khóa học



Hình 2.5 Biểu đồ UC xem danh sách khóa học

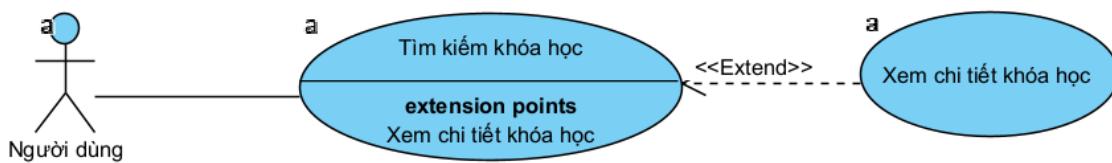
- Kịch bản chức năng xem danh sách khóa học

Use case	Xem danh sách khóa học
Actor	Người dùng
Tiền điều kiện	Người dùng truy cập vào website
Hậu điều kiện	Người dùng xem được thông tin giới thiệu của các khóa học
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng truy cập trang “Danh sách khóa học”.</li> <li>2. Hệ thống hiển thị danh sách các khóa học dưới dạng các card, mỗi card gồm các thông tin:           <ul style="list-style-type: none"> <li>- Tên khóa học</li> <li>- Mô tả ngắn</li> <li>- Giá tiền</li> </ul> </li> <li>3. Người dùng chọn một khóa học để xem chi tiết, ví dụ: “Lập trình Python cơ bản”.</li> <li>4. Hệ thống hiển thị thông tin chi tiết của khóa học, bao gồm:           <ul style="list-style-type: none"> <li>- Tên khóa học: Lập trình Python cơ bản</li> <li>- Mô tả chi tiết khóa học</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>- Danh sách bài học</li> <li>- Giá tiền: 1.200.000 VNĐ</li> <li>- Trạng thái khóa học</li> </ul>
Ngoại lệ	<p>2. Không có khóa học nào tồn tại</p> <p>2.1. Hệ thống hiển thị thông báo “Hiện tại chưa có khóa học nào”</p> <p>2.2. Người dùng rời khỏi trang “Danh sách khóa học”</p>

Bảng 2.4 Kịch bản chuẩn chức năng xem danh sách khóa học

e. Tìm kiếm khóa học



Hình 2.6 Biểu đồ UC tìm kiếm khóa học

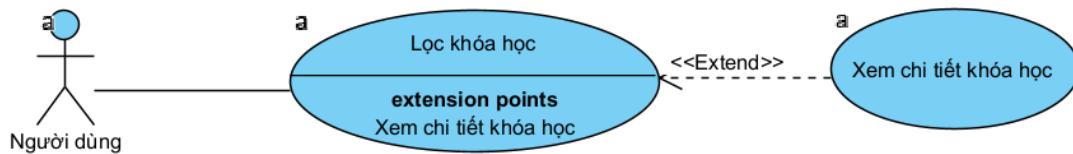
- Kịch bản chức năng tìm kiếm khóa học

Use case	Tìm kiếm khóa học
Actor	Người dùng
Tiền điều kiện	Người dùng đã truy cập vào website
Hậu điều kiện	Người dùng xem được thông tin khóa học đang tìm
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng nhập thông tin khóa học vào ô tìm kiếm “Khóa học lập trình C++” + bấm nút “Tìm kiếm”.</li> <li>2. Hệ thống hiển thị danh sách các khóa học dưới dạng các card, mỗi card gồm các thông tin:           <ul style="list-style-type: none"> <li>- Tên khóa học</li> <li>- Mô tả ngắn</li> <li>- Giá tiền</li> </ul> </li> </ol>

	<p>3. Người dùng chọn một khóa học để xem chi tiết, “Khóa học lập trình C++”.</p> <p>4. Hệ thống hiển thị thông tin chi tiết của khóa học, bao gồm:</p> <ul style="list-style-type: none"> <li>- Tên khóa học: Khóa học lập trình C++</li> <li>- Mô tả chi tiết khóa học</li> <li>- Danh sách bài học</li> <li>- Giá tiền: 1.200.000 VNĐ</li> </ul>
Ngoại lệ	<p>2. Không có khóa học nào tồn tại với thông tin tìm kiếm của người dùng</p> <p>2.1. Hệ thống hiển thị thông báo “Hiện tại chưa có khóa học nào”</p> <p>2.2. Người dùng nhập lại thông tin tìm kiếm</p> <p>2.3. Hệ thống hiển thị các khóa học thỏa mãn</p>

Bảng 2.5 Kịch bản chuẩn chức năng tìm kiếm khóa học

#### f. Lọc khóa học



Hình 2.7 Biểu đồ UC lọc khóa học

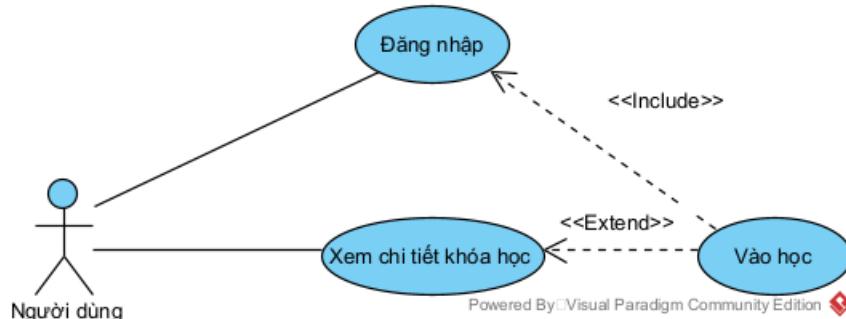
- Kịch bản chức năng lọc khóa học

Use case	Lọc khóa học
Actor	Người dùng
Tiền điều kiện	Người dùng đã truy cập vào website
Hậu điều kiện	Người dùng xem được thông tin khóa học đang tìm
Kịch bản chính	<p>1. Người dùng chọn các thông tin muốn lọc: Loại ngôn ngữ + cấp độ + giá, sau đó bấm Lọc</p>

	<p>2. Hệ thống hiển thị danh sách các khóa học thỏa mãn điều kiện tìm kiếm người dùng đã nhập dưới dạng các card gồm các thông tin: tên khóa học, mô tả ngắn, giá tiền</p> <p>3. Người dùng chọn một khóa học để xem chi tiết, “Khóa học lập trình C++”.</p> <p>4. Hệ thống hiển thị thông tin chi tiết của khóa học, bao gồm:</p> <ul style="list-style-type: none"> <li>- Tên khóa học: Khóa học lập trình C++</li> <li>- Mô tả chi tiết khóa học</li> <li>- Danh sách bài học</li> <li>- Giá tiền: 1.200.000 VNĐ</li> </ul>
Ngoại lệ	<p>2. Không có khóa học nào tồn tại với thông tin được lọc của người dùng</p> <p>2.1. Hệ thống hiển thị thông báo “Hiện tại chưa có khóa học nào”</p> <p>2.2. Người dùng nhập lại thông tin tìm kiếm</p> <p>2.3. Hệ thống hiển thị các khóa học thỏa mãn</p>

Bảng 2.6 Kịch bản chuẩn chức năng lọc khóa học

#### g. Xem chi tiết khóa học, bài học



Hình 2.8 UC chi tiết xem chi tiết khóa học, bài học

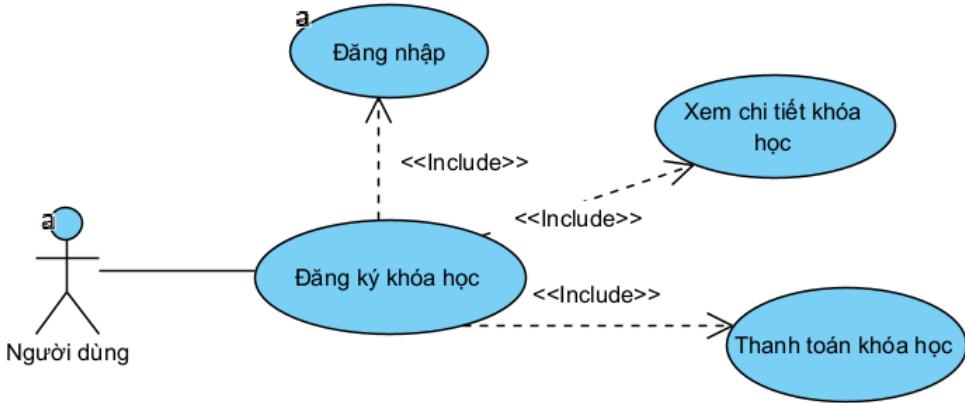
- Kịch bản chức năng xem chi tiết khóa học, bài học

Use case	Xem chi tiết khóa học, bài học
Actor	Học viên
Tiền điều kiện	Học viên đã đăng nhập thành công vào hệ thống

Hậu điều kiện	Học viên xem được khóa học, bài học
Kịch bản chính	<p>1. Học viên sau khi đăng nhập, từ giao diện chính, ấn vào menu Khóa học ở thanh menu</p> <p>2. Hệ thống hiển thị giao diện danh sách khóa học cho học viên, ví dụ có các khóa học:</p> <ul style="list-style-type: none"> <li>- Khóa học Python cơ bản</li> <li>- Java – Beginner</li> <li>- Java – Springboot</li> </ul> <p>3. Học viên chọn một khóa học cụ thể mà mình đã đăng ký hoặc khóa học miễn phí</p> <p>4. Hệ thống hiển thị chi tiết khóa học cho học viên gồm các thông tin tiêu đề, mô tả, giảng viên, số bài học, ngôn ngữ, cấp độ và danh sách bài học và nút vào học, nút quay lại.</p> <p>5. Học viên ấn vào nút vào học</p> <p>6. Hệ thống hiển thị bài học cho học viên gồm danh sách bài học bên trái, thông tin mỗi bài học bên phải tùy theo loại tài nguyên bài học (văn bản, file pdf, video,...)</p>
Ngoại lệ	<p>3. Học viên ấn vào khóa học mà mình chưa đăng ký nhưng vẫn được phép truy cập</p> <p>3.1 Hệ thống hiển thị giao diện chi tiết khóa học cho học viên gồm các thông tin tiêu đề, mô tả, giảng viên, số bài học, ngôn ngữ, cấp độ và danh sách bài học và nút đăng ký khóa học, nút quay lại.</p> <p>3.2 Học viên quay trở lại giao diện danh sách khóa học và ấn vào khóa học mà học viên đã đăng ký</p>

Bảng 2.7 Kịch bản chuẩn chức năng xem chi tiết khóa học, bài học

#### **h. Đăng ký khóa học**



Hình 2.9 Biểu đồ UC đăng ký khóa học

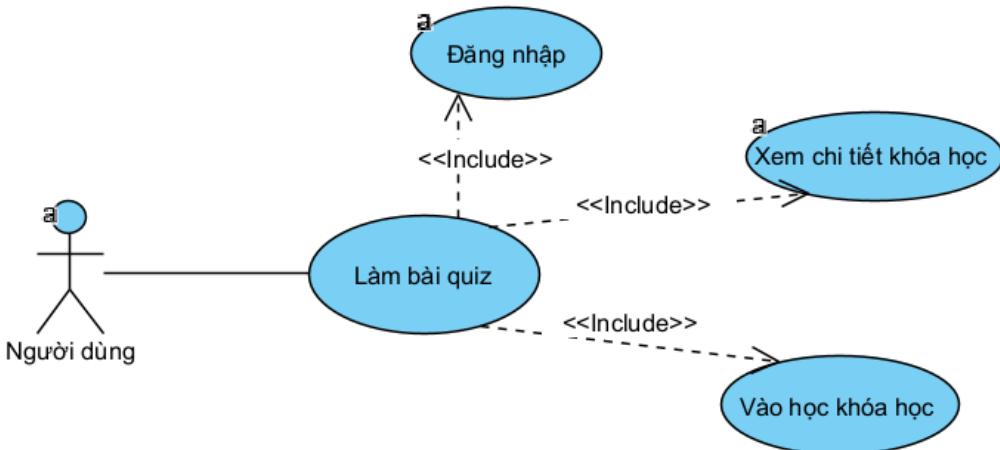
- Kịch bản chức năng đăng ký khóa học

Use case	Đăng ký tham gia khóa học
Actor	Người dùng
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Người dùng đăng ký khóa học thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>Người dùng truy cập trang “Danh sách khóa học”.</li> <li>Hệ thống hiển thị danh sách các khóa học để người dùng lựa chọn.</li> <li>Người dùng chọn một khóa học muốn đăng ký, ví dụ: “Lập trình Web với Django”.</li> <li>Hệ thống hiển thị thông tin chi tiết của khóa học, bao gồm: <ul style="list-style-type: none"> <li>Tên khóa học: Lập trình Web với Django</li> <li>Mô tả khóa học</li> <li>Danh sách bài học</li> <li>Giá khóa học: 299.000 VNĐ</li> </ul> </li> <li>Người dùng nhấn nút “Đăng ký khóa học”.</li> </ol>

	<p>6. Hệ thống hiển thị popup xác nhận đăng ký, hiển thị thông tin khóa học và số tiền cần thanh toán (299.000 VNĐ), kèm theo hai nút “Tiếp tục thanh toán” và “Hủy”.</p> <p>7. Người dùng chọn “Tiếp tục thanh toán”.</p> <p>8. Hệ thống hiển thị các phương thức thanh toán cho người dùng lựa chọn: app ngân hàng và ví điện tử; thẻ nội địa; thẻ thanh toán quốc tế; app VNPay.</p> <p>9. Người dùng chọn phương thức thanh toán, ví dụ: app ngân hàng và ví điện tử.</p> <p>10. Hệ thống chuyển hướng người dùng đến trang thanh toán của cổng VNPay.</p> <p>11. Người dùng thực hiện thanh toán và giao dịch thành công.</p> <p>12. Hệ thống nhận thông báo xác nhận giao dịch từ cổng thanh toán và hiển thị thông báo “Đăng ký khóa học thành công”.</p>
Ngoại lệ	<p>11. Giao dịch bị hủy hoặc lỗi</p> <p>11.1. Người dùng thanh toán nhưng báo lỗi và không thành công</p> <p>11.2. Hệ thống hiển thị thông báo giao dịch bị lỗi và yêu cầu thực hiện lại</p> <p>11.3. Người dùng thực hiện lại giao dịch và thanh toán thành công</p>

Bảng 2.8 Kích bản chuẩn chức năng đăng ký khóa học

*i. Làm bài quiz trắc nghiệm*



Hình 2.10 Biểu đồ UC làm bài quiz trắc nghiệm

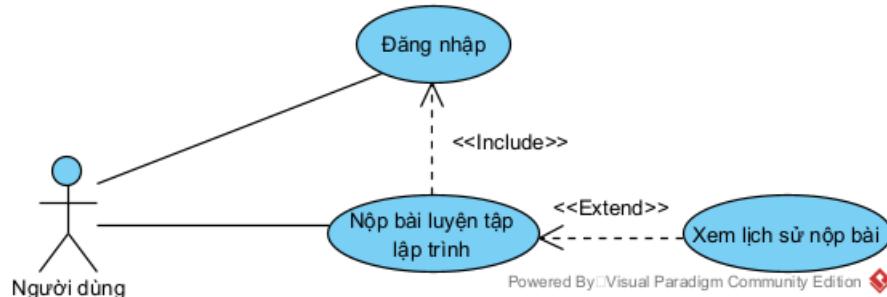
- Kịch bản chức năng làm bài quiz trắc nghiệm

Use case	Làm bài quiz trắc nghiệm
Actor	Người dùng
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có khóa học được đăng ký
Hậu điều kiện	Người dùng hoàn thành bài quiz và nhận được kết quả bài quiz
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng bấm vào phần danh sách khóa học đã đăng ký</li> <li>2. Hệ thống hiển thị danh sách các khóa học của người dùng</li> <li>3. Người dùng chọn 1 khóa học “Lập trình C++”</li> <li>4. Hệ thống hiển thị chi tiết khóa học bao gồm các bài học và các bài Quiz trắc nghiệm</li> <li>5. Người dùng chọn 1 bài quiz mà mình muốn làm “Lý thuyết về mảng trong C++”</li> <li>6. Hệ thống hiển thị các câu hỏi liên quan đến bài quiz mà người dùng lựa chọn</li> <li>7. Người dùng lựa chọn các đáp án cho từng câu hỏi, sau đó nhấn nút “Nộp bài” khi hoàn thành</li> <li>8. Hệ thống hiển thị thông báo “Nộp bài thành công” và hiển thị kết quả của bài Quiz.</li> </ol>

Ngoại lệ	<p>4. Khóa học người dùng đăng ký không có bài quiz</p> <p>4.1. Hệ thống hiển thị chi tiết khóa học không có phần quiz</p> <p>4.2. Người dùng quay lại giao diện danh sách khóa học và chọn khóa học khác.</p> <p>7. Hết thời gian làm bài</p> <p>7.1. Người dùng hết thời gian làm bài nhưng chưa bấm nộp bài</p> <p>7.2. Hệ thống tự động nộp bài và hiển thị kết quả của bài quiz</p> <p>7. Người dùng rời khỏi trang làm bài khi vẫn còn thời gian</p> <p>7.1. Trong thời gian làm bài, người dùng chuyển tab hoặc rời khỏi trang</p> <p>7.2. Hệ thống hiển thị cảnh báo với người dùng, nếu vượt quá số lần quy định tự động nộp bài</p>
----------	---

Bảng 2.9 Kịch bản chuẩn chức năng làm bài quiz trắc nghiệm

#### j. Nộp bài luyện tập lập trình



Hình 2.11 UC chi tiết chức năng nộp bài luyện tập lập trình

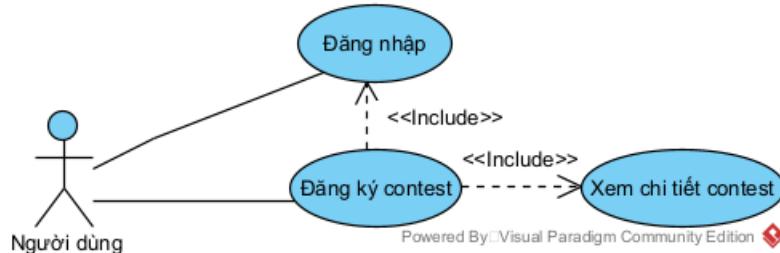
- Kịch bản chức năng nộp bài luyện tập lập trình

Use case	Nộp bài tập lập trình
Actor	Học viên
Tiền điều kiện	Học viên đã đăng nhập thành công vào hệ thống và có quyền vào trang chi tiết problem và đã vào trang chi tiết problem
Hậu điều kiện	Học viên nộp được bài lập trình của mình

Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi học viên đã vào được trang chi tiết problem, ở cuối trang sẽ có hai nút gồm “Code Editor” và “Upload File” và form thông tin submissions tùy theo loại bài nộp của User, mặc định là Code Editor</li> <li>2. Học viên tiến hành viết bài tập lập trình của mình vào Code Editor trên giao diện hoặc chọn Upload File để chọn file mã nguồn trong máy của học viên và tiến hành nộp bài</li> <li>3. Hệ thống ghi nhận bài nộp của học viên và hiển thị trạng thái bài submissions của học viên đang là “judging” rồi gửi submissions qua Domjudge để tiến hành chấm bài</li> <li>4. Domjudge tiến hành chấm bài của học viên và cập nhật trạng thái của submissions</li> <li>5. Hệ thống nhận kết quả từ Domjudge và gửi về trên giao diện cho học viên gồm kết quả chấm bài, chi tiết lỗi nếu có.</li> <li>6. User xem danh sách và chi tiết các bài mình đã nộp ngay bên dưới cùng của problem</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>2. Học viên upload file khác định dạng file được hệ thống hỗ trợ             <ol style="list-style-type: none"> <li>2.1 Hệ thống không nhận file</li> <li>2.2 Học viên upload lại file khác đúng định dạng được hỗ trợ</li> </ol> <p>Quay lại bước 3</p> </li> </ol>

Bảng 2.10 Kịch bản chuẩn chức năng nộp bài luyện tập lập trình

#### k. Tham gia contest



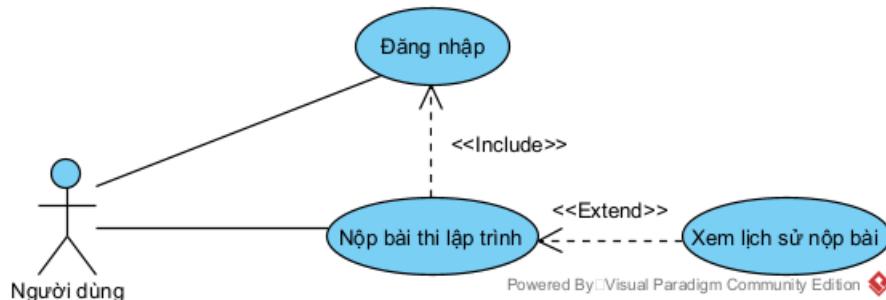
Hình 2.12 UC chi tiết chức năng tham gia contest

- Kịch bản chính chức năng tham gia contest

Use case	Tham gia contest
Actor	Học viên
Tiền điều kiện	Học viên đã đăng nhập thành công vào hệ thống
Hậu điều kiện	Học viên đăng ký tham gia contest thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, học viên chọn menu thi đấu ở toolbar</li> <li>2. Hệ thống hiển thị danh sách các contest đang diễn ra và sắp diễn ra và đã kết thúc cho học viên</li> <li>3. Học viên ấn vào một contest sắp diễn ra hoặc đang diễn ra</li> <li>4. Hệ thống hiển thị trang chi tiết contest gồm danh sách problem và nút “Đăng ký” cho học viên</li> <li>5. Học viên ấn vào nút “Đăng ký”</li> <li>6. Hệ thống báo thành công và quay về trang chi tiết contest đã ấn nút “Đăng ký”.</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>3. Học viên ấn vào một contest đã kết thúc           <ol style="list-style-type: none"> <li>3.1. Hệ thống hiển thị trang chi tiết contest gồm danh sách problem không có nút “Đăng ký”.</li> </ol> </li> </ol>

Bảng 2.11 Kịch bản chuẩn chức năng tham gia contest

#### I. Nộp bài trong contest



Hình 2.13 UC chi tiết chức năng nộp bài trong contest

- Kịch bản chức năng nộp bài trong contest

Use case	Nộp bài tập trong contest
----------	---------------------------

Actor	Học viên
Tiền điều kiện	Học viên đã đăng nhập thành công vào hệ thống và đã đăng ký contest, có quyền xem chi tiết contest và đã vào trang chi tiết contest
Hậu điều kiện	Học viên nộp được bài của mình trong contest
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Học viên chọn 1 problem muốn nộp bài trong danh sách problem của contest</li> <li>2. Hệ thống hiển thị giao diện chi tiết problem cho học viên, ở phía dưới trang sẽ có hai nút gồm “Code Editor” và “Upload File” và form thông tin submissions tùy theo loại bài nộp của User, mặc định là Code Editor</li> <li>3. Học viên tiến hành viết bài tập lập trình của mình vào Code Editor trên giao diện hoặc chọn Upload File để chọn file mã nguồn trong máy của học viên và tiến hành nộp bài</li> <li>4. Hệ thống ghi nhận bài nộp của học viên và hiển thị trạng thái bài submissions của học viên đang là “judging” rồi gửi submissions qua Domjudge để tiến hành chấm bài</li> <li>5. Domjudge tiến hành chấm bài của học viên và cập nhật trạng thái của submissions</li> <li>6. Hệ thống nhận kết quả từ Domjudge và gửi về trên giao diện cho học viên gồm kết quả chấm bài, chi tiết lỗi nếu có.</li> <li>7. User xem danh sách và chi tiết các bài mình đã nộp ngay bên dưới cùng của problem</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>3. Học viên upload file khác định dạng file được hệ thống hỗ trợ             <ol style="list-style-type: none"> <li>3.1 Hệ thống không nhận file</li> <li>3.2 Học viên chọn đúng file được hệ thống hỗ trợ và tiến hành nộp bài</li> </ol> </li> </ol> <p>Quay lại bước 4</p>

Bảng 2.12 Kịch bản chuẩn chức năng nộp bài trong contest

**m. Xem bảng xếp hạng contest**



Hình 2.14 UC chi tiết chức năng xem bảng xếp hạng contest

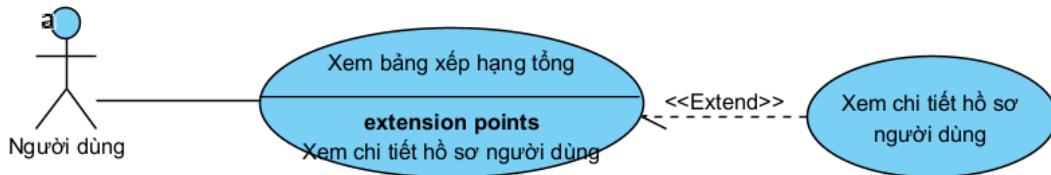
- Kịch bản chức năng xem bảng xếp hạng contest

Use case	Xem xếp hạng contest
Actor	Học viên
Tiền điều kiện	Học viên đã đăng nhập thành công vào hệ thống
Hậu điều kiện	Học viên nộp được bài lập trình của mình
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi đăng nhập thành công, học viên chọn menu Thi đấu toolbar từ giao diện chính</li> <li>Hệ thống hiển thị danh sách các contest đang diễn ra và sắp diễn ra cho học viên</li> <li>Học viên chọn 1 contest trong danh sách contest</li> <li>Hệ thống hiển thị trang chi tiết contest gồm 2 tab chính, 'Chi tiết' và 'Bảng xếp hạng'</li> <li>Học viên chọn tab 'Bảng xếp hạng'</li> <li>Hệ thống hiển thị Bảng xếp hạng contest cho học viên với 3 loại bảng xếp hạng:             <ul style="list-style-type: none"> <li>Bảng xếp hạng contest luyện tập: xếp theo số bài giải được giảm dần, sau đó xếp theo số bài thử giảm dần. Hiển thị 3 card chứa 3 top đầu với thông tin thứ tự, họ và tên, avatar, số bài nộp, số bài đã thử; hiển thị table chứa bảng xếp hạng đã phân</li> </ul> </li> </ol>

	<p>trang gồm các cột thông tin: thứ hạng, họ và tên, số bài nộp, số bài đã thử</p> <ul style="list-style-type: none"> <li>- Bảng xếp hạng contest ICPC: xếp theo số bài giải được giảm dần, sau đó xếp theo số penalty tăng dần, thời gian nộp bài lần cuối tăng dần, hiển thị table chứa bảng xếp hạng gồm các cột thông tin thứ tự, họ và tên, số bài đã giải, penalty, các cột tiếp theo là label các bài tập lập trình trong contest</li> <li>- Bảng xếp hạng contest OI: xếp theo số điểm giảm dần sau đó xếp theo thời gian nộp bài lần cuối tăng dần, hiển thị table chứa bảng xếp hạng gồm các cột thông tin thứ tự, họ và tên, số điểm</li> </ul>
Ngoại lệ	

Bảng 2.13 Kịch bản chuẩn chức năng xem bảng xếp hạng contest

#### n. Xem bảng xếp hạng global



Hình 2.15 Biểu đồ UC xem bảng xếp hạng tổng

- Kịch bản chức năng xem bảng xếp hạng global

Use case	Xem bảng xếp hạng Global
Actor	Người dùng
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống
Hậu điều kiện	Hiển thị xếp hạng của từng người trong hệ thống
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Người dùng truy cập mục “Bảng xếp hạng” (Global Ranking) từ thanh menu của hệ thống.</li> </ol>

2. Hệ thống hiển thị danh sách xếp hạng Global sắp xếp theo thứ hạng giảm dần với các thông tin:

- Thứ hạng (Rank #)
- Tên người dùng
- Cấp bậc (Pupil, Specialist, Grandmaster, ...)
- Rating hiện tại
- Rating cao nhất (Max Rating)
- Số contest đã tham gia

Hạng	Người dùng	Rank	Rating	Max Rating	Contests
🏆	User 54	GRANDMASTER	2494	1500	33
🏅	User 46	GRANDMASTER	2492	1500	37

Đồng thời hiển thị khu vực “Thứ hạng của bạn” để người dùng tiện theo dõi vị trí cá nhân:

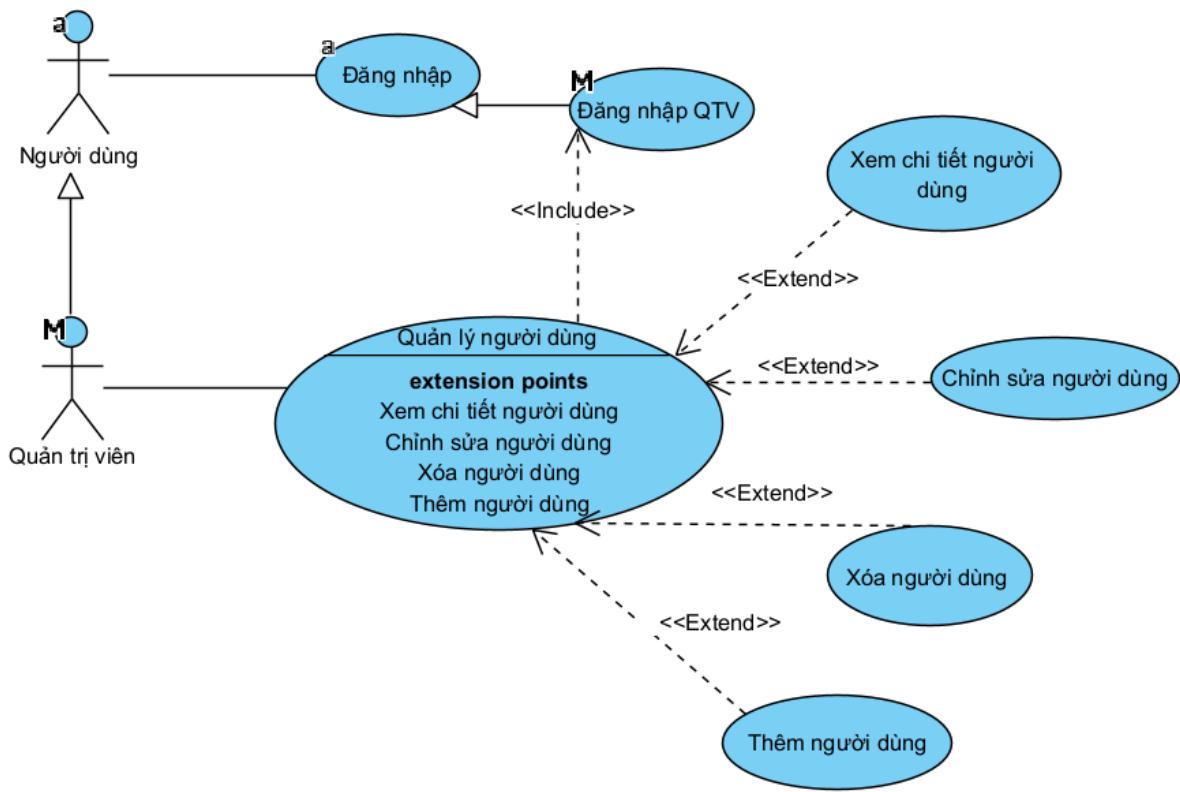
- Thứ hạng hiện tại: #152
- Cấp bậc: SPECIALIST
- Rating: 1580 / 1600 (Hiện tại / Cao nhất)
- Số contest đã tham gia: 12
- 3. Người dùng chọn một người dùng cụ thể trong danh sách, ví dụ: user\_0054.
- 4. Hệ thống chuyển hướng đến trang chi tiết của user\_0054 và hiển thị các thông tin:
  - Thông tin cơ bản:
    - + Tên người dùng: user\_0054
    - + Cấp bậc: GRANDMASTER
    - + Rating hiện tại: 2494
  - Chỉ số thống kê tổng quan:
    - + Số bài đã giải: 532
    - + Số contest đã tham gia: 45
    - + Tỷ lệ AC: 68.5%
    - + Tổng số lượt nộp bài: 1,240

