

Văn Minh Thức  
Thái Phan Anh Hiểu  
Trần Khánh Bình

# SYSTEM REQUEST & FEASIBILITY

## 1. Case Understanding

### 1.1 Vấn đề doanh nghiệp (Business Problem)

Thị trường bán hàng công nghệ hiện nay phân mảnh: mỗi cửa hàng vận hành riêng, người mua khó so sánh giá, đơn hàng không đồng nhất, thanh toán – giao hàng phải xử lý qua nhiều hệ thống khác nhau.

Do đó cần một **multi-vendor marketplace** tập trung để:

- Cho phép nhiều Seller đăng bán sản phẩm công nghệ.
- Buyer tìm kiếm, so sánh, đặt hàng nhanh chóng
- Thanh toán online tích hợp Payment Gateway.
- Giao hàng tự động qua Shipping Provider.
- Theo dõi trạng thái đơn thời gian thực.

### 1.2 Giới thiệu dự án (Project Description)

- **Vấn đề:** Nhu cầu mua sắm thiết bị công nghệ trực tuyến ngày càng cao, nhưng các nền tảng hiện tại có thể thiếu tính chuyên biệt hoặc trải nghiệm người dùng chưa tối ưu về tốc độ và tra cứu thông số kỹ thuật.
- **Giải pháp:** Xây dựng hệ thống E-commerce chuyên dụng cho đồ công nghệ với kiến trúc hiện đại, tập trung vào hiệu năng cao và trải nghiệm người dùng mượt mà.
- **Mục tiêu:** Hoàn thiện hệ thống cho phép tìm kiếm, đặt hàng, thanh toán và quản lý đơn hàng hiệu quả.

### 1.3 Khả thi về Kỹ thuật (Technical Feasibility)

- **Kiến trúc:** Sử dụng kiến trúc Microservices (hoặc Modular Monolith tùy thực tế triển khai) để đảm bảo tính linh hoạt và khả năng mở rộng (Scalability).
- **Backend:** ASP.NET Core (Mạnh mẽ, bảo mật cao, phù hợp enterprise), Entity Framework Core (ORM), SignalR (Real-time notification).
- **Frontend:** React với TypeScript (Type-safe, dễ bảo trì), React Router v7, Context API (State management), Axios.
- **Database:** SQL Server (Quan hệ, chặt chẽ về dữ liệu).
- **Lưu trữ & UI:** Firebase (Lưu trữ ảnh tối ưu tốc độ), Ant Design + Tailwind CSS (Giao diện chuẩn mực, Responsive).

## 1.4 Khả thi về Kinh tế (Economic Feasibility)

- **Chi phí phát triển:** Thấp, chủ yếu là chi phí thời gian và công sức của thành viên nhóm (nhân lực nội bộ). Sử dụng các công cụ Open-source hoặc License sinh viên (Visual Studio, SQL Server Dev).
- **Chi phí vận hành:** Tận dụng các gói Free-tier của Cloud (Azure/AWS) hoặc chi phí thấp cho VPS và Domain trong giai đoạn demo.

**Nhân sự lập trình:** 80.000.000 - 150.000.000 VNĐ.

**Thiết kế UI/UX:** 10.000.000 - 20.000.000 VNĐ

**Chi phí hạ tầng kỹ thuật (Infrastructure Costs)**

- **Tên miền (Domain):** 300.000 - 750.000 VNĐ/năm.
- **Máy chủ/Hosting (Cloud Server):** 4 - 12 triệu/năm
- **Chứng chỉ bảo mật (SSL):** 200.000 - 5.000.000 VNĐ/năm

## 1.5 Khả thi về Vận hành (Operational Feasibility)

- **Người dùng cuối (Khách hàng):** Giao diện thân thiện, dễ sử dụng nhờ Ant Design, quy trình mua hàng đơn giản hóa, hoạt động tốt trên đa thiết bị.
- **Người quản trị (Admin):** Hệ thống Dashboard cung cấp đầy đủ công cụ quản lý sản phẩm, đơn hàng và báo cáo thống kê.
- **Quy trình:** Hệ thống được thiết kế bám sát quy trình nghiệp vụ mua bán thực tế, không gây khó khăn khi chuyển đổi từ quy trình thủ công sang tự động.

## 1.6 Khả thi về Thời gian (Schedule Feasibility)

- **Tổng thời gian:** Cố định theo lịch trình: 3 - 4 tháng
- **Các mốc chính:**
  - Phân tích & Thiết kế: Tuần 1-4.
  - Phát triển (Coding): Tuần 5-12 (Backend & Frontend song song).
  - Kiểm thử (Testing) & Fix bug: Tuần 13-14.
  - Viết báo cáo & Triển khai (Deploy): Tuần 15-16.

## 1.7 Khả thi về Pháp lý (Legal Feasibility)

- **Bản quyền phần mềm:** Sử dụng các thư viện và Framework mã nguồn mở (MIT License), tuân thủ đúng quy định sử dụng.
- **Dữ liệu:** Cam kết bảo mật thông tin khách hàng (mật khẩu mã hóa, không chia sẻ thông tin cá nhân), tuân thủ các quy tắc cơ bản về thương mại điện tử.

