|  |
| --- |
| NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN |
| **Giảng viên hướng dẫn**  *(Ký và ghi rõ họ tên)*  Nội dung nhận xét:   * **Đồng ý** hay **không đồng ý** cho sinh viên báo cáo TTCK; Nếu không đồng ý cần ghi rõ lý do. * Kết quả đạt được so với yêu cầu; * Ý kiến khác (nếu có) |
|  |

Lời cảm ơn

Em xin gởi lời cảm ơn chân thành và sự tri ân sâu sắc đối với các thầy cô của trường Đại học An Giang, đặc biệt là các thầy cô khoa Kỹ thuật - Công nghệ - Môi trường đã tạo điều kiện để em có thể hoàn thành tốt bài báo cáo này. Và em cũng xin chân thành cám ơn cô Nguyễn Minh Vi đã nhiệt tình hướng dẫn em trong suốt quá trình thực tập.

Do kinh nghiệm và kiến thức của em còn hạn chế nên không tránh khỏi sai sót. Em rất mong nhận được ý kiến của thầy, cô để em học thêm được nhiều kinh nghiệm làm hành trang sau khi ra trường.

Em xin chân thành cảm ơn!

Tóm Tắt

Trong sự phát triển của công nghệ thông tin, việc ứng dụng tin học vào quản lý trong giáo dục là xu thế tất yếu, đặc biết là các ngành giáo dục liên quan đến công nghệ. Đó là động lực để tôi khởi đầu dự án Xây dựng trang web quản lý tài liệu tham khảo cho khoa Kỹ thuật – Công nghệ – Môi trường.

Thông tin về khoa Kỹ thuật – Công nghệ – Môi trường và các lĩnh vực giáo dục sẽ được trình bày ở chương 1. Tiếp theo đó, điều kiện khách quan dẫn đến nhu cầu xây dựng hệ thống quản lý tài liệu tham khảo, cách thức tổ chức hệ thống, phạm vi hệ thống.. các tiền đề lý thuyết để xây dựng hệ thống sẽ lần lượt được trình bày ở chương 2. Đến chương 3, các kết quả khảo sát, thu thập yêu cầu thực tế sẽ được phân tích và mô hình hóa để tạo nền tảng cho việc xây dựng hệ thống. Sau đó các mô hình trên sẽ được cụ thể bằng dữ dữ liệu, cũng như các xử lí được trình bày tại chương 4. Những tồn động, vướn mắc trong quá trình xây dựng cũng như những giải pháp cho tương lai sẽ xuất hiện trong phần Kết luận và hướng phát triển.

Tuy hệ thống còn nhiều thiếu sót. Song song với quá trình sử dụng, hệ thống sẽ luôn luôn được bảo trì nâng cấp các tính năng đảm bảo mọi yêu cầu từ phía người dùng cũng như người quản lí website.

Chương 1

GIỚI THIỆU VỀ CƠ QUAN THỰC TẬP

1.1. Giới thiệu tổng quan về cơ quan thực tập

Trường Đại học An Giang được thành lập theo quyết định số 241/1999/QĐ-TTg ngày 30 tháng 12 năm 1999 của Thủ tướng chính phủ và khai giảng niên học đầu tiên ngày 9 tháng 9 năm 2000. Xây dựng trên cơ sở trường Cao đẳng Sư phạm An Giang, Trường Đại học An Giang là trường đại học công lập trong hệ thống các trường đại học Việt Nam, trực thuộc Ủy ban nhân dân tỉnh An Giang. Trường có nhiệm vụ đào tạo cán bộ đại học và các trình độ thấp hơn phục vụ nhu cầu phát triển kinh tế xã hội của tỉnh An Giang, Đồng Tháp, Kiên Giang và các tỉnh lân cận, đồng thời trường có nhiệm vụ nghiên cứu, ứng dụng và chuyển giao công nghệ trong vùng.

Khoa Kỹ thuật – Công nghệ – Môi trường thuộc trường ĐH An Giang được thành lập theo Quyết định số 241/QĐ-TC của UBND tỉnh An Giang ký ngày 21 tháng 02 năm 2001. Khoa gồm 3 bộ môn: Bộ môn Tin học, Bộ môn Kỹ thuật và Bộ môn Môi trường và Phát triển bền vững. Khoa có chức năng đào tạo các ngành học về công nghệ thông tin, môi trường, các ngành kỹ thuật, nghiên cứu, chuyển giao các ứng dụng về công nghệ thông tin và kỹ thuật môi trường nhằm đáp ứng nhu cầu phát triển kinh tế của tỉnh An Giang và khu vực Đồng bằng sông Cửu Long. Hiện Khoa có các ngành đào tạo sau:

**Ngành Công nghệ Thông tin:**

Đào tạo Cử nhân Công nghệ thông tin có kiến thức cơ bản, cơ sở và chuyên ngành thuộc lĩnh vực công nghệ thông tin; có năng lực tham mưu, tư vấn và khả năng tổ chức thực hiện nhiệm vụ với tư cách của một chuyên viên trong lĩnh vực công nghệ thông tin. Trong quá trình học, sinh viên được trang bị kiến thức về phân tích, thiết kế và quản lý các dự án công nghệ thông tin; có khả năng xây dựng, bảo trì và quản lý các hệ thống mạng máy tính và truyền thông.

**Ngành Kỹ thuật phần mềm:**

Đào tạo Cử nhân Kỹ thuật phần mềm có khả năng thiết kế các phần mềm theo đúng qui trình sản xuất phần mềm; có năng lực tham mưu, tư vấn và khả năng tổ chức thực hiện nhiệm vụ của một chuyên viên trong lĩnh vực kỹ thuật phần mềm. Trong quá trình học, sinh viên được trang bị kiến thức về quy trình xây dựng, quản lý và bảo trì hệ thống phần mềm.

**Ngành Công nghệ Kỹ thuật Môi trường:**

Chương trình đào tạo ngành Công nghệ Kỹ thuật Môi trường ở Đại học An Giang được thiết kế để đào tạo kỹ sư vững vàng về cơ sở lý thuyết, có khả năng nghiên cứu, thiết kế – thi công – vận hành hệ thống xử lý và kiểm soát ô nhiễm môi trường bao gồm cả môi trường đất, nước và khí. Chương trình được cấu trúc theo phương thức đáp ứng những nhu cầu cấp bách của môi trường, nhấn mạnh trong công nghệ xử lý nước cấp và nước thải, công nghệ xử lý khí thải, công nghệ xử lý chất thải rắn tại các khu dân cư và công nghiệp nhằm đảm bảo an toàn vệ sinh và sức khỏe môi trường. Ở Đại học An Giang, sinh viên được đào tạo theo phương châm “đào tạo gắn liền với thực tiễn”.

**Ngành Quản lý Tài nguyên và Môi trường:**

Cung cấp cho sinh viên kiến thức chung về khoa học tự nhiên, khoa học xã hội, ngoại ngữ, khối kiến thức chuyên sâu và đa dạng về lĩnh vực tài nguyên và môi trường như: quản lý đất đai, tài nguyên khoáng sản, tài nguyên nước, tài nguyên rừng và đa dạng sinh học, tài nguyên biển; quản lý hệ thống xử lý ô nhiễm môi trường cho các nhà máy, khu công nghiệp và khu dân cư; kiểm tra giám sát và dự báo ô nhiễm; tư vấn môi trường và công nghệ sạch; đánh giá tác động môi trường cho các dự án đầu tư và phát triển. Ngoài ra sinh viên còn trang bị nhiều kỹ năng: quản lý, nghiên cứu và thực hiện các chương trình dự án về bảo vệ môi trường.

**Ngành Cao đẳng Công nghệ Thông tin:**

Đào tạo Cử nhân Cao đẳng Công nghệ thông tin có kiến thức nền tảng trong lĩnh vực công nghệ thông tin; có năng lực tư vấn và khả năng tổ chức thực hiện nhiệm vụ của một chuyên viên trong lĩnh vực công nghệ thông tin. Trong quá trình học, sinh viên được trang bị kiến thức về bảo trì, quản lý các hệ thống thông tin và mạng máy tính.

Hoạt động kinh doanh của khoa tập trung vào đào tạo nguồn nhân lực chất lượng cao cho nhu cầu phát triển kinh tế tỉnh An Giang và khu vực Đồng bằng sông Cửu Long.

Qua thời gian cùng sự tín nhiệm của học viên và các đối tác, ĐH An Giang đã trở thành điểm đến quen thuộc của của các bạn học sinh An Giang và khu vực Đồng bằng sông Cửu Long. Đáp lại sự tín nhiệm của học viên và các đối tác, toàn bộ giảng viên luôn đặt “chất lượng đào tạo” làm phương châm hoạt động.

Dưới đây là thông tin chi tiết khoa:

- Tên khoa: Kỹ thuật – Công nghệ – Môi trường.

- Người đại diện:

- Địa chỉ khoa: Dãy Nhà D, Khu Trung tâm, Đại học An Giang, số 18 Ung Văn Khiêm, phường Đông Xuyên, thành phố Long Xuyên, tỉnh An Giang.

- Điện thoại: 0766 516 132 – 0766 256565

- Website: tech.agu.edu.vn

- Ngành nghề: Giáo dục.

