```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace AdventureLog.Models
{
    #region Adventure Information

    public class Adventure
    {
        [Key]
        public long Adventure_PK { get; set; }

        [Required, StringLength(256, ErrorMessage = "Name cannot be longer than 256 characters."), Display(Name="Adventure Name")]
        public string Name { get; set; }

        [MaxLength(256, ErrorMessage = "System Name cannot be longer than 256 characters."), Display(Name = "System")]
        public string SystemName { get; set; }

        [MaxLength(4000, ErrorMessage = "Short Descripition cannot be longer than 4000 characters."), Display(Name = "Summary")]
        public string ShortDescription { get; set; }

        [AllowHtml]
        public string Description { get; set; }

        [AllowHtml, Display(Name = "Gamemaster Notes (Cannot be seen by players)")]
        public string GamemasterNotes { get; set; }

        [MaxLength(100, ErrorMessage ="Invite Password must be less than 100 characters."), Display(Name = "Invite Password")]
        public string InvitePassword { get; set; }

        public bool IsSecured { get; set; }

        public bool IsPublic { get; set; }

        [Required]
        public bool IsActive { get; set; }

        [Required]
        public DateTime CreatedDate { get; set; }

        [Required]
        public DateTime LastModifiedDate { get; set; }

        [Required]
        public string LastModifiedUser { get; set; }

        [InverseProperty("Adventure")]
        public virtual ICollection<Player> Players { get; set; }

        [InverseProperty("Adventure")]
        public virtual ICollection<Item> Items { get; set; }

        [InverseProperty("Adventure")]
        public virtual ICollection<AdventureNote> AdventureNotes { get; set; }
    }

    public class Item
    {
        [Key]
        public long Item_PK { get; set; }

        [Required]
        public long Adventure_PK { get; set; }

        public long? ParentItem_PK { get; set; }

        [Required, StringLength(256, ErrorMessage = "Name cannot be longer than 256 characters."), Display(Name = "Item Name")]
        public string Name { get; set; }

        [MaxLength(4000, ErrorMessage = "Short Descripition cannot be longer than 4000 characters."), Display(Name = "Summary")]
        public string ShortDescription { get; set; }

        [AllowHtml]
        public string Description { get; set; }

        [AllowHtml, Display(Name = "Gamemaster Notes (Cannot be seen by players)")]
        public string GamemasterNotes { get; set; }

        public string MapFileName { get; set; }

        [Required, Display(Name = "Keep Item: WARNING DELETES Item")]
        public bool IsActive { get; set; }

        [Required]
        public DateTime CreatedDate { get; set; }

        [Required]
        public DateTime LastModifiedDate { get; set; }

        [Required]
        public string LastModifiedUser { get; set; }

        [ForeignKey("Adventure_PK")]
        public Adventure Adventure { get; set; }

        [ForeignKey("ParentItem_PK")]
        public Item ParentItem { get; set; }

        [InverseProperty("Item")]
        public virtual ICollection<ItemNote> ItemNotes { get; set; }

        [InverseProperty("ParentItem")]
        public virtual ICollection<Item> ChildItems { get; set; }
    }

    #region Player

    public class Player
    {
        [Key]
        public long Player_PK { get; set; }

        [Required]
        public string UserId_PK { get; set; }

        [Required]
        public long Adventure_PK { get; set; }

        [Required]
        public long PlayerRole_PK { get; set; }

        [Required]
        public bool IsActive { get; set; }

        [ForeignKey("UserId_PK")]
        public ApplicationUser ApplicationUser { get; set; }

        [ForeignKey("Adventure_PK")]
        public Adventure Adventure { get; set; }

        [ForeignKey("PlayerRole_PK")]
        public PlayerRole PlayerRole { get; set; }
    }

    public class PlayerRole
    {
        [Key]
        public long PlayerRole_PK { get; set; }

        [Required, MaxLength(128)]
        public string RoleName { get; set; }

        [Required, MaxLength(256)]
        public string RoleDescription { get; set; }

        [Required]
        public DateTime CreatedDate { get; set; }

        [Required]
        public DateTime LastModifiedDate { get; set; }

        [Required]
        public string LastModifiedUser { get; set; }

        public enum PlayerRoleKey
        {
            Player = 1,
            Gamemaster = 2
        }
    }

    #endregion
    #region Hotspot Information

    /* Removed as the Map Feautre is being postponed for a later version.
    public class Hotspot
    {
        [Key]
        public long Hotspot_PK { get; set; }

        [Required]
        public bool IsActive { get; set; }

        [Required]
        public DateTime CreatedDate { get; set; }

        [Required]
        public DateTime LastModifiedDate { get; set; }

        [Required]
        public string LastModifiedUser { get; set; }

        [InverseProperty("Hotspot")]
        public virtual ICollection<Coordinate> Coordinates { get; set; }
    }

    public class Coordinate
    {
        [Key]
        public long Coordinate_PK { get; set; }

        [Required]
        public long Hotspot_PK { get; set; }

        [Required]
        public int XCoord { get; set; }

        [Required]
        public int YCoord { get; set; }

        [Required]
        public DateTime CreatedDate { get; set; }

        [Required]
        public DateTime LastModifiedDate { get; set; }

        [Required]
        public string LastModifiedUser { get; set; }

        [ForeignKey("Hotspot_PK")]
        public Hotspot Hotspot { get; set;}
    }
    */
    #endregion

    #region Note Information
    public class ItemNote
    {
        [Key]
        public long ItemNote_PK { get; set; }

        [Required]
        public long Item_PK { get; set; }

        [Required]
        public string UserId_PK { get; set; }

        public long? ParentItemNote_PK { get; set; }

        [Required]
        [AllowHtml]
        public string Text { get; set; }

        [Required]
        public bool IsActive { get; set; }

        [Required]
        public DateTime CreatedDate { get; set; }

        [Required]
        public DateTime LastModifiedDate { get; set; }

        [Required]
        public string LastModifiedUser { get; set; }

        [ForeignKey("Item_PK")]
        public Item Item { get; set; }

        [ForeignKey("UserId_PK")]
        public ApplicationUser ApplicationUser { get; set; }

        [ForeignKey("ParentItemNote_PK")]
        public ItemNote ParentItemNote { get; set; }

        [InverseProperty("ParentItemNote")]
        public virtual ICollection<ItemNote> ChildNotes { get; set; }
    }

    public class AdventureNote
    {
        [Key]
        public long AdventureNote_PK { get; set; }

        [Required]
        public long Adventure_PK { get; set; }

        [Required]
        public string UserId_PK { get; set; }

        public long? ParentAdventureNote_PK { get; set; }

        [Required]
        [AllowHtml]
        public string Text { get; set; }

        [Required]
        public bool IsActive { get; set; }

        [Required]
        public DateTime CreatedDate { get; set; }

        [Required]
        public DateTime LastModifiedDate { get; set; }

        [Required]
        public string LastModifiedUser { get; set; }

        [ForeignKey("Adventure_PK")]
        public Adventure Adventure { get; set; }

        [ForeignKey("UserId_PK")]
        public ApplicationUser ApplicationUser { get; set; }

        [ForeignKey("ParentAdventureNote_PK")]
        public AdventureNote ParentAdventureNote {get;set;}

        [InverseProperty("ParentAdventureNote")]
        public virtual ICollection<AdventureNote> ChildNotes { get; set; }
    }
    #endregion

    #endregion
}
```