

```

1  <Window x:Class="Assignment3Mvvm.Views.AddMemberView"
2      x:Name="AddMemberWindow"
3      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
7      xmlns:local="clr-namespace:Assignment3Mvvm.Views"
8      mc:Ignorable="d"
9      Title="Add Member" Height="400" Width="500" ResizeMode="NoResize">
10  <Grid DataContext="{Binding AddViewModel, Source={StaticResource
    ViewModelLocator}}">
11      <Grid.ColumnDefinitions>
12          <ColumnDefinition Width="150" />
13          <ColumnDefinition Width="350"/>
14      </Grid.ColumnDefinitions>
15      <Grid.RowDefinitions>
16          <RowDefinition Height="80"/>
17          <RowDefinition Height="80"/>
18          <RowDefinition Height="80"/>
19          <RowDefinition Height="80"/>
20      </Grid.RowDefinitions>
21      <TextBlock Grid.Row="0" Grid.Column="0" Text="First Name:"
    VerticalAlignment="Center" HorizontalAlignment="Center"
    FontWeight="Bold"/>
22      <TextBox Grid.Row="0" Grid.Column="1" Height="30" Width="300"
    HorizontalAlignment="Right" Margin="0,0,30,0" BorderThickness="1"
    BorderBrush="Black" Text="{Binding FirstNameTextBox}" />
23      <TextBlock Grid.Row="1" Grid.Column="0" Text="Last Name:"
    VerticalAlignment="Center" HorizontalAlignment="Center"
    FontWeight="Bold"/>
24      <TextBox Grid.Row="1" Grid.Column="1" Height="30" Width="300"
    HorizontalAlignment="Right" Margin="0,0,30,0" BorderBrush="Black"
    BorderThickness="1" Text="{Binding LastNameTextBox}" />
25      <TextBlock Grid.Row="2" Grid.Column="0" Text="Email:"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontWeight="Bold" />
26      <TextBox Grid.Row="2" Grid.Column="1" Height="30" Width="300"
    HorizontalAlignment="Right" Margin="0,0,30,0" BorderBrush="Black"
    BorderThickness="1" Text="{Binding EmailTextBox}"/>
27      <Button Grid.ColumnSpan="2" Content="Save" HorizontalAlignment="Left"
    Margin="106,34.6,0,0" Grid.Row="3" VerticalAlignment="Top" Width="75"
    Command="{Binding AddCommand, Mode=OneWay}" CommandParameter="{Binding
    ElementName=AddMemberWindow}"/>
28      <Button Content="Cancel" Grid.Column="1" HorizontalAlignment="Left"
    Margin="165,34.6,0,0" Grid.Row="3" VerticalAlignment="Top" Width="75"
    Command="{Binding CancelCommand}" CommandParameter="{Binding
    ElementName=AddMemberWindow}"/>
29  </Grid>
30 </Window>

```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Windows;
7 using System.Windows.Input;
8 using Assignment3Mvvm.Views;
9 using Assignment3Mvvm.Messages;
10 using GalaSoft.MvvmLight.Messaging;
11 using GalaSoft.MvvmLight;
12 using GalaSoft.MvvmLight.Command;
13 using Assignment3Mvvm.Model;
14
15 namespace Assignment3Mvvm.ViewModel
16 {
17     public class AddViewModel : ViewModelBase
18     {
19         public ICommand AddCommand { get; private set; }
20         public ICommand CancelCommand { get; private set; }
21
22         private string firstNameText;
23         private string lastNameText;
24         private string emailText;
25
26         public AddViewModel()
27         {
28             AddCommand = new RelayCommand<IClosable>(this.OnClickAddMember);
29             CancelCommand = new RelayCommand<IClosable>
30                 (this.OnClickCloseWindow, null);
31         }
32
33         public string FirstNameTextBox
34         {
35             get { return firstNameText; }
36             set
37             {
38                 firstNameText = value;
39                 RaisePropertyChanged("FirstNameTextBox");
40             }
41         }
42
43         public string LastNameTextBox
44         {
45             get { return lastNameText; }
