Team: Antara Kolar
         Hariram Natarajan
         Shriya Rodi

Title: QuickShop

Project Summary:
This is an online grocery delivery system that saves a customer's shopping time. Admins, Customers and Sellers will have their own account. There will be multiple supermarket retailers options (Safeway, Whole Foods Market, King Soopers) where sellers can list and add , remove or modify their products from product list. Customers can select the retailer and add groceries to their cart.Customers can choose only one store at a time for an order. Customers can search products, buy, rate and review products ,pay for their orders. Products will have two options for shipping method : home delivery and take away.

Business Requirements:

ID	Requirement	Topic Area	Actor	Priority
BR-01	Customers should have option to live chat with customer service representative	Customer Service	Customer service representative and customers	high
BR-02	Sellers with more than 4 ratings can get special deals from Admin.	Sellers Marketplace	Admin	medium
BR-03	Purchase exceeding $200, will have free delivery option.	Payment processing.	Admin	medium
BR-04	About 20% discount for customers who purchase for the first time.	Payment Processing	Admin	Nice to have

User Requirements:

ID	Requirement	Topic Area	Actor	Priority
UR-01	User should be able to create an account to get started	Account Management	Customer and Seller	Critical
UR-02	User should be able to login	Account Management	Customer and seller	Critical
UR-03	Customers can choose the store	Order Management	Customer	critical
UR-04	Customers can choose the delivery or pick up option.	Order Management	Customer	High
UR-05	Customers can search for product.	Product search	Customer	high
UR-06	Customers can add or remove items from cart	Order management	Customer	high
UR-07	Customers can save products	Order management	Customer	low
UR-08	Sellers can view products and read review	Product management	Seller	Med
UR-09	Customers can make payment for their orders.	Payment Processing	Customers	Critical
UR-10	Customers can read, rate and provide product reviews	Review	Customers	high
UR-11	Users can view placed orders	Order Viewing	Sellers	Critical

UR-12	Sellers can remove products from product list	Product management	Sellers	Critical
UR-13	Admin can add or remove accounts	Account management	Admin	Critical
UR-14	Sellers can add products to the product list	Product Management	Sellers	high
UR-15	Customers should be able to view product list based on its category. .	Order viewing	Customer	high
UR-16	Sellers can modify products from the product list	Product management	Seller	high
UR-17	Users can update profile.	Account management	Seller and customer	high

Functional Requirements:

ID	Requirements	Topic Area	Actor	Priority
FR-01	When the customer confirms the order, system will notify the seller about the order.	Notification to seller.	system	critical

Non Functional Requirements:

ID	Requirements	Topic Area	Actor	Priority
NR-01	Each user should have unique username	Reliability	System	Critical
NR-02	Passwords must	Authentication.	System	Critical

	be minimum 8 characters long which must contain at least one uppercase letter, numeric character and special character.			
NR-03	All the pages should be loaded within 5 seconds when user gives an input in all load conditions.	Performance	system	high
NR-04	The system can be restarted in case of failure without loss of data as long as database is up and running.	Reliability	System	high

Data Storage:

In order to store data in backend we will be using MySQL database systems .We will use relational database schema to store the user account data as well as Stores data in separate tables.

UI Mockups:
1. Customer Dashboard: Home page.

2. Customer Dashboard: Cart

My Account
Login / Logout
Cart
Past Orders
Settings
help
Rate

STORE NAME

STORE ITEMS   ⊙∨

HOME	DEPARTMENT	CART

ITEM LISTS

PRODUCT   QUANTITY   REMOVE

SHIPPING ADDRESS
NAME
STREET ADDRESS
CITY [   ]   STATE [▼]   PINCODE [   ]

PAYMENT
CARD TYPE [▼]
CARD NUMBER
CVV [   ]   EXPIRATION [/]
NAME [   ]

