| **TRƯỜNG ĐẠI HỌC SÀI GÒN**  **KHOA CÔNG NGHỆ THÔNG TIN**  **A blue circle with text  Description automatically generated**  **Kiểm Thử Phần Mềm - Lab 2**  **Tên sinh viên: - Phạm Nguyễn Viết Vinh  3122411249  - Võ Minh Thư   3122411202  - Đỗ Đức Kỳ Vỹ 3122411258**  **Giảng viên hướng dẫn: TS. Đỗ Như Tài** |
| --- |

Mục Lục

[**Bảng phân công 3**](#_bfu3fp9fuswz)

[**Bài tập minh họa 4**](#_k4gq3y5tqqj9)

[Bài 1: 4](#_993zdwrwrk93)

[Bài 2: 5](#_45uqwlhq6398)

[Bài 3: 6](#_dr8gge3jlk60)

[Bài 4: 9](#_e6djsbl5hk5r)

[Bài 5: 10](#_od2dg1r2w0ud)

[Bài 6: 12](#_pvb7ekgcnzfs)

[Bài 7: 13](#_sauce6vymzn4)

[**Bài tập ứng dụng 17**](#_w3rwnvqwrn2x)

[Câu 1: 17](#_ksm0847ick9j)

[Câu 2: 19](#_isdejf8ptl60)

[Câu 3: 20](#_oke61d68pzlp)

[Câu 4: 21](#_11kc445gpe3i)

[Câu 5: 23](#_du6jtmaf9wop)

[Câu 6: 25](#_jbi948crbvbm)

[Câu 7: 26](#_cl4ynjc207zb)

[Câu 8: 26](#_yla04o17gf9z)

[Câu 9: 28](#_j5c78x15xzod)

# Bảng phân công

| Họ Tên | MSSV | Công việc |
| --- | --- | --- |
| Phạm Nguyễn Viết Vinh | 3122411249 | Bài 1, 2, 3 bài tập minh họa và câu 4, 5, 6 bài tập ứng dụng |
| Võ Minh Thư | 3122411202 | Bài 4 , 5 bài tập minh họa và câu 1, 2 , 3 bài tập ứng dụng |
| Đỗ Đức Kỳ Vỹ | 3122411258 | Bài 6, 7 bài tập minh họa và câu 7, 8, 9 bài tập ứng dụng |

# Bài tập minh họa

## Bài 1:

## Bài 2:

## Bài 3:

a) Các tác nhân (Actors):

Trong sơ đồ có 3 tác nhân chính và 1 hệ thống ngoài:

1. **Student** (Sinh viên)

* Đăng ký môn học cho học kỳ mới.
* Liên kết với hệ thống thanh toán (Billing System) để xử lý học phí.

1. **Professor** (Giảng viên)

Chọn môn học mà mình sẽ dạy trong kỳ học.

1. **Registrar** (Phòng đào tạo / Quản trị viên hệ thống)

* Duy trì chương trình học (Curriculum).
* Quản lý thông tin sinh viên và giảng viên.

1. **Billing System** (Hệ thống thanh toán – hệ thống ngoài)

* Nhận thông tin đăng ký môn học để xử lý thanh toán học phí.

b) Các Use Cases

Các trường hợp sử dụng trong sơ đồ:

1. Register for Courses

* Sinh viên đăng ký các môn học mong muốn.
* Gửi thông tin đến hệ thống thanh toán để tính học phí.

1. Select Courses to Teach

* Giảng viên chọn các môn học mà họ sẽ phụ trách giảng dạy trong kỳ.

1. Maintain Curriculum

* Registrar cập nhật, thêm, xóa các môn học trong chương trình đào tạo.

1. Maintain Student Information

* Registrar thêm mới, cập nhật thông tin, hoặc xóa hồ sơ sinh viên.
* Bao gồm thông tin cá nhân, tình trạng học tập, lớp học, điểm số, v.v.

1. Maintain Professor Information

* Registrar quản lý thông tin giảng viên (thêm mới, cập nhật, nghỉ việc).

1. Registrar Validation

* Use case phụ trợ (được gọi bởi các use case khác qua quan hệ «uses»).
* Kiểm tra thông tin mà registrar nhập vào trước khi lưu (vd: dữ liệu hợp lệ, không trùng lặp)

c) Phân tích các mối quan hệ

1. **Quan hệ giữa Actors và Use Cases:**

* Student → Register for Courses → Billing System
* Professor → Select Courses to Teach
* Registrar → Maintain Curriculum / Maintain Student Information / Maintain Professor Information

1. **Quan hệ giữa các Use Cases:**

* Maintain Curriculum «uses» Registrar Validation
* Maintain Student Information «uses» Registrar Validation
* Maintain Professor Information «uses» Registrar Validation

Nghĩa là trước khi dữ liệu được lưu, hệ thống sẽ thực hiện kiểm tra hợp lệ (validation).

d) Kịch bản chi tiết cho Use Case “Register for Courses”

**Tên Use Case**: Register for Courses  
 **Tác nhân chính**: Student  
 **Mục tiêu**: Sinh viên đăng ký thành công các môn học mong muốn.

