

```

1  /*-----
2  * Author:      Dan Cassidy
3  * Date:        2015-06-23
4  * Assignment:  cView-P4
5  * Source File: Search.aspx.cs
6  * Language:    C#
7  * Course:      CSCI-C 490, C# Programming, MoWe 08:00
8  * Purpose:     Code-behind file for Search.aspx. Controls the interface whereby a user can search
9  *              the tables.
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 Search : System.Web.UI.Page
23     {
24         /*-----
25         * Name:      btnSearch_Click
26         * Type:      Event Handler Method
27         * Purpose:   Handles the actual search and display of the results.
28         * Input:     object sender, holds a reference to the object that raised this event.
29         * Input:     EventArgs e, holds data related to this event.
30         * Output:    Nothing.
31         -----*/
32         protected void btnSearch_Click(object sender, EventArgs e)
33         {
34             // Hide things until needed.
35             lblError.Visible = false;
36             lblResult.Visible = false;
37             mViewSearchResults.ActiveViewIndex = -1;
38
39             // Pre-conversions to save some space.
40             string toSearchFor = txtSearch.Text.Trim();
41             byte toSearchForByte = SimpleConvert.ToByte(toSearchFor);
42             DateTime toSearchForDateTime = SimpleConvert.ToDateTime(toSearchFor);
43             decimal toSearchForDecimal = SimpleConvert.ToDecimal(toSearchFor);
44
45             int resultCount = 0;
46             int viewToDisplay = -1;
47             GridView gViewToDisplay;
48
49             int selectedIndex;
50             object baseObject;
51
52             // Determine what comparator to use; default is "|" for "contains".
53             string comparator = "|";
54             if (ddlComparatorsStrings.Visible == true)
55             {
56                 switch ((Global.Enums.ComparatorsStrings)ddlComparatorsStrings.SelectedIndex)
57                 {
58                     case Global.Enums.ComparatorsStrings.NotContain:
59                         comparator = "!=";
60                         break;
61
62                     case Global.Enums.ComparatorsStrings.Equal:
63                         comparator = "==";
64                         break;
65
66                     case Global.Enums.ComparatorsStrings.NotEqual:

```

```

67         comparator = "!=";
68         break;
69
70     default:
71         break;
72     }
73 }
74 else if (ddlComparatorsNotStrings.Visible == true)
75 {
76     switch ((Global.Enums.ComparatorsNotStrings)ddlComparatorsNotStrings.SelectedIndex)
77     {
78         case Global.Enums.ComparatorsNotStrings.NotContain:
79             comparator = "!=";
80             break;
81
82         case Global.Enums.ComparatorsNotStrings.Equal:
83             comparator = "==";
84             break;
85
86         case Global.Enums.ComparatorsNotStrings.NotEqual:
87             comparator = "!=";
88             break;
89
90         case Global.Enums.ComparatorsNotStrings.Greater:
91             comparator = ">";
92             break;
93
94         case Global.Enums.ComparatorsNotStrings.Less:
95             comparator = "<";
96             break;
97
98         case Global.Enums.ComparatorsNotStrings.GreaterEqual:
99             comparator = ">=";
100            break;
101
102         case Global.Enums.ComparatorsNotStrings.LessEqual:
103             comparator = "<=";
104             break;
105
106         default:
107             break;
108     }
109 }
110
111 // Do the search.
112 //
113 // The method I ended up using is a bit kludgy, but I had to resort to this because LINQ
114 // to Entities is stupid and doesn't even try to evaluate what it can server-side prior
115 // to attempting to translate the query to SQL and failing because indexers are not a
116 // SQL thing.