46             set
47             {
48                 lastNameText = value;
```

```
49         RaisePropertyChanged("LastNameTextBox");
50     }
51 }
52
53 public string EmailTextBox
54 {
55     get { return emailText; }
56     set
57     {
58         emailText = value;
59         RaisePropertyChanged("EmailTextBox");
60     }
61 }
62
63 private void OnClickAddMember(IClosable window)
64 {
65
66     if (Validator.IsPresent("First name", firstNameText)
67         && Validator.IsPresent("Last name", lastNameText)
68         && Validator.IsPresent("Email", emailText)
69         && Validator.IsValidEmail(emailText)
70         && Validator.IsWithinRange(firstNameText, 1, 25)
71         && Validator.IsWithinRange(lastNameText, 1, 25)
72         && Validator.IsWithinRange(emailText, 1, 25))
73     {
74         var memberMessage = new ViewModelMessage()
75         {
76             WindowCommandText = "Add",
77             FirstNameText = FirstNameTextBox,
78             LastNameText = LastNameTextBox,
79             EmailText = EmailTextBox
80         };
81         string notificationText = String.Format("Member Updated: {0} {1} - {2}",
82             firstNameText, lastNameText, emailText);
83         Messenger.Default.Send<NotificationMessage>(new NotificationMessage(notificationText));
84         Messenger.Default.Send(memberMessage);
85     }
86     if (window != null)
87     {
88         window.Close();
89     }
90 }
91
92
93 private void OnClickCloseWindow(IClosable window)
94 {
95     var closeWindowMessage = new ViewModelMessage()
```

```
96         {
97             WindowCommandText = "Cancel",
98             FirstNameText = null,
99             LastNameText = null,
100             EmailText = null
101         };
102         Messenger.Default.Send(closeWindowMessage);
103         window.Close();
104     }
105 }
106 }
107
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Assignment3Mvvm.ViewModel
8 {
9     public interface IClosable
10     {
11         void Close();
12     }
13 }
14
```

```
1 using GalaSoft.MvvmLight;
2 using GalaSoft.MvvmLight.Ioc;
3 using GalaSoft.MvvmLight.Messaging;
4 using GalaSoft.MvvmLight.Command;
5 using System.Collections.ObjectModel;
6 using GalaSoft.MvvmLight.Views;
7 using Assignment3Mvvm.Model;
8 using System.Windows.Input;
9 using Assignment3Mvvm.Views;
10 using Assignment3Mvvm.Messages;
11 using System.Windows;
12
13 namespace Assignment3Mvvm.ViewModel
14 {
15     /// <summary>
16     /// This class contains properties that the main View can data bind to.
17     /// <para>
18     /// Use the <strong>mvvmminpc</strong> snippet to add bindable properties to ↗
19     /// this ViewModel.
20     /// </para>
21     /// <para>
22     /// You can also use Blend to data bind with the tool's support.
23     /// </para>
24     /// <para>
25     /// See http://www.galasoft.ch/mvvm
26     /// </para>
27     /// </summary>
28     public class MainViewModel : ViewModelBase
29     {
30         /// <summary>
31         /// Initializes a new instance of the MainViewModel class.
32         /// </summary>
33         private MembershipList members;
34         private Member selectedMember;
35         public ICommand ShowAddCommand { get; private set; }
36         public ICommand DeleteMemberCommand { get; private set; }
37         public ICommand ShowUpdateCommand { get; private set; }
38         public ICommand ExitAppCommand { get; private set; }
39         public MainViewModel()
40         {
41             ///if (IsInDesignMode)
42             ///{
43             ///    // Code runs in Blend --> create design time data.