1.2. Lĩnh vực hoạt động

**Giáo dục:**

- Các ngành đào tạo:

* + - **Công nghệ Thông tin**
    - **Kỹ thuật phần mềm**
    - **Công nghệ Kỹ thuật Môi trường**
    - **Quản lý Tài nguyên và Môi trường**
    - **Cao đẳng Công nghệ Thông tin**

1.3. Tầm nhìn – sứ mệnh

1.3.1. Tầm nhìn

Trở thành trung tâm hàng đầu tỉnh An Giang và khu vực Đồng bằng sông Cửu Long về đào tạo nguồn nhân lực chất lượng cao phục vụ các ngành công nghệ và kỹ thuật.

1.3.2. Sứ mệnh

Khoa Kỹ thuật – Công nghệ – Môi trường lấy học viên làm trung tâm, lấy tổ chức chuyên nghiệp làm sức mạnh, không ngừng nổ lực nâng cao chất lượng đào tạo cung cấp cho các học viên kiến thức vững chắc cho công việc trong tương lai.

Chương 2

TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

2.1. Lý do chọn đề tài

Cùng với sự bùng nổ của công nghệ cũng như Internet,việc áp dụng công nghệ thông tin vào trường học đã trở nên phổ biến. Việc áp dụng công nghệ thông tin vào quản lý sẽ giúp công việc trở nên thuận tiện hơn.

Hiện tại việc mượn trả tài liệu hoàn toàn thủ công, sinh viên phải liên hệ với khoa tìm kiếm tài liệu để mượn mà không biết tài liệu có hay không, đã được mượn hay chưa. Việc có nhiều sinh đến mượn cùng một thời điểm người quản lý phải trả cứu tuần từ cho từng sinh viên, người quản lý phải tra xem sinh viên muốn mượn tài liệu đã vượt quá số tài liệu được mượn hay không làm mất nhiều thời gian. Người quản lý phải thường xuyên kiểm tra xem có tài liệu đã hết hạn trả hay chưa, việc quản lý lượng lớn tai liệu gây nhiều khó khăn.

Đó chính là động lực để tôi xây dựng trang web quản lý tài liệu cho khoa.

2.2. Tổng quan hệ thống

Hệ thống quản lý tài liệu trực tuyến được xây dựng nhằm đưa thông tin chi tiết về những tài liệu mà khoa đang có đến sinh viên, và nó cũng là một kênh giao tiếp của khoa đến sinh viên.

Hệ thống phục vụ cho các đối tượng sau:

* Admin: người có toàn quyền trong việc quản trị hệ thống. Các quyền đó gồm cập nhật danh mục, tài liệu,tài khoản...
* Khách: khách vãng lai ghé thăm có thể xem thông tin chi tiết về các tài liệu, tìm được tài liệu cần thiết, kiểm tra tình trạng các tài liệu.
* Sinh viên: ghé thăm có thể xem thông tin chi tiết về các tài liệu, tìm được tài liệu cần thiết, kiểm tra tình trạng các tài liệu. Sinh viên có thể đăng ký mượn tài liệu và thời gian nhận.

2.3. Đối tượng và phạm vi hệ thống

Bảng 1: Mô tả đối tượng và phạm vi ứng dụng

|  |  |
| --- | --- |
| Tên dự án | Xây dựng trang web Quản lý tài liệu |
| Người thiết kế và xây dựng dự án | Nguyễn Văn Thạnh – MSSV: DPM134988 |
| Phạm vi ứng dụng | - Xây dựng hệ thống mượn trả tài liệu cho khoa |
| Mục tiêu | * - Giao diện thân thiện, dễ sử dụng. * - Hệ thống tối ưu về mặt dữ liệu và xử lí, giúp phản hồi nhanh chóng. Phải đảm bảo tính sẵn sáng sử dụng. * - Website hỗ trợ tốt cho cả thiết bị máy tính PC lẫn điện thoại di động. * - Không vượt quá chi phí và thời gian. * - Hệ thống phải có độ bảo mật cao. * - Hệ thống phải chính xác về mặt xử lí dữ liệu. * - Hệ thống chạy độc lập, không phụ thuộc vào các hệ thống khác. |
| Lợi ích mang lại | - Sinh viên dễ dàng tìm kiếm thông tin tài liệu, dễ dàng liên hệ với khoa. |
| Các bước xây dựng và thực hiện | * - Khởi tạo dự án. * - Lập kế hoạch. * - Thu thập yêu cầu. * - Phân tích yêu cầu và thiết kế hệ thống. * - Cài đặt hệ thống. * - Kiểm thử. * - Sửa lỗi (có thể quay lại bước Cài đặt hệ thống). * - Hướng dẫn sử dụng. * - Bảo trì (tương lai). |

2.4. Cơ sở lý thuyết

2.4.1. Ngôn ngữ JSP/Java Servlet

**Java Servlet API** cho phép nhà phát triển phần mềm thêm những nội dung động vào Web server sử dụng Java platform. Nó phát sinh nội dung thông thường là HTML, đôi khi cũng có thể là các ngôn ngữ khác như XML. Servlets là bản sao Java của các kỹ thuật trình bày nội dung Web động khác như là PHP, CGI và ASP.NET. Servlets có thể duy trì tình trạng thông qua các giao dịch của máy chủ bằng việc sử dụng HTTP cookies, các giá trị session hoặc URL rewriting. Servlet API, được chứa trong gói Java kế thừa từ Bản mẫu:Javadoc:EE, định nghĩa những tương tác cần thiết giữa một Web container và một servlet. A Web container là một thành phần cơ bản và cần thiết của Web Server để tương tác với các Servlet. The Web container chịu trách nhiệm quản lý vòng đời của các Servlet, ánh xạ đường dẫn đến một Servlet cụ thể và đảm bảo rằng có thể truy cập vào.

**JSP (Java Server Pages)** là một công nghệ cho phép thực hiện dễ dàng việc viết các trang web động (dĩ nhiên ta cũng có thể viết các trang web tĩnh với JSP). Cụ thể, một trang JSP là một trang HTML (hay XML) trong đó có trộn lẫn các mã Java, tức các thành phần JSP, cho phép thực hiện nội dung động. Các trang JSP có phần mở rộng .jsp. Một trang jsp có thể là một bao gồm nhiều tập tin, mà các tập tin này có thể là một trang jsp hoàn chỉnh hay những đoạn (fragment) của một trang jsp. Những đoạn jsp này có phần mở rộng là .jspf. Như ta đã biết, việc viết nội dung html trả lời trong servlet bằng cách dùng các lệnh out.println() là một việc tốn nhiều công sức, đặc biệt khi nội dung html trả lời càng phức tạp. Việc này còn tệ hơn khi ta muốn bảo trì hay thay đổi nội dung trả lời. JSP đem lại một giải pháp tiện lợi hơn khi cho phép chèn nội dung động vào trang web, nó còn cho phép việc tách bạch phần trình bày và logic xử lý nội dung. Với servlet thì ta chèn mã html để phát sinh trong lớp java còn JSP thì ngược lại, ta chèn mã java trong trang html.

**Đặc điểm:**

**-** Thư viện phong phú.

- Hỗ trợ nhiều hệ quản trị Cơ sở dữ liệu.

- Nền tảng là Java – một ngôn ngữ thuần hướng đối tượng nên JSP/Servlet có hầu hết các điểm mạnh của ngôn ngữ hướng đối tượng như trừu tượng hóa, đóng gói, đa hình và kế thừa.

- Cung cấp nhiều cơ chế bảo mật cho ứng dụng.

2.4.2. Hệ quản trị cơ sở dữ liệu MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở 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 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, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,... 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).

Cấu trúc của một cơ sở dữ liệu: CSDL, data base (gọi tắt là base) là một cấu trúc chứa nhiều quan hệ (gọi là một table). Một table chứa các loại dữ liệu khác nhau, trong đó các cột gọi là các trường (fields) còn các hàng gọi là row hay record.

**Các đặc điểm của MySQL:**

- Tốc độ: MySQL rất nhanh. Những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất có thể có.

- Dễ sử dụng: MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn .

- Giá thành: MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức.

- Hỗ trợ ngôn ngữ truy vấn: MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity -một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).

- Năng lực: Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Có thể truy cập MySQL tương tác với sử dụng một vài giao diện để có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web…

- Kết nối và bảo mật: MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó có thể chia sẽ dữ liệu với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người nào không nên nhìn thấy dữ liệu thì không thể nhìn được.

- Tính linh động: MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.

- Sự phân phối rộng: MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web. Nếu không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, thì có thể lấy mã nguồn và tìm tòi nó. Nếu không thích một vài thứ thì có thể thay đổi nó.

- Sự hỗ trợ: giờ đây có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.

2.4.3. CSS

CSS là chữ viết tắt của Cascading Style Sheets, nó chỉ đơn thuần là một dạng file text với phần tên mở rộng là .css. Trong Style Sheet này chứa những câu lệnh CSS. Mỗi một lệnh của CSS sẽ định dạng một phần nhất định của HTML ví dụ như: font của chữ, đường viền, màu nền, căn chỉnh hình ảnh... Trước đây khi chưa có CSS, những người thiết kế web phải trộn lẫn giữa các thành phần trình bày và nội dung với nhau. Nhưng với sự xuất hiện của CSS, người ta có thể tách rời hoàn toàn phần trình bày và nội dung. Giúp cho phần code của trang web cũng gọn hơn và quan trọng hơn cả là dễ chỉnh sửa hơn.

