



Dharmsinh Desai University, Nadiad
Faculty of Technology
Department of Computer Engineering

B. Tech – CE, Semester: IV

Aim: Project report on online pantry shopping (Grocery Factory)

Project title: E-commerce web application for pantry shopping.

Subject: Software Project

Prepared by: Birva Babaria (CE006, 19CEUON064)

Riddhi Chaudhari (CE014, 19CEUTG005)

Guided by: Prof. Pinkal Chauhan

Prof. Brijesh Bhatt

Prof. Jigar Pandya

Contents

1) Abstract.....	03
2) Introduction.....	04
3) Software Requirements Specifications.....	05
4) Design	
a) Use Case diagram.....	12
b) Class diagram.....	13
c) Sequence diagram.....	14
d) Activity diagram.....	16
e) Data flow diagram.....	18
f) Structure diagram.....	21
g) Data dictionary.....	22
5) Implementation Details	
a) Modules.....	25
b) Major Functionality.....	26
6) Testing.....	29
7) Screen-shots.....	32
8) Conclusion.....	38
9) Limitation and future extension.....	39
10) Bibliography.....	39

1) **Abstract**

An online shopping system that permits a customer to submit online orders for items and/or services from a store that serves online customers. The online shopping system presents an online display of an order cut off time and an associated delivery window for items selected by the customer.

The system accepts the customer's submission of a purchase order for the item in response to a time of submission being before the order cut off time. The online shopping system does not settle with a credit supplier of the customer until the item selected by the customer is picked from inventory but before it is delivered. Therefore, the customer can go online and make changes to the order.

When ordering goods, many shopping systems provide a virtual shopping cart for holding items selected for purchase. Successive items selected for purchase are placed into the virtual shopping cart until a customer completes their shopping trip. Virtual shopping carts may be examined at any time, and their contents can be deleted at the option of the customer.

Once the customer decides to submit a purchase order, the customer can track their order from their order history itself. Customer can check their previous order and can cancel the pending order.

2) Introduction

Grocery Factory is online pantry shopping system that permits a customer to shop the products of their requirement. COVID-19 is disrupting India in an unprecedented manner. In order to deal with the Coronavirus, with a country of over 1.3 billion people going into lock-down for a long period of time. “Grocery Factory” is developed to encourage use of online shopping to avoid interaction of people at the store. While essential categories, such as grocery and pharmacy is most important thing that every human being rely on.

Online shopping helps customer to buy the products from any place and the products gets delivered to their doorstep. Each customer has a unique account with which they can proceed shopping. User is authenticated before login to the system. Grocery Factory provides major functionalities that gives user an amazing experience to shop online.

Technologies/Tools used

Technologies:

- Django
- Python
- MySQL
- Html
- Jinja
- CSS
- Bootstrap

Tools:

- Git
- Visual studio code

3) Software Requirement Specifications

1) Manage customers

R.1.1: Profile

Description: This section includes every information related to customers and their purchases.

R.1.1.1: View profile

Description: This option shows the profile of the customer. It displays their username, address, email id, update profile options.

Input: Click on the view profile option.

Output: Profile screen would be displayed.

R.1.1.2: Change shipping address

Description: This function will let the customer to update their shipping address.

Input: Click on the change shipping address option.

Output: New page regarding update details would appear. User is supposed to enter new details and click on the save option.

R.1.1.3: Change password

Description: This function will let the customer to change their password.

Input: Click on the change password option.

Output: New page regarding update details would appear. User is supposed to enter new password and click on the save option.

R.1.1.4: View previous orders

Description: This function will let the customer to view their order history

Input: Click on the view previous orders option.

Output: New page will be displayed will all the previous orders information and status of the order.

R.1.2: About us page

Description: This function will let customer to see the information about website and can contact admin through mail or contact number.

Input: Click on the about us option from the navigation bar.

Output: About us page would be displayed.

2) Manage registration/login

R.2.1: Customer

R.2.1.1: Registration

Description: If customer doesn't have any exiting account then they have to register themselves to purchase products. Customer will be able to just check up on the site without registration but for purchasing an item they have to login or register.

Input: Customer have to provide their name, address, email, and password.

Output: User would be redirected to login page.

R.2.1.2: Login

Description: If customer already have an account then this option will be used in purchase process.

Input: Customer have to give their username and password.

Output: The home page will appear.

R.2.1.3: Forget password

Description: If customer already have an account and forgets the password during login then he/she can reset the password.

Input: Customer have to enter the registered email and click on ‘send email’ option.

Output: Email will be sent to registered id with a link to password reset page and from there customer can reset their password.

3) Manage product

R.3.1: View product

Description: This function will show the description of the product. It includes full name of the product, brand name, price, detailed information about the product.

Input: Click on the product directly.

Output: All the details mentioned above would be displayed on the screen.

R.3.2: Buy now

Description: This function will let customer to buy the product directly without adding it to the cart.

Input: Click on buy now icon on the product.

Output: Place order page will be displayed with all the information about the order.

R.3.3: Add to cart

Description: This function will let customer to add that particular product to the cart.

Input: Click on add to cart icon on the product.

Output: Product would be added to the cart and cart icon would be incremented by one.

4) Manage cart

R.4.1: View cart

Description: This function will display the cart and will show all the products that customers have added. If the cart is empty then no product would be viewed.

Input: Click on the cart icon from the navigation bar.

Output: Another page for the cart would be displayed.

R.4.1.1: Placing the order

Description: After entering to cart this function will let the user to buy all the products that he/she added to the cart.

Input: Click on 'Place order' option.

Output: Bill will be generated and will ask to confirm order.

R.4.1.2: Removing the product

Description: This function will let customer to remove the product if they do not want to buy it after adding it to the cart.

Input: Click on the 'remove' option.

Output: The product will be disappeared from the cart.

R.4.2: Add to Cart

Description: When customers don't want to purchase one particular product but want to add it in cart for future reference then they can use this option. Cart will store all the items that customer wants to purchase.

Input: Click on 'Add to cart' option.

Output: The product would be added in the cart.

R.4.3: Calculate price

Description: This function will automatically display the total price of all the products present in the cart.

Output: After the end of the list of all the products total price would be displayed including the tax.

5) Manage orders

R.5.1: Delivery date

Description: After the order has been placed by the user, delivery date would be displayed to the user. It would be approximately after 1-2 days after the ordered date.

Input: Click on confirm order option.

Output: All the details regarding delivery would be displayed after clicking to 'your orders' link.

R.5.2: Cancel order

Description: This function will let the user to cancel the order after the order has been placed and if delivery of the order is pending.

Input: Click on 'cancel order' option.

Output: Cancelled order would be disappeared from the users order history.

R.5.3: Order status

Description: This function will show the status of the order. It can be 'pending' or 'delivered' as per its status.

Input: Go to previous orders in profile section.

Output: It will display the appropriate message as per delivery date.

R.5.4: Buy now

Description: If customer wants to purchase product directly by looking at the details of the product then this option will be useful.

Input: Click on 'Buy now' option.

Output: User will see place order page where he/she can proceed towards payment.

6) Manage payment

R.6.1: Payment method

R.6.1.1: Cash on delivery

Description: Customer have to pay products price at delivery time with cash.

Output: Order will be placed without paying the money. Confirmation would be sent to user.

