




<b>Course Title:</b>	Object Oriented Analysis and Design
<b>Course Number:</b>	COE 528
<b>Semester/Year (e.g.F2016)</b>	W2021

<b>Instructor:</b>	DR. Olivia Das
--------------------	----------------

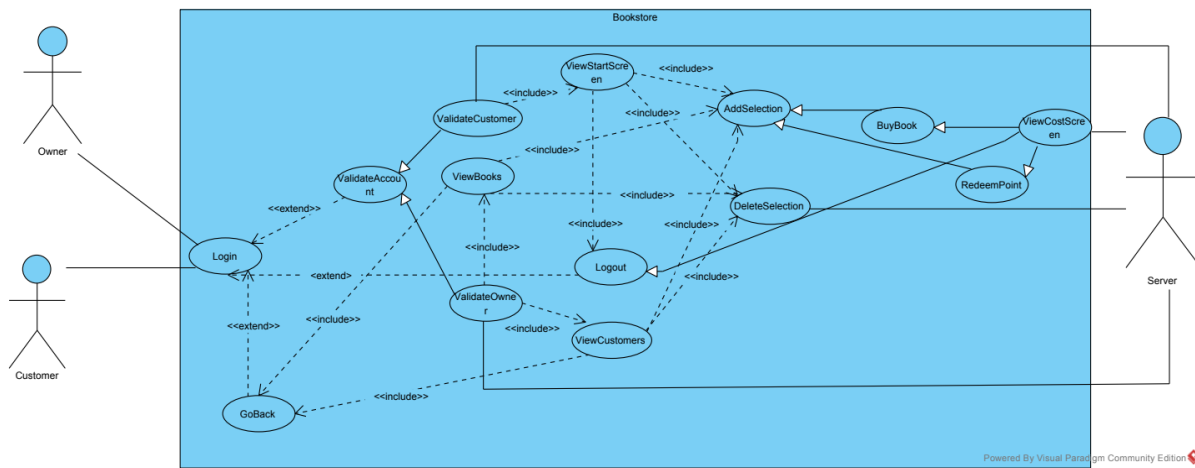
<i>Assignment/Lab Number:</i>	Final Project
<i>Assignment/Lab Title:</i>	Bookstore Application

<i>Submission Date:</i>	Sunday, April 4th, 2021
<i>Due Date:</i>	Sunday, April 4th, 2021

<b>Student LAST Name</b>	<b>Student FIRST Name</b>	<b>Student Number</b>	<b>Section</b>	<b>Signature*</b>
Damas	Miguel	500840736	09	MD
Faisal	Mahir	500896206	09	MF
Ibrahim	Abdul	500984022	09	AI
Samaan	Mena	500911425	09	

\*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>

## UML Use-Case Diagram Description



The functionality of the bookstore application is as follows: when the program is run as is, there is a login screen. In order to use the program, the user must either log in with the admin or with a customer account with the appropriate credentials. For the administrator when he or she logs in, they will be greeted with a screen that has three buttons which consist of books, customers, and logout. When the books option is selected the user is presented with a new screen that displays the current books and prices saved in the text file. The user then has the option to add a new book along with its price as well as delete any books they wish. Once they are finished, they may click the back button and return to the previous screen. When the admin selects the customers option, they are greeted with a page that shows all the active customers with their respective usernames and passwords that are again saved in a text file. The user then has the option to add a new customer with the use of their login credentials or delete an existing customer through item selection on the table. Once finished, the user may then select back as well, and return to the previous window. The final option that is left is the logout button which brings the user back to the login page. Now if the user decided to login as a customer they must use an existing account along with its respected password. If the user does not do this correctly, the program will give an error message saying that either the username or the password entered is incorrect. Once this is done correctly, the user will be met with a page that displays their username and how many points they have they also will see the books that are available for purchase and have the option to buy said books using money or their stockpiled points once either of these actions is completed they will promptly be brought to a new screen that displays their purchased item, total cost and the points redeemed. They will then be asked to logout and then return to the main screen.

Use case 1

**Name**

Login

**Participating actors**

Owner

Customer

Server

**Entry Conditions**

In order for anyone to access the application they must have either a valid administrator account or a valid customer account.

**Flow of Events**

1. User enters their login credentials for an administrator account or a customer account.
2. The server process the data entered by the user and logs them in

**Exit Conditions**

Once the user is finished using the application while logged in they can click the logout button to exit.

**Exceptions**

If the user enters the wrong login information, they will be met with an error message.

