

# Phân tích thiết kế hệ thống bán hàng second hand

## I. Kiến trúc tổng thể hệ thống (High-Level Architecture)

### 1.1. Thành phần chính

#### 1. Storefront (Angular Client – Customer UI)

- Chạy trên trình duyệt, build bằng Angular.
- Chức năng: xem/lọc sản phẩm, chi tiết sản phẩm, giỏ hàng, checkout, đăng ký/đăng nhập, lịch sử đơn hàng.
- Giao tiếp với Backend qua **REST API + JWT**.

#### 2. Admin Dashboard (Angular Client – Admin UI)

- Chạy trên trình duyệt, cũng là một Angular app (có thể là project riêng hoặc module riêng).
- Chức năng: quản lý sản phẩm, danh mục, đơn hàng, thống kê.
- Giao tiếp với Backend qua **REST API + JWT**, nhưng các API yêu cầu **Role = Admin**.

#### 3. ASP.NET Core Web API (.NET 8)

Áp dụng **Layered Architecture**:

- **Presentation Layer (API Controllers)**
  - Expose các endpoint REST: `/api/products`, `/api/orders`, `/api/auth`, `/api/categories`, `/api/admin/*...`

- Xử lý HTTP: routing, model binding, validation cơ bản, trả HTTP status code.
- Không chứa business logic phức tạp (ủy quyền cho Service Layer).

#### ○ **Business Logic Layer (Services)**

- Chứa toàn bộ quy tắc nghiệp vụ:
  - Xử lý hàng second-hand: status **Available / Sold / Hidden**.
  - Đảm bảo tính **duy nhất** của sản phẩm limited: khi checkout thành công, khóa sản phẩm.
  - Luồng đơn hàng: Pending → Shipping → Completed / Cancelled.
  - Tính toán **TotalAmount**, kiểm tra quyền (Customer/Admin).
- Là nơi dễ test và mở rộng.

#### ○ **Data Access Layer (Repositories)**

- Đảm nhiệm truy xuất dữ liệu: đọc/ghi MySQL thông qua **EF Core + Pomelo**.
- Đóng gói truy vấn: lấy list sản phẩm theo filter, lấy đơn hàng theo user, cập nhật trạng thái sản phẩm, v.v.
- Không chứa logic nghiệp vụ, chỉ tập trung CRUD, query.

#### 4. MySQL Database

- Các bảng chính:
  - **Users, Categories, Products, Orders, OrderDetails.**
- Được truy cập thông qua **EF Core** (DbContext + entity mapping).

#### 5. Cross-cutting concerns (chạy xuyên tầng):

- **Authentication/Authorization (JWT):** middleware trong Web API.
- **CORS:** cấu hình chỉ cho phép Angular Frontend (localhost:4200).
- **Logging, Error Handling:** middleware ở backend (không cần vẽ chi tiết, có thể chú thích chung).

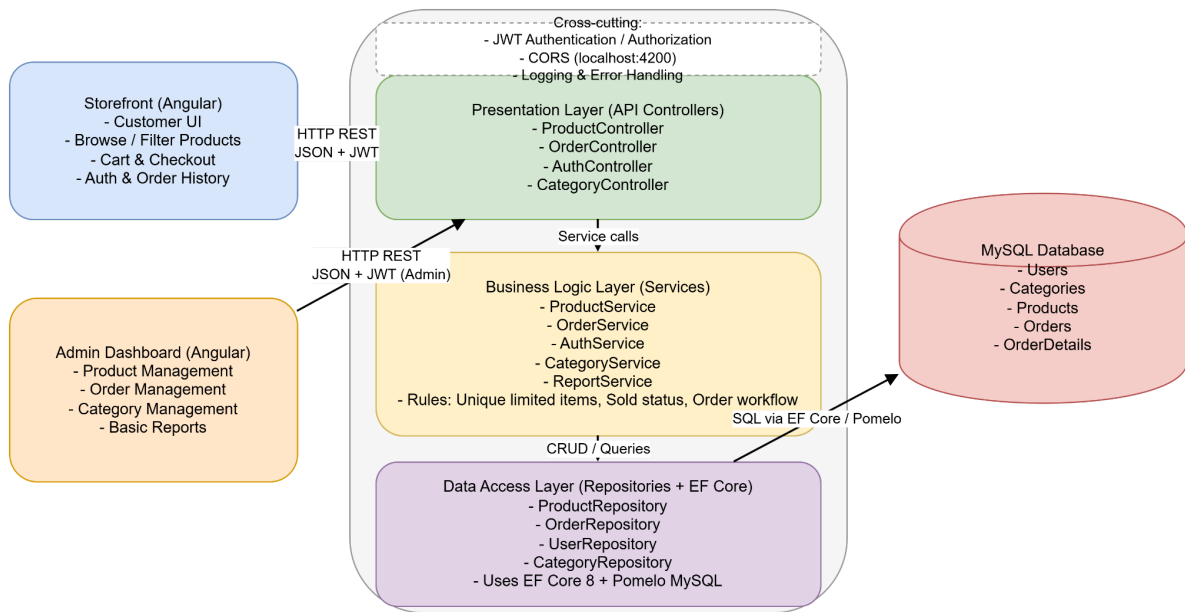
## II. Mô hình Layered: Mapping thành phần → Tầng

