**WPF AutoMobileLibary Lab**

Tên: Phạm Dương Minh Nhật

Mã Sinh Viên: 19IT182

**CarRepository.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using AutoMobileLibrary.DataAccess;

namespace AutoMobileLibrary.Repository

{

public class CarRepository : ICarRepository

{

public Car GetCarByID(int carId) => CarManagement.Instance.GetCarByID(carId);

public IEnumerable<Car> GetCars() => CarManagement.Instance.GetCarList();

public void InsertCar(Car car) => CarManagement.Instance.AddNew(car);

public void DeleteCar(Car car) => CarManagement.Instance.Remove(car);

public void UpdateCar(Car car) => CarManagement.Instance.Update(car);

}

}

**IcarRepository.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using AutoMobileLibrary.DataAccess;

namespace AutoMobileLibrary.Repository

{

public interface ICarRepository

{

IEnumerable<Car> GetCars();

Car GetCarByID(int carId);

void InsertCar(Car car);

void DeleteCar(Car car);

void UpdateCar(Car car);

}

}

**CarManagement.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

namespace AutoMobileLibrary.DataAccess

{

class CarManagement

{

private static CarManagement instance = null;

private static readonly object instanceLock = new object();

private CarManagement() { }

public static CarManagement Instance

{

get

{

lock (instanceLock)

{

if (instance == null)

{

instance = new CarManagement();

}

return instance;

}

}

}

public IEnumerable<Car> GetCarList()

{

List<Car> cars;

try

{

var myStockDB = new MyStockContext();

cars = myStockDB.Cars.ToList();

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

return cars;

}

public Car GetCarByID(int carID)

{

Car car = null;

try

{

var myStockDB = new MyStockContext();

car = myStockDB.Cars.SingleOrDefault(car => car.CarId == carID);

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

return car;

}

public void AddNew(Car car)

{

try

{

Car \_car = GetCarByID(car.CarId);

if (\_car == null)

{

var myStockDB = new MyStockContext();

myStockDB.Cars.Add(car);

myStockDB.SaveChanges();

}

else

{

throw new Exception("The car is already exist.");

}

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

public void Update(Car car)

{

try

{

Car c = GetCarByID(car.CarId);

if (c != null)

{

var myStockDB = new MyStockContext();

myStockDB.Entry<Car>(car).State = EntityState.Modified;

myStockDB.SaveChanges();

}

else

{

throw new Exception("The car does not already exist.");

}

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

public void Remove(Car car)

{

try

{

Car \_car = GetCarByID(car.CarId);

if (\_car != null)

{

var myStockDB = new MyStockContext();

myStockDB.Cars.Remove(car);

myStockDB.SaveChanges();

}

else

{

throw new Exception("The car does not already exist.");

}

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

}

}

}

**WindowCarManagement.xaml**

<Window x:Class="AutomobileWPF.WindowCarManagement"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:AutomobileWPF"

mc:Ignorable="d"

Title="Car Management" Height="800" Width="600"

WindowStartupLocation="CenterScreen" ResizeMode="NoResize">

<Grid Margin="0,0,10,10">

<Grid.Background>

<LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">

<GradientStop Color="Black"/>

<GradientStop Color="White" Offset="1"/>

</LinearGradientBrush>

</Grid.Background>

<Label Content="Car Information" HorizontalAlignment="Left" Height="40" Margin="10,56,0,0" VerticalAlignment="Top" Width="160" FontWeight="Bold" FontSize="20" Foreground="Red"/>

<Label Content="Car ID" HorizontalAlignment="Left" Margin="10,118,0,0" VerticalAlignment="Top" Foreground="White"/>

<Label Content="Car Name" HorizontalAlignment="Left" Margin="10,180,0,0" VerticalAlignment="Top" Foreground="White"/>

<Label Content="Manufacturer" HorizontalAlignment="Left" Margin="10,228,0,0" VerticalAlignment="Top" Foreground="White"/>

<Label Content="Price" HorizontalAlignment="Left" Margin="10,286,0,0" VerticalAlignment="Top" Foreground="White"/>

<Label Content="Released Year" HorizontalAlignment="Left" Margin="10,337,0,0" VerticalAlignment="Top" Foreground="White"/>

<TextBox x:Name="txtCarID" IsReadOnly="True" HorizontalAlignment="Left" Height="26" Margin="10,150,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="487" Text="{Binding Path=CarID, Mode=OneWay}" DataContext="{Binding ElementName=lsvCar, Path=SelectedItem}"/>

<TextBox x:Name="txtCarName" HorizontalAlignment="Left" Height="26" Margin="10,206,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="487" Text="{Binding Path=CarName, Mode=OneWay}" DataContext="{Binding ElementName=lsvCar, Path=SelectedItem}"/>

<TextBox x:Name="txtManufacturer" HorizontalAlignment="Left" Height="26" Margin="10,260,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="487" Text="{Binding Manufacturer, Mode=OneWay}" DataContext="{Binding SelectedItem, ElementName=lsvCar}"/>

<TextBox x:Name="txtPrice" HorizontalAlignment="Left" Height="26" Margin="10,311,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="487" Text="{Binding Price, Mode=OneWay}" DataContext="{Binding SelectedItem, ElementName=lsvCar}"/>

