**BỘ KHOA HỌC VÀ CÔNG NGHỆ**

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

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



**BÁO CÁO   
ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI SỐ: 23**

**TRANG BÁN SÁCH TẠI CỬA HÀNG**

**Môn học:** Đảm bảo chất lượng phần mềm

**Giảng viên hướng dẫn:** Thầy Nguyễn Anh Hào

**Thực hiện bởi nhóm sinh viên, bao gồm:**

1. Nguyễn Ngọc Thiên Phúc N21DCCN066 Trưởng nhóm

2. Trần Thị Thùy Ngân N21DCCN055 Thành viên

**TP.HCM, tháng 06/2025**

# MỤC LỤC

[MỤC LỤC 2](#_Toc200737644)

[DANH SÁCH HÌNH, BẢNG 3](#_Toc200737645)

[TÓM TẮT 4](#_Toc200737646)

[CHƯƠNG I. TỔNG QUAN 5](#_Toc200737647)

[1. Giới thiệu đề tài 5](#_Toc200737648)

[2. Vai trò và nhiệm vụ của mỗi thành viên: 5](#_Toc200737649)

[3. Cơ sở lý thuyết 5](#_Toc200737650)

[CHƯƠNG II. PHÂN TÍCH & THIẾT KẾ HỆ THỐNG 6](#_Toc200737651)

[1. Yêu cầu từ stake-holders (người sử dụng): 6](#_Toc200737652)

[2. Phân tích hệ thống 6](#_Toc200737653)

[3. Thiết kế hệ thống 6](#_Toc200737654)

[CHƯƠNG III. TRIỂN KHAI HỆ THỐNG 7](#_Toc200737655)

[1. Phần mềm ứng dụng 7](#_Toc200737656)

[2. Kiểm thử 8](#_Toc200737657)

[CHƯƠNG IV. KẾT LUẬN 12](#_Toc200737658)

[1. Nhận xét về điểm mạnh, điểm yếu trong cách tiến hành đồ án 12](#_Toc200737659)

[2. Khả năng cải tiến cách tiến hành đồ án 12](#_Toc200737660)

[TÀI LIỆU THAM KHẢO 13](#_Toc200737661)

# DANH SÁCH HÌNH, BẢNG

# TÓM TẮT

**Mô tả đề tài**

Các yêu cầu chính của đề tài:

1. ***Quản lý sách:*** *(1 usecase)*

* Nhân viên có thể thêm sách mới vào hệ thống với các thông tin như ISBN, tiêu đề, tác giả, danh mục, nhà xuất bản, năm xuất bản, giá, ảnh, số lượng.
* Nhân viên có thể sửa thông tin sách khi cần cập nhật nội dung hoặc chỉnh sửa lỗi.
* Hệ thống cho phép xóa mềm sách – soft delete (đánh dấu isDeleted = true) để tránh mất dữ liệu.
* Nhân viên có thể tìm kiếm sách theo từ khóa hoặc lọc theo danh mục, tác giả, nhà xuất bản, ....
* Hệ thống giúp cửa hàng quản lý tốt hơn về việc tồn kho (lọc số lượng sách từ thấp đến cao) để cân nhắc nên nhập sách nào cho đợt tiếp theo.

# CHƯƠNG I. TỔNG QUAN

## 1. Giới thiệu đề tài

......

## 2. Vai trò và nhiệm vụ của mỗi thành viên:

<Họ tên thành viên 01> <MSSV>: .....

<Họ tên thành viên 02> <MSSV>: ....

<Họ tên thành viên 03> <MSSV>: ....

.....

