UỶ BAN NHÂN DÂN

THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC SÀI GÒN**



**HỌC PHẦN KIỂM THỬ PHẦN MỀM**

**Lab 2**

Giáo viên hướng dẫn: Thầy Đỗ Như Tài

Thành viên nhóm:

1. Nguyễn Văn Khánh Duy – 3122411027
2. Âu Thái Khang – 3122411086
3. Nguyễn Đức Gia Bảo – 3122411011
4. Huỳnh Hữu Tứ- 3122411240

Mục lục

[Phân công việc 3](#_Toc209128236)

[BÀI TẬP MINH HỌA 4](#_Toc209128237)

[Bài 1 4](#_Toc209128238)

[Bài 2 5](#_Toc209128239)

[Bài 3 5](#_Toc209128240)

[Bài 4 7](#_Toc209128241)

[Bài 5 8](#_Toc209128242)

[Bài 6 10](#_Toc209128243)

[Bài 7 10](#_Toc209128244)

[BÀI TẬP ỨNG DỤNG 17](#_Toc209128245)

[Bài 1 17](#_Toc209128246)

[Bài 2 18](#_Toc209128247)

[Bài 3 20](#_Toc209128248)

[Bài 4 20](#_Toc209128249)

[Bài 5 21](#_Toc209128250)

[Bài 6 22](#_Toc209128251)

[Bài 7 22](#_Toc209128252)

[Bài 8 24](#_Toc209128253)

[Bài 9 26](#_Toc209128254)

# Phân công việc

|  |  |
| --- | --- |
| Thành viên | Công việc |
| Nguyễn Văn Khánh Duy | Bài tập minh họa: bài 3,4,5.  Bài tập vận dụng: bài 1,2,3,4,5,9. |
| Âu Thái Khang | Bài tập minh họa: bài 6,7.  Bài tập vận dụng: bài 6,7. |
| Nguyễn Đức Gia Bảo | Bài tập minh họa: bài 2.  Bài tập vận dụng: bài 7. |
| Huỳnh Hữu Tứ | Bài tập minh họa: bài 1.  Bài tập vận dụng: bài 8. |

# BÀI TẬP MINH HỌA

## Bài 1

A diagram of a flowchart

Description automatically generated

## Bài 2

Mô hình khá niệm được vẽ lại trong file Bai2.drawio.

A diagram of a car

Description automatically generated

## Bài 3

Businesss Usecase được vẽ lại trong file bai3.drawio.

a/

Các tác nhân chính (primary actor) bao gồm:

* Student: thực hiện các hành động như đăng ký môn học.
* Professor: chọn các môn phần để giảng dạy.
* Registrar: quản lý chương trình học, thông tin của Student và Professor.

Các tác nhân phụ (secondary actor) bao gồm:

* Billing System: vì thực thể này không kích hoạt hay sử dụng bất kỳ chức năng nào của hệ thống đang xét tới. Ngược lại, hệ thống đang được xét lại sử dụng tính năng thanh toán của Billing System.

b/

Các Usecase bao gồm:

* Register for Courses: usecase này được dùng để miêu tả sự tương tác giữa Student và hệ thống để đăng ký học phần. Sau khi đăng ký xong, hệ thống trong usecase này sẽ tương tác với hệ thống bên ngoài là Billing System để thực hiện thanh toán.
* Select Course to Teach: usecase này được dùng để miêu tả sự tương tác giữa Professor và hệ thống để chọn học phần giảng dạy.
* Maintain Curriculum: usecase này được dùng để miêu tả sự tướng tác giữa Registrar và hệ thống để điều chỉnh chương trình học.
* Maintain Student Information: usecase này được dùng để miêu tả sự tướng tác giữa Registrar và hệ thống để điều chỉnh thông tin của Student.
* Maintain Student Information: usecase này được dùng để miêu tả sự tướng tác giữa Registrar và hệ thống để điều chỉnh thông tin của Professor.
* Registrar Validation: usecase được dùng để mô tả lại việc hệ thống xác thực lại cập nhật về thông tin của Student, Professor, Curriculum bởi Registrar

c/

Mối quan hệ use giữa Maintain Curriculum và Registrar Validation: usecase Maintain Curriculum sẽ sử dụng usecase Registrar Validation để xác thực lại thông tin về chương trình học sau khi Registrar thực hiện việc cập nhật thông tin này.