7) Logout user

R.7.1: logout

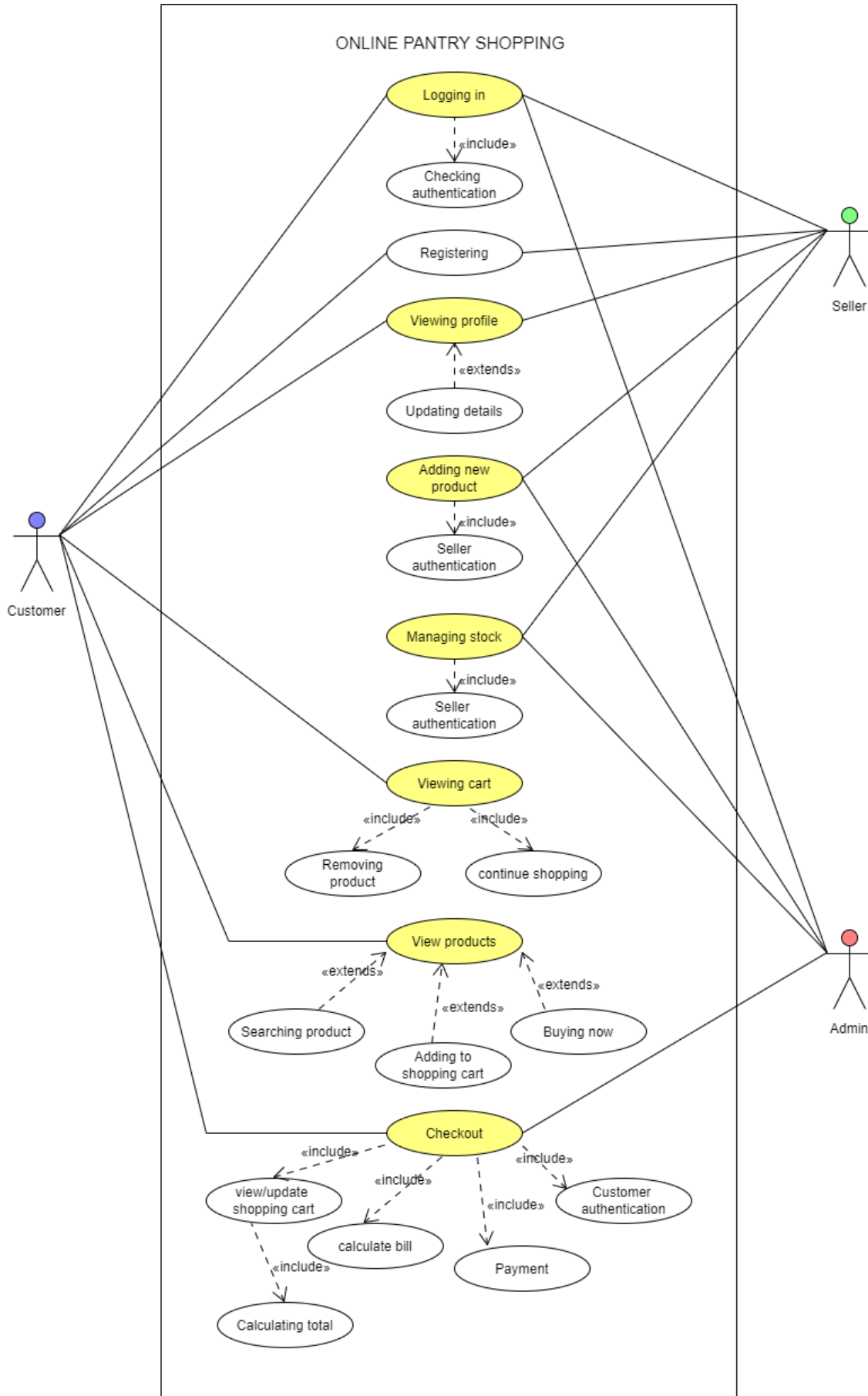
Description: Customer can logout after completion of shopping if he/she wants. The previous orders and all the information related to customer would be saved till next login.

Input: Click on logout option in navigation bar.

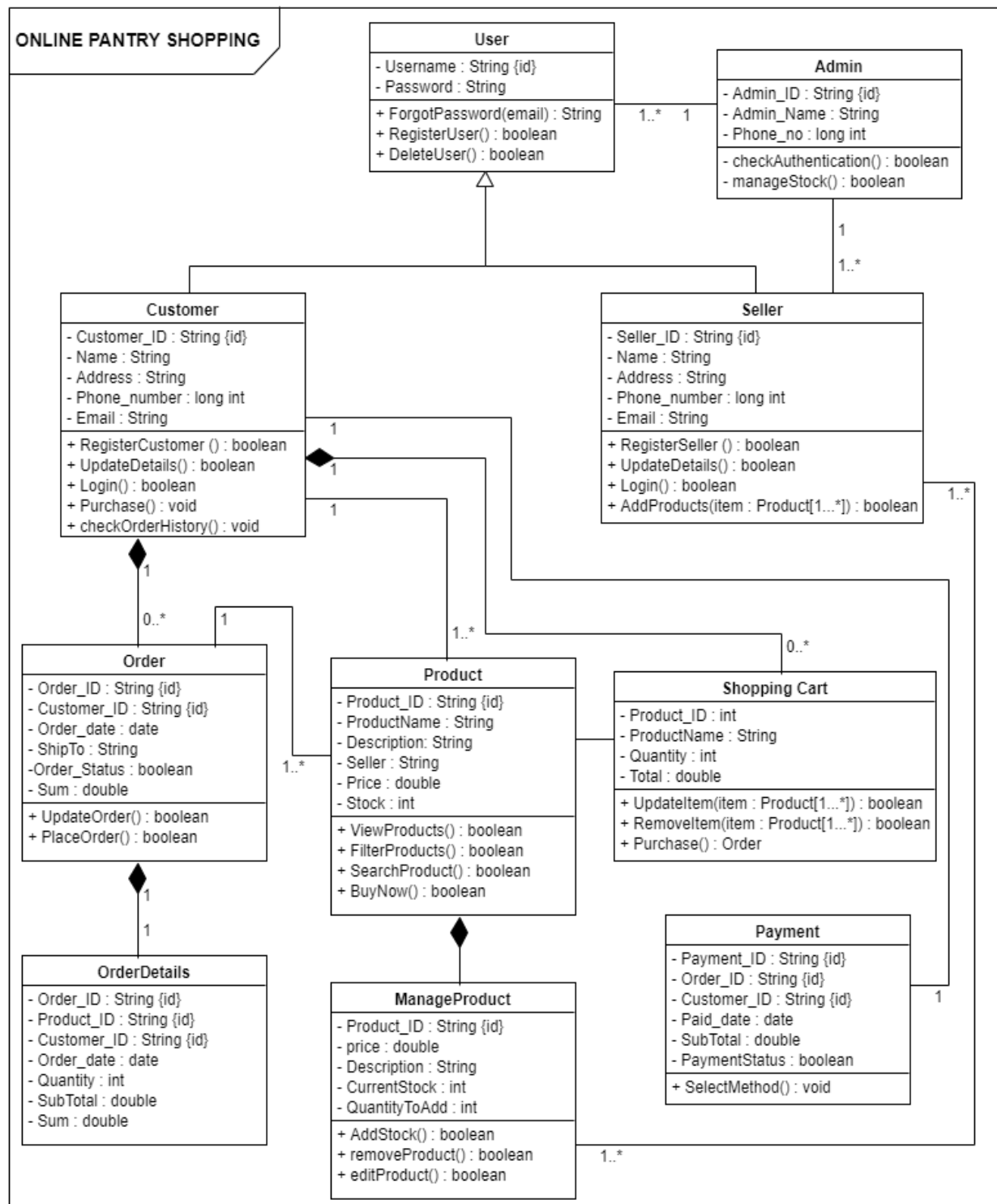
Output: Customer would be redirected to logout page with a link to login again.