## 3. Cơ sở lý thuyết

# CHƯƠNG II. PHÂN TÍCH & THIẾT KẾ HỆ THỐNG

## 1. Yêu cầu từ stake-holders (người sử dụng):

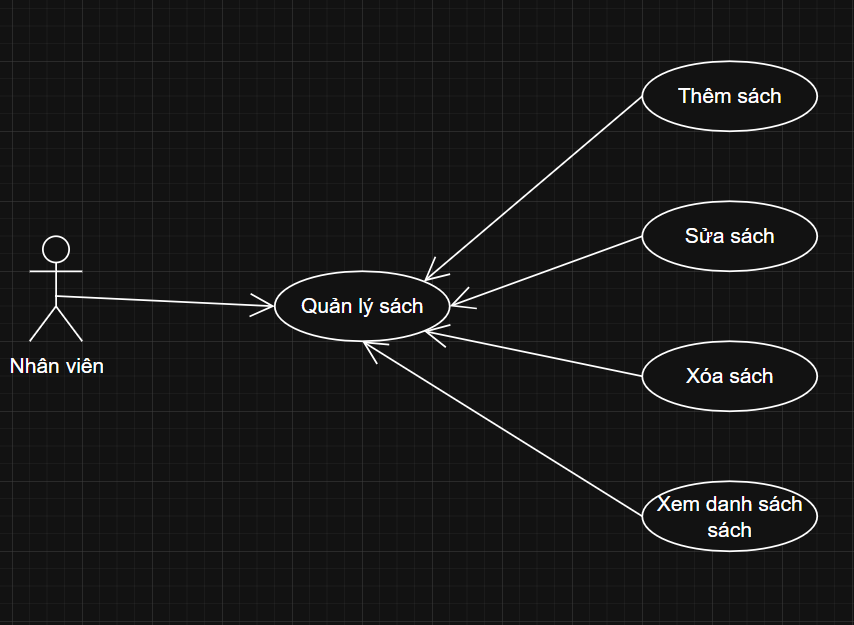
**Usecase “Quản lý sách”:**

* Người dùng chính: Nhân viên cửa hàng sách
* Mong muốn: Có thể dễ dàng thêm mới sách, chỉnh sửa hoặc xóa khi cần, tìm kiếm sách nhanh chóng phục vụ việc tư vấn và thanh toán.

## 2. Phân tích hệ thống

**Usecase “Quản lý sách”:**

Usecase tổng thể "Quản lý sách" mô tả các chức năng nghiệp vụ chính mà hệ thống hỗ trợ cho nhân viên cửa hàng. Đây là một trong những nghiệp vụ cốt lõi, ảnh hưởng trực tiếp đến khả năng vận hành và phục vụ khách hàng.



**Giải thích sơ đồ usecase**

**Quản lý sách** là usecase tổng thể của nghiệp vụ, bao gồm nhiều tình huống người dùng tương tác với hệ thống để duy trì danh mục sách.

* **Thêm sách**: Nhân viên nhập thông tin sách mới vào hệ thống (ISBN, tiêu đề, giá, số lượng, ...)
* **Sửa sách**: Nhân viên cập nhật thông tin nếu có thay đổi hoặc chỉnh sửa sai sót.
* **Xóa sách (soft delete)**: Nhân viên ẩn sách không còn bán mà không làm mất dữ liệu.
* **Tìm kiếm / lọc sách**: Hỗ trợ tìm nhanh sách theo từ khóa, danh mục, tác giả hoặc nhà xuất bản.

Mỗi nhánh usecase con ở trên có thể được mở rộng thành các dòng xử lý chính (main flow) và dòng ngoại lệ (alternative flow) để phục vụ việc thiết kế test case và kiểm thử.

## 3. Thiết kế hệ thống

**a. Usecase “Quản lý sách”:**

**Thêm sách**

* Giao diện:
* DevExpress Grid kết hợp Popup Edit Form.
* Trường dữ liệu: ISBN, Title, Category, Author, Publisher, YearOfPublication, Price, Image, Quantity, isDeleted
* API cho form:
* POST /api/book → Gửi BookCreateDTO
* Gọi đến BookController.Add() → Service xử lý validate và lưu DB
* Bảng liên quan:
* Bảng Book
* Khóa ngoại: authorId, categoryId, publisherId
* Ràng buộc: isbn unique, price > 0, quantity >= 0

**Sửa sách**

* Giao diện:
* Chọn dòng cần sửa từ Grid → hiển thị lại Popup Edit Form
* Người dùng chỉnh sửa trường và lưu lại
* API cho form:
* PUT /api/book/{id} → Gửi BookUpdateDTO
* BookController.Update() xử lý kiểm tra hợp lệ, ISBN không trùng, và cập nhật vào DB
* Bảng liên quan:
* Bảng Book → cập nhật toàn bộ các trường (trừ ID)

