**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO HỌC PHẦN KIỂM THỬ PHẦN MỀM

**PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**Giáo viên hướng dẫn: Thầy Đỗ Như Tài**

**Thành viên nhóm:**

**Đinh Trung Hội 3122411058**

**Huỳnh Phúc Hưng 3122411073**

**Nguyễn Hãi Đăng 3122411031**

**Thành phố Hồ Chí Minh, tháng 09 năm 2025**

**PHÂN CÔNG NHÓM**

|  |  |  |
| --- | --- | --- |
| **Tên Thành Viên** | **Phân công** |  |
| Đinh Trung Hội | Làm các yêu cầu của bài **5, 6 và bài tập ứng dụng** | 100% |
| Huỳnh Phúc Hưng | Làm các yêu cầu của bài **3, 4 và bài tập ứng dụng** | 100% |
| Nguyễn Hãi Đăng | Làm các yêu cầu của bài **1, 2, 7 và viết word** | 100% |

**MỤC LỤC**

**[DANH MỤC HÌNH ẢNH 1](#_Toc23641)**

**[NỘI DUNG 2](#_Toc929)**

[Bài 1: Qui trình nghiệp vụ 2](#_Toc25138)

[Bài 2: Mô hình khái niệm 3](#_Toc4914)

[Bài 3: Business use case 4](#_Toc31819)

*[a) Xác định và mô tả các tác nhân 5](#_Toc26153)*

*[b) Liệt kê và giải thích các trường hợp sử dụng trong sơ đồ: 6](#_Toc11459)*

*[c) Phân tích các mối quan hệ giữa Use Cases 6](#_Toc12615)*

*[d) Kịch bản chi tiết cho Use Case “Register for Courses” 7](#_Toc11250)*

*[e) Đề xuất cải tiến / bổ sung Use Cases 8](#_Toc12217)*

[Bài 4: Sơ đồ hoạt động và tương tác 8](#_Toc25535)

[Bài 5: Lược đồ lớp 10](#_Toc31117)

[Bài 6: Lược đồ 12](#_Toc4792)

[Bài 7: Phân tích dữ liệu cho ABC Bank 13](#_Toc9959)

*[1. Thiết kế Cơ sở dữ liệu ở mức Khái niệm, Luận lý và Vật lý 13](#_Toc15606)*

*[2. Query 15](#_Toc28128)*

**[BÀI TẬP ỨNG DỤNG 18](#_Toc273)**

[1) Các yêu cầu chức năng và phi chức năng 19](#_Toc6180)

[2) Vẽ qui trình nghiệp vụ 20](#_Toc22952)

[3) Xây dựng use case diagram 20](#_Toc22228)

[4) Mô tả chi tiết một Use Case 21](#_Toc22611)

[5) Sơ đồ Activity Diagram để mô tả quy trình mua hàng 22](#_Toc28222)

[6) Sơ đồ Class Diagram cho module Giỏ hàng. 23](#_Toc3318)

[7) Giao diện màn hình cho chức năng “Quản lý giỏ hàng” 23](#_Toc16471)

[8) Đề xuất cách tổ chức kiến trúc phần mềm cho hệ thống CoolStore theo mô hình kiến trúc ba lớp (3-tierarchitecture) 25](#_Toc9417)

[9) 3 ca kiểm thử cho chức năng “Thêm sản phẩm vào giỏ hàng” 26](#_Toc13353)

**DANH MỤC HÌNH ẢNH**

[Hình 1 : Qui trình nghiệp vụ ( mẫu ) 2](#_Toc25714)

[Hình 2 : Qui trình nghiệp vụ ( vẽ lại ) 2](#_Toc8718)

[Hình 3 : Mô hình khái niệm ( mẫu ) 3](#_Toc12956)

[Hình 4 : Mô hình khái niệm ( vẽ lại ) 3](#_Toc22707)

[Hình 5 : Business use case ( mẫu ) 4](#_Toc6078)

[Hình 6 : Business use case ( vẽ lại ) 4](#_Toc15309)

[Hình 7 : Sơ đồ hoạt động ( mẫu ) 8](#_Toc10445)

[Hình 8 : Sơ đồ hoạt động ( vẽ lại ) 9](#_Toc29848)

[Hình 9 : Sơ đồ tương tác ( mẫu ) 9](#_Toc10884)

[Hình 10 : Sơ đồ tương tác ( vẽ lại ) 10](#_Toc19537)

[Hình 11 : Lược đồ lớp cho bài toán Quản lý thư viện ( mẫu ) 10](#_Toc10046)

[Hình 12 : Lược đồ lớp cho bài toán Quản lý thư viện ( vẽ lại ) 11](#_Toc2673)

[Hình 13 : Lược đồ bài 6 ( mẫu ) 12](#_Toc719)

[Hình 14 : Lược đồ bài 6( vẽ lại ) 12](#_Toc28282)

[Hình 15 : CSDL mức khái niệm 13](#_Toc1939)

[Hình 16 : CSDL mức Luận lý 14](#_Toc15023)

[Hình 17 : CSDL mức Vật lý 14](#_Toc26136)

[Hình 18 : Kết quả câu 7b 17](#_Toc28707)

[Hình 19 : Kết quả câu 7c 17](#_Toc21785)

[Hình 20 : Kết quả câu 7d 18](#_Toc4487)

[Hình 21 : Kết quả câu 7e 18](#_Toc13298)

[Hình 22 : Kết quả câu 7f 18](#_Toc31040)

[Hình 23 : Quy trình nghiệp vụ cho website CoolStore 20](#_Toc12630)

[Hình 24 : Use case diagram cho website CoolStore 21](#_Toc21949)

[Hình 25 : sơ đồ Activity Diagram 22](#_Toc19012)

[Hình 26 : Sơ đồ Class Diagram 23](#_Toc16391)

[Hình 27 : Giao diện màn hình cho chức năng “Quản lý giỏ hàng” 25](#_Toc13623)

[Hình 28 : Kiến trúc 3-tier cho hệ thống CoolStore 25](#_Toc20622)

**NỘI DUNG**

**Bài 1: Qui trình nghiệp vụ**

|  |  |
| --- | --- |
| **Yêu cầu :** Vẽ lại qui trình nghiệp vụ sau (Độc giả, NV Tiếp tân, Thành viên) | |
| **Hình mẫu** | IMG_256  Hình 1: Qui trình nghiệp vụ ( mẫu ) |
| **Hình vẽ lại** | Hình 2: Qui trình nghiệp vụ ( vẽ lại ) |

**Bài 2: Mô hình khái niệm**

|  |  |
| --- | --- |
| **Yêu cầu:** Vẽ mô hình khái niệm sau | |
| **Hình mẫu** | **IMG_256**  Hình 3: Mô hình khái niệm ( mẫu ) |
| **Hình vẽ lại** | Hình 4: Mô hình khái niệm ( vẽ lại ) |

**Bài 3: Business use case**

|  |  |
| --- | --- |
| **Yêu cầu:** Vẽ lại business use case sau | |
| **Hình mẫu** | Hình 5: Business use case ( mẫu ) |
| **Hình vẽ lại** | Bai3-Trang-5.drawio  Hình 6: Business use case ( vẽ lại ) |

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

**Xác định:** Có 4 actor,các actor xuất hiện là:

1. Student (Sinh viên)
2. Billing System (Hệ thống thanh toán)
3. Professor (Giảng viên)
4. Registrar (Phòng đào tạo / Nhân viên quản lý đào tạo)

**Mô tả:**

1. **Student (Sinh viên):** Đăng ký môn học

Vai trò: Người sử dụng chính của hệ thống để đăng ký học phần.

Nhiệm vụ:

Chọn và đăng ký môn học.

Thông tin đăng ký sẽ được hệ thống xử lý và chuyển sang Billing System để tính học phí.

Mối quan hệ: Liên kết trực tiếp với use case Register for Courses.

