



# 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

Điểm tự đánh giá

10

### THÀNH VIÊN THỰC HIỆN (Nhóm xx):

STT	Họ và tên	MSSV
1	Nguyễn Lê Nhật Đăng	23520231

### 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

```
Name: John Doe
Address: 911 Hollywood Blvd
```

```
NguyenLeNhatDang_23520231_Lab01_1a
```

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

```
6 usages
private LinearLayout llNameContainer, llAddressContainer, llParentContainer;
```

Hình 2: Khởi tạo biến private

```
private void createNameContainer() {
    llNameContainer = new LinearLayout(context);
    llNameContainer.setLayoutParams(new LinearLayout.LayoutParams(LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.WRAP_CONTENT));
    llNameContainer.setOrientation(LinearLayout.HORIZONTAL);
    TextView tvName = new TextView(context);
    tvName.setText("Name: ");
    llNameContainer.addView(tvName);
    TextView tvNameValue = new TextView(context);
    tvNameValue.setText("John Doe");
    llNameContainer.addView(tvNameValue);
}
```

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.

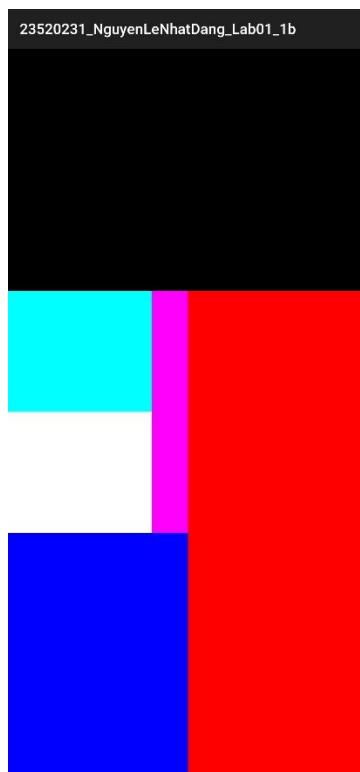
```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = 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

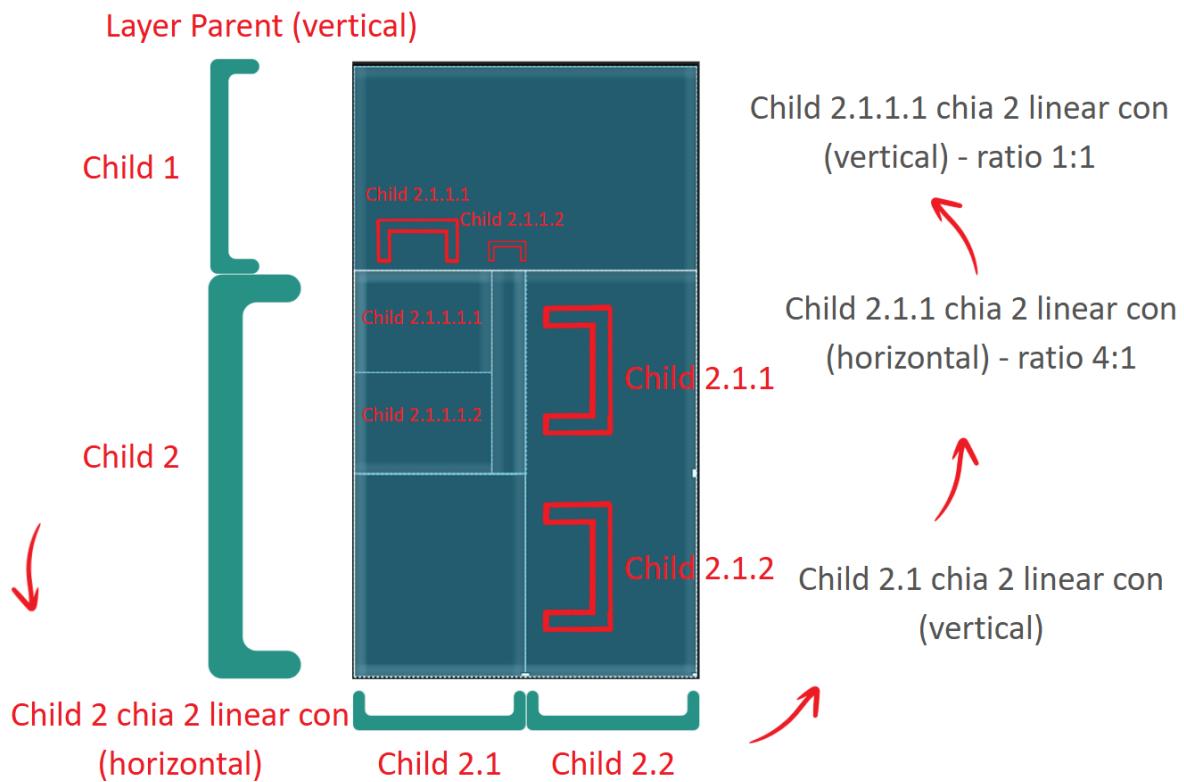


Hình 7: Kết quả sau khi thực hiện chỉnh sửa đoạn code

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:weightSum="5"
        android:background="@color/color5"
        android:orientation="horizontal">
        <LinearLayout
            android:layout_width="0dp"
            android:layout_height="match_parent"
            android:layout_weight="4"
            android:weightSum="2"
            android:orientation="vertical"
            android:background="@color/color7">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="0dp"
                android:layout_weight="1"
                android:background="@color/color8"/>
        </LinearLayout>
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"/>

