BÁO CÁO BÀI TẬP 1

Họ và tên: Hà Hoàng Oanh

Lớp: CNTT2-K61

MSV: 201200267

1. Tìm hiểu

- Các dạng layout trên Android:

+ConstrainLayout: Đây là dạng layout mà các view nằm trong đó sẽ được định hình vị trí tương đối với các view khác.

+FrameLayout: là dạng layout cơ bản nhất khi gắn các view lên layout này thì nó sẽ luôn giữ các view này ở phía góc trái màn ảnh và không cho chúng ta thay đồi vị trí của chúng, các view mang vào sau sẽ đè lên view ở trước trừ khi bạn thiết lập transparent cho view sau đó.

+TableLayout: nó sẽ sắp xếp các View con bên trong thành dạng bảng. Mỗi hàng là một đối tượng view TableRow bên trong TableRow chứa các View con, mỗi View con này nằm ở vị trí một ô bảng (cell). Cột / hàng trong bảng bắt đầu từ số 0.

+LinearLayout: là loại layout hay được sử dụng nhất , LinearLayout sẽ bố trí các view theo dạng khối và không đè lên nhau. Linear Layout có hai chiều bố trí bố cục là:

* Vertical Orientation – các view bên trong sẽ được sắp xếp theo chiều dọc
* Horizontal Orientation – Tương tự nhưng theo chiều ngang

+RelativeLayout: là loại Layout mà trong đó vị trí của mỗi view con sẽ được xác định so với view khác hoặc so với thành phần cha của chúng thông qua ID. Bạn có thể sắp xếp 1 view ở bên trái, bên phải view khác hoặc ở giữa màn hình.

+GridLayout: đơn giản chỉ là 1 layout dạng lưới và ta có thể chia các cột và dòng cho cái lưới đó, các view sẽ được dặt vào các ô trong cái lưới này

- Thiết lập kích thước cho các view, các đơn vị đo kích thước:

+ Thuộc tính xác định kích thước: layout\_width, layout\_height

+ Đơn vị đo kích thước tuyệt đối: dp,px

+ Đơn vị đo kích thước tương đối:

+ wrap content: Đủ để chứa nội dung

+ match parent: Bằng vs thẻ cha

- Font chữ: kiểu, màu, cỡ

+Thuộc tính textStyle: để set kiểu chữ cho text có thể nhận phối hợp các giá trị *bold* (đậm), *italic* (nghiêng), *normal* (bình thường)

+ Thuộc tính fontFamily: để set font chữ cho text

+ Thuộc tính textColor: để set màu chữ cho text, có thể truyền vào giá trị là các mã màu hoặc là các giá trị được set up sẵn ở file colors.xml

+ Thuộc tính textSize: để set cỡ chữ cho text, có thể sử dụng các đơn vị sp, dp, mm

- Căn lề cho nội dung trong view:

+ Để căn lề cho nội dung trong View, ta sử dụng các thuộc tính padding (padding-Top, padding-Bottom, padding-Left, Padding-Right, padding\_Start, padding\_End)

+ Đơn vị sử dụng dp, px

- Viền, nền

+ Thuộc tính background: để set màu nền cho View

+ Tạo viền:

Tạo 1 file vien.xml trong folder drawable

Sử dụng thẻ shape, bên trong là thẻ stroke. Trong thẻ stroke dùng thuộc tính width (độ dày viền) và color(màu viền)

Để sử dụng file này tạo viền cho các view, ta dùng thuộc tính android:backround=”@drawable/vien”