1. **Billing System (Hệ thống thanh toán):** Xử lý học phí

Vai trò: Hệ thống bên ngoài, tích hợp với hệ thống quản lý đào tạo để xử lý học phí khi sinh viên đăng ký môn.

Nhiệm vụ:

Nhận dữ liệu từ use case Register for Courses.

Tính toán học phí, tạo hóa đơn, lưu trữ và hỗ trợ thanh toán.

Mối quan hệ: Kết nối trực tiếp với use case Register for Courses (do Student khởi tạo).

1. **Professor (Giảng viên): Chọn môn giảng dạy**

Vai trò: Người tham gia vào hệ thống để quản lý việc giảng dạy.

Nhiệm vụ:

Chọn các môn học mà họ sẽ giảng dạy (use case Select Courses to Teach).

Thông tin giảng dạy được cập nhật trong hệ thống để sinh viên có thể đăng ký.

Mối quan hệ: Liên kết trực tiếp với use case Select Courses to Teach.

1. **Registrar (Phòng đào tạo / Nhân viên quản lý đào tạo):** quản lý toàn bộ dữ liệu (chương trình, sinh viên, giảng viên)

Vai trò: Quản trị viên chính của hệ thống, chịu trách nhiệm duy trì thông tin quản lý học vụ.

Nhiệm vụ:

Maintain Curriculum: Quản lý chương trình học.

Maintain Student Information: Cập nhật, chỉnh sửa dữ liệu sinh viên.

Maintain Professor Information: Quản lý hồ sơ giảng viên.

Các thao tác này đều cần qua bước Registrar Validation để xác minh, đảm bảo dữ liệu chính xác.

Mối quan hệ: Liên kết với nhiều use case quản lý trong sơ đồ.

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

**Liệt kê:**

* Register for Courses (Đăng ký môn học)
* Select Courses to Teach (Chọn môn giảng dạy)
* Maintain Curriculum (Quản lý chương trình học)
* Maintain Student Information (Quản lý thông tin sinh viên)
* Maintain Professor Information
* Registrar Validation (Xác thực bởi phòng đào tạo)

**Giải thích:**

**Register for Courses:** Do Student thực hiện, sinh viên chọn và đăng ký các môn học trong học kỳ. Hệ thống sau đó liên kết với Billing System để xử lý học phí.

**Select Courses to Teach:** Do Professor thực hiện, giảng viên chọn môn mà mình sẽ phụ trách giảng dạy trong học kỳ.

**Maintain Curriculu:** Do Registrar thực hiện,c ập nhật, chỉnh sửa chương trình đào tạo: các môn học, số tín chỉ, cấu trúc học kỳ.

**Maintain Student Information:** Do Registrar thực hiện, cập nhật dữ liệu cá nhân, hồ sơ học tập, trạng thái của sinh viên.

**Maintain Professor Information:** Do Registrar thực hiện, quản lý dữ liệu giảng viên: thông tin cá nhân, bộ môn, môn dạy.

**Registrar Validation:** Một use case hỗ trợ, được gọi khi cần xác thực hoặc là kiểm tra tính hợp lệ dữ liệu (chương trình học, thông tin sinh viên, thông tin giảng viên).

1. **Phân tích các mối quan hệ giữa Use Cases**

**Register for Courses Billing System:** không phải include/extend, mà là tương tác trực tiếp với actor ngoài để xử lý thanh toán.

**Maintain Curriculum «uses» Registrar Validation:** Khi quản lý chương trình học, bắt buộc phải gọi đến chức năng xác thực để đảm bảo dữ liệu hợp lệ.

**Maintain Student Information «uses» Registrar Validation:** Khi cập nhật thông tin sinh viên, hệ thống dùng chức năng xác thực để đảm bảo dữ liệu đúng.

**Maintain Professor Information «uses» Registrar Validation:** Khi cập nhật hồ sơ giảng viên, cũng cần thông qua xác thực của hệ thống.

**Các mối quan hệ «uses» (tương đương «include») nghĩa là:** use case chính luôn gọi use case con trong quá trình thực hiện.

1. **Kịch bản chi tiết cho Use Case “Register for Courses”**

**Tên use case:** Register for Courses

**Tác nhân chính:** Student

**Mục tiêu:** Đăng ký thành công một hoặc nhiều học phần trong học kỳ và nhận được thông báo xác nhận từ hệ thống.

**Tiền điều kiện:**

Sinh viên đã đăng nhập vào hệ thống.

Học kỳ đang trong thời gian mở đăng ký.

**Luồng chính:**

1. Sinh viên chọn chức năng “Register for Courses”.
2. Hệ thống hiển thị danh sách các học phần mà sinh viên có thể đăng ký.
3. Sinh viên chọn học phần mà mình mong muốn.
4. Hệ thống sẽ kiểm tra:

Sinh viên đã đáp ứng đủ các tiền điều kiện (điều kiện tiên quyết) chưa.

Lớp học còn slot hay không không.

1. Nếu hợp lệ, hệ thống tạm thời ghi nhận đăng ký.
2. Hệ thống chuyển thông tin đăng ký sang phần Billing System để tính học phí.
3. Billing System trả về kết quả (thành công/thất bại).
4. Nếu thành công: hệ thống lưu đăng ký, cập nhật số lượng sinh viên đã có trong lớp (Vd: 27/40 em) và hiển thị xác nhận cho sinh viên.

**Luồng thay thế:**

Nếu lớp đã đầy thì hệ thống sẽ thông báo lỗi và đề nghị sinh viên chọn lớp hoặc học phần khác.

Nếu điều kiện tiên quyết chưa đạt yêu cầu thì hệ thống sẽ thông báo không thể đăng ký.

Nếu sinh viên thực hiện thanh toán thất bại thì hệ thống sẽ hủy bỏ đăng ký và yêu cầu sinh viên thử lại.

**Hậu điều kiện:**

Nếu thành công: môn học được thêm vào danh sách đăng ký của sinh viên, Billing System đã ghi nhận học phí.

Nếu thất bại: không có thay đổi dữ liệu, sinh viên vẫn chưa đăng ký môn học.

1. **Đề xuất cải tiến / bổ sung Use Cases**

**Đề xuất cải tiến:** Theo bản thân em, sơ đồ trên có vẻ đã ổn, chỉ cần bổ sung vài use case để cho nó trông được đầy đủ hơn thôi.

**Bổ sung Use Cases:**

Drop Course (Hủy môn học): Cho phép các bạn sinh viên hủy đăng ký một học phần trước hạn chót, có thể là do không đủ kinh phí hoặc không có nhu cầu tiếp thu kiến thức, trải nghiệm học phần đó.

View Course Catalog (Xem danh mục môn học): Sinh viên có thể tra cứu các học phần mà mình đã đăng ký được mở trong học kỳ.

**Bài 4: Sơ đồ hoạt động và tương tác**

|  |  |
| --- | --- |
| **Yêu cầu:** Vẽ sơ đồ hoạt động | |
| **Hình mẫu** | Hình 7: Sơ đồ hoạt động ( mẫu ) |
| **Hình vẽ lại** | Bai4-trang1.drawio  Hình 8: Sơ đồ hoạt động ( vẽ lại ) |

Tóm tắt ý nghĩa của sồ đồ: Sơ đồ hoạt động này ta thấy được mô tả dòng chảy xử lý của hệ thống bán hàng tự động (giống máy bán hàng tự động bên ngoài chúng ta hay gặp), tập trung vào trình tự công việc và các nhánh điều kiện (đủ tiền, hết hàng, sản phẩm thay thế, vân vân mây mây).

|  |  |
| --- | --- |
| **Yêu cầu:** Vẽ sơ đồ tương tác | |
| **Hình mẫu** | Hình 9: Sơ đồ tương tác ( mẫu ) |
| **Hình vẽ lại** | Bai4-trang2.drawio  Hình 10: Sơ đồ tương tác ( vẽ lại ) |

Tóm tắt ý nghĩa của sồ đồ: Sơ đồ hoạt động này ta thấy được mô tả dòng chảy xử lý của hệ thống bán hàng tự động (giống máy bán hàng tự động bên ngoài chúng ta hay gặp), tập trung vào trình tự công việc và các nhánh điều kiện (đủ tiền, hết hàng, sản phẩm thay thế, vân vân mây mây).

