Nguyễn Thanh Tịnh - 3122411215

Nguyễn Tấn Thành - 3122411192



ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC SÀI GÒN

KIỂM THỬ PHẦN MỀM

NGÀNH : CÔNG NGHỆ THÔNG TIN

Giảng viên

Sinh viên thực hiện:

TS. Đỗ Như Tài

Nguyễn Thị Diệu Linh - 3122411112

Đỗ Nguyễn Nhật Trường – 3122411230

Thành phố Hồ Chí Minh, 2025

**LAB 2**

**Nhiệm vụ:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Thành viên** | Đỗ Nguyễn Nhật Trường | Nguyễn Thanh Tịnh | Nguyễn Thị Diệu Linh | Nguyễn Tấn Thành |
| **Nhiệm vụ** | - Bài 5,7 phần bài tập minh họa  - Bài 6,8,9 phần bài tập ứng dụng | Bài 1,2,3,4,5 phần bài tập ứng dụng | -Bài 3,4 phần bài tập minh họa  - Bài 7 phần bài tập ứng dụng | Bài 1,2,6 phần bài tập minh họa |

Mục lục

[**I.** **BÀI TẬP MINH HỌA** 3](#_Toc209123256)

[**Câu 1.** **Vẽ lại qui trình nghiệp vụ sau (Độc giả, NV Tiếp tân, Thành viên)** 3](#_Toc209123257)

[**Câu 2.** **Vẽ mô hình khái niệm sau** 5](#_Toc209123258)

[**Câu 3.** **Vẽ lại business use case sau:** 6](#_Toc209123259)

[**Câu 4.** **Vẽ sơ đồ hoạt động và tương tác sau và tóm tắt ý nghĩa sơ đồ** 10](#_Toc209123260)

[**Câu 5.** **Vẽ lược đồ lớp cho bài toán Quản lý thư viện** 12](#_Toc209123261)

[**1. Các thực thể chính (Entity)** 13](#_Toc209123262)

[**2. Các quan hệ (Relationship)** 14](#_Toc209123263)

[**3. Ý nghĩa tổng thể** 14](#_Toc209123264)

[**Câu 6.** **Vẽ lược đồ sau:** 15](#_Toc209123265)

[**Câu 7.** **Phân tích dữ liệu cho ABC Bank** 17](#_Toc209123266)

[**II.** **BÀI TẬP ỨNG DỤNG** 20](#_Toc209123267)

[**Câu 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.** 20](#_Toc209123268)

[**Câu 2.** **Vẽ qui trình nghiệp vụ cho website CoolStore** 26](#_Toc209123269)

[**Câu 3.** **Xây dựng use case diagram cho website CoolStore.** 27](#_Toc209123270)

[**Câu 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.** 30](#_Toc209123271)

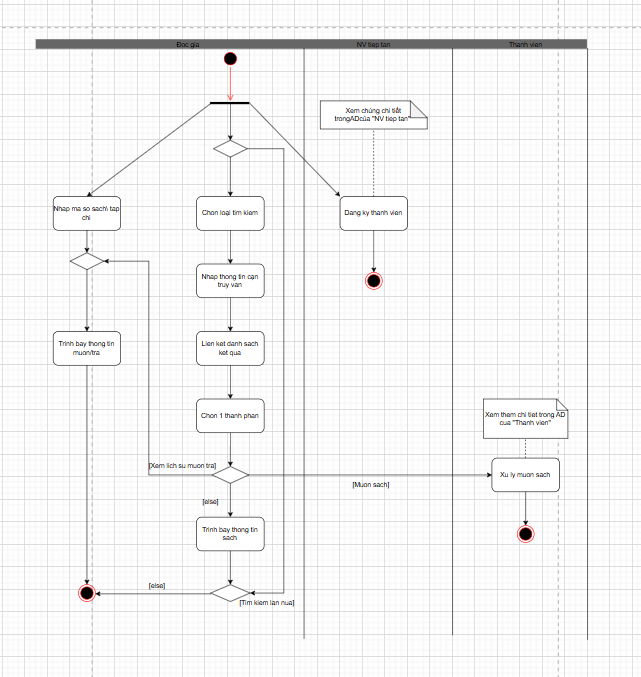
[**Câu 6.** **Thiết kế sơ đồ Class Diagram cho module Giỏ hàng.** 31](#_Toc209123272)

[**Câu 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:** 32](#_Toc209123273)

[**Câu 8.** **Đề xuất cách tổ chức kiến trúc phần mềm cho hệ thống CoolStore theo mô hình kiến trúc ba lớp (3-tier architecture).** 35](#_Toc209123274)

[**Câu 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** 37](#_Toc209123275)

1. **BÀI TẬP MINH HỌA**
2. **Vẽ lại qui trình nghiệp vụ sau (Độc giả, NV Tiếp tân, Thành viên)**



**1. Các "swimlane" (cột vai trò)**

* **Độc giả**: người đến tìm kiếm, mượn sách.
* **NV tiếp tân**: nhân viên thư viện, hỗ trợ đăng ký thành viên.
* **Thành viên**: độc giả sau khi đã đăng ký thành viên.

**2. Luồng hoạt động của Độc giả**

* Bắt đầu → Độc giả chọn một trong các hành động:
  + **Nhập mã số sách/tạp chí** → hệ thống hiển thị thông tin mượn/trả.
  + **Chọn loại tìm kiếm** → Nhập thông tin cần truy vấn → Liên kết danh sách kết quả → Chọn 1 thành phần →
    - Nếu muốn xem → **Trình bày thông tin sách** → Nếu chưa hài lòng → Tìm kiếm lần nữa.
    - Nếu muốn mượn → chuyển sang lane **Thành viên**.
  + **Xem lịch sử mượn/trả** → hiển thị thông tin → kết thúc.

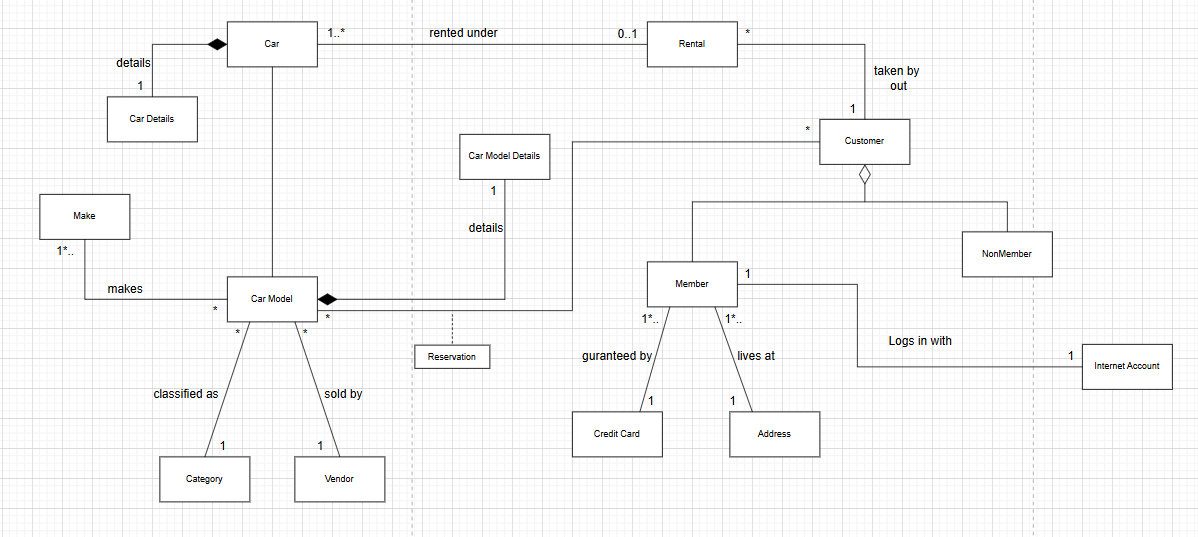
**3. Luồng hoạt động của NV tiếp tân**

* Nếu độc giả chưa là thành viên, có nhánh **Đăng ký thành viên** → sau đó trở thành **Thành viên**.

