Android Applications

Session 2

1

Objective

- Application fundamentals
- Application components
- The Manifest file
- Application resources
- The Application lifecycle
- TO-DO List example
- Types of Android applications

2

Application Fundamentals

- Các thành phần (components) cơ bản tạo nên một ứng dụng Android bao gồm:
  - Activities
  - Services
  - Content providers
  - Broadcast receivers
- Activity, service, broadcast receiver components được kích hoạt (activated) bởi một thông báo không đồng bộ gọi là intent
- Manifest file (.xml) khai báo các components trong ứng dụng
- Resources chứa các tài nguyên tĩnh của ứng dụng (ảnh, âm thanh)

3

Activities

- Một Activity đại diện cho một màn hình với giao diện người dùng, ví dụ *quay số điện thoại, chụp ảnh, gửi email, bản đồ*
- Mỗi ứng dụng có một *"main activity"*, activity này được gọi khi ứng dụng khởi động, hiển thị màn hình giao diện của ứng dụng cho phép người dùng tương tác
- Một ứng dụng có thể có một hoặc nhiều activity
- Một Activities được thực hiện như một lớp con của lớp Activity

```
public class MainActivity extends Activity{
...}
```

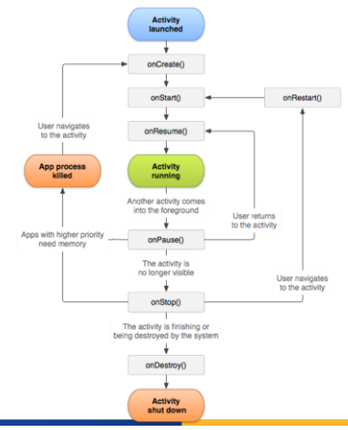
4

Activities

- Mỗi activity có thể gọi một activity khác để thực hiện các hoạt động khác nhau.
- Mỗi khi có activity mới khởi động, activity trước đó được dừng lại, nhưng hệ thống xếp activity đó vào stack ("back stack")
- Khi activity mới khởi động, nó được xếp vào back stack và tương tác với user.
- Back stack hoạt động theo cơ chế "last in, first out"
- Khi user bấm phím BACK, activity sẽ được lấy ra từ stack (destroyed) và activity trước đó sẽ được khôi phục (hoạt động)

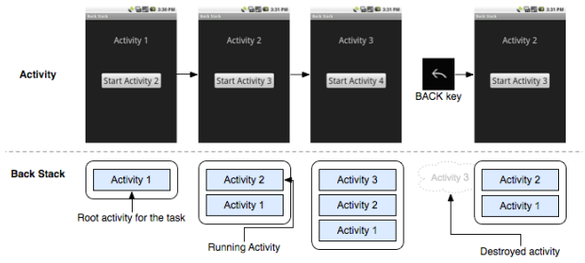
5

Activity Life Cycle



6

Tasks and Back Stack



Services

- Là thành phần chạy ẩn của Android. Sử dụng để update dữ liệu, đưa ra các cảnh báo (notification). Ví dụ: một service có thể chơi nhạc trong khi user đang ở trong ứng dụng khác.
- Không cung cấp giao diện
- Hai kiểu service:
  - Started:
    - Bắt đầu bằng gọi hàm startService(), thực hiện một hành động đơn lẻ và không trả kết quả cho đối tượng gọi
    - Ví dụ: thực hiện tải file, khi kết thúc tự động dừng lại
  - Bound:
    - Ràng buộc với các components, gọi hàm binService(), hoạt động kiểu client-server
    - Hủy khi các components không còn ràng buộc

Content Providers

- Cung cấp nội dung dữ liệu từ một ứng dụng khác theo yêu cầu
- Dữ liệu có thể được lưu trữ trong hệ thống file, SQLite database, hoặc trên web, ...
  - Content provider cho phép các ứng dụng có thể truy vấn dữ liệu. Ví dụ: audio, video, images, thông tin contact cá nhân...
- Content providers cũng có thể dùng để đọc ghi dữ liệu cục bộ và không chia xẻ

```
public class MyContentProvider extends ContentProvider{
...}
```


Content Providers

- Có hai cách để public data:
  - Tạo một content provider
  - Thêm dữ liệu vào một provider đã có
- Mỗi content provider đưa ra một URI với định danh duy nhất cho dữ liệu.
- Ví dụ nội dung URI:

```
content://com.example.transportationprovider/trains/1122
```

A B C D

Broadcast Receivers

- Một *broadcast receiver* là một component có thể phản hồi các thông báo (announcements) phát ra từ các ứng dụng khác hoặc từ hệ thống.
- Ví dụ các thông báo:
  - Tắt màn hình
  - Pin yếu
  - Chụp ảnh,…
- Broadcast receivers không hiển thị giao diện người dùng
- Broadcast receivers có thể tạo thông báo dạng *status bar notification* để cảnh báo người dùng khi có sự kiện xảy ra

Activating Components

- Ba loại thành phần (activities, services, and broadcast receivers) được kích hoạt bởi một thông báo không đồng bộ gọi là *intent*
- Intent được sử dụng để truyền các thông báo nhằm khởi tạo một Activity hoặc service để thực hiện công việc nào đó
- Các thông báo không chờm nhau:
  - Broadcast intents chỉ được chuyển đến broadcast receivers, không chuyển đến activities or services.
  - Một intent tới startActivity() chỉ chuyển tới activity, không chuyển tới service hoặc broadcast receiver.

