TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

**KHOA KỸ THUẬT – CÔNG NGHỆ**



**BÁO CÁO TỐT NGHIỆP**

**Đề tài: XÂY DỰNG WEBSITE HỆ THỐNG THÔNG TIN VỀ BẤT ĐỘNG SẢN BẰNG CÔNG NGHỆ WEBGIS**

GIẢNG VIÊN HƯỚNG DẪN: **ThS. NGUYỄN THỊ THỦY**

SINH VIÊN THỰC HIỆN:

**PHẠM TRUNG HIẾU** MSSV: **1624801040020**

NIÊN KHÓA: **2016-2020**

**08/2019**

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

**KHOA KỸ THUẬT – CÔNG NGHỆ**



**BÁO CÁO TỐT NGHIỆP**

**Đề tài: XÂY DỰNG WEBSITE HỆ THỐNG THÔNG TIN VỀ BẤT ĐỘNG SẢN BẰNG CÔNG NGHỆ WEBGIS**

GIẢNG VIÊN HƯỚNG DẪN: **ThS. NGUYỄN THỊ THỦY**

SINH VIÊN THỰC HIỆN:

**PHẠM TRUNG HIẾU** MSSV: **1624801040020**

NIÊN KHÓA: **2016-2020**

**08/2019**

# QUYẾT ĐỊNH GIAO ĐỀ TÀI

# NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

*Bình Dương, ngày 17 tháng 7 năm 2020*

***Giáo viên hướng dẫn***

**ThS. Nguyễn Thị Thủy**

# NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

*Bình Dương, ngày 17 tháng 7 năm 2020*

***Giáo viên phản biện***

**ThS. Nguyễn Tấn Lộc**

# LỜI CẢM ƠN

Trước tiên, Em xin chân thành cám ơn tới quý thầy cô **Khoa Kỹ thuật – Công nghệ** đã tận tình dạy dỗ, chỉ bảo để em có được những kiến thức nền tảng cho việc làm đề tài tốt nghiệp và đáp ứng yêu cầu công việc sau này.

Em xin gửi lời cám ơn sâu sắc nhất đến cô **ThS. Nguyễn Thị Thủy** (Khoa Kỹ thuật - Công nghệ, trường Đại học Thủ Dầu Một) đã tận tình hướng dẫn em trong suốt quá trình tìm hiểu nghiên cứu và hoàn thành tiểu luận tốt nghiêp̣.

Xin chân thành cám ơn thầy cô, bạn bè đã hỗ trợ và tạo động lực để em hoàn thành tiểu luận đúng thời gian.

Mặc dù đã có nhiều cố gắng song do hạn chế về thời gian, chắc chắn đề tài không thể tránh khỏi những thiết sót, rất mong nhận được ý kiến đóng góp của thầy cô và bạn bè để đề tài có thể hoàn thiện tốt hơn.

Xin chân thành cảm ơn!

.

*Sinh viên thực tập*

**Phạm Trung Hiếu**

# LỜI CAM ĐOAN

# TÓM TẮT

# SUMMARY

# MỤC LỤC

[**LỜI MỞ ĐẦU 7**](#_Toc20261483)

[**CHƯƠNG I. GIỚI THIỆU VỀ CƠ SỞ THỰC TẬP 8**](#_Toc20261484)

[**1. Giới thiệu chung 8**](#_Toc20261485)

[**2. Chức năng và nhiệm vụ 9**](#_Toc20261486)

[**CHƯƠNG II. CƠ SỞ LÍ THUYẾT 10**](#_Toc20261487)

[**1. Tìm hiểu RESTFULL API 10**](#_Toc20261488)

[**2. Tìm hiểu Git 11**](#_Toc20261489)

[2.1. Cài đặt và thiết lập ban đầu để làm việc với Git 11](#_Toc20261490)

[2.2. Cách tạo một Repository 13](#_Toc20261491)

[2.3. Các lệnh cơ bản 14](#_Toc20261492)

[2.4. Nhánh trong Git (Branch) 15](#_Toc20261493)

[**3. Tìm hiểu Agile 16**](#_Toc20261494)

[3.1. Tôn chỉ và các nguyên tắc trong Agile 16](#_Toc20261495)

[3.2. Quy trình Agile được thực hiện như thế nào? 17](#_Toc20261496)

[**4. Tìm hiểu Scrum 18**](#_Toc20261497)

[4.1. Vai trò trong Scrum 18](#_Toc20261498)

[4.2. Các sự kiện trong Scrum 19](#_Toc20261499)

[4.3. Đồ nghề trong Scrum 20](#_Toc20261500)

[**5. Tìm hiểu Spring MVC 22**](#_Toc20261501)

[5.1. Tổng quan về Spring MVC 22](#_Toc20261502)

[5.2. DispathcherServlet 24](#_Toc20261503)

[5.3. Controllers 24](#_Toc20261504)

[5.4. Handler Mapping 25](#_Toc20261505)

[5.5. ViewResolver 26](#_Toc20261506)

[**6. Tìm hiểu MySQL 28**](#_Toc20261507)

[**7. Tìm hiểu Hibernate Framework 29**](#_Toc20261508)

[**8. Tìm hiểu ORM 30**](#_Toc20261509)

[**9. Tìm hiểu HQL 31**](#_Toc20261510)

[**CHƯƠNG III. PHẦN THỰC HÀNH 35**](#_Toc20261511)

[**1. Phân tích hệ thống 35**](#_Toc20261512)

[1.1. Xác định các tác nhân 35](#_Toc20261513)

[1.2. Biểu đồ UC tổng quát 36](#_Toc20261514)

[1.3. Đặc tả các UC 36](#_Toc20261515)

[1.4. Biểu đồ hoạt động 39](#_Toc20261516)

[**2. Khởi tạo server với Spring Boot,TomCat,Mango 46**](#_Toc20261517)

[2.1. Thiết kế cơ sở dữ liệu 46](#_Toc20261518)

[2.2. Các Url thao tác với dữ liệu 50](#_Toc20261519)

[**3. Giao diện hệ thống reponsive sử dựng Bootstrap, AngularJS 56**](#_Toc20261520)

[3.1. Giao diện trang chủ cho Độc Giả 56](#_Toc20261521)

[3.2. Giao diện trang Kết Quả Tìm Kiếm Sách 56](#_Toc20261522)

[3.3. Giao diện trang Chi Tiết Sách 57](#_Toc20261523)

[3.4. Giao diện trang Quản lý Sách 57](#_Toc20261524)

[3.5. Giao diện Thêm Sách 58](#_Toc20261525)

[3.6. Giao diện Sửa Sách 58](#_Toc20261526)

[3.7. Giao diện trang quản lý Nhà Xuất Bản 59](#_Toc20261527)

[3.8. Giao diện trang quản lý Độc Giả 59](#_Toc20261528)

[3.9. Giao diện trang Quản lý Tác Giả 60](#_Toc20261529)

[3.10.Giao diện trang Quản lý Mượn 60](#_Toc20261530)

[3.11.Giao diện trang Quản lý Trả 61](#_Toc20261531)

[3.12.Giao diện trang Thống Kê 61](#_Toc20261532)

[**KẾT LUẬN 62**](#_Toc20261533)

[**1. Kết luận 62**](#_Toc20261534)

[**2. Kiến nghị 63**](#_Toc20261535)

[**TÀI LIỆU THAM KHẢO 64**](#_Toc20261536)

# DANH MỤC CÁC HÌNH

Hình I.1.1 Văn phòng trụ sở chính – ISC Quang Trung 9

Hình I.2.1 Các chuyên ngành đào tạo và thực tập của ISC Quang Trung 9

Hình II.1.1 Cách thức hoạt động cua RESTFull API 10

Hình II.2.2.1.1 Chọn Git Bash here 12

Hình II.2.2.1.2 Cấu hình Tên và Email 12

Hình II.2.2.2.1 Tạo Local Repository 13

Hình II.2.2.2.2 Tạo Remote Respository 13

Hình II.2.2.2.3 Điền thông tin tạo Remote Repository 14

Hình II.2.2.4.1 Nhánh trong Git 15

Hình II.4.4.3.1 Biểu Đồ Burndown 21

Hình II.4.4.3.2 Bảng công việc 21

Hình II.5.5.1.1 Quy trình hoạt động của Spring MVC 23

Hình II.5.5.1.2 DispatcherServlet 23

Hình II.5.5.2.1 Khai báo và định nghĩa DispatcherSerlvelt 24

Hình II.5.5.3.1 Controller Interface 25

Hình II.5.5.4.1 Ánh xạ HTTP Request 26

Hình II.5.5.5.1 Controller Internal Resource 27

Hình II.5.5.5.2 Khai báo Internal Resource trong WEB.XML 27

Hình II.9.1 Mệnh đề Form 32

Hình II.9.2 Mệnh đề Form 32

Hình II.9.3 Mệnh đề AS 32

Hình II.9.4 Mệnh đề Select 32

Hình II.9.5 Mệnh đề Where 33

Hình II.9.6 Mệnh đề Order By 33

Hình II.9.7 Mệnh đề Group By 33

Hình II.9.8 Mệnh đề Update 33

Hình II.9.9 Mệnh đề Delete 34

Hình II.9.10 Mệnh đề Insert 34

Hình II.9.11 Mệnh đề Thêm Đối Tượng 34

Hình III.1.1.4.1 Biểu đồ hoạt động của UC Tìm Kiếm Sách 40

Hình III.1.1.4.2 Biểu đồ hoạt động của UC Them, Xoa, Sua SACH 41

Hình III.1.1.4.3 Biểu đồ hoạt động của UC Them, Xoa, Sua NHAXUATBAN 41

Hình III.1.1.4.4 Biểu đồ hoạt động của UC Them, Xoa, Sua THELOAI 42

Hình III.1.1.4.5 Biểu đồ hoạt động của UC Them, Xoa, Sua TACGIA 43

Hình III.1.1.4.6 Biểu đồ hoạt động của UC Them, Xoa, Sua DOCGIA 43

Hình III.1.1.4.7 Biểu đồ hoạt động của UC QuanLy Muon 44

Hình III.1.1.4.8 Biểu đồ hoạt động của UC QuanLy Tra 45

Hình III.1.1.4.9 Biểu đồ hoạt động của UC ThongKe 45

Hình III.3.3.1 Giao diện Trang Chủ 55

Hình III.3.3.2 Giao diện trang Kết Quả Tìm Kiếm Sách 55

Hình III.3.3.3 Giao diện trang Chi Tiết Sách 56

Hình III.3.3.4 Giao diện trang Quản lý Sách 56

Hình III.3.3.5 Giao diện Thêm Sách 57

Hình III.3.3.6 Giao diện Sửa Sách 57

Hình III.3.3.7 Giao diện trang Quản lý Nhà Xuất Bản 58

Hình III.3.3.8 Giao diện trang Quản lý Độc Giả 58

Hình III.3.3.9 Giao diện trang Quản lý Tác Giả 59

Hình III.3.3.10 Giao diện trang Quản lý Mượn 59

Hình III.3.3.11 Giao diện trang Quản lý Trả 60

Hình III.3.3.12 Giao diện trang Thống Kê 60

# DANH MỤC CÁC BẢNG

Bảng DAUSACH 47

Bảng SACH 47

Bảng DOCGIA 48

Bảng DS\_TG 48

Bảng LOAISACH 48

Bảng NXB 48

Bảng PHIEUMUON 49

Bảng PHIEUTRA 49

Bảng PM\_S 49

Bảng PT\_S 49

Bảng TACGIA 50

Bảng THUTHU 50

Bảng mô tả Url ControllerBookTitle 50

Bảng mô tả Url ControllerBook 50

Bảng mô tả Url ControllerAuthor 50

Bảng mô tả Url BorrowBooksController 50

Bảng mô tả Url BorrowBooksDetailController 52

Bảng mô tả Url ControllerLibrarian 52

Bảng mô tả Url ControllerPublisher 53

Bảng mô tả Url ControllerReader 53

Bảng mô tả Url ReturnBooksController 54

Bảng mô tả Url ReturnBooksDetailController 54

Bảng mô tả Url SubjectController 55

# DANH MỤC THUẬT NGỮ, VIẾT TẮT

# CHƯƠNG 1. KHẢO SÁT HIỆN TRẠNG VÀ SÁT ĐỊNH YÊU CẦU

## GIỚI THIỆU CHUNG

Ngày nay, ngành Công nghệ thông tin đang phát triển mạnh mẽ và có mặt trong hầu hết các lĩnh vực của đời sống xã hội. Nó cũng tạo nên bước đột phá mạnh mẽ trong việc duy trì và thúc đẩy sự phát triển của các ngành khác.

Việc xây dựng các website thương mại điện tử không còn mấy xa lạ so với người dùng Internet. Đối với những doanh nghiệp, công ty, cá nhân có nhu cầu về lĩnh vực nhà đất thì một website hỗ trợ việc buôn bán nhà đất là thực sự rất cần thiết. Nó có thể giúp người có nhu cầu mua hoặc bán đăng tin rao để thu hút khách hàng một cách nhanh chóng, tiện lợi và tiết kiệm nhất. Từ những nhu cầu thực tế trên, em đã chọn đề tài “**Xây dựng Website hệ thống thông tin về Bất Động Sản bằng công nghệ WEBGIS**”.

