|  |
| --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO  TRƯỜNG ĐẠI HỌC NÔNG LÂM TP.HỒ CHÍ MINH  KHOA CÔNG NGHỆ THÔNG TIN |

|  |
| --- |
| **Web Bán Sách** |

|  |  |
| --- | --- |
| Giảng viên hướng dẫn: | Nguyễn Đức Công Song |
| Sinh viên thực hiện: | Hoàng Lê Nguyên Mạnh - 22130163 |

|  |
| --- |
| Tp. Hồ Chí Minh, tháng 03 năm 2025 |

DANH MỤC HÌNH ẢNH

[Hình 2.1: Use Case Tổng 6](#_Toc199100252)

[Hình 3.1: Kiến trúc hệ thống 6](#_Toc199100253)

[Hình 4.1: Use case book Management 7](#_Toc199100254)

[Hình 6.1: Sequence Book Management (UC13.1) 14](#_Toc199100255)

[Hình 7.1: Ui Add Book Form 15](#_Toc199100256)

[Hình 7.2: GUI button (thêm sản phẩm) 15](#_Toc199100257)

[Hình 8.1: Sơ đồ ERD 16](#_Toc199100258)

**MỤC LỤC**

[Chương 1 Giới Thiệu Tổng Quan. 5](#_Toc199093776)

[1.1 Lý do chọn đề tài 5](#_Toc199093777)

[1.2 Mục tiêu 5](#_Toc199093778)

[1.3 Phạm vi thực hiện 5](#_Toc199093779)

[1.4 Các chức năng chính: 5](#_Toc199093780)

[1.5 Công nghệ và công cụ sử dụng. 6](#_Toc199093781)

[Chương 2 Lược Đồ Use Case. 6](#_Toc199093782)

[Chương 3 Kiến Trúc Hệ Thống. 7](#_Toc199093783)

[Chương 4 Đặc Tả Use Case 7](#_Toc199093784)

[4.1 Use Case Name 7](#_Toc199093785)

[4.1.1 Introduction 7](#_Toc199093786)

[4.1.2 Brief Description 7](#_Toc199093787)

[4.2 Scope 8](#_Toc199093788)

[4.3 Actor 8](#_Toc199093789)

[4.4 Pre – Conditions 8](#_Toc199093790)

[4.5 Post – Conditions 8](#_Toc199093791)

[4.6 Flows of Events 9](#_Toc199093792)

[4.6.1 Basic Flow 9](#_Toc199093793)

[4.6.2 Alternative Flow 11](#_Toc199093794)

[4.7 Special Requirements 12](#_Toc199093795)

[Chương 5 Test Case Cho Use Case Đã Chọn. 12](#_Toc199093796)

[Chương 6 Lược Đồ Sequence. 14](#_Toc199093797)

# Giới Thiệu Tổng Quan.

## Lý do chọn đề tài

Với sự phát triển mạnh mẽ của thương mại điện tử, nhu cầu mua sách trực tuyến ngày càng tăng. Một website bán sách giúp người dùng dễ dàng tìm kiếm, lựa chọn và mua sách nhanh chóng, tiện lợi.

## Mục tiêu

Website bán sách được xây dựng nhằm cung cấp nền tảng thương mại điện tử giúp người dùng dễ dàng tìm kiếm, mua sách trực tuyến và quản lý đơn hàng. Đồng thời, website hỗ trợ quản trị viên trong việc quản lý sản phẩm, đơn hàng và tài khoản người dùng một cách hiệu quả.

## Phạm vi thực hiện

Phạm vi của website bán sách online tập trung vào các cửa hàng có quy mô nhỏ đến vừa, số lượng sách vừa phải, thao tác quản lý đơn giản.

Hệ thống được xây dựng trên nền tảng web, phục vụ cho nhu cầu mua sách online và quản lý bán hàng cơ bản.

Phù hợp triển khai trong môi trường học tập và có thể mở rộng trong tương lai.

