

İsimler Sözlüğü Uygulaması

Çocuklarına isim koymak isteyenler için bir Windows Phone uygulaması

Table of Contents

Symbol Reference	1
IsimSozlugu Namespace	1
IsimSozlugu.Resources Namespace	1
Classes	2
AppResources Class	2
Classes	4
AboutPage Class	4
AboutPage>AboutPage Constructor	5
All Class	6
All Properties	6
AlphaKeyGroup Class	7
AlphaKeyGroup.AlphaKeyGroup Constructor	8
AlphaKeyGroup Methods	8
AlphaKeyGroup Properties	9
AlphaKeyGroup Delegates	9
App Class	9
App.App Constructor	10
App Methods	11
App Properties	11
AppSettings Class	11
AppSettings Properties	12
BackgroundColorSettingsPage Class	13
BackgroundColorSettingsPage.BackgroundColorSettingsPage Constructor	14
BackgroundColorSettingsPage Fields	14
BackgroundColorSettingsPage Methods	14
Favourite Class	15
Favourite Properties	15
Female Class	16
Female Properties	16
FontFamilySettingsPage Class	17
FontFamilySettingsPage.FontFamilySettingsPage Constructor	17
FontFamilySettingsPage Fields	18
FontFamilySettingsPage Methods	18
FontSizeSettingsPage Class	19
FontSizeSettingsPage.FontSizeSettingsPage Constructor	20
FontSizeSettingsPage Fields	20

FontSizeSettingsPage Methods	20
GeneralSettingsPage Class	21
GeneralSettingsPage.GeneralSettingsPage Constructor	22
GeneralSettingsPage Methods	23
LocalizedStrings Class	23
LocalizedStrings Properties	24
MainPage Class	24
MainPage.MainPage Constructor	24
Male Class	25
Male Properties	25
MyUpdate Class	26
MyUpdate Properties	26
NameDataContext Class	27
NameDataContext.NameDataContext Constructor	27
NameDataContext Fields	27
NameDetailPage Class	29
NameDetailPage.NameDetailPage Constructor	29
NameDetailPage Fields	30
NameDetailPage Methods	31
SearchPage Class	33
SearchPage.SearchPage Constructor	33
SearchPage Methods	34
StatisticsPage Class	34
StatisticsPage.StatisticsPage Constructor	35
StatisticsPage Methods	35
Unisex Class	36
Unisex Properties	36
Files	37
AboutPage.xaml.cs	37
All.cs	40
AlphaKeyGroup.cs	41
App.xaml.cs	42
AppResources.Designer.cs	49
AppSettings.cs	52
AssemblyInfo.cs	52
BackgroundColorSettingsPage.xaml.cs	53
Favourite.cs	56
Female.cs	57
FontFamilySettingsPage.xaml.cs	57
FontSizeSettingsPage.xaml.cs	60

GeneralSettingsPage.xaml.cs	62
IsimSozlugu.csproj	67
IsimSozlugu.sln	67
LocalizedStrings.cs	67
MainPage.xaml.cs	68
Male.cs	72
MyUpdate.cs	73
NameDataContext.cs	73
NameDetailPage.xaml.cs	74
SearchPage.xaml.cs	84
StatisticsPage.xaml.cs	87
Unisex.cs	90

Index

a

1 Symbol Reference





















1.1 IsimSozlugu Namespace

This is namespace IsimSozlugu.

Namespaces

Name	Description
Resources (see page 1)	This is namespace IsimSozlugu.Resources.

Classes

	Name	Description
	AboutPage (see page 4)	This is class IsimSozlugu>AboutPage.
	All (see page 6)	This is class IsimSozlugu.All.
	AlphaKeyGroup (see page 7)	This is class IsimSozlugu.AlphaKeyGroup.
	App (see page 9)	This is class IsimSozlugu.App.
	AppSettings (see page 11)	This is class IsimSozlugu.AppSettings.
	BackgroundColorSettingsPage (see page 13)	This is class IsimSozlugu.BackgroundColorSettingsPage.
	Favourite (see page 15)	This is class IsimSozlugu.Favourite.
	Female (see page 16)	This is class IsimSozlugu.Female.
	FontFamilySettingsPage (see page 17)	This is class IsimSozlugu.FontFamilySettingsPage.
	FontSizeSettingsPage (see page 19)	This is class IsimSozlugu.FontSizeSettingsPage.
	GeneralSettingsPage (see page 21)	This is class IsimSozlugu.GeneralSettingsPage.
	LocalizedString (see page 23)	Provides access to string resources.
	MainPage (see page 24)	This is class IsimSozlugu.MainPage.
	Male (see page 25)	This is class IsimSozlugu.Male.
	MyUpdate (see page 26)	This is class IsimSozlugu.MyUpdate.
	NameDataContext (see page 27)	This is class IsimSozlugu.NameDataContext.
	NameDetailPage (see page 29)	This is class IsimSozlugu.NameDetailPage.
	SearchPage (see page 33)	This is class IsimSozlugu.SearchPage.
	StatisticsPage (see page 34)	This is class IsimSozlugu.StatisticsPage.
	Unisex (see page 36)	This is class IsimSozlugu.Unisex.

1.1.1 IsimSozlugu.Resources Namespace

This is namespace IsimSozlugu.Resources.


Classes

	Name	Description
	AppResources (see page 2)	A strongly-typed resource class, for looking up localized strings, etc.

1.1.1.1 Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	AppResources (see page 2)	A strongly-typed resource class, for looking up localized strings, etc.

1.1.1.1.1 AppResources Class

A strongly-typed resource class, for looking up localized strings, etc.

Class Hierarchy

[IsimSozlugu.Resources.AppResources](#)

C#

```
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",
"4.0.0.0")]
[global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
[global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
public class AppResources;
```


File

AppResources.Designer.cs ([see page 49](#))








Description

This class was auto-generated by the StronglyTypedResourceBuilder class via a tool like ResGen or Visual Studio. To add or remove a member, edit your .ResX file then rerun ResGen with the /str option, or rebuild your VS project.

Methods

	Name	Description
	AppResources (see page 3)	This is AppResources, a member of class AppResources.

AppResources Properties

	Name	Description
	AppBarButtonText (see page 3)	Looks up a localized string similar to button.
	AppBarMenuItemText (see page 3)	Looks up a localized string similar to menu item.
	ApplicationTitle (see page 3)	Looks up a localized string similar to MY APPLICATION.
	Culture (see page 3)	Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.
	ResourceFlowDirection (see page 3)	Looks up a localized string similar to LeftToRight.
	ResourceLanguage (see page 3)	Looks up a localized string similar to us-EN.
	ResourceManager (see page 4)	Returns the cached ResourceManager instance used by this class.

1.1.1.1.1.1 AppResources.AppResources Constructor

C#

```
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",  
"CA1811:AvoidUncalledPrivateCode")]  
internal AppResources();
```

Description

This is AppResources, a member of class AppResources.

Body Source

```
1:  
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",  
"CA1811:AvoidUncalledPrivateCode")]  
2: internal AppResources()  
3: {  
4: }
```

1.1.1.1.1.2 AppResources Properties

1.1.1.1.1.2.1 AppResources.AppBarButtonText Property

Looks up a localized string similar to button.

C#

```
public static string AppBarButtonText;
```

1.1.1.1.1.2.2 AppResources.AppBarMenuItemText Property

Looks up a localized string similar to menu item.

C#

```
public static string AppBarMenuItemText;
```

1.1.1.1.1.2.3 AppResources.ApplicationTitle Property

Looks up a localized string similar to MY APPLICATION.

C#

```
public static string ApplicationTitle;
```

1.1.1.1.1.2.4 AppResources.Culture Property

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

C#

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]  
public static global::System.Globalization.CultureInfo Culture;
```

1.1.1.1.1.2.5 AppResources.ResourceFlowDirection Property

Looks up a localized string similar to LeftToRight.

C#

```
public static string ResourceFlowDirection;
```

1.1.1.1.1.2.6 AppResources.ResourceLanguage Property

Looks up a localized string similar to us-EN.

C#

```
public static string ResourceLanguage;
```

1.1.1.1.1.2.7 AppResources.ResourceManager Property

Returns the cached ResourceManager instance used by this class.





















C#

```
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
public static global::System.Resources.ResourceManager ResourceManager;
```

1.1.2 Classes

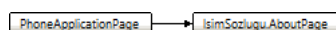
The following table lists classes in this documentation.

Classes

	Name	Description
	AboutPage (see page 4)	This is class IsimSozlugu>AboutPage.
	All (see page 6)	This is class IsimSozlugu.All.
	AlphaKeyGroup (see page 7)	This is class IsimSozlugu.AlphaKeyGroup.
	App (see page 9)	This is class IsimSozlugu.App.
	AppSettings (see page 11)	This is class IsimSozlugu.AppSettings.
	BackgroundColorSettingsPage (see page 13)	This is class IsimSozlugu.BackgroundColorSettingsPage.
	Favourite (see page 15)	This is class IsimSozlugu.Favourite.
	Female (see page 16)	This is class IsimSozlugu.Female.
	FontFamilySettingsPage (see page 17)	This is class IsimSozlugu.FontFamilySettingsPage.
	FontSizeSettingsPage (see page 19)	This is class IsimSozlugu.FontSizeSettingsPage.
	GeneralSettingsPage (see page 21)	This is class IsimSozlugu.GeneralSettingsPage.
	LocalizedStrings (see page 23)	Provides access to string resources.
	MainPage (see page 24)	This is class IsimSozlugu.MainPage.
	Male (see page 25)	This is class IsimSozlugu.Male.
	MyUpdate (see page 26)	This is class IsimSozlugu.MyUpdate.
	NameDataContext (see page 27)	This is class IsimSozlugu.NameDataContext.
	NameDetailPage (see page 29)	This is class IsimSozlugu.NameDetailPage.
	SearchPage (see page 33)	This is class IsimSozlugu.SearchPage.
	StatisticsPage (see page 34)	This is class IsimSozlugu.StatisticsPage.
	Unisex (see page 36)	This is class IsimSozlugu.Unisex.

1.1.2.1 AboutPage Class

Class Hierarchy



C#

```
public class AboutPage : PhoneApplicationPage;
```

File

AboutPage.xaml.cs (see page 37)

Description

This is class IsimSozlugu.AboutPage.

Methods

	Name	Description
	AboutPage (see page 5)	This is AboutPage, a member of class AboutPage.

1.1.2.1.1 AboutPage.AboutPage Constructor**C#**

```
public AboutPage();
```

Description

This is AboutPage, a member of class AboutPage.

Body Source

```
1: public AboutPage()
2: {
3:     InitializeComponent();
4:
5:     SetBackgroundColor();
6:
7:     ApplicationBar = new ApplicationBar();
8:
9:     ApplicationBarIconButton button2 = new ApplicationBarIconButton();
10:    button2.IconUri = new Uri("/Assets/SendWithMail.png", UriKind.Relative);
11:    button2.Text = "İletisime Geç";
12:    ApplicationBar.Buttons.Add(button2);
13:    button2.Click += new EventHandler(SendMailButton_Click);
14:
15:    ApplicationBarIconButton button3 = new ApplicationBarIconButton();
16:    button3.IconUri = new Uri("/Assets/Rate.png", UriKind.Relative);
17:    button3.Text = "Puanla";
18:    ApplicationBar.Buttons.Add(button3);
19:    button3.Click += new EventHandler(RateButton_Click);
20:
21:    //lblAboutTheApp.Text = AppResources.AboutTheApp;
22:    //txtAbout2.Text = AppResources.AboutTheAppText;
23:    //var paragraph = new Paragraph();
24:    //paragraph.Inlines.Add(AppResources.AboutTheAppText);
25:    //txtAbout.Blocks.Add(paragraph);
26:    txtAbout.Text = "Herkes kendi isminin ne anlama geldigini merak eder. Bu uygulama
ile sadece kendi" +
27:        " isminizin degil ayni zamanda pek çok ismin de anlamini öğrenebileceksiniz." +
28:        " Yeni baba veya anne olacak insanların (benim gibi) aklinda cocuguna ne isim
koyacagi gibi" +
29:        " bir soru olmamasi için (yani aslında biraz da kendim için) bu uygulamayi
gelistirdim." +
30:        " Ancak bu uygulamayi hep beraber gelistirebiliriz. Program vasitasiyla bana
göndereceginiz " +
31:        " kelimeler incelendikten sonra bu havuza atilacak ve herkesin
yararlanabilecegi güzel bir " +
32:        " kaynak ortaya çıkacak. Simdiden iyi günler.";
33:    //txtAbout.IsEnabled = false;
34:    txtAbout.IsReadOnly = true;
35:    //this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
```

```
36: }
```

1.1.2.2 All Class

Class Hierarchy

C#

```
[Table]
public class All;
```

File

All.cs (see page 40)

Description

This is class IsimSozlugu.All.

All Properties

	Name	Description
	AllGender (see page 6)	This is AllGender, a member of class All.
	AllId (see page 6)	This is AllId, a member of class All.
	AllMeaning (see page 6)	This is AllMeaning, a member of class All.
	AllName (see page 7)	This is AllName, a member of class All.
	AllNameMeaning (see page 7)	This is AllNameMeaning, a member of class All.

1.1.2.2.1 All Properties

1.1.2.2.1.1 All.AllGender Property

C#

```
[Column]
public string AllGender;
```

Description

This is AllGender, a member of class All.

1.1.2.2.1.2 All.AllId Property

C#

```
[Column(IsPrimaryKey = true, IsDbGenerated = true, DbType = "INT NOT NULL Identity",
CanBeNull = false)]
public int AllId;
```

Description

This is AllId, a member of class All.

1.1.2.2.1.3 All.AllMeaning Property

C#

```
[Column]
public string AllMeaning;
```

Description

This is AllMeaning, a member of class All.

1.1.2.2.1.4 All.AllName Property**C#**

```
[Column]
public string AllName;
```

Description

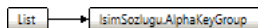
This is AllName, a member of class All.

1.1.2.2.1.5 All.AllNameMeaning Property**C#**

```
[Column]
public string AllNameMeaning;
```

Description

This is AllNameMeaning, a member of class All.

1.1.2.3 AlphaKeyGroup Class**Class Hierarchy****C#**

```
public class AlphaKeyGroup<T> : List<T>;
```

File

AlphaKeyGroup.cs (see page 41)

Description

This is class IsimSozlugu.AlphaKeyGroup.

Methods

	Name	Description
	AlphaKeyGroup (see page 8)	Public constructor.

AlphaKeyGroup Delegates

Name	Description
GetKeyDelegate (see page 9)	The delegate that is used to get the key information.

AlphaKeyGroup Methods

	Name	Description
	CreateGroups (see page 8)	Create a list of AlphaGroup with keys set by a SortedLocaleGrouping.

AlphaKeyGroup Properties

	Name	Description
	Key (see page 9)	The Key of this group.

1.1.2.3.1 AlphaKeyGroup.AlphaKeyGroup Constructor

Public constructor.

C#

```
public AlphaKeyGroup(string key);
```

Parameters

Parameters	Description
string key	The key for this group.

Body Source

```
1: public AlphaKeyGroup(string key)
2: {
3:     Key = key;
4: }
```

1.1.2.3.2 AlphaKeyGroup Methods

1.1.2.3.2.1 AlphaKeyGroup.CreateGroups Method

Create a list of AlphaGroup with keys set by a SortedLocaleGrouping.

C#

```
public static List<AlphaKeyGroup<T>> CreateGroups(IEnumerable<T> items, CultureInfo ci,
GetKeyDelegate getKey, bool sort);
```

Parameters

Parameters	Description
IEnumerable<T> items	The items to place in the groups.
CultureInfo ci	The CultureInfo to group and sort by.
GetKeyDelegate getKey	A delegate to get the key from an item.
bool sort	Will sort the data if true.

Returns

An items source for a LongListSelector

Body Source

```
1: public static List<AlphaKeyGroup<T>> CreateGroups(IEnumerable<T> items, CultureInfo ci,
GetKeyDelegate getKey, bool sort)
2: {
3:     SortedLocaleGrouping slg = new SortedLocaleGrouping(ci);
4:     List<AlphaKeyGroup<T>> list = CreateGroups(slg);
5:
6:     foreach (T item in items)
7:     {
8:         int index = 0;
9:         if (slg.SupportsPhonetics)
10:        {
11:            //check if your database has yomi string for item
12:            //if it does not, then do you want to generate Yomi or ask the user for
this item.
13:            //index = slg.GetGroupIndex(getKey(Yomiof(item)));
14:        }
15:        else
16:        {
17:            index = slg.GetGroupIndex(getKey(item));
18:        }
```

```

19:         if (index >= 0 && index < list.Count)
20:         {
21:             list[index].Add(item);
22:         }
23:     }
24:
25:     if (sort)
26:     {
27:         foreach (AlphaKeyGroup<T> group in list)
28:         {
29:             group.Sort((c0, c1) => { return ci.CompareInfo.Compare(getKey(c0),
30:             getKey(c1)); });
31:         }
32:     }
33:     return list;
34: }

```

1.1.2.3.3 AlphaKeyGroup Properties

1.1.2.3.3.1 AlphaKeyGroup.Key Property

The Key of this group.

C#

```
public string Key;
```

1.1.2.3.4 AlphaKeyGroup Delegates

1.1.2.3.4.1 AlphaKeyGroup.GetKeyDelegate Delegate

The delegate that is used to get the key information.

C#

```
public delegate string GetKeyDelegate(T item);
```

Parameters

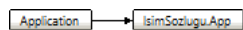
Parameters	Description
item	An object of type T

Returns

The key value to use for this object

1.1.2.4 App Class

Class Hierarchy



C#

```
public class App : Application;
```


File

App.xaml.cs (see page 42)

Description

This is class IsimSozlugu.App.


Methods

	Name	Description
	App (see page 10)	Constructor for the Application object.

App Methods

	Name	Description
	ReadFile (see page 11)	This is ReadFile, a member of class App.

App Properties

	Name	Description
	RootFrame (see page 11)	Provides easy access to the root frame of the Phone Application.

1.1.2.4.1 App.App Constructor

Constructor for the Application object.

C#

```
public App();
```

Body Source

```
1: public App()
2: {
3:     // Global handler for uncaught exceptions.
4:     UnhandledException += Application_UnhandledException;
5:
6:     // ayarlardan teması açık renk bile olsa
7:     // kapalı gibi çalışmasını sağlayacak bir nuget paketi yüklendi
8:     // bu sorunu gideriyor
9:     ThemeManager.ToDarkTheme();
10:
11:     // Standard XAML initialization
12:     InitializeComponent();
13:
14:     // Phone-specific initialization
15:     InitializePhoneApplication();
16:
17:     // Language display initialization
18:     InitializeLanguage();
19:
20:     // Show graphics profiling information while debugging.
21:     if (Debugger.IsAttached)
22:     {
23:         // Display the current frame rate counters.
24:         Application.Current.Host.Settings.EnableFrameRateCounter = true;
25:
26:         // Show the areas of the app that are being redrawn in each frame.
27:         //Application.Current.Host.Settings.EnableRedrawRegions = true;
28:
29:         // Enable non-production analysis visualization mode,
30:         // which shows areas of a page that are handed off to GPU with a colored
31:         // overlay.
32:         //Application.Current.Host.Settings.EnableCacheVisualization = true;
33:
34:         // Prevent the screen from turning off while under the debugger by disabling
35:         // the application's idle detection.
36:         // Caution:- Use this under debug mode only. Application that disables user
37:         // idle detection will continue to run
38:         // and consume battery power when the user is not using the phone.
39:         PhoneApplicationService.Current.UserIdleDetectionMode =
IdleDetectionMode.Disabled;
40:     }
```

```
40: }
```

1.1.2.4.2 App Methods

1.1.2.4.2.1 App.ReadFile Method

C#

```
public string ReadFile(string filePath);
```

Description

This is ReadFile, a member of class App.

Body Source

```
1: public string ReadFile(string filePath)
2: {
3:     var ResrouceStream = Application.GetResourceStream(new Uri(filePath,
UriKind.Relative));
4:     if (ResrouceStream != null)
5:     {
6:         Stream myFileStream = ResrouceStream.Stream;
7:         if (myFileStream.CanRead)
8:         {
9:             StreamReader myStreamReader = new StreamReader(myFileStream);
10:
11:             return myStreamReader.ReadToEnd();
12:         }
13:     }
14:     return "";
15: }
```

1.1.2.4.3 App Properties

1.1.2.4.3.1 App.RootFrame Property

Provides easy access to the root frame of the Phone Application.

C#

```
public static PhoneApplicationFrame RootFrame;
```

Returns

The root frame of the Phone Application.

1.1.2.5 AppSettings Class

Class Hierarchy

```
IsimSozlugu.AppSettings
```

C#

```
[Table]
public class AppSettings;
```







File

AppSettings.cs (📄 see page 52)

Description

This is class IsimSozlugu.AppSettings.

AppSettings Properties

	Name	Description
	AppBackgroundColor (see page 12)	This is AppBackgroundColor, a member of class AppSettings.
	AppBackgroundImage (see page 12)	This is AppBackgroundImage, a member of class AppSettings.
	AppLangName (see page 12)	This is AppLangName, a member of class AppSettings.
	AppSettingsId (see page 12)	This is AppSettingsId, a member of class AppSettings.
	FontFamily (see page 13)	This is FontFamily, a member of class AppSettings.
	FontSize (see page 13)	This is FontSize, a member of class AppSettings.

1.1.2.5.1 AppSettings Properties**1.1.2.5.1.1 AppSettings.AppBackgroundColor Property****C#**

```
[Column]
public string AppBackgroundColor;
```

Description

This is AppBackgroundColor, a member of class AppSettings.

1.1.2.5.1.2 AppSettings.AppBackgroundImage Property**C#**

```
[Column(DbType = "Image", UpdateCheck = UpdateCheck.Never)]
public byte AppBackgroundImage;
```

Description

This is AppBackgroundImage, a member of class AppSettings.

1.1.2.5.1.3 AppSettings.AppLangName Property**C#**

```
[Column]
public string AppLangName;
```

Description

This is AppLangName, a member of class AppSettings.

1.1.2.5.1.4 AppSettings.AppSettingsId Property**C#**

```
[Column(IsPrimaryKey = true, IsDbGenerated = true, DbType = "INT NOT NULL Identity",
CanBeNull = false)]
public int AppSettingsId;
```

Description

This is AppSettingsId, a member of class AppSettings.

1.1.2.5.1.5 AppSettings.FontFamily Property

C#

```
[Column]
public string FontFamily;
```

Description

This is FontFamily, a member of class AppSettings.

1.1.2.5.1.6 AppSettings.FontSize Property

C#

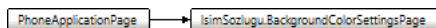
```
[Column]
public string FontSize;
```

Description

This is FontSize, a member of class AppSettings.

1.1.2.6 BackgroundColorSettingsPage Class

Class Hierarchy



C#