	<ul style="list-style-type: none"> <li>- Thông tin chi tiết:           <ul style="list-style-type: none"> <li>+ Rating hiện tại: 2494</li> <li>+ Rating cao nhất: 2510</li> <li>+ Số contest đã tham gia: 45</li> </ul> </li> </ul>
Ngoại lệ	<ul style="list-style-type: none"> <li>2. Chưa có ai được xếp hạng           <ul style="list-style-type: none"> <li>2.1. Hệ thống hiển thị chưa có người được xếp hạng</li> <li>2.2. Người dùng quay trở về trang trước đó</li> </ul> </li>   <li>4. Không tìm thấy kết quả thỏa mãn thông tin lọc hoặc tìm kiếm của người dùng           <ul style="list-style-type: none"> <li>4.1. Hệ thống hiển thị danh sách trống</li> <li>4.2. Người dùng chọn lại các tiêu chí lọc, tìm kiếm</li> <li>4.3. Hệ thống hiển thị danh sách thỏa mãn</li> </ul> </li> </ul>

Bảng 2.14 Kịch bản chuẩn chức năng xem bảng xếp hạng global

#### 2.1.3.2. Chức năng của Admin

##### a. Quản lý người dùng



Hình 2.16 Biểu đồ UC quản lý người dùng

- Kịch bản chức năng xem chi tiết người dùng

Use case	Xem chi tiết thông tin người dùng
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin xem được thông tin chi tiết của người dùng
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý người dùng”</li> <li>2. Hệ thống hiển thị danh sách người dùng có trong hệ thống với một số thông tin: tên, email, vai trò,...</li> <li>3. Admin bấm chọn xem chi tiết</li> <li>4. Trang chi tiết hiển thị các thông tin chính: hồ sơ cá nhân, hoạt động, vai trò, lịch sử, trạng thái...</li> </ol>

Ngoại lệ	<p>4. Admin không có đủ quyền xem chi tiết</p> <p>4.1. Hệ thống thông báo admin không có quyền xem chi tiết thông tin về user này (người dùng cùng có vai trò admin)</p> <p>4.2. Admin chọn xem chi tiết 1 người dùng khác</p> <p>4.3. Hệ thống hiển thị chi tiết của người dùng đó</p>
----------	---

Bảng 2.15 Kịch bản chuẩn chức năng xem chi tiết người dùng

- Kịch bản chức năng thêm người dùng mới

Use case	Thêm người dùng mới
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin thêm người dùng thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý người dùng”</li> <li>2. Hệ thống hiển thị danh sách người dùng có trong hệ thống với một số thông tin: tên, email, vai trò,... + Nút “Thêm người dùng”</li> <li>3. Admin bấm chọn “Thêm người dùng”</li> <li>4. Hệ thống hiển thị form thêm người dùng gồm các trường: <ul style="list-style-type: none"> <li>- Họ tên</li> <li>- Email</li> <li>- Số điện thoại</li> <li>- Mật khẩu</li> <li>- Vai trò</li> <li>- Trạng thái</li> </ul> </li> <li>5. Admin tiến hành điền đầy đủ những thông tin bắt buộc + Bấm “Xác nhận”</li> <li>6. Hệ thống hiển thị thông báo “Thêm người dùng thành công”</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>6. Thêm người dùng thất bại do email đã tồn tại <ul style="list-style-type: none"> <li>6.1. Hệ thống thông báo “Thất bại do email đã tồn tại”</li> </ul> </li> </ol>

	<p>6.2. Admin nhập email khác và bấm Xác nhận</p> <p>6.3. Hệ thống thông báo “Thêm người dùng thành công”</p> <p>6. Dữ liệu nhập thiếu hoặc không hợp lệ</p> <p>6.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ”</p> <p>6.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận</p> <p>6.3. Hệ thống thông báo “Thêm người dùng thành công”</p>
--	--

Bảng 2.16 Kịch bản chuẩn chức năng thêm người dùng mới

- Kịch bản chức năng chỉnh sửa người dùng

Use case	Chỉnh sửa thông tin người dùng mới
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin chỉnh sửa thông tin người dùng thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý người dùng”</li> <li>2. Hệ thống hiển thị danh sách người dùng có trong hệ thống với một số thông tin: tên, email, vai trò,...</li> <li>3. Admin bấm chọn chỉnh sửa 1 người dùng trong danh sách</li> <li>4. Hệ thống hiển thị trang Form chỉnh sửa người dùng gồm các trường: <ul style="list-style-type: none"> <li>- Họ tên</li> <li>- Email</li> <li>- Số điện thoại</li> <li>- Avatar (tùy chọn)</li> <li>- Vai trò (User / Admin...)</li> <li>- Trạng thái tài khoản (Active / Inactive / Banned)</li> </ul> </li> <li>5. Admin thay đổi một hoặc nhiều trường: sửa tên, đổi vai trò, khóa tài khoản,...</li> </ol>

	<p>6. Hệ thống hiển thị thông báo: “Cập nhật thông tin người dùng thành công.”</p>
Ngoại lệ	<p>4. Không có quyền chỉnh sửa người dùng</p> <p>4.1. Hệ thống hiển thị thông báo admin không có quyền chỉnh sửa người dùng do quyền hạn tương đương</p> <p>4.2. Admin bấm Xác nhận, chọn chỉnh sửa người dùng khác</p> <p>4.3. Hệ thống hiển thị form chỉnh sửa người dùng</p> <p>6. Dữ liệu nhập thiếu hoặc không hợp lệ</p> <p>6.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ”</p> <p>6.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận</p> <p>6.3. Hệ thống thông báo “Thêm người dùng thành công”</p>

Bảng 2.17 Kịch bản chuẩn chức năng chỉnh sửa người dùng

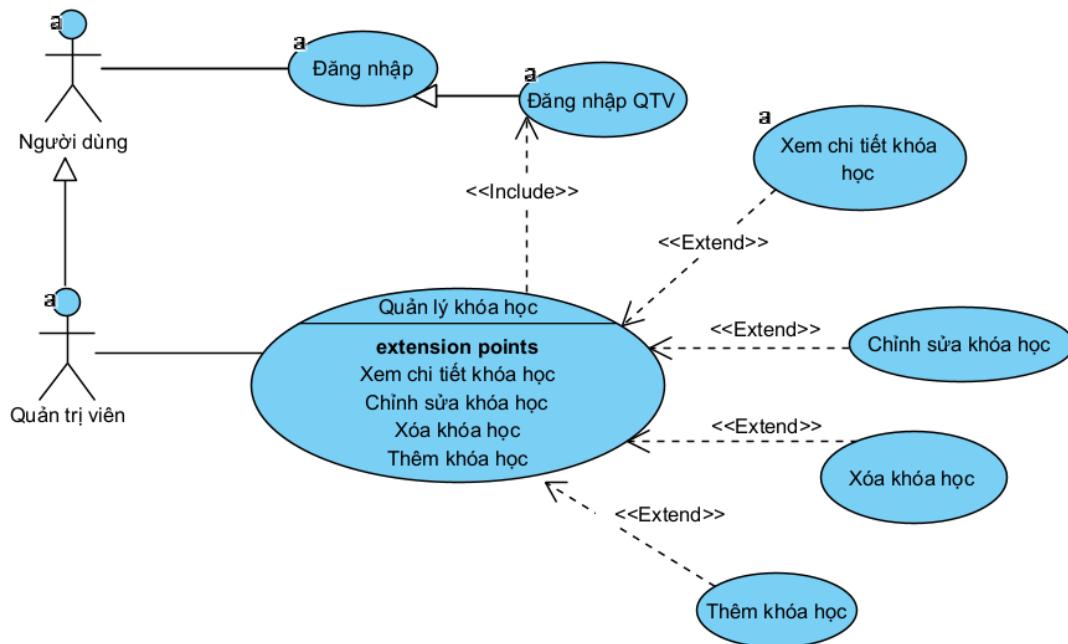
- Kịch bản chức năng xóa người dùng

Use case	Xóa người dùng
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin chỉnh xóa thông tin người dùng thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý người dùng”</li> <li>2. Hệ thống hiển thị danh sách người dùng có trong hệ thống với một số thông tin: tên, email, vai trò,...</li> <li>3. Admin bấm chọn xóa 1 người dùng trong danh sách</li> <li>4. Hệ thống hiển thị 1 popup chứa thông tin “Bạn có chắc chắn muốn xóa user này?” + nút “Xác nhận”</li> <li>5. Admin bấm nút “Xác nhận”</li> <li>6. Hệ thống hiển thị thông báo: “Xóa người dùng thành công”</li> </ol>

Ngoại lệ	4. Không có quyền xóa người dùng 4.1. Hệ thống hiển thị thông báo admin không có quyền xóa người dùng do quyền hạn tương đương 4.2. Admin bấm Xác nhận, chọn xóa người dùng khác 4.3. Hệ thống hiển thị form người dùng
----------	--

Bảng 2.18 Kịch bản chuẩn chức năng xóa người dùng

b. Quản lý khóa học



Hình 2.17 Biểu đồ UC quản lý khóa học

- Kịch bản chức năng tạo khóa học mới

Use case	Thêm khóa học mới
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin thêm khóa học thành công
Kịch bản chính	1. Admin truy cập trang “Quản lý khóa học”

	<p>2. Hệ thống hiển thị danh sách các khóa học có trong hệ thống với một số thông tin: tên khóa học, cấp độ, ngôn ngữ, trạng thái,... + Nút “Thêm khóa học”</p> <p>3. Admin bấm chọn “Thêm khóa học”</p> <p>4. Hệ thống hiển thị form thêm khóa học gồm các trường:</p> <ul style="list-style-type: none"> <li>- Tên khóa học</li> <li>- Slug</li> <li>- Giá</li> <li>- Cấp độ</li> <li>- Trạng thái (Đã xuất bản / Nháp / Đã ẩn)</li> <li>- Mô tả ngắn</li> <li>- Mô tả chi tiết</li> <li>- Ngôn ngữ lập trình</li> <li>- Tags</li> </ul> <p>5. Admin tiến hành điền đầy đủ những thông tin bắt buộc + Bấm “Xác nhận”</p> <p>6. Hệ thống hiển thị thông báo “Thêm khóa học thành công”</p>
Ngoại lệ	<p>6. Dữ liệu nhập thiếu hoặc không hợp lệ</p> <p>6.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ”</p> <p>6.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận</p> <p>6.3. Hệ thống thông báo “Thêm khóa học thành công”</p>

Bảng 2.19 Kịch bản chuẩn chức năng tạo khóa học mới

- Kịch bản chức năng chỉnh sửa khóa học

Use case	Chỉnh sửa thông tin khóa học
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin chỉnh sửa khóa học thành công

Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý khóa học”</li> <li>2. Hệ thống hiển thị danh sách các khóa học có trong hệ thống với một số thông tin: tên khóa học, cấp độ, ngôn ngữ, trạng thái,... + Nút “Thêm khóa học”</li> <li>3. Admin bấm chọn “Chỉnh sửa khóa học” vào khóa học muốn chỉnh sửa</li> <li>4. Hệ thống hiển thị form chỉnh sửa khóa học và dữ liệu của khóa học gồm các trường: <ul style="list-style-type: none"> <li>- Tên khóa học</li> <li>- Slug</li> <li>- Giá</li> <li>- Cấp độ</li> <li>- Trạng thái (Đã xuất bản / Nháp / Đã ẩn)</li> <li>- Mô tả ngắn</li> <li>- Mô tả chi tiết</li> <li>- Ngôn ngữ lập trình</li> <li>- Tags</li> </ul> </li> <li>5. Admin tiến hành thay đổi một số thông tin về khóa học + Bấm “Xác nhận”</li> <li>6. Hệ thống hiển thị thông báo “Chỉnh sửa khóa học thành công”</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>6. Dữ liệu nhập thiếu hoặc không hợp lệ <ol style="list-style-type: none"> <li>6.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ”</li> <li>6.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận</li> <li>6.3. Hệ thống thông báo “Chỉnh sửa khóa học thành công”</li> </ol> </li> </ol>

Bảng 2.20 Kịch bản chuẩn chức năng chỉnh sửa khóa học

- Kịch bản chức năng xem chi tiết khóa học

Use case	Xem chi tiết khóa học
Actor	Admin

Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin xem chi tiết khóa học thành công
Kịch bản chính	<p>1. Admin truy cập trang “Quản lý khóa học” từ menu điều hướng.</p> <p>2. Hệ thống hiển thị danh sách các khóa học hiện có với đầy đủ các cột thông tin:</p> <ul style="list-style-type: none"> <li>- ID: 101, 102, 103...</li> <li>- Tiêu đề: Lập trình Fullstack với Next.js, Database Design, MongoDB Advanced...</li> <li>- Cấp độ: Trung bình, Nâng cao.</li> <li>- Ngôn ngữ: JavaScript, TypeScript, Python.</li> <li>- Thống kê: Số bài học (15), Số học viên (120).</li> <li>- Trạng thái: Đã xuất bản.</li> </ul>  <p>3. Admin nhấn chọn xem chi tiết khóa học “MongoDB Advanced Course”.</p> <p>4. Hệ thống hiển thị trang chi tiết khóa học với bộ dữ liệu mẫu đầy đủ:</p> <ul style="list-style-type: none"> <li>- Tên khóa học: MongoDB Advanced Course</li> <li>- Slug: mongodb-advanced-mastery-2024</li> <li>- Giá: 499.000 đ</li> <li>- Cấp độ: Nâng cao</li> <li>- Trạng thái: Đã xuất bản</li> <li>- Mô tả ngắn: Làm chủ cơ sở dữ liệu NoSQL từ cơ bản đến tối ưu hóa hiệu năng thực tế.</li> <li>- Mô tả chi tiết: Khóa học cung cấp kiến thức sâu về Sharding, Replication và Aggregation Framework trong MongoDB.</li> <li>- Ngôn ngữ lập trình: JavaScript, Node.js</li> </ul>

	<ul style="list-style-type: none"> <li>- Tags: #Database, #NoSQL, #Backend, #Optimization</li> <li>- Danh sách bài học: Hiện tại chưa có bài học nào (Hiển thị biểu tượng trống).</li> </ul> <p>5. Admin bấm chọn nút “+ Thêm bài học có sẵn”.</p> <p>6. Hệ thống hiển thị một cửa sổ (Popup) chứa danh sách các bài học để có trên hệ thống để Admin lựa chọn:</p> <ul style="list-style-type: none"> <li>- Bài học 01: Tổng quan về NoSQL và Document Database.</li> <li>- Bài học 02: Cài đặt và cấu hình Cluster trên MongoDB Atlas.</li> <li>- Bài học 03: Phương pháp tối ưu hóa Index cho Query.</li> </ul> <p>7. Admin thực hiện các thao tác:</p> <ul style="list-style-type: none"> <li>- Tích chọn Bài học 01 và Bài học 03.</li> <li>- Bấm nút “Xác nhận”.</li> </ul> <p>8. Hệ thống thực hiện các phản hồi sau:</p> <ul style="list-style-type: none"> <li>- Hiển thị thông báo: “Thêm 02 bài học vào khóa học thành công!”</li> <li>- Tự động cập nhật danh sách bài học tại trang chi tiết gồm: <ul style="list-style-type: none"> <li>+ STT 1: Tổng quan về NoSQL và Document Database.</li> <li>+ STT 2: Phương pháp tối ưu hóa Index cho Query.</li> </ul> </li> <li>- Cập nhật số lượng bài học ở phần thống kê tổng quan từ 0 lên 2.</li> </ul>
Ngoại lệ	<p>2. Không có khóa học nào</p> <p>2.1. Hệ thống hiển thị danh sách rỗng</p> <p>2.2. Admin tiến hành thêm khóa học mới</p>

Bảng 2.21 Kịch bản chuẩn chức năng xem chi tiết khóa học

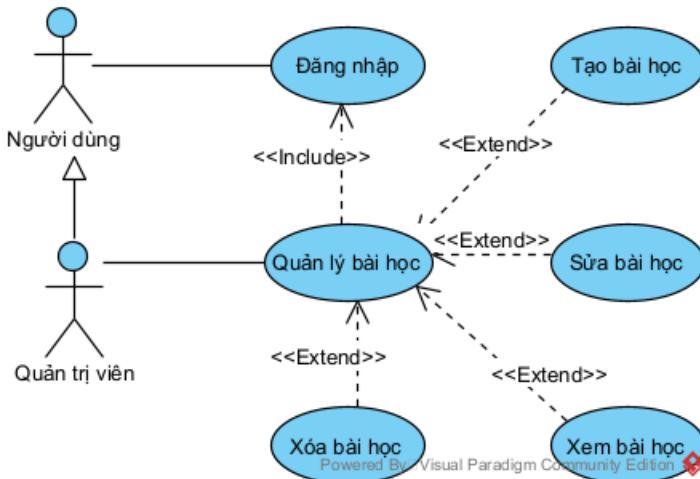
- Kịch bản chức năng xóa khóa học

Use case	Xóa khóa học
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin

Hậu điều kiện	Admin chỉnh xóa thông khóa học thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý khóa học”</li> <li>2. Hệ thống hiển thị danh sách các khóa học có trong hệ thống với một số thông tin: tên khóa học, cấp độ, ngôn ngữ, trạng thái,...</li> <li>3. Admin bấm chọn xóa 1 khóa học trong danh sách</li> <li>4. Hệ thống hiển thị 1 popup chứa thông tin “Bạn có chắc chắn muốn xóa khóa học này?” + nút “Xác nhận”</li> <li>5. Admin bấm nút “Xác nhận”</li> <li>6. Hệ thống hiển thị thông báo: “Xóa khóa học thành công”</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>6. Xóa khóa học thất bại do còn học viên             <ol style="list-style-type: none"> <li>6.1. Hệ thống hiển thị thông báo admin không thể xóa khóa học do còn học viên</li> <li>6.2. Admin bấm Xác nhận, chọn xóa khóa học khác không có học viên</li> </ol> </li> </ol> <p>Hệ thống hiển thị thông báo “Xóa khóa học thành công”</p>

Bảng 2.22 Kịch bản chuẩn chức năng xóa khóa học

### c. Quản lý bài học



Hình 2.18 UC chi tiết chức năng quản lý bài học

- Kịch bản chức năng xem chi tiết bài học

Use case	Xem chi tiết bài học
----------	----------------------

Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xem được danh sách và chi tiết bài học
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý khóa học</li> <li>6. Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Quản lý khóa học, trong đó có chức năng “Danh sách bài học”</li> <li>7. Admin chọn chức năng Danh sách bài học</li> <li>8. Hệ thống hiển thi trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút thêm bài học, table chứa danh sách bài học, mỗi bài học hiển thị tên, ngôn ngữ, khóa học đính kèm và các nút xem, sửa, xóa.</li> <li>9. Admin chọn nút xem bài học</li> <li>10. Hệ thống hiển thi trang chi tiết bài học gồm tên bài học, tên khóa học, ngôn ngữ, danh sách tài nguyên bài học</li> </ol>
Ngoại lệ	<p>8. Hệ thống không tìm thấy bài học</p> <p>8.1 Hiển thị table trống cùng ghi chú không tìm thấy bài học nào</p>

Bảng 2.23 Kịch bản chuẩn chức năng xem chi tiết bài học

- Kịch bản chức năng thêm bài học

Use case	Thêm bài học
----------	--------------

Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin thêm được bài học
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý khóa học</li> <li>6. Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Quản lý khóa học, trong đó có chức năng “Danh sách bài học”</li> <li>7. Admin chọn chức năng Danh sách bài học</li> <li>8. Hệ thống hiển thi trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút thêm bài học, table chứa danh sách bài học, mỗi bài học hiển thị tên, ngôn ngữ, khóa học đính kèm và các nút xem, sửa, xóa.</li> <li>9. Admin chọn nút thêm bài học</li> <li>10. Hệ thống hiển thi trang thêm bài học gồm các ô nhập thông tin cho tên bài học, khóa học, mô tả, nút thêm tài nguyên bài học, 2 nút “Tạo bài học” và “Hủy”</li> <li>11. Admin nhập các thông tin tên bài học, khóa học, mô tả bài học, thứ tự bài học và click vào nút thêm tài nguyên bài học</li> <li>12. Hệ thống hiển thi thêm form ở ngay dưới nút thêm bài học gồm các ô nhập thông tin tên tài nguyên bài học, loại tài nguyên, và nội dung tài nguyên tùy theo từng loại</li> </ol>

	<p>13. Admin nhập đầy đủ thông tin tên, loại, nội dung cho tài nguyên bài học và ấn nút “Tạo bài học”</p> <p>14. Hệ thống thông báo tạo bài học thành công và hiển thị trang chi tiết bài học vừa thêm cho admin</p>
Ngoại lệ	<p>13. Admin nhập thiếu thông tin cần thiết cho bài học</p> <p>13.1 Hệ thống báo lỗi thiếu thông tin</p> <p>13.2 Admin nhập lại đầy đủ thông tin</p> <p>Tiếp tục bước 14</p> <p>13. Admin chọn nút “Hủy”</p> <p>13.1 Hệ thống quay về trang danh sách bài học</p>

Bảng 2.24 Kịch bản chuẩn chức năng thêm bài học

- Kịch bản chức năng sửa bài học

Use case	Sửa bài học
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin sửa được bài học
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>Admin chọn Quản trị</li> <li>Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>Quản trị tìm và chọn chức năng Quản lý khóa học</li> </ol>

	<p>6. Hệ thống hiển thị dropdown gồm các chức năng nhỏ trong Quản lý khóa học, trong đó có chức năng “Danh sách bài học”</p> <p>7. Admin chọn chức năng Danh sách bài học</p> <p>8. Hệ thống hiển thị trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút thêm bài học, table chứa danh sách bài học, mỗi bài học hiển thị tên, ngôn ngữ, khóa học đính kèm và các nút xem, sửa, xóa.</p> <p>9. Admin chọn nút sửa bài học ở một bài học muốn sửa</p> <p>10. Hệ thống hiển thị trang sửa bài học gồm các ô nhập thông tin cho tên bài học, khóa học, mô tả, nút thêm tài nguyên bài học, 2 nút “Lưu” và “Hủy”</p> <p>11. Admin sửa các thông tin tên bài học, khóa học, mô tả bài học và các thông tin của tài nguyên bài học, thêm, xóa tài nguyên bài học và ấn nút “Lưu”</p> <p>12. Hệ thống thông báo tạo bài học thành công và hiển thị trang chi tiết bài học vừa sửa cho admin</p>
Ngoại lệ	<p>11. Admin nhập thiếu thông tin cần thiết cho bài học</p> <p>11.1 Hệ thống báo lỗi thiếu thông tin</p> <p>11.2 Admin nhập lại đầy đủ thông tin</p> <p>Tiếp tục bước 12</p> <p>11. Admin chọn nút “Hủy”</p> <p>11.1 Hệ thống quay về trang danh sách bài học</p>

Bảng 2.25 Kịch bản chuẩn chức năng sửa bài học

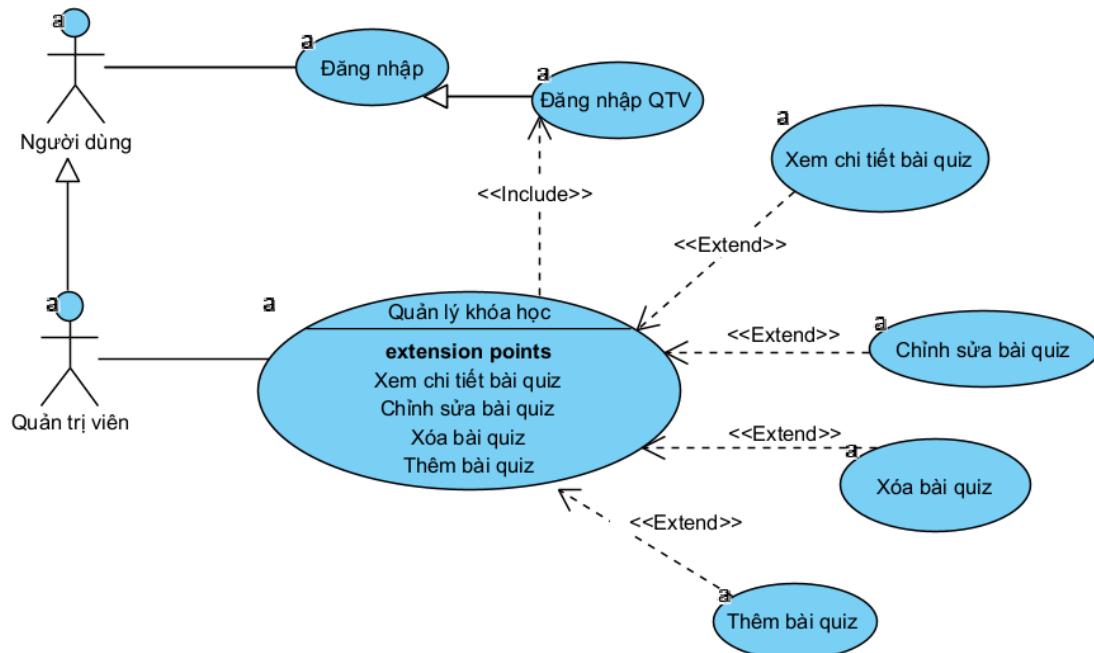
- Kịch bản chức năng xóa bài học

Use case	Xóa bài học
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công

Hậu điều kiện	Admin xóa được bài học
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý khóa học</li> <li>6. Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Quản lý khóa học, trong đó có chức năng “Danh sách bài học”</li> <li>7. Admin chọn chức năng Danh sách bài học</li> <li>8. Hệ thống hiển thi trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút thêm bài học, table chứa danh sách bài học, mỗi bài học hiển thị tên, ngôn ngữ, khóa học đính kèm và các nút xem, sửa, xóa.</li> <li>9. Admin chọn nút xóa bài học ở một bài học muốn sửa</li> <li>10. Hệ thống hiển thi thông báo xác nhận xóa bài học gồm 2 nút “Xóa” và “Hủy”</li> <li>11. Admin chọn “Xóa”</li> <li>12. Hệ thống thông báo xóa bài học mà admin đã chọn và thông báo xóa thành công, quay trở lại giao diện danh sách bài học trong đó không còn bài học mà admin vừa xác nhận xóa</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>11. Admin chọn “Hủy”</li> <li>11.1. Hệ thống quay trở về giao diện danh sách bài học trong đó vẫn còn bài học mà admin vừa chọn</li> </ol>

Bảng 2.26 Kịch bản chuẩn chức năng xóa bài học

*d. Quản lý bài quiz*



Hình 2.19 Biểu đồ UC quản lý bài Quiz

- Kịch bản chức năng xem bài quiz

Use case	Xem chi tiết bài quiz
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin xem được thông tin chi tiết của bài quiz
Kịch bản chính	<ol style="list-style-type: none"> <li>Admin truy cập trang “Quản lý Quiz”</li> <li>Hệ thống hiển thị danh sách người dùng có trong hệ thống với một số thông tin: tên, số lượng câu hỏi, trạng thái,...</li> <li>Admin bấm chọn xem chi tiết</li> <li>Trang chi tiết hiển thị các thông tin chính về bài quiz bao gồm: Tên quiz, mô tả, trạng thái, thời gian làm bài, điểm tối đa, giới hạn số lần làm + Danh sách các câu hỏi trong bài Quiz:</li> </ol>

	<ul style="list-style-type: none"> <li>- Nội dung câu hỏi</li> <li>- Loại câu hỏi (1 đáp án / nhiều đáp án)</li> <li>- Danh sách lựa chọn</li> <li>- Đáp án đúng</li> <li>- Điểm từng câu</li> </ul>
Ngoại lệ	<p>2. Danh sách bài quiz rỗng</p> <p>2.1. Hệ thống hiển thị danh sách rỗng về các bài quiz</p> <p>2.2. Admin bấm “Thêm bài quiz mới”</p>

