Android Development

Ch-01

Preview

General information

- تعودُ بدايات نظام الآندرويد إلى عام 2003, وذلكَ قبلَ ظهور الهواتف الذكيّة وانتشارها أصلًا، إذ أنّ النظام كان في بداياته موجّهًا للكاميرات الرقميّة كنظام تشغيل يسهّل التعامل مع الكاميرات وإدارة الصور وملفّات الفيديو؛ ولكن حدث تغيير في الخطط ورؤية المشروع إذ أنّ سوق الكاميرات الرقميّة صغير مقارنةً بسوق الهواتف المحمولة ولذا تمّ التحويل لدعم الهواتف الذكية.
- وفي عام 2005 قامت شركة جوجل بالاستحواذ على شركة آندرويد, وعمِلَت على متابعة تطويره بالإضافة لإدخال تطبيقاتها على النظام بشكل متكامل كخرائط جوجل، يوتيوب، وغيرها. إذ تمّ إطلاق أوّل نسخة تجريبيّة للمطوّرين في 11/ 2007.

General information etc.

- **Android**:
- ➢ Is a mobile operating system based on a modified version of the Linux kernel.
- **Java Development Kit (JDK):**
- ➢ is a package of tools for developing Java-based software.
- **System Development Kit (SDK):**
- ➢ developed by Google for the Android platform. It is a collection of software development tools and libraries required to develop Android applications -development tools-. Like: emulator, documentations, debugging.
- **Android Virtual Device (AVD):**
- ➢ defines the characteristics of an emulator device.
- **Integrated Development Environment (IDE):**
- ➢ consists of a source code editor, build automation tools and a debugger, compiler and interpreter .
- **Extensible Markup Language (XML).**
- ➢ a data markup/descriptive language.

General information etc.

- **Android Native development.**
- **Cross-platform development.**
- **System apps & User apps.**
- **Bugs:**

➢ something gives you unexpected results.

- **Causes of bugs:**

1. **Human errors**: like syntax errors.
2. **Software errors**: like run apps on not suitable versions of android.
3. **Hardware errors**: like no space on device to run app.

- **Debugging:**

➢ something where you fix those bugs.

- **Logcat tool:**

➢ gives you the feedback while the application running about what's happening in the device/emulator. One of famous Android studio debugging tools.

General information etc.

- **Log Types:**

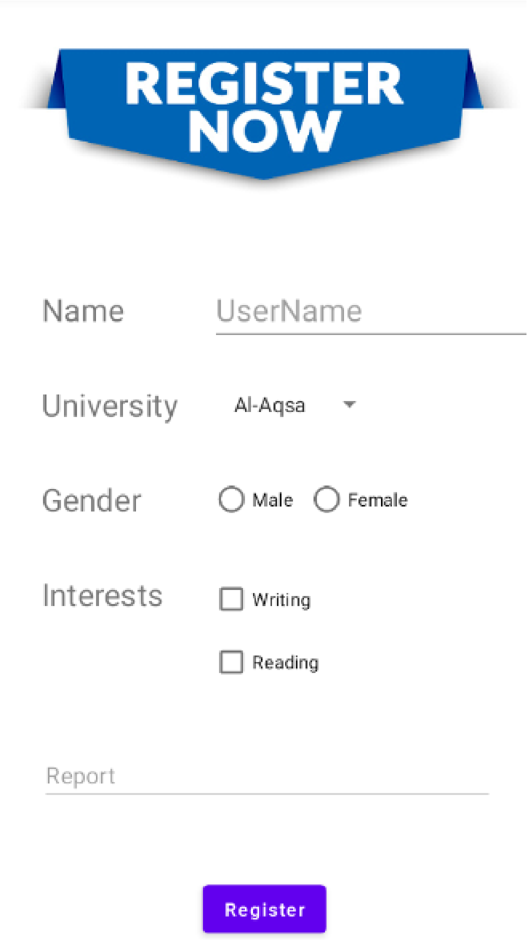
1. **Debug:** Show debug log messages that are useful during development.
2. **Warn:** Show possible issues that are not yet errors.
3. **Error:** Show issues that have caused errors.
4. **Info:** Show expected log messages for regular usage.
5. **Verbose:** Show all log messages (the default).

