**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**ĐỒ ÁN TỐT NGHIỆP**

**Website học trực tuyến**

**PHẠM HỒNG SƠN**

son.ph173344@sis.hust.edu.vn

**Ngành Công nghệ thông tin**

**Chuyên ngành Công nghệ thông tin**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ThS. Vũ Đức Vượng |
| **Bộ môn:** | Công nghệ phần mềm |
| **Viện:** | Công nghệ thông tin – Truyền thông |
| **HÀ NỘI, 03/2021** | |

# Lời cam kết

Họ và tên sinh viên: Phạm Hồng Sơn

Điện thoại liên lạc: 0966389332 Email: son.ph173344@sis.hust.edu.vn

Lớp: CNTT-09 Hệ đào tạo: Cử nhân kỹ thuật

Tôi – *Phạm Hồng Sơn* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Vũ Đức Vương*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm 2021*  Tác giả ĐATN  *Phạm Hồng Sơn* |

# Mục lục

[Lời cam kết 1](#_Toc73323645)

[Mục lục 2](#_Toc73323646)

[Danh mục hình vẽ 5](#_Toc73323647)

[Danh mục bảng 7](#_Toc73323648)

[Danh mục các từ viết tắt 9](#_Toc73323649)

[Danh mục thuật ngữ 10](#_Toc73323650)

[Chương 1 Giới thiệu đề tài 11](#_Toc73323651)

[Chương 2 Khảo sát và phân tích yêu cầu 12](#_Toc73323652)

[2.1 Tổng quan chức năng 13](#_Toc73323653)

[2.1.1 Biểu đồ use case tổng quan 13](#_Toc73323654)

[2.1.2 Biểu đồ use case phân rã 15](#_Toc73323655)

[2.1.2.1 Biểu đồ use case phân rã Authenticate 15](#_Toc73323656)

[2.1.2.2 Biểu đồ use case phân rã Quản lý thông tin cá nhân 16](#_Toc73323657)

[2.1.2.3 Biểu đồ use case phân rã Xem danh sách khóa học 18](#_Toc73323658)

[2.1.2.4 Biểu đồ use case phân rã Tạo chương học 19](#_Toc73323659)

[2.1.2.5 Biểu đồ use case phân rã Lên lớp 21](#_Toc73323660)

[2.1.2.6 Biểu đồ use case phân rã Học 23](#_Toc73323661)

[2.1.2.7 Biểu đồ use case Nhắn tin 25](#_Toc73323662)

[2.1.2.8 Biểu đồ use case Quản lý khóa học 26](#_Toc73323663)

[2.1.2.9 Biểu đồ use case phân rã Quản lý người dùng 27](#_Toc73323664)

[2.1.2.10 Biểu đồ use case phân rã quản lý topic 28](#_Toc73323665)

[2.1.2.11 Biểu đồ use case phân rã thống kê hệ thống 29](#_Toc73323666)

[2.1.3 Quy trình nghiệp vụ 30](#_Toc73323667)

[2.1.3.1 Quy trình nghiệp vụ xác thực người dùng 30](#_Toc73323668)

[2.1.3.2 Quy trình nghiệp vụ tạo khóa học 31](#_Toc73323669)

[2.1.3.3 Quy trình nghiệp vụ học sinh tham gia phòng học trực tuyến 33](#_Toc73323670)

[2.1.3.4 Quy trình nghiệp vụ giảng viên tham gia khoá học trực tuyến 35](#_Toc73323671)

[2.2 Đặc tả chức năng 36](#_Toc73323672)

[2.2.1 Đặc tả use case đăng nhập bằng tài khoản bên thứ ba 36](#_Toc73323673)

[2.2.2 Đặc tả use case Đăng ký 39](#_Toc73323674)

[2.2.3 Đặc tả use case Đăng nhập 41](#_Toc73323675)

[2.2.4 Đặc tả use case Nhắn tin 43](#_Toc73323676)

[2.2.5 Đặc tả use case Tạo khoá học 46](#_Toc73323677)

[2.2.6 Đặc tả use case Tham gia phòng học trực tuyến 53](#_Toc73323678)

[2.3 Yêu cầu phi chức năng 54](#_Toc73323679)

[2.3.1 Yêu cầu về bảo mật 54](#_Toc73323680)

[2.3.2 Yêu cầu về giao diện 55](#_Toc73323681)

[2.3.3 Yêu cầu về kỹ thuật 55](#_Toc73323682)

[Chương 3 Công nghệ sử dụng 56](#_Toc73323683)

[3.1 Frontend 56](#_Toc73323684)

[3.1.1 React 56](#_Toc73323685)

[3.2 Backend 56](#_Toc73323686)

[3.2.1 Laravel 56](#_Toc73323687)

[3.2.2 Flask 57](#_Toc73323688)

[3.2.3 Websocket 57](#_Toc73323689)

[3.2.4 WebRTC 57](#_Toc73323690)

[3.2.5 Redis 58](#_Toc73323691)

[3.2.6 Elasticsearch 58](#_Toc73323692)

[3.2.7 Recommendation systems 58](#_Toc73323693)

[3.3 Server 62](#_Toc73323694)

[3.3.1 Amazon web service 62](#_Toc73323695)

[3.3.2 Microservice - Docker 63](#_Toc73323696)

[Chương 4 Phát triển và triển khai ứng dụng 64](#_Toc73323697)

[4.1 Thiết kế kiến trúc 64](#_Toc73323698)

[4.1.1 Kiến trúc phần mềm 64](#_Toc73323699)

[4.1.2 Thiết kế tổng quan 65](#_Toc73323700)

[4.2 Thiết kế chi tiết 68](#_Toc73323701)

[4.2.1 Thiết kế giao diện 68](#_Toc73323702)

[4.2.2 Thiết kế lớp 70](#_Toc73323703)

[4.2.3 Thiết kế cơ sở dữ liệu 74](#_Toc73323704)

[4.2.3.1 Sơ đồ thực thể liên kết 74](#_Toc73323705)

[4.2.3.2 Thiết kế chi tiết cơ sở dữ liệu 75](#_Toc73323706)

[4.3 Xây dựng ứng dụng 87](#_Toc73323707)

[4.3.1 Thư viện và công cụ sử dụng 87](#_Toc73323708)

[4.3.2 Kết quả đạt được 88](#_Toc73323709)

[4.3.3 Minh hoạ các chức năng chính 88](#_Toc73323710)

[4.4 Triển khai 98](#_Toc73323711)

# Danh mục hình vẽ

[**Hình 2.1** Biểu đồ use case tổng quan 14](#_Toc73323712)

[**Hình 2.2** Biểu đồ use case phân rã Authenticate 15](#_Toc73323713)

[**Hình 2.3** Biểu đồ use case phân rã quản lý thông tin cá nhân 16](#_Toc73323714)

[**Hình 2.4** Biểu đồ use case phân rã Xem danh sách khóa học 18](#_Toc73323715)

[**Hình 2.5** Biểu đồ use case phân rã tạo chương học 19](#_Toc73323716)

[**Hình 2.6** Biểu đồ use case phân rã Lên lớp 21](#_Toc73323717)

[**Hình 2.7** Biểu đồ use case phân rã Học 23](#_Toc73323718)

[**Hình 2.8** Biểu đồ use case phân rã nhắn tin 25](#_Toc73323719)

[**Hình 2.9** Biểu đồ use case phân rã quản lý khóa học 26](#_Toc73323720)

[**Hình 2.10** Biểu đồ use case phân rã quản lý người dùng 27](#_Toc73323721)

[**Hình 2.11** Biểu đồ use case phân rã quản lý topics 28](#_Toc73323722)

[**Hình 2.12** Biểu đồ use case phân rã thống kê hệ thống 29](#_Toc73323723)

[**Hình 2.13** Biểu đồ quy trình nghiệp vụ tạo xác thực người dùng 30](#_Toc73323724)

[**Hình 2.14** Biểu đồ quy trình nghiệp vụ tạo khóa học 31](#_Toc73323725)

[**Hình 2.15** Biểu đồ quy trình nghiệp vụ học sinh tham gia phòng học trực tuyến 33](#_Toc73323726)

[**Hình 2.16** Biểu đồ quy trình nghiệp vụ giảng viên tham gia khoá học trực tuyến 35](#_Toc73323727)

[**Hình 4.1** Sơ đồ gói backend 65](#_Toc73323728)

[**Hình 4.2** Sơ đò gói frontend 67](#_Toc73323729)

[**Hình 4.3** Giao diện thiết bị bàn hình lớn 69](#_Toc73323730)

[**Hình 4.4** Giao diện thiết bị bàn hình nhỏ 70](#_Toc73323731)

[**Hình 4.5** Thiết kế lớp CourseController phía backend 71](#_Toc73323732)

[**Hình 4.6** Thiết kế lớp LesonInput phía frontend 72](#_Toc73323733)

[**Hình 4.7** Thiết kế lớp EditCourseReducer phía frontend 73](#_Toc73323734)

[**Hình 4.8** Sơ đồ thực thể liên kết 74](#_Toc73323735)

[**Hình 4.9** Giao diện soạn thảo khoá học 88](#_Toc73323736)

[**Hình 4.10** Thao tác kéo, thả 89](#_Toc73323737)

[**Hình 4.11** Cài đặt chương học 89](#_Toc73323738)

[**Hình 4.12** Soạn thảo nội dung định dạng được 90](#_Toc73323739)

[**Hình 4.13** Soạn thảo nội dung trên popup 90](#_Toc73323740)

[**Hình 4.14** Preview nội dung khoá học 91](#_Toc73323741)

[**Hình 4.15** Phòng học trực tuyến 91](#_Toc73323742)

[**Hình 4.16** Trình chiếu màn hình 92](#_Toc73323743)

[**Hình 4.17** Giao diện bảng trắng 92](#_Toc73323744)

[**Hình 4.18** Tài liệu bài học 93](#_Toc73323745)

[**Hình 4.19** Danh sách và thông tin tiến độ học tập 93](#_Toc73323746)

[**Hình 4.20** Quản lý khoá học 94](#_Toc73323747)

[**Hình 4.21** Quản lý người dùng 94](#_Toc73323748)

[**Hình 4.22** Giao diện bài học mobile 95](#_Toc73323749)

[**Hình 4.23** Giao diện nhắn tin mobile 96](#_Toc73323750)

[**Hình 4.24** Giao diện thông tin cá nhân mobile 97](#_Toc73323751)

[**Hình 4.25** Sơ đồ triển khai tổng quan của hệ thống 98](#_Toc73323752)

# Danh mục bảng

[**Bảng 2.1** Đặc tả use case đăng nhập bằng tài khoản bên thứ ba 36](#_Toc73323753)

[**Bảng 2.2** Dữ liệu đầu vào đăng ký với tài khoản bên thứ ba 38](#_Toc73323754)

[**Bảng 2.3** Đặc tả use case Đăng ký 39](#_Toc73323755)

[**Bảng 2.4** Dữ liệu form đăng ký 41](#_Toc73323756)

[**Bảng 2.5** Đặc tả use case Đăng nhập 41](#_Toc73323757)

[**Bảng 2.6** Dữ liệu form đăng nhập 43](#_Toc73323758)

[**Bảng 2.7** Đặc tả use case Nhắn tin 43](#_Toc73323759)

[**Bảng 2.8** Dữ liệu đầu vào của tin nhắn 45](#_Toc73323760)

[**Bảng 2.9** Đặc tả use case Tạo khoá học 46](#_Toc73323761)

[**Bảng 2.10** Dữ liệu đầu vào các thông tin của khoá học 50](#_Toc73323762)

[**Bảng 2.11** Đặc tả use case Tham gia phòng học trực tuyến 53](#_Toc73323763)

[**Bảng 3.1** Ma trận student-course 60](#_Toc73323764)

[**Bảng 3.2** Ma trận student-course đã chuẩn hoá 60](#_Toc73323765)

[**Bảng 3.3** Dự đoán giá trị rating còn thiếu 61](#_Toc73323766)

[**Bảng 3.4** Giá trị dự đoán rating cuối cùng 62](#_Toc73323767)

[**Bảng 4.1** Thiết kết chi tiết bảng ues 75](#_Toc73323768)

[**Bảng 4.2** Thiết kết chi tiết bảng accounts 76](#_Toc73323769)

[**Bảng 4.3** Thiết kết chi tiết bảng student\_settings 76](#_Toc73323770)

[**Bảng 4.4** Thiết kết chi tiết bảng instructor\_setting 77](#_Toc73323771)

[**Bảng 4.5** Thiết kết chi tiết bảng notifications 77](#_Toc73323772)

[**Bảng 4.6** Thiết kết chi tiết bảng assets 78](#_Toc73323773)

[**Bảng 4.7** Thiết kết chi tiết bảng mesages 79](#_Toc73323774)

[**Bảng 4.8** Thiết kết chi tiết bảng rooms 79](#_Toc73323775)

[**Bảng 4.9** Thiết kết chi tiết bảng room\_user 80](#_Toc73323776)

[**Bảng 4.10** Thiết kết chi tiết bảng user\_topic 80](#_Toc73323777)

[**Bảng 4.11** Thiết kết chi tiết bảng topics 80](#_Toc73323778)

[**Bảng 4.12** Thiết kết chi tiết bảng course\_topic 81](#_Toc73323779)

[**Bảng 4.13** Thiết kết chi tiết bảng courses 81](#_Toc73323780)

[**Bảng 4.14** Thiết kết chi tiết bảng course\_student 82](#_Toc73323781)

[**Bảng 4.15** Thiết kết chi tiết bảng sections 83](#_Toc73323782)

[**Bảng 4.16** Thiết kết chi tiết bảng section\_student 84](#_Toc73323783)

[**Bảng 4.17** Thiết kết chi tiết bảng lessons 84](#_Toc73323784)

[**Bảng 4.18** Thiết kết chi tiết bảng live\_lessons 85](#_Toc73323785)

[**Bảng 4.19** Thiết kết chi tiết bảng quesstions 86](#_Toc73323786)

[**Bảng 4.20** Thiết kết chi tiết bảng answers 86](#_Toc73323787)

[**Bảng 4.21** Thư viện và công cụ sử dụng 87](#_Toc73323788)

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **ĐATN** | Đồ án tốt nghiệp |
| **CNTT** | Công nghệ thông tin |
| **JWT** | JSON Web Token  Mã thông báo web JSON |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **SV** | Sinh viên |
| **WYSIWYG** | What You See Is What You Get  Công cụ soạn thảo văn bản có định dạng |

# Danh mục thuật ngữ

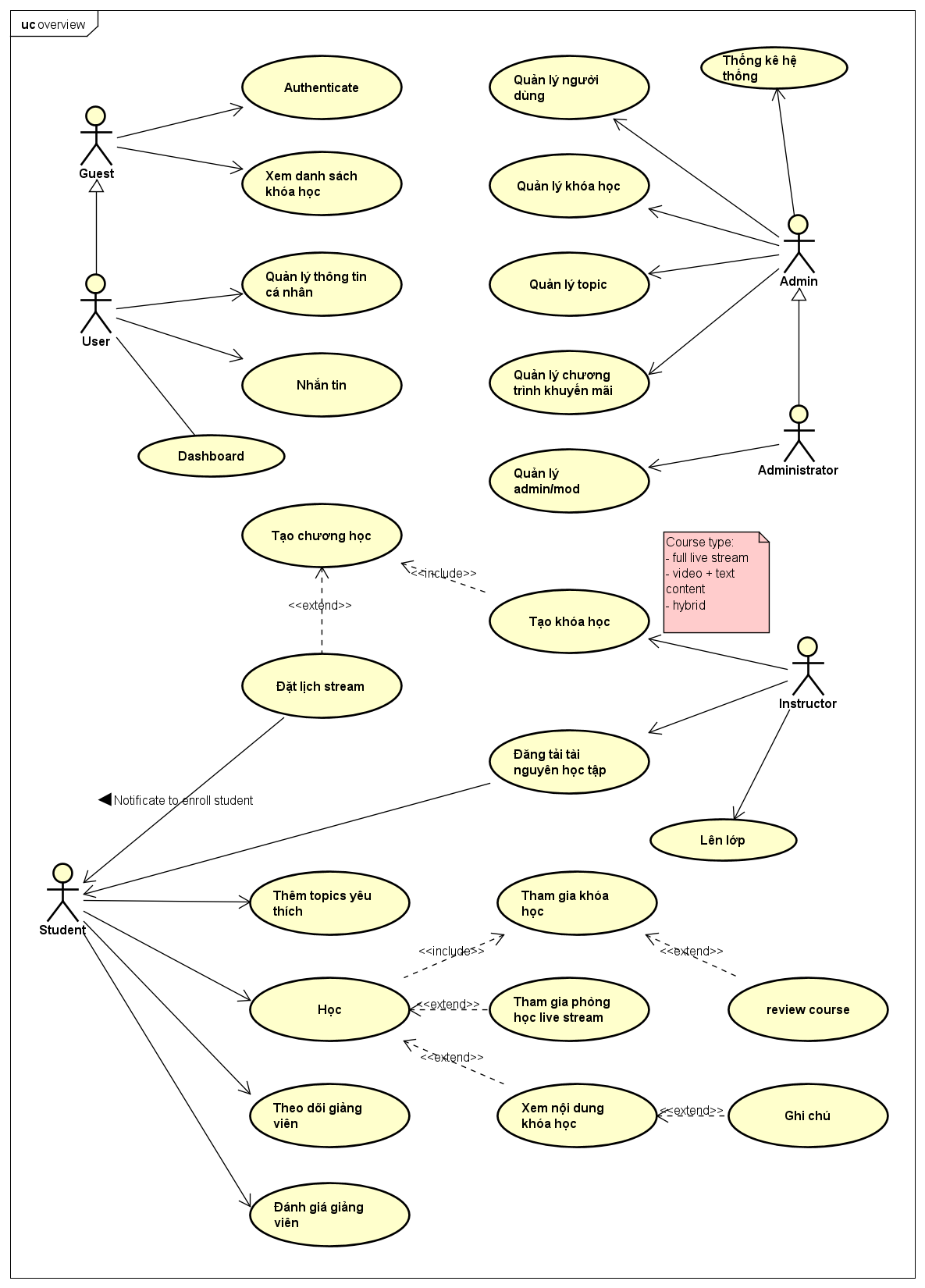
|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **Authentication** | Xác thực danh tính |
| **Authorization** | Xác định quyền truy cập |
| **Access token** | Mã truy cập |
| **Refresh token** | Mã làm mới – có tác dụng làm mới mã truy cập |
|  |  |
|  |  |

# Giới thiệu đề tài

# Khảo sát và phân tích yêu cầu

## Tổng quan chức năng

### Biểu đồ use case tổng quan

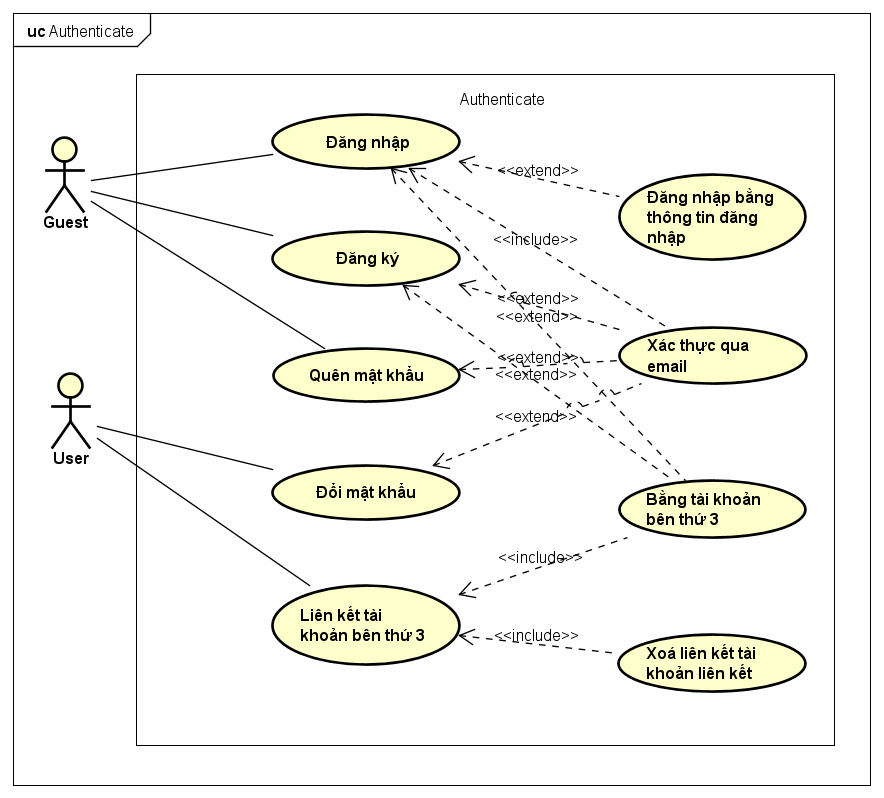


**Hình 2.1** Biểu đồ use case tổng quan

* Khách (Guest) là những người chưa đăng nhập. Khách có thể tìm kiếm, xem danh sách khóa học, thông tin giới thiệu và đánh giá người dùng của khóa học.
* Người dùng (User) là những người đăng nhập vào hệ thống. Người dùng bao gồm Học sinh (Student), Giảng viên (Instructor), Quản trị viên (Administrator/Admin).
* Admin là những tài khoản đặc biệt của hệ thống, là bên quản lý hệ thống và đưa ra thông báo cho người dùng.
* Hệ thống cung cấp nền tảng để giảng viên tạo và đăng tải các khóa học với video bài học được quay sẵn, nội dung văn bản được định dạng, có thể chứa tập tin đa phương tiện, biểu thức toán học, block code… hoặc các khóa học hoàn toàn livestream trực tuyến cho học sinh.
* Hệ thống cũng cung cấp một môi trường giao tiếp, trao đổi, giải đáp thắc mắc giữa các học trong các khóa học, bài học và tin nhắn riêng. Hệ thống có cơ chế đánh giá giảng viên, khóa học, các topic giúp học sinh dễ dàng tìm kiếm, chọn lựa khóa học phù hợp.