44             ///}
45             ///else
46             ///{
47             ///    // Code runs "for real"
48             ///}
49             members = new MembershipList();
```

```
49         members.Write();
50         ShowAddCommand = new RelayCommand(ShowAddMethod);
51         DeleteMemberCommand = new RelayCommand(DeleteMemberMethod);
52         ExitAppCommand = new RelayCommand(ExitAppMethod);
53         ShowUpdateCommand = new RelayCommand(ShowUpdateMethod);
54         Messenger.Default.Register<ViewModelMessage>(this,           ↗
            OnReceiveMessageAction);
55     }
56
57     public ObservableCollection<Member> MemberList
58     {
59         get
60         {
61             return members.Members;
62         }
63     }
64
65     public Member SelectedMember
66     {
67         get { return selectedMember; }
68         set
69         {
70             selectedMember = value;
71             //System.Console.WriteLine(selectedMember.GetDisplayText);
72             RaisePropertyChanged("SelectedMember");
73         }
74     }
75
76     public void ShowAddMethod()
77     {
78         var addMemberView = new AddMemberView();
79         addMemberView.Show();
80     }
81
82     public void DeleteMemberMethod()
83     {
84         members.Remove(selectedMember);
85         members.Save();
86         this.RaisePropertyChanged(() => this.MemberList);
87     }
88
89
90     private void OnReceiveMessageAction(ViewModelMessage m)
91     {
92         if (m.WindowCommandText != null && m.WindowCommandText.Equals           ↗
            ("Cancel"))
93         {
94             return;
95         }
```

```
196         else if (m.WindowCommandText != null && m.WindowCommandText.Equals "Update") ↗
197         {
198             int selectedIndex = members.Members.IndexOf(selectedMember);
199             SelectedMember.FirstName = m.FirstNameText;
200             SelectedMember.LastName = m.LastNameText;
201             SelectedMember.Email = m.EmailText;
202             members[selectedIndex] = new Member(m.FirstNameText, ↗
203                 m.LastNameText, m.EmailText);
204             this.RaisePropertyChanged(() => this.MemberList);
205             members.Save();
206         }
207     else
208     {
209         members.Add(new Member(m.FirstNameText, m.LastNameText, ↗
210             m.EmailText));
211         members.Save();
212         this.RaisePropertyChanged(() => this.MemberList);
213     }
214 }
215
216 public void ShowUpdateMethod()
217 {
218     if (selectedMember != null)
219     {
220         var updateMemberView = new UpdateView();
221         updateMemberView.Show();
222         var selectedMemberMessage = new ViewModelMessage()
223         {
224             FirstNameText = selectedMember.FirstName,
225             LastNameText = selectedMember.LastName,
226             EmailText = selectedMember.Email
227         };
228         MessengerInstance.Unregister(this);
229         Messenger.Default.Send(selectedMemberMessage);
230
231         Messenger.Default.Register<ViewModelMessage>(this, ↗
232             OnReceiveMessageAction);
233     }
234 }
235
236 public void ExitAppMethod()
237 {
238     System.Windows.Application.Current.Shutdown();
239 }
```



```

1 <Window x:Class="Assignment3Mvvm.MainWindow"
2     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3     xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4     xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5     xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6     xmlns:i="http://schemas.microsoft.com/expression/2010/interactivity"
7     xmlns:ei="http://schemas.microsoft.com/expression/2010/interactions"
8     xmlns:local="clr-namespace:Assignment3Mvvm"
9     mc:Ignorable="d"
10    Title="Membership Maintenance" Height="500" Width="800"
11    ResizeMode="NoResize">
12    <DockPanel LastChildFill="True" DataContext="{Binding MainViewModel, Source=
13    {StaticResource ViewModelLocator}}">
14    <TextBlock Text="Customers: " FontWeight="Bold" DockPanel.Dock="Top"
15    Padding="0,20,0,0" HorizontalAlignment="Center"/>
16    <ListBox DockPanel.Dock="Left" BorderBrush="Black" BorderThickness="2"
17    HorizontalAlignment="Center" VerticalAlignment="Center"
18    ItemsSource="{Binding MemberList}"
19    DisplayMemberPath="GetDisplayText" SelectedItem="{Binding
20    SelectedMember}" Width="500" Height="400">
21    <i:Interaction.Triggers>
22    <i:EventTrigger EventName="MouseDoubleClick">
23    <i:InvokeCommandAction Command="{Binding
24    ShowUpdateCommand}"/>
25    </i:EventTrigger>
26    </i:Interaction.Triggers>
