

FPT POLYTECHNIC

# MINI PROJECT REQUIREMENT

NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM)

**Người thực hiện: Đào Hữu Trí**

## MỤC LỤC

<b>1. Giới Thiệu Dự Án</b>	<b>2</b>
<b>2. Phân Tích Yêu Cầu Khách Hàng</b>	<b>2</b>
2.1 Sơ Đồ Use case	2
2.2 Sơ đồ triển khai và yêu cầu hệ thống	3
2.2.1 Mô hình triển khai	3
<b>3. THIẾT KẾ ỨNG DỤNG</b>	<b>4</b>
3.1 MÔ HÌNH CÔNG NGHỆ ỨNG DỤNG	4
3.2 THỰC THỂ	5
Error! Bookmark not defined.	
3.3 GIAO DIỆN	8
<b>4. THỰC HIỆN DỰ ÁN</b>	<b>11</b>
4.1 TẠO CSDL VỚI MYSQL	11
4.1.1 CHI TIẾT CÁC BẢNG	11
4.2 Lập trình CSDL	14
4.2.1 Mô hình tổ chức lập trình CSDL	14
4.3 Lập trình nghiệp vụ	15
<b>5. Kiểm thử</b>	<b>16</b>
<b>6. Đóng gói và triển khai</b>	<b>16</b>

## 1. Giới Thiệu Dự Án

Thiết kế và phát triển website với các yêu cầu như sau:

- ❖ Yêu cầu về chức năng nghiệp vụ:
  - Quản lý người dùng: quản lý các thông tin người sử dụng bao gồm các công việc như: thêm mới, cập nhật thông tin và xóa tài khoản.
  - Quản lý vai trò: quản lý được vai trò của mỗi tài khoản để thực hiện các công việc khác nhau.
  - Đăng nhập và đăng xuất vào website.
  - Hồ sơ người dùng: quản lý thông tin cá nhân của người dùng: họ tên, địa chỉ, email, hình ảnh.
- ❖ Yêu cầu về mô hình:
  - Sử dụng lombok, spring boot, mysql để lập trình phía back-end. Back end được thiết kế và lập trình để cung cấp các API để truy xuất ở front-end.
  - Sử dụng angular để lập trình giao diện front-end.
  - Sử dụng swagger để document các API.

## 2. Phân Tích Yêu Cầu Khách Hàng

### 2.1 Sơ Đồ Use case

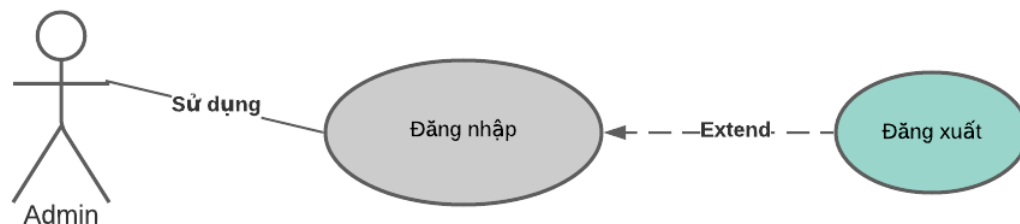
#### 2.1.1 Use Case đối tượng



### 2.1.2 Use Case đối tượng PM



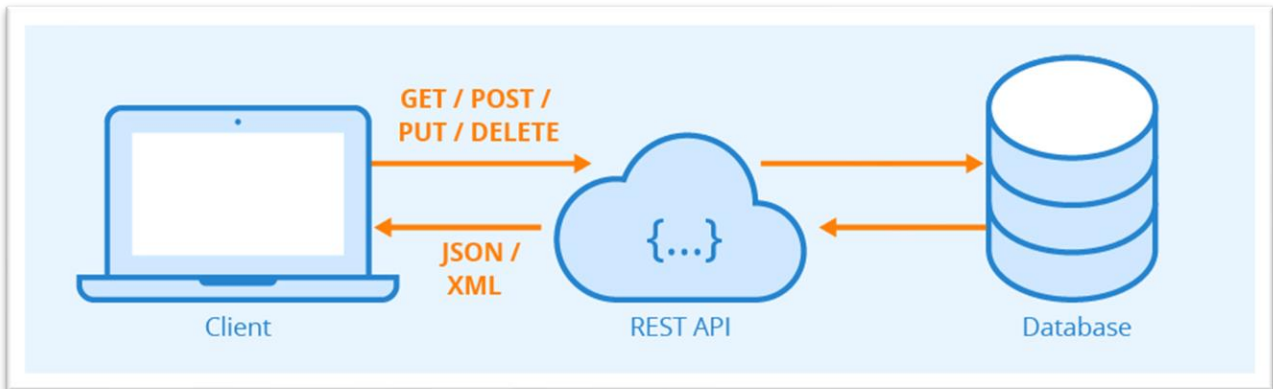
### 2.1.3 Use Case đối tượng USER



## 2.2 Sơ đồ triển khai và yêu cầu hệ thống

### 2.2.1 Mô hình triển khai

- ❖ Hệ quản trị CSDL MySQL và CSDL sẽ được cài đặt và sử dụng trên một máy chuyên biệt.