Sau thời gian học tập tại trường Đại học Thủ Dầu Một, dưới sự tận tâm truyền đạt của các thầy cô giáo trong khoa Kỹ thuật – Công nghệ, em đã trao dồi và tích lũy được vốn kiến thức nhất định. Từ việc thực hiện đề tài, em mong muốn cải thiện khả năng lập trình cũng như nâng cao khả năng tự nghiên cứu học hỏi để có thể đáp ứng tốt cho công việc sau này.

## Lý do thực hiện đề tài

Ngày nay, tin học phát triển mạnh mẽ tạo bước tiến vượt bậc cho đa số các ngành, đồng thời thúc đẩy sự phát triển của kinh tế, văn hóa, xã hội, giáo dục. Tin học đã dần dần trở thành nhu cầu rất cần thiết và không thể thiếu trong cuộc sống của con người.

Hiện nay, nhu cầu về lĩnh vực nhà đất tăng cao đột biến. Với phương pháp thông thường là người cần mua hoặc cần bán sẽ thuê người rao tin bằng cách phát tờ rơi, hoặc giới thiệu thông qua bạn bè, người thân. Phương pháp này chỉ áp dụng có kết quả với những người nhận được tờ rao, truyền miệng và người đó có nhu cầu, nhưng chỉ có tác dụng ở một phạm vi xác định và một số người nhất định nên không đạt được hiệu quả cao như mong muốn, đôi khi nó khiến cho người cần rao tin mất thời gian cũng như chi phí rao tin hoặc người có nhu cầu không tiếp cận được tin rao đó. Chính vì thế, để giải quyết được vấn đề trên, tin học hóa việc buôn bán ở lĩnh vực nhà đất nên em chọn thực hiền đề tại “ **Xây dựng Website hệ thống thông tin về Bất Động Sản bằng công nghệ WEBGIS** ” để nghiên cứu và thực hiện.

Nhờ tính hiệu quả và nhanh chóng do hệ thống mới đem lại làm tăng hiệu quả trong việc rao tin, tiết kiệm được khá nhiều thời gian, công sức. Vì thế việc đăng tin để tìm khách hàng có nhu cầu cũng như tra cứu thông tin nhà đất theo ý muốn cũng trở nên dễ dàng và nhanh chóng, đảm bảo tính xác thực cao, đồng thời tính hiện đại hóa được giải quyết.

## Mục đích nghiên cứu

Thông tin sẽ được biểu diễn có hệ thống hơn, chính xác hơn so với cách thực hiện truyền thống. Ngoài ra, còn giảm được những tin rao thông tin chưa chính xác hoặc chưa xác minh được người đăng.

Giảm được thời gian cho việc trực tiếp tìm kiếm.

Giảm chi phí khác hỗ trợ trong việc đăng tin rao. Hê ̣thống sẽ giúp người dùng đăng tin một cách dễ dàng, nhanh chóng và đạt hiệu quả cao.

## Đối tượng và phạm vi nghiên cứu

Đối tượng: Công ty và Khách hàng Nghiệp vụ về mua bán trong lĩnh vực nhà đất.

Môi trường nghiên cứu: Tìm hiểu các quy trình nghiệp vụ chung về lĩnh vực nhà đất, qua đó xây dựng ứng dụng Web thực hiện chức năng giúp công ty và người dùng xem tin rao theo nhu cầu của họ, thực hiện đăng tin, và hệ thống giúp người quản trị quản lý tất cả thông tin của người dùng trong hệ thống trên nền tảng website.

Thời gian thực hiện: Tiểu luận được giao cho 01 người thực hiện trong thời gian 12 tuần kể từ ngày nhận.

## Phương pháp nghiên cứu

Khảo sát thực tế, tìm hiểu công nghệ, phân tích thiết kế hệ thống và lập trình ra ứng dụng là sản phẩm của đề tài.

### **Các phần mềm hỗ trợ:**

+ Hệ quản trị cơ sở dữ liệu: PostgresSQL.

+ Số hóa dữ liệu :

+ Xây dựng Restful API: JAVA.

+ Thiết kế và xây dựng chức năng Website: Re, Openlayers, AngularJS

### **Cách tổ chức xây dựng hệ thống :**

+ Xây dựng Restful api với ngôn ngữ lập trình JAVA

+ Lưu trữ cơ sở dữ liệu với PostgreSQL

+ Lưu trữ hình ảnh với FileBase

+ Xây dựng giao diện Website với HTML, CSS, JAVASCRIPT, BOOSTRAP

+ Thiết kế chức năng với Framework AngularJS.

## Ý nghĩa khoa học và thực tiễn

### **Ý nghĩa khoa học**

Xây dựng được hệ thống website giúp công ty và người dùng có thể đăng tin rao bất độg sản

### **Ý nghĩa thực tiễn**

Công ty và người dùng kênh để đăng tin rao bán bất động sản

## Bố cục bài báo cáo

Đề tài gồm các phần được phân chương như sau:

* + Chương 1: Khảo sát hiện trạng và xác định yêu cầu
  + Chương 2: Phân tích hệ thống
  + Chương 3: Thiết kế hê ̣thống
  + Kết luận và hướng phát triển
  + Tài liệu tham khảo

## Hiện trạng

Lĩnh vực nhà đất đang trong trong thời kỳ phát triển nhanh do nhu cầu của người dân ngày càng tăng nhanh trong thời gian gần đây. Nhưng với cách rao tin bằng tờ rơi theo phương pháp thủ công và truyền miệng như trước đây thì không còn phù hợp nữa vì nó khiến chúng ta mất khá nhiều thời gian và hiệu quả không cao, đồng thời dữ liệu phân tán lẻ tẻ không có sự tập trung nên khó quản lý, thống kê.

Một số nghiệp vụ như tra cứu, thống kê, và hiệu chỉnh thông tin thì rất khó để thực hiện được. Ngoài ra còn một số khó khăn về vấn đề lưu trữ khá lớn, dễ bị thất lạc, tốn kém … vì số lượng tin rao là rất lớn.

Khi muốn lập một danh sách, hay thống kê số liệu đáp ứng nhu cầu thực tế thì đây là công việc rất vất vả và tốn rất nhiều thời gian.

Với sự giúp đỡ, hỗ trợ của tin học việc thực hiên đăng tin rao và quản lý sẽ trở nên đơn giản hơn, thuận tiện, nhanh chóng, chính xác và hiệu quả hơn rất nhiều.

Dễ dàng tiếp cận với người có nhu cầu mua Bất Động Sản.

## Yêu cầu chức năng hệ thống

Admin: Người quản trị hệ thống

NhanVien: Người dùng thông thường trang quản trị

KhacHang: Người dùng trang user.

Bảng 1.1 Bảng chức năng hệ thống

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Chức năng** | **Mô tả** | **Nhóm**  **Người dùng** |
| 1 | Tìm kiếm | Tìm kiếm với nhiều thông tin | KhacHang |
| 2 | Xem thông tin tin rao | Xem thông tin chi tiết tin rao | KhacHang |
| 3 | Đăng Ký | Đăng ký tài khoản | KhacHang |
| 4 | Đăng Nhập | Đăng nhập website dành cho Khách Hàng | KhacHang |
| 5 | Quên Mật Khẩu | Lấy lại mật khẩu đã quên | KhacHang |
|  | Đặt lại mật khẩu | Thay đổi mật khẩu mới | KhacHang |
| 6 | Đăng tin | Đăng tin rao | KhacHang |
| 7 | Chỉnh sửa tin đăng | Chỉnh sửa thông tin tin đã đăng | KhacHang |
|  | Chỉnh sửa hình ảnh bất động sản | Chỉnh sửa nhiều hình ảnh chi tiết bất động sản | KhacHang |
| 8 | Cập Nhật thông tin tài khoản | Chỉnh sửa thông tin tài khoản | KhacHang |
| 9 | Bất động sản yêu thích | Thêm bất động sản đã xem vào danh sách yêu thích | KhacHang |
|  | Đăng nhập | Đăng nhập website quản trị | Admin, NhanVien |
|  | Xem danh sách phường | Hiển thị tên các phường trong Thủ Dầu Một | Admin, NhanVien |
|  | Xem danh sách đường | Hiển thị tên các đường trong Thủ Dầu Một | Admin, NhanVien |
|  | Quản lý tin đăng Bất Động Sản Công ty | Thêm tin đăng bất động sản công ty | Admin, NhanVien |
| Sửa tin đăng bất động sản công ty | Admin, NhanVien |
| Xóa tin đăng bất động sản công ty | Admin, NhanVien |
|  | Quản lý tài khoản khách hàng | Thêm mới tài khoản khách hàng | Admin, NhanVien |
| Sửa thông tin tài khoản khách hàng | Admin, NhanVien |
| Xóa tài khoản | Admin, NhanVien |
|  | Quản lý tin đăng bất động sản Khách Hàng | Xem thông tin tin đăng | Admin, NhanVien |
| Xóa tin đăng | Admin, NhanVien |
|  | Quản lý tin tức | Thêm tin tức | Admin, NhanVien |
| Sửa thông tin tin tức | Admin, NhanVien |
| Xóa tin tức | Admin, NhanVien |
|  | Thống kê tin đăng Bất Động Sản công ty | Thống kê theo loại Bất Động Sản, Thống kê bất động sản theo Phường | Admin, NhanVien |
|  | Bản đồ | Hiển thị những tin đăng bất động sản trên bản đồ | Admin, NhanVien |
|  | Duyệt tin đăng | Xem và duyệt tin đăng của khách hàng | Admin, NhanVien |
|  | Xóa tin không hợp lệ | Admin, NhanVien |
|  | Quản lý tài khoản Admin | Thêm mới tài khoản | Admin |
| Sửa thông tin tài khoản | Admin |
| Xóa tài khoản | Admin |

## Phân quyền hệ thống

*Admin* : Người quản trị hệ thống, toàn quyền sử dụng các chức năng trong trang quản trị

*NhanVien* : Nhân viên quản trị hệ thống, sử dụng đủ các chức năng của trang quản trị, ngoại trừ chức năng quản lý Admin

*KhachHang* : Thao tác đầy đủ các chức năng trên trang User

# CHƯƠNG 2. PHÂN TÍCH HỆ THỐNG

## 2.1. Danh Sách Actor

Một actor hay tác nhân ngoài là một vai trò của một hay nhiều người hay vật thể trong sự tương tác với hệ thống.

Bảng 2.1 Danh sách Actor

|  |  |  |
| --- | --- | --- |
| **STT** | **Tá c nhân** | **Mô tả** |
| 1 | Quản tri ̣viên (Admin) | Có toàn quyền sử dụng trong hệ thống. |
| 2 | Nhân Viên(NhanVien) | Nhân viên đăng nhập sử dụng hệ thống, được phép sử dụng các chức năng như Quản lý tin đăng bất động sản công ty, tin đăng bất động sản người dùng, … |
| 3 | Khách Hàng(KhachHang) | Người dùng đã đăng ký tài khoản và được gọi là khách hàng. Được hưởng tất cả các chức năng trên trang user. |

## Danh sách các Use Case

Bảng 2.2 Danh sách các UseCase

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Usecase** | **Ý nghĩa** | **Chứ c năng** |
| 1 | TimKiem | Tìm kiếm tin rao theo những trường đơn giản, phổ biến nhất | Tìm kiếm |
| UC\_002 | TimKiemNangCao | Tìm kiếm tin rao theo nhiều trường chi tiết hơn, sát hơn so với yêu cầu cần tìm kiếm của người dùng | Tìm kiếm nâng cao |
|  |  |  |  |

## Biểu đồ UseCase

## Tìm hiểu Git

Khi bạn phát triển một phần mềm và bạn đang sửa một file nào đó nhưng bạn muốn quay lại trạng thái trước đó của file đang sửa thì sẽ làm như thế nào?

Cách đơn giản nhất đó là sao chép lại file trước khi chỉnh sửa. Trường hợp dùng phương pháp này thì sẽ phải thường xuyên thực hiện việc thêm ngày đã thay đổi vào tên thư mục hay file. Tuy nhiên, việc tự mình sao chép file mỗi lần chỉnh sửa thì sẽ rất vất vả, và cũng dễ xảy ra nhầm lẫn.

Hơn nữa, trường hợp đã đặt tên một cách vô trật tự như hình trên thì sẽ không thể phân biệt được file nào là bản mới nhất. Thêm nữa, với file đang chia sẻ và làm việc trong nhóm thì cũng sẽ có gắn tên của người chỉnh sửa. Nhưng lại không thể dễ dàng biết được đã thực hiện thay đổi như thế nào.

Và để giải quyết những vấn đề này thì các hệ thống quản lý phiên bản như Git được tạo ra.

