Khai, Phan Thanh

100901164

Mobile App Development (Android)

March 26, 2020

Desmond Lobo

**FINAL ASSESSMENT**

**(4 Marks each)**

1. **How are Adapter objects used in Android development?**
   * Act as a 3rd party that connects between **Adapter** **Object** and the content information use for that view. It has access to the data content and views the data content into AdapterView for display. In Lab5, we have two separate Adapter to display the flag country for each side, called CustomAdapterLeft and CustomAdapterRight. The main method creates an object of both CustomAdapter and passes appropriate data there. Each CustomAdapter then connects with a separate layout for display order. Once the order has decided, it parses the information it gets to from the main method to the field in the design by using discretion.
2. **What is the relationship between Content Providers and Content Resolvers?**
   * **Content Provider** help programmers to send data from one program to another using Content Uri. The format of the Uri can specify to as content://com.example.finalAssessment/final, whereas content:// is the Scheme with the value of content to know it is sharing the content, com.example.finalAssessment is the Authority (or sources location) and the /final is the path to the provider. Upon receives this Uri, the Content Resolvers quickly search for the Content Providers to find the matching Uri and then connect the request source with the result found together.
   * **Content** **Resolvers** need to specify where the program is looking to share data (by implementing the getContentResolver() method). When sharing the data, the Providers who are accessing can either create, read, update and delete upon the target. First, the Content Resolvers need to specify the Uri it received. Then create an object of ContentValues to hold the values from the database to the ContentValues purpose when injecting it, then depend on the CRUD method, the appropriate number of arguments is required to fill to complete the request and return a result.
3. **What is the difference between the ListView class and the RecyclerView class?**
   * **ListView** is a view displayed as a list of information (in horizontal) that has a separate field in between them. The number of views that needs to specify to the app can create the exact amount of view. Otherwise, the view can scroll to infinity. ListView uses more memory to load but ensures that data are ready by creating all the views at once.
   * **RecyclerView** is the same with ListView, but it saves memory by reusing the view that is out of the screen (in the direction of scrolling). What that means is, let say that we have on the screen a total of 5 rows, when we scroll up to display the next two view, the first two view that has scrolled are putting to the bottom to display information in the following two views. However, it can cause some trouble when the user is scrolling fast that causes the RecyclerView not to load the data on time. Previous data that has loaded remain when scrolling backward. Some features that RecyclerView has over ListView is the ability to scroll horizontal and vertical (only horizontal in ListView), support more layout than just a simple LinearLayout in ListView.
4. **What are Memory Leaks? Why should Android developers be concerned about Memory Leaks?** 
   * **Memory** **Leaks** is an event that occurs when the program is running, and some method/function is open. Still, without closing or destroying it at the end, so when the program closes, these external functions/methods are still run in the background and cost memory leak. Depends on the size of the program the memory leak could be between a few MB up to a GB
   * One of the popular Memory Leaks in Java is user input control. Sometimes, develop when they create a new input for users tend to forget to close them, and so when the app finished, the input method is still run in the background and cause some memory leak.
5. **What are the security risks associated with Rooting your Android smartphone?** 
   * While **rooting an Android phone** can bring you from a regular user with limited power to a superuser with almost unlimited power on the phone and many more advantages, it can also contain some security risks that user has to exchange.
   * Before rooting the phone, the user must know about their devices and the system very well because it is effortless to mess up with the system data, especially when installing an app from unknown sources, where they can secretly put a malware inside the app. Once the user roots their phone, some security features that prevent pirates app to install and block a user from messing their phone has been disabled.
   * Because when the phone is rooted, almost every user who understands how phone works can access your data, and it is dangerous. Some trusted sources app had installed an API called SafetyNet that allows the app to check for the state of the phone before installing. Once the app installed, it could reveal some of the information that the user has entered previously and then accidentally delete the app (for example, Samsung Pay contents card information so a user can tap on their phone to make a payment anywhere).
   * The other issue would be warranty void when the manufacturer found the phone been root because you can unlock some of the hardware features that are strongly discouraged by the manufacturer (such as CPU and GPU overclocked that could potentially damage the phone). Also, rooting your devices means you can no longer access to OS update
   * An example of this would be, let say, a user is using Samsung Pay as a method for payment and has added several cards. These cards ID when the phone isn’t root is going to store as a cache and session cookie in place that is not available to access by the user (the disk on Android phone created a partition to store this information, and generally, this disk always keep private). Different from the rooted phone, the caches information is available for almost any user who knows the path (to find the data files) and take the information. Although the data are encrypted, which one is better, not releasing files to have no risk at all or to take some risk and reveal the file?