**4. Luồng hoạt động của Thành viên**

* Thành viên sau khi đăng ký → có thể **mượn sách**.
* Quá trình mượn:
  + Xem chi tiết thông tin trong Actor "Thành viên".
  + Nhân viên xử lý yêu cầu mượn sách.
  + Kết thúc quy trình.

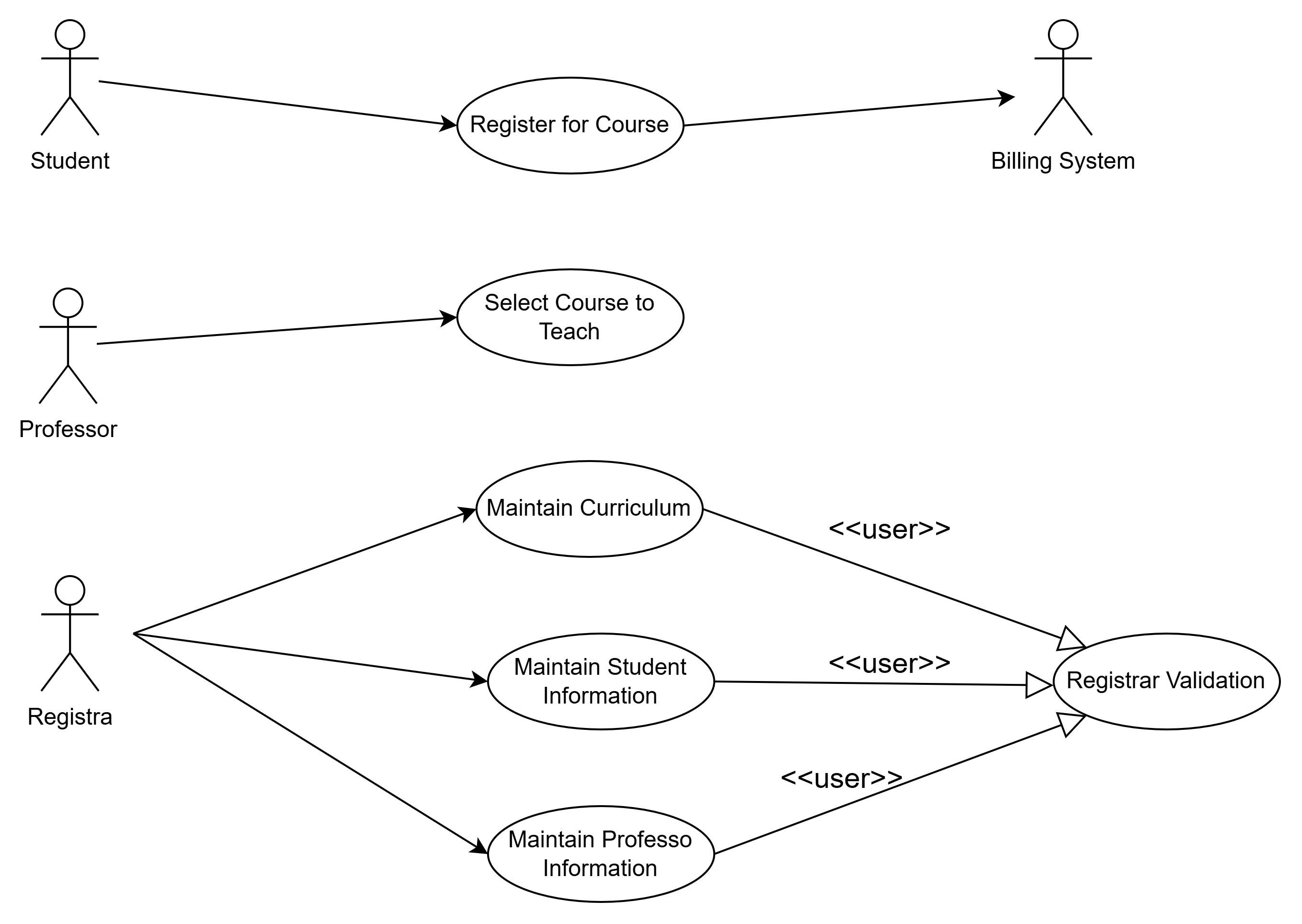
1. **Vẽ mô hình khái niệm sau**



**Các quan hệ chính**

* Một **Car Model** được tạo bởi một **Make**.
* Một **Car Model** được phân loại (Category) và bán bởi (Vendor).
* Một **Car** thuộc một **Car Model** nhưng có thể có nhiều chi tiết khác nhau.
* **Rental** kết nối giữa **Car** và **Customer**.
* **Customer** chia thành **Member** và **NonMember** (quan hệ tổng quát hoá).
* **Member** có thể bảo đảm bằng nhiều **Credit Card**, sống ở một địa chỉ, và đăng nhập qua **Internet Account**.

1. **Vẽ lại business use case sau:**
2. **Xác định và mô tả các tác nhân (Actors) xuất hiện trong sơ đồ trên.**
3. **Liệt kê và giải thích các trường hợp sử dụng (Use cases) được thể hiện trong sơ đồ.**
4. **Phân tích các mối quan hệ giữa các use case (bao gồm cả mối quan hệ «uses» hoặc «include»).**
5. **Viết kịch bản chi tiết cho một trường hợp sử dụng cụ thể (ví dụ: “Register for Courses” hoặc “Maintain Student Information”).**
6. **Đề xuất cải tiến hoặc bổ sung thêm các use case mà bạn thấy cần thiết.**



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

**-Student** (Sinh viên): đăng ký môn học, tương tác với hệ thống thanh toán học phí.

**-Professor** (Giảng viên): chọn các môn học mình sẽ dạy.

**-Registrar** (Phòng đào tạo/Quản trị viên): duy trì (maintain) thông tin chương trình học, thông tin sinh viên, thông tin giảng viên.

**-Billing System** (Hệ thống thanh toán): xử lý việc thanh toán khi sinh viên đăng ký môn học.

**b) Liệt kê và giải thích các Use Case trong sơ đồ**

**-Register for Courses** (Đăng kí môn học): Sinh viên đăng ký môn học, hệ thống kiểm tra học phí qua Billing System.

**-Select Courses to Teach** (Chọn môn học để giảng dạy): Giảng viên chọn các môn học mà mình sẽ giảng dạy trong học kỳ.

**-Maintain Curriculum** (Duy trì chương trình đào tạo): Quản trị viên cập nhật, duy trì chương trình đào tạo.

**-Maintain Student Information** (Duy trì thông tin sinh viên): Quản trị viên cập nhật hồ sơ, thông tin cá nhân, kết quả học tập của sinh viên.

-**Maintain Professor Information** (Duy trì thông tin giảng viên): Quản trị viên cập nhật hồ sơ, thông tin giảng dạy của giảng viên.

**-Registrar Validation** (Xác thực bởi phòng đào tạo): Xác thực/thẩm định dữ liệu bởi phòng đào tạo, được sử dụng bởi nhiều Use Case khác.

**c) Phân tích các mối quan hệ**

**-Maintain Curriculum** «uses» **Registrar Validation**

**-Maintain Student Information** «uses» **Registrar Validation**

**-Maintain Professor Information** «uses» **Registrar Validation**  
 → Các nghiệp vụ duy trì dữ liệu đều cần phải qua bước kiểm tra xác thực.