## Các chức năng chính:

**Chung:**

Đăng ký, đăng nhập, đổi mật khẩu, cập nhật thông tin cá nhân.

**Khách hàng:**

* Xem danh sách sách theo danh mục, tìm kiếm sách.
* Xem chi tiết sách (mô tả, giá, đánh giá, tác giả, nhà xuất bản, số lượng tồn kho).
* Thêm sách vào giỏ hàng, đặt hàng và thanh toán.
* Xem lịch sử đơn hàng, hủy đơn hàng, mua lại đơn hàng.

**Admin:**

* Quản lý danh mục sách, thêm/sửa/xóa sách.
* Quản lý đơn hàng, thay đổi trạng thái đơn hàng.
* Quản lý người dùng, phân quyền truy cập.

## Công nghệ và công cụ sử dụng.

**Frontend**:

* JSP, HTML5, CSS3, JavaScript (jQuery / AJAX)
* Bootstrap (hoặc Tailwind) để responsive và giao diện nhanh chóng

**Backend**:

* Java 21, Spring Boot 3 (Spring MVC, Spring Data JPA)
* Thực thi dưới Tomcat embedded (hoặc Tomcat 10.1 deploy WAR)

**Database**: MySQL (Aiven.io)

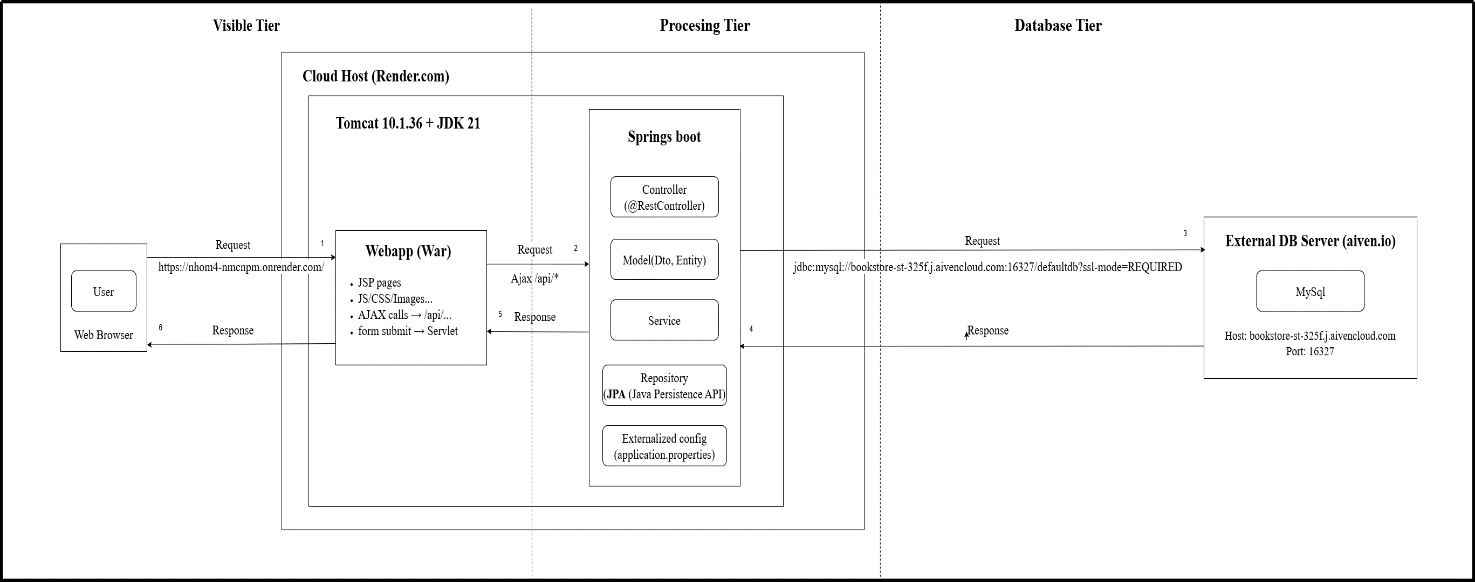
**DevOps & Triển khai**:

* Docker (multi-stage build WAR)
* Render.com (hoặc Railway) để deploy
* GitHub + GitHub Actions cho CI/CD

# Lược Đồ Use Case.

Use Case Tổng

# Kiến Trúc Hệ Thống.



Kiến trúc hệ thống

# Đặc Tả Use Case

## Use Case Name

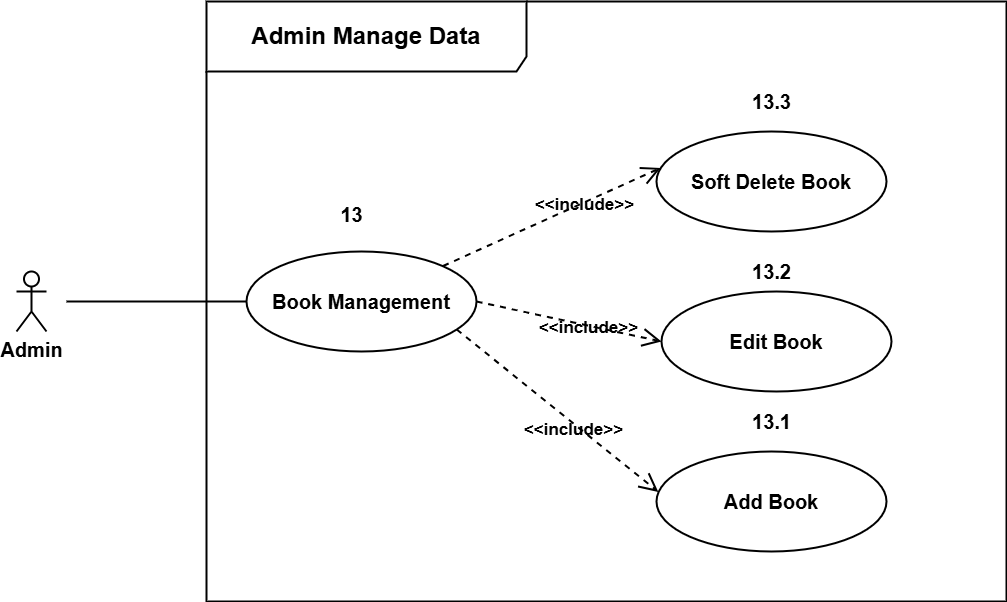
UC13.1 – Add Book

### Introduction

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Name** | **Use Case ID** | **Date Created** | **Created By** |
| Thêm sản phẩm | UC-13.1 | 17/05/2025 | Hoàng Lê Nguyên Mạnh |
| Primary Actors | Secondary Actors | | |
| User | N/A | | |

### Brief Description

Admin có thể thêm, sửa, xóa và xem danh sách sách đang có trong hệ thống. Đây là chức năng quản lý nội dung sản phẩm chính trên website bán sách trực tuyến.



Use case book Management

## Scope

Hệ thống bán sách trực tuyến (Online Bookstore System)

## Actor

Admin

## Pre – Conditions

User đăng nhập có quyền của quản trị viên hoặc có quyền phù hợp để có thể truy cập

## Post – Conditions

**Đảm bảo thành công:** Danh sách sách phản ánh các thay đổi (thêm, sửa, xóa).

**Bảo đảm tối thiểu**: Không có dữ liệu sách bị lỗi. Nếu thao tác thất bại, hệ thống vẫn duy trì trạng thái nhất quán.

## Flows of Events

### Basic Flow

#### 13.1.1.0 **Admin** chọn button thêm sách mới.

• HTML: button với onclick="showAddProductForm()"

• JS: showAddProductForm() ⇒ overlay.style.display = "flex"