### Biểu đồ use case phân rã

#### Biểu đồ use case phân rã Authenticate



**Hình 2.2** Biểu đồ use case phân rã Authenticate

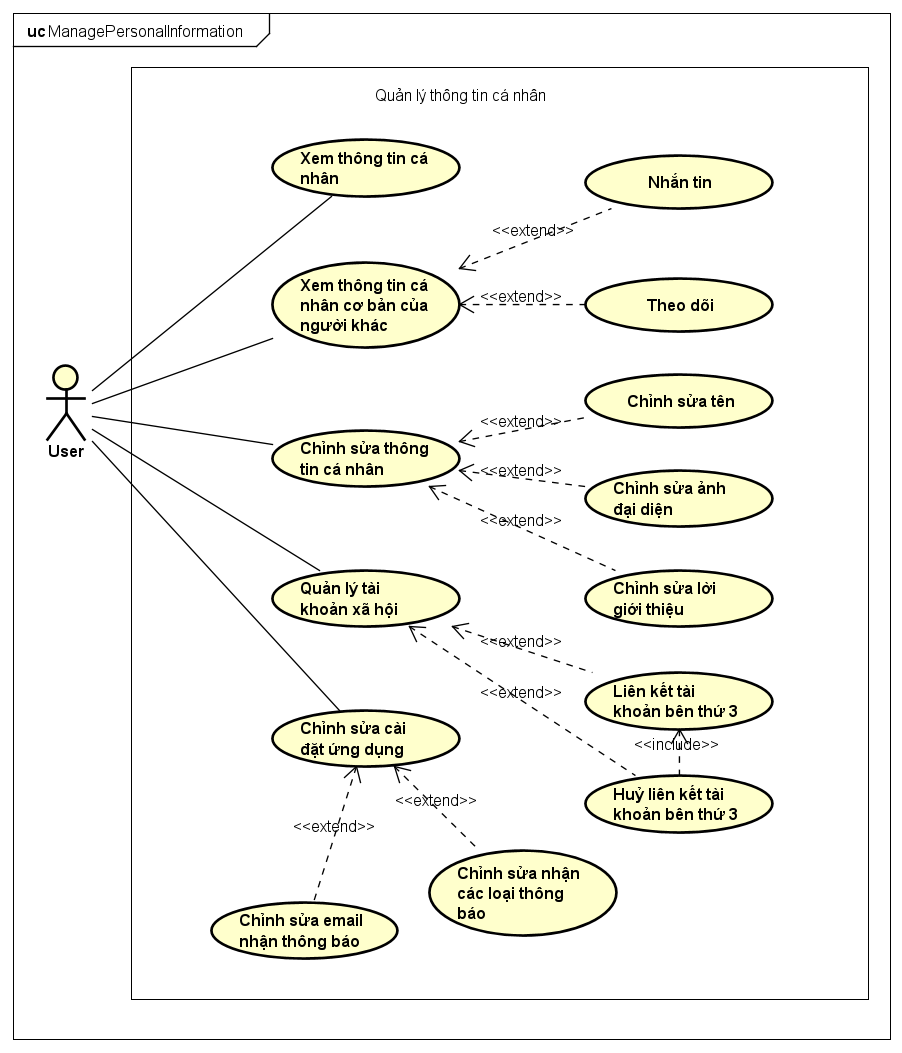
*Hình 2.2* mô tả nhóm use case Authenticate bao gồm hai tác nhân là *(i) khách* và *(ii) người dùng*. *Khách* muốn đăng nhập vào hệ thống để trở thành *người dùng* cần phải đăng ký tài khoản. Tài khoản được đăng ký bằng cách cung cấp những thông tin email, username, password, loại tài khoản (giảng viên, học sinh)… như thông thường. Ngoài ra, *khách* cũng có thể đăng ký tài khoản nhanh chóng bằng Oauth2 của bên thứ ba (tài khoản xã hội) - hiện tại hỗ trợ Google, Facebook, Github, và cung cấp thêm username.

Sau khi đăng ký với tài khoản xã hội, *khách* hoàn toàn có thể đăng nhập thông qua tài khoản liên kết hoặc đăng nhập bằng email/username và password như thông thường nếu đã cung cấp password lúc đăng ký tài khoản.

Một tài khoản của *người dùng* có thể liên kết với nhiều tài khoản xã hội khác nhau và đăng nhập bằng các tài khoản này.

Các thao tác đăng ký, quên mật khẩu… yêu cầu bảo mật cao đều có email gửi về email chính của người dùng để xác thực danh tính.

#### Biểu đồ use case phân rã Quản lý thông tin cá nhân



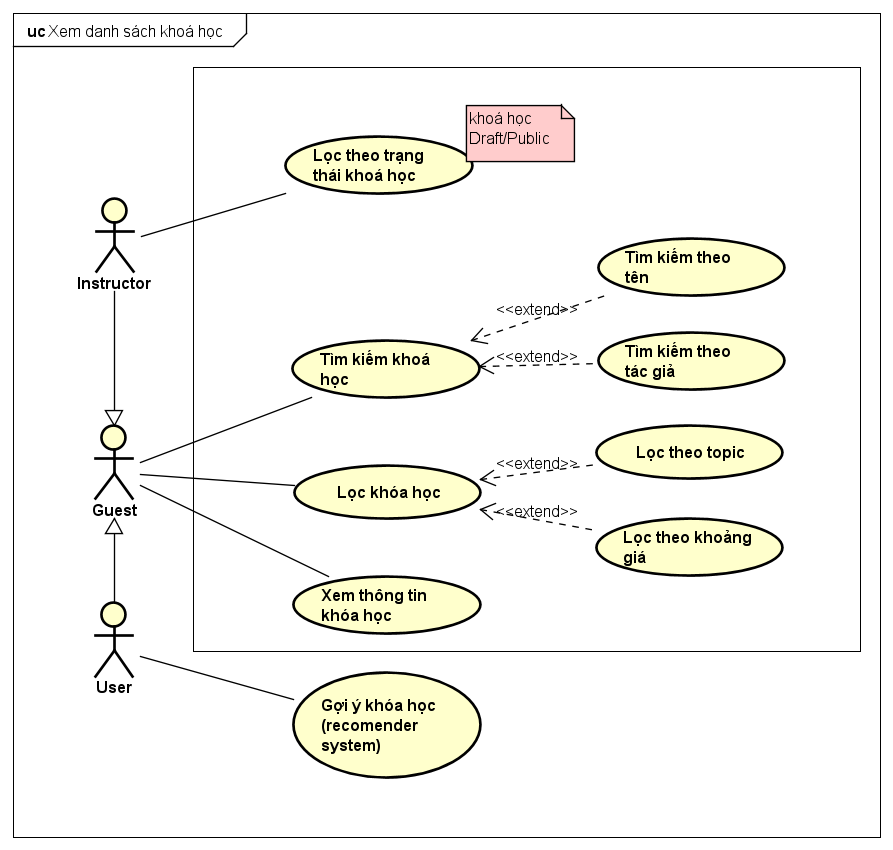
**Hình 2.3** Biểu đồ use case phân rã quản lý thông tin cá nhân

*Hình 2.3* mô tả nhóm use case quản lý thông tin cá nhân của *người dùng*. Sau khi đã đăng nhập vào hệ thống, *người dùng* có thể tuỳ ý thay đổi và cập nhật các thông tin *(i) Họ tên, (ii) Ảnh đại diện, (iii) thông tin giới thiệu, (iv) thêm tài khoản xã hội, (v) xoá tài khoản xã hội đã liên kết.*

Mặc định, khi đăng ký tài khoản, các cài đặt nhận thông báo về khoá học mới, lượt mua, lượt thích khoá học… sẽ được tắt. *Người dùng* có thể bật các cài đạt thông báo này và xem email nhận thông báo trong phần thông tin cá nhân.

Ngoài ra, *người dùng* cùng có thể xem thông tin cá nhân cơ bản như tên, ảnh đại diện, thông tin giới thiệu của *người dùng* khác. Tại trang này, *người dùng* có thể bắt đầu tạo chat room với nhau – những tài khoản đã tạo cùng chat room sẽ hiển thị trong danh sách chat. Đối với tài khoản được xem là *giảng viên,* trang sẽ có thêm chức năng theo dõi để có thể nhận các thông báo về khoá học mới mà *giảng viên* này tạo.

#### Biểu đồ use case phân rã Xem danh sách khóa học

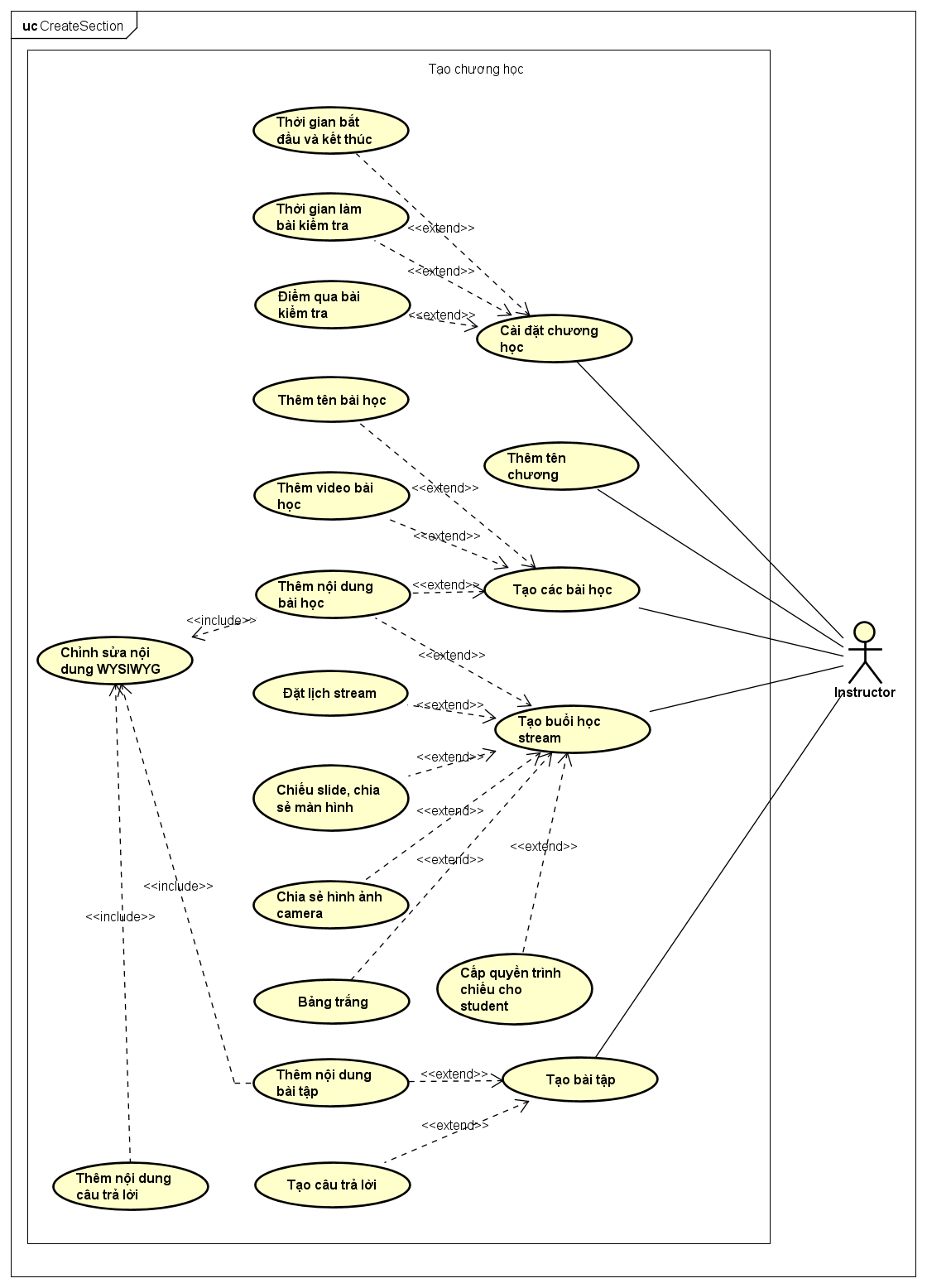


**Hình 2.4** Biểu đồ use case phân rã Xem danh sách khóa học

*Hình 2.4* mô tả nhóm use case xem danh sách khoá học của các tác nhân *(i) khách, (ii) học sinh, (iii) giảng viên.*Tất cả tác nhân đều có thể xem danh sách khoá học đã được publish (xuất bản) trên hệ thống và tìm kiếm, lọc các khoá học bằng một số tiêu chí.

Ngoài ra, *giảng viên* có thể xem thêm các khoá học ở trạng thái draft (nháp) của mình.

#### Biểu đồ use case phân rã Tạo chương học



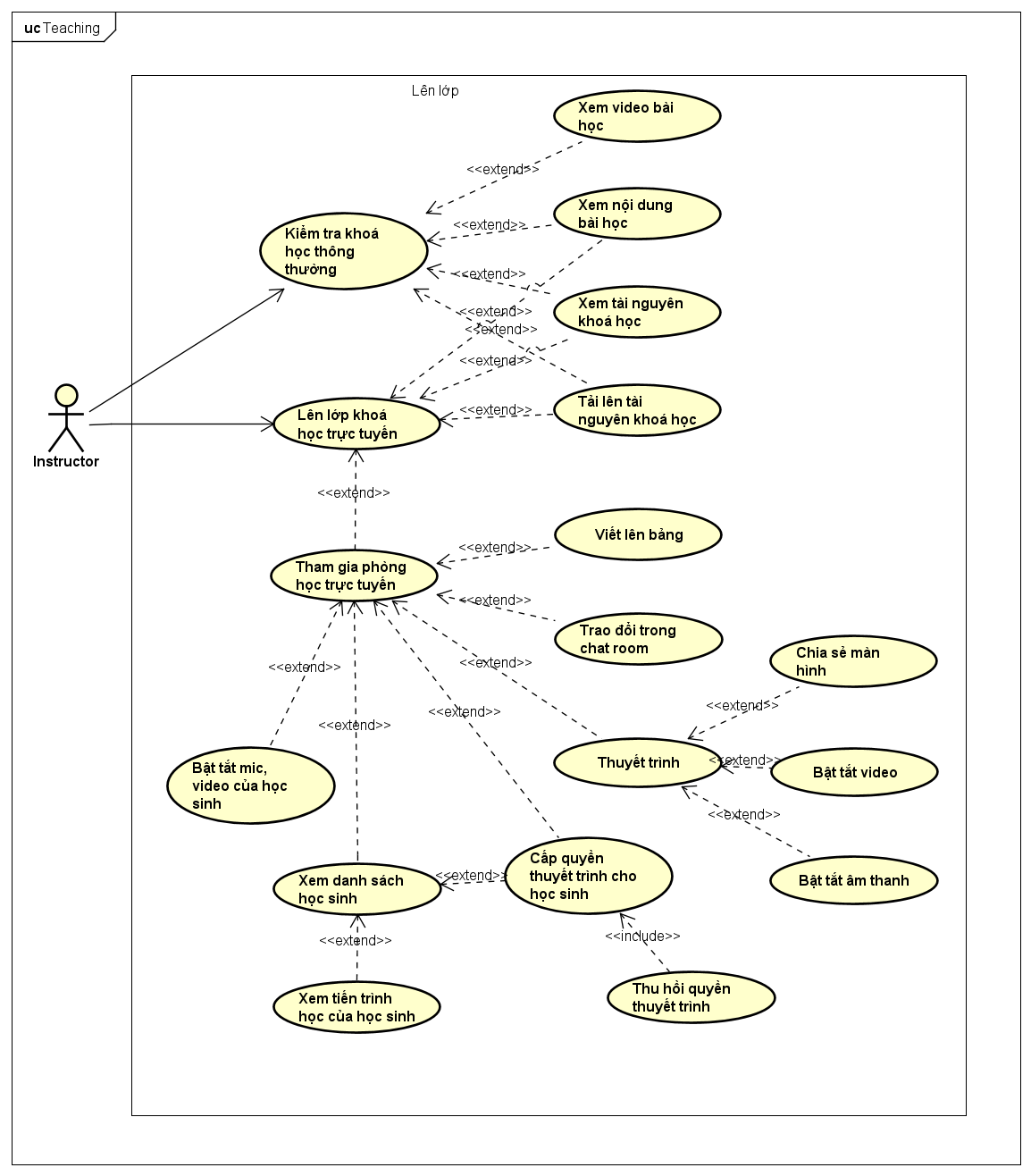
**Hình 2.5** Biểu đồ use case phân rã tạo chương học

*Hình 2.5* mô tả nhóm use case tạo chương học của tác nhân *giảng viên.* *Giảng viên* tạo các chương học sau khi đã tạo khoá học.

Đối với *khoá học thông thường, giảng viên* có thể thêm các *bài học (Lesson)* thông thường – có 1 video bài học quay sẵn và phần nội dung được định dạng. Đối với *khoá học trực tuyến, giảng viên* có thể thêm các *bài học trực tuyến (Live lesson)* – có phần nội dung được định dạng, lịch học (thời gian bắt đầu và kết thúc) và phòng học trực tuyến.

Mỗi chương học, *giảng viên* có thể thêm nhiều câu hỏi trắc nghiệm và cài đặt thời gian làm bài, điểm qua, giới hạn thời gian giữa các lần làm bài tập. Khi *học sinh* học khoá học, các câu hỏi này sẽ xuất hiện và có thể làm ở mỗi cuối chương học. Nếu *giảng viên* không cài đặt những cài đặt về câu hỏi, các câu hỏi sẽ không tính thời gian làm.

#### Biểu đồ use case phân rã Lên lớp



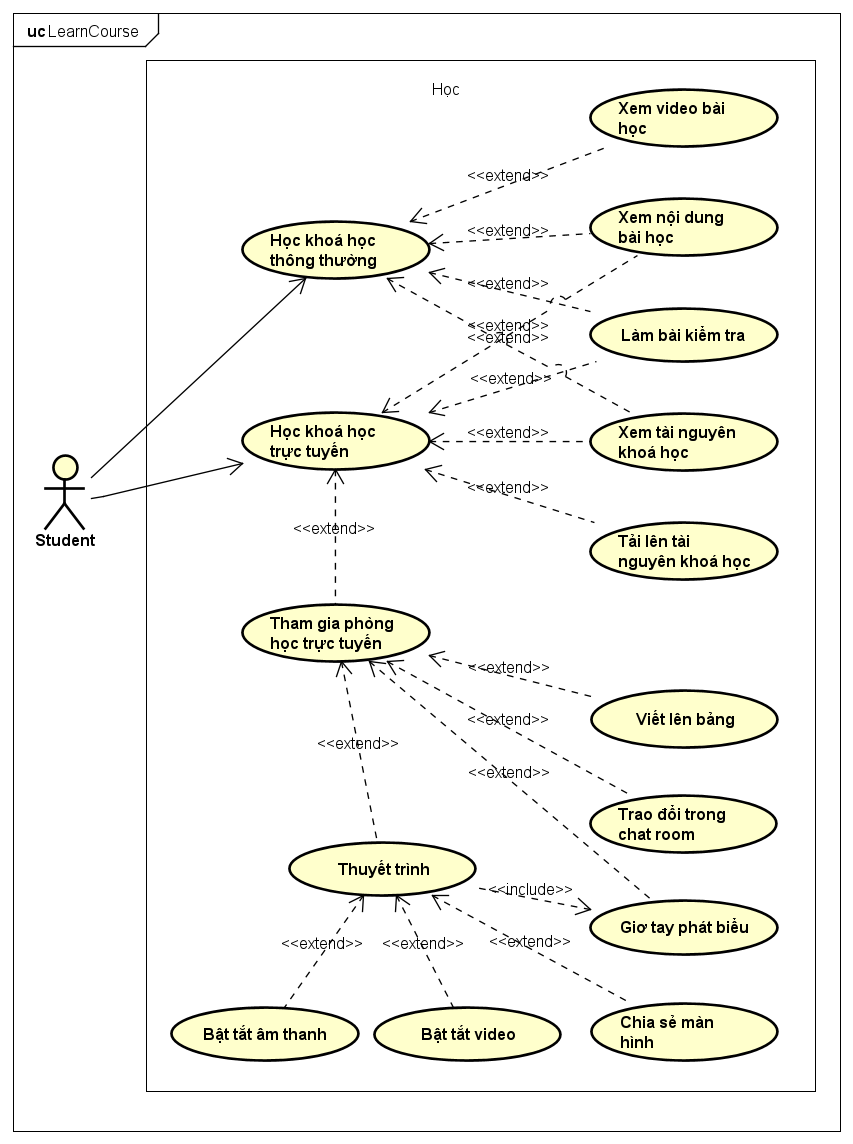
**Hình 2.6** Biểu đồ use case phân rã Lên lớp

*Hình 2.6* mô tả nhóm use case lên lớp của tác nhân *giảng viên.* Hệ thống hiện hỗ trợ hai loại khoá học là khoá học thông thường – mỗi bài học có phần video nội dung quay sẵn và khoá học trực tuyến – mỗi bài học có một phòng học trực tuyến cho phép nhiều người cùng tham gia thuyết trình.

Đối với khoá học thông thường, *giảng viên* có thể vào khoá học để kiểm tra lại nội dung bài học, câu hỏi đã soạn, đăng tải các tài nguyên giảng dạy vào mỗi bài học, trao đổi , giải đáp thắc mắc của *học sinh* tham gia khoá học trên chat room của bài học.

Đối với khoá học trực tuyến, *giảng viên* cũng có thể vào khoá học để kiểm tra lại nội dung bài học câu hỏi đã soạn, đăng tải các tài nguyên cần thiết. Ngoài ra khi đến giờ học được *giảng viên* cài đặt sẵn, *giảng viên* vào bài học sẽ để tham gia phòng học trực tuyến. Tại đây, giảng viên có thể bắt đầu dạy bằng cách thuyết trình, trình chiếu, viết bảng, gọi phát biểu, trao đổi, giải đáp, kiểm tra tiến độ học sinh… bằng các công cụ được hệ thống cung cấp.

