

Database Model	2
Code First Migration Seed Database	8
Application Database Context Class	11
Adventure Home Controller	12
Adventure Home Index	35
Adventure Create View	37
Adventure Detail View	39
Adventure Edit View	50
Item Create View	56
Item Detail view	58
Item Edit View	66
Search Results View	68

```
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 }
```

```
1 namespace AdventureLog.Migrations
2 {
3     using AdventureLog.Models;
4     using Microsoft.AspNet.Identity;
5     using Microsoft.AspNet.Identity.EntityFramework;
6     using System;
7     using System.Data.Entity;
8     using System.Data.Entity.Migrations;
9     using System.Linq;
10
11     internal sealed class Configuration : DbMigrationsConfiguration<AdventureLog.Models.ApplicationDbContext> ↗
12     {
13         public Configuration()
14         {
15             AutomaticMigrationsEnabled = true;
16             AutomaticMigrationDataLossAllowed = true;
17         }
18
19         protected override void Seed(AdventureLog.Models.ApplicationDbContext context ↗)
20         {
21             // This method will be called after migrating to the latest version. ↗
22             SeedRoles(context);
23             SeedAdminUser(context);
24             SeedPlayerRole(context);
25         }
26
27         /// <summary>
28         /// Adds Asp.Net user Roles
29         /// </summary>
30         /// <param name="context"></param>
31         private void SeedRoles(ApplicationDbContext context)
32         {
33             if (!context.Roles.Any(r => r.Name == "Admin"))
34             {
35                 var store = new RoleStore<IdentityRole>(context);
36                 var manager = new RoleManager<IdentityRole>(store);
37                 var role = new IdentityRole
38                 {
39                     Name = "Admin"
40                 };
41
42                 manager.Create(role);
43
44             }
45         }
46
47         /// <summary>
48         /// test@gmail.com
49         /// Test123456!
50         /// </summary>
51         /// <param name="context"></param>
52         private void SeedAdminUser(ApplicationDbContext context)
53         {
```



```
54         if (!context.Users.Any(u => u.UserName == "AdventureLogAdmin"))
55         {
56             var store = new UserStore<ApplicationUser>(context);
57             var manager = new UserManager<ApplicationUser>(store);
58             var user = new ApplicationUser
59             {
60                 UserName = "AdventureLogAdmin",
61                 Email = "adventurequestlog@gmail.com",
62                 LastLogin = DateTime.Now,
63                 JoinDate = DateTime.Now
64             };
65
66             manager.Create(user, "Test123456!");
67             manager.AddToRole(user.Id, "Admin");
68         }
69     }
70
71     private void SeedPlayerRole(ApplicationDbContext context)
72     {
73         if (!context.PlayerRoles.Any(r => r.PlayerRole_PK == (int)
74             PlayerRole.PlayerRoleKey.Player))
75         {
76             var role = new PlayerRole()
77             {
78                 PlayerRole_PK = (int)PlayerRole.PlayerRoleKey.Player,
79                 RoleName = "Player",
80                 RoleDescription = "Player in a game, can comment and
81                 view",
82                 CreatedDate = DateTime.Now,
83                 LastModifiedDate = DateTime.Now,
84                 LastModifiedUser = "SysAdmin"
85             };
86
87             context.PlayerRoles.Add(role);
88         }
89
90         if (!context.PlayerRoles.Any(r => r.PlayerRole_PK == (int)
91             PlayerRole.PlayerRoleKey.Gamemaster))
92         {
93             var role = new PlayerRole()
94             {
95                 PlayerRole_PK = (int)PlayerRole.PlayerRoleKey.Gamemaster,
96                 RoleName = "Gamemaster",
97                 RoleDescription = "Gamemaster of a game, can control all
98                 features of the adventure",
99                 CreatedDate = DateTime.Now,
100                 LastModifiedDate = DateTime.Now,
101                 LastModifiedUser = "SysAdmin"
102             };
103
104             context.PlayerRoles.Add(role);
105         }
106
107         context.SaveChanges();
108     }
109 }
```

106 }

107

```
1 using Microsoft.AspNet.Identity.EntityFramework;
2 using System;
3 using System.Collections.Generic;
4 using System.Data.Entity;
5 using System.Linq;
6 using System.Web;
7
8 namespace AdventureLog.Models
9 {
10     public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
11     {
12         public DbSet<Adventure> Adventures { get; set; }
13         public DbSet<Player> Players { get; set; }
14         public DbSet<PlayerRole> PlayerRoles { get; set; }
15         //public DbSet<Hotspot> Hotspots { get; set; }
16         //public DbSet<Coordinate> Coordinates { get; set; }
17         public DbSet<Item> Items { get; set; }
18         public DbSet<ItemNote> ItemNotes { get; set; }
19         public DbSet<AdventureNote> AdventureNotes { get; set; }
20
21         public ApplicationDbContext()
22             : base("DefaultConnection", throwIfV1Schema: false)
23         {
24         }
25
26         public static ApplicationDbContext Create()
27         {
28             return new ApplicationDbContext();
29         }
30     }
31 }
```

```

1 using System;
2 using System.Globalization;
3 using System.Linq;
4 using System.Security.Claims;
5 using System.Threading.Tasks;
6 using System.Web;
7 using System.Web.Mvc;
8 using Microsoft.AspNet.Identity;
9 using Microsoft.AspNet.Identity.Owin;
10 using Microsoft.Owin.Security;
11 using AdventureLog.Models;
12 using System.Data.Entity;
13 using System.Collections.Generic;
14 using System.Security.Principal;
15 using System.Text.RegularExpressions;
16
17 namespace AdventureLog.Controllers
18 {
19     /// <summary>
20     /// Primary controller for managing adventures
21     /// Index Route: ~/adventure
22     /// </summary>
23     public class AdventureHomeController : Controller
24     {
25         public AdventureHomeController()
26         {
27         }
28
29         #region Index
30         /// <summary>
31         /// navigates to the ~/adventure index page.
32         /// </summary>
33         /// <returns>Index View</returns>
34         [Authorize]
35         [Route("adventure", Name = "adventurehome")]
36         public ActionResult Index()
37         {
38             // Instantiate the model.
39             var model = new AdventureHomeModel();
40             // Create the view.
41             var view = View(model);
42
43             // Users must be logged in.
44             if (Request.IsAuthenticated)
45             {
46                 User.Identity.GetUserId();
47
48                 // Get the users adventures.
49                 using (ApplicationDbContext dbContext = new
50                     ApplicationDbContext())
51                 {
52                     var currentUserId = User.Identity.GetUserId();
53
54                     var adventures = (from a in dbContext.Adventures

```

```
55         where player.IsActive
56         && a.IsActive
57         select a).AsEnumerable();
58
59         model.Adventures = adventures.ToList();
60     }
61
62     // Populate View with Administration messages.
63     model.Messages.Add(new KeyValuePair<string, string>(
64         "First Deployment on Azure",
65         "Adventure Log is now running on Azure! This is a big
66         step forward for Adventure Log."
67         + "Our next push will be for an official product with the
68         features we wanted to create in Adventure Log.));
69
70     }
71
72     return view;
73 }
74
75 #endregion
76
77 #region Adventure Actions
78
79 #region Create
80
81 /// <summary>
82 /// HTTP Get: for the create adventure page. Route: ~/adventure/
83 Create
84 /// </summary>
85 [Authorize]
86 [Route("adventure/Create")]
87 public ActionResult Create()
88 {
89     // Create empty adventure with default values.
90     var model = new Adventure()
91     {
92         CreatedDate = DateTime.Now,
93         LastModifiedDate = DateTime.Now,
94         LastModifiedUser = User.Identity.GetUserName()
95     };
96     var view = View("Create", model);
97
98     return View();
99 }
100
101 // Post: AdventureHome/Create
102 /// <summary>
103 /// HTTP Post for create adventure page.
104 /// </summary>
105 /// <param name="model"></param>
106 /// <returns></returns>
107 [HttpPost]
108 [ValidateAntiForgeryToken]
109 [Route("adventure/Create")]
110 public ActionResult Create(Adventure model)
111 {
```

```
108         // If required fields are filled out.
109         if (ModelState.IsValid)
110         {
111             // Activate the adventure.
112             model.IsActive = true;
113
114             // Add the user as the gamemaster of the adventure.
115             var newPlayer = new Player()
116             {
117                 UserId_PK = User.Identity.GetUserId(),
118                 Adventure = model,
119                 PlayerRole_PK = (int)PlayerRole.PlayerRoleKey.Gamemaster,
120                 IsActive = true
121             };
122
123             // Update the database.
124             using (var dbContext = new ApplicationDbContext())
125             {
126                 dbContext.Adventures.Add(model);
127                 dbContext.Players.Add(newPlayer);
128                 dbContext.SaveChanges();
129             }
130
131             // Go to the adventure home page.
132             return RedirectToAction("Index");
133         }
134
135         // If we got this far, something failed, redisplay form
136         return View(model);
137     }
138
139     #endregion
140
141     #region Details
142
143     /// <summary>
144     /// Http Get for detail view of a specified Adventure
145     /// </summary>
146     /// <param name="id"></param>
147     /// <returns></returns>
148     [Authorize]
149     [Route("adventure/{id:long}")]
150     public ActionResult Details(long id)
151     {
152         Adventure adventure = null;
153         // Display the access invalid view if the adventure could not be
154         // found or the user does not have access to view it.
155         ActionResult view = View("AccessInvalid");
156
157         using (var dbContext = new ApplicationDbContext())
158         {
159             string userId = User.Identity.GetUserId();
160
161             // Get the adventure.
162             adventure = (from a in dbContext.Adventures
163                         let player = a.Players.FirstOrDefault(p =>
```

```

        p.UserId_PK == userId)
        where a.Adventure_PK == id
            && ( a.IsPublic || player.IsActive)
            && a.IsActive
        select a)
        // Include Items and areas as they are displayed
in cards.
        .Include(a => a.AdventureNotes.Select(n =>
n.ApplicationUser))
        .Include(a => a.AdventureNotes.Select(n =>
n.ChildNotes))
        .Include(a => a.Items.Select(w => w.ChildItems))
        .FirstOrDefault();
    }

    // If the adventure was found.
    if (adventure != null)
    {
        // Display the adventure.
        view = View("Details", adventure);
    }

    return view;
}

#endregion

#region Edit

/// <summary>
/// HTTP Get for editing an adventure
/// </summary>
/// <param name="id"></param>
/// <returns></returns>
[Authorize]
[Route("adventure/{id:long}/Edit")]
public ActionResult Edit(long id)
{
    Adventure adventure = null;
    // Display the access invalid view if the adventure could not be
    found or the user does not have access to edit it.
    ActionResult view = View("AccessInvalid");

    using (var dbContext = new ApplicationDbContext())
    {
        string userId = User.Identity.GetUserId();

        // Get the adventure.
        adventure = (from a in dbContext.Adventures
            let player = a.Players.FirstOrDefault(p =>
p.UserId_PK == userId)
            where a.Adventure_PK == id
                && player.IsActive
                // Assure the user is a gamemaster
                && player.PlayerRole.PlayerRole_PK == (long)
PlayerRole.PlayerRoleKey.Gamemaster

```

```
212         && a.IsActive
213         select a)
214         // Persist Player Role and Application User for displaying.
215         .Include(a => a.Players.Select(p =>
216         p.PlayerRole))
217         .Include(a => a.Players.Select(p =>
218         p.ApplicationUser))
219         .FirstOrDefault();
220     }
221     // If the adventure was found.
222     if (adventure != null)
223     {
224         // Display the edit view.
225         view = View("Edit", adventure);
226     }
227     return view;
228 }
229
230 /// <summary>
231 /// HTTP Post for saving an edited adventure.
232 /// </summary>
233 /// <param name="model"></param>
234 /// <returns></returns>
235 [HttpPost]
236 [ValidateAntiForgeryToken]
237 [Authorize]
238 [Route("adventure/{id:long}/Edit")]
239 public ActionResult Edit(Adventure model, long[] activePlayers =
240     null)
241 {
242     ActionResult result = View(model);
243
244     // If the model is valid, update database, and return to detail
245     // view.
246     if (ModelState.IsValid)
247     {
248         using (var dbContext = new ApplicationDbContext())
249         {
250             // Update model values
251             model.LastModifiedDate = DateTime.Now;
252             model.LastModifiedUser = User.Identity.GetUserName();
253
254             // If everything goes well, update database with listed
255             // properties.
256             dbContext.Entry(model).State = EntityState.Modified;
257
258             // Update active player list.
259             var players = (from p in dbContext.Players
260                 where p.Adventure_PK == model.Adventure_PK
261                 select p).AsEnumerable();
262
263             foreach (var player in players)
264             {

