

PROJECT REPORT

Danish Butt
Sadia Khan
Somal Atta Baloch

October 25, 2024

—

Visual Programming

—

Mam Aatka

Shopping Cart System Overview

Introduction

The Shopping Cart System is a sophisticated console-based application designed to enhance the online shopping experience for both consumers and retailers. In today's digital marketplace, where e-commerce is rapidly expanding, a robust shopping cart system is essential for facilitating seamless transactions. This system serves as a bridge between users and retailers, enabling a streamlined process for browsing products, managing selections, and completing purchases.

Key Components

1. Product Class

- **Properties:**
 - **Name:** The name of the product.
 - **Price:** The price of the product.
 - **Category:** The category to which the product belongs.
- **Methods:**
 - **ToString():** Returns the product details in a formatted string.

2. Cart Class

- **Properties:**
 - Private lists for item management and a reference to all possible products.
 - Discount rates and tax rates.
 - Variables to manage cart expiration.
- **Methods:**
 - **AddProduct(Product):** Adds a product to the cart and resets the expiration timer.
 - **RemoveProductByIndex(int):** Removes a product from the cart by its index.
 - **Checkout():** Computes the total, applies discounts and taxes, and clears the cart.
 - **ViewCartWithIndices():** Displays all items in the cart with their indices.
 - **CalculateTotal():** Calculates and displays the subtotal, tax, and total amounts.

- **IsExpired()**, **ResetCartTimer()**, and related expiration handling methods.
- **ProductRecommendation()**: Offers recommendations based on product price criteria.

3. Program Class (Main Class)

- Hosts the **Main()** method where the application lifecycle is controlled.
- Provides a console interface that allows users to interact with the shopping cart through various options like adding or removing products, viewing the cart, and checking out.
- Manages user interactions and input validations.

Key Features

The Shopping Cart System is equipped with several key features that greatly enhance the shopping experience for users while providing essential tools for retailers. Below are the major features of the system:

Product Categorization

The system allows for efficient product categorization, enabling users to browse through various product categories with ease. This structure helps users quickly find items of interest without having to sift through unrelated products. By organizing products into categories, the system improves navigation, making it more user-friendly. Retailers can also manage and update their product listings effortlessly, ensuring that customers always have access to the latest offerings.

Inventory Management

Real-time inventory management is a cornerstone of the Shopping Cart System. This feature allows retailers to monitor stock levels as sales occur, ensuring that users have access to up-to-date information regarding product availability. When an item is sold out, the system can automatically update its status, preventing customer frustration and enhancing satisfaction. This functionality helps retailers manage their inventory effectively, reducing the risk of overstocking or stockouts.

Payment Gateways

To facilitate secure transactions, the Shopping Cart System supports multiple payment gateways. This integration allows customers to choose from various payment methods, including credit cards, digital wallets, and other online payment solutions. By providing a range of payment options, the system caters to diverse customer preferences, increasing the likelihood of completed purchases. Additionally, the secure processing of transactions builds trust with users, as they can shop with confidence knowing their payment information is protected.

Discount Mechanisms

The Shopping Cart System incorporates a sophisticated discount mechanism that automatically applies discounts based on predefined thresholds. This feature not only incentivizes larger purchases but also enhances customer satisfaction by providing savings opportunities. Retailers can customize discount rules based on promotions or seasonal sales, allowing for flexible marketing strategies. By effectively managing discounts, the system helps retailers increase their sales while providing added value to customers.

These key features collectively contribute to a seamless and enjoyable shopping experience, ensuring that both users and retailers can navigate the e-commerce landscape with ease.

User Interaction

Users interact with the Shopping Cart System primarily through a console application, which facilitates a straightforward and engaging shopping experience. The interface is designed to be intuitive, guiding users through various functionalities such as viewing products, managing their shopping cart, and completing the checkout process.

Viewing Products

Upon launching the application, users are greeted with a welcome message and presented with a menu of options. They can select the option to view available products, which triggers the system to display a list of items, complete with their names, prices, and categories. This allows users to explore the catalog and identify products that interest them. Each product is displayed in a structured format, making it easy to navigate and select desired items.

Managing the Cart

Once users have identified products they wish to purchase, they can add items to their cart by entering the product ID and specifying the desired quantity. The application prompts users to confirm their selections, ensuring they have control over their shopping choices. Users can also view their cart at any time, which displays a summary of items, including subtotals, discounts, sales tax, and the total amount due. If they decide to change their selections, they can remove items or adjust quantities, making it easy to manage their cart effectively.

