Miniproject On

SUPERMARKET BILLING SYSTEM

Class: B.E CSE-B Semester: III

By

K.Sree Sai Chandana	1602-22-733-122
M.Deekshitha Reddy.	1602-22-733-081

Department of Computer Science & Engineering

Vasavi College of Engineering (Autonomous)
Ibrahimbagh, Hyderabad-31

2023-2024

Abstract: Supermarket Billing System in Java

The Supermarket Billing System implemented in Java provides an efficient and user-friendly solution for managing transactions in a supermarket. The system includes features such as displaying available items, adding items to the cart, managing stock quantities, generating bills, and supporting multiple payment options.

Key Components:
1. **Item and CartItem Classes:**
   - The system utilizes the `Item` class to represent supermarket items with attributes such as name, price, and stock quantity.
   - The `CartItem` class is employed to represent items added to the shopping cart, associating an item with its quantity.

2. **Menu-Driven User Interface:**
   - Users interact with the system through a simple menu-driven interface, providing options to display available items, add items to the cart, view the cart, generate bills, and exit the system.

3. **Stock Management:**
   - The program maintains an inventory of available items, displaying their names, prices, and current stock quantities. Users can add items to the cart while ensuring the availability of stock is appropriately managed.

4. **Billing and Payment Processing:**
   - The system calculates the total bill based on items in the cart, displaying a detailed breakdown of each item's contribution.
   - Multiple payment options, including cash, credit card, and digital wallets, are supported. The user selects a payment method, and the system acknowledges successful transactions.

Implementation Details:
- The code employs ArrayLists to manage available items and items in the cart, ensuring dynamic data management.
- Stock quantities are updated as users add items to the cart, preventing over-purchasing.
- The system integrates a straightforward payment processing mechanism, allowing users to choose their preferred payment option.

Conclusion:
The Supermarket Billing System provides a robust foundation for managing supermarket transactions. It balances simplicity with functionality, ensuring a seamless shopping experience for users while incorporating key features such as stock management, detailed billing, and flexible payment options. The code can be extended and customized to accommodate additional features and requirements, making it adaptable for diverse supermarket environments.