## 1.8 Rủi ro & Giải pháp (Risk Analysis)

- **Rủi ro kỹ thuật:** Khó khăn khi triển khai Microservices (phức tạp trong giao tiếp giữa các service). -> **Giải pháp:** Sử dụng thiết kế đơn giản hóa ban đầu, focus vào các luồng chính, tham khảo kỹ tài liệu Microsoft.
- **Rủi ro thời gian:** Chậm tiến độ do phát sinh lỗi. -> **Giải pháp:** Ưu tiên phát triển các tính năng cốt lõi (Core features) trước, tính năng phụ ("Nice to have") để lại sau.

## 2. System Request Form

Field	Description
Project Name	Technology Product Sales System
Sponsor	School of Computing – SE Department
Business Need	Tạo nền tảng marketplace đa người bán.
Requirements	Seller đăng bán, Buyer mua hàng, thanh toán, vận chuyển
Business Value	Tăng doanh thu, mở rộng seller, chuẩn hóa quy trình
Constraints	Phụ thuộc API payment/shipping, hệ thống phải ổn định 24/7

## 3. Feasibility Matrix

Feasibility	Rating	Justification
Technical	Medium	Tích hợp hệ thống ngoài + dữ liệu lớn
Operational	High	Người dùng quen mô hình Shopee/Lazada
Economic	Medium–High	Chi phí moderate, lợi nhuận lâu dài
Schedule	Medium	Nhiều module, cần 3–4 tháng cho bản MVP