**Xóa sách (soft delete)**

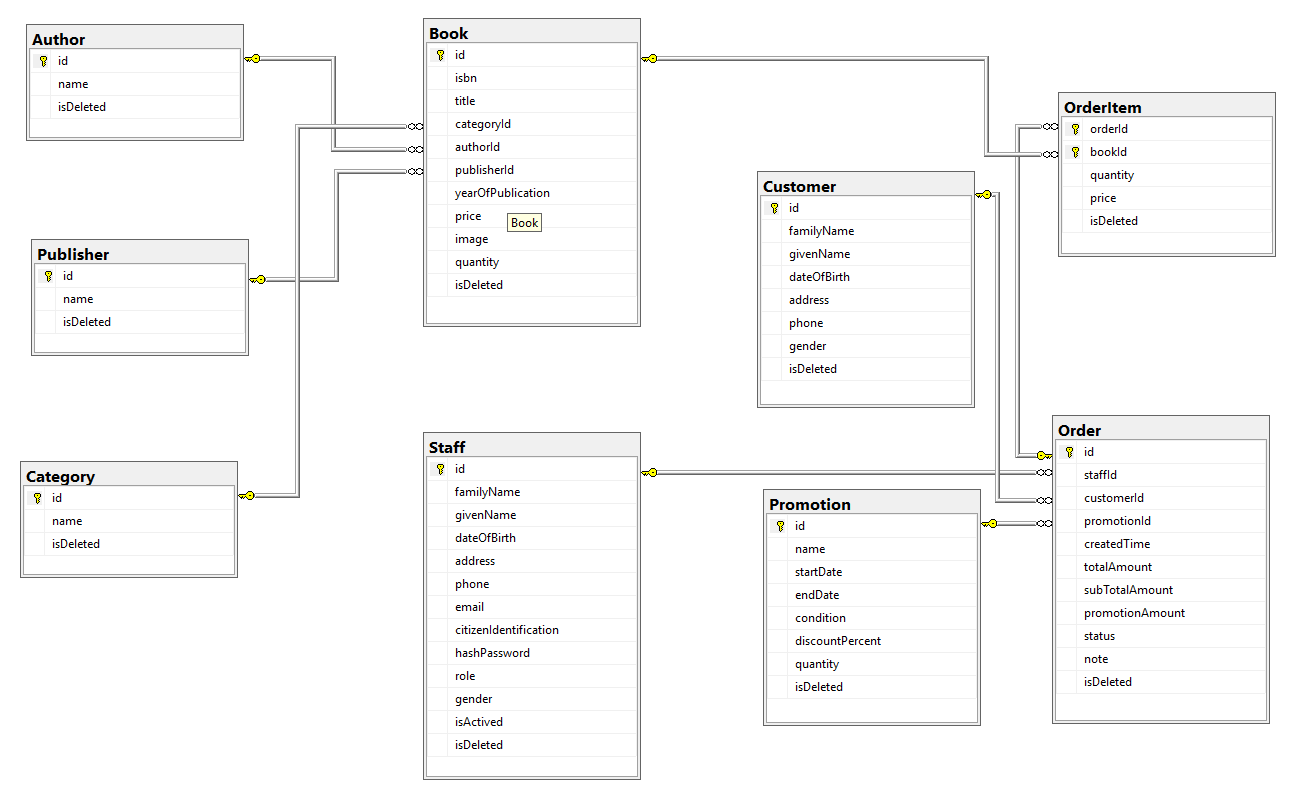
* Giao diện:
* Chọn dòng trong Grid → ấn nút Xóa → hiện thông báo xác nhận → đánh dấu isDeleted = true
* API cho form:
* DELETE /api/book/{id}
* BookController.Delete() → chỉ đánh dấu isDeleted = true, không thực hiện xóa vật lý
* Bảng liên quan:
* Bảng Book, cột isDeleted

**Xem danh sách / tìm kiếm sách**

* Giao diện:
* Grid hiển thị toàn bộ sách, hỗ trợ filter theo cột, có ô tìm kiếm từ khóa (title, tác giả...)
* Cho phép lọc theo Category, Author, Publisher thông qua ComboBox filter
* API cho form:
* GET /api/book – Lấy toàn bộ sách sách
* GET /api/book/search?keyword= – Tìm sách theo từ khóa
* Bảng liên quan:
* Bảng Book kết hợp Category, Author, Publisher qua khóa ngoại

