

**Project Report**

**Visual Programming C#**

**Project Members:**

Abdul Moiz (233507)

Abdul Rehman (233589)

Kinza (233561)

**Submitted To:**

Professor Atka Ali

**Project Title:**

Shopping Cart using C#

**Requirements:**

* Add and remove products from the cart.
* View Cart.
* Item Quantity Management
* Calculate the total cost of items in the cart.
* Apply discounts and sales tax.
* Product Recommendations
* Cart Expiration
* Checkout Process

**Introduction**

This project involves designing and implementing a **Shopping Cart System** in C#. The application simulates the functionality of a shopping cart that allows users to browse products, add and remove items, view the cart, and proceed to checkout. The system utilizes basic data management, timer functionalities for cart expiration, and a simple command-line interface to enhance user interaction.

**Overview:**

The objective of this project is to create a user-friendly shopping cart that supports:

* Adding, viewing, and removing products in the cart.
* Persisting cart data between sessions.
* Calculating a final total with sales tax and discount at checkout.
* Automating cart expiration to ensure cart data resets after a specified period.

**Classes and their Roles:**

1. **Product Class**

* Represents an individual product with the following properties:
  + Id: Product ID (int)
  + Name: Product name (string)
  + Price: Product price (decimal)
  + Quantity: Number of units (int)
* Constructor initializes these properties and assigns default quantity to 1 if not provided.

1. **User Class**

* Represents a user with:
  + Id: Unique identifier for the user (string)
  + Name: User's name (string)

1. **Shopping Cart Class**

* Manages the cart's contents and actions related to the cart.
* **Attributes:**
  + \_products: List that stores Product objects.
  + DataFile: Constant string representing the file where cart data is stored (cart.txt).
  + \_cartExpiryTimer: Timer object that triggers cart expiration after a specific time (120 seconds).

**Methods:**

* **AddProduct()**: Adds a product to the cart or updates its quantity if it already exists.
* **RemoveProduct():** Removes a product from the cart by its ID.
* **ViewCart():** Displays the current items in the cart.
* **CalculateTotal():** Computes the total cost of items, applies taxes and a 5% discount, and handles checkout.
* **SaveCheckoutToFile():** Saves the checkout details, including the total bill and products purchased, to a file.
* **LoadCart():** Loads the cart data from a file.
* **SaveCart():** Saves the current cart state to a file.
* **InitializeCartExpiryTimer():** Initializes the timer for automatic cart expiry.
* **CartExpiryTimer\_Elapsed():** Handles the event when the cart expiry timer elapses, clearing the cart.
* **DisposeCart():** Clears the cart and resets the data file.

**Timer**:

* A timer for cart expiration is implemented using System.Timers.Timer.
* The timer automatically clears the cart after 160 seconds, notifying the user of cart disposal.

1. **Program Class**

* Contains the main loop for user interaction.
* The user is prompted to add, remove, view, and checkout from the cart using numbered options.

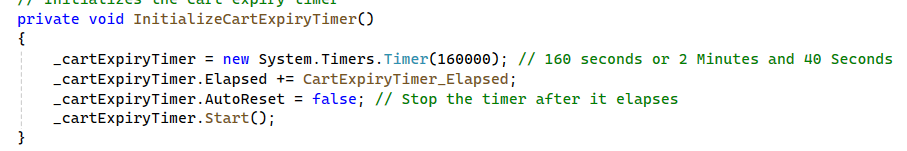
**Workflow**

The **Shopping Cart System** workflow follows these main steps:

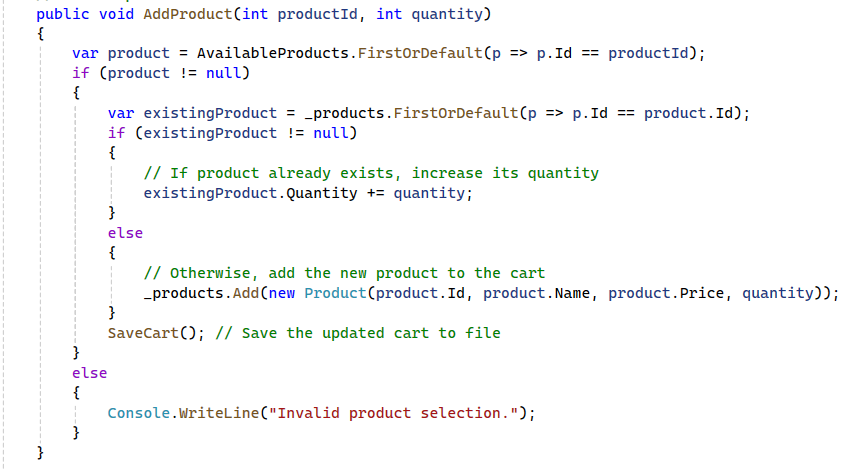
1. **Initializing the Cart**:
   * A list of available products is loaded and stored in memory.
   * Cart data is loaded from a file (cart.txt) if it exists, restoring any saved cart state.
   * A timer is set to expire the cart if it's inactive for a defined period (2 minutes and 40 seconds in this case).
2. **Main Menu**:
   * A menu is displayed to the user, with options to add products, view the cart, remove products, proceed to checkout, or exit the application.
3. **Adding Products**:
   * Users choose from a list of available products by entering a product number and specifying a quantity.
   * If the product is already in the cart, its quantity is updated; if not, it's added to the cart.
4. **Viewing and Managing Cart**:
   * The user can view current cart contents, including each product's ID, name, price, and quantity.
   * Products can be removed from the cart using their unique IDs.
5. **Checkout Process**:
   * Calculates the subtotal, adds an 18% sales tax, applies a 5% discount, and calculates the final total.
   * User details are collected, and a unique checkout file is created to store the purchase record.
   * Cart is cleared after a successful checkout.
6. **Cart Expiry**:
   * After a specified time of inactivity, the cart expires, automatically clearing its contents.
7. **Exit**:
   * Disposes of any cart data and closes the application.

