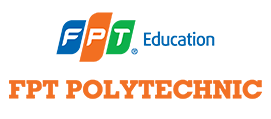
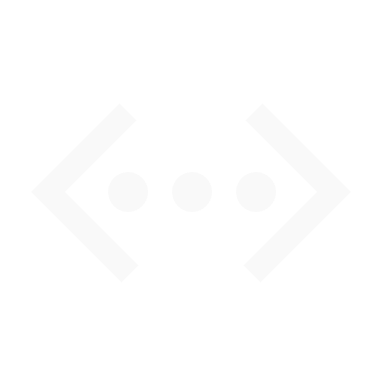
**TRƯỜNG CAO ĐẲNG FPT POLYTECHNIC**

**ỨNG DỤNG PHẦN MỀM**

**∙ • 🙣 - SOF2041 - 🙡 • ∙**

****

**THỰC TẬP**

**BÁO CÁO DỰ ÁN THỰC TẬP XƯỞNG**

**Mã môn**: SOF2041Ứng dụng phần mềm

**Giáo viên HD**: *thầy* Trần Quang Bình

**Mã SV**: Ps18107 **Tên SV**: Ngô Duy Hòa

Polytechnic – HCM – K16\_3 – Kì 4

**Tóm tắt nội dung**

[**I.** **PHÂN TÍCH** 4](#_Toc84918987)

[1. Hiện trạng 4](#_Toc84918988)

[2. Yêu cầu hệ thống 4](#_Toc84918989)

[**✓ Yêu cầu chức năng nghiệp vụ** 4](#_Toc84918990)

[**✓ Yêu cầu về bảo mật** 4](#_Toc84918991)

[**✓ Yêu cầu về môi trường công nghệ** 4](#_Toc84918992)

[3. Use case 4](#_Toc84918993)

[a. Use case hệ thống quản lý đào tạo 4](#_Toc84918994)

[b. Use case quản lý trạng thái đơn 5](#_Toc84918995)

[c. Use case quản lý tài khoản 6](#_Toc84918996)

[d. Use case quản lý tổng hợp thống kê 6](#_Toc84918997)

[e. Use case quản lý giỏ hàng 7](#_Toc84918998)

[f. Use case quản lý chi tiết đơn hàng 7](#_Toc84918999)

[g. Use case quản lý sản phẩm 8](#_Toc84919000)

[**II.** **THIẾT KẾ** 9](#_Toc84919001)

[1. Mô hình triển khai 9](#_Toc84919002)

[2. Thiết kế cơ sở dữ liệu 9](#_Toc84919003)

[a. Sơ đồ quan hệ thực thể 9](#_Toc84919004)

[b. Thiết kế chi tiết các thực thể 10](#_Toc84919005)

[3. Thiết kế giao diện 11](#_Toc84919006)

[a. Sơ đồ tổ chức giao diện 11](#_Toc84919007)

[b. Thiết kế giao diện cho các chức năng nghiệp vụ 12](#_Toc84919008)

[c. Trong lập trình giao diện thường qui ước tiếp đầu ngữ giao diện như sau: 12](#_Toc84919009)

[**III.** **THỰC HIỆN VIẾT MÃ** 24](#_Toc84919010)

[1. Viết mã tạo cơ sở dữ liệu 24](#_Toc84919011)

[a. Tạo cơ sở dữ liệu, database 24](#_Toc84919012)

[b. SQL truy vấn và thao tác bảng 26](#_Toc84919013)

[c. Các thủ tục lưu và tổng hợp thống kê 28](#_Toc84919014)

[2. Lập trình JDBC 29](#_Toc84919015)

[a. Lớp hỗ trợ 29](#_Toc84919016)

[b. Model class - Các lớp mô tả dữ liệu thực thể 29](#_Toc84919017)

[c. Package **ClassesDAO** thao tác dữ liệu với XJdbc 38](#_Toc84919018)

[d. Package **viewWeb** 49](#_Toc84919019)

[e. Package **Model** chứa các phương thức hỗ trợ tiện ích 52](#_Toc84919020)

[f. Package **view** các form hiển thị và thao tác dữ liệu 68](#_Toc84919021)

[**IV.** **KIỂM THỬ** 110](#_Toc84919022)

[1. Kiểm thử đăng nhập 110](#_Toc84919023)

[2. Kiểm thử đổi mật khẩu khi quên 111](#_Toc84919024)

[3. Kiểm thử đổi mật khẩu 111](#_Toc84919025)

[4. Kiểm thử quản lý sản phẩm 111](#_Toc84919026)

[5. Kiểm thử quản lý tài khoản 112](#_Toc84919027)

[6. Kiểm thử quản lý giỏ hàng 112](#_Toc84919028)

[7. Kiểm thử quản lý chi tiết đơn hàng 112](#_Toc84919029)

[8. Kiểm thử quản lý trạng thái đơn 113](#_Toc84919030)

[9. Kiểm thử chức năng thống kê biểu đồ 114](#_Toc84919031)

[10. Kiểm thử chức năng trợ giúp khác hàng 114](#_Toc84919032)

[**V.** **ĐÓNG GÓI VÀ TRIỂN KHAI** 114](#_Toc84919033)

[1. Tạo file jar và các tài nguyên đóng gói 114](#_Toc84919034)

[2. Tạo file exe ứng dụng 115](#_Toc84919035)

[3. Sử dụng Inno setup đóng gói cài đặt 122](#_Toc84919036)

1. **PHÂN TÍCH**
2. Hiện trạng

* *Xậy dựng website quản lý và bán hàng chuyên các món đặc sản miền trung, trong đó người dùng có thể đăng ký tài khoản làm đối tác kinh doanh buôn bán hoặc vận chuyển đơn hàng.*
* *Website phải có thể hiện thân thiện phù hợp với từng loại thiết bị mà khách hàng sử dụng và sẽ phải được đưa lên mạng để hoạt động với yêu cầu hoạt động tốc độ nhanh.*

1. Yêu cầu hệ thống

### **✓ Yêu cầu chức năng nghiệp vụ**

* Chủ website có giao diện quản lý
* Admin quản lý giao đơn và tài khoản
* Người bán quản lý sản phẩm đăng tải
* Người giao hàng cập nhật trạng thái và mô tả đơn
* Người mua có thể mua sản phẩm và thay dổi trong thời gian chờ
* Thống kê số đơn đã bán, số sản phẩm đã bán được và doanh thu được tính theo hàng tháng, hiển thị bằng sơ đồ.

### **✓ Yêu cầu về bảo mật**

* Chủ website nắm toàn bộ quyền hệ thống
* Admin có quyền quản lý tài khoản và đơn hàng
* Shipper chỉ được cập nhật trạng thái và mô trạng thái đơn
* Buyer được phép mua bàng và thanh toán, vẫn có thể chỉnh sủa trong thời gian chờ đợi chuẩn bị hàng
* Seller có quyền đăng tải sản phẩm của họ

### **✓ Lựa chọn môi trường và công nghệ**

* Môi trường triển khai hệ thống trên Tomcat server
* Hệ quản trị CSDL SQL Server 2019
* Các công nghệ sử dụng:
  + Html – Css – Bootstrap 5 - MomenJS
  + Javascript – Angular JS – ChartJS
  + Thymeleaf – Loombok – Spring-boot – Mô hình MVC

1. Use case

*Dựa vào yêu cầu hệ thống của đề bài* ***Đặc Sản Miền Trung****, có sơ đồ* ***use case*** *như dưới đây:*

Diagram

Description automatically generated

1. **ĐẶC TẢ RSR**
   1. Giới thiệu dự án

Dự án “đặc sản miền trung” là trang web kinh doanh online các loại thực phẩm miền trung, người dùng có thể là khách hàng đặt mua sản phẩm hoặc là đối tác đăng ký kinh doanh trên hệ thống của đặc sản miên trung.

