

```
1 <Application x:Class="Assignment3Mvvm.App"
2           xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3           xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4           xmlns:local="clr-namespace:Assignment3Mvvm"
5           StartupUri="MainWindow.xaml"
6           xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
7           d1p1:Ignorable="d"
8           xmlns:d1p1="http://schemas.openxmlformats.org/markup-compatibility/2006">
9     <Application.Resources>
10       <ResourceDictionary>
11         <vm:ViewModelLocator x:Key="ViewModelLocator" d:IsDataSource="True"
12           xmlns:vm="clr-namespace:Assignment3Mvvm.ViewModel" />
13       </ResourceDictionary>
14     </Application.Resources>
15 </Application>
```

```
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 AddMemberView addMemberView;
35         private Member selectedMember;
36         public ICommand ShowAddCommand { get; private set; }
37         public ICommand DeleteMemberCommand { get; private set; }
38         public ICommand ExitAppCommand { get; private set; }
39
40         public ICommand UpdateMemberCommand { get; private set; }
41         public MainViewModel()
42         {
43             ////if (IsInDesignMode)
44             ////{
45             ////    // Code runs in Blend --> create design time data.
46             ////}
47             ////else
48             ////{
49             ////    // Code runs "for real"
```

```
49         ///}
50         members = new MembershipList();
51         members.Write();
52         ShowAddCommand = new RelayCommand(ShowAddMethod);
53         DeleteMemberCommand = new RelayCommand(DeleteMemberMethod);
54         ExitAppCommand = new RelayCommand(ExitAppMethod);
55         Messenger.Default.Register<ViewModelMessage>(this,           ↗
            OnReceiveMessageAction);

56
57     }
58
59     public ObservableCollection<Member> MemberList
60     {
61         get
62         {
63             return members.Members;
64         }
65     }
66
67     public Member SelectedMember
68     {
69         get { return selectedMember; }
70         set
71         {
72             selectedMember = value;
73             RaisePropertyChanged("SelectedMember");
74         }
75     }
76
77     public AddMemberView AddMemberViewWindow { get => addMemberView; set => ↗
        addMemberView = value; }
78
79     public void ShowAddMethod()
80     {
81         addMemberView = new AddMemberView();
82         addMemberView.Show();
83     }
84
85     public void DeleteMemberMethod()
86     {
87         members.Remove(selectedMember);
88         members.Save();
89         this.RaisePropertyChanged(() => this.MemberList);
90     }
91
92     private void OnReceiveMessageAction(ViewModelMessage m)
93     {
94
95         if (m.WindowCommandText != null && m.WindowCommandText.Equals           ↗
```

```
        ("Close"))
    {
        addMemberView.Close();
    }
    else
    {
        members.Add(new Member(m.FirstNameText, m.LastNameText,
                                m.EmailText));
        addMemberView.Close();
        members.Save();
        this.RaisePropertyChanged(() => this.MemberList);
    }
}

public void ExitAppMethod()
{
    System.Windows.Application.Current.Shutdown();
}
}
```

```
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:local="clr-namespace:Assignment3Mvvm"
7     mc:Ignorable="d"
8     Title="Membership Maintenance" Height="500" Width="800"
9     ResizeMode="NoResize">
10    <DockPanel LastChildFill="True" DataContext="{Binding MainViewModel, Source=
11    {StaticResource ViewModelLocator}}">
12        <TextBlock Text="Customers: " FontWeight="Bold" DockPanel.Dock="Top"
13        Padding="0,20,0,0" HorizontalAlignment="Center"/>
14        <ListBox DockPanel.Dock="Left" BorderBrush="Black" BorderThickness="2"
15        HorizontalAlignment="Center" VerticalAlignment="Center"
16        ItemsSource="{Binding MemberList}"
17        DisplayMemberPath="GetDisplayText" SelectedItem="{Binding
18        SelectedMember}" Width="500" Height="400">
19            <ListBox.InputBindings>
20                <!-- Here is where double click for update member would be -->
21            </ListBox.InputBindings>
22        </ListBox>
23        <DockPanel DockPanel.Dock="Right">
24            <StackPanel Orientation="Vertical" VerticalAlignment="Center">
25                <Button Content="Add" Width="150" Height="25" Command="{Binding
26                ShowAddCommand}" />
27                <Button Content="Delete" Width="150" Height="25"
28                Command="{Binding DeleteMemberCommand}"/>
29                <Button Content="Exit" Width="150" Height="25" Command="{Binding
30                ExitAppCommand}" />
31            </StackPanel>
32        </DockPanel>