```
public class BackgroundColorSettingsPage : PhoneApplicationPage;
```

File

BackgroundColorSettingsPage.xaml.cs ([see page 53](#))

Description

This is class IsimSozlugu.BackgroundColorSettingsPage.

Methods

	Name	Description
	BackgroundColorSettingsPage (see page 14)	This is BackgroundColorSettingsPage, a member of class BackgroundColorSettingsPage.

BackgroundColorSettingsPage Fields

	Name	Description
	artistId (see page 14)	This is artistId, a member of class BackgroundColorSettingsPage.

BackgroundColorSettingsPage Methods

	Name	Description
	OnFragmentNavigation (see page 14)	This is OnFragmentNavigation, a member of class BackgroundColorSettingsPage.
	OnNavigatedFrom (see page 15)	This is OnNavigatedFrom, a member of class BackgroundColorSettingsPage.
	OnNavigatedTo (see page 15)	This is OnNavigatedTo, a member of class BackgroundColorSettingsPage.

1.1.2.6.1 BackgroundColorSettingsPage.BackgroundColorSettingsPage Constructor

C#

```
public BackgroundColorSettingsPage();
```

Description

This is BackgroundColorSettingsPage, a member of class BackgroundColorSettingsPage.

Body Source

```
1: public BackgroundColorSettingsPage()  
2: {  
3:     InitializeComponent();  
4:  
5:     lstBackgroundColor.Items.Clear();  
6:     lstBackgroundColor.Items.Add("Black");  
7:     lstBackgroundColor.Items.Add("Blue");  
8:     lstBackgroundColor.Items.Add("Brown");  
9:     lstBackgroundColor.Items.Add("Gray");  
10:    lstBackgroundColor.Items.Add("Green");  
11:    lstBackgroundColor.Items.Add("Orange");  
12:    lstBackgroundColor.Items.Add("Purple");  
13:    lstBackgroundColor.Items.Add("Red");  
14:    lstBackgroundColor.Items.Add("Yellow");  
15:    lstBackgroundColor.SelectedIndex = -1;  
16:  
17:    SetBackgroundColor();  
18: }
```

1.1.2.6.2 BackgroundColorSettingsPage Fields

1.1.2.6.2.1 BackgroundColorSettingsPage.artistId Field

C#

```
public int artistId;
```

Description

This is artistId, a member of class BackgroundColorSettingsPage.

1.1.2.6.3 BackgroundColorSettingsPage Methods

1.1.2.6.3.1 BackgroundColorSettingsPage.OnFragmentNavigation Method

C#

```
protected override void OnFragmentNavigation(FragmentNavigationEventArgs e);
```

Description

This is OnFragmentNavigation, a member of class BackgroundColorSettingsPage.

Body Source

```
1: protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)  
2: {  
3:     // displays "Fragment: Detail"  
4:     //MessageBox.Show("Folder Id: " + e.Fragment);  
5:     base.OnFragmentNavigation(e);  
6: }
```

1.1.2.6.3.2 BackgroundColorSettingsPage.OnNavigatedFrom Method

C#

```
protected override void OnNavigatedFrom(NavigationEventArgs e);
```

Description

This is OnNavigatedFrom, a member of class BackgroundColorSettingsPage.

Body Source

```
1: protected override void OnNavigatedFrom(NavigationEventArgs e)
2: {
3:     base.OnNavigatedFrom(e);
4:     //while (NavigationService.CanGoBack)
5:     //NavigationService.RemoveBackEntry();
6:
7: }
```

1.1.2.6.3.3 BackgroundColorSettingsPage.OnNavigatedTo Method

C#

```
protected override void OnNavigatedTo(NavigationEventArgs e);
```

Description

This is OnNavigatedTo, a member of class BackgroundColorSettingsPage.

Body Source

```
1: protected override void OnNavigatedTo(NavigationEventArgs e)
2: {
3:     base.OnNavigatedTo(e);
4:     //SetBackgroundColor();
5:     //while (NavigationService.CanGoBack)
6:     //NavigationService.RemoveBackEntry();
7:
8: }
```

1.1.2.7 Favourite Class

Class Hierarchy

IsimSozlugu.Favourite

C#

```
[Table]
public class Favourite;
```



File

Favourite.cs (see page 56)

Description

This is class IsimSozlugu.Favourite.

Favourite Properties

	Name	Description
	FavouriteId (see page 16)	This is FavouriteId, a member of class Favourite.
	FavouriteName (see page 16)	This is FavouriteName, a member of class Favourite.

1.1.2.7.1 Favourite Properties

1.1.2.7.1.1 Favourite.FavouriteId Property

C#

```
[Column(IsPrimaryKey = true, IsDbGenerated = true, DbType = "INT NOT NULL Identity",
CanBeNull = false)]
public int FavouriteId;
```

Description

This is FavouriteId, a member of class Favourite.

1.1.2.7.1.2 Favourite.FavouriteName Property

C#

```
[Column]
public string FavouriteName;
```

Description

This is FavouriteName, a member of class Favourite.

1.1.2.8 Female Class

Class Hierarchy

IsimSozlugu.Female

C#

```
[Table]
public class Female;
```




File

Female.cs (see page 57)

Description

This is class IsimSozlugu.Female.

Female Properties

	Name	Description
	FemaleId (see page 16)	This is FemaleId, a member of class Female.
	FemaleMeaning (see page 17)	This is FemaleMeaning, a member of class Female.
	FemaleName (see page 17)	This is FemaleName, a member of class Female.

1.1.2.8.1 Female Properties

1.1.2.8.1.1 Female.FemaleId Property

C#

```
[Column(IsPrimaryKey = true, IsDbGenerated = true, DbType = "INT NOT NULL Identity",
CanBeNull = false)]
public int FemaleId;
```

Description

This is FemaleId, a member of class Female.

1.1.2.8.1.2 Female.FemaleMeaning Property

C#

```
[Column]
public string FemaleMeaning;
```

Description

This is FemaleMeaning, a member of class Female.

1.1.2.8.1.3 Female.FemaleName Property

C#

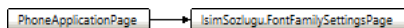
```
[Column]
public string FemaleName;
```

Description

This is FemaleName, a member of class Female.

1.1.2.9 FontFamilySettingsPage Class

Class Hierarchy



C#

```
public class FontFamilySettingsPage : PhoneApplicationPage;
```

File

FontFamilySettingsPage.xaml.cs (see page 57)

Description

This is class IsimSozlugu.FontFamilySettingsPage.

Methods

	Name	Description
	FontFamilySettingsPage (see page 17)	This is FontFamilySettingsPage, a member of class FontFamilySettingsPage.

FontFamilySettingsPage Fields

	Name	Description
	artistId (see page 18)	This is artistId, a member of class FontFamilySettingsPage.

FontFamilySettingsPage Methods

	Name	Description
	OnFragmentNavigation (see page 18)	This is OnFragmentNavigation, a member of class FontFamilySettingsPage.
	OnNavigatedFrom (see page 19)	This is OnNavigatedFrom, a member of class FontFamilySettingsPage.
	OnNavigatedTo (see page 19)	This is OnNavigatedTo, a member of class FontFamilySettingsPage.

1.1.2.9.1 FontFamilySettingsPage.FontFamilySettingsPage Constructor

C#

```
public FontFamilySettingsPage();
```

Description

This is FontFamilySettingsPage, a member of class FontFamilySettingsPage.

Body Source

```
1: public FontFamilySettingsPage()  
2: {  
3:     InitializeComponent();  
4:  
5:     lstFontFamily.Items.Clear();  
6:     lstFontFamily.Items.Add("Arial");  
7:     lstFontFamily.Items.Add("Arial Black");  
8:     lstFontFamily.Items.Add("Baskerville Old Face");  
9:     lstFontFamily.Items.Add("Berlin Sans FB");  
10:    lstFontFamily.Items.Add("Albumman Old Style");  
11:    lstFontFamily.Items.Add("Calibri");  
12:    lstFontFamily.Items.Add("Cambria");  
13:    lstFontFamily.Items.Add("Candara");  
14:    lstFontFamily.Items.Add("Comic Sans MS");  
15:    lstFontFamily.Items.Add("Consolas");  
16:    lstFontFamily.Items.Add("Constantia");  
17:    lstFontFamily.Items.Add("Courier New");  
18:    lstFontFamily.Items.Add("DokChampa");  
19:    lstFontFamily.Items.Add("Ebrima");  
20:    lstFontFamily.Items.Add("Georgia");  
21:    lstFontFamily.Items.Add("Lucida Sans Unicode");  
22:    lstFontFamily.Items.Add("Meiryo UI");  
23:    lstFontFamily.Items.Add("Microsoft YaHei");  
24:    lstFontFamily.Items.Add("Malgun Gothic");  
25:    lstFontFamily.Items.Add("Segoe UI");  
26:    lstFontFamily.Items.Add("Segoe WP");  
27:    lstFontFamily.Items.Add("Tahoma");  
28:    lstFontFamily.Items.Add("Trebuchet MS");  
29:    lstFontFamily.Items.Add("Times New Roman");  
30:    lstFontFamily.Items.Add("Verdana");  
31:    lstFontFamily.SelectedIndex = -1;  
32: }
```

1.1.2.9.2 FontFamilySettingsPage Fields

1.1.2.9.2.1 FontFamilySettingsPage.artistId Field

C#

```
public int artistId;
```

Description

This is artistId, a member of class FontFamilySettingsPage.

1.1.2.9.3 FontFamilySettingsPage Methods

1.1.2.9.3.1 FontFamilySettingsPage.OnFragmentNavigation Method

C#

```
protected override void OnFragmentNavigation(FragmentNavigationEventArgs e);
```

Description

This is OnFragmentNavigation, a member of class FontFamilySettingsPage.

Body Source

```
1: protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
```

```

2: {
3:     // displays "Fragment: Detail"
4:     //MessageBox.Show("Folder Id: " + e.Fragment);
5:     base.OnFragmentNavigation(e);
6: }

```

1.1.2.9.3.2 FontFamilySettingsPage.OnNavigatedFrom Method

C#

```
protected override void OnNavigatedFrom(NavigationEventArgs e);
```

Description

This is OnNavigatedFrom, a member of class FontFamilySettingsPage.

Body Source

```

1: protected override void OnNavigatedFrom(NavigationEventArgs e)
2: {
3:     base.OnNavigatedFrom(e);
4: }

```

1.1.2.9.3.3 FontFamilySettingsPage.OnNavigatedTo Method

C#

```
protected override void OnNavigatedTo(NavigationEventArgs e);
```

Description

This is OnNavigatedTo, a member of class FontFamilySettingsPage.

Body Source

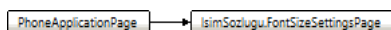
```

1: protected override void OnNavigatedTo(NavigationEventArgs e)
2: {
3:     base.OnNavigatedTo(e);
4:     SetBackgroundColor();
5: }

```

1.1.2.10 FontSizeSettingsPage Class

Class Hierarchy



C#

```
public class FontSizeSettingsPage : PhoneApplicationPage;
```

File

FontSizeSettingsPage.xaml.cs (📄 see page 60)

Description

This is class IsimSozlugu.FontSizeSettingsPage.

Methods

	Name	Description
📄	FontSizeSettingsPage (📄 see page 20)	This is FontSizeSettingsPage, a member of class FontSizeSettingsPage.

FontSizeSettingsPage Fields

	Name	Description
📄	artistId (📄 see page 20)	This is artistId, a member of class FontSizeSettingsPage.

FontSizeSettingsPage Methods

	Name	Description
☞	OnFragmentNavigation (☞ see page 20)	This is OnFragmentNavigation, a member of class FontSizeSettingsPage.
☞	OnNavigatedFrom (☞ see page 21)	This is OnNavigatedFrom, a member of class FontSizeSettingsPage.
☞	OnNavigatedTo (☞ see page 21)	This is OnNavigatedTo, a member of class FontSizeSettingsPage.

1.1.2.10.1 FontSizeSettingsPage.FontSizeSettingsPage Constructor**C#**

```
public FontSizeSettingsPage();
```

Description

This is FontSizeSettingsPage, a member of class FontSizeSettingsPage.

Body Source

```
1: public FontSizeSettingsPage()
2: {
3:     InitializeComponent();
4:
5:     lstFontSize.Items.Clear();
6:     lstFontSize.Items.Add("14");
7:     lstFontSize.Items.Add("18");
8:     lstFontSize.Items.Add("22");
9:     lstFontSize.Items.Add("26");
10:    lstFontSize.Items.Add("28");
11:    lstFontSize.Items.Add("30");
12:    lstFontSize.Items.Add("32");
13:    lstFontSize.Items.Add("34");
14:    lstFontSize.Items.Add("36");
15:    lstFontSize.Items.Add("38");
16:    lstFontSize.Items.Add("40");
17:    lstFontSize.Items.Add("42");
18:    lstFontSize.Items.Add("44");
19:    lstFontSize.Items.Add("64");
20:    lstFontSize.Items.Add("72");
21:    lstFontSize.SelectedIndex = -1;
22: }
```

1.1.2.10.2 FontSizeSettingsPage Fields**1.1.2.10.2.1 FontSizeSettingsPage.artistId Field****C#**

```
public int artistId;
```

Description

This is artistId, a member of class FontSizeSettingsPage.

1.1.2.10.3 FontSizeSettingsPage Methods**1.1.2.10.3.1 FontSizeSettingsPage.OnFragmentNavigation Method****C#**

```
protected override void OnFragmentNavigation(FragmentNavigationEventArgs e);
```


Description

This is OnFragmentNavigation, a member of class FontSizeSettingsPage.

Body Source

```
1: protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
2: {
3:     // displays "Fragment: Detail"
4:     //MessageBox.Show("Folder Id: " + e.Fragment);
5:     base.OnFragmentNavigation(e);
6: }
```

1.1.2.10.3.2 FontSizeSettingsPage.OnNavigatedFrom Method**C#**

```
protected override void OnNavigatedFrom(NavigationEventArgs e);
```

Description

This is OnNavigatedFrom, a member of class FontSizeSettingsPage.

Body Source

```
1: protected override void OnNavigatedFrom(NavigationEventArgs e)
2: {
3:     base.OnNavigatedFrom(e);
4: }
```

1.1.2.10.3.3 FontSizeSettingsPage.OnNavigatedTo Method**C#**

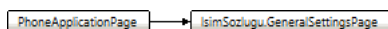
```
protected override void OnNavigatedTo(NavigationEventArgs e);
```

Description

This is OnNavigatedTo, a member of class FontSizeSettingsPage.

Body Source

```
1: protected override void OnNavigatedTo(NavigationEventArgs e)
2: {
3:     base.OnNavigatedTo(e);
4:     SetBackgroundColor();
5: }
```

1.1.2.11 GeneralSettingsPage Class**Class Hierarchy****C#**

```
public class GeneralSettingsPage : PhoneApplicationPage;
```


File

GeneralSettingsPage.xaml.cs (see page 62)



Description

This is class IsimSozlugu.GeneralSettingsPage.

Methods

	Name	Description
	GeneralSettingsPage (see page 22)	This is GeneralSettingsPage, a member of class GeneralSettingsPage.

GeneralSettingsPage Methods

	Name	Description
	OnNavigatedFrom (see page 23)	This is OnNavigatedFrom, a member of class GeneralSettingsPage.
	OnNavigatedTo (see page 23)	This is OnNavigatedTo, a member of class GeneralSettingsPage.

1.1.2.11.1 GeneralSettingsPage.GeneralSettingsPage Constructor**C#**

```
public GeneralSettingsPage();
```

Description

This is GeneralSettingsPage, a member of class GeneralSettingsPage.

Body Source

```
1: public GeneralSettingsPage()
2: {
3:     InitializeComponent();
4:     SetBackgroundColor();
5:
6:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
7:     {
8:         var appSettings = context.AppSettings.First() as AppSettings;
9:         if (appSettings.AppBackgroundColor == "BLA")
10:         {
11:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Siyah)";
12:         }
13:         if (appSettings.AppBackgroundColor == "BLU")
14:         {
15:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Mavi)";
16:         }
17:         if (appSettings.AppBackgroundColor == "BRO")
18:         {
19:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Kahverengi)";
20:         }
21:         if (appSettings.AppBackgroundColor == "RED")
22:         {
23:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Kirmizi)";
24:         }
25:         if (appSettings.AppBackgroundColor == "GRE")
26:         {
27:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Yesil)";
28:         }
29:         if (appSettings.AppBackgroundColor == "YEL")
30:         {
31:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Sari)";
32:         }
33:         if (appSettings.AppBackgroundColor == "GRA")
34:         {
35:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Gri)";
36:         }
37:         if (appSettings.AppBackgroundColor == "ORA")
38:         {
39:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Turuncu)";
40:         }
41:         if (appSettings.AppBackgroundColor == "PUR")
42:         {
43:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Mor)";
```

```
44:         }
45:
46:         lblFontFamily.Text = "Font Ailesi (Seçilen: " + appSettings.FontFamily + ")";
47:         lblFontSize.Text = "Font Boyutu (Seçilen: " + appSettings.FontSize + ")";
48:
49:     }
50: }
```

1.1.2.11.2 GeneralSettingsPage Methods

1.1.2.11.2.1 GeneralSettingsPage.OnNavigatedFrom Method

C#

```
protected override void OnNavigatedFrom(NavigationEventArgs e);
```

Description

This is OnNavigatedFrom, a member of class GeneralSettingsPage.

Body Source

```
1: protected override void OnNavigatedFrom(NavigationEventArgs e)
2: {
3:     base.OnNavigatedFrom(e);
4:     //while (NavigationService.CanGoBack)
5:     //NavigationService.RemoveBackEntry();
6:
7: }
```

1.1.2.11.2.2 GeneralSettingsPage.OnNavigatedTo Method

C#

```
protected override void OnNavigatedTo(NavigationEventArgs e);
```

Description

This is OnNavigatedTo, a member of class GeneralSettingsPage.

Body Source

```
1: protected override void OnNavigatedTo(NavigationEventArgs e)
2: {
3:     base.OnNavigatedTo(e);
4:     SetBackgroundColor();
5:     //while (NavigationService.CanGoBack)
6:     //NavigationService.RemoveBackEntry();
7:
8: }
```

1.1.2.12 LocalizedStrings Class

Provides access to string resources.

Class Hierarchy

IsimSozlugu.LocalizedStrings



C#

```
public class LocalizedStrings;
```

File

LocalizedStrings.cs (see page 67)

LocalizedStrings Properties

	Name	Description
	LocalizedResources ( see page 24)	This is LocalizedResources, a member of class LocalizedStrings.

1.1.2.12.1 LocalizedStrings Properties**1.1.2.12.1.1 LocalizedStrings.LocalizedResources Property****C#**

```
public AppResources LocalizedResources;
```


Description

This is LocalizedResources, a member of class LocalizedStrings.

1.1.2.13 MainPage Class**Class Hierarchy****C#**

```
public class MainPage : PhoneApplicationPage;
```


File

MainPage.xaml.cs ( see page 68)

Description

This is class IsimSozlugu.MainPage.

Methods

	Name	Description
	MainPage ( see page 24)	This is MainPage, a member of class MainPage.

1.1.2.13.1 MainPage.MainPage Constructor**C#**

```
public MainPage();
```

Description

This is MainPage, a member of class MainPage.

Body Source

```

1: public MainPage()
2: {
3:     InitializeComponent();
4:     ApplicationBar = new ApplicationBar();
5:
6:     ApplicationBarIconButton button1 = new ApplicationBarIconButton();
7:     button1.IconUri = new Uri("/Assets/Add.png", UriKind.Relative);
8:     button1.Text = "Isim Ekle";
9:     ApplicationBar.Buttons.Add(button1);
10:    button1.Click += new EventHandler(AddNameButton_Click);
11: }

```

```

12:     ApplicationBarItem button2 = new ApplicationBarItem();
13:     button2.IconUri = new Uri("/Assets/Search.png", UriKind.Relative);
14:     button2.Text = "Ara";
15:     ApplicationBar.Buttons.Add(button2);
16:     button2.Click += new EventHandler(SearchButton_Click);
17:
18:     ApplicationBarItem button3 = new ApplicationBarItem();
19:     button3.IconUri = new Uri("/Assets/Settings.png", UriKind.Relative);
20:     button3.Text = "Ayarlar";
21:     ApplicationBar.Buttons.Add(button3);
22:     button3.Click += new EventHandler(SettingsButton_Click);
23:
24:     ApplicationBarItem button4 = new ApplicationBarItem();
25:     button4.IconUri = new Uri("/Assets/Statistics.png", UriKind.Relative);
26:     button4.Text = "Istatistikler";
27:     ApplicationBar.Buttons.Add(button4);
28:     button4.Click += new EventHandler(StatisticsButton_Click);
29:
30:     ApplicationBarItem menuItem1 = new ApplicationBarItem();
31:     menuItem1.Text = "Hakkında";
32:     ApplicationBar.MenuItems.Add(menuItem1);
33:     menuItem1.Click += new EventHandler(AboutMenuItem_Click);
34:
35:     SetBackgroundColor();
36:
37: }

```

1.1.2.14 Male Class

Class Hierarchy

IsimSozlugu.Male

C#

```

[Table]
public class Male;

```




File

Male.cs (see page 72)

Description

This is class IsimSozlugu.Male.

Male Properties

	Name	Description
	MaleId (see page 25)	This is MaleId, a member of class Male.
	MaleMeaning (see page 26)	This is MaleMeaning, a member of class Male.
	MaleName (see page 26)	This is MaleName, a member of class Male.

1.1.2.14.1 Male Properties

1.1.2.14.1.1 Male.MaleId Property

C#

```

[Column(IsPrimaryKey = true, IsDbGenerated = true, DbType = "INT NOT NULL Identity",
CanBeNull = false)]
public int MaleId;

```

Description

This is MaleId, a member of class Male.

1.1.2.14.1.2 Male.MaleMeaning Property**C#**

```
[Column]
public string MaleMeaning;
```

Description

This is MaleMeaning, a member of class Male.

1.1.2.14.1.3 Male.MaleName Property**C#**

```
[Column]
public string MaleName;
```

Description

This is MaleName, a member of class Male.

1.1.2.15 MyUpdate Class**Class Hierarchy**
C#

```
[Table]
public class MyUpdate;
```

File

MyUpdate.cs (see page 73)

Description

This is class IsimSozlugu.MyUpdate.

MyUpdate Properties

	Name	Description
	MyUpdateId (see page 26)	This is MyUpdateId, a member of class MyUpdate.
	MyUpdateName (see page 27)	This is MyUpdateName, a member of class MyUpdate.

1.1.2.15.1 MyUpdate Properties**1.1.2.15.1.1 MyUpdate.MyUpdateId Property****C#**

```
[Column(IsPrimaryKey = true, IsDbGenerated = true, DbType = "INT NOT NULL Identity",
CanBeNull = false)]
public int MyUpdateId;
```

Description

This is MyUpdateId, a member of class MyUpdate.

1.1.2.15.1.2 MyUpdate.MyUpdateName Property

C#

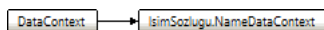
```
[Column]
public string MyUpdateName;
```

Description

This is MyUpdateName, a member of class MyUpdate.

1.1.2.16 NameDataContext Class

Class Hierarchy



C#