Bảng 2.27 Kịch bản chuẩn chức năng xem bài quiz

- Kịch bản chức năng tạo bài quiz

Use case	Thêm bài quiz mới
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin thêm bài quiz thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý Quiz”</li> <li>2. Hệ thống hiển thị danh sách người dùng có trong hệ thống với một số thông tin: tên, số lượng câu hỏi, trạng thái,... + Nút “Thêm bài Quiz”</li> <li>3. Admin bấm chọn “Thêm bài Quiz”</li> <li>4. Hệ thống hiển thị form thêm bài quiz gồm: <ul style="list-style-type: none"> <li>- Các trường thông tin Quiz</li> <li>- Danh sách câu hỏi</li> <li>- Nút “Thêm câu hỏi mới”</li> <li>- Nút “Lưu”</li> </ul> </li> <li>5. Admin tiến hành nhập thông tin về quiz: <ul style="list-style-type: none"> <li>- Tên quiz</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>- Mô tả</li> <li>- Thời gian làm bài</li> <li>- Điểm tối đa</li> <li>- Shuffle câu hỏi / đáp án</li> <li>- Giới hạn số lần làm</li> <li>- Trạng thái (nháp / xuất bản)</li> </ul> <p>Vào phần danh sách câu hỏi bấm thêm câu hỏi mới</p> <ol style="list-style-type: none"> <li>6. Hệ thống hiển thị 1 form để điền các thông tin về câu hỏi:</li> <ul style="list-style-type: none"> <li>- Nội dung câu hỏi</li> <li>- Loại câu hỏi (1 lựa chọn / nhiều lựa chọn)</li> <li>- Danh sách lựa chọn (A, B, C, D...)</li> <li>- Đáp án đúng</li> <li>- Điểm số cho câu hỏi</li> <li>- Giải thích đáp án</li> </ul> <li>7. Admin tiến hành điền các thông tin cho câu hỏi, hoàn tất các bước sau đó bấm “Lưu”</li> <li>8. Hệ thống hiển thị thông báo “Thêm bài Quiz thành công”</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>8. Dữ liệu nhập thiếu hoặc không hợp lệ           <ol style="list-style-type: none"> <li>8.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ”</li> <li>8.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận</li> <li>8.3. Hệ thống thông báo “Thêm bài quiz thành công”</li> </ol> </li> </ol>

Bảng 2.28 Kịch bản chuẩn chức năng tạo bài quiz

- Kịch bản chức năng sửa bài quiz

Use case	Chỉnh sửa bài quiz
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin

Hậu điều kiện	Admin thêm bài quiz thành công
Kịch bản chính	<p>1. Admin truy cập trang “Quản lý Quiz”</p> <p>2. Hệ thống hiển thị danh sách bài Quiz có trong hệ thống với một số thông tin: tên, số lượng câu hỏi, trạng thái,...</p> <p>3. Admin bấm chọn “Chỉnh sửa bài Quiz” với bài quiz muốn thay đổi</p> <p>4. Hệ thống hiển thị form chỉnh sửa bài quiz gồm:</p> <ul style="list-style-type: none"> <li>- Các trường thông tin Quiz</li> <li>- Danh sách câu hỏi</li> <li>- Nút “Thêm câu hỏi mới”</li> <li>- Nút “Lưu”</li> </ul> <p>5. Admin tiến hành thay đổi một số thông tin về quiz:</p> <ul style="list-style-type: none"> <li>- Tên quiz</li> <li>- Mô tả</li> <li>- Thời gian làm bài</li> <li>- Điểm tối đa</li> <li>- Shuffle câu hỏi / đáp án</li> <li>- Giới hạn số lần làm</li> <li>- Trạng thái (nhập / xuất bản)</li> </ul> <p>Trong danh sách câu hỏi có các lục thao tác: thêm, sửa, xóa. Người dùng muốn thay đổi câu hỏi bấm vào chỉnh sửa</p> <p>6. Hệ thống hiển thị 1 form để chỉnh sửa các thông tin về câu hỏi:</p> <ul style="list-style-type: none"> <li>- Nội dung câu hỏi</li> <li>- Loại câu hỏi (1 lựa chọn / nhiều lựa chọn)</li> <li>- Danh sách lựa chọn (A, B, C, D...)</li> <li>- Đáp án đúng</li> <li>- Điểm số cho câu hỏi</li> <li>- Giải thích đáp án</li> </ul> <p>7. Admin tiến hành điền các thông tin muốn chỉnh sửa cho câu hỏi, hoàn tất các bước sau đó bấm “Lưu”</p>

	8. Hệ thống hiển thị thông báo “Chỉnh sửa bài Quiz thành công”
Ngoại lệ	8. Dữ liệu nhập thiếu hoặc không hợp lệ 8.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ” 8.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận 8.3. Hệ thống thông báo “Chỉnh sửa bài quiz thành công”

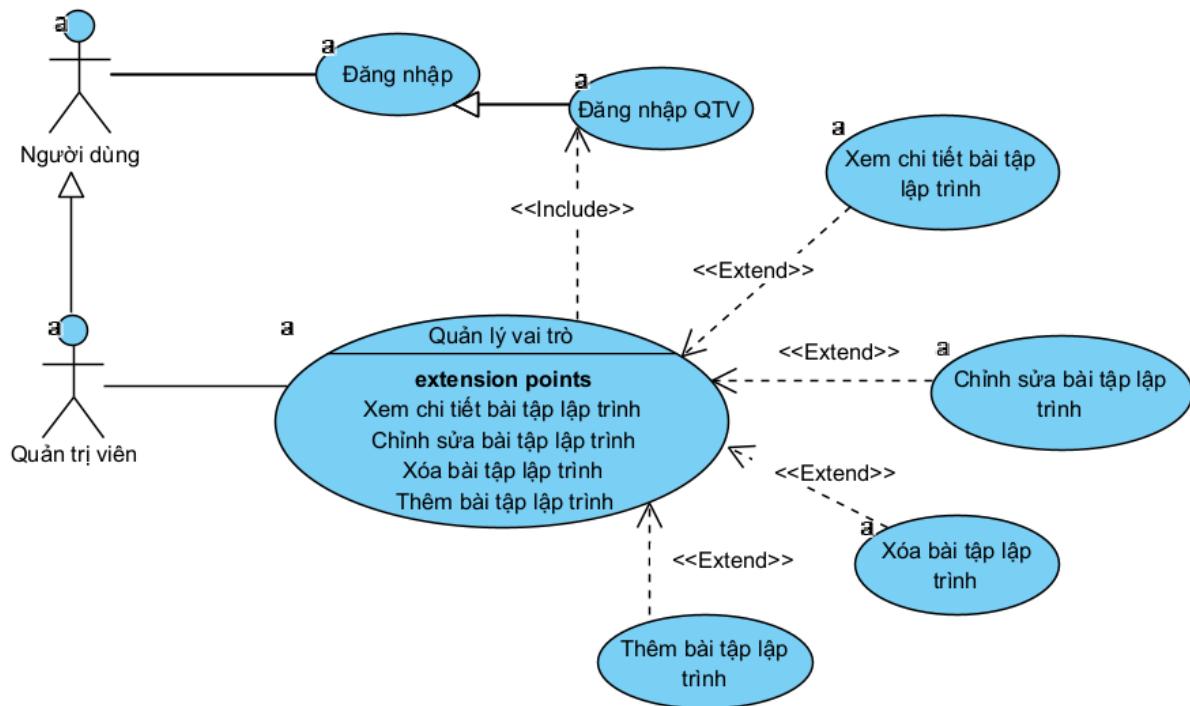
Bảng 2.29 Kịch bản chuẩn chức năng sửa bài quiz

- Kịch bản chức năng xóa bài quiz

Use case	Xóa bài Quiz
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin xóa bài Quiz thành công
Kịch bản chính	1. Admin truy cập trang “Quản lý bài Quiz” 2. Hệ thống hiển thị danh sách bài Quiz có trong hệ thống với một số thông tin: tên, số lượng câu hỏi, trạng thái,... 3. Admin bấm chọn xóa 1 bài quiz trong danh sách 4. Hệ thống hiển thị 1 popup chứa thông tin “Bạn có chắc chắn muốn xóa khóa học này?” + nút “Xác nhận” 5. Admin bấm nút “Xác nhận” 6. Hệ thống hiển thị thông báo: “Xóa bài Quiz thành công”
Ngoại lệ	4. Admin hủy xóa 4.1. Admin chọn “Hủy xóa” 4.2. Hệ thống hiển thị danh sách các bài Quiz

Bảng 2.30 Kịch bản chuẩn chức năng xóa bài quiz

#### e. Quản lý bài tập lập trình



Hình 2.20 Biểu đồ UC quản lý bài tập lập trình

- Xem chi tiết bài tập lập trình (problem)

Use case	Xem chi tiết bài lập trình
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin xem chi tiết bài lập trình
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý bài lập trình”</li> <li>2. Hệ thống hiển thị danh sách các bài lập trình có trong hệ thống với một số thông tin: slug, tiêu đề, độ khó, số lượng test case, trạng thái, trạng thái domjudge</li> <li>3. Admin bấm chọn xem chi tiết 1 bài lập trình</li> <li>4. Hệ thống hiển thị trang chi tiết của khóa học gồm các thông tin chính:</li> </ol>

	<ul style="list-style-type: none"> <li>- Các thông tin cơ bản: tiêu đề, độ khó, trạng thái, mô tả ngắn, đề bài</li> <li>- Giới hạn về bộ nhớ và thời gian khi nộp</li> <li>- Thông tin các test case: điểm, trạng thái,...</li> <li>- tag, ngôn ngữ hỗ trợ</li> <li>- Phần nộp bài để test thử: hỗ trợ 2 mode: nhập code trực tiếp, upload file từ máy</li> <li>- Lịch sử nộp bài</li> </ul> <p>5. Admin tiến hành nhập code trực tiếp để tiến hành test + Bấm “Nộp bài”</p> <p>6. Hệ thống sẽ gửi dữ liệu nộp của user đến DOMjudge để tiến hành chấm bài</p> <p>7. DOMjudge chấm bài dựa trên các test case và trả về kết quả cho hệ thống</p> <p>8. Hệ thống hiển thị lịch sử nộp bài của admin với trạng thái kết quả là AC.</p>
Ngoại lệ	<p>2. Không có bài lập trình nào</p> <p>2.1. Hệ thống hiển thị danh sách rỗng</p> <p>2.2. Admin tiến hành thêm bài lập trình mới</p> <p>7. Hệ thống kết nối với DOMjudge thất bại</p> <p>7.1. Hệ thống không thể kết nối được với DOMjudge</p> <p>7.2. Admin bấm nút “sync” để gửi lại đến DOMjudge tiến hành chấm bài</p> <p>7.3. DOMjudge chấm bài và trả về kết quả</p>

Bảng 2.31 Kịch bản chuẩn chức năng bài tập lập trình (problem)

- Tạo bài lập trình

Use case	Thêm bài lập trình
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin

Hậu điều kiện	Admin thêm bài lập trình thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý bài lập trình”</li> <li>2. Hệ thống hiển thị danh sách các bài lập trình có trong hệ thống với một số thông tin: slug, tiêu đề, độ khó, số lượng test case, trạng thái, trạng thái domjudge + nút “Thêm bài lập trình”</li> <li>3. Admin bấm chọn “Thêm bài lập trình”</li> <li>4. Hệ thống hiển thị form thêm bài quiz gồm: <ul style="list-style-type: none"> <li>- Các trường thông tin của bài lập trình</li> <li>- Danh sách các test case</li> <li>- Nút “Thêm mới test case” (Hỗ trợ 2 mode: thêm trực tiếp, upload file zip)</li> <li>- Nút “Lưu”</li> </ul> </li> <li>5. Admin tiến hành nhập thông tin về bài lập trình: <ul style="list-style-type: none"> <li>- Slug</li> <li>- Tiêu đề</li> <li>- Độ khó</li> <li>- Mô tả ngắn</li> <li>- nguồn</li> <li>- Đề bài: nội dung đề bài, định dạng đầu vào, định dạng đầu ra</li> </ul> Vào phần danh sách test case bấm test case mới </li> <li>6. Hệ thống hiển thị 1 form để điền các thông tin về câu hỏi: <ul style="list-style-type: none"> <li>- Loại test case (Công khai / Ân)</li> <li>- Input</li> <li>- Output</li> </ul> </li> <li>7. Admin tiến hành điền các thông tin cho câu hỏi, hoàn tất các bước sau đó bấm “Lưu”</li> <li>8. Hệ thống tiến hành lưu vào DB và gửi dữ liệu cho DOMjudge</li> <li>9. DOMjudge tiến hành lưu thông tin của bài lập trình và các test case vào DB của nó phục vụ cho việc chấm bài, trả về lưu thành công</li> <li>10. Hệ thống gửi thông báo “Lưu bài lập trình thành công”</li> </ol>

Ngoại lệ	<p>8. Dữ liệu nhập thiếu hoặc không hợp lệ</p> <p>8.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ”</p> <p>8.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận</p> <p>9. Lỗi khi gửi dữ liệu đến DOMjudge</p> <p>9.1. DOMjudge không kết nhận được dữ liệu được gửi đến</p> <p>9.2. Hệ thống thông báo “Bài lập trình đã lưu nhưng không thể đồng bộ với DOMjudge. Vui lòng thử lại.</p>
----------	---

Bảng 2.32 Kịch bản chuẩn chức năng tạo bài lập trình

- Chính sửa bài lập trình

Use case	Chỉnh sửa bài lập trình
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin chỉnh sửa bài lập trình thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Admin truy cập trang “Quản lý bài lập trình”.</li> <li>2. Hệ thống hiển thị danh sách các bài lập trình với các thông tin: slug, tiêu đề, độ khó, số lượng test case, trạng thái, trạng thái DOMjudge, nút Chính sửa.</li> <li>3. Admin bấm “Chỉnh sửa” bên cạnh bài lập trình muốn chỉnh sửa.</li> <li>4. Hệ thống hiển thị form chỉnh sửa bài lập trình: <ul style="list-style-type: none"> <li>• Các trường thông tin cơ bản: slug, tiêu đề, độ khó, mô tả ngắn, nguồn, đề bài (nội dung, định dạng đầu vào/ra).</li> <li>• Danh sách test case hiện có.</li> <li>• Nút Thêm mới test case, hỗ trợ 2 mode: thêm trực tiếp, upload file zip.</li> <li>• Nút Lưu.</li> </ul> </li> </ol>

	<p>5. Admin tiến hành chỉnh sửa thông tin bài lập trình và/hoặc test case:</p> <ul style="list-style-type: none"> <li>Chỉnh sửa thông tin cơ bản (slug, tiêu đề, độ khó, mô tả...).</li> <li>Thêm, sửa hoặc xóa test case.</li> </ul> <p>Admin bấm “Lưu” sau khi chỉnh sửa xong.</p> <p>6. Hệ thống cập nhật thông tin bài lập trình và các test case vào cơ sở dữ liệu của hệ thống. Hệ thống gửi dữ liệu cập nhật cho DOMjudge để đồng bộ.</p> <p>7. DOMjudge lưu thông tin mới và trả về trạng thái thành công.</p> <p>8. Hệ thống hiển thị thông báo “Cập nhật bài lập trình thành công”.</p>
Ngoại lệ	<p>6. Dữ liệu nhập thiếu hoặc không hợp lệ</p> <p>6.1. Hệ thống thông báo “Thất bại do nhập thiếu hoặc không hợp lệ”</p> <p>6.2. Admin nhập đầy đủ các thông tin và bấm Xác nhận</p> <p>9. Lỗi khi gửi dữ liệu đến DOMjudge</p> <p>9.1. DOMjudge không kết nhận được dữ liệu được gửi đến</p> <p>9.2. Hệ thống thông báo “Bài lập trình đã lưu nhưng không thể đồng bộ với DOMjudge. Vui lòng thử lại.</p>

Bảng 2.33 Kịch bản chuẩn chúc năng chỉnh sửa bài lập trình

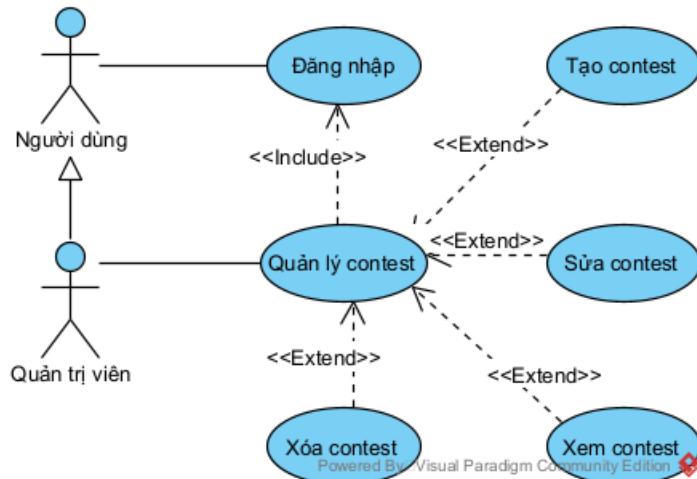
- Xóa bài lập trình

Use case	Xóa bài lập trình
Actor	Admin
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin
Hậu điều kiện	Admin xóa bài lập trình thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>Admin truy cập trang “Quản lý bài lập trình”</li> <li>Hệ thống hiển thị danh sách các bài lập trình với các thông tin: slug, tiêu đề, độ khó, số lượng test case, trạng thái, trạng thái DOMjudge, nút xóa.</li> </ol>

	<ol style="list-style-type: none"> <li>3. Admin bấm chọn xóa 1 bài lập trình trong danh sách</li> <li>4. Hệ thống hiển thị 1 popup chứa thông tin “Bạn có chắc chắn muốn xóa khóa học này?” + nút “Xác nhận”</li> <li>5. Admin bấm nút “Xác nhận”</li> <li>6. Hệ thống gửi yêu cầu xóa cho DOMjudge</li> <li>7. DOMjudge xóa các dữ liệu liên quan đến bài lập trình và phản hồi kết quả</li> <li>8. Hệ thống hiển thị thông báo: “Xóa bài lập trình thành công”</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>4. Admin hủy xóa             <ol style="list-style-type: none"> <li>4.1. Admin chọn “Hủy xóa”</li> <li>4.2. Hệ thống hiển thị danh sách các bài Quiz</li> </ol> </li> <li>7. Không kết nối được với DOMjudge             <ol style="list-style-type: none"> <li>7.1. Dữ liệu không được gửi đến DOMjudge</li> <li>7.2. Hệ thống gửi thông báo cho người dùng “Xóa thất bại”</li> </ol> </li> </ol>

Bảng 2.34 Kịch bản chuẩn chức năng xóa bài lập trình

#### f. Quản lý contest



Hình 2.21 UC chi tiết chức năng quản lý contest

- Kịch bản chức năng xem chi tiết contest

Use case	Xem chi tiết contest
----------	----------------------

Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xem được danh sách và chi tiết contest
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý contest</li> <li>6. Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Quản lý contest, trong đó có chức năng “Danh sách contest”</li> <li>7. Admin chọn chức năng Danh sách contest</li> <li>8. Hệ thống hiển thi trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút tạo contest, table chứa danh sách contest, mỗi contest hiển thị tên, mã, thời gian, trạng thái, số lượng problem và các nút xem, sửa, xóa.</li> <li>9. Admin chọn nút xem contest ở 1 contest muốn xem</li> <li>10. Hệ thống hiển thi trang chi tiết contest gồm tên contest, mã contest, mô tả, thời gian bắt đầu, kết thúc, số problem, các settings tính điểm cho contest, thời gian đóng băng bảng xếp hạng và table chứa danh sách problem trong contest</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>8. Hệ thống không tìm thấy contest phù hợp             <ol style="list-style-type: none"> <li>8.1 Hệ thống hiển thị ghi chú không tìm thấy contest cho admin</li> </ol> </li> </ol>

Bảng 2.35 Kịch bản chuẩn chức năng xem chi tiết contest

- Kịch bản chức năng tạo contest

Use case	Tạo contest
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin tạo được contest
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý contest</li> <li>6. Hệ thống hiển thị dropdown gồm các chức năng nhỏ trong Quản lý contest, trong đó có chức năng “Danh sách contest”</li> <li>7. Admin chọn chức năng Danh sách contest</li> <li>8. Hệ thống hiển thị trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút tạo contest, table chứa danh sách contest, mỗi contest hiển thị tên, mã, thời gian, trạng thái, số lượng problem và các nút xem, sửa, xóa.</li> <li>9. Admin chọn nút tạo contest</li> <li>10. Hệ thống hiển thị trang tạo contest gồm các ô nhập thông tin tên contest, mã contest, mô tả, thời gian bắt đầu, kết thúc, số problem, các settings tính điểm cho contest, thời gian đóng bảng bảng xếp hạng và 2 nút “Tạo contest” và “Hủy”</li> <li>11. Admin nhập đầy đủ các thông tin contest và chọn “tạo contest”</li> </ol>

	12. Hệ thống lưu contest và thông báo tạo contest thành công, chuyển đến trang chi tiết contest vừa tạo cho admin
Ngoại lệ	11. Admin chọn “Hủy” 11.1 Hệ thống quay trở về trang danh sách contest

Bảng 2.36 Kịch bản chuẩn chức năng tạo contest

- Kịch bản chức năng chỉnh sửa contest

Use case	Chỉnh sửa contest
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin chỉnh sửa được contest
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>Admin chọn Quản trị</li> <li>Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>Quản trị tìm và chọn chức năng Quản lý contest</li> <li>Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Quản lý contest, trong đó có chức năng “Danh sách contest”</li> <li>Admin chọn chức năng Danh sách contest</li> <li>Hệ thống hiển thi trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút tạo contest, table chứa danh sách contest, mỗi contest</li> </ol>

	<p>hiển thị tên, mã, thời gian, trạng thái, số lượng problem và các nút xem, sửa, xóa.</p> <p>9. Admin chọn nút sửa contest</p> <p>10. Hệ thống hiển thị trang sửa contest gồm các ô nhập thông tin tên contest, mã contest, mô tả, thời gian bắt đầu, kết thúc, số problem, các settings tính điểm cho contest, thời gian đóng băng bảng xếp hạng, danh sách problem cùng nút “Lưu” và “Hủy”</p> <p>11. Admin nhập đầy đủ các thông tin contest và chọn “Lưu”</p> <p>12. Hệ thống lưu contest và thông báo tạo contest thành công, chuyển đến trang chi tiết contest vừa sửa cho admin</p>
Ngoại lệ	<p>11. Admin chọn “Hủy”</p> <p>11.1 Hệ thống quay trở về trang danh sách contest</p>

Bảng 2.37 Kịch bản chuẩn chức năng chỉnh sửa contest

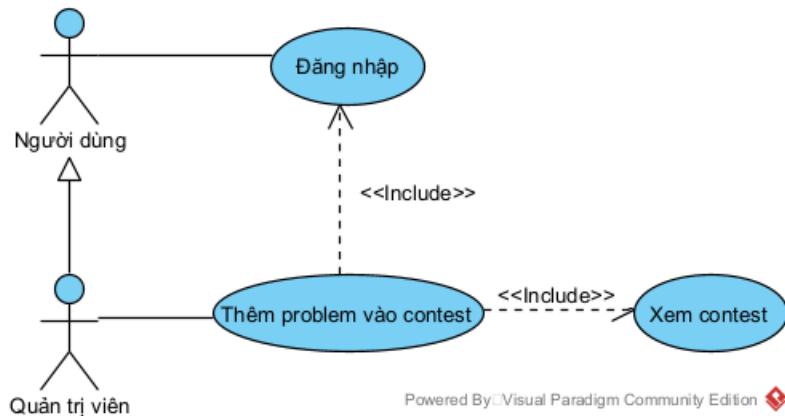
- Kịch bản chức năng xóa contest

Use case	Xóa contest
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xóa được contest
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>Admin chọn Quản trị</li> <li>Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> </ol>

	<p>5. Quản trị tìm và chọn chức năng Quản lý contest</p> <p>6. Hệ thống hiển thị dropdown gồm các chức năng nhỏ trong Quản lý contest, trong đó có chức năng “Danh sách contest”</p> <p>7. Admin chọn chức năng Danh sách contest</p> <p>8. Hệ thống hiển thị trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút tạo contest, table chứa danh sách contest, mỗi contest hiển thị tên, mã, thời gian, trạng thái, số lượng problem và các nút xem, sửa, xóa.</p> <p>9. Admin chọn nút xóa contest</p> <p>10. Hệ thống hiển thị thông báo xác nhận xóa contest và 2 nút “Xóa”, “Hủy” cho admin</p> <p>11. Admin chọn “Xóa”</p> <p>12. Hệ thống xóa contest và quay trở lại trang danh sách contest trong đó không còn contest vừa xóa</p>
Ngoại lệ	<p>11. Admin chọn “Hủy”</p> <p>11.1 Hệ thống quay trở về trang danh sách contest</p>

Bảng 2.38 Kịch bản chuẩn chức năng xoá contest

#### g. Thêm problem vào contest



Hình 2.22 UC chi tiết chức năng thêm problem vào contest

- Kịch bản chính chức năng thêm problem vào contest

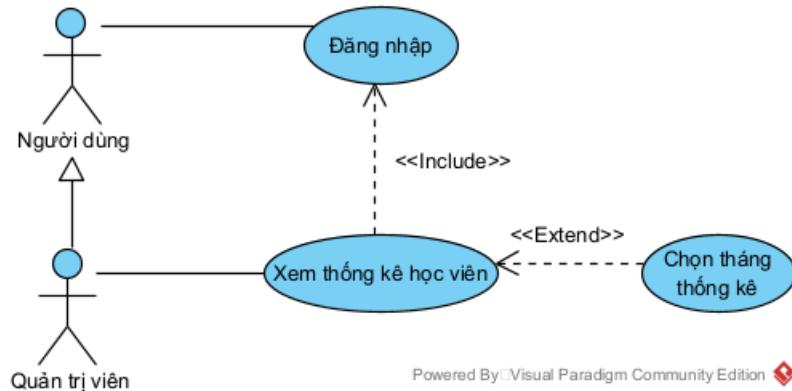
Use case	Thêm problem vào contest
----------	--------------------------

Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin thêm được problem vào contest
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý contest</li> <li>6. Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Quản lý contest, trong đó có chức năng “Danh sách contest”</li> <li>7. Admin chọn chức năng Danh sách contest</li> <li>8. Hệ thống hiển thi trang danh sách bài học cho Admin gồm bộ lọc, tìm kiếm, nút tạo contest, table chứa danh sách contest, mỗi contest hiển thị tên, mã, thời gian, trạng thái, số lượng problem và các nút xem, sửa, xóa.</li> <li>9. Admin chọn nút xem contest ở 1 contest muốn xem</li> <li>10. Hệ thống hiển thi trang chi tiết contest gồm tên contest, mã contest, mô tả, thời gian bắt đầu, kết thúc, số problem, các settings tính điểm cho contest, thời gian đóng băng bảng xếp hạng và table chứa danh sách problem cùng nút thêm problem trong contest</li> <li>11. Admin chọn nút thêm problem</li> <li>12. Hệ thống hiển thi form chứa các ô nhập thông tin cho 1 problem trong contest gồm selector problem, label, color, points và 2 nút “Thêm”, “Hủy”</li> </ol>

	<p>13. Admin nhập đầy đủ thông tin và chọn “Thêm”</p> <p>14. Hệ thống thêm problem vào danh sách problem trong trang chi tiết contest hiện tại cho admin và ẩn form thêm problem khỏi giao diện chi tiết contest</p>
Ngoại lệ	<p>13. Admin nhập thiếu thông tin cần thiết</p> <p>13.1 Hệ thống thông báo thiếu thông tin</p> <p>13.2 Admin nhập đầy đủ thông tin và chọn “Thêm”</p> <p>Tiếp tục bước 14</p> <p>13. Admin chọn “Hủy”</p> <p>13.1 Hệ thống ẩn form thêm problem khỏi giao diện chi tiết contest</p>

Bảng 2.39 Kịch bản chuẩn chức năng thêm problem vào contest

#### **h. Xem thông kê học viên**



Hình 2.23 UC chi tiết chức năng xem thông kê học viên

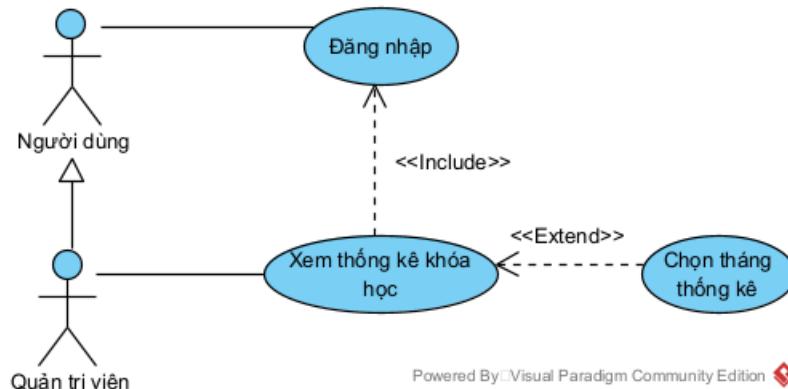
- Kịch bản chính chức năng xem thông kê học viên

Use case	Xem thông kê học viên
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xem được thông kê học viên

Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Thông kê</li> <li>6. Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Thông kê, trong đó có chức năng “Thông kê học viên”</li> <li>7. Admin chọn chức năng Thông kê học viên</li> <li>8. Hệ thống hiển thi dữ liệu thông kê với tháng hiện tại và ô chọn theo tháng</li> <li>9. Admin chọn tháng muốn xem</li> <li>10. Hệ thống hiển thi bảng thông kê, biểu đồ với dữ liệu theo các tùy chọn của admin</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>10. Dữ liệu thông kê không tồn tại           <ol style="list-style-type: none"> <li>10.1 Hệ thống trả về rỗng và ghi chú không có dữ liệu trong khoảng thời gian này hoặc biểu đồ trống.</li> </ol> </li> </ol>

Bảng 2.40 Kịch bản chuẩn chức năng xem thông kê học viên

#### i. Xem thông kê khóa học



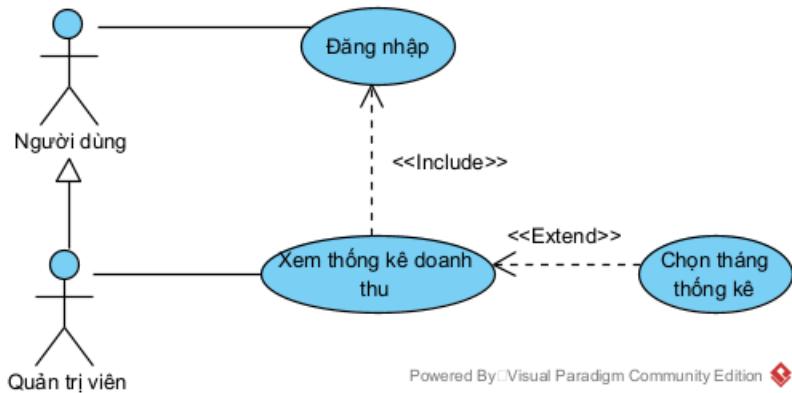
Hình 2.24 UC chi tiết chức năng xem thông kê khóa học

