



## BÁO CÁO THỰC HÀNH

### Bài thực hành số 01: Basic Layouts

**Môn học:** NT118 – Phát triển ứng dụng trên thiết bị di động

**Lớp:** NT118.Q11.2

#### THÀNH VIÊN THỰC HIỆN (Nhóm xx):

STT	Họ và tên	MSSV
1	Nguyễn Lê Nhật Đăng	23520231

Điểm tự đánh giá

10

#### ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	2 ngày
Phân chia công việc	
Ý kiến (nếu có) + Khó khăn + Đề xuất, kiến nghị	Link chi tiết mã nguồn bài Lab: <a href="#">MobileApplicationDevelopment/LAB01 at main · tofumapu/MobileApplicationDevelopment</a>

Phần bên dưới của báo cáo này là báo cáo chi tiết của n thực hiện



## MỤC LỤC

A. BÁO CÁO CHI TIẾT .....	2
1. Câu 1. LinearLayout .....	2
a. Khởi tạo LinearLayout bằng code.....	2
b. Sử dụng LinearLayout trong file xml.....	4
2. Câu 2. RelativeLayout .....	6
3. Câu 3. ConstraintLayout .....	8



## A. BÁO CÁO CHI TIẾT

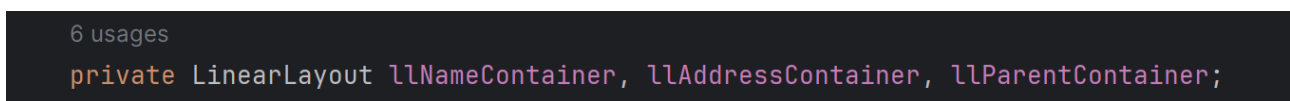
### 1. Câu 1. LinearLayout

#### a. Khởi tạo LinearLayout bằng code



Hình 1: Kết quả sau khi khởi động đoạn code

- Khai báo biến: tạo 3 LinearLayout chứa các thành phần giao diện:
  - o llNameContainer: chứa thông tin Name
  - o llAddressContainer: chứa thông tin địa chỉ
  - o llParentContainer: layout cha chứa cả 2 layout trên



Hình 2: Khởi tạo biến private



Hình 3: Hàm createNameContainer()



- Cụ thể, hàm tạo một LinearLayout ngang với bề rộng là chiếm hết màn hình và chiều cao là vừa đủ nội dung. Tạo TextView hiển thị tên và giá trị của tên, sau đó thêm 2 TextView này vào llNameContainer.

```
private void createAddressContainer() {  
    llAddressContainer = new LinearLayout( context: this);  
    llAddressContainer.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,  
        LinearLayout.LayoutParams.WRAP_CONTENT));  
    llAddressContainer.setOrientation(LinearLayout.HORIZONTAL);  
    TextView tvAddress = new TextView( context: this);  
    tvAddress.setText("Address:");  
    llAddressContainer.addView(tvAddress);  
    TextView tvAddressValue = new TextView( context: this);  
    tvAddressValue.setText("911 Hollywood Blvd");  
    llAddressContainer.addView(tvAddressValue);  
}
```

Hình 4: Hàm createAddressContainer()

- Hàm có chức năng tương tự với createNameContainer(), hiển thị nội dung (Address: 911 Hollywood Blvd).

```
private void createParentContainer() {  
    llParentContainer = new LinearLayout( context: this);  
    llParentContainer.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,  
        LinearLayout.LayoutParams.MATCH_PARENT));  
    llParentContainer.setOrientation(LinearLayout.VERTICAL);  
    llParentContainer.addView(llNameContainer);  
    llParentContainer.addView(llAddressContainer);  
}
```

Hình 5: Hàm createParentContainer()

- Hàm khởi tạo một LinearLayout cha theo dạng cột, với chiều dài và bề rộng là toàn bộ màn hình, sau đó lần lượt llNameContainer và llAddressContainer vào Layout cha.

