ỦY BAN NHÂN DÂN TP . HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO BÀI TẬP GIỮA KỲ

Học phần: Kiểm thử phần mềm

Đề tài: **Bài tập 2 – Phân tích thiết kế hệ thống**

Giảng viên hướng dẫn: **TS. Đỗ Như Tài**

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	**Trang Gia Huy**	**3122411068**
2	**Nguyễn Lê Quỳnh Hương**	**3122411078**

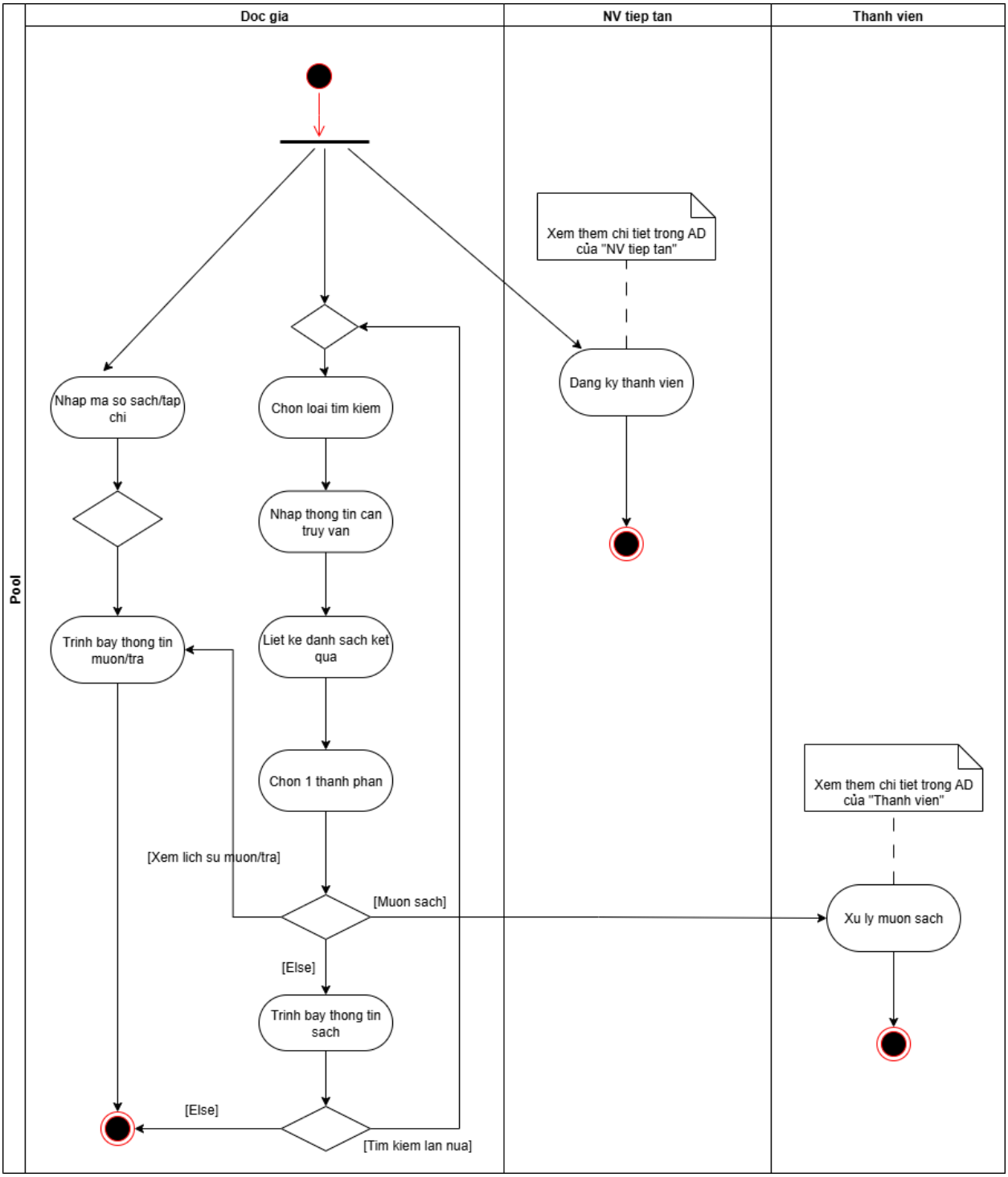
TP. HỒ CHÍ MINH, THÁNG 4 NĂM 2025

MỤC LỤC

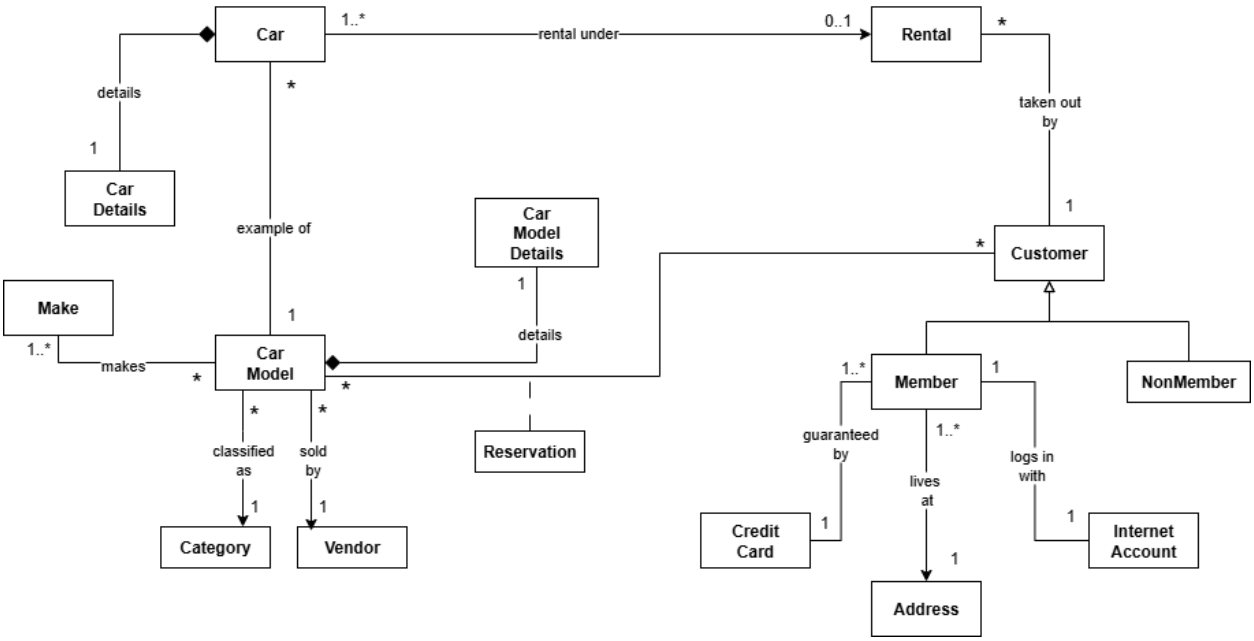
BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	MSSV	Nội dung công việc
1	Trang Gia Huy	3122411068	- Soạn báo cáo - Bài 1, Bài 2, Bài 3, Bài 4, Bài 5, Bài 6, Bài 7
2	Nguyễn Lê Quỳnh Hương	3122411078	- Bài tập ứng dụng (Bài 8)

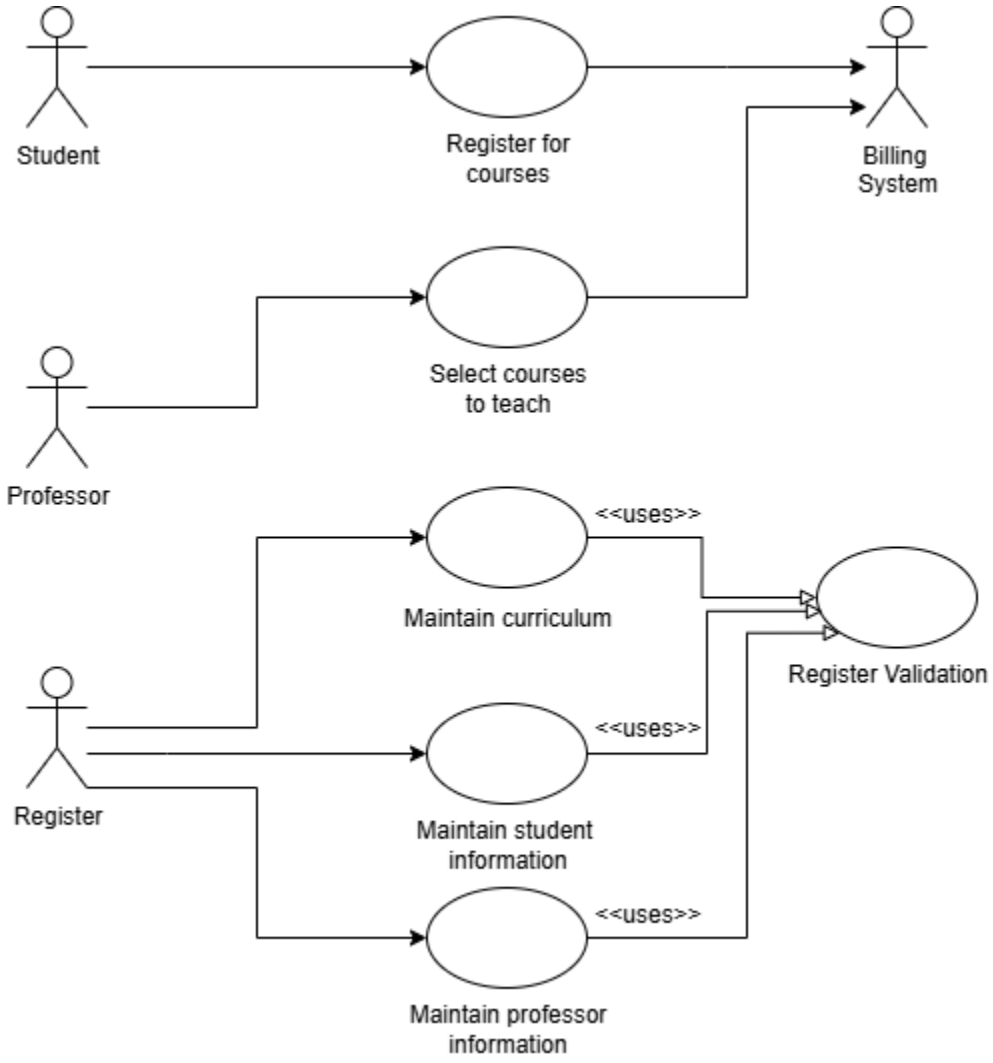
Bài 1: Vẽ lại quy trình nghiệp vụ sau (Độc giả, NV Tiếp tân, Thành viên)



Bài 2: Vẽ mô hình khái niệm sau



Bài 3: Vẽ lại business use case sau



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

1. Student (Sinh viên)

- Người dùng hệ thống để đăng ký môn học.
- Có liên kết với hệ thống thanh toán (Billing System) khi đăng ký.