Mối quan hệ use giữa Maintain Student Information và Registrar Validation: usecase Maintain Student Information sẽ sử dụng usecase Registrar Validation để xác thực lại thông tin của Student sau khi Registrar thực hiện việc cập nhật thông tin này.

Mối quan hệ use giữa Maintain Professor Information và Registrar Validation: usecase Maintain Professor Information sẽ sử dụng usecase Registrar Validation để xác thực lại thông tin của Professor sau khi Registrar thực hiện việc cập nhật thông tin này.

d/ [not done: có cần tìm thêm không]

Tên: Register for Course

Mô tả: Student đăng ký học phần

Actor: Student

Tiền điều kiện:

* Đã đăng nhập thành công vào hệ thống với loại tài khoản là Student.
* Trong thời gian đăng ký học phần.

Kịch bản chính:

1. Hệ thống hiện lên giao diện tóm tắt hoạt động đăng ký (bao gồm các học phần ở trạng thái Pending và Registered).
2. Actor tìm học phần dựa trên môn học, mã học phần, tên học phần.
3. Hệ thống liệt kê ra các học phần thỏa điều kiện lọc.
4. Actor chọn các học phần muốn học, hủy các học phần đã được đăng ký trong giao diện tóm tắt hoạt động đăng ký.
5. Hệ thống tạm lưu các học phần được chọn ở trạng thái Pending và cập nhật lại giao diện tóm tắt hoạt động đăng ký.
6. Actor chọn submit để lưu hoạt động đăng ký học phần.
7. Hệ thống xác thực hoạt động đăng ký học phần.
8. Hệ thống lưu lại hoạt động đăng ký học phần, chuyển các học phần ở trạng thái Pending thành Registered, hủy các học phần được sinh viên chọn hủy.
9. Hệ thống tương tác với Billing System để đưa ra báo cáo chi phí cho từng học phần và tổng số tiền phải trả.

Kịch bản phụ:

6.a. Actor chọn hủy.

1. Hệ thống xóa bộ nhớ tạm lưu.

7.a. Việc xác thực xảy ra lỗi

1. Hệ thống hiện thị các lỗi tương ứng

Ghi chú:

* Việc xác thực hoạt động đăng ký học phần bao gồm: kiểm tra xem các học phần được chọn có thõa điều kiện tiến quyết về các mỗi phải học trước đó. [not done: search thêm các tiêu chí xác thực]

e/

## Bài 4

Sơ đồ hoạt động được vẽ lại file bai4-1.drawio.

A diagram of a flowchart

Description automatically generated

Sơ đồ tương tác được vẽ lại trong file bai4-2.drawio.

A diagram of a diagram

Description automatically generated

Ý nghĩa của hai sơ đồ đều được dùng để miêu tả quá trình mua và thanh toán sản phẩm. Trong đó, activity diagram được sử dụng để miêu tả luồng hoạt động: các hoạt động liên quan tới việc mua và thanh toán, các điều kiện rẽ nhánh khi mua và thanh toán. Trong khi, sequence diagram được dùng để miêu tả sự tương tác của các thành phần bên trong hệ thống (Man hinh, Register, Database) để thực hiện kịch bản mua và thanh toán.

## Bài 5

Lược đồ lớp được vẽ lại trong file bai5.drawio

A diagram of a company

Description automatically generated

## Bài 6

State diagram được vẽ lại trong file bai6.drawio.

A diagram of a flowchart

Description automatically generated

A diagram of a flowchart

Description automatically generated

## Bài 7

Thiết kế cơ sở dữ liệu ở mức quan niệm được vẽ lại trong bai7.drawio:

A diagram of a company

Description automatically generated with medium confidence

Thiết kế cơ sở dữ liệu ở mức vật lý:

1.Bảng Branch

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | Name | nvarchar | 50 | No | Tên chi nhánh |
| 2 | AddressNo | nvarchar | 10 | No | Số nhà của chi nhánh |
| 3 | Street | nvarchar | 50 | No | Tên đường |
| 4 | District | nvarchar | 30 | No | Quận/Huyện |
| 5 | City | nvarchar | 30 | No | Thành phố |
| 6 | Region | nvarchar | 30 | No | Vùng/Khu vực |
| 7 | Email | nvarchar | 50 | Yes | Email liên lạc của chi nhánh |
| 8 | MngCode | nvarchar | 20 | No | Mã quản lý chi nhánh (khóa chính) |