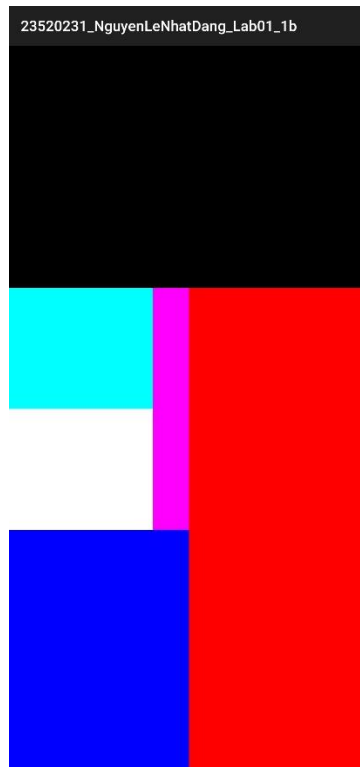
```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);  
    setContentView(R.layout.activity_main);  
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
        return insets;  
    });  
    createNameContainer();  
    createAddressContainer();  
    createParentContainer();  
    setContentView(llParentContainer);  
}
```

Hình 6: Hàm onCreate()

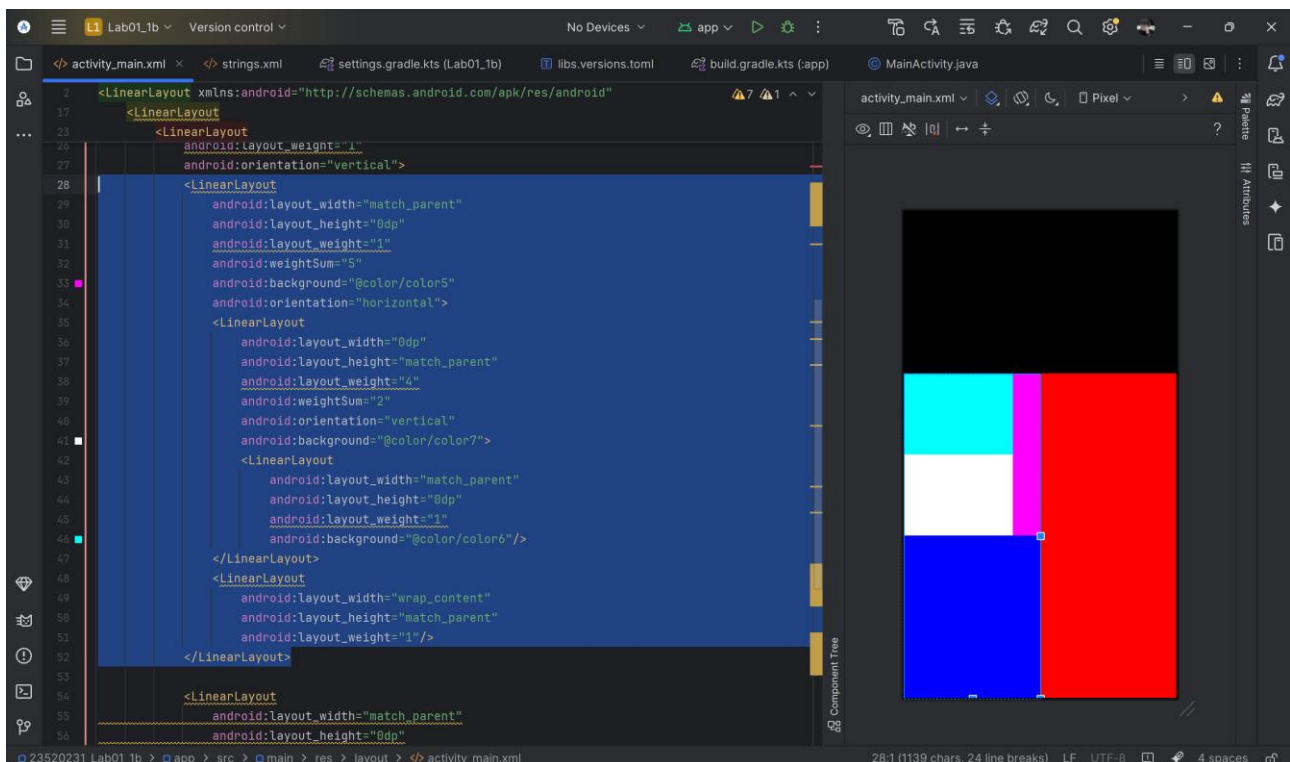
- Tại hàm onCreate(), thêm 3 hàm tạo layout động, hành động này sẽ ghi đè layout hiển thị bằng llParentContainer (nên activity\_main.xml không hiển thị trên màn hình).



