**Brewery and wholesale Management Assessment**

*By Mark Choueiry*

1. **Overview**

In this assignment, I have chosen to work with MVC Architecture using “*ASP.NET Web Application”* template.

In order to complete this assessment, I will be working with a local database and using entity framework code first approach.

For the controller part, I will be using Web APIs with Rest architecture. Please note that an additional console application with some testing methods will also be provided.

In the end, for unit testing I will be using “*Nunit*” with “*Moq”* in order to simulate the DB Context.

1. **Models**

For the modes part we will have 6 classes listed as below:

* Brewery
* Beer
* WholeSaler
* Stock
* Quote
* Sale

These classes will contain the respective fields and the business logic in case found. The details of each class can be found below:

* Brewery: The brewery class will contain two properties ID which will be the primary key in the DB and name which will represent the name of the brewery
* Beer: The beer class will contain five properties; the ID which will be the primary key in the DB, the brewery id which is a foreign key that links the beer to the brewer, the alcohol content which is a descriptive property and the price which represent the price of this type of beers
* WholeSaler: the wholesaler class will contain two properties ID which will be the primary key in the DB and name which will represent the name of the wholesaler
* Stock: The stock class will contain four properties; the ID which will be the primary key in the DB the Beer ID which will represent the type of beer in the stock, the wholesaler ID which will represent this stock is related to which wholesaler and the quantity which will represent the number of available beers in the stock
* Sale: The sale class will contain three properties; the ID which will be the primary key in the DB, the Date of the sale and the Order ID which will contain the details of the order
* Order: The order class will contain seven fields; the ID which will be the primary key in the DB, the wholesaler ID which will represent the wholesaler related to this order, the beer ID which will represent which beer is being bought, the Quantity which represents how many beers did the customer order, the original amount which represents the amount before discount, the discount percentage and the total amount after the discount if applicable. This class will also have a method called calculate discount which will calculate the needed discount for the order if applicable.

1. **DAL**

This layer will have two classes which will be used to create the Database and fill it with temp data.

The first class will be called “*ManagementSystemContext*” which will be used in order to create the controllers.

The second class will be called “*ManagementSystemInitializer”* which will be used in order to pre-fill the database with some sample data

1. **Controllers**

The controller’s layer will be representing the Web APIs that will be consumed by the clients in the future. We will have seven controller classes each one representing one class from our model layer listed as follow:

* BreweriesController
* BeersController
* WholesalersController
* StockController
* SaleController
* QuoteController

As per the requirements of this assessment some controllers might only have the default generated methods, in case we will not be using them in our work.

All the methods that we will be using will have a custom name and custom routing for example: “api/Beers/GetBeerByBrewery/{id}”

1. **Unit Testing**

As mentioned in our overview, for unit testing we will be creating a class library called UnitTests. This library will be responsible for all the unit tests by using NUnit framework and Moq in order to create the needed mocks.