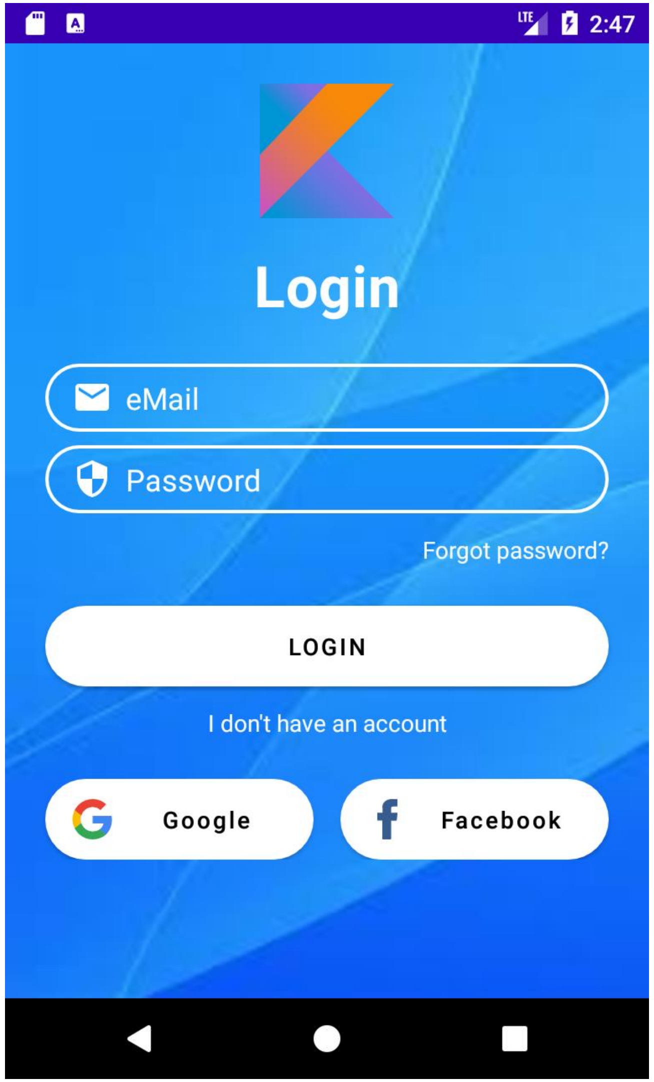
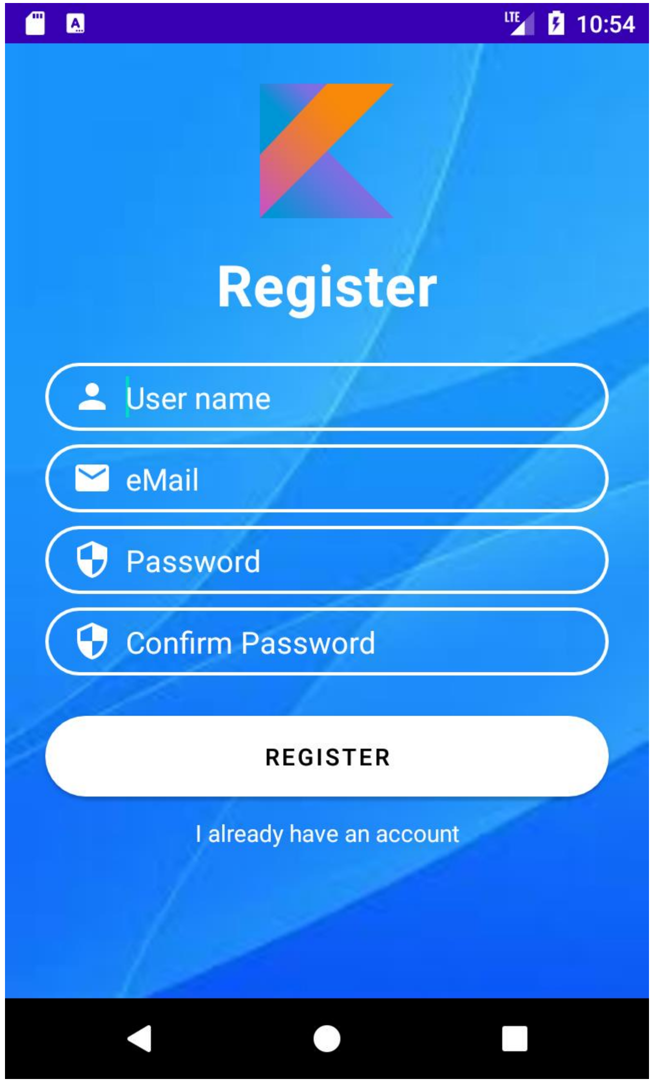
Resources