Activating Components

- Activity: được kích hoạt bởi Context.startActivity() hoặc Activity.startActivityForResult()
- Service: được khởi động hoặc chuyển chỉ thị tới service khác thông qua hàm Context.startService(). Tương tự, một intent có thể gọi Context.bindService() để thiết lập kết nối giữa các component và service
- Các đối tượng intent gọi tới tới bất kỳ broadcast methods nào (Context.sendBroadcast(), Context.sendOrderedBroadcast(), Context.sendStickyBroadcast()) đều được chuyển tới broadcast receivers

The Manifest File

- Là file xml chứa các thành phần của ứng dụng:
  - Liệt kê các permission cho phép dùng các phần cứng hoặc ứng dụng, ví dụ: camera, bluetooth service, multitouch screen, truy cập internet access hay đọc danh bạ.
  - Khai báo mức API bé nhất yêu cầu bởi ứng dụng.
  - Các thư viện API mà ứng dụng cần liên kết, ví dụ Google Maps library
  - …

Structure of the Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
  <manifest>
      <uses-permission />
      <permission />
       …
      <application>
          <activity>
              <intent-filter>
                  <action />
                  <category />
                  <data />
              </intent-filter>
              <meta-data />
          </activity>
          <activity-alias>
              <intent-filter> . . . </intent-filter>
              <meta-data />
          </activity-alias>
          <service>
              <intent-filter> . . . </intent-filter>
              <meta-data/>
          </service>
              …
      </application>
  </manifest>
</xml>
```


Setting up Development Environment

1. Preparing development computer
2. Adding SDK Packages

Declaring Components

- Nhiệm vụ chính của manifest là báo cho hệ thống biết về các thành phần của ứng dụng.
- Ví dụ, một file manifest có thể khai báo một activity như sau:

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


Declaring Application Requirements

- Người lập trình có thể sử dụng file manifest để khai báo yêu cầu thiết bị hoặc phần mềm
- Khi ứng dụng được đưa lên Android Market, Market sử dụng các khai báo này để lọc những ứng dụng nào phù hợp với thiết bị.
- Một số đặc tính của thiết bị cần chú ý khi khai báo:
  - Screen size + density
  - Input configurations
  - Device features
  - Platform Version

Application Resources

- Một ứng dụng android bao gồm mã lệnh và tài nguyên
- Resources được lưu trữ tách biệt khỏi mã nguồn, gồm ảnh, audio...
- Tài nguyên đặt trong thư mục res/ , mỗi loại đặt trong các thư mục con khác nhau.
- Có hai kiểu resources:
  - Default resources
  - Alternative resources

Application Resources

Two different devices, each using the default layout (the app provides no alternative layouts).

Two different devices, each using a different layout provided for different screen sizes.

Application Resources

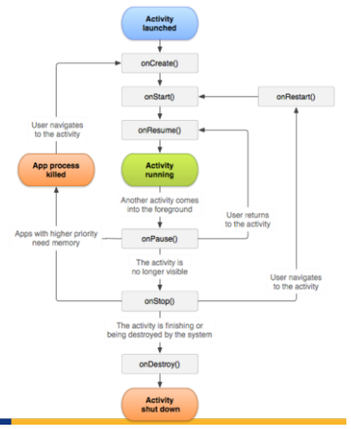
- Android quản lý các process theo chế độ ưu tiên
  - Các process có chế độ ưu tiên thấp sẽ bị giải phóng mà không cần cảnh báo
- Thời gian sống của tiến trình không được điều khiển trực tiếp bởi chính nó. Hệ thống sẽ xác định thời gian sống thông qua các kết hợp:
  - Những phần của ứng dụng mà hệ thống biết đang chạy.
  - Những phần đó quan trọng như thế nào đối với người dùng
  - Bao nhiêu vùng nhớ chiếm lĩnh trong hệ thống.

The Application Lifecycle

- Hầu hết trình quản lý thời gian sống được thực hiện tự động bởi hệ thống qua activity stack.
- Lớp activity có các phương thức giúp quản lý app:
  - onCreate()
  - onStart()
  - onResume()
  - onPause()
  - onStop()
  - onRestart()
  - onDestroy()

Life Cycle States

- Một activity có các state sau:
  - active or running: foreground of the screen
  - Paused: if it has lost focus but is still visible to the user.
  - Stopped: if it is completely obscured by another activity.



Life Cycle Methods

- onStart()
  - Gọi khi lớp activity được khởi tạo, dùng để thiết lập giao diện
- onRestart()
  - được gọi khi ứng dụng chuyển sang onStop(), nhưng muốn khởi động lại bằng onStart().

Life Cycle Methods

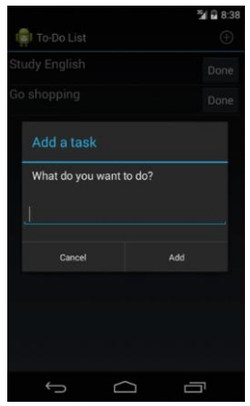
- onPause()
  - hàm được gọi khi hệ thống đang focus đến 1 activity trước đó
- onResume()
  - gọi ngay sau khi onStart() được gọi. Ở phương thức này chúng ta có thể tương tác được với UI

Life Cycle Methods

- onStop()
  - hàm được gọi khi một activity khác được khởi động và focus.
- onDestroy()
  - gọi khi nhấn back từ activity, hoặc call method finish() của activity.

TO-DO List Example

1. Create a new project
2. Design a user interface
3. Override the onCreate() method
4. Run or debug the application



Summary

- In this session, we learnt:
  - Application fundamentals
  - Application components
  - The Manifest file
  - Application resources
  - The Application lifecycle
  - TO-DO List example
  - Types of Android Applications