117 try
118 {
119     using (CViewDataEntities database = new CViewDataEntities())
120     {
121         switch ((Global.Enums.ItemTypes)ddlItemType.SelectedIndex)
122         {
123             case Global.Enums.ItemTypes.Business:
124                 viewToDisplay = 0;
125                 gViewToDisplay = gViewBusinessResults;
126                 selectedIndex = ddlBusiness.SelectedIndex;
127                 baseObject = database.Businesses.First()[selectedIndex];
128                 if (baseObject != null)
129                 {
130                     IEnumerable<Business> searchResults = null;
131                     DbSet<Business> tableToSearch = database.Businesses;
132                     switch (comparator)

```

```

133     {
134     case "|":
135         searchResults = tableToSearch.AsEnumerable().Where(i =>
136             i[selectedIndex].ToString().IndexOf(
137                 toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0);
138         break;
139
140     case "!=":
141         searchResults = tableToSearch.AsEnumerable().Where(i =>
142             i[selectedIndex].ToString().IndexOf(
143                 toSearchFor, StringComparison.OrdinalIgnoreCase) < 0);
144         break;
145
146     case "==":
147         if (baseObject is byte)
148             searchResults = tableToSearch.AsEnumerable().Where(i =>
149                 (byte)i[selectedIndex] ==
150                 toSearchForByte);
151         else if (baseObject is DateTime)
152             searchResults = tableToSearch.AsEnumerable().Where(i =>
153                 (DateTime)i[selectedIndex] ==
154                 toSearchForDateTime);
155         else if (baseObject is decimal)
156             searchResults = tableToSearch.AsEnumerable().Where(i =>
157                 (decimal)i[selectedIndex] ==
158                 toSearchForDecimal);
159         else
160             searchResults = tableToSearch.AsEnumerable().Where(i =>
161                 ((string)i[selectedIndex]).ToLower() == toSearchFor.
162                 ToLower());
163         break;
164
165     case "!=":
166         if (baseObject is byte)
167             searchResults = tableToSearch.AsEnumerable().Where(i =>
168                 (byte)i[selectedIndex] !=
169                 toSearchForByte);
170         else if (baseObject is DateTime)
171             searchResults = tableToSearch.AsEnumerable().Where(i =>
172                 (DateTime)i[selectedIndex] !=
173                 toSearchForDateTime);
174         else if (baseObject is decimal)
175             searchResults = tableToSearch.AsEnumerable().Where(i =>
176                 (decimal)i[selectedIndex] !=
177                 toSearchForDecimal);
178         else
179             searchResults = tableToSearch.AsEnumerable().Where(i =>
180                 ((string)i[selectedIndex]).ToLower() != toSearchFor.
181                 ToLower());
182         break;
183
184     case ">":
185         if (baseObject is byte)
186             searchResults = tableToSearch.AsEnumerable().Where(i =>
187                 (byte)i[selectedIndex] >
188                 toSearchForByte);
189         else if (baseObject is DateTime)
190             searchResults = tableToSearch.AsEnumerable().Where(i =>
191                 (DateTime)i[selectedIndex] >
192                 toSearchForDateTime);
193         else if (baseObject is decimal)
194             searchResults = tableToSearch.AsEnumerable().Where(i =>
195                 (decimal)i[selectedIndex] >
196                 toSearchForDecimal);
197         else
198             throw new InvalidOperationException(

```

```

199         "Comparator \">>\" cannot be applied to strings.");
200     break;
201
202     case "<":
203         if (baseObject is byte)
204             searchResults = tableToSearch.AsEnumerable().Where(i =>
205                 (byte)i[selectedIndex] <
206                 toSearchForByte);
207         else if (baseObject is DateTime)
208             searchResults = tableToSearch.AsEnumerable().Where(i =>
209                 (DateTime)i[selectedIndex] <
210                 toSearchForDateTime);
211         else if (baseObject is decimal)
212             searchResults = tableToSearch.AsEnumerable().Where(i =>
213                 (decimal)i[selectedIndex] <
214                 toSearchForDecimal);
215         else
216             throw new InvalidOperationException(
217                 "Comparator \"><\" cannot be applied to strings.");
218     break;