* 1. Yêu cầu khái quoát

Toàn bộ sản phẩm hiển thị trên phần mềm là đặc sản miền trung,

* + - Giao diện thể hiện tính phù hợp với nhiều loại thiết bị phù hợp
    - Sản phẩm hiển thị theo từng loại
    - Trang chủ hiển thị sản phẩm **hot**, **khuyến mãi** và **bán chạy**
    - Trang chủ có phần tin tức riêng, logo các đối tác.
    - Hoạt động tốc độ nhanh
  1. Yêu cầu bảo mật
     + Những ai đăng nhập hệ thống mới có sử dụng hệ thống
     + Biểu đồ đơn hàng do admin tiếp nhận và xử lý
     + Shipper tiếp nhận đơn hàng và cập nhật biểu đồ đơn
     + Cần có giao diện để quản lý cho mỗi loại người dùng
  2. Một số yêu cầu khác
     + Website cần có tính năng cơ bản như **tìm kiếm** theo tên, **sắp xếp** theo giá cả.
     + Phần thống kê hiển thị theo tháng các thông tin sau:
       1. Số sản phẩm bán được.
       2. Số đơn hàng đã bán.
       3. Tổng doanh thu.

1. **THIẾT KẾ**
   1. Sơ đồ hoạt động theo mô hình MVC hệ thống

Pages controller vs Restfull đều nằm trong lớp Control của hệ thống, Trong đó mode sẽ làm việc trực tiếp với database, dữ liệu từ lớp control sẽ trả về cho client

Diagram

Description automatically generated

* 1. Sơ đồ hoạt động

Diagram

Description automatically generated

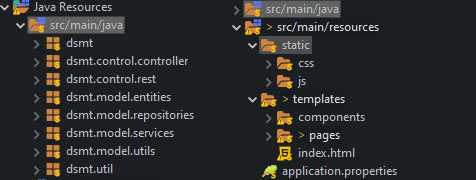
* 1. Thiết kế cơ sở dữ liệu
     + 1. Diagram

          Description automatically generated Mô tả các trường dữ liệu

|  |  |  |
| --- | --- | --- |
| **ACCOUNTS** | | |
|  | **tên cột** | **mô tả** |
| khóa chính | username | tên đăng nhập |
|  | password | mật khẩu |
|  | email | email |
|  | name | tên người dùng |
|  |  |  |
| **ROLES** | | |
|  | **tên cột** | **mô tả** |
| khóa chính | id | mã phân quyền |
|  | name | tên phân quyền |
|  |  |  |
| **AUTHORITIES** | | |
|  | **tên cột** | **mô tả** |
| khóa chính | role\_id, account\_id | mã phân quyền |
| khóa phụ | và tài khoản được phân quyền |
|  |  |  |
| **CATEGORIES** | | |
|  | tên cột | mô tả |
| khóa chính | id | mã phân loại |
|  | name | tên phân loại sản phẩm |
|  |  |  |
| **PRODUCTS** | | |
|  | **tên cột** | **mô tả** |
| khóa chính | id | mã sản phẩm |
| khóa phụ | category\_id | phân loại sản phẩm |
| account\_id | tài khoản đăng tải |
|  | cosPrice | giá gốc |
|  | proPrice | khuyến mãi |
|  | name | tên sản phẩm |
|  | descript | mô tả sản phẩm |
|  |  |  |
| **PRODUCT\_IMAGES** | | |
| khóa chính | product\_id | mã sản phẩm |
| khóa chính | image | hình ảnh sản phẩm |
| khóa phụ | product\_id | mã liên kết sản phẩm |
|  |  |  |
| **COMMENTS** | | |
|  | **tên cột** | **mô tả** |
| khóa chính | account\_id, product\_id | tài khoản và sản phẩm |
| khóa phụ | account\_id | tài khoản mình luận |
| product\_id | sản phẩm được bình luận |
|  | regTime | mốc thời gian bình luận |
|  | descript | bình luận |
|  |  |  |
| **ORDERS** | | |
|  | tên cột | mô tả |
| khóa chính | id | mã đơn hàng |
| khó phụ | account\_id | tài khoản order |
|  | regTime | thời gian đặt hàng |
|  | Address | Địa chỉ giao nhận hàng |
|  |  |  |
| **ORDER\_DETAILS** | | |
|  | tên cột | mô tả |
| khóa chính | order\_id, product\_id | Sản phẩm trong đơn hàng |
| khóa phụ | order\_id | đơn hàng |
| product\_id | sản phẩm được đặt mua |
|  | oldPrice | giá gốc tại thời điểm mua |
|  | quantity | số lượng mua |
|  |  |  |
| **ORDER\_STATUS** | | |
|  | tên cột | mô tả |
| khóa chính | order\_id | Sản phẩm trong đơn hàng |
| khóa phụ | order\_id | trạng thái đơn hàng |
| account\_id | shipper giao hàng |
|  | status | trạng thái đơn hàng |
|  | descript | mô tả trạng thái |
|  |  |  |

1. **THỰC HIỆN VIẾT MÃ**

# Cấu trúc dự án mã



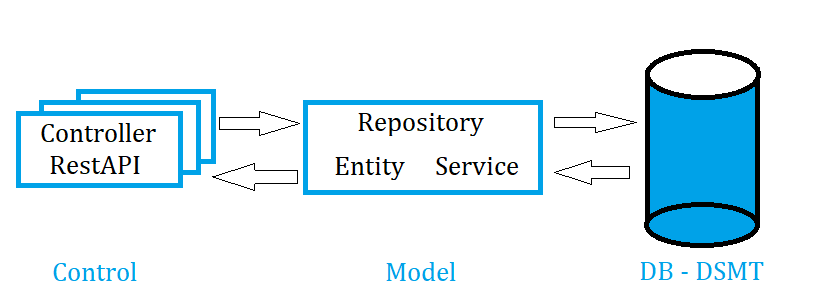
1. Viết mã tạo cơ sở dữ liệu
   1. Tạo cơ sở dữ liệu, database