2.Bảng BRANCHPHONE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | BraName | nvarchar | 50 | No | Tên chi nhánh (khóa ngoại) |
| 2 | Phone | nvarchar | 15 | No | Số điện thoại của chi nhánh |

3.Bảng BRANCHFAX

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | BraName | nvarchar | 50 | No | Tên chi nhánh (khóa ngoại) |
| 2 | Fax | nvarchar | 15 | No | Số fax của chi nhánh |

4.Bảng EMPLOYEE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | Code | nvarchar | 20 | No | Mã nhân viên (khóa chính) |
| 2 | FName | nvarchar | 50 | No | Tên đệm và tên của nhân viên |
| 3 | LName | nvarchar | 50 | No | Họ của nhân viên |
| 4 | AddressNo | nvarchar | 10 | No | Số nhà của nhân viên |
| 5 | Street | nvarchar | 50 | No | Tên đường |
| 6 | District | nvarchar | 30 | No | Quận/Huyện |
| 7 | City | nvarchar | 30 | No | Thành phố |
| 8 | DOB | date |  | No | Ngày sinh của nhân viên |
| 9 | Email | nvarchar | 50 | Yes | Email liên lạc của nhân viên |
| 10 | BraName | nvarchar | 50 | No | Tên chi nhánh (khóa ngoại) |

5.Bảng EMPLOYEEPHONE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | EmpCode | nvarchar | 20 | No | Mã nhân viên (khóa ngoại) |
| 2 | Phone | nvarchar | 15 | No | Số điện thoại của nhân viên |

6.Bảng CUSTOMER

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | Code | nvarchar | 20 | No | Mã khách hàng (khóa chính) |
| 2 | FName | nvarchar | 50 | No | Tên đệm và tên của khách hàng |
| 3 | LName | nvarchar | 50 | No | Họ của khách hàng |
| 4 | OfficeAddress | nvarchar | 100 | Yes | Địa chỉ văn phòng |
| 5 | HomeAddress | nvarchar | 100 | Yes | Địa chỉ nhà riêng |
| 6 | Email | nvarchar | 50 | Yes | Email liên lạc của khách hàng |
| 7 | EmpCode | nvarchar | 20 | No | Mã nhân viên phụ trách (khóa ngoại) |

7. Bảng CUSTOMERPHONE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | CusCode | nvarchar | 20 | No | Mã khách hàng (khóa ngoại) |
| 2 | Phone | nvarchar | 15 | No | Số điện thoại của khách hàng |

8. Bảng ACCOUNT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | Number | nvarchar | 20 | No | Số tài khoản (khóa chính) |
| 2 | CusCode | nvarchar | 20 | No | Mã khách hàng (khóa ngoại) |

9.Bảng SAVINGACCOUNT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | AccNumber | nvarchar | 20 | No | Số tài khoản tiết kiệm (khóa chính) |
| 2 | Balance | decimal |  | No | Số dư tài khoản |
| 3 | InterestRate | decimal |  | No | Lãi suất (%) |
| 4 | OpenedDate | date |  | No | Ngày mở tài khoản |

10.Bảng CHECKINGACCOUNT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | AccNumber | nvarchar | 20 | No | Số tài khoản thanh toán (khóa chính) |
| 2 | Balance | decimal |  | No | Số dư tài khoản |
| 3 | OpenedDate | date |  | No | Ngày mở tài khoản |

11 Bảng Loan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Filed Name** | **Type** | **Size** | **Null** | **Description** |
| 1 | AccNumber | nvarchar | 20 | No | Số tài khoản vay (khóa chính) |
| 2 | BalanceDue | decimal |  | No | Số tiền còn nợ |
| 3 | InterestRate | decimal |  | No | Lãi suất (%) |
| 4 | TakenDate | date |  | No | Ngày vay |

Câu truy vấn:

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

INSERT INTO BRANCH (Name, AddressNo, Street, District, City, Region, Email, MngCode)

VALUES ('Main Branch', '101', 'High Street', 'Downtown', 'Hanoi', 'North', 'main@bank.com', 'M001'),

('West Branch', '202', 'West Road', 'Westside', 'Hanoi', 'North', 'west@bank.com', 'M002'),

('East Branch', '303', 'East Avenue', 'Eastside', 'Hanoi', 'North', 'east@bank.com', 'M003'),

('South Branch', '404', 'South Lane', 'Southside', 'Hanoi', 'North', 'south@bank.com',

'M004').

