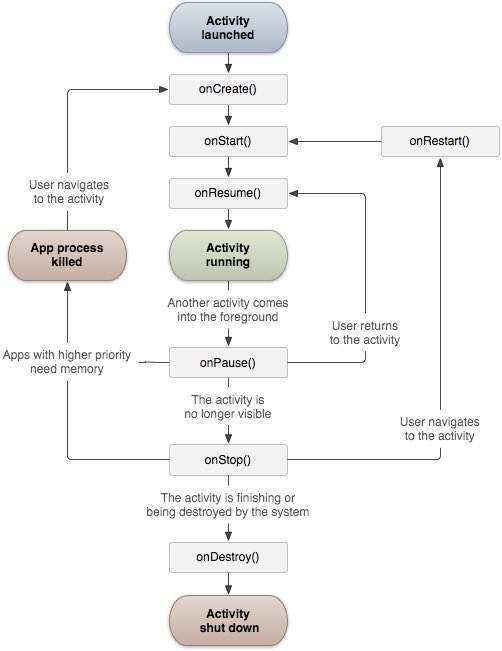
Nguyễn Ngọc Trung – Fresher Android GEM tháng 5

Báo cáo buổi 1

1. Android Basic
2. Activity

Vòng đời của một ứng dụng Android



Giải thích

* onCreate

+ Được gọi khi activity được tạo

+ Đây là nơi ta nên thực hiện khởi tạo tĩnh thông thường, view, nối dữ liệu với các danh sách

+ onStart được gọi sau đó

* onStart

+ Hàm này được gọi khi lớp ứng dụng xuất hiện trên màn hình

* onResume():

+ Hàm được gọi ngay sau OnStart hoặc khi người dùng focus ứng dụng, hàm này sẽ đưa ứng dụng lên top màn hình-

- onPause():

+ Hàm được gọi khi hệ thống đang focus đến 1 activity trước đó.  
- onStop():

+ Hàm được gọi khi một activity khác được khởi động và focus.  
- onRestart():

+ Được gọi khi ứng dụng chuyển sang onStop(), nhưng muốn khởi động lại bằng onStart().

1. Service

Service là một trong những đối tượng chính của một ứng dụng android. Nó là một thành phần luôn chạy dưới lớp background của hệ thống mà không cần sự tương tác của người dùng.  
Một ví dụ đơn giản nhất có thể hình dung được về service là ta mở một bài nhạc trong ứng dụng nghe nhạc lên. Khi ta thoát ứng dụng ra bài nhạc đó vẫn chạy, chỉ khi nào ta vào lại ứng dụng đó để dừng bài hát lại thì nhạc mới dừng.

1. Broadcast Receivers
2. Chức năng

* Nhận và phản hồi thông tin
* xử lý và phát các thông điệp từ các ứng dụng khác hoặc từ hệ thống

Ví dụ: các ứng dụng có thể bắt đầu phát thông điệp đến các ứng dụng khác để cho biết rằng một số dữ liệu đã được tải thành công xuống thiết bị và sẵn sàng cho việc sử dụng

* **Broadcast Receivers** sẽ đảm nhận việc thông báo vào đưa ra những hành động thích hợp.

Có hai bước quan trọng để cho **Broadcast Receivers** làm việc:

1. Tạo Broadcast Receiver.

2. Đăng ký Broadcast Receiver.

1. Content providers

* **Content providers** là thành phần cung cấp dữ liệu từ một ứng dụng đến một ứng dụng khác dựa trên các **Request**. Mỗi **Request** được xử lý bằng các phương thức của class **ContentResolver**.
* Mỗi ứng dụng Android chạy trong các tiến trình riêng của chính mình và nó có các điều khoản riêng của nó, điều mà giữ dữ liệu của ứng dụng ẩn với các ứng dụng khác. Tuy nhiên, thỉnh thoảng nó được yêu cầu chia sẻ dữ liệu đến các ứng dụng khác. Sử dụng **Content Provider** trong việc này rất hữu ích.

1. Fragments

**Fragment** là một phần giao diện người dùng hoặc hành vi của một ứng dụng. **Fragment** có thể được đặt trong **Activity**

Để thiết kế một giao diện, bạn có thể thiết kế nhiều mảnh (fragment) và ghép lại với nhau.

1. Intentss/Filters

* Intent: Dùng để chuyển, truyền dữ liệu giữa các activity

II. User Interface

1. UI Layout
2. Linear Layout

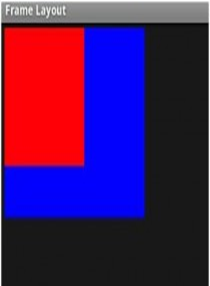
Layout này cho phép sắp xếp các control theo 2 hướng trên giao diện: Hướng từ trái qua phải và hướng từ trên xuống dưới.