```



```
262         // If there has been a change to the players active status...
263         if (player.IsActive != activePlayers.Any(ap =>
264             ap.Equals(player.Player_PK)))
265         {
266             dbContext.Entry(player).Entity.IsActive = !
267             dbContext.Entry(player).Entity.IsActive;
268             dbContext.Entry(player).State =
269             EntityState.Modified;
270         }
271         // Save Changes
272         dbContext.SaveChanges();
273     }
274
275     // Return to the detail view
276     result = RedirectToAction("Details", model.Adventure_PK);
277 }
278
279 return result;
280 }
281
282 #endregion
283
284 #region Invite
285
286 // Get: adventure/{id}/Invite/{Invite Password}
287 [Authorize]
288 [Route("adventure/{id:long}/Invite/{invitePassword?}")]
289 public ActionResult Invite(long id, string invitePassword = "")
290 {
291     ActionResult result = View("InviteFailed");
292
293     using (var dbContext = new ApplicationDbContext())
294     {
295         string userId = User.Identity.GetUserId();
296
297         // User must be logged in to use Invite Links
298         if (userId != null)
299         {
300             // Find the adventure
301             var adventure = (from a in dbContext.Adventures
302                             where a.Adventure_PK == id
303                                 && a.IsActive
304                                 select a).FirstOrDefault();
305
306             // If the adventure was found and the password is valid.
307             if (adventure != null
308                 && (string.IsNullOrEmpty
309                     (adventure.InvitePassword)
310                     || string.Equals
311                     (adventure.InvitePassword, invitePassword)))
312             {
313                 var player = adventure.Players.Where(p => p.UserId_PK ==
```

```
== userId).FirstOrDefault();

312
313     // Add the player if they do not exist.
314     if (player == null)
315     {
316         // Add the user as a new player.
317         var newPlayer = new Player()
318         {
319             UserId_PK = userId,
320             Adventure = adventure,
321             PlayerRole_PK = (int)
322             PlayerRole.PlayerRoleKey.Player
323             };
324
325         // If the adventure is secured, do not whitelist
326         the player.
327         newPlayer.IsActive = !adventure.IsSecured;
328
329         dbContext.Players.Add(newPlayer);
330         dbContext.SaveChanges();
331     }
332
333     if (!adventure.IsSecured || player.IsActive)
334     {
335         // Redirect to the details page of the newly
336         joined adventure.
337         result = RedirectToAction("Details", id);
338     }
339     else
340     {
341         // Redirect to the success page.
342         result = View("InviteSuccess");
343     }
344 }
345
346 return result;
347 }
348
349 #endregion
350
351 #region Comments
352 /// <summary>
353 /// HTTP Post for creating a comment.
354 /// </summary>
355 [Authorize, HttpPost, ValidateInput(false)]
356 public ActionResult CreateComment(long Adventure_PK, string
357 newComment, long? parentComment = null)
358 {
359     // Return to the detail view of the adventure.
360     ActionResult result = RedirectToAction("Details", new { id =
361 Adventure_PK });
362
363     // The comment must contain something.
364     if (!string.IsNullOrEmpty(newComment))
```

```
362     {
363         var userId = User.Identity.GetUserId();
364
365         // Create the comment.
366         var note = new AdventureNote()
367         {
368             Adventure_PK = Adventure_PK,
369             UserId_PK = userId,
370             ParentAdventureNote_PK = parentComment,
371             Text = newComment,
372             IsActive = true,
373             CreatedDate = DateTime.Now,
374             LastModifiedDate = DateTime.Now,
375             LastModifiedUser = User.Identity.GetUserName()
376         };
377
378         using (var dbContext = new ApplicationDbContext())
379         {
380             // Check if the adventure exists that meets the qualifications.
381             var isPlayer = (from a in dbContext.Adventures
382                             let player = a.Players.FirstOrDefault(p
383                                 => p.UserId_PK == userId)
384                             where a.Adventure_PK == Adventure_PK
385                                 && a.IsActive
386                                 && player.IsActive
387                             select a).Any();
388
389             // Add the comment.
390             if (isPlayer)
391             {
392                 dbContext.AdventureNotes.Add(note);
393                 dbContext.SaveChanges();
394             }
395         }
396
397         return result;
398     }
399
400     /// <summary>
401     /// HTTP Post for deleting a comment.
402     /// </summary>
403     [Authorize, HttpPost]
404     public ActionResult DeleteComment(long AdventureNote_PK)
405     {
406         ActionResult result = RedirectToAction("Index");
407
408         using (var dbContext = new ApplicationDbContext())
409         {
410             // Get the comment.
411             var note = (from n in dbContext.AdventureNotes
412                         where n.AdventureNote_PK == AdventureNote_PK
413                             && n.IsActive
414                         select n).FirstOrDefault();
415         }
```

```
416         if (note != null)
417         {
418             // If the user is the gamemaster.
419             var isGamemaster = IsInRole(User.Identity,
note.Adventure_PK, PlayerRole.PlayerRoleKey.Gamemaster);
420
421             if (isGamemaster)
422             {
423                 // Delete all replies including the deleted comment.
424                 DeleteOrphanAdventureNotes(note.AdventureNote_PK,
dbContext);
425                 dbContext.SaveChanges();
426                 result = RedirectToAction("Details", new { id =
note.Adventure_PK });
427             }
428         }
429     }
430
431     return result;
432 }
433
434 /// <summary>
435 /// Http Post for editing the comment.
436 /// </summary>
437 [Authorize, HttpPost, ValidateInput(false)]
438 public ActionResult EditComment(long AdventureNote_PK, string
commentText, long? parentComment = null)
439 {
440     ActionResult result = null;
441     var userId = User.Identity.GetUserId();
442
443     using (var dbContext = new ApplicationDbContext())
444     {
445         // Get the comment to edit.
446         var note = (from n in dbContext.AdventureNotes
447                     where n.AdventureNote_PK == AdventureNote_PK
448                     select n)
449                     .FirstOrDefault();
450
451         if (note != null)
452         {
453             // Check if the adventure exists that meets the
454             // qualifications.
455             var isPlayer = (from a in dbContext.Adventures
456                             let player = a.Players.FirstOrDefault(p
=> p.UserId_PK == userId)
457                             where a.Adventure_PK == note.Adventure_PK
458                             && a.IsActive
459                             && player.IsActive
460                             select a).Any();
461
462             // If the user is the player that posted the comment,
463             // edit it.
464             if (isPlayer)
465             {
466                 note.Text = commentText;
```

```
465
466         dbContext.Entry(note).Property("Text").IsModified = true;
467         dbContext.SaveChanges();
468
469         result = RedirectToAction("Details", new { id = note.Adventure_PK});
470     }
471 }
472
473
474
475     return result;
476 }
477
478 /// <summary>
479 /// Recursively deletes all items associated with an comment.
480 /// </summary>
481 /// <param name="AdventureNote_PK">Comment to delete.</param>
482 /// <param name="dbContext">database access to use.</param>
483 private void DeleteOrphanAdventureNotes(long AdventureNote_PK, ApplicationDbContext dbContext)
484 {
485     var note = (from n in dbContext.AdventureNotes
486                 where n.AdventureNote_PK == AdventureNote_PK
487                 select n)
488                 .Include(n => n.ChildNotes)
489                 .FirstOrDefault();
490
491     foreach (var child in note.ChildNotes)
492     {
493         DeleteOrphanAdventureNotes(child.AdventureNote_PK, dbContext);
494     }
495
496     note.IsActive = false;
497     dbContext.Entry(note).Property("IsActive").IsModified = true;
498 }
499
500 #endregion
501
502 #region Delete Adventure
503
504 /// <summary>
505 /// Http Post to delete an adventure.
506 /// </summary>
507 /// <param name="adventure_PK"></param>
508 [Authorize, HttpPost]
509 public ActionResult DeleteAdventure(long adventure_PK)
510 {
511     ActionResult result = RedirectToAction("Index");
512
513     using (var dbContext = new ApplicationDbContext())
514     {
515         var userId = User.Identity.GetUserId();
516     }
```

```
517         // Find the adventure.
518         var adventure = (from a in dbContext.Adventures
519                         let player = a.Players.FirstOrDefault(p => p.UserId_PK == userId)
520                         where a.Adventure_PK == adventure_PK
521                             && a.IsActive
522                             && player.IsActive
523                             // Assure the user is the gamemaster.
524                             && player.PlayerRole.PlayerRole_PK == (long)PlayerRole.PlayerRoleKey.Gamemaster
525                         select a)
526             .Include(a => a.Items)
527             .Include(a => a.AdventureNotes)
528             .FirstOrDefault();
529
530         if (adventure != null)
531         {
532             // disable the adventure.
533             adventure.IsActive = false;
534
535             // update the database.
536             dbContext.Entry(adventure).Property("IsActive").IsModified = true;
537
538             // Delete all sub-items.
539             foreach (var item in adventure.Items)
540             {
541                 DeleteOrphanItems(item.Item_PK, dbContext);
542             }
543
544             // Delete all notes associated with the adventure.
545             foreach (var note in adventure.AdventureNotes)
546             {
547                 DeleteOrphanAdventureNotes(note.AdventureNote_PK, dbContext);
548             }
549
550             dbContext.SaveChanges();
551         }
552     }
553
554     return result;
555 }
556
557 #endregion
558
559 #region Search
560 /// <summary>
561 /// Http Post for searching for an item in an adventure.
562 /// </summary>
563 /// <param name="adventure_PK"></param>
564 /// <param name="searchText"></param>
565 /// <returns></returns>
566 [Authorize, HttpPost]
567 public ActionResult AdventureSearch(long adventure_PK, string searchText)
```

```
568     {
569         ActionResult result = RedirectToAction("SearchResults", new { id =
            = adventure_PK });
570
571         using (var dbContext = new ApplicationDbContext())
572         {
573             var userId = User.Identity.GetUserId();
574
575             // Check to see if there is an exact match on the name.
576             var results = (from i in dbContext.Items
577                 let player =
                    i.Adventure.Players.FirstOrDefault(p => p.UserId_PK ==
                    userId)
578                 where i.Adventure_PK == adventure_PK
579                     && i.Name == searchText
580                     && i.IsActive
581                     && (i.Adventure.IsPublic || player.IsActive)
582                 select i)
583                 .ToList();
584
585             if (results.Count() == 1)
586             {
587                 // If there is only one exact match, go to the detail
                    page.
588                 result = RedirectToAction("ItemDetails", new { id =
                    results.First().Item_PK });
589             }
590             else
591             {
592                 // If there are multiple matches, start the fuzzy search.
593                 // Get all items to search through.
594                 results = (from i in dbContext.Items
595                     where i.Adventure_PK == adventure_PK
596                     && i.IsActive
597                     select i)
598                     .ToList();
599
600                 // Find a match on any names that contain the non-case-
                    sensitive text of the search.
601                 Regex regex = new Regex("^.*" + searchText.ToLower() +
                    ".*$");
602
603                 // Remove all items that do not match.
604                 results = results.Where(i => regex.IsMatch(i.Name.ToLower
                    ())).ToList();
605
606                 // Save the list for the view in temporary data
607                 TempData["searchResults"] = results;
608                 result = RedirectToAction("SearchResults", new { id =
                    adventure_PK });
609             }
610         }
611
612         return result;
613     }
614
```

```

615     /// <summary>
616     /// Http Get for the search results page.
617     /// </summary>
618     /// <param name="id"></param>
619     /// <returns></returns>
620     [HttpGet]
621     [Route("adventure/{id:long}/Search/Results")]
622     public ActionResult SearchResults(long id)
623     {
624         // Get the results from temporary data.
625         var searchResults = TempData["searchResults"] as List<Item>;
626         // Create the view model.
627         var viewModel = new SearchResultsViewModel(null, searchResults);
628
629         using (var dbContext = new ApplicationDbContext())
630         {
631             var adventure = (from a in dbContext.Adventures
632                             where a.Adventure_PK == id
633                                 && a.IsActive
634                                 select a)
635                             // Get the list of items for the search
636                             menu.
637                             .Include(a => a.Items)
638                             .FirstOrDefault();
639
640             if (adventure != null)
641             {
642                 viewModel.Adventure = adventure;
643             }
644
645             return View(viewModel);
646         }
647
648     #endregion
649
650     #endregion
651
652     #region Item Actions
653
654     #region Create
655
656     /// <summary>
657     /// HTTP Get. for creating an item. Route: ~/adventure/{adventureId}/
658     Item/Create
659     /// </summary>
660     /// <param name="adventureId"></param>
661     [Authorize]
662     [Route("adventure/{adventureId:long}/CreateItem/{parentId:long?}"),
663     Name = "ItemCreate"]
664     public ActionResult ItemCreate(long adventureId, long? parentId)
665     {
666         Adventure adventure = null;
667         // Display the access invalid view if the adventure could not be
668         found or the user does not have access to view it.
669         ActionResult result = View("AccessInvalid");