```
public class NameDataContext : DataContext;
```

File

NameDataContext.cs (see page 73)

Description

This is class IsimSozlugu.NameDataContext.

Methods

	Name	Description
	NameDataContext (see page 27)	This is NameDataContext, a member of class NameDataContext.

NameDataContext Fields

	Name	Description
	AllNames (see page 28)	This is AllNames, a member of class NameDataContext.
	AppSettings (see page 28)	This is AppSettings, a member of class NameDataContext.
	ConnectionString (see page 28)	This is ConnectionString, a member of class NameDataContext.
	Favourites (see page 28)	This is Favourites, a member of class NameDataContext.
	FemaleNames (see page 28)	This is FemaleNames, a member of class NameDataContext.
	MaleNames (see page 28)	This is MaleNames, a member of class NameDataContext.
	MyUpdates (see page 28)	This is MyUpdates, a member of class NameDataContext.
	UnisexNames (see page 29)	This is UnisexNames, a member of class NameDataContext.

1.1.2.16.1 NameDataContext.NameDataContext Constructor

C#

```
public NameDataContext(string connectionString);
```

Description

This is NameDataContext, a member of class NameDataContext.

Body Source

```
1: public NameDataContext(string connectionString)
2: : base(connectionString) { }
```

1.1.2.16.2 NameDataContext Fields

1.1.2.16.2.1 NameDataContext.AllNames Field

C#

```
public Table<All> AllNames;
```

Description

This is AllNames, a member of class NameDataContext.

1.1.2.16.2.2 NameDataContext.AppSettings Field

C#

```
public Table<AppSettings> AppSettings;
```

Description

This is AppSettings, a member of class NameDataContext.

1.1.2.16.2.3 NameDataContext.ConnectionString Field

C#

```
public const string ConnectionString = @"Data Source=isostore:/IsimSozlugu.sdf";
```

Description

This is ConnectionString, a member of class NameDataContext.

1.1.2.16.2.4 NameDataContext.Favourites Field

C#

```
public Table<Favourite> Favourites;
```

Description

This is Favourites, a member of class NameDataContext.

1.1.2.16.2.5 NameDataContext.FemaleNames Field

C#

```
public Table<Female> FemaleNames;
```

Description

This is FemaleNames, a member of class NameDataContext.

1.1.2.16.2.6 NameDataContext.MaleNames Field

C#

```
public Table<Male> MaleNames;
```

Description

This is MaleNames, a member of class NameDataContext.

1.1.2.16.2.7 NameDataContext.MyUpdates Field

C#

```
public Table<MyUpdate> MyUpdates;
```


Description

This is MyUpdates, a member of class NameDataContext.

1.1.2.16.2.8 NameDataContext.UnisexNames Field**C#**

```
public Table<Unisex> UnisexNames;
```

Description

This is UnisexNames, a member of class NameDataContext.

1.1.2.17 NameDetailPage Class**Class Hierarchy****C#**

```
public class NameDetailPage : PhoneApplicationPage;
```

File

NameDetailPage.xaml.cs (see page 74)

Description

This is class IsimSozlugu.NameDetailPage.

Methods

	Name	Description
	NameDetailPage (see page 29)	This is NameDetailPage, a member of class NameDetailPage.

NameDetailPage Fields

	Name	Description
	flag (see page 30)	This is flag, a member of class NameDetailPage.
	isFilled (see page 31)	This is isFilled, a member of class NameDetailPage.
	pageName (see page 31)	This is pageName, a member of class NameDetailPage.
	ratingValue (see page 31)	This is ratingValue, a member of class NameDetailPage.

NameDetailPage Methods

	Name	Description
	OnFragmentNavigation (see page 31)	This is OnFragmentNavigation, a member of class NameDetailPage.
	OnNavigatedFrom (see page 32)	This is OnNavigatedFrom, a member of class NameDetailPage.
	OnNavigatedTo (see page 32)	This is OnNavigatedTo, a member of class NameDetailPage.

1.1.2.17.1 NameDetailPage.NameDetailPage Constructor**C#**

```
public NameDetailPage();
```

Description

This is NameDetailPage, a member of class NameDetailPage.

Body Source

```
1: public NameDetailPage()
2: {
3:     InitializeComponent();
4:
5:     ApplicationBar = new ApplicationBar();
6:
7:     ApplicationBarIconButton button1 = new ApplicationBarIconButton();
8:     button1.IconUri = new Uri("/Assets/Save.png", UriKind.Relative);
9:     button1.Text = "Kaydet";
10:    ApplicationBar.Buttons.Add(button1);
11:    button1.Click += new EventHandler(SaveButton_Click);
12:
13:    ApplicationBarIconButton button2 = new ApplicationBarIconButton();
14:    button2.IconUri = new Uri("/Assets/SendWithMail.png", UriKind.Relative);
15:    button2.Text = "E-Mail ile Gönder";
16:    ApplicationBar.Buttons.Add(button2);
17:    button2.Click += new EventHandler(SendMailButton_Click);
18:
19:    ApplicationBarIconButton button3 = new ApplicationBarIconButton();
20:    button3.IconUri = new Uri("/Assets/SendWithSMS.png", UriKind.Relative);
21:    button3.Text = "SMS ile Gönder";
22:    ApplicationBar.Buttons.Add(button3);
23:    button3.Click += new EventHandler(SendSMSButton_Click);
24:
25:    ApplicationBarIconButton button4 = new ApplicationBarIconButton();
26:    button4.IconUri = new Uri("/Assets/Share.png", UriKind.Relative);
27:    button4.Text = "Paylas";
28:    ApplicationBar.Buttons.Add(button4);
29:    button4.Click += new EventHandler(ShareNameButton_Click);
30:
31:    isFilled = false;
32:
33:    ApplicationBarMenuItem menuItem1 = new ApplicationBarMenuItem();
34:    menuItem1.Text = "Sil";
35:    ApplicationBar.MenuItems.Add(menuItem1);
36:    menuItem1.Click += new EventHandler>DeleteNameMenuItem_Click);
37:
38:    List<string> genderList = new List<string>();
39:    genderList.Add("Lütfen seçiniz");
40:    genderList.Add("Erkek");
41:    genderList.Add("Kadin");
42:    genderList.Add("Erkek-Kadin");
43:    lpGender.ItemsSource = genderList;
44:    lpGender.SelectedIndex = 0;
45:
46:    SetBackgroundColor();
47: }
```

1.1.2.17.2 NameDetailPage Fields

1.1.2.17.2.1 NameDetailPage.flag Field

C#

```
public bool flag;
```

Description

This is flag, a member of class NameDetailPage.

1.1.2.17.2 NameDetailPage.isFilled Field

C#

```
public bool isFilled;
```

Description

This is isFilled, a member of class NameDetailPage.

1.1.2.17.3 NameDetailPage.pageName Field

C#

```
public string pageName;
```

Description

This is pageName, a member of class NameDetailPage.

1.1.2.17.4 NameDetailPage.ratingValue Field

C#

```
public double ratingValue = 0;
```

Description

This is ratingValue, a member of class NameDetailPage.

1.1.2.17.3 NameDetailPage Methods

1.1.2.17.3.1 NameDetailPage.OnFragmentNavigation Method

C#

```
protected override void OnFragmentNavigation(FragmentNavigationEventArgs e);
```

Description

This is OnFragmentNavigation, a member of class NameDetailPage.

Body Source

```
1: protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
2: {
3:     // displays "Fragment: Detail"
4:     //MessageBox.Show("Folder Id: " + e.Fragment);
5:     base.OnFragmentNavigation(e);
6:     string fragmentName = e.Fragment.ToString();
7:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
8:     {
9:         var name = context.AllNames.Where(j =>
j.AllName.Equals(fragmentName)).SingleOrDefault() as All;
10:         allName = name;
11:         for (int i = 0; i < lpGender.Items.Count; i++)
12:         {
13:             if (lpGender.Items[i].ToString() == name.AllGender)
14:             {
15:                 lpGender.SelectedIndex = i;
16:                 break;
17:             }
18:         }
19:         txtName.Text = name.AllName;
20:         txtName.IsEnabled = false;
21:         txtMeaning.Text = name.AllMeaning;
22:     }
```

```

23:         var favourite = context.Favourites.Where(j =>
j.FavouriteName.Equals(fragmentName)).SingleOrDefault() as Favourite;
24:         if (favourite != null)
25:         {
26:             ApplicationBarItem menuItem4 = new ApplicationBarItem();
27:             menuItem4.Text = "Favorilerimden Çıkar";
28:             ApplicationBar.MenuItems.Add(menuItem4);
29:             menuItem4.Click += new EventHandler(RemoveFavouritesMenuItem_Click);
30:         }
31:         else
32:         {
33:             ApplicationBarItem menuItem3 = new ApplicationBarItem();
34:             menuItem3.Text = "Favorilerime Ekle";
35:             ApplicationBar.MenuItems.Add(menuItem3);
36:             menuItem3.Click += new EventHandler(AddFavouritesMenuItem_Click);
37:         }
38:
39:         var myUpdate = context.MyUpdates.Where(j =>
j.MyUpdateName.Equals(fragmentName)).SingleOrDefault() as MyUpdate; ;
40:         if (myUpdate != null)
41:         {
42:             ApplicationBarItem menuItem2 = new ApplicationBarItem();
43:             menuItem2.Text = "Sisteme Eklenmesi İçin Gönder";
44:             ApplicationBar.MenuItems.Add(menuItem2);
45:             menuItem2.Click += new EventHandler(SaveAndSendMenuItem_Click);
46:         }
47:
48:
49:
50:         isFilled = true;
51:         pvName.SelectedIndex = 0;
52:     }
53: }

```

1.1.2.17.3.2 NameDetailPage.OnNavigatedFrom Method

C#

```
protected override void OnNavigatedFrom(NavigationEventArgs e);
```

Description

This is OnNavigatedFrom, a member of class NameDetailPage.

Body Source

```

1: protected override void OnNavigatedFrom(NavigationEventArgs e)
2: {
3:     base.OnNavigatedFrom(e);
4:     //while (NavigationService.CanGoBack)
5:     //NavigationService.RemoveBackEntry();
6:
7: }

```

1.1.2.17.3.3 NameDetailPage.OnNavigatedTo Method

C#

```
protected override void OnNavigatedTo(NavigationEventArgs e);
```

Description

This is OnNavigatedTo, a member of class NameDetailPage.

Body Source

```

1: protected override void OnNavigatedTo(NavigationEventArgs e)
2: {
3:     base.OnNavigatedTo(e);
4:
5:     var lastPage = NavigationService.BackStack.FirstOrDefault();

```

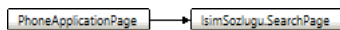
```

6:     pageName = lastPage.Source.ToString();
7:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
8:     {
9:         var appSettings = context.AppSettings.First();
10:
11:         FontFamily temp = new FontFamily(appSettings.FontFamily);
12:         double fontsize = double.Parse(appSettings.FontSize);
13:         txtName.FontFamily = temp;
14:         txtName.FontSize = fontsize;
15:         txtMeaning.FontFamily = temp;
16:         txtMeaning.FontSize = fontsize;
17:         lpGender.FontFamily = temp;
18:         lpGender.FontSize = fontsize;
19:     }
20:
21:     txtName.IsEnabled = true;
22:     isFilled = false;
23:     //SetBackgroundColor();
24:     //while (NavigationService.CanGoBack)
25:     //NavigationService.RemoveBackEntry();
26:
27: }

```

1.1.2.18 SearchPage Class

Class Hierarchy



C#

```
public class SearchPage : PhoneApplicationPage;
```

File

SearchPage.xaml.cs (see page 84)

Description

This is class IsimSozlugu.SearchPage.

Methods

	Name	Description
	SearchPage (see page 33)	This is SearchPage, a member of class SearchPage.

SearchPage Methods

	Name	Description
	OnNavigatedFrom (see page 34)	This is OnNavigatedFrom, a member of class SearchPage.
	OnNavigatedTo (see page 34)	This is OnNavigatedTo, a member of class SearchPage.

1.1.2.18.1 SearchPage.SearchPage Constructor

C#

```
public SearchPage();
```

Description

This is SearchPage, a member of class SearchPage.

Body Source

```

1: public SearchPage()
2: {
3:     InitializeComponent();

```

```

4:      SetBackgroundColor();
5:
6:      txtSearchResult.Text = "Arama Sonuçlari";
7:      //txtSearchResult.Text = AppResources.SearchResults;
8:      //lblSearch.Text = AppResources.Search;
9:      //btnSearch.Content = AppResources.Search;
10:     //lstSearch.SelectedIndex = -1;
11: }

```

1.1.2.18.2 SearchPage Methods

1.1.2.18.2.1 SearchPage.OnNavigatedFrom Method

C#

```
protected override void OnNavigatedFrom(NavigationEventArgs e);
```

Description

This is OnNavigatedFrom, a member of class SearchPage.

Body Source

```

1: protected override void OnNavigatedFrom(NavigationEventArgs e)
2: {
3:     base.OnNavigatedFrom(e);
4: }

```

1.1.2.18.2.2 SearchPage.OnNavigatedTo Method

C#

```
protected override void OnNavigatedTo(NavigationEventArgs e);
```

Description

This is OnNavigatedTo, a member of class SearchPage.

Body Source

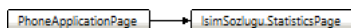
```

1: protected override void OnNavigatedTo(NavigationEventArgs e)
2: {
3:     base.OnNavigatedTo(e);
4: }

```

1.1.2.19 StatisticsPage Class

Class Hierarchy



C#

```
public class StatisticsPage : PhoneApplicationPage;
```

File

StatisticsPage.xaml.cs (see page 87)

Description

This is class IsimSozlugu.StatisticsPage.

Methods

	Name	Description
	StatisticsPage (see page 35)	This is StatisticsPage, a member of class StatisticsPage.

StatisticsPage Methods

	Name	Description
🔗	OnNavigatedFrom (🔗 see page 35)	This is OnNavigatedFrom, a member of class StatisticsPage.
🔗	OnNavigatedTo (🔗 see page 35)	This is OnNavigatedTo, a member of class StatisticsPage.

1.1.2.19.1 StatisticsPage.StatisticsPage Constructor**C#**

```
public StatisticsPage();
```

Description

This is StatisticsPage, a member of class StatisticsPage.

Body Source

```
1: public StatisticsPage()
2: {
3:     InitializeComponent();
4:     SetBackgroundColor();
5:     SetStatistic();
6: }
```

1.1.2.19.2 StatisticsPage Methods**1.1.2.19.2.1 StatisticsPage.OnNavigatedFrom Method****C#**

```
protected override void OnNavigatedFrom(NavigationEventArgs e);
```

Description

This is OnNavigatedFrom, a member of class StatisticsPage.

Body Source

```
1: protected override void OnNavigatedFrom(NavigationEventArgs e)
2: {
3:     base.OnNavigatedFrom(e);
4:     //while (NavigationService.CanGoBack)
5:     //NavigationService.RemoveBackEntry();
6:
7: }
```

1.1.2.19.2.2 StatisticsPage.OnNavigatedTo Method**C#**

```
protected override void OnNavigatedTo(NavigationEventArgs e);
```

Description

This is OnNavigatedTo, a member of class StatisticsPage.

Body Source

```
1: protected override void OnNavigatedTo(NavigationEventArgs e)
2: {
3:     base.OnNavigatedTo(e);
4: }
```

1.1.2.20 Unisex Class

Class Hierarchy

IsimSozlugu.Unisex

C#

```
[Table]
public class Unisex;
```




File

Unisex.cs (see page 90)

Description

This is class IsimSozlugu.Unisex.

Unisex Properties

	Name	Description
	UnisexId (see page 36)	This is UnisexId, a member of class Unisex.
	UnisexMeaning (see page 36)	This is UnisexMeaning, a member of class Unisex.
	UnisexName (see page 36)	This is UnisexName, a member of class Unisex.

1.1.2.20.1 Unisex Properties

1.1.2.20.1.1 Unisex.UnisexId Property

C#

```
[Column(IsPrimaryKey = true, IsDbGenerated = true, DbType = "INT NOT NULL Identity",
CanBeNull = false)]
public int UnisexId;
```

Description

This is UnisexId, a member of class Unisex.

1.1.2.20.1.2 Unisex.UnisexMeaning Property

C#

```
[Column]
public string UnisexMeaning;
```

Description

This is UnisexMeaning, a member of class Unisex.

1.1.2.20.1.3 Unisex.UnisexName Property

C#

```
[Column]
public string UnisexName;
```

Description

This is UnisexName, a member of class Unisex.

1.2 Files

The following table lists files in this documentation.

Files

Name	Description
AboutPage.xaml.cs (see page 37)	This is file AboutPage.xaml.cs.
All.cs (see page 40)	This is file All.cs.
AlphaKeyGroup.cs (see page 41)	This is file AlphaKeyGroup.cs.
App.xaml.cs (see page 42)	This is file App.xaml.cs.
AppResources.Designer.cs (see page 49)	This code was generated by a tool. Runtime Version:4.0.30319.17626 Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.
AppSettings.cs (see page 52)	This is file AppSettings.cs.
AssemblyInfo.cs (see page 52)	This is file AssemblyInfo.cs.
BackgroundColorSettingsPage.xaml.cs (see page 53)	This is file BackgroundColorSettingsPage.xaml.cs.
Favourite.cs (see page 56)	This is file Favourite.cs.
Female.cs (see page 57)	This is file Female.cs.
FontFamilySettingsPage.xaml.cs (see page 57)	This is file FontFamilySettingsPage.xaml.cs.
FontSizeSettingsPage.xaml.cs (see page 60)	This is file FontSizeSettingsPage.xaml.cs.
GeneralSettingsPage.xaml.cs (see page 62)	This is file GeneralSettingsPage.xaml.cs.
IsimSozlugu.csproj (see page 67)	This is file IsimSozlugu.csproj.
IsimSozlugu.sln (see page 67)	This is file IsimSozlugu.sln.
LocalizedStrings.cs (see page 67)	This is file LocalizedStrings.cs.
MainPage.xaml.cs (see page 68)	This is file MainPage.xaml.cs.
Male.cs (see page 72)	This is file Male.cs.
MyUpdate.cs (see page 73)	This is file MyUpdate.cs.
NameDataContext.cs (see page 73)	This is file NameDataContext.cs.
NameDetailPage.xaml.cs (see page 74)	This is file NameDetailPage.xaml.cs.
SearchPage.xaml.cs (see page 84)	This is file SearchPage.xaml.cs.
StatisticsPage.xaml.cs (see page 87)	This is file StatisticsPage.xaml.cs.
Unisex.cs (see page 90)	This is file Unisex.cs.

1.2.1 AboutPage.xaml.cs

This is file AboutPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;
5: using System.Net;
6: using System.Text;

```

```

7: using System.Windows;
8: using System.Windows.Controls;
9: using System.Windows.Controls.Primitives;
10: using System.Windows.Input;
11: using System.Windows.Media;
12: using System.Windows.Media.Imaging;
13: using System.Windows.Navigation;
14: using Microsoft.Phone.Tasks;
15: using IsimSozlugu.Resources;
16: using Microsoft.Phone.Controls;
17: using Microsoft.Phone.Shell;
18:
19: namespace IsimSozlugu
20: {
21:     public partial class AboutPage : PhoneApplicationPage
22:     {
23:         public AboutPage()
24:         {
25:             InitializeComponent();
26:
27:             SetBackgroundColor();
28:
29:             ApplicationBar = new ApplicationBar();
30:
31:             ApplicationBarIconButton button2 = new ApplicationBarIconButton();
32:             button2.IconUri = new Uri("/Assets/SendWithMail.png", UriKind.Relative);
33:             button2.Text = "İletisime Geç";
34:             ApplicationBar.Buttons.Add(button2);
35:             button2.Click += new EventHandler(SendMailButton_Click);
36:
37:             ApplicationBarIconButton button3 = new ApplicationBarIconButton();
38:             button3.IconUri = new Uri("/Assets/Rate.png", UriKind.Relative);
39:             button3.Text = "Puanla";
40:             ApplicationBar.Buttons.Add(button3);
41:             button3.Click += new EventHandler(RateButton_Click);
42:
43:             //lblAboutTheApp.Text = AppResources.AboutTheApp;
44:             //txtAbout2.Text = AppResources.AboutTheAppText;
45:             //var paragraph = new Paragraph();
46:             //paragraph.Inlines.Add(AppResources.AboutTheAppText);
47:             //txtAbout.Blocks.Add(paragraph);
48:             txtAbout.Text = "Herkes kendi isminin ne anlama geldigini merak eder. Bu
uygulama ile sadece kendi" +
49:                 " isminizin degil ayni zamanda pek çok ismin de anlamini
ögrenebileceksiniz." +
50:                 " Yeni baba veya anne olacak insanların (benim gibi) aklinda cocuguna
ne isim koyacagi gibi" +
51:                 " bir soru olmamasi için (yani aslinda biraz da kendim için) bu
uygulamayi gelistirdim." +
52:                 " Ancak bu uygulamayi hep beraber gelistirebiliriz. Program
vasitasiyla bana göndereceginiz " +
53:                 " kelimeler incelendikten sonra bu havuza atilacak ve herkesin
yararlanabilecegi güzel bir " +
54:                 " kaynak ortaya çıkacak. Simdiden iyi günler.";
55:             //txtAbout.IsEnabled = false;
56:             txtAbout.IsReadOnly = true;
57:             //this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
58:         }
59:
60:         private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
61:         {
62:             if (this.NavigationService.CanGoBack)
63:             {
64:                 this.NavigationService.Navigate(new Uri("/MainPage.xaml",
UriKind.Relative));
65:             }
66:         }
67:

```

```

68:         private void SendMailButton_Click(object sender, EventArgs e)
69:         {
70:             // burada birden fazla e-posta hesabi varsa birini seçmesi söyleniyor
71:             //EmailAddressChooserTask emailAddressChooserTask;
72:             //emailAddressChooserTask = new EmailAddressChooserTask();
73:             //emailAddressChooserTask.Completed += new
EventHandler<EmailResult>(emailAddressChooserTask_Completed);
74:             //emailAddressChooserTask.Show();
75:             StringBuilder sb = new StringBuilder();
76:             EmailComposeTask emailComposeTask = new EmailComposeTask();
77:
78:
79:             sb.AppendLine();
80:             sb.AppendLine();
81:             sb.AppendLine("İsim Sözlüğü Uygulaması İle Gönderildi");
82:
83:             emailComposeTask.Subject = "İsim Sözlüğü Uygulaması Hakkında";
84:             emailComposeTask.Body = sb.ToString();
85:             emailComposeTask.To = "coderserdar@outlook.com";
86:             emailComposeTask.Cc = "";
87:             emailComposeTask.Bcc = "";
88:
89:             emailComposeTask.Show();
90:             //MessageBox.Show(AppResources.SuccessfulSendWithMail);
91:         }
92:
93:         private void RateButton_Click(object sender, EventArgs e)
94:         {
95:             MarketplaceReviewTask marketplaceReviewTask = new MarketplaceReviewTask();
96:             marketplaceReviewTask.Show();
97:         }
98:
99:         private void SetBackgroundColor()
100:        {
101:            AppSettings appSettings = new AppSettings();
102:            using (var context = new NameDataContext(NameDataContext.ConnectionString))
103:            {
104:                appSettings = context.AppSettings.First() as AppSettings;
105:            }
106:
107:            if (appSettings.AppBackgroundImage != null)
108:            {
109:                MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
110:                BitmapImage image = new BitmapImage();
111:                image.SetSource(stream);
112:                ImageBrush ib = new ImageBrush();
113:                ib.ImageSource = image;
114:                this.LayoutRoot.Background = ib;
115:            }
116:            else
117:            {
118:                switch (appSettings.AppBackgroundColor)
119:                {
120:                    case "BLA":
121:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
122:                        break;
123:                    case "BLU":
124:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
125:                        break;
126:                    case "BRO":
127:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
128:                        break;
129:                    case "RED":
130:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
131:                        break;
132:                    case "GRE":
133:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
134:                        break;
135:                    case "GRA":

```

```

136:             this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
137:             break;
138:         case "YEL":
139:             this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
140:             break;
141:         case "ORA":
142:             this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
143:             break;
144:         case "PUR":
145:             this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
146:             break;
147:         default:
148:             this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
149:             break;
150:     }
151: }
152: }
153: }
154: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.2 All.cs

This is file All.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu
10: {
11:     [Table]
12:     public class All
13:     {
14:         [Column(IsPrimaryKey = true,
15:             IsDbGenerated = true,
16:             DbType = "INT NOT NULL Identity",
17:             CanBeNull = false)]
18:         public int AllId { get; set; }
19:
20:         [Column]
21:         public string AllName { get; set; }
22:
23:         [Column]
24:         public string AllMeaning { get; set; }
25:
26:         [Column]
27:         public string AllNameMeaning { get; set; }
28:
29:         [Column]
30:         public string AllGender { get; set; }
31:     }

```

```
32: }
```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.3 AlphaKeyGroup.cs

This is file AlphaKeyGroup.cs.

