

## Contents

<b>DANH MỤC KÝ HIỆU VÀ VIẾT TẮT .....</b>	<b>ii</b>
<b>DANH MỤC HÌNH VẼ .....</b>	<b>iii</b>
<b>DANH MỤC BẢNG .....</b>	<b>v</b>
<b>CHƯƠNG 1. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>1</b>
1.1. Khảo sát và phân tích yêu cầu.....	1
1.1.1. <i>Mô tả Bài toán và Hiện trạng</i> .....	1
1.1.2. <i>Xác định Đối tượng Người dùng và Vai trò</i> .....	1
1.1.3. <i>Yêu cầu Chức năng (Functional Requirements)</i> .....	2
1.1.4. <i>Yêu cầu Phi chức năng (Non-Functional Requirements)</i> .....	3
1.2. Thiết kế kiến trúc hệ thống tổng thể .....	4
1.2.1. <i>Mô hình kiến trúc tổng quan</i> .....	5
1.2.2. <i>Thiết kế kiến trúc API Backend</i> .....	7
1.2.3. <i>Thiết kế kiến trúc ứng dụng Client (MAUI)</i> .....	9
1.3. Thiết kế mô hình .....	12
1.3.1. <i>Mô hình UseCase – UseCase Diagram</i> .....	13
1.3.2. <i>Mô hình hoạt động - Activity Diagram</i> .....	18
1.3.3. <i>Mô hình tuần tự - Sequence Diagram</i> .....	19
1.3.4. <i>Mô hình lớp - Class Diagram</i> .....	19
1.4. Thiết kế cơ sở dữ liệu.....	21
1.4.1. <i>Mô hình dữ liệu quan hệ ERD</i> .....	21
1.4.2. <i>Mô tả mô hình quan hệ</i> .....	23
1.5. Kết luận chương .....	28
<b>CHƯƠNG 2. TRIỂN KHAI HỆ THỐNG .....</b>	<b>29</b>
2.1. Kết quả thực nghiệm .....	29
2.1.1. <i>Giao diện khởi động và xác thực</i> .....	29
2.1.2. <i>Chức năng duyệt sản phẩm</i> .....	30
2.1.3. <i>Chức năng tương tác cá nhân</i> .....	31
2.1.4. <i>Chức năng quản lý tài khoản</i> .....	32
2.1.5. <i>Chức năng cho luồng đặt hàng Online</i> .....	33
2.1.6. <i>Chức năng Admin DashBoard và báo cáo</i> .....	34
2.1.7. <i>Chức năng quản lý đơn hàng</i> .....	34
2.1.8. <i>Chức năng quản lý sản phẩm</i> .....	35
2.1.9. <i>Chức năng quản lý kho hàng</i> .....	35
2.1.10. <i>Các chức năng quản lý khác</i> .....	36
2.2. Đánh giá hiệu năng.....	36
<b>PHỤ LỤC .....</b>	<b>38</b>
PHỤ LỤC : SƠ ĐỒ TUẦN TỰ - ACTIVITY DIAGRAM.....	38
PHỤ LỤC D: SƠ ĐỒ TUẦN TỰ - SEQUENCE DIAGRAM.....	46

## **DANH MỤC KÝ HIỆU VÀ VIẾT TẮT**

Ký tự viết tắt	Giải thích
.NET MAUI	.NET Multi-platform App UI
API	Application Programming Interface
MVVM	Model-View-ViewModel
EF Core	Entity Framework Core
JWT	JSON Web Token
DTO	Data Transfer Object
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
GCS	Google Cloud Storage
UML	Unified Modeling Language
ERD	Entity Relationship Diagram
RDBMS	Relational Database Management System
ORM	Object-Relational Mapping
XAML	Extensible Application Markup Language
UI	User Interface
SDK	Software Development Kit
ISBN	International Standard Book Number
CRUD	Create, Read, Update, Delete
POS	Point of Sale
CDN	Content Delivery Network
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery
DI	Dependency Injection
BFD	Business Function Diagram

## DANH MỤC HÌNH VẼ

Hình 7. Yêu cầu chức năng của hệ thống quản lý nhà sách.....	3
Hình 8. Mô hình kiến trúc tổng quan .....	5
Hình 9. Sơ đồ các lớp và luồng phụ thuộc trong API Backend .....	8
Hình 10. Mô hình kiến trúc MVVM.....	10
Hình 11. Cấu trúc dự án MAUI.....	12
Hình 12. Sơ đồ Use Case tổng quát cho hệ thống quản lý nhà sách.....	13
Hình 13. Mô hình lớp – Class Diagram tổng quát .....	20
Hình 14. Mô hình dữ liệu quan hệ ERD .....	22
Hình 15. Giao diện khởi động và xác thực .....	30
Hình 16. Giao diện duyệt sản phẩm.....	31
Hình 17. Giao diện tương tác cá nhân.....	32
Hình 18. Giao diện quản lý tài khoản .....	33
Hình 19. Giao diện luồng đặt hàng Online .....	33
Hình 20. Giao diện Admin DashBoard và báo cáo.....	34
Hình 21. Giao diện quản lý đơn hàng .....	34
Hình 22. Giao diện chức năng quản lý sản phẩm .....	35
Hình 23. Giao diện chức năng quản lý kho hàng.....	35
Hình 24. Giao diện các chức năng quản lý khác.....	36
Hình 25. Đánh giá hiệu năng của một số API nổi bật.....	37
Hình 27. Activity Diagram - Tìm kiếm và xem chi tiết sách.....	39
Hình 28. Activity Diagram - Thêm vào yêu thích .....	40
Hình 29. Activity Diagram - Thêm sách vào giỏ hàng .....	41
Hình 30. Activity Diagram - Admin quản lý người dùng.....	42
Hình 31. Activity Diagram - Admin thêm sách .....	43
Hình 32. Activity Diagram - Admin cập nhật trạng thái đơn hàng .....	44
Hình 33. Activity Diagram - Admin nhập kho .....	45
Hình 34. Sequence Diagram - Đăng nhập.....	46
Hình 35. Sequence Diagram - Thêm vào yêu thích .....	47
Hình 36. Sequence Diagram - Xem nhà cung cấp .....	47
Hình 37. Sequence Diagram - Xem chi tiết sách .....	48
Hình 38. Sequence Diagram - Thanh toán .....	49
Hình 39. Sequence Diagram - Tạo phiếu nhập .....	50
Hình 40. Sequence Diagram - Tạo đơn hàng tại quầy .....	51



## **DANH MỤC BẢNG**

Bảng 2. Use Case Đăng ký tài khoản.....	13
Bảng 3. Use Case Đăng nhập hệ thống .....	14
Bảng 4. Use Case Đặt hàng sách online.....	15
Bảng 5. Use Case Đánh giá sản phẩm.....	16
Bảng 6. Use Case Tạo đơn hàng tại quầy .....	16
Bảng 7. Use Case Nhập kho sách.....	17
Bảng 8. Use Case Quản lý sản phẩm .....	17
Bảng 9. Use Case Xem báo cáo doanh thu .....	18
Bảng 10. Roles Table .....	23
Bảng 11. Users Table .....	23
Bảng 12. Categories Table .....	23
Bảng 13. Authors Table .....	24
Bảng 14. Books Table .....	24
Bảng 15. Addresses Table.....	25
Bảng 16. CartItems Table.....	25
Bảng 17. WishlistItems Table .....	26
Bảng 18. Orders Table .....	26
Bảng 19. OrderDetails Table.....	26
Bảng 20. Promotions Table.....	27
Bảng 21. Reviews Table .....	27
Bảng 22. InventoryLogs Table.....	28

# CHƯƠNG 1. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 1.1. Khảo sát và phân tích yêu cầu

### 1.1.1. Mô tả Bài toán và Hiện trạng

Hiện nay, nhu cầu mua sắm và quản lý sách ngày càng tăng cao. Các nhà sách truyền thống đang dần chuyển đổi và mở rộng sang mô hình kinh doanh trực tuyến để tiếp cận được nhiều khách hàng hơn. Đồng thời, việc quản lý nội bộ một nhà sách, bao gồm quản lý sản phẩm, tồn kho, đơn hàng, khách hàng, và nhân viên, cũng đòi hỏi một hệ thống thông tin hiệu quả để tối ưu hóa vận hành và nâng cao chất lượng dịch vụ.

Bài toán đặt ra là phát triển một hệ thống "Quản lý Nhà sách" toàn diện, bao gồm hai thành phần chính:

- Ứng dụng di động (.NET MAUI): Dành cho Khách hàng thực hiện các thao tác mua sắm trực tuyến và quản lý tài khoản cá nhân. Đồng thời, ứng dụng này cũng cung cấp giao diện cho Nhân viên (Staff) và Quản trị viên (Admin) để thực hiện các tác vụ quản lý và vận hành cửa hàng ngay trên thiết bị di động.
- API Backend (ASP.NET Core Web API): Là trung tâm xử lý logic nghiệp vụ, quản lý dữ liệu và cung cấp các endpoints cho ứng dụng di động tương tác.

Hệ thống này không chỉ phục vụ cho việc bán hàng online mà còn hỗ trợ quản lý các hoạt động tại cửa hàng vật lý, bao gồm việc tạo đơn hàng tại quầy và quản lý nhà cung cấp. Mục tiêu là tạo ra một giải pháp tích hợp, giúp nhà sách vận hành hiệu quả hơn, tăng cường trải nghiệm khách hàng và cung cấp công cụ quản lý mạnh mẽ cho nhân viên và quản trị viên.

### 1.1.2. Xác định Đối tượng Người dùng và Vai trò

Hệ thống quản lý nhà sách được xây dựng nhằm phục vụ ba nhóm người dùng chính, mỗi nhóm có đặc điểm riêng biệt về quyền truy cập và mục tiêu sử dụng. Đầu tiên là khách hàng, những người sử dụng ứng dụng để tra cứu thông tin sách, đặt hàng trực tuyến, theo dõi đơn hàng và quản lý thông tin cá nhân. Với nhóm đối tượng này, trải nghiệm người dùng là yếu tố then chốt; hệ thống cần mang lại giao diện thân thiện, tốc độ tìm kiếm nhanh chóng, quy trình đặt hàng thuận tiện và khả năng quản lý thông tin một cách linh hoạt.

Nhóm thứ hai là nhân viên cửa hàng, những người trực tiếp hỗ trợ khách tại điểm bán. Vai trò của họ thiên về các tác vụ vận hành như tạo đơn hàng tại quầy, kiểm tra thông tin tồn kho, xử lý đơn hàng và nhập kho hàng hóa. Đối với họ, hệ

thống cần đảm bảo hiệu suất cao, thao tác đơn giản và cung cấp thông tin chính xác để hỗ trợ quy trình làm việc hiệu quả.

Cuối cùng là quản trị viên, người có quyền kiểm soát toàn bộ hệ thống. Nhóm này không chỉ cần công cụ quản lý dữ liệu sách, người dùng, đơn hàng và kho, mà còn cần các chức năng nâng cao như quản lý khuyến mãi, kiểm duyệt đánh giá người dùng và theo dõi hoạt động kinh doanh thông qua các báo cáo thống kê trực quan. Vai trò của quản trị viên mang tính điều hành và chiến lược, đòi hỏi hệ thống có khả năng mở rộng, bảo mật cao và trực quan trong vận hành.

### *1.1.3. Yêu cầu Chức năng (Functional Requirements)*

Dựa trên phân tích về các nhóm người dùng và nhu cầu đặc thù của từng vai trò, hệ thống cần đáp ứng một tập hợp chức năng toàn diện. Đối với người dùng nói chung, kể cả chưa đăng nhập, hệ thống cần cung cấp khả năng duyệt danh sách sách, lọc và tìm kiếm theo từ khóa hoặc tiêu chí cụ thể như tên, tác giả, ISBN. Người dùng cũng cần xem chi tiết thông tin sách, bao gồm mô tả, giá cả, tình trạng tồn kho, và đánh giá từ các người dùng khác.

Khi đã đăng nhập, khách hàng có thể thực hiện các tác vụ liên quan đến cá nhân như đăng ký, đăng nhập, chỉnh sửa hồ sơ, quản lý địa chỉ giao hàng, danh sách yêu thích và giỏ hàng. Họ cũng có khả năng thực hiện quá trình đặt hàng trực tuyến với các bước như chọn địa chỉ, áp dụng mã giảm giá và thực hiện thanh toán (giả lập). Sau khi đặt hàng, người dùng có thể theo dõi trạng thái đơn hàng, xem lịch sử giao dịch, hủy đơn hàng nếu được phép, và để lại đánh giá cho các sản phẩm đã mua.

Nhân viên cửa hàng ngoài việc kế thừa các quyền của khách hàng, còn có thể tạo đơn hàng tại quầy cho khách lẻ hoặc khách thành viên, cập nhật trạng thái đơn hàng, kiểm tra tồn kho và quản lý các nghiệp vụ nhập kho sản phẩm. Điều này hỗ trợ hoạt động bán hàng trực tiếp tại cửa hàng một cách hiệu quả và minh bạch.

Đối với quản trị viên, hệ thống cần hỗ trợ quản lý toàn bộ dữ liệu trọng yếu như người dùng, danh mục sách, tác giả, nhà cung cấp và các chương trình khuyến mãi. Ngoài ra, quản trị viên còn thực hiện việc kiểm duyệt đánh giá từ người dùng và theo dõi các chỉ số kinh doanh như doanh thu, tình trạng tồn kho và sách bán chạy thông qua các báo cáo có khả năng lọc và trực quan hóa.

Tất cả các chức năng nêu trên sẽ được hệ thống tổ chức và triển khai hợp lý theo mô hình phân rã chức năng, đảm bảo khả năng mở rộng, bảo trì và nâng cấp trong tương lai. Nếu cần, có thể tham khảo sơ đồ BFD được trình bày ở cuối chương hoặc trong phần phụ lục để hiểu rõ hơn về cấu trúc chức năng của toàn hệ thống.



Hình 1. Yêu cầu chức năng của hệ thống quản lý nhà sách

#### 1.1.4. Yêu cầu Phi chức năng (Non-Functional Requirements)

Ngoài các chức năng chính, hệ thống cần tuân thủ một tập hợp các yêu cầu phi chức năng nhằm đảm bảo tính hiệu quả, bảo mật, độ tin cậy và khả năng mở rộng trong quá trình vận hành. Các yêu cầu này được phân loại cụ thể như sau:

##### *Hiệu năng (Performance):*

- Thời gian phản hồi cho các yêu cầu API phổ biến (ví dụ: truy vấn danh sách sách, xem chi tiết sách) cần được tối ưu hóa, mục tiêu trung bình dưới 500 mili-giây.
- Ứng dụng MAUI phải đảm bảo khởi động nhanh, thao tác cuộn danh sách mượt mà và chuyển trang không bị gián đoạn.
- Hệ thống cần hỗ trợ xử lý đồng thời lượng truy cập ở mức vừa phải mà không bị nghẽn, ước tính khoảng 50–100 yêu cầu/giây cho API.

##### *Bảo mật (Security):*

- Mật khẩu người dùng phải được mã hóa an toàn bằng thuật toán hash trước khi lưu trữ trong cơ sở dữ liệu.
- Giao tiếp giữa ứng dụng MAUI và API backend phải được bảo vệ bởi giao thức HTTPS để tránh nguy cơ nghe lén hoặc tấn công trung gian.
- Việc xác thực người dùng sử dụng JWT (JSON Web Token), với token được truyền thông qua header Authorization, có thời gian hiệu lực hợp lý và cơ chế làm mới an toàn.
- Cơ chế phân quyền truy cập tài nguyên và chức năng dựa trên vai trò người dùng (role-based access control) phải được áp dụng một cách nghiêm ngặt trên cả client và server.

- Dữ liệu đầu vào phải được xác thực (validation) nghiêm ngặt ở cả phía client (.NET MAUI) và phía server (Web API) nhằm ngăn chặn các lỗi logic và các lỗ hổng bảo mật phổ biến như SQL Injection, Cross-Site Scripting (XSS), và Cross-Site Request Forgery (CSRF).

#### *Độ tin cậy và tính sẵn sàng (Reliability & Availability):*

- Hệ thống phải duy trì hoạt động ổn định, hạn chế tối đa thời gian ngừng dịch vụ (downtime).
  - Cơ sở dữ liệu cần được sao lưu định kỳ, hỗ trợ khôi phục nhanh chóng trong trường hợp xảy ra sự cố.
  - Việc xử lý lỗi phải thông nhất, đồng thời cung cấp thông báo rõ ràng, có tính hướng dẫn cho người dùng cuối.

#### *Tính dễ sử dụng (Usability):*

- Giao diện người dùng của ứng dụng MAUI cần được thiết kế theo hướng trực quan, dễ học và dễ thao tác, phù hợp với đặc điểm của ba nhóm người dùng: khách hàng, nhân viên và quản trị viên.
  - Luồng thực hiện các tác vụ chính (như đặt hàng, quản lý tồn kho, xem báo cáo) cần logic, thuận tiện và giảm thiểu số bước không cần thiết.

#### *Khả năng bảo trì (Maintainability):*

- Mã nguồn cần tuân thủ kiến trúc phân lớp rõ ràng: Web API sử dụng mô hình Layered Architecture; ứng dụng MAUI tuân theo kiến trúc MVVM.
- Áp dụng quy ước đặt tên nhất quán, đồng thời cung cấp chú thích (comment) cho các đoạn mã phức tạp.
- Hệ thống phải dễ dàng bảo trì, sửa lỗi và mở rộng tính năng, trong khi vẫn đảm bảo hạn chế tối đa ảnh hưởng đến các thành phần khác.

#### *Khả năng mở rộng (Scalability):*

- Thiết kế cơ sở dữ liệu và API phải có khả năng thích ứng với sự gia tăng dữ liệu và người dùng trong tương lai.
- Việc sử dụng GUID làm khóa chính cùng với kiến trúc hướng dịch vụ (service-oriented architecture) giúp nâng cao khả năng mở rộng và triển khai theo mô hình phân tán nếu cần thiết.

