M04 Introduction to Universal Windows Platform (UWP)

Multi Page UWP App

Author: Yi Chen
Date: 2021.02.14

Table of Contents

List of Tables

List of Figures

1. Multi Page UWP App

Create a new Universal Windows Platform (UWP) application including multiple pages and a hamburger style navigation pane on the left.

2. Project Requirements

- *Make use of at least 2 or 3 pages.*
- *Use a hamburger style navigation menu to switch between pages.*
- *Incorporate creating, writing/appending and reading a file (e.g. text file) in your application.*
- *Store file(s) in one of the app folders (local, temp, roaming).*
- *Incorporate a user content upload feature in your application. For example, profile picture upload.*

○ Derived Requirements

- *A prepared document of your application with the following components:*
    - *All necessary meta information, such as but not limited to: Title, Name, Data, Course, Section Headers.*
    - *All your commented C# code.*
- *A zipped copy of your executable files.*

3. Design Plans

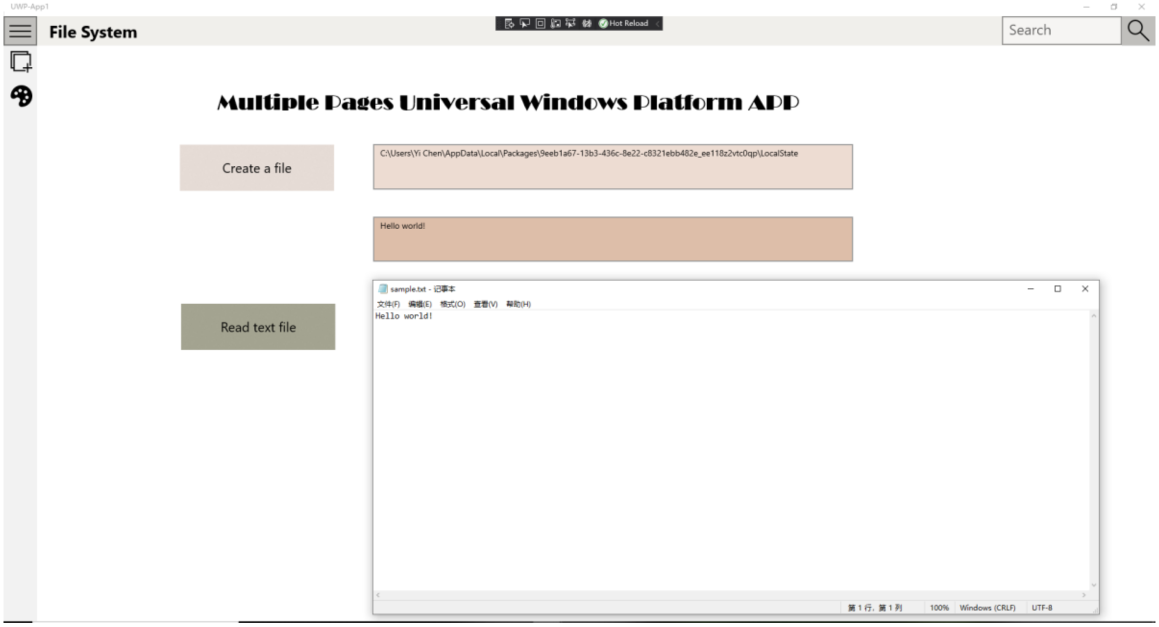
Using UWP to make to page of file read and write system.

3.1. File System && Image System

User can create and write files in file system (First page), user can upload an image to second page which is image system.

