CSCI 6962: Server-Side Development and Programming

Project Assignment 3:
Validation Pages

Due Wednesday, November 19

Introduction

This assignment is the third of four assignments, in which you will continue to build up the web site for your term project. It is meant to give you practice in creating a “cart” for your site.

Cart Steps and Abilities

As specified in the project requirements, something on your site must require “user choices” to be stored in a “cart” of whatever sort is most relevant to your site. The type of validation will depend on your site, of course.

At a minimum, however, users should have the ability to:

- **Add selections to the cart.**
  This can be done directly from your list of “choices” (as we have done in the examples in class), or from the “details” page of those choices – whatever you find more appropriate.

- **View the cart.**
  Selections should be displayed in either a **DataTable** or a **GridView**. Each row should display the relevant information about that selection (such as price, quantity, etc.). This will also be determined by the type of thing that you have in your cart.

- **Modify the cart.**
  The user should have the ability to modify what they have added to the cart in at least one way. This might include removing items, changing the quantity of an item, or returning to the “details” page of that item to enter more information (this might be appropriate if your cart contains answers to “survey” questions, for example).

Support Class for Cart Items

If you have not done so already, this is the point at which you will need to create a simple “support” class for individual items in a cart. This can be very simple, with getters and setters for the properties that your items have, and you may feel free to use my “Widget” class as a starting point.

The main design decision that you will have at this point is simply **what information** (that is, what member variables) you will need to store/display for instances of your “choice” objects.
• As with the examples I have shown you, you will need a constructor that takes your ID (or whatever key field you use), and constructs an object with that ID. At this point, it can simple be a “hardwired” prototype constructor, that creates one of a small number (such as 3) possible items.

• You will also need a static method that returns a list of all choices the user has – in other words, a method that constructs an instance of each of your 3 possible objects, and returns them as a list.

Modified “Choices” Page

Because of this, your page of “choices” the user has to choose from should no longer be “hardwired” as it was in the first couple of assignments. It should now be generated from the above static method, and displayed using a DataTable or GridView.

Support Class for the Cart Itself

You are also to create a class that stores the “cart” of those items (that is, the list of all item objects that have been added to the cart). This can either be a separate support class (as I showed you for the APS examples), or it can be implemented within the managed bean of a JSF site. Again, feel free to use my code as a starting point.

As shown in the class examples, your cart will need:

• A method that returns all cart items as list, to be displayed in the DataTable/GridView.

• A method that adds an item to the cart. It should probably take the “id” of that item as a parameter, and then use the constructor described above to create the corresponding item and add it to the cart list.

• At least one additional method to support whatever ability that you are giving users to modify the contents of their cart.

What to Turn In

You are to email me your web site files. My email is jrsullins@ysu.edu.

As before, you can find the source code as a subdirectory of the NewBeans Projects directory of Documents, or as a subdirectory of the Visual Studio 2013/WebSites directory of Documents. Please use a zip program such as gzip to compress the entire directory and send it to me.