Faculty of Engineering and Architectural Science
Department of Electrical and Computer Engineering
Laboratory Report Cover Page

Course Number	COE528
Course Title	Object Oriented Analysis Design
Semester/Year	Winter 2023
Instructor	Olivia Das
TA Name	Jeanne Alcantara

Lab/Tutorial Report No.	Final Project

Report Title	Bookstore Application Project

Section No.	01
Group No.	03
Submission Date	April 2nd, 2023
Due Date	April 2nd, 2023

Student Name	Student ID	Signature
Victor Do	501137174	V.D.
Anmol Panchal	501107696	A.P.
Deep Patel	501090050	D.P.
Yanny Patel	501102555	Y.P
Vlad Scraba	501102382	V.S

Describe a use-case following the format discussed in class

Name: Login

Actors: Customers, Owner

Entry Conditions: The program is opened and the user selects the login button.

Flow of Events:

1. The user presses the login button to be taken to a login screen with a username and password text field.
2. The user enters their username and password into their respective text fields and presses the login button.
3. The program checks if there is a matching username and password in the AccountManager's data which is loaded from save files when the program starts.
4. If there is a match, the user is logged in and directed to a homepage corresponding to their user type (Customer or Owner).
5. If there is not a match, an error message is displayed to the user ("Incorrect Username or Password. Please Try Again").

Exit Conditions: The user is either logged in or remains on the login screen. If the back button is pressed on the login page then the user will be taken back to the start screen where they have the options to quit the program, sign up as a new user, or return to the login page.

Exceptions: If the X button is pressed then the program will save all the existing user and book data to the local files before closing.

Special Requirements:

There is no limit to the number of login attempts.

There are no additional security measures such as two-factor authentication.

There are no password complexity or expiration requirements.

If the user forgets their password, there is no password reset option available, only the owner can manage account details.

There is no logging or tracking of login attempts for security purposes.

There is no potential for multiple users to have the same username.

If the user enters incorrect login information, an error message is displayed ("Incorrect Username or Password. Please Try Again").

There is no way for the user to change their password once logged in.

Describe the rationale behind using the State Design Pattern

The State Design Pattern is a behavioral design pattern that allows an object to change its behavior when its internal state changes. In our login manager class, we have used this pattern to dynamically change the GUI class based on the type of user that has logged in.

When a user logs in, the login manager class searches the database for a user with a matching username and password. If a matching user is found, the login manager creates an instance of the appropriate GUI class using a factory, which is then stored as a state variable. By using this approach, we are able to easily add new user types in the future without modifying the existing code.

One of the advantages of using the State Design Pattern in this context is that it makes our code more modular and maintainable. We are able to centralize the logic for selecting the appropriate GUI class in a single location, which reduces duplicated code and improves the organization of our program. Furthermore, this approach adheres to the SOLID design principle of Open/Closed, which means that our code is open for extension but closed for modification.

In conclusion, the State Design Pattern is an effective approach for implementing a login manager in a JavaFX application. By allowing us to dynamically change the behavior of our program based on its internal state, we are able to create code that is modular, maintainable, and extensible.