**Final Project Report**

**INFO 6250 – Web Development Tools and Methods Sec 01**

**Vipul Krishna**

**001226499**

**DOMAIN: e-Commerce**

**Title: SmartShoppers**

**Description:** I have created an online shopping portal that allows customers to make online purchases of items. There are customers and vendors associated with the project. Customers can sign in and buy the items they wish to buy by searching for the items of their choice.

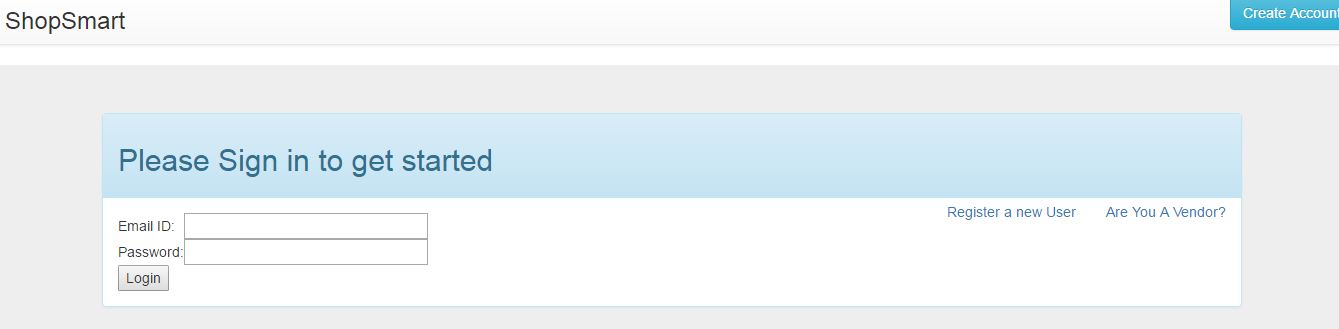
Summary of the Functionality Performed:

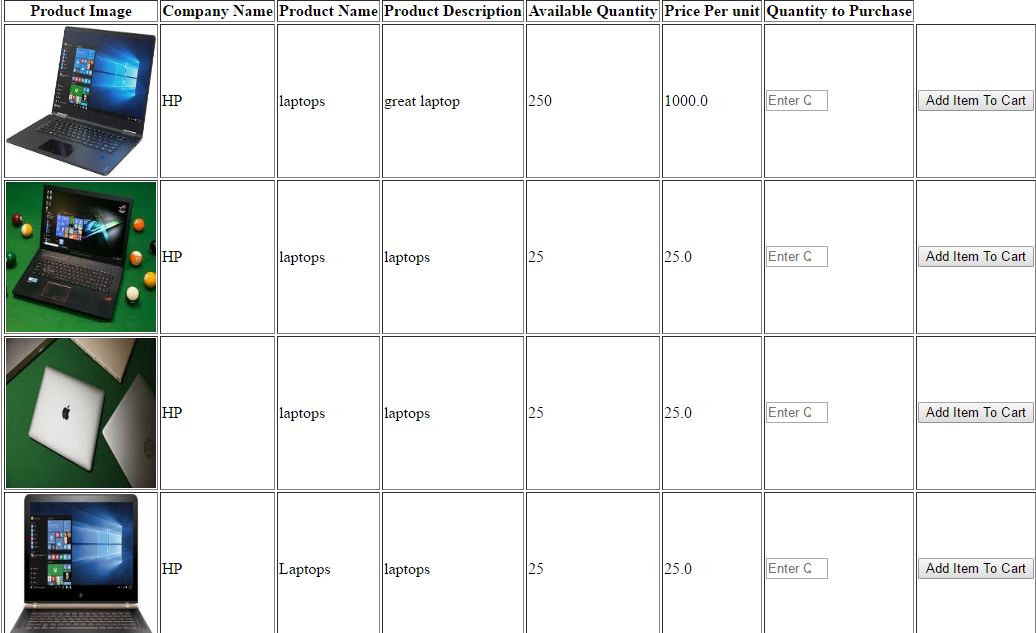
The application allows customers to search for products they wish to buy, add them to cart and place orders. Once the order is placed, customers are notified via an email stating the order number of their recent purchase. Customers can also track their orders and see the details of what order placed. Apart from these features, there are vendors also who have a role to play in this application. The vendors can post their products for sale which are purchased by the customers. Vendors can also see their existing catalog of items on their website. There are authentication checks for individual roles. All data is properly sanitized and validated before persisting them into the database. Technologies and key patters used are:

* Spring MVC(Annotated Controllers)
* Hibernate(Annotated Pojos)
* DAO pattern
* Authentication and Authorization
* Exception handling by individual Exception classes.
* Interceptors
* Validators
* Javax Mail Features.
* File Upload.
* JQuery

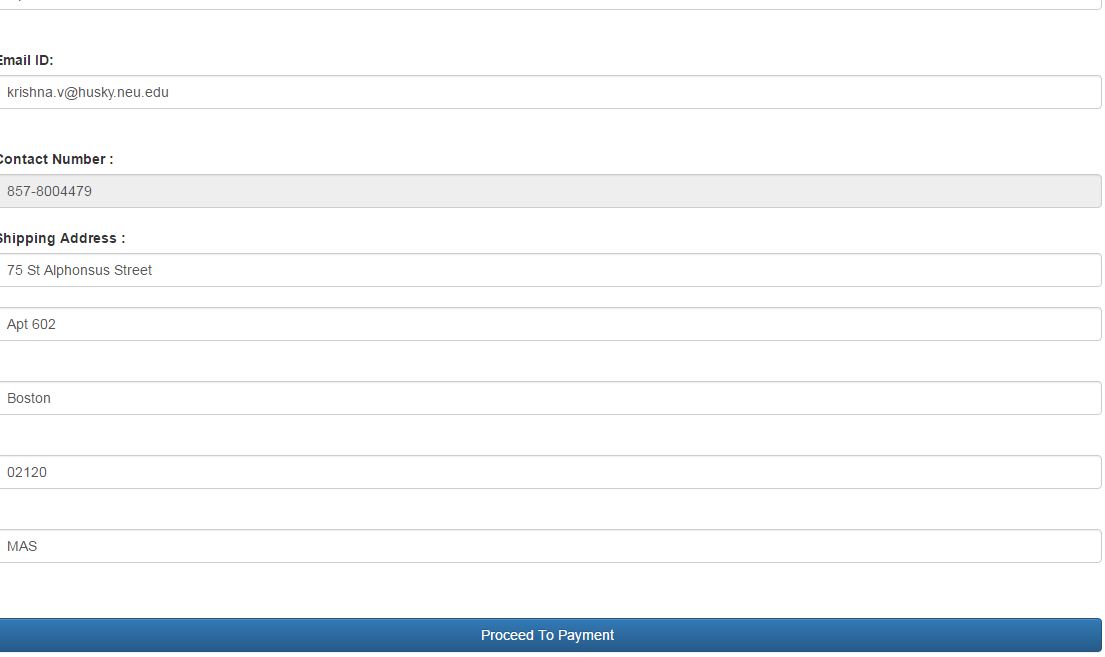
Screenshots of Key Screens:

The major steps are highlighted by screenshots below:

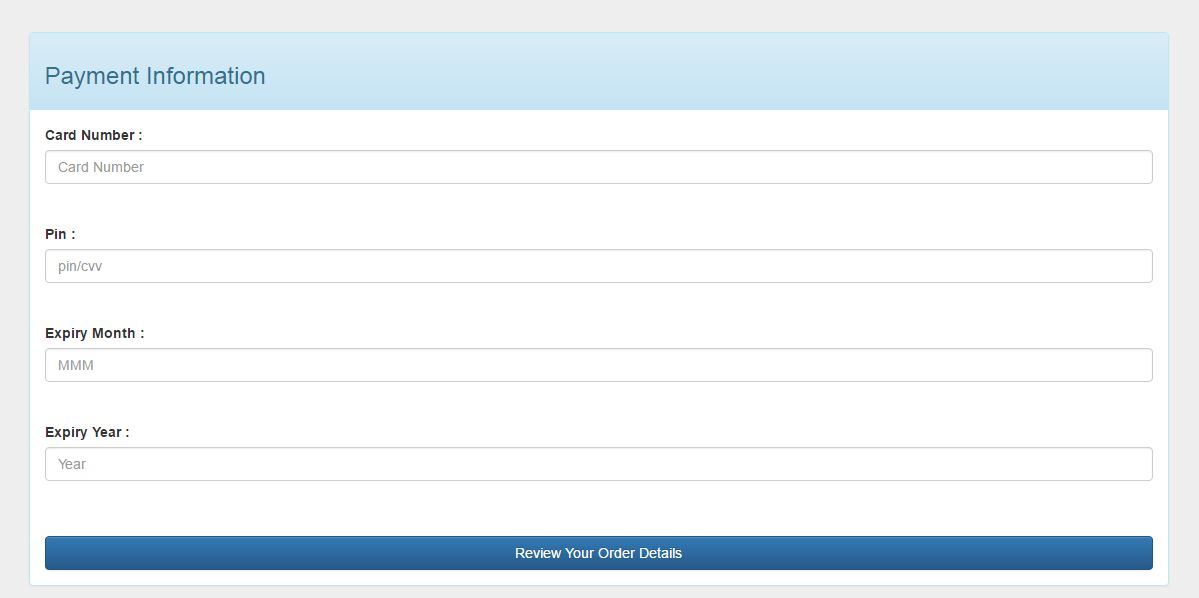
1. User Authentication:
2. Item Search:



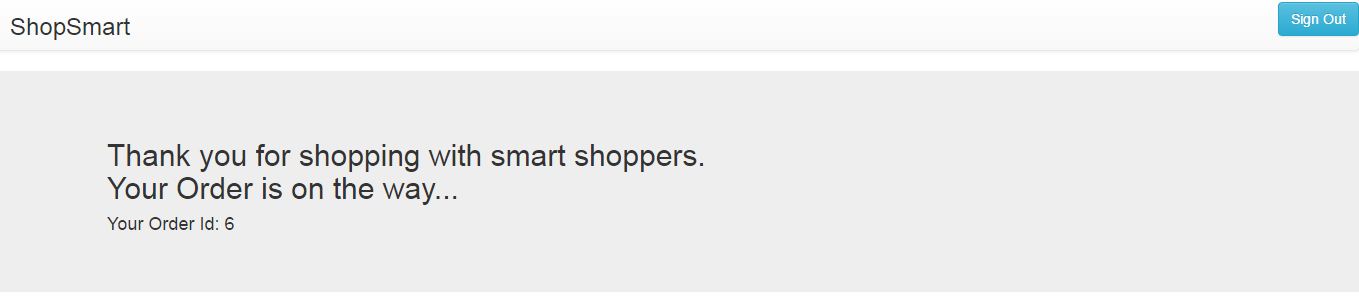
1. .Delivery Information:



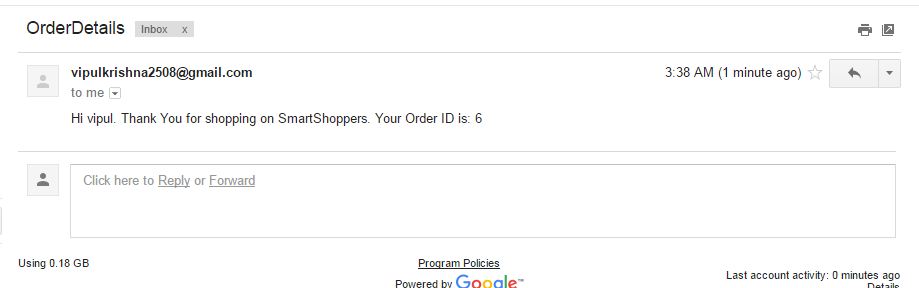
1. Payment:



1. Order Confirmation:



1. Email Sent To Customers:



1. Vendor Work Area:



CONTROLLER CODE:

1. User Controller:

package com.neu.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.http.HttpRequest;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.neu.validator.UserValidator;

import com.neu.validator.VendorValidator;

import com.neu.exception.UserException;

import com.neu.dao.UserDAO;

import com.neu.pojo.Customer;

import com.neu.pojo.Product;

import com.neu.pojo.Vendor;

@Controller

@RequestMapping("/user/\*")

public class UserController {

@Autowired

@Qualifier("userDao")

UserDAO userDao;

@Autowired

@Qualifier("userValidator")

UserValidator userValidator;

@Autowired

@Qualifier("vendorValidator")

VendorValidator vendorValidator;

@InitBinder("customer")

private void customerBinder(WebDataBinder binder) {

binder.setValidator(userValidator);

}

@InitBinder("vendor")

private void vendorBinder(WebDataBinder binder){

binder.setValidator(vendorValidator);

}

@RequestMapping(value = "/", method = RequestMethod.GET)

protected String goToUserHome(HttpServletRequest request) throws Exception {

return "home";

}

@RequestMapping(value = "/user/login", method = RequestMethod.POST)

protected ModelAndView loginUser(HttpServletRequest request) throws Exception {

HttpSession session = (HttpSession) request.getSession();

try {

System.out.print("loginUser");

String emailid = request.getParameter("emailid").replaceAll("<|>|;|,|=", "");

String password = request.getParameter("password").replaceAll("<|>|;|,|=", "");

Customer c = userDao.get(emailid, password);

if(c == null){

System.out.println("UserName/Password does not exist");

session.setAttribute("errorMessage", "UserName/Password does not exist");

return new ModelAndView("error");

}

session.setAttribute("customer", c);

return new ModelAndView("home","customer",c);

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

session.setAttribute("errorMessage", "error while login");

return new ModelAndView("error");

}

}

@RequestMapping(value = "/user/register", method = RequestMethod.GET)

protected ModelAndView registerUser() throws Exception {

System.out.print("registerUser");

return new ModelAndView("signup", "customer", new Customer());

}

@RequestMapping(value = "/user/register", method = RequestMethod.POST)

protected ModelAndView registerNewUser(HttpServletRequest request, @ModelAttribute("customer") Customer customer, BindingResult result) throws Exception {

userValidator.validate(customer, result);

if (result.hasErrors()) {

return new ModelAndView("signup", "customer", customer);

}

try {

System.out.print("registerNewUser");

Customer c = userDao.register(customer);

request.getSession().setAttribute("customer", c);

return new ModelAndView("home", "customer", c);

} catch (UserException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while login");

}

}

////////////////////////////////VENDOR PAGE MAPPINGS//////////////////////////////////////////////////////////////

@RequestMapping(value = "/user/vendorhome", method = RequestMethod.GET)

protected ModelAndView vendorHome() throws Exception {

System.out.print("Vendor Home Page");

return new ModelAndView("vendorhome");

}

@RequestMapping(value = "/user/vendorsignup", method = RequestMethod.GET)

protected ModelAndView vendorsignup() throws Exception {

System.out.println("Vendor SignUp Page");

return new ModelAndView("vendor-signup", "vendor", new Vendor());

}

@RequestMapping(value = "/user/vendorsignup", method = RequestMethod.POST)

protected ModelAndView registerNewVendor(HttpServletRequest request, @ModelAttribute("vendor") Vendor vendor, BindingResult result) throws Exception {

System.out.println("Saving A New Vendor");

vendorValidator.validate(vendor, result);

if (result.hasErrors()) {

return new ModelAndView("vendor-signup", "vendor", vendor);

}

try{

// String firstName = request.getParameter("firstName");

// String lastName = request.getParameter("lastName");

// String email = request.getParameter("emailId");

// String password = request.getParameter("password");

// String companyName = request.getParameter("companyName");

String firstName = vendor.getFirstName();

String lastName = vendor.getLastName();

String email = vendor.getEmailId();

String password = vendor.getPassword();

String companyName = vendor.getCompanyName();

Vendor v = userDao.registerVendor(firstName, lastName, email, password, companyName);

System.out.println("check1");

if (v != null){

request.getSession().setAttribute("vendor", v);