CONFIRM ORDER

3. Customer Dashboard: Department

STORE NAME

MY ACCOUNT
LOGIN / LOGOUT
CART
PAST ORDERS
SETTINGS
help
RATE

SEARCH ITEMS ▼

HOME | DEPARTMENT | CART

CATEGORY LIST

BACK | CATEGORY NAME

PRODUCT LIST

4. Seller Dashboard: Managing Products.

SELLER DASHBOARD

| STORE NAME | MY ACCOUNT |

HOME   VIEW   MANAGE
              ORDERS   PRODUCTS

ADD A NEW
ITEM

VIEW ITEMS

REMOVE ITEMS

MODIFY

QUICK SHOP   HOME   ORDERS   PRODUCTS   MYACCOUNT

ADD A NEW ITEM

CATEGORY   DAIRY ▼

ITEM NAME

PRICE

ITEM DESCRIPTION

SUBMIT

5. Create account and Login.

QUICK SHOP.

CREATE ACCOUNT   OR   LOGIN.

FIRST NAME [   ]   EMAIL [   ]

LAST NAME [   ]   PASSWORD [   ]

EMAIL [   ]   [LOGIN]

PASSWORD [   ]

CONFIRM [   ]

PHONE [   ]

ADDRESS
STREET NUMBER [   ]
STREET NAME [   ]
   Apt   NUMBER [   ]
         CITY [   ]   STATE [   ]
   ZIP CODE [   ]

TYPE   O SELLER   O CUSTOMER

[REGISTER.]

Class Diagram:

User

-UserProfile: Profile
-emailID : string
-password : string

+login (emailid : string, password : string)
+logout ()
+getEmailid() : string
+getPassword() : string
+getUserProfile() : Profile
+setEmailid(EmailID : string)
+setPassword(Password : string)
+getOrder() : Order

Profile

-firstName : string
- lastName
-addr : Address
-phoneNumber : string
- type : string

+getFirstName() : string
+getLastName() : string
+getAddress() : Address
+getPhoneNumber() : string
+setFirstName(FirstName : string)
+setLastName(LastName : string)
+setAddress (addr : Address)
+setPhoneNumber(PhoneNumber : string)
+ getType() : string
+ setType(Type : string)

Address

- streetNumber : string
- streetName : string
- aptNumber : string
- city : string
- state : string
- zip: string

+getStreetNumber():string
+getStreetName():string
+getAptNumber():string
+getCity():string
+getState():string
+getZip():string
+setStreetNumber(streetNumber:string)
+setStreetName(streetName:string)
+setAptNumber(aptNumber:string)
+setCity(city:string)
+setState(state:string)
+setZip(zip:string)

Customer

- customerID : int

+ getCustomerID() : int
+setCustomerID(CustomerID : int)
+searchProduct()
+addProduct()
+removeProduct
+checkout() : Order
+writeReview(Product,Review) : void
+getCart() : Cart
+getPayment: Payment
+ saveProduct(Product)
+searchStore() : Seller

Admin

- AdminID : int

+addAccount()
+removeAccount()

Seller

-sellerID : int

+removeProduct(Product)
+getReview()
+getSellerProduct()
+setSellerProduct()
+ updateProduct()
+viewProducts: Product
+searchcategory()
+viewCategory() : Category

Category

- categoryName : string

+getCategoryName()
+setCategoryName()
+searchProduct() : Product

Cart

- cartID : int
- product : Product
- numberofProducts: int
- quantity : int

+ addProduct()
+ removeProduct(Product)
+ updateQuantity ()
# viewProducts ()
+ addPaymentDetails : Payment

Order

- orderNumber : int
- orderDate : date
- orderDetails : string

+ PaymentTotal() : Payment
+getShippingMethod()
+setShippingMethod()
+getAddress : Address
viewOrder (orderNumber : int) :void
+placeOrder (cart : Cart) : int

Review

- customerID : int
- productID : int
- rate : int
-review : string
- reviewID : int

+ postReview(customerID : int, productID : int)
+getReviewID ()
+setReviewID()
+postRating(customerID : int, productID : int)

Product

- productID : int
- productName: string
-productDescription : string
- productPrice : int
- productCategory : string

+getProductID()
+setProductID()
+getProductName()
+setproductNAme()
+getproductDescription()
+setproductDescription()
+getproductPrice()
+setproductPrice()
+ isProduct () : bool
+removeProduct()