3.1.1. File System

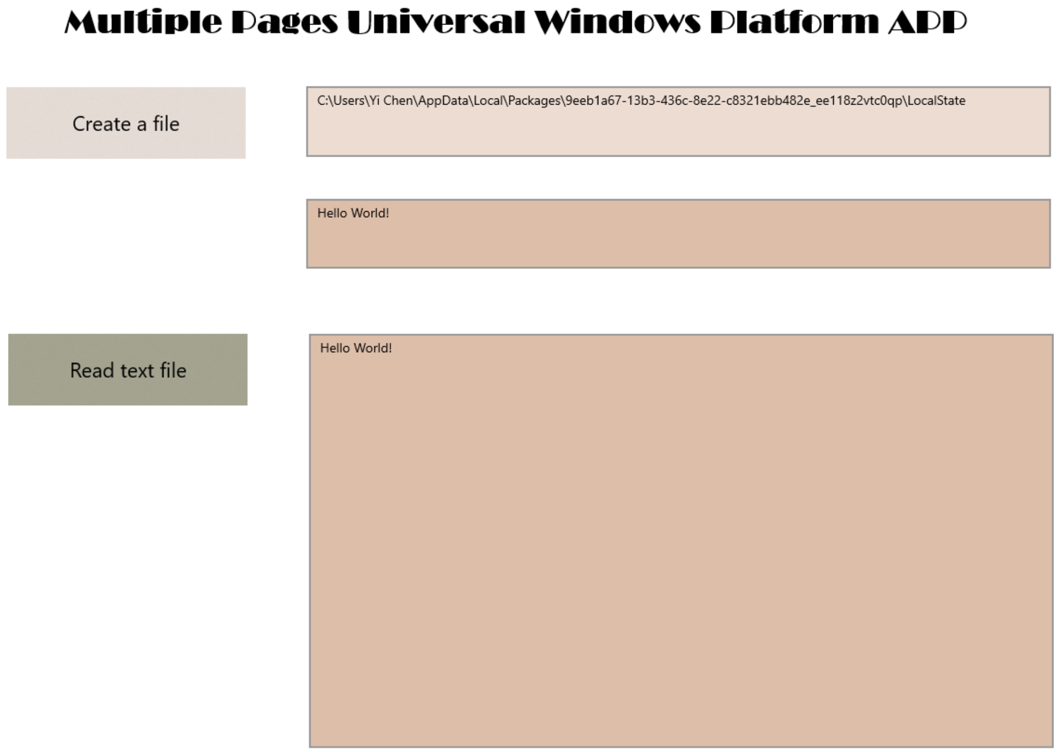
- *Button:*
  - *Create a file*
    - *when user clicks button, a file be created in a specifical location.*
    - *Show the file location.*
  - *Read text file – when user clicks button, read a file which is be created in a specifical location.*
- *Text box:*
  - *FilePath_TxtBox: Show file path.*
  - *userInput: User can input file that they want to input file*
  - *readBox: Read the file which is just created.*


UWP-App1
File System
Hot Reload
Search
Multiple Pages Universal Windows Platform APP
Create a file
Read text file
sample.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
第 1 行，第 1 列 100% Windows (CRLF) UTF-8

C:\Users\Yi Chen\AppData\Local\Packages\9eeb1a67-13b3-436c-8e22-c8321ebb482e_ee118z2vtc0qp\LocalState

Hello world!

```
Hello world!
```



Multiple Pages Universal Windows Platform APP

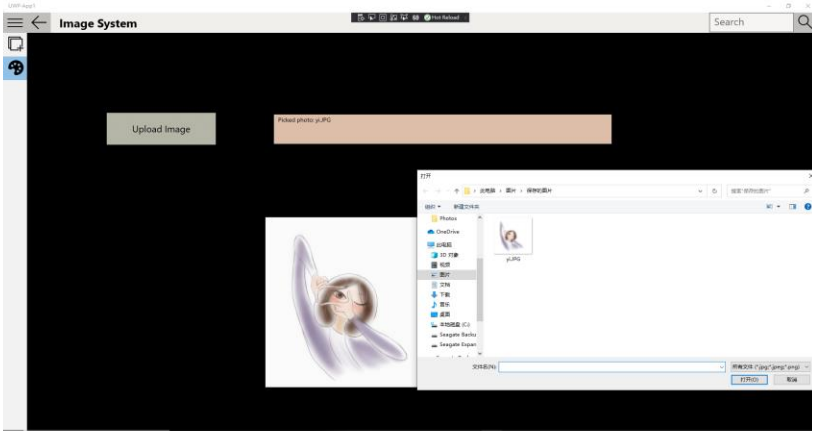
C:\Users\Yi Chen\AppData\Local\Packages\9eeb1a67-13b3-436c-8e22-c8321ebb482e_ee118z2vtc0qp\LocalState

Hello World!

Hello World!