|  |
| --- |
| USE master |
| GO |
| IF EXISTS (SELECT name FROM sys.databases WHERE name = 'DSMT') DROP DATABASE DSMT |
| GO |
| CREATE DATABASE DSMT |
| GO |
| USE DSMT |
| GO |
|  |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'ACCOUNTS') DROP TABLE ACCOUNTS |
| GO |
| CREATE TABLE ACCOUNTS (  username varchar(20) primary key,  password varchar(30) not null,  email varchar(50) unique,  name nvarchar(50)  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'ROLES') DROP TABLE ROLES |
| GO |
| CREATE TABLE ROLES (  id varchar (10) primary key,  name nvarchar(20)  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'AUTHORITIES') DROP TABLE AUTHORITIES |
| GO |
| CREATE TABLE AUTHORITIES (  account\_id varchar(20) foreign key references ACCOUNTS(username)  on update cascade on delete cascade not null,  role\_id varchar (10) foreign key references ROLES(id) not null,  primary key (account\_id, role\_id)  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'CATEGORIES') DROP TABLE CATEGORIES |
| GO |
| CREATE TABLE CATEGORIES (  id char(8) primary key,  name nvarchar(30) not null  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'PRODUCTS') DROP TABLE PRODUCTS |
| GO |
| CREATE TABLE PRODUCTS (  id int identity primary key,  cosPrice float,  proPrice float,  name nvarchar(30),  descript nvarchar(255),  category\_id char(8) foreign key references CATEGORIES(id) not null,  account\_id varchar(20) foreign key references ACCOUNTS(username)  on update cascade on delete cascade not null,  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'PRODUCT\_IMAGES') DROP TABLE PRODUCT\_IMAGES |
| GO |
| CREATE TABLE PRODUCT\_IMAGES (  product\_id int foreign key references PRODUCTS(id)  on update cascade on delete cascade not null,  image nvarchar(255) not null,  primary key(product\_id, image)  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'COMMENTS') DROP TABLE COMMENTS |
| GO |
| CREATE TABLE COMMENTS (  account\_id varchar(20) foreign key references ACCOUNTS(username)  on update cascade on delete cascade not null,  product\_id int foreign key references PRODUCTS(id),  descript nvarchar(255),  regTime datetime default getDate(),  primary key(account\_id, product\_id)  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'ORDERS') DROP TABLE ORDERS |
| GO |
| CREATE TABLE ORDERS (  id int identity primary key,  address nvarchar(80),  regTime datetime default getDate(),  account\_id varchar(20) foreign key references ACCOUNTS(username)  on update cascade on delete cascade not null,  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'ORDER\_DETAILS') DROP TABLE ORDER\_DETAILS |
| GO |
| CREATE TABLE ORDER\_DETAILS (  order\_id int foreign key references ORDERS(id)  on update cascade on delete cascade not null,  product\_id int foreign key references PRODUCTS(id),  oldPrice float default 0,  quantity int,  ); |
|  |
| IF EXISTS (SELECT name FROM sys.tables WHERE name = 'ORDER\_STATUS') DROP TABLE ORDER\_STATUS |
| GO |
| CREATE TABLE ORDER\_STATUS (  order\_id int foreign key references ORDERS(id)  on update cascade on delete cascade not null,  account\_id varchar(20) foreign key references ACCOUNTS(username),  status int default 0,  descript nvarchar(150),  primary key(order\_id)  ); |

* 1. Các thủ tục lưu và tổng hợp thống kê

|  |
| --- |
| USE DSMT  GO  -- +++++++++++++++++++++++++++++++++++ SIZE - MỐC THỜI GIAN ĐĂNG TẢI  IF EXISTS (SELECT name FROM sys.views WHERE name = 'VIEW\_SO\_RANGE')      DROP VIEW VIEW\_SO\_RANGE  GO  CREATE VIEW VIEW\_SO\_RANGE AS      SELECT          COUNT(id) as 'length',          CASE              WHEN MIN(regTime) IS NULL THEN GETDATE()              ELSE MIN(regTime)          END as 'st',          CASE              WHEN MAX(regTime) IS NULL THEN GETDATE()              ELSE MAX(regTime)              END as 'et'      FROM ORDERS  GO  SELECT \* FROM VIEW\_SO\_RANGE  GO  -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT QUANTITY AND TOTAL OF PRODUCT BY STATUS  /\*      PROC\_SS\_TIME: STATISTIC ORDER SOLD        @Top: số lượng rows | mặc định 1000      @at: GROUP THEO (1 - YEAR | 2 - MONTH | 3 - DAY)      @status: trạng thái đơn hàng (1 | 2 | 3 | 4)      @start: thời gian bắt đầu | mặc định min regTime      @end: thời gian kết thúc | mặc định max regTime      @desc: sắp xếp theo số lượng  \*/  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_SS\_TIME')      DROP PROCEDURE PROC\_SS\_TIME  GO  CREATE PROC PROC\_SS\_TIME      @Top int, @at int, @status int, @start datetime, @end datetime, @desc bit  AS BEGIN      IF @start IS NULL SET @start = (SELECT st FROM VIEW\_SO\_RANGE)      IF @end IS NULL SET @end = (SELECT et FROM VIEW\_SO\_RANGE)      IF @status IS NULL SET @status = 3      IF @at IS NULL SET @at = 2      SET @at = @at\*3      -- SELECT INTO TEMPORARY TABLE      SELECT          SUBSTRING(CONVERT(varchar(8), regTime, 2), 0 , @at) as 'month',          SUM(d.quantity) as 'quantity',          SUM(d.oldPrice) as 'total' INTO #TEMP      FROM ORDER\_DETAILS d          INNER JOIN ORDER\_STATUS s ON s.order\_id=d.order\_id          INNER JOIN ORDERS o ON o.id = d.order\_id          WHERE regTime BETWEEN @start and @end AND s.status >= @status      GROUP BY SUBSTRING(CONVERT(varchar(8), regTime, 2), 0 , @at)      -- SELECT DATA TO RETURN      IF @desc IS NULL SELECT TOP(ISNULL(@top, 1000)) \* FROM #TEMP      ELSE IF @desc=0 SELECT  TOP(ISNULL(@top, 1000)) \* FROM #TEMP o ORDER BY o.total      ELSE SELECT TOP(ISNULL(@top, 1000)) \* FROM #TEMP o ORDER BY o.total DESC  END  GO      -- @at IS NULL THEN SET @at default = 2      -- @status IS NULL THEN SET @status default = 3      EXEC PROC\_SS\_TIME NULL, 2, 2, '2021-6-1', NULL, NULL  GO  -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT ORDER BY TIME  /\*      PROC\_SO\_TIME: CONTENT UPLOAD BY TIME        @Top: số lượng rows | mặc định 1000      @start: thời gian bắt đầu | mặc định min regTime      @end: thời gian kết thúc | mặc định max regTime      @at: Chọn theo 1(YEAR) | 2(MONTH) | 3(DAY)      @desc: sắp xếp theo số lượng  \*/  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_SO\_TIME')      DROP PROCEDURE PROC\_SO\_TIME  GO  CREATE PROC PROC\_SO\_TIME      @Top int, @start datetime, @end datetime, @at int, @desc bit  AS BEGIN      IF @start IS NULL SET @start = (SELECT st FROM VIEW\_SO\_RANGE)      IF @end IS NULL SET @end = (SELECT et FROM VIEW\_SO\_RANGE)      DECLARE @CUT\_AT TINYINT = 3\*@at      -- SELECT INTO TEMPORARY TABLE      SELECT          SUBSTRING(CONVERT(varchar(8), regTime, 2), 0 , @CUT\_AT) as 'month',          SUM(d.quantity) as quantity INTO #TEMP      FROM ORDERS o          INNER JOIN ORDER\_DETAILS d ON o.id=d.order\_id      WHERE regTime BETWEEN @start AND @end      GROUP BY SUBSTRING(CONVERT(varchar(8), regTime, 2), 0 , @CUT\_AT)      -- SELECT DATA TO RETURN      IF @desc IS NULL SELECT TOP(ISNULL(@top, 1000)) \* FROM #TEMP      ELSE IF @desc=0 SELECT  TOP(ISNULL(@top, 1000)) \* FROM #TEMP o ORDER BY o.quantity      ELSE SELECT TOP(ISNULL(@top, 1000)) \* FROM #TEMP o ORDER BY o.quantity DESC  END  GO  EXEC PROC\_SO\_TIME 12, null, '2022', 2, null  GO  GO  -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT ACCOUNT BY ROLE ID  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_ACCOUNT\_BY\_ROLE')      DROP PROC PROC\_ACCOUNT\_BY\_ROLE  GO  CREATE PROCEDURE PROC\_ACCOUNT\_BY\_ROLE      @role\_id varchar(10)  AS BEGIN      SELECT \* FROM ACCOUNTS      WHERE username IN (          SELECT account\_id FROM AUTHORITIES          WHERE role\_id = @role\_id      )  END  GO  EXEC PROC\_ACCOUNT\_BY\_ROLE 'SHIPPER'    -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT ORDER BY STATUS  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_ORDER\_ORDERS')      DROP PROC PROC\_ORDER\_ORDERS  GO  CREATE PROCEDURE PROC\_ORDER\_ORDERS      @status int  AS BEGIN      IF @status IS NULL          SELECT \* FROM ORDERS          WHERE id NOT IN(SELECT order\_id FROM ORDER\_STATUS)      ELSE SELECT \* FROM ORDERS o          INNER JOIN ORDER\_STATUS s          ON s.order\_id=o.id WHERE s.status = @status  END  GO  EXEC PROC\_ORDER\_ORDERS null  -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT ORDER BY ACCOUNT\_ID  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_ORDER\_STATUS\_A')      DROP PROC PROC\_ORDER\_STATUS\_A  GO  CREATE PROCEDURE PROC\_ORDER\_STATUS\_A      @account\_id varchar(20)  AS BEGIN      SELECT s.\* FROM dbo.ORDER\_STATUS s      INNER JOIN ORDERS o ON o.id=s.order\_id      GROUP BY s.account\_id, s.descript, s.order\_id, s.status, o.account\_id      HAVING o.account\_id = @account\_id  END  GO  EXEC PROC\_ORDER\_STATUS\_A 'buyer1'  GO  -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT ORDER BY PRODUCT\_ID  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_ORDER\_STATUS\_P')      DROP PROC PROC\_ORDER\_STATUS\_P  GO  CREATE PROCEDURE PROC\_ORDER\_STATUS\_P      @product\_id int  AS BEGIN      SELECT s.\* FROM ORDERS o      INNER JOIN ORDER\_STATUS s ON s.order\_id=o.id      WHERE s.order\_id IN (          SELECT d.order\_id FROM ORDER\_DETAILS d          WHERE d.product\_id=@product\_id      ) GROUP BY s.account\_id, s.descript, s.order\_id, s.status  END  GO  EXEC PROC\_ORDER\_STATUS\_P 1  -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT DISCOUNT PRODUCT  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_DIS\_PRODUCT')      DROP PROC PROC\_DIS\_PRODUCT  GO  CREATE PROCEDURE PROC\_DIS\_PRODUCT      @top int  AS BEGIN      SELECT TOP (ISNULL(@top, 1000)) p.\*, SUM(o.quantity) as 'quantity'      FROM PRODUCTS p INNER JOIN ORDER\_DETAILS o          ON p.id = o.product\_id      GROUP BY id, cosPrice, proPrice, name, descript, category\_id, account\_id      ORDER BY SUM(o.quantity) DESC  END  GO  EXEC PROC\_DIS\_PRODUCT 10  -- ++++++++++++++++++++++++++++++++++++++++++++++ SELECT TOP PRODUCT  IF EXISTS (SELECT name FROM sys.procedures WHERE name = 'PROC\_TOP\_PRODUCT')      DROP PROC PROC\_TOP\_PRODUCT  GO  CREATE PROCEDURE PROC\_TOP\_PRODUCT      @top int  AS BEGIN      SELECT TOP(ISNULL(@top, 1000)) p.\*, COUNT(\*) as 'quantity' FROM ORDER\_DETAILS d          INNER JOIN ORDERS o ON o.id = d.order\_id          INNER JOIN PRODUCTS p ON p.id = d.product\_id      GROUP BY p.id, cosPrice, proPrice, name, descript, category\_id, p.account\_id      ORDER BY COUNT(\*) DESC  END  GO  EXEC PROC\_TOP\_PRODUCT 10 |