Có ba cách để định dạng trang web là: cục bộ, nhúng vào trang và liên kết đến một file CSS riêng biệt. Trong thực tế thì cách cuối cùng là liên kết đến một file riêng biệt được sử dụng phổ biến nhất.

2.4.4. JavaScript

JavaScript, theo phiên bản hiện hành, là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Nó vốn được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. .js là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript. Phiên bản mới nhất của JavaScript là phiên bản 1.5, tương ứng với ECMA-262 bản 3. ECMAScript là phiên bản chuẩn hóa của JavaScript. Trình duyệt Mozilla phiên bản 1.8 beta 1 có hỗ trợ không đầy đủ cho E4X - phần mở rộng cho JavaScript hỗ trợ làm việc với XML, được chuẩn hóa trong ECMA-357.

JavaScript là một ngôn ngữ lập trình dựa trên nguyên mẫu với cú pháp phát triển từ C. Giống như C, JavaScript có khái niệm từ khóa, do đó, JavaScript gần như không thể được mở rộng. Cũng giống như C, JavaScript không có bộ xử lý xuất/nhập (input/output) riêng. Trong khi C sử dụng thư viện xuất/nhập chuẩn, JavaScript dựa vào phần mềm ngôn ngữ được gắn vào để thực hiện xuất/nhập. Trên trình duyệt, rất nhiều trang web sử dụng JavaScript để thiết kế trang web động và một số hiệu ứng hình ảnh thông qua DOM. JavaScript được dùng để thực hiện một số tác vụ không thể thực hiện được với chỉ HTML như kiểm tra thông tin nhập vào, tự động thay đổi hình ảnh,... Ở Việt Nam, JavaScript còn được ứng dụng để làm bộ gõ tiếng Việt giống như bộ gõ hiện đang sử dụng trên trang Wikipedia tiếng Việt. Tuy nhiên, mỗi trình duyệt áp dụng JavaScript khác nhau và không tuân theo chuẩn W3C DOM, do đó trong rất nhiều trường hợp lập trình viên phải viết nhiều phiên bản của cùng một đoạn mã nguồn để có thể hoạt động trên nhiều trình duyệt. Một số công nghệ nổi bật dòng JavaScript để tương tác với DOM bao gồm DHTML, Ajax và SPA. Bên ngoài trình duyệt, JavaScript có thể được sử dụng trong tập tin PDF của Adobe Acrobat và Adobe Reader. Điều khiển Dashboard trên hệ điều hành Mac OS X phiên bản 10.4 cũng có sử dụng JavaScript. Công nghệ kịch bản linh động (active scripting) của Microsoft có hỗ trợ ngôn ngữ JScript làm một ngôn ngữ kịch bản dùng cho hệ điều hành. JScript.NET là một ngôn ngữ tương thích với CLI gần giống JScript nhưng có thêm nhiều tính năng lập trình hướng đối tượng. Mỗi ứng dụng này đều cung cấp mô hình đối tượng riêng cho phép tương tác với môi trường chủ, với phần lõi là ngôn ngữ lập trình JavaScript gần như giống nhau.

**Đặc điểm của javaScript:**

- Javascript có thể xử lý các mã HTML. Javascript nó được đính kèm và thực thi trong các tài tài liệu HTML. Hầu hết các đối tượng Javascript đều liên quan đến các thẻ HTML (HTML tags). Javascript dựa vào HTML để giúp các ứng dụng web hoạt động

- Javascript phụ thuộc môi trường hoạt động. Javascript là một ngôn ngữ kịch bản, một chương trình chạy trực tiếp trên trình duyệt web (Firefox, Opera, Netscape Navigator, Internet Explorer, Safari, etc..). Việc quan tâm đến môi trường hoạt động của Javascript rất quan trọng vì các ứng dụng Javascript sẽ hoạt động khác nhau trên các trình duyệt khác nhau.

- Javascript hoàn toàn là một ngôn ngữ biên dịch – nó được trình duyệt biên dịch và thực thi. Nó không cần một trình biên dịch tiền xử lý.

- Javascript là một ngôn ngữ linh hoạt. Trong Javascript ta có thể khai báo nhiều kiểu biến, thực thi các chương trình với các biến không xác định.

- Javascript dựa trên các đối tượng – Javascript là một ngôn ngữ lập trình hướng đối tượng giống như Java. Thực ra phải nói chính xác là “dựa trên các đối tượng” vì các đối tượng của Javascript đáp ứng tức thời và không có tính kế thừa.

- Javascript được điều khiển bởi các sự kiện – Hầu hết các đoạn code Javascript đáp ứng các sự kiện do người dùng hoặc hệ thống gây nên. Các đối tượng HTML thường dùng để tăng cường hỗ trợ các sự kiện.

- JavaScript không phải là Java - hai ngôn ngữ được tạo ra bởi các công ty khác nhau. Mặc dù sự trùng tên không phải ngẫu nhiên mà vì lý do tiếp thị.

- Javascript là một ngôn ngữ đa dụng. Ngôn ngữ lập trình này được dùng trong các trường hợp để giải quyết nhiều vấn đề khác nhau như: toán học, đồ họa…

- Javascript đang được phát triển không ngừng. Đây là một điều tốt nhưng nó cũng gây ra vấn đề đối với các lập trình viên, họ phải luôn luôn cập nhật kiến thức để tạo ra các ứng dụng phù hợp cho người dùng trên các trình duyệt khác nhau.

- Javascript bao hàm rất nhiều lĩnh vực. Mặc dù Javascript được tạo ra chủ yếu dành cho máy khách (client) nhưng nó cũng được sử dụng ở các server nữa. Javascript ngôn ngữ tự nhiên của các công cụ phát triển web như Macromedia Dreamweaver hay IntraBuilder Borland.

2.5. Đánh giá mức độ khả thi và mức độ rủi ro

2.5.1. Mức độ khả thi

2.5.1.1. Khả thi về hoạt động

Hệ thống được xây dựng với giao diện thân thiện, dễ dàng sử dụng, không cần có chuyên môn cao cũng có thể quản lí website.

2.5.1.2. Khả thi về kĩ thuật

Đối với khoa: không yêu cầu phần cứng cao, không cần đường truyền internet băng thông rộng, hỗ trợ cả máy tính PC lẫn smartphone. Chi phí xây dựng bảo trì thấp.

Đối với người thực hiện: Sử dụng mã nguồn mở (JSP/Java-Servlet, html, css..), công cụ lập trình miễn phí (eclipse), hệ quản trị cơ sở dữ liệu và các công cụ phân tích được phát hành miễn phí (MySQL, StarUML..).

2.5.2. Mức độ rủi ro

Bảng 2: Mô tả mức độ rủi ro của dự án

|  |  |  |  |
| --- | --- | --- | --- |
| Dự án: Quản lý tài liệu tham khảo | | | |
| Rủi ro | Tỉ lệ | Mô tả | Hướng khắc phục |
| Sự thay đổi yêu cầu từ phía cửa hàng | Thấp | Các yêu cầu thay đổi từ phía cửa hàng như thay đổi cách thức hoạt động, cách thức tổ chức dữ liệu, giao diện hiển thị.. | Thu thập yêu cầu một cách đầy đủ và chính xác |
| Thay đổi độ lớn của dự án | Rất thấp | Yêu cầu tăng/giảm độ lớn của dự án như thêm chức năng, thêm các ràng buộc. | Tùy theo mức độ, có thể yêu cầu tăng thời gian/chi phí để đáp ứng yêu cầu. |
| Trình độ tin học người dùng thấp | Cao | Người dùng không có nhiều kinh nghiệm tin học, khả năng quản trị website | Thiết kế giao diện thân thiện, dễ sử dụng. Viết các tài liệu hướng dẫn hoặc trực tiếp hướng dẫn. |
| Tấn công mạng | Thấp | Các đợt tấn công mạng từ sự cạnh tranh không lành mạnh hoặc các các nhân chống phá | Sử dụng các liệu pháp an toàn thông tin, các biện pháp phòng chống DDOS. |

Chương 3

PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Kết quả thu thập yêu cầu

3.1.1. Yêu cầu phi chức năng

- Trang web hoạt động liên tục.

- Phản hồi nhanh.

- Đáp ứng các chỉ tiêu: sẵn sàng sử dụng, bảo mật, toàn vẹn, chính xác, nhất quán.

- Menu đáp ứng quy tắc 3 lần click.

- Không cần máy tính cấu hình cao.

- Tương thích với cách trình duyệt thông dụng, tương thích tốt với trình duyệt trên điện thoại.

- Giao diện hài hòa, trang nhã, không vi phạm thuần phong mỹ tục Việt Nam.

3.1.2. Yêu cầu chức năng

- Trang web dành cho nhân viên cho phép quản lí các danh mục hàng hóa (điện thoại, phụ kiện), thêm bớt số lượng sản phẩm phù hợp với hiện trạng cửa hàng, quản lí đơn hàng, bảo hành, thống kê lượng sản phẩm bán ra, doanh thu... Đăng bài quảng cáo, giới thiệu sản phẩm..

