Online Food Ordering System

Online Food Ordering System

Team Members:

1. Muhammed Ali Muhammed
2. Muhammed Fathi Salah
3. Mustafa Mahmoud Mustafa
4. Mahmoud Muhammed Abdelwehab
5. Muhammed Hani Abo Elela

Supervised By:

Dr. Abdel Wahab Kamel Asammak

Business Objectives

1.1 System Scope:

The goal of the system is providing online food ordering service to save time and effort of users with easy user interface that encourage them to purchase.

System Requirements (Cont.)

1.2 Business Objectives:

Our project covers a wide scope designed to implement computerized system for the following Requirements:

1. **<u>Functional Requirements:</u>**
   - **Order entry subsystem**
     - Create new order
     - Add item to cart
     - Update cart
     - Look up for meal

System Requirements (Cont.)

- **Order state subsystem**
  - Check order state
  - Update order state
  - Cancel order
  - Update order
- **Administration subsystem**
  - Add meal
  - Update meal description
  - Delete meal
  - Add new offer
  - Create monthly summary reports for sales

System Requirements (Cont.)

- **Customer maintenance subsystem**
  - Create account
  - Add meal feedback

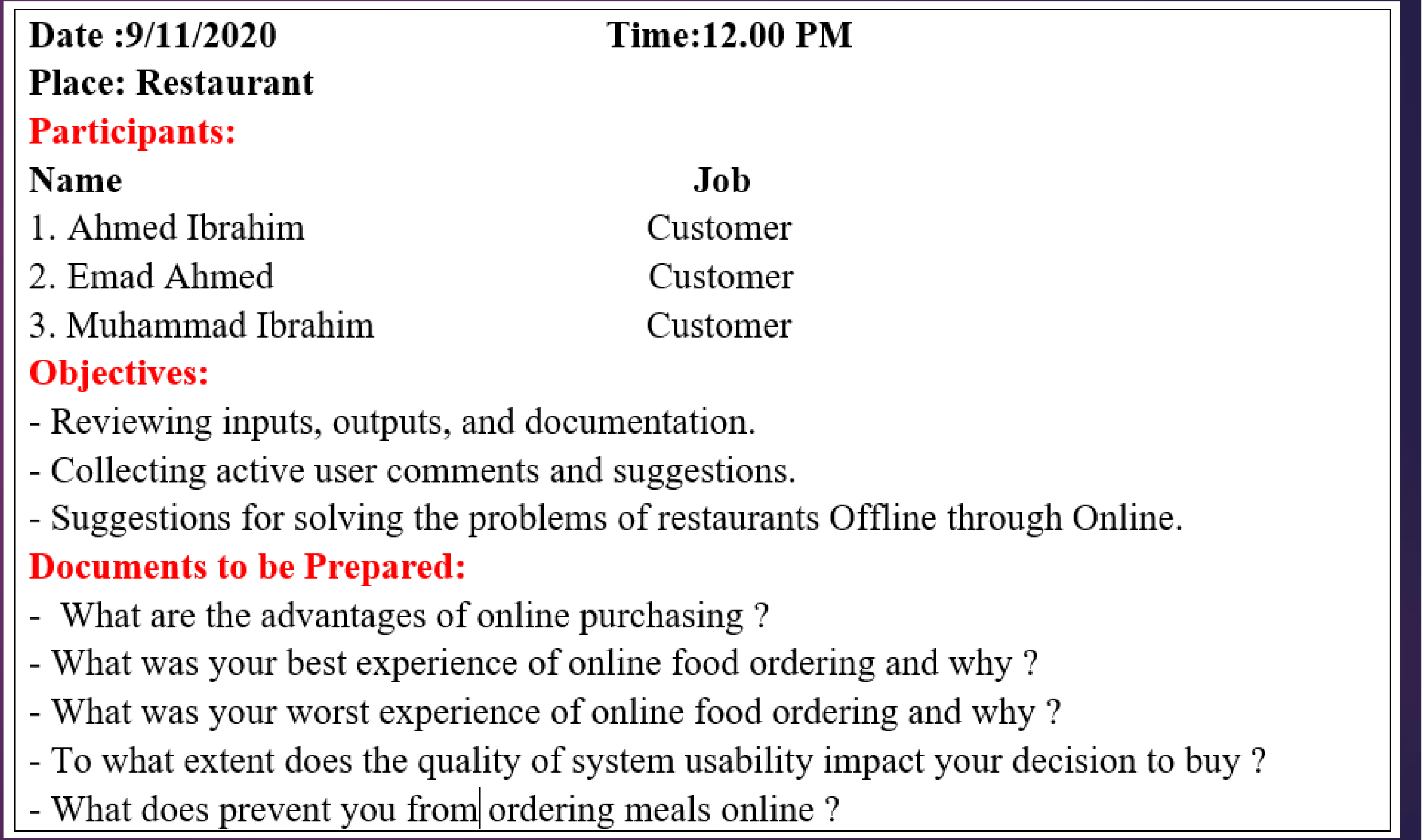
System Requirements (Cont.)

2. **Nonfunctional Requirements:**
   - Easy user interface
   - Little failure rate
   - Several recovery method and data backups
   - High response time
   - Secured access control
   - High performance servers

System Requirements (Cont.)

1.3 Fact Finding: Interviews

Interview Agenda



Date :9/11/2020 **Time:12.00 PM**

Place: Restaurant

Participants:

Name	Job
1. Ahmed Ibrahim	Customer
2. Emad Ahmed	Customer
3. Muhammad Ibrahim	Customer

Objectives:

- Reviewing inputs, outputs, and documentation.
- Collecting active user comments and suggestions.
- Suggestions for solving the problems of restaurants Offline through Online.

Documents to be Prepared:

- What are the advantages of online purchasing ?
- What was your best experience of online food ordering and why ?
- What was your worst experience of online food ordering and why ?
- To what extent does the quality of system usability impact your decision to buy ?
- What does prevent you from ordering meals online ?

System Requirements (Cont.)

Interview Summary:

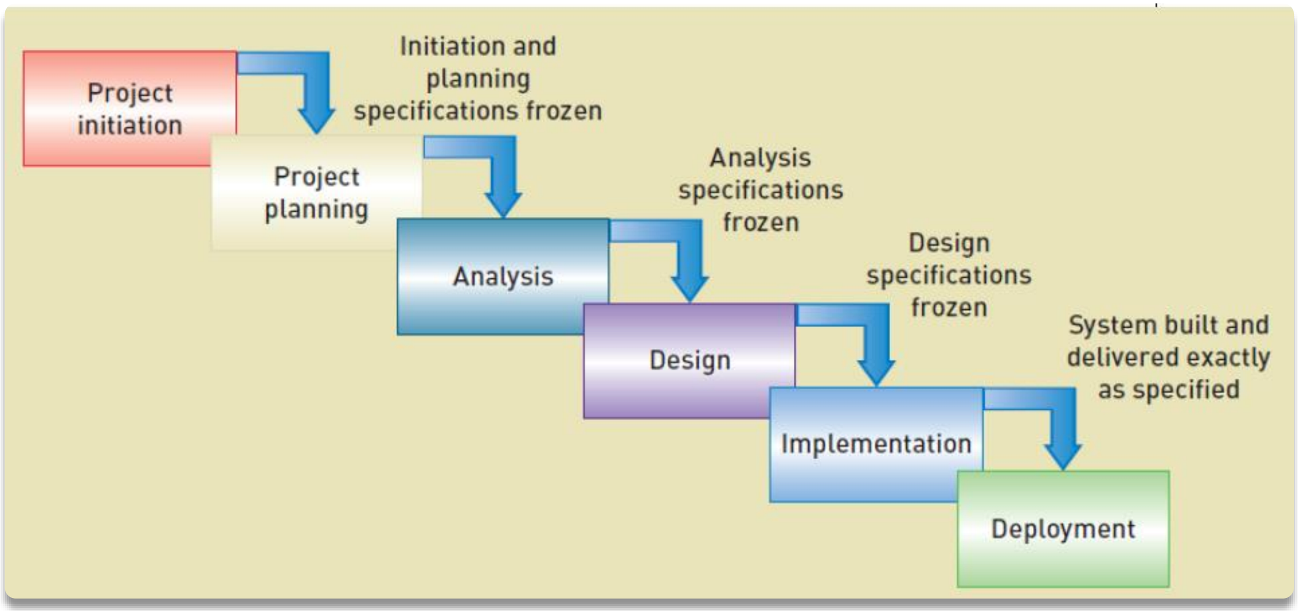
According to customer opinions we reached to that :

- Online ordering saves their time and effort.
- They can compare various products and prices.
- Examples for good experiences of online food ordering Systems.
- The effect of easy user-interface systems on purchasing decision.
- Examples for bad experiences of online food ordering systems.

System Requirements (Cont.)

