Họ và Tên: Phạm Dương Minh Nhật

Mã Sinh Viên: 19IT182

# WPF Lab

## MainWindow.xml

<Window x:Class="MyWDPApp.MainWindow"

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:MyWDPApp"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Frame x:Name="mainFrame"/>

<TextBox x:Name="txtWelcome" HorizontalAlignment="Left" Margin="329,194,0,0" TextWrapping="Wrap" Text="Welcome" VerticalAlignment="Top" Width="120" Background="#FFEFB0B0" FontSize="14" TextAlignment="Center"/>

<Button Margin="282,337,432,11" Content="To page 1" Click="btnToPage01\_Click">

<Button.Template>

<ControlTemplate TargetType="Button">

<Grid>

<Ellipse Stroke="Black"

StrokeThickness="2">

<Ellipse.Fill>

<RadialGradientBrush>

<GradientStop Offset="1"

Color="#FFFDA3A3" />

<RadialGradientBrush.Transform>

<TransformGroup>

<ScaleTransform ScaleY="0.65" />

</TransformGroup>

</RadialGradientBrush.Transform>

</RadialGradientBrush>

</Ellipse.Fill>

</Ellipse>

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</Grid>

</ControlTemplate>

</Button.Template>

</Button>

<Button Content="To page 2" Name="btnToPage02" Click="btnToPage02\_Click"

HorizontalAlignment="Left" Margin="400,337,0,10" Width="83" Background="#FF9E4D4D">

<Button.Template>

<ControlTemplate TargetType="Button">

<Grid>

<Ellipse Stroke="Black"

StrokeThickness="2">

<Ellipse.Fill>

<RadialGradientBrush>

<GradientStop Offset="1"

Color="#FF6DFD69" />

<RadialGradientBrush.Transform>

<TransformGroup>

<ScaleTransform ScaleY="0.65" />

</TransformGroup>

</RadialGradientBrush.Transform>

</RadialGradientBrush>

</Ellipse.Fill>

</Ellipse>

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</Grid>

</ControlTemplate>

</Button.Template>

</Button>

<Button Margin="670,333,44,15" Content="Demo Canvas" x:Name="DemoCanvas" Click="DemoCanvas\_Click">

<Button.Template>

<ControlTemplate TargetType="{x:Type Button}">

<Grid>

<Ellipse Stroke="Black"

StrokeThickness="2">

<Ellipse.Fill>

<RadialGradientBrush>

<RadialGradientBrush.Transform>

<TransformGroup>

<ScaleTransform ScaleY="0.65" />

</TransformGroup>

</RadialGradientBrush.Transform>

<GradientStop Offset="1"

Color="#FFF9F583" />

</RadialGradientBrush>

</Ellipse.Fill>

</Ellipse>

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</Grid>

</ControlTemplate>

</Button.Template>

</Button>

</Grid>

</Window>

## Mainwindow.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;

namespace MyWDPApp

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void btnToPage01\_Click(object sender, RoutedEventArgs e)

{

mainFrame.Content = new Page1();

}

private void btnToPage02\_Click(object sender, RoutedEventArgs e)

{

mainFrame.Content = new Page2();

}

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

{

Canvas canvas = new Canvas();

canvas.Show();

this.Close();

}

}

}

## Page1.xaml

<Page x:Class="MyWDPApp.Page1"

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

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

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

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

xmlns:local="clr-namespace:MyWDPApp"

mc:Ignorable="d"

d:DesignHeight="450" Width="800"

Title="Page1">

<Grid Background="LightGreen">

<TextBlock TextAlignment="Center" Width="362"

FontSize="24" FontWeight="Bold" Foreground="Red"

Text="Welcome to Window Page 1"

Margin="69,130,69,151"/>

</Grid>

</Page>

## Page2.xaml

<Page x:Class="MyWDPApp.Page2"

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

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

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

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

xmlns:local="clr-namespace:MyWDPApp"

mc:Ignorable="d"

d:DesignHeight="450" Width="800"

Title="Page2">

<Grid Background="PaleTurquoise">

<TextBlock TextAlignment="Center" Width="362"

FontSize="24" FontWeight="Bold"