#### 13.1.1.1 **adminPage** hiển thị form overlay-addProduct hiển thị.

• <section class="overlay-addProduct"> chứa <form id="add-product-form">

• CSS display: flex để overlay phủ màn hình

#### 13.1.1.2 **Admin** nhập thông tin và nhấn Lưu .

• Form có onsubmit="createBook(event)"

• JS: createBook(event) gọi event.preventDefault()

#### 13.1.1.3 **adminPage**: JS thu thập dữ liệu từ form thành JSON.

• getFormData() đọc FormData từ form

• build object { title, author, description, publishedDate, price, stockQty, imageUrl, categoryIds }

#### 13.1.1.4 **adminPage**: JS gửi HTTP POST 🡺 **BookController.**

• fetch("/WebBanSach/api/books", { method: "POST", headers: { "Content-Type": "application/json" }, body: JSON.stringify(data) })

#### 13.1.1.5 **BookController** nhận dữ liệu JSON từ **adminPage**, ánh xạ vào DTO, và kiểm tra ràng buộc.

• @PostMapping -> addBook(@Valid @RequestBody BookDto dto)

• Spring tự validate BookDto theo annotation

#### 13.1.1.6 **BookController** gọi **BookService** xử lý nghiệp vụ

• bookService.createBook(dto) trong @Transactional

#### 13.1.1.7 **BookService** kiểm tra sách đã tồn tại chưa.

• bookRepo.existsByTitleAndAuthor(dto.getTitle(), dto.getAuthor())

#### 13.1.1.8 Database **bookstore** trả về kết quả sách chưa tồn tại.

#### 13.1.1.9 **BookService** khởi tạo sách, lưu danh mục và lưu sách.

• new Books(), set fields từ dto, set default image nếu null

• categoryRepo.findAllById(dto.getCategoryIds()) và book.getCategories().addAll(cats)

• Books saved = bookRepo.save(book) ➔ persist books và relation

#### 13.1.1.10 Database **bookstore** trả về sách mới và ID.

#### 13.1.1.11 **BookService** lưu sách vào transactions, và trả về đôi tượng đã lưu bao gồm Id mới sinh

• @transacstional

• public Books createBook(BookDto dto){}

#### 13.1.1.12 **BookController** trả về HTTP 201 cùng JSON { "id": savedId } khi thêm sách thành công

• ResponseEntity.status(CREATED).body(Map.of("id", saved.getId()))

• JS result = await response.json(); alert(`✅ ID = ${result.id}`)\

#### 13.1.1.13 **adminPage** hiển thị thông báo thêm sách thành công, sau đó reset form và ẩn form

• const result = await response.json();

• alert(`✅ Thêm sách thành công! ID = ${result.id}`);

• form.reset();

• hideAddProductForm()

### ****Alternative Flow****

#### Nếu Không nhập đủ dữ liệu.

#### 13.1.2.3 Hiển thị thông báo "Vui lòng nhập trường này"

• <input .... required>

#### Nếu Sách đã tồn tại.

#### 13.1.2.11 **BookService** trả về lỗi sách đã tồn tại

• throw new DuplicateBookException("BOOK\_DUPLICATE", "Sách với tiêu đề và tác giả này đã tồn tại")

#### 13.1.2.12 **BookController** trả về HTTP 400 cùng JSON type: "BOOK\_DUPLICATE", message: "Sách với tiêu đề và tác giả đã tồn tại"

• return 400 với body { type: code, message }

#### 13.1.2.13 **adminPage** hiển thị thông báo lỗi “BOOK\_DUPLICATE” từ server.

• async function parseErrorResponse(response) {}

#### Nếu Dữ liệu không hợp lệ.

#### 13.1.2.6 **BookController** trả về HTTP 400 cùng JSON type: "VALIDATION\_ERROR", message: "..."

• return 400 với body { type: "VALIDATION\_ERROR", message, validationErrors}

#### 13.1.2.7 **adminPage** hiển thị thông báo lỗi “VALIDATION\_ERROR” từ server.

• async function parseErrorResponse(response) {}

#### Lưu sách thất bại.

#### 13.1.3.1 Nếu xảy ra lỗi bất ngờ thì **BookController** trả về HTTP 500 cùng JSON { type: "INTERNAL\_ERROR", message: "Có lỗi trong server" }