**Bài 5: Lược đồ lớp**

|  |  |
| --- | --- |
| **Yêu cầu:** Vẽ lược đồ lớp cho bài toán Quản lý thư viện | |
| **Hình mẫu** | Hình 11: Lược đồ lớp cho bài toán Quản lý thư viện ( mẫu ) |
| **Hình vẽ lại** | Hình 12: Lược đồ lớp cho bài toán Quản lý thư viện ( vẽ lại ) |

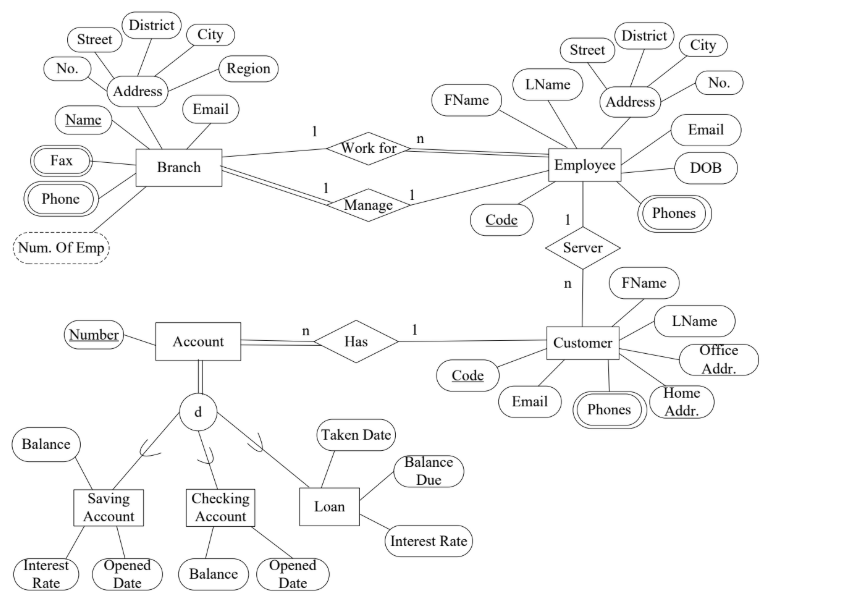
**Bài 6: Lược đồ**

|  |  |
| --- | --- |
| **Yêu cầu:** Vẽ lược đồ sau | |
| **Hình mẫu** | Hình 13: Lược đồ bài 6 ( mẫu ) |
| **Hình vẽ lại** | Hình 14: Lược đồ bài 6( vẽ lại ) |

**Bài 7: Phân tích dữ liệu cho ABC Bank**

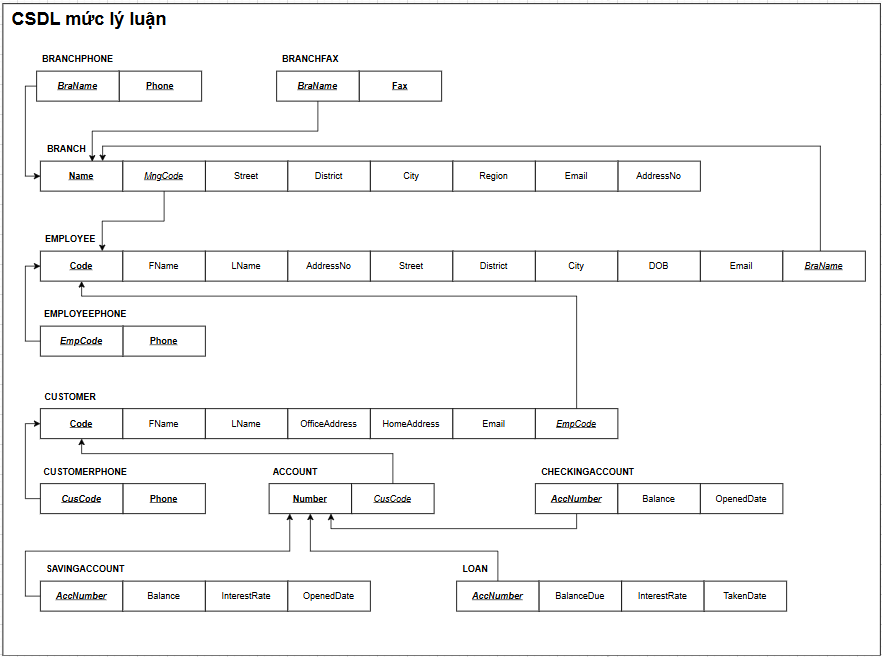
ABC Bank has many branches. Each branch has a unique name, an address (including No., Street, District, City, and Region), phone numbers, fax numbers, email, and total number of employees working there. Each branch has many employees and an employee must work at a branch. There is always one employee 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.

1. **Thiết kế Cơ sở dữ liệu ở mức Khái niệm, Luận lý và Vật lý**
2. **Mức Khái niệm**



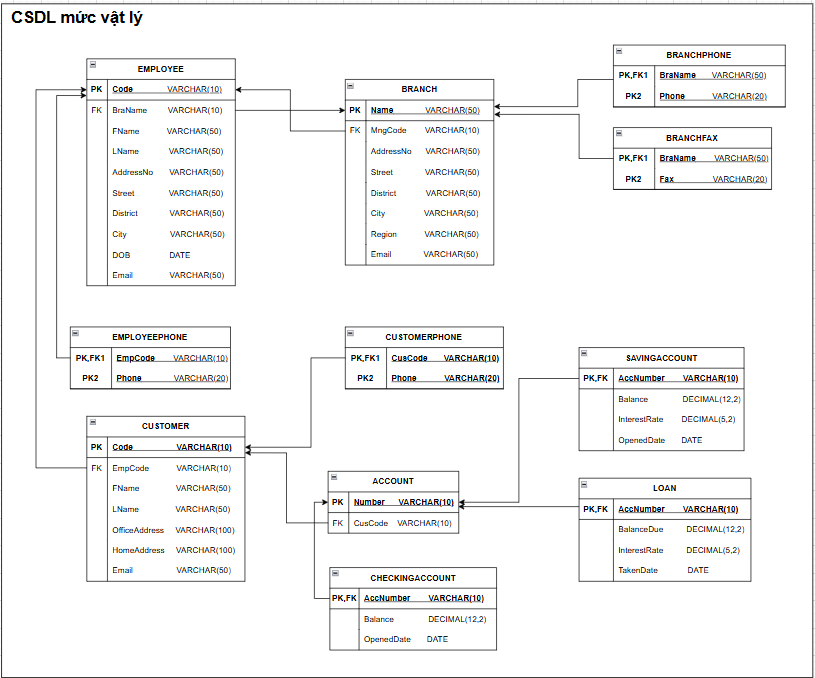
Hình 15: CSDL mức khái niệm

1. **Mức Luận lý**



Hình 16: CSDL mức Luận lý

1. **Mức Vật lý**



Hình 17: CSDL mức Vật lý

1. **Query: Insert, Update, Delete, Select**
2. **You are required to insert valid and meaningful data into the database. Each table has at least 4 rows.**

Bảng BRANCH

INSERT INTO BRANCH VALUES

('B001','123','Tran Hung Dao','Dist1','Hanoi','North','branch001@bank.com','N1001'),

('B002','45','Le Loi','Dist2','Hanoi','North','branch002@bank.com','N1002'),

('B003','12','Nguyen Trai','Dist1','HCM','South','branch003@bank.com','N1003'),

('B004','78','Ly Thuong Kiet','Dist3','Danang','Central','branch04@bank.com','N1080');