- Kịch bản chính chức năng xem thống kê khóa học

Use case	Xem thống kê khóa học
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xem được thống kê khóa học
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>Admin chọn Quản trị</li> <li>Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>Quản trị tìm và chọn chức năng Thống kê</li> <li>Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Thống kê, trong đó có chức năng “Thống kê khóa học”</li> <li>Admin chọn chức năng Thống kê khóa học</li> <li>Hệ thống hiển thi trang thống kê khóa học theo tháng hiện tại cùng ô chọn tháng thống kê</li> <li>Admin chọn tháng thống kê muốn xem</li> <li>Hệ thống hiển thi bảng thống kê, biểu đồ với dữ liệu theo các tùy chọn của admin</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>Dữ liệu thống kê không tồn tại             <ol style="list-style-type: none"> <li>Hệ thống trả về rỗng và ghi chú không có dữ liệu thống kê đã chọn hoặc biểu đồ trống.</li> </ol> </li> </ol>

Bảng 2.41 Kịch bản chuẩn chức năng xem thống kê khóa học

j. Xem báo cáo doanh thu



Hình 2.25 UC chi tiết chức năng xem báo cáo doanh thu

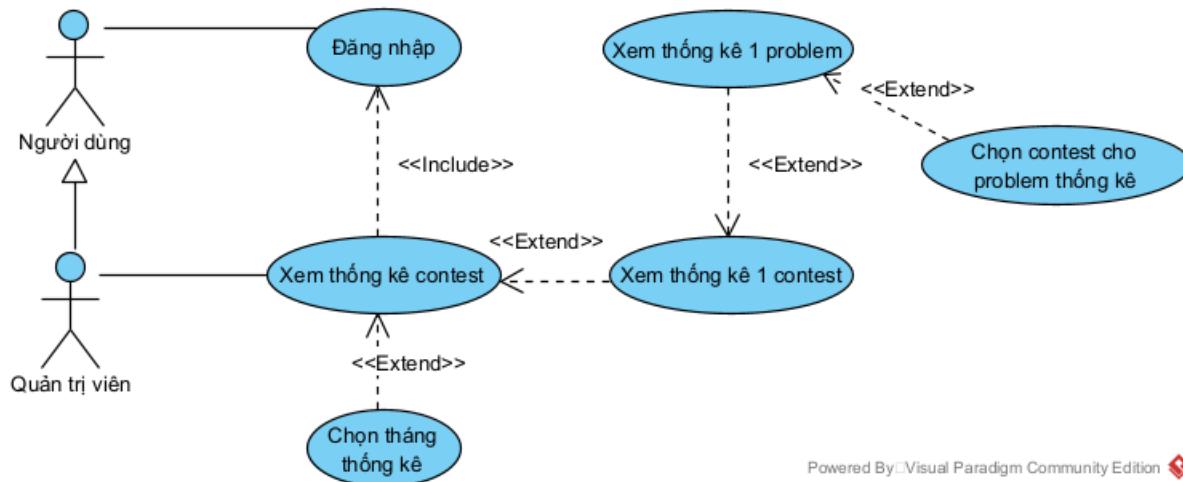
- Kịch bản chính chức năng xem báo cáo doanh thu

Use case	Xem báo cáo doanh thu
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xem được báo cáo doanh thu
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>Admin chọn Quản trị</li> <li>Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>Quản trị tìm và chọn chức năng Thống kê</li> <li>Hệ thống hiển thị dropdown gồm các chức năng nhỏ trong Thống kê, trong đó có chức năng “Báo cáo doanh thu”</li> <li>Admin chọn chức năng Báo cáo doanh thu</li> <li>Hệ thống hiển thị dữ liệu thống kê tổng và ô chọn tháng hoặc tổng.</li> <li>Admin chọn thời gian cần xem</li> </ol>

	10. Hệ thống hiển thị bảng thống kê, biểu đồ với dữ liệu theo các tùy chọn của admin
Ngoại lệ	<p>10. Dữ liệu thống kê không tồn tại</p> <p>10.1 Hệ thống trả về rỗng và ghi chú không có dữ liệu thống kê đã chọn hoặc biểu đồ trống.</p>

Bảng 2.42 Kịch bản chuẩn chức năng xem báo cáo doanh thu

#### k. Xem thống kê contest



Powered By: Visual Paradigm Community Edition

Hình 2.26 UC chi tiết chức năng xem thống kê contest

- Kịch bản chính chức năng xem thống kê contest

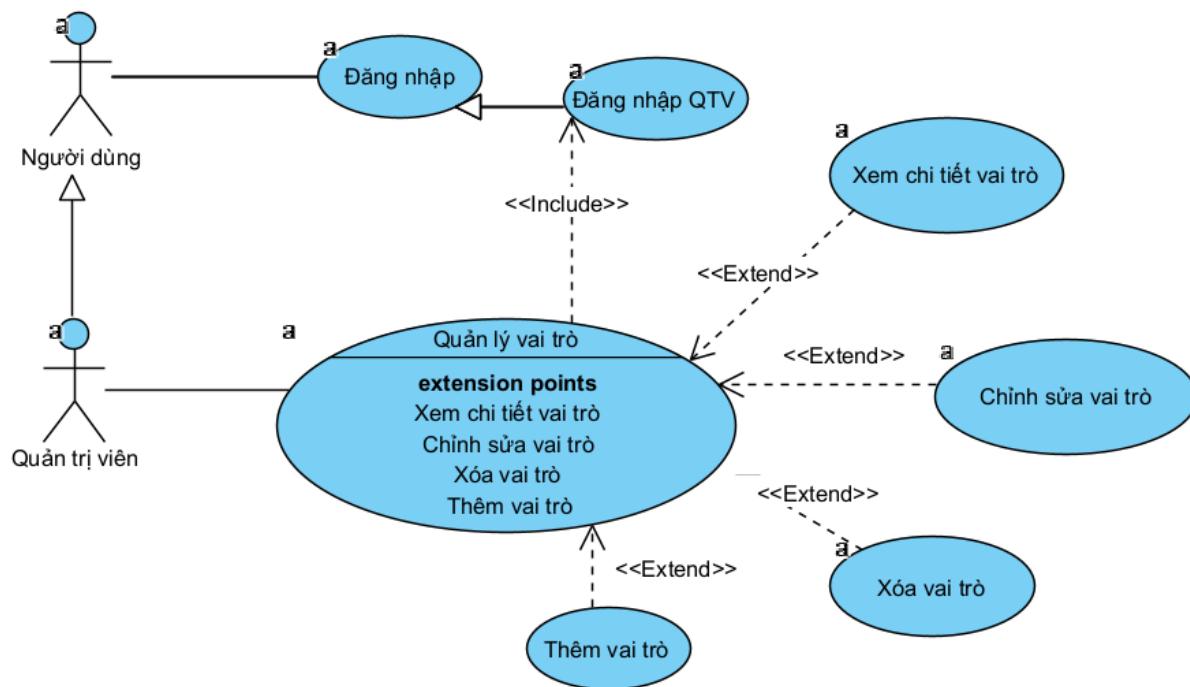
Use case	Xem thống kê contest
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công

Hậu điều kiện	Admin xem được thống kê contest
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Thông kê</li> <li>6. Hệ thống hiển thi dropdown gồm các chức năng nhỏ trong Thông kê, trong đó có chức năng “Thông kê contest”</li> <li>7. Admin chọn chức năng Thông kê contest</li> <li>8. Hệ thống hiển thị dữ liệu thống kê tổng và ô chọn theo tháng hoặc tổng.</li> <li>9. Admin chọn tháng muốn xem.</li> <li>10. Hệ thống hiển thị bảng thống kê, biểu đồ với dữ liệu số người đăng ký, danh sách contest, số người tham gia, tổng số bài nộp, số người hoàn thành, tổng thời gian trung bình để hoàn thành, tỷ lệ làm được số bài,...</li> <li>11. Admin chọn 1 contest trong danh sách contest</li> <li>12. Hệ thống hiển thị bảng thống kê, biểu đồ với dữ liệu số người đăng ký, danh sách problem, số người tham gia, tổng số bài nộp, số người hoàn thành, tổng thời gian trung bình để hoàn thành, tỷ lệ làm được bài, phân bố lỗi,...</li> <li>13. Admin chọn 1 problem trong danh sách problem</li> <li>14. Hệ thống hiển thị dữ liệu thống kê cho problem gồm tổng submissions, số bài AC, tỷ lệ AC, biểu đồ submissions theo ngày, phân bố trạng thái submissions,...</li> </ol>

Ngoại lệ	<p>10. Dữ liệu thống kê không tồn tại</p> <p>10.1 Hệ thống trả về rỗng và ghi chú không có dữ liệu thống kê hoặc biểu đồ trống.</p> <p>12. Dữ liệu thống kê không tồn tại</p> <p>12.1 Hệ thống trả về rỗng và ghi chú không có dữ liệu thống kê hoặc biểu đồ trống.</p> <p>14. Dữ liệu thống kê không tồn tại</p> <p>14.1 Hệ thống trả về rỗng và ghi chú không có dữ liệu thống kê hoặc biểu đồ trống.</p>
----------	---

Bảng 2.43 UC chi tiết chức năng xem thống kê contest

### I. Quản lý vai trò



Hình 2.27 Biểu đồ UC quản lý vai trò

- Kịch bản chức năng thêm vai trò

Use case	Thêm vai trò
Actor	Quản trị viên

Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin, ở giao diện quản trị
Hậu điều kiện	Quản trị viên thêm vai trò thành công
Kịch bản chính	<p>1. Admin truy cập vào mục “Quản lý quyền” (hoặc Quản lý vai trò) từ menu hệ thống.</p> <p>2. Hệ thống hiển thị trang Quản lý vai trò gồm danh sách các vai trò hiện có:</p> <ul style="list-style-type: none"> <li>- ID: 1, 2</li> <li>- Tên vai trò: admin, user</li> <li>- Mô tả: Quản trị viên, User with limited permissions</li> <li>- Số quyền: 104, 27</li> <li>- Số người dùng: 1, 201</li> <li>- Nút chức năng: + Tạo vai trò mới</li> </ul> <p>3. Admin nhấn nút “+ Tạo vai trò mới”.</p> <p>4. Hệ thống hiển thị trang “Tạo vai trò mới” với các khu vực nhập liệu:</p> <ul style="list-style-type: none"> <li>- Thông tin cơ bản: Ô nhập Tên vai trò, Ô nhập Mô tả.</li> <li>- Quyền: Danh sách các nhóm quyền (User Management, Content Management, Course Management, Problem Management,...) kèm checkbox để lựa chọn.</li> <li>- Nút bấm: Hủy, Tạo mới.</li> </ul> <p>5. Admin nhập các thông tin mẫu cho vai trò mới:</p> <ul style="list-style-type: none"> <li>- Tên vai trò: Content_Manager</li> <li>- Mô tả: Chịu trách nhiệm quản lý nội dung khóa học và bài tập, không có quyền quản lý hệ thống.</li> <li>- Chọn quyền: <ul style="list-style-type: none"> <li>+ Tích chọn nhóm Content Management (Read/View files, Update/Edit files,...).</li> </ul> </li> </ul>

	<p>+ Tích chọn nhóm Course Management (Read/View courses, Update/Edit courses).</p> <p>+ Tích chọn nhóm Problem Management (Read/View problems).</p> <p>Sau đó, Admin nhấn nút “Tạo mới”.</p> <p>6. Hệ thống thực hiện kiểm tra dữ liệu, lưu vai trò mới và phản hồi:</p> <ul style="list-style-type: none"> <li>- Hiển thị thông báo: “Tạo vai trò mới thành công!”</li> <li>- Tự động điều hướng Admin quay trở lại trang Quản lý vai trò.</li> </ul>
Ngoại lệ	<p>6. Tạo vai trò thất bại do nhập thiếu trường</p> <p>6.1. Hệ thống thông báo tạo thất bại do người dùng không nhập trường bắt buộc</p> <p>6.2. Người dùng nhập đầy đủ các trường và bấm tạo mới</p> <p>6.3. Hệ thống thông báo tạo thành công</p> <p>6. Tạo vai trò thất bại do tên vai trò đã tồn tại</p> <p>6.1. Hệ thống thông báo thất bại do tên vai trò đã tồn tại</p> <p>6.2. Người dùng nhập lại tên vai trò</p> <p>6.3. Hệ thống thông báo tạo thành công</p>

Bảng 2.44 Kịch bản chuẩn chức năng thêm vai trò

- Kịch bản chức năng sửa vai trò

Use case	Chỉnh sửa vai trò
Actor	Quản trị viên
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin, ở giao diện quản trị
Hậu điều kiện	Quản trị viên thêm vai trò thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Quản trị viên truy cập vào mục “Quản lý vai trò” từ menu hệ thống.</li> </ol>

2. Hệ thống hiển thị trang Quản lý vai trò gồm danh sách các vai trò hiện có:

- ID: 1, 2
- Tên vai trò: admin, user
- Mô tả: Quản trị viên, User with limited permissions
- Số quyền: 104, 27
- Số người dùng: 1, 201

ID	TÊN VAI TRÒ	MÔ TẢ	SỐ QUYỀN	SỐ NGƯỜI DÙNG	THAO TÁC
1	admin	Quản trị viên	104	1	
2	user	User with limited permissions	27	0	

3. Quản trị viên nhấn nút “Chỉnh sửa”.
4. Hệ thống hiển thị trang “Chỉnh sửa” với các trường dữ liệu:
- Thông tin cơ bản: Ô nhập Tên vai trò, Ô nhập Mô tả.
  - Quyền: Danh sách các nhóm quyền (User Management, Content Management, Course Management, Problem Management,...) kèm checkbox để lựa chọn.
  - Nút bấm: Hủy, Cập nhật.
5. Quản trị viên nhập các thông tin muốn cập nhật:
- Tên vai trò: Content\_Manager
  - Mô tả: Chịu trách nhiệm quản lý nội dung khóa học và bài tập, không có quyền quản lý hệ thống.
  - Chọn quyền:
    - + Tích chọn nhóm Content Management (Read/View files, Update/Edit files,...).
    - + Tích chọn nhóm Course Management (Read/View courses, Update/Edit courses).
    - + Tích chọn nhóm Problem Management (Read/View problems).
- Sau đó, Admin nhấn nút “Cập nhật”.
6. Hệ thống thực hiện kiểm tra dữ liệu, cập nhật vai trò và phản hồi:
- Hiển thị thông báo: “cập nhật vai trò mới thành công!”

	<ul style="list-style-type: none"> <li>- Tự động điều hướng Admin quay trở lại trang Quản lý vai trò.</li> </ul>
Ngoại lệ	<ul style="list-style-type: none"> <li>6. Cập nhật vai trò thất bại do nhập thiếu trường           <ul style="list-style-type: none"> <li>6.1. Hệ thống thông báo thất bại do người dùng không nhập trường bắt buộc</li> <li>6.2. Người dùng nhập đầy đủ các trường và bấm tạo mới</li> <li>6.3. Hệ thống thông báo tạo thành công</li> </ul> </li>   <li>6. Cập nhật vai trò thất bại do tên vai trò đã tồn tại           <ul style="list-style-type: none"> <li>6.1. Hệ thống thông báo thất bại do tên vai trò đã tồn tại</li> <li>6.2. Người dùng nhập lại tên vai trò</li> <li>6.3. Hệ thống thông báo tạo thành công</li> </ul> </li> </ul>

Bảng 2.45 Kịch bản chuẩn chức năng sửa vai trò

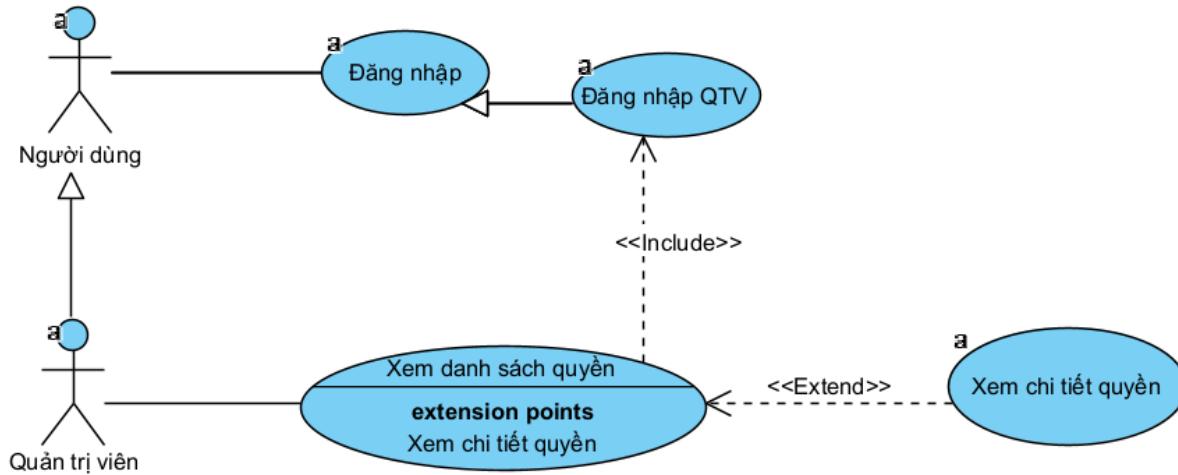
- Kịch bản chức năng xóa vai trò

Use case	Xóa vai trò
Actor	Quản trị viên
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin, ở giao diện cho quản trị
Hậu điều kiện	Quản trị viên xóa vai trò thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Quản trị viên truy cập trang “Quản lý vai trò”</li> <li>2. Hệ thống hiển thị trang Quản lý vai trò gồm danh sách các vai trò hiện có: id, tên vai trò, mô tả, số quyền, số người dùng, thao tác: chỉnh sửa, xóa</li> <li>3. Admin bấm chọn xóa 1 vai trò</li> <li>4. Hệ thống hiển thị 1 popup chứa thông tin “Bạn có chắc chắn muốn xóa vai trò này?” + nút “Xác nhận”</li> <li>5. Admin bấm nút “Xác nhận”</li> <li>6. Hệ thống hiển thị thông báo: “Xóa vai trò thành công”</li> </ol>

Ngoại lệ	4. Admin hủy xóa 4.1. Admin chọn “Hủy xóa” 4.2. Hệ thống hiển thị danh sách các vai trò
----------	---

Bảng 2.46 Kịch bản chuẩn chức năng xóa vai trò

**m. xem danh sách quyền**



Hình 2.28 Biểu đồ UC xem danh sách quyền

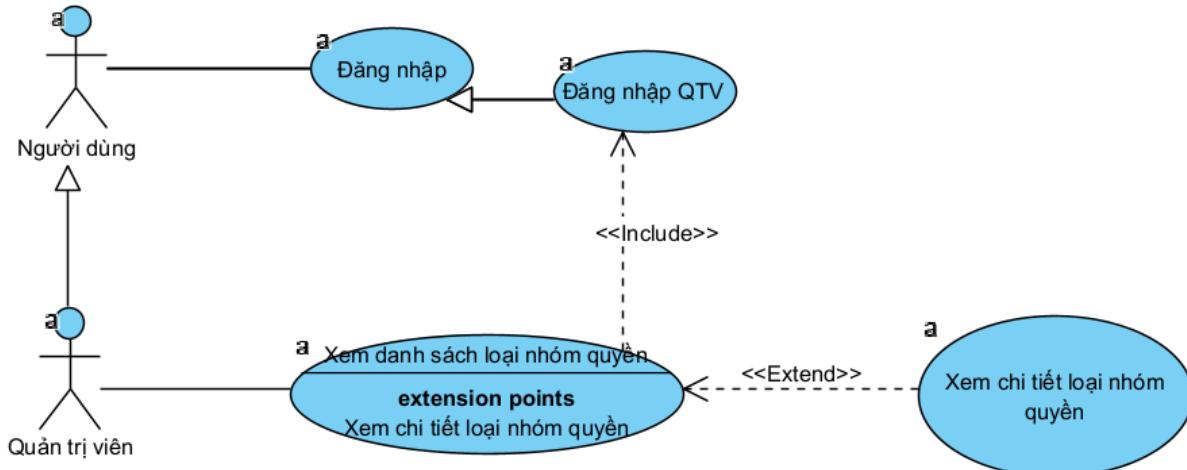
- Kịch bản chức năng xem danh sách quyền

Use case	Xem danh sách quyền
Actor	Quản trị viên
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin, ở giao diện quản trị
Hậu điều kiện	Quản trị viên xem được thông tin chi tiết của quyền
Kịch bản chính	5. Quản trị viên truy cập trang “Quản lý phân quyền” 6. Hệ thống hiển thị danh sách quyền có trong hệ thống với một số thông tin: id, mã phân quyền, mô tả, loại phân quyền, thao tác

	<table border="1"> <thead> <tr> <th>ID</th><th>MÃ PHÂN QUYỀN</th><th>MÔ TẢ</th><th>LOẠI PHÂN QUYỀN</th></tr> </thead> <tbody> <tr> <td>84</td><td>contest_participants.create</td><td>Create contest_participants</td><td>Contest Management</td></tr> <tr> <td>87</td><td>contest_participants.delete</td><td>Delete contest_participants</td><td>Contest Management</td></tr> </tbody> </table> <p>7. Quản trị viên bấm chọn xem chi tiết      8. Trang chi tiết hiển thị các thông tin chính về quyền gồm: mã phân quyền, mô tả, loại phân quyền</p>	ID	MÃ PHÂN QUYỀN	MÔ TẢ	LOẠI PHÂN QUYỀN	84	contest_participants.create	Create contest_participants	Contest Management	87	contest_participants.delete	Delete contest_participants	Contest Management
ID	MÃ PHÂN QUYỀN	MÔ TẢ	LOẠI PHÂN QUYỀN										
84	contest_participants.create	Create contest_participants	Contest Management										
87	contest_participants.delete	Delete contest_participants	Contest Management										
Ngoại lệ													

Bảng 2.47 Kịch bản chuẩn chức năng xem danh sách quyền

#### n. Xem danh sách loại nhóm quyền



Hình 2.29 Biểu đồ UC xem danh sách loại nhóm quyền

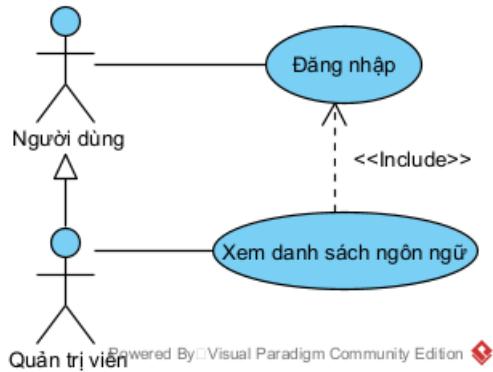
- Kịch bản chức năng xem danh sách loại nhóm quyền

Use case	Xem danh sách loại quyền
Actor	Quản trị viên
Tiền điều kiện	Người dùng đã đăng nhập vào hệ thống và có quyền admin, ở giao diện quản trị
Hậu điều kiện	Quản trị viên xem được thông tin chi tiết của loại quyền

Kịch bản chính	<ol style="list-style-type: none"> <li>1. Quản trị viên truy cập trang “Quản lý loại nhóm quyền”</li> <li>2. Hệ thống hiển thị danh sách nhóm quyền có trong hệ thống với một số thông tin: id, tên loại nhóm quyền, mô tả</li> </ol> <table border="1" data-bbox="481 363 1367 487"> <thead> <tr> <th>ID</th><th>TÊN LOẠI PHẦN QUYỀN</th><th>MÔ TẢ</th></tr> </thead> <tbody> <tr> <td>4</td><td><b>Content Management</b></td><td>Permissions related to content management</td></tr> <tr> <td>7</td><td><b>Contest Management</b></td><td>Permissions related to contest management</td></tr> </tbody> </table> <ol style="list-style-type: none"> <li>3. Quản trị viên bấm chọn xem chi tiết</li> <li>4. Trang chi tiết hiển thị các thông tin chính về loại nhóm quyền gồm: tên loại nhóm quyền, mô tả</li> </ol>	ID	TÊN LOẠI PHẦN QUYỀN	MÔ TẢ	4	<b>Content Management</b>	Permissions related to content management	7	<b>Contest Management</b>	Permissions related to contest management
ID	TÊN LOẠI PHẦN QUYỀN	MÔ TẢ								
4	<b>Content Management</b>	Permissions related to content management								
7	<b>Contest Management</b>	Permissions related to contest management								
Ngoại lệ										

Bảng 2.48 Kích bản chuẩn chức năng xem danh sách loại nhóm quyền

**o. Xem danh sách ngôn ngữ lập trình**



Hình 2.30 UC chi tiết chức năng xem danh sách ngôn ngữ lập trình

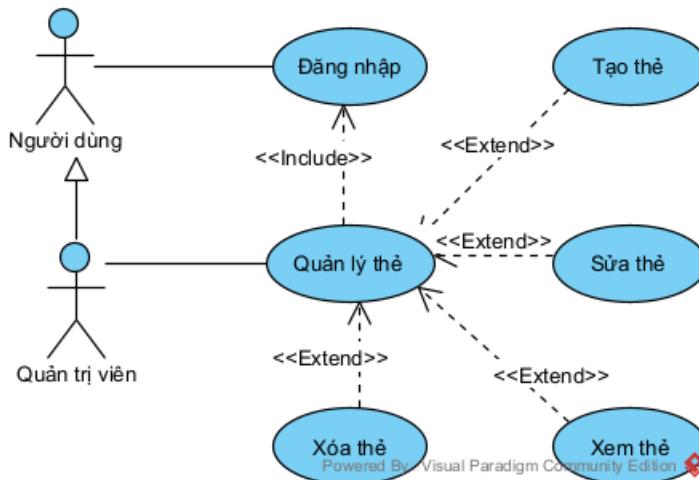
- Kịch bản chính chức năng xem danh sách ngôn ngữ

Use case	Xem danh sách ngôn ngữ lập trình
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công

Hậu điều kiện	Admin xem được danh sách ngôn ngữ lập trình
Kịch bản chính	<ol style="list-style-type: none"> <li>Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>Admin chọn Quản trị</li> <li>Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>Quản trị tìm và chọn chức năng Ngôn ngữ lập trình</li> <li>Hệ thống hiển thị bảng chứa danh sách ngôn ngữ lập trình gồm các cột thông tin mã, tên, external ID (sử dụng cho bên domjudge), extension, active.</li> </ol>
Ngoại lệ	

Bảng 2.49 Kịch bản chuẩn chức năng xem danh sách ngôn ngữ

p. Quản lý thẻ (tag)



Hình 2.31 UC chi tiết chức năng quản lý thẻ

- Kịch bản chính chức năng xem danh sách thẻ

Use case	Xem danh sách thẻ
----------	-------------------

Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xem được danh sách thẻ
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý thẻ</li> <li>6. Hệ thống hiển thị bảng chứa danh sách thẻ gồm các cột dữ liệu tên, slug, ngày tạo</li> </ol>
Ngoại lệ	6. Không có dữ liệu thẻ, hệ thống hiển thị bảng trống và ghi chú không có thẻ để hiển thị, hãy thêm thẻ nếu muốn.