Git là một phần mềm, hệ thống quản lý mã nguồn phân tán. Ban đầu Git được phát triển phục vụ cho Linux nhưng bây giờ bạn có thẻ sử dụng được nó trên cả Windows. Với Git bạn có thể lưu lại lịch sử và trạng thái của các file trong dự án, sau đó bạn có thể rollback về một lịch sử bất kì mà không cần phải backup lại.

### **Cài đặt và thiết lập ban đầu để làm việc với Git**

Nếu sử dụng máy tính sử dụng MacOs. Mặc định MacOs đã có sẵn Git nên không cần phải cài đặt, nếu dùng Window thì bạn có thể sử dụng GitBash.

**A screenshot of a cell phone

Description automatically generated**Để sử dụng command line trong Git Bash thì click chuột phải và chọn **Git Bash Here**

Hình II.2.2.1.1 Chọn Git Bash here

Sau đó cửa sổ Git Bash sẽ hiện ra. Và cần cấu hình tài khoản Git vào Git Bash.

Sử dụng các lệnh: **+ $ git config --global user.name “your\_name”**

A screenshot of a computer

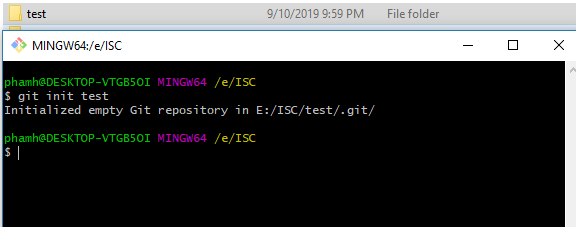
Description automatically generated **+ $ git config --global user.email** [**your\_email@youremail.com**](mailto:your_email@youremail.com)

Hình II.2.2.1.2 Cấu hình Tên và Email

### **Cách tạo một Repository**

Repository (kho chứa) nghĩa là nơi lưu trữ mã nguồn và một người khác có thể sao chép (clone) lại mã nguồn đó nhằm làm việc. Repository có hai loại là **Local Repository** (Kho chứa trên máy cá nhân) và **Remote Repository** (Kho chứa trên một máy chủ từ xa).

**Cách tạo Local Repository**

Ở thư mục chứa đồ án trên máy cá nhân, click chuột phải và chọn Git Bash Here, sau đó gõ lệnh: **git init tên đồ án.** Sau đó 1 thư mục sẽ được tạo và đó sẽ là nơi chứa code và những gì liên quan đến đồ án

Hình II.2.2.2.1 Tạo Local Repository

**Cách tạo Remote Repository**

A screenshot of a cell phone

Description automatically generatedTrước tiên cần đăng nhập vào Github, sau đó ấn vào dấu + trên menu và chọn **New repository** hoặc **New** ở bên trái màn hình.

Hình II.2.2.2.2 Tạo Remote Respository

A screenshot of a computer

Description automatically generated Sau đó chúng ta cần đặt tên trên **Repository** và **Description** và chọn **Public** (Ai cũng có thể tải về) hay **Private** (chỉ những người được cấp quyền mới có thể tải về)

Hình II.2.2.2.3 Điền thông tin tạo Remote Repository

Khi tạo xong nó sẽ dẫn bạn tới trang hướng dẫn làm việc với kho chứa vừa tạo. Và kho chứa của bạn bây giờ sẽ có địa chỉ là **https://github.com/$user-name/$repository**, ví dụ [**https://github.com/phamhieudev/test.git**](https://github.com/phamhieudev/test.git)

Chúng ta có thể sử dụng địa chỉ trên để **clone** (tải) về hoặc **push** (đẩy) lên khi cần.

### **Các lệnh cơ bản**

Tạo một kho chứa: **git init** *tên đồ án*

Sao chép một kho chứa đã tồn tại: **git clone** *link repository*

Cập nhật thay đổi file: **git add.**

Sau khi sử dụng lệnh add, chúng ta cần sử dụng lệnh Commit để đẩy thông tin thay đã thay đổi và cho người khác biết mình đã làm gì: **git -m commit** *“Message”*

Cập nhật lên Repository: **git push origin** *nhánh*

Xem lại lịch sử Commit: **git log**

### **Nhánh trong Git (Branch)**

Branch là cái dùng để phân nhánh và ghi lại luồng của lịch sử. Branch đã phân nhánh sẽ không ảnh hưởng đến branch khác nên có thể tiến hành nhiều thay đổi đồng thời trong cùng 1 repository.

Khi các thành viên làm việc chung một đồ án, họ có thể tạo ra những Branch cho công việc của mình từ branch chính để không ảnh hưởng đến công việc của các thành viên khác. Sau đó, những thành viên đã hoàn thành công việc của mình sẽ thực hiện đưa thay đổi của mình vào branch chính.

A close up of a map

Description automatically generatedBranch mặc định là Branch **master**

Hình II.2.2.4.1 Nhánh trong Git

**Các lệnh cơ bản để sử dụng Branch**

Kiểm tra branch hiện tại: **git branch**

Tạo mới một branch: **git branch** *<name\_branch>*

Chuyển branch: **git checkout** *<name\_branch>*

Sau một thời gian cập nhật các file và push lên git trên branch mới, bây giờ chúng ta cần ghép (merge) code lại vào nhánh gốc (master). Trước tiên, cần phải checkout ra khỏi branch hiện tại cần gộp để vào branch master, sau đó thì dùng lệnh merge để ghép branch mới vào master: **git checkout** *master* và **git merge** *<new\_branch>*

## Tìm hiểu Agile

Agile là tập hợp các nguyên lý tạo ra phương thức phát triển phần mềm linh hoạt với mục tiêu đưa sản phẩm đến tay người dùng càng nhanh càng tốt. Tiêu chí mà phương pháp Agile hướng đến: Lập kế hoạch thích ứng, phát triển tăng dần, sự chuyển giao sớm và cải tiến liên tục.

Mô hình Agile trong phát triển phần mềm là mô hình gia tăng, hoạt động trên quy trình phát triển lặp với những điểm cơ bản như:

* Phân chia dự án thành mảng nhỏ để dễ xử lý và thay đổi khi khách hàng yêu cầu thay đổi.
* Trong quá trình làm dự án thì từng phần nhỏ sẽ được test ngay.
* Nhóm dự án Agile yêu cầu cùng có mặt để trao đổi thường xuyên.

So với mô hình dự án truyền thống thực hiện theo kế hoạch thì mô hình Agile khuyến khích cải tiến để tạo ra bước đột phá riêng, hướng đến tạo ra phần mềm hoàn thiện nhất.

### **Tôn chỉ và các nguyên tắc trong Agile**

**4 tôn chỉ cần tuân thủ trong phương pháp Agile**

* “Cá nhân và sự tương hỗ quan trọng hơn quy trình và công cụ”: Trọng tâm đặt lên con người, xây dựng tương tác và hỗ trợ giữa các thành viên trong nhóm. Những thành viên có năng lực, chịu tương trợ nhau trong công việc sẽ mang đến thành công cho dự án.
* “Sản phẩm dùng được tốt hơn tài liệu đầy đủ”: Tập trung thời gian để làm ra phần mềm hoàn chỉnh đáp ứng hoàn hảo yêu cầu khách hàng.
* “Cộng tác với khách hàng quan trọng hơn đàm phán hợp đồng”: Hiểu được khách hàng cần gì để tư vấn và điều chỉnh sản phẩm thay vì chỉ dựa vào các điều khoản trong hợp đồng.
* “Phản hồi thay đổi hơn là bám sát kế hoạch”: Agile khuyến khích thích nghi với sự thay đổi, đó có thể là thay đổi về công nghệ, nhân sự, deadline,...

**12 nguyên tắc quan trọng trong Agile**

* Đáp ứng toàn diện nhu cầu khách hàng thông qua việc giao hàng sớm và sản phẩm có giá trị.
* Thay đổi yêu cầu được chào đón, thậm chí là rất muộn trong quá trình phát triển.
* Giao phần mềm chạy được cho khách hàng một cách thường xuyên.
* Nhà kinh doanh và các kỹ sư phần mềm cần làm việc cùng nhau trong suốt dự án.
* Xây dựng dự án xung quanh các cá nhân có động lực. Cung cấp sự hỗ trợ cần thiết, môi trường làm việc và niềm tin để hoàn thành công việc.
* Trao đổi trực tiếp là cách truyền đạt thông tin hiệu quả nhất.
* Thước đo chính của tiến độ là phần mềm chạy tốt.
* Phát triển liên tục và bền vững.
* Cải tiến sự linh hoạt bằng cách quan tâm đến kỹ thuật và thiết kế.
* Nghệ thuật tối đa hóa lượng công việc chưa xong - Sự đơn giản là cần thiết.
* Nhóm tự tổ chức
* Thích ứng thường xuyên với những thay đổi.

### **Quy trình Agile được thực hiện như thế nào?**

Có thể hiểu cách đơn giản, quy trình Agile là quy trình phát triển lặp. Mỗi dự án được phân chia thành các giai đoạn nhỏ, linh hoạt thay đổi khi có yêu cầu thay đổi từ khách hàng.

Trong từng giai đoạn, cứ mỗi giai đoạn nhỏ được giao cho khách hàng thì khách hàng sẽ đưa ra yêu cầu mới hoặc thay đổi để nhóm phát triển cập nhật sản phẩm mà không cần thực hiện lại từ đầu. Agile phát huy giá trị về sự linh hoạt và cải tiến để thích nghi với thay đổi.

Về cơ bản, quy trình Agile được phân chia thành các giai đoạn chính như sau:

* Lập kế hoạch
* Phân tích
* Thiết kế và lập trình
* Kiểm thử sản phẩm
* Bàn giao sản phẩm

Không chỉ giới hạn trong quy trình phát triển phần mềm, phương pháp Agile còn là sự thay thế tối ưu cho các mô hình quản lý công việc truyền thống. Các nhà quản lý ngoài ngành công nghệ thông tin bắt đầu sử dụng Agile trong quản lý các dự án như: Marketing, thiết kế, giáo dục, khởi nghiệp,...

Phương pháp Agile trong quản lý dự án giúp nâng cao hiệu suất công việc, hoàn thành nhịp nhàng và đúng tiến độ các phần việc để tạo ra những sản phẩm và dịch vụ chất lượng hơn, đáp ứng yêu cầu của khách hàng.

## Tìm hiểu Scrum

Scrum là một phương pháp Agile dùng cho phát triển sản phẩm, đặc biệt là phát triển phần mềm. Scrum là một khung quản lý dự án được áp dụng rất rộng rãi, từ những dự án đơn giản với một nhóm phát triển nhỏ cho đến những dự án có yêu cầu rất phức tạp với hàng trăm người tham gia, và kể cả những dự án đòi hỏi khung thời gian cố định. Trong Scrum, công việc được thực hiện bởi Nhóm Scrum thông qua từng phân đoạn lặp liên tiếp nhau được gọi là Sprint.

### **Vai trò trong Scrum**

Trong Scrum có 3 vai trò: Chủ sản phẩm (Product Owner), Nhóm phát triển, và Scrum Master. Trong Scrum bạn sẽ không có vai trò Quản lý dự án (Project manager) hay Trưởng nhóm kỹ thuật (Technical lead)

* **Chủ sản phẩm:** Là người chịu trách nhiệm cao nhất đối với sản phẩm và nhóm phát triển. Chủ sản phẩm có trách nhiệm làm việc với chủ đầu tư để hiểu yêu cầu về sản phẩm, quản lý những yêu cầu đó, tạo ra những “câu chuyện người dùng” đối với sản phẩm và truyền đạt những thông tin đó đến đội phát triển. Cơ bản là nếu nhóm gặp những vấn đề hay thắc mắc gì liên quan đến sản phẩm, hãy tìm gặp Chủ sản phẩm.
* **Nhóm phát triển:** Là một tập hợp những kỹ sư “liên chức năng”- nghĩa là công việc của họ không cố định ở lập trình, kiểm thử, phân tích hay thiết kế. Tùy theo yêu cầu công việc mà họ sẽ đảm nhận những vai trò tương ứng. Nhóm phát triển được quyền chủ động tổ chức công việc, ước lượng khối lượng công việc và cam kết hoàn thành công việc đã cam kết. Trong Sprint, nhóm phát triển có tiếng nói lớn nhất và những bộ phận khác có nhiệm vụ hỗ trợ những điều kiện tốt nhất để nhóm làm việc hiệu quả
* **Scrum Master:** Nhiệm vụ của Scrum Master là giúp mọi người trong nhóm hiểu được Scrum, làm theo Scrum đồng thời hỗ trợ nhóm phát triển để họ có thể toàn tâm toàn ý làm việc. Vừa giống ông chủ vừa giống osin, kiểu vậy. Nếu có ai đó thắc mắc về quy trình trong Scrum, ý nghĩa của Scrum hay những vấn đề liên quan đến Scrum khác, hãy tìm gặp Scrum Master.

