FSIS– Client Server Setup

*Client-Server*

Setup a new Client-Server solution in **Visual Studio 2019** with the following characteristics.

* Solution Name – **Fort Sasquatch Indoor Soccer**
  + All of the following projects (Class Library and Web Application) are physically placed in the Solution folder.
* Web Application Folder Name – **BigFootWebApp** 
  + Set Framework to .Net Framework 4.6.2 +
  + Create a folder call ExercisePages. This will contain all the web forms that will be developed for the FSIS exercises. Add a web form to this folder called **ExerciseHome.aspx**. Add a title banner on the web form (**Fort Sasquatch Indoor Soccer)**, an image of the database ERD and header tag with the text Known Problems. As you do your exercises, you will indicate under this Known Problems tag, any known problems related to a specific exercise.
* Class Library Name – Include the first character of your first name along with the first four characters of your last name immediately after the “FSISSystem” portion of your class library names.

For example, if your name is “Dan Gilleland”, then your class library project would be named “*FSISSystem.DGill*”

* + Server Project (for BLL and DAL) – **FSISSystem.*FLast***
    - Set Framework to .Net Framework 4.6.2 +
    - BLL folder
    - DAL folder
    - Entities folder
* Adding References
  + Web Application
    - Project: FSISSystem.*FLast*
* Replace your default site.master menu with the nav-bar menu demonstrated in class. Add 3 root level nav-items for the give default page, about page and contact page generated when the web application was created. Add a drop down menu called Exercises. Under this drop down menu, add a nav-item associated with the default web page for your Exercises (ExerciseHome.aspx).

FSIS\_Database



Checking of the Client/Server exercises will be done in class.

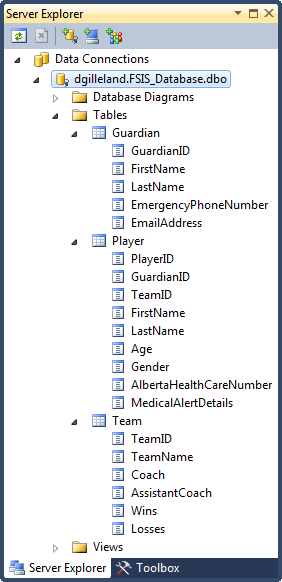
**Checking Guide**

|  |  |
| --- | --- |
| **** | **Item** |
|  | 1. Solution has **correct name** |
|  | 2. Class Library has **correct name** |
|  | 3. Class Library’s Assembly & Namespace properties are correct (i.e. – the names of the project were made when the project were ***created***) |
|  | 4. Server project class Library has folders for **BLL** and **DAL** and **Entities** |
|  | 5. Web Application has **correct name** and is the **Startup Project** |
|  | 6. Web Application **references** the Class Library |
|  | 7. Web Application has the ExercisePages folder and web page setup |
|  | 8. Default site.master menu has been replaced with the class demonstrated nav-bar menu. |
|  | 9. Solution, Web Application and Class Library are in the same physical folder. |
|  | 10. Solution complies with no errors |
|  | 11. Menu works and displaying 4 pages (Default, About, Contact and ExercisePages/ExerciseHome) |
|  |  |

FSIS-Client Server Entities

Entities for Database Tables

Create the following Entity classes for the tables from the database. Place your entities in the Entities folder of your class library. View your tables in your sql server to verify datatype and nullability. Be sure to

* use proper standards for class and property names (e.g.: TitleCase);
* select the proper data type for all properties, based upon the SQL data type of the corresponding table column AND whether or not the table column supports database ***Null*** values;
* no constructors required

You must create Entity classes for the following. Ensure you use appropriate Entity Framework notation. Add the System.ComponentDataModel.DataAnnotations to the application library project.

* **Guardian**
  + An extra read-only not mapped property called **FullName** that is in the format of “*LastName*, *FirstName*”
* **Player**
  + **Gender** must be a fully-implemented property that ensures the Set method converts **value** to upper-case.
  + **MedicalAlertDetails** must be a fully-implemented property that ensures the background field is set to **null** if an empty string is in **value**.
  + An extra read-only not mapped property called **FullName** that is in the format of “*LastName*, *FirstName*”
* **Team**
  + No additional requirements

### Checking Guide

|  |  |
| --- | --- |
| **✓** | **Item** |
|  | Each class has the correct property names and data types. |
|  | Necessary notation is in place. |
|  | Nullable types (if any) are correctly applied to the appropriate  properties |
|  | Selected property created in the requested implementation. |
|  |  |

FSIS-Client Server Context Setup

DAL Layer – Content Class

1. Using Manage NuGet Packages, add *EntityFramework* (version 6.2.0 +) to your FSISSystem and web application projects. Add a reference to System.Data.Entity to the projects.
2. Create your Data Access Layer class FSISContext which inherits DbContext. Create the default constructor which will assign the base value of “FSIS\_db” for DbContext class. Add a property using the datatype DbSet<T> for each of your entities. Access type for this class will be “internal”.
3. Configure your solution’s knowledge of the database by making the following changes to your system.
   * Edit the web application’s *web.config* file to have the following for the <connectionStrings> tag  
     <connectionStrings configSource="WebConnectionStrings.config"/>
   * Add a new configuration file to the root of the web application and name it “*WebConnectionStrings.config*”; on the inside, put the following code  
     <connectionStrings>  
      <add name="FSIS\_db"  
      connectionString="Data Source=.;Initial Catalog=FSIS\_Database;Integrated Security=True;"  
      providerName="System.Data.SqlClient"/>  
     </connectionStrings>
   * Edit the web application’s *web.config* file to have the following tag <contexts> added to the <entityFramework> tag. Replace *FLast* appropriately to match your application library project.   
     <contexts>

<context disableDatabaseInitialization="true"  
 type="FSISSystem.*FLast*.DAL.FSISContext,FSISSystem.*FLast*">

<context>  
</contexts>

### Checking Guide

|  |  |
| --- | --- |
| **✓** | **Item** |
|  | The provided **files** and package **libraries** are correctly added to the project |
|  | References have been properly set up. |
|  | Context class has been properly set up. |
|  | Correctly modified *web.config* file and correctly added/edited the *WebConnectionStrings.config* file. |
|  | Correctly modified *web.config* file and correctly added the entityframework *<contexts>* tag. |