CS4985
Final Exam
**Due Date: Monday, April 20, 11:55 PM**

# Completing the Digital Upscale Manager for Ballgames Site

## Overview:

We'll clean up the issues from previous iterations and add some new functionality. This is your FINAL EXAM. You are free to use your text and other static materials, but may not discuss this material with anyone else. This includes, but is not limited to classmates, TAs, online discussion forums, students outside the class, employers, neighbors, or anyone else. It is to be your work exclusively. (And honestly, this stuff is almost just like what you did for the homeworks, so there really shouldn't be much of a need to talk with anyone).

## What to do:

### *Step 1 – Revisit Projects 01 - 03*

Be sure to read the comments in the Gradebook for the Projects so far. Start by fixing all of the old problem so that your site is up to date meeting these requirements.

### *Step 2 – New Features*

Once you are comfortable that your previous work is up to speed, then start concentrating on the new material:

**Maintaining the Support Staff Details**

So far our site hasn't had much to do with the Support Staff. We'll change that in this project by creating a page named `MaintainSupport.aspx` to do just that. This page should link to the Support table in the Access database. Include a DropDownList control to display a sorted list of the names of all Support Staff in the database. When the user selects a Support Staff member, display all of that person's details (ID, name, email, and phone number) in a FormView control. Have this control provide the ability to update, delete, or create a new Support Staff member. Include validation so that users are required to enter a value for each of the fields. If any validation issues come up, be sure to flag the appropriate field and include a complete, detailed description of the problem in a ValidationSummary control under the FormView. As always, be sure to handle any exceptions that may come about so that the page does not crash.

**Feedback Incidents by Support Staff**

Your website already provides the ability to view the feedback by Customers. However you realize that it's often useful to view them by the Support Staff who handled them as well. Provide a page named `FeedbackBySupport.aspx` for this functionality. Additionally you'll need to use your knowledge of Object Data Sources and create two of them for this page.

The first will be used for the DropDownList displaying the Support Staff name. The page will also have a GridView to display the DateOpened, Software Name, and Customer name for all *open* feedback incidents associated with that Support Staff. In order to get this page working correctly, you'll need to define two classes, each with a single method:

**SupportDatabase**

```
public static IEnumerable GetAllSupportStaff()
```

**FeedbackDatabase**

```
public static IEnumerable GetOpenFeedbackIncidents(int supportStaffId)
```

Be sure to bind the DropDownList control to an Object Data Source associated with the SupportDatabase. You can then use the SELECT method present to get all of the Support Staff and sort them by name.

The GridView should be bound to a second Object Data Source associated with the FeedbackDatabase. This SELECT method should select only *open* feedback incidents for the selected Support Staff. Note that you'll need to join the Feedback and Customer tables in order to get the Customer's name. Be sure to sort the results by the date the Feedback was opened.

Because your data objects need to get their data from the database, you'll need to create another class called BallgameDatabase that contains a public function to return the connection string from web.config. Be sure to place all of these classes create in an appropriate location in the Solution.

### Updating Feedback Incidents

Thankfully our awesome support staff does an excellent job to address the feedback incidents in a timely manner. However, there's no way for these folks to update the database. We'll fix that next.

Create a page named `UpdateFeedback.aspx` to display a sorted list of Customer names in a DropDownList. When the user selects a Customer, display all of their feedback (open and closed) in a GridView listing the Support ID, Software ID, Date Opened, Date Closed, Feedback Title, and Feedback Description. Because the description can contain up to 2,000 characters, be sure that this is displayed in a multi-line TextBox (Hint: use a template field). Sort the items in the GridView by Support ID. Be sure to display an Edit button next to each feedback incident. Allow the user to edit the Date Closed and Description fields *only*.

When doing this, please be sure to use two Object Data Sources on this page:

One data source for the Customer DropDownList. To do this correctly, you'll need to develop a CustomerDatabase class with a SELECT method:

```
public static IEnumerable GetCustomersWithFeedback()
```

This should only get the Customers with Feedback that have been assigned to a Support member, then sort these results by name. Hint: You can use a subquery in your SELECT statement such as:

```
SELECT CustomerID, Name
FROM Customers
WHERE CustomerID IN
     (SELECT DISTINCT CustomerID FROM Feedback
      WHERE SupportID IS NOT NULL)
ORDER By Name
```

The second Object Data Source will be used for the GridView control. To do this, you'll need to add to the FeedbackDatabase class a SELECT method:

```
public static IEnumerable GetCustomerFeedback(int CustomerId)
```

It will also need to define the following UPDATE method:

```
public static int UpdateFeedback(Feedback originalFeedback,
                                 Feedback newFeedback)
```