#### Biểu đồ use case phân rã Học



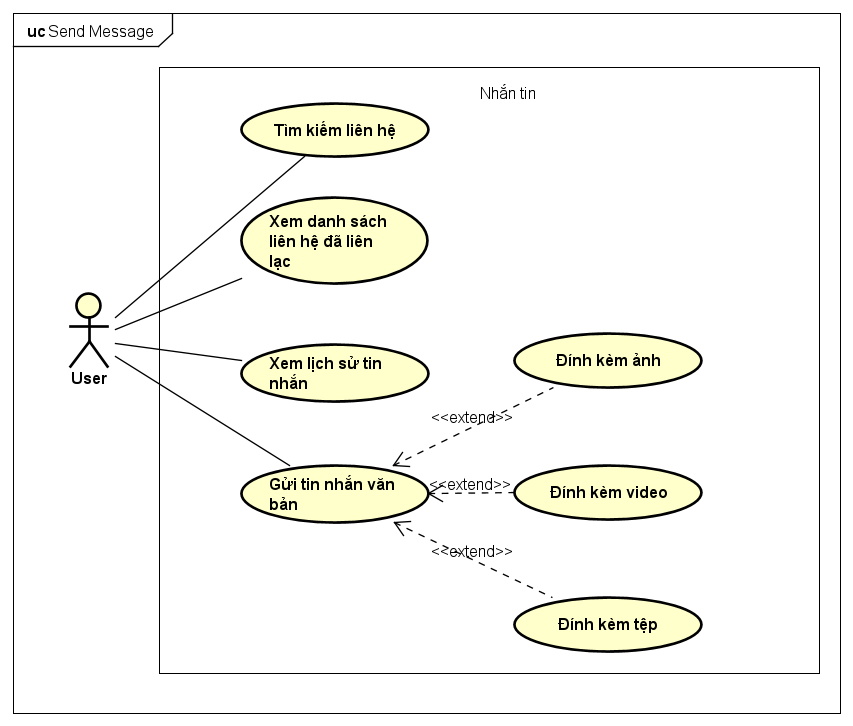
**Hình 2.7** Biểu đồ use case phân rã Học

*Hình 2.7* mô tả nhóm use case học khoá học của tác nhân *học sinh.* Như ở mục 2.1.2.6 đã đề cập, hệ thống cung cấp cho *học sinh* hai loại khoá học là khoá học thông thường và khoá học trực tuyến.

Đối với khoá học thông thường, *học sinh* có thể học bằng các tài nguyên, bài giảng được *giảng viên* soạn sẵn. Các tài nguyên này bao gồm *(i) video bài học, (ii) nội dung văn bản bài học, (iii) tài liệu được giảng viên đăng tải, (iv) chat room để hỏi đáp, trao đổi, (v) bài kiểm tra để test lại kiến thức đã học.*Mô hình này giống với một số nền tảng học trực tuyến hiện tại như Coursera, edX, Edumal. *Học sinh* có thể tham gia khoá học loại này vào bất cứ thời điểm nào.

Đối với khoá học trực tuyến, *học sinh* cũng có các nội dung tương tự như khoá học thông thường. Ngoài ra, *học sinh* sẽ tham gia phòng học trực tuyến ở mỗi bài học.Tại phòng học trực tuyến, học sinh sẽ theo dõi bài giảng trực tiếp của *giảng viên*. Khi cần phát biểu, *học sinh* sẽ sử dụng chức năng giơ tay và có thể thuyết trình, phát biểu khi được *giảng viên* cho phép. Phòng học trực tuyến cũng có bảng trắng dùng chung cho cả lớp học, chat room để trao đổi với *giảng viên* và *học sinh* khác. Khác với khoá học thông thường, khoá học trực tuyến chỉ có thể được đăng ký và tham gia học vào khung giờ nhất định mà giảng viên quy dịnh sẵn.

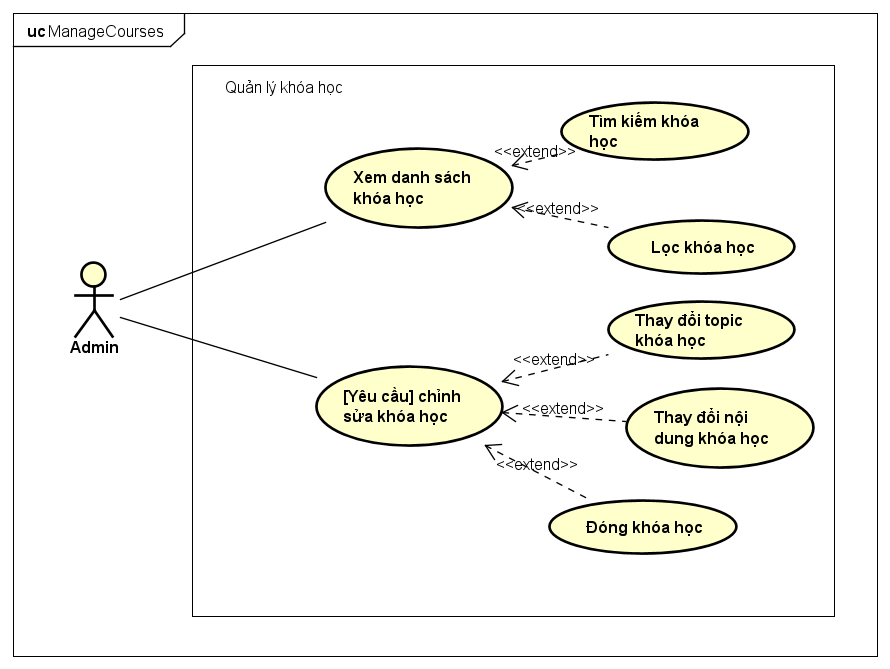
#### Biểu đồ use case Nhắn tin



**Hình 2.8** Biểu đồ use case phân rã nhắn tin

*Hình 2.5* mô tả nhóm use case nhắn tin của *người dùng*. *Người dùng* có hệ thống có thể gửi tin nhắn đến bất cứ *người dùng* khác nào có trong hệ thống. Tin nhắn có thể đính kèm các tệp, chèn hyperlink. Nếu các tệp đính kèm là tệp đa phương tiện (hình ảnh, video), hệ thống sẽ hiển thị bản xem trước đối với những tệp này. Nếu là các tệp khác, người dùng nhấn vào icon của các tệp để tiến hành tải tệp đính kèm về. Mỗi tệp mà *người dùng* gửi đều được đăng tải lên storage server của hệ thống và được hệ thống lưu vết.

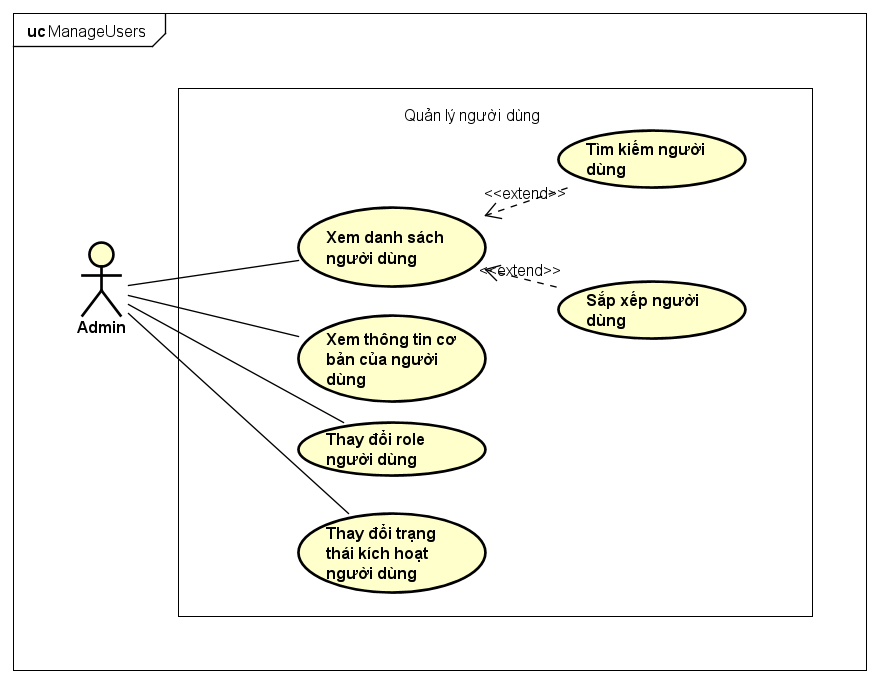
#### Biểu đồ use case Quản lý khóa học



**Hình 2.9** Biểu đồ use case phân rã quản lý khóa học

*Hình 2.6* mô tả nhóm use case quản lý khoá học của *quản trị viên.* *Quản trị viên* chỉ có thể xem, kiểm duyệt và yêu cầu chỉnh sửa nội dung các khoá học chứ không có quyền tự chỉnh sửa các khoá học của *giảng viên*.

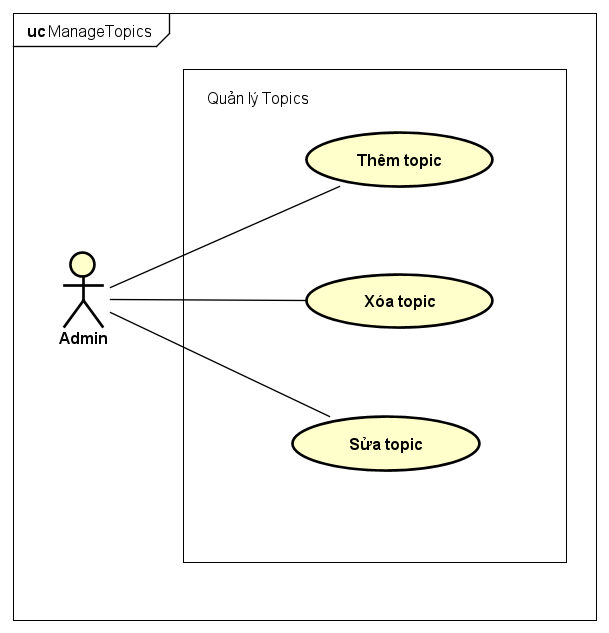
#### Biểu đồ use case phân rã Quản lý người dùng



**Hình 2.10** Biểu đồ use case phân rã quản lý người dùng

*Hình 2.7* mô tả nhóm use case quản lý người dùng của *quản trị viên.* *Quản trị viên* có thể xem danh sách người dùng và tạo các tài khoản *quản trị viên* khác.

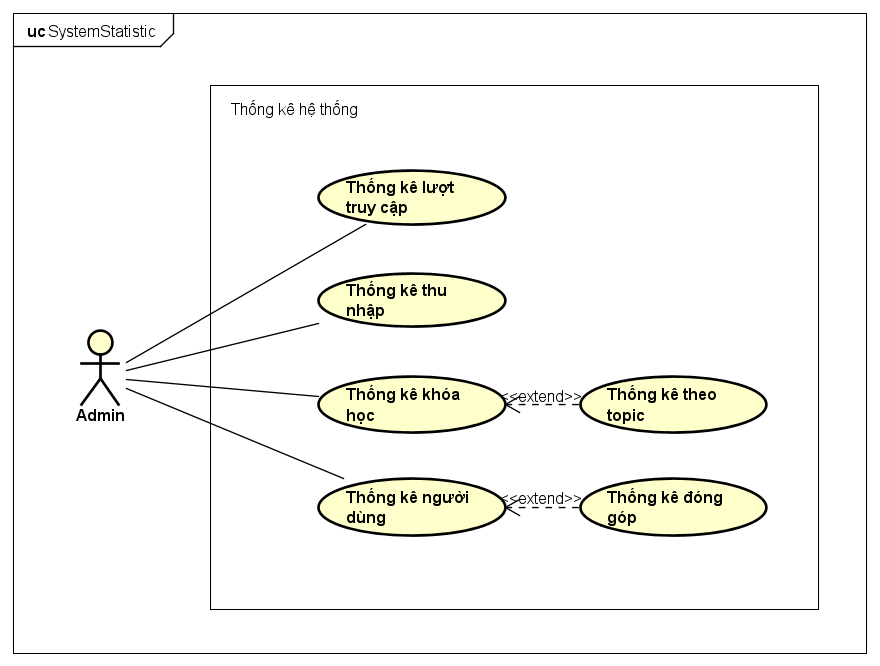
#### Biểu đồ use case phân rã quản lý topic



**Hình 2.11** Biểu đồ use case phân rã quản lý topics

*Hình 2.8* mô tả nhóm use case quản lý topics của *quản trị viên.* *Quản trị viên* có thể xem danh sách các topic và tạo các topic mới.

#### Biểu đồ use case phân rã thống kê hệ thống

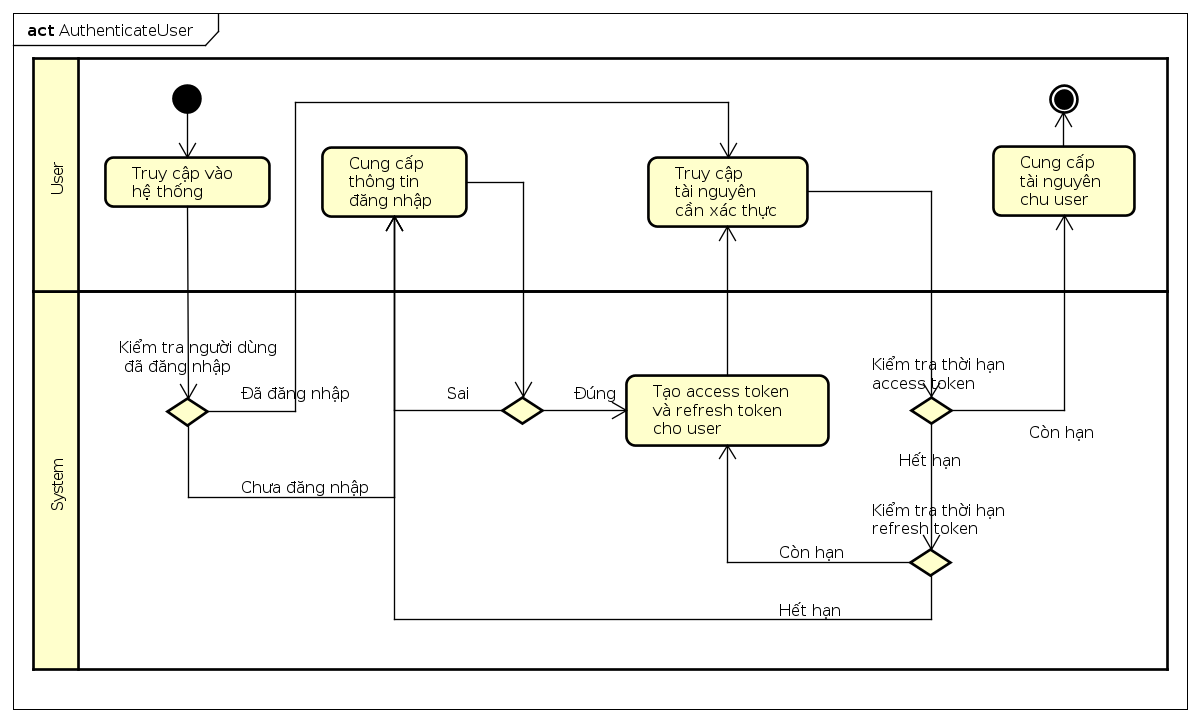


**Hình 2.12** Biểu đồ use case phân rã thống kê hệ thống

*Hình 2.9* mô tả nhóm use case thống kê hệ thống của *quản trị viên.* *Quản trị viên* có thể xem các chỉ số thống kê của hệ thống.

### Quy trình nghiệp vụ

#### Quy trình nghiệp vụ xác thực người dùng

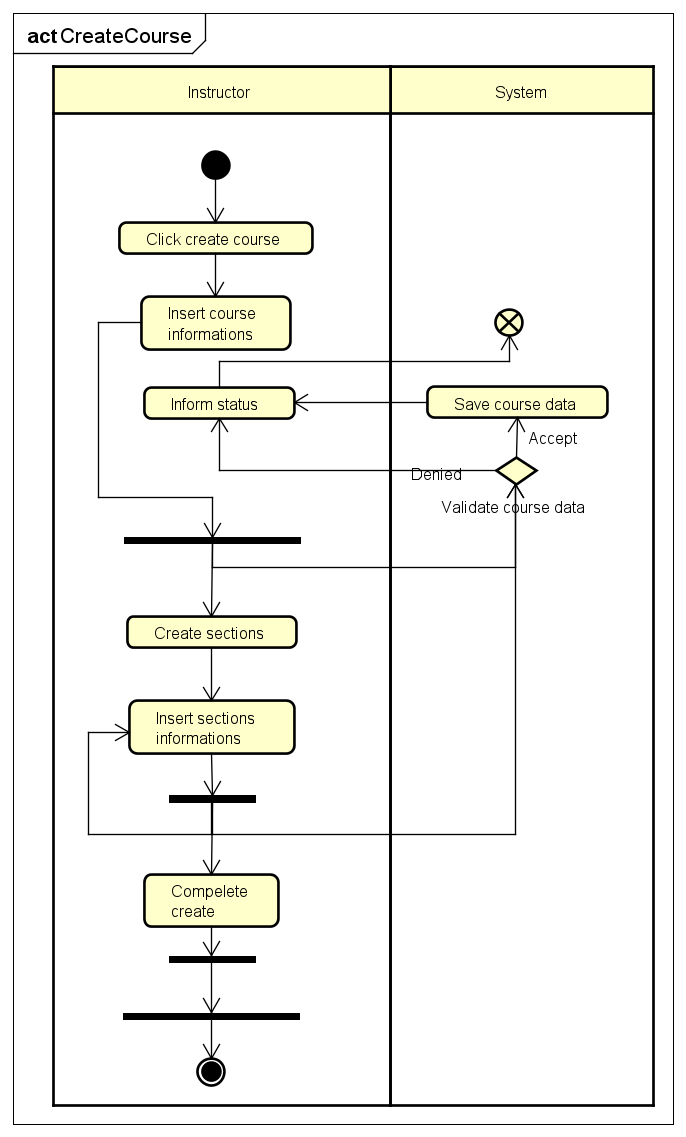


**Hình 2.13** Biểu đồ quy trình nghiệp vụ tạo xác thực người dùng

*Hình 2.10* Quy trình nghiệp vụ xác thực người dùng bao gồm các use case trong nhóm use case Authenticate và có sức ảnh hưởng bao trùm toàn bộ hệ thống. Nghiệp vụ này giúp xác thực danh tính của *người dùng* trong hệ thống (authentication), qua đó quyết định việc *người dùng* có quyền gì đối với tài nguyên trong hệ thống (authorization).

Quy trình xác thực người dùng sử dụng công nghệ JWT để tạo ra token chứa thông tin xác thực của *người dùng.* Token này đượclưu trữ tại browser của end user hoặc một nơi an toàn và cung cấp lại cho server để xác thực danh tính.

#### Quy trình nghiệp vụ tạo khóa học



**Hình 2.14** Biểu đồ quy trình nghiệp vụ tạo khóa học

*Hình 2.11* Mô tả quy trình nghiệp vụ tạo khoá học. Hệ thống cung cấp cho *giảng viên* một giao diện tạo khoá học trực quan. Các chương học, bài học, câu hỏi được tổ chức dưới dạng các thanh có thể thu gọn và mở rộng nội dung hiển thị, kéo, thả để sắp xếp, xoá bỏ. Hệ thống cũng cung cấp công cụ soạn thảo WYSIWYG cho *giảng viên* khi soạn thảo các nội dung của khoá học. Với công cụ này, *giảng viên* có thể căn chỉnh lề đoạn văn, định dạng kiểu chữ, cỡ chữ, đậm nhạt, màu sắc của văn bản. Công cụ này cũng hỗ trợ chèn ảnh, video, block code, công thức toán học vào văn bản.

Khi *giảng viên* soạn thảo văn bản, hệ thống sẽ lưu trữ, đồng bộ dữ liệu của khoá học theo **thời gian thực**. Vì vậy, nếu chẳng may gặp sự cố mất mạng, mất điện hoặc vô tình tắt browser, dữ liệu khoá học của *giảng viên* vẫn được bảo toàn nguyên vẹn.

#### Quy trình nghiệp vụ học sinh tham gia phòng học trực tuyến



**Hình 2.15** Biểu đồ quy trình nghiệp vụ học sinh tham gia phòng học trực tuyến

*Hình 2.11* Mô tả quy trình nghiệp vụ tham gia phòng học trực tuyến, đây là một nghiệp vụ quan trọng của hệ thống. Khi *học sinh* tham gia phòng học trực tuyến, hệ thống sẽ kiểm tra quyền và thời gian học được *giảng viên* định trước. Khi đã vào phòng học thành công, hệ thống cung cấp cho *học sinh* giao diện để học tập bao gồm *(i) màn hình livestream, (ii) giơ tay phát biểu, (iii) whiteboard của phòng học, (iv) nội dung khoá học được biên soạn, (v) trao đổi, tương tác với giảng viên và bạn học trong phòng chat, (vi) bài tập sau mỗi chương học.*

Chức năng *whiteboard* cung cấp cho *giảng viên* và *học sinh* một “bảng trắng” để có thể cùng nhau viết công thức, ghi chú, nháp… trên đó. Nội dung của *whiteboard* được cập nhật theo thời gian thực với mỗi *người dùng* trong phòng học. *Whiteboard* cũng được lưu trữ cũng với bài học để *người dùng* có thể xem lại sau khi bài học đã kết thúc.