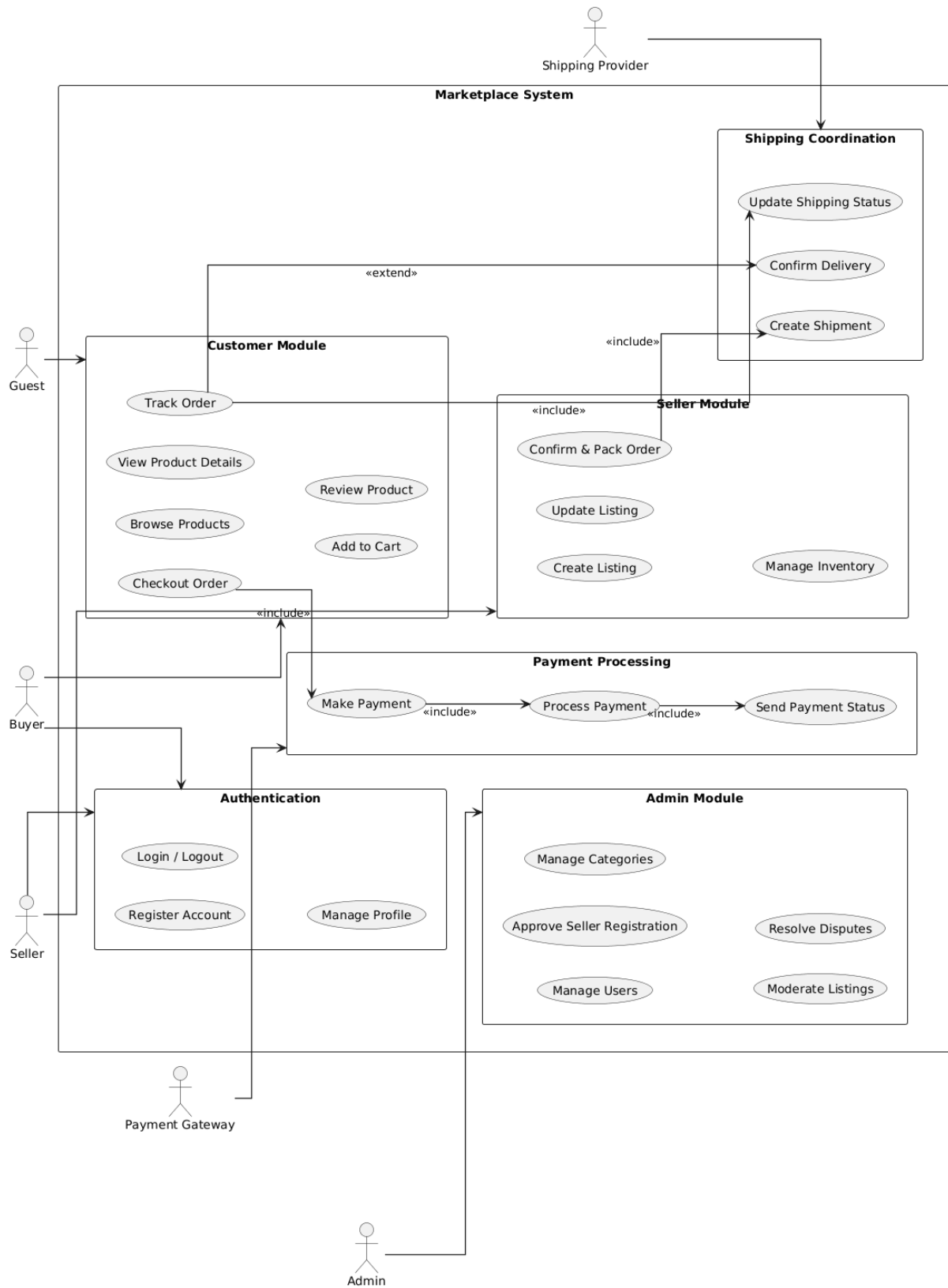
# USE CASE MODELING

## 1. Actor & Use Case Table

Actor	Use Cases
Guest	View Homepage, Search Product, View Product Details, Register
Buyer	Login, Manage Profile, Add to Cart, Checkout, Make Payment, Track Order, Review Product
Seller	Login, Manage Store, Create Listing, Update Product, Manage Inventory, Confirm Order, Pack Order
Admin	Manage Users, Approve Seller Registration, Manage Categories, Moderate Listings, Resolve Disputes
Payment Gateway	Process Payment, Validate Transaction, Send Payment Status
Shipping Provider	Create Shipping Label, Pick Up Order, Update Tracking Status, Mark Delivered

## 2. Use Case Diagram

Technology Product Sales System - Use Case Diagram



### 3. Fully Dressed Use Cases

#### UC01 – Checkout Order

**Scope:** Marketplace System

**Level:** User Goal

**Primary Actor:** Buyer

**Supporting Actors:** Payment Gateway, Seller, Shipping Provider

**Preconditions:**

- Buyer đã login
- Cart không rỗng

**Trigger:** Buyer nhấn “Checkout”

**Main Success Scenario:**

1. Buyer xem lại giỏ hàng
2. Buyer chọn địa chỉ giao hàng
3. Buyer chọn phương thức thanh toán
4. System gửi Payment Request — Payment Gateway
5. Payment Gateway verify giao dịch
6. System tạo Order + OrderItems
7. System thông báo Seller
8. System gửi Shipping Request — Shipping Provider
9. Shipping Provider trả Tracking Number
10. System hiển thị “Checkout Successful”

**Extensions:**

- 4a. Payment error → Order = PaymentFailed
- 8a. Shipping error → Order = PendingShipment

