COE528 Project Report

Group Information

- Group Members: Tommy, Catalin, Leo, Brendan

  Group number : 60

Main Flow:

1. The owner clicks on the [Books] button in the owner-start-screen.
2. The application shows the owner-books-screen that has a table of books and form fields to create a new book.
3. The owner provides the book title and price through the corresponding text boxes.
4. The owner clicks on the [Add] button.
5. The form validates the input and adds the new book to the table.
6. The table was shown on the screen.
7. The internal books' in-memory data structure.

Postconditions:

- The new book forms part of the inventory of the bookstore.
- If the program is terminated, the list of new books is saved to books.txt.

Why Did We Use State Design Pattern

The State Design Pattern was used in this project for changing customer status (Silver or Gold) dynamically depending on their points earned. This design pattern allows for nice separation of behavior for each state and easy extension or modification of functions later on.

How It Works:

1. Each customer has a CustomerState object associated with them that stores their current status (SilverStatus or GoldStatus).
2. If a customer's points changes (e.g., following an order), their status automatically changes based on the following:
   - Points < 1000 → SilverStatus
   - Points ≥ 1000 → GoldStatus
3. Each state defines some action:
   - Silver clients earn points at a standard rate (10 points for each CAD spent).
   - Gold customers have benefits such as points redemption for discounts

Benefits of Using State Design Pattern:

- Customer behaviors are encapsulated within state classes, reducing the complexity of the main Customer class.
- Adding new states or modifying states that already exist will not affect other parts of the code.
- The design is simple in adding new customer statuses or reward programs in later versions.

Implementation Overview

Class Design

The system was designed using object-oriented principles and includes several key classes:

1. Bookstore: Manages books and customers using ArrayList<Book> and ArrayList<Customer>.
2. Book: Represents individual books with attributes name and price.
3. Customer: Represents registered customers with attributes username, password, points, and their current state (CustomerState).
4. CustomerState: Abstract base class for customer statuses (SilverStatus, GoldStatus).
5. SilverStatus: Implements behavior for customers with fewer than 1000 points.
6. GoldStatus: Implements behavior for customers with 1000 or more points.

Key Features Implemented:

Owner functionalities:

- Add/Delete books.
- Add/Delete customers.

Customer functionalities:

- View available books.
- Purchase books with options to redeem points.
- Accumulate points based on purchases.

To access the program, the owner’s username is “admin”, and the password is “admin”.