1. Lập trình JDBC
   1. Lớp hỗ trợ

*Trong dự án sử dụng công nghệ spring có mô hình như sau:*

**Trong đó:**

- **Control**: Trả về dữ liệu cho client

- **DB**: cơ sở dữ liệu

- **Model**: là các lớp mô tả dữ liệu theo cấu trúc Spring-MVC

*# tách thành các phần riêng biết giúp quản lý dự án tốt hơn, dễ bảo trì nâng cấp hơn.*

* 1. Mapping các thực thể

|  |
| --- |
| **Thực thể tài khoản** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @Entity(name = "ACCOUNTS")  public class Account {        @Id private String username;      private String password;      private String email;      private String name;        @ElementCollection(fetch = FetchType.EAGER) @Column(name = "role\_id") // phan quyen he thong      @CollectionTable(name = "AUTHORITIES", joinColumns = { @JoinColumn(name = "account\_id") })      private Set<String> roles;        public Account(String username) {          this.username = username;      }      public Account(String username, String password) {          this.username = username;          this.password = password;      }  } |
| **Bảng phân loại** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @Entity(name = "CATEGORIES")  public class Category {      // @formatter:off      @Id @Builder.ObtainVia @Column(length = 8)      private String id = Random.UppAndLow("", 8);      private String name;      // @formatter:on  } |
| **Bảng đánh giá** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @IdClass(Comment.class)  @Entity(name = "COMMENTS")  public class Comment implements Serializable {        private static final long serialVersionUID = 1985192875367924598L;        @Id private Integer product\_id;      @DateTimeFormat(pattern = "yyyy-MM-dd hh:mm:ss.SSS")      @ObtainVia @Column(name = "regtime")      @Id private Date regTime = new Date();        private String descript;        @JoinColumn(name = "account\_id", referencedColumnName = "username", insertable = false, updatable = false)      @ManyToOne @JsonIncludeProperties({"username", "email", "name"})      @Id private Account account;        public Comment(String account\_id, Integer product\_id) {          this.product\_id = product\_id;      }    } |
| **Bảng hóa đơn** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @Entity(name = "ORDERS")  public class Order {        // @formatter:off        @Id @GeneratedValue(strategy = GenerationType.IDENTITY)      private Integer id;      private String address;      @ObtainVia @Column(name = "regtime")      @DateTimeFormat(pattern = "yyyy-MM-dd hh:mm:ss.SSS")      private Date regTime = new Date();      private String account\_id;        @JoinColumn(name = "id", referencedColumnName = "order\_id")      @JsonIgnore @OneToOne(cascade = CascadeType.ALL)      private OrderStatus status;        @JoinColumn(name = "order\_id", insertable = false, updatable = false) @OneToMany      private List<OrderDetail> order\_details;      public Order(String account\_id) {          this.account\_id = account\_id;      }      public Order(String address, Date regTime, String account\_id) {          this.address = address;          this.regTime = regTime;          this.account\_id = account\_id;      }      // @formatter:on  } |
| **Bảng hóa đơn chi tiết** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @IdClass(OrderDetail.class)  @Entity(name = "ORDER\_DETAILS")  public class OrderDetail implements Serializable {      private static final long serialVersionUID = -3076955176935224191L;      @Id private Integer order\_id;      @Id private Integer product\_id;        // @formatter:off      @ObtainVia @Column(name = "oldprice")      private Float oldPrice = 0F;      @ObtainVia      private Integer quantity = 1;      @JoinColumn(name = "product\_id", referencedColumnName = "id", insertable = false, updatable = false)      @JsonIgnoreProperties({"category", "account\_id", "account"})      @ManyToOne private Product product;      // @formatter:on  } |
| **Bảng trạng thái hóa đơn** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @Entity(name = "ORDER\_STATUS")  public class OrderStatus implements Serializable {        private static final long serialVersionUID = 5686806031673720756L;      // @formatter:off      @Id private Integer order\_id;      private String account\_id;      @ObtainVia private Integer status = 0;      private String descript;        @JoinColumn(name = "order\_id", referencedColumnName = "id")      @OneToOne(cascade = CascadeType.ALL, orphanRemoval = true)      private Order order;        // @formatter:on  } |
| **Bảng sản phẩm** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @Entity(name = "PRODUCTS")  public class Product {        // @formatter:off        @Id @GeneratedValue(strategy = GenerationType.IDENTITY) private Integer id;      @Column(name = "cosprice") private float cosPrice;      @Column(name = "proprice") private float proPrice;      private String name;      private String descript;        @ManyToOne      private Category category;      @ManyToOne @JoinColumn(name="account\_id")      @JsonIncludeProperties({"username","name","email"})      private Account account;      @ElementCollection @Column(name = "image")      @CollectionTable (          name = "PRODUCT\_IMAGES",          joinColumns = @JoinColumn(name = "product\_id")      ) private List<String> images;      public Product(Integer id) {          this.id = id;      }      // @formatter:on  } |
| **Bảng phân quyền hệ thống** |
| @Data  @NoArgsConstructor  @AllArgsConstructor  @Entity(name = "ROLES")  public class Role {      // @formatter:off        @Id private String id;      private String name;        // @formatter:on  } |

