CAKE ORDER SYSTEM

Submitted By :- Piyush kalidas Wadhai

Project Guide:- Gopal JBK

JULY 17, 2024

PSK_TECHNOLOGIES PRIVATE LIMITADED IT COMPANY

Task 1: Cake Ordering System

- **The Cake class has been created with attributes name, price, and description.**
- **The CakeOrder class has been created with attributes cake, quantity, and totalPrice.**
- **A method has been implemented to calculate the total price of an order based on the quantity of cakes ordered.**
- **A method has been implemented to display the order details, including the cake name, quantity, and total price.**

```
package Cake_Oredering_system;

public class Cake {
    private String name;
    private double price;
    private String description;

    public Cake(String name, double price, String
description) {
        this.name = name;
        this.price = price;
        this.description = description;
    }

    public String getName() {
        return name;
    }

    public double getPrice() {
        return price;
    }

    public String getDescription() {
        return description;
    }
}
```

```java
package Cake_Oredering_system;
// Task 1: Cake Ordering System



    public class Cake_Order {
        private Cake cake;
        private int quantity;
        private double totalPrice;

        public Cake_Order(Cake cake, int quantity)
{
            this.cake = cake;
            this.quantity = quantity;
            this.totalPrice =
calculateTotalPrice();
        }

        public double calculateTotalPrice() {
            return cake.getPrice() * quantity;
        }

        public void displayOrderDetails() {
            System.out.println("Cake Name: " +
cake.getName());
            System.out.println("Quantity: " +
quantity);
            System.out.println("Total Price: " +
totalPrice);
        }
    }
```

Task 2: Inventory Management System

- **The Ingredient class has been created with attributes name, quantity, and unitPrice.**
- **The BakedGood class has been created with attributes name, price, and ingredients (a list of Ingredient objects).**
- **A method has been implemented to calculate the total cost of ingredients for a baked good.**
- **A method has been implemented to update the inventory levels of ingredients when a baked good is sold.**

```
package Inventory_Management_system;

public class Ingredient {
  private String name;
  private int quantity;
  private double unitPrice;

  public Ingredient(String name, int quantity,
double unitPrice) {
      this.name = name;
      this.quantity = quantity;
      this.unitPrice = unitPrice;
  }

  public String getName() {
      return name;
  }

  public int getQuantity() {
      return quantity;
  }

  public double getUnitPrice() {
      return unitPrice;
  }

  /**
   * @param quantity the quantity to set
   */
```

```
    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

package Inventory_Management_system;

import java.util.List;

public class BakedGood {
    private String name;
    private double price;
    private List<Ingredient> ingredients;

    public  BakedGood(String  name,  double  price,
List<Ingredient> ingredients) {
        this.name = name;
        this.price = price;
        this.ingredients = ingredients;
    }

    public double calculateTotalCost() {
        double totalCost = 0;
        for (Ingredient ingredient : ingredients) {
            totalCost += ingredient.getQuantity() *
ingredient.getUnitPrice();
        }
        return totalCost;
    }

    public void updateInventoryLevels() {
        for (Ingredient ingredient : ingredients) {

    ingredient.setQuantity(ingredient.getQuantity()
- 1);
        }
    }
}
```

Task 3: Point of Sale (POS) System

- **The Transaction class has been created with attributes date, amount, and paymentType.**
- **The Payment class has been created with attributes paymentType and amount.**
- **A method has been implemented to process a payment and update the transaction amount.**
- **A method has been implemented to generate a receipt for a transaction.**

```
package Point_of_Sale__System;

import java.sql.Date;

// Transaction.java
public class Transaction {
  private Date date;
  private double amount;
  private String paymentType;

  public Transaction(Date date, double amount,
String paymentType) {
      this.date = date;
      this.amount = amount;
      this.paymentType = paymentType;
  }

  public Date getDate() {
      return date;
  }

  public double getAmount() {
      return amount;
  }

  public String getPaymentType() {
      return paymentType;
```

```
    }

    public void processPayment(Payment payment) {
        this.amount -= payment.getAmount();
    }

    public String generateReceipt() {
        return "Receipt for transaction on " + date
  + "\n" + "Amount: " + amount + "\n" + "Payment
  Type: "
                + paymentType;
    }
  }
```

```
package Point_of_Sale__System;

public class Payment {
    private String paymentType;
    private double amount;

    public Payment(String paymentType, double
amount) {
        this.paymentType =paymentType;
        this.amount = amount;
    }

    public String getPaymentType() {
        return paymentType;
    }

    public double getAmount() {
        return amount;
    }
}
```

