FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Faculty 2: Computer Science and Engineering

VIETNAMESE-GERMAN UNIVERSITY (VGU)

Electrical Engineering and Information Technology (EEIT)

Windows Phone 8 Application

by

Nguyen Le Khoi Nguyen

Luu Nghiem Bao Trung

Supervisor

Prof. Pech

Second Supervisor

Mr. Ha Ngoc An

Table of Contents

Table of Figures

Abstract: As the development of the technology today, wireless data acquisition and observation are crucial in many industrial processes which contain a lot of parameters. This report discusses a smartphone application of Windows Phone 8 compatible for data acquisition and display in.

Windows Phone 8 is one of the available smartphone compatible operation systems and is chosen as the main platform in order to meet the requirement of the course Information Processing at the University of Applied Sciences and the Vietnamese-German University. The application shows continuously updated maximum of 14 parameters from an assigned webpage with the function of choosing which parameters to be shown. The application consists of two main part of data acquisition and data display. Data acquisition is supported by the method WebClient and the data display process exploits the ItemViewModel class.

Keywords: Windows Phone 8, smartphone application, information processing, data acquisition, data display, WebClient, ItemViewModel

Introduction

Background

Information Processing is a standard module included in the bachelor program of Electrical Engineering and Information Technology at the Vietnamese-German University, Ho Chi Minh city, Vietnam in association with the Frankfurt University of Applied Sciences, Frankfurt, Germany. This report is the technical report of the theme project provided by the professor in charge and acts as a part of quality assessment of the course. The score of the report will indicate the completeness of the module of the students whose names are written at the front page. The project is due on June 15th, 2016 accomplished by this written document submission.

Project description

The project will be conducted in a group consisting of two students. Students are asked to develop a Windows Phone 8 application the following criteria:

- With dashboard displaying from 5 to 15 parameters from a simulated industrial process which will be kept continuously updated.
- The data is fetched from

http://ivlab.azurewebsites.net/

http://ivlab.azurewebsites.net/tank1

http://ivlab.azurewebsites.net/pumpe3

- The data must contain when shown: < parameter name > < datatype > < minimum > to < maximum > < Current Value >

Windows Phone 8 Application

Structure and User Interface of the application

The application consists of two main page, the first page displays the parameters of a simulated industrial process fetched from website [figure 1]. The second page shows the setting of the application, which allows users to activate or inactivate the visibility of parameters on the first page [figure 2].

Figure 1: Dashboard Page

Figure 2: Properties Page

The application is designed and programmed by using Visual Studio 2013 environment with C# language and .NET framework. The result is displayed by running the emulator "WVGA 512MB", which is integrated in the windows phone 8 SDK of Visual Studio 2013 [figure 3].

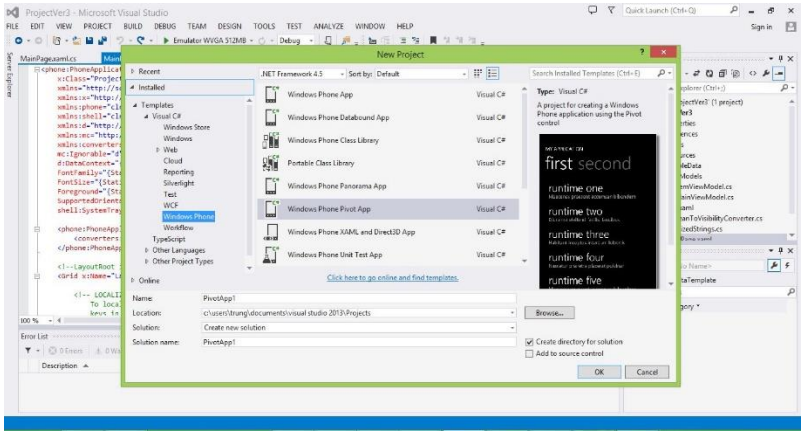


Figure 3: Visual Studio 2013 Interface

How it work

When the program is started, the emulator is automatically launched and install the application on the emulator. Besides, one network connection is created by the emulator as a local network to connect to the internet [figure 4]. All the parameters is updated and performed on the dashboard page.

Figure 4: Network Connection of Emulator

User Guide

The following steps are detail instruction for users to launch and use the application:

Step 1: After installing, users will see the icon 

in the application page of the phone or the emulator. Tap on the icon to start launching the application.

Step 2: The Dashboard page will be opened, with parameter name, current value, min and max value perform on the screen as figure below. User can swipe up or down on the screen to view all the parameters.