* 1. Lập trình hệ thống

|  |
| --- |
| **Phần service** |
| public abstract class AbstractService<E, K> implements InterDAO<E, K> {        @Autowired protected JpaRepository<E, K> rep;      @Autowired protected HttpServletRequest req;      protected abstract K getId(E entity);        @Override      public List<E> getList() {          return rep.findAll();      }      @Override      public List<E> getList(Sort sort) {          return rep.findAll(sort);      }      @Override      public Page<E> getPage(Pageable pageable) {          return rep.findAll(pageable);      }      @Override      public Optional<E> getOptional(K id) throws IllegalArgumentException {          return rep.findById(id);      }      @Override      public <S extends E> S save(S entity) throws IllegalArgumentException {          K id = this.getId(entity);          Optional<E> optional = rep.findById(id);          if(optional.isEmpty()) {              return rep.save(entity);          } else throw new IllegalArgumentException(id+" đã tồn tại, không thể thêm mới.");      }      @Override      public <S extends E> S update(S entity) throws IllegalArgumentException {          K id = this.getId(entity);          Optional<E> optional = rep.findById(id);          if(optional.isPresent()) {              return rep.save(entity);          } else throw new IllegalArgumentException(id+" không tồn tại, không thể cập nhật.");      }      @Override      public void remove(K id) throws IllegalArgumentException {          rep.deleteById(id);      }      protected String getUser(String defaultUser) {          Principal p = req.getUserPrincipal();          return p == null ? defaultUser : p.getName();      }  } |
| **Phần controller** |
| /\*\*   \* @param <E> is entity   \* @param <K> is the type of entity's key   \*\*/  public abstract class AbstractRESTful<E, K> {      // @formatter:off      @Autowired protected InterDAO<E, K> dao;      @Autowired protected HttpServletRequest req;      @GetMapping({"","/{id}"}) // reading method to get data      public ResponseEntity<Object> getData(@PathVariable(required = false) K id) throws IllegalArgumentException {          if(id!=null) { // get one by id or get all entities              Optional<E> optional = dao.getOptional(id);              return optional.isPresent()                      ? ResponseEntity.ok(optional.get())                      : ResponseEntity.noContent().build();          } else return ResponseEntity.ok(dao.getList());      }        @PostMapping({"","/{id}"}) // Post method to create entity      public ResponseEntity<E> save(@RequestBody E entity) throws IllegalArgumentException {          return ResponseEntity.ok(dao.save(entity));      }        @PutMapping({"","/{id}"}) // Put method to update entity      public ResponseEntity<E> update(@RequestBody E entity) throws IllegalArgumentException  {          return ResponseEntity.ok(dao.update(entity));      }        @DeleteMapping({"/{id}"}) // Delete method to remove entity      public ResponseEntity<Void> delete(@PathVariable(required = false) K id) throws IllegalArgumentException {          if(id != null) {              dao.remove(id);              return ResponseEntity.ok().build();          } else return ResponseEntity.noContent().build();      }      // @formatter:on      protected String getUser(String defaultUser) {          Principal p = req.getUserPrincipal();          return p == null ? defaultUser : p.getName();      }  } |
| **Phần DAO** |
| public interface InterDAO<E, K> {      /\*\*       \* <h1 style="text-align: center; color: yellow; font-size: 2em; text-transform: uppercase;">       \*      find all entities       \* </h1>       \* @return all entities       \*/      public List<E> getList();      /\*\*       \* <h1 style="text-align: center; color: yellow; font-size: 2em; text-transform: uppercase;">       \*      find all entities by sort conditional       \* </h1>       \* @return all entities       \*/      public List<E> getList(Sort sort);        /\*\*       \* <h1 style="text-align: center; color: yellow; font-size: 2em; text-transform: uppercase;">       \*      find all entities by {@link Pageable} conditional       \* </h1>       \* @param pageable be the condition to find.       \* @return {@link Page}       \*/      public Page<E> getPage(Pageable pageable);      /\*\*       \* <h1 style="text-align: center; color: yellow; font-size: 2em; text-transform: uppercase;">       \*      find {@link Optional<E>} by id       \* </h1>       \* @param id is key       \* @return {@link Optional} entity -> {@link Optional#isPresent()} or {@link Optional#isEmpty()}       \* @throws IllegalArgumentException       \*/      public Optional<E> getOptional(K id) throws IllegalArgumentException;        /\*\*       \* <h1 style="text-align: center; color: yellow; font-size: 2em; text-transform: uppercase;">       \*      save entity to database       \* </h1>       \* @param entity to save       \* @return entity saved successfully       \*/      public <S extends E> S save(S entity) throws IllegalArgumentException;        /\*\*       \* <h1 style="text-align: center; color: yellow; font-size: 2em; text-transform: uppercase;">       \*      update entity into database       \* </h1>       \* @param entity to update       \* @return entity updated successfully       \* @throws IllegalArgumentException       \*/      public <S extends E> S update(S entity) throws IllegalArgumentException;      /\*\*       \* <h1 style="text-align: center; color: yellow; font-size: 2em; text-transform: uppercase;">       \*      delete data by {@link Id}       \* </h1>       \* @param id is key       \* @throws IllegalArgumentException       \*/      public void remove(K id) throws IllegalArgumentException;  } |