219
220     case ">=":
221         if (baseObject is byte)
222             searchResults = tableToSearch.AsEnumerable().Where(i =>
223                 (byte)i[selectedIndex] >=
224                 toSearchForByte);
225         else if (baseObject is DateTime)
226             searchResults = tableToSearch.AsEnumerable().Where(i =>
227                 (DateTime)i[selectedIndex] >=
228                 toSearchForDateTime);
229         else if (baseObject is decimal)
230             searchResults = tableToSearch.AsEnumerable().Where(i =>
231                 (decimal)i[selectedIndex] >=
232                 toSearchForDecimal);
233         else
234             throw new InvalidOperationException(
235                 "Comparator \">>=\\" cannot be applied to strings.");
236     break;
237
238     case "<=":
239         if (baseObject is byte)
240             searchResults = tableToSearch.AsEnumerable().Where(i =>
241                 (byte)i[selectedIndex] <=
242                 toSearchForByte);
243         else if (baseObject is DateTime)
244             searchResults = tableToSearch.AsEnumerable().Where(i =>
245                 (DateTime)i[selectedIndex] <=
246                 toSearchForDateTime);
247         else if (baseObject is decimal)
248             searchResults = tableToSearch.AsEnumerable().Where(i =>
249                 (decimal)i[selectedIndex] <=
250                 toSearchForDecimal);
251         else
252             throw new InvalidOperationException(
253                 "Comparator \"><=\\" cannot be applied to strings.");
254     break;
255
256     default:
257         throw new InvalidOperationException("Invalid comparator.");
258 }
259 resultCount = searchResults.Count();
260 gridViewToDisplay.DataSource = searchResults.ToList();
261 }
262 break;
263
264 case Global.Enums.ItemTypes.Park:

```

```

265 viewToDisplay = 1;
266 gViewToDisplay = gViewParkResults;
267 selectedIndex = ddlPark.SelectedIndex;
268 baseObject = database.Parks.First()[selectedIndex];
269 if (baseObject != null)
270 {
271     IEnumerable<Park> searchResults = null;
272     DbSet<Park> tableToSearch = database.Parks;
273     switch (comparator)
274     {
275         case "|":
276             searchResults = tableToSearch.AsEnumerable().Where(i =>
277                 i[selectedIndex].ToString().IndexOf(
278                     toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0);
279             break;
280
281         case "!=":
282             searchResults = tableToSearch.AsEnumerable().Where(i =>
283                 i[selectedIndex].ToString().IndexOf(
284                     toSearchFor, StringComparison.OrdinalIgnoreCase) < 0);
285             break;
286
287         case "==":
288             if (baseObject is byte)
289                 searchResults = tableToSearch.AsEnumerable().Where(i =>
290                     (byte)i[selectedIndex] ==
291                     toSearchForByte);
292             else if (baseObject is DateTime)
293                 searchResults = tableToSearch.AsEnumerable().Where(i =>
294                     (DateTime)i[selectedIndex] ==
295                     toSearchForDateTime);
296             else if (baseObject is decimal)
297                 searchResults = tableToSearch.AsEnumerable().Where(i =>
298                     (decimal)i[selectedIndex] ==
299                     toSearchForDecimal);
300             else
301                 searchResults = tableToSearch.AsEnumerable().Where(i =>
302                     ((string)i[selectedIndex]).ToLower() == toSearchFor.
303                     ToLower());
304             break;
305
306         case "!=":
307             if (baseObject is byte)
308                 searchResults = tableToSearch.AsEnumerable().Where(i =>
309                     (byte)i[selectedIndex] !=
310                     toSearchForByte);
311             else if (baseObject is DateTime)
312                 searchResults = tableToSearch.AsEnumerable().Where(i =>
313                     (DateTime)i[selectedIndex] !=
314                     toSearchForDateTime);
315             else if (baseObject is decimal)
316                 searchResults = tableToSearch.AsEnumerable().Where(i =>
317                     (decimal)i[selectedIndex] !=
318                     toSearchForDecimal);
319             else
320                 searchResults = tableToSearch.AsEnumerable().Where(i =>
321                     ((string)i[selectedIndex]).ToLower() != toSearchFor.