3.1.2. Image System

- *Button:*
  - *Upload Image – upload image from local.*
- *Text Box:*
  - *user can see the image's name*
  - *If an user doesn't upload, it shows Operation cancelled.*

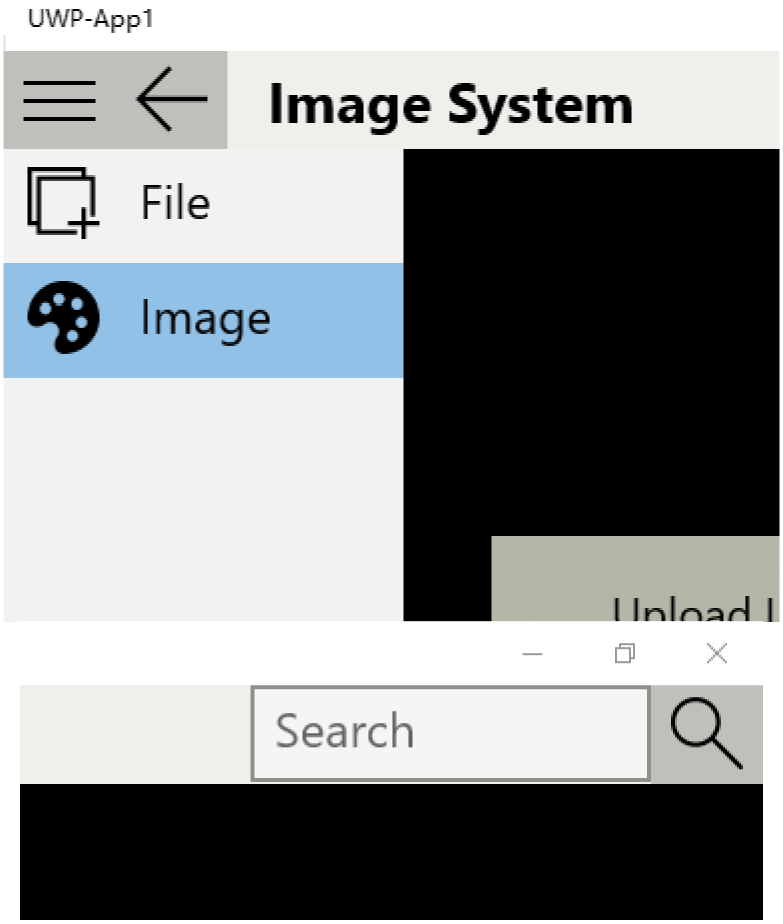


Picked photo: yi.JPG

3.1.3. Hamburger Button && Search

- *Hamburger button :*
    - *fold a page.*
    - *File ListBoxItem: When a user clicks it shows the file system page.*
    - *Image ListBoxItem: When a user clicks it shows the Image system page.*
- *Search box.*

Table 1: Hamburger button && search



4. Implementation

- *MainPage.xaml.cs:*
  - *Implement Hamburger button*
- *MainPage.xaml:*
  - *Implement Hamburger button layout*
- *FileSystem.xaml.cs:*
  - *Implement File page – create, write, save, and read file.*
- *FileSystem.xaml:*
  - *Implement File Page layout*
- *ImageSystem.xaml.cs:*
  - *Implement Image system page – uploading an image from local, and read image name.*
- *ImageSystem.xaml:*
  - *Implement Image Page layout*

4.1. Player & Computer Implementation

- *MainPage.xaml.cs:*
  - *Implement Hamburger button*
- *MainPage.xaml:*
  - *Implement Hamburger button layout*

Figure 1: Hamburger button && Back button

Hamburger button

```
namespace UWP_App1
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    /// Author: Yi Chen
    public sealed partial class MainPage : Page
    {
        public MainPage()
        {
            this.InitializeComponent();
            BackButton.Visibility = Visibility.Collapsed;
            MyFrame.Navigate(typeof(FileSystem));
            TitleTextBlock.Text = "File System";
            FileSystem.IsSelected = true;
        }


        // click Hamburger button  open foled page
        private void HamburgerButton_Click(object sender, RoutedEventArgs e)
        {
            MySplitView.IsPaneOpen = !MySplitView.IsPaneOpen;
        }
        // On the last page, when click back arrow come back to home page
        private void BackButton_Click(object sender, RoutedEventArgs e)
        {
            if (MyFrame.CanGoBack)
            {
                MyFrame.GoBack();
                FileSystem.IsSelected = true;
            }
        }

        // User can Change to file or image page.
        private void ListBox_SelectionChanged(object sender, 
SelectionChangedEventArgs e)
        {
            if (FileSystem.IsSelected)
            {
                BackButton.Visibility = Visibility.Collapsed;
                MyFrame.Navigate(typeof(FileSystem));
                TitleTextBlock.Text = "File System";
            }
            else if (ImageSystem.IsSelected)
            {
                BackButton.Visibility = Visibility.Visible;
                MyFrame.Navigate(typeof(ImageSystem));
                TitleTextBlock.Text = "Image System";
            }
        }
    }
}
```