Bảng 2.50 Kịch bản chuẩn chức năng xem danh sách thẻ

- Kịch bản chính chức năng thêm thẻ

Use case	Thêm thẻ
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin thêm được thẻ

Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý thẻ</li> <li>6. Hệ thống hiển thị bảng chứa danh sách thẻ gồm các cột dữ liệu tên, slug, ngày tạo. Trên bảng dữ liệu có 2 ô để nhập tên thẻ và slug cùng nút thêm thẻ</li> <li>7. Admin nhập tên thẻ và slug sau đó click thêm thẻ</li> <li>8. Hệ thống lưu thẻ và thông báo thành công, hiển thị thẻ vừa thêm vào danh sách thẻ</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>7. Admin nhập thiếu thông tin thẻ hoặc slug hoặc cả 2 sau đó click thêm thẻ             <ol style="list-style-type: none"> <li>7.1. Hệ thống báo thiếu thông tin và thêm thẻ thất bại</li> <li>7.2 admin nhập đầy đủ dữ liệu tên thẻ và slug rồi ấn thêm thẻ</li> </ol> </li> </ol> <p>Quay lại bước 8</p>

Bảng 2.51 Kịch bản chuẩn chức năng thêm thẻ

- Kịch bản chính chức năng sửa thẻ

Use case	Sửa thẻ
Actor	Admin
Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin sửa thẻ thành công

Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý thẻ</li> <li>6. Hệ thống hiển thị bảng chứa danh sách thẻ gồm các cột dữ liệu tên, slug, ngày tạo. Ở mỗi dòng dữ liệu thẻ có thẻ action chỉnh sửa</li> <li>7. Admin chọn nút sửa ở thẻ muốn sửa</li> <li>8. Hệ thống đổi trạng thái đọc tên và slug thẻ thành ô input có thể nhập dữ liệu, ấn nút chỉnh sửa và thay vào đó là nút đồng ý hoặc hủy</li> <li>9. Admin chỉnh sửa dữ liệu thẻ và ấn đồng ý</li> <li>10. Hệ thống lưu lại dữ liệu thẻ mới và thông báo thành công, trả lại danh sách thẻ như ở bước 6</li> </ol>
Ngoại lệ	<ol style="list-style-type: none"> <li>9. Admin nhập thiếu thông tin thẻ hoặc slug hoặc cả 2 sau đó click đồng ý             <ol style="list-style-type: none"> <li>9.1. Hệ thống báo thiếu thông tin và thêm thẻ thất bại</li> <li>9.2. admin nhập đầy đủ dữ liệu tên thẻ và slug rồi ấn thêm thẻ</li> </ol> </li> <li>Quay lại bước 10</li> <li>9. Admin ấn hủy</li> <li>9.1. Hệ thống trả lại danh sách thẻ như ở bước 6</li> </ol>

Bảng 2.52 Kịch bản chuẩn chức năng sửa thẻ

- Kịch bản chính chức năng xóa thẻ

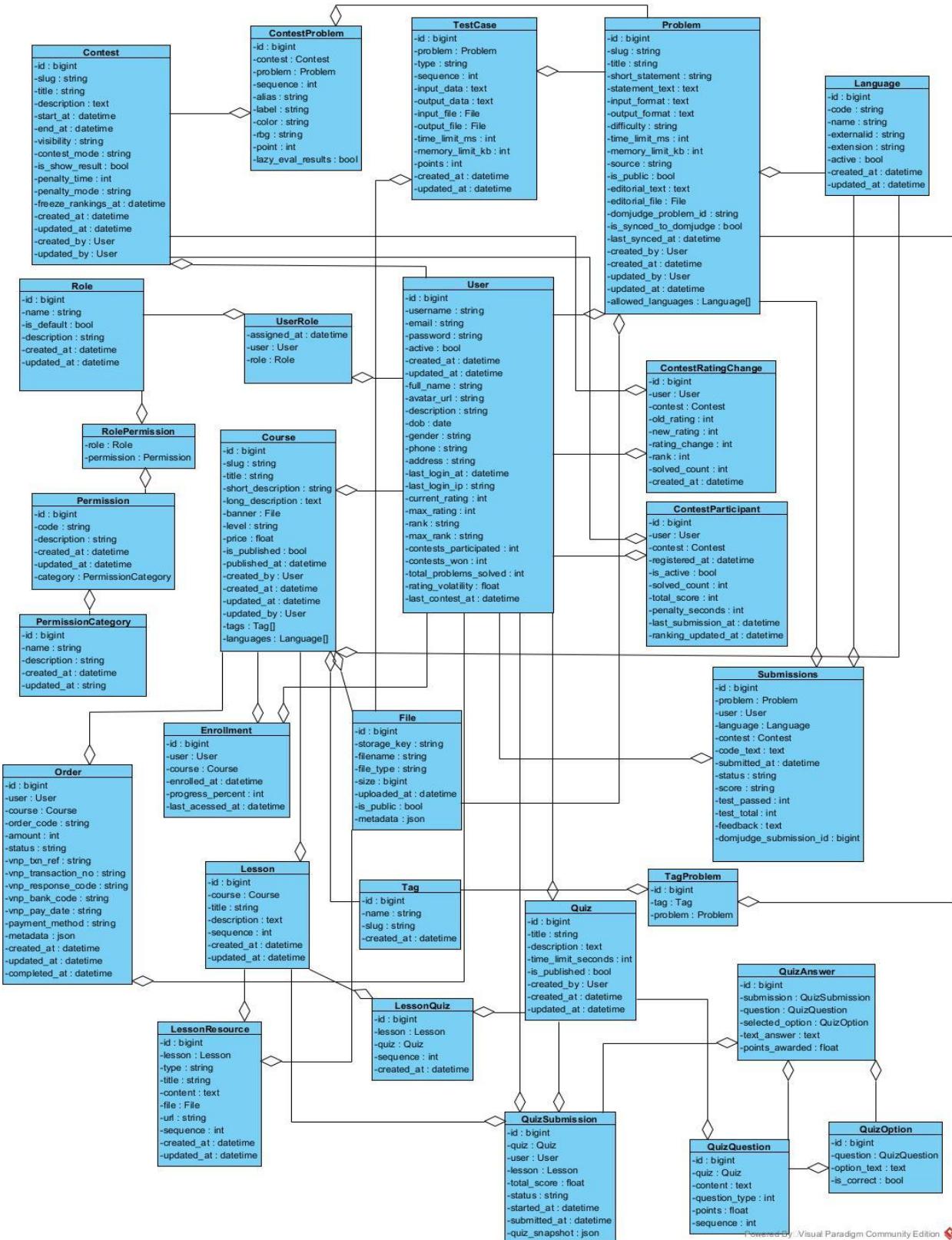
Use case	Xóa thẻ
Actor	Admin

Tiền điều kiện	Admin đăng nhập vào hệ thống thành công
Hậu điều kiện	Admin xóa thẻ thành công
Kịch bản chính	<ol style="list-style-type: none"> <li>1. Sau khi đăng nhập, admin chọn avatar hoặc tên của mình ở góc trên bên phải</li> <li>2. Hệ thống hiển thị dropdown gồm “Hồ sơ cá nhân”, “Quản trị”, “Đăng xuất”</li> <li>3. Admin chọn Quản trị</li> <li>4. Hệ thống hiển thị trang quản trị cho admin gồm sidebar bên trái là các chức năng quản trị, phần bên phải là nội dung chức năng.</li> <li>5. Quản trị tìm và chọn chức năng Quản lý thẻ</li> <li>6. Hệ thống hiển thị bảng chứa danh sách thẻ gồm các cột dữ liệu tên, slug, ngày tạo. Ở mỗi dòng dữ liệu thẻ có thể action chỉnh sửa và xóa thẻ</li> <li>7. Admin chọn nút xóa ở thẻ muốn xóa</li> <li>8. Hệ thống hiển thị form xác nhận có muốn xóa thẻ không cùng 2 nút đồng ý và hủy</li> <li>9. Admin ấn đồng ý</li> <li>10. Hệ thống báo xóa thành công và nút OK</li> <li>11. Admin ấn nút OK</li> <li>12. Hệ thống trả lại danh sách thẻ như ở bước 6 mà không có thẻ vừa bị xóa</li> </ol>
Ngoại lệ	<p>9. Admin ấn hủy</p> <p>9.1. Hệ thống trả lại danh sách thẻ như ở bước 6</p>

Bảng 2.53 Kịch bản chuẩn chức năng xóa thẻ

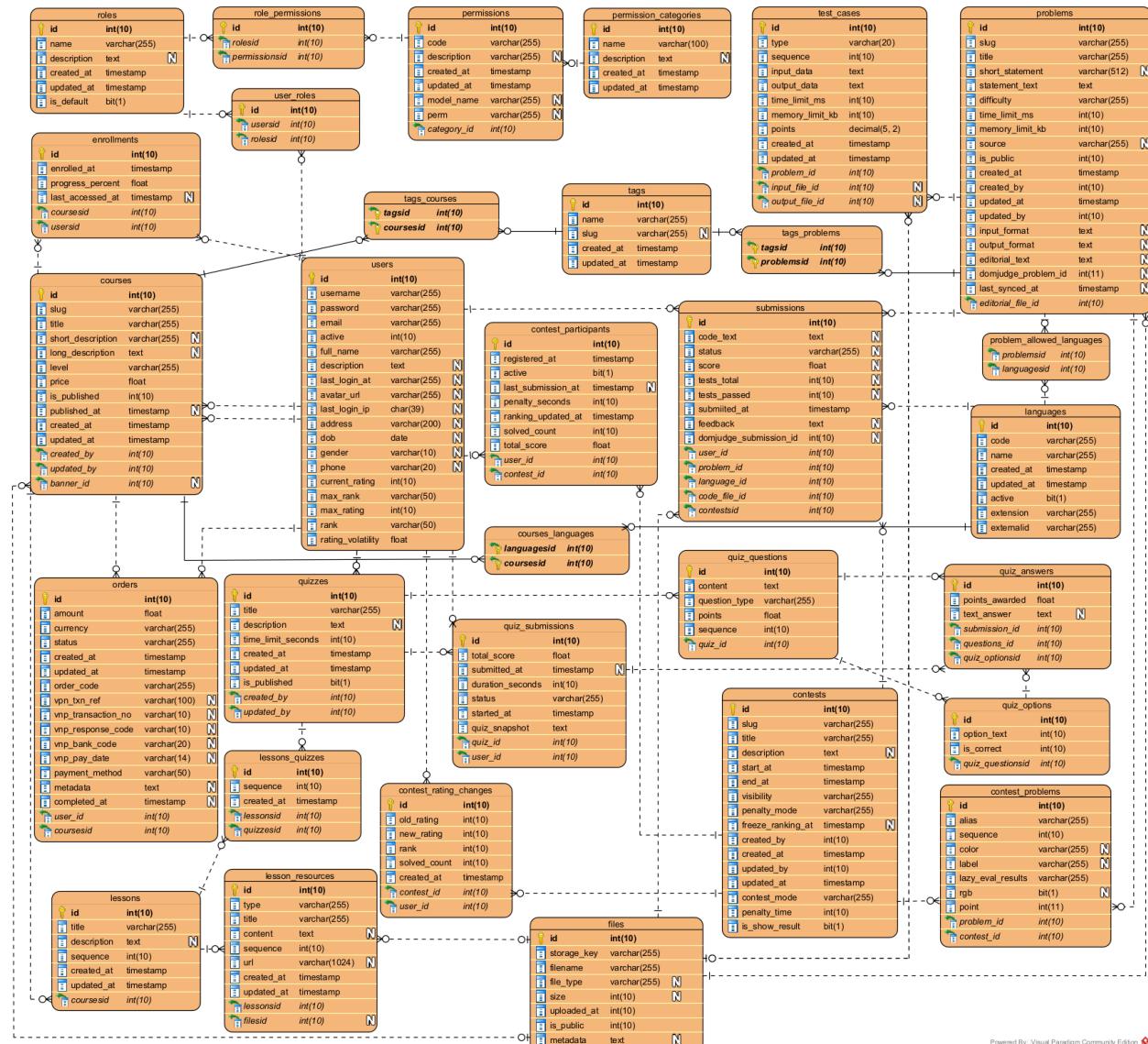
## 2.2. Thiết kế hệ thống

### 2.2.1. Biểu đồ lớp



Hình 2.32 Biểu đồ lớp toàn hệ thống

## 2.2.2. Thiết kế cơ sở dữ liệu

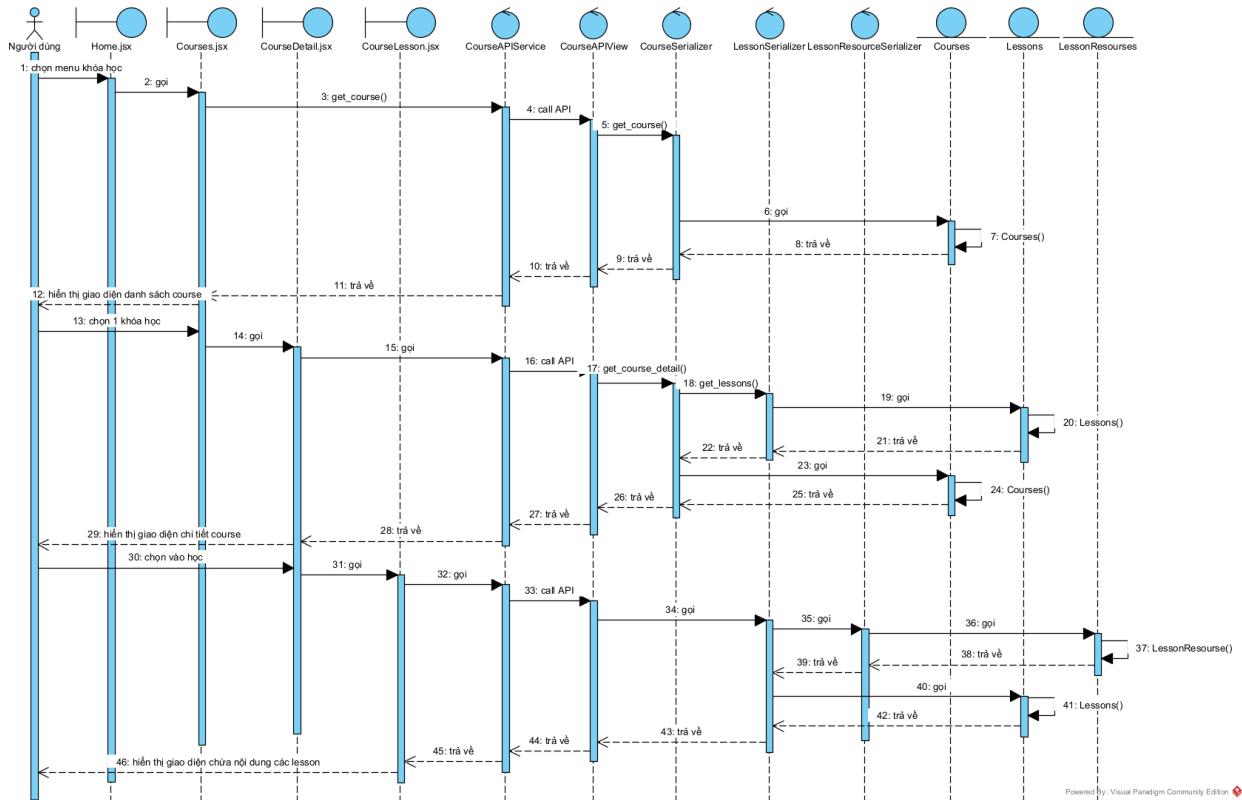


Hình 2.33 Cơ sở dữ liệu

## 2.2.3. Biểu đồ tuần tự cho từng chức năng

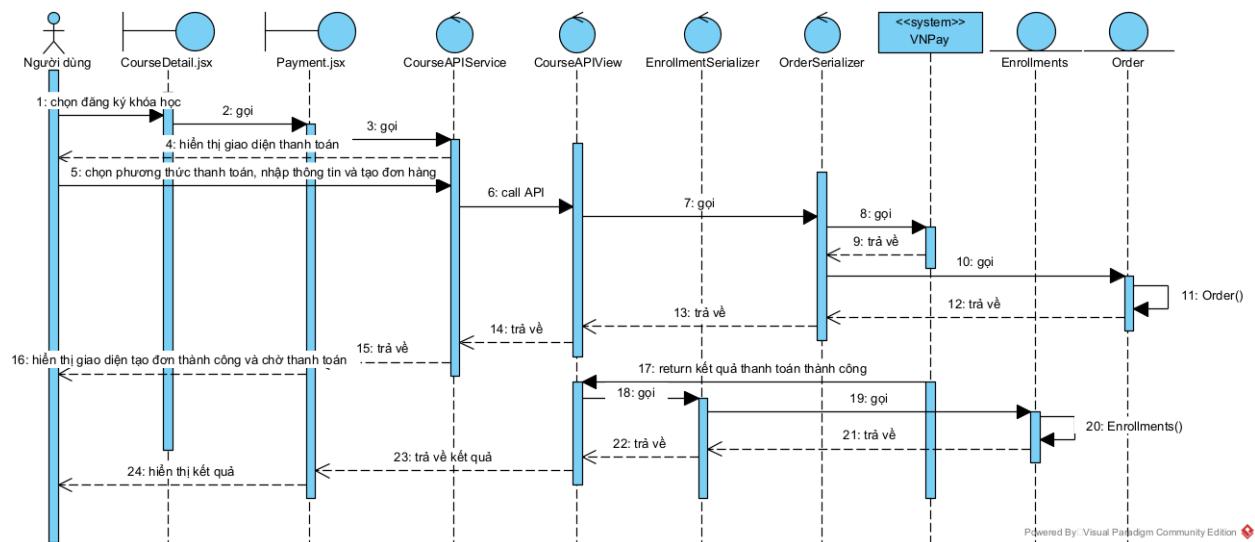
### 2.2.3.1. Người dùng

### a. Xem khóa học



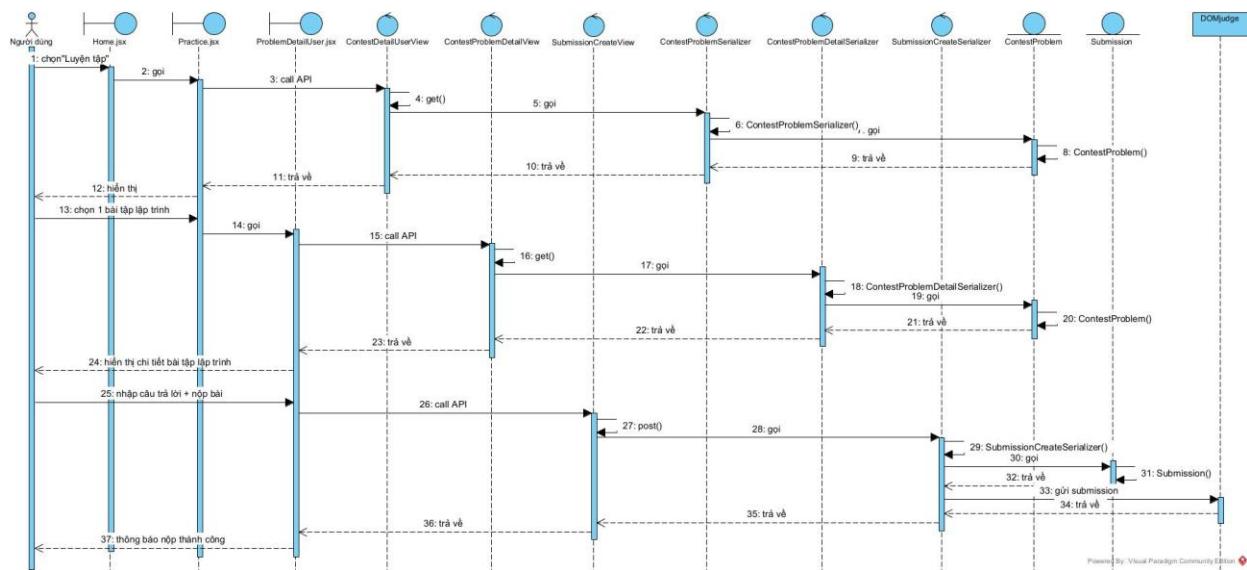
Hình 2.34 Sơ đồ tuần tự chức năng xem khóa học

### b. Đăng ký khóa học



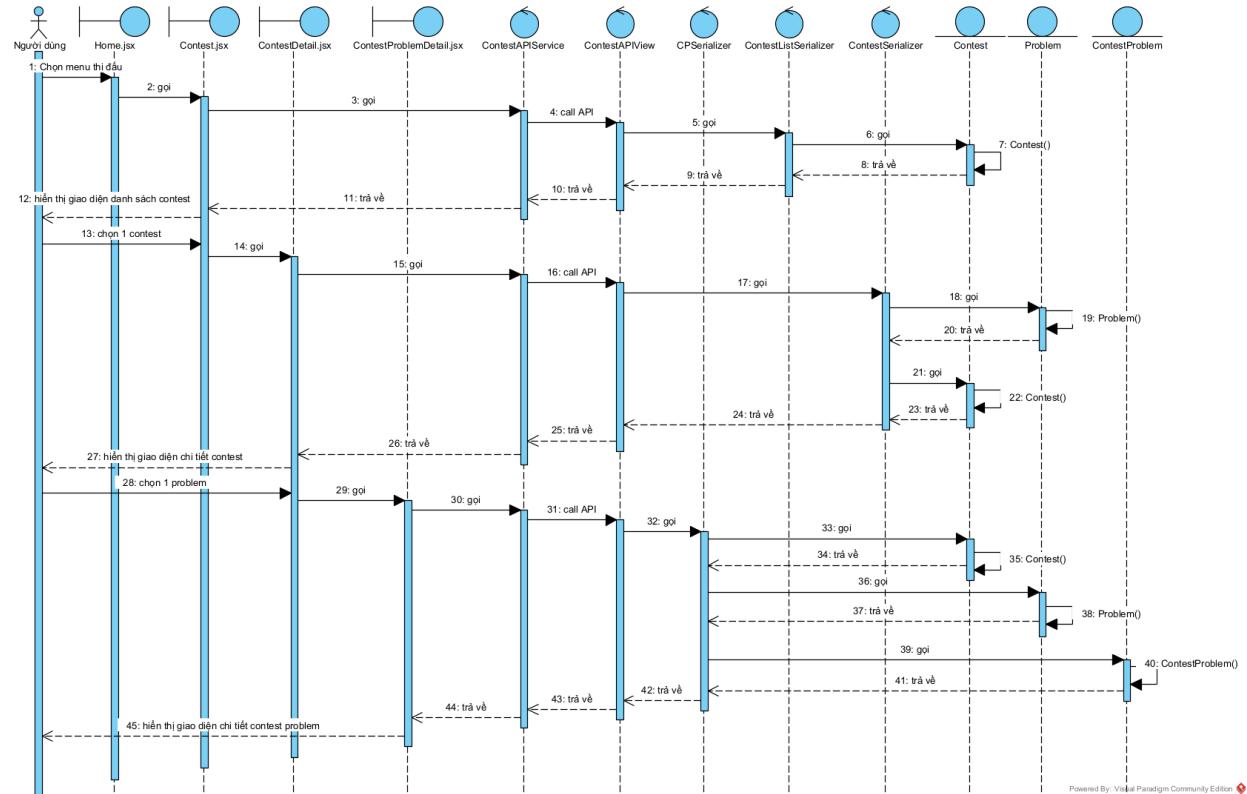
Hình 2.35 Sơ đồ tuần tự chức năng đăng ký khóa học

### c. Nộp bài tập luyện tập



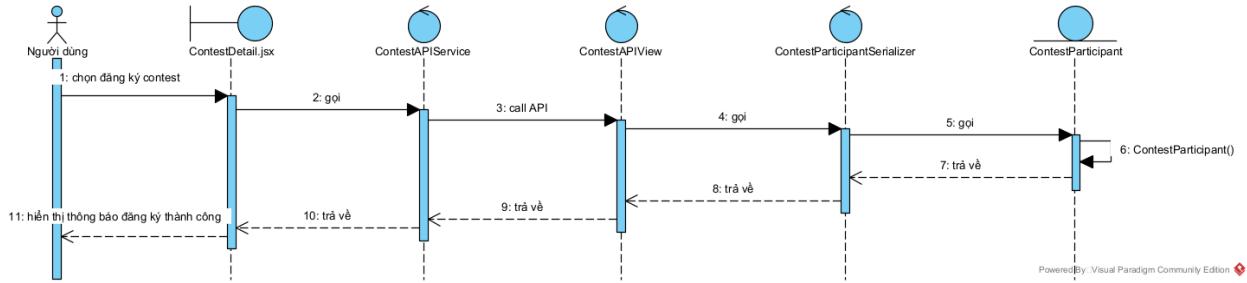
Hình 2.36 Sơ đồ tuần tự chức năng nộp bài tập luyện tập

#### d. Xem cuộc thi lập trình



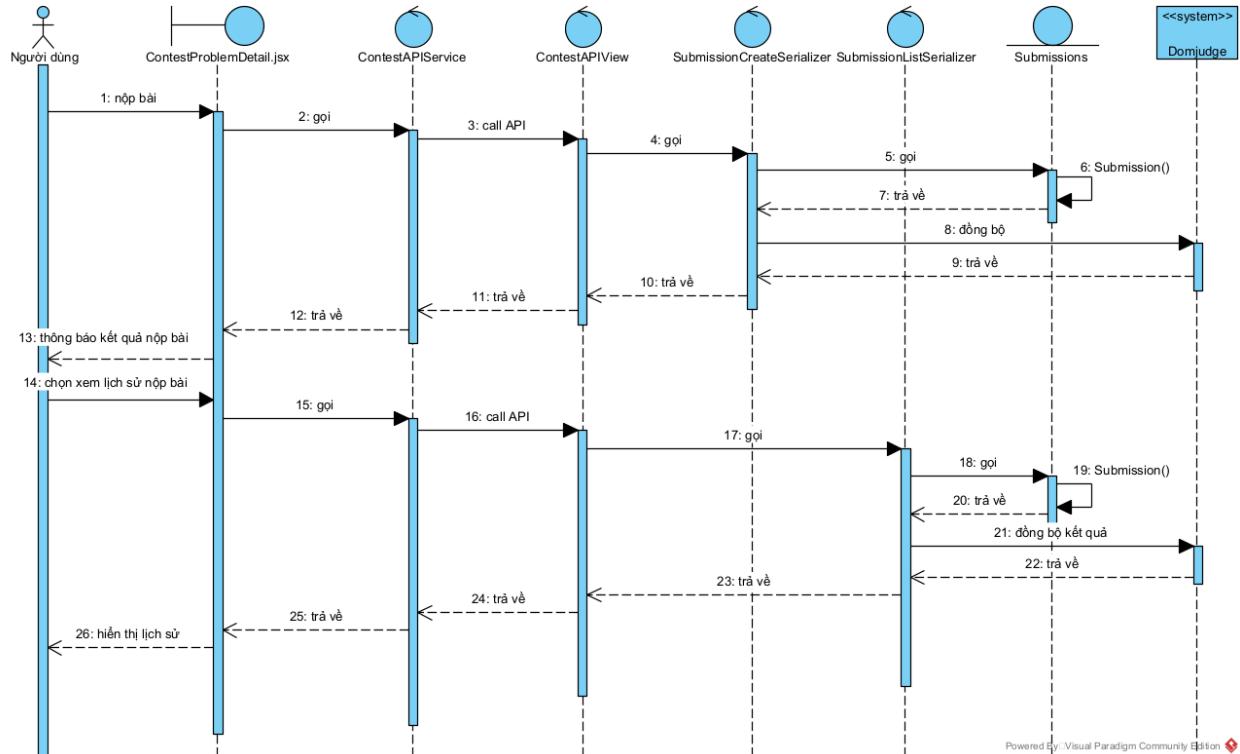
Hình 2.37 Sơ đồ tuần tự chức năng Xem cuộc thi lập trình

#### e. Đăng ký thi lập trình



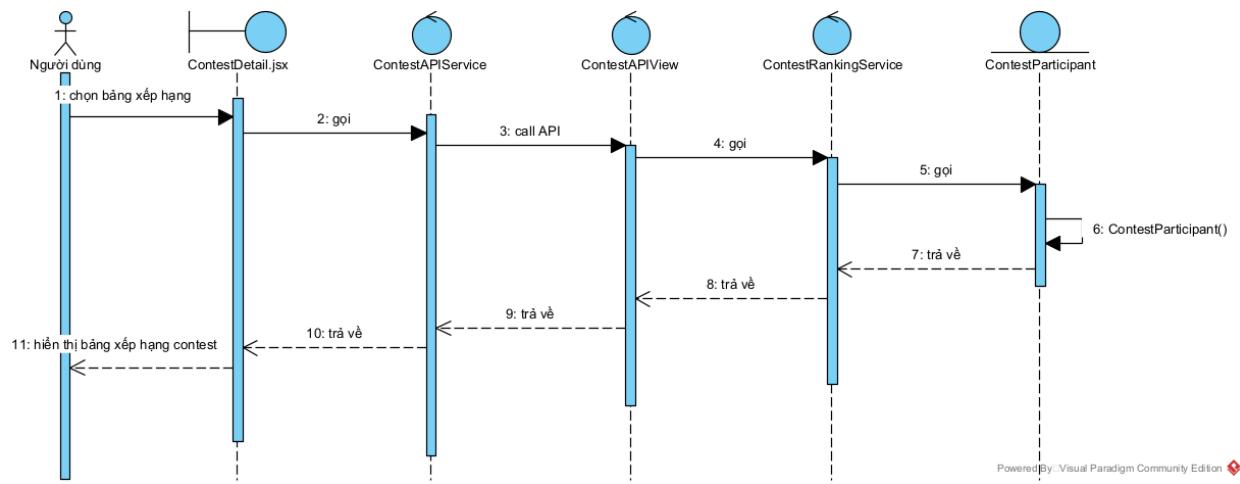
Hình 2.38 Sơ đồ tuần tự chức năng đăng ký thi lập trình

**f. Nộp bài tập trong cuộc thi lập trình**



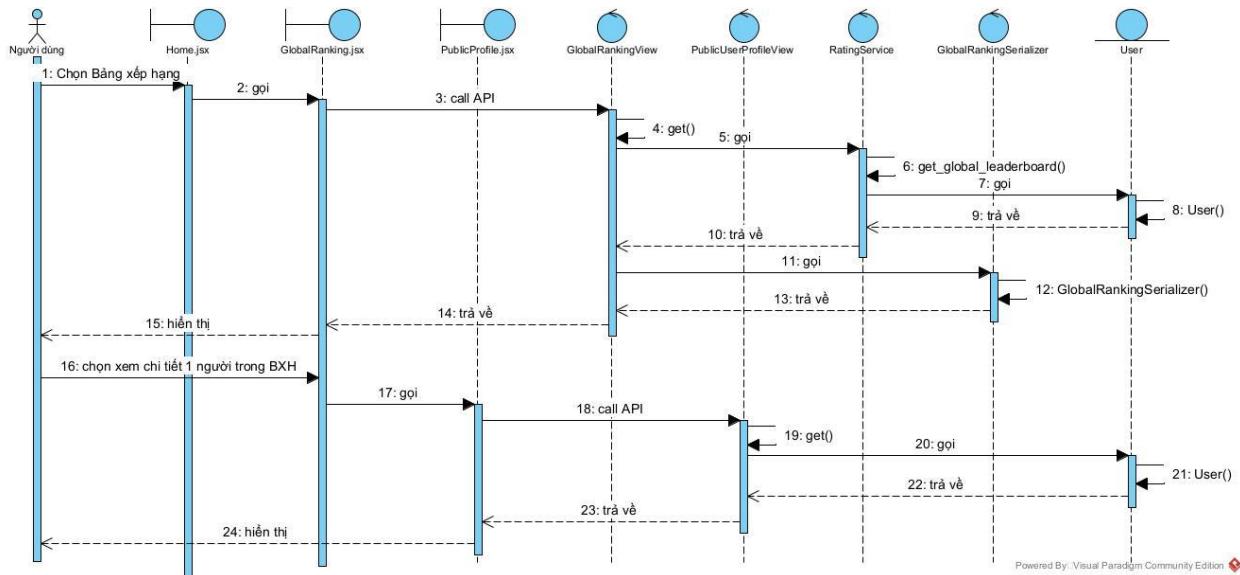
Hình 2.39 Sơ đồ tuần tự chức năng nộp bài tập trong cuộc thi lập trình