27    </ListBox>
28    <DockPanel DockPanel.Dock="Right">
29    <StackPanel Orientation="Vertical" VerticalAlignment="Center">
30    <Button Content="Add" Width="150" Height="25" Command="{Binding
31    ShowAddCommand}" />
32    <Button Content="Delete" Width="150" Height="25"
33    Command="{Binding DeleteMemberCommand}"/>
34    <Button Content="Exit" Width="150" Height="25" Command="{Binding
35    ExitAppCommand}" />
36    </StackPanel>
37    </DockPanel>
38    </DockPanel>
39 </Window>
40

```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using GalaSoft.MvvmLight;
7 using GalaSoft.MvvmLight.Messaging;
8 using GalaSoft.MvvmLight.Command;
9 using System.Windows.Input;
10 using System.Collections.ObjectModel;
11
12 namespace Assignment3Mvvm.Model
13 {
14     public class Member : ObservableObject
15     {
16         private string firstName;
17         private string lastName;
18         private string email;
19
20         public Member()
21         {
22
23         }
24
25         public Member(string _firstName, string _lastName, string _email)
26         {
27             firstName = _firstName;
28             lastName = _lastName;
29             email = _email;
30         }
31
32         public string FirstName
33         {
34             get
35             {
36                 return firstName;
37             }
38             set
39             {
40                 Set<string>(() => this.FirstName, ref firstName, value);
41             }
42         }
43
44         public string LastName
45         {
46             get
47             {
48                 return lastName;
49             }
50         }
51     }
52 }
```

```
50         set
51         {
52             Set<string>(() => this.LastName, ref lastName, value);
53         }
54     }
55
56     public string Email
57     {
58         get
59         {
60             return email;
61         }
62         set
63         {
64             Set<string>(() => this.Email, ref email, value);
65         }
66     }
67
68     public string GetDisplayText
69     {
70         get { return firstName + " " + lastName + " - " + email; }
71     }
72 }
73 }
74
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.IO;
6 using System.Threading.Tasks;
7 using System.Windows.Input;
8 using GalaSoft.MvvmLight;
9 using GalaSoft.MvvmLight.Messaging;
10 using System.Collections.ObjectModel;
11
12 namespace Assignment3Mvvm.Model
13 {
14     public class MembershipData
15     {
16         private string filename;
17         public MembershipData()
18         {
19             filename = "Memberships.txt";
20         }
21
22         public ObservableCollection<Member> GetMemberships()
23         {
24             ObservableCollection<Member> m = new ObservableCollection<Member>();
25             string currentLine;
26             try
27             {
28                 System.IO.StreamReader file = new System.IO.StreamReader      ↗
29                     (filename);
30                 while ((currentLine = file.ReadLine()) != null)
31                 {
32                     string[] memberFields = currentLine.Split(',');
33                     m.Add(new Member(memberFields[0], memberFields[1],      ↗
34                         memberFields[2]));
35                 }
36                 file.Close();
37                 return m;
38             }
39             catch (FileNotFoundException e)
40             {
41             }
42             return null;
43         }
44
45         public void SaveMemberships(ObservableCollection<Member> members)
46         {
47             string filename = "Memberships.txt";
48             File.WriteAllText(filename, String.Empty);
49         }
50     }
51 }
```

```
48         foreach (Member m in members)
49         {
50             string text = m.FirstName + "," + m.LastName + "," + m.Email +
51                 Environment.NewLine;
52             File.AppendAllText(filename, text);
53         }
54     }
55 }
56
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Windows.Input;
7 using GalaSoft.MvvmLight;
8 using GalaSoft.MvvmLight.Command;
9 using System.Collections.ObjectModel;
10
11 namespace Assignment3Mvvm.Model
12 {
13     public class MembershipList : ObservableObject
14     {
15         public event ChangeHandler Changed;
16         private ObservableCollection<Member> members;
17         private int propertyCount;
18         public MembershipList()
19         {
20             members = new ObservableCollection<Member>();
21             propertyCount = 0;
22         }
23
24         public int PropertyCount
