Group 3 - Phase 2 Description

Matthew Jeyapaul
Anthony Matthews
Anjan Narayanaswamy
Matthew Lay

**(1) Refined project description**

Our team project is to create an online order service for Pantera Bread (off brand Panera Bread). This service would keep track of the quantity of each item and total the price at the end during checkout. We will have to create a directory for the price of each item. A user will be able to pick which food they want, what delivery type they want, and enter their payment information. An employee will be able to receive this information in the form of a ticket and process the order while the meal is prepared and delivered, essentially updating the ticket. Based on the delivery type chosen by the user, the ticket will be updated to either be ready for pickup in store or delivered. A manager would be able to see both the customer view and the employee view.

**(2) Use cases and their variations**

Use Case #1 - Customer

1. The customer visits the Pantera website to order food.
2. The customer logs in with username and password.
   Username:
   Password:
3. Customer is able to add food items
4. Customer then click Finalize Cart button when done ordering
5. Customer is then taken to the Delivery Menu screen.
   Pickup
   Delivery
6. The customer is then taken to the payment screen.
7. The customer is asked to enter information.
   CCNUM
   CVC
   EXP
   ZIP
8. Order is placed

Variation #1

1.1 Customer enters invalid username and password.
1.2 The website displays information.
   Sign In Denied
1.3 Customer is returned to login screen.

Variation #2

2.1 At Step 5, if the customer chooses Delivery, he will be asked for his address for the location of where the food should be delivered

Variation #3

3.1 Customer enters invalid credit card information

3.2 Given two options of canceling order or reentering credit card information

Use Case #2 - Employee

1. The employee visits the website the view the Current Ticket
2. Employees can change the status of the Current Ticket by acknowledging that the food is ready.
3. Once the Current Ticket is done, the employee logs out and then logs in to view the next ticket in the queue.

Variations:

1.1 If the order type is delivery, once the food is prepared, the status will change to “Order Delivery in Progress”, wait 5 seconds, then change to “Food is Delivered”

Use Case #3 - Manager

1. The manager visits the website to view the current Order Queue.
2. The manager can view the Tickets in the OrderQueue.
3. The manager can logout when they are done.

**(3) CRC cards**

| OrderQueue | |
|---|---|
| Has a queue of Tickets<br>Display queue<br>Check logistics (order fulfilled, unfulfilled) | Ticket<br>Viewer<br>Viewer |

| Connection | |
|---|---|
| Has a state for menu to use<br>Allows user to login or log out | Interface |

| Interface | |
|---|---|
| GUI that allows for user input<br>Send input to another class so it can use it | Connection |

| Manager | |
|---|---|
| See fulfilled orders<br>See unfulfilled orders<br>See order logistics | OrderQueue<br><br>Ticket |

| Employee | |
|---|---|
| Viewqueue<br>Change order status | Viewer<br>Ticket |

| Ticket | |
|---|---|
| Hold order status<br>Display order status<br>Add/Remove from Queue<br>Get order number (#0001 ~ 9999)<br>Get time (for delivery)<br>Get fulfilled time (after fulfilled)<br>Get food/quantity information | Viewer<br>OrderQueue<br><br><br>Model |

| Food | |
|---|---|
| getPrice(int price); adds item price to total<br>getType (returns the type of food)<br>getSize (returns the item's size)<br>getKind (gets any modifiers, such as temperature) | Extends to:<br>Soup, Sandwich, Salad, Grain Bowl, MacNCheese |

| MacNCheese | |
|---|---|
| getType: MacNCheese<br>getSize: Small or large<br>getPrice: Based on size | (extends Food) |

## Salad

| | |
|---|---|
| getType: Salad<br>getSize: Small or large<br>getPrice: Based on size | (extends Food) |

## Sandwich

| | |
|---|---|
| getType: Sandwich<br>getSize: Small or large<br>getPrice: Based on size<br>getKind: Hot or cold | (extends Food) |

