Lab 2: **Android Form Validation**

In any app input validation is a very important thing to do. Input Validation eliminates the errors that can be done by user while giving inputs to our app. For example, if we want to get the user’s email we can check the entered email is a valid email or not before storing it inside the database.

**Experiment**: Develop a Mobile App to get Customer’s data with basic fields USER NAME (First, middle and Last Name), Email ID (Primary, Alternate) Mobile Phone Number (Primary, alternate of Work / Residence/Friend) and Address (Residence, Work). Explain the data structure (field name, field type and field width) used here. Do validation check to take proper inputs for these fields.

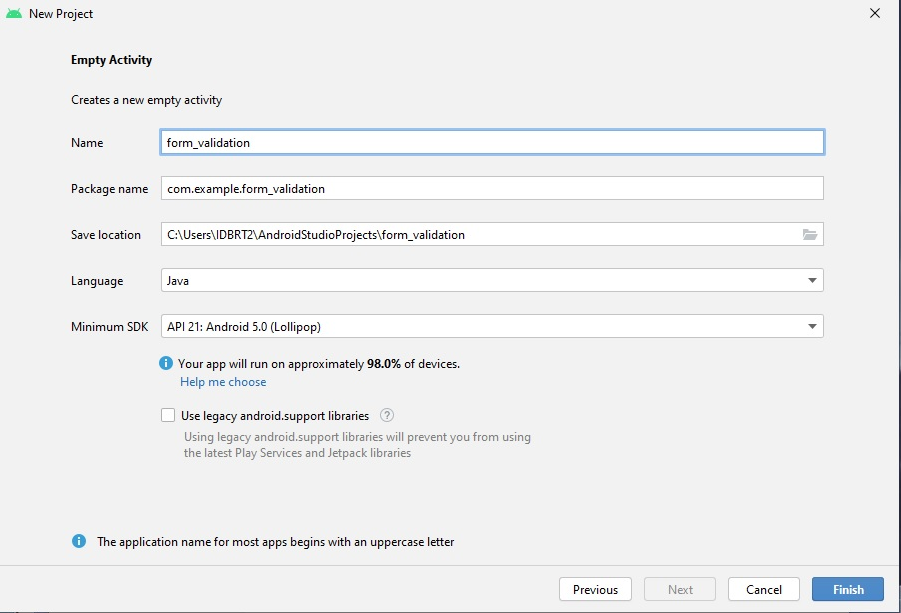
**Solution**:

Four steps:

1. Project creation
2. Creating a Form (.XML file)
3. Android Form Validation
4. Validate the input (MainActivity.java).
5. Testing the App.

**Step 1: Project Creation**:

Open Android Studio and create a new project with an empty activity.

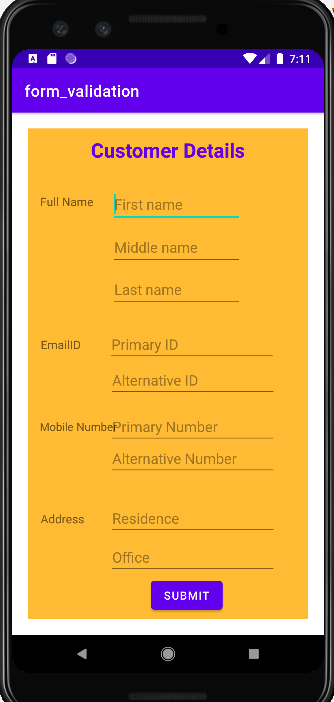


Wait for Android Studio to finish creating your project.

**Step 2: Creating a Form**:

Open app > res > layout > activity\_main.xml. This file defines the layout for the user interface (UI). A UI in Android is defined in XML files.

In this project, we will use the ConstraintLayout. A ConstraintLayout is a layout that provides a flexible way for creating views. For our project, I have designed the screen given below.



In the above screen we have 5 TextViews, 9 EditTexts and 1 Button. Now we will validate these fields for the connect input when the button is pressed.

1. Add a Title to the Layout

In the Palette panel, click Text and add it to the design. You might need to move it around so that it stays at the top of the layout. This TextView control will display the form's description and purpose to the user.

The XML for the TextView should look like this:

<TextView  
 android:id="@+id/customer\_Details"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Customer Details"  
 android:textSize="25dp"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:padding="10dp"  
 android:textColor="@color/purple\_500"  
 />

Notice the **id** attribute on the text view; it is recommended that each control element is assigned an **id** so that it's easy to reference from the Java files.

1. Add a Full Name field

Use ViewText element to label the text field.