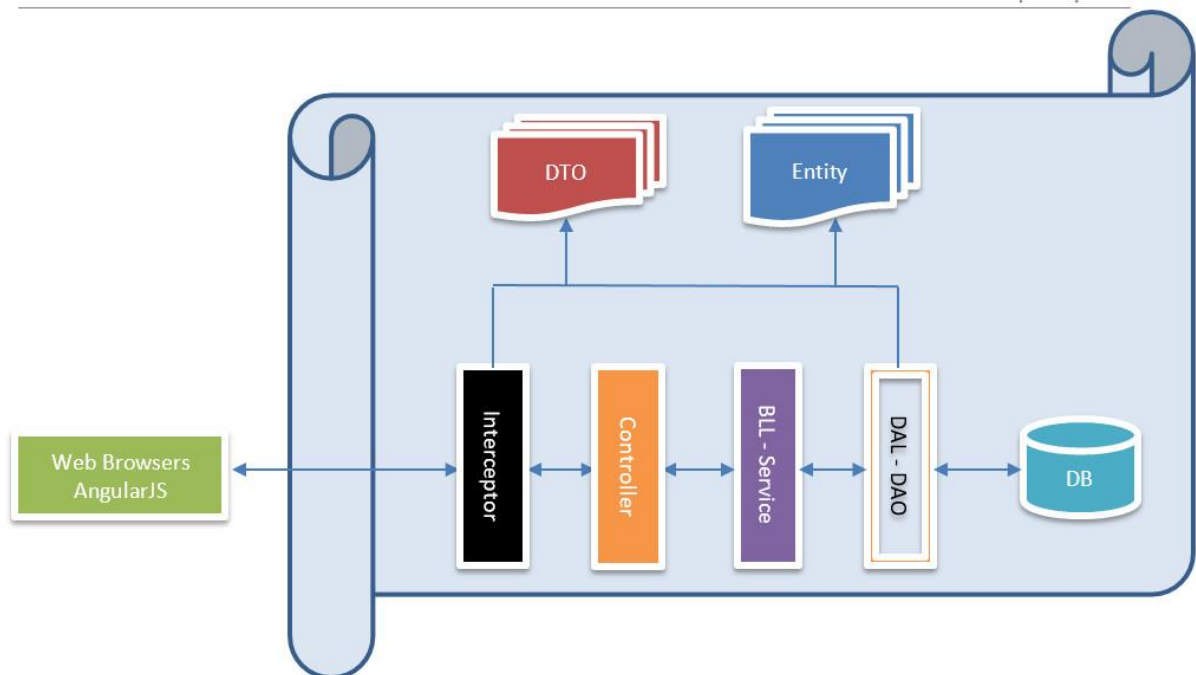


### 2.2.2 YÊU CẦU HỆ THỐNG

- ❖ Hệ điều hành : tối thiểu Windows 10
- ❖ Hệ quản trị CSDL : MySQL
- ❖ Ram: tối thiểu 4GB.

## 3. THIẾT KẾ ỨNG DỤNG

### 3.1 MÔ HÌNH CÔNG NGHỆ ỨNG DỤNG



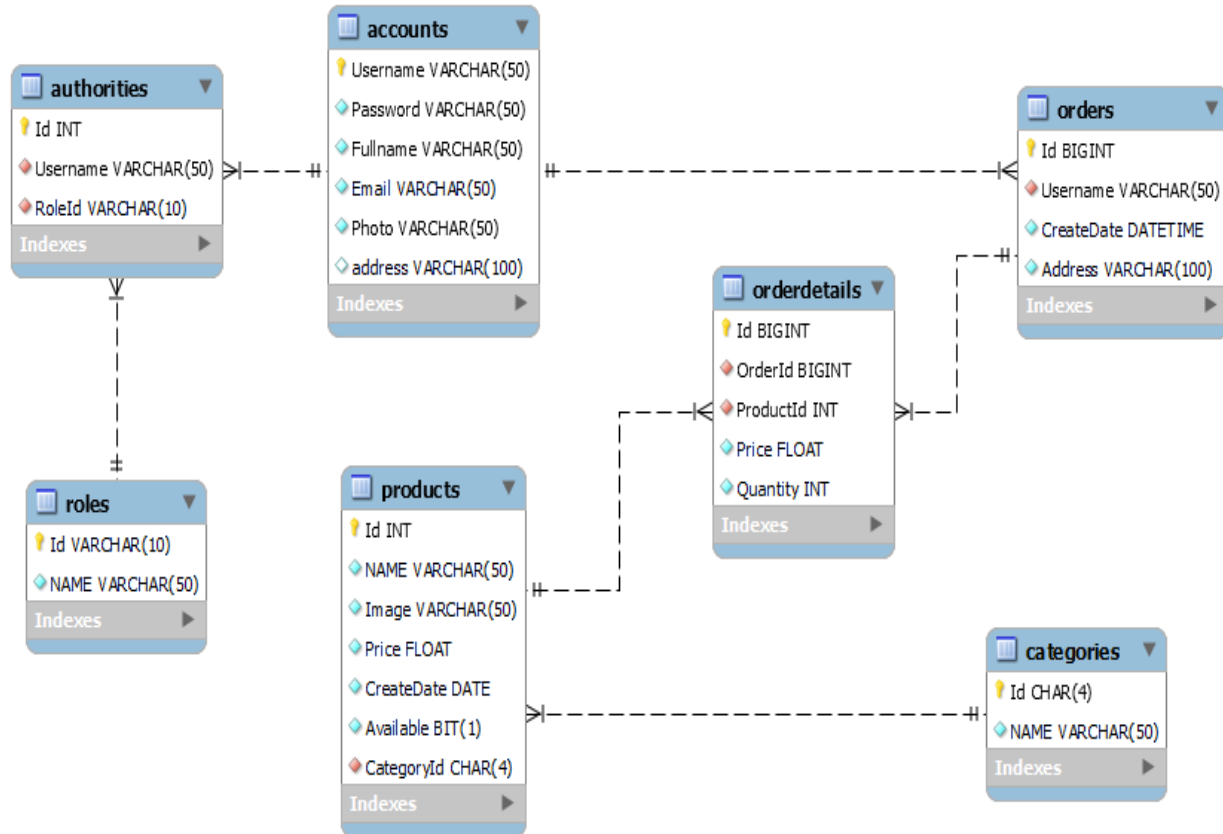
Trong đó:

- ❖ @Entity
- ❖ DAO extends JpaRepository (Data Access Layer (DAL))
- ❖ @Service (Business Logic Layer(BLL))

- ❖ @Controller, @RestController
- ❖ DTO (Data Transfer Object)

## 3.2 THỰC THỂ

### 3.2.1 SƠ ĐỒ QUAN HỆ THỰC THỂ



### 3.2.2 CHI TIẾT THỰC THỂ

#### 3.2.2.1 THỰC THỂ ACCOUNT

Accounts		
PK	Username	String
	Password	String
	Fullname	String
	Email	String
	Photo	String
	Address	String

THUỘC TÍNH	KIỂU DỮ LIỆU	MÔ TẢ
Username	String	Tên tài khoản
Password	String	Mật khẩu
Fullname	String	Họ và tên
Email	String	Địa chỉ email
Photo	String	Ảnh đại diện của người dùng
Address	String	Địa chỉ

#### 3.2.2.2 THỰC THỂ CATEGORIES

Categories		
PK	Id	Int
	Name	String

THUỘC TÍNH	KIỂU DỮ LIỆU	MÔ TẢ
Id	Int	Id Loại
Name	String	Tên loại

### 3.2.2.3 THỰC THỂ PRODUCTS

Products		
PK	Id	Int
	Name	String
	Image	String
	Price	Float
	CreateDate	Date
	Available	Bit
FK	CategoryId	Categories

THUỘC TÍNH	KIỂU DỮ LIỆU	MÔ TẢ
Id	Int	ID sản phẩm
Name	String	Tên sản phẩm
Image	String	Ảnh sản phẩm
Price	Float	Giá sản phẩm
CreateDate	Date	Ngày tạo sản phẩm
Available	Bit	Trạng thái
CategoryId	Categories	Id thể loại

### 3.2.2.4 THỰC THỂ LOẠI AUTHORITIES

Authorities		
PK	Id	Int
FK	Username	Accounts
FK	RoleId	Roles

THUỘC TÍNH	KIỂU DỮ LIỆU	MÔ TẢ
Id	Int	Id quyền sử dụng
Username	Accounts	Tên tài khoản của người dùng
RoleId	Roles	Id vai trò



