* Linear Layout:
  + GroupView đặt các Vieww trong đó theo thứ tự ngang hoặc dọc
  + Một số thuộc tính cơ bản:
    - Orientation:
      * Có thể thay đổi bằng setOrientation() bằng mã chương trình đang chạy
    - Fill Model:
      * Các View của linearlayout bắt buộc p thiết lập các chỉ số kích thước
    - Weight(giống thuộc tính flex trong css): Mặc định = 0
    - Gravity: Dùng xác định cách căn lề của các đối tượng trên màn hình
      * Mặc định sẽ căn lề bên trái
      * Nếu chỉ gọi android:gravity\_layout thì sẽ là căn lề của cả view đó, android:gravity thì sẽ chỉ căn lề cái attribute chính của view đó(Ví dụ như text trong text\_view)
    - Margin & padding:
      * Margin: Lề ngoài view, thiết lập không gian giữa các view con của viewgroup
      * Padding: lề trong view, khoảng không gian trong view
* Relative Layout:
  + Các view trong nó sẽ tính vị trí theo các view lân cận (các view lân cận cũng có thể tự quyết)
* Absolute Layout:
  + Chỉ định các view con thông qua các tọa độ x, y
* Table Layout:
  + Gồm các View con là row được sắp xếp theo dạng bảng
  + Các row cũng là một viewgroup
  + Một số thuộc tính:
    - android:stretchColumns cho phép chỉ định các cột sẽ được kéo dài (stretched) để lấp đầy không gian còn trống
    - android:shrinkColumns chỉ định các cột sẽ bị co lại (shrinked) để tránh việc các View con tràn ra ngoài TableLayout
    - android:layout\_span áp dụng cho View con để chỉ định số ô liên tiếp trong một TableRow sẽ được hợp nhất với nhau.
    - android:layout\_column được áp dụng cho một View con trong một TableRow để chỉ định vị trí
* ConstraintLayout:
  + Tương tự relative layout, hỗ trợ kéo thả mạnh mẽ(Giống giống figma)