

```
1 /*-----
2 * Author:      Dan Cassidy
3 * Date:        2015-06-23
4 * Assignment:  cView-P4
5 * Source File: Modify.aspx.cs
6 * Language:    C#
7 * Course:      CSCI-C 490, C# Programming, MoWe 08:00
8 * Purpose:     Code-behind file for Modify.aspx. Controls the process of modifying an item in the
9 *              database via the Entity Framework model.
10 -----*/
11
12 using System;
13 using System.Collections.Generic;
14 using System.Data.Entity;
15 using System.Linq;
16 using System.Web;
17 using System.Web.UI;
18 using System.Web.UI.WebControls;
19
20 namespace cView_P4_DanCassidy
21 {
22     public partial class Modify : System.Web.UI.Page
23     {
24         /*-----
25          * Name:      btnModify_Click
26          * Type:      Event Handler Method
27          * Purpose:   Handles showing and hiding the various controls to aid in allowing the user to
28          *            change the data associated with the chosen item.
29          * Input:    object sender, holds a reference to the object that raised this event.
30          * Input:    EventArgs e, holds data related to this event.
31          * Output:   Nothing.
32          -----*/
33         protected void btnModify_Click(object sender, EventArgs e)
34         {
35             // Quick check to make sure that something is selected.
36             switch ((Global.Enums.ItemTypes)ddlItemType.SelectedIndex)
37             {
38                 case Global.Enums.ItemTypes.Business:
39                     if (gViewBusiness.SelectedIndex == -1)
40                         return;
41                     break;
42
43                 case Global.Enums.ItemTypes.Park:
44                     if (gViewPark.SelectedIndex == -1)
45                         return;
46                     break;
47
48                 case Global.Enums.ItemTypes.PublicFacility:
49                     if (gViewPublicFacility.SelectedIndex == -1)
50                         return;
51                     break;
52
53                 default:
54                     break;
55             }
56
57             // Hide things until needed.
58             lblError.Visible = false;
59             lblResult.Visible = false;
60             mViewDisplay.ActiveViewIndex = -1;
61             mViewModifyBasic.ActiveViewIndex = -1;
62             mViewModifySpecific.ActiveViewIndex = -1;
63             btnModify.Visible = false;
64             btnSaveChanges.Visible = false;
65             btnBack.Visible = false;
66         }
67     }
68 }
```



```

133         txtFeatureBaseball.Text = parkToModify.FeatureBaseball.ToString();
134         txtFeatureBasketball.Text = parkToModify.FeatureBasketball.ToString();
135         txtFeatureGolf.Text = parkToModify.FeatureGolf.ToString();
136         txtFeatureLargeMPField.Text = parkToModify.FeatureLargeMPField.
137             ToString();
138         txtFeatureTennis.Text = parkToModify.FeatureTennis.ToString();
139         txtFeatureVolleyball.Text = parkToModify.FeatureVolleyball.ToString();
140         break;
141
142     case Global.Enums.ItemTypes.PublicFacility:
143         // Basic UI prep.
144         mViewModifyBasic.ActiveViewIndex = 0;
145         lblName.Text = Global.Strings.PublicFacilityName +
146             Global.Strings.Separator;
147         lblType.Text = Global.Strings.PublicFacilityType +
148             Global.Strings.Separator;
149         btnSaveChanges.Visible = true;
150         btnBack.Visible = true;
151
152         // Get object to modify.
153         keyToModify = gViewPublicFacility.SelectedDataKey.Value;
154         PublicFacility publicFacilityToModify = database.PublicFacilities.
155             Find(keyToModify);
156
157         // Fill in textbox values.
158         txtName.Text = publicFacilityToModify.Name;
159         txtName.Enabled = false;
160         txtType.Text = publicFacilityToModify.Type;
161         txtStreetAddress.Text = publicFacilityToModify.StreetAddress;
162         txtCity.Text = publicFacilityToModify.City;
163         txtState.Text = publicFacilityToModify.State;
164         txtZip.Text = publicFacilityToModify.Zip;
165         txtLatitude.Text = publicFacilityToModify.Latitude.ToString();
166         txtLongitude.Text = publicFacilityToModify.Longitude.ToString();
167         txtPhone.Text = publicFacilityToModify.Phone;
168         break;
169
170     default:
171         throw new InvalidOperationException(
172             "Invalid item type dropdown value.");
173     }
174 }
175 }
176 catch (Exception ex)
177 {
178     lblError.Text = "Error: " + ex.Message;
179     lblError.Visible = true;
180 }
181 }
182
183 /*-----
184 * Name:      btnSaveChanges_Click
185 * Type:      Event Handler Method
186 * Purpose:   Handles saving the changed data for the specific item back to the database via
187 *            the Entity Framework model.