322                     ToLower());
323             break;
324
325         case ">":
326             if (baseObject is byte)
327                 searchResults = tableToSearch.AsEnumerable().Where(i =>
328                     (byte)i[selectedIndex] >
329                     toSearchForByte);
330             else if (baseObject is DateTime)

```

```

331         searchResults = tableToSearch.AsEnumerable().Where(i =>
332             (DateTime)i[selectedIndex] >
333             toSearchForDateTime);
334     else if (baseObject is decimal)
335         searchResults = tableToSearch.AsEnumerable().Where(i =>
336             (decimal)i[selectedIndex] >
337             toSearchForDecimal);
338     else
339         throw new InvalidOperationException(
340             "Comparator \">>\" cannot be applied to strings.");
341     break;
342
343     case "<":
344         if (baseObject is byte)
345             searchResults = tableToSearch.AsEnumerable().Where(i =>
346                 (byte)i[selectedIndex] <
347                 toSearchForByte);
348         else if (baseObject is DateTime)
349             searchResults = tableToSearch.AsEnumerable().Where(i =>
350                 (DateTime)i[selectedIndex] <
351                 toSearchForDateTime);
352         else if (baseObject is decimal)
353             searchResults = tableToSearch.AsEnumerable().Where(i =>
354                 (decimal)i[selectedIndex] <
355                 toSearchForDecimal);
356         else
357             throw new InvalidOperationException(
358                 "Comparator \"><\" cannot be applied to strings.");
359         break;
360
361     case ">=":
362         if (baseObject is byte)
363             searchResults = tableToSearch.AsEnumerable().Where(i =>
364                 (byte)i[selectedIndex] >=
365                 toSearchForByte);
366         else if (baseObject is DateTime)
367             searchResults = tableToSearch.AsEnumerable().Where(i =>
368                 (DateTime)i[selectedIndex] >=
369                 toSearchForDateTime);
370         else if (baseObject is decimal)
371             searchResults = tableToSearch.AsEnumerable().Where(i =>
372                 (decimal)i[selectedIndex] >=
373                 toSearchForDecimal);
374         else
375             throw new InvalidOperationException(
376                 "Comparator \">>=\" cannot be applied to strings.");
377         break;
378
379     case "<=":
380         if (baseObject is byte)
381             searchResults = tableToSearch.AsEnumerable().Where(i =>
382                 (byte)i[selectedIndex] <=
383                 toSearchForByte);
384         else if (baseObject is DateTime)
385             searchResults = tableToSearch.AsEnumerable().Where(i =>
386                 (DateTime)i[selectedIndex] <=
387                 toSearchForDateTime);
388         else if (baseObject is decimal)
389             searchResults = tableToSearch.AsEnumerable().Where(i =>
390                 (decimal)i[selectedIndex] <=
391                 toSearchForDecimal);
392         else
393             throw new InvalidOperationException(
394                 "Comparator \"><=\" cannot be applied to strings.");
395         break;
396

```

```

397         default:
398             throw new InvalidOperationException("Invalid comparator.");
399     }
400     resultCount = searchResults.Count();
401     gridViewToDisplay.DataSource = searchResults.ToList();
402 }
403 break;
404
405 case Global.Enums.ItemTypes.PublicFacility:
406     viewToDisplay = 2;
407     gridViewToDisplay = gridViewPublicFacilityResults;
408     selectedIndex = ddlPublicFacility.SelectedIndex;
409     baseObject = database.PublicFacilities.First()[selectedIndex];
410     if (baseObject != null)
411     {
412         IEnumerable<PublicFacility> searchResults = null;
413         DbSet<PublicFacility> tableToSearch = database.PublicFacilities;
414         switch (comparator)
415         {
416             case "|":
417                 searchResults = tableToSearch.AsEnumerable().Where(i =>
418                     i[selectedIndex].ToString().IndexOf(
419                         toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0);
420                 break;
421
422             case "!!":
423                 searchResults = tableToSearch.AsEnumerable().Where(i =>
424                     i[selectedIndex].ToString().IndexOf(
425                         toSearchFor, StringComparison.OrdinalIgnoreCase) < 0);
426                 break;
427
428             case "==" :
429                 if (baseObject is byte)
430                     searchResults = tableToSearch.AsEnumerable().Where(i =>
431                         (byte)i[selectedIndex] ==
432                         toSearchForByte);
433                 else if (baseObject is DateTime)
434                     searchResults = tableToSearch.AsEnumerable().Where(i =>
435                         (DateTime)i[selectedIndex] ==
436                         toSearchForDateTime);
437                 else if (baseObject is decimal)
438                     searchResults = tableToSearch.AsEnumerable().Where(i =>
439                         (decimal)i[selectedIndex] ==
440                         toSearchForDecimal);
441                 else
442                     searchResults = tableToSearch.AsEnumerable().Where(i =>
443                         ((string)i[selectedIndex]).ToLower() == toSearchFor.