INSERT INTO EMPLOYEE (Code, FName, LName, AddressNo, Street, District, City, DOB, Email, BraName)

VALUES ('N1080', 'John', 'Doe', '105', 'Park Ave', 'Downtown', 'Hanoi', '1985-03-15', 'john.doe@bank.com', 'Main Branch'),

('N1081', 'Jane', 'Smith', '206', 'Hill St', 'Westside', 'Hanoi', '1990-06-20', 'jane.smith@bank.com', 'West Branch'),

('N1082', 'Mike', 'Brown', '307', 'Oak Rd', 'Eastside', 'Hanoi', '1988-09-10', 'mike.brown@bank.com', 'East Branch'),

('N1083', 'Anna', 'Lee', '408', 'Pine Ln', 'Southside', 'Hanoi', '1992-12-05', 'anna.lee@bank.com', 'South Branch');

INSERT INTO CUSTOMER (Code, FName, LName, OfficeAddress, HomeAddress, Email, EmpCode)

VALUES ('C001', 'Peter', 'Johnson', '101 Office St', '102 Home Rd', 'peter.j@customer.com', 'N1080'),

('C002', 'Mary', 'Williams', '203 Office Ave', '204 Home St', 'mary.w@customer.com', 'N1081'),

('C003', 'Robert', 'Brown', '305 Office Rd', '306 Home Ln', 'robert.b@customer.com', 'N1082'),

('C004', 'Sarah', 'Davis', '407 Office Pl', '408 Home Ave', 'sarah.d@customer.com', 'N1083');

INSERT INTO BRANCHPHONE (BraName, Phone)

VALUES ('Main Branch', '02412345678'),

('West Branch', '02412345679'),

('East Branch', '02412345680'),

('South Branch', '02412345681');

INSERT INTO BRANCHFAX (BraName, Fax)

VALUES ('Main Branch', '02498765432'),

('West Branch', '02498765433'),

('East Branch', '02498765434'),

('South Branch', '02498765435');

INSERT INTO EMPLOYEEPHONE (EmpCode, Phone)

VALUES ('N1080', '0912345678'),

('N1081', '0912345679'),

('N1082', '0912345680'),

('N1083', '0912345681');

INSERT INTO CUSTOMERPHONE (CusCode, Phone)

VALUES ('C001', '0901234567'),

('C002', '0901234568'),

('C003', '0901234569'),

('C004', '0901234570');

INSERT INTO ACCOUNT (Number, CusCode)

VALUES ('A001', 'C001'),

('A002', 'C002'),

('A003', 'C003'),

('A004', 'C004');

INSERT INTO SAVINGACCOUNT (AccNumber, Balance, InterestRate, OpenedDate)

VALUES ('A001', 5000.00, 2.5, '2020-01-15'),

('A002', 7500.00, 2.5, '2020-02-20'),

('A003', 3000.00, 2.0, '2020-03-10'),

('A004', 9000.00, 2.5, '2020-04-05');

INSERT INTO CHECKINGACCOUNT (AccNumber, Balance, OpenedDate)

VALUES ('A001', 2000.00, '2020-01-15'),

('A002', 1500.00, '2020-02-20'),

('A003', 1000.00, '2020-03-10'),

('A004', 2500.00, '2020-04-05');

INSERT INTO LOAN (AccNumber, BalanceDue, InterestRate, TakenDate)

VALUES ('A001', 10000.00, 5.0, '2021-01-15'),

('A002', 15000.00, 5.5, '2021-02-20'),

('A003', 8000.00, 4.5, '2021-03-10'),

('A004', 12000.00, 5.0, '2021-04-05');

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';

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';

**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(sa.Balance) AS TotalBalance

FROM SAVINGACCOUNT sa

JOIN ACCOUNT a ON sa.AccNumber = a.Number

JOIN CUSTOMER c ON a.CusCode = c.Code

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

# BÀI TẬP ỨNG DỤNG

## Bài 1

Functional requirement:

* Duyệt sản phẩm dựa trên chức năng lọc, sắp xếp kết quả lọc theo tên sản phẩm và giá
* Xem chi tiết sản phẩm:
  + Bao gồm các thông tin của sản phẩm như tên, mô tả, số lượng tồn kho, thông tin kho như địa chỉ kho, trạng thái "sản phẩm nổi bật" (nếu có) và đánh giá.
* Quản lý sản phẩm
  + Admin CRUD sản phẩm
  + Admin gán sản phẩm vào kho hiện có
* Mua sản phẩm
  + Mua sản phẩm trực tiếp trên danh sách sản phẩm qua nút “Mua hàng”
  + Mua sản phẩm trực tiếp trên trang chi tiết sản phẩm
  + Bảng tổng hợp và giỏ hàng sẽ được cập nhật
* Xem giỏ hàng
  + Có 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.
* Xóa sản phẩm khỏi giỏ hàng
  + Bảng tổng hợp và giỏ hàng sẽ được cập nhật
* Admin xem toàn bộ giỏ hàng của bất kỳ người nào
* Admin Kích hoạt hoặc vô hiệu hóa giỏ hàng
* Thanh toán giỏ hàng
  + Xác thực thông tin sản phẩm trước khi thanh toán
  + Xử lý thanh toán
  + Thống báo email tới khách hàng
* Admin quản lý kho hàng
* Đánh giá sản phẩm trên thang điểm từ 1-5 sao
* Đăng nhập
  + Khi đăng nhập xong thì đưa người dùng đến đúng trang bắt đầu. Đưa tới trang danh mục sản phẩm nếu là tài khoản khách hàng, đưa tới trang quản trị nếu là quản trị viên.
* Đăng xuất

Non-functional requirement:

## Bài 2

Quy trình nghiệp vụ cho khách hàng

A white paper with black lines and a rectangular object with a yellow and red text

Description automatically generated with medium confidence

Quy trình nghiệp vụ cho quản trị viên

A screenshot of a white paper

Description automatically generated

## Bài 3

A diagram of a diagram

Description automatically generated

## Bài 4

Tên usecase: Mua sản phẩm từ trang chi tiết sản phẩm

Mô tả: Khác hàng mua sản phẩm từ trang chi tiết sản phẩm

Mục tiêu: Thêm sản phẩm vào giỏ hàng

Tác nhân chính: Khách hàng

Tiền điều kiện:

Kịch bản chính:

1. Actor thực hiện usecase “Xem chi tiết sản phảm”
2. Hệ thống hiện thị thông tin chi tiết sản phẩm
3. Actor chọn mua sản phẩm
4. Hệ thống thêm sản phẩm vào giỏ hàng và thông báo thành công

Kịch bản phụ:

4.a. Actor chưa đăng nhập

1. Hệ thống thông báo khách hàng chưa đăng nhập
2. Hệ thống chuyển đến trang đăng nhập

4.b. Actor đăng nhập với vai trò không phải khách hàng

1. Hệ thống thông báo “Tài khoản đăng nhập phải là khách hàng”

4.c. Sản phẩm đã có trong giỏ hàng

1. Hệ thống thông báo sản phẩm đã có trong giỏ hàng

Hậu điều kiện: Sản phẩm được thêm vào giỏ hàng

Ghi chú:

## Bài 5

A diagram of a flowchart

Description automatically generated

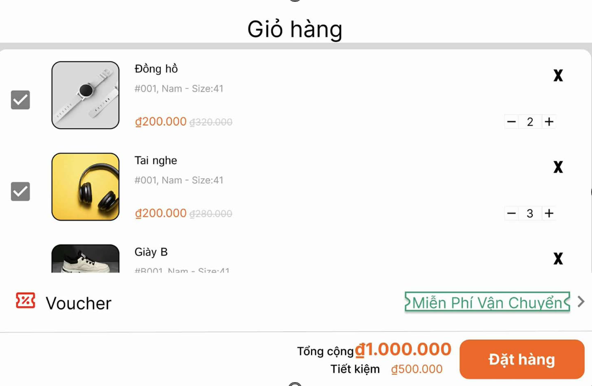
## Bài 6

A diagram of a computer

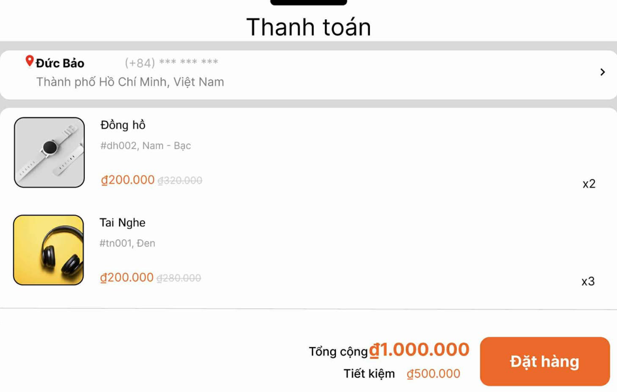
Description automatically generated

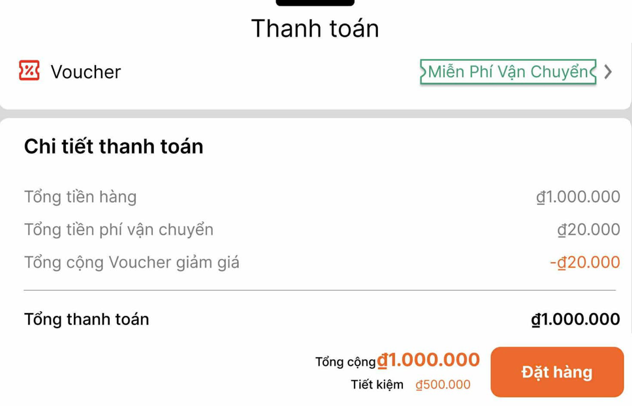
## Bài 7

Giao diện giỏ hàng:

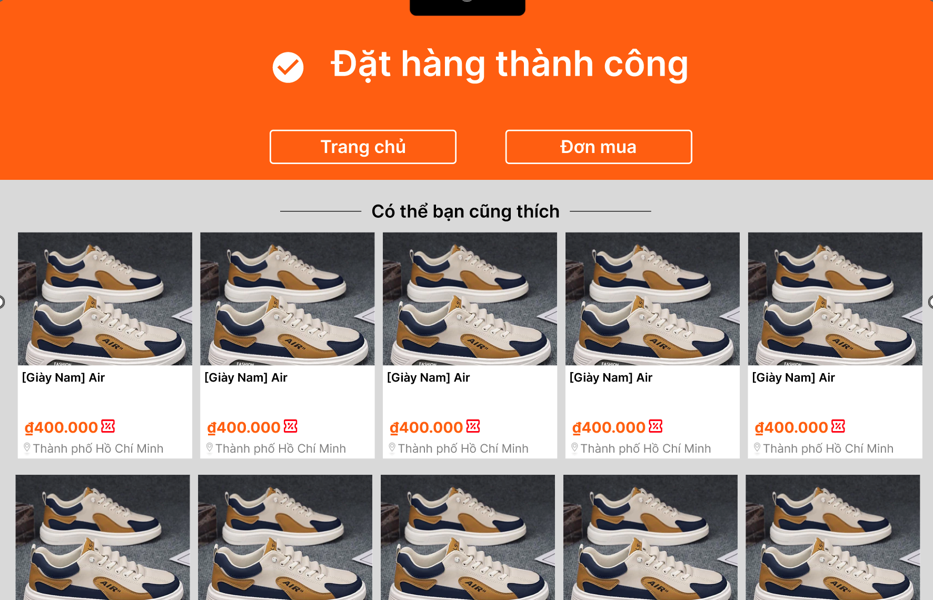


Giao diện thanh toán:





Giao diện sau thanh toán:



## Bài 8

a) Presentation Layer (UI / Client Layer):

* Công nghệ**:** React/Angular
* Thành phần chính**:**
  + Buyer UI: Product Catalog, Product Detail, Cart Page, Checkout Page, Login/Register, Rating.
  + SysAdmin UI: Admin Dashboard (quản lý sản phẩm, kho, giỏ hàng).
  + Shared: Authentication (login/logout), Navigation.

b) Business Logic Layer:

* Công nghệ**:** Node.js (Express/NestJS)
* Cấu trúc gợi ý:
  + ProductController (quản lý danh mục, chi tiết, filter/sort).
  + CartController(thêm/xóa/cập nhật giỏ hàng, tính toán tổng tiền).
  + OrderController (checkout, payment, gửi email).
  + InventoryController (quản lý kho, tồn kho).
  + RatingController (quản lý đánh giá).
  + UserController (auth, access control).

c) Data Access Layer

* Công nghệ:
  + ORM: Sequelize (Node.js)
  + Database: PostgreSQL
* Thành phần chính (Repository):
  + ProductRepository
  + CartRepository
  + OrderRepository
  + InventoryRepository
  + UserRepository
  + RatingRepository

