Cart DTO,DAO,DAOImpl,Payment page,Payment DTO

1.Update the User model class to match the requirement.

**package** net.sports.ZenSportsBackEnd.model;

**import** java.io.Serializable;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Entity;

**import** javax.persistence.FetchType;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.OneToMany;

**import** javax.persistence.OneToOne;

**import** javax.persistence.Transient;

**import** javax.validation.constraints.Size;

**import** org.hibernate.validator.constraints.NotBlank;

@Entity

**public** **class** User **implements** Serializable{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 3286175755443236838L;

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** **int** userId;

@Size(min=1, max=30,message="user name should be between 1 to 30 characters long")

**private** String userName;

@Size(min=1, max=30,message="user full name should be between 1 to 30 characters long")

**private** String userFullName;

**public** String getUserFullName() {

**return** userFullName;

}

**public** **void** setUserFullName(String userFullName) {

**this**.userFullName = userFullName;

}

**public** **static** **long** getSerialversionuid() {

**return** ***serialVersionUID***;

}

@OneToMany(cascade=CascadeType.***ALL***, mappedBy="user")

List<Address> address=**new** ArrayList<>();

@OneToOne(cascade=CascadeType.***ALL***,mappedBy = "user", fetch = FetchType.***EAGER***)

Cart cart;

**public** List<Address> getAddress() {

**return** address;

}

**public** **void** setAddress(List<Address> address) {

**this**.address = address;

}

**public** Cart getCart() {

**return** cart;

}

**public** **void** setCart(Cart cart) {

**this**.cart = cart;

}

@NotBlank

**private** String role;

@Size(min=1, max=30,message="user password should be between 1 to 30 characters long")

**private** String password;

@Size(min=1, max=30,message="user confirm password should be between 1 to 30 characters long")

@Transient

**private** String userConPassword;

@Size(min=1, max=50,message="user email should be between 1 to 50 characters long")

**private** String userEmail;

@Size(min=1, max=30,message="user name should be between 1 to 30 characters long")

**private** String userMobile;

@NotBlank

**private** String userSecurityQ;

@Size(min=1, max=30,message="user security answer should be between 1 to 30 characters long")

**private** String userSecurityA;

**private** **boolean** enabled;

**public** **boolean** isEnabled() {

**return** enabled;

}

**public** **void** setEnabled(**boolean** enabled) {

**this**.enabled = enabled;

}

**public** User(**int** userId, String userName,String userFullName ,String role, String userPassword, String userConPassword,

String userEmail, String userMobile, String userSecurityQ, String userSecurityA) {

**super**();

**this**.userId = userId;

**this**.userName = userName;

**this**.userFullName = userFullName;

**this**.role = role;

**this**.password = userPassword;

**this**.userConPassword = userConPassword;

**this**.userEmail = userEmail;

**this**.userMobile = userMobile;

**this**.userSecurityQ = userSecurityQ;

**this**.userSecurityA = userSecurityA;

**this**.enabled=**true**;

}

**public** User() {

**super**();

}

**public** String getRole() {

**return** role;

}

**public** **void** setRole(String role) {

**this**.role = role;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** **int** getUserId() {

**return** userId;

}

**public** **void** setUserId(**int** userId) {

**this**.userId = userId;

}

**public** String getUserName() {

**return** userName;

}

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** String getUserConPassword() {

**return** userConPassword;

}

**public** **void** setUserConPassword(String userConPassword) {

**this**.userConPassword = userConPassword;

}

**public** String getUserEmail() {

**return** userEmail;

}

**public** **void** setUserEmail(String userEmail) {

**this**.userEmail = userEmail;

}

**public** String getUserMobile() {

**return** userMobile;

}

**public** **void** setUserMobile(String userMobile) {

**this**.userMobile = userMobile;

}

**public** String getUserSecurityQ() {

**return** userSecurityQ;

}

**public** **void** setUserSecurityQ(String userSecurityQ) {

**this**.userSecurityQ = userSecurityQ;

}

**public** String getUserSecurityA() {

**return** userSecurityA;

}

**public** **void** setUserSecurityA(String userSecurityA) {

**this**.userSecurityA = userSecurityA;

}

}

2.Cart DTO

**package** net.sports.ZenSportsBackEnd.model;

**import** java.io.Serializable;

**import** java.util.List;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.OneToMany;

**import** javax.persistence.OneToOne;

@Entity

**public** **class** Cart **implements** Serializable{

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 7268342053662024973L;

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**private** **int** cartId;

**private** **int** itemCounts;

**private** **int** grandTotal;

@OneToOne(cascade=CascadeType.***PERSIST***)

