

ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



LAP 2 - PHÂN TÍCH THIẾT KẾ HỆ

SOFTWARE TESTING

Tác giả:

Võ Kiều Anh - 3122411009

Đặng Nguyễn Tâm Như - 3122411142

Phạm Nguyễn Minh Châu - 3122411021

Lớp: DCT122C5

Thành phố Hồ Chí Minh, tháng 9 năm 2025

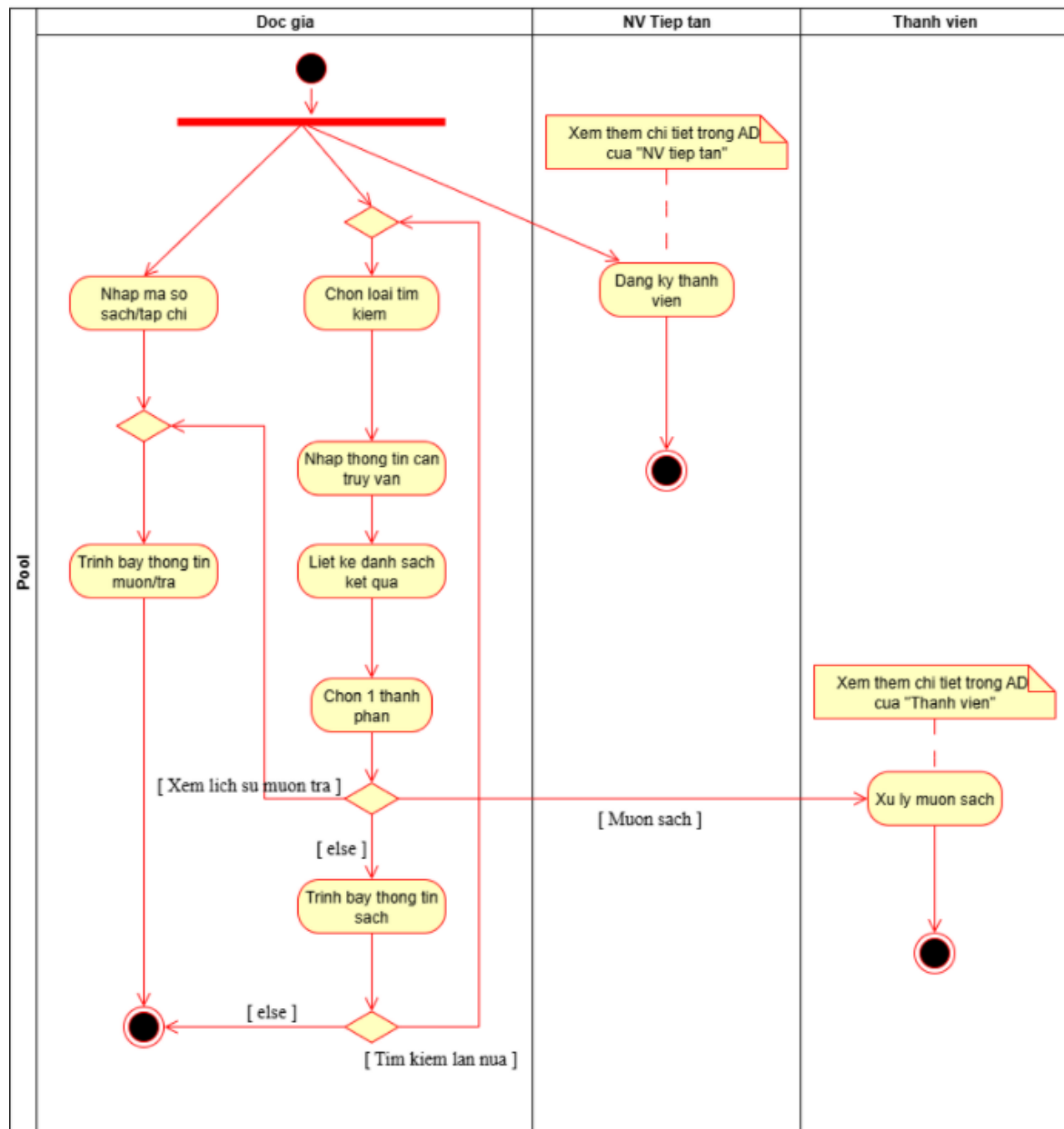
Mục lục

Phân công nhiệm vụ:	3
1. Quy trình nghiệp vụ (Độc giả, NV Tiếp tân, Thành viên).....	4
2. Mô hình khái niệm.....	5
3. Business use case.....	6
4. Sơ đồ hoạt động và tương tác	12
5. Lược đồ lớp cho bài toán Quản lý thư viện.....	16
6. Lược đồ.....	17
7. Phân tích dữ liệu cho ABC Bank.....	18
7.1. Thiết kế Cơ sở dữ liệu ở mức Khái niệm, Luận lý và Vật lý.....	18
7.2. Query: Insert, Update, Delete, Select.....	20
8. BÀI TẬP ỨNG DỤNG.....	23
8.1 Yêu cầu chức năng và phi chức năng của hệ thống CoolStore	24
8.2. Quy trình nghiệp vụ cho website CoolStore	25
1. Danh mục sản phẩm (Product Catalog)	25
2. Quy trình Giỏ hàng (Shopping Cart).....	25
3. Quy trình Thanh toán (Payment Process)	26
6. Quy trình Kiểm soát truy cập (Access Control).....	27
8.3. Xây dựng use case diagram cho website CoolStore.	28
8.4. Mô tả chi tiết một Use Case	33
8.5. Sơ đồ Activity Diagram - mô tả quy trình mua hàng.....	35
8.6. Thiết kế sơ đồ Class Diagram cho module Giỏ hàng.....	37
8.7. Thiết kế giao diện màn hình cho chức năng “Quản lý giỏ hàng”	37
8.8. Đề xuất cách tổ chức kiến trúc phần mềm cho hệ thống CoolStore theo mô hình kiến trúc ba lớp (3-tier architecture)	42
8.9. Kiểm thử (test cases) cho chức năng “Thêm sản phẩm vào giỏ hàng”.....	44

Phân công nhiệm vụ:

Bài tập	Tâm Như	Minh Châu	Kiều Anh
1	X		
2	X		
3			X
4		X	
5		X	
6			X
7.1	X		
7.2a			X
7.2b			X
7.2c		X	
7.2d		X	
7.2e	X		
7.2f	X		
8.1	X	X	X
8.2	X	X	X
8.3	X	X	X
8.4			X
8.5	X		
8.6		X	
8.7	X	X	X
8.8	X	X	X
8.9	X	X	X

1. Quy trình nghiệp vụ (Độc giả, NV Tiếp tân, Thành viên)



Mô tả hoạt động trong sơ đồ

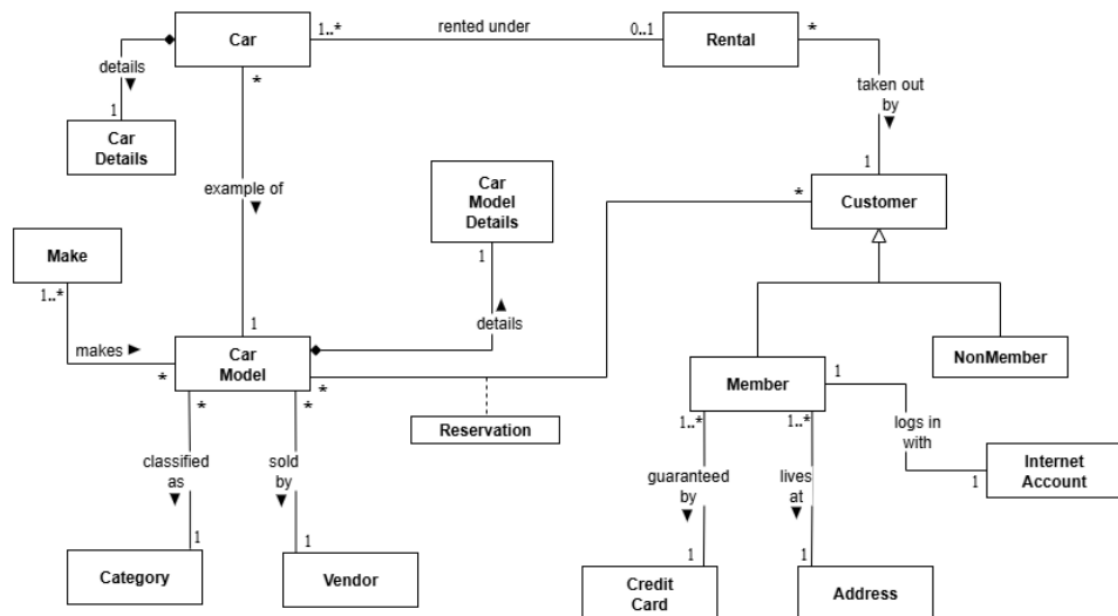
- Độc giả có 2 lựa chọn ban đầu:
 - Nhập mã số sách/tạp chí, sau đó hệ thống trình bày thông tin mượn/trả
 - Chọn loại tìm kiếm, nhập thông tin cần truy vấn, sau đó hệ thống sẽ liệt kê danh sách kết quả
- Từ danh sách kết quả:
 - Người dùng chọn 1 thành phần
 - Nếu chọn xem lịch sử mượn/trả thì hệ thống sẽ hiển thị lịch sử
 - Nếu chọn xem thông tin sách thì hệ thống trình bày thông tin chi tiết
 - Nếu muốn tìm kiếm lần nữa thì hệ thống quay lại bước tìm kiếm

- Nếu độc giả muốn mượn sách, quy trình sẽ rẽ nhánh:
 - Nếu chưa là thành viên: sẽ sang luồng NV Tiếp tân để đăng ký thành viên
 - Nếu là thành viên rồi: sang luồng Thành viên để thực hiện xử lý mượn sách

Ý nghĩa sơ đồ

- Pool: Đại diện cho toàn bộ hệ thống thư viện
- Swimlane: Phân công rõ trách nhiệm:
 - Độc giả: tìm kiếm, xem thông tin sách
 - NV Tiếp tân: đăng ký thành viên mới
 - Thành viên: mượn sách
- Luồng công việc: Thể hiện rõ quy trình từ khi độc giả tra cứu cho tới khi thực hiện mượn sách
- Điều kiện rẽ nhánh: giúp phân biệt giữa các lựa chọn (xem lịch sử, xem thông tin, mượn sách...)
- Ý nghĩa tổng quan: Đây là sơ đồ dùng để mô tả nghiệp vụ thư viện, minh họa cách độc giả tương tác với hệ thống và cách nhân viên hoặc thành viên tham gia vào quy trình

2. Mô hình khái niệm



Mô tả nghiệp vụ cơ bản thông qua mô hình:

- Quản lý xe

- Công ty cho thuê lưu trữ thông tin từng xe cụ thể (Car) trong Car Details
- Mỗi xe thuộc về một mẫu xe (Car Model), ví dụ Toyota Vios 2021, Honda Civic 2020
- Mẫu xe được sản xuất bởi một hãng (Make) được phân loại theo Category và được cung cấp bởi một Vendor
- Thông tin chi tiết của mẫu xe (Car Model Details) được lưu riêng để so sánh và tham khảo

➔ Hệ thống đảm bảo vừa quản lý được xe thực tế (cho thuê từng chiếc cụ thể) vừa quản lý được thông tin chung của mẫu xe

- Quản lý khách hàng

- Customer có thể là Member hoặc NonMember
 - Member: Khách hàng đăng ký làm thành viên sẽ có thông tin địa chỉ (Address), có thể gắn nhiều Credit Card để bảo đảm giao dịch
 - NonMember: Khách hàng không đăng ký thành viên thì chỉ thuê xe online thông qua Internet Account

➔ Hệ thống phân biệt được khách hàng thân thiết (Member) và khách hàng vắng lai (NonMember), từ đó quản lý dịch vụ và quyền lợi khác nhau

- Quản lý giao dịch thuê

- Một giao dịch Rental liên kết giữa Customer và Car
- Một khách hàng có thể thực hiện nhiều Rental khác nhau, một xe có thể được thuê nhiều lần nhưng tại một thời điểm chỉ thuộc về một Rental

➔ Ghi lại toàn bộ quá trình thuê xe dùng để tính phí, quản lý tình trạng xe và lịch sử khách hàng

- Quản lý đặt chỗ

- Khách hàng có thể tạo Reservation để đặt trước một Car Model
- Việc đặt chỗ chưa gắn với xe cụ thể (Car) mà chỉ gắn với mẫu xe (Car Model). Khi khách đến nhận, hệ thống sẽ phân bổ xe thuộc mẫu đó

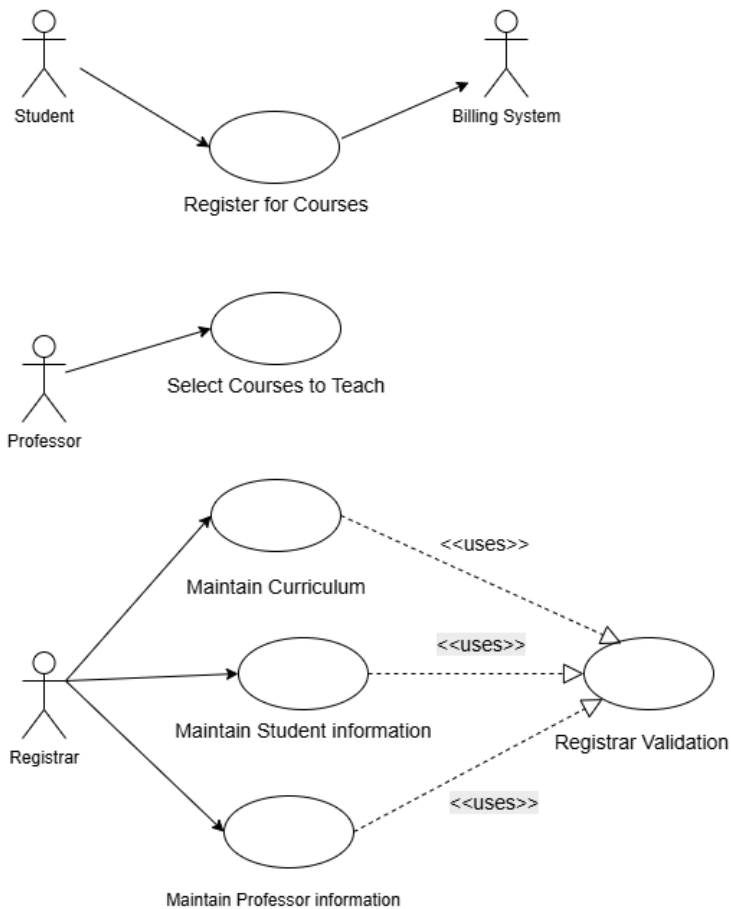
➔ Hỗ trợ khách lên kế hoạch thuê xe trước, đảm bảo trải nghiệm dịch vụ tốt hơn và tối ưu sử dụng xe

- Đảm bảo giao dịch (Member – Credit Card)

- Với khách hàng là thành viên, hệ thống yêu cầu thông tin Credit Card để bảo đảm khả năng thanh toán

➔ Điều này giảm rủi ro khi cho thuê xe và đảm bảo lợi ích cho công ty

3. Business use case



a) Xác định và mô tả các tác nhân (Actors) xuất hiện trong sơ đồ:

- **Student (Sinh viên):** Người dùng hệ thống để đăng ký môn học, tương tác với hệ thống để thực hiện việc chọn và đăng ký các khóa học.
- **Professor (Giảng viên):** Người dùng hệ thống để chọn các môn học mà họ sẽ giảng dạy.
- **Registrar (Cán bộ phòng đào tạo):** Quản trị và duy trì dữ liệu về chương trình học, thông tin sinh viên, thông tin giảng viên.
- **Billing System (Hệ thống thanh toán học phí):** Hệ thống bên ngoài được tích hợp, đảm nhận việc xử lý thanh toán sau khi sinh viên đăng ký môn học.

b) Liệt kê và giải thích các trường hợp sử dụng (Use cases) được thể hiện trong sơ đồ:

- **Register for Courses (Sinh viên đăng ký môn học):** Sinh viên chọn các môn muốn học và hệ thống gửi dữ liệu sang Billing System để xử lý học phí.
- **Select Courses to Teach (Giảng viên chọn môn để giảng dạy):** Giảng viên chọn từ danh sách môn học mà họ có thể đảm nhận trong học kỳ.