188 * Input:     object sender, holds a reference to the object that raised this event.
189 * Input:     EventArgs e, holds data related to this event.
190 * Output:    Nothing.
191 -----*/
192 protected void btnSaveChanges_Click(object sender, EventArgs e)
193 {
194     // Hide things until needed.
195     lblError.Visible = false;
196     lblResult.Visible = false;
197     mViewDisplay.ActiveViewIndex = -1;
198     mViewModifyBasic.ActiveViewIndex = -1;

```

```

199     mViewModifySpecific.ActiveViewIndex = -1;
200     btnModify.Visible = false;
201     btnSaveChanges.Visible = false;
202     btnBack.Visible = false;
203
204     try
205     {
206         using (CViewDataEntities database = new CViewDataEntities())
207         {
208             object keyToModify = null;
209
210             // Choose what to do based on the selected item type.
211             switch ((Global.Enums.ItemTypes)ddlItemType.SelectedIndex)
212             {
213                 case Global.Enums.ItemTypes.Business:
214                     // Get object to modify.
215                     keyToModify = gViewBusiness.SelectedDataKey.Value;
216                     Business businessToModify = database.Businesses.Find(keyToModify);
217
218                     // Save values to object.
219                     businessToModify.Name = txtName.Text.Trim();
220                     businessToModify.Type = txtType.Text.Trim();
221                     businessToModify.StreetAddress = txtStreetAddress.Text.Trim();
222                     businessToModify.City = txtCity.Text.Trim();
223                     businessToModify.State = txtState.Text.Trim();
224                     businessToModify.Zip = txtZip.Text.Trim();
225                     businessToModify.Latitude = SimpleConvert.ToDecimal(txtLatitude.Text.
226                         Trim());
227                     businessToModify.Longitude = SimpleConvert.ToDecimal(txtLongitude.Text.
228                         Trim());
229                     businessToModify.Phone = txtPhone.Text.Trim();
230                     businessToModify.LicenseIssueDate = SimpleConvert.ToDateTime(
231                         txtLicenseIssueDate.Text.Trim());
232                     businessToModify.LicenseExpirDate = SimpleConvert.ToDateTime(
233                         txtLicenseExpirDate.Text.Trim());
234                     businessToModify.LicenseStatus = txtLicenseStatus.Text.Trim();
235                     businessToModify.CouncilDistrict = txtCouncilDistrict.Text.Trim();
236
237                     // Modelled off http://stackoverflow.com/a/15339512.
238                     database.Businesses.Attach(businessToModify);
239                     database.Entry(businessToModify).State = EntityState.Modified;
240                     break;
241
242                 case Global.Enums.ItemTypes.Park:
243                     // Get object to modify.
244                     keyToModify = gViewPark.SelectedDataKey.Value;
245                     Park parkToModify = database.Parks.Find(keyToModify);
246
247                     // Save values to object.
248                     parkToModify.Type = txtType.Text.Trim();
249                     parkToModify.StreetAddress = txtStreetAddress.Text.Trim();
250                     parkToModify.City = txtCity.Text.Trim();
251                     parkToModify.State = txtState.Text.Trim();
252                     parkToModify.Zip = txtZip.Text.Trim();
253                     parkToModify.Latitude = SimpleConvert.ToDecimal(txtLatitude.Text.
254                         Trim());
255                     parkToModify.Longitude = SimpleConvert.ToDecimal(txtLongitude.Text.
256                         Trim());
257                     parkToModify.Phone = txtPhone.Text.Trim();
258                     parkToModify.FeatureBaseball = SimpleConvert.ToByte(
259                         txtFeatureBaseball.Text.Trim());
260                     parkToModify.FeatureBasketball = SimpleConvert.ToDecimal(
261                         txtFeatureBasketball.Text.Trim());
262                     parkToModify.FeatureGolf = SimpleConvert.ToDecimal(
263                         txtFeatureGolf.Text.Trim());
264                     parkToModify.FeatureLargeMPField = SimpleConvert.ToByte(

```

```

265         txtFeatureLargeMPField.Text.Trim());
266     parkToModify.FeatureTennis = SimpleConvert.ToByte(
267         txtFeatureTennis.Text.Trim());
268     parkToModify.FeatureVolleyball = SimpleConvert.ToByte(
269         txtFeatureVolleyball.Text.Trim());
270
271     // Modelled off http://stackoverflow.com/a/15339512.
