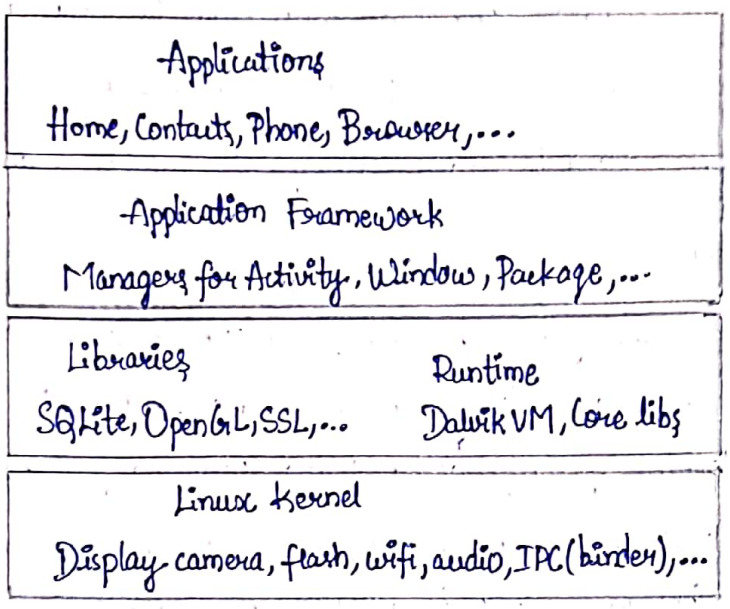
Mobile Application Development
Assignment - 1

G. Kaushik Raju
CSE-B, R19CS452

1) Describe in detail about five sections and four main layers of Android Operating System with a neat diagram.

A)



(*) Linux Kernel:

(*) At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches.

(*) This provides a level of abstraction b/w the device hardware and it contains all the essential hardware drivers like camera, keypad, display, etc.

(*) The kernel take the pain out of interfacing to peripheral hardware.

(*) Libraries:

(*) On the top of Linux kernel there is a set of libraries including open-source web browser engine WebKit, well known library lib.

(*) SQLite database which is a useful repository for storage and sharing of application data.

(*) Libraries to play and record audio & video.

(*) Android Libraries:

(*) android.app: Provides access to the application model and is the cornerstone of all Android applications.

(*) android.content: Facilities content access, publishing & messaging b/w applications & application components.

(*) android.database : Used to access data published by content providers and including SQLite database management classes.

(*) android.opengl : A Java interface to the OpenGL ES 3D graphics rendering API.

(*) android.os : Provides applications with access to standard OS services including messages, system services & inter-process communication.

(*) android.text : Used to render & manipulate text on a device display.

(*) Android Runtime :

(*) This is 3rd section of the architecture & available on the second layer from the bottom.

(*) This section provides a key component called Dalvik VM which is kind of Java Virtual Machine specially designed and optimized for Android.

(*) The Dalvik VM makes the use of Linux core features like memory management & multi-threading, which is intrinsic in Java Language

(*) Application Framework :

(*) The Application Framework layer provides many higher-level services to applications in the form of Java classes.

→ The Android Framework includes the following key services

(*) Activity Manager : Controls all aspects of the application lifecycle & activity stack.

(*) Content Providers : Allows applications to publish & share data with other appln's.

(*) Resource Manager : Provides access to non-code embedded resources such as strings, colors, settings & user interface layouts.

(*) Notifications Manager : Allows appln's to display alerts & notifications to user.

(*) View System : An extensible set of views used to create application user interfaces

(*) Application Applications :

(*) You will find all the Android application at the top layer. You will write your application to be installed on this layer only.

Examples :- Such applications are Contacts Books, Browser, Games, etc.

1) Activities : They dictate the UI & handle the user interaction to the smart phone screen.

2) Services : They handle background processing associated with an application.

3) Broadcast Services : They handle communication b/w Android OS & appln's

4) Content Services : They handle data & database management issues.

2) Explain the five Android application development fundamentals

A) 1) Master the Language :

(*) Java & XML are the two main programming languages used in Android App development.

(*) Some of the fundamentals of the Java programming language include :

(*) Packages
(*) Objects & classes
(*) Inheritance & interfaces
(*) Strings & numbers, generics
(*) Collections
(*) Concurrency.

2) Familiarity with Right Development Tools & Environment :

(*) Android app studio IDE or the Eclipse for the tools

(*) They will help you learn the basics and many other things that will help improve your code.

(*) Apache Maven, Apache Ant, Gradle as they provide a powerful set of tools to help in managing your builds.

3) Knowledge of the Application Components :

(*) Application components are the essential building blocks of Android app development.

(*) Each of the components is different point by which the system can enter the your app

(*) There are five different point types of app components.

1) Activities : This is component that represents a single screen with a user Interface

2) Services : This component runs in the background to perform work for remote processes or long-running operations. It does not provide user Interface

3) Content providers : This is the component that manages a shared set of app data Through this component, the data that you store either in the file system, on the web, a SQL database can be queried or even modified.

4) Broadcast receivers : This is the component that responds to system-wide broadcast announcements. They can create a status bar notification that alerts the user when a broadcast event occurs. It is a gateway to the other components.

5) Activating components : A synchronous message whether the component belongs to your app or not.

4) Awareness over Fragmentation, Android Application, Threads, Loaders & Tasks :

(*) Android is fragmented with many different devices & OS Versions.

(*) If your device supports more OS versions it will definitely require more maintenance & testing.

