**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC SÀI GÒN**

A blue circle with text

Description automatically generated

**BÁO CÁO BÀI TẬP**

**MÔN: KIỂM THỬ PHẦN MỀM**

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

**Thành viên thực hiện:**

Nguyễn Võ Minh Hiếu - 3122411057

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 5 NĂM 2025**

**MỤC LỤC**

[BÀI TẬP MINH HỌA 3](#_Toc209122216)

[Bài 1. Vẽ lại qui trình nghiệp vụ sau (Độc giả, NV Tiếp tân, Thành viên) 3](#_Toc209122217)

[Bài 2. Vẽ mô hình khái niệm 3](#_Toc209122218)

[Bài 3. Vẽ lại business use case 4](#_Toc209122219)

[Bài 4. Vẽ sơ đồ hoạt động và tương tác sau và tóm tắt ý nghĩa sơ đồ 6](#_Toc209122220)

[Bài 5. Vẽ lược đồ lớp cho bài toán Quản lý thư viện 8](#_Toc209122221)

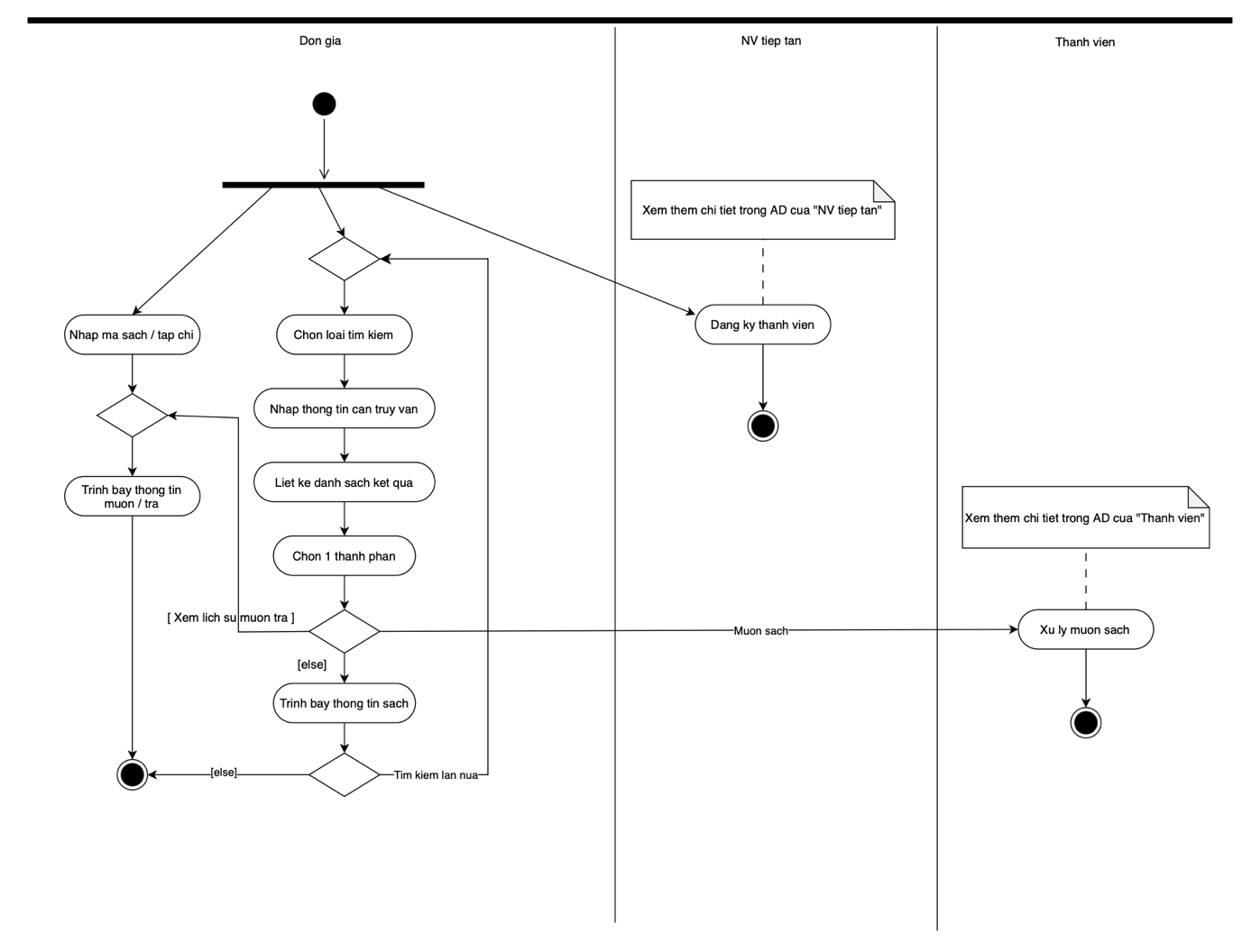
[Bài 6. Vẽ lược đồ : 9](#_Toc209122222)

[Bài 7. Phân tích dữ liệu cho ABC Bank 9](#_Toc209122223)

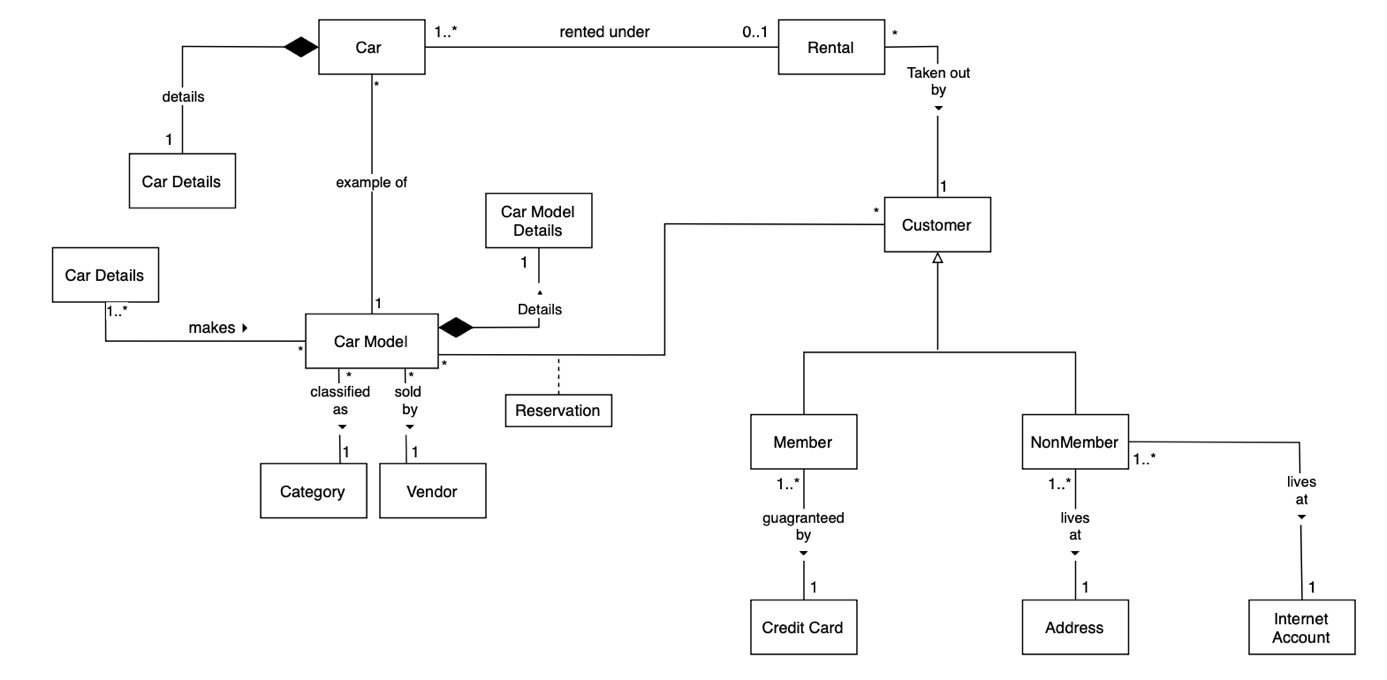
[Bài 8: Bài tập ứng dụng 15](#_Toc209122224)

# BÀI TẬP MINH HỌA

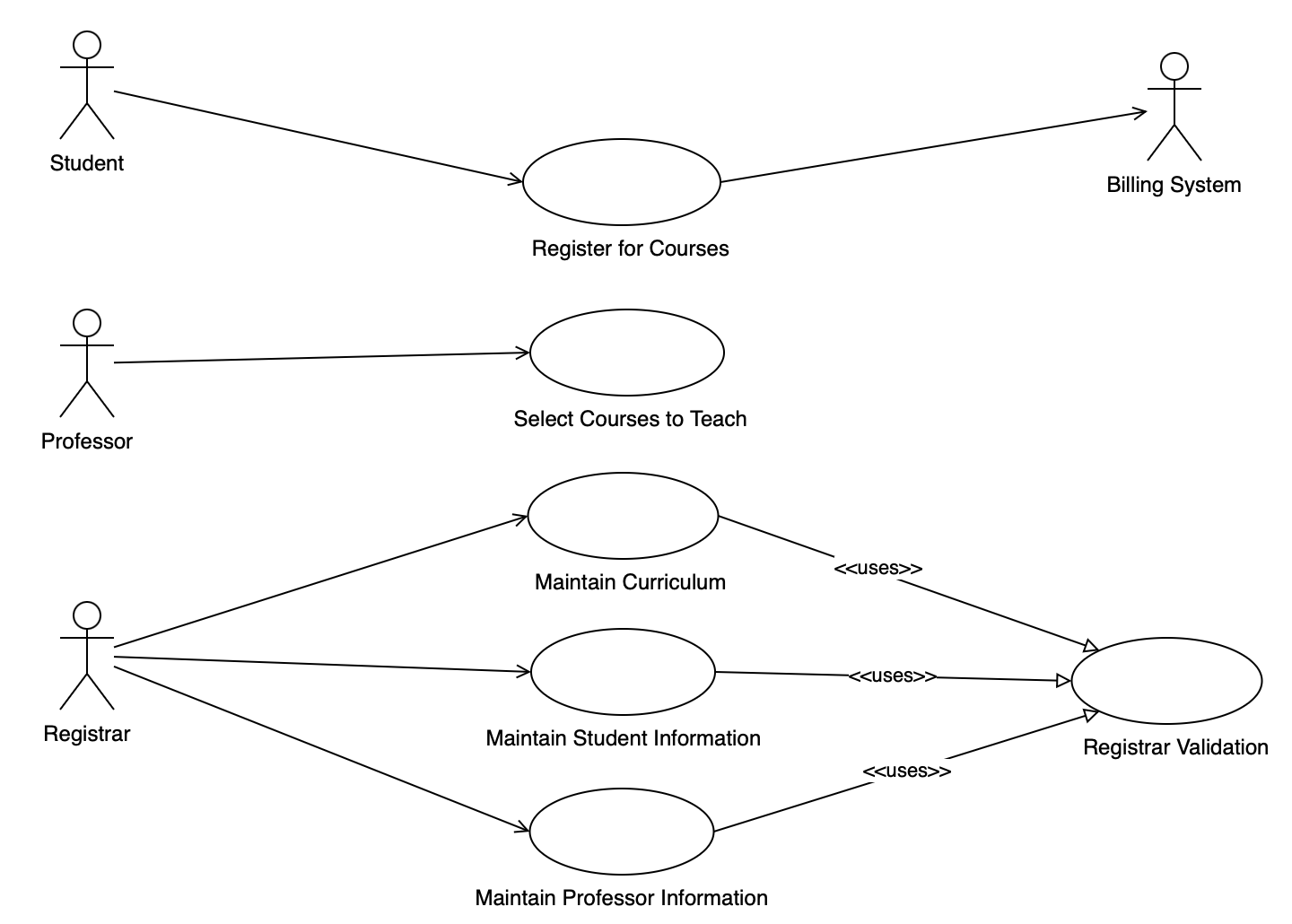
## Bài 1. Vẽ lại qui trình nghiệp vụ sau (Độc giả, NV Tiếp tân, Thành viên)



## Bài 2. Vẽ mô hình khái niệm



## Bài 3. Vẽ lại business use case



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

Trong sơ đồ có các tác nhân :

1. Student

* Đăng ký môn học
* Tương tác với hệ thống thanh toán học phí sau khi đăng ký

1. Professor

* Chọn môn học để giảng dạy

1. Registrar

* Quản lý chương trình học
* Quản lý thông tin sinh viên
* Quản lý thông tin giảng viên

1. Maintain Student Information

* Xử lý thông tin thanh toán 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.**

Các use cases trong sơ đồ:

* Register for Courses – Sinh viên đăng ký môn học.
* Select Courses to Teach – Giảng viên chọn môn để dạy.
* Maintain Curriculum – Quản lý chương trình học (cập nhật môn học, tín chỉ).
* Maintain Student Information – Quản lý thông tin sinh viên (hồ sơ, điểm, trạng thái).
* Maintain Professor Information – Quản lý thông tin giảng viên.
* Registrar Validation – Kiểm tra, xác thực dữ liệu trong hệ thống.

