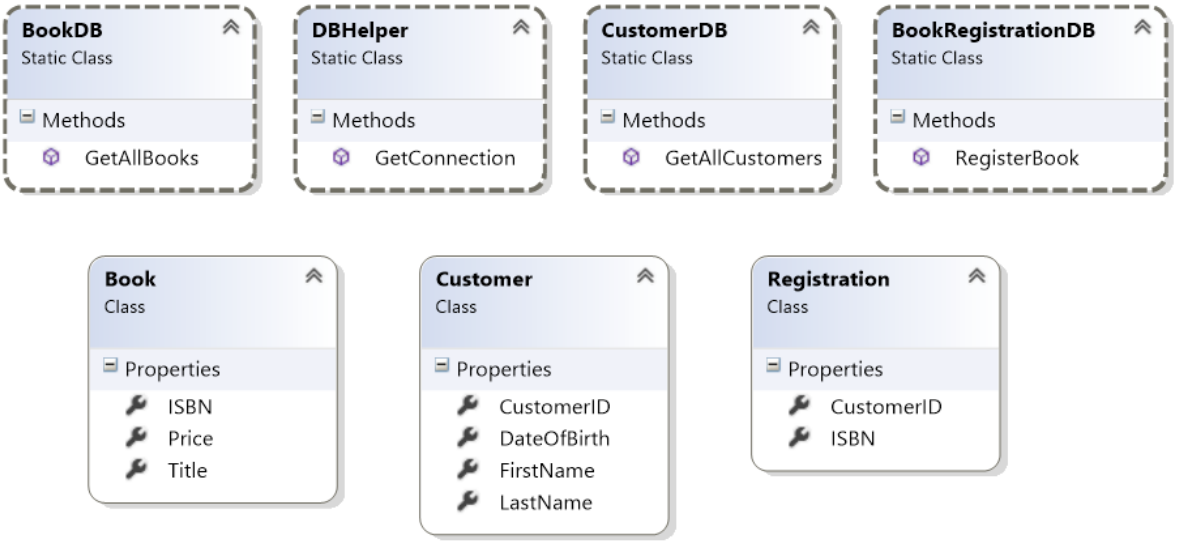
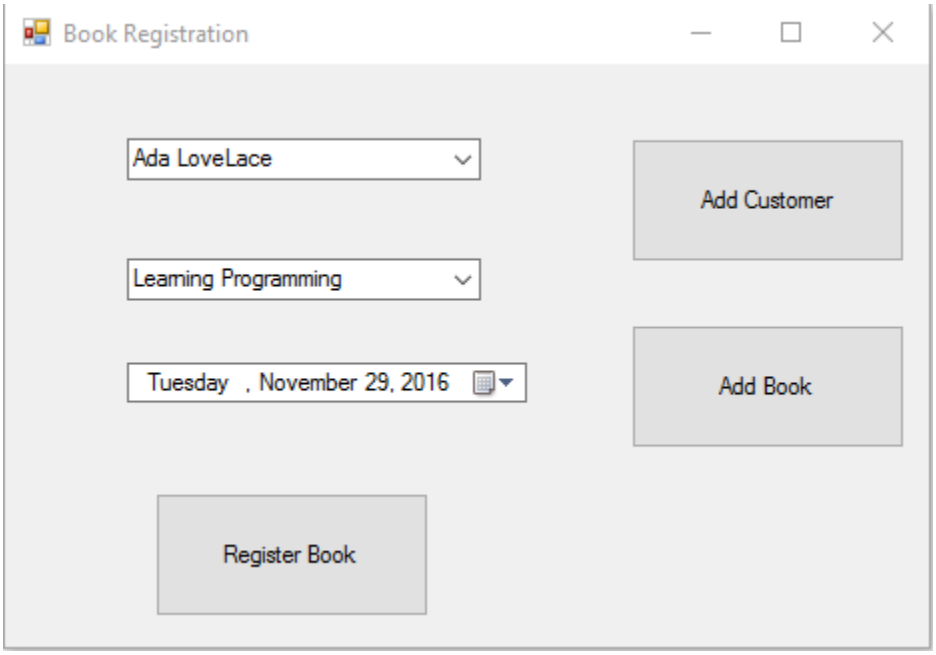
CPW 116 Final

Instructions: You are tasked with creating an application for customers to register books. Below is a class diagram. You must structure your classes as the diagram shows with the specified methods and properties. If your application is not object-oriented, you will not receive a passing grade.

*NOTE: The diagram may not show all required properties/methods



Sample Screenshot



Operation

- The user selects a Customer name and Book title from the combo boxes, enters a date in the text box, and clicks the Register Book button to add a new row to the Registrations table.
- After the user registers a product or clicks the Cancel button to cancel the add operation, the Add Registration form is cleared.

The GetAllBooks method in the BookDB class should return a List<Book> object that can be bound to the Books combo box on the main form, the GetAllCustomers method in the CustomerDB class should return a List<Customer> object that can be bound to the Customer combo box, and the RegisterBook method in the BookRegistrationDB class should accept a Registration object and return a Boolean value that indicates if the operation was successful. Use the value that's returned by the RegisterBook method to display a message indicating the result of the operation.

Once you have those steps completed you will need to add a couple additional things.

- Add a new form to add a customer to the database
- Add a button on the main form to open the add customer form
- Add a new method in the CustomerDB class to insert a customer into the database
- Add a new form to add a Book to the database
- Add a button on the main form to open the add book form
- Add a method in the BookDB class to insert a product into the database
- Refresh the customer/book lists when a new item is added to the database

EXTRA CREDIT CRITERIA (No guarantee of amount of extra credit points. This is mainly for students who are ready for an extra challenge)

- When adding a book, ensure the ISBN number is only composed of digits
    - Or allow the user to input dashes and remove them before storing in database
- Add validation on forms
    - All required field textboxes should have input
    - Customer DateOfBirth should be entered as a valid date/time
    - No empty strings or whitespace strings should be entered into the database
    - Leading/Trailing space should be removed from strings
- When registering a book, ensure a Customer and Book are selected, along with a valid date
- When a customer is selected, only show books that they have NOT registered yet
- Create forms to view all books/customers
    - When a customer/book is selected, allow edit and delete functionality
    - The Edit and Add customer forms could be combined into a single Windows Form
    - The Edit and Add book forms could be combined into a single Windows Form

- Add a tab order and Access Keys to your form to make them easier to navigate
- Add a new form to display customers and their associated registrations
    - Enable the ability to remove a registration from a customer