**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-----------------------------



**BÁO CÁO PHÂN TÍCH THIẾT KẾ VÀ ĐẢM BẢO CHẤT LƯỢNG PHẦN MỀM**

***Đề tài*: “ *Xây dựng website đọc truyện tiểu thuyết* ”**

|  |  |
| --- | --- |
| **Người hướng dẫn :** | **ThS. NGUYỄN ANH HÀO** |
| **Sinh viên thực hiện :** | |  |  | | --- | --- | | **BÙI PHI LONG** | **N18DCCN106** | | **NGUYỄN PHẠM NHẬT MINH** | **N17DCCN095** | |
| **Hệ :** | **ĐẠI HỌC CHÍNH QUY** |

**TP.HCM, tháng 11/2023**

# MỞ ĐẦU

1. **Giới thiệu**

Chuyển đổi số đang là xu hướng của thời đại hiện nay. Có thể nói, với sự phát triển như vũ bão của khoa học và công nghệ trong thời gian qua đã tạo ra những sản phầm công nghệ mới để rồi đem lại rất nhiều tiện ích cho cuộc sống. Nó đang chiếm phần lớn trong việc phục vụ của nhiều ngành nghề cũng như phục vụ đời sống của con người. Đặc biệt là trong công tác giải trí. Các cuốn sách được số hóa tạo điều kiện cho con người thuận tiện hơn trong việc truy cập. Các hình thức giải trí thư giãn đã được gói gọn trong các phần mềm, trang web.

Trong bối cảnh đó, việc xây dựng chương trình chuyển đổi các cuốn tiểu thuyết vật lý thành các thông tin điện tử là thiết thực, nhằm nâng cao hiêu quả cũng như làm cho việc tiếp cận văn hóa đại chúng trở nên dễ dàng hơn bao giờ hết. Do đó, em quyết định chọn đề tài “***Xây dựng website đọc truyện tiểu thuyết***” nhằm phục vụ mục đích giải trí thư giãn.

1. **Cơ sở lý thuyết**

Tìm hiểu về Java Spring để xây dựng website back-end cùng với thư viện hỗ trợ Spirng Framework. Tìm hiểu xây dựng cơ sở dữ liệu trên hệ quản trị cơ sở dữ liệu MySQL.

1. **Thực hành**

Xây dựng website với các chức năng:

* Trang chủ, tìm kiếm tiểu thuyết.
* Tạo tài khoản, đăng nhập.
* Các chức năng quản trị cho admin (Thêm, Sửa, Xóa).
* Chức năng đọc tiểu thuyết.
* Chức năng đánh giá.
* Chức năng bình luận.

Công cụ hỗ trợ:

* IntelliJ IDEA.
* Postgre SQL

# LỜI CẢM ƠN

Em xin chân thành cảm ơn giảng viên hướng dẫn là Thầy Nguyễn Anh Hào đã truyền dạy những kiến thức cần thiết trong việc thực hiện “***Xây dựng website đọc truyện tiểu thuyết***”. Để thực hiện đề tài này, em đã rất cố găng trong việc tìm hiểu các quy tắc nghiệp vụ cũng như quản trị để xây dựng nên ứng dụng.

Em rất mong nhận được sự quan tâm, thông cảm và những đóng góp quý báu của các thầy cô để báo cáo này ngày càng hoàn thiện hơn. Sau cùng, em xin kính chúc các thầy cô trong Khoa Công Nghệ Thông Tin sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Một lần nữa em xin chân thành cảm ơn!

TP. HCM, Ngày …. tháng …. năm 2023

Sinh viên thực hiện

# MỤC LỤC

[MỞ ĐẦU 3](#_Toc150849857)

[LỜI CẢM ƠN 4](#_Toc150849858)

[MỤC LỤC 5](#_Toc150849859)

[KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT 8](#_Toc150849860)

[Danh mục các bảng biểu 9](#_Toc150849861)

[Danh mục các hình vẽ 10](#_Toc150849862)

[Chương 1: GIỚI THIỆU ĐỀ TÀI 12](#_Toc150849863)

[1.1. Mục đích đề tài “Xây dựng website đọc truyện tiểu thuyết” 12](#_Toc150849864)

[1.2. Mục tiêu 12](#_Toc150849865)

[Chương 2: CƠ SỞ KHOA HỌC 12](#_Toc150849866)

[2.1. Hiện trạng thực tế 12](#_Toc150849867)

[2.2. Các giải pháp nâng cao hiệu quả 12](#_Toc150849868)

[2.3. Mô hinh nghiệp vụ của hệ thống 13](#_Toc150849869)

[2.3.1. Sơ đồ cơ cấu tổ chức 13](#_Toc150849870)

[2.3.2. Chức năng từng vai trò 13](#_Toc150849871)

[2.3.3. Quy trình nghiệp vụ 13](#_Toc150849872)

[2.4. Yêu cầu chức năng 14](#_Toc150849873)

[2.5. Yêu cầu phi chức năng 14](#_Toc150849874)

[2.6. Nền tảng công nghệ 15](#_Toc150849875)

[2.6.1. Spring Boot 15](#_Toc150849876)

[2.6.2. Công cụ lập trình 17](#_Toc150849877)

[2.6.3. Tổng quan về RESTful API 19](#_Toc150849878)

[Chương 3: PHÂN TÍCH HỆ THỐNG 21](#_Toc150849879)

[3.1. Sơ đồ phân cấp chức năng 21](#_Toc150849880)

[3.1.1. Sơ đồ phân cấp chức năng mức đỉnh 21](#_Toc150849881)

[3.1.2. Sơ đồ phân rã chức năng 22](#_Toc150849882)

[3.2. Các sơ đồ use-case của website đọc truyện tiểu thuyết 23](#_Toc150849883)

[3.2.1. Xác định các Actor 23](#_Toc150849884)

[3.2.2. Xác định các Use-case 23](#_Toc150849885)

[3.2.3. Xây dựng sơ đồ use-case tổng quát 24](#_Toc150849886)

[3.2.4. Đặc tả use-case 26](#_Toc150849887)

[3.3. Yêu cầu ràng buộc đối với phần mềm 38](#_Toc150849888)

[3.3.1. Yêu cầu từ môi trường nghiệp vụ 38](#_Toc150849889)

[3.3.2. Yêu cầu từ môi trường vận hành 39](#_Toc150849890)

[3.3.3. Yêu cầu từ môi trường phát triển 39](#_Toc150849891)

[Chương 4: THIẾT KẾ HỆ THỐNG 40](#_Toc150849892)

[4.1. Các use-case thiết kế 40](#_Toc150849893)

[4.1.1. Người đọc 40](#_Toc150849894)

[4.1.2. Người kiểm duyệt 48](#_Toc150849895)

[4.1.3. Người quản trị 49](#_Toc150849896)

[4.2. Thiết kế phần mềm để xử lý use-case 50](#_Toc150849897)

[4.2.1. UD01 50](#_Toc150849898)

[4.2.2. UD02 52](#_Toc150849899)

[4.2.3. UD03 53](#_Toc150849900)

[4.2.4. UD04 55](#_Toc150849901)

[4.2.5. UD05 58](#_Toc150849902)

[4.2.6. UD06 59](#_Toc150849903)

[4.2.7. UD07 60](#_Toc150849904)

[4.2.8. UD08 62](#_Toc150849905)

[4.2.9. UD09 63](#_Toc150849906)

[Chương 5: XÂY DỰNG CƠ SỞ DỮ LIỆU 65](#_Toc150849907)

[5.1. Thiết kế cơ sở dữ liệu quan hệ cho website đọc truyện tiểu thuyết 65](#_Toc150849908)

[5.2. Sơ đồ Thực thể quan hệ (ERD) cho website đọc truyện tiểu thuyết 67](#_Toc150849909)

[5.3. Mô hình dữ liệu (ER) 68](#_Toc150849910)

[5.3.1. Xét Role – User 68](#_Toc150849911)

[5.3.2. Xét User – Rating 68](#_Toc150849912)

[5.3.3. Xét User – Comment 68](#_Toc150849913)

[5.3.4. Xét User – Novel 68](#_Toc150849914)

[5.3.5. Xét Novel – Genre 68](#_Toc150849915)

[5.3.6. Xét Novel – Author 69](#_Toc150849916)

[5.3.7. Xét Novel – Chapter 69](#_Toc150849917)

[5.3.8. Xét Novel – Rating 69](#_Toc150849918)

[5.3.9. Xét Novel – Comment 69](#_Toc150849919)

[5.4. Sơ đồ của cơ sở dữ liệu cho website đọc truyện tiểu thuyết được hiển thị trong MS SQL Server 70](#_Toc150849920)

[5.5. Chi tiết các thực thể trong Cơ sở dữ liệu 70](#_Toc150849921)

[5.5.1. Thực thể tiểu thuyết (NOVEL) 70](#_Toc150849922)

[5.5.2. Thực thể tác giả (AUTHOR) 71](#_Toc150849923)

[5.5.3. Thực thể thể loại (GENRE) 71](#_Toc150849924)

[5.5.4. Thực thể chương tiểu thuyết (CHAPTER) 71](#_Toc150849925)

[5.5.5. Thực thể quyền hạn (ROLE) 72](#_Toc150849926)

[5.5.6. Thực thể người dùng (USER) 72](#_Toc150849927)

[5.5.7. Thực thể đánh giá của người dùng (USER\_RATING) 72](#_Toc150849928)

[5.5.8. Thực thể bình luận của người dùng (USER\_COMMENT) 72](#_Toc150849929)

[5.5.9. Thực thể NOVEL\_GENRE 73](#_Toc150849930)

[5.5.10. Thực thể NOVEL\_AUTHOR 73](#_Toc150849931)

[Chương 6: Website đọc truyển tiểu thuyết 73](#_Toc150849932)

[6.1. Yêu cầu hệ thống: 73](#_Toc150849933)

[6.2. Giao diện chương trình 73](#_Toc150849934)

[6.2.1. Trang chủ: Hiển thị danh sách các tiểu thuyết 74](#_Toc150849935)

[6.2.2. Trang đăng nhập: 75](#_Toc150849936)

[6.2.3. Trang đăng tiểu thuyết của người đọc: 75](#_Toc150849937)

[6.2.4. Trang chi tiêt tiểu thuyết 76](#_Toc150849938)

[6.2.5. Trang nội dung chương 76](#_Toc150849939)

[6.2.6. Hộp thoại thông báo xác nhận: 77](#_Toc150849940)

[Chương 7: KẾT LUẬN 78](#_Toc150849941)

[7.1. Đánh giá kết quả đạt được 78](#_Toc150849942)

[7.2. Những kết quả đạt được 78](#_Toc150849943)

[7.3. Những hạn chế 78](#_Toc150849944)

[7.4. Hướng phát triển 79](#_Toc150849945)

[7.5. Kết luận 79](#_Toc150849946)

[DANH MỤC TÀI LIỆU THAM KHẢO 80](#_Toc150849947)

# KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

|  |  |
| --- | --- |
| API: Application Programming Interface | Giao diện lập trình ứng dụng |
| CSS: Cascading Style Sheets | Cơ chế để định dạng hiển thị HTML |
| ERD: Entity Rlationship Diagram | Mô hình thực thể quan hệ |
| HTML: HyperText Markup Language | Ngôn ngữ đánh dấu siêu văn bản |
| MS: Microsoft | Tập đoàn phát triển sản phẩm liên quan đến máy tính |
| MVC: Model-View-Controller | Mẫu kiến trúc phần mềm để phát triển ứng dụng |
| SQL: Structured Query Language | Ngôn ngữ truy vấn có cấu trúc |

# Danh mục các bảng biểu

[Bảng 2.1 Các giải pháp nâng cao hiệu quả 13](#_Toc150849837)

[Bảng 2.2 Yêu cầu chức năng 14](#_Toc150849838)

[Bảng 3.1 Xác định các Use-case 24](#_Toc150849839)

[Bảng 3.2 Đặc tả use-case đăng tiểu thuyết 27](#_Toc150849840)

[Bảng 3.3 Đặc tả use-case duyệt tiểu thuyết 29](#_Toc150849841)

[Bảng 3.4 Đặc tả use-case xem tiểu thuyết 30](#_Toc150849842)

[Bảng 3.5 Đặc tả use-case tìm kiếm 32](#_Toc150849843)

[Bảng 3.6 Đặc tả use-case đánh giá tiểu thuyết 34](#_Toc150849844)

[Bảng 3.7 Đặc tả use-case bình luận 36](#_Toc150849845)

[Bảng 3.8 Đặc tả use-case quản lý tài khoản người dùng 38](#_Toc150849846)

[Bảng 5.1 Thực thể tiểu thuyết (NOVEL) 71](#_Toc150849847)

[Bảng 5.2 Thực thể tác giả (AUTHOR) 71](#_Toc150849848)

[Bảng 5.3 Thực thể thể loại (GENRE) 71](#_Toc150849849)

[Bảng 5.4 Thực thể chương tiểu thuyết (CHAPTER) 71](#_Toc150849850)

[Bảng 5.5 Thực thể quyền hạn (ROLE) 72](#_Toc150849851)

[Bảng 5.6 Thực thể người dùng (USER) 72](#_Toc150849852)

[Bảng 5.7 Thực thể đánh giá của người dùng (USER\_RATING) 72](#_Toc150849853)

[Bảng 5.8 Thực thể bình luận của người dùng (USER\_COMMENT) 73](#_Toc150849854)

[Bảng 5.9 Thực thể NOVEL\_GENRE 73](#_Toc150849855)

[Bảng 5.10 Thực thể NOVEL\_AUTHOR 73](#_Toc150849856)

# Danh mục các hình vẽ

[Hình 2.1 Sơ đồ cơ cấu tổ chức 13](#_Toc150849948)

[Hình 2.2 Lược đồ cộng tác đọc tiểu thuyết 13](#_Toc150849949)

[Hình 2.3 Lược đồ cộng tác đăng tiểu thuyết 14](#_Toc150849950)

[Hình 2.4 Lược đồ cộng tác đánh giá tiểu thuyết 14](#_Toc150849951)

[Hình 2.5 Lược đồ cộng tác bình luận 14](#_Toc150849952)

[Hình 3.1 Sơ đồ phân cấp chức năng mức đỉnh 21](#_Toc150849953)

[Hình 3.2 Chức năng quản lý người dùng 22](#_Toc150849954)

[Hình 3.3 Chức năng quản lý tiểu thuyết 22](#_Toc150849955)

[Hình 3.4 chức năng đánh giá 23](#_Toc150849956)

[Hình 3.5 Chức năng bình luận 23](#_Toc150849957)

[Hình 3.6 Sơ đồ use-case tổng quát của Người quản trị 24](#_Toc150849958)

[Hình 3.7 Sơ đồ use-case tổng quát của Người đọc 25](#_Toc150849959)

[Hình 3.8 Sơ đồ use-case tổng quát của Người kiểm duyệt 25](#_Toc150849960)

[Hình 3.9 Use-case đăng tiểu thuyết 26](#_Toc150849961)

[Hình 3.10 Biểu đồ tuần tự chức năng đăng tiểu thuyết 28](#_Toc150849962)

[Hình 3.11 Use-case duyệt tiểu thuyết 29](#_Toc150849963)

[Hình 3.12 Use-case xem tiểu thuyết 30](#_Toc150849964)

[Hình 3.13 Biểu đồ tuần tự chức năng xem tiểu thuyết 31](#_Toc150849965)

[Hình 3.14 Use-case tìm kiếm 31](#_Toc150849966)

[Hình 3.15 Biểu đồ tuần tự chức năng tìm kiếm 32](#_Toc150849967)

[Hình 3.16 Biểu đồ tuần tự chức năng lọc theo thể loại 33](#_Toc150849968)

[Hình 3.17 Use-case đánh giá tiểu thuyết 33](#_Toc150849969)

[Hình 3.18 Biểu đồ tuần tự chức năng đánh giá tiểu thuyết 35](#_Toc150849970)

[Hình 3.19 Use-case bình luận 35](#_Toc150849971)

[Hình 3.20 Biểu đồ tuần tự chức năng bình luận 37](#_Toc150849972)

[Hình 3.21 Use-case quản lý tài khoản người dùng 37](#_Toc150849973)

[Hình 3.22 Biểu đồ tuần tự chức năng quản lý tài khoản người dùng 38](#_Toc150849974)

[Hình 4.1 Use-case tổng quát của người đọc 40](#_Toc150849975)

[Hình 4.2 Use-case đăng nhập 40](#_Toc150849976)

[Hình 4.3 Sơ đồ tuần tự chức năng đăng nhập 41](#_Toc150849977)

[Hình 4.4 Use-case đăng ký 41](#_Toc150849978)

[Hình 4.5 Sơ đồ tuần tự chức năng đăng ký 42](#_Toc150849979)

[Hình 4.6 Use-case xem tiểu thuyết 42](#_Toc150849980)

[Hình 4.7 Sơ đồ tuần tự chức năng xem tiểu thuyết 43](#_Toc150849981)

[Hình 4.8 Use-case đăng tiểu thuyết 43](#_Toc150849982)

[Hình 4.9 Sơ đồ tuần tự chức năng đăng tiểu thuyết 44](#_Toc150849983)

[Hình 4.10 Use-case đánh giá 45](#_Toc150849984)

[Hình 4.11 Sơ đồ tuần tự chức năng đánh giá 45](#_Toc150849985)

[Hình 4.12 Use-case bình luận 46](#_Toc150849986)

[Hình 4.13 Sơ dồ tuần tự chức năng bình luận 46](#_Toc150849987)

[Hình 4.14 Use-case tìm kiếm 47](#_Toc150849988)

[Hình 4.15 Sơ đồ tuần tự chức năng tìm kiếm 47](#_Toc150849989)

[Hình 4.16 Use-case duyệt tiểu thuyết 48](#_Toc150849990)

[Hình 4.17 Sơ đồ tuàn tụ chức năng duyệt tiểu thuyết 48](#_Toc150849991)

[Hình 4.18 Use-case quản lý tài khoản 49](#_Toc150849992)

[Hình 4.19 Sơ đồ tuần tự chức năng quản lý tài khoản 49](#_Toc150849993)

[Hình 4.20 Form Login 50](#_Toc150849994)

[Hình 4.21 Form Home 51](#_Toc150849995)

[Hình 4.22 Form Signup 52](#_Toc150849996)

[Hình 4.23 Form Novel Detail 53](#_Toc150849997)

[Hình 4.24 Form Chapter Detail 54](#_Toc150849998)

[Hình 4.25 Form List Novels 56](#_Toc150849999)

[Hình 4.26 Form Chapter Detail 59](#_Toc150850000)

[Hình 4.27 Form danh sách tiểu thuyết 60](#_Toc150850001)

[Hình 4.28 Form duyệt tiểu thuyết 62](#_Toc150850002)

[Hình 4.29 Form chỉnh sửa tiểu thuyết 62](#_Toc150850003)

[Hình 4.30 Form danh sách tài khoản 64](#_Toc150850004)

[Hình 4.31 Form chỉnh sửa tài khoản 64](#_Toc150850005)

[Hình 5.1 Sơ đồ thực thể quan hệ (ERD) 67](#_Toc150850006)

[Hình 5.2 Sơ đồ của cơ sở dữ liệu cho website đọc truyện tiểu thuyết được hiển thị trong MS SQL Server 70](#_Toc150850007)

[Hình 6.1 Trang chủ 74](#_Toc150850008)

[Hình 6.2 Trang đăng nhập 75](#_Toc150850009)

[Hình 6.3 Trang đăng tiểu thuyết 75](#_Toc150850010)

[Hình 6.4 Trang chi tiết tiểu thuyết 76](#_Toc150850011)

[Hình 6.5 Trang nội dung chương 76](#_Toc150850012)

[Hình 6.6 Hộp thoại thông báo xác nhận 77](#_Toc150850013)

# GIỚI THIỆU ĐỀ TÀI

## Mục đích đề tài “Xây dựng website đọc truyện tiểu thuyết”

* Phần mềm dùng để cung cấp các tiểu thuyết cho người đọc.
* Tạo một cộng đồng có cùng sở thích đọc tiểu thuyết.