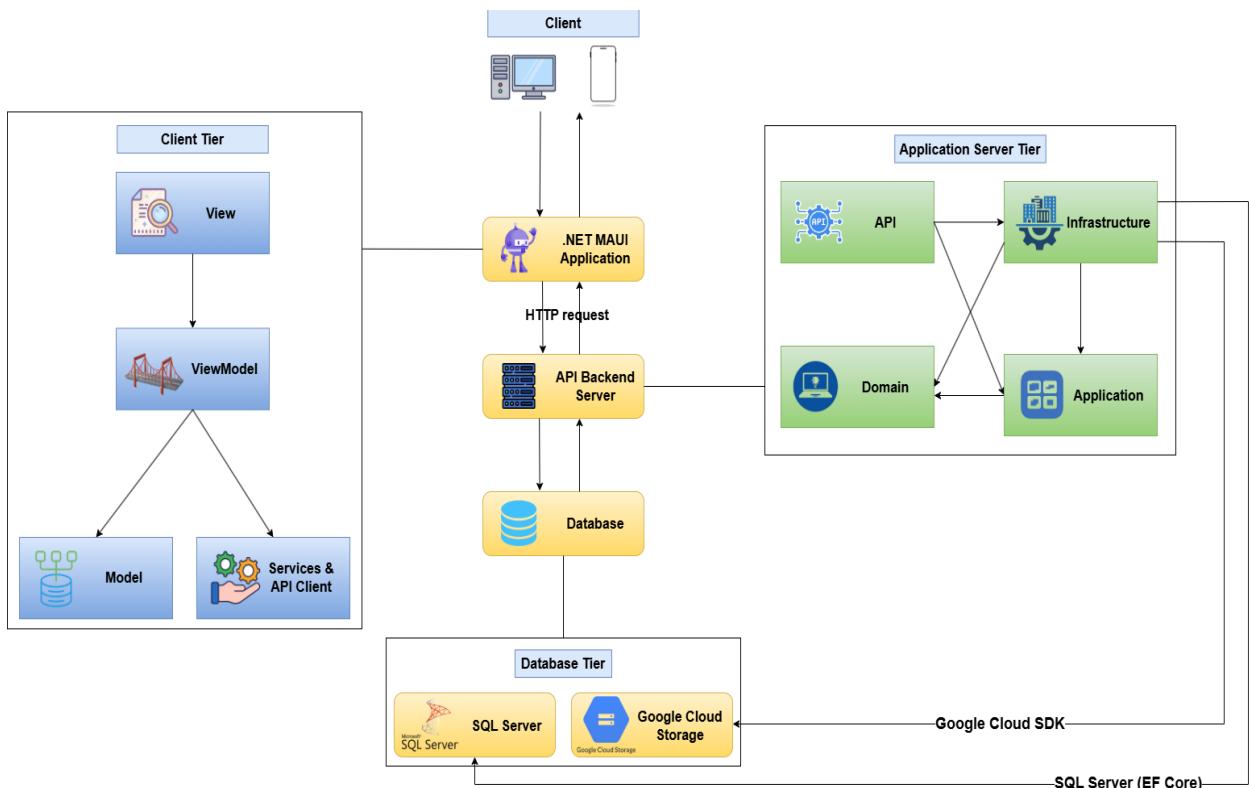
## **1.2. Thiết kế kiến trúc hệ thống tổng thể**

Sau khi đã xác định rõ các yêu cầu chức năng và phi chức năng, bước tiếp theo là thiết kế kiến trúc tổng thể cho hệ thống Quản lý Nhà sách. Kiến trúc này đóng vai trò là bản đồ khung, định hình cách các thành phần chính của hệ thống được tổ

chức, tương tác với nhau và đáp ứng các yêu cầu đã đặt ra. Một kiến trúc được thiết kế tốt sẽ đảm bảo tính module hóa, khả năng bảo trì, mở rộng và hiệu năng của toàn bộ hệ thống.

### 1.2.1. Mô hình kiến trúc tổng quan

Hệ thống Quản lý Nhà sách được thiết kế theo mô hình kiến trúc nhiều lớp (multi-tier architecture) phổ biến, bao gồm các thành phần chính tương tác với nhau thông qua các giao diện được định nghĩa rõ ràng. Sơ đồ dưới đây minh họa kiến trúc tổng quan của hệ thống:



Hình 2. Mô hình kiến trúc tổng quan

#### Tầng Client (Client Tier):

Tầng này bao gồm ứng dụng .NET MAUI, đóng vai trò là giao diện người dùng (UI), nơi các nhóm người dùng như Khách hàng, Nhân viên (Staff) và Quản trị viên (Admin) tương tác trực tiếp với hệ thống. Ứng dụng MAUI chịu trách nhiệm:

- Hiển thị dữ liệu từ hệ thống theo vai trò người dùng.
- Thu thập và xác thực dữ liệu đầu vào từ người dùng thông qua kiểm tra đầu vào (client-side validation).
- Gửi các yêu cầu đến API backend thông qua giao thức HTTPS nhằm đảm bảo an toàn truyền thông.

- Trao đổi dữ liệu với API ở định dạng JSON, sử dụng các đối tượng trung chuyển dữ liệu (Data Transfer Objects – DTOs) để đảm bảo tính thống nhất và rõ ràng trong giao tiếp.

#### *Tầng Máy chủ Ứng dụng (Application Server Tier):*

Thành phần cốt lõi ở tầng này là ASP.NET Core Web API, đóng vai trò xử lý toàn bộ logic nghiệp vụ và điều phối hoạt động của hệ thống. Cụ thể, tầng này:

- Cung cấp các điểm truy cập RESTful thông qua HTTP endpoints, cho phép ứng dụng MAUI gửi và nhận dữ liệu một cách linh hoạt.
- Thực hiện xác thực (authentication) và phân quyền (authorization) người dùng dựa trên vai trò.
- Thực hiện xử lý logic nghiệp vụ như tạo đơn hàng, tính phí vận chuyển, kiểm tra kho, thống kê doanh thu, v.v.
- Giao tiếp với tầng dữ liệu thông qua Entity Framework Core, cho phép truy xuất và cập nhật cơ sở dữ liệu một cách hiệu quả và an toàn.

#### *Tầng Dữ liệu (Data Tier):*

Tầng dữ liệu lưu trữ toàn bộ thông tin có cấu trúc của hệ thống và cung cấp nền tảng dữ liệu cho các thao tác nghiệp vụ. Bao gồm hai thành phần chính:

- Cơ sở dữ liệu SQL Server: Lưu trữ các thực thể chính của hệ thống như người dùng, sách, danh mục, đơn hàng, tồn kho, lịch sử giao dịch,... Các thao tác dữ liệu được thực hiện thông qua Entity Framework Core để đảm bảo tính nhất quán và dễ bảo trì.
- Google Cloud Storage (GCS): Được sử dụng cho việc lưu trữ các tệp hình ảnh, đặc biệt là ảnh bìa sách. API backend sẽ thực hiện tải ảnh lên GCS và lưu lại đường dẫn URL tương ứng trong CSDL. Ứng dụng MAUI sau đó sẽ hiển thị ảnh thông qua các URL này, giúp tối ưu hóa hiệu suất tải và giảm tải cho máy chủ.

#### **Luồng tương tác chính:**

1. Người dùng (Customer, Staff, Admin) thực hiện một thao tác trên ứng dụng .NET MAUI.
2. Ứng dụng MAUI tạo một HTTP request (ví dụ: GET, POST, PUT) chứa các thông tin cần thiết (header, body DTO) và gửi đến một endpoint cụ thể của ASP.NET Core Web API qua HTTPS.
3. ASP.NET Core Web API tiếp nhận request, thực hiện xác thực (nếu cần), validation dữ liệu đầu vào.
4. API gọi đến các Application Service để xử lý logic nghiệp vụ.

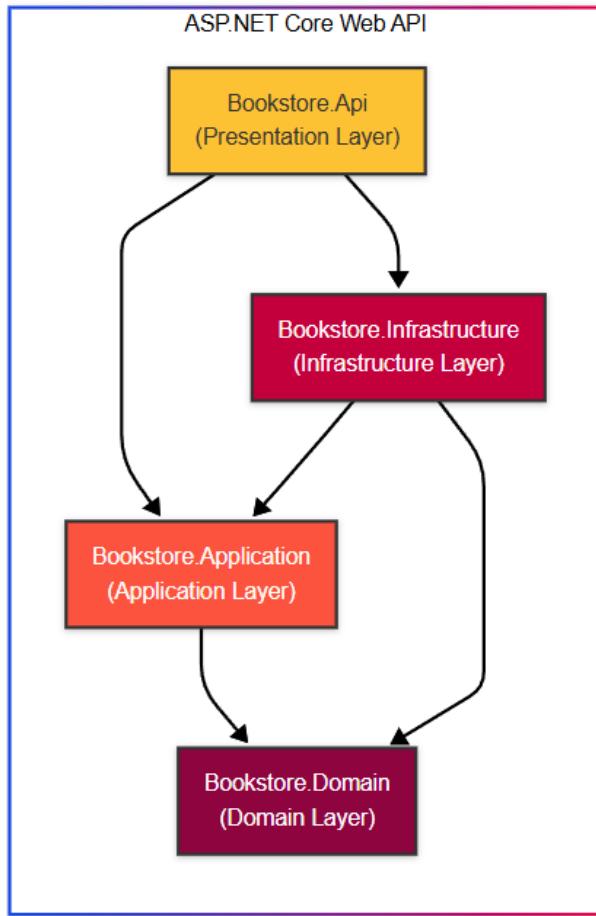
5. Application Service tương tác với các Repository (qua Unit of Work) để truy vấn hoặc cập nhật dữ liệu trong CSDL SQL Server.
6. Nếu thao tác liên quan đến file (ví dụ: upload ảnh bìa), API sẽ tương tác với Google Cloud Storage.
7. Sau khi xử lý xong, API Backend trả về một HTTP response (chứa dữ liệu DTO ở dạng JSON hoặc chỉ status code) cho ứng dụng MAUI.
8. Ứng dụng MAUI nhận response, xử lý dữ liệu (deserialize JSON) và cập nhật giao diện người dùng tương ứng.

#### *1.2.2. Thiết kế kiến trúc API Backend*

API Backend được xây dựng bằng ASP.NET Core, tuân theo Kiến trúc Phân lớp (Layered Architecture) đã được trình bày chi tiết ở Mục 1.4.3. Việc áp dụng kiến trúc này nhằm mục đích:

- Tăng tính module hóa (Modularity): Chia nhỏ hệ thống thành các thành phần độc lập với trách nhiệm rõ ràng.
- Giảm sự phụ thuộc (Decoupling): Các lớp chỉ tương tác với nhau thông qua các interface được định nghĩa rõ ràng, giảm thiểu sự phụ thuộc trực tiếp vào implementation cụ thể.
- Tăng khả năng kiểm thử (Testability): Dễ dàng viết Unit Test cho từng lớp một cách độc lập bằng cách mock các dependency.
- Tăng khả năng bảo trì và mở rộng (Maintainability & Scalability): Việc thay đổi hoặc thêm mới chức năng trong một lớp ít ảnh hưởng đến các lớp khác.

Sơ đồ các lớp và luồng phụ thuộc trong API Backend được minh họa như sau:



Hình 3. Sơ đồ các lớp và luồng phụ thuộc trong API Backend

- *Bookstore.Domain (Domain Layer)*: Là lớp lõi, chứa các Entities (Sách, Đơn hàng, Người dùng...), Enums, và các Repository Interfaces (ví dụ: IBook Repository, IOrder Repository). Lớp này không phụ thuộc vào bất kỳ lớp nào khác.
- *Bookstore.Application (Application Layer)*: Chứa logic nghiệp vụ ứng dụng và điều phối các use case. Nó định nghĩa các Service Interfaces (ví dụ: IBookService, IOrderService) và các lớp Service Implementations. Nó cũng chứa các Data Transfer Objects (DTOs) dùng để trao đổi dữ liệu, các Validator (FluentValidation), và các AutoMapper Profile. Lớp Application phụ thuộc vào Domain Layer. Giao diện IUnitOfWork cũng được định nghĩa ở đây để quản lý các transaction nghiệp vụ.
- *Bookstore.Infrastructure (Infrastructure Layer)*: Cung cấp các triển khai cụ thể cho các interface từ Domain và Application. Bao gồm Application DbContext (EF Core), các Repository Implementations (truy cập CSDL SQL Server), UnitOfWork implementation, và các dịch vụ kỹ thuật khác như Token-Service (tạo JWT), GoogleCloudStorageService (lưu file ảnh). Lớp Infrastructure phụ thuộc vào Domain và Application.

- *Bookstore.Api (Presentation Layer)*: Chứa các API Controller, Middleware, và cấu hình khởi tạo ứng dụng (Program.cs). Controllers nhận HTTP request, sử dụng các Application Service (được inject qua DI) để xử lý, và trả về HTTP response. Lớp này phụ thuộc vào Application Layer để gọi services và Infrastructure Layer (chủ yếu để đăng ký các dịch vụ vào DI container).

Luồng xử lý một request API điển hình qua các lớp:

1. Client gửi một HTTP request đến endpoint trong Bookstore.Api.
2. Controller nhận Request DTO, thực hiện xác thực dữ liệu nếu cần và gọi phương thức trong Application Service.
3. Application Service xử lý logic nghiệp vụ, sử dụng các Repository Interface (từ lớp Domain) thông qua IUnitOfWork.
4. UnitOfWork sử dụng các repository đã được triển khai trong Bookstore.Infrastructure để tương tác với cơ sở dữ liệu thông qua EF Core.
5. Nếu có thao tác với file (ví dụ: ảnh bìa sách), Application Service gọi đến IFileStorageService (được GoogleCloudStorageService triển khai).
6. Kết quả truy vấn dữ liệu (Entities) được ánh xạ sang Response DTO nhờ thư viện AutoMapper.
7. Application Service trả Response DTO cho Controller.
8. Controller gửi HTTP Response về phía client.

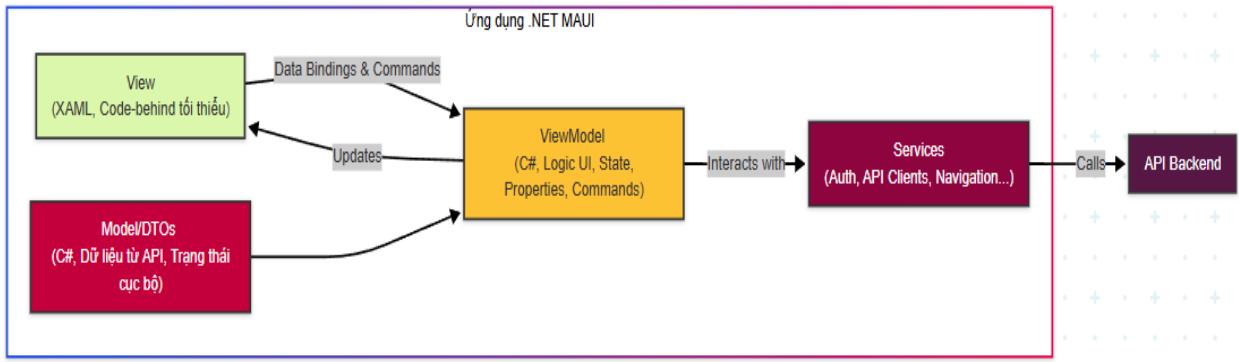
Kiến trúc phân lớp này giúp cô lập các mối quan tâm, làm cho hệ thống API trở nên rõ ràng, dễ quản lý và linh hoạt hơn trong quá trình phát triển và bảo trì.

#### *1.2.3. Thiết kế kiến trúc ứng dụng Client (MAUI)*

Ứng dụng client .NET MAUI đóng vai trò là giao diện người dùng chính, cung cấp trải nghiệm tương tác cho cả Khách hàng (Customer), Nhân viên (Staff) và Quản trị viên (Admin). Để đảm bảo ứng dụng MAUI có cấu trúc tốt, dễ phát triển, bảo trì và kiểm thử, kiến trúc Model-View-ViewModel (MVVM) đã được lựa chọn làm mô hình thiết kế chủ đạo.

##### **a) Mô hình kiến trúc MVVM (Model-View-ViewModel)**

MVVM là một mẫu thiết kế kiến trúc giao diện người dùng phổ biến, giúp phân tách rõ ràng giữa giao diện người dùng (View), logic trình bày và trạng thái của View (ViewModel), và dữ liệu nghiệp vụ (Model).



Hình 4. Mô hình kiến trúc MVVM

#### *View (Giao diện người dùng):*

Xây dựng bằng XAML, chịu trách nhiệm hiển thị dữ liệu và giao tiếp với người dùng. Mã code-behind chỉ dùng để khởi tạo và gán BindingContext. View sử dụng cơ chế Data Binding để liên kết với thuộc tính và lệnh ( ICommand) trong ViewModel. Các View được tổ chức theo chức năng trong thư mục Views/.

#### *ViewModel (Logic trình bày):*

Là các lớp C# chứa logic trình bày và trạng thái giao diện. ViewModel không phụ thuộc trực tiếp vào UI, dễ kiểm thử. Kế thừa từ BaseViewModel (dùng ObservableObject) và sử dụng [ObservableProperty], [RelayCommand] để hỗ trợ binding và xử lý tương tác. ViewModel gọi các dịch vụ nghiệp vụ và quản lý trạng thái như IsBusy, ErrorMessage. Được tổ chức trong thư mục ViewModels.

#### *Model (Mô hình dữ liệu):*

Gồm các DTO (ví dụ: BookDto, OrderDto) đại diện dữ liệu từ API backend. ViewModel sử dụng các DTO này để binding với View, thường dưới dạng ObservableCollection<T>. Trong trường hợp cần thiết, có thể dùng Model cục bộ và ánh xạ bằng AutoMapper, nhưng chủ yếu dùng trực tiếp DTO.

### b) Các Thành phần Hỗ trợ và Cấu trúc Dự án

Ngoài ba thành phần chính của MVVM, kiến trúc ứng dụng MAUI còn bao gồm các thành phần sau:

#### *Services (Dịch vụ phía Client):*

- Các lớp dịch vụ như AuthService đóng gói logic không thuộc riêng ViewModel nào hoặc được dùng chung.
- AuthService: Quản lý trạng thái đăng nhập, lưu/truy xuất JWT từ SecureStorage, cung cấp thông tin người dùng, phát sự kiện khi thay đổi xác thực.

#### *API Clients (Refit):*

Các interface định nghĩa các API endpoint sử dụng của Refit.

- Refit tự động tạo các HTTP client tương ứng.

- Inject vào ViewModel hoặc Service để gọi API.
- Tổ chức trong Interfaces/Apis/.

#### *Handlers (Middleware):*

- AuthHeaderHandler: Tự động thêm JWT vào header Authorization cho các HTTP request.
  - Được gắn vào các Refit client cần xác thực.
  - Lưu tại Handlers/.

#### *Converters (Giá trị chuyển đổi):*

- Các lớp IValueConverter dùng trong XAML để chuyển dữ liệu giữa View-Model và View (vd: BoolToVisibilityConverter).
- Lưu tại Converters/.

#### *Helpers:*

- Chứa các hàm, hằng số dùng chung trong toàn app.
- Lưu tại Helpers/.

#### *Resources (Giao diện & Chủ đề):*

- Colors.xaml: Định nghĩa bảng màu (hỗ trợ Light/Dark).
- Styles.xaml: Định nghĩa các style UI thống nhất.
- Converters.xaml: Khai báo ResourceDictionary cho converter.
- Tổ chức tại Resources/Styles/.

#### *Dependency Injection (MauiProgram.cs):*

- Là nơi đăng ký các thành phần: Services, ViewModels, Views, Refit Clients, Handlers, v.v.
- Sử dụng AddSingleton, AddTransient, AddScoped tùy theo vòng đời.