2. Professor (Giảng viên)

- Chọn các khóa học/môn học mà họ sẽ giảng dạy.

3. Register (Nhân viên quản lý/Phòng đào tạo)

- Quản lý thông tin sinh viên.
- Quản lý thông tin giảng viên.

- Duy trì, cập nhật chương trình học.

4. Billing System (Hệ thống thanh toán)

- Hệ thống bên ngoài nhận dữ liệu từ việc sinh viên đăng ký môn học để tính phí/thu học phí

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

1. Register for Courses (Đăng ký khóa học)

- Sinh viên chọn môn học và gửi thông tin đăng ký.
- Thông tin được chuyển cho hệ thống thanh toán.

2. Select Courses to Teach (Chọn môn để giảng dạy)

- Giảng viên chọn những môn học mà họ phụ trách trong kỳ học.

3. Maintain Curriculum (Quản lý chương trình học)

- Nhân viên quản lý cập nhật, thêm, sửa, xóa nội dung chương trình học.
- Bao gồm kiểm tra hợp lệ (Register Validation).

4. Maintain Student Information (Quản lý thông tin sinh viên)

- Nhập và cập nhật hồ sơ sinh viên.
- Bao gồm xác thực dữ liệu nhập (Register Validation).

5. Maintain Professor Information (Quản lý thông tin giảng viên)

- Nhập và cập nhật hồ sơ giảng viên.
- Bao gồm xác thực dữ liệu nhập (Register Validation).

6. Register Validation (Xác thực đăng ký/thông tin)

- Use case chung được sử dụng bởi nhiều tác vụ quản lý để đảm bảo dữ liệu hợp lệ.

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

1. «uses» (include)

- Maintain Curriculum → sử dụng Register Validation.
- Maintain Student Information → sử dụng Register Validation.
- Maintain Professor Information → sử dụng Register Validation.

=> Điều này nghĩa là mỗi khi thực hiện các thao tác quản lý này, hệ thống đều gọi chức năng kiểm tra hợp lệ để đảm bảo tính chính xác.

2. Association

- Student ↔ Register for Courses ↔ Billing System.
- Professor ↔ Select Courses to Teach.
- Register ↔ (Maintain Curriculum, Maintain Student Information, Maintain Professor Information).

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

Tên Use Case: Register for Courses

Tác nhân chính: Student

Mục tiêu: Sinh viên đăng ký môn học cho học kỳ.

Điều kiện tiên quyết:

- Sinh viên đã có tài khoản trong hệ thống.
- Hệ thống đã mở kỳ đăng ký môn học.

Luồng chính:

1. Sinh viên đăng nhập vào hệ thống.
2. Hệ thống hiển thị danh sách các môn học có thể đăng ký.
3. Sinh viên chọn các môn muốn đăng ký.
4. Hệ thống kiểm tra điều kiện tiên quyết (số tín chỉ tối đa, môn học đã đạt trước đó, xung đột lịch học).
5. Nếu hợp lệ, hệ thống ghi nhận đăng ký.
6. Hệ thống gửi thông tin đăng ký sang Billing System để tính học phí.
7. Billing System phản hồi trạng thái thành công.
8. Hệ thống thông báo "Đăng ký thành công" cho sinh viên.

Luồng thay thế:

- Nếu vi phạm điều kiện tiên quyết (ví dụ trùng lịch, quá số tín chỉ), hệ thống báo lỗi và yêu cầu sinh viên chỉnh sửa lựa chọn.
- Nếu Billing System báo lỗi, hệ thống thông báo cho sinh viên và yêu cầu thử lại.

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

1. Use Case: View Course Schedule (Xem thời khóa biểu)

Tác nhân chính: Student (Sinh viên)

Mục tiêu: Sinh viên xem lịch học sau khi đã đăng ký môn học thành công.

Điều kiện tiên quyết:

- Sinh viên đã đăng ký ít nhất một môn học.
- Sinh viên đã đăng nhập vào hệ thống.

Luồng chính:

1. Sinh viên chọn chức năng "Xem thời khóa biểu".
2. Hệ thống lấy thông tin từ các môn học mà sinh viên đã đăng ký.
3. Hệ thống hiển thị thời khóa biểu (theo ngày/tuần/kỳ học).
4. Sinh viên có thể in hoặc tải xuống lịch học.

Luồng thay thế:

- Nếu sinh viên chưa đăng ký môn học nào → hệ thống hiển thị thông báo "Bạn chưa có môn học nào trong học kỳ này".

2. Use Case: Drop/Change Courses (Hủy/Đổi môn học)

Tác nhân chính: Student (Sinh viên)

Mục tiêu: Cho phép sinh viên thay đổi môn học đã đăng ký trong thời gian cho phép.

Điều kiện tiên quyết:

- Sinh viên đã đăng ký môn học.
- Đang trong thời gian hiệu lực cho phép thay đổi đăng ký.

Luồng chính:

1. Sinh viên chọn chức năng "Hủy/Đổi môn học".
2. Hệ thống hiển thị danh sách các môn mà sinh viên đã đăng ký.
3. Sinh viên chọn môn muốn hủy hoặc đổi.
4. Nếu đổi môn: sinh viên chọn môn thay thế từ danh sách môn khả dụng.
5. Hệ thống kiểm tra điều kiện (tín chỉ tối đa, tiên quyết, trùng lịch).
6. Nếu hợp lệ, hệ thống cập nhật đăng ký mới.
7. Hệ thống gửi thay đổi cho Billing System để điều chỉnh học phí.
8. Hệ thống thông báo “Cập nhật đăng ký thành công”.

Luồng thay thế:

- Nếu không hợp lệ (trùng lịch, vượt tín chỉ, chưa đạt môn tiên quyết) → hệ thống báo lỗi và yêu cầu chọn lại.

- Nếu đã hết hạn thay đổi môn học → hệ thống báo lỗi “Không thể thay đổi đăng ký sau hạn chót”.

3. Use Case: Generate Reports (Báo cáo thống kê)

Tác nhân chính: Register (Phòng đào tạo)

Mục tiêu: Tạo báo cáo thống kê phục vụ quản lý đào tạo.

Điều kiện tiên quyết:

- Người dùng (Register) có quyền truy cập báo cáo.
- Dữ liệu sinh viên và giảng viên đã được cập nhật trong hệ thống.

Luồng chính:

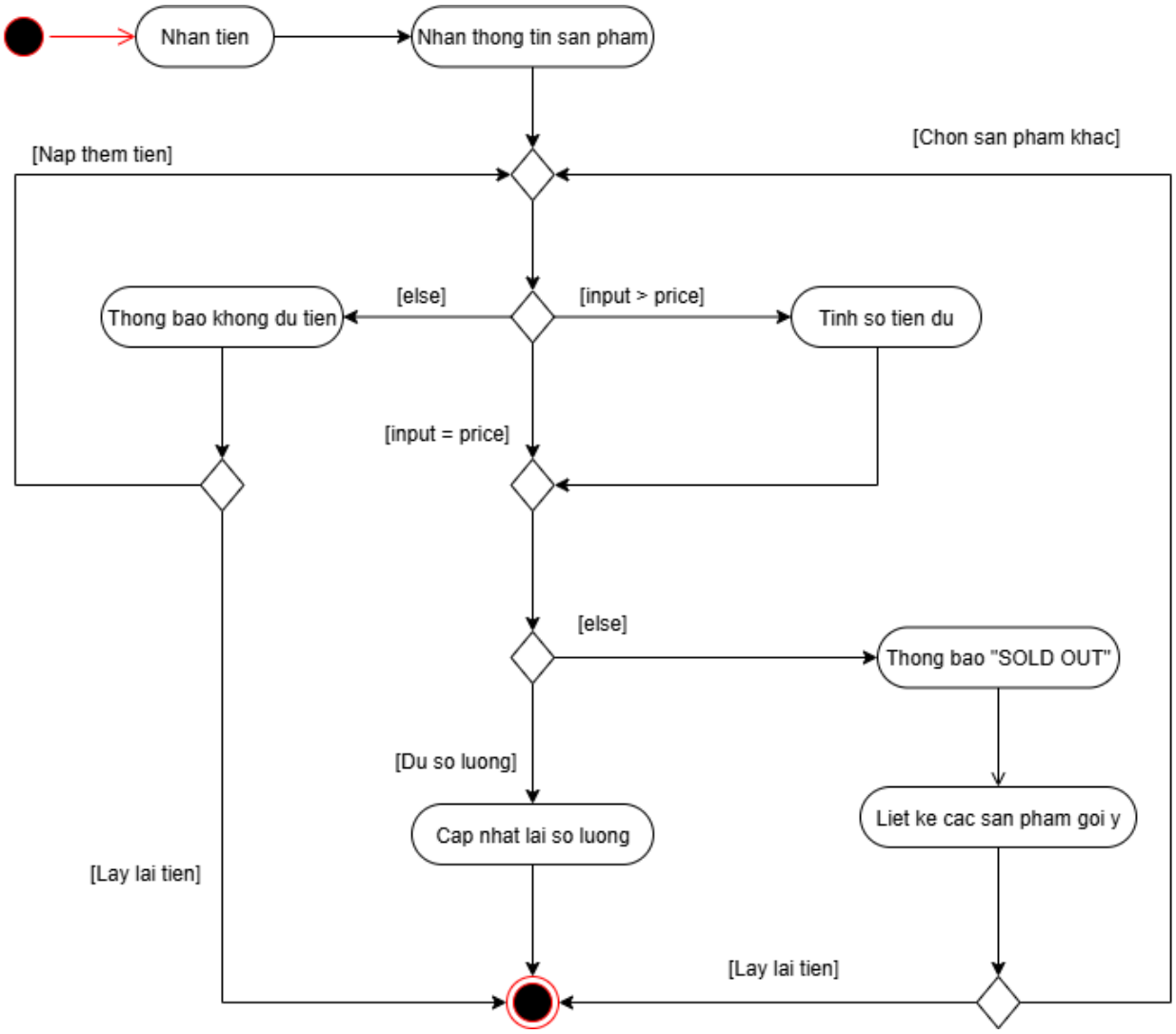
1. Register chọn chức năng "Tạo báo cáo thống kê".
2. Hệ thống hiển thị các tùy chọn báo cáo (ví dụ: số sinh viên đăng ký theo môn, danh sách môn học của giảng viên, tình trạng đăng ký theo học kỳ).
3. Register chọn loại báo cáo mong muốn.
4. Hệ thống truy xuất dữ liệu từ cơ sở dữ liệu.
5. Hệ thống tạo báo cáo và hiển thị kết quả (bảng, biểu đồ, hoặc file xuất ra).
6. Register có thể tải về hoặc in báo cáo.

