

Accepted and presented at the:

The 10th International Conference on

Education and Information Systems, Technologies and Applications: EISTA 2012

July 17 - 20, 2012 – Orlando, Florida, USA

Developing Android Applications: Case Study of Course Design

By

Dr. Bill Rosener

Northeastern State University

INTRODUCTION

The following is a case study over designing a new course on developing applications for mobile devices (smart phones and tablets) running the Android operating system. The course is Information Systems 3323 “Developing Android Applications for Mobile Devices” and was taught for the first time at Northeastern State University during spring of 2012. This paper focuses on how the course was developed (including: analyzing student interest, analyzing the job market, determining instructional materials, installing the software, determining the structure of the class, etc.), problems that were encountered, and conclusions that were reached.

Analysis of student interest:

Are students interested in taking a course on this topic?

Yes, during early fall 2011 two surveys were conducted in general business courses. In both surveys approximately 25% of students questioned expressed an interested in taking a course on this topic. The results of these surveys were reflected during the enrollment period. Sure enough, enrollment in the class reached capacity well before classes started in spring 2012. There was even a waiting list for students to enroll in this course.

How should the class be taught?

Of those students interested, a majority of the students preferred the class be taught using either a blended or online format. For the two reasons listed below, it was decided to offer the class in a blended format. Unlike a completely online course, the blended format:

- Allowed for student presentations. Many job advertisements in this area mentioned communication skills and keeping up-to-date on the latest Android developments as important skills. As discussed later, this presentation assignment would help the students meet both of these requirements.
- Allowed the students to begin using and understand the software immediately. By pre-installing the software in the classroom computer lab, students were able to complete their first “simple” Android application during the first class meeting. Installing the software is somewhat time-consuming and challenging. If this course were offered

online, students could have easily spent 1-2 hours downloading and installing the software. During this process, some students might have gotten confused or frustrated and dropped the course before they even started the first assignment.

Analysis of the job market:

What skills are employers looking for in Android application developers?

After examining various online advertisements for “Android Developers” the following items emerged as important skills: learning the Android SDK, Java, XML, and communication skills. Below was a typical advertisement.

Responsibilities:

Design and implement mobile applications for the Android OS based platforms. Lead the design and development of a number of Android applications for tablet and smart phone platforms. Translate specs into technical requirements used to design the product. Keep up-to-date on the latest Android SDK developments. Respond to customer feedback and testing.

Requirements:

- Experience using and manipulating the Android SDK.
- Proficient with SDK, Eclipse, Java and XML.
- Demonstrable portfolio of Android and Android-related apps is a must.
- Your presence in the Android marketplace is a plus.
- BS in Computer Science, Information Systems, Computer Engineering, or equivalent experience.
- Ability to communicate and collaborate effectively with product management and team members.

As listed in the example above (and many other job advertisements), the word “portfolio” kept reappearing. Many employers are looking for employees with a strong portfolio that showcases their work. It is highly recommended that a portfolio requirement be added to any similar course.

Instructional materials

Should a textbook be required?

Since there are numerous websites containing free instructional material on Android development, it was decided not to require a textbook. For students wanting a hard-copy reference source, the textbook “Sams Teach Yourself Android Application Development” was listed on the syllabus as “optional.” Most students simply opted to complete the class using materials from the instructor and online resources. One website found very useful is (<http://developer.android.com>).

Should owning a mobile device that uses the Android OS be a requirement?

For the following two reasons, the instructor felt it should not be a requirement for students to own their own Android mobile device. First, the instructor felt this would possibly place some students in a financially difficult situation. Secondly, the Android editor comes with an emulator that allows the developer to view the work as it would appear on an Android device. The emulator can be configured to match any Android target (API level) and any resolution. However, at least in the instructor's opinion, a sense of complete understanding is not gained until the application has been downloaded, installed, and executed on a mobile device. Fortunately during the spring 2012 semester, 19 of the 26 students enrolled in the course owned either an Android based smart phone or tablet. Three of the students, not owning an Android based mobile device indicated they were seriously considering purchasing an Android smart phone, when their current phone contract expired. For students without access to an Android mobile device, the instructor would occasionally install the student's app on the instructor's smart phone so the students could view their final product on a "real" device.

Installing the software:

Is installing the software easy?

Installing the software in a computer lab was a difficult task for two reasons. First the following software applications must be installed.

- Java SE Development Kit
- Android SDK
- Eclipse

Next, at least one SDK Platform API had to be installed. We chose to install SDK Platform Android 2.1 API 7. At the time of this publication, an application created using this platform could be executed on approximately 97% of all Android mobile devices. It should be mentioned, that the software necessary to create applications for the Android OS is free. When installing this software in a restricted computer lab, it is important to make sure the software is installed so that students can read/write to any relevant files.