272     database.Parks.Attach(parkToModify);
273     database.Entry(parkToModify).State = EntityState.Modified;
274     break;
275
276     case Global.Enums.ItemTypes.PublicFacility:
277         // Get object to modify.
278         keyToModify = gViewPublicFacility.SelectedDataKey.Value;
279         PublicFacility publicFacilityToModify = database.PublicFacilities.
280             Find(keyToModify);
281
282         // Save values to object.
283         publicFacilityToModify.Type = txtType.Text.Trim();
284         publicFacilityToModify.StreetAddress = txtStreetAddress.Text.Trim();
285         publicFacilityToModify.City = txtCity.Text.Trim();
286         publicFacilityToModify.State = txtState.Text.Trim();
287         publicFacilityToModify.Zip = txtZip.Text.Trim();
288         publicFacilityToModify.Latitude = SimpleConvert.ToDecimal(
289             txtLatitude.Text.Trim());
290         publicFacilityToModify.Longitude = SimpleConvert.ToDecimal(
291             txtLongitude.Text.Trim());
292         publicFacilityToModify.Phone = txtPhone.Text.Trim();
293
294         // Modelled off http://stackoverflow.com/a/15339512.
295         database.PublicFacilities.Attach(publicFacilityToModify);
296         database.Entry(publicFacilityToModify).State = EntityState.Modified;
297         break;
298
299     default:
300         throw new InvalidOperationException(
301             "Invalid item type dropdown value.");
302     }
303     // Save any changes to the database and refresh gridviews.
304     database.SaveChanges();
305     mViewDisplay.DataBind();
306 }
307 // Go back to item selection and let the user know the operation was successful.
308 ddlItemType_SelectedIndexChanged(sender, e);
309 lblResult.Text = "Item modified successfully.";
310 lblResult.Visible = true;
311 }
312 catch (Exception ex)
313 {
314     lblError.Text = "Error: " + ex.Message;
315     lblError.Visible = true;
316 }
317 }
318
319 /*-----
320 * Name:     ddlItemType_SelectedIndexChanged
321 * Type:     Event Handler Method
322 * Purpose:  Handles showing and hiding the various controls to aid in allowing the user to
323 *           select an item to modify.
324 * Input:    object sender, holds a reference to the object that raised this event.
325 * Input:    EventArgs e, holds data related to this event.
326 * Output:   Nothing.
327 -----*/
328 protected void ddlItemType_SelectedIndexChanged(object sender, EventArgs e)
329 {
330     // Hide things until needed.

```

```
331         lblError.Visible = false;
332         lblResult.Visible = false;
333         mViewDisplay.ActiveViewIndex = -1;
334         mViewModifyBasic.ActiveViewIndex = -1;
335         mViewModifySpecific.ActiveViewIndex = -1;
336         btnModify.Visible = false;
337         btnSaveChanges.Visible = false;
338         btnBack.Visible = false;
339
340         // Reset selected indexes of GridView controls.
341         gridViewBusiness.SelectedIndex = -1;
342         gridViewPark.SelectedIndex = -1;
343         gridViewPublicFacility.SelectedIndex = -1;
344
345         // Show the appropriate view.
346         switch ((Global.Enums.ItemTypes)ddlItemType.SelectedIndex)
347         {
348             case Global.Enums.ItemTypes.Business:
349                 mViewDisplay.ActiveViewIndex = 0;
350                 btnModify.Visible = true;
351                 break;
352
353             case Global.Enums.ItemTypes.Park:
354                 mViewDisplay.ActiveViewIndex = 1;
355                 btnModify.Visible = true;
356                 break;
357
358             case Global.Enums.ItemTypes.PublicFacility:
359                 mViewDisplay.ActiveViewIndex = 2;
360                 btnModify.Visible = true;
361                 break;
362
363             default:
364                 break;
365         }
366     }
367 }
368 }
```