#### *Điều hướng (AppShell):*

- AppShell.xaml: Định nghĩa cấu trúc điều hướng chính (Flyout, Tab bar).
- Dùng Shell.Current.GoToAsync("routeName") để điều hướng.
- Truyền tham số qua query string hoặc ShellNavigationQueryParameters.
- AppShellViewModel: Quản lý hiển thị menu dựa trên vai trò người dùng.

#### **Luồng tương tác một chức năng điển hình trên MAUI:**

1. Người dùng tương tác với một control trên View (ví dụ: nhấn Button).
2. Hành động này kích hoạt một Command trong ViewModel liên kết.
3. ViewModel thực thi logic:
  - Cập nhật trạng thái (IsBusy = true).
  - (Nếu cần) Validate dữ liệu đầu vào (sử dụng FluentValidator đã inject).

- Gọi một phương thức trong một Application Service (ví dụ: AuthService) hoặc trực tiếp một Refit API Client (ví dụ: IBooksApi).

4. Refit API Client (through qua AuthHeaderHandler nếu cần) gửi HTTP request đến API Backend.

5. ViewModel nhận HTTP response (đã được Refit deserialize thành DTO).

6. ViewModel xử lý kết quả:

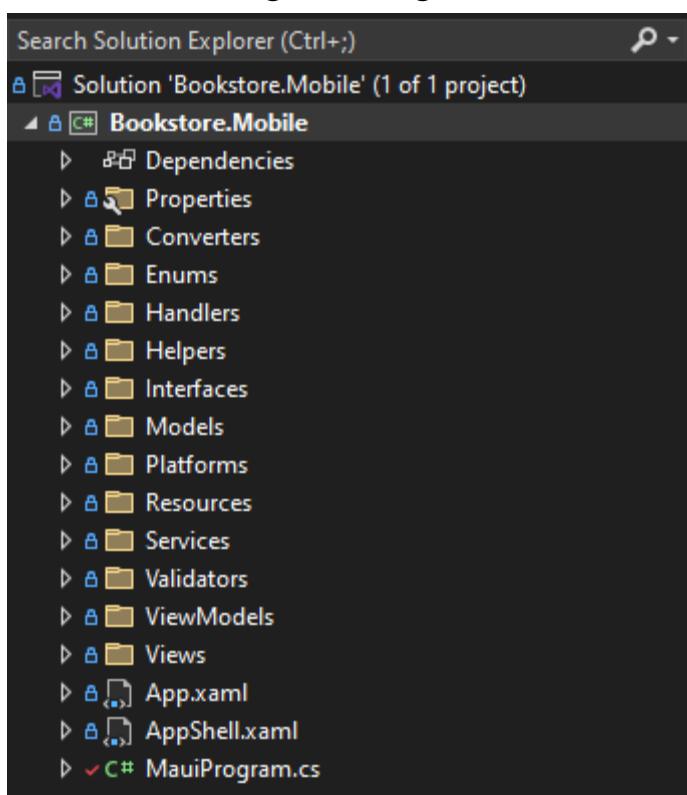
- Nếu thành công, cập nhật các ObservableProperty bằng dữ liệu DTO mới.

View sẽ tự động cập nhật nhờ Data Binding.

- Nếu thất bại, cập nhật ErrorMessage hoặc hiển thị DisplayAlertAsync.

- Cập nhật trạng thái (IsBusy = false).

- (Nếu cần) Thực hiện điều hướng đến trang khác.



Hình 5. Cấu trúc dự án MAUI

Kiến trúc MVVM kết hợp với các thành phần hỗ trợ như Refit, DI, và Shell giúp xây dựng ứng dụng .NET MAUI Quản lý Nhà sách một cách có cấu trúc, dễ mở rộng, dễ bảo trì và có khả năng kiểm thử tốt.

### 1.3. Thiết kế mô hình

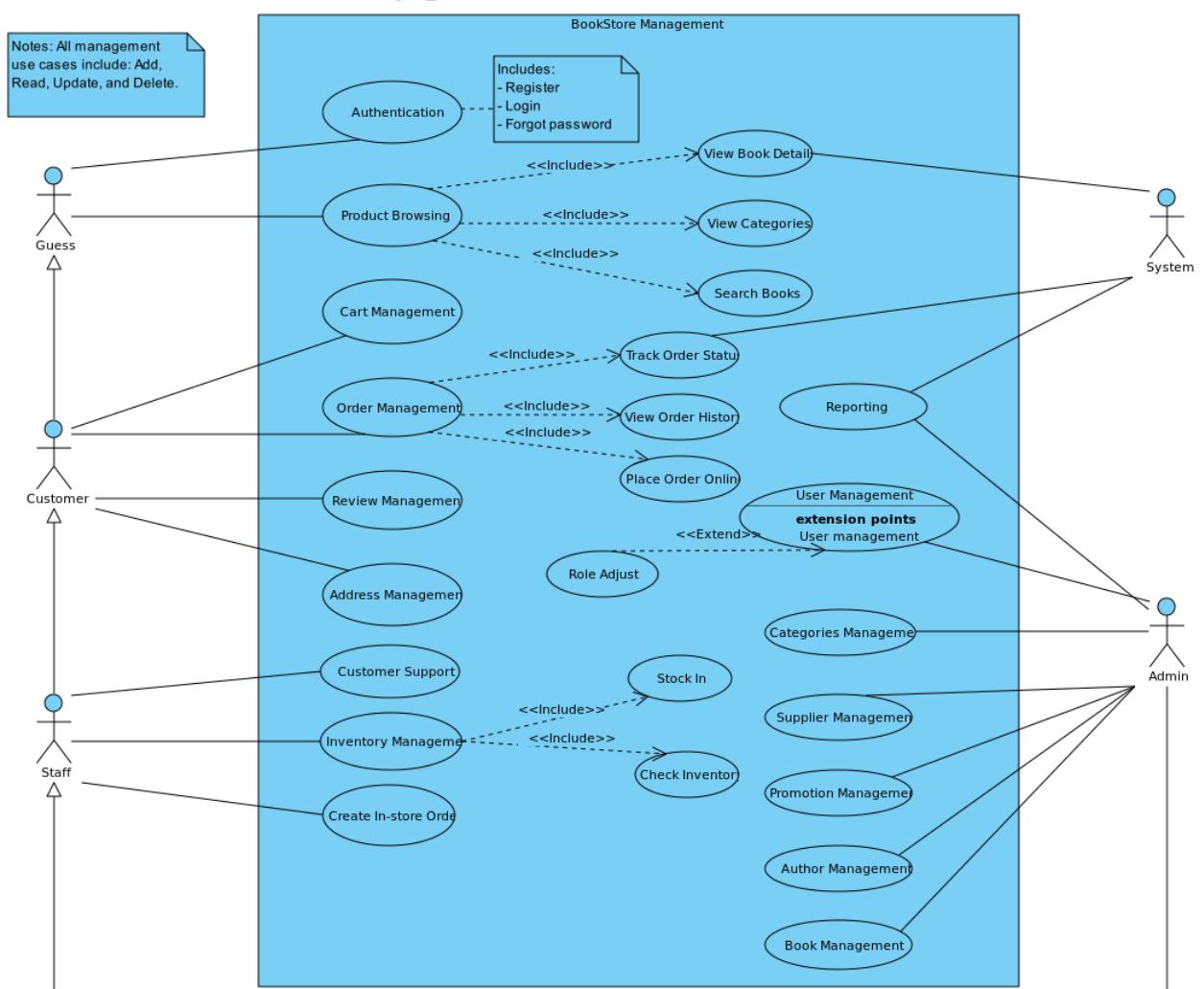
Để hỗ trợ quá trình hiện thực hóa hệ thống, các mô hình trực quan được xây dựng nhằm làm rõ cấu trúc, hành vi và tương tác giữa các thành phần. Các mô hình này đóng vai trò quan trọng trong việc truyền đạt ý tưởng thiết kế, giảm thiểu sai sót khi triển khai và đảm bảo hệ thống hoạt động đúng với yêu cầu đã phân tích.

### 1.3.1. Mô hình UseCase – UseCase Diagram

Sơ đồ Use Case (Sơ đồ Trường hợp Sử dụng) trong Ngôn ngữ Mô hình Hóa Thống nhất (UML) là một biểu diễn trực quan minh họa sự tương tác giữa người dùng (các tác nhân) và hệ thống. Nó mô tả các yêu cầu chức năng của hệ thống, thể hiện cách người dùng khác nhau tương tác với các use case (trường hợp sử dụng) – tức các chức năng cụ thể trong hệ thống.

Sơ đồ Use Case cung cấp một cái nhìn tổng quan về hành vi của hệ thống, giúp các bên liên quan (stakeholders), nhà phát triển (developers) và nhà phân tích (analysts) hiểu được cách hệ thống hoạt động từ góc nhìn người dùng, cũng như mối quan hệ giữa các quy trình khác nhau. Đây là công cụ quan trọng để xác định phạm vi và yêu cầu của hệ thống [10].

#### Sơ đồ Use Case tổng quát:



Hình 6. Sơ đồ Use Case tổng quát cho hệ thống quản lý nhà sách

#### Mô tả các Use Case chính có trong hệ thống:

Bảng 1. Use Case Đăng ký tài khoản

Use Case ID	UC-AUTH-01
-------------	------------

Use Case Name	User Registration
Description	Người dùng tạo tài khoản mới để sử dụng các chức năng của hệ thống.
Actor(s)	Customer
Priority	High
Trigger	Người dùng nhấn nút "Đăng ký" trên màn hình đăng nhập/đăng ký.
Pre-Condition(s)	Người dùng chưa có tài khoản với email/SDT này trên hệ thống.
Post-Condition	<ul style="list-style-type: none"> <li>- Tài khoản mới được tạo, thông tin lưu vào hệ thống.</li> <li>- Người dùng có thể đăng nhập bằng tài khoản vừa tạo.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>1. Người dùng truy cập màn hình đăng ký.</li> <li>2. Nhập thông tin (họ tên, email, mật khẩu, xác nhận mật khẩu, v.v.).</li> <li>3. Nhấn "Đăng ký".</li> <li>4. Hệ thống kiểm tra hợp lệ, tạo tài khoản mới, gửi email xác thực (nếu có).</li> <li>5. Thông báo đăng ký thành công, chuyển sang màn hình đăng nhập hoặc tự động đăng nhập.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>• 4a. Nếu email đã tồn tại: Hệ thống báo lỗi, yêu cầu nhập email khác.</li> <li>• 2a. Nếu nhập thiếu/thừa thông tin: Hệ thống báo lỗi, yêu cầu nhập lại.</li> </ul>

Bảng 2. Use Case Đăng nhập hệ thống

Use Case ID	UC-AUTH-02
Use Case Name	User Login
Description	Người dùng đăng nhập vào hệ thống để sử dụng các chức năng phù hợp với vai trò.
Actor(s)	Customer, Staff, Admin
Priority	High
Trigger	Người dùng nhập thông tin và nhấn nút "Đăng nhập".
Pre-Condition(s)	Người dùng đã có tài khoản hợp lệ.

Post-Condition	Người dùng được xác thực, nhận JWT token, truy cập các chức năng phù hợp với vai trò.
Basic Flow	<ol style="list-style-type: none"> <li>Người dùng truy cập màn hình đăng nhập.</li> <li>Nhập email/SDT và mật khẩu.</li> <li>Nhấn "Đăng nhập".</li> <li>Hệ thống kiểm tra thông tin, xác thực thành công, trả về token và thông tin người dùng.</li> <li>Người dùng được chuyển vào giao diện chính của ứng dụng.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>4a. Nếu thông tin không đúng: Hệ thống báo lỗi "Sai tài khoản hoặc mật khẩu".</li> <li>4b. Nếu tài khoản bị khóa: Hệ thống báo lỗi, không cho đăng nhập.</li> </ul>

Bảng 3. Use Case Đặt hàng sách online

Use Case ID	UC-CUS-01
Use Case Name	Place Online Order
Description	Khách hàng đặt mua một hoặc nhiều cuốn sách qua ứng dụng.
Actor(s)	Customer
Priority	High
Trigger	Khách hàng nhấn nút "Đặt hàng" sau khi chọn sách và nhập thông tin giao hàng.
Pre-Condition(s)	<ul style="list-style-type: none"> <li>Khách hàng đã đăng nhập.</li> <li>Giỏ hàng có ít nhất một sản phẩm.</li> <li>Đã nhập địa chỉ giao hàng hợp lệ.</li> </ul>
Post-Condition	<ul style="list-style-type: none"> <li>Đơn hàng được tạo, lưu vào hệ thống.</li> <li>Tồn kho sách được cập nhật.</li> <li>Khách hàng nhận được xác nhận đặt hàng.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>Khách hàng chọn sách, thêm vào giỏ hàng.</li> <li>Khách hàng vào giỏ hàng, kiểm tra lại sản phẩm.</li> <li>Khách hàng nhập/chọn địa chỉ giao hàng.</li> <li>Khách hàng nhấn "Đặt hàng".</li> </ol>

	<p>5. Hệ thống kiểm tra tồn kho, tạo đơn hàng, giảm tồn kho.</p> <p>6. Hệ thống gửi xác nhận đơn hàng cho khách hàng.</p>
Alternative Flow	<ul style="list-style-type: none"> <li>5a. Nếu tồn kho không đủ: Hệ thống thông báo lỗi, không tạo đơn hàng.</li> <li>3a. Nếu địa chỉ không hợp lệ: Hệ thống yêu cầu nhập lại.</li> </ul>

Bảng 4. Use Case Đánh giá sản phẩm

Use Case ID	UC-CUS-02
Use Case Name	Review Product
Description	Khách hàng đánh giá sách đã mua.
Actor(s)	Customer
Priority	Medium
Trigger	Khách hàng nhấn "Đánh giá" trên trang chi tiết đơn hàng đã nhận.
Pre-Condition(s)	<ul style="list-style-type: none"> <li>Khách hàng đã đăng nhập.</li> <li>Đơn hàng đã hoàn thành (đã nhận sách).</li> </ul>
Post-Condition	Đánh giá được lưu vào hệ thống, hiển thị cho các khách hàng khác.
Basic Flow	<ol style="list-style-type: none"> <li>Khách hàng vào lịch sử đơn hàng, chọn đơn đã nhận.</li> <li>Chọn sách muốn đánh giá, nhập nội dung, số sao.</li> <li>Nhấn "Gửi đánh giá".</li> <li>Hệ thống lưu đánh giá, cập nhật điểm đánh giá sản phẩm.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>2a. Nếu khách hàng chưa nhập đủ thông tin: Hệ thống yêu cầu nhập lại.</li> </ul>

Bảng 5. Use Case Tạo đơn hàng tại quầy

Use Case ID	UC-STF-01
Use Case Name	Create POS Order
Description	Nhân viên tạo đơn hàng cho khách tại cửa hàng.
Actor(s)	Staff
Priority	High

Trigger	Khách hàng đến mua sách trực tiếp tại quầy.
Pre-Condition(s)	<ul style="list-style-type: none"> <li>Nhân viên đã đăng nhập hệ thống.</li> <li>Sách còn tồn kho.</li> </ul>
Post-Condition	<ul style="list-style-type: none"> <li>Đơn hàng được tạo, lưu vào hệ thống.</li> <li>Tồn kho sách được cập nhật.</li> </ul>
Basic Flow	<ol style="list-style-type: none"> <li>Nhân viên tìm kiếm, chọn sách cho khách.</li> <li>Nhập thông tin khách hàng (nếu cần).</li> <li>Xác nhận đơn hàng, nhận thanh toán.</li> <li>Hệ thống tạo đơn hàng, giảm tồn kho.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>1a. Nếu sách hết hàng: Thông báo cho khách, không tạo đơn hàng.</li> </ul>

Bảng 6. Use Case Nhập kho sách

Use Case ID	UC-STF-02
Use Case Name	Stock Receipt
Description	Nhân viên nhập sách mới vào kho từ nhà cung cấp.
Actor(s)	Staff
Priority	Medium
Trigger	Có sách mới nhập về từ nhà cung cấp.
Pre-Condition(s)	<ul style="list-style-type: none"> <li>Nhân viên đã đăng nhập.</li> <li>Có thông tin nhà cung cấp, sách nhập kho.</li> </ul>
Post-Condition	Phiếu nhập kho được tạo, tồn kho sách được cập nhật.
Basic Flow	<ol style="list-style-type: none"> <li>Nhân viên vào chức năng nhập kho.</li> <li>Chọn nhà cung cấp, nhập danh sách sách và số lượng.</li> <li>Xác nhận nhập kho.</li> <li>Hệ thống tạo phiếu nhập, cập nhật tồn kho.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>2a. Nếu thông tin không hợp lệ: Hệ thống yêu cầu nhập lại.</li> </ul>

Bảng 7. Use Case Quản lý sản phẩm

Use Case ID	UC-ADM-01
Use Case Name	Manage Books
Description	Admin thêm, sửa, xóa, cập nhật thông tin sách.
Actor(s)	Admin

Priority	High
Trigger	Có yêu cầu cập nhật danh mục sách.
Pre-Condition(s)	Admin đã đăng nhập.
Post-Condition	Thông tin sách được cập nhật trong hệ thống.
Basic Flow	<ol style="list-style-type: none"> <li>1. Admin vào chức năng quản lý sách.</li> <li>2. Thêm mới/sửa/xóa sách, nhập thông tin chi tiết.</li> <li>3. Lưu thay đổi.</li> <li>4. Hệ thống cập nhật dữ liệu, đồng bộ với kho, hiển thị cho khách hàng.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>• 2a. Nếu nhập thiếu/thừa thông tin: Hệ thống báo lỗi, yêu cầu nhập lại.</li> </ul>

Bảng 8. Use Case Xem báo cáo doanh thu

Use Case ID	UC-ADM-02
Use Case Name	View Sales Report
Description	Admin xem báo cáo doanh thu, đơn hàng, tồn kho.
Actor(s)	Admin
Priority	Medium
Trigger	Admin truy cập chức năng báo cáo.
Pre-Condition(s)	Admin đã đăng nhập.
Post-Condition	Admin truy cập chức năng báo cáo.
Basic Flow	<ol style="list-style-type: none"> <li>1. Admin chọn loại báo cáo (doanh thu, đơn hàng, tồn kho...).</li> <li>2. Chọn khoảng thời gian, bộ lọc.</li> <li>3. Hệ thống tổng hợp, hiển thị báo cáo.</li> </ol>
Alternative Flow	<ul style="list-style-type: none"> <li>• 3a. Nếu không có dữ liệu: Hệ thống thông báo "Không có dữ liệu".</li> </ul>

### 1.3.2. Mô hình hoạt động - Activity Diagram

Activity Diagram (biểu đồ hoạt động) là một phần quan trọng của Ngôn ngữ Mô hình Hóa Thống nhất (UML), giúp hình dung các quy trình công việc, quy trình hoặc hoạt động trong một hệ thống. Chúng mô tả cách các hành động khác nhau được kết nối và cách hệ thống chuyển từ trạng thái này sang trạng thái khác. Bằng cách cung cấp một bức tranh rõ ràng về cả các quy trình đơn giản và phức tạp, biểu

đồ hoạt động giúp các nhà phát triển và các bên liên quan dễ dàng hiểu cách các yếu tố khác nhau tương tác trong hệ thống [11].