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

Mối quan hệ <<uses>> giữa Maintain Curriculum và Registrar Validation liên quan đến việc duy trì dữ liệu học tập. Nghĩa là khi có một khóa học mới được thêm vào, hệ thống sẽ xác nhận xem khóa học này có bị trùng lặp không.

Mối quan hệ <<uses>> giữa Maintain Student Information và Registrar Validation nhằm xác thực tính hợp lệ của dữ liệu và tránh sự trùng lặp. Nghĩa là khi Registrar thao tác trên Maintain Student Information, hệ thống sẽ xác nhận xem mã số sinh viên nhập vào có hợp lệ không và dữ liệu khi thao tác cập nhật có bị trùng lặp không.

Mối quan hệ <<uses>> giữa Maintain Professor Information và Registrar Validation có ý nghĩa xác thực và kiểm tra dữ liệu khi thực hiện thay đổi tương tự như quan hệ <<use>> giữa Maintain Student Information và Registrar Validation.

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

Ví dụ: Register for Courses

1. Sinh viên đăng nhập vào hệ thống.
2. Hệ thống hiển thị danh sách môn học mở trong kỳ.
3. Sinh viên chọn môn học muốn đăng ký.
4. Hệ thống kiểm tra điều kiện tiên quyết (số tín chỉ, môn học).
5. Hệ thống gửi yêu cầu đến Billing System để xử lý học phí.
6. Nếu thanh toán thành công, hệ thống xác nhận đăng ký.
7. Hệ thống lưu lại kết quả đăng ký và thông báo cho sinh viên.

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

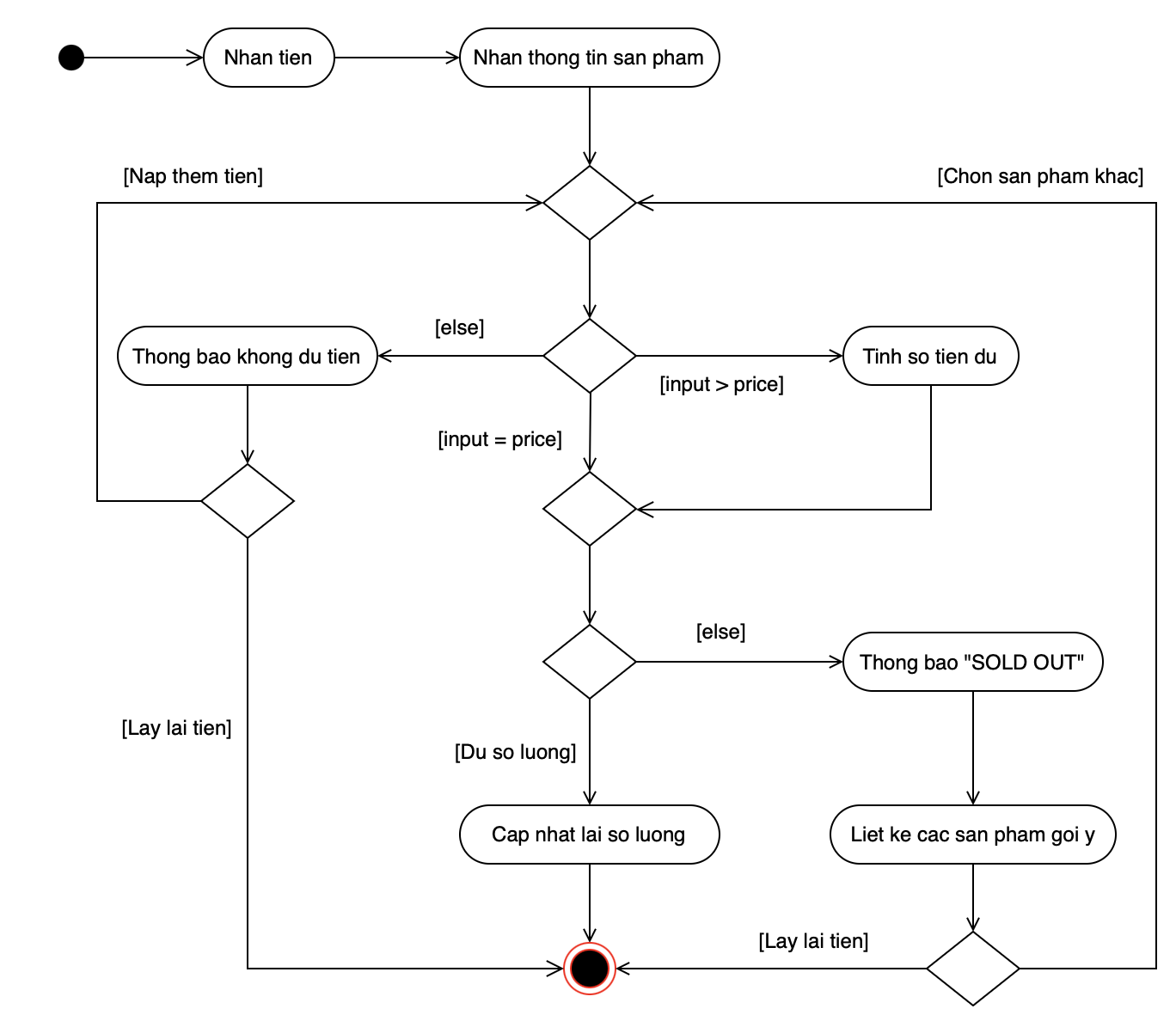
Drop / Withdraw Course – Sinh viên hủy môn đã đăng ký.

View Grades / Transcript – Sinh viên xem điểm số, bảng điểm.

Schedule Courses –  Hệ thống lập lịch học / lịch thi

Generate Reports – Registrar tạo báo cáo thống kê (sinh viên, giảng viên, môn học).

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

****

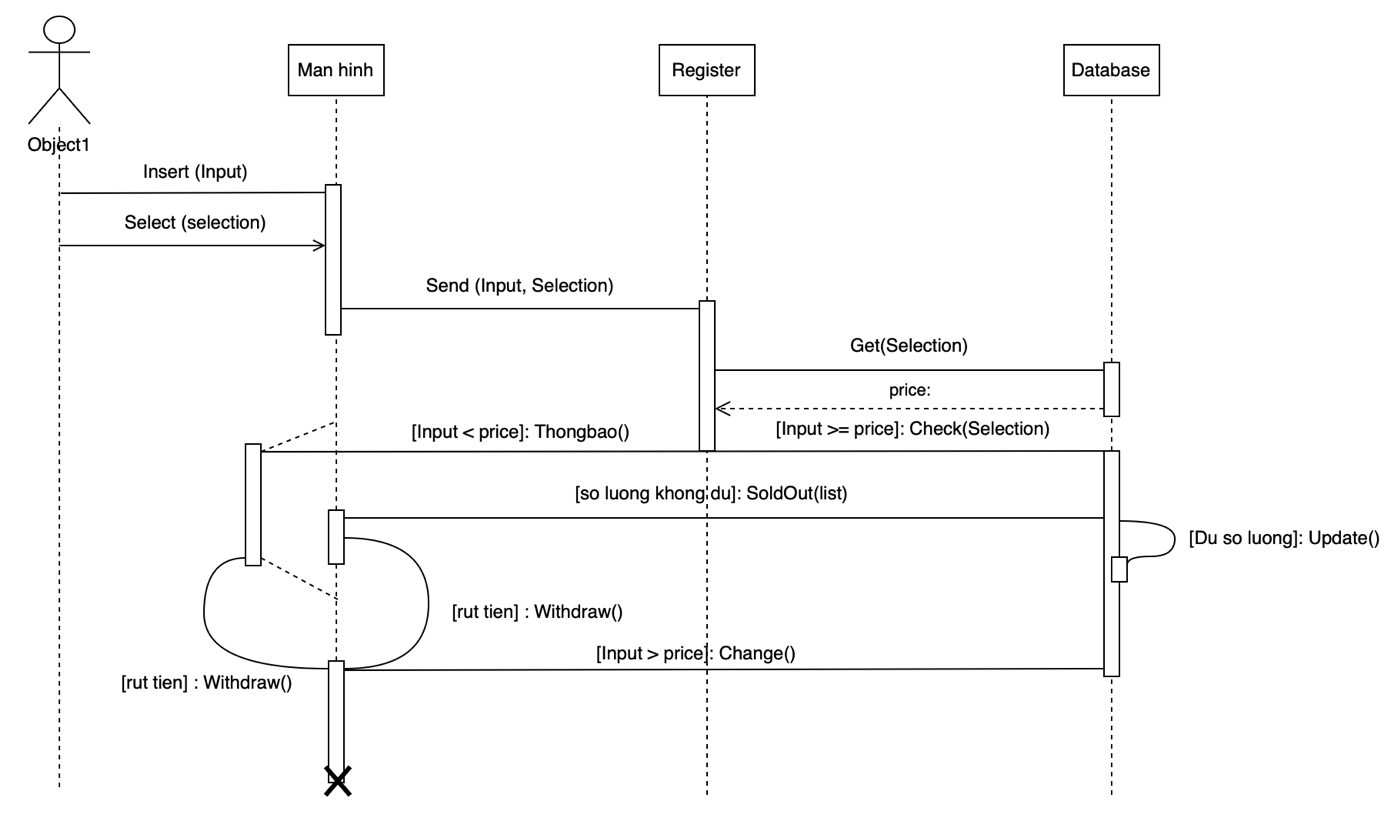
**Ngữ cảnh:** Một người dùng (khách hàng) nạp tiền và chọn sản phẩm từ máy bán hàng tự động.

**Diễn giải:**

* Người dùng nạp tiền.
* Nhập thông tin sản phẩm muốn mua.

**Hệ thống kiểm tra:**

* Nếu tiền < giá → Thông báo "không đủ tiền" → cho nạp thêm hoặc lấy lại tiền.
* Nếu tiền > giá → Hệ thống tính tiền dư để trả lại.
* Nếu tiền = giá → Kiểm tra số lượng tồn kho:
* Nếu còn hàng → Cập nhật số lượng → Kết thúc.
* Nếu hết hàng → Thông báo "SOLD OUT" → Đưa gợi ý sản phẩm khác hoặc cho lấy lại tiền.