**Tiền điều kiện** (Preconditions):

* Sinh viên đã đăng nhập hệ thống và có tài khoản hợp lệ.
* Danh sách môn học được mở đăng ký.

**Luồng chính** (Basic Flow):

1. Sinh viên mở chức năng đăng ký môn học.
2. Hệ thống hiển thị danh sách các môn học có thể đăng ký.
3. Sinh viên chọn các môn học muốn đăng ký.
4. Hệ thống kiểm tra điều kiện tiên quyết (prerequisite) và số tín chỉ tối đa.
5. Nếu hợp lệ, hệ thống ghi nhận các môn học đã đăng ký.
6. Hệ thống gửi thông tin đăng ký đến Billing System để tạo hóa đơn học phí.
7. Billing System phản hồi kết quả xử lý.
8. Hệ thống thông báo sinh viên đăng ký thành công.

**Luồng ngoại lệ** (Alternative/Exception Paths):

* A1: Môn học đã đầy → Hệ thống thông báo lỗi, yêu cầu chọn môn khác.
* A2: Sinh viên vượt quá số tín chỉ cho phép → Hệ thống yêu cầu giảm số môn.
* A3: Billing System không phản hồi → Hệ thống lưu tạm đăng ký và thông báo sẽ xử lý sau.

**Hậu điều kiện** (Postconditions):

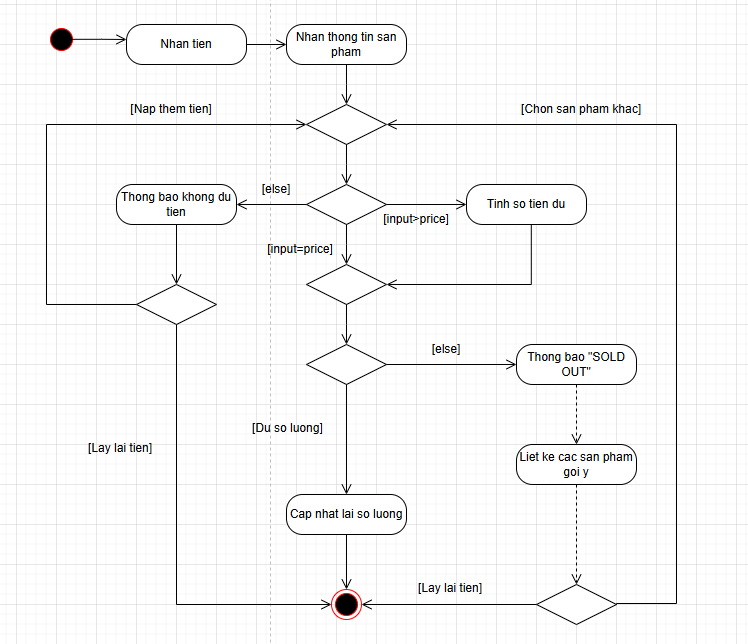
* Đăng ký được lưu vào cơ sở dữ liệu.
* Sinh viên có thể xem lịch học đã đăng ký.
* Hóa đơn học phí được tạo.

e) Đề xuất cải tiến hoặc bổ sung

1. **Use Case “Drop/Withdraw Course”** – cho phép sinh viên hủy đăng ký môn học.
2. **Use Case “View Registration History”** – sinh viên xem lại các môn đã đăng ký trong quá khứ.
3. **Use Case “Generate Reports”** – Registrar tạo báo cáo thống kê số lượng sinh viên/môn học.
4. **Use Case “Assign Professors to Courses”** – Registrar phân công giảng viên dạy môn học.

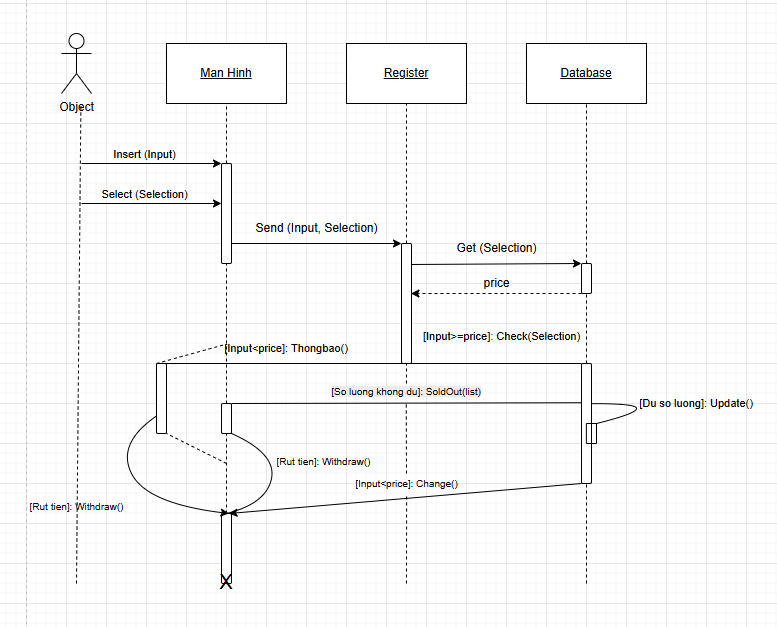
Những bổ sung này giúp hệ thống quản lý học vụ hoàn thiện hơn và hỗ trợ các tình huống thực tế.