Task 4: Customer Management System

- **The Customer class has been created with attributes name, email, and phone.**
- **A method has been implemented to add a new customer to the system.**
- **A method has been implemented to retrieve a customer's details based on their email or phone number.**

```
package Customer_Management_System;

public class Customer {
    private String name;
    private String email;
    private String phone;

    public Customer(String name, String email,
String phone) {
        this.name = name;
        this.email = email;
        this.phone = phone;
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    public String getPhone() {
        return phone;
    }
}
```

```java
package Customer_Management_System;

public class CustomerSystem {
    private Customer customer;

    public void addCustomer(String name, String
email, String phone) {
        customer = new Customer(name, email,
phone);
        System.out.println("Customer added
successfully.");
    }

    public void getCustomerDetails(String
emailOrPhone) {
        if
(customer.getEmail().equals(emailOrPhone) ||
customer.getPhone().equals(emailOrPhone)) {
            System.out.println("Customer details:
");
            System.out.println("Name: " +
customer.getName());
            System.out.println("Email: " +
customer.getEmail());
            System.out.println("Phone: " +
customer.getPhone());
        }
        else {
            System.out.println();
        }
    }
}
```

Task 5: Order Management System

- **The Order class has been created with attributes customer, date, and status.**
- **A method has been implemented to create a new order and assign it to a customer.**
- **A method has been implemented to update the status of an order (e.g. from "pending" to "delivered").**

```
package Order_Management_System;
import java.sql.Date;
public class Order {
    private String customer;
    private Date date;
    private String status;
    public Order(String customer, Date date, String
status) {
        this.customer = customer;
        this.date = date;
        this.status = status;
    }
    public String getCustomer() {
        return customer;
    }
    public Date getDate() {
        return date;
    }
    public String getStatus() {
        return status;
    }
    public void updateStatus(String newStatus) {
        this.status = newStatus;
    }
}
```

Task 6: Reporting System

- The Reporting System generates reports on daily sales and top-selling baked goods. The system consists of two methods:
- The generateDailySalesReport method generates a report of daily sales, including the total amount sold and the number of transactions.
- The generateTopSellingBakedGoodsReport method generates a report of top-selling baked goods, including the name and quantity sold.

```
package com.Piyush.Reporting_System;

import java.util.ArrayList;

import java.util.List;

import com.Piyush.COS_Main.BakedGood1;

public class ReportingSystem {

    private List<Transaction1> transactions;

    private List<BakedGood1> bakedGoods1;

    public ReportingSystem(List<Transaction1>
transactions, List<BakedGood1> bakedGoods1) {

        this.transactions = transactions;

        this.bakedGoods1 = bakedGoods1;

    }

    public void generateDailySalesReport() {

        double totalAmount = 0;

        int numTransactions = 0;
```

```
        for (Transaction1 transaction :
transactions) {

            totalAmount += transaction.getAmount();

            numTransactions++;

        }

        System.out.println("Total Amount Sold: " +
totalAmount);

        System.out.println("Number of Transactions:
" + numTransactions);

    }

    public void
generateTopSellingBakedGoodsReport() {

        List<BakedGood1> topSellingBakedGoods = new
ArrayList<BakedGood1>();

        topSellingBakedGoods.add(new
BakedGood1("butter", 5));

        topSellingBakedGoods.add(new
BakedGood1("Bread", 10));

        topSellingBakedGoods.add(new
BakedGood1("cookie", 19));

        topSellingBakedGoods.add(new
BakedGood1("Pastries", 16));

        topSellingBakedGoods.add(new
BakedGood1("Muffins", 15));

        topSellingBakedGoods.add(new
BakedGood1("Pies", 22));
```

```
        topSellingBakedGoods.add(new
BakedGood1("Apple tart bakery", 11));

        topSellingBakedGoods.add(new
BakedGood1("Artisanal crusts", 12));

        for (BakedGood1 bakedGood :
topSellingBakedGoods) {

            System.out.println("Top Selling Baked
Goods: "+bakedGood.getName() + ": " +
bakedGood.getQuantitySold());

        }

    }

    @Override

    public String toString() {

        return "ReportingSystem [transactions=" +
transactions + ", bakedGoods1=" + bakedGoods1 +
"]";

    }}
```

```
package com.Piyush.Reporting_System;

public class Transaction1 {
     private double amount;

     public Transaction1(double amount) {
          this.amount = amount;
     }

     public double getAmount() {
          return amount;
     }

}
```

```
package com.Piyush.COS_Main;

public class BakedGood1 {
```

```
	private String name;
	private int quantitySold;

	public BakedGood1(String name, int quantitySold) {
		this.name = name;
		this.quantitySold = quantitySold;
	}

	public String getName() {
		return name;
	}

	public int getQuantitySold() {
		return quantitySold;
	}

	public void setName(String name) {
		this.name = name;
	}

	public void setQuantitySold(int quantitySold) {
		this.quantitySold = quantitySold;
	}

	@Override
	public String toString() {
		return "BakedGood1 [name=" + name + ", quantitySold=" +
quantitySold + "]";
	}

}
```