1.4 Development Process:

- We decided to apply the **Iterative development technique**.
- The reason for using this technique is that it Assumes the project must be more flexible and adapt to changing needs as the project progresses and Requirements and needs are uncertain and/or high technical risk


Project initiation
Initiation and planning specifications frozen
Project planning
Analysis specifications frozen
Analysis
Design specifications frozen
Design
System built and delivered exactly as specified
Implementation
Deployment

System Analysis

2.1 Used Methodology:

Object-Oriented approach

2.2 Analysis Models: Event Table

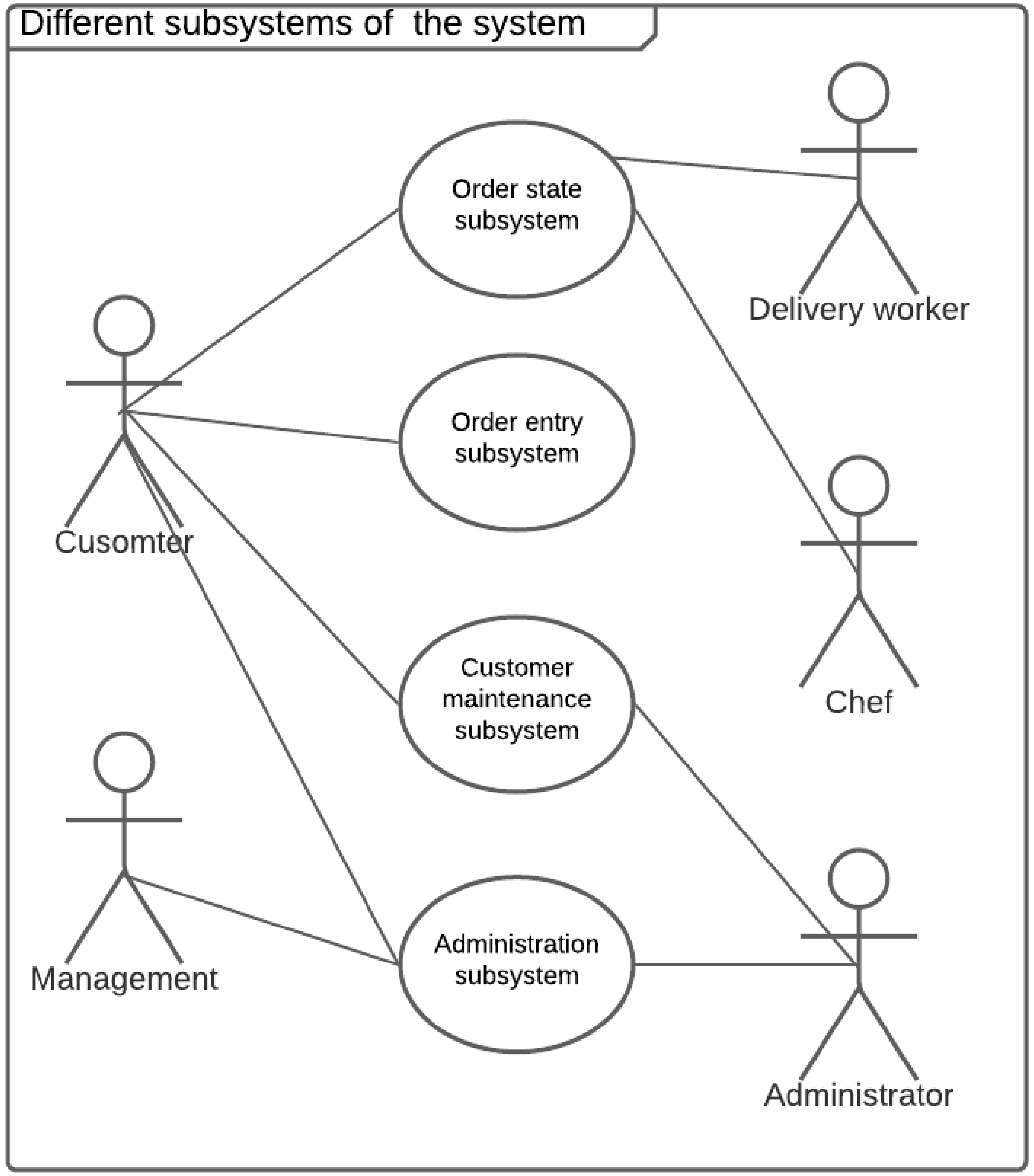
Event	Trigger	Source	Use case	Response	Destination
Customer creates account	Customer details	Customer	Create new account	Account ID	Customer
Customer creates new order	Customer account credentials	Customer	Create new order	Order summary	Customer
Customer checks order state	Customer information Order summary	Customer	Check order state	Order state	Customer
Customer updates order	- Order ID - Order state	Customer	Update order	New order summary	Customer
Customer updates order state	Order ID	- Customer - Chief - Delivery worker	Update order state	Confirmation for updating	Customer
Customer cancels order	- Order state - Order ID	Customer	Cancel order	Canceling confirmation	- Customer - Delivery worker - Chief
Customer searches for a meal	Meal name	Customer	Look up for a meal	Meal information	Customer

2.2 Analysis Models: Event Table

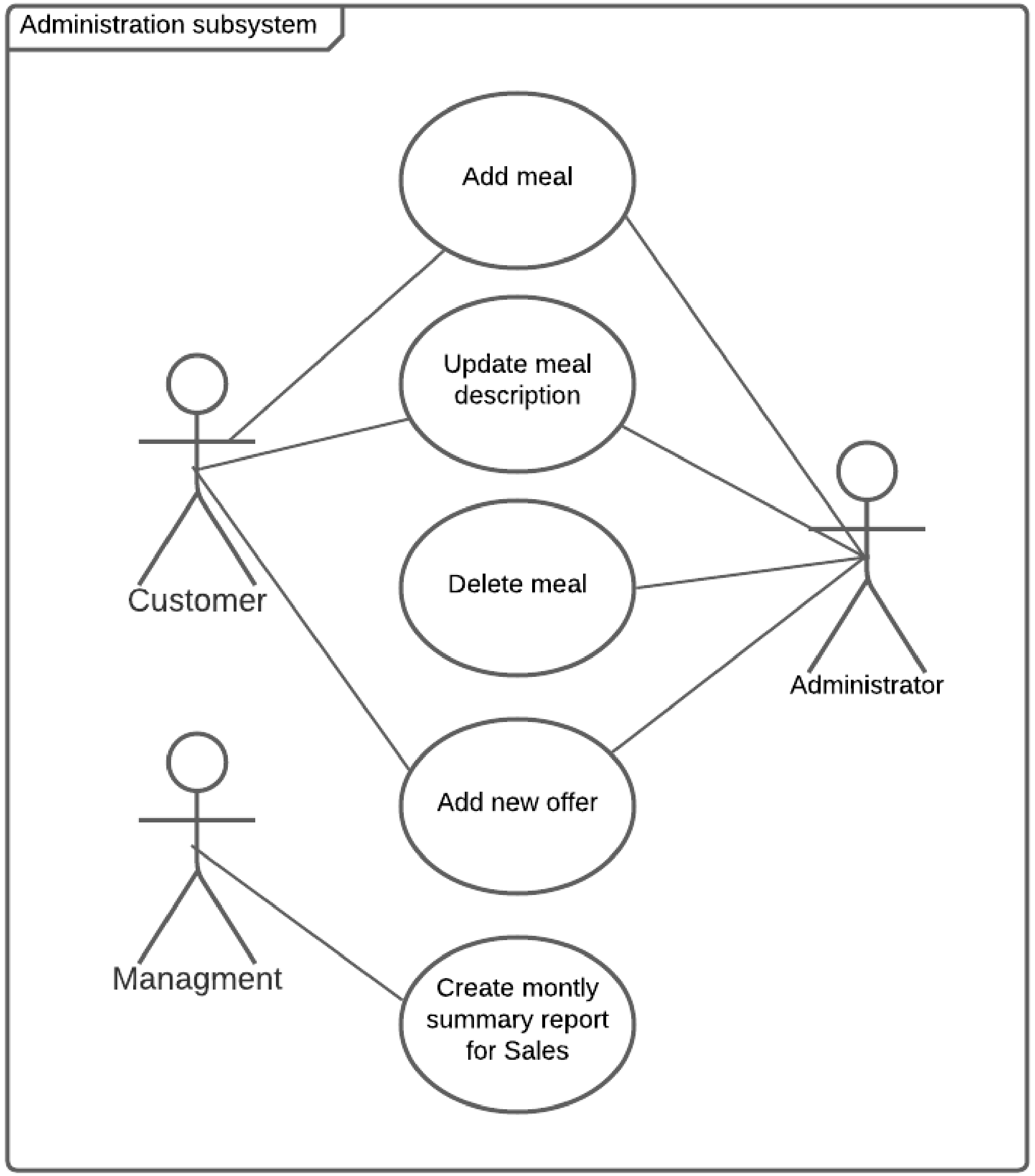
Event	Trigger	Source	Use case	Response	Destination
Customer gives feedback for a meal	Customer account credentials	Customer	Add meal feedback	Feedback report	Customer - -Administrator
Administrator adds a new meal	- Meal name - Meal picture - Meal description	Administrator	Add new meal	Meal ID	- Administrator - Customer
Administrator updates meal	Meal ID	Administrator	Update meal description	Meal details	- Administrator - Customer
Administrator deletes a meal from the system	Meal ID	Administrator	Delete meal	Delete Confirmation	Administrator
Administrator adds new offer for a meal	Offer details	Administrator	Add new offer	Start and end time of offer	Customer
Customer adds item to cart	Item name	Customer	Add item to cart	Adding confirmation	Customer
Customer updates cart	Cart content details	Customer	Update cart	Cart content summary	Customer
Time to create monthly summary reports for sales	End of month		Create monthly Summary reports	Sales summary reports	Management

