

Assignment 02

Building a Sales Management Application with Windows Forms

Introduction

Imagine you're an employee of a product retailer named **FStore**. Your manager has asked you to develop a Windows Forms application for member management, product management, and order management. The application has a default account whose email is “**admin@fstore.com**” and password is “**admin@@**” that stored in the **appsettings.json**.

The application has to support adding, viewing, modifying, and removing products—a standardized usage action verb better known as Create, Read, Update, Delete (CRUD) and Search. This assignment explores creating an application using Windows Forms with .NET Core, C#, and ADO.NET / Entity Framework. The MS SQL Server database will be created to persist the data and it will be used for reading and managing data.

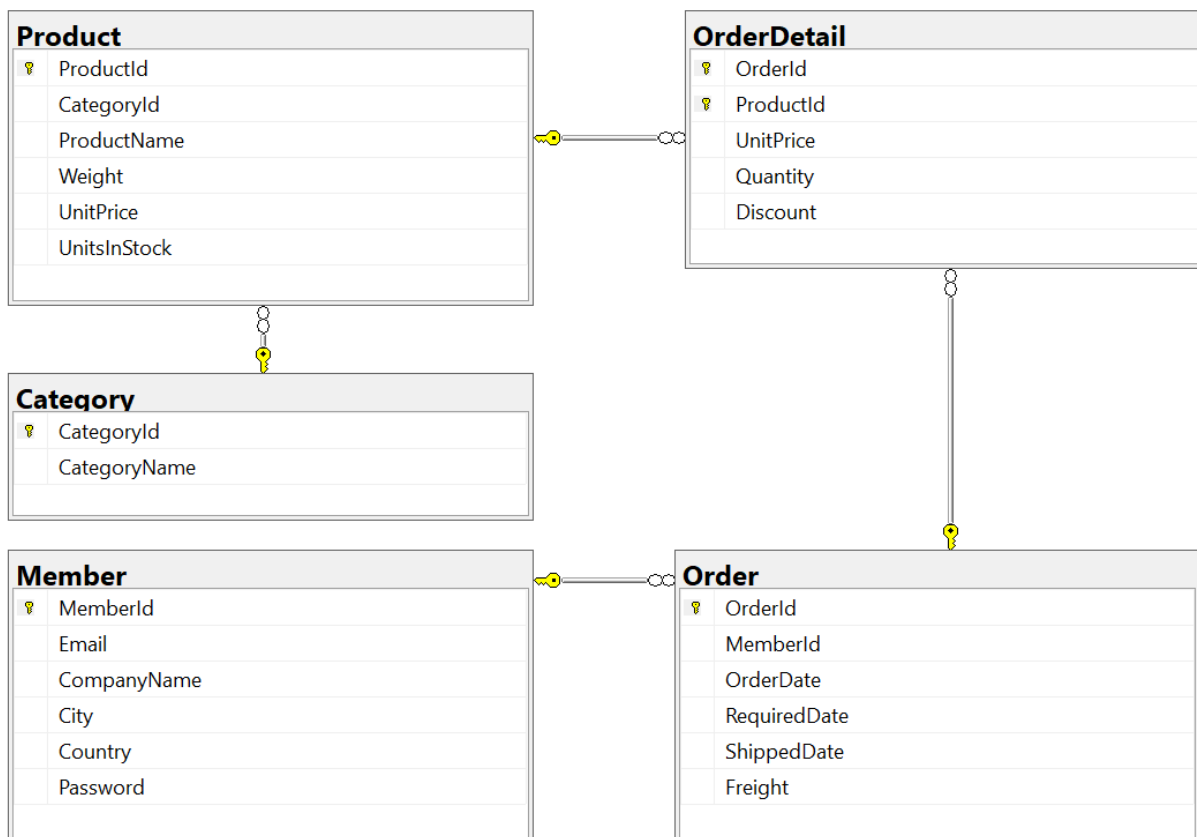
Assignment Objectives

In this assignment, you will:

- Use the Visual Studio.NET to create Windows Forms and Class Library (.dll) projects.
- Develop MDI (Multiple Document Interface) application using WinForms.
- Perform CRUD actions using ADO.NET and Entity Framework Core
- Use LINQ to query and sort data
- Apply passing data in WinForms application
- Apply 3-layers architecture to develop an application
- Apply MPV (Model-Presenter-View) pattern in WinForms application
- Apply Repository pattern and Singleton pattern in a project
- Add CRUD and searching actions to WinForms application.

- Apply to validate data type for all fields
- Run the project and test the WinForms actions.

Database Design



Main Functions

- Member authentication by Email and Password. If the user is “**Admin**” (*get from appsettings.json file*) then allows to perform all actions, otherwise, the **normal user** (*get from the Member table in database*) is allowed to view/create/update the profile and view their orders history.
- Member management, Product management, and Order management: Read, Create, Update and Delete actions. Creating and Updating actions must be performed by popup dialog. *One order will have many products; one product will have in many orders.*
- Search Product by ID, ProductName (by keyword of ProductName), UnitPrice, and UnitInStock.
- Create a report statistics sales by the period from StartDate to EndDate, and sort sales in descending order.