Tầng	Thành phần cụ thể
<b>Presentation Layer</b>	Angular Storefront, Angular Admin Dashboard, API Controllers (ProductController, OrderController, ...)
<b>Business Logic Layer</b>	Services (ProductService, OrderService, AuthService, CategoryService, ReportService...)
<b>Data Access Layer</b>	Repositories (ProductRepository, OrderRepository, UserRepository, CategoryRepository, ...) + DbContext
<b>Database Layer</b>	MySQL (Users, Products, Categories, Orders, OrderDetails)

Lưu ý:

- Về mặt **frontend**, Angular apps cũng thuộc Presentation Layer (Client side).
- Về mặt **backend**, Controllers cũng là Presentation Layer (Server side).
- Business Logic & Data Access nằm hoàn toàn trong Web API.

### III. Sơ đồ kiến trúc hệ thống cho hệ thống:



Mô tả ngắn gọn luồng hoạt động của hệ thống:

#### Luồng từ Client (Angular) → Web API

Có 2 ứng dụng phía client:

##### 1. Storefront (Angular – Customer)

- Người dùng xem sản phẩm, lọc, xem chi tiết, thêm vào giỏ, checkout.
- Khi thao tác, Angular gửi request **HTTP REST (JSON + JWT)** đến Web API.  
Ví dụ:

- **GET /api/products**
- **POST /api/orders**
- **POST /api/auth/login**

##### 2. Admin Dashboard (Angular – Admin)

- Admin đăng nhập, quản lý sản phẩm, đơn hàng, danh mục, thống kê.

- Request gửi kèm **JWT Token có Role=Admin**.

Ví dụ:

- **POST /api/admin/products**
- **GET /api/admin/orders**

### **3. Web API – Presentation Layer (Controllers)**

Khi client gửi request đến:

- Controllers tiếp nhận HTTP request.
- Xác thực JWT (và Role nếu là Admin).
- Thực hiện:
  - Routing
  - Validation input
  - Trả mã lỗi đúng chuẩn HTTP

Sau đó **controllers chuyển công việc chính sang Services** (không xử lý logic nghiệp vụ phức tạp).

### **Business Logic Layer (Services)**

Tầng này xử lý toàn bộ **ng nghiệp vụ** của hệ thống:

- Kiểm tra sản phẩm còn "Available" hay đã "Sold".
- Xử lý logic mặt hàng giới hạn (Limited).
- Đặt hàng (Order workflow):
  - Pending → Shipping → Completed.

- Tính tổng hóa đơn.
- Quản lý sản phẩm.
- Quản lý danh mục.
- Quyền của người dùng.
- Báo cáo đơn giản.

**Services chỉ tập trung đúng nghiệp vụ** → không truy cập database trực tiếp.

Khi cần dữ liệu, Services gọi tiếp xuống Repository.

#### **4. Data Access Layer (Repositories + EF Core)**

Nhiệm vụ:

- Thực hiện các thao tác CRUD:
  - Lấy danh sách sản phẩm (kèm filter).
  - Tạo đơn hàng.
  - Cập nhật trạng thái sản phẩm.
  - Lấy đơn theo user.
- Dùng EF Core + Pomelo để giao tiếp MySQL.
- Không chứa logic nghiệp vụ.