```



```
667
668     using (var dbContext = new ApplicationDbContext())
669     {
670         string userId = User.Identity.GetUserId();
671
672         adventure = (from a in dbContext.Adventures
673                     let player = a.Players.FirstOrDefault(p =>      ↗
674                         p.UserId_PK == userId)
675                     where a.Adventure_PK == adventureId
676                         && a.IsActive
677                         && player.IsActive
678                         // Ensure the user is a gamemaster.
679                         && player.PlayerRole.PlayerRole_PK == (long) ↗
680                         PlayerRole.PlayerRoleKey.Gamemaster
681                     select a).FirstOrDefault();
682
683     }
684
685     if (adventure != null)
686     {
687         // Create the default item with preset values.
688         var model = new Item()
689         {
690             Adventure = adventure,
691             Adventure_PK = adventure.Adventure_PK,
692             ParentItem_PK = parentId,
693             CreatedDate = DateTime.Now,
694             LastModifiedDate = DateTime.Now,
695             LastModifiedUser = User.Identity.GetUserName()
696         };
697
698         result = View("ItemCreate", model);
699     }
700
701     return result;
702 }
703
704 /// <summary>
705 /// HTTP Post for Creating a new Item.
706 /// </summary>
707 /// <param name="model"></param>
708 /// <returns></returns>
709 [HttpPost]
710 [Authorize]
711 [Route("adventure/{adventureId:long}/CreateItem")]
712 public ActionResult ItemCreate(Item model)
713 {
714     ActionResult result = View(model);
715
716     if (ModelState.IsValid)
717     {
718         model.IsActive = true;
719
720         using (var dbContext = new ApplicationDbContext())
721         {
722             dbContext.Items.Add(model);
723             dbContext.SaveChanges();
724         }
725     }
726 }
```

```

721     }
722
723     if (model.ParentItem_PK != null)
724     {
725         // Return to the parent.
726         result = RedirectToAction("ItemDetails", new { id =
727                                     model.ParentItem_PK });
728     }
729     else
730     {
731         // Return to the adventure view.
732         result = RedirectToAction("Details", new { id =
733                                     model.Adventure_PK });
734     }
735
736     // If we got this far, redisplay the view for corrections.
737     return result;
738 }
739 #endregion
740
741 #region Details
742 /// <summary>
743 /// HTTP Get for item details. Route: ~/Item/id
744 /// </summary>
745 /// <param name="id"></param>
746 /// <returns></returns>
747 [Authorize]
748 [Route("Item/{id:long}", Name = "ItemDetails")]
749 public ActionResult ItemDetails(long id)
750 {
751     Item item = null;
752     ActionResult result = View("AccessInvalid");
753
754     using (var dbContext = new ApplicationDbContext())
755     {
756         string userId = User.Identity.GetUserId();
757
758         item = (from i in dbContext.Items
759                let player = i.Adventure.Players.FirstOrDefault(p =>
760                    p.UserId_PK == userId)
761                where i.Item_PK == id
762                    && i.IsActive
763                    // Assure that the adventure is one the player
764                    can view AND is not deleted.
765                    && i.Adventure.IsActive
766                    && (i.Adventure.IsPublic || player.IsActive)
767                select i)
768                .Include(w => w.ItemNotes.Select(n =>
769                    n.ApplicationUser))
770                .Include(w => w.ItemNotes.Select(n => n.ChildNotes))
771                // Load 2 levels of children
772                .Include(w => w.ChildItems.Select(c =>
773                    c.ChildItems))
774                .FirstOrDefault();

```

```

771     }
772
773     if (Item != null)
774     {
775         result = View("ItemDetails", Item);
776     }
777
778     return result;
779 }
780 #endregion
781
782 #region Edit
783 /// <summary>
784 /// HTTP Get for editing an item. Route: ~/item/id/Edit
785 /// </summary>
786 /// <param name="id"></param>
787 /// <returns></returns>
788 [Authorize]
789 [Route("Item/{id:long}/Edit", Name="ItemEdit")]
790 public ActionResult ItemEdit(long id)
791 {
792     Item Item = null;
793     // If there is an issue, go to the detail view of the item.
794     ActionResult view = RedirectToAction("ItemDetails", id);
795
796     using (var dbContext = new ApplicationDbContext())
797     {
798         string userId = User.Identity.GetUserId();
799
800         Item = (from i in dbContext.Items
801                 let player = i.Adventure.Players.FirstOrDefault(p => p
802                                     .UserId_PK == userId)
803                 where i.Item_PK == id
804                     // Assure that the adventure is one the player
805                     // can edit AND is not deleted.
806                     && i.IsActive
807                     && i.Adventure.IsActive
808                     && player.IsActive
809                     // Ensure the user is a gamemaster.
810                     && player.PlayerRole.PlayerRole_PK == (long)
811                         PlayerRole.PlayerRoleKey.Gamemaster
812                     select i).FirstOrDefault();
813
814     if (Item != null)
815     {
816         view = View("ItemEdit", Item);
817     }
818
819     return view;
820 }
821
822 /// <summary>
823 /// Http Post for editing an item.
824 /// </summary>

```

```

824     /// <param name="model"></param>
825     /// <returns></returns>
826     [HttpPost, ValidateAntiForgeryToken, Authorize]
827     [Route("Item/{id:long}/Edit")]
828     public ActionResult ItemEdit(Item model)
829     {
830         ActionResult result = View(model);
831
832         // If the model is valid, update database, and return to detail  ➤
833         view.
834         if (ModelState.IsValid)
835         {
836             using (var dbContext = new ApplicationDbContext())
837             {
838                 // Update model values
839                 model.LastModifiedDate = DateTime.Now;
840                 model.LastModifiedUser = User.Identity.GetUserName();
841
842                 // If everthing goes well, update database with listed  ➤
843                 properties.
844                 dbContext.Entry(model).State = EntityState.Modified;
845                 dbContext.SaveChanges();
846             }
847             result = RedirectToAction("ItemDetails", model.Item_PK);
848         }
849         return result;
850     }
851 #endregion
852
853 #region Delete Item
854     /// <summary>
855     /// Http Post for deleting an item.
856     /// </summary>
857     /// <param name="item_PK"></param>
858     /// <param name="adventure_pk"></param>
859     /// <returns></returns>
860     [Authorize, HttpPost]
861     public ActionResult DeleteItem(long item_PK, long adventure_pk)
862     {
863         // If it is successfully deleted, go to the adventure detail  ➤
864         page.
865         ActionResult result = RedirectToAction("Details", new { id =  ➤
866             adventure_pk });
867
868         using (var dbContext = new ApplicationDbContext())
869         {
870             var userId = User.Identity.GetUserId();
871
872             // Find the item.
873             var item = (from i in dbContext.Items
874                 let player = i.Adventure.Players.FirstOrDefault(p ➤
875                     => p.UserId_PK == userId)
876                 where i.Item_PK == item_PK
877                 && player.IsActive

```

```
875         // ensure the user is the gamemaster.
876         && player.PlayerRole.PlayerRole_PK == (long) ↗
            PlayerRole.PlayerRoleKey.Gamemaster
            select i).FirstOrDefault();
877
878
879         if (item != null)
880         {
881             // Delete all child items.
882             DeleteOrphanItems(item.Item_PK, dbContext);
883             dbContext.SaveChanges();
884         }
885     }
886
887     return result;
888 }
889
890 /// <summary>
891 /// Recursively delete all child items of an item.
892 /// </summary>
893 /// <param name="Item_PK">item to delete.</param>
894 /// <param name="dbContext">database context to use.</param>
895 private void DeleteOrphanItems(long Item_PK, ApplicationDbContext ↗
    dbContext)
896 {
897     var item = (from i in dbContext.Items
898                 where i.Item_PK == Item_PK
899                 select i)
900                 .Include(i => i.ChildItems)
901                 .Include(i => i.ItemNotes)
902                 .FirstOrDefault();
903
904     // Delete childs of the children.
905     foreach (var child in item.ChildItems)
906     {
907         DeleteOrphanItems(child.Item_PK, dbContext);
908     }
909
910     // Delete all notes on the item.
911     foreach (var note in item.ItemNotes)
912     {
913         DeleteOrphanComments(note.ItemNote_PK, dbContext);
914     }
915
916     // Delete the item.
917     item.IsActive = false;
918     dbContext.Entry(item).Property("IsActive").IsModified = true;
919 }
920
921 #endregion
922
923 #region Comments
924 /// <summary>
925 /// Http Post for creating a comment.
926 /// </summary>
927 /// <param name="Item_PK"></param>
928 /// <param name="newComment"></param>
```

```

929     /// <param name="parentComment"></param>
930     /// <returns></returns>
931     [Authorize, HttpPost, ValidateInput(false)]
932     public ActionResult CreateItemComment(long Item_PK, string      ↗
933         newComment, long? parentComment = null)
934     {
935         ActionResult result = RedirectToAction("ItemDetails", new { id = ↗
936             Item_PK });
937
938         if (!string.IsNullOrEmpty(newComment))
939         {
940             var userId = User.Identity.GetUserId();
941
942             var note = new ItemNote()
943             {
944                 Item_PK = Item_PK,
945                 UserId_PK = userId,
946                 ParentItemNote_PK = parentComment,
947                 Text = newComment,
948                 IsActive = true,
949                 CreatedDate = DateTime.Now,
950                 LastModifiedDate = DateTime.Now,
951                 LastModifiedUser = User.Identity.GetUserName()
952             };
953
954             using (var dbContext = new ApplicationDbContext())
955             {
956                 // Check if an item exists that meets the qualifications.
957                 var isPlayer = (from i in dbContext.Items
958                     let player =      ↗
959                     i.Adventure.Players.FirstOrDefault(p => p.UserId_PK == ↗
960                         userId)
961                     where i.Item_PK == Item_PK
962                     && i.IsActive
963                     && i.Adventure.IsActive
964                     && player.IsActive
965                     select i).Any();
966
967                 if (isPlayer)
968                 {
969                     dbContext.ItemNotes.Add(note);
970                     dbContext.SaveChanges();
971                 }
972             }
973
974             return result;
975         }
976     }
977
978     /// <summary>
979     /// Http Post for deleting an item.
980     /// </summary>
981     /// <param name="ItemNote_PK"></param>
982     /// <returns></returns>
983     [Authorize, HttpPost]
984     public ActionResult DeleteItemComment(long ItemNote_PK)

```

```

981     {
982         ActionResult result = RedirectToAction("Index");
983
984         using (var dbContext = new ApplicationDbContext())
985         {
986             var note = (from n in dbContext.ItemNotes
987                         where n.ItemNote_PK == ItemNote_PK
988                             && n.IsActive
989                         select n)
990                 .Include(n => n.Item)
991                 .FirstOrDefault();
992
993             if (note != null)
994             {
995                 // Ensure the user is the gamemaster.
996                 var isGamemaster = IsInRole(User.Identity,
997                                     note.Item.Adventure_PK,
998                                     PlayerRole.PlayerRoleKey.Gamemaster);
999
1000                 if (isGamemaster)
1001                 {
1002                     // Delete the comment and all child comments.
1003                     DeleteOrphanComments(note.ItemNote_PK, dbContext);
1004                     dbContext.SaveChanges();
1005                     result = RedirectToAction("ItemDetails", new { id =
1006                                     note.Item_PK });
1007                 }
1008             }
1009         }
1010
1011         return result;
1012     }
1013
1014     /// <summary>
1015     /// Http Post to edit the comment.
1016     /// </summary>
1017     /// <param name="ItemNote_PK"></param>
1018     /// <param name="commentText"></param>
1019     /// <param name="parentComment"></param>
1020     /// <returns></returns>
1021     [Authorize, HttpPost, ValidateInput(false)]
1022     public ActionResult EditItemComment(long ItemNote_PK, string
1023         commentText, long? parentComment = null)
1024     {
1025         ActionResult result = null;
1026         var userId = User.Identity.GetUserId();
1027
1028         using (var dbContext = new ApplicationDbContext())
1029         {
1030             var note = (from n in dbContext.ItemNotes
1031                         where n.ItemNote_PK == ItemNote_PK
1032                         select n)
1033                 .Include(n => n.Item)
1034                 .FirstOrDefault();
1035
1036             if (note != null)

