CPRG 102 Final Assignment
Asset Tracking System

Marks: 100 **Weight:** 50%

Learning Outcomes

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

Purpose

In keeping with the previous courses in the .NET certificate program, this final assignment will assess your programming skills in developing an MVC application with a Web API service across multiple software layers using Entity Framework.

Instructions

You may start working on the project any time, but it is recommended that you start no later than the midway point of the course. You will do all work outside of class. Once you have completed the assignment, zip the solution and upload it to the Assignment section of the SAIT online learning site. The due date for submission is 11:59 p.m. on the final day of the course.

Marking Criteria

Marks will be awarded based on successful implementation of the following items:

- 5 projects @ 3 marks each = 15 marks
- 4 domain classes @ 5 marks each = 20 marks
- Code-first Entity Framework model and database creation = 10 marks
- Web API service = 10 marks
- Main assets view = 15 marks
- New asset view = 5 marks
- Assignments view = 5 marks
- HR Entity Framework model = 10 marks
- Use of jQuery/Ajax/Bootstrap = 10 marks

The Assignment Scenario

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

The assets currently being tracked include desktop, laptop and tablet computers, computer monitors, iPhone and Android cellphones, and IP-enabled desk phones.

The company has an existing HR database that cannot be modified but can be accessed to get employee information required by the system under development. A new database will need to be developed for tracking assets based on the business requirements below. The company has departments located in various cities:

- Administration: Calgary
- IT: Calgary
- Engineering: Edmonton
- Sales: Vancouver

Business Requirements

1. The following entities with their properties have been identified in the business domain:
   a. Employee (to be defined in HRService)
      i. Id: int
      ii. EmployeeNumber: string
      iii. FirstName: string
      iv. LastName: string
      v. Position: String
      vi. Phone: string
      vii. DepartmentId: int
   b. Department (to be defined in HRService)
      i. Id: int
      ii. Name:string
      iii. Location: string

c. Asset (to be defined in Domain)
   i. Id: int
   ii. TagNumber: string
   iii. AssetTypeId: int (plus an AssetType navigation property)
   iv. ManufacturerId: int (plus a Manufacturer navigation property)
   v. ModelId: int (plus a Model navigation property)
   vi. Description: string
   vii. AssignedTo: string (the employee number from the HR database)
   viii. SerialNumber: string

d. Manufacturer (to be defined in Domain)
   i. Id: int
   ii. Name: string
   iii. Assets: ICollection<Asset> (navigation property)
   iv. Models: ICollection<Model> (navigation property)

e. Model (to be defined in Domain)
   i. Id: int
   ii. Name: string
   iii. ManufacturerId: int
   iv. Assets: ICollection<Asset> (navigation property)

f. AssetType (to be defined in Domain)
   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:

   Desktop PC
   - Dell Inspiron
   - Dell XPS
   - HP Elite
   - Acer Aspire

   Laptop
   - Dell Latitude E4550
   - Dell Latitude E5550
   - Apple MacBook Air
   - Apple MacBook Pro

Tablet
- Apple iPad mini
- Apple iPad Air
- Samsung Galaxy Tab3

Monitor
- Acer S200
- Acer STQ414
- LG 22MP
- HP Pavilion

Mobile Phone
- Apple iPhone 5
- Apple iPhone 6
- Samsung Galaxy S4
- Samsung Galaxy S5
- Samsung Galaxy Note5

Desk Phone
- Avaya 9612G
- Polycom SoundPoint 331
- Cisco SPA525G2

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
      v. Employee Name
      vi. Department Location
   b. A navigational menu that can collapse to a button when the window size decreases. Menu items exist for the following application functionality:
      i. Home (main page displaying all assets)
      ii. New Asset (page to add new assets to the system)
      iii. Assignments (page to assign/reassign/remove assets attached to an employee)

Technical Requirements

1. Create the HR database based on the script provided. It is a stand-alone database and needs to be accessed only from a service application.

2. Create the WebAPI service that will access the employee data required from the HR database.

3. Follow a Domain Driven Design (DDD) by starting with the domain model.

4. Create the asset tracking database following the Entity Framework code-first approach. The AssetType, Manufacturer and Model tables are seeded based on the information in the business requirements section above. There is no need to provide any controller methods or views to manage these tables. They are only static lookup tables.

5. Create a total of five 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. HRService (WebAPI – no dependencies)

   e. AssetTracking (MVC Web Application – dependent on Domain and BLL)

6. The Domain project contains the definitions of Asset, Asset Type, Manufacturer and Model classes.

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 (the instructor will update this connection string so it points to his/her server when the assignment is received), 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 HRService project is a Web API project. Use the supplied database script to create the database in SQL Server. Use a database-first Entity Framework model. The ApiControllers will use the Entity Framework context to retrieve the data required by the rest of the application. All ApiController methods return JSON results. The data being returned is all employee based. Use the employee number instead of the id for identifying employees. Define a controller for each entity. There is no need to manage that database in the

application. Assume the database belongs to the administration department and you are only accessing it through this Web API service.

10. 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 (assigned and unassigned) in the system by default
            2. Filter the display to display only assets that are assigned
            3. Filter the display to show only assets that are not assigned
            4. Filter the assets assigned to a selected employee
            5. Filter the assets to show only assets of a selected asset type
            6. Use buttons or links to filter
            7. Filtered data is all displayed asynchronously
            8. Use the Bootstrap grid system to display the links/buttons and the employee and asset type drop-down lists in one content area and the table in the other. They display in a horizontal row when the window size is large and will stack as the windows size decreases.
        ii. The new asset view is used to add an asset to the system. Ensure it has the following functionality:
            1. The Asset Type, Manufacturer, Model and Employee are all selected from a drop-down list.
            2. The ViewModel contains the Asset properties and collections of SelectListItems for the drop-down lists.
            3. Selections are posted back to the controller for adding to the system. If successful, redirect back to the main home view.
        iii. The assignments view is used to assign unassigned assets to a selected employee. The view requires the following functionality:
            1. A list box displays only employees who have not been assigned any assets.
            2. Drop-down lists for assets of that asset type. A selection is made from each of the drop-down lists.
            3. Selections are posted back to the controller so each asset can be updated with the assignment to that employee.
            4. Use the Bootstrap grid system on this form also, so each of the drop-downs is placed in column of the grid. The grid only has one row.
    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 of your choice
        iii. A simple footer with copyright for your company