(*) You also require appropriate fonts, assets & layouts that will help in ensuring that the best possible experiences in the various screen characteristics are given.

(*) Services for background tasks that should run continuously.

(*) The long operations (computations, I/O, network, etc.) should all be run asynchronously in the background

5) Making the Right Choice over Needed Tools :

(*) The simple tools that you need for Android App development are just a Mac or Windows PC, any type of Linux, & Eclipse, the ADT Plug (Accessory Development Tool) is a plugin for Eclipse

(*) ADK (Accessory Development kit) is used for implementation of hardware use as a starting point for building accessories for Android.

(*) Each ADK release is provided with source code and hardware specifications to make the process of developing your own accessories in, and the Android SDK (Software Development Kit) - all of which are free.

(*) Android has some unique parameters that you should consider when writing an android app:

1) Performance & responsiveness: You should always respond to user input within five seconds otherwise the operating system will answer you.

2) Lags of more than 100ms will be noticed by the user: As mentioned above, the UI thread should never be blocked because it is only one.

3) With a neat diagram (Window/Dialog boxes) explain the Android Virtual Device which is the easiest way of testing the application.

A) (*) An Android Virtual Device (AVD) represents a device configuration. Based on configuration we can able to test the compatibility of the application over the configured device.

Ex: you can create an AVD that represents an Android device running version 4.1 of the SDK with a 64MB SD Card.

(*) After creating AVDs, you point the emulator to each one when developing and testing the application.

(*) AVDs are the easiest way of testing the application with various configuration.

(*) To create AVDs in Eclipse, select the window, AVD Manager option.

(*) An Android Virtual Device Manager dialog opens.

(*) The dialog box displays a list of existing AVDs.

(*) Letting you create new AVDs and manage existing AVDs.

(*) Selecting the New button to define a new AVD. A Create new AVD dialog box, appears. The fields are as follows:

Name - Used to specify the name of the AVD.

Target - Used to specify the target API level. Our application will be tested against the specified API level.

CPU/ABI - Determines the processor that we want to emulate on our device.

SD Card - Used for extending the storage capacity of the device.

Snapshot - Enable this option to avoid booting of the emulator and start it from the last saved snapshot. Hence, this option is used to start the Android emulator quickly.

Skin - Used for setting the screen size.

Hardware - Used for set properties representing various optional hardware that may be present in the target price.

4) The Android emulator is used for testing and debugging applications before they are loaded onto a real handset. The Android Emulator is integrated into Eclipse through the ADT plug-in. Explain the limitations of android Emulator.

A) Limitations of the Android Emulator :

(*) The Android emulator is useful to test Android applications for compatibility with devices of different configurations.

(*) Emulators no doubt help in knowing how an application may operate within a given environment, but they still don't provide the actual environment to an application.

(*) For example, an actual device has memory, CPU or other physical limitations that an emulator doesn't reveal

(*) Emulator's just simulate certain handset behaviour.

(*) Features such as GPS, sensors, battery, power settings, and network connectivity can be easily simulated on computer.

(*) SMS Messages are also simulated and do not use a real network.

(*) Phone calls cannot be placed or received but are simulated.

(*) No support for device-attached headphones is available.

(*) Peripherals such as camera/video capture are not fully functional.

(*) No USB or Bluetooth support is available.

5) Explain the android framework key Services

A) Application Framework :

The Application Framework layer provides many higher-level services to applications in the form of Java classes.

The Android Framework includes the following key services

(*) Activity Manager - Controls all aspects of the application lifecycle and activity stack.

(*) Content Providers – allows applications to publish & share data with other applications.

(*) Resource Manager – Provide access to non code embedded resources such as strings, color settings and user interface layouts.

(*) Notifications Manager – Allows appln's to display alerts & notifications to the user

(*) View System – An extensible set of views used to create application user interfaces.

6) Application components are the essential building blocks of Android app development. Explain five different types of app components.

A) The five different types of app components.

(*) Activities – This is a component that represents a single screen with a user interface.

(*) Services – This is a component which runs in the background to perform work for remote processes or long-running operations. It doesn't provide user interface.

(*) Content providers – This is the component that manages a shared set of app data. Through this component, the data that you store either in the file system, on the web, a SQL database can be queried or even modified.

(*) Broadcast receivers – This component responds to system-wide broadcast announcements. They can create a status bar notifications that alerts the user when a broadcast event occurs. Generally, It is a gateway to the other components.

(*) Activating components – A synchronous message whether the component belongs to your app or not.

7) Describe how the android application can be created using the Eclipse with relevant example.

A) 1) To create the applin, Open Eclipse & choose file, New, Android App Project.

(or)

Click on Android Project Creator icon on the Eclipse toolbar.

2) A dialog box appears asking you to select the wizard you want to use for the new applin.

3) Select - Android appl'n project & click on Next
4) Give the project / appl'n name as HelloWorldApp
5) Then select the package required for the appl'n which is

Com.example.helloworldapp

6) Give the package name as com.androidunleashed..helloworldapp.

There are two important parts during the Creation of Android appl'n which can be reflected to important files of the Android

1) The XML file, which will be residing under res / layout folder - the file where action code of the controls defined in the layout file

activity_hello_world_app.xml

This file defines the user interface of the appl'n.

This file contains various controllers such as text views, Button, edit text & check Box which are used for the user interface.

2) The java file, HelloWorldAppActivity.java, found in the src folder -

This files contains the action of the controller which are defined in XML file. There can be different events occurs via the controllers can be handled this java code.

XML is namespace which can be defined as the no. of attributes required to do the particular programming are defined by using the help of XMLns