Lab

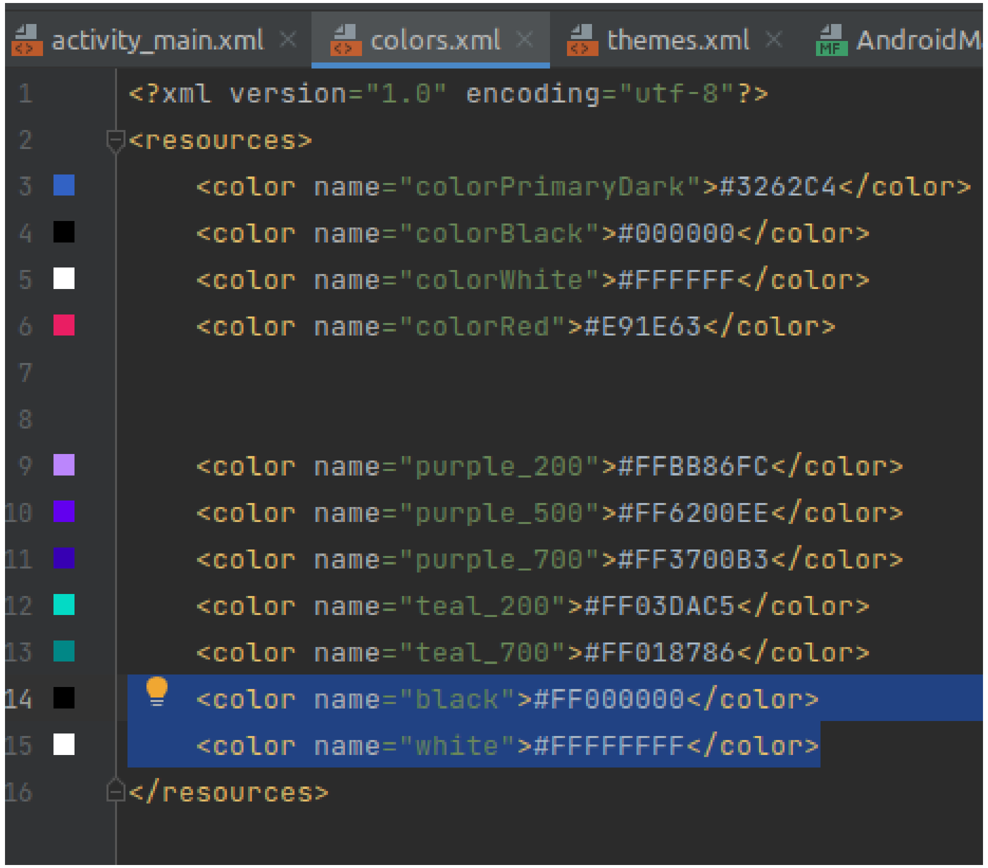
Realize a Login Activity

- Using the following resources:
  - Drawable
    - Vector Asset
  - Strings
  - Layout
  - Colors
  - Styles
  - Theme
  - Mipmap
    - Image Asset

