How to implement session-time out in a WP7 app

This article shows how to implement session time out in a Windows Phone 7 app using a DispatcherTimer

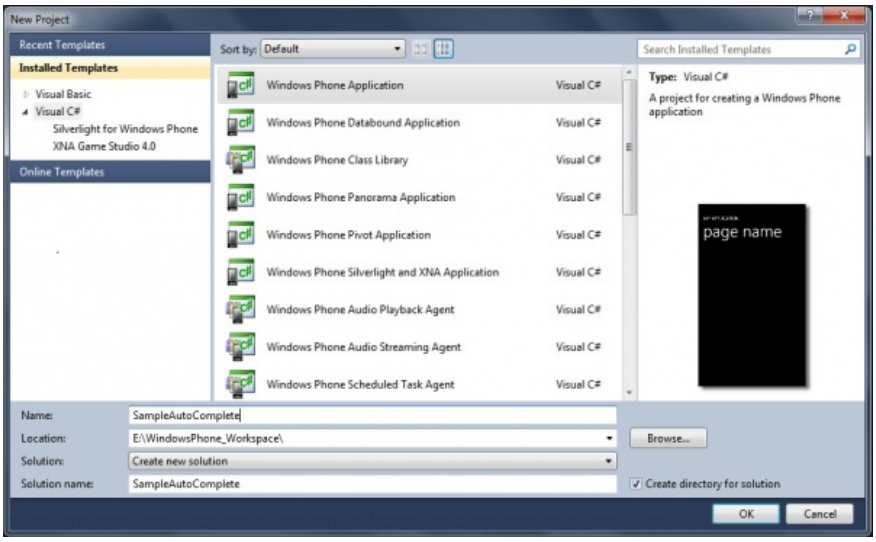
Introduction

Sometimes it is a useful "security feature" close a login (or other) page if the user does not enter the required details within a specified time. This code example demonstrates how to implement a session time-out.

The app has three pages: login, registration and detail. When the app is launched the login page is displayed. The user is asked to enter login credentials or can navigate to the registeration page if they are a new users. If the user does not provide their registration details within a specific time the session ends and the login page is displayed. Upon successful registration user is sent to the detail page.

Starting a new project

- Create a new project in Visual Studio 2010 from **File > New Project > Windows Phone Application - Visual C#**. Give a desired name of the project.



Designing LoginPage

In the **MainPage.xaml** file, we will drag and drop a TextBlock & TextBox controls. We will also use few App bar buttons. We will set the desired properties like shown below:

```
<Grid x:Name="LayoutRoot" Background="Transparent">
      <Grid.RowDefinitions>
          <RowDefinition Height=".50*"/>
          <RowDefinition Height=".40*"/>
          <RowDefinition Height=".30*"/>
          <RowDefinition Height=".20*"/>
          <RowDefinition Height=".30*"/>
          <RowDefinition Height="*"/>
      </Grid.RowDefinitions>

      <TextBlock x:Name="PageTitle"      Text="login"    Grid.Row="0" FontSize="80" Margin="10,0" VerticalAlignment="Bottom"/>
      <TextBlock x:Name="txt_username"   Text="UserName" Grid.Row="1" FontSize="40" Margin="15,0" VerticalAlignment="Bottom" Fontw
      <TextBox x:Name="txtbx_username" Text=""         Grid.Row="2" TextWrapping="Wrap" Margin="10,0" />
      <TextBlock x:Name="txt_password"   Text="Password" Grid.Row="3" FontSize="40" Margin="15,0" VerticalAlignment="Bottom" Fontw
      <PasswordBox x:Name="txtbx_password" Password=""     Grid.Row="4"  Margin="10,0" />
  </Grid>

  <!--Sample code showing usage of ApplicationBar-->
  <phone:PhoneApplicationPage.ApplicationBar>
      <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
          <shell:ApplicationBarIconButton x:Name="btn_login"    Click="btn_login_Click"    IconUri="/Icons/Tick.png"  Text="login"
          <shell:ApplicationBarIconButton x:Name="btn_reset"    Click="btn_reset_Click"    IconUri="/Icons/Reset.png" Text="reset"
          <shell:ApplicationBarIconButton x:Name="btn_register" Click="btn_register_Click" IconUri="/Icons/Reset.png" Text="regist
      </shell:ApplicationBar>
  </phone:PhoneApplicationPage.ApplicationBar>
```

Designing Registeration Page

In the **Page1.xaml** file, we will drag and drop a TextBlock & TextBox controls. We will also use few App bar buttons. We will set the desired properties like shown below:

```
<Grid x:Name="LayoutRoot" Tap="LayoutRoot_Tap">
      <Grid.RowDefinitions>
```