**private** User user;

@OneToMany(cascade = CascadeType.***ALL***, mappedBy="cart")

**private** List<CartItem> cartItem;

**public** Cart() {

**super**();

}

**public** Cart(**int** cartId, **int** itemCounts, **int** grandTotal) {

**super**();

**this**.cartId = cartId;

**this**.itemCounts = itemCounts;

**this**.grandTotal = grandTotal;

}

**public** **int** getCartId() {

**return** cartId;

}

**public** **void** setCartId(**int** cartId) {

**this**.cartId = cartId;

}

**public** **int** getItemCounts() {

**return** itemCounts;

}

**public** **void** setItemCounts(**int** itemCounts) {

**this**.itemCounts = itemCounts;

}

**public** **int** getGrandTotal() {

**return** grandTotal;

}

**public** **void** setGrandTotal(**int** grandTotal) {

**this**.grandTotal = grandTotal;

}

**public** User getUser() {

**return** user;

}

**public** **void** setUser(User user) {

**this**.user = user;

}

**public** **static** **long** getSerialversionuid() {

**return** ***serialVersionUID***;

}

}

3.Cart DAO file(ICartDAO)

**package** net.sports.ZenSportsBackEnd.dao;

**import** java.util.List;

**import** net.sports.ZenSportsBackEnd.model.Cart;

**import** net.sports.ZenSportsBackEnd.model.CartItem;

**public** **interface** ICartItemDAO

{

**public** List<CartItem> getAllCartItem(Cart item);

**public** CartItem getCartItem(**int** id);

**public** **boolean** addCartItem(CartItem ci);

**public** **boolean** updateCartItem(CartItem ci);

**public** **boolean** deleteCartItem(**int** cartItemId);

}

4.CartDAOImpl

**package** net.sports.ZenSportsBackEnd.daoimpl;

**import** java.util.List;

**import** org.hibernate.SessionFactory;

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

**import** org.springframework.stereotype.Repository;

**import** org.springframework.transaction.annotation.Transactional;

**import** net.sports.ZenSportsBackEnd.dao.ICartItemDAO;

**import** net.sports.ZenSportsBackEnd.dao.IUserDAO;

**import** net.sports.ZenSportsBackEnd.model.Cart;

**import** net.sports.ZenSportsBackEnd.model.CartItem;

@Repository("cartItemDAO")

@Transactional

**public** **class** CartItemDAOImpl **implements** ICartItemDAO

{

@Autowired

SessionFactory sessionFactory;

IUserDAO userDAO;

**public** List<CartItem> getAllCartItem(Cart cart){

**return** sessionFactory.getCurrentSession().createQuery("from CartItem", CartItem.**class**).getResultList();

}

**public** CartItem getCartItem(**int** id){

**return** sessionFactory.getCurrentSession().get(CartItem.**class**,Integer.*valueOf*(id));

}

**public** **boolean** addCartItem(CartItem ci){

**try** {

sessionFactory.getCurrentSession().persist(ci);

**return** **true**;

} **catch** (Exception e) {

e.printStackTrace();

**return** **false**;

}

}

**public** **boolean** updateCartItem(CartItem ci){

**try** {

sessionFactory.getCurrentSession().update(ci);

**return** **true**;

} **catch** (Exception e) {

e.printStackTrace();

**return** **false**;

}

}

**public** **boolean** deleteCartItem(**int** cartItemId){

**try** {

sessionFactory.getCurrentSession().delete(getCartItem(cartItemId));

**return** **true**;

} **catch** (Exception e) {

e.printStackTrace();

**return** **false**;

}

}

}

5.CartItem model

**package** net.sports.ZenSportsBackEnd.model;

**import** java.io.Serializable;

**import** javax.persistence.CascadeType;

**import** javax.persistence.FetchType;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.ManyToOne;

**import** javax.persistence.OneToOne;

**public** **class** CartItem **implements** Serializable {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 7921880822075220902L;

**public** **static** **long** getSerialversionuid() {

**return** ***serialVersionUID***;

}

@Id

@GeneratedValue(strategy=GenerationType.***IDENTITY***)

**int** cartItemId;

**int** quantity;

**double** totalPrice;

@ManyToOne(cascade = CascadeType.***ALL***,fetch = FetchType.***EAGER***)

Cart cart;

@OneToOne(fetch = FetchType.***EAGER***)

Product product;

