Họ tên: Nguyễn Văn Khỏe.

Thực tập khóa GEM Android Fresher\_Tháng 5.

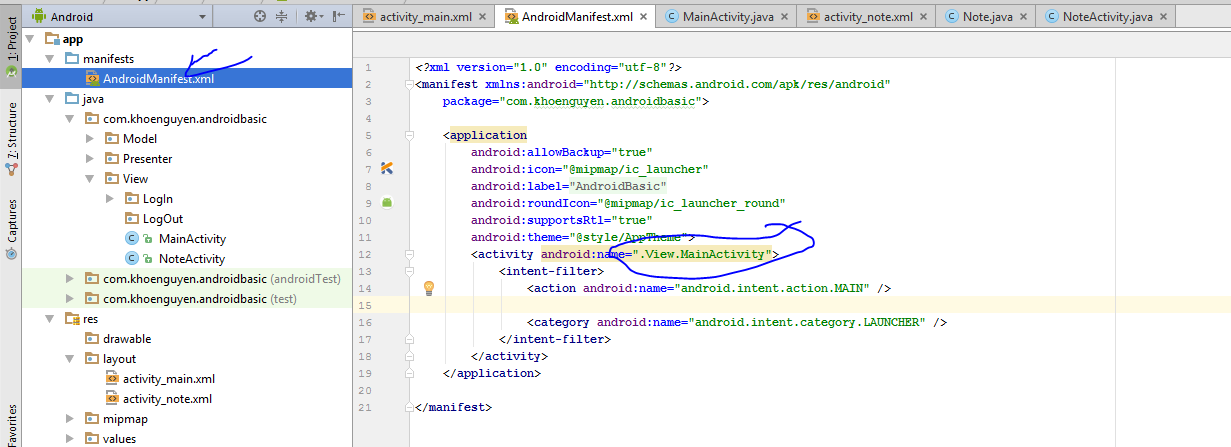
**Báo Cáo**

**Kiến thức Android Basic**

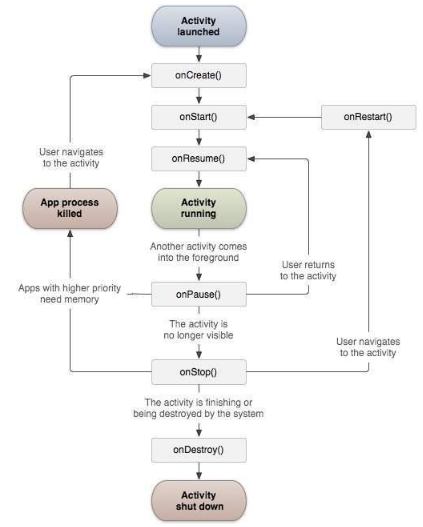
**I. Android Basic**

1. Activities

Mỗi ứng dụng sẽ có một hoặc nhiều Activity. Mỗi một activity có một vòng đời riêng độc lập hoàn toàn với các activity khác. Để được gọi đến thì bắt buộc activity đó cần phải được khai báo trong manifest, ví dụ: trong MainAcitivy được khởi tạo trong file AndroidManifest.xml:



Vòng đời của một activity được thể hiện qua biểu đồ sau:



* Các hàm thực thi:

+ onCreate(): được gọi khi một activity được khởi tạo.

+ onStart(): được gọi khi layout tương ứng của activity được hiện lên màn hình.

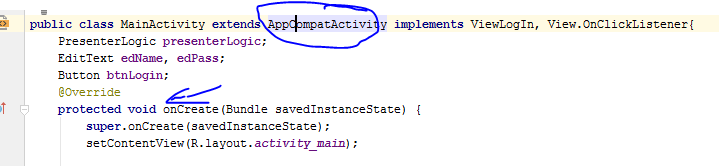
+ onResume(): được gọi khi người dùng bắt đầu tương tác với ứng dụng.

+ onPause(): gọi khi layout activity không nhận được tương tác của người dùng và không thực hiện bất kỳ đoạn mã nào.

