**BÁO CÁO ASSIGNMENT PRJ301**

**NHÓM 6**

Vũ Thị Bích Phương

Nguyễn Xuân Thao

**Link bài code:**

https://drive.google.com/drive/folders/140dMRGA\_hRhp2a57u0XaFTfq8Nbsejyy

1. **Login/Signin - Logout:**
2. **Log In/ Sign In:**

Những Class cần tạo:

- *Trang hiển thị .jsp:*

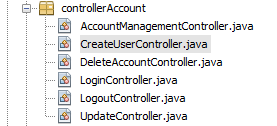






=> index.jsp là trang chính đi tới các trang login.jsp và createUser.jsp.

*- Servlet:*



=> Xử lý các yêu cầu.

*- MainController:*



=> Tất cả các yêu cầu đi qua MainController để đi tiếp tránh xâm nhập.

*- DAO cho Account:*



=> Check thông tin và vận chuyển data qua lại giữa sql với các dữ liệu bên ngoài.

*- Moldel Account:*



=> Khai báo các loại data có trong SQL.

*- lib.DBUtils:*



=> Thư viện giúp kết nối netbeans với SQL với các tên project tương ứng.

**\* Login:**

- Điều kiện: đang là khách (login\_user ==null) và đã có username và password.

Sơ đồ các luồng đi của Log In:

Ấn Login trên Trang “index.jsp”-> (qua MainController) đến “login.jsp” -> [input username, password] ấn nút Log in -> nhận data vào servlet (qua MainController) “LoginController” -> Check data **DAOAccount.Tools.getAccount(username, password) - > 3 trường hợp:**

+ data == null -> báo lỗi ngay trang “login.jsp”

+ data != null -> chuyển qua trang “index.jsp” của các role AD/US

+ Còn lại -> Báo bị ban ngay trang “login.jsp”

**\*Sign in:**

- Điều kiện: đang là khách (login\_user ==null) và chưa có username và password.

Sơ đồ các luồng đi của Sign In:

Ấn Sign in trên Trang “index.jsp”-> (qua MainController) đến “createUser.jsp” -> [input username, password, fullname, phone, address] ấn nút Sign in -> nhận data vào servlet (qua MainController) “CreateUserController” -> Check data  **DAOAccount.Tools.createUserAccount(new Account("", username, password, fullname, phone, address,"",""))**

**-> 2 trường hợp:**

+ data == null: báo lỗi ngay trang “createUser.jsp”

+data != null: Lưu data chuyển vào SQL và chuyển qua trang “login.jsp” để đăng nhập lại.

1. **Log Out:**

- Điều kiện: đã login (login\_user !=null).

Sơ đồ các luồng đi của Log Out:

Ấn nút Log out trên trang “index.jsp”-> (qua MainController) đến servlet “LogoutController” -> Lưu session cho remove “login\_user” và “cart” -> chuyển đến trang “index.jsp” với role = null.

1. **Management:**

- Điều kiện: login\_user.role == ‘AD’

1. **User management:**

\*Note: Chức năng **C**: Create user = Sign In (ở trên).

Chức năng **R**: Read = Show all User:

Tạo các trang:

- Servlet “AccountManagementController”:



- Trang hiển thị:



- Tạo DAOAccount.Tools.getAllAccount(search):





Sơ đồ các luồng đi của chức năng Show All User:

Login với role = ‘AD’ thì chuyển tới trang index.jsp-> Ấn nút Management chọn nút User Management trên trang “index.jsp” -> (qua MainController) đến servlet “AccountManagementController” -> check **DAOAccount.Tools.getAllAccount(search)** thông qua search user name.-> chuyển qua trang “accountManagement.jsp” show tất cả các Account có trong data SQL thông qua search (search == null thì show all acccount).

Chức năng **D**: Delete = Delete User:

- Tạo các servlet:





- Tạo **DAOAccount.Tools.deleteAccount(id):**



Sơ đồ các luồng đi của chức năng Delete User:

Tạo mỗi nút Delete bên cạnh mỗi account trong trang “accountManagement.jsp”:

Ấn nút Delete -> (qua MainController) đến servlet “DeleteAccountController” -> lưu session và check login\_user bằng id: Nếu id = id mà User đang login thì báo lỗi, còn lại -> tiến hành vào **DAOAccount.Tools.deleteAccount(id)** kiểm tra nếu có id đó trong SQL thì xóa dữ liệu trong SQL luôn

-> trả kết quả lại vào trang “accountManagement.jsp”.