## Mục tiêu

* Thay thế các ấn phẩm vật lý: Hiện nay, đa số mọi người đểu có khả năng truy cập các tiện ích trên internet. Nhu cầu giải trí cũng theo đó tăng cao. Các ấn phẩm vật lý đểu trở nên số hóa.
* Tiện ích: Phần mềm được xây dựng trên nền tảng web application, người sử dụng phần mềm có thể truy cập mọi lúc, mọi nơi qua trình duyệt web.
* Cộng đồng: tạo nên một nơi giao lưu thảo luận về một tựa tiểu thuyết mà bản thân có hứng thú với người đọc khác.

# CƠ SỞ KHOA HỌC

## Hiện trạng thực tế

Do nhu cầu tiện ích ngày càng phát triển, việc sở hữu các ấn phẩm sách, truyện vật lý dần trở nên không được ưa chuộng. Đôi khi có những tác phẩm mà độc giả muốn đọc nhưng các Nhà xuất bản ở Việt Nam không thể đáp ứng do các điều lệ, giới hạn về bản quyền, dịch thuật. Ngoài ra, việc tìm kiếm những đầu sách của độc giả rất mất thời gian. Thêm vào đó, các ấn phẩm vật lý thường rất hay bị bào mòn, hư hỏng theo thời gian và cũng chiếm một không gian nhất định.

Những vấn đề trên đây làm hạn chế khả năng đáp ứng nhu cầu đọc sách của độc giả.

## Các giải pháp nâng cao hiệu quả

Những vấn đề đặt ra trên đây cho thấy, cần có một giải pháp để đáp ứng nhu cầu người đọc một cách tốt nhất.

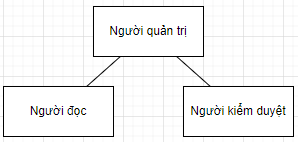
Một trong các giải pháp để nâng cao hiệu quả là số hóa các ấn phẩm vật lý và vận hành trong điều kiện trang thiết bị cho phép. Từ những vấn đề phát hiện, ta có thể có các giải pháp theo cách số hóa các đầu sách/tiểu thuyết:

|  |  |
| --- | --- |
| **Vấn đề** | **Giải pháp** |
| Người đọc muốn đọc truyện mọi lúc mọi nơi, không thể sỡ hữu ấn bản vật lý | Xây dựng phần mềm trên nền tảng website |
| Người đọc muốn thảo luận về một tựa sách với người khác | Chức năng bình luận để người dùng có thể thảo luận với người dùng khác |
| Người đọc muốn biết tựa sách có chất lượng về nội dung | Chức năng đánh giá để giúp người đọc có cái nhìn tổng quan về chất lượng của tụa sách |
| Người dùng muốn chia sẻ một tụa sách mà học yêu thích | Phần mềm cho phép người dùng có thể đăng tiểu thuyết. |

Bảng . Các giải pháp nâng cao hiệu quả

## Mô hinh nghiệp vụ của hệ thống

### Sơ đồ cơ cấu tổ chức



Hình . Sơ đồ cơ cấu tổ chức

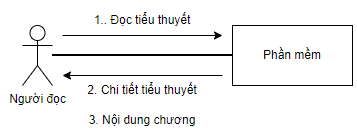
### Chức năng từng vai trò

* Người quản trị: Có vai trò cao nhất trong hệ thống, quản lý và phân quyền cho người dùng.
* Người đọc: Có vai trò đóng góp thêm truyện cho hệ thống (tùy chọn).
* Người kiểm duyệt: Kiểm duyệt các truyện được đăng bởi người đọc.

### Quy trình nghiệp vụ

#### Đọc tiểu thuyết

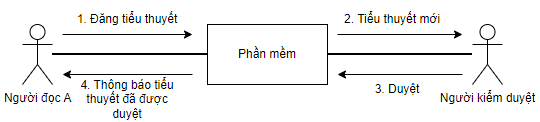
* Người đọc (không yêu cầu xác thực): Đọc nội dung của tiểu thuyết như chi tiết về tiểu thuyết, nội dung chương



Hình . Lược đồ cộng tác đọc tiểu thuyết

#### Đăng truyện

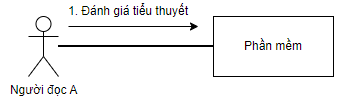
* Người đọc:
  + Đăng một truyện với các chi tiết về truyện như thể loại, tác giả, ảnh bìa, mô tả.
  + Đăng nội dung của chương truyện.
* Người kiểm duyệt:
  + Xem chi tiết truyện và quyết định truyện có được hiển thị đến với người dùng khác hay không.



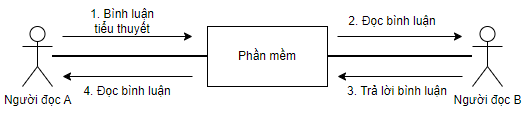
Hình . Lược đồ cộng tác đăng tiểu thuyết

#### Đánh giá, bình luận

* Người đọc có thể thêm đánh giá và bình luận trên truyện mà họ yêu thích để chia sẻ ý kiến với người dùng khác.



Hình . Lược đồ cộng tác đánh giá tiểu thuyết



Hình . Lược đồ cộng tác bình luận

#### Quản lý thể loại, tác giả, người dùng

* Người quản trị giữ vai trò cập nhật, thêm mới thể loại và quản trị tác giả.
* Cung cấp phân quyền cho người dùng có trách nhiệm: người kiểm duyệt

## Yêu cầu chức năng

|  |  |
| --- | --- |
| **Chức năng** | **Ý nghĩa** |
| Chức năng đăng truyện | Quản lý thông tin liên quan đến truyện như chi tiết truyện, chương truyện. |
| Chức năng kiểm duyệt | Quản lý các truyện đã đăng bởi Người đọc |
| Chức năng đánh giá | Đánh giá truyện theo mức từ 1 – 5 |
| Chức năng bình luận | Thảo luận với Người đọc khác về nội dung truyện |
| Quản lý tác giả, thể loại | Người quản trị quản lý các thông tin liên quan đến tác giả, thể loại |

Bảng . Yêu cầu chức năng

## Yêu cầu phi chức năng

* Hệ thống chạy đúng, tốt, ổn định
* Giao diện thân thiện, dễ sử dụng
* Đáp ứng hầu hết các yêu cầu của người dùng.

## Nền tảng công nghệ

### Spring Boot



Spring Boot là một Spring framework hiện đang được nhiều lập trình viên lựa chọn vì nó giúp cho các lập trình viên chúng ta đơn giản hóa quá trình lập trình một ứng dụng với Spring, chỉ tập trung vào việc phát triển business cho ứng dụng. Spring Boot được coi như là điểm khởi đầu để xây dựng tất cả các ứng dụng dựa trên nền tảng Spring.

#### Tìm hiểu cách xây dựng một thực thể

Trong JPA – Hibernate, một entity là một Java class (đối tượng) tương ứng với một table trong database, mỗi entity object tương ứng với một dòng trong bảng. JPA yêu cầu một entity class phải cung cấp đầy đủ các metadata để ánh xạ giữa entity class và table như tên cột, kiểu dữ liệu, khoá chính, khoá ngoại, etc.

Việc đối tượng hoá entity diễn được thực hiện thông qua các tệp cấu hình XML hoặc thông qua annotations, sau đây là cách sử dụng annotation:

Class AuthorEntity:

@Entity

@Table(name = "author")

public class AuthorEntity extends BaseEntity {

@Column

private String name;

@Id

@Column

private String code;

//*Các getter, setter*

}

Tương ứng với các cột trong table trong database, trên đối tượng Author cũng phải có các thuộc tính tương ứng: Mã tác giả, Tên tác giả. Các annotation được sử dụng:

* @Entity để xác định lớp này sẽ ánh xạ với thực thể trong CSDL.
* @Table(name = "author") để xác định chính xác table trong CSDL, ở đây là table author.
* @Id để xác định khóa chính.
* @Column để xác định với thuộc tính trong table. Cách viết khác là @Column(name= “name”).

Do một tiểu thuyết sẽ có thẻ có nhiều tác giả nên ta sẽ phải xác định quan hệ Many to many với tiểu thuyết.

Class NovelEntity:

@Entity

@Table(name = "author")

public class AuthorEntity extends BaseEntity {

@Id

@Column

private String code;

@Column

private String title;

@Column

private String description;

//*Các getter, setter*

@ManyToMany(fetch = FetchType.LAZY)

@JoinTable(name = " post\_author",

joinColumns = @JoinColumn(name = " novel\_id ", nullable = false),

inverseJoinColumns = @JoinColumn(name = "author\_id", nullable = false))

private List<AuthorEntity> authors = new ArrayList<>();

}

Class AuthorEntity:

@ManyToMany(fetch = FetchType.LAZY)

private List<PostEntity> posts = new ArrayList<>();

Trong đó:

* @ManyToMany để xác định quan hệ nhiều nhiều với bảng novel.
* @JoinTable để xác định mối quan với với thực thể khác, trong đó sử dụng các tham số joinColumns = @JoinColumn(name = “author\_id”) và inverseJoinColumns = @JoinColumn(name = "author\_id", nullable = false) để chỉ ra khóa nào sẽ tham gia quan hệ Many to many này. Ở đây là 2 khóa chính của “author\_id” và “novel\_id” 2 bảng Author và Novel.
* Sử dụng List<Entity> entity = new ArrayList<>(); để lấy ra danh sách những dữ liệu liên quan.
* Ở bảng author ta chỉ cần viết @ManyToMany và một list để lấy những dữ liệu tiểu thuyết liên quan đến tác giả và ngược lại.

#### Tìm hiểu về JPA repository

JPA Repository là một phần mở rộng của Spring Framework cho việc làm việc với dữ liệu trong ứng dụng Java, giúp đơn giản hóa việc thao tác với cơ sở dữ liệu bằng cách tự động tạo các phương thức CRUD (Create, Read, Update, Delete) cho các đối tượng Entity.

Các bước để có thể sử dụng JPA Repository:

Mỗi Entity, khai báo một interface tương ứng, interface này dùng để mô tả các phương thức mà người lập trình muốn sử dụng. Spring Data JPA sẽ tự động cài đặt các phương thức này dựa trên tên của phương thức và cấu trúc Entity. Thay vì viết mã SQL hoặc tạo các truy vấn tùy chỉnh, người lập trình có thể sử dụng JPA Repository để tương tác với cơ sở dữ liệu một cách dễ dàng và an toàn.

Ví dụ, để sử dụng các phương thức tương tác với table user của entity Author, khai báo interface như sau:

AuthorRepository:

import org.springframework.data.jpa.repository.JpaRepository;

public interface AuthorRepository extends JpaRepository<AuthorEntity, Long> {

}

* Kế thừa JpaRepository<[Entity của thực thể], [Kiểu dữ liệu của khoá chính của thực thể]>
* Khi kế thừa lớp trên, Spring Data JPA tự động cung cấp các phương thức như save(), findAll(), delete()...
* Ngoài ra, ta còn có thể tự định nghĩa các phương thức mới phù hợp với yêu cầu và bài toán:

import org.springframework.data.jpa.repository.JpaRepository;

public interface AuthorRepository extends JpaRepository<AuthorEntity, Long> {

AuthorEntity findOneByCode(String code);

long countByNameContainingIgnoreCase(String name);

void deleteAllByIdIn(long[] ids);

}

* Ví dụ:
  + findOneByCode(String code): Tìm một tác giả bằng cách truyền Mã tác giả. Phương thức này trả về một đối tượng AuthorEntity và các tiểu thuyết của tác giả này.
  + countByNameContainingIgnoreCase(String name): Trả về số lượng tác giả.
  + deleteAllByIdIn(long[] ids): Xóa một danh sách các tác giả.

### Công cụ lập trình

#### IntelliJ IDEA

IntelliJ IDEA cung cấp trình soạn thảo thông minh, trình phân tích mã và tập hợp mạnh mẽ của refactorings hỗ trợ một loạt các ngôn ngữ lập trình, các khuôn khổ và công nghệ, và đã sẵn sàng để sử dụng.



**Lợi ích chính của phần mềm IntelliJ IDEA:**

* Cho phép các nhà phát triển tập trung phát triển và quản lý tất cả các tác vụ thông thường
* Cho phép viết, gỡ lỗi, tái cấu trúc, kiểm tra và tìm hiểu mã của bạn mà không bị ảnh hưởng.
* Xử lý liền mạch cơ sở mã hỗn hợp của Java, Ruby, Groovy, Python và Scala.
* Tự động duy trì chất lượng mã.
* Theo dõi và sửa lỗi trên tất cả các cấp độ – từ các câu đến kiến trúc tổng thể.
* Tạo mã “sạch”, nhanh chóng thực hiện mã trong thời gian ngắn nhất.
* Được thiết kế để làm việc trên tất cả các quy mô – từ cá nhân đến doanh nghiệp.
* Hỗ trợ tất cả các ngôn ngữ, công nghệ và framework chính.
* Làm việc với các hệ thống điều khiển phiên bản phổ biến và TeamCity, server tích hợp liên tục.

#### Cách tạo một dự án trên IntelliJ IDEA:

Mở IntelliJ và chọn “Create New Project” (Tạo dự án mới).  
Chọn “Java” trong danh sách các ngôn ngữ và “Web Application” (Ứng dụng web) trong danh sách các mô hình. Đặt tên và chọn đường dẫn cho dự án và nhấp vào nút “Finish” (Hoàn thành).

**Cấu hình cài đặt dự án:**

1. Thêm Spring framework

* Nhấp chuột phải vào tên dự án trong IntelliJ, chọn “Open Module Settings” (Mở thiết lập mô-đun)
* Ở phần “Modules”, Chọn tab “Dependencies” (Phụ thuộc).
* Chọn “Add” > “Library” > “From Maven”
* Tìm kiếm “springframework”, chọn, tải và cài đặt phiên bản phù hợp (đề xuất bản mới nhất).

1. Cấu hình file application.properties:

* server.port=8080 hoặc port tùy ý.
* spring.datasource.url=jdbc:postgresql://[tên server]:[port]/[tên database]
* spring.datasource.username=[tên đăng nhập CSDL]
* spring.datasource.password=[mật khẩu]

#### Postman

Postman là một công cụ cho phép chúng ta thao tác với API, phổ biến nhất là REST. Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.

Postman hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, …). Bên cạnh đó, nó còn cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.



**Những lợi ích khi sử dụng Postman:**

* Sử dụng Collections (Bộ sưu tập) – Postman cho phép người dùng tạo bộ sưu tập cho các lệnh gọi API của họ. Mỗi bộ sưu tập có thể tạo các thư mục con và nhiều yêu cầu (request). Điều này giúp việc tổ chức các bộ thử nghiệm.
* Collaboration – Collections và environment có thể được import hoặc export giúp chia sẻ tệp dễ dàng.
* API Testing – Test trạng thái phản hồi HTTP.
* Gỡ lỗi – Bảng điều khiển Postman giúp kiểm tra dữ liệu nào đã được truy xuất giúp dễ dàng gỡ lỗi kiểm tra.

### Tổng quan về RESTful API

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

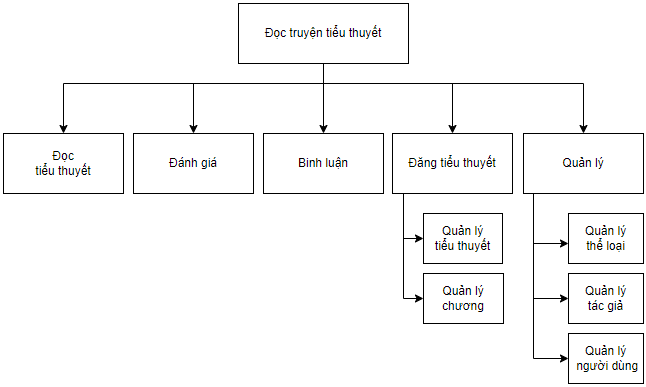
REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

# PHÂN TÍCH HỆ THỐNG

## Sơ đồ phân cấp chức năng



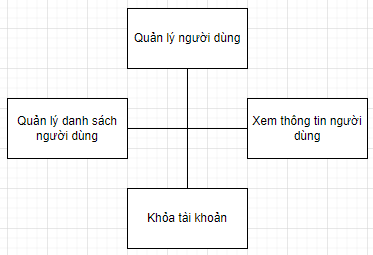
Hình . Sơ đồ phân cấp chức năng mức đỉnh

### Sơ đồ phân cấp chức năng mức đỉnh

Hệ thông nghiệp vụ được xét gồm các chức năng chính: Quản lý người dùng, quản lý thể loại, quản lý tác giả, đăng tiểu thuyết, bình luận, đánh giá, thống kê. Mỗi chức năng này được phân ra thành một số chức năng khác nhỏ hơn.

### Sơ đồ phân rã chức năng

#### Quản lý người dùng

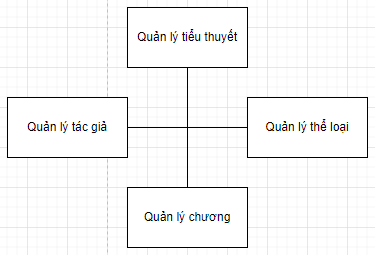
****

Hình . Chức năng quản lý người dùng

Mô tả chi tiết các chức năng:

* Quản lý danh sách người dùng: Người quản trị có thể tạo mới tài khoản người đọc và cập nhật các thông tin của nhân viên đó.
* Xem thông tin người dùng: Sau khi cập nhật thông tin ngườ dùng, người quản trị thể kiểm tra và hiệu chỉnh thông tin người dùng.
* Khóa tài khoản: Người quản lý có khả năng khóa thông tin người đọc để hạn chế đăng nhập.

#### Quản lý tiểu thuyết

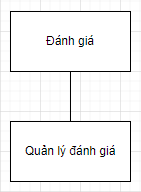


Hình . Chức năng quản lý tiểu thuyết

Mô tả chi tiết các chức năng:

* Quản lý tác giả: Người quản trị thực hiện các chức năng thêm mới, xóa và cập nhật tác giả.
* Quản lý thể loại: Người quản trị thực hiện các chức năng thêm mới, xóa và cập nhật thể loại.
* Quản lý chương: Sau khi đã hoàn thành thêm tiểu thuyết, người dùng thực hiện các chức năng thêm mới, xóa và cập nhật danh sách và nội dung chương.

#### Đánh giá

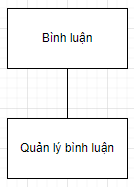


Hình . chức năng đánh giá

Mô tả chi tiết các chức năng:

* Quản lý đánh giá: Người dùng thực hiện các chức năng thêm mới, xóa và cập nhật đánh giá trên tiểu thuyết.

#### Bình luận



Hình . Chức năng bình luận

Mô tả chi tiết các chức năng:

* Quản lý bình luận: Người dùng thực hiện các chức năng thêm mới, xóa và cập nhật bình luận trên tiểu thuyết.

## Các sơ đồ use-case của website đọc truyện tiểu thuyết

### Xác định các Actor

Dựa vào yêu cầu của bài toán, ta có các actor: Người quản trị, người đọc, người kiểm duyệt.