Figure 5: Parameters performance

Step 3: You can swipe left or right to change view the Properties Page. On the Properties Page, there are the parameter name and the check box for users to control the number of parameters displayed on the Dashboard screen. By checking the checkbox on the line of that parameter name, that parameter will be shown on the Dashboard and vice versa.

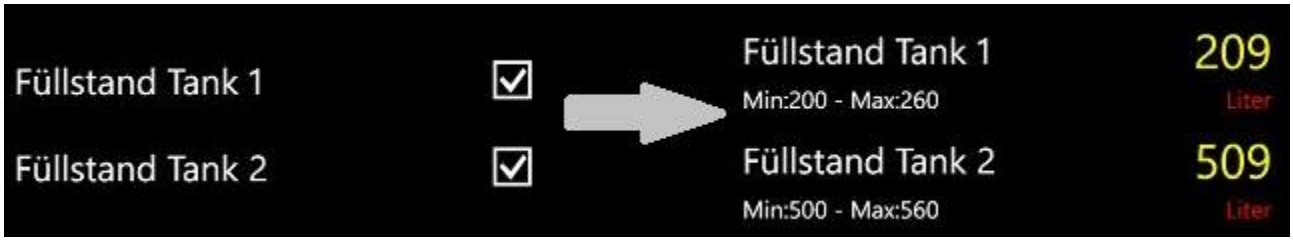


Figure 6: Checked Checkbox

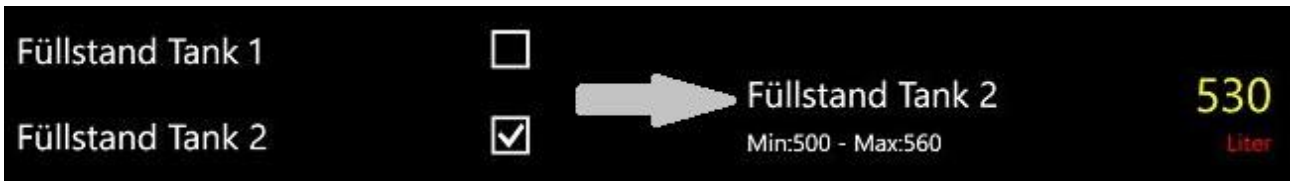
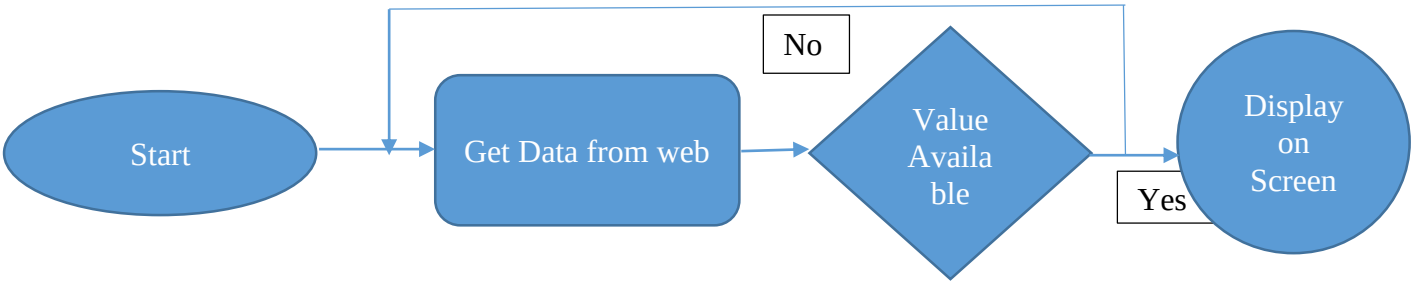


Figure 7: Unchecked Checkbox

Step 4: To quit the program, you only need to tap  button at the bottom of the phone.

Code Description

The program under the application can briefly understand by the flow below:



To retrieve the value from the given website, the method WebClient is used. With the event "DownloadStringCompleted", the result is returned from service call. Besides, the method "DownloadStringAsync" is used because the URL is defined and its values can be represented as string. This method is used to take data from website. The result after fetching from web, will be assign for the current value. For the continuous updating reason, the random number must be created to trick the web client get new cache each time it loaded.

```
private void GetDataFromWebpage(int currentWeb)
{
    Random random = new Random();
    ItemViewModel view = Items[currentWeb];
    var address = "http://ivlab.azurewebsites.net" + view.Path + "?=Uni" + random.Next(0, 1000);
    webClient = new WebClient();
    String result;
    webClient.DownloadStringCompleted += (sender, e) =>
    {
        result = e.Result;
        if (result == null)
        {
            view.CurrentValue = "updating";
        }
        else
        {
            view.CurrentValue = result;
        }
    };
    webClient.DownloadStringAsync(new Uri(address));
}
```

Figure 8: Get Data from website

Inside the WebClient.DownloadStringCompleted[1] event, we assigned the result that get from the website to the view of current value.