- Trang web dành cho khách hàng cho phép xem các sản phẩm theo các thuộc tính (giá, loại, hãng sản xuất,...), xem chi tiết sản phẩm, đặt hàng, thanh toán online (optional), quản lí đơn hàng, quản lí tình trạng bảo hành. Cho phép người dùng đăng ký thành viên để dễ dàng quản lí đơn hàng hơn cũng như nhận được các khuyến mãi. Người dùng có thể gửi những thông tin liên hệ đến người quản trị.

3.2. Phân tích yêu cầu

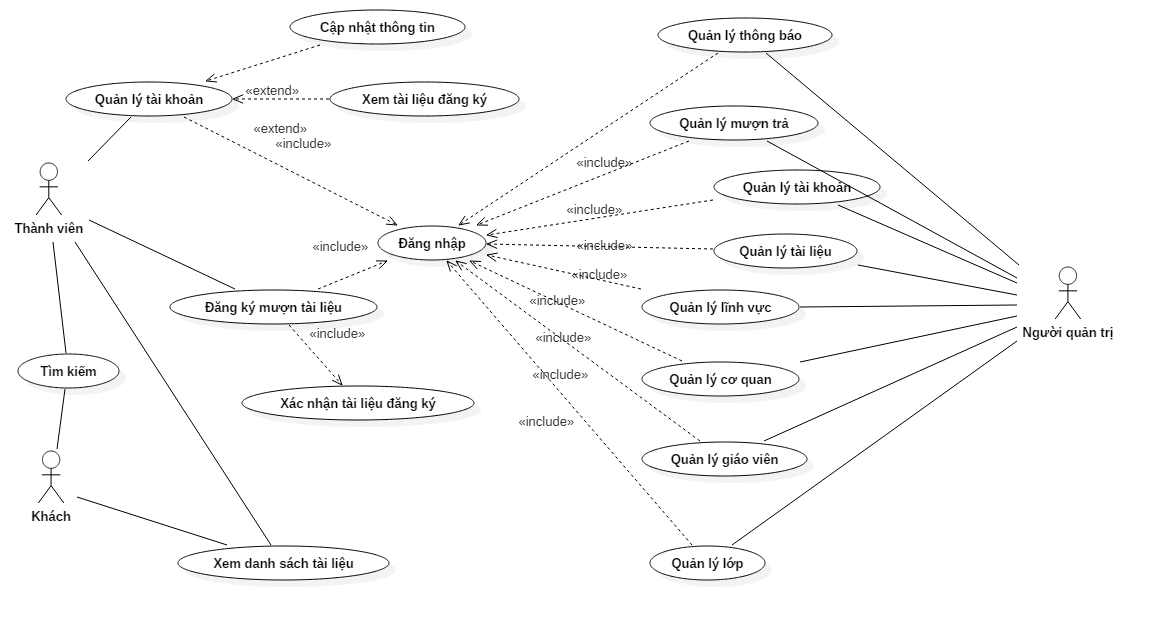
Từ các yêu cầu, qua phân tích ta có bảng sau:

Bảng 3: Bảng phân tích các chức năng chính của website

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên chức năng | Người dùng | Mô tả | Yêu cầu |
| 1 | Quản lí lĩnh vực | Người quản lí website | Thêm, xóa, chỉnh sửa các lĩnh vực tài liệu. | Đăng nhập với tài khoản quản trị. |
| 2 | Quản lí tài liệu | Người quản lí website | Thêm, xóa, chỉnh sửa các thuộc tính của tài liệu. | Đăng nhập với tài khoản quản trị. |
| 3 | Quản lí lớp | Người quản lí website | Thêm, xóa, chỉnh sửa lớp. | Đăng nhập với tài khoản quản trị. |
| 4 | Quản lí cơ quan thực tập | Người quản lí website | Thêm, xóa, chỉnh sửa cơ quan thực tập. | Đăng nhập với tài khoản quản trị. |
| 5 | Quản lí giáo viên hướng dẫn | Người quản lí website | Thêm, xóa, chỉnh sửa giáo viên hướng dẫn. | Đăng nhập với tài khoản quản trị. |
| 6 | Quản lí xét duyệt yêu cầu mượn tài liệu | Hệ thống, người quản lí website | Duyệt, hủy, nhận trả tài liệu. | Đăng nhập với tài khoản quản trị. |
| 7 | Quản lí người dùng | Người quản lí website | Thêm, xóa, sửa thông tin người dùng, cấp quyền người dùng. | Đăng nhập với tài khoản quản trị. |
| 8 | Quản lý thông báo | Người quản lí website | Thêm, xóa, sửa thông báo. | Đăng nhập với tài khoản quản trị. |
| 8 | Hiển thị thông tin | Mọi người | Hiển thị danh sách tài liệu | Tùy loại thông tin. |
| 9 | Đăng nhập | Thành viên | Đăng nhập vào hệ thống để sử dụng các chức năng dành riêng cho thành viên. | Cung cấp tên tài khoản, mật khẩu. |
| 10 | Kiểm tra yêu cầu mượn tài liệu | Thành viên | Kiểm tra thông tin yêu cầu mượn tài liệu (ngày dự kiến nhận, trạng thái yêu cầu.) | Cung cấp tên tài khoản, mật khẩu. |
| 11 | Đặt mượn tài liệu | Thành viên | Đặt mượn tài liệu, dự kiến thời gian nhận | Cung cấp thông tin tài liệu, thời gian nhận |
| 12 | Chỉnh sửa thông tin | Thành viên | Chỉnh sửa thông tin cá nhân | Đăng nhập, cung cấp thông tin cần sửa đổi |

