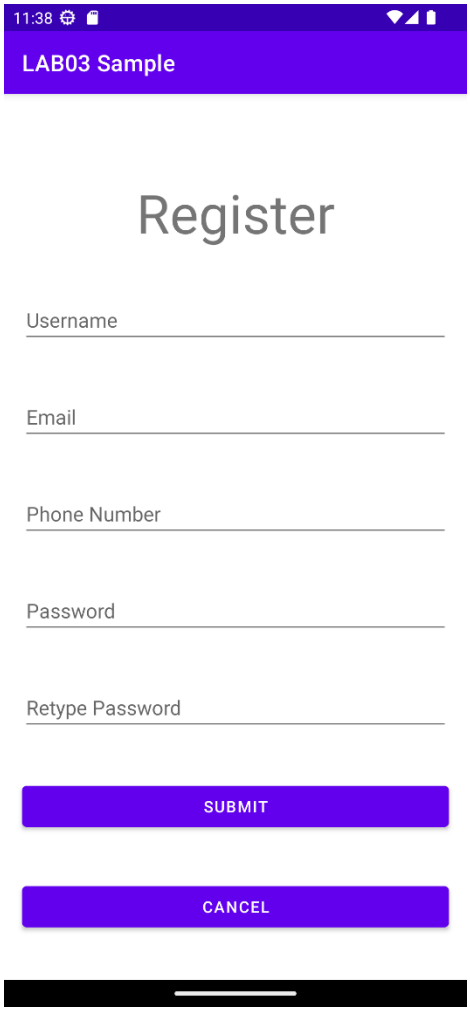
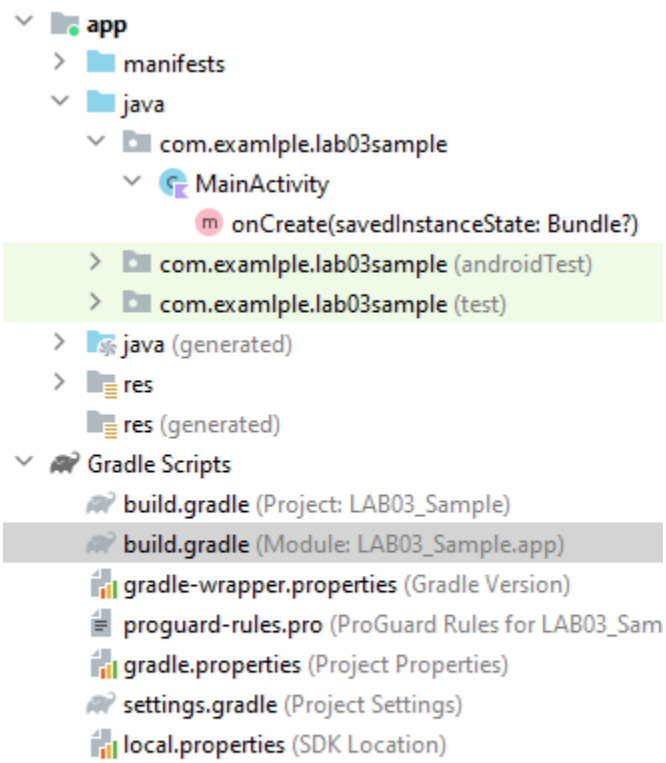
IT2010 – Mobile Application Development
BSc (Hons) in Information Technology
2nd Year
Faculty of Computing
SLIIT
2023 – Lab 03

1. Open a new Android Project in Android Studio
2. Design the following user interface.



3. In your app's build.gradle file, add the following dependencies for data binding:

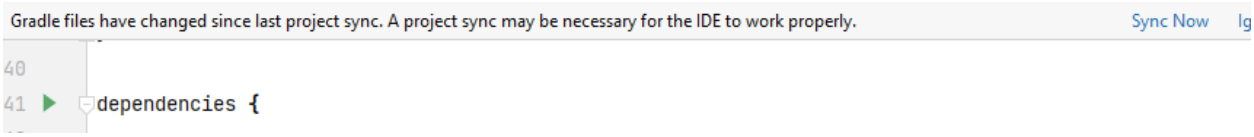



```
plugins {
    …// insert the following line to the plugins section
    id 'kotlin-kapt' // If using Kotlin
}
```

and

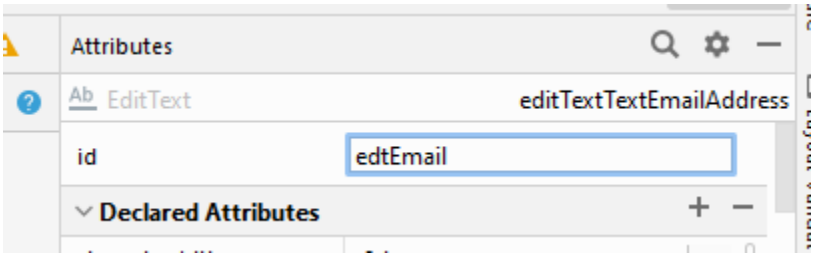
```
android {
  …//Insert here the below code
  buildFeatures {
    dataBinding true
  }
}
```