Use Case Diagrams

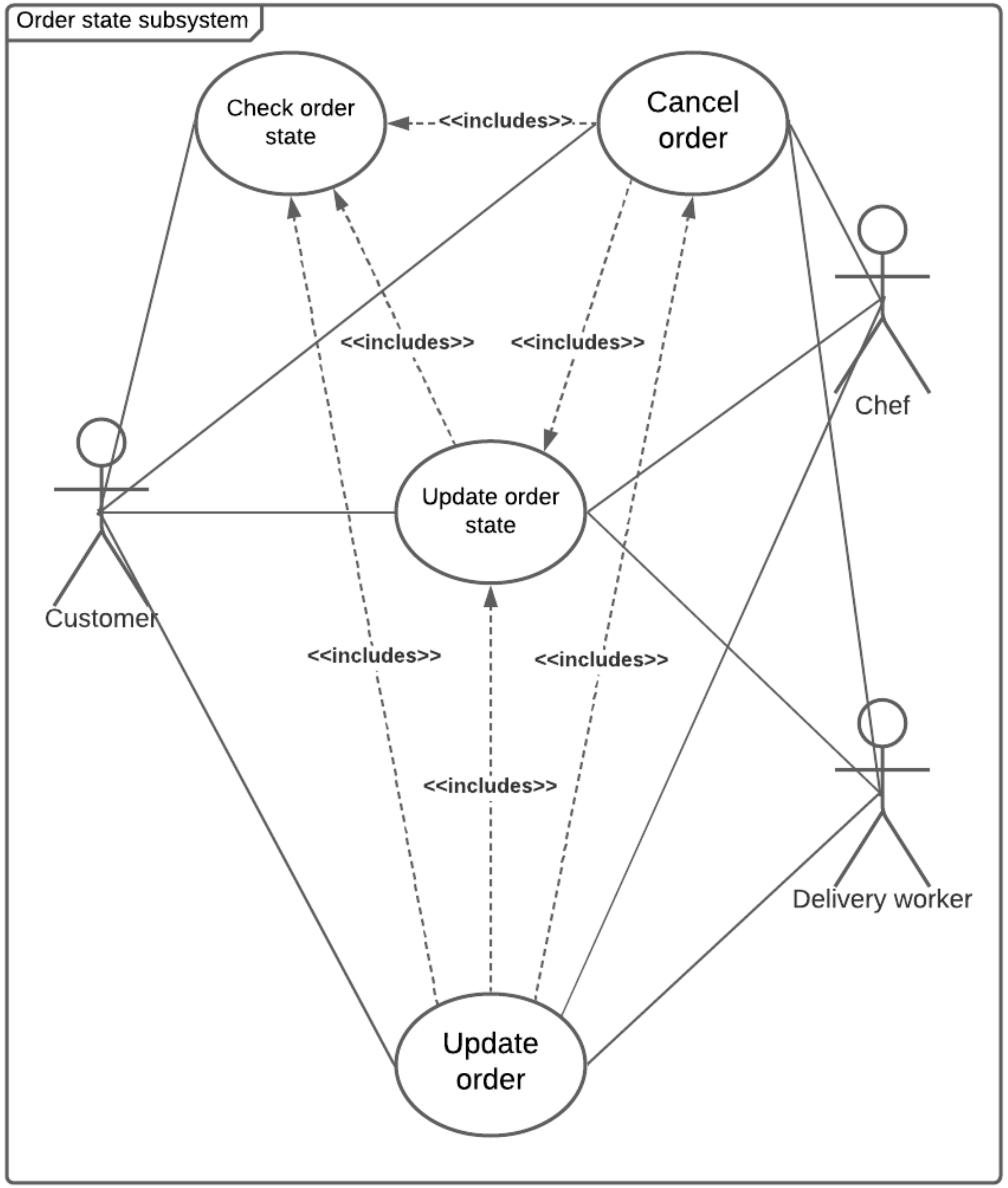
Different subsystems of the system



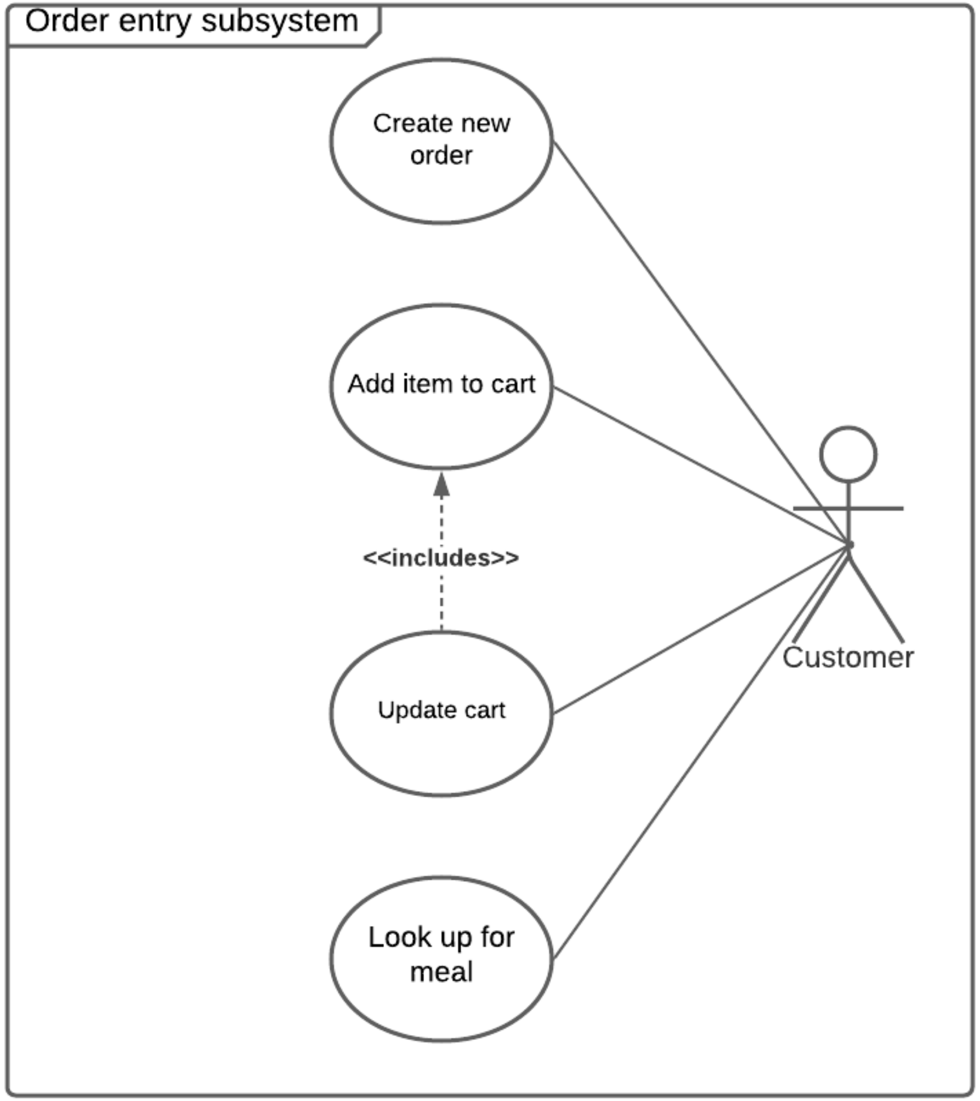
Administration subsystem use cases



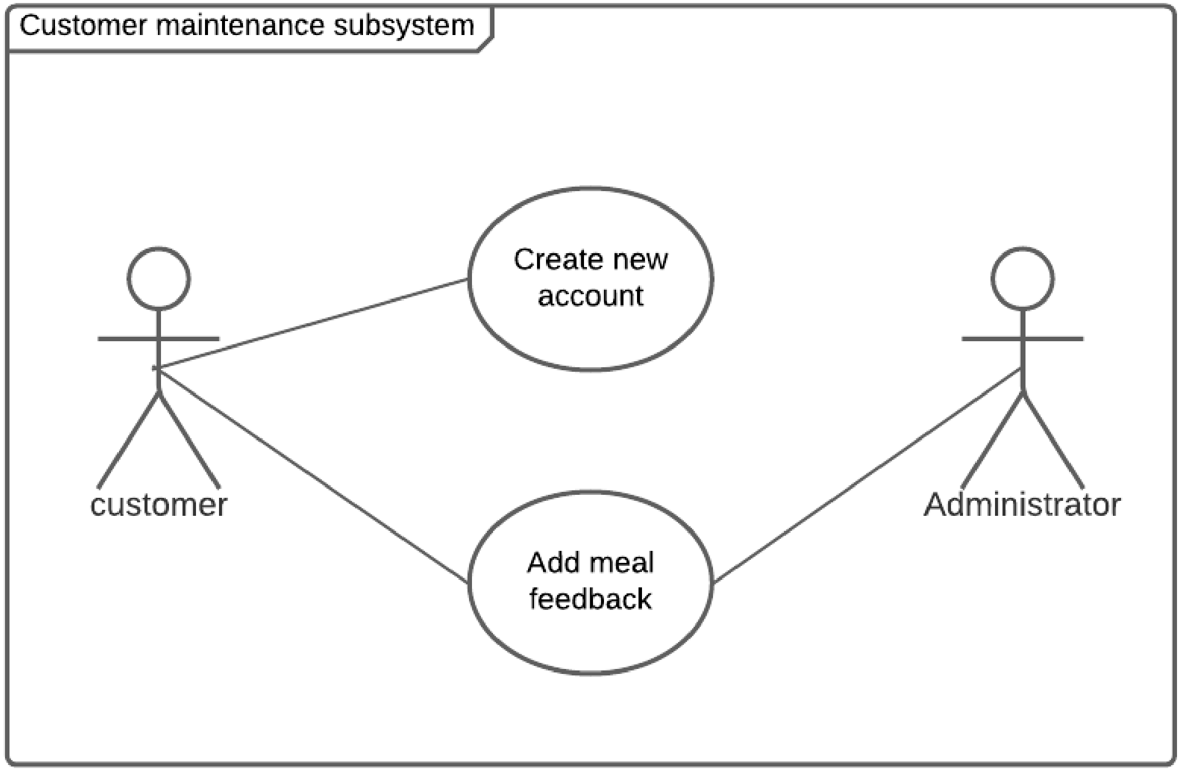
Order state subsystem use cases