Saved Products

- SavedProductID : int
- product : Product
- numberofProducts : int
- customerID

+ getProduct()
+ setProduct()
#viewProducts()

Payment

- paymentID : int
- PaymentDate : Date
- paymentTotal : int

+ getPaymentID()
+ setPaymentID()
+ getPaymentDate()
+setPaymentDate()
+getPaymentTotal()
+setPaymentTotal()
+getPaymentDetails : CreditCard
+makePayment()

CreditCard

- cardNumber : int
- cardName : string
- cardValidity : Date
- cardCVV: int

+ getCardNumber()
+ setCardNumber()
+setCardName()
+getCardName()
+setCardValidity()
+getCardValidity()
+getCardCVV()
+setCardCVV()

Use Case Documents:

Use case ID	UR - 03
Use Case Name	Customer can search for stores
Description	Customer can search for the store according to their convenience
Actors	Customer
Precondition	Valid customer account
Post Condition	Display search results
Frequency of Use	Frequently used by users
Flow of Events	

Step	Actor Action	System Response
1	Click on Login button	Display Home page.
2	Select preferred store from store list	Display the store option
3	Click on the store you preferred.	Display the home page with store name.

Variations	Use drop down menu
Exceptions	No results based on added preferences
Developer Notes	

Use case ID	UR - 05
Use Case Name	Customer can search for product
Description	Customer can search for product based on directly searching for the product or categorizing based on departments.
Actors	Customer
Precondition	Customer should have selected the store.
Post Condition	Display product lists.
Frequency of Use	Frequently used by users
Flow of Events	

Step	Actor Action	System Response
1	Click on the departments tab	Display various departments and the products within those departments.
2	Click on the product you wish to buy	Display product details i.e product description and price

Variations	Customer can search from the search box below the store name.
Exceptions	Required product not found.
Developer Notes	

Use case ID	UR - 09
Use Case Name	Customers makes payment
Description	Customers can make payment for their orders.
Actors	Customers
Precondition	Add products to cart.
Post Condition	Order has been approved and payment is received.
Frequency of Use	Often.
Flow of Events	

Step	Actor Action	System Response
1	Click Cart.	Displays product list in the cart.
2	Add card details	Displays card details to confirm.
3	Click on confirm option.	Payment is received and order is processed.

Variations	
Exceptions	Invalid card details.
Developer Notes	If payment incomplete, notify customer to add different details.

Use case ID	UR - 10
Use Case Name	Provide reviews and rate products
Description	Customers can rate the products and provide reviews for their past orders.
Actors	Customer
Precondition	Customer should search the product for which he wants to rate and provide the review
Post Condition	The system accepts the rating and review of the product
Frequency of Use	Often
Flow of Events	

Step	Actor Action	System Response
1	Customer will search the product either by name or by category	System displays the product
2	Customer clicks on the Rate/Review	System displays the Rate/Review interface
3	Customer clicks on the "Rate/Review" button in front of the Product and submits	System accepts the Rating/Review from the Customer and displays a success message

Variations	
Exceptions	System reloads the same page if the customer clicks "submit" button without providing any rating/review.
Developer Notes	

Use case ID	UR - 13
Use Case Name	Add or remove accounts
Description	Admin has authority to add or remove customer and seller accounts
Actors	Admin
Precondition	Login into the system. Admin is authenticated.
Post Condition	Accounts added or removed.
Frequency of Use	Occasionally
Flow of Events	

Step	Actor Action	System Response
1	Click on add user	Form to add username, email ID and details.
2	Entered email address and username	
3	Click Done.	Return to home page.
4.	Click on remove user.	Form to enter name and email address to remove.
5.	Click Done.	Return to home page.

|	
Variations	While adding account, email ID already exists. While removing account, email ID not found.
Exceptions	
Developer Notes	

Use case ID	UR - 12
Use Case Name	Sellers remove products.
Description	Sellers can remove products in the product list of their respective stores.
Actors	Seller
Precondition	Login into the system.
Post Condition	Product should be removed from the system
Frequency of Use	Occasionally
Flow of Events	

