User interface design

for Window Phone 8, using C#

# Lab 4 – Submitting an app to the Windows Phone store

## Functional Goals

Prepare a complete Windows Phone 8 app for submission to the Windows Phone store.

## Learning Goals

* Understand Windows Phone app manifests and related materials

## 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) are enabled 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), though any functional Windows Phone app can be used with this lab.

## Submission Instructions

Submit answers to the **1** (or **2**, with extra credit)questions in this lab as a .pdf to the appropriate Moodle submission form.

## Make it ship-ready: Give it a name and a picture

If you deploy your app to the emulator or a Windows Phone device right now, you’ll notice that the listing in the app list is the name of your assembly, not a proper name for your app. To change this, you’ll need to do the following:

1. In the project, under the Properties folder, open WMAppManifest.xml.
2. Fill in the Display Name, Description, and Tile Title with descriptive titles and information.
3. If you haven’t already, replace Assets/ApplicationIcon.png with a 101x101px icon that describes your application.
4. Consider adding a Live Tile to your app; see Lab 3 for more instructions if you’re interested.

## Other manifest parts: Capabilities, Requirements, and Packaging

In WMAppManifest.xml, notice that there are 3 tabs after the **Application UI** tab you just worked in. These tabs are important as you get into more advanced application design; to use most hardware and network functions (e.g. Xbox Live integration, the microphone, or push notification), you have to request the capability in the appropriate tab. Likewise, if your app requires a feature (e.g. NFC), you can prevent it from being installed on non-compatible devices using the **Requirements** tab.

Finally, the **Packaging** tab specifies some metadata for your app. To finish up your store preparation,

1. Update the Author and Publisher listings on the Packaging tab.

Question 1: Submit a screenshot of your app (with proper title and app icon) in the app list. (10 points)

## Ready, go: Submit your app to the store

If you’re using the Bandwidth Monitor codebase that precedes these labs, you may not want to actually submit your app, as it might be a duplicate. It is possible to earn full credit on the lab without doing this section. If you do submit a Bandwidth Monitor app, or another app that relies on internal Rose-Hulman websites, you’ll need to add a “demo mode” to your code to enable the app certification team to successfully certify your app; see https://github.com/alexmullans/RHITBandwidth/tree/master /RoseHulmanBandwidthMonitorApp for one such implementation.

To submit an app, you’ll need a Windows Phone Dev Center account. These are $99/year, but you can get one for free as a student through DreamSpark (not Rose-Hulman’s DreamSpark Premium aka MSDN Academic Alliance, but a separate service you use directly through Microsoft):

1. Go to http://dev.windowsphone.com.
2. Sign in with your Microsoft Account.
3. Click the Dashboard tab and follow the registration process as a student.

When you’ve registered successfully and verified your student credentials, return to the Dashboard tab to submit your application:

1. Click Submit App.
2. Click App Info and follow the process to submit your app successfully.
   1. For the screenshots, remember that you can take screenshots using the Windows Phone Emulator, or by simultaneously pressing the Power and Windows keys on an actual device.

## Congratulations

You’re done! Don’t forget: submit answers to the **1** (or **2**, with extra credit)question(s) in this lab as a .pdf to the appropriate Moodle submission form.

## Above and beyond: In-app advertising

From here on is extra credit. It is possible to earn full credit on the lab without doing this section.

For free apps, the primary way to make money is through small advertisements in your application. Microsoft Advertising provides a relatively easy way to do this.

Question 2: Create a Microsoft Advertising pubCenter account and add an ad unit to your app. Submit a screenshot of the ad within your app. (10 points)