Kết quả truy vấn được trả lên Services.

## 5. MySQL Database

Các Repository thực hiện truy cập:

- **Users**
- **Products**
- **Categories**
- **Orders**
- **OrderDetails**

Data trả về → Repository → Service → Controller → Angular UI.

## 5. Kết luận luồng dữ liệu

**Client Angular → Controllers → Services → Repositories → MySQL**  
và sau đó response trả ngược lại theo chiều ngược lại.

Cách luồng di chuyển:

1. Angular gửi request
2. API Controllers nhận request
3. Controllers gọi Services xử lý nghiệp vụ
4. Services gọi Repository để truy vấn MySQL
5. Repository đọc/ghi dữ liệu
6. Kết quả trả về Services
7. Service trả về Controller
8. Controller trả JSON response cho Angular UI

### III. Danh sách Use Case:

Dưới đây là bảng Use Case (Functional Decomposition), chia theo actor:

khách, admin, thành viên:

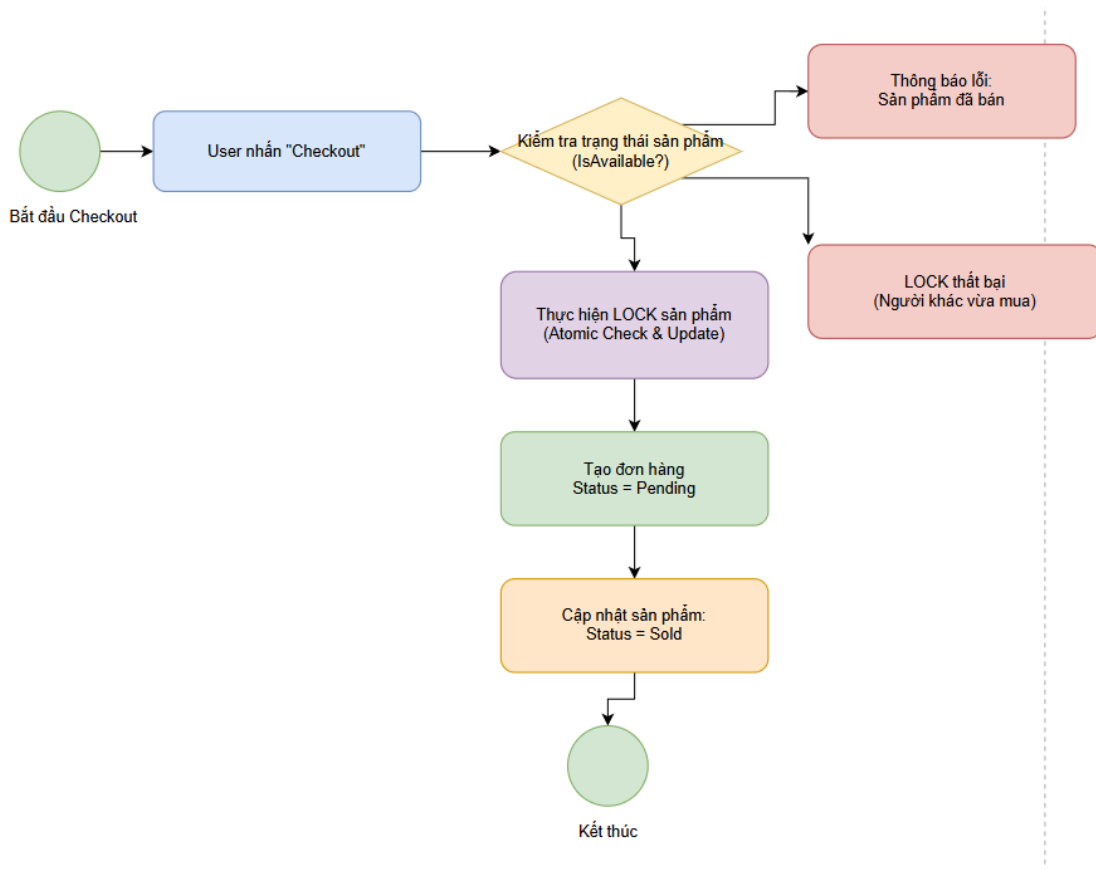
Actor	Use Case ID	Tên Use Case	Mô tả ngắn	Input	Output/kết quả
Khách (Guest)	UC-G-01	Duyệt & Lọc sản phẩm	Người không đăng nhập xem danh sách sản phẩm và áp dụng bộ lọc.	Tham số filter (giá, loại, size, độ mới).	Hiển thị danh sách sản phẩm phù hợp + nhãn “Hàng mới về”, “Đã bán”.
Khách (Guest)	UC-G-02	Xem chi tiết sản phẩm	Xem mô tả, ảnh, thông tin đầy đủ của sản phẩm.	Chọn 1 sản phẩm.	Hiển thị chi tiết sản phẩm, trạng thái Available/Sold.
Khách (Guest)	UC-G-03	Đăng ký tài khoản	Khách muốn trở thành thành viên để mua hàng.	Họ tên, username, email, mật khẩu, địa chỉ.	Tạo tài khoản Customer; thông báo thành công / lỗi.
Khách (Guest)	UC-G-04	Đăng nhập	Khách đăng nhập để mua hàng.	Username/email, mật khẩu.	JWT token; hiển thị trạng thái đăng nhập; lỗi nếu sai.
Thành viên (Customer)	UC-C-01	Duyệt & Lọc sản phẩm	Thành viên đăng nhập xem danh sách sản phẩm và filter.	Filter (giá, loại, size, độ mới).	Hiển thị danh sách sản phẩm theo filter.
Thành viên (Customer)	UC-C-02	Xem chi tiết sản phẩm	Xem đầy đủ thông tin + trạng thái mua.	Chọn sản phẩm.	Hiển thị chi tiết + nút “Thêm vào giỏ” (nếu Available).
Thành viên (Customer)	UC-C-03	Thêm sản phẩm vào giỏ	Thêm sản phẩm Available vào giỏ.	Sản phẩm chọn.	Giỏ hàng cập nhật; báo thành công hoặc lỗi (nếu Sold).