The XML for the TextView should look like this:

<TextView  
 android:id="@+id/User\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Full Name"  
 android:textSize="15dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.054"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.137" />

For the “Firstname“field, an EditText element is used to enter and modify text. Add an EditText control for the name just below the TextView. You must also specify the input type attribute. In this case, we will use android:inputType="text" for plain text. Here is the full XML.

<EditText  
 android:id="@+id/first\_name"  
 android:layout\_width="165dp"  
 android:layout\_height="47dp"  
 android:hint="First name"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.558"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.127" />

1. Add a Phone Number Field

This EditText will be used to fill in the phone number. The attribute for this element will be android:inputType="phone". Setting the EditText to inputType to phone is helpful since it will be easy to add digits. When the user touches this field, they will be presented with the screen below.



Here is the full XML.

<EditText  
 android:id="@+id/Primary\_Number"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Primary Number"  
 android:inputType="phone"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.72"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.617" />

1. Add an Address Field to the Layout

Add another EditText below the phone field. This EditText will be used for the address of the user. You might also want to increase the space provided for the address field. The attribute for this element will be android:inputType=" textPostalAddress ".

<EditText  
 android:id="@+id/Residence"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Residence"  
 android:inputType="textPostalAddress"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.72"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.818" />

1. Add a Button to the layout

Lastly, add the submit button. You can change the background color of the button to fit your brand's identity. The most important attribute of the button is the onClick attribute.

<Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Submit"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.59"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.99" />

1. Add email field to the layout

Add another EditText below the phone field. This EditText will be used for the email of the user. The attribute for this element will be android:inputType=" textEmailAddress ".

<EditText  
 android:id="@+id/primaryid"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Primary ID"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.714"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.436" />

The complete code of .xml file

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:layout\_margin="20dp"  
 android:background="@android:color/holo\_orange\_light"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/customer\_Details"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Customer Details"  
 android:textSize="25dp"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:padding="10dp"  
 android:textColor="@color/purple\_500"  
 />  
 <TextView  
 android:id="@+id/User\_name"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Full Name"  
 android:textSize="15dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.054"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.137" />  
 <EditText  
 android:id="@+id/first\_name"  
 android:layout\_width="165dp"  
 android:layout\_height="47dp"  
 android:hint="First name"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.558"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.127" />  
 <EditText  
 android:id="@+id/Middle\_name"  
 android:layout\_width="165dp"  
 android:layout\_height="47dp"  
 android:hint="Middle name"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.558"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.222" />  
 <EditText  
 android:id="@+id/Last\_name"  
 android:layout\_width="165dp"  
 android:layout\_height="47dp"  
 android:hint="Last name"  
 android:inputType="textPersonName"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.558"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.315" />  
 <TextView  
 android:id="@+id/EmailID"  
 android:layout\_width="74dp"  
 android:layout\_height="32dp"  
 android:text="EmailID"  
 android:textSize="15dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.057"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.447" />  
 <EditText  
 android:id="@+id/primaryid"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Primary ID"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.714"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.436" />  
 <EditText  
 android:id="@+id/AlterID"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Alternative ID"  
 android:inputType="textEmailAddress"  
 app:layout\_constraintBottom\_toBottomOf="@+id/button"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.72"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.52" />  
 <TextView  
 android:id="@+id/MNumber"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Mobile Number"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.058"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.611" />  
  
 <EditText  
 android:id="@+id/Primary\_Number"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Primary Number"  
 android:inputType="phone"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.72"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.617" />  
  
 <EditText  
 android:id="@+id/Alternative\_Number"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Alternative Number"  
 android:inputType="phone"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.72"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.687" />  
  
 <EditText  
 android:id="@+id/Residence"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Residence"  
 android:inputType="textPostalAddress"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.72"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.818" />  
  
 <TextView  
 android:id="@+id/Address"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Address"  
 android:textSize="15dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.054"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.806" />  
  
 <EditText  
 android:id="@+id/office"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:inputType="textPostalAddress"  
 android:hint="Office"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.72"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.904" />  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Submit"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.59"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="@+id/customer\_Details"  
 app:layout\_constraintVertical\_bias="0.99" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Step 3: Android Form Validation

* Adding **AwesomeValidation** Library

So we are going to use AwesomeValidation. So first come inside app level **build.gradle** file and add the following line inside dependencies block.

implementation 'com.basgeekball:awesome-validation:4.2'  
implementation group: 'com.google.guava', name: 'guava', version: '27.0.1-android'

* Now after adding the above line **sync** your project.

Step 4: MainActivity.java.

What we need to do?