Order entry subsystem use cases

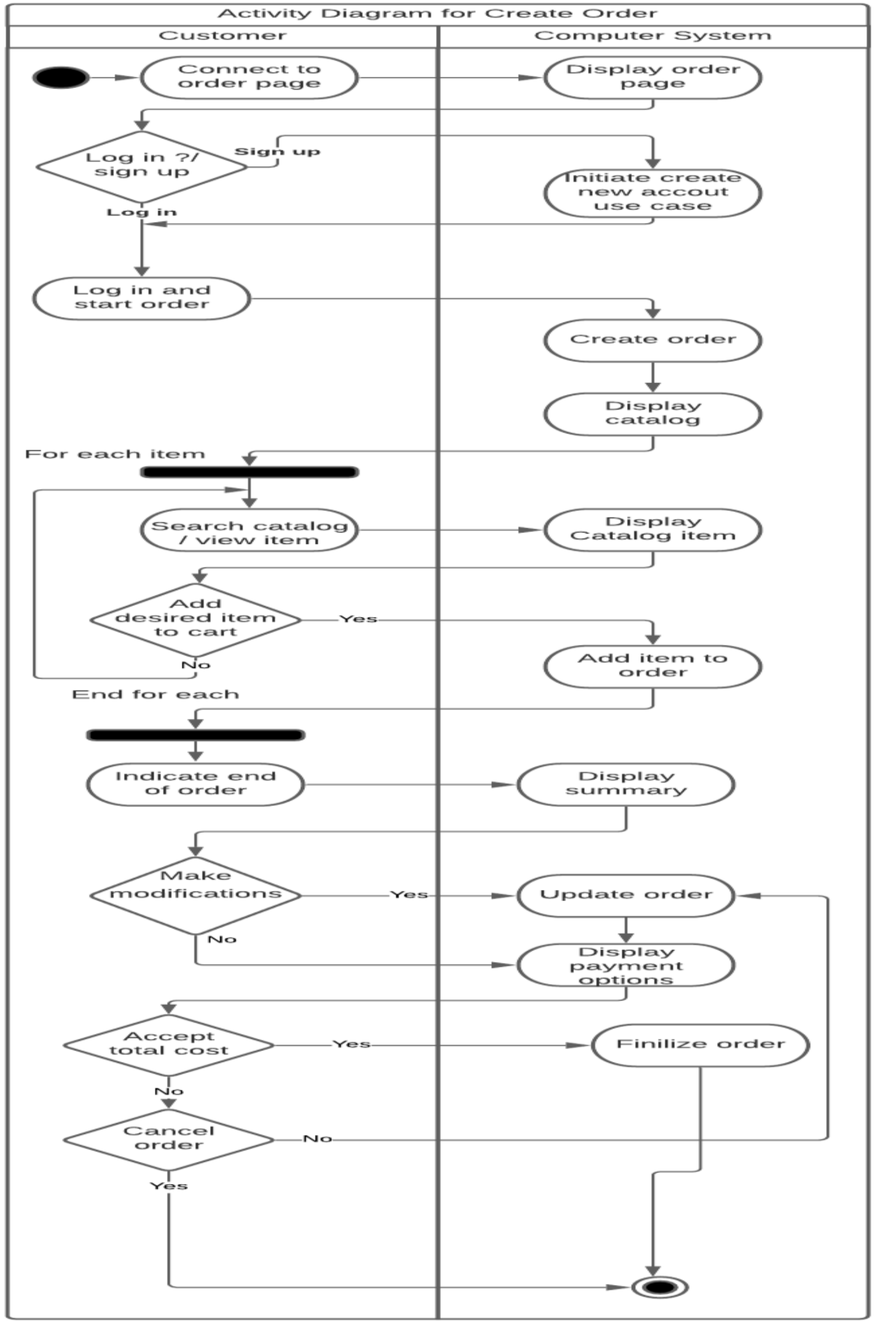


Customer maintenance subsystem use cases



Use Cases Description

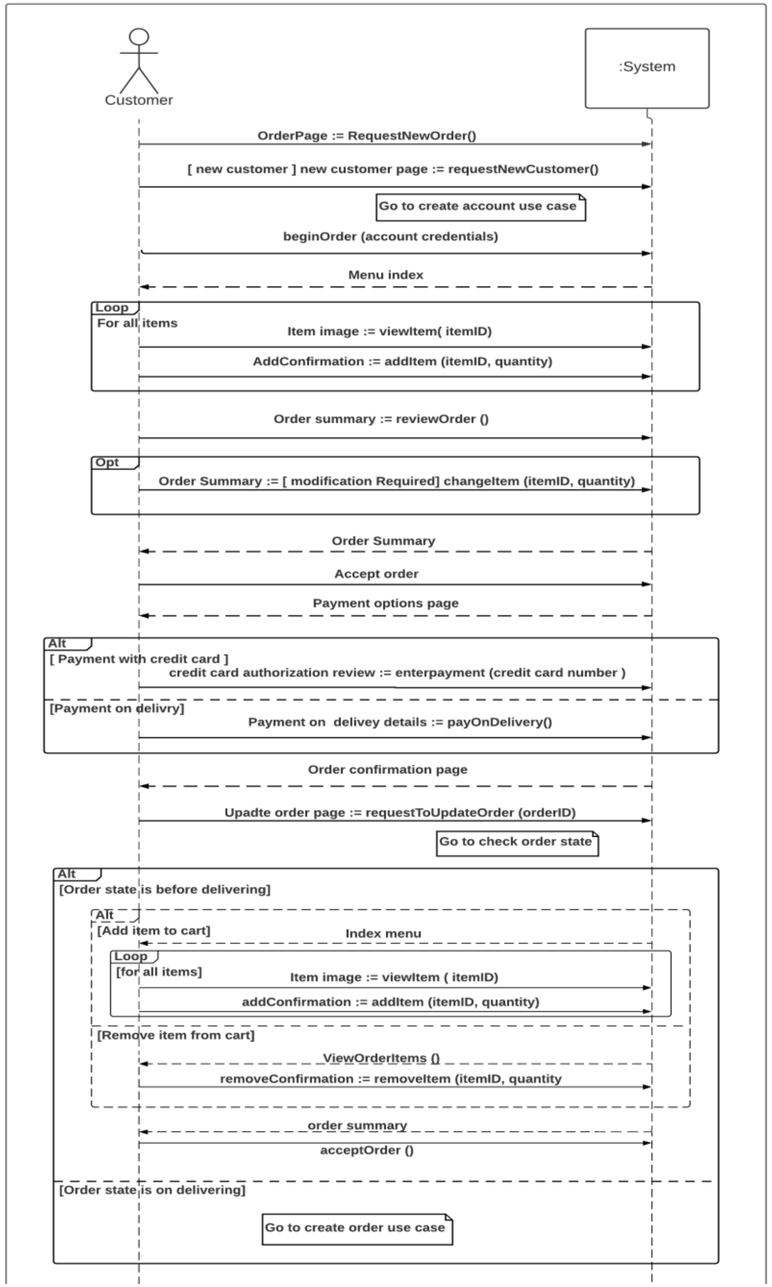
Activity diagram for Create order use case



