NANYANG POLYTECHNIC

School of Information Technology

Course:	Diploma in Information Technology
Module:	IT2161 Mobile Application Development

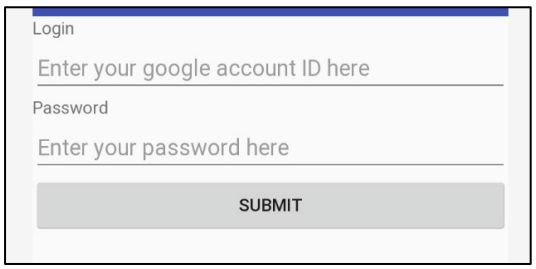
Practical 3.1: Layouts & Controls

Objectives:

- To learn how to use layouts to place widgets in an Android application
- To learn how to create basic widgets in Android application.
- To learn how to control widgets in Android application.
- To learn how to implement event listener for widgets.

Exercise 1: Layouts and Controls [Basic]

In this exercise, you will create the following layout and set a button click listener.



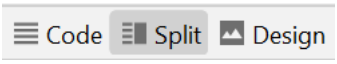
Steps

1. Start Android Studio and create a new Android project called "LayoutsAndControls".

activity_main.xml

2. Open **activity_main.xml** file in the layout folder.

3. Switch from **Design** to **Split** view by using the tab.



4. You should see the codes below.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

5. Update the base layout by changing "android.support.constraint.ConstraintLayout" to "Linearlayout"

6. Set the orientation to "vertical"

7. Remove the "TextView".

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity_Basic"
    android:orientation="vertical"
    >



</LinearLayout>
```

8. Add in the following TextViews, EditText and Buttons into the LinearLayout that you created on top.

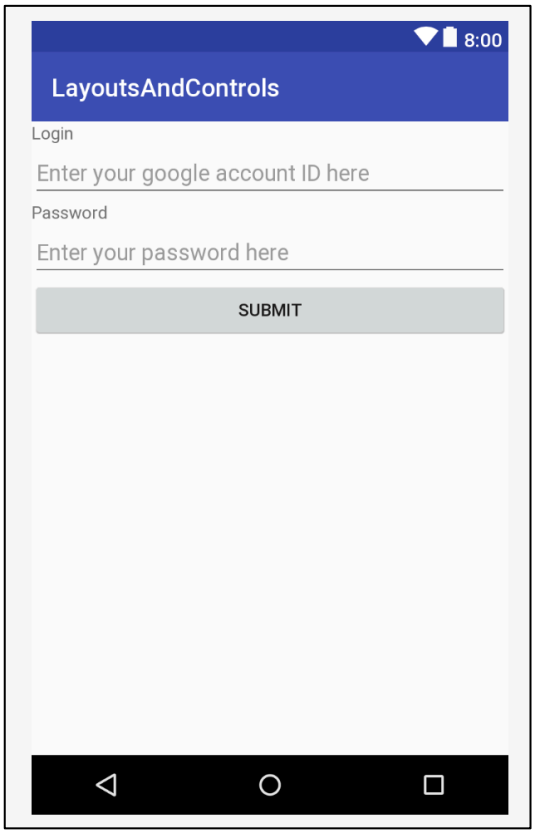
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Login" />

<EditText
    android:id="@+id/loginET"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Enter your google account ID here"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Password" />

<EditText
    android:id="@+id/passwordET"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="Enter your password here"
    android:inputType="textPassword"
    />

<Button
    android:id="@+id/btnSubmit"
    android:text="Submit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

9. Update the width of “loginET” and “passwordET” to match_parent.

```
<EditText
    android:id="@+id/loginET"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your google account ID here"/>
```

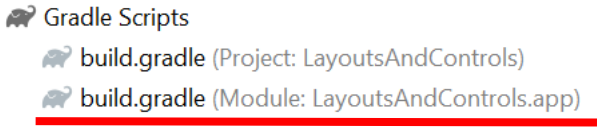
```
<EditText
    android:id="@+id/passwordET"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your password here"
    android:inputType="textPassword"
    />
```

10. You should see the following layout in the preview panel or when try to run the application in the emulator.



build.gradle (app)

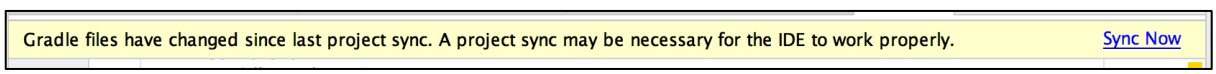
11. Open **build.gradle** (LayoutsAndControls.**app**).



12. Add the following plugin

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-android-extensions'
}
```