**Special Requirements**

Account must exist in the server's database (found in text file)

## Use case 2

### **Name**

ViewCustomers

### **Participating actors**

Owner, Server

### **Entry Conditions**

The user is logged in as an administrator

### **Flow of Events**

1. The user logs into the application with an administrator account
2. The admin clicks on the Customers button, which is located on his start screen
3. The admin is directed to a page with a table of the customer users who are registered into his or her application.
4. The server processes the start page when the admin logs in as well as the list of customers when the admin clicks the customers button located on said start page.

### **Exit Conditions**

In order to exit this use case, the administrator must click on the back button to return their start screen. Then the admin user should click the logout button to return to the login page and exit the use case.

### **Exceptions**

Not Applicable

### **Special Requirements**

The user must be logged in as an administrator. If the user is logged in as a customer, they will not be able to access this use case.

### Use case 3

#### **Name**

Addselection (Refers to the addition of a book or customer made by the owner)

#### **Participating actors**

Owner, Server

#### **Entry Conditions**

The user is logged in as the owner.

#### **Flow of Events**

If logged in as the Owner:

Case 1:

1. The owner selects the books button, he/she is directed to a page with a table that contains a list of books with the respective prices.
2. To select a book to be added, the owner must enter the name of the book followed by the price of the book in the appropriately labeled "Name" and "Price" fields respectively
3. To add a selected book, the owner simply has to click on the add button that is located next to the two text fields.

Case 2:

1. The owner selects the customers button, he/she is directed to a page with a table that contains a list of customer usernames each with their respective passwords.
2. To select a customer to be added, the owner must enter the username of the customer followed by the password of the customer in the appropriately labeled "Username" and "Password" fields respectively.
3. To add a selected customer, the owner simply has to click on the add button that is located next to the two text fields.

All button clicks, page switches, and item additions are processed through backend methods by the server.

#### **Exit Conditions**

To exit this use case, the owner simply needs to click the back button to return to the start screen. By doing so, the information in the text fields that are used to add a selected book or customer are cleared. The owner may also manually clear the information entered in the text fields for a given page by pressing the backspace key.

### **Exceptions**

If the owner visits the books page and they would like to add a selected book of their choice, he or she must ensure that the selected price data is of type double. If not, the page will give the owner an error message stating that the selected price is invalid.

### **Special Requirements**

The user must be logged in as the Owner. A customer does not have the ability to add or delete any selected items to the bookstore. They are able to buy books from the bookstore but this principle is significantly different from adding or deleting books and customer data from the bookstore application's database.

## Use case 4

### **Name**

Deleteselection

### **Participating actors**

Owner, Server

### **Entry Conditions**

The user is logged in as the owner.

### **Flow of Events**

If logged in as the Owner:

Case 1:

1. The owner selects the books button, he/she is directed to a page with a table that contains a list of books with the respective prices.
2. To select a book to be deleted, the owner must select the row of the book that they wish to delete
3. To delete a selected book, the owner simply has to click on the delete button that is located next to the two text fields on the Owner's books page.

Case 2:

1. The owner selects the customers button, he/she is directed to a page with a table that contains a list of books with the respective prices.
2. To select a customer to be deleted, the owner must select the row of the customer that they wish to delete
3. To delete a selected customer, the owner simply has to click on the delete button that is located next to the two text fields on the Owner's customers page.

All button clicks, page switches, and item additions are processed through backend methods by the server.

### **Exit Conditions**

To exit this use case, the owner simply needs to click the back button to return to the start screen. By doing so, any book rows or customer rows that are selected by the user will be deselected.

**Exceptions**

Not applicable. Any user or book item can be deleted from the table.

**Special Requirements**

The user must be logged in as the Owner. A customer does not have the ability to add or delete any selected items to the bookstore. They are able to buy books from the bookstore but this principle is significantly different from adding or deleting books and customer data from the bookstore application's database.



## Use case 5

### **Name**

BuyBook

### **Participating actors**

Customer, Server

### **Entry Conditions**

The user must be logged in as a customer

### **Flow of Events**

1. After logging into the bookstore as a customer, the user is redirected to a page with an organized table of books that are available for purchase.
2. The customer selects the books that they wish to buy by clicking a checkbox that is located next to the book name of their choice.
3. After selecting all the books that the customer wishes to purchase, the customer must click on the buy button which is located at the bottom left hand corner of the customer-start page.

### **Exit Conditions**

To exit this use case, the customer can simply click on the logout button after completing all of their necessary purchases and transactions.

### **Exceptions**

Not applicable

### **Special Requirements**

The user must be logged in to the application as a customer. This is because the owner does not have the ability to purchase any books on their start pages.

