|  |
| --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO  TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HỒ CHÍ MINH  1-02  ĐỒ ÁN CƠ SỞ  NGÀNH CÔNG NGHỆ THÔNG TIN  Đề tài: Xây dựng website quản lý đồ án  cho sinh viên    Giảng viên hướng dẫn: **Nguyễn Mạnh Hùng**  Sinh viên thực hiện:   1. Phạm Trần Mạnh Hùng – 1611061115 2. Nguyễn Nhật Trường - 1611060403     TP.Hồ Chí Minh  2019 |

**PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN C** **ơ S**Ở

* Thông tin sinh viên

Họ và tên sinh viên: **Phạm Trần Mạnh Hùng**

Điện thoại liên lạc: **01687035058** Email: [qmanhhung055@gmail.com](mailto:qmanhhung055@gmail.com)

Lớp: **16DTHC2** Hệ đào tạo: **Đại học chính quy**

Họ và tên sinh viên: **Nguyễn Nhật Trường**

Điện thoại liên lạc: **0972171200** Email: nguyennhattruong199@gmail.com

Lớp: **16DTHC2** Hệ đào tạo: **Đại học chính quy**

Đồ án tốt nghiệp được thực hiện tại: **Bộ môn công nghệ phần mềm – Đại học công nghệ Thành phố HCM.**

Thời gian làm ĐACS: Từ ngày 30/8/2017 đến 20/12/2017

* **Mục đích nội dung của ĐACS**
* Nắm vững kĩ năng và các công nghệ lập trình Web như HML, CSS, JavaScriprt, ASP.net MVC, C#, SQL Sever và các thư viện hỗ trợ khác.
* Nắm vững và sử dụng thành thạo công nghệ WEB cho ứng dụng cần truyền thông dữ liệu da phương tiện theo thời gian thực
* Tìm hiểm và nắm được mô hình hệ thống giáo dục đào tạo trực tuyến
* **Các nhiệm vụ cụ thể của ĐACS**
* Tìm hiểu ngôn ngữ lâp trình web HTML CSS, Javascript,C#, ASP.net MVC
* Tìm hiểu hệ quản trị cơ sở dữ liệu MySQL và các framewok, thư viện hỗ trợ như Jquery, Bootstrap, cách lưu trữ dữ liệu trên Google, GitHub,…
* Tìm hiểu chi tiết về các API hỗ trợ cho Web và sự tương thích với các trình duyệt khác nhau
* Tìm hiểu mô hình quản lý trực tuyến và cách tổ chức quản lý đồ án của sinh viên khoa CNTT
* Lời cam đoan của sinh viên:

Em **Phạm Trần Mạnh Hùng** & **Nguyễn Nhật Trờng** cam kết ĐACS là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *giảng viên* ***Nguyễn Mạnh Hùng****.*

Các kết quả nêu trong ĐACS là trung thực, không phải là sao chép hoàn toàn của bất kỳ công trình nào.

Tp.Hồ Chí Minh , ngày 6 tháng 11 năm 2019

Sinh viên thực hiện ĐACS

* Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐACS và cho phép bảo vệ:

Tp.Hồ Chí Minh , ngày 6 tháng 11 năm 2019

Giáo viên hướng dẫn

**Mục lục**

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570624)

[DANH SÁCH BẢNG BIỂU 6](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570625)

[DANH SÁCH HÌNH VẼ 7](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570626)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 9](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570627)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 10](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570628)

[LỜI CẢM ƠN 11](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570629)

[CHƯƠNG 1: TỔNG QUAN VỀ MÔ HÌNH ĐÀO TẠO TRỰC TUYẾN 12](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570630)

[1.1. Tổng quan về mô hình đào tạo trực tuyến 12](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570631)

[1.2. Bài toán đặt ra 14](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570632)

[1.2.1. Nhu cầu giáo dục trực tuyến 14](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570633)

[1.2.2. Thực trạng việc học trực tuyến 15](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570634)

[1.3. Mục tiêu đề bài 16](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570635)

[1.4. Khó khăn và thách thức 17](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570636)

[1.5. Giải pháp đề xuất 17](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570637)

[CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG 18](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570638)

[2.1. Lựa chọn công nghệ 18](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570639)

[2.2. Giới thiệu tổng quan một số công nghệ sử dụng 18](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570640)

[2.2.1. Công nghệ WebRTC 18](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570641)

[2.2.2. Ngôn ngữ lập trình PHP 22](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570642)

[2.2.3. Framework Laravel 24](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570643)

[2.2.4. Hệ quản trị cơ sở dữ liệu MySQL 27](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570644)

[2.2.5. Giao thức kết nối HTTP 27](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570645)

[2.2.6. Kiến trúc Web Service và Restful Service 28](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570646)

[CHƯƠNG 3: PHÂN TÍCH 31](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570647)

[3.1. Mô hình tổng quát hệ thống 31](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570648)

[3.2. Nội dung yêu cầu 33](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570649)

[3.2.1. Yêu cầu chức năng 33](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570650)

[3.2.2. Yêu cầu phi chức năng 34](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570651)

[3.3. Phân tích và đặc tả yêu cầu 35](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570652)

[3.3.1. Các tác nhân 35](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570653)

[3.3.2. Biểu đồ use-case tổng quan 35](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570654)

[3.3.3. Biểu đồ use-case phân rã 36](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570655)

[3.3.4. Đặc tả các chức năng chính 43](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570656)

[CHƯƠNG 4: THIẾT KẾ 54](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570657)

[4.1. Kiến trúc hệ thống 54](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570658)

[4.2. Biểu đồ trình tự các chức năng chính 56](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570659)

[4.3. Thiết kế cơ sở dữ liệu 62](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570660)

[4.4. Thiết kế lớp 68](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570661)

[4.5. Thiết kế giao diện 73](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570662)

[CHƯƠNG 5: TRIỂN KHAI HỆ THỐNG VÀ KIỂM THỬ 77](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570663)

[5.1. Cài đặt hệ thống 77](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570664)

[5.1.1. Môi trường thử nghiệm hệ thống 77](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570665)

[5.1.2. Cài đặt 77](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570666)

[5.2. Kiểm thử hệ thống 78](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570667)

[5.3. Một số giao diện chức năng 79](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570668)

[5.4. Đánh giá hệ thống 90](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570669)

[CHƯƠNG 6: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 91](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570670)

[6.1. Tổng kết 91](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570671)

[6.2. Hướng phát triển trong tương lai 92](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570672)

[TÀI LIỆU THAM KHẢO 93](file:///C:\Users\nguye\Desktop\csada.docx#_Toc501570673)

**DANH SÁCH BẢNG BIỂU**

Bảng 1. Đặc tả use-case thêm quản lí cho trường 43

Bảng 2. Đặc tả use-case quản lí ngân hàng câu hỏi 47

Bảng 3. Đặc tả use-case upload tài liệu 48

Bảng 4. Đặc tả use-case tạo buổi học 49

Bảng 5. Đặc tả use-case làm bài thi 50

Bảng 6. Đặc tả use-case vào phòng học 51

Bảng 7. Đặc tả use-case chia sẻ màn hình 52

Bảng 8. Đặc tả use-case chấm thi 53

Bảng 9. Cơ sở dữ liệu cho bảng users 64

Bảng 10. Cơ sở dữ liệu cho bảng Department 64

Bảng 11. Cơ sở dữ liệu cho bảng question 65

Bảng 12. Cơ sở dữ liệu cho bảng test 65

Bảng 13. Cơ sở dữ liệu cho bảng exam 65

Bảng 14. Cơ sở dữ liệu cho bảng Assignment 66

Bảng 15. Cơ sở dữ liệu cho bảng lớp học 66

Bảng 16. Cơ sở dữ liệu cho bảng section 67

Bảng 17. Cơ sở dữ liệu cho bảng topic 67

Bảng 18. Cơ sở dữ liệu cho bảng key 68

Bảng 19. Bảng tổng kết kết quả các test-case 79

**DANH SÁCH HÌNH VẼ**

Hình 1: WebRTC trên các thiết bị 19

Hình 2. Chương trình Php Heloword 24

Hình 3. Biểu đồ thể hiện tính phổ biến của Laravel năm 2013 25

Hình 4. Giới thiệu Web Service 1 28

Hình 5. Giới thiệu Web Service 2 29

Hình 6. Mô hình tổng quản hệ thống 31

Hình 7. Use-case tổng quan 35

Hình 8. Use-case quản lý trường 36

Hình 9. Use-case quản lí ngân hàng câu hỏi 37

Hình 10. Use-case quản lý đề thi 38

Hình 11 Use-case quản lý kết quả thi 39

Hình 12. Use-case quản lí ngân hàng bài giảng 40

Hình 13. Use-case quản lí lớp học 41

Hình 14. Use-case học tương tác 42

Hình 15. Mô hình kiến trúc hệ thống 54

Hình 16. Biểu đồ trình tự use-case tạo câu hỏi 56

Hình 17.Biểu đò trình tự use-case tìm kiếm câu hỏi 56

Hình 18. Biểu đồ trình tự use-case xóa câu hỏi 57

Hình 19. Biểu đồ trình tự cập nhật câu hỏi 57

Hình 20. Biểu đồ trình tự use-case xem chi tiết câu hỏi 58

Hình 21. Biểu đồ trình tự use-case upload tài liệu 58

Hình 22. Biểu đồ trình tự use-case làm bài thi 59

Hình 23. Biểu đồ trình tự use-case vào phòng học 60

Hình 24. Biểu đồ trình tự use-case chia sẻ màn hình 61

Hình 25. Biểu đồ trình tự use-case chấm thi 62

Hình 26. Cơ sở dữ liệu 63

Hình 27. Cấu trúc module của hệ thống 69

Hình 28. Biểu đồ lớp use-case quản lí ngân hàng câu hỏi 70

Hình 29. Biểu lớp quá trình thi của học viên 71

Hình 30. Biểu đồ lớp quá trình quản lí lớp và học 73

Hình 31. Giao diện tổng quan cho nghiệp vụ quản lý 74

Hình 32. Giao diện học viên vào khóa học 75

Hình 33. Giao diện phòng học 76

Hình 34. Giao diện trang chủ phía admin 80

Hình 35. Giao diện trang chủ phía quản lý 81

Hình 36. Giao diện quản lí ngân hàng câu hỏi 82

Hình 37. Giao diện tạo đề thi TOEIC 83

Hình 38. Giao diện tạo bài giảng 84

Hình 39. Giao diện quản lí kì thi 85

Hình 40. Giao diện trang chủ phía học viên 86

Hình 41. Giao diện danh sách bài thi phía học viên 87

Hình 42. Giao diện thi phía học viên 88

Hình 43. Giao diện vào khóa học phía học viên 89

Hình 44. Giao diện lớp học 90

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

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ và các viết tắt** | **Giải nghĩa** |
| 2. | Web | Web thường chỉ nằm trong một tên miền hoặc tên miền phụ trên World Wide Web của Internet. Một trang web là tập tin HTML hoặc XHTML có thể truy nhập dùng giao thức HTTP. |
| 3. | NAT | Network Address Translation cho phép một hay nhiều địa chỉ IP nội miền được ánh xạ với một hay nhiều địa chỉ IP ngoại miền |
| 4. | FLASH | Là công cụ để tạo ra các ứng dụng thiết kế các phần mềm mô phỏng, các hoạt cảnh. |
| 5. | IETF | Internet Engineering Task Force - là nhóm có chức năng nghiên cứu phát triển và quyết định các chuẩn dùng trong Internet thúc đẩy và phát triển các tiêu chuẩn Internet. |
| 6. | W3C | World Wide Web Consortium - là nhiệm vụ hướng dẫn World Wide Web với đầy đủ các tiềm năng phát triển của giao thức và các hướng dẫn mà bảo đảm lâu dài sự tăng trưởng của các trang web |
| 7. | VoIP | Voice over Internet Protocol - phương thức truyền thông sử dụng giao thức TCP/IP dựa trên cơ sở hạ tầng sẵn có của mạng Internet |
| 8. | ICE | Interactive Communication Establishment - là một giao thức được cùng để thiết lập phiên media dựa trên UDP đi qua NAT một cách nhanh nhất. |
| 9. | STUN | Session Traversal Utilities for NAT - là một giao thức mạng cho phép các máy khách tìm ra địa chỉ công khai của máy |
| 10. | TURN | Traversal Using NAT Relay - là một giao thức mạng cho phép các máy khách tìm ra địa chỉ công khai của máy có hỗ trọ giao thức TCP |

**TÓM TẮT NỘI DUNG ĐỒ ÁN Cơ Sở**

Trong xu thế nền công nghiệp 4.0, việc liên kết giáo dục - đào tạo được xem là xu hướng, biện pháp quan trọng hàng đầu để phát triển ở nhiều quốc gia trên thế giới trong đó có Việt Nam. Để đáp ứng nhu cầu kết nối giáo dục giữa các quốc gia, vùng miền… đã có rất nhiều phần mềm, hệ thống được sinh ra đời để giải quyết các vấn đề trên. Tuy nhiên, đa số các chương trình/hệ thống kia chưa đáp ứng được nhu cầu ngày càng mở rộng.

Hiện nay, công nghệ thông tin đã và đang phát triển vượt bậc và đã gặt hái được nhiều thành công trong một số lĩnh vực nên việc học truyền thống dần được thay thế bởi hình thức học tập khác linh hoạt hơn tiện lợi hơn về thời gian và địa điểm đó là mô hình đào tạo trực tuyến.

Việc quản lý đồ án tốt nghiệp sinh viên là một yêu cầu thiết yếu của mỗi trường đại học. Để quản lý đồ án và lưu trữ có hiệu quả thì không đơn giản bởi đòi hỏi kỹ năng của người quản lý. Làm sao để vừa có thể kiểm soát được số lượng đề tài của các khóa, các hệ đào tạo khác nhau vừa có thể lưu trữ nó làm tài liệu cho các bạn sinh viên khóa sau đồng thời phân công giảng viên hướng dẫn đồ án một cách hợp lý. Việc quản lý và lưu trữ đồ án trước đây chủ yếu sử dụng phương pháp thủ công, bản mềm, sourcecode đồ án thường lưu trên đĩa … chiếm diện tích lưu trữ rất lớn. Do đó, khi quản lý hay tìm kiếm mất rất nhiều thời gian và công sức mà hiệu quả công việc đem lại không cao đôi khi còn xảy ra sai sót mất mát dữ liệu không đáng có. Hiện nay công tác quản lý đồ án tốt nghiệp sinh viên của một số trường đại học nói chung và khoa CNTT nói riêng là chưa được hoàn chỉnh. Do đó việc đòi hỏi có một trang web chuyên dụng trợ giúp cho công việc quản lý và lưu trữ đồ án tốt nghiệp sinh viên là một nhu cầu tất yếu để đảm bảo hiệu quả và tiết kiệm thời gian trong công việc.

Xuất phát từ nhu cầu trên, em muốn xây dựng một hệ thống kết nối hỗ trợ kết nối giữa những sinh viên và giảng viên, nhầm nâng cao hiệu quả của việc thực hiện đồ án. Giúp nâng cao sự tương tác giữa giảng viên và sinh viên thông qua việc phân chia các công việc.

Đồ án gồm 5 phần:

Chương 1 Tổng quan về mô hình quản lý đồ án sinh viên online

Chương 2 Công nghệ sử dụng: trình bày giới thiệu tổng quan về công nghệ sử dụng cho hệ thống.

Chương 3 Phân tích: trình bày các yêu cầu đặt ra, phân tích đánh giá, đặc tả nội dung yêu cầu đưa ra.

Chương 4 Thiết kế: thiết kế chi tiết hệ thống, thiết kế cơ sở dữ liệu, thiết kế lớp, giao diện cho hệ thống

Chương 5 Triển khai hệ thống và kiểm thử

Chương 6 Tổng kết và hướng phát triển: Trình bày kết luận chung đề tài, đánh giá ưu nhược điểm của hệ thống và định hướng trong tương lai.

**CHƯƠNG 1: TỔNG QUAN VỀ MÔ HÌNH quản lý đồ án Sinh Viên**

|  |
| --- |
| ***Nội dung chương này sẽ trình bày các vấn đề chính sau đây: Thứ nhất, trình bày tổng quan về mô hình website quản lý đồ án. Thứ hai, em sẽ trình bày bài toán đặt ra, sau đó là mục tiêu đề tài, khó khăn thách thức và cuối cùng là giải pháp đề xuất.*** |

* **Tổng quan về mô hình Website quản lý đồ án.**

**Website quản lý đồ án** là một hệ thống được xây dựng nhầm quản lý đồ án của sinh viên khoa CNTT. Hệ thống website quản lý đồ án sinh viên được sử dụng để đăng ký các loại đồ án như: đồ án cơ sở, đồ án chuyên ngành, đồ án tổng hợp,…và được phân chia hướng dẫn của giảng viên hướng dẫn đồ án. Mục đích của website quản lý đồ án có chức năng hiển thị đăng ký đồ án, đăng ký đề tài, đăng ký và tạo nhóm,…Ngoài ra còn website còn có thể hiển thị thông tin sinh viên, thông tin giảng viên, thời gian thực hiện đồ án,…

**Ưu điểm của Website quản lý đồ án sinh viên**

Những ưu điểm của website là:

* Tính linh hoạt: Website có thể đăng ký, báo cáo, tạo nhóm,...theo mô hình trực tuyến mà không cần phải giống như trước kia phải lên tận khoa để xin giấy đăng ký.
* Dễ tiếp cận và dễ truy cập: đối với website, sinh viên có thể sữ dụng các trình duyệt web để trực tiếp sử dụng website mà không cần tốn quán nhiều thời gian, với tên miền dễ nhớ và dễ truy cập thì việc quản lý đồ án thật sự sẽ dễ dàng hơn.
* Tính cập nhật: tình hình đồ án sẽ được cập nhật rất nhanh bởi sinh viên và giảng viên, thư ký khoa, quản lý khoa,…
* Học có sự hợp tác, phối hợp (Collaborative learning): Website sẽ giúp giảng viên hướng dẫn và sinh viên thực hiện đồ án dễ dàng hơn, tiết kiệm chi phí và thời gian đi lại cho những buổi gặp mặt trực tiếp để trao đổi về yêu cầu và xem xét đồ án.

**Nhược điểm của Website Quản Lý Đồ Án Sinh Viên**

Bên cạnh những ưu điểm nổi trội của website đã kể trên, hình thức quản lý còn một số khó khăn sau:

* Về phía sinh viên

Phải thường xuyên truy cập website để cập nhật những thông báo từ giáo viên về những vấn đề của đồ án( báo cáo tuần, dealine, chỉnh sửa báo cáo,…). Sinh viên phải tự phân tích yêu cầu của giảng viên một cách rõ ràng và chính xác.

* Về phía giảng viên hướng dẫn

Trong nhiều trường hợp, không thể và không nên đưa ra các nội dung quá trừu tượng, quá phức tạp. Sinh viên sẽ khó hình dung ra yêu cầu để tiếp xúc với nhiệm vụ mà giảng viên đã đưa ra. Nên đòi hỏi giáo viên phải đưa ra những yêu cầu một cách trình tự và rành mạch giúp sinh viên dễ tiếp thu những yêu cầu đó.

**Bài toán đặt ra**

* **Nhu cầu quản lý đồ án sinh viên**

Hiện nay, nhu cầu về việc quản lý đồ án online của sinh viên khá là mới, chưa có nhiều trường đại học áp dụng vào để quản lý đồ án sinh viên qua website một cách chuẩn mực. Vì thế nhu cầu này vẫn chưa được đáp ứng đúng như những gì mà các trường đại học đặt ra.

* **Thực trạng quản lý đồ án thông qua website**

Hiện nay, đã có các trường đại học dần đưa hệ thống quản lý đồ án thông qua ứng dụng, website,…vào quá trình thực hiện đồ án của sinh viên khoa công nghệ. Nhưng vẫn chưa có trường nào hoàn thành tốt những yêu cầu đặt ra để quản lý tốt quá trình thực hiện đồ án của sinh viên. Có những trường chỉ đưa vào thực nghiệm rồi cũng đứng ở một chỗ mà không thể triển khai phát triển được. Nên thực trạng chung hiện nay về việc quản lý đồ án sinh viên online vẫn còn đang khá gay gắt, nếu sở hữu được hệ thống quản lý đồ án sinh viên thông qua các công cụ web, app thì các trường đại học sẽ giảm nhẹ lại quá trình làm việc từ thư ký khoa và quản lý khoa từ việc quản lý đồ án sinh viên. Và giúp giảng viên và sinh viên giảm thiểu đi thời gian gặp mặt để trao đổi yêu cầu về quá trình thực hiện đồ án.

* **Mục tiêu đề bài**

Từ những yêu cầu trên chúng em đã hình dung ra giải pháp xây dựng hệ thống website quản lý đồ án sinh viên. Hệ thống cho phép sinh viên đăng ký các loại đồ án ( đồ án cơ sở, đồ án chuyên ngành, đồ án tổng hợp,…). Không chỉ có vậy, các sinh viên sẽ được đăng ký một nhóm đồ án gồm 2 đến 3 thành viên cho một nhóm. Nhầm giảm nhẹ quá trình làm đồ án và nâng cao chất lượng của đồ án lên. Trong lúc đăng ký đồ án, sinh viên sẽ được phân chia giảng viên hướng dẫn để thực hiện đồ án.

* **Khó khăn và thách thức**

Quá trình xây dựng hệ thống website quản lý đồ án sinh viên yêu cầu cần phải có hiểu biết rõ về vấn đề quán lý đồ án tại trường đại học. Để phát triển tốt hệ thống cần có những nguyên cứu rõ ràng và chặt chẽ các quá trình quản lý đồ án sinh viên.

* **Giải pháp đề xuất**

Có thể tham khảo thông qua giáo viên hướng dẫn, sinh viên có thể tìm hiểu rõ hơn về vấn đề quản lý đồ án sinh viên. Tham khảo thêm một số tài liệu có sẳn trên mạng, có thể hiểu rõ hơn về quá trình xây dựng và hỗ trợ thêm cho việc xây dựng website.