**g. Xem bảng xếp hạng cuộc thi**



Hình 2.40 Sơ đồ tuần tự chức năng xem bảng xếp hạng cuộc thi

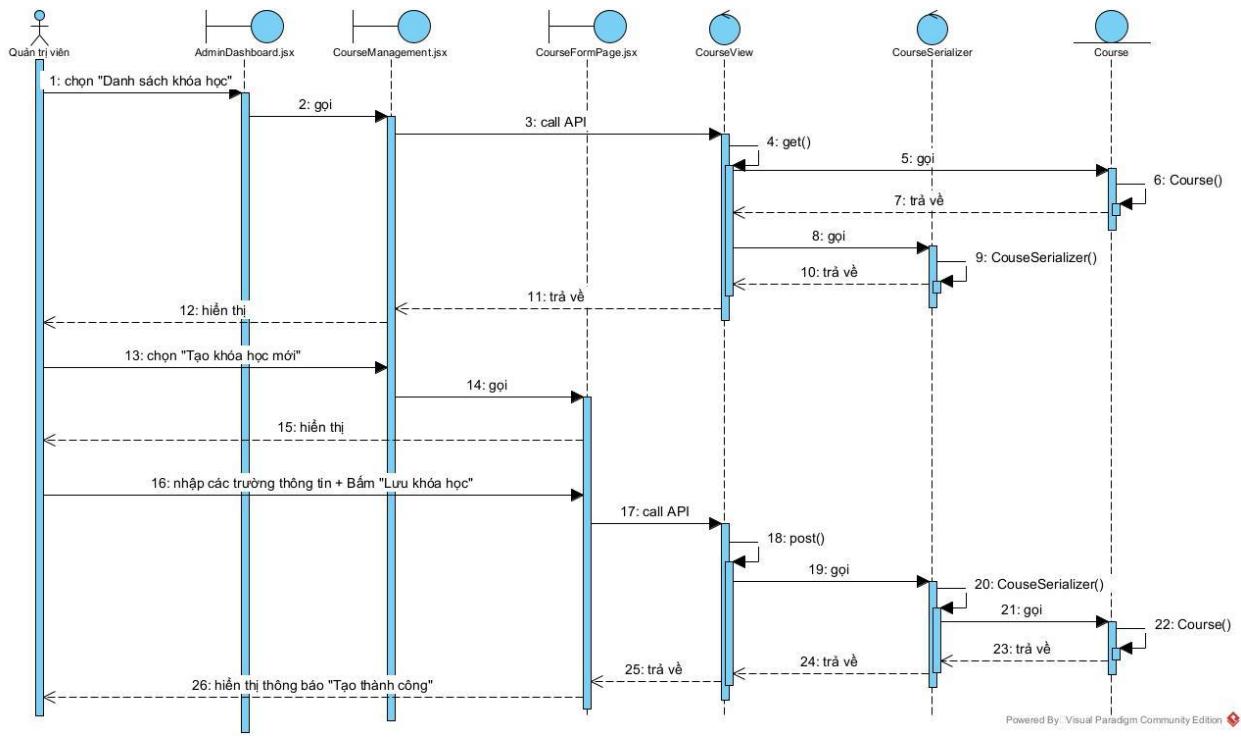
#### **h. Xem bảng xếp hạng tổng**



Hình 2.41 Sơ đồ tuần tự chức năng xem bảng xếp hạng tổng

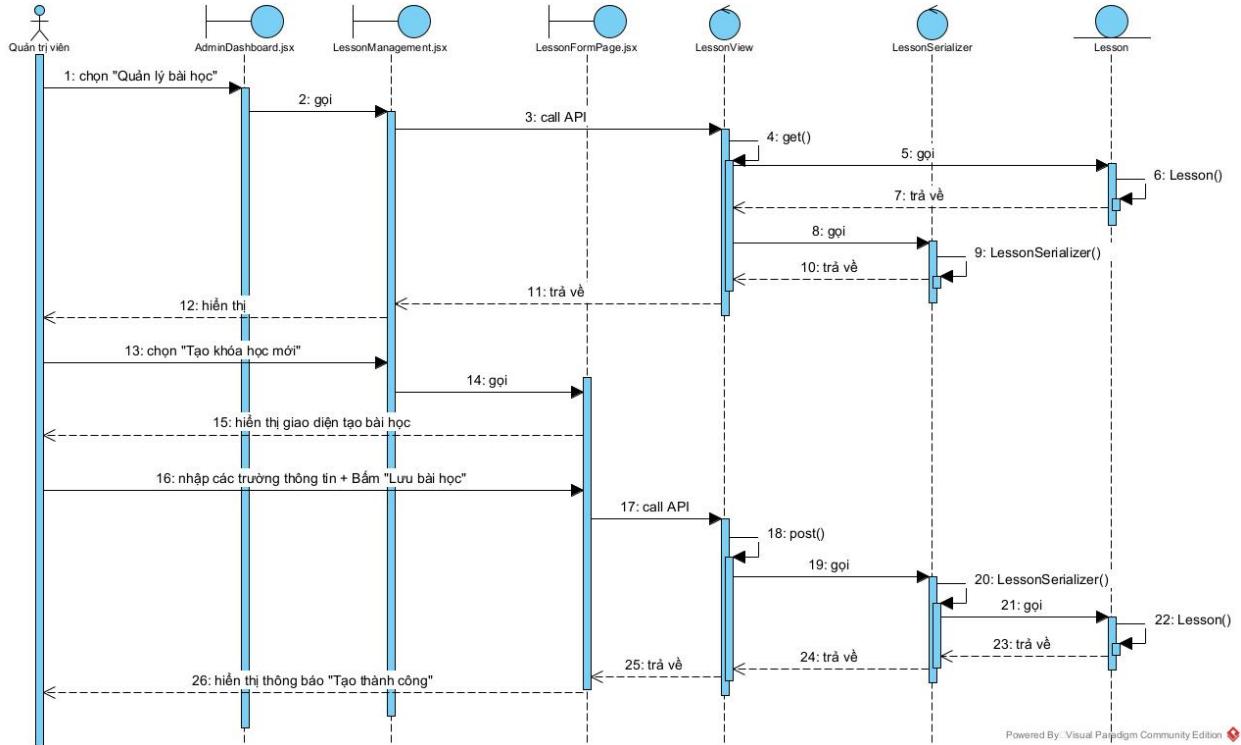
#### 2.2.3.2. Người quản trị

##### a. Tạo khóa học



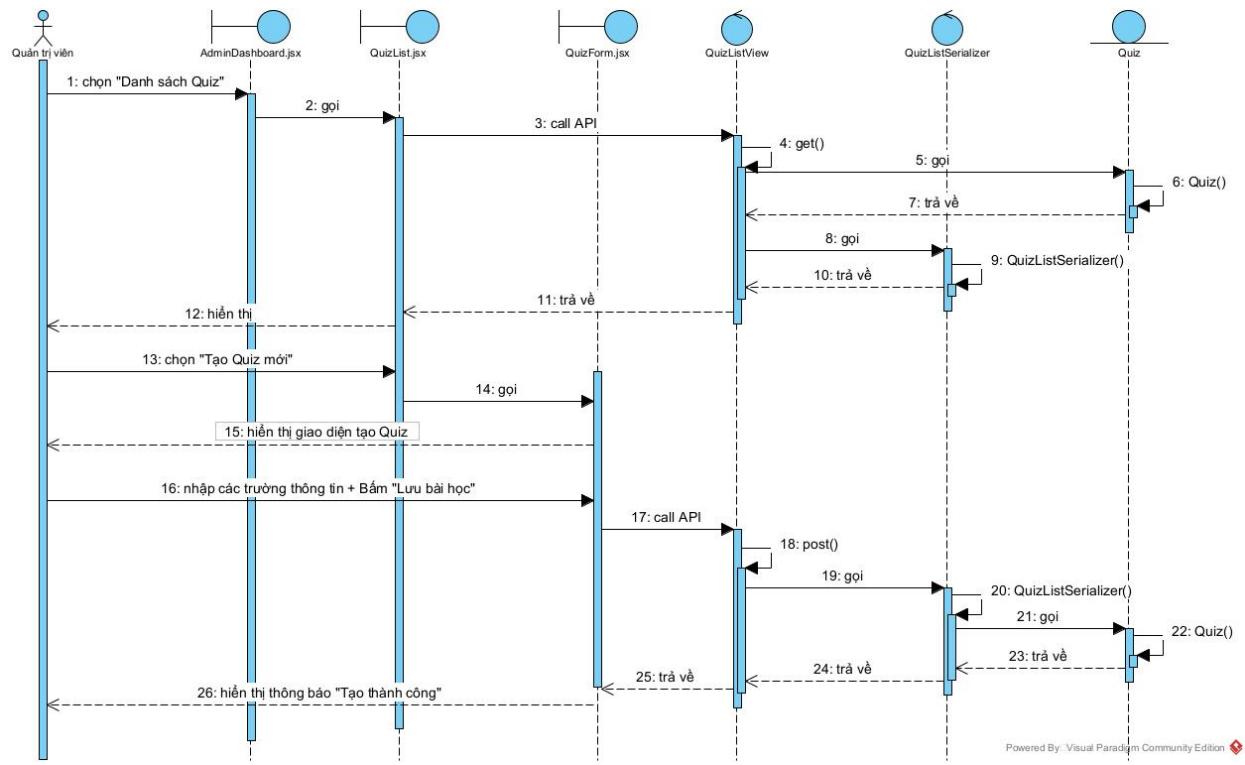
Hình 2.42 Sơ đồ tuần tự chức năng tạo khóa học

### b. Tạo bài học



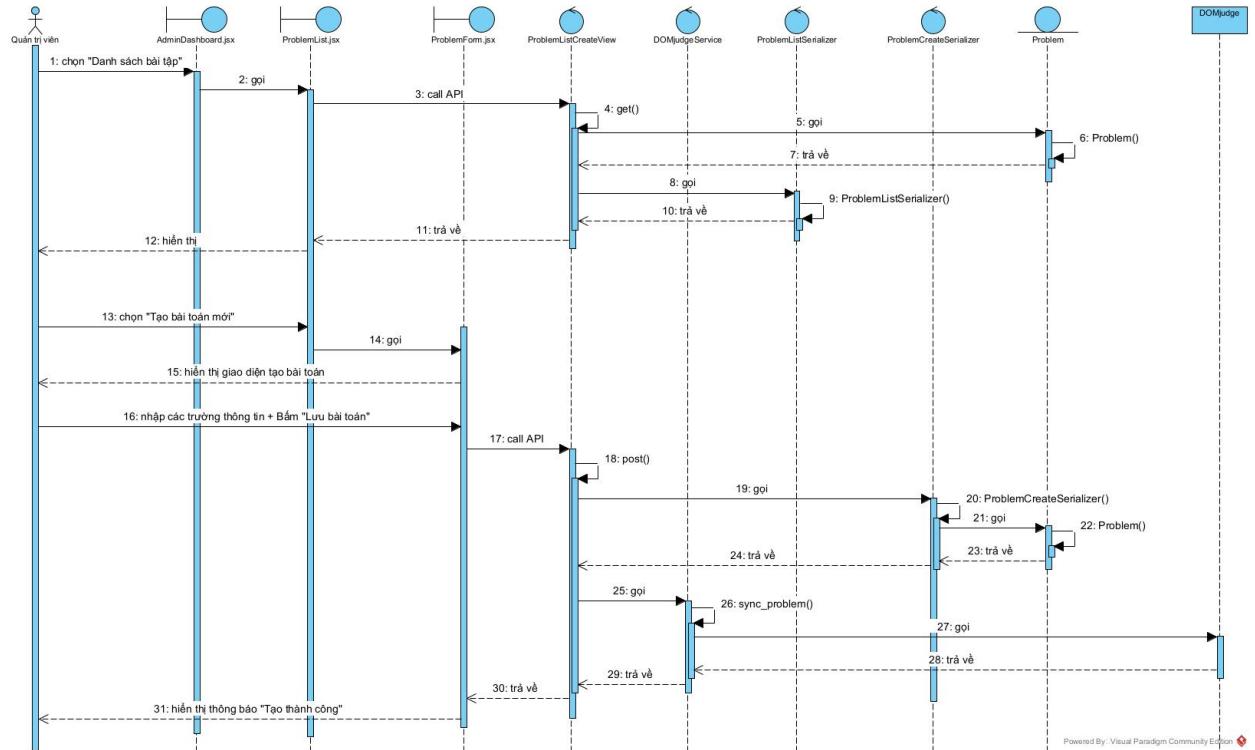
Hình 2.43 Sơ đồ tuần tự chức năng tạo bài học

*c. Tạo quiz*



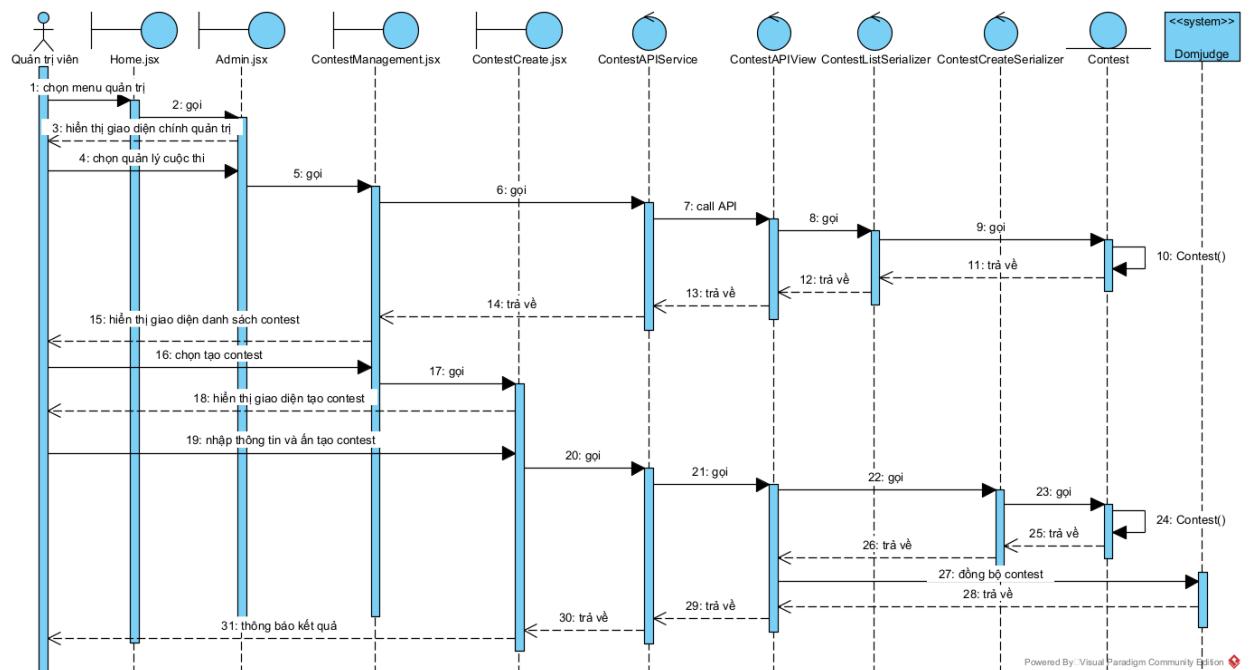
Hình 2.44 Sơ đồ tuần tự chức năng tạo quiz

*d. Tạo bài tập lập trình*



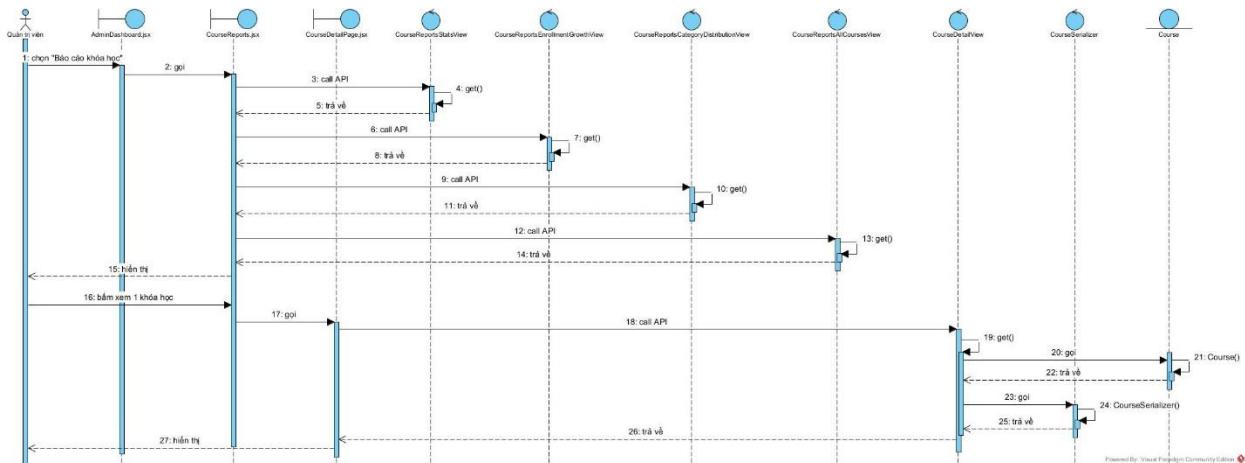
Hình 2.45 Sơ đồ tuần tự chức năng tạo bài tập lập trình

#### e. Tạo cuộc thi lập trình

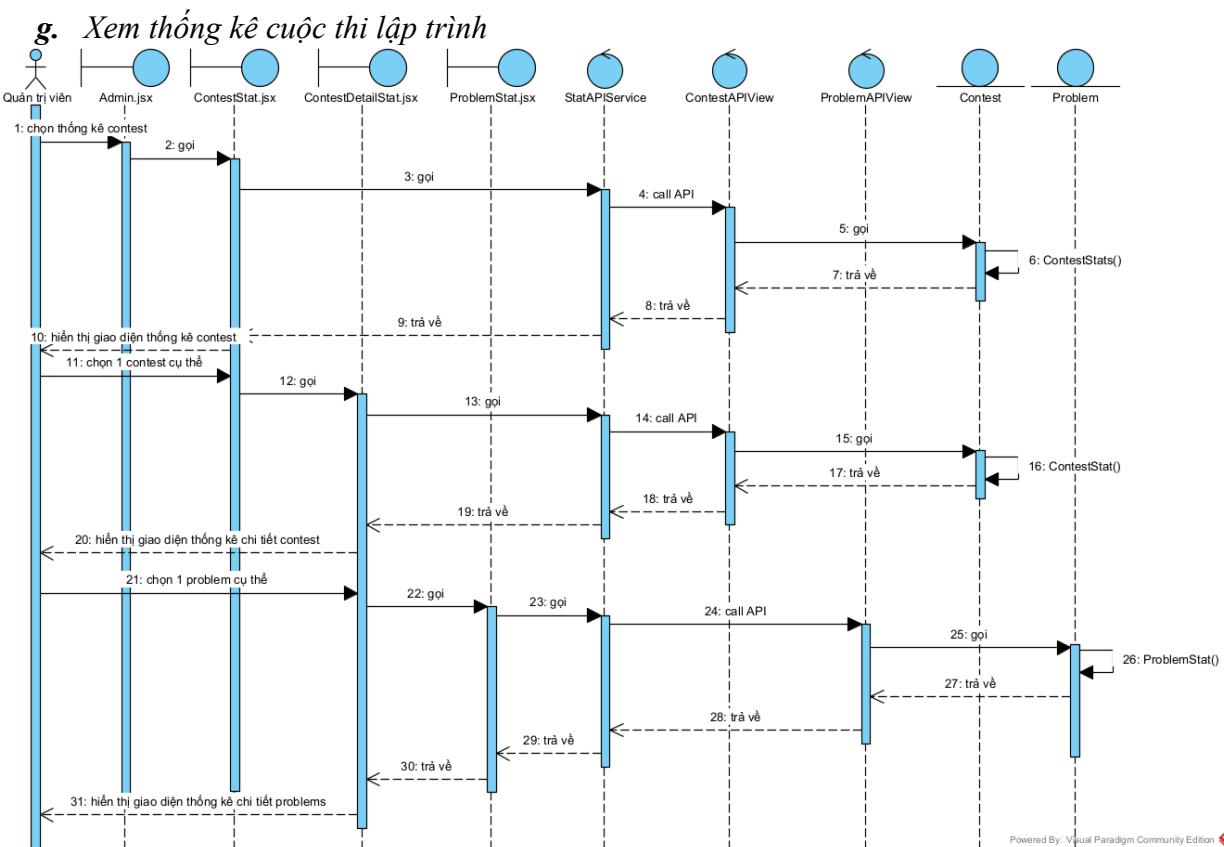


Hình 2.46 Sơ đồ tuần tự chức năng tạo cuộc thi lập trình

#### f. Xem thống kê khóa học



Hình 2.47 Sơ đồ tuần tự chức năng xem thống kê khóa học



Hình 2.48 Sơ đồ tuần tự chức năng xem thống kê cuộc thi lập trình

## 2.3. Tổng kết

Tóm tắt chương

## CHƯƠNG 3: HỆ THỐNG GỢI Ý SỬ DỤNG MÔ HÌNH LAI GHÉP

### 3.1. Tổng quan về bài toán gợi ý trong môi trường học lập trình

#### 3.1.1. Đặt vấn đề

Sự bùng nổ của các nền tảng học lập trình trực tuyến mang lại nguồn tri thức khổng lồ nhưng đồng thời gây ra tình trạng "quá tải thông tin" (Information Overload), khiến người học, đặc biệt là người mới bắt đầu, dễ mất phương hướng. Bên cạnh đó, hiệu quả học tập chỉ đạt tối đa khi độ khó của bài toán tương thích với năng lực hiện tại của người giải. Một bài tập quá dễ sẽ gây nhảm chán, trong khi bài quá khó dẫn đến nản lòng và bỏ cuộc.

Trong bối cảnh đó, việc xây dựng hệ thống gợi ý tự động (Recommender System) trở nên cần thiết. Hệ thống này đóng vai trò như một "người cố vấn ảo", phân tích lịch sử hành vi để đưa ra các đề xuất bài tập tiếp theo chính xác nhất, giúp tối ưu hóa thời gian và duy trì động lực cho người học.

#### 3.1.2. Phân tích đặc thù dữ liệu

Việc xây dựng hệ thống gợi ý trong miền lập trình phải đổi mới với những đặc điểm dữ liệu khác biệt so với các lĩnh vực thương mại điện tử hay giải trí.

Thứ nhất là vấn đề dữ liệu thưa (Data Sparsity). Trong không gian hàng nghìn bài tập, một người dùng trung bình thường chỉ giải quyết được một tỷ lệ rất nhỏ (thường dưới 1%), tạo ra các ma trận tương tác với mật độ lấp đầy cực thấp, gây khó khăn lớn cho việc tìm kiếm láng giềng trong các thuật toán lọc cộng tác thông thường.

Thứ hai là tính chất phản hồi ẩn (Implicit Feedback). Khác với các hệ thống xem phim hay mua sắm nơi người dùng để lại đánh giá 1-5 sao (Explicit rating), các nền tảng lập trình chủ yếu ghi nhận hành động nộp bài với các trạng thái như "Accepted" (Đúng) hoặc "Wrong Answer" (Sai). Hệ thống phải tự suy diễn mức độ quan tâm và sở thích của người dùng thông qua tần suất và kết quả của các lần nộp bài này thay vì dựa vào điểm số trực tiếp.

Cuối cùng, các đặc trưng nội dung (Content features) của bài tập mang tính cấu trúc cao. Mỗi bài toán đều gắn liền với các thẻ thuật toán (Tags) cụ thể (như quy hoạch động, đồ thị) và chỉ số độ khó (Difficulty). Đây là nguồn thông tin bổ trợ quan trọng để khắc phục điểm yếu của dữ liệu thưa.

#### 3.1.3. Kết luận

Để giải quyết các thách thức về dữ liệu thưa và phản hồi ẩn nêu trên, hệ thống được thiết kế dựa trên mô hình lai ghép (Hybrid) kết hợp giữa hai nhánh tiếp cận: lọc cộng tác dựa trên mô hình (Model-based Collaborative Filtering) và lọc theo nội dung (Content-based Filtering).

### 3.2. Cơ sở lý thuyết

#### 3.2.1. Lọc theo nội dung (Content-based Filtering)

##### 3.2.1.1. Nguyên lý hoạt động

Phương pháp lọc theo nội dung (Content-based Filtering - CB) hoạt động dựa trên giả định rằng nếu người dùng thích một sản phẩm trong quá khứ, họ cũng sẽ thích những sản phẩm có đặc tính tương tự trong tương lai. Trong ngữ cảnh bài toán lập trình, "đặc tính" của bài tập được xác định thông qua các siêu dữ liệu (metadata) như: dạng bài (Tags/Topics), độ khó (Difficulty Level). Hệ thống Content-based xây dựng hai loại hồ sơ:

- Item Profile: Một vector biểu diễn các đặc trưng của bài tập.
- User Profile: Tổng hợp các đặc trưng của những bài tập mà người dùng đó đã giải quyết thành công.

Quá trình gợi ý thực chất là việc tìm kiếm những bài tập mới có khoảng cách gần nhất với hồ sơ sở thích của người dùng trong không gian vector. Ưu điểm lớn nhất của phương pháp này là khả năng giải quyết vấn đề “Khởi động lạnh cho bài tập mới” (New Item Cold Start) – khi một bài tập mới được thêm vào, hệ thống vẫn có thể gợi ý nó dựa trên nội dung mà không cần đợi ai đó giải trước. Tuy nhiên, nhược điểm của nó là "Over-specialization" (chuyên môn hóa quá mức), nghĩa là hệ thống chỉ gợi ý những thứ người dùng đã biết, thiếu tính khám phá cái mới.

##### 3.2.1.2. Tìm hiểu về mô hình toán học được sử dụng cho Content-based Filtering

Trong phương pháp lọc theo nội dung, hệ thống sử dụng thuật toán k lâng giềng gần nhất (k-Nearest Neighbors - KNN) để cụ thể hóa việc tìm kiếm sự tương đồng. KNN là một thuật toán học máy thuộc nhóm học lười (Lazy Learning) và phi tham số (Non-parametric):

- Học lười: KNN không "học" một hàm phân loại tổng quát từ dữ liệu huấn luyện ngay lập tức (như Hồi quy hay Mạng nơ-ron). Thay vào đó, nó lưu trữ toàn bộ các vector đặc trưng của bài tập vào bộ nhớ (hoặc cấu trúc cây như KD-Tree) và chỉ thực hiện tính toán khi cần đưa ra gợi ý.

- Phi tham số: Thuật toán không đưa ra giả định về phân phôi của dữ liệu cơ sở (như giả định dữ liệu tuân theo phân phôi chuẩn), giúp nó linh hoạt và phù hợp với các dữ liệu đặc thù như tags hay độ khó bài tập.

Trong dự án này, KNN được sử dụng để tìm ra k bài tập có đặc trưng gần nhất với hồ sơ sở thích của người dùng. Quy trình toán học được thực hiện qua các bước sau:

#### Bước 1: Biểu diễn vector và Tính độ đo khoảng cách

Mỗi bài tập  $i$  được biểu diễn thành vector  $d$  chiều  $\vec{V}_i$ . Độ tương đồng giữa hai bài tập  $i$  và  $j$  được tính bằng Cosine Similarity:

$$Sim(i, j) = \cos(\vec{V}_i, \vec{V}_j) = \frac{\vec{V}_i \cdot \vec{V}_j}{\|\vec{V}_i\| \times \|\vec{V}_j\|}$$

#### Bước 2: Xác định tập láng giềng (Neighborhood Set)

Với mỗi bài tập  $i$  mà người dùng đã làm, mô hình xác định tập hợp  $N_k(i)$  gồm  $k$  bài tập khác có độ tương đồng cao nhất:

$$N_k(i) = \{j \in I \mid j \neq i, rank(Sim(i, j)) \leq k\}$$

#### Bước 3: Hàm dự đoán điểm (Prediction Function)

Điểm gợi ý cho một bài tập ứng viên  $c$  đối với người dùng  $u$  được tính bằng tổng trọng số độ tương đồng của  $c$  với tập hợp các bài tập  $H_u$  mà người dùng đã giải quyết trong quá khứ:

$$Score_{CB}(u, c) = \frac{\sum_{i \in H_u} w_{ui} \cdot Sim(i, c)}{\sum_{i \in H_u} w_{ui}}$$

Trong đó:  $w_{ui}$  là mức độ yêu thích/trọng số của user  $u$  đối với bài tập cũ  $i$  (ví dụ:  $w=1$  nếu đã AC).

### 3.2.2. Lọc cộng tác dựa trên mô hình (Model-based Collaborative Filtering)

#### 3.2.2.1. Nguyên lý hoạt động

Lọc cộng tác (Collaborative Filtering - CF) tiếp cận vấn đề theo hướng hành vi xã hội: "Những người có hành vi giống nhau trong quá khứ sẽ có xu hướng hành động giống nhau trong tương lai". Khác với phương pháp dựa trên bộ nhớ (Memory-based/KNN) vốn tính toán trực tiếp

trên dữ liệu thô và gặp khó khăn khi dữ liệu thưa, phương pháp Model-based sử dụng các kỹ thuật học máy để tìm ra các "yếu tố ẩn" (latent factors) chi phối hành vi người dùng.

Kỹ thuật tiêu biểu nhất là phân rã ma trận (Matrix Factorization). Ý tưởng cốt lõi là ánh xạ cả người dùng và bài tập vào cùng một không gian ẩn có số chiều thấp (ví dụ: k=20 hoặc 50). Trong không gian này, mỗi chiều có thể đại diện cho một khái niệm trừu tượng mà máy tự học được (ví dụ: "kỹ năng đệm quy", "tư duy toán học", hay "độ kiên nhẫn"). Mức độ phù hợp giữa một người dùng và một bài tập chính là mức độ tương thích của các tọa độ ẩn này.

Đối với dữ liệu phản hồi ẩn (Implicit Feedback) như lịch sử nộp bài (chỉ biết nộp hay chưa, đúng hay sai, không có rating 1-5 sao), thuật toán **Alternating Least Squares (ALS)** được chứng minh là hiệu quả nhất. Nó mô hình hóa việc "chưa nộp bài" không phải là "không thích" (rating = 0), mà là "chưa biết" với độ tin cậy thấp, từ đó cho phép hệ thống khai phá sở thích tiềm ẩn tốt hơn.

### 3.2.2.2. Mô hình toán học

Hệ thống sử dụng thuật toán bình phương tối thiểu thay thế (ALS) để giải quyết bài toán tối ưu hóa trong phân rã ma trận. Lý do lựa chọn ALS thay vì các phương pháp như Gradient Descent nằm ở bản chất của hàm mục tiêu: nếu xét đồng thời cả hai biến ma trận người dùng ( $X$ ) và ma trận bài tập ( $Y$ ), hàm mục tiêu là không lồi (non-convex) và rất khó tìm nghiệm tối ưu.