****

**Ngữ cảnh:** Cùng tình huống trên nhưng chi tiết về thông điệp trao đổi giữa các đối tượng.

**Các đối tượng:**

* Người dùng (Object1)
* Màn hình (Man hinh)
* Bộ xử lý/đăng ký sản phẩm (Register)
* Cơ sở dữ liệu (Database)

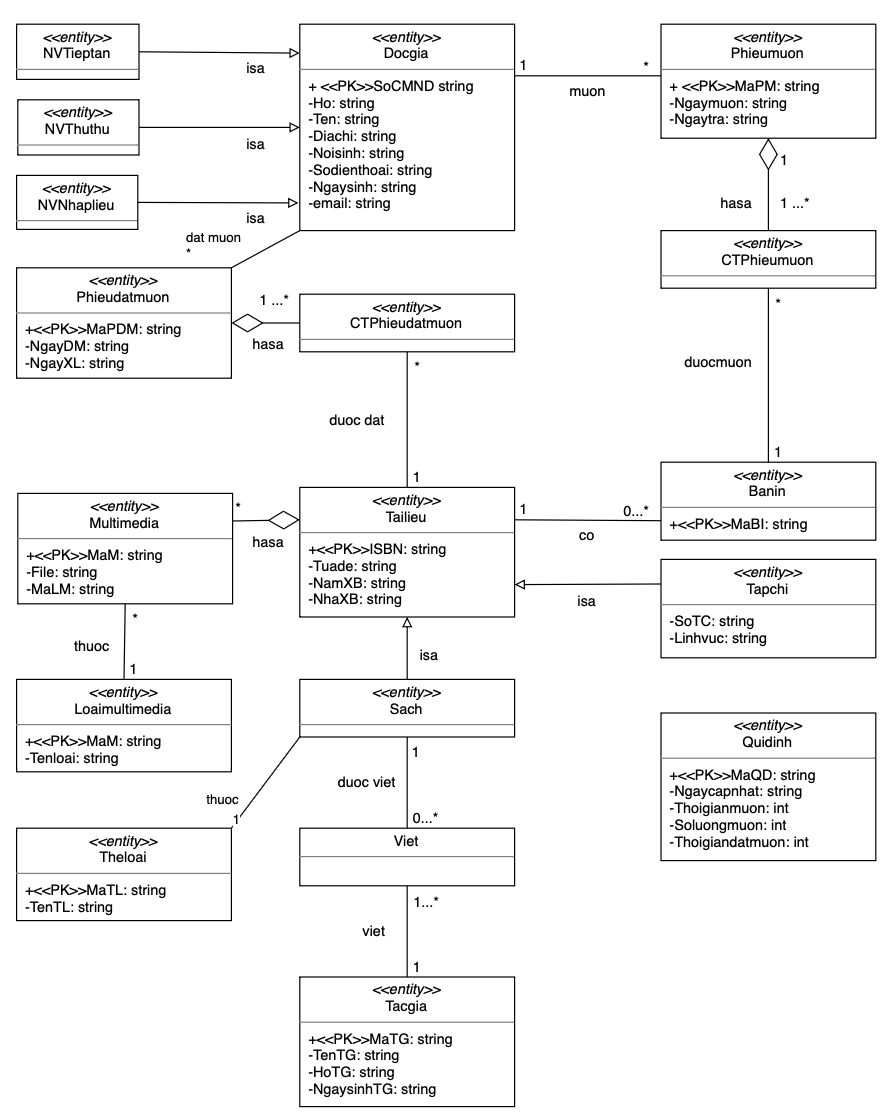
**Luồng chính:**

* Người dùng nhập tiền và chọn sản phẩm.
* Màn hình gửi yêu cầu đến Register.
* Register gọi Database để kiểm tra thông tin sản phẩm.

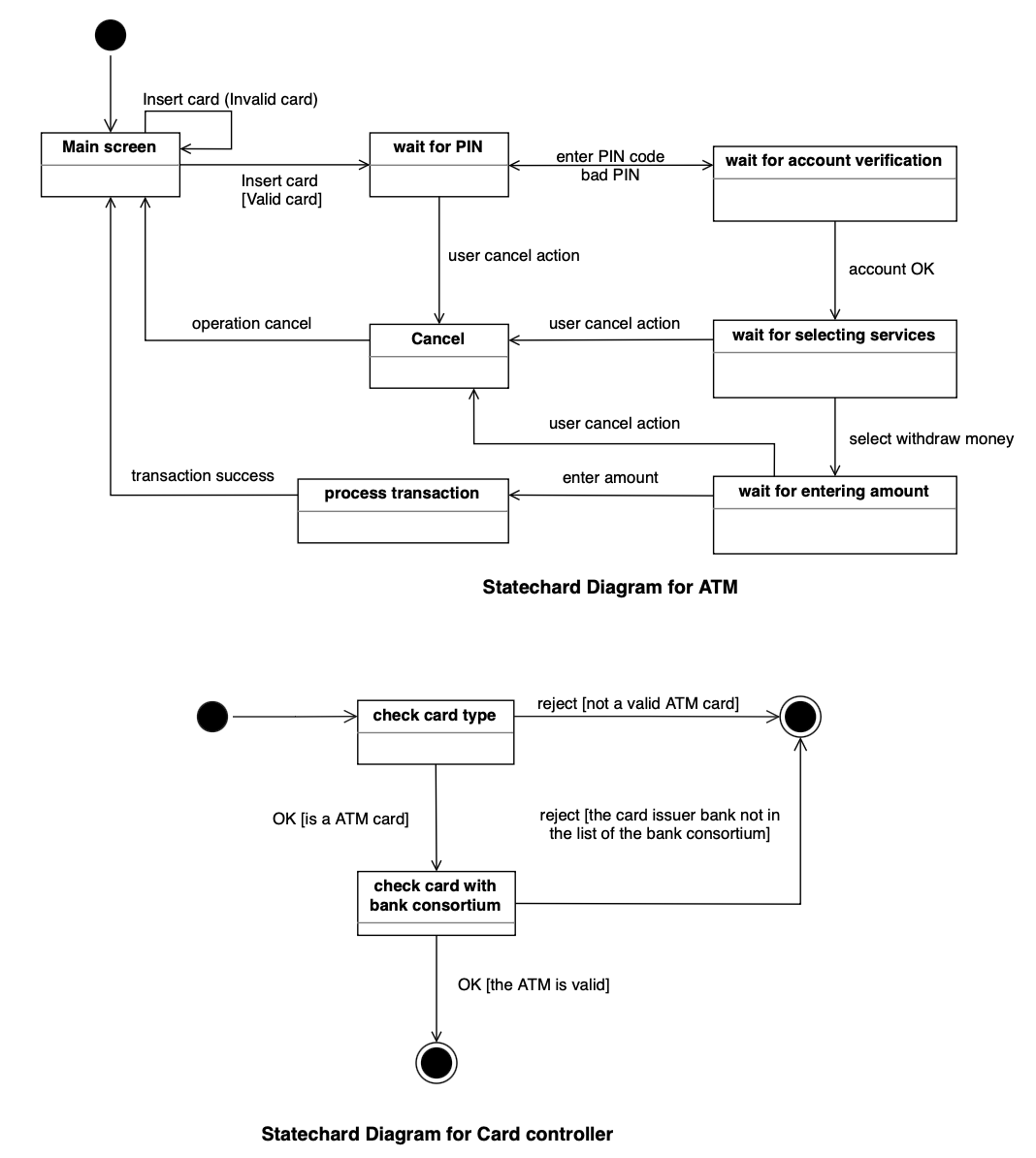
**Kiểm tra số tiền và số lượng tồn kho:**

* Nếu tiền < giá → Thông báo lỗi.
* Nếu hết hàng → Thông báo SoldOut + gợi ý.
* Nếu tiền = giá và còn hàng → Database cập nhật số lượng.
* Nếu tiền > giá → Database cập nhật số lượng + Register tính tiền thừa → Trả lại cho người dùng.
* Người dùng có thể rút lại tiền nếu muốn.

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

****

## Bài 6. Vẽ lược đồ :

****

## Bài 7. Phân tích dữ liệu cho ABC Bank

\_ ABC Bank có nhiều Branch.

* Mỗi Branch có một unique name, address (gồm No., Street, District, City, Region), phone numbers, fax numbers, email, và total number of employees làm việc tại đó.
* Một Branch có nhiều Employee, và mỗi Employee bắt buộc phải thuộc về một Branch.
* Luôn có một Employee được giao nhiệm vụ manage chi nhánh đó. Một Employee chỉ được manage đúng Branch mà mình thuộc về.

\_ Với mỗi Employee, cần lưu:

* unique code, first name, last name, date of birth, phone numbers, email, và home address (gồm No., Street, District, City).

\_ Ngân hàng phát hành 3 loại Account khác nhau cho khách hàng:

* Savings Account: lưu balance, interest rate, date opened.
* Checking Account: chỉ lưu balance, date opened (không có interest rate).
* Loan: lưu date loan taken, balance due, interest rate.

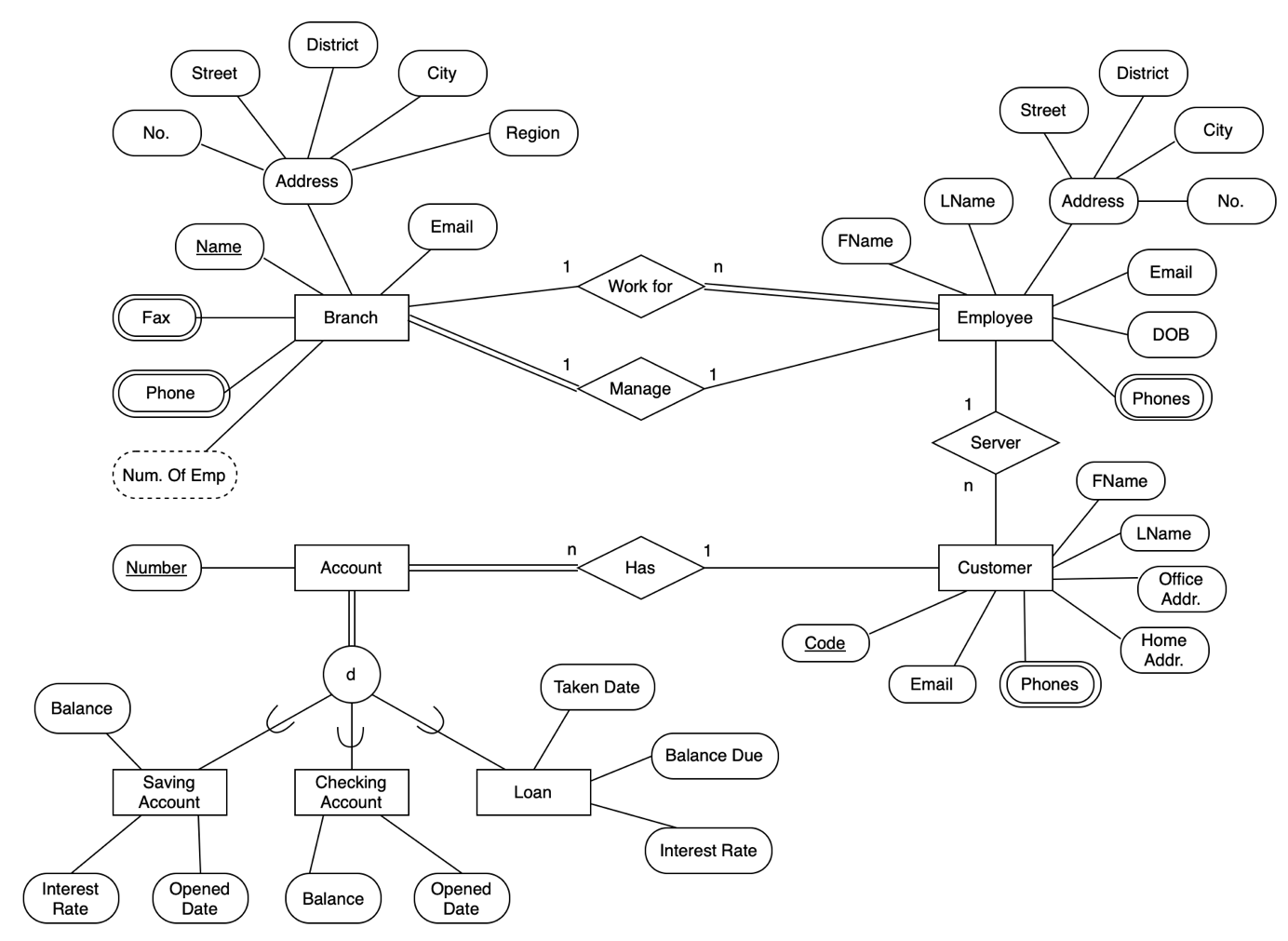
\_ Mỗi Customer có thể sở hữu nhiều Account, và tất cả các Account đều có một unique account number.

