**Case Study for WPF application.**

**Summary:**

WPF application has developed for doing CRUD Operation using Restful service with Unit Test. It is developed by .netCore 3.1 with Web API to consume service. In this application has SOLID Principles, Design Pattern and Three-tier architecture are followed to make application to understandable manner. Proper Name convention has been used for Class, Methods and Controls.

**The Following Points are to understand the application and there layers:**

**Layers in application:**

Totally there are 5 Projects in this application with Unit Test. The Layer’s are :

**Projects:**

**1. EmployeeDetailsUI** 🡺 This project has only UI part of this application.

**2. BusinessLayer** 🡺 Contains only business logic like button click evets. Get the values from UI and it has send the values to data access layer.

**3. DataAccessLayer** 🡺 This layer has access the data from **ReST End Point.**

**4. Entities** 🡺 It has entities data like Name, Id and Gender etc. Are present in this layer.

**5. Common Layer** 🡺 Apart from the common logic like getting the data from XML or JSON file and any common functionality are applied in this layer.

**Unit Test:**

**UnitTestProject** 🡺 It is has to perform the unit for the API end point and common methods are working properly or not using **MS Unit Test.**

**Definition for Layers:**

1. **EmployeeDetailsUI :**

In this Project we have only UI presentation layer. UI layer has five text Boxes ID, Name, Email, Gender and Status used for create or Update Details. DataGridView is used to view all the details in the grid view itself it has Edit and Delete option. Export button has in top of the grid to export the details in CSV file. To search individual details we have SearchById button to get the one record. Also have Update and Create button to Create new employee or update the Existing employee.

1. **BusinessLayer:**

Business Layer is to perform only the business action it act as model layer for is to perform to get the data from UI layer and send it into Data access layer. It is connection between UI and Data Access layer, the code separation has used in Singleton pattern.

1. **Data Access Layer :**

In the name itself having Data, the purpose of this layer is establish the connection between API End Point to consume and fetch the data from End Point. To consume and fetch the data used Async methods operation to synchronously fetch and send the data to end point.

For communicate the API we need the information like Base Address, Endpoint and Access Token. To access the values directly may causes the security violation in application. To avoid this we have settings.xml file to store the value of all the above three value and to get those value we have used XMLDocument method and we passed the value to respective action methods.

1. **Entities Layer :**

All the application has entities, it is nothing but the data for application to process. Like a Person details like Name, Gender, Age or an object are the entities. We used EmployeeEntites.

1. **Common Layer :**

This Layer is used for perform the common operation for an application. In the common layer we can use the logic which is to perform in all the projects can be done on this layer. Here we used get methods to fetch the data from XML. Apart from we can do validation part and done on this layer and use on other projects as well.

**UNIT TEST:**

**UnitTestProject :**

MS Unit Test is implemented to test the API Methods are working.

**Comments Raised in Mail:**

1. **Not Able to run application:** Now all the Project has been created with Separate Repository in GitHub with this Link.

<https://github.com/ARVINDBALAN/EmployeeDetails>

1. **Unit Test Project Not Loaded:** Issues resolved able, now you can load unit test project.
2. **Remove Unwanted Branches:** All the unwanted branches has removed in repository.
3. **Brief Info about target Framework:** Include Detail explanation about project file in ReadMe file in repo and added the document file for brief on project.
4. **Implement Dependency Injection:** SOLID Principles has implement, create interface, layer separation and Codes are loosely couple. Facing some conflict issues while making DI. **Working on it same to implement it.**
5. **Naming Convention :**
6. Using Camal and Pascal casing for Member, Methods and Classes as per the standards.
7. Aligned name for Class and business layer with common terms instead of using technical terms.
8. **Creating Different project for UI and Business Layer:** Layer Separation has implement for all the Logical and UI Layer with respective operations.
9. **Implement Design Pattern**: Implemented design pattern like introduce interface. Trying to improve the pattern design all the application.