**2. Database Design sơ bộ**

* **User** (UserID, Username, PasswordHash, Role).
* **Product** (ProductID, Name, Description, Price, HotFlag).
* **Inventory** (InventoryID, Stock, Address, Location, ProductID).
* **Cart** (CartID, UserID, Status).
* **CartItem** (CartItemID, CartID, ProductID, Quantity, Price).
* **Order** (OrderID, CartID, PaymentStatus, TotalAmount).
* **Rating** (RatingID, UserID, ProductID, Stars, Comment, Date).

**3. Luồng xử lý ví dụ (Checkout Flow)**

* Buyer click **Checkout** trên React UI.
* React gọi API POST /orders/checkout
* OrderController -> checkout: kiểm trả giỏ hàng (CartController), xác nhận tồn kho (validate) (InventoryController), xử lý thanh toán, lưu order vào Postgres.
* CartController cập nhật trạng thái giỏ hàng (Closed).

Trả kết quả về React UI -> hiển thị thông báo thành công.

## Bài 9

Testcase 1:

* Mô tả: Đăng nhập thành công với vai trò là khách hàng. Giỏ hàng của khách hàng đang không chứ bất kỳ sản phẩm nào.
* Dữ liệu đầu vào: Sản phẩm A với giá mua là 10.000đ
* Bước thực hiện: Duyệt sản phẩm theo tên để tìm sản phẩm A, chọn mua sản phẩm A trực tiếp trên danh sách sản phẩm hoặc vào trang chi tiết sản phẩm để mua.
* Kết quả mong muốn: Sản phẩm A được thêm vào giỏ hàng với số lượng là 1, giỏ hàng và bảng tổng hợp chỉ chứa mô tả về sản phẩm A.

Testcase 2:

* Mô tả: Đăng nhập thành công với vai trò là khách hàng. Giỏ hàng của khách hàng đang chứa 3 sản phẩm B, 4 sản phẩm C.
* Dữ liệu đầu vào: Sản phẩm A với giá mua là 10.000đ
* Bước thực hiện: Duyệt sản phẩm theo tên để tìm sản phẩm A, chọn mua sản phẩm A trực tiếp trên danh sách sản phẩm hoặc vào trang chi tiết sản phẩm để mua.
* Kết quả mong muốn: Sản phẩm A được thêm vào giỏ hàng với số lượng là 1, giỏ hàng và bảng tổng hợp cập nhật lại thông tin để chứa thêm 1 sản phẩm A.

Testcase 3:

* Mô tả: Đăng nhập thành công với vai trò là khách hàng. Giỏ hàng của khách hàng đang chứa sản phẩm A.
* Dữ liệu đầu vào: Sản phẩm A với giá mua là 10.000đ
* Bước thực hiện: Duyệt sản phẩm theo tên để tìm sản phẩm A, chọn mua sản phẩm A trực tiếp trên danh sách sản phẩm hoặc vào trang chi tiết sản phẩm để mua.
* Kết quả mong muốn: Hệ thống thông báo sản phẩm A đã có trong giỏ hàng, từ chối thêm lại sản phẩm A vào giỏ hàng, giỏ hàng không cập nhật.

Testcase 4:

* Mô tả: Chưa đăng nhập.
* Dữ liệu đầu vào: Sản phẩm A với giá mua là 10.000đ
* Bước thực hiện: Duyệt sản phẩm theo tên để tìm sản phẩm A, chọn mua sản phẩm A trực tiếp trên danh sách sản phẩm hoặc vào trang chi tiết sản phẩm để mua.
* Kết quả mong muốn: Hệ thống thông báo phải đăng nhập với vai trò là khách khàng để mua hàng. Hệ thống chuyển sang trang đăng nhập

Testcase 5:

* Mô tả: Đăng nhập với vai trò là quản trị viên.
* Dữ liệu đầu vào: Sản phẩm A với giá mua là 10.000đ
* Bước thực hiện: Duyệt sản phẩm theo tên để tìm sản phẩm A, chọn mua sản phẩm A trực tiếp trên danh sách sản phẩm hoặc vào trang chi tiết sản phẩm để mua.
* Kết quả mong muốn: Hệ thống thông báo không phải khách hàng, từ chối việc mua hàng.