**Postconditions:**

- Order trạng thái = “Paid”
- Stock giảm
- TrackingNumber gán cho Order

## UC02 – Create Product Listing

**Primary Actor:** Seller

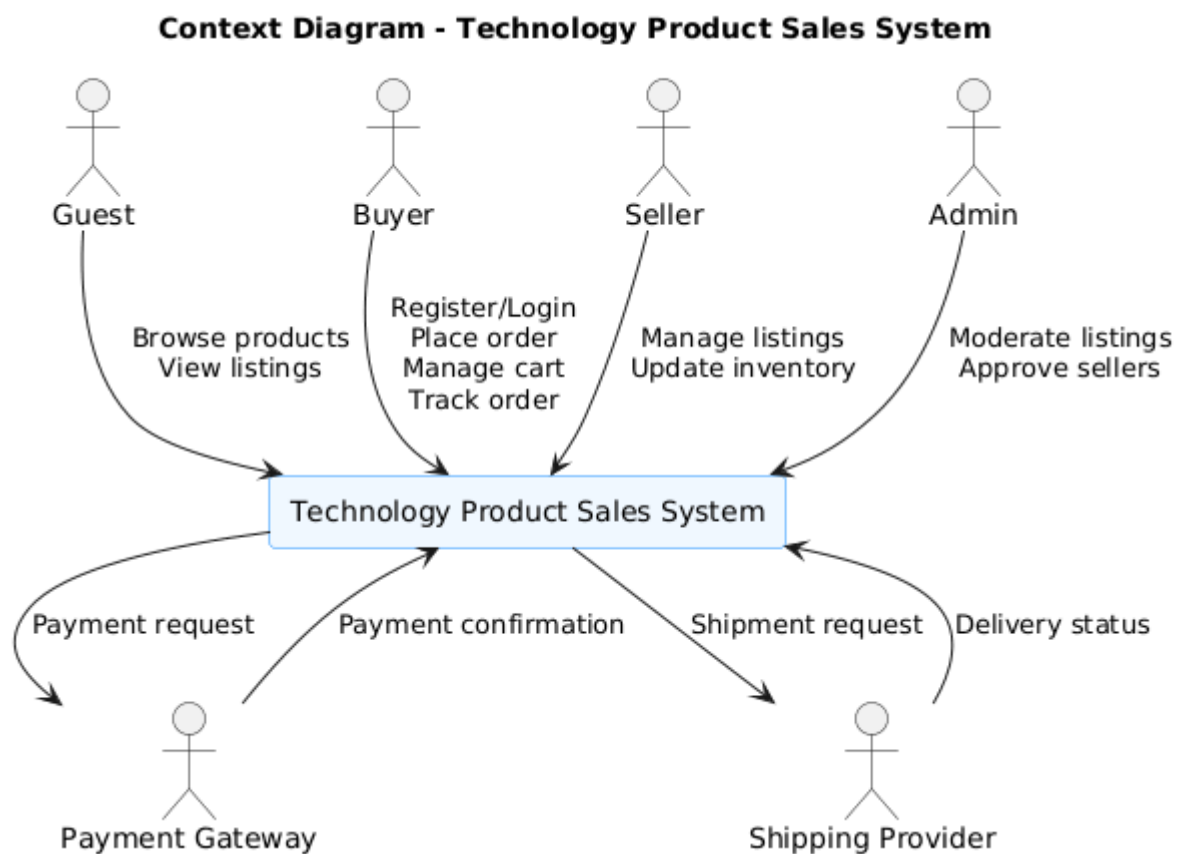
**Preconditions:** Seller đã login

**Main Scenario:**

1. Seller chọn “Add Product”
2. Nhập thông tin sản phẩm
3. System validate
4. Lưu sản phẩm vào Product DB
5. Hiển thị trên marketplace

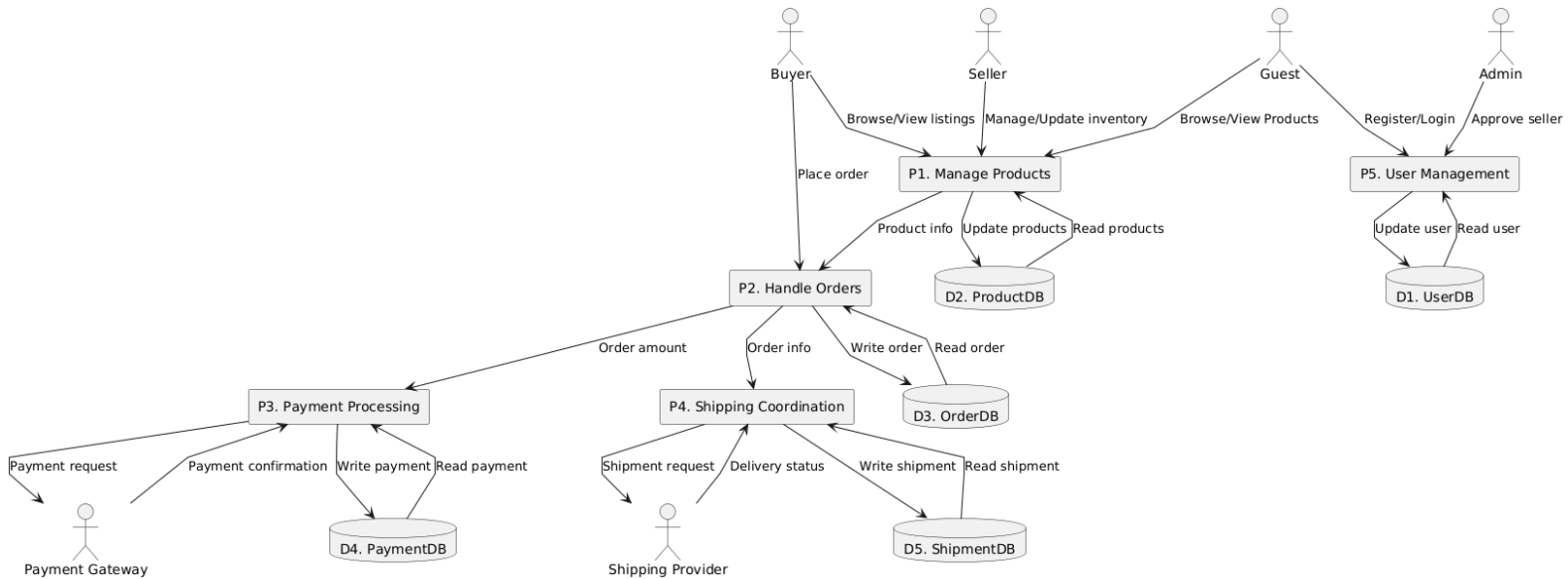
# DFD + DATA DICTIONARY

## 1. Context Diagram



## 2. Level-0 DFD

Level-0 DFD - Technology Product Sales System



### 3. Full Data Dictionary

#### A. Data Elements

(đầy đủ mẫu)

#### B. Data Structures

(Products, Orders, OrderItems...)

#### C. Data Stores

(tbl Users, Products, Orders...)

#### D. Data Flows

(CheckoutRequest, PaymentRequest, PaymentResult...)