```

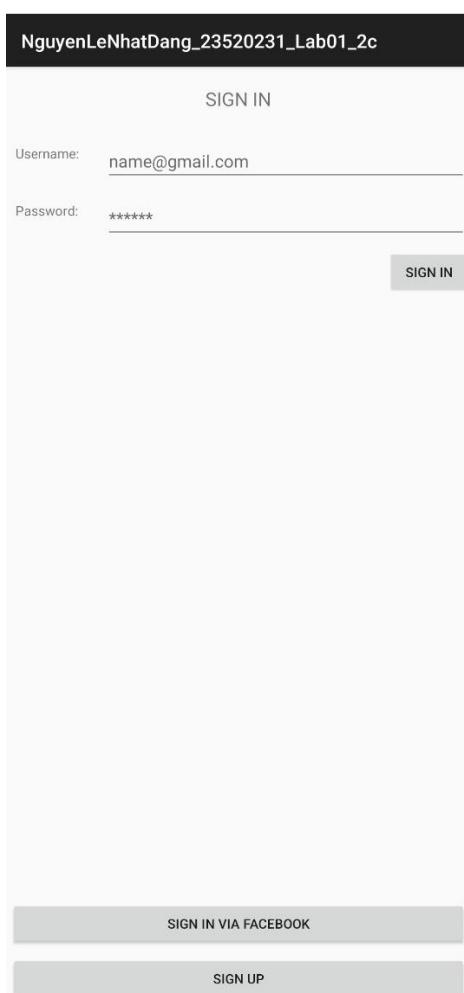
Hình 8: Đoạn code sau chỉnh sửa theo yêu cầu đề bài