### **Các sự kiện trong Scrum**

* Sprint: Sprint là những chu kỳ nhỏ để phát triển sản phẩm. Trong Sprint, nhóm sẽ tập trung phát triển những chức năng cụ thể nào đó và hoàn hiện nó vào cuối mỗi Sprint. Mỗi Sprint sẽ có thời gian cố định được thống nhất, thường là 2 tuần và thường không hơn 4 tuần.
* Họp kế hoạch Sprint (Sprint Planning Meeting): Diễn ra vào đầu mỗi Sprint bao gồm Chủ sản phẩm, Scrum Master và Nhóm phát triển. Chủ sản phẩm sẽ trình bày mục tiêu của Sprint (Sprint goal) và những đầu mục công việc có độ ưu tiên cao trong danh sách các đầu mục công việc (được gọi là Product Backlog) sau đó đội phát triển sẽ thảo luận (đặt câu hỏi, ước lượng độ lớn, định ra những công việc cần phải làm v.v). Những đầu mục công việc mà nhóm thống nhất sẽ làm trong Sprint sẽ được chuyển qua một danh sách công việc khác gọi là Sprint Backlog. Cơ bản, sau buổi họp kế hoạch Sprint, ta sẽ biết được Mục tiêu của Sprint và những việc đầu mục cần làm trong Sprint.
* Họp Scrum hằng ngày (Daily Scrum Meeting): Sau họp kế hoạch Sprint, nhóm sẽ bắt tay vào công việc phát triển và nhóm sẽ có cuộc họp ngắn vào mỗi đầu ngày. Buổi họp này thường diễn ra ngắn khoảng 15 phút và cố định về thời gian, địa điểm họp. Trong cuộc họp này, từng người trong nhóm phát triển lần lượt trình bày để trả lời 3 câu hỏi sau:

*Hôm qua đã làm gì?*

*Hôm nay sẽ làm gì?*

*Có khó khăn trở ngại gì không?*

Scrum Master là người chịu trách nhiệm giải quyết hoặc tìm giải pháp cho những khó khăn, trở ngại mà nhóm gặp phải và đảm bảo là nó được giải quyết sớm nhất có thể để nhóm có thể hoàn thành công việc.

* Họp sơ kết Sprint (Sprint Review Meeting): Vào cuối mỗi Sprint, nhóm sẽ trình bày những phần mình đã làm được trong Sprint hay còn gọi là demo trên sản phẩm thật. Thành phần tham dự là tất cả những ai quan tâm đến sản phẩm. Cuộc họp sẽ giúp đánh giá xem nhóm có đạt được mục tiêu đề ra ở buổi họp kế hoạch Sprint hay không.
* Họp cải tiến Sprint (Sprint Retrospective Meeting): Buổi họp này thường diễn ra ngay sau buổi họp sơ kết Sprint và mất tầm khoảng 1-2 giờ thảo luận. Trong buổi họp nhóm sẽ đánh giá những việc mình đã làm và cách để làm cho nó tốt hơn. Về cơ bản, buổi họp sẽ xoay quanh trả lời những câu hỏi:

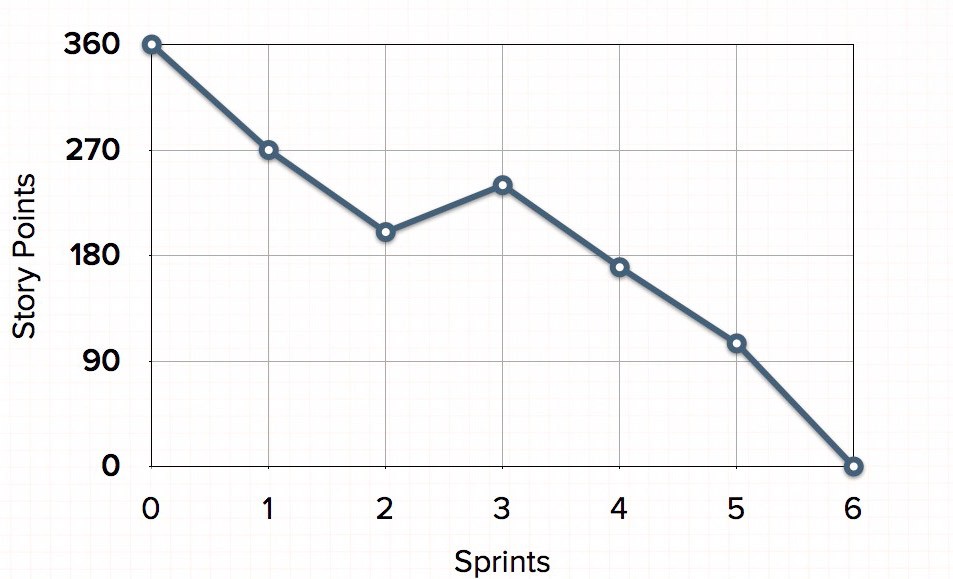
*Những việc nào chúng ta nên bắt đầu làm*

*Những việc nào chúng ta không nên làm tiếp*

*Những việc nào chúng ta nên duy trình làm tiếp*

Sau khi có được danh sách những việc “cần bắt đầu làm”, “không nên làm tiếp”, “nên duy trì làm tiếp”, nhóm sẽ thống nhất chọn ra vài thứ mà nhóm sẽ tập trung để cải tiến trong Sprint kế tiếp. Kết quả thực thi những cải tiến này sẽ được thảo luận trong buổi họp cải tiến của Sprint sau.

### **Đồ nghề trong Scrum**

* Product Backlog: Là một danh sách các đầu mục cần phải làm để phát triển sản phẩm bao gồm đủ loại như chức năng của sản phẩm, lỗi cần sửa, nghiên cứu công nghệ hay những việc công việc liên quan khác. Những đầu mục này thường được mô tả dưới dạng “câu chuyện người dùng” (user story).
* Sprint Backlog: Là một danh sách các đầu mục mà nhóm cam kết hoàn thành trong Sprint sau buổi họp sơ kết Sprint. Trong thời gian diễn ra Sprint, nhóm phát triển sẽ chỉ tập trung vào danh sách này và cập nhật tiến độ hoàn thành của chúng.
* Biểu đồ Burndown (Burndown chart): Được dùng để đo tiến độ của Sprint hay của dự án. Burndown sẽ cho thấy nhóm còn bao nhiêu thời gian để hoàn thành công việc đã được định ra lúc đầu.

Hình II.4.4.3.1 Biểu Đồ Burndown

* A screenshot of a cell phone

  Description automatically generatedBảng công việc: Được dùng để hiển thị những “câu chuyện người dùng” trong Sprint, những công việc nào cần làm cho “câu chuyện” đó, trạng thái của chúng cũng như những ghi chú cần thiết khác.

Hình II.4.4.3.2 Bảng công việc

## Tìm hiểu Spring MVC

### **Tổng quan về Spring MVC**

A screenshot of a cell phone

Description automatically generatedSpring MVC là một module con trong Spring framework, cung cấp kiến trúc Model-View-Controller và các components sẵn có để sử dụng và phát triển ứng dụng web một cách linh hoạt.

Hình II.5.5.1.1 Quy trình hoạt động của Spring MVC

* Model: đóng gói dữ liệu ứng dụng và bao gồm các POJO.
* View: Chịu trách nhiệm nhận giá trị của model và vẽ ra trang HTML mà trình duyệt có thể hiển thị được.
* Controller: Chịu trách nhiệm nhận và xử lý các yêu cầu từ người dùng và tạo các model phù hợp và trả về cho view. Spring MVC được thiết kế xung quanh DispatcherServlet để xử lý tất cả các HTTP request và HTTP response.

Spring MVC được thiết kế xung quanh DispatcherServlet để xử lý tất cả các HTTP request và HTTP response



Hình II.5.5.1.2 DispatcherServlet

Đây là chuỗi sự kiện tương ứng khi nhận một yêu cầu HTTP gửi đến DispatcherServlet:

1. Sau khi nhận một HTTP request, DispatcherServlet gửi yêu cầu đến HandlerMapping (một bản đồ cấu hình URL) để xác định controller nào sẽ xử lý yêu cầu này.
2. Controller sẽ nhận các request và gọi các phương thức dịch vụ (service methods) phù hợp dựa trên việc sử dụng các method GET/POST… Nếu yêu cầu đó cần truy xuất cơ sở dữ liệu thì Controller sẽ ủy nhiệm cho một business logic hay nhiều hơn một model để lấy thông tin và gửi dữ liệu về cho Controller. Lúc này Controller đóng gói mô hình dữ liệu và tên của một view sẽ được tải lên thành đối tượng ModelAndView và gửi trả về cho DispathcherServlet
3. DispatcherServlet gửi gói ModelAndView cho ViewResolver để tìm xem view (thông thường là một trang JSP, HTML) nào sẽ được load lên.
4. Sau khi hoàn tất, DispatcherServlet sẽ chuyển dữ liệu từ model đến view và cuối cùng hiển thị trên trình duyệt.

### **DispathcherServlet**

DispathcherServlet là thành phần tiếp nhận tất cả các request từ browser, điều khiển luồng xử lý và trung chuyển giữa các thành phần (components) trong MVC. Hơn thế nữa, nó còn tích hợp các Spring IoC container, cho phép ta sử dụng đến tất cả các tính năng của Spring.

A screenshot of a cell phone

Description automatically generatedDispatcherServlet thực sự là một Servlet (nó thừa kế từ lớp cơ sở HttpServlet), và được khai báo trong tập tin web.xml của ứng dụng web. Các requests muốn DispatcherServlet xử lý sẽ phải được ánh xạ bằng cách sử dụng một bộ ánh xạ URL trong cùng một tập tin web.xml. Dưới đây là ví dụ về khai báo và định nghĩa DispatcherServlet.

Hình II.5.5.2.1 Khai báo và định nghĩa DispatcherSerlvelt

Như ta thấy, tất cả các requests bắt đầu bằng **/music** sẽ được DispatcherServlet **‘music-manager’** xử lý. Khi DispatcherServlet được khởi tạo, Spring sẽ gọi file có tên **[tên-servlet]servlet.xml** trong thư mục **WEB-INF**. Với ví dụ trên, Spring sẽ gọi file **musicmanager-servlet.xml**.

### **Controllers**

Controllers là các thành phần được DispatcherServlet gọi để xử lý các nghiệp vụ business. Tất cả các Controller trong Spring đều implement từ Controller interface trong package org.springframework.web.servlet.mvc

A screenshot of a cell phone

Description automatically generatedController interface định nghĩa ra 1 phương thức chịu trách nhiệm xử lí 1 request và trả về model và view thích hợp. Ba khái niệm này chính là nền tảng của Spring MVC framework: Model – View – Controller.

Hình II.5.5.3.1 Controller Interface

Sau đây là các Controller sử dụng được trong Spring:

- SimpleFormController

- AbstractController

- AbstractCommandController

- CancellableFormController

- AbstractCommandController

- MultiActionController

- ParameterizableViewController

- ServletForwardingController

- ServletWrappingController   
- UrlFilenameViewController

Kể từ khi Spring 2.5 giới thiệu và đưa vào mô hình lập trình cho các MVC controller dựa trên các annotation như @Controller, @RequestMapping, @RequestParam, @PathVariable,…thì việc tạo ra một controller trở nên vô cùng đơn giản. Để tạo ra một controller chúng ta không cần kế thừa từ một lớp cơ sở hoặc thực thi bất kỳ một interface nào.

### **Handler Mapping**

Sử dụng Handler Mapping để có thể ánh xạ từng requests đến các handler thích hợp. Các chức năng cơ bản của một HandlerMapping là đưa ra một Handler Execution Chain (chuỗi hành động), và có thể chứa một số bộ chặn. Khi requests đến, các DispatcherServlet sẽ trao cho các HandlerMapping để nó kiểm tra và xuất ra HandlerExecutionChain thích hợp. Sau đó, các DispatcherServlet sẽ gọi handler và bộ chặn trong hệ thống (nếu có).

Một Handler Mapping đơn giản nhưng rất mạnh mẽ là Bean Name Url Handler Mapping, nó ánh xạ các HTTP requests đến tên của các beans, được định nghĩa trong bối cảnh ứng dụng web. Ví dụ chúng ta muốn cho phép một người dùng tạo một tài khoản, ở đây ta đã có một form controller thích hợp và một trang JSP để hiển thị.

Khi sử dụng Bean Name Url Handler Mapping, ta có thể ánh xạ HTTP requests với địa chỉ http://samples.com/editaccount vào form Controller như sau:

A screenshot of a cell phone

Description automatically generated

Hình II.5.5.4.1 Ánh xạ HTTP Request

Cơ chế handler mapping của Spring có khái niệm xử lý đánh chặn, nó có thể cực kỳ hữu ích khi chúng ta muốn áp dụng các chức năng cụ thể cho một request xác định, ví dụ kiểm tra điều kiện chẳng hạn.

### **ViewResolver**

Views trong Spring được gán “view name” và được phân giải bởi ViewResolver. Sau đây là các ViewResolver trong Spring:

- BeanNameViewResolver

- FreeMarkerViewResolver

- InternalResourceViewResolver

- JasperReportsViewResolver

- ResourceBundleViewResolver

- UrlBasedViewResolver

- VelocityLayoutViewResolver

- VelocityViewResolver

- XmlViewResolver

- XsltViewResolver

Ở đây xin trình bày Internal Resource View Resolver vì nó thường được sử dụng nhiều nhất. View Resolver này sẽ ánh xạ Logical name của Resource (ModelAndView object được trả về bởi Controller) đến một Physical View (myView.jsp,…).