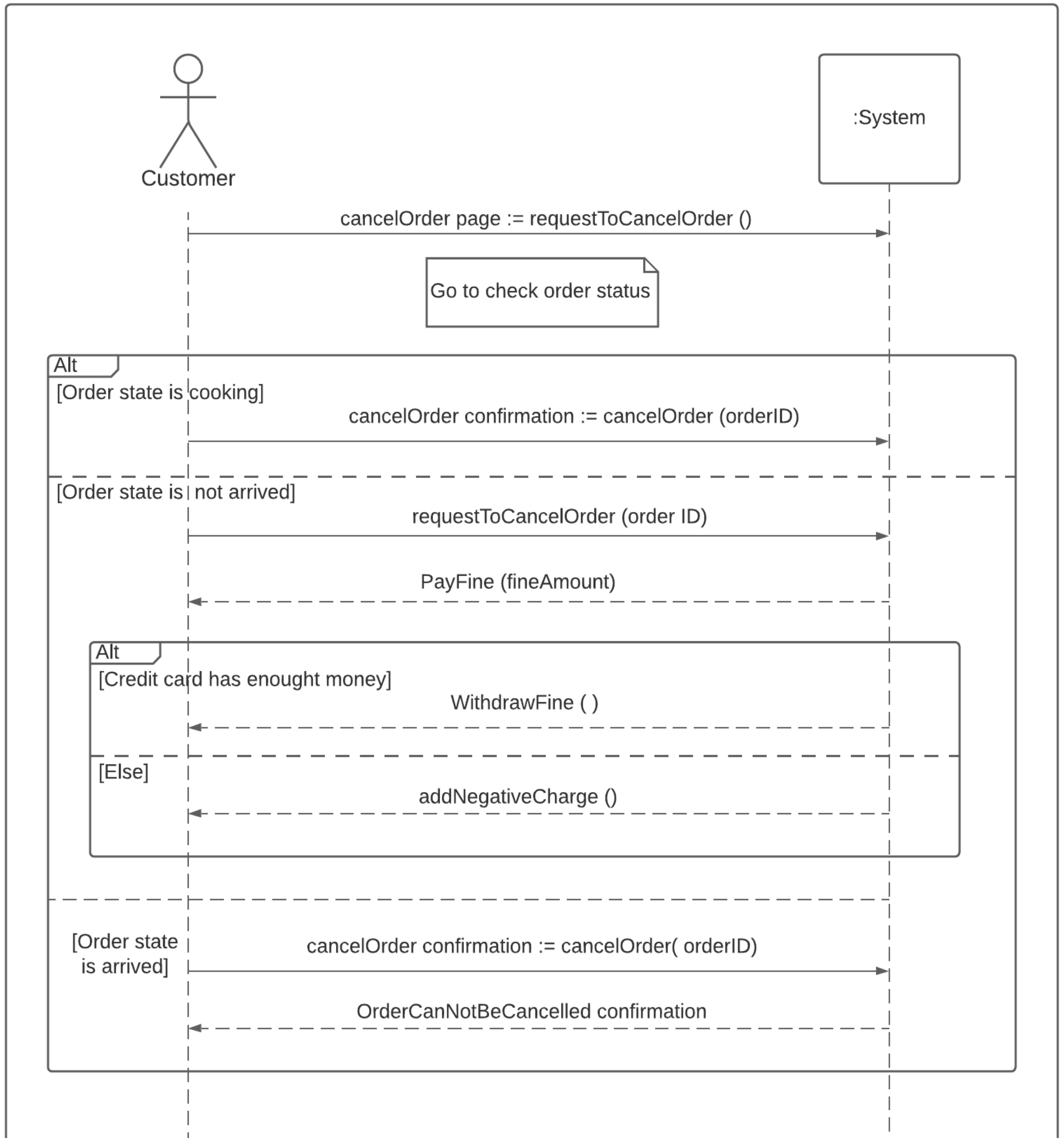
Scenario for Create account use case

Use Case Name:	**Create account**	
Actor:	Customer	
Preconditions:	----------------------	
Postconditions:	System issues email to customer with new account ID and data	
Flow of activities:	Actor	System
	1. Customer requests to create account. 2. Customer enters his/her name, phone number and email. 3. Customer enters his/her address 4. Customer chooses payment method	1. System creates a new account 1.2 System prompts for customer data 2.1 System verifies customer data 2.2 System prompts for customer address 3.1 System verifies customer address 3.2 System prompts for credit/debit card information or payment on delivery . 4.1 System verifies credit/debit card authorization. 4.2 System returns valid customer account details.
Exceptions	2.1 If customer data is incorrect then the system prompts customer for entering correct data. 3.1 If customer address is incorrect then the system prompts customer to enter the correct address. 4.1 if credit/debit card information is incorrect then system a. prompts for correct credit/debit card information, or b. suggests payment on delivery.	

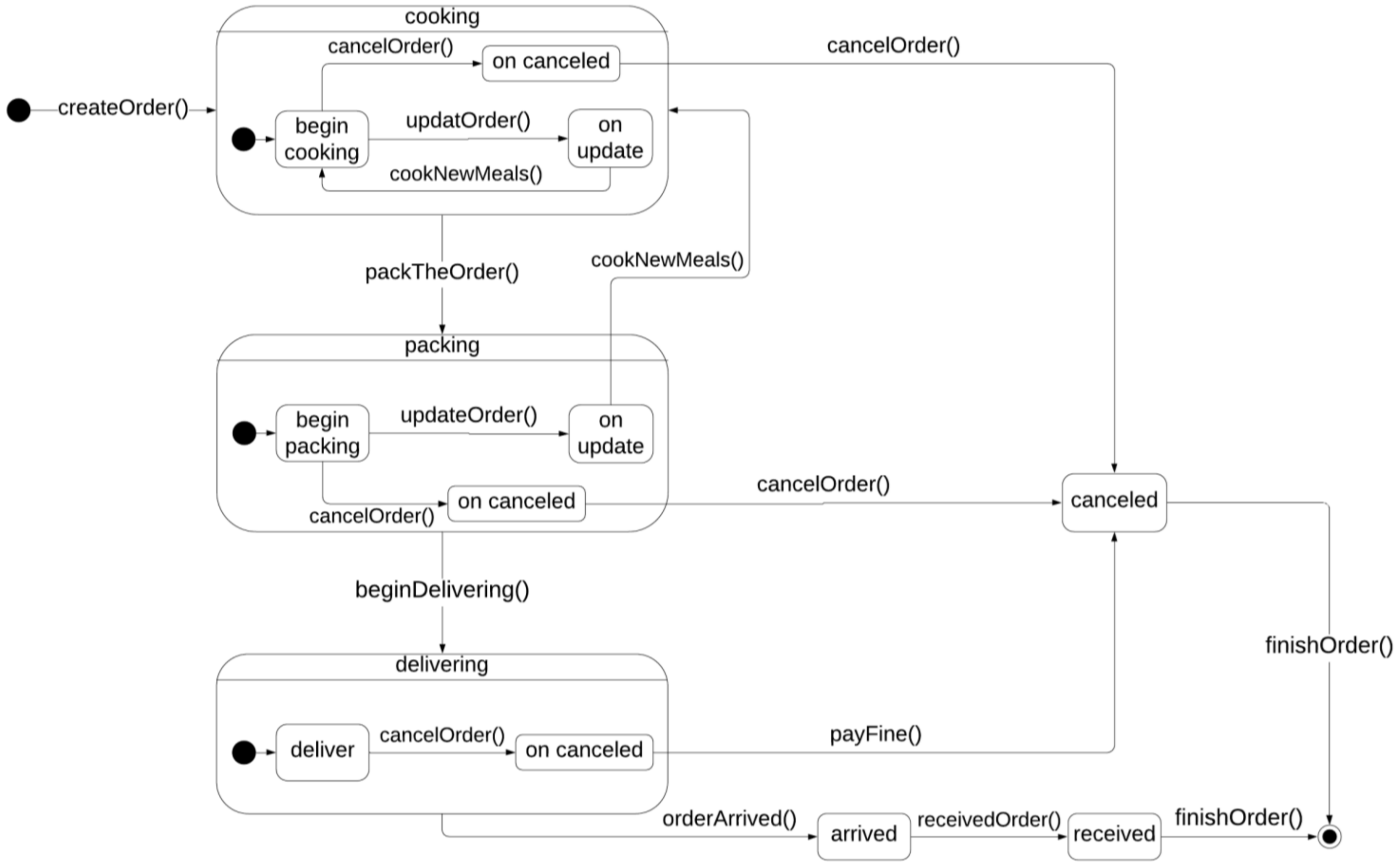
Sequence diagram for Order entry subsystem