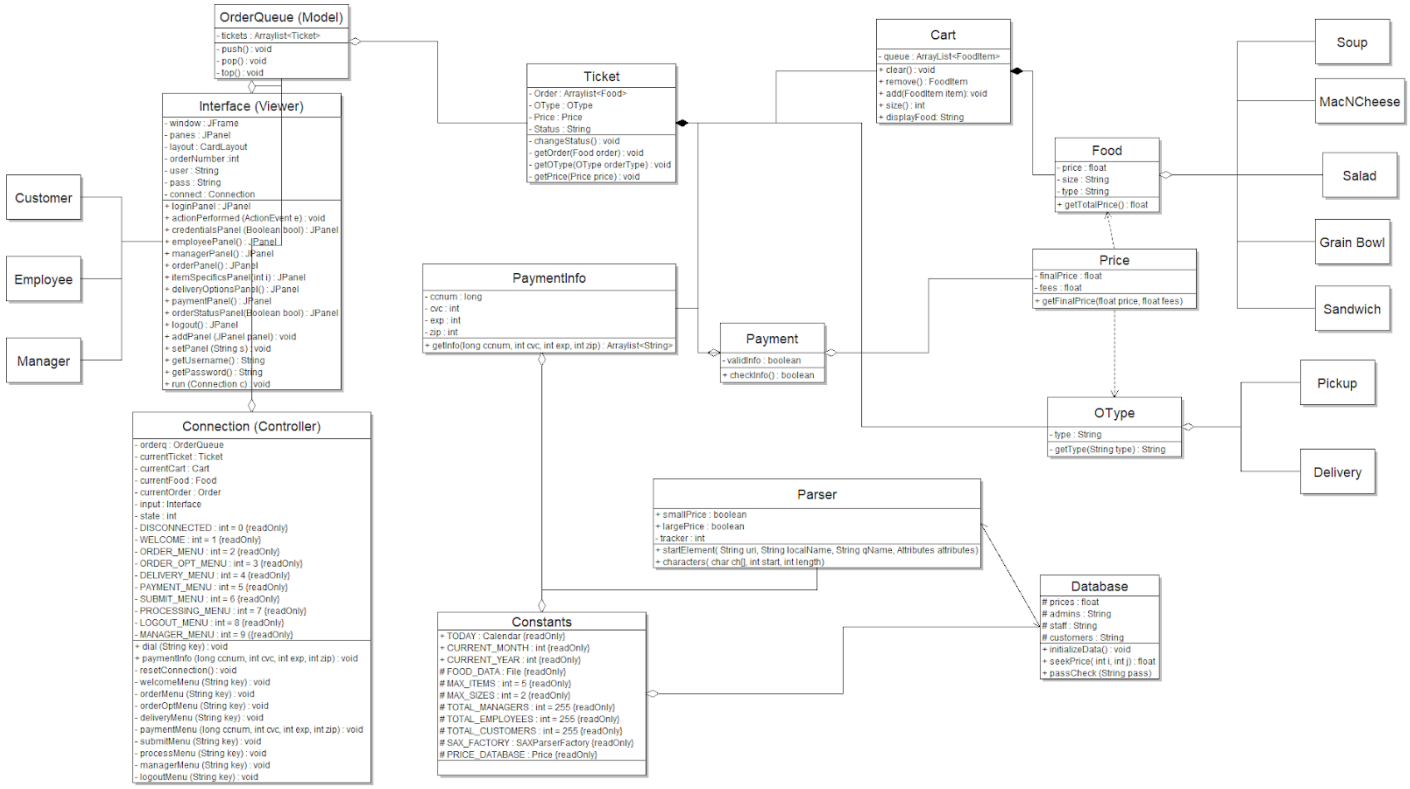
| Soup | |
|---|---|
| getType: Soup<br>getSize: Small or large<br>getPrice: Based on type size | (extends Food) |

| Grain Bowl | |
|---|---|
| getType: Grain Bowl<br>getSize: Large<br>getPrice: Will be a constant | (extends Food) |