```

```
1033     {
1034         // Check if the adventure exists that meets the
1035         // qualifications.
1036         var isPlayer = (from a in dbContext.Adventures
1037             let player = a.Players.FirstOrDefault(p
1038                 => p.UserId_PK == userId)
1039             where a.Adventure_PK ==
1040                 note.Item.Adventure_PK
1041                 && a.IsActive
1042                 && player.IsActive
1043                 select a).Any();
1044
1045         if (isPlayer)
1046         {
1047             note.Text = commentText;
1048
1049             dbContext.Entry(note).Property("Text").IsModified =
1050             true;
1051             dbContext.SaveChanges();
1052
1053             result = RedirectToAction("ItemDetails", new { id =
1054                 note.Item_PK });
1055         }
1056     }
1057
1058     return result;
1059 }
1060
1061 /// <summary>
1062 /// Delete all child comments of a comment.
1063 /// </summary>
1064 /// <param name="ItemNote_PK"></param>
1065 /// <param name="dbContext"></param>
1066 private void DeleteOrphanComments(long ItemNote_PK,
1067     ApplicationDbContext dbContext)
1068 {
1069     var note = (from n in dbContext.ItemNotes
1070         where n.ItemNote_PK == ItemNote_PK
1071         select n)
1072         .Include(n => n.ChildNotes)
1073         .FirstOrDefault();
1074
1075     foreach (var child in note.ChildNotes)
1076     {
1077         DeleteOrphanComments(child.ItemNote_PK, dbContext);
1078     }
1079
1080     note.IsActive = false;
1081     dbContext.Entry(note).Property("IsActive").IsModified = true;
1082 }
1083
1084 #endregion
1085
1086 #endregion
```



```
1083
1084     #region Utilities
1085     /// <summary>
1086     /// Method to ascertain if a user is in a role for a specified adventure.
1087     /// </summary>
1088     /// <param name="user">User Id to check for.</param>
1089     /// <param name="adventure_Pk">Specified Adventure.</param>
1090     /// <param name="playerRoleKey">Player Role to check for.</param>
1091     /// <returns></returns>
1092     public static bool IsInRole(IIdentity user, long adventure_Pk,
1093                                PlayerRole.PlayerRoleKey playerRoleKey)
1094     {
1095         bool isInRole = true;
1096         string userId = user.GetUserId();
1097
1098         using (var dbContext = new ApplicationDbContext())
1099         {
1100             isInRole = (from p in dbContext.Players
1101                         where p.UserId_PK == userId
1102                             && p.Adventure_PK == adventure_Pk
1103                             && p.PlayerRole_PK == (int)playerRoleKey
1104                             && p.IsActive
1105                         select p).Any();
1106         }
1107
1108         return isInRole;
1109     }
1110     /// <summary>
1111     /// Check if a user is in any role for a specified adventure.
1112     /// </summary>
1113     /// <param name="user">User Id to check for.</param>
1114     /// <param name="adventure_Pk">Specified adventure.</param>
1115     /// <param name="playerRoleKeys">enumerable item list of player roles
1116     /// to check for.</param>
1117     /// <returns></returns>
1118     public static bool IsInAnyRole(IIdentity user, long adventure_Pk,
1119                                    IEnumerable<PlayerRole.PlayerRoleKey> playerRoleKeys)
1120     {
1121         bool isInAnyRole = false;
1122
1123         foreach (var key in playerRoleKeys)
1124         {
1125             if (IsInRole(user, adventure_Pk, key))
1126             {
1127                 isInAnyRole = true;
1128                 break;
1129             }
1130         }
1131
1132         return isInAnyRole;
1133     }
1134     /// <summary>
1135     /// Checks if a user is the original creator of a comment.
```

```
1135     /// </summary>
1136     /// <param name="user">User Id to check for.</param>
1137     /// <param name="AdventureNote_PK">key of adventure comment to check for.</param>
1138     /// <returns></returns>
1139     public static bool IsAdventureNoteOwner(IIdentity user, long AdventureNote_PK)
1140     {
1141         bool isAdventureNoteOwner = true;
1142         string userId = user.GetUserId();
1143
1144         using (var dbContext = new ApplicationDbContext())
1145         {
1146             isAdventureNoteOwner = (from n in dbContext.AdventureNotes
1147                                     where n.AdventureNote_PK ==
1148                                             AdventureNote_PK
1149                                             && n.UserId_PK == userId
1150                                     select n).Any();
1151         }
1152         return isAdventureNoteOwner;
1153     }
1154
1155     /// <summary>
1156     /// Checks if a user is the original creator of a comment.
1157     /// </summary>
1158     /// <param name="user">User Id to check for.</param>
1159     /// <param name="ItemNote_PK">Key of adventure comment to check for.</param>
1160     /// <returns></returns>
1161     public static bool IsItemNoteOwner(IIdentity user, long ItemNote_PK)
1162     {
1163         bool isAdventureNoteOwner = true;
1164         string userId = user.GetUserId();
1165
1166         using (var dbContext = new ApplicationDbContext())
1167         {
1168             isAdventureNoteOwner = (from n in dbContext.ItemNotes
1169                                     where n.ItemNote_PK == ItemNote_PK
1170                                     && n.UserId_PK == userId
1171                                     select n).Any();
1172         }
1173         return isAdventureNoteOwner;
1174     }
1175 }
1176 #endregion
1177 }
1178 }
```

```

1  @model AdventureLog.Models.AdventureHomeModel
2  @using Microsoft.AspNet.Identity
3  @{
4      ViewBag.Title = "Recent Adventures";
5      ViewBag.MessageTitle = "From the Adventure Log Team:";
6  }
7  <div class="row">
8      <div class="col-8">
9          <div class="jumbotron mb-1">
10             <h1>@ViewBag.Title</h1>
11          </div>
12          <div class="btn-group mb-1" role="group">
13              @Html.ActionLink("Create Game", "Create", null, new { @class = "btn btn-primary" })
14          </div>
15          <div class="row mb-0">
16              @foreach (var adventure in Model.Adventures)
17              {
18                  <div class="col-6 pb-2">
19                      <div class="card border-dark">
20                          <div class="card-header">
21                              <h2 class="float-left">@adventure.Name</h2>
22                              @Html.ActionLink("Join Game", "Details", new { id = adventure.Adventure_PK }, new { @class = "btn btn-primary float-right" })
23                          </div>
24                          <div class="card-body text-dark">
25                              <div class="card-title">
26                                  System: @adventure.SystemName
27                              </div>
28                              <hr class="col-xs-12">
29                              <div class="card-text text-dark">
30                                  Summary: <br>
31                                  @adventure.ShortDescription
32                              </div>
33                          </div>
34                      </div>
35                  </div>
36              }
37          </div>
38      </div>
39      <div class="col-4">
40          <div class="card border border-primary">
41              <h2 class="card-header">@ViewBag.MessageTitle</h2>
42              @foreach (var message in Model.Messages)
43              {
44                  <div class="card-body p-2">
45                      <div class="card border border-secondary">
46                          <div class="card-header">
47                              <h4>
48                                  @message.Key
49                              </h4>
50                          </div>
51                          <div class="card-body">
52                              @message.Value
53                          </div>

```

```
54         </div>
55     </div>
56 }
57 </div>
58 </div>
59 </div>
```

```
1 @model AdventureLog.Models.Adventure
2
3 @{
4     ViewBag.Title = "Create Adventure";
5 }
6
7 <h2>@ViewBag.Title</h2>
8
9 @using (Html.BeginForm())
10 {
11     @Html.AntiForgeryToken()
12
13     <div class="form-horizontal">
14         <hr />
15         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
16         <div class="form-group">
17             @Html.LabelFor(model => model.Name, htmlAttributes: new { @class =
18                 "control-label col-md-2" })
19             <div class="col-md-10">
20                 @Html.EditorFor(model => model.Name, new { htmlAttributes = new
21                     { @class = "form-control" } })
22                 @Html.ValidationMessageFor(model => model.Name, "", new
23                     { @class = "text-danger" })
24             </div>
25         </div>
26
27         <div class="form-group">
28             @Html.LabelFor(model => model.SystemName, htmlAttributes: new
29                 { @class = "control-label col-md-2" })
30             <div class="col-md-10">
31                 @Html.EditorFor(model => model.SystemName, new { htmlAttributes
32                     = new { @class = "form-control" } })
33                 @Html.ValidationMessageFor(model => model.SystemName, "", new
34                     { @class = "text-danger" })
35             </div>
36         </div>
37
38         <div class="form-group">
39             @Html.LabelFor(model => model.ShortDescription, htmlAttributes: new
40                 { @class = "control-label col-md-2" })
41             <div class="col-md-10">
42                 @Html.EditorFor(model => model.ShortDescription, new
43                     { htmlAttributes = new { @class = "form-control" } })
44                 @Html.ValidationMessageFor(model => model.ShortDescription, "",
45                     new { @class = "text-danger" })
46             </div>
47         </div>
48
49         <div class="form-group">
50             @Html.LabelFor(model => model.Description, htmlAttributes: new
51                 { @class = "control-label col-md-2" })
52             <div class="col-md-10">
53                 @Html.TextAreaFor(model => model.Description, new
54                     { htmlAttributes = new { @class = "form-control" } })
55                 <script>CKEDITOR.replace('Description', { customConfig: '/
56                     Scripts/ckeditor_config.js' })</script>
57             </div>
58         </div>
59     </div>
60 }
```

```
45         @Html.ValidationMessageFor(model => model.Description, "", new
           { @class = "text-danger" })
46     </div>
47 </div>
48
49 <div class="form-group">
50     <div class="col-md-10">
51         @Html.HiddenFor(model => model.CreatedDate, new
           { htmlAttributes = new { @class = "form-control" } })
52     </div>
53 </div>
54
55 <div class="form-group">
56     <div class="col-md-10">
57         @Html.HiddenFor(model => model.LastModifiedDate, new
           { htmlAttributes = new { @class = "form-control" } })
58     </div>
59 </div>
60
61 <div class="form-group">
62     <div class="col-md-10">
63         @Html.HiddenFor(model => model.LastModifiedUser, new
           { htmlAttributes = new { @class = "form-control" } })
64     </div>
65 </div>
66
67 <div class="form-group">
68     <div class="col-md-offset-2 col-md-10">
69         <input type="submit" value="Create" class="btn btn-primary" />
70     </div>
71 </div>
72 </div>
73 }
74
75 <div>
76     @Html.RouteLink("Back to List", "adventurehome", null, new { @class="btn
       btn-secondary"})
77 </div>
78
```

```

1  @model AdventureLog.Models.Adventure
2  @using AdventureLog.Controllers;
3  @using AdventureLog.Models;
4  @{
5      ViewBag.Title = "Adventure Details";
6      ViewBag.SearchToolTip = "Use '#' to search for tags";
7      ViewBag.SearchPlaceholder = "Search for Items...";
8
9      var ViewPermissionList = new List<PlayerRole.PlayerRoleKey>()
10     { PlayerRole.PlayerRoleKey.Gamemaster,
11       PlayerRole.PlayerRoleKey.Player };
12 }
13
12 <div class="row">
13     <div class="col-lg-12">
14         <h2 class="font-weight-bold">@Model.Name</h2>
15     </div>
16     <div class="col-lg-6">
17         <h4 class="text-secondary">Last edited by: @Model.LastModifiedUser on
18         @Model.LastModifiedDate.ToShortDateString()</h4>
19     </div>
20     <div class="col-lg-6">
21         @using (Html.BeginForm("AdventureSearch", "AdventureHome"))
22         {
23             <div class="input-group" id="adv-search">
24                 <input type="text"
25                     id="search"
26                     name="searchText"
27                     class="form-control col-md-8"
28                     placeholder=@ViewBag.SearchPlaceholder
29                     data-toggle="tooltip"
30                     data-html="true"
31                     data-placement="top"
32                     title="@ViewBag.SearchToolTip"/>
33                 @Html.HiddenFor(model => model.Adventure_PK)
34                 <div class="input-group-append">
35                     <button type="submit"
36                         class="btn btn-primary">
37                         <span class="fa fa-search" aria-hidden="true"></span>
38                     </button>
39                 </div>
40             </div>
41             <script>
42                 $(function () {
43                     $('#search').autocomplete
44                     (
45                         {
46                             minChars: 1,
47                             source: function (term, suggest) {
48                                 term = term.toLowerCase();
49                                 var choices = [];
50                                 @foreach (var item in Model.Items.Where(i =>
51                                     i.IsActive))
52                                     {
53                                         @: choices.push("@item.Name");
54                                     }
55                             }
56                         }
57                     );
58                 });
59             </script>
60         }
61     </div>
62 </div>

```

```

53         var suggestions = [];
54         for (i = 0; i < choices.length; i++)
55             if (~choices[i].toLowerCase().indexOf
56                 (term))
57                 suggestions.push(choices[i]);
58
59             suggest(suggestions);
60         }
61     }
62 });
63 </script>
64 }
65 </div>
66 </div>
67 <hr />
68 <nav>
69     <div class="nav nav-tabs" id="nav-tab" role="tablist">
70         <a class="nav-item nav-link active"
71             id="nav-Summary-tab"
72             data-toggle="tab"
73             href="#nav-summary"
74             role="tab"
75             aria-controls="nav-home"
76             aria-selected="true">Details</a>
77         <a class="nav-item nav-link"
78             id="nav-Items-tab"
79             data-toggle="tab"
80             href="#nav-Items"
81             role="tab"
82             aria-controls="nav-home"
83             aria-selected="true">Items</a>
84
85         @if (AdventureHomeController.IsInAnyRole(User.Identity,
86             Model.Adventure_PK, ViewPermissionList))
87         {
88             <a class="nav-item nav-link"
89                 id="nav-comments-tab"
90                 data-toggle="tab"
91                 href="#nav-comments"
92                 role="tab"
93                 aria-controls="nav-home"
94                 aria-selected="true">Comments</a>
95         }
96     </div>
97 </nav>
98 <div class="tab-content" id="nav-tabContent">
99     <div class="tab-pane fade show active p-3" id="nav-summary"
100         role="tabpanel" aria-labelledby="nav-home-tab">
101         <!--Summary-->
102         <dl class="dl-horizontal">
103             <dt>
104                 @Html.LabelFor(model => model.SystemName):
105             </dt>
106             <dd>