Checking Out

When users are ready to finalize their purchases, they can proceed to the checkout option presented in the menu. This step provides a detailed summary of their order, allowing users to review their selections before completing the transaction. The system calculates the total amount due, applying any relevant discounts and sales tax

automatically. Once satisfied with their order, users confirm their purchase, and the system processes their payment through the selected payment gateway. After successful payment, users receive a confirmation of their order, closing the interaction loop.

This simple yet effective user interaction model ensures that customers have a seamless experience throughout their shopping journey, enhancing satisfaction and encouraging repeat use.

Project Features

The Shopping Cart System is designed with a variety of essential features that enhance the overall user experience and streamline e-commerce operations. Below are the key functionalities that define the system:

Add and Remove Products

Users can easily add products to their shopping cart by specifying the product ID and quantity. This straightforward process allows for quick selection, enabling users to build their cart efficiently. If users change their minds, they can also remove items from the cart at any time. The system updates the cart instantly, reflecting the current selections, which helps maintain an organized shopping experience.

Item Quantity Management

The system supports item quantity management, allowing users to specify the number of units they wish to purchase for each product. This feature is particularly useful for users looking to buy multiple items of the same type. As users adjust the quantity, the system recalculates the total cost in real-time, ensuring users have accurate information on their potential expenditure.

Apply Discounts

To enhance customer satisfaction, the Shopping Cart System includes an automated discount mechanism. This feature applies discounts based on predefined purchase thresholds or promotional events. For instance, if a user's cart total exceeds a certain amount, the system automatically applies a relevant discount, encouraging larger purchases. This functionality not only benefits customers through savings but also aids retailers in driving sales during promotional periods.

Product Recommendations

The Shopping Cart System employs a recommendation engine that suggests additional products based on items already in the user's cart. By analyzing the categories and types of products selected, the system can propose complementary items that may

interest the user. This feature not only enhances the shopping experience by offering personalized suggestions but also increases sales opportunities for retailers by encouraging users to explore more products.

Cart Expiration Management

To enhance security and manage user sessions effectively, the system includes a cart expiration feature. If a cart remains inactive for 30 minutes, it expires, prompting users to restart their shopping session. This function helps protect user data and ensures that inventory remains accurate by clearing abandoned carts that could lead to confusion over product availability.

These project features work together to create a comprehensive shopping experience that meets the needs of both consumers and retailers, fostering an environment conducive to successful online transactions.

Code Structure

The Shopping Cart System is built upon several key classes that work together to deliver a cohesive and functional application. The primary classes involved are Product, CartItem, ShoppingCart, and Program. Each class plays a specific role in the system, contributing to its overall architecture and functionality.

Product Class

The Product class serves as a blueprint for individual items available for purchase in the shopping cart system. It contains essential properties such as Id, Name, Price, and Category. The Id acts as a unique identifier for each product, while Name provides a human-readable title. The Price is crucial for transaction calculations, and the Category enables effective product organization. The constructor in this class initializes these properties, and the ToString() method returns a string representation for easy display within the application.

Cart Item Class

Closely related to the Product class, the CartItem class represents a product that has been added to the shopping cart. It includes properties such as Product (an instance of the Product class) and Quantity, which specifies how many units of the product are in the cart. Similar to the Product class, it also has a constructor for initialization and a ToString() method for representation. Additionally, the GetTotalPrice() method calculates the total cost of that specific cart item based on its quantity, allowing for accurate pricing in the shopping cart.

Shopping Cart Class

The `ShoppingCart` class is central to the application, managing the collection of `CartItem` objects. It contains properties like `items`, which is a list of `CartItem`, `salesTax`, and `cartExpiration`, which tracks the cart's validity. Key methods within this class include `AddProduct()`, which adds a product or updates its quantity if it already exists in the cart, and `RemoveProduct()`, which removes a specified quantity of a product. The class also features calculation methods such as `GetSubtotal()`, `GetDiscountAmount()`, and `GetTotal()`, which facilitate the overall pricing logic of the shopping experience.