Body Source

```
1: ?using System.Collections.Generic;
2: using System.Globalization;
3: using Microsoft.Phone.Globalization;
4:
5: namespace IsimSozlugu
6: {
7:     public class AlphaKeyGroup<T> : List<T>
8:     {
9:         /// <summary>
10:         /// The delegate that is used to get the key information.
11:         /// </summary>
12:         /// <param name="item">An object of type T</param>
13:         /// <returns>The key value to use for this object</returns>
14:         public delegate string GetKeyDelegate(T item);
15:
16:         /// <summary>
17:         /// The Key of this group.
18:         /// </summary>
19:         public string Key { get; private set; }
20:
21:         /// <summary>
22:         /// Public constructor.
23:         /// </summary>
24:         /// <param name="key">The key for this group.</param>
25:         public AlphaKeyGroup(string key)
26:         {
27:             Key = key;
28:         }
29:
30:         /// <summary>
31:         /// Create a list of AlphaGroup<T> with keys set by a SortedLocaleGrouping.
32:         /// </summary>
33:         /// <param name="slg">The </param>
34:         /// <returns>The items source for a LongListSelector</returns>
35:         private static List<AlphaKeyGroup<T>> CreateGroups(SortedLocaleGrouping slg)
36:         {
37:             List<AlphaKeyGroup<T>> list = new List<AlphaKeyGroup<T>>();
38:
39:             foreach (string key in slg.GroupDisplayNames)
40:             {
41:                 list.Add(new AlphaKeyGroup<T>(key));
42:             }
43:
44:             return list;
45:         }
46:
47:         /// <summary>
48:         /// Create a list of AlphaGroup<T> with keys set by a SortedLocaleGrouping.
49:         /// </summary>
50:         /// <param name="items">The items to place in the groups.</param>
51:         /// <param name="ci">The CultureInfo to group and sort by.</param>
52:         /// <param name="getKey">A delegate to get the key from an item.</param>
```

```

53:         /// <param name="sort">Will sort the data if true.</param>
54:         /// <returns>An items source for a LongListSelector</returns>
55:         public static List<AlphaKeyGroup<T>> CreateGroups(IEnumerable<T> items,
CultureInfo ci, GetKeyDelegate getKey, bool sort)
56:         {
57:             SortedLocaleGrouping slg = new SortedLocaleGrouping(ci);
58:             List<AlphaKeyGroup<T>> list = CreateGroups(slg);
59:
60:             foreach (T item in items)
61:             {
62:                 int index = 0;
63:                 if (slg.SupportsPhonetics)
64:                 {
65:                     //check if your database has yomi string for item
66:                     //if it does not, then do you want to generate Yomi or ask the user
for this item.
67:                     //index = slg.GetGroupIndex(getKey(Yomiof(item)));
68:                 }
69:                 else
70:                 {
71:                     index = slg.GetGroupIndex(getKey(item));
72:                 }
73:                 if (index >= 0 && index < list.Count)
74:                 {
75:                     list[index].Add(item);
76:                 }
77:             }
78:
79:             if (sort)
80:             {
81:                 foreach (AlphaKeyGroup<T> group in list)
82:                 {
83:                     group.Sort((c0, c1) => { return ci.CompareInfo.Compare(getKey(c0),
getKey(c1)); });
84:                 }
85:             }
86:
87:             return list;
88:         }
89:     }
90: }
91: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.4 App.xaml.cs

This is file App.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Diagnostics;
3: using System.Globalization;
4: using System.Linq;
5: using System.Resources;
6: using System.Threading;
7: using System.IO;
8: using System.Windows;
9: using System.Windows.Markup;
10: using System.Windows.Navigation;
11: using Microsoft.Phone.Controls;

```

```

12: using Microsoft.Phone.Shell;
13: using IsimSozlugu.Resources;
14: using System.Collections.Generic;
15: using Microsoft.Phone.Marketplace;
16:
17: namespace IsimSozlugu
18: {
19:     public partial class App : Application
20:     {
21:         /// <summary>
22:         /// Provides easy access to the root frame of the Phone Application.
23:         /// </summary>
24:         /// <returns>The root frame of the Phone Application.</returns>
25:         public static PhoneApplicationFrame RootFrame { get; private set; }
26:
27:         /// <summary>
28:         /// Constructor for the Application object.
29:         /// </summary>
30:         public App()
31:         {
32:             // Global handler for uncaught exceptions.
33:             UnhandledException += Application_UnhandledException;
34:
35:             // ayarlardan temasi açık renk bile olsa
36:             // kapali gibi çalışmasını sağlayacak bir nuget paketi yüklendi
37:             // bu sorunu gideriyor
38:             ThemeManager.ToDarkTheme();
39:
40:             // Standard XAML initialization
41:             InitializeComponent();
42:
43:             // Phone-specific initialization
44:             InitializePhoneApplication();
45:
46:             // Language display initialization
47:             InitializeLanguage();
48:
49:             // Show graphics profiling information while debugging.
50:             if (Debugger.IsAttached)
51:             {
52:                 // Display the current frame rate counters.
53:                 Application.Current.Host.Settings.EnableFrameRateCounter = true;
54:
55:                 // Show the areas of the app that are being redrawn in each frame.
56:                 //Application.Current.Host.Settings.EnableRedrawRegions = true;
57:
58:                 // Enable non-production analysis visualization mode,
59:                 // which shows areas of a page that are handed off to GPU with a
60:                 //Application.Current.Host.Settings.EnableCacheVisualization = true;
61:                 colored overlay.
62:                 // Prevent the screen from turning off while under the debugger by
63:                 // the application's idle detection.
64:                 // Caution:- Use this under debug mode only. Application that disables
65:                 // and consume battery power when the user is not using the phone.
66:                 PhoneApplicationService.Current.UserIdleDetectionMode =
67:                 IdleDetectionMode.Disabled;
68:             }
69:         }
70:
71:         // Code to execute when the application is launching (eg, from Start)
72:         // This code will not execute when the application is reactivated
73:         private void Application_Launching(object sender, LaunchingEventArgs e)
74:         {
75:             using (var context = new NameDataContext(NameDataContext.ConnectionString))
76:             {

```

```

77:         if (!context.DatabaseExists())
78:         {
79:             context.CreateDatabase();
80:             DilAyariOlustur(context);
81:
82:             // burada metin belgeleri okunuyor ve veritabanindaki tablolara
ekleme yapiliyor
83:             var male = ReadFile(@"Sources\ErkekAdlari.txt");
84:             var maleLines = male.Split('\n').ToList() as List<string>;
85:             for (int i = 1; i < maleLines.Count; i++)
86:             {
87:                 var information = maleLines[i].Split('_').ToList() as
List<string>;
88:                 Male maleName = new Male();
89:                 maleName.MaleName =
information[0].ToUpper().TrimEnd().TrimStart();
90:                 maleName.MaleMeaning = information[1].TrimEnd().TrimStart();
91:
92:                 context.MaleNames.InsertOnSubmit(maleName);
93:             }
94:
95:             var female = ReadFile(@"Sources\KizAdlari.txt");
96:             var femaleLines = female.Split('\n').ToList() as List<string>;
97:             for (int i = 1; i < femaleLines.Count; i++)
98:             {
99:                 var information = femaleLines[i].Split('_').ToList() as
List<string>;
100:                 Female femaleName = new Female();
101:                 femaleName.FemaleName =
information[0].ToUpper().TrimEnd().TrimStart();
102:                 femaleName.FemaleMeaning =
information[1].TrimEnd().TrimStart();
103:
104:                 context.FemaleNames.InsertOnSubmit(femaleName);
105:             }
106:
107:             context.SubmitChanges();
108:
109:             var maleNamesTemp = context.MaleNames.ToList() as List<Male>;
110:             var femaleNamesTemp = context.FemaleNames.ToList() as List<Female>;
111:
112:             for (int i = 0; i < maleNamesTemp.Count; i++)
113:             {
114:                 for (int k = 0; k < femaleNamesTemp.Count; k++)
115:                 {
116:                     if (maleNamesTemp[i].MaleName ==
femaleNamesTemp[k].FemaleName)
117:                     {
118:                         Unisex unisex = new Unisex();
119:                         unisex.UnisexName = maleNamesTemp[i].MaleName;
120:                         unisex.UnisexMeaning = maleNamesTemp[i].MaleMeaning;
121:                         context.UnisexNames.InsertOnSubmit(unisex);
122:                         var maleNameTemp = context.MaleNames.Where(j =>
j.MaleId.Equals(maleNamesTemp[i].MaleId)).SingleOrDefault() as Male;
123:                         context.MaleNames.DeleteOnSubmit(maleNameTemp);
124:                         var femaleNameTemp = context.FemaleNames.Where(j =>
j.FemaleId.Equals(femaleNamesTemp[k].FemaleId)).SingleOrDefault() as Female;
125:                         context.FemaleNames.DeleteOnSubmit(femaleNameTemp);
126:                         break;
127:                     }
128:                 }
129:             }
130:
131:             context.SubmitChanges();
132:
133:             var males = context.MaleNames.ToList() as List<Male>;
134:             for (int i = 0; i < males.Count; i++)
135:             {
136:                 All allName = new All();

```



```

137:         allName.AllName = males[i].MaleName;
138:         allName.AllMeaning = males[i].MaleMeaning;
139:         allName.AllNameMeaning = males[i].MaleName + " " +
males[i].MaleMeaning;
140:         allName.AllGender = "Erkek";
141:
142:         context.AllNames.InsertOnSubmit(allName);
143:     }
144:
145:     var females = context.FemaleNames.ToList() as List<Female>;
146:     for (int i = 0; i < females.Count; i++)
147:     {
148:         All allName = new All();
149:         allName.AllName = females[i].FemaleName;
150:         allName.AllMeaning = females[i].FemaleMeaning;
151:         allName.AllNameMeaning = females[i].FemaleName + " " +
females[i].FemaleMeaning;
152:         allName.AllGender = "Kadin";
153:
154:         context.AllNames.InsertOnSubmit(allName);
155:     }
156:
157:     var unisexes = context.UnisexNames.ToList() as List<Unisex>;
158:     for (int i = 0; i < unisexes.Count; i++)
159:     {
160:         All allName = new All();
161:         allName.AllName = unisexes[i].UnisexName;
162:         allName.AllMeaning = unisexes[i].UnisexMeaning;
163:         allName.AllNameMeaning = unisexes[i].UnisexName + " " +
unisexes[i].UnisexMeaning;
164:         allName.AllGender = "Erkek-Kadin";
165:
166:         context.AllNames.InsertOnSubmit(allName);
167:     }
168:
169:     context.SubmitChanges();
170: }
171: else
172: {
173:     using (var context2 = new
NameDataContext(NameDataContext.ConnectionString))
174:     {
175:
176:         AppSettings lang =
177:             context2.AppSettings.First() as AppSettings;
178:         string culture = "";
179:         switch (lang.AppLangName)
180:         {
181:             case "TR":
182:                 culture = "tr";
183:                 break;
184:             case "EN":
185:                 culture = "en";
186:                 break;
187:             case "DE":
188:                 culture = "de";
189:                 break;
190:             case "ES":
191:                 culture = "es";
192:                 break;
193:             case "FR":
194:                 culture = "fr";
195:                 break;
196:             case "IT":
197:                 culture = "it";
198:                 break;
199:             case "AR":
200:                 culture = "ar";
201:                 break;

```

```

202:         case "FA":
203:             culture = "fa-IR";
204:             break;
205:         case "ZH":
206:             culture = "zh";
207:             break;
208:         case "PT":
209:             culture = "pt";
210:             break;
211:         case "RU":
212:             culture = "ru";
213:             break;
214:         case "JA":
215:             culture = "ja";
216:             break;
217:         case "SA":
218:             culture = "sa";
219:             break;
220:         case "TH":
221:             culture = "th";
222:             break;
223:         default:
224:             culture = "tr-TR";
225:             break;
226:     }
227:     CultureInfo newCulture = new CultureInfo(culture);
228:     Thread.CurrentThread.CurrentCulture = newCulture;
229:     Thread.CurrentThread.CurrentUICulture = newCulture;
230: }
231: }
232:
233:     // kullanıcının programla ilgili bilgilendirici notları kendi dilinde
görebilmesi için burada ekliyoruz.
234:     }
235: }
236:
237: // Code to execute when the application is activated (brought to foreground)
238: // This code will not execute when the application is first launched
239: private void Application_Activated(object sender, ActivatedEventArgs e)
240: {
241: }
242:
243: // Code to execute when the application is deactivated (sent to background)
244: // This code will not execute when the application is closing
245: private void Application_Deactivated(object sender, DeactivatedEventArgs e)
246: {
247: }
248:
249: // Code to execute when the application is closing (eg, user hit Back)
250: // This code will not execute when the application is deactivated
251: private void Application_Closing(object sender, ClosingEventArgs e)
252: {
253: }
254:
255: // Code to execute if a navigation fails
256: private void RootFrame_NavigationFailed(object sender,
NavigationFailedEventArgs e)
257: {
258:     if (Debugger.IsAttached)
259:     {
260:         // A navigation has failed; break into the debugger
261:         Debugger.Break();
262:     }
263: }
264:
265: // Code to execute on Unhandled Exceptions
266: private void Application_UnhandledException(object sender,
ApplicationUnhandledExceptionEventArgs e)
267: {

```

```

268:         if (Debugger.IsAttached)
269:         {
270:             // An unhandled exception has occurred; break into the debugger
271:             Debugger.Break();
272:         }
273:     }
274:
275:     #region Phone application initialization
276:
277:     // Avoid double-initialization
278:     private bool phoneApplicationInitialized = false;
279:
280:     // Do not add any additional code to this method
281:     private void InitializePhoneApplication()
282:     {
283:         if (phoneApplicationInitialized)
284:             return;
285:
286:         // Create the frame but don't set it as RootVisual yet; this allows the
287:         // screen to remain active until the application is ready to render.
288:         RootFrame = new PhoneApplicationFrame();
289:         RootFrame.Navigated += CompleteInitializePhoneApplication;
290:
291:         // Handle navigation failures
292:         RootFrame.NavigationFailed += RootFrame_NavigationFailed;
293:
294:         // Handle reset requests for clearing the backstack
295:         RootFrame.Navigated += CheckForResetNavigation;
296:
297:         // Ensure we don't initialize again
298:         phoneApplicationInitialized = true;
299:     }
300:
301:     // Do not add any additional code to this method
302:     private void CompleteInitializePhoneApplication(object sender,
303:     NavigationEventArgs e)
304:     {
305:         // Set the root visual to allow the application to render
306:         if (RootVisual != RootFrame)
307:             RootVisual = RootFrame;
308:
309:         // Remove this handler since it is no longer needed
310:         RootFrame.Navigated -= CompleteInitializePhoneApplication;
311:     }
312:
313:     private void CheckForResetNavigation(object sender, NavigationEventArgs e)
314:     {
315:         // If the app has received a 'reset' navigation, then we need to check
316:         // on the next navigation to see if the page stack should be reset
317:         if (e.NavigationMode == NavigationMode.Reset)
318:             RootFrame.Navigated += ClearBackStackAfterReset;
319:     }
320:
321:     private void ClearBackStackAfterReset(object sender, NavigationEventArgs e)
322:     {
323:         // Unregister the event so it doesn't get called again
324:         RootFrame.Navigated -= ClearBackStackAfterReset;
325:
326:         // Only clear the stack for 'new' (forward) and 'refresh' navigations
327:         if (e.NavigationMode != NavigationMode.New && e.NavigationMode !=
328:         NavigationMode.Refresh)
329:             return;
330:
331:         // For UI consistency, clear the entire page stack
332:         while (RootFrame.RemoveBackEntry() != null)
333:         {
334:             ; // do nothing
335:         }

```

```

334:         }
335:
336:         #endregion
337:
338:         // Initialize the app's font and flow direction as defined in its localized
resource strings.
339:         //
340:         // To ensure that the font of your application is aligned with its supported
languages and that the
341:         // FlowDirection for each of those languages follows its traditional
direction, ResourceLanguage
342:         // and ResourceFlowDirection should be initialized in each resx file to match
these values with that
343:         // file's culture. For example:
344:         //
345:         // AppResources.es-ES.resx
346:         //     ResourceLanguage's value should be "es-ES"
347:         //     ResourceFlowDirection's value should be "LeftToRight"
348:         //
349:         // AppResources.ar-SA.resx
350:         //     ResourceLanguage's value should be "ar-SA"
351:         //     ResourceFlowDirection's value should be "RightToLeft"
352:         //
353:         // For more info on localizing Windows Phone apps see
http://go.microsoft.com/fwlink/?LinkId=262072.
354:         //
355:         private void InitializeLanguage()
356:         {
357:             try
358:             {
359:                 // Set the font to match the display language defined by the
360:                 // ResourceLanguage resource string for each supported language.
361:                 //
362:                 // Fall back to the font of the neutral language if the Display
363:                 // language of the phone is not supported.
364:                 //
365:                 // If a compiler error is hit then ResourceLanguage is missing from
366:                 // the resource file.
367:                 RootFrame.Language =
XmlLanguage.GetLanguage(AppResources.ResourceLanguage);
368:
369:                 // Set the FlowDirection of all elements under the root frame based
370:                 // on the ResourceFlowDirection resource string for each
371:                 // supported language.
372:                 //
373:                 // If a compiler error is hit then ResourceFlowDirection is missing
from
374:                 // the resource file.
375:                 FlowDirection flow = (FlowDirection)Enum.Parse(typeof(FlowDirection),
AppResources.ResourceFlowDirection);
376:                 RootFrame.FlowDirection = flow;
377:             }
378:             catch
379:             {
380:                 // If an exception is caught here it is most likely due to either
381:                 // ResourceLangauge not being correctly set to a supported language
382:                 // code or ResourceFlowDirection is set to a value other than
LeftToRight
383:                 // or RightToLeft.
384:
385:                 if (Debugger.IsAttached)
386:                 {
387:                     Debugger.Break();
388:                 }
389:
390:                 throw;
391:             }
392:         }
393:

```

```

394:         public string ReadFile(string filePath)
395:         {
396:             var ResrouceStream = Application.GetResourceStream(new Uri(filePath,
UriKind.Relative));
397:             if (ResrouceStream != null)
398:             {
399:                 Stream myFileStream = ResrouceStream.Stream;
400:                 if (myFileStream.CanRead)
401:                 {
402:                     StreamReader myStreamReader = new StreamReader(myFileStream);
403:
404:                     return myStreamReader.ReadToEnd();
405:                 }
406:             }
407:             return "";
408:         }
409:
410:         private static void DilAyariOlustur(NameDataContext context)
411:         {
412:             var appSettings = new AppSettings()
413:             {
414:                 //AppLangId = 42,
415:                 AppLangName = "TR",
416:                 AppBackgroundColor = "BLA",
417:                 FontFamily = "Verdana",
418:                 FontSize = "30",
419:                 AppBackgroundImage = null
420:             };
421:
422:             context.AppSettings.InsertOnSubmit(appSettings);
423:             context.SubmitChanges();
424:
425:             CultureInfo newCulture = new CultureInfo("tr");
426:             Thread.CurrentThread.CurrentCulture = newCulture;
427:             Thread.CurrentThread.CurrentUICulture = newCulture;
428:         }
429:     }
430: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.5 AppResources.Designer.cs

This code was generated by a tool. Runtime Version:4.0.30319.17626

Changes to this file may cause incorrect behavior and will be lost if the code is regenerated.

Body Source

```

1: ?//-----
2: // <auto-generated>
3: //     This code was generated by a tool.
4: //     Runtime Version:4.0.30319.17626
5: //
6: //     Changes to this file may cause incorrect behavior and will be lost if
7: //     the code is regenerated.
8: // </auto-generated>
9: //-----
10:
11: namespace IsimSozlugu.Resources
12: {
13:     using System;

```

```

14:
15:
16:     /// <summary>
17:     ///     A strongly-typed resource class, for looking up localized strings, etc.
18:     /// </summary>
19:     /// This class was auto-generated by the StronglyTypedResourceBuilder
20:     /// class via a tool like ResGen or Visual Studio.
21:     /// To add or remove a member, edit your .ResX file then rerun ResGen
22:     /// with the /str option, or rebuild your VS project.
23:
[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder",
"4.0.0.0")]
24:     [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
25:     [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
26:     public class AppResources
27:     {
28:
29:         private static global::System.Resources.ResourceManager resourceMan;
30:
31:         private static global::System.Globalization.CultureInfo resourceCulture;
32:
33:
[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance",
"CA1811:AvoidUncalledPrivateCode")]
34:         internal AppResources()
35:         {
36:         }
37:
38:         /// <summary>
39:         ///     Returns the cached ResourceManager instance used by this class.
40:         /// </summary>
41:
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
42:         public static global::System.Resources.ResourceManager ResourceManager
43:         {
44:             get
45:             {
46:                 if (object.ReferenceEquals(resourceMan, null))
47:                 {
48:                     global::System.Resources.ResourceManager temp = new
global::System.Resources.ResourceManager("IsimSozlugu.Resources.AppResources",
typeof(AppResources).Assembly);
49:                     resourceMan = temp;
50:                 }
51:                 return resourceMan;
52:             }
53:         }
54:
55:         /// <summary>
56:         ///     Overrides the current thread's CurrentUICulture property for all
57:         ///     resource lookups using this strongly typed resource class.
58:         /// </summary>
59:
[global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]
60:         public static global::System.Globalization.CultureInfo Culture
61:         {
62:             get
63:             {
64:                 return resourceCulture;
65:             }
66:             set
67:             {
68:                 resourceCulture = value;
69:             }
70:         }
71:

```