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

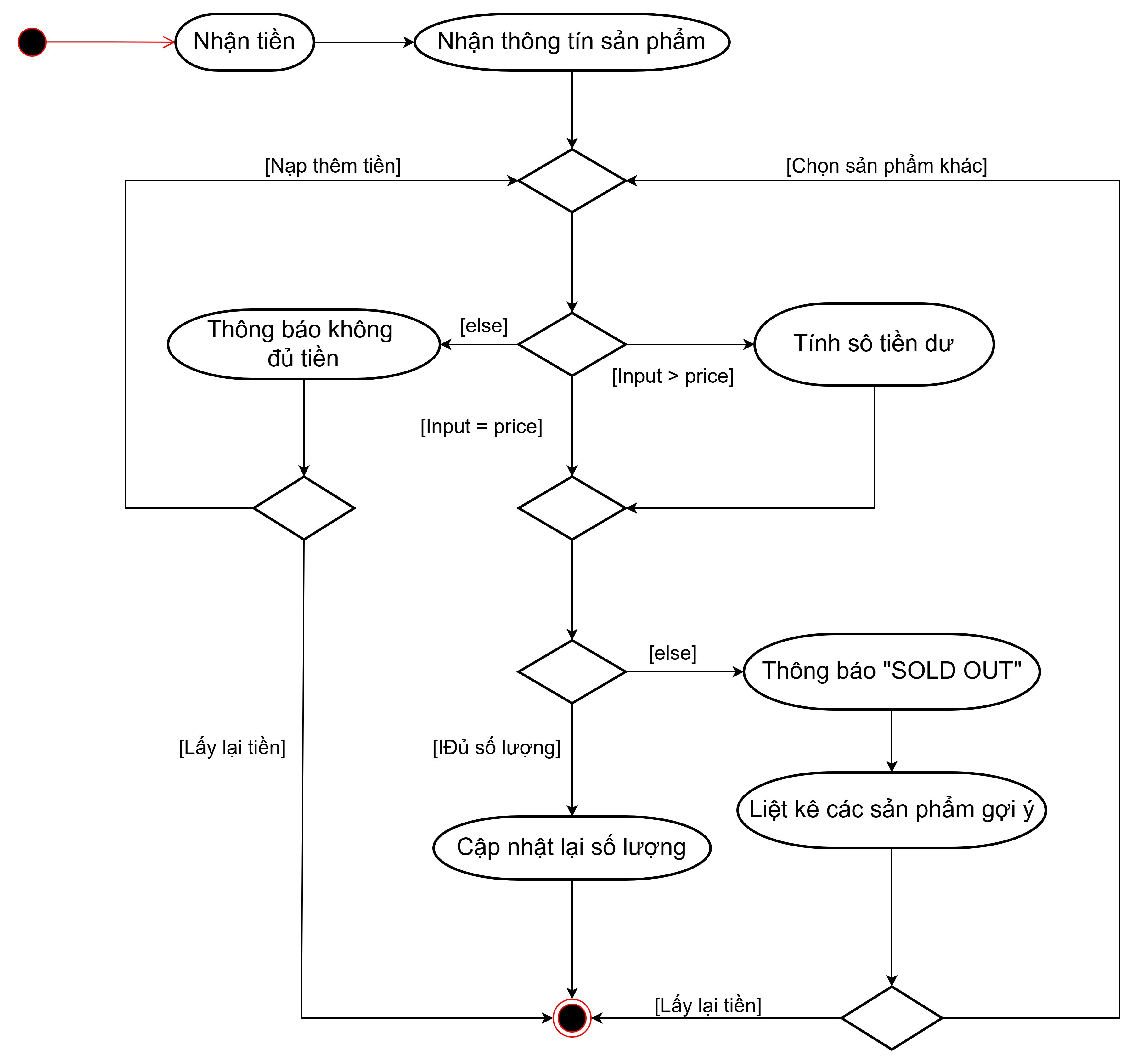
-Kịch bản chi tiết cho **Use Case: Maintain Student Information (Duy trì thông tin sinh viên)**

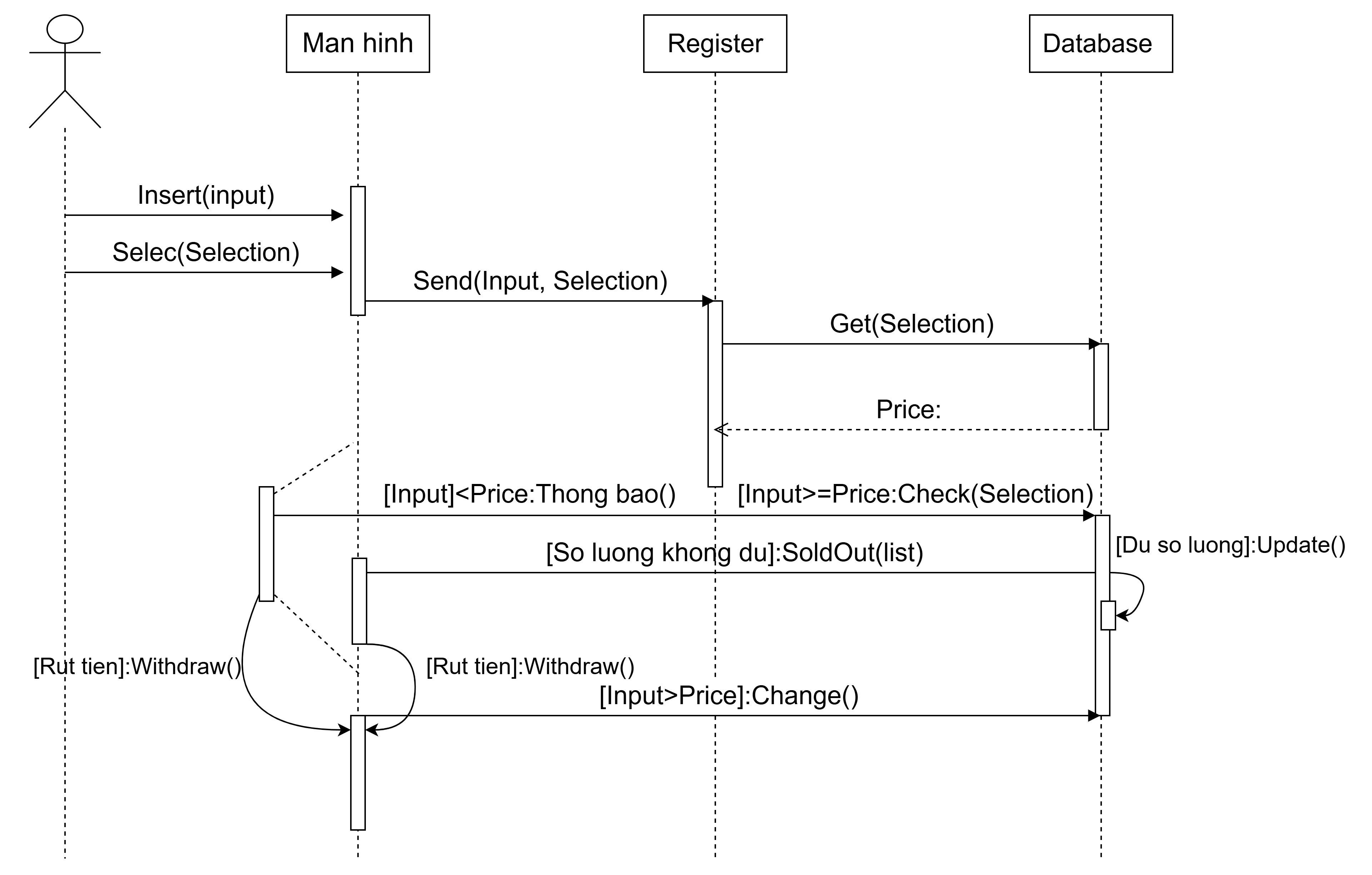
|  |  |  |
| --- | --- | --- |
| **Tên Use Case** | Duy trì thông tin sinh viên (*Maintain Student Information*) | |
| **Tác nhân chính** | Registrar | |
| **Mục tiêu** | Cập nhật, thêm mới, chỉnh sửa, hoặc xóa thông tin sinh viên trong hệ thống. | |
| **Điều kiện tiên quyết** | Quản trị viên đã đăng nhập hệ thống thành công. | |
| **Mô tả** | Thông tin sinh viên được cập nhật hợp lệ và lưu trữ trong hệ thống. | |
|  | Hệ thống | Người dùng |
| **Luồng sự kiện chính** | 1.Hiển thị giao diện đăng nhập |  |
|  | 2.Đăng nhập vào hệ thống.E1 |
|  | 3.Chọn chức năng **“Duy trì thông tin sinh viên”**. |
| 4.Hiển thị danh sách sinh viên |  |
|  | 5.Tìm kiếm hoặc chọn sinh viên.E2 |
|  | 6. Thực hiện thao tác: cập nhật, thêm mới, hoặc xóa |
| 7. Hệ thống gửi dữ liệu đến **Registrar Validation** để xác thực. |  |
| 8. Nếu hợp lệ → hệ thống lưu thay đổi và báo **“Cập nhật thành công”**. |  |
| **Luồng sự kiện thay thế** | **A1.Cập nhật thông tin** | |
|  | 1.Giảng viên chọn đối tượng và chỉnh sửa thông tin |
| 2.Hệ thống kiểm tra dữ liệu → cập nhật vào CSDL và báo thành công.E3 |  |
| **A2 – Thêm mới thông tin** | |
|  | 1. Giảng viên nhập dữ liệu mới |
| 2.Hệ thống kiểm tra dữ liệu → thêm mới vào CSDL và báo thành công.E4 |  |
| **A3 – Xóa thông tin** | |
|  | 1. iảng viên chọn đối tượng cần xóa |
| 2. Hệ thống xác nhận → xóa dữ liệu khỏi CSDL và báo thành công |  |
| **Luồng ngoại lệ** | E1.Thông tin đăng nhập không hợp lệ->Hệ thống báo lỗi và yêu cầu nhập lại.  E2.Sinh viên không tồn tại hoặc không hợp lệ ->Hệ thống báo lỗi và yêu cầu nhập lại.  E3.Thông tin không hợp lệ ->Hệ thống báo lỗi và yêu cầu nhập lại.  E4. Thông tin không hợp lệ ->Hệ thống báo lỗi và yêu cầu nhập lại. | |

