**Online Food Delivery Application (Qwick Foodz)**

Create an online application for Food Delivery Platform.

The application will show the following main menu.

1. **Customer Registration**
2. **Customer Login**
3. **Exit**
4. **Customer Registration:**
   1. Collect all the details of user.
      1. Name,
      2. FatherName,
      3. Gender- {Select, Male, Female, Transgender},
      4. Mobile,
      5. DOB,
      6. MailID,
   2. Location and register him as a customer and show: **“Customer registration successful Your Customer ID: CID1001.”**
5. **Customer Login**
   1. Get **CustomerID** from user and check whether his CustomerID exists. If not means show “Invalid user ID”. If exists means Show following sub menu.
      1. **Show Profile**
      2. **Order Food**
      3. **Cancel Order**
      4. **Modify Order**
      5. **Order History**
      6. **Recharge Wallet**
      7. **Show Wallet Balance**
      8. **Exit**
6. **Show Profile:**

Show the current logged in customer profile information in console.

1. **Order Food:**
   1. Create **OrderDetails** object with **CustomerID**, **TotalPrice** = 0, **DateOfOrder** as today and **OrderStatus** = Initiated.
   2. Create **localItemList** for adding your wishlist.
   3. Show all the list of food items available in store for processing the order.
   4. Ask the **FoodID**, order quantity from user and check whether it is available. If not show **FoodID** Invalid or **FoodQuantity** unavailable based on the scenario.
   5. If it is available then, create **ItemDetails** object with created **OrderID**, **FoodID**, **Quantity** and **Price** of this order, deduct the available quantity and store the **ItemDetails** object in **localItemList**. Calculate total price of this order by summing it with current item’s price.
   6. Ask the user whether he want to order more. If “**Yes**” means again show food details and **repeat from step “c”** to create new **ItemDetails** object. Repeat this process until he says “**No**”.
   7. If user select “No” then show “Do you want to confirm purchase.” If he says “**No**” return the **localItemList** of items count back to **FoodDetails** list.
   8. If he says “**Yes**”.
   9. Check whether the customer wallet has sufficient balance.
   10. If he has balance then, modify **OrderDetails** object which was created at beginning with **TotalPrice** and **OrderStatus** to “**Ordered**”. Deduct the total amount from user’s wallet balance. Add the **localItemList** to **ItemList**.
   11. If the balance is insufficient, inform the customer that the wallet has insufficient balance and whether wish to recharge /not.
   12. If he says “No” return the **localItemList** item’s count to **FoodList**.
   13. If “Yes” ask for the amount to be recharged. Show the balance after recharge and goto step **“i”** to proceed purchase again.
2. **Cancel Order:**
   1. Show the list of orders placed by current logged in user whose status is **“Ordered”**.
   2. Ask the user to pick an order to be cancelled by **OrderID**.
   3. If **OrderID** is present, then change the order status to **“Cancelled”**. Refund the total price amount of current order to user’s **WalletBalance** then return the food items of the current order to **FoodList**.
3. **Modify Order** 
   1. Show the orders placed by current logged in user whose order status is booked.
   2. Ask **OrderID** to modify orders
   3. Check **OrderID** is valid, and it is of current user’s and its status is **Ordered**. Then fetch the item details of corresponding order and show it.
   4. Ask **ItemID** and check **ItemID** is valid. Then ask user to provide new item quantity.
   5. Based on new item quantity the item object needs to be updated its price.
   6. If item quantity increased, then item amount will be collected from current user if he has enough balance. If he has no balance, ask him to recharge with that amount and proceed. If the item quantity reduced, then balance amount should be returned to current user.
   7. Update the total amount of order and show **“Order ID + (OID3001) + modified successfully**”.
4. **Order History**
   1. Show the order history of the current logged-in user.
5. **Recharge Wallet:**
   1. Ask the user to enter the amount to be recharged. Deposit the recharge amount to current user’s wallet.
6. **Show Wallet Balance**

Display the customer’s wallet balance in console.

1. **Exit**
   1. Exit from submenu and show main menu.

**Personal Details Class:**

**Properties**: Name, FatherName, Gender- {Select, Male, Female, Transgender}, Mobile, DOB, MailID, Location

**Interface IBalance**

Properties: WalletBalance

Method: WalletRecharge, DeductBalance

**CustomerDetails class: Inherits Personal Details, IBalance**

**Field:** \_balance

**Properties**: CustomerID, WalletBalance

**Methods**: WalletRecharge, DeductBalance

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CustomerID** | **WalletBalance** | **Name** | **FatherName** | **Gender** | **Mobile** | **DOB** | **MailID** | **Location** |
| CID1001 | 10000 | Ravi | Ettapparajan | Male | 974774646 | 11/11/1999 | ravi@gmail.com | Chennai |
| CID1002 | 15000 | Baskaran | Sethurajan | Male | 847575775 | 11/11/1999 | baskaran@gmail.com | Chennai |

**Values of CustomerDetails Object:**

**FoodDetails Class:**

**Properties**: FoodID, FoodName, PricePerQuantity, QuantityAvailable

**Values of FoodDetails Object**

|  |  |  |  |
| --- | --- | --- | --- |
| **FoodID** | **FoodName** | **PricePerQuantity** | **QuantityAvailable** |
| FID2001 | Chicken Briyani 1Kg | 100 | 12 |
| FID2002 | Mutton Briyani 1Kg | 150 | 10 |
| FID2003 | Veg Full Meals | 80 | 30 |
| FID2004 | Noodles | 100 | 40 |
| FID2005 | Dosa | 40 | 40 |
| FID2006 | Idly (2 pieces) | 20 | 50 |
| FID2007 | Pongal | 40 | 20 |
| FID2008 | Vegetable Briyani | 80 | 15 |
| FID2009 | Lemon Rice | 50 | 30 |
| FID2010 | Veg Pulav | 120 | 30 |
| FID2011 | Chicken 65 (200 Grams) | 75 | 30 |

**OrderDetails Class:**

**Properties**: OrderID, CustomerID, TotalPrice, DateOfOrder, OrderStatus – {Default, Initiated, Ordered, Cancelled}.

**Object Details of OrderDetails class:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **TotalPrice** | **DateOfOrder** | **OrderStatus** |
| OID3001 | CID1001 | 580 | 26/11/2022 | Ordered |
| OID3002 | CID1002 | 870 | 26/11/2022 | Ordered |
| OID3003 | CID1001 | 820 | 26/11/2022 | Cancelled |

**ItemDetails Class:**

**Properties**: ItemID – (ITID100), OrderID, FoodID, PurchaseCount, PriceOfOrder

**Default objects for ItemDetails class:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ItemID** | **OrderID** | **FoodID** | **PurchaseCount** | **PriceOfOrder** |
| ITID4001 | OID3001 | FID2001 | 2 | 200 |
| ITID4002 | OID3001 | FID2002 | 2 | 300 |
| ITID4003 | OID3001 | FID2003 | 1 | 80 |
| ITID4004 | OID3002 | FID2001 | 1 | 100 |
| ITID4005 | OID3002 | FID2002 | 4 | 600 |
| ITID4006 | OID3002 | FID2010 | 1 | 120 |
| ITID4007 | OID3002 | FID2009 | 1 | 50 |
| ITID4008 | OID3003 | FID2002 | 2 | 300 |
| ITID4009 | OID3003 | FID2008 | 4 | 320 |
| ITID4010 | OID3003 | FID2001 | 2 | 200 |