\_ Với mỗi Customer, cần lưu:

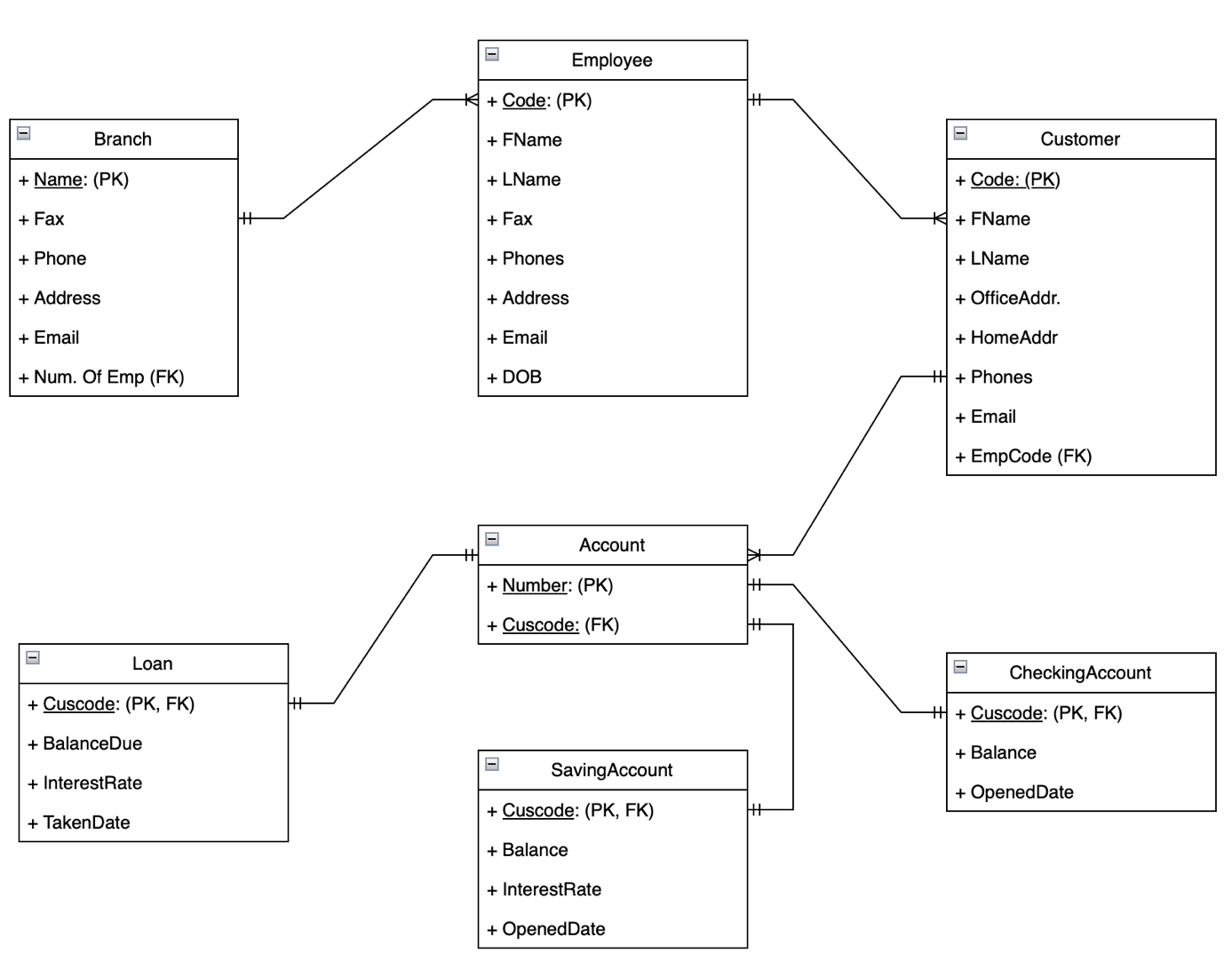
* unique code, first name, last name, home address, office address, phone numbers, email.

\_ Mỗi Customer được phục vụ bởi một Employee, và một Employee có thể phục vụ nhiều Customer.

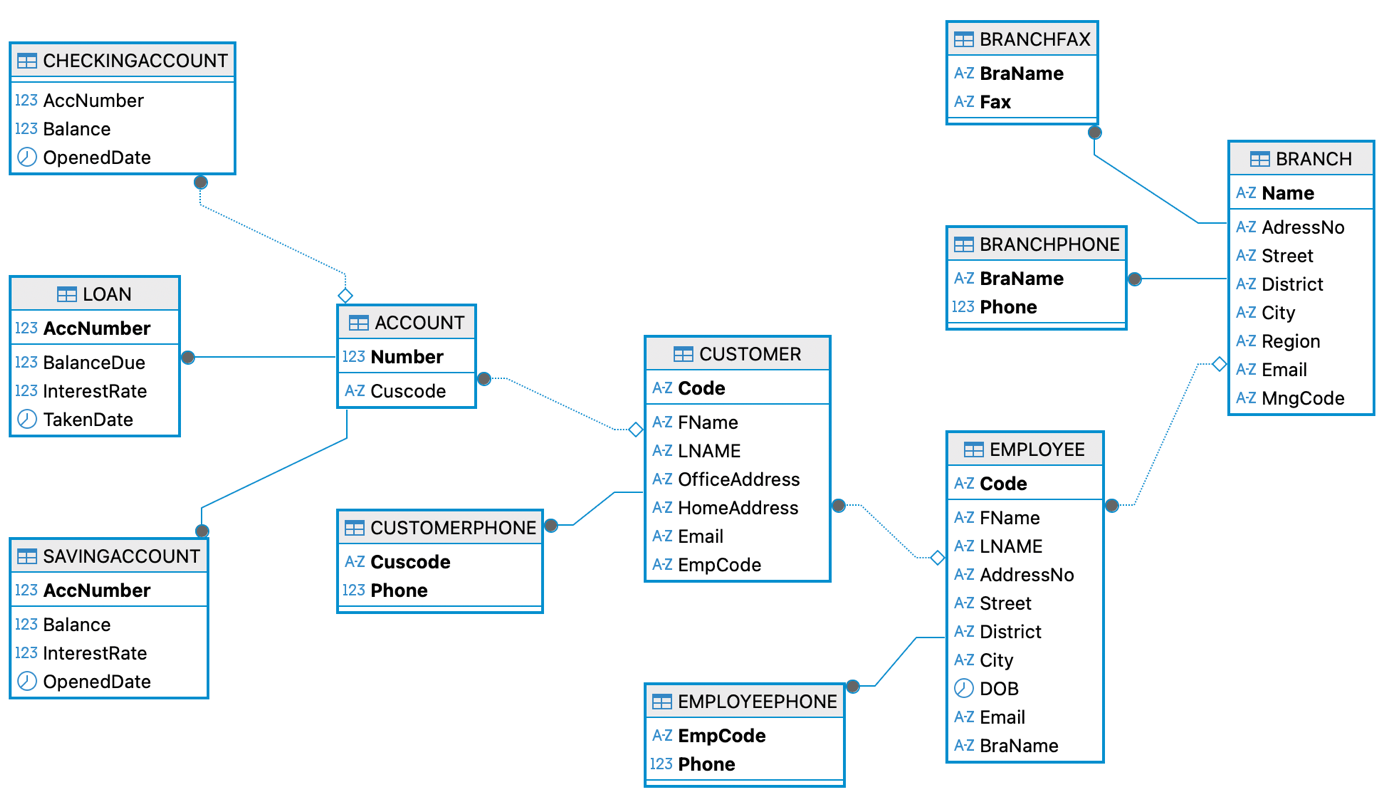
1. Thiết kế cơ sở dữ liệu ở mức khái niệm