**public** **int** getCartItemId() {

**return** cartItemId;

}

**public** **void** setCartItemId(**int** cartItemId) {

**this**.cartItemId = cartItemId;

}

**public** **int** getQuantity() {

**return** quantity;

}

**public** **void** setQuantity(**int** quantity) {

**this**.quantity = quantity;

}

**public** **double** getTotalPrice() {

**return** totalPrice;

}

**public** **void** setTotalPrice(**double** totalPrice) {

**this**.totalPrice = totalPrice;

}

**public** Cart getCart() {

**return** cart;

}

**public** **void** setCart(Cart cart) {

**this**.cart = cart;

}

**public** Product getProduct() {

**return** product;

}

**public** **void** setProduct(Product product) {

**this**.product = product;

}

}

6.ICartDAO

package net.sports.ZenSportsBackEnd.dao;

import java.util.List;

import net.sports.ZenSportsBackEnd.model.Cart;

import net.sports.ZenSportsBackEnd.model.CartItem;

public interface ICartItemDAO

{

public List<CartItem> getAllCartItem(Cart item);

public CartItem getCartItem(int id);

public boolean addCartItem(CartItem ci);

public boolean updateCartItem(CartItem ci);

public boolean deleteCartItem(int cartItemId);

}

7.CartItemDAOImpl

package net.sports.ZenSportsBackEnd.daoimpl;

import java.util.List;

import org.hibernate.SessionFactory;

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

import org.springframework.stereotype.Repository;

import org.springframework.transaction.annotation.Transactional;

import net.sports.ZenSportsBackEnd.dao.ICartItemDAO;

import net.sports.ZenSportsBackEnd.dao.IUserDAO;

import net.sports.ZenSportsBackEnd.model.Cart;

import net.sports.ZenSportsBackEnd.model.CartItem;

@Repository("cartItemDAO")

@Transactional

public class CartItemDAOImpl implements ICartItemDAO

{

@Autowired

SessionFactory sessionFactory;

IUserDAO userDAO;

public List<CartItem> getAllCartItem(Cart cart){

return sessionFactory.getCurrentSession().createQuery("from CartItem", CartItem.class).getResultList();

}

public CartItem getCartItem(int id){

return sessionFactory.getCurrentSession().get(CartItem.class,Integer.valueOf(id));

}

public boolean addCartItem(CartItem ci){

try {

sessionFactory.getCurrentSession().persist(ci);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

public boolean updateCartItem(CartItem ci){

try {

sessionFactory.getCurrentSession().update(ci);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

public boolean deleteCartItem(int cartItemId){

try {

sessionFactory.getCurrentSession().delete(getCartItem(cartItemId));

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

}

8,cartFlow.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<flow xmlns=*"http://www.springframework.org/schema/webflow"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/webflow*

*http://www.springframework.org/schema/webflow/spring-webflow.xsd"*>

<var name=*"billing"* class=*"net.sports.ZenSportsFrontEnd.model.Address"*/>

<var name=*"shipping"* class=*"net.sports.ZenSportsFrontEnd.model.Address"*/>

<var name=*"cartitem"* class=*"net.sports.ZenSportsFrontEnd.model.CartItem"*/>

<!-- Executed when webFlow is triggered -->

<on-start>

<!-- registerHandler is java class available through annotated RegisterHandler.java class. -->

<evaluate expression=*"cartHandler.initializeFlow()"* result=*"flowScope.cartModel"*/>

</on-start>

<view-state id=*"start"* view=*"cart/cartview"* model=*"cartitem"*>

<transition on=*"submit"* to=*"billingAddress"*/>

<on-exit>

<evaluate expression=*"cartHandler.saveCart(cartModel,cartitem)"*></evaluate>

</on-exit>

</view-state>

<!-- User enters Billing Address -->

<view-state id=*"billingAddress"* view=*"cart/billing"* model=*"billing"*>

<transition on=*"submit"* to=*"shippingAddress"*/>

<on-exit>

<evaluate expression=*"cartHandler.saveBillingAddress(cartModel,billing)"*></evaluate>

</on-exit>

</view-state>

<!-- User enters Shipping Address -->

<view-state id=*"shippingAddress"* view=*"cart/shipping"* model=*"shipping"*>

<transition on=*"submit"* to=*"orderpayment"*/>

<on-exit>

<evaluate expression=*"cartHandler.saveShippingAddress(cartModel,shipping)"*></evaluate>

</on-exit>

</view-state>

<!-- User view order payment -->

<view-state id=*"orderpayment"* view=*"cart/payment"* model=*"pay"*>

<transition on=*"submit"* to=*"orderreceipt"*/>

<on-exit>

<evaluate expression=*"cartHandler.savePayment(cartModel,pay)"*></evaluate>

</on-exit>

</view-state>

<!-- User view order receipt -->

<view-state id=*"orderreceipt"* view=*"cart/orderreceipt"* model=*"order"*>

<transition on=*"submit"* to=*"confirm"*/>

<on-exit>

<evaluate expression=*"cartHandler.saveOrderReceipt(cartModel,order)"*></evaluate>

