

BÁO CÁO THỰC HÀNH

Môn học: Phát triển ứng dụng trên thiết bị di động

Buổi báo cáo: Lab 01

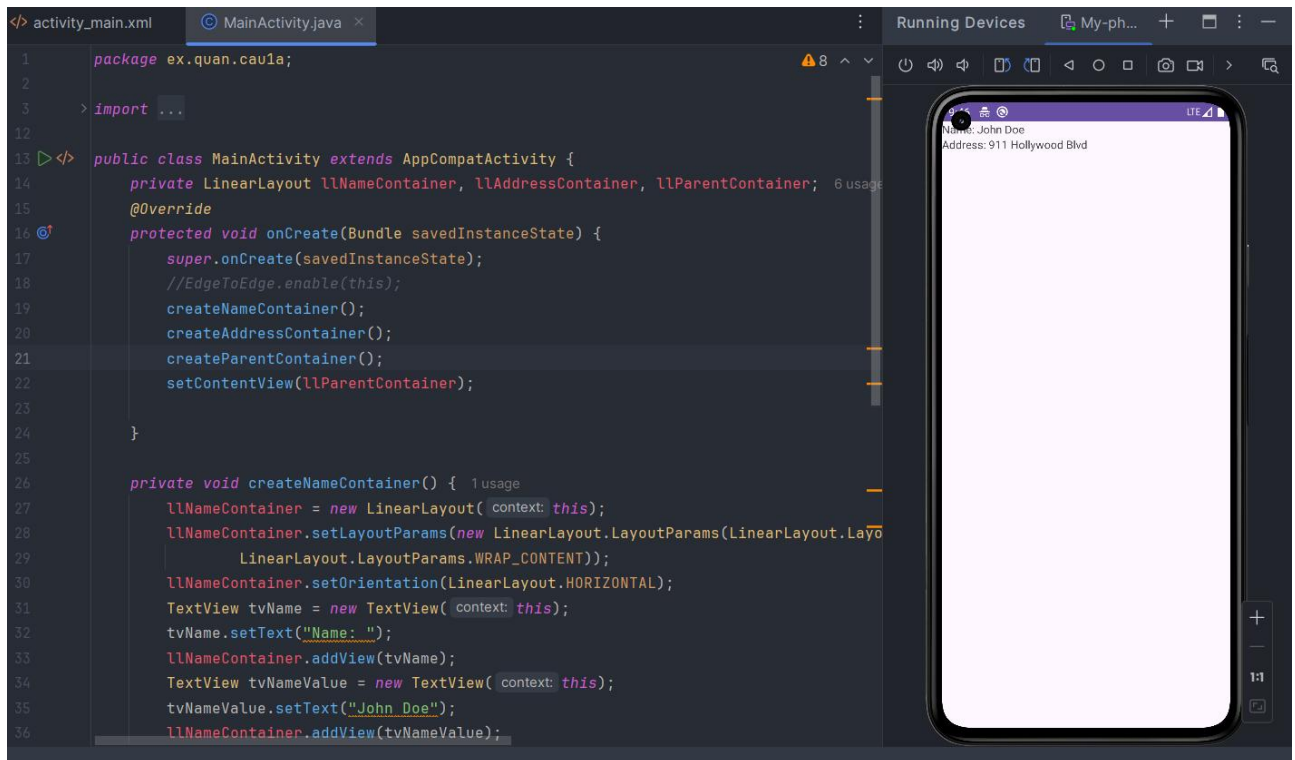
Tên chủ đề: Basic layouts

Lớp: NT118.P13.2

BÁO CÁO CHI TIẾT

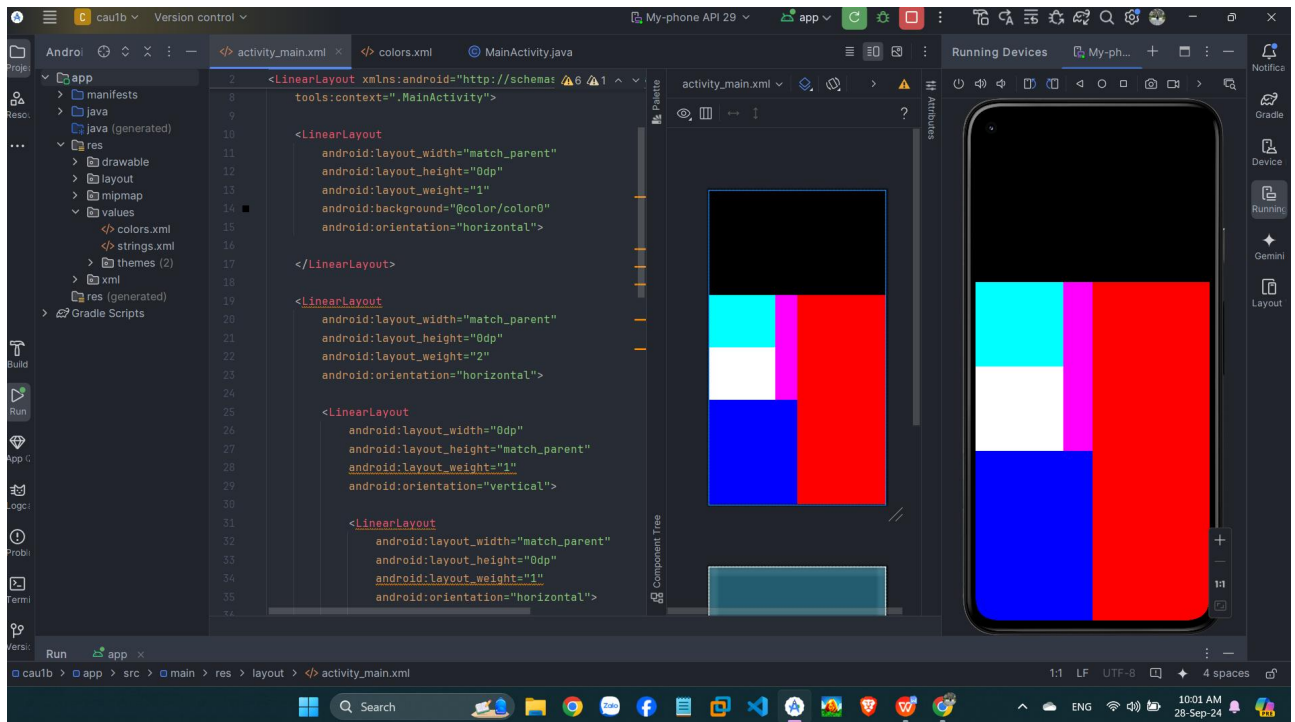
1. Linear layout