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

* **View Course Schedule: Sinh viên xem lịch học các môn đã đăng ký.**
* **Drop Course: Sinh viên hủy đăng ký môn học.**
* **View Transcript: Sinh viên xem kết quả học tập.**
* **Assign Classroom: Quản trị viên phân công phòng học cho từng môn.**
* **Generate Reports: Quản trị viên xuất báo cáo thống kê (số sinh viên đăng ký, số môn học mở…).**

1. **Vẽ sơ đồ hoạt động và tương tác sau và tóm tắt ý nghĩa sơ đồ**





Đây là sơ đồ mô tả quy trình mua hàng trong một hệ thống bán hàng tự động (ví dụ: máy bán hàng tự động).

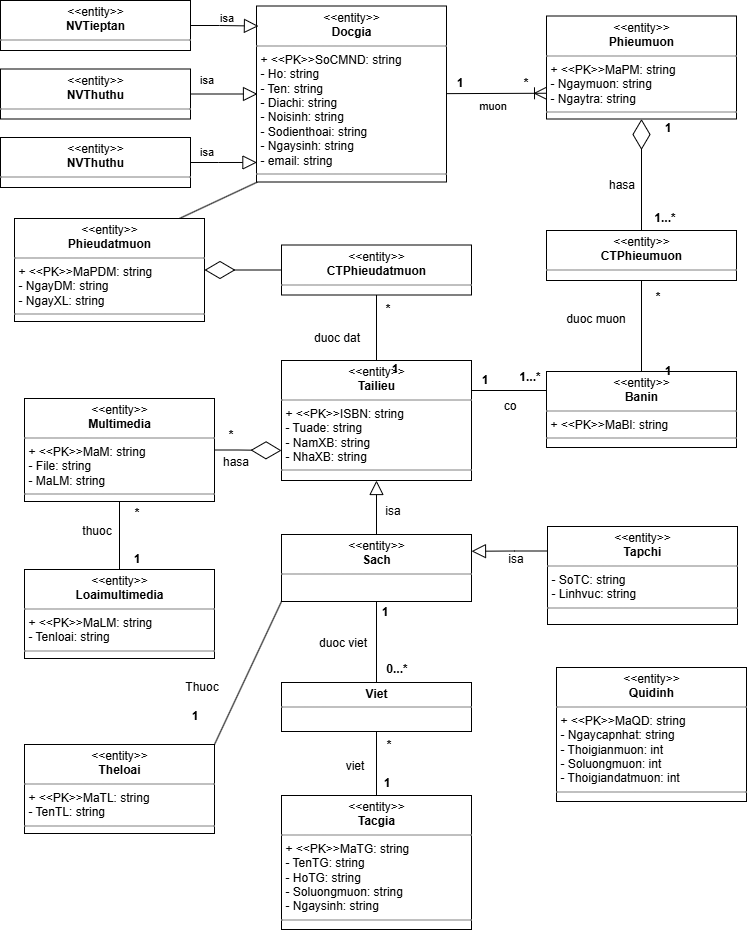
1. **Người dùng nạp tiền** → hệ thống nhận thông tin sản phẩm.
2. Hệ thống kiểm tra:

* Nếu **tiền nạp < giá sản phẩm** → thông báo "không đủ tiền" → yêu cầu nạp thêm hoặc cho người dùng lấy lại tiền.
* Nếu **tiền nạp ≥ giá sản phẩm** → tính số tiền dư.

1. Kiểm tra tồn kho sản phẩm:

* Nếu sản phẩm hết hàng → thông báo "SOLD OUT" → gợi ý sản phẩm khác → người dùng có thể chọn sản phẩm khác hoặc lấy lại tiền.
* Nếu sản phẩm còn hàng → cập nhật lại số lượng tồn kho → hoàn tất giao dịch.

1. **Kết thúc quy trình**: hoặc giao dịch thành công, hoặc khách hàng nhận lại tiền.
2. **Vẽ lược đồ lớp cho bài toán Quản lý thư viện**



### **1. Các thực thể chính (Entity)**

* **Docgia** (Độc giả):
  + Thuộc tính: Số CMND (PK), Họ, Tên, Địa chỉ, Nơi sinh, Số điện thoại, Ngày sinh, Email.
  + Quan hệ: Độc giả có thể **mượn** nhiều phiếu mượn.
* **Phieumuon** (Phiếu mượn):
  + Thuộc tính: Mã PM (PK), Ngày mượn, Ngày trả.
  + Quan hệ: 1 Phiếu mượn có **nhiều chi tiết phiếu mượn (CTPhieumuon)**.
* **CTPhieumuon** (Chi tiết phiếu mượn):
  + Chứa thông tin từng tài liệu được mượn trong 1 phiếu mượn.
  + Quan hệ: liên kết giữa **Phieumuon** và **Banin** (bản in).
* **Banin** (Bản in):
  + Thuộc tính: Mã BI (PK).
  + Quan hệ: mỗi bản in thuộc về **1 Tài liệu**.
* **Tailieu** (Tài liệu):
  + Thuộc tính: ISBN (PK), Tựa đề, Năm XB, Nhà XB.
  + Có 2 dạng con (ISA):
    - **Sach** (Sách)
    - **Tapchi** (Tạp chí)
* **Tacgia** (Tác giả):
  + Thuộc tính: Mã TG (PK), Tên TG, Họ TG, Số lượng mượn, Ngày sinh.
  + Quan hệ: Sách có thể được **viết bởi nhiều tác giả**.
* **Multimedia** (Tài liệu đa phương tiện):
  + Thuộc tính: Mã MM (PK), File, Mã loại.
  + Quan hệ: Thuộc về **Loaimultimedia** (Loại multimedia).
* **Loaimultimedia**:
  + Thuộc tính: Mã LM (PK), Tên loại.
* **Theloai**:
  + Thuộc tính: Mã TL (PK), Tên TL.
  + Quan hệ: **Sách thuộc 1 thể loại**.
* **Phieudatmuon** (Phiếu đặt mượn):
  + Thuộc tính: Mã PDM (PK), Ngày đặt mượn, Ngày xử lý.
  + Quan hệ: liên kết với **CTPhieudatmuon**.
* **CTPhieudatmuon**:
  + Chi tiết phiếu đặt mượn (các tài liệu được đặt).
* **Quidinh** (Quy định):
  + Thuộc tính: Mã QĐ (PK), Ngày cập nhật, Thời gian mượn, Số lượng mượn, Thời gian đặt mượn.
* **Nhanvien**:
  + Có các vai trò (ISA): **NVtieptan, NVthuthu**.

