**INSTRUCTIONS**

**Please read the instructions carefully before doing the questions.**

* You are **NOT allowed** to use any other materials. You are **NOT allowed** to use any device to share data with others.
* You must use IDE as **Visual Studio 2019 or later, MSSQL Server 2016 or** **later database** for your development tools.

**IMPORTANT – Before you start doing your solution, MUST do the following steps:**

1. Create Solution/Project in Visual Studio named **PRN211PE\_SU22\_StudentFullName**.

For Instance: **PRN211PE\_SU22\_NguyenThuyLinh**

1. You must use Windows Forms application using 3-Layers architecture.
   1. In the layer Presentation (Windows Forms), student create **appsettings.json** and add ConnectionString same as the bellow to config connect to SQL Server Database.

-=============================

*{*

*"ConnectionStrings": {*

*"BankAccountTypeDB": "server =(local); database=****BankAccountType****;uid=****sa****;pwd=****sa****;TrustServerCertificate=True"*

*}*

*}*

-=============================

Note: Student can change **uid** and **pwd** to advaiable in the local machine

b. Set property "Copy to output Directory" of appsettings.json file to "Copy if newer"

c. Using Manage Nuget packages to install packages the following in Windows Form project

Microsoft.Extensions.Configuration -Version 5.0.0

Microsoft.Extensions.Configuration.Json -Version 5.0.0

d. Using *ConfigurationBuilder* to init Configuration object for reading appsettings.json file same as this code:

-=========================

private string GetConnectionString()

{

IConfiguration config = new ConfigurationBuilder()

.SetBasePath(Directory.GetCurrentDirectory())

.AddJsonFile("appsettings.json",true,true)

.Build();

var strConn = config["**ConnectionStrings**:**BankAccountTypeDB**"];

return strConn;

}

-=========================

e. After that, durring development, student can bypass the ConnectionString (which read from appsettings.json) to Data access layer by constructor or other ways

For example:

-=========================

public partial class BankAccountTypeContext: DbContext

{

public BankAccountTypeContext (string connectionString)

{

this.Database.SetConnectionString(connectionString);

}

}

-=========================

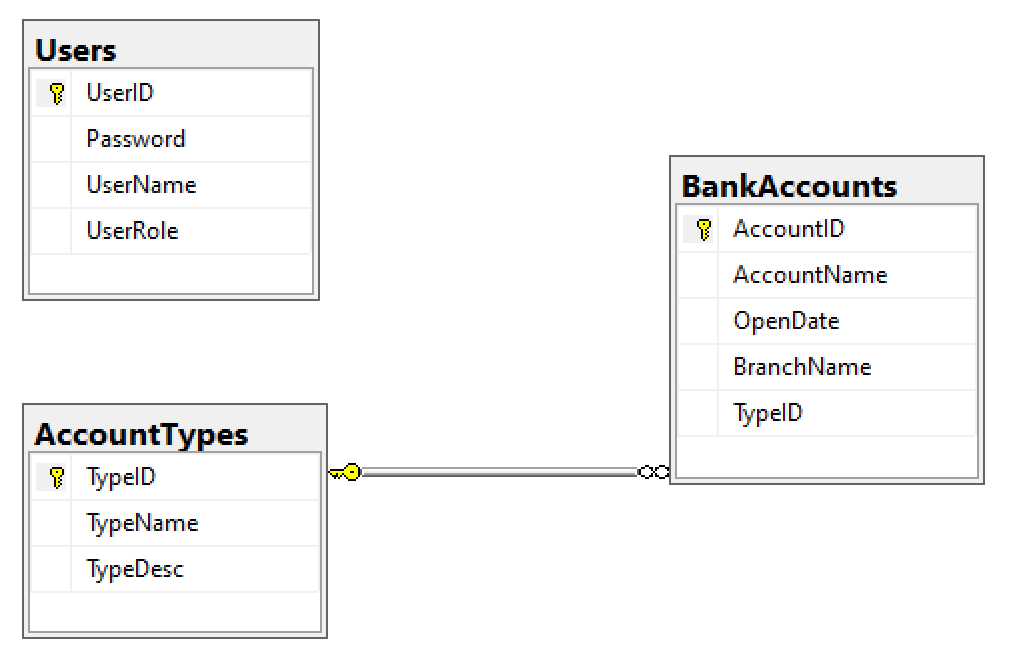
1. Create your MS SQL database named**BankAccountType** by running code in script **BankAccountType.sql.**

Note: The script only run one time.

1. Set the default form for your project as **Login** form.

Create an application using **Windows Forms** with .NET Core (.NET 5), C#, and Entity Framework Core ( or ADO.NET). A MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

**BankAccountType** **management database (1:n):**

****

*User role: Administrator = 1; Manager = 2; Staff = 3*

***Note***

**Package using for .NET 5:**

*- Install package using Tools → NuGet Package Manager → Package Manager Console*

Install-Package Microsoft.EntityFrameworkCore.SqlServer -Version 5.0.17

Install-Package Microsoft.EntityFrameworkCore.Tools -Version 5.0.17

Install-Package Microsoft.EntityFrameworkCore.Design -Version 5.0.17

Install-Package Microsoft.Data.SqlClient -Version 3.0.1

- *Install package using CLI or Power Shell*

dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 5.0.17

dotnet add package Microsoft.EntityFrameworkCore.Design --version 5.0.17

dotnet add package Microsoft.EntityFrameworkCore.Tools --version 5.0.17

dotnet add package Microsoft.Data.SqlClient --version 3.0.1

Entity Framework Core

*- Install dotnet-ef for CLI*

dotnet tool install --global dotnet-ef --version 5.0.17

*- Use Entity Framework Core to generate Object Model from existing database – CLI*

dotnet ef dbcontext scaffold "Server= DESKTOP-QKQ1PQ7\SQLEXPRESS;uid=sa;pwd=sa;database=BankAccountType;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer --output-dir Models

*- Generate database from domain classes – CLI.*

dotnet ef migrations add "InitialDB"

dotnet ef database update

Entity Framework Core

*- Use Entity Framework Core to generate Object Model from existing database – Package Manager Console*

Scaffold-DbContext "Server=(local);uid=sa;pwd=sa;database= BankAccountType;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

*- Generate database from domain classes – Package Manager Console*

Add-Migration "InitialDB"

Update-Database -verbose