

Assignment #2

Due at .

Overview

You may work individually or work in pairs to create a series of connected web forms. The work you do on this assignment will likely be re-used in your next assignment.

Context

You will be continuing the web application that allows book salespeople (called book reps) to easily research their publisher's offerings and manage their customers.

Submitting

Put your assignment in a folder based on your last name+assign2 (e.g., mine would be ConnollyAssign2). Put any resources used by your assignment into this folder. Copy the folder to the submit drive.

Grading

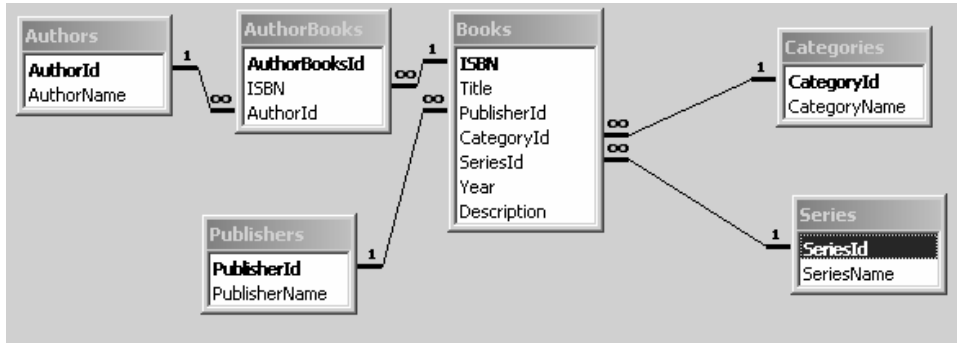
The grade for this assignment will be broken down as follows:

Usability and Visual Design	15%
Program Design and Documentation	25%
Features	60%
Extra Marks	10 – 15%

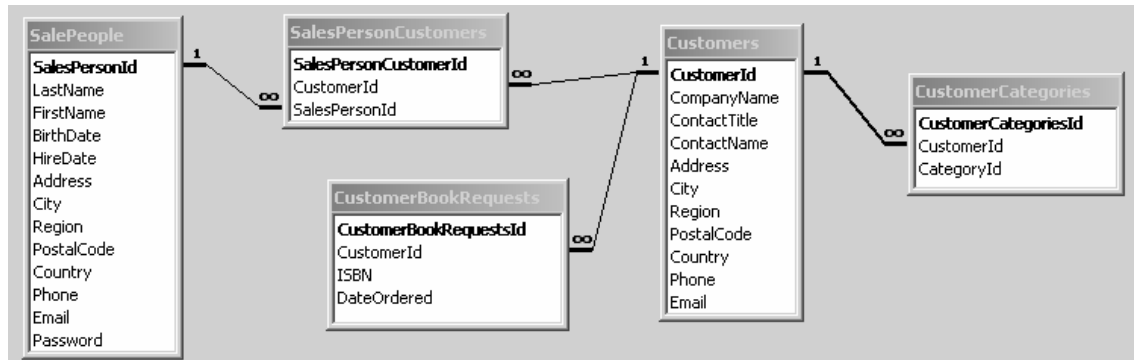
Database

You will use the same two databases in this assignment as the last assignment.

- **BookCatalog.** This represents the product catalog for the publisher. Salespeople have limited, read-only access to this database.



- **CustomerRelations.** This represents the customer information for the salesperson. The site will allow salespeople to view and edit the data in this database.



Functional Requirements

You must create the following web forms.

1. **Select Customer Form.** This form must display a list of all customers in the Customers table. Each customer name must be a link to the *Manage Customer Form* that passes the selected `CustomerID` for that customer via the querystring. There must be a button as well for creating a new customer; this will redirect to the *Manage Customer Form*.
2. **Manage Customer Form.** This form must allow the user to enter all the modifiable values in the Customers table. If the `CustomerID` is passed in via a querystring, then let the user edit the existing values in the Customers tables. If no `CustomerID` is passed in, then assume you are creating a new customer.

`CompanyName`, `ContactName`, `Phone` and `Email` are required fields. `Email` must be a valid email and must be unique (i.e., no other customers can have the same email). When the user submits the form and the data is valid either add a new record if this customer doesn't exist, otherwise update the database table.

The form must also supply some mechanism for the user to view/add/remove which categories a customer is interested in (from the `CustomerCategories` table).

The form must also supply some mechanism for the user to view or remove (but not add) a customer's book requests (from the `CustomerBookRequests` table). It should display the ISBN, a small image of the book cover, the book title, and the date ordered. The book image and the book title should be links that will display the *Book Information Form* for that book.

3. **Book Listing Form.** This form will display a listing of the books in the Books table. The list should show the ISBN, a small image of the book cover, the title, and the year. The book image and the book title should be links that will display the *Book Information Form* for that book.

This form can display a variable number of books in the list. If there are no querystring parameters, then the page should display all the books in the table. It should also support the ability to pass in either an `AuthorId`, `PublisherId`, `CategoryId`, or a `SeriesId`; in such a case, then the form would only display the books for that author, publisher, category, or series. In such a case, there should be heading at the top of the list that specifies what subset of data is being listed.

4. **Book Information Form.** This form will display the information for a single book. The book to display should be specified by passing in the requested book's ISBN via a querystring parameter. It should display the larger cover image, the title, the publisher's name, category name, series name, authors' name(s), the year and the description. The publisher, category, author, and series names should be links. When the link is clicked, this should display the *Book Listing Form* and pass it the appropriate `PublisherId`, `AuthoId`, etc.

This form should have the ability to add the displayed book to the *Request Cart* (sort of like a shopping cart).

5. **Request Cart Form.** This form will display the contents of the Request Cart. It should display the ISBN, a small image of the book cover, and the title. The book image and the book title should be links that will display the *Book Information Form* for that book. As well, each book should have a link to remove it from the cart. There should also be a button to remove all items from the cart.

The form should have an area in which the user can enter the customer email and a button that will request the books (requesting the books will also empty the request cart). If the customer email exists, then the books should be added to the `CustomerBookRequests` table.

6. **Home Page.** A central page with links to all pages. The home page should list your name or the names of the people in your group. It should also mention that this is assignment #2.

Additional Requirements

1. All of these web forms must be consistently formatted using a single consistent Theme and be contained within a Master Page.
2. The Master Page must contain links to the other pages using the Menu control.
3. Data access code must be contained as much as possible within separate classes (i.e., within a data access layer). Your code-behind pages should thus contain little or no ADO.NET code.

Optional Requirements (for extra marks)

1. Provide a drop-down on the Master Page that allows the user to change the site's Theme. Create at least three themes for the user to choose between.
2. Provide the user with the ability to change the sort order of the Book Listing Form. The user should be able to sort by title, ISBN, or date.