3.3. Mô hình hóa hệ thống

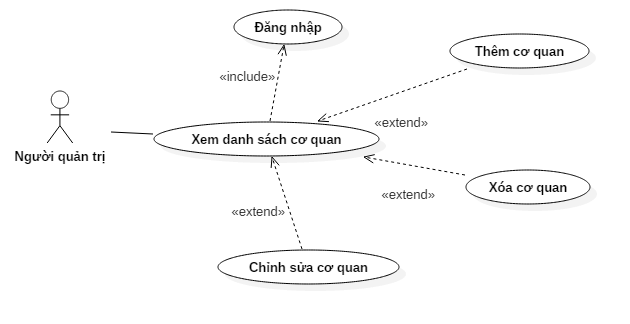
3.3.1. Các use-case

3.3.1.1. Sơ đồ tổng quát các use-case

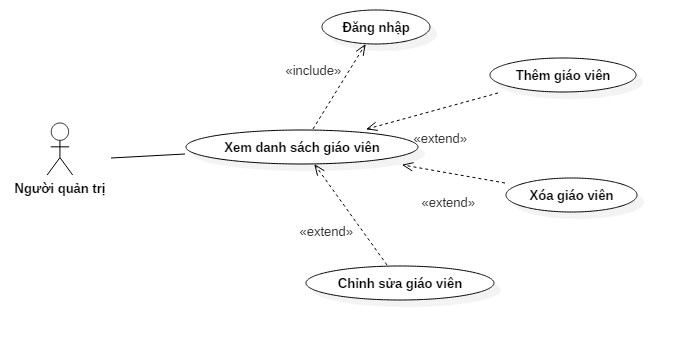
Hình 1: Sơ đồ use-case tổng quát

3.3.1.2. Phân rã các use-case chính.

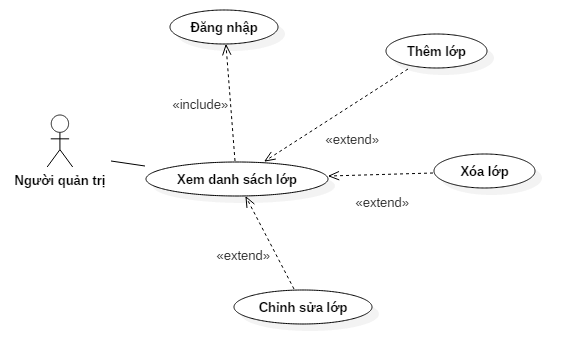
- Các use-case dành cho người quản trị



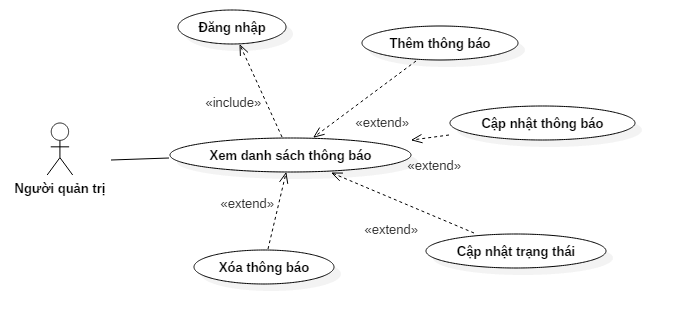
Hình 2: Sơ đồ use-case quản lí cơ quan

****

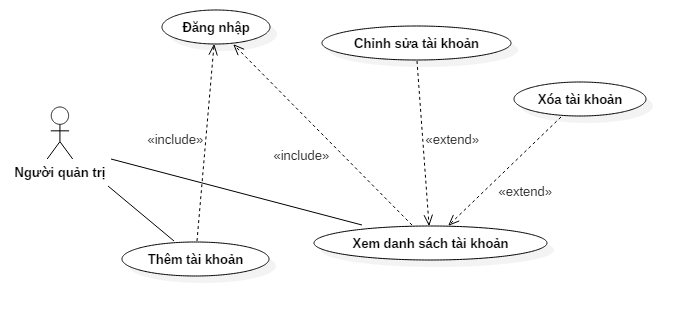
Hình 3: Sơ đồ use-case quản giáo viên

****

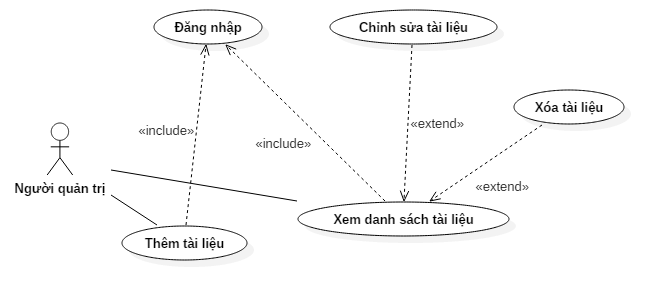
Hình 4: Sơ đồ use-case quản lí lớp

****

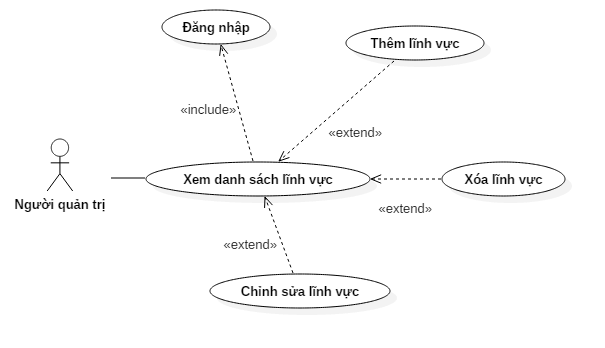
Hình 5: Sơ đồ use-case quản lí thông báo



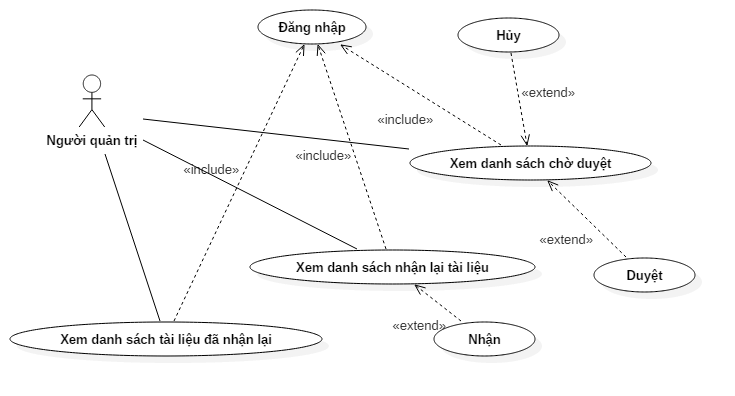
Hình 6: Sơ đồ use-case quản tài khoản



Hình 7: Sơ đồ use-case quản lí tài liệu

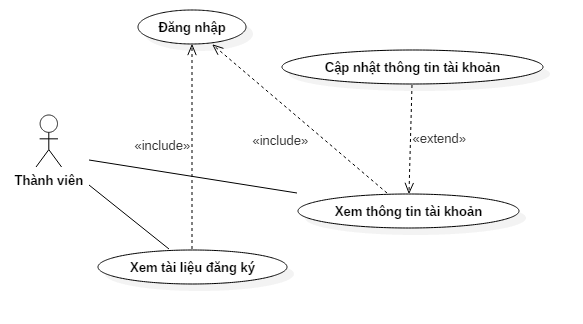


Hình 8: Sơ đồ use-case quản lí lĩnh vực

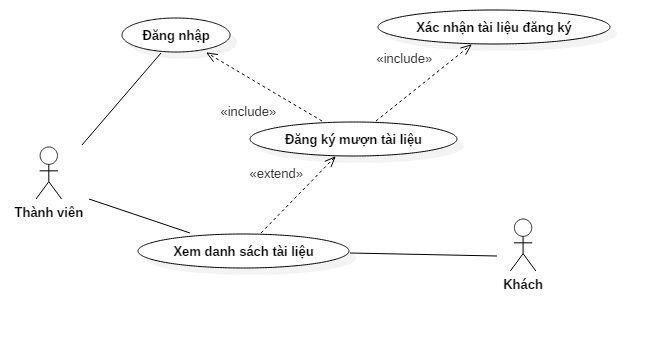


Hình 9: Sơ đồ use-case quản lí duyệt tài liệu

- Các use-case dành cho thành viên và khách vãng lai



Hình 10: Sơ đồ use-case quản lí tài khoản người dùng



Hình 11: Sơ đồ use-case đăng ký mượn – xem tài liệu

3.3.1.3. Đặc tả use-case

Bảng 4: Đặc tả use-case thêm cơ quan

|  |  |
| --- | --- |
| **Tên use-case:** Thêm cơ quan | **ID:** P01 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí cơ quan | **Mô tả ngắn gọn:** Người quản trị tạo thêm cơ quan. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí cơ quan. * - Nhập vào thông tin: tên cơ quan. * - Bấm vào nút thêm để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí cơ quan; nếu thông tin không hợp lệ, hệ thống sẽ thông báo và chuyển về trang Quản lí cơ quan. * - Use-case kết thúc. | |

Bảng 5: Đặc tả use-case chỉnh sửa cơ quan

|  |  |
| --- | --- |
| **Tên use-case:** Chỉnh sửa cơ quan | **ID:** P02 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí cơ quan | **Mô tả ngắn gọn:** Người quản trị chỉnh sửa cơ quan. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí cơ quan. * - Người quản trị ấn váo nút sửa với cơ quan tương ứng. * - Nhập vào thông tin: tên cơ quan. * - Bấm vào nút cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí lĩnh vực; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 6: Đặc tả use-case thêm giáo viên

|  |  |
| --- | --- |
| **Tên use-case:** Thêm giáo viên | **ID:** P03 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí giáo viên | **Mô tả ngắn gọn:** Người quản trị tạo thêm giáo viên. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí giáo viên. * - Nhập vào thông tin: tên giáo viên. * - Bấm vào nút thêm để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí giáo viên; nếu thông tin không hợp lệ, hệ thống sẽ thông báo và chuyển về trang Quản lí giáo viên. * - Use-case kết thúc. | |

Bảng 7: Đặc tả use-case chỉnh sửa giáo viên

|  |  |
| --- | --- |
| **Tên use-case:** Chỉnh sửa giáo viên | **ID:** P04 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí giáo viên | **Mô tả ngắn gọn:** Người quản trị chỉnh sửa giáo viên. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí giáo viên. * - Người quản trị ấn váo nút sửa với giáo viên tương ứng. * - Nhập vào thông tin: tên giáo viên. * - Bấm vào nút cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí giáo viên; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 8: Đặc tả use-case thêm lớp

|  |  |
| --- | --- |
| **Tên use-case:** Thêm lớp | **ID:** P05 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí lớp | **Mô tả ngắn gọn:** Người quản trị tạo thêm lớp. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí lớp. * - Nhập vào thông tin: tên lớp. * - Bấm vào nút thêm để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí lớp; nếu thông tin không hợp lệ, hệ thống sẽ thông báo và chuyển về trang Quản lí lớp. * - Use-case kết thúc. | |

Bảng 9: Đặc tả use-case chỉnh sửa lớp

|  |  |
| --- | --- |
| **Tên use-case:** Chỉnh sửa lớp | **ID:** P06 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí lớp | **Mô tả ngắn gọn:** Người quản trị chỉnh sửa lớp. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí lớp. * - Người quản trị ấn váo nút sửa với lớp tương ứng. * - Nhập vào thông tin: tên lớp. * - Bấm vào nút cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí lĩnh vực; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 10: Đặc tả use-case thêm thông báo

|  |  |
| --- | --- |
| **Tên use-case:** Thêm thông báo | **ID:** P07 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí thông báo | **Mô tả ngắn gọn:** Người quản trị tạo thêm thông báo. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí thông báo. * - Nhập vào thông tin: nội dung thông báo. * - Bấm vào nút thêm để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí thông báo; nếu thông tin không hợp lệ, hệ thống sẽ thông báo và chuyển về trang Quản lí thông báo. * - Use-case kết thúc. | |