Cake Control:-

```
package com.Piyush.COS_Main;

import java.sql.Date;

import java.util.ArrayList;

import java.util.List;

import
```

```
com.Piyush.Customer_Management_System.CustomerSystem;

import com.Piyush.Inventory_Management_system.BakedGood;

import com.Piyush.Inventory_Management_system.Ingredient;

import com.Piyush.Order_Management_System.Order;

import com.Piyush.Point_of_Sale__System.Payment;

import com.Piyush.Point_of_Sale__System.Transaction;

import com.Piyush.Reporting_System.ReportingSystem;

import com.Piyush.Reporting_System.Transaction1;

import com.Piyush.piyush.Cake_Oredering_system.Cake;

import com.Piyush.piyush.Cake_Oredering_system.Cake_Order;


public class Manager {


    public static void main(String[] args) {

        // Task 1: Cake Ordering System

        System.out.println("__________😋😋WELCOME PIYUSH
CAKE SHOPE😋😋😋 __________");

        System.out.println("......🍰🍰Detail of cake
🍰🍰.....");


        Cake a = new Cake("Blackforest cake", 250.0,
"Moist chocolate cake");

        Cake b = new Cake("whiteforest cake", 300.0,
"Moist chocolate cake");
```

```
        Cake c = new Cake("Strawberry Shortcake", 500.0,
                "A sweet and tangy strawberry cake
layered with whipped cream and fresh ");
        Cake d = new Cake("Carrot Cake", 600.0,
                "A moist and flavorful carrot cake
filled with chopped nuts and spices, topped with a creamy
cream cheese frosting");
        Cake e = new Cake("emon Lavender Pound Cake",
900.0, "A refreshing lemon pound cake ");
        Cake_Order A = new Cake_Order(a, 2);
        A.displayOrderDetails();

    System.out.println("_____
____");

        Cake_Order B = new Cake_Order(a, 2);
        B.displayOrderDetails();

    System.out.println("_____
____");

        Cake_Order C = new Cake_Order(b, 1);
        C.displayOrderDetails();

    System.out.println("_____
____");

        Cake_Order D = new Cake_Order(a, 2);
```

```
        D.displayOrderDetails();

    System.out.println("_____
____");

        Cake_Order E = new Cake_Order(a, 2);
        E.displayOrderDetails();

    System.out.println("_____
____");

        // Task 2: Inventory Management System

        System.out.println("====Details of Ingedient
Add====");

        Ingredient sugar = new Ingredient("sugar", 50,
3.5);
        Ingredient cocoaPowder = new Ingredient("cocoa
Powder", 100, 3.5);
        Ingredient Butter = new Ingredient("Butter", 50,
1.5);
        Ingredient eggs = new Ingredient("Eggs", 20,
3.0);

        List<Ingredient> ingredients = new ArrayList<>();
```

```
        ingredients.add(Butter);

        ingredients.add(sugar);

        ingredients.add(eggs);


        BakedGood = new BakedGood("Blackforest cake",
30.0, ingredients);

        System.out.println("Name:- " +
bakedGood.getName());


        System.out.println("Total Cost of Ingedients:" +
bakedGood.calculateTotalCost());

        bakedGood.updateInventoryLevels();


        // Task3. Point of sale system

    System.out.println("_________________********_______
___________");


        System.out.println("__💰💰Payment Recipt
details💰💰__");


        Transaction = new Transaction(new Date(2024, 07,
10), 100.0, "Credit Card");

        Payment = new Payment("Cash", 250.0);

    System.out.println("_________________*****_________
________");
```

```
        Transaction transaction1 = new Transaction(new
Date(2024, 07, 20), 100.0, "online");

        Payment payment1 = new Payment("online", 300.0);



        Transaction transaction2 = new Transaction(new
Date(2024, 07, 21), 100.0, "Googlepay");

        Payment payment2 = new Payment("online", 500.0);



        Transaction transaction3 = new Transaction(new
Date(2024, 07, 22), 100.0, "cash");

        Payment payment3 = new Payment("online", 600.0);



        Transaction transaction4 = new Transaction(new
Date(2024, 07, 23), 100.0, "paytm");

        Payment payment4 = new Payment("online", 9000.0);



        Transaction transaction5 = new Transaction(new
Date(2024, 07, 24), 100.0, "Googlepay");

        Payment payment5 = new Payment("online", 1000.0);



        transaction.processPayment(payment);


    System.out.println(transaction.generateReceipt());


    System.out.println("_____
____");
```

```
        System.out.println(transaction1.generateReceipt());

        System.out.println("_____
____");

        System.out.println(transaction2.generateReceipt());

        System.out.println("_____
____");

        System.out.println(transaction3.generateReceipt());

        System.out.println("_____
____");

        System.out.println(transaction4.generateReceipt());

        System.out.println("_____
____");

        System.out.println(transaction5.generateReceipt());

        System.out.println("_____*******_____
_____");


            // Task 4 customer management system

        System.out.println("_____
