```
1  /*--------------------------------------------------------------------------------------------
2   * Author:      Dan Cassidy
3   * Date:        2015-06-23
4   * Assignment:  cView-P4
5   * Source File: Global.asax.cs
6   * Language:    C#
7   * Course:      CSCI-C 490, C# Programming, MoWe 08:00
8   * Purpose:     Code-behind file for Global.asax.  Contains things that should be globally
9   *              accessible, such as enums, exceptions, and strings.
10  --------------------------------------------------------------------------------------------*/
11
12  using System;
13  using System.Collections.Generic;
14  using System.Linq;
15  using System.Web;
16  using System.Web.Security;
17  using System.Web.SessionState;
18
19  namespace cView_P4_DanCassidy
20  {
21      public class Global : System.Web.HttpApplication
22      {
23          public static class Enums
24          {
25              /*--------------------------------------------------------------------------------
26               * Name:    BusinessFields
27               * Type:    Enum
28               * Purpose: Represents the different fields present in the Business class.
29              --------------------------------------------------------------------------------*/
30              public enum BusinessFields
31              {
32                  Name = 1,
33                  Type,
34                  StreetAddress,
35                  City,
36                  State,
37                  Zip,
38                  Latitude,
39                  Longitude,
40                  Phone,
41                  LicenseNumber,
42                  LicenseIssueDate,
43                  LicenseExpirDate,
44                  LicenseStatus,
45                  CouncilDistrict
46              }
47
48              /*--------------------------------------------------------------------------------
49               * Name:    ComparatorsNotStrings
50               * Type:    Enum
51               * Purpose: Represents the possible comparators available for use on non-strings.
52              --------------------------------------------------------------------------------*/
53              public enum ComparatorsNotStrings
54              {
55                  Contain = 1,
56                  NotContain,
57                  Equal,
58                  NotEqual,
59                  Greater,
60                  Less,
61                  GreaterEqual,
62                  LessEqual
63              }
64
65              /*--------------------------------------------------------------------------------
66               * Name:    ComparatorsStrings
```

```
67              * Type:    Enum
68              * Purpose: Represents the possible comparators available for use on strings.
69             ----------------------------------------------------------------------------------------*/
70             public enum ComparatorsStrings
71             {
72                 Contain = 1,
73                 NotContain,
74                 Equal,
75                 NotEqual
76             }
77
78             /*----------------------------------------------------------------------------------------
79              * Name:    ItemTypes
80              * Type:    Enum
81              * Purpose: Represents the different types of items.
82             ----------------------------------------------------------------------------------------*/
83             public enum ItemTypes
84             {
85                 Business = 1,
86                 Park,
87                 PublicFacility,
88             }
89
90             /*----------------------------------------------------------------------------------------
91              * Name:    ParkFields
92              * Type:    Enum
93              * Purpose: Represents the different fields present in the Park class.
94             ----------------------------------------------------------------------------------------*/
95             public enum ParkFields
96             {
97                 Name = 1,
98                 Type,
99                 StreetAddress,
100                City,
101                State,
102                Zip,
103                Latitude,
104                Longitude,
105                Phone,
106                FeatureBaseball,
107                FeatureBasketball,
108                FeatureGolf,
109                FeatureLargeMPField,
110                FeatureTennis,
111                FeatureVolleyball
112            }
113
114            /*----------------------------------------------------------------------------------------
115             * Name:    PublicFacilityFields
116             * Type:    Enum
117             * Purpose: Represents the different fiels present in the Public Facility class.
118            ----------------------------------------------------------------------------------------*/
119            public enum PublicFacilityFields
120            {
121                Name = 1,
122                Type,
123                StreetAddress,
124                City,
125                State,
126                Zip,
127                Latitude,
128                Longitude,
129                Phone
130            }
131
132        }
```

```
133
134         public class Exceptions
135         {
136             /*-----------------------------------------------------------------------------------
137              * Name:    DuplicatePKException
138              * Type:    Exception
139              * Purpose: Intended to describe a situation where an object with a duplicate primary
140              *          key attempted to be inserted into a primary keyed data structure.
141             -----------------------------------------------------------------------------------*/
142             [Serializable]
143             public class DuplicatePKException : Exception
144             {
145                 public DuplicatePKException() { }
146
147                 public DuplicatePKException(string message)
148                     : base(message) { }
149
150                 public DuplicatePKException(string keyName, object keyValue)
151                     : base(string.Format("An item already exists with a {0} of {1}.",
152                           keyName, keyValue)) { }
153
154                 public DuplicatePKException(string message, Exception inner)
155                     : base(message, inner) { }
156
157                 protected DuplicatePKException(
158                   System.Runtime.Serialization.SerializationInfo info,
159                   System.Runtime.Serialization.StreamingContext context)
160                     : base(info, context) { }
161             }
162
163             /*-----------------------------------------------------------------------------------
164              * Name:    EmptyOrNullPKException
165              * Type:    Exception
166              * Purpose: Inteded to describe a situation where an object with an empty or null
167              *          primary key attempted to be inserted into a primary keyed data structure.
168             -----------------------------------------------------------------------------------*/
169             [Serializable]
170             public class EmptyOrNullPKException : Exception
171             {
172                 public EmptyOrNullPKException() { }
173
174                 public EmptyOrNullPKException(string keyName)
175                     : base(string.Format("{0} cannot be empty or null.", keyName)) { }
176
177                 public EmptyOrNullPKException(string message, Exception inner)
178                     : base(message, inner) { }
179
180                 protected EmptyOrNullPKException(
181                   System.Runtime.Serialization.SerializationInfo info,
182                   System.Runtime.Serialization.StreamingContext context)
183                     : base(info, context) { }
184             }
185         }
186
187         /*-----------------------------------------------------------------------------------
188          * Name:    Strings
189          * Type:    Class
190          * Purpose: Contains common strings used throughout the application in a centrally-managed
191          *          location.
192         -----------------------------------------------------------------------------------*/
193         public static class Strings
194         {
195             public const string Separator = ":";
196
197             public const string BusinessName = "Business Name";
198             public const string ParkName = "Park Name";
```

```
199             public const string PublicFacilityName = "Public Facility Name";
200
201             public const string BusinessType = "Type of Business";
202             public const string ParkType = "Type of Park";
203             public const string PublicFacilityType = "Type of Public Facility";
204
205             public const string BusinessKey = "License Number";
206             public const string ParkKey = ParkName;
207             public const string PublicFacilityKey = PublicFacilityName;
208         }
209
210     }
211 }
```