|  |  |  |
| --- | --- | --- |
|  |  |  |
| **CAO ĐẲNG THỰC HÀNH**  **FPT** **POLYTECHNIC** |  | **CÔNG TY CỔ PHẦN R2S** |

|  |
| --- |
| *Cần Thơ, ngày 08 tháng 09 năm 2021* |

|  |
| --- |
| **Người thực hiện: LÊ HOÀNG PHI – PC00552**  **Số điện thoại: 0868 323 770**  **Email: philhpc00552@fpt.edu.vn** |

|  |
| --- |
| **MINI PROJECT REQUIREMENT** |

|  |
| --- |
|  |



MỤC LỤC

[1. Giới Thiệu Dự Án 2](#_Toc81908711)

[2. Phân Tích Yêu Cầu Khách Hàng 2](#_Toc81908712)

[2.1 Sơ Đồ Use case 2](#_Toc81908713)

[2.2 Sơ đồ triển khai và yêu cầu hệ thống 4](#_Toc81908714)

[2.2.1 Mô hình triển khai 4](#_Toc81908715)

[ Hệ quản trị CSDL MySQL và CSDL sẽ được cài đặt và sử dụng trên một máy chuyên biệt. 4](#_Toc81908716)

[**2.2.2 Yêu Cầu Hệ Thống** 4](#_Toc81908717)

[3. Thiết kế ứng dụng 4](#_Toc81908718)

[3.1 mô hình công nghệ ứng dụng 4](#_Toc81908719)

[4](#_Toc81908720)

[3.2 Thực Thể 5](#_Toc81908721)

[**3.2.1** **Sơ đồ quan hệ thực thể** 5](#_Toc81908722)

[5](#_Toc81908723)

[**3.2.2** **Chi tiết thực thể** 6](#_Toc81908724)

[3.3 Giao diện 9](#_Toc81908725)

[4. Thực hiện dự án 9](#_Toc81908726)

[4.1 Tạo CSDL với MySQL 9](#_Toc81908727)

[**4.1.1** **Chi tiết các bảng** 9](#_Toc81908728)

[4.2 Lập trình CSDL 13](#_Toc81908729)

[**4.2.1** **Mô hình tổ chức lập trình CSDL** 13](#_Toc81908730)

[**4.2.2** **Entity class và DAO** 13](#_Toc81908731)

[4.2 Lập trình nghiệp vụ 14](#_Toc81908732)

[**4.2.3** **Chức năng thêm sản phẩm vào giỏ hàng và checkout giỏ hàng** 14](#_Toc81908733)

[**4.2.4** **Chức năng quản lý người dùng** 15](#_Toc81908734)

[**4.2.5** **Chức năng quán lý phân quyền** 15](#_Toc81908735)

[5. Kiểm thử 15](#_Toc81908736)

[6. Đóng gói và triển khai 16](#_Toc81908737)

[6.1 Sản phẩm phần mềm 16](#_Toc81908738)

[6.2 Hướng dẫn cài đặt 16](#_Toc81908739)

# **Giới Thiệu Dự Án**

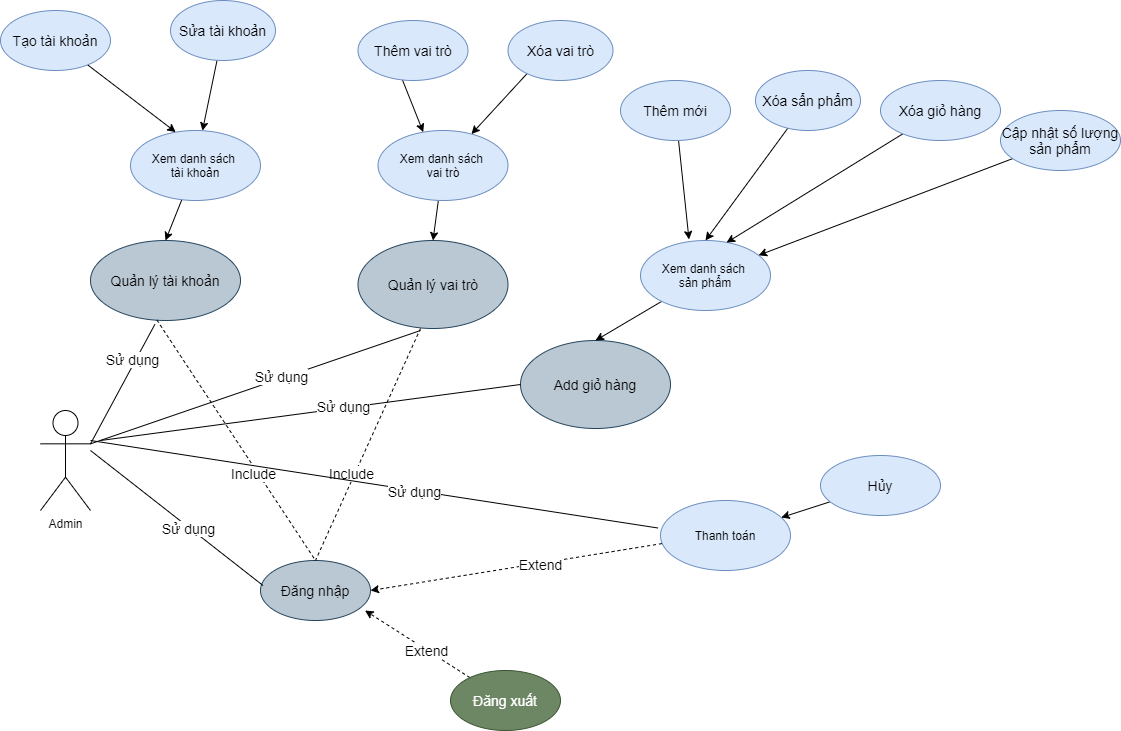
Thiết kế và phát triển website với các yêu cầu như sau:

* 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 tài khoản, chặn không cho người dùng đăng nhập.
* 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.
* Quản lý giỏ hàng: tài khoản thực hiện chọn các sản phẩm yêu thích vào giỏ hàng và tiến hành thanh toán với các sản phẩm có trong giỏ hàng.
* Yêu cầu về mô hình:
* Sử dụng lombook, 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.

# **Phân Tích Yêu Cầu Khách Hàng**

## 2.1 Sơ Đồ Use case

* + 1. Use Case đối tượng ADMIN



* + 1. Use Case đối tượng PM



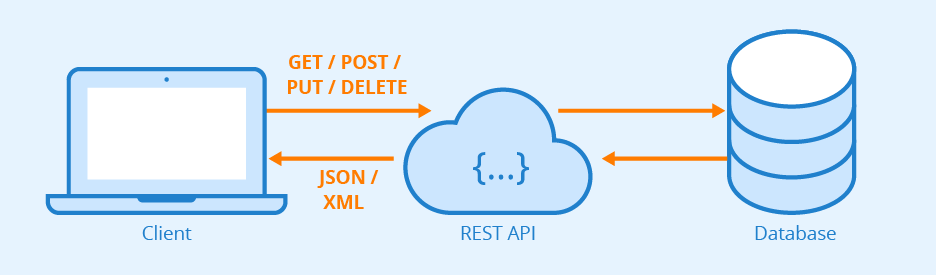
* + 1. Use Case đối tượng USER



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

### 2.2.1 Mô hình triển khai

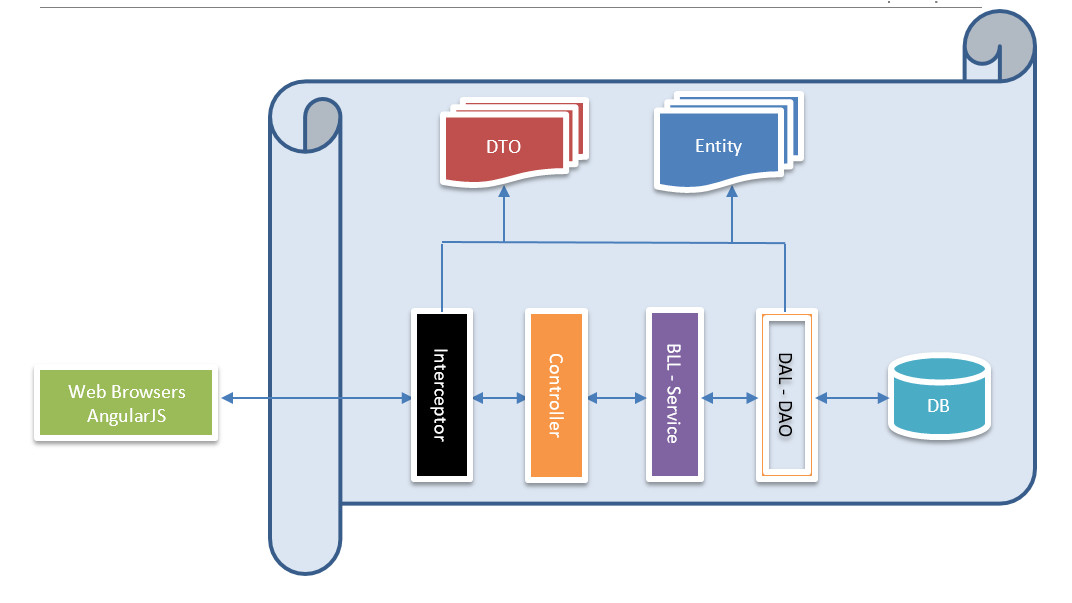
* Hệ quản trị CSDL MySQL và CSDL sẽ được cài đặt và sử dụng trên một máy chuyên biệt.