### **2. Các quan hệ (Relationship)**

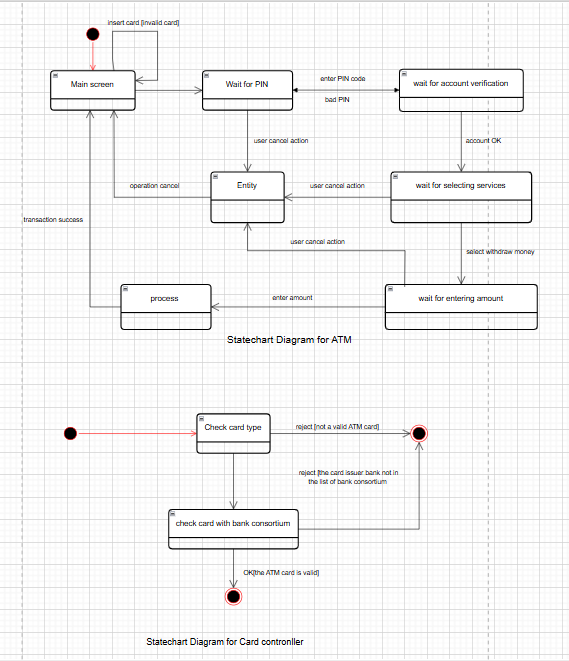
* **Docgia – Phieumuon**: 1 độc giả có thể mượn nhiều phiếu.
* **Phieumuon – CTPhieumuon – Banin – Tailieu**: mô tả chi tiết việc mượn sách/tạp chí/bản in.
* **Sach – Tacgia**: quan hệ n-n, một sách có thể nhiều tác giả, một tác giả viết nhiều sách.
* **Multimedia – Loaimultimedia**: mỗi multimedia thuộc một loại.
* **Sach – Theloai**: mỗi sách thuộc một thể loại.
* **Phieudatmuon – CTPhieudatmuon – Tailieu**: mô tả việc đặt tài liệu trước khi mượn.

### **3. Ý nghĩa tổng thể**

Sơ đồ này mô tả **toàn bộ quy trình quản lý thư viện**:

* Độc giả đến mượn/đặt mượn tài liệu (sách, tạp chí, multimedia).
* Hệ thống quản lý phiếu mượn, phiếu đặt mượn.
* Có quản lý nhân viên thư viện.
* Có quy định về số lượng mượn, thời gian mượn, thời gian đặt.

1. **Vẽ lược đồ sau:**



**1. Statechart Diagram for ATM**

Quy trình chính khi người dùng thao tác với máy ATM:

1. **Main screen**: Người dùng đưa thẻ vào. Nếu thẻ không hợp lệ → kết thúc.
2. **Wait for PIN**: Nhập mã PIN.
   * Nếu PIN sai → quay lại màn hình chính.
   * Nếu hủy thao tác → vào trạng thái **Entity** (kết thúc).
3. **Wait for account verification**: Hệ thống xác minh tài khoản.
   * Nếu OK → chuyển bước tiếp theo.
4. **Wait for selecting services**: Người dùng chọn dịch vụ (ví dụ rút tiền).
5. **Wait for entering amount**: Nhập số tiền cần rút.
6. **Process**: ATM xử lý giao dịch.
7. Trả kết quả: nếu thành công → về **Main screen**.

**2. Statechart Diagram for Card Controller**

Quy trình kiểm tra thẻ ATM:

1. **Check card type**: Kiểm tra xem có phải thẻ ATM hợp lệ không.
   * Nếu không hợp lệ → reject.
2. **Check card with bank consortium**: Kiểm tra xem ngân hàng phát hành có nằm trong liên minh ngân hàng không.
   * Nếu không → reject.
   * Nếu hợp lệ → OK, thẻ dùng được.

**3. Ý nghĩa tổng thể**

* Sơ đồ đầu mô tả **quy trình giao dịch ATM** (từ khi đưa thẻ đến khi hoàn tất giao dịch).
* Sơ đồ sau mô tả **cách hệ thống kiểm tra thẻ** (loại thẻ, ngân hàng phát hành).

1. **Phân tích dữ liệu cho ABC Bank**
2. **Thiết kế Cơ sở dữ liệu ở mức Khái niệm, Luận lý và Vật lý**

A black and white diagram of circles and squares

AI-generated content may be incorrect.

**Các mối quan hệ chính**

* **Branch – Employee**: Chi nhánh có nhiều nhân viên, nhân viên làm việc cho một chi nhánh. Một nhân viên có thể *quản lý* chi nhánh.
* **Employee – Customer**: Nhân viên *phục vụ* khách hàng.
* **Customer – Account**: Khách hàng có thể có nhiều tài khoản (savings/checking).
* **Customer – Loan**: Khách hàng có thể vay nhiều khoản vay.

**Ý nghĩa tổng thể**

Sơ đồ này mô tả hệ thống quản lý ngân hàng:

* Quản lý chi nhánh, nhân viên, khách hàng.
* Mối quan hệ giữa nhân viên – khách hàng.
* Các loại tài khoản (tiết kiệm, thanh toán) và khoản vay mà khách hàng có

1. 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.

-- table BRANCH --

INSERT INTO BRANCH VALUES

('BranchA', '12', 'Main St', 'District1', 'Hanoi', 'North', '[brancha@email.com](mailto:brancha@email.com)', 'M001'),

('BranchB', '45', 'Tran Hung Dao', 'District3', 'Ho Chi Minh', 'South', '[branchb@email.com](mailto:branchb@email.com)', 'M002'),

('BranchC', '78', 'Le Loi', 'District5', 'Da Nang', 'Central', '[branchc@email.com](mailto:branchc@email.com)', 'M003'),

('BranchD', '23', 'Nguyen Hue', 'District7', 'Can Tho', 'South', '[branchd@email.com](mailto:branchd@email.com)', 'M004');

-- table EMPLOYEE --

INSERT INTO EMPLOYEE VALUES

('N1080', 'John', 'Smith', '45', '2nd St', 'District2', 'Hanoi', '1985-06-20', '[john@email.com](mailto:john@email.com)', 'BranchA'),

('N1081', 'Anna', 'Nguyen', '101', 'Pham Van Dong', 'District1', 'Hanoi', '1990-04-15', '[anna@email.com](mailto:anna@email.com)', 'BranchA'),

('N1082', 'David', 'Tran', '202', 'Nguyen Van Cu', 'District4', 'Ho Chi Minh', '1988-09-10', '[david@email.com](mailto:david@email.com)', 'BranchB'),

('N1083', 'Sophia', 'Le', '56', 'Hai Ba Trung', 'District3', 'Da Nang', '1992-12-05', '[sophia@email.com](mailto:sophia@email.com)', 'BranchC');

-- table CUSTOMER --

INSERT INTO CUSTOMER VALUES

('C001', 'Peter', 'Johnson', 'Office1', 'Home1', '[peter@email.com](mailto:peter@email.com)', 'N1080'),

('C002', 'Maria', 'Pham', 'Office2', 'Home2', '[maria@email.com](mailto:maria@email.com)', 'N1081'),

('C003', 'James', 'Tran', 'Office3', 'Home3', '[james@email.com](mailto:james@email.com)', 'N1082'),

('C004', 'Linda', 'Hoang', 'Office4', 'Home4', '[linda@email.com](mailto:linda@email.com)', 'N1083');

-- table ACCOUNT --

INSERT INTO ACCOUNT VALUES

('A1001', 'C001'),

('A1002', 'C002'),

('A1003', 'C003'),

('A1004', 'C004');

-- table SAVINGACCOUNT --

INSERT INTO SAVINGACCOUNT VALUES

('A1001', 5000, 0.05, '2020-03-12'),

('A1002', 8000, 0.045, '2021-07-01'),

('A1003', 12000, 0.04, '2022-01-15'),

('A1004', 3000, 0.055, '2023-05-20');

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 EMPLOYEE

WHERE Code = 'N1080';

*Giải thích:* Các khách hàng được phục vụ bởi nhân viên này (trong bảng CUSTOMER) sẽ bị ảnh hưởng, có thể bị mất thông tin quản lý.