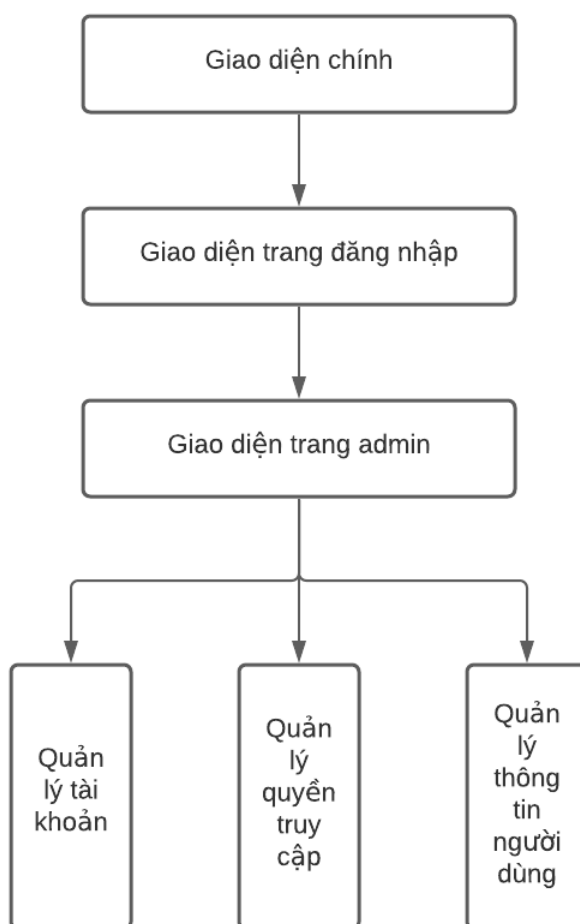
### 3.2.2.5 THỰC THỂ ROLES



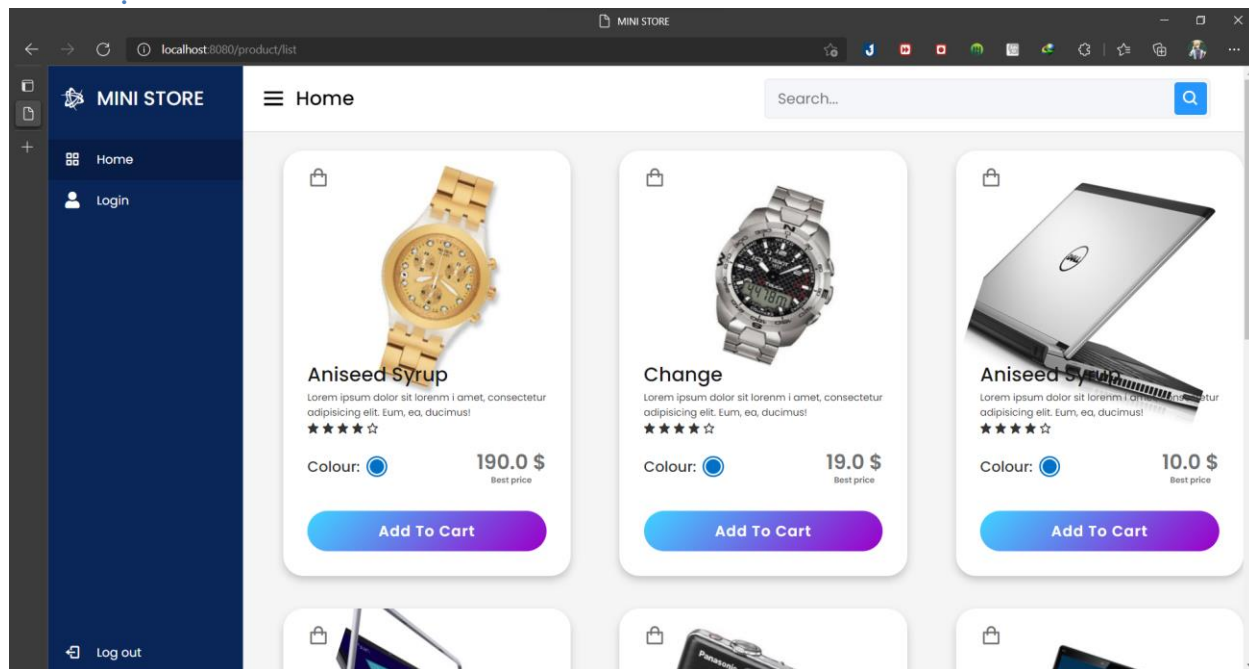
THUỘC TÍNH	KIỂU DỮ LIỆU	MÔ TẢ
Id	Int	Id vai trò
Name	String	Tên vai trò