Luồng thay thế:

- Nếu không có dữ liệu phù hợp (ví dụ chưa có sinh viên đăng ký) → hệ thống hiển thị thông báo “Không có dữ liệu để tạo báo cáo”.

Bài 4. Vẽ sơ đồ hoạt động và tương tác sau và tóm tắt ý nghĩa sơ đồ

a) Sơ đồ Activity:

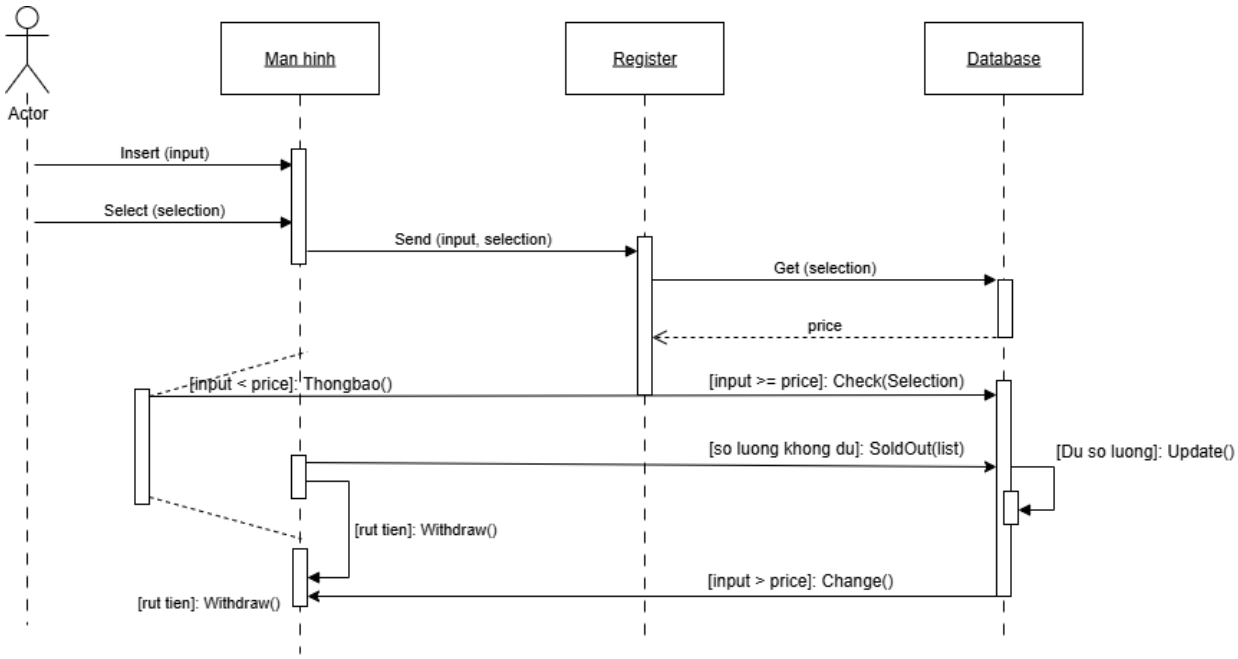


Sơ đồ mô tả quy trình mua sản phẩm từ máy bán hàng tự động:

1. Người dùng nạp tiền và chọn sản phẩm.
2. Hệ thống so sánh số tiền nhập với giá sản phẩm:
   - Nếu tiền < giá → thông báo không đủ, yêu cầu nạp thêm hoặc lấy lại tiền.
   - Nếu tiền > giá → tính tiền thừa, trả lại tiền thừa cho khách.
   - Nếu tiền = giá → chuyển bước tiếp theo.
3. Hệ thống kiểm tra số lượng sản phẩm:
   - Nếu còn hàng → cập nhật số lượng, xuất sản phẩm cho khách.
   - Nếu hết hàng → thông báo “SOLD OUT” và gợi ý sản phẩm khác.

4. Kết thúc giao dịch: khách nhận sản phẩm và/hoặc tiền thừa, hoặc lấy lại tiền nếu không mua được.

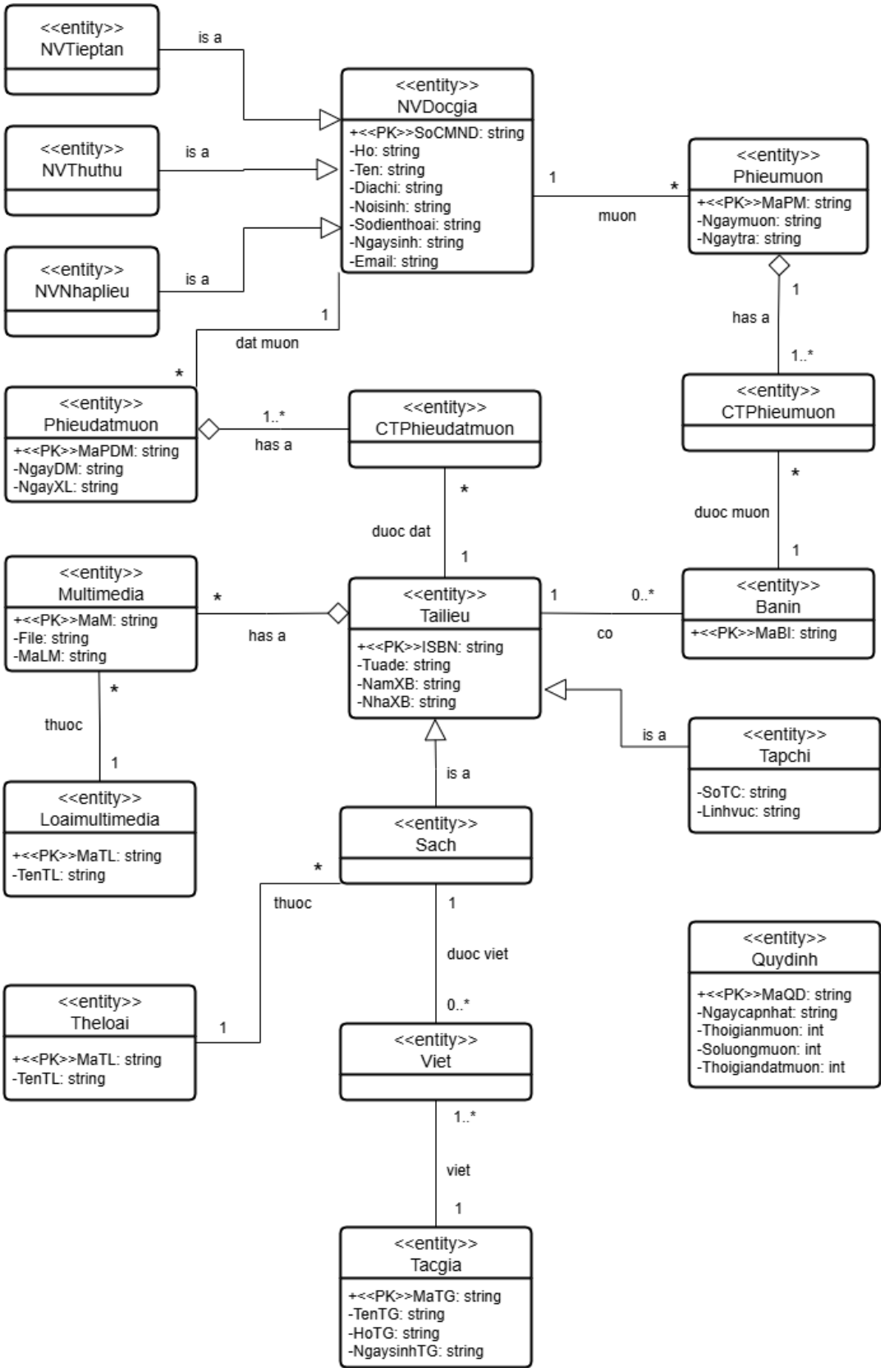
b) Sơ đồ tương tác:



Biểu đồ mô tả luồng tương tác giữa Người dùng (Actor), Màn hình, Register (bộ xử lý trung tâm) và Cơ sở dữ liệu (Database) khi mua hàng:

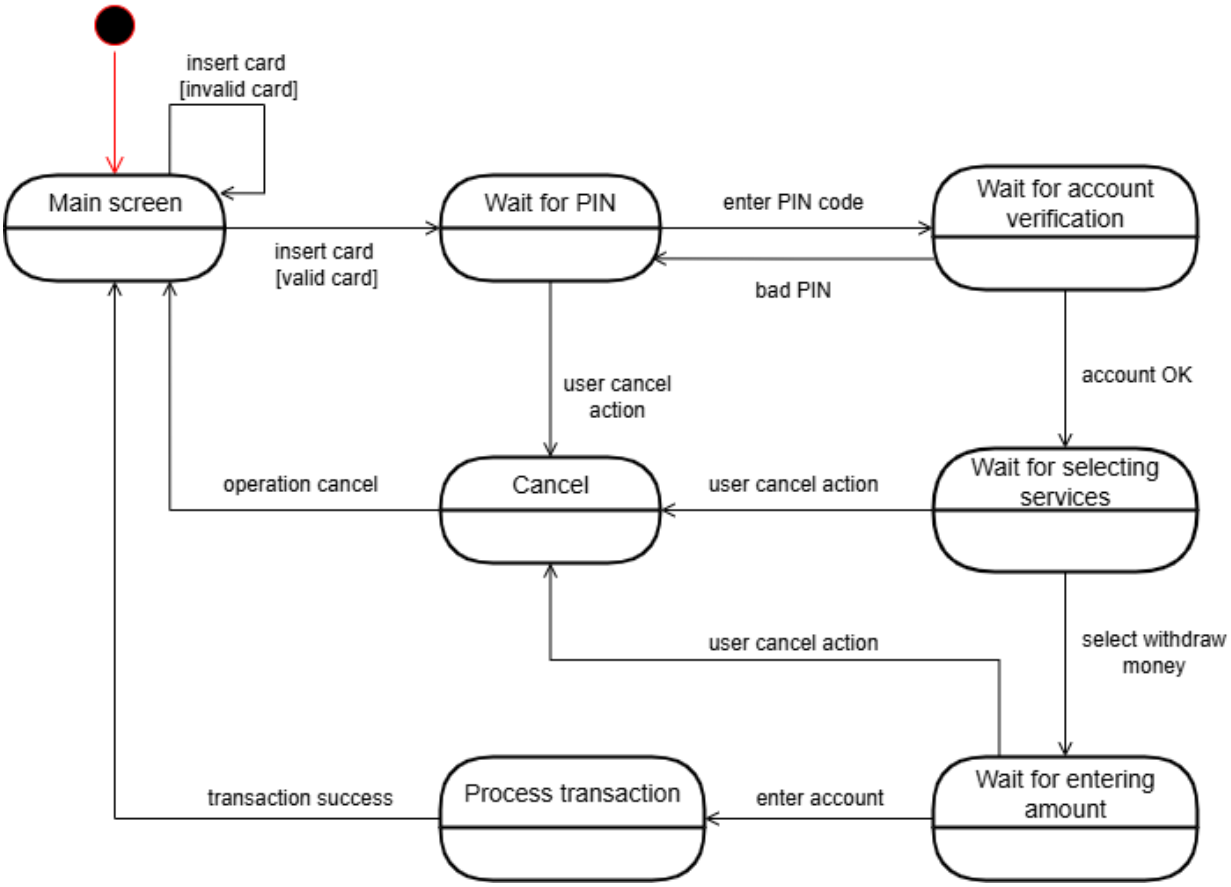
1. Người dùng nạp tiền (Insert) và chọn sản phẩm (Select).
2. Màn hình gửi thông tin (số tiền, lựa chọn) đến Register.
3. Register truy vấn Database để lấy giá sản phẩm.
4. Kiểm tra điều kiện:
    - Nếu tiền < giá → Màn hình hiển thị thông báo "Không đủ tiền", cho phép rút lại tiền.
    - Nếu tiền ≥ giá → kiểm tra số lượng trong Database.
        - Nếu sản phẩm hết hàng → thông báo "Sold Out" và gợi ý sản phẩm khác.
        - Nếu còn hàng → Database cập nhật số lượng.
5. Nếu tiền > giá → hệ thống trả lại tiền thừa (Change).
6. Giao dịch kết thúc khi người dùng nhận sản phẩm và/hoặc tiền thừa.

Bài 5: Vẽ lược đồ lớp cho bài toán Quản lý thư viện

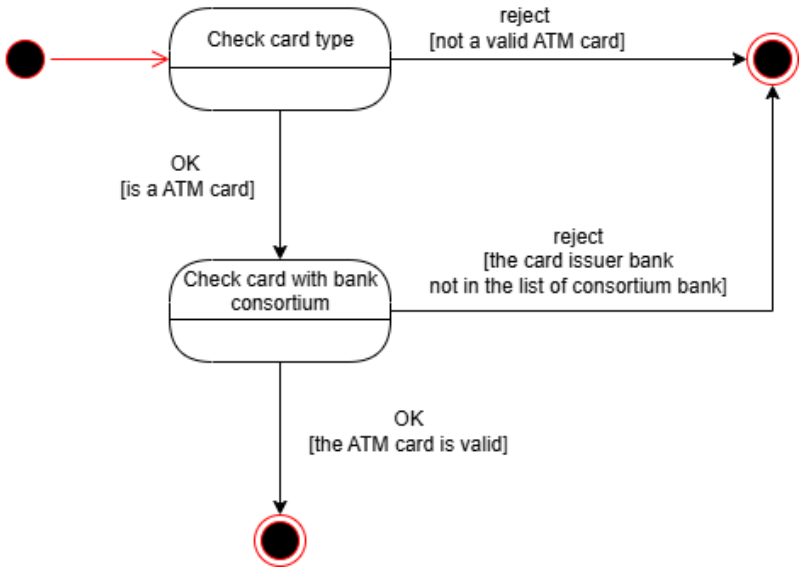


Bài 6: Vẽ lược đồ sau

a) Statechart Diagram for ATM



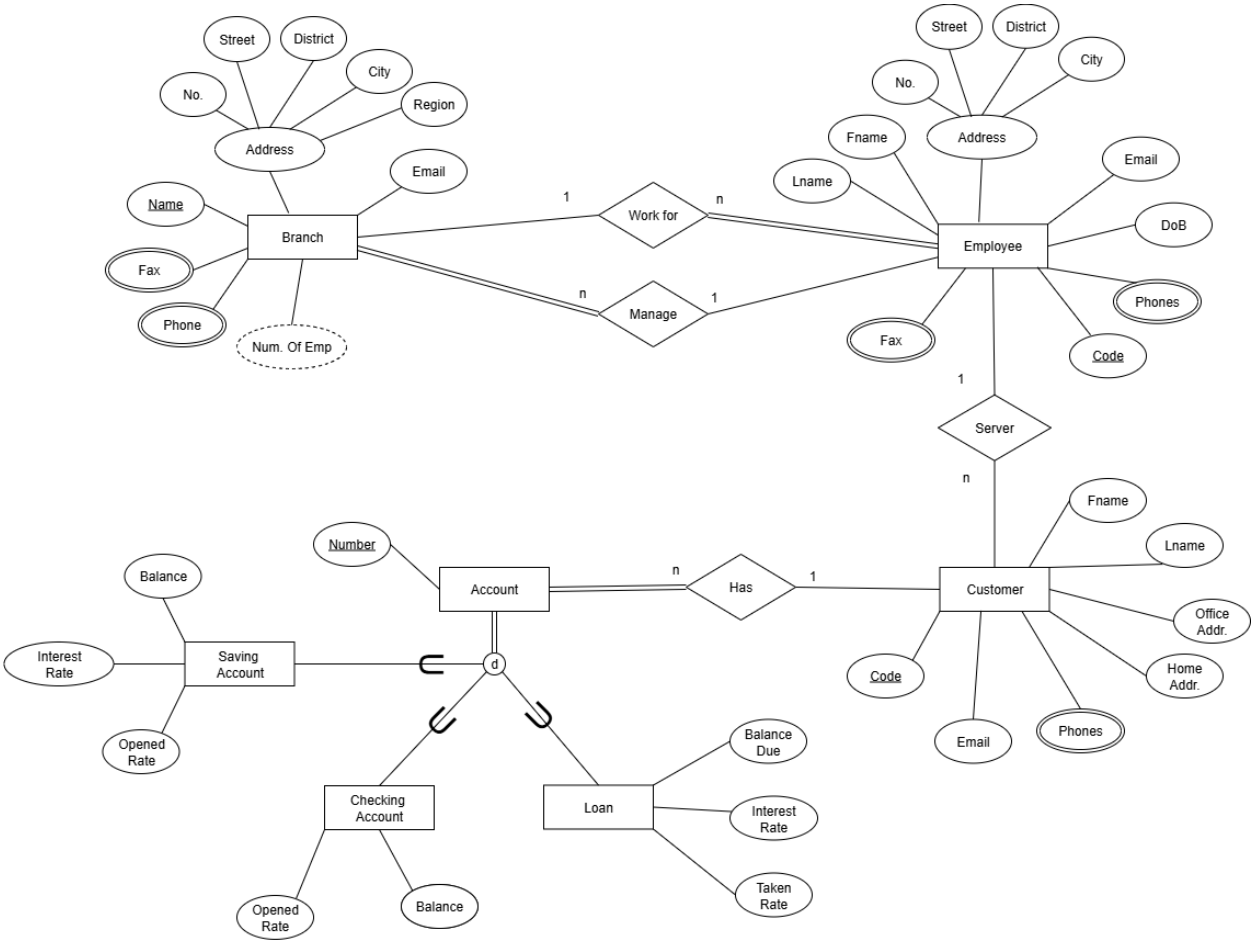
b) Statechart Diagram for Card controller



Bài 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 mức khái niệm



7.2. Thiết kế cơ sở dữ liệu mức luận lý

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)

7.3. Thiết kế cơ sở dữ liệu mức vật lý

7.3.1. Bảng chi nhánh:

```
CREATE TABLE BRANCH (
    Name        VARCHAR(100) PRIMARY KEY,
    AddressNo   VARCHAR(20),
    Street      VARCHAR(100),
    District    VARCHAR(100),
    City        VARCHAR(100),
    Region      VARCHAR(100),
    Email       VARCHAR(100),
    MngCode     VARCHAR(20)
);
```

7.3.2. Bảng số điện thoại chi nhánh:

```
CREATE TABLE BRANCHPHONE (
    BraName     VARCHAR(100),
    Phone       VARCHAR(20),
    PRIMARY KEY (BraName, Phone),
    FOREIGN KEY (BraName) REFERENCES BRANCH(Name) ON DELETE CASCADE
);
```

7.3.3. Bảng fax chi nhánh:

```
CREATE TABLE BRANCHFAX (
    BraName     VARCHAR(100),
    Fax         VARCHAR(20),
    PRIMARY KEY (BraName, Fax),
    FOREIGN KEY (BraName) REFERENCES BRANCH(Name) ON DELETE CASCADE
);
```

7.3.4. Bảng nhân viên:

```sql
CREATE TABLE EMPLOYEE (
    Code        VARCHAR(20) PRIMARY KEY,
    FName       VARCHAR(50),
    LName       VARCHAR(50),
    AddressNo   VARCHAR(20),
    Street      VARCHAR(100),
    District    VARCHAR(100),
    City        VARCHAR(100),
    DOB         DATE,
    Email       VARCHAR(100),
    BraName     VARCHAR(100),
    FOREIGN KEY (BraName) REFERENCES BRANCH(Name) ON DELETE SET NULL
);
```

7.3.5. Bảng số điện thoại nhân viên:

```sql
CREATE TABLE EMPLOYEEPHONE (
    EmpCode     VARCHAR(20),
    Phone       VARCHAR(20),
    PRIMARY KEY (EmpCode, Phone),
    FOREIGN KEY (EmpCode) REFERENCES EMPLOYEE(Code) ON DELETE CASCADE
);
```