Khi mới vào phòng học, chỉ *giảng viên* mới sử dụng được chức năng *thuyết trình* ((i) bật microphone, (ii) bật webcam, (iii) chia sẻ màn hình thiết bị, trình chiếu) *học sinh* chỉ có thể xem *livestream* và trao đổi trên *phòng chat.* Để có quyền thực hiện các chức *năng thuyết* trình của *màn hình livestream*, học sinh có thể tương tác với *giảng viên* để xin cấp quyền này bằng cách chat trong *phòng chat* hoặc sử dụng chức năng *giơ tay phát biểu.*Khi được cấp quyền *thuyết trình*, *học sinh* có thể bắt đầu bật microphone, camera, chia sẻ màn hình để thuyết trình hoặc chia sẻ, trao đổi trực tiếp với giáo viên và các bạn khác qua màn hình trực tuyến.

Sau mỗi chương học, có thể có những câu hỏi được *giảng viên* chuẩn bị sẵn. Những câu hỏi này có thể có các giới hạn thời gian làm bài, điểm qua bài test, thời gian tối thiểu giữa 2 lần làm bài.

#### Quy trình nghiệp vụ giảng viên tham gia khoá học trực tuyến

Diagram

Description automatically generated

**Hình 2.16** Biểu đồ quy trình nghiệp vụ giảng viên tham gia khoá học trực tuyến

## Đặc tả chức năng

### Đặc tả use case đăng nhập bằng tài khoản bên thứ ba

**Bảng 2.1** Đặc tả use case đăng nhập bằng tài khoản bên thứ ba

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | **UC001** | Tên Use case | **Đăng nhập bằng tài khoản bên thứ ba** |
| **Tác nhân** | Khách (Guest) | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Khách | Yêu cầu liên kết với bên thứ ba | | 2. | Hệ thống | Trả về url của bên thứ ba tương ứng | | 3. | Khách | Đăng nhập vào tài khoản bên thứ ba và cấp quyền cho hệ thống | | 4. | Hệ thống | Kiểm tra thông tin người dùng bên thứ ba cung cấp | | 5. | Hệ thống | Email bên thứ ba cung cấp chưa tồn tại trong hệ thống, trả về form điền thêm thông tin cần thiết cho khách | | 6. | Khách | Điền thông tin và submit form | | 7. | Hệ thống | Tạo tài khoản mới liên kết với tài khoản bên thứ ba | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 5.a | Hệ thống | Email đã tồn tại trong hệ thống | | 6.a | Hệ thống | Tài khoản trong hệ thống chưa liên kết với bên thứ ba | | 7.a | Hệ thống | Request người dùng liên kết tài khoản | | 8.a | Khách | Xác nhận liên kết tài khoản | | 9.a | Hệ thống | Liên kết tài khoản đã tồn tại với tài khoản bên thứ ba | | 6.a.b | Hệ thống | Tài khoản trong hệ thống đã liên kết | | | |
| **Hậu điều kiện** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Hệ thống | Tạo access token và refresh token cho người dùng | | 2 | Hệ thống | Chuyển về trang trước đó | | | |

\* Dữ liệu đầu vào của form điền thông tin:

**Bảng 2.2** Dữ liệu đầu vào đăng ký với tài khoản bên thứ ba

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1. | username | Tên tài khoản | Có | Duy nhất |  |
| 2. | password | Mật khẩu dùng để đăng nhập | Không | Phải lớn hơn 7 ký tự, chứa cả chữ hoa, thường, số và ký tự đặc biệt | Aa@123123 |
| 3. | rePassword | Nhập lại password | Không | Trùng với password đã nhập |  |
| 4. | role | Loại tài khoản muốn đăng ký | Có |  |  |

### Đặc tả use case Đăng ký

**Bảng 2.3** Đặc tả use case Đăng ký

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | **UC002** | Tên Use case | **Đăng ký** |
| **Tác nhân** | Khách (Guest) | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Khách | Yêu cầu form đăng ký | | 2. | Hệ thống | Trả về form đăng ký | | 3. | Khách | Submit form đăng ký | | 4. | Hệ thống | Tạo tài khoản mới | | 5. | Hệ thống | Gửi email xác thực vào email đăng ký | | 6. | Khách | Chọn xác nhận email đăng ký | | 7. | Hệ thống | Xác nhận tài khoản của khách | | 8. | Hệ thống | Chuyển về trang chủ | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1.b | Khách | Đăng ký bằng tài khoản bên thứ ba | | 2.b | Hệ thống | Thực hiện luồng hoạt động OAuth | | 4.b | Hệ thống | Thông tin đăng ký của người dùng không hợp lệ | | 5.b | Khách | Quay lại luồng sự kiện 3 | | | |
| **Hậu điều kiện** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Hệ thống | Tạo access token cho người dùng | | | |

\* Dữ liệu đầu vào của form đăng ký:

**Bảng 2.4** Dữ liệu form đăng ký

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1. | Họ và tên người dùng |  | Có | Ký tự | Nguyễn Văn A |
| 2. | email | email người dùng là duy nhất | Có | Đúng định dạng email  Duy nhất | s.v.o.a.26@gmail.com |
| 3. | username | Tên tài khoản | Có | Duy nhất |  |
| 4. | password | Mật khẩu dùng để đăng nhập | Có | Phải lớn hơn 7 ký tự, chứa cả chữ hoa, thường, số và ký tự đặc biệt | Aa@123123 |
| 5. | rePassword | Nhập lại password | Có | Trùng với password đã nhập |  |
| 6. | role | Loại tài khoản muốn đăng ký | Có |  |  |

### Đặc tả use case Đăng nhập

**Bảng 2.5** Đặc tả use case Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | **UC003** | Tên Use case | **Đăng nhập** |
| **Tác nhân** | Khách (Guest) | | |
| **Tiền điều kiện** | Khách đã đăng ký tài khoản  Chưa đăng nhập vào hệ thống | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Khách | Yêu cầu form đăng nhập | | 2. | Hệ thống | Trả về form đăng nhập | | 3. | Khách | Submit form đăng nhập | | 4. | Hệ thống | Xác thực thông tin đăng nhập | | 5. | Hệ thống | Chuyển về trang chủ | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1.b | Khách | Đăng nhập bằng tài khoản bên thứ ba | | 2.b | Hệ thống | Thực hiện luồng hoạt động OAuth | | 4.b | Hệ thống | Thông tin đăng ký của người dùng không hợp lệ | | 5.b | Khách | Quay lại luồng sự kiện 3 | | | |
| **Hậu điều kiện** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1 | Hệ thống | Tạo access token cho người dùng | | | |

\* Dữ liệu đầu vào của form đăng nhập:

**Bảng 2.6** Dữ liệu form đăng nhập

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1. | thông tin đăng nhập | email hoặc username của người dùng | Có |  | [s.v.o.a.26@gmail.com](mailto:s.v.o.a.26@gmail.com)/  UserName |
| 2. | password | Mật khẩu dùng để đăng nhập | Có | Phải lớn hơn 7 ký tự, chứa cả chữ hoa, thường, số và ký tự đặc biệt | Aa@123123 |

### Đặc tả use case Nhắn tin

**Bảng 2.7** Đặc tả use case Nhắn tin

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | **UC004** | Tên Use case | **Nhắn tin** |
| **Tác nhân** | Người dùng | | |
| **Tiền điều kiện** |  | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Người dùng | Chọn người dùng để nhắn tin | | 2. | Hệ thống | Mở giao diện tin nhắn | | 3. | Người dùng | Soạn tin nhắn vào hộp thoại nhập tin nhắn | | 4. | Người dùng | Nhấn nút đính kèm tệp, chọn các tệp muốn đính kèm trên thiết bị của người dùng | | 5. | Hệ thống | Tải tệp của người dùng lên server lưu trữ, lưu trữ thông tin tệp vào hệ thống | | 6. | Hệ thống | Đính thông tin tệp vừa tạo vào tin nhắn đang soạn | | 7 | Hệ thống | Hiển thị tệp mà người dùng vừa tải lên. Nếu tệp là hình ảnh sẽ hiển thị hình ảnh | | 8. | Người dùng | Gửi tin nhắn | | 9. | Hệ thống | Hiển thị tin nhắn vừa gửi lên giao diện người dùng. Nếu tệp đính kèm là tệp đa phương tiên - hình ảnh, video hay tệp âm thanh, hiển thị tệp có thể tương tác lên tin nhắn. Nếu văn bản tin nhắn chứa URL, nhúng hyperlink vào văn bản đó. | | 10. | Hệ thống | Lưu tin nhắn vào hệ thống, liên kết tệp đính kèm vào tin nhắn | | 11. | Hệ thống | Chuyển tiếp tin nhắn đến người nhận theo thời gian thực. Hiển thị tương tự sự kiện 9. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 10.a | Hệ thống | Kiểm tra thấy hai người dùng chưa có phòng chung, tạo phòng mới | | 11.a | Hệ thống | Thực hiện tiếp luồng sự kiện chính từ 10 | | 8.b | Người dùng | Nhấn vào icon tệp vừa tải lên | | 9.b | Hệ thống | Huỷ liên kết tệp với tin nhắn, xoá tệp | | 11.c | Hệ thống | Có lỗi khi tạo tin nhắn, xoá tin nhắn đã hiển thị trước cho người dùng, thông báo có lỗi | | | |
| **Hậu điều kiện** | Không | | |

\* Dữ liệu đầu vào của tin nhắn

**Bảng 2.8** Dữ liệu đầu vào của tin nhắn

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1. | Văn bản tin nhắn | Nội dung tin nhắn bằng văn bản mà người dùng muốn gửi | Có | Là văn bản thông thường, có thể xuống dòng, có thể thêm URL của trang web (URL sẽ tự động thay đổi thành hyperlink) | Đây là trang web <google.com> |
| 2. | Tệp đính kèm | Người dùng có thể tải lên các loại tệp | Không | Giới hạn mỗi tệp 5GB, số tệp mỗi tin nhắn là 100 |  |

### Đặc tả use case Tạo khoá học

**Bảng 2.9** Đặc tả use case Tạo khoá học

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | **UC005** | Tên Use case | **Tạo khoá học** |
| **Tác nhân** | Giảng viên | | |
| **Tiền điều kiện** |  | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Giảng viên | Nhấn nút tạo khoá học | | 2. | Hệ thống | Hiển thị form nhập tên khoá học và chọn loại khoá học | | 3. | Giảng viên | Nhập tên và chọn loại khoá học | | 4. | Hệ thống | Lưu lại khoá học và hiển thị trang soạn nội dung khoá học | | 5. | Giảng viên | Soạn thảo khoá học | | 6. | Hệ thống | Lưu lại khoá học | | 7. | Hệ thống | Thông báo trạng thái lưu | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 5.a | Giảng viên | Chỉnh sửa mô tả khoá học | | 5.a.1 | Giảng viên | Dán ảnh vào trình soạn thảo | | 5.a.2 | Hệ thống | Lưu ảnh vào server lưu trữ | | 5.a.3 | Hệ thống | Lấy URL ảnh đã lưu, gán vào ảnh trong trình soạn thảo | | 5.b | Giảng viên | Chỉnh sửa thời gian khoá học | | 5.c | Giảng viên | Thêm chương học | | 6.c | Giảng viên | Chỉnh sửa tên chương học | | 5.d | Giảng viên | Cài đặt chương học | | 5.d.1 | Giảng viên | Cài đặt thời gian dự kiến của chương học | | 5.d.2 | Giảng viên | Cài đặt thời gian làm bài kiểm tra, giới hạn thời giạn 2 lần làm bài kiểm tra, điểm qua bài kiểm tra | | 5.e | Giảng viên | Thêm bài học | | 6.e | Giảng viên | Chỉnh sửa nội dung bài học | | 7.e | Giảng viên | Thực hiện luồng hoạt động sự kiện thay thế 5.a.1 | | 8.e | Giảng viên | Thêm video bài học nếu là khoá học thông thường | | 9.e | Hệ thống | Lưu video lên server lưu trữ, tạo tài nguyên và gán URL video vào bài học | | 8.e.1 | Giảng viên | Cài đặt thời gian học bài học đối với khoá học trực tuyến | | 5.f | Giảng viên | Thêm câu hỏi | | 6.f | Giảng viên | Soạn nội dung câu hỏi | | 7.f | Giảng viên | Thực hiện luồng hoạt động sự kiện thay thế 5.a.1 | | 5.g | Giảng viên | Thêm câu trả lời | | 6.g | Giảng viên | Đánh dấu câu trả lời đúng | | 7.g | Giảng viên | Soạn nội dung câu trả lời | | 8.g | Giảng viên | Thực hiện luồng hoạt động sự kiện thay thế 5.a.1 | | 5.h | Giảng viên | Chỉnh sửa topic của khoá học | | 5.i | Giảng viên | Chỉnh sửa giá của khoá học | | 5.j | Giảng viên | Xoá câu trả lời | | 5.k | Giảng viên | Xoá câu hỏi | | 5.l | Giảng viên | Xoá bài học | | 5.m | Giảng viên | Xoá bài học trực tuyến | | 5.n | Giảng viên | Xoá chương học | | | |
| **Hậu điều kiện** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Hệ thống | Lưu lại khoá học | | 2. | Hệ thống | Thông báo trạng thái lưu | | 3. | Giáo viên | Tiếp tục soạn thảo, quay lại luồng sự kiện chính 5 | | 4. | Giảng viên | Xuất bản (Publish) khoá học | | 5. | Hệ thống | Thông báo đến người dùng theo dõi giảng viên | | 6. | Hệ thống | Hiển thị khoá học đã publish trong danh sách khoá học của người dùng khác | | | |

\* Dữ liệu đầu vào các thông tin của khoá học

**Bảng 2.10** Dữ liệu đầu vào các thông tin của khoá học

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc** | **Điều kiện hợp lệ** | **Ví dụ** |
| 1. | Tên khoá học | Tên của khoá học sắp tạo | Có | Là văn bản |  |
| 2. | Loại khoá học | Lựa chọn khoá học thông thường hoặc khoá học trực tuyến | Có |  |  |
| 3. | Mô tả khoá học | Mô tả , giới thiệu khoá học sẽ hiển thị ở chế độ preview | Không | Văn bản được định dạng | Đây là mô tả khoá học   * Một số điểm nhấn * Có thể định dạng được * Thêm được ảnh, nhúng video |
| 4. | Thời gian bắt đầu khoá học | Thời gian khoá học bắt đầu | Có đối với khoá học trực tuyến | Văn bản thời gian/timestamp | 2021-06-18T09:50:00  1519211809934 |
| 5. | Thời gian khoá học kết thúc | Thời gian dự kiến khoá học kết thúc | Không | Như trên | Như trên |
| 6. | Tên chương học |  | Không | Văn bản thông thường |  |
| 7. | Thời gian bắt đầu chương học | Thời gian chương học bắt đầu | Không | Văn bản thời gian/timestamp | 2021-06-18T09:50:00  1519211809934 |
| 8. | Thời gian chương học kết thúc | Thời gian dự kiến chương học kết thúc | Không | Như trên | Như trên |
| 9. | Thời gian làm bài | Thời gian giới hạn làm bài tập. Nếu là 0 hoặc null, thời gian làm bài sẽ không có giới hạn | Không | Thời gian theo đơn vị phút | 60 *(60 phút)* |
| 10. | Thời gian giữa 2 lần làm bài | Thời gian giới hạn giữa 2 lần làm bài gần nhất của mỗi học sinh. Nếu là 0 hoặc null sẽ không có giới hạn, nếu lớn hơn 365, bài kiểm tra chỉ được làm 1 lần | Không | Thời gian theo đơn vị ngày | 2 *(2ngày)* |
| 11. | Điểm qua bài kiểm tra | Điểm nhỏ nhất mà học sinh cần đạt để coi là pass bài kiểm tra tính theo phần trăm | Không | Sô từ 0 đến 100 | 50 *(50 điểm, đúng 50%)* |
| 11. | Tên bài học |  | Không | Văn bản thông thường |  |
| 12. | Nội dung chương học |  | Không | Văn bản được định dạng |  |
| 13. | Video bài học | Với điều kiện là bài học thông thường | Không | URL của video/File video |  |
| 14. | Thời gian bắt đầu bài học | Thời gian bài học bắt đầu. Đối với bài học trực tuyến | Không | Văn bản thời gian/timestamp | 2021-06-18T09:50:00  1519211809934 |
| 15. | Thời gian bài học kết thúc | Thời gian dự kiến bài học kết thúc. Đối với bài học trực tuyến | Không | Như trên | Như trên |
| 16. | Nội dung câu hỏi |  | Không | Văn bản được định dạng |  |
| 17. | Nội dung câu trả lời |  | Không | Văn bản được định dạng |  |
| 18. | Câu trả lời đúng |  | Ít nhất 1 | True/false |  |
| 19. | Topic khoá học |  | Không | ID trong hệ thống |  |
| 20. | Giá khoá học |  | Không | Số thực |  |
| 21. | Câu trả lời xoá |  | Không | ID trong hệ thống |  |
| 22. | Câu hỏi xoá |  | Không | ID trong hệ thống |  |
| 23. | Bài học xoá |  | Không | ID trong hệ thống |  |
| 24. | Bài học trực tuyến xoá |  | Không | ID trong hệ thống |  |
| 25. | Chương học xoá |  | Không | ID trong hệ thống |  |

### Đặc tả use case Tham gia phòng học trực tuyến

**Bảng 2.11** Đặc tả use case Tham gia phòng học trực tuyến

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use case** | **UC006** | Tên Use case | **Tham gia phòng học trực tuyến** |
| **Tác nhân** | Học sinh (Student) | | |
| **Tiền điều kiện** | Học sinh đã mua khóa học | | |
| **Luồng sự kiện chính**  **(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Học sinh | Yêu cầu tham gia phòng học | | 2. | Hệ thống | Kiểm tra học sinh đã mua khóa học | | 3. | Hệ thống | Lấy thông tin kết nối của phòng học đã lưu trong cơ sở dữ liệu | | 4. | Hệ thống | Kiểm tra lịch học của bài học | | 5. | Hệ thống | Thiết lập kết nối luồng stream video | | 6. | Hệ thống | Thiết lập kết nối kênh chat | | 7. | Hệ thống | Hiển thị giao diện cho người dùng | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 3.a | Hệ thống | Học sinh chưa mua khóa học, thông báo và kết thúc luồng sự kiện | | 5.a | Hệ thống | Không phải thời gian học bài học (chưa đến hoặc đã quá thời gian), thông báo cho người dùng | | | |
| **Hậu điều kiện** | Không | | |

## Yêu cầu phi chức năng

### Yêu cầu về bảo mật

Hệ thống có cơ chế xác thực danh tính và phân quyền người dùng theo vai trò. Cơ chế bảo mật đảm bảo việc các tài nguyên hạn chế không thể bị sửa đổi nếu không có quyền hạn chính xác. Mật khẩu người dùng được xử lý qua hàm băm (hash) trước khi lưu trữ vào cơ sơ dữ liệu, vì vậy, không một ai có thể xem được mật khẩu người dùng, kể cả quản trị viên.

Sau khi đăng nhập, người dùng được cung cấp cặp access token và refresh token mã hoá theo chuẩn JWT (JSON Web Token). Mỗi token sẽ có các phần (i) header chứa thông tin định dạng token, thuật toán được sử dụng để ký, header được mã hoá base64, (ii) payload chứa thông tin tài khoản cơ bản của người dùng dưới dạng JSON để cung cấp danh tính cho hệ thống, payload được mã hoá base64, (iii) signature là chữ ký, signature được ký trên header và payload của token bằng thuật toán mã hoá quy định trên header (thường là HS256 hoặc RS256). Signature là phần bảo mật chính của token, hệ thống tạo token và ký lên nó bằng mã bảo mật bí mật, khi nhận lại token từ người dùng, chỉ những token nào hệ thống giải mã được chữ ký và đọc được thông tin hợp lệ mới được chấp nhận và cho phép truy cập. Người dùng cũng không sợ bị lấy cắp access token vì access token có hạn sử dụng khá ngắn, nếu bị lấy cắp token cũ, token này cũng không thể sử dung để truy cập hệ thống bởi vì payload sẽ có thông tin về hạn sử dụng của token này.

### Yêu cầu về giao diện

Giao diện responsive, thân thiện với người dùng, có thể hoạt động tốt trên nhiều thiết bị và nền tảng. Hệ thống được thiết kế frontend tách biệt với backend, giao tiếp bằng REST API và websocket, giúp dễ dàng xây dựng frontend trên nhiều nền tảng (web, mobile, desktop app ...)

### Yêu cầu về kỹ thuật

Hệ thống được xây dựng với cấu trúc dễ dàng chuyển đổi và mở rộng theo chiều ngang (microservice). Các service có sự phân hóa và giao tiếp với nhau qua interface chung, đảm bảo tính chất open/close, dễ dàng mở rộng, thay đổi các service riêng biệt.

# Công nghệ sử dụng

## Frontend

### React

React[[1]](#footnote-1) là một thư viện javascript được xây dựng bởi Facebook, thích hợp để xây dựng UI cho các single-page application (SPA).

Các trang web viết bằng React có khả năng lấy dữ liệu từ server mà không cần tải lại trang giúp tăng trải nghiệm người dùng.

## Backend

### Laravel

Laravel[[2]](#footnote-2) là một PHP framework, có mã nguồn mở và miễn phí, được xây dựng nhằm hỗ trợ phát triển các phần mềm, ứng dụng, theo kiến trúc MVC.

Một số tính năng của Laravel là hệ thống package module composer giúp quản lý các thư viện dễ dàng, nhiều cách truy cập cơ sở dữ liệu quan hệ đa dạng và tiện lợi, các tiện ích hỗ trợ triển khai và bảo trì ứng dụng, cấu trúc và cú pháp có nguyên tắc thống nhất. Hiện nay, Laravel đang là PHP framework phổ biến nhất và tốt nhất.

Đây là framework chính để phát triển và xây dựng những nghiệp vụ cơ bản của hệ thống, cung cấp API cho Frontend.

### Flask

Flask[[3]](#footnote-3) là một Python micro web framework, giúp xây dựng hệ thống API nhanh chóng và đơn giản. Flask không yêu cầu tool hay thư viện cụ thể nào nên có thể dễ dàng tích hợp mọi thư viện của python.