The second big challenge was making sure students could view their work and take screen shots of their smart phones. In some cases, additional software had to be installed (e.g., device drivers) to allow students to perform screen shots of their particular phone. Fortunately, installing the software "Samsung Kies" [[HREF 1](#)], allowed most students the capability to take screen shots. For the students to view the apps they created on their own mobile device, it was also necessary for students to change their settings to allow installation of non-Market applications (see Figure 1) and allow USB debugging (see Figure 2).



Figure 1: Allowing installation of non-Market apps

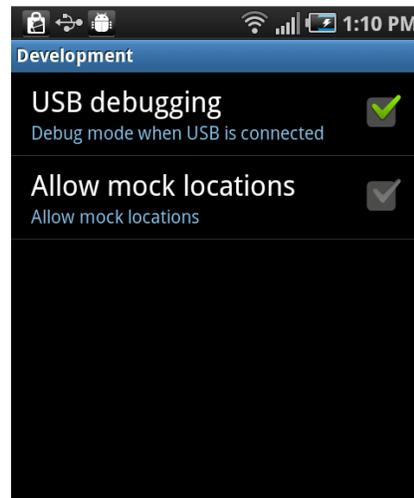


Figure 2: Allowing USB debugging.

Prerequisites:

Should there be prerequisites for this course?

It was decided that no prerequisite courses were required to enroll in this class. This decision allowed any student interested in this topic to enroll in this course. It should be mentioned that any programming course (in particular the programming language Java), any understanding of XML, or any knowledge of HTML would be beneficial.

Structure of the class:

How was the class structured?

The class was taught in an “optionally” blended format on a Tuesday/Thursday schedule. The instructor was present on both days. However, for the students, attendance was required on Tuesdays and optional on Thursdays. On Tuesdays new content was introduced, new assignments were made, homework was collected, and graded work was returned. The class was taught in a computer lab using a very hands-on approach to learning. Typically, assignments consisted of two somewhat similar exercises as shown in Figures 3 and 4.



Figure 3: Simple Calculator

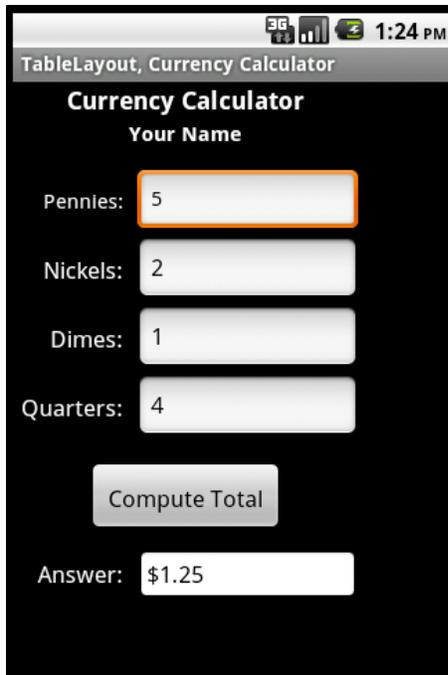


Figure 4: Currency Calculator

For example, during the Tuesday class, the instructor walked the students through completing the application shown in Figure 3. This included: providing an overview of relative layouts, walking the students through the XML code to create the layout, and writing the JAVA source code to implement the app. The students then had one week to create the application shown in Figure 4. During Thursday classes, which were optional, the instructor would answer any questions, especially those dealing with completing the exercise shown in Figure 4. Typically, about 50% of the students would need help and would attend Thursday's class as well. The remaining students, would either figure out the assignment themselves, or consult the web, or the class discussion board for help. Below is the partial XML code to create the interface shown in Figure 3. The complete XML code can be found in [HREF 2].

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

<TextView
    android:id="@+id/space"
    android:layout_width="wrap_content"
    android:layout_height="20px"
    android:textColor="#FFFFFF"
    android:text="  "/>

<TextView
    android:id="@+id/txtNumber1"
    android:layout_width="80px"
    android:layout_height="35px"
```

```

    android:textColor="#FFFFFF"
    android:layout_margin="5pt"
    android:layout_below= "@id/space"
    android:text="Number 1:"/>

<EditText
    android:id="@+id/number1"
    android:layout_width="100px"
    android:layout_height="wrap_content"
    android:text="4"
    android:textSize="20px"
    android:layout_below="@id/space"
    android:layout_toRightOf="@id/txtNumber1" />

    // ...
</RelativeLayout>

```

Below is the Java source code to implement the simple calculator example shown in Figure 3.