- **Maintain Curriculum (Quản lý chương trình học):** Cán bộ đào tạo cập nhật, chỉnh sửa hoặc bổ sung thông tin chương trình đào tạo (môn học, tín chỉ...).
- **Maintain Student Information (Quản lý thông tin sinh viên):** Cập nhật hồ sơ sinh viên (thông tin cá nhân, kết quả học tập...).
- **Maintain Professor Information (Quản lý thông tin giảng viên):** Cập nhật hồ sơ giảng viên (thông tin cá nhân, môn học giảng dạy...).
- **Registrar Validation (Xác thực dữ liệu của phòng đào tạo):** Đảm bảo dữ liệu cập nhật (curriculum, student info, professor info) hợp lệ trước khi lưu vào hệ thống chính thức.

c) Phân tích các mối quan hệ giữa các use case (bao gồm cả mối quan hệ «uses» hoặc «include»):

Maintain Curriculum → Registrar Validation

- Quan hệ «uses» : include.
- Mỗi khi có bất kỳ thay đổi nào trong chương trình học, hệ thống đều phải gọi đến chức năng xác thực của Registrar.

Maintain Student Information → Registrar Validation

- Quan hệ «uses» : include
- Khi cập nhật thông tin sinh viên, hệ thống cần chạy bước xác thực dữ liệu.

Maintain Professor Information → Registrar Validation

- Quan hệ «uses» : include
- Khi cập nhật hồ sơ giảng viên, cũng cần xác thực dữ liệu.

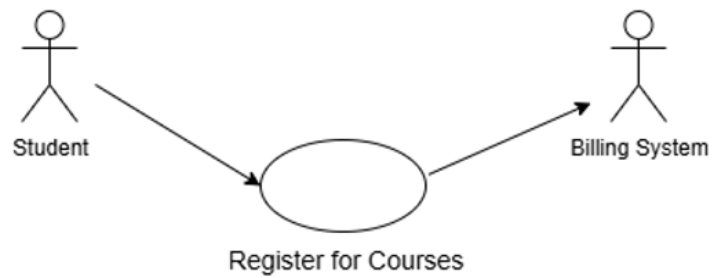
Register for Courses → Billing System

- Quan hệ kết nối (association).
- Sau khi đăng ký môn, hệ thống phải tương tác với hệ thống bên ngoài (Billing System) để xử lý học phí.

Select Course to Teach

- Hiện tại không có include/extend với Use Case khác.
- Có thể bổ sung quan hệ «**extend**» với “Maintain Curriculum” (vì giảng viên chỉ được chọn trong số môn có trong chương trình đào tạo)

d) Kịch bản chi tiết cho một trường hợp sử dụng cụ thể (ví dụ: “Register for Courses” hoặc “Maintain Student Information”):



Use Case Name:	Register for Courses	
Actor (s):	Student, billing system	
Maturity:	Focused	
Summary:	Sinh viên đăng ký các môn học mong muốn trong học kỳ.	
Basic Course of Events:	Actor Action	System Response
	1. Sinh viên nhập tài khoản và mật khẩu	
		2. Hệ thống xác thực thông tin đăng nhập E1 .
	3. Sinh viên chọn chức năng "Register for Courses".	
		4. Hệ thống hiển thị danh sách các môn học khả dụng.
	5. Sinh viên chọn một hoặc nhiều môn học. A1,A2	
		6. Hệ thống kiểm tra: sĩ số lớp còn trống, thông tin thỏa điều kiện tiên quyết. E2, E3
		7. Hệ thống ghi nhận các môn học đã chọn và gửi thông tin sang Billing

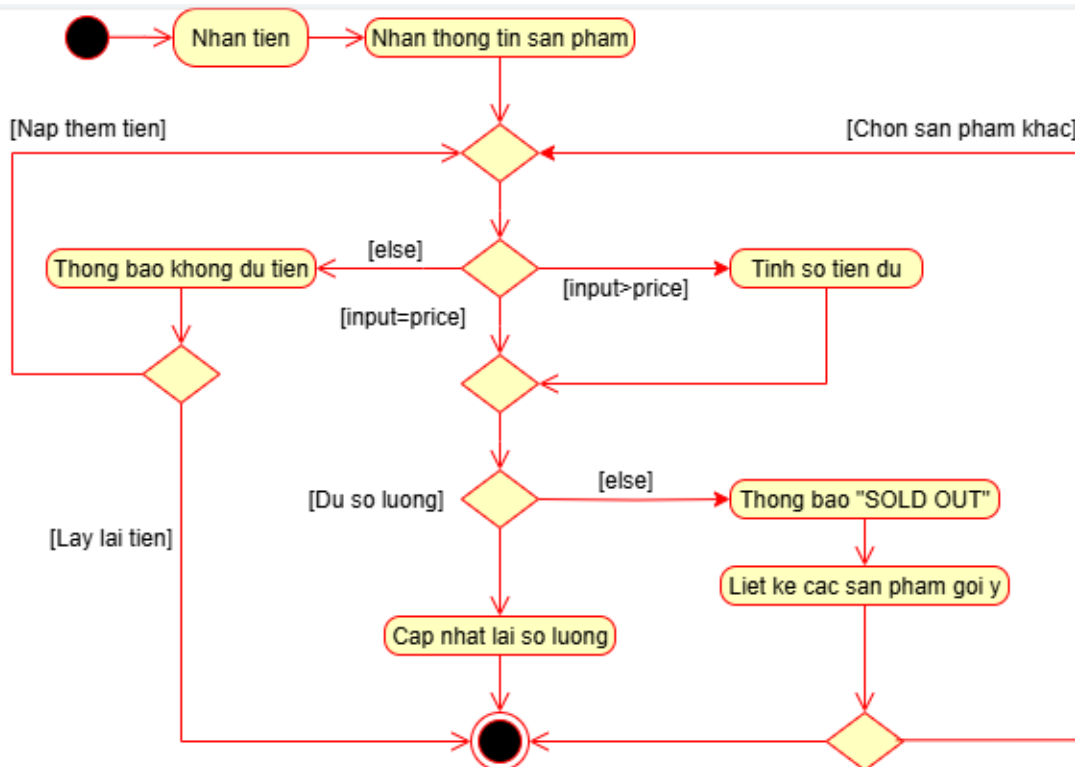
		System để tạo hóa đơn học phí. E4
		8. Hệ thống thông báo cho sinh viên “Đăng ký thành công”.
Alternative Paths:	A1.	
	A1.1 Cancel	
	Actor Action	System Response
	1. Sinh viên hủy thao tác “Cancel”.	
		2. hệ thống giữ nguyên thông tin đăng ký trước đó.
	A1.2 Reset	
	Actor Action	System Response
	1. Sinh viên chọn “Reset”.	
		2. hệ thống xóa tất cả môn vừa chọn trong phiên hiện tại và quay về bước 4.
Exception Paths:	<p>E1. UNFID không tồn tại Nếu hệ thống không tìm thấy mã số sinh viên (UNFID), hệ thống báo lỗi: “Không tìm thấy thông tin sinh viên” → quay lại bước 1.</p> <p>E2. Lỗi điều kiện môn học (Một môn học chỉ có thể được đăng ký nếu sinh viên đã hoàn thành (hoặc đang đồng thời học) các môn học yêu cầu trước đó.)</p> <ul style="list-style-type: none"> - Điều kiện tiên quyết - Đăng ký quá số môn cho phép (ví dụ > 6 môn/kỳ). - Đăng ký môn đã học qua. <p>Hệ thống hiển thị lỗi và quay lại bước 4.</p> <p>E3. Nếu môn học đã đầy, hệ thống sẽ báo lỗi và quay lại bước 4.</p>	

Extension Points:	
Preconditions:	<p>Sinh viên đã có tài khoản hợp lệ trong hệ thống.</p> <p>Sinh viên đã đăng nhập thành công.</p> <p>Học kỳ hiện tại đang mở đăng ký môn.</p>
Post Conditions:	<p>Môn học được đăng ký thành công, thông tin được lưu và hóa đơn học phí được tạo.</p> <p>Không có thay đổi nào được lưu, hệ thống báo lỗi rõ ràng cho sinh viên.</p>

e) Đề xuất cải tiến hoặc bổ sung thêm các use case mà bạn thấy cần thiết.

- **View Transcript (Xem bảng điểm)** – cho phép sinh viên xem điểm số và kết quả học tập.
- **Generate Timetable (Xem lịch học)** – sinh viên có thể xem thời khóa biểu tự động sau khi đăng ký môn.
- **Approve Courses (Phê duyệt môn học)** – giảng viên hoặc cố vấn học tập có thể phê duyệt môn học mà sinh viên chọn (nếu chính sách yêu cầu).
- **Manage Billing Records (Quản lý hóa đơn học phí)** – liên quan đến Billing System, hỗ trợ sinh viên kiểm tra tình trạng thanh toán.

4. Sơ đồ hoạt động và tương tác



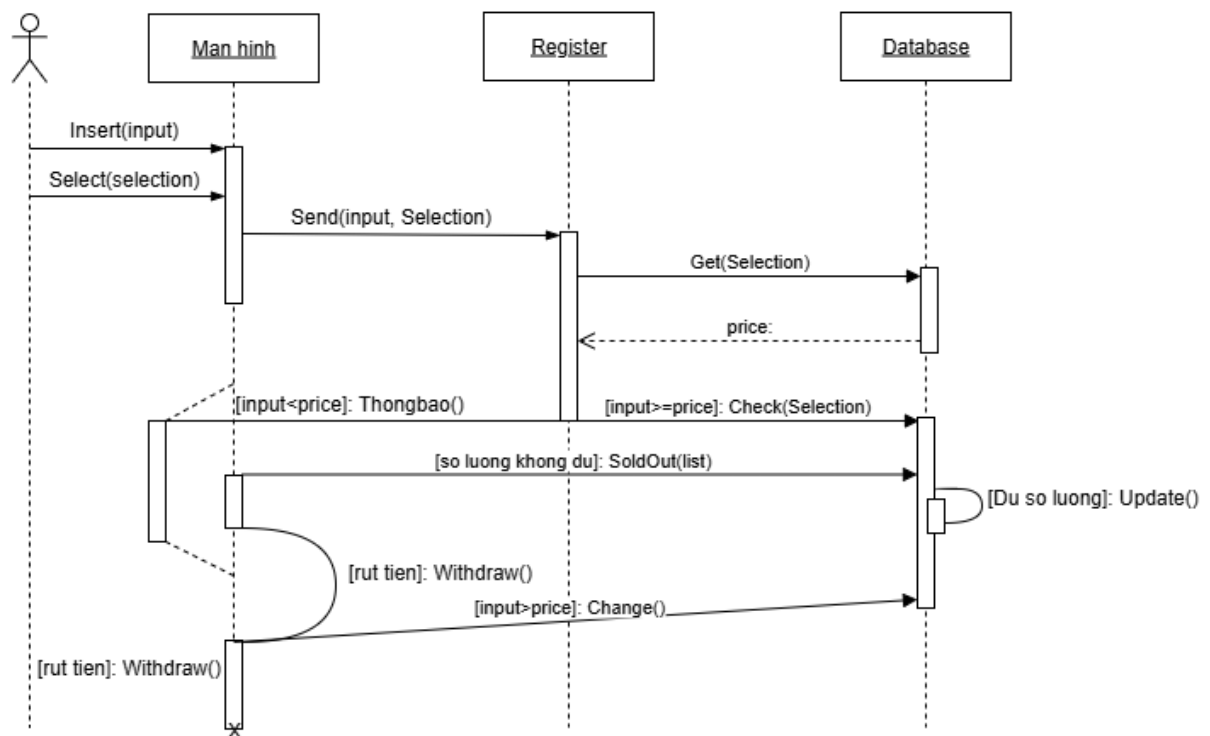
Tóm tắt ý nghĩa sơ đồ:

Mục tiêu: Mô tả quy trình xử lý của máy bán hàng tự động

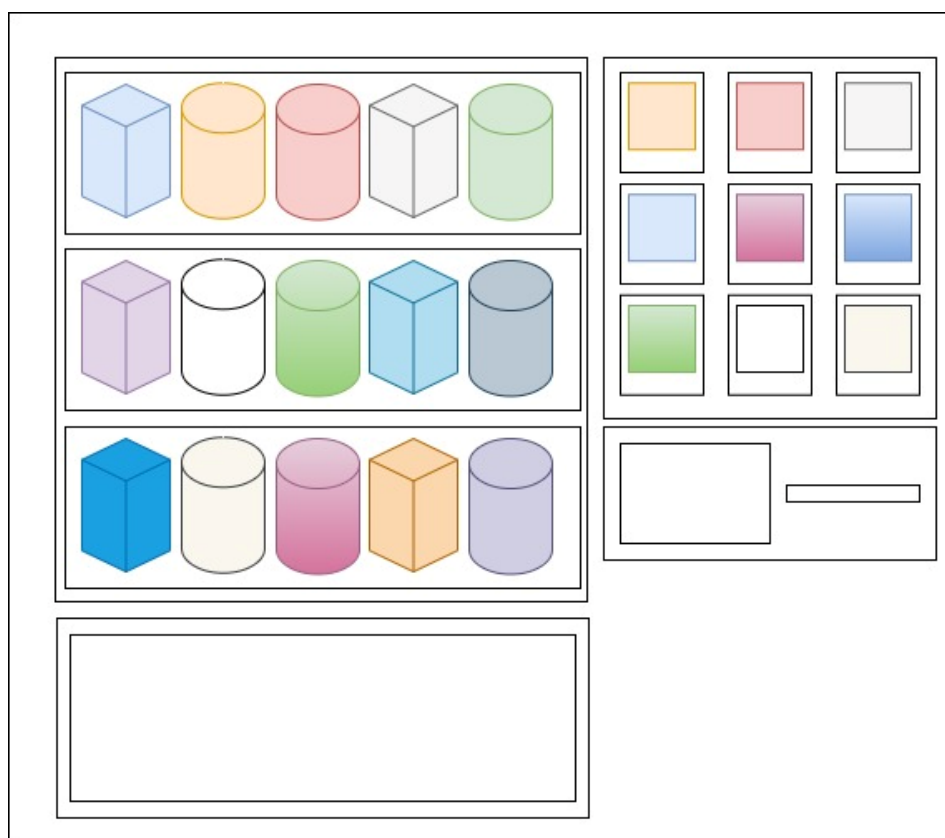
Các bước chính:

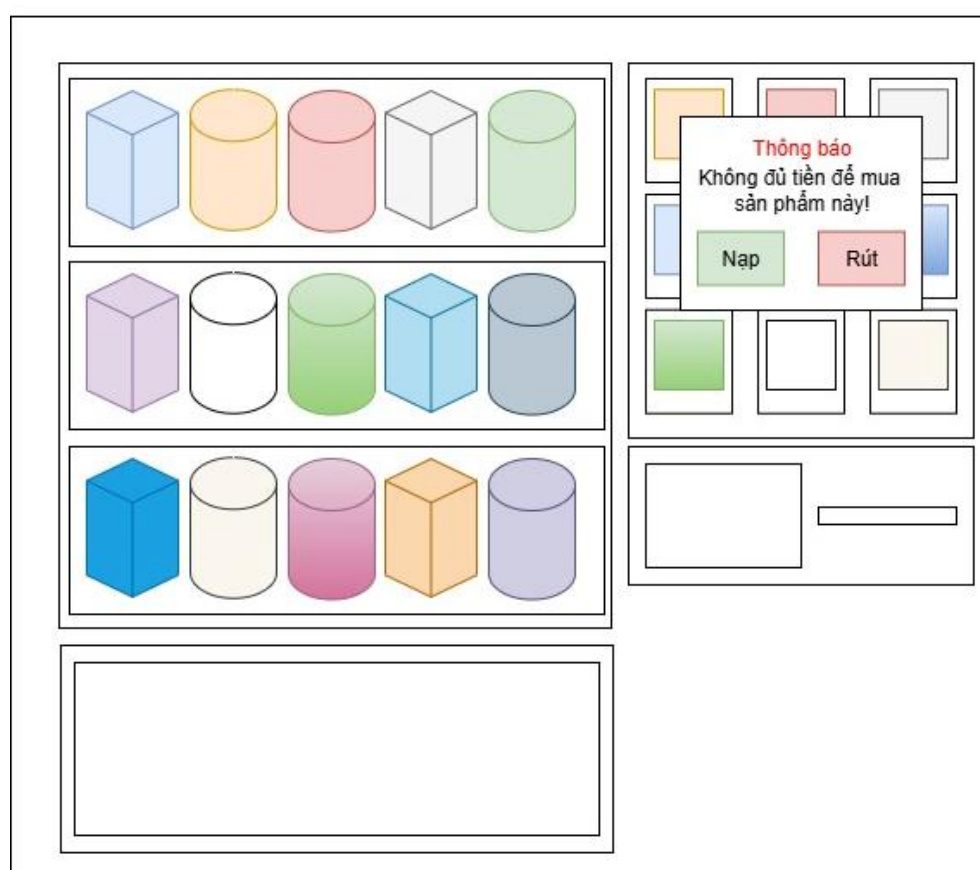
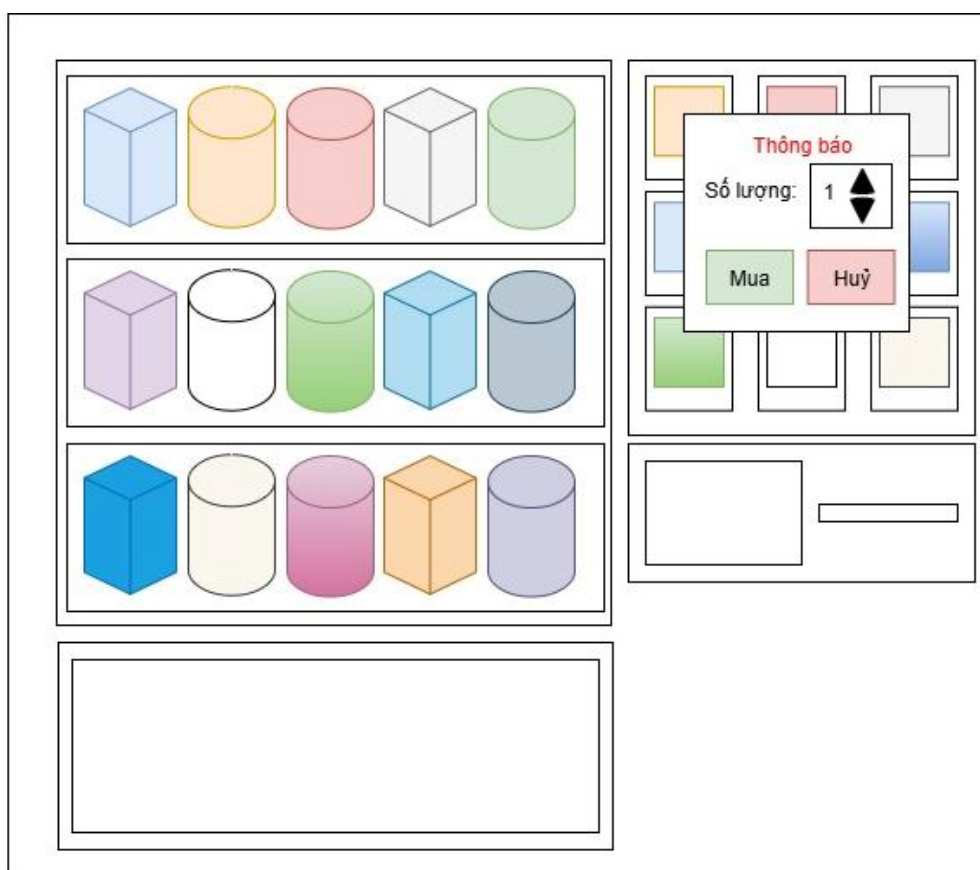
1. Người dùng **nạp tiền** và **chọn sản phẩm** .
2. Hệ thống so sánh số tiền với giá sản phẩm:
 - Nếu **không đủ tiền** → thông báo lỗi, cho phép nạp thêm hoặc rút lại tiền.
 - Nếu **dư tiền** → tính toán số tiền thừa để trả lại.
 - Nếu **đủ tiền** → kiểm tra số lượng sản phẩm.
3. Nếu còn hàng → cập nhật lại số lượng sản phẩm và hoàn tất giao dịch.
4. Nếu hết hàng → thông báo "SOLD OUT" và gợi ý sản phẩm khác.

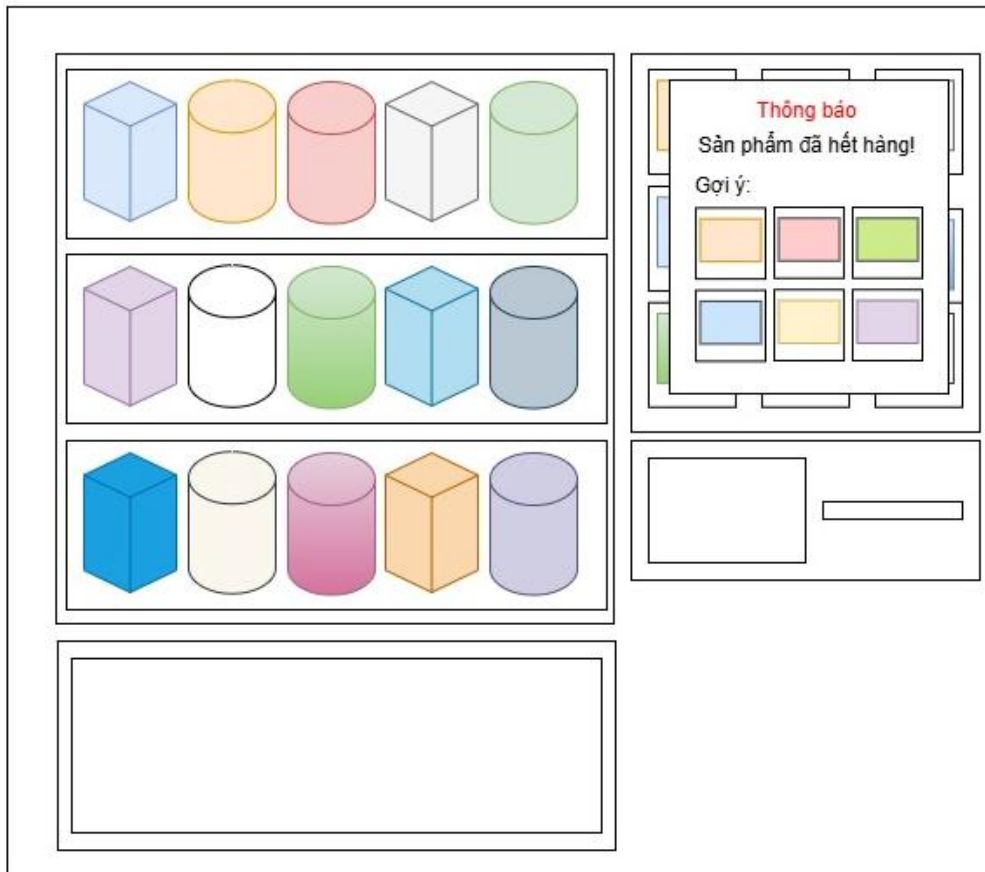
Ý nghĩa: Mô tả luồng công việc tổng thể khi mua hàng bằng máy bán hàng tự động.



Tóm tắt ý nghĩa sơ đồ:







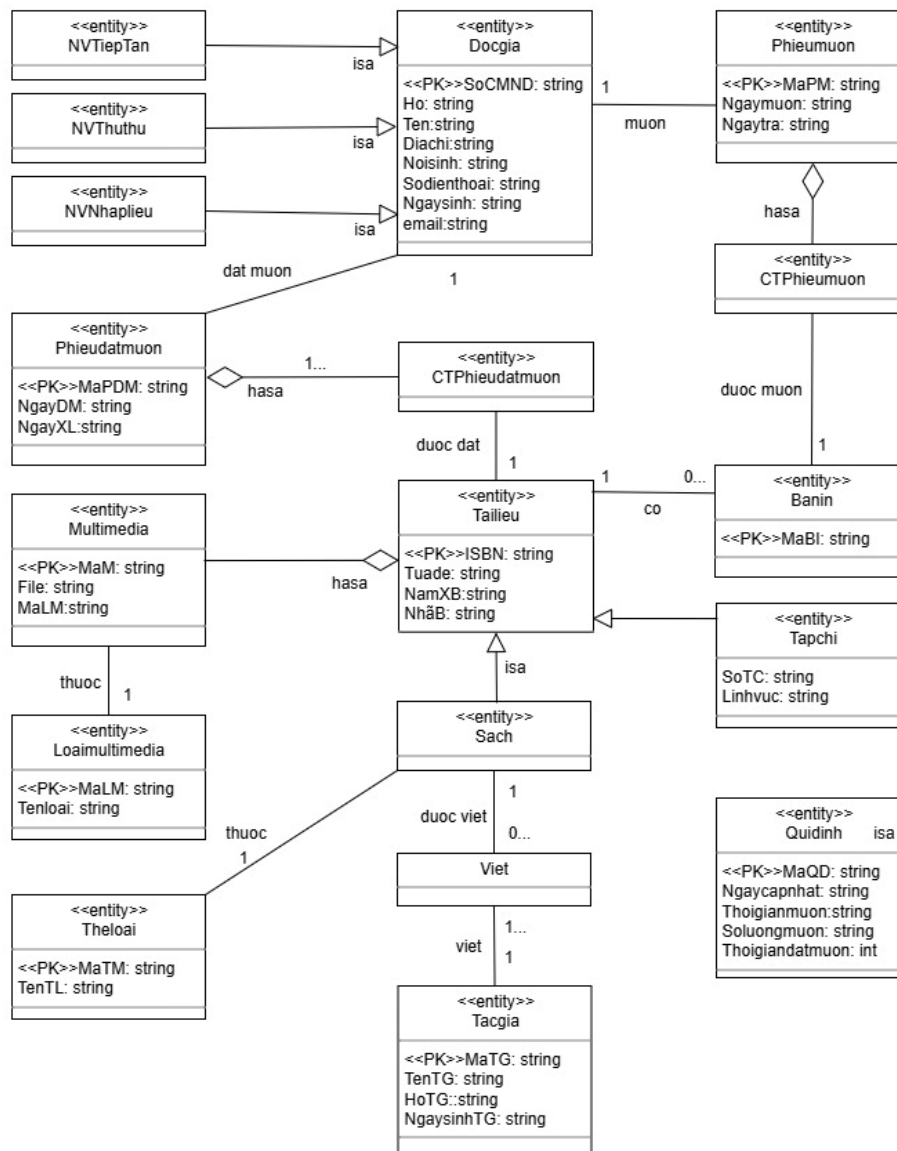
Mục tiêu: Mô tả sự tương tác giữa các đối tượng (Màn hình, Register, Database) theo thời gian.

