

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.DataAnnotations;
4 using System.ComponentModel.DataAnnotations.Schema;
5 using System.Linq;
6 using System.Web;
7 using System.Web.Mvc;
8
9 namespace AdventureLog.Models
10 {
11     #region Adventure Information
12
13     public class Adventure
14     {
15         [Key]
16         public long Adventure_PK { get; set; }
17
18         [Required, StringLength(256, ErrorMessage = "Name cannot be longer
19             than 256 characters."), Display(Name="Adventure Name")]
20         public string Name { get; set; }
21
22         [MaxLength(256, ErrorMessage = "System Name cannot be longer than 256
23             characters."), Display(Name = "System")]
24         public string SystemName { get; set; }
25
26         [MaxLength(4000, ErrorMessage = "Short Description cannot be longer
27             than 4000 characters."), Display(Name = "Summary")]
28         public string ShortDescription { get; set; }
29
30         [AllowHtml]
31         public string Description { get; set; }
32
33         [AllowHtml, Display(Name = "Gamemaster Notes (Cannot be seen by
34             players)")]
35         public string GamemasterNotes { get; set; }
36
37         [MaxLength(100, ErrorMessage = "Invite Password must be less than 100
38             characters."), Display(Name = "Invite Password")]
39         public string InvitePassword { get; set; }
40
41         public bool IsSecured { get; set; }
42
43         public bool IsPublic { get; set; }
44
45         [Required]
46         public bool IsActive { get; set; }
47
48         [Required]
49         public DateTime CreatedDate { get; set; }
50
51         [Required]
52         public DateTime LastModifiedDate { get; set; }
53
54         [Required]
55         public string LastModifiedUser { get; set; }
56     }
57 }
```

```
52     [InverseProperty("Adventure")]
53     public virtual ICollection<Player> Players { get; set; }
54
55     [InverseProperty("Adventure")]
56     public virtual ICollection<Item> Items { get; set; }
57
58     [InverseProperty("Adventure")]
59     public virtual ICollection<AdventureNote> AdventureNotes { get; set; }
60 }
61
62 public class Item
63 {
64     [Key]
65     public long Item_PK { get; set; }
66
67     [Required]
68     public long Adventure_PK { get; set; }
69
70     public long? ParentItem_PK { get; set; }
71
72     [Required, StringLength(256, ErrorMessage = "Name cannot be longer
73         than 256 characters."), Display(Name = "Item Name")]
74     public string Name { get; set; }
75
76     [MaxLength(4000, ErrorMessage = "Short Description cannot be longer
77         than 4000 characters."), Display(Name = "Summary")]
78     public string ShortDescription { get; set; }
79
80     [AllowHtml]
81     public string Description { get; set; }
82
83     [AllowHtml, Display(Name = "Gamemaster Notes (Cannot be seen by
84         players)")]
85     public string GamemasterNotes { get; set; }
86
87     public string MapFileName { get; set; }
88
89     [Required]
90     public bool IsActive { get; set; }
91
92     [Required]
93     public DateTime CreatedDate { get; set; }
94
95     [Required]
96     public DateTime LastModifiedDate { get; set; }
97
98     [Required]
99     public string LastModifiedUser { get; set; }
100
101     [ForeignKey("Adventure_PK")]
102     public Adventure Adventure { get; set; }
103
104     [ForeignKey("ParentItem_PK")]
105     public Item ParentItem { get; set; }
106
107     [InverseProperty("Item")]
108 }
```

```
105     public virtual ICollection<ItemNote> ItemNotes { get; set; }
106
107     [InverseProperty("ParentItem")]
108     public virtual ICollection<Item> ChildItems { get; set; }
109 }
110
111 #region Player
112
113 public class Player
114 {
115     [Key]
116     public long Player_PK { get; set; }
117
118     [Required]
119     public string UserId_PK { get; set; }
120
121     [Required]
122     public long Adventure_PK { get; set; }
123
124     [Required]
125     public long PlayerRole_PK { get; set; }
126
127     [Required]
128     public bool IsActive { get; set; }
129
130     [ForeignKey("UserId_PK")]
131     public ApplicationUser ApplicationUser { get; set; }
132
133     [ForeignKey("Adventure_PK")]
134     public Adventure Adventure { get; set; }
135
136     [ForeignKey("PlayerRole_PK")]
137     public PlayerRole PlayerRole { get; set; }
138 }
139
140 public class PlayerRole
141 {
142     [Key]
143     public long PlayerRole_PK { get; set; }
144
145     [Required, MaxLength(128)]
146     public string RoleName { get; set; }
147
148     [Required, MaxLength(256)]
149     public string RoleDescription { get; set; }
150
151     [Required]
152     public DateTime CreatedDate { get; set; }
153
154     [Required]
155     public DateTime LastModifiedDate { get; set; }
156
157     [Required]
158     public string LastModifiedUser { get; set; }
159
160     public enum PlayerRoleKey
```