## Use case 6

### **Name**

Redeem Point

### **Participating actors**

Server, Customer

### **Entry Conditions**

The user is logged in as a customer

### **Flow of Events**

1. After logging into the bookstore as a customer, the user is redirected to a page with an organized table of books that are available for purchase.
2. The customer selects the books that they wish to buy by clicking a checkbox that is located next to the book name of their choice.
3. After selecting all the books that the customer wishes to purchase, the customer must click on the Redeem points and buy button which is located at the bottom left hand corner of the customer-start page.
4. After clicking said button, the customer is redirected to a page that contains the total points that the customer has left over as well as the total cost of the books purchased which should be zero.

Point accumulation and redemption calculations are processed by the application server

### **Exit Conditions**

To exit this use case, the customer should click the logout button after redeeming their points to purchase their books.

### **Exceptions**

Not applicable

### **Special Requirements**

The user must be logged in as a customer as owners cannot redeem points to buy books. The points must be used to cover the total cost of the books purchased by the user. Hence the total cost at the end of the transaction is always zero

## Use case 7

### **Name**

View cost Screen

### **Participating actors**

Customers, server

### **Entry Conditions**

The user is logged in as a customer

### **Flow of Events**

1. After logging into the bookstore as a customer, the user is redirected to a page with an organized table of books that are available for purchase.
2. The customer selects the books that they wish to buy by clicking a checkbox that is located next to the book name of their choice.
3. After selecting all the books that the customer wishes to purchase, the customer must click on the buy button which is located at the bottom left hand corner of the customer-start page.
4. After clicking said button, the customer is redirected to a page that contains the total points that the customer has left over as well as the total cost of the books purchased.

### **Exit Conditions**

To exit this use case, the customer should click the logout button after purchasing their books.

### **Exceptions**

There are no exceptions for this use case

### **Special Requirements**

The user must be logged in as a customer as owners cannot redeem points to buy books. The points must be used to cover the total cost of the books purchased by the user. Hence the total cost at the end of the transaction is always zero

Use case 8

**Name**

logout

**Participating actors**

Owner

customer

**Entry Conditions**

Must have logged in as a administrator or customer

**Flow of Events**

1. Once logged in as either a customer or administrator they will be greeted with their respective start screens
2. Once on that screen there is the option to logout
3. Simply click that button and you will be logged out and be returned to the login screen

**Exit Conditions**

There are no Exit conditions for this use case

**Exceptions**

There are no exceptions for this use case

**Special Requirements**

There are no special requirements for this use case

Use case 9

**Name**

View start screen

**Participating actors**

Customers

**Entry Conditions**

Must have a valid customer login in order to access it

**Flow of Events**

1. Customer must log in using their login credentials
2. They will then be greeted with their start screen
3. They have the ability to logout, buy a book or use points to purchase a book

**Exit Conditions**

Must click logout in order to exit this screen

**Exceptions**

There are no exceptions for this use case

**Special Requirements**

There are no special requirements for this use case

Use case 10

**Name**

Viewbooks

**Participating actors**

Owner

**Entry Conditions**

Must have a valid administrator account

**Flow of Events**

1. Admin must select this option from their start screen.
2. Then once selected they will be greeted with a screen that shows all the available books as well as the option to add or delete them

**Exit Conditions**

Must click back in order to exit

**Exceptions**

There are no exceptions for this use case

**Special Requirements**

There are no special requirements for this use case

Use case 11

**Name**

Validate owner

**Participating actors**

Owner

**Entry Conditions**

Must have a valid administrator account

**Flow of Events**

1. Users account has already been checked and it has passed
2. The owner start screen will then be displayed

**Exit Conditions**

There are no exit conditions for this use case

**Exceptions**

There are no exceptions for this use case

**Special Requirements**

There are no special requirements for this use case

Use case 12

**Name**

Validate account

**Participating actors**

Customers

**Entry Conditions**

Must have a valid customer account

**Flow of Events**

1. Users account has already been checked and it has passed
2. The customer start screen will then be displayed

**Exit Conditions**

There are no exit conditions for this use case

**Exceptions**

There are no exceptions for this use case

**Special Requirements**

There are no special requirements for this use case



Use case 13

**Name**

Validate customer

**Participating actors**

Customer

**Entry Conditions**

Must have a valid customer account in order to work

**Flow of Events**

1. Customer must enter their credentials into the login screen
2. The credentials are checked to see if they are valid in the text file
3. The user will continue if their information is valid

**Exit Conditions**

No exit conditions

**Exceptions**

Invalid or incorrect credentials will be met with an error message

**Special Requirements**

No special requirements

Use case 14

**Name**

GoBack

**Participating actors**

Owner

Customer

**Entry Conditions**

Both must be logged in to their respective accounts.