+ onStop(): khi layout của activity đó không hiển thị nữa và được vào lại thì phương thức được gọi tới.

+ onRestart(): được gọi khi activity được khởi động trở lại (trước đó activity đang tạm dừng).

Ví dụ trong đoạn mã này thì MainActivity đã ghi đè phương thức onCreate(). Các phương thức onStart(), onResume(), onPause()... khai báo ở trong class cha AppCompatActivity.

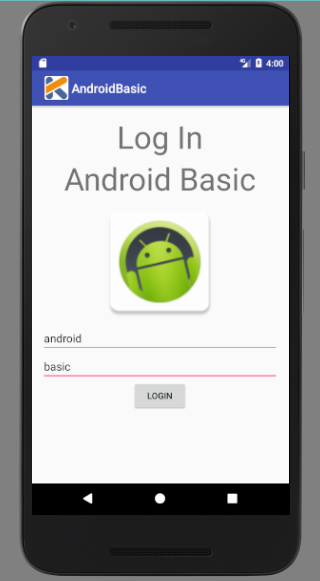


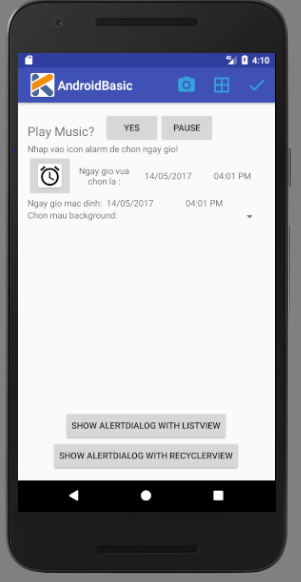
2. Content Providers:

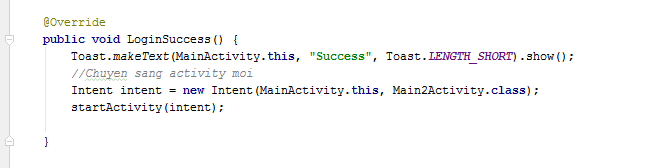
3.Intents/Filters:

Intent: Dùng để mở Activity khác và có thể kiểm soát kết quả trả về.

Ví dụ:







Khi người dùng nhập đúng username và pass đăng nhập sau đó click button LOGIN. Intent được khởi tạo bắt sự kiện click button sẽ giúp chuyển tới layout activity mới mà ở đây là từ MainActivity sang Main2Activity.

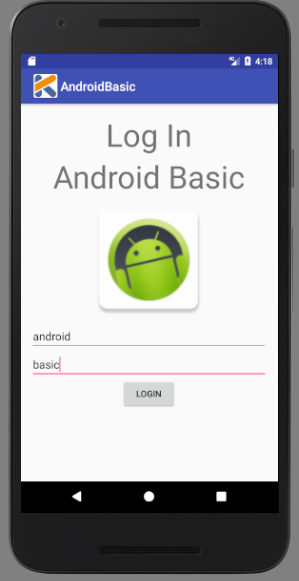
**II. User Interface**

1. UI layouts

1.1. Layout Types

* LinearLayout: là một layout mà các view con trong layout có thể sắp xếp một cách tuyến tính theo chiều dọc hoặc chiều ngang. Được thiết lập qua phương thức setOrientation().

Ví dụ tại màn hình login sử dụng LinearLayout:



TextViewww

Button

EditText

EditText

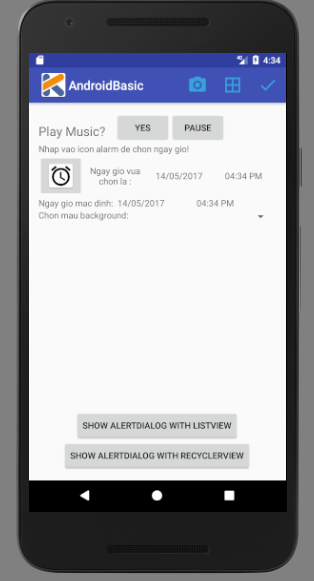
ImageView

TextView

Layout sử dụng LinearLayout, được bố cục tuyến tính theo chiều dọc (vertical):

* RelativeLayout: là một layout trong đó các view gốc và các view con, view con và view con có mối quan hệ tương đối với nhau về vị trí (trái, phải, giữa, trên, dưới...).