Diễn tiến chính:

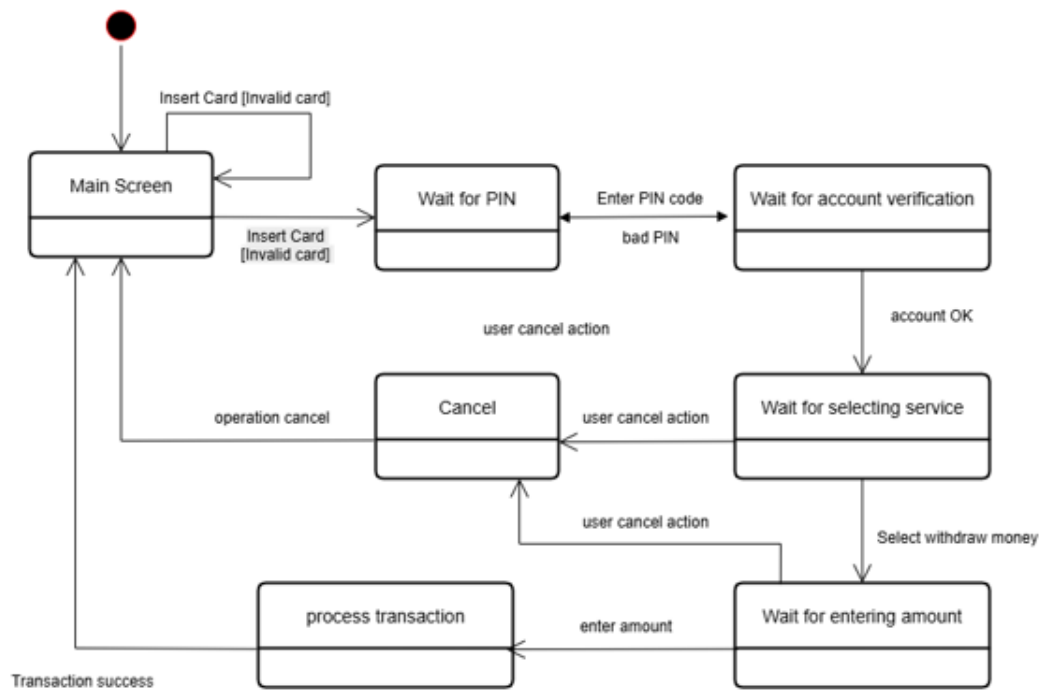
1. Người dùng **Insert(input)** (nạp tiền) và **Select(selection)** (chọn sản phẩm).
2. Màn hình gửi thông tin đến **Register** → Register lấy dữ liệu giá từ **Database**.
3. Kiểm tra điều kiện:
 - Nếu $\text{input} < \text{price}$ → thông báo không đủ tiền.
 - Nếu $\text{input} \geq \text{price}$ → Register kiểm tra số lượng sản phẩm trong Database.
 - Nếu hết hàng → trả về thông báo “SoldOut” kèm danh sách gợi ý.
 - Nếu còn hàng → cập nhật số lượng trong Database.
4. Xử lý tiền thừa (nếu có) và cho phép người dùng rút tiền.

Ý nghĩa: Mô tả thứ tự trao đổi thông điệp giữa người dùng, màn hình, bộ xử lý và cơ sở dữ liệu để hoàn tất giao dịch.

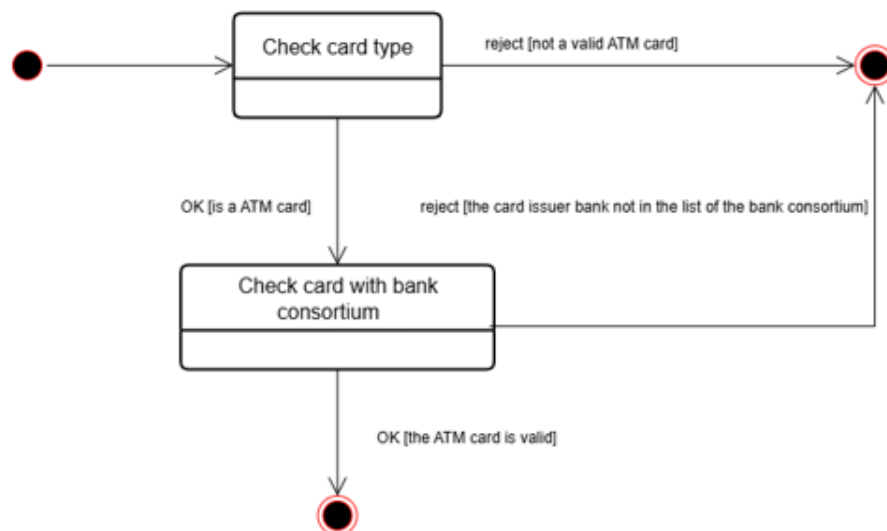
5. Lược đồ lớp cho bài toán Quản lý thư viện



6. Lược đồ



Statechart Diagram for ATM



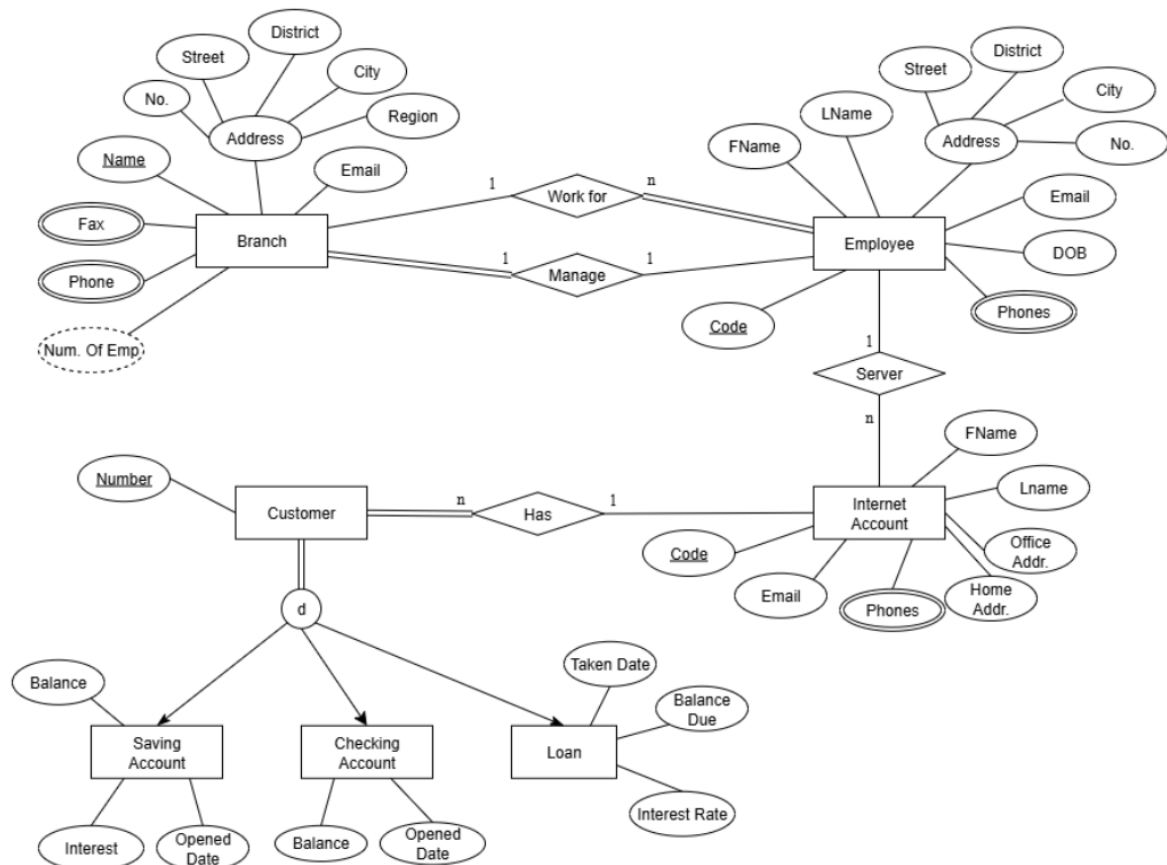
Statechart Diagram for Card controller

7. Phân tích dữ liệu cho ABC Bank

ABC Bank has many branches. Each branch has a unique name, an address (including No., Street, District, City, and Region), phone numbers, fax numbers, email, and total number of employees working there. Each branch has many employees and an employee must work at a branch. There is always one employee assigned to manage that branch. The employee can only manage the office to which he or she is assigned. For each employee, we need to store a unique code, first name, last name, date of birth, phone numbers, email, and home address (including No., Street, District, and City). The bank issues three different types of bank accounts for their customers. For Savings Accounts, the bank must keep track of the account's balance, interest rate, and the date the account was opened. Checking Accounts pay no interest, so the bank keeps track of just the balance and date opened. The third type of account, Loans, requires tracking the date the loan was taken, the balance due, and the interest rate of the loan. Each customer may have any number of bank accounts, and all accounts have a unique account number. The bank save the following information of each customer: a unique code, first name, last name, home address, Office address, phone numbers, email. Each customer is attended by an employee and an employee can serve many customers.

7.1. Thiết kế Cơ sở dữ liệu ở mức Khái niệm, Luận lý và Vật lý

Solution Example



- **Lược đồ CSDL quan hệ:**

BRANCH (Name, AddressNo, Street, District, City, Region, Email, MngCode)

BRANCHPHONE (BraName, Phone)

BRANCHFAX (BraName, Fax)

EMPLOYEE (Code, FName, LName, AddressNo, Street, District, City, DOB, Email, BraName)

EMPLOYEEPHONE (EmpCode, Phone)

CUSTOMER (Code, FName, LName, OfficeAddress, HomeAddress, Email, EmpCode)

CUSTOMERPHONE (CusCode, Phone)

ACCOUNT (Number, CusCode)

SAVINGACCOUNT (AccNumber, Balance, InterestRate, OpenedDate)

CHECKINGACCOUNT (AccNumber, Balance, OpenedDate)

LOAN (AccNumber, BalanceDue, InterestRate, TakenDate)

7.2. Query: Insert, Update, Delete, Select

a. You are required to insert valid and meaningful data into the database. Each table has at least 4 rows

```
-- BRANCH
INSERT INTO BRANCH VALUES
('BranchA', '12', 'Le Loi', '1', 'HCM', 'South',
'brancha@abc.com', 'N1001'),
('BranchB', '45', 'Tran Hung Dao', '3', 'Hanoi', 'North',
'branchb@abc.com', 'N1002'),
('BranchC', '78', 'Nguyen Trai', '5', 'Danang', 'Central',
'branchc@abc.com', 'N1003'),
('BranchD', '90', 'Ly Thuong Kiet', '10', 'Can Tho', 'South',
'branchd@abc.com', 'N1004');

-- BRANCHPHONE
INSERT INTO BRANCHPHONE VALUES
('BranchA', '0281234567'),
('BranchB', '0242345678'),
('BranchC', '0236365987'),
('BranchD', '0292233456');

-- BRANCHFAX
INSERT INTO BRANCHFAX VALUES
('BranchA', '0281111111'),
('BranchB', '0242222222'),
('BranchC', '0236333333'),
('BranchD', '0292444444');

-- EMPLOYEE
INSERT INTO EMPLOYEE VALUES
('N1001', 'John', 'Smith', '10', 'Nguyen Hue', '1', 'HCM',
'1985-03-20', 'john@abc.com', 'BranchA'),
('N1002', 'Anna', 'Lee', '22', 'Pho Hue', '3', 'Hanoi',
'1990-06-15', 'anna@abc.com', 'BranchB'),
('N1003', 'David', 'Tran', '30', 'Hai Ba Trung', '5',
'Danang', '1988-09-10', 'david@abc.com', 'BranchC'),
('N1080', 'Mark', 'Taylor', '40', 'Le Duan', '10', 'Can Tho',
'1992-12-05', 'mark@abc.com', 'BranchD');

-- EMPLOYEEPHONE
INSERT INTO EMPLOYEEPHONE VALUES
('N1001', '0903000001'),
('N1002', '0903000002');
```

```

('N1003', '0903000003'),
('N1080', '0903000004');

-- CUSTOMER
INSERT INTO CUSTOMER VALUES
('C2001', 'Peter', 'Johnson', 'CTY A - HCM', '12 Nguyen Trai
HCM', 'peter@xyz.com', 'N1001'),
('C2002', 'Mary', 'Nguyen', 'CTY B - Hanoi', '23 Pho Hue HN',
'mary@xyz.com', 'N1002'),
('C2003', 'Tom', 'Pham', 'CTY C - Danang', '34 Le Loi DN',
'tom@xyz.com', 'N1003'),
('C2004', 'Linda', 'Tran', 'CTY D - Can Tho', '45 Ly Tu Trong
CT', 'linda@xyz.com', 'N1080');

-- CUSTOMERPHONE
INSERT INTO CUSTOMERPHONE VALUES
('C2001', '0912000001'),
('C2002', '0912000002'),
('C2003', '0912000003'),
('C2004', '0912000004');

-- ACCOUNT
INSERT INTO ACCOUNT VALUES
('A3001', 'C2001'),
('A3002', 'C2001'),
('A3003', 'C2002'),
('A3004', 'C2003'),
('A3005', 'C2004');

-- SAVINGACCOUNT
INSERT INTO SAVINGACCOUNT VALUES
('A3001', 5000, 0.05, '2021-01-01'),
('A3002', 7000, 0.04, '2022-06-15'),
('A3003', 3000, 0.03, '2020-12-20'),
('A3004', 2000, 0.05, '2023-03-12');

-- CHECKINGACCOUNT
INSERT INTO CHECKINGACCOUNT VALUES
('A3003', 1500, '2021-05-05'),
('A3004', 2200, '2022-07-10'),
('A3005', 1800, '2023-01-01'),
('A3001', 500, '2020-03-03');

-- LOAN
INSERT INTO LOAN VALUES

```

```
('A3005', 10000, 0.07, '2023-02-01'),  
( 'A3002', 20000, 0.06, '2022-11-15' ),  
( 'A3004', 5000, 0.08, '2021-08-09' ),  
( 'A3003', 7000, 0.07, '2023-04-22' );
```

b. Change last name of an employee whose code is “N1080” to “Brown”

```
UPDATE EMPLOYEE  
SET LName = 'Brown'  
WHERE Code = 'N1080';
```

c. Delete the employee having code “N1080”. Explain what will happen to the customers who are severed by this employee

```
DELETE FROM EMPLOYEEPHONE  
WHERE EmpCode = 'N1080';  
  
DELETE FROM CUSTOMER  
WHERE EmpCode = 'N1080';  
  
DELETE FROM EMPLOYEE  
WHERE Code = 'N1080';
```

Giải thích: Nếu bạn xóa nhân viên N1080 thì mặc định SQL sẽ báo lỗi và không cho phép xóa, bởi vì vẫn còn khách hàng C2004 tham chiếu đến nhân viên đó. Nếu nhân viên bị xóa, mà trong CUSTOMER vẫn còn khách hàng tham chiếu đến nhân viên đó sẽ vi phạm ràng buộc toàn vẹn tham chiếu. Để xóa thành công, cần xử lý quan hệ khách hàng trước (chuyển hoặc xóa khách hàng).

d. Retrieve all account numbers of a customer whose name is Peter Johnson

```
SELECT A.Number  
FROM ACCOUNT A  
JOIN CUSTOMER C ON A.CusCode = C.Code  
WHERE C.FName = 'Peter' AND C.LName = 'Johnson';
```

e. Find all employees who have their birthdays in March 2010

```
SELECT *  
FROM EMPLOYEE  
WHERE YEAR(DOB) = 2010 AND MONTH(DOB) = 3;
```

f. Retrieve the total balance of all saving accounts owned by a customer named “Peter Johnson” (Supposing that he has many saving accounts)

```
SELECT SUM(S.Balance) AS TotalBalance  
FROM SAVINGACCOUNT S
```

```
JOIN ACCOUNT A ON S.AccNumber = A.Number  
JOIN CUSTOMER C ON A.CusCode = C.Code  
WHERE C.FName = 'Peter' AND C.LName = 'Johnson';
```

8. BÀI TẬP ỨNG DỤNG

Website CoolStore có kịch bản kinh doanh cơ bản cho Danh mục sản phẩm (Product Catalog), Giỏ hàng (Shopping Cart), Quy trình thanh toán (Payment Process), Tồn kho (Inventory), Đánh giá (Rating) và Kiểm soát truy cập (Access Control).