Step	Actor Action	System Response
1	Seller login into the system	Display home page.
2	Click on products tab.	Display options to add, remove, modify product.
3	Select remove product	Displays category list.
4.	Search the category	Displays the products in category
5.	Search product	Display product information
6.	Select Remove	Remove the product

Variations	
Exceptions	Search Product fails if product to be removed is not in the database
Developer Notes	

Use Case Overview:

Admin
- UR-13 — Add user
- UR-02 — Login
- UR-11
- UR-13 — Remove user

Seller
- UR-11 — View Order
- UR-17 — Update Profile
- UR-02
- UR-01 — Create account
- UR-14 — Add New Product
- UR-12
- UR-08
- UR-08 — View products in store
- UR-16 — Modify Product
- Remove product from store

Customer
- UR-02 — Login
- UR-05 — Search Products
- UR-07 — Save products
- UR-11 — View Order
- UR-17 — Update Profile
- UR-01 — Create account
- UR-06 — Add Product to cart
- UR-06 — Remove Product from cart
- UR-10
- UR-09 — Pay for item
- UR-10 — Write Review
- UR-03
- UR-04
- UR-15

Add Product to cart

Remove Product from cart

Write Review

Pay for item

Read Reviews

Search Store

Choose delivery or pickup

View products based on category

Use case Sub diagrams:

1. UR -05 Customer can search product
2. UR -14 Seller can add product to product list.
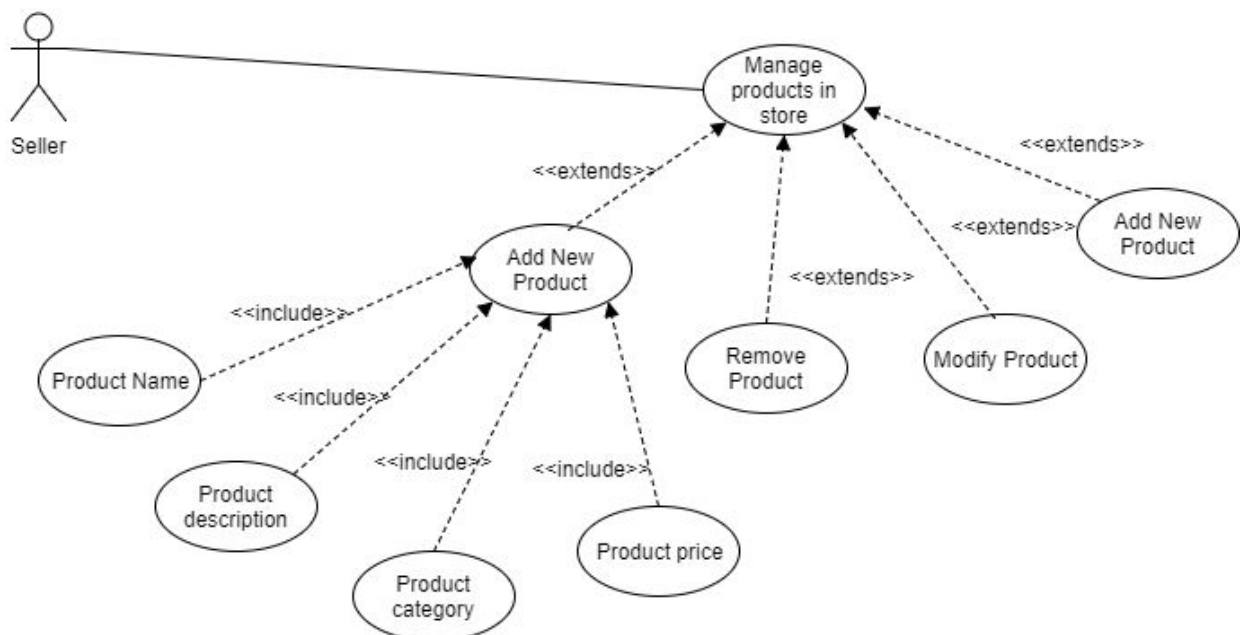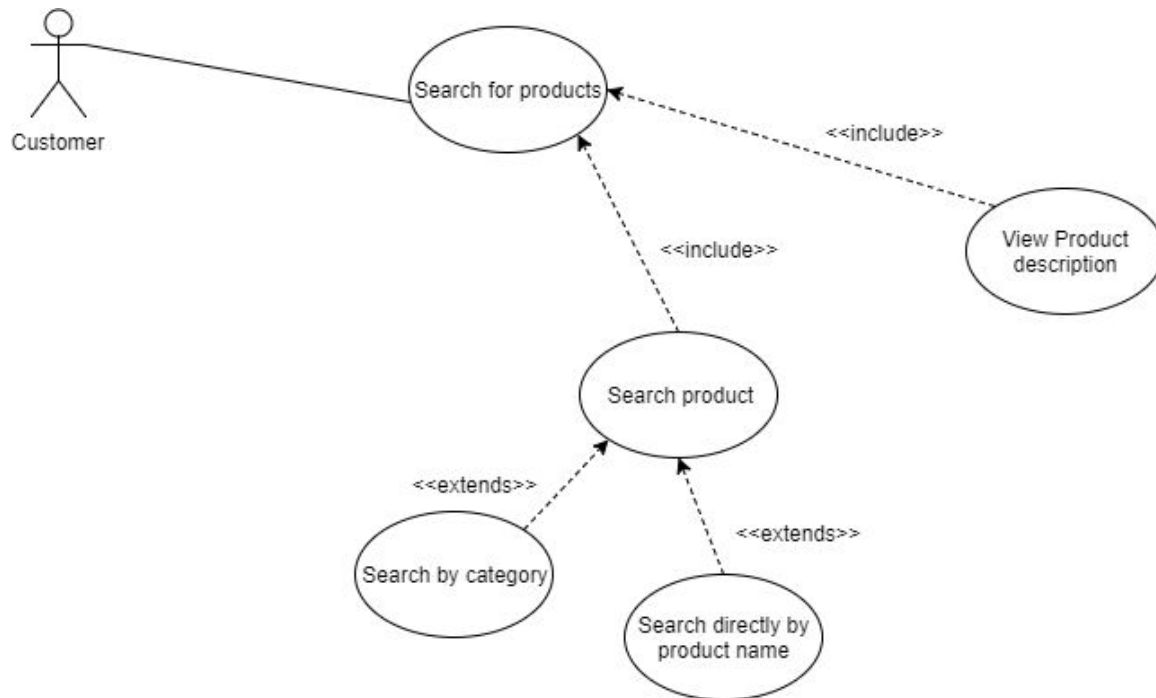
Activity Diagrams:
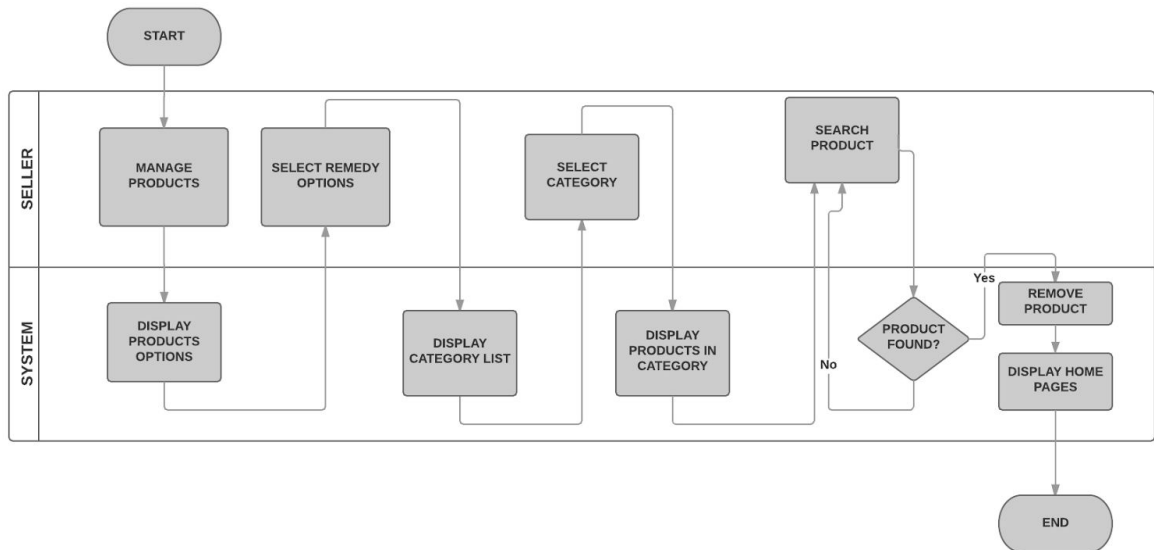
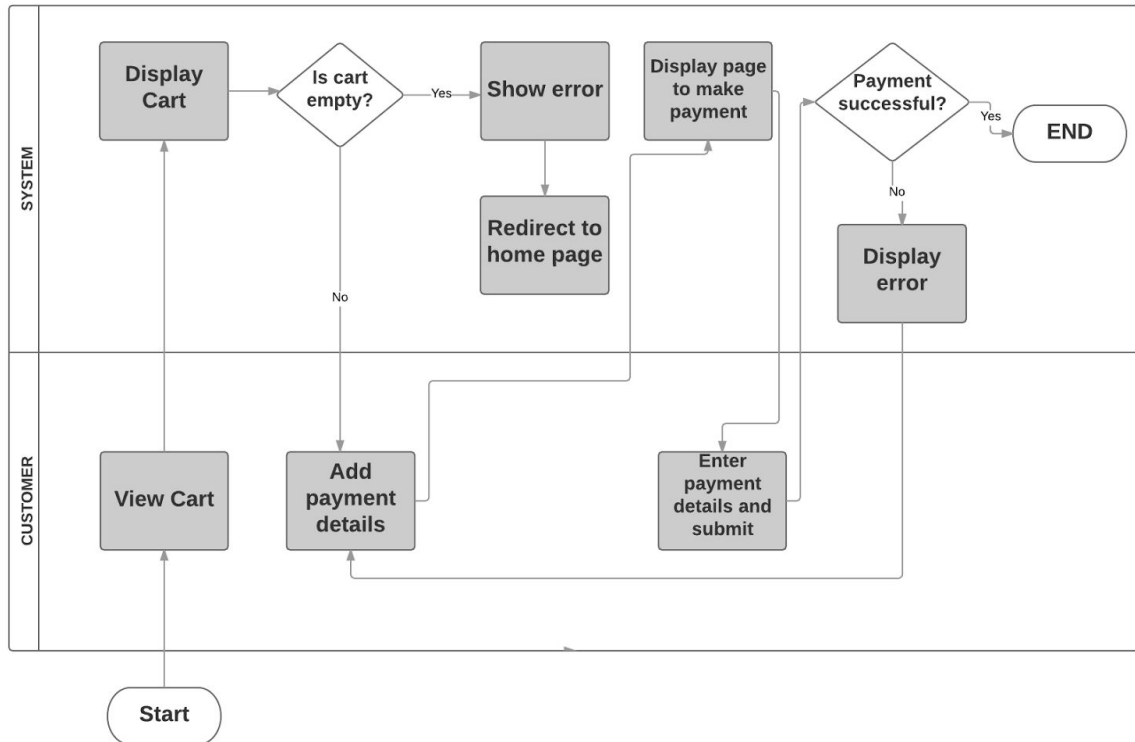