25         {
26             get
27             {
28                 return propertyCount;
29             }
30
31             set
32             {
33                 Set<int>(() => this.PropertyCount, ref propertyCount, value);
34             }
35         }
36
37         public Member this[int i]
38         {
39             get
40             {
41                 return members[i];
42             }
43             set
44             {
45                 var m = members.ElementAt(i);
46                 Set(()=>Members[i], ref m, value);
47                 members[i] = m;
48             }
49         }
50     }
51 }
```

```
50
51     public void Add(Member m)
52     {
53         members.Add(m);
54         propertyCount++;
55         OnChange(EventArgs.Empty);
56     }
57
58     public void Remove(Member m)
59     {
60         members.Remove(m);
61         propertyCount--;
62         OnChange(EventArgs.Empty);
63     }
64
65     public void Write()
66     {
67         MembershipData data = new MembershipData();
68         ObservableCollection<Member> tempM = data.GetMemberships();
69         if (tempM != null)
70         {
71             foreach (Member m in tempM)
72             {
73                 members.Add(m);
74             }
75         }
76     }
77
78     public void Save()
79     {
80         MembershipData data = new MembershipData();
81         data.SaveMemberships(members);
82     }
83
84     public ObservableCollection<Member> Members
85     {
86         get
87         {
88             return members;
89         }
90         set
91         {
92             Set<ObservableCollection<Member>>(() => this.Members, ref members, value);
93         }
94     }
95
96     protected virtual void OnChange(EventArgs e)
97     {
```

```
98         Changed?.Invoke( this);
99     }
100
101
102     }
103
104     public delegate void ChangeHandler(MembershipList m);
105 }
106
```



```

1 <Window x:Class="Assignment3Mvvm.Views.UpdateView"
2       x:Name="UdateMemberWindow"
3       xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
4       xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5       xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
6       xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
7       xmlns:local="clr-namespace:Assignment3Mvvm.Views"
8       mc:Ignorable="d"
9       Title="Update Member" Height="400" Width="500" ResizeMode="NoResize">
10 <Grid DataContext="{Binding UpdateViewModel, Source={StaticResource
    ViewModelLocator}}">
11     <Grid.ColumnDefinitions>
12         <ColumnDefinition Width="150" />
13         <ColumnDefinition Width="350"/>
14     </Grid.ColumnDefinitions>
15     <Grid.RowDefinitions>
16         <RowDefinition Height="80"/>
17         <RowDefinition Height="80"/>
18         <RowDefinition Height="80"/>
19         <RowDefinition Height="80"/>
20     </Grid.RowDefinitions>
21     <TextBlock Grid.Row="0" Grid.Column="0" Text="First Name:"
    VerticalAlignment="Center" HorizontalAlignment="Center"
    FontWeight="Bold"/>
22     <TextBox Grid.Row="0" Grid.Column="1" Height="30" Width="300"
    HorizontalAlignment="Right" Margin="0,0,30,0" BorderThickness="1"
    BorderBrush="Black" Text="{Binding FirstNameTextBox}" />
23     <TextBlock Grid.Row="1" Grid.Column="0" Text="Last Name:"
    VerticalAlignment="Center" HorizontalAlignment="Center"
    FontWeight="Bold"/>
24     <TextBox Grid.Row="1" Grid.Column="1" Height="30" Width="300"
    HorizontalAlignment="Right" Margin="0,0,30,0" BorderBrush="Black"
    BorderThickness="1" Text="{Binding LastNameTextBox}" />
25     <TextBlock Grid.Row="2" Grid.Column="0" Text="Email:"
    HorizontalAlignment="Center" VerticalAlignment="Center"
    FontWeight="Bold" />
26     <TextBox Grid.Row="2" Grid.Column="1" Height="30" Width="300"
    HorizontalAlignment="Right" Margin="0,0,30,0" BorderBrush="Black"
    BorderThickness="1" Text="{Binding EmailTextBox}"/>
27     <Button Grid.ColumnSpan="2" Content="Save" HorizontalAlignment="Left"
    Margin="106,34.6,0,0" Grid.Row="3" VerticalAlignment="Top" Width="75"
    Command="{Binding UpdateCommand, Mode=OneWay}"
    CommandParameter="{Binding ElementName=UdateMemberWindow}"/>
28     <Button Content="Cancel" Grid.Column="1" HorizontalAlignment="Left"
    Margin="165,34.6,0,0" Grid.Row="3" VerticalAlignment="Top" Width="75"
    Command="{Binding CancelCommand}" CommandParameter="{Binding
    ElementName=UdateMemberWindow}"/>
29 </Grid>
30 </Window>

```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Windows;
7 using System.Windows.Input;
8 using Assignment3Mvvm.Views;