Customer
:System
OrderPage := RequestNewOrder()
[new customer] new customer page := requestNewCustomer()
Go to create account use case
beginOrder (account credentials)
Menu index
Loop
For all items
Item image := viewItem(itemID)
AddConfirmation := addItem (itemID, quantity)
Order summary := reviewOrder ()
Opt
Order Summary := [modification Required] changeItem (itemID, quantity)
Order Summary
Accept order
Payment options page
Alt
[Payment with credit card]
credit card authorization review := enterpayment (credit card number)
[Payment on delivry]
Payment on delivey details := payOnDelivery()
Order confirmation page
Upadte order page := requestToUpdateOrder (orderID)
Go to check order state
Alt
[Order state is before delivering]
Alt
[Add item to cart]
Index menu
Loop
[for all items]
Item image := viewItem (itemID)
addConfirmation := addItem (itemID, quantity)
[Remove item from cart]
ViewOrderItems ()
removeConfirmation := removeItem (itemID, quantity
order summary
acceptOrder ()
[Order state is on delivering]
Go to create order use case

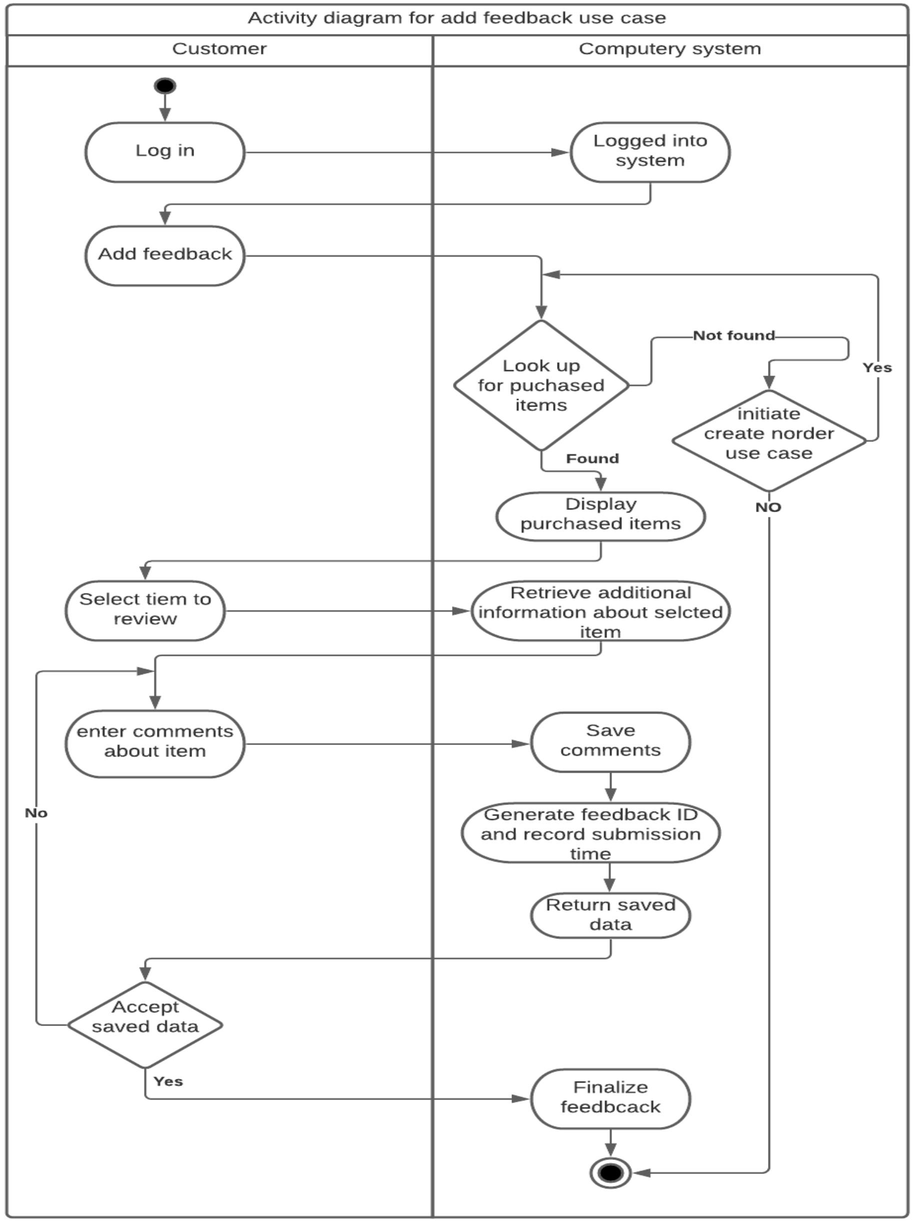
Sequence diagram for Order entry subsystem


Customer
:System
cancelOrder page := requestToCancelOrder ()
Go to check order status
Alt
[Order state is cooking]
cancelOrder confirmation := cancelOrder (orderID)
[Order state is not arrived]
requestToCancelOrder (order ID)
PayFine (fineAmount)
Alt
[Credit card has enought money]
WithdrawFine ( )
[Else]
addNegativeCharge ()
[Order state is arrived]
cancelOrder confirmation := cancelOrder(orderID)
OrderCanNotBeCancelled confirmation