Bảng 11: Đặc tả use-case chỉnh sửa thông báo

|  |  |
| --- | --- |
| **Tên use-case:** Chỉnh sửa thông báo | **ID:** P08 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí thông báo | **Mô tả ngắn gọn:** Người quản trị chỉnh sửa thông báo. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí lớp. * - Người quản trị ấn váo nút sửa với thông báo tương ứng. * - Nhập vào thông tin: nội dung thông báo. * - Bấm vào nút cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí thông báo; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 12: Đặc tả use-case thêm tài khoản

|  |  |
| --- | --- |
| **Tên use-case:** Thêm tài khoản | **ID:** P09 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí tài khoản | **Mô tả ngắn gọn:** Người quản trị tạo thêm tài khoản. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí tài khoản. * - Chọn mục thêm tài khoản * - Nhập vào các thông tin: tên đăng nhập, mật khẩu, email... * - Bấm vào nút thêm để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 13: Đặc tả use-case chỉnh sửa tài khoản

|  |  |
| --- | --- |
| **Tên use-case:** Chỉnh sửa tài khoản | **ID:** P10 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí tài khoản | **Mô tả ngắn gọn:** Người quản trị chỉnh sửa tài khoản. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí tài khoản. * - Người quản trị ấn váo nút sửa với tài khoản tương ứng. * - Nhập vào các thông tin: mật khẩu, email, họ tên... * - Bấm vào nút cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo cập nhật mới thành công; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 14: Đặc tả use-case thêm tài liệu

|  |  |
| --- | --- |
| **Tên use-case:** Thêm tài liệu | **ID:** P11 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí tào liệu | **Mô tả ngắn gọn:** Người quản trị tạo thêm tài liệu. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí tài liệu. * - Chọn mục thêm tài liệu * - Nhập vào các thông tin: tên tài liệu, lĩnh vực, giáo viên... * - Bấm vào nút thêm để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 15: Đặc tả use-case chỉnh sửa tài liệu

|  |  |
| --- | --- |
| **Tên use-case:** Chỉnh sửa tài liệu | **ID:** P12 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí tài liệu | **Mô tả ngắn gọn:** Người quản trị chỉnh sửa tài liệu. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí tài liệu. * - Người quản trị ấn váo nút sửa với tài liệu tương ứng. * - Nhập vào các thông tin: tên tài liệu, lĩnh vực, giáo viên... * - Bấm vào nút cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo cập nhật mới thành công; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 16: Đặc tả use-case thêm lĩnh vực

|  |  |
| --- | --- |
| **Tên use-case:** Thêm lĩnh vực | **ID:** P13 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí lĩnh vực | **Mô tả ngắn gọn:** Người quản trị tạo thêm lĩnh vực. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí lĩnh vực. * - Nhập vào thông tin: tên lĩnh vực. * - Bấm vào nút thêm để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thêm mới thành công, chuyển về trang Quản lí lĩnh vực; nếu thông tin không hợp lệ, hệ thống sẽ thông báo và chuyển về trang Quản lí lĩnh vực. * - Use-case kết thúc. | |

Bảng 17: Đặc tả use-case chỉnh sửa lĩnh vực

|  |  |
| --- | --- |
| **Tên use-case:** Chỉnh sửa lĩnh vực | **ID:** P14 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí lĩnh vực | **Mô tả ngắn gọn:** Người quản trị chỉnh sửa lĩnh vực. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang quản lí cơ quan. * - Người quản trị ấn váo nút sửa với cơ quan tương ứng. * - Nhập vào thông tin: tên cơ quan. * - Bấm vào nút cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, chuyển về trang Quản lí lĩnh vực; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

Bảng 18: Đặc tả use-case xem danh sách chờ duyệt

|  |  |
| --- | --- |
| **Tên use-case:** xem danh sách chờ duyệt | **ID:** P15 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí mượn – trả | **Mô tả ngắn gọn:** Người quản trị duyệt yêu cầu mượn tài liệu. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang chờ duyệt * - Chọn duyệt hoặc hủy để xác nhận yêu cầu. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, chuyển về trang chờ duyệt. * - Use-case kết thúc. | |

Bảng 19: Đặc tả use-case xem danh sách nhận lại tài liệu

|  |  |
| --- | --- |
| **Tên use-case:** xem danh sách nhận lại tài liệu | **ID:** P16 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí mượn – trả | **Mô tả ngắn gọn:** Người quản trị duyệt nhận lại tài liệu. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang nhận lại. * - Chọn nhận lại để xác nhận yêu cầu. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, chuyển về trang nhận lại. * - Use-case kết thúc. | |

Bảng 20: Đặc tả use-case xem danh sách tài liệu đã nhận lại

|  |  |
| --- | --- |
| **Tên use-case:** xem danh sách tài liệu đã nhận lại | **ID:** P17 |
| **Tác nhân chính:** Người quản trị | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản quản trị và chọn vào trang quản lí mượn – trả | **Mô tả ngắn gọn:** Người quản trị xem các tài liệu đã nhận lạ. |
| **Luồng sự kiện chính:**   * - Người quản trị chọn vào trang đã nhận lại. * - Hệ thống hiển thị danh sách tài liệu đã nhận lại. * - Use-case kết thúc. | |

Bảng 21: Đặc tả use-case xem tài liệu đăng ký

|  |  |
| --- | --- |
| **Tên use-case:** xem tài liệu đăng ký | **ID:** P18 |
| **Tác nhân chính:** Thành viên | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản thành viên và chọn vào trang tài liệu đăng ký mượn | **Mô tả ngắn gọn:** Thành viên xem tài liệu đã đăng ký. |
| **Luồng sự kiện chính:**   * - Thành viên chọn vào trang tài liệu đăng ký mượn. * - Hệ thống hiển thị danh sách tài liệu đăng ký mượn. * - Use-case kết thúc. | |

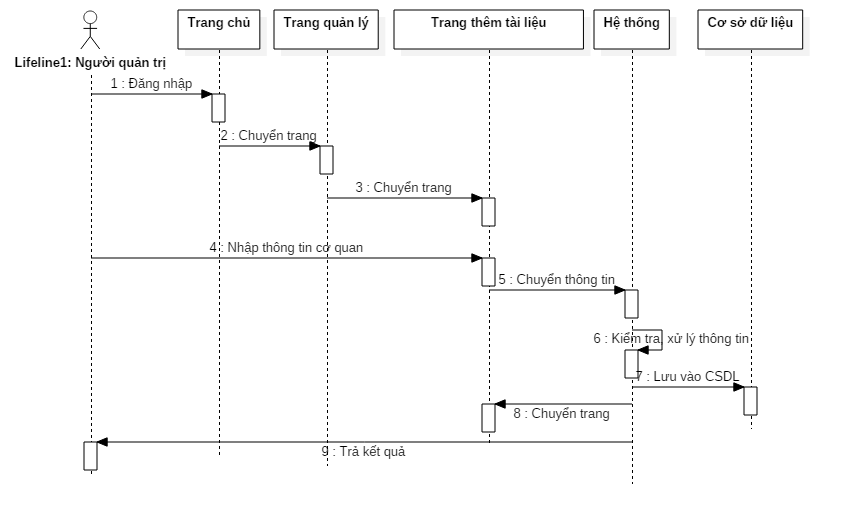
Bảng 22: Đặc tả use-case xem thông tin tài khoản

|  |  |
| --- | --- |
| **Tên use-case:** xem thông tin tài khoản | **ID:** P19 |
| **Tác nhân chính:** Thành viên | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản thành viên và chọn vào trang thông tin tài khoản | **Mô tả ngắn gọn:** Thành viên xem thông tin tài khoản. |
| **Luồng sự kiện chính:**   * - Thành viên chọn vào trang thông tin tài khoản. * - Hệ thống hiển thị thông tin tài khoản. * - Nếu muốn cập nhật thông tin, nhập các thông tin cần thay đổi. * - Ấn cập nhật để xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, thông báo thành công; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

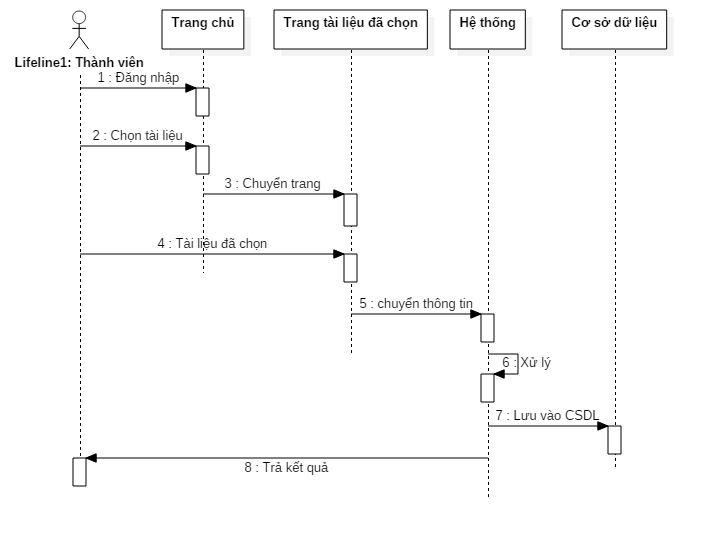
Bảng 23: Đặc tả use-case đăng ký mượn tài liệu

|  |  |
| --- | --- |
| **Tên use-case:** đăng ký mượn tài liệu | **ID:** P20 |
| **Tác nhân chính:** Thành viên | **Điều kiện:** Trang web đang hoạt động. |
| **Phạm vi:** Khi người dùng đăng nhập vào hệ thống với tài khoản thành viên và chọn trang chủ | **Mô tả ngắn gọn:** Thành viên đăng ký mượn tài liệu. |
| **Luồng sự kiện chính:**   * - Thành viên chọn vào trang chủ. * - Chọn các tài liệu cần mượn. * - Ấn mượn * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ chuyển đến trang xác nhận đăng ký; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Thành viên chọc mốc thời gian và xác nhận. * - Hệ thống tiến hành kiểm tra và trả kết quả. * - Nếu thông tin hợp lệ hệ thống sẽ lưu vào cơ sở dữ liệu, chuyển đến trang tài liệu đăng ký; nếu thông tin không hợp lệ, hệ thống sẽ thông báo lỗi. * - Use-case kết thúc. | |

