

# MINI 1 SAMPLES that all earned a 5/5 evaluation.

## NOTES:

- a. All these samples happen to be from the same MINI 1 submission of a single student.
- b. This student preferred to write a précis for each source. That is one method. It is also possible to write a single, longer précis, covering all the sources for one topic.

## 1: ANDROID OPERATING SYSTEM

- a) [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))  
<http://www.slideshare.net/HunkaTech/google-android-operating-system-uses-and-features-in-a-nutshell>  
<http://www.android.com/about/>
- b) Android is a Linux based operating system built and maintained by Google. It was developed to be used primarily on tablets and smart phones instead of laptops and desk top computers. As such it natively supports a wide array of connectivity methods such as GSM, Edge, CDMA, Bluetooth, Wifi. It natively integrates with many google services, which allows app designers to leverage those tools as well in their work and supports SMS, MMS, Threaded Text Messages and C2DM protocols. Being designed for lower power and memory systems, it has an “as needed process and memory management system that only loads system processes that are needed when they are needed, as well as an array of more or less effective power management tools to maintain battery life on mobile devices. It also provides native support for an array of interface options like multi-touch, accelerometers, touch screens, cameras, and video cameras

Android is built on a modified version of the Linux Kernel, and is a mostly open operating system. It can be modified to work in most of the situations that other forks of linux can be made to work in. Android uses the Linux “ask for permission first” model for system resource access. When your users aren’t actively working against it, this means that android provides a reasonably secure system. It does not natively support windows style file management and some customization features that seem like they should be part of the operating system need to have third party software in order to function. But since android is tied into the Google Play store, it is usually simple to find free or cheap solutions to this sort of problem.

- c) The key features of the android operating system, from the view point of an application developer are the platform portability and the native support for a wide array of functions and tools. Instead of designing applications for a single type of machine, an Android developer should keep in mind the array of forms and functionalities of the hardware an android user might be using their application on. On the other hand, many features that in another operating system, you would have to code by hand or find other people’s solutions for are part of the basic tool kit for Android OS.
- d) Will Android ever gain native file management interfaces? (I’d really like to not need 3<sup>rd</sup> party software to make shortcuts to text files on my main screen)

## 2: JAVA AND XML IN ANDROID

- a) <http://www.ibm.com/developerworks/xml/library/x-android/index.html?ca=drs->  
<http://programmers.stackexchange.com/questions/188128/java-xml-intracction-in-android>  
<http://www.sitepoint.com/menus-in-android/>
- b) In Android applications, XML serves two main purposes. The first is to act as a standard method of interacting with web resources. Much of the data on the internet is formatted using XML, so any application that interacts with data on the internet, either sending or receiving, is likely to need to use XML. The Android version of Java does not natively support every xmp parser that is part of the standard Java SDK, but it supports several popular ones and has its own parsers that emulate a few others. The supported DOM and SAX parsers will fulfill most XML needs.

The second use of XML in Android applications is to control UI layout features. While this could be done as pure hardcoded Java, using XML allows a programmer to generalize their UI for multiple different screen sizes and types as well as different types of machines that might be running an android application. Since Android devices cover an array of things from 3 inch cell phone screens to 10 inch tablets to home computers, the ability to use xml files to control the UI allows the programmer a lot of flexibility in their work without recoding for each system.

- c) In the Android OS, the two main uses of XML are to build and control UI layout across a wide array of device formats and to parse and send data to and from the internet. The java interpreter for Android supports several XML libraries, mostly aimed at quick and light weight use of the language.
- d) Just how deep of a knowledge of XML does one need to use these features?

### 3: ANDROID DESIGN GUIDELINES

- a) <http://developer.android.com/training/best-performance.html>  
<http://developer.android.com/training/best-ux.html>  
<http://developer.android.com/design/patterns/app-structure.html>
- b) Android application design has two competing and complementary sets of best practices. The first set focuses on improving software performance, which is particularly important when you are developing applications for an OS where there is such a wide array of hardware that is supported by the OS. The second is best practices for UI/UX development. This is how you create a uniform feel for the users of your application and thus keep from confusing users.

In broad strokes, designing for performance consists of two things: Don't do unnecessary tasks. This is process optimization, and consists of removing unnecessary steps and making sure that necessary things only happen when they are actually needed. The second is don't allocate memory if you can avoid it. Modern computer architecture deals with that more fluidly than you can if you do so ahead of need. These goals can be furthered by avoiding things like the creation of unneeded objects, none of which occur for free. Before you ever try to optimize for performance, make sure that you can accurately measure your current performance. If you can't compare future performance with the original baselines, you have no way to see if you are doing anything useful with optimization.

Designing for User Experience involves tasks like making sure that your app gives feedback when things go wrong so that the users know what is happening and can try to fix problems instead of just guessing. You should always make screens run from general information to specific, and the interface should be more visual than textual. Navigation from general information to specific information and back should be easy to understand and it should always be possible to track what the user has done. Whenever information is presented in a textual format, it is important to use the included search functionality to aid in access. Especially given the small size of many android screens, doing visual searches of text is difficult and frustrating. Make sure that in every screen, objects that do related tasks are near each other and that on each screen, an object that does a specific task always does that task everywhere it appears. Make sure that you know the current standards so that you can make your application respond to inputs in the same way that other current applications do. Unfortunately, this is subject to very frequent change.

- c) Best practices for Android development span two different goals. The first is a software oriented one, keep your work simple and flexible because it is running in an environment with very limited resources. Make use of native memory management tools and optimize whenever possible. The second is user experience oriented. You

should make sure that the design affordances of your application are similar to the user's expectation. Present visual information when ever possible and make navigation and feed back very clear.

d) Exactly how often do the UI standards change in the Android community?