```
161     {
162         Player = 1,
163         Gamemaster = 2
164     }
165 }
166
167 #endregion
168 #region Hotspot Information
169
170 /* Removed as the Map Feautre is being postponed for a later version.
171 public class Hotspot
172 {
173     [Key]
174     public long Hotspot_PK { get; set; }
175
176     [Required]
177     public bool IsActive { get; set; }
178
179     [Required]
180     public DateTime CreatedDate { get; set; }
181
182     [Required]
183     public DateTime LastModifiedDate { get; set; }
184
185     [Required]
186     public string LastModifiedUser { get; set; }
187
188     [InverseProperty("Hotspot")]
189     public virtual ICollection<Coordinate> Coordinates { get; set; }
190 }
191
192 public class Coordinate
193 {
194     [Key]
195     public long Coordinate_PK { get; set; }
196
197     [Required]
198     public long Hotspot_PK { get; set; }
199
200     [Required]
201     public int XCoord { get; set; }
202
203     [Required]
204     public int YCoord { get; set; }
205
206     [Required]
207     public DateTime CreatedDate { get; set; }
208
209     [Required]
210     public DateTime LastModifiedDate { get; set; }
211
212     [Required]
213     public string LastModifiedUser { get; set; }
214
215     [ForeignKey("Hotspot_PK")]
216     public Hotspot Hotspot { get; set; }
```

```
217     }
218     */
219     #endregion
220
221     #region Note Information
222     public class ItemNote
223     {
224         [Key]
225         public long ItemNote_PK { get; set; }
226
227         [Required]
228         public long Item_PK { get; set; }
229
230         [Required]
231         public string UserId_PK { get; set; }
232
233         public long? ParentItemNote_PK { get; set; }
234
235         [Required]
236         [AllowHtml]
237         public string Text { get; set; }
238
239         [Required]
240         public bool IsActive { get; set; }
241
242         [Required]
243         public DateTime CreatedDate { get; set; }
244
245         [Required]
246         public DateTime LastModifiedDate { get; set; }
247
248         [Required]
249         public string LastModifiedUser { get; set; }
250
251         [ForeignKey("Item_PK")]
252         public Item Item { get; set; }
253
254         [ForeignKey("UserId_PK")]
255         public ApplicationUser ApplicationUser { get; set; }
256
257         [ForeignKey("ParentItemNote_PK")]
258         public ItemNote ParentItemNote { get; set; }
259
260         [InverseProperty("ParentItemNote")]
261         public virtual ICollection<ItemNote> ChildNotes { get; set; }
262     }
263
264     public class AdventureNote
265     {
266         [Key]
267         public long AdventureNote_PK { get; set; }
268
269         [Required]
270         public long Adventure_PK { get; set; }
271
272         [Required]
```

```
273     public string UserId_PK { get; set; }
274
275     public long? ParentAdventureNote_PK { get; set; }
276
277     [Required]
278     [AllowHtml]
279     public string Text { get; set; }
280
281     [Required]
282     public bool IsActive { get; set; }
283
284     [Required]
285     public DateTime CreatedDate { get; set; }
286
287     [Required]
288     public DateTime LastModifiedDate { get; set; }
289
290     [Required]
291     public string LastModifiedUser { get; set; }
292
293     [ForeignKey("Adventure_PK")]
294     public Adventure Adventure { get; set; }
295
296     [ForeignKey("UserId_PK")]
297     public ApplicationUser ApplicationUser { get; set; }
298
299     [ForeignKey("ParentAdventureNote_PK")]
300     public AdventureNote ParentAdventureNote {get;set;}
301
302     [InverseProperty("ParentAdventureNote")]
303     public virtual ICollection<AdventureNote> ChildNotes { get; set; }
304 }
305 #endregion
306
307 #endregion
308 }
```