1. Thiết kế cơ sở dữ liệu ở mức luận lý



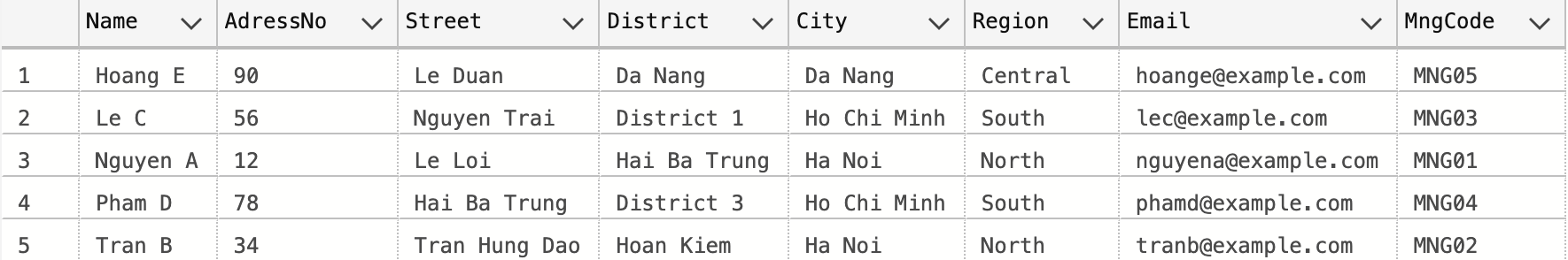
3. Thiết kế cơ sở dữ liệu ở mức vật lý

****

**Bài tập:**

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

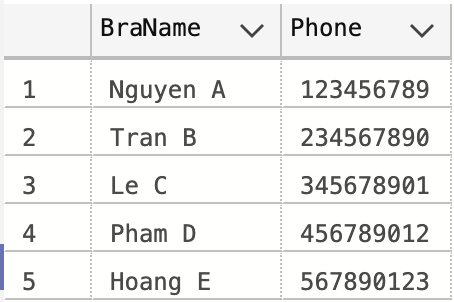
* Bảng BRANCH



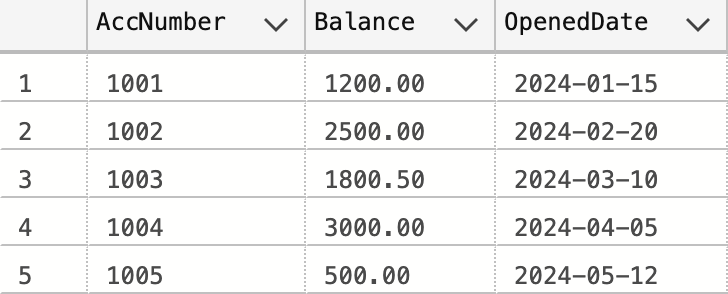
* Bảng BRANCHFAX



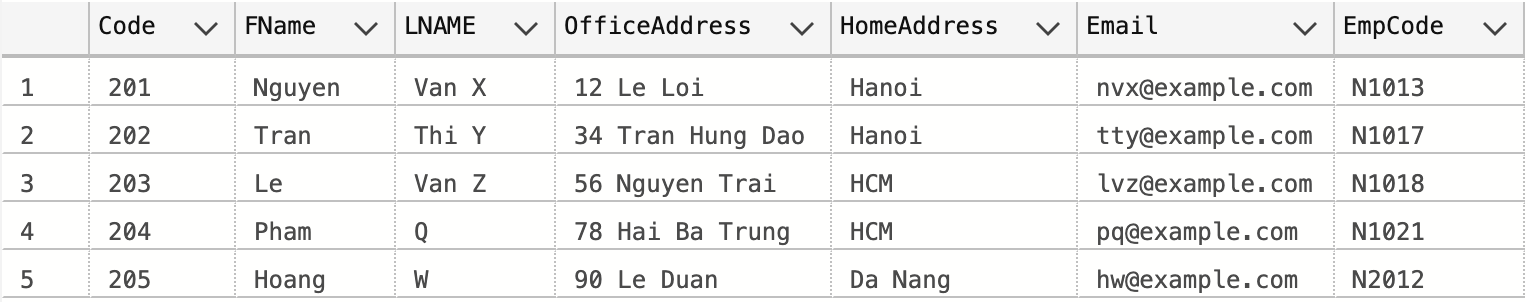
* Bảng BRANCHPHONE



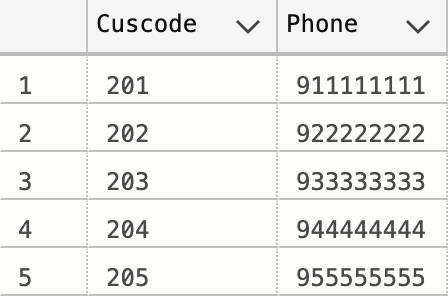
* Bảng CHECKINGACCOUNT



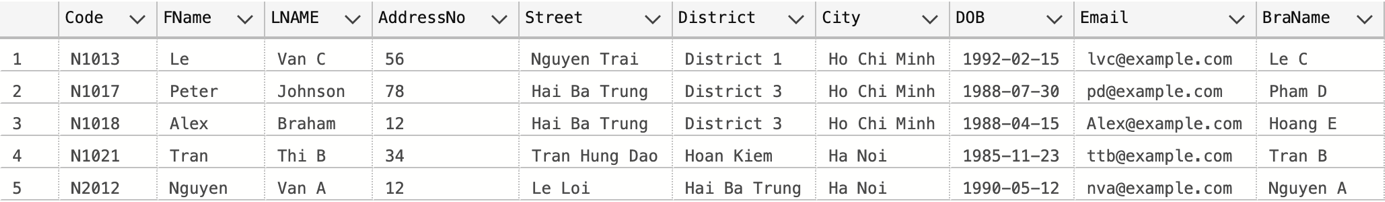
* Bảng CUSTOMER



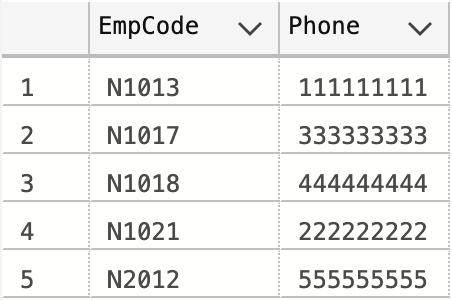
* Bảng CUSTOMERPHONE



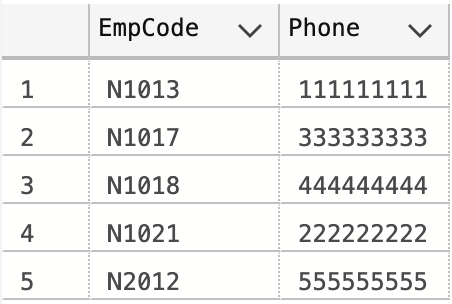
* Bảng EMPLOYEE



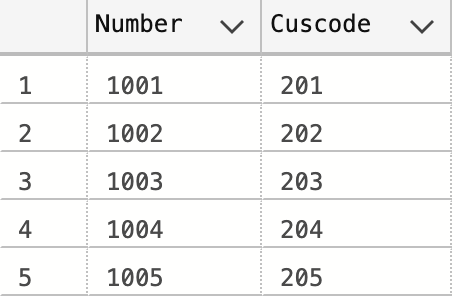
* Bảng EMPLOYEEPHONE



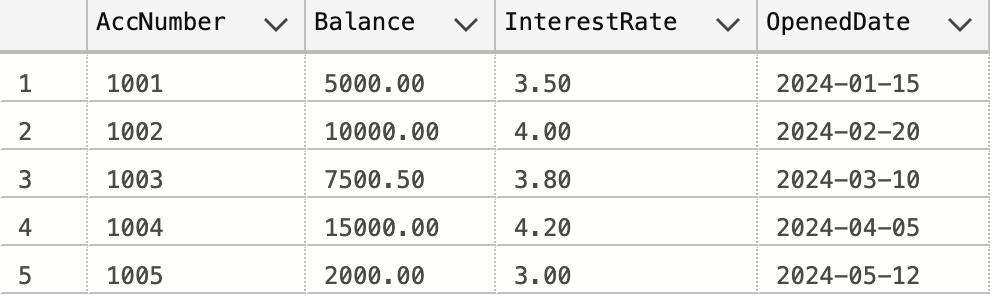
* Bảng LOAN



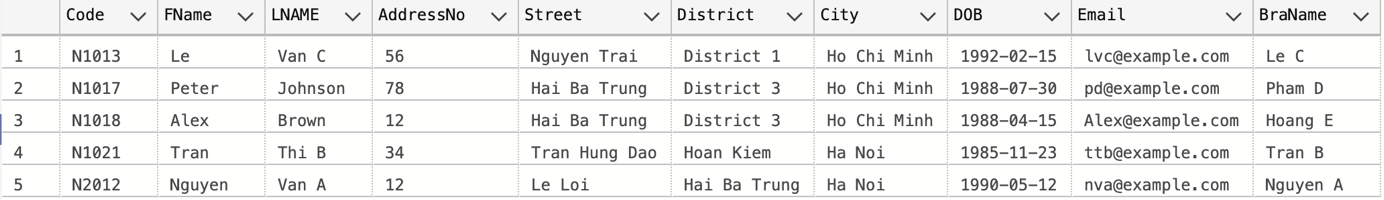
* Bảng ACCOUNT



* Bảng SAVINGACCOUNT



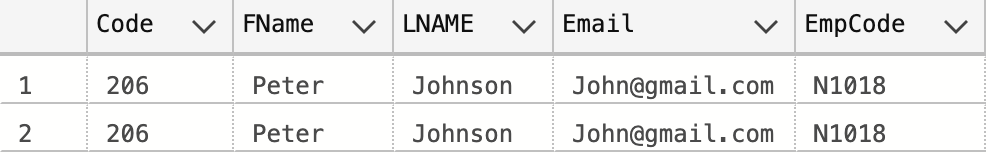
1. Đổi LName của nhân viên có mã ‘N1018’ thành ‘Brown’



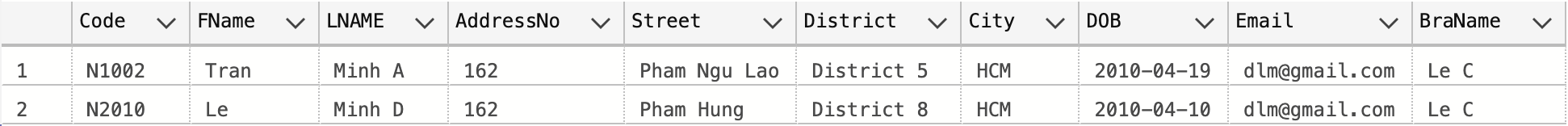
1. Xóa nhân viên có mã “N1080”. Giải thích điều gì sẽ xảy ra với những khách hàng bị nhân viên này sa thải.

* Tất cả các khách hàng thuộc nhân viên có mã ‘N1080’ cũng sẽ bị xóa luôn vì bảng EMPLOYEE và bảng CUSTOMER có quan hệ khóa ngoại ràng buộc thông qua mã nhân viên.

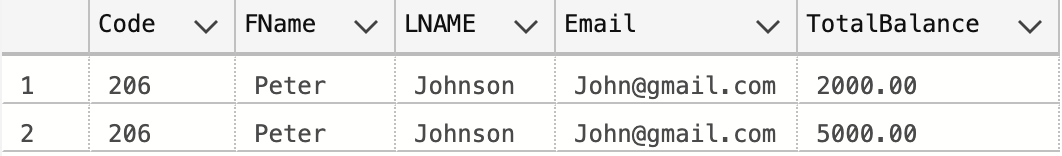
1. Lấy tất cả số tài khoản của một khách hàng có tên là Peter Johnson.



1. Tìm tất cả nhân viên có sinh nhật vào tháng 4 năm 2010



1. Lấy tổng số dư của tất cả các tài khoản tiết kiệm của một khách hàng tên “Peter Johnson” (Giả sử anh ta có nhiều tài khoản tiết kiệm)





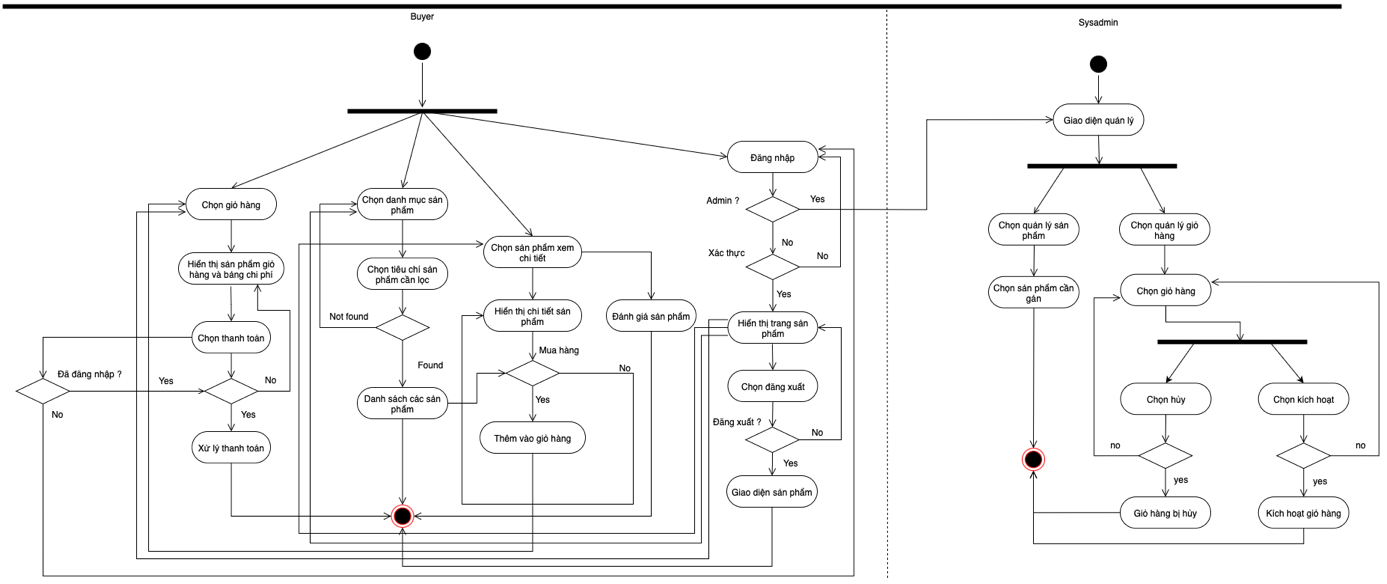
## Bài 8: Bài tập ứng dụng

1. Các yêu cầu chức năng và phi chức năng

|  |  |
| --- | --- |
| Các yêu cầu chức năng | |
| Danh mục sản phẩm (Product Catalog) | Người mua có thể:   * Duyệt danh sách sản phẩm. * Lọc sản phẩm theo tên, giá. * Sắp xếp sản phẩm theo tên, giá. * Xem chi tiết sản phẩm (tên, mô tả, tồn kho, địa chỉ kho, trạng thái nổi bật, đánh giá).   Quản trị viên (SysAdmin) có thể:   * Thêm, sửa, xóa sản phẩm. * Gán sản phẩm vào kho hiện có. |
| Giỏ hàng (Shopping Cart) | Người mua có thể:   * Thêm sản phẩm vào giỏ từ danh mục hoặc trang chi tiết. * Xóa sản phẩm khỏi giỏ. * Xem giỏ hàng và bảng tổng hợp (tổng chi phí, tổng sau khuyến mãi, chi phí vận chuyển, tiết kiệm, tổng giá trị đơn hàng). * Thực hiện thanh toán từ giỏ hàng.   Quản trị viên có thể:   * Xem giỏ hàng của mọi người dùng. * Kích hoạt/vô hiệu hóa giỏ hàng không hợp lệ. |
| Quy trình thanh toán (Payment Process) | Sau khi người mua nhấn **Thanh toán**:   * Hệ thống xác thực thông tin sản phẩm. * Xử lý thanh toán. * Gửi email thông báo cho người mua. |
| Tồn kho (Inventory) | Quản trị viên có thể quản lý kho hàng (thêm, sửa, xóa kho). |
| Đánh giá (Rating) | Người mua có thể đánh giá 1-5 sao. |
| Kiểm soát truy cập (Access Control) | Người dùng có thể đăng nhập/đăng xuất.  Nếu là Người dùng → vào trang danh mục sản phẩm.  Nếu là Admin → vào trang quản trị. |
| Thiết lập ban đầu (Initial Setup) | Tạo người dùng quản trị (Admin).  Tạo ít nhất 2 người mua (Người dùng).  Khởi tạo dữ liệu mẫu (sản phẩm, kho, đánh giá). |

