CS 4720 Lecture 2

Two types of application:

1. Two-tier architecture
   1. Client apps ---- data source (server)
   2. No intermediate between the client and server (faster connection!)
2. Three-tier architecture
   1. Client, business, data layers

MVC – Model, View, Controller

*Controller*

-“traffic cop”

-takes request from user (w/ assistance of server & routing rules) turns it into a method of some sorts

-finds appropriate model/view to load

*Model*

-representation of the data

-may or may not be directly linked to a database (often is in larger apps)

-model is often translated directly into a DB table, with the columns as its attributes

-think “class definition w/ DB backend”

-often contains relationship rules (e.g. a Student has many classes, for instance)

*View*

-closest thing to what you’ve been dealing with so far is the view

-effectively an HTML template that will be populated with the appropriate data from the loaded model

-often has PHP (or whatever) embedded in it

-All UI components go here

**Android Architecture**

-general all apps are written in Java

-compiled Android app is an .apk

-android apps must be digitally signed in some way to execute

-this digital signature can be a debug certificate that comes default with any installation

-android is a multi-user linux OS

-each app is given its own user id when it is installed

-every app is given its own “sandbox” where the files are set to permissions only for that app to read and write

-every app runs in its own VM (i.e. code runs in isolation from other apps)

-every app is a separate Linux process

-possible for apps to share data with other apps if:

1. two apps can have the same Linux user id (and thus share resources) if and only if they are signed by the same digital certificate

2. apps can setup data sharing privileges through the permissions xml system in each manifest

Main Components

- activities – represent a single screen with a UI

- services – represents a process running in the background

- content provider – a link back to the data

- broadcast receiver – listens for system-wide messages to respond to

- application – a set of activities that make up a cohesive unit

- intent – a message to be passed

*Service*

-component that runs in the background to perform long-running operations

-service has no UI

-examples of services:

1. playing music in background

2. gathering GPS data

3. downloading a data set from the server

*Content Provider*

- manages a shared set of app data

- shared set of data could be a file, an SQLite DB, a remote link to a web service, etc.

- apps can query a content provider for data if they have permission

*Intent*

- is a message that requests an action from another component of the system; it is a message package

- everything that is transmitted from one activity to another activity is an intent

- includes the “please start up your App” intent that the system sends when a user clicks on your app icon

- an intent filter is where the android operating system sets up a registry of all the intents that the application should be watching out for

*Tying it all Together*

- The AndroidManifest.xml file

1. holds all of the parts together

2. sets up all permissions the user has to agree to (i.e. Internet, GPS, contacts, etc.)

3. declares the API level of the app

4. requests hardware features needed

5. needed libraries

6. which activities are part of this app