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 CUSTOMER C

JOIN ACCOUNT A ON C.Code = A.CusCode

JOIN SAVINGACCOUNT S ON A.Number = S.AccNumber

WHERE C.FName = 'Peter' AND C.LName = 'Johnson';

1. **BÀI TẬP ỨNG DỤNG**
2. **Hãy liệt kê các yêu cầu chức năng và phi chức năng của hệ thống CoolStore dựa trên mô tả kịch bản nghiệp vụ.**
   1. **Yêu cầu chức năng của hệ thống.**

|  |  |  |
| --- | --- | --- |
|  | Buyer | SysAdmin |
| **Product Catalog** | * Duyện danh muc sản phẩm * Lọc sản phẩm * Sắp xếp sản phẩm * Xem chi tiết sản phẩm | * Thêm/sửa/xoá sản phẩm * Gán sản phẩm vào kho |
| **Shopping Cart** | * Thêm sản phẩm vào giỏ hàng * Xoá sản phẩm khỏi giỏ hàng * Cập nhật giỏ hàng | * Xem giỏ hàng người dùng * Kích hoạt hoặc vô hiệu hoá giỏ hàng |
| **Payment Precess** | * Thanh toán giỏ hàng |  |
| **Inventory** |  | * Thêm/sửa/xoá hàng trong kho * Xem và cập nhật số lượng trong kho |
| **Rating** | * Tạo đánh giá sản phẩm |  |
| **Access Controll** | * Đăng nhập vào hệ thống * Đăng xuất khỏi hệ thộng | * Đăng nhập vào hệ thống * Đăng xuất khỏi hệ thộng |

* 1. **Yêu cầu phi chức năng**

|  |  |
| --- | --- |
| **Performance** | * Hệ thống có thể xử lý nhiều yêu cầu đồng thời * Thời gian phản hồi nhanh dưới 2 giây cho các thao tác duyệt sản phẩm/giỏ hàng |
| **Scalability** | * Có thể mở rộng thêm sản phẩm, kho, người dùng mà không làm giảm hiệu năng. * Có khả năng mở rộng để hỗ trợ số lượng lớn người dùng đồng thời. |
| **Portability** | * Tương thích với đa nền tảng, nhiều hệ thống. |
| **Usability** | * Giao diện thân thiện, dễ dùng * Hỗ trợ đa ngôn ngữ |
| **Security** | * Xác thực Buyer và SysAdmin * Phân quyền Buyer và SysAdmin * Mã hoá mật khẩu |
| **Mantainability** | * Dễ nâng cấp vào sửa lỗi * Module hoá (catalog, shopping cart, rating, inventory, access control) |

* 1. **Phần đặt vấn đề**
* **Thu thập data story**

|  |  |
| --- | --- |
| **Danh sách câu hỏi (Product Catalog)** | **Trả Lời Của Khách Hàng** |
| Bạn muốn người mua có thể tìm kiếm sản phẩm theo những tiêu chí nào? | **Với tư cách là Người mua hàng**, tôi muốn xem danh sách sản phẩm trên trang chủ có chức năng **lọc và sắp xếp** (theo tên, hình ảnh, mô tả ngắn, đánh giá, và cờ “sản phẩm hot” – tức là sản phẩm có nhiều lượt xem hoặc mua).  Khi thực hiện **lọc** theo bất kỳ mức giá và tên sản phẩm nào, danh sách sản phẩm cần được thu hẹp lại chỉ còn những sản phẩm phù hợp.  Khi thực hiện **sắp xếp** tăng dần hoặc giảm dần theo giá hoặc tên sản phẩm, danh sách sản phẩm cần được hiển thị theo thứ tự sắp xếp này.  Khi vừa **lọc vừa sắp xếp**, danh sách sản phẩm cần tuân theo cả hai điều kiện trên.  **Với tư cách là Người mua hàng**, tôi muốn có thể xem chi tiết một sản phẩm với các thuộc tính cơ bản như: tên, mô tả, số lượng sản phẩm còn trong kho, thông tin kho hàng (địa chỉ và vị trí lưu kho), cờ “sản phẩm hot” (nếu có), và đánh giá.  **Với tư cách là Quản trị viên hệ thống (SysAdmin)**, tôi muốn quản lý sản phẩm (thực hiện các hành động **CRUD: tạo, đọc, cập nhật, xóa**) và gán một kho hàng hiện có cho sản phẩm. |
| Khi lọc/sắp xếp sản phẩm, bạn muốn có những tùy chọn nào ngoài tên và giá? |
| Trên trang chi tiết sản phẩm, ngoài thông tin cơ bản (tên, mô tả, giá, tồn kho, địa chỉ kho), bạn có muốn hiển thị thêm hình ảnh hoặc video minh họa không? |
| Với trạng thái “sản phẩm nổi bật”, ai sẽ gán trạng thái này? Hệ thống tự động hay quản trị viên thủ công? |
| **Danh sách câu hỏi (Shopping Cart)** | **Trả Lời Của Khách Hàng** |
| Khi người mua thêm sản phẩm vào giỏ, bạn có muốn giới hạn số lượng sản phẩm tối đa trong giỏ không? | **Với tư cách là Người mua hàng**, tôi muốn mua bất kỳ sản phẩm nào trên trang danh mục sản phẩm (thêm sản phẩm này vào giỏ hàng – mặc định sẽ thêm 1 sản phẩm).  **Với tư cách là Người mua hàng**, tôi muốn xem chi tiết sản phẩm và mua sản phẩm đó nếu tôi thích (thêm sản phẩm này vào giỏ hàng – mặc định sẽ thêm 1 sản phẩm).  **Với tư cách là Người mua hàng**, tôi muốn xem danh sách các sản phẩm vừa thêm vào giỏ hàng, và tôi muốn xem bảng thông tin tóm tắt của giỏ hàng hiện tại bao gồm: tổng chi phí giỏ hàng, số tiền tiết kiệm từ khuyến mãi sản phẩm, chi phí tạm tính, phí vận chuyển, số tiền tiết kiệm từ khuyến mãi vận chuyển, và tổng số tiền đơn hàng trên trang này.  **Với tư cách là Người mua hàng**, tôi muốn **cập nhật số lượng sản phẩm** trong giỏ hàng.  Khi cập nhật số lượng sản phẩm, bảng thông tin tóm tắt cần được cập nhật theo thay đổi.  **Với tư cách là Người mua hàng**, tôi muốn **xóa bất kỳ sản phẩm nào** trong giỏ hàng mà tôi không muốn mua nữa.  Khi xóa sản phẩm, bảng thông tin tóm tắt cần được cập nhật theo thay đổi.  **Với tư cách là Người mua hàng**, tôi muốn **thanh toán (checkout) giỏ hàng của mình**.  Khi số lượng sản phẩm trong giỏ hàng bằng 0 thì quá trình thanh toán sẽ không diễn ra.  Khi giỏ hàng được thanh toán, quá trình **thanh toán (payment process)** sẽ bắt đầu.  **Với tư cách là Quản trị viên hệ thống (SysAdmin)**, tôi muốn xem giỏ hàng của tất cả người mua với các thông tin: tổng chi phí giỏ hàng, số tiền tiết kiệm từ khuyến mãi sản phẩm, chi phí tạm tính, phí vận chuyển, số tiền tiết kiệm từ khuyến mãi vận chuyển, và tổng số tiền đơn hàng.  **Với tư cách là Quản trị viên hệ thống (SysAdmin)**, tôi muốn **bật/tắt (enable/disable)** bất kỳ giỏ hàng nào của người mua. |
| Bạn có muốn giỏ hàng lưu lại khi người dùng đăng xuất và đăng nhập lại không? |
| Thông tin khuyến mãi: bạn muốn áp dụng theo mã giảm giá, chương trình tự động, hay cả hai? |
| Ai sẽ có quyền vô hiệu hóa một giỏ hàng “không hợp lệ”? |
| **Danh sách câu hỏi (Payment Process)** |  |
| Bạn muốn người mua thanh toán bằng những phương thức nào? (Ví dụ: thẻ tín dụng, ví điện tử, COD, chuyển khoản?) | **Bất kỳ Người mua hàng nào** cũng có thể thực hiện **thanh toán**.  Tại thời điểm **quá trình thanh toán** diễn ra, hệ thống sẽ bắt đầu:   1. **Xác thực thông tin sản phẩm**, 2. **Xử lý thanh toán**, 3. **Gửi email cho Người mua hàng** (do đây chỉ là bản demo nên không thực sự tích hợp với cổng thanh toán).   Nếu **bất kỳ thông tin sản phẩm nào không hợp lệ**, thì quá trình thanh toán sẽ **bị hủy bỏ** và một email sẽ được gửi cho Người mua hàng để thông báo.  Khi **kết thúc quá trình thanh toán**, hệ thống sẽ đánh dấu giỏ hàng này ở trạng thái **đã xử lý thanh toán** và gửi email để Người mua hàng biết. |
| Sau khi thanh toán thành công, bạn muốn hệ thống gửi email xác nhận, hay cả SMS thông báo? |
| Trường hợp thanh toán thất bại, hệ thống cần xử lý thế nào? (Ví dụ: hiển thị thông báo lỗi, cho phép thử lại, giữ sản phẩm trong giỏ hàng?) |
| **Danh sách câu hỏi (Inventory)** |  |
| Quản trị viên có thể quản lý kho ở mức nào? (Chỉ số lượng tồn hay cả thông tin chi tiết như địa chỉ, trạng thái hoạt động?) | **Với tư cách là Quản trị viên hệ thống (SysAdmin)**, tôi muốn **quản lý kho hàng** (thực hiện các thao tác **CRUD: tạo, đọc, cập nhật, xóa**). |
| Bạn có cần cảnh báo khi số lượng sản phẩm trong kho xuống dưới một ngưỡng tối thiểu không? |
| Có cần báo cáo tổng hợp về tồn kho theo thời gian thực không? |
| **Danh sách câu hỏi (Rating)** |  |
| Ai được phép đánh giá sản phẩm? (Bất kỳ người dùng nào hay chỉ những người đã mua sản phẩm?) | **Với tư cách là Người mua hàng**, tôi muốn **đánh giá từng sản phẩm mà tôi thấy tốt** bằng thang điểm từ **1 đến 5 sao**. |
| Bạn có muốn người mua để lại bình luận kèm theo số sao đánh giá không? |
| Có cần cơ chế kiểm duyệt đánh giá trước khi hiển thị công khai không? |
| **Danh sách câu hỏi (Access Control)** |  |
| Bạn muốn hỗ trợ những loại tài khoản nào ngoài Buyer và SysAdmin? | Mỗi Người mua (Buyer) / Quản trị viên hệ thống (SysAdmin) đều là một **Người dùng** (User).  **Với tư cách là Người mua/Quản trị viên hệ thống**, tôi muốn **đăng nhập vào hệ thống**.Khi một người dùng có vai trò **Buyer** đăng nhập, tôi sẽ được đưa đến **trang danh mục sản phẩm**.Khi một người dùng có vai trò **SysAdmin** đăng nhập, tôi sẽ được đưa đến **trang quản trị**.  **Với tư cách là Người mua/Quản trị viên hệ thống**, tôi muốn **đăng xuất khỏi hệ thống**. |
| Khi người dùng quên mật khẩu, bạn muốn quy trình reset mật khẩu qua email hay SMS? |
| Bạn có yêu cầu đăng nhập hai lớp (2FA) cho quản trị viên không? |

