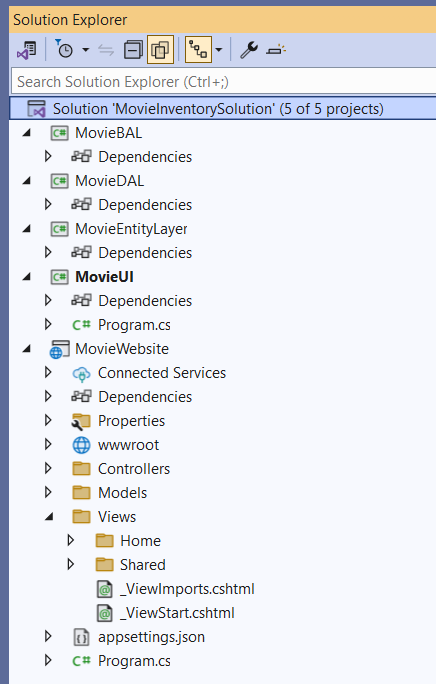
1. Create an Blank Solution in Visual Studio.

name it "MovieInventorySolution"

2.More then One projects will be there:

ConsoleUI Console App

WebUI ASP.NET Web Application

EntityLayer Class Library Project

MovieBAL Class Library Project

MovieDAL Class Library Project

MovieInventoryDB SQL Server Database

**All the tables will be having Id Column and that is an int type with Identity true and 1,1 Seed and Increment**

**Whenever we will use Delete in any menu option the respective record should be stored in archive table.**

**Requirement 1:**

1. Use Entity Layer and create a **Movie** Class with the following attributes:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_name | string |
| \_boxoffice | double |
| \_budget | double |
| \_rating | double |

1. Mark all the attributes as private.
2. Create / Generate appropriate properties.
3. Add a default constructor and a parameterized constructor to take in all attributes in the given order:   
   **Movie ( string \_name, double \_boxoffice, double \_budget, double \_rating ).**
4. When the “Movie” object is printed, it should display the following details: **[Override the ToString method].**  
   Print format:  
   Name: "\_name"  
   Box Office: "\_boxoffice"  
   Budget: "\_budget"  
   Rating: "\_rating"

Two Movies are considered same if they have the same name, and rating. Implement the logic in the appropriate function. (Case – Insensitive) **[Override the Equals method]**

**Requirement 2:**

In the same Entity Layer create a Class Cast with the following properties:

|  |  |
| --- | --- |
| **Member Field Name** | **Type** |
| \_name | string |
| \_nationality | string |
| \_gender | string |
| \_dob | DateTime |

Mark all the Properties as private,  
Create / Generate properties Getters & Setters,  
 Add a default constructor and a parameterized constructor to take in all properties in the given order:  
    Cast( string \_name, string \_nationality, string \_gender, DateTime \_dob )

**Requirement 3:**

Use MovieBAL and Create a class **CastBO**with the following methods,

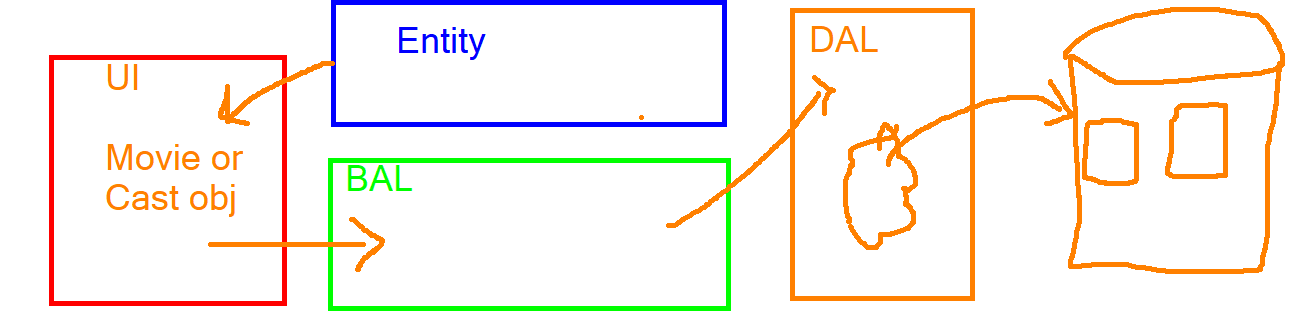
|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public List<Cast> FindCast(List<Cast> castList, string nationality) | This method accepts a list of casts and nationality as arguments and returns a list of casts that match with the given nationality. |
| public List<Cast> FindCast(List<Cast> castList,DateTime dob) | This method accepts a list of casts and dob as arguments and returns a list of casts that match with the given dob. |

The cast details should be given as a comma-separated value in the below order,  
\_name, \_nationality, \_gender, \_dob

**Requirement 4:**

Use MovieBal and Create a class MovieBO with following methods

|  |  |
| --- | --- |
| **Method Name** | **Description** |
| public static Movie CreateMovie(string detail) | This method accepts a string which contains movie details separated by commas. Split the details and create a movie object from the details and return it. |
| public static List<Movie> GetTopFilms(List<Movie> movieList) | This method accepts movieList as an argument and returns the top 5 films list based on the rating provided to that film. |
| public List<Movie> GetSuccessMovies(List<Movie> movieList) | This method accepts a list of movies as arguments and returns a list of successful movies if none of the movie is successful return **empty list**.A movie will be successful if profit is greater than or equals to half the budget.Profit is  calculated using boxoffice and budget |

  
**Requirement 5:**

* 1. Create plan for SQL Server Database
  2. Create on ER Diagram
  3. Sample Data

**Requirement 6:**

In MovieDal project

Create on SQL Connection class (Helper type)

Class for MovieDAO

Responsible for performing CRUD over Movie Table

Class for CastDAO

Responsible for performing CRUD over Cast Table