**CHƯƠNG 2: CÔNG NGHỆ SỬ DỤNG**

|  |
| --- |
| ***Nội dung chương 2 em sẽ trình bày hai vấn đề chính sau đây: Đầu tiên, em sẽ trình bày lựa chọn công nghệ cho hệ thống, tiếp theo là giới thiệu tổng quan về các công nghệ chính mà em sử dụng cho đồ án.*** |

* **Lựa chọn công nghệ**

**Ngôn ngữ lập trình**

HTML (HyperText Markup Language) – Ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo các tài liệu có thể truy cập trên mạng. Tài liệu HTML được tạo nhờ dùng các thẻ và các phần tử của HTML. File được lưu trên máy chủ dịch vụ web với phần mở rộng “.htm” hoặc “.html”. Các trình duyệt sẽ đọc tập tin HTML và hiển thị chúng dưới dạng trang web. Các thẻ HTML sẽ được ẩn đi, chỉ hiển thị nội dung văn bản và các đối tượng khác: hình ảnh, media. Với các trình duyệt khác nhau đều hiển thị một tập HTML với một kết quả nhất định. Các trang HTML được gửi đi qua mạng internet theo giao thức HTTP. HTML không những cho phép nhúng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép nhúng các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web. Để trình bày trang web hiệu quả hơn thì HTML cho phép sử dụng kết hợp với CSS. HTML không những cho phép nhúng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép nhúng các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web. HTML là một chuẩn ngôn ngữ internet được tạo ra và phát triển bởi tổ chức World Wide Web Consortium còn được viết tắc là W3C. Trước đó thì HTML xuất bản theo chuẩn của RFC. HTML được tương thích với mọi hệ điều hành cùng các trình duyệt của nó. Khả năng dễ học, dễ viết là một ưu điểm của HTML không những vậy việc soạn thảo đòi hỏi hết sức đơn giản, chúng ta có thể dùng word, notepad hay bất cứ một trình soạn thảo văn bản nào để viết và chỉ cần lưu với định dạng “.html “ hoặc “.htm” là đã có thể tạo ra một file chứa HTML. Hiện nay, phiên bản mới nhất của HTML là HTML5 với nhiều tính năng ưu việt so với các phiên bản cũ HTML cải tiến khá nhiều đặc biệt hỗ trợ mạnh mẽ các phần tử multimedia mà không cần các plugin. Một tập tin HTML bao gồm trong đó là các đoạn văn bản HTML, được tạo lên bởi các thẻ HTML. HTML5 nói chung mạnh mẽ hơn nhiều không chỉ về tốc độ và độ thích ứng cao mà chính là khả năng hỗ trợ API (Application Programming Interface - giao diện lập trình ứng dụng) và DOM (Document Object Model – các đối tượng thao tác văn bản).

* **Giới thiệu tổng quan một số công nghệ sử dụng**
* **Công nghệ Web**

**Định nghĩa**

**Website** còn gọi là **trang web**, trang mạng, là một tập hợp trang web, thường chỉ nằm trong một tên miền hoặc tên miền phụ trên World Wide Web của Internet. Một trang web là tập tin HTML hoặc XHTML có thể truy nhập dùng giao thức HTTP. Website có thể được xây dựng từ các tệp tin HTML (website tĩnh) hoặc vận hành bằng các CMS chạy trên máy chủ (website động). Website có thể được xây dựng bằng nhiều ngôn ngữ lập trình khác nhau (PHP,.NET, Java, Ruby on Rails...)

**Lịch sử phát triển**

Ngày 6/8/1991 là một ngày đáng nhớ bởi đó là thời điểm mã máy tính "non nớt" cho www được đăng trên alt.hypertext để mọi người có thể tải và tìm hiểu nó. Cũng bắt đầu từ hôm ấy công nghệ web được thế giới biết đến.

Jeff Groff, người cùng tham gia viết mã với Berners-Lee, cho biết ý tưởng tạo dựng web thực ra được hình thành rất đơn giản: "Chúng tôi luôn nghĩ rằng người sử dụng không cần phải xoay sở với những vấn đề kỹ thuật phức tạp". Web giống như một tấm khăn trải giường với nhiệm vụ cố che phủ sự rắc rối của những dữ liệu được lưu hành trên Internet.

Paul Kunz, nhà khoa học đã thiết lập máy chủ web đầu tiên ở châu Âu vào tháng 12/1991, cho biết đầu thập niên 90, máy tính giống như những "ốc đảo" thông tin. Một lần đăng nhập chỉ có thể truy cập tài nguyên của một hệ thống. Chuyển sang máy tính khác đồng nghĩa với việc họ phải đăng nhập thêm lần nữa và phải sử dụng những bộ lệnh khác nhau để truy xuất dữ liệu.

Web đã lôi kéo sự chú ý của Kunz khi ông chứng kiến Berners-Lee trình diễn khả năng hoạt động của web trên hệ thống IBM. Sau đó, Kunz đã thiết lập máy chủ web, cho phép các chuyên gia vật lý rà soát hơn 200.000 dữ liệu dễ dàng hơn bao giờ hết.

Tuy nhiên, dù các nhà vật lý đã bị web quyến rũ, đa số mọi người lại không nhận biết được khả năng tiềm ẩm của nó. Kunz cho rằng điều này là do nhiều tổ chức cũng đang thực hiện ý tưởng tương tự. Công nghệ nổi tiếng nhất khi đó là Gopher của Đại học Minnesota (Mỹ), cũng với tham vọng hóa giải sự phức tạp của những máy tính kết nối Internet. Gopher được ra mắt vào mùa xuân năm 1991 và lưu thông Gopher cao hơn hẳn so với lưu thông web trong vài năm tiếp theo.

Dự án www chỉ thực sự thăng hoa khi chuyên gia Marc Andreessen thuộc Đại học Illinois (Mỹ) giới thiệu trình duyệt web máy tính đầu tiên vào tháng 4/1993. Trình duyệt Mosaic đã quá thành công và một số tính năng vẫn được coi là quy ước trong công nghệ web ngày nay. Cũng vào năm 1993, Đại học Minnesota thu phí Gopher khiến người ta bắt đầu phải tìm đến các giải pháp thay thế.

Ngoài ra, theo Ed Vielmetti, nhà nghiên cứu thuộc Đại học Michigan, ngay từ những năm đầu, web đã chứng minh được tính hữu ích với người sử dụng thông thường. Mọi người có thể sử dụng các trang web để tự bộc lộ mình, điều mà những công nghệ khác không cho phép (hình thức mới hiện nay của nó chính là [blog](http://www.vnexpress.net/Topic/?ID=3434)).

Cuối năm 1994, lưu thông web rốt cuộc cũng vượt qua Gopher và từ đó chưa bao giờ bị tụt lại. Hiện nay, gần 100 triệu website đã xuất hiện và người ta gần như đồng nhất công nghệ web với Net.

Kunz cho biết ý tưởng hình thành www là để tạo điều kiện cho mọi người vừa đọc vừa đóng góp nội dung. Những công cụ mới như site chia sẻ ảnh, [mạng xã hội](http://www.vnexpress.net/Topic/?ID=3805), blog, các trang wiki... đang dần hoàn thành lời hứa ban đầu của nhóm chuyên gia phát triển web.

Và như thế, theo Kunz, web bây giờ mới chỉ bắt đầu những bước đi đầu tiên.

**Tính phổ biến và tiềm năng phát triển**

Ban đầu, các trang Web là tĩnh, người dùng gửi yêu cầu một tài nguyên nào đó, và server sẽ trả về tài nguyên đó. Các trang Web không có gì hơn là một văn bản được định dạng và phân tán. Đối với các trình duyệt, thì các trang Web tĩnh không phải là các vấn đề khó khăn, và trang Web lúc đầu chỉ để thông tin về các sự kiện, địa chỉ, hay lịch làm việc qua Internet mà thôi, chưa có sự tương tác qua các trang Web. Năm 1990, Tim Berners-Lee, tại CERN, đã sáng chế ra HTML (Hyper Text Markup Language), ngôn ngữ đánh dấu siêu văn bản. HTML rất đơn giản và dễ dùng, và nó trở thành một ngôn ngữ rất phổ biến và cơ bản.

Tuy nhiên, không lâu sau đó, nhu cầu về các trang Web động, có sự tương tác ngày một tăng, chính vì thế sự ra đời các công nghệ Web động là một điều tất yếu. Sau đây là một số công nghệ Web động cơ bản: JavaScript, ASP.net, XML, HTML,..



*Hình 1: Mô tả website*

**Ưu điểm, lợi ích**

* Tăng khả năng tiếp cận với người dùng
* Tăng phạm vi tiếp cận
* Tính tương tác cao
* Nền tảng vững
* Xác định được tìm năng của website
* Tăng hiệu quả cạnh tranh
* Dễ dàng quản lý
* Dễ dàng lấy ý kiến từ người dùng
* Dễ dàng phát triển thêm tính năng
* **Ngôn ngữ lập trình CSS**
* **Ngôn ngữ CSS**

CSS (Cascading Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, SVG, hay UML,…CSS quy định cách hiển thị của các thẻ HTML bằng cách quy định các thuộc tính của các thẻ đó(font chữ, kích thước, màu sắc...).

CSS có cấu trúc đơn giản và sử dụng các từ tiếng anh để đặt tên cho các thuộc tính. CSS khi sử dụng có thể viết trực tiếp xen lẫn vào mã HTML hoặc tham chiếu từ một file css riêng biệt. Hiện nay CSS thường được viết riêng thành một tập tin với mở rộng là “.css”. Chính vì vậy mà các trang web có sử dụng CSS thì mã HTML sẽ trở nên ngắn gọn hơn. Ngoài ra có thể sử dụng một tập tin CSS đó cho nhiều website tiết kiệm rất nhiều thời gian và công sức. Một đặc điểm quan trọng đó là tính kế thừa của CSS do đó sẽ giảm được số lượng dòng code mà vẫn đạt được yêu cầu.

Tuy nhiên, đối với CSS thì các trình duyệt hiểu theo kiểu riêng của nó. Do vậy, việc trình bày một nội dung trên các trình duyệt khác nhau là không thống nhất. CSS cung cấp hàng trăm thuộc tính trình bày dành cho các đối tượng với sự sáng tạo cao trong kết hợp các thuộc tính giúp mang lại hiệu quả.

* **NGÔN NGỮ LẬP TRÌNH C# .**

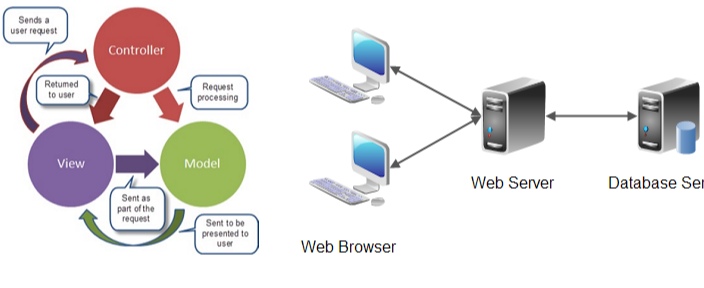
C# thường được đọc là C thăng hoặc “See Sharp” hay “C-sharp”. Nó là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Tập đoàn Microsoft. Ngôn ngữ này được xem là ngôn ngữ lập trfinh hướng đối tượng trong sáng và thuần nhất. Nó hiện thực hầu hết các tính chất tốt của mô hình hướng đối tượng giống như ngôn ngữ lập trình Java. C# là ngôn ngữ được Microsoft phát triển dựa trên 2 ngôn ngữ huyền thoại đó là C++ và Java. Và nó cũng được miêu tả là loại ngôn  ngữ có được sự cân bằng giữa C++, Visual Basic , Delphi và Java .

C# làm việc chủ yếu trên bộ khung .NET (.NET framework). Ngôn ngữ lập trình này có khả năng tạo ra nhiều ứng dụng mạnh mẽ và an toàn cho nền tảng Windows.

* **Framework MVC ASP.net**

**Giới thiệu**

ASP.NET là một nền tảng ứng dụng web (*web application framework*) được phát triển và cung cấp bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft), cho phép những người lập trình tạo ra những trang web động, những [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web) và những [dịch vụ web](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_v%E1%BB%A5_web). Lần đầu tiên được đưa ra thị trường vào tháng 1 năm 2002 cùng với phiên bản 1.0 của.NET framework, là công nghệ nối tiếp của Microsoft's Active Server Pages(ASP). ASP.NET được biên dịch dưới dạng [Common Language Runtime](https://vi.wikipedia.org/w/index.php?title=Common_Language_Runtime&action=edit&redlink=1) (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi.NET language.



*Hình 3. Mô hình MVC ASP.net*

Laravel hỗ trợ một loạt các cơ sở dữ liệu, cung cấp một hệ sinh thái rộng lớn. Những lập trình viên mới có thể tìm hiểu về Laravel một cách nhanh chóng. Laravel tái sử dụng và sắp xếp các thành phần hiện có để cung cấp cho bạn một framework để xây dựng các ứng dụng web một cách có cấu trúc và rất thực tế. Nó kết hợp được nhiều ý tưởng tốt từ các ngôn ngữ lập trình cũng như các framework có tiếng như CodeIgniter, Yii, ASP.NET MVC, Ruby on Rails…

**Lịch sử phát triển**

*Asp.Net 1.0 :* Phiên bản đầu tiên được phát hành vào tháng 3 năm 2009 trên nền tảng bộ .Net Framework 3.5 bộ Visual Studio 2008.  
*Asp.Net 2.0 :* Phiên bản này được phát hành vào tháng 3 năm 2010 trên bộ .Net Framework 3.5 và .Net Framework 4.0 bộ Visual Studio 2010.   
*Asp.Net MVC 3 :* Phiên bản này được phát hành vào tháng 1 năm 2011 trên nền tảng bộ .Net Framework 4.0, các đặc điểm chính của phiển bản này :

* So với phiên bản ASP.NET MVC2 thì tính năng mới được hỗ trợ thêm HTML5 và CSS3; tính năng này chỉ hỗ trợ cho những trình duyệt mới;
* Cải thiện về Model validation
* Hỗ trợ 2 View Engine: RAZOR, ASPX, kể cả Open source trong khi ASP.NET MVC2 chỉ có ASPX;
* Controller được cải tiến hơn như thuộc tính ViewBag và kiểu ActionResult
* Cải thiện Dependency Injection với IDpendencyResolver (có 2 phần: DependencyResolver và interface IDpendencyResolver)trong ASP.NET MVC3;đây là lớp thực thi mô hình Service Locator, cho phép framework gọi DIContainer khi cần làm việc với 1 lớp thực thi từ 1 kiểu cụ thể
* Cách tiếp cận với JavaScript được hạn chế
* Hỗ trợ caching trong Partial page

*Asp.Net 4.0 :* Phiên bản này được phát hành vào tháng 8 năm 2012 trên nền tảng bộ .Net Framework 4.0 và .Net Framework 4.5 trên phiên bản Visual Studio 2010 SP1 and Visual Studio 2012. Phiên bản này có các đặc điểm chính như sau:

* ASP.NET Web API ra đời, nhằm đơn giản hoá việc lập trình với HTML hiện đại và đây là một cải tiến mới thay cho WCF Web API
* Mặc định của dự án được cải thiện hơn về hình thức bố trí, giúp dễ nhìn hơn;
* Mẫu Empty Project là project trống, phù hợp cho những tín đồ developer muốn nâng cao khả năng lập trình với ASP.NET MVC4
* Giới thiệu jQuery Mobile, và mẫu Mobile Project cho dự án
* Hỗ trợ Asynchrnous Controller
* Kiểm soát Bundling và Minification thông qua web.config
* Hỗ trợ cho việc đăng nhập OAuth và OpenID bằng cách sử dụng thư viện DotNetOpenAuth
* Phiên bản mới Windows Azure SDK 1.6 được phát hành

*Asp.Net MVC 5 :* Phiên bản này được phát hành vào tháng 10 năm 2013 trên nền tảng bộ .Net Framework 4.5 và .Net Framework 4.5.1 trên phiên bản Visual Studio 2013. Các đặc điểm chính của phiên bản này như sau:

* Với MVC5 thì cải tiến hơn so với ASP.NET MVC4, Bootstrap được thay thế mẫu MVC mặc định
* Chứng thực người dùng Authentication Filter được tuỳ chỉnh hoặc chứng thực từ hãng thứ 3 cung cấp
* Với Filter overrides, chúng ta có thể Filter override trên Method hoặc Controller;
* Thuộc tính Routing được tích hợp vào MVC5

**Ưu điểm nổi bật**

ASP.NET 5 cho phép bạn linh hoạt tốt hơn khi host ứng dụng của bạn với hai môi trường sau: full .Net CLR (Common Language Runtime) và .Net Core CLR. full .Net CLR là hệ thống thực thi (runtime engine) mặc định mà có thể cung cấp khả năng tương thích với các ứng dụng lỗi thời. Nó chỉ chạy trên hệ điều hành Windows.

.NET Core thì hoàn toàn mới và được chia nhỏ thành các mô-đun, chạy đa nền tảng trong một thiết kế nhỏ gọn. Core CLR là mã nguồn mở và đa nền tảng, với các port dành cho Windows, Linux, Mac OS X, và FreeBSD. Nói tóm lại, hiện tại bạn sẽ có thể xây dựng ứng dụng trên một nền tảng và vận hành nó trên một hay thậm chí nhiều nền tảng khác.

Một trong những tính năng hấp dẫn nhất của ASP.NET5 là việc hợp nhất mô hình lập trình của ASP.Net MVC, ASP.Net Web API, và ASP.Net Web.Trong các phiên bản trước của ASP.NET, những framework này được thực hiện một cách riêng biệt và có phần trùng lặp cũng như mâu thuẫn.

Mô hình lập trình hợp nhất có nghĩa là bạn có thể tạo ra một ứng dụng web duy nhất xử lý các giao diện Web và dịch vụ dữ liệu mà không phải dung hòa sự khác biệt giữa 3 framework. Bây giờ bạn chỉ có một loại bộ điều khiển để xử lý các yêu cầu mà chủ yếu là các ứng dụng của MVC, Web API và Web Pages. Về bản chất, bây giờ bạn có một routing framework , một binding framework, và một filter pipeline.

Thể hiện tính chuyên nghiệp trong lập trình, phân tích thiết kế. Do được chia thành các thành phần độc lập nên giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì…

**Nhược điểm**

Đối với dự án nhỏ việc áp dụng mô hình MC gây cồng kềnh, tốn thời gian trong quá trình phát triển. Tốn thời gian trung chuyển dữ liệu của các thành phần.

* **Hệ quản trị cơ sở dữ liệu MySQL**

**MySQL** là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X), [Unix](https://vi.wikipedia.org/wiki/Unix), [FreeBSD](https://vi.wikipedia.org/wiki/FreeBSD), [NetBSD](https://vi.wikipedia.org/w/index.php?title=NetBSD&action=edit&redlink=1), [Novell NetWare](https://vi.wikipedia.org/w/index.php?title=Novell_NetWare&action=edit&redlink=1), [SGI Irix](https://vi.wikipedia.org/w/index.php?title=SGI_Irix&action=edit&redlink=1), [Solaris](https://vi.wikipedia.org/wiki/Solaris), [SunOS](https://vi.wikipedia.org/w/index.php?title=SunOS&action=edit&redlink=1) v.v.

**MySQL** là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL).

**MySQL** được sử dụng cho việc bổ trợ [PHP](https://vi.wikipedia.org/wiki/PHP), [Perl](https://vi.wikipedia.org/wiki/Perl), và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,..

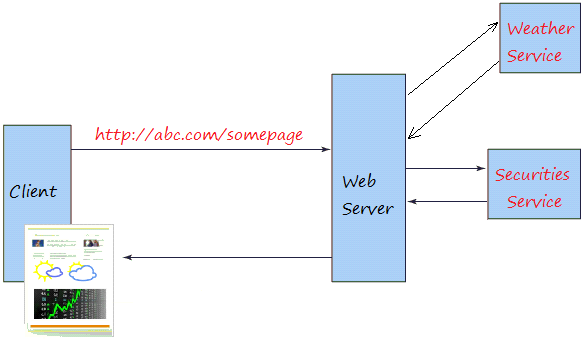
* **Giao thức kết nối HTTP**

Trong giao thức HTTP, client truyền thông điệp cho server thông qua hai phương thức là POST và GET. Với phương thức POST, lập trình viên có thể giấu các tham số truyền lên server, điều này sẽ giúp bảo mật hơn cho server, tuy nhiên sẽ gây khó khăn hơn cho lập trình viên phía client. Với phương thức GET, lập trình viên sẽ gửi tất cả các tham số yêu cầu qua url, điều này sẽ dễ dàng hơn cho lập trình viên phía client tuy nhiên sẽ rất khó để giấu url cho server.

Việc sử dụng kết nối HTTP có ưu điểm là dễ dàng cho lập trình viên nhưng nó lại có một số điểm bất lợi sau:

* HTTP khiến cho kết nối giữa Client-Server không được liên tục. Hệ thống chỉ thực hiện kết nối khi có yêu cầu, và đóng lại sau khi nhận được dữ liệu. Do đó, việc request sẽ tốn một khoảng thời gian để thiết lập kết nối, điều mà kết nối socket không mắc phải.
* Ngoài ra do việc sử dụng giao thức kết nối HTTP thực hiện truy vấn trực tiếp lên service không đảm bảo được tính bảo mật và trong suốt của hệ thống do có thể dễ dàng phát hiện các câu truy vấn.
* **Kiến trúc Web Service và Restful Service**

**Web Service** là một dịch vụ web, nó là một khái niệm rộng hơn so với khái niệm web thông thường, nó cung cấp các thông tin thô, và khó hiểu với đa số người dùng, chính vì vậy nó được sử dụng bởi các ứng dụng. Các ứng dụng này sẽ chế biến các dữ liệu thô trước khi trả về cho người dùng cuối cùng.

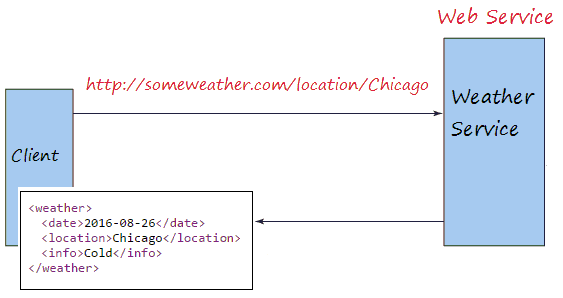


*Hình 4. Giới thiệu Web Service 1*

Ví dụ bạn vào một trang web bất kì để xem thông tin về về lĩnh vực nào đó thì trang web đó sẽ hiển thị cho bạn các thông tin bạn muốn.

Để có được các dữ liệu hiển thị người dùng hệ thống cần phải lấy thông tin từ một nguồn nào đó, nó có thể là một dịch vụ web chuyên cung cấp các số liệu thời tiết ứng với các vùng miền khác nhau.

Các dữ liệu sẽ được chế biến trước khi trả về cho bạn là một trang web hoàn chỉnh. **Web Service** thường cung cấp các dữ liệu thô mà nó khó hiểu đối với đa số người dùng thông thường, chúng thường được trả về dưới dạng XML hoặc JSON.



*Hình 5. Giới thiệu Web Service 2*

REST định nghĩa các quy tắc kiến trúc để bạn thiết kế Web services, chú trọng vào tài nguyên hệ thống, bao gồm các trạng thái tài nguyên được định dạng như thế nào và được truyền tải qua HTTP, và được viết bởi nhiều ngôn ngữ khác nhau. Nếu tính theo số dịch vụ mạng sử dụng, REST đã nổi lên trong vài năm qua như là một mô hình thiết kế dịch vụ chiếm ưu thế. Trong thực tế, REST đã có những ảnh hưởng lớn và gần như thay thế SOAP và WSDL vì nó đơn giản và dễ sử dụng hơn rất nhiều.

REST là một bộ quy tắc để tạo ra một ứng dụng Web Service, mà nó tuân thủ 4 nguyên tắc thiết kế cơ bản sau:

* Sử dụng các phương thức HTTP một cách rõ ràng
* Phi trạng thái
* Hiển thị cấu trúc thư mục như các URL
* Truyền tải JavaScript Object Notation (JSON), XML hoặc cả hai.