1. Relative Layout

* RelativeLayout cho phép sắp xếp các control theo vị trí tương đối giữa các control khác trên giao diện (kể cả control chứa nó). Thường nó dựa vào Id của các control khác để sắp xếp theo vị trí tương đối
* Khi làm RelativeLayout phải chú ý là đặt Id control cho chuẩn xác, nếu sau khi Layout xong mà bạn lại đổi Id của các control thì giao diện sẽ bị xáo trộn

1. FrameLayout

Là loại Layout cơ bản nhất, đặc điểm của nó là khi gắn các control lên giao diện thì các control này sẽ luôn được “Neo” ở góc trái trên cùng màn hình, nó không cho phép chúng ta thay đổi vị trí của các control theo một Location nào đó.



1. List View

* Để hiển thị dữ liệu theo danh sách



* Android cung cấp giao diện sẵn để ta sử dụng ListView nhưng thường ta sửa lại cho đúng theo ý muốn, gọi là Custom ListView

1. RecyclerView – nâng cao của ListView

* RecyclerView là một View mới trong android giống như ListView nhưng mạnh mẽ hơn rất nhiều.  
  - RecyclerView cho phép chúng ta

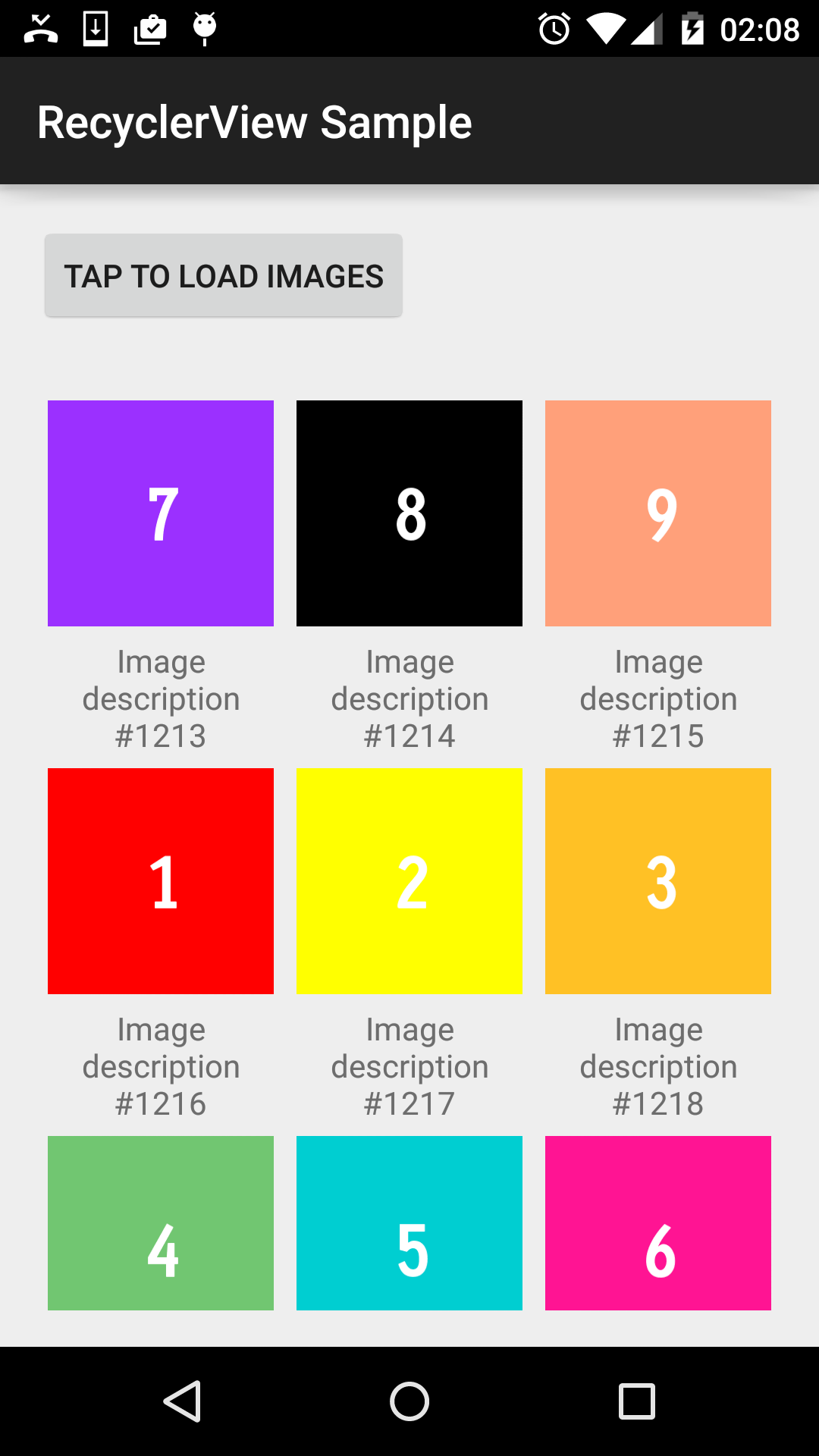
+ Load dữ liệu nhiều hơn ListView,

+ Vuốt mượt hơn

+ Hiệu ứng đẹp hơn

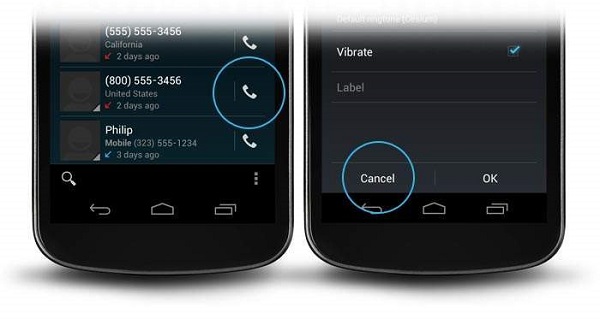
+ Hỗ trợ đa dạng layout(ngang, dọc, khung lưới) của các phần tử trong danh sách.

Ví dụ



1. UI Controls
2. Button

* Nhấn vào để thực hiện hành động nào đó



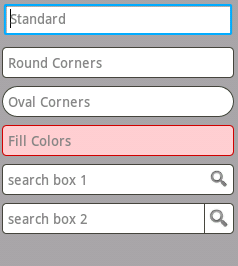
1. TextView

* Để hiển thị thông tin, nội dung



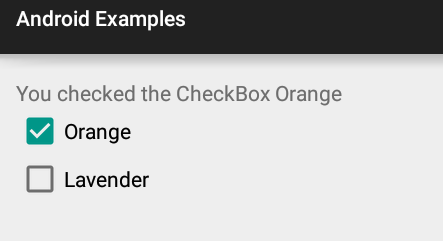