4) Design

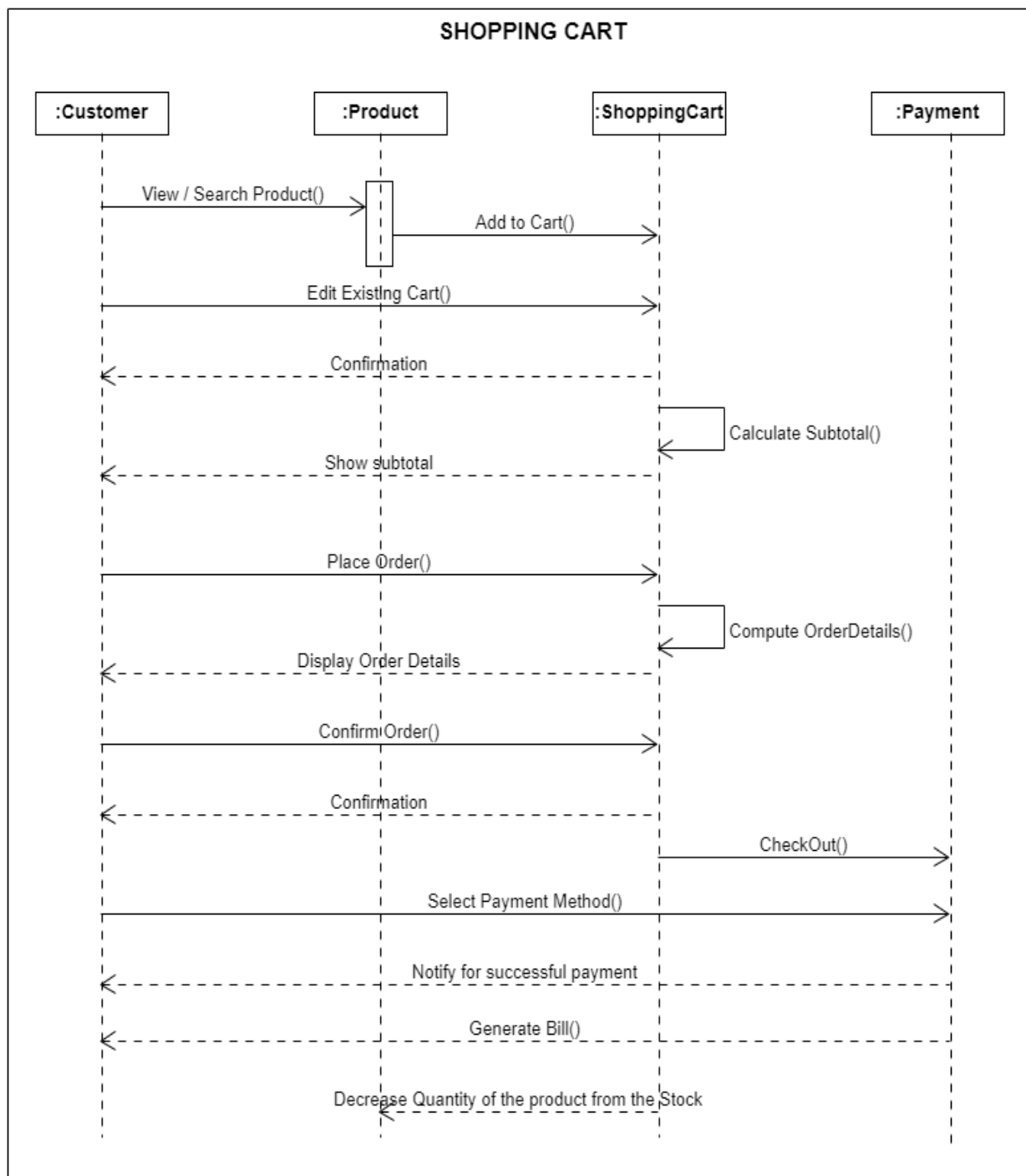
a) Use case diagram

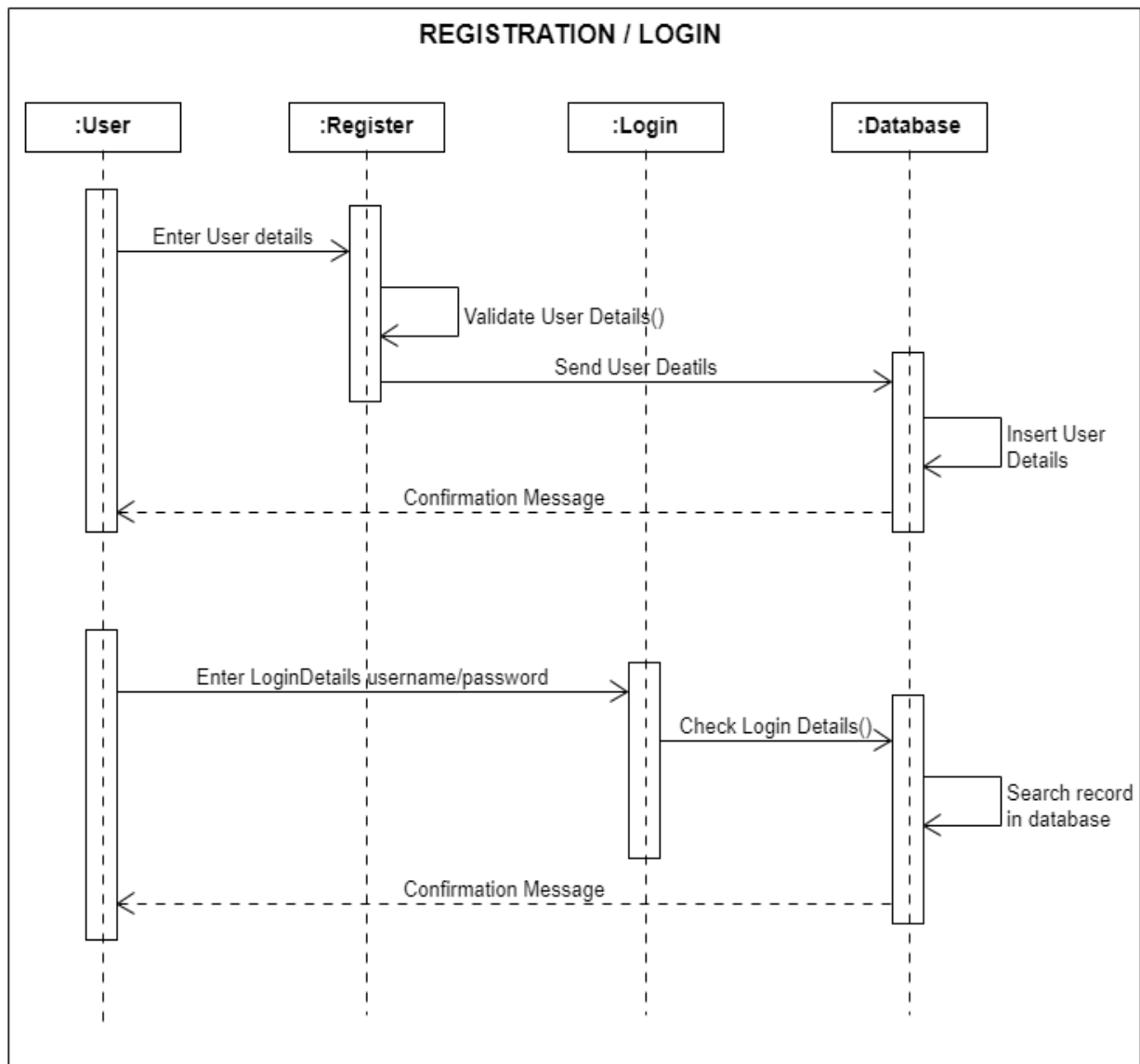


b) Class diagram

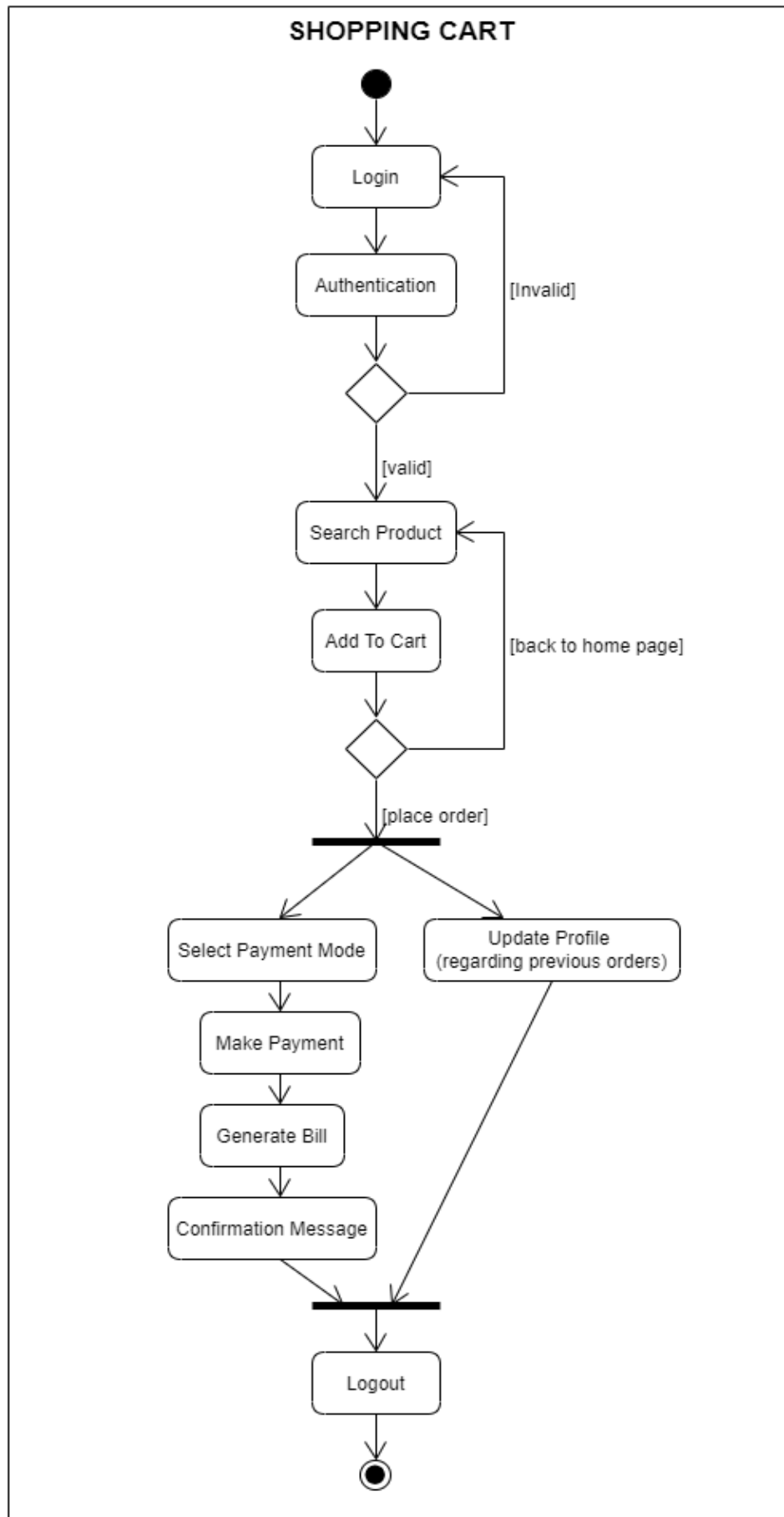


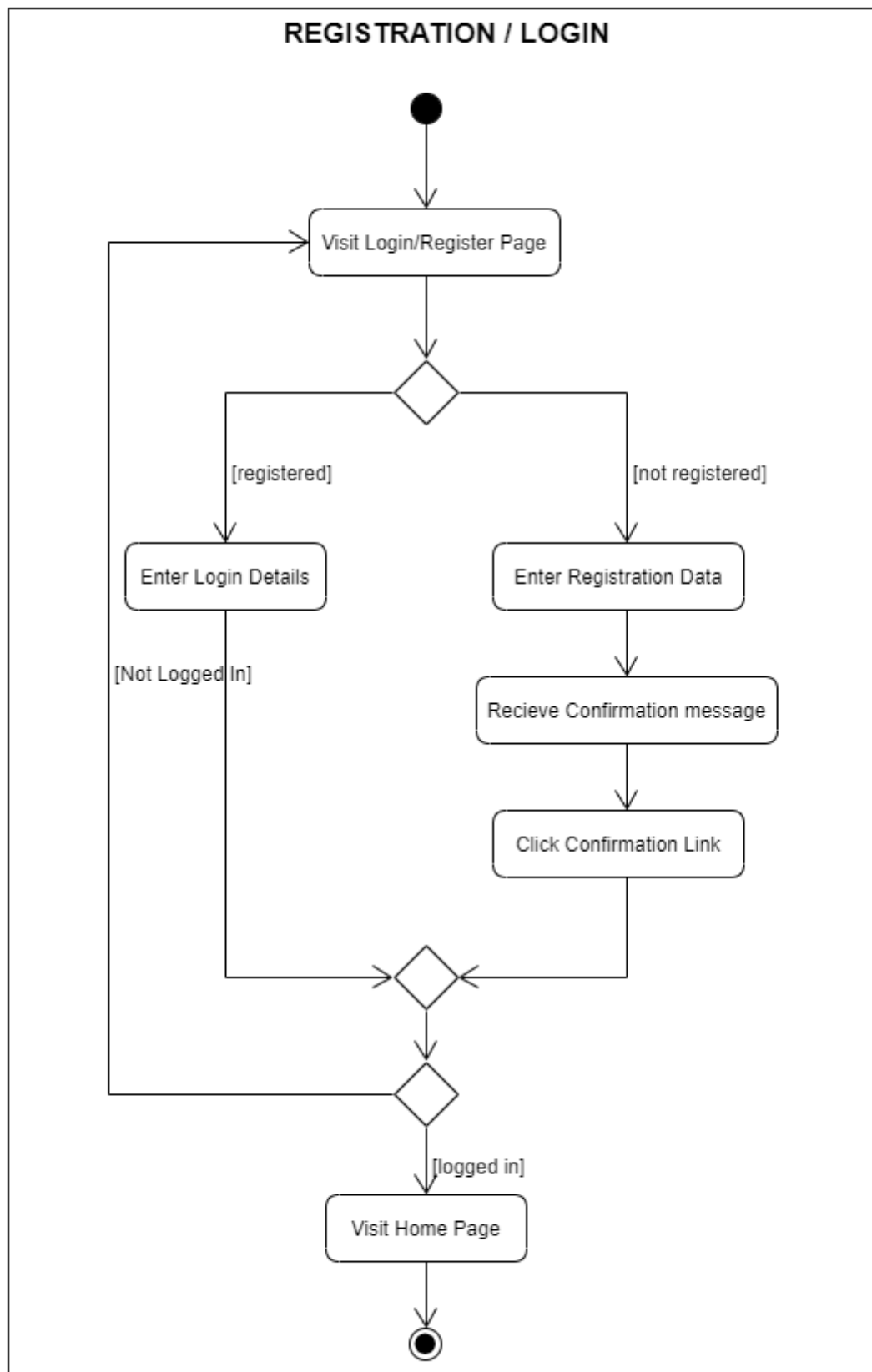
