Rushabh Shah
CPRE 388
Section F

HW 4

1. Question 1 - Intents
   a. Explain the difference between implicit and explicit intents.
   ➔ Implicit intent: This type of intent does not provide specific target. These are generally used by outside applications.
   Explicit intent: This type of intent provides a specific target, the class name, to be used for the action. This type of intent is generally used for internal messages.

   b. How does the Android framework the target of an implicit Intent? What happens if there is more than one target?
   ➔ Android framework uses intent filter to search for components that is capable of performing the task specified by implicit intent. In the even that there are more than one components are found, android framework will provide user a list of options to choose from.

   c. What changes are needed in manifest file to enable an activity to respond to one or more intents – explicit and implicit? Show the manifest file sections for an activity which can respond to multiple intents
   ➔ Android manifest file of the application must contain intent filters for each activity that is capable of handling specified tasks, such as action, category or data. Following is an activity tag of Android Manifest file that can respond to request for opening pure text, image and video.

```
<activity android:name=”ViewActivity”>
    <intent-filter>
      <data android:mimeType=”image/*” />
      <data android:mimeType=”video/*” />
      <data android:mimeType=”text/plain” />
    </intent-filter>
</activity>
```

   d. Why is it recommended to explicitly specify the target of a Pending Intent? Support your answer with scenarios where user experience is degraded by using implicit target.
   ➔ Let’s suppose user has multiple email applications installed, Gmail and Outlook and both Gmail and Outlook have registered activity that can handle email viewing.
   Now say user gets a Gmail notification. Now if user clicks on the notification, he/she expects to see her detailed email in Gmail’s specific activity and not any another. Thus, use of explicit intent is required to ensure intent is handled correctly. User will definitely not be happy, if a Gmail notification was handled by Outlook.

2. Question 2 – Broadcast Receivers
   a. What is the difference between normal and ordered broadcast?
   ➔ Ordered broadcast: An ordered broadcast is sent synchronously to the registered receivers.
   Normal broadcast: Normal broadcast is also sent to everyone, though there is no defined order in which the receiver will receive broadcast.

b. The intent object acts as the communication channel for starting activity and also sending broadcasts. How are these two scenarios different?

➔ Activity: In this context, Intents are used for inner communication (in case of explicit intent).
Broadcast: In this context, Intents are used for outside communication and are generally associated with some form of action.

3. Question 3 - Services

a. How does service differ from background threads?

➔ Main difference between service and background threads is user interaction.
Background thread is used when some computation or data manipulation needs to be done while user is interacting with the activity. A background thread would allow UI to be responsive to user interaction.
Service runs in the background even when user is not interacting with application, for example a music player service that plays music even when screen is turned off.

b. How does a client bind to the service component of an app?

➔ Service component must implement a onBind() callback and return an IBinder interface.

c. Explain how to perform IPC using messenger object between a client and service.

➔ IPC can be performed by creating a messenger object. The service will need to define a Handler that will response to clients' requests in form of Message objects. The handler will share an IBinder with client to communication. Messenger will create a thread containing queue of requests and then they will be handled sequentially.

d. What is the difference between three return values from onStartCommand() callback?

➔ Three possible return value and their differences are:
START_NOT_STICKY: If the service is killed by system after callback returns, DO NOT recreate service UNLESS there are pending intents to deliver.
START_STICKY: If the service is killed by system after callback returns, recreate service but DO NOT redeliver the last intent. Use pending intents if available otherwise use null intent.
START_REDELIVER_INTENT: If the service is killed by system after callback returns, recreate the service with last delivered intent.