Các Activity Diagram chi tiết được mô tả trong Phụ lục C: Sơ đồ hoạt động - Activity Diagram

#### *1.3.3. Mô hình tuần tự - Sequence Diagram*

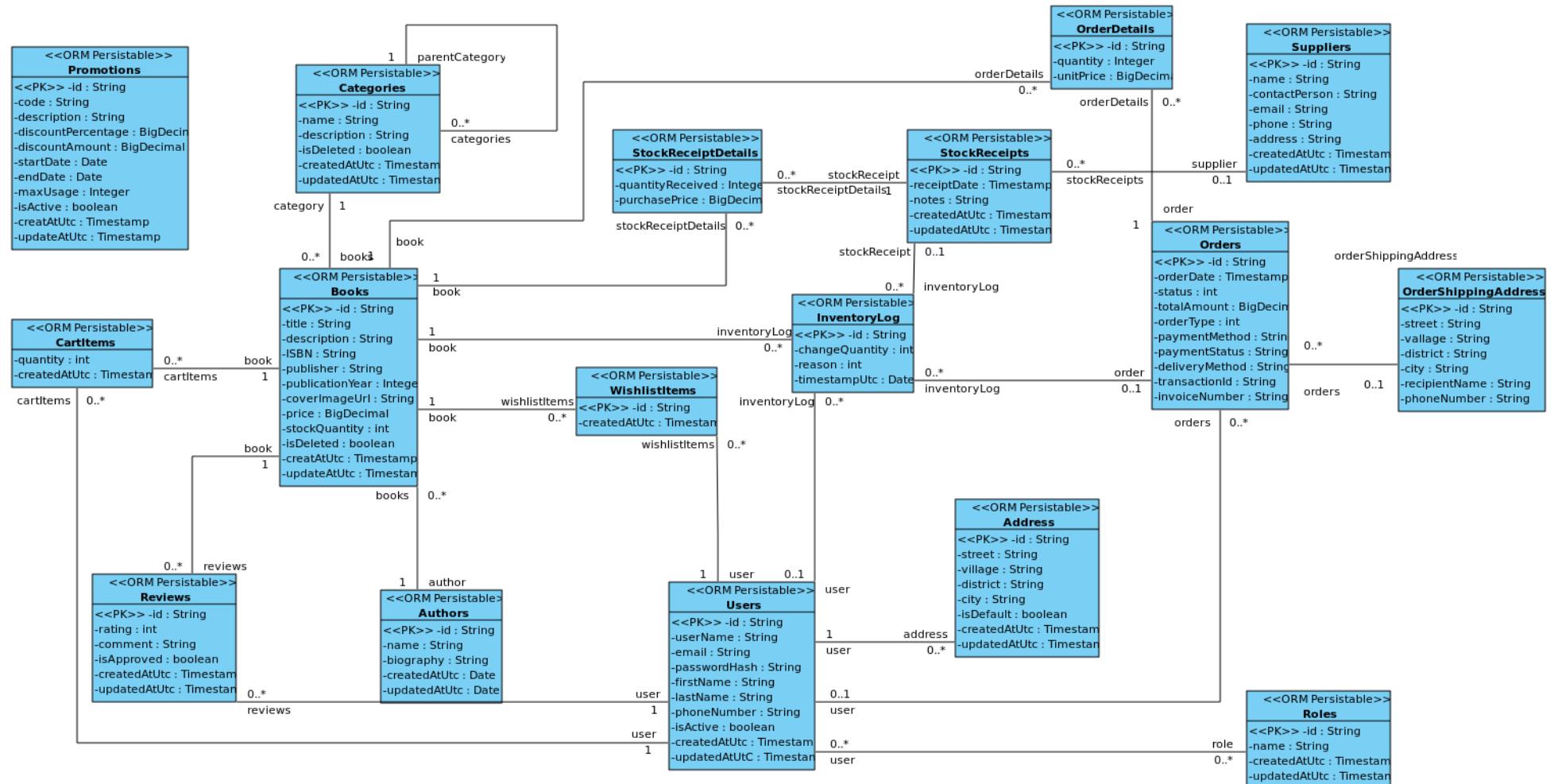
Sơ đồ tuần tự (sequence diagram) là một loại sơ đồ trong Ngôn ngữ Mô hình Hóa Thống nhất (UML) minh họa trình tự các thông điệp trao đổi giữa các đối tượng trong một tương tác. Sơ đồ này bao gồm một nhóm các đối tượng được biểu diễn bằng các "lifelines" (đường đời) cùng với các thông điệp mà chúng trao đổi theo thời gian trong quá trình tương tác. Sơ đồ tuần tự thể hiện trình tự các thông điệp được truyền giữa các đối tượng. Nó cũng có thể mô tả các cấu trúc điều khiển giữa các đối tượng [12].

Các mô hình tuần tự tiêu biểu được mô tả trong Phụ lục D: Sơ đồ tuần tự - Sequence Diagram

#### *1.3.4. Mô hình lớp - Class Diagram*

Biểu đồ lớp (Class Diagram) UML là một công cụ trực quan biểu diễn cấu trúc của một hệ thống bằng cách hiển thị các lớp, thuộc tính, phương thức và mối quan hệ giữa chúng. Nó giúp mọi người tham gia vào dự án—như nhà phát triển và nhà thiết kế—hiểu được cách hệ thống được tổ chức và cách các thành phần tương tác với nhau.

Dưới đây là Class Diagram chi tiết của ứng dụng:



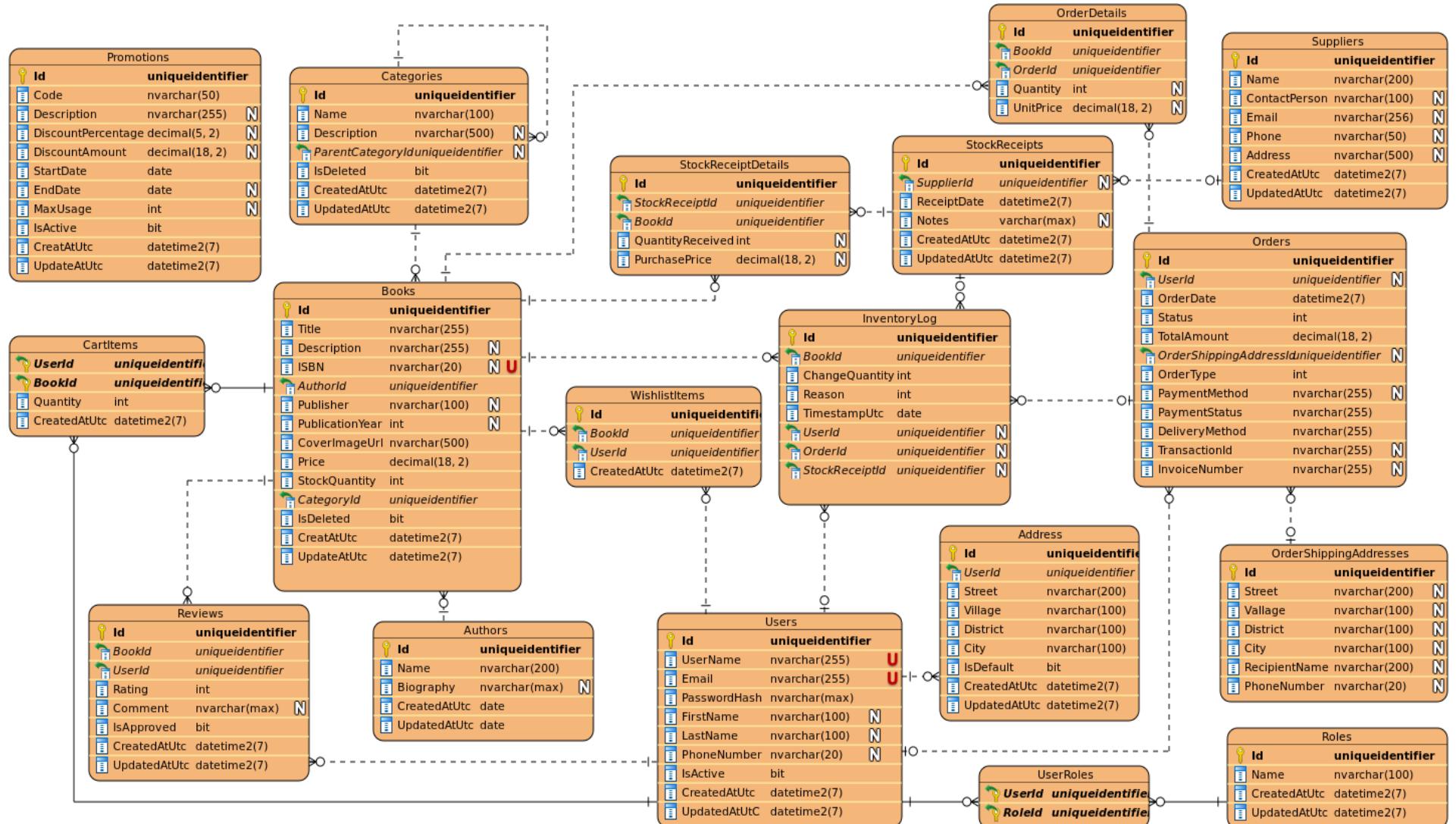
Hình 7. Mô hình lớp – Class Diagram tổng quát

## **1.4. Thiết kế cơ sở dữ liệu**

Phần này trình bày quá trình thiết kế cơ sở dữ liệu cho hệ thống quản lý nhà sách, nhằm đảm bảo khả năng lưu trữ, truy xuất và xử lý dữ liệu một cách hiệu quả và nhất quán. Dựa trên các yêu cầu đã phân tích ở các mục trước, mô hình dữ liệu được xây dựng theo hướng quan hệ (Relational Model), đảm bảo tính toàn vẹn và phù hợp với hệ quản trị cơ sở dữ liệu SQL Server được sử dụng trong dự án. Mô hình quan hệ thể hiện rõ các thực thể chính, thuộc tính đặc trưng và mối quan hệ giữa chúng, làm cơ sở để xây dựng cấu trúc bảng trong cơ sở dữ liệu thực tế.

### *1.4.1. Mô hình dữ liệu quan hệ ERD*

Dưới đây là mô hình dữ liệu quan hệ ERD chi tiết:



Hình 8. Mô hình dữ liệu quan hệ ERD

#### 1.4.2. Mô tả mô hình quan hệ

Bảng 9. Roles Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	Name	NVARCHAR(50)	X	X	
3	CreatedAtUtc	DATETIME2(7)	X		
4	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 10. Users Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	UserName	NVARCHAR(256)	X	X	
3	Email	NVARCHAR(256)	X	X	
4	PasswordHash	NVARCHAR(max)	X		
5	FirstName	NVARCHAR(100)			
6	LastName	NVARCHAR(100)			
7	PhoneNumber	NVARCHAR(20)			
8	IsActive	BIT	X		
9	CreatedAtUtc	DATETIME2(7)	X		
10	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 11. Categories Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	Name	NVARCHAR(100)	X		

3	Description	NVARCHAR(500)			
4	ParentCategory	Category	UNIQUEIDENTIFIER		FK
5	IsDeleted	BIT	X		
6	CreatedAtUtc	DATETIME2(7)	X		
7	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 12. Authors Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	Name	NVARCHAR(200)	X		
3	Biography	NVARCHAR(max)			
4	CreatedAtUtc	DATETIME2(7)	X		
5	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 13. Books Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	Title	NVARCHAR(256)	X		
3	Description	NVARCHAR(max)			
4	ISBN	NVARCHAR(20)			
5	AuthorId	UNIQUEIDENTIFIER			FK
6	Publisher	NVARCHAR(100)			
7	PublicationYear	INT			
8	CoverImageUrl	NVARCHAR(max)			

9	Price	DECIMAL(18, 2)	X		
10	StockQuantity	INT	X		
11	CategoryId	UNIQUEIDENTIFIER	X		FK
12	IsDeleted	BIT	X		
13	CreatedAtUtc	DATETIME2(7)	X		
14	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 14. Addresses Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	UserId	UNIQUEIDENTIFIER	X		FK
3	Street	NVARCHAR(200)	X		
4	Village	NVARCHAR(100)	X		
5	District	NVARCHAR(100)	X		
6	City	NVARCHAR(100)	X		
7	IsDefault	BIT	X		
8	CreatedAtUtc	DATETIME2(7)	X		
9	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 15. CartItems Table

#	Field name	Type	Not Null	Unique	PK/FK
1	UserId	UNIQUEIDENTIFIER	X		FK
2	BookId	UNIQUEIDENTIFIER	X		FK
3	Quantity	INT	X		
4	CreatedAtUtc	DATETIME2(7)	X		

Bảng 16. WishlistItems Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	UserId	UNIQUEIDENTIFIER	X		FK
3	BookId	UNIQUEIDENTIFIER	X		FK
4	CreatedAtUtc	DATETIME2(7)	X		

Bảng 17. Orders Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	UserId	UNIQUEIDENTIFIER			FK
3	OrderDate	DATETIME2(7)	X		
4	Status	TINYINT	X		
5	TotalAmount	DECIMAL(18, 2)	X		
6	OrderShippingAddressId	UNIQUEIDENTIFIER			
7	OrderType	INT	X		
8	PaymentMethod	NVARCHAR(50)			
9	PaymentStatus	NVARCHAR(50)	X		
10	DeliveryMothod	NVARCHAR(50)	X		
11	TransactionId	NVARCHAR(256)			
12	InvoiceNumber	NVARCHAR(50)			

Bảng 18. OrderDetails Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	OrderId	UNIQUEIDENTIFIER	X		FK

3	BookId	UNIQUEIDENTIFIER	X		FK
4	Quantity	INT			
5	UnitPrice	DECIMAL(18, 2)			

Bảng 19. Promotions Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	Code	NVARCHAR(50)	X	X	
3	Description	NVARCHAR(256)			
4	DiscountPercentage	DECIMAL(5, 2)			
5	DiscountAmount	DECIMAL(18, 2)			
6	StartDate	DATETIME2(7)	X		
7	EndDate	DATETIME2(7)			
8	MaxUsage	INT			
9	CurrentUsage	INT	X		
10	IsActive	BIT	X		
11	CreatedAtUtc	DATETIME2(7)	X		
12	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 20. Reviews Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	BookId	UNIQUEIDENTIFIER	X		FK
3	UserId	UNIQUEIDENTIFIER	X		FK
4	Rating	TINYINT	X		
5	Comment	NVARCHAR(max)			
6	IsApproved	BIT	X		
7	CreatedAtUtc	DATETIME2(7)	X		
8	UpdatedAtUtc	DATETIME2(7)	X		

Bảng 21. InventoryLogs Table

#	Field name	Type	Not Null	Unique	PK/FK
1	Id	UNIQUEIDENTIFIER	X		PK
2	BookId	UNIQUEIDENTIFIER	X		FK
3	ChangeQuantity	INT	X		
4	Reason	TINYINT	X		
5	TimestampUtc	DATETIME2(7)	X		
6	OrderId	UNIQUEIDENTIFIER			FK
7	UserId	UNIQUEIDENTIFIER			FK

### 1.5. Kết luận chương

Trong chương này, các công việc chuẩn bị nền tảng cho quá trình xây dựng ứng dụng đã được tiến hành một cách hệ thống. Trước tiên, nội dung khảo sát và phân tích yêu cầu đã giúp xác định rõ các chức năng cần thiết, đối tượng sử dụng, cũng như các ràng buộc kỹ thuật và nghiệp vụ liên quan. Tiếp theo, kiến trúc tổng thể của hệ thống đã được thiết kế dựa trên mô hình client-server, kết hợp giữa .NET MAUI và ASP.NET Core Web API, đảm bảo khả năng mở rộng, tái sử dụng và dễ bảo trì. Cuối cùng, cơ sở dữ liệu đã được thiết kế chi tiết nhằm phản ánh đầy đủ mối quan hệ giữa các thực thể và phục vụ hiệu quả cho việc lưu trữ, truy vấn dữ liệu trong toàn hệ thống.

Các kết quả đạt được ở chương này là cơ sở quan trọng cho việc triển khai và hiện thực hóa hệ thống sẽ được trình bày trong các chương tiếp theo.

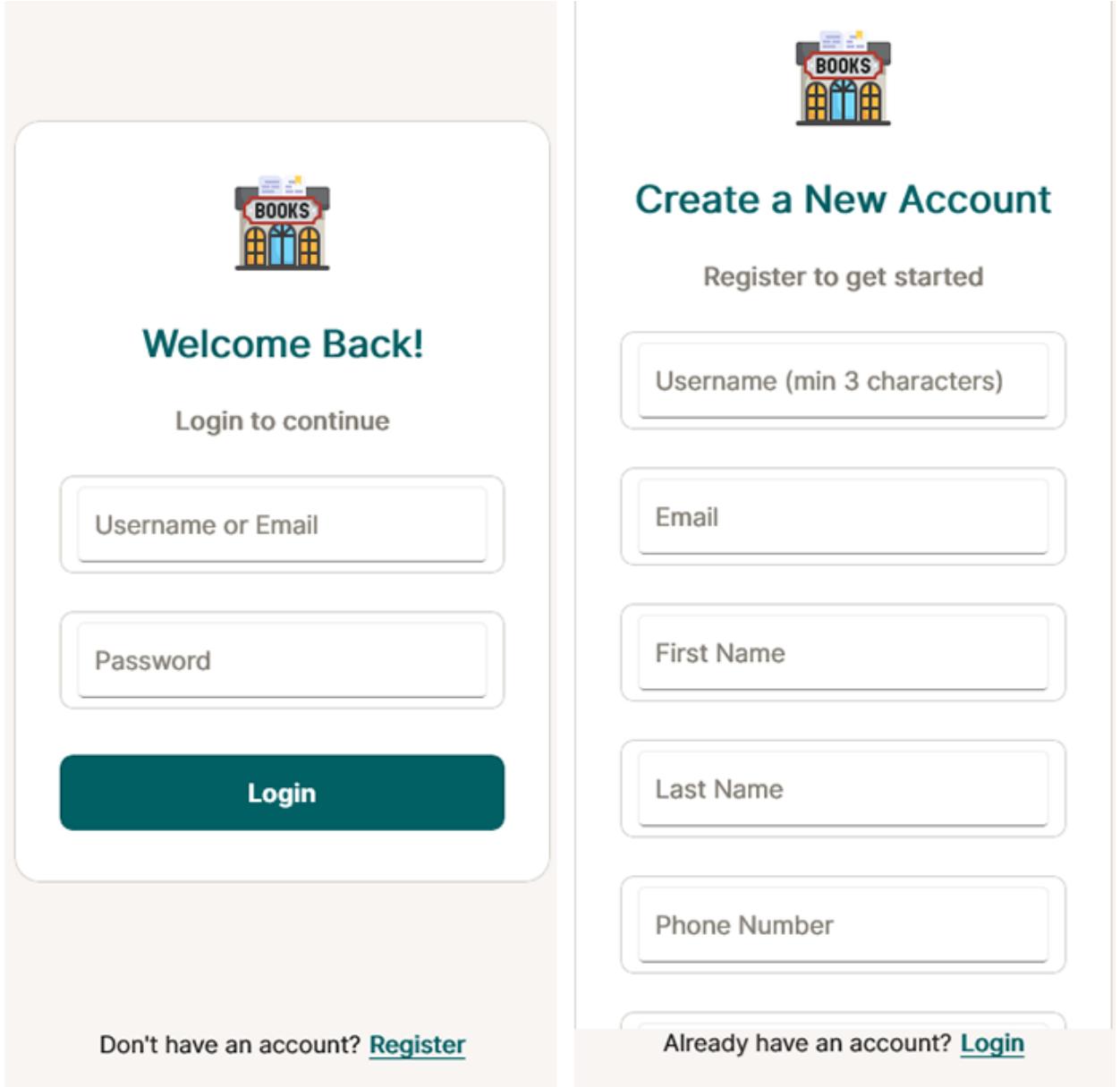
## CHƯƠNG 2. TRIỂN KHAI HỆ THỐNG

### 2.1. Kết quả thực nghiệm

Phần này trình bày kết quả sau quá trình triển khai và kiểm thử hệ thống ứng dụng quản lý nhà sách. Các giao diện và tính năng được minh họa cụ thể thông qua hình ảnh và mô tả, làm rõ tính đúng đắn, tính khả dụng và mức độ hoàn thiện của hệ thống. Việc thực nghiệm nhằm đánh giá khả năng vận hành thực tế, mức độ thân thiện với người dùng, và sự tương thích với yêu cầu chức năng đã đề ra.