9 using Assignment3Mvvm.Messages;
10 using GalaSoft.MvvmLight.Messaging;
11 using GalaSoft.MvvmLight;
12 using GalaSoft.MvvmLight.Command;
13 using Assignment3Mvvm.Model;
14
15 namespace Assignment3Mvvm.ViewModel
16 {
17     public class UpdateViewModel : ViewModelBase
18     {
19         public ICommand UpdateCommand { get; private set; }
20         public ICommand CancelCommand { get; private set; }
21
22         private string firstNameText;
23         private string lastNameText;
24         private string emailText;
25
26         public UpdateViewModel()
27         {
28             Messenger.Default.Register<ViewModelMessage>(this,      ↗
                OnReceiveMessageAction);
29             UpdateCommand = new RelayCommand<IClosable>             ↗
                (this.OnClickUpdateMember, null);
30             CancelCommand = new RelayCommand<IClosable>           ↗
                (this.OnClickCloseWindow, null);
31
32         }
33
34         public void OnReceiveMessageAction(ViewModelMessage m)
35         {
36             firstNameText = m.FirstNameText;
37             RaisePropertyChanged("FirstNameTextBox");
38             lastNameText = m.LastNameText;
39             RaisePropertyChanged("LastNameTextBox");
40             emailText = m.EmailText;
41             RaisePropertyChanged("EmailTextBox");
42         }
43
44         public string FirstNameTextBox
45         {
46             get { return firstNameText; }
```

```
47         set
48         {
49             firstNameText = value;
50             RaisePropertyChanged("FirstNameTextBox");
51         }
52     }
53
54     public string LastNameTextBox
55     {
56         get { return lastNameText; }
57         set
58         {
59             lastNameText = value;
60             RaisePropertyChanged("LastNameTextBox");
61         }
62     }
63
64     public string EmailTextBox
65     {
66         get { return emailText; }
67         set
68         {
69             emailText = value;
70             RaisePropertyChanged("EmailTextBox");
71         }
72     }
73
74     private void OnClickUpdateMember(IClosable window)
75     {
76         if (Validator.IsPresent("First name", firstNameText)
77             && Validator.IsPresent("Last name", lastNameText)
78             && Validator.IsPresent("Email", emailText)
79             && Validator.IsValidEmail(emailText)
80             && Validator.IsWithinRange(firstNameText, 1, 25)
81             && Validator.IsWithinRange(lastNameText, 1, 25)
82             && Validator.IsWithinRange(emailText, 1, 25))
83         {
84             var memberMessage = new ViewModelMessage()
85             {
86                 WindowCommandText = "Update",
87                 FirstNameText = FirstNameTextBox,
88                 LastNameText = LastNameTextBox,
89                 EmailText = EmailTextBox
90             };
91             MessengerInstance.Unregister(this);
92             string notificationText = String.Format("Member Updated: {0} {1} - {2}",
93                 firstNameText, lastNameText, emailText);
94             Messenger.Default.Send<NotificationMessage>(new NotificationMessage(notificationText));
```

```
94         Messenger.Default.Send(memberMessage);
95         //window.Close();
96         if (window != null)
97         {
98             window.Close();
99         }
100     }
101 }
102
103 private void OnClickCloseWindow(IClosable window)
104 {
105     var closeWindowMessage = new ViewModelMessage()
106     {
107         WindowCommandText = "Cancel",
108         FirstNameText = null,
109         LastNameText = null,
110         EmailText = null
111     };
112     Messenger.Default.Send(closeWindowMessage);
113     window.Close();
114 }
115 }
116 }
117
```

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using GalaSoft.MvvmLight;
7 using GalaSoft.MvvmLight.Command;
8 using Assignment3Mvvm.ViewModel;
9 using System.Windows;
10 using System.Windows.Input;
11
12 namespace Assignment3Mvvm.Model
13 {
14     public class Validator
15     {
16         private static string title = "Entity Error";
17
18         public static string Title
19         {
20             get
21             {
22                 return title;
23             }
24             set
25             {
26                 title = value;
27             }
28         }
29
30         public static bool IsPresent(string textBox, string text)
31         {
32             if (text == null)
33             {
34                 MessageBox.Show(textBox + " is required", Title);
35                 //textBox.Focus();
36                 return false;