c) Sequence diagram





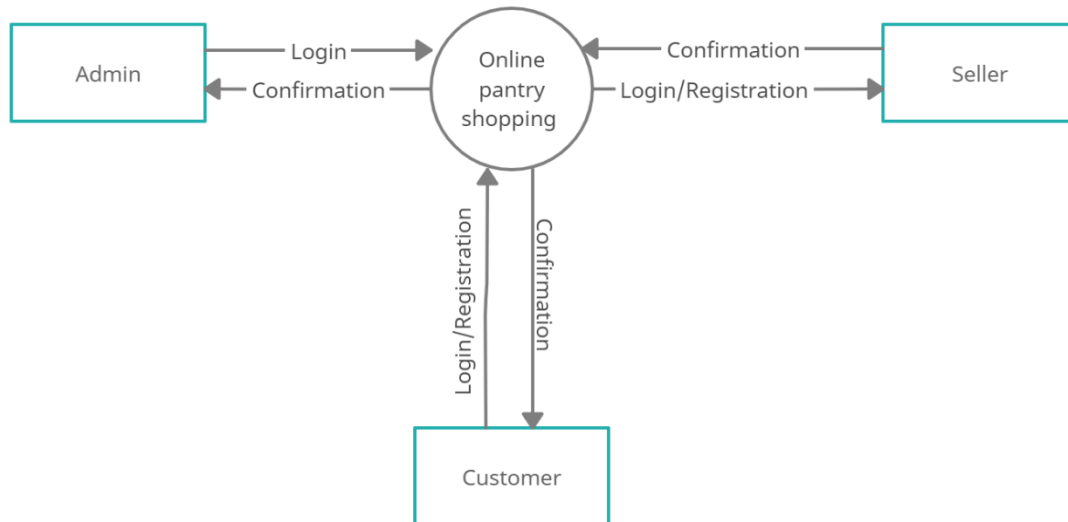
d) Activity diagram



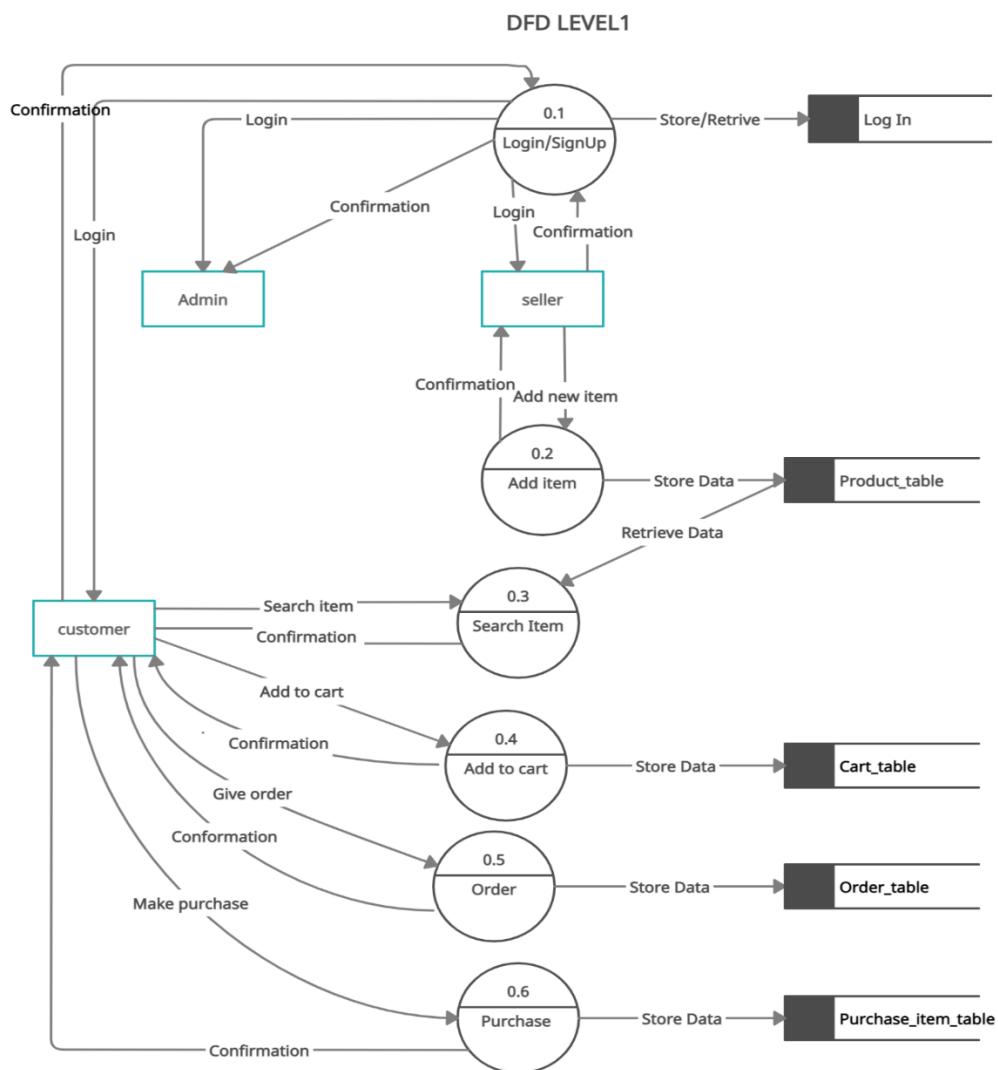


e) DFD model

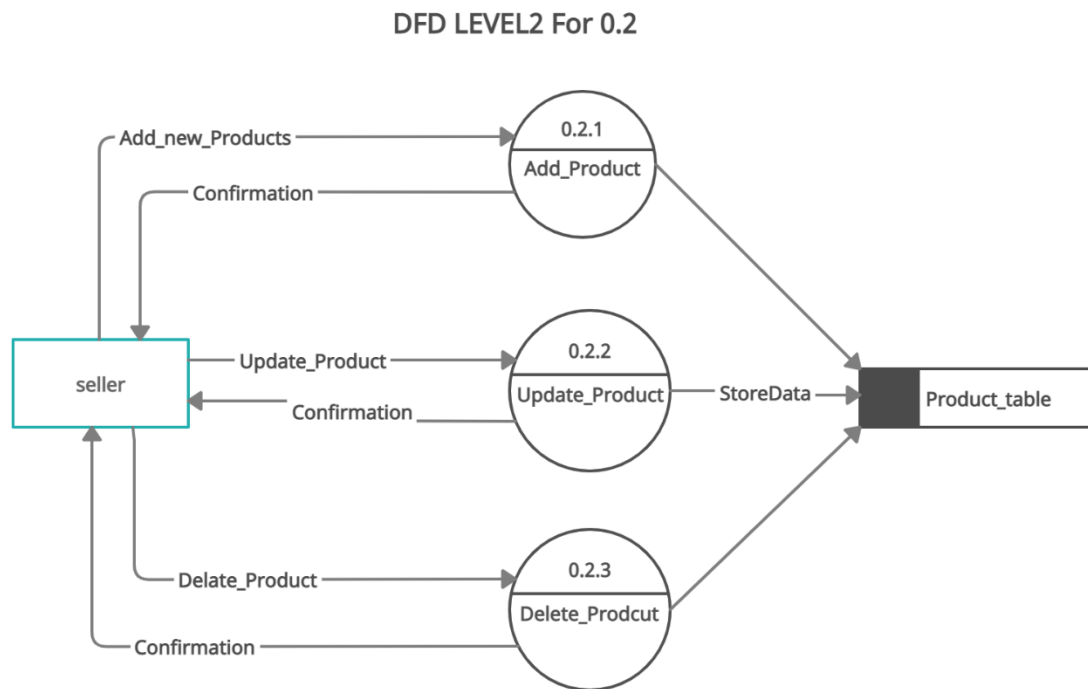
- Context diagram (Level 0)