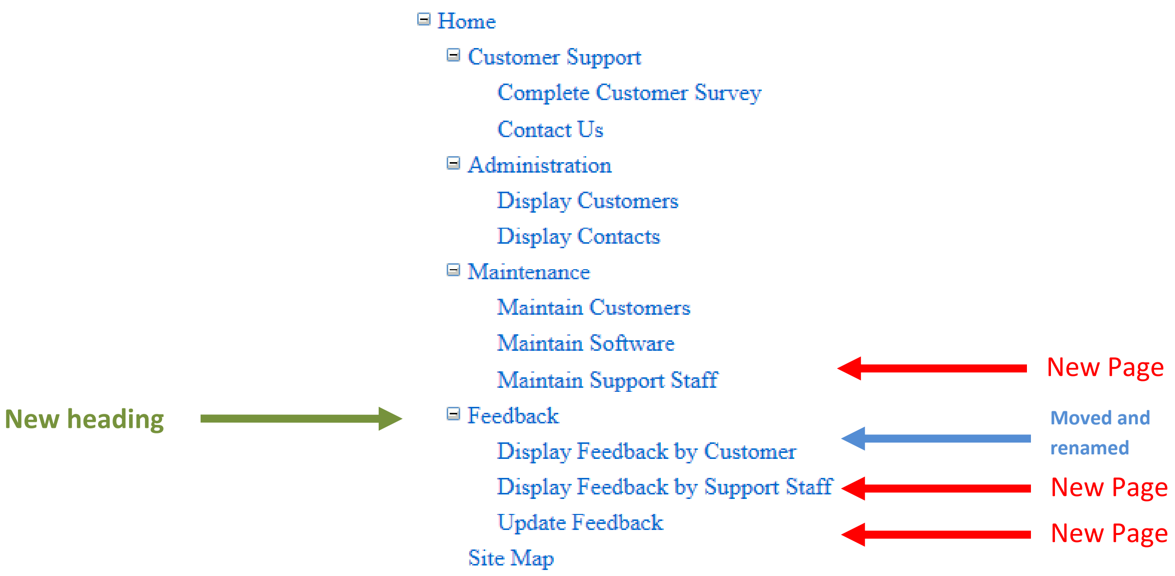
NOTE: Remember that it is perfectly acceptable to have a NULL value for a Feedback incident's DateClosed field. To do this, you'll need a little extra C# code. That's because the value "Jan 1 0001 12:00:00 AM" will be passed to the method if the field is null, and that's not a valid DateTime value. Therefore your method will need to set the DateClosed parameter with:

```
if (newFeedback.DateClosed ==
                  Convert.ToDateTime("01/01/0001 12:00:00 AM"))
{
      updateCommand.Parameters.AddWithValue("DateClosed",
                                            DBNull.Value);
}
else
{
      updateCommand.Parameters.AddWithValue("DateClosed",
                        newFeedback.DateClosed);
}
```

As always, be sure to handle any exceptions that may come about so that the page does not crash.

**Updating the Navigation**

Because we have three new pages, you'll need to modify the site's navigation. We'll ask that you modify the existing navigation so that it is arranged as shown:



- ⊟ Home
  - ⊟ Customer Support
    - Complete Customer Survey
    - Contact Us
  - ⊟ Administration
    - Display Customers
    - Display Contacts
  - ⊟ Maintenance
    - Maintain Customers
    - Maintain Software
    - Maintain Support Staff
  - ⊟ Feedback
    - Display Feedback by Customer
    - Display Feedback by Support Staff
    - Update Feedback
  - Site Map

Note the 'Maintain Incidents' page has been renamed to 'Display Feedback by Customer' and moved to the 'Feedback' area.  Also the 'Maintain Support' page has been placed in the 'Maintenance' area.

**IMPORTANT**

We've made a big deal about style / formatting and repositories in this course (and previous courses) and I expect that you'll continue to follow those guidelines here as well.  The grading guide will be based on functionality, design, and style.  We know that ReSharper has problems with the call to BIND (which isn't a method at all!), so those things are acceptable; *however no other issues are acceptable.  If any issues (other than BIND) show, then you will lose all of the Style points*.

When you are finished and have tested the site to be sure that everything works correctly and passes ReSharper, please close Visual Studio and use 7-Zip to zip -the entire- Solution folder.  Please use 7-Zip to create a file with the extension .zip.  Also be sure to include the URL of your online repository in the Comments section on your submission.

Note that Visual Studio is going to create a lot of files and folders and I need them all to run (and grade) your work.  It's probably worth you taking a minute out of your day and downloading your Zip file to a new location and reopening it to be sure what you think is there, actually is there.

Upload your Zip file to Moodle.