#### E. Processes

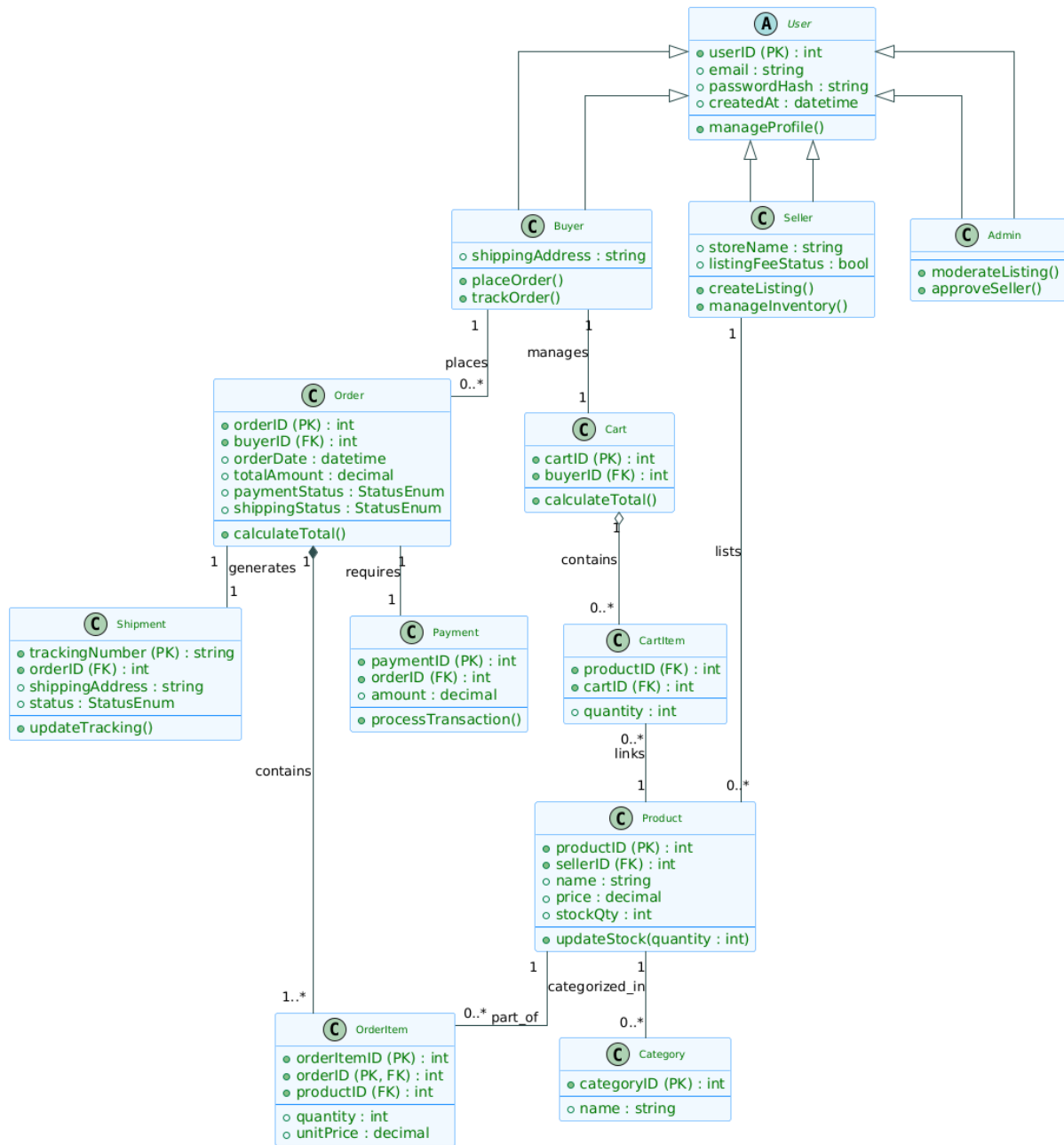
(P1–P5)