Thành viên (Customer)	UC-C-04	Quản lý giỏ hàng	Xem giỏ, thay đổi số lượng, xóa sản phẩm.	Thao tác trên giỏ hàng.	Giỏ hàng cập nhật; tổng tiền mới.
Thành viên (Customer)	UC-C-05	Checkout / Đặt hàng	Nhập thông tin giao hàng và xác nhận đặt hàng; hệ thống khóa sản phẩm.	Thông tin giao hàng; xác nhận Checkout JWT Customer	Tạo đơn hàng (Pending); sản phẩm chuyển sang Sold; trả mã đơn.
Thành viên (Customer)	UC-C-06	Xem lịch sử đơn hàng	Thành viên xem lại đơn hàng cũ và trạng thái.	Yêu cầu xem lịch sử; JWT Customer	Hiển thị danh sách đơn: Pending, Shipping, Completed, Cancelled.
Admin	UC-A-01	Đăng nhập Admin	Nhân viên đăng nhập vào Dashboard quản trị.	Username, password Admin.	JWT role = Admin; vào trang Dashboard.
Admin	UC-A-02	Thêm mới sản phẩm	Tạo mới sản phẩm second-hand.	Tên, danh mục, giá, độ mới, số đo, mô tả, ảnh, trạng thái.	Sản phẩm mới được lưu; hiển thị trong hệ thống.
Admin	UC-A-03	Cập nhật sản phẩm	Sửa thông tin sản phẩm; thay đổi trạng thái (Available, Hidden, Sold).	Thông tin sửa đổi.	Sản phẩm được cập nhật; Storefront phản ánh trạng thái.
Admin	UC-A-04	Quản lý danh mục (CRUD)	Thêm, sửa, xóa danh mục quần áo.	Input danh mục: tên, mô tả.	Danh mục được tạo/cập nhật/xóa; Storefront cập nhật filter.