## Bài 4:



Activity Diagram (sơ đồ hoạt động): mô tả luồng xử lý khi người dùng nạp tiền, chọn sản phẩm, kiểm tra số tiền và số lượng hàng, sau đó hoặc nhận thông báo lỗi / SOLD OUT hoặc cập nhật số lượng và hoàn tất giao dịch.

* Ý nghĩa: Đây là sơ đồ nghiệp vụ, cho thấy toàn bộ các bước xử lý và điều kiện rẽ nhánh trong quy trình mua hàng (như ở máy bán hàng tự động).



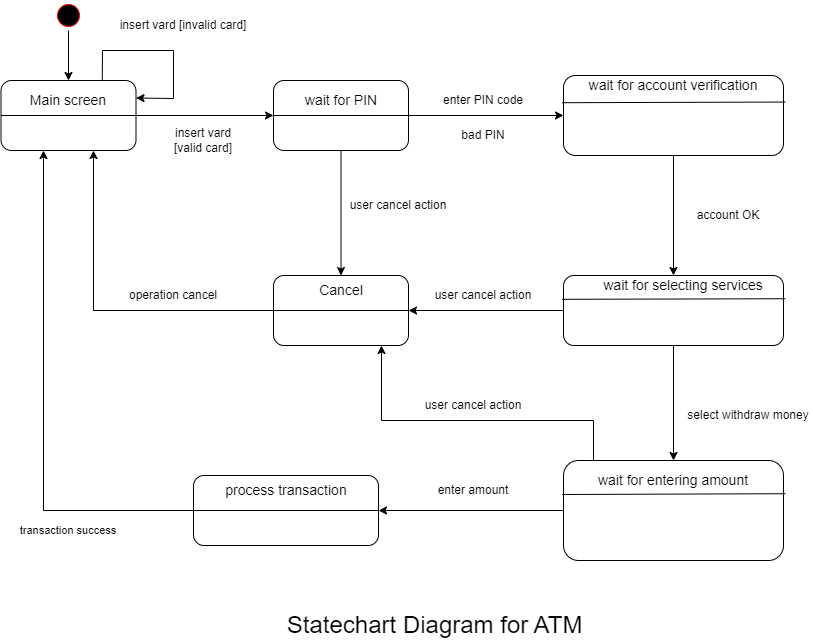
Sequence Diagram (sơ đồ tuần tự): mô tả sự tương tác giữa các đối tượng (Màn hình, Register, Database) theo thời gian.

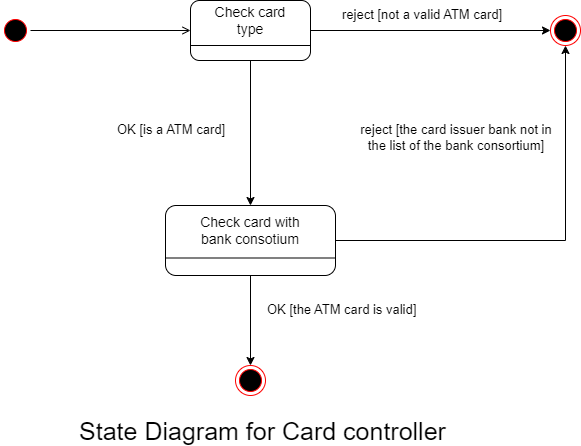
* Ý nghĩa: Cho thấy cách các thành phần trao đổi thông điệp (message) để hoàn tất giao dịch: nhập dữ liệu, kiểm tra, thông báo, cập nhật số lượng, trả lại tiền.