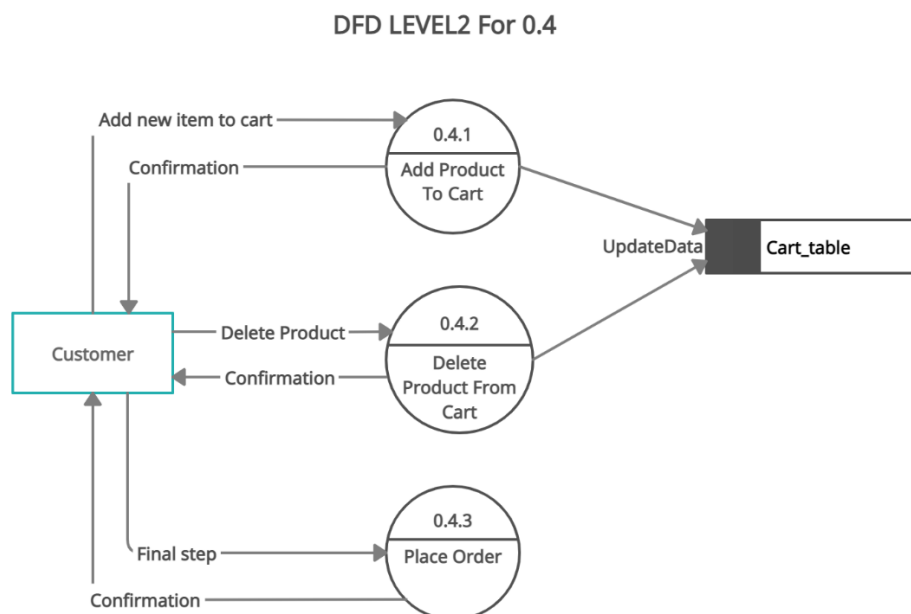
- Level 1 DFD



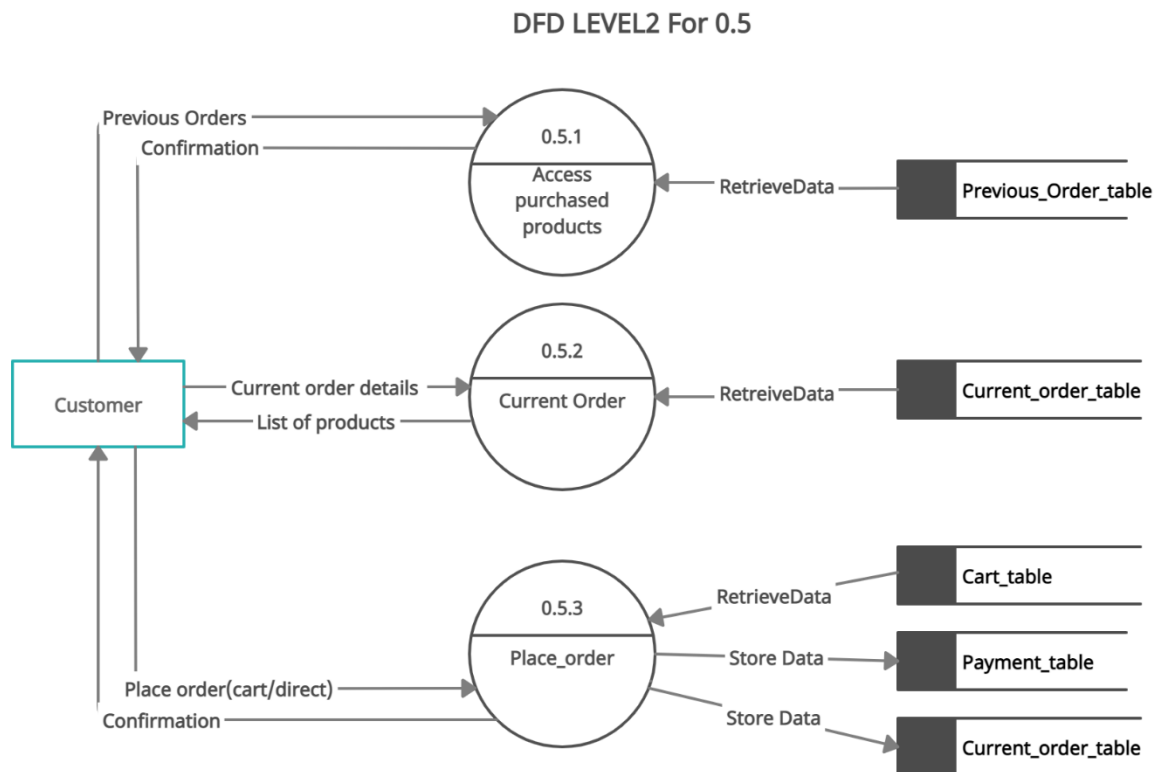
- Level 2 DFD for 0.2



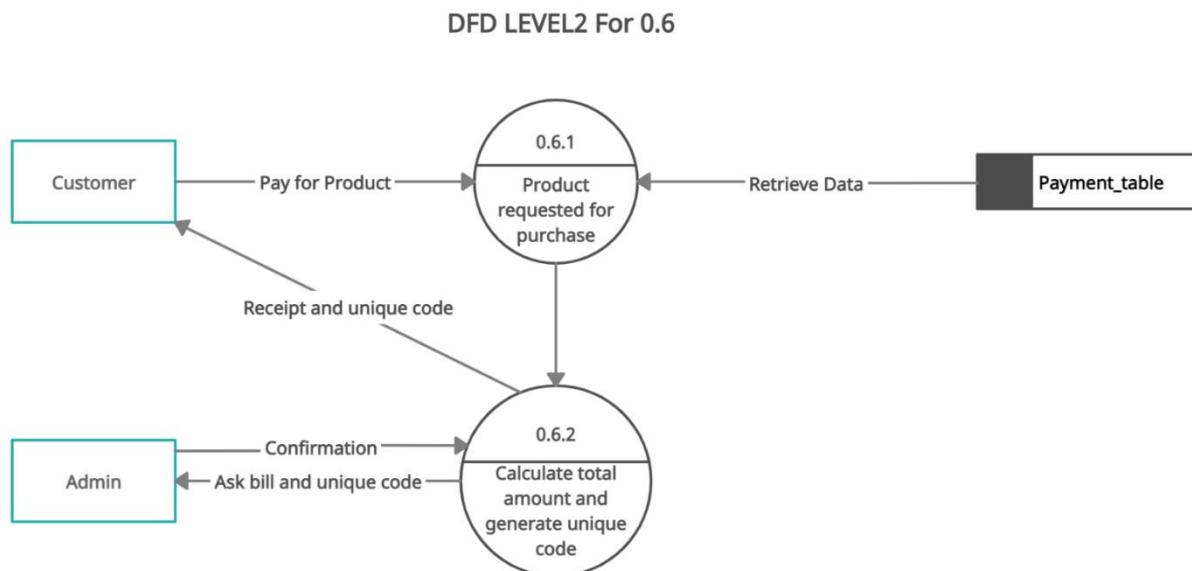
- Level 2 DFD for 0.4



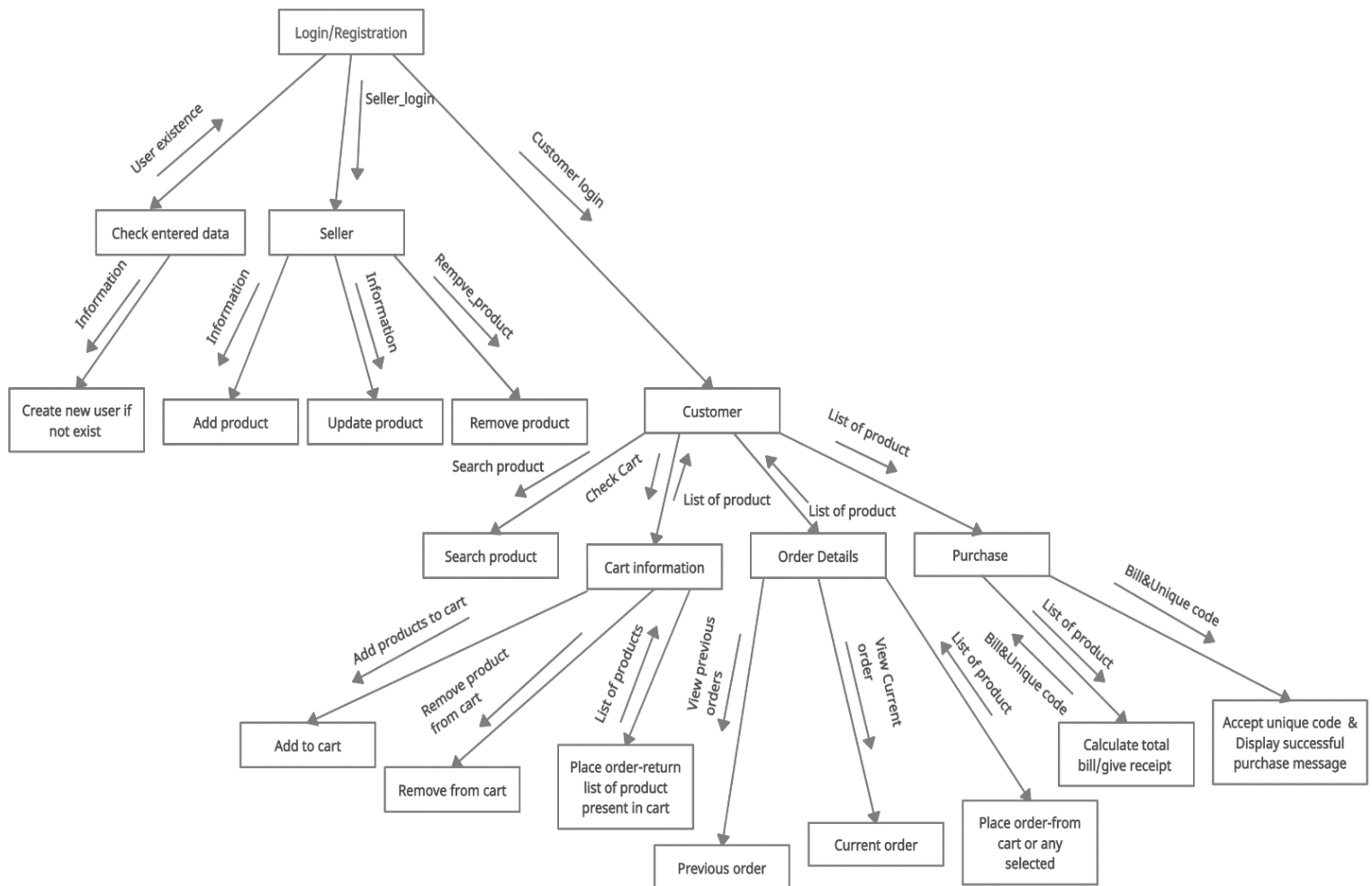
- Level 2 DFD for 0.5



- Level 2 DFD for 0.6



f) Structure diagram



g) Data dictionary

customer								
Sr No.	Field name	Data type	width	Required	Unique	PK/FK	Referred table	Description
1	cust_id	int	-	yes	yes	yes	-	Auto increment
2	username	varchar	30	yes	yes	no	-	-
3	address	varchar	100	yes	no	no	-	-
4	email	varchar	50	yes	no	no	-	-
5	password	varchar	10	yes	yes	no	-	-

product								
Sr No.	Field name	Data type	width	Required	Unique	PK/FK	Referred table	Description
1	prod_id	int	-	yes	yes	yes	-	Auto increment
2	description	varchar	200	yes	no	no	-	-
3	price	float	-	yes	no	no	-	-
4	image	varchar	100	yes	yes	no	-	-

shoppingcart								
Sr No.	Field name	Data type	width	Required	Unique	PK/FK	Referred table	Description
1	cart_id	int	-	yes	yes	yes	-	Auto increment
2	cart_prod_id	int	-	yes	no	yes	product	-
3	cart_cust_id	int	-	yes	no	yes	customer	-

order								
Sr No.	Field name	Data type	width	Required	Unique	PK/FK	Referred table	Description
1	order_id	int	-	yes	yes	yes	-	Auto increment
2	order_cust_id	int	-	yes	no	yes	customer	-
3	price	float	-	yes	no	no	-	-
4	orderdate	date	-	yes	no	no	-	-
5	deliverdate	date	-	yes	no	no	-	-
6	status	varchar	20	yes	no	no	-	Default= pending

orderitems								
Sr No.	Field name	data type	width	Required	Unique	PK/ FK	Referred table	Description
1	orderitems_id	int	-	yes	yes	yes	-	Auto increment
2	orderitems_order_id	int	-	yes	no	yes	order	-
3	orderitems_prod_id	int	-	yes	no	yes	product	-

5) **Implementation Details**

a). Modules

The system consists of four basic modules:

- 1) loginmodule
- 2) shoppingmodule
- 3) authen

Each module consists of several methods to implement the required Functionality. Implementation is done using Django. Database used in these modules is MySQL.

loginmodule

This module is the base for authentication and authorization to ensure the security aspect of the user.

It consist of all the login and registration functionality. User can login to the system if account already exist. User can register with unique username to create an account.

User can use forget password functionality if he/she forgets the password. An email is received to the users registered email address with the link of reset password page.

User can logout from the system whenever he/she wants. This functions would not delete the activities performed by user.

shoppingmodule

This module handles the functions related to shopping. User can add the products to the cart and can also edit the cart.

Order can be placed from the cart or directly from the home page. Pending orders can be cancelled if user wants.

In profile section user can view their profile and can change shipping address, password as per their requirement. User can view their order history and can see the status of orders.

In about us section user can contact admin directly form mail or contact number.