Bảng BRANCHPHONE

INSERT INTO BRANCHPHONE VALUES

('B001','0241111111'),

('B002','0242222222'),

('B003','0283333333'),

('B004','0236666666');

Bảng BRANCHFAX

INSERT INTO BRANCHFAX VALUES

('B001','0241112222'),

('B002','0242223333'),

('B003','0283334444'),

('B004','0236667777');

Bảng EMPLOYEE

INSERT INTO EMPLOYEE VALUES

('N1001','Nguyen','An','123','Tran Hung Dao','Dist1','Hanoi','2010-03-15','an@gmail.com','B001'),

('N1002','Tran','Binh','45','Le Loi','Dist2','Hanoi','1988-05-22','binh@gmail.com','B002'),

('N1003','Le','Chi','12','Nguyen Trai','Dist1','HCM','1992-11-10','chi@gmail.com','B003'),

('N1080','Pham','Dung','78','Ly Thuong Kiet','Dist3','Danang','2010-03-20','dung@gmail.com','B004');

Bảng EMPLOYEEPHONE

INSERT INTO EMPLOYEEPHONE VALUES

('N1001','0912345678'),

('N1002','0912345679'),

('N1003','0912345680'),

('N1080','0912345681');

Bảng CUSTOMER

INSERT INTO CUSTOMER VALUES

('C001','Peter','Johnson','123 Office','12 Home','peter@gmail.com','N1001'),

('C002','John','Smith','45 Office','45 Home','john@gmail.com','N1002'),

('C003','Anna','Tran','56 Office','56 Home','anna@gmail.com','N1003'),

('C004','Mary','Pham','78 Office','78 Home','mary@gmail.com','N1080');

Bảng CUSTOMERPHONE

INSERT INTO CUSTOMERPHONE VALUES

('C001','0981111111'),

('C002','0982222222'),

('C003','0983333333'),

('C004','0984444444');

Bảng ACCOUNT

INSERT INTO ACCOUNT VALUES

('A001','C001'),

('A002','C001'),

('A003','C002'),

('A004','C003');

Bảng SAVINGACCOUNT

INSERT INTO SAVINGACCOUNT VALUES

('A001',10000,5.00,'2020-01-10'),

('A002',15000,4.50,'2020-02-15'),

('A003',20000,3.75,'2020-03-20'),

('A004',5000,6.00,'2020-04-12');

Bảng CHECKINGACCOUNT

INSERT INTO CHECKINGACCOUNT VALUES

('A001',5000,'2020-01-10'),

('A002',7000,'2020-02-15'),

('A003',3000,'2020-03-20'),

('A004',8000,'2020-04-12');

Bảng LOAN

INSERT INTO LOAN VALUES

('A001',2000,7.50,'2020-05-01'),

('A002',8000,8.20,'2020-06-12'),

('A003',3000,5.30,'2020-07-14'),

('A004',12000,9.00,'2020-08-20');

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

UPDATE EMPLOYEE

SET LName = 'Brown'

WHERE Code = 'N1080';



Hình 18: Kết quả câu 7b

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



Hình 19: Kết quả câu 7c

Giải thích:

1. Nếu không có ràng buộc gì (NO FOREIGN KEY)

Hệ quản trị sẽ xóa nhân viên N1080 khỏi bảng EMPLOYEE.

Các khách hàng có EmpCode = 'N1080' trong bảng CUSTOMER vẫn còn dữ liệu, nhưng sẽ bị NULL, tức là khách hàng đó không còn gắn với nhân viên nào.

1. Nếu có ràng buộc FOREIGN KEY (CUSTOMER.EmpCode với EMPLOYEE .Code) và không có ON DELETE CASCADE

Hệ quản trị không cho phép xóa nhân viên này, vì còn khách hàng đang được nhân viên N1080 phục vụ.

Bạn sẽ nhận lỗi dạng:

The DELETE statement conflicted with the REFERENCE constraint ...

1. Nếu có ràng buộc FOREIGN KEY kèm ON DELETE CASCADE

Khi xóa nhân viên N1080, tất cả khách hàng (CUSTOMER) có EmpCode = 'N1080' cũng sẽ bị xóa theo.

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

SELECT A.AccNumber

FROM CUSTOMER C

JOIN ACCOUNT A ON C.CustCode = A.CustCode

WHERE C.FName = 'Peter'

AND C.LName = 'Johnson';



Hình 20: Kết quả câu 7d

1. **Find all employees who have their birthdays in March 2010.**

SELECT \*

FROM EMPLOYEE

WHERE YEAR(DOB) = 2010

AND MONTH(DOB) = 3;



Hình 21: Kết quả câu 7e

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

SELECT C.FName, C.LName, SUM(S.Balance) AS TotalBalance

FROM CUSTOMER C

JOIN ACCOUNT A ON C.CustCode = A.CustCode

JOIN SAVINGACCOUNT S ON A.AccNumber = S.AccNumber

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

GROUP BY C.FName, C.LName;



Hình 22: Kết quả câu 7f

**BÀI TẬP ỨNG DỤNG**

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.**

**a) Yêu cầu chức năng**

1. Quản lý sản phẩm:

Người quản trị có thể thêm, sửa, xóa sản phẩm.

Người dùng có thể tìm kiếm, duyệt theo danh mục, xem chi tiết sản phẩm.

1. Giỏ hàng (Shopping Cart)

Người dùng có thể thêm sản phẩm vào giỏ hàng.

Người dùng có thể chỉnh sửa số lượng hoặc xóa sản phẩm trong giỏ hàng.

Giỏ hàng hiển thị tổng tiền.

1. Mua hàng / Thanh toán

Người dùng có thể đặt hàng từ giỏ hàng.

Hệ thống ghi nhận thông tin đơn hàng, địa chỉ giao hàng, phương thức thanh toán.

Người dùng nhận được thông báo xác nhận đơn hàng.

1. Quản lý tài khoản người dùng

Đăng ký, đăng nhập, đăng xuất.

Người dùng chỉnh sửa thông tin cá nhân.

Xem lịch sử mua hàng.

1. Quản lý đơn hàng

Người quản trị xem, cập nhật trạng thái đơn hàng (đang xử lý, đã giao,…).

Người dùng tra cứu tình trạng đơn hàng của mình.

1. Quản lý khuyến mãi / voucher (nếu có)

Người quản trị tạo mã khuyến mãi.

Người dùng nhập mã khuyến mãi khi thanh toán.

**b) Yêu cầu phi chức năng**

1. Hiệu năng: Hệ thống đáp ứng < 3 giây cho thao tác tìm kiếm hoặc thêm vào giỏ hàng.

2. Bảo mật:

Mật khẩu người dùng phải được mã hóa.

Giao dịch thanh toán bảo mật qua HTTPS.