Ví dụ:



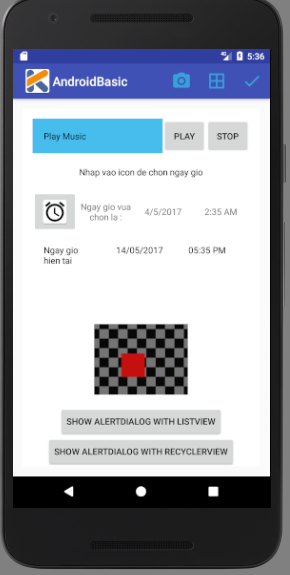
Tạm thời bỏ qua các view trên, xét 2 button cuối cùng màn hình:

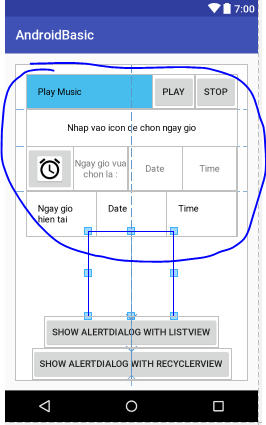
Button 1

Button 2

Ta thấy được mối quan hệ giữa 2 button: Button 1 ở trên Button 2. Và vị trí của 2 button với View gốc (toàn màn hình điện thoại).

* TableLayout: là 1 một layout mà tổ chức các view con xếp chung các hàng hoặc các cột. ví dụ:

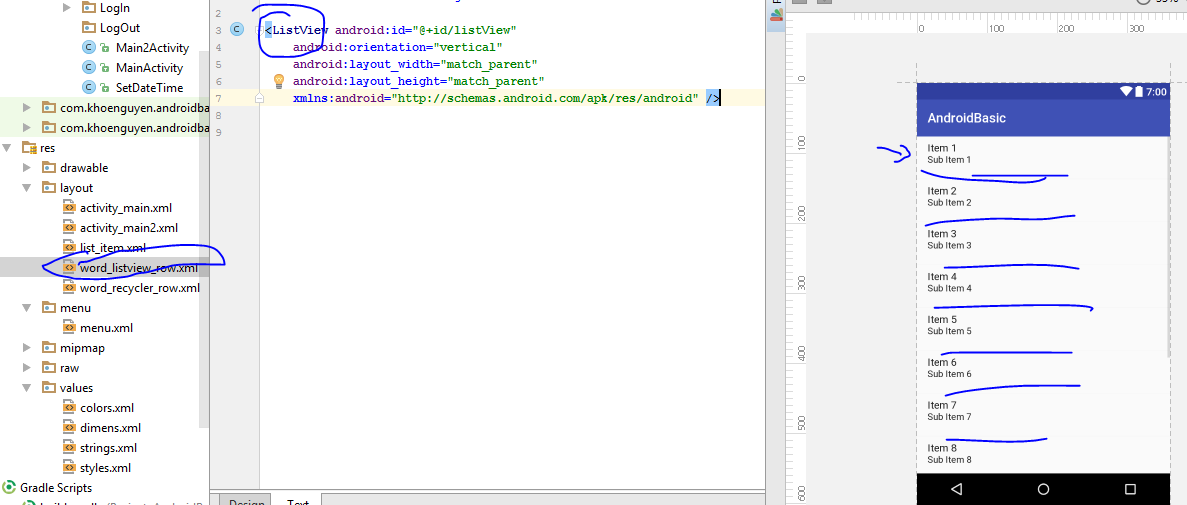




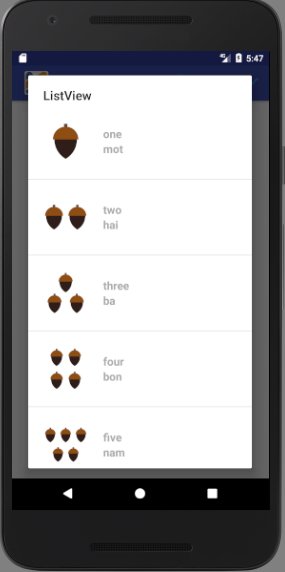
Vùng khoanh tròn chính là vùng sử dụng table layout.