Admin	UC-A-05	Xem danh sách đơn hàng	Xem tất cả đơn hàng trong hệ thống.	(Tuỳ chọn) Filter theo ngày / trạng thái.	Danh sách đơn + thông tin chi tiết.
Admin	UC-A-06	Cập nhật trạng thái đơn	Cập nhật luồng: Pending → Shipping → Completed/Cancelled	Trạng thái mới.	Đơn hàng cập nhật trạng thái; báo cáo doanh thu thay đổi.
Admin	UC-A-07	Thống kê doanh thu	Xem báo cáo doanh thu theo ngày/tháng.	Khoảng thời gian.	Bảng/biểu đồ doanh thu.
Admin	UC-A-08	Thống kê tồn kho	Xem số lượng sản phẩm còn Available theo danh mục.	Yêu cầu xem tồn kho.	Báo cáo số lượng tồn kho.

## IV. Luồng hoạt động (Activity Diagram) - Logic Cốt lõi:



Mô tả chi tiết luồng hoạt động:

### LUỒNG HOẠT ĐỘNG CHECKOUT — MÔ TẢ CHI TIẾT

Luồng này mô tả **logic cốt lõi của một hệ thống bán đồ Second-hand**, nơi mỗi sản phẩm chỉ có **1 bản duy nhất**, và có thể xảy ra **2 người cùng lúc bấm mua** → hệ thống phải xử lý cạnh tranh (race condition).

#### 1. Bắt đầu quy trình Checkout

Quy trình bắt đầu khi **người dùng đã có sản phẩm trong giỏ hàng và nhấn nút "Checkout"**.

## 2. User nhấn “Checkout”

Hệ thống tiếp nhận yêu cầu:

- Lấy thông tin giỏ hàng
- Lấy danh sách sản phẩm chuẩn bị thanh toán
- Chuẩn bị bước kiểm tra tình trạng sản phẩm

Sau khi nhận request, hệ thống chuyển sang bước **kiểm tra trạng thái sản phẩm**.

## 3. Kiểm tra trạng thái sản phẩm (IsAvailable?)

Hệ thống truy vấn database để biết sản phẩm có còn **Available** hay không.

Có 2 trường hợp:

### Trường hợp 1: Sản phẩm **KHÔNG** còn Available

Ví dụ:

Một người khác đã mua trước vài giây.

**Kết quả:**

- Trả về lỗi: “**Sản phẩm đã bán**”
- Checkout kết thúc — không tạo đơn

→ Đây là nhánh “Không” trong sơ đồ.

### Trường hợp 2: Sản phẩm vẫn còn Available

Hệ thống cho phép đi tiếp đến bước quan trọng nhất: **khóa sản phẩm (locking)**.

#### 4. LOCK sản phẩm (Concurrency Control – Atomic Check & Update)

Đây là bước trọng tâm của logic hệ thống:

- Backend dùng transaction hoặc câu lệnh atomic (UPDATE WHERE Status = Available)
- Chỉ một người có thể lock thành công

Có 2 nhánh:

##### Trường hợp LOCK thất bại

Nguyên nhân:

Một user khác **đã lock sản phẩm trước trong cùng thời điểm.**

Hệ thống trả về:

- Thông báo: “**Người khác vừa mua sản phẩm này**”
- Hủy quá trình Checkout

→ Đây là logic chống **race-condition**.

##### Trường hợp LOCK thành công

Hệ thống đánh dấu sản phẩm **tạm thời khóa** để đảm bảo:

- Không ai khác có thể thao tác mua trong lúc đơn đang được tạo
- Tránh mất đồng bộ giữa lúc kiểm tra và lúc tạo đơn

Tiếp tục sang bước tiếp theo.

## 5. Tạo đơn hàng (Order = Pending)

Khi đã lock thành công:

- Tạo Order:
  - UserId
  - List sản phẩm
  - Giá tiền
  - Thông tin giao hàng
  - Thời gian tạo
  - Status = **Pending**
- Tạo OrderDetails cho từng sản phẩm

Đây là bước chính thức xác nhận giao dịch được tạo trong hệ thống.

## 6. Cập nhật sản phẩm thành Sold

Sau khi order được tạo:

- Sản phẩm được cập nhật trạng thái:  
**Status = Sold**

→ Điều này chính thức “khóa” sản phẩm khỏi hệ thống

→ Các user khác vào website sẽ thấy sản phẩm bị làm mờ / không mua được.