### **2.2.2 Yêu Cầu Hệ Thống**

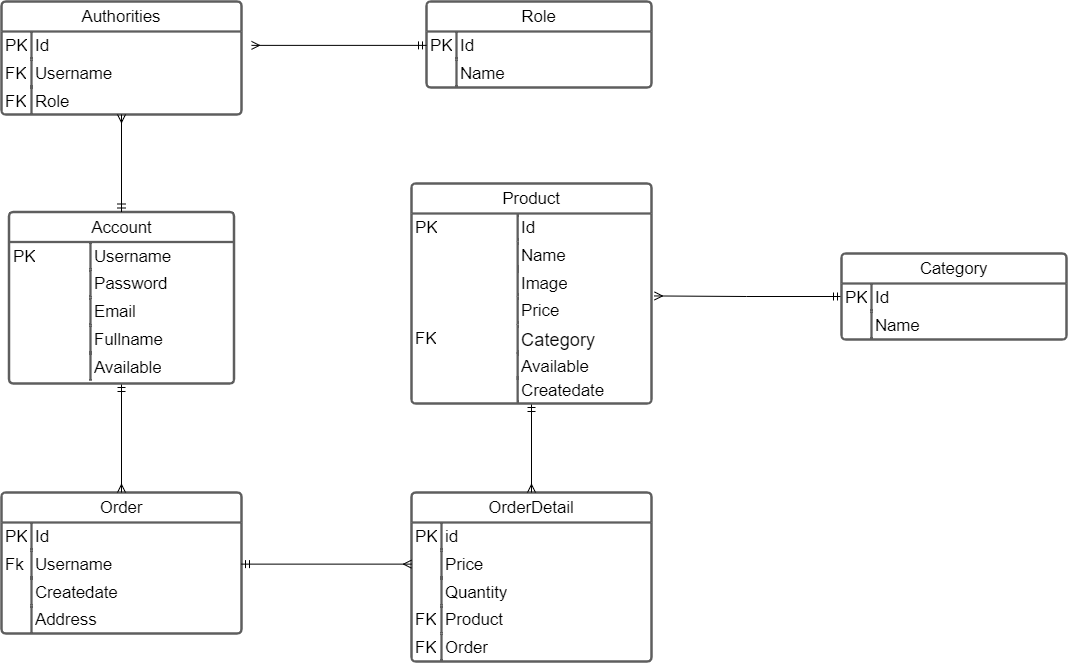
* Hệ đều hành : tối thiểu Windows 10
* Hệ quản trị CSDL : MySQL
* Ram: tối thiểu 4GB.

1. **Thiết kế ứng dụng**
   1. **mô hình công nghệ ứng dụng**

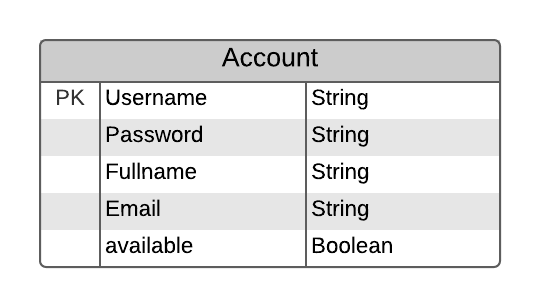


Trong đó:

* @Entity
* DAO extends JpaRepository (Data Access Layer (DAL))
* @Service (Business Logic Layer(BLL))
* @Controller, @RestController
* DTO (Data Transfer Object)
  1. **Thực Thể**
     1. **Sơ đồ quan hệ thực thể**

