Intents and Intent Filters
Intent

- Intent is an messaging object. There are three fundamental use cases:
  - Starting an activity:
    - `Intent intent = new Intent(this, SecondActivity.class);`
    - `startActivity(intent);`
  - Starting a service:
    - A service is a component that performs operations in the background without a user interface.
  - Delivering a broadcast:
    - A broadcast is a message that any app can receive.
Types of Intents

• Explicit intents: specify the component to start by class name

• Implicit intents: do not name a specific component, but instead declare a general action to perform, which allow a component from another app to handle it.
  – implicit intents are handled by the Android system and it in turn chooses the best suited activity to perform the required action as stated in the intent.
Security Hazard

• Caution: To ensure that your app is secure, always use an explicit intent when starting a Service and do not declare intent filters for your services. Using an implicit intent to start a service is a security hazard because you can't be certain what service will respond to the intent, and the user can't see which service starts.
Intent Filters

• Intent Filters are used with implicit intents.

• Activities which want to handle implicit intents will associate themselves with Intent Filters. As the name itself suggests, an Intent Filter essentially signifies what types of intents an activity would like to receive and which ones it would like to filter.

• Intent Filters are defined in the manifest file using <intent-filter> tag. An Intent Filter has fields which map one to one with the action, data, and category fields of an Intent object. For intent to be delivered to an implicit activity, it must pass all three criteria of action, data and category.
Intent Filter and its associated fields (action, data and category)

• Action
  – ACTION_VIEW: such as a photo to view in a gallery app, an address to view in a map app
  – ACTION_SEND: such as an email app or social sharing app

• Category
  – CATEGORY_BROWSABLE: such as a web browser
  – CATEGORY_LAUNCHER:

• Data
  – The URI (a Uri object)
Implicit intent

- Implicit intents specify the action which should be performed and optionally data which provides content for the action.

- For example, the following tells the Android system to view a webpage. All installed web browsers should be registered to the corresponding intent data via an intent filter.

  ```java
  Intent i = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.com"));
  startActivity(i);
  ```

- If an implicit intent is sent to the Android system, it searches for all components which are registered for the specific action and the fitting data type.

- If only one component is found, Android starts this component directly. If several components are identified by the Android system, the user will get a selection dialog and can decide which component should be used for the intent.
Explicit intent vs. Implicit intent

• Android Intents to trigger actions in your own apps (**explicit**). Explicit Intent directly define the target component with the name of the component.

• But developers often do not know the other components of the app name (**implicit**) where Android resolves matching intents which Not specify the name of the target component. it is more widely used to pass messages between different applications.

• In the explicit Intent message, decided the only elements of the target component is the component name, Therefore, if you Intent has been clearly defined in the name of the target component, then you completely do not have to define the contents of the other Intent.

• For implicit Intent , because there is no clear target component name, Android system must find components that most closely matches the intent of the request with Intent.

• The IntentFilter contains all the possible components to be selected. IntentFilter matches a component implicit Intent requested content based on the specs in the intentfilter in Android-the Manifest.xml (that can match what Intent request).
Register a component of our app that can receive an implicit intent

- **Note:** An explicit intent has its target specified (while creation) does not need registration.

- To register a component of our app that can receive an implicit intent for a specific action and data, we can declare an `<intent-filter>` for that particular component (activity, service or a broadcast receiver) in the manifest file.

```xml
<activity android:name="com.pycitup.pyc.WebViewActivity">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="http" />
    </intent-filter>
</activity>
```

- Make sure that both the sending and receiving activities have requested permission for the type of action (ACTION_VIEW) to be performed. Need the INTERNET permission in our manifest:

```xml
<uses-permission android:name="android.permission.INTERNET" />
```

- When the intent code is fired, the user will be given multiple choices among which one of them will be our own app that’ll trigger WebViewActivity. Inside the Activity you could then do things like show up the webpage in a WebView.
An implicit intent tells Android system to run an activity (ActivityB in this case) that can do these things that are specified in the intent filter in a manifest.

```java
Intent intent = new Intent();
intent.addAction("thisAction");
intent.addCategory("thisCategory");
startActivity(intent);
```

Android manifest file:

```xml
<activity android:name="ActivityB">
    <intent-filter>
        <action android:name="thisAction"/>
        <category android:name="thisCategory"/>
    </intent-filter>
</activity>
```

Android system will locate a best matched component such as ActivityB based on action and category requirement. Implicit intent also leaves a security vulnerability hole.
Reference

https://sites.google.com/site/mobilesecuritylabware/home/android-components/activity-intent