```
72:         /// <summary>
73:         ///     Looks up a localized string similar to LeftToRight.
74:         /// </summary>
75:         public static string ResourceFlowDirection
76:         {
77:             get
78:             {
79:                 return ResourceManager.GetString("ResourceFlowDirection",
resourceCulture);
80:             }
81:         }
82:
83:         /// <summary>
84:         ///     Looks up a localized string similar to us-EN.
85:         /// </summary>
86:         public static string ResourceLanguage
87:         {
88:             get
89:             {
90:                 return ResourceManager.GetString("ResourceLanguage", resourceCulture);
91:             }
92:         }
93:
94:         /// <summary>
95:         ///     Looks up a localized string similar to MY APPLICATION.
96:         /// </summary>
97:         public static string ApplicationTitle
98:         {
99:             get
100:             {
101:                 return ResourceManager.GetString("ApplicationTitle", resourceCulture);
102:             }
103:         }
104:
105:         /// <summary>
106:         ///     Looks up a localized string similar to button.
107:         /// </summary>
108:         public static string AppBarButtonText
109:         {
110:             get
111:             {
112:                 return ResourceManager.GetString("AppBarButtonText", resourceCulture);
113:             }
114:         }
115:
116:         /// <summary>
117:         ///     Looks up a localized string similar to menu item.
118:         /// </summary>
119:         public static string AppBarMenuItemText
120:         {
121:             get
122:             {
123:                 return ResourceManager.GetString("AppBarMenuItemText",
resourceCulture);
124:             }
125:         }
126:     }
127: }
```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.6 AppSettings.cs

This is file AppSettings.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu
10: {
11:     [Table]
12:     public class AppSettings
13:     {
14:         [Column(IsPrimaryKey = true,
15:             IsDbGenerated = true,
16:             DbType = "INT NOT NULL Identity",
17:             CanBeNull = false)]
18:         public int AppSettingsId { get; set; }
19:
20:         [Column]
21:         public string AppLangName { get; set; }
22:
23:         [Column]
24:         public string AppBackgroundColor { get; set; }
25:
26:         [Column]
27:         public string FontFamily { get; set; }
28:         [Column]
29:         public string FontSize { get; set; }
30:
31:         [Column(DbType = "Image", UpdateCheck = UpdateCheck.Never)]
32:         public byte[] AppBackgroundImage { get; set; }
33:     }
34: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.7 AssemblyInfo.cs

This is file AssemblyInfo.cs.

Body Source

```

1: ?using System.Reflection;
2: using System.Runtime.CompilerServices;
3: using System.Runtime.InteropServices;
4: using System.Resources;
5:
6: // General Information about an assembly is controlled through the following
7: // set of attributes. Change these attribute values to modify the information
8: // associated with an assembly.
9: [assembly: AssemblyTitle("Isimler Sözlüğü")]

```



```

10: [assembly: AssemblyDescription("İsimler ve Anlamları")]
11: [assembly: AssemblyConfiguration("")]
12: [assembly: AssemblyCompany("ÇMS Software")]
13: [assembly: AssemblyProduct("İsimSozlugu")]
14: [assembly: AssemblyCopyright("Copyright © 2015")]
15: [assembly: AssemblyTrademark("")]
16: [assembly: AssemblyCulture("")]
17:
18: // Setting ComVisible to false makes the types in this assembly not visible
19: // to COM components. If you need to access a type in this assembly from
20: // COM, set the ComVisible attribute to true on that type.
21: [assembly: ComVisible(false)]
22:
23: // The following GUID is for the ID of the typelib if this project is exposed to COM
24: [assembly: Guid("92fac230-d845-45b6-b7e1-8949ce4619d4")]
25:
26: // Version information for an assembly consists of the following four values:
27: //
28: //      Major Version
29: //      Minor Version
30: //      Build Number
31: //      Revision
32: //
33: // You can specify all the values or you can default the Revision and Build Numbers
34: // by using the '*' as shown below:
35: [assembly: AssemblyVersion("1.0.1.0")]
36: [assembly: AssemblyFileVersion("1.0.1.0")]
37: [assembly: NeutralResourcesLanguageAttribute("en-US")]

```

1.2.8 BackgroundColorSettingsPage.xaml.cs

This is file BackgroundColorSettingsPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;
5: using System.Net;
6: using System.Windows;
7: using System.Windows.Controls;
8: using System.Windows.Media;
9: using System.Windows.Media.Imaging;
10: using System.Windows.Navigation;
11: using IsimSozlugu.Resources;
12: using Microsoft.Phone.Controls;
13: using Microsoft.Phone.Shell;
14:
15: namespace IsimSozlugu
16: {
17:     public partial class BackgroundColorSettingsPage : PhoneApplicationPage
18:     {
19:         public int artistId;
20:
21:         public BackgroundColorSettingsPage()
22:         {
23:             InitializeComponent();
24:
25:             lstBackgroundColor.Items.Clear();
26:             lstBackgroundColor.Items.Add("Black");
27:             lstBackgroundColor.Items.Add("Blue");
28:             lstBackgroundColor.Items.Add("Brown");
29:             lstBackgroundColor.Items.Add("Gray");
30:             lstBackgroundColor.Items.Add("Green");
31:             lstBackgroundColor.Items.Add("Orange");

```

```

32:         lstBackgroundColor.Items.Add("Purple");
33:         lstBackgroundColor.Items.Add("Red");
34:         lstBackgroundColor.Items.Add("Yellow");
35:         lstBackgroundColor.SelectedIndex = -1;
36:
37:         SetBackgroundColor();
38:     }
39:
40:     private void SetBackgroundColor()
41:     {
42:         AppSettings appSettings = new AppSettings();
43:         using (var context = new NameDataContext(NameDataContext.ConnectionString))
44:         {
45:             appSettings = context.AppSettings.First() as AppSettings;
46:         }
47:
48:         if (appSettings.AppBackgroundImage != null)
49:         {
50:             MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
51:             BitmapImage image = new BitmapImage();
52:             image.SetSource(stream);
53:             ImageBrush ib = new ImageBrush();
54:             ib.ImageSource = image;
55:             this.LayoutRoot.Background = ib;
56:         }
57:         else
58:         {
59:             switch (appSettings.AppBackgroundColor)
60:             {
61:                 case "BLA":
62:                     this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
63:                     break;
64:                 case "BLU":
65:                     this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
66:                     break;
67:                 case "BRO":
68:                     this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
69:                     break;
70:                 case "RED":
71:                     this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
72:                     break;
73:                 case "GRE":
74:                     this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
75:                     break;
76:                 case "GRA":
77:                     this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
78:                     break;
79:                 case "YEL":
80:                     this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
81:                     break;
82:                 case "ORA":
83:                     this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
84:                     break;
85:                 case "PUR":
86:                     this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
87:                     break;
88:                 default:
89:                     this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
90:                     break;
91:             }
92:         }
93:     }
94:
95:     protected override void OnNavigatedTo(NavigationEventArgs e)
96:     {
97:         base.OnNavigatedTo(e);

```

```
98:         //SetBackgroundColor();
99:         //while (NavigationService.CanGoBack)
100:         //NavigationService.RemoveBackEntry();
101:     }
102:
103:
104:     protected override void OnNavigatedFrom(NavigationEventArgs e)
105:     {
106:         base.OnNavigatedFrom(e);
107:         //while (NavigationService.CanGoBack)
108:         //NavigationService.RemoveBackEntry();
109:     }
110:
111:
112:     protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
113:     {
114:         // displays "Fragment: Detail"
115:         //MessageBox.Show("Folder Id: " + e.Fragment);
116:         base.OnFragmentNavigation(e);
117:     }
118:
119:     private void lstBackgroundColor_SelectionChanged(object sender,
SelectionChangedEventArgs e)
120:     {
121:         int index = lstBackgroundColor.SelectedIndex;
122:         string backgroundColor = "";
123:         if (index == 0)
124:         {
125:             backgroundColor = "BLA";
126:         }
127:         else if (index == 1)
128:         {
129:             backgroundColor = "BLU";
130:         }
131:         else if (index == 2)
132:         {
133:             backgroundColor = "BRO";
134:         }
135:         else if (index == 3)
136:         {
137:             backgroundColor = "GRA";
138:         }
139:         else if (index == 4)
140:         {
141:             backgroundColor = "GRE";
142:         }
143:         else if (index == 5)
144:         {
145:             backgroundColor = "ORA";
146:         }
147:         else if (index == 6)
148:         {
149:             backgroundColor = "PUR";
150:         }
151:         else if (index == 7)
152:         {
153:             backgroundColor = "RED";
154:         }
155:         else if (index == 8)
156:         {
157:             backgroundColor = "YEL";
158:         }
159:         else
160:         {
161:             backgroundColor = "BLA";
162:         }
163:
164:         using (var context = new NameDataContext(NameDataContext.ConnectionString))
165:         {
```

```

166:         var appSettings = context.AppSettings;
167:         foreach (var appSetting in appSettings)
168:         {
169:             appSetting.AppBackgroundColor = backgroundColor;
170:         }
171:         context.SubmitChanges();
172:         //CustomMessageBox messageBox = new CustomMessageBox()
173:         //{
174:             //    Caption = "BackgroundColor,
175:             //    Message = "SuccessfulBackgroundColorChanged,
176:             //    Background = messageBackGround
177:         //};
178:         //messageBox.Show();
179:         MessageBox.Show("Arka Plan Rengi Basariyla Degistirildi");
180:     }
181:     SetBackgroundColor();
182:     NavigationService.Navigate(new Uri("/GeneralSettingsPage.xaml",
UriKind.Relative));
183:     }
184:
185:     private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
186:     {
187:         if (this.NavigationService.CanGoBack)
188:         {
189:             this.NavigationService.Navigate(new Uri("/GeneralSettingsPage.xaml",
UriKind.Relative));
190:         }
191:     }
192:
193:     private void PhoneApplicationPage_Loaded(object sender, RoutedEventArgs e)
194:     {
195:         //SetBackgroundColor();
196:     }
197: }
198: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.9 Favourite.cs

This is file Favourite.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu
10: {
11:     [Table]
12:     public class Favourite
13:     {
14:         [Column(IsPrimaryKey = true,
15:             IsDbGenerated = true,
16:             DbType = "INT NOT NULL Identity",
17:             CanBeNull = false)]

```

```

18:         public int FavouriteId { get; set; }
19:
20:         [Column]
21:         public string FavouriteName { get; set; }
22:     }
23: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.10 Female.cs

This is file Female.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu
10: {
11:     [Table]
12:     public class Female
13:     {
14:         [Column(IsPrimaryKey = true,
15:             IsDbGenerated = true,
16:             DbType = "INT NOT NULL Identity",
17:             CanBeNull = false)]
18:         public int FemaleId { get; set; }
19:
20:         [Column]
21:         public string FemaleName { get; set; }
22:
23:         [Column]
24:         public string FemaleMeaning { get; set; }
25:
26:     }
27: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.11 FontFamilySettingsPage.xaml.cs

This is file FontFamilySettingsPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;

```

```

5: using System.Net;
6: using System.Windows;
7: using System.Windows.Controls;
8: using System.Windows.Controls.Primitives;
9: using System.Windows.Media;
10: using System.Windows.Media.Imaging;
11: using System.Windows.Navigation;
12: using System.Data.Common;
13: using Microsoft.Phone.Controls;
14: using Microsoft.Phone.Controls.Primitives;
15: using Microsoft.Phone.Shell;
16: using IsimSozlugu.Resources;
17:
18: namespace IsimSozlugu
19: {
20:     public partial class FontFamilySettingsPage : PhoneApplicationPage
21:     {
22:         public int artistId;
23:         public FontFamilySettingsPage()
24:         {
25:             InitializeComponent();
26:
27:             lstFontFamily.Items.Clear();
28:             lstFontFamily.Items.Add("Arial");
29:             lstFontFamily.Items.Add("Arial Black");
30:             lstFontFamily.Items.Add("Baskerville Old Face");
31:             lstFontFamily.Items.Add("Berlin Sans FB");
32:             lstFontFamily.Items.Add("Albumman Old Style");
33:             lstFontFamily.Items.Add("Calibri");
34:             lstFontFamily.Items.Add("Cambria");
35:             lstFontFamily.Items.Add("Candara");
36:             lstFontFamily.Items.Add("Comic Sans MS");
37:             lstFontFamily.Items.Add("Consolas");
38:             lstFontFamily.Items.Add("Constantia");
39:             lstFontFamily.Items.Add("Courier New");
40:             lstFontFamily.Items.Add("DokChampa");
41:             lstFontFamily.Items.Add("Ebrima");
42:             lstFontFamily.Items.Add("Georgia");
43:             lstFontFamily.Items.Add("Lucida Sans Unicode");
44:             lstFontFamily.Items.Add("Meiryo UI");
45:             lstFontFamily.Items.Add("Microsoft YaHei");
46:             lstFontFamily.Items.Add("Malgun Gothic");
47:             lstFontFamily.Items.Add("Segoe UI");
48:             lstFontFamily.Items.Add("Segoe WP");
49:             lstFontFamily.Items.Add("Tahoma");
50:             lstFontFamily.Items.Add("Trebuchet MS");
51:             lstFontFamily.Items.Add("Times New Roman");
52:             lstFontFamily.Items.Add("Verdana");
53:             lstFontFamily.SelectedIndex = -1;
54:         }
55:
56:         private void lstFontFamily_SelectionChanged(object sender,
57: SelectionChangedEventArgs e)
58:         {
59:             if (lstFontFamily.SelectedIndex != -1)
60:             {
61:                 using (var context = new
62: NameDataContext(NameDataContext.ConnectionString))
63:                 {
64:                     var appSettings = context.AppSettings;
65:                     foreach (var item in appSettings)
66:                     {
67:                         item.FontFamily = lstFontFamily.SelectedItem.ToString();
68:                     }
69:                     context.SubmitChanges();
70:                     MessageBox.Show("Font Ailesi Basariyla Degistirildi");
71:                 }
72:                 NavigationService.Navigate(new Uri("/GeneralSettingsPage.xaml#" +

```

```

artistId, UriKind.Relative));
72:         }
73:
74:         protected override void OnNavigatedTo(NavigationEventArgs e)
75:         {
76:             base.OnNavigatedTo(e);
77:             SetBackgroundColor();
78:         }
79:
80:         protected override void OnNavigatedFrom(NavigationEventArgs e)
81:         {
82:             base.OnNavigatedFrom(e);
83:         }
84:
85:         protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
86:         {
87:             // displays "Fragment: Detail"
88:             //MessageBox.Show("Folder Id: " + e.Fragment);
89:             base.OnFragmentNavigation(e);
90:         }
91:
92:         private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
93:         {
94:             if (this.NavigationService.CanGoBack)
95:             {
96:                 this.NavigationService.Navigate(new Uri("/GeneralSettingsPage.xaml#" +
artistId, UriKind.Relative));
97:             }
98:         }
99:
100:        private void SetBackgroundColor()
101:        {
102:            AppSettings appSettings = new AppSettings();
103:            using (var context = new NameDataContext(NameDataContext.ConnectionString))
104:            {
105:                appSettings = context.AppSettings.First() as AppSettings;
106:            }
107:
108:            if (appSettings.AppBackgroundImage != null)
109:            {
110:                MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
111:                BitmapImage image = new BitmapImage();
112:                image.SetSource(stream);
113:                ImageBrush ib = new ImageBrush();
114:                ib.ImageSource = image;
115:                this.LayoutRoot.Background = ib;
116:            }
117:            else
118:            {
119:                switch (appSettings.AppBackgroundColor)
120:                {
121:                    case "BLA":
122:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
123:                        break;
124:                    case "BLU":
125:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
126:                        break;
127:                    case "BRO":
128:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
129:                        break;
130:                    case "RED":
131:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
132:                        break;
133:                    case "GRE":
134:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
135:                        break;
136:                    case "GRA":
137:                        this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);

```

```

138:                 break;
139:             case "YEL":
140:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
141:                 break;
142:             case "ORA":
143:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
144:                 break;
145:             case "PUR":
146:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
147:                 break;
148:             default:
149:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
150:                 break;
151:         }
152:     }
153: }
154:
155: }
156: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.12 FontSizeSettingsPage.xaml.cs

This is file FontSizeSettingsPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;
5: using System.Net;
6: using System.Windows;
7: using System.Windows.Controls;
8: using System.Windows.Controls.Primitives;
9: using System.Windows.Media;
10: using System.Windows.Media.Imaging;
11: using System.Windows.Navigation;
12: using System.Data.Common;
13: using Microsoft.Phone.Controls;
14: using Microsoft.Phone.Controls.Primitives;
15: using Microsoft.Phone.Shell;
16: using IsimSozlugu.Resources;
17:
18: namespace IsimSozlugu
19: {
20:     public partial class FontSizeSettingsPage : PhoneApplicationPage
21:     {
22:         public int artistId;
23:         public FontSizeSettingsPage()
24:         {
25:             InitializeComponent();
26:
27:             lstFontSize.Items.Clear();
28:             lstFontSize.Items.Add("14");
29:             lstFontSize.Items.Add("18");
30:             lstFontSize.Items.Add("22");
31:             lstFontSize.Items.Add("26");

```



```

32:         lstFontSize.Items.Add("28");
33:         lstFontSize.Items.Add("30");
34:         lstFontSize.Items.Add("32");
35:         lstFontSize.Items.Add("34");
36:         lstFontSize.Items.Add("36");
37:         lstFontSize.Items.Add("38");
38:         lstFontSize.Items.Add("40");
39:         lstFontSize.Items.Add("42");
40:         lstFontSize.Items.Add("44");
41:         lstFontSize.Items.Add("64");
42:         lstFontSize.Items.Add("72");
43:         lstFontSize.SelectedIndex = -1;
44:     }
45:
46:     private void lstFontSize_SelectionChanged(object sender,
SelectionChangedEventArgs e)
47:     {
48:         if (lstFontSize.SelectedIndex != -1)
49:         {
50:             using (var context = new
NameDataContext(NameDataContext.ConnectionString))
51:             {
52:                 var appSettings = context.AppSettings;
53:                 foreach (var item in appSettings)
54:                 {
55:                     item.FontSize = lstFontSize.SelectedItem.ToString();
56:                 }
57:                 context.SubmitChanges();
58:                 MessageBox.Show("Font Boyutu Basariyla Degistirildi");
59:             }
60:         }
61:         NavigationService.Navigate(new Uri("/GeneralSettingsPage.xaml#" +
artistId, UriKind.Relative));
62:     }
63:
64:     protected override void OnNavigatedTo(NavigationEventArgs e)
65:     {
66:         base.OnNavigatedTo(e);
67:         SetBackgroundColor();
68:     }
69:
70:     protected override void OnNavigatedFrom(NavigationEventArgs e)
71:     {
72:         base.OnNavigatedFrom(e);
73:     }
74:
75:     protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
76:     {
77:         // displays "Fragment: Detail"
78:         //MessageBox.Show("Folder Id: " + e.Fragment);
79:         base.OnFragmentNavigation(e);
80:     }
81:
82:     private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
83:     {
84:         if (this.NavigationService.CanGoBack)
85:         {
86:             this.NavigationService.Navigate(new Uri("/GeneralSettingsPage.xaml#" +
artistId, UriKind.Relative));
87:         }
88:     }
89:
90:     private void SetBackgroundColor()
91:     {
92:         AppSettings appSettings = new AppSettings();
93:         using (var context = new NameDataContext(NameDataContext.ConnectionString))
94:         {
95:             appSettings = context.AppSettings.First() as AppSettings;

```

```

96:         }
97:
98:         if (appSettings.AppBackgroundImage != null)
99:         {
100:             MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
101:             BitmapImage image = new BitmapImage();
102:             image.SetSource(stream);
103:             ImageBrush ib = new ImageBrush();
104:             ib.ImageSource = image;
105:             this.LayoutRoot.Background = ib;
106:         }
107:     else
108:     {
109:         switch (appSettings.AppBackgroundColor)
110:         {
111:             case "BLA":
112:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
113:                 break;
114:             case "BLU":
115:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
116:                 break;
117:             case "BRO":
118:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
119:                 break;
120:             case "RED":
121:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
122:                 break;
123:             case "GRE":
124:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
125:                 break;
126:             case "GRA":
127:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
128:                 break;
129:             case "YEL":
130:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
131:                 break;
132:             case "ORA":
133:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
134:                 break;
135:             case "PUR":
136:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
137:                 break;
138:             default:
139:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
140:                 break;
141:         }
142:     }
143: }
144: }
145: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.13 GeneralSettingsPage.xaml.cs

This is file GeneralSettingsPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Globalization;
4: using System.IO;
5: using System.IO.IsolatedStorage;
6: using System.Linq;
7: using System.Net;
8: using System.Text;
9: using System.Threading;
10: using System.Threading.Tasks;
11: using System.Windows;
12: using System.Windows.Controls;
13: using System.Windows.Media;
14: using System.Windows.Media.Imaging;
15: using System.Windows.Navigation;
16: using IsimSozlugu.Resources;
17: using Microsoft.Phone.Controls;
18: using Microsoft.Phone.Shell;
19: using Microsoft.Phone.Tasks;
20:
21: namespace IsimSozlugu
22: {
23:     public partial class GeneralSettingsPage : PhoneApplicationPage
24:     {
25:
26:         public GeneralSettingsPage()
27:         {
28:             InitializeComponent();
29:             SetBackgroundColor();
30:
31:             using (var context = new NameDataContext(NameDataContext.ConnectionString))
32:             {
33:                 var appSettings = context.AppSettings.First() as AppSettings;
34:                 if (appSettings.AppBackgroundColor == "BLA")
35:                 {
36:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Siyah)";
37:                 }
38:                 if (appSettings.AppBackgroundColor == "BLU")
39:                 {
40:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Mavi)";
41:                 }
42:                 if (appSettings.AppBackgroundColor == "BRO")
43:                 {
44:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Kahverengi)";
45:                 }
46:                 if (appSettings.AppBackgroundColor == "RED")
47:                 {
48:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Kırmızı)";
49:                 }
50:                 if (appSettings.AppBackgroundColor == "GRE")
51:                 {
52:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Yeşil)";
53:                 }
54:                 if (appSettings.AppBackgroundColor == "YEL")
55:                 {
56:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Sarı)";
57:                 }
58:                 if (appSettings.AppBackgroundColor == "GRA")
59:                 {
60:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Gri)";
61:                 }
62:                 if (appSettings.AppBackgroundColor == "ORA")
63:                 {
64:                     lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Turuncu)";
65:                 }
66:                 if (appSettings.AppBackgroundColor == "PUR")