**b. Usecase 2: Form - Api cho form - stored procedures/tables cho Api**

**c. Database của PM:**

****

**Từ điển dữ liệu:**

* **Bảng Category:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | name | nvarchar(100) | UK, NOT NULL | Tên thể loại |
| 3 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng Author:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | name | nvarchar(100) | NOT NULL | Tên tác giả (có thể trùng, đã có trường hợp trùng bút danh) |
| 3 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng Publisher:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Publisher** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | name | nvarchar(100) | UK, NOT NULL | Tên nhà xuất bản |
| 3 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng Book:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Book** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | isbn | char(13) | UK, NOT NULL | Mã isbn |
| 3 | title | nvarchar(100) | NOT NULL | Tên sách |
| 4 | categoryId | uniqueidentifier | FK, NOT NULL | Mã thể loại |
| 5 | authorId | uniqueidentifier | FK, NOT NULL | Mã tác giả |
| 6 | publisherId | uniqueidentifier | FK, NOT NULL | Mã nhà xuất bản |
| 7 | yearOfPublication | smallint | NOT NULL, > 1500 | Năm xuất bản |
| 8 | price | decimal(8, 0) | NOT NULL, > 1000 | Giá tiền ở thời điểm hiện tại |
| 9 | image | varchar(255) | NOT NULL | URL hình ảnh |
| 10 | quantity | int | NOT NULL, >= 0 | Số lượng tồn |
| 11 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng Customer:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Customer** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | familyName | nvarchar(70) | NOT NULL | Họ và tên đệm |
| 3 | givenName | nvarchar(30) | NOT NULL | Tên khách hàng |
| 4 | dateOfBirth | date | NOT NULL | Ngày sinh |
| 5 | address | nvarchar(50) | NOT NULL | Địa chỉ |
| 6 | phone | char(10) | UK, NOT NULL | Số điện thoại |
| 7 | gender | bit | NOT NULL, DEFAULT = 0 | Giới tính, 0 = nam, 1 = nữ |
| 8 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng Staff:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Staff** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | familyName | nvarchar(70) | NOT NULL | Họ và tên đệm |
| 3 | givenName | nvarchar(30) | NOT NULL | Tên |
| 4 | dateOfBirth | date | NOT NULL | Ngày sinh |
| 5 | address | nvarchar(50) | NOT NULL | Địa chỉ |
| 6 | phone | char(10) | UK, NOT NULL | Số điện thoại |
| 7 | email | varchar(50) | UK, NOT NULL | Địa chỉ mail |
| 8 | citizenIdentification | char(12) | UK, NOT NULL | Mã định danh |
| 9 | hashPassword | varchar(255) | NOT NULL | Mật khẩu mã hóa |
| 10 | role | bit | NOT NULL, DEFAULT = 0 | Quyền, 0 = staff, 1 = admin |
| 11 | gender | bit | NOT NULL, DEFAULT = 0 | Giới tính, 0 = nam, 1 = nữ |
| 12 | isActived | bit | NOT NULL, DEFAULT = 0 | Trạng thái kích hoạt tài khoản, 0 = chưa, 1 = rồi |
| 13 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng Promotion:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Promotion** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | name | nvarchar(100) | UK, NOT NULL | Tên khuyến mãi, không được trùng |
| 3 | startDate | datetime2 | NOT NULL | Ngày bắt đầu |
| 4 | endDate | datetime2 | NOT NULL, > startDate | Ngày kết thúc |
| 5 | condition | decimal(8, 0) | NOT NULL, > 1000 | Điều kiện tối thiếu |
| 6 | discountPercent | decimal(3, 2) | NOT NULL, > 0.0 | Phần trăm khuyến mãi |
| 7 | quantity | smallint | NOT NULL, >= 0 | Số lượng |
| 8 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng Order:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Order** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | id | uniqueidentifier | PK, DEFAULT = NEWID() | ID, tự động generate khi tạo mới |
| 2 | staffId | uniqueidentifier | FK, NOT NULL | Mã nhân viên |
| 3 | customerId | uniqueidentifier | FK, NOT NULL | Mã khách hàng |
| 4 | promotionId | uniqueidentifier | FK | Mã khuyến mãi |
| 5 | createdTime | datetime2 | NOT NULL, DEFAULT = SYSDATETIME() | Thời gian tạo hóa đơn |
| 6 | totalAmount | decimal(11, 3) | NOT NULL | Tổng tiền khách phải trả |
| 7 | subTotalAmount | decimal(11, 3) | NOT NULL | Tiền trước khi giảm giá |
| 8 | promotionAmount | decimal(11, 3) | NOT NULL | Tiền giảm giá |
| 9 | status | bit | NOT NULL, DEFAULT = 1 | Trạng thái hóa đơn, 0 = hóa đơn có lỗi, 1 = hóa đơn thanh toán thành công hoặc đã được giải quyết |
| 10 | note | nvarchar(MAX) |  | Ghi chú hóa đơn (khi gặp lỗi) |
| 11 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