## CLASS DIAGRAM + ERD

### 1. UML Class Diagram



UML Class Diagram - Technology Product Sales System



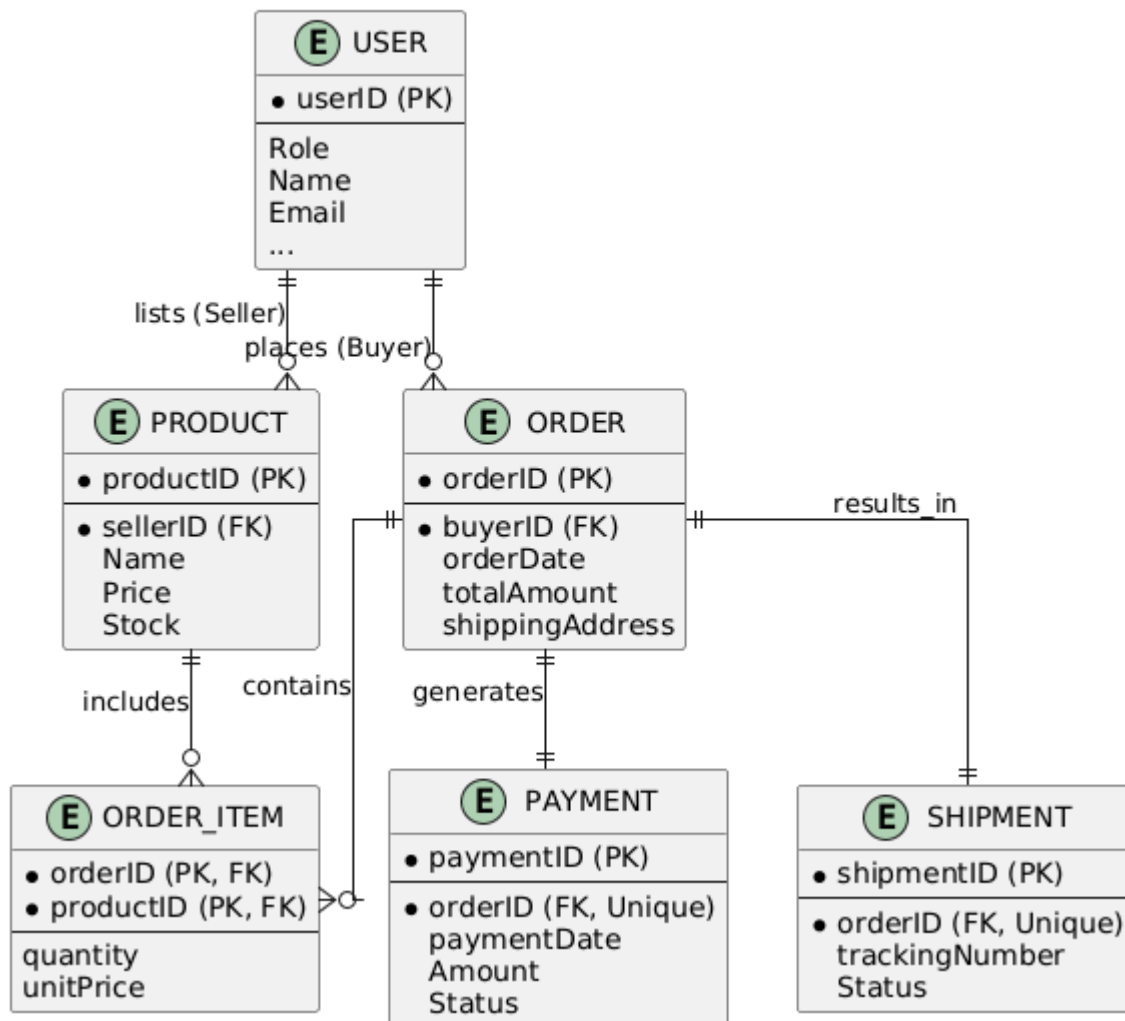
Bao gồm:

- User (Buyer, Seller, Admin kế thừa)
- Product
- Cart → CartItem
- Order → OrderItem
- Payment

- Shipment

## 2. ERD (Entity Relationship Diagram)

**ERD - Technology Product Sales System (Crow's Foot)**



### 2.1 Entities & Attributes

#### User

- userID (PK)
- fullName
- email
- password
- role (Buyer/Seller/Admin)

#### Product

- productID (PK)
- sellerID (FK → User.userID)
- name
- description
- price
- stockQty
- categoryID (FK)

### **Category**

- categoryID (PK)
- categoryName

### **Order**

- orderID (PK)
- buyerID (FK → User.userID)
- totalAmount
- paymentStatus
- shippingStatus
- createdAt

### **OrderItem**

- orderItemID (PK)
- orderID (FK → Order.orderID)
- productID (FK → Product.productID)
- quantity

- unitPrice

### Payment

- paymentID (PK)
- orderID (FK → Order.orderID)
- status
- transactionTime

### Shipment

- trackingNumber (PK)
- orderID (FK → Order.orderID)
- status
- updatedAt

---

## 2.2 Relationship Summary

Relationship	Type	Description
User – Product	1:N	Seller đăng nhiều Product
Category – Product	1:N	Mỗi sản phẩm thuộc 1 category
User – Order	1:N	Buyer đặt nhiều Order
Order – OrderItem	1:N	Một Order có nhiều OrderItems
Product – OrderItem	1:N	Một Product xuất hiện trong nhiều OrderItem
Order – Payment	1:1	Mỗi Order có 1 Payment record
Order – Shipment	1:1	Mỗi Order có 1 Shipment record