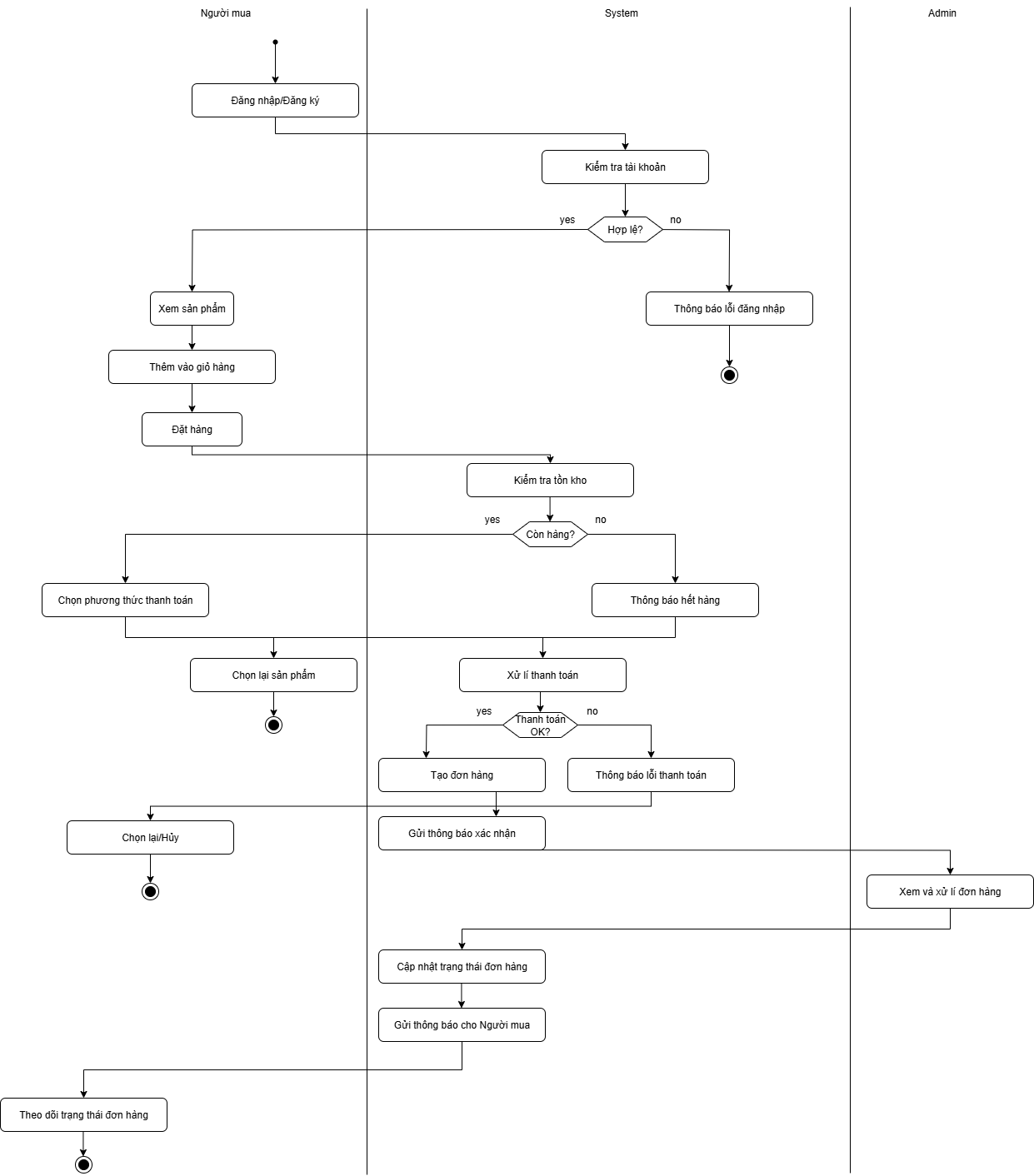
3. Khả dụng: Website hoạt động 24/7 với uptime ≥ 99%.

4. Khả năng mở rộng: Cho phép nhiều người dùng truy cập đồng thời.

5. Dễ sử dụng: Giao diện thân thiện, hỗ trợ cả mobile và desktop.

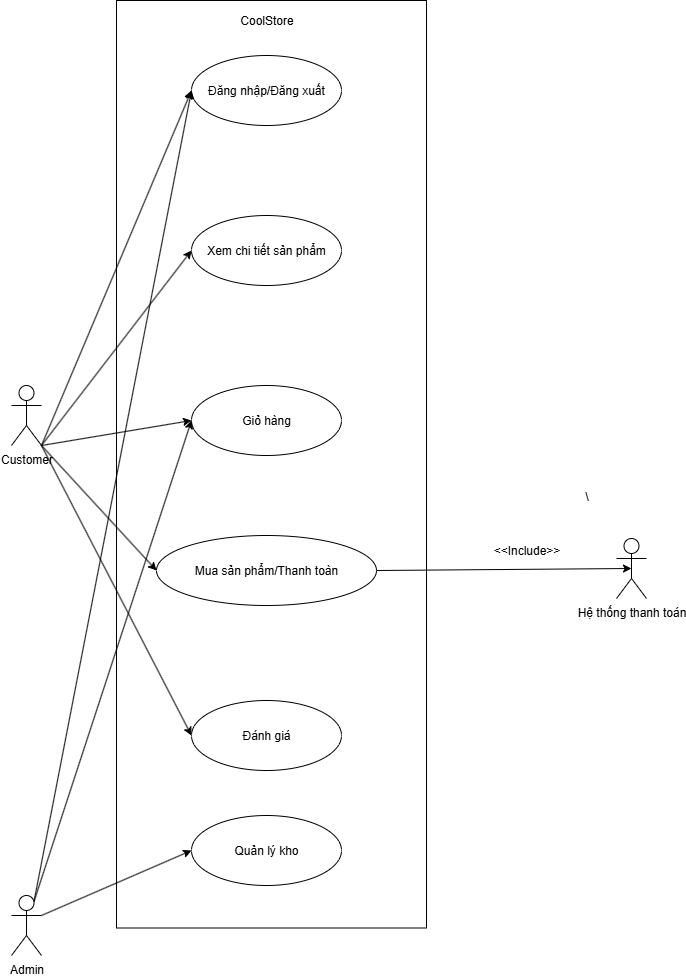
6. Khả năng phục hồi (Reliability & Recovery): Dữ liệu đơn hàng không bị mất khi hệ thống gặp sự cố, có cơ chế backup.

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



Hình 23: Quy trình nghiệp vụ cho website CoolStore

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



Hình 24: Use case diagram cho website CoolStore

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).**

**Tên use case**: Mua sản phẩm từ trang chi tiết sản phẩm

**Tác nhân chính:** Khách hàng

**Mục tiêu:** Khách hàng mua được sản phẩm mong muốn từ trang chi tiết sản phẩm.

**Tiền điều kiện:**

Khách hàng đã đăng nhập hệ thống.

Sản phẩm tồn tại trong kho (còn hàng).

**Luồng chính:**

Khách hàng truy cập trang chi tiết sản phẩm.

Hệ thống hiển thị thông tin sản phẩm (tên, giá, mô tả, tồn kho).

Khách hàng chọn số lượng và nhấn “Mua ngay” hoặc “Thêm vào giỏ hàng”. (1)

Hệ thống thêm sản phẩm vào giỏ hàng và hiển thị giỏ hàng.

Khách hàng nhấn “Thanh toán”.

Hệ thống yêu cầu nhập thông tin giao hàng và chọn phương thức thanh toán.

Khách hàng xác nhận.

Hệ thống gửi yêu cầu đến cổng thanh toán và nhận phản hồi thành công. (2)

Hệ thống tạo đơn hàng và thông báo “Đặt hàng thành công”.

**Luồng thay thế:**

1. Nếu sản phẩm hết hàng thì Hệ thống báo “Sản phẩm tạm hết hàng”.

2. Nếu thanh toán thất bại thì Hiển thị thông báo lỗi, cho phép khách hàng chọn lại phương thức.

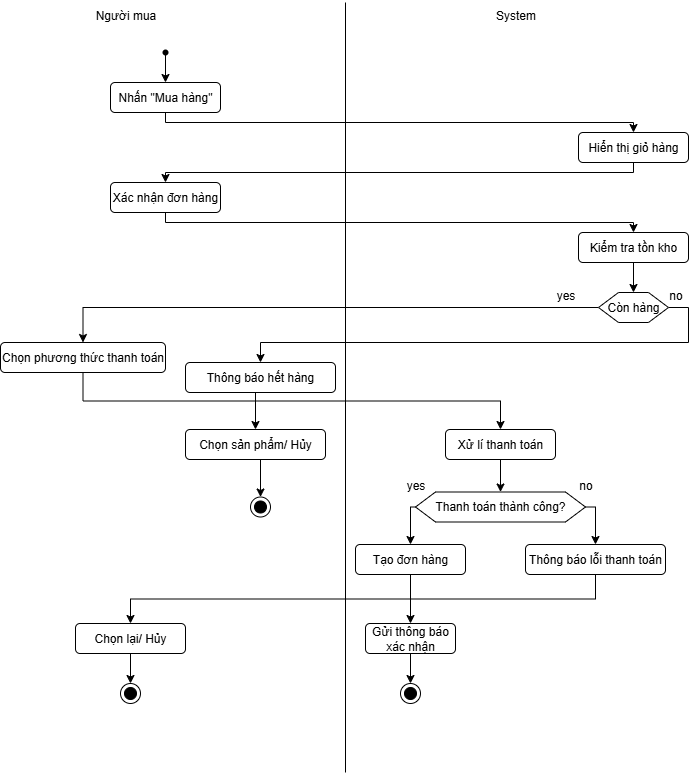
**Hậu điều kiện:**

Đơn hàng được tạo trong hệ thống.

