Use Case

Name	Handle Customers
Participating Actors	Owner
Flow of Events	1. The Owner logins into the Bookstore application 2. The Owner opens the Customer Database screen from the owner start screen 3. The Owner inputs a new Username and Password for the Customer and adds them into the database. 4. The Owner selects certain Users on the table and deletes them off of the database one by one. 5. The Owner exits the Customer database screen with the back button into the owner start screen and logs out.
Entry Conditions	The Owner is logged into the Bookstore application with their correct username and password (admin).
Exit Conditions	● The Owner has added a new and unique customer account into the database ● The Owner has deleted an existing customer account from the database
Exceptions	The Owner is notified immediately if the account being added already exists in the database
Special Requirements	Once the Owner activates the “Handle Customers” from his or her terminal, the Owner is not allowed to logout.

State Design Pattern

The screen of our Bookstore application relies on a State Design Pattern, as this pattern allows our application to transition from one screen to another in a single window. Although monolithic if and switch statements can also be implemented towards screen transition of our application, State Design Pattern is much preferable since a new type of screen can be added easily by defining subclasses - rather than creating multiple of conditional logic in a single method which is notoriously difficult to manage.

There are 10 classes that are participating in the State Design Pattern including CurrentScreen, ScreenType, LogInScreen, OwnerScreen, BooksScreen, CustomerScreen, OwnerStartScreen, CustomerScreen, CustomerStartScreen, and CostScreen. The "context" class, CurrentScreen, initially holds the concrete state class, LogInScreen; and it transitions into a new state based on the condition or input received from the user. The transitions, conditions, and activities of these classes are revealed in the diagram, Figure 1.

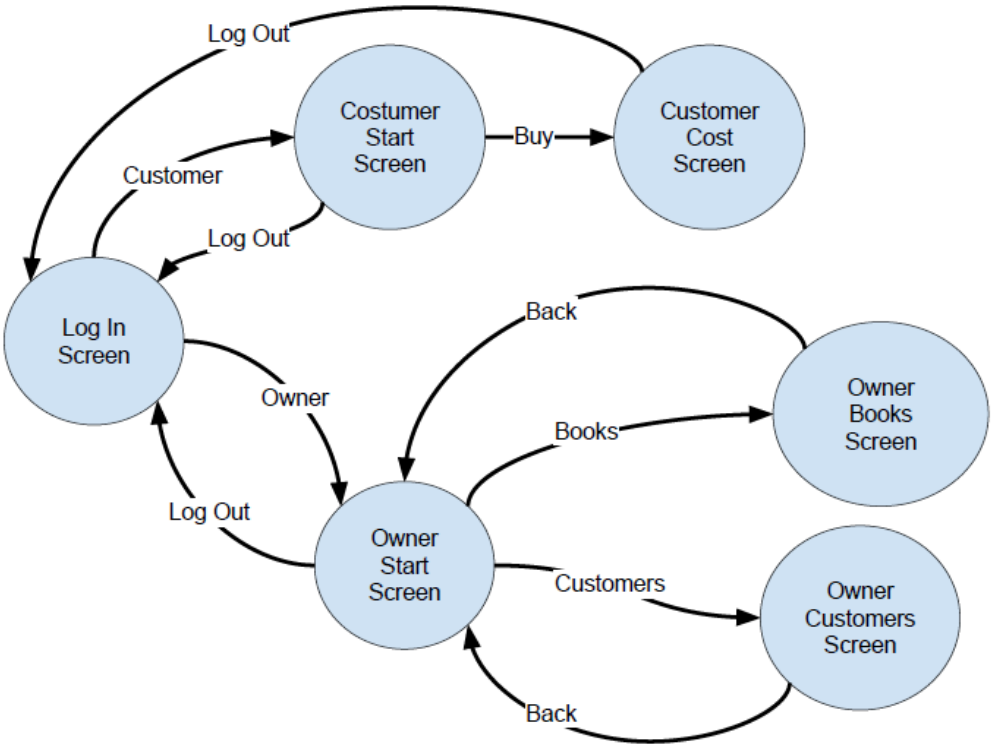

Figure 1. A state diagram describing the behavior of the implemented State Design Pattern.