Tuy nhiên, nếu cố định ma trận  $X$ , hàm mục tiêu trở thành hàm bậc hai lồi (convex) đối với  $Y$ , và ngược lại. ALS tận dụng tính chất này để giải quyết bài toán bằng cách luân phiên cố định một ma trận và tối ưu hóa ma trận kia, đảm bảo hàm lồi giảm dần sau mỗi bước lặp và hội tụ nhanh chóng, đồng thời hỗ trợ tính toán song song hiệu quả.

Công thức toán học cho phản hồi ẩn (Implicit Feedback Variation): Giả sử có ma trận tương tác  $R$  kích thước  $M \times N$ . Mục tiêu là xấp xỉ  $R$  bằng tích của hai ma trận ẩn  $X \in \mathbb{R}^{M \times k}$  và  $Y \in \mathbb{R}^{N \times k}$ .

Ta định nghĩa biến nhị phân sở thích (Preference)  $p_{ui}$  và biến độ tin cậy (Confidence)  $c_{ui}$ :

$$p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases}$$

$$c_{ui} = 1 + \alpha r_{ui}$$

Trong đó,  $\alpha$  là tham số tăng cường độ tin cậy.

Hàm mất mát(Loss Function) cần tối ưu hóa:

$$L(X, Y) = \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

Thuật toán ALS giải bài toán này bằng cách lặp lại hai bước tối ưu hóa luân phiên cho đến khi hội tụ:

**Bước 1:** Cố định ma trận  $Y$ , tính toán lại vector người dùng  $x_u$ :

$$x_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p_u$$

**Bước 2:** Cố định ma trận  $X$ , tính toán lại vector bài tập  $y_i$ :

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p_i$$

Trong đó:  $C^u$  và  $C^i$  là các ma trận đường chéo chứa độ tin cậy  $c_{ui}$ .

### 3.2.3. Mô hình gợi ý lai ghép (Hybrid Recommendation System)

#### 3.2.3.1. Nguyên lý hoạt động

Mỗi phương pháp đơn lẻ đều có nhược điểm chí mạng: Content-based thiếu tính đa dạng và không tận dụng được hành vi người dùng; Collaborative Filtering bị tê liệt khi gặp người dùng mới (Cold Start).

Mô hình lai ghép (Hybrid) ra đời để tận dụng ưu điểm của cả hai. Có nhiều chiến lược lai ghép như: xếp tầng (Cascade), chuyển đổi (Switching), trộn đặc trưng (Feature Augmentation). Trong dự án này sử dụng chiến lược lai ghép có trọng số (Weighted Hybrid).

Chiến lược này xem mỗi mô hình thành phần, bao gồm lọc theo nội dung và lọc cộng tác dựa trên mô hình, như các thành phần dự đoán độc lập. Mỗi mô hình tạo ra một điểm gợi ý riêng cho từng đối tượng, và kết quả gợi ý cuối cùng được xác định thông qua việc kết hợp các điểm này bằng một hàm tổ hợp tuyến tính. Cách tiếp cận này giúp hệ thống tăng tính linh hoạt và ổn định, đặc biệt trong các trường hợp một mô hình hoạt động kém hiệu quả, chẳng hạn khi lọc cộng tác thiếu dữ liệu đối với người dùng mới, thì mô hình còn lại có thể bù đắp và duy trì chất lượng gợi ý.

#### 3.2.3.2. Mô hình toán học

Điểm số gợi ý tổng hợp  $Score_{hybrid}$  cho người dùng  $u$  và bài tập  $i$  được tính như sau:

$$Score_{final}(u, i) = \alpha \cdot \tilde{S}_{CF}(u, i) + (1 - \alpha) \cdot \tilde{S}_{CB}(u, i)$$

Trong đó:

- $\alpha \in [0,1]$ : Hệ số trọng số lai ghép.
- $\tilde{S}_{CF}(u,i)$ : Điểm số dự đoán từ mô hình ALS đã được chuẩn hóa.
- $\tilde{S}_{CB}(u,i)$ : Điểm số tương đồng từ mô hình Content-based đã được chuẩn hóa.

Do điểm dự đoán từ mô hình lọc cộng tác dựa trên ALS được tính thông qua tích vô hướng và có miền giá trị không bị chặn  $(-\infty, +\infty)$ , trong khi độ tương đồng cosine trong mô hình lọc theo nội dung có miền giá trị  $[0,1]$ , các điểm số này không thể được kết hợp trực tiếp. Vì vậy, cần áp dụng phương pháp chuẩn hóa Min–Max để đưa các điểm dự đoán về cùng một miền giá trị trước khi thực hiện phép kết hợp.

Cụ thể, điểm số  $S(u,i)$  được chuẩn hóa như sau:

$$\tilde{S}(u,i) = \frac{S(u,i) - \min(S)}{\max(S) - \min(S)}$$

Việc chuẩn hóa này đảm bảo các thành phần trong mô hình lai ghép có mức độ đóng góp tương đương, từ đó giúp danh sách gợi ý cuối cùng vừa phản ánh xu hướng tương tác chung của cộng đồng (through qua lọc cộng tác), vừa phù hợp với sở thích nội dung cá nhân của từng người dùng (through qua lọc theo nội dung).

### 3.2.4. Lý thuyết đánh giá độ khó động

#### 3.2.4.1. Nguyên lý hoạt động

Trong các hệ thống truyền thống, độ khó của bài tập thường được gán tĩnh bởi tác giả, dẫn đến tính chủ quan và không phản ánh đầy đủ mức độ khó thực tế trong quá trình sử dụng. Để khắc phục hạn chế này, hệ thống áp dụng cơ chế điều chỉnh độ khó của bài tập một cách động dựa trên hành vi giải bài của người dùng.

Mỗi lần người dùng nộp bài được xem như một sự kiện đánh giá độ khó của bài tập. Dựa trên kết quả nộp bài (đúng hoặc sai), hệ thống cập nhật rating độ khó của bài tập. Việc cập nhật này cho phép độ khó của bài tập phản ánh sát hơn mức độ thử thách thực tế, từ đó hỗ trợ hiệu quả cho các chức năng gợi ý và phân phối bài tập phù hợp.

#### 3.2.4.2. Mô hình toán học

Giả sử user có rating năng lực  $R_u$  và problem có rating độ khó hiện tại  $R_p$ .

**Bước 1:** Tính xác suất thắng kỳ vọng (Expected Score): xác suất để user giải được bài tập (user thắng):

$$E_u = \frac{1}{1 + 10^{\frac{R_p - R_u}{400}}}$$

Xác suất để user không giải được problem:  $E_p = 1 - E_u$

**Bước 2:** Xác định kết quả thực tế ( $S_p$ ) và cập nhật rating của problem: kết quả khi user giải bài tập:

- Nếu user giải đúng (Accepted):  $S_p = 0$ .
- Nếu user giải sai (Wrong Answer/TLE):  $S_p = 1$ .

Công thức được sử dụng để cập nhật rating của problem:

$$R'_p = R_p + K \cdot (S_p - E_p)$$

Trong đó:

- $K$ : hệ số K-factor, quyết định mức độ biến động của rating sau mỗi lần nộp.
- $(S_p - E_p)$ : Sai số giữa thực tế và kỳ vọng.

### 3.3. Thiết kế chi tiết và thực hiện cài đặt

#### 3.3.1. Dữ liệu và tiền xử lý

##### 3.3.1.1. Bộ dữ liệu thực nghiệm

Hệ thống được huấn luyện và đánh giá trên bộ dữ liệu thực tế được thu thập từ nền tảng, bao gồm:

- Users: ~2,000 người dùng với mức Elo Rating trải dài từ 800 đến 3500 (trung bình ~1522).
- Problems: 2,915 bài tập được gán nhãn tags và rating ban đầu.
- Submissions: 98,997 lượt nộp bài, trong đó bao gồm các trạng thái: AC (Accepted), WA (Wrong Answer), TLE, MLE, ...

3.3.1.2. Phân tích độ thưa (Sparsity Analysis): Ma trận tương tác User-Item có kích thước  $2000 \times 2915$

- Tổng số ô dữ liệu (Cells): 5,830,000.
- Số lượng tương tác thực tế (Non-zero entries): ~95,000.
- Độ thưa (Sparsity): ~98.37%.

**Nhận xét:** Độ thua dữ liệu rất cao là lý do chính đáng để lựa chọn mô hình Matrix Factorization (ALS) thay vì các phương pháp dựa trên láng giềng truyền thống.

### 3.3.1.3. Kỹ thuật trích xuất đặc trưng (Feature Engineering)

#### a. Xử lý phản hồi ẩn

Hệ thống chuyển đổi lịch sử nộp bài thành điểm trọng số tương tác  $r_{ui}$  theo công thức logic mờ:

$$r_{ui} = \begin{cases} 1 \\ \min(0.2 + 0.1 \times total\_attempts, 0.8) \end{cases}$$

- $r_{ui}=1$  nếu problem đã được giải AC bởi người dùng
- $r_{ui}$  bằng giá trị còn lại nếu problem chưa được AC. Lý do không sử dụng giá trị 0 là vì người dùng có thể làm không đúng nhưng người dùng vẫn quan tâm đến problem chứ không phải là không quan tâm. Độ quan tâm này được đánh giá thông qua số lượng submission của người dùng cho 1 problem.

#### b. Vector hóa bài tập

Mỗi bài tập được biểu diễn bởi một vector đặc trưng kết hợp:

- Tags: Sử dụng One-hot encoding cho các loại tags (DP, Greedy, Graph, ...).
- Rating: Sử dụng Min-Max Scaling để chuẩn hóa về [0, 1]. Đặc biệt, đặc trưng rating được nhân trọng số RATING\_WEIGHT=10 để tăng tầm quan trọng của độ khó trong việc tìm kiếm bài tập tương đồng.

### 3.3.2. Cài đặt thuật toán và chiến lược lai ghép

#### 3.3.2.1. Cấu hình các mô hình

Hệ thống sử dụng các thư viện mã nguồn mở tối ưu cho hiệu năng:

- Collaborative Filtering: sử dụng thư viện implicit để triển khai thuật toán ALS
  - o Tham số: factors=30 (số chiều ẩn), iterations=15, regularization=0.1.
  - o Confidence Weighting:  $\alpha = 40$ . Đầu vào là ma trận  $C = 1 + 40 \times R$ .

```

# Khởi tạo mô hình ALS
als_model = implicit.als.AlternatingLeastSquares(
    factors=N_LATENT_FACTORS,
    regularization=0.1,
    iterations=15,
    random_state=SEED
)

# Áp dụng confidence weighting
ALPHA_VAL = 40
data_conf = (user_item_matrix * ALPHA_VAL).astype('double')

als_model.fit(data_conf)

```

Hình 3.1 Model ALS

- Content-based Filtering: Sử dụng thư viện scikit-learn với thuật toán KNN
  - o Tham số: n\_neighbors=15, metric='cosine'.

```

knn_model = NearestNeighbors(n_neighbors=KNN_NEIGHBORS, metric='cosine', algorithm='brute')
knn_model.fit(item_features)

```

Hình 3.2 Model KNN

### 3.3.2.2. Chiến lược lai ghép cải tiến

Ngoài việc kết hợp tuyênlính giữa CF và CB, hệ thống đẽ xuất thêm thành phần Rating Matching Boost để giải quyết vấn đề chênh lệch trình độ.

Cơ chế Rating Matching Boost: hệ thống tính toán độ lệch:  $\Delta = \text{Problem\_Rating} - \text{User\_Elo}$

- Vùng tối ưu:  $-100 \leq \Delta \leq 50$  (bài vừa sức)  $\rightarrow$  Boost score = 1.0.
- Vùng thử thách:  $50 < \Delta \leq 200$  (bài hơi khó)  $\rightarrow$  Boost score giảm dần từ 0.9.
- Vùng phạt:  $\Delta < -100$  (quá dễ) hoặc  $\Delta > 400$  (quá khó)  $\rightarrow$  Boost score thấp (< 0.4).

Công thức tổng quát:

$$Score_{final} = (1-w) \times \left[ \alpha \cdot Score_{CF\_norm} + (1-\alpha) \cdot Score_{CB\_norm} \right] + w \times Score_{RatingBoost}$$

Trong đó:

- $\alpha = 0.6$  (trọng số ưu tiên CF).
- $w = 0.1$  (trọng số ảnh hưởng của Rating Boost).

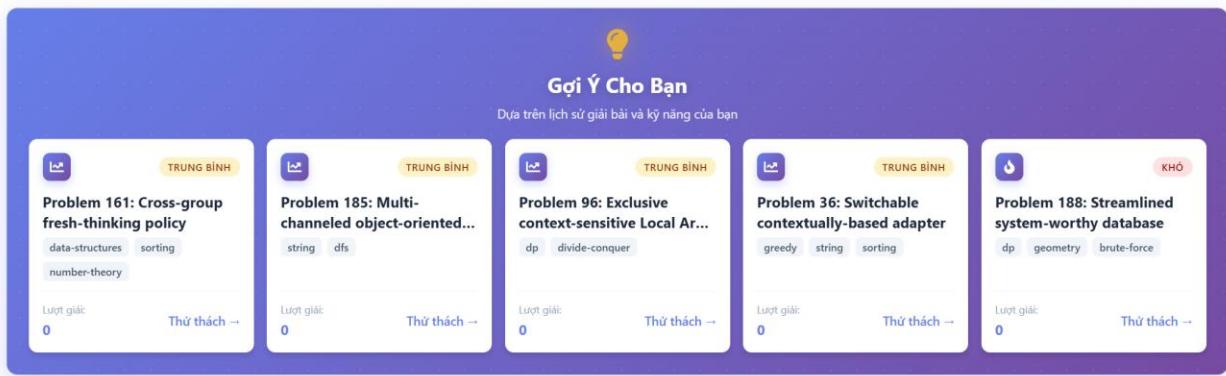
### 3.3.3. Kết quả thực nghiệm

- Hệ thống đđã đưa ra đđược các bài tập gợi ý cho người dùng dựa trên các bài tập mà người dùng đđã làm với mức độ độ khó vừa phải phù hợp với người dùng.

[TEST GỌI Ý CHO USER - IMPROVED]										
User ID: 241										
Username: user_0241										
Elo Rating: 1454										
Số bài đã AC: 280										
Tổng số submissions: 615										
Rank	Prob ID	CF	CB	Rating	Final	P.Rating	Diff	Tags		
1	1637	0.9455	0.9327	1.0000	0.9583	1400	-54	*special, greedy, imple...		
2	2151	0.8180	0.9987	1.0000	0.9232	1400	-54	constructive algorithm...		
3	1422	0.7854	0.9978	1.0000	0.9093	1400	-54	constructive algorithm...		
4	2841	0.7744	1.0000	1.0000	0.9052	1400	-54	greedy, implementation...		
5	2541	0.7676	1.0000	1.0000	0.9024	1500	+46	brute force, implementa...		
6	573	0.8038	0.9442	1.0000	0.9020	1500	+46	binary search, data str...		
7	2397	0.7654	0.9978	1.0000	0.9008	1400	-54	constructive algorithm...		
8	1430	0.7876	0.9571	1.0000	0.8988	1400	-54	greedy, implementation...		
9	2059	0.7584	0.9990	1.0000	0.8982	1500	+46	brute force, implementa...		
10	2144	0.7445	0.9979	1.0000	0.8921	1400	-54	constructive algorithm...		

Hình 3.3 Kết quả chạy mô hình

- Kết quả hiển thị gợi ý trên giao diện:



Hình 3.4 Kết quả gợi ý trên giao diện

# CHƯƠNG 4: TRIỂN KHAI HỆ THỐNG

## 4.1. Backend

### 4.1.1. Cấu trúc thư mục

```
backend/
  └── backend/          # Django project configuration
      ├── __init__.py
      ├── settings.py     # Cấu hình toàn hệ thống
      ├── urls.py         # Root URL routing
      ├── wsgi.py          # WSGI entry point
      └── asgi.py          # ASGI entry point (async support)

  └── common/           # Shared utilities
      ├── authentication.py # Custom JWT Authentication
      ├── connection.py    # Database connection utils
      ├── media_views.py   # File upload/download handlers
      └── urls.py

  └── users/            # User management module
      ├── models.py        # User, Role, Permission models
      ├── views.py         # Authentication, profile APIs
      ├── serializers.py   # Data serialization
      ├── rating_service.py # Elo rating calculation
      ├── public_profile_views.py
      ├── user_reports_views.py
      └── urls.py

  └── contests/          # Contest management module
      ├── models.py        # Contest, ContestProblem, ContestParticipant
      ├── views.py         # Contest CRUD, leaderboard APIs
      ├── serializers.py
      ├── ranking_service.py # Leaderboard calculation (ICPC/OI mode)
      ├── domjudge_service.py # DOMjudge API integration
      └── urls.py

  └── problems/          # Problem management module
      ├── models.py        # Problem, TestCase, Submissions
      ├── views.py         # Problem CRUD, submission APIs
      ├── serializers.py
      ├── domjudge_service.py # Problem sync to DOMjudge
      ├── user_profile_views.py
      └── urls.py
```

```

course/           # Course & Payment module
    ├── models.py      # Course, Lesson, Enrollment, Order
    ├── views.py       # Course CRUD, enrollment APIs
    ├── serializers.py
    ├── vnpay_service.py   # VNPay payment integration
    ├── revenue_views.py   # Revenue statistics
    ├── course_reports_views.py
    └── urls.py

quizzes/          # Quiz module
    ├── models.py      # Quiz, QuizQuestion, QuizAttempt
    ├── views.py
    ├── serializers.py
    └── urls.py

media/            # User-uploaded files
    ├── images/avatars/
    └── files/uploads/

static/           # Static files (CSS, JS, images)

docker-compose.yml    # DOMjudge container orchestration
requirements.txt      # Python dependencies
manage.py            # Django management CLI
.env                 # Environment variables

```

#### **4.1.2. Mô hình kiến trúc backend django**

Backend Django được tổ chức theo Layered Architecture, kết hợp mô hình MVT và Service Layer:

- Mô hình MVT:
  - Model định nghĩa các cấu trúc bảng dữ liệu, thiết lập quan hệ giữa các thực thể, ánh xạ dữ liệu giữa Python Object và CSDL thông qua Django ORM
  - View đóng vai trò như Controller tiếp nhận HTTP request từ client, gọi lớp xử lý nghiệp vụ và trả về kết quả cho client
  - API (Json Response) thay thế cho Template để trả về dữ liệu dạng json cho frontend xử lý (Django REST Framework)
- Service Layer:
  - Ranking Service: tính điểm dùng cho bảng xếp hạng contest và bảng xếp hạng toàn cầu
  - Domjudge Service: thực hiện các nghiệp vụ liên quan tới hệ thống chấm bài Domjudge
  - VNPay Service: thực hiện các nghiệp vụ thanh toán khóa học

#### **4.1.3. Tích hợp với hệ thống bên ngoài**

- a. Domjudge

- Mục đích: Chấm bài tự động
  - Cơ chế:
    - Django Backend gọi API để thực hiện các chức năng thêm contest, problems, nộp bài để chấm tự động.
    - Query trực tiếp Domjudge Database Server để lấy dữ liệu cần thiết gồm các submissions, kết quả chấm bài (read only)
  - Các thao tác chính:
    - Tạo contest
    - Tạo problems và testcases
    - Nộp bài
    - Lấy kết quả chấm bài
    - Lấy chi tiết từng kết quả chấm testcase của submissions
- b. *VNPay*
- Mục đích: Thanh toán khóa học trực tuyến
  - Cơ chế: Sau khi người dùng chọn đăng ký khóa học, ở phía backend sẽ tạo order ở trạng thái pending và gọi VNPAY Service, VNPAY Service chuyển hướng cho user đến trang thanh toán của VNPAY, sau khi người dùng thanh toán, VNPAY sẽ gọi API ở backend đã được cấu hình trước để xác nhận trạng thái thanh toán, backend xử lý và trả về kết quả cho người dùng.

#### **4.1.4. Triển khai**

- Chuẩn bị môi trường:
  - Clone repository trên github đồ án
  - Cd backend
  - Copy example.env .env
- Build domjudge và lấy admin password
  - docker-compose -f docker-compose.production.yml up -d db domserver
  - lấy password khi domjudge khởi tạo bằng lệnh docker logs domjudge\_server 2>&1 | Select-String "Initial admin password"
- Cấu hình lại file .env
- Build toàn bộ hệ thống: docker-compose -f docker-compose.production.yml up -d
- Khởi tạo dữ liệu
  - docker exec django\_backend python manage.py migrate
  - docker exec django\_backend python manage.py init\_permissions

### **4.2. Frontend**

#### **4.2.1. Tổng quan**

##### **4.2.1.1. Công nghệ sử dụng**

- React Framework
  - **React 19.1.1:** Framework JavaScript chính để xây dựng giao diện người dùng
  - **React DOM 19.1.1:** Thư viện render React components
  - **React Scripts 5.0.1:** Bộ công cụ build và development từ Create React App
  - **React Router DOM 7.9.3:** Quản lý routing và điều hướng trong ứng dụng SPA
- Rich Text và Code Editor

- **CKEditor 5 (44.3.0)**: Editor WYSIWYG cho việc soạn thảo nội dung bài học, mô tả bài toán
- **Monaco Editor (4.7.0)**: Code editor mạnh mẽ (được sử dụng trong VS Code) cho tính năng code submission
- UI Components và Icons
  - **Lucide React (0.546.0)**: Bộ icon hiện đại và nhất quán
  - **React Icons (5.5.0)**: Thư viện icon phong phú với nhiều bộ icon phổ biến
  - **SweetAlert2 (11.26.2)**: Thư viện tạo modal và notification đẹp mắt, tương tác
- Data Visualization
  - **Chart.js (4.5.1)**: Thư viện vẽ biểu đồ JavaScript mạnh mẽ
- Utilities và Helpers
  - **Axios (1.12.2)**: HTTP client để gọi API
  - **File Saver (2.0.5)**: Lưu file từ client-side
  - **React Easy Crop (5.5.3)**: Component crop ảnh cho avatar/thumbnaill
  - **@hello-pangea/dnd (18.0.1)**: Drag and drop functionality (fork của react-beautiful-dnd)

#### 4.2.1.2. Cấu trúc thư mục

```

frontend/src/
  └── assets/      # Tài nguyên tĩnh (images, fonts)
  └── components/  # Shared components tái sử dụng
    ├── CKEditor.jsx  # WYSIWYG editor wrapper
    ├── CodeEditor.jsx # Monaco code editor wrapper
    ├── Pagination.jsx # Component phân trang
    ├── PermissionGuard.jsx # Guard kiểm tra permission
    ├── RoleGuard.jsx # Guard kiểm tra role
    └── RecommendedProblems.jsx # Gợi ý bài tập
  └── contexts/    # React Context API
    └── UserContext.jsx # Context quản lý trạng thái user
  └── layouts/     # Layout components
    ├── AdminLayout.jsx # Layout cho admin panel
    └── MainLayout.jsx # Layout chính cho user
    └── components/  # Header, Footer, Sidebar
  └── pages/       # Các trang của ứng dụng
    └── admin/       # Trang quản trị
      ├── contest/   # Quản lý contest
      ├── course/    # Quản lý khóa học
      ├── dashboard/ # Dashboard thống kê
      ├── language/  # Quản lý ngôn ngữ lập trình
      ├── lesson/    # Quản lý bài học
      ├── permission/ # Quản lý quyền
      ├── problem/   # Quản lý bài toán
      ├── quiz/      # Quản lý quiz
      └── role/      # Quản lý vai trò

```

```

    └── statistic/ # Thống kê
        └── user/ # Quản lý người dùng
            └── contests/ # Trang contest cho user
                ├── Contests.jsx
                ├── ContestDetailUser.jsx
                ├── ContestLeaderboard.jsx
                └── PracticeLeaderboard.jsx
            └── courses/ # Trang khóa học
                ├── Courses.jsx
                ├── CourseDetail.jsx
                ├── CourseLearning.jsx
                ├── QuizTaking.jsx
                └── QuizResult.jsx
        └── GlobalRanking/ # Bảng xếp hạng toàn cầu
        └── practice/ # Trang luyện tập
        └── problem/ # Chi tiết bài toán và submission
        └── PublicProfile/ # Profile công khai
        └── UserProfile/ # Profile cá nhân
    └── routes/ # Routing configuration
        ├── index.jsx # Route definitions
        └── PrivateRoute.jsx # Protected route wrapper
    └── services/ # API service layer
        ├── api.js # Axios instance và interceptors
        ├── AuthService.js
        ├── ContestService.js
        ├── CourseService.js
        ├── LessonService.js
        ├── PaymentService.js
        ├── ProblemService.js
        ├── QuizService.js
        ├── RatingService.js
        ├── RoleService.js
        ├── SubmissionService.js
        └── UserService.js
    └── utils/ # Utility functions
        └── codeTemplates.js # Templates code cho các ngôn ngữ
    └── notification.js # Helper functions cho notifications

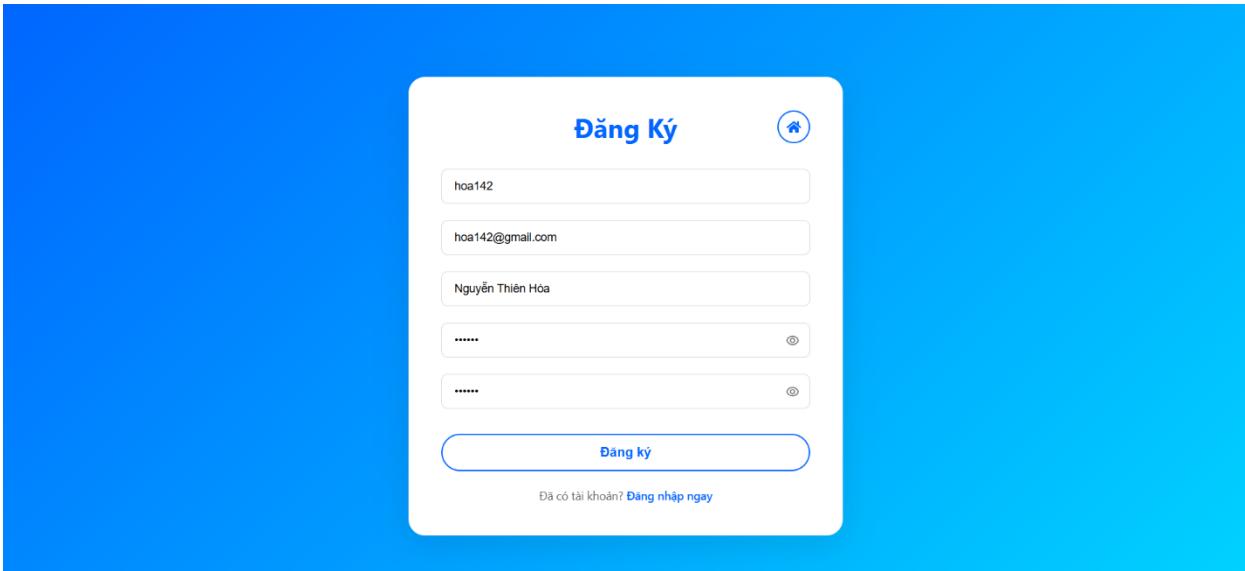
```

#### 4.2.1.3. Triển khai

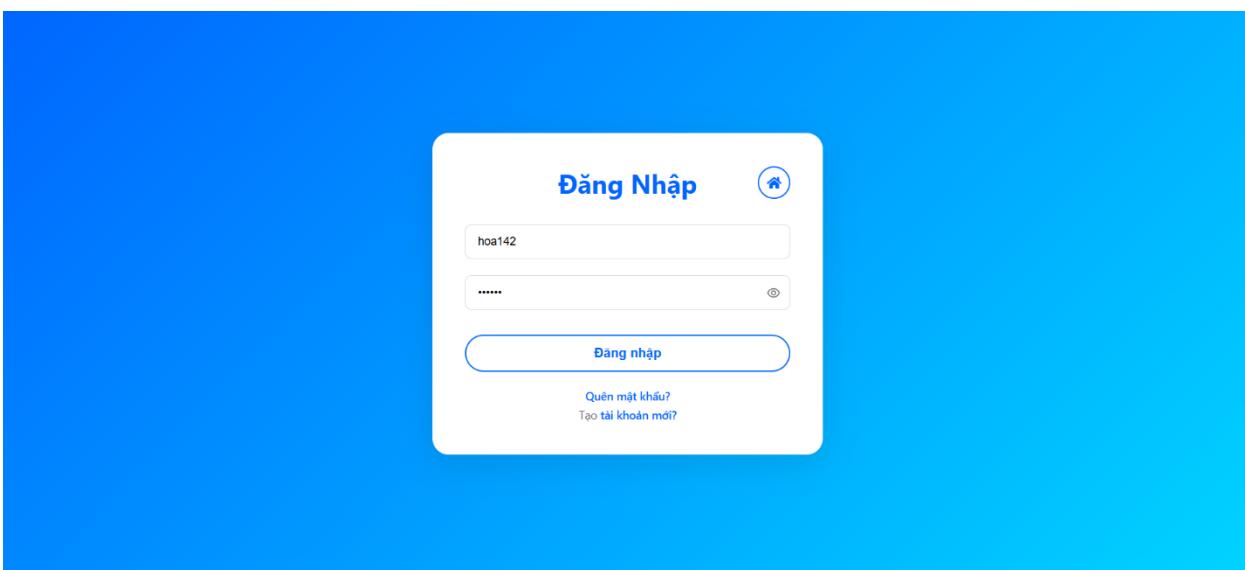
Build docker cùng backend server ở trên

#### 4.2.2. Giao diện chức năng chính

##### 4.2.2.1. Người dùng



Hình 4.1 Giao diện chức năng đăng ký tài khoản



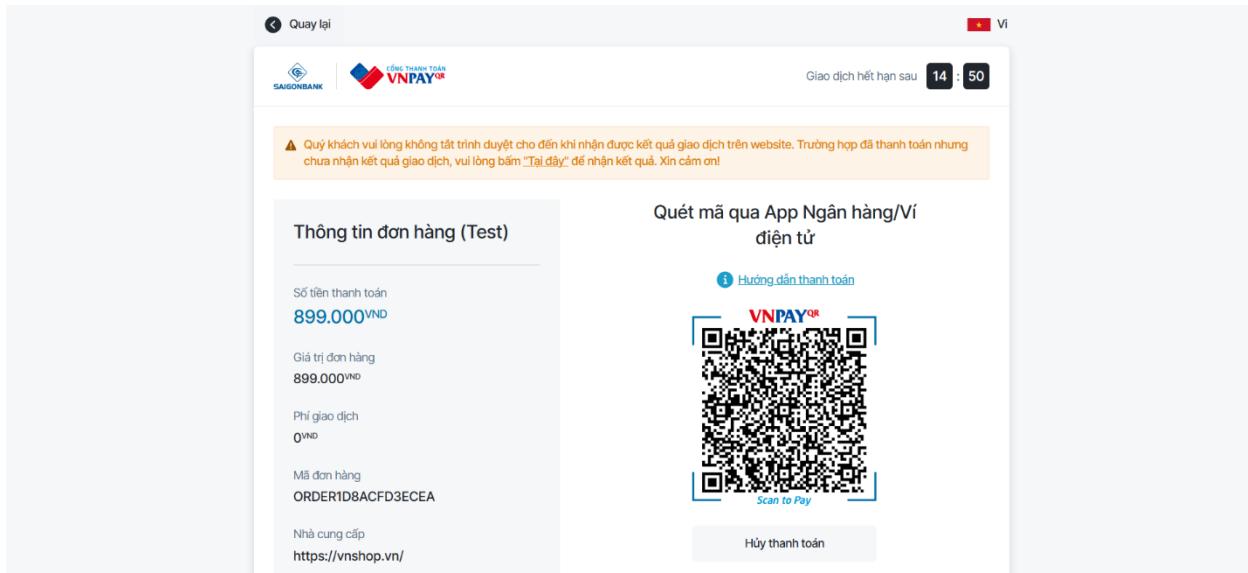
Hình 4.2 Giao diện chức năng đăng nhập

The screenshot shows the CodeLearn user profile interface. At the top, there are four summary cards: 'BÀI ĐÃ GIẢI' (0), 'CONTEST THAM GIA' (0), 'TỶ LỆ AC' (0%), and 'LUỢT NỘP BÀI' (2). Below these are navigation tabs: 'Tổng quan' (selected), 'Khóa học', 'Bài tập', 'Lịch sử nộp', and 'Contest'. The main area is divided into two sections: 'Thông tin cá nhân' (Personal Information) and 'Thống kê chi tiết' (Statistics). The personal information section lists fields like Username, Email, Họ tên, Giới tính, Ngày sinh, Số điện thoại, and Địa chỉ, all with placeholder values. The statistics section shows metrics: Tổng số lần nộp (2), Số bài AC (0), Bài đã giải (0), Contest tham gia (0), and Tỷ lệ AC (0%).