7.3.6. Bảng khách hàng:

```sql
CREATE TABLE CUSTOMER (
    Code            VARCHAR(20) PRIMARY KEY,
    FName           VARCHAR(50),
    LName           VARCHAR(50),
    OfficeAddress   VARCHAR(200),
    HomeAddress     VARCHAR(200),
    Email           VARCHAR(100),
    EmpCode         VARCHAR(20),
    FOREIGN KEY (EmpCode) REFERENCES EMPLOYEE(Code) ON DELETE SET NULL
);
```

7.3.7. Bảng số điện thoại khách hàng:

```sql
CREATE TABLE CUSTOMERPHONE (
    CusCode     VARCHAR(20),
    Phone       VARCHAR(20),
    PRIMARY KEY (CusCode, Phone),
    FOREIGN KEY (CusCode) REFERENCES CUSTOMER(Code) ON DELETE CASCADE
);
```

7.3.8. Bảng tài khoản chung:

```
CREATE TABLE ACCOUNT (
    Number      VARCHAR(20) PRIMARY KEY,
    CusCode     VARCHAR(20),
    FOREIGN KEY (CusCode) REFERENCES CUSTOMER(Code) ON DELETE CASCADE
);
```

7.3.9. Bảng tài khoản tiết kiệm:

```
CREATE TABLE SAVINGACCOUNT (
    AccNumber       VARCHAR(20) PRIMARY KEY,
    Balance         NUMERIC(15,2),
    InterestRate    NUMERIC(5,2),
    OpenedDate      DATE,
    FOREIGN KEY (AccNumber) REFERENCES ACCOUNT(Number) ON DELETE CASCADE
);
```

7.3.10. Bảng tài khoản thanh toán:

```
CREATE TABLE CHECKINGACCOUNT (
    AccNumber   VARCHAR(20) PRIMARY KEY,
    Balance     NUMERIC(15,2),
    OpenedDate  DATE,
    FOREIGN KEY (AccNumber) REFERENCES ACCOUNT(Number) ON DELETE CASCADE
);
```

7.3.11. Bảng khoản vay:

```
CREATE TABLE LOAN (
    AccNumber       VARCHAR(20) PRIMARY KEY,
    BalanceDue      NUMERIC(15,2),
    InterestRate    NUMERIC(5,2),
    TakenDate       DATE,
    FOREIGN KEY (AccNumber) REFERENCES ACCOUNT(Number) ON DELETE CASCADE
);
```

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

1. Insert vào Bảng Branch:

```
INSERT INTO BRANCH VALUES
('Branch_HCM', '12A', 'Nguyen Hue', '1', 'HCM', 'South', 'hcm@bank.com', 'M101'),
('Branch_HN', '45B', 'Tran Hung Dao', 'Hoan Kiem', 'Hanoi', 'North', 'hn@bank.com', 'M102'),
('Branch_DN', '78C', 'Le Duan', 'Hai Chau', 'Da Nang', 'Central', 'dn@bank.com', 'M103'),
('Branch_CT', '90D', 'Hoa Binh', 'Ninh Kieu', 'Can Tho', 'Mekong', 'ct@bank.com', 'M104');
```

2. Insert vào Bảng BranchPhone:

```
INSERT INTO BRANCHPHONE VALUES
('Branch_HCM', '028-1111'),
('Branch_HN', '024-2222'),
('Branch_DN', '0236-3333'),
('Branch_CT', '0292-4444');
```

3. Insert vào Bảng BranchFax:

```
INSERT INTO BRANCHFAX VALUES
('Branch_HCM', '028-5555'),
('Branch_HN', '024-6666'),
('Branch_DN', '0236-7777'),
('Branch_CT', '0292-8888');
```

4. Insert vào Bảng Employee:

```
INSERT INTO EMPLOYEE VALUES
('N1001', 'Nguyen', 'An', '10', 'Le Loi', '1', 'HCM', '1990-05-12', 'an.nguyen@bank.com', 'Branch_HCM'),
('N1002', 'Tran', 'Binh', '22', 'Hang Bai', 'Hoan Kiem', 'Hanoi', '1988-03-20', 'binh.tran@bank.com', 'Branch_HN'),
('N1003', 'Le', 'Chi', '33', 'Nguyen Van Linh', 'Hai Chau', 'Da Nang', '1992-07-08', 'chi.le@bank.com', 'Branch_DN'),
('N1080', 'Peter', 'Johnson', '44', 'Hoa Binh', 'Ninh Kieu', 'Can Tho', '2010-03-15', 'peter.johnson@bank.com', 'Branch_CT');
```

5. Insert vào Bảng EmployeePhone:

```
INSERT INTO EMPLOYEEPHONE VALUES
('N1001', '0909000001'),
('N1002', '0912000002'),
('N1003', '0933000003'),
('N1080', '0944000004');
```

6. Insert vào Bảng Customer:

```
INSERT INTO CUSTOMER VALUES
('C2001', 'Vo', 'Khanh', '12 Tran Hung Dao, HCM', '25 Nguyen Hue, HCM', 'khanh.vo@gmail.com', 'N1001'),
('C2002', 'Ho', 'Lan', '34 Ly Thuong Kiet, HN', '56 Hang Bai, HN', 'lan.ho@gmail.com', 'N1002'),
('C2003', 'Nguyen', 'Minh', '78 Le Loi, Da Nang', '99 Nguyen Van Linh, DN', 'minh.nguyen@gmail.com', 'N1003'),
('C2004', 'Tran', 'Hoa', '22 Cach Mang, Can Tho', '44 Hoa Binh, CT', 'hoa.tran@gmail.com', 'N1080');
```

7. Insert vảo Bảng CustomerPhone:

```
INSERT INTO CUSTOMERPHONE VALUES
('C2001', '0901234567'),
('C2002', '0912345678'),
('C2003', '0923456789'),
('C2004', '0934567890');
```

8. Insert vảo Bảng Account:

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

9. Insert vảo Bảng SavingAccount:

```
INSERT INTO SAVINGACCOUNT VALUES
('A3001', 50000000.00, 5.5, '2022-01-01'),
('A3002', 30000000.00, 5.0, '2022-02-15'),
('A3003', 70000000.00, 6.0, '2022-03-10'),
('A3004', 25000000.00, 4.8, '2022-04-20');
```

10. Insert vảo Bảng CheckingAccount:

```
INSERT INTO CHECKINGACCOUNT VALUES
('A3001', 10000000.00, '2022-01-01'),
('A3002', 15000000.00, '2022-02-15'),
('A3003', 20000000.00, '2022-03-10'),
('A3004', 12000000.00, '2022-04-20');
```

11. Insert vảo Bảng Loan:

```
INSERT INTO LOAN VALUES
('A3001', 30000000.00, 8.0, '2022-05-01'),
('A3002', 20000000.00, 7.5, '2022-06-01'),
('A3003', 40000000.00, 8.2, '2022-07-01'),
('A3004', 15000000.00, 7.0, '2022-08-01');
```

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

Query | Query History

```
1  UPDATE EMPLOYEE
2  SET LName = 'Brown'
3  WHERE Code = 'N1080';
4
```

Data Output | Messages | Notifications

```
UPDATE 1

Query returned successfully in 66 msec.
```

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

Query Query History

```
1  DELETE FROM EMPLOYEE
2  WHERE Code = 'N1080';
```

Data Output Messages Notifications

```
DELETE 1

Query returned successfully in 92 msec.
```

Giải thích: CUSTOMER.EmpCode tham chiếu EMPLOYEE(Code) với ON DELETE SET NULL. Nên khi xóa nhân viên N1080, các khách hàng do nhân viên này phục vụ sẽ KHÔNG bị xóa, mà EmpCode của họ sẽ thành NULL

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

Query Query History

```
1  SELECT a.Number
2  FROM ACCOUNT a
3  JOIN CUSTOMER c ON a.CusCode = c.Code
4  WHERE c.FName = 'Peter' AND c.LName = 'Johnson';
```

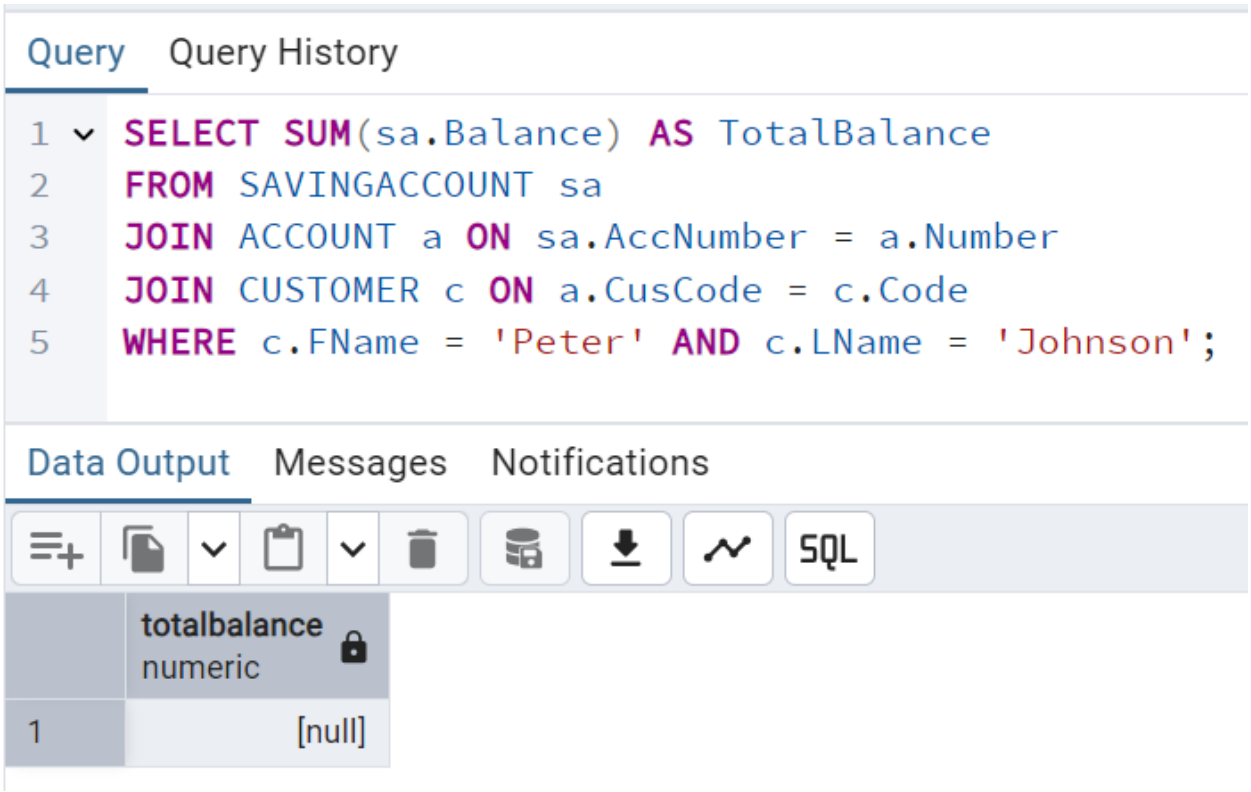
Data Output Messages Notifications

number [PK] character varying (20)

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

f) Retrieve the total balance of all saving accounts owned by a customer named "Peter Johnson"

(Supposing that he has many saving accounts)



Bài 8. Bài tập ứng dụng

Tham khảo: https://vietnam-devs.github.io/coolstore-microservices

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. Hãy liệt kê các yêu cầu chức năng và phi chức năng của hệ thống CoolStore dựa trên mô tả kịch bản nghiệp vụ

Yêu cầu chức năng:

1. Danh mục sản phẩm (Product catalog):

- Người mua có thể duyệt danh sách sản phẩm với chức năng lọc và sắp xếp theo tên sản phẩm và giá.
- Người mua có thể xem chi tiết sản phẩm bằng cách nhấp vào sản phẩm từ trang danh sách.
- 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 có thể quản lý sản phẩm (thêm, sửa, xóa) và có quyền gán sản phẩm vào kho hiện có.