Giỏ hàng cập nhật trạng thái rỗng hoặc đã bớt sản phẩm vừa mua.

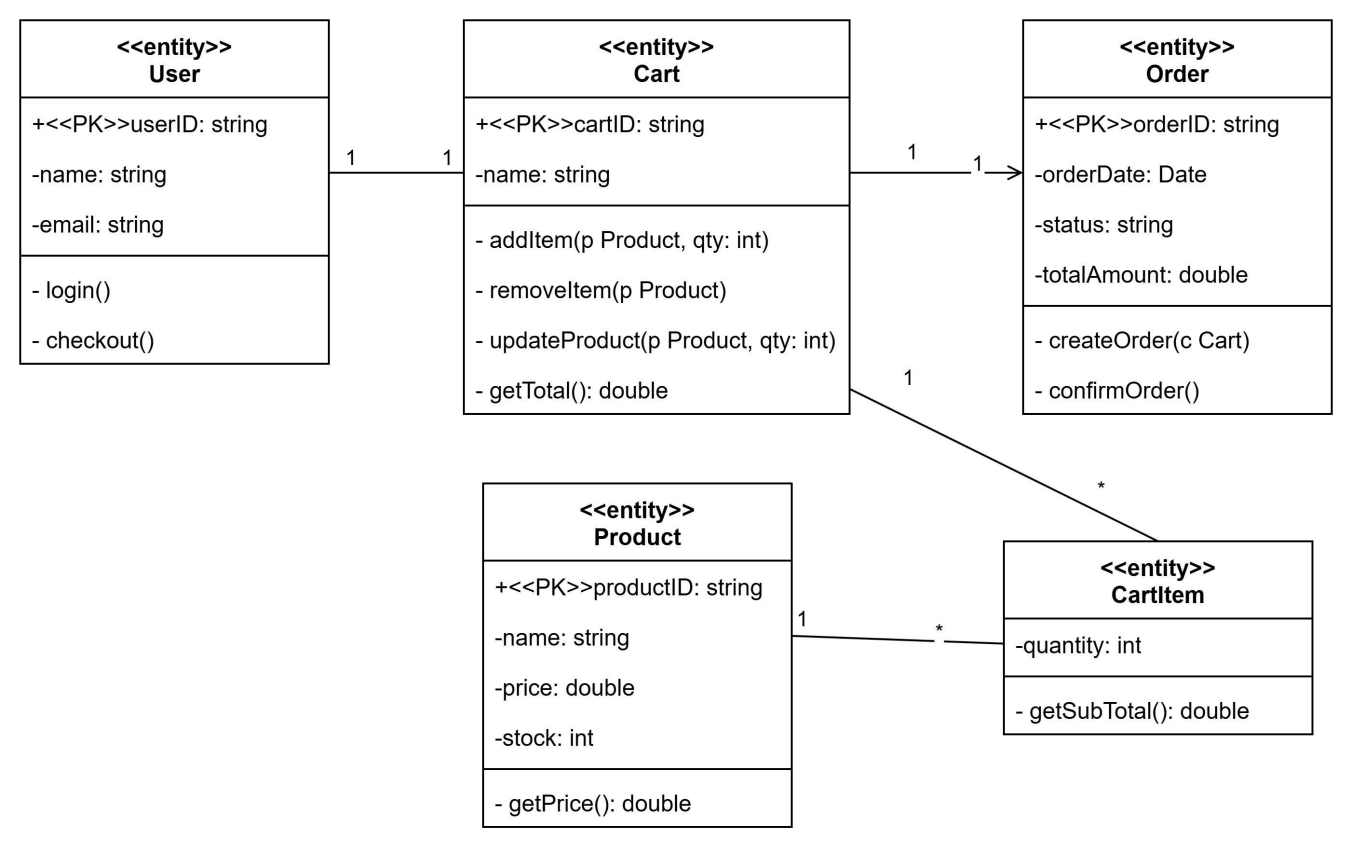
Ghi chú: Có thể lưu lịch sử đơn hàng để khách hàng tra cứu sau.

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.**



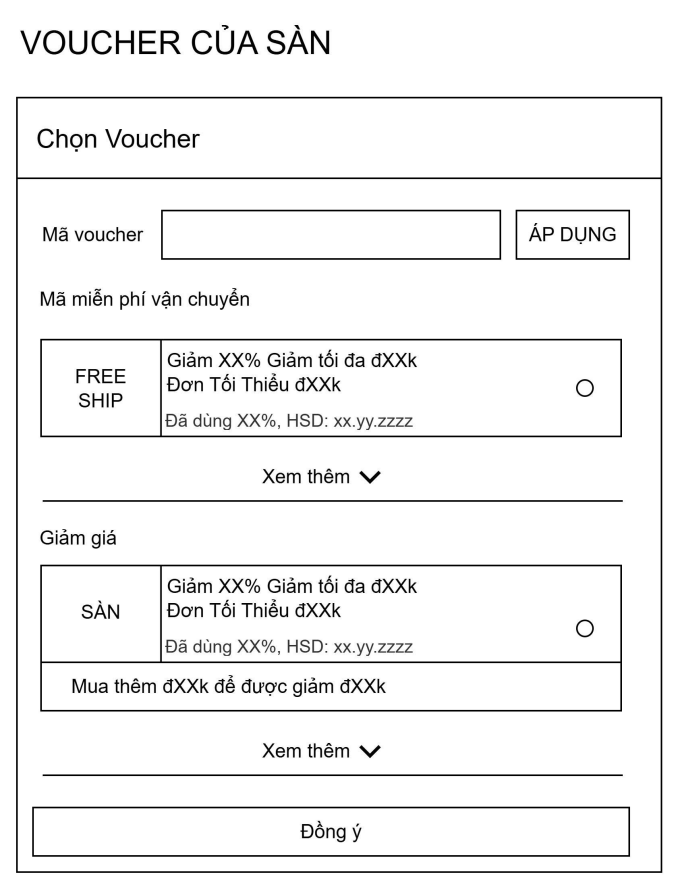
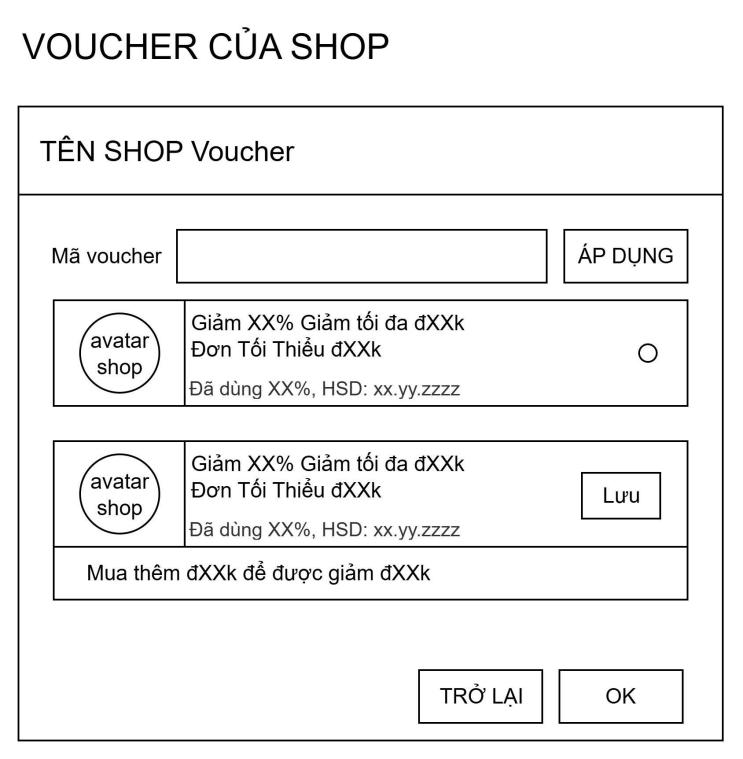
Hình 25: sơ đồ Activity Diagram