444                         ToLower());
445                 break;
446
447             case "!=":
448                 if (baseObject is byte)
449                     searchResults = tableToSearch.AsEnumerable().Where(i =>
450                         (byte)i[selectedIndex] !=
451                         toSearchForByte);
452                 else if (baseObject is DateTime)
453                     searchResults = tableToSearch.AsEnumerable().Where(i =>
454                         (DateTime)i[selectedIndex] !=
455                         toSearchForDateTime);
456                 else if (baseObject is decimal)
457                     searchResults = tableToSearch.AsEnumerable().Where(i =>
458                         (decimal)i[selectedIndex] !=
459                         toSearchForDecimal);
460                 else
461                     searchResults = tableToSearch.AsEnumerable().Where(i =>
462                         ((string)i[selectedIndex]).ToLower() != toSearchFor.

```

```

463         ToLower());
464     break;
465
466     case ">":
467         if (baseObject is byte)
468             searchResults = tableToSearch.AsEnumerable().Where(i =>
469                 (byte)i[selectedIndex] >
470                 toSearchForByte);
471         else if (baseObject is DateTime)
472             searchResults = tableToSearch.AsEnumerable().Where(i =>
473                 (DateTime)i[selectedIndex] >
474                 toSearchForDateTime);
475         else if (baseObject is decimal)
476             searchResults = tableToSearch.AsEnumerable().Where(i =>
477                 (decimal)i[selectedIndex] >
478                 toSearchForDecimal);
479         else
480             throw new InvalidOperationException(
481                 "Comparator \">>\<" cannot be applied to strings.");
482     break;
483
484     case "<":
485         if (baseObject is byte)
486             searchResults = tableToSearch.AsEnumerable().Where(i =>
487                 (byte)i[selectedIndex] <
488                 toSearchForByte);
489         else if (baseObject is DateTime)
490             searchResults = tableToSearch.AsEnumerable().Where(i =>
491                 (DateTime)i[selectedIndex] <
492                 toSearchForDateTime);
493         else if (baseObject is decimal)
494             searchResults = tableToSearch.AsEnumerable().Where(i =>
495                 (decimal)i[selectedIndex] <
496                 toSearchForDecimal);
497         else
498             throw new InvalidOperationException(
499                 "Comparator \"><\<" cannot be applied to strings.");
500     break;
501
502     case ">=":
503         if (baseObject is byte)
504             searchResults = tableToSearch.AsEnumerable().Where(i =>
505                 (byte)i[selectedIndex] >=
506                 toSearchForByte);
507         else if (baseObject is DateTime)
508             searchResults = tableToSearch.AsEnumerable().Where(i =>
509                 (DateTime)i[selectedIndex] >=
510                 toSearchForDateTime);
511         else if (baseObject is decimal)
512             searchResults = tableToSearch.AsEnumerable().Where(i =>
513                 (decimal)i[selectedIndex] >=
514                 toSearchForDecimal);
515         else
516             throw new InvalidOperationException(
517                 "Comparator \">>=\<" cannot be applied to strings.");
518     break;
519
520     case "<=":
521         if (baseObject is byte)
522             searchResults = tableToSearch.AsEnumerable().Where(i =>
523                 (byte)i[selectedIndex] <=
524                 toSearchForByte);
525         else if (baseObject is DateTime)
526             searchResults = tableToSearch.AsEnumerable().Where(i =>
527                 (DateTime)i[selectedIndex] <=
528                 toSearchForDateTime);

```



```

529         else if (baseObject is decimal)
530             searchResults = tableToSearch.AsEnumerable().Where(i =>
531                 (decimal)i[selectedIndex] <=
532                 toSearchForDecimal);
533         else
534             throw new InvalidOperationException(
535                 "Comparator \"<=\" cannot be applied to strings.");
536         break;
537
538         default:
539             throw new InvalidOperationException("Invalid comparator.");
540     }
541     resultCount = searchResults.Count();
542     gridViewToDisplay.DataSource = searchResults.ToList();
543 }
544 break;
545
546     default:
547         throw new InvalidOperationException(
548             "Invalid item type dropdown value.");
549     }
550     lblResult.Text = string.Format("{0} result{1} found.", resultCount,
551         resultCount == 1 ? "" : "s");
552     lblResult.Visible = true;
553     mViewSearchResults.ActiveViewIndex = viewToDisplay;
554     gridViewToDisplay.DataBind();
555 }
556 }
557 catch (Exception ex)
558 {
559     lblError.Text = "Error: " + ex.Message;
560     lblError.Visible = true;
561 }
562 }
563
564 /*-----
565 * Name:     ddlBusiness_SelectedIndexChanged
566 * Type:     Event Handler Method
567 * Purpose:  Handles the change of visibility on controls when the user chooses a business
568 *           field.