## 7. Kết thúc quy trình

Hệ thống gửi phản hồi:

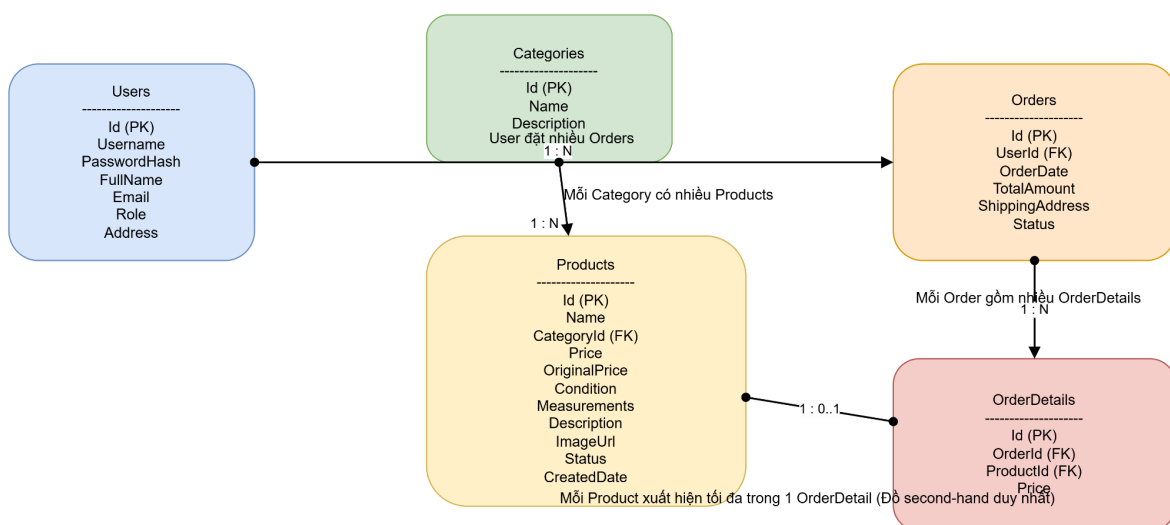
**Thành công:**

- “Checkout thành công”
- Trả về mã đơn hàng
- User được chuyển sang màn hình Order Success

**Thất bại (ở nhánh song song):**

- “Sản phẩm đã bán”
- “Người khác vừa mua sản phẩm này”
- Không tạo đơn

## V. Sơ đồ Thực thể Quan hệ (ERD - Conceptual Level):



## **Giải thích nhanh về ERD (logic conceptual):**

### **Users – Orders:**

- 1 User có thể có **N Orders**.
- Một Order thuộc về đúng 1 User.

### **Categories – Products:**

- 1 Category có **nhiều Products**.
- Mỗi Product thuộc đúng 1 Category.

### **Orders – OrderDetails:**

- 1 Order có **nhiều dòng OrderDetails**.
- Mỗi OrderDetail thuộc về đúng 1 Order.

### **Products – OrderDetails (điểm quan trọng):**

- Mỗi sản phẩm second-hand là **duy nhất** → **tại thời điểm đã bán**, Product chỉ được gắn vào **tối đa 1 OrderDetail**.
- Quan hệ khái niệm: **Product 1 – 0..1 OrderDetail** (0 nếu chưa bán, 1 nếu đã bán).
- Trong sơ đồ: hiển thị là **1 : 0..1** để nhấn mạnh logic “mỗi sản phẩm chỉ được bán đúng 1 lần”.