Với Danh mục sản phẩm, người mua (Buyer) có thể duyệt danh sách sản phẩm với các chức năng lọc và sắp xếp theo tên sản phẩm và giá. Người mua có thể xem chi tiết sản phẩm trên trang danh sách sản phẩm bằng cách nhấp vào sản phẩm đó. Trên trang chi tiết, người mua có thể xem tên, mô tả, số lượng tồn kho, thông tin kho như địa chỉ kho, trạng thái "sản phẩm nổi bật" (nếu có) và đánh giá. Quản trị viên (SysAdmin) trong hệ thống có thể quản lý sản phẩm và có quyền gán sản phẩm vào kho hiện có.

Với Giỏ hàng, người mua có thể mua bất kỳ sản phẩm nào trên danh sách sản phẩm thông qua nút "Mua hàng" trên mỗi sản phẩm. Ngoài ra, người mua cũng có thể mua sản phẩm tại trang chi tiết sản phẩm. Sau khi mua sản phẩm, người mua sẽ thấy các sản phẩm này trong giỏ hàng và bảng tổng hợp với thông tin cơ bản như tổng chi phí giỏ hàng, tổng chi phí sau khuyến mãi, chi phí vận chuyển, khuyến mãi tiết kiệm, tổng giá trị đơn hàng. Bất cứ khi nào người mua thêm sản phẩm hoặc xóa sản phẩm khỏi giỏ hàng, thì bảng tổng hợp và giỏ hàng sẽ được cập nhật. Sau đó, người mua có thể thực hiện quy trình thanh toán bằng cách nhấn nút Thanh toán trên trang giỏ hàng. Quản trị viên có thể xem toàn bộ giỏ hàng của bất kỳ người dùng nào để có thể kích hoạt hoặc vô hiệu hóa bất kỳ giỏ hàng không hợp lệ nào trên website CoolStore.

Với Quy trình thanh toán, sau khi người mua nhấn nút thanh toán, hệ thống sẽ bắt đầu xác thực thông tin sản phẩm, xử lý thanh toán, và sau đó gửi email cho người mua để người mua biết chuyện gì đang xảy ra.

Với Tồn kho, Quản trị viên có thể quản lý kho hàng.

Với Đánh giá, người mua có thể đánh giá bất kỳ sản phẩm nào mà cô ấy thấy tốt (thang điểm 1 -> 5 sao).

Với Kiểm soát truy cập, người mua hoặc quản trị viên có thể đăng nhập/đăng xuất hệ thống. Nếu là người mua, người đó sẽ được đưa đến trang danh mục sản phẩm, còn nếu là quản trị viên, sẽ được đưa đến trang quản trị.

Một số nhiệm vụ thiết lập ban đầu cần được thực hiện khi website CoolStore khởi động như: tạo người dùng quản trị, hai người dùng mua hàng, và dữ liệu mẫu cho sản phẩm, kho, đánh giá cho một vài sản phẩm.

8.1 Yêu cầu chức năng và phi chức năng của hệ thống CoolStore

❖ Yêu cầu chức năng của hệ thống CoolStore

- *Danh mục sản phẩm (Product Catalog)*

- Người mua duyệt sản phẩm (lọc, sắp xếp theo tên/giá)
- Người mua xem chi tiết sản phẩm (tên, mô tả, tồn kho, kho, trạng thái nổi bật, đánh giá)
- Quản trị viên quản lý sản phẩm (CRUD)
- Quản trị viên gán sản phẩm vào kho

- *Giỏ hàng (Shopping Cart)*

- Người mua thêm/xóa sản phẩm vào giỏ
- Hệ thống cập nhật giỏ hàng và bảng tổng hợp (chi phí sản phẩm, vận chuyển, khuyến mãi, tổng giá trị đơn hàng)
- Người mua nhấn "Thanh toán"
- Quản trị viên xem giỏ hàng của người dùng
- Quản trị viên có thể kích hoạt/vô hiệu hóa giỏ hàng không hợp lệ

- *Quy trình thanh toán (Payment Process)*

- Hệ thống xác thực sản phẩm trước khi thanh toán
- Xử lý giao dịch thanh toán
- Gửi email thông báo tình trạng thanh toán cho người mua

- *Tồn kho (Inventory)*

- Quản trị viên quản lý kho hàng (thêm, sửa, xóa kho, cập nhật địa chỉ, số lượng tồn)

- *Đánh giá (Rating)*

- Người mua đánh giá sản phẩm (1-5 sao)

- *Kiểm soát truy cập (Access Control)*

- Người dùng (Buyer) đăng ký, đăng nhập, đăng xuất
- Nếu Buyer thì chuyển tới trang Product Catalog
- Nếu Admin thì chuyển tới trang quản trị

❖ Yêu cầu phi chức năng

- Hiệu năng: Hệ thống xử lý nhanh các thao tác duyệt, tìm kiếm, giỏ hàng, thanh toán
- Bảo mật: Kiểm soát truy cập theo vai trò (Buyer/Admin), bảo mật thông tin người dùng và giao dịch

- Khả năng mở rộng: Có thể mở rộng thêm nhiều sản phẩm, kho, người dùng, tính năng khuyến mãi
- Khả năng sử dụng (Usability): Giao diện thân thiện, dễ thao tác cho cả Buyer và Admin
- Độ tin cậy (Reliability): Giỏ hàng và thanh toán phải luôn chính xác, tránh mất dữ liệu
- Khả năng duy trì (Maintainability): Dễ dàng thêm sản phẩm mới, mở rộng quy trình nghiệp vụ
- Khả dụng (Availability): Hệ thống luôn sẵn sàng, downtime thấp

8.2. Quy trình nghiệp vụ cho website CoolStore

1. Danh mục sản phẩm (Product Catalog)

- ❖ Buyer:
 - Buyer đăng nhập vào hệ thống.
 - Hệ thống hiển thị danh mục sản phẩm.
 - Buyer có thể:
 - Tìm kiếm sản phẩm
 - Lọc sản phẩm tăng/giảm theo giá hoặc A-Z/Z-A.
 - Buyer chọn 1 sản phẩm → hệ thống hiển thị chi tiết gồm:
 - Tên sản phẩm.
 - Mô tả.
 - Số lượng tồn kho.
 - Thông tin kho (địa chỉ).
 - Trạng thái “Nổi bật” (nếu có).
 - Đánh giá từ khách hàng khác.
- ❖ Admin:
 - Admin đăng nhập.
 - Vào trang Quản trị sản phẩm.
 - Admin có thể thực hiện:
 - Thêm sản phẩm mới.
 - Cập nhật thông tin sản phẩm.
 - Xóa sản phẩm.
 - Gán sản phẩm vào kho hiện có.

2. Quy trình Giỏ hàng (Shopping Cart)

- ❖ Buyer:
 - Buyer chọn “Mua hàng” từ trang danh mục hoặc trang chi tiết sản phẩm.
 - Hệ thống thêm sản phẩm vào giỏ hàng.
 - Hệ thống cập nhật bảng tổng hợp giỏ hàng gồm:

- Tổng chi phí sản phẩm.
- Tổng chi phí sau khuyến mãi.
- Chi phí vận chuyển.
- Tổng giá trị đơn hàng.
- Buyer có thể:
 - Tiếp tục thêm sản phẩm.
 - Xóa sản phẩm khỏi giỏ.
 - Thay đổi số lượng sản phẩm.
 - Hệ thống cập nhật lại giỏ hàng ngay lập tức.
- Buyer nhấn nút “Thanh toán” để chuyển sang quy trình thanh toán.
- ❖ Admin:
 - Admin có thể truy cập giỏ hàng của bất kỳ Buyer nào.
 - Admin kiểm tra tình trạng giỏ hàng.
 - Nếu phát hiện giỏ hàng không hợp lệ (vd: spam, lỗi thanh toán, số lượng ảo) → Admin khóa hoặc vô hiệu hóa giỏ hàng.

3. Quy trình Thanh toán (Payment Process)

- ❖ Buyer:
 - Buyer nhấn Thanh toán từ giỏ hàng.
 - Buyer nhập thông tin thanh toán (thẻ, ví điện tử, COD...).
 - Hệ thống gọi Payment Gateway để xử lý giao dịch.
 - Nếu giao dịch thành công:
 - Trừ số lượng tồn kho.
 - Cập nhật trạng thái đơn hàng.
 - Gửi email xác nhận thành công cho Buyer.
 - Nếu giao dịch thất bại:
 - Giữ nguyên giỏ hàng.
 - Gửi email thông báo thất bại và lý do.

4. Quy trình Quản lý Tồn kho (Inventory)

- ❖ Admin:
 - Admin đăng nhập.
 - Vào mục Quản lý kho hàng.
 - Admin có thể:
 - Thêm kho mới.
 - Sửa thông tin kho (địa chỉ, số lượng tồn).
 - Xóa kho không còn hoạt động.
 - Cập nhật số lượng sản phẩm trong kho.
 - Hệ thống đồng bộ dữ liệu tồn kho với danh mục sản phẩm và giỏ hàng.

5. Quy trình Đánh giá sản phẩm (Rating)

❖ Buyer:

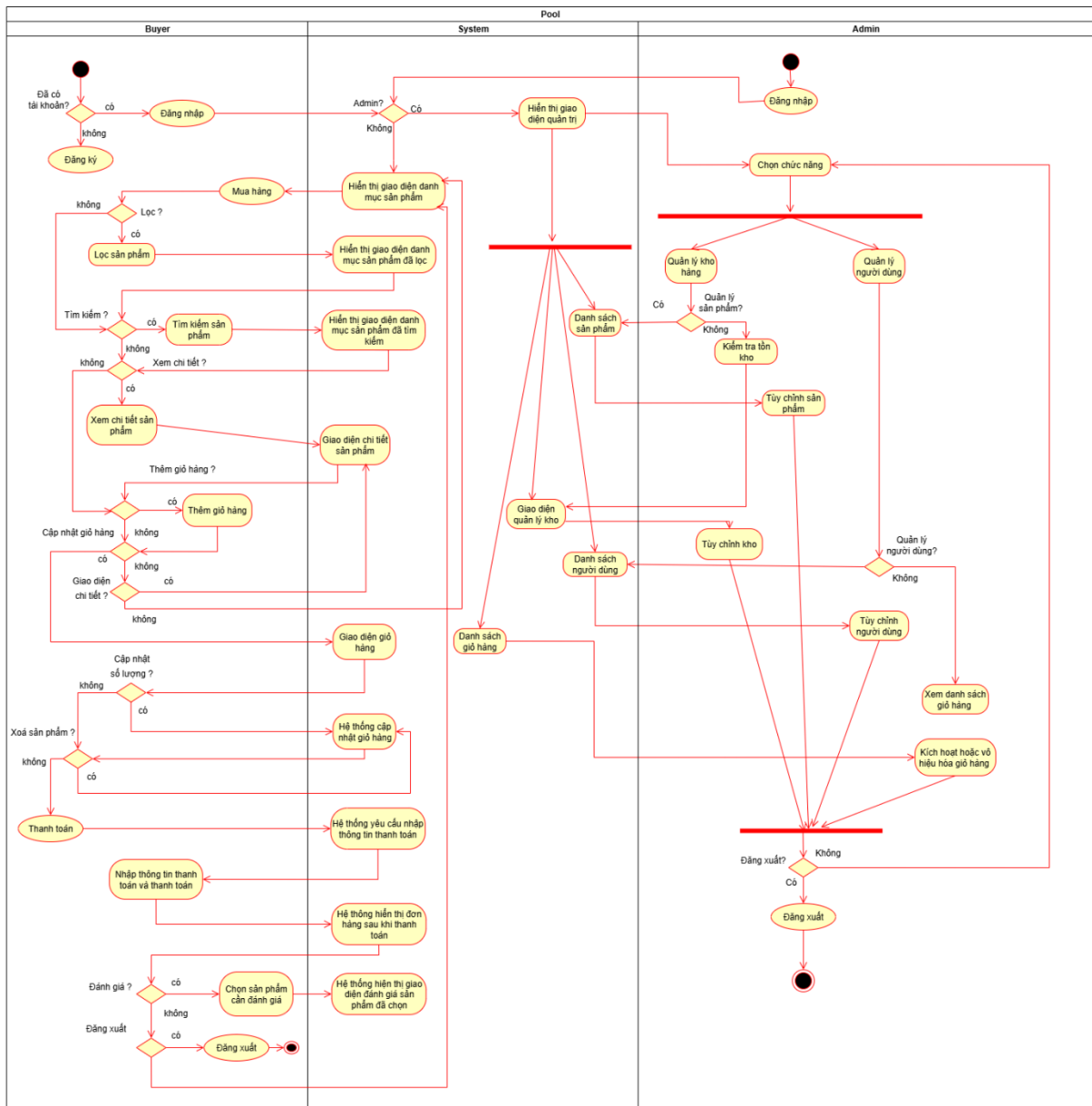
- Buyer đã mua/đã xem sản phẩm → chọn Đánh giá.
- Buyer nhập số sao (1–5) và/hoặc bình luận.
- Hệ thống lưu đánh giá.
- Hệ thống hiển thị đánh giá trên trang chi tiết sản phẩm (để các Buyer khác tham khảo).

6. Quy trình Kiểm soát truy cập (Access Control)

❖ User (Buyer/Admin):

- Người dùng đăng ký tài khoản (email, mật khẩu, thông tin cơ bản).
- Đăng nhập:
 - Nếu Buyer → chuyển tới trang Danh mục sản phẩm.
 - Nếu Admin → chuyển tới trang Quản trị.
- Người dùng có thể đăng xuất.