Code :

```
import java.util.ArrayList;
import java.util.Scanner;

class Item {
    String name;
    double price;
    int stock;

    public Item(String name, double price, int stock) {
        this.name = name;
        this.price = price;
        this.stock = stock;
    }
}

class CartItem {
    Item item;
    int quantity;

    public CartItem(Item item, int quantity) {
        this.item = item;
        this.quantity = quantity;
    }
}

public class Main{

    static ArrayList<Item> availableItems = new ArrayList<>();
    static ArrayList<CartItem> cart = new ArrayList<>();

    public static void main(String[] args) {
        initializeItems();

        while (true) {
            showMenu();
            int choice = getUserChoice();

            switch (choice) {
                case 1:
                    displayAvailableItems();
                    break;
                case 2:
                    addToCart();
                    break;
                case 3:
                    viewCart();
                    break;
                case 4:
                    generateBill();
                    break;
                case 5:
                    System.out.println("Thank you for shopping with us!");
                    System.exit(0);
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }

    static void initializeItems() {
        availableItems.add(new Item("Bread", 35, 50));
        availableItems.add(new Item("Milk", 25, 30));
        // Add more items with their prices and initial stock
    }

    static void showMenu() {
        System.out.println("\n*** Supermarket Billing System ***");
        System.out.println("1. Display available items");
        System.out.println("2. Add items to cart");
        System.out.println("3. View cart");
        System.out.println("4. Generate bill");
        System.out.println("5. Exit");
    }

    static int getUserChoice() {
        System.out.print("Enter your choice: ");
        Scanner scanner = new Scanner(System.in);
        return scanner.nextInt();
    }

    static void displayAvailableItems() {
        System.out.println("\nAvailable Items:");
        for (Item item : availableItems) {
            System.out.println(item.name + " - RS" + item.price + " (Stock: " + item.stock + ")");
        }
    }

    static void addToCart() {
        displayAvailableItems();

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the name of the item to add to cart: ");
        String itemName = scanner.next();

        Item selectedItem = findItemByName(itemName);
        if (selectedItem != null) {
            System.out.print("Enter the quantity: ");
            int quantity = scanner.nextInt();

            if (quantity > 0 && quantity <= selectedItem.stock) {
                CartItem cartItem = new CartItem(selectedItem, quantity);
                cart.add(cartItem);
                selectedItem.stock -= quantity;

                System.out.println(quantity + " " + itemName + "(s) added to the cart.");
            } else {
                System.out.println("Invalid quantity or insufficient stock. Please try again.");
            }
        } else {
            System.out.println("Item not found. Please enter a valid item name.");
        }
    }

    static Item findItemByName(String itemName) {
        for (Item item : availableItems) {
            if (item.name.equalsIgnoreCase(itemName)) {
                return item;
            }
        }
        return null;
    }

    static void viewCart() {
        if (cart.isEmpty()) {
            System.out.println("Your cart is empty.");
        } else {
            System.out.println("\nItems in Cart:");
            for (CartItem cartItem : cart) {
                System.out.println(cartItem.item.name + " - RS" + cartItem.item.price + " x " +
cartItem.quantity);
            }
        }
    }

    static void generateBill() {
        double totalBill = 0.0;

        if (cart.isEmpty()) {
            System.out.println("Your cart is empty. No bill generated.");
        } else {
            System.out.println("\n*** Bill ***");
            for (CartItem cartItem : cart) {
                double itemTotal = cartItem.item.price * cartItem.quantity;
                System.out.println(cartItem.item.name + " - RS" + cartItem.item.price + " x " +
cartItem.quantity
                        + " = RS" + itemTotal);
                totalBill += itemTotal;
            }
            System.out.println("Total: RS" + totalBill);

            // Allow multiple payment options (Cash, Credit Card, Digital Wallet)
            System.out.println("Choose payment option:");
            System.out.println("1. Cash");
            System.out.println("2. Credit Card");
            System.out.println("3. Digital Wallet");

            int paymentChoice = getUserChoice();
            processPayment(paymentChoice, totalBill);

            cart.clear();
        }
    }

    static void processPayment(int paymentChoice, double totalBill) {
        switch (paymentChoice) {
            case 1:
                System.out.println("Payment successful. Thank you!");
                break;
            case 2:
                System.out.println("Credit card payment successful. Thank you!");
                break;
            case 3:
                System.out.println("Digital wallet payment successful. Thank you!");
                break;
            default:
                System.out.println("Invalid payment option.");
        }
    }
}
```

Output:

```
*** Supermarket Billing System ***
1. Display available items
2. Add items to cart
3. View cart
4. Generate bill
5. Exit
Enter your choice: 1

Available Items:
Bread - RS35.0 (Stock: 50)
Milk - RS25.0 (Stock: 30)

*** Supermarket Billing System ***
1. Display available items
2. Add items to cart
3. View cart
4. Generate bill
5. Exit
Enter your choice: 2

Available Items:
Bread - RS35.0 (Stock: 50)
Milk - RS25.0 (Stock: 30)
Enter the name of the item to add to cart: Bread
Enter the quantity: 2
2 Bread(s) added to the cart.

*** Supermarket Billing System ***
1. Display available items
2. Add items to cart
3. View cart
4. Generate bill
5. Exit
Enter your choice: 2
```

```
Available Items:
Bread - RS35.0 (Stock: 48)
Milk - RS25.0 (Stock: 30)
Enter the name of the item to add to cart: Milk
Enter the quantity: 3
3 Milk(s) added to the cart.

*** Supermarket Billing System ***
1. Display available items
2. Add items to cart
3. View cart
4. Generate bill
5. Exit
Enter your choice: 3

Items in Cart:
Bread - RS35.0 x 2
Milk - RS25.0 x 3

*** Supermarket Billing System ***
1. Display available items
2. Add items to cart
3. View cart
4. Generate bill
5. Exit
Enter your choice: 4

*** Bill ***
Bread - RS35.0 x 2 = RS70.0
Milk - RS25.0 x 3 = RS75.0
Total: RS145.0
Choose payment option:
1. Cash
2. Credit Card
3. Digital Wallet
Enter your choice: 2
Credit card payment successful. Thank you!

*** Supermarket Billing System ***
1. Display available items
2. Add items to cart
3. View cart
4. Generate bill
```

```
5. Exit
Enter your choice: 5
Thank you for shopping with us!
```