## **VI. UI FLOW – NAVIGATION MAP (Từ Trang Chủ → Mua Hàng Thành Công)**

### **1. Home Page (Trang chủ)**

#### **Mục đích:**

- Hiển thị danh sách sản phẩm mới nhất
- Gợi ý các loại sản phẩm / danh mục
- Cho phép người dùng bắt đầu hành trình mua sắm



**Từ Home Page, user có thể điều hướng đến:**

- ▶ **Product Listing (Danh sách sản phẩm)**
- ▶ **Product Detail (Chi tiết sản phẩm)**
- ▶ **Login / Register**
- ▶ **Cart (Giỏ hàng)**
- ▶ **Category Listing (lọc theo danh mục)**

## **2. Product Listing Page (Danh sách sản phẩm)**

**Chức năng:**

- Lọc theo: giá, loại, size, độ mới
- Xem nhãn “Hàng mới về”, “Đã bán (mò)”

**Từ đây, user có thể chuyển đến:**

- ▶ **Product Detail Page** (khi nhấn vào 1 sản phẩm)
- ▶ **Home Page** (back)
- ▶ **Login / Register**
- ▶ **Cart**

## **3. Product Detail Page (Chi tiết sản phẩm)**

**Hiển thị:**

- Ảnh sản phẩm (zoom / gallery)
- Giá bán, độ mới, số đo, mô tả lỗi
- Trạng thái: Available — cho phép thêm vào giỏ

- Trạng thái: Sold — hiển thị mờ, không mua được

**User có thể điều hướng đến:**

- ▶ **Add to Cart** → chuyển đến **Cart Page**
- ▶ **Product Listing**
- ▶ **Home Page**
- ▶ **Login/Register** (nếu chưa đăng nhập và muốn checkout)

#### **4. Cart Page (Giỏ hàng)**

**Chức năng:**

- Xem danh sách sản phẩm đã thêm
- Xóa sản phẩm
- Tổng tiền tạm tính
- Nút **Proceed to Checkout**

**Điều hướng:**

- ▶ **Checkout Page** (nếu user đã đăng nhập)
- ▶ **Login Page** (nếu user chưa đăng nhập → bị redirect)
- ▶ **Product Listing / Product Detail**
- ▶ **Home Page**

#### **5. Login Page (Đăng nhập)**

(Dành cho user chưa đăng nhập nhưng bấm Checkout)

**Chức năng:**

- Nhập username/email + password
- Lấy JWT token

**Điều hướng:**

- ▶ **Register Page** (nếu chưa có tài khoản)
- ▶ **Back to Cart**
- ▶ **Checkout Page** (khi login thành công)

## **6. Register Page (Đăng ký)**

**Chức năng:**

- Tạo tài khoản thành viên

**Điều hướng:**

- ▶ **Login Page** (sau khi đăng ký thành công)
- ▶ **Back to Cart / Product Listing**

## **7. Checkout Page (Thông tin giao hàng + xác nhận thanh toán)**

**Hiển thị:**

- Danh sách sản phẩm trong đơn
- Tổng tiền
- Form nhập thông tin giao hàng
- Nút **Confirm Order**

**Điều hướng:**

- ▶ **Checkout Processing** (gọi API – kiểm tra Available)

- ► **Cart Page** (quay lại chỉnh)

## **8. Checkout Processing (Nút Xác nhận → Gọi API)**

### **Logic Backend:**

1. Kiểm tra trạng thái sản phẩm (Available?)
2. LOCK sản phẩm (atomic)
3. Nếu fail → chuyển user đến màn hình lỗi
4. Nếu success → tạo đơn + mark Sold

### **Điều hướng:**

- ✓ **Order Success Page**
- ✗ **Order Failed Page (Sản phẩm đã bán)**

## **9. Order Success Page (Đặt hàng thành công)**

### **Hiển thị:**

- Mã đơn hàng
- Tóm tắt đơn
- Button: “Xem lịch sử đơn hàng”

### **Điều hướng:**

- ► **Order History Page**
- ► **Home Page**
- ► **Product Listing**

## **10. Order Failed Page (Sản phẩm Sold trong lúc Checkout)**

### **Trường hợp xảy ra race condition**

Hiển thị:

- “Sản phẩm đã được người khác mua trước”
- Gợi ý thêm sản phẩm tương tự

### **Điều hướng:**

- ▶ **Product Listing**
- ▶ **Cart Page**

## **11. Order History Page (Lịch sử đơn hàng)**

(Dành cho user đã đăng nhập)

**Hiển thị:**

- Danh sách đơn: Pending, Shipping, Completed, Cancelled
- Xem chi tiết đơn hàng

### **Điều hướng:**

- ▶ **Order Detail Page**
- ▶ **Home Page**
- ▶ **Product Listing**