Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

Mobile Application Development

By,
Prof. Himanshu H Patel,
Prof. Hiten M Sadani

U. V. Patel College of Engineering, Ganpat University

Android Manifest

- Android projects use a special **configuration file** called the Android manifest file
- The Android application manifest file is a specially formatted **XML** file that must accompany each Android application.
- This file contains important information about the application's **identity**.

Android Manifest

- Every app project must have an **AndroidManifest.xml** file (with precisely that name) at the root of the project source set.
- The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.
- The **components of the app**, which include all activities, services, broadcast receivers, and content providers.
- Each component must define basic **properties** such as the name of its Kotlin or Java class.
- It can also declare capabilities such as which **device configurations** it can handle, and intent filters that describe how the component can be started.

Android Manifest

- The **permissions** that the app needs in order to access protected parts of the system or other apps. It also declares any permissions that other apps must have if they want to access content from this app.
- The **hardware and software features** the app requires, which affects which devices can install the app from Google Play.
- If you're using Android Studio to build your app, the manifest file is created for you, and most of the essential manifest elements are added as you build your app (especially when using code templates).

App components

- For each app component that you create in your app, you must declare a corresponding XML element in the manifest file:
  - <activity> for each subclass of Activity.
  - <service> for each subclass of Service.
  - <receiver> for each subclass of BroadcastReceiver.
  - <provider> for each subclass of ContentProvider.

App components

- If you subclass any of these components without declaring it in the manifest file, the system cannot start it.

- The name of your subclass must be specified with the name attribute, using the full package designation. For example, an Activity subclass can be declared as follows:

```
<manifest ... >
    <application ... >
        <activity android:name="com.example.myapp.MainActivity" ... >
        </activity>
    </application>
</manifest>
```

App components

- Intent filters
  - App activities, services, and broadcast receivers are activated by intents. An intent is a message defined by an Intent object that describes an action to perform, including the data to be acted upon, the category of component that should perform the action, and other instructions.
  - When an app issues an intent to the system, the system locates an app component that can handle the intent based on intent filter declarations in each app's manifest file.
- Icons and Labels
  - The icon and label that are set in the **<application>** element are the default icon and label for each of the app's components (such as all activities).

App components

- Permissions
- Android apps must request permission to access sensitive user data (such as contacts and SMS) or certain system features (such as the camera and internet access). Each permission is identified by a unique label.

```
<manifest ... >
    <uses-permission android:name="android.permission.SEND_SMS"/>
    ...
</manifest>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1"
    android:versionName="1.0">

    <!-- Beware that these values are overridden by the build.gradle file -->
    <uses-sdk android:minSdkVersion="15" android:targetSdkVersion="26" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <!-- This name is resolved to com.example.myapp.MainActivity
             based upon the namespace property in the `build.gradle` file -->
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name=".DisplayMessageActivity"
            android:parentActivityName=".MainActivity" />
    </application>
</manifest>
```

Example of Manifest file