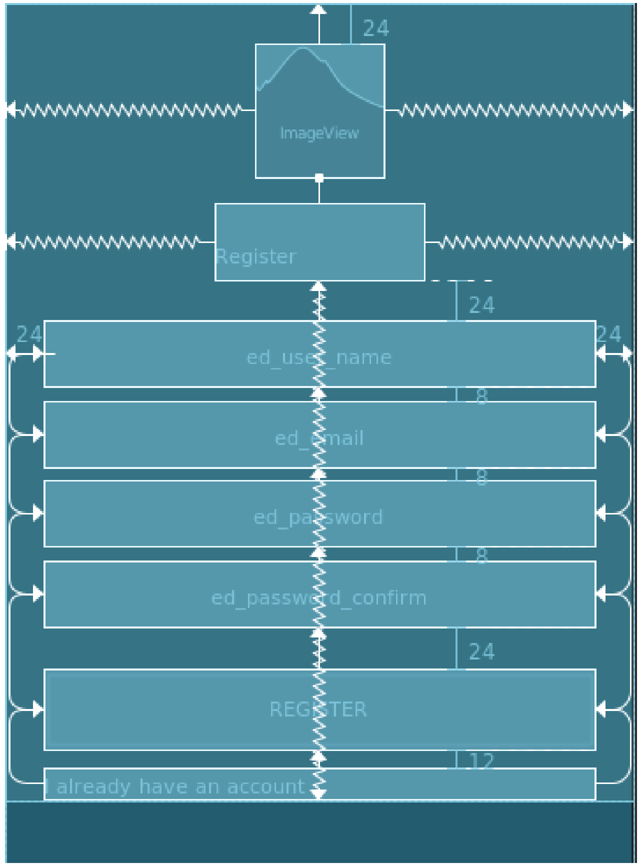


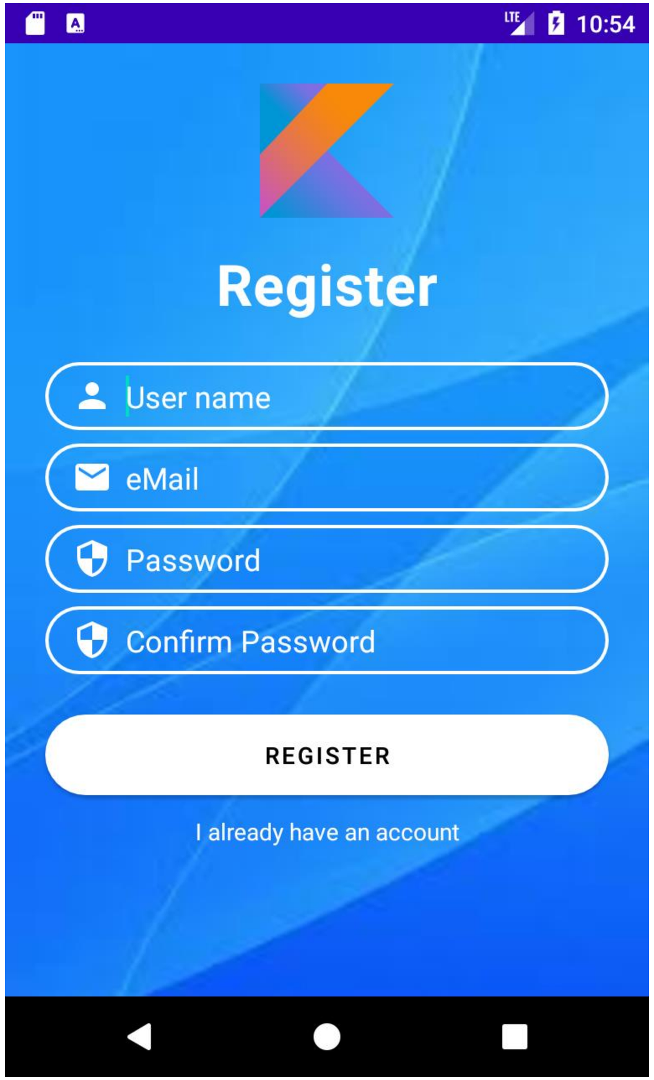


Define Layout and colors

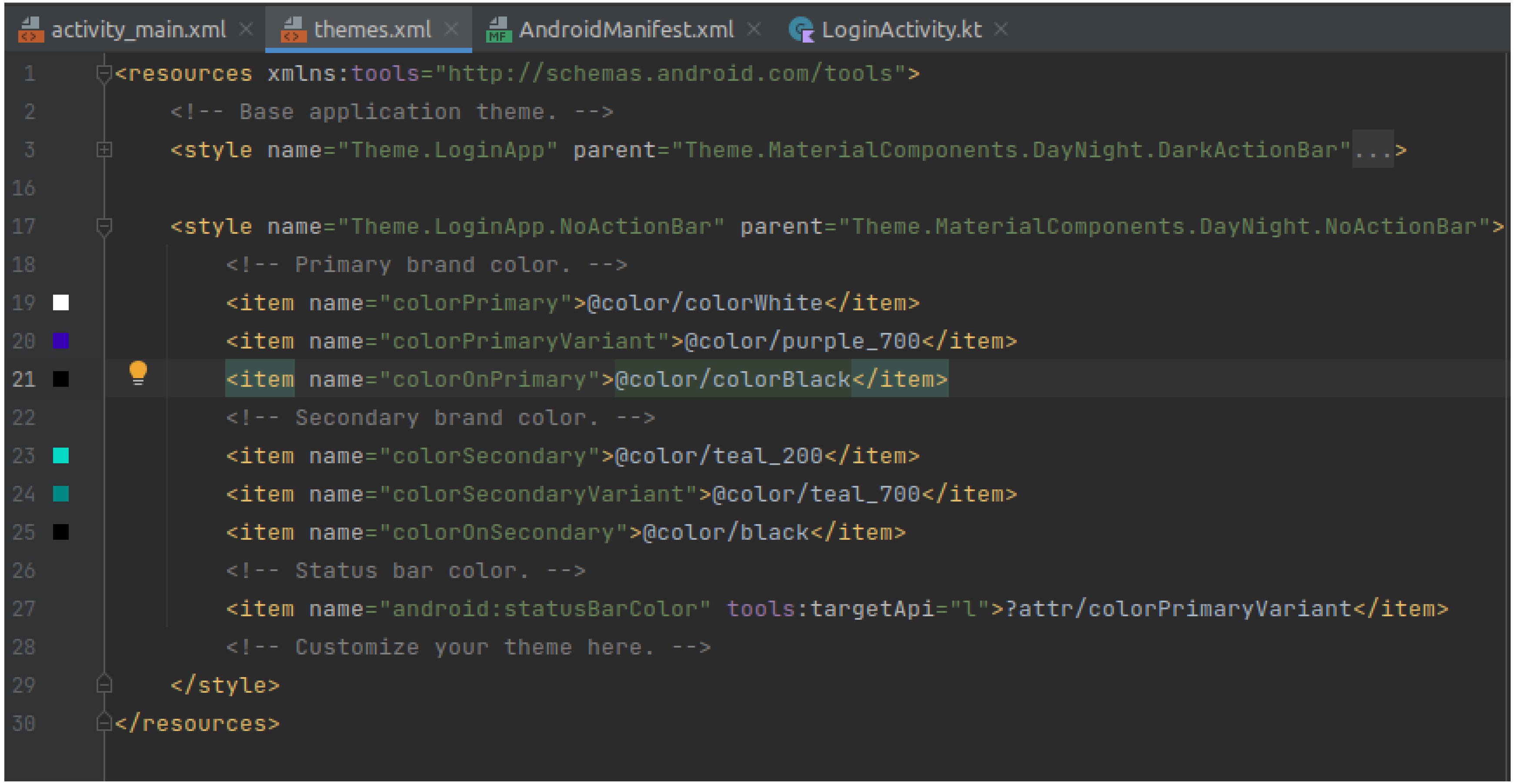


```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimaryDark">#3262C4</color>
    <color name="colorBlack">#000000</color>
