<Window ...

Title="MainWindow" Height="350" Width="525"

DataContext="{Binding MainViewModel, Source={StaticResource Locator}}">

<Grid>

<ListBox x:Name="listBox" HorizontalAlignment="Left" Height="243" Margin="10,54,0,0" VerticalAlignment="Top" Width="354" ItemsSource="{Binding MemberList}" DisplayMemberPath="GetDisplayText" SelectedItem="{Binding SelectedMember}"/>

<Label x:Name="customerLabel" Content="Member:" HorizontalAlignment="Left" Margin="10,28,0,0" VerticalAlignment="Top"/>

<Button x:Name="addButton" Content="Add" HorizontalAlignment="Left" Margin="402,54,0,0" VerticalAlignment="Top" Width="76" Command="{Binding AddMemberCommand}"/>

<Button x:Name="delButton" Content="Delete" HorizontalAlignment="Left" Margin="403,130,0,0" VerticalAlignment="Top" Width="75" Command="{Binding DelMemberCommand}"/>

<Button x:Name="exitButton" Content="Exit" HorizontalAlignment="Left" Margin="403,168,0,0" VerticalAlignment="Top" Width="75" Command="{Binding ExitCommand}"/>

<Button x:Name="Update" Content="Update" HorizontalAlignment="Left" Margin="403,91,0,0" VerticalAlignment="Top" Width="75" Command="{Binding UpdateCommand}"/>

</Grid>

</Window>

using MembershipMaintenanceMVVM.Message;

namespace MembershipMaintenanceMVVM.ViewModel

{

public class MainViewModel : ViewModelBase

{

/// <summary>

/// properties

/// </summary>

private ObservableCollection<Member> members;

private Member selectedMember;

public ICommand AddMemberCommand { get; private set; }

public ICommand DelMemberCommand { get; private set; }

public ICommand UpdateCommand { get; private set; }

public ICommand ExitCommand { get; private set; }

public ObservableCollection<Member> MemberList

{

get

{

return members;

}

}

public Member SelectedMember

{

get

{

return selectedMember;

}

set

{

selectedMember = value;

RaisePropertyChanged("selectedMember");

}

}

/// <summary>

/// Initializes a new instance of the MainViewModel class.

/// </summary>

public MainViewModel()

{

members = new ObservableCollection<Member>();

AddMemberCommand = new RelayCommand(AddMember);

DelMemberCommand = new RelayCommand(DelMember);

UpdateCommand = new RelayCommand(UpdateMember);

ExitCommand = new RelayCommand(ExitAction);

LoadMembers();

}

/// <summary>

/// loads the list box

/// </summary>

public void LoadMembers()

{

this.RaisePropertyChanged(() => this.MemberList);

Messenger.Default.Register<Member>(this, OnReceiveMessageAction);

}

/// <summary>

/// updates selected member

/// </summary>

public void UpdateMember()

{

upMemWindow = new View.UpdateMemberWindow();

UpdateMemberViewModel View = new UpdateMemberViewModel(new System.Action(() => upMemWindow.Close()));

View.FNameText = SelectedMember.firstName;

View.LNameText = SelectedMember.lastName;

View.emailText = SelectedMember.email;

upMemWindow.DataContext = View;

upMemWindow.Show();

Messenger.Default.Register<Messages>(this, OnReceiveMessageUAction);

}

/// <summary>

/// replaces the selected member with new data

/// </summary>

/// <param name="obj"></param>

private void OnReceiveMessageUAction(Messages obj)

{

Member m = new Member(obj.firstName, obj.lastName, obj.email);

int i = MemberList.IndexOf(SelectedMember);

MemberList[i] = m;

this.RaisePropertyChanged(() => this.MemberList);

Messenger.Default.Unregister<Messages>(this, OnReceiveMessageUAction);

}

.

.

.

}

<Window ...

Title="UpdateMemberWindow" Height="219.394" Width="300"

DataContext="{Binding UpdateMemberViewModel, Source={StaticResource Locator}}">

<Grid>

<Label x:Name="fNameText" Content="First Name:" HorizontalAlignment="Left" Margin="23,31,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="fNameBox" HorizontalAlignment="Left" Height="23" Margin="109,31,0,0" TextWrapping="Wrap" Text="{Binding FNameText}" VerticalAlignment="Top" Width="147"/>

...

</Grid>

</Window>

using MembershipMaintenanceMVVM.Message;

namespace MembershipMaintenanceMVVM.ViewModel

{

public class UpdateMemberViewModel : ViewModelBase

{

/// <summary>

/// properties to bind to the view

/// </summary>

string fn, ln, em;

public string FNameText

{

get

{ return fn; }

set

{

fn = value;

RaisePropertyChanged("FNameText");

}

}

public string LNameText

{

get

{ return ln; }

set

{

ln = value;

RaisePropertyChanged("LNameText");

}

}

public string emailText

{

get

{ return em; }

set

{

em = value;

RaisePropertyChanged("emailText");

}

}

public Action Exit { get; set; }

public ICommand UpdateCommand { get; set; }

public ICommand CancelCommand { get; set; }

/// <summary>

/// constructor of the view model

/// </summary>

/// <param name="exit"></param>

public UpdateMemberViewModel(Action exit)

{

Exit = exit;

UpdateCommand = new RelayCommand(UpdateCommandAction);

CancelCommand = new RelayCommand(exit);

}

/// <summary>

/// sends data to the mainview model

/// </summary>

/// <param name="exit"></param>

public void UpdateCommandAction()

{

var messagesMem = new Messages();

messagesMem.firstName = FNameText;

messagesMem.lastName = LNameText;

messagesMem.email = emailText;

Messenger.Default.Send(messagesMem);

Exit();

}

}

}

namespace MembershipMaintenanceMVVM.Message

{

/// <summary>

/// to transfer data between views

/// </summary>

class Messages

{

public string firstName

{

get; set;

}

public string lastName

{

get; set;

}

public string email

{

get; set;

}

}

}