#### 2.1.1. Giao diện khởi động và xác thực

Bao gồm màn hình splash screen với biểu tượng ứng dụng, cùng với giao diện đăng nhập và đăng ký, cho phép người dùng truy cập vào hệ thống với tài khoản cá nhân.



Hình 9. Giao diện khởi động và xác thực

#### 2.1.2. *Chức năng duyệt sản phẩm*

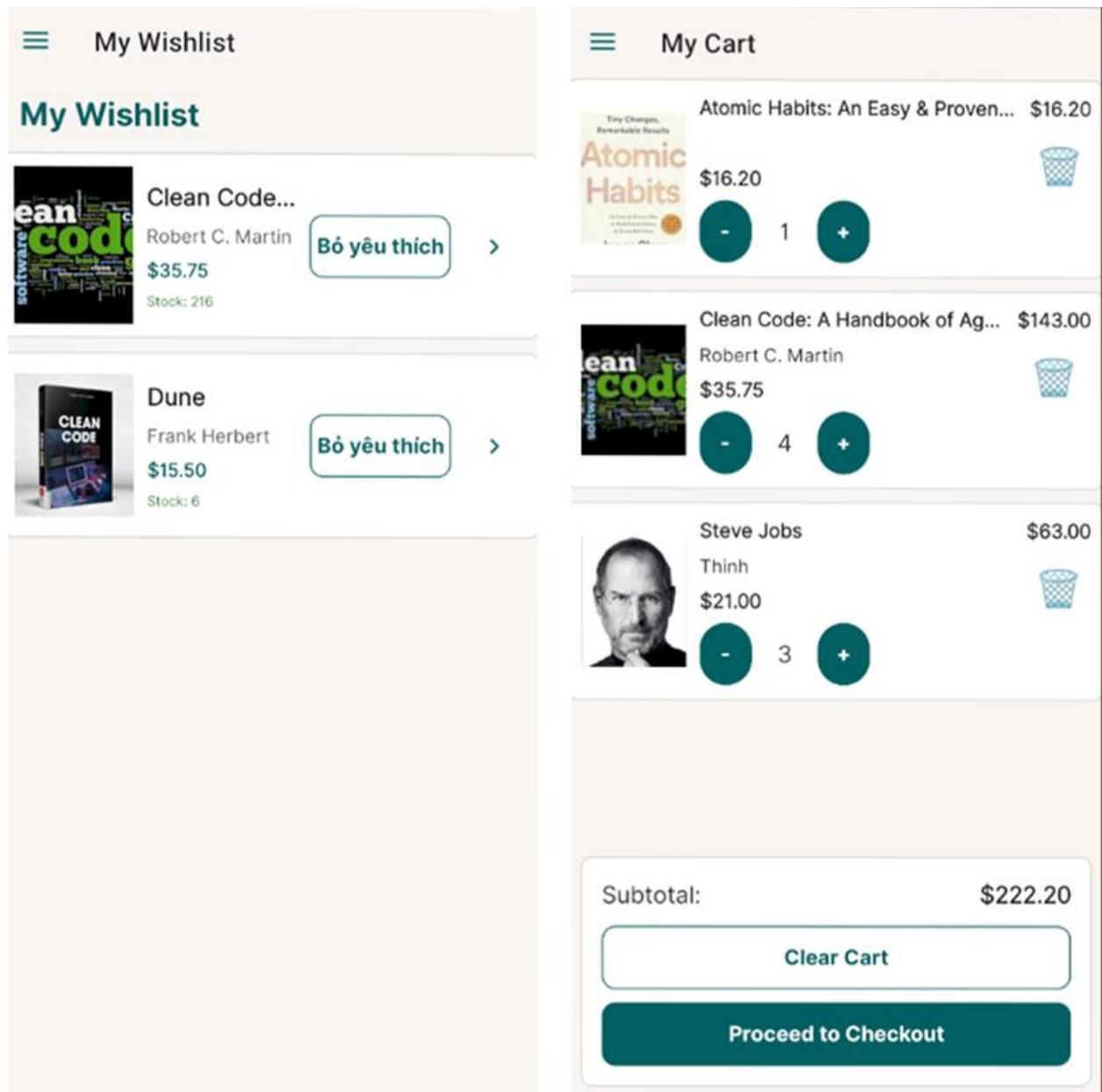
Mô tả và minh họa các chức năng chính mà khách hàng có thể sử dụng, từ việc tìm kiếm sách, quản lý thông tin cá nhân đến quá trình đặt hàng và theo dõi đơn hàng.

The figure consists of three screenshots of a mobile application interface for a bookstore. The first screenshot shows the main home screen with a search bar, a welcome message 'Welcome, Admin!', and sections for 'Newest Arrivals' and 'Bestsellers'. The second screenshot shows the 'Bestsellers' section with cards for 'Clean Code' and 'The Notebook'. The third screenshot shows a detailed product page for 'Clean Code: A Handbook of Agile Software' by Robert C. Martin, displaying the book cover, price (\$35.75), stock count (221 copies), quantity selector, and an 'Add to Cart' button.

Hình 10. Giao diện duyệt sản phẩm

### 2.1.3. Chức năng tương tác cá nhân

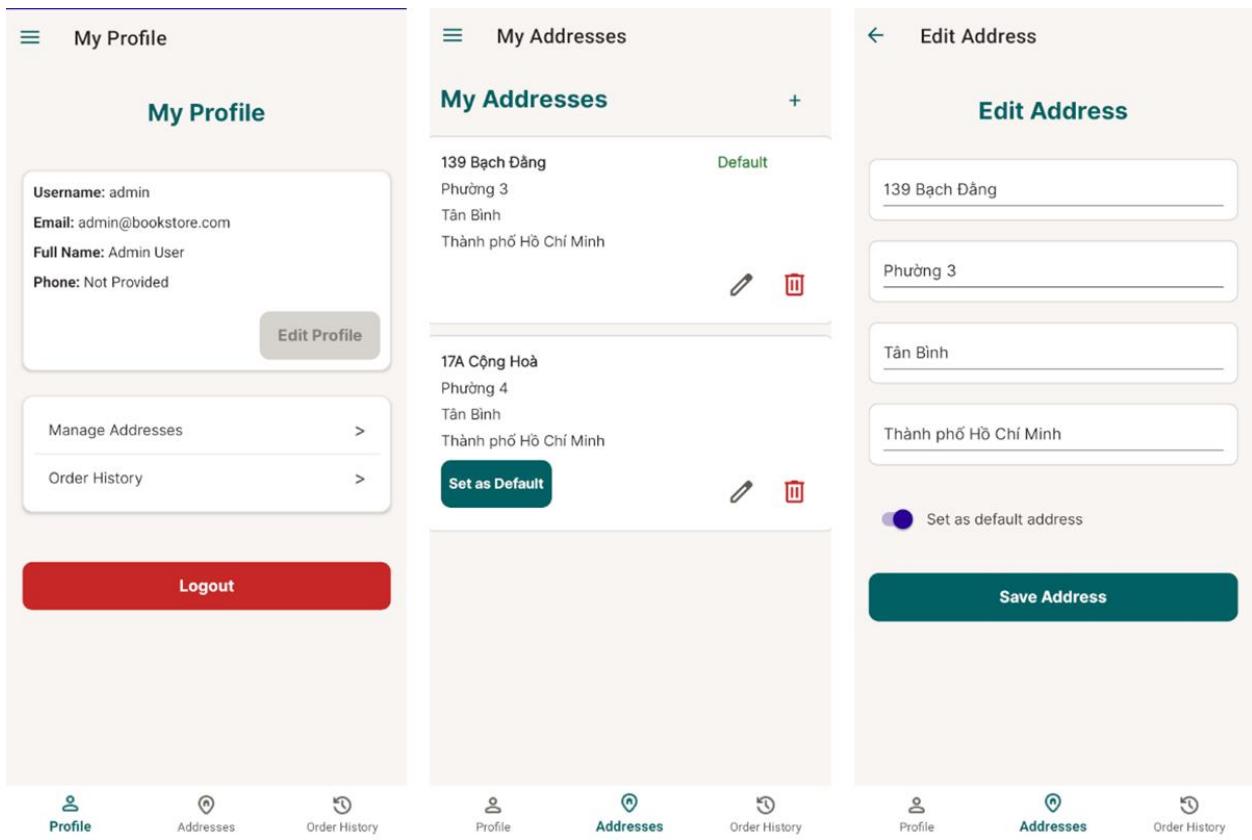
Bao gồm các giao diện cho chức năng như thêm sách vào danh sách yêu thích hoặc giỏ hàng, cho phép người dùng quản lý sở thích và các lựa chọn mua sắm một cách tiện lợi.



Hình 11. Giao diện tương tác cá nhân

#### 2.1.4. Chức năng quản lý tài khoản

Hiển thị thông tin cá nhân và cho phép người dùng cập nhật hồ sơ, quản lý danh sách địa chỉ nhận hàng một cách linh hoạt.



Hình 12. Giao diện quản lý tài khoản

### 2.1.5. *Chức năng cho luồng đặt hàng Online*

Cung cấp giao diện xem lại đơn hàng đã đặt, kiểm tra chi tiết từng đơn và gửi đánh giá, hỗ trợ khách hàng theo dõi lịch sử mua sắm và phản hồi dịch vụ, minh họa quy trình đặt hàng trực tuyến.

The screenshot illustrates the online ordering process across four main sections:

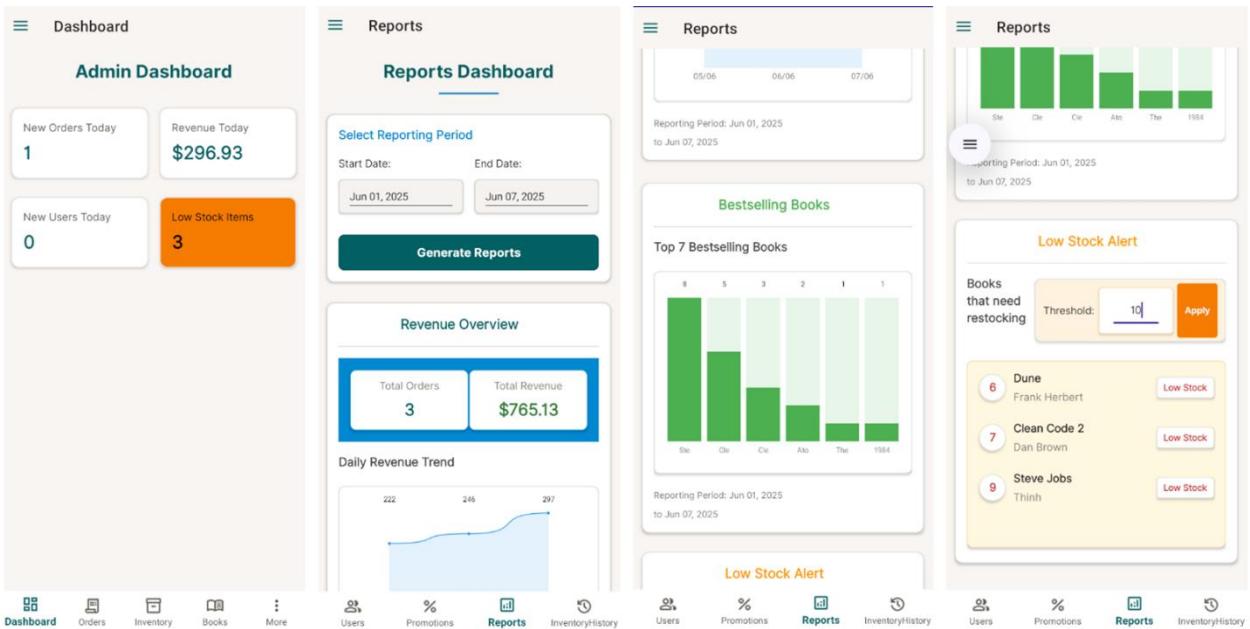
- Checkout**: Allows selecting a shipping address from a list (139 Bạch Đằng, Phường 3, Tân Bình, Thành phố Hồ Chí Minh (Default); 17A Cộng Hòa, Phường 4, Tân Bình, Thành phố Hồ Chí Minh) and adding a new address.
- Order History**: Displays a list of orders:
  - Order #F180D3EE (Confirmed): Jun 07, 2025, 1 item(s) - Total: \$123.00
  - Order #249117A7 (Completed): Jun 07, 2025, 10 item(s) - Total: \$296.93
  - Order #7B6081B3 (Completed): Jun 05, 2025, 8 item(s) - Total: \$222.20
  - Order #92CD4CA6 (Completed): May 23, 2025, 3 item(s) - Total: \$46.50
  - Order #8185832D (Pending): May 21, 2025, 5 item(s) - Total: \$105.00
  - Order #AE874A3E (Completed): May 21, 2025, 1 item(s) - Total: \$15.50
  - Order #FB5906B2 (Completed): May 15, 2025, 1 item(s) - Total: \$200,000.00
- Order #F180D3EE**: Provides detailed information for a specific order:
  - Order ID: F180D3EE
  - Order Date: 07 Jun 2025 14:27
  - Status: Processing
  - Invoice #: INV-20250607-F180
  - Payment: Completed
 Includes a shipping address section and a list of items.
- Review 'Clean Code 2'**: A review page where users can rate and leave comments for the product 'Clean Code 2' by Dan Brown. The review 'Really Good!!!' is shown.

At the bottom, there are navigation icons for Profile, Addresses, and Order History.

Hình 13. Giao diện luồng đặt hàng Online

## 2.1.6. Chức năng Admin DashBoard và báo cáo

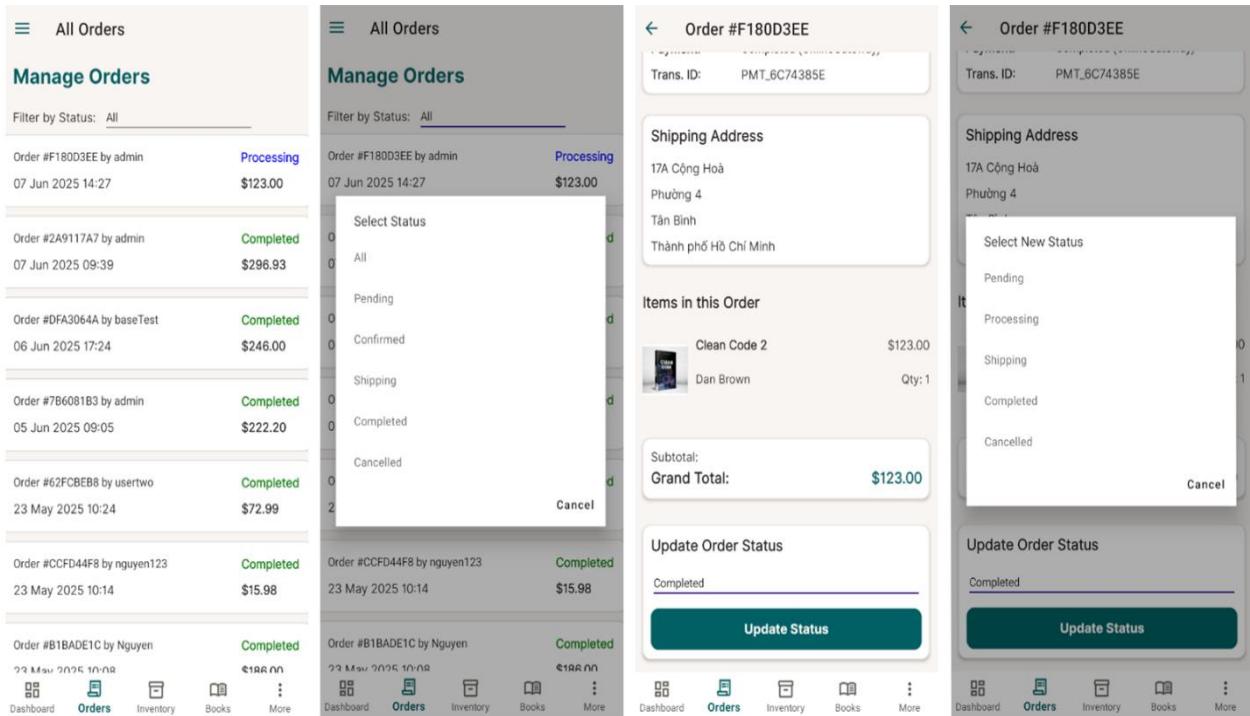
Hiển thị tổng quan hệ thống và các báo cáo chuyên sâu về doanh thu, sản phẩm bán chạy và tồn kho, hỗ trợ ra quyết định nhanh chóng và chính xác.



Hình 14. Giao diện Admin DashBoard và báo cáo

## 2.1.7. Chức năng quản lý đơn hàng

Cung cấp công cụ cho phép quản trị viên theo dõi và cập nhật trạng thái đơn hàng, đảm bảo quy trình xử lý đơn hàng rõ ràng và hiệu quả.



Hình 15. Giao diện quản lý đơn hàng

### 2.1.8. Chức năng quản lý sản phẩm

Hỗ trợ quản trị viên quản lý thông tin sách, danh mục và tác giả, bao gồm tìm kiếm, lọc, cập nhật nội dung và hình ảnh sản phẩm.