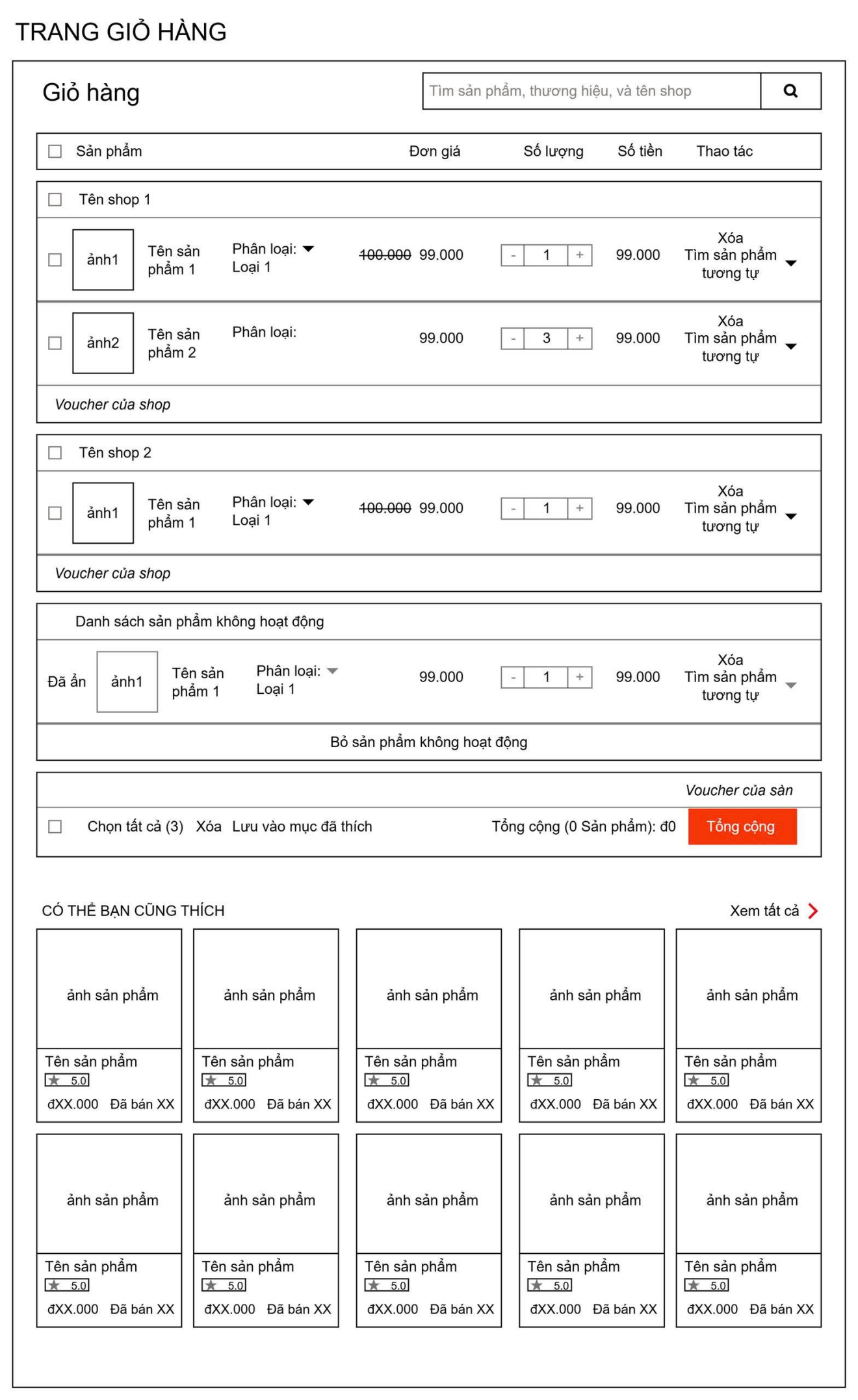
1. **Thiết kế sơ đồ Class Diagram cho module Giỏ hàng.**