A screenshot of a cell phone

Description automatically generatedChúng ta có controller như sau:

**Hình II.5.5.5.1 Controller Internal Resource**

A screenshot of a cell phone

Description automatically generatedChúng ta cần phải khai báo một Internal Resource View Resolver trong file web.xml như sau:

Hình II.5.5.5.2 Khai báo Internal Resource trong WEB.XML

Khi đó nếu như request của người dùng thỏa condition1(), thì view **“myView.jsp”** trong thư mục **/WEB-INF/views** sẽ được hiển thị, nếu không thì hoặc **“myView2.jsp”** hoặc **“myView3.jsp”** sẽ được hiển thị.

## Tìm hiểu MySQL

MySQL là chương trình dùng để quản lý hệ thống cơ sở dữ liệu (CSDL), trong đó CSDL là một hệ thống lưu trữ thông tin. được sắp xếp rõ ràng, phân lớp ngăn nắp những thông tin mà mình lưu trữ.

Vì thế, chúng ta có thể truy cập dữ liệu một cách thuận lợi, nhanh chóng. MySQL hỗ trợ đa số các ngôn như lập trình phổ biến hiện nay

**Tại sao lại sử dụng MySQL?**

*Khả năng mở rộng và tính linh hoạt:* Máy chủ cơ sở dữ liệu MySQL đáp ứng nhiều tính năng linh hoạt, nó có sức chứa để xử lý các ứng dụng được nhúng sâu với 1MB dung lượng để chạy kho dữ liệu khổng lồ lên đến hàng terabytes thông tin. Đặc tính đáng chú ý của MySQL là sự linh hoạt về flatform với tất cả các phiên bản của Windows, Unix và Linux đang được hỗ trợ.

*Hiệu năng cao:* Với kiến trúc storage-engine cho phép các chuyên gia cơ sở dữ liệu cấu hình máy chủ cơ sở dữ liệu MySQL đặc trưng cho các ứng dụng chuyên biệt. Dù ứng dụng là website dung lượng lớn phục vụ hàng triệu người/ngày hay hệ thống xử lý giao dịch tốc độ cao thì MySQL đều đáp ứng được khả năng xử lý khắt khe của mọi hệ thống.

*Tính sẵn sàng cao:* MySQL đảm bảo sự tin cậy và có thể sử dụng ngay. MySQL đưa ra nhiều tùy chọn có thể “mì ăn liền” ngay từ cấu hình tái tạo chủ/tớ tốc độ cao, để các nhà phân phối thứ 3 có thể đưa ra những điều hướng có thể dùng ngay duy nhất cho server cơ sở dữ liệu MySQL.

*Hỗ trợ giao dịch mạnh mẽ:* MySQL đưa ra một trong số những engine giao dịch cơ sở dữ liệu tốt nhất trên thị trường. Các đặc trưng bao gôm, khóa mức dòng không hạn chế, hỗ trợ giao dịch ACID hoàn thiện, khả năng giao dịch được phân loại và hỗ trợ giao dịch đa dạng (multi-version) mà người đọc không bao giờ cản trở cho người viết và ngược lại.

*Điểm mạnh của Web và Data Warehouse:* MySQL là nơi cho các website trao đổi thường xuyên bởi nó có engine xử lý tốc độ cao, khả năng chèn dữ liệu cực nhanh và hỗ trợ mạnh các chức năng chuyên dụng của web. Các tính năng này cũng được sử dụng cho môi trường lưu trữ dữ liệu mà MySQL tăng cường đến hàng terabyte cho các server đơn.

*Bảo vệ dữ liệu mạnh mẽ:* Việc quan trong của các doanh nghiệp là bảo mật dữ liệu, MySQL tích hợp các tính năng bảo mật an toàn tuyệt đối. Với việc xác nhận truy cập cơ sở dữ liệu, MySQL trang bị các kỹ thuật mạnh, chỉ có người sử dụng đã được xác nhận mới truy cập được vào cơ sở dữ liệu. SSH và SSL cũng được hỗ trợ để đảm bảo kết nối an toàn và bảo mật.

*Phát triển ứng dụng toàn diện:* MySQL trở thành cơ sở dữ liệu mã nguồn mở phổ biến nhất hiện nay một phần là do cung cấp hỗ trợ hỗn hợp cho bất cứ sự phát triển ứng dụng nào cần. Trong cơ sở dữ liệu, hỗ trợ có thể được tìm thấy trong các trigger, stored procedure, cursor, view, ANSI-standard SQL,… MySQL cũng cung cấp các bộ kết nối như: JDBC, ODBC,… để tất cả các form của ứng dụng sử dụng MySQL như một server quản lí dữ liệu được đề xuất hàng đầu.

*Quản lý dễ dàng:* Cài đặt MySQL khá nhanh và trung binh từ khi download phần mềm tới khi cài đặt thành công chỉ mất chưa đầy 15 phút. Cho dù flatform là Linux, Microsoft Windows, Macintosh hoặc Unix thì quá trình cũng diễn ra nhanh chóng. Khi đã cài đặt, tính năng quản lý như tự khởi động lại, tự động mở rộng không gian và cấu hình động sẵn sàng cho người quản trị cơ sở dữ liệu bắt đầu làm việc.

*Mã nguồn mở tự do và hỗ trợ 24/7:* Nhiều doanh nghiệp còn băn khoăn trong việc giao toàn bộ cho phần mềm mã nguồn mở bởi khó có thể tìm được hỗ trợ hay bảo mật an toàn phục vụ chuyên nghiệp. Với MySQL mọi sự cam kết đều rõ ràng, MySQL cam kết bồi thường khi gặp sự cố.

*Chi phí sở hữu thấp:* Sử dụng MySQL cho các dự án, doanh nghiệp đều nhận thấy sự tiết kiệm chi phí đáng kể. Người dùng MySQL cũng không mất nhiều thời gian để sữa chữa hoặc vấn đề thời gian chết.

## Tìm hiểu Hibernate Framework

Hibernate là một dịch vụ lưu trữ và truy vấn dữ liệu quan hệ mạnh mẽ và nhanh. Hibernate giúp bạn phát triển các class dùng để lưu trữ dữ liệu theo cách thức rất là hướng đối tượng: association, inheritance, polymorphism, composition và collections. Hibernate cho phép chúng ta thực hiện các câu truy vấn dữ liệu bằng cách sử dụng ngôn ngữ SQL mở rộng của Hibernate (HQL) hoặc là ngôn ngữ SQL nguyên thuỷ cũng như là sử dụng các API.

Không giống như các persistence layer khác, Hibernate không ẩn đi sức mạnh của ngôn ngữ SQL khỏi bạn mà Hibernate còn đảm bảo cho bạn việc bạn đầu tư vào công nghệ và tri thức cơ sở dữ liệu quan hệ là luôn luôn chính xác. Và điều quan trọng hơn nữa là Hibernate được license theo LGPL (Lesser GNU Public License). Theo đó, chúng ta có thể thoải mái sử dụng Hibernate trong các dự án open source hoặc các dự án thương mại (commercial).

**Các thành phần của Hibernate project:**

* Hibernate Core: Cung cấp các chức năng cơ bản của một persistence layer cho các ứng dụng Java với các APIs và hỗ trợ XML mapping metadata.
* Hibernate Annotations: Các map class với JDK 5.0 annotations, bao gồm Hibernate Validator.
* Hibernate EntityManager: Sử dụng EJB 3.0 EntityManager API trong JSE hoặc với bất kỳ JEE server nào.
* Hibernate Tools: Các tool tích hợp với Eclipse và Ant dùng cho việc sinh ra các persistence object từ một schema có sẵn trong database (reverse-engineering) và từ các file hbm sinh ra các class java để hiện thực các persistence object, rồi Hibernate tự tạo các object trong database (forward-engineering).
* NHibernate: Hibernate cho .Net framework.
* JBoss Seam: Một Java EE 5.0 framework cho phát triển các ứng dụng JSF, Ajax và EJB 3.0 với sự hỗ trợ của Hibernate. Seam hiện rất mới và tỏ ra rất mạnh để phát triển các ứng dụng Web 2.0. Nó tích hợp đầy đủ tất cả các công nghệ “hot” nhất hiện nay.

## Tìm hiểu ORM

ORM (Object Relational Mapping), là một kỹ thuật - cơ chế lập trình thực hiện ánh xạ Cơ Sở Dữ Liệu sang các đối tượng trong các ngôn ngữ lập trình hướng đối tượng như Java, C# …

Việc sử dụng ORM cho phép lập trình viên thao tác với database một cách hoàn toàn tự nhiên, dễ hiểu thông qua các đối tượng. Lập trình viên không cần tới loại database, kiểu dữ liệu trong database…

**Ưu nhược điểm của ORM**

**Ưu điểm**

* OOP: ORM giúp lập trình viên tập trung vào lập trình hướng đối tượng
* Tính độc lập: Làm việc được với nhiều loại database(hệ quản trị cơ sở dữ liệu), nhiều kiểu dữ liệu khác nhau. Dễ dàng thay đổi loại database hơn. Các câu lệnh SQL không phụ thuộc vào loại database.
* Đơn giản, dễ sử dụng: Hỗ trợ HSQL, cung cấp nhiều nhiều API truy vấn.
* Năng suất hơn: viết code ít hơn, dễ hiểu hơn. Phù hợp các case CRUD (Create, Read, Update, Delete)
* Khả năng sử dụng lại code.

**Nhược điểm**

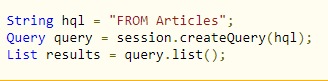
* Khả năng truy vấn bị hạn chế, nhiều trường hợp ta vẫn phải dùng native SQL để truy vấn database.
* Khó tối ưu câu lệnh SQL (do câu lệnh SQL được ORM tự động sinh ra).

## Tìm hiểu HQL

Tương tự như ngôn ngữ truy vấn SQL (Structured Query Language), Hibernate Query Language (HQL) được thiết kế nhằm mục đính tạo các truy vấn vào cơ sở dữ liệu và thao tác với chúng. Tuy nhiên, thay vì thao tác với cột, hàng như SQL, ngôn ngữ HQL xem các bảng như một đối tượng (lớp/class) và các cột của chúng là những thuộc tính. Do đó, HQL được coi là ngôn ngữ truy vấn hướng đối tượng.

Việc sử dụng HQL không quá phức tạp, do những câu truy vấn được viết bằng HQL sẽ được biên dịch bởi Hibernate và chuyển thành các câu truy vấn SQL thông thường để giao tiếp với cơ sở dữ liệu. Dưới đây là những từ khóa, mệnh đề thường được sử dụng trong HQL.

**Những mệnh đề cơ bản:**

**From:** Khi giao tiếp với cơ sở dữ liệu, mỗi bảng (Table) sẽ được HQL coi là một đối tượng và những thuộc tính là các cột. Chúng ta có thể lấy toàn bộ dữ liệu từ bảng Articles với truy vấn như sau.

Hình II.9.1 Mệnh đề Form

A picture containing object

Description automatically generatedBên cạnh đó, chúng ta có thể khai báo rõ ràng package class.

Hình II.9.2 Mệnh đề Form

Lưu ý, các mệnh đề Select, From,.. có thể viết hoa thường tùy ý, tuy nhiên tên lớp hay package phải viết đúng như khai báo.

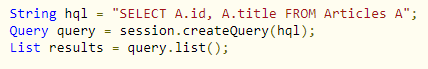
A picture containing object

Description automatically generated **AS**: Trong SQL, mệnh đề AS được dùng để gán tên lớp với tên khai báo khác ngắn hơn, giúp tiết kiệm thời gian khi viết các câu truy vấn phức tạp. Chức năng này của AS cũng tương tự trong HQL.

Hình II.9.3 Mệnh đề AS

**SELECT:** Mệnh đề thông dụng nhất trong việc truy vấn dữ liệu, Select cho phép chúng ta khai báo rõ ràng hơn việc sẽ chọn những thuộc tính, cột nào trong class.

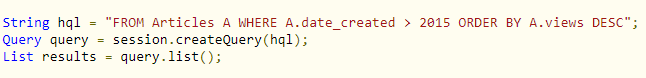
Hình II.9.4 Mệnh đề Select

****

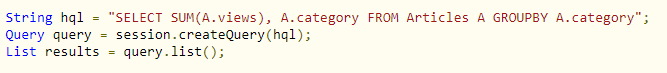
**A screenshot of a cell phone

Description automatically generatedWHERE:** Mệnh đề Where được dùng để thêm các điều kiện, giúp xác định được đối tượng cần truy vấn trong cơ sở dữ liệu.

Hình II.9.5 Mệnh đề Where

**ORDER BY:** Sau khi câu truy vấn đã tìm được những đối tượng phù hợp với yêu cầu, chúng ta có thể sắp xếp các đối tượng tăng dần hay giảm dần theo thuộc tính nào đó.