    <color name="colorWhite">#FFFFFF</color>
    <color name="colorRed">#E91E63</color>


    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

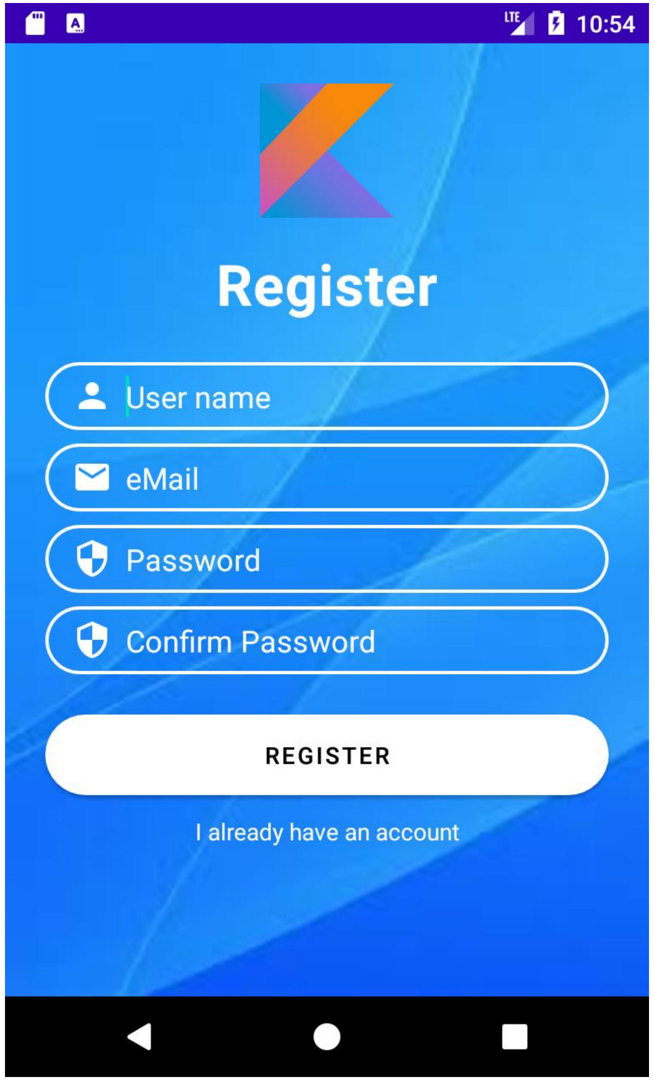




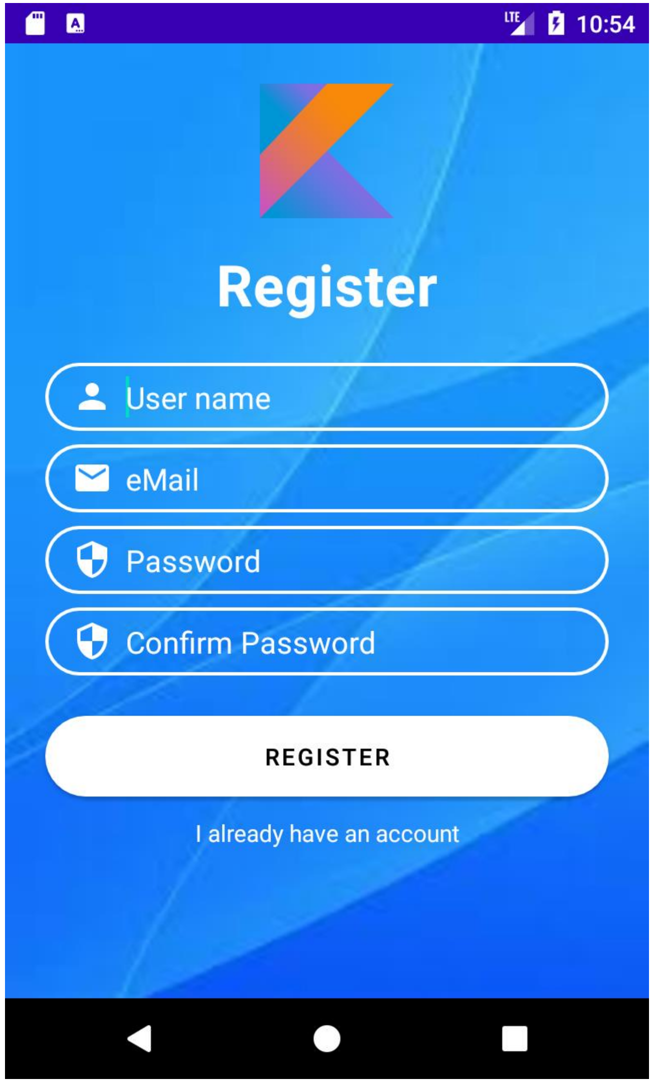
Define a style

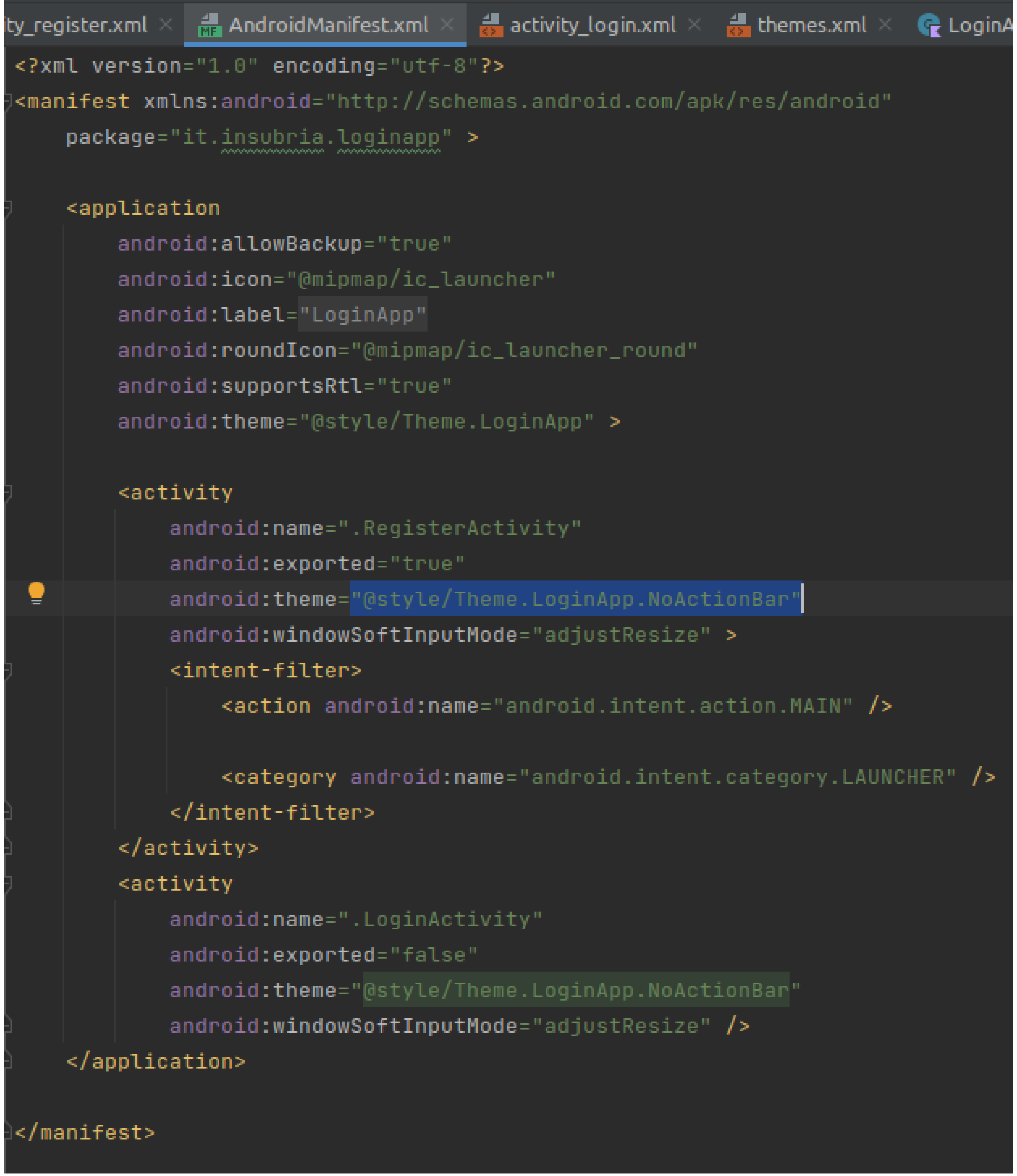