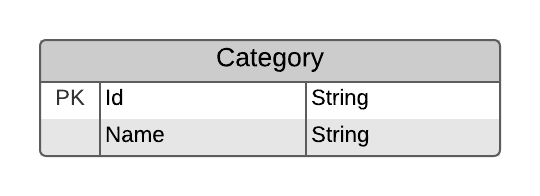
****

* + 1. **Chi tiết thực thể**
       1. **Thực thể Account**

****

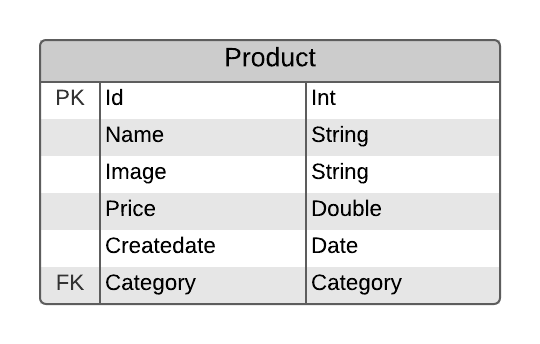
|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Kiểu Dữ Liệu | Mô Tả |
| Username | String | User name |
| Password | String | Pass word |
| Fullname | String | Full name |
| Available | Boolean |  |

* + - 1. **Thực thể Categories**

****

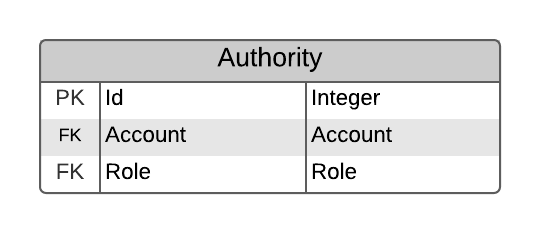
|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Kiểu Dữ Liệu | Mô Tả |
| Id | String | Id category |
| Name | String | Name category |

* + - 1. **Thực thể Products**

****

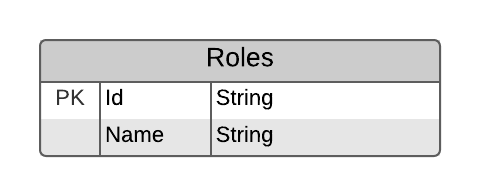
|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Kiểu Dữ Liệu | Mô Tả |
| Id | String | Id product |
| Name | String | Name product |
| Image | String |  |
| Price | Double |  |
| Createdate | Date | Create Date |
| Available | Boolean |  |
| Category | Category | Object category |

* + - 1. **Thực thể loại Authorities**

****

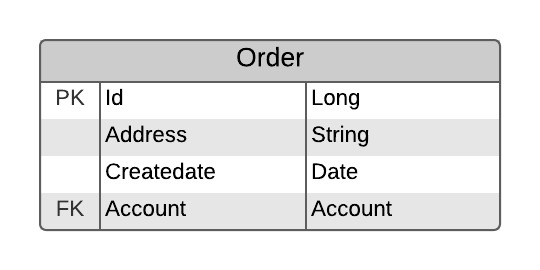
|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Kiểu Dữ Liệu | Mô Tả |
| Id | Integer | Id product |
| Account | Account | Object account |
| Role | Role | Object role |

* + - 1. **Thực thể Roles**

****

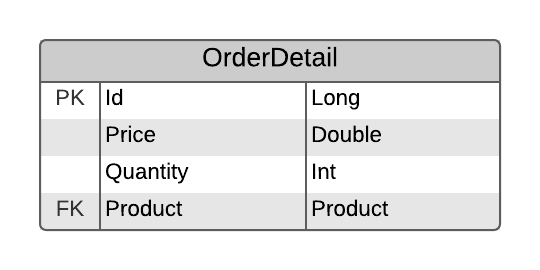
|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Kiểu Dữ Liệu | Mô Tả |
| Id | String | Id roles |
| Name | String | Name role |

* + - 1. **Thực thể Orders**

****

|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Kiểu Dữ Liệu | Mô Tả |
| Id | Long | Id order |
| Addesss | String |  |
| Createdate | Date | Create date |
| Account | Account | Object account |