```



```

106         @Html.DisplayFor(model => model.SystemName)
107     </dd>
108
109     <dt>
110         @Html.DisplayNameFor(model => model.Description):
111     </dt>
112
113     <dd>
114         @Html.Raw(Model.Description)
115     </dd>
116
117     @if (AdventureHomeController.IsInRole(User.Identity,
118         Model.Adventure_PK, PlayerRole.PlayerRoleKey.Gamemaster))
119     {
120         <dt>
121             @Html.DisplayNameFor(model => model.GamemasterNotes):
122         </dt>
123
124         <dd>
125             @Html.Raw(Model.GamemasterNotes)
126         </dd>
127     }
128 </dl>
129 @if (AdventureHomeController.IsInRole(User.Identity,
130     Model.Adventure_PK, PlayerRole.PlayerRoleKey.Gamemaster))
131 {
132     @Html.ActionLink("Edit", "Edit", new { id = Model.Adventure_PK },
133         new { @class = "btn btn-primary" })
134 }
135 </div>
136 <div class="tab-pane fade p-3" id="nav-Items" role="tabpanel" aria-
137     labelledby="nav-profile-tab">
138     <!--Item List-->
139     <div class="row p-3">
140         <div class="nav flex-column nav-pills col-2" id="v-pills-tab"
141             role="tablist" aria-orientation="vertical">
142             @if (AdventureHomeController.IsInRole(User.Identity,
143                 Model.Adventure_PK, PlayerRole.PlayerRoleKey.Gamemaster))
144             {
145                 @Html.ActionLink("Create New Item", "ItemCreate", new
146                     { adventureId = Model.Adventure_PK }, new { @class = "btn
147                         btn-primary mb-1" })
148             }
149             @
150             bool first = true;
151             foreach (var Item in Model.Items.Where(w => w.ParentItem
152                 == null && w.IsActive).AsEnumerable())
153             {
154                 <a class="btn btn-outline-secondary mb-1
155                     @(first ? "active" : "")"
156                     id="v-pills-tab-@Item.Item_PK"
157                     data-toggle="pill"
158                     href="#v-pills-item-@Item.Item_PK"
159                     role="tab"
160                     aria-controls="v-pills-item"

```

```

153         aria-selected="true">@Item.Name</a>
154         first = false;
155     }
156 }
157 </div>
158
159 <div class="tab-content col-10" id="v-pills-tabContent">
160     @{
161         first = true;
162         foreach (var Item in Model.Items.Where(w => w.ParentItem == null && w.IsActive).AsEnumerable())
163         {
164             <div class="tab-pane fade show @(first ? "active" :
165                 "" )"
166                 id="v-pills-item-@Item.Item_PK"
167                 role="tabpanel"
168                 aria-labelledby="v-pills-tab-@Item.Item_PK">
169                 <div class="card border-dark">
170                     <div class="card-header">
171                         <h2 class="float-left text-
172                             dark">@Item.Name</h2>
173                         @Html.ActionLink("See More",
174                             "ItemDetails", new { id = Item.Item_PK }, new { @class =
175                             "btn btn-primary float-right" })
176                     </div>
177                     <div class="card-body text-dark">
178                         <div class="card-text text-dark mb-2">
179                             Summary: <br>
180                             @Item.ShortDescription
181                         </div>
182                         @if (Item.ChildItems.Count > 0)
183                         {
184                             <button class="btn btn-outline-
185                                 secondary mb-2"
186                                     data-toggle="collapse"
187                                     href="#Item-@Item.Item_PK"
188                                     aria-expanded="true"
189                                     aria-controls="Item-
190                                     @Item.Item_PK"
191                                     role="button">
192                                 Sub-Items <i class="fa" aria-
193                                     hidden="true"></i>
194                             </button>
195                             }
196                             @if (AdventureHomeController.IsInRole
197                                 (User.Identity, Model.Adventure_PK,
198                                 PlayerRole.PlayerRoleKey.Gamemaster))
199                             {
200                                 @Html.ActionLink("Create New Sub-
201                                     Item", "ItemCreate",
202                                     new { adventureId =
203                                     Model.Adventure_PK, parentId = Item.Item_PK },
204                                     new { @class = "btn btn-outline-
205                                     primary mb-2" })
206                             }
207                             @ShowItemTree(Item)

```

```

196         </div>
197
198         @helper ShowItemTree(Item parent)
199         {
200             <div class="collapse show" id="Item-
201             @parent.Item_PK">
202                 @foreach (var child in
203                 parent.ChildItems)
204                 {
205                     <div class="card card-inner
206                     border-secondary">
207                         <div class="card-header">
208                             <h4 class="float-left
209                             text-dark">@child.Name</h4>
210                             @Html.ActionLink("See
211                             More", "ItemDetails", new { id = child.Item_PK }, new
212                             { @class = "btn btn-secondary float-right" })
213                         </div>
214                         <div class="card-body text-
215                         secondary">
216                             <div class="card-text
217                             text-secondary mb-2">
218                                 Summary: <br />
219                                 @child.ShortDescription
220                             </div>
221                             @if
222                             (child.ChildItems.Count > 0)
223                             {
224                                 <button class="btn
225                                 btn-outline-secondary mb-2"
226                                 data-
227                                 toggle="collapse"
228                                 href="#Item-
229                                 @child.Item_PK"
230                                 aria-
231                                 expanded="true"
232                                 aria-
233                                 controls="Item-@child.Item_PK"
234                                 role="button">
235                                     Sub-Items <i
236                                     class="fa" aria-hidden="true"></i>
237                                 </button>
238                             }
239                             @if
240                             (AdventureHomeController.IsInRole(User.Identity,
241                             Model.Adventure_PK, PlayerRole.PlayerRoleKey.Gamemaster))
242                             {
243                                 @Html.ActionLink
244                                 ("Create New Sub-Item", "ItemCreate",
245                                 new { adventureId = parent.Adventure_PK, parentId = child.Item_PK },
246                                 new { @class = "btn btn-outline-primary mb-2" })
247                             }
248                             @ShowItemTree(child)

```

```

231         </div>
232     </div>
233     }
234 </div>
235 }
236 </div>
237 </div>
238
239     first = false;
240 }
241 }
242 </div>
243 </div>
244 </div>
245
246 @if (AdventureHomeController.IsInAnyRole(User.Identity,
247     Model.Adventure_PK, ViewPermissionList))
248 {
249     <div class="tab-pane fade p-3" id="nav-comments" role="tabpanel" aria-
250         labelledby="nav-contact-tab">
251         <!--Comments-->
252         <button type="button" class="btn btn-primary mb-2" data-
253             toggle="collapse" data-target="#newCommentCollapse">New Comment</
254             button>
255         <div class="collapse" id="newCommentCollapse">
256             <div class="card mb-2 border border-primary">
257                 <div class="card-body">
258                     <div class="row">
259                         <div class="col-md-12">
260                             <p>
261                                 <strong>New Comment</strong>
262                             </p>
263                             <div class="clearfix"></div>
264                             <hr />
265                             <div class="p-2">
266                                 @using (Html.BeginForm("CreateComment",
267                                     "AdventureHome"))
268                                 {
269                                     <textarea rows="5"
270                                         id="NewComment"
271                                         name="newComment"></textarea>
272                                     <script>CKEDITOR.replace('NewComment',
273                                         { customConfig: '/Scripts/ckeditor_config.js' })</script>
274                                     @Html.HiddenFor(model =>
275                                         model.Adventure_PK);
276                                     <input type="submit" value="Post Comment"
277                                         class="btn btn-primary ml-2 mt-2 float-right" />
278                                 }
279                             </div>
280                         </div>
281                     </div>
282                 </div>
283             </div>
284         </div>
285     </div>
286 </div>
287 </div>
288

```

```

279     @{
280         var parentComments = Model.AdventureNotes.Where(n =>
281             n.ParentAdventureNote == null && n.IsActive).AsEnumerable();
282
283         foreach (var parent in parentComments)
284         {
285             <div class="card mb-2 border border-primary">
286                 <div class="card-header">
287                     <button class="btn btn-outline-primary mr-1"
288                         data-toggle="collapse"
289                         data-target="#card-body-
290                             @parent.AdventureNote_PK">
291                         <i class="fa fa-chevron-down" aria-
292                             hidden="true"></i>
293                     </button>
294                     <strong>@parent.ApplicationUser.UserName</strong>
295                     <small class="float-right">@parent.LastModifiedDate</
296                         small>
297                 </div>
298                 <div class="card-body collapse show" id="card-body-
299                     @parent.AdventureNote_PK">
300                     <div class="row">
301                         <div class="col-md-12">
302                             <div class="accordion p-0" id="commentCard-
303                                 @parent.AdventureNote_PK">
304                                 <div id="text-@parent.AdventureNote_PK"
305                                     class="collapse show" data-parent="#commentCard-
306                                     @parent.AdventureNote_PK">
307                                     @Html.Raw(parent.Text)
308                                 </div>
309                                 @if
310                                     (AdventureHomeController.IsAdventureNoteOwner
311                                     (User.Identity, parent.AdventureNote_PK))
312                                 {
313                                     <div id="text-
314                                     @parent.AdventureNote_PK" class="collapse" data-
315                                     parent="#commentCard-@parent.AdventureNote_PK">
316
317                                         @using (Html.BeginForm
318                                             ("EditComment", "AdventureHome"))
319                                         {
320                                             @Html.TextArea("commentText",
321                                                 parent.Text, new { @id = "comment-text-" +
322                                                 parent.AdventureNote_PK })
323                                             @Html.Hidden
324                                                 ("AdventureNote_PK", parent.AdventureNote_PK)
325                                             <input type="submit"
326                                                 value="Submit" class="btn btn-primary m-2" />
327                                             <script>CKEDITOR.replace
328                                                 ('comment-text-@parent.AdventureNote_PK', { customConfig:
329                                                 '/Scripts/ckeditor_config.js' })</script>
330                                         }
331                                     </div>
332                                 }
333                             </div>
334                         </div>
335                     </div>
336                 </div>
337             <hr />

```

```

...tureLog\AdventureLog\Views\AdventureHome\Details.cshtml 8
316         @if 2
317         (AdventureHomeController.IsAdventureNoteOwner 2
318         (User.Identity, parent.AdventureNote_PK)) 2
319         { 2
320             <button class="btn btn-outline-warning 2
321             mb-2" 2
322                 data-toggle="collapse" 2
323                 data-target="#text- 2
324             @parent.AdventureNote_PK"> 2
325             Edit 2
326             </button> 2
327         } 2
328         @ShowChildTree(parent) 2
329     </div> 2
330 </div> 2
331 2
332     @helper ShowChildTree(AdventureNote parentNote) 2
333     { 2
334         <!--Reply Section--> 2
335         <button type="button" class="btn btn-outline- 2
336         primary mb-2" data-toggle="collapse" data- 2
337         target="#newReplyCollapse- 2
338         @parentNote.AdventureNote_PK">Reply</button> 2
339         <div class="collapse" id="newReplyCollapse- 2
340         @parentNote.AdventureNote_PK"> 2
341             <div class="card card-inner mb-1 mt-1 border 2
342             border-secondary"> 2
343                 <div class="card-header"> 2
344                     <strong>New Reply</strong> 2
345                 </div> 2
346                 <div class="card-body"> 2
347                     <div class="row"> 2
348                         <div class="col-md-12"> 2
349                             <div class="p-0"> 2
350                                 @using (Html.BeginForm 2
351                                 ("CreateComment", "AdventureHome")) 2
352                                 { 2
353                                     <textarea rows="5" 2
354                                     id="NewReply-@parentNote.AdventureNote_PK" 2
355                                     name="newComment"></textarea> 2
356                                     <script>CKEDITOR.replace('NewReply-@parentNote.AdventureNote_PK', 2
357                                     { customConfig: '/Scripts/ckeditor_config.js' })</script> 2
358                                     @Html.HiddenFor(model 2
359                                     => model.Adventure_PK) 2
360                                     @Html.Hidden 2
361                                     ("parentComment", parentNote.AdventureNote_PK) 2
362                                     <input type="submit" 2
363                                     value="Post Reply" class="btn btn-primary ml-2 mt-2 float- 2
364                                     right" /> 2
365                                 } 2
366                             </div> 2
367                         </div> 2
368                     </div> 2
369                 </div> 2
370             </div> 2
371         } 2
372     } 2
373 } 2