```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.LoginApp" parent="Theme.MaterialComponents.DayNight.DarkActionBar"...>

    <style name="Theme.LoginApp.NoActionBar" parent="Theme.MaterialComponents.DayNight.NoActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/colorWhite</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/colorBlack</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>
</resources>
```



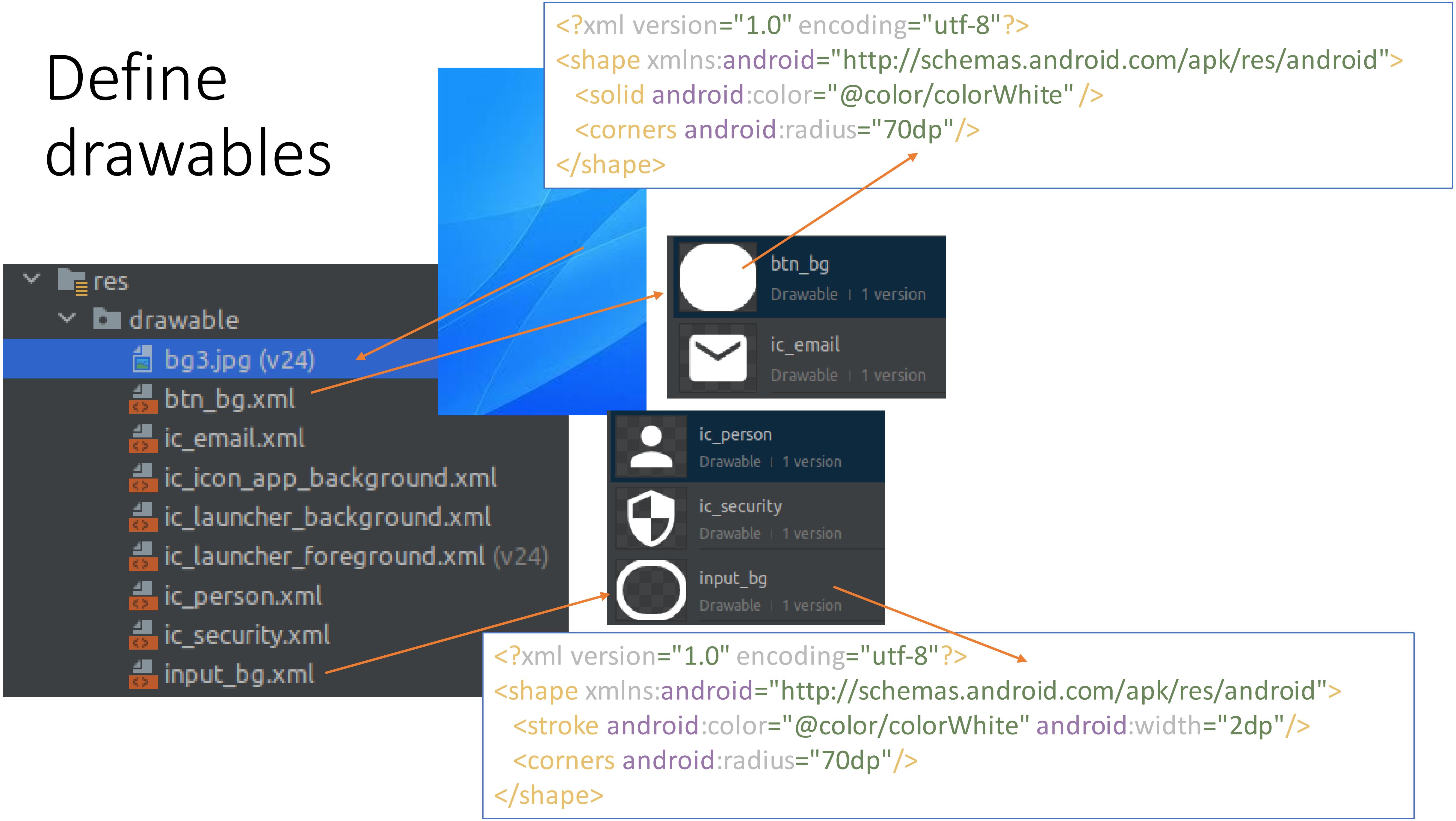
Using style





```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="it.insubria.loginapp" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="LoginApp"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.LoginApp" >

        <activity
            android:name=".RegisterActivity"
            android:exported="true"
            android:theme="@style/Theme.LoginApp.NoActionBar"
            android:windowSoftInputMode="adjustResize" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".LoginActivity"
            android:exported="false"
            android:theme="@style/Theme.LoginApp.NoActionBar"
            android:windowSoftInputMode="adjustResize" />
    </application>

</manifest>
```

Define drawables



- res
  - drawable
    - bg3.jpg (v24)
    - btn_bg.xml
    - ic_email.xml
    - ic_icon_app_background.xml
    - ic_launcher_background.xml
    - ic_launcher_foreground.xml (v24)
    - ic_person.xml
    - ic_security.xml
    - input_bg.xml

btn_bg
Drawable | 1 version

ic_email
Drawable | 1 version

ic_person
Drawable | 1 version

ic_security
Drawable | 1 version

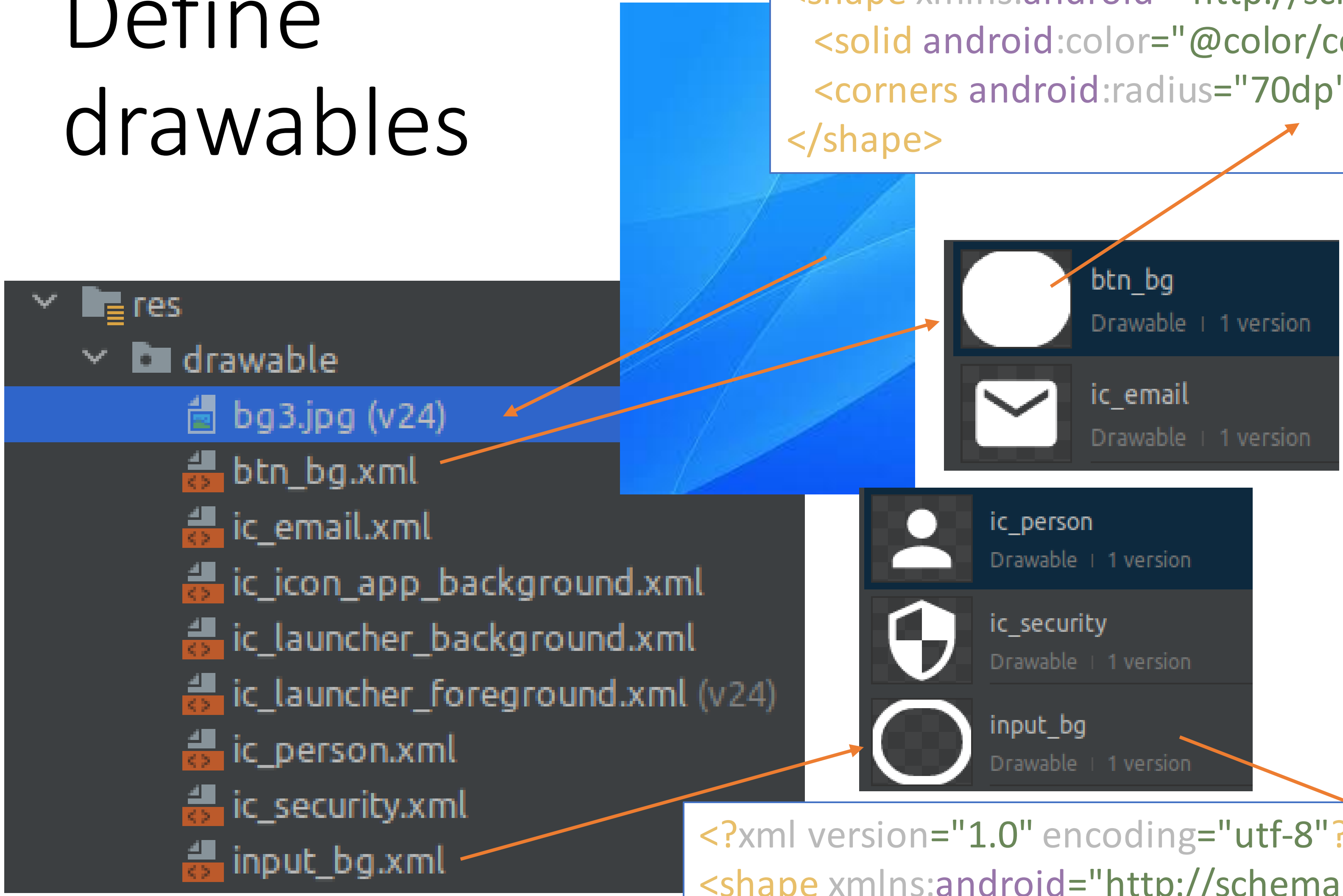
input_bg
Drawable | 1 version