Hình II.9.6 Mệnh đề Order By

**GROUP BY:** Group by cho phép chúng ta gộp các đối tượng dựa trên một thuộc tính nào đó. Câu truy vấn có mệnh đề này thường sẽ sử dụng thêm các phương thức tính toán như SUM, COUNT,...

Hình II.9.7 Mệnh đề Group By

**A screenshot of a cell phone

Description automatically generatedMệnh Đề UPDATE, DELETE**

Hình II.9.8 Mệnh đề Update

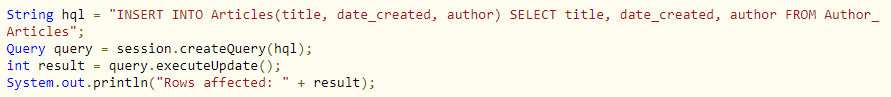
Trong đó, executeQuery() chỉ thực hiện câu lệnh Update thông tin đối tượng nên kết quả trả về sẽ là số dòng bị thay đổi trong lớp.

A screen shot of a social media post

Description automatically generatedCâu lệnh Delete cũng được thực hiện tương tự.

Hình II.9.9 Mệnh đề Delete

Lưu ý, việc Update hay Delete đều được coi là làm thay đổi cơ sở dữ liệu, do đó ta đều dùng hàm executeUpdate.

**Mệnh Đề INSERT:** Hibernate không hỗ trợ Insert theo cách thông thường như các câu lệnh SQL (insert into TABLE values..). Chúng ta chỉ có thể Insert đối tượng từ một lớp khác đã có sẵn.

Hình II.9.10 Mệnh đề Insert

Câu lệnh trên sẽ Insert các đối tượng articles từ lớp Author\_Articles (có sẵn) vào lớp Articles.

A close up of a logo

Description automatically generated**Thêm Đối Tượng Vào Lớp:** Hibernate chỉ hỗ trợ chúng ta Insert đối tượng từ một lớp có sẵn, do đó để thêm mới một đối tượng, chúng ta sẽ sử dụng **session.save()**.

Hình II.9.11 Mệnh đề Thêm Đối Tượng

**Các hàm tính toán:** Ngôn ngữ truy vấn Hibernate sử dụng các hàm tính toán tương tự như SQL để thao tác với các đối tượng.

* AVG (property): tính trung bình các giá trị theo thuộc tính
* COUNT (property): đếm số lần xuất hiện của giá trị
* MAX (property): lấy giá trị lớn nhất theo thuộc tính
* MIN (property): lấy giá trị nhỏ nhất theo thuộc tính
* SUM (property): tính tổng các giá trị theo thuộc tính

# CHƯƠNG III. PHẦN THỰC HÀNH

## Phân tích hệ thống

### **Xác định các tác nhân**

* Tác nhân độc giả
* Tìm kiếm sách trên Website.
* Tác nhân người quản trị
* Thêm, xóa, sửa sách
* Thêm, xóa, sửa nhà xuất bản
* Thêm, xóa, sửa thể loại
* Thêm, xóa, sửa tác giả
* Thêm, xóa, sửa độc giả
* Quản lý mượn, trả
* Thống kê.

### **Biểu đồ UC tổng quát**

Hình III.1.1.2.1 Biểu đồ USE CASE tổng quát

### **Đặc tả các UC**

* + 1. Tím kiếm sách trên website

a. Tác nhân: Độc giả

b. Mô tả: cho phép độc giả tìm kiếm sách trên thư viện.

c. Tiền điều kiện: độc giả truy cập Website.

d. Luồng sự kiện:

- Độc giả truy cập Website.

- Nhập tên sách cần tìm vào Form tìm.

- Nhấn submit.

- Hệ thống chuyển sang trang kết quả tìm kiếm.

- UC kết thúc.

e. Hậu điều kiện: Kết quả tìm kiếm sách.

* + 1. Thêm, xóa, sửa sách

a. Tác nhân: quản trị.

b. Mô tả: UC cho phép quản trị quản lý sách

c. Tiền điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Chọn mục quản lý Sách.

- Quản trị viên chọn chức năng thêm, sửa, xóa nhà xuất bản.

- Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa, sửa).

- Quản trị thao tác.

- Nhấn nút hoàn thành thao tác.

- Hệ thống kiểm tra và lưu trữ thông tin.

- UC kết thúc.

e. Hậu điều kiện:

* + 1. *Thêm, xóa, sửa nhà xuất bản*

a. Tác nhân: quản trị hệ thống

b. Mô tả: UC cho phép quản trị thêm, xóa, sửa nhà xuất bản.

c. Tiền Điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Chọn mục quản lý nhà xuất bản

- Quản trị viên chọn chức năng thêm, sửa, xóa nhà xuất bản.

- Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa, sửa).

- Quản trị thao tác.

- Nhấn nút hoàn thành thao tác.

- Hệ thống kiểm tra và lưu trữ thông tin.

- UC kết thúc.

* + 1. *Thêm, xóa, sửa thể loại*

a. Tác nhân: quản trị hệ thống

b. Mô tả: UC cho phép quản trị thêm, xóa, sửa thể loại.

c. Tiền Điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Chọn mục quản lý thể loại

- Quản trị viên chọn chức năng thêm, sửa, xóa thể loại.

- Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa, sửa).

- Quản trị thao tác.

- Nhấn nút hoàn thành thao tác.

- Hệ thống kiểm tra và lưu trữ thông tin.

- UC kết thúc.

* + 1. *Thêm, xóa, sửa tác giả*

a. Tác nhân: quản trị hệ thống

b. Mô tả: UC cho phép quản trị thêm, xóa, sửa tác giả.

c. Tiền Điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Chọn mục quản lý tác giả

- Quản trị viên chọn chức năng thêm, sửa, xóa tác giả.

- Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa, sửa).

- Quản trị thao tác.

- Nhấn nút hoàn thành thao tác.

- Hệ thống kiểm tra và lưu trữ thông tin.

- UC kết thúc.

* + 1. *Thêm, xóa, sửa độc giả*

a. Tác nhân: quản trị hệ thống

b. Mô tả: UC cho phép quản trị thêm, xóa, sửa tác giả.

c. Tiền Điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Chọn mục quản lý độc giả

- Quản trị viên chọn chức năng thêm, sửa, xóa độc giả.

- Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa, sửa).

- Quản trị thao tác.

- Nhấn nút hoàn thành thao tác.

- Hệ thống kiểm tra và lưu trữ thông tin.

- UC kết thúc.

* + 1. *Quản lý mượn*

a. Tác nhân: quản trị hệ thống

b. Mô tả: UC cho phép quản trị tạo phiếu mượn.

c. Tiền Điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Chọn mục quản lý mượn

- Quản trị viên nhập vào độc giả mượn và sách được mượn, ngày trả.

- Nhấn nút mượn sách

- Hệ thống kiểm tra và lưu trữ thông tin.

- UC kết thúc.

* + 1. *Quản lý trả*

a. Tác nhân: quản trị hệ thống

b. Mô tả: UC cho phép quản trị trả sách.

c. Tiền Điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Chọn mục quản lý trả

- Quản trị viên nhập vào mã sách được trả

- Nhấn nút trả sách

- Hệ thống kiểm tra và lưu trữ, cập nhật lại Cơ Sở Dữ Liệu.

- UC kết thúc.

* + 1. Chức năng thống kê khảo sát

a. Tác nhân: quản trị hệ thống.

b. Mô tả: UC cho phép quản trị thống kê sách đã mượn theo ngày, tháng

c. Tiền Điều kiện: quản trị truy cập vào site hệ thống.

d. Luồng sự kiện:

- Quản trị viên chọn chức năng thống kê.

- Chọn từ ngày, tháng đến ngày, tháng cần xem

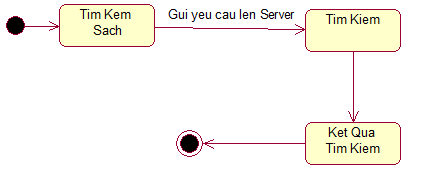
- Hệ thống hiển thị danh sách cho quản trị chọn để thống kê.

- UC kết thúc.

### **Biểu đồ hoạt động**

* + 1. *Tìm kiếm Sách*

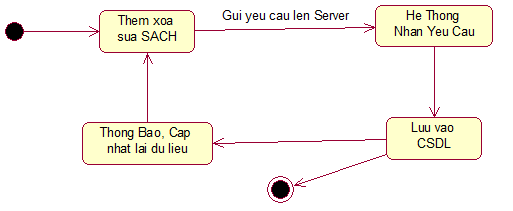
|  |  |
| --- | --- |
| **Tên** | Tìm kiếm sách |
| **Actor** | Độc giả |
| **Yêu cầu** | Truy cập vào site client |
| **Luồng dữ liệu** | Bước 1: Nhập tên sách cần tìm vào form nhập  Bước 2: Submit lên server |
| **Kết quả** | Thành công: Trả về dữ liệu, thông tin sách. |



Hình III.1.1.4.1 Biểu đồ hoạt động của UC Tìm Kiếm Sách

* + 1. Thêm, xóa, sửa sách

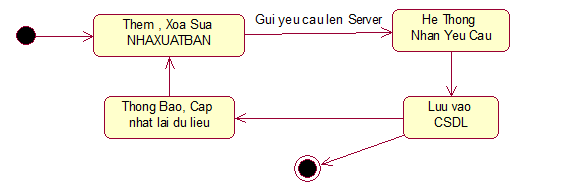
|  |  |
| --- | --- |
| **Tên** | Thêm, xóa, sửa sách |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1: Quản trị viên chọn chức năng thêm, sửa , xóa sách.  Bước 2: Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa , sửa).  Bước 3: Quản trị thao tác |
| **Kết quả** | Thành công: Hiển thị thông tin sách. |



Hình III.1.1.4.2 Biểu đồ hoạt động của UC Them, Xoa, Sua SACH

* + 1. Thêm, xóa, sửa nhà xuất bản

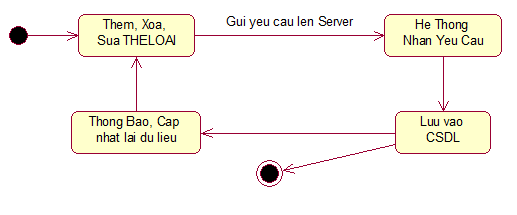
|  |  |
| --- | --- |
| **Tên** | Thêm, xóa, sửa nhà xuất bản |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1: Quản trị viên chọn chức năng thêm, sửa , xóa nhà xuất bản.  Bước 2: Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa , sửa).  Bước 3: Quản trị thao tác |
| **Kết quả** | Thành công: Hiển thị thông tin nhà xuất bản |



Hình III.1.1.4.3 Biểu đồ hoạt động của UC Them, Xoa, Sua NHAXUATBAN

* + 1. *Thêm, xóa, sửa thể loại*

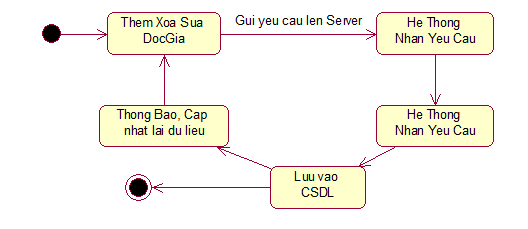
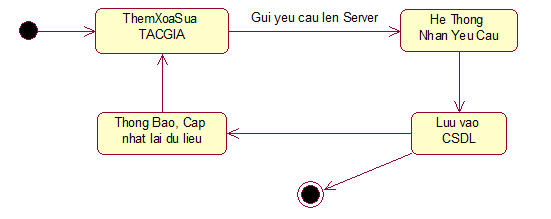
|  |  |
| --- | --- |
| **Tên** | Thêm, xóa, sửa thể loại |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1: Quản trị viên chọn chức năng thêm, sửa , xóa thể loại.  Bước 2: Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa , sửa).  Bước 3: Quản trị thao tác |
| **Kết quả** | Thành công: Hiển thị thông tin thể loại sách |



Hình III.1.1.4.4 Biểu đồ hoạt động của UC Them, Xoa, Sua THELOAI

* + 1. *Thêm, xóa, sửa tác giả*

|  |  |
| --- | --- |
| **Tên** | Thêm, xóa, sửa tác giả |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1: Quản trị viên chọn chức năng thêm, sửa, xóa tác giả.  Bước 2: Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa , sửa).  Bước 3: Quản trị thao tác |
| **Kết quả** | Thành công: Hiển thị thông tin tác giả |

* + 1. *Thêm, xóa, sửa độc giả*

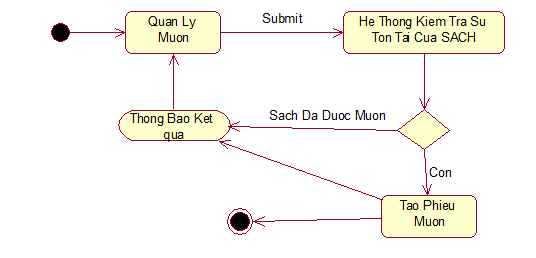
Hình III.1.1.4.5 Biểu đồ hoạt động của UC Them, Xoa, Sua TACGIA

|  |  |
| --- | --- |
| **Tên** | Thêm, xóa, sửa độc giả |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1: Quản trị viên chọn chức năng thêm, sửa, xóa độc giả.  Bước 2: Hệ thống hiển thị thông tin tùy theo chức năng mà quản trị chọn (thêm, xóa , sửa).  Bước 3: Quản trị thao tác |
| **Kết quả** | Thành công: Hiển thị thông tin độc giả |