* First we define an AwesomeValidation object.
* We initialize the object with BASIC style, we have two more different style COLORATION and UNDERLABEL.
* Then we add the validation to required EditTexts using addValidation() method with AwesomeValidation object.
* addValidation(): This method takes 4 arguments.

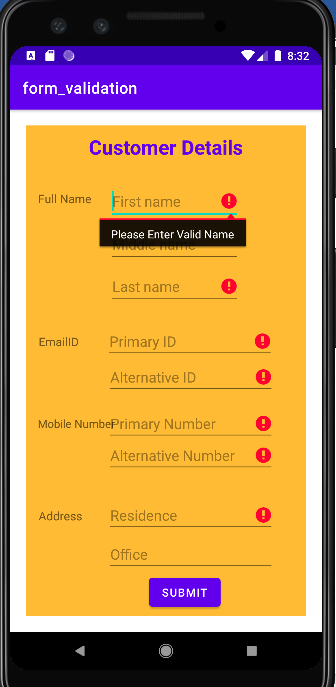
1. Activity Context, we used this to pass the current activity context.
2. Resource id of EditText in which we want to add validation, we used R.id.viewid
3. Regex String for adding validation, you can use Regex for more kind of validations
4. String resource id for error message to show the error if validation fails

* Then in setOnClickListener () method we are validating the inputs with validate() method. The method returns true or false, true if validation succeeds and false if validation fails. To validate the input, the modify the code as below:

package com.example.form\_validation;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.util.Patterns;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.basgeekball.awesomevalidation.AwesomeValidation;  
import com.basgeekball.awesomevalidation.ValidationStyle;  
import com.basgeekball.awesomevalidation.utility.RegexTemplate;  
  
public class MainActivity extends AppCompatActivity {  
 *//Initialize Variables* EditText efName,emName,elName,epEmail,eaEmail,epMobile,eaMobile,erAddress,ewAddress;  
 Button bsub;  
 AwesomeValidation awesomeValidation;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 *//Assign variables* efName=findViewById(R.id.*first\_name*);  
 emName=findViewById(R.id.*Middle\_name*);  
 elName=findViewById(R.id.*Last\_name*);  
 epEmail=findViewById(R.id.*primaryid*);  
 eaEmail=findViewById(R.id.*AlterID*);  
 epMobile=findViewById(R.id.*Primary\_Number*);  
 eaMobile=findViewById(R.id.*Alternative\_Number*);  
 erAddress=findViewById(R.id.*Residence*);  
 ewAddress=findViewById(R.id.*office*);  
 bsub=findViewById(R.id.*button*);  
 *//Initialize validation Style* awesomeValidation =new AwesomeValidation(ValidationStyle.*BASIC*);  
 *//Add Validation For Name* awesomeValidation.addValidation(this,R.id.*first\_name*, RegexTemplate.*NOT\_EMPTY*,R.string.*invalid\_name*);  
 awesomeValidation.addValidation(this,R.id.*Last\_name*, RegexTemplate.*NOT\_EMPTY*,R.string.*invalid\_lname*);  
  
 *//For Mobile Number* awesomeValidation.addValidation(this,R.id.*Primary\_Number*,"[5-9]{1}[0-9]{9}$",R.string.*invalid\_mobile*);  
 awesomeValidation.addValidation(this,R.id.*Alternative\_Number*,"[5-9]{1}[0-9]{9}$",R.string.*invalid\_mobile*);  
 *//For Email* awesomeValidation.addValidation(this,R.id.*primaryid*, Patterns.*EMAIL\_ADDRESS*,R.string.*invalid\_email*);  
 awesomeValidation.addValidation(this,R.id.*AlterID*, Patterns.*EMAIL\_ADDRESS*,R.string.*invalid\_aemail*);  
 *//For Address* awesomeValidation.addValidation(this,R.id.*Residence*, RegexTemplate.*NOT\_EMPTY*,R.string.*invalid\_aname*);  
 *//awesomeValidation.addValidation(this,R.id.office, RegexTemplate.NOT\_EMPTY,R.string.invalid\_oname);* bsub.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 *//Check Validation* if (awesomeValidation.validate()){  
 *//On Success* Toast.*makeText*(getApplicationContext()  
 , "Form Validation Successfully...",Toast.*LENGTH\_SHORT*).show();  
 }else{  
 Toast.*makeText*(getApplicationContext(),"Validation Faild.",Toast.*LENGTH\_SHORT*).show(); } }  
 }); }}

**Step 5: Testing the App.**

Now you can run the app to test whether it is working or not.



* You can see in the above screenshot it is working absolutely fine.