**REST** đặt ra một quy tắc đòi hỏi lập trình viên xác định rõ ý định của mình thông qua các phương thức của **HTTP**. Thông thường ý định đó bao gồm lấy dữ liệu, trèn dữ liệu, cập nhập dữ liệu hoặc xóa dữ liệu. Vậy khi bạn muốn thực hiện một trong các ý định trên hãy lưu ý các quy tắc sau:

Để tạo một tài nguyên trên máy chủ, bạn cần sử dụng phương thức **POST**.

Để truy xuất một tài nguyên, sử dụng **GET**.

Để thay đổi trạng thái một tài nguyên hoặc để cập nhật nó, sử dụng **PUT**.

Để huỷ bỏ hoặc xoá một tài nguyên, sử dụng **DELETE**.

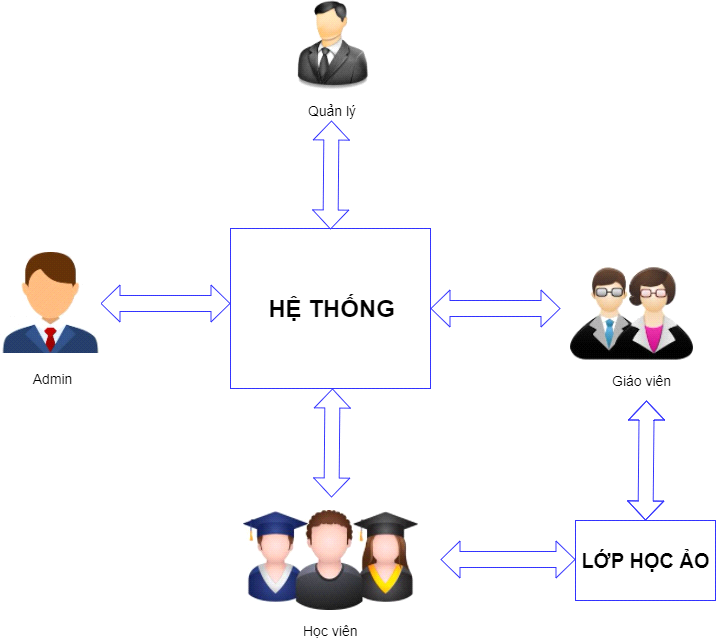
Chú ý rằng các nguyên tắc ở trên là không bắt buộc, thực tế có thể sử dụng phương thức **GET** để yêu cầu lấy dữ liệu, trèn, sửa hoặc xóa dữ liệu trên Server. Tuy nhiên **REST** đưa ra các nguyên tắc ở trên mục đích đưa mọi thứ trở lên rõ ràng và dễ hiểu.

**CHƯƠNG 3: PHÂN TÍCH**

|  |
| --- |
| ***Nội dung chương này em sẽ trình bày về các vấn đề sau đây. Đầu tiên, em xin trình bày về mô hình tổng quát hệ thống, tiếp theo là nội dung yêu cầu đặt ra cho hệ thống và cuối cùng là phân tích đặt tả các yêu cầu.*** |

* **Mô hình tổng quát hệ thống**

Hệ thống được xây dựng với mục đích chính là hỗ trợ sinh viên và giáo viên thuận tiện hơn trong việc thực hiện đồ án. Vì vậy đối tượng hướng đến của hệ thống là sinh viên, giáo viên, thư ký khoa, phòng đào tạo, khoa cntt . Mô hình tổng quát của hệ thống được thể hiện trong hình Hình 6.



*Hình 6. Mô hình tổng quản hệ thống*

Nhóm sử dụng hệ thống bao gồm 4 nhóm người sử dụng

* Admin: là người dùng cấp cao nhất trong hệ thống. Admin có vai trò quản lý tất cả các hoạt động của người dùng, tạo và thêm, xóa các tài khoản người dùng trên hệ thống. Admin có thể theo dõi, quản lý các nhóm đồ án đang thực hiện trên hệ thống.
* Thư ký khoa: là người dùng có vai trò quản lý toàn bộ các hoạt động trong phạm vi hệ thống của khoa cntt . Thư ký khoa có các chức năng chính: thông báo thời gian xây dựng đồ án, quản lý đồ án của sinh viên, phân công giáo viên hướng dẫn sinh viên tham gia đồ án, lưu trữ và thống kê số lượng đồ án đã hoàn thành của sinh viên đã hoàn thành.
* Giảng viên: là người dùng có chức năng chính là tham gia hướng dẫn sinh viên xây dựng đồ án. Khi xây dựng đồ án sẽ có những buổi trao đổi trực tiếp với giảng viên hàng tuần và để hạn chế về thời gian củng như đi lại thì giữa giảng viên và sinh viên sẽ trao đổi trực tiếp trên hệ thống. Giảng viên còn có chức năng quản lý, xem, chấm điểm và thống kê lại số lượng sinh viên và đồ án mà mình hướng dẫn.
* Sinh viên: là nhóm người dùng quan trọng nhất của hệ thống vì các nhu cầu và lợi ích của học viên ảnh hưởng trực tiếp tới việc xây dựng các chức năng cho hệ thống. Học viên có chức năng đăng kí tham gia xây dựng đồ án, có thể tham gia một mình hoặc thành lập nhóm tham gia tối đa 3 thành viên, sinh viên tham gia phải báo cáo nhiệm vụ hoàn thành hàng tuần trên hệ thống và đợi giảng viên phân công nhiệm vụ từng tuần. Khi hoàn thành đồ án phải nộp đồ án trên hệ thống website.
* **Nội dung yêu cầu**
* **Yêu cầu chức năng**

Admin: Admin phải có giao diện đăng nhập vào hệ thống. Sau khi đăng nhập vào hệ thống admin có thể tạo trường học (cơ sở đào tạo) trên hệ thống, thêm quản lý cho trường vừa tạo. xem thông tin và quản lý các trường có trên hệ thống.

Manager (Quản lý): Quản lý có giao diện đăng nhập vào hệ thống với vai trò là quản lý một hệ thống. Quản lý có thể xây dựng ngân hàng dữ liệu bao gồm ngân hàng câu hỏi, ngân hàng đề thi, ngân hàng bài giảng. Người quản lý có thể tạo mới, xem, tìm kiếm các loại ngân hàng dữ liệu. Về dữ liệu thi quản lý có thể xuất các đề thi dưới dạng các file tài liệu có thể in ra bản cứng dùng tài liệu cho việc giảng dạy. Về kì thi quản lý phải quản lý các kì thi về thời gian, nội dung. Khi kì thi kết thúc hệ thống cần cung cấp chức năng thống kế kết quả của toàn bộ học viên trong một kì thi bất kì. Hệ thống cần có chức năng lưu trữ kết quả kì thi của học viên dưới dạng bản cứng. Về dữ liệu bài giảng thì cần chia làm hai loại bài giảng có thể tạo trực tiếp trên hệ thống và loại bài giảng được chuẩn bị trước và đưa lên hệ thống dưới dạng các file tài liệu dùng làm tài liệu học tập hoặc bài giảng trực tiếp cho khóa học. Hệ thống phải có giao diện cho quản lý có thể tạo bài giảng theo các chủ đề khác nhau và quản ý các chủ đề đã tạo. Quản lý có thể tạo các lớp học. Một khoa học sẽ bao gồm 2-3 học viên với một giáo viên. Nội dung khóa học chia thành các buổi học Mỗi buổi học sẽ có các hình thức học khác nhau đó là học tương tác trực tiếp với giáo viên hoặc học không tương tác với giáo viên.

Teacher (Giáo viên): Giáo viên sau khi đăng nhập có thể xem toàn bộ bài giảng của lớp học mà họ sẽ lên lớp, xem thông tin các học viên trong lớp học. Giáo viên có thể vào phòng học và trở thành thành viên của lớp học ảo. Đối với kì thi có nhưng câu hỏi không thể chấm tự động như câu hỏi speaking, writing thì giáo viên cần có tham gia chấm thi theo sự phân công của quản lý.

Student (Học viên): Học viên có thể đăng kí là thành viên của một trung tâm để tham gia các khóa học sau khi đã có tài khoản và tài khoản được chấp nhận bởi quản lý. Học viên có thể tham gia vào kì thi mà trường tổ chức. Sau khi thi sẽ có kết quả ngay đối với các phần thi có thể chấm tự động. Xem thông tin các thành viên trong lớp học và xem thông tin giáo viên.

Lớp học ảo: Lớp học ảo sẽ có 1 giáo viên và 2 đến 3 học viên tùy theo vào phần cứng và chính sách của các cơ sở đào tạo. Các học viên có thể tham gia tương tác với giáo viên hay các thành viên còn lại thông qua các chức năng của hệ thống chat Realtime, video call, share screen, tạo bảng ảo. Lớp học cần có bài giảng được hiển thị trong lớp học.

* **Yêu cầu phi chức năng**

Hệ thống phải triển khai được trên Unix server. Giao diện đơn giản dễ sử dụng phù hợp với các đối tượng khác nhau.

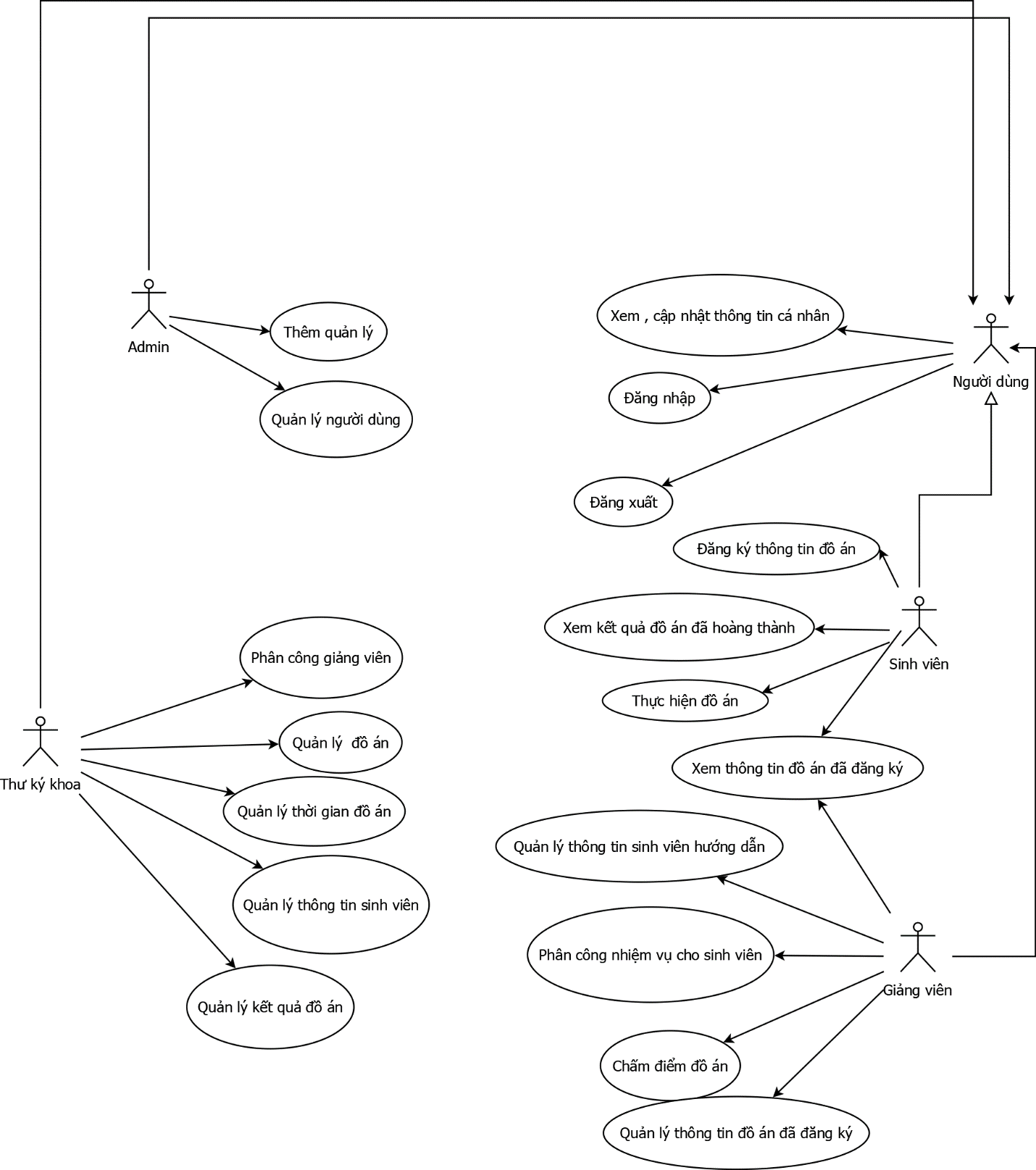
Chức năng học tương tác khi internet bị ngắt kết nối thì học viên hoặc giáo viên có thể vào lại phòng bình thường mà không gặp bất cứ vấn đề gì. Đảm với tốc độ mạng bình thường (tốc độ mạng các hộ gia đình) và cấu hình máy khách có thể chịu được tối thiểu 4 thành viên trong một lớp học.

Chức năng thi của học viên phải hoạt động tốt trên các hệ điều hành khác nhau, trên các trình duyệt khác nhau trong đó có IE, Firefox, Chrome, Opera v.v. Trong quá trình thi, khi sự cố mất kết nối internet xảy ra thì nội dung làm bài trước đó của học viên cần phải được cập nhật để khi kết nối lại phần bài làm của học viên trước đó không bị mất và vẫn còn trên hệ thống.

* **Phân tích và đặc tả yêu cầu**
* **Các tác nhân**

Phần mềm có 4 tác nhân chính là sinh viên, giảng viên, thư ký khoa, admin. Các tác nhân đều có chức năng cơ bản đăng nhập, đăng xuất, xem cập nhật thông tin cá nhân.

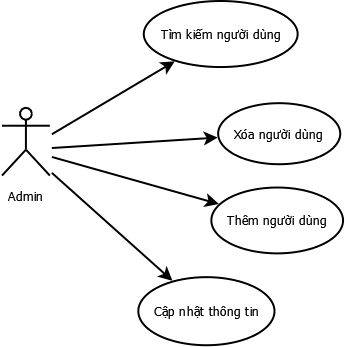
* **Biểu đồ use-case tổng quan**



*Hình 7. Use-case tổng quan*

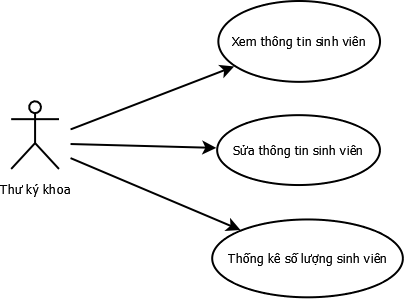
Biểu đồ use-case tổng quan trên thể hiện các chức năng chính theo nhóm người dùng.

* **Biểu đồ use-case phân rã**
* **Admin**



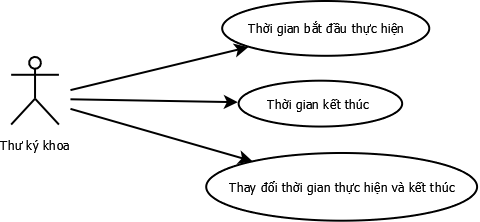
*Hình 8. Use-case quản lí người dùng*

**Phân rã use-case quản lý thông tin sinh viên .**



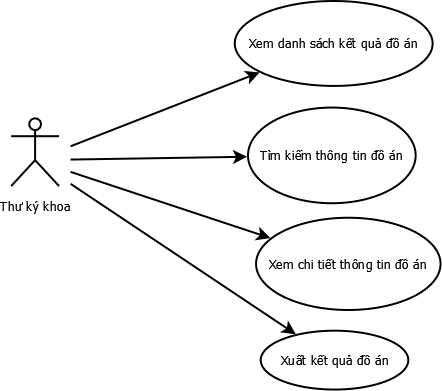
*Hình 9. Use-case quản lí thông tin sinh viên*

**Phân rã use-case quản lý thời gian đồ án**



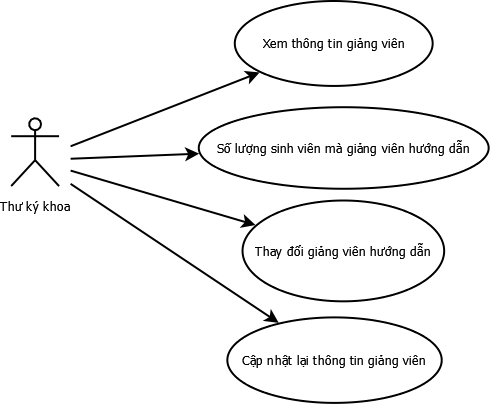
*Hình 10. Use-case quản lý thời gian đồ án*

**Phân rã use-case quản lí kết quả đồ án**



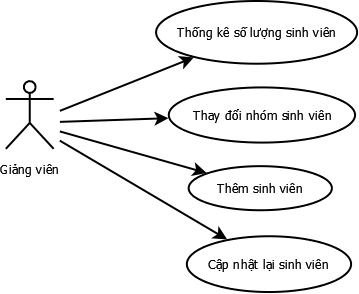
*Hình 11 Use-case quản lý kết quả đồ án*

**Phân rã use-case quản lí phân công giảng viên**



*Hình 12. Use-case quản lí phân công giảng viên*

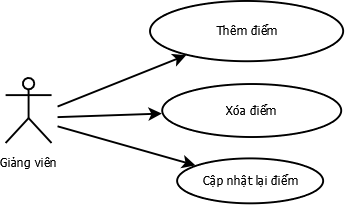
**Phân ra use-case quản lí thông tin sinh viên hướng dẫn**



*Hình 13. Use-case quản lí thông tin sinh viên hướng dẫn*

Hình 13 thể hiển use-case các chức năng quản lí lớp học bao gồm: thêm, sửa, xóa, lớp học. Với chức năng xem thông tin lớp học thì người dùng có thể xem thông tin giáo viên, học viên hoặc xem thông tin bài giảng. Chức năng tạo lớp học bao gồm các chức năng thêm thành viên lớp học, thêm nội dung buổi học. Chức năng thêm thành viên lớp học bao gồm chức năng thêm giáo viên, chức năng thêm học viên. Chức năng thêm nội dung bài giảng bao gồm chức năng thêm chủ để và chức năng thêm tài liệu cho bài giảng.

**Phân rã use-case chấm điểm đồ án**



*Hình 14. Use-case học tương tác*

Hình 14 thể hiện phân rã use-case học tương tác. Các use-case con bao gồm use-case vào phòng học, gọi video, bật tắt audio, video, thay đổi slide, chat realtime, mở thao tác với bảng ảo, rời phòng.

* **Đặc tả các chức năng chính**

**Đặc tả use-case thêm quản lí cho trường**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC001 | **Tên use-case** | Thêm quản lý , người dùng cho hệ thống . |
| **Tác nhân** | Admin | | |
| **Tìền điều kiện** | Người dùng đăng nhập dưới tài khoản admin | | |
| **Luồng sự kiện chính(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Admin | Chọn chức năng quản lý người dùng | | 2. | Hệ thống | Hiển thị giao diện danh sách người dùng | | 3. | Admin | Nhập đầy đủ các thông tin bắt buộc | | 4. | Hệ thống | Validate các giá trị nhập vào | | 5. | Hệ thống | Tạo mới thông tin với các thông tin được nhập bởi admin. | | 6. | Hệ thống | Hiển thị kết quả thông báo cho thêm người dùng thành công cho hệ thống | | | |
| **Luồn sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 4a. | Hệ thống | Hiển thị thông báo chưa nhập đủ trường yêu cầu | | 5a. | Hệ thống | Thông báo tồn tại giá trị không hợp lệ | | 6a. | Hệ thống | Hiển thị thông báo xảy ra lỗi trong quá trình thêm quản lí cho trường | | | |
| **Hậu điều kiện** | Không | | |

*Bảng 1. Đặc tả use-case thêm quản lí cho trường*

**Đặc tả use-case quản lý thông tin sinh viên**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC002 | **Tên use-case** | Quản lý thông tin sinh viên |
| **Tác nhân** | Hệ thống | | |
| **Tìền điều kiện** | Người dùng đã đăng nhập với vai trò là thư ký khoa. Quản lí đang ở giao diện danh sách các câu hỏi. | | |
| **Xem**   |  |  |  |  | | --- | --- | --- | --- | | **Luồng sự kiện chính(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Thư ký khoa | Chọn chức năng xem chi tiết thông tin sinh viên | | 2. | Hệ thống | Hiển thị danh sách sinh viên | |  |  |  | | **Luồng sự kiện thay thế** | Không | | | | **Hậu điều kiện** | Không | | |   **Sửa(U)**   |  |  |  |  | | --- | --- | --- | --- | | **Luồng sự kiện chính(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Thư ký khoa | Chọn chức năng sửa thông tin sinh viên | | 2. | Hệ thống | Lấy dữ liệu thông tin sinh viên | | 3. | Hệ thống | Hiển thị giao diện sửa thông tin sinh viên | | 4. | Quản lý | Nhập các thông tin các trường muốn sửa thông tin sinh viên | | 5. | Hệ thống | Kiểm tra xem các trường bắt buộc đã được nhập đầy đủ chưa | | 6. | Hệ thống | Validate các thông tin dữ liệu các trường | | 7. | Hệ thống | Thông báo kết quả sửa thông tin sinh viên thành công | |  |  |  | | **Luồng sự kiện thay thế** | 6a. | Hệ thống | Thông báo lỗi chưa nhập các trường bắt buộc | | 7a. | Hệ thống | Thông báo lỗi một số trường nhập giá trị không hợp lệ | | 8a. | Hệ thống | Thông báo xảy ra lỗi trong quá trình sửa thông tin. | | **Hậu điều kiện** | Thông báo kết quả trả về sửa thông tin thành công hay thất bại | | |   **Xóa(D)**   |  |  |  |  | | --- | --- | --- | --- | | **Luồng sự kiện chính(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Quản lý | Chọn chức năng xóa thông tin sinh viên | | 2. | Hệ thống | Hiển thị thông báo xác thực việc xóa thông tin sinh viên | | 3. | Quản lý | Xác thực lại muốn xóa thông tin. | | 4. | Hệ thống | Hiển thị thông báo xóa thông tin thành công | |  |  |  | | **Luồng sự kiện thay thế** | 3a. | Quản lý | Xác thực lại không muốn xóa thông tin nữa. | | 3b. | Hệ thống | Đóng hộp thoại xác thực xóa thông tin, quay lại trạng thái ban đầu. | | 4a. | Hệ thống | Thông báo xảy ra lỗi trong quá trình xóa thông tin | | **Hậu điều kiện** | Hiển thị thông báo kết quả của việc xóa thông tin (thành công hoặc thất bại) | | |   **Tìm kiếm(Search)**   |  |  |  |  | | --- | --- | --- | --- | | **Luồng sự kiện chính(Thành công)** | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Quản lý | Chọn chức năng tìm kiếm sinh viên | | 2. | Hệ thống | Hiển thị ô tìm kiếm cho người dung nhập dự liệu tìm kiếm | | 3. | Quản lý | Nhập thông tin liên quan đến sinh viên muốn tìm | | 4. | Hệ thống | Tìm kiếm sinh viên trong bộ dữ liệu thông tin theo dữ liệu người dùng vừa nhập | | 5. | Hệ thống | Hiển thị kết quả tìm kiếm lên giao diện. | | **Luồng sự kiện thay thế** | Không | | | | **Hậu điều kiện** | Không | | | | | | |

*Bảng 2. Đặc tả use-case quản lí ngân hàng câu hỏi*

Các use-case quản lí trường, quản lý đề thi, quản lí kì thi, quản lí bài giảng, quản lí bài giảng tương tự như quản lý câu hỏi.