a) Khởi tạo Linear layout bằng code

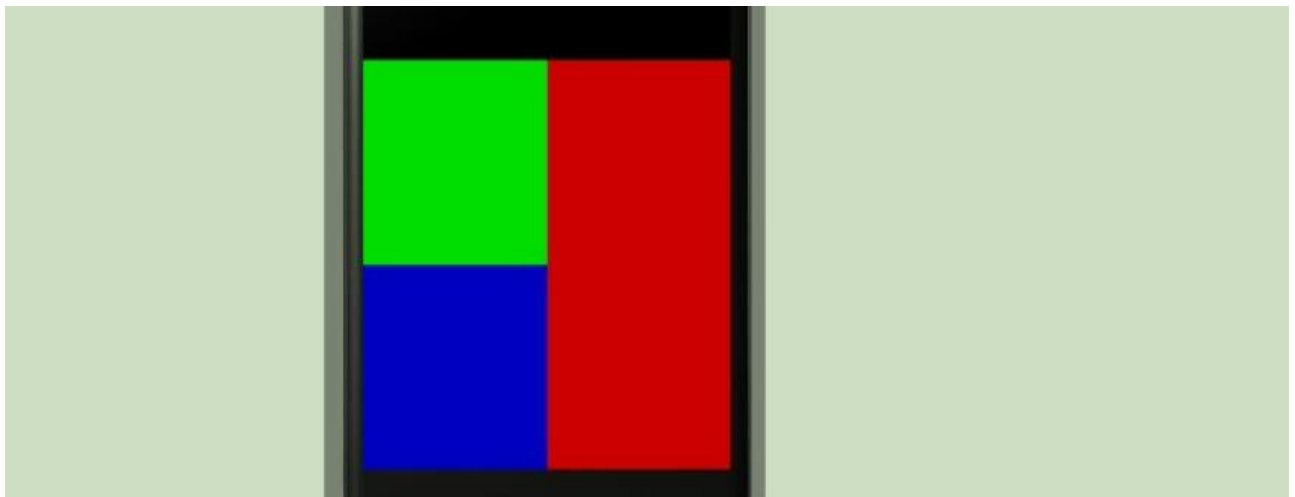


Bài 1a dùng code java để tạo một giao diện đơn giản. Ta có 4 hàm, hàm createNameContainer() dùng để tạo thông tin tên, createAddressContainer() tạo thông tin địa