System.out.println("check222222222222222");

return new ModelAndView("vendor-action", "product", new Product());

}

else{

System.out.println("Herreeeeeeeeee");

return new ModelAndView("error");

}

} catch (Exception e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error");

}

}

@RequestMapping(value = "/user/vendoraction", method = RequestMethod.POST)

protected ModelAndView vendorlogin(HttpServletRequest request) throws Exception {

System.out.println("Vendor signing in");

String emailid = request.getParameter("emailid").replaceAll("<|>|;|,|=", "");

String password = request.getParameter("password").replaceAll("<|>|;|,|=", "");

Vendor v = userDao.loginVendor(emailid, password);

if(v != null){

request.getSession().setAttribute("vendor", v);

return new ModelAndView("vendor-action", "product", new Product());

}

else{

return new ModelAndView("error");

}

}

@RequestMapping(value = "/user/logout", method = RequestMethod.GET)

public String logoutUser(HttpServletRequest request){

System.out.println("Entered Logout area");

request.getSession().invalidate();

return "redirect: /edu";

}

}

1. Product Controller:

package com.neu.controller;

import java.io.File;

import java.util.ArrayList;

import javax.servlet.ServletContext;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.validation.BindingResult;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.multipart.commons.CommonsMultipartFile;

import org.springframework.web.servlet.ModelAndView;

import com.neu.dao.ProductDAO;

import com.neu.exception.ProductException;

import com.neu.pojo.Product;

import com.neu.pojo.Vendor;

import com.neu.validator.ProductValidator;

@Controller

@RequestMapping("/product/\*")

public class ProductController {

@Autowired

@Qualifier("productDao")

ProductDAO productDao;

@Autowired

@Qualifier("productValidator")

ProductValidator productValidator;

@Autowired

ServletContext servletContext;

@InitBinder("product")

private void initBinder(WebDataBinder binder) {

binder.setValidator(productValidator);

}

@RequestMapping(value="/product/add.htm", method = RequestMethod.POST)

public ModelAndView addProduct(HttpServletRequest request, @ModelAttribute("product") Product product, BindingResult result)throws Exception{

productValidator.validate(product, result);

if (result.hasErrors()) {

System.out.println("Point 1");

return new ModelAndView("vendor-action", "product", product);

}

System.out.println("Point 5");

try {

//FIle

System.out.println("Test1111111");

File directory;

String check = File.separator; // Checking if system is linux

// based or windows based by

// checking seprator used.

String path = null;

if (check.equalsIgnoreCase("\\")) {

path = servletContext.getRealPath("").replace("build\\", ""); // gives real path as Lab9/build/web/

// so we need to replace build in the path

}

if (check.equalsIgnoreCase("/")) {

System.out.println("inside here 2");

path = servletContext.getRealPath("").replace("build/", "");

path += "/"; // Adding trailing slash for Mac systems.

System.out.println(path+"some path");

}

directory = new File(path);

//directory = new File("");

boolean temp = directory.exists();

if (!temp) {

temp = directory.mkdir();

}

CommonsMultipartFile photoInMemory = product.getPhoto();

String fileName = photoInMemory.getOriginalFilename();

System.out.println("filename is: "+fileName);

File local = new File("E:\\photo\\images",fileName);

photoInMemory.transferTo(local);

String lp = local.getPath();

// String lp2 = "file:///"+lp1;

// String lp = lp2.replace("\\","/");

product.setFilePath(lp);

System.out.println("File Path is: "+lp);

/////////////////////////////////////////////////////////////////////////////////

Vendor v = (Vendor)request.getSession().getAttribute("vendor");

System.out.println("Vendor Name is: "+v.getFirstName());

System.out.println("He represents company: "+v.getCompanyName());

System.out.println("product brand will be set as: "+product.getProductBrand());

Product p = productDao.addNewProduct(product,v,lp,fileName);

request.getSession().setAttribute("product", p);

return new ModelAndView("addproduct-success", "product", p);

} catch(ProductException e) {

System.out.println("Error while adding items to catalog");

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while adding items to catalog");

