User interface design

for Window Phone 8, using C#

# Lab 2 – Asynchronously connecting to data

## Functional Goals

Take a functional Windows Phone 8 app that contains the necessary controls to display bandwidth usage data from Rose-Hulman’s network usage tool and add asynchronous data connections.

## Learning Goals

* Understand Windows Phone UI paradigms, including the Panorama control and the use of “Metro” design to create a straightforward, attractive Windows Phone application
* Understand the basics of the User Control extension paradigm of C# and Windows Phone

## Prerequisites

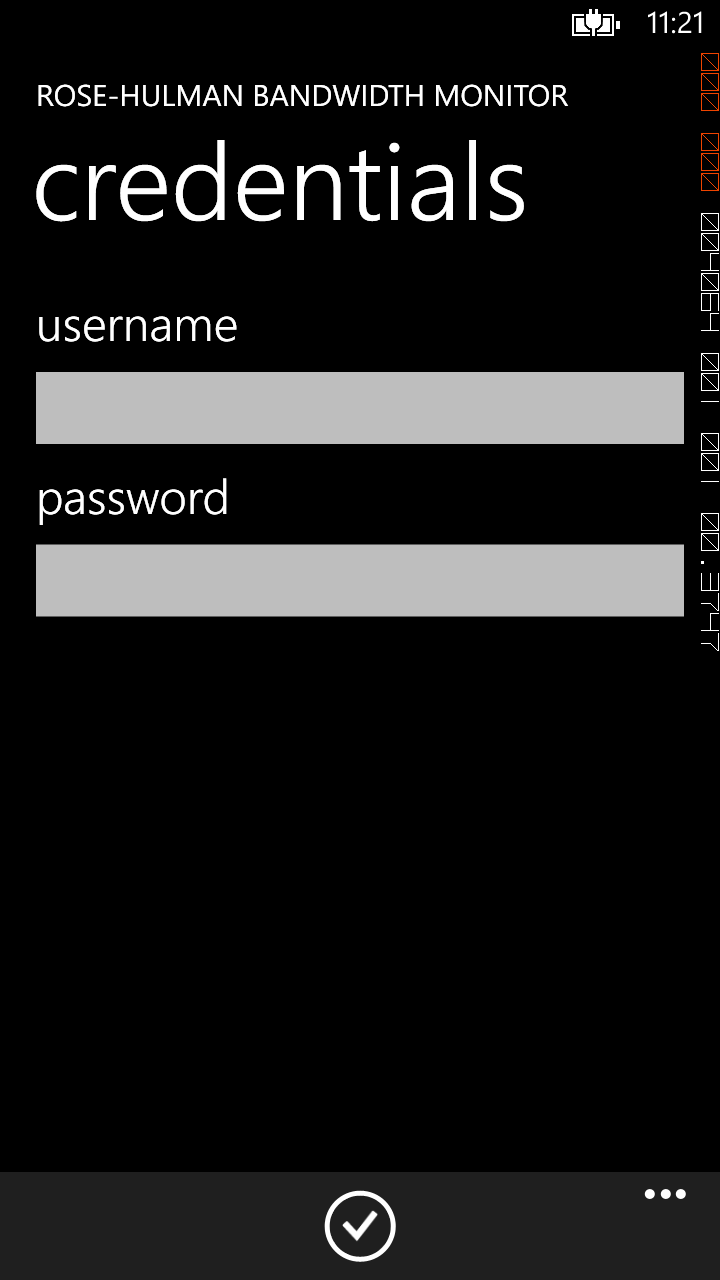
* You’ll need to install Visual Studio (2012 Ultimate was used to create this lab) from the MSDN/DreamSpark service on ANGEL’s RosePortal.
* You’ll also need to download and install the Windows Phone SDK (8.0 was used to create this lab) from https://dev.windowsphone.com/en-us/downloadsdk
  + To properly run the Windows Phone emulator, you’ll need to ensure that **second-level address translation (SLAT)** and **hardware Data Execution Prevention** (Execute Disable on Intel systems; No Execute on AMD) in your BIOS settings.
* A basic understanding of Visual Studio and C# development, such as that gained from the **User interface design in C#, using WPF** series in this document’s repository.
* Code from Lab3 (a complete version may be available from your instructor)

## Submission Instructions

Submit answers to the **3** (or **4**, with extra credit)questions in this lab as a .pdf to the appropriate Moodle submission form.

## Get started: Make a Settings page

To start with, you’ll need to collect the user’s Rose-Hulman network credentials in order to connect to the bandwidth tool. To do that, you’re going to make a new page called SettingsPage.xaml.

1. Create a new Portrait Page called SettingsPage.xaml.
2. Add controls to the page so that it looks like the screenshot on the right.
3. To create the checkmark in the bottom, you’ll need to add an ApplicationBar:

<phone:PhoneApplicationPage.ApplicationBar>

<shell:ApplicationBar>

<shell:ApplicationBarIconButton

IconUri="/Assets/AppBar/check.png"

IsEnabled="True" Text="Save" Click="SaveClick"/>

</shell:ApplicationBar>

</phone:PhoneApplicationPage.ApplicationBar>

1. Copy check.png from C:\Program Files (x86)\Microsoft SDKs\Windows Phone\v8.0\Icons\Light to Assets\AppBar in your solution folder.

## Saving information: IsolatedStorage

To use the credentials the user enters on the Settings page, you’ll need to first add them to IsolatedStorage so that you can access them across pages and app sessions.

1. In SettingsPage.xaml.cs, add/modify the following code:

public SettingsPage()

{

InitializeComponent();

var settings = IsolatedStorageSettings.ApplicationSettings;

if (settings.Contains("user"))

UsernameTextBox.Text = (string) settings["user"];

if (settings.Contains("pass"))

PasswordBox.Password = (string) settings["pass"];

}

private void SaveClick(object sender, EventArgs e)

{

var settings = IsolatedStorageSettings.ApplicationSettings;

if (settings.Contains("user"))

settings["user"] = UsernameTextBox.Text;

else

settings.Add("user", UsernameTextBox.Text);

if (settings.Contains("pass"))

settings["pass"] = PasswordBox.Password;

else settings.Add("pass", PasswordBox.Password);

NavigationService.Navigate(new Uri("/MainPage.xaml", UriKind.Relative));

}

Question 1: What is the data structure type (don’t forget to specify the inner types) that IsolatedStorageSettings.ApplicationSettings uses to store the settings you added above? (9 points)

## Making it accessible: MainPage AppBar

To get to the Settings page from MainPage, you’ll need an AppBar button and a call to NavigationService. Follow the example code in the previous two sections if you’re not sure how to do this.

Question 2: Add the AppBar and navigation logic as described above. Make sure to add an appropriate icon and subtext. Submit the XAML code and codebehind methiod. If you are using a word processor that doesn’t retain Visual Studio’s text formatting on copy/paste, please take a screenshot of your code so that it remains properly formatted and colored. (18 points)

## Congratulations

You’re done; run your app and check it out! It doesn’t do anything just yet – you’ll take care of that in the next lab. Don’t forget: submit answers to the **3** (or **4**, with extra credit)questions in this lab as a .pdf to the appropriate Moodle submission form.