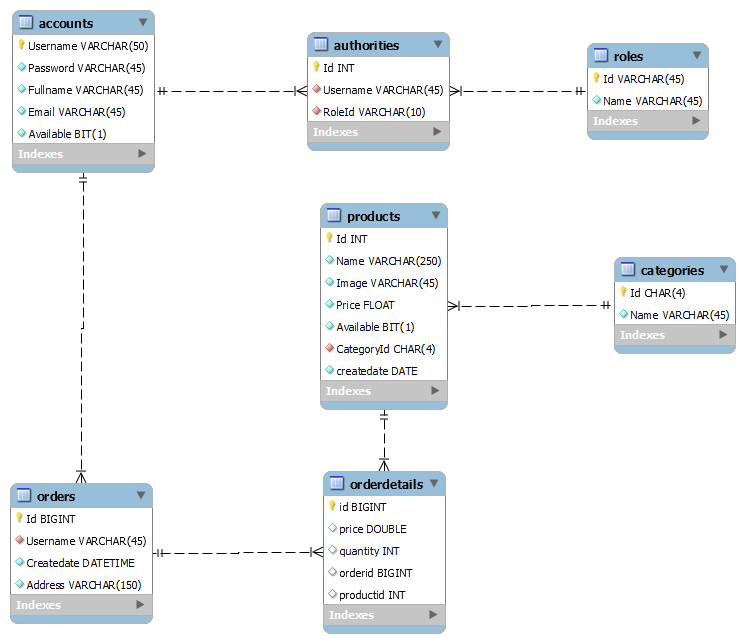
* + - 1. **Thực thể OrderDetails**

****

|  |  |  |
| --- | --- | --- |
| Thuộc Tính | Kiểu Dữ Liệu | Mô Tả |
| Id | Long | Id order detail |
| Price | String |  |
| Quantity | Date |  |
| Product | Product | Object Product |

* 1. **Giao diện**
     1. **Sơ đổ tổ chức giao diện**
     2. **Giao diện cửa sổ chính**
     3. **Các giao diện hổ trợ khác**

1. **Thực hiện dự án**
   1. **Tạo CSDL với MySQL**
      1. **Database Diagram**

****

* + 1. **Chi tiết các bảng**

#### **Bảng Account**

#### Cấu trúc bảng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Cột | Kiểu | Ràng buộc | Ghi chú |
| Username | VARCHAR(50) | PK, NOT NULL | User name |
| Password | VARCHAR(45) | NOT NULL | Pass word |
| Fullname | VARCHAR(45 | NOT NULL | Full name |
| Email | VARCHAR(45) | NOT NULL |  |
| Availible | BIT | NOT NULL |  |

#### Mã lệnh tạo bảng

|  |
| --- |
| CREATE TABLE `accounts` (  `Username` varchar(50) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Password` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Fullname` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Email` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Available` bit(1) NOT NULL,  PRIMARY KEY (`Username`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8\_unicode\_ci |

#### **Bảng Authorities**

#### Cấu trúc bảng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Cột | Kiểu | Ràng buộc | Ghi chú |
| Id | INT | PK, NOT NULL, AI | Id authorities |
| Username | VARCHAR(45) | FK, NOT NULL | User name |
| RoleId | VARCHAR(10) | FK, NOT NULL | Id role |

#### Mã lệnh tạo bảng

|  |
| --- |
| CREATE TABLE `authorities` (  `Id` int NOT NULL AUTO\_INCREMENT,  `Username` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `RoleId` varchar(10) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  PRIMARY KEY (`Id`),  UNIQUE KEY `UKhtk0tjw6uf67w0vkhy5n3j1oq` (`Username`,`RoleId`),  KEY `RoleId` (`RoleId`),  CONSTRAINT `authorities\_ibfk\_1` FOREIGN KEY (`Username`) REFERENCES `accounts` (`Username`),  CONSTRAINT `authorities\_ibfk\_2` FOREIGN KEY (`RoleId`) REFERENCES `roles` (`Id`)  ) ENGINE=InnoDB AUTO\_INCREMENT=16 DEFAULT CHARSET=utf8mb3 COLLATE=utf8\_unicode\_ci |

#### **Bảng Categories**

#### Cấu trúc bảng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Cột | Kiểu | Ràng buộc | Ghi chú |
| Id | CHAR(4) | PK, NOT NULL | Id category |
| Name | VARCHAR(45) | NOT NULL | Name category |

#### Mã lệnh tạo bảng

|  |
| --- |
| CREATE TABLE `categories` (  `Id` char(4) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci NOT NULL,  `Name` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  PRIMARY KEY (`Id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8\_unicode\_ci |

#### **Bảng Order**

#### Cấu trúc bảng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Cột | Kiểu | Ràng buộc | Ghi chú |
| Id | CHAR(4) | PK, NOT NULL, AI | Id order |
| Username | VARCHAR(45) | FK, NOT NULL | Name category |
| Createdate | DATETIME | NOT NULL | Create date |
| Address | VARCHAR(150) | NOT NULL |  |