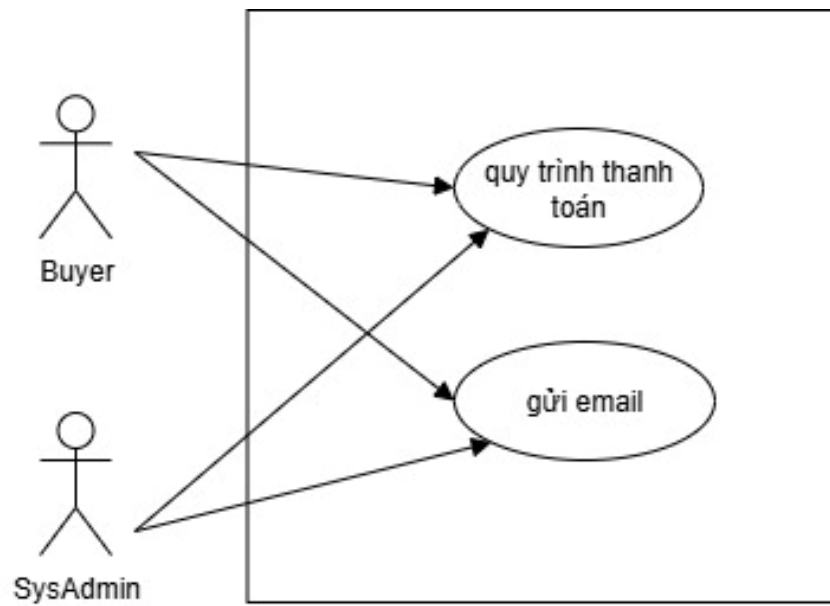
Sơ đồ quy trình nghiệp vụ:



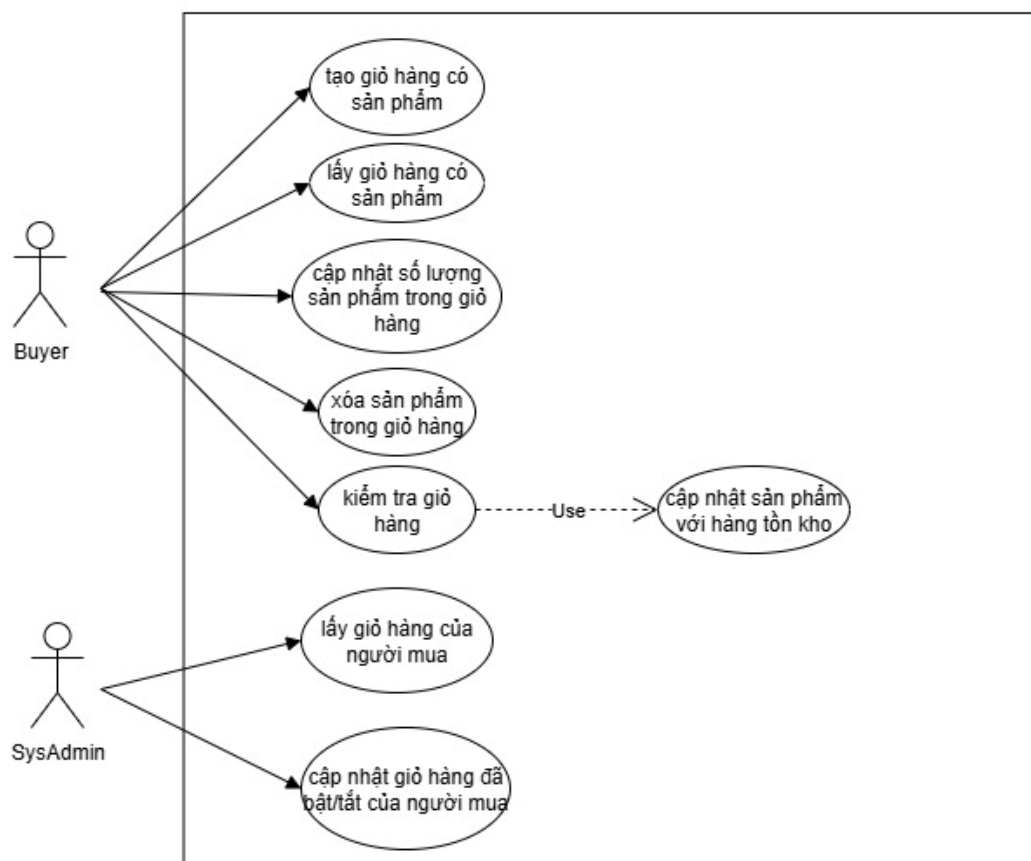
8.3. Xây dựng use case diagram cho website CoolStore.

1. Buyer (Người mua)

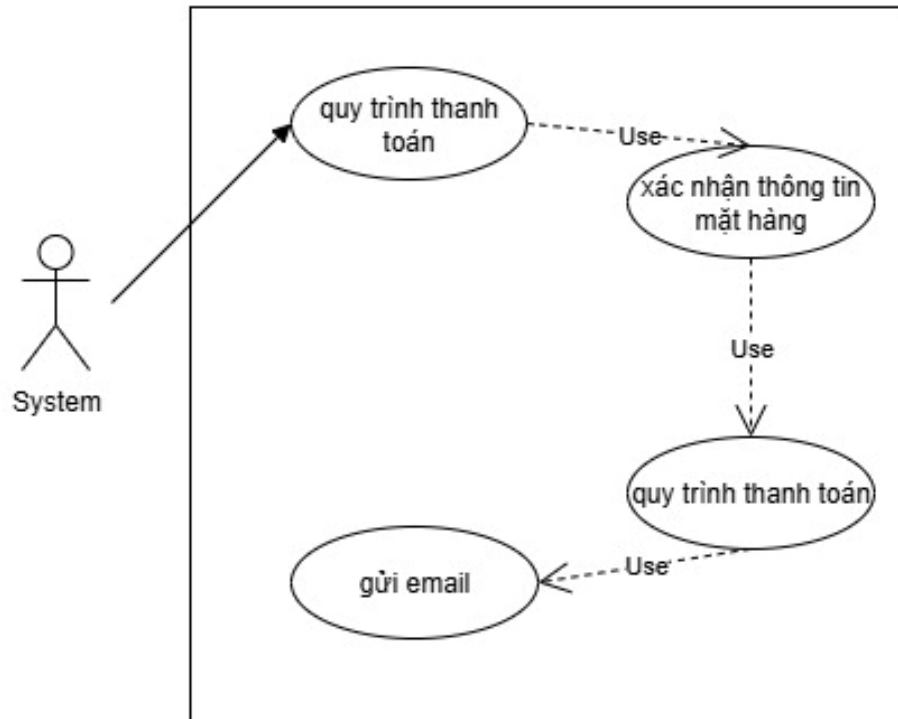
1.1. UC1. Kiểm soát truy cập



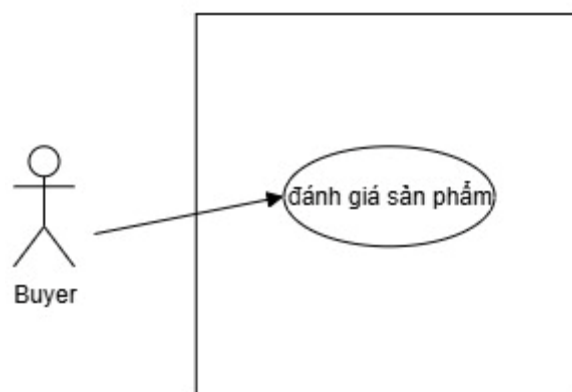
1.2. UC2. Giỏ hàng (Mua hàng)