33    </DockPanel>
34</Window>
```

```
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             set
31             {
32                 Set<int>(() => this.PropertyCount, ref propertyCount, value);
33             }
34         }
35
36         public void Add(Member m)
37         {
38             members.Add(m);
39             propertyCount++;
40             OnChange(EventArgs.Empty);
41         }
42
43         public void Remove(Member m)
44         {
45             members.Remove(m);
46             propertyCount--;
47             OnChange(EventArgs.Empty);
48         }
49     }
```

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

```
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 }
53
54 public MainViewModel MainViewModel
55 {
56     get
57     {
58         return ServiceLocator.Current.GetInstance<MainViewModel>();
59     }
60 }
61
62 public AddViewModel AddViewModel
63 {
64     get
65     {
66         return ServiceLocator.Current.GetInstance<AddViewModel>();
67     }
68 }
69
70 private void NotifyUserMethod(NotificationMessage message)
71 {
72     MessageBox.Show(message.Notification);
73 }
74
75 public static void Cleanup()
76 {
77     // TODO Clear the ViewModels
78 }
79 }
80 }
```

```
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
```



```

1 <Window x:Class="Assignment3Mvvm.Views.AddMemberView"
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:local="clr-namespace:Assignment3Mvvm.Views"
7     mc:Ignorable="d"
8     Title="Add Member" Height="400" Width="500" ResizeMode="NoResize">
9     <Grid DataContext="{Binding AddViewModel, Source={StaticResource
      ViewModelLocator}}">
10         <Grid.ColumnDefinitions>
11             <ColumnDefinition Width="150" />
12             <ColumnDefinition Width="350"/>
13         </Grid.ColumnDefinitions>
14         <Grid.RowDefinitions>
15             <RowDefinition Height="80"/>
16             <RowDefinition Height="80"/>
17             <RowDefinition Height="80"/>
18             <RowDefinition Height="80"/>
19         </Grid.RowDefinitions>
20         <TextBlock Grid.Row="0" Grid.Column="0" Text="First Name:"
      VerticalAlignment="Center" HorizontalAlignment="Center"
      FontWeight="Bold"/>
21         <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}" />
22         <TextBlock Grid.Row="1" Grid.Column="0" Text="Last Name:"
      VerticalAlignment="Center" HorizontalAlignment="Center"
      FontWeight="Bold"/>
23         <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}" />
24         <TextBlock Grid.Row="2" Grid.Column="0" Text="Email:"
      HorizontalAlignment="Center" VerticalAlignment="Center"
      FontWeight="Bold" />
25         <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}"/>
26         <Button Grid.ColumnSpan="2" Content="Save" HorizontalAlignment="Left"
      Margin="106,34.6,0,0" Grid.Row="3" VerticalAlignment="Top" Width="75"
      Command="{Binding AddCommand}"/>
27         <Button Content="Cancel" Grid.Column="1" HorizontalAlignment="Left"
      Margin="165,34.6,0,0" Grid.Row="3" VerticalAlignment="Top" Width="75"
      Command="{Binding CancelCommand}"/>
28     </Grid>
29 </Window>
30

```

```
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(OnClickAddMember, null);
29             CancelCommand = new RelayCommand(OnClickCloseWindow, null);
30
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()
64 {
65     if (Validator.IsPresent("First name", firstNameText)
66         && Validator.IsPresent("Last name", lastNameText)
67         && Validator.IsPresent("Email", emailText)
68         && Validator.IsValidEmail(emailText)
69         && Validator.IsWithinRange(firstNameText, 1, 25)
70         && Validator.IsWithinRange(lastNameText, 1, 25)
71         && Validator.IsWithinRange(emailText, 1, 25))
72     {
73         var memberMessage = new ViewModelMessage()
74         {
75             FirstNameText = FirstNameTextBox,
76             LastNameText = LastNameTextBox,
77             EmailText = EmailTextBox
78         };
79         string notificationText = String.Format("New member added: {0} {1} - {2}",
80             firstNameText, lastNameText, emailText);
81         firstNameText = null;
82         lastNameText = null;
83         emailText = null;
84         Messenger.Default.Send<NotificationMessage>(new NotificationMessage(notificationText));
85         Messenger.Default.Send(memberMessage);
86     }
87
88 private void OnClickCloseWindow()
89 {
90     var closeWindowMessage = new ViewModelMessage()
91     {
92         WindowCommandText = "Close",
93         FirstNameText = null,
94         LastNameText = null,
95         EmailText = null
96     };
97 }
```

```
97         Messenger.Default.Send(closeWindowMessage);
98     }
99 }
100 }
101
```