2. Giỏ hàng (Shopping cart)

- 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, hệ thống hiển thị giỏ hàng với các sản phẩm đã thêm và bảng tổng hợp: 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.
- 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.

3. Quy trình thanh toán (Payment process)

- 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.

4. Tồn kho (Inventory)

- Quản trị viên có thể quản lý kho hàng (thêm, sửa xóa)

5. Đánh giá (Rating)

- Người mua có thể đánh giá bất kỳ sản phẩm nào mà cô ấy thấy tốt (thang điểm 1 đến 5 sao).

6. Kiểm soát truy cập (Access control)

- Người mua hoặc quản trị viên có thể đăng nhập/đăng xuất hệ thống.
- Sau đăng nhập: Người mua được chuyển đến trang danh mục sản phẩm; Quản trị viên đến trang quản trị.

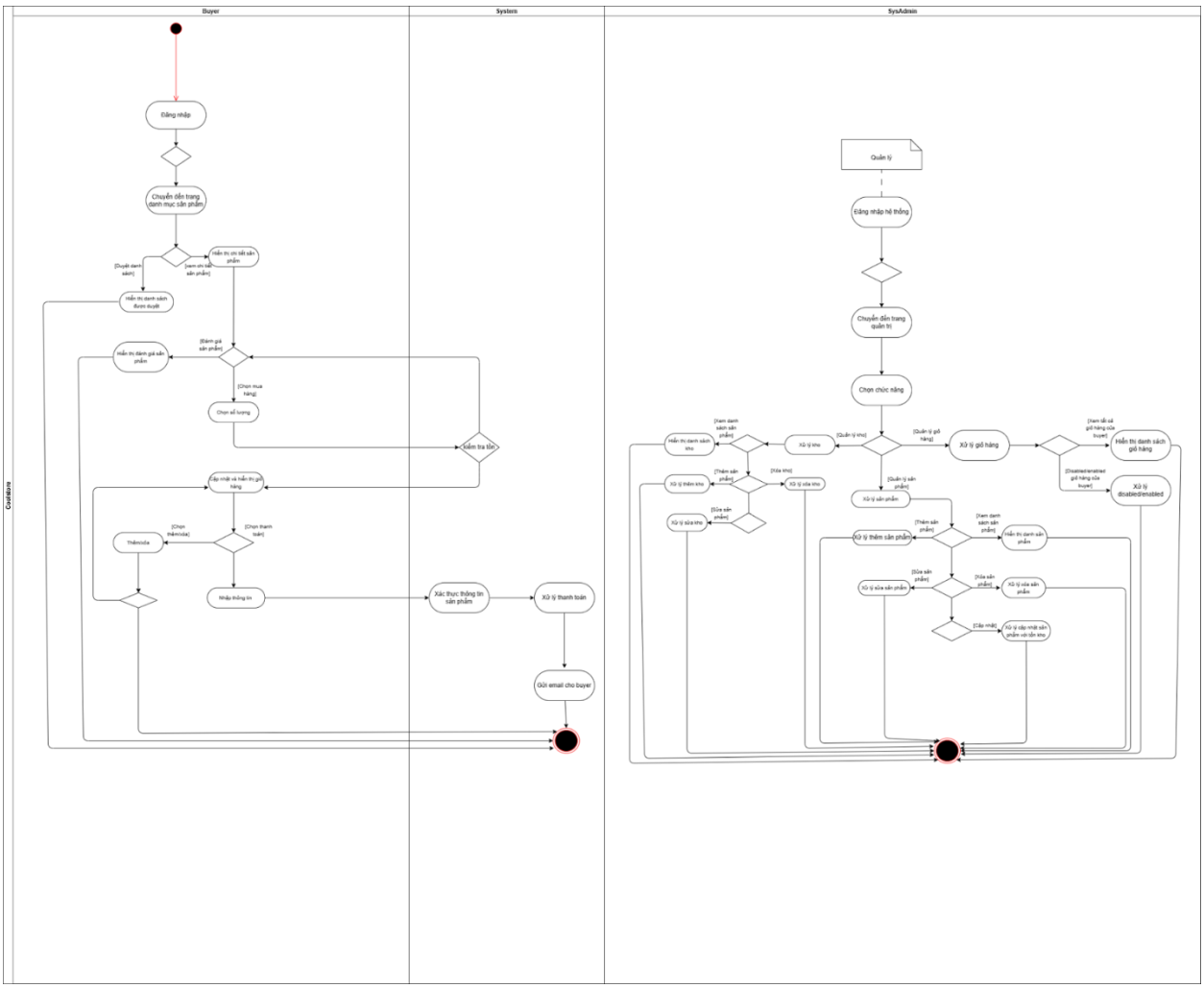
7. Thiết lập ban đầu

- 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.

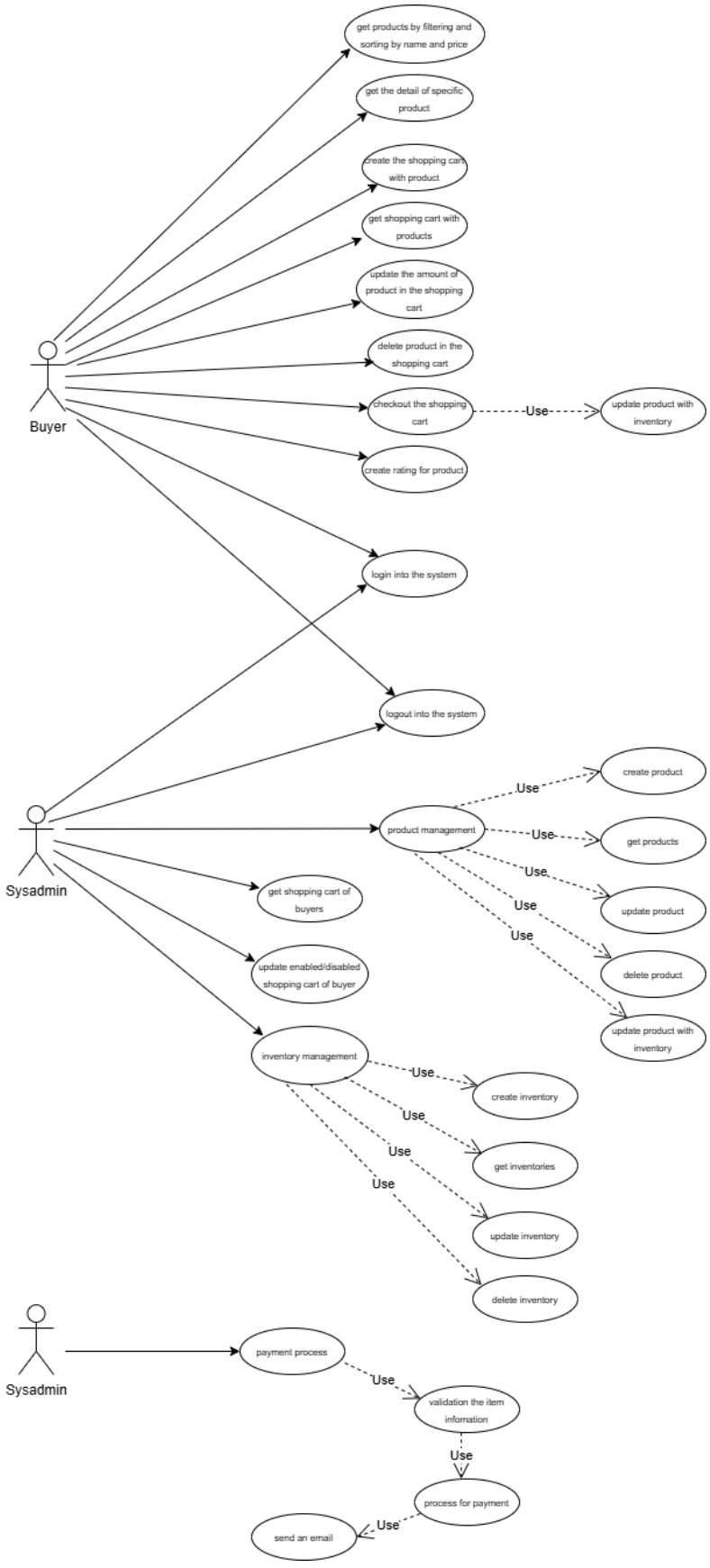
Yêu cầu phi chức năng:

- **Bảo mật (security):** Mã hóa thông tin người dùng, toàn bộ quá trình thanh toán phải an toàn, sử dụng HTTPS.
- **Sự tiện lợi (usability):** Giao diện trực quan, dễ sử dụng, hệ thống phải dễ sử dụng – phải đạt một test usability
- **Sự tin cậy (reliability):** Các số liệu như Thời gian trung bình giữa các lần hỏng hóc (MTBF), hệ thống duy trì 99.9% thời gian hoạt động với MTBF $\geq$ 100 giờ và thời gian trung bình để phục hồi (MTTR) $\leq$ 2 phút sau sự cố.
- **Hiệu năng (performance):** Thời gian phản hồi của hệ thống không vượt quá x giây cho các thao tác chính như filter, sort, thêm vào giỏ hàng và thanh toán.
- **Khả năng chịu đựng (supportability):** Hệ thống cần là ứng dụng Web, chạy được tại tất cả các hệ điều hành và hầu hết các trình duyệt, ứng dụng Web phải triển khai được tại các server tiêu chuẩn như GlassFish hoặc Tomcat

