E-Shopping System

Software design

Kun Wang 542115502

Lin Li 542115503

Yiyun Wang 542115510

URS

URS 1: The user can register as a customer.

URS 2: The user can login to the system as customer.

URS 3: The customer can update personal information.

URS 4: The customer can browse the product categories.

URS 5: The customer can search for the product name.

URS 6: The customer can add products to shopping cart.

URS 7: The customer can select multi product.

URS 8: The customer can check out to summary.

URS 8.1: The customer can view what products she/he selected.

URS 8.2: The customer can view the total price of products he/she has to pay.

URS 8.3: The customer can select a payment option.

URS 8.3.1: The customer can select the way of money transfer to pay.

URS 8.3.2: The customer can select the way of credit card to pay.

URS8.3.3: The customer can select the way of PayPal to pay.

URS 8.4: The customer can delete, update products from the shopping cart.

URS 9: The customer can click confirm button to finish buying process.

URS 10: The customer can save shopping cart.

URS 11: The customer can browse shopping history.

URS 12: The administrator can edit name, description, and pictures of the product.

URS 13: The administrator can add products to products list.

URS 14: The administrator can delete products from products list.

URS 15: The administrator can browse the shopping history of all customers.

SRS

SRS 1: The system provides the UI which receive the username, password, re-enter password, address and picture that the customer registered.

SRS 2: The system shall check the username format and the username format should be email format.

SRS 3: The system provides the UI which the customer can update personal information later.

SRS 4: The system shall check the password format. The password format must be 4-20 characters. The password must contain the small letter, capital letter, and number.

SRS 5: The system shall check re-enter password format and consistency with password format.

SRS 6: The system shall verify the customer's address.

SRS 7: The system shall verify the customer's picture.

SRS 8: The system shall display an error message "the username must be an e-mail".

SRS 9: The system shall display an error message "the password length should be 4-20 characters".

SRS 10: The system shall display an error message "the re-enter password must be consistency with password format".

SRS 11: The system shall display an error message "The customer address should fill in".

SRS 12: The system shall display an error message "The customer picture must upload".

SRS 13: The system shall send a successful registration email to customer.

SRS 14: The system shall display successful register message on the registration page.

SRS 15: The system provides the log in the interface for user.

SRS 16: The system shall check whether username and password are match or not .

SRS 17: The system shall display an error message "the username is not correct"

SRS 18: The system shall display an error message “the password is not correct”

SRS 19: The system shall list the product with name, short description and picture.

SRS 20: The system provides search option for product name.

SRS 21: The system provides product choose option when customer wants add the product to shopping carts.

SRS 22: The system provides multi product option to customer.

SRS 23: The system shall show the summary of product in the shopping cart what customer selected when check out.

SRS 24: The system shall list all products name that customer selected.

SRS 25: The system shall list price of each product and total price.

SRS 26: The system provides payment options, e- bank transfer, credit card and PayPal.

SRS 27: The system provides confirm options for the customer to confirm buying process.

SRS 28: The system provides delete option for the customer to delete products where in the shopping cart.

SRS 29: The system shall show buying transaction history what customer has bought.

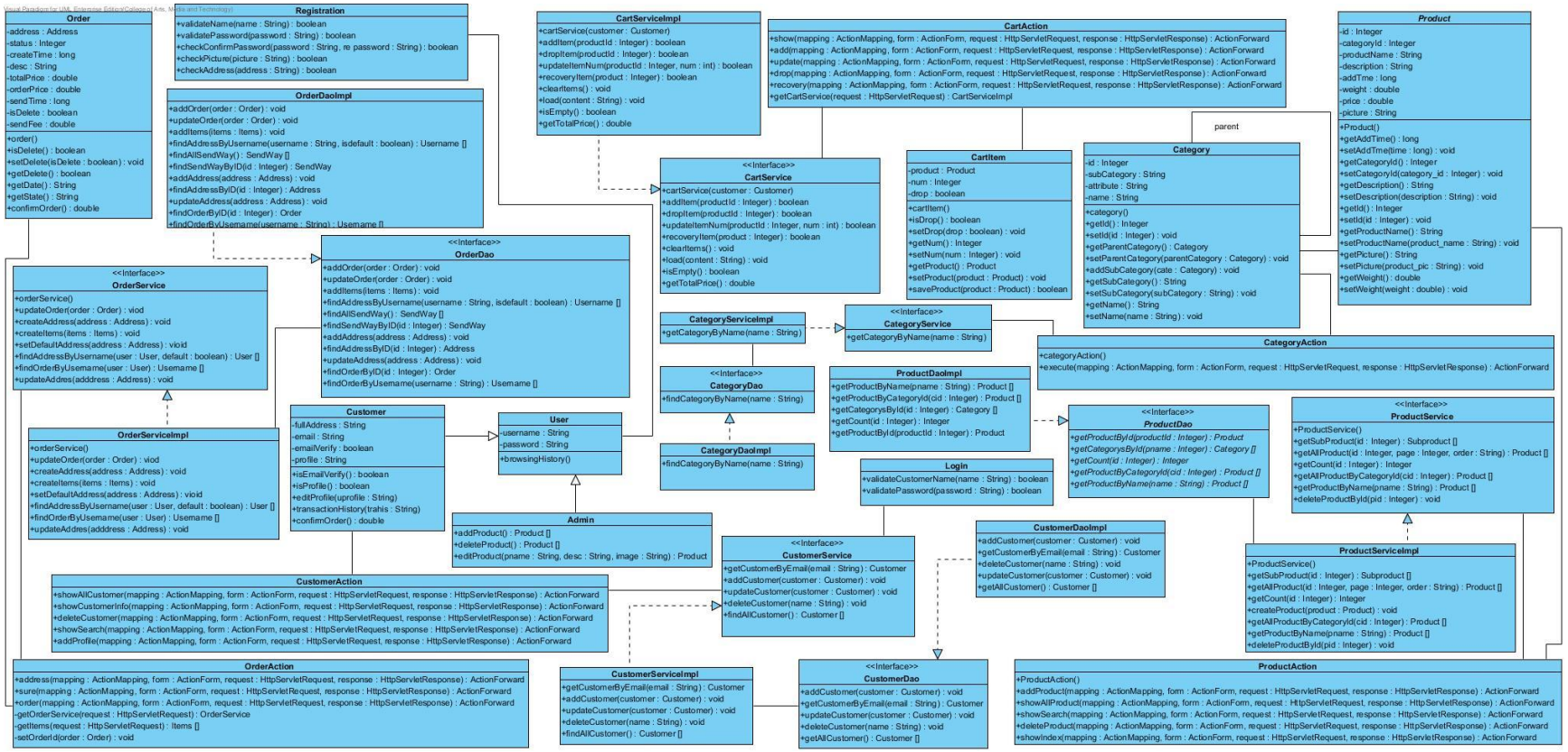
SRS 30: The system shall list all products what customer selected and saved in the shopping cart.