• Return 500 với body { type: "INTERNAL\_ERROR", "message", ex.getMessage() != null ? ex.getMessage() : "Có lỗi trong server"}

#### 13.1.3.2 Hiển thị thông báo lỗi chung "INTERNAL\_ERROR" từ server.

• async function parseErrorResponse(response) {}

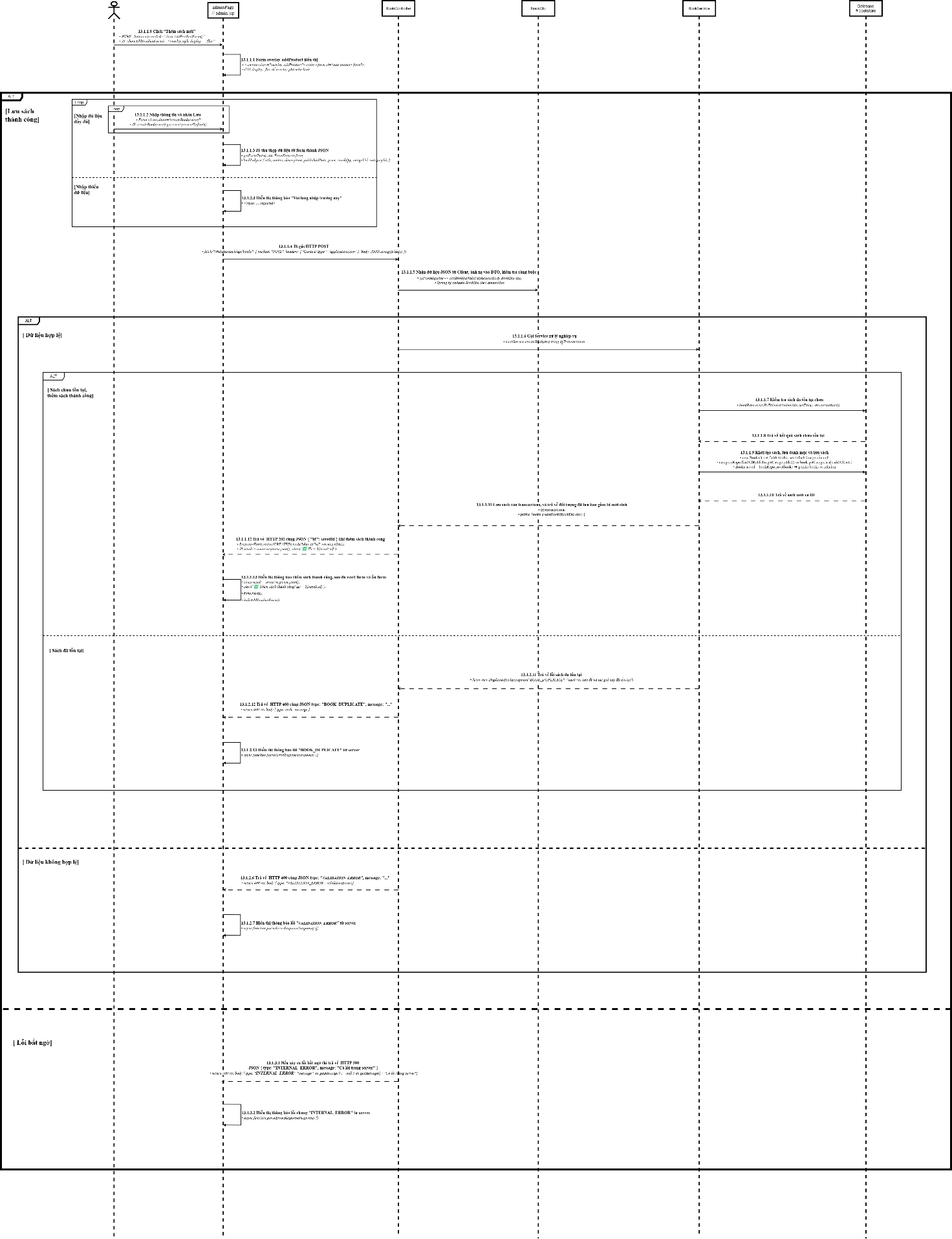
## Special Requirements

Bảo mật: Chỉ có tài khoản có thẩm quyền mới được truy cập vào trang quản lý.