* **Bảng OrderItem:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderItem** | | | | |
| **ID** | **Attribute** | **Type** | **Constraint** | **Note** |
| 1 | orderId | uniqueidentifier | PK, FK | Mã hóa đơn |
| 2 | bookId | uniqueidentifier | Mã sách |
| 3 | quantity | smallint | NOT NULL, > 0 | Số lượng |
| 4 | price | decimal(8, 0) | NOT NULL, > 1000 | Giá (ở thời điểm bán) |
| 5 | isDeleted | bit | NOT NULL, DEFAULT = 0 | Trạng thái xóa, mặc định là false |

# CHƯƠNG III. TRIỂN KHAI HỆ THỐNG

## 1. Phần mềm ứng dụng

**1.1. Mô tả Usecase: Quản lý sách**

Usecase "Quản lý sách" mô tả các chức năng nghiệp vụ mà phần mềm cung cấp để nhân viên có thể quản lý danh mục sách trong cửa hàng. Đây là nghiệp vụ quan trọng giúp hệ thống cập nhật kịp thời thông tin về sách phục vụ cho việc bán hàng, khuyến mãi và theo dõi tồn kho.

* Nhân viên có thể **thêm sách mới** vào hệ thống với các thông tin như ISBN, tiêu đề, tác giả, danh mục, nhà xuất bản, năm xuất bản, giá, ảnh, số lượng.
* Nhân viên có thể **sửa thông tin sách** khi cần cập nhật nội dung hoặc chỉnh sửa lỗi.
* Hệ thống hỗ trợ **xóa mềm (soft delete)** sách, tức là đánh dấu isDeleted = true để không hiện thị trong giao diện nhưng không mất dữ liệu gốc.
* Cho phép **tìm kiếm sách theo từ khóa** hoặc **lọc sách** theo các tiêu chí như danh mục, tác giả, nhà xuất bản, năm xuất bản, giá.
* Nhân viên có thể kiểm tra **số lượng tồn kho** để hỗ trợ quyết định nhập hàng sách mới cho kỳ tiếp theo.

**1.2. Giao diện phần mềm**

Hệ thống sử dụng giao diện WebApp kết hợp DevExpress Grid, cho phép người dùng tương tác trực quan với danh sách sách. Các chức năng bao gồm:

* **Danh sách sách** hiển thị toàn bộ thông tin (ISBN, tiêu đề, tác giả, danh mục, giá, tồn kho...)
* **Các nút thao tác**: Create, View, Delete, Order, Search
* **Popup form** để thêm/sửa sách hiển thị đầy đủ các trường nhập liệu
* **Bộ lọc (filter)** theo từng cột, hỗ trợ tìm nhanh

Hình minh họa giao diện (xem trong phụ lục hoặc phần hình ảnh demo)

**1.3. Các API hỗ trợ**

Các API backend được xây dựng bằng ASP.NET Core MVC, phân tách rõ controller, service và repository:

* GET /api/book – Lấy danh sách sách
* GET /api/book/{id} – Lấy chi tiết sách theo ID
* GET /api/book/search?keyword= – Tìm kiếm sách
* POST /api/book – Thêm sách mới
* PUT /api/book/{id} – Cập nhật sách
* DELETE /api/book/{id} – Xóa mềm sách

## 2. Kiểm thử

**2.1. Rà soát tài liệu (Verification & Traceability)**

**a. Truy vết yêu cầu → thiết kế → test case**

|  |  |  |  |
| --- | --- | --- | --- |
| **Yêu cầu chức năng** | **Giao diện (BookPage.razor)** | **DTO (BookCreateDTO)** | **Logic (BookService)** |
| ISBN không trùng | Trường nhập ISBN + ValidationMessage | [Required], Regex, Unique | Check tồn tại trong DB |
| Title không rỗng | Trường Title + kiểm lỗi | [Required], Regex | Kiểm tra null/space |
| Quantity ≥ 0 | SpinEdit Quantity | Kiểu int | Check >= 0 |
| Image không null | Textbox | [Required] | Kiểm tra khác null |
| Year > 0 | SpinEdit năm | short | Check > 1500 |
| Category/Author/Publisher hợp lệ | ComboBox | GUID | Kiểm tra tồn tại và chưa xóa |