```

```

354         </div>
355     </div>
356 </div>
357 </div>
358 </div>
359
360 <!-- Delete Comment -->
361     if (AdventureHomeController.IsInRole
362         (User.Identity, Model.Adventure_PK,
363         PlayerRole.PlayerRoleKey.Gamemaster))
364     {
365         <div class="float-right">
366             <button class="btn btn-danger mb-2"
367                 data-toggle="collapse"
368                 href="#deleteCollapse-
369                 @parentNote.AdventureNote_PK"
370                 aria-expanded="true"
371                 aria-controls="deleteCollapse-
372                 @parentNote.AdventureNote_PK"
373                 role="button">
374                 Delete
375             </button>
376             <div class="collapse mr-4 mb-2 float-left"
377                 id="deleteCollapse-@parentNote.AdventureNote_PK">
378                 @using (Html.BeginForm
379                     ("DeleteComment", "AdventureHome"))
380                 {
381                     <input hidden
382                         name="AdventureNote_PK"
383                         value="@parentNote.AdventureNote_PK" />
384                     <input type="submit"
385                         value="Confirm Delete" class="btn btn-outline-danger" />
386                 }
387             </div>
388         </div>
389     }
390
391 <!-- Child Comments -->
392     <div class="collapse show" id="reply-
393         @parentNote.AdventureNote_PK">
394         @foreach (var child in
395             parentNote.ChildNotes.Where(c => c.IsActive).AsEnumerable
396             ())
397         {
398             <div class="card card-inner mb-1 mt-1
399                 border border-secondary">
400                 <div class="card-header">
401                     <button class="btn btn-outline-
402                         secondary mr-1"
403                         data-toggle="collapse"
404                         data-target="#card-body-
405                         @child.AdventureNote_PK">
406                         <i class="fa fa-chevron-down"
407                             aria-hidden="true"></i>
408                     </button>

```

```

<strong>@child.ApplicationUser.UserName</strong>
394         <small class="float-           2
right">@child.LastModifiedDate</small>
395     </div>
396     <div class="card-body collapse show" 2
id="card-body-@child.AdventureNote_PK">
397         <div class="row">
398             <div class="col-md-12">
399                 <div class="accordion p-0" 2
id="commentCard-@child.AdventureNote_PK">
400                     <div id="text-           2
@child.AdventureNote_PK" class="collapse show" data-           2
parent="#commentCard-@child.AdventureNote_PK">
401                         @Html.Raw           2
(child.Text)
402                     </div>
403                     @if           2
(AdventureHomeController.IsAdventureNoteOwner
404                     (User.Identity, child.AdventureNote_PK))
405                         {
406                             <div id="text-           2
@child.AdventureNote_PK" class="collapse" data-           2
parent="#commentCard-@child.AdventureNote_PK">
407                                 @using           2
(Html.BeginForm("EditComment", "AdventureHome"))
408                                 {
409                                     @Html.TextArea("commentText", child.Text, new { @id = "comment-text-" +
@child.AdventureNote_PK })
410                                     @Html.Hidden("AdventureNote_PK", child.AdventureNote_PK)
411                                     <input           2
type="submit" value="Submit" class="btn btn-primary m-2" / 2
>
412                                     <script>CKEDITOR.replace('comment-text-@child.AdventureNote_PK',
{ customConfig: '/Scripts/ckeditor_config.js' })</script>
413                                     }
414                                 </div>
415                             }
416                         </div>
417                         <hr />
418                         @if           2
(AdventureHomeController.IsAdventureNoteOwner
419                         (User.Identity, child.AdventureNote_PK))
420                             {
421                                 <button class="btn           2
btn-outline-warning mb-2"
422                                     data-           2
toggle="collapse"
423                                     data-           2
target="#text-@child.AdventureNote_PK">
424                                     Edit
425                                 </button>
}

```



```
426         @ShowChildTree(child)
427     </div>
428 </div>
429 </div>
430 </div>
431     }
432 </div>
433     }
434 </div>
435 </div>
436     }
437 }
438 </div>
439 }
440 </div>
441 <hr />
442 <p>
443     @Html.RouteLink("Back to Adventure List", "adventurehome", null, new
444         { @class = "btn btn-secondary" })
```

```
1 @model AdventureLog.Models.Adventure
2 @{
3     ViewBag.Title = "Edit Adventure";
4 }
5
6 <h2>@ViewBag.Title</h2>
7 <hr />
8
9 @using (Html.BeginForm())
10 {
11     @Html.AntiForgeryToken()
12
13     <div class="form-horizontal">
14         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
15         @Html.HiddenFor(model => model.Adventure_PK)
16
17         <div class="form-group">
18             @Html.LabelFor(model => model.Name, htmlAttributes: new { @class =
19                 "control-label col-md-2" })
20             <div class="col-md-10">
21                 @Html.EditorFor(model => model.Name, new { htmlAttributes = new
22                     { @class = "form-control" } })
23                 @Html.ValidationMessageFor(model => model.Name, "", new { @class =
24                     "text-danger" })
25             </div>
26         </div>
27
28         <div class="form-group">
29             @Html.LabelFor(model => model.SystemName, htmlAttributes: new { @class
30                 = "control-label col-md-2" })
31             <div class="col-md-10">
32                 @Html.EditorFor(model => model.SystemName, new { htmlAttributes =
33                     new { @class = "form-control" } })
34                 @Html.ValidationMessageFor(model => model.SystemName, "", new
35                     { @class = "text-danger" })
36             </div>
37         </div>
38
39         <div class="form-group">
40             @Html.LabelFor(model => model.ShortDescription, htmlAttributes: new
41                 { @class = "control-label col-md-2" })
42             <div class="col-md-10">
43                 @Html.TextAreaFor(model => model.ShortDescription, 10, 80, new
44                     { htmlAttributes = new { @class = "form-control" } })
45                 @Html.ValidationMessageFor(model => model.ShortDescription, "",
46                     new { @class = "text-danger" })
47             </div>
48         </div>
49
50         <div class="form-group">
51             @Html.LabelFor(model => model.Description, htmlAttributes: new
52                 { @class = "control-label col-md-2" })
53             <div class="col-md-10">
54                 @Html.TextAreaFor(model => model.Description, new { htmlAttributes
55                     = new { @class = "form-control" } })
56                 <script>CKEDITOR.replace('Description', { customConfig: '/Scripts/
```

```

        ckeditor_config.js' }</script>
46         @Html.ValidationMessageFor(model => model.Description, "", new
            { @class = "text-danger" })
47     </div>
48 </div>
49
50 <div class="form-group">
51     @Html.LabelFor(model => model.GamemasterNotes, htmlAttributes: new
            { @class = "control-label col-md-6" })
52     <div class="col-md-10">
53         @Html.TextAreaFor(model => model.GamemasterNotes, new
            { htmlAttributes = new { @class = "form-control" } })
54         <script>CKEDITOR.replace('GamemasterNotes', { customConfig: '/
            Scripts/ckeditor_config.js' }</script>
55         @Html.ValidationMessageFor(model => model.GamemasterNotes, "", new
            { @class = "text-danger" })
56     </div>
57 </div>
58
59
60
61 <hr />
62 <h4>Manage Players</h4>
63
64 <script>
65     $(document).ready(function () {
66         var selection = "private";
67         if (@Model.IsPublic == "True")
68             selection = "public";
69         if (@Model.IsSecured == "True")
70             selection = "secure";
71         $("input:radio[name='securitySettings'][value='" + selection +
            "']").prop('checked', true);
72         $("#" + selection).addClass("active");
73     });
74 </script>
75 <div class="row pl-3 form-group">
76     <p class="col-md-12">General Security Settings:</p>
77
78     <div id="tab" class="btn-group-vertical col-md-4" data-
        toggle="buttons">
79         <a href="#public" class="btn btn-secondary" data-toggle="tab">
80             <input class="form-check-input"
81                 type="radio"
82                 name="securitySettings"
83                 id="publicRadio"
84                 value="public"
85                 checked />
86             Public
87         </a>
88         <a href="#private" class="btn btn-secondary" data-toggle="tab">
89             <input class="form-check-input"
90                 type="radio"
91                 name="securitySettings"
92                 id="privateRadio"
93                 value="private" />

```

```

94         Private
95     </a>
96     <a href="#secure" class="btn btn-secondary" data-toggle="tab">
97         <input class="form-check-input"
98             type="radio"
99             name="securitySettings"
100             id="secureRadio"
101             value="secure" />
102     Secure
103 </a>
104 </div>
105
106 <div class="tab-content col-md-8">
107     <div class="tab-pane card" id="public">
108         <h6 class="card card-header">Public:</h6>
109         <div class="card card-body">
110             Can View: Anyone<br />
111             Can Comment: Invited Players Only<br />
112             Can Edit/Delete: Gamemaster Only<br />
113         </div>
114     </div>
115     <div class="tab-pane card" id="private">
116         <h6 class="card card-header">Private:</h6>
117         <div class="card card-body">
118             Can View: Anyone with Invite Link<br />
119             Can Comment: Invited Players Only<br />
120             Can Edit/Delete: Gamemaster Only<br />
121         </div>
122     </div>
123     <div class="tab-pane card" id="secure">
124         <h6 class="card card-header">Secured:</h6>
125         <div class="card card-body">
126             Can View: Invited and Approved Players only<br />
127             Can Comment: Invited and Approved Players only<br />
128             Can Edit/Delete: Gamemaster Only<br />
129         </div>
130     </div>
131 </div>
132 </div>
133
134
135 <script type="text/javascript">
136     $("#publicRadio, #privateRadio, #secureRadio").change(function () {
137         if (this.value == 'public') {
138             $("#IsPublic").prop('checked', true);
139             $("#IsSecured").prop('checked', false);
140         }
141         else if (this.value == 'secure') {
142             $("#IsPublic").prop('checked', false);
143             $("#IsSecured").prop('checked', true);
144         }
145         else if (this.value == 'private') {
146             $("#IsPublic").prop('checked', false);
147             $("#IsSecured").prop('checked', false);
148         }
149     });

```

```

150     </script>
151
152     <div hidden>
153         @Html.CheckBoxFor(model => model.IsPublic, new { htmlAttributes = new
154             { @class = "form-control" } })
155         @Html.CheckBoxFor(model => model.IsSecured, new { htmlAttributes = new
156             { @class = "form-control" } })
157     </div>
158
159     <div class="form-group">
160         @Html.ValidationMessageFor(model => model.InvitePassword, "", new
161             { @class = "text-danger" })
162         @Html.LabelFor(model => model.InvitePassword, htmlAttributes: new
163             { @class = "control-label col-md-4" })
164         <div class="col-md-8 input-group">
165             @Html.EditorFor(model => model.InvitePassword, new
166                 { htmlAttributes = new { @class = "form-control" } })
167             <div class="input-group-append">
168                 <script type="text/javascript">
169                     function copyInviteLink(data) {
170                         var textarea = document.createElement("textarea");
171
172                         textarea.style.position = 'fixed';
173                         textarea.style.top = 0;
174                         textarea.style.left = 0;
175
176                         textarea.style.width = '2em';
177                         textarea.style.height = '2em';
178
179                         textarea.style.padding = 0;
180
181                         textarea.style.border = 'none';
182                         textarea.style.outline = 'none';
183                         textarea.style.boxShadow = 'none';
184                         textarea.style.background = 'transparent';
185
186                         textarea.value = data;
187                         document.body.appendChild(textarea);
188                         textarea.select();
189
190                         try {
191                             var isCopied = document.execCommand('copy');
192                             var msg = isCopied ? 'Copied to Clipboard ' +
193                                 data : 'Failed to Copy';
194                             console.log(msg)
195                         }
196                         catch (err) {
197                             console.log(err);
198                         }
199
200                         document.body.removeChild(textarea);
201                     }
202                 </script>
203                 <button type="button"
204                     class="btn btn-primary"
205                     onclick="copyInviteLink('@String.Format("{0}Invite/

```

```

        {1}", Request.Url.AbsoluteUri.Substring(0,
        Request.Url.AbsoluteUri.Length - 4),
        Model.InvitePassword)')">
200     Invite Link <i class="fa fa-copy"></i>
201 </button>
202 </div>
203 </div>
204 </div>
205
206
207 <div class="form-group">
208     @Html.LabelFor(model => model.Players, htmlAttributes: new { @class =
        "control-label col-md-2" })
209     <table class="table">
210         <thead>
211             <tr>
212                 <th scope="col">User Id</th>
213                 <th scope="col">Role</th>
214                 <th scope="col">Can Comment/View?</th>
215             </tr>
216         </thead>
217         <tbody>
218             @{
219                 List<AdventureLog.Models.Player> activePlayers =
        Model.Players.ToList();
220
221                 foreach (var player in activePlayers)
222                 {
223                     @:<tr>
224                         @:<td>@player.ApplicationUser.UserName</td>
225                         @:<td>@player.PlayerRole.RoleName</td>
226                         @:<td>
227                             <input type="checkbox"
228                                 name="activePlayers"
229                                 value="@player.Player_PK"
230                                 @(Html.Raw(player.IsActive ? "checked="
        "\"checked\" : \"\") ) />
231                         @:</td>
232                     @:</tr>
233                 }
234             }
235         </tbody>
236     </table>
237 </div>
238
239 <hr />
240
241
242 @Html.HiddenFor(model => model.IsActive)
243 @Html.HiddenFor(model => model.CreatedDate, new { htmlAttributes = new
        { @class = "form-control" } })
244 @Html.HiddenFor(model => model.LastModifiedDate, new { htmlAttributes =
        new { @class = "form-control" } })
245 @Html.HiddenFor(model => model.LastModifiedUser, new { htmlAttributes =
        new { @class = "form-control" } })
246