# Test Case Cho Use Case Đã Chọn.

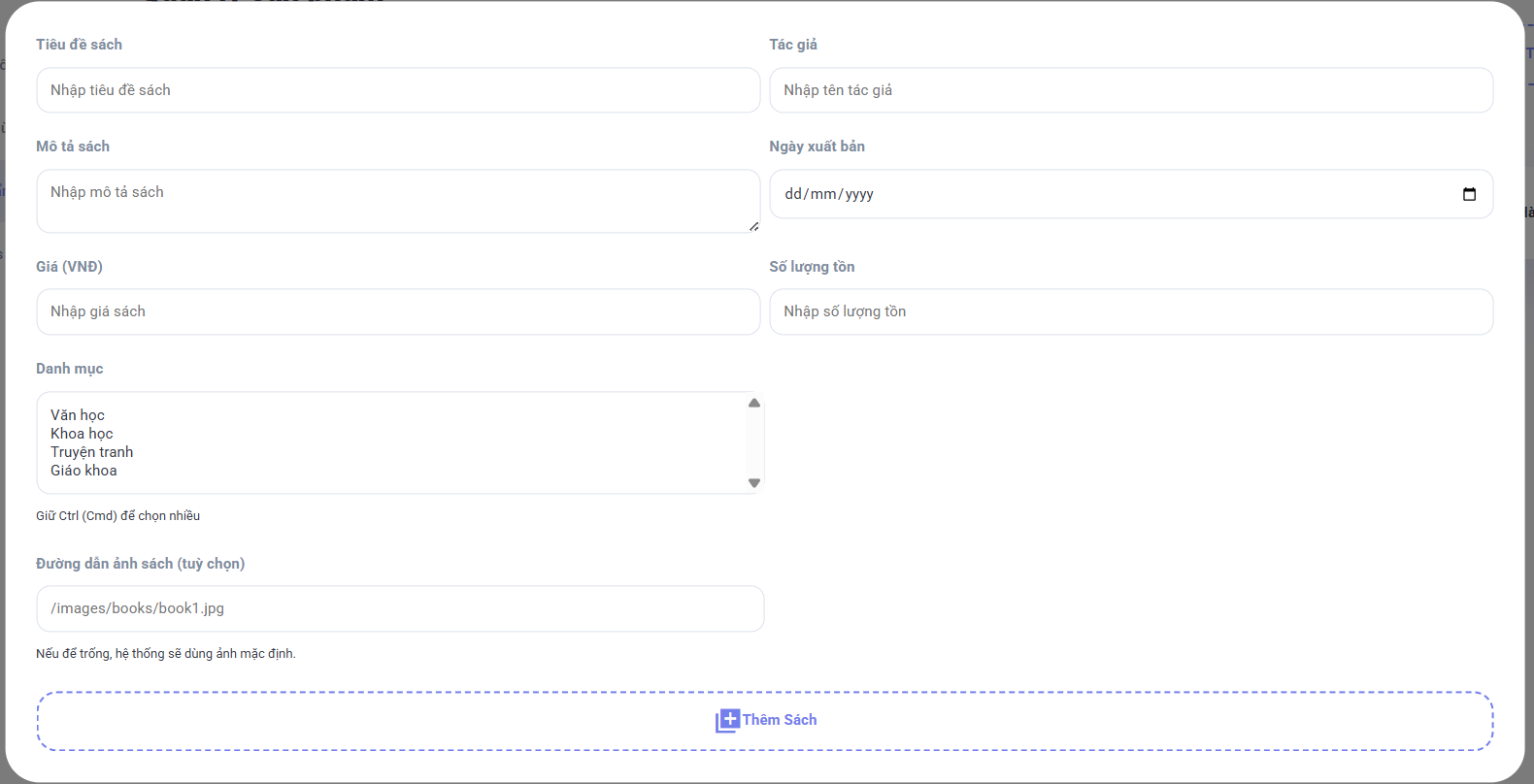
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên | Mô tả | Đầu vào | Kết quả |
| 1 | Hiển thị form thêm sách | Kiểm tra admin click nút “Thêm sách mới” thì hiển thị form | Click nút có onclick="showAddProductForm()" | Form thêm sách hiển thị dạng overlay (display: flex) |
| 2 | Submit khi không điền dữ liệu | Không nhập dữ liệu gì và nhấn Lưu | Bỏ trống các input, nhấn nút Lưu | Trình duyệt hiện cảnh báo “Vui lòng nhập trường này” ở các trường có required |
| 3 | Submit dữ liệu hợp lệ | Thêm sách với dữ liệu hợp lệ và nhấn Lưu | Dữ liệu: title, author, description, publishedDate, price, stockQty, categoryIds | Sách được lưu, trả về JSON có id, hiển thị alert thành công, reset form, ẩn form |
| 4 | Sách bị trùng tiêu đề và tác giả | Thêm sách có title + author trùng với sách đã tồn tại | Dữ liệu giống với sách đã lưu | Server trả HTTP 400, JSON {type: "BOOK\_DUPLICATE"}, hiển thị alert lỗi "BOOK\_DUPLICATE" |