SRS 31: The system provides edit product details option for administrator.

SRS 32: The system provides add product option for administrator.

SRS 33: The system provides delete product option for administrator.

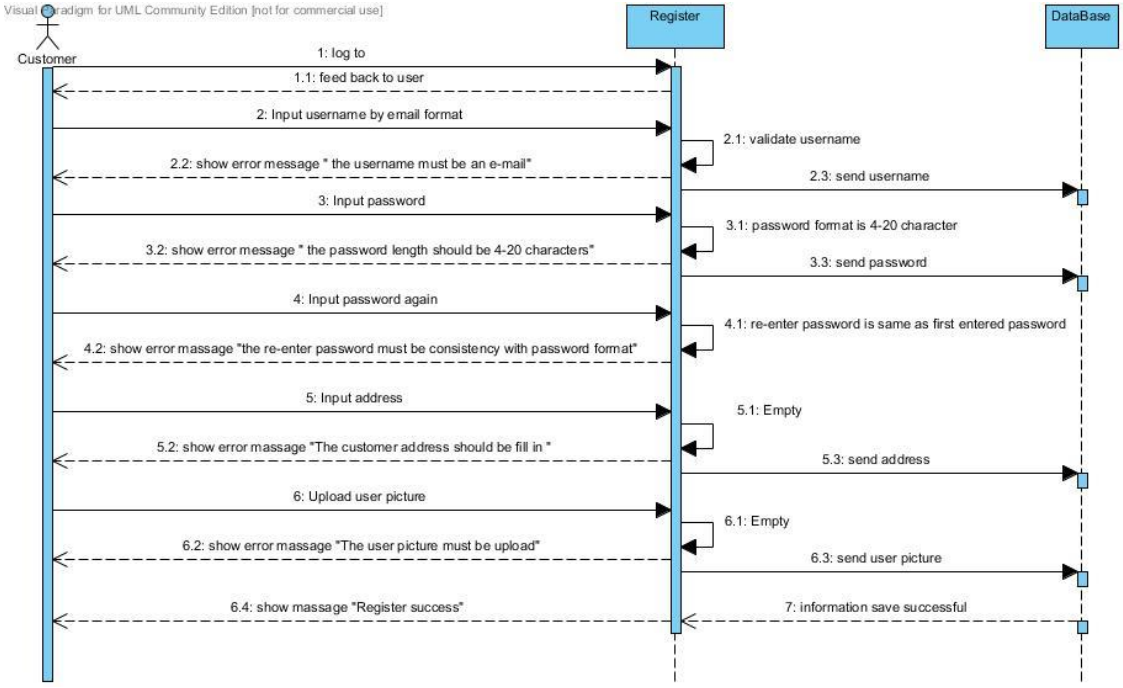
SRS 34: The system allowed administrator to view customers shopping history.