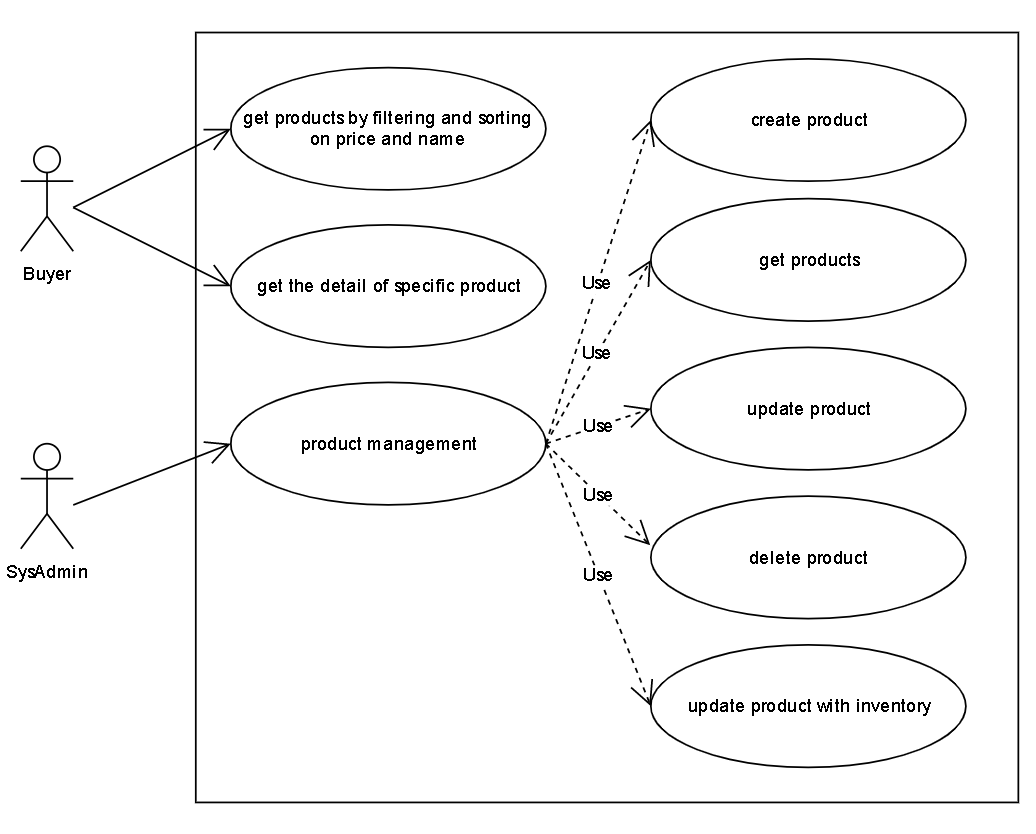
|  |  |
| --- | --- |
| Các yêu cầu phi chức năng | |
| Hiệu năng (Performance) | Hệ thống phải phản hồi nhanh (danh mục sản phẩm tải trong < 2 giây).  Giỏ hàng cập nhật tức thì khi thêm/xóa sản phẩm. |
| Bảo mật (Security) | Tài khoản và mật khẩu được mã hóa.  Kiểm soát quyền truy cập (Người dùng/Admin).  Giao dịch thanh toán an toàn (SSL/TLS). |
| Tính sẵn sàng (Availability) | Hệ thống hoạt động 24/7.  Ít downtime, đảm bảo khả năng xử lý nhiều người mua cùng lúc. |
| Khả năng mở rộng (Scalability) | Hỗ trợ mở rộng số lượng sản phẩm, kho và người dùng lớn. |
| Khả năng sử dụng (Usability) | Giao diện thân thiện, dễ sử dụng cho Người dùng và Admin.  Chức năng lọc, sắp xếp, tìm kiếm trực quan. |
| Khả năng bảo trì (Maintainability) | Code dễ chỉnh sửa, hệ thống dễ nâng cấp.  Có log hệ thống để kiểm tra lỗi. |
| Tương thích (Compatibility) | Hỗ trợ trên nhiều trình duyệt (Chrome, Edge, Safari…).  Hỗ trợ trên desktop và mobile. |

