Please Consider the following

ModelBinder at Server’s end was created and implemented to exactly abide by the given specifications however it broke this early morning and the code had to be refactored so that a same class is shared between the client and server for posting the data. If time was available it could have been fixed.

# **Technology and Tools**

* WPF
* ASP.NET Web API 2
* Ninject
* Entity Framework 6, Code First Migrations (Seeding done)

# **How it works**

The server runs at remote.

The application when opened checks if the server is running. The address of server is stored inside the config folder at client application.

If the server is not available application assumes the address might have changed and prompts the user to provide the correct address.

If server is up and running, the user is prompted for username and password which are then sent to server for basic authentication.

The valid user is provided back with his/her username, password, fullname and address (location information)

This fullname and location information is used to post the transactions back to server.

For every contact to server username and password is sent and it has been enforced with a filter HLeisureAuthorizeFilter.cs.

The Server sends back the sales id to the client which is displayed through messagebox. This is a deviation from the provided spec as the spec suggests the sales id be sent in json as well which could have been easily implemented but I thought it would be wise for the sale id to be provided by the server.

# **Authentication**

Simple HTTP Authentication has been used.

# **Database**

SQL Server 2014 was used.

Three tables have been created for product master, sales master and sales detail.

Products and Sales Master has many-to-many relationship and the detail of individual sale is shown in sales detail table.

# **Client**

There is a config folder in the client inside which hostConfig.dat file is located which stores the address of the server.

If the client cannot communicate with the server then it allows the user to edit the content of this file through the interface.

A class will be shared with the clients according to which the client shall develop the application.

There have been some simple modification in the json provided as simplicity was expected and clarity was desirable.

User Authentication is developed by myself to make it simple as it was also an optional element. Basic user information is kept and during authentication user's fullname and location address is passed which is later sent back to the server.

# **Testing**

Dependency Injection

Ninject is used for dependency injection. It is required so that the code can be tested.

To run the test you will have to edit the serverPath.dat file located in the test projects inside the config folder.

Unit Tests have been written for posting of data, getting product list and authenticating of users.

# **Limitations**

Product stock has not been checked during transaction

Unit Tests are not developed for client end

Versioning is not used as the project is for the test of coding only

# **Deviation from specification**

The program was developed inline with the specs. The modelbinding was used (please see the ProductModelBinder.cs file inside the AppStart folder to confirm) and unfortunately whole code had to be refactored to make it work.