4. Then click sync now

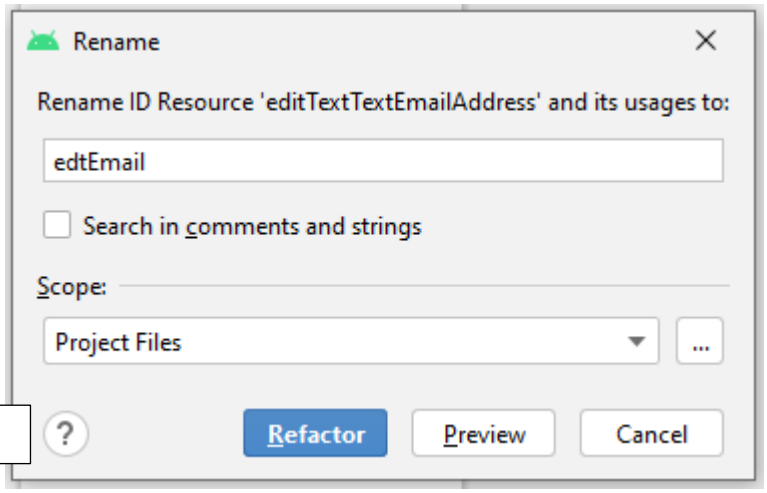


5. After that open the activity_main.xml.

6. Modify the id's so that it is easy to identify



Press enter.



Click Refactor

7. Modify all the ids.

```
edtName
edtEmail
edtPhone
edtPassword
edtRePassword
btnSubmit
btnCancel
```

8. Then switch to the xml view and surround the existing layout with the below tag.

```
<layout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

…// old layout goes here.

</layout>
```

9. Now, open the MainActivity.kt
10. Create a variable named binding as follows before the onCreate method

```
private lateinit var binding: ActivityMainBinding
```

11. Initialize the binding variable inside the onCreate method.

```
binding = DataBindingUtil.setContentView(this, R.layout.activity_main)
```

12. Initialize the views as follows. This should be in the onCreate method.

```
val edtName:EditText = binding.edtName
val edtEmail:EditText = binding.edtEmail
val edtPhone:EditText = binding.edtPhone
val edtPassword:EditText = binding.edtPassword
val edtRePassword:EditText = binding.edtRePassword
val btnSubmit:Button = binding.btnSubmit
val btnCancel:Button = binding.btnCancel
```

13. Implement the showAlertBox function

```
fun showAlertBox(
    context: Context,
    name: String,
    email: String,
    phone: String,
    password: String,
    rePassword: String
) {
    val builder = AlertDialog.Builder(context)
    val message = "Email: $email\n" +
            "Phone: $phone\n" +
            "Passwords: ${if (password == rePassword) "Matching" else "Not
Matching"}."

    builder.setTitle("Welcome $name!")
    builder.setMessage(message)

    builder.setPositiveButton("Ok") { _, _ ->
        Toast.makeText(context, "Submitted", Toast.LENGTH_LONG).show()
    }

    builder.setNegativeButton("Cancel"){_, _ ->

    }

    val dialog = builder.create()
    dialog.show()
```

14. Implement the button click event for the submit button

```
btnSubmit.setOnClickListener {
    showAlertBox(
        this,
        edtName.text.toString(),
        edtEmail.text.toString(),
        edtPhone.text.toString(),
        edtPassword.text.toString(),
        edtRePassword.text.toString()
    )
}
```

15. Implement the button click even for the cancel button

```
btnCancel.setOnClickListener{
    edtName.setText("")
    edtEmail.setText("")
    edtPhone.setText("")
    edtPassword.setText("")
    edtRePassword.setText("")
}
```

16. Run the program and observe the output
17. Modify the code by introducing a data class to store the form data.
18. Try to handle the empty values by informing them to enter values using Toast messages.