1.3. UC3. Thanh toán đơn hàng

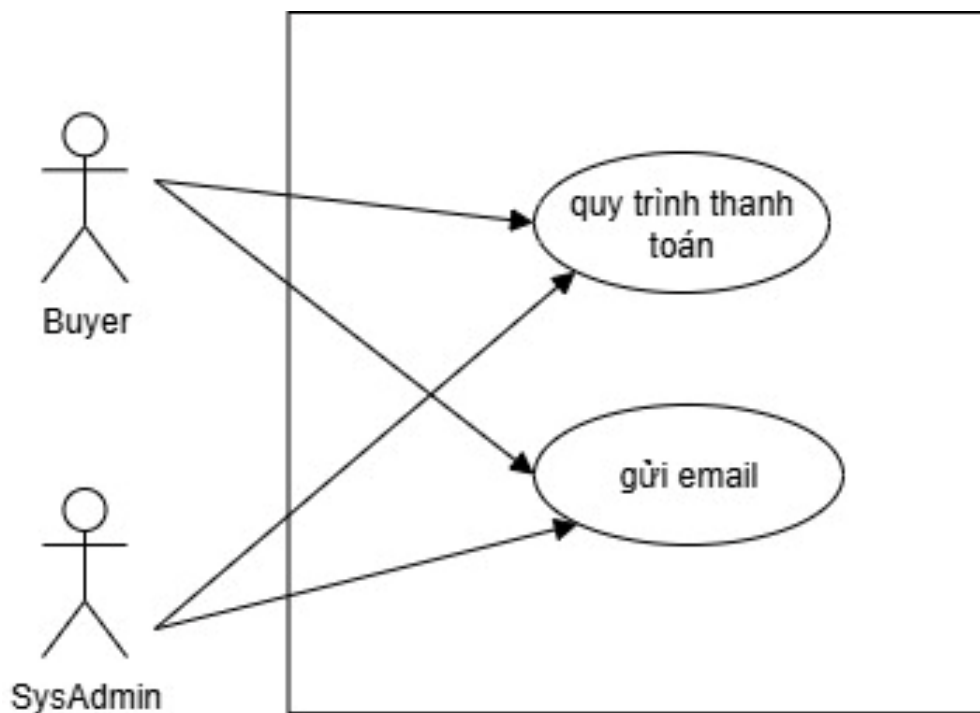


1.4. UC4. Đánh giá sản phẩm (1–5 sao)

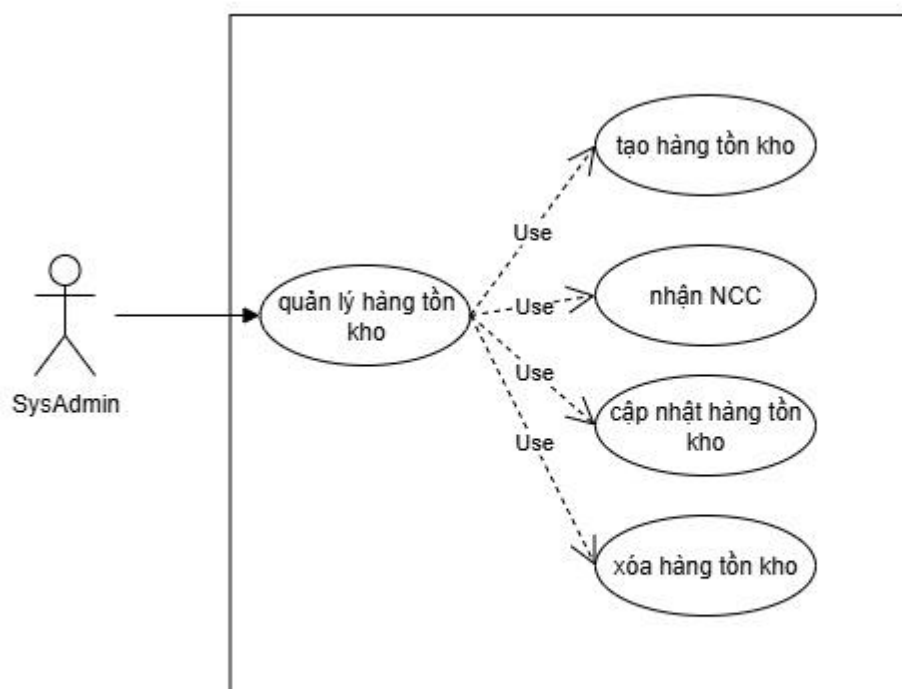


2. SysAdmin (Quản trị viên)

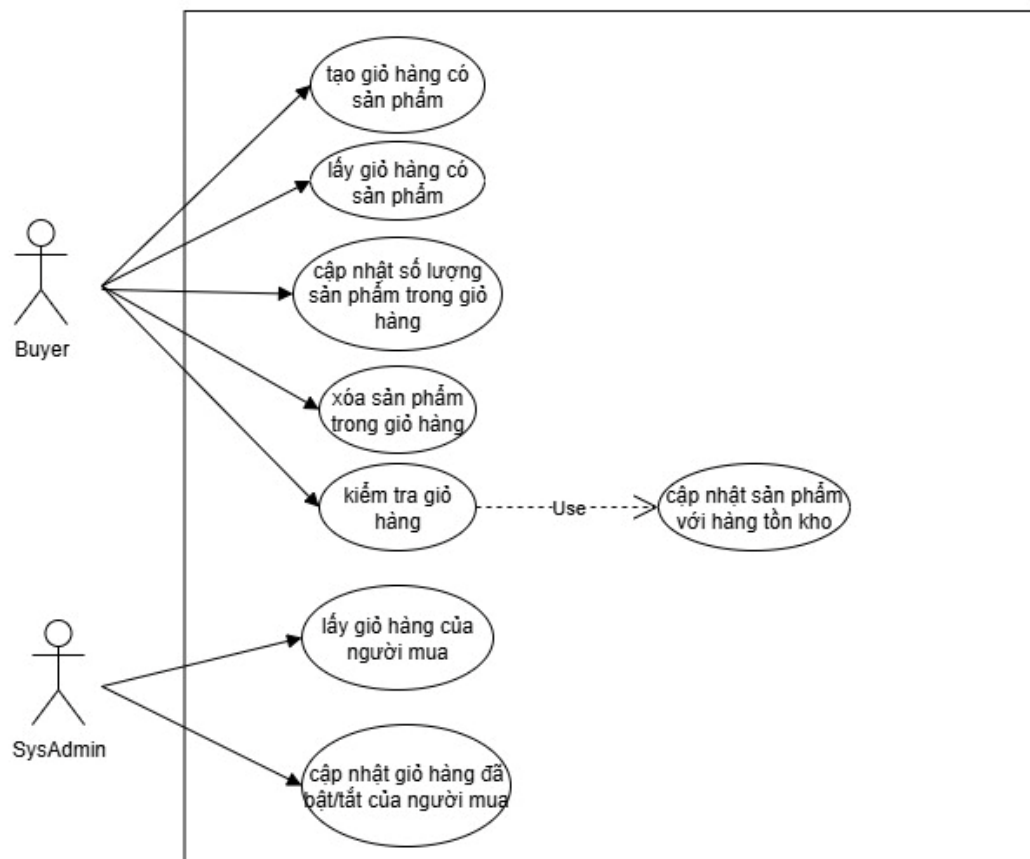
2.1. UC5. Kiểm soát truy cập



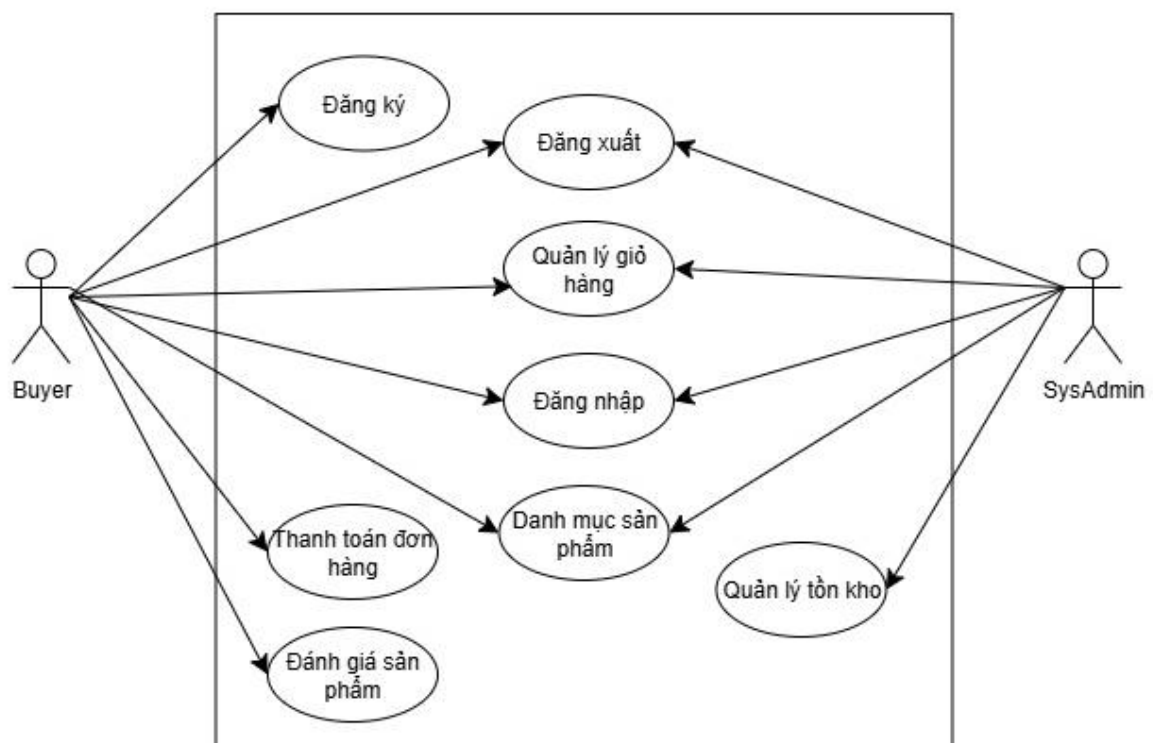
2.2. UC6. Quản lý tài khoản



2.3. UC7. Quản lý giỏ hàng

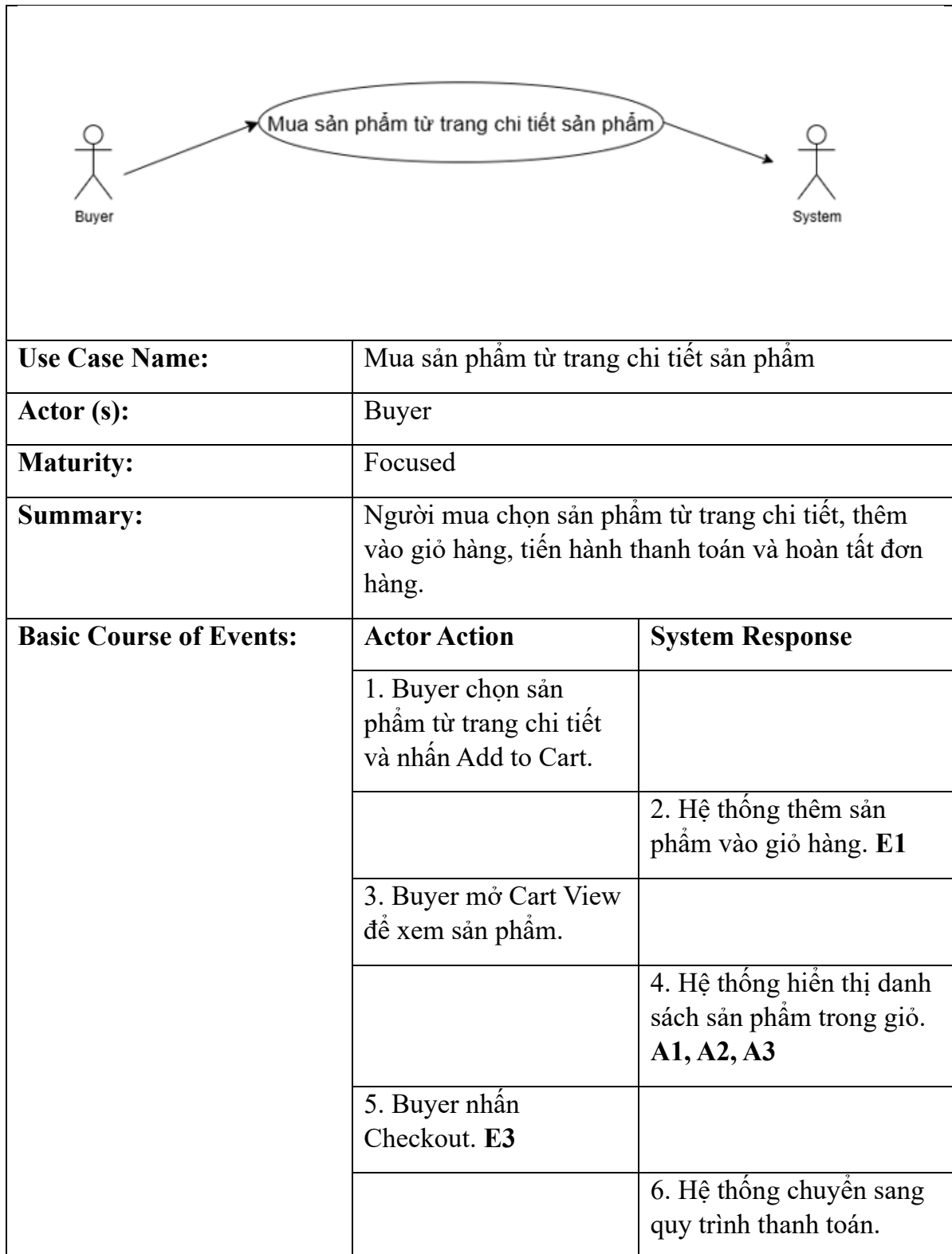


UC tổng



8.4. Mô tả chi tiết một Use Case

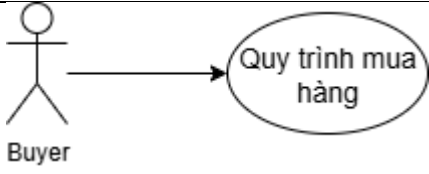
"Mua sản phẩm từ trang chi tiết sản phẩm". Trình bày theo mẫu chuẩn gồm: Tên use case, Tác nhân chính, Mục tiêu, Tiền điều kiện, Luồng chính, Luồng thay thế, Hậu điều kiện, Ghi chú (nếu có).



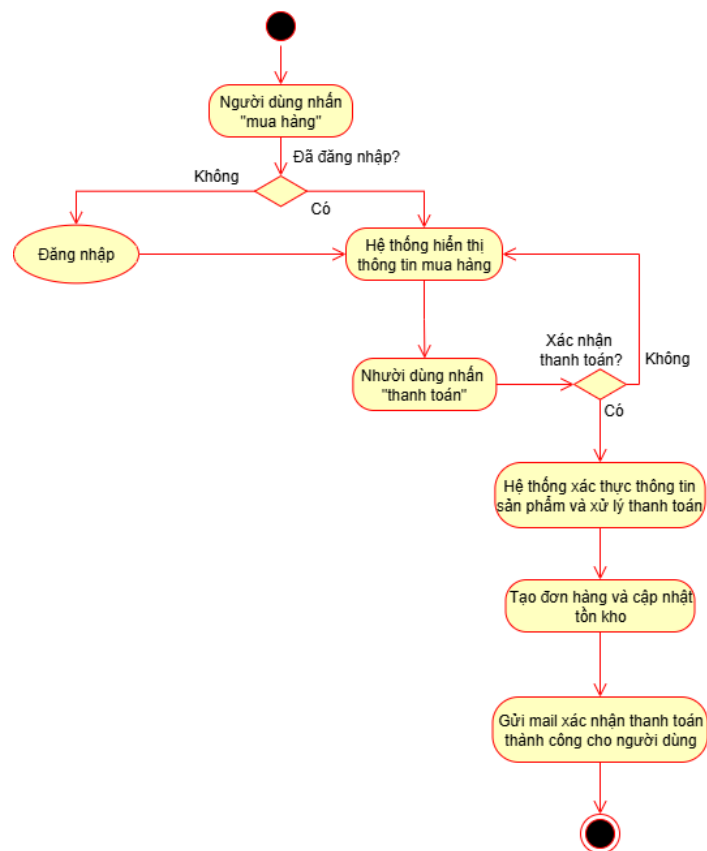
	7. Buyer xác nhận thông tin đơn hàng và phương thức thanh toán.	
	8. Buyer hoàn tất thanh toán.	
		9. Hệ thống xử lý giao dịch và gửi email xác nhận thanh toán thành công. E2
Alternative Paths:	<p>A1. Buyer có thể thay đổi số lượng thì hệ thống sẽ cập nhật giỏ hàng cộng/trừ số lượng.</p> <p>A2. Buyer có thể chọn Xóa sản phẩm thì hệ thống sẽ cập nhật xóa sản phẩm trong giỏ hàng.</p> <p>A3. Buyer có thể bấm vào xem chi tiết sản phẩm.</p>	
Exception Paths:	<p>E1. Nếu sản phẩm hết hàng thì Hệ thống báo lỗi và yêu cầu Buyer xem sản phẩm khác.</p> <p>E2. Nếu thanh toán thất bại thì hệ thống hiển thị thông báo lỗi.</p>	
Extension Points:	E3. Hệ thống có thể mở rộng thêm quy trình áp dụng khuyến mãi/giảm giá.	
Triggers:	Người mua nhấn vào nút "Mua hàng" tại trang chi tiết sản phẩm.	
Assumptions:	Người mua đã đăng nhập hệ thống và sản phẩm còn trong danh mục.	
Preconditions:	Người mua phải có tài khoản và đã đăng nhập vào hệ thống và sản phẩm tồn tại trong cơ sở dữ liệu.	
Post Conditions:	Đơn hàng được tạo trong hệ thống.	
Reference: Business Rules	https://vietnam-devs.github.io/coolstore-microservices/model-microservices/#event-storming-roles-commands-and-events-mapping	
Author(s):	XXXXX	
Date:	14-09-2025	

8.5. Sơ đồ Activity Diagram - mô tả quy trình mua hàng

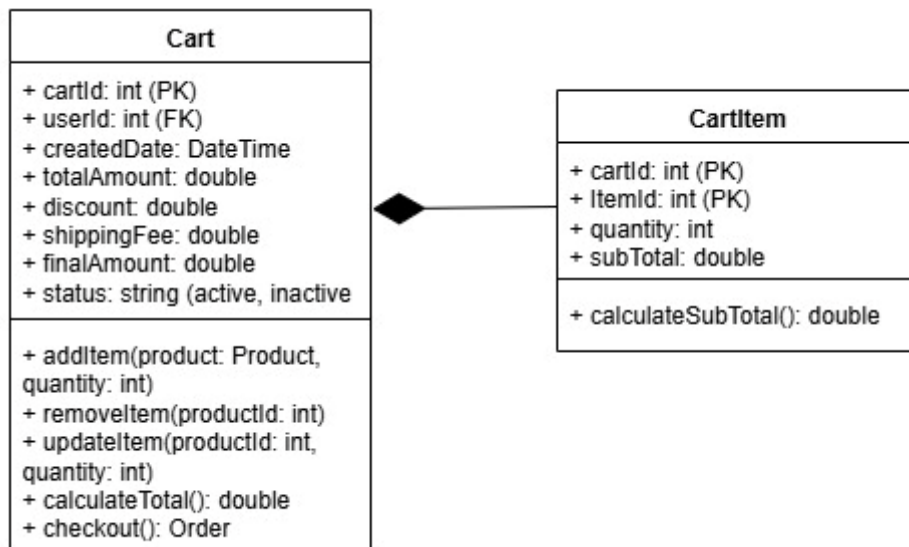
(từ khi người dùng nhấn “Mua hàng” cho đến khi hoàn tất thanh toán)

		
Use Case Number:	UC1	
Use Case Name:	Quy trình mua hàng	
Actor (s):	Buyer	
Maturity:	Focused	
Summary:	Người dùng	
Basic Course of Events:	Actor Action	System Response
	1. Người dùng nhấn vào mua hàng A1	
		2. Hệ thống hiển thị thông tin mua hàng
	3. Người dùng nhấn vào thanh toán A2	
		4. Hệ thống xác thực thông tin sản phẩm và xử lý thanh toán
		5. Tạo đơn hàng và cập nhật tồn kho
		6. Gửi mail xác nhận thanh toán thành công cho người dùng
Alternative Paths:	A1. Nếu chưa đăng nhập người dùng thực hiện đăng nhập { Đăng nhập }	
	A2. Nếu không xác nhận thanh toán sẽ trở lại bước 2 của Basic Course of Events	
Extension Points:	Không có	
Triggers:	Actor muốn mua hàng	
Assumptions:	Actor đã đăng nhập để mua hàng	

Preconditions:	Người dùng đã đăng ký tài khoản
Post Conditions:	Tài khoản đã được tạo
Reference: Business Rules	https://vietnam-devs.github.io/coolstore-microservices/model-microservices/#event-storming-roles-commands-and-events-mapping
Author(s):	XXXXX
Date:	9/14/2025




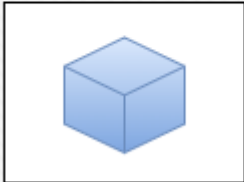
8.6. Thiết kế sơ đồ Class Diagram cho module Giỏ hàng



8.7. Thiết kế giao diện màn hình cho chức năng “Quản lý giỏ hàng”

(có thể thiết kế một hoặc nhiều màn hình) gồm các thông tin:

- Danh sách sản phẩm trong giỏ hàng: tên sản phẩm, hình ảnh, số lượng, giá từng món, tổng phụ

← Giỏ hàng (2)		Sửa
<input checked="" type="checkbox"/> NVCA STORE >		
<input checked="" type="checkbox"/>	 <div> <p>Áo thun đỏ siêu mịn</p> <p>Áo thun đỏ ▾</p> <p>145.000 đ</p> </div>	- 1 +
<input type="checkbox"/> Dream Shop >		
<input type="checkbox"/>	 <div> <p>Khối rubik</p> <p>Rubik xanh ▾</p> <p>100.000 đ</p> </div>	- 1 +
<input type="checkbox"/> Voucher		<input type="text" value="-12k"/> <input type="text" value="Free ship"/>
<input type="checkbox"/> Tất cả		<div>133.000 đ</div> <div>Mua hàng (1)</div>

- Chức năng thao tác nhanh: Tăng/giảm số lượng sản phẩm; Xóa sản phẩm khỏi giỏ; Lưu sản phẩm để mua sau.