1. Vẽ qui trình nghiệp vụ cho website CoolStore

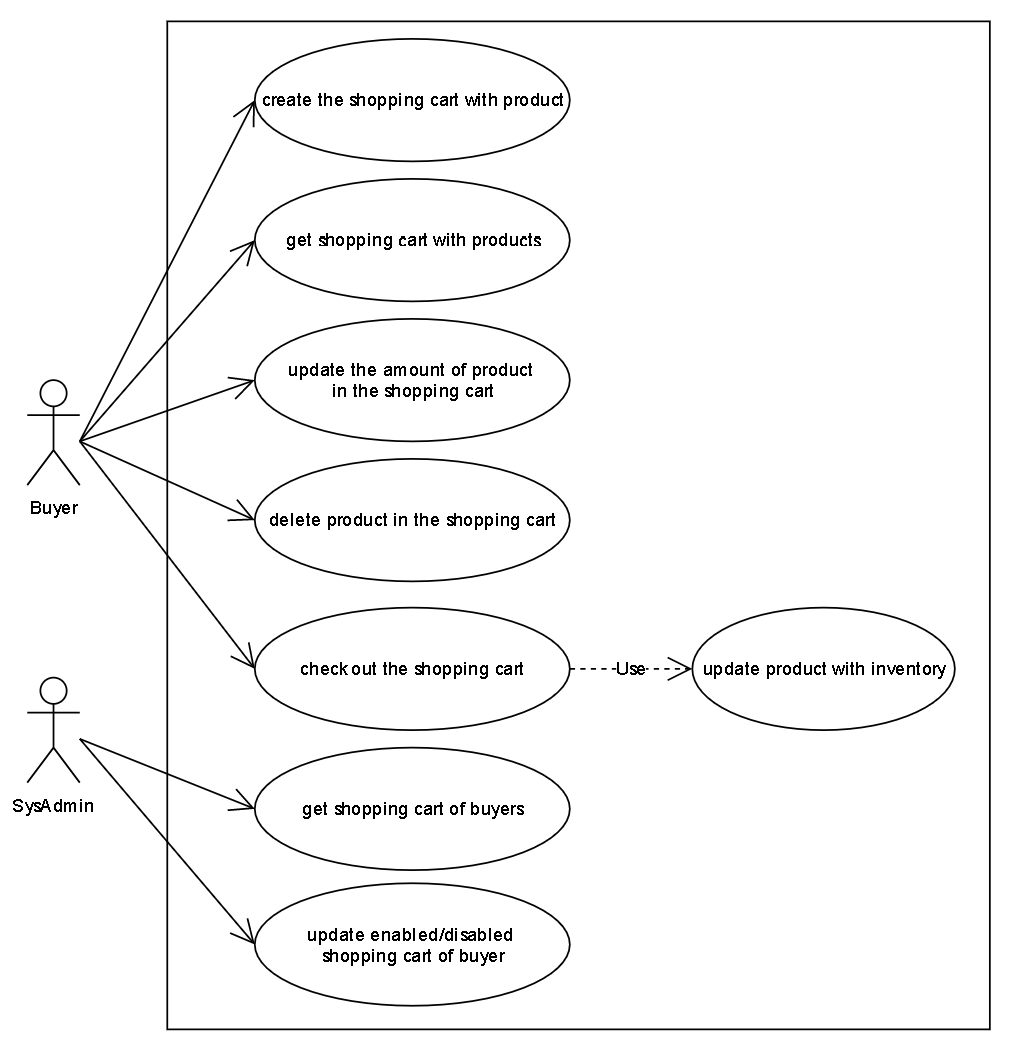


1. Xây dựng use case diagram cho website CoolStore.

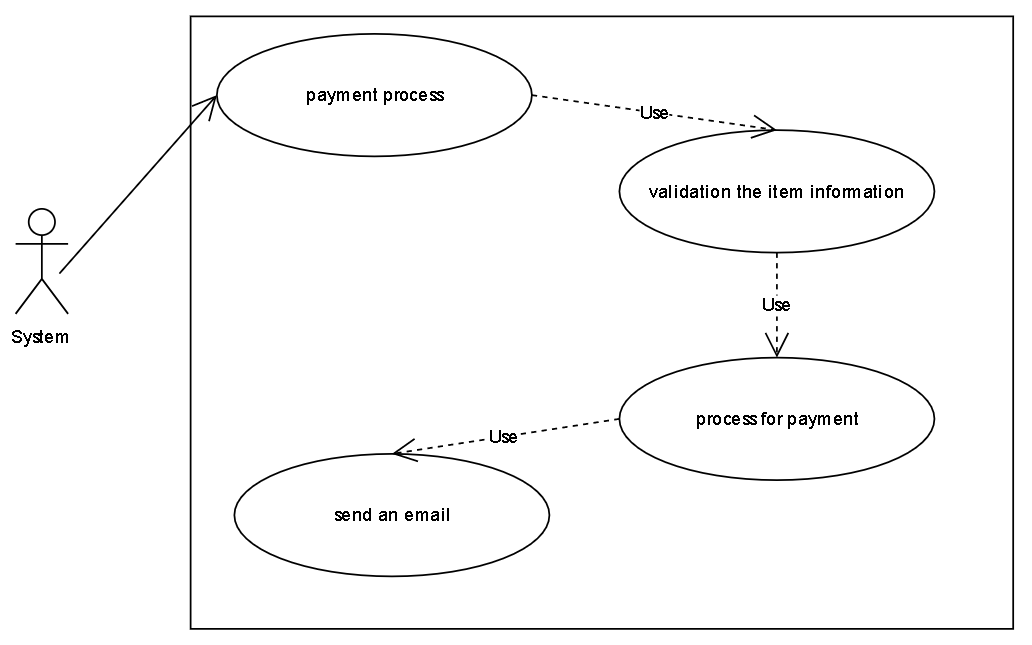
* Product catalog



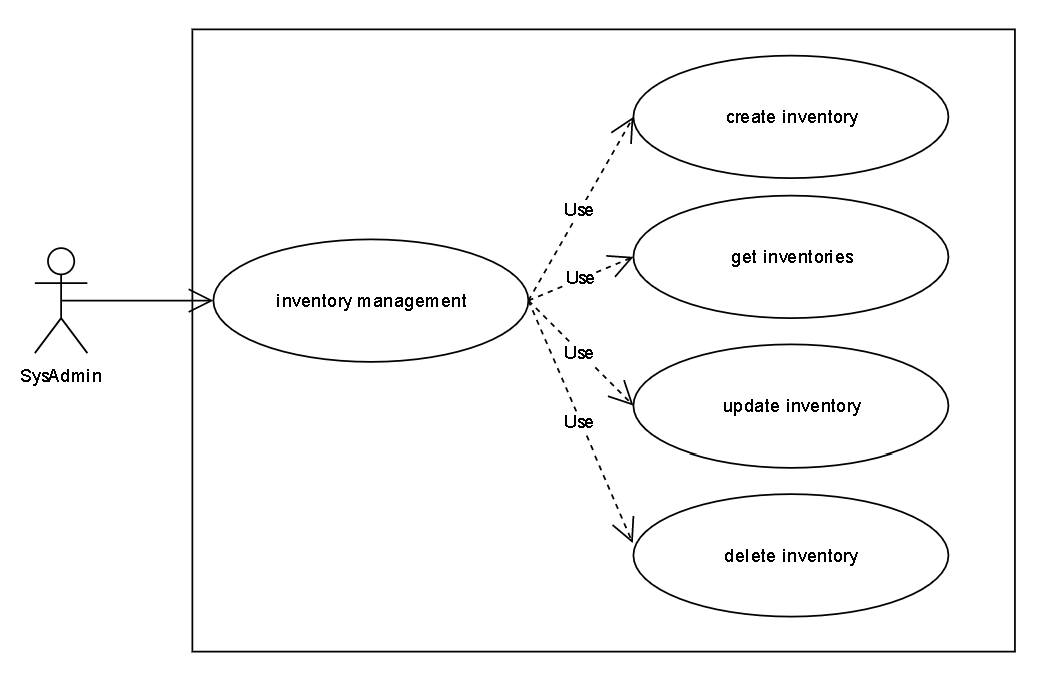
* Shopping cart



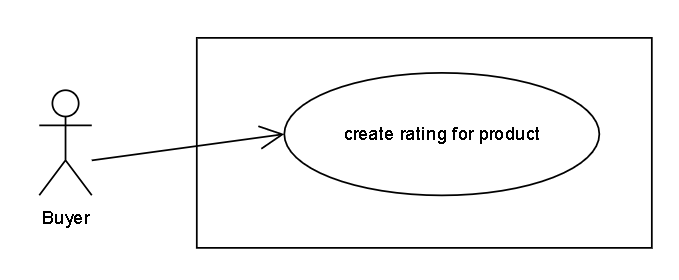
* Payment Process



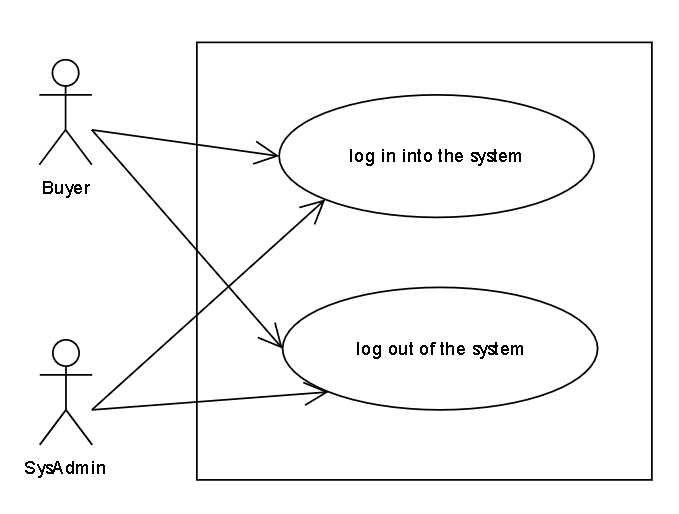
* Inventory



* Rating



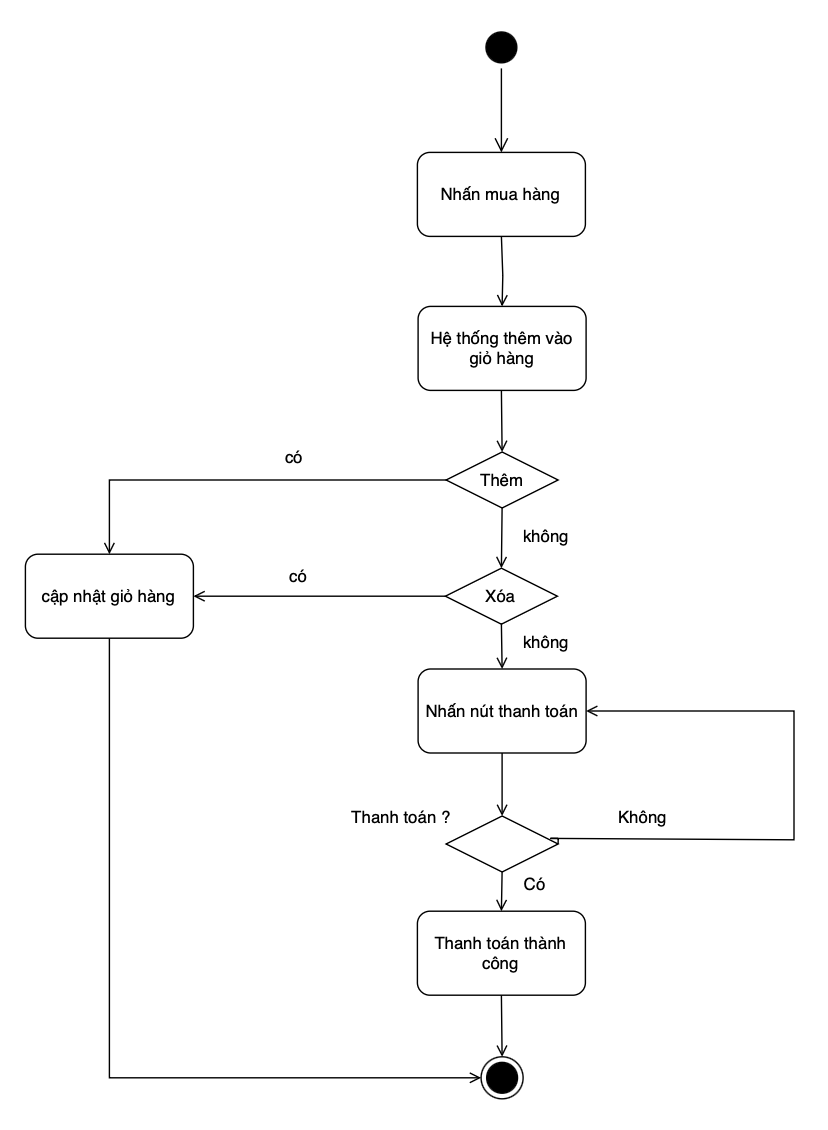
* Acess control



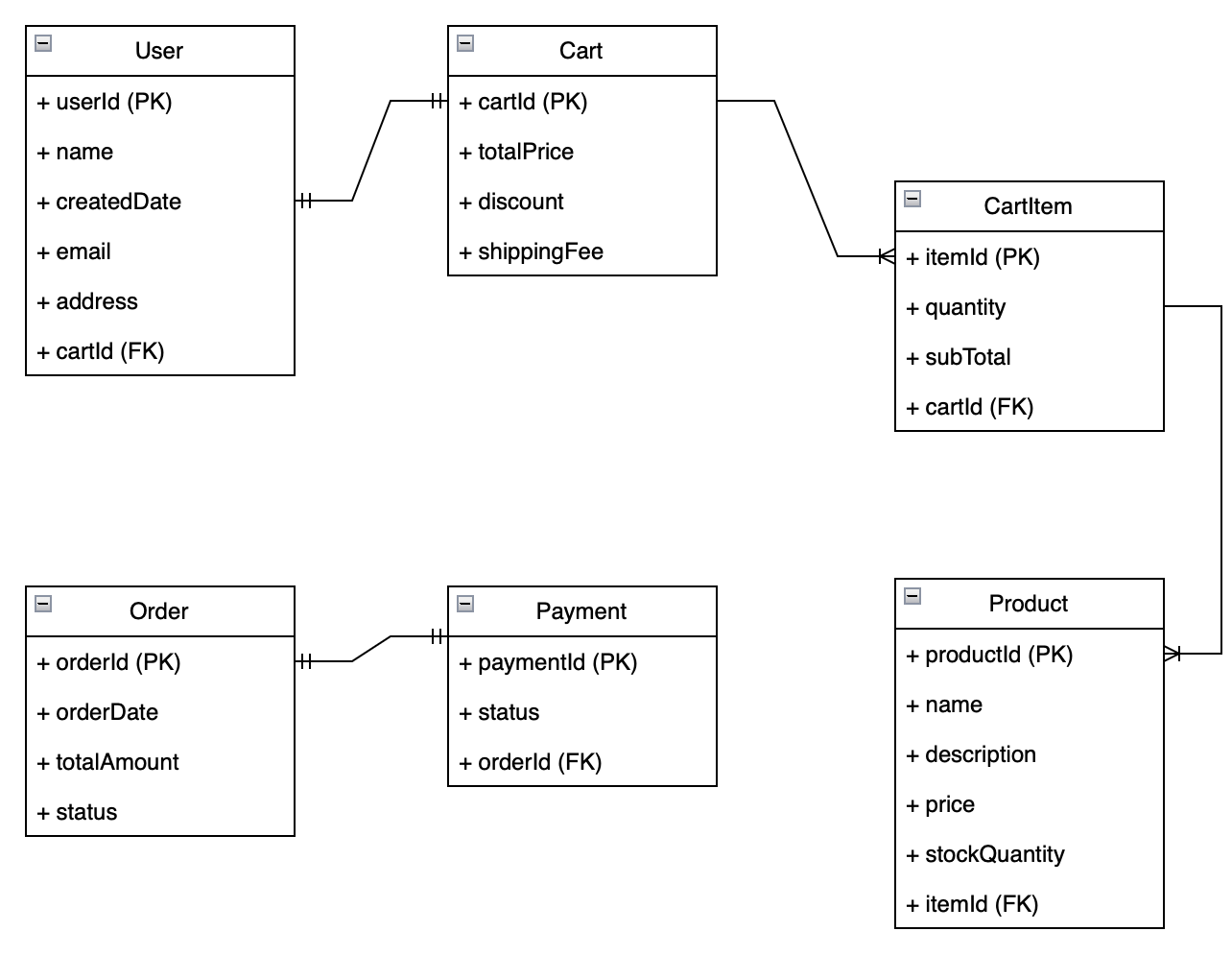
1. Mô tả chi tiết một Use Case "Mua sản phẩm từ trang chi tiết sản phẩm". Trình bày theo mẫu chuẩn gồm: Tên use case, Tác nhân chính, Mục tiêu, Tiền điều kiện, Luồng chính, Luồng thay thế, Hậu điều kiện, Ghi chú (nếu có).

|  |  |  |
| --- | --- | --- |
| **Use Case Number:** | **UC1** | |
| **Use case Name:** | Mua sản phẩm từ trang chi tiết sản phẩm | |
| **Actor (s):** | Khách hàng | |
| **Maturity:** | Focused | |
| **Summary:** | Use case này cho phép | |
| **Basic Course of Events:** | **Actor Action** | **System Response** |
| 1. Người dùng đăng nhập thành công vào hệ thống |  |
|  | 2. Hệ thống hiển thị giao diện các sản phẩm. |
| 3. Người dùng nhấp vào sản phẩm muốn xem |  |
|  | 4. Hệ thống hiển thị chi tiết sản phẩm |
| 5. Người dùng chọn mua hàng |  |
|  | 6. Hệ thống kiểm tra số lượng tồn kho. Nếu còn hàng thì tiếp tục. |
| 8. Hệ thống thêm sản phẩm vào **giỏ hàng** của khách hàng |  |
|  | 9. Hệ thống cập nhật bảng tổng hợp giỏ hàng (tổng chi phí, khuyến mãi, vận chuyển, tổng đơn hàng) |
| 10. Hệ thống thông báo: "Sản phẩm đã được thêm vào giỏ hàng" |  |
| **Alternative Paths:** | **A1. Sản phẩm hết hàng** | |
|  | 1. Tại bước 4, nếu hệ thống phát hiện sản phẩm đã hết hàng |
|  | 2. Hệ thống hiển thị thông báo "Sản phẩm hiện đã hết hàng, vui lòng chọn sản phẩm khác" |
|  | 1. Kết thúc use case |
| **A2. Người dùng chưa đăng nhập** | |
|  | 1. Hệ thống chuyển hướng đến trang đăng nhập. |
| 2. Người dùng đăng nhập |  |
|  | 3. Sau khi đăng nhập thành công, hệ thống đưa người dùng trở lại trang chi tiết sản phẩm. |
| 3. Người dùng tiếp tục thực hiện bước 3 của luồng chính. |  |
| **Extension Points:** | None | |
| **Pre-conditions:** | \_ Khách hàng đã đăng nhập.  \_ Khách hàng đã truy cập vào trang chi tiết sản phẩm cần mua.  \_ Sản phẩm còn hàng trong kho. | |
| **Post-conditions** | \_ Sản phẩm đã được thêm thành công vào giỏ hàng của khách.  \_ Giỏ hàng và bảng tổng hợp được cập nhật chính xác. | |

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.