## Bài 5:

|  |
| --- |

## Bài 6:





## Bài 7:

**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)

1. Chèn dữ liệu hợp lý vào mỗi bảng (ít nhất 4 hàng)

* Bảng **BRANCH**

**INSERT INTO BRANCH VALUES**

('BranchA', '12', 'Nguyen Hue', 'District 1', 'HCM', 'South', 'branchA@abc.com', 'E001'),

('BranchB', '45', 'Le Loi', 'District 3', 'HCM', 'South', 'branchB@abc.com', 'E002'),

('BranchC', '78', 'Tran Hung Dao', 'District 5', 'HCM', 'South', 'branchC@abc.com', 'E003'),

('BranchD', '90', 'Vo Van Kiet', 'District 10', 'HCM', 'South', 'branchD@abc.com', 'E004');

* Bảng **BRANCHPHONE**

**INSERT INTO BRANCHPHONE VALUES**

('BranchA', '02812345678'),

('BranchB', '02823456789'),

('BranchC', '02834567890'),

('BranchD', '02845678901');

* Bảng **BRANCHFAX**

**INSERT INTO BRANCHFAX VALUES**

('BranchA', '02887654321'),

('BranchB', '02876543210'),

('BranchC', '02865432109'),

('BranchD', '02854321098');

* Bảng **EMPLOYEE**

**INSERT INTO EMPLOYEE VALUES**

('E001', 'John', 'Smith', '101', 'Ly Tu Trong', 'District 1', 'HCM', '1980-03-15', 'john.smith@abc.com', 'BranchA'),

('E002', 'Anna', 'Brown', '202', 'Pasteur', 'District 3', 'HCM', '1985-07-20', 'anna.brown@abc.com', 'BranchB'),

('E003', 'David', 'Tran', '303', 'CMT8', 'District 5', 'HCM', '1990-12-01', 'david.tran@abc.com', 'BranchC'),

('E004', 'Linda', 'Nguyen', '404', 'Nguyen Trai', 'District 10', 'HCM', '1992-05-10', 'linda.nguyen@abc.com', 'BranchD');

* Bảng **EMPLOYEEPHONE**

**INSERT INTO EMPLOYEEPHONE VALUES**

('E001', '0901234567'),

('E002', '0902345678'),

('E003', '0903456789'),

('E004', '0904567890');

* Bảng **CUSTOMER**

**INSERT INTO CUSTOMER VALUES**

('C001', 'Peter', 'Johnson', '12 Nguyen Hue, District 1, HCM', '101 Le Lai, District 1, HCM', 'peter.johnson@gmail.com', 'E001'),

('C002', 'Mary', 'Pham', '45 Le Loi, District 3, HCM', '202 Tran Hung Dao, District 3, HCM', 'mary.pham@gmail.com', 'E002'),

('C003', 'Tom', 'Nguyen', '78 CMT8, District 5, HCM', '303 Vo Van Kiet, District 5, HCM', 'tom.nguyen@gmail.com', 'E003'),

('C004', 'Lisa', 'Tran', '90 Nguyen Trai, District 10, HCM', '404 Pasteur, District 10, HCM', 'lisa.tran@gmail.com', 'E004');

* Bảng **CUSTOMERPHONE**

**INSERT INTO CUSTOMERPHONE VALUES**

('C001', '0911234567'),

('C002', '0912345678'),

('C003', '0913456789'),

('C004', '0914567890');

* Bảng **ACCOUNT**

**INSERT INTO ACCOUNT VALUES**

('A1001', 'C001'),

('A1002', 'C001'),

('A1003', 'C002'),

('A1004', 'C003');

* Bảng **SAVINGACCOUNT**

**INSERT INTO SAVINGACCOUNT VALUES**

('A1001', 5000000, 0.05, '2023-01-10'),

('A1003', 3000000, 0.04, '2023-03-15'),

('A1004', 7000000, 0.045, '2023-06-20'),

('A1002', 2000000, 0.03, '2023-07-01');

* Bảng **CHECKINGACCOUNT**

**INSERT INTO CHECKINGACCOUNT VALUES**

('A1003', 1500000, '2023-04-01'),

('A1004', 2500000, '2023-05-10'),

('A1002', 1000000, '2023-08-01'),

('A1001', 500000, '2023-09-01');

* Bảng **LOAN**

**INSERT INTO LOAN VALUES**

('A1001', 10000000, 0.06, '2023-02-01'),

('A1002', 8000000, 0.055, '2023-03-01'),

('A1003', 12000000, 0.05, '2023-04-01'),

('A1004', 9000000, 0.045, '2023-05-01');

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

**SELECT \* FROM EMPLYOEE**

**UPDATE EMPLOYEE**

**SET LName = “Brown”**

**WHERE Code = “N1080”**

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

**DELETE FROM EMPLOYEE**

**WHERE Code = 'N1080';**

Không thể xóa Employee với code = “N1080” vì có ràng buộc khóa ngoại giữa employee và customer, nếu xóa Employee có code bằng N1080 thì sẽ xảy ra lỗi

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

**SELECT \***

**FROM ACCOUNT A**

**JOIN CUSTOMER C ON A.CusCode = C.Code**

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

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

**SELECT \***

**FROM EMPLOYEE**

**WHERE MONTH(DOB) = 3 AND YEAR(DOB) = 2010;**

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

**SELECT \***

**FROM SAVINGACCOUNT S**

**JOIN ACCOUNT A ON S.AccNumber = A.Number**

**JOIN CUSTOMER C ON A.CusCode = C.Code**

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

# Bài tập ứng dụng

## 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.

\* Yêu cầu chức năng

a) Danh mục sản phẩm (Product Catalog)