1. EditText

* Để nhập thông tin



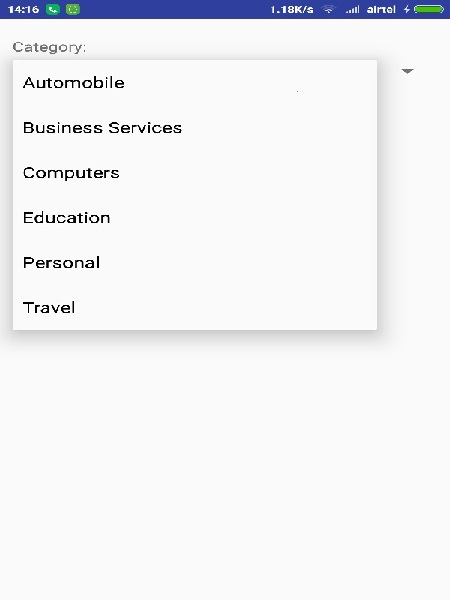
1. CheckBox

* Dạng ô để ta thực hiện lựa chọn



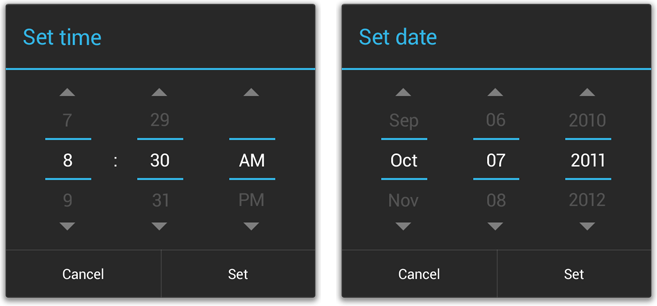
1. Spinner

* Dạng danh sách để ta lựa chọn



1. Timepicker android

* Giúp ta lựa chọn thời gian



Ngoài ta còn nhiều UI Control khác như radio button, process bar, image button, …

1. Event Handling

Dùng để xử lý các sự kiện khi tương tác với màn hình ,như click, long click, touch, drag, zoom,…

Các sự kiên thường dùng

* OnClickListener: xử lý khi người dùng click
* OnLongClickListener: xử lý khi người dùng giữ trong một khoảng thời gian và nhả ra
* onTouchListener: xử lý ngay khi người dùng chạm vào màn hinh
* Styles and Themes: Chỉnh giao diện của ứng dụng custom giao diện

1. Styles và Themes
2. Styles

* Một Style có thể được áp dụng cho một View trong một file layout hoặc trong toàn bộ  Activity hoặc ứng dụng

1. Themes

* Theme là một Style trong Android được áp dụng cho toàn bộ Activity hoặc ứng dụng thay vì được áp dụng cho một View nào đó.

III. MVP

- Model tổ chức business logic của ứng dụng. Nó kiểm soát như thế nào dữ liệu có thể được tạo ra, lưu trữ, và sửa đổi.

- View là một giao diện thụ động hiển thị dữ liệu và các tuyến đường hành động dùng cho Presenter.

- Presenter đóng vai trò trung gian, nó lấy dữ liệu từ Model và cho thấy nó trong các View. Nó cũng xử lý hành động người dùng mong muốn được nó bằng các View.

