

Course Title:	Object Oriented Eng Analysis and Design
Course Number:	COE528
Semester/Year (e.g.F2016)	W2022

Instructor:	Olivia Das
--------------------	------------

Assignment/Lab Number:	Final Project
Assignment/Lab Title:	BookStore Application

Submission Date:	March 3, 2022
Due Date:	March 3, 2022

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Cheema	Iman	501039614	09	Iman Cheema
Sivajilingam	Sanjith	501030692	09	Sanjith Sivajilingam
Patel	Vasu	501032020	04	Vasu Patel
Shah	Chintan	501028779	04	Chintan Shah

*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>

Describe a use case following the format discussed in class.

1. Name of Use Case: AddBooks

2. Participating Actors: Owner

3. Flow of Events:

1. The owner logs in with the correct credentials.
2. Owner selected [Books] on *owner-screen*.
3. The owner can add a book by inputting the book name and price, followed by clicking the [Add] button.

4. Entry Condition: The owner must login by entering their username and password and clicking the [Books] button to enter the screen.

5. Exit Conditions: The owner can exit the system when they press the [Back] button and a book is created or 'X' is clicked on top-right of the screen.

6. Exceptions: If the owner inputs a non number price in the 'Price' text field, then an alert message box that says 'Invalid Input' will appear. If the owner does not enter anything for the book name or book price, an alert message box that says 'Invalid Input' will appear. If the owner enters a book name and price that already exists in the table, an alert message box that says 'Book already exists in the database' will appear.

7. Special Requests: It is also assumed that the owner has only one copy of a book.

8. Quality Requirements: When an owner adds a book, this use case can include validateInput use case. The use case is called to check for valid user input.

Describe the rationale behind using the State Design Pattern

The rationale behind using the State Design Pattern is to be able to easily scale and maintain our code. If we needed to add a different customer status, such as Diamond or Bronze, not much would need to change. We would just need to update the checkStatus method in customer class and create a DiamondCustomer or BronzeCustomer class which extends CustomerStatus. This way our code is organized and can be easily modified to add additional features.

Additional Features

- Alert Dialog Box:
 - On *login-screen* if the username and password are not valid, an alert dialog message shows up indicating that the username or password is incorrect.
 - *owner-books-screen*, if the username and password inputs are not valid or the username already exists, an alert dialog message shows up indicating that the username or password is incorrect.
 - *owner-customers-screen*, if book name, and price inputs are not valid or book name already exists, an alert dialog message shows up indicating that username or password is incorrect.