**Key Code Elements**

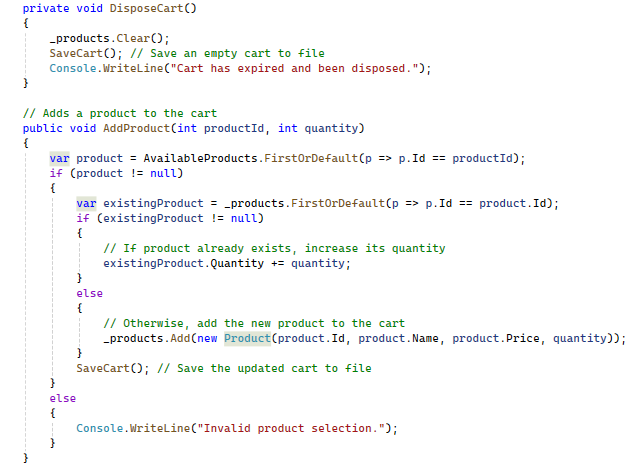
* **Cart Expiry Timer**



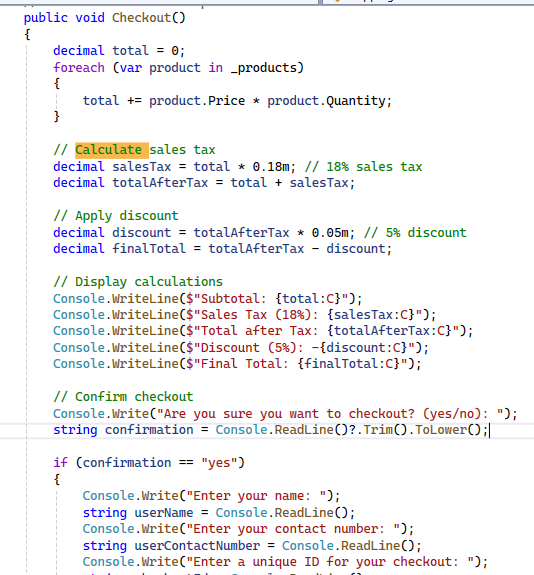
* **Add Product to Cart:**

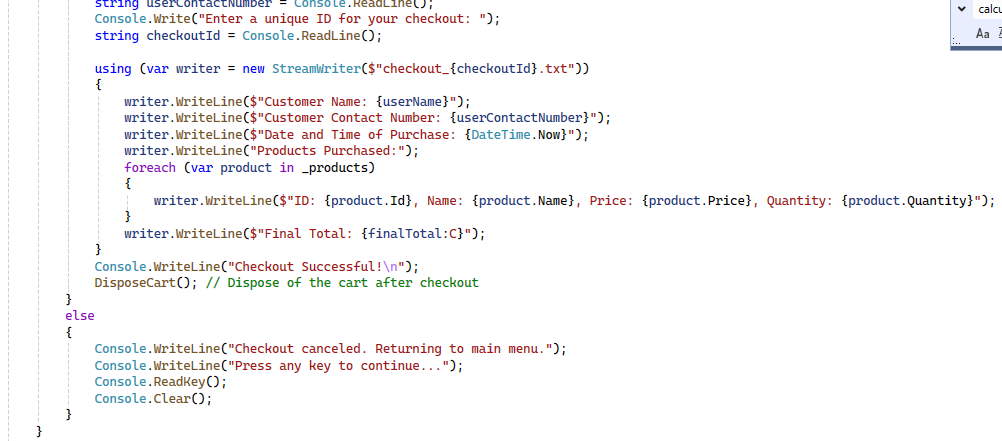


* **Dispose cart:**

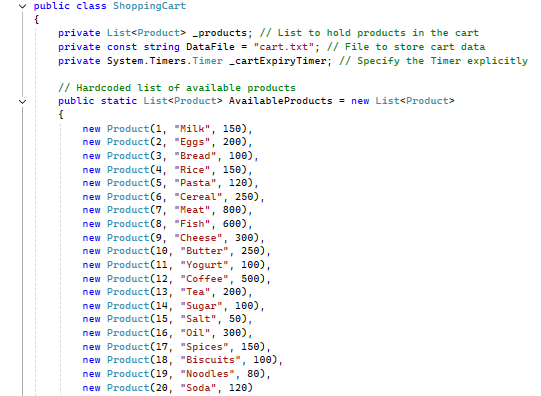


* **checkout:**





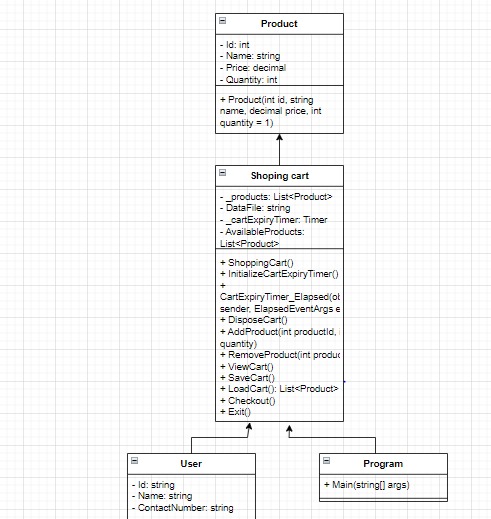
* **shopping cart**



**Conclusion**

This C# Shopping Cart System provides a complete, simple, and functional experience for users simulating an online shopping cart with basic functionalities. The use of file I/O for persistence, along with a timer to manage cart expiry, makes this application a useful tool to understand core C# functionalities like file handling, timers, and list manipulation within a structured, object-oriented framework.

**UML DIAGRAM:**



**FLOW CHART:**