The screenshot displays four panels of a web-based application for managing books:

- Manage Books:** Shows a list of books with columns for title, author, price, and stock level. Actions like 'Edit' and 'Delete' are available for each item.
- Manage Authors:** Shows a list of authors with columns for name, description, and actions.
- Manage Categories:** Shows a list of categories with columns for name and actions.
- Edit Book:** A detailed view for a book titled 'Clean Code 2'. It includes fields for 'Cover Image' (with a preview of the book cover), 'Title' ('Clean Code 2'), 'Description' ('Clean struture code'), 'ISBN' ('0362500213'), and a 'Tags' section ('Khoa Hoc').

Hình 16. Giao diện chức năng quản lý sản phẩm

### 2.1.9. Chức năng quản lý kho hàng

Bao gồm các chức năng lập phiếu nhập kho, điều chỉnh tồn kho và theo dõi lịch sử nhập/xuất kho, giúp kiểm soát hàng hóa một cách chính xác.

The screenshot displays three panels of an inventory management application:

- Stock Receipts:** Shows a list of receipts with details like receipt number, date, supplier, and description.
- Inventory Adjustment:** A form for adjusting inventory levels. It includes a search bar for 'Select Book', a dropdown for 'Adjustment' (e.g., '+100'), and a note about a donation of 100 'clean code 2' books.
- Inventory History:** A list of historical transactions showing changes in stock levels. It includes columns for book title, reason, date, and quantity.

Hình 17. Giao diện chức năng quản lý kho hàng

### 2.1.10. Các chức năng quản lý khác

Trình bày các màn hình quản lý nhà cung cấp, khuyến mãi và người dùng hệ thống, giúp hoàn thiện quy trình vận hành và quản trị toàn diện.

The screenshot displays the application's main dashboard with four primary sections:

- Manage Suppliers**: Shows a list of suppliers with columns for Name, Contact, and Actions (Edit, Delete). Suppliers listed include Hoang Nguyen, Dang Thinh Company, Beta Books Distribution, and Nha Cung Cấp Alpha.
- Manage Promotions**: Shows a list of promotions with columns for Name, Description, Discount, Valid Until, and Usage Status (Active: True or False, Used: 0-5). Promotions listed include HEYEUTHUONG, XUAN25, LIMIT10, and SUMMER20.
- Manage Users**: Shows a list of users with columns for Username, Email, Role, and Status (Active or Inactive). It includes an active admin user and several inactive test accounts like baseTest, nguyen, and postman\_user.
- User Details**: A detailed view of a specific user account, showing User Information (User ID: f54527eb-f806-40db-bf76-c7b0e5fa6d39, Username: admin, Email: admin@bookstore.com), Account Status (Active), and a Save Status Change button.

At the bottom, there is a navigation bar with icons for Dashboard, Orders, Inventory, Books, More, Users, Promotions, Reports, and InventoryHistory.

Hình 18. Giao diện các chức năng quản lý khác

## 2.2. Đánh giá hiệu năng

Để đánh giá hiệu năng các API trong hệ thống, công cụ K6 (Javascript) đã được sử dụng với cấu hình 16 luồng đồng thời trong thời gian 30 giây. Kết quả cho thấy tất cả các API đều xử lý 100% yêu cầu thành công, không có lỗi xảy ra trong quá trình kiểm thử.

Nhìn chung, hệ thống phản hồi nhanh, ổn định và đáp ứng tốt với tải đồng thời từ người dùng. Các API chính phục vụ thao tác khách hàng và quản lý đều đạt tiêu chuẩn về độ tin cậy và thời gian phản hồi.

Dưới đây là hình ảnh thông số cụ thể khi chạy kiểm tra một số API nổi bật:

### EVALUATION RESULT OF APIs PERFORMANCE

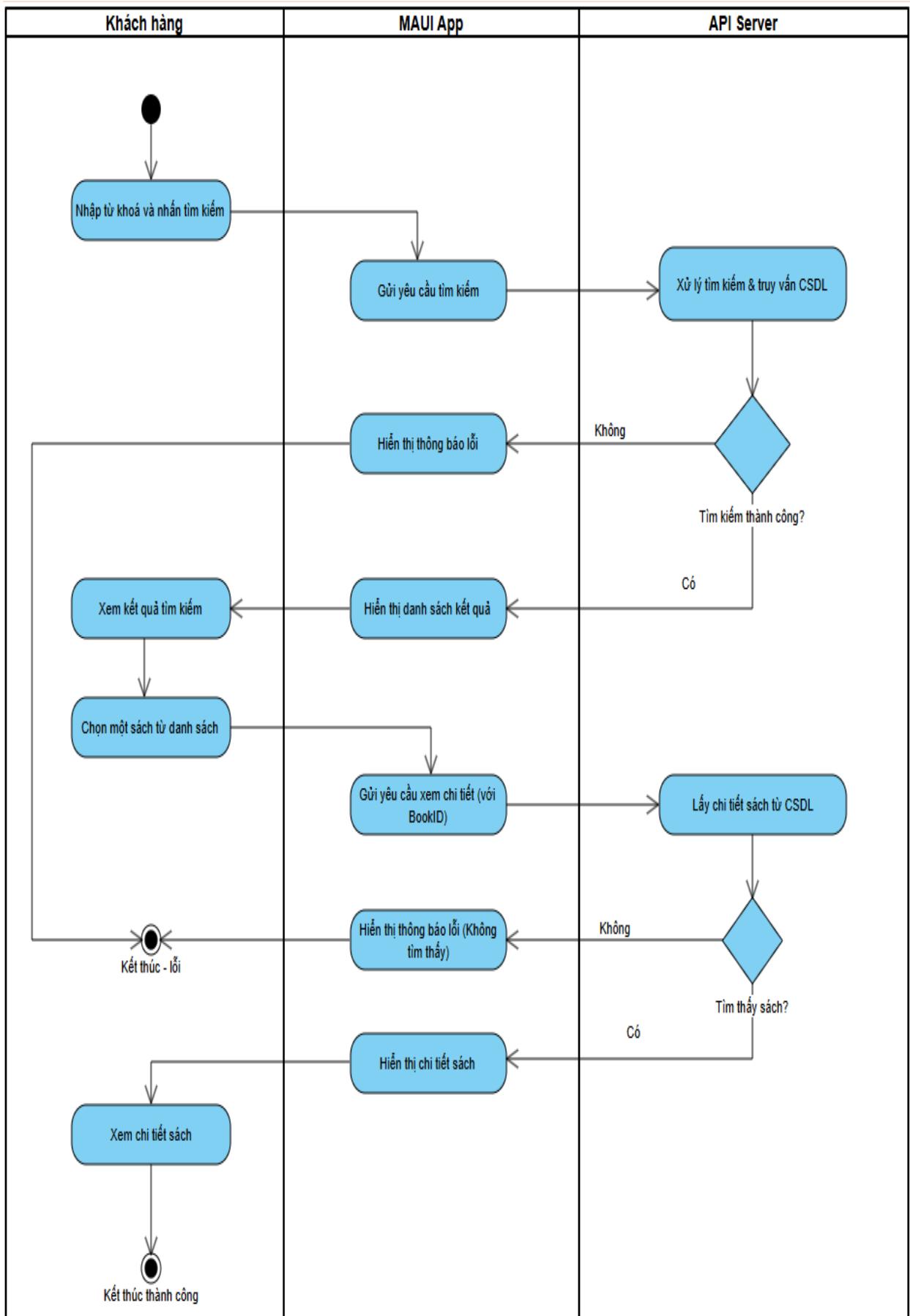
(Threads: 16, Duration: 30 seconds)

Transaction type	Result		Total requests	Response Time (ms)			Request (req/sec)
	Succ	Fail		Min	Max	Avg	
GET All Books	481	0	481	3.24	278.91	13.20	16
GET Order History	240	0	240	6.87	232.54	28.15	8
POST Add to Cart	471	0	471	7.77	238.91	37.38	15
POST Stock Receipt	163	0	163	6.10	386.18	74.16	5
PUT Update Book	240	0	240	13.03	349.96	41.03	8
PUT Update Order Status	240	0	240	7.42	230.13	31.84	8
DELETE Address	496	0	496	2.52	159.67	12.88	16
DELETE Author	496	0	496	5.05	293.36	21.44	15

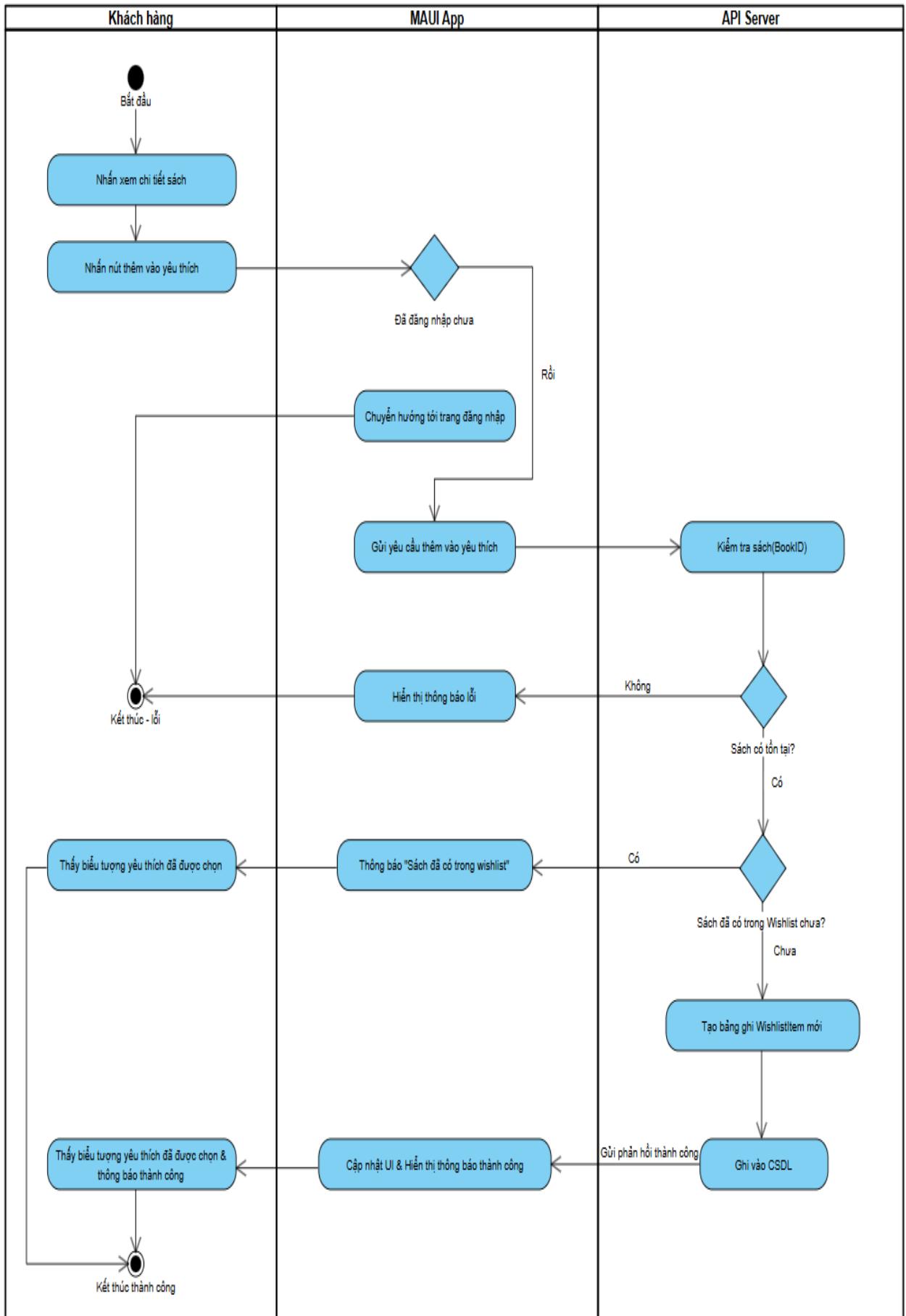
Hình 19. Đánh giá hiệu năng của một số API nổi bật

## **PHỤ LỤC**

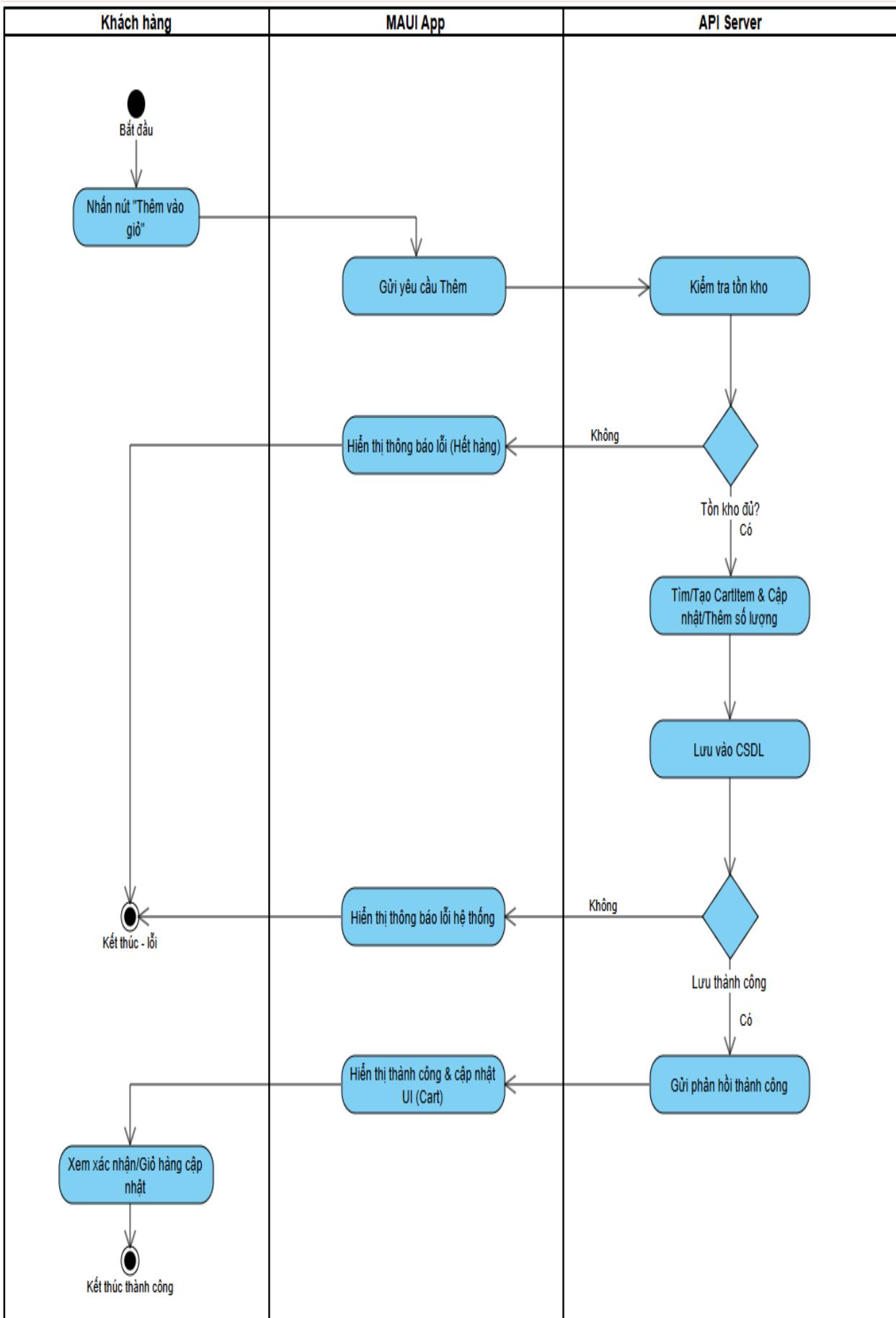
### **PHỤ LỤC : SƠ ĐỒ TUẦN TỰ - ACTIVITY DIAGRAM**



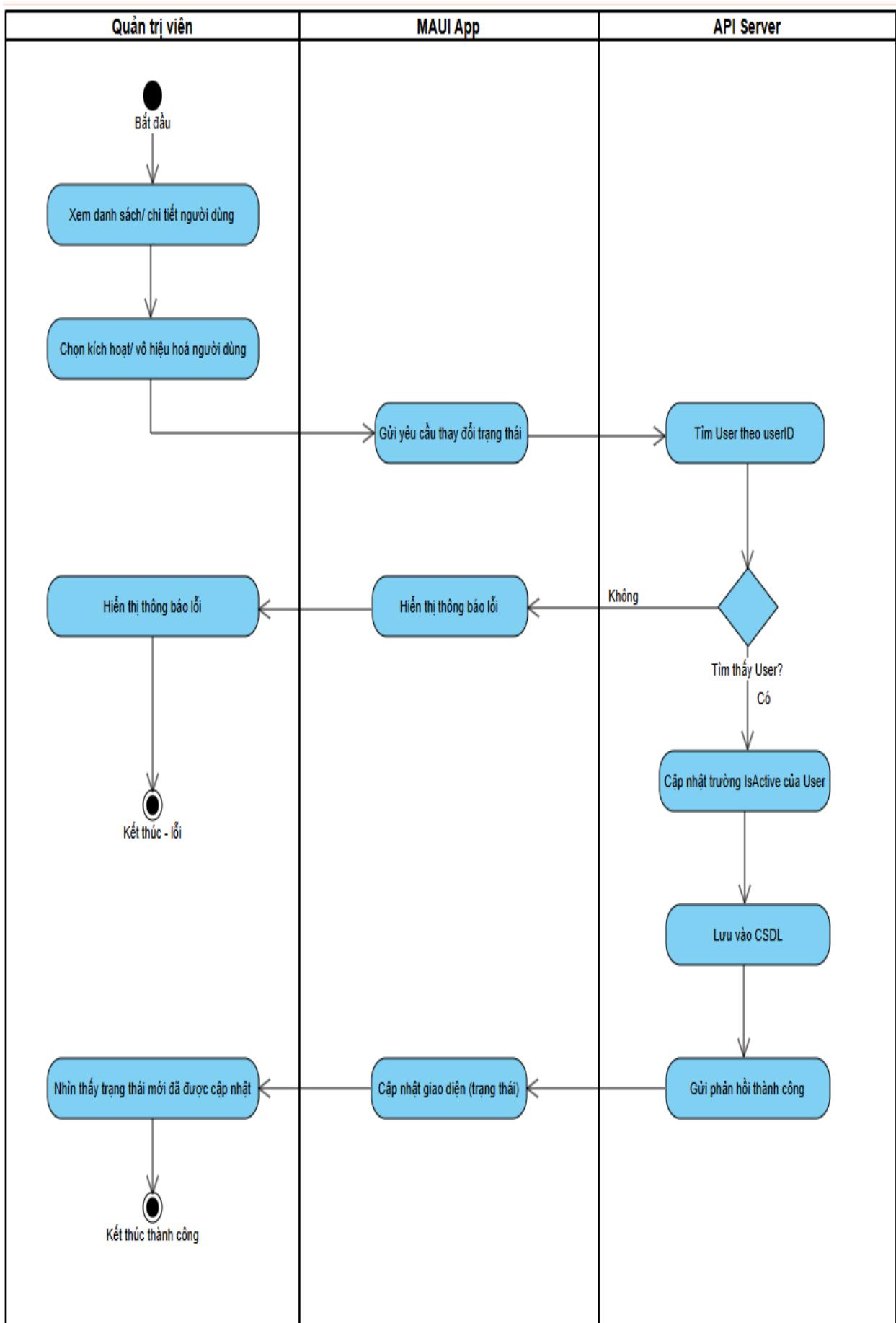
Hình 20. Activity Diagram - Tìm kiếm và xem chi tiết sách