### Xác định các Use-case

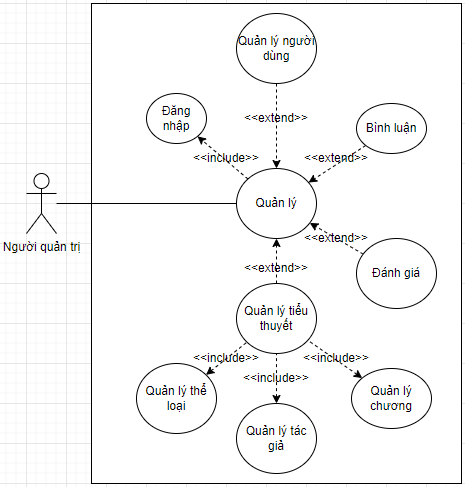
Từ những yêu cầu chức năng ứng với từng actor ta có thể xác định được các use case như sau:

|  |  |
| --- | --- |
| **Actor** | **Use-case** |
| Người quản trị | Đăng nhập, quản lý tài khoản người dùng, quản lý tiểu thuyết. |
| Người đọc | Xem tiểu thuyết, đăng nhập, đăng tiểu thuyết, đăng chương tiểu thuyết, đánh giá, bình luận. |
| Người kiểm duyệt | Đăng nhập, phê duyệt tiểu thuyết, quản lý tài khoản người đọc. |

Bảng . Xác định các Use-case

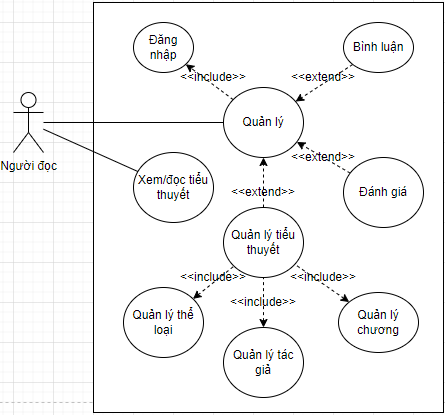
### Xây dựng sơ đồ use-case tổng quát

#### Sơ đồ use-case tổng quát của Người quản trị



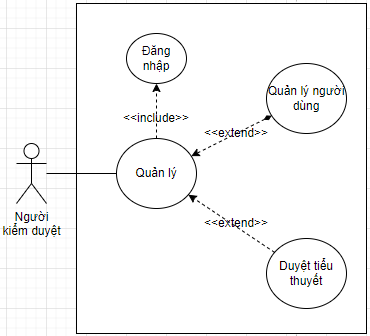
Hình . Sơ đồ use-case tổng quát của Người quản trị

#### Sơ đồ use-case tổng quát của Người đọc



Hình . Sơ đồ use-case tổng quát của Người đọc

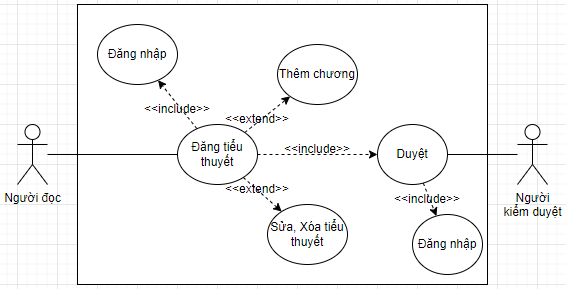
#### Sơ đồ use-case tổng quát của Người kiểm duyệt



Hình . Sơ đồ use-case tổng quát của Người kiểm duyệt

### Đặc tả use-case

#### Use-case đăng tiểu thuyết (UC01)



Hình . Use-case đăng tiểu thuyết

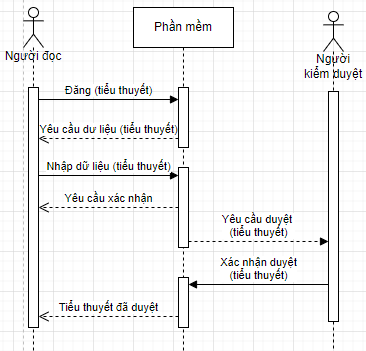
##### Đặc tả use-case đăng tiểu thuyết

* Quản lý tiểu thuyết có 3 chức năng chính là tạo mới, chỉnh sửa, xóa tiểu thuyết.
* Ngoài ra, người sử dụng còn có thể quản lý chương của tiểu thuyết.

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên chức năng | Đăng tiểu thuyết |
| Tác nhân | Người đọc, người kiểm duyệt |
| Điều kiện trước | Phải ở trạng thái xác thực |
| Điều kiện sau | Cần phải có sự kiểm soắt nội dung từ người kiểm duyệt |
| Mô tả | Tạo mới tiểu thuyết như tên tiểu thuyết, mô tả, ảnh bìa, các thể loại, các tác giả |
| Kịch bản | Người đọc có sẵn một hoặc nhiều các tiểu thuyết và muốn chia sẻ đến người đọc khác |
| Tương tác | |  |  | | --- | --- | | Hành động của tác nhân | Hồi đáp của hệ thống | | Người đọc (đã xác thực) yêu cầu đăng tiểu thuyêt | Yêu cầu dữ liệu (tiểu thuyết) | | Cung cấp dữ liệu (tiểu thuyết): | Yêu cầu kiểm tra hợp lệ thông tin, xử lý và trả kết quả | |  | Yêu cầu Người kiểm duyệt duyệt (tiểu thuyêt) | | Người kiểm duyệt duyệt (tiểu thuyết) |  | |  | Nếu thành công tiểu thuyết được công khai với người đọc khác | |

Bảng . Đặc tả use-case đăng tiểu thuyết

##### Biểu đồ tuần tự chức năng đăng tiểu thuyết



Hình . Biểu đồ tuần tự chức năng đăng tiểu thuyết

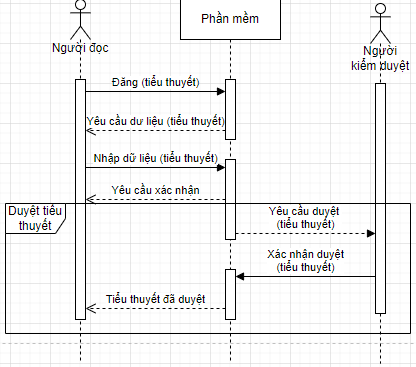
#### Use-case duyệt tiểu thuyết (UC01-01)

##### Đặc tả use-case duyệt tiểu thuyết

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên chức năng | Duyệt tiểu thuyết |
| Tác nhân | Người kiểm duyệt |
| Điều kiện trước | Phải ở trạng thái xác thực, đã có bài đăng chưa duyệt từ Người đọc |
| Điều kiện sau |  |
| Mô tả | Duyệt nội dung tiểu thuyết như tên tiểu thuyết, mô tả, ảnh bìa, các thể loại, các tác giả, nội dung chương |
| Kịch bản | Người duyệt duyệt tiểu thuyết được đăng bởi người đọc |
| Tương tác | |  |  | | --- | --- | | Hành động của tác nhân | Hồi đáp của hệ thống | | Người đọc (đã xác thực) đăng tiểu thuyêt | Chuyển tiểu thuyết đến cho người kiểm duyệt | | Người kiểm duyệt duyệt tiểu thuyết | Yêu cầu kiểm tra hợp lệ thông tin, xử lý và trả kết quả | |  | Nếu thành công tiểu thuyết được công khai với người đọc khác | |

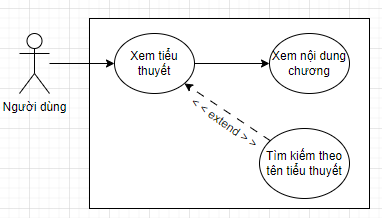
Bảng . Đặc tả use-case duyệt tiểu thuyết

##### Biểu đồ tuần tự chức năng duyệt tiểu thuyết



Hình . Use-case duyệt tiểu thuyết

#### Use-case xem tiểu thuyết (UC02)



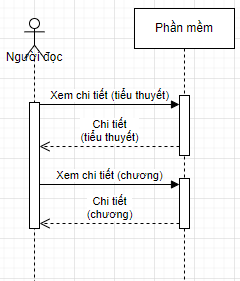
Hình . Use-case xem tiểu thuyết

##### Đặc tả use-case xem tiểu thuyết

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Tên chức năng | Xem tiểu thuyết |
| Tác nhân | Người đọc |
| Điều kiện trước |  |
| Điều kiện sau |  |
| Mô tả | Xem nội dung tiểu thuyết như tên tiểu thuyết, mô tả, ngày đăng, người đăng, lượt xem, tên chương, nội dung chương, bình luận, đánh giá. |
| Kịch bản | Người đọc muốn xem tiểu thuyết: nội dung chương, bình luận, đánh giá |
| Tương tác | |  |  | | --- | --- | | Hành động của tác nhân | Hồi đáp của hệ thống | | Yêu cầu tiểu thuyết | Hiện nội dung tiểu thuyết | | Yêu cầu nội dung chương | Hiện nôi dung chương | |

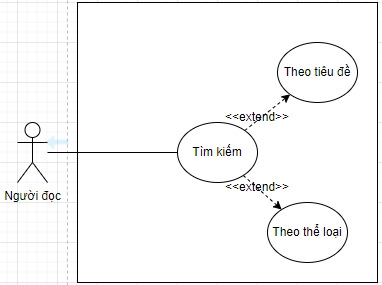
Bảng . Đặc tả use-case xem tiểu thuyết

##### Biểu đồ tuần tự chức năng xem tiểu thuyết



Hình . Biểu đồ tuần tự chức năng xem tiểu thuyết

#### Use-case tìm kiếm (UC03)



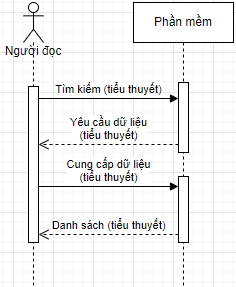
Hình . Use-case tìm kiếm

##### Đặc tả use-case tìm kiếm

|  |  |
| --- | --- |
| Thuộc tính | Mô tả |
| Tên chức năng | Xem tiểu thuyết |
| Tác nhân | Người đọc |
| Điều kiện trước |  |
| Điều kiện sau |  |
| Mô tả | Tìm kiếm tiểu thuyết thông qua tên tiểu thuyết |
| Kịch bản | Người đọc cần tìm kiếm tựa của một tiểu thuyết. Hoặc lọc tiểu thuyết theo thể loại mong muốn |
| Tương tác | |  |  | | --- | --- | | Hành động của tác nhân | Hồi đáp của hệ thống | | Yêu cầu tìm kiếm/lọc | Yêu cầu dữ liệu nhập (tiêu đề tiểu thuyết/thể loại) | |  | Hiện danh sách tiểu thuyết theo tiểu đề/danh sách tiểu thuyết theo thể loại | |

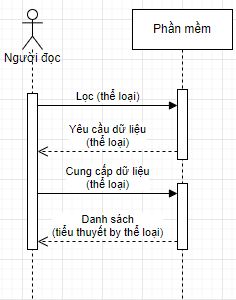
Bảng . Đặc tả use-case tìm kiếm

##### Biểu đồ tuần tự chức năng tìm kiếm



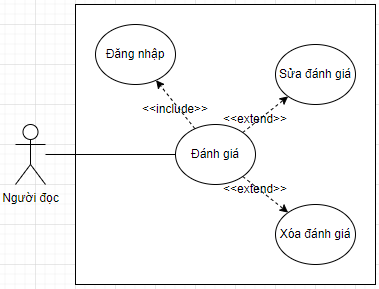
Hình . Biểu đồ tuần tự chức năng tìm kiếm

##### Biểu đồ tuần tự chức năng lọc theo thể loại



Hình . Biểu đồ tuần tự chức năng lọc theo thể loại

#### Use-case đánh giá tiểu thuyết (UC04)



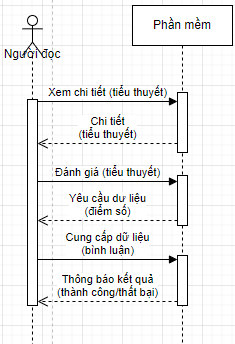
Hình . Use-case đánh giá tiểu thuyết

##### Đặc tả use-case đánh giá tiểu thuyết

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên chức năng | Đánh giá tiểu thuyết |
| Tác nhân | Người đọc |
| Điều kiện trước | Phải ở trạng thái đăng nhập |
| Điều kiện sau | Đánh giá tiểu thuyết đã có trên hệ thống |
| Mô tả | Đánh giá chất lượng của một tiểu thuyết.Điểm số đánh giá theo thang điểm 1 đến 5. Điểm số của tiểu thuyết bằng trung bình cộng tất cả dánh giá từ tất cả người đọc. Dữ liệu của đánh giá gồm có mã đánh giá, điểm số, ngày đánh giá. |
| Kịch bản | Người đọc đánh giá nội dung tiểu thuyết dựa trên cảm nhận cá nhân |
| Tương tác | |  |  | | --- | --- | | Hành động của tác nhân | Hồi đáp của hệ thống | | Người đọc (đã xác thực) Yêu cầu xem chi tiết tiểu thuyêt | Hiện chi tiết (tiểu thuyết) | | Đánh giá “sao” theo thang điểm từ 1 đến 5 | Nếu thành công thì thêm mới đánh giá vào cơ sở dữ liệu, hiện thông báo thành công. Nếu thất bại thì trả về thông báo thất bại | |

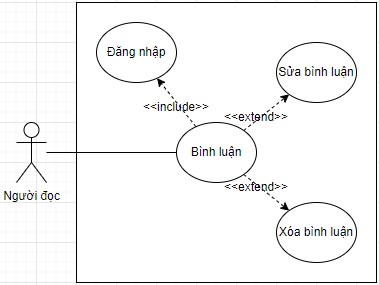
Bảng . Đặc tả use-case đánh giá tiểu thuyết

##### Biểu đồ tuần tự chức năng đánh giá tiểu thuyết



Hình . Biểu đồ tuần tự chức năng đánh giá tiểu thuyết

#### Use-case bình luận (UC05)



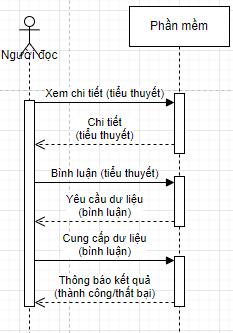
Hình . Use-case bình luận

##### Đặc tả use-case bình luận

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên chức năng | Bình luận tiểu thuyết |
| Tác nhân | Người đọc |
| Điều kiện trước | Phải ở trạng thái đăng nhập |
| Điều kiện sau | Bình luận tiểu thuyết đã có trên hệ thống |
| Mô tả | Nhập nội dung bình luận trên một tiểu thuyết để chia sẻ quan điểm của bản thân hoặc thảo luận với người đọc khác. Dữ liệu của bình luận gồm có mã bình luận, nội dung bình luận, ngày đăng và ngày chỉnh sửa |
| Kịch bản | Người đọc mong có thảo luận về nội dung tiểu thuyết với người đọc khác |
| Tương tác | |  |  | | --- | --- | | Hành động của tác nhân | Hồi đáp của hệ thống | | Người đọc (đã xác thực) Yêu cầu xem chi tiết tiểu thuyêt/ nội dung chương | Chi tiết (tiểu thuyết) | | Thêm bình luận | Nếu thành công thì thêm mới bình luận vào cơ sở dữ liệu, hiện thông báo thành công. Nếu thất bại thì trả về thông báo thất bại | |

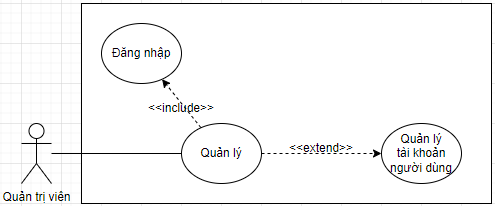
Bảng . Đặc tả use-case bình luận

##### Biểu đồ tuần tự chức năng bình luận



Hình . Biểu đồ tuần tự chức năng bình luận

#### Use-case quản lý tài khoản người dùng (UC06)



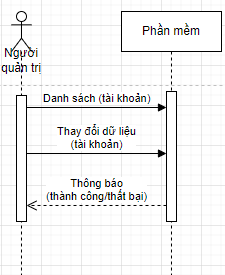
Hình . Use-case quản lý tài khoản người dùng

##### Đặc tả use-case quản lý tài khoản người dùng

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Tên chức năng | Quản lý tài khoản người dùng |
| Tác nhân | Người quản trị |
| Điều kiện trước | Phải ở trạng thái đăng nhập |
| Điều kiện sau | Người dùng có tài khoản bị khóa không thể tham gia đăng tiểu thuyết, đánh giá, bình luận |
| Mô tả | Quản lý tài khoản người dùng như thay đổi quyền, khóa tài khoản. Tài khoản gồm có các dư liệu tên đăng nhập, tên hiển thị, mật khẩu, email, quyền hạn. |
| Kịch bản | Người quản trị thay đổi quyền hoặc khóa tài khoản của người dùng |
| Tương tác | |  |  | | --- | --- | | Hành động của tác nhân | Hồi đáp của hệ thống | | Người quản trị (đã xác thực) Yêu cầu xem chi tiết tài khoản người dùng | Danh sách chi tiết (tài khoản) | | Thay đổi dữ liệu (Quyền/Khóa tài khoản) | Nếu thành công thì cập nhật dữ liệu vào cơ sở dữ liệu, hiện thông báo thành công. Nếu thất bại thì trả về thông báo thất bại | |

Bảng . Đặc tả use-case quản lý tài khoản người dùng

##### Biểu đồ tuần tự chức năng quản lý tài khoản người dùng



Hình . Biểu đồ tuần tự chức năng quản lý tài khoản người dùng

## Yêu cầu ràng buộc đối với phần mềm

### Yêu cầu từ môi trường nghiệp vụ

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-case** | **ID** | **Nội dung yêu cầu** | **Stake-Holder** |
| UC01 | F01.1 | Người dùng phải được chia sẻ thông tin  Người dùng được phép đăng tiểu thuyết, đánh giá, bình luận bất kì nội dung nào dưới dạng chữ, hình ảnh,… | Người dùng |
| UC01-01 | F01.2 | Nội dung được đăng tù người đọc phải được kiểm duyệt  Nội dung được đăng từ người dùng trước khi hiển thị đến người đọc khác phải đươc kiểm soát. Ví dụ: Tiểu thuyết được yêu cầu gỡ xuống vì vấn đề bản quyền. | Người quản trị, Luật bản quyền |
| UC06 | F01.3 | Người quản trị phải được quyền quản lý tài khoản  Người quản trị được phép thay đổi quyền hạn của tài khoản cũng như khóa tài khoản để hạn chế các hành vi vi phạm chính sách của website/luật pháp | Người quản trị |

### Yêu cầu từ môi trường vận hành

|  |  |  |  |
| --- | --- | --- | --- |
| **Use-case** | **ID** | **Nội dung yêu cầu** | **Stake-Holder** |
| UC01, UC05 | F01.1 | Trợ giúp người dùng viết nội dung  Tạo interface giúp đỡ người dùng điều chỉnh font, kích cỡ chữ, thêm hình ảnh, biểu tượng đặc biệt, emoji. Ví dụ: Ck Editor. | Website |
| UC02 | F01.2 | Trợ giúp người dùng xem bài viết  Người dùng phải xem được các bài viết do người dùng khác viết | Website |

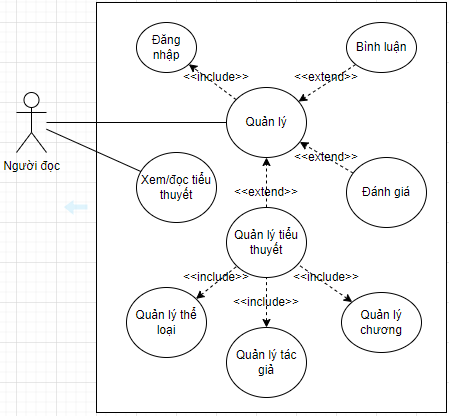
### Yêu cầu từ môi trường phát triển

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Đối tượng** | **Nội dung yêu cầu** | **Stake-Holder** |
| TA01 | Application | Server: Intel, Windows 10,  cpu I7, hdd 5T, mem 32G | Khách hàng |
| TA02 | Software | Phần mềm viết trên IntelliJ IDEA, sử dụng hệ quản trị cơ sở dữ liệu Postgre SQL | Lập trình viên |
| TA03 | Software | Dùng Spring framework cho back-end và front-end. | Lập trình viên |

# THIẾT KẾ HỆ THỐNG

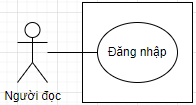
## Các use-case thiết kế

### Người đọc



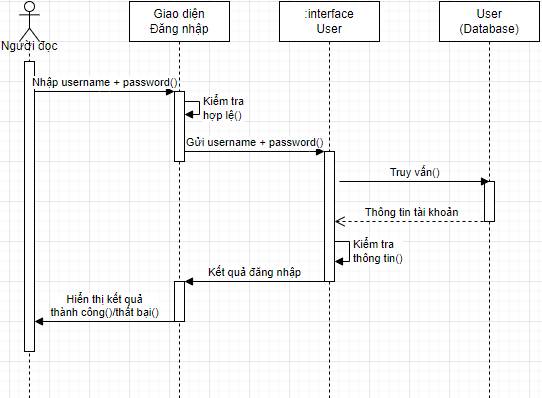
Hình . Use-case tổng quát của người đọc

#### Use-case đăng nhập (UD01)



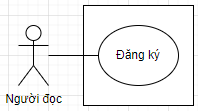
Hình . Use-case đăng nhập

**Sơ đồ tuần tự:**



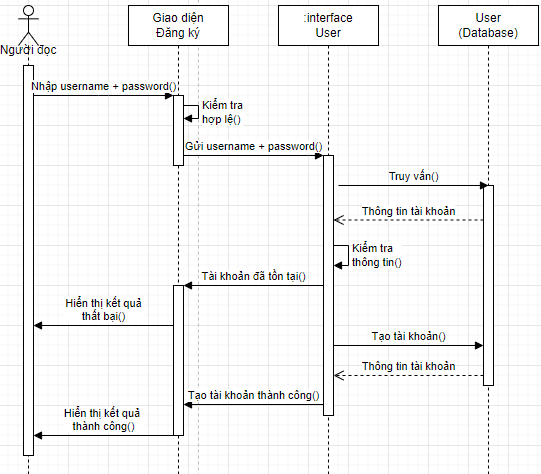
Hình . Sơ đồ tuần tự chức năng đăng nhập

