Solution Overview

Technology used:

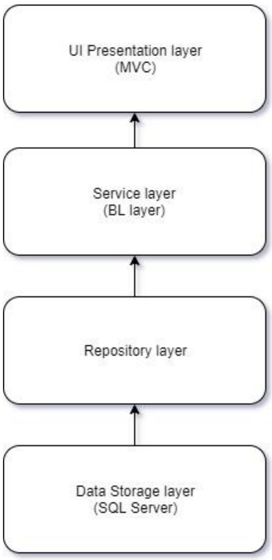
1. ASP.NET MVC.
2. Microsoft SQL Server.

Project Architecture:

The solution project architecture is mainly database centric architecture, it consists of N-Tier layers.

Each layer is a class library encapsulates the logic and the main functionality separated form other layers, the following diagram refers to the layers.

The last layer it the MVC layer responsible for handling the incoming requests from the user and render the appropriate view.



a) Data Storage layer (ChemiClean.Core) contains the generated database first ADO.NET EDMX file, so that it contains the DBContext and the models for the tables.

b) Repository layer (ChemiClean.Repository) contains the IBaseRepository interface that has the contract for basic retrieving and modifying data form dbSet methods, and BaseRepository implement the IBaseRepository, and other repositories for each entity in the DBContext.

c) Service layer (ChemiClean.Services) main operation is to extend the logic of the repository layer, so it contains the business logic for the project, also each entity in service layer depends on an interface to encapsulate the methods for this entity, and a service that implements the interface.

d) UI layer (ChemiClean.Web) main concern is to handle the incoming request, using controllers and the views, the controller has the appropriate service injects in to it using dependency injection.

e) All layers applies the dependency inversion principals and uses dependency injection in the constructor, the dependencies are resolved using Autofac.

Project running setup:

1. Restore the attached database or run the attached generation script to create a database with the tables and data in it, please note that the database name is critical as it is used in the project connection string -database name is ChemiCleanDB-.

2. Open project solution.

3. Build solution and Run.

Point to Be Enhanced and Improved:

1. Use ViewModels instead of the database model in the views with mappers like AutoMapper, this will separate the database models used for the database from the models (ViewModels) used for displayed in the view.

2. The Sync operation that process the displayed products documents in the current displayed page in paging may takes some time, so it can be run over a separate Thread to not hold the user to wain until it finishes, and then notify the user with the result with notification.

3. Write more unit-testing cases to cover more scenarios.