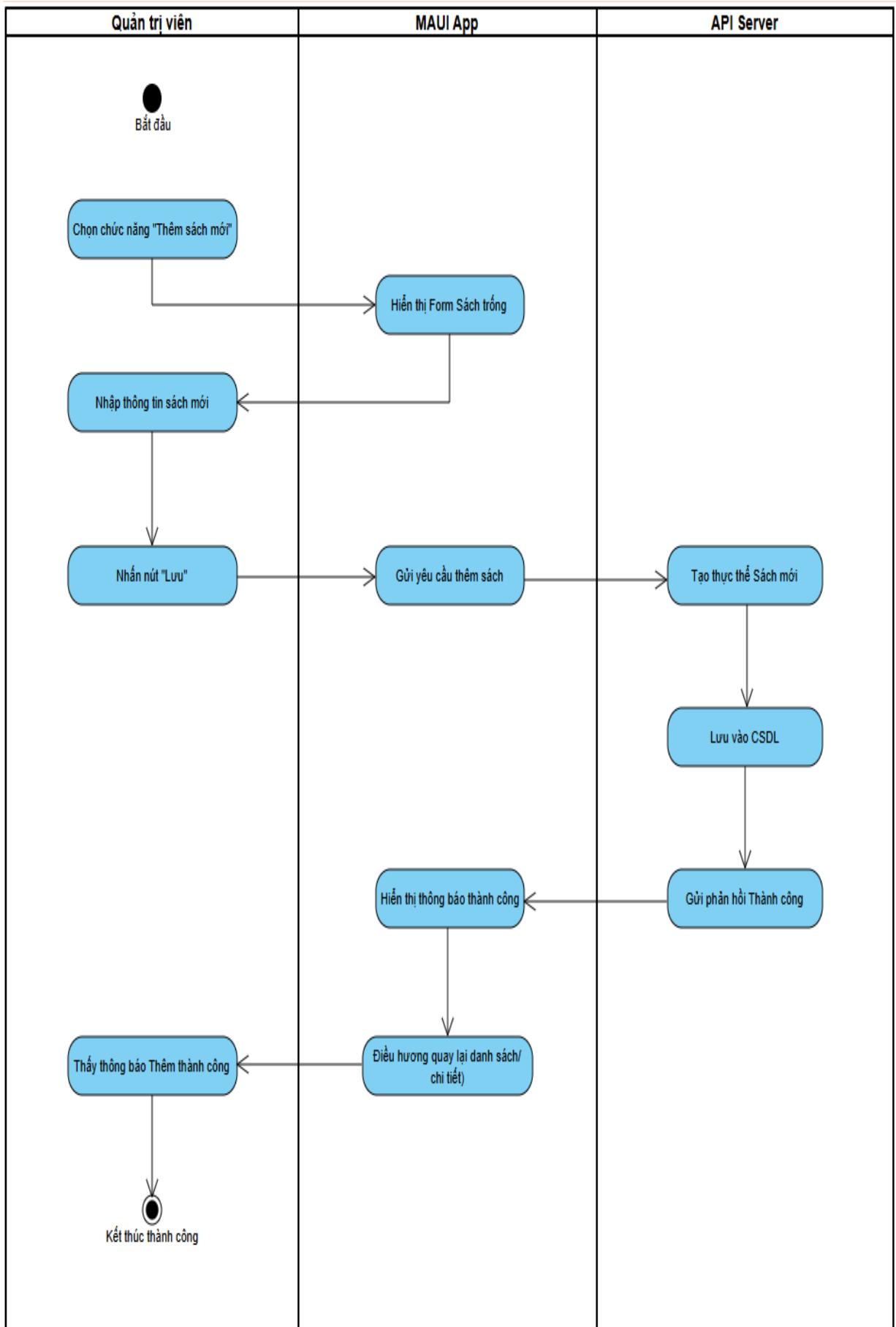
Hình 21. Activity Diagram - Thêm vào yêu thích



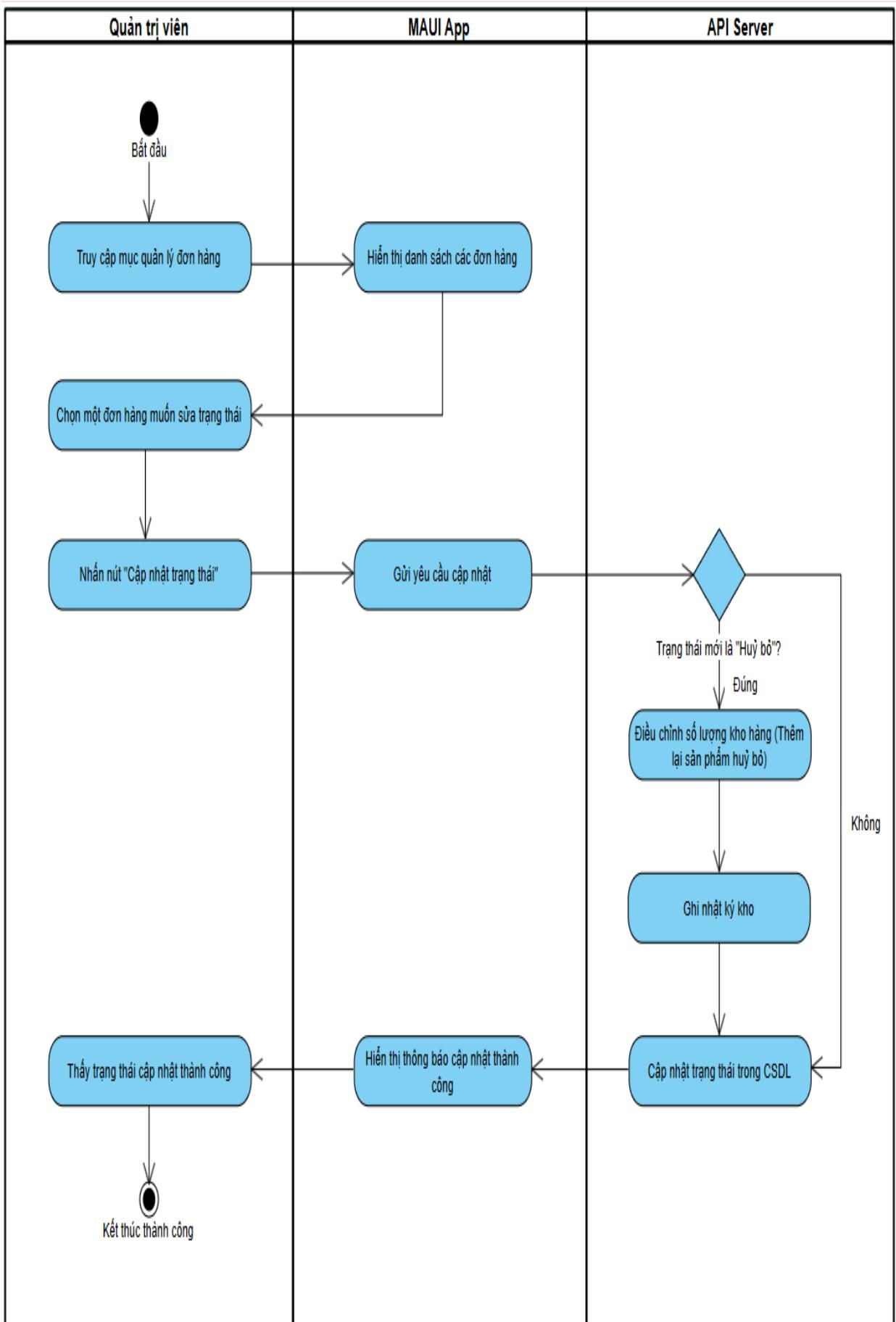
Hình 22. Activity Diagram - Thêm sách vào giỏ hàng



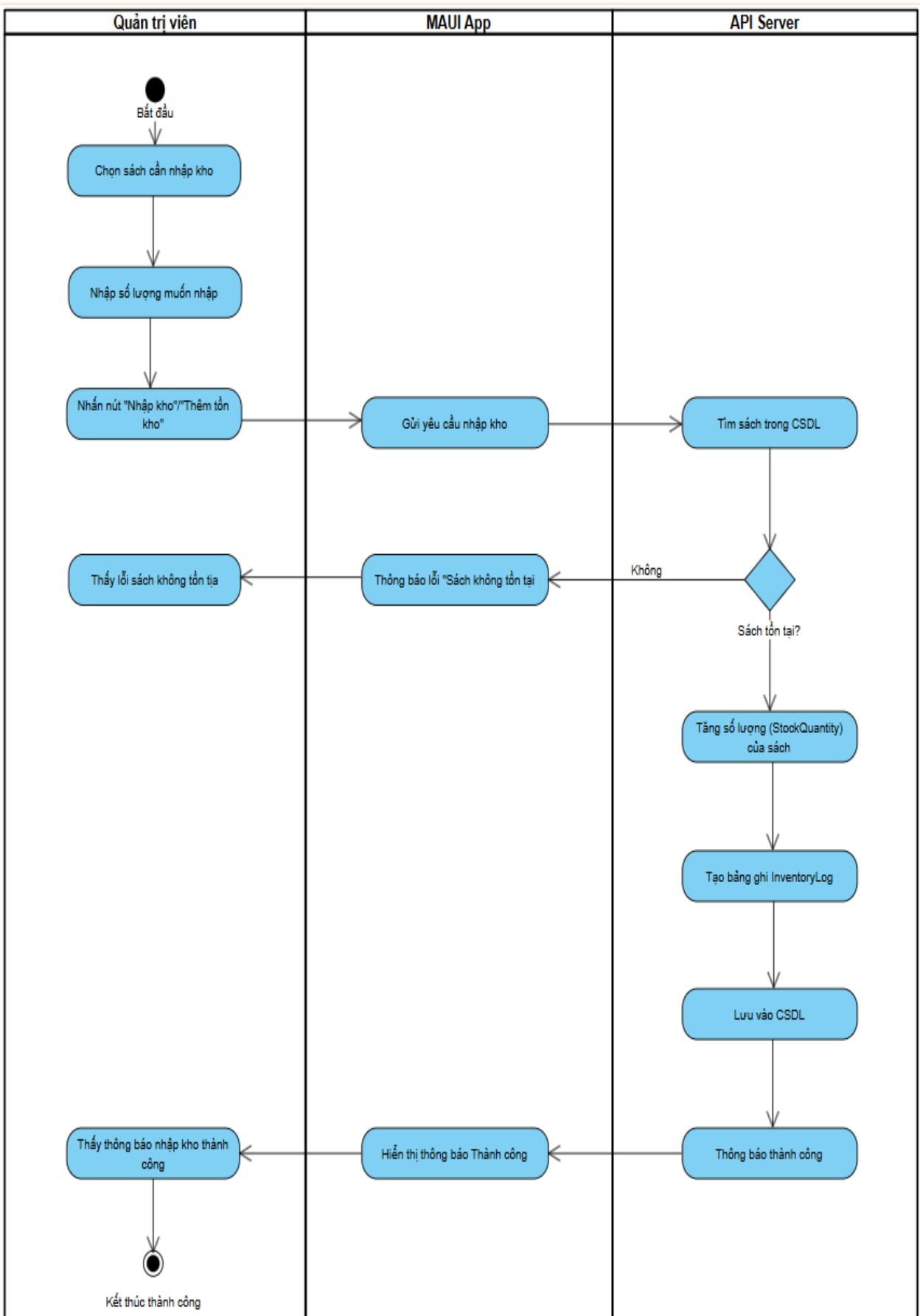
Hình 23. Activity Diagram - Admin quản lý người dùng



Hình 24. Activity Diagram - Admin thêm sách



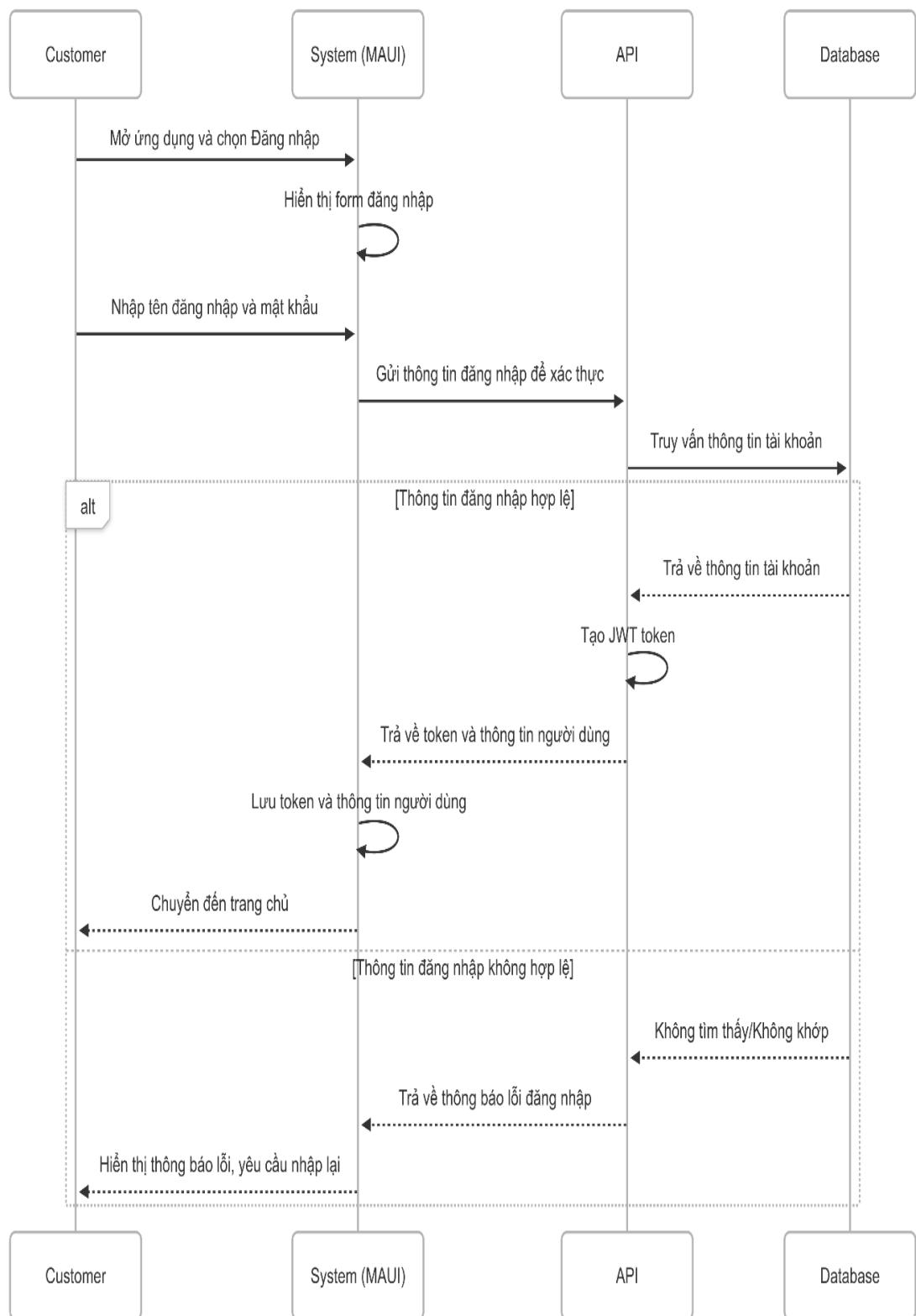
Hình 25. Activity Diagram - Admin cập nhật trạng thái đơn hàng



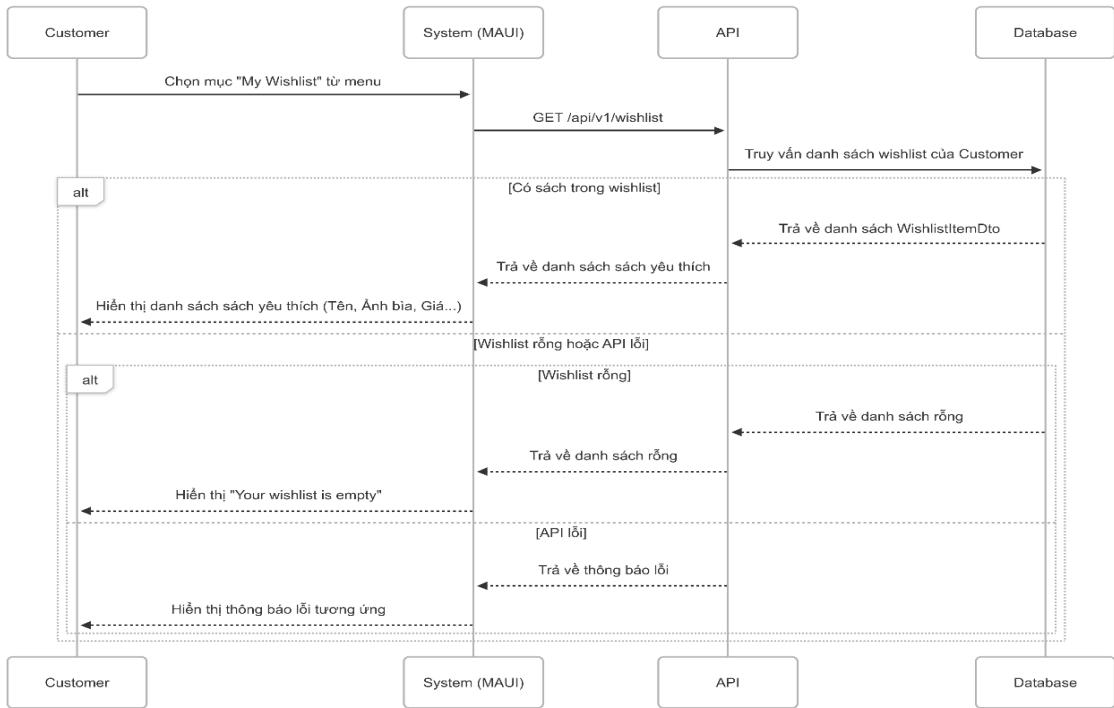
Hình 26. Activity Diagram - Admin nhập kho