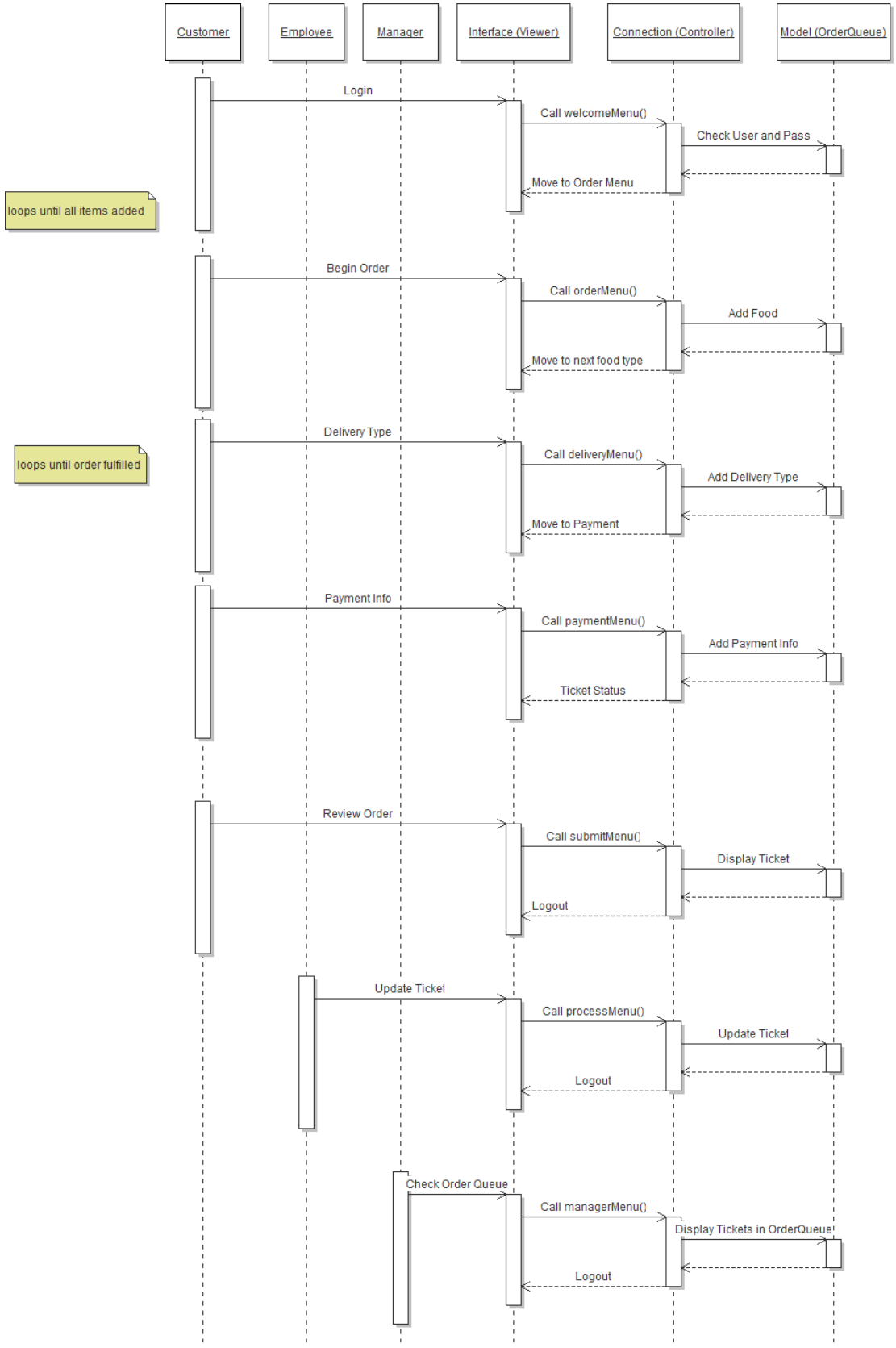
| Parser | |
|---|---|
| Reads from an .XML file to obtain data | |

| Database | |
|---|---|
| Uses a parser to get data, such as prices<br>initializeData() via the parser<br>seekPrice() when determining final price | Parser |

## (4) Class diagram



## (5) Sequence diagram



## (6) State diagrams