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



### 2. Câu 2. RelativeLayout



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

## Bài thực hành số 01: Basic Layouts



The screenshot shows the Android Studio interface. On the left, the code editor displays the XML code for `activity_main.xml`. On the right, the design preview shows a login screen with fields for Username and Password, and buttons for SIGN IN, SIGN UP, and SIGN IN VIA FACEBOOK. Two green curly braces highlight the "SIGN UP" button and the "SIGN IN" button.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android">
    ...
    <EditText android:id="@+id/et_email" android:layout_width="wrap_content" android:layout_height="wrap_content" android:padding="@dimen/margin_base2" android:text="@string/sign_in_email" android:layout_marginTop="@dimen/margin_base" android:layout_below="@+id/et_password" android:layout_alignParentRight="true"/>
    <Button android:id="@+id/signup" android:padding="@dimen/margin_base2" android:layout_margin="@dimen/margin_base" android:layout_width="match_parent" android:layout_height="wrap_content" android:text="@string/signup_capital" android:layout_alignParentBottom="true" android:layout_centerHorizontal="true"/>
    <Button android:id="@+id/signin_via_fb" android:padding="@dimen/margin_base2" android:layout_margin="@dimen/margin_base" android:layout_width="match_parent" android:layout_height="wrap_content" android:layout_centerHorizontal="true" android:text="@string/signin_with_fb" android:layout_above="@+id/signup"/>