authen

This module consist of all the models required for the system.

b). Major functionalities

- Cart function

```
25 def addtocart(request):
26     p_id = request.GET['p_id']
27     c = shoppingcart(cart_cust_id=customer.objects.get(cust_id = request.session['cust_id']), cart_prod_id=product.objects.get(prod
28     c.save()
29     return HttpResponseRedirect('/')
30
31 def viewcart(request):
32     prods = shoppingcart.objects.all().filter(cart_cust_id = customer.objects.get(cust_id = request.session['cust_id']))
33     price = 0.0
34     TAX = 19.05
35     for prod in prods:
36         price = price + prod.cart_prod_id.price
37     subtotal = price + TAX
38     return render(request, 'cart.html', {'prods': prods, 'price': price, 'TAX': TAX, 'subtotal': subtotal})
39
40 def removefromcart(request):
41     c_id = request.GET['c_id']
42     prod = shoppingcart.objects.get(cart_id = c_id)
43     prod.delete()
44     return HttpResponseRedirect('/viewcart/')
45
```

- Reset password urls

```
15 #reset password urls
16 path('reset_password/',
17     auth_views.PasswordResetView.as_view(template_name = 'passwordreset.html'),
18     name='reset_password'),
19
20 path('reset_password_sent/',
21     auth_views.PasswordResetDoneView.as_view(template_name = 'passwordresetsent.html'),
22     name='password_reset_done'),
23
24 path('reset/<uidb64>/<token>/',
25     auth_views.PasswordResetConfirmView.as_view(template_name = 'resetpasswordform.html'),
26     name='password_reset_confirm'),
27
28 path('reset_password_complete/',
29     auth_views.PasswordResetCompleteView.as_view(template_name = 'resetpasswordcomplete.html'),
30     name='password_reset_complete'),
31
```

- Confirm order function

```
78 def confirmorder(request):
79     prods = shoppingcart.objects.all().filter(cart_cust_id = customer.objects.get(cust_id = request.session['cust_id']))
80     subtotal = 0.0
81     for prod in prods:
82         subtotal = subtotal + prod.cart_prod_id.price
83     subtotal = subtotal + 19.05
84     o = order(order_cust_id=customer.objects.get(cust_id=request.session['cust_id']), price=subtotal)
85     o.save()
86     ord = order.objects.latest('order_id')
87
88     for prod in prods:
89         id = prod.cart_prod_id.prod_id
90         o_item = orderitems(orderitems_order_id=ord, orderitems_prod_id=product.objects.get(prod_id=id))
91         o_item.save()
92     for prod in prods:
93         prod.delete()
94     return HttpResponseRedirect('/viewconfirmorder/')
95
96 def confirmorderb(request):
97     p_id = request.GET['p_id']
98     prod = product.objects.get(prod_id = p_id)
99     subtotal = prod.price + 19.05
100     o = order(order_cust_id=customer.objects.get(cust_id=request.session['cust_id']), price=subtotal)
101     o.save()
102     ord = order.objects.latest('order_id')
103     o_item = orderitems(orderitems_order_id=ord, orderitems_prod_id=prod)
104     o_item.save()
105     return HttpResponseRedirect('/viewconfirmorder/')
106
107 def viewconfirmorder(request):
108     return render(request, 'confirmorder.html')
```