569 * Input:    object sender, holds a reference to the object that raised this event.
570 * Input:    EventArgs e, holds data related to this event.
571 * Output:   Nothing.
572 -----*/
573 protected void ddlBusiness_SelectedIndexChanged(object sender, EventArgs e)
574 {
575     // Hide things until needed.
576     lblError.Visible = false;
577     lblResult.Visible = false;
578
579     ddlComparatorsStrings.Visible = false;
580     ddlComparatorsStrings.SelectedIndex = 0;
581     ddlComparatorsNotStrings.Visible = false;
582     ddlComparatorsNotStrings.SelectedIndex = 0;
583
584     txtSearch.Visible = false;
585     txtSearch.Text = "";
586     btnSearch.Visible = false;
587
588     mViewSearchResults.ActiveViewIndex = -1;
589
590     // Display the needed control.
591     switch ((Global.Enums.BusinessFields)ddlBusiness.SelectedIndex)
592     {
593         case Global.Enums.BusinessFields.Name:
594         case Global.Enums.BusinessFields.Type:

```

```

595         case Global.Enums.BusinessFields.StreetAddress:
596         case Global.Enums.BusinessFields.City:
597         case Global.Enums.BusinessFields.State:
598         case Global.Enums.BusinessFields.Zip:
599         case Global.Enums.BusinessFields.Phone:
600         case Global.Enums.BusinessFields.LicenseNumber:
601         case Global.Enums.BusinessFields.LicenseStatus:
602         case Global.Enums.BusinessFields.CouncilDistrict:
603             ddlComparatorsStrings.Visible = true;
604             break;
605
606         case Global.Enums.BusinessFields.Latitude:
607         case Global.Enums.BusinessFields.Longitude:
608         case Global.Enums.BusinessFields.LicenseIssueDate:
609         case Global.Enums.BusinessFields.LicenseExpirDate:
610             ddlComparatorsNotStrings.Visible = true;
611             break;
612
613         default:
614             ddlComparatorsStrings.Visible = false;
615             ddlComparatorsNotStrings.Visible = false;
616             break;
617     }
618 }
619
620 /*-----
621  * Name:      ddlComparatorsNotStrings_SelectedIndexChanged
622  * Type:      Event Handler Method
623  * Purpose:   Handles the change of visibility on controls when the user chooses what type of
624  *            comparison to make.
625  * Input:     object sender, holds a reference to the object that raised this event.
626  * Input:     EventArgs e, holds data related to this event.
627  * Output:    Nothing.
628  -----*/
629 protected void ddlComparatorsNotStrings_SelectedIndexChanged(object sender, EventArgs e)
630 {
631     // Hide things until needed.
632     lblError.Visible = false;
633     lblResult.Visible = false;
634
635     txtSearch.Visible = false;
636     txtSearch.Text = "";
637     btnSearch.Visible = false;
638
639     mViewSearchResults.ActiveViewIndex = -1;
640
641     // Display the needed control.