<TextBox x:Name="txtReleasedYear" HorizontalAlignment="Left" Height="26" Margin="10,364,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="487" Text="{Binding ReleasedYear, Mode=OneWay}" DataContext="{Binding SelectedItem, ElementName=lsvCar}"/>

<ListView x:Name="lsvCar" HorizontalAlignment="Left" Height="179" Margin="2,453,0,0" VerticalAlignment="Top" Width="502" d:ItemsSource="{d:SampleData ItemCount=5}">

<ListView.View>

<GridView>

<GridViewColumn Header="Car ID"

Width="100" DisplayMemberBinding="{Binding Path=CarID}"/>

<GridViewColumn Width="100" Header="Car Name" DisplayMemberBinding="{Binding Path=CarName}"/>

<GridViewColumn Width="100" Header="Manufacturer" DisplayMemberBinding="{Binding Path=Manufacturer}"/>

<GridViewColumn Width="100" Header="Price" DisplayMemberBinding="{Binding Path=Price}"/>

<GridViewColumn Width="100" Header="Released Year" DisplayMemberBinding="{Binding Path=ReleasedYear}"/>

</GridView>

</ListView.View>

</ListView>

<Button x:Name="btnLoad" Content="Load" HorizontalAlignment="Left" Height="32" Margin="10,404,0,0" VerticalAlignment="Top" Width="117" Click="btnLoad\_Click" Foreground="#FF03807E" FontWeight="Bold" FontSize="14"/>

<Button x:Name="btnInsert" Content="Insert" HorizontalAlignment="Left" Height="32" Margin="132,404,0,0" VerticalAlignment="Top" Width="117" Click="btnInsert\_Click" Foreground="#FF25D816" FontSize="14" FontWeight="Bold"/>

<Button x:Name="btnUpdate" Content="Update" HorizontalAlignment="Left" Height="32" Margin="257,404,0,0" VerticalAlignment="Top" Width="118" Click="btnUpdate\_Click" Foreground="#FF2496FF" FontSize="14" FontWeight="Bold"/>

<Button x:Name="btnDel" Content="Delete" HorizontalAlignment="Left" Height="32" Margin="380,404,0,0" VerticalAlignment="Top" Width="117" Click="btnDel\_Click" Foreground="Red" FontSize="14" FontWeight="Bold"/>

</Grid>

</Window>

**appsettings.json**

{

"exclude": [

"\*\*/bin",

"\*\*/bower\_components",

"\*\*/jspm\_packages",

"\*\*/node\_modules",

"\*\*/obj",

"\*\*/platforms"

],

"ConnectionStrings": {

"MyStockDB": "Server=SONNE;Trusted\_Connection=True;database=MyStock"

}

}

**App.xaml.cs**

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Threading.Tasks;

using System.Windows;

using Microsoft.Extensions.DependencyInjection;

using AutoMobileLibrary.Repository;

namespace AutomobileWPFApp

{

/// <summary>

/// Interaction logic for App.xaml

/// </summary>

public partial class App : Application

{

private ServiceProvider serviceProvider;

public App()

{

ServiceCollection services = new ServiceCollection();

ConfigureServices(services);

serviceProvider = services.BuildServiceProvider();

}

private void ConfigureServices(ServiceCollection services)

{

services.AddSingleton(typeof(ICarRepository), typeof(CarRepository));

services.AddSingleton<WindowCarManagement>();

}

private void OnStartup(object sender, StartupEventArgs e)

{

var windowCarManagement = serviceProvider.GetService<WindowCarManagement>();

windowCarManagement.Show();

}

}

}

**WindowCarManagement.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using AutoMobileLibrary.Repository;

using AutoMobileLibrary.DataAccess;

namespace AutomobileWPFApp

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class WindowCarManagement : Window

{

ICarRepository carRepository;

public WindowCarManagement(ICarRepository repository)

{

InitializeComponent();

carRepository = repository;

}

private Car GetCarObject()

{

Car car = null;

try

{

car = new Car

{

CarId = int.Parse(txtCarId.Text),

CarName = txtCarName.Text,

Manufacturer = txtManufacturer.Text,

Price = decimal.Parse(txtPrice.Text),

ReleasedYear = int.Parse(txtReleasedYear.Text)

};

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Get car");

}

return car;

}

public void LoadCarList()

{

lvCars.ItemsSource = carRepository.GetCars();

}

private void btnLoad\_Click(object sender, RoutedEventArgs e)

{

try

{

LoadCarList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Load car list");

}

}

private void btnInsert\_Click(object sender, RoutedEventArgs e)

{

try

{

Car car = GetCarObject();

carRepository.InsertCar(car);

LoadCarList();

MessageBox.Show($"{car.CarName} inserted successfully", "Insert car");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Insert car");

}

}

private void btnUpdate\_Click(object sender, RoutedEventArgs e)

{

try

{

Car car = GetCarObject();

carRepository.UpdateCar(car);

LoadCarList();

MessageBox.Show($"{car.CarName} updated successfully", "Update car");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Update car");

}

}

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

try

{

Car car = GetCarObject();

carRepository.DeleteCar(car);

LoadCarList();

MessageBox.Show($"{car.CarName} deleted successfully", "Delete car");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Delete car");

}

}

private void btnClose\_Click(object sender, RoutedEventArgs e)

{

Close();

}

}

}