#### Use-case đăng ký (UD02)



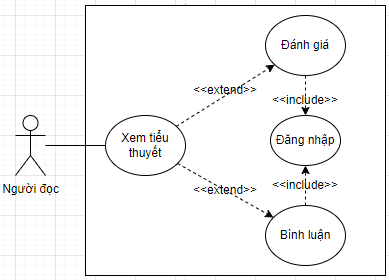
Hình . Use-case đăng ký

**Sơ đồ tuần tự:**



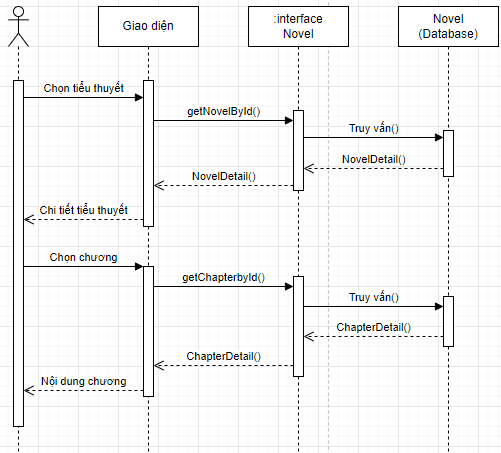
Hình . Sơ đồ tuần tự chức năng đăng ký

#### Use-case xem tiểu thuyết (UD03)



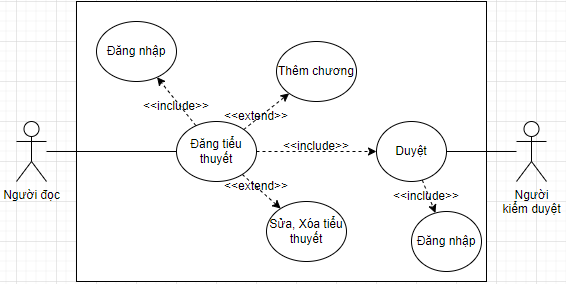
Hình . Use-case xem tiểu thuyết

**Sơ dồ tuần tự:**

****

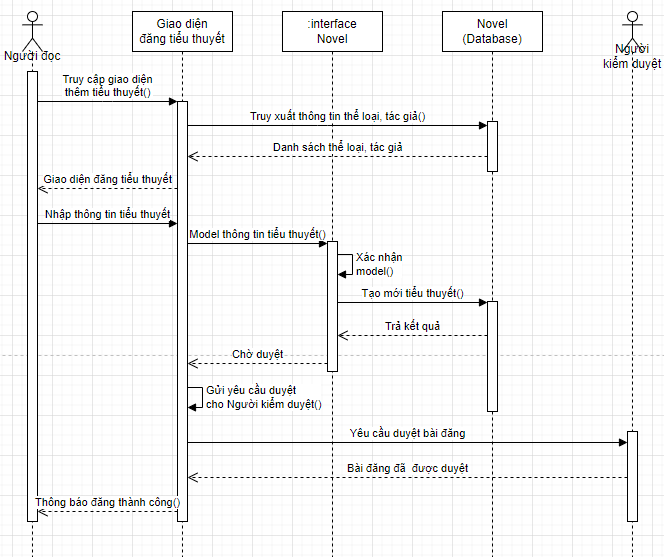
Hình . Sơ đồ tuần tự chức năng xem tiểu thuyết

#### Use-case đăng tiểu thuyết (UD04)



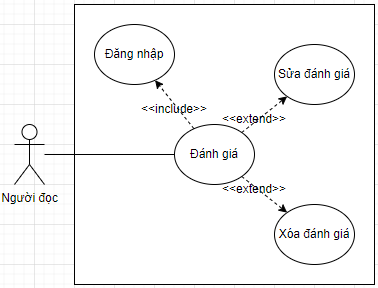
Hình . Use-case đăng tiểu thuyết

**Sơ đồ tuần tự:**

****

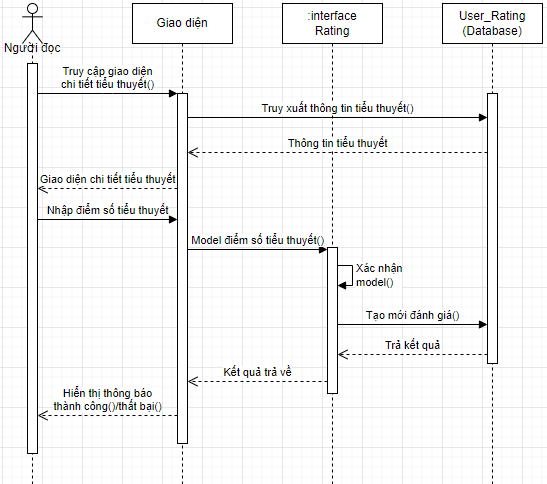
Hình . Sơ đồ tuần tự chức năng đăng tiểu thuyết

#### Use-case đánh giá (UD05)

****

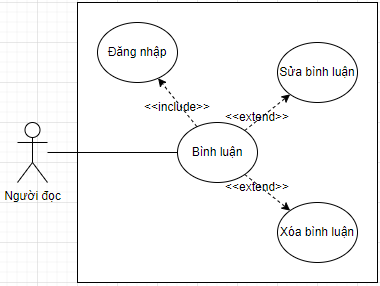
Hình . Use-case đánh giá

**Sơ đồ tuần tự:**



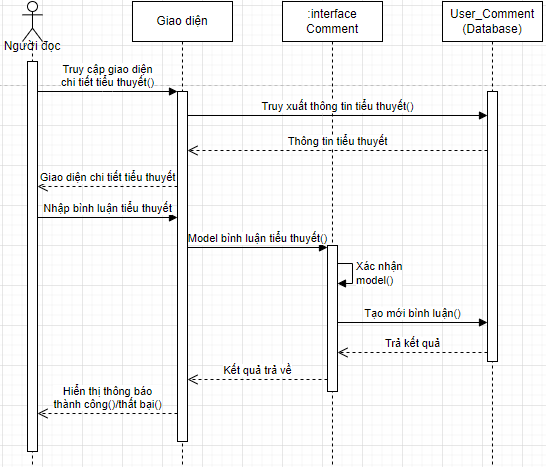
Hình . Sơ đồ tuần tự chức năng đánh giá

#### Use-case bình luận (UD06)



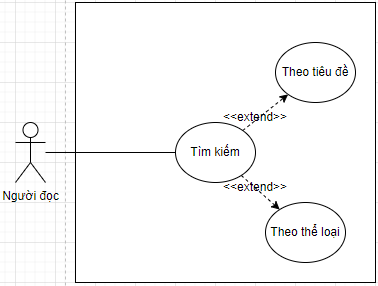
Hình . Use-case bình luận

**Sơ đồ tuần tự:**



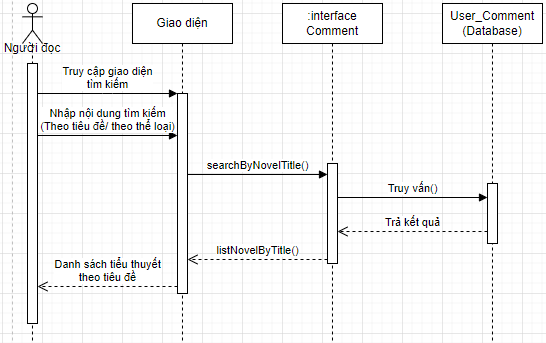
Hình . Sơ dồ tuần tự chức năng bình luận

#### Use-case tìm kiếm (UD07)



Hình . Use-case tìm kiếm

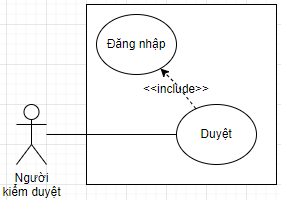
**Sơ dồ tuần tự:**



Hình . Sơ đồ tuần tự chức năng tìm kiếm

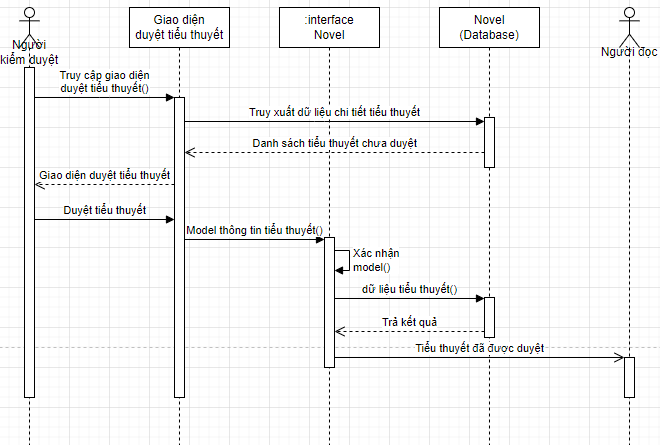
### Người kiểm duyệt

#### Use-case duyệt tiểu thuyết (UD08)



Hình . Use-case duyệt tiểu thuyết

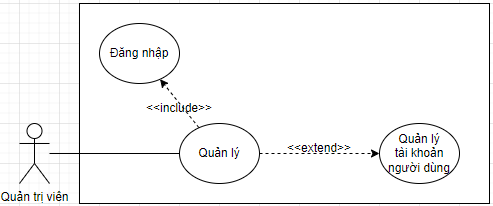
**Sơ đồ tuần tự:**

****

Hình . Sơ đồ tuàn tụ chức năng duyệt tiểu thuyết

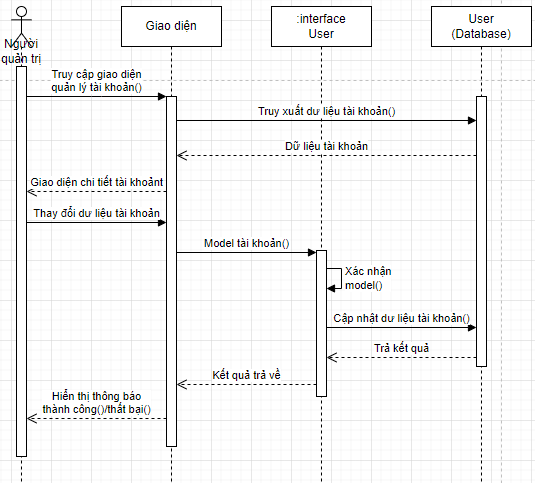
### Người quản trị

#### Use-case quản lý tài khoản (UD09)



Hình . Use-case quản lý tài khoản

Sơ đồ tuần tự:



Hình . Sơ đồ tuần tự chức năng quản lý tài khoản

## Thiết kế phần mềm để xử lý use-case

### UD01

#### Form: Đăng nhập

* Sử dụng cho: UD01

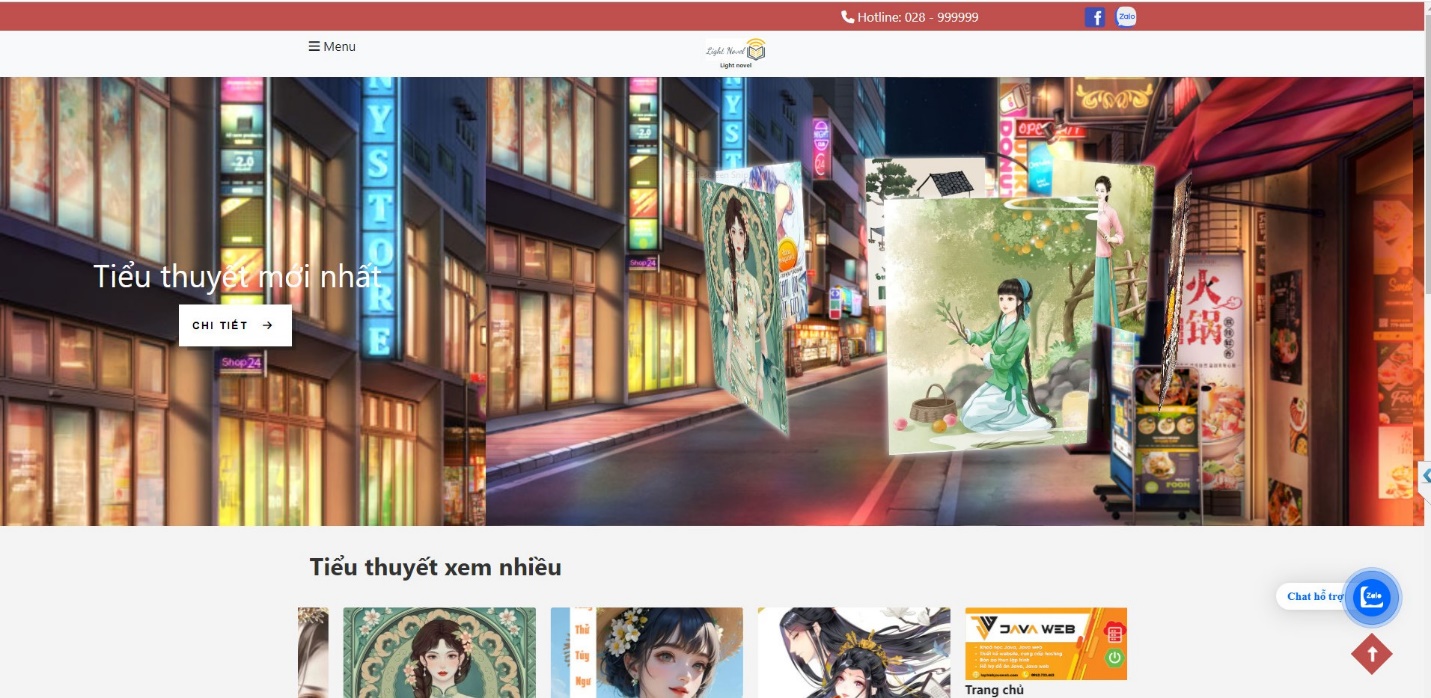


Hình . Form Login

* Users: Người đọc, người kiểm duyệt, người quản trị
* Inputs:

|  |  |
| --- | --- |
| **Control** | **Loại dữ liệu** |
| Text input Tên đăng nhập | Text |
| Password input Mật khẩu | Text |

* Outputs: Nếu đăng nhập thành công, redirect về form Home. Nếu không thành công hiển thị lỗi.



Hình . Form Home

* Xử lý: Dùng API 1.1 để kiểm tra dữ liệu

#### API 1.1:

* **Path**: /service/impl/CustomUserDetailService
* **Inputs**: request body

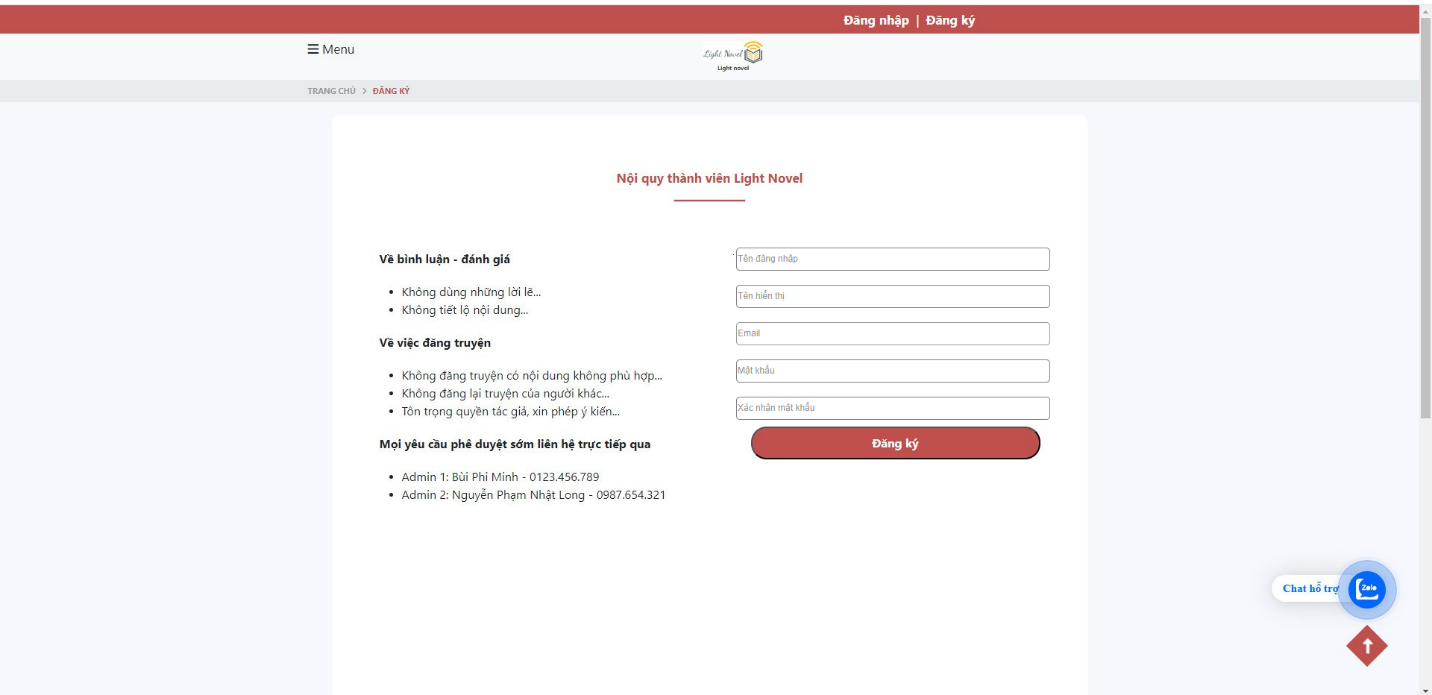
|  |  |
| --- | --- |
| **Biến** | **Mô tả** |
| User\_id | Id người dùng |
| User\_password | Mật khẩu |

* **Outputs**:
  + Mã trạng thái HTTP (200/400/500).
  + Message: Kết quả đăng nhập.
* **Xử** **lý**:
  + Kiểm tra dữ liệu đầu vào:
    - Các trường không được bỏ trống
    - Nếu kiểm tra không có lỗi thì thực hiện bước tiếp theo.
  + Truy cập đến table User trong cơ sở dữ liệu.
  + Khởi tạo một đối tượng UserEntity gồm các thuộc tính tương ứng với bộ dữ liệu truyền vào, đối tượng này tương ứng với table User trong cơ sở dữ liệu.
  + Sử dụng thư viện Security – phương thức loadUserByUsername để tìm kiếm tồn tại của user, xử lý này được đặt trong try-catch để kiểm tra có lỗi không xác định xảy ra trong quá trình đăng nhập hay không.
  + Nếu không tìm thấy, trả về status code 500 kèm message là “Username không tồn tại”
  + Nếu có Exception – lỗi không xác định xảy ra thì trả về status code là 500 kèm message là “Có lỗi xảy ra, vui lòng thử lại”.
  + Nếu không có lỗi xảy ra, thực hiện bước tiếp theo.
  + Sử dụng thư viện Authentication – phương thức onAuthenticationSuccess để lưu dữ liệu đăng nhập của user (username, role) vào session. Dữ liệu này sẽ cho phép user sử dụng các chức năng yêu cầu xác thực mà không cần đăng nhập lại.
  + Nếu không có bất kỳ lỗi nào xảy ra, quá trình đăng nhập thành công với status code 200 kèm message là “Đăng nhập thành công”.