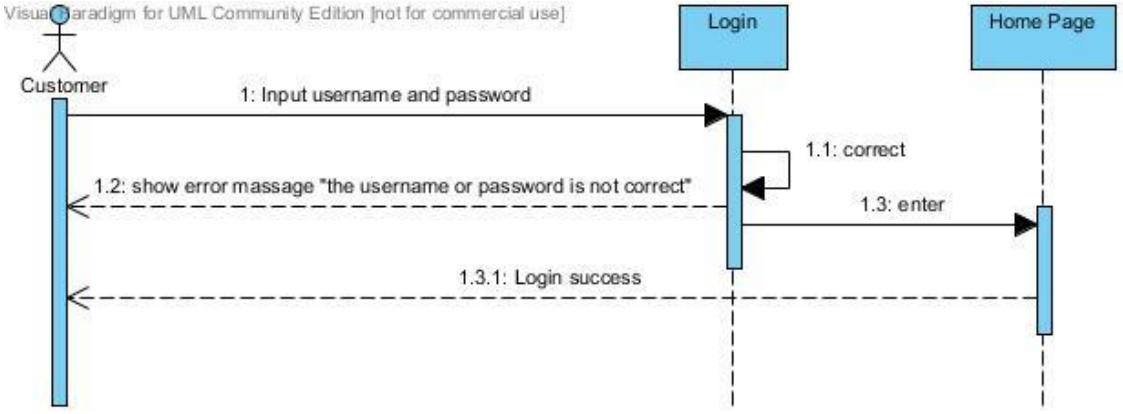
Class Diagram

Sequence Diagram

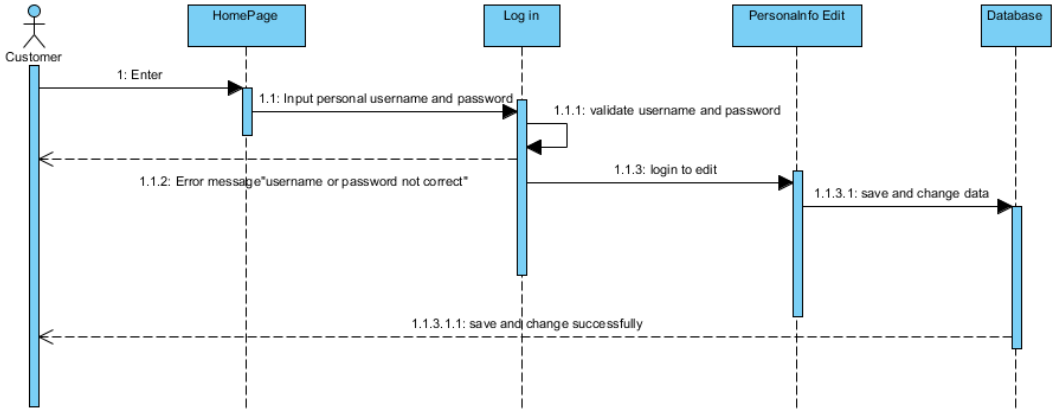
Register UI



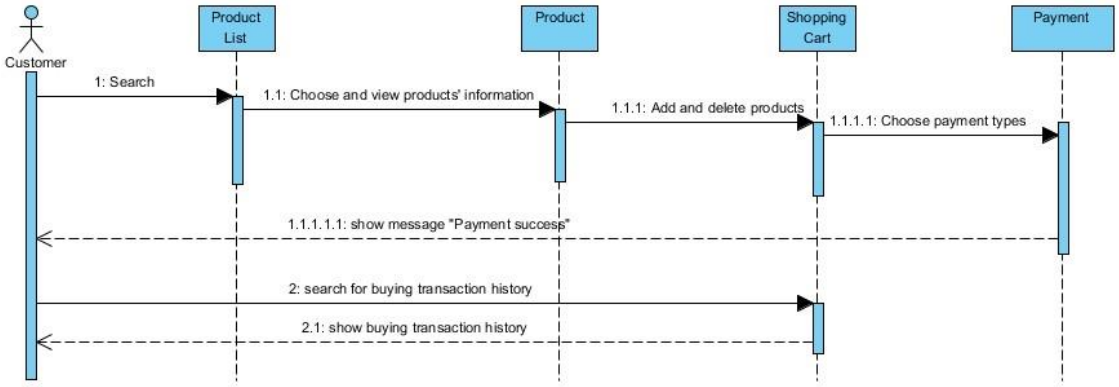
Login



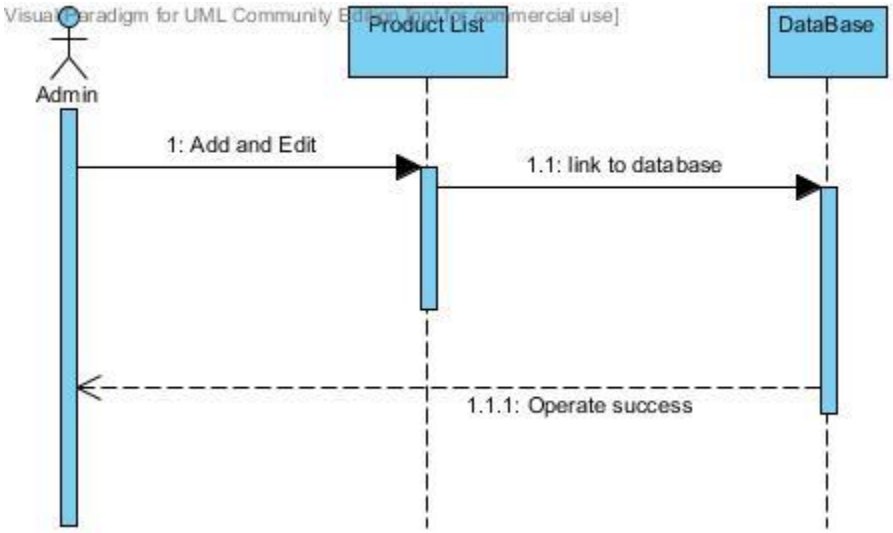
Personal Information Edit



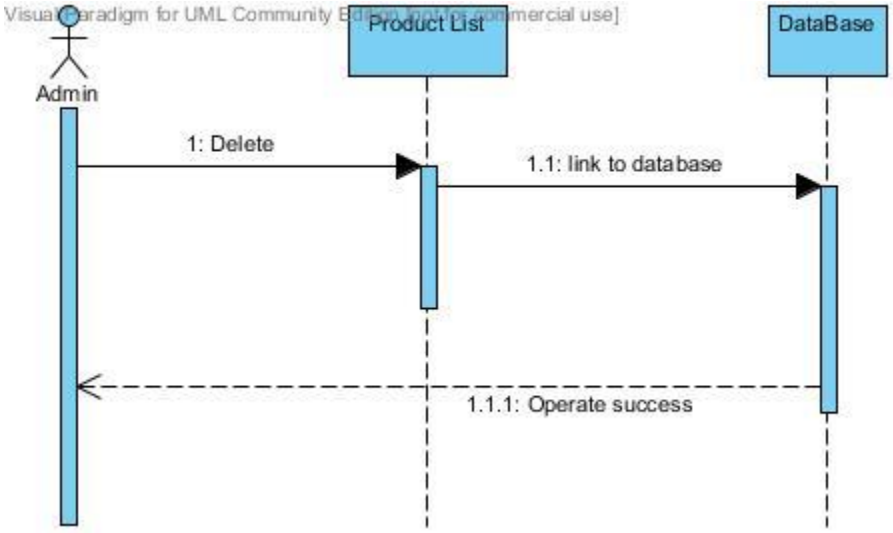
Order Product



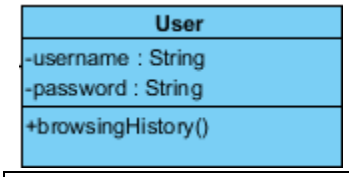
Add Product



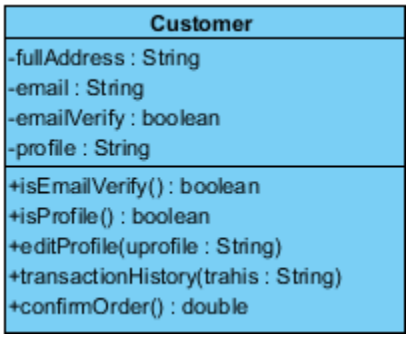
Delete Product



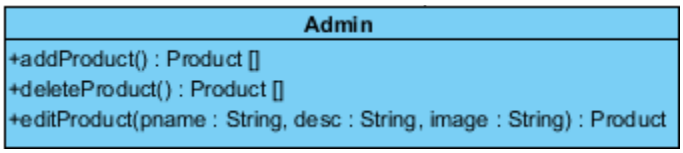
Method Design



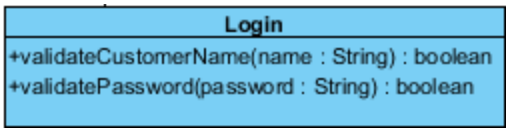
Method	parameter	Description
+browsingHistory()	-	This method let customer and admin can brow shopping history.



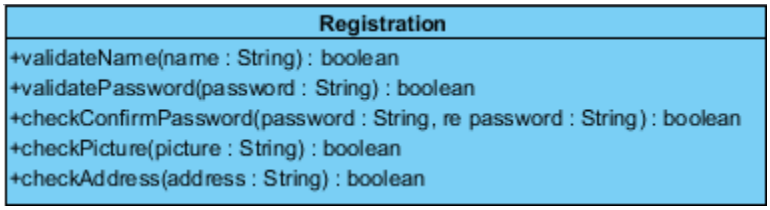
Method	parameter	Description
+isEmailVerify()	-	This method validate that the username is an email format.
+isProfile()	-	This method validate that the user picture is a correct format.
+editProfile()	uprofile:String	This method let customer edit personal information after register.
+transactionHistory()	trahis:String	This method allow customer view transaction history.
+confirmOrder()	-	This method let customer confirm order before payment.



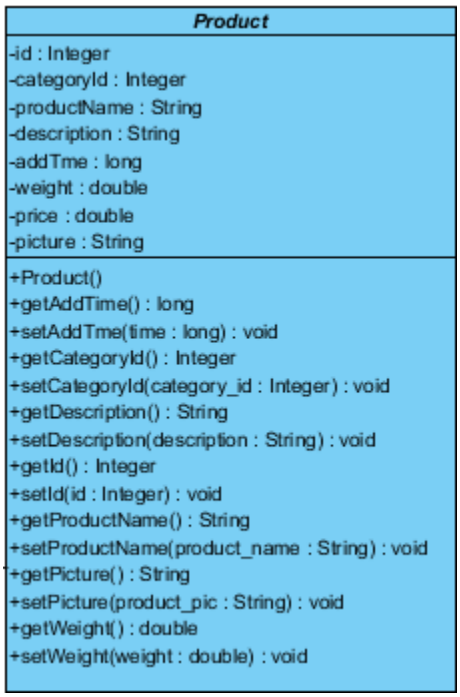
Method	Parameter	Description
+addProduct()	Product[]	This method let admin add product to product list.