```

```
247     <div class="form-group">
248         <div class="col-md-offset-2 col-md-10">
249             <input type="submit" value="Save" class="btn btn-primary" />
250         </div>
251     </div>
252 </div>
253 }
254 <hr />
255 <div class="d-flex">
256     <button class="btn btn-danger"
257         data-toggle="collapse"
258         href="#deleteCollapse"
259         aria-expanded="true"
260         aria-controls="deleteCollapse"
261         role="button">
262         Delete Adventure
263     </button>
264     <div class="collapse ml-4" id="deleteCollapse">
265         @using (Html.BeginForm("DeleteAdventure", "AdventureHome"))
266         {
267
268             @Html.HiddenFor(model => model.Adventure_PK)
269             <input type="submit" value="Confirm Delete" class="btn btn-
270                 outline-danger" />
271         }
272     </div>
273 </div>
274 <hr />
275 <div>
276     @Html.ActionLink("Back to Details", "Details", Model.Adventure_PK, new
277         { @class = "btn btn-secondary" })
278 </div>
```

```
1  @model AdventureLog.Models.Item
2
3  @{
4      ViewBag.Title = "Create Item";
5  }
6
7  <h2>@ViewBag.Title</h2>
8
9  @using (Html.BeginForm())
10 {
11     @Html.AntiForgeryToken()
12
13     <div class="form-horizontal">
14         <hr />
15         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
16
17         @Html.HiddenFor(model => model.Adventure_PK, new { htmlAttributes = new
18             { @class = "form-control" } })
19
20         <div class="form-group">
21             @Html.LabelFor(model => model.Name, htmlAttributes: new { @class =
22                 "control-label col-md-2" })
23             <div class="col-md-10">
24                 @Html.EditorFor(model => model.Name, new { htmlAttributes = new
25                     { @class = "form-control" } })
26                 @Html.ValidationMessageFor(model => model.Name, "", new
27                     { @class = "text-danger" })
28             </div>
29         </div>
30
31         <div class="form-group">
32             @Html.LabelFor(model => model.ShortDescription, htmlAttributes: new
33                 { @class = "control-label col-md-2" })
34             <div class="col-md-10">
35                 @Html.EditorFor(model => model.ShortDescription, new
36                     { htmlAttributes = new { @class = "form-control" } })
37                 @Html.ValidationMessageFor(model => model.ShortDescription, "",
38                     new { @class = "text-danger" })
39             </div>
40         </div>
41
42         <div class="form-group">
43             @Html.LabelFor(model => model.Description, htmlAttributes: new
44                 { @class = "control-label col-md-2" })
45             <div class="col-md-10">
46                 @Html.TextAreaFor(model => model.Description, new
47                     { htmlAttributes = new { @class = "form-control" } })
48                 <script>CKEDITOR.replace('Description', { customConfig: '/
49                     Scripts/ckeditor_config.js' })</script>
50                 @Html.ValidationMessageFor(model => model.Description, "", new
51                     { @class = "text-danger" })
52             </div>
53         </div>
54
55         @Html.HiddenFor(model => model.MapFileName, new { htmlAttributes = new
56             { @class = "form-control" } })
57     </div>
58 }
```



```
45     @Html.HiddenFor(model => model.ParentItem_PK)
46     @Html.HiddenFor(model => model.IsActive)
47     @Html.HiddenFor(model => model.CreatedDate, new { htmlAttributes = new
48         { @class = "form-control" } })
49     @Html.HiddenFor(model => model.LastModifiedDate, new { htmlAttributes =
50         new { @class = "form-control" } })
51     @Html.HiddenFor(model => model.LastModifiedUser, new { htmlAttributes =
52         new { @class = "form-control" } })
53
54     <div class="form-group">
55         <div class="col-md-offset-2 col-md-10">
56             <input type="submit" value="Create" class="btn btn-primary" />
57         </div>
58     </div>
59
60     <div>
61         @Html.ActionLink("Back to Adventure", "Details", new { id =
62             Model.Adventure_PK }, new { @class = "btn btn-secondary" })
63     </div>
```

```

1  @model AdventureLog.Models.Item
2  @using AdventureLog.Controllers
3  @using AdventureLog.Models
4  @{
5      ViewBag.Title = "Item Details";
6
7      var ViewPermissionList = new List<PlayerRole.PlayerRoleKey>()
          { PlayerRole.PlayerRoleKey.Gamemaster,
            PlayerRole.PlayerRoleKey.Player };
8  }
9  <div class="row">
10     <div class="col-lg-12">
11         <h2 class="font-weight-bold">@Model.Name</h2>
12     </div>
13     <div class="col-lg-6">
14         <h4 class="text-secondary">Last edited by: @Model.LastModifiedUser on
            @Model.LastModifiedDate.ToShortDateString()</h4>
15     </div>
16 </div>
17 <hr />
18 <nav>
19     <div class="nav nav-tabs" id="nav-tab" role="tablist">
20         <a class="nav-item nav-link active"
21             id="nav-Summary-tab"
22             data-toggle="tab"
23             href="#nav-summary"
24             role="tab"
25             aria-controls="nav-home"
26             aria-selected="true">Details</a>
27         <a class="nav-item nav-link"
28             id="nav-Items-tab"
29             data-toggle="tab"
30             href="#nav-areas"
31             role="tab"
32             aria-controls="nav-home"
33             aria-selected="true">Sub-Items</a>
34         @if (AdventureHomeController.IsAnyRole(User.Identity,
            Model.Adventure_PK, ViewPermissionList))
35         {
36             <a class="nav-item nav-link"
37                 id="nav-comments-tab"
38                 data-toggle="tab"
39                 href="#nav-comments"
40                 role="tab"
41                 aria-controls="nav-home"
42                 aria-selected="true">Comments</a>
43         }
44     </div>
45 </nav>
46 <div class="tab-content" id="nav-tabContent">
47     <div class="tab-pane fade show active p-3" id="nav-summary"
48         role="tabpanel" aria-labelledby="nav-home-tab">
49         <dl class="dl-horizontal">
50             <dt>
51                 @Html.DisplayNameFor(model => model.Description)

```

```

52
53     <dd>
54         @Html.Raw(Model.Description)
55     </dd>
56
57     @if (AdventureHomeController.IsInRole(User.Identity,
58         Model.Adventure_PK, PlayerRole.PlayerRoleKey.Gamemaster))
59     {
60         <dt>
61             @Html.DisplayNameFor(model => model.GamemasterNotes):
62         </dt>
63
64         <dd>
65             @Html.Raw(Model.GamemasterNotes)
66         </dd>
67
68         @Html.ActionLink("Edit", "ItemEdit", new { id = Model.Item_PK },
69             new { @class = "btn btn-primary" })
70     }
71 </div>
72 <div class="tab-pane fade p-3" id="nav-areas" role="tabpanel" aria-
73     labelledby="nav-profile-tab">
74     <div class="row p-3">
75         <div class="nav flex-column nav-pills col-2" id="v-pills-tab"
76             role="tablist" aria-orientation="vertical">
77             @if (AdventureHomeController.IsInRole(User.Identity,
78                 Model.Adventure_PK, PlayerRole.PlayerRoleKey.Gamemaster))
79             {
80                 @Html.ActionLink("Create Sub-Item", "ItemCreate", new
81                     { adventureId = Model.Adventure_PK, parentId =
82                         Model.Item_PK }, new { @class = "btn btn-primary mb-1" })
83             }
84             @
85             {
86                 bool first = true;
87                 foreach (var child in Model.ChildItems.Where(w =>
88                     w.ParentItem_PK == Model.Item_PK).AsEnumerable())
89                 {
90                     <a class="btn btn-outline-secondary mb-1" @(first ?
91                         "active" : "")
92                         id="v-pills-tab-@child.Item_PK"
93                         data-toggle="pill"
94                         href="#v-pills-item-@child.Item_PK"
95                         role="tab"
96                         aria-controls="v-pills-item"
97                         aria-selected="true">@child.Name</a>
98
99                     first = false;
100                 }
101             }
102         </div>
103         <div class="tab-content col-10" id="v-pills-tabContent">
104             @
105             {
106                 first = true;
107                 foreach (var Item in Model.ChildItems.Where(w =>

```

```

w.ParentItem_PK == Model.Item_PK).AsEnumerable())
{
99     <div class="tab-pane fade show @(first ? "active" :
100     "" )"
101         id="v-pills-item-@Item.Item_PK"
102         role="tabpanel"
103         aria-labelledby="v-pills-tab-@Item.Item_PK">
104         <div class="card border-dark">
105             <div class="card-header">
106                 <h2 class="float-left text-
107                 dark">@Item.Name</h2>
108                 @Html.ActionLink("See More",
109                 "ItemDetails", new { id = Item.Item_PK }, new { @class =
110                 "btn btn-primary float-right" })
111             </div>
112             <div class="card-body text-dark">
113                 <div class="card-text text-dark mb-2">
114                     Summary: <br>
115                     @Item.ShortDescription
116                 </div>
117                 <div class="list-group-flush">
118                     @foreach (var child in Item.ChildItems)
119                     {
120                         <a href="@Url.Action("ItemDetails",
121                         new { id =
122                         child.Item_PK})"
123                         class="list-group-item list-group-
124                         item-action list-group-item-secondary">
125                             <h6>@child.Name: </h6>
126                             <small>
127                                 @child.ShortDescription
128                             </small>
129                         </a>
130                     }
131                 </div>
132             </div>
133         </div>
134         first = false;
135     }
136 </div>
137
138 @if (AdventureHomeController.IsInAnyRole(User.Identity,
139     Model.Adventure_PK, ViewPermissionList))
140 {
141     <div class="tab-pane fade p-3" id="nav-comments" role="tabpanel" aria-
142     labelledby="nav-contact-tab">
143         <!--Comments-->
144         <button type="button" class="btn btn-primary mb-2" data-
145         toggle="collapse" data-target="#newCommentCollapse">New Comment</
146         button>
147         <div class="collapse" id="newCommentCollapse">

```

```

144         <div class="card mb-2 border border-primary">
145             <div class="card-header">
146                 <strong>New Comment</strong>
147             </div>
148             <div class="card-body">
149                 <div class="row">
150                     <div class="col-md-12">
151                         <div class="p-0">
152                             @using (Html.BeginForm("CreateItemComment",
153                                 "AdventureHome"))
154                             {
155                                 <textarea rows="5"
156                                     id="NewComment"
157                                     name="newComment"></textarea>
158                                 <script>CKEDITOR.replace('NewComment',
159                                     { customConfig: '/Scripts/ckeditor_config.js' })</script>
160                                 @Html.HiddenFor(model => model.Item_PK);
161                                 <input type="submit" value="Post Comment"
162                                     class="btn btn-primary ml-2 mt-2 float-right" />
163                             }
164                         </div>
165                     </div>
166                 </div>
167             </div>
168         @if
169         var parentComments = Model.ItemNotes.Where(n => n.ParentItemNote
170             == null && n.IsActive).AsEnumerable();
171
172         foreach (var parent in parentComments)
173         {
174             <div class="card mb-2 border border-primary">
175                 <div class="card-header">
176                     <button class="btn btn-outline-primary mr-1"
177                         data-toggle="collapse"
178                         data-target="#card-body-@parent.ItemNote_PK">
179                         <i class="fa fa-chevron-down" aria-
180                             hidden="true"></i>
181                     </button>
182                     <strong>@parent.ApplicationUser.UserName</strong>
183                     <small class="float-right">@parent.LastModifiedDate</
184                         small>
185                 </div>
186                 <div class="card-body collapse show" id="card-body-
187                     @parent.ItemNote_PK">
188                     <div class="row">
189                         <div class="col-md-12">
190                             <div class="accordion p-0" id="commentCard-
191                                 @parent.ItemNote_PK">
192                                 <div id="text-@parent.ItemNote_PK"
193                                     class="collapse show" data-parent="#commentCard-
194                                     @parent.ItemNote_PK">
195                                     @Html.Raw(parent.Text)
196                                 </div>

```

```

...Log\AdventureLog\Views\AdventureHome\ItemDetails.cshtml 5
190 @if 7
    (AdventureHomeController.IsItemNoteOwner(User.Identity, 7
    parent.ItemNote_PK))
191 {
192     <div id="text-@parent.ItemNote_PK" 7
    class="collapse" data-parent="#commentCard- 7
    @parent.ItemNote_PK">
193
194     @using (Html.BeginForm 7
    ("EditItemComment", "AdventureHome"))
195     {
196         @Html.TextArea("commentText", 7
    parent.Text, new { @id = "comment-text-" + 7
    parent.ItemNote_PK })
197         @Html.Hidden("ItemNote_PK", 7
    parent.ItemNote_PK)
198         <input type="submit" 7
    value="Submit" class="btn btn-primary m-2" />
199         <script>CKEDITOR.replace 7
    ('comment-text-@parent.ItemNote_PK', { customConfig: '/ 7
    Scripts/ckeditor_config.js' })</script>
200     }
201     </div>
202 }
203 </div>
204 <hr />
205 @if (AdventureHomeController.IsItemNoteOwner 7
    (User.Identity, parent.ItemNote_PK))
206 {
207     <button class="btn btn-outline-warning 7
    mb-2"
208         data-toggle="collapse"
209         data-target="#text- 7
    @parent.ItemNote_PK">
210         Edit
211     </button>
212 }
213 @ShowChildTree(parent)
214 </div>
215 </div>
216
217
218 @helper ShowChildTree(ItemNote parentNote)
219 {
220     <!--Reply Section-->
221     <button type="button" class="btn btn-outline- 7
    primary mb-2" data-toggle="collapse" data- 7
    target="#newReplyCollapse-@parentNote.ItemNote_PK">Reply</ 7
    button>
222     <div class="collapse" id="newReplyCollapse- 7
    @parentNote.ItemNote_PK">
223         <div class="card card-inner mb-1 mt-1 border 7
    border-secondary">
224             <div class="card-header">
225                 <strong>New Reply</strong>
226             </div>