* Người mua (Buyer) có thể:
  + Duyệt danh sách sản phẩm.
  + Lọc và sắp xếp sản phẩm theo tên, giá.
  + Nhấp vào sản phẩm để xem chi tiết.
  + Xem thông tin chi tiết gồm: tên, mô tả, số lượng tồn kho, thông tin kho (địa chỉ), trạng thái sản phẩm nổi bật, đánh giá.
* Quản trị viên (SysAdmin) có thể:
  + Quản lý sản phẩm (thêm, sửa, xóa).
  + Gán sản phẩm vào kho hiện có.

b) Giỏ hàng (Shopping Cart)

* Người mua có thể:
  + Thêm sản phẩm vào giỏ từ trang danh sách hoặc trang chi tiết.
  + Xem giỏ hàng với các thông tin tổng hợp: tổng chi phí, chi phí sau khuyến mãi, chi phí vận chuyển, khuyến mãi tiết kiệm, tổng giá trị đơn hàng.
  + Xóa sản phẩm khỏi giỏ.
  + Thấy giỏ hàng và bảng tổng hợp được cập nhật ngay khi thay đổi.
  + Thực hiện thanh toán qua nút Thanh toán.
* Quản trị viên có thể:
  + Xem toàn bộ giỏ hàng của người dùng.
  + Kích hoạt hoặc vô hiệu hóa giỏ hàng không hợp lệ.

c) Quy trình thanh toán (Payment Process)

* Hệ thống phải:
  + Xác thực thông tin sản phẩm trong giỏ.
  + Xử lý thanh toán.
  + Gửi email thông báo trạng thái giao dịch cho người mua.

d) Tồn kho (Inventory)

* Quản trị viên có thể quản lý kho hàng (tạo, chỉnh sửa, xóa).
* Sản phẩm gắn với kho để quản lý tồn kho.

e) Đánh giá sản phẩm (Rating)

* Người mua có thể đánh giá sản phẩm (thang điểm 1–5 sao).
* Hiển thị đánh giá trên trang chi tiết sản phẩm.

f) Kiểm soát truy cập (Access Control)

* Người dùng có thể đăng nhập/đăng xuất hệ thống.
* Hệ thống phân quyền:
  + Buyer → chuyển đến trang danh mục sản phẩm.
  + Admin → chuyển đến trang quản trị.

g) Thiết lập ban đầu (Initialization)

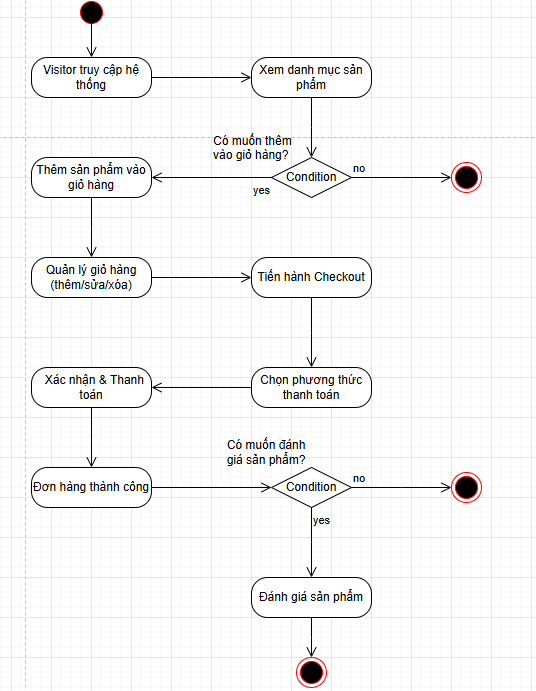
* Khi hệ thống khởi động:
  + Tạo tài khoản quản trị viên.
  + Tạo sẵn hai tài khoản Buyer.
  + Tạo dữ liệu mẫu cho sản phẩm, kho, đánh giá.

\* Yêu cầu phi chức năng

* Hiệu năng (Performance):
  + Danh sách sản phẩm phải tải nhanh, kể cả khi có nhiều sản phẩm.
  + Giỏ hàng và bảng tổng hợp phải cập nhật theo thời gian thực (hoặc gần như ngay lập tức).
* Tính bảo mật (Security):
  + Cơ chế đăng nhập an toàn (mật khẩu mã hóa).
  + Phân quyền rõ ràng giữa Buyer và Admin.
  + Chỉ Admin được phép quản lý sản phẩm, kho, giỏ hàng người khác.
* Khả dụng (Availability & Reliability):
  + Hệ thống luôn sẵn sàng cho người dùng truy cập.
  + Thông tin giỏ hàng, kho, sản phẩm không bị mất khi hệ thống gặp sự cố.
* Tính mở rộng (Scalability):
  + Hỗ trợ thêm nhiều sản phẩm, kho, và người dùng mà không làm giảm hiệu năng.
* Khả năng sử dụng (Usability):
  + Giao diện thân thiện, dễ tìm kiếm, dễ thao tác cho Buyer.
  + Quản trị viên có trang quản trị riêng, đơn giản để thao tác.
* Tính toàn vẹn dữ liệu (Data Integrity):
  + Đảm bảo dữ liệu tồn kho và đơn hàng nhất quán (vd: không bán sản phẩm hết hàng).
  + Đánh giá của Buyer gắn đúng sản phẩm.
* Khả năng bảo trì (Maintainability):
  + Code và hệ thống dễ mở rộng, bảo trì.
  + Cho phép thêm module mới (ví dụ: voucher, vận chuyển nâng cao).
* Khả năng kiểm thử (Testability):
  + Có thể mô phỏng dễ dàng các kịch bản nghiệp vụ (thêm sản phẩm, thanh toán, quản lý kho).