Text="Welcome to Window Page 2" Foreground="ForestGreen"

Margin="69,130,69,151"/>

</Grid>

</Page>

## Product

A picture containing graphical user interface

Description automatically generated

A picture containing graphical user interface

Description automatically generated

## Canvas.xaml

<Window x:Class="MyWDPApp.Canvas"

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:MyWDPApp"

mc:Ignorable="d"

Title="Canvas" Height="378" Width="408">

<Grid>

<Canvas Background="LightBlue">

<Button x:Name="btnDisplay" Canvas.Left="94" Height="28" Canvas.Top="203"

Width="80" Content="Display" Click="btnDisplay\_Click" />

<Label x:Name="lblInstructions" Canvas.Left="17" Canvas.Top="14"

Width="328" Height="27" FontSize="15" Content="Enter Car Information"/>

<Label x:Name="lblCarName" Canvas.Left="17" Canvas.Top="60" Content="Car Name"/>

<TextBox x:Name="txtCarName" Canvas.Left="94" Canvas.Top="60" Width="193" Height="25"/>

<Label x:Name="lblColor" Canvas.Left="17" Canvas.Top="109" Content="Color"/>

<TextBox x:Name="txtColor" Canvas.Left="94" Canvas.Top="107" Width="193" Height="25"/>

<Label x:Name="lblBrand" Canvas.Left="17" Canvas.Top="155" Content="Brand"/>

<TextBox x:Name="txtBrand" Canvas.Left="94" Canvas.Top="153" Width="193" Height="25"/>

</Canvas>

</Grid>

</Window>

## Canvas.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.Shapes;

namespace MyWDPApp

{

/// <summary>

/// Interaction logic for Canvas.xaml

/// </summary>

public partial class Canvas : Window

{

public Canvas()

{

InitializeComponent();

}

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

{

string CarInfo = $"Car Name: {txtCarName.Text} \n" + $"Color: {txtColor.Text}\n Brand: {txtBrand.Text}";

MessageBox.Show(CarInfo, "Car details");

WrapPanel wrapPanel = new WrapPanel();

wrapPanel.Show();

}

}

}

## Product

Graphical user interface

Description automatically generated

## WrapPanel.xaml

<Window x:Class="MyWDPApp.WrapPanel"

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:MyWDPApp"

mc:Ignorable="d"

Title="WrapPanel" Height="323" Width="340">

<Grid>

<WrapPanel Background="LightBlue" Orientation ="Vertical">

<Label Name="lblInstruction" Width="328" Height="27"

FontSize="15" Content="Enter Car Information"/>

<Label Name="lblCarName" Content="Car Name"/>

<TextBox Name="txtCarName" Width="193" Height="25"/>

<Label Name="lblColor" Content="Color"/>

<TextBox Name="txtColor" Width="193" Height="25"/>

<Label Name="lblBrand" Content="Brand"/>

<TextBox Name="txtBrand" Width="193" Height="25"/>

<Button Name="btnDisplay" Width="80"

Margin="0,10,0,0" Content="Display" Click="btnDisplay\_Click"/>

</WrapPanel>

</Grid>

</Window>

Graphical user interface

Description automatically generated

## StackPanel.xaml

<Window x:Class="MyWDPApp.StackPanel"

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:MyWDPApp"

mc:Ignorable="d"

Title="StackPanel" Height="450" Width="800">

<Grid>

<StackPanel Background="LightBlue" Orientation ="Vertical">

<Label Name="lblInstruction"

FontSize="15" Content="Enter Car Information"/>

<Label Name="lblCarName" Content="Car Name"/>

<TextBox Name="txtCarName" Height="25"/>

<Label Name="lblColor" Content="Color"/>

<TextBox Name="txtColor" Height="25"/>

<Label Name="lblBrand" Content="Brand"/>

<TextBox Name="txtBrand" Height="25"/>

<Button Name="btnDisplay" Width="80"

Margin="0,10,0,0" Content="Display" Click="btnDisplay\_Click"/>

</StackPanel>

</Grid>

</Window>

Chart

Description automatically generated

## GridPanel.xaml

<Window x:Class="MyWDPApp.GridPanel"