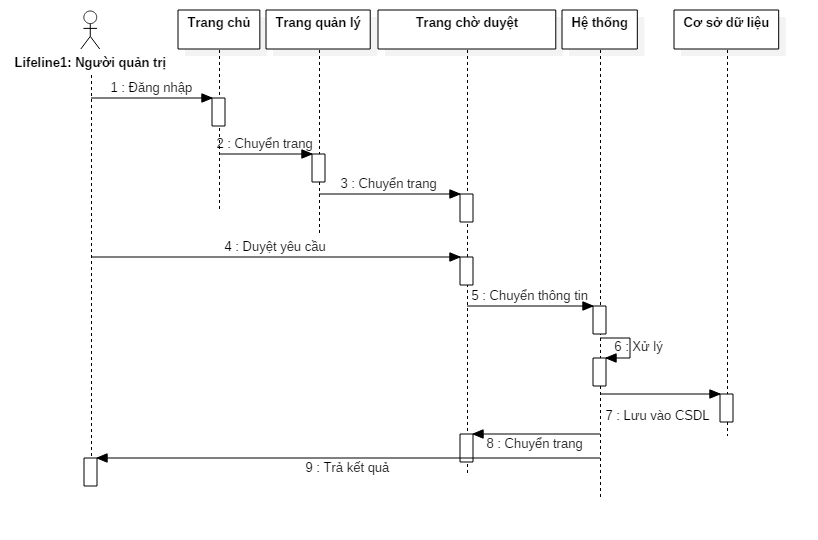
3.3.2. Sơ đồ tuần tự các use-case



Hình 12: Sơ đồ tuần tự use-case Thêm tài liệu



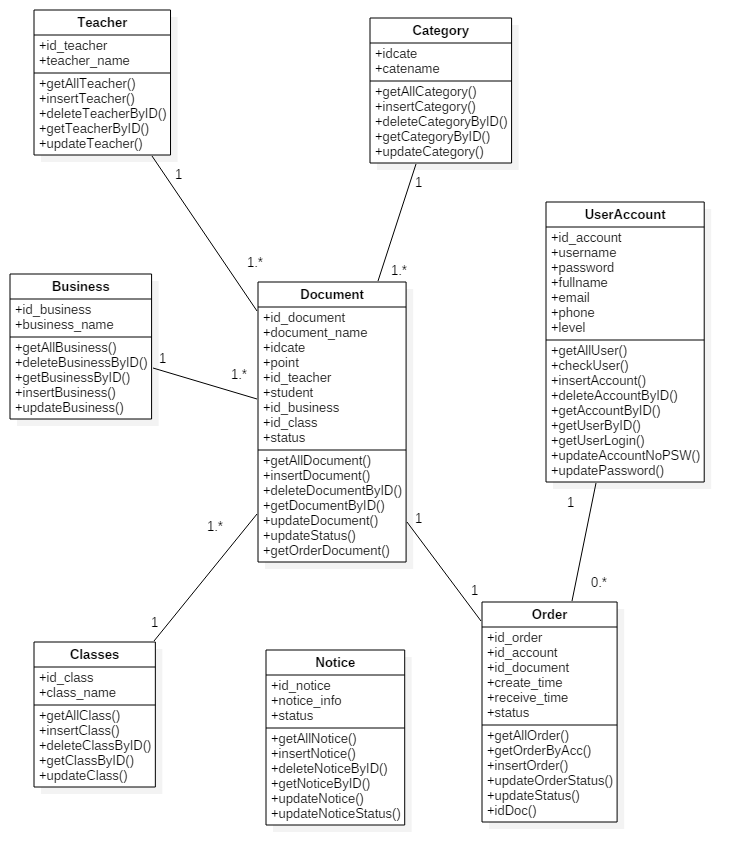
Hình 13: Sơ đồ tuần tự use-case Đăng ký mượn tài liệu

****

Hình 14: Sơ đồ tuần tự use-case Duyệt yêu cầu đăng ký mượn tài liệu

3.3.3. Các lớp đối tượng

3.3.3.1. Sơ đồ lớp



Hình 15: Sơ đồ lớp tổng quát

3.3.3.2. Chi tiết một số lớp đối tượng chính

Bảng 24: Chi tiết một số lớp đối tượng chính

|  |  |  |
| --- | --- | --- |
| Tên lớp | Các thuộc tính | Các phương thức |
| Category | +idcate: int;  +catename: String | Constructors, getters, setters.  +getCategoryByID(): Category;  +getAllCategory(): ArrayList<Category>;  +insertCategory(): void;  +deleteCategoryByID(): void;  +updateCategory(): void; |
| Teacher | +id\_teacher: int;  +teacher\_name: String; | Constructors, getters, setters.  +getTeacherByID(): Teacher;  +getAllTeacher(): ArrayList<Teacher>;  +insertTeacher(): void;  +deleteTeacherByID(): void;  +updateTeacher(): void; |
| Business | +id\_business: int;  +business\_name: String; | Constructors, getters, setters.  +getBusinessByID(): Business;  +getAllBusiness (): ArrayList<Business>;  +insertBusiness(): void;  +deleteBusinessBy ID(): void;  +updateBusiness(): void; |
| Classes | +id\_class: int;  +class\_name: String; | Constructors, getters, setters.  +getClassByID(): Classes;  +getAllClass(): ArrayList<Classes>;  +insertClass (): void;  +deleteClassByID(): void;  +updateClass(): void; |
| UserAccount | +id\_account: int  +username: String;  +password: String;  +fullname: String;  +email: String;  +phone: String;  +level: boolean | Constructors, getters, setters.  +getAccountByID(): UserAccount;  +getAllUser():  ArrayList< UserAccount >;  +insertAccount(): void;  +deleteAccountByID(): void;  +getUserByID(): UserAccount ;  +getUserLogin(): UserAccount ;  +updateAccountNoPSW() : void ;  +updatePassword(): void ;  +checkUser(): boolean; |
| Document | +id\_document: int;  +document\_name:String;  +idcate: int;  +id\_teacher: int;  +point: float;  +student: String;  +id\_business: int;  +id\_class: int;  +status: int; | Constructors, getters, setters.  +getDocumentByID(): Document;  +getAllDocument(): ArrayList<Document>;  +insertDocument(): void;  +deleteDocumentByID(): void;  +updateDocument(): void;  +updateStatus(): void;  +getOrderDocument(): ArrayList<Document>; |
| Order | +id\_order: int;  +id\_account: int;  +id\_document: int;  +create\_time: String;  +recieve\_time: String;  +status: int; | Constructors, getters, setters.  +getOrderByAcc(): ArrayList<Order>;  +getAllOrder(): ArrayList< Order>;  +insertOrder(): void;  +deleteOrder(): void;  +updateOrderStatus(): void;  +updateStatus(): void;  +idDoc(): int; |
| Notice | +id\_notice: int;  +notice\_info: String;  +status: boolean | Constructors, getters, setters.  +getNoticeByID(): Notice;  +getAllNotice(): ArrayList<Notice>;  +insertNotice (): void;  +deleteNoticeByID(): void;  +updateNotice(): void;  +updateNoticeStatus(): void; |

Chương 4

THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG

4.1. Tổ chức dữ liệu

Các đối tượng dữ liệu được tổ chức thành các lớp theo mô hình lớp như Hình 11 và được lưu trữ trong cơ sở dữ liệu với các bảng tương ứng như sau.

Bảng 25: Chi tiết bảng categorys

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| idcate | int(11) | Khóa chính, not null | Mã lĩnh vực |
| catename | varchar(20) |  | Tên lĩnh vực |

Bảng 26: Chi tiết bảng business

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id\_business | int(11) | Khóa chính, not null | Mã cơ quan |
| business\_name | varchar(200) |  | Tên cơ quan |

Bảng 27: Chi tiết bảng class

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id\_class | int(11) | Khóa chính, not null | Mã lớp |
| class\_name | varchar(6) |  | Tên lớp |

Bảng 28: Chi tiết bảng teacher

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id\_teacher | int(11) | Khóa chính, not null | Mã giáo viên |
| teacher\_name | varchar(50) |  | Tên giáo viên |

Bảng 29: Chi tiết bảng documents

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id\_document | int(11) | Khóa chính, not null | Mã tài liệu |
| document\_name | varchar(500) | Not null | Tên tài liệu |
| idcate | int(11) | Khóa ngoại | Mã lĩnh vực |
| id\_teacher | int(11) | Khóa ngoại | Mã giáo viên |
| point | float |  | Điểm |
| student | varchar(50) |  | Tên sinh viên |
| id\_business | int(11) | Khóa ngoại | Mã cơ quan |
| id\_class | int(11) | Khóa ngoại | Mã lớp |
| status | int |  | Trạng thái |

Bảng 30: Chi tiết bảng user

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id\_account | int(11) | Khóa chính, not null | Mã tài khoản |
| username | varchar(20) | Not null | Tên tài khoản |
| password | varchar(200) |  | Mật khẩu |
| fullname | varchar(50) |  | Họ tên |
| email | varchar(50) |  | Điểm |
| phone | varchar(10) |  | Tên sinh viên |
| level | tinyint(1) |  | Mã cơ quan |

Bảng 9: Chi tiết bảng orders

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id\_order | int(11) | Khóa chính, not null | Mã đặt tài liệu |
| id\_account | int(11) | Khóa ngoại | Mã tài khoản |
| id\_document | int(11) | Khóa ngoại | Mã tài liệu |
| create\_time | datetime |  | Thời gian tạo |
| recieve\_time | datetime |  | Thời gian nhận |
| status | int(11) |  | Trạng thái |