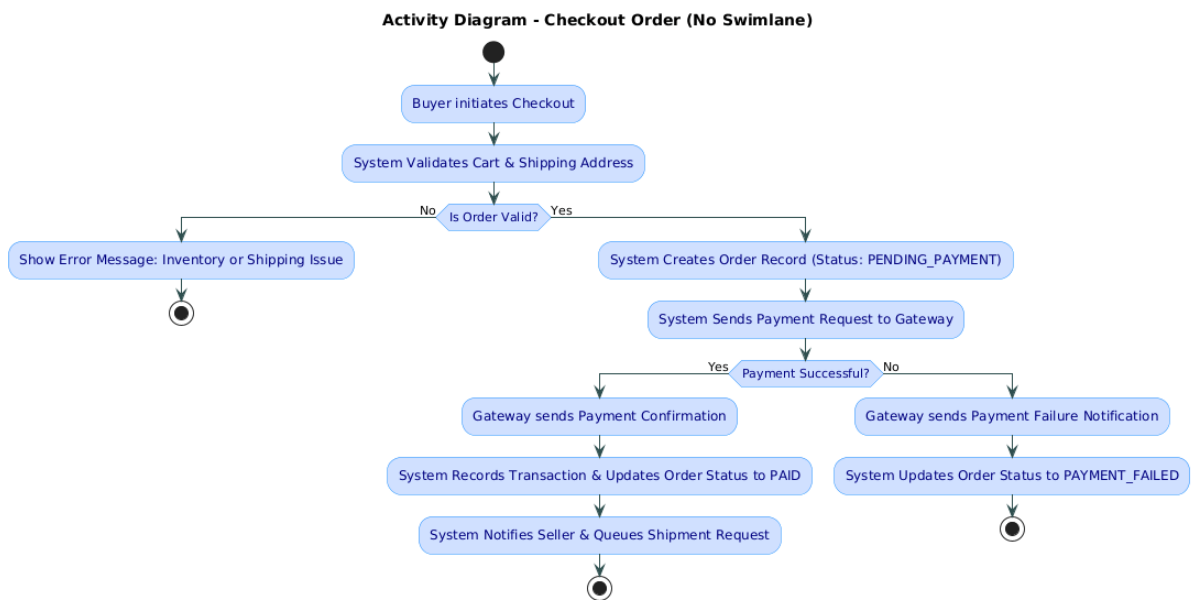
# ACTIVITY DIAGRAMS

# 1. Action Lists

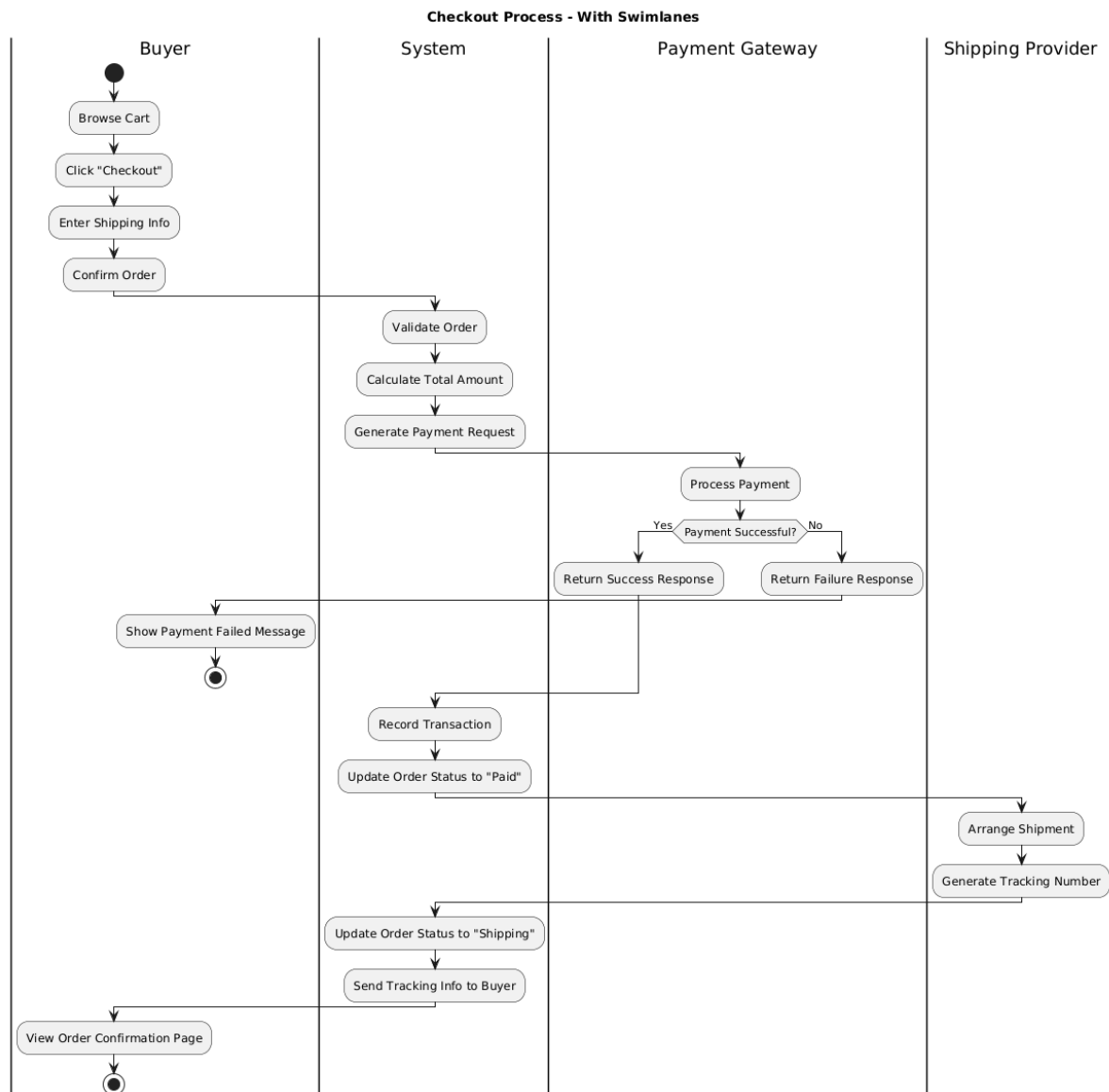
(Checkout + Shipping Flow)

- Borrow Equipment (login → search → validate → submit → approve/reject → notify).
- Return Equipment (bring item → staff check → system update → availability update → maintenance if needed).