| 5 | Dữ liệu không hợp lệ | Nhập sai kiểu dữ liệu (VD: giá là chữ, ngày không đúng format) | Ví dụ: "price": "abc", "publishedDate": "ngày mai" | Server trả HTTP 500, JSON {type: "INTERNAL\_ERROR"}, alert "Có lỗi trong server" |
| 6 | Gọi API thất bại (500 server error) | Giả lập lỗi trong server khi lưu sách | Dữ liệu hợp lệ nhưng BookService ném exception | Server trả HTTP 500, JSON {type: "INTERNAL\_ERROR"}, alert "Có lỗi trong server" |
| 7 | Không có danh mục | Gửi sách với categoryIds rỗng | categoryIds = [] | Server có thể chặn (tùy validate), nếu không → lưu sách không có danh mục |
| 8 | Ảnh sách để trống | Không điền imageUrl | Các trường khác hợp lệ, imageUrl = "" | Server gán ảnh mặc định, lưu sách thành công, alert hiển thị ID |
| 9 | Form bị đóng sau khi thêm | Kiểm tra sau khi thêm sách thì form bị ẩn | Sau khi Lưu thành công | overlay.style.display = "none", form reset, không còn trên màn hình |
| 10 | Kiểm tra gửi đúng JSON định dạng | JS gửi request đúng cấu trúc JSON | Xem nội dung request bằng Dev Tools → tab Network → JSON body | JSON gửi đúng định dạng các trường { title, author, ... }, content-type: application/json |

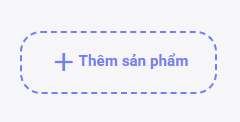
# Lược Đồ Sequence.