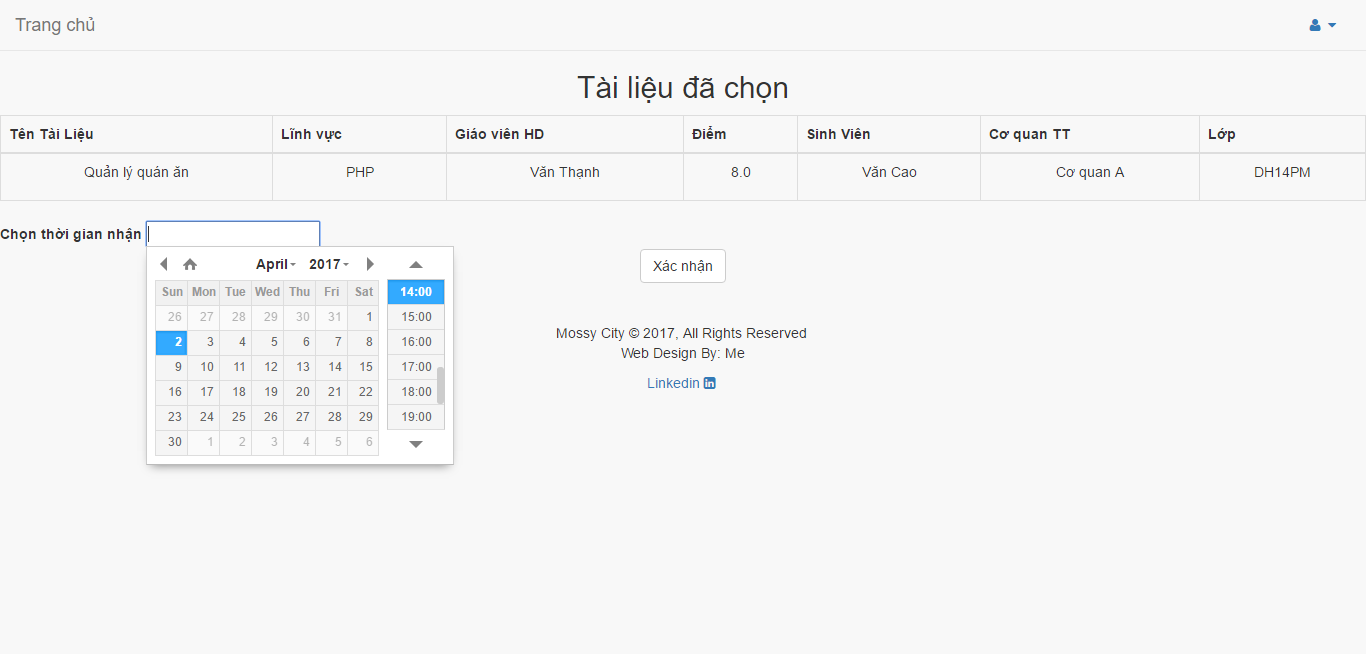
Bảng 31: Chi tiết bảng notice

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id\_notice | int(11) | Khóa chính, not null | Mã thông báo |
| notice\_name | varchar(500) |  | Nội dung thông báo |
| status | tinyint(1) |  | Trạng thái |

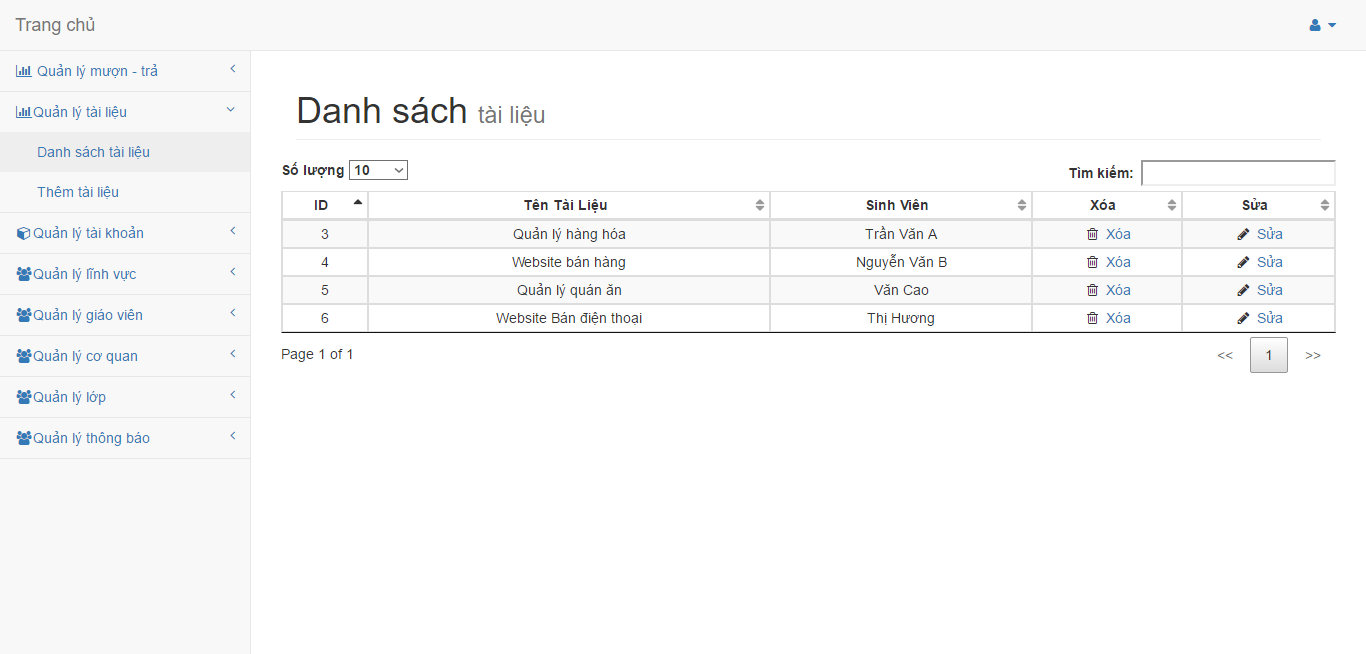
4.2. Giao diện Website



Hình 16: Giao diện trang chủ.



Hình 17: Giao diện đăng ký mượn tài liệu.



Hình 18: Giao diện trang quản lý.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

5.1.1. Sản phẩm

Trang web thực hiện được các yêu cầu cơ bản sau:

* Các chức năng quản lí: quản lí lĩnh vực, quản lí tài liệu, quản lí người dùng, quản lí yêu cầu mượn tài liệu.
* Các chức năng dành cho khách hàng: Xem danh sách tài liệu, đặt mượn tài liệu.

5.1.2. Kiến thức

5.1.2.1. Về lý thuyết

Kỳ thực tập cuối khóa này giúp em tích lũy thêm kiến thức mới cũng như trau dồi những kiến thức đã có về java, mysql, html, uml..

5.1.2.2. Về kỹ năng

Trong quá trình thực tập em đã gặp rất nhiều khó khăn từ những lỗ hỏng kiến thức của mình, tuy nhiên qua đó em học được cách tự nghiên cứu tài liệu, cách giải quyết vấn đề. Ngoài ra, em còn áp dụng được những kĩ thuật phân tích, thiết kế đã học vào thực tiễn, làm giảm khả năng thất bại khi bắt đầu một dự án.

5.1.2.3. Về kinh nghiệm thực tiễn

Đợt thực tập này còn giúp em phát triển kĩ năng giao tiếp, ứng xử cũng như hòa nhập và chấp hành các qui định của tổ chức.

Bên cạnh đó, em còn được rèn luyện các kĩ năng quản lí thời gian, chi phí, kĩ năng viết báo cáo, biết xác định khả năng, công việc phù hợp.

5.2. Các hạn chế

5.2.1. Về sản phẩm

Vẫn còn tồn tại một số lỗi:

* Các ràng buộc dữ liệu chưa thực sự hiệu quả.
* Sử dụng dữ liệu chưa tiết kiệm.

5.2.2. Về bản thân

Trong quá trình thực tập bản thân em cũng gặp phải một số lỗi không đáng có:

* Chưa viết code theo mô hình chuẩn.
* Tốn quá nhiều thời gian tìm hiểu kĩ thuật.

5.3. Hướng phát triển

Cần phải hoàn thiện các xử lí trong phần web server:

* Khắc phục các sự cố còn lại.
* Chuẩn hóa cách tổ chức đối tượng theo mô hình Spring MVC.
* Tối ưu hóa việc sử dụng dữ liệu.
* Hoàn thiện các ràng buộc dữ liệu để đảm bảo tính nhất quán.
* Tối đa hóa việc bắt lỗi và thông báo lỗi.
* Tăng cường tính tương tác giữa người dùng phía quản trị website bằng cách tích hợp các chức năng hỗ trợ trực tuyến.
* Tăng cường tính bảo mật của website.
* Đưa website lên host và đưa vào sử dụng thực tế sau khi hoàn thiện chức năng.

Tăng cường khả năng tương thích cũng như tăng tính thẩm mỹ trong phần web client.

* Phát triển khả năng responsive – tương thích với mọi loại thiết bị.
* Đơn giản hóa các menu, để người không có kiến thức tin học vẫn sử dụng được.

# TÀI LIỆU THAM KHẢO