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:MyWDPApp"

mc:Ignorable="d"

Title="GridPanel" Height="450" Width="800">

<Grid Background="LightBlue">

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="\*" />

<RowDefinition Height="28" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="300" />

</Grid.ColumnDefinitions>

<Label Grid.Row="0" Grid.Column="0" Content="Name:" />

<Label Grid.Row="1" Grid.Column="0" Content="Email:" />

<Label Grid.Row="2" Grid.Column="0" Content="Comment:" />

<TextBox Grid.Column="1" Grid.Row="0" Margin="3" />

<TextBox Grid.Column="1" Grid.Row="1" Margin="3" />

<TextBox Grid.Column="1" Grid.Row="2" Margin="3" />

<Button Grid.Column="1" Grid.Row="3"

HorizontalAlignment="Center"

MinWidth="80" Margin="3" Content="Send" />

</Grid>

</Window>

## DockPanel.xaml

<Window x:Class="MyWDPApp.DockPanel"

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:MyWDPApp"

mc:Ignorable="d"

Title="DockPanel" Height="450" Width="800">

<Grid>

<DockPanel Background="LightBlue">

<Button DockPanel.Dock="Top" Height="50">Top</Button>

<Button DockPanel.Dock="Bottom" Height="50">Bottom</Button>

<Button DockPanel.Dock="Left" Width="50">Left</Button>

<Button DockPanel.Dock="Right" Width="50">Right</Button>

<Button Click="Button\_Click">Center</Button>

</DockPanel>

</Grid>

</Window>

Graphical user interface

Description automatically generated

## ListBox.xaml

<Window x:Class="MyWDPApp.ListBox"

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:MyWDPApp"

mc:Ignorable="d"

Title="ListBox" Height="450" Width="800">

<Grid Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

<RowDefinition Height="auto"></RowDefinition>

</Grid.RowDefinitions>

<ListBox x:Name="lsb" Margin="0,0,0,179">

<CheckBox Margin="3" Content="POP"/>

<CheckBox Margin="3" Content="Classic"/>

<CheckBox Margin="3" Content="EDM"/>

<CheckBox Margin="3" Content="Trap House"/>

</ListBox>

<StackPanel Grid.Row="1" Margin="0, 10, 0, 0">

<TextBlock TextWrapping="Wrap" Text="TextBlock" Height="60"><Run Text="Your Music:"/></TextBlock>

<TextBlock TextWrapping="Wrap" Text="TextBlock" x:Name="txtSelection"/>

<Button Margin="0, 10, 0 ,0" Click="Button\_Click">View</Button>

</StackPanel>

</Grid>

</Window>

Graphical user interface, application, Word

Description automatically generated

## ManageCategoriesApp.xaml

<Window x:Class="MyWDPApp.ManageCategoriesApp"

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:MyWDPApp"

mc:Ignorable="d"

Title="ManageCategoriesApp" Height="450" Width="419" Loaded="Window\_Loaded" WindowStartupLocation="CenterScreen" >

<DockPanel VerticalAlignment="Top" Margin="10">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<StackPanel Background="LightBlue" Orientation="Vertical"

HorizontalAlignment="Left" Width="400">

<Label Name="lblInstruction" Foreground="Red" FontWeight="DemiBold"

FontSize="20" Content="Category Information"/>

<Label Name="lblCategoryID" Content="CategoryID"/>

<TextBox Name="txtCategoryID" HorizontalAlignment="Left"

IsReadOnly="True" Height="25" Width="300" Text="{Binding Path=CategoryID, Mode=OneWay}" DataContext="{Binding ElementName=lvCategories, Path=SelectedItem}" />

<Label Name="lbCategoryName" Content="Category Name" />

<TextBox Name="txtCategoryName" HorizontalAlignment="Left"

Height="25" Width="300" Text="{Binding Path=CategoryName, Mode=OneWay}" DataContext="{Binding ElementName=lvCategories, Path=SelectedItem}" />

<StackPanel Orientation="Horizontal" HorizontalAlignment="Left">

<Button x:Name="btnInsert" Margin="10" Width="80" Content="Insert"