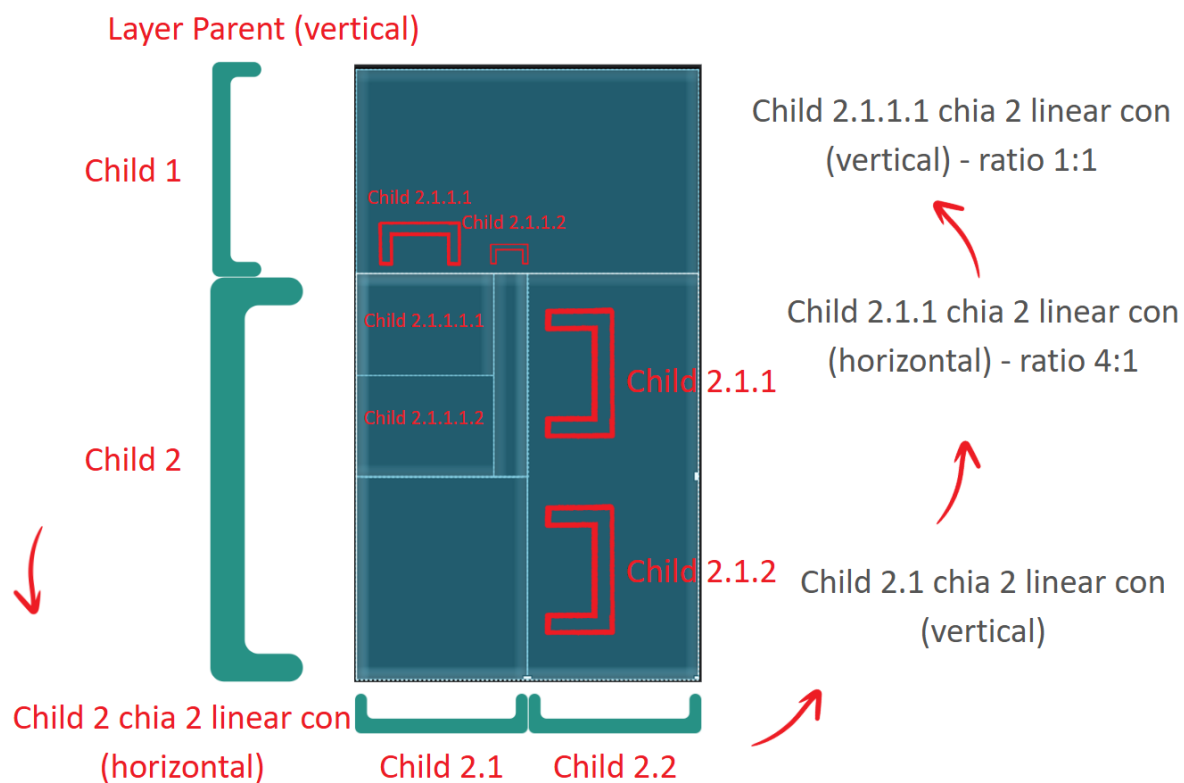
### b. Sử dụng LinearLayout trong file xml