#### Mã lệnh tạo bảng

|  |
| --- |
| CREATE TABLE `orders` (  `Id` bigint NOT NULL AUTO\_INCREMENT,  `Username` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Createdate` datetime NOT NULL,  `Address` varchar(150) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  PRIMARY KEY (`Id`),  KEY `Username` (`Username`),  CONSTRAINT `orders\_ibfk\_1` FOREIGN KEY (`Username`) REFERENCES `accounts` (`Username`)  ) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8mb3 COLLATE=utf8\_unicode\_ci |

#### **Bảng Product**

#### Cấu trúc bảng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Cột | Kiểu | Ràng buộc | Ghi chú |
| Id | INT | PK, NOT NULL, AI | Id product |
| Name | VARCHAR(250) | NOT NULL | Name product |
| Image | DATETIME | NOT NULL |  |
| Price | FLOAT | NOT NULL |  |
| Createdate | DATETIME | NOT NULL | Create Date |
| Available | BIT | NOT NULL |  |
| CategoryId | CHAR(4) | FK, NOT NULL | Id category |

#### Mã lệnh tạo bảng

|  |
| --- |
| CREATE TABLE `products` (  `Id` int NOT NULL AUTO\_INCREMENT,  `Name` varchar(250) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Image` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Price` float NOT NULL,  `Available` bit(1) NOT NULL,  `CategoryId` char(4) CHARACTER SET utf8 COLLATE utf8\_unicode\_ci NOT NULL,  `createdate` date NOT NULL,  PRIMARY KEY (`Id`),  KEY `CategoryId` (`CategoryId`),  CONSTRAINT `products\_ibfk\_1` FOREIGN KEY (`CategoryId`) REFERENCES `categories` (`Id`)  ) ENGINE=InnoDB AUTO\_INCREMENT=35 DEFAULT CHARSET=utf8mb3 COLLATE=utf8\_unicode\_ci |

#### **Bảng OrderDetails**

#### Cấu trúc bảng

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

#### Mã lệnh tạo bảng

|  |
| --- |
| CREATE TABLE `orderdetails` (  `id` bigint NOT NULL AUTO\_INCREMENT,  `price` double DEFAULT NULL,  `quantity` int DEFAULT NULL,  `orderid` bigint DEFAULT NULL,  `productid` int DEFAULT NULL,  PRIMARY KEY (`id`),  KEY `FKj4gc2ja2otvwemf4rho2lp3s8` (`orderid`),  KEY `FKaltatpxipsjtcih4d1h6bn0xr` (`productid`),  CONSTRAINT `FKaltatpxipsjtcih4d1h6bn0xr` FOREIGN KEY (`productid`) REFERENCES `products` (`Id`),  CONSTRAINT `FKj4gc2ja2otvwemf4rho2lp3s8` FOREIGN KEY (`orderid`) REFERENCES `orders` (`Id`)  ) ENGINE=InnoDB AUTO\_INCREMENT=17 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci |

#### **Bảng Roles**

#### Cấu trúc bảng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Cột | Kiểu | Ràng buộc | Ghi chú |
| Id | VARCHAR(45) | PK, NOT NULL | Id role |
| Name | VARCHAR(45) | NOT NULL | Name role |

#### Mã lệnh tạo bảng

|  |
| --- |
| CREATE TABLE `roles` (  `Id` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  `Name` varchar(45) CHARACTER SET utf8 COLLATE utf8\_general\_ci NOT NULL,  PRIMARY KEY (`Id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3 COLLATE=utf8\_unicode\_ci |