Hình III.1.1.4.6 Biểu đồ hoạt động của UC Them, Xoa, Sua DOCGIA

* + 1. Quản lý mượn

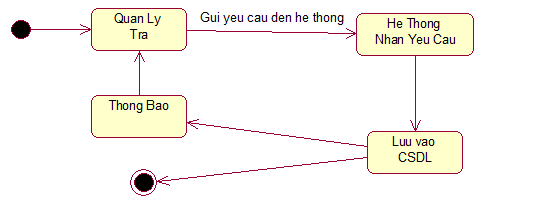
|  |  |
| --- | --- |
| **Tên** | Quản lý mượn |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1: Quản trị viên chọn chức năng quản lý mượn.  Bước 2: Nhập vào thông tin độc giả , mã sách mượn, ngày trả  Bước 4: Nhấn chọn mượn sách  Bước 3: Hệ thống kiểm tra sách còn hay không |
| **Kết quả** | Thành công: Hiển thị kết quả |



Hình III.1.1.4.7 Biểu đồ hoạt động của UC QuanLy Muon

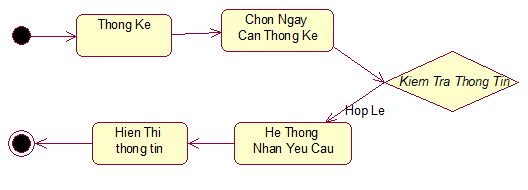
* + 1. Quản lý trả

|  |  |
| --- | --- |
| **Tên** | Quản lý trả |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1: Quản trị viên chọn chức năng quản lý trả.  Bước 2: Nhập vào thông tin mã sách được trả  Bước 3: Nhấn chọn trả sách.  Bước 4: Hệ thống cập nhật lại dữ liệu |
| **Kết quả** | Thành công: Hiển thị kết quả |

* + 1. Thống kê

Hình III.1.1.4.8 Biểu đồ hoạt động của UC QuanLy Tra

|  |  |
| --- | --- |
| **Tên** | Thống kê |
| **Actor** | Quản trị viên |
| **Yêu cầu** | Đã đăng nhập hệ thống |
| **Luồng dữ liệu** | Bước 1:Quản trị viên chọn chức năng thống kê.  Bước 2:Chọn từ ngày, tháng đến ngày, tháng cần xem.  Bước 3: Hiển thị thông tin. |
| **Kết quả** | Thành công: Hiển thị thông tin sách sau khi lọc theo ngày tháng. |



Hình III.1.1.4.9 Biểu đồ hoạt động của UC ThongKe

## Khởi tạo server với Spring Boot,TomCat

REST là viết tắt của Representational State Tranfer. REST là một chuẩn web dựa vào các kiến trúc cơ bản sử dụng giao thức HTTP. Web Server được thiết kế sử dụng công nghệ Spring Boot bao gồm:

* **GET** - Được sử dụng chỉ để đọc các nguồn tài nguyên.
* **PUT** - Được sử đụng để tạo mới các nguồn tài nguyên.
* **DELETE** - Được sử dụng để xóa các nguồn tài nguyên.
* **POST** - Được sử dụng để cập nhật các bản ghi hiện tại và tạo mới nguồn tài nguyên.

### **Thiết kế cơ sở dữ liệu**

Bảng DAUSACH: Lưu trữ thông tin của Đầu Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_ISBN ( khóa chính ) | bigint |  |
| ID\_LOAISACH ( khóa ngoại ) | bigint |  |
| ID\_NXB ( khóa ngoại ) | bigint |  |
| TENDS | nvarchar(50) |  |
| TOMLUOCNOIDUNG | nvarchar(300) |  |
| KHOSACH | nchar(20) |  |
| SOTRANG | int |  |
| DINHKEM | nvarchar(300) |  |
| VITRI | nchar(20) |  |
| NGONNGU | nvarchar(50) |  |
| PHIENBAN | int |  |
| NAMXUATBAN | date |  |
| HINHANH | varchar(50) |  |

Bảng SACH: Lưu trữ thông tin của Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| BARCODE ( khóa chính ) | bigint | Tự động tăng |
| ID\_ISBN ( khóa ngoại ) | bigint |  |
| TRANGTHAI | nvarchar(20) |  |

Bảng DOCGIA: Lưu trữ thông tin của Độc Gỉa

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_THEDG ( khóa chính ) | bigint | Tự động tăng |
| HOTENDG | bigint |  |
| SDTDG | nvarchar(20) |  |
| EMAILDG | nvarchar(50) |  |
| DIACHIDG | nvarchar(50) |  |
| NGAYSINHDG | date |  |
| GIOITINHDG | nvarchar(3) |  |
| HSDTHE | date |  |
| USERNAME\_DG | nvarchar(50) |  |
| PASSWORD\_DG | nvarchar(50) |  |

Bảng DS\_TG: Lưu trữ mã Tác Giả tương ứng với Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_ISBN ( khóa chính ) | bigint |  |
| ID\_TACGIA ( khóa chính ) | bigint |  |

Bảng LOAISACH: Lưu trữ thông tin Loại Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_LOAISACH ( khóa chính ) | bigint | Tự động tăng |
| TENLOAISACH | nvarchar(50) |  |

Bảng NXB: Lưu trữ thông tin Nhà Xuất Bản

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_NXB ( khóa chính ) | bigint | Tự động tăng |
| TENNXB | nvarchar(50) |  |
| DIACHINXB | nvarchar(50) |  |
| SDTNXB | varchar(10) |  |
| EMAILNXB | nvarchar(50) |  |

Bảng PHIEUMUON: Lưu trữ thông tin Phiếu Mượn Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_PHIEUMUON ( khóa chính ) | bigint | Tự động tăng |
| ID\_THUTHU ( khóa ngoại ) | bigint |  |
| ID\_THEDG ( khóa ngoại ) | bigint |  |
| NGAYLAPPM | date |  |

Bảng PHIEUTRA: Lưu trữ thông tin Phiếu Trả Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_PHIEUMUON ( khóa chính ) | bigint | Tự động tăng |
| ID\_THUTHU ( khóa ngoại ) | bigint |  |
| ID\_THEDG ( khóa ngoại ) | bigint |  |
| NGAYLAPPT | date |  |

Bảng PM\_S: Lưu trữ thông tin Chi Tiết Phiếu Mượn Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_PHIEUMUON ( khóa chính ) | bigint |  |
| BARCODE ( khóa chính ) | bigint |  |
| NGAYHUATRA | date |  |

Bảng PT\_S: Lưu trữ thông tin Chi Tiết Phiếu Trả Sách

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_PHIEUTRA ( khóa chính ) | bigint |  |
| BARCODE ( khóa chính ) | bigint |  |
| NGAYTRA | date |  |

Bảng TACGIA: Lưu trữ thông tin Tác Giả

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_TACGIA ( khóa chính ) | bigint | Tự động tăng |
| HOTENTG | nvarchar(50) |  |
| SDTTG | varchar(10) |  |
| EMAILTG | nvarchar(50) |  |
| TRINHDO | nvarchar(50) |  |

Bảng THUTHU: Lưu trữ thông tin Thủ Thư

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu dữ liệu | **Mô tả** |
| ID\_THUTHU ( khóa chính ) | bigint |  |
| HOTENTT | nvarchar(50) |  |
| SDTTT | varchar(10) |  |
| EMAILTT | nvarchar(50) |  |
| DIACHITT | nvarchar(50) |  |
| NGAYSINHTT | date |  |
| GIOITINHTT | nvarchar(3) |  |
| USERNAME\_TT | nvarchar(50) |  |
| PASSWORD\_TT | nvarchar(50) |  |

### **Các Url thao tác với dữ liệu**

Web Services chạy localhost Web Server tại **http://localhost:8080**. RESTful cung cấp các url để thao tác với dữ liệu và trả về dữ liệu kiểu JSON để có thể dễ dàng sử dụng với Angular JS. Các Url được cung cấp bao gồm:

Bảng mô tả Url ControllerBookTitle

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/bookTitle** | GET | Trả về danh sách tất cả các Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/createBookTitle** | POST | Thêm Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/updateBookTitle/{id\_ISBN}** | PUT | Sửa thông tin Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/deleteBookTitle/{id\_ISBN}** | DELETE | Xóa Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/bookTitleDetail/{id\_ISBN}** | GET | Hiển thị chi tiết Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/searchByContain/{name}** | GET | Tìm kiếm Đầu Sách |

Bảng mô tả Url ControllerBook

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/book** | GET | Trả về danh sách tất cả các Sách |
| [**http:/**](http://somee.com)**/localhost:8080/createBook** | POST | Thêm Sách |
| [**http:/**](http://somee.com)**/localhost:8080/updateBook/{** **barcode}** | PUT | Sửa thông tin Sách |
| [**http:/**](http://somee.com)**/localhost:8080/deleteBook/{barcode}** | DELETE | Xóa Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/bookByISBN/{id\_ISBN}** | GET | Hiển thị Sách dựa trên mã Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/bookByBarcode/{barcode}** | GET | Tìm kiếm Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/booksByStatus/{trangThai}** | GET | Hiển thị sách dựa trên Trạng Thái |
| [**http:/**](http://somee.com)**/localhost:8080/checkExistBarcodeByStatus/{barcode}** | GET | Kiểm tra trạng thái Sách |

Bảng mô tả Url ControllerAuthor

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/author** | GET | Trả về danh sách tất cả các Đầu Sách |
| [**http:/**](http://somee.com)**/localhost:8080/createAuthor** | POST | Thêm Tác Giả |
| [**http:/**](http://somee.com)**/localhost:8080/updateAuthor/{id\_TG}** | PUT | Sửa thông tin Tác Giả |
| [**http:/**](http://somee.com)**/localhost:8080/deleteAuthor/{id\_TG}** | DELETE | Xóa Tác Giả |

Bảng mô tả Url BorrowBooksController

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/borrowBooks** | GET | Trả về danh sách tất cả các Phiếu Mượn |
| [**http:/**](http://somee.com)**/localhost:8080/createBorrowBooks** | POST | Thêm Phiếu Mượn |
| [**http:/**](http://somee.com)**/localhost:8080/updateBorrowBooks/{id\_phieuMuon}** | PUT | Sửa thông tin Phiếu Mượn |
| [**http:/**](http://somee.com)**/localhost:8080/deleteBorrowBooks/{id\_phieuMuon}** | DELETE | Xóa Phiếu Mượn |
| [**http:/**](http://somee.com)**/localhost:8080/borrowBooksIdByReader/{id\_TheDG}** | GET | Hiển thị Phiếu Mượn theo Mã Độc Giả |

Bảng mô tả Url BorrowBooksDetailController

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/borrowBooksDetail** | GET | Trả về danh sách tất cả các Chi Tiết Phiếu Mượn |
| [**http:/**](http://somee.com)**/localhost:8080/createBorrowBooksDetail** | POST | Thêm Chi Tiết Phiếu Mượn |
| [**http:/**](http://somee.com)**/localhost:8080/** **updateBorrowBooksDetail/{id\_PhieuMuon}&{barcode}** | PUT | Sửa thông tin Chi Tiết Phiếu Mượn |
| [**http:/**](http://somee.com)**/localhost:8080/borrowBooksDetailById/{id\_PhieuMuon}** | GET | Hiển thị Chi Tiết Phiếu Mượn theo ID Phiếu |
| [**http:/**](http://somee.com)**/localhost:8080/borrowBooksDetailByManyId/{id\_PhieuMuon}** | GET | Hiển thị Chi Tiết Phiếu Mượn theo nhiều ID Phiếu |
| [**http:/**](http://somee.com)**/localhost:8080/borrowBooksDetailByBarcode/{barcode}** | GET | Hiển thị Chi Tiết Phiếu Mượn theo Mã Sách |
| **http://localhost:8080/readerByBarcode/{barcode}** | GET | Hiển thị Độc Giả Mượn theo Mã Sách |
| [**http:/**](http://somee.com)**/localhost:8080/deleteBorrowBooksDetail/{id\_PhieuMuon}&{barcode}** | DELETE | Xóa Chi Tiết Phiếu Mượn |

Bảng mô tả Url ControllerLibrarian

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/** **librarians** | GET | Trả về danh sách tất cả các Thủ Thư |

Bảng mô tả Url ControllerPublisher

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/publisher** | GET | Trả về danh sách tất cả các Nhà Xuất Bản |
| [**http:/**](http://somee.com)**/localhost:8080/createPublisher** | POST | Thêm Nhà Xuất Bản |
| [**http:/**](http://somee.com)**/localhost:8080/updatePublisher/{id\_NXB}** | PUT | Sửa thông tin Nhà Xuất Bản |
| [**http:/**](http://somee.com)**/localhost:8080/deletePublisher/{id\_NXB}** | DELETE | Xóa Nhà Xuất Bản |
| [**http:/**](http://somee.com)**/localhost:8080/publisher/{id\_NXB}** | GET | Hiển thị thông tin Nhà Xuất Bản theo Mã Nhà Xuất Bản |