* ListView: là một View group hiển thị một danh sách có thế cuộn được trên màn hình. Cuộn ngang hoặc dọc, tùy người lập trình cài đặt. Ví dụ: danh sách các bài hát, danh bạ ...

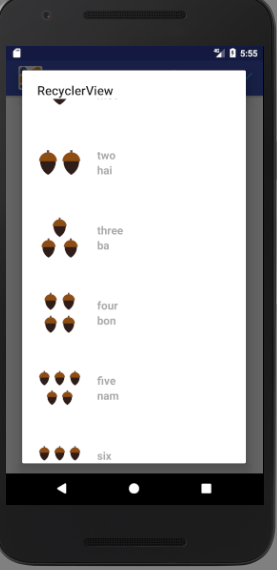
Trong project ta truy cập vào layout



và đây là kết quả của ListView kết hợp với ArrayAdapter trên điện thoại khi chạy:



* RecyclerView: là một view group có tác dụng giống ListView tuy nhiên có khả năng tái sử dụng View bằng cách sử dụng ViewHolder, từ đó cải thiện hiệu suất giúp ứng dụng chạy mượt mà hơn. Ví dụ kết quả in ra màn hình khi sử dụng Recycler + Recycler.Adapter



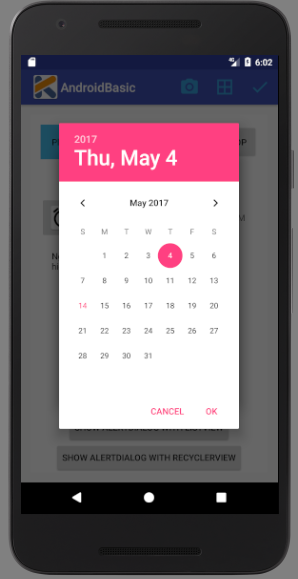
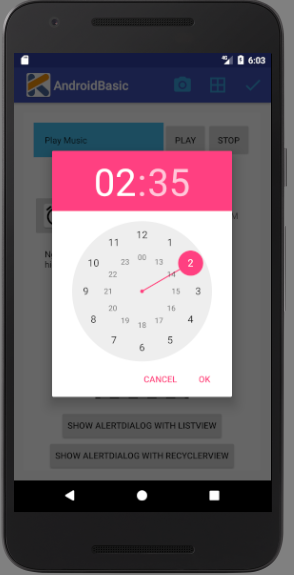
- Thấy màn hình cũng tương tự như của ListView tuy nhiên bạn sẽ cảm nhận rõ hơn sự thay đổi khi lướt màn hình, độ mượt tăng lên rõ rệt.

* GribView: là một view group hiện thị các item theo hai chiều. Thường được kết hợp với ListAdapter. Ví dụ: ứng dụng Note nhắc nhở ...

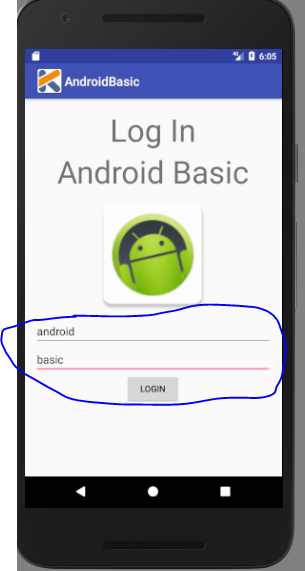
2.2. Một số UI controls:

* DatePicker, TimePicker: là một UI cho phép xem ngày giờ, thiết lập ngày giờ.