1. **Vẽ qui trình nghiệp vụ cho website CoolStore**

A black and white diagram

Description automatically generated

1. **Xây dựng use case diagram cho website CoolStore.**

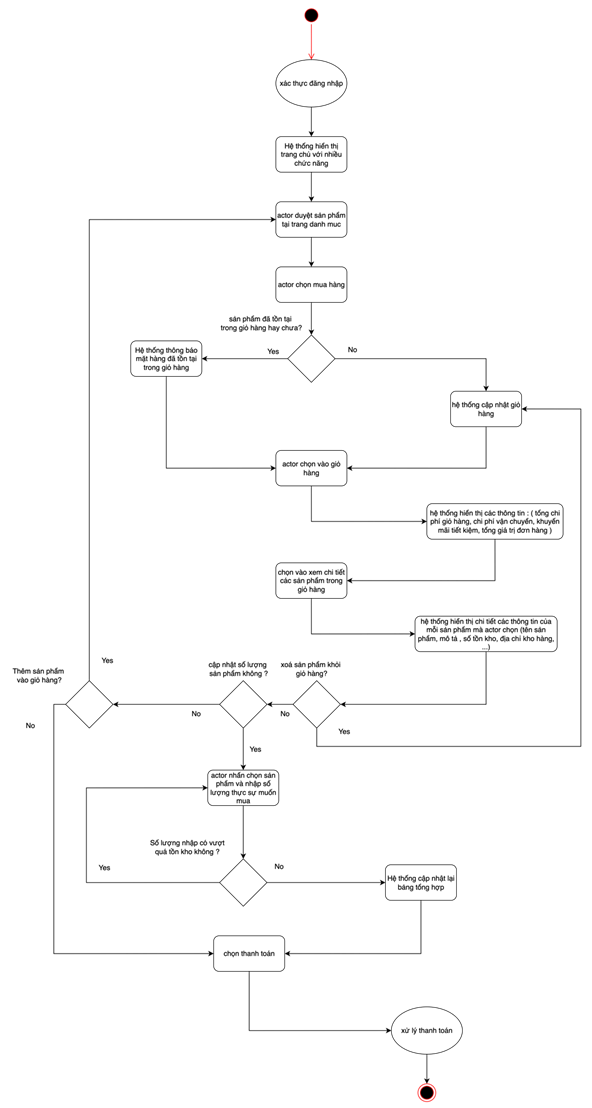
A diagram of a diagram

Description automatically generated with medium confidence

1. 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 : | Thêm Sản phẩm vào giỏ hàng từ trang chi tiết | |
| Actor(s) : | Khách Hàng | |
| Maturity : | Focused | |
| Summary : | Khách hàng sau khi duyệt sản phẩm nhấn vào trang chi tiết sản phẩm để xem chi tiết và có nhu cầu mua sản phẩm từ trang chi tiết này. | |
| Basic Flow Of Event : | Actor Action | System Response |
|  | 1. Xác thực đăng nhập |  |
|  |  | 2. Hệ thống hiển thị trang chủ |
|  | 3. Actor duyệt sản phẩm tại trang danh mục sản phẩm |  |
|  | 4. Actor chọn vào xem chi tiết sản phẩm |  |
|  |  | 5. Hệ thống hiển thị trang chi tiết sản phẩm gồm các thông tin:  + tên sản phẩm  + mô tả  + số lượng tồn kho  + thông tin địa chỉ kho  + trạng thái sản phẩm nổi bật (nếu có)  + đánh giá |
|  | 6. Actor nhập số lượng  (E1) |  |
|  | 7. Actor nhấn chọn mua hàng (A1) |  |
|  |  | 8. Hệ thống thêm sản phẩm này vào giỏ hàng với các thông tin cơ bản như :  + tổng chi phí giỏ hàng  + chi phí vận hành  + khuyến mãi tiết kiệm  + tổng giá trị đơn hàng  (Usecase kết thúc tại đây) (A2, A3, A4) |
| Alternative Flow : | Actor Action | System Response |
|  |  | A1.1. Hệ thống hiển thị thông báo sản phẩm đã có sản trong giỏ hàng. |
|  | A1.2. Actor nhấn chọn vào Giỏ Hàng. ( Quay lại Bước 7 Basic Flow ) |  |
|  | A2. Actor có khả năng xoá sản phẩm khỏi giỏ hàng. (Quay lại bước 3 Basic flow)  A3. Actor có khả năng cập nhật số lượng sản phẩm. (Quay lại bước 3 Basic flow)  A4. Actor có thể thực hiện thêm sản phẩm mới vào giỏ hàng. (Quay lại bước 3 Basic flow) | |
| Exception Path : | E1. Actor nhập số lượng vượt quá hàng còn tồn trong kho. | |
| Extention Point : | none | |
| Triggers : | Actor muốn thêm sản phẩm vào giỏ hàng và mua | |
| Assumptions : | none | |
| Preconditions : | none | |

1. **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.**



1. **Thiết kế sơ đồ Class Diagram cho module Giỏ hàng.**

A diagram of a product

AI-generated content may be incorrect.