**Đặc tả use-case upload tài liệu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC003 | **Tên use-case** | Upload tài liệu |
| **Tác nhân** | Quản lý | | |
| **Tìền điều kiện** | Người dùng đã đăng nhập bằng tài khoản quản lý | | |
| **Luồng sự kiện chính(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Quản lý | Chọn chức năng upload tài liệu | | 2. | Hệ thống | Gọi đến chức năng quản lí file trên máy client cho phép người dùng có thể chọn đường dẫn đến file tài liệu muốn upload | | 3. | Quản lý | Chọn file muốn upload lên hệ thống | | 4. | Hệ thống | Kiểm tra người dùng chọn file upload chưa | | 5. | Hệ thống | Liên kết với hệ thông bên thứ ba để upload file lên hệ thống lưu trữ bên thứ ba | | 6. | Hệ thống | Trả về đường dẫn của file | | 7. | Hệ thống | Thông báo file đã upload thành công | | | |
| **Luồn sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 5a. | Hệ thống | Thông báo chưa chọn file upload | | 6a. | Hệ thống | Thông báo lỗi liên kết với hệ thống lưu trữ bên thứ ba | | | |
| **Hậu điều kiện** | Không | | |

*Bảng 3. Đặc tả use-case upload tài liệu*

**Đặc tả use-case tạo buổi học**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC004 | **Tên use-case** | Tạo buổi học |
| **Tác nhân** | Quản lý | | |
| **Tìền điều kiện** | Người dùng đã đăng nhập bằng tài khoản quản lý | | |
| **Luồng sự kiện chính(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Quản lý | Chọn chức năng tạo buổi học trong lớp học | | 2. | Hệ thống | Hiển thị giao diện tạo buổi học | | 3. | Quản lý | Nhập thông tin các trường bắt buộc lớp học | | 4. | Hệ thống | Kiểm tra người dùng nhập đủ thông tin các trường bắt buộc | | 5. | Hệ thống | Thêm buổi học mới vào database | | 6. | Hệ thống | Thông báo tạo buổi học thành công, hiển thị danh sách buổi học mới nhất của lớp học | | | |
| **Luồn sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 5a. | Hệ thống | Thông báo chưa nhập đủ thông tin | | 6a. | Hệ thống | Thông báo xảy ra lỗi khi tạo buổi học | | | |
| **Hậu điều kiện** | Không | | |

*Bảng 4. Đặc tả use-case tạo buổi học*

**Đặc tả use-case làm bài thi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC005 | **Tên use-case** | Làm bài thi |
| **Tác nhân** | Học viên | | |
| **Tìền điều kiện** | Người dùng đã đăng nhập, đã nhập key của kì thi để lấy bài thi | | |
| **Luồng sự kiện chính(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Học viên | Click vào đề thi muốn vào thi | | 2. | Hệ thống | Hiển thị giao diện thi. Giao diện thi bao gồm:  Phía trên chứa thông tin người làm bài thi  Phía giữa và dưới chứa nội dung bài thi các các phím điều hướng  Hiển thị bộ đếm thời gian làm bài | | 3. | Học viên | Làm bài thi bằng cách click chuột vào đáp án lựa chọn đối với các câu hỏi trắc nghiệm, hoặc điền nội dung vào các câu hỏi điền từ.v.v…  Học viên có thể chuyển sang các câu hỏi, các phần khác nhau bằng các phím điều hướng | | 4. | Hệ thống | Hiển thị các nội dung học viên vừa tương tác | | 5. | Hệ thống | Liên tục cập nhật nội dung bài làm theo khoản thời gian cố định | | 6. | Học viên | Chọn chức năng nộp bài | | 7. | Hệ thống | Hiển thị hộp thoại xác thực việc nộp bài | | 8. | Học viên | Xác thực lại việc muốn nộp bài | | 9. | Hệ thống | Lấy thông tin bài làm từ giao diện | | 10. | Hệ thống | Chấm điểm tự động dựa trên nội dung bài làm | | 11. | Hệ thống | Lưu lại các thông tin bài làm và kết quả bài làm trên hệ thống | | 12. | Hệ thống | Hiển thị kết quả thi trên giao diện máy tính | | | |
| **Luồn sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 9a. | Hệ thống | Khi người dùng hủy bỏ việc nộp bài hệ thống đóng hộp thoại xác thực việc nộp bài quay lại giao diện làm bài như ban đầu | | | |
| **Hậu điều kiện** | Cần lưu lại được nội dung bài làm của học viên dù học viên chưa nộp bài | | |

*Bảng 5. Đặc tả use-case làm bài thi*

**Đặc tả use-case vào phòng học**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC006 | **Tên use-case** | Vào phòng học |
| **Tác nhân** | Thành viên lớp học(giáo viên hoặc học viên) | | |
| **Tìền điều kiện** | Người dùng đã đăng nhập, được quản lý thêm vào lớp học | | |
| **Luồng sự kiện chính(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Thành viên lớp học | Chọn chức năng vào phòng học | | 2. | Hệ thống | Lấy thông tin người dùng, thông tin key phòng học | | 3. | Hệ thống | Kiểm tra thông tin người dùng và kiểm tra key có tồn tại không và key có hợp lệ không | | 4. | Hệ thống | Kiểm tra phòng đã tồn tại chưa nếu chưa thì mở phòng còn nếu đã tồi tại thì tham gia vào phòng | | 5. | Hệ thống | Hiển thị giao diện phòng học khi vào phòng. Giao diện hiển thị phải bao gồm giáo trình buổi học video các thành viên lớp học, menu chat, các phím điều khiển trong phòng học | | | |
| **Luồn sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 4a. | Hệ thống | Thông báo không hợp lệ cho việc tham gia vào phòng | | | |
| **Hậu điều kiện** | Không | | |

*Bảng 6. Đặc tả use-case vào phòng học*

**Đặc tả use-case chia sẻ màn hình**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC007 | **Tên use-case** | Chia sẻ màn hình |
| **Tác nhân** | Thành viên lớp học(giáo viên hoặc học viên) | | |
| **Tìền điều kiện** | Người dùng đã đăng nhập, được quản lý thêm vào lớp học, đã vào phòng học, máy tính phía người dùng cần cài đặt plugin screen capturing trên trình duyệt | | |
| **Luồng sự kiện chính(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Thành viên lớp học | Chọn chức chia sẻ màn hình | | 2. | Hệ thống | Hệ thống lấy thông tin màn hình máy tính đang hiển thị, hiển thị giao diện cho người dùng chọn màn hính muốn share(Do tại một thời điểm máy tính có thể đang làm việc với nhiểu màn hình khác nhau). | | 3. | Thành viên lớp học | Chọn màn hình muốn chia sẻ | | 4. | Hệ thống | Quay lại video màn hình tại máy client muốn chia sẻ | | 5. | Hệ thống | Tạo stream video để chia sẻ dữ liệu màn hình tới các thành viên khác trong cùng phòng học | | | |
| **Luồn sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 4a. | Hệ thống | Đóng cửa sổ chọn màn hình nếu người dùng không muốn chia sẻ màn hình | | | |
| **Hậu điều kiện** | Các máy tính khác trong cùng một phòng phải hiển thị được màn hình được chia sẻ | | |

*Bảng 7. Đặc tả use-case chia sẻ màn hình*

**Đặc tả use-case chấm thi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã Use-case** | UC008 | **Tên use-case** | Chấm thi |
| **Tác nhân** | Giáo viên | | |
| **Tìền điều kiện** | Người dùng đã đăng nhập với vai trò là giáo viên, và đã được qaunr lý phân công bài chi chấm thi | | |
| **Luồng sự kiện chính(Thành công)** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 1. | Giáo viên | Chọn bài thi muốn chấm, chọn chức chấm thi | | 2. | Hệ thống | Hiển thị nội dung bài làm của học viên với trường điểm bỏ trống cho các câu hỏi cần chấm | | 3. | Giáo viên | Nhập giá trị điểm cho câu cần chấm | | 4. | Hệ thống | Kiểm tra giá trị nhập có hợp lệ không | | 5. | Giáo viên | Submit kết quả chấm thi lên hệ thống | | 6. | Hệ thống | Sẽ tính toán giá trị điểm tổng cập nhập điểm của bài thi lên hệ thống lưu trữ | | 7. | Hệ thống | Hiển thị thông báo lưu kết quả chấm thành công | | | |
| **Luồn sự kiện thay thế** | |  |  |  | | --- | --- | --- | | **STT** | **Thực hiện bởi** | **Hành động** | | 4a | Hệ thống | Thông báo giá trị điểm nhập không hợp lệ | | | |
| **Hậu điều kiện** | Không | | |

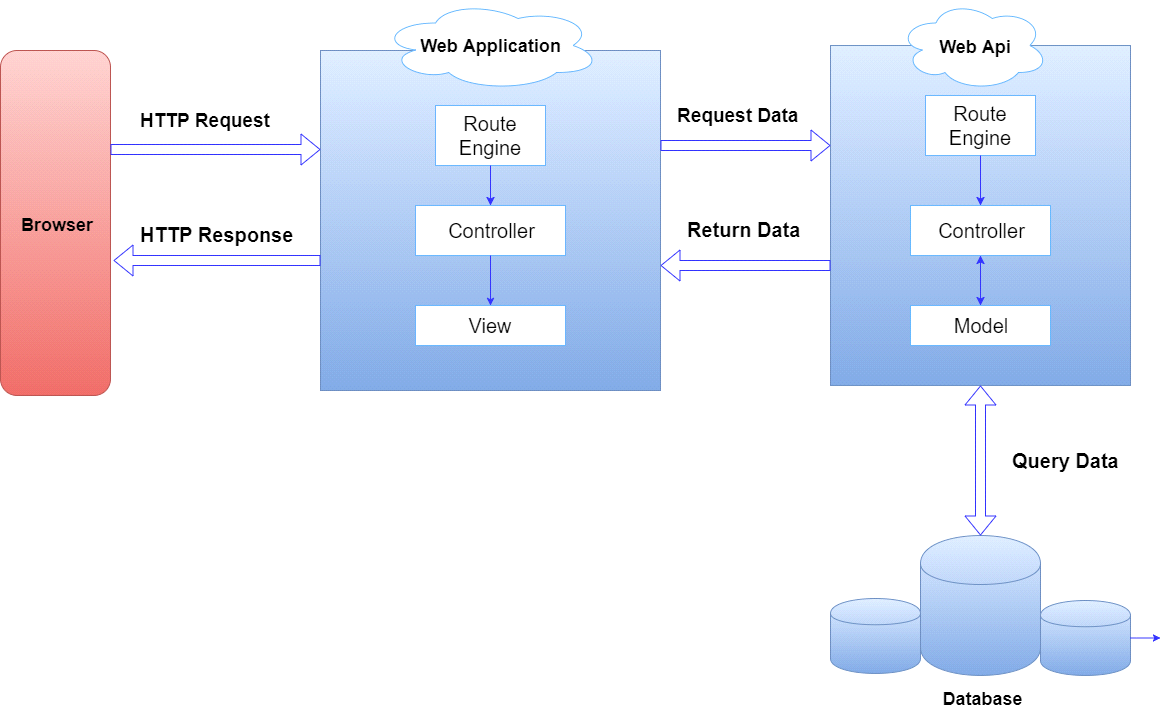
*Bảng 8. Đặc tả use-case chấm thi*

**CHƯƠNG 4: THIẾT KẾ**

|  |
| --- |
| ***Nội dung chương này em sẽ trình bày các vấn đề chính sau đây: Thứ nhất, em sẽ trình bày kiến trúc hệ thống, biểu đồ trình tự cho các chức năng chính hệ thống, sau đó là thiết kế cơ sở dữ liệu, thiết kế lớp và thiết kế giao diện.*** |

* **Kiến trúc hệ thống**

Các thành phần chính của hệ thống bao gồm WebApplication, Web API, Hệ quản trị cơ sở dữ liệu (xem Hình 15)



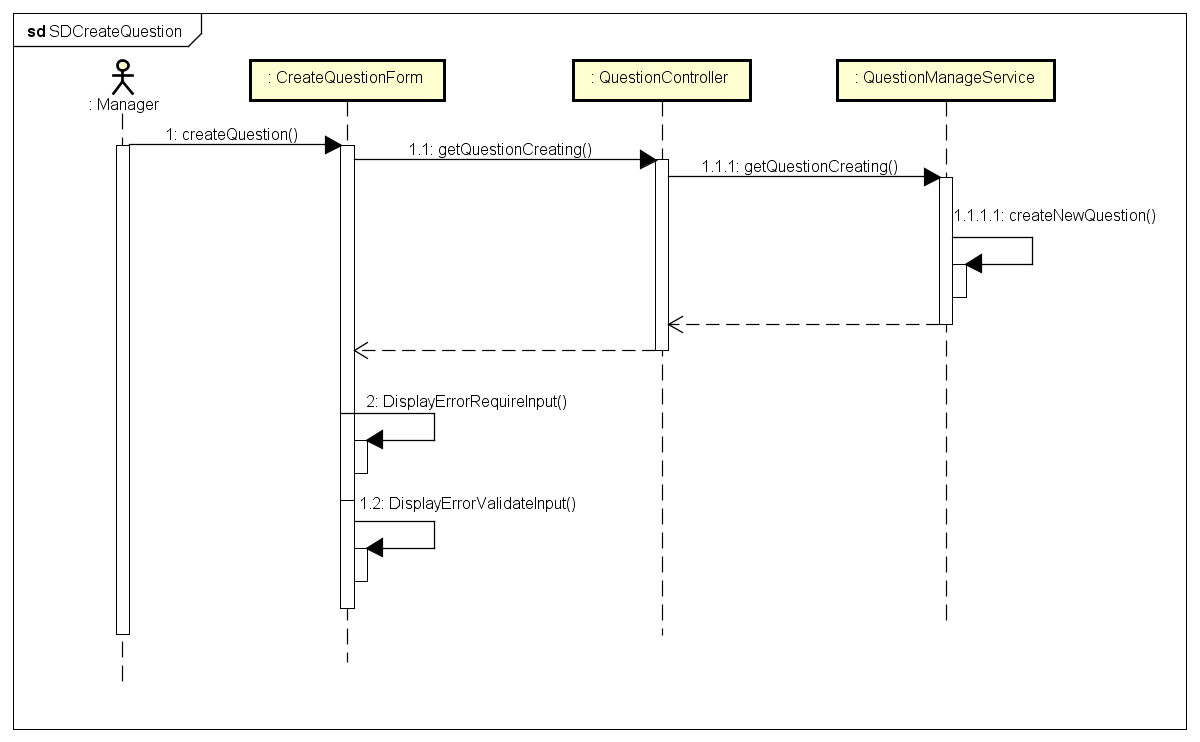
*Hình 15. Mô hình kiến trúc hệ thống*

* Database: nơi sẽ quản lý toàn bộ dữ liệu của hệ thống và cung cấp các các chức năng thao tác với dữ liệu.
* Web Service là thành phần sẽ kết nối trực tiếp với hệ quản trị cơ sở dữ liệu làm việc với database. Thành phần này sẽ cung cấp các api cho các ứng dụng gọi tới nó. Mọi hoạt động lấy dữ liệu, hay thao tác với dữ liệu đều phải thông qua thành phần này. Việc tách biệt riêng một service để làm việc với database sẽ giúp hệ thống có thể dễ dàng mở rộng trên các nền tảng khác nhau.
* Web Application đảm nhiệm chức năng tương tác với người dùng. Web Application sẽ lấy dữ liệu thông qua các hàm được cung cấp từ Web Api để

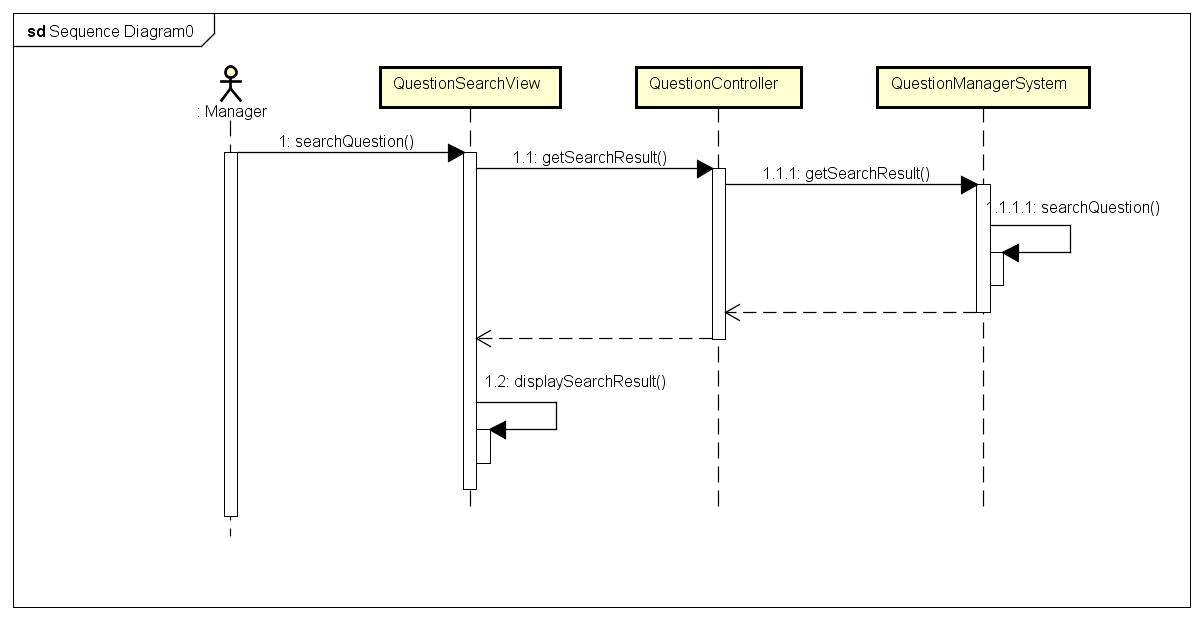
Xử lý và hiển thị trên giao diện người dùng.

* **Biểu đồ trình tự các chức năng chính**

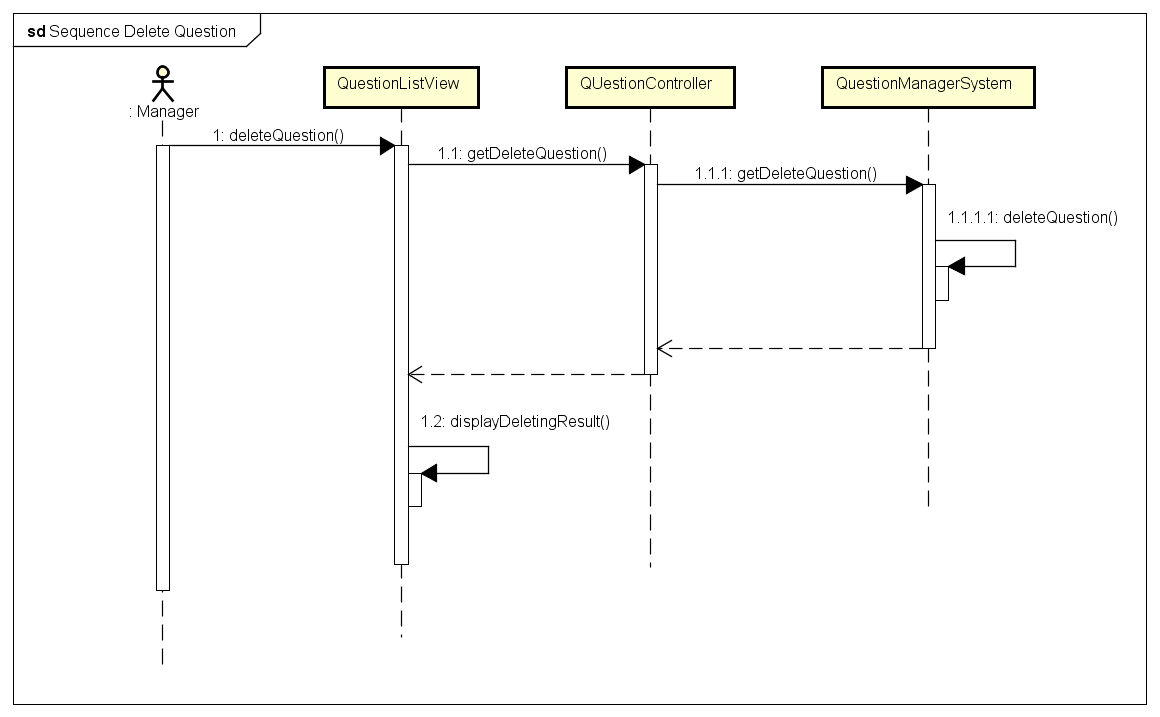
**Use-case quản lý ngân hàng câu hỏi**



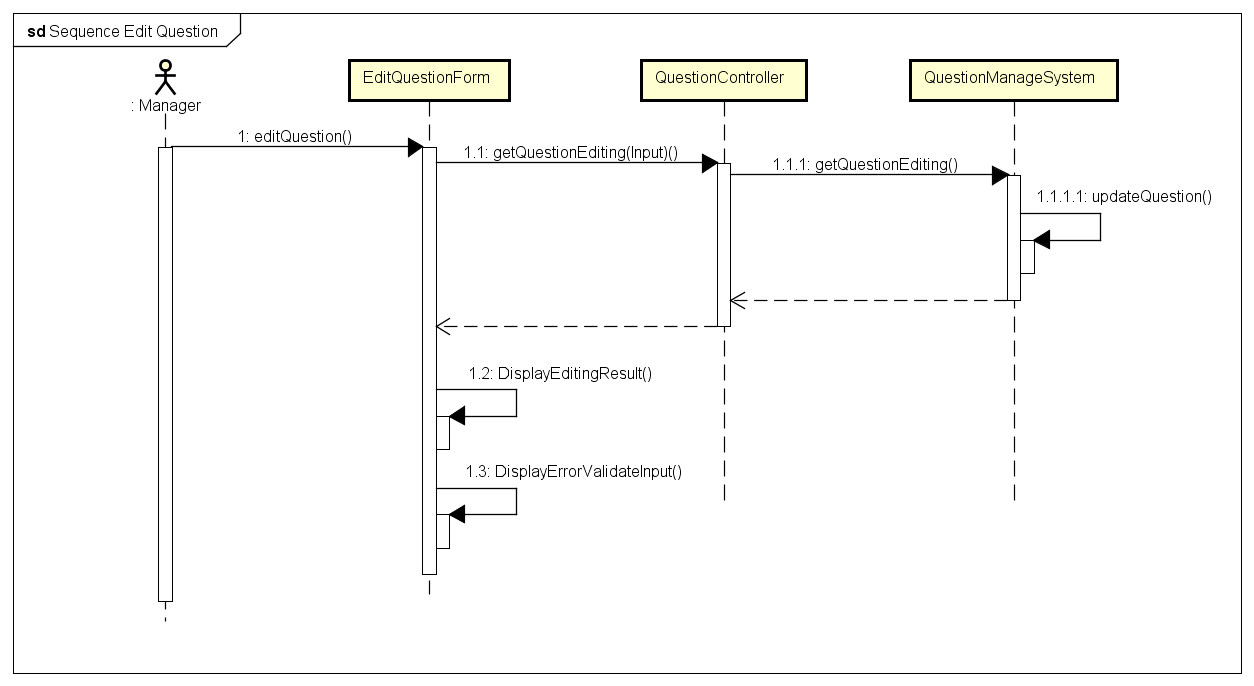
*Hình 16. Biểu đồ trình tự use-case tạo câu hỏi*