(C) Copyright Nokia 2013. All rights reserved.

```
            <RowDefinition Height=".20*"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title-->
        <TextBlock x:Name="PageTitle" Text="register" Grid.Row="0" FontSize="80" Margin="10,0" VerticalAlignment="Bottom"/>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="10">
            <ScrollViewer>
                <Grid  >
                    <Grid.RowDefinitions>
                        <RowDefinition Height=".20*"/>
                        <RowDefinition Height=".30*"/>
                        <RowDefinition Height=".20*"/>
                        <RowDefinition Height=".30*"/>
                        <RowDefinition Height=".20*"/>
                        <RowDefinition Height=".30*"/>
                        <RowDefinition Height=".20*"/>
                        <RowDefinition Height=".30*"/>
                    </Grid.RowDefinitions>

                    <TextBlock x:Name="txt_username"   Text="UserName" Grid.Row="0" FontSize="40" Margin="15,0" VerticalAlignment="B
                    <TextBox x:Name="txtbx_username" Text=""         Grid.Row="1" TextWrapping="Wrap" Margin="10,0" />
                    <TextBlock x:Name="txt_password"   Text="Password" Grid.Row="2" FontSize="40" Margin="15,0" VerticalAlignment="B
                    <PasswordBox x:Name="txtbx_password" Password=""     Grid.Row="3"  Margin="10,0" />

                    <TextBlock x:Name="txt_Emailid"   Text="E-mail Id" Grid.Row="4" FontSize="40" Margin="15,0" VerticalAlignment="B
                    <TextBox x:Name="txtbx_Emailid" Text=""          Grid.Row="5" TextWrapping="Wrap" Margin="10,0" />
                    <TextBlock x:Name="txt_Country"   Text="Country"   Grid.Row="6" FontSize="40" Margin="15,0" VerticalAlignment="B
                    <TextBox x:Name="txtbx_Country" Text=""          Grid.Row="7" TextWrapping="Wrap" Margin="10,0" />
                </Grid>
            </ScrollViewer>
        </Grid>
    </Grid>

    <!--Sample code showing usage of ApplicationBar-->
    <phone:PhoneApplicationPage.ApplicationBar>
        <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
            <shell:ApplicationBarIconButton x:Name="btn_register" Click="btn_register_Click" IconUri="/Icons/Tick.png"  Text="regist
            </shell:ApplicationBar>
    </phone:PhoneApplicationPage.ApplicationBar>
```

Designing Details Page

In the **welcome.xaml** file, we will drag and drop a TextBlock & Image control. We will set the desired properties like shown below:

```
<Grid x:Name="LayoutRoot" Background="Transparent">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*"/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title-->
        <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
            <TextBlock x:Name="PageTitle" Text="Welcome !!" Margin="9,-7,0,0" Style="{StaticResource PhoneTextTitle1Style}"/>
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="10">
            <Image Source="/Session-TimeOut%20App;component/Images/Penguins.jpg" Stretch="Fill" />
        </Grid>
    </Grid>
```

Code Behind

We have a session time-out class named `SessionOut` which keeps track of the dispatch timer.

Once the user is navigated to registration page the dispatch timer is invoked, as shown below:

```
public void start_timer(int time)
{
            if (_timer.IsEnabled == true)
            {
                _timer.Stop();
            }
            _timeLeft = time;
            _timer.Tick += TimerTick;     // event handler
            _timer.Start();
}
```

time parameter used in the `start_timer()` is the default session time.

- Timer tick event :

```
void TimerTick(object sender, EventArgs e)
{
            _timeLeft--;
            if (_timeLeft == 0)
            {
                _timer.Stop();
                log_off = true;
            }
}
```

In the above function we are decrement the `_timeLeft`. Once the value for `_timeLeft` becomes zero the timer is stopped and `log_off` variable is set to true. Now, if the user will tap on the registration page "Session Timed-Out" is popped and the user is navigated back to the login page.

Where as if the user will tap the registration page before the value of `_timeLeft` becomes 0, timer is reset to default session time and in case he/she will

(C) Copyright Nokia 2013. All rights reserved.

provide the credentials before the value of `_timeLeft` becomes 0, timer is stopped and user is navigated to details page.

Below is the method used to re-invoke the dispatch timer:

```
 private void LayoutRoot_Tap(object sender, GestureEventArgs e)
{
            if (SessionOut.log_off == true)  // if SessionOut flag is set to true, user is navigated back to main screen
            {
                MessageBox.Show("Session Expired");
                SessionOut.log_off = false;
                NavigationService.Navigate(new Uri("/MainPage.xaml", UriKind.Relative));
            }
            else
            {
                _SessionOut.start_timer(25); // reset the timer
            }
}
```

`LayoutRoot_Tap` is the event used to capture the tap event in registration page.

- Now build the project and run it using F5.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.