Hình 7: Kết quả sau khi thực hiện chỉnh sửa đoạn code



Hình 8: Đoạn code sau chỉnh sửa theo yêu cầu đề bài



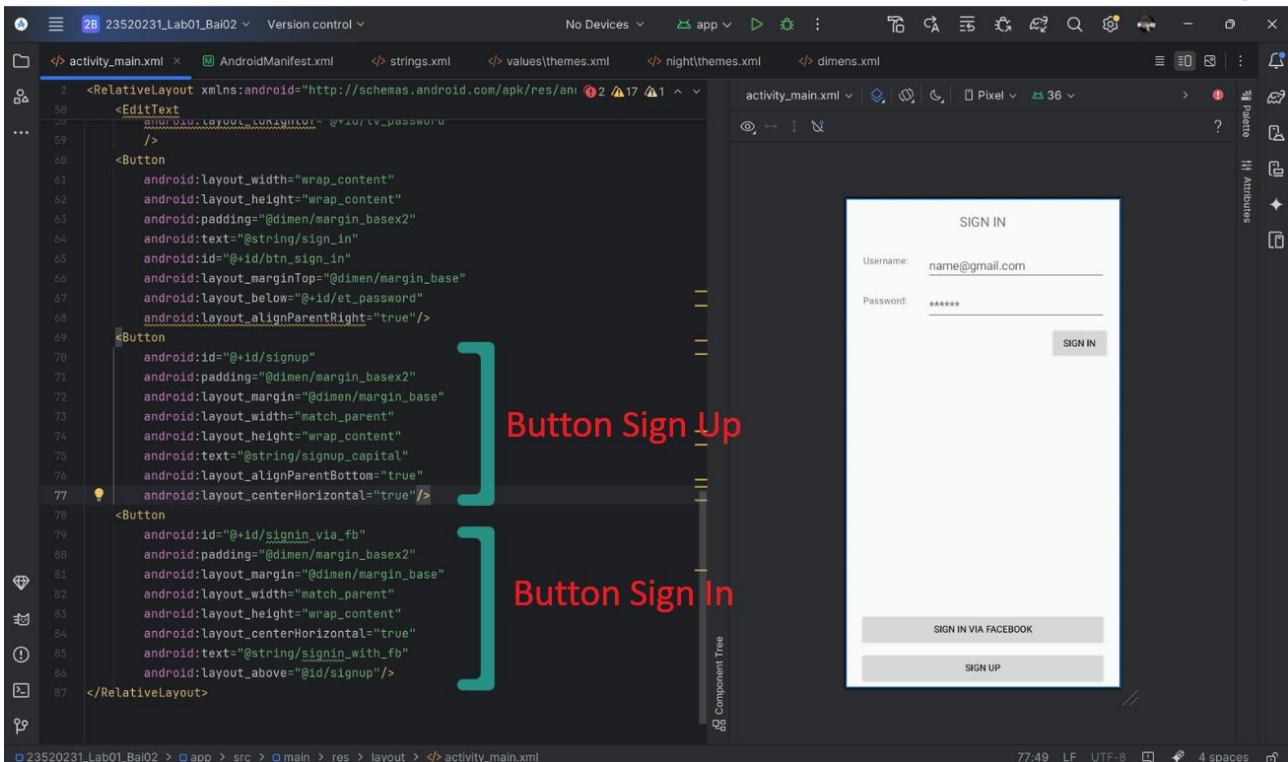
Hình 9: Bố cục thiết kế của lab theo mức độ hiểu



### 2. Câu 2. RelativeLayout

A screenshot of a mobile application interface. At the top, there is a black header bar with the text 'NguyenLeNhatDang\_23520231\_Lab01\_2c' in white. Below the header, the main content area has a light gray background. At the top of this area, the text 'SIGN IN' is centered. Below it, there are two input fields. The first is labeled 'Username:' and contains the text 'name@gmail.com'. The second is labeled 'Password:' and contains the text '\*\*\*\*\*'. To the right of the password field, there is a gray button with the text 'SIGN IN'. At the bottom of the screen, there are two more gray buttons. The top one is labeled 'SIGN IN VIA FACEBOOK' and the bottom one is labeled 'SIGN UP'.

Hình 10: Kết quả sau khi thực hiện chỉnh sửa đoạn code



Hình 11: Nội dung đoạn code đã thực hiện

- Đề bài yêu cầu thiết kế 2 Button “SIGN IN VIA FACEBOOK” và “SIGN UP”. Trong đó:
  - Button “SIGN UP”
    - Có chiều rộng là match\_parent và chiều dài là wrap\_content.
    - Text là “SIGN UP”, khai báo trong string.xml.
    - Được căn dưới và căn giữa nhờ android:layout\_alignParentBottom và android:layout\_centerHorizontal.
    - Có sử dụng padding và margin cho Button.
  - Button “SIGN IN VIA FACEBOOK”
    - Có chiều rộng là match\_parent và chiều dài là wrap\_content.
    - Text là “SIGN IN VIA FACEBOOK”, khai báo trong string.xml.
    - Button có quan hệ với button SIGN UP là android:layout\_above (button SIGN IN VIA FACEBOOK sẽ nằm trên button SIGN UP).
    - Được căn giữa nhờ android:layout\_centerHorizontal.
    - Có sử dụng padding và margin cho Button.





### 3. Câu 3. ConstraintLayout

NguyenLeNhatDang\_23520231\_Lab01\_Bai03

SIGN IN

Username:

Password:

SIGN IN

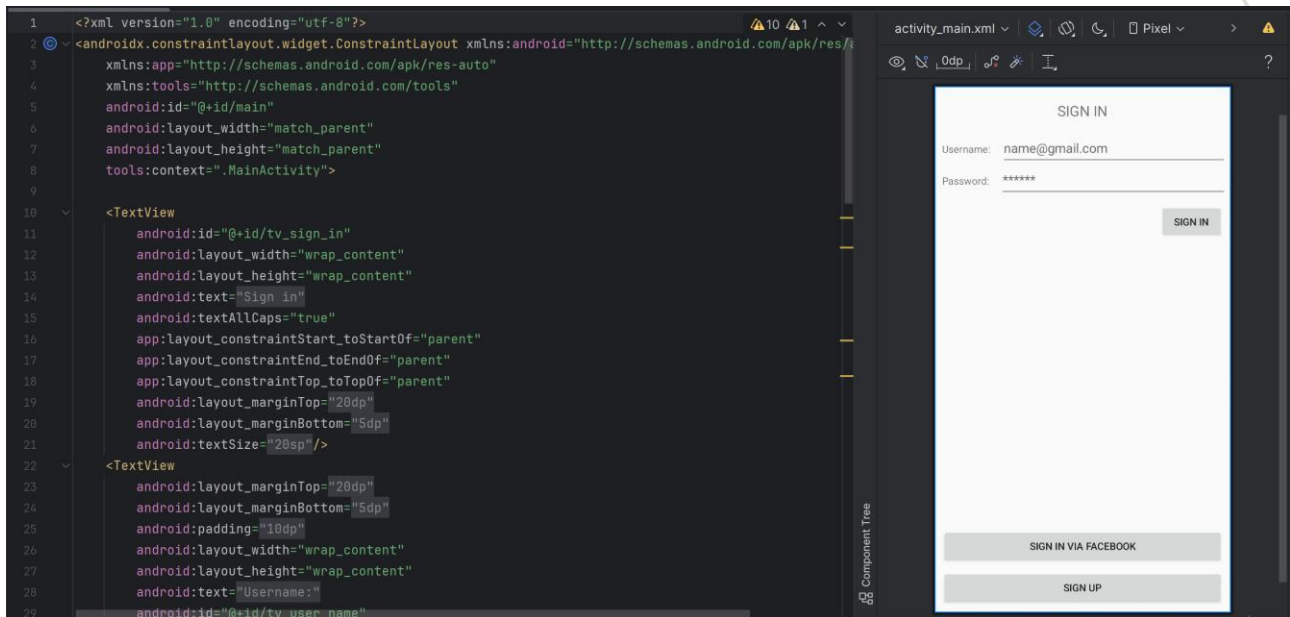
SIGN IN VIA FACEBOOK

SIGN UP

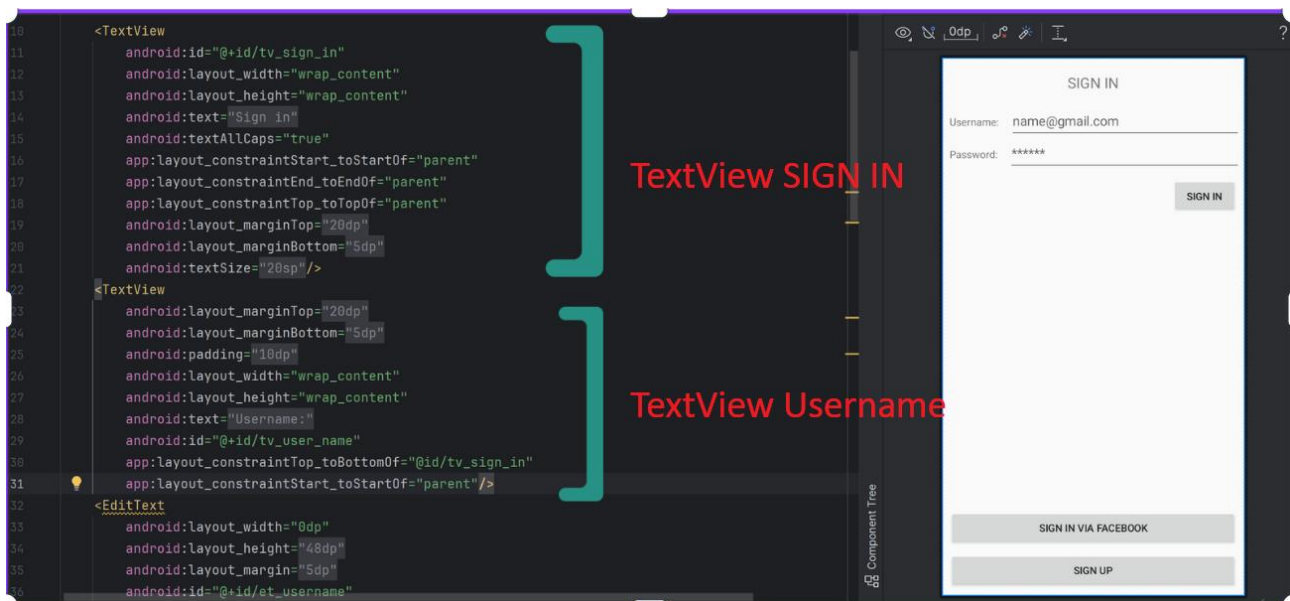
Hình 12: Kết quả sau khi thực thi đoạn code