37             }
38             return true;
39         }
40
41         public static bool IsDecimal(string textBox)
42         {
43             decimal number = 0m;
44             if (Decimal.TryParse(textBox, out number))
45             {
46                 return true;
47             }
48             else
49             {
```

```
50         MessageBox.Show(textBox + " must be a decimal value.", Title);
51         //textBox.Focus();
52         return false;
53     }
54 }
55
56 public static bool IsInt32(string textBox)
57 {
58     int number = 0;
59     if (Int32.TryParse(textBox, out number))
60     {
61         return true;
62     }
63     else
64     {
65         MessageBox.Show(textBox + " must be an integer.", Title);
66         //textBox.Focus();
67         return false;
68     }
69 }
70
71 public static bool IsWithinRange(string textBox, decimal min, decimal max)
72 {
73     int number = textBox.Length;
74     if (number < min || number > max)
75     {
76         MessageBox.Show(textBox + " must be between " + min + " and " + max + " characters.", Title);
77         //textBox.Focus();
78         return false;
79     }
80     return true;
81 }
82
83 public static bool IsValidEmail(string textBox)
84 {
85     if (textBox.IndexOf("@") == -1 || textBox.IndexOf(".") == -1)
86     {
87         MessageBox.Show(textBox + " must be a valid email address", Title);
88         //textBox.Focus();
89         return false;
90     }
91     return true;
92 }
93 }
94 }
95
```

```
1  /*
2   In App.xaml:
3   <Application.Resources>
4       <vm:ViewModelLocator xmlns:vm="clr-namespace:Assignment3Mvvm"
5                               x:Key="Locator" />
6   </Application.Resources>
7
8   In the View:
9   DataContext="{Binding Source={StaticResource Locator}, Path=ViewModelName}"
10
11   You can also use Blend to do all this with the tool's support.
12   See http://www.galasoft.ch/mvvm
13  */
14
15  using GalaSoft.MvvmLight;
16  using GalaSoft.MvvmLight.Ioc;
17  using Microsoft.Practices.ServiceLocation;
18  using System;
19  using System.Windows;
20  using Assignment3Mvvm.ViewModel;
21  using GalaSoft.MvvmLight.Messaging;
22
23  namespace Assignment3Mvvm.ViewModel
24  {
25      /// <summary>
26      /// This class contains static references to all the view models in the
27      /// application and provides an entry point for the bindings.
28      /// </summary>
29      public class ViewModelLocator
30      {
31          /// <summary>
32          /// Initializes a new instance of the ViewModelLocator class.
33          /// </summary>
34          public ViewModelLocator()
35          {
36              ServiceLocator.SetLocatorProvider(() => SimpleIoc.Default);
37
38              ////if (ViewModelBase.IsInDesignModeStatic)
39              ////{
40              ////    // Create design time view services and models
41              ////    SimpleIoc.Default.Register<IDataService, DesignDataService>
42              ////        ();
43              ////}
44              ////else
45              ////{
46              ////    // Create run time view services and models
47              ////    SimpleIoc.Default.Register<IDataService, DataService>();
48              ////}
```

```
49     SimpleIoc.Default.Register<MainViewModel>();
50     Messenger.Default.Register<NotificationMessage>(this, NotifyUserMethod);
51     SimpleIoc.Default.Register<AddViewModel>();
52     SimpleIoc.Default.Register<UpdateViewModel>();
53 }
54
55 public MainViewModel MainViewModel
56 {
57     get
58     {
59         return ServiceLocator.Current.GetInstance<MainViewModel>();
60     }
61 }
62
63 public AddViewModel AddViewModel
64 {
65     get
66     {
67         return ServiceLocator.Current.GetInstance<AddViewModel>();
68     }
69 }
70
71 public UpdateViewModel UpdateViewModel
72 {
73     get
74     {
75         return ServiceLocator.Current.GetInstance<UpdateViewModel>();
76     }
77 }
78
79 private void NotifyUserMethod(NotificationMessage message)
80 {
81     MessageBox.Show(message.Notification);
82 }
83
84 public static void Cleanup()
85 {
86
87     // TODO Clear the ViewModels
88 }
89 }
90 }
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using GalaSoft.MvvmLight;
7 using GalaSoft.MvvmLight.Messaging;
8
9 namespace Assignment3Mvvm.Messages
10 {
11     public class ViewModelMessage : MessageBase
12     {
13         public string FirstNameText { get; set; }
14         public string LastNameText { get; set; }
15         public string EmailText { get; set; }
16         public string WindowCommandText { get; set; }
17     }
18 }
19
```