____");
```

```
        System.out.println("=== 👤 👤 Customer
Details 👤 👤 ===");

    System.out.println("_____
____");

        CustomerSystem = new CustomerSystem();

    System.out.println("_____
____");

        customerSystem.addCustomer("Piyush Wadhai",
"piyushwadhai146@.com", "9021744384");

    customerSystem.getCustomerDetails("piyushwadhai146@.c
om");

    System.out.println("_____
____");

        customerSystem.addCustomer("Nandini Wadhai",
"NandiniWadhai@.com", "9021744384");

    customerSystem.getCustomerDetails("NandiniWadhai@.com
");

    System.out.println("_____
____");

    System.out.println("_____
____");
```

```
        customerSystem.addCustomer("Akash Wadhai",
"Akashwadhai146@.com", "9021744384");

    customerSystem.getCustomerDetails("Akashwadhai146@.co
m");

    System.out.println("_____________________________________
____");

        customerSystem.addCustomer(" Ankush Wadhai",
"Ankushwadhai146@.com", "9021744384");

    customerSystem.getCustomerDetails("Ankushwadhai146@.c
om");

    System.out.println("_____________________________________
____");

        // Task 5 order management system

    System.out.println("_____________________________________
____");

        // Create a new order

        System.out.println("===New order Details===");

    System.out.println("_____________________________________
____");
```

```
        Order = new Order("Piyush Wadhai", new Date(2024,
07, 17), "pending");

        System.out.println("Order created: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


    System.out.println("_____
____");



        Order order1 = new Order("Nandini Wadhai", new
Date(2024, 07, 15), "pending");

        System.out.println("Order created: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


    System.out.println("_____
____");



        Order order2 = new Order("Ankush Wadhai", new
Date(2024, 07, 16), "pending");

        System.out.println("Order created: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


    System.out.println("_____
____");



        Order order3 = new Order("Akash wadhai", new
Date(2024, 07, 20), "pending");
```

```
        System.out.println("Order created: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


    System.out.println("_____________________________________
____");



        System.out.println("========= ORDER  STATUS
=======");

        // Update the status of the order

        order.updateStatus("delivered");

        System.out.println("Order updated: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


    System.out.println("_____________________________________
____");



        order1.updateStatus("delivered");

        System.out.println("Order updated: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


    System.out.println("_____________________________________
____");



        order2.updateStatus("delivered");

        System.out.println("Order updated: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());
```

```
     System.out.println("_____
____");



          order3.updateStatus("delivered");

          System.out.println("Order updated: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


     System.out.println("_____
____");



          order.updateStatus("delivered");

          System.out.println("Order updated: " +
order.getCustomer() + ", " + order.getDate() + ", " +
order.getStatus());


     System.out.println("_____
____");



          // Task 6 Report system

          System.out.println("=======//ALL TOTAL
REPORT//=======");



          List<Transaction1> transactions = new
ArrayList<>();

          transactions.add(new Transaction1(200.0));

          transactions.add(new Transaction1(300.0));
```

```
            transactions.add(new Transaction1(400.0));
            transactions.add(new Transaction1(355.0));
            transactions.add(new Transaction1(34.0));
            transactions.add(new Transaction1(3054.0));
            transactions.add(new Transaction1(34.0));
            transactions.add(new Transaction1(355.0));
            transactions.add(new Transaction1(600.0));
            transactions.add(new Transaction1(665.0));
            transactions.add(new Transaction1(300.0));
            transactions.add(new Transaction1(354.0));
            transactions.add(new Transaction1(32.0));
            transactions.add(new Transaction1(359.0));
            transactions.add(new Transaction1(30.0));
            transactions.add(new Transaction1(900.0));
            transactions.add(new Transaction1(100.0));
            List<BakedGood1> ll = null;
ReportingSystem = new ReportingSystem(transactions, ll);
reportingSystem.generateDailySalesReport();
System.out.println("_____________________________________"
);
reportingSystem.generateTopSellingBakedGoodsReport();
System.out.println("_____________________________________"
);
     }
```

```
}
```