- **Android studio Keyboard Shortcuts:**

- **Ctrl + P: Show methods arguments**
- **Ctrl + Space: Show expected result options**
- **Alt + Enter: Import Library**
- **Ctrl + O: to override/implement methods.**
- **Ctrl + Alt + L: for Code rearrangement**
- **Ctrl + shift + '-' : to minimize the functions of a class.**
- **Ctrl + shift + '+': to maximize the functions of a class.**

Design the following Activity

- **Activity Views:**
- **ImageView**
- **TextView**
- **EditText**
- **Spinner**
- **RadioGroup**
- **RadioButton**
- **CheckBox**
- **Button**



Name UserName

University Al-Aqsa ▾

Gender ○ Male ○ Female

Interests □ Writing

□ Reading

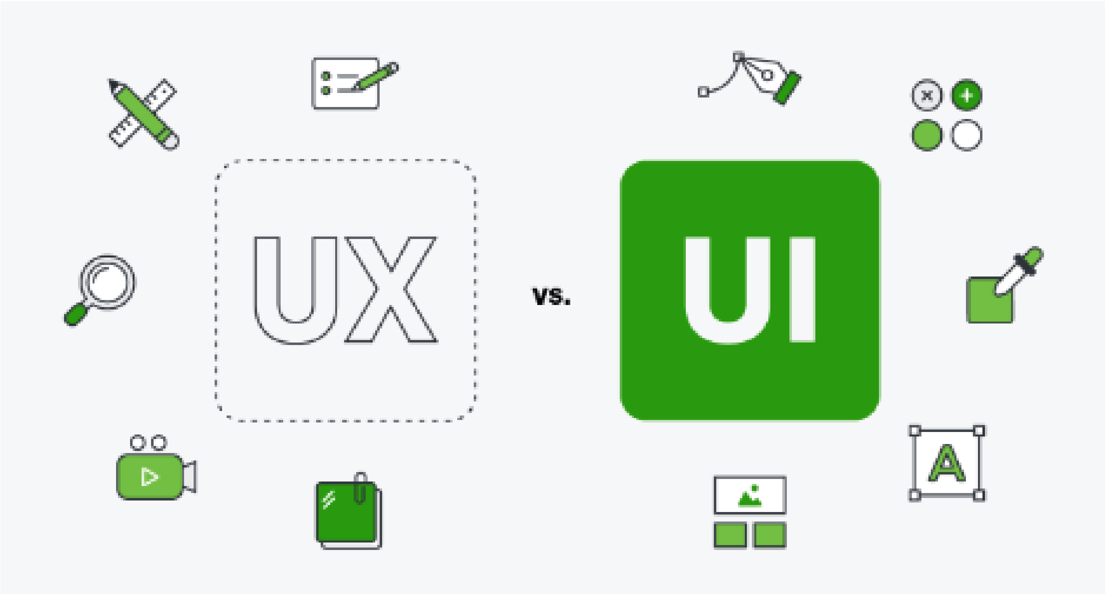
Report

Android App. Components

- The basic **core application components** that can be used in Android application:
  - Activities
  - Intents
  - Broadcast Receivers
  - Services
  - Content Providers
- All the **application components** are defined in the android app description file **(AndroidMainfest.xml)**

Activities

- An activity represents **a single screen** with **a user interface (UI).**
- activities in the app are **independent of each other** but will **work together to provide a better User Experience (UX)**.



Activities etc.

- An Android application can have **zero or more than one activities.**
- The main purpose of an activity is to **interact with the user.**
- **Every activity** must be **declared** in your **AndroidManifest.xml file.**
- Every activity has its own **lifecycle**.
- **Life Cycle:** From the moment an activity appears on the screen to the moment it is hidden, it goes through **a number of stages**.

Intents

- In android, Intent is **a messaging object** which is used to **request an action from another component**.
- In android, intents are mainly used to **perform** the following:
- Starting an **Activity**
- Starting a **Service**
- Delivering a **Broadcast**
- There are **two types of intents** available in android, those are:
- Explicit Intents
- Implicit Intents

Services

- Service is a component that keeps an **app running in the background - don't have any user interface -** to perform **long running operations.**
- Ex: play music in background when the user in different app.

Broadcast Receivers

- Broadcast Receiver is a component that will **allow a system to deliver events to the app.**
- Ex: sending **a low battery message** to the app.

Content Providers

- Content Providers are useful to **exchange the data between the apps** based on the requests.
- It can **share the app data** that stores in the **file system**, **SQLite database** or **any other storage location** that our app can access.