```xml
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/colorWhite" />
  <corners android:radius="70dp"/>
</shape>
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <stroke android:color="@color/colorWhite" android:width="2dp"/>
  <corners android:radius="70dp"/>
</shape>
```

Define drawables

```xml
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/colorWhite" />
  <corners android:radius="70dp"/>
</shape>
```

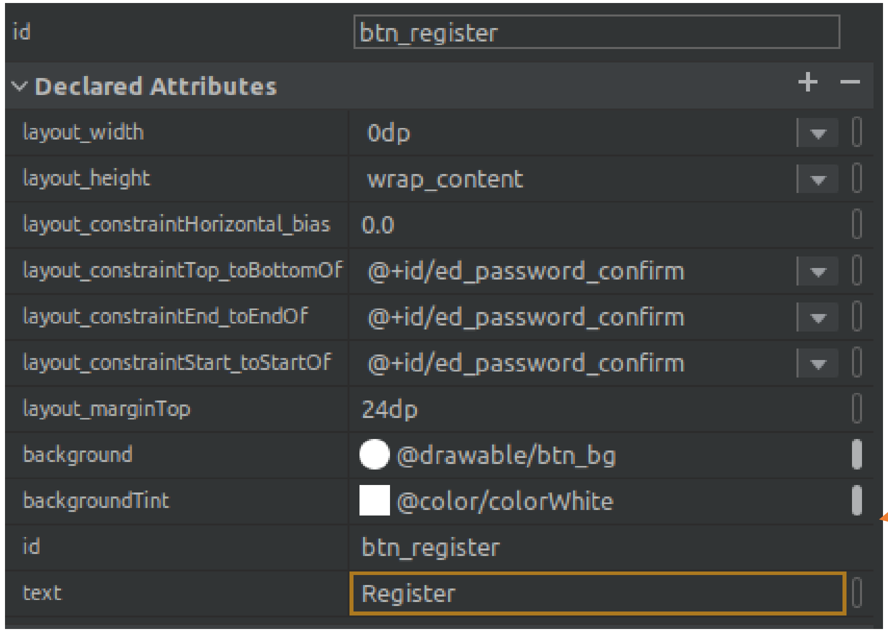


```
res
  drawable
    bg3.jpg (v24)
    btn_bg.xml
    ic_email.xml
    ic_icon_app_background.xml
    ic_launcher_background.xml
    ic_launcher_foreground.xml (v24)
    ic_person.xml
    ic_security.xml
    input_bg.xml
```

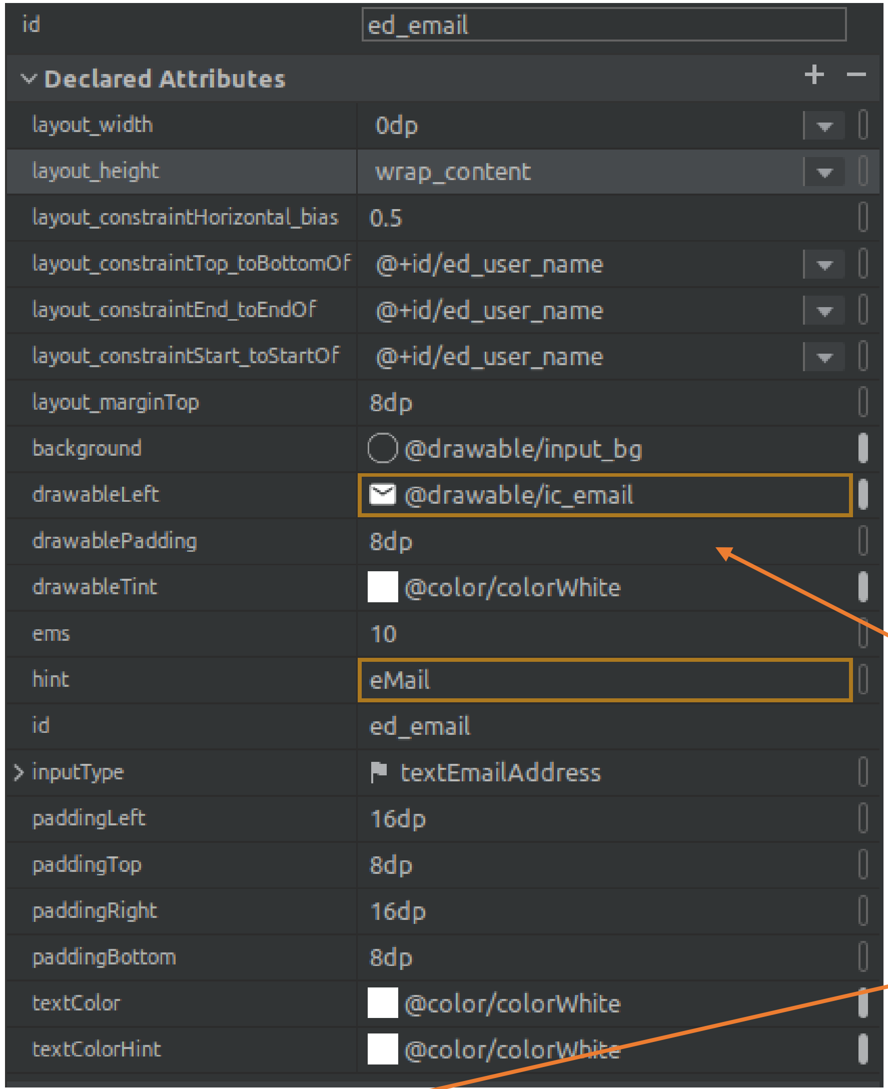
```
btn_bg
Drawable | 1 version
ic_email
Drawable | 1 version
ic_person
Drawable | 1 version
ic_security
Drawable | 1 version
input_bg
Drawable | 1 version
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <stroke android:color="@color/colorWhite" android:width="2dp"/>
  <corners android:radius="70dp"/>
</shape>
```

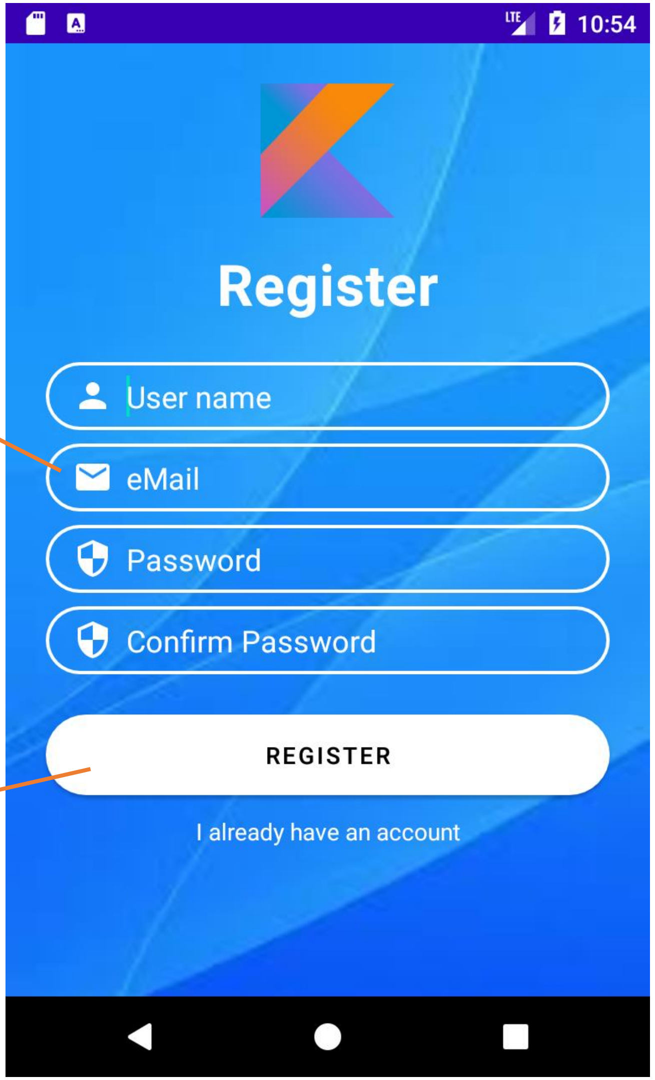
Attributes



id	btn_register
Declared Attributes	+ −
layout_width	0dp
layout_height	wrap_content
layout_constraintHorizontal_bias	0.0
layout_constraintTop_toBottomOf	@+id/ed_password_confirm
layout_constraintEnd_toEndOf	@+id/ed_password_confirm
layout_constraintStart_toStartOf	@+id/ed_password_confirm
layout_marginTop	24dp
background	○ @drawable/btn_bg
backgroundTint	□ @color/colorWhite
id	btn_register
text	Register



id	ed_email
Declared Attributes	+ −
layout_width	0dp
layout_height	wrap_content
layout_constraintHorizontal_bias	0.5
layout_constraintTop_toBottomOf	@+id/ed_user_name
layout_constraintEnd_toEndOf	@+id/ed_user_name
layout_constraintStart_toStartOf	@+id/ed_user_name
layout_marginTop	8dp
background	○ @drawable/input_bg
drawableLeft	✉ @drawable/ic_email
drawablePadding	8dp
drawableTint	□ @color/colorWhite
ems	10
hint	eMail
id	ed_email
inputType	textEmailAddress
paddingLeft	16dp
paddingTop	8dp
paddingRight	16dp
paddingBottom	8dp
textColor	□ @color/colorWhite
textColorHint	□ @color/colorWhite



Styles: override EditText style