Sequence Book Management (UC13.1)

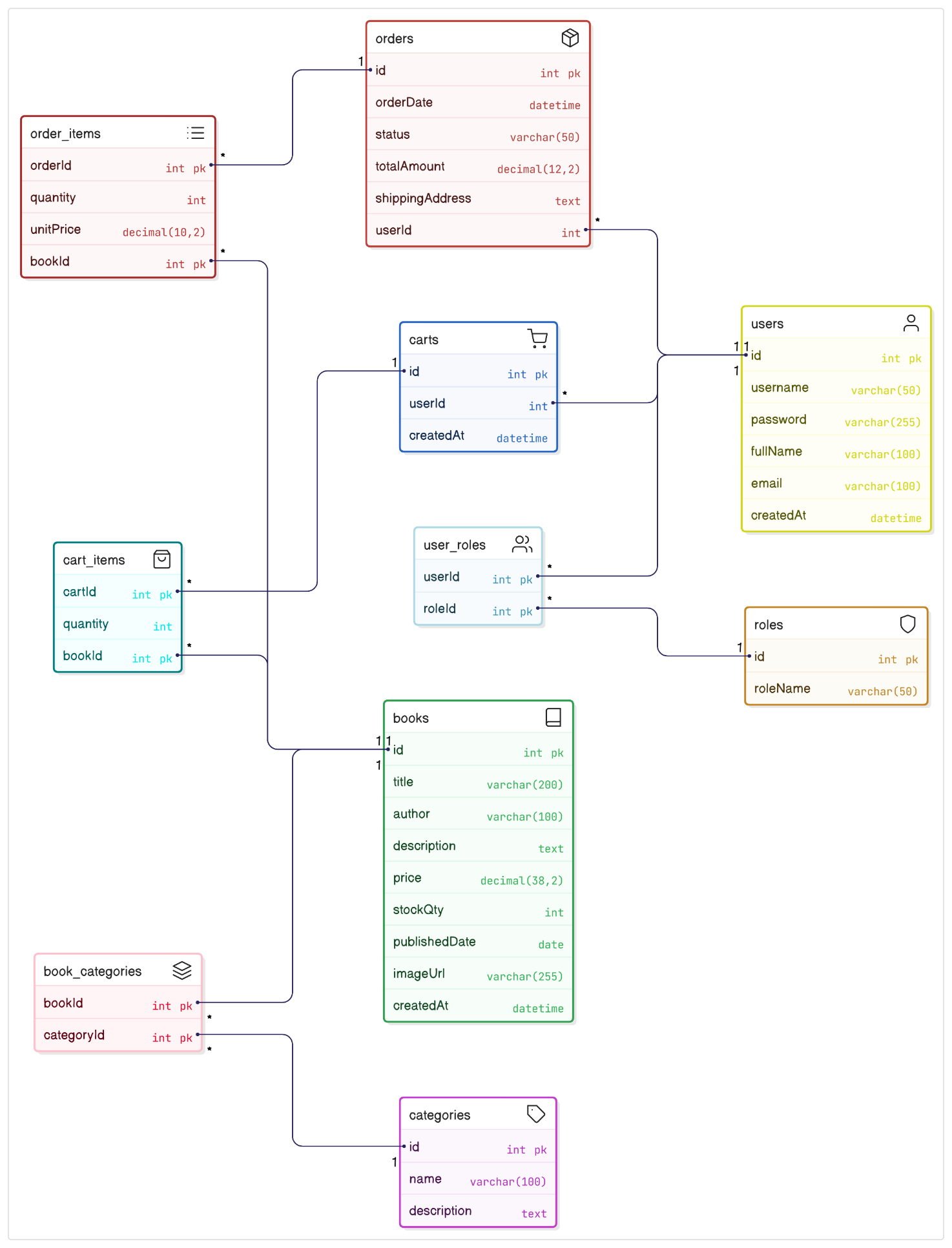
# Các Giao Diện (GUI/UI)



Ui Add Book Form

GUI button (thêm sản phẩm)

# Sơ đồ ERD



Sơ đồ ERD