Nội dung giao diện giống như bài 2, nhưng được thiết kế bằng ConstraintLayout

## Bài thực hành số 01: Basic Layouts



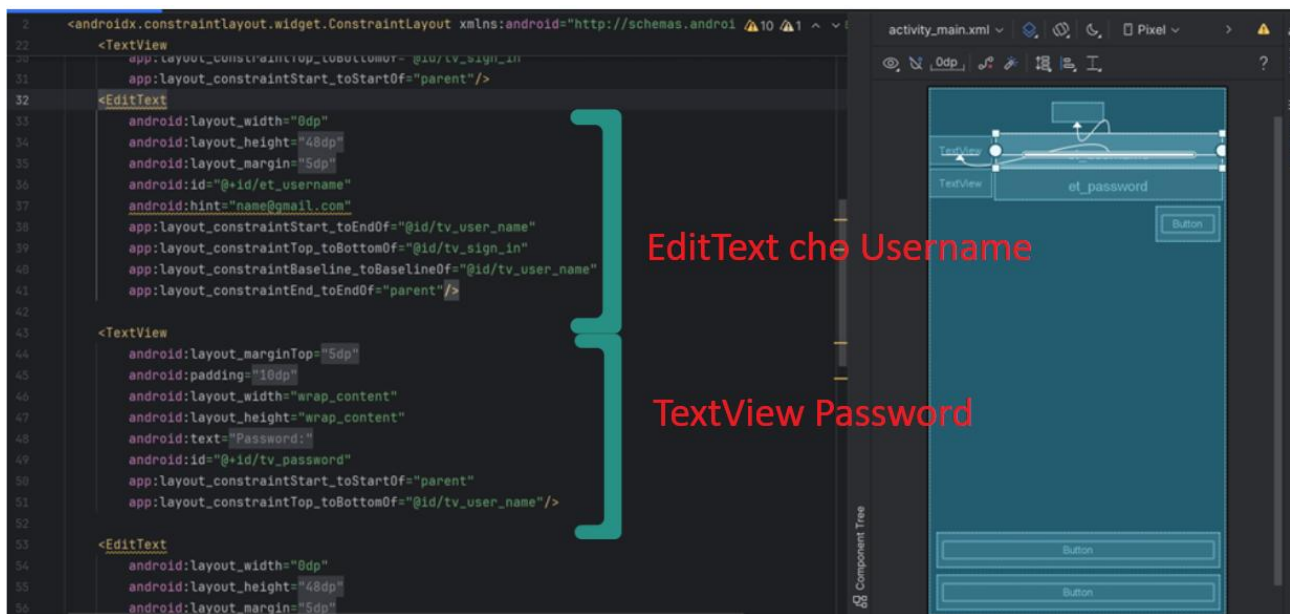
Hình 13: Thiết kế giao diện bằng ConstraintLayout



Hình 14: Thiết kế giao diện bằng ConstraintLayout (P2)

- TextView SIGNIN:
  - Có chiều rộng là wrap\_content, chiều cao là wrap\_content.
  - Nội dung là “SIGN IN”, được in hoa nhờ textAllCaps.
  - Ràng buộc chữ được đưa ra giữa màn hình với  
app:layout\_constraintStart\_toStartOf="parent";  
app:layout\_constraintEnd\_toEndOf="parent";  
app:layout\_constraintTop\_toTopOf="parent".
  - Sử dụng margin để ngăn cách với các View khác.
- TextView Username:

- Có chiều rộng là wrap\_content, chiều cao là wrap\_content.
- Nội dung là Username, được khai báo trong string.xml.
- Ràng buộc với TextView SIGNIN: app:layout\_constraintTop\_toBottomOf (Nằm bên dưới TextView SIGNIN)
- Ràng buộc nằm bên trái với parent: app:layout\_constraintStart\_toStartOf
- Sử dụng margin và padding để cách đều.