642     switch ((Global.Enums.ComparatorsNotStrings)ddlComparatorsNotStrings.SelectedIndex)
643     {
644         case Global.Enums.ComparatorsNotStrings.Contain:
645         case Global.Enums.ComparatorsNotStrings.NotContain:
646         case Global.Enums.ComparatorsNotStrings.Equal:
647         case Global.Enums.ComparatorsNotStrings.NotEqual:
648         case Global.Enums.ComparatorsNotStrings.Greater:
649         case Global.Enums.ComparatorsNotStrings.Less:
650         case Global.Enums.ComparatorsNotStrings.GreaterEqual:
651         case Global.Enums.ComparatorsNotStrings.LessEqual:
652             txtSearch.Visible = true;
653             btnSearch.Visible = true;
654             break;
655
656         default:
657             break;
658     }
659 }
660

```

```

661  /*-----
662  * Name:      ddlComparatorsStrings_SelectedIndexChanged
663  * Type:      Event Handler Method
664  * Purpose:   Handles the change of visibility on controls when the user chooses what type of
665  *            comparison to make.
666  * Input:     object sender, holds a reference to the object that raised this event.
667  * Input:     EventArgs e, holds data related to this event.
668  * Output:    Nothing.
669  -----*/
670  protected void ddlComparatorsStrings_SelectedIndexChanged(object sender, EventArgs e)
671  {
672      // Hide things until needed.
673      lblError.Visible = false;
674      lblResult.Visible = false;
675
676      txtSearch.Visible = false;
677      txtSearch.Text = "";
678      btnSearch.Visible = false;
679
680      mViewSearchResults.ActiveViewIndex = -1;
681
682      // Display the needed control.
683      switch ((Global.Enums.ComparatorsStrings)ddlComparatorsStrings.SelectedIndex)
684      {
685          case Global.Enums.ComparatorsStrings.Contain:
686          case Global.Enums.ComparatorsStrings.NotContain:
687          case Global.Enums.ComparatorsStrings.Equal:
688          case Global.Enums.ComparatorsStrings.NotEqual:
689              txtSearch.Visible = true;
690              btnSearch.Visible = true;
691              break;
692
693          default:
694              break;
695      }
696  }
697
698  /*-----
699  * Name:      ddlItemType_SelectedIndexChanged
700  * Type:      Event Handler Method
701  * Purpose:   Handles showing and hiding the various controls to allow a user to search the
702  *            database.
703  * Input:     object sender, holds a reference to the object that raised this event.
704  * Input:     EventArgs e, holds data related to this event.
705  * Output:    Nothing.
706  -----*/
707  protected void ddlItemType_SelectedIndexChanged(object sender, EventArgs e)
708  {
709      // Hide things until needed.
710      lblError.Visible = false;
711      lblResult.Visible = false;
712
713      ddlBusiness.Visible = false;
714      ddlBusiness.SelectedIndex = 0;
715      ddlPark.Visible = false;
716      ddlPark.SelectedIndex = 0;
717      ddlPublicFacility.Visible = false;
718      ddlPublicFacility.SelectedIndex = 0;
719      ddlComparatorsStrings.Visible = false;
720      ddlComparatorsStrings.SelectedIndex = 0;
721      ddlComparatorsNotStrings.Visible = false;
722      ddlComparatorsNotStrings.SelectedIndex = 0;
723
724      txtSearch.Visible = false;
725      txtSearch.Text = "";
726      btnSearch.Visible = false;

```

```

727         mViewSearchResults.ActiveViewIndex = -1;
728
729         // Display the needed control.
730         switch ((Global.Enums.ItemTypes)ddlItemType.SelectedIndex)
731         {
732             case Global.Enums.ItemTypes.Business:
733                 ddlBusiness.Visible = true;
734                 break;
735
736             case Global.Enums.ItemTypes.Park:
737                 ddlPark.Visible = true;
738                 break;
739
740             case Global.Enums.ItemTypes.PublicFacility:
741                 ddlPublicFacility.Visible = true;
742                 break;
743
744             default:
745                 break;
746         }
747     }
748 }
749
750 /*-----
751  * Name:     ddlPark_SelectedIndexChanged
752  * Type:     Event Handler Method
753  * Purpose:  Handles the change of visibility on controls when the user chooses a park field.