**Các lớp chính**

1. **User**
   * **Thuộc tính:**
     + user\_id: int
     + name: string
     + role: string (Buyer/Admin)
   * **Phương thức:**
     + login()
     + logout()
2. **Cart**
   * **Thuộc tính:**
     + cart\_id: int
     + user\_id: int (FK → User)
     + status: string (active, inactive, checked-out)
     + total\_amount: double
     + discount\_amount: double
     + shipping\_cost: double
     + final\_amount: double
   * **Phương thức:**
     + addItem(product, quantity)
     + removeItem(product)
     + updateQuantity(product, quantity)
     + calculateTotal()
     + checkout()
3. **CartItem** (giỏ hàng có nhiều sản phẩm)
   * **Thuộc tính:**
     + cart\_item\_id: int
     + cart\_id: int (FK → Cart)
     + product\_id: int (FK → Product)
     + quantity: int
     + subtotal: double
   * **Phương thức:**
     + calculateSubtotal()
4. **Product**
   * **Thuộc tính:**
     + product\_id: int
     + name: string
     + description: string
     + price: double
     + is\_featured: boolean
   * **Phương thức:**
     + getInfo()

**Quan hệ giữa các lớp**

* **User (1) —— (1..\*) Cart**: Một người dùng có thể có nhiều giỏ hàng (nhưng thường chỉ có 1 giỏ đang active).
* **Cart (1) —— (1..\*) CartItem**: Mỗi giỏ hàng chứa nhiều sản phẩm.
* **Product (1) —— (1..\*) CartItem**: Một sản phẩm có thể nằm trong nhiều giỏ hàng khác nhau.

1. **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.
   * Chức năng thao tác nhanh: Tăng/giảm số lượng sản phẩm; Xóa sản phẩm khỏi giỏ; Lưu sản phẩm để mua sau.
   * Bảng tổng hợp đơn hang: Tổng chi phí giỏ hang; Khuyến mãi tiết kiệm được; Phí vận chuyển; Tổng thanh toán cuối cùng.
   * Hành động chính: Nút “Thanh toán”; Gợi ý sản phẩm liên quan để thêm vào giỏ.

A screenshot of a menu

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

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

Presentation Layer

* Buyer UI
  + Trang đăng nhập/đăng kí: người dùng tạo tài khoản và đăng nhập, sau khi đăng nhập xong sẽ chuyển về trang Danh mục sản phẩm
  + Trang Danh mục sản phẩm: có chức năng lọc và sắp xếp sản phẩm theo tên và giá; có nút “mua hàng” và “xem chi tiết”
  + Trang Chi tiết sản phẩm: hiện ra khi bấm vào sản phẩm trên trang Danh mục sản phẩm; trang sẽ hiện các thông tin như: tên, mô tả, số lượng tồn kho, thông tin kho như địa chỉ kho và đánh giá; có tích hợp nút mua hàng ở trang.
  + Trang giỏ hàng: sản phẩm khách hàng vừa mua sẽ xuất hiện trong giỏ hàng. 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 sẽ xuất hiện. Khi người mua tăng hoặc xóa sản phẩm thì sẽ được tự động cập nhật.
  + Trang thanh toán: chứa thông tin các sản phẩm vừa mua, thông tin người mua như địa chỉ, số điện thoại và tên. Phương thức thanh toán và chi phí vận chuyển.
  + Trang lịch sử mua hàng: sẽ chứa thông tin các đơn hàng đã được thanh toán, trạng thái đơn hàng (được xử lý, được giao, giao thành công,…).
  + Trang Đánh giá: sau khi đơn hàng được giao hoàn tất, bên cạnh thông tin đơn hàng sẽ xuất hiện nút Đánh giá, khách hàng có thể đánh giá từ 1-5 sao cho sản phẩm.
* SysAdmin
  + Trang quản lý sản phẩm: gán sản phẩm vào kho hiện có.
  + Trang quản lý giỏ hàng: quản trị viên có thể thể kích hoạt hoặc vô hiệu hóa bất kỳ giỏ hàng không hợp lệ nào trên website CoolStore.
  + Trang xử lý thanh toán: xem được chi tiết các đơn hàng của khách hàng; có thể cập nhật tự động hoặc thủ công trạng thái đơn hàng hay hủy bỏ đơn hàng.
  + Trang Tồn kho: có thể cập nhật sản phẩm trong kho
  + Trang Đăng nhập/đăng kí: khi SysAdmin đăng nhập, sẽ chuyển về trang Quản lý sản phẩm.

Business Logic Layer

* Product Catalog Service
  + Buyer: duyệt sản phẩm, lọc, sắp xếp, xem chi tiết.
  + Admin: thêm/sửa/xóa sản phẩm, gán sản phẩm vào kho.
* Shopping Cart Service
  + Buyer: thêm/xóa sản phẩm vào giỏ
  + Admin: xem và quản lý giỏ hàng người dùng.
* Payment Service
  + Xác thực đơn hàng → xử lý thanh toán (tích hợp cổng thanh toán).
  + Gửi email xác nhận đơn hàng.
* Inventory Service
  + Admin: quản lý kho hàng (tạo/sửa/xóa kho, cập nhật tồn kho).
  + Buyer: kiểm tra số lượng tồn khi xem sản phẩm.
* Rating Service
  + Buyer: đánh giá sản phẩm (1–5 sao).
  + Tính toán điểm trung bình hiển thị trên sản phẩm.
* Access Control / Authentication Service
  + Buyer: đăng nhập → đưa về trang sản phẩm.
  + Admin: đăng nhập → đưa về trang quản trị.
  + Đăng xuất.

Data Layer (Database)

* **User** (user\_id, username, password, email, role [Buyer/Admin], status)
* **Product** (product\_id, name, description, price, is\_featured, inventory\_id)
* **Inventory** (inventory\_id, address, stock\_quantity)
* **Category\_Product** (category\_id, name, description, parent\_id)
* **Product\_Category\_Mapping** (id, product\_id, category\_id)
* **Cart** (cart\_id, user\_id, status, total\_amount, shipping\_cost, final\_amount)
* **Cart\_Detail** (cart\_detail\_id, cart\_id, product\_id, quantity, unit\_price, total\_price)
* **Order** (order\_id, cart\_id, payment\_status, created\_at, shipped\_at)
* **Rating** (rating\_id, user\_id, product\_id, stars, comment, created\_at)

1. **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 hợp lệ vào giỏ hàng trống

* **Dữ liệu đầu vào:**
  + Người mua đã đăng nhập.
  + Sản phẩm mã P101 còn trong kho 20 sản phẩm.
  + Người mua muốn thêm 2 sản phẩm.
* **Bước thực hiện:**
  + Người mua mở trang danh mục sản phẩm.
  + Nhấn nút **“Mua hàng”** cho sản phẩm P101, số lượng = 2.
  + Hệ thống kiểm tra tồn kho.
* **Kết quả mong đợi:**
  + Sản phẩm P101 được thêm vào giỏ hàng với số lượng 2.
  + Bảng tổng hợp giỏ hàng hiển thị đúng tổng tiền (= đơn giá × số lượng(2) ).
  + Tồn kho giảm từ 20 -> 18.

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

* **Dữ liệu đầu vào:**
  + Người mua đã đăng nhập
  + Sản phẩm mã P102 còn trong kho 5 cái.
  + Người mua muốn thêm 10 cái.
* **Bước thực hiện:**
  + Người mua mở trang danh mục sản phẩm.
  + Nhấn nút **“Mua hàng”** cho sản phẩm P102, số lượng = 10.
  + Hệ thống kiểm tra tồn kho.
* **Kết quả mong đợi:**
  + Hệ thống hiển thị thông báo lỗi: **“Số lượng sản phẩm không đủ trong kho.”**
  + Sản phẩm **không** được thêm vào giỏ hàng.
  + Tồn kho giữ nguyên = 5.

Test case 3: Thêm số lượng không tồn tại (số âm, số thực) vào giỏ hàng

* **Dữ liệu đầu vào:**
  + Người mua đã đăng nhập
  + Sản phẩm mã P103 còn trong kho 5 cái.
  + Người mua muốn thêm âm 10 (-10) cái.
* **Bước thực hiện:**
  + Người mua mở trang danh mục sản phẩm.
  + Nhấn nút **“Mua hàng”** cho sản phẩm P102, số lượng = -10.
  + Hệ thống kiểm tra tồn kho.
* **Kết quả mong đợi:**
  + Hệ thống hiển thị thông báo lỗi: **“Số lượng sản phẩm bạn nhập không tồn tại.”**
  + Sản phẩm **không** được thêm vào giỏ hàng.
  + Tồn kho giữ nguyên = 5.