</on-exit>

</view-state>

<!-- Confirm shopping -->

<view-state id=*"confirm"* view=*"cart/preOrder"* model=*"flowScope.cartModel"*>

<transition on=*"edit"* to=*"start"* />

<transition on=*"submit"* to=*"storedetail"* />

</view-state>

<action-state id=*"storedetail"*>

<evaluate expression=*"cartHandler.storeDetail(cartModel)"* ></evaluate>

<transition on=*"success"* to=*"welcome"* />

<transition on=*"failure"* to=*"start"* />

</action-state>

<end-state id=*"welcome"* view=*"cart/welcome"* />

<end-state id=*"toHome"* view=*"externalRedirect:contextRelative:/home"* />

<end-state id=*"toException"* view=*"externalRedirect:contextRelative:exception.jsp"* />

<!-- these events are available in every state-->

<global-transitions>

<transition on=*"home"* to=*"toHome"* />

<transition on=*"error"* to=*"toException"* />

</global-transitions>

</flow>

9.CartModel has to be added inside the model folder.

**package** net.sports.ZenSportsBackEnd.model;

**import** java.io.Serializable;

**public** **class** CartModel **implements** Serializable

{

**public** **static** **long** getSerialversionuid() {

**return** ***serialVersionUID***;

}

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = -8137835796335609183L;

**private** CartItem cartItem;

**private** Address billing;

**private** Address shipping;

//Payment payment;

//Order order;

// public Payment getPayment() {

// return payment;

// }

//

// public void setPayment(Payment payment) {

// this.payment = payment;

// }

//

// public Order getOrder() {

// return order;

// }

//

// public void setOrder(Order order) {

// this.order = order;

// }

**public** CartItem getCartItem() {

**return** cartItem;

}

**public** **void** setCartItem(CartItem cartItem) {

**this**.cartItem = cartItem;

}

**public** Address getBilling() {

**return** billing;

}

**public** **void** setBilling(Address billing) {

**this**.billing = billing;

}

**public** Address getShipping() {

**return** shipping;

}

**public** **void** setShipping(Address shipping) {

**this**.shipping = shipping;

}

**public** CartModel() {

cartItem=**new** CartItem();

billing=**new** Address();

shipping=**new** Address();

//order=new Order();

//payment=new Payment();

}

}

10.Payment.jsp has to be added

<div class=*"content"*>

       <div class=*"row"*>

       <div class=*"col-md-2"*></div>

       <div class=*"col-md-8 col-centered"* class=*"table-centered"*>

             <div class=*"form-container col-md-8 well"*>

                    <form class=*"form-horizontal"* method=*"POST"*>

                           <div

                                 style="height: *50px*;width:*100%*; background: *rgb(73, 186, 142)*; border-radius: *5px*;">

                                 <h3 style="margin-left: *20px*;">Share your

                                        contact details</h3>

                           </div><br/>

                           <div class=*"form-group"*>

                                 <input name=*"email"* type=*"text"* placeholder=*"abc@xyz.com"*

                                        class=*"form-control"* style="width: *55%*; display: *inline*" /> <input

                                        type=*"text"* placeholder=*"+919945678902"* class=*"form-control"*

                                        name=*"phone"* maxlength=*"13"* minlength=*"13"*

                                        style="width: *35%*; display: *inline*" />

                           </div>

                           <div

                                 style="height: *50px*;width:*100%*; background: *rgb(73, 186, 142)*; border-radius: *5px*;">