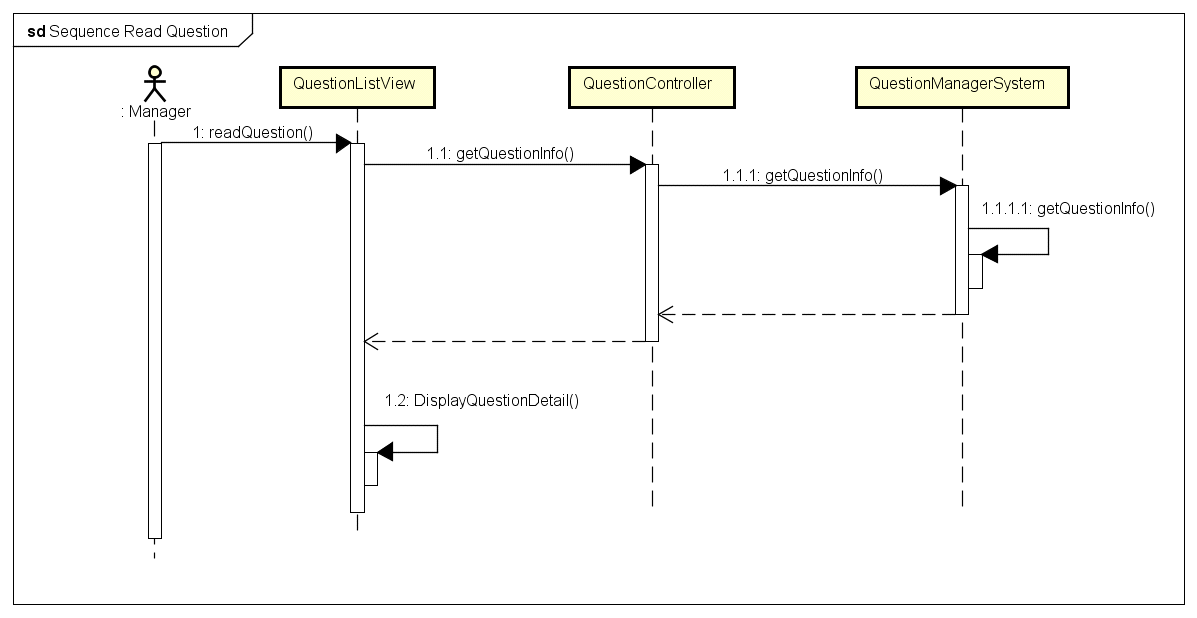
*Hình 17.Biểu đò trình tự use-case tìm kiếm câu hỏi*



*Hình 18. Biểu đồ trình tự use-case xóa câu hỏi*



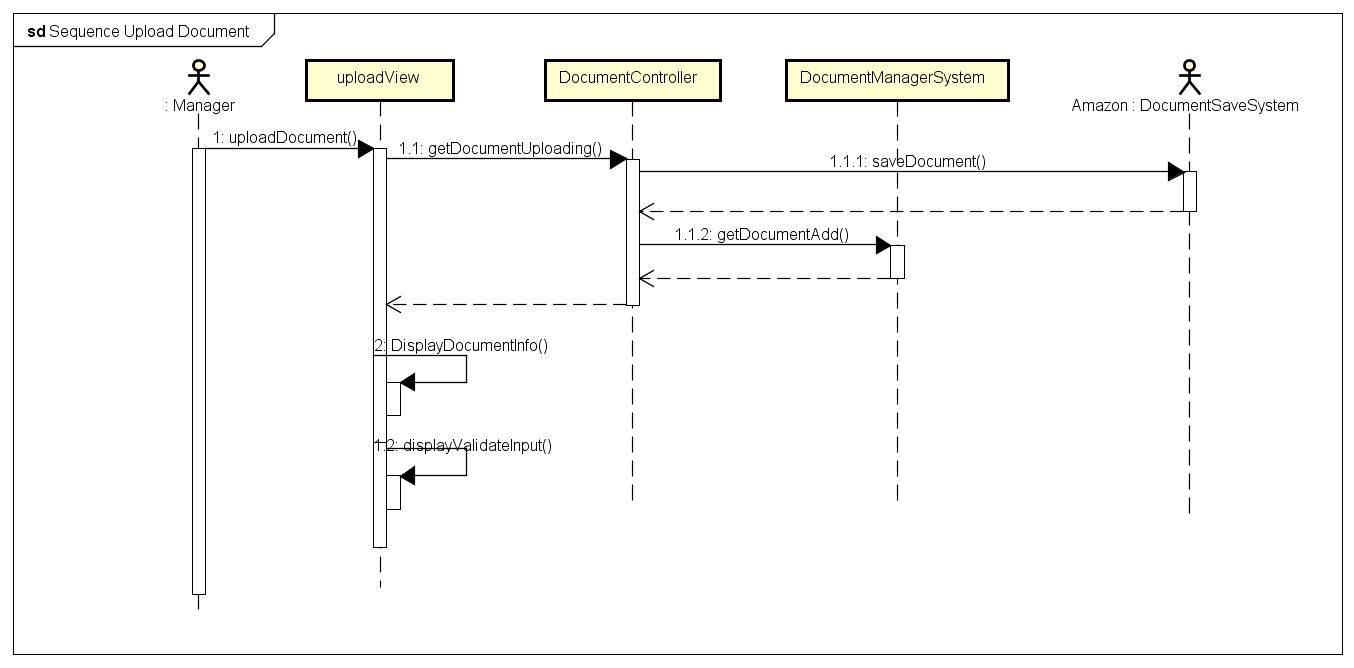
*Hình 19. Biểu đồ trình tự cập nhật câu hỏi*



*Hình 20. Biểu đồ trình tự use-case xem chi tiết câu hỏi*

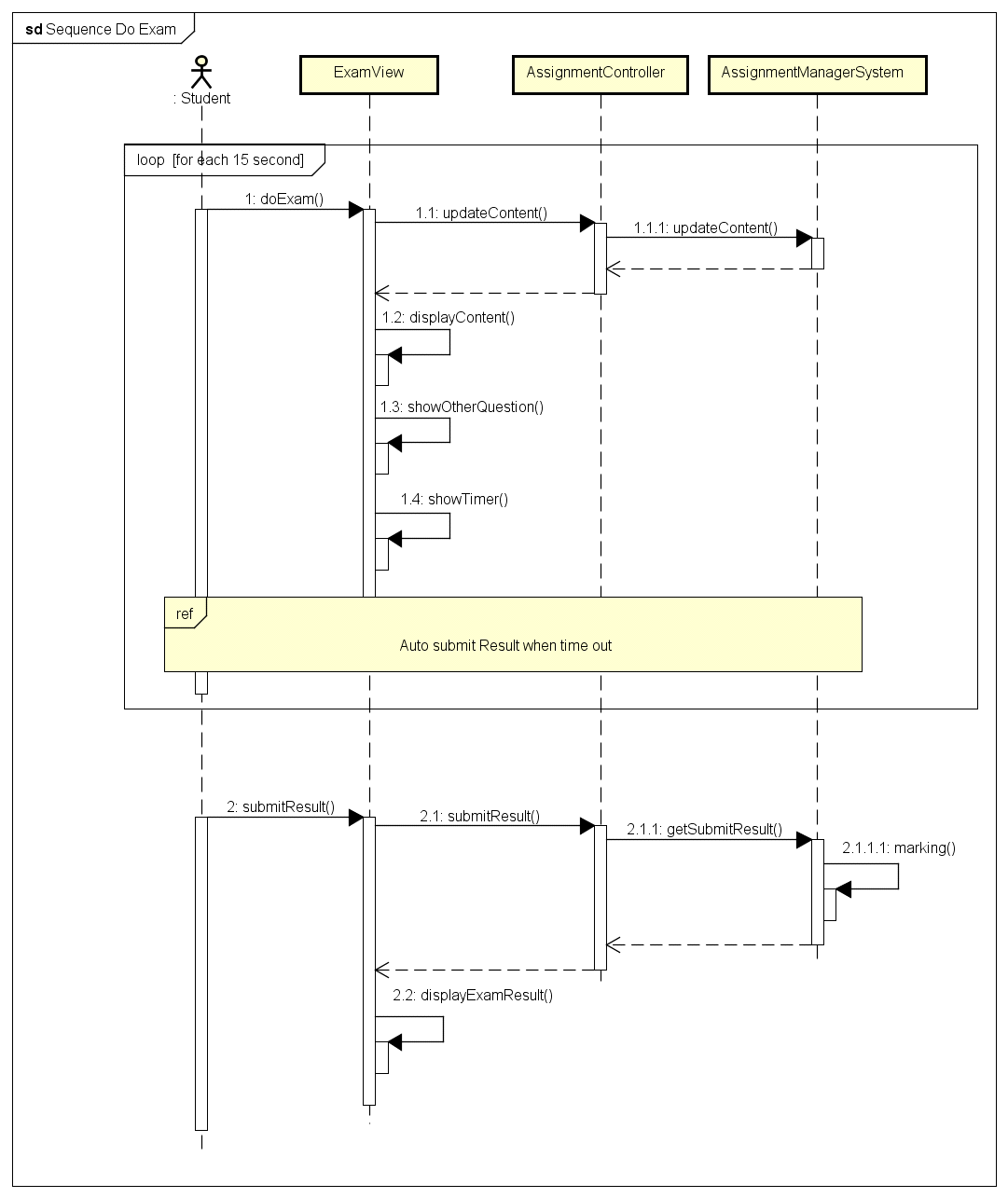
Các use-case quản lý đề thi, quản lí kì thi, quản lí ngân hàng bài giảng cũng tương tự như use-case quản lí ngân hàng câu hỏi.

**Use-case upload tài liệu**



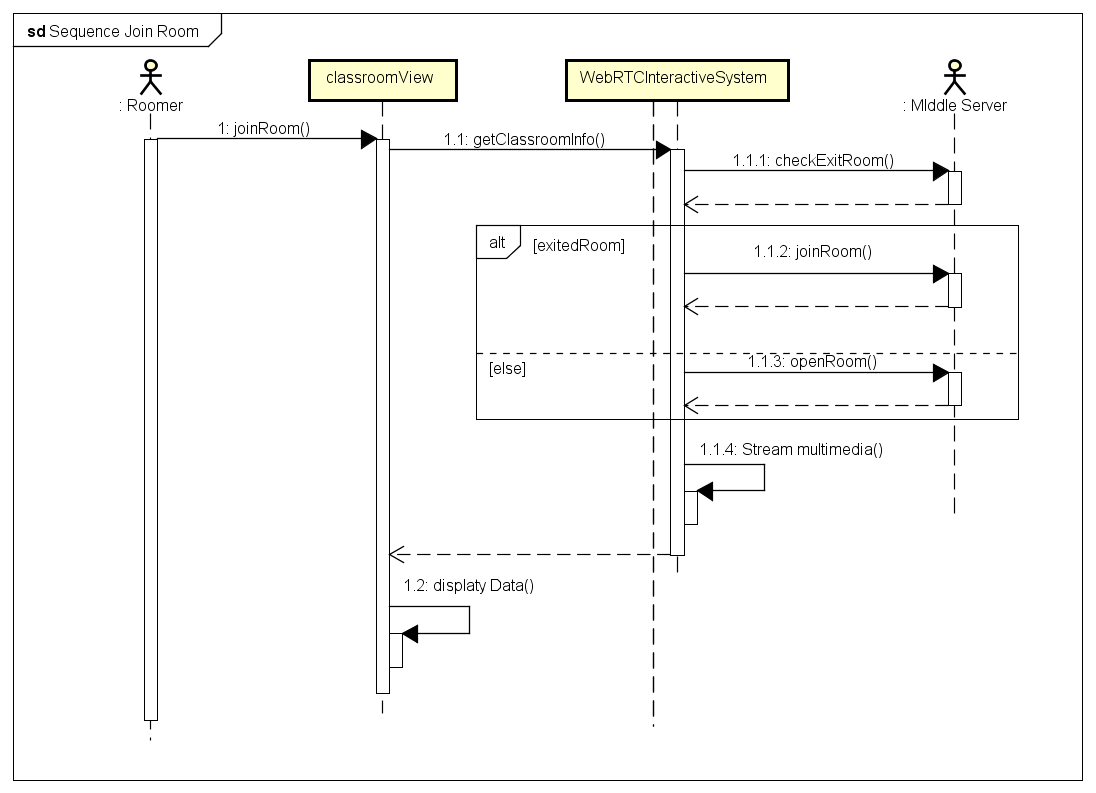
*Hình 21. Biểu đồ trình tự use-case upload tài liệu*

**Use-case làm bài thi**



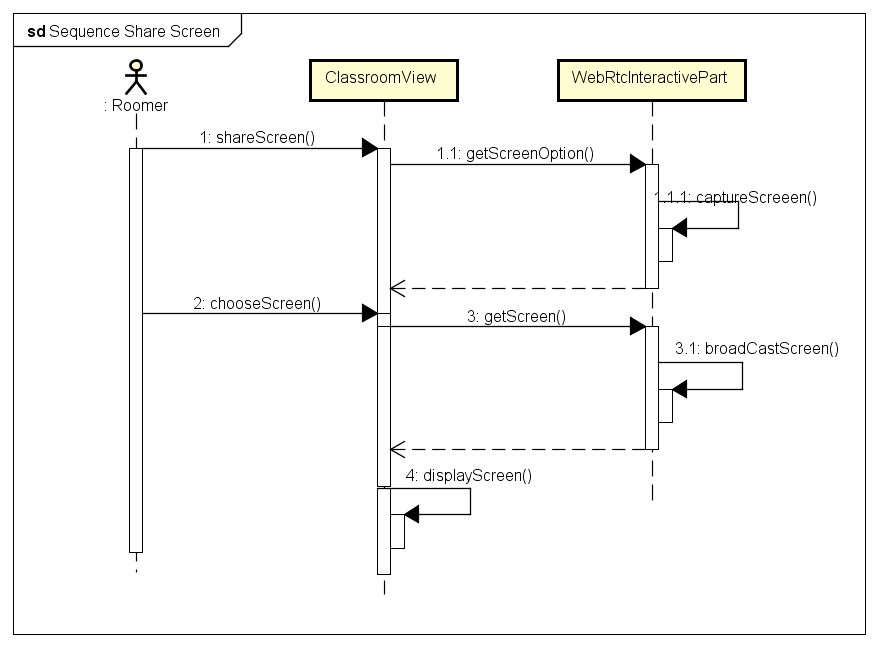
*Hình 22. Biểu đồ trình tự use-case làm bài thi*

**Use-case vào phòng học**



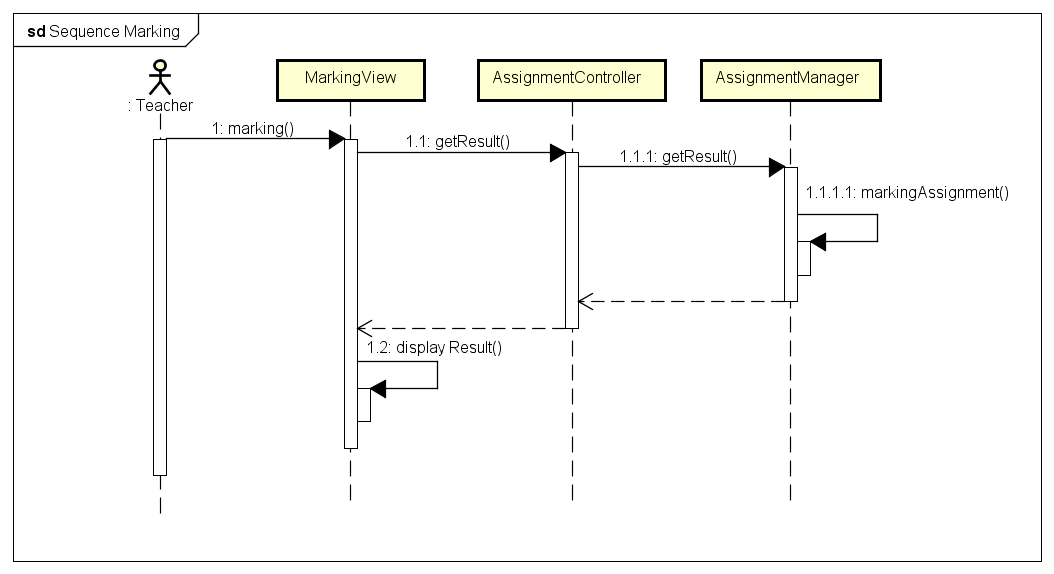
*Hình 23. Biểu đồ trình tự use-case vào phòng học*

**Use-case chia sẻ màn hình**



*Hình 24. Biểu đồ trình tự use-case chia sẻ màn hình*

**Use-case chấm thi**

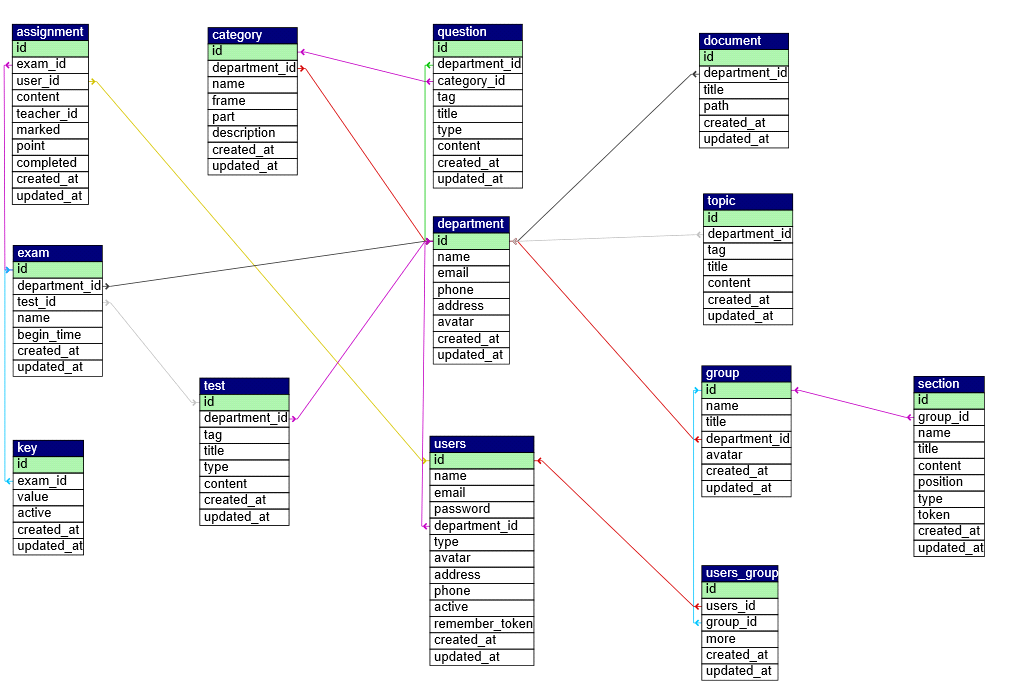


*Hình 25. Biểu đồ trình tự use-case chấm thi*

* **Thiết kế cơ sở dữ liệu**

Hệ thống sử dụng hệ quản trị cơ sở dữ liệu MySQL cho việc lưu trữ dữ liệu.

Hình 26 thể hiện thể hiện cấu trúc các bảng dữ liệu chính và quan hệ giữa các bảng với nhau.