State machine for Order object



Activity diagram for Add feedback use case



Activity diagram for add feedback use case

Customer

Computery system

Log in

Logged into system

Add feedback

Look up for puchased items

Not found

Yes

initiate create norder use case

Found

NO

Display purchased items

Select tiem to review

Retrieve additional information about selcted item

enter comments about item

Save comments

No

Generate feedback ID and record submission time

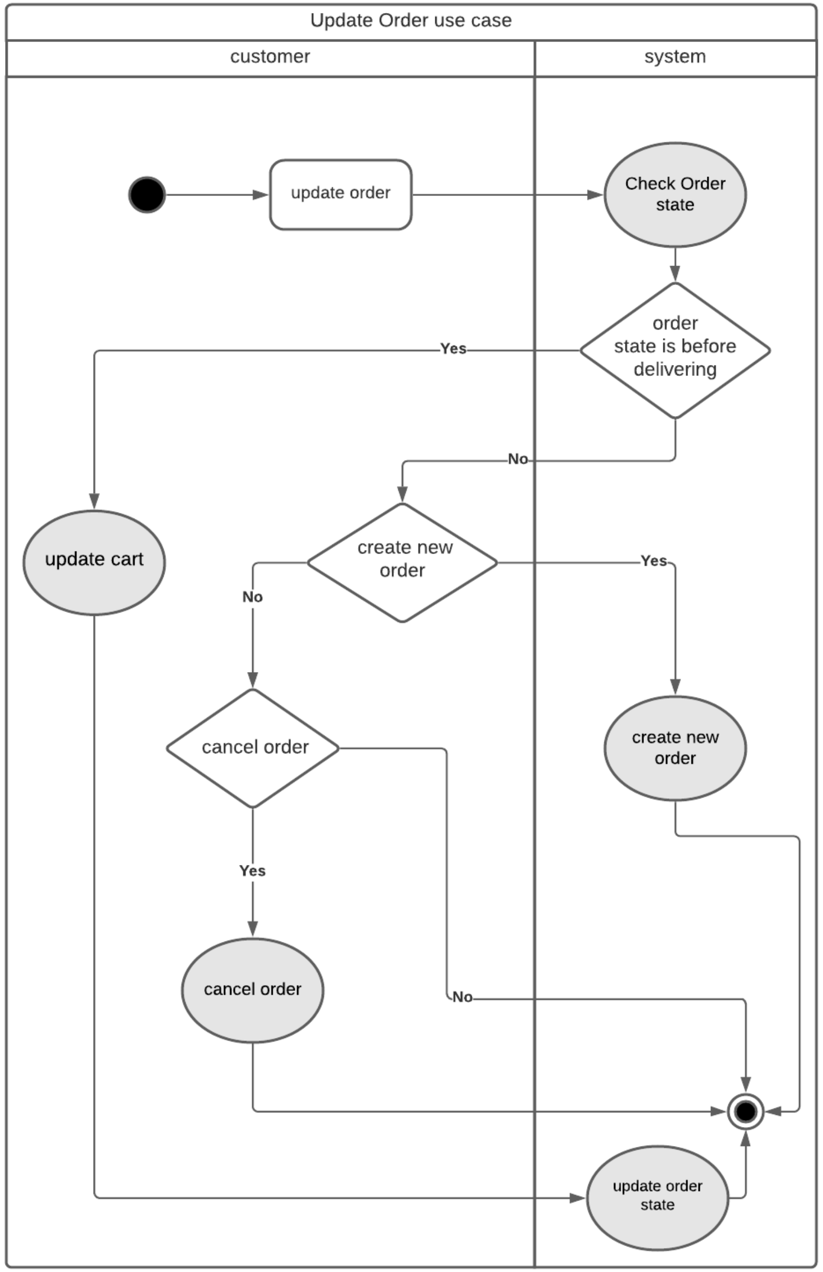
Return saved data

Accept saved data

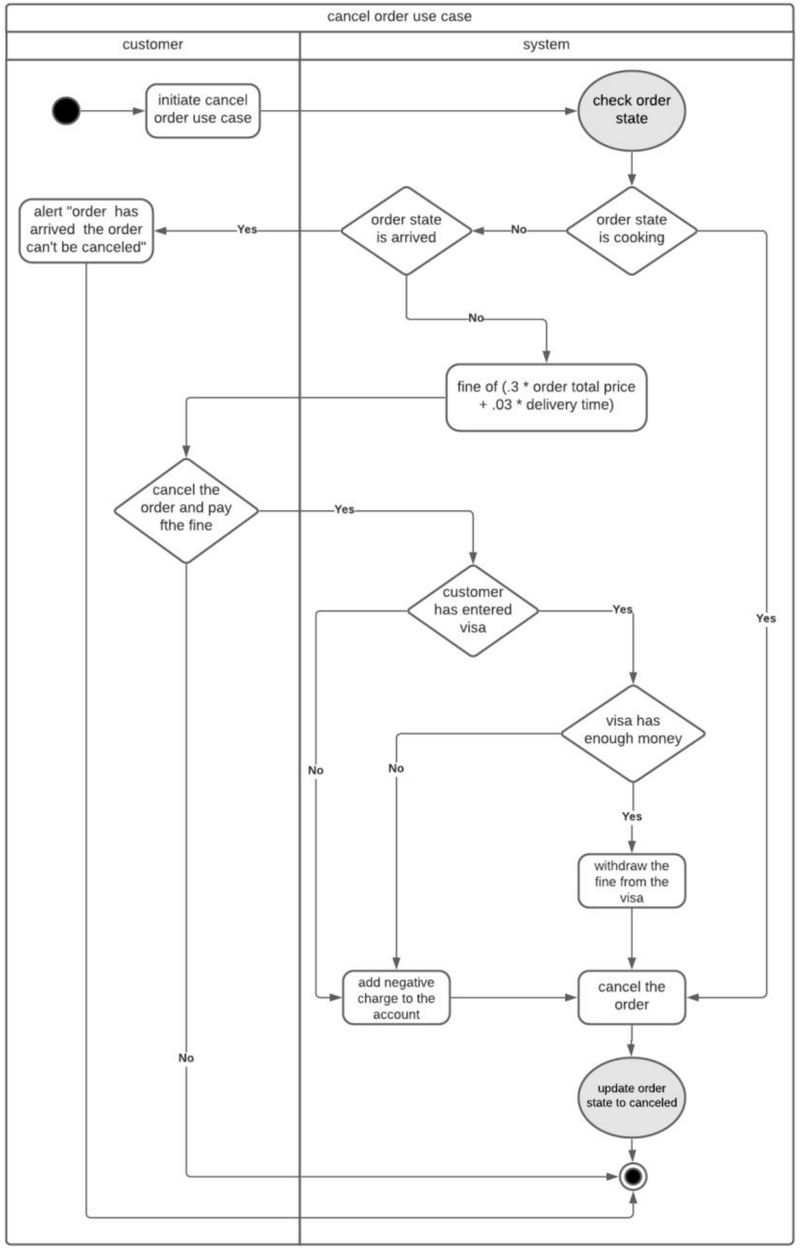
Yes

Finalize feedbcack

Activity diagram for Update order use case



Activity Diagram for Cancel order use case



cancel order use case

customer

system

initiate cancel order use case

check order state

order state is cooking

order state is arrived

alert "order has arrived the order can't be canceled"

fine of (.3 * order total price + .03 * delivery time)

cancel the order and pay fthe fine

customer has entered visa

visa has enough money

withdraw the fine from the visa

add negative charge to the account

cancel the order

update order state to canceled

Yes

No

Scenario for Add meal use case

Use case name:	**Add meal.**	
Actors:	Admin.	
Precondition:	Meal is not in menu	
Postcondition:	Meal is in menu	
Flow of activities:	Actor	System
	1. Admin initiates adding a new meal. 2. Admin enters meal name. 3. Admin enters meal description and ingredients. 4. Admin attaches meal's pictures. 5. Admin verifies returned data. 6. Admin confirm returned data.	1.1 System adds a new meal. 1.2 System creates a new page for the meal. 1.3 System requests to enter meal name. 2.1 System requests to enter meal description and ingredients. 3.1 System requests to attach meal's pictures. 4.1 System returns valid data to confirm. 6.1 System receive confirmation 6.2 System finishes add new me.
Exception conditions:	1.1 If the meal name is already on the website, then the system notify the admin to Change the name of the meal. 1.2 if the meal was in the database of the site and was removed before then a. the data about the meal can be reinserted by system, or b. admin insert new data about the meal. 5.1 if Admin wants to change some details of the meal then he can update the description of the meal.	

Scenario for Update meal use case

Use case name:	**Update meal description.**	
Actors:	Admin.	
Precondition:	Meal is existing in menu.	
Postcondition:	None.	
Flow of activities:	Actor	System
	1. Admin starts to update a meal.	1.1 System starts updating a meal.
	2. Admin selects a meal from the menu to update.	1.2 System displays menu.
	3. Admin checks meal name to update	2.1 System display all details about the selected meal
	4. Admin checks meal description to update.	3.1 System updates meal name.
	5. Admin checks meal picture to update.	4.1 System updates meal description.
	6. Admin check meal price to update	5.1 System updates the meal picture.
	7. Admin acknowledging finishing updating.	5.1 System returns valid data to confirm updates.
	8. Admin verifies returned valid data.	6.1 System updates meal price
	9. Admin confirms all updates	7.1 System returns all updated data to confirm
		9.1 System finishes updating data.
Exception conditions:	3.1 If the admin does want to update the meal name then he leaves it as it is. 4.1 If the admin does want to update the meal description then he leaves it as it is. 5.1 If the admin does want to update the meal picture then he leaves it as it is 6.1 If the admin does want to update the meal price then he leaves it as it is. 8.1 if admin refuses returned data, then a. Admin can apport the updating process and changes won't apply, or b. Admin can change the inserted data.	

Scenario for Delete meal use case

Use case name:	Delete Meal.	
Actors:	Admin.	
Precondition:	Meal is existing in menu.	
Postcondition:	None.	
Flow of activities:	Actor	System
	1. Admin initiates desire to delete a meal 2. Admin selects a meal to remove from the menu. 3. Admin confirms removing the meal.	1.1 System starts deleting a meal. 1.2 System displays the menu. 2.1 System returns confirmation for removing the meal. 3.1 System receives confirmation. 3.2 System removes the meal and keeps data related to the meal in the database.
Exception conditions:		

Scenario for Add new offer use case

Use case name:	**Add new offer.**	
Actors:	Admin.	
Precondition:	1.The offer should be added to an existing meal.	
Postcondition:	1.The offer is terminated by the system after a time assigned by the admin.	
Flow of activities:	Actor	System
	1. Admin selects a meal for adding an offer to the meal. 2. Admin enters the discount percentage. 3. Admin enters the duration of the offer. 4. Admin verifies returned valid data. 5. Admin confirm valid data.	1.1 System starts to add an offer for the meal. 2.1 System adds the discount to the meal's prince showing the customer the normal price and the offered one. 3.1 System adds duration to the offer and shows the remaining time of the offer until canceling the offer. 3.2 System return valid data for confirmation. 5.1 System receives confirmation. 5.2 System finishes adding offer.
Exception conditions:	1.if the offer is added to non existing meal, then a.the system notify the admin that the offer can't be applied. 4.1 if admin refuses the returned data,then a. admin can terminate the process. b. admin can change the inserted data.	

Scenario for Look up for meal use case

Use case name	Look up for meal	
Actor	Customer	
Preconditions	1. keywords entered 2. Search button pressed	
Postcoditions	Find meal or not	
Main activity flow	actor	system
	1-customer enters search meal	1.1-system receives keywords meal 1.2-system searches In meals 1.3-system may find meal 1.4-system requests information meal 1.5-system returns information meal
Exceptions	1. if customer enters incorrect meal , system display message alerting customer . 2. -if customer enters null, system display message alerting customer .	

