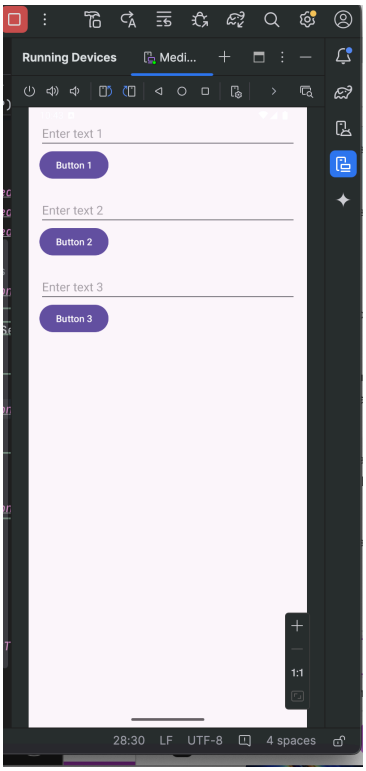
# My first Android application

EREN KIZILIRMAK 210704025

DO NOT DELETE ANYTHING FROM THIS DOCUMENT

- Run your first application and paste the screen capture of your emulator showing the output here




Lets now add a second activity to your Android application that includes a button which, when clicked, opens the MainActivity.

This will involve creating the new activity, updating the AndroidManifest, designing the layout for the new activity, and implementing the intent to navigate back to MainActivity.

## Step 1: Create the Second Activity

1. **Create a New Activity**: In Android Studio, go to File > New > Activity > Empty Activity.
2. **Name the Activity**: Call it SecondActivity and click Finish. Android Studio will automatically generate SecondActivity.kt and a corresponding layout file activity_second.xml.

## Step 2: Update AndroidManifest.xml

Android Studio should automatically add the SecondActivity to your AndroidManifest.xml, but it's good practice to check if it's there. The entry should look something like this:

<activity android:name=".SecondActivity"/>

### Step 3: Design the Layout for SecondActivity

Edit the activity_second.xml to include a button that will be used to open the MainActivity. Here's how you can design it:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/openMainActivityButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Open Main Activity"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

### Step 4: Implement Intent in SecondActivity

In SecondActivity.kt, set up an OnClickListener for the button that will start the MainActivity using an Intent:

```
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

class SecondActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)

        val openMainActivityButton = findViewById<Button>(R.id.openMainActivityButton)
        openMainActivityButton.setOnClickListener {
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
        }
    }
}
```

### Step 5: Modify MainActivity to Open SecondActivity

We will use one of the buttons in MainActivity to open SecondActivity. Let's choose button1 for this example. Update button1's OnClickListener in MainActivity.kt:

```
button1.setOnClickListener {
    val intent = Intent(this, SecondActivity::class.java)
    startActivity(intent)
}
```

### Step 6: Test the Application

1. **Run the Application**: Use the emulator or a physical device to run your app.
2. **Navigate Between Activities**: Click on button1 in MainActivity to open SecondActivity, and then use the button in SecondActivity to navigate back to MainActivity.

This setup allows you to have two activities that can navigate back and forth between each other. Adjust the button and activity names as needed to fit the context of your specific application.