Guidelines

Activity 01: Build a solution

Create a Blank Solution named *Ass02Solution_ClassCode_StudentName* that includes Class Library Project: **DataAccessObjects**, **BusinessObjects**, **Repositories**, and a Windows Forms project named **SaleManagementWinApp**

Step 01. Open the Visual Studio .NET application and create a Blank solution named **Ass02Solution_ClassCode_StudentName**

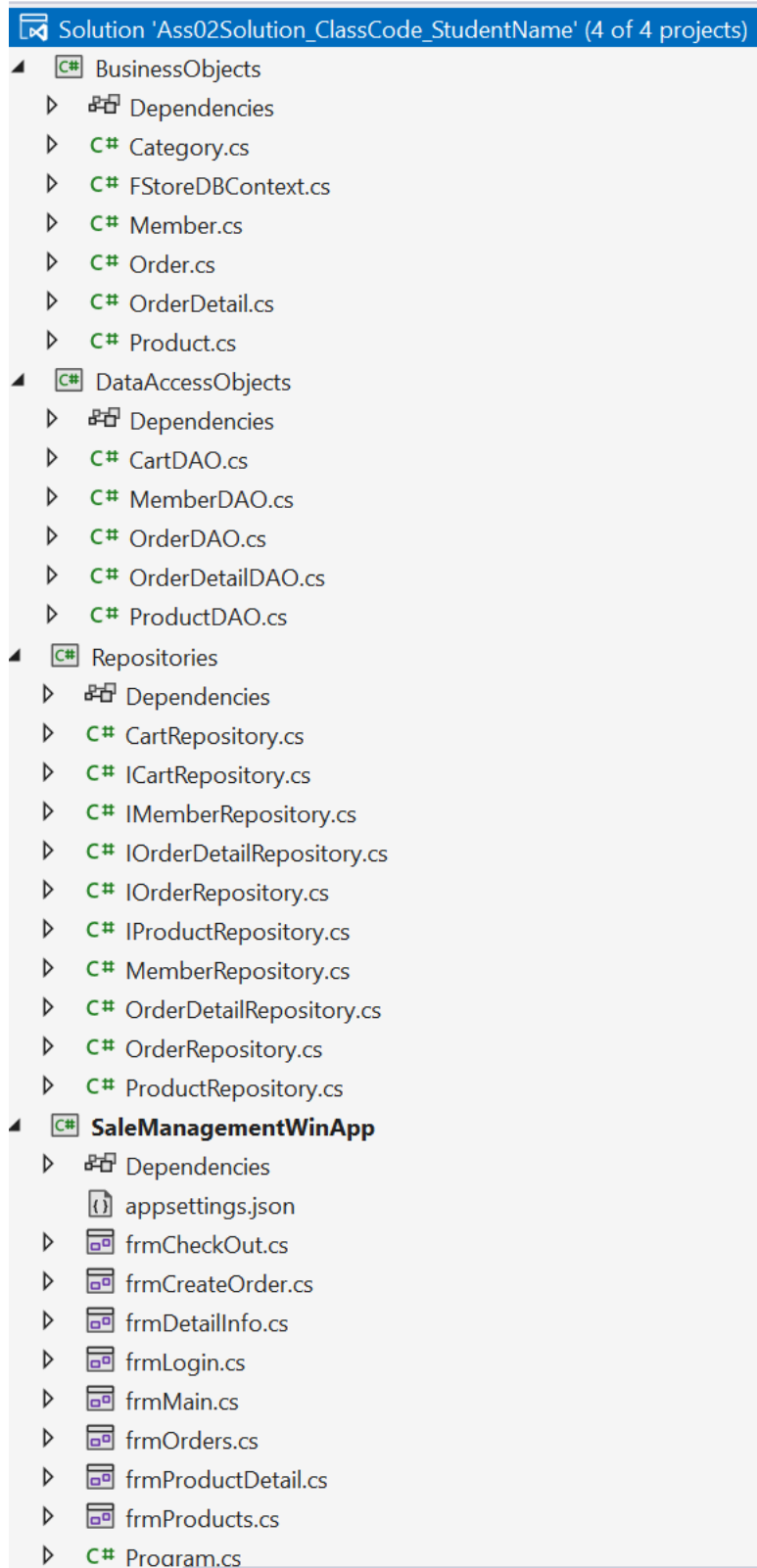
Step 02. Create a Class Library project named **BusinessObjects**

Step 03. Repeat **Step 02** to create a **DataAccessObjects** project.

Step 04. Repeat **Step 02** to create a **Repositories** project.

Step 05. Create a Windows Forms project named **SaleManagementWinApp**

Step 06. Create folders and add classes to the projects as follows:



Activity 02: Develop BusinessObjects project

Activity 03: Develop DataAccessObjects project

Activity 04: Develop Repositories project

Activity 05: Develop SaleManagementWinApp project

Step 01. Add the project reference to **Repositories** projects

Step 02. Design UI forms and write codes to perform functions

Hints: Use MenuStrip, ToolStrip, StatusStrip on the MDI form (frmMain form)

Activity 06: Run the WinForms project and test all actions