</RelativeLayout>
```

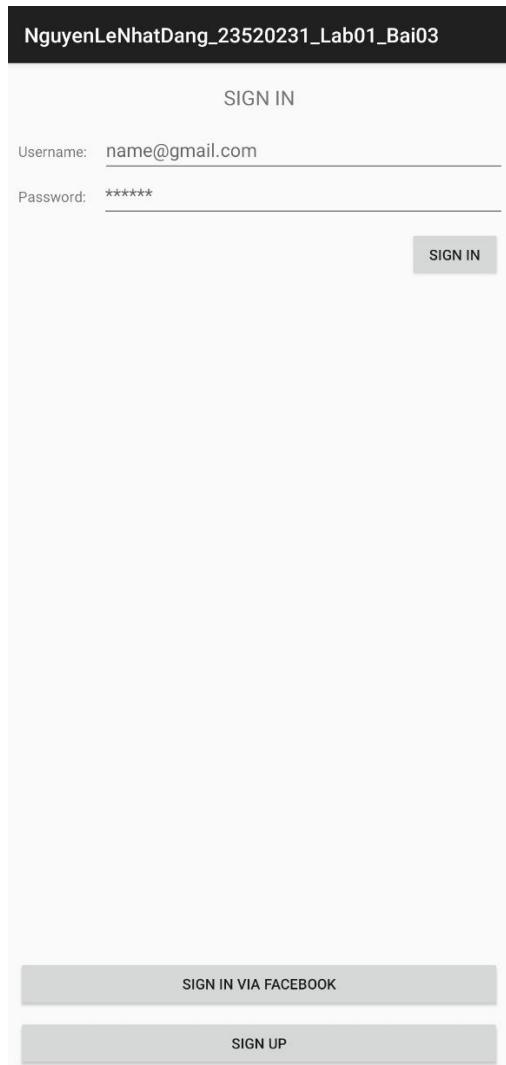
Hình 11: Nội dung đoạn code đã thực hiện

- 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



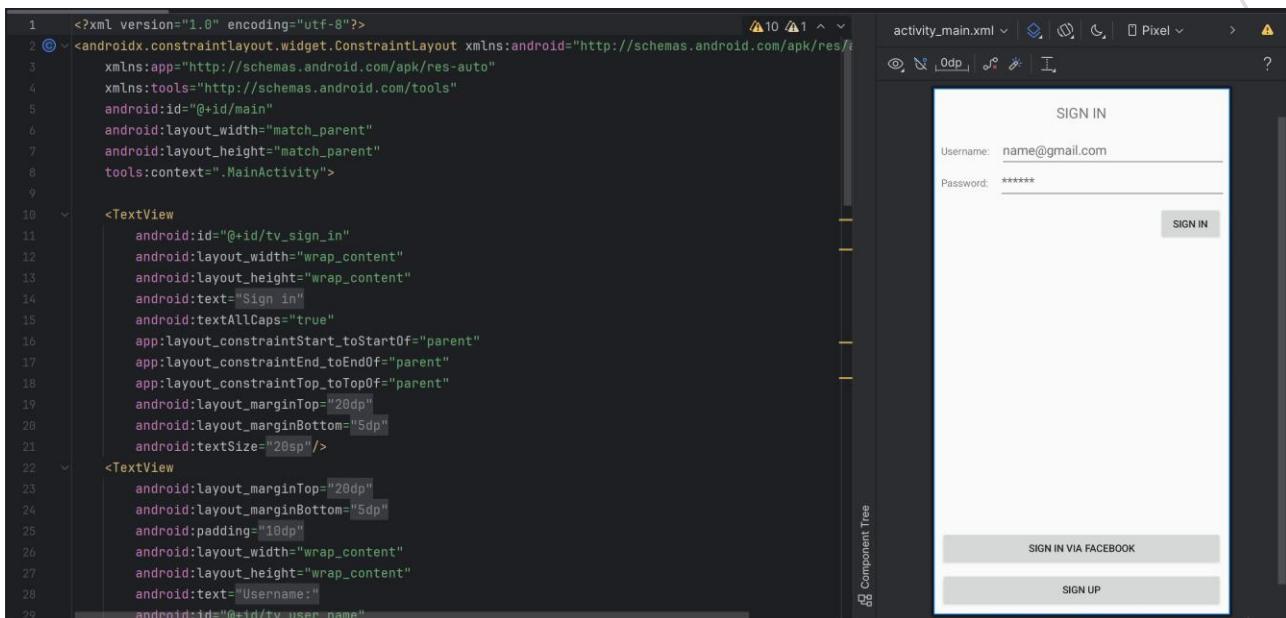
Hình 12: Kết quả sau khi thực thi đoạn code

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

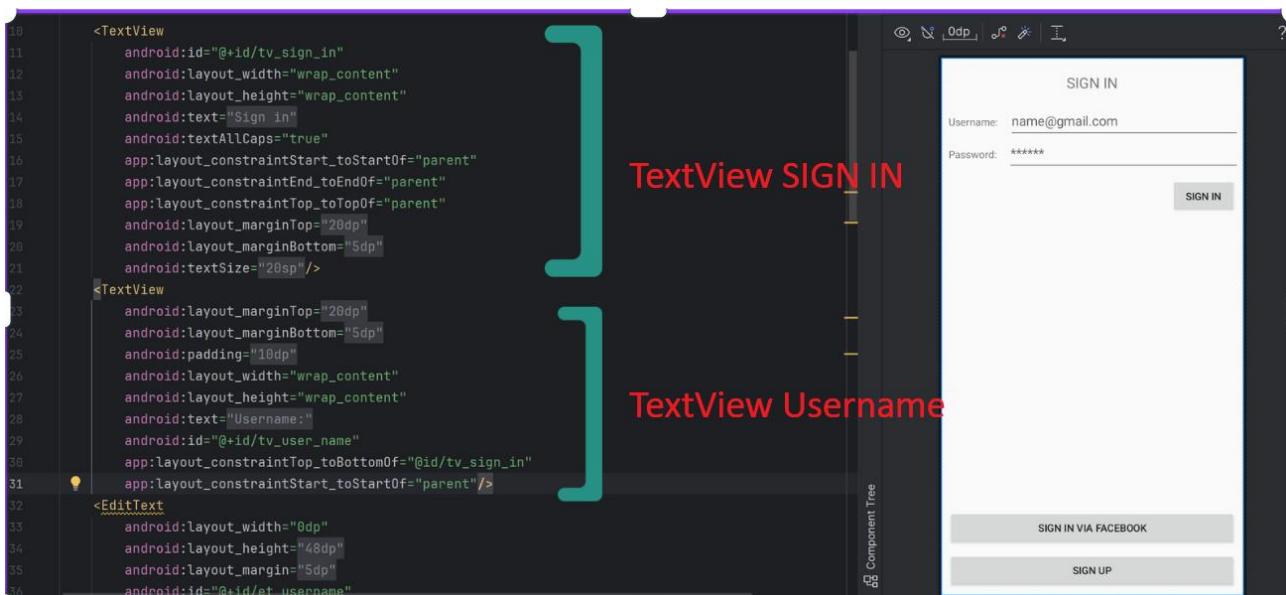
## Bài thực hành số 01: Basic Layouts



6



Hình 13: Thiết kế giao diện bằng ConstraintLayout



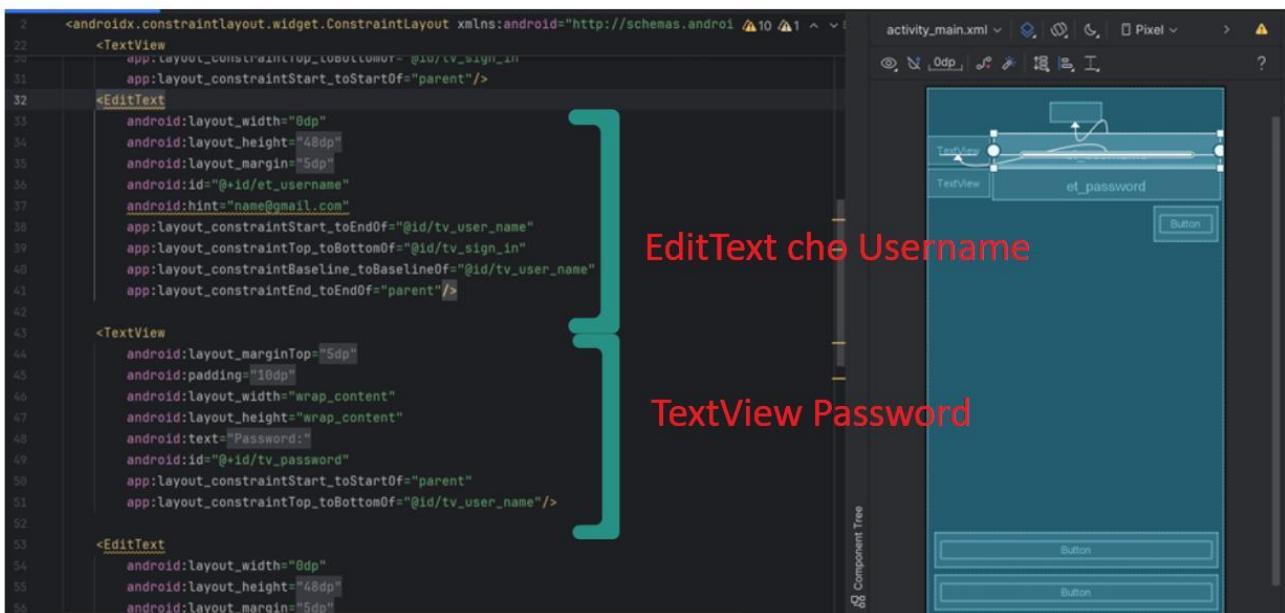
Hình 14: Thiết kế giao diện bằng ConstraintLayout (P2)

### - 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`, 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.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/tv_user_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <EditText
        android:id="@+id/et_password"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_margin="15dp"
        android:layout_toEndOf="@+id/tv_user_name"
        android:inputType="textPassword"/>
    <EditText
        android:id="@+id/tv_password"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_constraintStart_toEndOf="@+id/et_password"
        android:layout_constraintEnd_toEndOf="parent"/>
    <Button
        android:id="@+id/btn_sign_in"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <Button
        android:id="@+id/signup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>