## 3.3 GIAO DIỆN

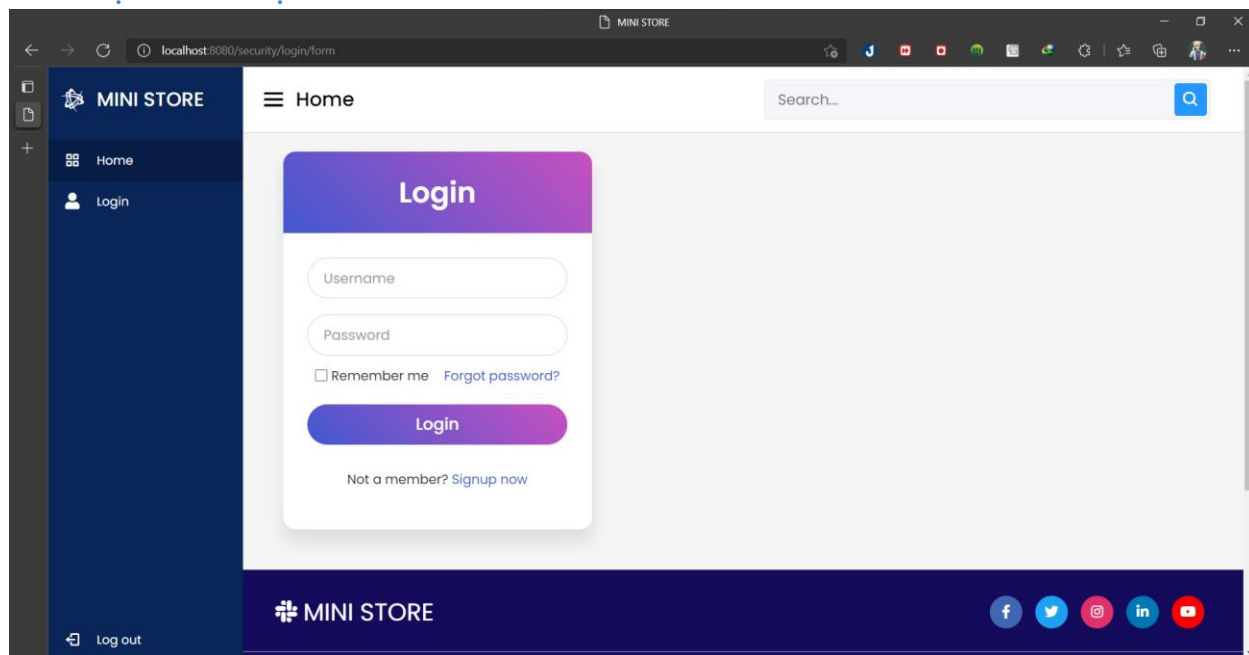
### 3.3.1 SƠ ĐỒ TỔ CHỨC GIAO DIỆN



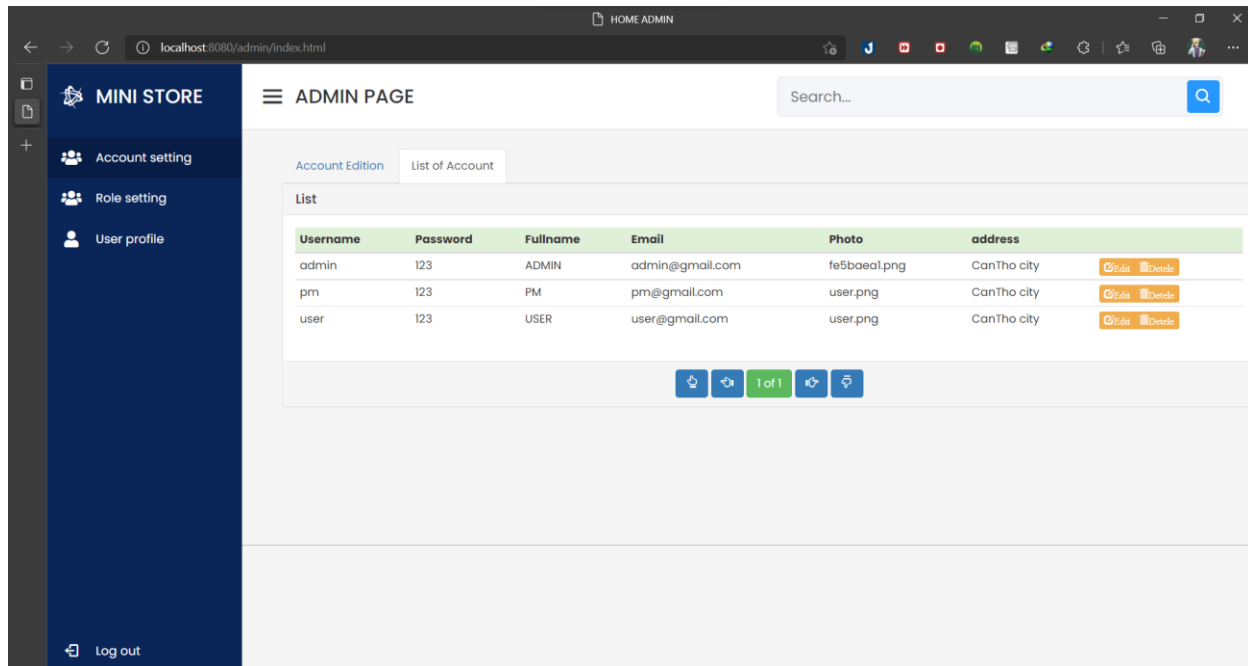
### 3.3.2 GIAO DIỆN CỬA SỔ CHÍNH



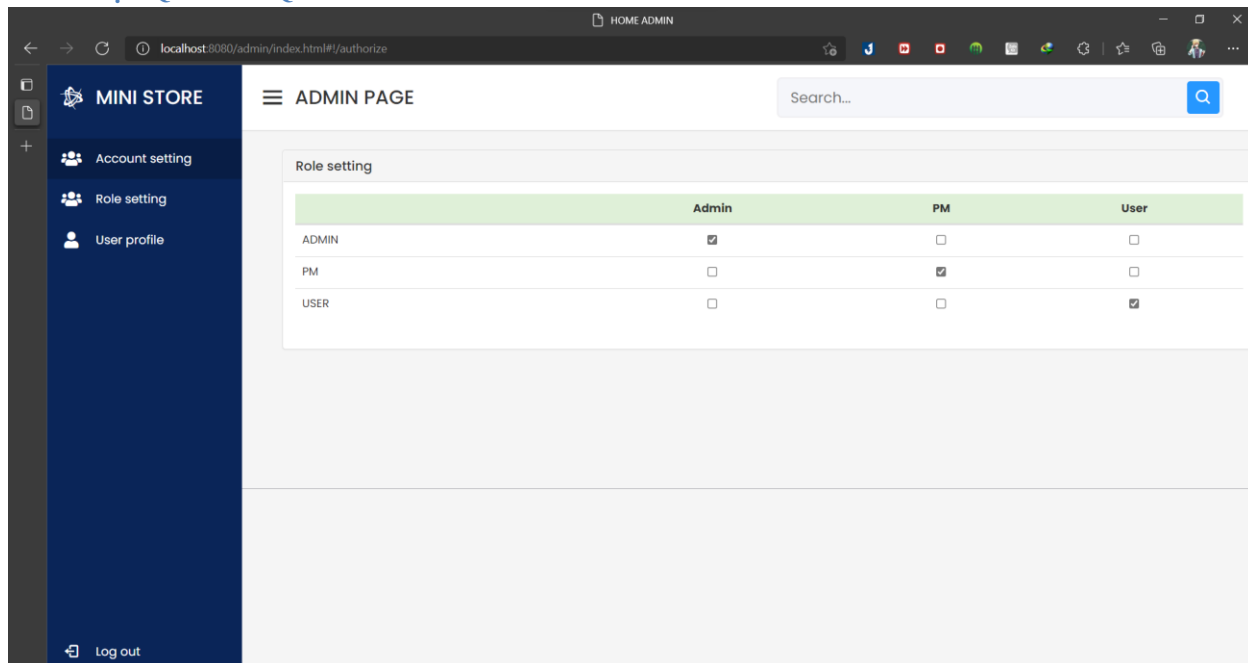
### 3.3.3 GIAO DIỆN ĐĂNG NHẬP



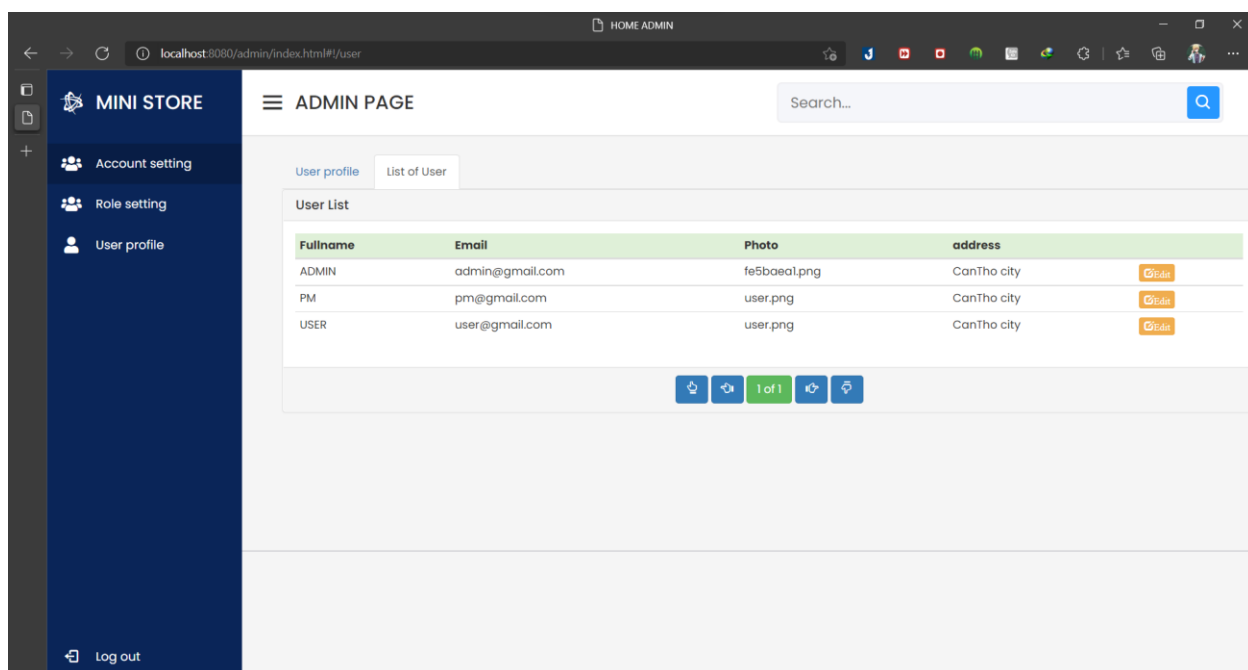
### 3.3.4 GIAO DIỆN QUẢN LÝ TÀI KHOẢN



### 3.3.5 GIAO DIỆN QUẢN LÝ QUYỀN



### 3.3.6 GIAO DIỆN QUẢN LÝ THÔNG TIN NGƯỜI DÙNG



## 4. THỰC HIỆN DỰ ÁN

### 4.1 TẠO CSDL VỚI MySQL

#### 4.1.1 CHI TIẾT CÁC BẢNG

##### 4.1.1.1 BẢNG ACCOUNT

Cấu trúc bảng

Tên Cột	Kiểu	Ràng buộc	Ghi chú
<b>Username</b>	VARCHAR(50)	PK, NOT NULL	
<b>Password</b>	VARCHAR(50)	NOT NULL	
<b>Fullname</b>	VARCHAR(50)	NOT NULL	
<b>Email</b>	VARCHAR(50)	NOT NULL	
<b>Photo</b>	VARCHAR(50)	NOT NULL	
<b>address</b>	VARCHAR(100)	NULL	