## PHỤ LỤC D: SƠ ĐỒ TUẦN TỤ - SEQUENCE DIAGRAM

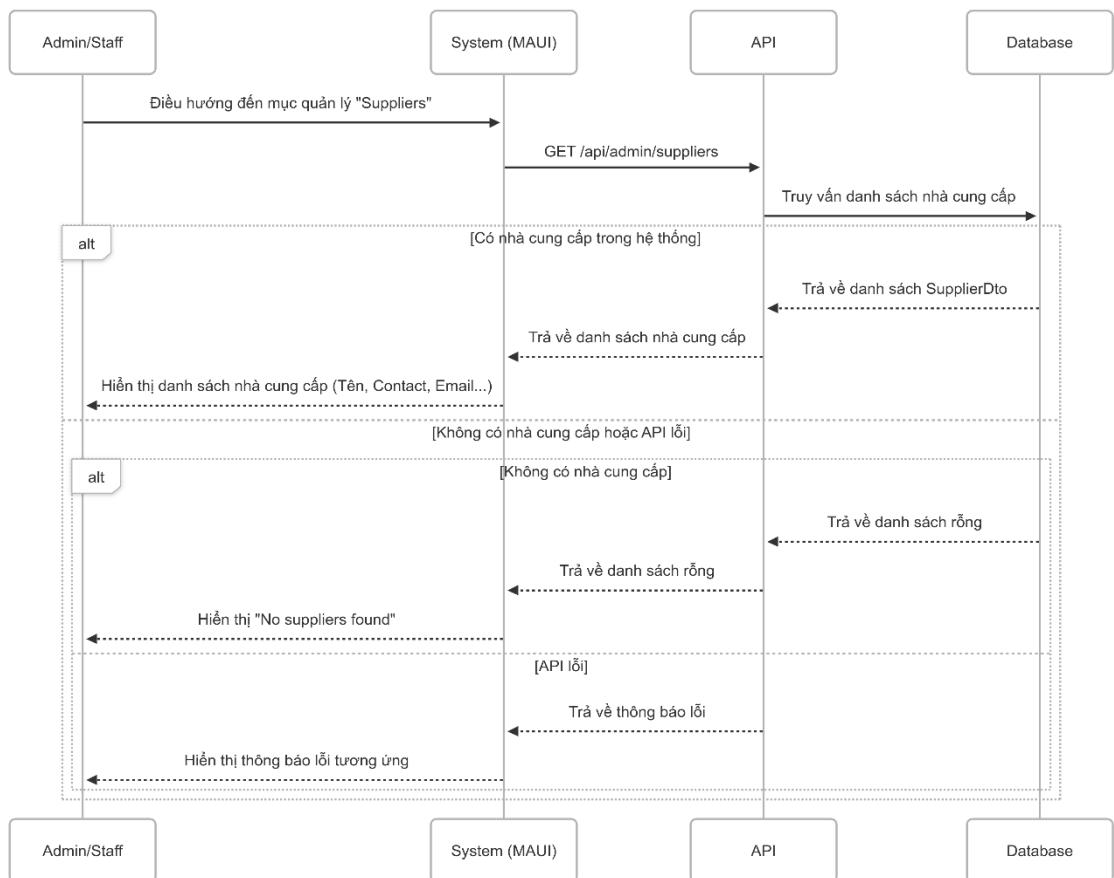
Dưới đây là một vài Sequence Diagram tiêu biểu trong ứng dụng:



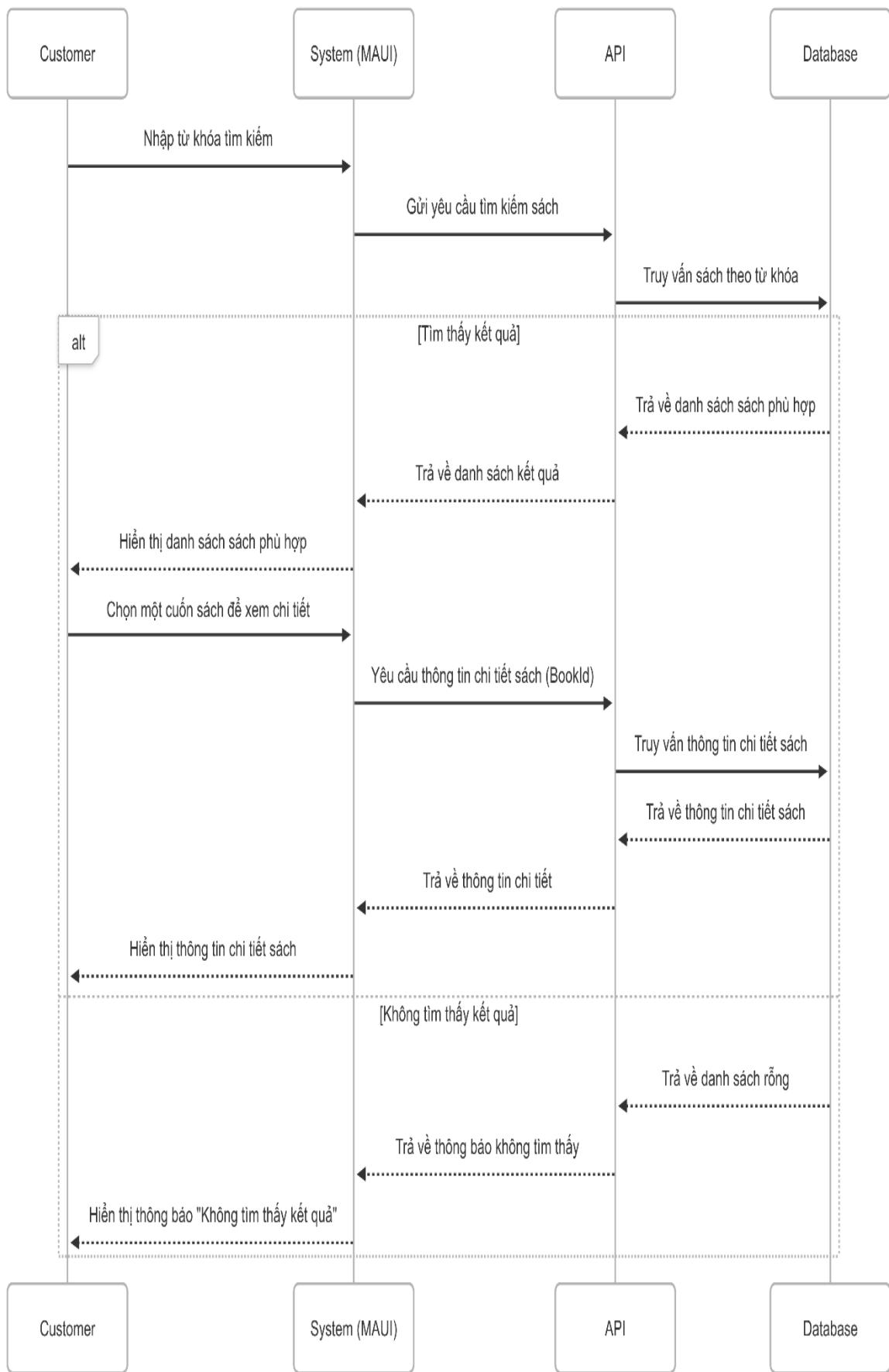
Hình 27. Sequence Diagram - Đăng nhập



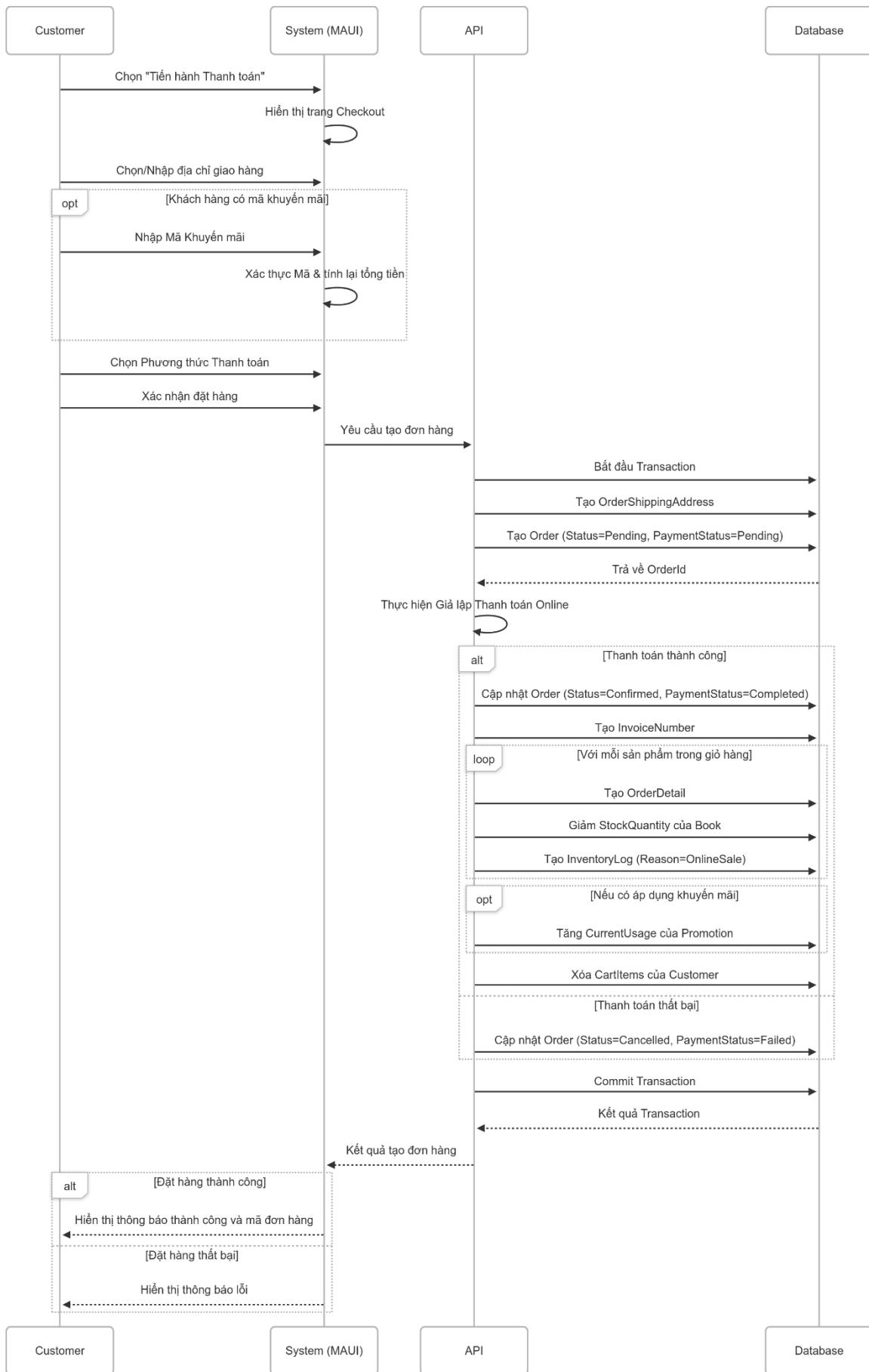
Hình 28. Sequence Diagram - Thêm vào yêu thích



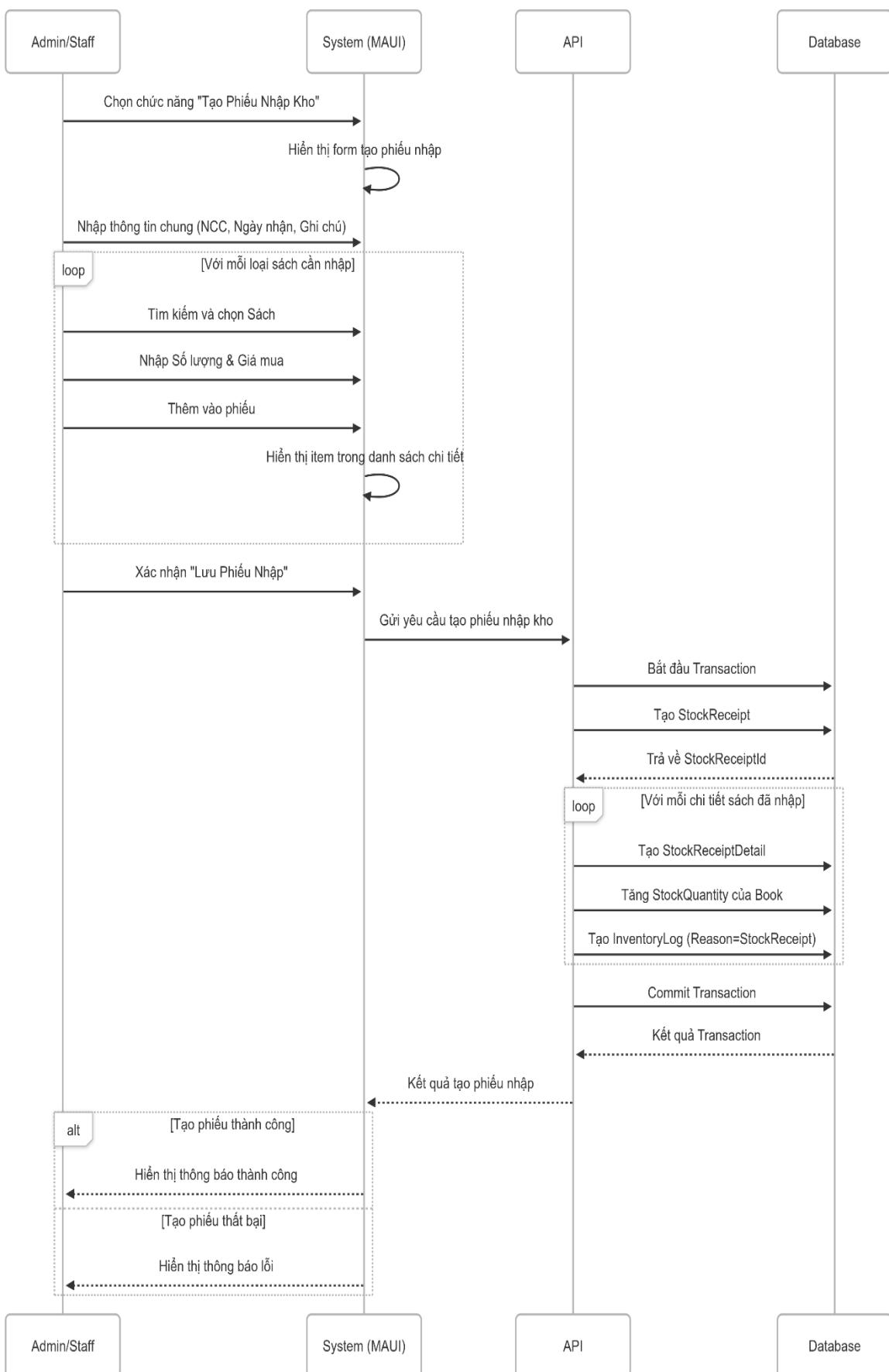
Hình 29. Sequence Diagram - Xem nhà cung cấp



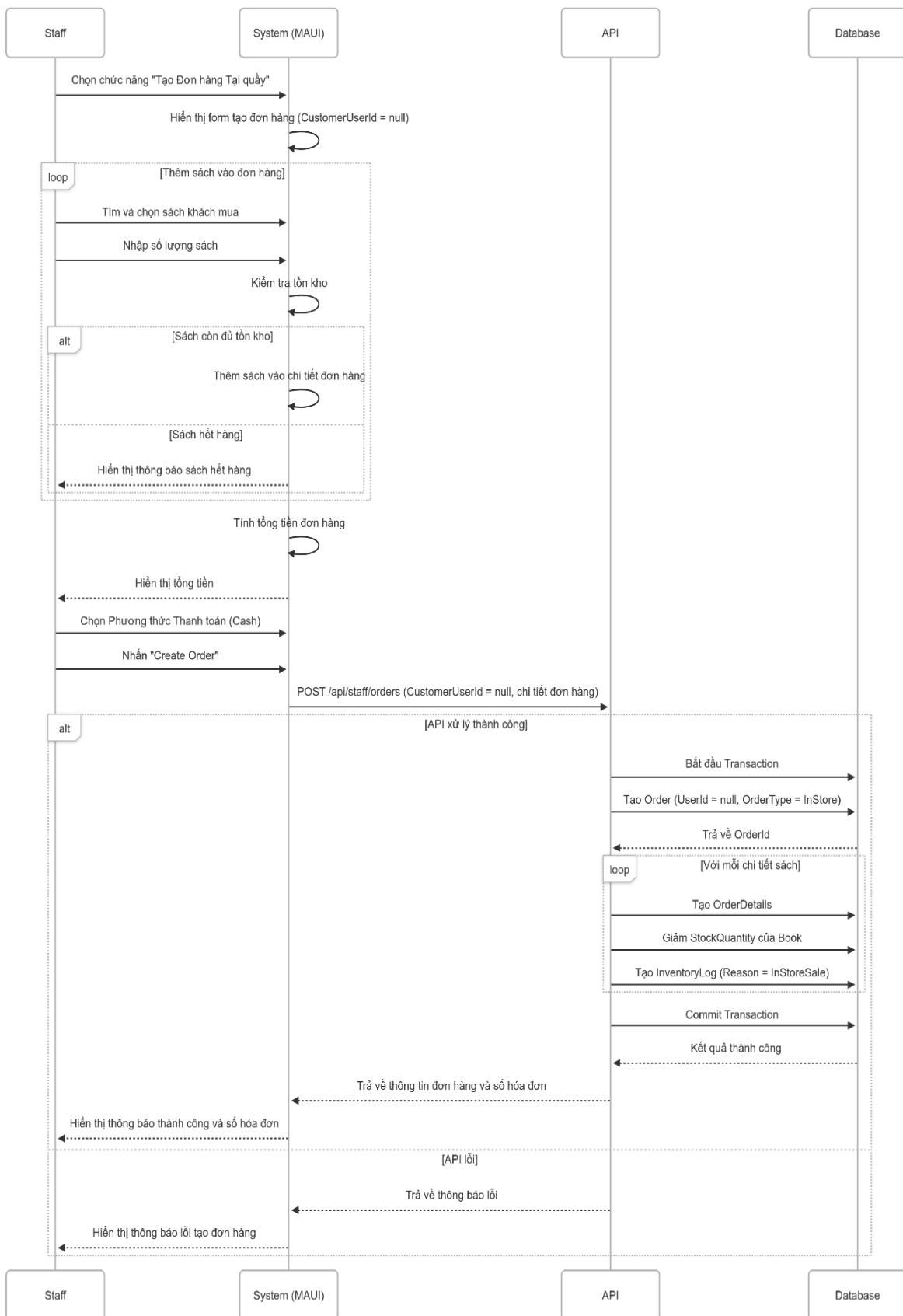
Hình 30. Sequence Diagram - Xem chi tiết sách



Hình 31. Sequence Diagram - Thanh toán



Hình 32. Sequence Diagram - Tạo phiếu nhập



Hình 33. Sequence Diagram - Tạo đơn hàng tại quầy