```
<!-- Override the attribute pointing to the EditText style -->
<item name="editTextStyle">@style/app_editTextStyle</item>
```

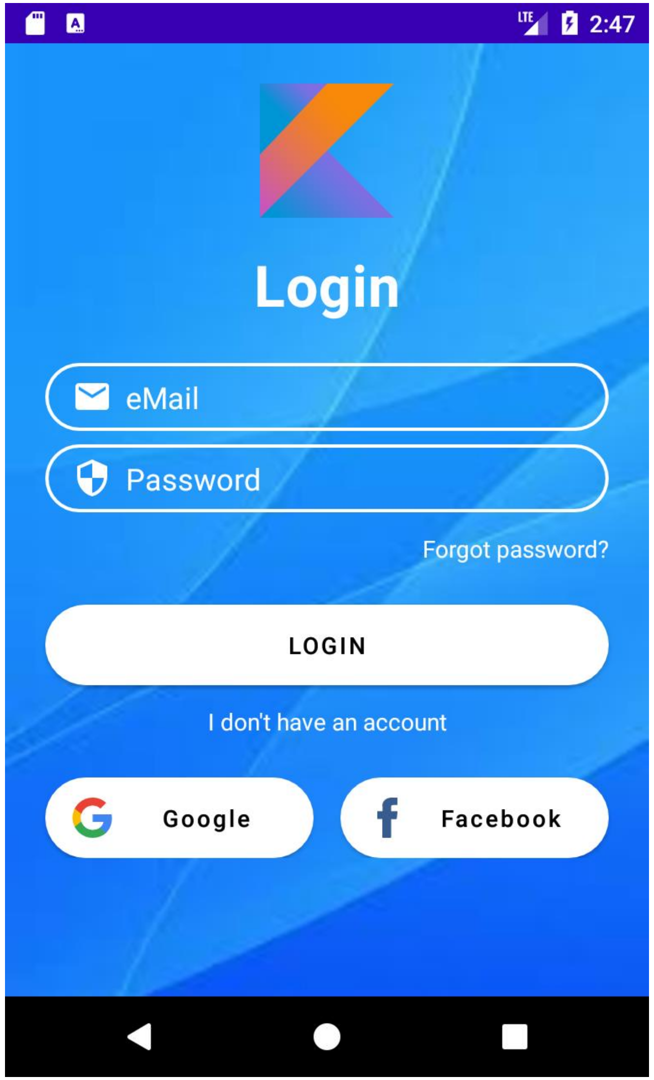
```
<!-- make your custom EditText style to extend Widget.EditText. -->
<style name="app_editTextStyle" parent="@android:style/Widget.EditText">
<item name="android:background">@drawable/input_bg</item>
<item name="android:textColor">?attr/colorPrimary</item>
<item name="android:paddingLeft">16dp</item>
<item name="android:paddingTop">8dp</item>
<item name="android:paddingRight">16dp</item>
<item name="android:paddingBottom">8dp</item>
<item name="android:drawablePadding">8dp</item>
<item name="android:drawableTint">?attr/colorPrimary</item>
<item name="android:textColorHint">?attr/colorPrimary</item>
</style>
```

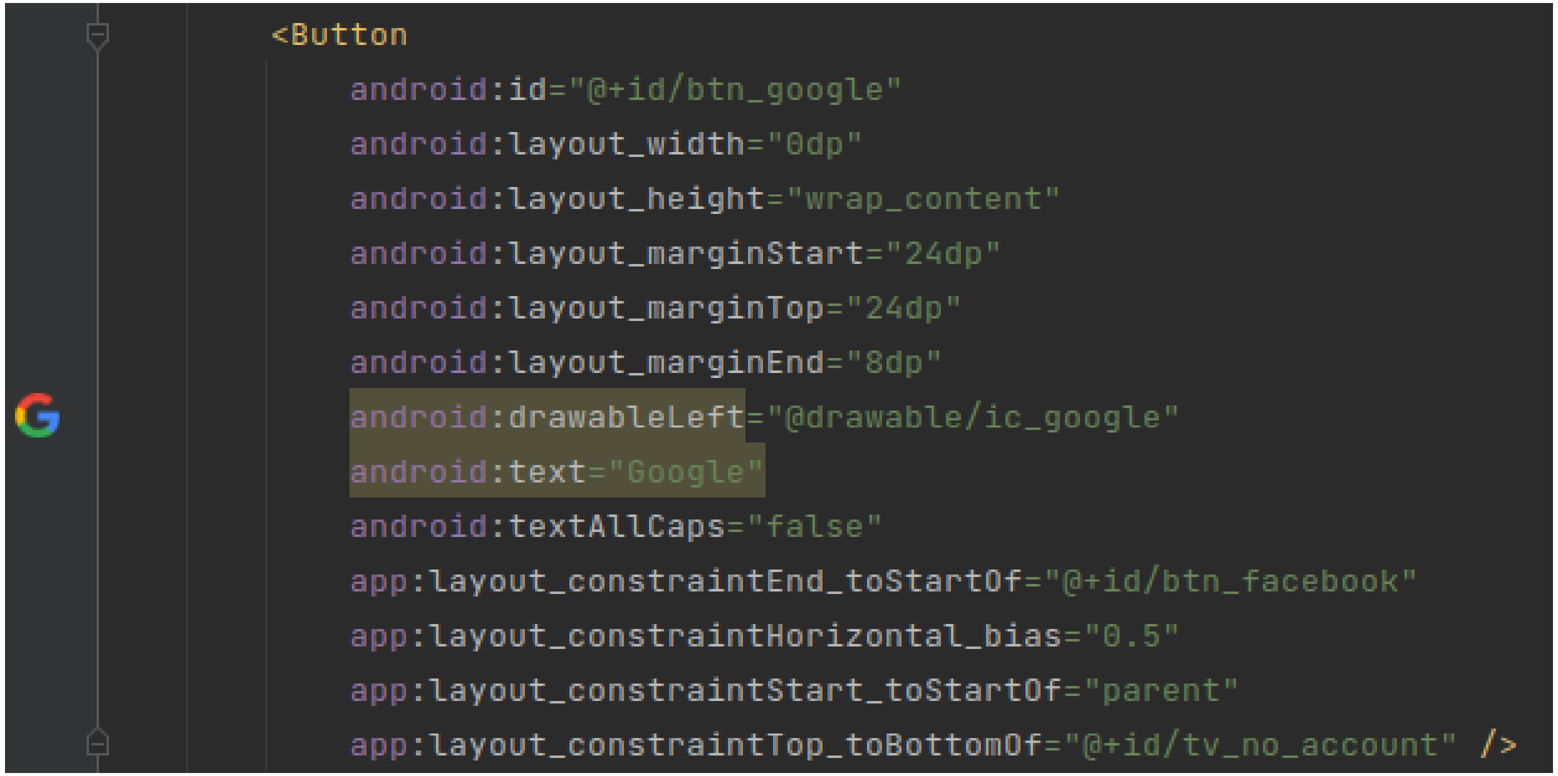
Styles: override MaterialButton style