Click="btnInsert\_Click" />

<Button x:Name="btnUpdate" Margin="10" Width="80" Content="Update"

Click="btnUpdate\_Click" />

<Button x:Name="btnDelete" Margin="10" Width="80" Content="Delete"

Click="btnDelete\_Click" />

</StackPanel>

</StackPanel>

<ListView Grid.Row="1" Name="lvCategories" Width="400" >

<ListView.View>

<GridView>

<GridViewColumn Header="Category ID" Width="100"

DisplayMemberBinding="{Binding Path=CategoryID }"/>

<GridViewColumn Header="Category Name" Width="200"

DisplayMemberBinding="{Binding Path=CategoryName}"/>

</GridView>

</ListView.View>

</ListView>

</Grid>

</DockPanel>

</Window>

## ManageCategoriesApp.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.Shapes;

namespace MyWDPApp

{

/// <summary>

/// Interaction logic for ManageCategoriesApp.xaml

/// </summary>

public partial class ManageCategoriesApp : Window

{

public ManageCategoriesApp()

{

InitializeComponent();

}

ManageCategories categories = new ManageCategories();

private void Window\_Loaded(object sender, RoutedEventArgs e) => LoadCategories();

private void LoadCategories()

{

lvCategories.ItemsSource = categories.GetCategories();

}

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

{

try

{

var category = new Category { CategoryName = txtCategoryName.Text };

categories.InsertCategory(category);

LoadCategories();

}

catch (Exception ex)

{

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

}

}

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

{

try

{

var category = new Category

{

CategoryID = int.Parse(txtCategoryID.Text),

CategoryName = txtCategoryName.Text

};

categories.UpdateCategory(category);

LoadCategories();

}

catch (Exception ex)

{

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

}

}

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

{

try

{

var category = new Category

{

CategoryID = int.Parse(txtCategoryID.Text)

};

categories.DeleteCategory(category);

LoadCategories();

}

catch (Exception ex)

{

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

}

}

}

}

## ManageCategories.cs

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace MyWDPApp

{

public record Category

{

public int CategoryID { get; set; }

public string CategoryName { get; set; }

}

public class ManageCategories

{

SqlConnection connection;

SqlCommand command;

string ConnectionString = "Server=SONNE;Integrated Security=true;database=MyStore;Encrypt=false;TrustServerCertificate=True\r\n";

public List<Category> GetCategories()

{

List<Category> categories = new List<Category>();

connection = new SqlConnection(ConnectionString);

string SQL = "Select CategoryID, CategoryName from Categories";

command = new SqlCommand(SQL, connection);

try

{

connection.Open();

SqlDataReader reader = command.ExecuteReader(CommandBehavior.CloseConnection);

if (reader.HasRows == true)

{

while (reader.Read())

{

categories.Add(new Category

{

CategoryID = reader.GetInt32("CategoryID"),

CategoryName = reader.GetString("CategoryName")

});

}

}

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

finally

{

connection.Close();

}

return categories;

}

public void InsertCategory(Category category)

{

connection = new SqlConnection(ConnectionString);

command = new SqlCommand("Insert Categories values(@CategoryName)", connection);

try

{

connection.Open();

command.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

finally

{

connection.Close();

}

}

public void UpdateCategory(Category category)

{

connection = new SqlConnection(ConnectionString);

string SQL = "Update Categories set CategoryName=@CategoryName where CategoryID=@CategoryID";

command = new SqlCommand(SQL, connection);

command.Parameters.AddWithValue("@CategoryID", category.CategoryID);

command.Parameters.AddWithValue("@CategoryName", category.CategoryName);

try

{

connection.Open();

command.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

finally { connection.Close(); }

}

public void DeleteCategory(Category category)

{

connection = new SqlConnection(ConnectionString);

string SQL = "Delete Categories where CategoryID=@CategoryID";

command = new SqlCommand(SQL, connection);

command.Parameters.AddWithValue("@CategoryID", category.CategoryID);

try

{

connection.Open();

command.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception(ex.Message);

}

finally { connection.Close(); }

}

}

}

Graphical user interface, application

Description automatically generated