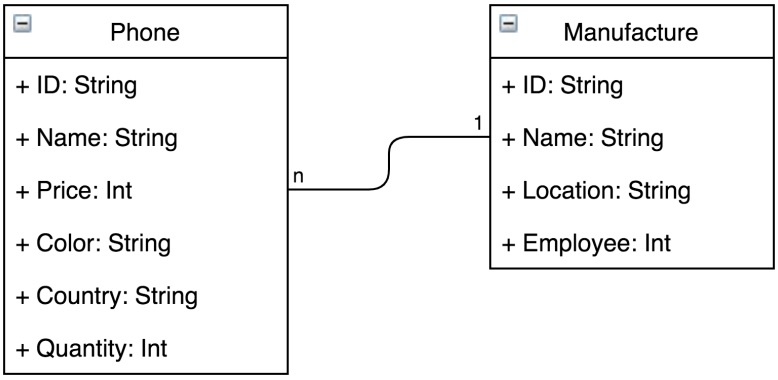
.NET Technology

(503112)

By Mai Van Manh

Lab 06: Entity Framework

Exercise 1. Utilize Windows **Console Application** and **Entity Framework Code First** to complete this assignment. Given two classes with descriptions as shown in the class diagram below.



Requirements:

1. Create POCO (Plain Old CLR Objects) classes based on the class diagram description.
    - Add a Reference Navigation properties in the Phone class and add a Collection Navigation properties in the Manufacture class to represent the 1-N relationship between Manufacture and Phone.
    - Specify the table names as **MobilePhone** and **Manufacture**.
    - In the MobilePhone table, ID is the primary key; Name, Price, and Color are mandatory attributes.
    - In the Manufacture table, ID is the primary key; Employee signifies the number of employees for the manufacturer.

- Phone and Manufacture names should be between 3 to 128 characters.

2. Create a **MarketingContext** class inheriting from the Entity Framework **DbContext** class. This class contains entity sets (of type DbSet<T>) used to perform read, create, delete, and update operations on Product and Category objects.
    - Specify that Entity Framework should create a database named 'OnlineMarketing'.
    - Automatically drop and recreate the database if the model changes.
3. Create a **PhoneDAO** class containing basic CRUD methods for Phone objects.
4. Create a **ManufactureDAO** class containing basic CRUD methods for Manufacture objects.
5. Add the following methods in the **PhoneDAO** class:
    - A method that returns the phone with the highest selling price.
    - A method that retrieves a list of phones sorted by country name, and if from the same country, sort in descending order of selling price.
    - A method that checks if there is any phone with a price above 50 million.
    - A methof that retrieves the first phone in the list that meets the criteria: has the color 'Pink' and a price above 15 million. If no phone meets the criteria, return null.
6. Add the following methods in the ManufactureDAO class:
    - A method that checks if all manufacturers have over 100 employees.
    - A method that returns the total number of employees across all factories.

- A method that returns the last manufacturer in the list that meets the criteria: has headquarters in the USA. If no manufacturer meets the criteria, throw an InvalidOperationException.

7. Validate all the functions written by calling them to create, add, delete, and update Phone and Manufacture objects.

Exercise 2. Do the same exercise as exercise 1 with other Entity Framework approaches, including: DB First and Model First.