}

}

@RequestMapping(value="/product/view.htm", method = RequestMethod.POST)

public ModelAndView viewProduct(HttpServletRequest request) throws Exception{

//Fetch the vendor object for that product in the same way as done in the above method

Vendor v = (Vendor)request.getSession().getAttribute("vendor");

String brand = v.getCompanyName();

System.out.println("brand is: "+brand);

try {

ArrayList<Product> myList = (ArrayList<Product>) productDao.fetchProductList(brand);

request.getSession().setAttribute("myList", myList);

return new ModelAndView("vendor-viewproduct");

} catch (ProductException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while fetching items from catalog");

}

}

}

1. Order Controller:

package com.neu.controller;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import javax.servlet.http.HttpServletRequest;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.WebDataBinder;

import org.springframework.web.bind.annotation.InitBinder;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import org.springframework.mail.MailSender;

import org.springframework.mail.SimpleMailMessage;

import com.neu.dao.OrderDAO;

import com.neu.dao.ProductDAO;

import com.neu.exception.OrderException;

import com.neu.exception.ProductException;

import com.neu.pojo.Customer;

import com.neu.pojo.Order;

import com.neu.pojo.OrderItem;

import com.neu.pojo.Product;

import com.neu.validator.OrderValidator;

import com.neu.validator.ProductValidator;

@Controller

public class OrderController {

@Autowired

@Qualifier("orderDao")

OrderDAO orderDao;

@Autowired

@Qualifier("orderValidator")

OrderValidator orderValidator;

@Autowired

@Qualifier("mailSender")

private MailSender mailSender;

@InitBinder("order")

private void initBinder(WebDataBinder binder) {

binder.setValidator(orderValidator);

}

@RequestMapping(value="/item/search.htm", method = RequestMethod.POST)

public ModelAndView searchProduct(HttpServletRequest request){

String searchWord = request.getParameter("category");

try {

ArrayList<Product> myPurchaseItemList = (ArrayList<Product>) orderDao.searchProducts(searchWord);

request.getSession().setAttribute("myPurchaseItemList", myPurchaseItemList);

return new ModelAndView("searchResults");

} catch (OrderException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while customer is searching items");

}

}

@RequestMapping(value="/item/search.htm", method = RequestMethod.GET)

public ModelAndView searchOrder(HttpServletRequest request){

long orderId = Long.parseLong(request.getParameter("orderid"));

System.out.println("Order Id Sent: "+orderId);

try{

Order order = orderDao.orderDetails(orderId);

System.out.println("Customer Name: "+order.getBuyerName());

return new ModelAndView("trackorder","order",order);

}

catch(Exception e1){

return new ModelAndView("error");

}

}

@RequestMapping(value="/item/cart.htm", method = RequestMethod.POST)

public ModelAndView cart(HttpServletRequest request){

// Customer c = (Customer)request.getSession().getAttribute("customer");

// System.out.println(c.getFirstName());

// long productID = Long.parseLong(request.getParameter("productID"));

// System.out.println("Product Id is: "+productID);

// String field = "purchaseQty"+String.valueOf(productID);

// int purchaseQty = Integer.parseInt(request.getParameter(field));

// System.out.println("Purchase Qty passed: "+purchaseQty);

// OrderItem item = orderDao.cartProcess(productID, purchaseQty, c);

// Order order = item.getOrder();

//

// request.getSession().setAttribute("item", item);

return new ModelAndView("cart");

}

@RequestMapping(value="/item/cartupdate.htm", method = RequestMethod.POST)

public ModelAndView cartUpdate(HttpServletRequest request){

Customer c = (Customer)request.getSession().getAttribute("customer");

long productID = Long.parseLong(request.getParameter("productID"));

String field = "purchaseQty"+String.valueOf(productID);

int purchaseQty = Integer.parseInt(request.getParameter(field));

ArrayList<OrderItem> orderItemList = orderDao.cartProcess(productID, purchaseQty, c);

for(OrderItem oi:orderItemList){

System.out.println("Printing order items name: "+oi.getItemName());

System.out.println("Printing order object:"+oi.getOrder());