8.2. Vẽ quy trình nghiệp vụ cho website CoolStore



8.3. Xây dựng use case diagram cho website CoolStore

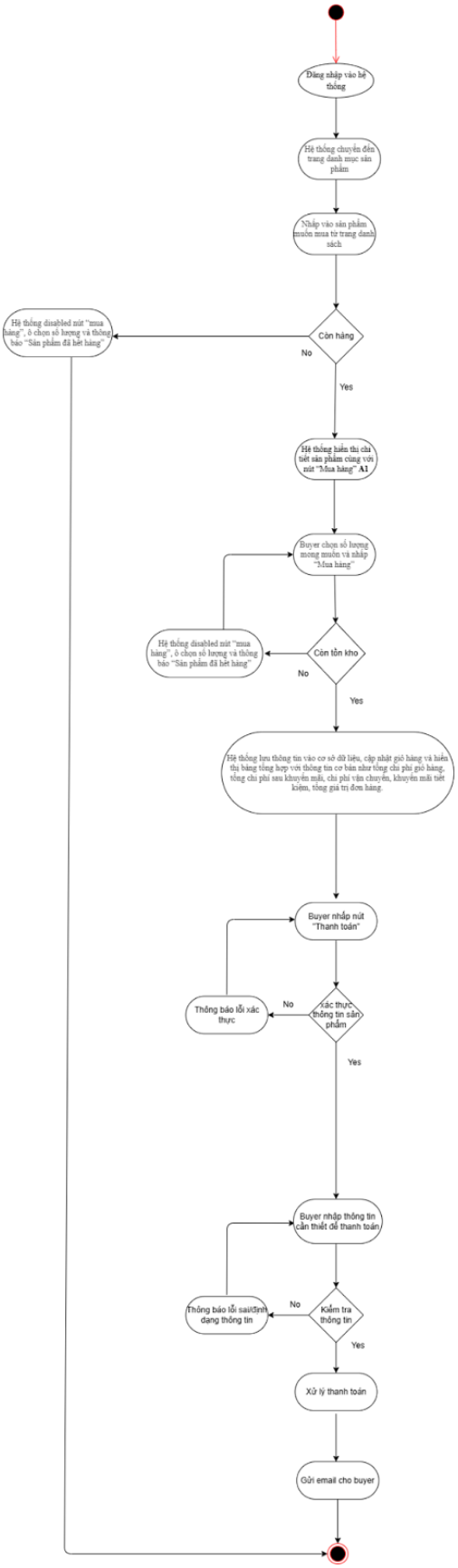

get products by filtering and sorting by name and price
get the detail of specific product
create the shopping cart with product
get shopping cart with products
update the amount of product in the shopping cart
delete product in the shopping cart
checkout the shopping cart
Use
update product with inventory
create rating for product
Buyer
login into the system
logout into the system
Sysadmin
product management
Use
create product
Use
get products
Use
update product
Use
delete product
Use
update product with inventory
get shopping cart of buyers
update enabled/disabled shopping cart of buyer
inventory management
Use
create inventory
Use
get inventories
Use
update inventory
Use
delete inventory
Sysadmin
payment process
Use
validation the item information
Use
process for payment
Use
send an email

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)

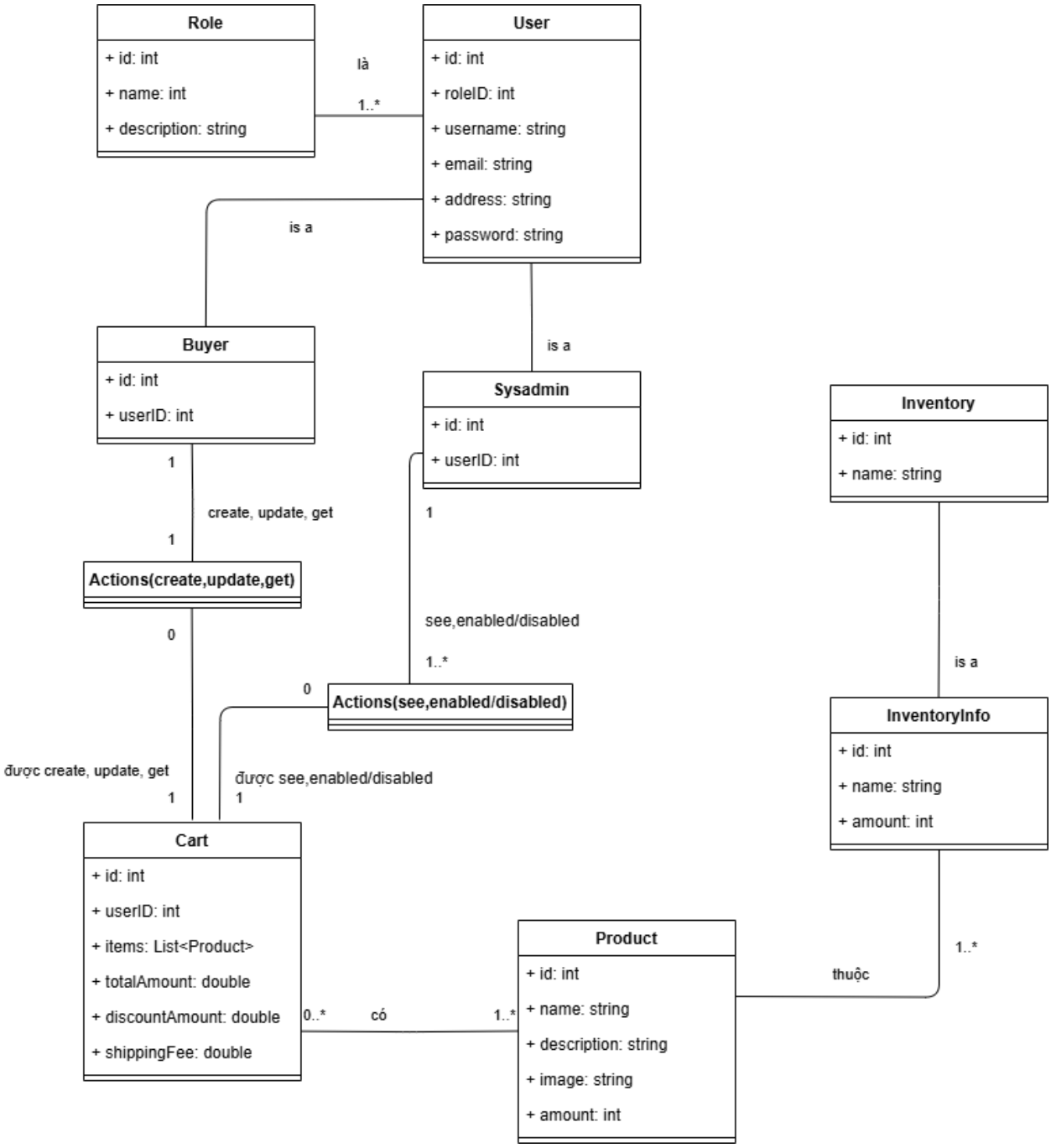
Use Case Name:	Mua sản phẩm từ trang chi tiết sản phẩm	
Actor (s):	Buyer	
Summary:	Thêm sản phẩm vào giỏ hàng từ trang chi tiết sản phẩm.	
Basic Course of Events:	**Actor Action**	**System Response**
	1. Đăng nhập vào hệ thống	
		2. Hệ thống chuyển đến trang danh mục sản phẩm
	3. Nhấp vào sản phẩm muốn mua từ trang danh sách	
		4. Hệ thống hiển thị chi tiết sản phẩm cùng với nút "Mua hàng" **A1**
	5. Chọn số lượng mong muốn và nhấp "Mua hàng"	
		6. Hệ thống kiểm tra tồn kho **E1.**
		7. Hệ thống lưu thông tin vào cơ sở dữ liệu, cập nhật giỏ hàng và hiển thị 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.

	8. Usecase kết thúc tại đây.	
Alternative Paths:	**A1.** Hệ thống disabled nút "mua hàng", ô chọn số lượng và thông báo "Sản phẩm đã hết hàng". Usecase kết thúc tại đây.	
Exception Paths:	**E1.** Hệ thống hiển thị thông báo "Số lượng vượt quá tồn kho (còn X sản phẩm)". Buyer có thể chỉnh lại số lượng và quay lại bước 5.	
Preconditions:	Buyer đã đăng nhập vào hệ thống, đang ở trang chi tiết, sản phẩm còn hàng (số lượng > 0).	
Post Conditions:	Mục được thêm vào giỏ hàng với số lượng chính xác. Bảng tổng hợp giỏ hàng được cập nhật. Cập nhật số lượng tồn kho	
Note	None	

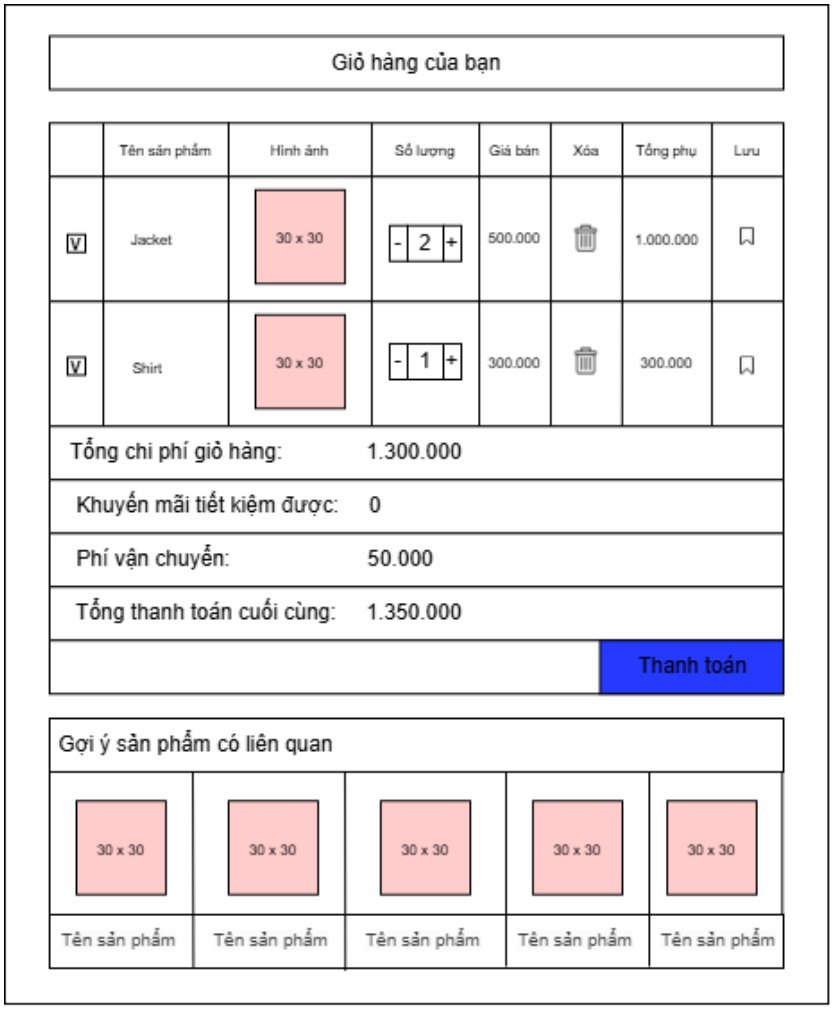
8.5. Sử dụng sơ đồ Activity Diagram để mô tả quy trình mua hàng từ khi người dùng nhấn "Mua hàng" cho đến khi hoàn tất thanh toán