← Giỏ hàng (2)		Sửa
<input checked="" type="checkbox"/> NVCA STORE >		
Áo thun đồ siêu mịn Áo thun đồ ▾ 145.000 đ	- 1 +	Xóa
<input type="checkbox"/> Dream Shop >		
<input type="checkbox"/>  Khối rubik Rubik xanh ▾ 100.000 đ	- 1 +	
<input type="checkbox"/> Voucher	<input type="checkbox"/> -12%	<input type="checkbox"/> Free ship
<input type="checkbox"/> Tất cả	133.000 đ	Mua hàng (1)

←

Giỏ hàng (2)

Sửa

☒ NVCA STORE >

☒


Áo thun đỏ siêu mịn

Áo thun đỏ ▾

145.000 đ

- 1 +

☐ Dream Shop >

★

☐


Khối rubik

Rubik xanh ▾

100.000 đ

- 1 +

☐ Voucher

-12%

Free ship

☐ Tất cả

133.000 đ


Mua hàng (1)

←

Giỏ hàng (2)

Sửa

☒ NVCA STORE >

☒



Áo thun đỏ siêu mịn

Áo thun đỏ ▾

145.000 đ

- 2 +

☐ Dream Shop >

☐


Khối rubik

Rubik xanh ▾

100.000 đ

- 1 +

☐ Voucher

-12%

Free ship

☐ Tất cả

133.000 đ

Mua hàng (1)

 Giỏ hàng (2)
 Sửa

 NVCA STORE >




Áo thun đồ siêu mìn
 Áo thun đồ ▾

- 2|+

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
,	0	⊗

- Bảng tổng hợp đơn hàng: Tổng chi phí giỏ hàng; Khuyến mãi tiết kiệm được; Phí vận chuyển; Tổng thanh toán cuối cùng.

Thanh toán

Kieu Anh (A16.12) (+84) 900123456
 Chung cư Green River, số 2225, Đường Phạm Thế Hiển, Phường 6, Quận 8

Dream Shop > NVCA STORE>

Áo thun đỏ siêu mịn
 Áo thun đỏ ▾
 đ260.000 đ350.000 x1

Phương thức vận chuyển

Xem tất cả >

Nhanh —đ18.000—

Đảm bảo nhận hàng từ 15 tháng 9 - 19 tháng 9
 Nhận voucher trị giá đ15.000 nếu đơn hàng giao đến bạn sau ngày 19 Tháng 9 2025

Tổng số tiền (1 sản phẩm)

đ268.000

Voucher

-8k >

Phương thức thanh toán

Xem tất cả >

Thanh toán khi nhận hàng

Chuyển khoản qua ngân hàng liên kết >

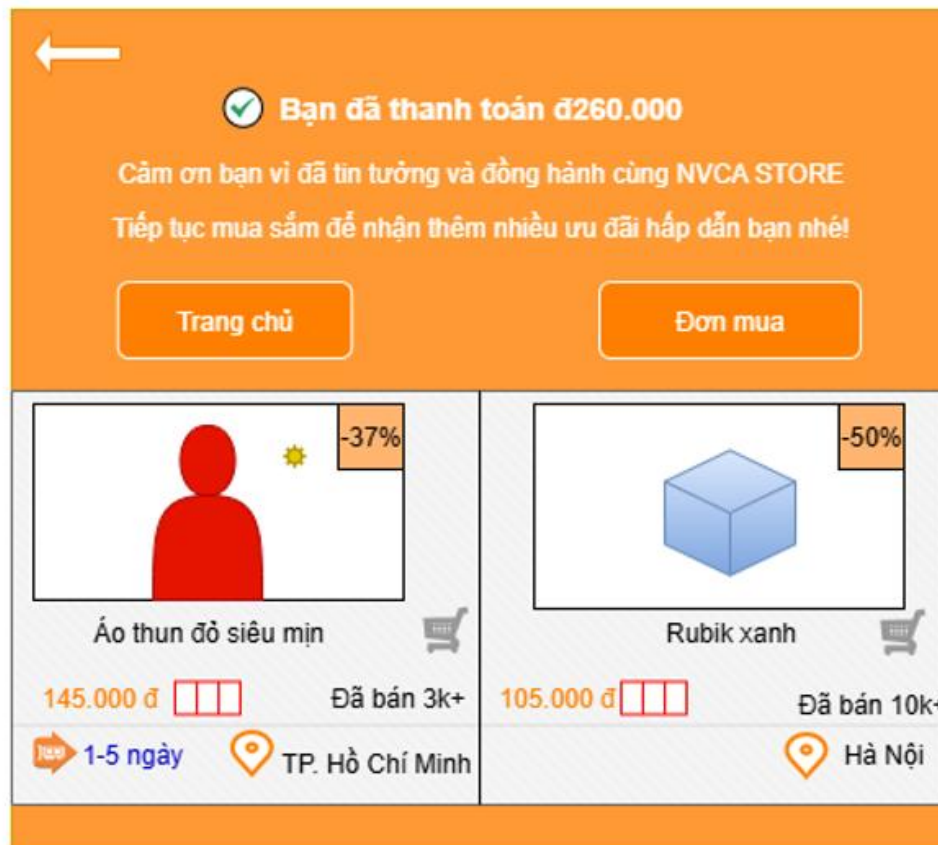
Chi tiết thanh toán

Tổng tiền hàng	đ288.000
Tổng tiền phí vận chuyển	đ18.000
Giảm giá phí vận chuyển	-đ18.000
Tổng cộng Voucher giảm giá	-đ28.000
Tổng thanh toán	đ260.000

Tổng cộng **đ260.000**
 Tiết kiệm **đ108.000**

Đặt hàng

- Hành động chính: Nút “Thanh toán”; Gợi ý sản phẩm liên quan để thêm vào giỏ.



8.8. Đề xuất cách tổ chức kiến trúc phần mềm cho hệ thống CoolStore theo mô hình kiến trúc ba lớp (3-tier architecture)

1. Presentation Layer (UI Layer)

❖ Công nghệ:

- ASP.NET Core MVC / Razor Pages hoặc SPA (React, Angular, Vue).
- Giao tiếp với Application Layer thông qua API (REST hoặc GraphQL).

❖ Các vai trò và màn hình:

- Người mua (Buyer):
 - **Đăng nhập/Đăng ký/Đăng xuất:** kiểm soát truy cập hệ thống.
 - **Danh mục sản phẩm:** xem danh sách, tìm kiếm, lọc, sắp xếp theo tên hoặc giá.
 - **Chi tiết sản phẩm:** xem mô tả, số lượng tồn kho, thông tin kho (địa chỉ, trạng thái), đánh giá, trạng thái “sản phẩm nổi bật”.
 - **Giỏ hàng:** thêm sản phẩm từ danh mục hoặc chi tiết sản phẩm, chỉnh sửa hoặc xóa sản phẩm khỏi giỏ hàng.
 - **Tóm tắt giỏ hàng:** hiển thị tổng chi phí, chi phí sau khuyến mãi, phí vận chuyển, khuyến mãi tiết kiệm, tổng giá trị đơn hàng.
 - **Thanh toán:** thực hiện quy trình thanh toán.
 - **Đánh giá sản phẩm:** để lại đánh giá (1–5 sao) cho sản phẩm.

- Quản trị viên (SysAdmin) gồm các màn hình:
 - **Đăng nhập/Đăng xuất:** truy cập vào trang quản trị.
 - **Danh mục sản phẩm:** quản lý danh sách sản phẩm, lọc/sắp xếp.
 - **Chi tiết sản phẩm:** xem và chỉnh sửa thông tin sản phẩm, gán sản phẩm vào kho, xem đánh giá.
 - **Giỏ hàng:** theo dõi giỏ hàng của người mua, có thể kích hoạt/vô hiệu hóa giỏ hàng không hợp lệ.
 - **Thanh toán:** giám sát quy trình thanh toán.
 - **Trang quản trị:** quản lý sản phẩm, quản lý kho, quản lý giỏ hàng và dữ liệu liên quan.
- ❖ Gửi request đến tầng Business qua API.
- 2. Business Logic Layer (BLL / Service Layer)
 - ❖ Xử lý quy tắc nghiệp vụ:
 - Quản lý giỏ hàng (Shopping Cart Service):
 - Thêm, xóa, cập nhật sản phẩm trong giỏ hàng.
 - Tính toán tổng tiền, phí ship, khuyến mãi, tổng giá trị đơn hàng.
 - Cập nhật bảng tóm tắt giỏ hàng theo thời gian thực.
 - Quản lý thanh toán (Payment Service):
 - Xác thực thông tin sản phẩm và tồn kho trước khi thanh toán.
 - Xử lý giao dịch thanh toán.
 - Gửi email thông báo tình trạng đơn hàng cho người mua.
 - Quản lý tồn kho (Inventory Service):
 - Theo dõi số lượng tồn kho sản phẩm.
 - Cập nhật số lượng khi phát sinh giao dịch hoặc khi quản trị viên thay đổi.
 - Cho phép SysAdmin gán sản phẩm vào kho hiện có.
 - Quản lý sản phẩm (Product Catalog Service):
 - Quản lý danh mục sản phẩm, chi tiết sản phẩm, trạng thái “sản phẩm nổi bật”.
 - Quản lý đánh giá (Rating Service):
 - Người mua đánh giá sản phẩm theo thang điểm 1–5 sao.
 - Quản trị viên theo dõi, quản lý các đánh giá.
 - Quản lý kiểm soát truy cập (Access Control / Identity Service):
 - Đăng nhập/đăng ký/đăng xuất.
 - Phân quyền người dùng (Buyer / SysAdmin).
 - Điều hướng Buyer đến trang danh mục sản phẩm, SysAdmin đến trang quản trị.
 - ❖ Công nghệ hỗ trợ:
 - REST API hoặc GraphQL API.
 - Tích hợp Email Service để gửi thông báo đơn hàng.
 - Áp dụng kiến trúc Microservices (Product, Cart, Payment, Inventory, Rating, Access Control) để đảm bảo khả năng mở rộng.

3. Data Access Layer (DAL)

❖ Công nghệ:

- SQL Server, PostgreSQL hoặc MySQL.
- Sử dụng ORM (Entity Framework Core, Dapper) để mapping dữ liệu với Entity.

❖ Các bảng chính:

- **User:** thông tin người dùng (Buyer, SysAdmin), tài khoản, phân quyền.
- **Product:** thông tin sản phẩm.
- **Inventory:** thông tin kho, địa chỉ, số lượng tồn kho.
- **Cart:** thông tin giỏ hàng của người mua.
- **CartItem:** chi tiết sản phẩm trong giỏ hàng.
- **Order:** thông tin đơn hàng khi thanh toán.
- **Payment:** thông tin giao dịch thanh toán.
- **Rating:** đánh giá sản phẩm (1–5 sao).

❖ Nhiệm vụ:

- Thực hiện CRUD dữ liệu cho các bảng.
- Đảm bảo tính toàn vẹn dữ liệu (quan hệ giữa User – Cart – Order – Payment, Product – Inventory – Rating).
- Hỗ trợ Business Layer truy xuất dữ liệu nhanh chóng và an toàn.

8.9. Kiểm thử (test cases) cho chức năng “Thêm sản phẩm vào giỏ hàng”

(bao gồm dữ liệu đầu vào, bước thực hiện và kết quả mong đợi)

Test Case 1: Thêm sản phẩm khi chưa đăng nhập

- ID: TC01
- Mục tiêu: Kiểm tra hệ thống khi người dùng chưa đăng nhập nhưng cố gắng thêm sản phẩm vào giỏ.
- Điều kiện tiên quyết: Người dùng chưa đăng nhập vào hệ thống.
- **Input:** Người dùng click nút “Mua hàng” trên sản phẩm “Áo thun CoolStore”.
- **Bước thực hiện:**
 1. Mở trang danh mục sản phẩm khi chưa đăng nhập.
 2. Nhấn “Mua hàng” trên sản phẩm “Áo thun CoolStore”.
- **Kết quả mong đợi (Expected Result):**
 - Hệ thống hiển thị thông báo “*Vui lòng đăng nhập để thêm sản phẩm vào giỏ hàng*”.
 - Hệ thống chuyển hướng tới trang Đăng nhập.
 - Giỏ hàng không thay đổi.
- **Test result:** OK(Pass) / NG(Fail)

⇒ Negative test-case

Test Case 2: Thêm sản phẩm hợp lệ vào giỏ hàng từ trang danh sách sản phẩm

- ID: TC02

- User Story ID: US-SC-02- Buyer thêm sản phẩm vào giỏ từ Catalog
- Mục tiêu: Kiểm tra việc thêm sản phẩm hợp lệ từ trang danh sách sản phẩm.
- Điều kiện tiên quyết: Người mua đã đăng nhập thành công, sản phẩm “Áo thun CoolStore” tồn tại trong danh mục và còn hàng trong kho.
- **Input:** Người dùng click nút “Mua hàng” trên sản phẩm “Áo thun CoolStore”.
- **Bước thực hiện:**
 1. Đăng nhập bằng tài khoản Buyer.
 2. Truy cập trang danh mục sản phẩm.
 3. Tìm sản phẩm “Áo thun CoolStore”.
 4. Nhấn “Mua hàng”.
- **Kết quả mong đợi (Expected Result):**
 - Hệ thống hiển thị thông báo “*Sản phẩm đã được thêm vào giỏ hàng*”.
 - Giỏ hàng hiển thị sản phẩm “Áo thun CoolStore” với số lượng = 1.
 - Bảng tổng hợp giỏ hàng cập nhật tổng chi phí chính xác khi chọn vào mặt hàng cần mua.
- **Test result:** OK(Pass) / NG(Fail)

⇒ Trường hợp hợp lệ (happy path).

Test Case 3: Thêm sản phẩm khi số lượng tồn kho = 0

- ID: TC03
- Mục tiêu: Kiểm tra khi người mua thêm sản phẩm đã hết hàng.
- Điều kiện tiên quyết: Người mua đã đăng nhập, sản phẩm “Áo thun CoolStore” tồn tại nhưng số lượng tồn kho = 0.
- **Input:** Người dùng click nút “Mua hàng” trên sản phẩm “Áo thun CoolStore”.
- **Bước thực hiện:**
 1. Đăng nhập bằng tài khoản Buyer.
 2. Truy cập trang danh mục sản phẩm.
 3. Nhấn “Mua hàng” trên sản phẩm “Áo thun CoolStore”.
- **Kết quả mong đợi (Expected Result):**
 - Hệ thống hiển thị thông báo “*Sản phẩm đã hết hàng, không thể thêm vào giỏ*”.
 - Giỏ hàng không có thêm sản phẩm này.
 - Bảng tổng hợp giữ nguyên giá trị trước đó.
- **Test result:** OK(Pass) / NG(Fail)

⇒ Trường hợp ngoại lệ (out of stock).