## 2. Activity Diagram – Checkout (No Swimlane)

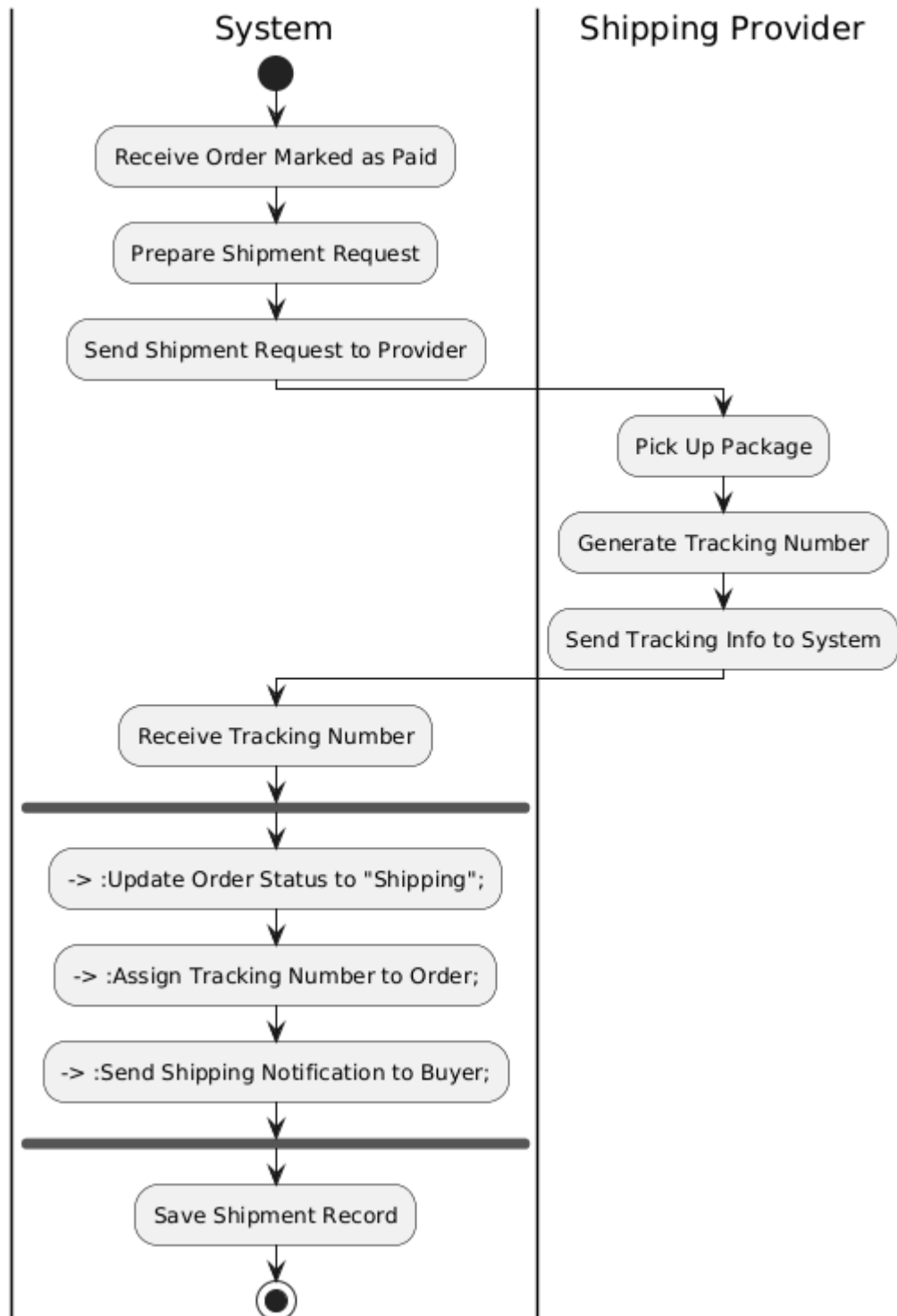


### 3. Activity Diagram – Checkout (With Swimlanes)



### 4. Activity Diagram – Order Fulfillment (Shipping)

### Order Fulfillment - Shipping Process (With Swimlanes)



## 5. Reflection

Activity Diagram giúp tôi hiểu rõ luồng xử lý dynamic của hệ thống marketplace, thay vì chỉ nhìn cấu trúc static từ Class Diagram. Các bước diễn ra tuần tự và các quyết định trở nên trực quan, nhất là khi phối hợp nhiều bên như Buyer, Seller, Payment Gateway và Shipping

Provider. Swimlanes làm rõ trách nhiệm của từng actor trong từng phase của quy trình. Ví dụ, phân thanh toán chỉ thuộc Payment Gateway, tránh nhầm lẫn với Seller. Trong quy trình fulfillment, luồng song song là cần thiết để cập nhật trạng thái đơn hàng và tạo vận đơn đồng thời, giảm thời gian xử lý và tăng tính real-time của hệ thống. Việc mô hình hóa trước giúp giảm rủi ro sai sót khi chuyển sang thiết kế API hoặc coding sau này.