Mobile Computing

Module 1

1.3 Android Application

MOAATH ALRAJAB

Learning Unit 1

- 1.1 Introduction
  - Android History
  - Android Architecture
  - Getting Started
- 1.2 Android Studio
  - IDE parts
  - Creating a project
  - Installing required plugins
- 1.3 Android Application
  - Concept and components
  - Manifest file

Goal

Understand applications and their components

Concepts:

- activity,
- service,
- broadcast receiver,
- content provider,
- intent,
- AndroidManifest

Applications

Written in Java (it’s possible to write native code – will not cover that here)

Good separation (and corresponding security) from other applications:

- Each application runs in its own process
- Each process has its own separate VM
- Each application is assigned a unique Linux user ID – by default files of that application are only visible to that application (can be explicitly exported)

Application Components

Activities – visual user interface focused on a single thing a user can do

Services – no visual interface – they run in the background

Broadcast Receivers – receive and react to broadcast announcements

Content Providers – allow data exchange between applications

Activities

Basic component of most applications

Most applications have several activities that start each other as needed

Each is implemented as a subclass of the base Activity class

Activities – The View

Each activity has a default window to draw in (although it may prompt for dialogs or notifications)

The content of the window is a view or a group of views (derived from View or ViewGroup)

Example of views: buttons, text fields, scroll bars, menu items, check boxes, etc.

View(Group) made visible via Activity.setContentView() method.

Services

Does not have a visual interface

Runs in the background indefinitely

Examples

- Network Downloads
- Playing Music
- TCP/UDP Server

You can bind to a an existing service and control its operation

Broadcast Receivers

Receive and react to broadcast announcements

Extend the class BroadcastReceiver

Examples of broadcasts:

- Low battery, power connected, shutdown, timezone changed, etc.
- Other applications can initiate broadcasts

Content Providers

Makes some of the application data available to other applications

It’s the only way to transfer data between applications in Android (no shared files, shared memory, pipes, etc.)

Extends the class ContentProvider;

Other applications use a ContentResolver object to access the data provided via a ContentProvider

Intents

An intent is an Intent object with a message content.

Activities, services and broadcast receivers are started by intents. ContentProviders are started by ContentResolvers:

- An activity is started by Context.startActivity(Intent intent) or Activity.startActivityForResult(Intent intent, int RequestCode)
- A service is started by Context.startService(Intent service)
- An application can initiate a broadcast by using an Intent in any of Context.sendBroadcast(Intent intent), Context.sendOrderedBroadcast(), and Context.sendStickyBroadcast()

Shutting down components

Activities

- Can terminate itself via finish();
- Can terminate other activities it started via finishActivity();

Services

- Can terminate via stopSelf(); or Context.stopService();

Content Providers

- Are only active when responding to ContentResolvers

Broadcast Receivers

- Are only active when responding to broadcasts

Android Manifest

Its main purpose in life is to declare the components to the system:

```
 <?xml version="1.0" encoding="utf-8"?>
<manifest . . . >
   <application . . . >
      <activity android:name="com.example.project.FreneticActivity"
             android:icon="@drawable/small_pic.png"
             android:label="@string/freneticLabel"
             . . .  >
      </activity>
      . . .
   </application>
</manifest>
```

Intent Filters

Declare Intents handled by the current application (in the AndroidManifest):

```
 <?xml version="1.0" encoding="utf-8"?>
<manifest . . . >
   <application . . . >
      <activity android:name="com.example.project.FreneticActivity"
              android:icon="@drawable/small_pic.png"
              android:label="@string/freneticLabel"
              . . .  >
         <intent-filter . . . >
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
         <intent-filter . . . >
            <action android:name="com.example.project.BOUNCE" />
            <data android:mimeType="image/jpeg" />
            <category android:name="android.intent.category.DEFAULT" />
         </intent-filter>
      </activity>
      . . .
   </application>
</manifest>
```

Shows in the Launcher and is the main activity to start

Handles JPEG images in some way