Chức năng **U**: Update = Update User:

-Tạo sevlet:



- Tạo **DAOAccount.Tools.updateAccount(id,role,status):**



Sơ đồ các luồng đi của chức năng Update User:

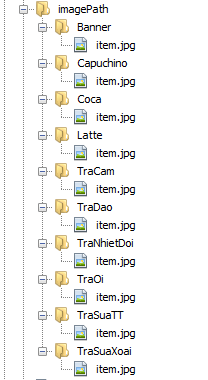
Tạo mỗi nút Update bên cạnh mỗi account trong trang “accountManagement.jsp”:

Thay đổi status (Banned/Active), role (AD/US) -> Ấn nút Update -> (qua MainController) đến servlet “UpdateAccountController” -> lưu session và check status hoặc role (vừa mới thay đổi) bằng id -> tiến hành vào **DAOAccount.updateAccount(id,role,status)** kiểm tra nếu có id đó trong SQL thì update dữ liệu trong SQL luôn -> trả kết quả lại vào trang “accountManagement.jsp”.

1. **Product management:**

Những Class cần tạo:

- *Thêm folder hình ảnh -> thêm các folder tên các product và bên trong đặt tên image là “item.jpg”.*

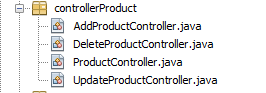


*- Trang hiển thị .jsp:*



=> index.jsp là trang chính đi tới trang “productManagement.jsp”.

*- Servlet:*



=> Xử lý các yêu cầu CRUD.

*- DAO cho Product:*



=> Check thông tin và vận chuyển data qua lại giữa sql với các dữ liệu bên ngoài.

*- Moldel Product:*





=> Khai báo các loại data có trong SQL.

* Chức năng **R**: Read = Show All Product:

- Tạo servlet:



- Tạo **DAOProduct.ToolProduct.getAllProduct(search)**:



Sơ đồ các luồng đi Show All Product:

Login với role = ‘AD’ thì chuyển tới trang index.jsp-> Ấn nút Management chọn nút Product Management trên trang “index.jsp” -> (qua MainController) đến servlet “ProductController” -> check **DAOProduct.ToolProduct.getAllProduct(search)** thông qua search product.-> chuyển qua trang “productManagement.jsp” show tất cả các Product có trong data SQL thông qua search (search == null thì show all product).

* Chức năng **C**: Create = Add Product:

- Tạo servlet:



- Tạo **DAOProduct.ToolProduct.addProduct(new Product("", pname, price, imagePath, "")**):



Sơ đồ các luồng đi Add Product:

Trong trang “productManagement.jsp” tạo các <input> (bao gồm: pname, price, imagePath) -> Ấn nút ADD -> (qua Main Controller) đến servlet “AddProductController” -> nếu data == null thì báo lỗi, ngược lại thì check **DAOProduct.ToolProduct.addProduct(new Product("", pname, price, imagePath, "")**)thông qua search pname -> Thêm data vào SQL -> đi qua “ProductController”-> trả kết quả vào trang “productManagement.jsp” + “index.jsp” và show các Product có trong data SQL thông qua search (search == null thì show all product).

* Chức năng **D**: Delete Product:

- Tạo servlet:



- Tạo **DAOProduct.ToolProduct.deleteProduct(id):**



Sơ đồ các luồng đi Delete Product :

Tạo mỗi nút Delete bên cạnh mỗi product trong trang “productManagement.jsp”:

Ấn nút Delete -> (qua MainController) đến servlet “DeleteProductController” -> lưu session -> tiến hành vào **DAOProduct.ToolProduct.deleteProduct(id)** kiểm tra nếu có id của product đó trong SQL thì xóa dữ liệu trong SQL luôn -> đi qua “ProductController” -> trả kết quả lại vào trang “productManagement.jsp”.

* Chức năng **U**: Update Product:

- Tạo servlet:



- Tạo **DAOProduct.ToolProduct.updateProduct(status, id):**



Sơ đồ các luồng đi của chức năng Update Product:

Tạo mỗi nút Update bên cạnh mỗi product trong trang “productManagement.jsp”:

Thay đổi status (Stocking/Sold Out) -> Ấn nút Update -> (qua MainController) đến servlet “UpdateProductController” -> lưu session và check status (vừa mới thay đổi) bằng id của product -> tiến hành vào **DAOProduct.ToolProduct.updateProduct(status, id)** kiểm tra nếu có id product đó trong SQL thì update dữ liệu trong SQL luôn -> đi qua “ProductController”-> trả kết quả lại vào trang “productManagement.jsp”.