13. Remember to sync the Gradle files before progressing on.



MainActivity.kt

14. Open **MainActivity** activity. Add the following import.

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import kotlinx.android.synthetic.main.activity_main.*
```

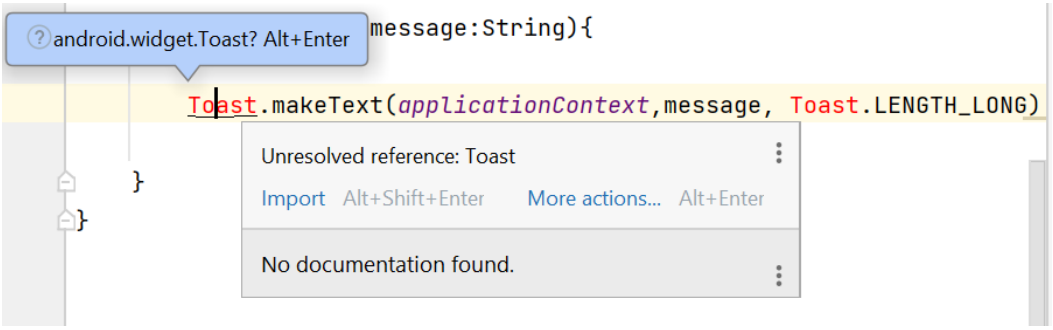
15. In **MainActivty**, create the method “displayToast” after onCreate.

```
class MainActivity_Basic : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main_basic)


    }

    fun displayToast(message:String){

        Toast.makeText( context: this,message,Toast.LENGTH_LONG).show()
    }
}
```

As you type **Toast**, Android Studio should auto add the import for you.

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Toast
import kotlinx.android.synthetic.main.activity_main.*
```

If it doesn't you can hover on **Toast** and click **Import** or use keyboard shortcut **Alt+Shift+Enter**.



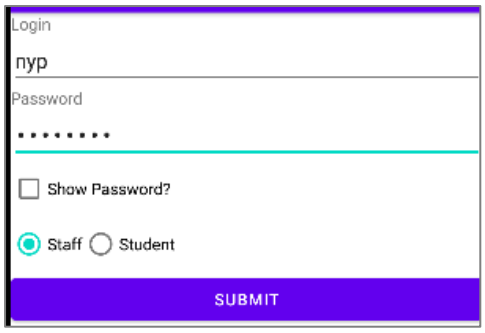
16. Inside onCreate, add in the onClickListener for btnSubmit.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main_basic)

    btnSubmit.setOnClickListener { it: View!


        if(passwordET.text.toString() == "password")
        {

            displayToast("Success")

        }
        else{

            displayToast("Error")

        }

    }
}
```

17. Run your app.
    Experiment with the correct/wrong password and click **Submit** button.

Exercise 2: Layouts and Controls [Intermediate]

In this exercise, you will set listener for checkbox and radio group.



Steps

activity_main.xml

18. Open the layout file and add in the following Checkbox and Radio buttons.

    Note : They are located between “passwordET” and “btnSubmit”.

```
    <EditText
        android:id="@+id/loginET"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your google account ID here"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password" />

    <EditText
        android:id="@+id/passwordET"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your password here"
        android:inputType="textPassword"
        />

    <CheckBox
        android:id="@+id/chkBoxShowPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Show Password?" />

    <RadioGroup
        android:id="@+id/employeeTypeRG"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <RadioButton
            android:id="@+id/rbtnStaff"
            android:text="Staff"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <RadioButton
            android:text="Student"
            android:id="@+id/rbtnStudent"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </RadioGroup>

    <Button
        android:id="@+id/btnSubmit"
        android:text="Submit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
```

MainActivity.kt

19. Open **MainActivity** Activity and add in a variable called “loginType”.

```
class MainActivity_Intermediate : AppCompatActivity() {

    var loginType: String = "Student"
```

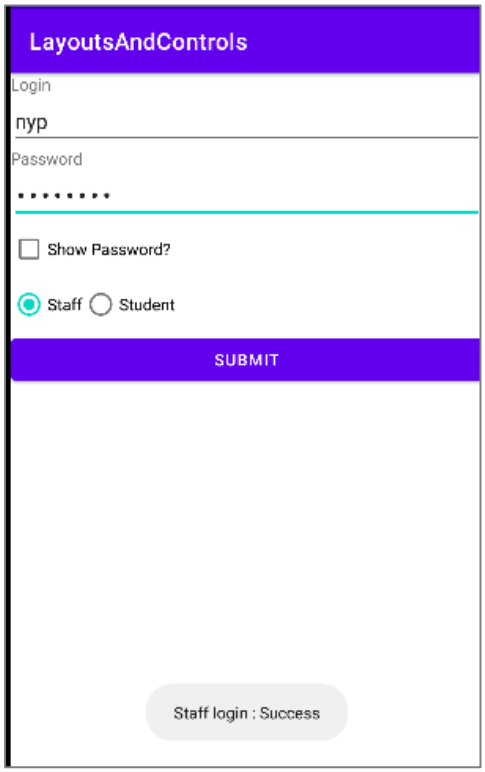
20. Inside btnSubmit's onClickListener, update the onCickListener to display login type in the Toast message.

