Multiple Values of Android

3.3.1 Multiple Values

If more than one value can be specified, the element is almost always repeated, rather than listing multiple values within a single element.

3.3.2 Resource Values

Some attributes have values that can be displayed to users, for example, a label and an icon for an activity. The values of these attributes should be localized and therefore set from a resource or theme.

The package name can be omitted if the resource is in the same package as the application, type is a type of resource, such as "string" or "drawable," and name is the name that identifies the specific resource.

3.3.3 Sting Values

Where an attribute value is a string, double backslashes ('\\') must be used to escape characters, for example, '\\n' for a newline or '\\uxxxx' for a Unicode character.

MainActivity.java × AndroidManifest.xml ×

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.csis.pace.edu.mypace" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="My Pace "
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="My Pace " >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Figure 3.3 Default AndroidManifest.xml

Fig. 3.3 is the default AndroidManifest.xml file generated by IDE after a blank Android application. In the third creating line, "com.example.csis.pace.edu.mypace," is the package name of the project, and it exactly as the same as the first line of *MainActivity.java*. Many elements inside the <application> and </application> correspond to Java objects, including activities (<activity>), services (<service>), broadcast receivers (<receiver>), and content providers (<provider>). In our project, we only create an activity, thus, there is only one <activity> in *AnroidManifest.xml* file. In this <activity>, the android:name=".MainActivity" shows the name of the activity.

The <intent-filter> specifies the type of intents that an activity, service, or broadcast receiver can respond to. An intent filter declares the capabilities of its parent component, what an activity or service can do and what types of broadcasts a receiver can handle. It opens the component to receiving intents of the advertised type, while filtering out those that are not meaningful for the component.

When adding an action to an intent filter. An <intent-filter> element must contain one or more <action> elements. If it doesn't contain any, no Intent objects will get through the filter. Some standard actions are defined in the Intent class as ACTION_string con- stants. To assign one of these actions to this attribute, prepend "an- droid.intent.action." to the string that follows ACTION_. In our project, use "android.intent.action.MAIN" for ACTION_MAIN.

The <category> is used to add a category name to an in-

tent filter. Standard categories are defined in the Intent class as CATEGORY_name constants. The name assigned here can be derived from those constants by prefixing

“android.intent.category.” to the name that follows CATEGORY_. In our project, the string value is “an- droid.intent.category.LAUNCHER” for CATEGORY_LAUNCHER.