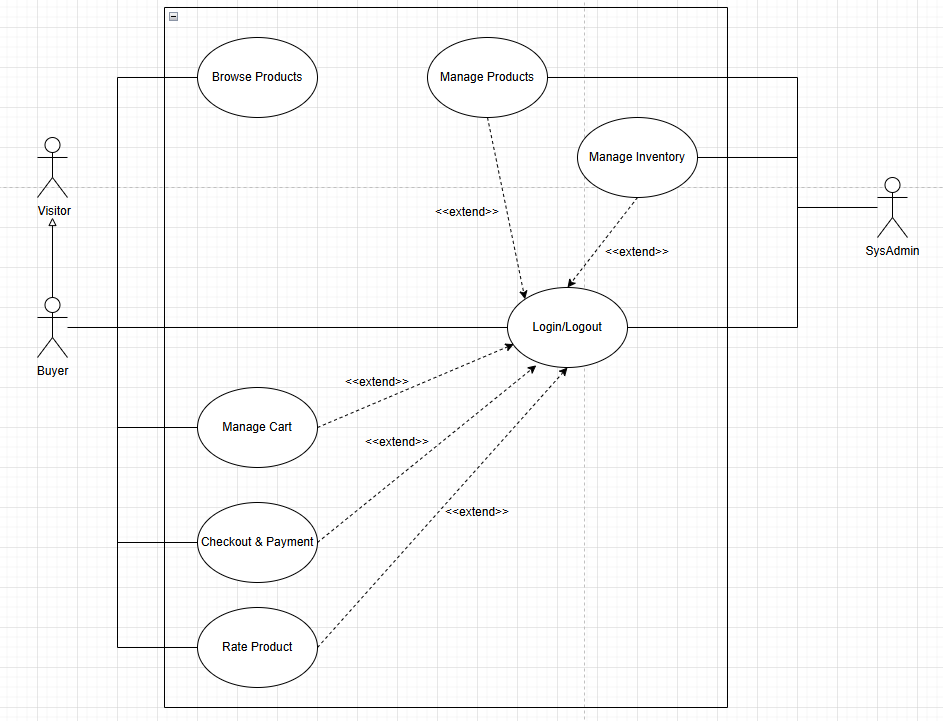
## Câu 2:

Vẽ qui trình nghiệp vụ cho website CoolStore



## Câu 3:

Xây dựng use case diagram cho website CoolStore.



## Câu 4:

Mô tả usecase “Mua sản phẩm từ trang chi tiết sản phẩm”

|  | | |
| --- | --- | --- |
| **Tên Usecase** | Mua sản phẩm từ trang chi tiết sản phẩm | |
| **Tác nhân chính** | Người mua (Buyer) | |
| **Mục tiêu** | Người mua thêm sản phẩm vào giỏ hàng trực tiếp từ trang chi tiết sản phẩm để chuẩn bị thanh toán | |
| **Tiền điều kiện** | -Người mua đã đăng nhập hệ thống  -Sản phẩm được hiển thị trên trang chi tiết sản phẩm. | |
| **Luồng chính** | **Tác nhân thực hiện** | **Hệ thống phản hồi** |
| 1. Người mua mở trang chi tiết của sản phẩm cần mua |  |
|  | 2. Hệ thống hiển thị đầy đủ thông tin sản phẩm (tên, mô tả, tồn kho, địa chỉ kho, trạng thái nổi bật, đánh giá). |
| 3. Người mua chọn số lượng sản phẩm cần mua |  |
| 4. Người mua nhấn nút "Mua hàng" |  |
|  | 5. Hệ thống kiểm tra số lượng tồn kho. **E1** |
| 6. Nếu còn hàng, hệ thống thêm sản phẩm với số lượng đã chọn vào giỏ hàng. |  |
|  | 7. Hệ thống cập nhật giỏ hàng và hiển thị thông báo “Sản phẩm đã được thêm vào giỏ hàng”. |
| Người mua có thể tiếp tục mua sắm hoặc chuyển sang trang giỏ hàng để thanh toán. |  |
| **Luồng thay thế** | Không có | |
| **Luồng ngoại lệ** | **E1**. Ở bước 5 nếu hệ thống phát hiện số lượng tồn kho < số lượng người mua yêu cầu. | |
|  | Hệ thống hiển thị thông báo lỗi: “Sản phẩm hiện không đủ số lượng trong kho.” |
| Người mua có thể chọn lại số lượng hoặc quay lại trang danh sách sản phẩm |  |
| Hậu điều kiện | -Sản phẩm đã được thêm vào giỏ hàng của người mua (nếu còn hàng).  -Giỏ hàng được cập nhật với tổng giá trị đơn hàng mới. | |
| Ghi chú | -Hệ thống cần đồng bộ tồn kho theo thời gian thực để tránh tình trạng người mua đặt hàng quá số lượng tồn kho thực tế.  -Có thể kết hợp hiển thị gợi ý “Sản phẩm tương tự” để tăng doanh thu. | |