- Bắt sự kiện cho view

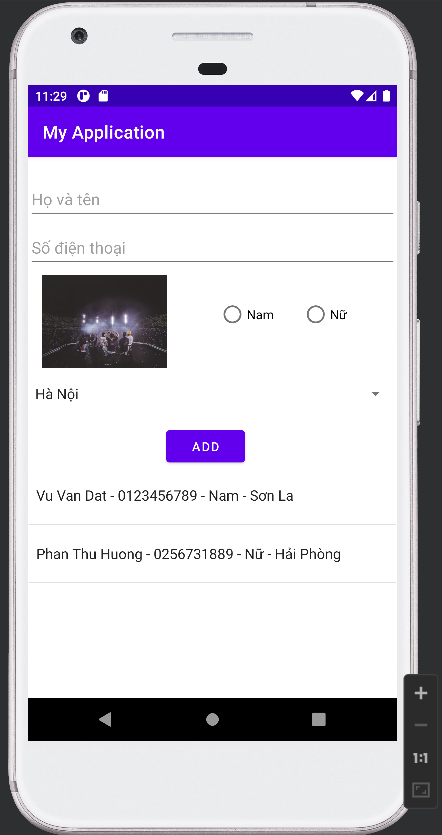
+ Khai báo đối tượng thuộc lớp tương ứng với view

+ Sau đó gọi hàm findViewById với tham số đầu vào là Id của view

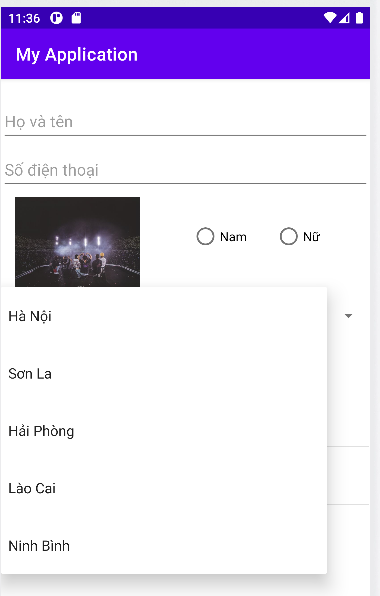
+ Sau đó gọi tới các đối tượng listener tương ứng