REPORT:--

```
😊😊WELCOME PIYUSH CAKE SHOPE😊😊😊 __________
......🍰🍰Detail of cake 🍰🍰.....
Cake Name: Blackforest cake
Quantity: 2
Total Price: 500.0
Description:-A refreshing lemon pound cake
_____________________________________
Cake Name: Blackforest cake
Quantity: 2
Total Price: 500.0
Description:-A refreshing lemon pound cake
_____________________________________
Cake Name: whiteforest cake
Quantity: 1
Total Price: 300.0
Description:-A refreshing lemon pound cake
_____________________________________
Cake Name: Blackforest cake
Quantity: 2
Total Price: 500.0
Description:-A refreshing lemon pound cake
_____________________________________
Cake Name: Blackforest cake
Quantity: 2
Total Price: 500.0
Description:-A refreshing lemon pound cake
_____________________________________
====Details of Ingedient Add====
Name:- Blackforest cake
Total Cost of Ingedients:310.0
_________________********_____________________
__💰💰Payment Recipt details💰💰__
____________________*****_____________________
Receipt for transaction on 3924-08-10
Amount: 250.0
Payment Type: Credit Card
_____________________________________
Receipt for transaction on 3924-08-20
Amount: 100.0
Payment Type: online
_____________________________________
Receipt for transaction on 3924-08-21
Amount: 100.0
Payment Type: Googlepay
```

```
_____________________________________
Receipt for transaction on 3924-08-22
Amount: 100.0
Payment Type: cash

_____________________________________
Receipt for transaction on 3924-08-23
Amount: 100.0
Payment Type: paytm

_____________________________________
Receipt for transaction on 3924-08-24
Amount: 100.0
Payment Type: Googlepay
_____________*******_____________
_____________________________________
===👤 👤Customer Details👤 👤===
_____________________________________
_____________________________________
Customer added successfully.
Customer details:
Name: Piyush Wadhai
Email: piyushwadhai146@.com
Phone: 9021744384

_____________________________________
Customer added successfully.
Customer details:
Name: Nandini Wadhai
Email: NandiniWadhai@.com
Phone: 9021744384

_____________________________________
_____________________________________
Customer added successfully.
Customer details:
Name: Akash Wadhai
Email: Akashwadhai146@.com
Phone: 9021744384

_____________________________________
Customer added successfully.
Customer details:
Name:  Ankush Wadhai
Email: Ankushwadhai146@.com
Phone: 9021744384

_____________________________________
_____________________________________
===New order Details===

_____________________________________
Order created: Piyush Wadhai, 3924-08-17, pending

_____________________________________
Order created: Piyush Wadhai, 3924-08-17, pending
_____________________________________
```

```
Order created: Piyush Wadhai, 3924-08-17, pending
_____________________________________
Order created: Piyush Wadhai, 3924-08-17, pending
_____________________________________
========= ORDER  STATUS =======
Order updated: Piyush Wadhai, 3924-08-17, delivered
_____________________________________
Order updated: Piyush Wadhai, 3924-08-17, delivered
_____________________________________
Order updated: Piyush Wadhai, 3924-08-17, delivered
_____________________________________
Order updated: Piyush Wadhai, 3924-08-17, delivered
_____________________________________
Order updated: Piyush Wadhai, 3924-08-17, delivered
_____________________________________
=======//ALL TOTAL REPORT//=======
Total Amount Sold: 8072.0
Number of Transactions: 17
_____________________________________
Top Selling Baked Goods: butter: 5
Top Selling Baked Goods: Bread: 10
Top Selling Baked Goods: cookie: 19
Top Selling Baked Goods: Pastries: 16
Top Selling Baked Goods: Muffins: 15
Top Selling Baked Goods: Pies: 22
Top Selling Baked Goods: Apple tart bakery: 11
Top Selling Baked Goods: Artisanal crusts: 12
_____________________________________
```

Overall, the tasks have been completed successfully, and the system is now capable of managing cake orders, inventory, payments, customers, and orders. The system can generate reports on cake orders, inventory levels, transactions, customer details, and order status.