+deleteProduct()	Product[]	This method let admin delete product from product list.
+editProduct()	pname:String, desc:String, image:String	This method let admin edits product where name, description and image.



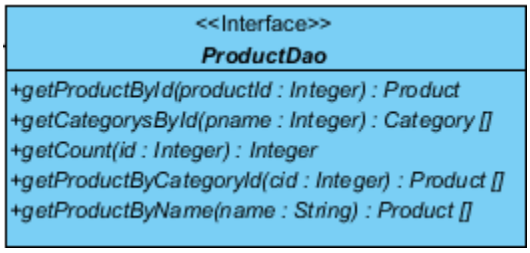
Method	parameter	Description
+validateCustomerName( )	name:String	This method validate customer name whether or not is a registered name.
+validatePassword()	password:String	This method validate customer password whether or not is right.



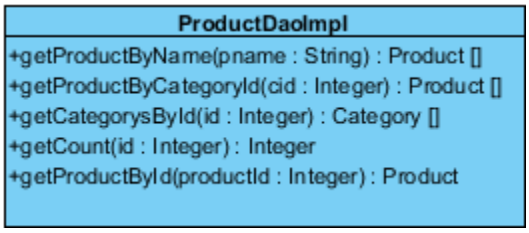
Method	parameter	Description
+validatePassword()	password : String	This method validate whether or not customer's password is a right format
+checkConfirmPassword()	password : String, repassword : String	This method checks customer confirm password whether or not is the same as an inputted password.
+validateName()	name : String	This method validate whether or not a right name format.
+checkPicture()	picture:String	This method checks customer picture whether or not is picture format.
+checkAddress()	address:String	This method is checks whether or not customer address is valid.



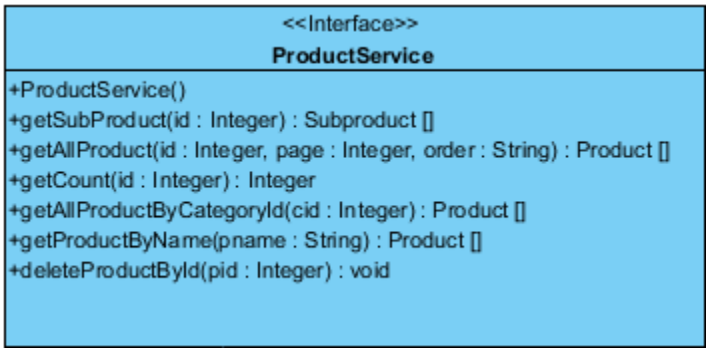
Method	parameter	Description
+product()	-	This method provides product project.