*Hình 26. Cơ sở dữ liệu*

**Bảng cơ sở dữ liệu người dùng**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Interger | x |  | Mã người dùng |
| **Name** | Varchar |  |  | Tên người dùng |
| **Email** | Varchar |  |  | Email người dùng để đăng nhập |
| **Password** | Varchar |  |  | Mật khẩu người dùng |
| **Avatar** | Varchar |  |  | Đường dẫn avatar người dùng |
| **Phone** | Varchar(15) |  |  | Số điện thoại liên hệ |
| **Type** | TinyInteger(4) |  |  | Loại thành viên mà người dùng muốn tham gia trường học |
| **Department\_id** | Varchar |  |  | Mã trường học mà học viên tham gia |
| **Address** | Varchar |  |  | Địa chỉ người dùng |
| **Active** | Boolean |  |  | Thể hiện trạng thái người dùng đã được quản lý trường phê duyệt là thành viên chưa |
| **Created\_at** | Timestamp |  |  | Thời gian tạo trường học |
| **Remember\_token** | Varchar |  |  | Token ghi nhớ đăng nhập |
| **Created\_at** | Timestamp |  |  | Thời gian người dùng đăng kí |
| **Updated\_at** | Timestamp |  |  | Thời gian sửa gần nhất |

*Bảng 9. Cơ sở dữ liệu cho bảng users*

**Bảng cơ sở dữ liệu trường học (trung tâm, cơ sở đào tạo)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã trường học |
| **Name** | Varchar |  |  | Tên trường học |
| **Email** | Varchar |  |  | Email trường học |
| **Phone** | Varchar(15) |  |  | Số điện thoại trường học |
| **Address** | Varchar |  |  | Địa chỉ |
| **Created\_at** | Timestamp |  |  | Thời gian tạo trường học |
| **Updated\_at** | Timestamp |  |  | Thời gian sửa gần nhất |

*Bảng 10. Cơ sở dữ liệu cho bảng Department*

**Bảng cơ sở dữ liệu câu hỏi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã câu hỏi |
| **Department\_id** | Integer |  | x | Mã trường học mà câu hỏi thuộc |
| **Category\_id** | Integer |  | x | Mã danh mục chứa câu hỏi |
| **Tag** | Varchar |  |  | Nhãn câu hỏi (phục vụ cho việc tìm kiếm nhanh) |
| **Title** | Varchar |  |  | Tiêu đề câu hỏi |
| **Type** | TinyInteger(4) |  |  | Loại câu hỏi(Câu hỏi có rất nhiều loại) |
| **Created\_at** | Timestamp |  |  | Thời gian tạo câu hỏi |
| **Updated\_at** | Timestamp |  |  | Thời gian lần cuối sửa câu hỏi |

*Bảng 11. Cơ sở dữ liệu cho bảng question*

**Bảng cơ sở dữ liệu đề thi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã đề thi |
| **Department\_id** | Integer |  | X | Mã trường học |
| **Tag** | Varchar |  |  | Nhãn đề hi |
| **Type** | TinyInteger(4) |  |  | Loại đề thi |
| **Content** | MediumText |  |  | Nội dung đề thi |
| **Created\_at** | Timestamp |  |  | Thời gian tạo đề thi |
| **Updated\_at** | Timestamp |  |  | Thời gian sửa đề thi gần nhất |

*Bảng 12. Cơ sở dữ liệu cho bảng test*

**Bảng cơ sở dữ liệu kì thi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã kì thi |
| **Department\_id** | Integer |  | X | Mã trường học |
| **Test\_id** | Integer |  | X | Mã đề thi |
| **Name** | Varchar |  |  | Tên kì thi |
| **Begin\_time** | Timestamp |  |  | Thời gian bắt đầu thi |
| **Created\_at** | Timestamp |  |  | Thời gian tạo kì thi |
| **Updated\_at** | Timestamp |  |  | Thời gian sửa kì thi gần nhất |

*Bảng 13. Cơ sở dữ liệu cho bảng exam*

**Bảng cơ sở dữ liệu bài thi của học viên**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã bài thi |
| **Exam\_id** | Integer |  | X | Mã kì thi |
| **User\_id** | Integer |  | X | Mã học viên |
| **Content** | MediumText |  |  | Nội dung bài làm |
| **Teacher\_id** | Integer |  |  | Mã giáo viên chấm thi(Có thể rỗng) |
| **Marked** | Boolean |  |  | Trạng thái bài thi đã chấm hay chưa chấm |
| **Point** | Double |  |  | Điểm của bài thi |
| **Completed** | Boolean |  |  | Trạng thái bài thi đã hoàn thành chưa |
| **Created\_at** | Timestamp |  |  | Thời gian tạo bài làm |
| **Updated\_at** | Timestamp |  |  | Thời gian cập nhật nội dung gần nhất |

*Bảng 14. Cơ sở dữ liệu cho bảng Assignment*

**Bảng cơ sở dữ liệu lớp học**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã lớp học |
| **Name** | Varcher |  |  | Tên lớp học |
| **Title** | Varchar |  |  | Tên lớp học |
| **Department\_id** | Integer |  | X | Mã trường học |
| **Avatar** | Varcher |  |  | Đường dẫn avatar của lớp học |
| **Created\_at** | Timestamp |  |  | Thời gian tạo lớp học |
| **Updated\_at** | Timestamp |  |  | Thời gian gần nhất sửa thông tin lớp học |

*Bảng 15. Cơ sở dữ liệu cho bảng lớp học*

**Bảng cơ sở dữ liệu buổi học**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã buổi học |
| **Group\_id** | Integer |  | X | Mã lớp học chứa buổi học |
| **Name** | Varchar |  |  | Tên buổi học |
| **Title** | Varchar |  |  | Tiêu đề buổi học |
| **Content** | MediumText |  |  | Nội dung buổi học |
| **Type** | TinyInteger(4) |  |  | Loại buổi học(buổi học tương tác hoặc buổi học không tương tác) |
| **Position** | TinyInteger(4) |  |  | Vị trí buổi học (thể hiện thứ tự buổi học trên tổng số buổi học) |
| **Token** | Varchar |  |  | Là chuỗi string vai trò như khóa đề vào phòng học đối với các buổi học tương tác |
| **Created\_at** | Timestamp |  |  | Thời gian tạo buổi học |
| **Updated\_at** | Timestamp |  |  | Thời gian gần nhất sửa thông tin buổi học |

*Bảng 16. Cơ sở dữ liệu cho bảng section*

**Bảng cơ sở dữ liệu chủ đề (bài giảng tạo trên hệ thống theo chủ đề)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã chủ đề |
| **Department\_id** | Integer |  | X | Mã trường học |
| **Tag** | Varchar |  |  | Nhãn chủ đề |
| **Title** | Varchar |  |  | Tiêu đề chủ đề bài giảng |
| **Content** | MediumText |  |  | Nội dung chủ đề |
| **Created\_at** | Timestamp |  |  | Thời gian tạo chủ đề |
| **Updated\_at** | Timestamp |  |  | Thời gian gần nhất sửa thông tin chủ đề |

*Bảng 17. Cơ sở dữ liệu cho bảng topic*

**Bảng cơ sở dữ liệu key**

Bảng này chứa các key được sinh ra từ mỗi kì thi, một một key sẽ tương ứng với một lượt thi của một học viên. Học viên muốn có bài thi trên tài khoản thì phải có một key.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Khóa chính** | **Khóa ngoài** | **Ý nghĩa** |
| **Id** | Integer | x |  | Mã key |
| **Exam\_id** | Integer |  | X | Mã kì thi |
| **Value** | Varchar |  |  | Chứa key(là chuỗi sinh ngẫu nhiên) |
| **Active** | Boolean |  |  | Trạng thái key đã được sử dụng hay chưa |
| **Created\_at** | Timestamp |  |  | Thời gian tạo key |
| **Updated\_at** | Timestamp |  |  | Thời gian gần nhất sửa thông tin key |

*Bảng 18. Cơ sở dữ liệu cho bảng key*

* **Thiết kế lớp**

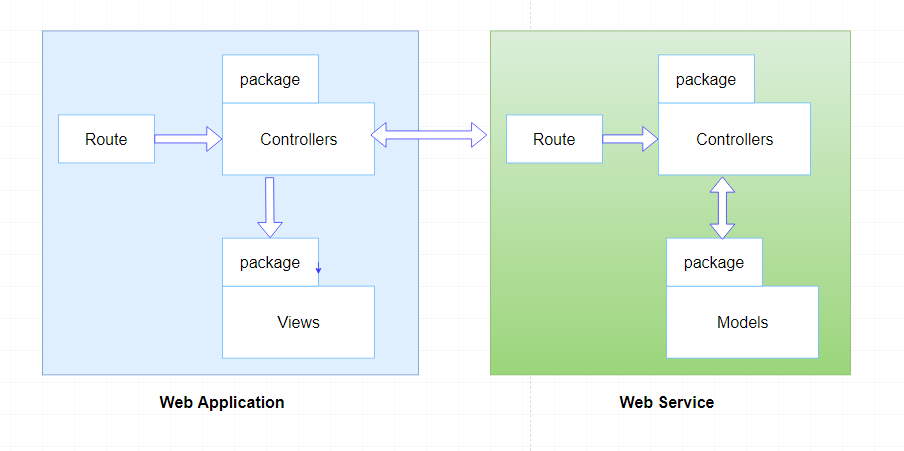
Hệ thống chia thành nhiều package, mỗi package đảm nhiệm một chức năng ( xem Hình 27)

Route là nơi chứa file điều hướng các request từ trình duyệt. Route giúp ánh xạ URL đến đúng tài nguyên server yêu cầu dựa trên các file route.

Controllers là package chứa các lớp điều khiển (Controller) hệ thống. Đây là nới điều hướng và xử lí dữ liệu trước khi yêu cầu dữ liệu từ API.

Views: Là package chứa các file view giúp hiển thị dữ liệu trên trình duyệt.

Models: Là package chứa các lớp Model, dùng để thao tác trực tiếp với dữ liệu và trả về cho Controller.



*Hình 27. Cấu trúc module của hệ thống*

**Biểu đồ lớp chức năng quản lý ngân hàng câu hỏi**

Hình 28 thể hiện các lớp sử dụng cho chức năng quản lý ngân hàng câu hỏi trong đó:

Lớp ListQuestionView có chức năng hiển thị danh sách các câu hỏi đã có trên hệ thống kèm theo đó là các phím chức năng điều hướng sang các trang khác

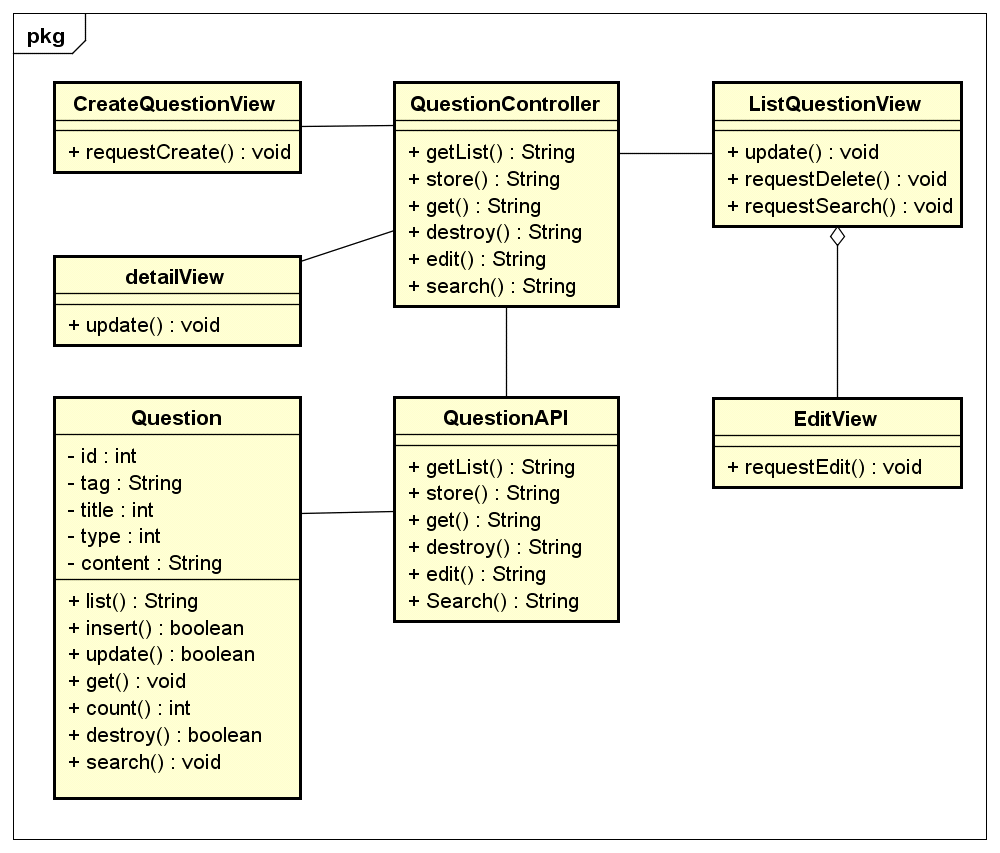
Lớp CreateQuestionView hiển thị giao diện cho người dùng tạo câu hỏi với các trường cho người dùng nhập và phương thức yêu cầu tạo câu hỏi

Lớp detailView hiển thị chi tiết câu hỏi với đầy đủ thông tin câu hỏi

Lớp EditView hiển thị giao diện form câu hỏi cho người dùng có thể sửa thông tin câu hỏi

Lớp QuestionController làm nhiệm vụ nhận các yêu cầu từ các view từ đó sẽ xử lí dữ liệu rồi gửi yêu cầu tới lớp Model để yêu cầu thao tác với dữ liệu rồi điều hướng dữ liệu đến các view tương ứng.

Lớp Question là lớp Model đảm nhiệm chức năng làm việc, thao tác với dữ liệu bảng của database.



*Hình 28. Biểu đồ lớp use-case quản lí ngân hàng câu hỏi*

Biểu đồ lớp chức năng quản lý ngân hàng câu hỏi cũng tương tự cho chức năng quản lý đề thi, quản lí kì thi, quản lí bài giảng, quản lí lớp học.

**Biểu đồ lớp cho quá trình thi của học viên**

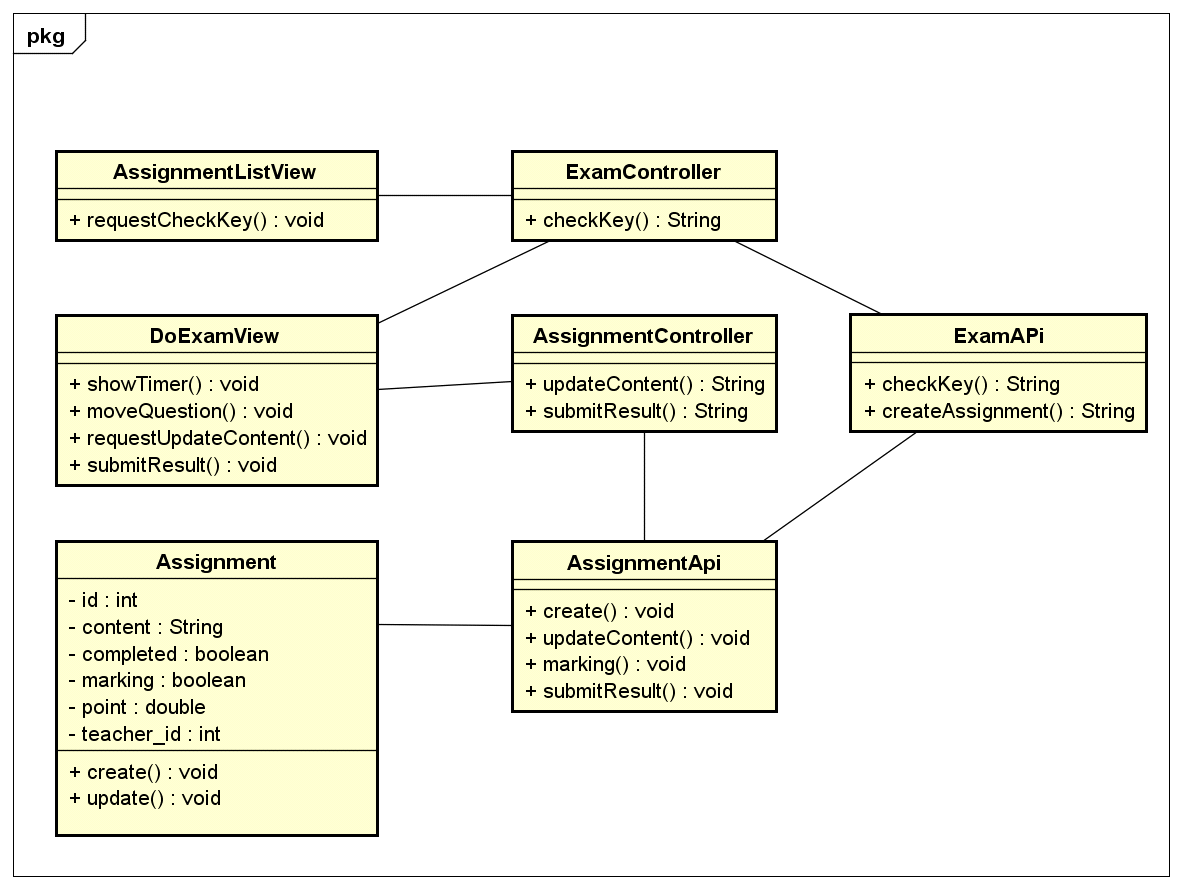
Hình 29thể hiện quá trình thi của học viên.

AssignmentListView sẽ hiện thị giao diện danh sách bài thi của học viên đã làm. Để tham gia một kì thi thì học viên sẽ được cấp một key và học viên chỉ cần nhập key trên giao diện sau đó AssignmentListView gọi hàm requestCheckKey kiểm tra key

Lớp ExamController sẽ lấy key chuyển từ view đến sau đó gọi Api checkKey() kiểm tra xem có hợp lệ không. Nếu key hợp lệ hàm createAssignment() sẽ được gọi.

Lớp DoExamView có chức năng hiển thị giao diện làm bài thi của học viên.Hàm showTimer() hiển thị bộ đếm thời gian làm bài. Hàm moveQuestion() dùng để chuyển sang các câu hỏi khác khi làm bài. Hàm requestUpdate() dùng để cập nhật nội dung bài làm của học viên trong trường hợp trong quá trình làm bài bị lỗi kết nối thì nội dung bài làm của học viên vẫn còn do đã được lưu lại nên không phải làm lại. Hàm submitResult() dùng để gọi chức năng submit kết quả sau khi đã hoàn thành bài thi.

Lớp AssignmentController có các hàm updateContent() dùng đề gọi api update cập nhật nội dung bài làm của học viên. Hàm submitResult() dùng để gọi api submitResult để cập nhật kết quả cuối cùng của bài thi.

Lớp Assignment dùng đề thao tác với bảng assignment để tạo, cập nhật nội dung bài làm của học viên 

*Hình 29. Biểu lớp quá trình thi của học viên*

**Biểu đồ lớp cho quá trình quản lý lớp học và vào phòng học**

Hình 30 thể hiện cho quá trình quản lí lớp của quản lý và quá trình học của học viên. Biểu đồ bào gồm các lớp sau:

Lớp ListClassView để hiển thị danh sách các lớp đang mở trên hệ thống.

Lớp DetailClassView để hiển thị chi tiết các lớp học. Hàm showListSection hiển thị danh sách các buổi học của học viên. Hàm showInfoMember hiển thị chi tiết các thành viên trong lớp học bao gồm giáo viên và các học viên khác.

Lớp CourseView hiển thị chi tiết các buổi học của khóa học. Chi tiết khóa học bao gồm các buổi học và nội dung đi kèm. Đối với các buổi học bình thường không tương tác với giáo viên thì sẽ có tài liệu đi kèm nên học viên có thể xem thông qua trang này. Đối với các buổi học có sự tương tác với giáo viên thì học viên có thể vào phòng học thông qua trang này.

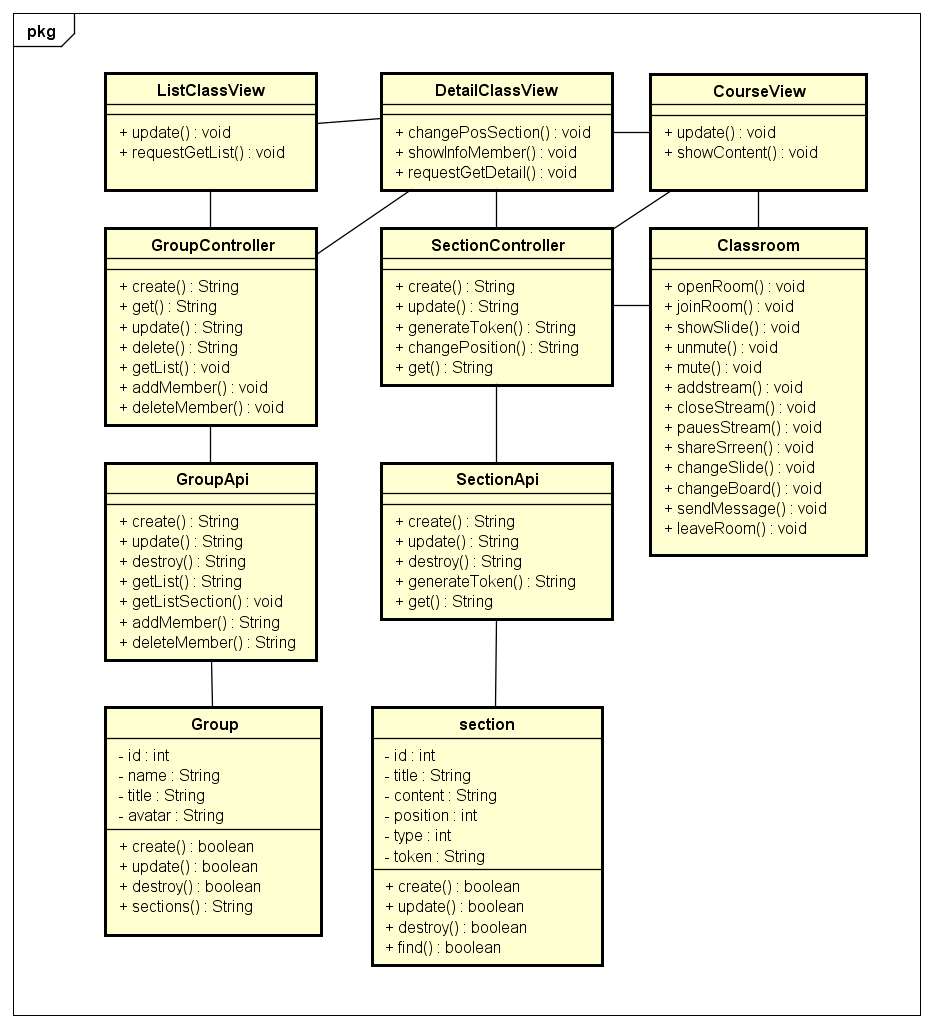
Lớp GroupController sẽ hiện các nghiệp vụ liên quan đến lớp học như xử lý dữ liệu trước khi gọi các api. Các chức năng chính củ lớp này gồm tạo mới, thêm, sửa, xóa lớp học, lấy danh sách lớp học, thêm thành viên, xóa thành viên lớp học.

SectionController đảm nhiệm chức năng tạo và quản lí các buổi học. Hàm generateToken() sẽ sinh ra các token. Token đóng vai trò như chìa khóa để vào đúng phòng học. Ngoài ra lớp này còn cung cấp thêm hàm changePosition() thay đổi vị trí các buổi học tùy theo ý của người tạo.

Các lớp GroupApi và SectionApi đảm nhiệm các chức năng tương ứng với các lớp GroupController, SectionController.

Các lớp Model Group và Section thực hiện chức năng tương tác với hệ quản trị cơ sở dữ liệu làm việc với dữ liệu.

Lớp Classroom sẽ hiển thi giao diện phòng học với các chức năng chính tạo phòng vào phòng, xem slide bài giảng tắt tiếng, bật tiếng, thêm stream(luồng chia sẻ dữ liệu với các máy khác) mới, close stream, tạm dừng stream, chia sẻ màn hình, thay đổi slide, bật tắt chế độ bảng ảo, gửi tin nhắn, rời phòng học.



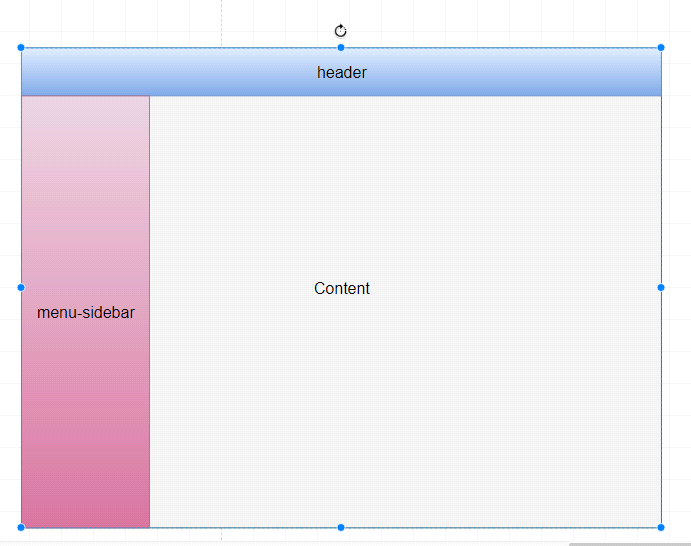
*Hình 30. Biểu đồ lớp quá trình quản lí lớp và học*

* **Thiết kế giao diện**

**Giao diện quản lí trường admin**

Hình 31 thể hiện giao các thành phần chính của giao diện các trang quản lí của người dùng.