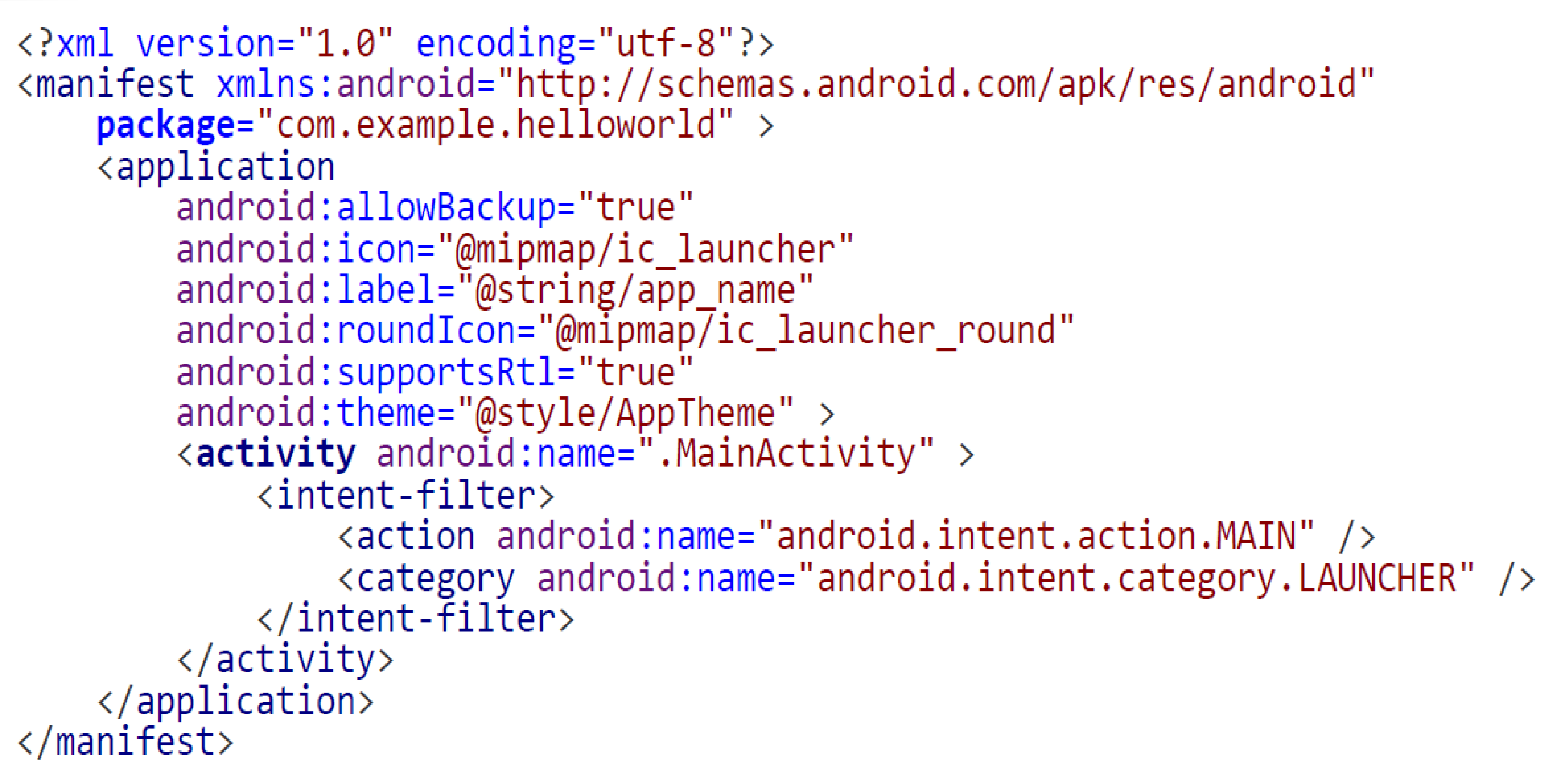
Additional Components

- **Fragments**: These are used **to represent the portion of User Interface (UI)** in an activity.
- **Layouts:** These are used to define the **User Interface (UI)** for an activity or app.
- **Views**: These are used **to build a user interface for an app** using UI elements like buttons, lists, etc.
- **Resources:** To build an android app we required **external elements** like images, audio files, strings, dimens, etc.
- **Manifest File.**

AndroidManifest.xml

- It’s a **configuration** file for the application.
- Every application must have an **AndroidManifest.xml** file.
- It presents **essential information about the application** to the Android system.
- It includes:
- ➢ App’s java package name: serves as **a unique identifier.**
- ➢ App’s Components.
- ➢ Permissions: GPS, Internet, … .
- ➢ Hardware and software features: SDK version to **filter** your app from devices that do not meet its platform version requirements.
- The manifest file also specifies the application metadata: icon, themes, etc.

AndroidManifest.xml snippet



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.helloworld" >
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

application: This element declares the application component such as activity etc.
intent-filter is the element of activity that describes its behavior.
action for activity’s action. The intent-filter must have at least one action element.

Activity life cycle

- **onCreate():** This is the first method and it **fires** when the system **creates an activity** for the **first time**.
  Use the onCreate() method to **create the objects** that you will be using in your application.

- **onStart():** called when the activity becomes **visible to the user**.

- **onResume():** called when the activity **starts interacting with the user.**
  The app will stay in this **Resumed state** until an another activity takes a focus away from the app like getting a phone call or screen turned off, etc.

Activity life cycle etc.

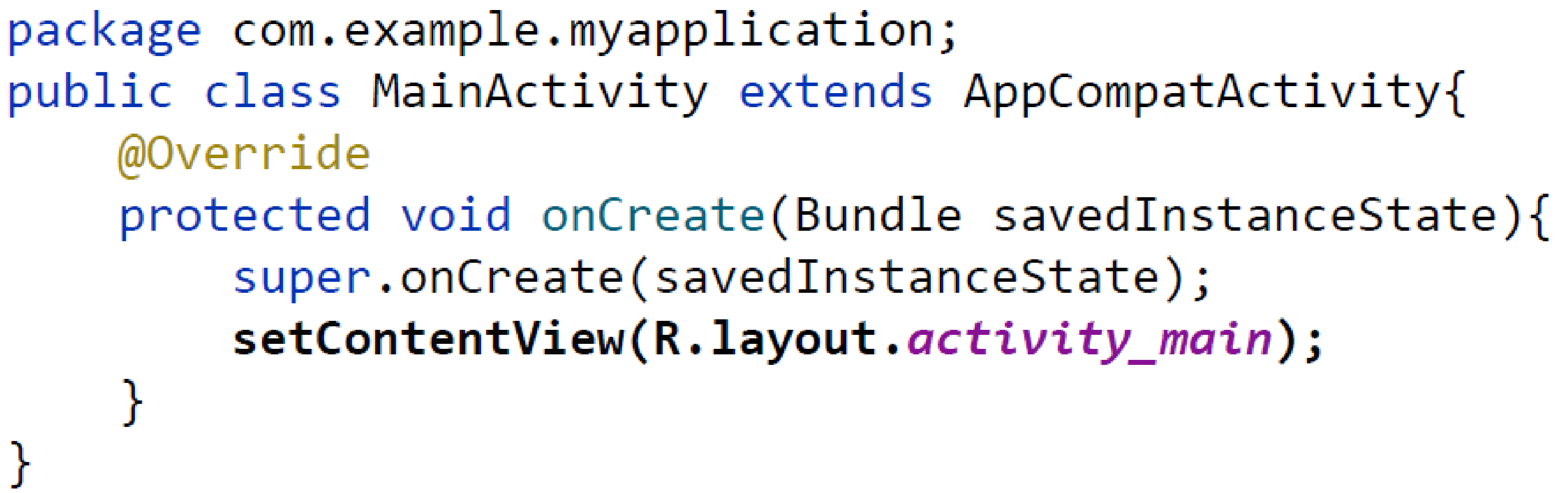
- **onPause():** In case if the user **leaves an activity or** any interruption events happen in **Resumed state**, the activity will enter into **Paused state** and the system will invoke **onPause()** method.

- **onStop():** called when the **activity is no longer visible** to the user. The **onStop()** method is useful to **release all the app resources** which are no longer needed to the user.

- **onRestart():** called when the **activity has been stopped** and is **restarting again.** Comes before **onStart()** instad of **onCreate().**

- **onDestroy():** The system will invoke this **onDestory() method** either the **activity is finishing or system destroying the activity** to save space.

Activity java.class snippet



```java
package com.example.myapplication;
public class MainActivity extends AppCompatActivity{
    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Workable Slide

Done

Fares Saleem