Method	parameter	Description
+getProductById()	productId: Integer	-
+getCategorysById()	Pname: Integer	-
+getCount()	id: Integer	-
+getProductByCategoryId()	cid:Integer	-
+getProductByName()	name: String	-

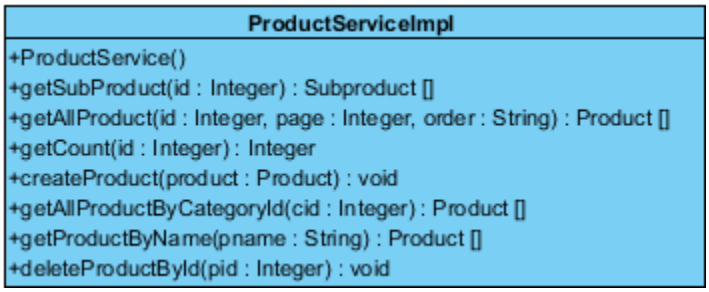


Method	parameter	Description
+getProductById()	productId: Integer	-
+getCategorysById()	Pname: Integer	-
+getCount()	id: Integer	-
+getProductByCategoryId()	cid:Integer	-
+getProductByName()	name: String	-

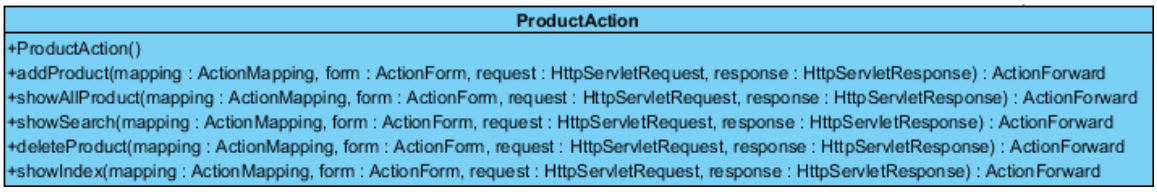


Method	parameter	Description
+productService()	-	Calling the method of product service where from

		product DAO implement class.
+deleteProductById()	pid: Integer	Calling the method of delete product by ID where from product DAO implement class.

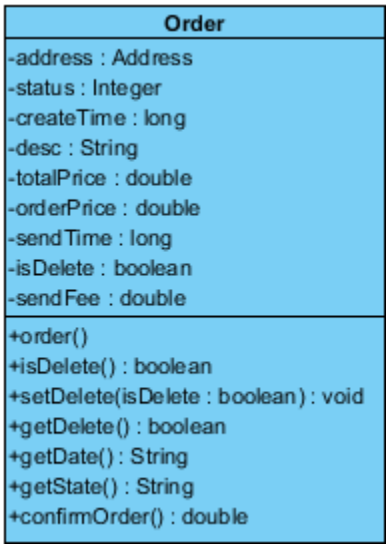


Method	**parameter**	**Description**
+productService()	-	Implement the method of product service where from product service interface.
+deleteProductById()	pid: Integer	Implement the method of delete product by ID where from product service interface.

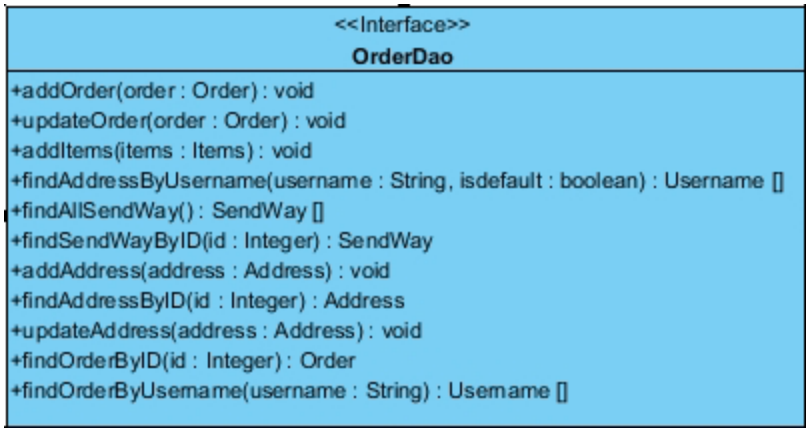


Method	**parameter**	**Description**
+productAction()	-	Calling the method of product action where from product service interface.
+addProduct()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of add product where from product service interface.
+showAllProduct()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of show all products where from product service interface.

+showSearch()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of show search where from product service interface.
+deleteProduct()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of delete product where from product service interface.
+showIndex()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of show index page where from product service interface.

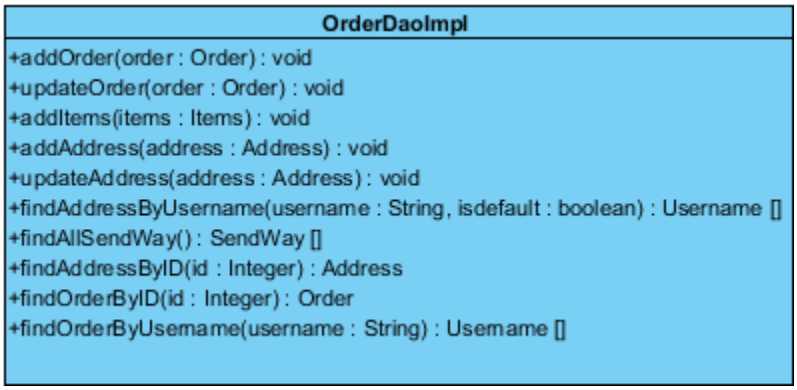


Method	**parameter**	**Description**
+order()	-	This method let customer order items.
+isDelete()	-	This method let customer can delete items that he ordered.
+confirmOrder()	-	This method let customer confirm order before payment.