Hamburger layout

```
  /**
         * Computer will choose a card randomly
         * 
         */

        private void Computer_play_Btn_Click(object sender, RoutedEventArgs e)
        {
            randomNumber = rnd.Next(0, ComputerchoiseList.Length);

            ComputerChoice = ComputerchoiseList[randomNumber];


            switch (ComputerChoice)
            {
                case "rock_r":
                    Computer.Source = new BitmapImage(new 
Uri(@"/Images/rock_r.jpg", UriKind.Relative));
                    break;

                case "paper_r":
                    Computer.Source = new BitmapImage(new 
Uri(@"/Images/paper_r.jpg", UriKind.Relative));
                    break;

                case "scissor_r":
                    Computer.Source = new BitmapImage(new 
Uri(@"/Images/scissors_r.jpg", UriKind.Relative));
```

```
                    break;

            }

            CheckGame();


        }
```

4.2. File System

- *FileSystem.xaml.cs:*
    - *Implement File page – create, write, save, and read file.*
- *FileSystem.xaml:*
    - *Implement File Page layout*

Table 2: File system

cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// The Blank Page item template is documented at
https://go.microsoft.com/fwlink/?LinkId=234238

namespace UWP_App1
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a
Frame.
    /// </summary>
    /// Author: Yi Chen
    public sealed partial class FileSystem : Page
    {


        public FileSystem()
        {
            this.InitializeComponent();
        }
        /*Create a file in localFolder. when user input text in "userInput
text box", save the text which is just inputed to the local file.*/
        private async void CreateAFile_BTN_Click(object sender,
RoutedEventArgs e)
        {

            // Create sample file; replace if exists.
            Windows.Storage.StorageFolder storageFolder =
Windows.Storage.ApplicationData.Current.LocalFolder;

            // show path
            FilePath_TxtBox.Text = storageFolder.Path;

            //Save file and name is sample.txt
            Windows.Storage.StorageFile sampleFile = await
storageFolder.CreateFileAsync("sample.txt",
Windows.Storage.CreationCollisionOption.ReplaceExisting);

            // user input text
            await Windows.Storage.FileIO.WriteTextAsync(sampleFile,
(string)userInput.Text);
        }

        /*read a file which user inputed*/
        private async void ReadFile_BTN_Click(object sender, RoutedEventArgs
e)
        {
            //A file is to get the file with StorageFolder.GetFileAsync.
            Windows.Storage.StorageFolder storageFolder =
Windows.Storage.ApplicationData.Current.LocalFolder;
            Windows.Storage.StorageFile sampleFile = await
storageFolder.GetFileAsync("sample.txt");

            //Read text from your file by calling the FileIO.ReadTextAsync
method.
            readBox.Text = await
Windows.Storage.FileIO.ReadTextAsync(sampleFile);
        }
    }

}
```

xaml:

```
<Page
    x:Class="UWP_App1.FileSystem"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:UWP_App1"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

    <Grid>

        <Button x:Name="CreateAFile_BTN" Content="Create a file"
Margin="238,165,0,0" VerticalAlignment="Top" Height="77" Width="257"
```

```
FontSize="22" Click="CreateAFile_BTN_Click">
            <Button.Background>
                <AcrylicBrush TintColor="#FFEDDCD2"/>
            </Button.Background>
            <Button.FocusVisualPrimaryBrush>
                <AcrylicBrush TintColor="#CC9C0A0A"/>
            </Button.FocusVisualPrimaryBrush>
        </Button>
        <TextBox x:Name="userInput" HorizontalAlignment="Left"
Margin="560,285,0,0" Text="" TextWrapping="Wrap" VerticalAlignment="Top"
Height="75" Width="800" Background="#FFDDBEA9"/>
        <TextBox x:Name="FilePath_TxtBox" HorizontalAlignment="Left"
Margin="560,164,0,0" Text="TextBox" TextWrapping="Wrap"
VerticalAlignment="Top" Height="76" Width="800" Background="#FFEDDCD2"/>

        <Button x:Name="ReadFile_BTN" Content="Read text file"
Margin="240,430,0,0" VerticalAlignment="Top" Height="77" Width="257"
FontSize="22" Click="ReadFile_BTN_Click">
            <Button.Background>
                <AcrylicBrush TintColor="#FFA5A58D"/>
            </Button.Background>
            <Button.FocusVisualPrimaryBrush>
                <AcrylicBrush TintColor="#CC9C0A0A"/>
            </Button.FocusVisualPrimaryBrush>
        </Button>
        <TextBox x:Name="readBox" HorizontalAlignment="Left"
Margin="563,430,0,0" Text="" TextWrapping="Wrap" VerticalAlignment="Top"
Height="445" Width="800" Background="#FFDDBEA9"/>