### UD02

#### Form: Đăng ký

* Sử dụng cho: UD02



Hình . Form Signup

* Users: Người đọc
* Inputs:

|  |  |
| --- | --- |
| **Control** | **Loại dữ liệu** |
| Text input Tên đăng nhập | Text |
| Text input Tên hiển thị | Text |
| Password input Mật khẩu | Text |
| Password input Xác nhận mật khẩu | Text |
| Text input Email | Text |

* Outputs: Redirect về form Đăng nhập
* Xử lý: Dùng API 2.1 để kiểm tra dữ liệu

#### API 2.1:

* Path: /api/user/register
* Inputs: UserEntity

|  |  |  |
| --- | --- | --- |
| **Biến** | **Mô tả** | **Giá trị khởi tạo** |
| user\_id | Tên đăng nhập |  |
| user\_name | Tên hiển thị |  |
| user\_password | Mật khẩu |  |
| user\_email | Email |  |
| role\_id | Mã quyền | ‘1’(Người đọc) |

* Outputs:
* Mã trạng thái HTTP (200/400/500)
* Message: Kết quả đăng ký tài khoản
* Xử lý:
  + Truy cập đến table User trong cơ sở dữ liệu.
  + Khởi tạo một đối tượng UserEntity gồm các thuộc tính tương ứng với bộ dữ liệu truyền vào, đối tượng này tương ứng với table User trong cơ sở dữ liệu.
  + Kiểm tra tồn tại user\_id. Nếu đã tồn tại, trả về status code 400 kèm message “Tên đăng nhập đã được sử dụng”.
  + Kiểm trả tồn tại user\_email. Nếu đã tồn tại, trả về status code 400 kèm message “Email đã được sử dụng”.
  + Sử dụng thư viện JPA Repository – phương thức save(UserEntity) để lưu dư liệu tài khoản người dùng vào cơ sở dữ liệu, xử lý này được đặt trong try-catch để kiểm tra có lỗi không xác định xảy ra trong quá trình tạo bình luận hay không.
  + Nếu có Exception – lỗi không xác định xảy ra thì trả về code là 500 kèm message là “Có lỗi xảy ra trong quá trình tạo tài khoản”
  + Nếu không có bất kỳ lỗi nào xảy ra, quá trình sửa tài khoản thành công với status code 200 kèm message là “Tạo tài khoản thành công”.

### UD03

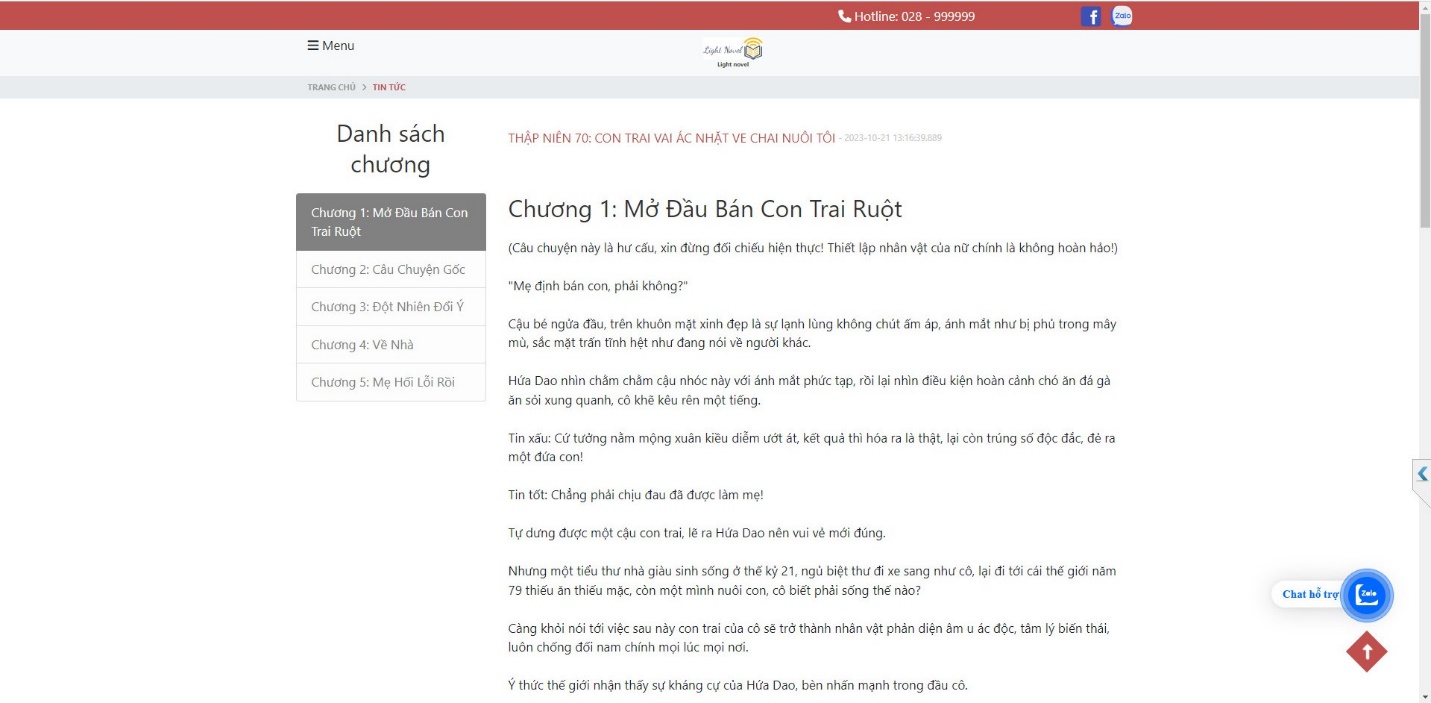
#### Form: Chi tiết tiểu thuyết

* Sử dụng cho: UD03, UD05



Hình . Form Novel Detail

* Users: Người đọc
* Outputs: Dữ liêu tiểu thuyết: mã tiểu thuyết (hidden), tên tiểu thuyết, thể loại, tác giả, người đăng, lượt xem, điểm đánh giá, danh sách chương.
* Xử lý: Dùng API 3.1 để tải dữ liệu tiểu thuyết.



Hình . Form Chapter Detail

* Sử dụng cho: xem nội dung chương
* Users: Người đọc
* Outputs: Danh sách chương, dữ liệu chương: Mã chương (hidden), Tên chương, Nội dung chương, Mã tiểu thuyết (hidden), Tên tiểu thuyết.
* Xử lý: Dùng API 3.2 để tải dữ liệu

#### API 3.1:

* Path: /repository/PostRepository.java
* Method: GET
* Inputs:

|  |  |
| --- | --- |
| **Biến** | **Mô tả** |
| novel\_id | Mã tiểu thuyết |

* Outputs: NovelEntity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Biến** | | **Mô tả** | | |
| novel\_id | | | Mã tiểu thuyết | |
| novel\_title | | Tiêu đề tiểu thuyết | | |
| novel\_cover | | Ảnh bìa | | |
| novel\_description | | Tóm tắt | | |
| novel\_view | | Lượt xem | |
| UserEntity | user\_id | Tên đăng nhập | |
| user\_name | Tên hiển thị | |
| List<GenreEntity> | genre\_id | Mã thể loại | |
| genre\_name | Tên thể loại | |
| List<AuthorEntity> | author\_id | Mã tác giả | |
| author\_name | Tên tác giả | |
| RatingEntity | rating\_id | Mã đánh giá | |
| rating\_score | Điểm đánh giá | |
| List<ChapterEntity>. | chapter\_id | Mã chương | |
| chapter\_title | Tiêu đề chương | |

* Xử lý:
* Truy cập đến table Novel trong cơ sở dũ liệu
* Khởi tạo một đối tượng NovelEntity gồm các thuộc tính tương ứng với dư liệu truyền vào, đối tượng này tương ứng với table Novel trong cơ sở dữ liệu.
* Sử dụng thư viện JPA repository – phương thức findByIdAndApprovedTrue để lấy ra dữ liệu tiểu thuyết từ cơ sở dữ liệu.
* Dữ liệu đầu ra được mô tả như trên.
* Nếu không tìm thấy sẽ trả về mã lỗi 404 kèm message “Không tìm thấy dữ liệu yêu cầu”.

#### API 3.2:

* Path: /repository/ChapterRepository
* Method: GET
* Inputs:

|  |  |
| --- | --- |
| **Biến** | **Mô tả** |
| novel\_id | Mã tiểu thuyết |
| chapter\_id | Mã chương |

* Outputs: NovelEntity

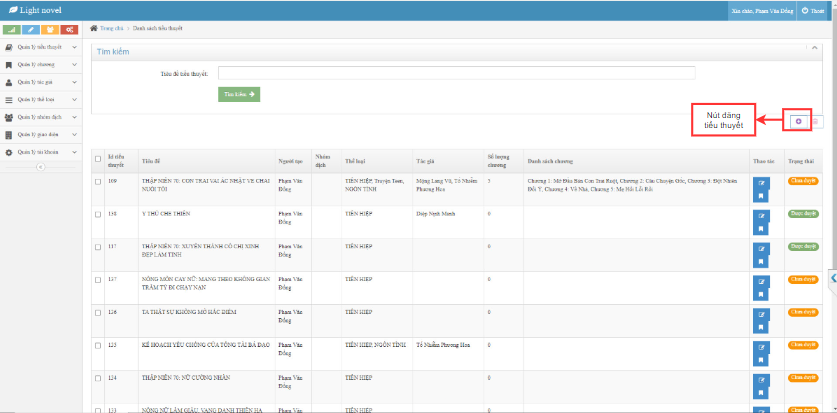
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Biến** | | **Mô tả** | | |
| chapter\_id | | | Mã chương | |
| chapter\_title | | Tiêu đề chương | | |
| chapter\_content | | Nôi dung chương | | |
| NovelEntity | novel\_id | Mã tiểu thuyết | |
| novel\_title | Tiểu đề tiểu thuyết | |

* Xử lý:
  + Truy cập đến table Chapter trong cơ sở dũ liệu.
  + Khởi tạo một đối tượng ChapterEntity gồm các thuộc tính tương ứng với dư liệu truyền vào, đối tượng này tương ứng với table Chapter trong cơ sở dữ liệu.
  + Sử dụng thư viện JPA repository – phương thức findByIdAndApprovedTrue để lấy ra dữ liệu chương từ cơ sở dữ liệu.
  + Nếu không tìm thấy sẽ trả về mã lỗi 404 kèm message “Không tìm thấy dữ liệu yêu cầu”.

### UD04

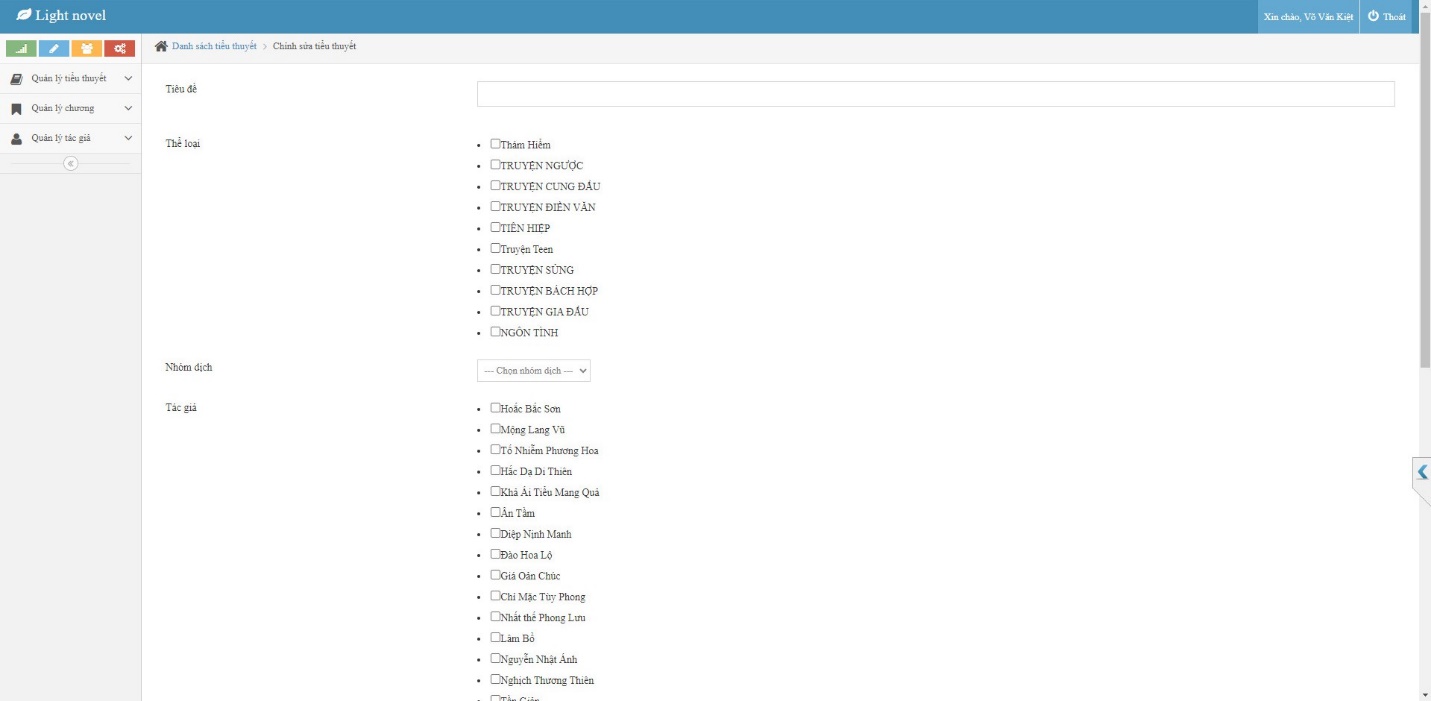
#### Form: Đăng tiểu thuyết

* Sử dụng cho: UD04



Hình . Form List Novels

* Users: Người đăng
* Inputs: Người dùng nhấn nút đăng tiểu thuyết
* Xử lý: Nhảy giao diện.



* Inputs:

|  |  |
| --- | --- |
| **Control** | **Loại dư liệu** |
| Text input Tiêu đề | Text |
| Checkboxes Thể loại | int |
| Checkboxes Tác giả | int |
| File selectorẢnh bìa | File |
| Text input Mô tả | Text |

* Outputs: Thông báo thành công hoặc thông báo lỗi
* Xử lý: Dùng API 4.1 để ghi dư liệu.

#### API 4.1

* Path: /service/PostService.java
* Method: POST
* Inputs:

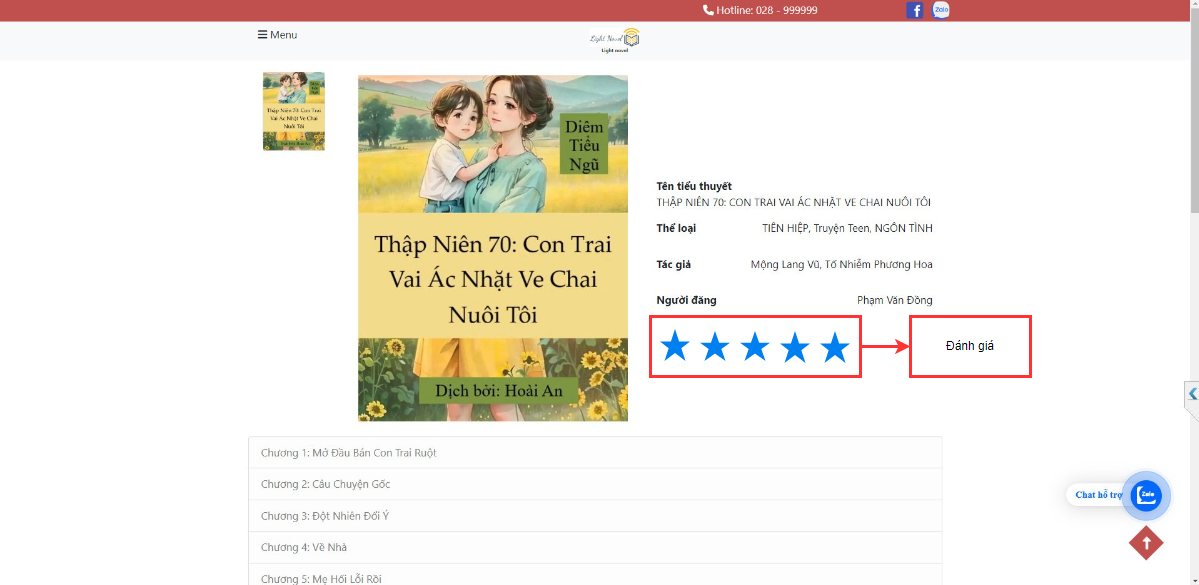
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Biến** | | **Mô tả** | **Giá trị khởi tạo** | |
| novel\_id | | Mã tiểu thuyết |  | |
| novel\_title | | Tiêu đề tiểu thuyết |  | |
| novel\_cover | | Ảnh bìa |  | |
| novel\_datepost | | Ngày đăng | Ngày hiện tại | |
| novel\_description | | Tóm tắt |  | |
| Novel\_view | | Lượt xem | 0 |
| isApproved | | Đã duyệt | false |
| user\_id | | Tên đăng nhập |  |
| List<GenreEntity> | genre\_id | Mã thể loại |  |
| genre\_name | Tên thể loại |  |
| List<AuthorEntity> | author\_id | Mã tác giả |  |
| author\_name | Tên tác giả |  |
| RatingEntity | rating\_id | Mã đánh giá |  |
| rating\_score | Điểm đánh giá | 0 |

* Outputs:
  + Mã trạng thái HTTP (200/400/500)
  + Message: Kết quả đăng tiểu thuyết
* Xử lý:
  + Kiểm tra dữ liệu đầu vào:
    - Các trường không được bỏ trống.
    - Nếu có dữ liệu vi phạm các kiểm tra ở trên thì trả về code là 400 kèm message “Bộ dữ liệu đầu vào có thông tin chưa hợp lệ”
    - Nếu kiểm tra không có lỗi thì thực hiện bước tiếp theo.
  + Truy cập đến table Novel trong cơ sở dữ liệu.
  + Khởi tạo một đối tượng NovelEntity gồm các thuộc tính tương ứng với bộ dữ liệu truyền vào, đối tượng này tương ứng với table Novel trong cơ sở dữ liệu.
  + Ảnh được mã hóa dưới định dạng Base64 trước khi ghi vào NovelEntity.
  + Nếu người đăng là quản trị viên, thuộc tính isApproved = ‘true’
  + Sử dụng thư viện JPA Repository – phương thức save(PostEntity) để lưu dư liệu tiểu thuyết vào cơ sở dữ liệu, xử lý này được đặt trong try-catch để kiểm tra có lỗi không xác định xảy ra trong quá trình tạo tiểu thuyết hay không.
  + Nếu có Exception – lỗi không xác định xảy ra thì trả về code là 500 kèm message là “Có lỗi xảy ra trong quá trình tạo tiểu thuyết”
  + Nếu không có bất kỳ lỗi nào xảy ra, quá trình tạo tiểu thuyết thành công với status code 200 kèm message là “Tạo tiểu thuyết thành công, đang chờ duyệt”.