Trong hệ thống này, Flask được sử dụng để cung cấp API và train model recommendation system cho hệ thống.

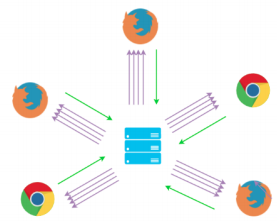
### Websocket

Websocket là giao thức hỗ trợ giao tiếp hai chiều giữa client và server để tạo một kết nối trao đổi dữ liệu. Websocket cung cấp các tính năng realtime như thông báo, bình luận, nhắn tin.

### WebRTC

WebRTC là các API viết bằng javascript giúp giao tiếp theo thời gian thực mà không cần cài plugin hay phần mềm hỗ trợ. WebRTC có khả năng hỗ trợ trình duyệt giao tiếp thời gian thực thông qua Video Call, Voice Call hay transfer data P2P(peer-to-peer), không cần đến plugin, phần mềm khác.

Tuy nhiên trong cài đặt của hệ thống này, các client browser không giao tiếp P2P với nhau mà thông qua một SFU Mediaserver giúp giảm tải cho browser của người dùng và dễ dàng quản lý.



**Hình 8.** Mô hình SFU Media Server*[[4]](#footnote-4)*

### Redis

Redis[[5]](#footnote-5) (REmote DIctionary Server) là một mã nguồn mở được dùng để lưu trữ dữ liệu có cấu trúc, có thể sử dụng như một database, bộ nhớ cache hay một message broker.

Trong hệ thống, Redis được sử dụng như một bộ nhớ cache vì dữ liệu của Redis được lưu trên RAM nên tốc độ đọc ghi rất nhanh. Redis cũng đóng vai trò queue để xử lý lần lượt các job không đồng bộ mà các thành phần khác sinh ra.

### Elasticsearch

Elasticsearch là một search engine (công cụ tìm kiếm) hoạt động như một web server, cung cấp API cho chức năng tìm kiếm.

### Recommendation systems

Recommendation systems (Hệ thống gợi ý) là một dạng của hệ hỗ trợ ra quyết định, cung cấp giải pháp mang tính cá nhân hóa mà không phải trải qua quá trình tìm kiếm phức tạp. Hệ gợi ý học từ người dùng và gợi ý các sản phẩm tốt nhất trong số các sản phẩm phù hợp.

Các sản phẩm được gợi ý dựa trên số lượng sản phẩm đó đã được bán, dựa trên các thông tin cá nhân của người sử dụng, dựa trên sự phân tích hành vi mua hàng trước đó của người sử dụng và những người sử dụng khác để đưa ra các dự đoán về hành vi mua hàng trong tương lai của khách hàng đó. Có hai phương pháp gợi ý chính, phổ biến là Content-base và Collaborative Filtering.

**Content-based** đánh giá đặc tính của sản phẩm được recommended. Ví dụ: một người dùng xem rất nhiều các bộ phim về cảnh sát hình sự, vậy thì gợi ý một bộ phim trong cơ sở dữ liệu có chung đặc tính hình sự tới người sử dụng này. Cách tiếp cận này yêu cầu việc sắp xếp các sản phẩm vào từng nhóm hoặc đi tìm các đặc trưng của từng sản phẩm. Tuy nhiên, có những items không có nhóm cụ thể và việc xác định nhóm hoặc đặc trưng của từng sản phẩm đôi khi là bất khả thi.

**Collaborative Filtering** gợi ý sản phẩm dựa trên sự tương quan (similarity) giữa các người dùng và/hoặc sản phẩm. Có thể hiểu rằng ở nhóm này một sản phẩm được recommended tới một người dùng dựa trên những người sử dụng có hành vi tương tự. Ví dụ: người dùng A, B, C đều thích các bài hát của Noo Phước Thịnh. Ngoài ra, hệ thống biết rằng người dùng B, C cũng thích các bài hát của Bích Phương nhưng chưa có thông tin về việc liệu người dùng A có thích Bích Phương hay không. Dựa trên thông tin của những người sử dụng tương tự là B và C, hệ thống có thể dự đoán rằng A cũng thích Bích Phương và gợi ý các bài hát của ca sĩ này tới A. Collaborative Filtering (CF) có các loại chính là (i) Memory-based và (ii) Model-based.

Hướng tiếp cận Memory-based sử dụng dữ liệu rating của người dùng để tính toán độ tương tự giữa những người sử dụng hoặc giữa những sản phẩm. Ví dụ điển hình của hướng tiếp cận này là neighbourhood-based CF và item-based/user-based top-N recommendations. Ví dụ trong hướng tiếp cận user-based, giá trị rating dự đoán của người sử dụng *u* cho sản phẩm *i* là tổng hợp dữ liệu rating đã có đối với sản phẩm *i* của một số người dùng tương tự *u.*

Hướng tiếp cận Model-based sử dụng models được xây dựng bằng nhiều giải thuật data mining, machine learning khác nhau để dự đoán rating của sản phẩm chưa được rate. Có rất nhiều giải thuật model-based: Bayesian networks, clustering models, singular value decomposition, Markov decision process based models...

Hệ thống sử dụng hướng tiếp cập memory-based, cụ thể hơn là user-based. Bởi vì (i) Content-based làm cho các gợi ý có tính chất giống nhau và giống với những sản phẩm mà người dùng đã sử dụng. Điều này không phù hợp với hệ thống cung cấp khóa học này vì các khóa học được gợi ý sẽ lặp lại nội dung gần tương tự với các khóa học mà người dùng đã tham gia. (ii) Model-based cần khối lượng tính toán lớn để xây dựng model cho hệ gợi ý. Những hệ thống có thông tin được cập nhật, làm mới nhiều sẽ gây ra tính toán dư thừa, giảm hiệu năng của hệ thống. (iii) User-base lọc cộng tác sẽ gợi ý những sản phẩm dựa trên xu hướng của những người dùng tương tự, giúp gợi ý cho người dùng các khóa học theo trend trong nhóm đối tượng người dùng giống nhau.

Các bước xây dựng lên hệ gợi ý:

* Tạo ma trận hai chiều student-course (R), mỗi ô của ma trận là đánh giá (rating) mà người dùng đánh giá khoá học

**Bảng 3.1** Ma trận student-course

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | s0 | s1 | s2 | s3 | s4 | s5 | s6 |
| **c0** | 5 | 5 | 2 | 0 | 1 | x | x |
| **c1** | 4 | x | x | 0 | x | 2 | x |
| **c2** | x | 4 | 1 | x | x | 1 | 1 |
| **c3** | 2 | 2 | 3 | 4 | 4 | x | 4 |
| **c4** | 2 | 0 | 4 | x | x | x | 5 |

* Vì mỗi student sẽ có tính cách khác nhau – có người xu hướng đánh giá cao, có người xu hướng đánh giá thấp hơn. Vì vậy, ta cần chuẩn hoá ma trận này bằng cách lấy rating mỗi ô trừ giá trị rating trung bình của một cột. Sau khi chuẩn hoá, giá trị dương tương ứng với student thích course này, giá trị âm là không thích. Những ô course chưa được rate, ta thay thế bằng số 0. Điều này sẽ khiên ma trận hai chiều có dạng ma trận thưa, chúng ta có thể lưu trữ ma trận thưa dưới dạng danh sách liên kết để tiết kiệm bộ nhớ hơn.

**Bảng 3.2** Ma trận student-course đã chuẩn hoá

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | s0 | s1 | s2 | s3 | s4 | s5 | s6 |
| **c0** | 1.75 | 2.25 | -0.5 | -1.33 | -1.5 | 0 | 0 |
| **c1** | 0.75 | 0 | 0 | -1.33 | 0 | 0.5 | 0 |
| **c2** | 0 | 1.25 | -1.5 | 0 | 0 | -0.5 | -2.33 |
| **c3** | -1.25 | -0.75 | 0.5 | 2.67 | 1.5 | 0 | 0.67 |
| **c4** | -1/25 | -2.75 | 1.5 | 0 | 0 | 0 | 1.67 |

* Để tính toán độ tương tự của mỗi student, ta sử dụng hàm cosine để tính góc lệch giữa mỗi vector ***s*** với nhau. Công thức của hàm cosin:

**Công thức 3.1** Hàm cosin

* Để dự đoán rating ***ri,j*** của ***si*** với một course ***cj*,**ta xem xét một số lượng ***k*** student ***sk*** đã rate course ***cj***và có độ tương tự cao nhất với ***si.*** Giá trị rating dự đoán chính là giá trị trung bình rating mỗi ***sk*** nhân với độ tương tự ***si,k***.

**Công thức 3.2** Công thức dự đoán rating

Tính toán các giá trị 0 ở *Bảng 3.2,* ta thu được bảng mới

**Bảng 3.3** Dự đoán giá trị rating còn thiếu

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | s0 | s1 | s2 | s3 | s4 | s5 | s6 |
| **c0** | 1.75 | 2.25 | -0.5 | -1.33 | -1.5 | 0.18 | -0.64 |
| **c1** | 0.75 | 0.48 | -0.17 | -1.33 | -1.33 | 0.5 | 0.05 |
| **c2** | 0.91 | 1.25 | -1.5 | -1.84 | -1.78 | -0.5 | -2.33 |
| **c3** | -1.25 | -0.75 | 0.5 | 2.67 | 1.5 | 0.59 | 0.67 |
| **c4** | -1/25 | -2.75 | 1.5 | 1.57 | 1.56 | 1.59 | 1.67 |

* Đưa các giá trị dự đoán về thang điểm 5 như ban đầu:

**Bảng 3.4** Giá trị dự đoán rating cuối cùng

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | s0 | s1 | s2 | s3 | s4 | s5 | s6 |
| **c0** | 5 | 5 | 2 | 0 | 1 | 1.68 | 2.70 |
| **c1** | 4 | 3.23 | 2.33 | 0 | 1.67 | 2 | 3.38 |
| **c2** | 4.15 | 4 | 1 | -0.5 | 0.71 | 1 | 1 |
| **c3** | 2 | 2 | 3 | 4 | 4 | 2.10 | 4 |
| **c4** | 2 | 0 | 4 | 4.06 | 4.06 | 3.10 | 5 |

## Server

### Amazon web service

Amazon web service[[6]](#footnote-6) (AWS) là dịch vụ điện toán đám mây cung cấp rất nhiều công cụ và tính năng tính năng cho việc deploy (triển khai) hệ thống.

Amazon Virtual Private Cloud (Amazon VPC) giúp tạo ra một môi trường tách biệt, nơi có thể triển khai hệ thống trong một hệ thống mạng ảo riêng tự định nghĩa. Nhà phát triển có toàn quyền quyết định môi trường mạng ảo này sẽ như thế nào, bao gồm lựa chọn dải IP riêng, tạo mạng con (subnet), bảng định tuyến (route table) và cổng kết nối mạng (net gateway), cũng có thể sử dụng cả IPv4 và IPv6 cho bảo mật và dễ dàng truy cập ứng dụng, tài nguyên.

Amazon Elastic Compute Cloud (Amazon EC2) là một cơ sở hạ tầng điện toán đám mây được cung cấp bởi AWS giúp cung cấp tài nguyên máy tính ảo hoá theo yêu cầu.

Amazon Relational Database Service (Amazon RDS) là dịch vụ đám mây do AWS phát triển với mục tiêu cung cấp giải pháp cài đặt, vận hành và mở rộng dành cho cơ sở dữ liệu có quan hệ.

Amazon Simple Storage Service (Amazon S3) là một kho lưu trữ cloud. Nó được thiết kế để lưu trữ dung lượng lớn, chi phí thấp và có server trên khắp thế giới. S3 cung cấp cho nhà phát triển kho lưu trữ bảo mật, tin cậy và khả năng mở rộng cao.

### Microservice - Docker

Docker[[7]](#footnote-7) là một nền tảng để cung cấp cách để building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa).

Các containers cho phép lập trình viên đóng gói một ứng dụng với tất cả các phần cần thiết, chẳng hạn như môi trường, thư viện, và gói tất cả ra dưới dạng một package. Bằng cách đó, ứng dụng sẽ chạy trên mọi máy Linux khác bất kể mọi cài đặt tùy chỉnh mà máy có thể có khác với máy được sử dụng để viết code.

Các phần của hệ thống chạy trên AWS EC2 đều được đóng gói trong Docker container, giúp rút ngắn thời gian và tiết kiệm công sức để cài đặt môi trường deploy.

# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

### Kiến trúc phần mềm

**Kiến trúc MVC**: MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.

Model (M) là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…

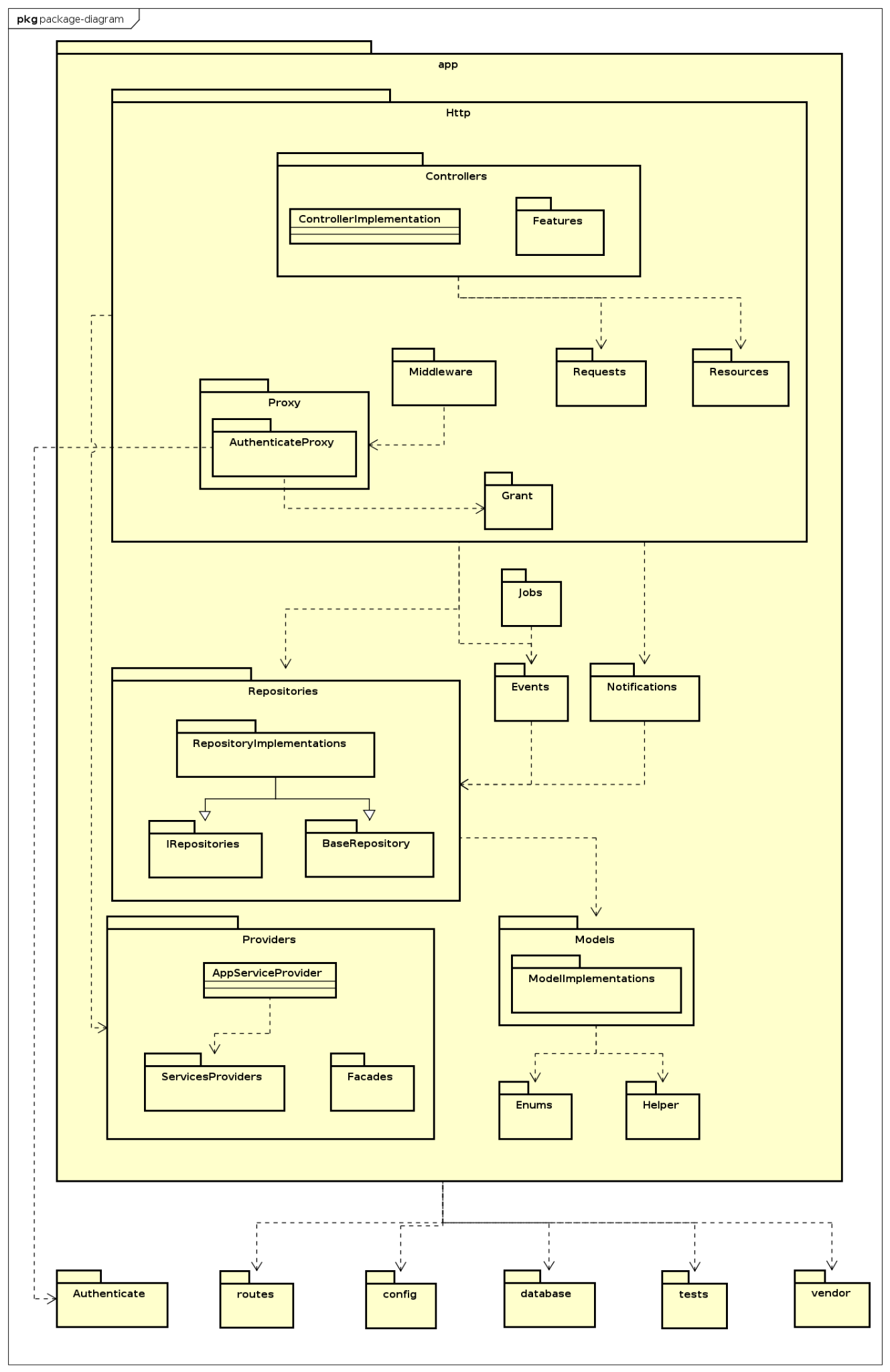
View (V) là phần giao diện dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. View có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller. Thông thường, các ứng dụng web sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Tuy nhiên trong hệ thống này, View là một phần tách rời giao tiếp với controller qua API.

Controller (C) là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

**Repository pattern**: Repository Pattern là lớp trung gian giữa tầng Business Logic (C) và Data Access (M), giúp cho việc truy cập dữ liệu chặt chẽ và bảo mật hơn. Thông thường thì các phần truy xuất, giao tiếp với database nằm rải rác ở trong code, khi bạn muốn thực hiện một thao tác lên database thì phải tìm trong code cũng như tìm các thuộc tính trong bảng để xử lý. Điều này gây lãng phí thời gian và công sức rất nhiều. Với Repository design pattern, thì việc thay đổi ở code sẽ không ảnh hưởng quá nhiều công sức chúng ra chỉnh sửa.

**Redux store**

### Thiết kế tổng quan



**Hình 4.1** Sơ đồ gói backend

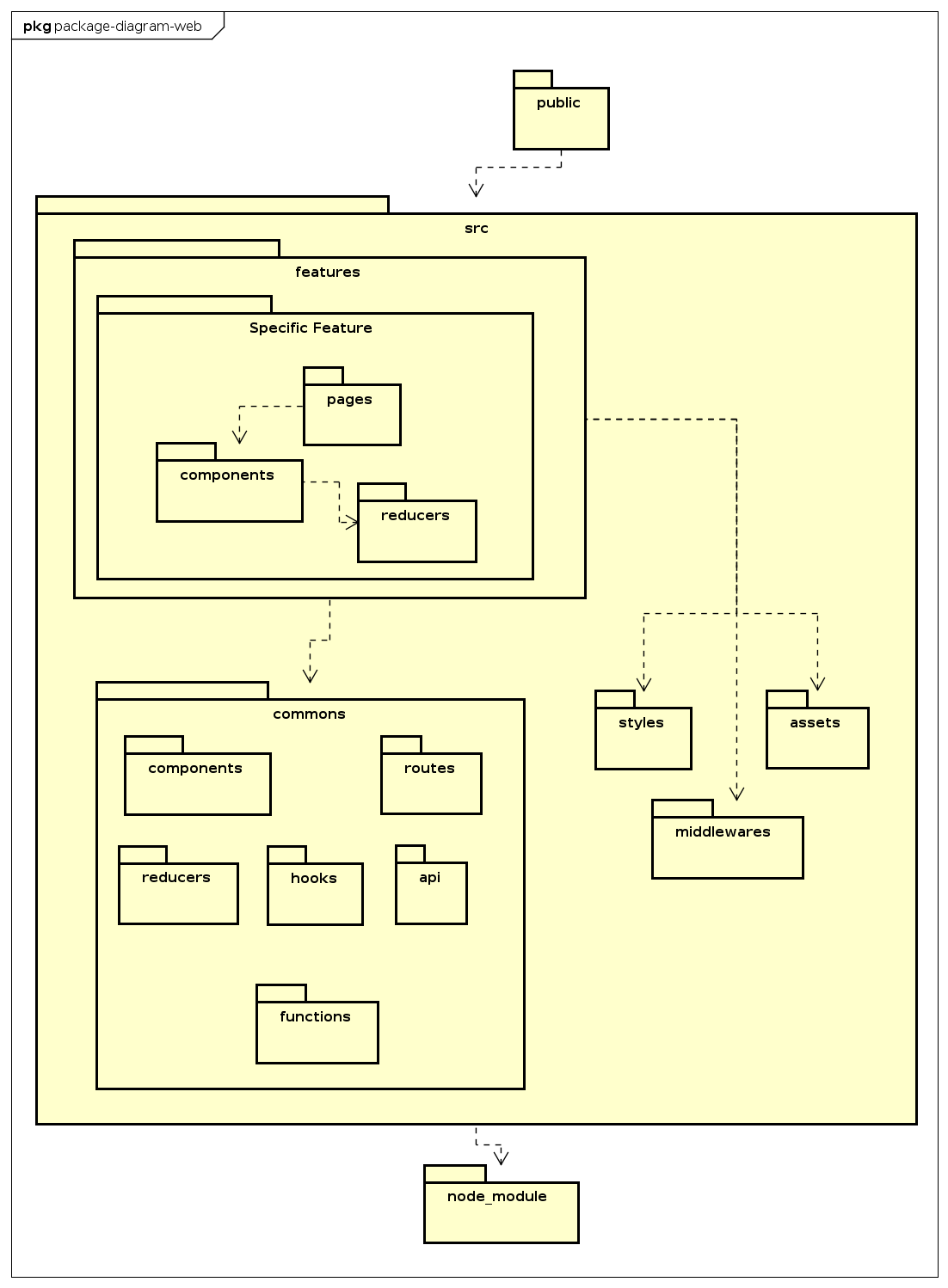
Package Models: Chứa các Class Model - các lớp có thực thể là mapping từ database chứa dữ liệu nghiệp vụ.

Package Repository: Chứa các Model Repository, có nhiệm vụ thực hiện các logic truy vấn cơ sở dữ liệu. Làm trung gian giữa Controller và Model để tránh phụ thuộc giữa Controller và Model.

Package Service Provider: Đăng ký và khởi tạo sẵn các entity của Service và Facade được sử dụng của hệ thống. Các Service được đăng ký có thể inject vào Controller, tránh gây phụ thuộc giữa các Class, đảm bảo tính loose coupling.

Package Controllers: Chứa các Controller của hệ thống, thực hiện các logic nghiệp vụ. Controller nhận request từ view, xử lý nghiệp vụ với các Model thông qua Repository và lưu trữ, trả lại kết quả xử lý cho controller.