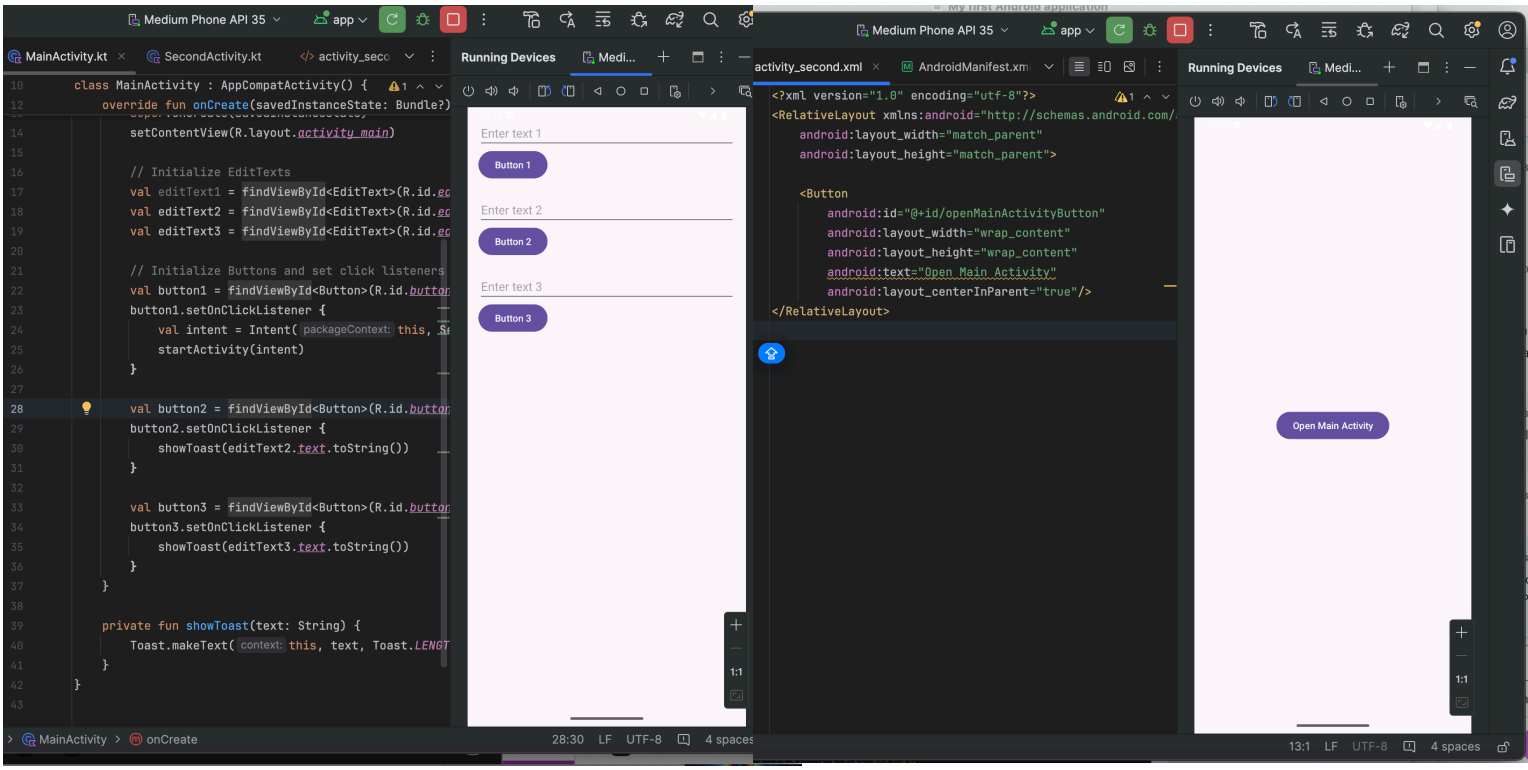
In a typical Android Studio project, when you create a new activity using the IDE's built-in template (like Empty Activity), the Android Manifest is updated automatically to include the new activity. It's good practice to check but usually not necessary to manually add the entry.