* 1. Lập trình giao diện

|  |
| --- |
| **Trang chủ** |
| <!DOCTYPE html>  <html xmlns="http://www.thymeleaf.org" th:fragment="dynamic(view)">  <head>  <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <!-- Bootstrap -->  <link      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"      rel="stylesheet"      integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"      crossorigin="anonymous">  <script      src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"      integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"      crossorigin="anonymous"></script>  <!-- Fontawesome -->  <link      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.1.2/css/all.min.css"      integrity="sha512-1sCRPdkRXhBV2PBLUdRb4tMg1w2YPf37qatUFeS7zlBy7jJI8Lf4VHwWfZZfpXtYSLy85pkm9GaYVYMfw5BC1A=="      crossorigin="anonymous" referrerpolicy="no-referrer" rel="stylesheet" />  <!-- Angular JS -->  <script      src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.7/angular.min.js"></script>  <script      src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.7/angular-route.min.js"></script>  <!-- Qrcode -->  <script      src="https://cdnjs.cloudflare.com/ajax/libs/qrcodejs/1.0.0/qrcode.min.js"      integrity="sha512-CNgIRecGo7nphbeZ04Sc13ka07paqdeTu0WR1IM4kNcpmBAUSHSQX0FslNhTDadL4O5SAGapGt4FodqL8My0mA=="      crossorigin="anonymous" referrerpolicy="no-referrer"></script>  <!-- CharJS  -->  <script      src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/3.9.1/chart.min.js"      integrity="sha512-ElRFoEQdI5Ht6kZvyzXhYG9NqjtkmlkfYk0wr6wHxU9JEHakS7UJZNeml5ALk+8IKlU6jDgMabC3vkumRokgJA=="      crossorigin="anonymous" referrerpolicy="no-referrer"></script>  <!-- MomentJS -->  <script      src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.4/moment.min.js"      crossorigin="anonymous" referrerpolicy="no-referrer"></script>  <script      src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.4/moment-with-locales.min.js"      integrity="sha512-42PE0rd+wZ2hNXftlM78BSehIGzezNeQuzihiBCvUEB3CVxHvsShF86wBWwQORNxNINlBPuq7rG4WWhNiTVHFg=="      crossorigin="anonymous" referrerpolicy="no-referrer"></script>  <!-- google fonts and style -->  <link rel="stylesheet"      href="https://fonts.googleapis.com/css?family=Merienda&effect=fire-animation">  <link rel="stylesheet"      href="https://fonts.googleapis.com/css?family=Pacifico&effect=3d">  <link rel="stylesheet"      href="https://fonts.googleapis.com/css?family=Merienda&effect=neon">  <!-- ----------------------------------- -->  <link th:href="@{/css/display.css}" rel="stylesheet" type="text/css">  <link th:href="@{/css/button.css}" rel="stylesheet" type="text/css">  <link th:href="@{/css/custom.css}" rel="stylesheet" type="text/css">  <title>CHI TIẾT ĐƠN HÀNG</title>  </head>  <body style="font-family: 'Merienda', monospace">      <div class="container-xxl border border-warning scroll-style position-relative" style="min-height: 100vh;">          <!-- TOGGLE BACKGROUND -->          <nav th:replace="@{components/\_toggle-background.htm}"></nav>          <!-- BUTTON SCROLL TO THE TOP -->          <nav th:replace="@{components/\_scroll-button.htm}"></nav>          <!-- NAVBAR -->          <nav th:replace="@{components/\_navbar.htm}"></nav>          <!-- CONTENTS -->          <main th:replace="${view}">REPLACE CONTENT AT HERE</main>          <!-- SHOW MESSAGE 2 -->          <div th:utext="${message}"></div>      </div>      <!-- Javascript files -->      <script type="text/javascript" th:src="@{/js/dsmt.js}"></script>      <script type="text/javascript" th:src="@{/js/support\_bootstrap.js}"></script>  </body>  </html> |
| **Javascript base “dsmt.js”** |
| /\*\*  \* @param host is protocol://domain:port  \* @param paths add at the end host  \* @returns url connect to server  \*/  function getLink(host, ...paths) {      return !paths instanceof String ? host.concat('/', paths) : host.concat('/', paths).replaceAll(',', '/')  }  /\*\*  \* @param column of entity to find in array  \* @param value of column  \* @param array to get index  \*/  function getIndex(column, value, array) {      // # 0==false      if (!value || !array) return -1;      if (column) {          for (let i = 0; i < array.length; i++)              if (array[i][column] == value) return i;      } else {          for (let i = 0; i < array.length; i++)              if (array[i] == value) return i;      } return -1;  }  // \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  /\*\*  \* <h3>show password form input</h3>  \* @param element is input[type="password"] => change type to password or text  \* @param eye is control to set type input  \*/  function showPass(element, eye) {      if (element.type == "text") {          element.type = "password";          if (eye) eye.setAttribute("class", "fa-solid fa-eye-slash");      } else {          element.type = "text";          if (eye) eye.setAttribute("class", "fa-solid fa-eye");      }  }  function setImage(input, toSet) {      if (input.files.length > 0) toSet.src = URL.createObjectURL(input.files[0]);  }  /\*\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   \* @param e element   \* @param cls class   \*   \*/  function setClass(e, name) {      let cls = e.getAttribute('class').trim();      e.setAttribute("class", cls.search(name ? name : 'show') > -1          ? cls.replace(` ${name ? name : "show"}`, '')          : (cls + ` ${name ? name : "show"}`)      );  }  // \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ QR Code  function generalQR(element, qrText, w, h) {      element.innerHTML = null;      new QRCode(element, {          text: qrText,          width: w ? w : 128,          height: h ? h : 128,          colorDark: "#000000",          colorLight: "#ff000000",          correctLevel: QRCode.CorrectLevel.H      })  }  this.generalQR(document.getElementById('qrCode'), "https://github.com/Duyhoa2002/Dac\_San\_Mien\_Trung");  // set images  function setImage(input, toSet) {      if(input.files.length>0) toSet.src = URL.createObjectURL(input.files[0]);  }  const LOCAL = {      get: (key) => JSON.parse(localStorage.getItem(key)),      set: (key, value) => localStorage.setItem(key, JSON.stringify(value)),      remove: (key) => localStorage.removeItem(key)  } |
| **Javascript làm việc với bootstrap “support\_bootstrap.js”** |
| // Create element fade on the screen  const fade = document.createElement("div");  fade.setAttribute('class', "offcanvas-backdrop fade show");  // \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  // Dropdown  function drdActive(p) {      setClass(p.getElementsByClassName("dropdown-menu")[0]);  }  // offcanvas  function canvasActive(id) {      var element = document.getElementById(id);      setClass(element);      // Set style visibility = boolean ? hidden : visible;      if (cls.search("show") > -1) {          document.body.removeChild(fade)          element.style.visibility = "hidden";      } else {          document.body.appendChild(fade);          element.style.visibility = "visible";      }  }  // Accordion  function acdActive(id) {      setClass(element, document.getElementById(id));  }  /\*\*  \* <h3>show acctive</h3>  \* @param p's param("this") of the class you want to active  \* @param clearClasses're any named of the class to inactive  \*/  function buttonActive(p, namedClasses) {      for (let e of document.getElementsByName(namedClasses)) {          var txt = e.getAttribute("class");          e.setAttribute("class", e!=p ? txt.replace(" active", '') :          txt.lastIndexOf(' active')>-1 ? txt : txt.concat(' active'));      }  }  function buttonClick(element, open) {      if (open) {          element.style.visibility = "visible";          cls = element.getAttribute("class");          element.setAttribute('class', (cls += " show"));      } else {          element.style.visibility = "hidden";          element.getAttribute("class").replace(" show", "");      }  }  const toast = new bootstrap.Toast(liveToast);  function showToast(title, content) {      var ltoast = document.getElementById('liveToast');      if(title) ltoast.getElementsByClassName('fil.$mes.t')[0].innerHTML=title;      if(content) ltoast.getElementsByClassName('fil.$mes.c')[0].innerHTML=content;      toast.show();  } |
| **Javascript angularJS các filter** |
| moment.locale('vi');  app.filter('dateRelative', () => {      return (date, format) => `${moment(date).format(format)} (${moment(date).fromNow()})`  })  app.filter('compare', () => {      return (array, col, value) => {          if(!array) return;          return array.filter(x => x[col] > value);      }  });  app.filter('sum', () => {      return (arr, cos, qty) => {          if(!arr) return 0;          let  amount = 0;          if(!arr.length) return amount;          for(e of arr) amount+=(eval(`e.${cos}`)\*eval(`e.${qty}`));          return amount;      }  })  app.filter('sort', () => {      return (arr, col, desc) => {          if(!arr) return          return desc              ? arr.sort((x,y) => eval(`x.${col}`).localeCompare(eval(`y.${col}`)))              : arr.sort((x,y) => eval(`y.${col}`).localeCompare(eval(`x.${col}`)));      }  })  app.filter('total', () => {      return (arr) => {          let total = 0;          for(x of arr) total+=x;          return total;      }  })  app.filter('map', () => {      return (arr, col) => arr ? arr.map(x=>x[col]) : map;  })  app.filter('hasAny', () => {      return (value, arr, col) => {          if(!arr) return;          for(x of arr) if(eval(`x.${col}`) == value)              return x;          return null;      }  }) |
| **Javascript angularJS controller** |
| app.controller('control', function($scope, $http) {      (async () => {          // get all partners          $http.get(getLink(serverIO,'rest/dir/documents/partners.json')).then(r => {              if(r.status==200) $scope.partners = r.data          }).catch(e => console.error(e.data ? e.data.message : e));          // get all partners          $http.get(getLink(serverIO,'rest/dir/documents/carousel.json')).then(r => {              if(r.status==200) $scope.carousels = r.data          }).catch(e => console.error(e.data ? e.data.message : e));          // get all categories          $http.get(getLink(serverIO,'rest/categories')).then(r => {              if(r.status==200) $scope.categories = r.data          }).catch(e => console.error(e.data ? e.data.message : e));          // get all products          $http.get(getLink(serverIO,'rest/products')).then(r => {              if(r.status==200) $scope.data = r.data;          }).catch(e => console.error(e.data ? e.data.message : e));            // get all discount data          $http.get(getLink(serverIO,'rest/products/dis')).then(r => {              if(r.status==200) $scope.data3 = r.data;          }).catch(e => console.error(e.data ? e.data.message : e));          // get all hotTOP data          $http.get(getLink(serverIO,'rest/products/hot')).then(r => {              if(r.status==200) $scope.data5 = r.data;          }).catch(e => console.error(e.dataa ? e.data.message : e));      })();        $scope.setLocal = (e) => {          LOCAL.set('item', e);          location.href = '/dsmt/pages/product\_detail';      }      $scope.range = (size) => {          var arr = [];          for (let i = 0; i <= size; arr.push(++i));          return arr;      }      // prepare image      $scope.getImage = function(imgName) {          if(imgName) return imgName.startsWith('http') || imgName.startsWith('blob') ? imgName.trim() : getLink(serverIO, 'data/images/product', imgName);          return getLink(serverIO, 'data/images/product/default.png');      }  }); |