The data model is created first in an "ItemViewModel" class, which implement the "INotifyPropertyChanged" and "PropertyChanged" to notify the change of property value initialized. In this class, we create the Boolean Visibility and string CurrentValue for notify the change of properties. And for the ViewModel, it is created to connect the Model and the View of the project. The ObservationCollection<ItemViewModel> is used to notify the view changes and drive the changes in the Interface. In XAML file, binding code connects data with View.

For the visible of parameters on the application interface, the class Boolean to Visibility [2] is created to convert the Boolean value to the Visibility enumeration value.

```
public class BooleanToVisibilityConverter: IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
    {
        return (bool)value ? Visibility.Visible : Visibility.Collapsed;
    }

    public object ConvertBack(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
    {
        throw new NotImplementedException();
    }
}
```

Figure 9: Boolean to Visibility class

The User Interface of the application is designed in a XAML file, which is used Data binding to bind data from ViewModel and show on the screen.

```
<!--Pivot Control-->
<phone:Pivot Title="Display Application">
    <!--Pivot item one-->
    <phone:PivotItem Header="DashBoard">
        <!--Double line list with text wrapping-->
        <phone:LongListSelector Margin="0,2,2,-37" ItemsSource="{Binding Items}" RenderTransformOrigin="0.476,0.573">
            <phone:LongListSelector.ItemTemplate>
                <DataTemplate>
                    <Grid Margin="0,0,0,10"
                          Visibility="{Binding Visibility, Converter={StaticResource BooleanToVisibility}}">
                        <TextBlock Text="{Binding ParameterName}" TextWrapping="Wrap" FontSize="30"
                                   Grid.Row="0" Grid.Column="0" VerticalAlignment="Center"/>
                        <TextBlock Text="{Binding MaxandMin}" TextWrapping="Wrap" FontSize="20"
                                   Grid.Row="1" Grid.Column="0" VerticalAlignment="Center" />
                        <TextBlock Text="{Binding CurrentValue}" TextWrapping="Wrap" FontSize="40"
                                   Grid.Row="0" Grid.Column="1"
                                   TextAlignment="Right" VerticalAlignment="Center" Height="54" RenderTransformOrigin="0.2,2.111">
                            <TextBlock.Foreground>
                                <SolidColorBrush Color="Yellow"/>
                            </TextBlock.Foreground>
                        </TextBlock>
                        <TextBlock Text="{Binding DataType}" TextWrapping="Wrap" FontSize="20"
                                   Grid.Row="1" Grid.Column="1"
                                   Foreground="Red"
                                   TextAlignment="Right" VerticalAlignment="Center"/>
                        <Grid.RowDefinitions>
                            <RowDefinition Height="1*"/>
                            <RowDefinition Height="2*"/>
                        </Grid.RowDefinitions>
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="5*"/>
                            <ColumnDefinition Width="3*"/>
```

Figure 10: XAML class

In this file, the converter is used to bind the Boolean parameter Visibility to the Visibility properties.

Conclusion

This paper explicates the idea behind the proposed project “Windows Phone 8 Application” and our approach. The application contents 2 main features: get data from defined URL and display on the screen. However, this work obviously can be improve to achieve the perfect goal with more functions and interaction with the users.

During the project, we gain much experience in programming windows phone application with .NET framework and C# language, we also can apply the knowledge of the Information Processing course into making plan and control the workload of the project.

References

[1]

Windows phones 8 Development Internal: https://books.google.de/books?id=27ZCAwAAQBAJ&pg=PT401&lpg=PT401&dq=webClient.DownloadStringCompleted+%2B%3D+(sender,+e)&source=bl&ots=G3TFtEOyDM&sig=GyI5NpDpNxVbBY_TMN8gSPjcIp8&hl=vi&sa=X&ved=0ahUKEwjxvu2P1qnNAhVMXBoKHY5zDXgQ6AEIQDAE#v=onepage&q=webClient.DownloadStringCompleted%20%2B%3D%20(sender%2C%20e)&f=false

[2] Microsoft Visual Studio webpage: https://msdn.microsoft.com/en-us/library/system.windows.controls.booleantovisibilityconverter(v=vs.110).aspx