Bảng mô tả Url ControllerReader

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/reader** | GET | Trả về danh sách tất cả các Độc Giả |
| [**http:/**](http://somee.com)**/localhost:8080/createReader** | POST | Thêm Độc Giả |
| [**http:/**](http://somee.com)**/localhost:8080/updateReader/{id\_THEDG}** | PUT | Sửa thông tin Độc Giả |
| [**http:/**](http://somee.com)**/localhost:8080/deleteReader/{id\_DG}** | DELETE | Xóa Độc Giả |
| [**http:/**](http://somee.com)**/localhost:8080**/**thongKeDG** | GET | Thống Kê Độc Giả |

Bảng mô tả Url ReturnBooksController

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/returnBooks** | GET | Trả về danh sách tất cả các Phiếu Trả |
| [**http:/**](http://somee.com)**/localhost:8080/createReturnBooks** | POST | Thêm Phiếu Trả |
| [**http:/**](http://somee.com)**/localhost:8080/updateReturnBooks/{id\_PhieuTra}** | PUT | Sửa thông tin Phiếu Trả |
| [**http:/**](http://somee.com)**/localhost:8080/deleteReturnBooks/{id\_PhieuTra}** | DELETE | Xóa Phiếu Trả |

Bảng mô tả Url ReturnBooksDetailController

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/returnBooksDetail** | GET | Trả về danh sách tất cả các Chi Tiết Phiếu Trả |
| [**http:/**](http://somee.com)**/localhost:8080/createReturnBooksDetail** | POST | Thêm Chi Tiết Phiếu Trả |
| [**http:/**](http://somee.com)**/localhost:8080/updateReturnBooksDetail/{id\_PhieuTra}&{barcode}** | PUT | Sửa thông tin Chi Tiết Phiếu Trả |
| [**http:/**](http://somee.com)**/localhost:8080/** **returnBooksDetailById/{id\_PhieuTra}** | GET | Hiển thị Chi Tiết Phiếu Trả theo ID Phiếu |
| [**http:/**](http://somee.com)**/localhost:8080/** **deleteReturnBooksDetail/{id\_PhieuTra}&{barcode}** | DELETE | Xóa Chi Tiết Phiếu Trả |

Bảng mô tả Url SubjectController

|  |  |  |
| --- | --- | --- |
| Url | Method hỗ trợ | Mô tả |
| [**http:/**](http://somee.com)**/localhost:8080/subject** | GET | Trả về danh sách tất cả các Thể Loại |
| [**http:/**](http://somee.com)**/localhost:8080/createSubject** | POST | Thêm Thể Loại |
| [**http:/**](http://somee.com)**/localhost:8080/updateSubject/{id\_TenLoaiSach}** | PUT | Sửa thông tin Thể Loại |
| [**http:/**](http://somee.com)**/localhost:8080/deleteSubject/{id\_TenLoaiSach}** | DELETE | Xóa Thể Loại |
| [**http:/**](http://somee.com)**/localhost:8080/subject/{id\_TenLoaiSach}** | GET | Hiển thị Thể Loại theo ID Thể Loại |

## Giao diện hệ thống reponsive sử dựng Bootstrap, AngularJS

### **Giao diện trang chủ cho Độc Giả**

Hình III.3.3.1 Giao diện Trang Chủ

### **Giao diện trang Kết Quả Tìm Kiếm Sách**

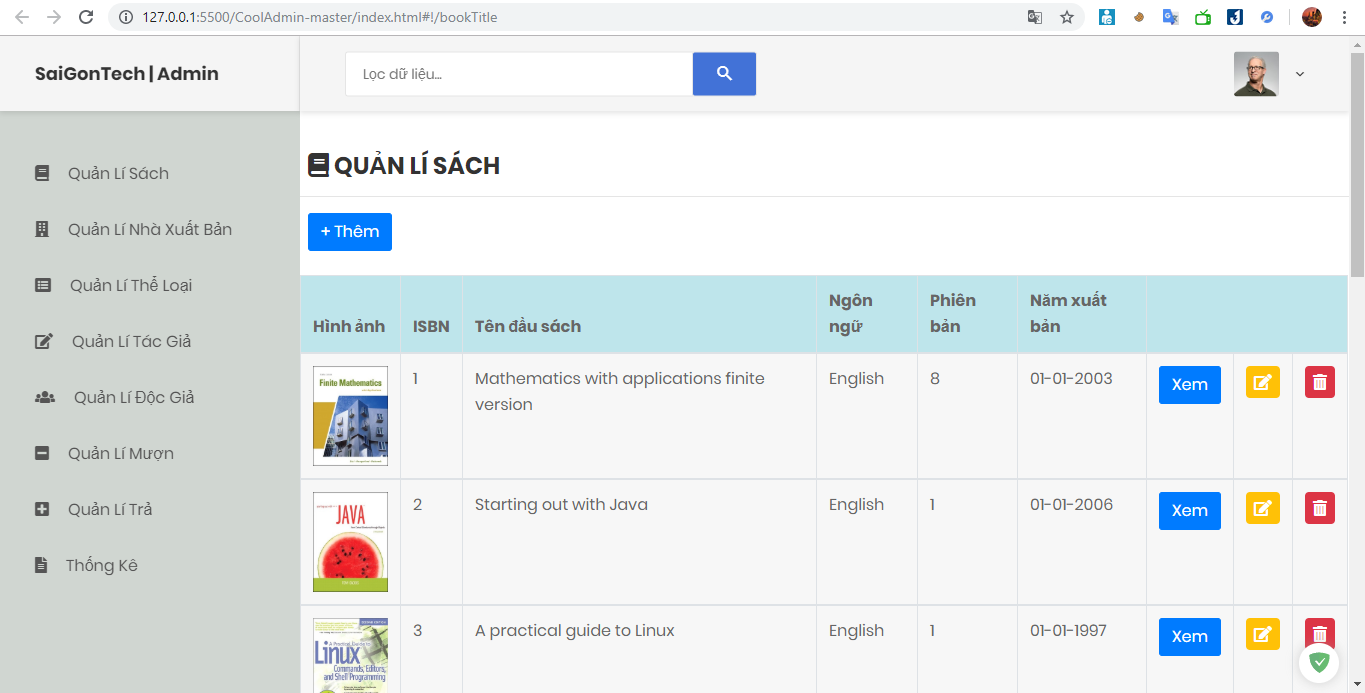
Hình III.3.3.2 Giao diện trang Kết Quả Tìm Kiếm Sách

### **Giao diện trang Chi Tiết Sách**

Hình III.3.3.3 Giao diện trang Chi Tiết Sách

### **Giao diện trang Quản lý Sách**

Hình III.3.3.4 Giao diện trang Quản lý Sách



### **Giao diện Thêm Sách**

Hình III.3.3.5 Giao diện Thêm Sách

### **Giao diện Sửa Sách**

Hình III.3.3.6 Giao diện Sửa Sách

### **Giao diện trang quản lý Nhà Xuất Bản**

Hình III.3.3.7 Giao diện trang Quản lý Nhà Xuất Bản

### **Giao diện trang quản lý Độc Giả**

Hình III.3.3.8 Giao diện trang Quản lý Độc Giả

### **Giao diện trang Quản lý Tác Giả**

Hình III.3.3.9 Giao diện trang Quản lý Tác Giả

### **Giao diện trang Quản lý Mượn**

Hình III.3.3.10 Giao diện trang Quản lý Mượn

### **Giao diện trang Quản lý Trả**

Hình III.3.3.11 Giao diện trang Quản lý Trả

### **Giao diện trang Thống Kê**

Hình III.3.3.12 Giao diện trang Thống Kê

# KẾT LUẬN

## Kết luận

Sau thời gian thực tập tại ISC Quang Trung nhờ sự giúp đỡ tận tình của thầy Võ Quốc Lương và anh Đinh Thanh Sơn cùng các anh chị trong công ty, đã tạo cho chúng em có được điều kiện tham quan môi trường làm việc thực tế, ứng dụng được những kiến thức đã học ở trường cũng như học hỏi thêm kiến thức mới, công nghệ mới, nghiên cứu, tìm hiểu và hoàn thành báo cáo với đề tài: “Xây dựng ứng dụng Website quản lý thư viện SaiGonTech”.

Tuy thời gian thực tập không dài nhưng đã cho chúng em thấy được tầm quan trọng của một người kỹ sư công nghệ thông tin trong môi trường làm việc thực tế. Qua đó chúng em học hỏi được nhiều điều quý giá từ đơn vị thực tập cũng như từ những anh chị trong công ty đã hỗ trợ giúp đỡ em về nhiều mặt như:

* **Về kiến thức**: Nắm được quy trình để tạo nên một phần mềm có tính thực tế và ứng dụng hiệu quả. Nắm được những phương pháp để đem những công nghệ mới vào sản phẩm, đồng thời đưa ra hướng đi đúng đắn để tiếp cận đề tài.
* **Về kỹ năng**: Vận dụng được kỹ năng giao tiếp không còn nhút nhát, e dè khi giao tiếp, tiếp xúc với mọi người, phát huy được khả năng phát biểu ý kiến; nắm được kỹ năng thuyết trình sao cho đầy đủ và người nghe không cảm thấy nhàm chán cũng như có thể hiểu được những nội dung trong bài thuyết trình.
* **Về thái độ**: Tuân thủ theo đúng quy định khi đến thực tập tại công ty, tôn trọng, lễ phép cũng như hòa đồng với tất cả anh chị trong công ty.

Vì thời gian hạn hẹp và chưa có nhiều kinh nghiệm thực tế. Vì vậy, đề tài còn nhiều thiếu sót, mong các Thầy Cô giúp đỡ và thông cảm.

Một lần nữa em xin chân thành cảm ơn các Ban lãnh đạo, Ban giám đốc công ty ISC Quang Trung, các Thầy Cô tại trường Đại Học Thủ Dầu Một cùng các anh chị trong công ty rất nhiều đã giúp đỡ và tạo cơ hội và điều kiện để thực tập tại đây trong thời gian qua.

## Kiến nghị

Sau khi kết thúc tất cả các môn học tại trường thì tất cả các sinh viên sẽ có khoảng thời gian là 2 tháng để thực tập, tiếp cận với môi trường làm việc thực tế. Tuy nhiên, với khoảng thời gian ngắn như thế, sinh viên vẫn chưa tích lũy được nhiều kinh nghiệm và kỹ năng làm việc để có thể chuẩn bị cho quá trình ra trường và làm việc sau này, vì vậy chúng em mong bên phía nhà trường có thể xem xét lại thời gian thực tập cho sinh viên cũng như tạo thêm nhiều điều kiện và cơ hội để sinh viên có thể có thêm nhiều thời gian học hỏi, tích lũy được nhiều kiến thức, kinh nghiệm làm việc.

Về những kiến thức được học ở trường theo chúng em thì chưa thực sự đủ để va chạm vào công việc thực tế mà trước hết là chưa đủ tự tin để trở thành một thực tập sinh. Vì vậy, mong nhà trường có thể tạo thêm nhiều khóa học về các cách quản lý, phân chia phát triển ứng dụng cho sinh viên khỏi bỡ ngỡ khi đi thực tập. Song song đó, chúng em cũng mong nhà trường có thể lược bớt lượng thời gian học lý thuyết, đồng thời tăng lượng thời gian thực hành. Hoặc có thể vừa học lý thuyết song song thực hành, như vậy sẽ có hiệu quả và không gây nhàm chán khi học.

Qua quá trình thực tập, chúng em đã có cơ hội trải nghiệm được công việc thực tế, hiểu được rõ hơn công việc của kỹ sư lập trình cũng như hình dung được những công việc mình có thể làm sau khi ra trường.

# TÀI LIỆU THAM KHẢO

1. https://en.wikipedia.org/wiki/HTML5
2. https://www.w3schools.com/
3. http://vietjack.com/
4. http://tutorialspoint.com/
5. https://scotch.io/tutorials/
6. <https://angularjs.org/>
7. https://bkaii.com.vn/tin-tuc/334-khai-niem-co-ban-ve-restful-api
8. https://backlog.com/git-tutorial/vn/intro/intro1\_1.html
9. https://itviec.com/blog/agile-la-gi-scrum-la-gi/
10. https://hocvienagile.com/agipedia/tong-quan-ve-scrum/
11. https://www.slideshare.net/thienan93/spring-framework-49322087
12. http://sqladvice.com/mysql-la-gi-tong-quan-ve-he-quan-tri-du-lieu-mysql/
13. https://voer.edu.vn/m/gioi-thieu-cong-nghe-hibernate/a10ab2a0
14. https://stackjava.com/uncategorized/orm-la-gi-tong-quan-ve-orm-framework.html
15. https://viettuts.vn/hibernate/hibernate-query-language