```

```

227         <div class="card-body">
228             <div class="row">
229                 <div class="col-md-12">
230                     <div class="p-0">
231                         @using (Html.BeginForm
232                             ("CreateItemComment", "AdventureHome"))
233                             {
234                                 <textarea rows="5"
235                                     id="NewReply-@parentNote.ItemNote_PK"
236                                     name="newComment"></textarea>
237                                 <script>CKEDITOR.replace('NewReply-@parentNote.ItemNote_PK', { customConfig:
238                                     '/Scripts/ckeditor_config.js' })</script>
239                                 @Html.HiddenFor(model
240                                     => model.Item_PK)
241                                 @Html.Hidden
242                                 ("parentComment", parentNote.ItemNote_PK)
243                                 <input type="submit"
244                                     value="Post Reply" class="btn btn-primary ml-2 mt-2 float-
245                                     right" />
246                             }
247                         </div>
248                     </div>
249                 </div>
250             </div>
251             <!-- Delete Comment -->
252             if (AdventureHomeController.IsInRole
253                 (User.Identity, Model.Adventure_PK,
254                 PlayerRole.PlayerRoleKey.Gamemaster))
255             {
256                 <div class="float-right">
257                     <button class="btn btn-danger mb-2"
258                         data-toggle="collapse"
259                         href="#deleteCollapse-
260                         @parentNote.ItemNote_PK"
261                         aria-expanded="true"
262                         aria-controls="deleteCollapse-
263                         @parentNote.ItemNote_PK"
264                         role="button">
265                         Delete
266                     </button>
267                     <div class="collapse mr-4 mb-2 float-left"
268                         id="deleteCollapse-@parentNote.ItemNote_PK">
269                         @using (Html.BeginForm
270                             ("DeleteItemComment", "AdventureHome"))
271                         {
272                             <input hidden name="ItemNote_PK"
273                                 value="@parentNote.ItemNote_PK" />
274                             <input type="submit"
275                                 value="Confirm Delete" class="btn btn-outline-danger" />

```

```

266         }
267     </div>
268 </div>
269 }
270
271 <!-- Child Comments -->
272 <div class="collapse show" id="reply-
@parentNote.ItemNote_PK">
273     @foreach (var child in
parentNote.ChildNotes.Where(child => child.IsActive))
274     {
275         <div class="card card-inner mb-1 mt-1
border border-secondary">
276             <div class="card-header">
277                 <button class="btn btn-outline-
secondary mr-1"
278                     data-toggle="collapse"
279                     data-target="#card-body-
@child.ItemNote_PK">
280                     <i class="fa fa-chevron-down"
aria-hidden="true"></i>
281                 </button>
282
<strong>@child.ApplicationUser.UserName</strong>
283                 <small class="float-
right">@child.LastModifiedDate</small>
284             </div>
285             <div class="card-body collapse show"
id="card-body-@child.ItemNote_PK">
286                 <div class="row">
287                     <div class="col-md-12">
288                         <div class="accordion p-0"
id="commentCard-@child.ItemNote_PK">
289                             <div id="text-
@child.ItemNote_PK" class="collapse show" data-
parent="#commentCard-@child.ItemNote_PK">
290                                 @Html.Raw
(child.Text)
291                             </div>
292                             @if
(AdventureHomeController.IsItemNoteOwner(User.Identity,
child.ItemNote_PK))
293                             {
294                                 <div id="text-
@child.ItemNote_PK" class="collapse" data-
parent="#commentCard-@child.ItemNote_PK">
295
296                                     @using
(Html.BeginForm("EditItemComment", "AdventureHome"))
297                                     {
298
@Html.TextArea("commentText", child.Text, new { @id = "comment-text-" +
child.ItemNote_PK })
299
@Html.Hidden("ItemNote_PK", child.ItemNote_PK)
300                                     <input

```



```

        type="submit" value="Submit" class="btn btn-primary m-2" />
    >
301
    <script>CKEDITOR.replace('comment-text-@child.ItemNote_PK', { customConfig: '/
Scripts/ckeditor_config.js' })</script>
302
303
304
305
306
307
    @if
    (AdventureHomeController.IsItemNoteOwner(User.Identity,
    child.ItemNote_PK))
308
309
    {
        <button class="btn
        btn-outline-warning mb-2"
310
        data-
        toggle="collapse"
311
        data-
        target="#text-@child.ItemNote_PK">
            Edit
        </button>
    }
    @ShowChildTree(child)
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
    @Html.ActionLink("Back to Adventure", "Details", new { id =
        Model.Adventure_PK }, new { @class = "btn btn-secondary" })
333
334
    </p>

```

```
1 @model AdventureLog.Models.Item
2
3 @{
4     ViewBag.Title = "Edit Item";
5 }
6
7 <h2>@ViewBag.Title</h2>
8
9 @using (Html.BeginForm())
10 {
11     @Html.AntiForgeryToken()
12
13     <div class="form-horizontal">
14         <hr />
15         @Html.ValidationSummary(true, "", new { @class = "text-danger" })
16         @Html.HiddenFor(model => model.Item_PK)
17         @Html.HiddenFor(model => model.Adventure_PK)
18         @Html.HiddenFor(model => model.ParentItem_PK)
19
20         <div class="form-group">
21             @Html.LabelFor(model => model.Name, htmlAttributes: new { @class =
22                 "control-label col-md-6" })
23             <div class="col-md-10">
24                 @Html.EditorFor(model => model.Name, new { htmlAttributes = new
25                     { @class = "form-control" } })
26                 @Html.ValidationMessageFor(model => model.Name, "", new { @class =
27                     "text-danger" })
28             </div>
29         </div>
30
31         <div class="form-group">
32             @Html.LabelFor(model => model.ShortDescription, htmlAttributes: new
33                 { @class = "control-label col-md-6" })
34             <div class="col-md-10">
35                 @Html.TextAreaFor(model => model.ShortDescription, 10, 80, new
36                     { htmlAttributes = new { @class = "form-control" } })
37                 @Html.ValidationMessageFor(model => model.ShortDescription, "", new
38                     { @class = "text-danger" })
39             </div>
40         </div>
41
42         <div class="form-group">
43             @Html.LabelFor(model => model.Description, htmlAttributes: new { @class
44                 = "control-label col-md-6" })
45             <div class="col-md-10">
46                 @Html.TextAreaFor(model => model.Description, new { htmlAttributes
47                     = new { @class = "form-control" } })
48                 <script>CKEDITOR.replace('Description', { customConfig: '/Scripts/
49                     ckeditor_config.js' })</script>
50                 @Html.ValidationMessageFor(model => model.Description, "", new
51                     { @class = "text-danger" })
52             </div>
53         </div>
54
55         <div class="form-group">
56             @Html.LabelFor(model => model.GamemasterNotes, htmlAttributes: new
```

```

    { @class = "control-label col-md-6" })
47     <div class="col-md-10">
48         @Html.TextAreaFor(model => model.GamemasterNotes, new
49             { htmlAttributes = new { @class = "form-control" } })
50         <script>CKEDITOR.replace('GamemasterNotes', { customConfig: '/
51             Scripts/ckeditor_config.js' })</script>
52         @Html.ValidationMessageFor(model => model.GamemasterNotes, "", new
53             { @class = "text-danger" })
54     </div>
55 </div>
56 @Html.HiddenFor(model => model.MapFileName, new { htmlAttributes = new
57     { @class = "form-control" } })
58 @Html.HiddenFor(model => model.CreatedDate, new { htmlAttributes = new
59     { @class = "form-control" } })
60 @Html.HiddenFor(model => model.LastModifiedDate, new { htmlAttributes = new
61     { @class = "form-control" } })
62 @Html.HiddenFor(model => model.LastModifiedUser, new { htmlAttributes = new
63     { @class = "form-control" } })
64 @Html.HiddenFor(model => model.IsActive)
65 <div class="form-group">
66     <div class="col-md-offset-2 col-md-10">
67         <input type="submit" value="Save" class="btn btn-primary" />
68     </div>
69 </div>
70 }
71 <hr />
72 <div class="d-flex">
73     <button class="btn btn-danger"
74         data-toggle="collapse"
75         href="#deleteCollapse"
76         aria-expanded="true"
77         aria-controls="deleteCollapse"
78         role="button">
79         Delete Adventure
80     </button>
81     <div class="collapse ml-4" id="deleteCollapse">
82         @using (Html.BeginForm("DeleteItem", "AdventureHome"))
83         {
84             @Html.HiddenFor(model => model.Item_PK)
85             @Html.HiddenFor(model => model.Adventure_PK)
86             <input type="submit" value="Confirm Delete" class="btn btn-outline-
87                 danger" />
88         }
89     </div>
90 </div>
91 <hr />
92 <div>
93     @Html.ActionLink("Back to Item", "ItemDetails", new { id = Model.Item_PK },
94         new { @class = "btn btn-secondary" })
95 </div>

```

```

1  @model AdventureLog.Models.SearchResultsViewModel
2  @{
3      ViewBag.Title = "Adventure Details";
4      ViewBag.SearchToolTip = "Use '#' to search for tags";
5      ViewBag.SearchPlaceholder = "Search for Items...";
6  }
7  <div class="row">
8      <div class="col-lg-12">
9          <h2 class="font-weight-bold">@Model.Adventure.Name</h2>
10     </div>
11     <div class="col-lg-6">
12         <h4 class="text-secondary">Last edited by:
13         @Model.Adventure.LastModifiedUser on
14         @Model.Adventure.LastModifiedDate.ToShortDateString()</h4>
15     </div>
16     <div class="col-lg-12">
17         @using (Html.BeginForm("AdventureSearch", "AdventureHome"))
18         {
19             <div class="input-group" id="adv-search">
20                 <input type="text"
21                     id="search"
22                     name="searchText"
23                     class="form-control col-md-8"
24                     placeholder=@ViewBag.SearchPlaceholder
25                     data-toggle="tooltip"
26                     data-html="true"
27                     data-placement="top"
28                     title="@ViewBag.SearchToolTip" />
29                 <input data-val="true"
30                     id="Adventure_PK"
31                     name="adventure_PK"
32                     type="hidden"
33                     value="@Model.Adventure.Adventure_PK" />
34                 <div class="input-group-append">
35                     <button type="submit"
36                         class="btn btn-primary">
37                         <span class="fa fa-search" aria-hidden="true"></span>
38                     </button>
39                 </div>
40             </div>
41             <script>
42                 $(function () {
43                     $('#search').autocomplete
44                     (
45                     {
46                         minChars: 1,
47                         source: function (term, suggest) {
48                             term = term.toLowerCase();
49                             var choices = [];
50                             @foreach (var item in
51                                 Model.Adventure.Items.Where(i => i.IsActive))
52                                 {
53                                     @: choices.push("@item.Name");
54                                 }
55                             var suggestions = [];
56                             for (i = 0; i < choices.length; i++)

```

```

54         if (~choices[i].toLowerCase().indexOf
                    (term))
55             suggestions.push(choices[i]);
56
57         suggest(suggestions);
58     }
59 }
60 )
61 });
62 </script>
63 }
64 </div>
65 </div>
66 <hr />
67 <div class="p-1">
68     <h6>
69         Result Count: @Model.SearchResults.Count()
70     </h6>
71 </div>
72 <hr />
73 <div class="p-3">
74     @foreach (var result in Model.SearchResults)
75     {
76         <div class="card border-dark mb-3">
77             <div class="card-header">
78                 <h2 class="float-left text-dark">@result.Name</h2>
79                 @Html.ActionLink("See More", "ItemDetails", new { id =
                    result.Item_PK }, new { @class = "btn btn-primary float-
                    right" })
80             </div>
81             <div class="card-body text-dark">
82                 <div class="card-text text-dark mb-2">
83                     Summary: <br>
84                     @result.ShortDescription
85                 </div>
86             </div>
87         </div>
88     }
89 </div>
90 <hr />
91 <div>
92     @Html.ActionLink("Back to Details", "Details",
        Model.Adventure.Adventure_PK, new { @class = "btn btn-secondary" })
93 </div>
94

```