Program Class

Finally, the `Program` class serves as the entry point for the application. It contains the `Main` method, which initializes instances of both `ShoppingCart` and `Product`. This class orchestrates the interaction with users, displaying a menu that allows them to view products, manage their cart, and proceed to checkout. It also includes several helper methods to facilitate user actions, such as `ViewProducts()` and `AddToCart()`, ensuring that the application runs smoothly and effectively responds to user inputs.

Together, these classes form a structured and modular design, making the Shopping Cart System not only functional but also easy to maintain and extend in the future.

Workflow Overview

The Shopping Cart System is designed to provide users with a seamless and intuitive workflow from the moment they enter the application to the completion of their purchase. Below is a detailed description of the typical steps a user would follow during their shopping journey.

Step 1: Browsing Products

When users launch the application, they are greeted with a welcome message and presented with a menu of options. The first step in their workflow is to browse available products. Users can select the option to view products, which prompts the system to display a categorized list of items, complete with their names, prices, and categories. This organized layout allows users to easily navigate through various product offerings and identify items of interest.

Step 2: Adding Products to the Cart

Once users find products they wish to purchase, they can add them to their shopping cart. This is done by entering the product ID and specifying the desired quantity. The system confirms their selections, ensuring that users have control over their choices.

Users can add multiple items at once, allowing for efficient cart management as they shop.

Step 3: Managing the Shopping Cart

After adding products, users can view their shopping cart at any time. This feature displays a summary of all items currently in the cart, along with subtotals, applied discounts, sales tax, and the total amount due. Users can adjust the quantities or remove items as needed, providing flexibility to refine their selections before checkout.

Step 4: Proceeding to Checkout

When users are satisfied with their cart contents, they can proceed to the checkout process. Selecting the checkout option presents them with a detailed summary of their order, allowing for a final review. The system automatically calculates the total amount due, applying any applicable discounts and sales tax. Satisfied with the summary, users confirm their purchase.

Step 5: Completing the Purchase

Upon confirmation, the system processes the payment through the selected payment gateway. After successful payment, users receive a confirmation of their order, completing the transaction. This structured workflow not only enhances the shopping experience but also encourages users to return for future purchases, as the process is designed to be both efficient and user-friendly.

Output

Worship

Welcome!

```
Please enter your name: danish
```

Hello, danish! Let's start shopping.

```
Press any key to continue...
```

```

*****
*****
*****
*****  SHOPPING CART SYSTEM  *****
*****
*****
*****
*****
*****

```

1. Add Product
2. Remove Product
3. View Cart
4. Checkout
5. Product Recommendation
6. Exit

Select an option: 1

```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****
```

Select a category:

1. Mobile Phones
2. Mobile Phone Accessories
3. Laptops
4. Computer Accessories
5. Smart Watches
6. Back
7. Exit

Select a category: 1

```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****
```

Choose a mobile phone brand:

1. Samsung
2. Apple
3. Xiaomi
4. Huawei
5. OnePlus
6. Back
7. Exit

Select a brand:

```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****
```

Choose a mobile phone brand:

1. Samsung
2. Apple
3. Xiaomi
4. Huawei
5. OnePlus
6. Back
7. Exit

Select a brand: 1

Select a product to add to your cart:

1. Samsung Galaxy S24 Ultra - \$1299 (Mobile Phones)
2. Samsung Galaxy S24 - \$1099 (Mobile Phones)
3. Samsung Galaxy Z Fold 6 - \$1999 (Mobile Phones)
4. Samsung Galaxy Z Flip 6 - \$1299 (Mobile Phones)
5. Samsung Galaxy A74 - \$499 (Mobile Phones)
6. Samsung Galaxy A54 - \$399 (Mobile Phones)
7. Samsung Galaxy S23 - \$999 (Mobile Phones)
8. Samsung Galaxy S22 - \$799 (Mobile Phones)
9. Samsung Galaxy S21 - \$699 (Mobile Phones)
10. Samsung Galaxy S20 - \$599 (Mobile Phones)
11. Back
12. Exit

Select a product: 2

Enter the quantity (1-5): 3

Samsung Galaxy S24 added to the cart.

Samsung Galaxy S24 added to the cart.

Samsung Galaxy S24 added to the cart.

Successfully added 3 unit(s) of Samsung Galaxy S24 to your cart.

You have 10 minute(s) and 0 second(s) until your cart expires.