- Order history

```
163 def vieworders(request):
164     ords = order.objects.all().filter(order_cust_id=customer.objects.get(cust_id=request.session['cust_id']))
165     date = timezone.now()
166     for o in ords:
167         if(date >= o.deliverdate):
168             o.status = "delivered"
169             o.save()
170     items = []
171     for o in ords:
172         items.append(orderitems.objects.all().filter(orderitems_order_id=o))
173     mylist = zip(items, ords)
174     if 'cancel' in request.session:
175         context = {
176             'list': mylist,
177             'cancelmsg': "Order cancelled successfully"
178         }
179         del request.session['cancel']
180     else:
181         context = {
182             'list': mylist
183         }
184     return render(request, 'previousorders.html', context)
```

- Change address and password

```
130 def changeaddress(request):
131     return render(request, 'changeaddress.html')
132
133 def addnewaddress(request):
134     addr = request.POST['newaddress']
135     cust = customer.objects.get(cust_id = request.session['cust_id'])
136     cust.address = addr
137     cust.save()
138     request.session['addresschanged'] = "Address changed successfully"
139     return HttpResponseRedirect('/viewprofile/')
140
141 def changepassword(request):
142     return render(request, 'changepassword.html')
143
144 def updatepassword(request):
145     oldpass = request.POST['oldpass']
146     newpass = request.POST['newpass']
147     cnewpass = request.POST['cnewpass']
148     u = User.objects.get(username=request.session['cust_name'])
149     if not check_password(oldpass, u.password):
150         return render(request, 'changepassword.html', {'msg': "*Old password is incorrect"})
151     elif(newpass != cnewpass):
152         return render(request, 'changepassword.html', {'msg': "*New passwords does not match"})
153     else:
154         user = User.objects.get(username=request.session['cust_name'])
155         user.set_password(newpass)
156         user.save()
157         cust = customer.objects.get(cust_id=request.session['cust_id'])
158         cust.password = newpass
159         cust.save()
160         request.session['passwordchanged'] = "Password changed successfully"
161         return HttpResponseRedirect('/viewprofile/')
162
```

- Register user

```
30 def register_user(request):
31     return render(request, 'register.html')
32
33 def register(request):
34     username = request.POST.get('username', '')
35     password = request.POST.get('password', '')
36     cpassword = request.POST.get('cpassword', '')
37     email = request.POST.get('email', '')
38     address = request.POST.get('address', '')
39     if(password != cpassword):
40         return render(request, 'register.html', {'msg': "*Passwords doesn't match"})
41     newuser = customer.objects.filter(username=username).first()
42     if newuser is not None:
43         return render(request, 'register.html', {'msg': "*username already taken"})
44     cust = customer(username = username, address = address, email = email, password = password)
45     cust.save()
46     user = User.objects.create_user(username = username, email = email, password = password)
47     return HttpResponseRedirect('/loginmodule/login/')
48
```

6) Testing

Testing method: Manual testing

Manual testing was performed in order to find and fix the bugs in development process.

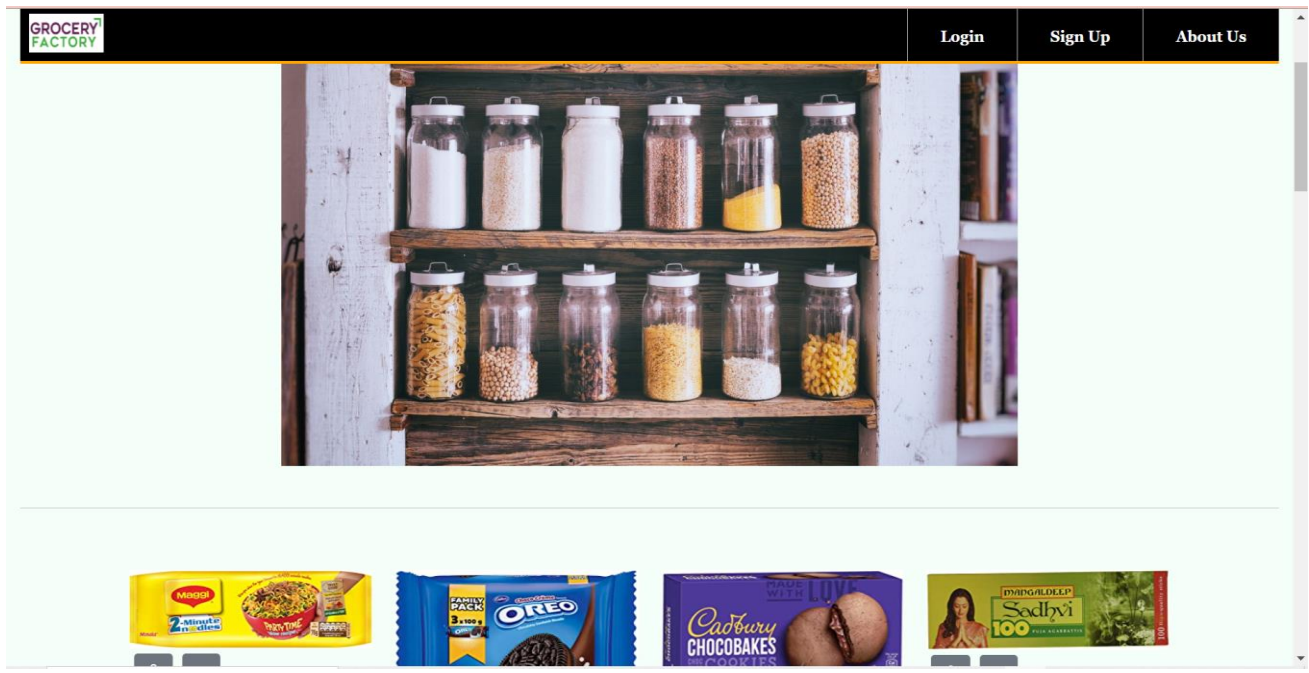
Sr no.	Test Scenario	Expected result	Actual result	Status
1	Enter to the website	Home page should be displayed with login and signup option in navigation bar	Home page is displayed with login and signup option in navigation bar	Success
2	Login with correct credentials	User should be able to login	User is redirected to home page	Success
3	Login with incorrect credentials	User should be redirected to invalid login page	User is redirected to invalid login page	Success
4	Registering with registered username	'username taken' message should be displayed	'username taken' message is displayed	Success
5	Entering different password and confirm password	'password does not match' message should be displayed	'password does not match' message is displayed	Success
6	Sending email on registered email id for reset password	Email with a link should be received	Email with a link is received and is redirected to reset password page	Success
7	Click on buy now icon	Bill page should be displayed with subtotal including tax	Bill page is displayed with subtotal including tax	Success
8	Click on add to cart icon on product	Product should be added to the cart	Product is added to the cart and total items in the cart icon is incremented	Success

9	Click on view cart icon in navigation bar	Cart should be viewed with all the items added by the customer	Cart is displayed with all the products added by the user and place order option is visible	Success
10	Remove product from cart	Particular product should be removed from the cart	Product is removed from the cart and existing products are shown in cart page	Success
11	Confirm order	Order should be placed successfully	Order is placed successfully and it can be tracked from order history page	Success
12	Click on profile icon in navigation bar	Profile information should be displayed	Profile information is displayed with change address, password and view previous order option	Success
13	Change password	It should display change password page	Change password page is displayed	Success
14	Entering incorrect credentials in change password page	Warning message should be displayed on the same page	Warning message of 'old password incorrect' or 'password doesn't match' is displayed	Success
15	Entering correct credentials	Password should be changed successfully	Password is changed successfully and confirmation message is displayed on profile page	Success
16	Change shipping address	Address should be changed successfully	Address is changed successfully with a confirmation message on profile page	Success

17	View previous orders	Order history should be displayed with the status of delivery	Order history is displayed with the status 'pending' or 'delivered' with cancel order button	Success
18	Cancel order	Order should be cancelled successfully	Order is cancelled successfully with a confirmation message in same page	Success
19	Click on about us link	About us page should be displayed	About us page is displayed and user can contact admin through mail or contact number	Success
20	Logout	User should logged out from the system	User is logged out from the system and cannot use the system until next login	Success

7) Screen-shots


- Home page



- Login page

The screenshot shows the login page of the 'GROCERY FACTORY' website. The header is black with the 'GROCERY FACTORY' logo on the left and an 'About Us' link on the right. The main content area has a light green background. At the top, it says 'Please login/register to shop!' followed by the heading 'Login'. Below this, there are two input fields: 'User Name : ' and 'Password : '. A light blue 'login' button is centered below the password field. At the bottom, there are two links: 'Forgot password? [reset password](#)' and 'New User? [register](#)'.

- Sign up page

	About Us
---	--------------------------

Register

User Name :

Email :


Address :

Password :

Confirm Password :

Register


- Reset password

	About Us
--	--------------------------

Enter your registered email address
We'll mail instructions for setting a new one.