754  * Input:    object sender, holds a reference to the object that raised this event.
755  * Input:    EventArgs e, holds data related to this event.
756  * Output:   Nothing.
757  -----*/
758 protected void ddlPark_SelectedIndexChanged(object sender, EventArgs e)
759 {
760     // Hide things until needed.
761     lblError.Visible = false;
762     lblResult.Visible = false;
763
764     ddlComparatorsStrings.Visible = false;
765     ddlComparatorsStrings.SelectedIndex = 0;
766     ddlComparatorsNotStrings.Visible = false;
767     ddlComparatorsNotStrings.SelectedIndex = 0;
768
769     txtSearch.Visible = false;
770     txtSearch.Text = "";
771     btnSearch.Visible = false;
772
773     mViewSearchResults.ActiveViewIndex = -1;
774
775     // Display the needed control.
776     switch ((Global.Enums.ParkFields)ddlPark.SelectedIndex)
777     {
778         case Global.Enums.ParkFields.Name:
779         case Global.Enums.ParkFields.Type:
780         case Global.Enums.ParkFields.StreetAddress:
781         case Global.Enums.ParkFields.City:
782         case Global.Enums.ParkFields.State:
783         case Global.Enums.ParkFields.Zip:
784         case Global.Enums.ParkFields.Phone:
785             ddlComparatorsStrings.Visible = true;
786             break;
787
788         case Global.Enums.ParkFields.Latitude:
789         case Global.Enums.ParkFields.Longitude:
790         case Global.Enums.ParkFields.FeatureBaseball:
791         case Global.Enums.ParkFields.FeatureBasketball:
792         case Global.Enums.ParkFields.FeatureGolf:

```

```

793         case Global.Enums.ParkFields.FeatureLargeMPField:
794         case Global.Enums.ParkFields.FeatureTennis:
795         case Global.Enums.ParkFields.FeatureVolleyball:
796             ddlComparatorsNotStrings.Visible = true;
797             break;
798
799         default:
800             break;
801     }
802 }
803
804 /*-----*/
805 * Name:     ddlPublicFacility_SelectedIndexChanged
806 * Type:     Event Handler Method
807 * Purpose:  Handles the change of visibility on controls when the user chooses a public
808 *           facility field.
809 * Input:    object sender, holds a reference to the object that raised this event.
810 * Input:    EventArgs e, holds data related to this event.
811 * Output:   Nothing.
812 /*-----*/
813 protected void ddlPublicFacility_SelectedIndexChanged(object sender, EventArgs e)
814 {
815     // Hide things until needed.
816     lblError.Visible = false;
817     lblResult.Visible = false;
818
819     ddlComparatorsStrings.Visible = false;
820     ddlComparatorsStrings.SelectedIndex = 0;
821     ddlComparatorsNotStrings.Visible = false;
822     ddlComparatorsNotStrings.SelectedIndex = 0;
823
824     txtSearch.Visible = false;
825     txtSearch.Text = "";
826     btnSearch.Visible = false;
827
828     mViewSearchResults.ActiveViewIndex = -1;
829
830     // Display the needed control.
831     switch ((Global.Enums.PublicFacilityFields)ddlPublicFacility.SelectedIndex)
832     {
833         case Global.Enums.PublicFacilityFields.Name:
834         case Global.Enums.PublicFacilityFields.Type:
835         case Global.Enums.PublicFacilityFields.StreetAddress:
836         case Global.Enums.PublicFacilityFields.City:
837         case Global.Enums.PublicFacilityFields.State:
838         case Global.Enums.PublicFacilityFields.Zip:
839         case Global.Enums.PublicFacilityFields.Phone:
840             ddlComparatorsStrings.Visible = true;
841             break;
842
843         case Global.Enums.PublicFacilityFields.Latitude:
844         case Global.Enums.PublicFacilityFields.Longitude:
845             ddlComparatorsNotStrings.Visible = true;
846             break;
847
848         default:
849             break;
850     }
851 }
852 }
853 }

```