ECM2425 Mobile and Ubiquitous Computing

Student No. 690021357

Question 1:

a) Fragment offers to compose apps in a highly modular way. It offers flexibility within the app and can be reused. Disadvantages include increasing complexity, making it difficult to read and edit. Also there may be security concerns.

b) Content provider is an abstraction layer between data source and user interface. First, we need to get permission, then get ContentResolver. Followed by picking one of four basic actions: query, insert, update and delete. Afterwards, identify the data to create a URI, and finally, display in the UI.

c) Custom broadcast intent example can be when other apps need to be notified that downloads has been completed and is available. System broadcast intent can be ACTION_BATTORY_CHANGED, record system state changes on battery level for example.

d) A notification is a message your app displays to user outside app's user interface or it can be shortcut to other apps. Design principle are relevance, whether the information is essential for user; timely, only appear when they are needed; and it should be short, includes as few words as possible. It should also allow users to choose the kinds of notifications and how they want to receive them.

e) Main difference between these two views is that generic view doesn't update as frequently as SurfaceView. Drawing to generic view is subclass of the View class. Steps include: create bitmap, associate bitmap with view, then create canvas with bitmap object, then use paint, followed by draw in onDraw(), then draw in onSizeChanged() when size change, and finally, call invalidate() to redraw. Drawing to surface view includes defining SurfaceView, set it up, and then draw. If SurfaceView isn't visible, implement the created and destroyed.

f) Expresso testing framework for testing views, it finds the view, specify the view we are interested in. then action on the view specifies the action to be performed, and finally check if it behaves correctly, check if everything is expected to happen will happen. For AdaptorView, onView() in expresso may not be able to find the necessary view in AdapterViews, which loads data dynamically from an adaptor. onData() loads the adapter item onto the screen before operating on it. Lastly, uses a dataoption method such as inAdaptorView() or atPosition() to specify the item in the AdapterView.

Question 2:

a) onPause() or onStop() will not be invoked if finish is called from within the onCreate() method. This might happen when you try onCreate() but detects an error while doing so, and finish is called before reaching onStop() or onPause().

b) Change configuration: onPause() onStop() onDestroy() onCreate() onStart() onResume()

c) Task back stack will first stack activity a when activity b is launched, then activity b will be the foreground and b will be stacked again on top of a when intent sent back to a. so stack should look like this: [a], [b, a], [a, b]

d) It is not safe, because the feature depends on an external activity, we need to test for availability before using it. Therefore, correct code:

```
Intent sendIntent = new Intent();

sendIntent.setAction(Intent.ACTION_SEND);

sendIntent.putExtra(Intent.EXTRA_TEXT, textMessage);

sendIntent.setType(HTTP.PLAIN_TEXT_TYPE);

if(sendintent.resolveactivity(getpackagemanager()!=null)){

        startactivity(sendIntent);

 }
```

Question 3:

a) In the activity context, broadcast is received if the activity is still running and not stopped. But for application context, broadcast is received if the application is still running.

b) If you register a broadcast receiver in onCreate(), you should unregister it with onDestroy() because it can prevent leaking receiver out of activity. If you register with onResume() you should unregister it with onPause(), to prevent registering multiple times.

c) Apply() will plan for the data to be written asynchronously. It will commit without informing the success or failure of operation. Commit() will plan for the data to be written synchronously, which means that it will inform you whether the operation was successful or not.

d) The limitation of Asyntask is that it is created along with an activity, but it is not destroyed along with the destruction of an activity. If we do not manually cancel Asyntask before activity is destroyed, it will keep running in the background and it can cause a memory leak as the memory of the activity will not be destroyed. Asyntask should be used for simple network operation which do not require downloading a lot of data.