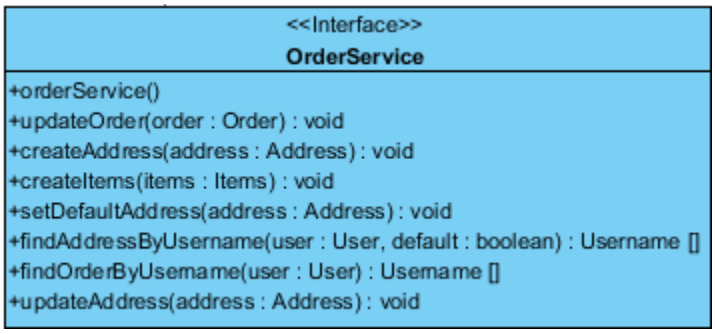
Method	**parameter**	**Description**
+addOrder()	order:Order	Calling the method of add order which from order DAO implement class.
+updateOrder()	order : Order	Calling the method of update order which from order DAO implement class.
+addItems()	items:Items	Calling the method of add items which from order DAO implement class.
+findAddressByUsername()	userid : Integer, isdefault : boolean	Calling the method of find address by username which from order DAO implement class.
+findAllSendWayByUsernam e()	username: String, isDefault:boolean	Calling the method of find all send way by username which from order DAO implement class.
+findAllSendWay()	SendWay[]	Calling the method of find all send way which from order DAO implement class.
+addAddress()	address:Address	Calling the method of add address which from order DAO implement class.
+findAddressByID()	id:intdger	Calling the method of find address by ID where from order DAO implement class.
+updateAddress()	address : .Address	Calling the method of update address where from order DAO implement class.
+findOrderById() :	id:integer	Calling the method of find order by Id where from order DAO implement class.
+findOrderByUsername()	Username:String	Calling the method of find order by username where

		from order DAO implement class.

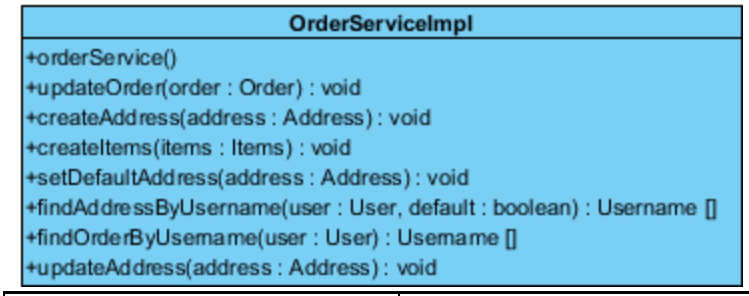


Method	parameter	Description
+addOrder()	order:Order	Implement the method of add order which from order DAO interface.
+updateOrder()	order : Order	Implement the method of update order which from order DAO interface.
+addItems()	items:Items	Implement the method of add item which from order DAO interface.
+findAddressByUsername()	userid : Integer, isdefault : boolean	Implement the method of find address by username which from order DAO interface.
+findAllSendWayByUsernam e()	username: String, isDefault:boolean	Implement the method of find all send way by username which from order DAO interface.
+findAllSendWay()	SendWay[]	Implement the method of find all wend which from order DAO interface.
+addAddress()	address:Address	Implement the method of add address which from order DAO interface.
+findAddressByID()	id:intdger	Implement the method of find address by ID which from order DAO interface.
+updateAddress()	address : .Address	Implement the method of update address which from order DAO interface.
+findOrderById() :	id:integer	Implement the method of find order by id which from order DAO interface.

+findOrderByUsername()	Username:String	Implement the method of find order by username where from order DAO interface.



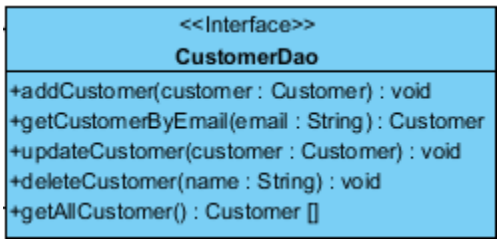
Method	**parameter**	**Description**
+orderService()	-	Calling the method of order service where from order DAO implement class.
+updateOrder()	order:Order	Calling the method of update order which from order DAO implement class.
+createAddress()	address : Address	Calling the method of create address which from order DAO implement class.
+createItems()	items : Items	Calling the method of create items which from order DAO implement class.
+findAddressByUsername()	user : User, default : boolean	Calling the method of find address by username which from order DAO implement class.
+findOrderByUsername()	user :User	Calling the method of find order by username which from order DAO implement class.
+updateaAddress()	address:Address	Calling the method of update address which from order DAO implement class.



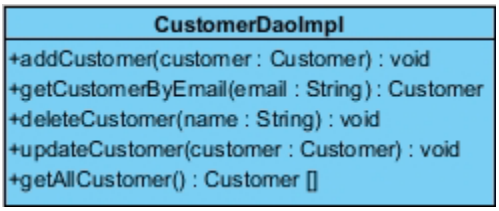
Method	parameter	description
+orderService()	-	Implement the method of order service where from order service interface.
+updateOrder()	order:Order	Implement the method of update order which from order service interface.
+createAddress()	address : Address	Implement the method of create address which from order service interface.
+createItems()	items : Items	Implement the method of create items which from order service interface.
+findAddressByUsername()	user : User, default : boolean	Implement the method of find address by username which from order service interface.
+findOrderByUsername()	user :User	Implement the method of find order by username which from order service interface.
+updateaAddress()	address:Address	Implement the method of update address which from order service interface.