Jobs, Events, Notification là cá tác vụ, thông báo sinh ra trong quy trình của những hoạt động khác. Những tác vụ này thường là những tác vụ khá nặng, tốn nhiều thời gian để làm việc và không cần user chờ đợi kết quả trong mỗi request nên chúng thường được xếp vào hàng đợi queue của hệ thống trong một luồng riêng để chạy bất đồng bộ với request của user. Những tác vụ này có thể là gửi Email, gửi thông điệp websocket, upload file lên storage server…



**Hình 4.2** Sơ đò gói frontend

Như đã đề cập trong *Chương 3,* frontend của hệ thống sử dụng thư viện React và quản lý state bằng Redux. Cấu trúc của project gồm có (i) node\_module là gói chứa những thư viện javascript được import và sử dụng, (ii) gói src chứa mã nguồn chính của project, (iii) gói public chứa tất cả tài nguyên công khai mà người dùng có thể truy cập được trên browser, mã nguồn trong gói src sau khi được biên dịch thành mã javascript browser có thể hiểu sẽ chuyển đến gói này.

Mã nguồn của project được chia thành các feature riêng biệt, trong feature có các pages tương ứng với mỗi màn hình giao diện người dùng. Các pages này sử dụng các components riêng bên trong cùng features để xây dựng nên giao diện và reducers để tác động lên luồng dữ liệu.

Gói commons chứa các tài nguyên dùng chung của các feature, gồm các (i) pure function, (ii) shared component, (iii) reducers, (iv) custom hooks, (v) api.

Các reducers dùng chung và mỗi reducers của các features đều được đăng ký vào một root reducer chung của hệ thống, quản lý bởi redux. Root reducer này sau đó được sử dụng bởi store – nơi quản lý luồng hoạt động của tất cả dữ liệu phía frontend. Mỗi action (hành động tác động đến dữ liệu store) đều được đi qua các middleware. Một middleware quan trọng là Authenticate Middleware, có nhiệm vụ kiểm tra hạn access token cho các action cần liên lạc với backend. Nếu access token đã hết hạn, middleware sẽ gửi refresh token về backend để nhận token mới.

## Thiết kế chi tiết

### Thiết kế giao diện

Giao diện được thiết kế tối ưu cho các thiết bị có màn hình lớn như laptop và máy tính để bàn. Bên cạch đó, giao diện cũng đáp ứng được cho thiết bị nhỏ hơn như điện thoại thông minh hay máy tính bảng. Thông tin trong giao diện đáp ứng đầy đủ các thông tin mà người dùng cần đồng thời che dấu các thông tin không cần thiết.

Giao diện ứng dụng có tính tương tác cao với người dùng, phản hồi thông tin khi người dùng tương tác với hệ thống. Ngoài ra giao diện ứng dụng mang tính gợi ý cho người dùng, giúp người dùng có thể tương tác với hệ thống mà không cần tìm hiểu nhiều bằng cách sử dụng các icon, các thanh điều hướng.

Về màu sắc, ứng dụng đáp ứng màu sắc tùy thuộc vào từng loại dữ liệu cũng như ngữ cảnh sử dụng. Ví dụ đối với lỗi thì có màu đỏ, thành công thì có màu xanh. Ngoài ra ứng dụng sử dụng tông màu dễ chịu với người dùng, không sử dụng nhiều các màu sắc gây lo lắng cho người dùng như màu đỏ, màu vàng. Bên cạnh việc xây dựng ứng dụng thân thiện với người dùng thì việc xây dựng ứng dụng cũng hạn chế tối đa việc sử dụng quá nhiều màu sắc, và ưu tiên xây dựng giao diện với màu sắc đơn giản nhưng bắt mắt.

Giao diện chung ứng dụng gồm có bốn phần chính là (i) header - phần thanh tiêu đề trang, (ii) side bar - phần menu điều hướng, (iii) content - phần nội dung trang. Bố cục giao diện ứng dụng cho hai loại thiết bị được trình bày trong *Hình 4.3* và *Hình 4.4*.

Diagram

Description automatically generated

**Hình 4.3** Giao diện thiết bị bàn hình lớn

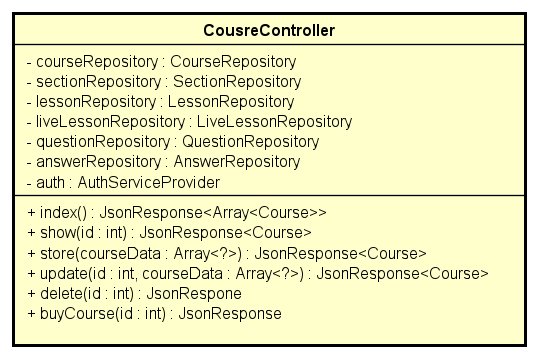
A picture containing graphical user interface

Description automatically generated

**Hình 4.4** Giao diện thiết bị bàn hình nhỏ

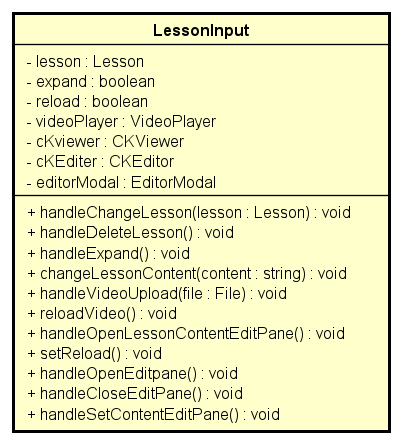
### Thiết kế lớp

*Hình 4.5*, *Hình 4*.6 và *Hình 4.7* trình bày thiết kế lớp của ba lớp đặc trưng trong nghiệp vụ chỉnh sửa khoá học



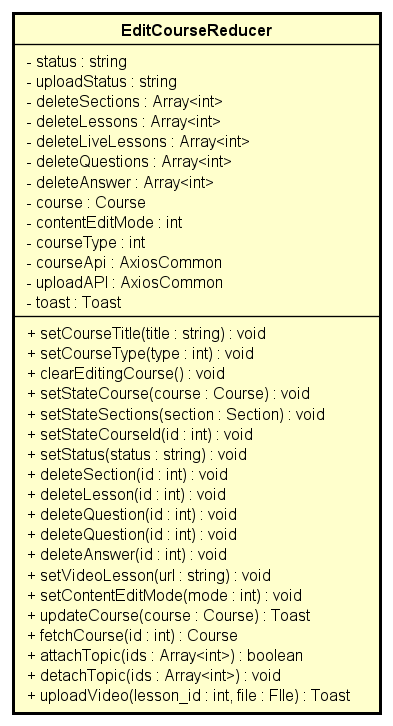
**Hình 4.5** Thiết kế lớp CourseController phía backend

Lớp CourseController là lớp nằm ở phía backend, có nhiệm vụ xử lý các nghiệp vụ liên quan đến khoá học. Cụ thể lớp này (i) cung cấp danh sách khoá học, (ii) cung cấp thông tin chi tiết khoá học, (iii) thêm mới khoá học, (iv) chỉnh sửa khoá họcm (iv) xoá khoá học, (v) mua khoá học.



**Hình 4.6** Thiết kế lớp LesonInput phía frontend

Lớp LessonInput tạo ra cho người dùng giao diện trực quan để quản lý, chính sửa bài học thông thường. Lớp này cung cấp các giao diện (i) hiển thị tên khoá học, (ii) hiển thị nội dung khoá học được định dạng, (iv) hiển thị popup chỉnh sửa nội dung bài học, (v) hiển thị video khoá học. Ngoài ra, giao diện này còn có các nút bấm tương tác (i) xoá khoá học, (ii) tải lên video, (iii) mở giao diện chỉnh sửa nội dung bài học, (iv) mở popup chỉnh sửa nội dung khoá học.

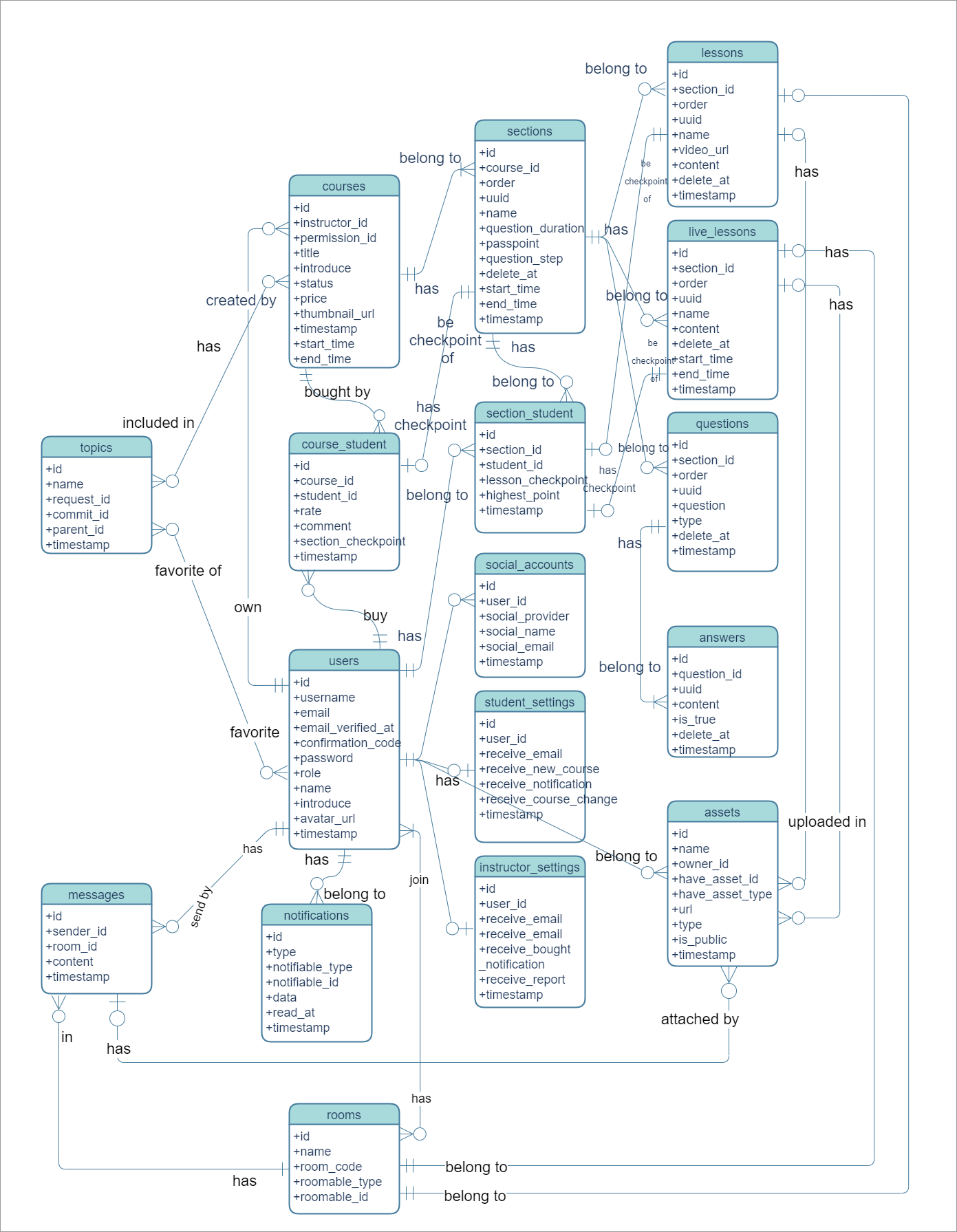


**Hình 4.7** Thiết kế lớp EditCourseReducer phía frontend

Lớp EditCourseReducer là lớp có nhiệm vụ khởi tạo state, quản lý luồng hoạt động của dữ liệu khoá học đang được *giảng viên* chỉnh sửa ở phía frontend và gửi dữ liệu này đến backend để lưu trữ.

### Thiết kế cơ sở dữ liệu

#### Sơ đồ thực thể liên kết



**Hình 4.8** Sơ đồ thực thể liên kết

#### Thiết kế chi tiết cơ sở dữ liệu

Chi tiết các bảng trong cơ sở dữ liệu:

* Users

**Bảng 4.1** Thiết kết chi tiết bảng ues

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| name | varchar(255) |  |  |
| username | varchar(255) |  |  |
| email | varchar(255) |  |  |
| email\_verified\_at | timestamp |  |  |
| confirmation\_code | varchar(255) |  |  |
| introduce | text |  |  |
| password | varchar(255) |  |  |
| avatar\_url | varchar(255) |  |  |
| role | tinyint |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Social Accounts

**Bảng 4.2** Thiết kết chi tiết bảng accounts

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| user\_id | unsigned int | Foreign key |  |
| social\_id | varchar(255) |  |  |
| social\_provider | varchar(255) |  |  |
| social\_name | varchar(255) |  |  |
| social\_email | varchar(255) |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Student settings

**Bảng 4.3** Thiết kết chi tiết bảng student\_settings

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **user\_id** | int | Primary key, Foreign key | not null, auto\_increment |
| receive\_email | varchar(255) |  | default '1' null, |
| receive\_follower\_new\_course | tinyint(1) |  | default 0 null |
| receive\_notification | tinyint(1) |  | default 0 null |
| receive\_course\_change | tinyint(1) |  | default 0 null, |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Instructor settings

**Bảng 4.4** Thiết kết chi tiết bảng instructor\_setting

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **user\_id** | int | Primary key, Foreign key | not null, auto\_increment |
| receive\_email | varchar(255) null, |  |  |
| receive\_bought\_notification | tinyint(1) default 0 null, |  |  |
| receive\_report | tinyint(1) default 0 null, |  |  |
| created\_at | timestamp null, |  |  |
| updated\_at | timestamp null |  |  |

* Notifications

**Bảng 4.5** Thiết kết chi tiết bảng notifications

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| type | varchar(255) |  | not null |
| notifiable\_type | varchar(255) |  | not null |
| notifiable\_id | bigint unsigned | Foreign key | not null |
| data | text |  | not null |
| read\_at | timestamp |  | null |
| created\_at | timestamp |  | null |
| updated\_at | timestamp |  | null |

* Assets

**Bảng 4.6** Thiết kết chi tiết bảng assets

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| name | varchar(255) |  | not null |
| owner\_id | int | Foreign key | not null |
| have\_asset\_id | int | Foreign key | null |
| have\_asset\_type | int | Foreign key | null |
| url | varchar(255) |  | not null |
| type | varchar(255) |  |  |
| size | int |  |  |
| created\_at | timestamp |  | null |
| updated\_at | timestamp |  | null |

* Messages

**Bảng 4.7** Thiết kết chi tiết bảng mesages

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| sender\_id | int | Foreign key | not null |
| room\_id | int | Foreign key | not null |
| content | varchar(255) |  | not null |
| created\_at | timestamp |  | null |
| updated\_at | timestamp |  | null |

* Rooms

**Bảng 4.8** Thiết kết chi tiết bảng rooms

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| name | varchar(255) |  | null |
| room\_code | varchar(255) |  | null |
| roomable\_type | varchar(255) |  | null |
| roomable\_id | int | Foreign key | null |
| created\_at | timestamp |  | null |
| updated\_at | timestamp |  | null |

* Room user

**Bảng 4.9** Thiết kết chi tiết bảng room\_user

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **room\_id** | unsigned int | Primary key | not null, |
| **user\_id** | unsigned int | Primary key | not null, |
| created\_at | timestamp |  | null |
| updated\_at | timestamp |  | null |

* User topics

**Bảng 4.10** Thiết kết chi tiết bảng user\_topic

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| user\_id | int | Foreign key | not null |
| course\_id | int | Foreign key | not null |
| created\_at | timestamp |  | null |
| updated\_at | timestamp |  | null |

* Topics

**Bảng 4.11** Thiết kết chi tiết bảng topics

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| name | varchar(255) |  | not null |
| request\_id int null, | timestamp |  | null |
| commit\_id int null, | timestamp |  | null |
| created\_at | timestamp |  | null |
| updated\_at | timestamp |  | null |

* Course topic

**Bảng 4.12** Thiết kết chi tiết bảng course\_topic

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| course\_id | int | Foreign key | not null |
| topic\_id | int | Foreign key | not null |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Courses

**Bảng 4.13** Thiết kết chi tiết bảng courses

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| instructor\_id | unsigned int | Foreign key | not null |
| status | int |  | default 0 null |
| title | varchar(255) |  |  |
| thumbnail\_url | varchar(255) |  |  |
| introduce | text |  |  |
| price | double |  | default 0 null |
| start\_time | datetime |  |  |
| end\_time | datetime |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Course student

**Bảng 4.14** Thiết kết chi tiết bảng course\_student

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| course\_id | int | Foreign key | Not null |
| student\_id | int | Foreign key | Not null |
| rate | int |  |  |
| comment | varchar(255) |  |  |
| section\_checkpoint | int | Foreign key | not null |
| deleted\_at | timestamp |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Sections

**Bảng 4.15** Thiết kết chi tiết bảng sections

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| course\_id | int |  | not null |
| order | int |  | not null |
| uuid | varchar(255) |  |  |
| name | varchar(255) |  |  |
| start\_time | datetime |  |  |
| end\_time | datetime |  |  |
| question\_duration | int |  |  |
| pass\_point | double(3, 2) |  |  |
| question\_step | int |  |  |
| deleted\_at | timestamp |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Section student

**Bảng 4.16** Thiết kết chi tiết bảng section\_student

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| section\_id | int | Foreign key | Not null |
| student\_id | int | Foreign key | Not null |
| highest\_point | double |  |  |
| lesson\_checkpoint | int | Foreign key | not null |
| deleted\_at | timestamp |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Lessons

**Bảng 4.17** Thiết kết chi tiết bảng lessons

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| section\_id | int | Foreign key | not null |
| name | varchar(255) |  |  |
| uuid | varchar(255) |  |  |
| estimate\_time | int |  |  |
| video\_url | varchar(255) |  |  |
| order | int |  |  |
| content | mediumtext |  |  |
| deleted\_at | timestamp |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Live Lessons

**Bảng 4.18** Thiết kết chi tiết bảng live\_lessons

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| section\_id | int | Foreign key | not null |
| uuid | varchar(255) |  |  |
| name | varchar(255) |  |  |
| start\_time | datetime |  |  |
| end\_time | datetime |  |  |
| order | int |  |  |
| deleted\_at | timestamp |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Questions

**Bảng 4.19** Thiết kết chi tiết bảng quesstions

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| section\_id | int | Foreign key | not null |
| uuid | varchar(255) |  |  |
| question | mediumtext |  |  |
| type | int |  | default 0 |
| order | int |  |  |
| deleted\_at | timestamp |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

* Answers

**Bảng 4.20** Thiết kết chi tiết bảng answers

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Ghi chú |
| **id** | unsigned int | Primary key | not null, auto\_increment |
| uuid | varchar(255) |  |  |
| question\_id | int | Foreign key | not null |
| content | text |  |  |
| is\_true | tinyint(1) |  | default |
| deleted\_at | timestamp |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

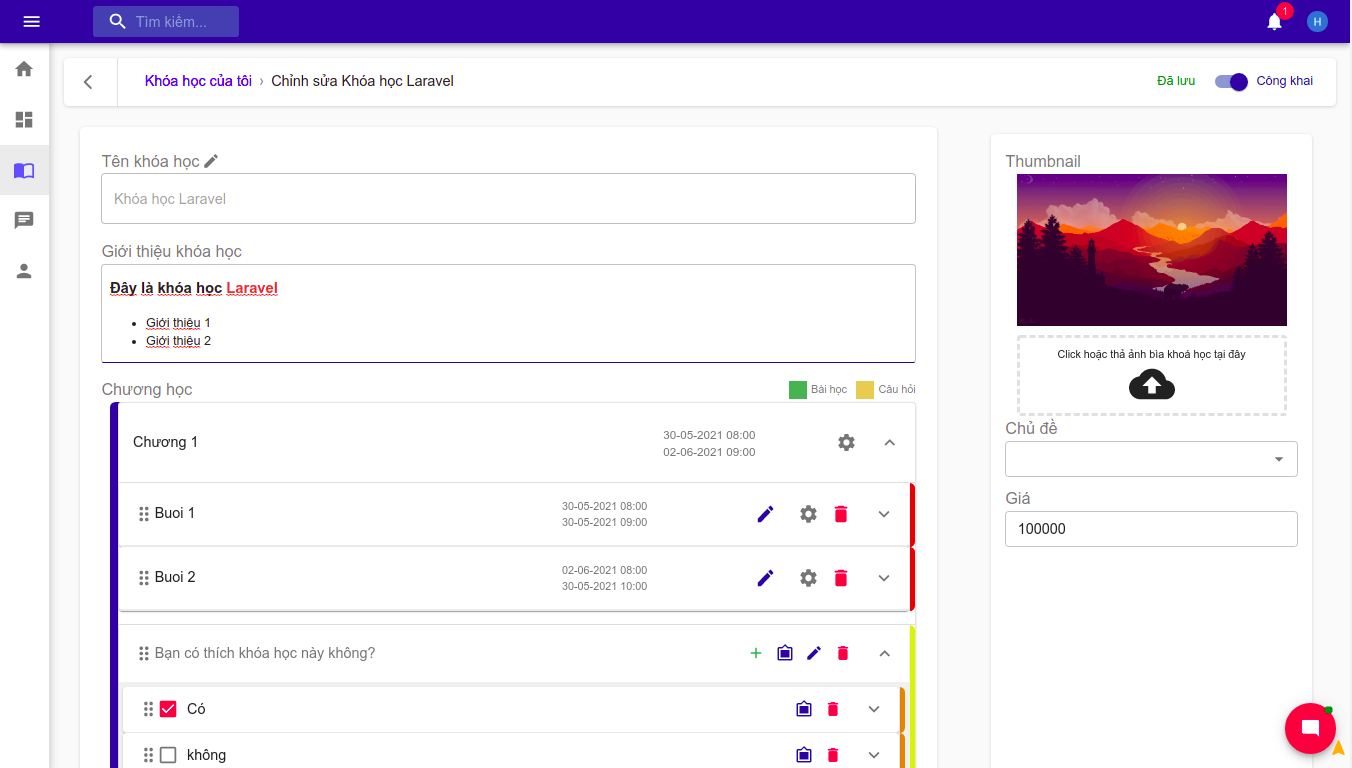
Các công cụ và thư viện đã được sử dụng để hoàn thành đồ án được trình bày trong *Bảng 4.1* dưới đây:

**Bảng 4.21** Thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Visual Studio Code | https://code.visualstudio.com/ |
| IDE lập trình | PHPStorm | https://www.jetbrains.com/phpstorm/ |
| Xây dựng backend chính | Laravel | https://laravel.com/ |
| Xây dụng server gợi ý | Flask | https://flask.palletsprojects.com/en/2.0.x/ |
| Xây dựng SFU mediaserver | Jitsi | https://jitsi.org/ |
| Xây dựng frontend ứng dụng | ReactJS | https://reactjs.org |
| Quản lý state frontend ứng dụng | Redux | https://redux.js.org |
| Database hệ thống | MySQL | https://www.mysql.com/ |
| Triển khai ứng dụng | Amazon web service | https://aws.amazon.com/ |
| Quản lý tên miền, bảo vệ server | Cloudflare | https://www.cloudflare.com/ |
| Ảo hoá server, triển khai ứng dụng | Docker | https://www.docker.com/ |

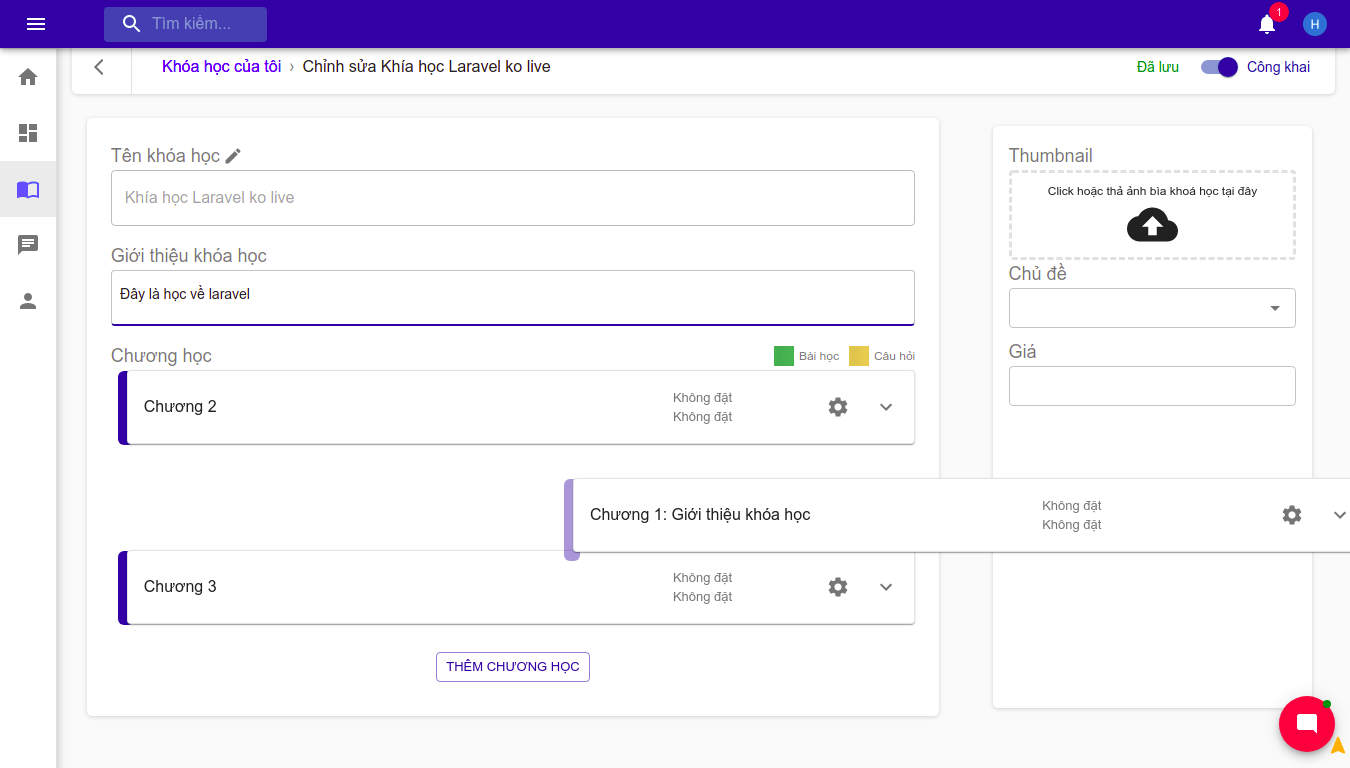
### Kết quả đạt được

### Minh hoạ các chức năng chính



**Hình 4.9** Giao diện soạn thảo khoá học

Các chương, bài, câu hỏi tổ chức dạng accordion có thể thu, phóng để xem thông tin.

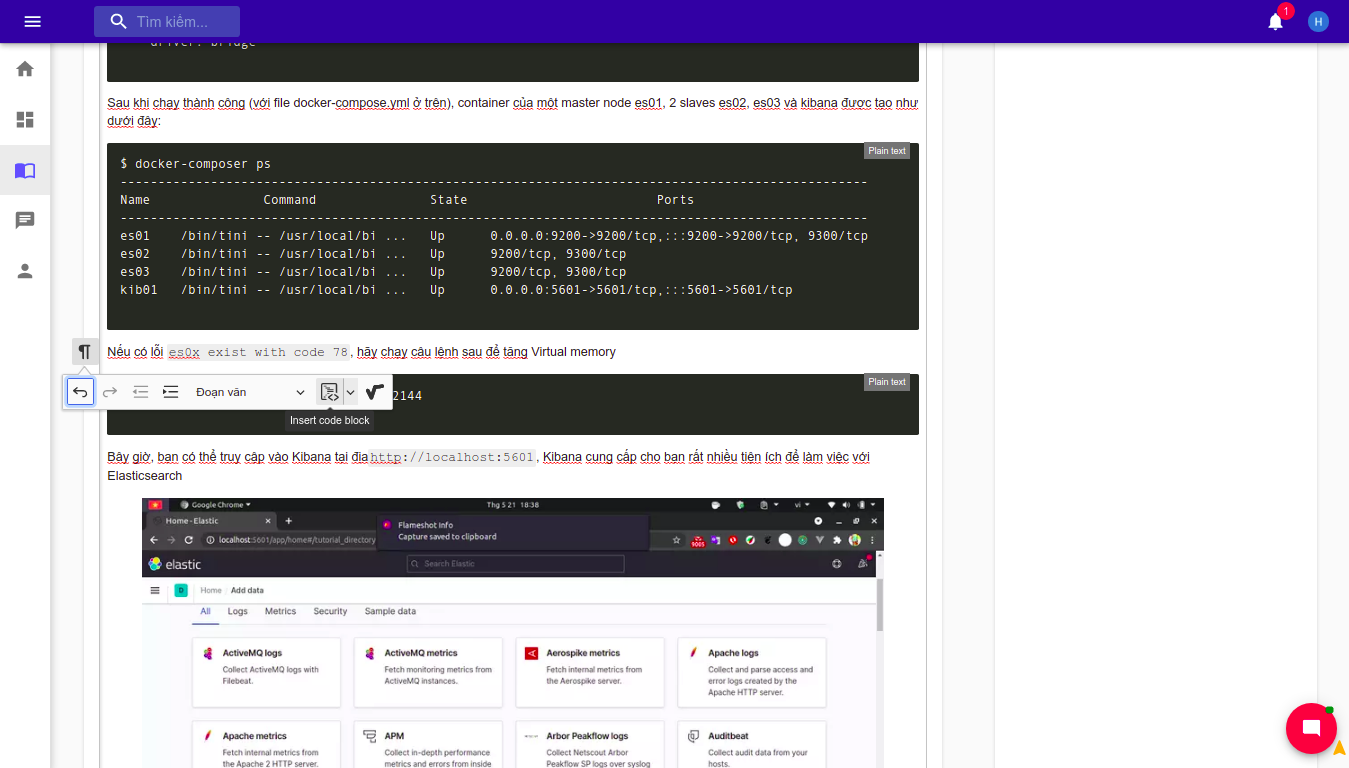


**Hình 4.10** Thao tác kéo, thả



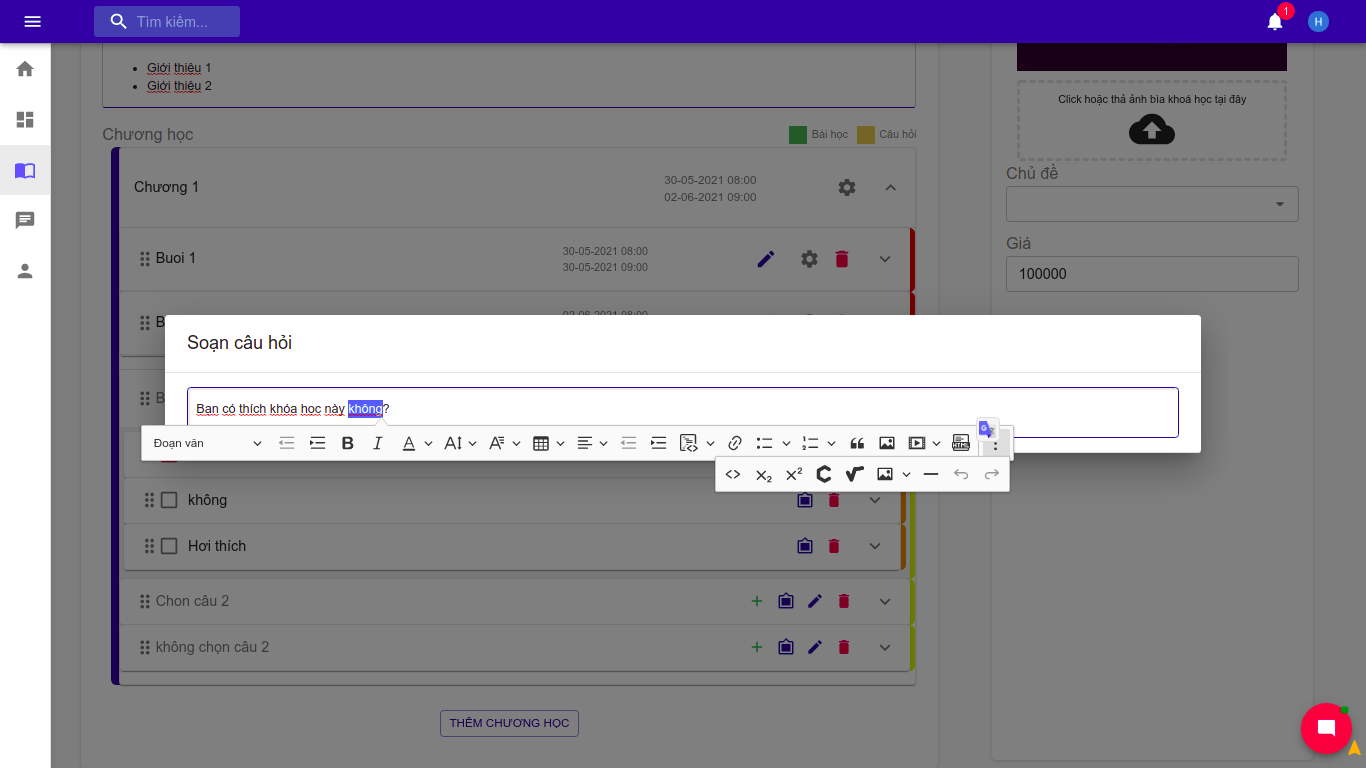
**Hình 4.11** Cài đặt chương học

Các chương học, bài học có thể kéo, thả để sắp xếp vị trí cũng như xoá bỏ. Cài đặt khoá học hiển thị popup mà không cần chuyển trang, thân thiện với người dùng



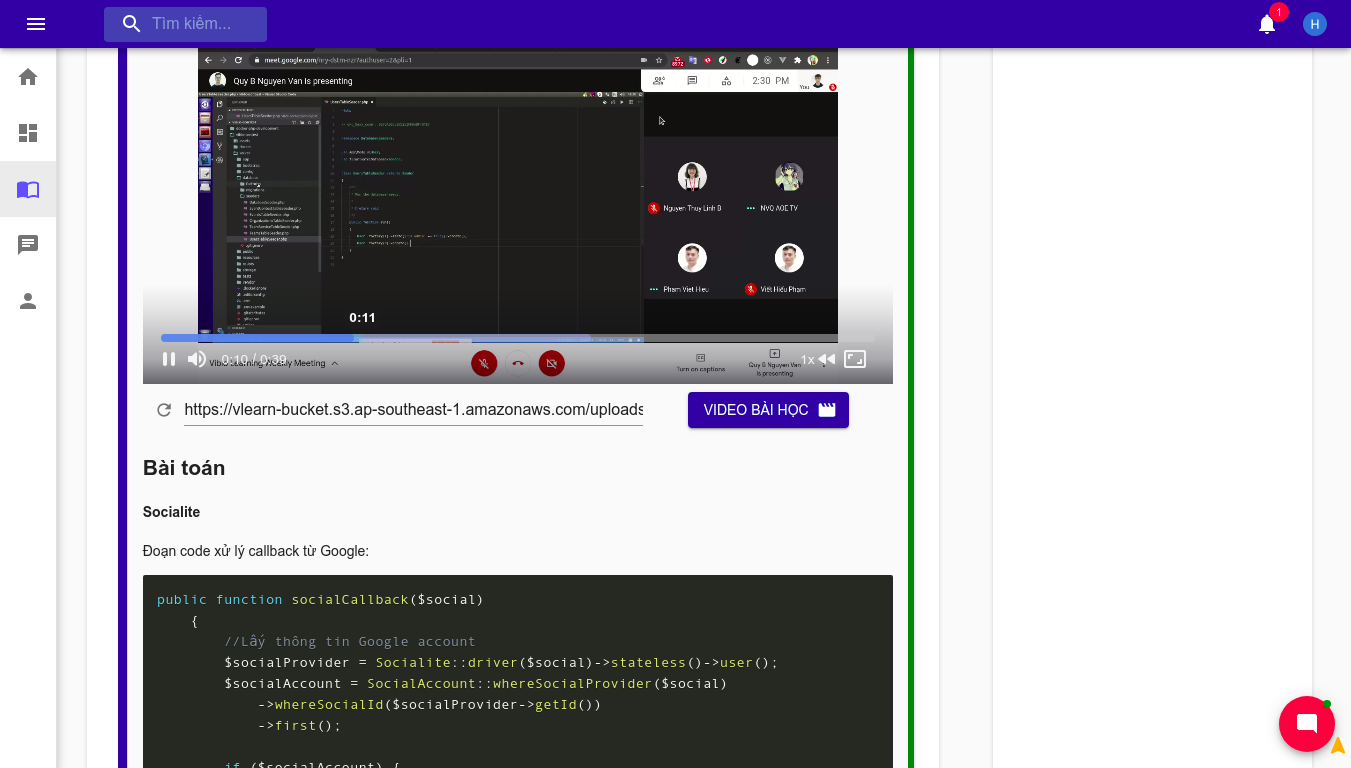
**Hình 4.12** Soạn thảo nội dung định dạng được

Hệ thống sử dụng công cụ soạn thảo với giao diện định dạng thân thiện, có thể định dạng kiểu chữ, màu sắc, cỡ chữ, căn chỉnh, thụt lền chèn bảng biểu, hinh ảnh, video, block code, quote…



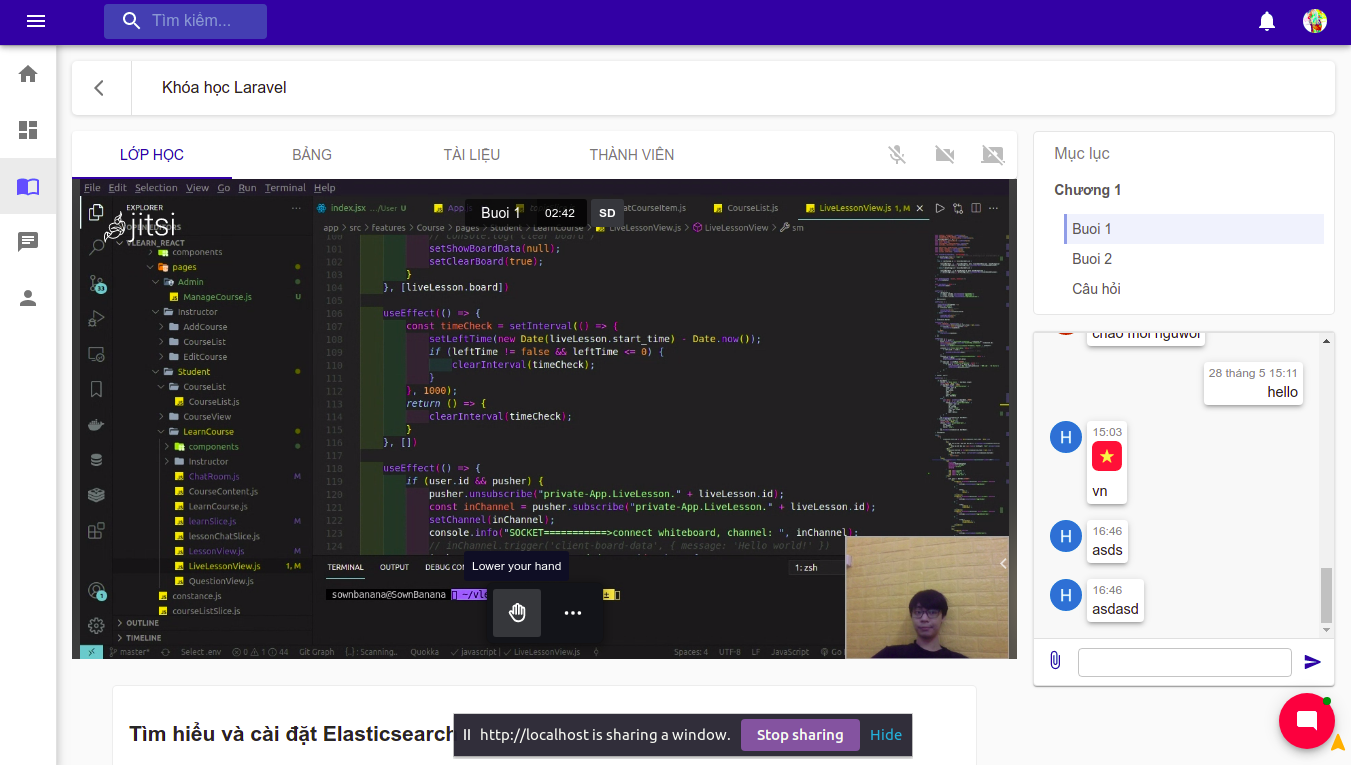
**Hình 4.13** Soạn thảo nội dung trên popup

Hệ thống cũng hỗ trợ chỉnh sửa nhanh chóng nội dung trên popup.