Hình 15: Thiết kế giao diện với ConstraintLayout (P3)

### - EditText cho Username:

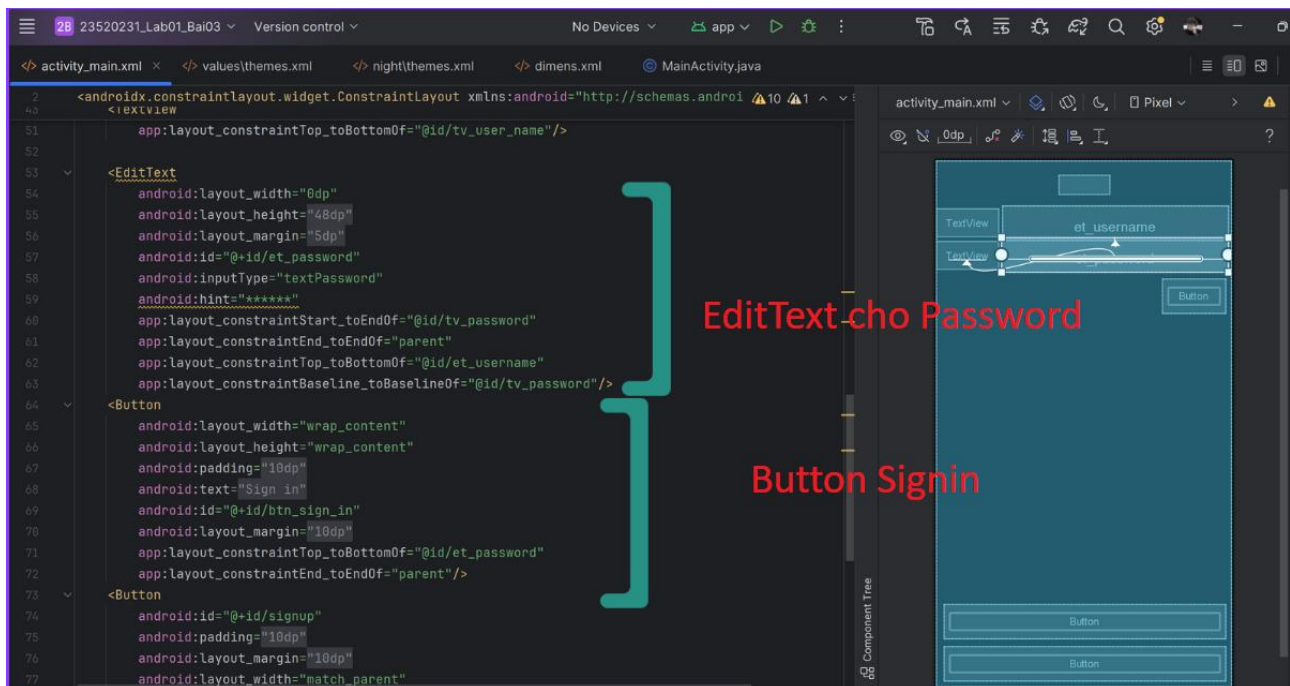
- Có chiều rộng là 0dp (sử dụng constraint\_layout sẽ được ràng buộc theo thiết kế), chiều cao là 48dp (là minimum size cho touch target).
- Nội dung hint là [name@gmail.com](mailto:name@gmail.com), dùng để gợi ý người dùng.
- Ràng buộc nằm ở dưới TextView SIGN UP: app:layout\_constraintTop\_toBottomOf.
- Ràng buộc nằm bên phải TextView Username: app:layout\_constraintStart\_toEndOf.
- Ràng buộc với parent: app:layout\_constraintEnd\_toEndOf.
- Sử dụng marginTop và padding để cách đều theo đề bài.

### - TextView Password:

- Có chiều rộng là wrap\_content, chiều cao là wrap\_content.
- Nội dung là Password, được khai báo trong string.xml.
- Ràng buộc nằm dưới TextView Username: app:layout\_constraintTop\_toBottomOf.
- Ràng buộc nằm bên trái với parent: app:layout\_constraintStart\_toStartOf



- Sử dụng margin và padding để cách đều.