                                 <h3 style="margin-left: *20px*;">Payment

                                        Details</h3>

                           </div><br/>

                                 <div class=*"form-group"*>

                                        <label for=*"cno"* style="color: *gray*">Card Number</label> <input

                                               type=*"text"* class=*"form-control"* name=*"cardNo"*

                                               placeholder=*"XXXX-XXXX-XXXX-XXXX"* maxlength=*"16"* minlength=*"16"* />

                                 </div>

                                 <div class=*"form-group"*>

                                        <label for=*"name"* style="color: *gray*">Name on the card</label> <input

                                               type=*"text"* class=*"form-control"* name=*"cardName"*

                                               placeholder=*"Name On The Card"* />

                                 </div>

                                 <div class=*"form-group"*>

                                        <label for=*"mm"* style="color: *gray*; display: *inline*">Expiry</label>

                                        <label for=*"cvv"* style="color: *gray*">CVV</label><br /> <input

                                               type=*"number"* name=*"mm"* min=*"1"* max=*"12"* class=*"form-control"*

                                               id=*"mm"* style="width: *65px*; display: *inline*" maxlength=*"3"*

                                               placeholder=*"mm"* /> <input type=*"number"* name=*"yy"* min=*"2017"*

                                               max=*"2027"* class=*"form-control"* maxlength=*"4"*

                                               style="width: *75px*; display: *inline*" placeholder=*"yy"* />

                                        <input type=*"text"* name=*"cvv"* minlength=*"3"* maxlength=*"3"*

                                               length=*"3"* class=*"form-control"*

                                               style="width: *75px*; display: *inline*" placeholder=*"CVV"* />

                                 </div>

                                 <div class=*"form-group"*>

                                        <input type=*"submit"* value=*"Make Payment"* action=*"payment"*

                                               class=*"btn btn-danger"* />

                                 </div>

                           <br/><br/><br/><br/>

                           <div class=*"form-group"*>

                                 <p style="color: *gray*">By clicking on 'Make Payment' you are

                                        agreeing to terms and conditions .</p>

                           </div>

                    </form>

             </div>

       </div>

       <div class=*"col-md-2"*></div>

       </div>

       </div>

11.Payment.java class had to be added in model.

**package** net.sports.ZenSportsBackEnd.model;

**import** java.io.Serializable;

**import** javax.persistence.CascadeType;

**import** javax.persistence.Entity;

**import** javax.persistence.GeneratedValue;

**import** javax.persistence.GenerationType;

**import** javax.persistence.Id;

**import** javax.persistence.OneToOne;

**import** javax.validation.constraints.Max;

**import** javax.validation.constraints.Min;

**import** javax.validation.constraints.Size;

**import** org.hibernate.validator.constraints.CreditCardNumber;

**import** org.hibernate.validator.constraints.Email;

@Entity

**public** **class** Payment **implements** Serializable

{

       /\*\*

       \*

        \*/

**private** **static** **final** **long** ***serialVersionUID*** = 6467634240249641816L;

**public** **static** **long** getSerialversionuid() {

**return** ***serialVersionUID***;

       }

       @Id

       @GeneratedValue(strategy = GenerationType.***IDENTITY***)

**private** **int** paymentId;

       @Email

**private** String email;

       @Size(min=1, max=30,message="user mobile number should be between 1 to 13 characters long")

**private** String phone;

       @Size(min=16, max=16,message="user card number should be 16 characters long")

**private** String cardNo;

       @Size(min=3, max=3,message="user cvv should be 3 characters long")

**private** String cvv;

       @CreditCardNumber

**private** String cardName;

       @OneToOne(cascade=CascadeType.***ALL***)

**private** User user;

**public** User getUser() {

**return** user;

       }

**public** **void** setUser(User user) {

**this**.user = user;

       }

       @Min(value=01)

       @Max(value=12)

**private** **int** mm;

       @Min(value=17)

       @Max(value=25)

**private** **int** yy;

**public** **int** getPaymentId() {

**return** paymentId;

       }

**public** **void** setPaymentId(**int** paymentId) {

**this**.paymentId = paymentId;

       }

**public** String getEmail() {

**return** email;

       }

**public** **void** setEmail(String email) {

**this**.email = email;

       }

**public** String getPhone() {

**return** phone;

       }

**public** **void** setPhone(String phone) {

**this**.phone = phone;

       }

**public** String getCardNo() {

**return** cardNo;

       }

**public** **void** setCardNo(String cardNo) {

**this**.cardNo = cardNo;

       }

**public** String getCvv() {

**return** cvv;

       }

**public** **void** setCvv(String cvv) {

**this**.cvv = cvv;

       }

**public** String getCardName() {

**return** cardName;

       }

**public** **void** setCardName(String cardName) {

**this**.cardName = cardName;

       }

**public** **int** getMm() {

**return** mm;

       }

**public** **void** setMm(**int** mm) {

**this**.mm = mm;

       }

**public** **int** getYy() {

**return** yy;

       }

**public** **void** setYy(**int** yy) {

**this**.yy = yy;

       }

**public** Payment(**int** paymentId, String email, String phone, String cardNo, String cvv, String cardName, **int** mm,

**int** yy) {

**super**();

**this**.paymentId = paymentId;

**this**.email = email;

**this**.phone = phone;

**this**.cardNo = cardNo;

**this**.cvv = cvv;

**this**.cardName = cardName;

**this**.mm = mm;

**this**.yy = yy;

       }

**public** Payment() {

**super**();

             // **TODO** Auto-generated constructor stub

       }

}