### Mã lệnh tạo bảng

```
CREATE TABLE Accounts(
    Username VARCHAR(50) NOT NULL PRIMARY KEY,
    Password VARCHAR(50) NOT NULL,
    Fullname VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    Photo VARCHAR(50) NOT NULL,
    address VARCHAR(100) NULL
);
```

#### 4.1.1.2 BẢNG AUTHORITIES

##### Cấu trúc bảng

Tên Cột	Kiểu	Ràng buộc	Ghi chú
Id	INT	PK, NOT NULL, AI	
Username	VARCHAR(50)	FK, NOT NULL	
RoleId	VARCHAR(10)	FK, NOT NULL	

### Mã lệnh tạo bảng

```
CREATE TABLE Authorities(
    Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    Username VARCHAR(50) NOT NULL,
    RoleId VARCHAR(10) NOT NULL,
    FOREIGN KEY (RoleId) REFERENCES Roles(Id),
    FOREIGN KEY (Username) REFERENCES Accounts(Username)
);
```

#### 4.1.1.3 BẢNG CATEGORIES

##### Cấu trúc bảng

Tên Cột	Kiểu	Ràng buộc	Ghi chú
Id	CHAR(4)	PK, NOT NULL	
Name	VARCHAR(50)	NOT NULL	

### Mã lệnh tạo bảng

```
CREATE TABLE Categories(
    Id CHAR(4) NOT NULL PRIMARY KEY,
    NAME VARCHAR(50) NOT NULL
);
```

#### 4.1.1.4 BẢNG ORDER

##### Cấu trúc bảng

Tên Cột	Kiểu	Ràng buộc	Ghi chú
---------	------	-----------	---------

Id	BIGINT	PK, NOT NULL, AI	
Username	VARCHAR(50)	FK, NOT NULL	
Createdate	DATETIME	NOT NULL	
Address	VARCHAR(100)	NOT NULL	

#### Mã lệnh tạo bảng

```
CREATE TABLE Orders (
    Id BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    Username VARCHAR (50) NOT NULL,
    CreateDate DATETIME NOT NULL,
    Address VARCHAR (100) NOT NULL,
    FOREIGN KEY (Username) REFERENCES Accounts(Username)
);
```

#### 4.1.1.5 BẢNG PRODUCT

##### Cấu trúc bảng

Tên Cột	Kiểu	Ràng buộc	Ghi chú
Id	INT	PK, NOT NULL, AI	Id product
Name	VARCHAR(50)	NOT NULL	
Image	VARCHAR(50)	NOT NULL	
Price	FLOAT	NOT NULL	
Createdate	DATE	NOT NULL	
Available	BIT	NOT NULL	
CategoryId	CHAR(4)	FK, NOT NULL	Id category

#### Mã lệnh tạo bảng

```
CREATE TABLE Products (
    Id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    Name VARCHAR (50) NOT NULL,
    Image VARCHAR (50) NOT NULL,
    Price FLOAT NOT NULL,
    CreateDate DATE NOT NULL,
    Available BIT NOT NULL,
    CategoryId CHAR (4) NOT NULL,
    FOREIGN KEY (CategoryId) REFERENCES Categories(Id)
);
```

#### 4.1.1.6 BẢNG ORDERDETAILS

##### Cấu trúc bảng

Tên Cột	Kiểu	Ràng buộc	Ghi chú
Id	BIGINT	PK, NOT NULL, AI	Id OrderDetails
OrderId	BIGINT	FK, NOT NULL	Order id
ProductId	INT	FK, NOT NULL	Product id
Price	FLOAT	NOT NULL	

Quantity	INT	NOT NULL	
----------	-----	----------	--

#### Mã lệnh tạo bảng

```
CREATE TABLE OrderDetails(
    Id BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    OrderId BIGINT NOT NULL,
    ProductId INT NOT NULL,
    Price FLOAT NOT NULL,
    Quantity INT NOT NULL,
    FOREIGN KEY (OrderId) REFERENCES Orders(Id),
    FOREIGN KEY (ProductId) REFERENCES Products(Id)
);
```