### UD05

#### Form: Chi tiết tiểu thuyết

* Sử dụng cho: UD03, UD05



* Users: Người đọc
* Inputs: Click chuột chọn từ 1 đến 5 sao tương đương với 1 đén 5 điểm

|  |  |
| --- | --- |
| **Control** | **Loại dư liệu** |
| Sao dánh giá | int |

* Outputs: Nếu thành công, highlight số sao người dùng đã chọn. Nêu thất bại trả về mã lỗi.
* Xử lý: Dùng API 5.1 để ghi dữ liệu

#### API 5.1

* Path: /service/ReviewService.java
* Method: POST
* Input:

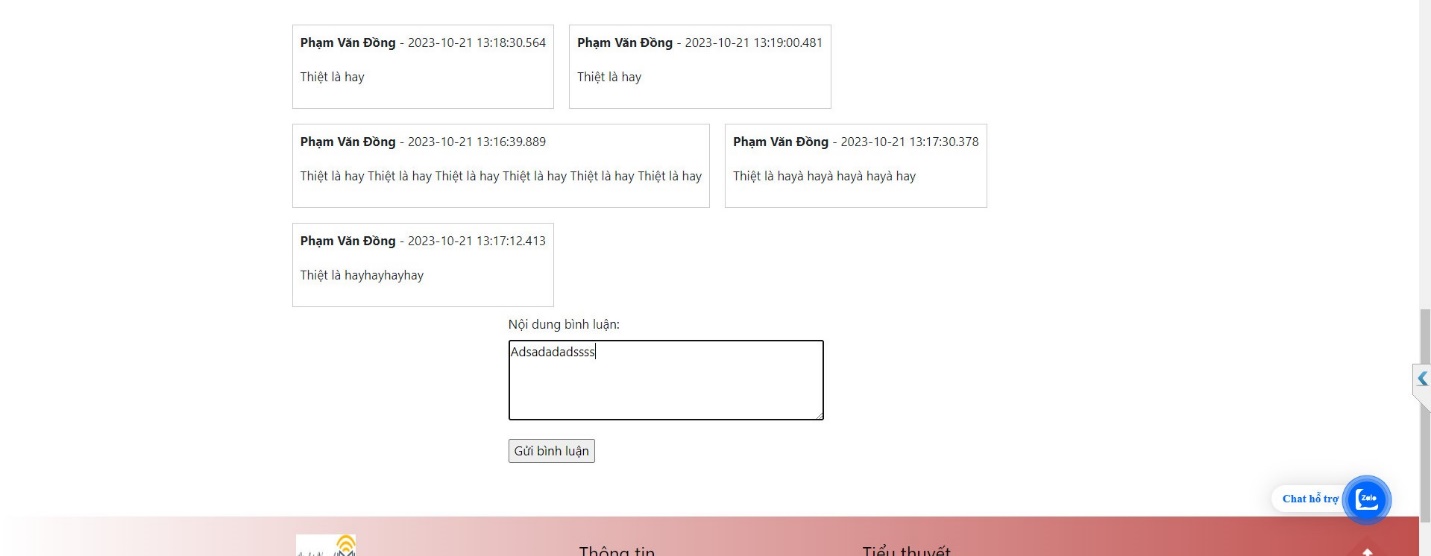
|  |  |  |
| --- | --- | --- |
| **Biến** | **Mô tả** | **Giá trị khởi tạo** |
| rating\_id | Mã đánh giá |  |
| rating\_score | Điểm số |  |
| rating\_datepost | Ngày đánh giá | Ngày hiện tại |
| novel\_id | Mã tiểu thuyết |  |
| user\_id | Tên đang nhập |  |

* Outputs:
  + Mã trạng thái HTTP (200/400/500)
  + Message: Kết quả đánh giá tiểu thuyết
* Xử lý:
  + Truy cập đến table Rating trong cơ sở dữ liệu.
  + Khởi tạo một đối tượng RatingEntity gồm các thuộc tính tương ứng với bộ dữ liệu truyền vào, đối tượng này tương ứng với table Rating trong cơ sở dữ liệu.
  + Sử dụng thư viện JPA Repository – phương thức save(RatingEntity) để lưu dư liệu đánh giá vào cơ sở dữ liệu, xử lý này được đặt trong try-catch để kiểm tra có lỗi không xác định xảy ra trong quá trình tạo đánh giá hay không.
  + Nếu có Exception – lỗi không xác định xảy ra thì trả về code là 500 kèm message là “Có lỗi xảy ra trong quá trình thêm đánh giá”
  + Nếu không có bất kỳ lỗi nào xảy ra, quá trình đánh giá thành công với status code 200 kèm message là “Thêm đánh giá thành công”.

### UD06

#### Form: Chi tiết chương

* Sử dụng cho: UD06



Hình . Form Chapter Detail

* Users: Người đọc
* Inputs:

|  |  |
| --- | --- |
| **Control** | **Loại dư liệu** |
| Text input Nội dung bình luận | text |

* Outputs: Thông báo thành công/ thất bại. Nếu thành công, bình luận công khai với người đọc khác. Nếu thất bại trả về mã lỗi
* Xử lý: Dùng API 6.1 để ghi dữ liệu

#### API 6.1

* Path: /servive/CommentService.java
* Method: POST
* Inputs:

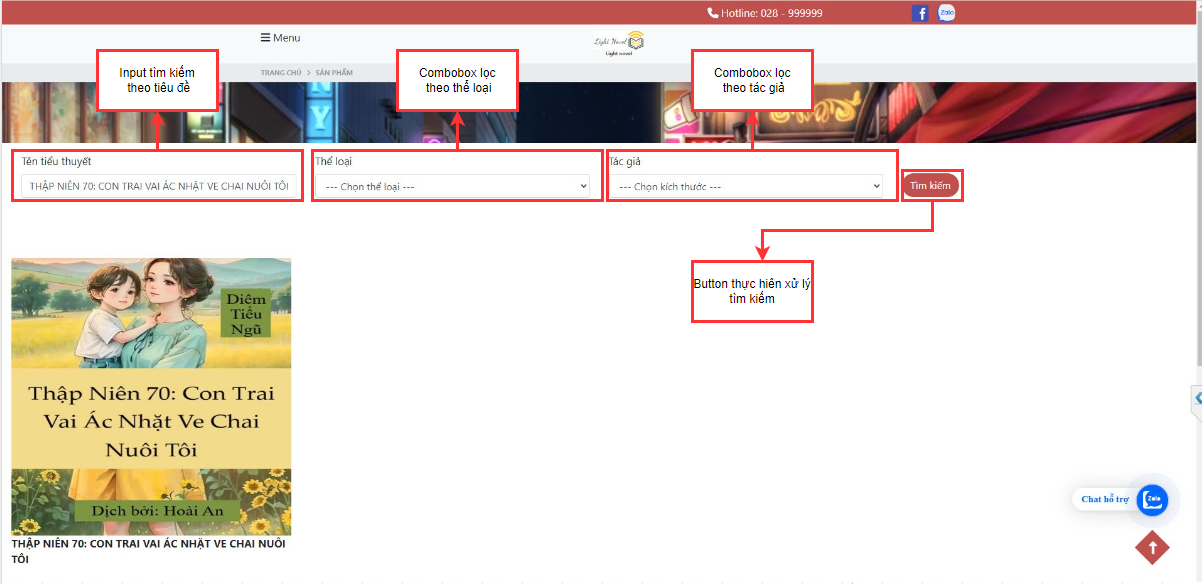
|  |  |  |
| --- | --- | --- |
| **Biến** | **Mô tả** | **Giá trị khởi tạo** |
| comment\_id | Mã đánh giá |  |
| comment\_content | Điểm số |  |
| comment\_datepost | Ngày đánh giá | Ngày hiện tại |
| comment\_dateedit | Ngày chỉnh sửa | Null |
| novel\_id | Mã tiểu thuyết |  |
| user\_id | Tên đang nhập |  |

* Outputs:
* Mã trạng thái HTTP (200/400/500)
* Message: Kết quả đánh giá tiểu thuyết
* Xử lý:
  + Truy cập đến table Comment trong cơ sở dữ liệu.
  + Khởi tạo một đối tượng CommentEntity gồm các thuộc tính tương ứng với bộ dữ liệu truyền vào, đối tượng này tương ứng với table Comment trong cơ sở dữ liệu.
  + Sử dụng thư viện JPA Repository – phương thức save(CommentEntity) để lưu dư liệu bình luận vào cơ sở dữ liệu, xử lý này được đặt trong try-catch để kiểm tra có lỗi không xác định xảy ra trong quá trình tạo bình luận hay không.
  + Nếu có Exception – lỗi không xác định xảy ra thì trả về code là 500 kèm message là “Có lỗi xảy ra trong quá trình thêm bình luận”
  + Nếu không có bất kỳ lỗi nào xảy ra, quá trình bình luận thành công với status code 200 kèm message là “Thêm bình luận thành công”.

### UD07

#### Form: Danh sách tiểu thuyết

* Sử dụng cho: UD07



Hình . Form danh sách tiểu thuyết

* Users: Người đọc
* Inputs:

|  |  |
| --- | --- |
| **Control** | **Loại dữ liệu** |
| Text input tìm theo tiêu đề | Text |
| Combobox lọc theo thể loại | Combobox |
| Combobox lọc theo tác giả | Combobox |

* Outputs: Danh sách tiểu thuyết theo lựa chọn
* Xử lý: Dung API 7.1 để lọc dữ liệu

#### API

* Path: /service/PostService.java
* Method: GET
* Inputs:

|  |  |
| --- | --- |
| **Biến** | **Mô tả** |
| novel\_title | Tiêu đề tiểu thuyết |
| genre\_id | Mã thể loại |
| author\_id | Mã tác giả |

* Outputs: List<NovelEntity>

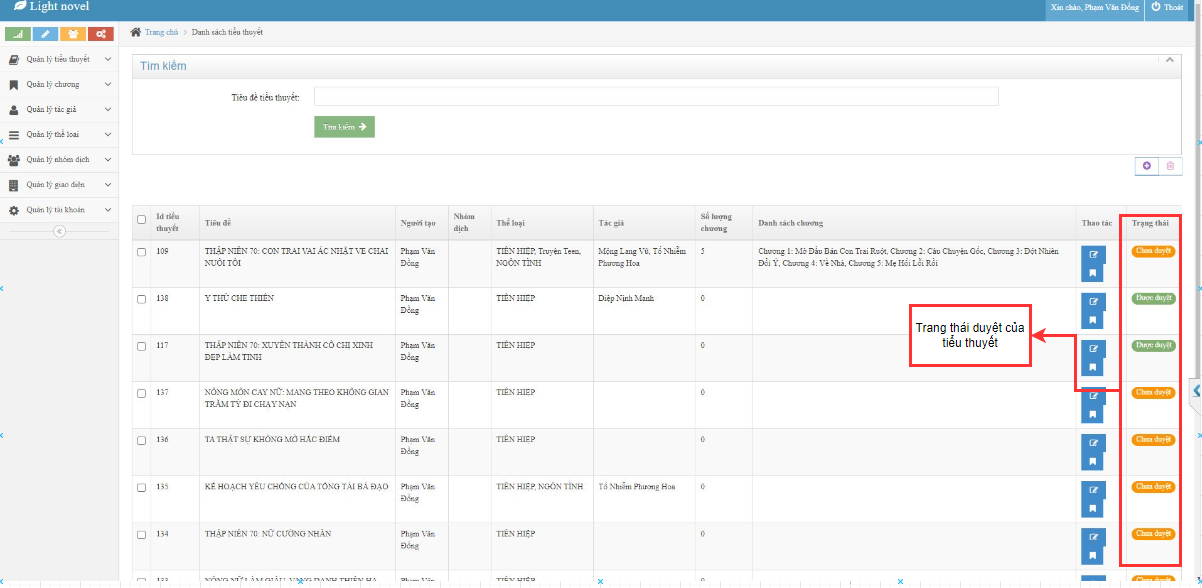
|  |  |  |  |
| --- | --- | --- | --- |
| **Biến** | **Mô tả** | | |
| novel\_id | | Mã tiểu thuyết | |
| novel\_title | Tiêu đề tiểu thuyết | | |
| novel\_cover | Ảnh bìa | | |
| novel\_description | Tóm tắt | | |
| novel\_view | Lượt xem | |
| isApproved | Trạng thái duyệt | |
| genre\_id | Mã thể loại | |
| author\_id | Mã tác giả | |
| user\_id | Người đăng | |

* Xử lý:
* Kiểm tra dữ liệu đầu vào:
* Nếu tất cả cá trường để không có dữ liệu, tải lại trang với danh sách sắp xếp theo A-Z
* Nếu bất kỳ trường nào trống, lọc theo các trường đã nhập/chọn.
* Truy cập đến table Novel trong cở sở dữ liệu
* Khởi tạo một đối tượng NovelEntity gồm các thuộc tính tương ứng với dư liệu truyền vào, đối tượng này tương ứng với table Novel trong cơ sở dữ liệu.
* Sử dụng hàm tạo thủ công filter(novel\_title, genre\_id, author\_id) để lấy ra dữ liệu tiểu thuyết từ cơ sở dữ liệu:
* Nếu bỏ trống thể loại và tác giả, sử dụng phương thức findByShortTitleContainingIgnoreCaseAndApprovedIsTrue(novel\_title) để trả về danh sách.
* Nếu bỏ trống thể loại, sử dụng phương thức findByAuthors\_CodeAndShortTitleContainingIgnoreCaseAndApprovedIsTrue(author\_id, novel\_id) để trả về danh sách.
* Nêu bỏ trống tác giả, sử dụng phương thức findByTags\_CodeAndShortTitleContainingIgnoreCaseAndApprovedIsTrue(genre\_id, novel\_id) để trả về danh sách.
* Nếu không bỏ trống trường nào, sử dụng phương thức findByTags\_CodeAndAuthors\_CodeAndShortTitleContainingIgnoreCaseAndApprovedIsTrue(genre\_id, author\_id, novel\_id) để trả về danh sách.
* Cuối cùng, sắp xếp danh sách theo A-Z.
* Nếu không tìm thấy sẽ trả về message “Không tìm thấy tiểu thuyết”.

### UD08

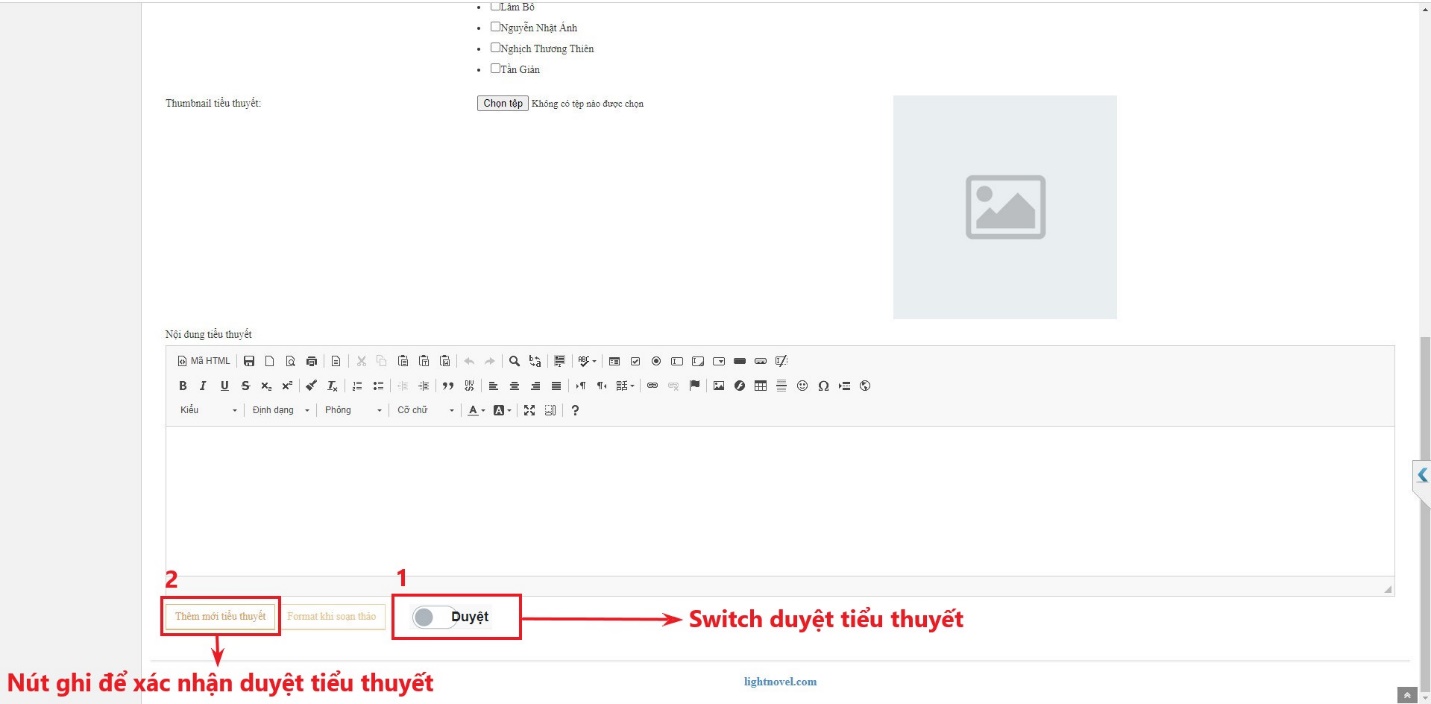
#### Form: Duyệt tiểu thuyết

* Sử dụng cho: UD08



Hình . Form duyệt tiểu thuyết

* Users: Người kiểm duyệt
* Inputs: Người dùng nhấn chọn tiểu thuyết chưa được duyệt.
* Xử lý: Nhảy giao diện đến chỉnh sửa tiểu thuyết



Hình . Form chỉnh sửa tiểu thuyết

* Inputs:

|  |  |
| --- | --- |
| **Control** | **Loại dư liệu** |
| Switch Duyệt | Boolean |

* Outputs: Thông báo thành công hoặc thông báo lỗi.
* Xử lý: Gọi API 8.1 để ghi dữ liệu.

#### API 8.1

* Path: /service/PostService.java
* Method: PUT
* Inputs:

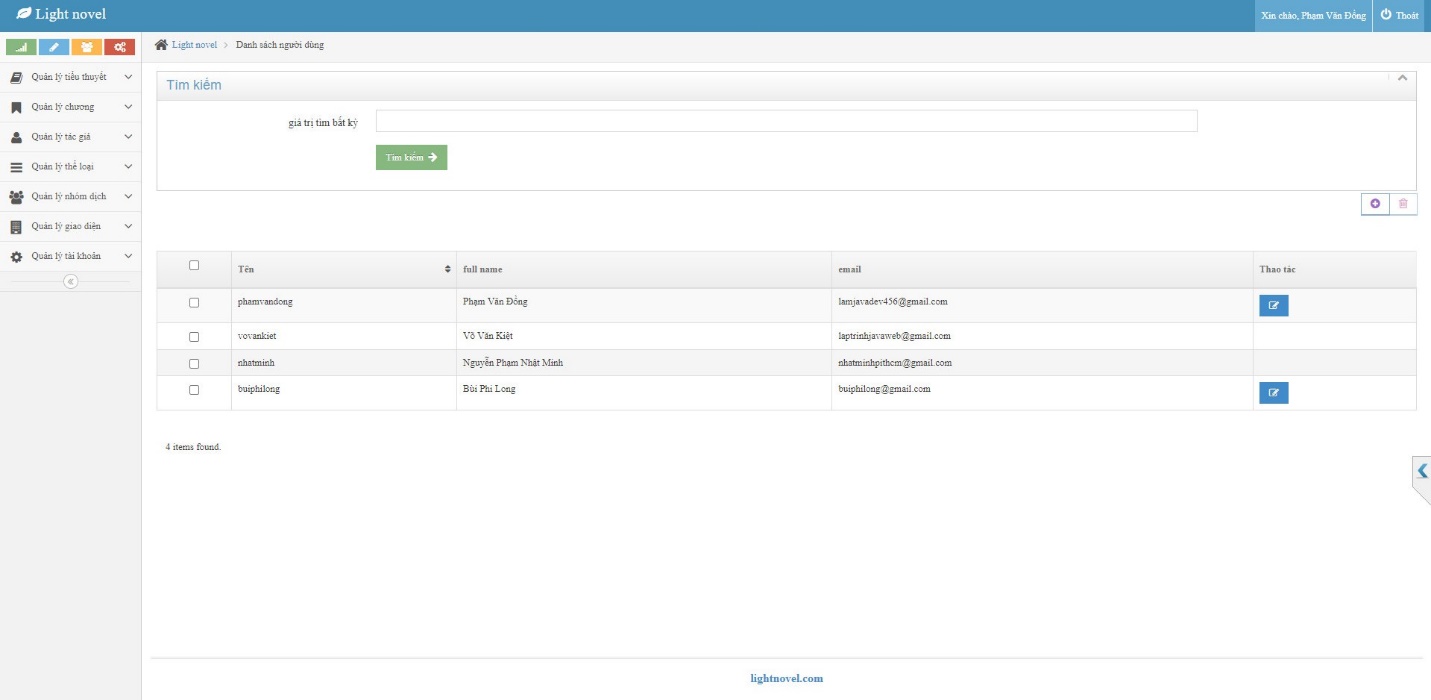
|  |  |  |  |
| --- | --- | --- | --- |
| **Biến** | **Mô tả** | **Giá trị khởi tạo** | |
| novel\_id | Mã tiểu thuyết | Lấy từ CSDL | |
| novel\_title | Tiêu đề tiểu thuyết | Lấy từ CSDL | |
| novel\_cover | Ảnh bìa | Lấy từ CSDL | |
| novel\_description | Tóm tắt | Lấy từ CSDL | |
| novel\_view | Lượt xem | Lấy từ CSDL |
| isApproved | Trạng thái duyệt | Lấy từ CSDL |
| genre\_id | Mã thể loại | Lấy từ CSDL |
| author\_id | Mã tác giả | Lấy từ CSDL |
| user\_id | Người đăng | Lấy từ CSDL |

* Outputs:
  + Mã trạng thái HTTP (200/400/500)
  + Message: Kết quả duyệt tiểu thuyết
* Xử lý:
  + Truy cập đến table Novel trong cơ sở dữ liệu.
  + Khởi tạo một đối tượng NovelEntity gồm các thuộc tính tương ứng với bộ dữ liệu lấy từ cơ sở dữ liệu, đối tượng này tương ứng với table Novel trong cơ sở dữ liệu.
  + Sử dụng thư viện JPA Repository – phương thức save(PostEntity) để lưu dư liệu tiểu thuyết vào cơ sở dữ liệu, xử lý này được đặt trong try-catch để kiểm tra có lỗi không xác định xảy ra trong quá trình tạo tiểu thuyết hay không.
  + Nếu có Exception – lỗi không xác định xảy ra thì trả về code là 500 kèm message là “Có lỗi xảy ra trong quá trình sửa tiểu thuyết”
  + Nếu không có bất kỳ lỗi nào xảy ra, quá trình duyệt tiểu thuyết thành công với status code 200 kèm message là “Chỉnh sửa tiểu thuyết thành công”.