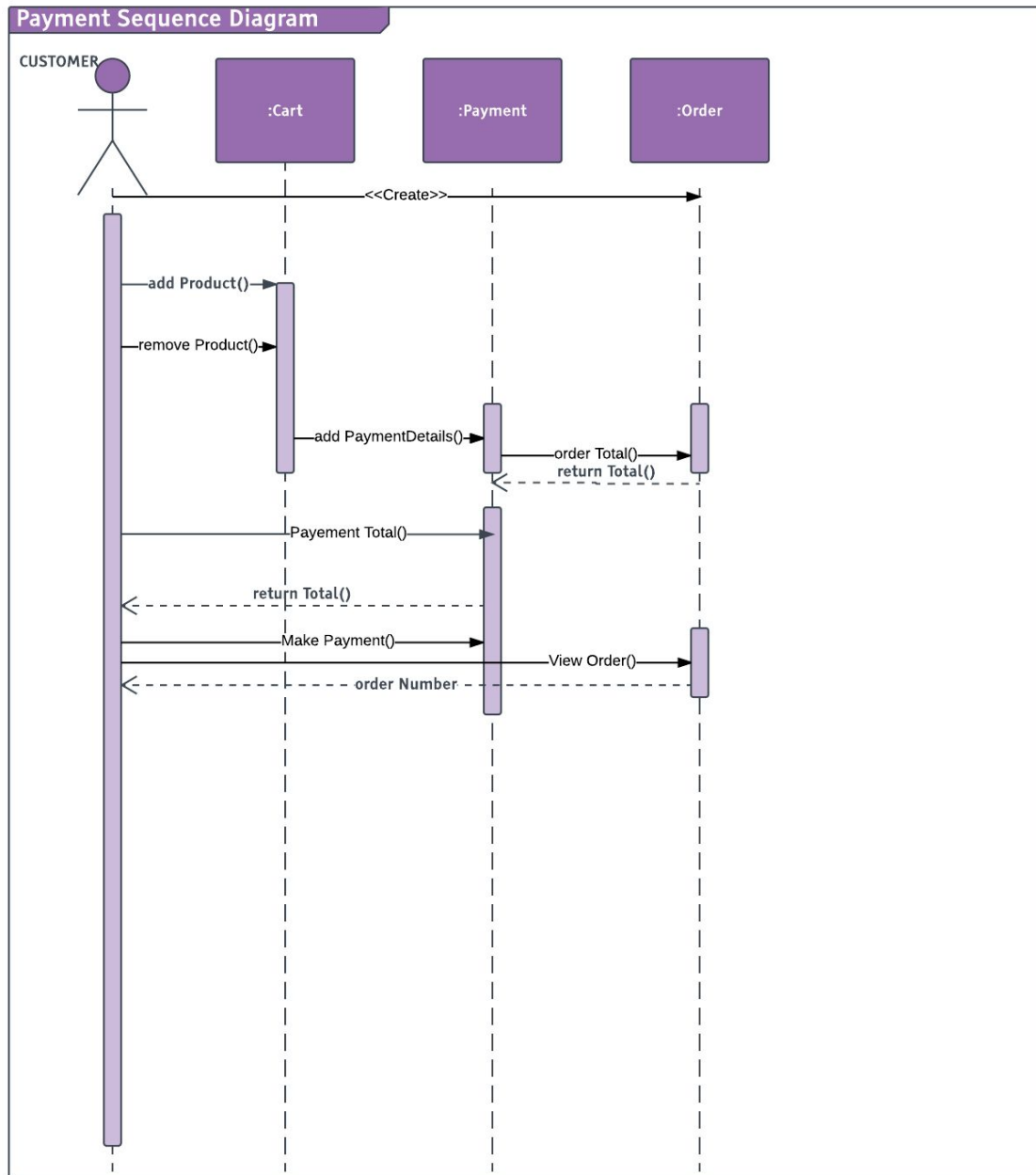## Seller can remove product from store (UR-12)

Customer makes payment for order (UR - 09)

CUSTOMER MAKES PAYMENT FOR RESPECTVE ORDER

SEQUENCE DIAGRAMS:

Customer makes payment for order (UR - 09)

Seller remove item part a(UR-12)

Seller remove product

customer

:SELLER CATEGORY PRODUCT

customer →removeProduct→ :SELLER

:SELLER →viewCategory()→ CATEGORY

:SELLER →searchProduct→ CATEGORY →isProduct?→ PRODUCT

CATEGORY ←-- failed -- PRODUCT

:SELLER ←-- failed -- CATEGORY

:SELLER →searchProduct→ CATEGORY