Method	parameter	Description
+address()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of address where from order service interface.
+sure()	mapping : ActionMapping,	Calling the method of sure

	form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	which from order service interface.
+order()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Callng the method of order which from order service interface.

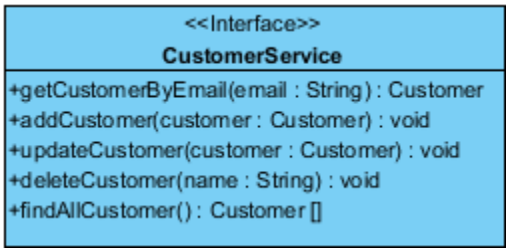


Method	**parameter**	**Description**
+addCustomer()	customer : Customer	Calling the method of add customer which from customer DAO implement class.
+updateCustomer()	customer :Customer	Calling the method of update customer where from customer DAO implement class.
+deleteCustomer()	name : String	Calling the method of delete customer where from customer DAO implement class.

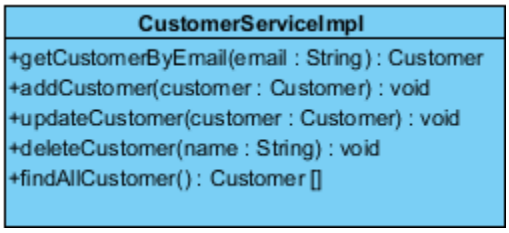


Method	**parameter**	**Description**
+addCustomer()	customer : Customer	Implement the method of add customer which from customer DAO interface.
+updateCustomer()	customer :Customer	Implement the method of update customer where from customer DAO

		interface.
+deleteCustomer()	name : String	Implement the method of delete customer where from customer DAO interface.



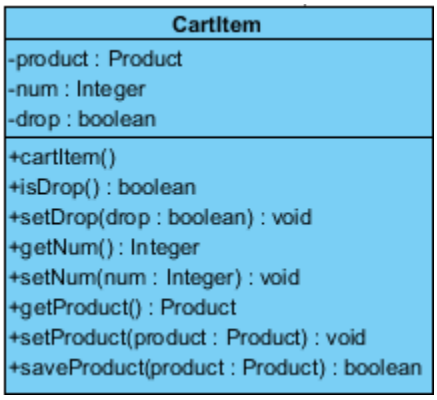
Method	parameter	Description
+addCustomer()	customer : Customer	Calling the method of add customer which from customer DAO implement class.
+updateCustomer()	customer :Customer	Calling the method of update customer which from customer DAO implement class.
+deleteCustomer()	name : String	Calling the method of delete customer where from customer DAO implement class.
+findAllCustomer()	-	Calling the method of find all customers which from customer DAO implement class.



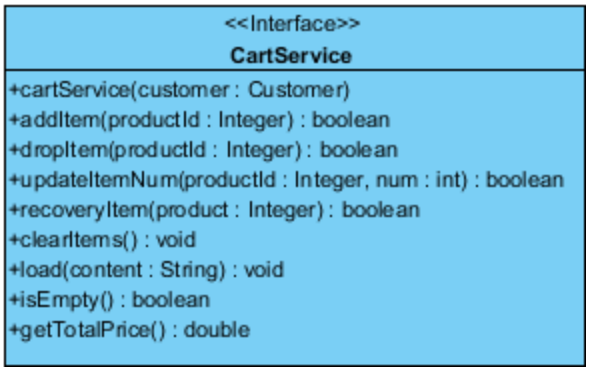
Method	parameter	Description
+addCustomer()	customer : Customer	Implement the method of add customer which from customer service interface.
+updateCustomer()	customer :Customer	Implement the method of update customer where from customer service interface.
+deleteCustomer()	name : String	Implement the method of

		delete customer where from customer service interface.
+findAllCustomer()	-	Implement the method of find customer where from customer service interface.

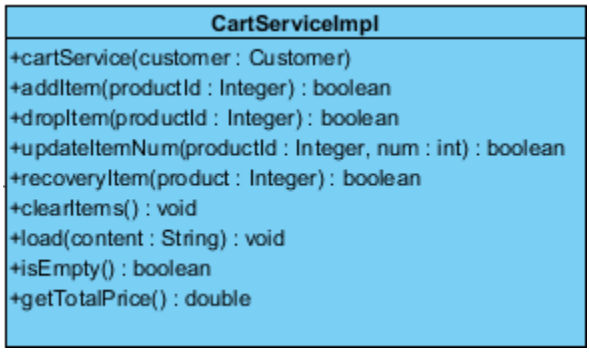
Method	**parameter**	**Description**
+showAllCustomer()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of show all customers which from customer service interface.
+showCustomerInfo()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of show customer information where from customer service interface.



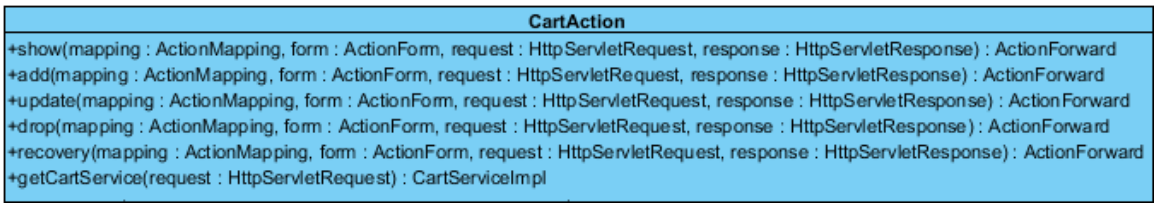
Method	**parameter**	**Description**
+cartItem()	-	This method provides cart item object.
+isDrop()	-	This method check cart item whether or not is drop.
+saveProduct()	product:Product	



Method	parameter	Description
+cartService()	customer : Customer	Calling the method of cart service where from cart DAO implement class.
+addItem()	productId : Integer	Calling the method of add item which from cart DAO implement class.
+dropItem()	productId : Integer	Calling the method of drop item which from cart DAO implement class.
+updateItemNum()	productId : Integer, num : int	Calling the method of update item number where from cart DAO implement class.
+recoveryItem()	product : Integer	Calling the method of recovery item which from cart DAO implement class.
+clearItems()	-	Calling the method of clear item which from cart DAO implement class.
+load()	content : String	Calling the method of load where from cart DAO implement class.
+isEmpty()	-	Calling the method of is empty which from cart DAO implement class.

