Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	COE
Course Number:	528
Semester/Year (e.g.F2016)	W2022

Instructor:	Olivia Das

Assignment/Lab Number:	5 (final)
Assignment/Lab Title:	Bookstore Project

Submission Date :	03 April 2022
Due Date:	03 April 2022

Student LAST Name	**Student FIRST Name**	**Student Number**	**Section**	**Signature***
Thomas	Abel		6	AT
Islam	Raiyan		6	Raiyan.I
Nazimuddin	Raidah		4	RN
Sharma	Karishab		6	KS

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/senate/current/pol60.pdf

Use case report

Use case name: BookstoreApp (to put it as a verb it can be called: Buying books)

Participating actor: Owner and the Customer

The flow of events:

1. Owner logs in which the program opens the owner start screen where the owner can choose to go to Customer or books screens.
   - 1.1. The owner can log out using the [logout] button at the start screen which sends the owner back to the login screen.
   - 1.2. The owner can add and delete customers or books on their respective screens after the start screen.
   - 1.3. Once the owner enters the book and the customer screens, they can go back to the start screen with the [back] button.
2. Customer logs in which the program opens the customer start screen where the customer can buy the books available.
   - 2.1. Customers can check the books wanted to buy and can checkout where the total cost, points, and status are shown.
   - 2.2. The customer can log out using the [logout] button at the customer start screen which sends the customer back to the login screen.
   - 2.3. Customers can either ‘buy’ it without using points (10CAD spent = 100points added) or use the points accumulated to ‘redeem and buy’ (10CAD reduced = 1000points deducted).
   - 2.4. At checkout, the customer is updated with their membership status (gold or silver) depending on their points accumulated or after the points that are redeemed after purchase.
3. Pressing ‘X’ or the window closing button saves the book and customer data for the next run.

Entry condition: The customer must be a part of the bookstore. The owner and customer must be logged in to access their respective screens.

Exit condition: The owner or customer is logged out of their respective screens.

Exceptions:

1. An incorrect username or password gives an error.
2. Blank text will give you the same error at login.
3. Incorrect data type or retrieving empty data will give an error.

Special Requirements: Since the project didn’t mention the number of copies of a book, it is assumed that only one copy of a book exists. This means that the book is deleted upon purchase by a customer.

The rationale behind using the State Design Pattern:

A state design pattern was used when a class was dependent on a state of an object while the program runs. It must be noted that this will only affect the object using the state and not the state itself changes. The behavior of the class is changed depending on how the user uses the program. This state pattern is seen in:

1. Customer changes state between gold and silver depending on the accumulated points.

2. Owner and customer have different states when they log in as the owner have more access than the customers.
3. Owner has three states: books, customers, and logout. Then the books and customers go to their corresponding lists. Logout goes back to the initial state to log in.
4. Customers also have three states: buy, redeem, and buy, and logout. In both ‘buy’, and ‘buy and redeem’, another screen is shown with the total cost and updated membership status, another state which is mentioned in point 1.

Ultimately, because of all these states within the program and how it is closely related to the state pattern, thus state design pattern was used for this program.