**Broadcast Receivers**

I learned a lot about Broadcast Receivers from this codelab. I learned how to create and implement a Custom Receiver class that extends the BroadcastReceiver, how to register and unregister for system broadcast intents, and how to send and receive a custom broadcast intent. Broadcast Receiver is one of the major components of an Android app. Broadcast Receivers can receive broadcasts from the system or by other apps. In the BroadcastReceiver class, the onReceive() method must be implemented in order to receive/process incoming broadcast intents. A BroadcastReceiver can be either static or dynamic receiver. To register a receiver statically, you must use the <receiver> element in the manifest xml file. On the other hand, a receiver can be registered dynamically in the app or activity context. Starting from Android 8.0 (API level 26 and higher), you can't use static receivers to receive most Android system broadcasts, with some exceptions.

There are three ways to send custom broadcasts, which are:

1. Normal broadcasts: are asynchronous. To send a normal broadcast, one must create a broadcast intent and pass it to sendBroadcast(Intent).
2. Local broadcasts: are send to receivers in the same app as the sender. To send a local broadcast, create a broadcast intent and pass it to LocalBroadcastManager.sendBrocast(Intent).
3. Ordered broadcasts: are delivered to one receiver at a time. One receiver can hold up the result to the next receiver. To send a broadcast, you must pass the intent to sendOrderedBroadcast(Intent, String).

Local broadcasts are private to the app and to register and send broadcasts, one must use the LocalBroadcastManager. A common practice in creating a unique intent action broadcast names is by prepending the action name with your package name. If your app targets API level 26 or higher, static receivers and registrations are not allowed.

**Answer these questions**

**Question 1**

What is a system broadcast?

* A message that your app sends and receives when an event of interest occurs in the app.
* A message that is sent from an app to a different component of the same app.
* **A message that the Android system sends when a system event occurs.**
* A message that the Android system receives when an event of interest occurs in your app.

**Question 2**

Which pair of methods do you use to register and unregister your broadcast receiver dynamically?

* registerBroadcast() and unRegisterBroadcast().
* registerComponentCallbacks() and unRegisterComponentCallbacks().
* registerBroadcastReceiver() and unRegisterBroadcastReceiver().
* **registerReceiver() and unRegisterReceiver().**

**Question 3**

Which of the following are true?

* **Broadcast receivers can't see or capture the intents used to start an activity.**
* **Using a broadcast intent, you can't find or start an activity.**
* You can use a broadcast intent to start an activity.
* You can receive the intent used to start activity in your broadcast receiver.

**Question 4**

Which class is used to mitigate the security risks of broadcast receivers when the broadcasts are not cross-application (that is, when broadcasts are sent and received by the same app)?

* SecureBroadcast
* **LocalBroadcastManager**
* OrderedBroadcast
* SecureBroadcastManager