**CECS - 475 SoftwareDevelopement and Frameworks**

**Lab - 4 Part 3 gym Fitness membership**

Submitted by : - Kamakshi Chaudhry

Process of transferring data from the Update View to the Main View

* **The ViewModelLocator must register and create instances of the UpdateViewModel and the MainViewModel. The registration is for finding the viewmodel during data context binding.**

public ViewModelLocator()

{

ServiceLocator.SetLocatorProvider(() => SimpleIoc.Default);

SimpleIoc.Default.Register<MainViewModel>();

SimpleIoc.Default.Register<AddViewModel>();

SimpleIoc.Default.Register<ChangeViewModel>();

}

* **The Data context must be set in a high-level container (in this case the window itself) to find out where we want to bind our data.**
* **For ChangeViewModel:**

DataContext="{Binding Source={StaticResource Locator}, Path=ChangeView}"

* **For MainViewModel:**

DataContext="{Binding Source={StaticResource Locator}, Path=Main}"

* **For AddViewModel**:

DataContext="{Binding Source={StaticResource Locator}, Path=AddView}"

* **The Update button must be bound to the ChangeViewModel ChangeCommand relay command.**

<Button x:Name="button" Content="Update" HorizontalAlignment="Center" VerticalAlignment="Top" Command="{Binding UpdateCommand}" CommandParameter="{Binding ElementName=changeWindow}" Width="75" Grid.Row="4" Grid.Column="0"/>

* **When the user clicks the update button, the UpdateCommand relay command is called in the ChangeViewModel .**
* **The relay command calls the UpdateCommandAction method to execute the command.**

UpdateCommand = new RelayCommand<IClosable>(UpdateMethod);

* **If the data is valid, the UpdateCommand will send a message to ManViewModel**

public void UpdateMethod(Window window)

{

try

{

Member to = new Member(EnteredFName, EnteredLName, EnteredEmail);

MessageMember mem = new MessageMember(EnteredFName, EnteredLName, EnteredEmail, "Update");

Messenger.Default.Send(mem);

window.Close();

}

* **The MainViewModel must set up a receiver for ChangeViewModel messages**

Messenger.Default.Register<MessageMember>(this, ReceiveMember);

* **The MainViewModel receives the message and calls ReceiveMember to execute. The dataContext variables are used to update a new object into the member list.**

public void ReceiveMember(MessageMember m)

{

if (m.Message == "Update")

{

var i = members.IndexOf(SelectedMember);

members[i] = new Member(m.FirtName, m.LastName, m.Email);

database.SaveMemberships();

}

else if (m.Message == "Add")

{

members.Add(new Member(m.FirtName, m.LastName, m.Email));

database.SaveMemberships();

}

}

* **The newest member in the list will now show in the Main View.**

Process of transferring data from the Main View to Update View

* **The constructor of MainViewModel has all the definations used to connect to AddViewModel and ChangeViewModel**

public MainViewModel()

{

selectedMember = new Member();

members = new ObservableCollection<Member>();

database = new MemberDB(members);

members = database.GetMemberships();

AddCommand = new RelayCommand(AddMethod);

ExitCommand = new RelayCommand<Window>(ExitMethod);

ChangeCommand = new RelayCommand(ChangeMethod);

Messenger.Default.Register<MessageMember>(this, ReceiveMember);

Messenger.Default.Register<NotificationMessage>(this, ReceiveMessage);

}

* **It calls the the ICommand ChangeCommand when the update button is clicked which calls the ChangeMethod.**

public void ChangeMethod()

{

if (SelectedMember != null)

{

ChangeWindow change = new ChangeWindow();

change.Show();

Messenger.Default.Send(SelectedMember);

}

}

* **This sends the SelectedMember details from the list box to the ChangeViewModel window.**