Use-Case Description

1.	Use Case Name	UpdateCustomerInfo
2.	Participating Actors	Initiated by Owner
3.	Flow of Events	3.1 The owner activates “Customer” function on the Owner Start Screen. 3.2 The owner may register a customer by typing the username and password for a specific customer in the provided textboxes and clicking the **[Add]** button. Then the added customer’s username and password are visible in the table. 3.3 The owner may delete a customer by clicking on the specific line which mentions the customer in the table. Then click on the **[Delete]** button. 3.4 The owner may go back to the previous screen by clicking the **[Back]** button. 3.5 The owner may click the [**minimize], [maximize] and [cancel]** buttons on the screen to initiate these functionalities of the application.
4.	Entry Condition	This use case starts only when the owner has logged in with the correct username: “admin” and password: “admin”
5.	Exit Condition	This use case terminates when the owner either decides to click the **[Back]** button or the **[cancel]** button on the top right corner of the screen.
6.	Exceptions	If customer information is not successfully read or written to the appropriate text files, then there will be appropriate exceptions thrown.
7.	Special Requirements	It is assumed that there is only one customer with a specific set of username and password.

Rational Behind Using the State Design Pattern

While designing and implementing the bookstore application, the state design pattern was used. The state design pattern is used when the behaviour of a class or method is variant on a state. The change occurs at runtime. Also, when states are changed, the current class is not affected. This implies that the state object changes the state, but the current state or class doesn't change itself.

The current application has two states: Owner and Customer. The owner state has different functionalities compared to the customer state. The User class is the 'client', the GUI is the 'context', and there are multiple sub-states of the two main states.

The first state is the Owner which has three sub-states: Books, Customers, and Logout. The list of available books is displayed when Books is clicked. The list of customers is displayed when Customers is clicked. The original login screen is displayed when Logout is clicked.

Similarly, the second state is Customers which has three sub-states: Buy, Redeem and Buy, and Logout. The buy state directs the customer to the final window displaying the total cost and status. The redeem and buy state again directs the customer to the final window displaying the total cost and status. The logout state logs the customer out of the application and directs to the login screen.

The status also implements the state design pattern where the states are: Gold and Silver depending on the points collected by the customer.