        <TextBlock x:Name="label" HorizontalAlignment="Left"
Margin="300,74,0,0" Text="Multiple Pages Universal Windows Platform APP"
TextWrapping="Wrap" VerticalAlignment="Top" Height="67" Width="1009"
FontSize="36" FontFamily="Broadway" FontWeight="Bold"/>
    </Grid>




</Page>
```

4.3. Image System

- *ImageSystem.xaml.cs:*
    - *Implement Image system page – uploading an image from local, and read image name.*
- *ImageSystem.xaml:*
    - *Implement Image Page layout*

Table 3: Image system implement

Cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices.WindowsRuntime;
using Windows.Foundation;
using Windows.Foundation.Collections;
using Windows.UI.Xaml;
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;
using Windows.UI.Xaml.Data;
using Windows.UI.Xaml.Input;
using Windows.UI.Xaml.Media;
using Windows.UI.Xaml.Navigation;

// The Blank Page item template is documented at
https://go.microsoft.com/fwlink/?LinkId=234238

namespace UWP_App1
{
    /// <summary>
    /// An empty page that can be used on its own or navigated to within a Frame.
    /// </summary>
    public sealed partial class ImageSystem : Page
    {
        public ImageSystem()
        {
            this.InitializeComponent();
        }

        private async void UploadImage_BTN_Click(object sender, RoutedEventArgs e)
        {
            var picker = new Windows.Storage.Pickers.FileOpenPicker();
            picker.ViewMode = Windows.Storage.Pickers.PickerViewMode.Thumbnail;
            picker.SuggestedStartLocation =
Windows.Storage.Pickers.PickerLocationId.PicturesLibrary;
            picker.FileTypeFilter.Add(".jpg");
            picker.FileTypeFilter.Add(".jpeg");
            picker.FileTypeFilter.Add(".png");

            Windows.Storage.StorageFile file = await picker.PickSingleFileAsync();
            if (file != null)
            {
                // Application now has read/write access to the picked file
                userInput.Text = "Picked photo: " + file.Name;
                string newPic = "ProfilePic" + file.FileType;
                var storageFolder =
Windows.Storage.ApplicationData.Current.LocalFolder;
                var newPicFile = await file.CopyAsync(storageFolder, newPic,
Windows.Storage.NameCollisionOption.ReplaceExisting);
                image_box.Source = new
Windows.UI.Xaml.Media.Imaging.BitmapImage(new System.Uri(newPicFile.Path));
            }
            else
            {
                userInput.Text = "Operation cancelled.";
            }
        }
    }
}
```

Xmal

```
<Page
    x:Class="UWP_App1.ImageSystem"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:UWP_App1"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

    <Grid Background="#FFFFE6E2">
        <TextBox x:Name="userInput" HorizontalAlignment="Left"
Margin="645,197,0,0" Text="" TextWrapping="Wrap" VerticalAlignment="Top"
Height="75" Width="800" Background="#FFDDBEA9"/>
        <Button x:Name="UploadImage_BTN" Content="Upload Image"
Margin="253,195,0,0" Height="77" Width="257" FontSize="22" VerticalAlignment="Top"
Click="UploadImage_BTN_Click">
            <Button.Background>
                <AcrylicBrush TintColor="#FFB7B7A4"/>
            </Button.Background>
            <Button.FocusVisualPrimaryBrush>
                <AcrylicBrush TintColor="#CC9C0A0A"/>
            </Button.FocusVisualPrimaryBrush>
        </Button>
        <Image x:Name="image_box" HorizontalAlignment="Left" Height="410"
Margin="562,445,0,0" VerticalAlignment="Top" Width="788"
FocusVisualPrimaryBrush="#FFF0EFEB" Opacity="0.985">
            <Image.FocusVisualSecondaryBrush>
                <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
                    <GradientStop Color="Black"/>
                    <GradientStop Color="#FFFFDFDF" Offset="1"/>
                </LinearGradientBrush>
            </Image.FocusVisualSecondaryBrush>
        </Image>
        <TextBlock x:Name="label" HorizontalAlignment="Left" Margin="300,74,0,0"
Text="Multiple Pages Universal Windows Platform APP" TextWrapping="Wrap"
VerticalAlignment="Top" Height="67" Width="1009" FontSize="36"
FontFamily="Broadway" FontWeight="Bold"/>
    </Grid>
</Page>
```