* 1. **Lập trình CSDL**
     1. **Mô hình tổ chức lập trình CSDL**

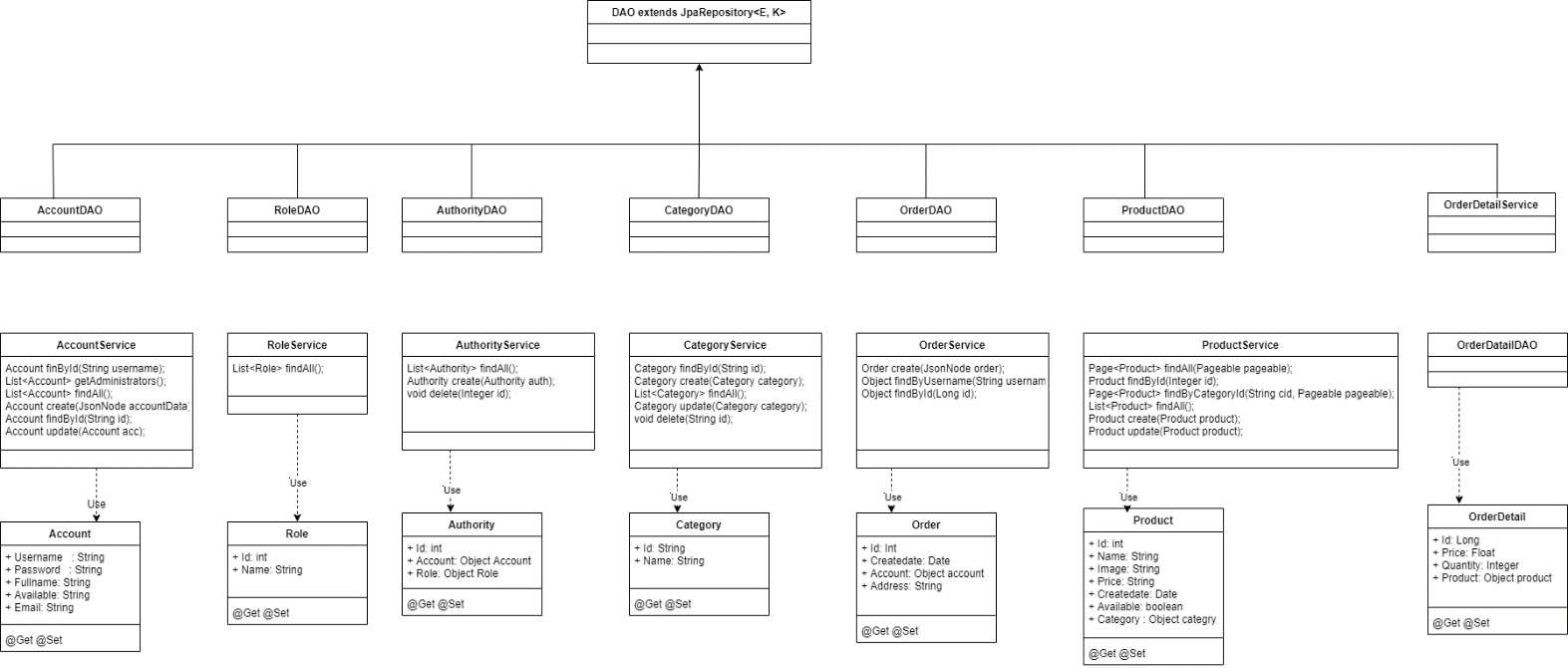
DB

Entity

DTO

|  |  |  |  |
| --- | --- | --- | --- |
| **Controller**  **(Behaviors)** | **Service**  **(Services)** | **Repository**  **(DAOs)** | **Database**  **(Tables)** |

* + 1. **Entity class và DAO**
       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.
* Các lớp Service thực hiện các nghiệp vụ và xử lý logic.



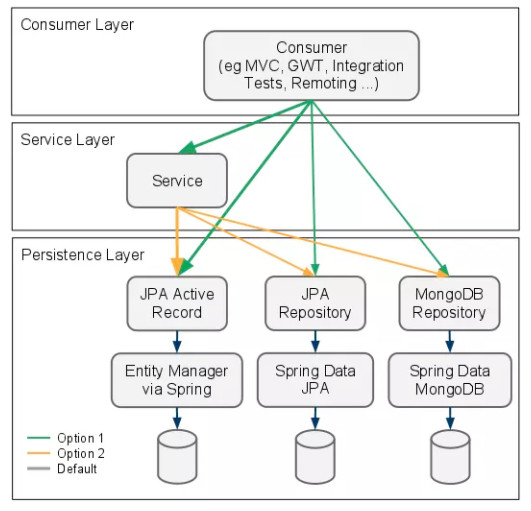
**4.2.1.2.2 Account và AccountDAO**

|  |  |
| --- | --- |
| THÀNH PHẦN | MÔ TẢ |
| **Account** | Là lớp thực thể mô tả dữ liệu hóa đơn chi tiết |
| **AccountDAO** | Là lớp DAO cụ thể hóa các hoạt động truy xuất CSDL với bảng   * E được cụ thể hóa là HoaDonChiTiet * K được cụ thể hóa là String |

## 

## **Lập trình nghiệp vụ**

* + 1. **Mô hình tổ chức**



Trong đó:

* Consumer Layer hay Controller: là tầng giao tiếp với bên ngoài và handler các request từ bên ngoài tới hệ thống.
* Service Layer: Thực hiện các nghiệp vụ và xử lý logic.
* Repository Layer: Chịu trách nhiệm giao tiếp với các DB, thiết bị lưu trữ, xử lý query và trả về các kiểu dữ liệu mà tầng Service yêu cầu.
  + 1. **Các chức năng chính** 
       1. **Chức năng thêm sản phẩm vào giỏ hàng và checkout giỏ hàng**

|  |  |  |
| --- | --- | --- |
| STT | Phương thức xử lý | Mô tả |
| 1 | Cart.add(id) | Thực hiển kiểm tra xem sản phẩm đã tồn tại chưa, nếu tồn tại thì “qty” tăng lên 1, cập nhật thông tin vào localstorage. Ngược lại lấy thông tin sản phẩm qua RestAPI “/rest/products/{id}” , lưu thông tin sản phẩm vừa tìm được lên localstorage. |
| 2 | Cart.remove(id) | Thực hiện xóa sản phẩm ra khỏi localstorage. |
| 3 | Cart.clear() | Xóa các sản phẩm được lưu trong localstorage. |
| 4 | Cart.count() | Tính số sản phấm có trong giỏ hàng. |
| 5 | Cart.amount() | Tính tổng tiển cần phải thanh toán trong giỏ hàng. |
| 6 | Cart.saveToLocalStorage() | Lưu sản phẩm được chọn vào localstorage. |
| 7 | Cart.loadFormLocalStorage() | Lấy thông tin được lưu ở localstorage lưu vào mảng có tên items. |

* + - 1. **Chức năng quản lý người dùng**

|  |  |  |
| --- | --- | --- |
| STT | Phương thức xử lý | Mô tả |
| 1 | Initialize() | Lấy dữ liệu từ RestAPI “/rest/accounts” gáng vào mảng items. |
| 2 | Edit(item) | Đưa giá trị item vào form. |
| 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() | Khởi tạo lại form với available: true. |
| 6 | Pager() | Khởi tạo phân trang |
| 7 | Pager.count() | Tính số trang được phân |
| 8 | Pager.first() | Quay về trang có vị trí là 0 |
| 9 | Pager.prev() | Giảm vị trí hiện tại xuống 1 |
| 10 | Pager.next() | Tăng vị trí hiện tại lên 1 |
| 11 | Pager.last() | Quay về vị trí cuối cùng của trang |

* + - 1. **Chức năng quán lý phân quyền**

|  |  |  |
| --- | --- | --- |
| STT | Phương thức xử lý | Mô tả |
| 1 | Scope.role; | Khởi tạo mảng role |
| 2 | Scope.admins | Khởi tạo mảng amin |
| 3 | Scope.authorities; | Khởi tạo mảng authorities |
| 4 | Initalize() | - Thực hiện lấy dữ liệu từ RestAPI “/rest/accounts” gáng vào mảng admins.  - Thực hiện lấy dữ liệu từ RestAPI “/rest/roles” gáng vào mảng roles.  - Thực hiện lấy dữ liệu từ RestAPI “/rest/authorities” gáng vào mảng authorities. |
| 5 | Authority\_of(acc, role) | Trả về true nếu authorities.account.username và authorities.role.id của authorities trùng với acc.account.username và roles.id tham số truyền vào và ngược lại. |
| 6 | Gant\_authority(authority) | - Thực hiện cấp quyền thông qua RestAPI “/rest/authorities”.  - Cập nhật dữ liệu cho mảng authorities thông qua pust data. |
| 7 | Gevoke\_authorities(authority) | Thu hồi quyền thông qua RestAPI “/rest/authorities/{id}”.  - Cập nhật dữ liệu cho mảng authorities thông qua pust data. |

# **Kiểm thử**

Đính kèm trong file excel.

# **Đóng gói và triển khai**

* 1. **Sản phẩm**

|  |  |  |
| --- | --- | --- |
| **TT** | **Thành phần** | **Mô tả** |
| 1 | philhpc00552\_R2S | File resources |
| 2 | test\_miniproject | Kiểm thử tự động |
| 3 | Test\_MiniProject.xlsx | TestCase |
| 4 | JavaStore.sql | Cơ sở dữ liệu |
| 5 | TaiLieuDuAn.doc | Tài liệu dự án |

* 1. **Hướng dẫn cài đặt**
* Bước 1. Cài đặt hệ cơ sở dữ liệu mysql trong file tài nguyên, đặt mật khẩu kết nối là “admin123”.
* Chạy SQL.sql để tạo CSDL JavaStore.

= HẾT =