## 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.

Activity Diagram: Quy trình mua hàng

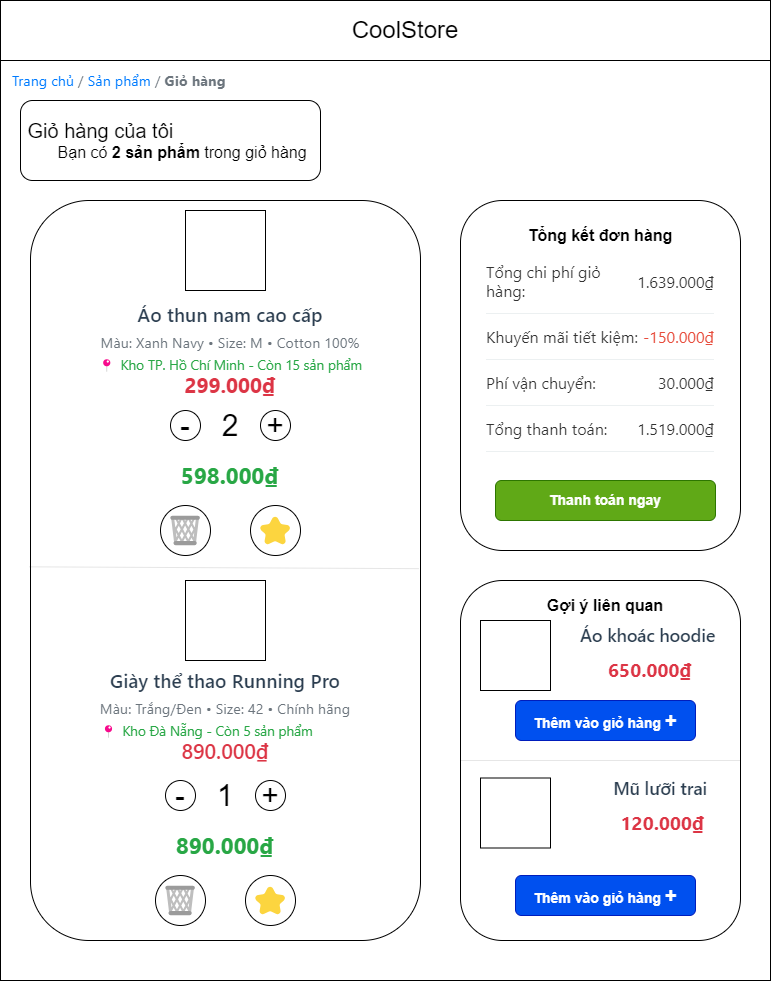
|  |
| --- |

## Câu 6:

Thiết kế sơ đồ Class Diagram cho module Giỏ hàng.

|  |
| --- |

## Câu 7:



## Câu 8:

– **Presentation Layer (GUI):**

Thành phần chính:

* Trang Danh mục sản phẩm: duyệt, lọc, sắp xếp.
* Trang Chi tiết sản phẩm: thông tin, đánh giá, tồn kho.
* Trang Giỏ hàng: hiển thị sản phẩm đã chọn, tổng tiền, khuyến mãi.  
  Trang Thanh toán: xác nhận đơn hàng, nhập thông tin thanh toán.
* Trang Đăng nhập/Đăng xuất: kiểm soát quyền truy cập.  
  Trang Quản trị (Admin Dashboard): quản lý sản phẩm, kho, giỏ hàng của người dùng.

Chức năng:

* Hiển thị dữ liệu từ lớp nghiệp vụ (BLL)
* Gửi yêu cầu người dùng (mua hàng, đánh giá, đăng nhập...) đến lớp nghiệp vụ
* Xử lý tương tác người dùng và phản hồi

**– Business Logic Layer (BLL):**

Thành phần chính:

* Product Catalog Service: quản lý sản phẩm (CRUD, lọc, sắp xếp, đánh dấu sản phẩm nổi bật).
* Shopping Cart Service: thêm/xóa sản phẩm vào giỏ, tính tổng tiền, khuyến mãi, phí ship.
* Payment Service: xử lý thanh toán, xác thực đơn hàng, gửi email.
* Inventory Service: quản lý kho (số lượng tồn, vị trí kho, gán sản phẩm vào kho).
* Rating Service: quản lý đánh giá sản phẩm (1–5 sao).  
  Access Control Service (Auth): xác thực người dùng (buyer/admin), phân quyền.