### UD09

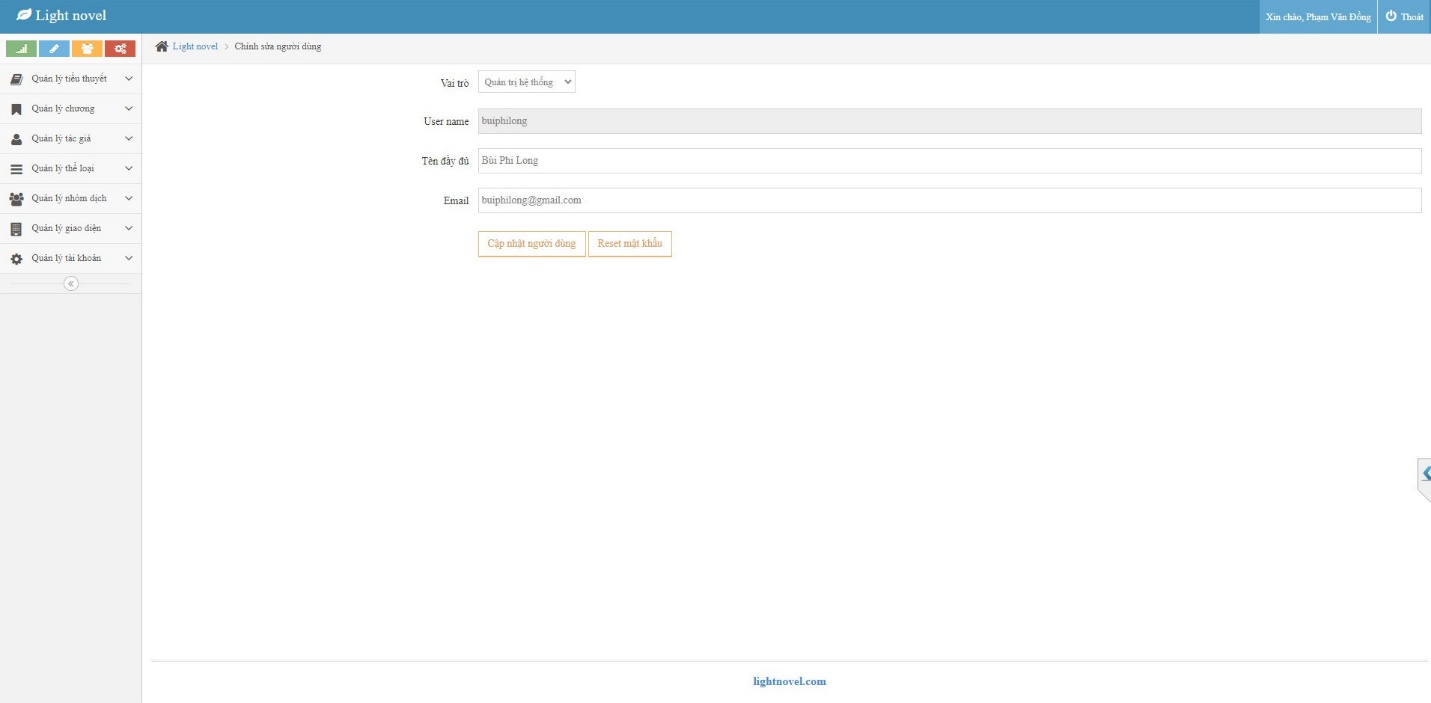
#### Form: Chi tiết tài khoản người dùng

* Sử dụng cho: UD09



Hình . Form danh sách tài khoản

* Users: Người quản trị
* Inputs: Người quản trị nhấn chọn chỉnh sửa tài khoản.
* Xử lý: Nhảy giao diện đến chỉnh sửa tài khoản



Hình . Form chỉnh sửa tài khoản

* Inputs:

|  |  |
| --- | --- |
| **Control** | **Loại dữ liệu** |
| Combobox Quyền | Int |
| Text input Tên hiển thị | Text |
| Text input Email | Text |

* Outputs: Thông báo thành công/thất bại. Nếu thành công hiển thị thông báo “Cập nhật dữ liệu thành công”. Nếu thất bại trả về mã lỗi
* Xử lý: Dung API 9.1 để ghi dữ liệu vào databasse

#### API 9.1

* Path: /service/UserService.java
* Method: PUT
* Inputs: UserEntity

|  |  |  |
| --- | --- | --- |
| **Biến** | **Mô tả** | **Giá trị khởi tạo** |
| user\_id | Tên đang nhập |  |
| user\_name | Tên hiển thị |  |
| user\_password | Mật khẩu |  |
| user\_email | Email |  |
| role\_id | Mã quyền |  |

* Output:
* Mã trạng thái HTTP (200/400/500)
* Message: Kết quả đánh giá tiểu thuyết
* Xử lý:
  + Truy cập đến table User trong cơ sở dữ liệu.
  + Khởi tạo một đối tượng UserEntity gồm các thuộc tính tương ứng với bộ dữ liệu truyền vào, đối tượng này tương ứng với table User trong cơ sở dữ liệu.
  + Sử dụng thư viện JPA Repository – phương thức save(UserEntity) để lưu dư liệu tài khoản người dùng vào cơ sở dữ liệu, xử lý này được đặt trong try-catch để kiểm tra có lỗi không xác định xảy ra trong quá trình tạo bình luận hay không.
  + Nếu có Exception – lỗi không xác định xảy ra thì trả về code là 500 kèm message là “Có lỗi xảy ra trong quá trình cập nhật tài khoản”
  + Nếu không có bất kỳ lỗi nào xảy ra, quá trình sửa tài khoản thành công với status code 200 kèm message là “Cập nhật tài khoản thành công”.

# XÂY DỰNG CƠ SỞ DỮ LIỆU

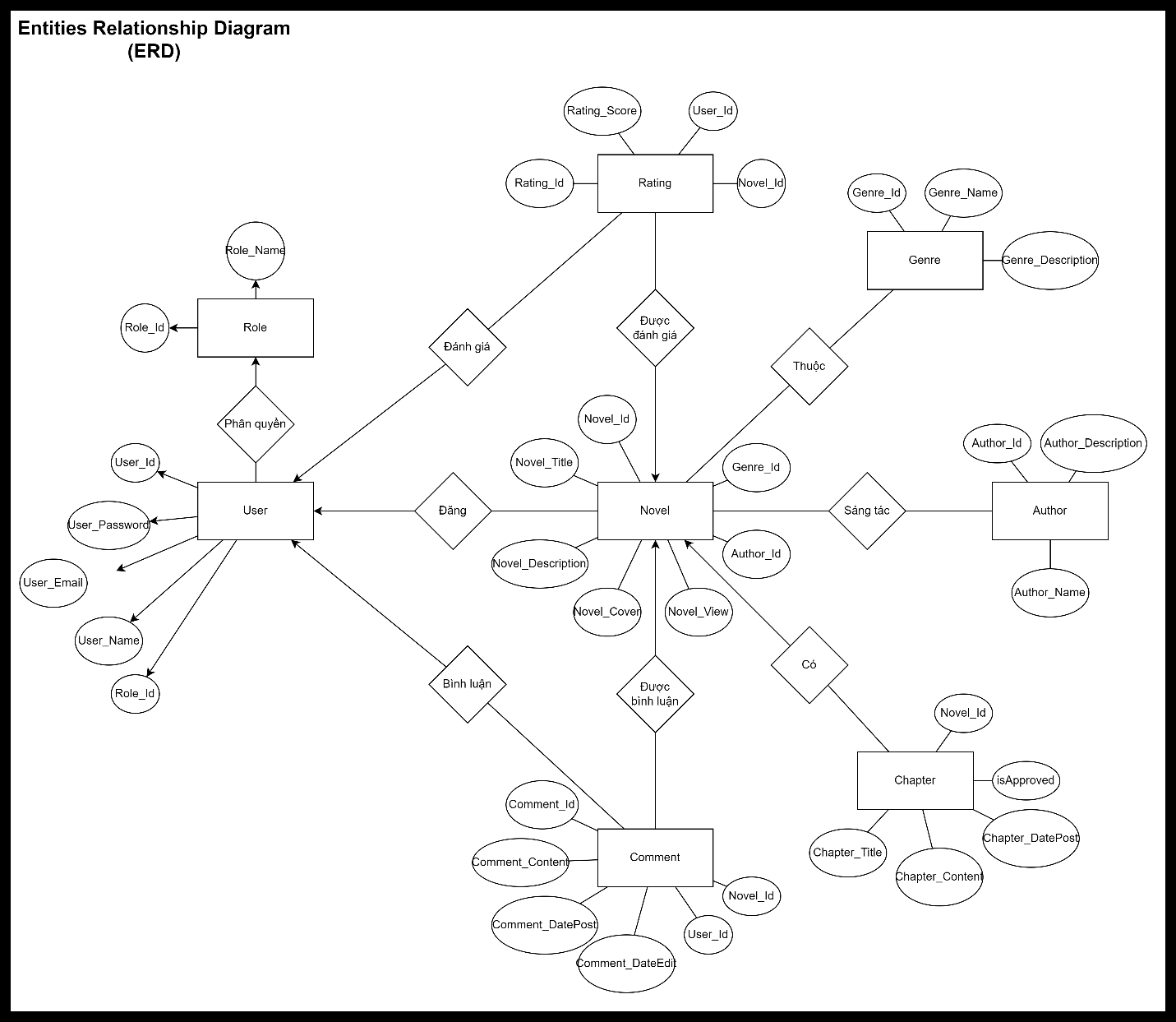
## Thiết kế cơ sở dữ liệu quan hệ cho website đọc truyện tiểu thuyết

Thiết kế cơ sở dữ liệu cho website đọc truyện tiểu thuyết:

Quy ước: Khóa chính: **In đậm + gạch chân**; Khóa ngoại: **In đậm**

* Tiểu thuyết có các thuộc tính: **Mã tiểu thuyết**, Tên tiểu thuyết, Ngày đăng, Mô tả sơ nội dung, Ảnh bìa, **Mã tác giả**, **ID người đăng**, Số lượt xem.
* Mỗi tiểu thuyết có thể được viết bởi nhiều tác giả, một tác giả có thể viết nhiều tiểu thuyết. Tác giả có các thuộc tính: **Mã tác giả**, Tên tác giả, Mô tả.
* Một tiểu thuyết có nhiều thể loại, một thể loại được sử dụng cho nhiều tiểu thuyết. Thể loại có các thuộc tính: **Mã thể loại**, Tên thể loại, Mô tả.
* Một tiểu thuyết có nhiều chương. Chương tiểu thuyết có các thuộc tính: **Mã chương**, Tiêu đề chương, Nội dung, Ngày đăng, **Mã tiểu thuyết**.
* Người dùng có thể đăng nhập vào website. Người dùng có các thuộc tính: **ID người dùng**, Mật khẩu, Tên hiển thị, Email, **Mã quyền hạn**.
* Người dùng có thể đánh giá nhiều tiểu thuyết, mỗi lượt đánh giá chi thuộc về một người dùng. Đánh giá có các thuộc tính: **Mã đánh giá**, Điểm số, Ngày đánh giá, **ID người dùng, Mã tiểu thuyết.**
* Người dùng có thể bình luận nhiều tiểu thuyết, mỗi bình luận chỉ thuộc về một người dùng. Bình luận có các thuộc tính: **Mã bình luận**, nội dung, ngày bình luận, ngày chỉnh sửa, **ID người dùng, Mã tiểu thuyết**.
* Website có phân quyền cho người dùng. Phân quyền có các thuộc tính: **Mã phân quyên**, Tên phân quyền.

## Sơ đồ Thực thể quan hệ (ERD) cho website đọc truyện tiểu thuyết

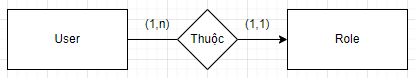


Hình 5.1 Sơ đồ thực thể quan hệ (ERD)

## Mô hình dữ liệu (ER)

### Xét Role – User

Mỗi User có một quyền, một quyền thuộc một hoặc nhiều User.



### Xét User – Rating

Mỗi User có thể đánh giá một hoặc nhiều tiểu thuyết, mỗi đánh giá chỉ thuộc về một User.



### Xét User – Comment

Mỗi User có thể bình luận một hoặc nhiều tiểu thuyết, mỗi bình luận chỉ thuộc về một User.



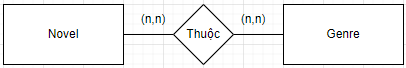
### Xét User – Novel

Mỗi User có thể đăng một hoặc nhiều tiểu thuyết, một tiểu thuyết chỉ được đăng bởi một User.



### Xét Novel – Genre

Mỗi tiểu thuyết có một hoặc nhiều thể loại, mỗi thể loại thuộc một hoặc nhiều tiểu thuyết.



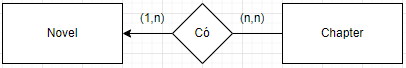
### Xét Novel – Author

Mỗi tiểu thuyết có một hoặc nhiều tác giả, mỗi tác giả thuộc một hoặc nhiều tiểu thuyết.



### Xét Novel – Chapter

Mỗi tiểu thuyết có một hoặc nhiều chương, mỗi chương chỉ thuộc một tiểu thuyết.



### Xét Novel – Rating

Mỗi tiểu thuyết có một hoặc nhiều lượt đánh giá, mỗi lượt đánh giá chỉ thuộc một tiểu thuyết.

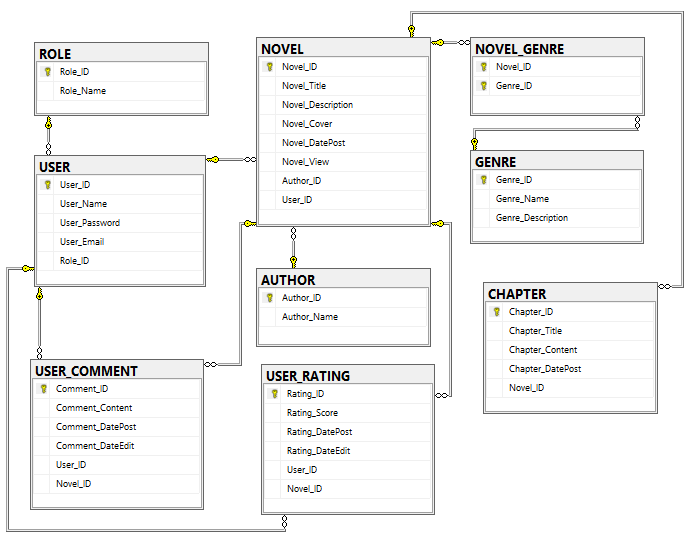


### Xét Novel – Comment

Mỗi tiểu thuyết có một hoặc nhiều lượt bình luận, mỗi lượt bình luận chỉ thuộc một tiểu thuyết.



## Sơ đồ của cơ sở dữ liệu cho website đọc truyện tiểu thuyết được hiển thị trong MS SQL Server



Hình 5.2 Sơ đồ của cơ sở dữ liệu cho website đọc truyện tiểu thuyết được hiển thị trong MS SQL Server

## Chi tiết các thực thể trong Cơ sở dữ liệu

### Thực thể tiểu thuyết (NOVEL)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Novel\_ID | PK, int, not null | Tự động tăng |
| Novel\_Title | nvarchar(100) , not null |  |
| Novel\_Description | nvarchar(500) , not null |  |
| Novel\_DatePost | date, not null |  |
| Novel\_View | int, not null |  |
| Author\_ID | FK, int, not null |  |
| User\_ID | FK, nvarchar(50), not null |  |

Bảng . Thực thể tiểu thuyết (NOVEL)

### Thực thể tác giả (AUTHOR)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Author\_ID | PK, int, not null | Tự động tăng |
| Author\_Name | nvarchar(50) , not null |  |

Bảng . Thực thể tác giả (AUTHOR)

### Thực thể thể loại (GENRE)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Genre\_ID | PK, int, not null | Tự động tăng |
| Genre\_Name | nvarchar(50) , not null |  |
| Genre\_Description | nvarchar(100) , not null |  |

Bảng . Thực thể thể loại (GENRE)

### Thực thể chương tiểu thuyết (CHAPTER)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Chapter\_ID | PK, int, not null | Tự động tăng |
| Chapter\_Title | nvarchar(50) , not null |  |
| Chapter\_Content | ntext , not null |  |
| Chapter\_DatePost | date , not null |  |
| Novel\_ID | FK, int , not null |  |

Bảng . Thực thể chương tiểu thuyết (CHAPTER)

### Thực thể quyền hạn (ROLE)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Role\_ID | PK, int, not null | Tự động tăng |
| Role\_Name | nvarchar(50) , not null |  |

Bảng . Thực thể quyền hạn (ROLE)

### Thực thể người dùng (USER)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| User\_ID | PK, nvarchar(50), not null |  |
| User\_Name | nvarchar(50) , not null |  |
| User\_Password | nvarchar(100), not null |  |
| User\_Email | nvarchar(50), not null |  |
| Role\_ID | FK, int , not null |  |

Bảng . Thực thể người dùng (USER)

### Thực thể đánh giá của người dùng (USER\_RATING)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Rating\_ID | PK, int, not null | Tự động tăng |
| Rating\_Score | int , not null | Từ 1 đến 5 |
| Rating\_DatePost | date, not null |  |
| Rating\_DateEdit | date, not null |  |
| User\_ID | FK, nvarchar(50), not null |  |
| Novel\_ID | FK, int, not null |  |

Bảng . Thực thể đánh giá của người dùng (USER\_RATING)

### Thực thể bình luận của người dùng (USER\_COMMENT)

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Comment\_ID | PK, int, not null | Tự động tăng |
| Comment\_Content | ntext , not null |  |
| Comment\_DatePost | date, not null |  |
| Comment\_DateEdit | date, not null |  |
| User\_ID | FK, nvarchar(50), not null |  |
| Novel\_ID | FK, int, not null |  |

Bảng . Thực thể bình luận của người dùng (USER\_COMMENT)

### Thực thể NOVEL\_GENRE

**Novel\_ID + Genre\_ID = Khóa chính**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Novel\_ID | PK, int, not null |  |
| Genre\_ID | PK, int , not null |  |

Bảng . Thực thể NOVEL\_GENRE

### Thực thể NOVEL\_AUTHOR

**Novel\_ID + Author\_ID = Khóa chính**

|  |  |  |
| --- | --- | --- |
| **Tên cột** | **Loại dữ liệu** | **Ghi chú** |
| Novel\_ID | PK, int, not null |  |
| Author\_ID | PK, int , not null |  |