**Flow of Events**

1. When navigating throughout the program the user will have the option to go to the previous page.
2. Once clicked it will bring the user back to the previous page.

**Exit Conditions**

Must be presently on a screen that poses this button.

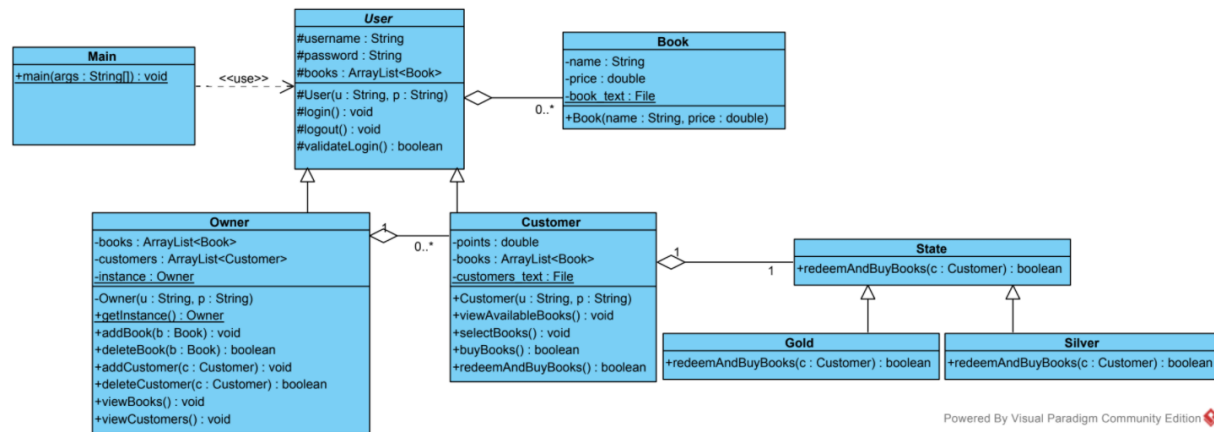
**Exceptions**

No exceptions for this use case

**Special Requirements**

Must be on a screen that has this button present

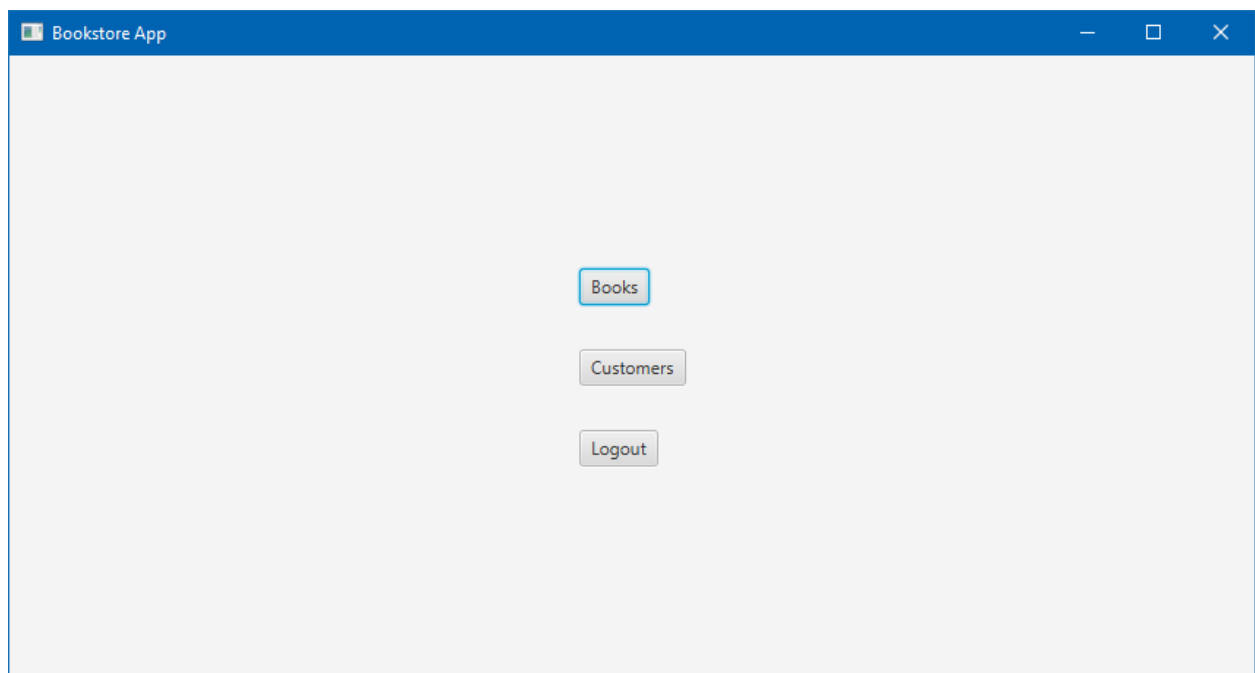
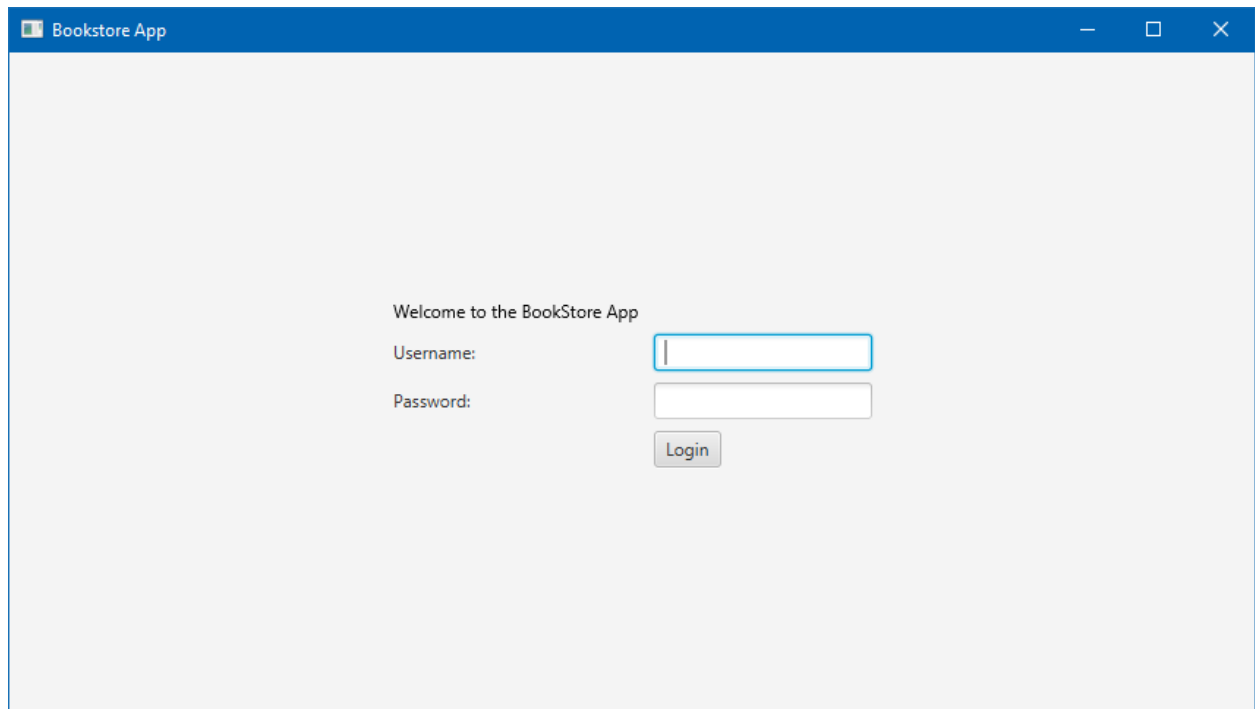
## UML-Class Diagram:



## State Design Pattern Rationale:

The rationale behind using the State Design Pattern is that with this design it is much easier to add more states if necessary without altering the code very much. For example, if the prof had decided that there would be 3 states (Bronze, Silver, and Gold, for example) then the only noticeable difference would be an extra class called “Bronze” and that would represent the Bronze state. That is the purpose / rationale of the State Design Pattern.

## Example ScreenShots of Program



Bookstore App

Book Name	Book Price
Book 1	50.0
Book 2	100.0
Book 3	200.0
Book 4	500.0
Book5	75.0

Name:

Price:

Add

Delete

Back

Bookstore App

Username	Password	Points
test	123	0.0
bob	chicken1	0.0
bill	newman23	0.0
jill	undersea123	0.0

Username:

Password:

Add

Delete

Back

Bookstore App

Welcome!

Books	Price	select	
Book 1	50.0		
Book 2	100.0		
Book 3	200.0		
Book 4	500.0		
Book5	75.0		

Buy

Redeem Points and Buy

Logout

Bookstore App

Total Cost: XXX

Points: XXX

Logout