1. **Order management (AD) và Order List (US):**

Những Class cần tạo:

-Trang JSP:



- Trang servlet:



- Model:



- Tool cho Order:

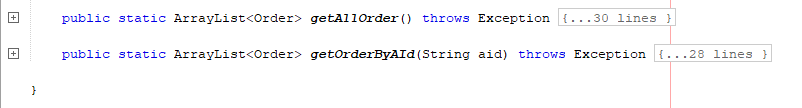


* Chức năng **R**: Read = Show All Order:

- Tạo servlet:



- Tạo **DAOOrder.OrderTool.getAllOrder() và DAOOrder.OrderTool.getOrderByAId(id)**:



Sơ đồ các luồng đi Show All Order:

Login với role = ‘AD’ thì chuyển tới trang index.jsp-> Ấn nút Management chọn nút

Order List History Management trên trang “index.jsp” -> (qua MainController) đến servlet “ShowOrderController” -> check trong SQL: aid = ‘ADMIN’ sẽ vào check **DAOOrder.OrderTool.getAllOrder()** -> trả kết quả là 1 List All các Order của tất cả User ở trang “orderList.jsp”

Login với role = ‘US’ thì chuyển tới trang index.jsp-> Ấn nút Order History List Management trên trang “index.jsp” -> (qua MainController) đến servlet “ShowOrderController” ->check trong SQL: aid = ‘USER’ sẽ vào check **DAOOrder.OrderTool.getOrderByAId(id)**-> trả kết quả là 1 list Order của riêng User đó ở trang “orderList.jsp”

* Chức năng **C**: Create = Create Order:

-Điều kiện: chỉ login\_user.role == ‘US’ mới tạo được.

- Tạo servlet Buy in controllerCart:





- Tạo **DAOOrder.OrderTool.createOrder(order)**:



Sơ đồ các luồng đi của chức năng Create Order:

Sau khi mua hàng, trong giỏ hàng (“cart.jsp”) User thực hiện ấn nút Order/Buy trong trang “cart.jsp”

-> (qua MainController) đến servlet “BuyController” -> lưu session và check data trong cart == null thì báo lỗi, còn ngược lại -> vào **DAOOrder.OrderTool.createOrder(order)** kiểm tra các thành phần của Order có tương ứng hay không và chuyển đổi dữ liệu như: String createDate = LocalDateTime.now(); description +=String.format("%d", cart.getQuantity(id)) + "-" + id + ","; .. -> Tạo dữ liệu cho Order trong SQL -> đi qua “ShowOrderController”-> trả kết quả vào trang “orderList.jsp”.

1. **Cart:**

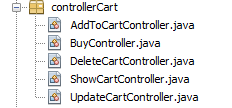
- Điều kiện: Khách (login\_user == null) và US (login\_user.role == ‘US’) sử dụng.

**Những Class cần tạo:**

-Trang JSP:



- Trang servlet:



- Model:



* Chức năng **C**: Create = Add to cart:

- Tạo servlet:



- Tạo **model.Cart.addItem(String id)**:





Sơ đồ các luồng đi Add to cart:

Trong trang “index.jsp” có các sản phẩm được đưa vào từ Product -> Ấn nút Add to Cart ở mỗi sản phẩm sẽ tạo Hashmap lên nút CART ở phía trên ( có number tăng số sản phẩm) -> Ấn nút CART -> (qua Main Controller) đến servlet “ShowCartController”-> trả kết quả add to cart qua trang “cart.jsp”.

* Chức năng **R**: Read = Show cart:

- Tạo servlet:



- Tạo **model.Cart.getItem()**:





Sơ đồ các luồng đi Show cart:

Sau khi ấn nút CART -> (qua Main Controller) đến servlet “ShowCartController” -> Lưu session và lấy session đó check: Nếu cart == null thì list bị trống, ngược lại thì check các product có trong **DAOProduct.ToolProduct.showProduct(id)** hay không. Nếu có -> thì làm **cart.getItem()** ->trả kết quả vào trang “cart.jsp”.

* Chức năng **D**: Delete cart:

- Tạo servlet:



- Tạo **model.Cart.deleteItem(id)**:





Sơ đồ các luồng đi Delete cart:

Tạo mỗi nút Delete bên cạnh mỗi product trong Cart ở trang “cart.jsp”:

Ấn nút Delete -> (qua MainController) đến servlet “DeleteCartController” -> lưu session -> tiến hành vào **cart.deleteItem(id)** kiểm tra và xóa khỏi CART -> đi qua “ShowCartController” -> trả kết quả lại vào trang “cart.jsp”.

* Chức năng **U**: Update cart:

- Tạo servlet:



- Tạo **model.Cart.deleteItem(id)**:





Sơ đồ các luồng đi Update cart:

Tạo mỗi nút Update bên cạnh mỗi product trong Cart ở trang “cart.jsp”:

Tăng Quantity từ 1 trở lên -> Ấn nút Update -> (qua MainController) đến servlet “UpdateCartController” -> lưu session -> tiến hành vào **cart.updateItem(id)** kiểm tra và thay đổi Quantity trong CART -> đi qua “ShowCartController” -> trả kết quả lại vào trang “cart.jsp”.