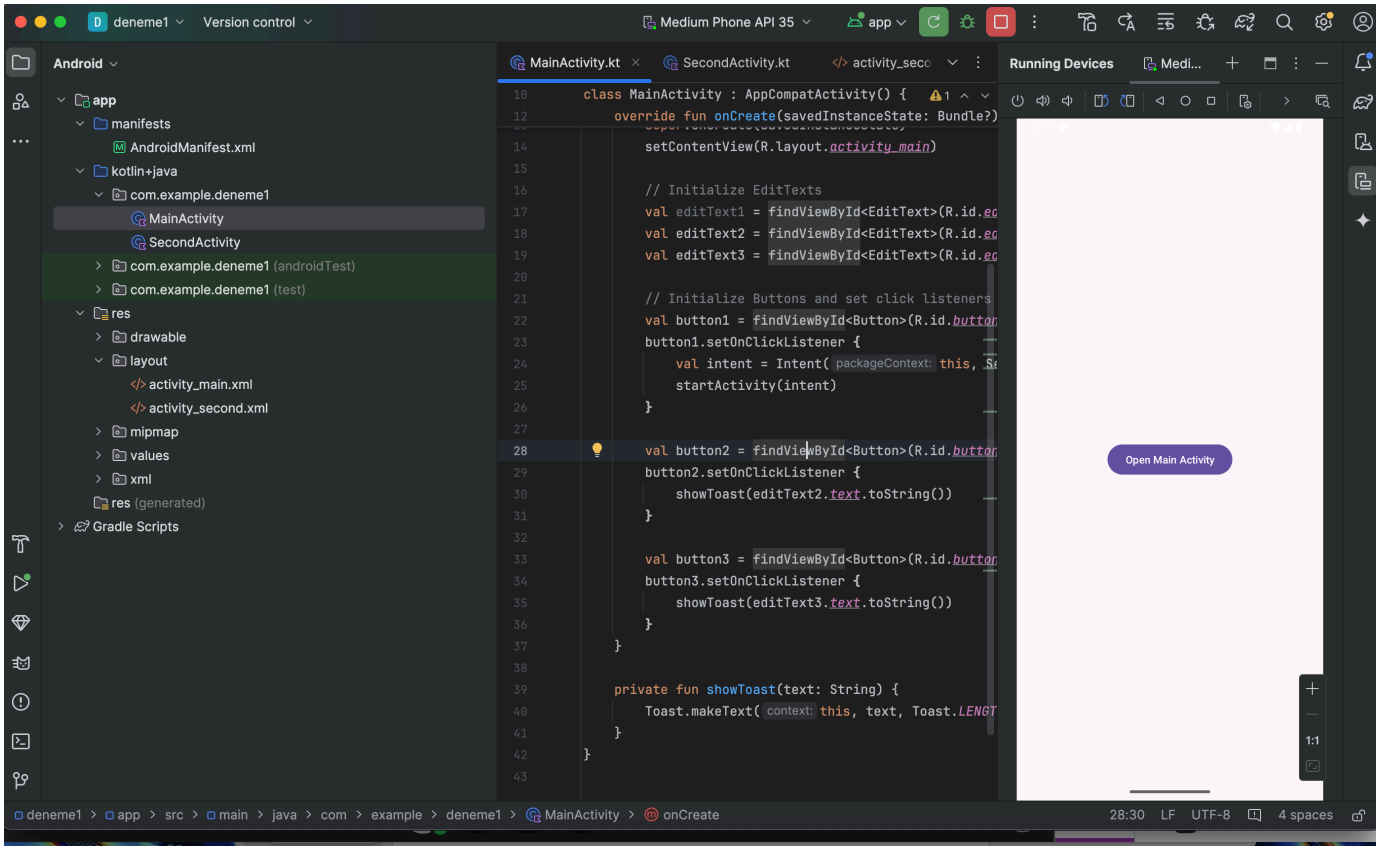
For clarity, the entry added by Android Studio looks like this:

<activity android:name=".SecondActivity" />

This entry ensures that Android's operating system recognizes SecondActivity as a legitimate part of your application, enabling it to be launched. If this entry is missing, you will encounter an error when trying to start SecondActivity because the Android system won't recognize it as a valid activity.

- Paste screen capture of your emulator showing all activities implemented here




deneme1
Version control
Medium Phone API 35
app
Android
app
manifests
AndroidManifest.xml
kotlin+java
com.example.deneme1
MainActivity
SecondActivity
com.example.deneme1 (androidTest)
com.example.deneme1 (test)
res
drawable
layout
activity_main.xml
activity_second.xml
mipmap
values
xml
res (generated)
Gradle Scripts
MainActivity.kt
SecondActivity.kt
activity_seco
class MainActivity : AppCompatActivity() {
override fun onCreate(savedInstanceState: Bundle?)
setContentView(R.layout.activity_main)
// Initialize EditTexts
val editText1 = findViewById<EditText>(R.id.ed
val editText2 = findViewById<EditText>(R.id.ed
val editText3 = findViewById<EditText>(R.id.ed
// Initialize Buttons and set click listeners
val button1 = findViewById<Button>(R.id.button
button1.setOnClickListener {
val intent = Intent( packageContext: this, Se
startActivity(intent)
}
val button2 = findViewById<Button>(R.id.button
button2.setOnClickListener {
showToast(editText2.text.toString())
}
val button3 = findViewById<Button>(R.id.button
button3.setOnClickListener {
showToast(editText3.text.toString())
}
}
private fun showToast(text: String) {
Toast.makeText( context: this, text, Toast.LENGT
}
}
Running Devices
Medi...
Open Main Activity
1:1
deneme1 › app › src › main › java › com › example › deneme1 › MainActivity › onCreate
28:30
LF
UTF-8
4 spaces