```

```

67:         {
68:             lblBackgroundColor.Text = "Arka Plan Rengi (Seçilen: Mor)";
69:         }
70:
71:         lblFontFamily.Text = "Font Ailesi (Seçilen: " + appSettings.FontFamily
+ " )";
72:         lblFontSize.Text = "Font Boyutu (Seçilen: " + appSettings.FontSize +
" )";
73:
74:     }
75: }
76:
77: protected override void OnNavigatedTo(NavigationEventArgs e)
78: {
79:     base.OnNavigatedTo(e);
80:     SetBackgroundColor();
81:     //while (NavigationService.CanGoBack)
82:     //NavigationService.RemoveBackEntry();
83:
84: }
85:
86: protected override void OnNavigatedFrom(NavigationEventArgs e)
87: {
88:     base.OnNavigatedFrom(e);
89:     //while (NavigationService.CanGoBack)
90:     //NavigationService.RemoveBackEntry();
91:
92: }
93:
94: private void SetBackgroundColor()
95: {
96:     AppSettings appSettings = new AppSettings();
97:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
98:     {
99:         appSettings = context.AppSettings.First() as AppSettings;
100:     }
101:
102:     if (appSettings.AppBackgroundImage != null)
103:     {
104:         MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
105:         BitmapImage image = new BitmapImage();
106:         image.SetSource(stream);
107:         ImageBrush ib = new ImageBrush();
108:         ib.ImageSource = image;
109:         this.LayoutRoot.Background = ib;
110:     }
111:     else
112:     {
113:         switch (appSettings.AppBackgroundColor)
114:         {
115:             case "BLA":
116:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
117:                 break;
118:             case "BLU":
119:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
120:                 break;
121:             case "BRO":
122:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
123:                 break;
124:             case "RED":
125:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
126:                 break;
127:             case "GRE":
128:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
129:                 break;
130:             case "GRA":
131:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
132:                 break;
133:             case "YEL":

```

```

134:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
135:                 break;
136:                 case "ORA":
137:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
138:                 break;
139:                 case "PUR":
140:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
141:                 break;
142:                 default:
143:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
144:                 break;
145:             }
146:         }
147:     }
148:
149:     private void btnBackgroundColor_Click(object sender, RoutedEventArgs e)
150:     {
151:         NavigationService.Navigate(new Uri("/BackgroundColorSettingsPage.xaml",
UriKind.Relative));
152:     }
153:
154:     private void btnBackgroundImage_Click(object sender, RoutedEventArgs e)
155:     {
156:         PhotoChooserTask objPhotoChooser = new PhotoChooserTask();
157:         objPhotoChooser.Completed += new
EventHandler<PhotoResult>(PhotoChooseCall);
158:         objPhotoChooser.Show();
159:     }
160:
161:     private void PhotoChooseCall(object sender, PhotoResult e)
162:     {
163:         switch (e.TaskResult)
164:         {
165:             case TaskResult.OK:
166:                 using (var context = new
NameDataContext(NameDataContext.ConnectionString))
167:                 {
168:                     var appSettings = context.AppSettings;
169:                     foreach (var appSetting in appSettings)
170:                     {
171:                         appSetting.AppBackgroundImage = new
byte[e.ChosenPhoto.Length];
172:                         e.ChosenPhoto.Position = 0;
173:                         e.ChosenPhoto.Read(appSetting.AppBackgroundImage, 0,
appSetting.AppBackgroundImage.Length);
174:                         //noteFolder.NoteFolderPassword = "";
175:                     }
176:                     context.SubmitChanges();
177:                     MessageBox.Show("Arka Plan Resmi Basariyla Degistirildi");
178:                 }
179:                 break;
180:             case TaskResult.Cancel:
181:                 //MessageBox.Show("Cancelled");
182:                 break;
183:             case TaskResult.None:
184:                 //MessageBox.Show("Nothing Entered");
185:                 break;
186:         }
187:         SetBackgroundColor();
188:     }
189:
190:     private void btnRemoveBackgroundImage_Click(object sender, RoutedEventArgs e)
191:     {
192:         using (var context = new NameDataContext(NameDataContext.ConnectionString))
193:         {
194:             var appSettings = context.AppSettings;

```

```

195:         foreach (var appSetting in appSettings)
196:         {
197:             appSetting.AppBackgroundImage = null;
198:         }
199:         context.SubmitChanges();
200:         MessageBox.Show("Arka Plan Resmi Basariyla Kaldirildi");
201:     }
202: }
203:
204: private void PhoneApplicationPage_Loaded(object sender, RoutedEventArgs e)
205: {
206:
207:     //pvGeneralSettings.Title = AppResources.GeneralSettings;
208:
209:     //piLanguage.Header = AppResources.Language;
210:     //piSync.Header = AppResources.Sync;
211:     //piOtherSettings.Header = AppResources.OtherSettings;
212:     //piBackground.Header = AppResources.Background;
213:
214:     ///lblOneDrive.Text = AppResources.OneDrive;
215:
216:     //btnCategoryOrder.Content = AppResources.Select;
217:     //btnCategoryOrderStyle.Content = AppResources.Select;
218:     //btnLanguage.Content = AppResources.Select;
219:     //btnBackgroundColor.Content = AppResources.Select;
220:     ///btnOneDrive.Content = AppResources.Login;
221:     ///btnOneDrive.SignInText = AppResources.SignIn;
222:     ///btnOneDrive.SignOutText = AppResources.SignOut;
223:     //btnOneDriveSync.Content = AppResources.Sync;
224:     //lblOneDrive.Text = AppResources.OneDrive;
225:     //txtSynchronizing.Text = AppResources.Synchronizing;
226:
227:     //pbSync.Visibility = Visibility.Collapsed;
228:     //txtSynchronizing.Visibility = Visibility.Collapsed;
229:     //txtSynchronizing.BorderBrush = this.LayoutRoot.Background;
230:
231:     //btnRemoveBackgroundImage.Content = AppResources.RemoveBackgroundImage;
232:     //lblBackgroundImage.Text = AppResources.BackgroundImage;
233:     //btnBackgroundImage.Content = AppResources.Select;
234:     //btnResetSettings.Content = AppResources.ResetSettings;
235:
236:     //btnOneDriveSync.IsEnabled = false;
237:     //cbSync.Content = AppResources.SyncOnOneFile;
238:     //cbSync.IsEnabled = false;
239:     //btnOneDrive.Content = "Sign In";
240:
241:     //SetBackgroundColor();
242: }
243:
244: private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
245: {
246:     if (this.NavigationService.CanGoBack)
247:     {
248:         this.NavigationService.Navigate(new Uri("/MainPage.xaml",
UriKind.Relative));
249:     }
250: }
251:
252: private void btnResetSettings_Click(object sender, RoutedEventArgs e)
253: {
254:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
255:     {
256:         var appSettings = context.AppSettings;
257:         foreach (var appSetting in appSettings)
258:         {
259:             appSetting.AppBackgroundImage = null;
260:             appSetting.AppBackgroundColor = "BLA";
261:         }

```

```

262:         context.SubmitChanges();
263:         this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
264:         MessageBox.Show("Arka Plan Ayarları Basarıyla Sifirlandı");
265:     }
266: }
267:
268: private void btnFontSize_Click(object sender, RoutedEventArgs e)
269: {
270:     this.NavigationService.Navigate(new Uri("/FontSizeSettingsPage.xaml",
UriKind.Relative));
271: }
272:
273: private void btnFontFamily_Click(object sender, RoutedEventArgs e)
274: {
275:     this.NavigationService.Navigate(new Uri("/FontFamilySettingsPage.xaml",
UriKind.Relative));
276: }
277: }
278: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.14 IsimSozlugu.csproj

This is file IsimSozlugu.csproj.

1.2.15 IsimSozlugu.sln

This is file IsimSozlugu.sln.

1.2.16 LocalizedStrings.cs

This is file LocalizedStrings.cs.

Body Source

```

1: using IsimSozlugu.Resources;
2:
3: namespace IsimSozlugu
4: {
5:     /// <summary>
6:     /// Provides access to string resources.
7:     /// </summary>
8:     public class LocalizedStrings
9:     {
10:         private static AppResources _localizedResources = new AppResources();
11:
12:         public AppResources LocalizedResources { get { return _localizedResources; } }
13:     }
14: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.17 MainPage.xaml.cs

This is file MainPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;
5: using System.Net;
6: using System.Text;
7: using System.Windows;
8: using System.Windows.Controls;
9: using System.Windows.Controls.Primitives;
10: using System.Windows.Input;
11: using System.Windows.Media;
12: using System.Windows.Media.Imaging;
13: using System.Windows.Navigation;
14: using Microsoft.Phone.Tasks;
15: using IsimSozlugu.Resources;
16: using Microsoft.Phone.Controls;
17: using Microsoft.Phone.Shell;
18:
19: namespace IsimSozlugu
20: {
21:     public partial class MainPage : PhoneApplicationPage
22:     {
23:         public MainPage()
24:         {
25:             InitializeComponent();
26:             ApplicationBar = new ApplicationBar();
27:
28:             ApplicationBarIconButton button1 = new ApplicationBarIconButton();
29:             button1.IconUri = new Uri("/Assets/Add.png", UriKind.Relative);
30:             button1.Text = "Isim Ekle";
31:             ApplicationBar.Buttons.Add(button1);
32:             button1.Click += new EventHandler(AddNameButton_Click);
33:
34:             ApplicationBarIconButton button2 = new ApplicationBarIconButton();
35:             button2.IconUri = new Uri("/Assets/Search.png", UriKind.Relative);
36:             button2.Text = "Ara";
37:             ApplicationBar.Buttons.Add(button2);
38:             button2.Click += new EventHandler(SearchButton_Click);
39:
40:             ApplicationBarIconButton button3 = new ApplicationBarIconButton();
41:             button3.IconUri = new Uri("/Assets/Settings.png", UriKind.Relative);
42:             button3.Text = "Ayarlar";
43:             ApplicationBar.Buttons.Add(button3);
44:             button3.Click += new EventHandler(SettingsButton_Click);
45:
46:             ApplicationBarIconButton button4 = new ApplicationBarIconButton();
47:             button4.IconUri = new Uri("/Assets/Statistics.png", UriKind.Relative);
48:             button4.Text = "İstatistikler";
49:             ApplicationBar.Buttons.Add(button4);
50:             button4.Click += new EventHandler(StatisticsButton_Click);
51:
52:             ApplicationBarMenuItem menuItem1 = new ApplicationBarMenuItem();
53:             menuItem1.Text = "Hakkında";

```



```

54:         ApplicationBar.MenuItems.Add(menuItem1);
55:         menuItem1.Click += new EventHandler(AboutMenuItem_Click);
56:
57:         SetBackgroundColor();
58:
59:     }
60:
61:     private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
62:     {
63:         if (MessageBox.Show("Uygulamadan Çıkmak İstediginize Emin Misiniz?",
64:             "Uygulamadan Çık", MessageBoxButton.OKCancel)
65:             != MessageBoxResult.OK)
66:         {
67:             e.Cancel = true;
68:         }
69:     }
70:     else
71:     {
72:         Application.Current.Terminate();
73:     }
74: }
75: private void pNames_SelectionChanged(object sender, SelectionChangedEventArgs
e)
76: {
77:     var context = new NameDataContext(NameDataContext.ConnectionString);
78:     if (pNames.SelectedIndex == 0)
79:     {
80:         List<Male> maleNames = context.MaleNames.OrderBy(j =>
j.MaleName).ToList() as List<Male>;
81:         List<AlphaKeyGroup<Male>> DataSource =
AlphaKeyGroup<Male>.CreateGroups(maleNames,
82:             System.Threading.Thread.CurrentThread.CurrentUICulture,
83:             (Male a) => { return a.MaleName; }, true);
84:         llsMaleNames.ItemsSource = DataSource;
85:     }
86:     else if (pNames.SelectedIndex == 1)
87:     {
88:         List<Female> femaleNames = context.FemaleNames.OrderBy(j =>
j.FemaleName).ToList() as List<Female>;
89:         List<AlphaKeyGroup<Female>> DataSource =
AlphaKeyGroup<Female>.CreateGroups(femaleNames,
90:             System.Threading.Thread.CurrentThread.CurrentUICulture,
91:             (Female a) => { return a.FemaleName; }, true);
92:         llsFemaleNames.ItemsSource = DataSource;
93:     }
94:     else if (pNames.SelectedIndex == 2)
95:     {
96:         List<Unisex> unisexNames = context.UnisexNames.OrderBy(j =>
j.UnisexName).ToList() as List<Unisex>;
97:         List<AlphaKeyGroup<Unisex>> DataSource =
AlphaKeyGroup<Unisex>.CreateGroups(unisexNames,
98:             System.Threading.Thread.CurrentThread.CurrentUICulture,
99:             (Unisex a) => { return a.UnisexName; }, true);
100:         llsUnisexNames.ItemsSource = DataSource;
101:     }
102:     else if (pNames.SelectedIndex == 3)
103:     {
104:         List<All> allNames = context.AllNames.OrderBy(j => j.AllName).ToList()
as List<All>;
105:         List<AlphaKeyGroup<All>> DataSource =
AlphaKeyGroup<All>.CreateGroups(allNames,
106:             System.Threading.Thread.CurrentThread.CurrentUICulture,
107:             (All a) => { return a.AllName; }, true);
108:         llsAllNames.ItemsSource = DataSource;
109:     }
110:     else if (pNames.SelectedIndex == 4)
111:     {
112:         List<MyUpdate> myUpdates = context.MyUpdates.OrderBy(j =>

```

```

j.MyUpdateName).ToList() as List<MyUpdate>;
113:         List<All> allNames = new List<All>();
114:         for (int i = 0; i < myUpdates.Count; i++)
115:         {
116:             var allName = context.AllNames.Where(j =>
j.AllName.Equals(myUpdates[i].MyUpdateName)).SingleOrDefault() as All;
117:             allNames.Add(allName);
118:         }
119:         List<AlphaKeyGroup<All>> DataSource =
AlphaKeyGroup<All>.CreateGroups(allNames,
120:             System.Threading.Thread.CurrentThread.CurrentUICulture,
121:             (All a) => { return a.AllName; }, true);
122:         llsMyUpdates.ItemsSource = DataSource;
123:     }
124:     else
125:     {
126:         List<Favourite> favourites = context.Favourites.OrderBy(j =>
j.FavouriteName).ToList() as List<Favourite>;
127:         List<All> allNames = new List<All>();
128:         for (int i = 0; i < favourites.Count; i++)
129:         {
130:             var allName = context.AllNames.Where(j =>
j.AllName.Equals(favourites[i].FavouriteName)).SingleOrDefault() as All;
131:             allNames.Add(allName);
132:         }
133:         List<AlphaKeyGroup<All>> DataSource =
AlphaKeyGroup<All>.CreateGroups(allNames,
134:             System.Threading.Thread.CurrentThread.CurrentUICulture,
135:             (All a) => { return a.AllName; }, true);
136:         llsFavourites.ItemsSource = DataSource;
137:     }
138: }
139:
140: private void SettingsButton_Click(object sender, EventArgs e)
141: {
142:     NavigationService.Navigate(new Uri("/GeneralSettingsPage.xaml",
UriKind.Relative));
143: }
144:
145: private void SearchButton_Click(object sender, EventArgs e)
146: {
147:     NavigationService.Navigate(new Uri("/SearchPage.xaml", UriKind.Relative));
148: }
149:
150: private void AboutMenuItem_Click(object sender, EventArgs e)
151: {
152:     NavigationService.Navigate(new Uri("/AboutPage.xaml", UriKind.Relative));
153: }
154: private void StatisticsButton_Click(object sender, EventArgs e)
155: {
156:     NavigationService.Navigate(new Uri("/StatisticsPage.xaml",
UriKind.Relative));
157: }
158:
159: private void AddNameButton_Click(object sender, EventArgs e)
160: {
161:     NavigationService.Navigate(new Uri("/NameDetailPage.xaml",
UriKind.Relative));
162:     //PhoneApplicationPage_Loaded(this, new RoutedEventArgs());
163: }
164:
165: private void llsMaleNames_SelectionChanged(object sender,
SelectionChangedEventArgs e)
166: {
167:     var maleName = llsMaleNames.SelectedItem as Male;
168:     NavigationService.Navigate(new Uri("/NameDetailPage.xaml#" +
maleName.MaleName, UriKind.Relative));
169: }
170:

```

```

171:         private void llsFemaleNames_SelectionChanged(object sender,
SelectionChangedEventArgs e)
172:         {
173:             var femaleName = llsFemaleNames.SelectedItem as Female;
174:             NavigationService.Navigate(new Uri("/NameDetailPage.xaml#" +
femaleName.FemaleName, UriKind.Relative));
175:         }
176:
177:         private void llsUnisexNames_SelectionChanged(object sender,
SelectionChangedEventArgs e)
178:         {
179:             var unisexName = llsUnisexNames.SelectedItem as Unisex;
180:             NavigationService.Navigate(new Uri("/NameDetailPage.xaml#" +
unisexName.UnisexName, UriKind.Relative));
181:         }
182:
183:         private void llsAllNames_SelectionChanged(object sender,
SelectionChangedEventArgs e)
184:         {
185:             var allName = llsAllNames.SelectedItem as All;
186:             NavigationService.Navigate(new Uri("/NameDetailPage.xaml#" +
allName.AllName, UriKind.Relative));
187:         }
188:
189:         private void llsFavourites_SelectionChanged(object sender,
SelectionChangedEventArgs e)
190:         {
191:             //var favourite = llsFavourites.SelectedItem as Favourite;
192:             var favourite = llsFavourites.SelectedItem as All;
193:             NavigationService.Navigate(new Uri("/NameDetailPage.xaml#" +
favourite.AllName, UriKind.Relative));
194:         }
195:
196:         private void llsMyUpdates_SelectionChanged(object sender,
SelectionChangedEventArgs e)
197:         {
198:
199:             //var myUpdate = llsMyUpdates.SelectedItem as MyUpdate;
200:             var myUpdate = llsMyUpdates.SelectedItem as All;
201:             NavigationService.Navigate(new Uri("/NameDetailPage.xaml#" +
myUpdate.AllName, UriKind.Relative));
202:         }
203:
204:         private void SetBackgroundColor()
205:         {
206:             AppSettings appSettings = new AppSettings();
207:             using (var context = new NameDataContext(NameDataContext.ConnectionString))
208:             {
209:                 appSettings = context.AppSettings.First() as AppSettings;
210:             }
211:
212:             if (appSettings.AppBackgroundImage != null)
213:             {
214:                 MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
215:                 BitmapImage image = new BitmapImage();
216:                 image.SetSource(stream);
217:                 ImageBrush ib = new ImageBrush();
218:                 ib.ImageSource = image;
219:                 this.LayoutRoot.Background = ib;
220:             }
221:             else
222:             {
223:                 switch (appSettings.AppBackgroundColor)
224:                 {
225:                     case "BLA":
226:                         this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
227:                         break;
228:                     case "BLU":
229:                         this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);

```

```

230:                 break;
231:             case "BRO":
232:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
233:                 break;
234:             case "RED":
235:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
236:                 break;
237:             case "GRE":
238:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
239:                 break;
240:             case "GRA":
241:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
242:                 break;
243:             case "YEL":
244:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
245:                 break;
246:             case "ORA":
247:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
248:                 break;
249:             case "PUR":
250:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
251:                 break;
252:             default:
253:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
254:                 break;
255:         }
256:     }
257: }
258:
259: private void llsFavourites_Hold(object sender,
System.Windows.Input.GestureEventArgs e)
260: {
261: }
262:
263: private void RemoveMenuItem_Click(object sender, RoutedEventArgs e)
264: {
265: }
266: }
267:
268: }
269: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.18 Male.cs

This is file Male.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu

```

```

10: {
11:     [Table]
12:     public class Male
13:     {
14:         [Column(IsPrimaryKey = true,
15:             IsDbGenerated = true,
16:             DbType = "INT NOT NULL Identity",
17:             CanBeNull = false)]
18:         public int MaleId { get; set; }
19:
20:         [Column]
21:         public string MaleName { get; set; }
22:
23:         [Column]
24:         public string MaleMeaning { get; set; }
25:     }
26: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.19 MyUpdate.cs

This is file MyUpdate.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu
10: {
11:     [Table]
12:     public class MyUpdate
13:     {
14:         [Column(IsPrimaryKey = true,
15:             IsDbGenerated = true,
16:             DbType = "INT NOT NULL Identity",
17:             CanBeNull = false)]
18:         public int MyUpdateId { get; set; }
19:
20:         [Column]
21:         public string MyUpdateName { get; set; }
22:     }
23: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.20 NameDataContext.cs

This is file NameDataContext.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu
10: {
11:     public class NameDataContext: DataContext
12:     {
13:         public const string ConnectionString = @"Data Source=isostore:/IsimSozlugu.sdf";
14:         public NameDataContext(string connectionString)
15:             : base(connectionString) { }
16:         public Table<Male> MaleNames;
17:         public Table<Female> FemaleNames;
18:         public Table<Unisex> UnisexNames;
19:         public Table<All> AllNames;
20:         public Table<Favourite> Favourites;
21:         public Table<MyUpdate> MyUpdates;
22:         public Table<AppSettings> AppSettings;
23:     }
24: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.21 NameDetailPage.xaml.cs

This is file NameDetailPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;
5: using System.Net;
6: using System.Text;
7: using System.Windows;
8: using System.Windows.Controls;
9: using System.Windows.Controls.Primitives;
10: using System.Windows.Input;
11: using System.Windows.Media;
12: using System.Windows.Media.Imaging;
13: using System.Windows.Navigation;
14: using Microsoft.Phone.Tasks;
15: using IsimSozlugu.Resources;
16: using Microsoft.Phone.Controls;
17: using Microsoft.Phone.Shell;
18:
19: namespace IsimSozlugu
20: {
21:     public partial class NameDetailPage : PhoneApplicationPage
22:     {
23:         public string pageName;
24:         double InputHeight = 0.0;
25:         public bool flag;
26:         public bool isFilled;
27:         public double ratingValue = 0;

```