Đăng nhập vào hệ thống
Hệ thống chuyển đến trang danh mục sản phẩm
Nhấp vào sản phẩm muốn mua từ trang danh sách
Còn hàng
No
Hệ thống disabled nút "mua hàng", ô chọn số lượng và thông báo "Sản phẩm đã hết hàng"
Yes
Hệ thống hiển thị chi tiết sản phẩm cùng với nút "Mua hàng" A1
Buyer chọn số lượng mong muốn và nhấp "Mua hàng"
Còn tồn kho
No
Hệ thống disabled nút "mua hàng", ô chọn số lượng và thông báo "Sản phẩm đã hết hàng"
Yes
Hệ thống lưu thông tin vào cơ sở dữ liệu, cập nhật giỏ hàng và hiển thị 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.
Buyer nhấp nút "Thanh toán"
xác thực thông tin sản phẩm
No
Thông báo lỗi xác thực
Yes
Buyer nhập thông tin cần thiết để thanh toán
Kiểm tra thông tin
No
Thông báo lỗi saiĩnh dạng thông tin
Yes
Xử lý thanh toán
Gửi email cho buyer

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


Giỏ hàng của bạn
Tên sản phẩm
Hình ảnh
Số lượng
Giá bán
Xóa
Tổng phụ
Lưu
Jacket
30 x 30
2
500.000
1.000.000
Shirt
30 x 30
1
300.000
300.000
Tổng chi phí giỏ hàng: 1.300.000
Khuyến mãi tiết kiệm được: 0
Phí vận chuyển: 50.000
Tổng thanh toán cuối cùng: 1.350.000
Thanh toán
Gợi ý sản phẩm có liên quan
30 x 30
Tên sản phẩm

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): là giao diện người dùng cuối, bao gồm các trang web hoặc ứng dụng di động mà các actors admin/buyer tương tác trực tiếp. Lớp này chịu trách nhiệm nhận yêu cầu người dùng và hiển thị dữ liệu trả về từ backend.

1.1. Giao diện admin của hệ thống CoolStore

- Trang đăng nhập cho quản trị viên.
- Dashboard hiển thị tổng quan hệ thống: tổng số sản phẩm, giỏ hàng, tồn kho.
- Trang quản lý sản phẩm: thêm, sửa, xóa sản phẩm, gán sản phẩm vào kho.
- Trang quản lý kho hàng: thêm/sửa kho, quản lý tồn kho.
- Trang quản lý giỏ hàng: xem toàn bộ giỏ hàng của người dùng, kích hoạt/vô hiệu hóa giỏ hàng không hợp lệ.

1.2. Giao diện buyer của hệ thống CoolStore

- Trang đăng nhập cho người mua.
- Trang danh mục sản phẩm: hiển thị danh sách sản phẩm, hỗ trợ lọc và sắp xếp theo tên/giá.
- Trang chi tiết sản phẩm: hiển thị thông tin sản phẩm, tồn kho, đánh giá, nút “Mua hàng”.
- Trang giỏ hàng: hiển thị sản phẩm đã chọn, hỗ trợ tăng/giảm số lượng, xóa sản phẩm, lưu để mua sau.
- Trang thanh toán: xác nhận đơn hàng, nhập thông tin thanh toán, đặt hàng.
- Trang đánh giá sản phẩm: để lại đánh giá từ 1–5 sao.

1.3. IT system

- Xác thực người dùng / phân quyền hiển thị giao diện phù hợp (Buyer hoặc Admin).
- Sử dụng REST API endpoints để kết nối Presentation Layer với Business Logic Layer.

2. Business logic layer: lớp này xử lý các logic nghiệp vụ

2.1. Admin

- Quản lý danh mục sản phẩm: thêm mới, chỉnh sửa, xóa sản phẩm, xem danh sách sản phẩm, gán sản phẩm vào kho
- Quản lý giỏ hàng: xem tất cả sản phẩm trong giỏ hàng của người mua; kích hoạt/ vô hiệu hóa giỏ hàng không hợp lệ
- Quản lý tồn kho: thêm, cập nhật, xóa tồn kho, xem danh sách tồn kho

2.2. Buyer

- Quản lý giỏ hàng: thêm sản phẩm vào giỏ hàng, kiểm tra số lượng tồn kho trước khi thêm; tăng/giảm số lượng sản phẩm trong giỏ, tự động cập nhật tổng tiền; xóa sản phẩm khỏi giỏ hàng.
- Quản lý danh mục sản phẩm: lọc sản phẩm theo tên/giá, xem danh sách sản phẩm, xem chi tiết sản phẩm.
- Thanh toán: xác thực thông tin giỏ hàng (kiểm tra tồn kho, tính chi phí); xử lý thanh toán (gửi yêu cầu thanh toán tới payment gateway); tạo đơn hàng; gửi email xác nhận
- Đánh giá sản phẩm: lưu đánh giá sản phẩm, tính trung bình rating.

2.3. Xác thực, phân quyền

- Đăng nhập/đăng xuất, kiểm tra quyền truy cập trước khi gọi API
- Thiết lập các quy tắc hệ thống: giới hạn số lần đăng nhập thất bại trước khi khóa tạm thời tài khoản.

3. Data access layer: lớp này chịu trách nhiệm lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu, tách biệt hoàn toàn khỏi logic nghiệp vụ, gồm có các lớp sau.

User:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã user, khóa chính
roleID	int (PK)	phân quyền
username	varchar	tên đăng nhập
email	varchar	địa chỉ email
address	varchar	địa chỉ nhà
password	varchar	mật khẩu

Role:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã role, khóa chính
name	varchar	tên phân quyền
description	varchar	mô tả phân quyền

Product:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã sản phẩm, khóa chính
name	varchar	tên sản phẩm
description	varchar	mô tả sản phẩm
image	varchar	hình ảnh
amount	int	số lượng
inventory_id	int (FK)	liên kết kho

Inventory:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã kho, khóa chính
name	varchar	tên kho
amount	int	số lượng

Cart:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã giỏ hàng
userID	int (FK)	mã chủ giỏ hàng

CartItem:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã giỏ hàng
cartID	int (FK)	mã giỏ hàng
productID	int (FK)	sản phẩm
quantity	int	số lượng

Order:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã giỏ hàng
userID	int (FK)	mã người mua
total_price	double	tổng tiền
total_amount	double	tổng số lượng
status	double	trạng thái

Payment:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã

orderID	int (FK)	mã giỏ hàng
method	varchar	phương thức thanh toán
status	varchar	trạng thái

Rating:

Thuộc tính	Kiểu dữ liệu	Mô tả
id	int (PK)	mã
userID	int (FK)	mã người đánh giá
productID	int (FK)	sản phẩm
rating	int	điểm đánh giá

8.9. Viết ít nhất 3 ca 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 với số lượng hợp lệ

- Dữ liệu đầu vào: productID = 1, quantity = 2 (tồn kho còn 5)

- Bước thực hiện:

1. Đăng nhập vào hệ thống
2. Truy cập danh sách sản phẩm
3. Chọn sản phẩm có productID = 1
4. Nhập số lượng (quantity) = 2
5. Nhấp nút "Mua hàng"

- Kết quả mong đợi: sản phẩm được thêm vào giỏ hàng, giỏ hàng hiển thị sản phẩm với đúng số lượng được thêm, bảng tổng hợp giỏ hàng được cập nhật chính xác.

Test case 2: Thêm sản phẩm với số lượng vượt quá tồn kho

- Dữ liệu đầu vào: productID = 1, quantity = 10 (tồn kho còn 5)

- Bước thực hiện:
    1. Đăng nhập vào hệ thống
    2. Truy cập danh sách sản phẩm
    3. Chọn sản phẩm có productID = 1
    4. Nhập số lượng (quantity) = 10
    5. Nhấp nút “Mua hàng”

- Kết quả mong đợi: hệ thống hiển thị thông báo “Số lượng vượt quá tồn kho” và sản phẩm không được thêm vào giỏ hàng,

Test case 3: Thêm sản phẩm với đầu vào không hợp lệ

- Dữ liệu đầu vào: productID = 1, quantity = -5 hoặc “xyz” (tồn kho còn 5)
- Bước thực hiện:
    1. Đăng nhập vào hệ thống
    2. Truy cập danh sách sản phẩm
    3. Chọn sản phẩm có productID = 1
    4. Nhập số lượng (quantity) = -5 hoặc “xyz”
    5. Nhấp nút “Mua hàng”

- Kết quả mong đợi: hệ thống hiển thị thông báo “Số lượng không hợp lệ” và sản phẩm không được thêm vào giỏ hàng,

Test case 4: Thêm sản phẩm khi chưa đăng nhập

- Dữ liệu đầu vào: productID = 1, quantity = 1 (tồn kho còn 5)
- Bước thực hiện:
    1. Không đăng nhập vào hệ thống
    2. Truy cập danh sách sản phẩm
    3. Chọn sản phẩm có productID = 1
    4. Nhập số lượng (quantity) = 1
    5. Nhấp nút “Mua hàng”

- Kết quả mong đợi: hệ thống chuyển hướng đến trang đăng nhập hoặc hiển thị thông báo yêu cầu đăng nhập trước khi mua hàng