1. **KIỂM THỬ**

## Kiểm thử đăng nhập

|  |  |  |
| --- | --- | --- |
| **Task ID** | **Mô tả test** | **Kết quả test** |
| **Chức năng đăng nhập** | | |
| KT001 | User: không - Pass: không | không đúng tài khoản! |
| KT002 | User: có(**sai**) - Pass: không | không đúng tài khoản! |
| KT003 | User: không - Pass: có (**sai**) | không đúng mật khẩu! |
| KT004 | User: **đúng** - Pass: **sai** | sai mật khẩu! |
| KT005 | User: **sai** -Pass: **đúng** | sai tài khoản! |
| KT006 | User: **đúng** -Pass: **đúng** | đăng nhập thành công |

## Kiểm thử đổi mật khẩu khi quên

|  |  |  |
| --- | --- | --- |
| **Chức năng lấy đổi mật khẩu khi quên** | | |
| KT007 | User: không - email: không | không đúng mã người dùng! |
| KT008 | User: có - email: không | không đúng mã người dùng! |
| KT009 | User: không - email: có | không đúng email! |
| KT010 | User: **đúng** - email: **sai** | sai email! |
| KT011 | User: **sai** - email: **đúng** | sai mã người dùng! |
| KT012 | Biểu đồ: **sai** - code: **sai** | không thể lấy biểu đồ! |
| KT013 | Biểu đồ: **sai** - code: **đúng** | không thể lấy biểu đồ! |
| KT014 | Biểu đồ: **đúng** - code: **sai** | Mã xác thực không chính xác! |
| KT015 | Biểu đồ: **đúng** - code: **đúng** | Mở cửa sổ thay đổi mật khẩu |

## Kiểm thử đổi mật khẩu

|  |  |  |
| --- | --- | --- |
| **Chức năng đổi mật khẩu** | | |
| KT016 | User: không - Pass: không | không đúng tài khoản! |
| KT017 | User: có(**sai**) - Pass: không | không đúng tài khoản! |
| KT018 | User: không - Pass: có (**sai**) | không đúng mật khẩu! |
| KT019 | User: **đúng** - Pass: **sai** | sai mật khẩu! |
| KT020 | User: **sai** -Pass: **đúng** | sai tài khoản! |
| KT021 | User: **đúng** -Pass: **đúng** | mở prompt nhập mật mới. |

## Kiểm thử quản lý sản phẩm

|  |  |  |
| --- | --- | --- |
| **Chức năng sản phẩm** | | |
| KT022 | Thêm: **tồn tại** | mã sản phẩm đã tồn tại! |
| KT023 | Thêm: **không tồn tại** | báo đã thêm mới sản phẩm. |
| KT024 | Sửa: **không tồn tại** | mã sản phẩm không tồn tại! |
| KT025 | Sửa: **đã tồn tại** | sửa biểu đồ sản phẩm. |
| KT026 | Xóa: (role: ADMIN) | trạng thái đơn không được xóa |
| KT027 | Xóa: (role: quản lý): **tồn tại** | thực hiện xóa. |
| KT028 | Xóa: (role: quản lý): **không có** | báo sản phẩm chưa thêm. |
| KT029 | sau khi thêm sản phẩm mới | form bị clear |
| KT030 | sau khi xóa sản phẩm | form hiển thị chuyên sau đó |
| KT031 | sau khi sửa sản phẩm | form được tẩy trắng biểu đồ |

## Kiểm thử quản lý tài khoản

|  |  |  |
| --- | --- | --- |
| **Chức năng quản lý tài khoản** | | |
| KT032 | Thêm: **tồn tại** | mã tài khoản đã tồn tại! |
| KT033 | Thêm: **không tồn tại** | báo đã thêm mới tài khoản. |
| KT034 | Sửa: **không tồn tại** | mã tài khoản không tồn tại! |
| KT035 | Sửa: **đã tồn tại** | sửa biểu đồ tài khoản. |
| KT036 | Xóa: (role: ADMIN) | trạng thái đơn không được xóa |
| KT037 | Xóa: (role: quản lý): **tồn tại** | thực hiện xóa. |
| KT038 | Xóa: (role: quản lý): **không có** | báo tài khoản chưa thêm. |
| KT039 | sau khi thêm tài khoản mới | form bị clear |
| KT040 | sau khi xóa tài khoản | form hiển thị chuyên sau đó |
| KT041 | sau khi sửa tài khoản | form được tẩy trắng biểu đồ |
| KT042 | Chức năng tìm kiếm | hiển thị bảng theo tên khớp |

## Kiểm thử quản lý giỏ hàng

|  |  |  |
| --- | --- | --- |
| **Chức năng quản giỏ hàng** | | |
| KT043 | Thêm: **tồn tại** | mã giỏ hàng đã tồn tại! |
| KT044 | Thêm: **không tồn tại** | báo đã thêm mới giỏ hàng. |
| KT045 | Sửa: **không tồn tại** | mã giỏ hàng không tồn tại! |
| KT046 | Sửa: **đã tồn tại** | sửa biểu đồ giỏ hàng. |
| KT047 | Xóa: (role: ADMIN) | trạng thái đơn không được xóa |
| KT048 | Xóa: (role: quản lý): **tồn tại** | thực hiện xóa. |
| KT049 | Xóa: (role: quản lý): **không có** | báo giỏ hàng chưa thêm. |
| KT050 | sau khi thêm giỏ hàng mới | form bị clear |
| KT051 | sau khi xóa giỏ hàng | form hiển thị chuyên sau đó |
| KT052 | sau khi sửa giỏ hàng | form được tẩy trắng biểu đồ |
| KT053 | giỏ hàng theo list combobox | hiển thị bảng theo tên khóa |

## Kiểm thử quản lý chi tiết đơn hàng

|  |  |  |
| --- | --- | --- |
| **Chức năng quản lý chi tiết đơn hàng** | | |
| KT054 | Thêm: **tồn tại** | mã chi tiết đơn hàng đã tồn tại! |
| KT055 | Thêm | báo đã thêm mới chi tiết đơn hàng. |
| KT056 | Sửa điểm chi tiết đơn hàng | điểm được sửa |
| KT057 | Xóa: (role: ADMIN) | trạng thái đơn không được xóa |
| KT058 | Xóa: (role: quản lý) | thực hiện xóa thành công. |
| KT059 | sau khi thêm chi tiết đơn hàng mới | chuyển tài khoản qua chi tiết đơn hàng |
| KT060 | sau khi xóa chi tiết đơn hàng | chuyển chi tiết đơn hàng qua tài khoản |
| KT061 | sau khi sửa chi tiết đơn hàng | thông báo và hỏi gửi tới email |
| KT062 | Hiển thị theo sản phẩm | giỏ hàng - chi tiết đơn hàng - tài khoản |
| KT063 | Hiển thị theo giỏ hàng | bảng chi tiết đơn hàng - tài khoản đổi |
| KT064 | Chức năng tìm kiếm | hiển thị khớp chuỗi keyCode. |
| KT065 | Gửi điểm về email | Gửi thành công tới email có |
| KT066 | Thêm chi tiết đơn hàng muốn gửi điểm | từ bảng 1 qua bảng 2 |
| KT067 | Xóa chi tiết đơn hàng muốn gửi điểm | từ bảng 2 qua bảng 1 |
| KT068 | Chọn tất cả muốn gửi email | chuyển hết qua bảng bên |
| KT069 | Xóa tất cả muốn gửi email | chuyển hết từ bảng 2 qua 1 |
| KT070 | Gửi điểm cho chi tiết đơn hàng | gửi từng email, xóa trên bảng 2 |
| KT071 | Email nhận điểm kết quả | điểm được gửi tới thành công. |

## Kiểm thử quản lý trạng thái đơn

|  |  |  |
| --- | --- | --- |
| **Chức năng quản lý trạng thái đơn** | | |
| KT072 | Thêm: **tồn tại** | mã trạng thái đơn đã tồn tại! |
| KT073 | Thêm: **không tồn tại** | báo đã thêm mới trạng thái đơn. |
| KT074 | Sửa: **không tồn tại** | mã trạng thái đơn không tồn tại! |
| KT075 | Sửa: (role: ADMIN - cá nhân) | sửa biểu đồ trạng thái đơn. |
| KT076 | Sửa: (role: ADMIN - khác) | thông báo không được sửa |
| KT077 | Xóa: (role: ADMIN) | trạng thái đơn không được xóa |
| KT078 | Xóa: (role: quản lý): **tồn tại** | thực hiện xóa. |
| KT079 | Xóa: (role: quản lý): **không có** | báo trạng thái đơn chưa thêm. |
| KT080 | sau khi thêm trạng thái đơn mới | form bị clear |
| KT081 | sau khi xóa trạng thái đơn | form hiển thị chuyên sau đó |
| KT082 | sau khi sửa trạng thái đơn | form được tẩy trắng biểu đồ |
| KT083 | Chức năng tìm kiếm | hiển thị theo khớp tên |
| KT084 | Chức năng hiển thị (trạng thái đơn) | không được xem mật khẩu |
| KT085 | Chức năng hiển thị (quản lý) | xem được toàn bộ biểu đồ |

## Kiểm thử chức năng thống kê biểu đồ

|  |  |  |
| --- | --- | --- |
| **Chức năng thống kê biểu đồ** | | |
| KT086 | Hiển thị bảng điểm theo khóa | hiển thị theo khóa |
| KT087 | Hiển thị tài khoản theo năm | hiển thị |
| KT088 | Hiển thị sản phẩm | xem theo các sản phẩm |
| KT089 | Hiển thị doanh thu (quản lý) | hiển thị sản phẩm theo năm |
| KT090 | Hiển thị doanh thu (trạng thái đơn) | không hiển thị + thông báo |

## Kiểm thử chức năng trợ giúp khác hàng

|  |  |  |
| --- | --- | --- |
| **Chức năng gửi email** | | |
| KT091 | Gửi email các thông tin **đúng** | gửi thành công email |
| KT092 | Gửi email các thông tin **sai** | thông báo biểu đồ thiếu |
| KT093 | Mở website hỗ trợ | mở và hiển thị thành công |
| KT094 | Mở website sản phẩm | mở và hiển thị thành công |

1. **ĐÓNG GÓI VÀ TRIỂN KHAI**
2. Tạo file jar và các tài nguyên đóng gói

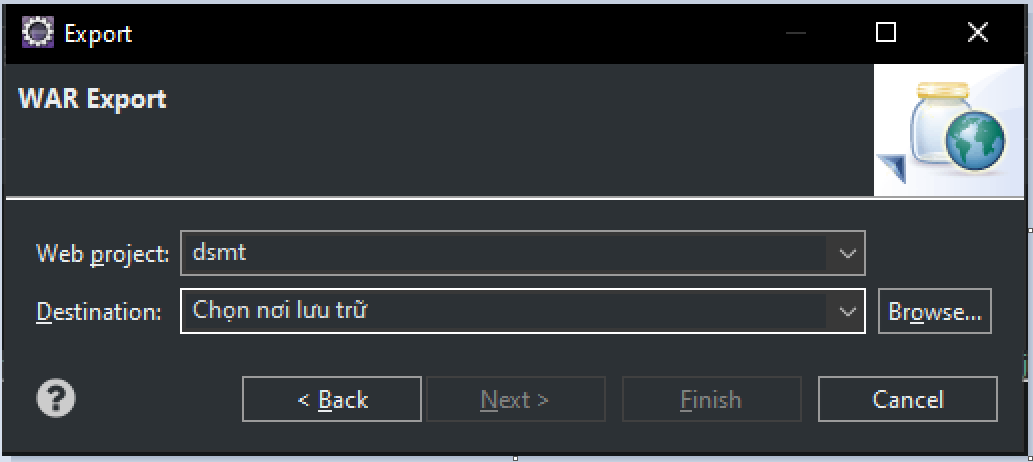
B1. Tạo file war File > Export > WAR file

Graphical user interface, application

Description automatically generated with medium confidenceGraphical user interface

Description automatically generated

B2. Chọn nơi lưu trữ file war



File war sau khi tạo

Text

Description automatically generated

1. Deploy war trên server tomcat

B1. Lựa chọn phiên bản tomcat tương thích với thiết bị

B2. Đưa file dsmt vào mục webapps tomcat trên thiết bị

Graphical user interface, application

Description automatically generated

File war sau khi đưa vào tomcat

Graphical user interface, application, website

Description automatically generated

Vào mục bin và thực hiện start server > localhost + kiểm tra web deploy

Calendar

Description automatically generated

**Hết.**