**b. Lỗi tiềm ẩn nếu không rà soát:**

* Nhập dữ liệu thiếu → gây lỗi logic hoặc hỏng giao diện
* ISBN trùng → phá vỡ tính toàn vẹn hệ thống
* Image null → lỗi hiển thị ảnh trong grid

**c. Coverage:**

* Bao phủ đầy đủ các input chính và ràng buộc logic
* Kết nối đầy đủ giữa tài liệu – giao diện – service – test case

**2.2. Thiết kế test case (Blackbox Testing)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC | Mô tả | Input | Kết quả mong đợi | Loại |
| TC01 | Thêm sách hợp lệ | ISBN = "1234567890123", đủ các trường | Đã thêm sách thành công. | Positive |
| TC02 | ISBN rỗng | ISBN = "" | Lỗi "ISBN là bắt buộc" | Negative |
| TC03 | ISBN trùng | ISBN đã tồn tại | Lỗi "ISBN đã tồn tại" | Negative |
| TC04 | Title rỗng | Title = "" | Lỗi "Tiêu đề là bắt buộc" | Negative |
| TC05 | Title là khoảng trắng | Title = " " | Lỗi Regex | Negative |
| TC06 | Quantity âm | Quantity = -2 | Lỗi logic | Boundary |
| TC07 | Price = 0 | Price = 0 | Lỗi "Giá > 0" | Boundary |
| TC08 | Image = null | Image = null | Lỗi "Không được null" | Negative |
| TC09 | CategoryId null | CategoryId = null | Lỗi "Danh mục không hợp lệ" | Negative |
| TC10 | AuthorId null | AuthorId = null | Lỗi "Tác giả không hợp lệ" | Negative |
| TC11 | PublisherId null | PublisherId = null | Lỗi "Nhà xuất bản không hợp lệ" | Negative |

Mỗi test case đã được thực thi và ghi lại kết quả (có ảnh minh chứng).

**2.3. Thực hiện kiểm thử (ghi log kết quả)**

|  |  |  |  |
| --- | --- | --- | --- |
| TC | Kết quả mong đợi | Thực tế | Trạng thái |
| TC01 | Thêm sách thành công | Hiển thị thông báo "Thành công" | PASSED |
| TC02 | Lỗi ISBN rỗng | Lỗi hiển thị đúng | PASSED |
| TC03 | Lỗi ISBN trùng | Thông báo lỗi chính xác | PASSED |
| TC04 | Lỗi Title rỗng | Hiển thị lỗi | PASSED |
| TC05 | Title là khoảng trắng | Regex không cho phép | PASSED |
| TC06 | Quantity âm | Không cho lưu, hiện lỗi | PASSED |
| TC07 | Price = 0 | Cảnh báo "Giá phải > 0" | PASSED |
| TC08 | Image null | Hiện thông báo lỗi | PASSED |
| TC09 | CategoryId null | Không thể submit, lỗi rõ ràng | PASSED |
| TC10 | AuthorId null | Lỗi "Tác giả không hợp lệ" | PASSED |
| TC11 | PublisherId null | Hiện lỗi chính xác | PASSED |

**2.4. Kiểm thử Whitebox (BookService.cs)**

**Mục tiêu:**

* Kiểm tra các logic điều kiện trong AddAsync()

**Cách thực hiện:**

* Viết Unit Test bằng xUnit hoặc MSTest
* Mock repository và kiểm tra từng điều kiện if

**Ví dụ:**

[TestMethod]

public void AddBook\_WithDuplicateISBN\_ShouldThrow()

{

var dto = new BookCreateDTO { Isbn = "123", Title = "abc", ... };

mockRepo.Setup(r => r.GetByIsbnAsync("123")).ReturnsAsync(new Book());

var ex = Assert.ThrowsExceptionAsync<InvalidOperationException>(() => service.AddAsync(dto));

Assert.AreEqual("Đã có một cuốn sách có cùng mã ISBN.", ex.Result.Message);

}

**Bao phủ:**

* Statement Coverage: Bao phủ các dòng code if, throw
* Branch Coverage: Có test pass/fail từng điều kiện
* Condition Coverage: Kiểm tra biểu thức logic và các nhánh lỗi

# CHƯƠNG IV. KẾT LUẬN

## 1. Nhận xét về điểm mạnh, điểm yếu trong cách tiến hành đồ án

## 2. Khả năng cải tiến cách tiến hành đồ án

# TÀI LIỆU THAM KHẢO