Press any key to continue...

```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****
```

```
1. Add Product
2. Remove Product
3. View Cart
4. Checkout
5. Product Recommendation
6. Exit
Select an option: 2
```

```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****
```

```
1. Add Product
2. Remove Product
3. View Cart
4. Checkout
5. Product Recommendation
6. Exit
Select an option: 2
Choose an item to remove:
1. Samsung Galaxy S24 - $1099 (Mobile Phones)
2. Samsung Galaxy S24 - $1099 (Mobile Phones)
3. Samsung Galaxy S24 - $1099 (Mobile Phones)
You have 9 minute(s) and 19 second(s) until your cart expires.
Enter the number of the item to remove: 2
Samsung Galaxy S24 removed from the cart.
Press any key to continue...
```

```
*****
*****
*****  SHOPPING  CART  SYSTEM  *****
*****
*****

1. Samsung Galaxy S24 - $1099 (Mobile Phones)
2. Samsung Galaxy S24 - $1099 (Mobile Phones)
You have 8 minute(s) and 45 second(s) until your cart expires.
Press any key to continue...
```

```
*****
*****
*****  SHOPPING  CART  SYSTEM  *****
*****
*****

1. Add Product
2. Remove Product
3. View Cart
4. Checkout
5. Product Recommendation
6. Exit
Select an option: 4
```

```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****
```

Checkout process:

Current Date and Time: 10/25/2024 8:53:38 PM

Subtotal: \$2198

Discount (10%): -\$219.80

Sales Tax (8%): +\$158.2560

Total: \$2136.4560

Proceeding to checkout...

Thank you for your purchase!

Press any key to continue...

Would you like to continue shopping? (y/n): y

```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****
```

Recommended Products with a special discount:

- Mechanical Keyboard - Original Price: \$120, Discounted Price: \$108.00
- Apple Watch Series 7 - Original Price: \$399, Discounted Price: \$359.10

Press any key to continue...


```
*****
*****
***** SHOPPING CART SYSTEM *****
*****
*****

Select a category:
1. Mobile Phones
2. Mobile Phone Accessories
3. Laptops
4. Computer Accessories
5. Smart Watches
6. Back
7. Exit
Select a category: 7
Exiting the shopping cart. Goodbye!
```

Data Management

Data management is a crucial aspect of the Shopping Cart System, influencing how information is stored, accessed, and manipulated within the application. The application utilizes two primary collections to manage data: a list of available products and a list of cart items. These collections ensure efficient handling of product information and user selections, enabling a smooth shopping experience.

Available Products

The list that holds available products is implemented as a `List<Product>` collection. Each `Product` object within this list contains essential attributes, including `Id`, `Name`, `Price`, and `Category`. This structure allows the application to maintain a catalog of products that users can browse through. The `Product` class serves as a blueprint, encapsulating all necessary properties and behaviors associated with the products. This modular design facilitates easy updates and management of product information, ensuring that retailers can quickly add new products or modify existing ones. When the user requests to view products, the application iterates through this list, presenting the information in a user-friendly format.

Cart Items

On the other hand, the shopping cart is managed using a `List<CartItem>`, where each `CartItem` represents a product added to the cart along with its quantity. This list allows users to keep track of their selected items and perform various operations, such as modifying quantities or removing items. The `CartItem` class links directly to the `Product` class, providing a reference to the product being purchased while also storing the quantity of that product. This setup ensures that updates to the cart are reflected in real-time, allowing for an accurate representation of the user's selections.

Data Operations

The `ShoppingCart` class orchestrates operations on these collections, including adding products, removing items, and calculating totals. Methods within this class, such as `AddProduct()` and `RemoveProduct()`, manipulate the `List<CartItem>` to accurately reflect the user's choices. Furthermore, the application includes functionality to handle discounts and apply sales tax, which are calculated based on the contents of the cart.

By utilizing these structured collections and maintaining clear relationships between products and cart items, the Shopping Cart System effectively manages data, ensuring a seamless interaction for users as they navigate through their shopping experience.

Conclusion

The C# Shopping Cart Application provides a robust and interactive framework for simulating an e-commerce shopping experience. It features a detailed implementation of product and cart management functionalities that can easily be extended or modified for educational or demonstrational purposes. This makes it an invaluable tool for understanding the dynamics of online shopping systems in a controlled, scalable environment.

In summary, this project serves as an effective demonstration of object-oriented programming principles, showcasing how a well-architected system can deliver a robust and user-friendly e-commerce solution.