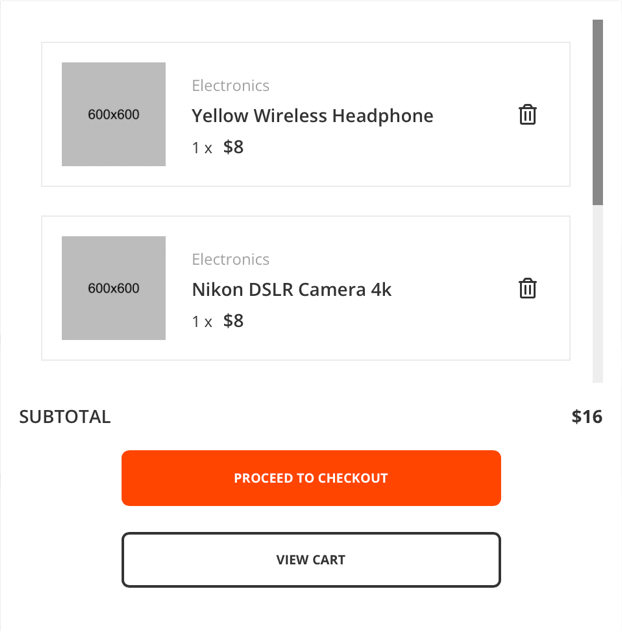


1. Thiết kế sơ đồ Class Diagram cho module Giỏ hàng.

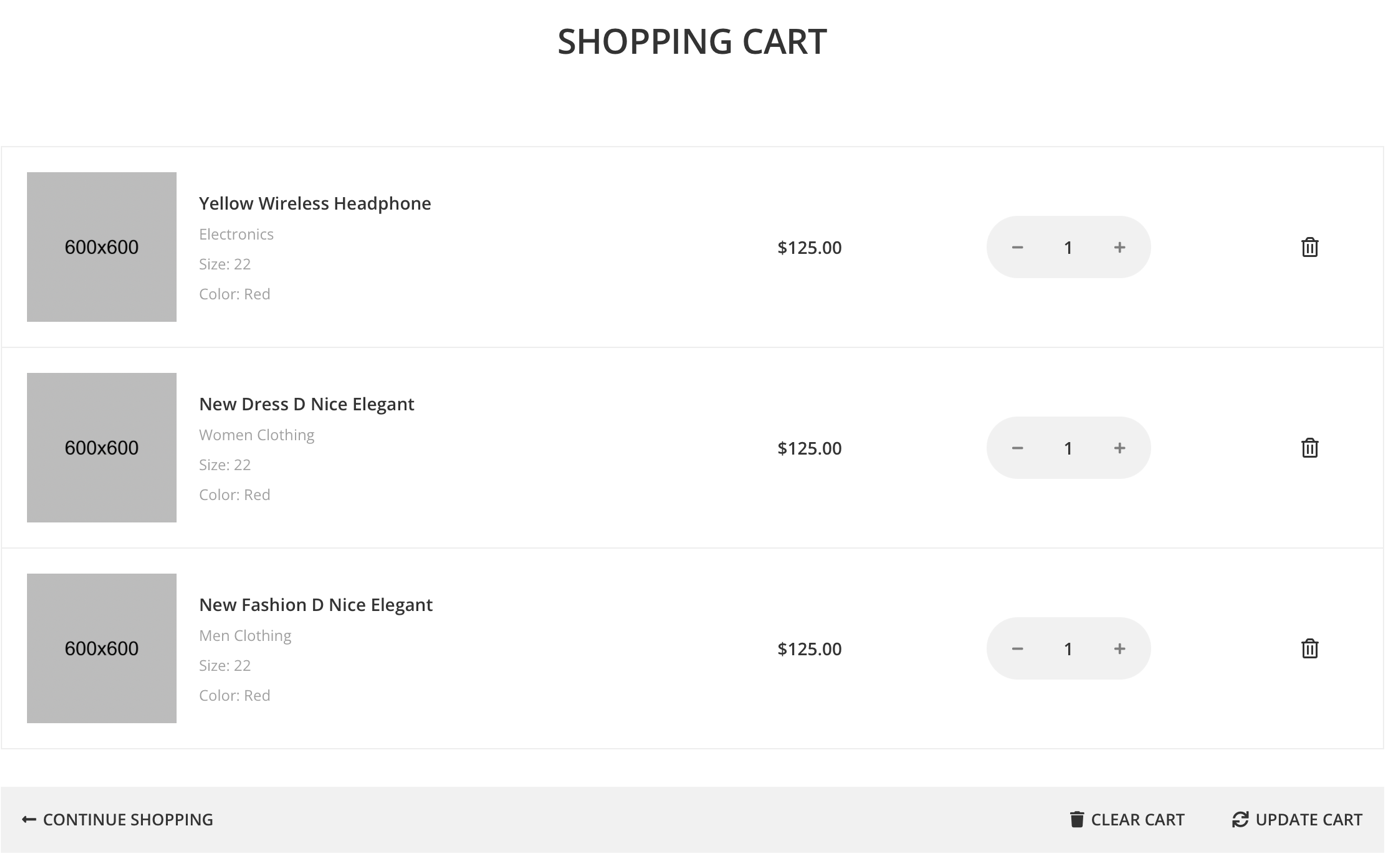


1. Thiết kế giao diện màn hình cho chức năng “Quản lý giỏ hàng” (có thể thiết kế một hoặc nhiều màn hình) gồm các thông tin:

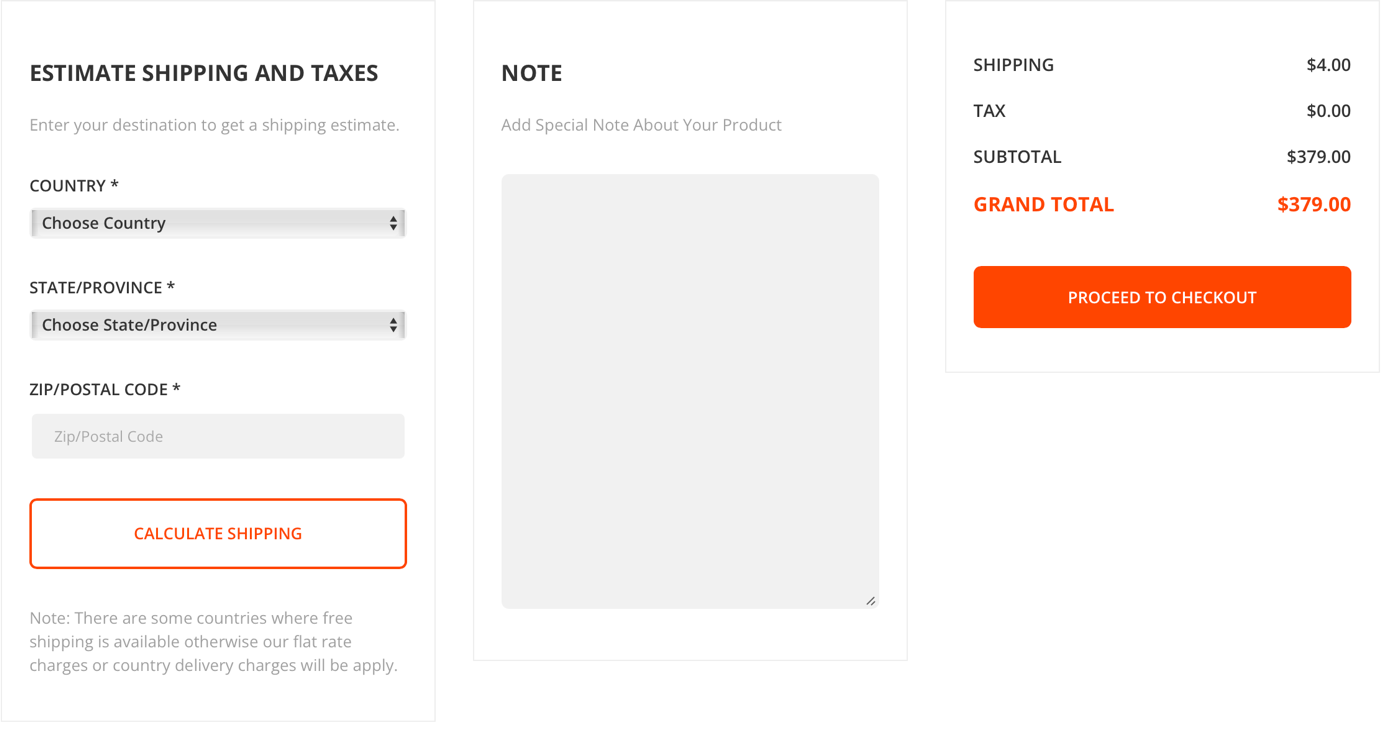
* Danh sách sản phẩm trong giỏ hàng: tên sản phẩm, hình ảnh, số lượng, giá từng món, tổng phụ.



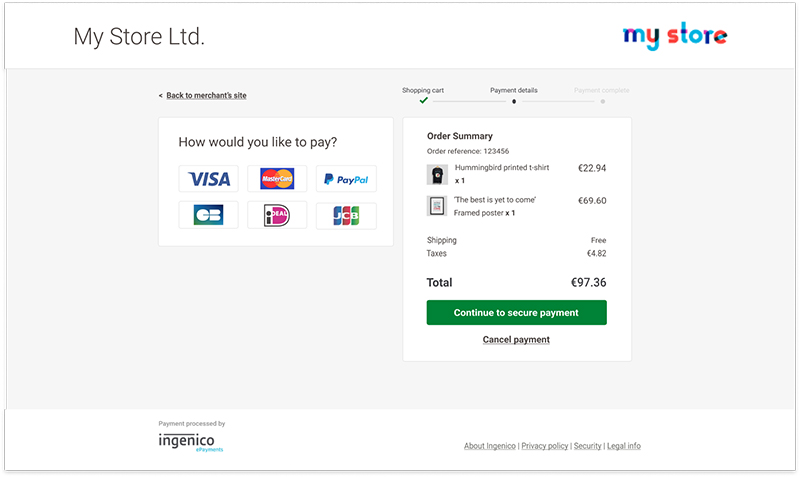
* Chức năng thao tác nhanh: Tăng/giảm số lượng sản phẩm; Xóa sản phẩm khỏi giỏ; Lưu sản phẩm để mua sau.



* 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.



1. 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 (Lớp trình bày)**

**Chức năng**: giao diện người dùng (UI/UX), hiển thị dữ liệu và nhận thao tác từ người dùng.

**Công nghệ gợi ý**:

Web: Angular / React / Vue / Blazor

Mobile: Flutter / React Native

**Thành phần trong CoolStore**:

Trang **Danh mục sản phẩm** (filter, sort, xem chi tiết).

Trang **Giỏ hàng** (xem, thêm, xóa, cập nhật).

Trang **Thanh toán** (checkout, xác nhận đơn hàng).

Trang **Đánh giá sản phẩm**.

Trang **Đăng nhập / Quản trị.**

**2. Business Logic Layer (Lớp nghiệp vụ / dịch vụ)**

**Chức năng**: xử lý quy trình nghiệp vụ, kiểm tra logic, xác thực, gọi API tích hợp (như thanh toán).

**Công nghệ gợi ý**: ASP.NET Core (C#), Java Spring Boot, Node.js (NestJS).

**Thành phần trong CoolStore:**

* ProductService: quản lý sản phẩm, chi tiết sản phẩm, sản phẩm nổi bật.
* CartService: thêm / xóa sản phẩm, tính tổng chi phí, áp dụng khuyến mãi, phí vận chuyển.
* PaymentService: xác thực thông tin sản phẩm, kết nối cổng thanh toán, gửi email xác nhận.
* InventoryService: quản lý kho, cập nhật tồn kho khi có đơn hàng.
* RatingService: quản lý đánh giá sản phẩm (1–5 sao).
* AuthService: quản lý đăng nhập, phân quyền (Người dùng, SysAdmin).

**3. Data Access Layer (Lớp dữ liệu)**

**Chức năng**: giao tiếp với cơ sở dữ liệu, thao tác CRUD (Create, Read, Update, Delete).

**Công nghệ gợi ý**: SQL Server / PostgreSQL / MySQL

**ORM**: Entity Framework Core (C#), Hibernate (Java), TypeORM (Node.js)

**Thành phần dữ liệu CoolStore:**

* Bảng Product, Category
* Bảng Cart, CartItem
* Bảng Order, Payment, Shipment
* Bảng Warehouse (kho)
* Bảng User, Role (Người dùng, SysAdmin)
* Bảng Rating

k. 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.

|  |  |  |
| --- | --- | --- |
| Dữ liệu đầu vào | Bước thực hiện | Dữ liệu đầu ra |
| **Test case 1: Thêm sản phẩm vào giỏ hàng** | | |
| * Người dùng: Đã đăng nhập. * Sản phẩm A còn 5 * Người dùng nhấn thêm giỏ hàng với số lượng 2 | * Người dùng mở trang chi tiết sản phẩm A. * Nhập số lượng = 2. * Nhấn nút "Thêm vào giỏ hàng". | * Hệ thống hiển thị thông báo "Sản phẩm đã được thêm vào giỏ hàng". * Giỏ hàng của Người dùng xuất hiện sản phẩm A với số lượng 2. * Tổng giá trị giỏ hàng được cập nhật chính xác. |
| **Test case 2: Thêm số lượng lớn hơn tồn kho** | | |
| * Người dùng đã đăng nhập. * Sản phẩm B còn tồn kho = 5. * Người dùng chọn số lượng = 10. | * Người dùng mở trang chi tiết sản phẩm B. * Nhập số lượng = 10. * Nhấn "Thêm vào giỏ hàng". | * Hệ thống hiển thị thông báo lỗi "Số lượng yêu cầu vượt quá tồn kho". * Không thêm sản phẩm vào giỏ hàng. |
| **Test case 3: Thêm sản phẩm khi chưa đăng nhập** | | |
| * Người dùng chưa đăng nhập. * Sản phẩm C tồn kho = 20. * Người dùng nhấn "Thêm vào giỏ hàng". | * Người dùng mở trang chi tiết sản phẩm C. * Nhấn "Thêm vào giỏ hàng". | * Hệ thống chuyển hướng đến trang đăng nhập. * Sau khi đăng nhập thành công, hệ thống có thể tự động thêm sản phẩm C vào giỏ hàng hoặc yêu cầu Người dùng thao tác lại. |