Hình 4.3 Giao diện chức năng hồ sơ người dùng

The screenshot shows the CodeLearn course catalog page. The header includes the logo, navigation tabs (Khóa học, Thực hành, Thi đấu, Bảng xếp hạng), and a user profile for Nguyễn Thiên Hòa. The main content features three course cards: 'Java nâng cao & Lập trình thi đấu' (Advanced), 'Lập trình C/C++ cơ bản' (Beginner), and 'Thuật toán & Cấu trúc dữ liệu với C++' (Intermediate). Each card includes a thumbnail, title, description, author (admin), number of lessons (0), and tags.

Hình 4.4 Giao diện chức năng danh sách khóa học



Hình 4.5 Giao diện chức năng đăng ký khóa học

The screenshot shows the CodeLearn platform's course registration interface. At the top, there are navigation links: "Khóa học", "Thực hành", "Thi đấu", and "Bảng xếp hạng". A user profile "Nguyễn Thiên Hòa" is on the right. Below, a breadcrumb trail shows "← Quay lại khóa học / Lập trình C/C++ cơ bản". The main content area has tabs "Nội dung" and "Lịch sử". Under "Giới thiệu C/C++", there is a section titled "Giới thiệu về C và C++" with the text: "C và C++ là hai ngôn ngữ lập trình phổ biến, được sử dụng rộng rãi trong phát triển phần mềm hệ thống, game, trình biên dịch và các ứng dụng hiệu năng cao." (C and C++ are two widely used programming languages, commonly used in system software development, games, compilers, and high-performance applications). Other sections include "Lịch sử phát triển" (Development history) and "Ứng dụng thực tế" (Practical applications).

Hình 4.6 Giao diện chức năng vào học

**Gợi ý Cho Bạn**

Dựa trên lịch sử giải bài và kỹ năng của bạn

- Đếm chữ số** ĐỄ  
math  
Lượt giắc: 0      [Thử thách →](#)
- Kiểm tra năm nhuận** ĐỄ  
math  
Lượt giắc: 0      [Thử thách →](#)
- Tổng từ 1 đến n** ĐỄ  
math  
Lượt giắc: 0      [Thử thách →](#)
- Giai thừa nhỏ** ĐỄ  
math  
Lượt giắc: 0      [Thử thách →](#)
- Đếm số phần tử dương** ĐỄ  
math  
Lượt giắc: 0      [Thử thách →](#)

Trạng thái:

[Tất cả](#)[Đã giải](#)[Chưa giải](#)[Tất cả](#)

Độ khó:

[ĐỄ](#)[Trung bình](#)[Khó](#)

TRANG THÁI TÊN BÀI

[Bảng cứu chương](#)

e10

ĐỘ KHÓ

[Trung Bình](#)

0

*Hình 4.7 Giao diện chức năng luyện tập***Đề bài**

tim hai số có tổng bằng n

**Định dạng đầu vào**

số nguyên n

**Định dạng đầu ra**

in ra 2 số nguyên có tổng bằng n

**Ví dụ**

Ví dụ 1

INPUT:

10

OUTPUT:

## Nhập code của bạn:

```

1 # Python Solution
2 def solve():
3     # Write your code here
4     pass
5
6 if __name__ == "__main__":
7     solve()
8

```

Python 3 • Light theme

[Code Editor](#)[Upload File](#)*Hình 4.8 Giao diện chức năng làm bài lập trình*

Practice Contest · practice



Top	Người dùng	AC	Đã thử
#1	Admin	2	4
#2	Amanda Harris	0	0
#3	Cheryl Cox	0	0
#4	Christopher Clark	0	0
#5	Christopher Gilbert	0	0
#6	Claire Steele	0	0
#7	Cynthia Larson	0	0

Hình 4.9 Giao diện chức năng bảng xếp hạng luyện tập

## 🏆 Cuộc thi lập trình

Tham gia các cuộc thi để thử thách bản thân và cạnh tranh với các lập trình viên khác

### ● Đang diễn ra

2 cuộc thi

Tên Cuộc thi	Bắt đầu	Kết thúc	Tham gia
Kỳ thi OI – Lập trình cơ bản	13/12/2025	23/01/2026	Tham gia ngay
ICPC – Thi thử cơ bản	13/12/2025	16/01/2026	Tham gia ngay

Hình 4.10 Giao diện chức năng cuộc thi lập trình

The screenshot shows a contest summary at the top with icons for start time (17:22 13/12/2025), end time (17:22 16/01/2026), solved problems (5 bài), and penalty (20 phút). Below this is a navigation bar with 'Danh sách bài tập' and 'Bảng xếp hạng' (selected). The 'Bảng xếp hạng' section displays a local leaderboard for 2 participants:

Hạng	Thành viên	Solved	Penalty	A	B	C	D	E
1	Admin	1	153	93 +4 tries	-	-	-	-
2	Nguyễn Thiên Hòa	0	0	-	-	-	-	-

Hình 4.11 Giao diện chức năng bảng xếp hạng cuộc thi

The screenshot shows a global ranking table titled 'Bảng Xếp Hạng Global'. It includes a sidebar with 'Thứ hạng của bạn' (#71, SPECIALIST, 1500 / 1500 max, 0 contests). The main table lists three users:

Hạng	Người dùng	Rank	Rating	Max Rating	Contests
#51	Cheryl Cox	EXPERT	1745	2674	44 🏆
#52	Maureen Freeman	EXPERT	1730	1658	34 🏆
#53	Travis Harris	EXPERT	1700	2389	37 🏆

Hình 4.12 Giao diện chức năng bảng xếp hạng toàn cầu

#### 4.2.2.2. Quản trị

**Dashboard Quản trị**

Tổng quan về hệ thống quản lý khóa học

Thể loại	Số lượng	Mô tả
TỔNG KHÓA HỌC	21	20 đã xuất bản
TỔNG BÀI HỌC	2	Tất cả khóa học
TỔNG ĐĂNG KÝ	305	Học viên tham gia
TÀI NGUYỄN	1	Video, PDF, Slide...
THỂ PHẢN LOẠI	36	Tags cho khóa học
TỔNG CONTEST	25	0 đang diễn ra
PRACTICE PROBLEMS	2	0 bài nộp
THÍ SINH CONTEST	365	Đã đăng ký tham gia

**Khóa học gần đây**

- Angular Complete** Learn Angular Complete Guide from scratch 25/11/2025 | 0 bài | 18 học viên
- MongoDB Complete Course** Learn MongoDB Complete Course from scratch 21/11/2025 | 0 bài | 15 học viên

**Thao tác nhanh**

- Quản lý khóa học** Tạo và chỉnh sửa khóa học
- Quản lý bài học** Tạo và chỉnh sửa bài học

Hình 4.13 Giao diện trang dashboard

**Quản lý Khóa học**

Quản lý các khóa học lập trình và tài liệu học tập

+ Tạo khóa học mới

ID	TIÊU ĐỀ	CẤP ĐỘ	NGÔN NGỮ	BÀI HỌC	HỌC VIÊN	TRANG THÁI	THAO TÁC
28	Angular Complete Guide	Trung bình	Java	0	18	X Nhập	[Edit, Delete]
40	MongoDB Complete Course	Trung bình	JavaScript, Java	0	15	Đã xuất bản	[Edit, Delete]
37	Algorithm Design	Trung bình	JavaScript, Go	0	11	Đã xuất bản	[Edit, Delete]
1	Lập trình C++	Cơ bản	C++	2	1	Đã xuất bản	[Edit, Delete]
22	Python for Beginners	Nâng cao	Go	0	21	Đã xuất bản	[Edit, Delete]
41	PostgreSQL Advanced	Cơ bản	Java	0	18	Đã xuất bản	[Edit, Delete]
29	Vue.js Essential Training	Trung bình	Java, C#	0	13	Đã xuất bản	[Edit, Delete]
32	Flask REST API	Nâng cao	Rust	0	15	Đã xuất bản	[Edit, Delete]
36	CI/CD Pipeline	Trung bình	C++	0	14	Đã xuất bản	[Edit, Delete]
38	System Design Interview	Trung bình	Rust	0	21	Đã xuất bản	[Edit, Delete]
39	Database Design	Trung bình	Go, Rust	0	10	Đã xuất bản	[Edit, Delete]

Hình 4.14 Giao diện quản lý khóa học

Tạo khóa học mới

Thông tin cơ bản

Banner khóa học

Nhập để tải lên banner  
JPG, PNG (tối đa 5MB)

Tiêu đề \*

Slug \*

Mô tả ngắn

Mô tả chi tiết

Hủy

Hình 4.15 Giao diện tạo mới khóa học

Quản lý Bài học

+ Tạo bài học mới

ID	TIÊU ĐỀ	KHÓA HỌC	THỦ TỤC	RESOURCES	THAO TÁC
#3	Bài test	System Design Interview	0	0	
#4	Tim hiểu về REST API	Flask REST API	0	0	
#5	Tim hiểu kiến thức Java	Data Structures in Java	0	0	
#1	Bài 1	Lập trình C++	1	1	
#2	Python Variables and Data Types Learn about variables and data types in Python	Lập trình C++	2	0	

Hình 4.16 Giao diện quản lý bài học

**Tạo bài học mới**

**Thông tin bài học**

Khóa học  
-- Không gắn khóa học --

Tiêu đề \*  
Nhập tiêu đề bài học

Mô tả  
Mô tả nội dung bài học

Thứ tự  
0

**Tài nguyên bài học**

+ Thêm tài nguyên

X Hủy | Lưu bài học

Hình 4.17 Giao diện tạo mới bài học

**Quản lý bài toán lập trình**

Tổng số: 210

ID	SLUG	TIÊU ĐỀ	ĐỘ KHÓ	TEST CASES	TRANG THÁI	DOMJUDGE	THAO TÁC
219	problem-0200	Problem 200: Organic even-keeled instruction set Source recognize new south health.	Dễ	10	Công khai  Chưa sync		
218	problem-0199	Problem 199: Persistent stable approach Source traditional central until start.	Khó	11	Công khai  Chưa sync		
217	problem-0198	Problem 198: Multi-channelled eco-centric time-frame Despite court book top decision clearly.	Trung bình	12	Công khai  Chưa sync		
216	problem-0197	Problem 197: Persevering secondary methodology Better behind audience change.	Khó	12	Công khai  Chưa sync		
215	problem-0196	Problem 196: Right-sized non-volatile utilization Play any till site pressure wall while.	Khó	12	Công khai  Chưa sync		
214	problem-0195	Problem 195: Optimized disintermediate projection Unit arrive him rest worker memory.	Dễ	14	Công khai  Chưa sync		
213	problem-0194	Problem 194: Versatile bi-directional hardware Play apply expect even material stop.	Khó	13	Công khai  Chưa sync		
212	problem-0193	Problem 193: Fully-configurable coherent projection	Trung bình	13	Công khai  Đã sync		

Hình 4.18 Giao diện quản lý bài tập lập trình

Tổng quan

Quản lý khóa học

- Danh sách khóa học
- Quản lý bài học

Bài tập lập trình

- Danh sách bài tập
- Thêm bài tập mới
- Test cases

Thi & Quiz

Contest

- Danh sách Contest
- Tạo Contest mới
- Bảng xếp hạng

Người dùng

- Danh sách học viên
- Quản lý quyền
- Phân quyền
- Loại phân quyền

## Tạo bài toán mới

Thêm bài toán lập trình mới vào hệ thống

### Thông tin cơ bản

Mã bài \*

Độ khó \*

Tối đa 10 ký tự, chỉ chữ thường và số (a-z, 0-9) - Ví dụ: twosum, dp01

Tiêu đề \*

Mô tả ngắn

Tìm hai số trong mảng có tổng bằng target

Tối đa 512 ký tự

Nguồn

---

### Đề bài

Nội dung đề bài \*

Paragraph

Nhập nội dung đề bài chi tiết...

Hình 4.19 Giao diện tạo bài tập lập trình

Tổng quan

Quản lý khóa học

- Danh sách khóa học
- Quản lý bài học

Bài tập lập trình

- Danh sách bài tập
- Thêm bài tập mới
- Test cases

**Thi & Quiz**

- Danh sách Quiz**
- Tạo Quiz mới

Contest

- Danh sách Contest
- Tạo Contest mới
- Bảng xếp hạng

Người dùng

- Danh sách học viên

## Quản lý Quiz

Quản lý danh sách các bài thi trắc nghiệm

+ Tạo Quiz mới

Tất cả trạng tháiTổng số: 3

ID	TIÊU ĐỀ	SỐ CÂU HỎI	THỜI GIAN	TRẠNG THÁI	NGƯỜI TẠO	NGÀY TẠO	THAO TÁC
3	test	2	60 phút	Đã công khai	Kiều Hiếu	30/11/2025	
2	Sample Quiz 1	2	30 phút	Đã công khai	Kiều Hiếu	28/11/2025	
1	Updated Quiz Title	1	40 phút	Đã công khai	Kiều Hiếu	28/11/2025	

Hình 4.20 Giao diện quản lý Quiz

**Tạo Quiz mới**

Điền thông tin để tạo quiz mới

**Thông tin cơ bản**

**Tiêu đề \***  
Nhập tiêu đề quiz

**Mô tả**  
Paragraph | B | I | O | H1 | H2 | H3 | H4 | H5 | H6 | C | C2  
Nhập mô tả quiz...

**Thời gian (giây) \*** **Trạng thái**

← Quay lại

Hình 4.21 Giao diện tạo mới bài Quiz

**Contest Management**

Manage programming contests and competitions

+ Create Contest

CONTEST	THỜI GIAN	TRẠNG THÁI	HÌNH THỊ	BÀI TẬP	PENALTY	HÀNH ĐỘNG
LeetCode Weekly #6 leetcode-weekly-6-6	10:52 09/12/2025 12:52 09/12/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
Speed Coding Challenge #12 speed-coding-challenge-12-12	10:52 03/12/2025 16:52 03/12/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
Competitive Programming Round #18 competitive-programming-round-18-18	10:52 30/11/2025 15:52 30/11/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
LeetCode Weekly #4 leetcode-weekly-4-4	10:52 27/11/2025 16:52 27/11/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
HackerRank Challenge #24 hackerrank-challenge-24-24	10:52 24/11/2025 15:52 24/11/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
Practice practice	14:23 15/11/2025 14:23 27/11/2027	RUNNING	public	2	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
Weekly Contest #20 weekly-contest-20-20	10:52 13/11/2025 15:52 13/11/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
test1 my-contest	13:27 01/11/2025 13:27 22/11/2025	FINISHED	private	1	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
Competitive Programming Round #23 competitive-programming-round-23-23	10:52 25/10/2025 14:52 25/10/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>
Competitive Programming Round #19	10:52 23/10/2025	FINISHED	public	0	20 phút	<span style="color: green;">Xem</span> <span style="color: blue;">Tạo</span> <span style="color: red;">Xóa</span>

Hình 4.22 Giao diện quản lý Contest

Hình 4.23 Giao diện tạo mới Contest

ID	USERNAME	EMAIL	HỌ TÊN	VAI TRÔ	TRANG THÁI	THAO TÁC
1	testuser123	test@example.com	Kiều Văn Hiếu	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
2	kvh12345	hieukv26102003@gmail.com	Kiều Hiếu	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
3	admin	test@gmail.com	Nguyễn Thiên Hòa	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
4	hieu	hieu@gmail.com	Kiều Hiếu	admin	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
105	user_0001	user_0001@example.com	User 1	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
106	user_0002	user_0002@example.com	User 2	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
107	user_0003	user_0003@example.com	User 3	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
108	user_0004	user_0004@example.com	User 4	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
109	user_0005	user_0005@example.com	User 5	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>
110	user_0006	user_0006@example.com	User 6	Chưa có	<span style="color: green;">✓</span> Hoạt động	<span style="color: blue;">[Edit]</span> <span style="color: red;">[Delete]</span>

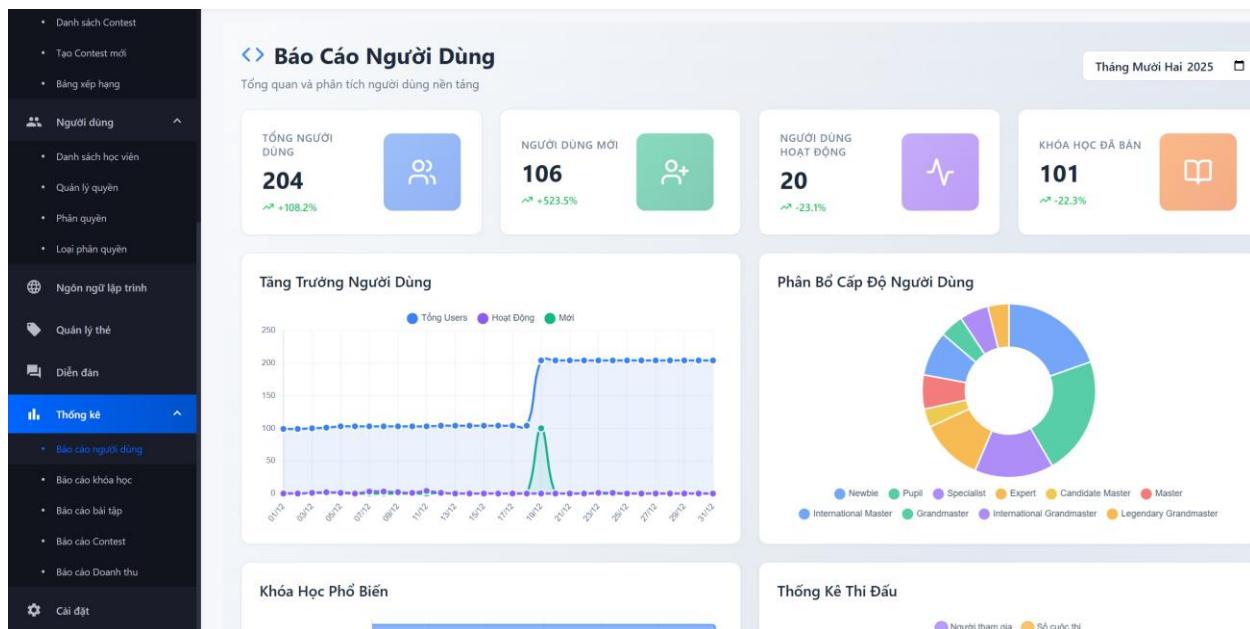
Hình 4.24 Giao diện quản lý người dùng

Hình 4.25 Giao diện tạo mới tài khoản người dùng

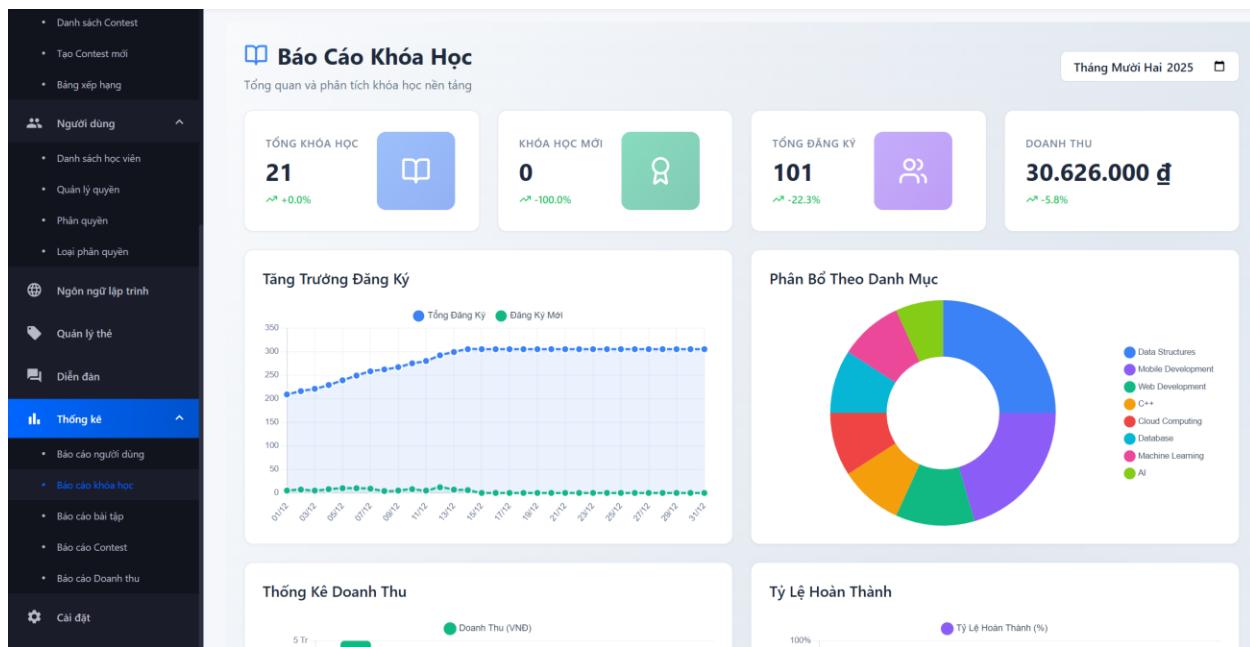
ID	TÊN VAI TRÒ	MÔ TẢ	SỐ QUYỀN	SỐ NGƯỜI DÙNG	THAO TÁC
1	admin	Quản trị viên	104	1	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>
2	user	User with limited permissions	27	0	<input checked="" type="checkbox"/> <input type="button" value="Delete"/>

Hình 4.26 Giao diện quản lý vai trò

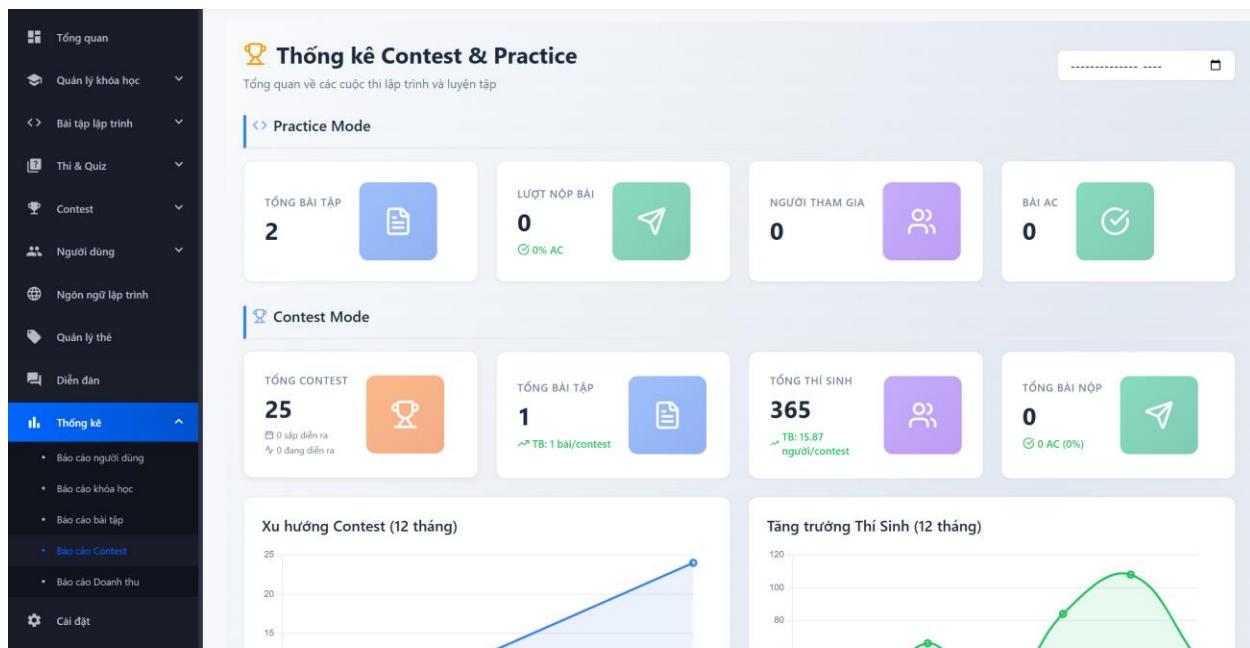
Hình 4.27 Giao diện tạo mới vai trò



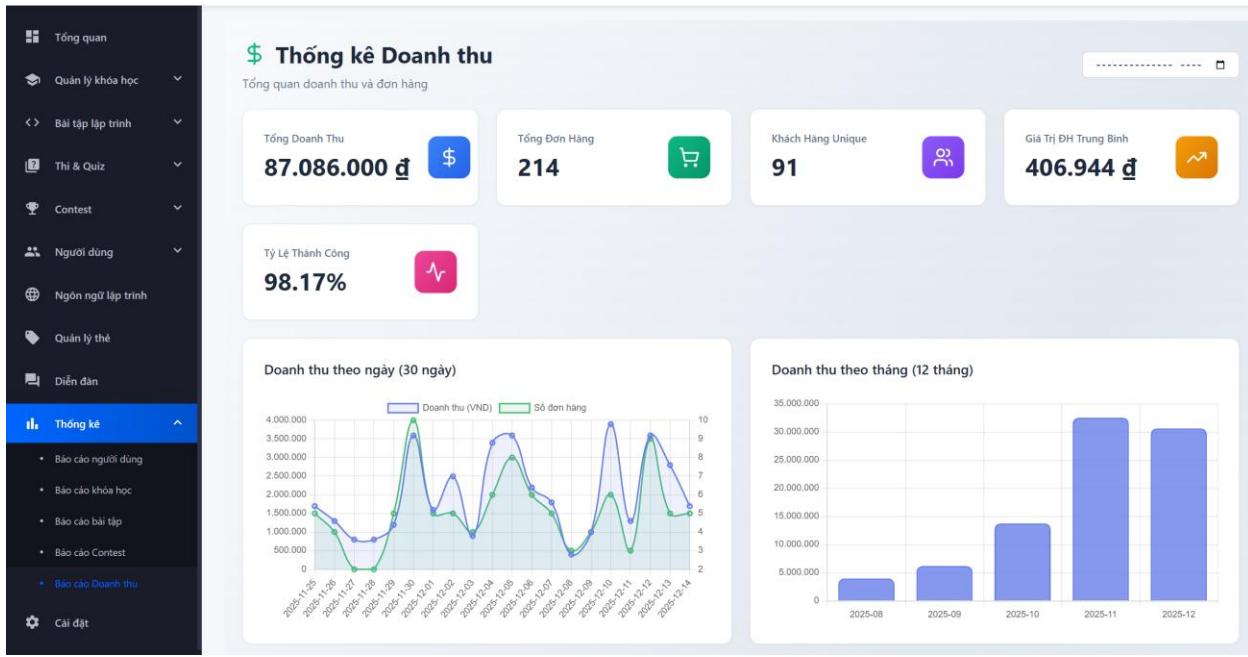
Hình 4.28 Giao diện thống kê người dùng



Hình 4.29 Giao diện thống kê khóa học



Hình 4.30 Giao diện thống kê Contest



Hình 4.31 Giao diện thống kê doanh thu

## **KẾT LUẬN**

## **TÀI LIỆU THAM KHẢO**

- [1] A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, O'Reilly Media., 2020.
- [2] N. M. Hùng and Đ. T. B. Ngọc, PHÂN TÍCH THIẾT KẾ ĐẦM BẢO CHẤT LUỢNG PHẦN MỀM, Hà Nội: PTIT.
- [3] "DOMjudge," [Online]. Available: <https://www.domjudge.org/>. [Accessed 2025].
- [4] Encode OSS Ltd, "Django REST framework," 2011. [Online]. Available: <https://www.django-rest-framework.org/>. [Accessed 2025].