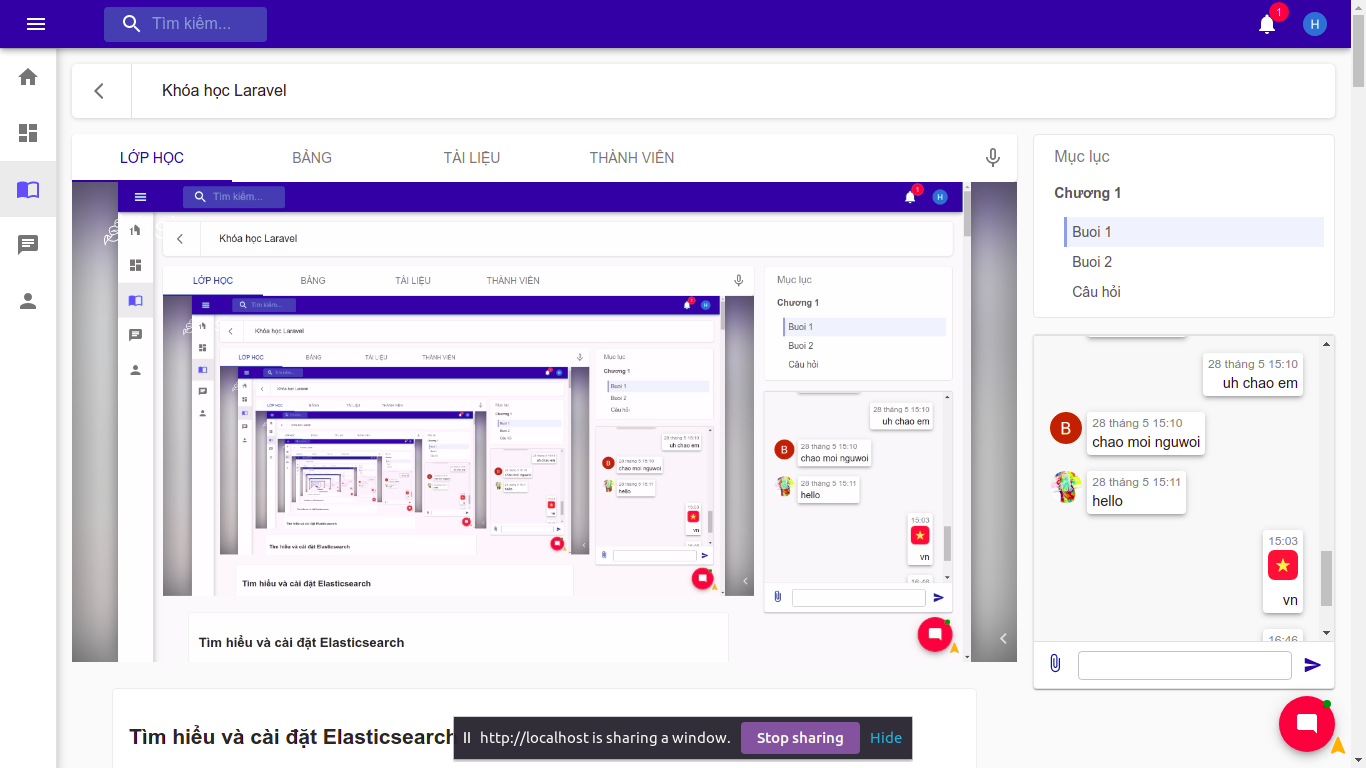


**Hình 4.14** Preview nội dung khoá học

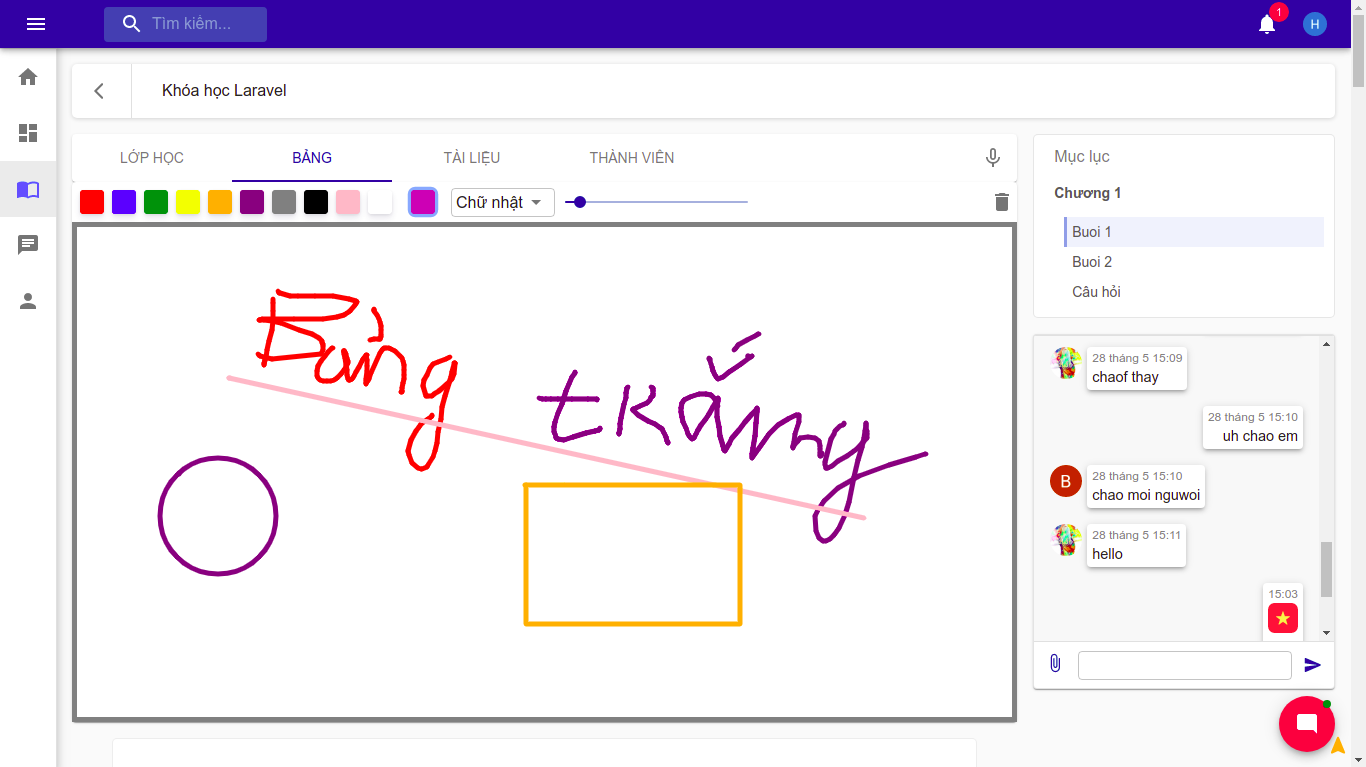
Hệ thống cung cấp sẵn cho *giảng viên* preview những nội dung của khoá học đồng thời trong lúc đang chỉnh sửa.



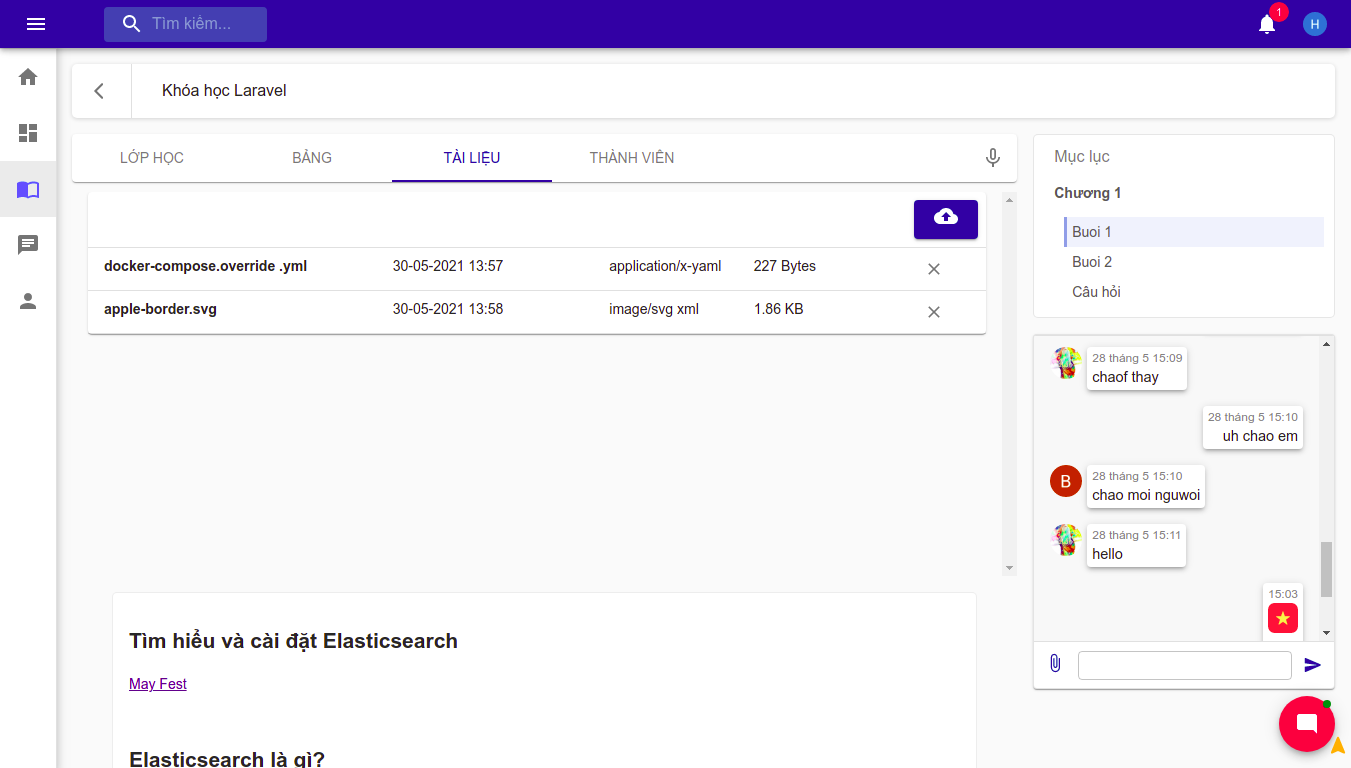
**Hình 4.15** Phòng học trực tuyến



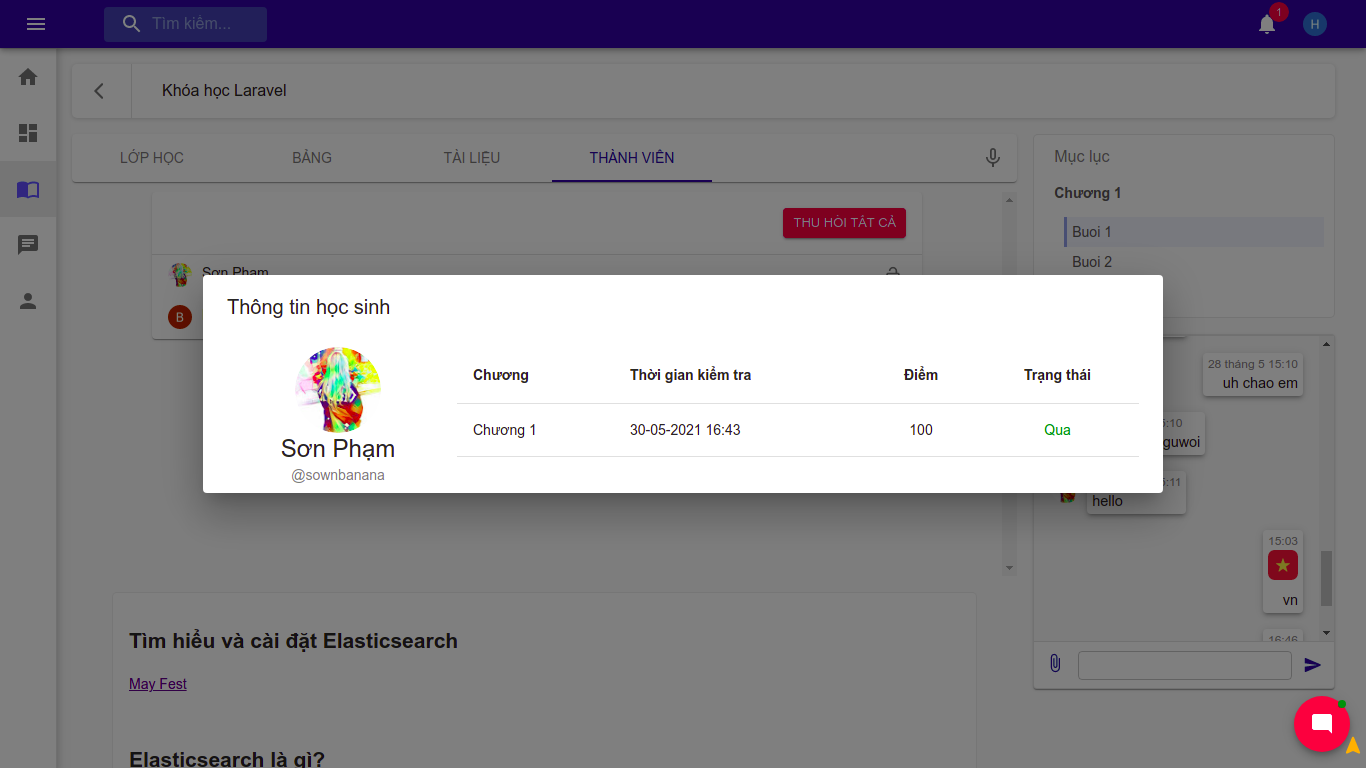
**Hình 4.16** Trình chiếu màn hình



**Hình 4.17** Giao diện bảng trắng

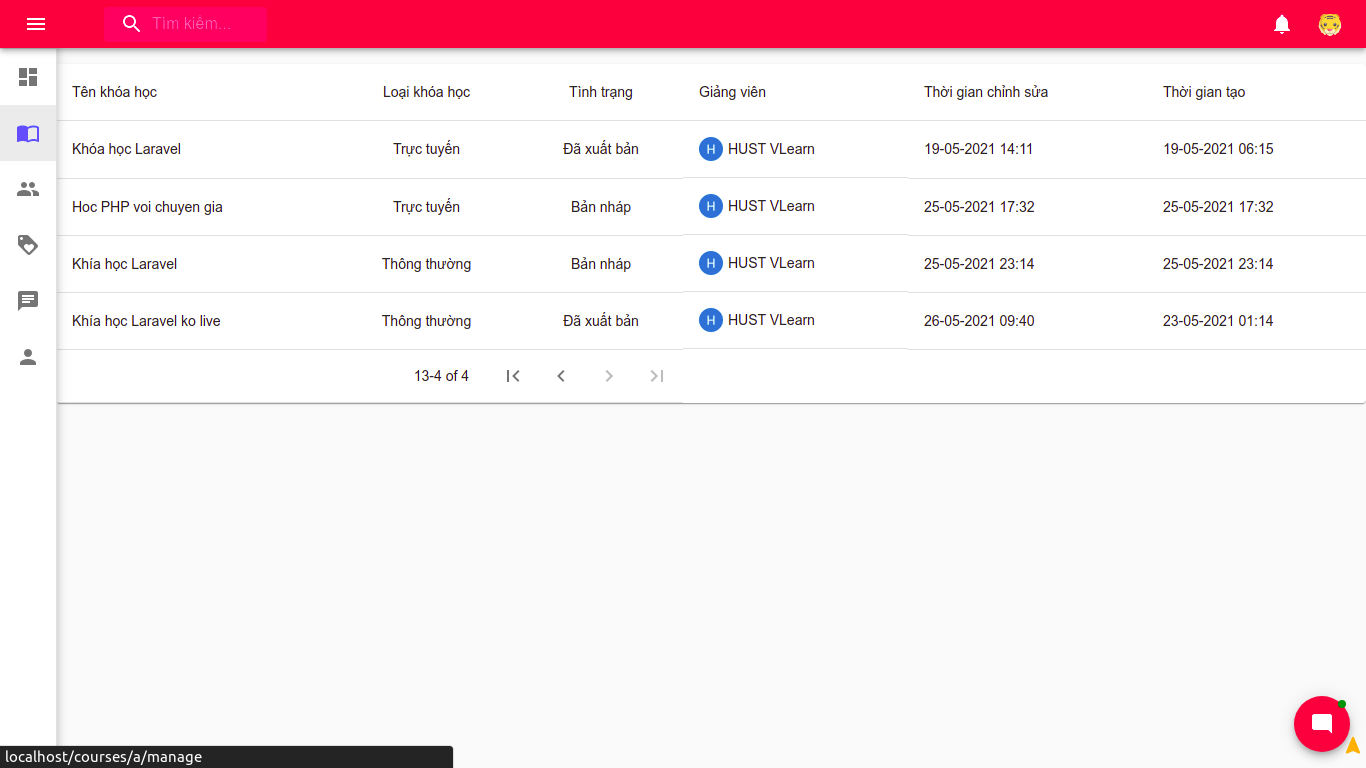


**Hình 4.18** Tài liệu bài học

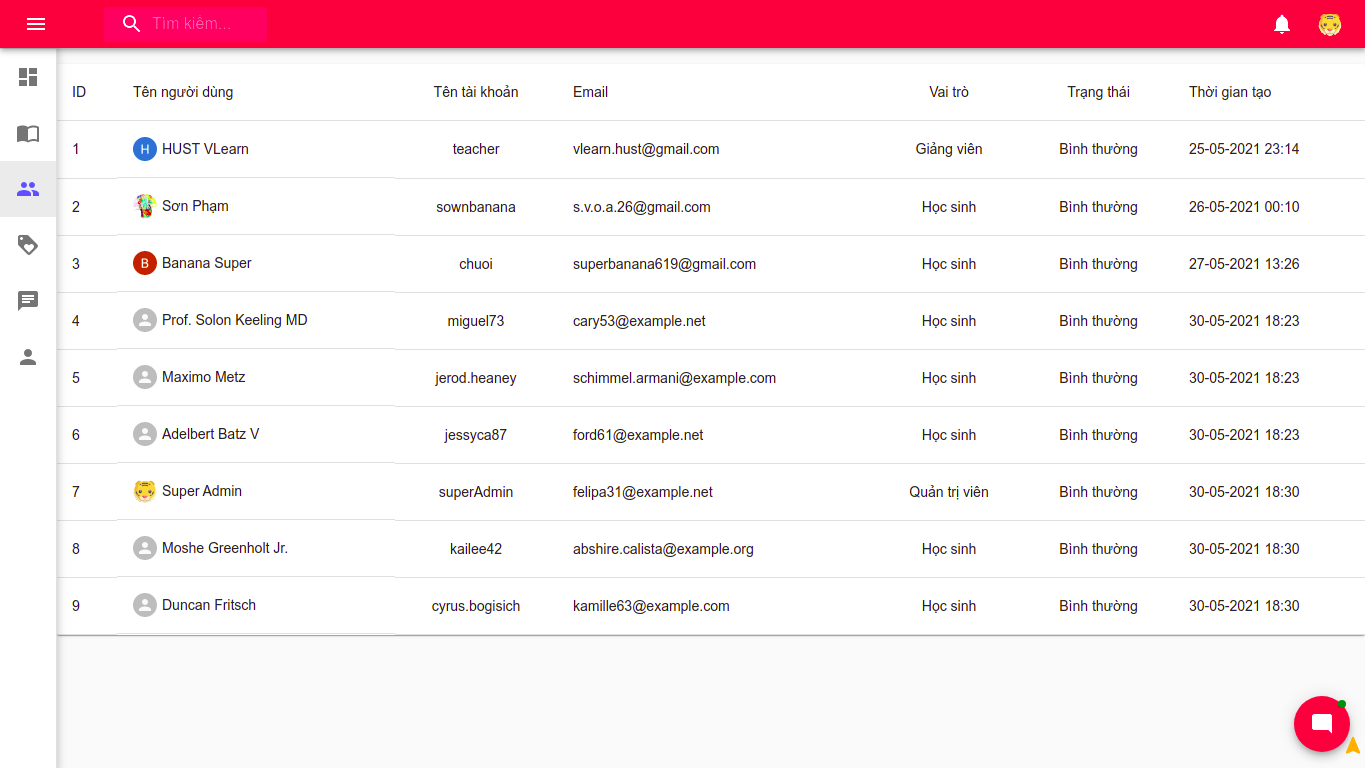


**Hình 4.19** Danh sách và thông tin tiến độ học tập

Giao diện phòng học trực tuyến có nhiều công cụ để hỗ trợ học tập.

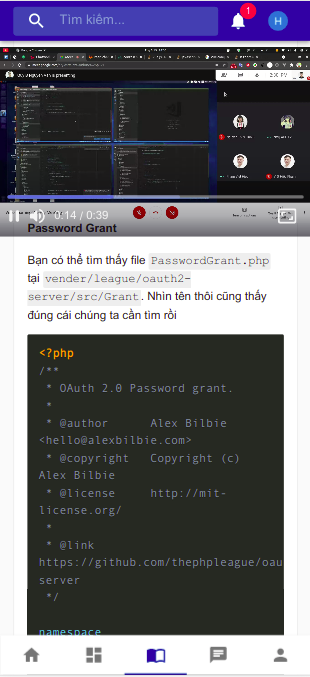


**Hình 4.20** Quản lý khoá học

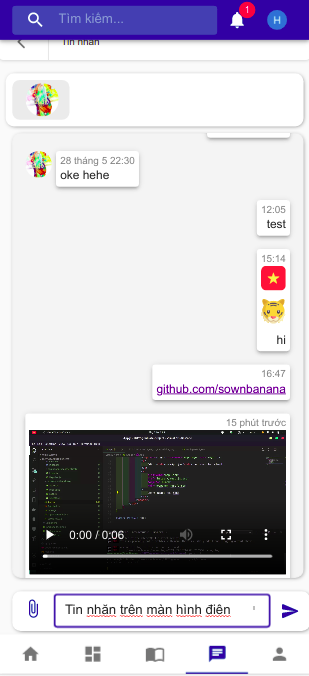


**Hình 4.21** Quản lý người dùng

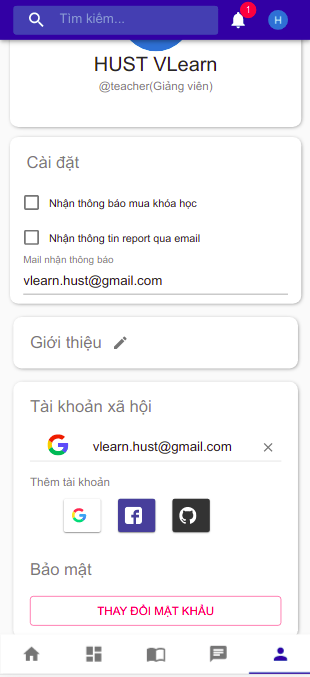
Một số giao diện quản lý của Quản trị viên



**Hình 4.22** Giao diện bài học mobile



**Hình 4.23** Giao diện nhắn tin mobile



**Hình 4.24** Giao diện thông tin cá nhân mobile

Ngoài giao diện cho những thiết bị màn hình lớn như máy tính, laptop, hệ thống cũng hỗ trợ tốt cho các thiết bi di động.

## Triển khai

Hệ thống được xây dựng và triển khai theo hướng microservice, các thành phần được chia nhỏ theo chức năng và triển khai trên các thiết bị riêng biệt. Các thành phần đã triển khai gồm có (i) Backend chính, (ii) File storage server, (iii) Database, (iv) Mediaserver truyền nhận, xử lý dữ liệu livestream, (v) client server.

Diagram

Description automatically generated

**Hình 4.25** Sơ đồ triển khai tổng quan của hệ thống

Hầu hết các thành phần đều sử dụng Docker và triển khai trên hệ thống môi trường ảo của Amazon Web Service:

|  |  |  |
| --- | --- | --- |
|  | Thiết bị | Cấu hình |
| Backend Laravel | AWS EC2 t2.micro | CPU up to3.3 GHz  RAM 1GiB |
| File Storage | AWS S3 | 5GB storage |
| Database | AWS EC2 t2.micro | CPU up to3.3 GHz  RAM 1GiB |
| Mediaserver | AWS EC2 t2.micro | CPU up to3.3 GHz  RAM 1GiB |
| Client web server | AWS EC2 t2.micro | CPU up to3.3 GHz  RAM 1GiB |

1. https://reactjs.org/ [↑](#footnote-ref-1)
2. https://laravel.com/ [↑](#footnote-ref-2)
3. https://flask.palletsprojects.com/ [↑](#footnote-ref-3)
4. Hector Zelaya, [Multi-Party WebRTC Option 3's strategy is SFU](https://webrtc.ventures/2018/07/multi-party-webrtc-option-3-sfu/), 2018 [↑](#footnote-ref-4)
5. https://redis.io/ [↑](#footnote-ref-5)
6. https://aws.amazon.com/ [↑](#footnote-ref-6)
7. https://www.docker.com/ [↑](#footnote-ref-7)