Bảng . Thực thể NOVEL\_AUTHOR

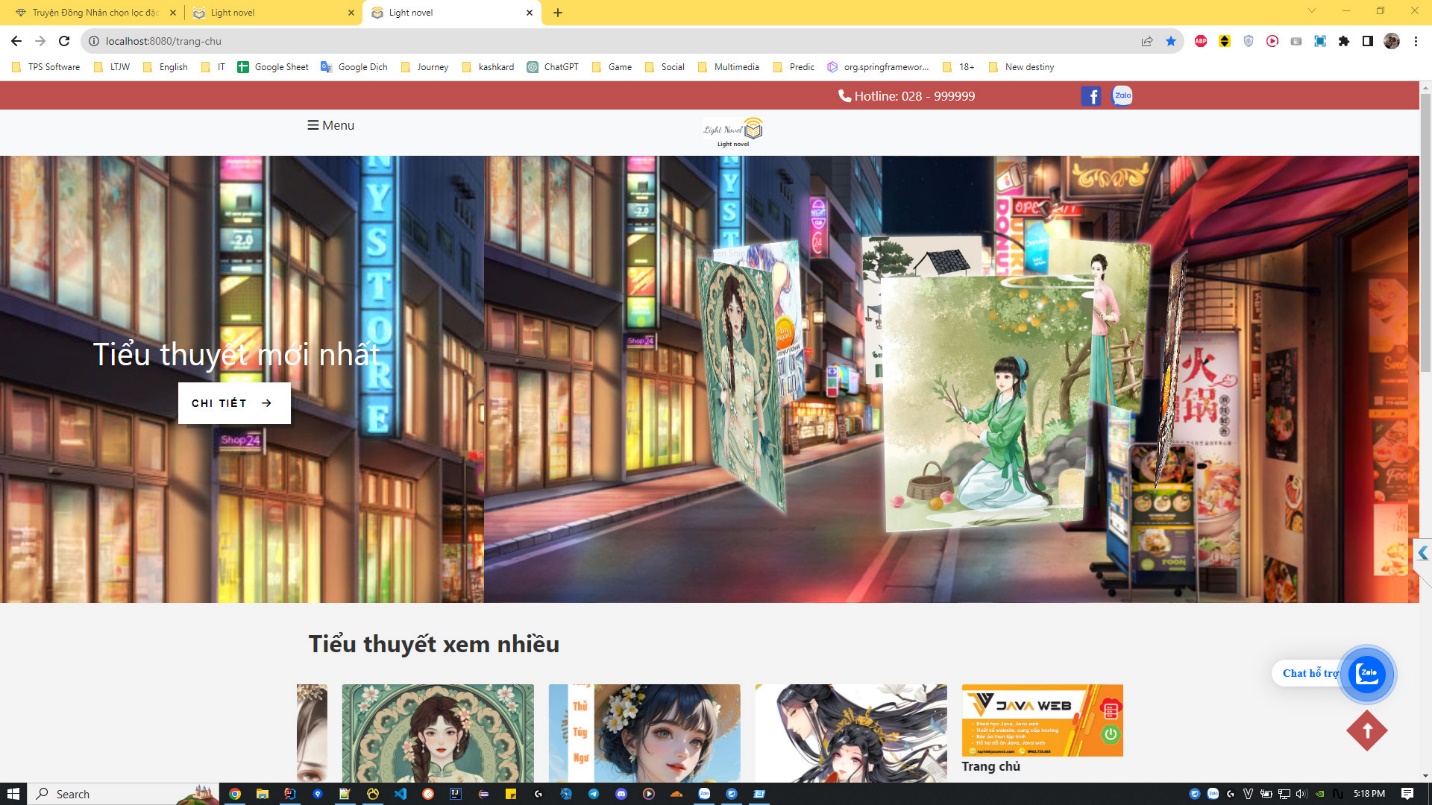
# Website đọc truyển tiểu thuyết

## Yêu cầu hệ thống:

* Hệ quản trị cơ sở dữ liệu: Postgre SQL.
* Công cụ lập trình: IntelliJ IDEA, Java SDK 17.
* Trình duyệt web: Google Chrome, Microsoft Edge, Mozilla Firefox,…

## Giao diện chương trình

### Trang chủ: Hiển thị danh sách các tiểu thuyết



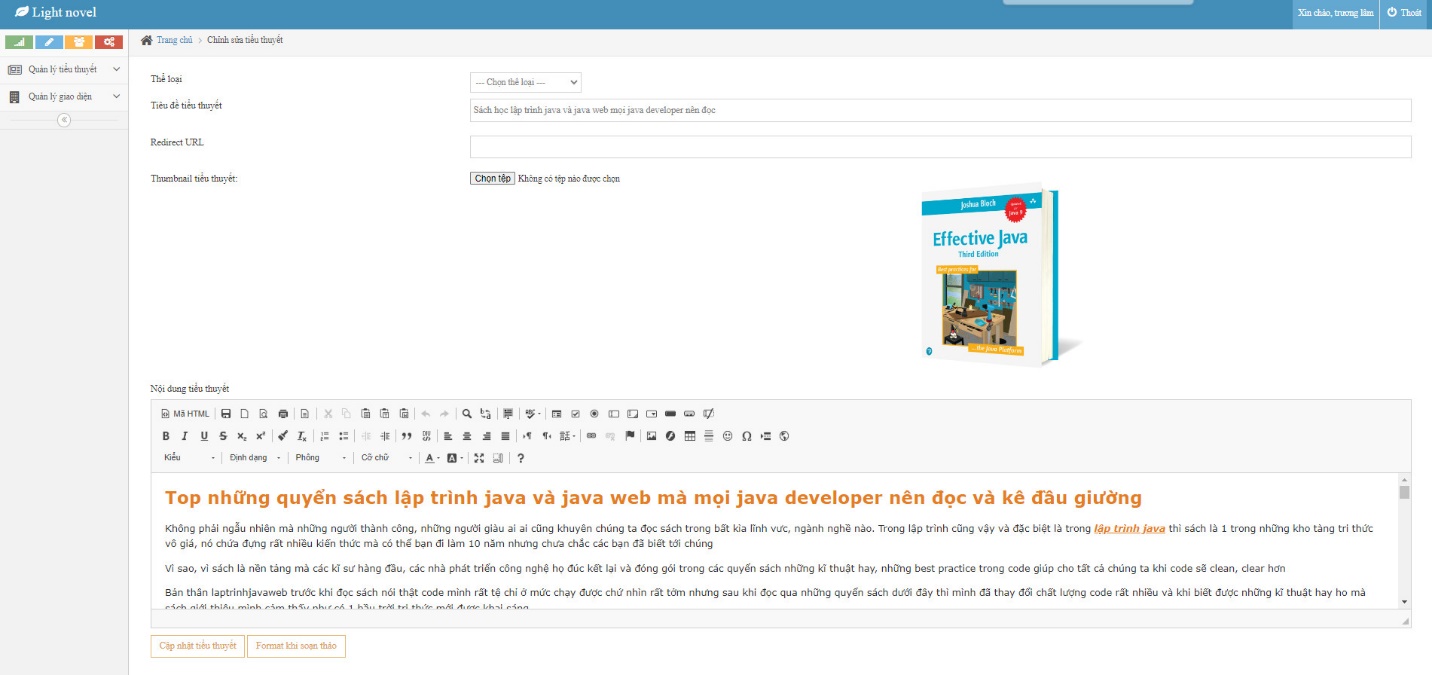
Hình . Trang chủ

### Trang đăng nhập:



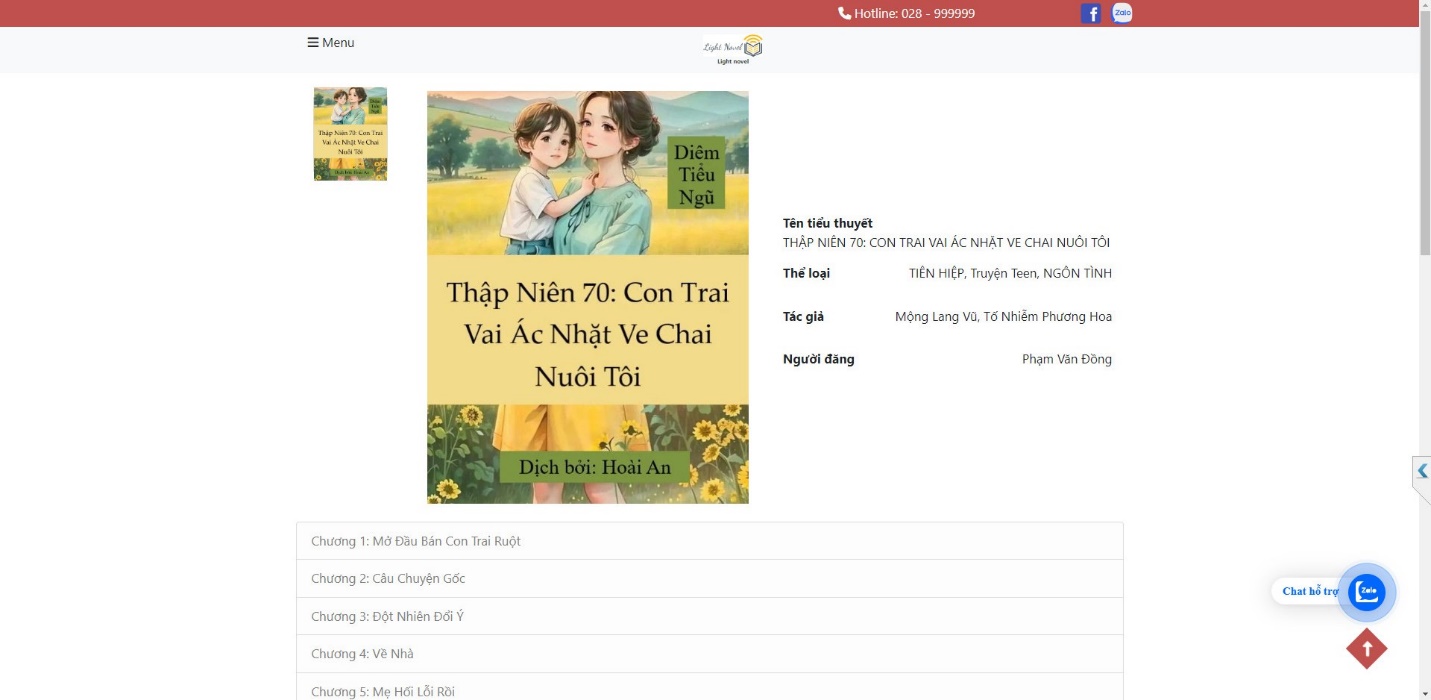
Hình . Trang đăng nhập

### Trang đăng tiểu thuyết của người đọc:



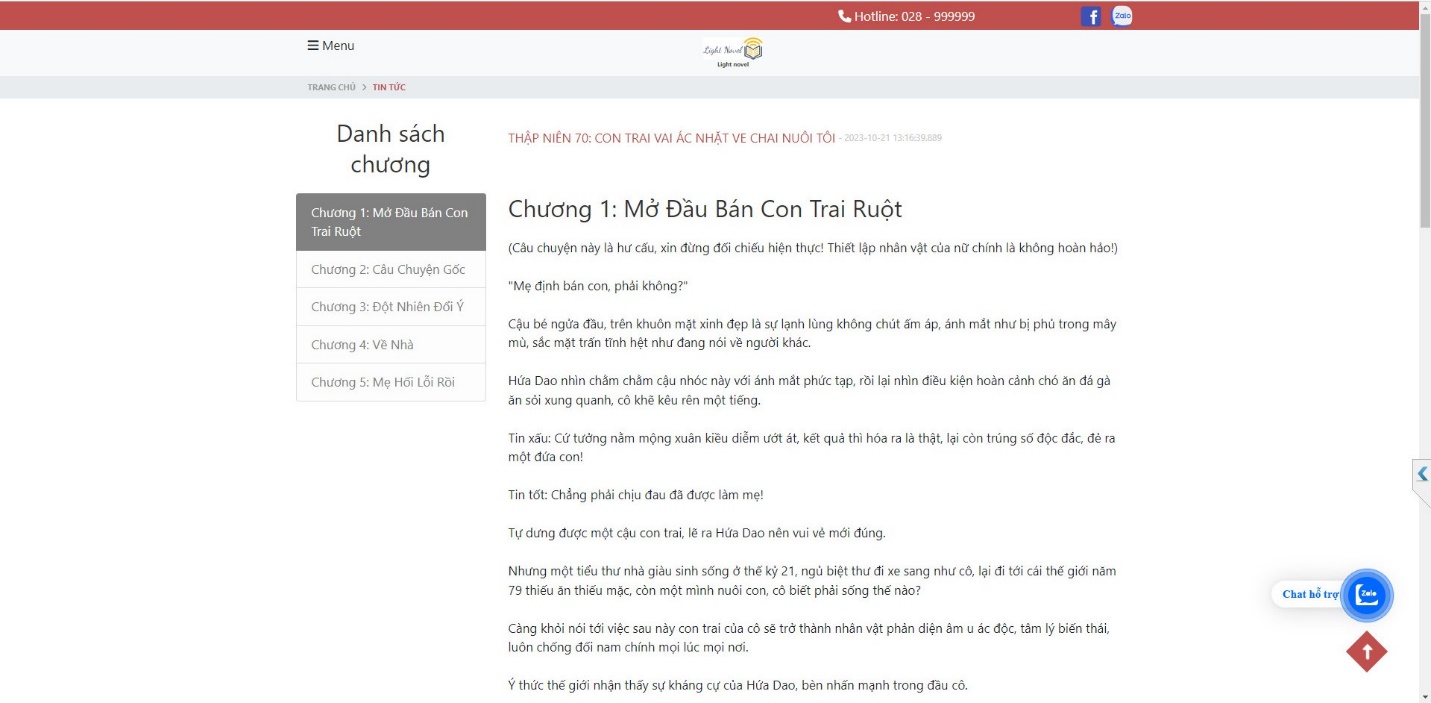
Hình . Trang đăng tiểu thuyết

### Trang chi tiêt tiểu thuyết



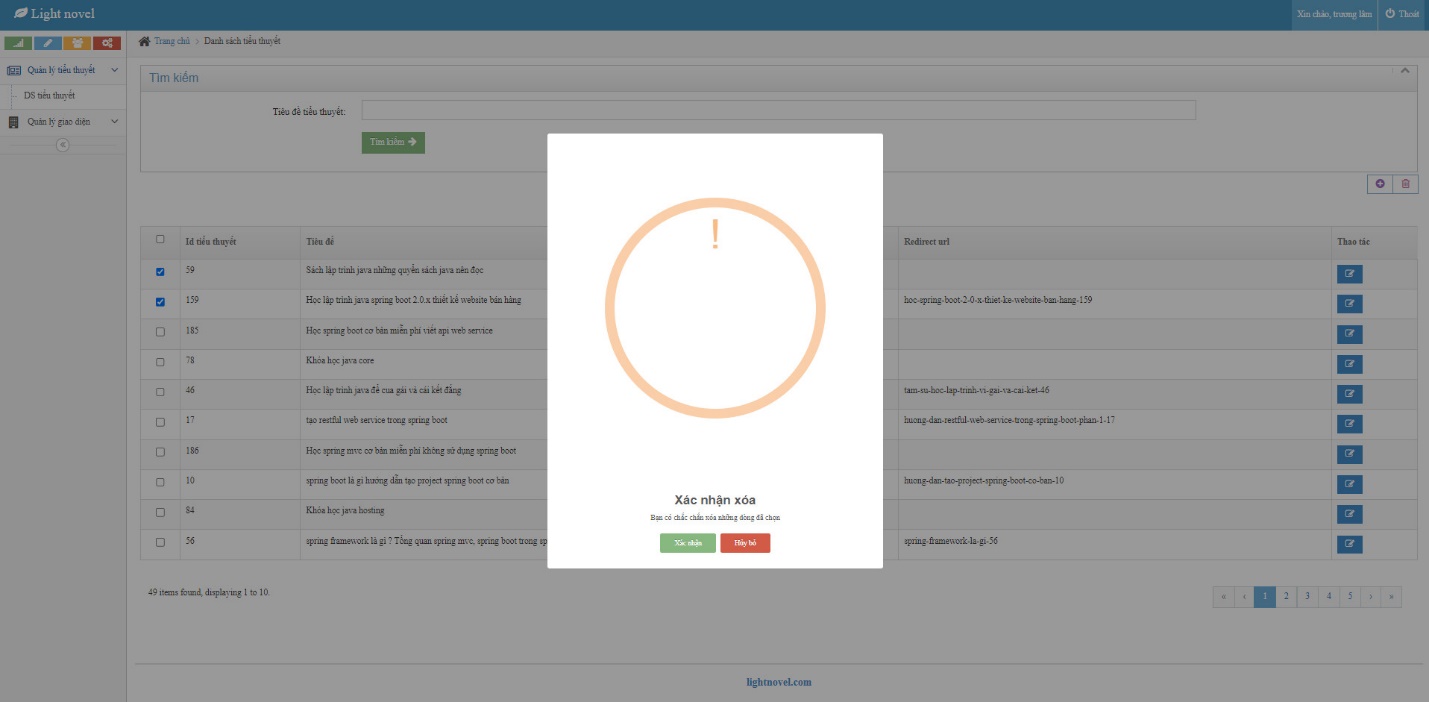
Hình . Trang chi tiết tiểu thuyết

### Trang nội dung chương



Hình . Trang nội dung chương

### Hộp thoại thông báo xác nhận:



Hình . Hộp thoại thông báo xác nhận

# KẾT LUẬN

## Đánh giá kết quả đạt được

Sau một thời gian tập trung triển khai đề tài, em vẫn chưa thể hoàn thiện được đầy đủ các chức năng đã đăng ký cho đề tài “website đọc truyện tiểu thuyết”.

## Những kết quả đạt được

Về công nghệ:

* Tìm hiểu tổng quan về công nghệ lập trình web: Spring framework, MySQL.
* Tìm hiểu và nắm bắt sơ lược cách sử dụng các công cụ: IntelliJ IDEA, Xampp
* Biết được cách thiết kế Web cũng như cách tổ chức cơ sở dữ liệu.

Tính năng của chương trình:

* Hoàn thành các chức năng quản trị của người quản trị.
* Cho phép người đọc xem tiểu thuyết và tìm kiếm tiểu thuyết.
* Cho phép người dùng đăng nhập, đăng ký tài khoản ứng dụng và phân quyền người dùng thông qua Claim.
* Cho phép người dùng được đăng tiểu thuyết.
* Cho phép người dùng được đánh giá tieur thuyết
* Cho phép người dùng được tham gia thảo luận về tiểu thuyết

## Những hạn chế

Bên cạnh những kết quả đạt được thì đồ án vẫn còn những hạn chế như sau:

* Giao diện đơn giản, chưa được chăm chút kỹ lưỡng.
* Website mang tính chất mô phỏng, chưa được tối ưu các chức năng và thiếu một số chức năng phụ của người dùng, bình luận của người dùng.
* Một số lỗi nhỏ phát sinh khi vận hành ứng dụng.

## Hướng phát triển

Tìm hiểu sâu hơn về các nền tảng công nghệ như ASP.NET nói chung và ASP.NET Core MVC nói riêng, Entity Framework,… kết hợp với hệ quản trị CSDL MS SQL Server để có thể đáp ứng nhiều hơn nữa nhu cầu của người sử dụng, phát triển và tối ưu hóa hệ thống để đem lại hiệu quả cao hơn, hoàn thiện chương trình để đưa website vào ứng dụng thực tế.

## Kết luận

Qua quá trình xây dựng dự án “Website đọc truyện tiểu thuyết” em đã hiểu rõ hơn về các quy trình nghiệp vụ cũng như quá trình phát triển dự án. Do đó, em đã biết cách thực hiện các dự án sau này với công nghệ framework ASP.NET Core MVC và hệ quản trị cơ sở dữ liệu MS SQL Server.

Em hy vọng bản thân có thể phát triển hơn nữa để phục vụ cho mục đích việc xây dựng các dự án thực tiễn.

Cuối cùng, em xin chúc các thầy cô sức khỏe để tiếp tục truyền thụ kiến thức cho thế hệ mai sau. Để cho đất nước Việt Nam được vươn lên sánh tầm với các cường quốc về mặt công nghệ.

# 

# DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt:

* Trần Đình Quế, Nguyễn Mạnh Sơn, PHÂN TÍCH & THIẾT KẾ HỆ THỐNG THÔNG TIN, Hà Nội, 2007.

Website tham khảo:

* Wikipedia, MS SQL Server: <https://en.wikipedia.org/wiki/Microsoft_SQL_Server>
* Wikipedia, API: <https://en.wikipedia.org/wiki/API>
* Bootstrap: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
* W3school, Spring tutorial: https://www.w3schools.blog/spring-tutorial