```

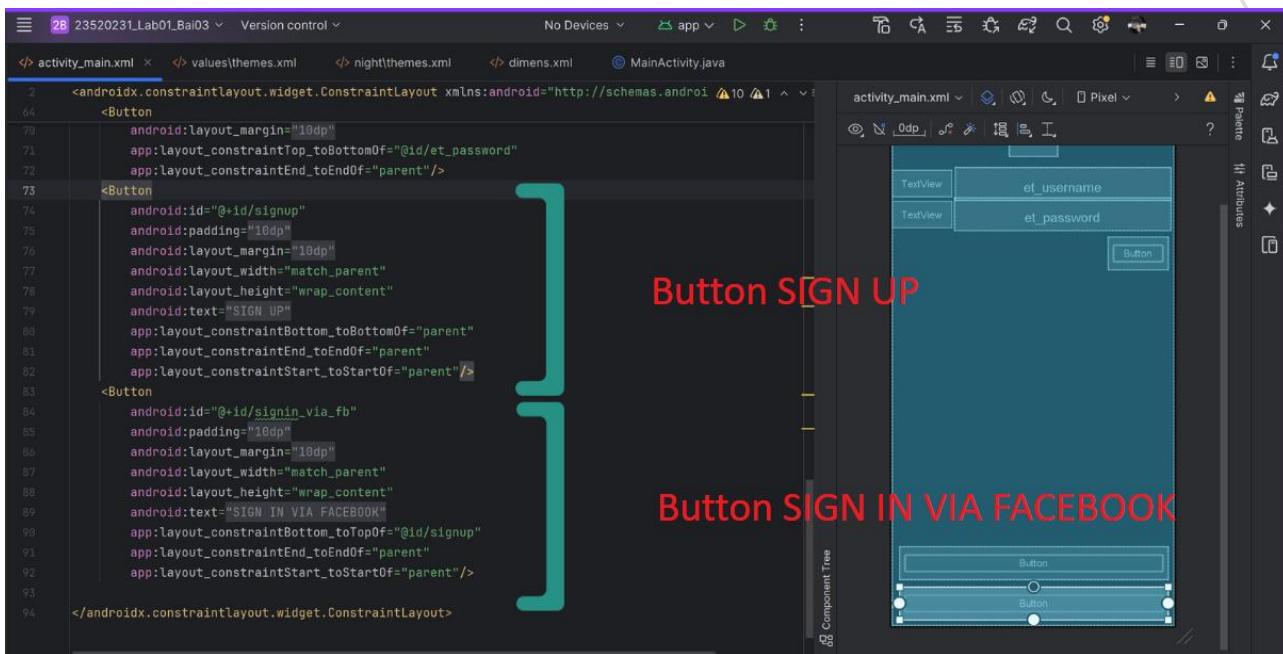
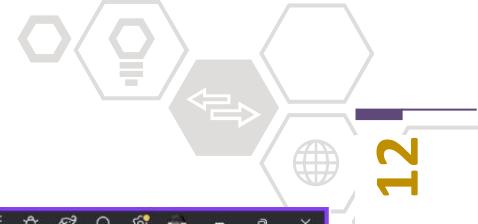
Hình 16: Thiết kế giao diện với ConstraintLayout (P4)

### - 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.