```

28:         string gender;
29:         All allName = new All();
30:
31:         public NameDetailPage()
32:         {
33:             InitializeComponent();
34:
35:             ApplicationBar = new ApplicationBar();
36:
37:             ApplicationBarIconButton button1 = new ApplicationBarIconButton();
38:             button1.IconUri = new Uri("/Assets/Save.png", UriKind.Relative);
39:             button1.Text = "Kaydet";
40:             ApplicationBar.Buttons.Add(button1);
41:             button1.Click += new EventHandler(SaveButton_Click);
42:
43:             ApplicationBarIconButton button2 = new ApplicationBarIconButton();
44:             button2.IconUri = new Uri("/Assets/SendWithMail.png", UriKind.Relative);
45:             button2.Text = "E-Mail ile Gönder";
46:             ApplicationBar.Buttons.Add(button2);
47:             button2.Click += new EventHandler(SendMailButton_Click);
48:
49:             ApplicationBarIconButton button3 = new ApplicationBarIconButton();
50:             button3.IconUri = new Uri("/Assets/SendWithSMS.png", UriKind.Relative);
51:             button3.Text = "SMS ile Gönder";
52:             ApplicationBar.Buttons.Add(button3);
53:             button3.Click += new EventHandler(SendSMSButton_Click);
54:
55:             ApplicationBarIconButton button4 = new ApplicationBarIconButton();
56:             button4.IconUri = new Uri("/Assets/Share.png", UriKind.Relative);
57:             button4.Text = "Paylas";
58:             ApplicationBar.Buttons.Add(button4);
59:             button4.Click += new EventHandler(ShareNameButton_Click);
60:
61:             isFilled = false;
62:
63:             ApplicationBarMenuItem menuItem1 = new ApplicationBarMenuItem();
64:             menuItem1.Text = "Sil";
65:             ApplicationBar.MenuItems.Add(menuItem1);
66:             menuItem1.Click += new EventHandler>DeleteNameMenuItem_Click);
67:
68:             List<string> genderList = new List<string>();
69:             genderList.Add("Lütfen seçiniz");
70:             genderList.Add("Erkek");
71:             genderList.Add("Kadin");
72:             genderList.Add("Erkek-Kadin");
73:             lpGender.ItemsSource = genderList;
74:             lpGender.SelectedIndex = 0;
75:
76:             SetBackgroundColor();
77:         }
78:
79:         private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
80:         {
81:             if (pageName.Contains("/SearchPage.xaml"))
82:             {
83:                 //this.NavigationService.Navigate(new Uri("/SearchPage.xaml",
UriKind.Relative));
84:             }
85:             else
86:             {
87:                 this.NavigationService.Navigate(new Uri("/MainPage.xaml",
UriKind.Relative));
88:             }
89:         }
90:
91:         protected override void OnNavigatedTo(NavigationEventArgs e)
92:         {
93:             base.OnNavigatedTo(e);

```

```

94:
95:     var lastPage = NavigationService.BackStack.FirstOrDefault();
96:     pageName = lastPage.Source.ToString();
97:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
98:     {
99:         var appSettings = context.AppSettings.First();
100:
101:         FontFamily temp = new FontFamily(appSettings.FontFamily);
102:         double fontsize = double.Parse(appSettings.FontSize);
103:         txtName.FontFamily = temp;
104:         txtName.FontSize = fontsize;
105:         txtMeaning.FontFamily = temp;
106:         txtMeaning.FontSize = fontsize;
107:         lpGender.FontFamily = temp;
108:         lpGender.FontSize = fontsize;
109:     }
110:
111:     txtName.IsEnabled = true;
112:     isFilled = false;
113:     //SetBackgroundColor();
114:     //while (NavigationService.CanGoBack)
115:     //NavigationService.RemoveBackEntry();
116:
117: }
118:
119: protected override void OnNavigatedFrom(NavigationEventArgs e)
120: {
121:     base.OnNavigatedFrom(e);
122:     //while (NavigationService.CanGoBack)
123:     //NavigationService.RemoveBackEntry();
124:
125: }
126:
127: protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
128: {
129:     // displays "Fragment: Detail"
130:     //MessageBox.Show("Folder Id: " + e.Fragment);
131:     base.OnFragmentNavigation(e);
132:     string fragmentName = e.Fragment.ToString();
133:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
134:     {
135:         var name = context.AllNames.Where(j =>
136: j.AllName.Equals(fragmentName)).SingleOrDefault() as All;
137:         allName = name;
138:         for (int i = 0; i < lpGender.Items.Count; i++)
139:         {
140:             if (lpGender.Items[i].ToString() == name.AllGender)
141:             {
142:                 lpGender.SelectedIndex = i;
143:                 break;
144:             }
145:         }
146:         txtName.Text = name.AllName;
147:         txtName.IsEnabled = false;
148:         txtMeaning.Text = name.AllMeaning;
149:         var favourite = context.Favourites.Where(j =>
150: j.FavouriteName.Equals(fragmentName)).SingleOrDefault() as Favourite;
151:         if (favourite != null)
152:         {
153:             ApplicationBarItem menuItem4 = new ApplicationBarItem();
154:             menuItem4.Text = "Favorilerimden Çıkar";
155:             ApplicationBar.MenuItems.Add(menuItem4);
156:             menuItem4.Click += new
EventHandler(RemoveFavouritesMenuItem_Click);
157:         }
158:         else
159:         {
160:             ApplicationBarItem menuItem3 = new ApplicationBarItem();

```



```

160:         menuItem3.Text = "Favorilerime Ekle";
161:         ApplicationBar.MenuItems.Add(menuItem3);
162:         menuItem3.Click += new EventHandler(AddFavouritesMenuItem_Click);
163:     }
164:
165:     var myUpdate = context.MyUpdates.Where(j =>
j.MyUpdateName.Equals(fragmentName)).SingleOrDefault() as MyUpdate; ;
166:     if (myUpdate != null)
167:     {
168:         ApplicationBarMenuItem menuItem2 = new ApplicationBarMenuItem();
169:         menuItem2.Text = "Sisteme Eklenmesi İçin Gönder";
170:         ApplicationBar.MenuItems.Add(menuItem2);
171:         menuItem2.Click += new EventHandler(SaveAndSendMenuItem_Click);
172:     }
173:
174:
175:
176:     isFilled = true;
177:     pvName.SelectedIndex = 0;
178: }
179:
180:
181: private void SetBackgroundColor()
182: {
183:     AppSettings appSettings = new AppSettings();
184:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
185:     {
186:         appSettings = context.AppSettings.First() as AppSettings;
187:     }
188:
189:     if (appSettings.AppBackgroundImage != null)
190:     {
191:         MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
192:         BitmapImage image = new BitmapImage();
193:         image.SetSource(stream);
194:         ImageBrush ib = new ImageBrush();
195:         ib.ImageSource = image;
196:         this.LayoutRoot.Background = ib;
197:     }
198:     else
199:     {
200:         switch (appSettings.AppBackgroundColor)
201:         {
202:             case "BLA":
203:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
204:                 break;
205:             case "BLU":
206:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
207:                 break;
208:             case "BRO":
209:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
210:                 break;
211:             case "RED":
212:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
213:                 break;
214:             case "GRE":
215:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
216:                 break;
217:             case "GRA":
218:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
219:                 break;
220:             case "YEL":
221:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
222:                 break;
223:             case "ORA":
224:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
225:                 break;

```

```

226:             case "PUR":
227:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
228:                 break;
229:             default:
230:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
231:                 break;
232:         }
233:     }
234: }
235:
236: private void txtName_KeyDown(object sender, KeyEventArgs e)
237: {
238:     if (e.Key == Key.Enter)
239:     {
240:         pvName.SelectedIndex = 2;
241:         txtMeaning.Focus();
242:     }
243: }
244:
245: private void txtMeaning_TextChanged(object sender, TextChangedEventArgs e)
246: {
247:     Dispatcher.BeginInvoke(() =>
248:     {
249:         double CurrentInputHeight = txtMeaning.ActualHeight;
250:
251:         if (CurrentInputHeight > InputHeight)
252:         {
253:             svMeaning.ScrollToVerticalOffset(svMeaning.VerticalOffset +
CurrentInputHeight - InputHeight);
254:         }
255:
256:         InputHeight = CurrentInputHeight;
257:     });
258: }
259:
260: private void txtMeaning_GotFocus(object sender, RoutedEventArgs e)
261: {
262:     App.RootFrame.RenderTransform = new CompositeTransform();
263:     flag = true;
264: }
265:
266: private void txtMeaning_Tap(object sender,
System.Windows.Input.GestureEventArgs e)
267: {
268:     txtMeaning.Focus();
269:     //txtMeaning.Select(txtMeaning.Text.Length, 1);
270:     svMeaning.ScrollToVerticalOffset(e.GetPosition(txtMeaning).Y - 80);
271: }
272:
273: private void txtMeaning_LostFocus(object sender, RoutedEventArgs e)
274: {
275:     if (!flag) return;
276:     txtMeaning.Focus();
277:     flag = false;
278:     this.pnlKeyboardPlaceholder.Visibility = Visibility.Collapsed;
279: }
280:
281: private void txtMeaning_KeyDown(object sender, KeyEventArgs e)
282: {
283:     if (e.Key == Key.Enter)
284:     {
285:         svMeaning.ScrollToVerticalOffset(txtMeaning.ActualHeight);
286:     }
287: }
288:
289: private void svMeaning_GotFocus(object sender, RoutedEventArgs e)
290: {
291:     this.svMeaning.ScrollToVerticalOffset(this.txtMeaning.ActualHeight);

```

```

292:         this.svMeaning.UpdateLayout();
293:     }
294:
295:     private void SendSMSButton_Click(object sender, EventArgs e)
296:     {
297:         SmsComposeTask smsComposeTask = new SmsComposeTask();
298:
299:         smsComposeTask.To = "";
300:         smsComposeTask.Body = CreateSendMaterial();
301:
302:         smsComposeTask.Show();
303:         //MessageBox.Show(AppResources.SuccessfulSendWithSMS);
304:     }
305:
306:     private void ShareNameButton_Click(object sender, EventArgs e)
307:     {
308:         ShareStatusTask shareStatusTask = new ShareStatusTask();
309:
310:         shareStatusTask.Status = CreateSendMaterial();
311:
312:         shareStatusTask.Show();
313:         //MessageBox.Show(AppResources.SuccessfulSendWithSMS);
314:     }
315:
316:     private void SendMailButton_Click(object sender, EventArgs e)
317:     {
318:         // burada birden fazla e-posta hesabi varsa birini seçmesi söyleniyor
319:         //EmailAddressChooserTask emailAddressChooserTask;
320:         //emailAddressChooserTask = new EmailAddressChooserTask();
321:         //emailAddressChooserTask.Completed += new
EventHandler<EmailResult>(emailAddressChooserTask_Completed);
322:         //emailAddressChooserTask.Show();
323:
324:         EmailComposeTask emailComposeTask = new EmailComposeTask();
325:
326:         emailComposeTask.Subject = "Isim ve Anlami (" + txtName.Text + ")";
327:         emailComposeTask.Body = CreateSendMaterial();
328:         emailComposeTask.To = "";
329:         emailComposeTask.Cc = "";
330:         emailComposeTask.Bcc = "";
331:
332:         emailComposeTask.Show();
333:         //MessageBox.Show(AppResources.SuccessfulSendWithMail);
334:     }
335:
336:     private string CreateSendMaterial()
337:     {
338:         StringBuilder sb = new StringBuilder();
339:         sb.AppendLine("Isim: " + txtName.Text);
340:         sb.AppendLine("Anlami: " + txtMeaning.Text);
341:         sb.AppendLine("Cinsiyeti: " + lpGender.Items[lpGender.SelectedIndex]);
342:         sb.AppendLine();
343:         sb.AppendLine();
344:         sb.AppendLine("Isim Sözlüğü Uygulaması ile Gönderildi");
345:         return sb.ToString();
346:     }
347:
348:     private void AddFavouritesMenuItem_Click(object sender, EventArgs e)
349:     {
350:         using (var context = new NameDataContext(NameDataContext.ConnectionString))
351:         {
352:             var favourites = context.Favourites.Where(j =>
j.FavouriteName.Equals(txtName.Text)).ToList() as List<Favourite>;
353:             if (favourites.Count > 0)
354:             {
355:                 MessageBox.Show("Bu Isim Zaten Favorileriniz Arasında.");
356:             }
357:             else
358:             {

```

```

359:             Favourite favourite = new Favourite();
360:             favourite.FavouriteName = txtName.Text;
361:             context.Favourites.InsertOnSubmit(favourite);
362:             context.SubmitChanges();
363:             MessageBox.Show("İsim, Favorilerinize Basariyla Eklendi");
364:         }
365:     }
366: }
367:
368: private void RemoveFavouritesMenuItem_Click(object sender, EventArgs e)
369: {
370:     if (MessageBox.Show("İsmi Favorilerinizden Çıkarmak İstediginize Emin
Misiniz?",
371:         "Favorilerden Çıkar", MessageBoxButton.OKCancel)
372:         != MessageBoxResult.OK)
373:     {
374:     }
375:     else
376:     {
377:         using (var context = new
NameDataContext(NameDataContext.ConnectionString))
378:         {
379:             var favourite = context.Favourites.Where(j =>
j.FavouriteName.Equals(txtName.Text)).ToList() as List<Favourite>;
380:             context.Favourites.DeleteAllOnSubmit(favourite);
381:             context.SubmitChanges();
382:             MessageBox.Show("İsim, Favorilerden çıkarildi");
383:         }
384:     }
385: }
386:
387: private void SaveAndSendMenuItem_Click(object sender, EventArgs e)
388: {
389:     EmailComposeTask emailComposeTask = new EmailComposeTask();
390:
391:     emailComposeTask.Subject = "İsim ve Anlami (" + txtName.Text + ")";
392:     emailComposeTask.Body = CreateSendMaterial();
393:     emailComposeTask.To = "coderserdar@outlook.com";
394:     emailComposeTask.Cc = "";
395:     emailComposeTask.Bcc = "";
396:
397:     emailComposeTask.Show();
398: }
399:
400: private void DeleteNameMenuItem_Click(object sender, EventArgs e)
401: {
402:     if (isFilled == true)
403:     {
404:         if (MessageBox.Show("İsmi Silmek İstediginize Emin Misiniz?",
405:             "İsmi Sil", MessageBoxButton.OKCancel)
406:             != MessageBoxResult.OK)
407:         {
408:         }
409:         else
410:         {
411:             using (var context = new
NameDataContext(NameDataContext.ConnectionString))
412:             {
413:
414:                 switch (allName.AllGender)
415:                 {
416:                     case "Erkek":
417:                         var maleName = context.MaleNames.Where(j =>
j.MaleName.Equals(allName.AllName)).SingleOrDefault() as Male;
418:                         context.MaleNames.DeleteOnSubmit(maleName);
419:                         break;
420:                     case "Kadin":
421:                         var femaleName = context.FemaleNames.Where(j =>
j.FemaleName.Equals(allName.AllName)).SingleOrDefault() as Female;

```

```

422:             context.FemaleNames.DeleteOnSubmit(femaleName);
423:             break;
424:             case "Erkek-Kadin":
425:                 var unisexName = context.UnisexNames.Where(j =>
j.UnisexName.Equals(allName.AllName)).SingleOrDefault() as Unisex;
426:                 context.UnisexNames.DeleteOnSubmit(unisexName);
427:                 break;
428:             default:
429:                 break;
430:         }
431:
432:         var allNames = context.AllNames.Where(j =>
j.AllName.Equals(allName.AllName)).SingleOrDefault() as All;
433:         context.AllNames.DeleteOnSubmit(allNames);
434:
435:         var myUpdates = context.MyUpdates.Where(j =>
j.MyUpdateName.Equals(allName.AllName)).SingleOrDefault() as MyUpdate;
436:         if (myUpdates != null)
437:         {
438:             context.MyUpdates.DeleteOnSubmit(myUpdates);
439:         }
440:
441:         var favourites = context.Favourites.Where(j =>
j.FavouriteName.Equals(allName.AllName)).SingleOrDefault() as Favourite;
442:         if (favourites != null)
443:         {
444:             context.Favourites.DeleteOnSubmit(favourites);
445:         }
446:
447:         context.SubmitChanges();
448:         MessageBox.Show("İsim, Sistemden Silindi");
449:     }
450: }
451: }
452: this.NavigationService.Navigate(new Uri("/MainPage.xaml",
UriKind.Relative));
453: }
454:
455: private void SaveButton_Click(object sender, EventArgs e)
456: {
457:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
458:     {
459:         if (isFilled == false)
460:         {
461:             if (lpGender.SelectedIndex < 1 ||
txtName.Text.TrimEnd().TrimStart().Length < 1 ||
txtMeaning.Text.TrimEnd().TrimStart().Length < 1)
462:             {
463:                 MessageBox.Show("Bütün Alanlari Dolduraniz Gerekmektedir.");
464:             }
465:             else
466:             {
467:                 var allName = context.AllNames.Where(j =>
j.AllName.ToLower().Equals(txtName.Text.TrimEnd().TrimStart().ToLower())).SingleOrDefault()
as All;
468:                 if (allName != null)
469:                 {
470:                     MessageBox.Show("Bu İsim Sistemde Mevcut");
471:                 }
472:                 else
473:                 {
474:
475:                     switch (lpGender.SelectedIndex)
476:                     {
477:                         case 1:
478:                             Male maleName = new Male();
479:                             maleName.MaleName = txtName.Text.ToUpper();
480:                             maleName.MaleMeaning = txtMeaning.Text;
481:                             context.MaleNames.InsertOnSubmit(maleName);

```

```

482:         break;
483:     case 2:
484:         Female femaleName = new Female();
485:         femaleName.FemaleName = txtName.Text.ToUpper();
486:         femaleName.FemaleMeaning = txtMeaning.Text;
487:         context.FemaleNames.InsertOnSubmit(femaleName);
488:         break;
489:     case 3:
490:         Unisex unisexName = new Unisex();
491:         unisexName.UnisexName = txtName.Text.ToUpper();
492:         unisexName.UnisexMeaning = txtMeaning.Text;
493:         context.UnisexNames.InsertOnSubmit(unisexName);
494:         break;
495:     default:
496:         break;
497:     }
498:
499:     All allNameTemp = new All();
500:     allNameTemp.AllName = txtName.Text.ToUpper();
501:     allNameTemp.AllMeaning = txtMeaning.Text;
502:     allNameTemp.AllNameMeaning = txtName.Text + " " +
txtMeaning.Text;
503:     allNameTemp.AllGender =
lpGender.Items[lpGender.SelectedIndex].ToString();
504:     context.AllNames.InsertOnSubmit(allNameTemp);
505:
506:     MyUpdate myUpdate = new MyUpdate();
507:     myUpdate.MyUpdateName = txtName.Text.ToUpper();
508:     context.MyUpdates.InsertOnSubmit(myUpdate);
509:
510:     context.SubmitChanges();
511:     MessageBox.Show("İsim, Sisteme Başarılı Bir Şekilde
Eklendi");
512:
513:     }
514: }
515: }
516: else
517: {
518:     if (allName.AllGender ==
lpGender.Items[lpGender.SelectedIndex].ToString() && allName.AllMeaning ==
txtMeaning.Text.TrimEnd().TrimStart())
519:     {
520:         MessageBox.Show("Herhangi bir değişiklik yapmadınız.");
521:     }
522:     else
523:     {
524:         // önce eski kayıt siliniyor
525:         switch (allName.AllGender)
526:         {
527:             case "Erkek":
528:                 var maleName = context.MaleNames.Where(j =>
j.MaleName.Equals(allName.AllName)).SingleOrDefault() as Male;
529:                 context.MaleNames.DeleteOnSubmit(maleName);
530:                 break;
531:             case "Kadin":
532:                 var femaleName = context.FemaleNames.Where(j =>
j.FemaleName.Equals(allName.AllName)).SingleOrDefault() as Female;
533:                 context.FemaleNames.DeleteOnSubmit(femaleName);
534:                 break;
535:             case "Erkek-Kadin":
536:                 var unisexName = context.UnisexNames.Where(j =>
j.UnisexName.Equals(allName.AllName)).SingleOrDefault() as Unisex;
537:                 context.UnisexNames.DeleteOnSubmit(unisexName);
538:                 break;
539:             default:
540:                 break;
541:         }
542:     }

```

```

543:         // sonrasında yeni kayıt ekleniyor
544:         switch (lpGender.SelectedIndex)
545:         {
546:             case 1:
547:                 Male maleName = new Male();
548:                 maleName.MaleName = txtName.Text.ToUpper();
549:                 maleName.MaleMeaning = txtMeaning.Text;
550:                 context.MaleNames.InsertOnSubmit(maleName);
551:                 break;
552:             case 2:
553:                 Female femaleName = new Female();
554:                 femaleName.FemaleName = txtName.Text.ToUpper();
555:                 femaleName.FemaleMeaning = txtMeaning.Text;
556:                 context.FemaleNames.InsertOnSubmit(femaleName);
557:                 break;
558:             case 3:
559:                 Unisex unisexName = new Unisex();
560:                 unisexName.UnisexName = txtName.Text.ToUpper();
561:                 unisexName.UnisexMeaning = txtMeaning.Text;
562:                 context.UnisexNames.InsertOnSubmit(unisexName);
563:                 break;
564:             default:
565:                 break;
566:         }
567:
568:         var all = context.AllNames.Where(j =>
569:         j.AllName.Equals(allName.AllName)).Select(j => j);
570:         foreach (var item in all)
571:         {
572:             item.AllGender =
573:             lpGender.Items[lpGender.SelectedIndex].ToString();
574:             item.AllMeaning = txtMeaning.Text.TrimEnd().TrimStart();
575:             item.AllNameMeaning = item.AllName + " " + item.AllMeaning;
576:         }
577:         var myUpdateTemp = context.MyUpdates.Where(j =>
578:         j.MyUpdateName.Equals(allName.AllName)).SingleOrDefault() as MyUpdate;
579:         if (myUpdateTemp == null)
580:         {
581:             MyUpdate myUpdate = new MyUpdate();
582:             myUpdate.MyUpdateName = allName.AllName;
583:             context.MyUpdates.InsertOnSubmit(myUpdate);
584:         }
585:         context.SubmitChanges();
586:         MessageBox.Show("İsim Bilgisi Başarılı Bir Şekilde
587: Güncellendi");
588:     }
589:     this.NavigationService.Navigate(new Uri("/MainPage.xaml",
590: UriKind.Relative));
591: }
592: private void lpGender_SelectionChanged(object sender,
593: SelectionChangedEventArgs e)
594: {
595:     if (lpGender.SelectedIndex != 0)
596:     {
597:         gender = lpGender.Items[lpGender.SelectedIndex].ToString();
598:         pvName.SelectedIndex = 1;
599:         txtName.Focus();
600:     }
601: }
602: }
603: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.22 SearchPage.xaml.cs

This is file SearchPage.xaml.cs.