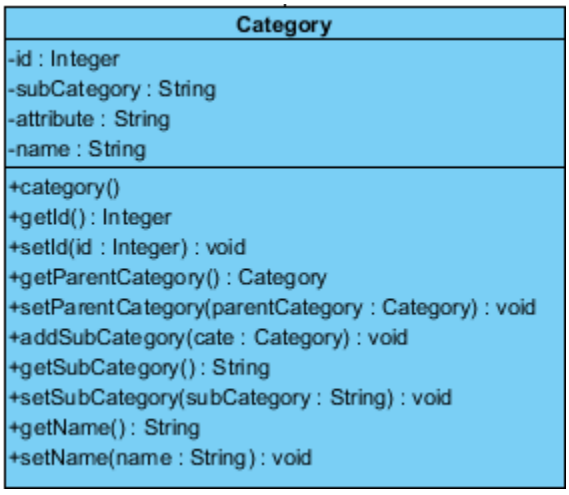


Method	parameter	Description
+cartService()	customer : Customer	Implement the method of cart service where from cart service interface.
+addItem()	productId : Integer	Implement the method of add item which from cart service interface.
+dropItem()	productId : Integer	Implement the method of drop item which from cart service interface.
+updateItemNum()	productId : Integer, num : int	Implement the method of update item number which from cart service interface.
+recoveryItem()	product : Integer	Implement the method of recovery item which from cart service interface.
+clearItems()	-	Implement the method of clear items which from cart service interface.
+load()	content : String	Implement the method of load where from cart service interface.
+isEmpty()	-	Implement the method of is empty which from cart service interface.



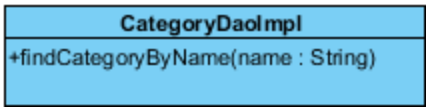
Method	parameter	Description
+show()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of show where from cart service interface.
+add()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of add which from cart service interface.
+update()	mapping : ActionMapping,	Calling the method of update

	form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	which from cart service interface.
+drop()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of drop wherefrom cart service interface.
+recovery()	mapping : ActionMapping, form : ActionForm, request : HttpServletRequest, response : HttpServletResponse	Calling the method of recovery where from cart service interface.



Method	**parameter**	**Description**
+category()	name : String	This method provides category object.

Method	**parameter**	**Description**
+findCategoryByName()	name : String	Calling the method of find category by name which from category DAO implement class.



Method	parameter	Description
+findCategoryByName()	name : String	Implement the method of find category by name where from category DAO interface.

Method	parameter	Description
+getCategoryByName()	name: String	-

Method	parameter	Description
+getCategoryByName()	name: String	-

Method	parameter	Description
+categoryAction()	-	Calling the method of category action where from cart service interface.
+execute()	mapping :ActionMapping, form :ActionForm, request :HttpServletRequest, response :http.HttpServletResponse	Calling the method of execute where from cart service interface.

Traceability Matrix

Traceability Matrix between URS / SRS

URS/SRS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
URS 1	√	√		√	√	√	√	√	√	√	√	√	√	√																				
URS 2															√	√	√	√																
URS 3			√																															
URS 4																			√															
URS 5																				√														
URS 6																					√													
URS 7																						√												
URS 8																							√	√	√	√								
URS 9																											√							
URS 10																												√						
URS 11																													√	√				
URS 12																															√			
URS 13																																√		
URS 14																																	√	
URS 15																																		√

Traceability Matrix between Sequence / URS

Sequence/URS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Register UI	√														
Login		√													
PersonalInfo Edit			√												
Product				√	√	√	√	√	√	√	√	√			√
Add Product													√		
Delete Produce														√	

Traceability Matrix between Sequence / SRS

Sequence/SRS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Register UI	√	√		√	√	√	√	√	√	√	√	√	√	√																				
Login															√	√	√	√																
PersonalInfo Edit			√																															
Product																			√	√	√	√	√	√	√	√	√	√	√	√	√			√
Add Product																																√		
Delete Produce																																	√	

class / SRS	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	34
User																																	√
Customer		√	√										√														√		√				√
Admin																															√	√	
Login		√		√											√	√	√	√															
Registration		√		√	√	√	√	√	√	√		√		√																			
Product																			√	√								√		√	√	√	
ProsuctDao																			√	√										√			
ProductDaoImpl																			√	√										√			
ProductService																			√	√								√		√	√	√	
ProductServiceImpl																			√	√								√		√	√	√	
ProductAction																				√										√	√	√	
Order											√										√						√	√					
OrderDao			√																		√												
OrderDaOImpl			√																		√												
OrderService			√																		√												
OrderServiceImpl			√																		√												

Traceability Matrix between Class / SRS

class / SRS	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	34
OrderAction			√																		√												
CustomerDao	√																																
CustomerDaoImpl	√																																
CustomerService																																	
CustomerServiceImpl																																	
CustomerAction																							√										
CartItem																					√	√								√			
CartService																					√	√								√			
CartServiceImpl																					√	√								√			
CartAction																					√	√	√	√	√					√			
Category																									√					√			
CategoryDao																									√					√			
CategoryDaoImpl																									√					√			
CategoryService																									√					√			
CategoeyServiceImpl																									√					√			
CategoryAction																									√					√			

Traceability Matrix between Class / SRS (cont.)