```
btnSubmit.setOnClickListener { it: View!


    if (passwordET.text.toString() == "password") {

        displayToast(loginType + " login : Success")

    } else {

        displayToast(loginType + " login : Error")

    }

}
```

21. Inside onCreate, add in the following codes to update employee type and to reveal the password that the user keyed in.

```
employeeTypeRG.setOnCheckedChangeListener(object:RadioGroup.OnCheckedChangeListener{

    override fun onCheckedChanged(p0: RadioGroup?, p1: Int) {
        if (p1 == R.id.rbtnStaff)
        {
            loginType = "Staff"
        }
        else if(p1 == R.id.rbtnStudent)
        {
            loginType = "Student"

        }
    }

})

chkBoxShowPassword.setOnClickListener { it: View!

    if(chkBoxShowPassword.isChecked == true)
        passwordET.inputType= InputType.TYPE_TEXT_VARIATION_PASSWORD
    else
        passwordET.inputType= InputType.TYPE_TEXT_VARIATION_PASSWORD or InputType.TYPE_CLASS_TEXT

}
```

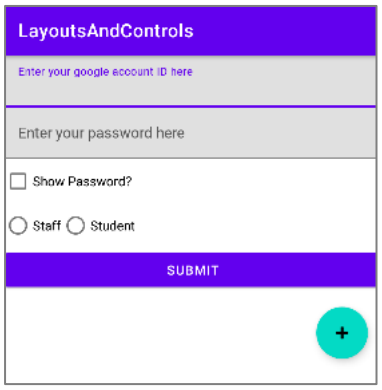
22. Run the app and experiment with the password, radio button and checkbox before clicking on the **Submit** button.


LayoutsAndControls
Login
nyp
Password
Show Password?
Staff
Student
SUBMIT
Staff login : Success

Exercise 3: Layouts and Controls [Advanced]

In this exercise you will create the following layout with material design.

You will use improve the **EditText**, add a **Floating Action Button** and replace **Toast** with **Snackbar**.



Steps

activity_main_advanced.xml

23. Create a new layout resource file called **activity_main_advanced.xml**.
24. Copy the codes from the current layout resource, i.e. **activity_main.xml**, and replace the codes into **activity_main_advanced.xml**.
25. Remove the two Textviews above the two EditText.

```
<TextView                                          Remove
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Login" />

<EditText
    android:id="@+id/loginET"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your google account ID here" />

<TextView                                          Remove
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Password" />

<EditText
    android:id="@+id/passwordET"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your password here"
    android:inputType="textPassword" />
```

26. Surround the existing loginET EditText with **TextInputLayout**.

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/input_layout_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <EditText
        android:id="@+id/loginET"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your google account ID here" />

</com.google.android.material.textfield.TextInputLayout>
```

27. Update the loginET EditText to **TextInputEditText** .

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/input_layout_login"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/loginET"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your google account ID here" />
</com.google.android.material.textfield.TextInputLayout>
```

28. Do the same for **passwordET** EditText.

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/input_layout_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/passwordET"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your password here"
        android:inputType="textPassword" />
</com.google.android.material.textfield.TextInputLayout>
```

29. Add in the floating action button after the submit button.

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end|bottom"
    android:layout_margin="16dp"
    android:src="@android:drawable/ic_input_add" />
```

MainActivity.kt

30. Comment the previous import.
    Replace with the new import.

```
// import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.activity_main_advanced.*
```

31. Update **MainActivity** to inflate **activity_main_advanced.xml**.

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main_advanced)
```

32. Update MainActivity to add in the onClickListener for the floating action button.

```
chkBoxShowPassword.setOnClickListener { it: View!

    if(chkBoxShowPassword.isChecked == true)

        passwordET.inputType= InputType.TYPE_TEXT_VARIATION_PASSWORD
    else
        passwordET.inputType= InputType.TYPE_TEXT_VARIATION_PASSWORD or InputType.TYPE_CLASS_TEXT

}

fab.setOnClickListener { it: View!

    Snackbar.make(it, text: "Snackbar",Snackbar.LENGTH_LONG)
            .setAction( text: "Action", listener: null)
            .show()

}
```

As you type, Android studio will help import Snackbar. If not, hover on Snackbar and click **Import**.

33. Update the onClickListener for btnSubmit to check for invalid input for both EditText.

```
if(loginET.text.toString().isEmpty()){

    loginET.error = "login cannot be empty"
}
else if(passwordET.text.toString().isEmpty())
    passwordET.error = "password cannot be empty"

else if (passwordET.text.toString() == "password") {
```

34. Run the application. Play around with the new EditText and Button.