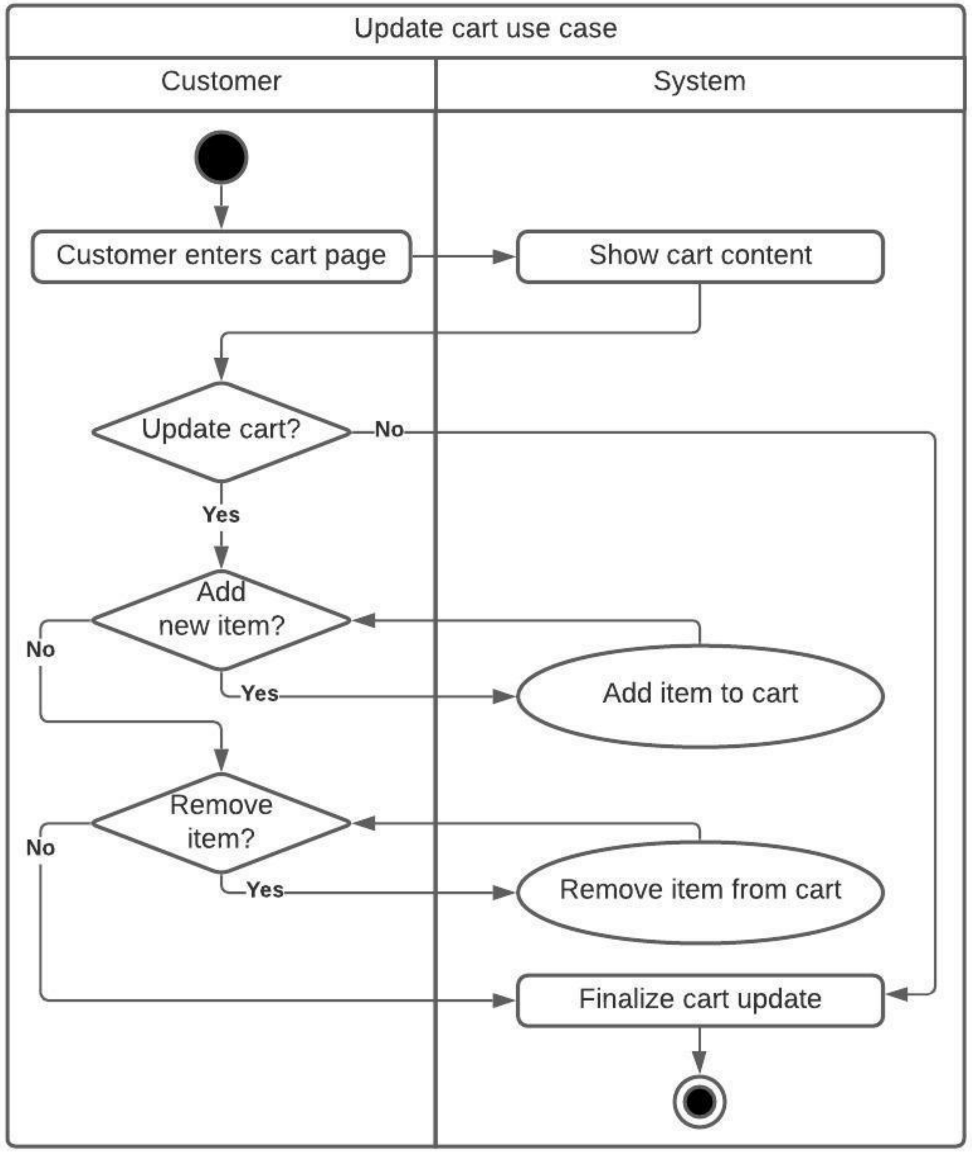
Scenario for Add item to cart use case

Use Case Name:	Add item to cart.	
Actors:	Customer.	
Preconditions:	Customer must have an account. Customer account must be logged in.	
Post conditions:	item information must be updated. (number of items left in stock). Cart content must be updated.	
Flow of activities:	Actor	System
	1. Customer wants to add item. 2. Customer adds item to his/her cart. 1. Customer enters number of items to be added. 1. Customer confirms the addition process.	1. 2. System checks product availability. 3. System prompts for number of items. 4. System verifies number of items left in stock. 5. System updates number of items left in stock. 6. System prompts for confirmation. 7. System returns product description. 8. System returns cart component.
Exception conditions:	1. If product is out of stock, then customer can a. Choose not to purchase item. b. Request item to be added as a back-ordered item. 1. If the number of items is not enough, then customer can a. Reduce the number of items. b. Request the rest number as a back-ordered items.	

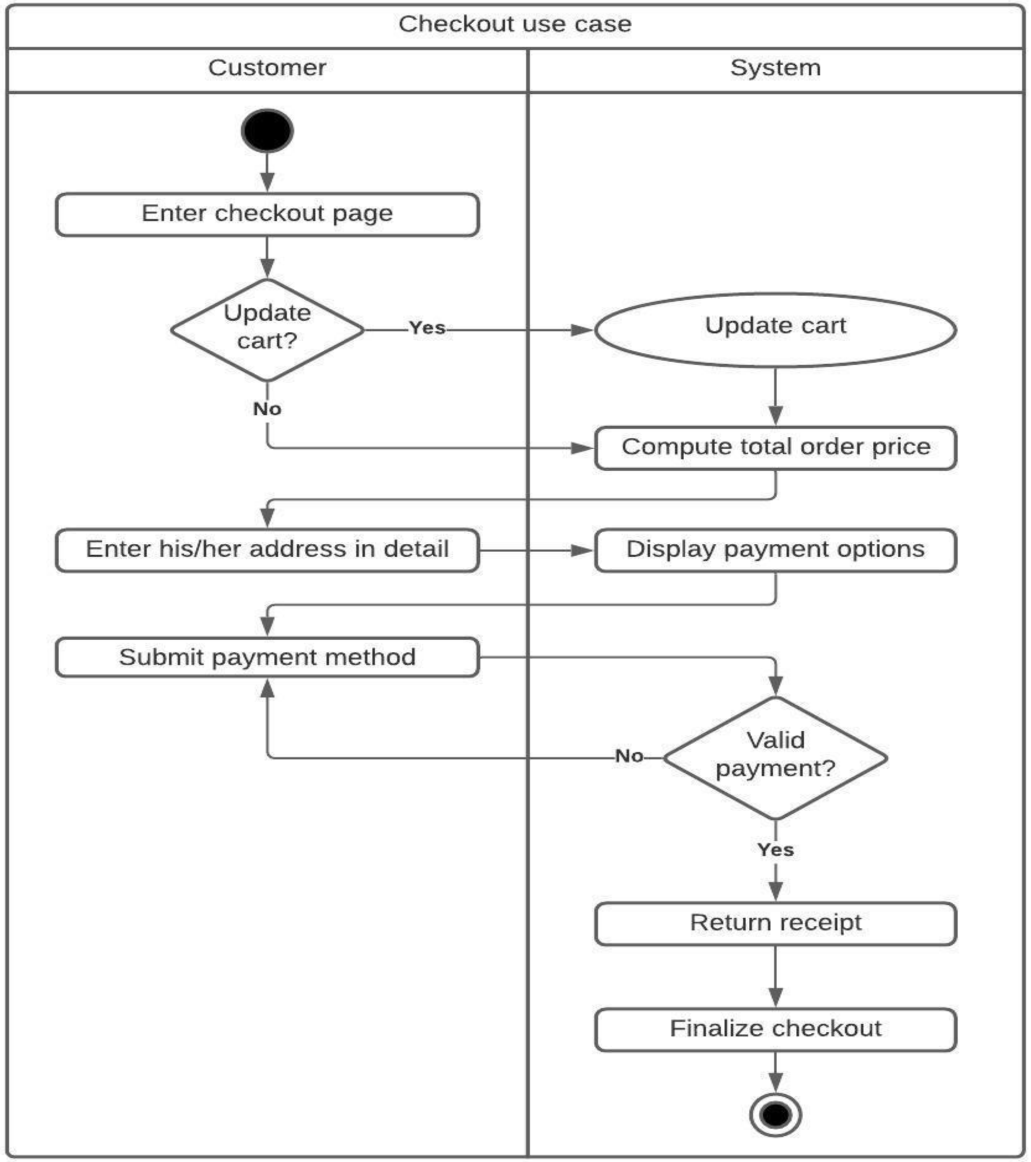
Scenario for Remove item from cart use case

Use Case Name:	Remove item from cart.	
Actors:	Customer.	
Preconditions:	Customer must have an account. Customer account must be logged in. Customer should provide the number of items will be removed.	
Post conditions:	Product information must be updated. (number of items left in stock). Cart content must be updated.	
Flow of activities:	Actor	System
	1. Customer wants to remove item. 2. Customer removes item to his/her cart. 1. Customer enters number of items to be removed. 1. Customer confirms removal process.	1. System checks cart content. 2. System prompts for number of items. 3. System updates number of items left in stock. 4. System prompts for confirmation. 5. System returns cart content. 6. System offers recommended similar products.
Exception conditions:	1. If product is not in the cart, then ask the customer to enter the name correctly. 2. If the number of items to be removed is larger than the number of items in cart, then ask the customer to enter a number less than or equal number of items in the cart.	

Activity diagram for update cart use case



Activity diagram for check out use case


Checkout use case
Customer
System
Enter checkout page
Update cart?
Yes
No
Update cart
Compute total order price
Enter his/her address in detail
Display payment options
Submit payment method
Valid payment?
No
Yes
Return receipt
Finalize checkout