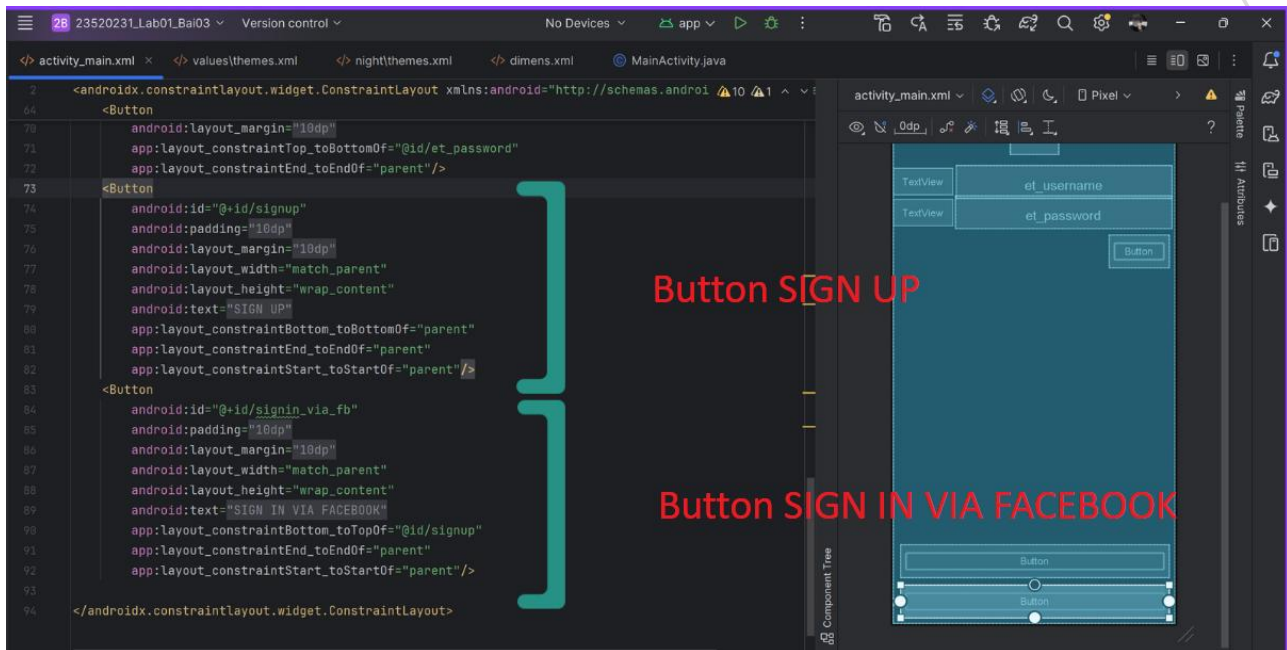
Hình 16: Thiết kế giao diện với ConstraintLayout (P4)

### - EditText cho Password:

- Có chiều rộng là 0dp (sử dụng constraint\_layout sẽ được ràng buộc theo thiết kế), chiều cao là 48dp (là minimum size cho touch target).
- Nội dung hint là \*\*\*\*\*.
- Ràng buộc nằm ở dưới EditText cho Username: `app:layout_constraintTop_toBottomOf`.
- Ràng buộc nằm bên phải TextView Password: `app:layout_constraintStart_toEndOf`.
- Ràng buộc với parent: `app:layout_constraintEnd_toEndOf`.
- Ràng buộc Baseline của EditText và TextView Password bằng nhau.
- Sử dụng `marginTop` và `padding` để cách đều theo đề bài.

### - Button Signin

- Có chiều rộng là `wrap_content`, chiều cao là `wrap_content`.
- Nội dung hiển thị là Sign In, được khai báo trong `string.xml`.
- Ràng buộc nằm dưới EditText Password: `app:layout_constraintTop_toBottomOf`.
- Ràng buộc nằm bên phải với parent: `app:layout_constraintEnd_toEndOf`.
- Có sử dụng `marginTop` để cách đều.



Hình 17: Thiết kế giao diện với ConstraintLayout (P5)

## - Button Sign up

- Có chiều rộng là match\_parent (độ phủ của nút được kéo dài ra theo chiều dài màn hình), chiều cao là wrap\_content.
- Nội dung hiển thị là SIGN UP, được khai báo trong string.xml.
- Ràng buộc nằm giữa và dưới màn hình với parent:  
app:layout\_constraintEnd\_toEndOf;  
app:layout\_constraintBottom\_toBottomOf;  
app:layout\_constraintStart\_toStartOf.
- Có sử dụng margin và padding để cách đều.

## - Button Sign in via Facebook

- Có chiều rộng là match\_parent (độ phủ của nút được kéo dài ra theo chiều dài màn hình), chiều cao là wrap\_content.
- Nội dung hiển thị là SIGN IN VIA FACEBOOK, được khai báo trong string.xml.
- Ràng buộc nằm giữa màn hình với parent:  
app:layout\_constraintEnd\_toEndOf; app:layout\_constraintStart\_toStartOf.
- Ràng buộc nằm trên nút SIGN UP với:  
app:layout\_constraintBottom\_toTopOf.
- Có sử dụng margin và padding để cách đều.