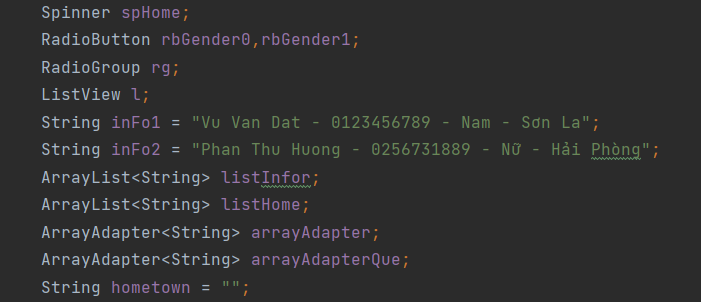
2. Giao diện

1. Hiển thị giao diện

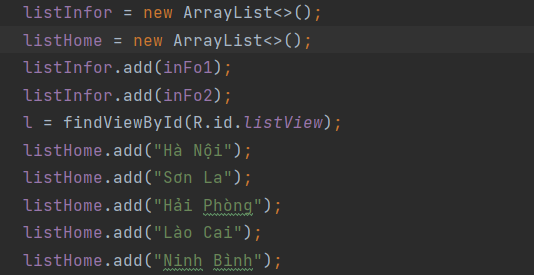


1. Tạo spinner nhập quê quán

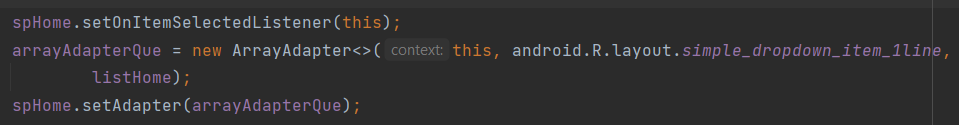




Khai báo arrayList listHome lưu thông tin quê quán

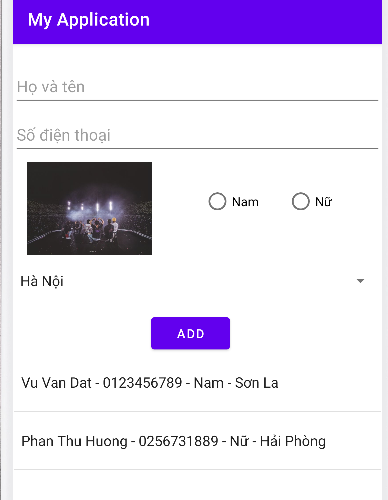


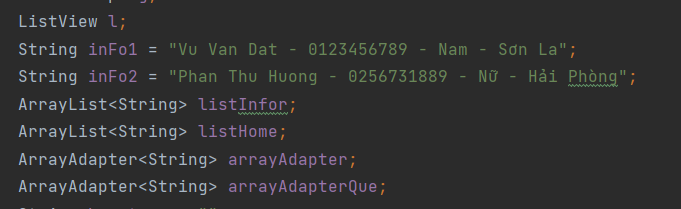
Add các tỉnh thành vào listHome



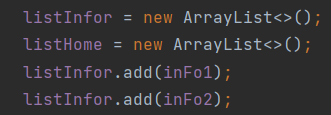
Bắt sự kiện select vào item của spinner, hiện ra danh sách tỉnh thành bằng setAdapter

1. Hiển thị một số dữ liệu mẫu fix sẵn cho listview

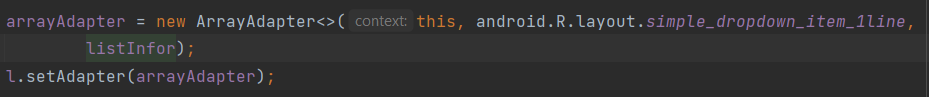




Tạo 2 info có sẵn và arrayList listInfo lưu các thông tin

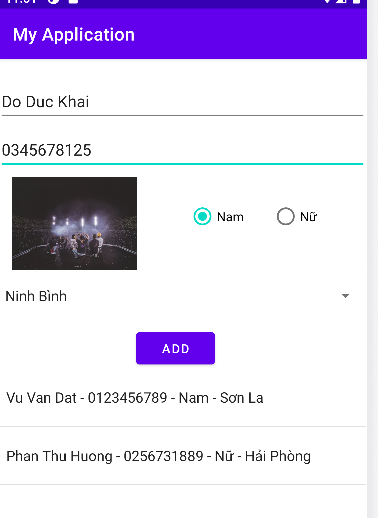


Sau đó add 2 info vào listInfo

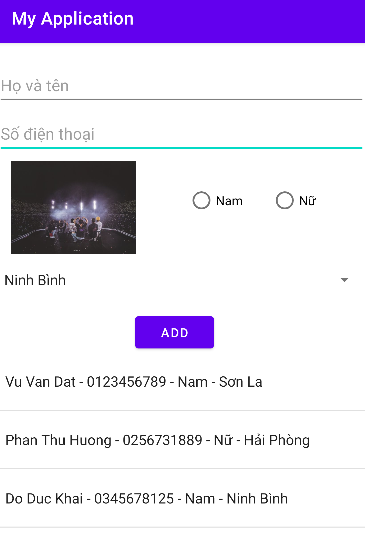


Hiển thị bằng setAdapter

1. Nhấn nút Add sẽ tạo thêm 1 phần tử cho listview từ các view ở trên



Sau khi add:





Bắt sự kiện click vào button.

Tạo StringBuilder info sử dụng để tạo chuỗi có thể thay đổi.

Sau khi get được tên và số điện thoại set lại 2 trường này thành rỗng.

Giới tính nào được check thì get text, sau đó set lại thành không check.

Thêm vào info các dấu gạch ngang.

Sau đó add info vào listInfo ban đầu rồi hiển thị sau khi dữ liệu được thêm vào.

\*FULL Code