}

request.getSession().setAttribute("orderItemList", orderItemList);

return new ModelAndView("searchResults" );

}

@RequestMapping(value="/item/delivery.htm", method = RequestMethod.POST)

public ModelAndView delivery(HttpServletRequest request){

//return "delivery";

float totalOrderWorth = 0;

ArrayList<OrderItem> myList = (ArrayList<OrderItem>)request.getSession().getAttribute("orderItemList");

for(OrderItem oi : myList){

totalOrderWorth = totalOrderWorth+(oi.getPurchaseQuantity()\*oi.getItemPrice());

//order.setOrderTotalValue(total);

}

Order order = myList.get(0).getOrder();

order.setOrderTotalValue(totalOrderWorth);

return new ModelAndView("delivery");

}

@RequestMapping(value="/item/payment.htm", method = RequestMethod.POST)

public String payment(HttpServletRequest request){

ArrayList<OrderItem> myList = (ArrayList<OrderItem>)request.getSession().getAttribute("orderItemList");

// for(OrderItem oi : myList){

Order order = myList.get(0).getOrder();

//OrderItem item = (OrderItem)request.getSession().getAttribute("item");

//Order order = oi.getOrder();

order.setBuyerName(request.getParameter("fullname"));

order.setBuyerEmail(request.getParameter("email"));

order.setBuyerPhone(request.getParameter("contact"));

order.setDeliveryAddress1(request.getParameter("address1"));

order.setDeliveryAddress2(request.getParameter("address2"));

order.setZip(request.getParameter("zip"));

order.setCity(request.getParameter("city"));

order.setState(request.getParameter("state"));

//break;

//}

return "payment";

}

@RequestMapping(value="/item/review.htm", method = RequestMethod.POST)

public String review(HttpServletRequest request){

ArrayList<OrderItem> myList = (ArrayList<OrderItem>)request.getSession().getAttribute("orderItemList");

//for(OrderItem oi : myList){

Order order = myList.get(0).getOrder();

//OrderItem item = (OrderItem)request.getSession().getAttribute("item");

//Order order = oi.getOrder();

order.setCardnumber(Long.parseLong(request.getParameter("cardnumber")));

order.setExpmonth(request.getParameter("expmonth"));

order.setExpyear(request.getParameter("expyear"));

order.setCvv(Integer.parseInt(request.getParameter("pin")));

//float total = oi.getPurchaseQuantity()\*oi.getItemPrice();

//order.setOrderTotalValue(total);

DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");

Date date = new Date();

System.out.println(dateFormat.format(date));

order.setDeliveryDate(dateFormat.format(date));

Customer customer = (Customer)request.getSession().getAttribute("customer");

order.setCustomer(customer);

//break;

//}

return "review";

}

@RequestMapping(value="/item/confirmation.htm", method = RequestMethod.POST)

public ModelAndView confirmation(HttpServletRequest request){

//return "confirmation";

ArrayList<OrderItem> myList = (ArrayList<OrderItem>)request.getSession().getAttribute("orderItemList");

//for(OrderItem oi : myList){

Order order = myList.get(0).getOrder();

try {

orderDao.saveOrder(order);

SimpleMailMessage email = new SimpleMailMessage();

email.setTo(order.getBuyerEmail());

email.setSubject("OrderDetails");

email.setText("Hi"+" "+order.getCustomer().getFirstName()+". Thank You for shopping on SmartShoppers. "+

"Your Order ID is: "+order.getOrderId());

mailSender.send(email);

} catch (OrderException e1) {

System.out.println("Exception: " + e1.getMessage());

return new ModelAndView("error", "errorMessage", "error while placing order");

}

for(OrderItem item:myList){

//OrderItem item = (OrderItem)request.getSession().getAttribute("item");

//Order order = item.getOrder();

try {

orderDao.orderSubmit(item);

} catch (OrderException e) {

System.out.println("Exception: " + e.getMessage());

return new ModelAndView("error", "errorMessage", "error while placing order");

}

}

return new ModelAndView("confirmation");

//return new ModelAndView("error");

}

}