Marks: 30

Introduction

In this second assignment, you are being tasked to create an MVC Core solution for an Asset Tracking System

Learning Outcomes

Create MVC Controllers
Create MVC Views
Create MVC Models using the Entity Framework
Create various user interfaces with HTML helpers, Tag helps and view models
Create view component
Create applications that are responsive
Enhance user experience using client-side technology
Create HTTP service for browser and mobile clients

Purpose

This assignment will be used to assess your programming skills in developing a multi-layered solution using MVC Core and Entity Framework Core in a code-first approach.

Instructions

Once the assignment is completed, the solution is to be zipped up and uploaded to the Assessments->Assignments section of the course D2L site. The due date for submission is 8:00 am on Tuesday, May 5.

Marking Criteria (50 marks – Weight 30)

REQUIREMENT	MARKS
4 projects used @ 2 marks each	8
2 domain classes coded and used @ 5 marks each	10
Code-first Entity Framework model and database creation	10
Main assets view	6
Add asset view	5
Add asset type view	5
Working asynchronous view component	6
Total	50

The Assignment Scenario

A company has approached you to develop a web-based asset tracking application. They are currently tracking assets in a spreadsheet but as the company has grown so has the need for a web-based tool to help them track their fixed assets.

The assets that are currently being tracked include desktop computers, computer monitors, and phones.

The company would like to see a prototype of what the application would look like by May 5. Due to the time constraints, seeing a page listing all asset types in the system as well a page viewing all assets that can be filtered by asset type is the scope for this project.

Business Requirements

1. The following entities with their properties have been identified in the business domain:
    a. Asset (to be defined in Domain) – all fields required except Model
        i. Id: int
        ii. TagNumber: string
        iii. AssetTypeId: int (plus an AssetType navigation property)
        iv. Manufacturer: string
        v. Model: string (optional)
        vi. Description: string
        vii. SerialNumber: string
    b. AssetType (to be defined in Domain) – Name is required
        i. Id: int
        ii. Name: string
        iii. Assets: ICollection<Asset> (navigation property)

2. The following information was gathered about manufacturers and the types of assets purchased from them as well as what models they supplied:

    Computer manufacturers: Dell, HP, Acer
    Monitors: Acer, LG, HP
    Phone: Avaya, Polycom, Cisco

3. The application requires the following capabilities:
    a. A main home page that displays all the assets in a table. Each line should display the following information:
        i. Asset Description
        ii. Asset Type Name
        iii. Asset Tag Number
        iv. Asset Serial Number
    b. A responsive navigational menu that can collapse to a button when the windows size decreases. Menu items exist for the following application functionality:
        i. Home (main page displaying all assets)
        ii. New Asset (view to add new assets to the system)
        iii. New Asset Type (view to add new asset types to the system)

Technical Requirements

1. Follow a Domain Driven Design (DDD) by starting with the domain model.
2. Create the asset tracking database following the Entity Framework code-first approach.
3. The AssetType table is seeded based on the information in the business requirements section above.
4. Seed the Asset table with at least three test records
5. Create a total of 4 projects in the following recommended order:
    a. Domain (class library -no dependencies)
    b. Data (class library - dependent on Domain)
    c. BLL (class library - dependent on Data and Domain)
    d. AssetTracking (MVC Core Web Application – dependent on Domain and BLL)
6. The Domain project contains the definitions of Asset and Asset Type.
7. The Data project contains the definition of the AssetContext class that inherits DbContext. This project needs the latest version of the Entity Framework added as a NuGet package. The app.config file needs a connection string added to contain the name (use “AssetConnection”), connection string to your server (I will update this connection string so it points to my server when I receive the assignment), and provider name (System.Data.SqlClient). Use migrations in the package manager console window to track the migrations as the code-first process creates the database.
8. The BLL project contains the manager classes that the controllers will use. One manager per domain entity

9. The main MVC project requires the following:
    a. Controllers for each domain entity. Action methods correspond to the views
    b. ViewModels defined for each view.
    c. Views for each of the menu items described earlier
        i. The main assets view needs the following functionality:
            1. Displays all assets in the system by default
            2. Filter the display to display assets by asset type. The asset type is displayed in a drop down list
            3. Use buttons or links to initiate the filtering
            4. The filtered data is all displayed asynchronously in a view component
        ii. The new asset view is used to add an asset to the system. Ensure it has the following functionality:
            1. The Asset Type selected from a drop down list
            2. The ViewModel contains the Asset properties and collection of SelectListItems for the asset type drop down list
            3. Selections are posted back to the controller for adding to the system. If successful, redirect back to the main home view
        iii. The asset type view is used to add asset types.
    d. The default master page (_Layout) needs to be modified to accommodate the following:
        i. A fixed navigation menu that is collapsible
        ii. Optional image and colors of your choice
        iii. A simple footer with copyright for your company fixed to the bottom of the page