chỉ, CreateParentContainer() tạo container cha chứa hai container con, setContentView() thiết lập giao diện.

b) Sử dụng Linear layout trong file xml

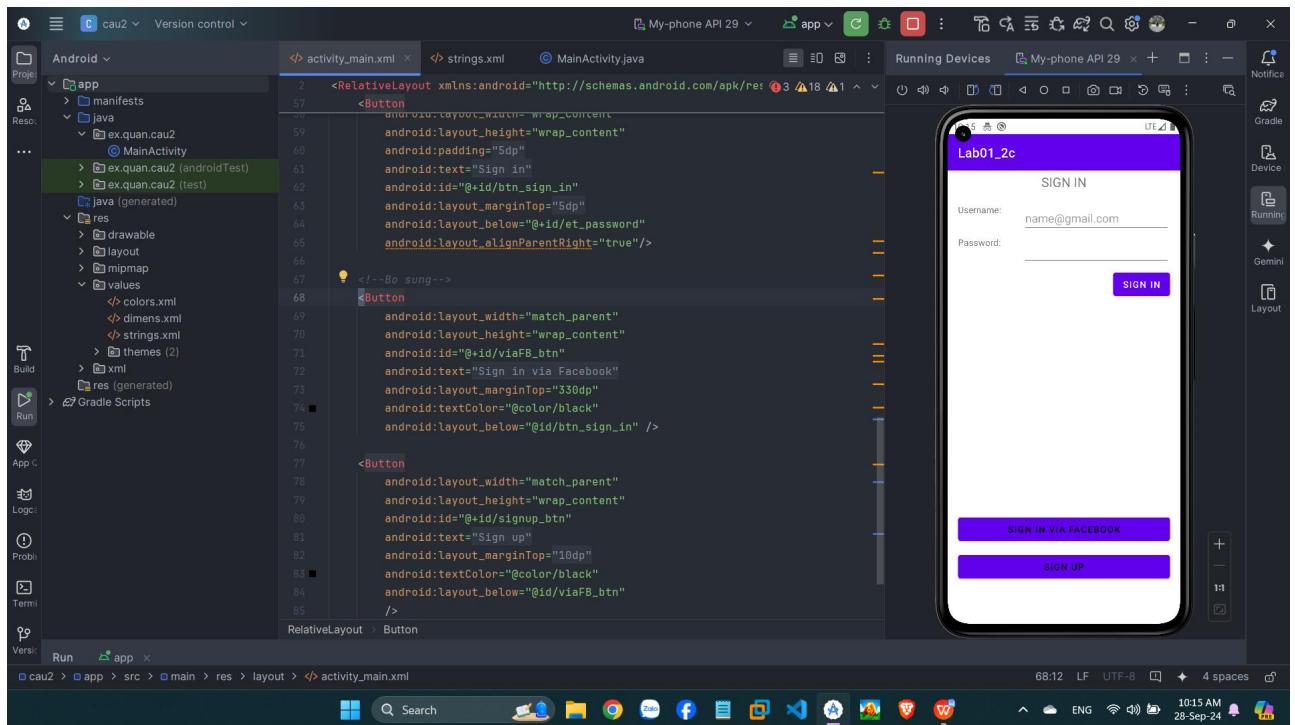


Dựa vào code đã cung cấp thì ta có giao diện dưới:



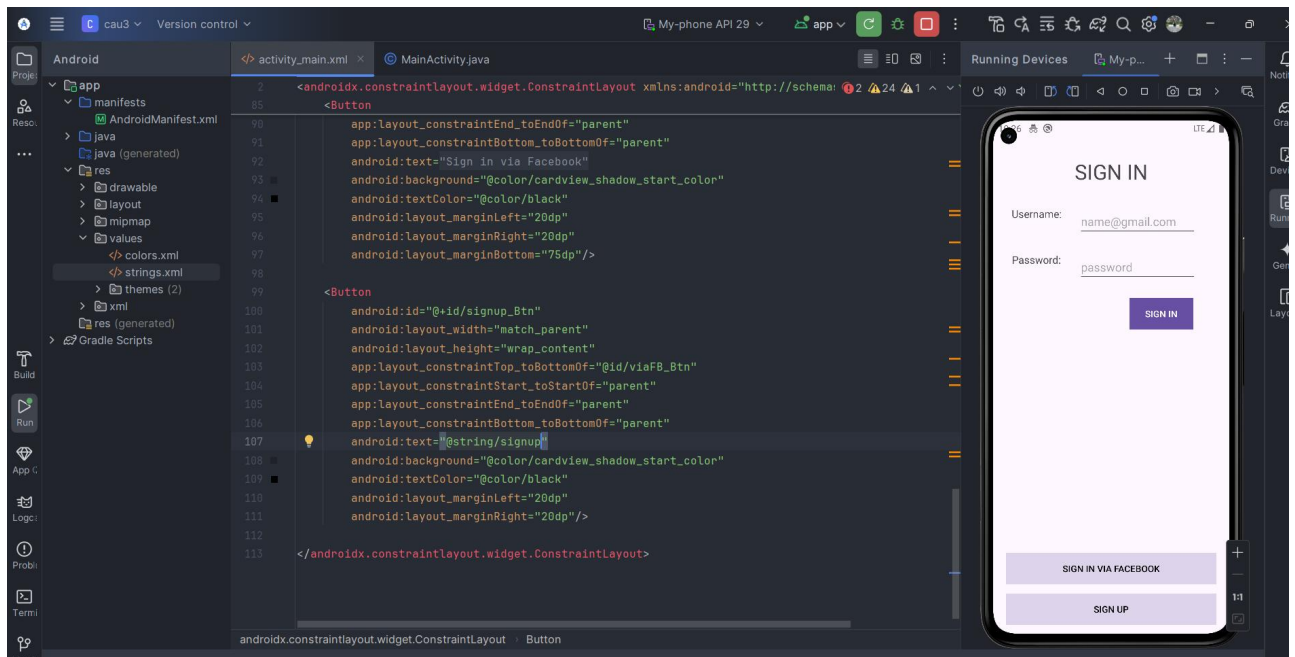
Để thay đổi giống như giao diện đầu thì ta chỉnh căn chỉnh ô màu xanh lá. Trong này ta sẽ chia thành 2 linear theo chiều ngang, bên trái chứa 2 màu, bên phải chứa màu tím, theo tỉ lệ 3:1. Trong khung có 2 màu thì ta chỉnh Linear layout theo chiều dọc với tỉ lệ 1:1 nữa nữa thì ta sẽ được kết quả

2. Relative layout



Dựa vào code đã cung cấp sẵn thì ta có giao diện, nhưng chưa có 2 button phía dưới. Ta phải thêm code như hình để được kết quả trên. Ta sẽ tạo 2 button với chiều ngang MATCH_PARENT, và chiều dọc thì WRAP_CONTENT. Tiếp đó, quan hệ giữa các button như sau: button phía trên phải nằm dưới button SIGN_IN, còn button phía dưới thì nằm dưới button phía trên. Để đặt đúng vị trí mong muốn và có khoảng cách đẹp thì ta thêm MARGINTOP để có khoảng cách giữa cạnh trên nó với view phía trên.

3. Constraint layout



Ta sẽ có ý tưởng giữa các view như Relative layout, nhưng khác ta sẽ dùng Constraint layout để thực hiện. Ví dụ như 2 button cuối, ta sẽ dùng câu lệnh dưới:

```
app:layout_constraintBottom_toBottomOf="parent"
```

Để đưa view xuống dưới, hai câu lệnh này

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintEnd_toEndOf="parent"
```

Để đưa view nằm giữa, sau đó ta thêm margin để nhìn cho cân đối và đẹp hơn.

HẾT