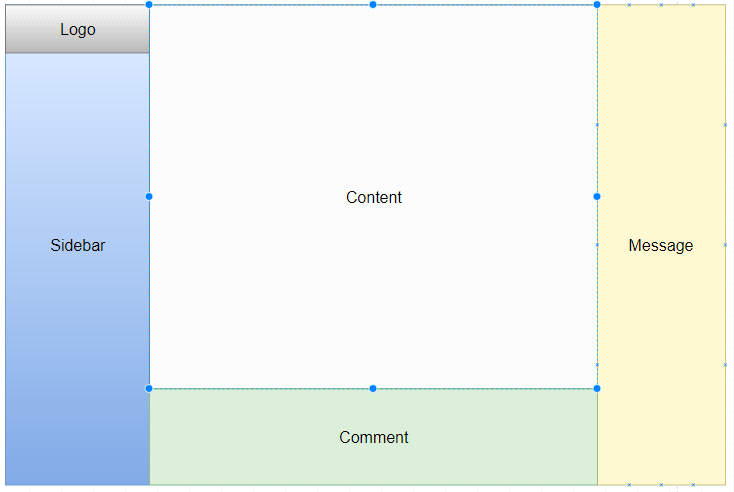
Phần **header** sẽ hiển thị thông tin về hệ thống như logo, thông tin người dùng, đăng xuất, avatar. Phần **menu-sidebar** hiển thị danh sách các chức năng tương ứng với người dùng. Phần **content** sẽ hiển thị nội dung chính của trang tùy thuộc vào các trang khác nhau thì phần này sẽ hiển thị khác nhau.



*Hình 31. Giao diện tổng quan cho nghiệp vụ quản lý*

**Giao diện học của học viên**

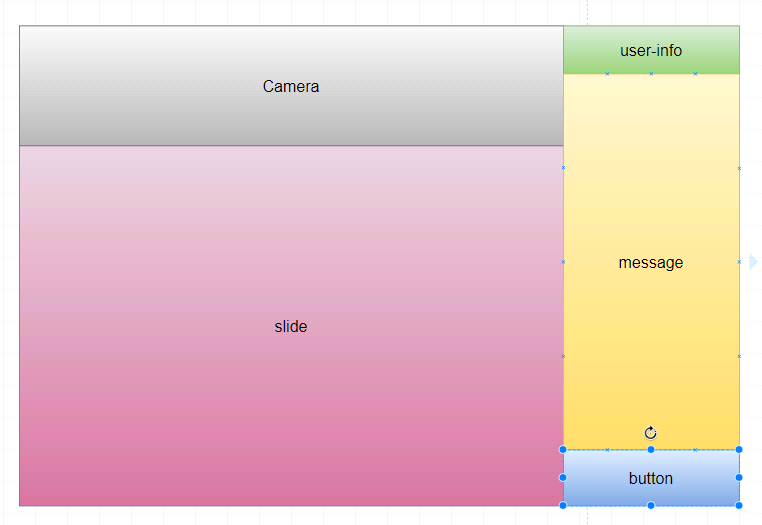
Hình 32 thể hiện giao diện của học viên khi vao khóa học. **Sidebar** hiển thị menu danh sách các bài giảng với các tiêu đề bài giảng được hiển thị ngắn gọn. Tại đây học viên có thể đi đến bất kì buổi học muốn học. Phần **Content** hiển thị nội dung chi tiết của buổi học bao gôm text, audio, video khóa học. Phần **Message** hiển thị tin nhắn đến của học viên trong lớp học. Phần **Comment** hiển thị form cho phép người dùng bình luận y kiến về buổi học



*Hình 32. Giao diện học viên vào khóa học*

**Giao diện phòng học của học viên**

Hình 33 thể hiện giao diện của phòng học ảo khi học viên hoặc giáo viên tham gia vào phòng học. Phần **Camera** hiển thị giao diện camera của các thành viên trong lớp. Phần **Slide** là nơi hiển thị bài giảng trực tuyến cho phòng học đó. Phần **User-info** hiển thị thông tin người đang ở trong phòng. Phần **Message** hiển thị phần chat của phòng học với các thành viên. Phần **Button** hiển thị các phím chức năng trong phòng học như các phím vào phòng, thoát phòng, share screen, chuyển chế độ bảng ảo.



*Hình 33. Giao diện phòng học*

**CHƯƠNG 5: TRIỂN KHAI HỆ THỐNG VÀ KIỂM THỬ**

|  |
| --- |
| ***Chương này em sẽ trình bày hướng dẫn cài đặt và triển khai hệ thống, kiểm thử hệ thống, một số giao diện chính sau khi cài đặt và đánh giá hệ thống.*** |

* **Cài đặt hệ thống**
* **Môi trường thử nghiệm hệ thống**
* Hệ điều hành: Window
* Máy chủ ảo: Xampp
* Trình duyệt: Chrome và Firefox
* Mạng: Wifi LAN
* Sản phẩm: Thư mục project bao gồm các thư mục web, api, server-socket, file database.sql
* **Cài đặt**

**Cài đặt phần mềm máy chủ ảo: Xampp**

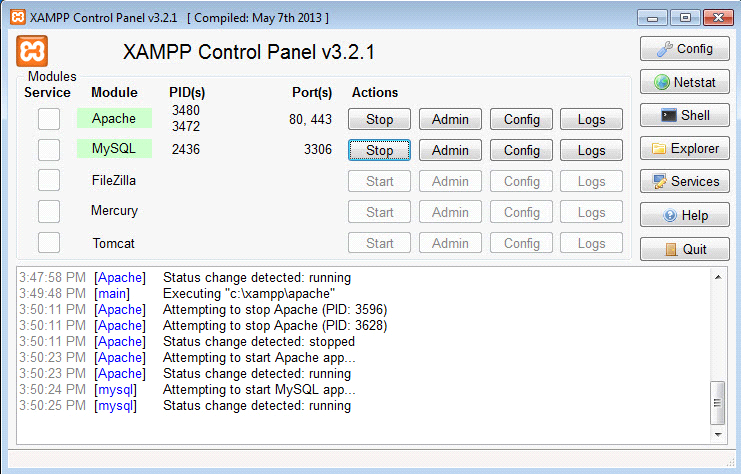
Máy chủ ảo localhost là một máy chủ được vận hành trực tiếp trên máy tính của bạn có tính năng tương đương như trên máy chủ thật. Để hỗ trợ lập trình viên có thể lập trình ngay trên máy tính nên tốc độ xử lý công việc sẽ nhanh hơn.

**Xampp** là chương trình tạo máy chủ ảo được tích hợp sẵn [Apache](https://vi.wikipedia.org/wiki/Apache_(HTTP)), [PHP](https://vi.wikipedia.org/wiki/PHP), [MySQL](https://vi.wikipedia.org/wiki/MySQL), [FTP Server](https://vi.wikipedia.org/w/index.php?title=FTP_Server&action=edit&redlink=1), [Mail Server](https://vi.wikipedia.org/w/index.php?title=Mail_Server&action=edit&redlink=1) và các công cụ như [phpMyAdmin](https://vi.wikipedia.org/wiki/PhpMyAdmin). Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.



Download tại <https://www.apachefriends.org/download.html> , chọn phiên bản tương ứng với hệ điều hành và tiến hành cài đặt như các phần mềm bình thường.

Sau đó khởi động Xampp Control Panel và chọn start 2 mục Apache và MySQL



**Cài đặt chương trình**

Chương trình bao gồm 2 thư mục mã nguồn *api*, web, file cơ sở dữ liệu *database.sql*

Sau khi cài Xampp, copy 2 thư mục mã nguồn vào thư mục htdocs của Xampp.

Vào phpmyadmin tạo database và import file *database.sql.*

Mở file database.php trong thư mục api/application/config vào sửa cho phù hợp với database vừa tạo.

Sau khi cài đặt xong, chạy chương trình bằng cách nhập địa chỉ localhost/project vào trình duyệt Chrome, Firefox hoặc Cốc Cốc. Nếu xuất hiện giao diện login thì cài đặt thành công.

* **Kiểm thử hệ thống**

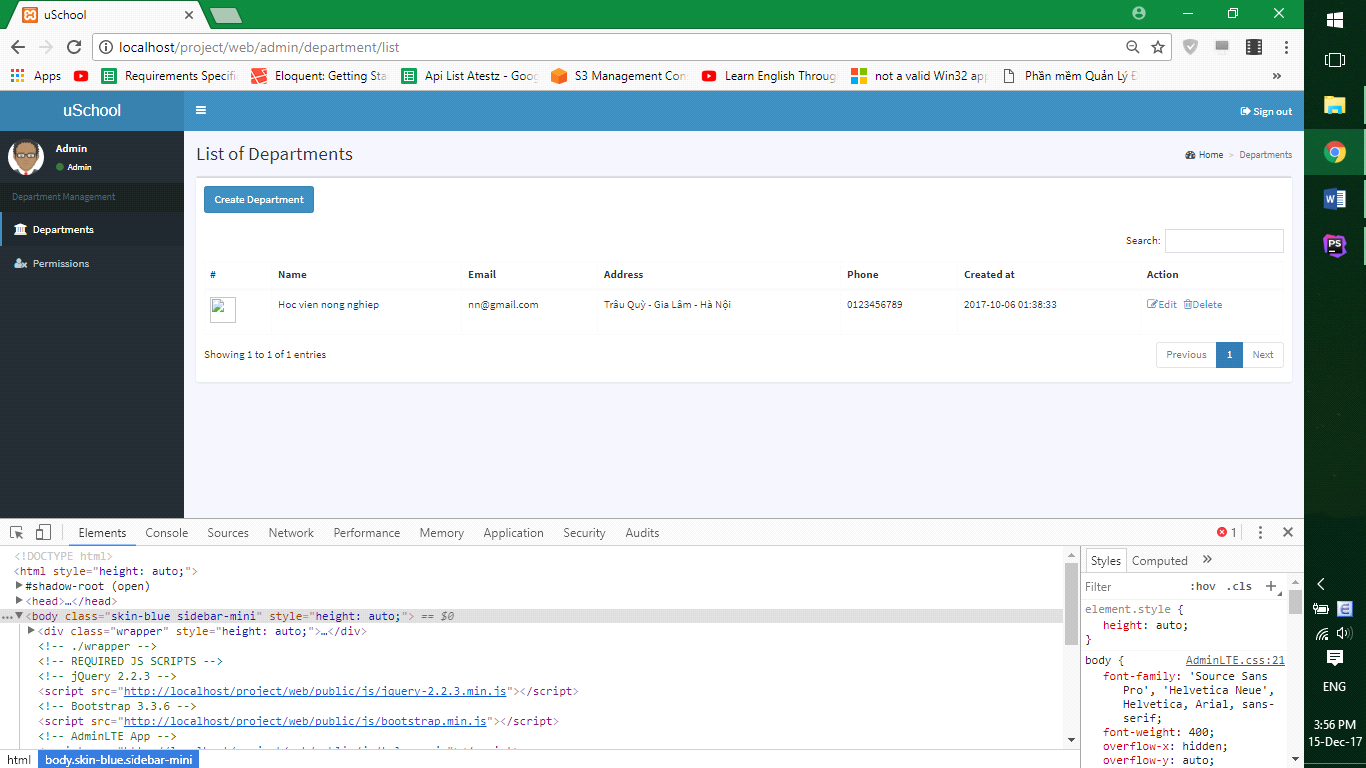
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Last update date** | 14-Dec-17 | | **Reviewed date** | 14-Dec-17 | | |
|  | **Pass** | **Fail** | **Untested** | **N/A** | **Accepted** | **Number of Test cases** |
| **Chức năng cơ bản người dùng** | **11** | **1** | **0** | **0** | **1** | **12** |
| **Quản lí ngân hàng câu hỏi** | 8 | 0 | 0 | 0 | 0 | 10 |
| **Quản lí ngân hàng đề thi** | 12 | 0 | 0 | 0 | 0 | 12 |
| **Quản lí kì thi** | 9 | 0 | 0 | 0 | 0 | 9 |
| **Quản lí ngân hàng bài giảng** | 16 | 0 | 0 | 0 | 0 | 16 |
| **Quản lí lớp học** | 20 | 0 | 0 | 0 | 0 | 20 |
| **Chức năng tương tác trong phòng học ảo** | 20 | 0 | 0 | 0 | 0 | 20 |
| **Quản lí trường** | 12 | 0 | 0 | 0 | 0 | 12 |
| **Chức năng học của học viên** | 8 | 0 | 0 | 0 | 0 | 8 |
| **Chức năng giáo viên** | 10 | 0 | 0 | 0 | 0 | 10 |
| **Tổng** | 119 | 1 | 0 | 0 | 1 | 120 |
| **Test coverage** |  |  |  |  |  | 99,17% |
| **Test successful coverage** |  |  |  |  |  | 99,17% |

*Bảng 19. Bảng tổng kết kết quả các test-case*

* **Một số giao diện chức năng**

Kết quả sau khi cài đặt, hệ thống đã thực hiện được một số chức năng quan trọng. Dưới đây là một số hình ảnh của các chức năng chạy trên hệ thống.

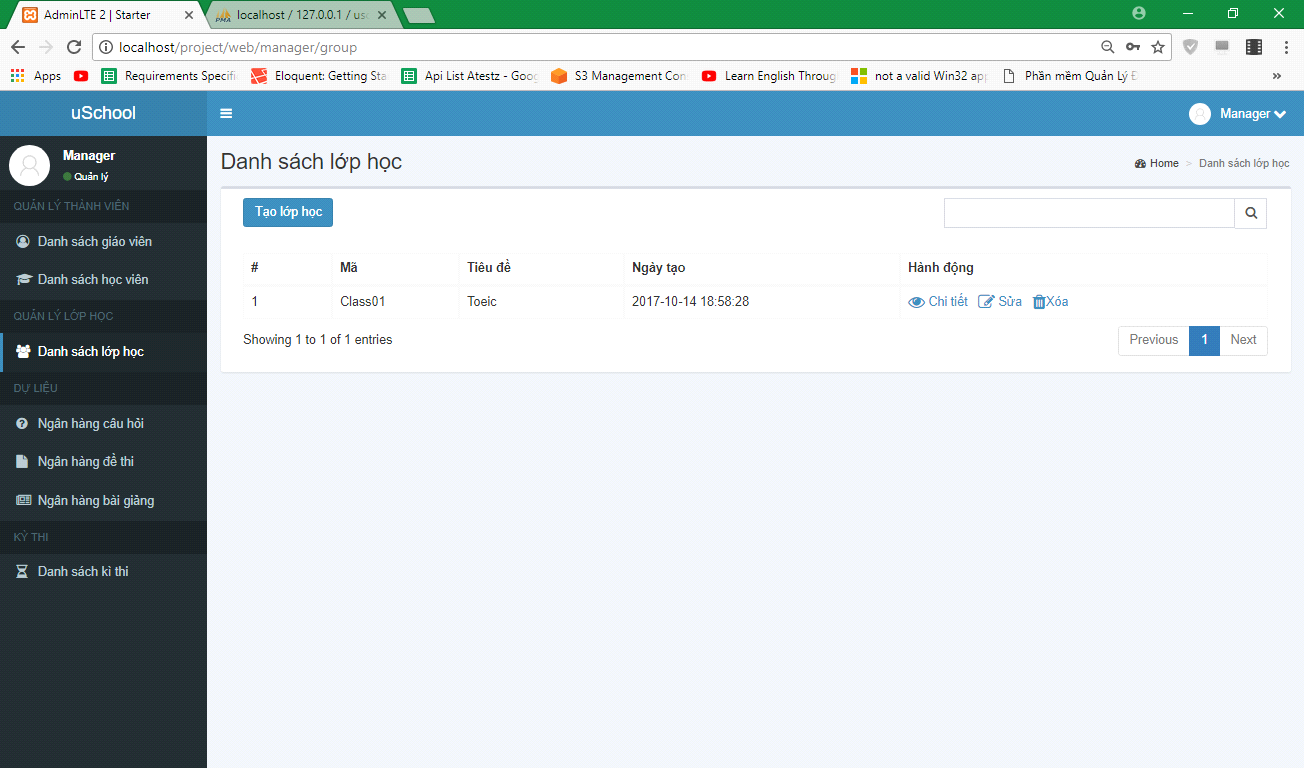
**Giao diện trang chủ phía admin**



*Hình 34. Giao diện trang chủ phía admin*

Hình 34 mô tả giao diện phía admin với chức năng chính quản lý trường và thêm quản lí cho trường.

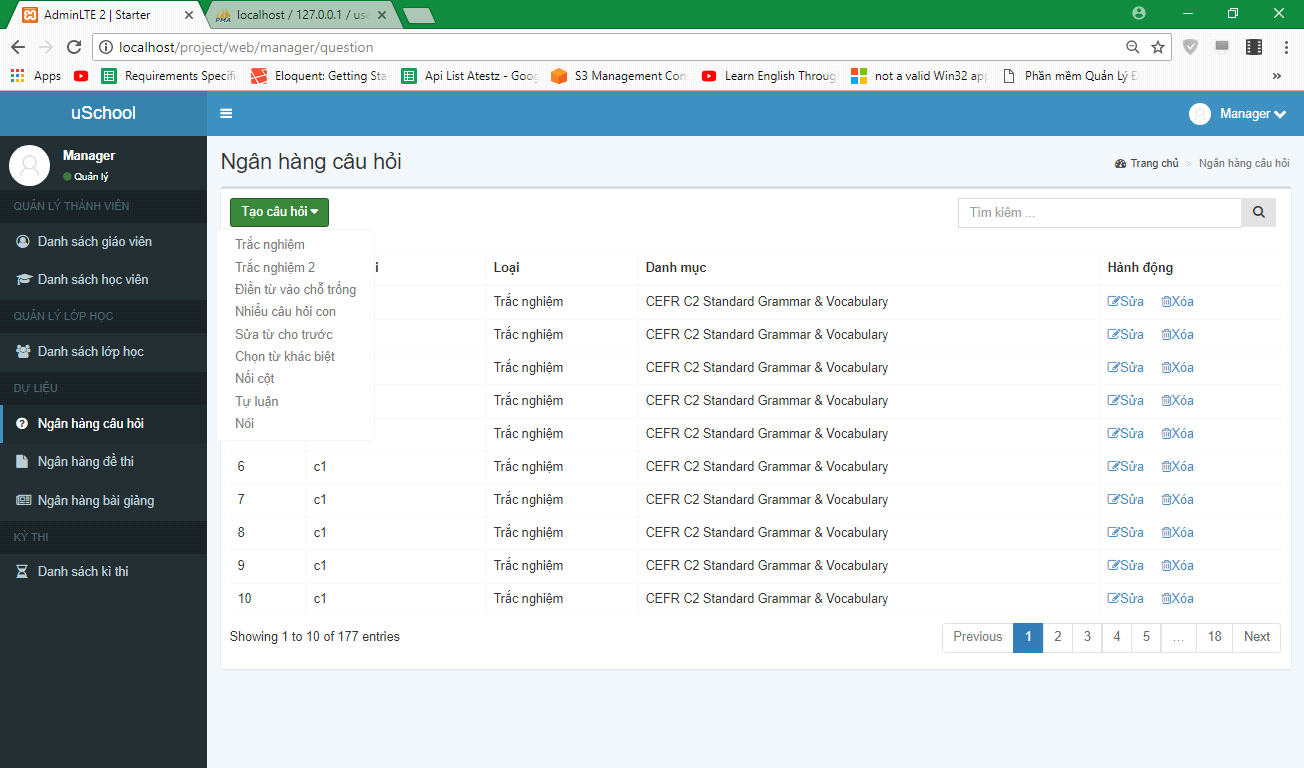
**Giao diện trang chủ phía quản lí**



*Hình 35. Giao diện trang chủ phía quản lý*

Hình 35 trên thể hiện giao diện trang chủ phía quản lí với phía bên trái là sidebar thể hiện danh sách các chức năng mà quản lý có thể làm. Thanh header phái trên hiển thị logo với thông tin cá nhân quản lí. Ở giữa là phần thể hiện nội dung chính.

**Giao diện quản lí ngân hàng câu hỏi**



*Hình 36. Giao diện quản lí ngân hàng câu hỏi*

Hình thể hiện giao diện ngân hàng câu hỏi với phần nội dụng hiển thị danh sách các câu hỏi với các chức năng quản lí câu hỏi.

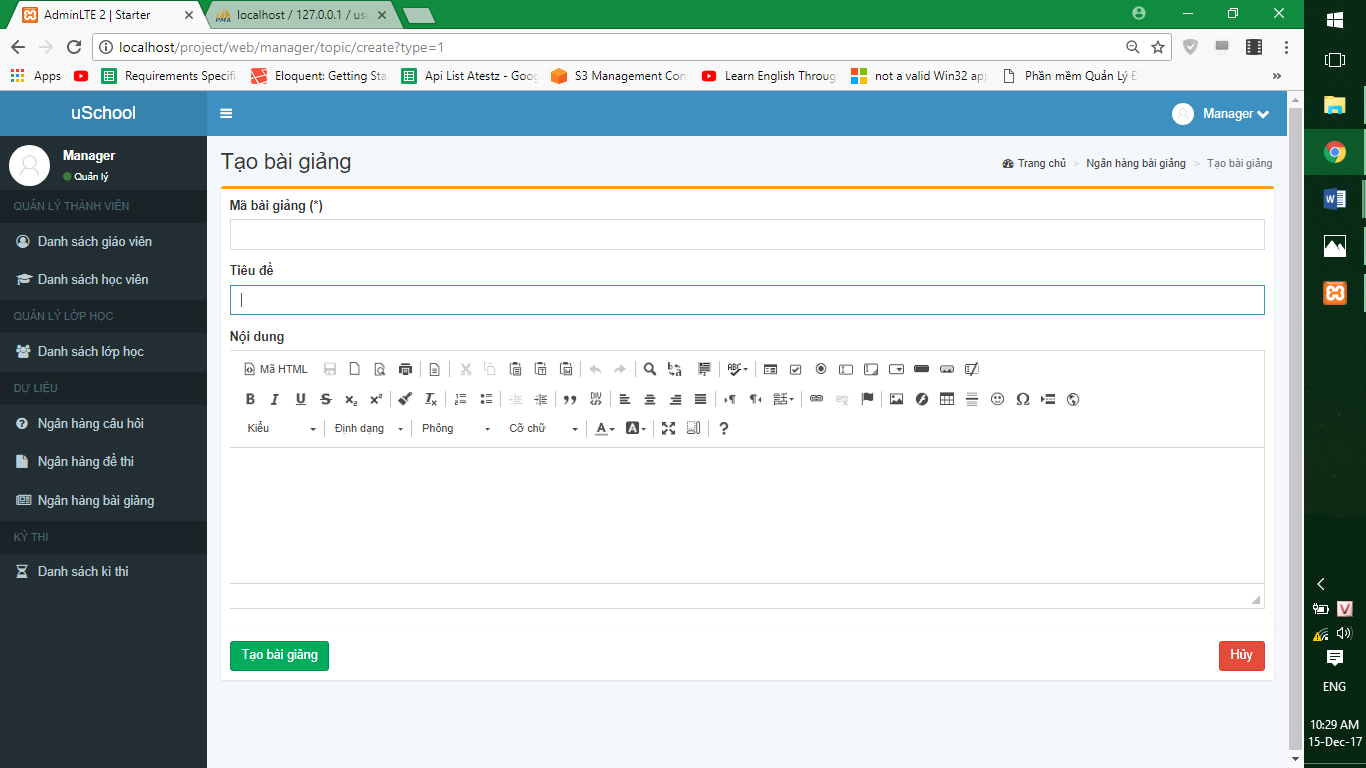
**Giao diện tạo đề thi TOEIC**



*Hình 37. Giao diện tạo đề thi TOEIC*

Hình 37 thể hiện giao diện đề thi TOEIC bao gồm các thông tin cơ bản khi tạo đề thi. Đề thi TOEIC có 7 part nên người dùng có thể chọn các câu hỏi cho từng part tương ứng.

