**Use Cases**

|  |  |  |
| --- | --- | --- |
| **Use Case** |  | **User Login** |
| Brief Description |  | This use case allows a user to login to the bookstore system |
| Actors |  | User |
| Pre-conditions |  | User has not logged into the system |
| Main Flow | A1 | This use case begins when the user clicks “Login” in the main window |
|  | A2 | The login window pops up |
|  | A3 | The user fills in the User Name and Password slots |
|  | A4 | The system checks the input and verifies the user |
|  | A5 | The system retrieves the full name, user ID, and user type of the logged in user |
|  | A6 | The system terminates the use case with indication of successful login and a welcome message with the user’s full name and user type. |
| Alternative Flows | A7 | An error message is displayed if login is unsuccessful |
|  | A8 | User will be allowed to try logging in up to 3 times |
| Exception | A9 | No message is provided if the user cancels the login action |
| Post-conditions |  | The login window is closed if login is successful |

|  |  |  |
| --- | --- | --- |
| **Use Case** |  | **Add Book** |
| Brief Description |  | This use case allows a Customer to add a Book to their Booklist |
| Actors |  | Customer |
| Pre-conditions |  | Customer has logged into the system |
| Main Flow | A1 | This use case begins when the customer is in the main window |
|  | A2 | The customer looks through the book catalog |
|  | A3 | The customer select a book they wish to add to their booklist |
|  | A4 | The customer clicks “Add Book” |
|  | A5 | The book is added to the booklist |
| Post-conditions |  | Customer remains in the main window |

|  |  |  |
| --- | --- | --- |
| **Use Case** |  | **Change Book Price** |
| Brief Description |  | This use case allows a Manager to change book prices |
| Actors |  | Manager |
| Pre-conditions |  | Manager has logged into the system |
| Main Flow | A1 | This use case begins when the Manager is in the main window |
|  | A2 | The Manager looks through the catalog |
|  | A3 | The Manager finds a book price they wish to change |
|  | A4 | The Manager enters a price in the “New Price” column |
|  | A5 | The Manager clicks “Change Price” |
|  | A6 | The Manager clicks “Confirm Changes” |
|  | A7 | The system saves the prices changed |
| Post-conditions |  | The Manager remains in the main window |

**Checkout books**

Once a customer has added books to their booklist using the Add Book use case, they are able to checkout the books. The customer clicks “Checkout” in the mainwindow and will be brought to the checkout page.

**Classes from Sequence Diagrams**

public class MainWindow()

{

private Booklist bl = new Booklist();

private BookDataBase bdb = new BookDataBase();

private UserData ud = new UserData();

public void login() {

public LoginWindow lw = new LoginWindow();

while(lw.result && lw.tries < 3 && !ud.result)

if(lw.result)

ud.login(username, password);

if(ud.result) {

print(“Welcome “ + ud.username + ud.type);

else{

print(“error. Invalid username or password”);

}

public void selectBook() {} //select book in in catalog

public void changePrice(int newPrice) {

updatePrice(); }

public void confirmChanges() { bdb.updatePrices(int bookID, int newprice); }

private void updatePrice() {} //update prices in mainwindow

public void addBook(int bookID) { bl.addBook(bookID); }

public void checkout() {} //show checkout page

}

public class LoginWindow()

{

public boolean result = true;

public void ok\_button() {}

public void cancel\_button() { result = false; }

}

public class UserData()

{

public boolean result;

public string type;

public string username;

public void login(usrname, password) {

result = loginVerifier(usrname, password);

}

private boolean loginVerifier(usrname, password) {

if( valid login ) {

type = usertype;

username = usrname;

return true;}

else

return false;

}

}

public class BookDataBase()

{

updatePrices(bookID, newPrice) {} //set new book price

}

public class Booklist()

{

private List<Integer> booklist = new List<Integer>;

public void addBook(bookID){

booklist.add(bookID);

}

}