#### 4.1.1.7 BẢNG ROLES

##### Cấu trúc bảng

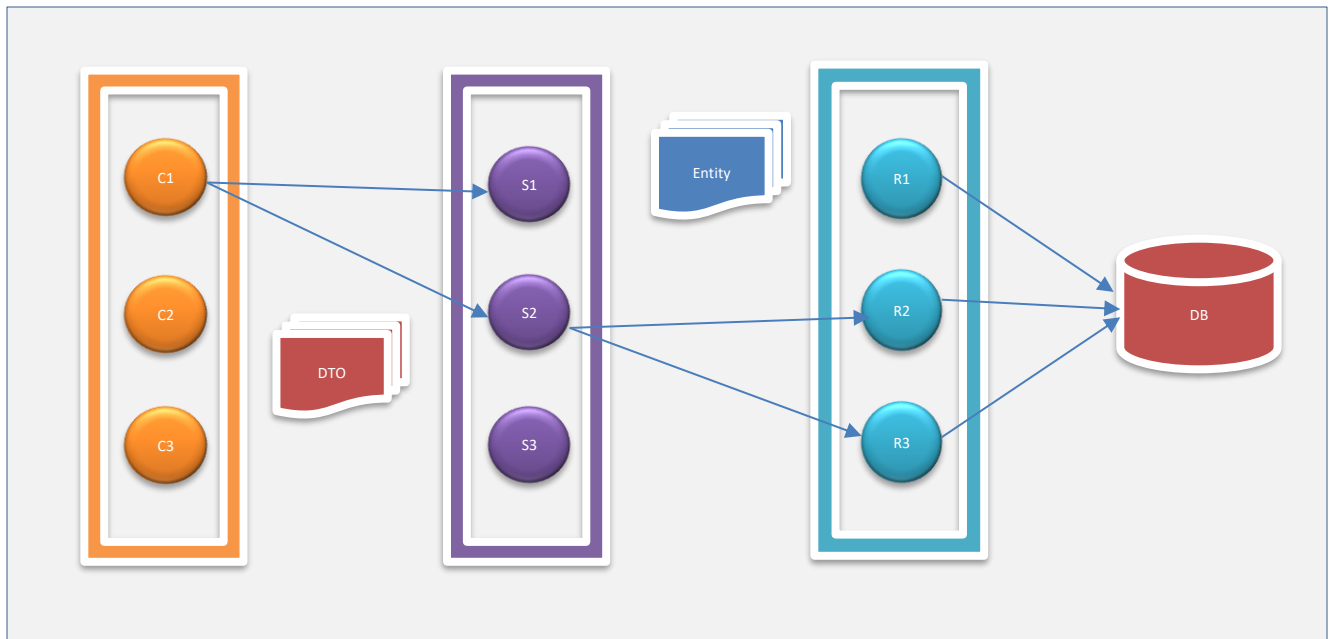
Tên Cột	Kiểu	Ràng buộc	Ghi chú
Id	VARCHAR(10)	PK, NOT NULL	
Name	VARCHAR(50)	NOT NULL	

#### Mã lệnh tạo bảng

```
CREATE TABLE Roles (
    Id VARCHAR (10) NOT NULL PRIMARY KEY,
    NAME VARCHAR (50) NOT NULL
);
```

## 4.2 Lập trình CSDL

### 4.2.1 Mô hình tổ chức lập trình CSDL



**Controller**  
(Behaviors)

**Service**  
(Services)

**Repository**  
(DAOs)

**Database**  
(Tables)

#### 4.2.1.1 Entity class và DAO

##### 4.2.1.1.1 Class Diagram

- ❖ Các lớp Entity là lớp dùng để miêu tả các dữ liệu từ các bảng tương ứng trong CSDL.
- ❖ Các lớp DAO dùng để truy xuất CSDL.

##### 4.2.1.2.2 Account và AccountDAO

THÀNH PHẦN	MÔ TẢ
AccountDAO	Là lớp thực thể mô tả dữ liệu tài khoản.
AccountDAO	Là lớp DAO cụ thể hóa các hoạt động truy xuất CSDL với bảng AccountDAO <ul style="list-style-type: none"> <li>✓ E được cụ thể hóa là AccountDAO</li> <li>✓ K được cụ thể hóa là Integer</li> <li>✓ SQL là các câu lệnh thao tác dữ liệu trên bảng hóa đơn chi tiết.</li> </ul>

## 4.3 Lập trình nghiệp vụ

### 4.3.1 Chức năng quản lý người dùng

STT	Phương thức xử lý	Mô tả
-----	-------------------	-------



1	Initialize()	
2	Edit()	
3	Create()	Thêm mới tài khoản thông qua RestAPI “/rest/accounts/{id}”
4	Update()	Cập nhật thông tin tài khoản thông qua RestAPI “/rest/accounts/{id}”
5	Reset()	
6	Pager()	Phân trang

#### 4.3.2 Chức năng quản lý phân quyền

STT	Phương thức xử lý	Mô tả
1	Initialize()	
2	Edit()	
3	Create()	Thêm mới tài khoản thông qua RestAPI “/rest/accounts/{id}”
4	Update()	Cập nhật thông tin tài khoản thông qua RestAPI “/rest/accounts/{id}”
5	Reset()	
6	Pager()	Phân trang

## 5. Kiểm thử

Đính kèm trong file excel.

## 6. Đóng gói và triển khai

### 6.1 Sản phẩm phần mềm

TT	THÀNH PHẦN	MÔ TẢ
1	1 file MiniProject_Tridthpc00719	File resources
2	ministore.sql	Cơ sở dữ liệu
3	Readme.txt	Hướng dẫn cài đặt
4	Guide.docx	Hướng dẫn sử dụng

## 6.2 Hướng dẫn cài đặt

- ✓ Bước 1. Cài đặt hệ cơ sở dữ liệu mysql trong file tài nguyên, đặt mật khẩu kết nối là “1234”.
- ✓ Chạy ministore.sql để tạo cơ sở dữ liệu.

= HẾT =