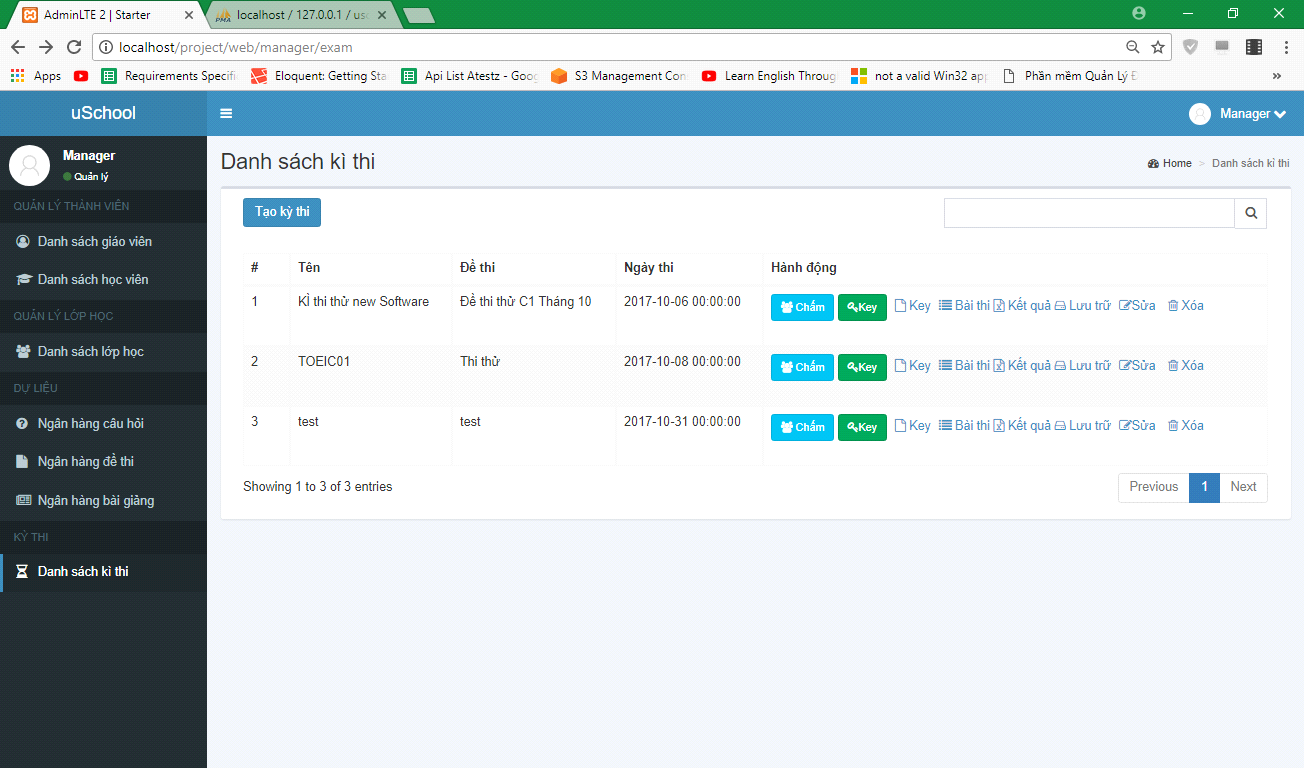
**Giao diện tạo bài giảng**



*Hình 38. Giao diện tạo bài giảng*

Hình 38. Giao diện tạo bài giảngHình 38 thể hiện giao diện tạo chủ đề bài giảng cho trường học. Phần nội dung bài giảng, người dùng có thể nhập nội dung chính liên quan đến chủ đề với gần đủ các tính năng soạn thảo văn bản như office.

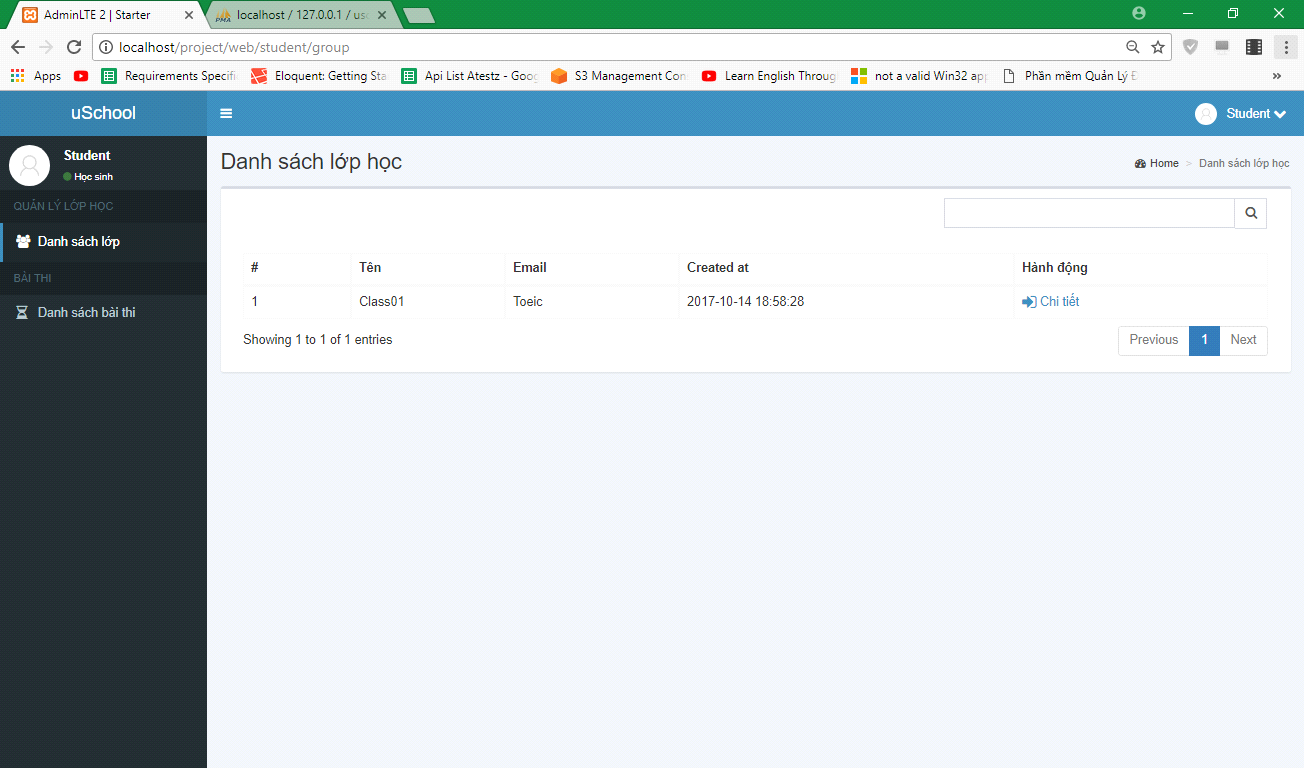
**Giao diện quản lí kì thi**.



*Hình 39. Giao diện quản lí kì thi*

Hình 39 thể hiện giao diện quản lí danh sách kì thi với các tính năng như sinh key cho kì thi, xem sanh sách các bài thi, export kết quả kì thi ra các file bản mềm để lưu trữ.

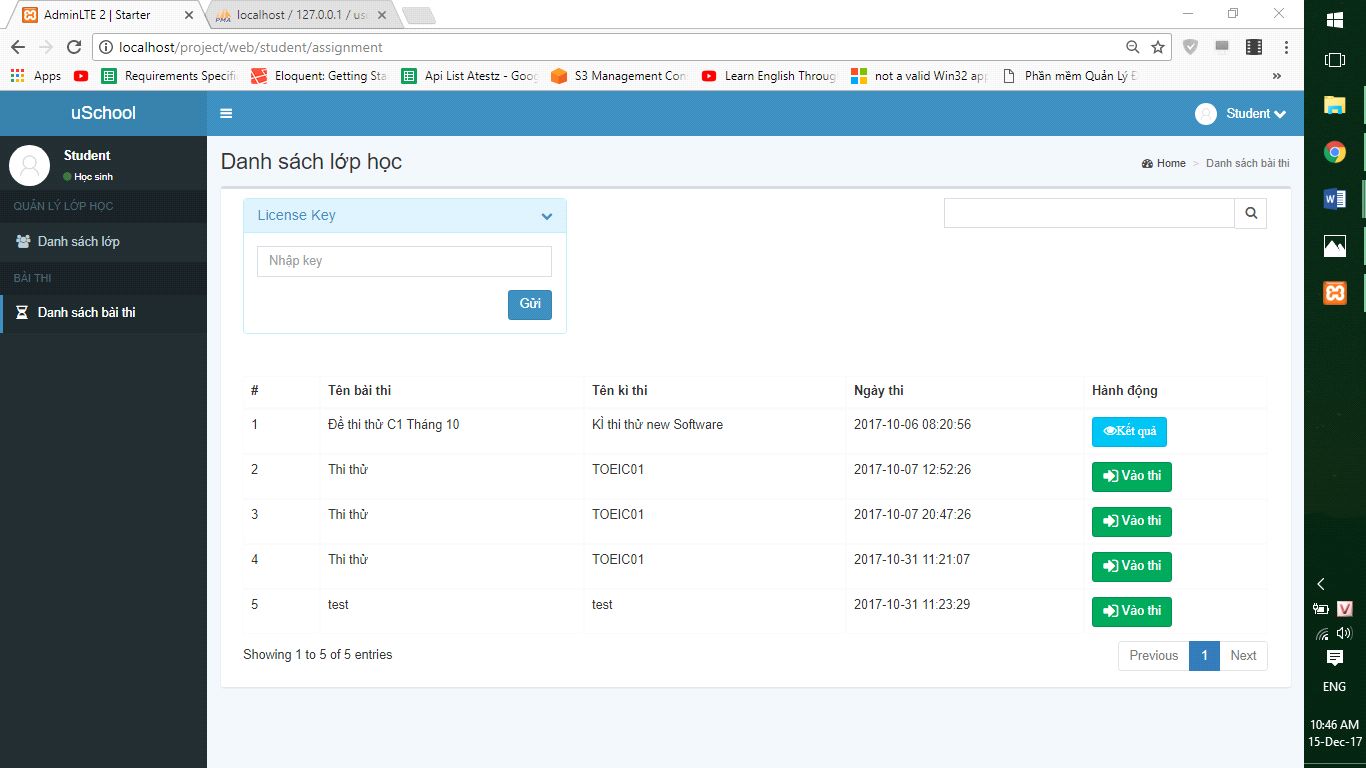
**Giao diện trang chủ phía học viên**



*Hình 40. Giao diện trang chủ phía học viên*

Hình 40 thể hiện giao diện trang chủ phía học viên với các menu bên trái bao gồm danh sách lớp học và danh sách bài thi.

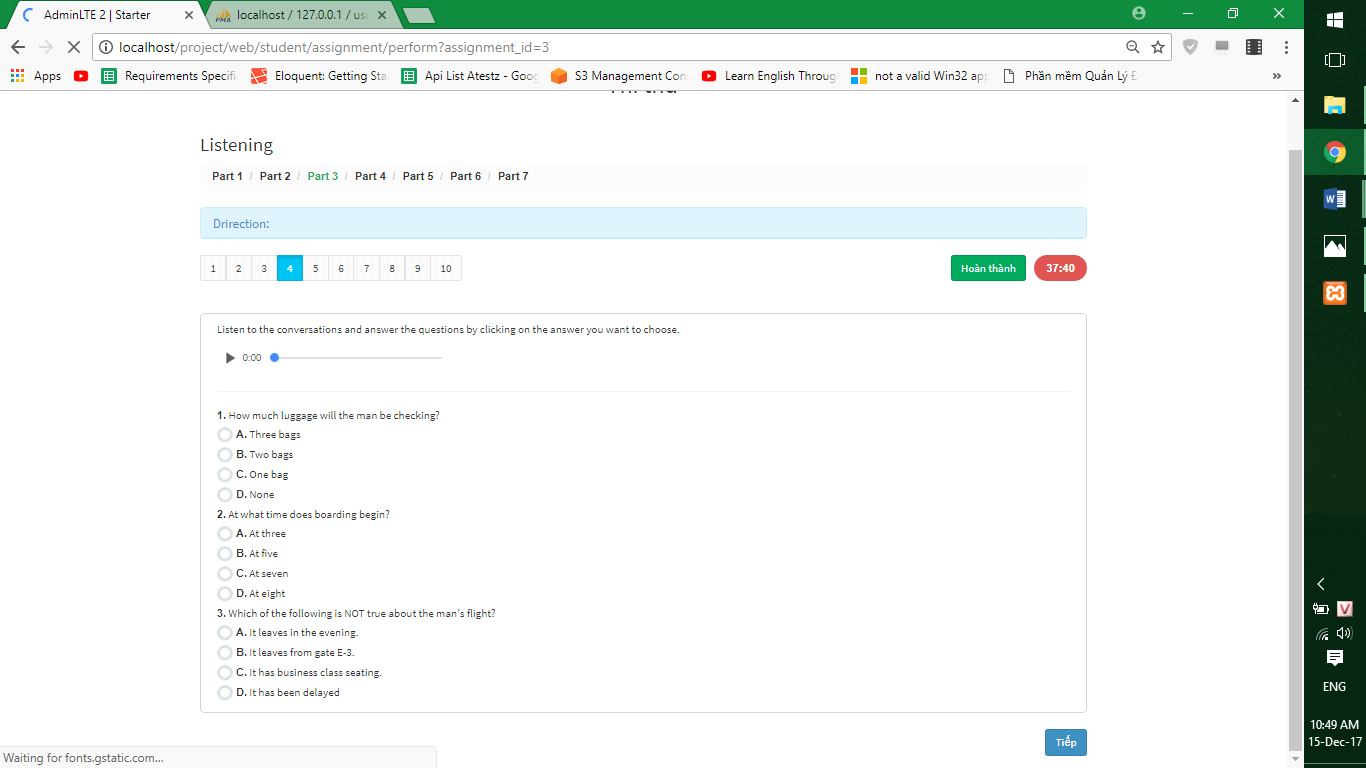
**Giao diện danh sách bài thi phía học viên**



*Hình 41. Giao diện danh sách bài thi phía học viên*

Hình 41 thể hiện giao diện trang chủ phía học viên. Phân nội dung có form cho phép nhập key để lấy đề thi về làm.

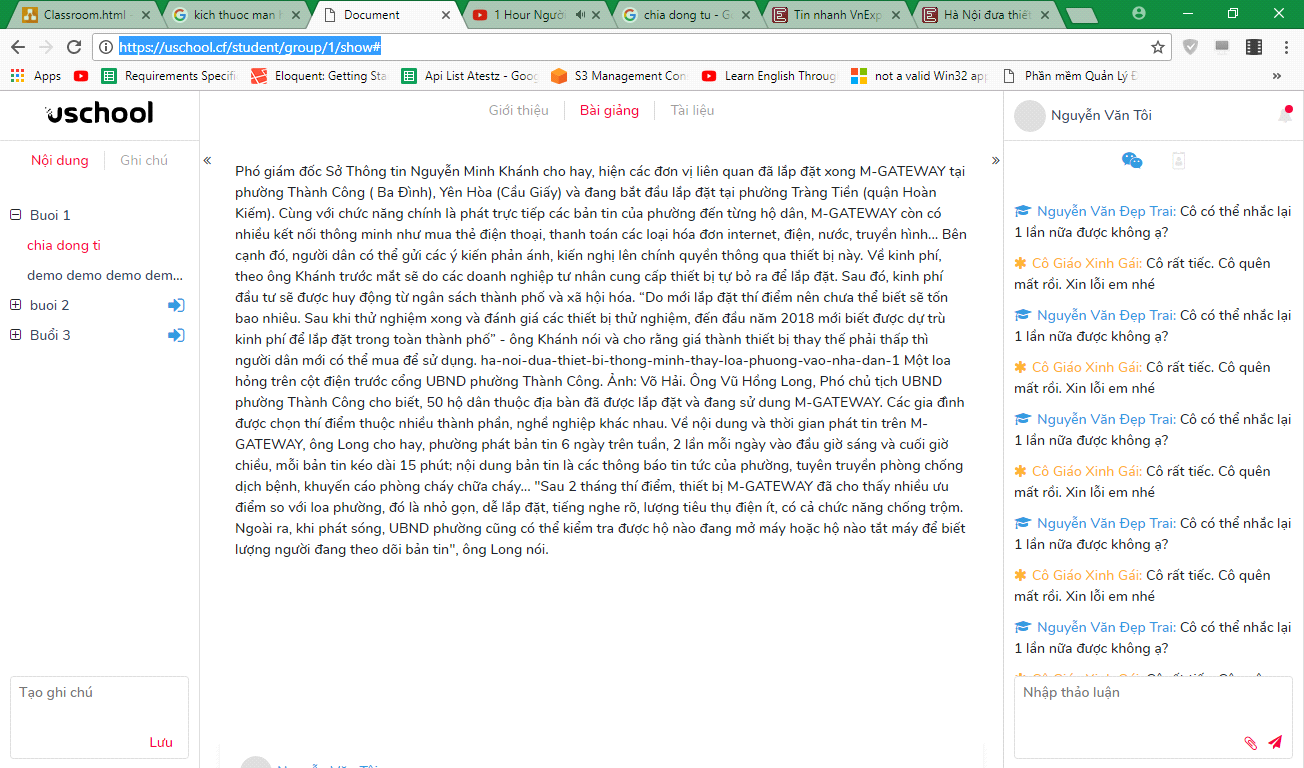
**Giao diện thi phía học viên**



*Hình 42. Giao diện thi phía học viên*

Hình 42 thể hiện giao diện hiển thị phần thi của học viên. Giao diện bao gồm phần hiển thị bộ đếm thời gian, danh sách các phần thi, danh sách các câu hỏi, nội dung các câu hỏi.

**Giao diện vào khóa học phía học viên**



*Hình 43. Giao diện vào khóa học phía học viên*

Hình 43 thể hiện giao diện của học viên khi vào khóa học. Phía bên trái là danh sách các buổi học hay bài giảng được tạo sẵn cho khóa học, học viên có thể chọn buổi học bất kì đề học. Nội dung của buổi học được hiển thị ở phía trung tâm dưới các dạng text, audio, video. Ngoài ra giao diện còn có thêm bình luận và hiển thị nội dung các bình luận trước đó.

**Giao diện lớp học**



*Hình 44. Giao diện lớp học*

Hình 44 thể hiện giao diện lớp học với 3 thành viên trong lớp học. Giao diện gồm video call, slide bài giảng lớp học, form chat, và các phím chức năng trong lớp học.

* **Đánh giá hệ thống**

Sau khi chạy thử nghiệm chương trình em có đánh giá như sau

**Ưu điểm:**

* Giao diện rõ ràng, dễ sử dụng
* Cơ bản hệ thống đáp ứng đầy đử các chức năng theo yêu cầu
* Các chức năng chạy ổn định

**Nhược điểm:**

* Giao diện phía học viên còn đơn giản, chưa đáp ứng được tính quảng bá sản phẩm
* Chưa có nền tảng di động
* Chưa có công cụ tạo nội dung bài giảng đầy đủ
* Lớp học chỉ chứa tối đa 4 người do phụ thuộc vào khả năng cấu hình mày và tốc độ mạng
* Chức năng phòng học ảo không chạy được trên một số trình duyệt không hỗ trợ webRTC như safari, IE v.v.

**CHƯƠNG 6: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN**

|  |
| --- |
| ***Chương cuối cùng em xin trình bày về các vấn đề sau: Thứ nhất tổng kết về quá trình làm đồ án, cuối cùng là hướng phát triển trong tương lai của hệ thống.*** |

* **6.1. Tổng kết**

Trong thời gian làm đồ án tốt nghiệp vừa qua em đã thực hiện đề tài và đã đạt được kết quả nhất định, tuy còn ở mức sơ sài nhưng đó là thành quả làm việc của bản thân với sự hướng dẫn, chỉ bảo tận tình của thầy Trịnh Tuấn Đạt

Trong phần cuối cùng của báo cáo này em xin tổng hợp về sản phẩm cũng như kết quả cuối cùng của đồ án. Chương trình mới chỉ minh họa được một phần trong cả quá trình xây dựng một hệ thống đào tạo trực tuyến. Sau đây em xin trình bày lợi ích hệ thống đem lại, ưu điểm, nhược điểm

**Lợi ích hệ thống**

Hệ thống có thể đem lại một số lợi ích đáng kể đối với xã hội, đóng góp một phần cho sự nghiệp giáo dục trong thời đại công nghệ 4.0.

* Đáp ứng nhu cầu học tập của nhiều học sinh, sinh viện mọi lúc mọi nơi
* Đem lại sự linh hoạt về thời gian, địa điểm cho người học,
* Có thể học tập và giao tiếp với mọi người mọi nơi trên thế giới
* Giảm chi phí cho việc học tập mà vẫn đảm bảo tính hiệu quả, và chất lượng giáo dục

**Ưu điểm**

* Hệ thống đã được xây dựng tương đối hoàn chỉnh các chức năng chính
* Giao diệndễ sử dụng, rõ ràng

**Nhược điểm**

* Do thời gian làm đồ án bị hạn chế nên hệ thống còn thiếu khá nhiều chức năng đối với 1 hệ thống đào tạo trực tuyến.Để có một hệ thống hoàn chỉnh thì sẽ cần hoàn thiện thêm nhiều chức năng khác mà em sẽ trình bày trong hướng phát triển.
* Do WebRTC ra đời khá muộn lại chưa được sử dụng rỗng rãi mới chỉ được hỗ trợ trên các trình duyệt phổ biến như Chrome, Firefox, Opera, cộng đồng hỗ trợ còn khá ít, nên tính phổ biến của ứng dụng còn hạn chế.
* **Hướng phát triển trong tương lai**

Trong tương lai em muốn hoàn thiện thêm hệ thống bằng việc xây dựng thêm các module hệ thống:

* Xây dựng thêm module marketing giúp quản bá trung tâm, giới thiệu các khóa học trên hệ thống. Hệ thống cho phép quản lí đưa ra các chương trình khuyến mại, tự động s lúc mọi nơi.

**TÀI LIỆU THAM KHẢO**

[1] Alan Forbes, A Beginner’s Guide, Createspace, 2012

[2] Kevin Yank, PHP & MySQL: Novice to Ninja, 6th Edition, SitePoint, 2017

[3] Lynn Beighley, Head First PHP & MySQL, Helion, 2010

[4] Francesco Malatesta, Learning Laravel's Eloquent, Packt, 2015

[5] Kelt Dockins, Design Patterns in PHP and Laravel, Apress, 2016

[6] Alan B. Johnston - Daniel C, WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web, 3th Edition, Digital Codex LLC, 2014

[7] Rob Manson, Getting Started with WebRTC, Packt, 2013

[8] Andrii Sergiienko, , 2014