Chức năng:

* Kiểm tra tồn kho khi mua hàng
* Tính toán khuyến mãi, phí vận chuyển
* Xác thực người dùng và phân quyền
* Gửi email xác nhận sau thanh toán
* Quản lý trạng thái giỏ hàng và sản phẩm

**– Data Access Layer (DAL):**

Thành phần chính:

* Cơ sở dữ liệu:
  + Bảng sản phẩm
  + Bảng kho hàng
  + Bảng giỏ hàng
  + Bảng đơn hàng
  + Bảng người dùng
  + Bảng đánh giá

Chức năng:

* Truy xuất dữ liệu theo yêu cầu từ lớp nghiệp vụ
* Lưu trạng thái giỏ hàng, đơn hàng, đánh giá
* Quản lý dữ liệu người dùng và phân quyền

## Câu 9:

**Test Case 1: Thêm sản phẩm hợp lệ vào giỏ hàng từ trang danh sách sản phẩm**

* **Kịch bản kiểm thử:**
* Người dùng chọn sản phẩm và thêm sản phẩm vào giỏ hàng
* **Dữ liệu đầu vào**:.
* Sản phẩm: "Áo Thun Nam" (ProductID = P01, tồn kho = 50, giá = 200.000 VNĐ).
* Số lượng thêm: 2
* **Các bước kiểm tra:**
* Người dùng truy cập danh mục sản phẩm.
* Nhấn nút **“Mua hàng”** tại sản phẩm P001.
* Nhập số lượng = 2 và xác nhận
* **Kết quả mong đợi**:
* Hệ thống thêm 2 sản phẩm "Áo Thun Nam" vào giỏ hàng.
* Giỏ hàng hiển thị sản phẩm với tổng giá = 400.000 VNĐ.
* Bảng tổng hợp giỏ hàng được cập nhật đúng (tổng tiền tăng thêm 400.000 VNĐ).

**Test Case 2: Thêm sản phẩm vượt quá tồn kho**

* **Kịch bản kiểm thử:**
* Người dùng chọn thêm sản phẩm vào giỏ hàng và số lượng vượt quá tồn kho
* **Dữ liệu đầu vào**:
  + Sản phẩm: "Giày Thể Thao" (ProductID = P002, tồn kho = 5, giá = 800.000 VNĐ).
  + Số lượng thêm: 10.
* **Bước thực hiện**:
  + Người dùng truy cập trang chi tiết sản phẩm P002.
  + Nhập số lượng = 10.
  + Nhấn nút **“Mua hàng”**.
* **Kết quả mong đợi**:
  + Hệ thống hiển thị thông báo lỗi: *“Số lượng vượt quá tồn kho (chỉ còn 5 sản phẩm)”*.
  + Giỏ hàng không thay đổi.

**Test Case 3: Thêm sản phẩm đã bị vô hiệu hóa**

* **Kịch bản kiểm thử**: Kiểm tra khi sản phẩm đã bị quản trị viên vô hiệu hóa (không còn bán).
* **Dữ liệu đầu vào**:
  + Sản phẩm: "Áo Khoác Mùa Đông" (ProductID = P006, trạng thái = Inactive, giá = 600.000 VNĐ).
  + Số lượng thêm: 1.
* **Bước thực hiện**:
  + Người dùng nhấn **“Mua hàng”** trên sản phẩm P006.
* **Kết quả mong đợi**:
  + Hệ thống hiển thị thông báo: *“Sản phẩm không khả dụng”*.
  + Giỏ hàng không thay đổi.

**Test Case 4: Thêm nhiều sản phẩm khác nhau vào giỏ hàng**

* **Kịch bản kiểm thử**: Kiểm tra hệ thống khi thêm nhiều loại sản phẩm vào cùng giỏ hàng.
* **Dữ liệu đầu vào**:
  + Sản phẩm 1: "Áo sơ mi" (P013, giá 300.000, số lượng thêm 1).
  + Sản phẩm 2: "Quần jean" (P014, giá 500.000, số lượng thêm 2).
* **Bước thực hiện**:
  + Người dùng thêm 1 sản phẩm P013 vào giỏ hàng.
  + Sau đó thêm 2 sản phẩm P014 vào giỏ hàng.
* **Kết quả mong đợi**:
  + Giỏ hàng hiển thị 2 dòng sản phẩm (P013 và P014).
  + Tổng giá trị = 300.000 + (500.000 × 2) = 1.300.000 VNĐ.