Email:

Send Email

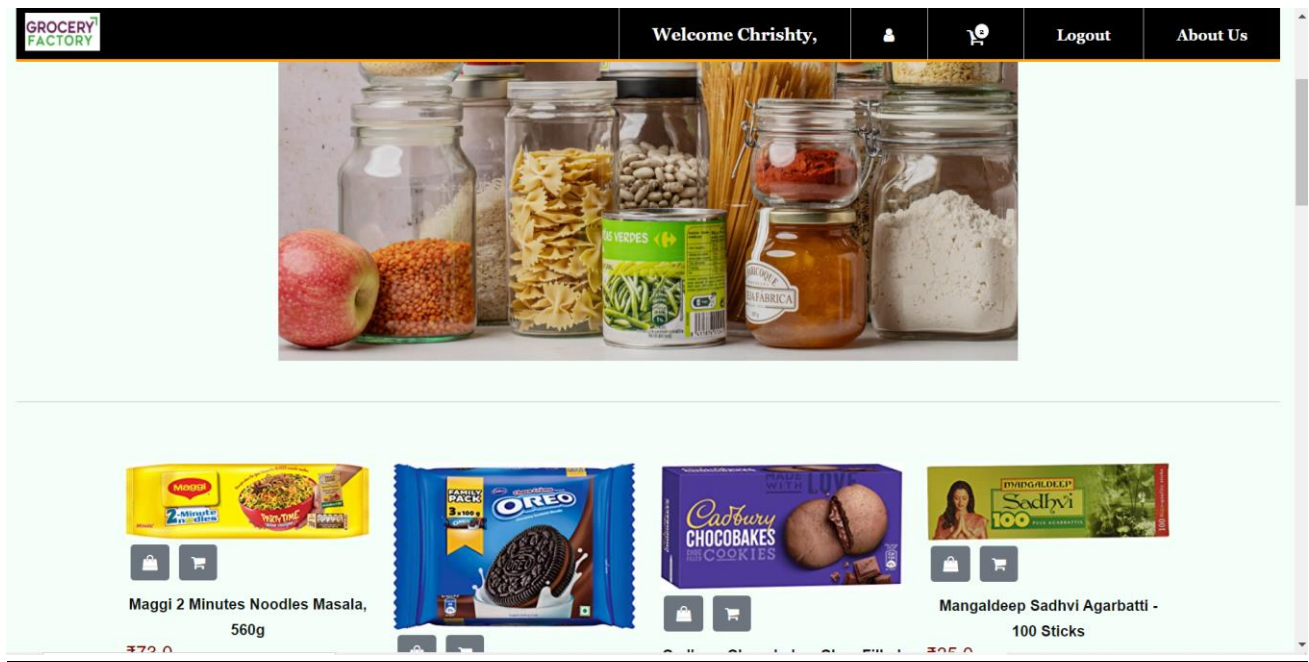
	About Us
---	--------------------------

Password reset sent

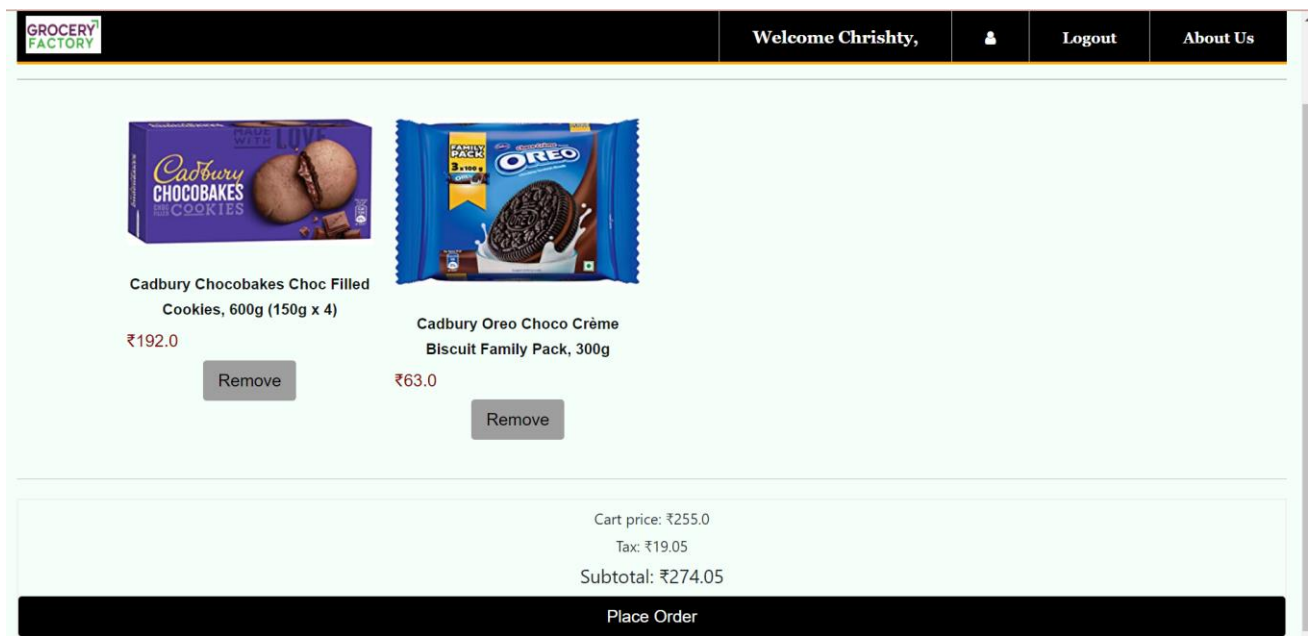
We've emailed you instructions for setting your password, if an account exists with the email you entered. You should receive them shortly.

If you don't receive an email, please make sure you've entered the address you registered with, and check your spam folder.


- Home page after login



- Cart



- Bill

GROCERY FACTORY	Welcome Chrishty,		Logout	About Us
----------------------------	--------------------------	---	---------------	-----------------

Order Details!

Username: Chrishty

Email: c123@gmail.com

Shipping Address: A-502 shreepad antiliya

Products :

- Cadbury Chocobakes Choc Filled Cookies, 600g (150g x 4) --> ₹192.0
- Cadbury Oreo Choco Crème Biscuit Family Pack, 300g --> ₹63.0

*free shipping

Payment mode: Cash on delivery


Total price: ₹255.0

Tax: ₹19.05

Subtotal: ₹274.05


Confirm Order

- Confirm order

GROCERY FACTORY	Welcome Chrishty,		Logout	About Us
----------------------------	--------------------------	---	---------------	-----------------



Your order is confirmed :)

Please check [your orders](#) for more information.




Click [here](#) to continue shopping :)

- Profile page


Welcome Chrishty,

Logout
About Us

My Profile




Username:	Chrishty
Email:	c123@gmail.com
Address:	A-502 shreepad antiliya

View Previous Orders
Change Shipping Address

- Change address and password


My Profile



Enter new address:

Save Changes
[Go Back](#)

My Profile



Enter old password:

Enter new password:

Confirm password:

Change Password
[Go Back](#)

- Order history


Welcome Chrishty,


Logout
About Us

● delivered

Order ID: #GF022

Description: Fortune Xpert (Vivo) Pro Sugar Conscious Edible Oil, Jar, 5L

Order Date: March 27, 2021, 12:28 p.m.
Delivery Date: March 28, 2021, 5:36 a.m.
Payment mode: COD
Subtotal: ₹758.05


Welcome Chrishty,

Logout
About Us

Order history!

Cancel Order

● pending

Order ID: #GF027



Description: Cadbury Chocobakes Choc Filled Cookies, 600g (150g x 4)



Description: Cadbury Oreo Choco Crème Biscuit Family Pack, 300g


Order Date: March 30, 2021, 11:51 a.m.

Delivery Date: March 31, 2021, 4:27 a.m.

Payment mode: COD


Subtotal: ₹274.05


- About us page


Welcome Chrishty,

Logout
About Us

GOCERY removes from you the routinely repetitive tasks of grocery shopping and allows you to focus your time on what really matters.






Let the goods come to you

Contact Us:

Email:[Send mail](#)

Phone no.:(311)55-2368

- Logout



You are logged out!!

Click [here](#) to login again.

8) Conclusion

The functionalities that are implemented in the system are prepared after understanding all modules according to software requirements specifications (SRS). Functionalities that are successfully implemented in the system are as follows:

- Login
- Forget password functionality (password reset)
- New user sign up
- User authentication
- Logout
- Buy now
- Add to cart
- Remove product from cart
- Place order
- Cancel order
- View profile for a particular user
- Change password
- Change shipping address
- View previous orders (order history)
- View status of orders

After implementing all these functionalities, comprehensive testing was performed on the system to determine possible errors.

9) **Limitations and future extensions**

Technology has made significant progress over the years to provide consumers a better online shopping experience and will continue to do so for years to come. Meanwhile many new technologies and functionalities are included rapidly.

The major demerit of the online shopping is lack of interaction. Customer cannot bargain about the price of the product. Delay in delivering the product may happen due to some reasons.

This project of Grocery Factory can be extended to support appropriate payment system. Quantity of the product can be managed properly. We can add functionality to search the product directly from the navigation bar.

10) **Bibliography**

Following links and websites were referred during the development of this project:

- <https://docs.djangoproject.com/en/3.1/>
- <https://github.com/>
- <https://stackoverflow.com/>
- <https://www.djangoproject.com/>