Body Source

```

1: ?using System;
2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;
5: using System.Net;
6: using System.Text;
7: using System.Windows;
8: using System.Windows.Controls;
9: using System.Windows.Controls.Primitives;
10: using System.Windows.Input;
11: using System.Windows.Media;
12: using System.Windows.Media.Imaging;
13: using System.Windows.Navigation;
14: using Microsoft.Phone.Tasks;
15: using IsimSozlugu.Resources;
16: using Microsoft.Phone.Controls;
17: using Microsoft.Phone.Shell;
18:
19: namespace IsimSozlugu
20: {
21:     public partial class SearchPage : PhoneApplicationPage
22:     {
23:         public SearchPage()
24:         {
25:             InitializeComponent();
26:             SetBackgroundColor();
27:
28:             txtSearchResult.Text = "Arama Sonuçlari";
29:             //txtSearchResult.Text = AppResources.SearchResults;
30:             //lblSearch.Text = AppResources.Search;
31:             //btnSearch.Content = AppResources.Search;
32:             //lstSearch.SelectedIndex = -1;
33:         }
34:
35:         private void SetBackgroundColor()
36:         {
37:             AppSettings appSettings = new AppSettings();
38:             using (var context = new NameDataContext(NameDataContext.ConnectionString))
39:             {
40:                 appSettings = context.AppSettings.First() as AppSettings;
41:             }
42:
43:             if (appSettings.AppBackgroundImage != null)
44:             {
45:                 MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
46:                 BitmapImage image = new BitmapImage();
47:                 image.SetSource(stream);
48:                 ImageBrush ib = new ImageBrush();
49:                 ib.ImageSource = image;
50:                 this.LayoutRoot.Background = ib;
51:             }
52:             else
53:             {

```



```

54:         switch (appSettings.AppBackgroundColor)
55:         {
56:             case "BLA":
57:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
58:                 break;
59:             case "BLU":
60:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
61:                 break;
62:             case "BRO":
63:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
64:                 break;
65:             case "RED":
66:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
67:                 break;
68:             case "GRE":
69:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
70:                 break;
71:             case "GRA":
72:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
73:                 break;
74:             case "YEL":
75:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
76:                 break;
77:             case "ORA":
78:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
79:                 break;
80:             case "PUR":
81:                 this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
82:                 break;
83:             default:
84:                 this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
85:                 break;
86:         }
87:     }
88: }
89:
90: private void btnSearch_Click(object sender, RoutedEventArgs e)
91: {
92:     lstSearch.Items.Clear();
93:     if (txtSearch.Text.TrimStart().Length < 1)
94:     {
95:         MessageBox.Show("Arama Alanına En Az Bir Harf Yazmak Zorundasınız");
96:     }
97:     else
98:     {
99:         using (var context = new
NameDataContext(NameDataContext.ConnectionString))
100:         {
101:             var nameList =
102:                 context.AllNames.Where(j =>
j.AllNameMeaning.ToLower().Contains(txtSearch.Text.ToLower())).ToList() as List<All>;
103:             //var noteList = context.Notes.ToList() as List<Note>;
104:
105:             if (nameList != null)
106:             {
107:                 txtSearchResult.Text = "Arama Sonuçları" + " (" +
nameList.Count() + ")";
108:             }
109:
110:             //List<All> allNames = nameList.OrderBy(j => j.AllName).ToList()
as List<All>;
111:             //List<AlphaKeyGroup<All>> DataSource =
AlphaKeyGroup<All>.CreateGroups(allNames,
112:                 // System.Threading.Thread.CurrentThread.CurrentUICulture,
113:                 // (All a) => { return a.AllName; }, true);
114:             //llsAllNames.ItemsSource = DataSource;

```

```

115:
116:         var nameList2 = nameList.OrderBy(j => j.AllName).ToList() as
List<All>;
117:         //lstSearch.ItemsSource = nameList;
118:         for (int i = 0; i < nameList2.Count; i++)
119:         {
120:             lstSearch.Items.Add(nameList2[i] as All);
121:         }
122:         //lstSearch.ItemTemplate.
123:         //lstSearch.DisplayMemberPath = "NoteName" + " (" + "CreationDate"
+ ")";
124:         lstSearch.DisplayMemberPath = "AllName";
125:         MessageBox.Show("Arama Tamamlandi");
126:     }
127: }
128: }
129:
130:     private void lstSearch_SelectionChanged(object sender,
SelectionChangedEventArgs e)
131:     {
132:         try
133:         {
134:             if (lstSearch.SelectedIndex != -1)
135:             {
136:                 All selectedName = lstSearch.SelectedItem as All;
137:                 NavigationService.Navigate(new Uri("/NameDetailPage.xaml#" +
selectedName.AllName, UriKind.Relative));
138:                 lstSearch.SelectedIndex = -1;
139:             }
140:         }
141:     }
142:     catch (Exception)
143:     {
144:         MessageBox.Show("Sistemde Bir Hata Olustur. Lütfen Sonra Tekrar
Deneyin");
145:     }
146: }
147:
148:     protected override void OnNavigatedTo(NavigationEventArgs e)
149:     {
150:         base.OnNavigatedTo(e);
151:     }
152:     protected override void OnNavigatedFrom(NavigationEventArgs e)
153:     {
154:         base.OnNavigatedFrom(e);
155:     }
156:
157:     private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
158:     {
159:         if (this.NavigationService.CanGoBack)
160:         {
161:             this.NavigationService.Navigate(new Uri("/MainPage.xaml",
UriKind.Relative));
162:         }
163:     }
164:
165:     private void txtSearch_KeyDown(object sender,
System.Windows.Input.KeyEventArgs e)
166:     {
167:         if (e.Key == Key.Enter)
168:         {
169:             if (txtSearch.Text.TrimStart().Length < 1)
170:             {
171:                 MessageBox.Show("Arama Alanina En Az Bir Harf Yazmak
Zorundasiniz");
172:             }
173:             else
174:             {

```

```

175:                lstSearch.Items.Clear();
176:                using (var context = new
NameDataContext(NameDataContext.ConnectionString))
177:                {
178:                    var nameList =
179:                        context.AllNames.Where(j =>
j.AllNameMeaning.ToLower().Contains(txtSearch.Text.ToLower())).ToList() as List<All>;
180:                    //var noteList = context.Notes.ToList() as List<Note>;
181:
182:                    if (nameList != null)
183:                    {
184:                        txtSearchResult.Text = "Arama Sonuçlari" + " (" +
nameList.Count() + ")";
185:                    }
186:
187:                    //List<All> allNames = nameList.OrderBy(j =>
j.AllName).ToList() as List<All>;
188:                    //List<AlphaKeyGroup<All>> DataSource =
AlphaKeyGroup<All>.CreateGroups(allNames,
189:                        // System.Threading.Thread.CurrentThread.CurrentUICulture,
190:                        // (All a) => { return a.AllName; }, true);
191:                    //llsAllNames.ItemsSource = DataSource;
192:
193:                    var nameList2 = nameList.OrderBy(j => j.AllName).ToList() as
List<All>;
194:                    //lstSearch.ItemsSource = noteList;
195:                    for (int i = 0; i < nameList2.Count; i++)
196:                    {
197:                        // if
(nameList2[i].NameDescriptionWithoutNewline.ToLower(Thread.CurrentThread.CurrentCulture).Ind
exOf(txtSearch.Text.ToLower(Thread.CurrentThread.CurrentCulture))
!= -1)
198:                        // {
199:                        lstSearch.Items.Add(nameList2[i] as All);
200:                        // }
201:                    }
202:                    //lstSearch.ItemTemplate.
203:                    //lstSearch.DisplayMemberPath = "NoteName" + " (" +
"CreationDate" + ")";
204:                    lstSearch.DisplayMemberPath = "AllName";
205:                    MessageBox.Show("Arama Tamamlandi");
206:                }
207:            }
208:        }
209:    }
210:
211:    private void PhoneApplicationPage_Loaded(object sender, RoutedEventArgs e)
212:    {
213:        txtSearch.Focus();
214:    }
215: }
216: }

```

Namespaces

Name	Description
IsimSozlugu (see page 1)	This is namespace IsimSozlugu.

1.2.23 StatisticsPage.xaml.cs

This is file StatisticsPage.xaml.cs.

Body Source

```
1: ?using System;
```

```

2: using System.Collections.Generic;
3: using System.IO;
4: using System.Linq;
5: using System.Net;
6: using System.Text;
7: using System.Windows;
8: using System.Windows.Controls;
9: using System.Windows.Controls.Primitives;
10: using System.Windows.Input;
11: using System.Windows.Media;
12: using System.Windows.Media.Imaging;
13: using System.Windows.Navigation;
14: using Microsoft.Phone.Tasks;
15: using IsimSozlugu.Resources;
16: using Microsoft.Phone.Controls;
17: using Microsoft.Phone.Shell;
18:
19: namespace IsimSozlugu
20: {
21:     public partial class StatisticsPage : PhoneApplicationPage
22:     {
23:         public StatisticsPage()
24:         {
25:             InitializeComponent();
26:             SetBackgroundColor();
27:             SetStatistic();
28:         }
29:
30:         private void PhoneApplicationPage_BackKeyPress(object sender,
System.ComponentModel.CancelEventArgs e)
31:         {
32:             if (this.NavigationService.CanGoBack)
33:             {
34:                 this.NavigationService.Navigate(new Uri("/MainPage.xaml",
UriKind.Relative));
35:             }
36:         }
37:
38:         private void SetBackgroundColor()
39:         {
40:             AppSettings appSettings = new AppSettings();
41:             using (var context = new NameDataContext(NameDataContext.ConnectionString))
42:             {
43:                 appSettings = context.AppSettings.First() as AppSettings;
44:             }
45:
46:             if (appSettings.AppBackgroundImage != null)
47:             {
48:                 MemoryStream stream = new MemoryStream(appSettings.AppBackgroundImage);
49:                 BitmapImage image = new BitmapImage();
50:                 image.SetSource(stream);
51:                 ImageBrush ib = new ImageBrush();
52:                 ib.ImageSource = image;
53:                 this.LayoutRoot.Background = ib;
54:             }
55:             else
56:             {
57:                 switch (appSettings.AppBackgroundColor)
58:                 {
59:                     case "BLA":
60:                         this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
61:                         break;
62:                     case "BLU":
63:                         this.LayoutRoot.Background = new SolidColorBrush(Colors.Blue);
64:                         break;
65:                     case "BRO":
66:                         this.LayoutRoot.Background = new SolidColorBrush(Colors.Brown);
67:                         break;
68:                     case "RED":

```

```

69:             this.LayoutRoot.Background = new SolidColorBrush(Colors.Red);
70:             break;
71:         case "GRE":
72:             this.LayoutRoot.Background = new SolidColorBrush(Colors.Green);
73:             break;
74:         case "GRA":
75:             this.LayoutRoot.Background = new SolidColorBrush(Colors.Gray);
76:             break;
77:         case "YEL":
78:             this.LayoutRoot.Background = new
SolidColorBrush(Colors.Yellow);
79:             break;
80:         case "ORA":
81:             this.LayoutRoot.Background = new
SolidColorBrush(Colors.Orange);
82:             break;
83:         case "PUR":
84:             this.LayoutRoot.Background = new
SolidColorBrush(Colors.Purple);
85:             break;
86:         default:
87:             this.LayoutRoot.Background = new SolidColorBrush(Colors.Black);
88:             break;
89:     }
90: }
91: }
92:
93: protected override void OnNavigatedTo(NavigationEventArgs e)
94: {
95:     base.OnNavigatedTo(e);
96: }
97:
98: protected override void OnNavigatedFrom(NavigationEventArgs e)
99: {
100:     base.OnNavigatedFrom(e);
101:     //while (NavigationService.CanGoBack)
102:     //NavigationService.RemoveBackEntry();
103: }
104: }
105:
106: private void SetStatistic()
107: {
108:     StringBuilder sb = new StringBuilder();
109:     int male, female, unisex, all, update, favourite;
110:
111:     using (var context = new NameDataContext(NameDataContext.ConnectionString))
112:     {
113:         male = context.MaleNames.ToList().Count();
114:         female = context.FemaleNames.ToList().Count();
115:         unisex = context.UnisexNames.ToList().Count();
116:         all = context.AllNames.ToList().Count();
117:         favourite = context.Favourites.ToList().Count();
118:         update = context.MyUpdates.ToList().Count();
119:     }
120:
121:     sb.AppendLine("Erkek Ismi Sayisi: " + male);
122:     sb.AppendLine("Kadin Ismi Sayisi: " + female);
123:     sb.AppendLine("Unisex Isim Sayisi: " + unisex);
124:     sb.AppendLine("Favori Isim Sayisi: " + favourite);
125:     sb.AppendLine("Düzenlenen Isim Sayisi: " + update);
126:     sb.AppendLine("Toplam Isim Sayisi: " + all);
127:
128:     txtStatistics.Text = sb.ToString();
129:     txtStatistics.IsReadOnly = true;
130: }
131: }
132: }

```

Namespaces

Name	Description
IsimSozlugu (🔗 see page 1)	This is namespace IsimSozlugu.

1.2.24 Unisex.cs

This is file Unisex.cs.

Body Source

```
1: ?using System;
2: using System.Collections.Generic;
3: using System.Data.Linq;
4: using System.Data.Linq.Mapping;
5: using System.Linq;
6: using System.Text;
7: using System.Threading.Tasks;
8:
9: namespace IsimSozlugu
10: {
11:     [Table]
12:     public class Unisex
13:     {
14:         [Column(IsPrimaryKey = true,
15:             IsDbGenerated = true,
16:             DbType = "INT NOT NULL Identity",
17:             CanBeNull = false)]
18:         public int UnisexId { get; set; }
19:
20:         [Column]
21:         public string UnisexName { get; set; }
22:
23:         [Column]
24:         public string UnisexMeaning { get; set; }
25:
26:     }
27: }
```

Namespaces

Name	Description
IsimSozlugu (🔗 see page 1)	This is namespace IsimSozlugu.

Index

A

AboutPage class 4
 about AboutPage class 4
 AboutPage 5
 AboutPage.xaml.cs 37
 All class 6
 about All class 6
 AllGender 6
 AllId 6
 AllMeaning 6
 AllName 7
 AllNameMeaning 7
 All.cs 40
 AlphaKeyGroup class 7
 about AlphaKeyGroup class 7
 AlphaKeyGroup 8
 CreateGroups 8
 GetKeyDelegate 9
 Key 9
 AlphaKeyGroup.cs 41
 App class 9
 about App class 9
 App 10
 ReadFile 11
 RootFrame 11
 App.xaml.cs 42
 AppResources class 2
 about AppResources class 2
 AppBarButtonText 3
 AppBarMenuItemText 3
 ApplicationTitle 3
 AppResources 3
 Culture 3
 ResourceFlowDirection 3
 ResourceLanguage 3
 ResourceManager 4
 AppResources.Designer.cs 49
 AppSettings class 11

 about AppSettings class 11
 AppBackgroundColor 12
 AppBackgroundImage 12
 AppLangName 12
 AppSettingsId 12
 FontFamily 13
 FontSize 13
 AppSettings.cs 52
 AssemblyInfo.cs 52

B

BackgroundColorSettingsPage class 13
 about BackgroundColorSettingsPage class 13
 artistId 14
 BackgroundColorSettingsPage 14
 OnFragmentNavigation 14
 OnNavigatedFrom 15
 OnNavigatedTo 15
 BackgroundColorSettingsPage.xaml.cs 53

F

Favourite class 15
 about Favourite class 15
 FavouriteId 16
 FavouriteName 16
 Favourite.cs 56
 Female class 16
 about Female class 16
 FemaleId 16
 FemaleMeaning 17
 FemaleName 17
 Female.cs 57
 Files 37
 FontFamilySettingsPage class 17
 about FontFamilySettingsPage class 17
 artistId 18
 FontFamilySettingsPage 17
 OnFragmentNavigation 18
 OnNavigatedFrom 19
 OnNavigatedTo 19

FontFamilySettingsPage.xaml.cs 57

FontSizeSettingsPage class 19

about FontSizeSettingsPage class 19

artistId 20

FontSizeSettingsPage 20

OnFragmentNavigation 20

OnNavigatedFrom 21

OnNavigatedTo 21

FontSizeSettingsPage.xaml.cs 60

G

GeneralSettingsPage class 21

about GeneralSettingsPage class 21

GeneralSettingsPage 22

OnNavigatedFrom 23

OnNavigatedTo 23

GeneralSettingsPage.xaml.cs 62

I

IsimSozlugu 1

IsimSozlugu namespace 1

Classes 4

IsimSozlugu.AboutPage 4

IsimSozlugu.AboutPage.AboutPage 5

IsimSozlugu.All 6

IsimSozlugu.All.AllGender 6

IsimSozlugu.All.AllId 6

IsimSozlugu.All.AllMeaning 6

IsimSozlugu.All.AllName 7

IsimSozlugu.All.AllNameMeaning 7

IsimSozlugu.AlphaKeyGroup 7

IsimSozlugu.AlphaKeyGroup.AlphaKeyGroup 8

IsimSozlugu.AlphaKeyGroup.CreateGroups 8

IsimSozlugu.AlphaKeyGroup.GetKeyDelegate 9

IsimSozlugu.AlphaKeyGroup.Key 9

IsimSozlugu.App 9

IsimSozlugu.App.App 10

IsimSozlugu.App.ReadFile 11

IsimSozlugu.App.RootFrame 11

IsimSozlugu.AppSettings 11

IsimSozlugu.AppSettings.AppBackgroundColor 12

IsimSozlugu.AppSettings.AppBackgroundImage 12

IsimSozlugu.AppSettings.AppLangName 12

IsimSozlugu.AppSettings.AppSettingsId 12

IsimSozlugu.AppSettings.FontFamily 13

IsimSozlugu.AppSettings.FontSize 13

IsimSozlugu.BackgroundColorSettingsPage 13

IsimSozlugu.BackgroundColorSettingsPage.artistId 14

IsimSozlugu.BackgroundColorSettingsPage.BackgroundColorSettingsPage 14

IsimSozlugu.BackgroundColorSettingsPage.OnFragmentNavigation 14

IsimSozlugu.BackgroundColorSettingsPage.OnNavigatedFrom 15

IsimSozlugu.BackgroundColorSettingsPage.OnNavigatedTo 15

IsimSozlugu.csproj 67

IsimSozlugu.Favourite 15

IsimSozlugu.Favourite.FavouriteId 16

IsimSozlugu.Favourite.FavouriteName 16

IsimSozlugu.Female 16

IsimSozlugu.Female.FemaleId 16

IsimSozlugu.Female.FemaleMeaning 17

IsimSozlugu.Female.FemaleName 17

IsimSozlugu.FontFamilySettingsPage 17

IsimSozlugu.FontFamilySettingsPage.artistId 18

IsimSozlugu.FontFamilySettingsPage.FontFamilySettingsPage 17

IsimSozlugu.FontFamilySettingsPage.OnFragmentNavigation 18

IsimSozlugu.FontFamilySettingsPage.OnNavigatedFrom 19

IsimSozlugu.FontFamilySettingsPage.OnNavigatedTo 19

IsimSozlugu.FontSizeSettingsPage 19

IsimSozlugu.FontSizeSettingsPage.artistId 20

IsimSozlugu.FontSizeSettingsPage.FontSizeSettingsPage 20

IsimSozlugu.FontSizeSettingsPage.OnFragmentNavigation 20

IsimSozlugu.FontSizeSettingsPage.OnNavigatedFrom 21

IsimSozlugu.FontSizeSettingsPage.OnNavigatedTo 21

IsimSozlugu.GeneralSettingsPage 21

IsimSozlugu.GeneralSettingsPage.GeneralSettingsPage 22

IsimSozlugu.GeneralSettingsPage.OnNavigatedFrom 23	IsimSozlugu.Resources.AppResources.Culture 3
IsimSozlugu.GeneralSettingsPage.OnNavigatedTo 23	IsimSozlugu.Resources.AppResources.ResourceFlowDirection 3
IsimSozlugu.LocalizedStrings 23	IsimSozlugu.Resources.AppResources.ResourceLanguage 3
IsimSozlugu.LocalizedStrings.LocalizedResources 24	IsimSozlugu.Resources.AppResources.ResourceManager 4
IsimSozlugu.MainPage 24	IsimSozlugu.SearchPage 33
IsimSozlugu.MainPage.MainPage 24	IsimSozlugu.SearchPage.OnNavigatedFrom 34
IsimSozlugu.Male 25	IsimSozlugu.SearchPage.OnNavigatedTo 34
IsimSozlugu.Male.MaleId 25	IsimSozlugu.SearchPage.SearchPage 33
IsimSozlugu.Male.MaleMeaning 26	IsimSozlugu.sln 67
IsimSozlugu.Male.MaleName 26	IsimSozlugu.StatisticsPage 34
IsimSozlugu.MyUpdate 26	IsimSozlugu.StatisticsPage.OnNavigatedFrom 35
IsimSozlugu.MyUpdate.MyUpdateId 26	IsimSozlugu.StatisticsPage.OnNavigatedTo 35
IsimSozlugu.MyUpdate.MyUpdateName 27	IsimSozlugu.StatisticsPage.StatisticsPage 35
IsimSozlugu.NameDataContext 27	IsimSozlugu.Unisex 36
IsimSozlugu.NameDataContext.AllNames 28	IsimSozlugu.Unisex.UnisexId 36
IsimSozlugu.NameDataContext.AppSettings 28	IsimSozlugu.Unisex.UnisexMeaning 36
IsimSozlugu.NameDataContext.ConnectionString 28	IsimSozlugu.Unisex.UnisexName 36
IsimSozlugu.NameDataContext.Favourites 28	
IsimSozlugu.NameDataContext.FemaleNames 28	
IsimSozlugu.NameDataContext.MaleNames 28	
IsimSozlugu.NameDataContext.MyUpdates 28	
IsimSozlugu.NameDataContext.NameDataContext 27	
IsimSozlugu.NameDataContext.UnisexNames 29	
IsimSozlugu.NameDetailPage 29	
IsimSozlugu.NameDetailPage.flag 30	
IsimSozlugu.NameDetailPage.isFilled 31	
IsimSozlugu.NameDetailPage.NameDetailPage 29	
IsimSozlugu.NameDetailPage.OnFragmentNavigation 31	
IsimSozlugu.NameDetailPage.OnNavigatedFrom 32	
IsimSozlugu.NameDetailPage.OnNavigatedTo 32	
IsimSozlugu.NameDetailPage.pageName 31	
IsimSozlugu.NameDetailPage.ratingValue 31	
IsimSozlugu.Resources 1	
IsimSozlugu.Resources namespace 1	
Classes 2	
IsimSozlugu.Resources.AppResources 2	
IsimSozlugu.Resources.AppResources.AppBarButtonText 3	
IsimSozlugu.Resources.AppResources.AppBarMenuItemText 3	
IsimSozlugu.Resources.AppResources.ApplicationTitle 3	
IsimSozlugu.Resources.AppResources.AppResources 3	
	LocalizedStrings class 23
	about LocalizedStrings class 23
	LocalizedResources 24
	LocalizedStrings.cs 67
	L
	M
	MainPage class 24
	about MainPage class 24
	MainPage 24
	MainPage.xaml.cs 68
	Male class 25
	about Male class 25
	MaleId 25
	MaleMeaning 26
	MaleName 26
	Male.cs 72
	MyUpdate class 26
	about MyUpdate class 26
	MyUpdateId 26
	MyUpdateName 27

MyUpdate.cs 73

N

NameDataContext class 27

about NameDataContext class 27

AllNames 28

AppSettings 28

ConnectionString 28

Favourites 28

FemaleNames 28

MaleNames 28

MyUpdates 28

NameDataContext 27

UnisexNames 29

NameDataContext.cs 73

NameDetailPage class 29

about NameDetailPage class 29

flag 30

isFilled 31

NameDetailPage 29

OnFragmentNavigation 31

OnNavigatedFrom 32

OnNavigatedTo 32

pageName 31

ratingValue 31

NameDetailPage.xaml.cs 74

S

SearchPage class 33

about SearchPage class 33

OnNavigatedFrom 34

OnNavigatedTo 34

SearchPage 33

SearchPage.xaml.cs 84

StatisticsPage class 34

about StatisticsPage class 34

OnNavigatedFrom 35

OnNavigatedTo 35

StatisticsPage 35

StatisticsPage.xaml.cs 87

U

Unisex class 36

about Unisex class 36

UnisexId 36

UnisexMeaning 36

UnisexName 36

Unisex.cs 90