```

public class calculatorActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button button1;
        button1 = (Button) findViewById(R.id.calculate);
        button1.setOnClickListener(new clicker());
    }

    class clicker implements Button.OnClickListener {
        public void onClick(View v) {
            float num1, num2, totalAmount;
            EditText number1 = (EditText) findViewById(R.id.number1);
            num1 = Float.valueOf(number1.getText().toString());

            EditText number2 = (EditText) findViewById(R.id.number2);
            num2 = Float.valueOf(number2.getText().toString());

            totalAmount = num1 + num2;
            TextView answer = (TextView) findViewById(R.id.answer);
            answer.setText(String.valueOf(totalAmount));
        }
    }
}

```

Throughout the semester, 15 homework exercises similar to the one above were assigned. Additionally, an exercise was assigned the first day of class that made sure students knew the basics and advanced features of Android mobile devices. Some of the items on this first task are shown in the table below. The complete Exercise #1 assignment can be viewed at [\[HREF 3\]](#).

1. Add an application icon to your **Home Screen**

2. Remove an application icon from your **Home Screen**
3. Create a new folder
4. Add a bookmark to your **Home Screen**
5. Change the input method (Swype, Android keyboard)
6. Display the battery level
7. Display Android (Firmware) version
8. Allow installation of non-Market applications
9. Allow USB debugging
10. Read a bar code or QR (Quick Response) code using an app like **Red Laser**
11. Display the Longitude and Latitude of your current location using an app like "GPS Status"
12. Open the "Task Manager"
 - View all active applications
 - End all active applications

A final project and a final exam were also assigned. The final project involved the students creating a more advanced app with at least the following:

- Splash screen
- Main menu
- Settings
- Help screen

Portfolio: During this class the students were highly encouraged to create a portfolio of their work that showcased the Android apps that they created during this class.

Student Presentations:

Should student presentations be a requirement?

Many job advertisements listed excellent communication skills and being up-to-date on the Android developments as requirements. To help improve the communication skills of the students and to help familiarize the students with some of the thousands and thousands of apps available for Android devices, students were required to create a 3-5 minute multimedia presentation over an Android app that they had downloaded. All presentations had to be over different apps. Presentations had to include a minimum of 5 screen captures. This requirement forced the students to learn how to take a screen capture of their mobile device. It also encouraged the students to present on more advanced apps, since basic apps like "Magnifying Glass" and "Flash Light" only used a single screen. The presentation was worth 50 points. Like most presentations, points were awarded for delivery and content. However, points were also assigned for marketability (i.e., trying to convince the audience to download the Android app). Shown below is a condensed breakdown of points. The complete "Presentation Evaluation Form" can be viewed at [HREF 4].

Delivery:

- The presenter maintained strong eye contact and used effective gestures
- The presenter seemed enthusiastic about the topic
- The presenter was dressed appropriately
- The presenter seemed knowledgeable of the subject

Content:

- Introduction - gained attention and interest / introduced smart phone app
- Body - presented main features of the app / organized / performed background research
- Conclusion - reinforced features and name of the smart phone app

Marketability:

The presentation encouraged the audience to download the Android app that was presented. A role play situation may be established; where the speaker asks the audience to assume they are members of a particular group. For example, the speaker may ask the audience to pretend they are active joggers before presenting on the app “RunKeeper” or “Buddy Runner”.

CONCLUSION

In this paper a case study was performed over designing a new course on developing applications for Android based mobile devices. In particular the following questions were addressed:

- Are students interested in taking a course on this topic?
- How should the class be taught?
- What skills are employers looking for in Android application developers?
- Should a textbook be required?
- Should owning a mobile device that uses the Android OS be a requirement?
- Is installing the software easy?
- How was the class structured?
- Should student presentations be a requirement?

Hopefully, the answers to these questions provide some insight into some of the problems and trade-offs encountered in designing a new course over developing Android applications.

REFERENCES

HREF 1: Samsung Kies

Available: <http://www.samsung.com/us/kies/>

Cited: February 1, 2012

HREF 2: XML code for simple calculator

Available: <http://arapaho.nsuok.edu/~rosener/papers/android-case-study/>

File name: calculator-xml-code.html

Cited: February 1, 2012

HREF 3: Exercise #1 - basics and advanced features of Android mobile devices

Available: <http://arapaho.nsuok.edu/~rosener/papers/android-case-study/>

File name: exercise1.html

Cited: February 1, 2012

HREF 4: Complete Presentation Evaluation Form

Available: <http://arapaho.nsuok.edu/~rosener/papers/android-case-study/>

File name: presentation.html

Cited: February 1, 2012