```
<!-- Override the attribute pointing to the MaterialButton style -->
<item name="materialButtonStyle">@style/app_ButtonStyle</item>
```

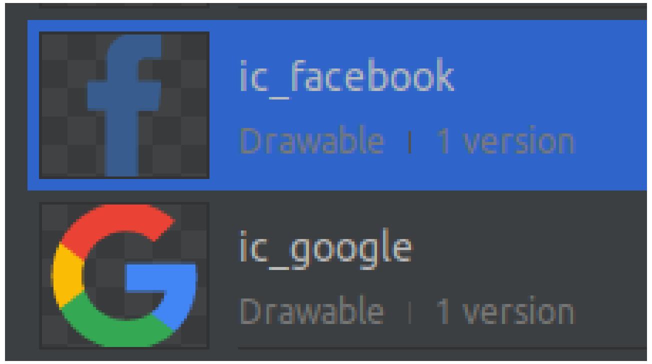
```
<!-- make your custom Button style to extend Widget.MaterialComponents.Button. -->
<style name="app_ButtonStyle" parent="Widget.MaterialComponents.Button">
  <item name="android:background">@drawable/btn_bg</item>
  <item name="android:textColor">?attr/colorOnPrimary</item>
  <item name="android:backgroundTint">?attr/colorPrimary</item>
</style>
```

Login Activity





```
<Button
    android:id="@+id/btn_google"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="24dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="8dp"
    android:drawableLeft="@drawable/ic_google"
    android:text="Google"
    android:textAllCaps="false"
    app:layout_constraintEnd_toStartOf="@+id/btn_facebook"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tv_no_account" />
```



onClick

```
// validating email id
private fun isValidEmail(email: String): Boolean {
  val EMAIL_PATTERN = ("^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-]+)*@"
                  + "[A-Za-z0-9-]+(\\.[A-Za-z0-9]+)*(\\.[A-Za-z]{2,})$")
  val pattern = Pattern.compile(EMAIL_PATTERN)
  val matcher = pattern.matcher(email)
  return matcher.matches()
}
```

```
fun checkLogin(v: View?) {
  val email: String = editTextUserName.getText().toString()
  if (!isValidEmail(email)) {
    //Set error message for email field
    editTextUserName.setError(getString(R.string.invalid_email))
  }
  val pass: String = editTextPassword.getText().toString()
  if (!isValidPassword(pass)) {
    //Set error message for password field
    editTextPassword.setError(getString(R.string.invalid_password))
  }
  if (isValidEmail(email) && isValidPassword(pass)) {
    // Validation Completed
  }
}
```

```
// validating password
private fun isValidPassword(pass: String?): Boolean {
  return if (pass != null && pass.length >= 4) {
    true
  } else false
}
```