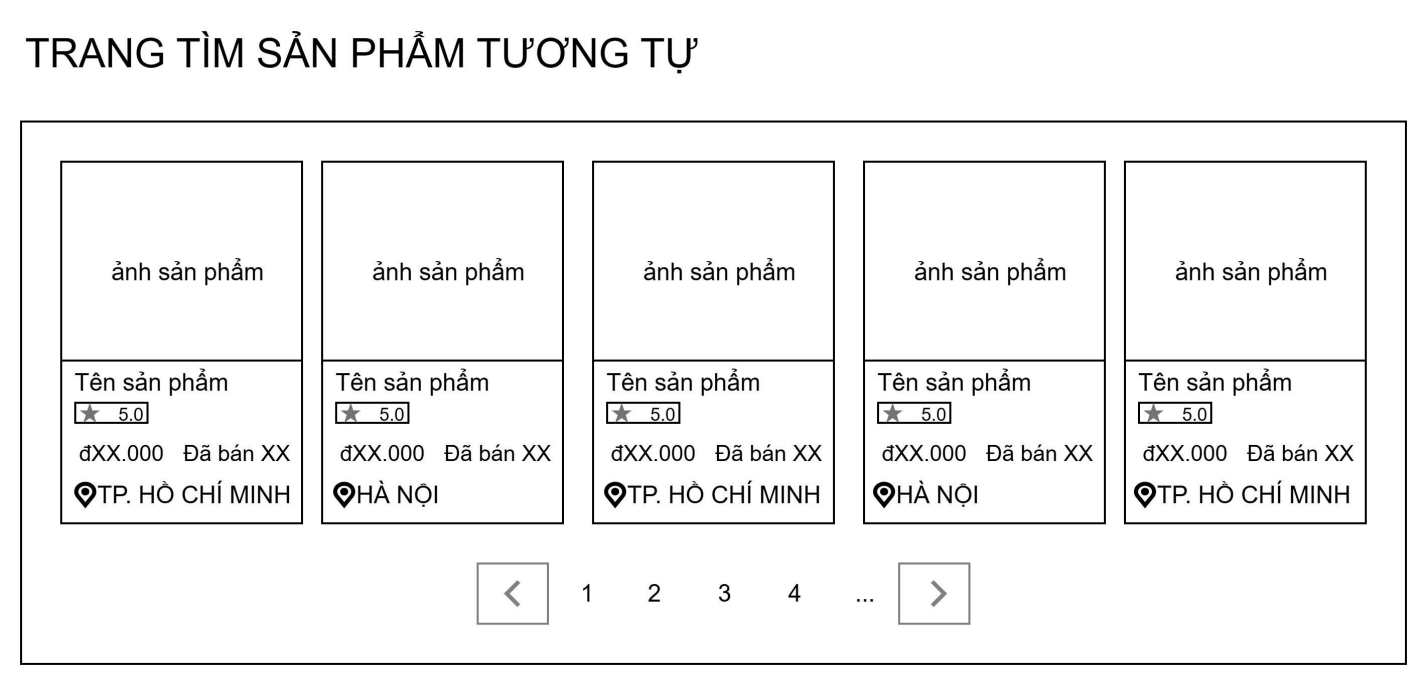


Hình 26: Sơ đồ Class Diagram

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

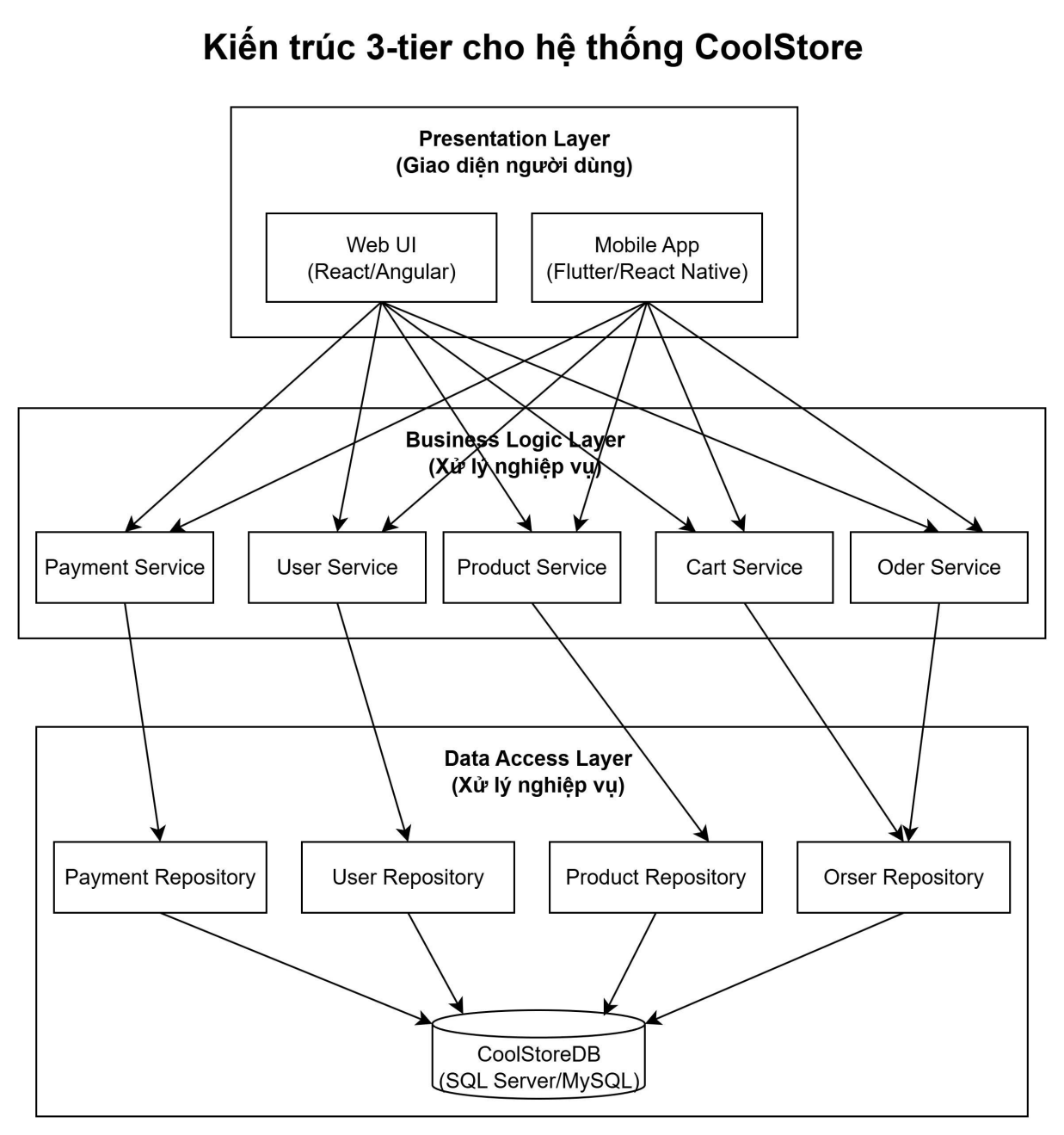






Hình 27: Giao diện màn hình cho chức năng “Quản lý giỏ hàng”

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



Hình 28: Kiến trúc 3-tier cho hệ thống CoolStore

Đề xuất kiến trúc cho hệ thống CoolStore

a) Presentation Layer

Các thành phần: Web UI, Mobile App (có thể dùng React, Angular, hoặc Flutter).

Chức năng chính:

Người mua đăng nhập, duyệt sản phẩm, giỏ hàng, thanh toán.

Admin quản lý sản phẩm, đơn hàng.

Tương tác với Business Layer qua API (REST/GraphQL).

b) Business Logic Layer

Các module nghiệp vụ:

User Service: đăng nhập, đăng ký, phân quyền.

Product Service: quản lý sản phẩm, tồn kho.

Cart Service: xử lý giỏ hàng, tính toán tổng tiền.

Order Service: tạo và quản lý đơn hàng.

Payment Service: xác thực và xử lý thanh toán.

Có thể tổ chức dưới dạng dịch vụ riêng (microservices) hoặc module trong 1 backend.

Công nghệ: Java Spring Boot, .NET Core, Node.js/Express.

c) Data Access Layer

Quản lý CSDL:

Bảng Users, Products, Cart, Orders, Payments.

Dùng ORM (Entity Framework, Hibernate, Sequelize) để truy xuất.

Có thể triển khai với SQL Server, MySQL, PostgreSQL.

Cung cấp API cho Business Layer, không cho Presentation truy cập trực tiếp.

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

ID: TC\_AddToCart\_01

Mô tả: Người dùng thêm một sản phẩm có sẵn trong kho vào giỏ hàng.

Dữ liệu đầu vào:

* User đã đăng nhập.
* ProductID = P001, Stock = 10, Quantity = 2.

Bước thực hiện:

1. Người dùng chọn sản phẩm P001.
2. Nhập số lượng = 2.
3. Nhấn nút “Thêm vào giỏ hàng”.

Kết quả mong đợi:

* Giỏ hàng hiển thị sản phẩm P001 với số lượng = 2.
* Tổng tiền được tính chính xác.
* Thông báo: “Thêm sản phẩm thành công.”

Test Case 2: Thêm số lượng vượt quá tồn kho

ID: TC\_AddToCart\_02

Mô tả: Người dùng thêm sản phẩm với số lượng lớn hơn tồn kho.

Dữ liệu đầu vào:

* User đã đăng nhập.
* ProductID = P002, Stock = 5, Quantity = 10.

Bước thực hiện:

1. Người dùng chọn sản phẩm P002.
2. Nhập số lượng = 10.
3. Nhấn nút “Thêm vào giỏ hàng”.

Kết quả mong đợi:

* Hệ thống không cho phép thêm.
* Hiển thị thông báo: “Số lượng vượt quá tồn kho. Chỉ còn 5 sản phẩm.”
* Giỏ hàng không thay đổi.

Test Case 3: Thêm sản phẩm chưa đăng nhập

ID: TC\_AddToCart\_03

Mô tả: Người dùng chưa đăng nhập nhưng muốn thêm sản phẩm.

Dữ liệu đầu vào:

* User chưa đăng nhập.
* ProductID = P003, Quantity = 1.

Bước thực hiện:

1. Người dùng chọn sản phẩm P003.
2. Nhấn “Thêm vào giỏ hàng”.

Kết quả mong đợi:

* Hệ thống yêu cầu đăng nhập hoặc cho phép lưu giỏ hàng tạm (guest cart).
* Thông báo: “Vui lòng đăng nhập để tiếp tục.”
* Sản phẩm chưa được ghi vào giỏ hàng chính thức.