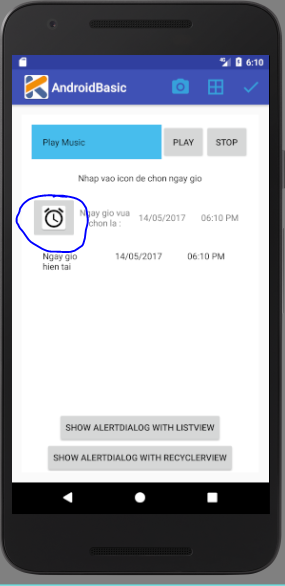
ví dụ. Trong project: khi click vào icon đồng hồ thì hai UI này hiển thị dưới dạng dialog cho phép thiết lập ngày giờ, và in ra cả ngày giờ mặc định.

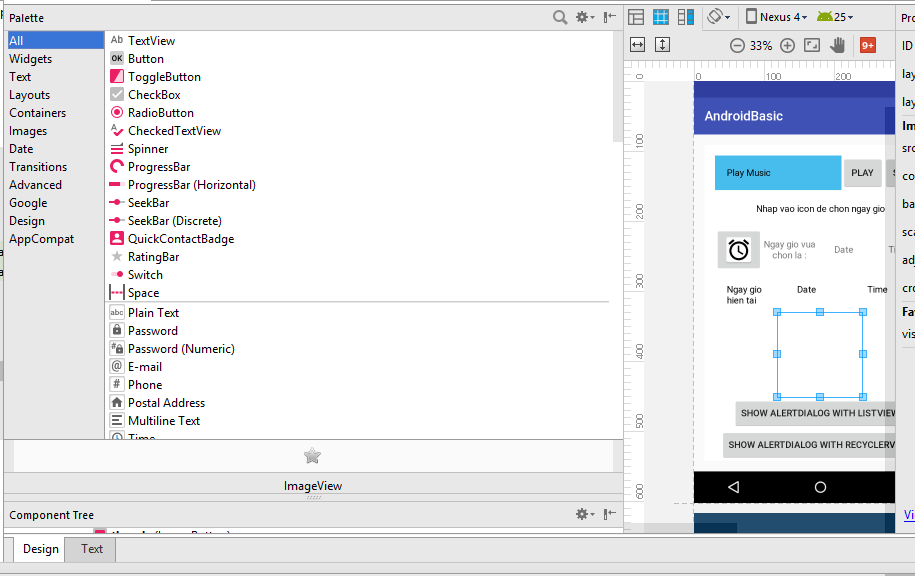
* EditText: cho phép nhập một đoạn văn bản. Ví dụ như trên màn hình đăng nhập có 2 EditText cho phép nhập username và password.



* Button: cho phép tạo một nút bấm. Từ đó người lập trình có thể tiến hành cài đặt sự kiện vào button để có tương tác cho user khi chạm vào nút bấm. Ví dụ trên hình trên có button LOGIN. Dùng để xác nhận đăng nhập.
* ImageButton: mang chức năng hiển thị hình ảnh và 1 button. Ví dụ: một imageButton trong project



* Ngoài ra còn rất nhiều UI controls chúng ta có thể sử dụng.



* + ToggleButton: một nút có 2 chế độ on/off.
  + CheckBox: một hộp giao diện nhỏ cho phép người dùng check. Ứng dụng lấy lựa chọn.
  + Spinner: tạo một thanh cuộn ẩn, khi click vào mới hiện thị các mục bên trong.
  + ProgressBar: tạo một thanh tiến trình ảo minh họa cho trường hợp ứng dụng đang được load.
  + RadioButton: tạo một hộp giao diện tròn. Chức năng giống checkbox.
  + RadioGroup: tạo một tập hợp các radio Button.

2.3. Coding Convention & Style

* Quy ước đặt tên các biến:
  + - * non-public, non-static bắt đầu với m.
      * static: bắt đầu với s.
      * others: bắt đầu bằng chứ thường.
      * public static final: tất cả là chữ hoa với gạch ngang được cho phép giữa các từ.

**III. Mô hình MVP**

Chia nhỏ code ra, giúp dễ quản lý, lập trình, debug.

MVP viết tắt của 3 từ: Model, View, Presenter

Được đơn giản hóa bằng sơ đồ sau:

