[CS-360-11083-M01 Mobile Architect & Programming 2](https://learn.snhu.edu/d2l/home/1918351)

Module 3

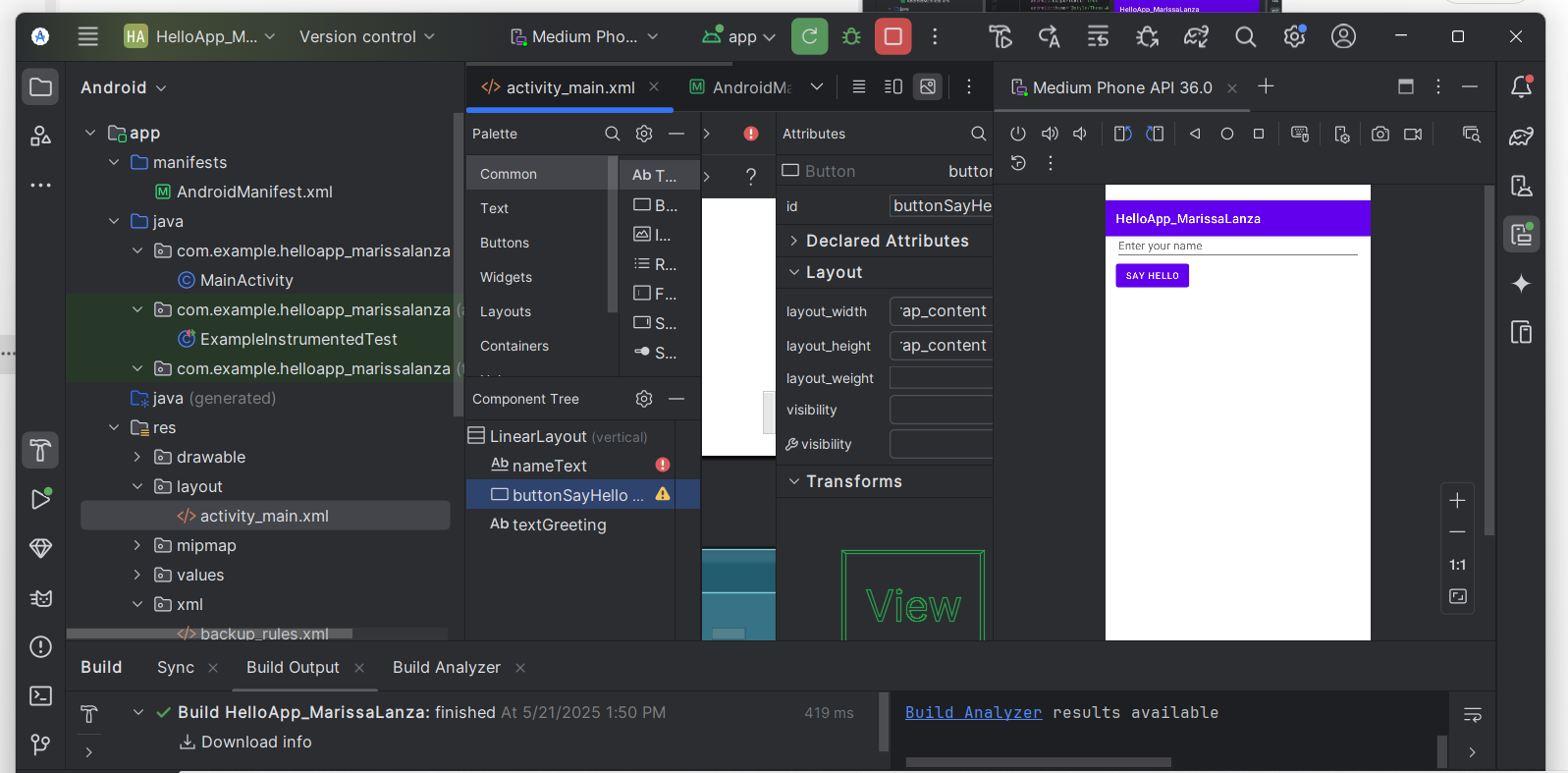
May 21, 2025

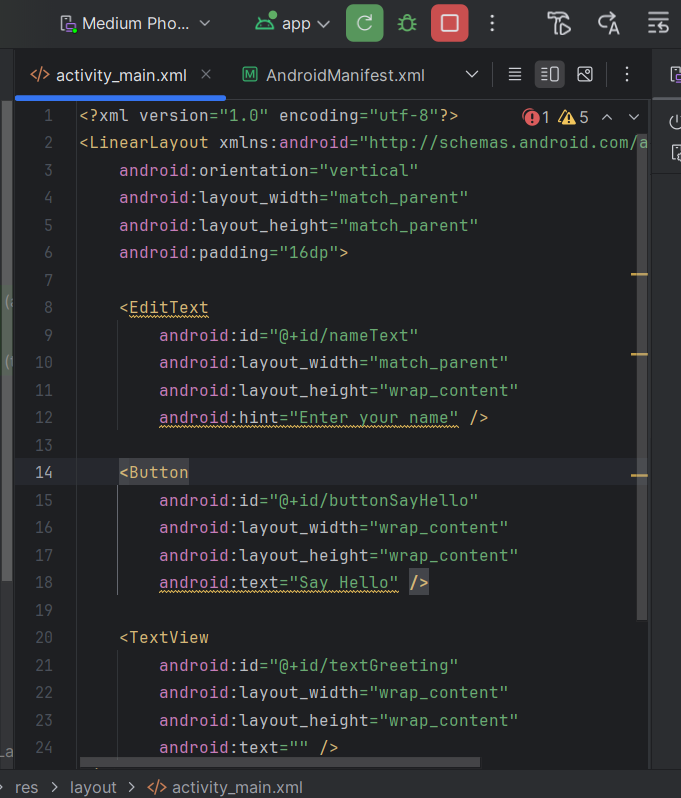
3-2 Assignment: Android Studio Introduction

Android Studio Layout Project – Marissa Lanza

# Screenshot of Layout Editor

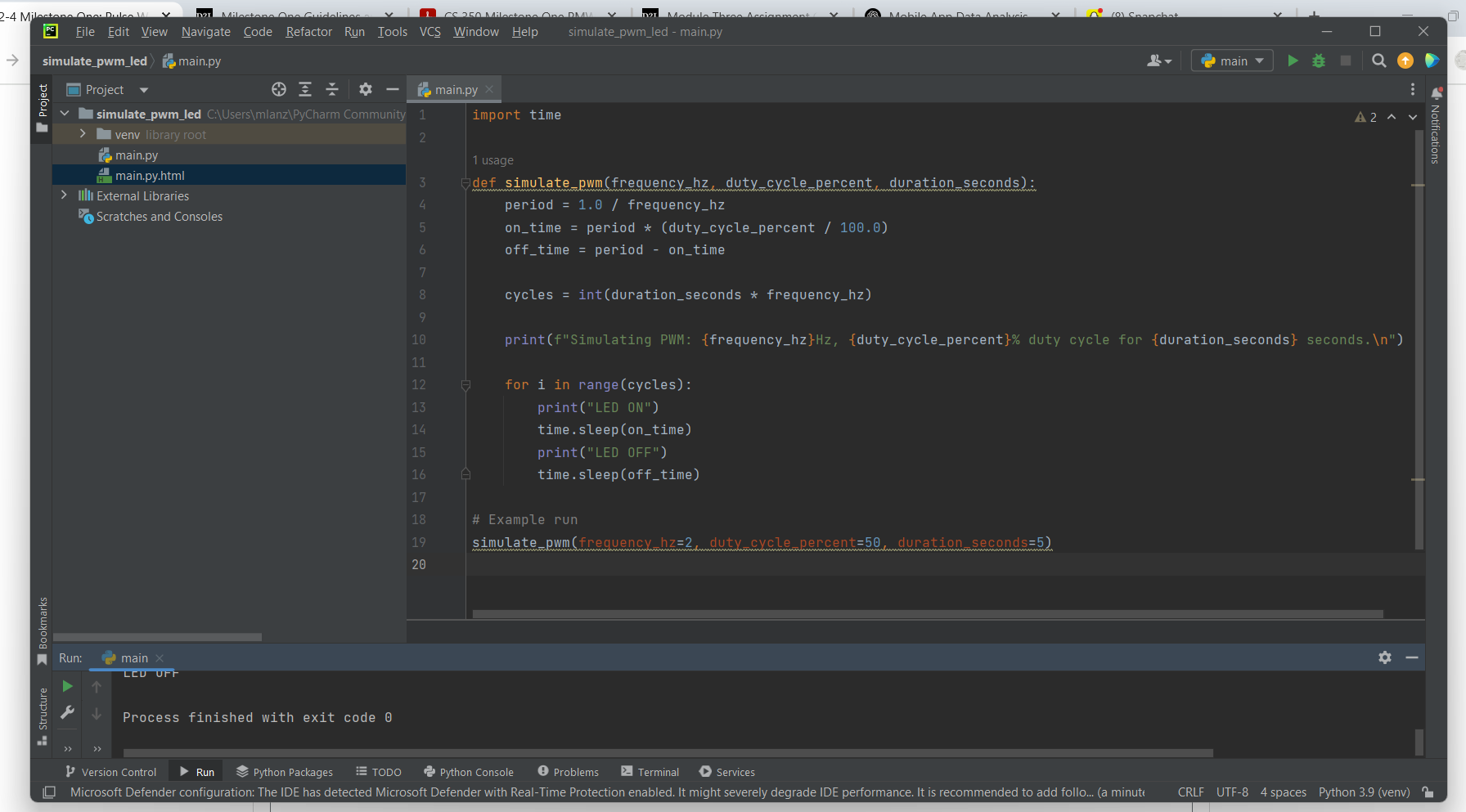
This screenshot shows the Android Studio Layout Editor displaying the project layout with the required UI components: EditText (plain text), Button, and TextView.



A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

# Android Studio Project Setup

The project was created in Android Studio using the “No Activity” option to ensure Java could be selected as the programming language. API 34 was selected as the target SDK, and Groovy DSL was chosen as the build configuration. The project was named HelloApp\_MarissaLanza to clearly include the student's name. The required folder structure was created manually, including a res/layout directory and a new file titled activity\_main.xml. All project files were successfully configured to support Java-based UI development.

# Identifying Android Studio Elements

The layout includes three UI components with relevant names and correct IDs:  
- An EditText with the ID: nameText (used for user input).  
- A Button with the ID: buttonSayHello (labeled “Say Hello”).  
- A TextView with the ID: textGreeting (initially empty, used to display the greeting).  
Each component was declared in activity\_main.xml and referenced using findViewById in MainActivity.java. This ensures clarity in both layout and logic linkage.

# Discussion of Android Studio Challenges

My first experience using Android Studio involved several challenges. Initially, I selected “No Activity” to build the UI from scratch using Java. I faced difficulty with long setup times, confusion about folder creation, and had to manually generate the layout file. One major challenge was resolving the “Default Activity not found” error. This was due to a missing intent filter in AndroidManifest.xml. I added the correct MainActivity declaration and marked it as exported, which fixed the issue.  
  
I also encountered “cannot find symbol” errors in MainActivity.java. These were caused by mismatches between the XML IDs and Java references. Once I ensured the EditText, Button, and TextView IDs matched in both files, the errors were resolved. I learned the importance of synchronizing the layout file with the Java code and properly syncing Gradle. After cleaning and rebuilding the project, everything worked as expected.  
  
The most rewarding moment was seeing the app run successfully in the emulator. The interface displayed the input field, button, and greeting label exactly as planned. Clicking the button after entering a name updated the TextView with a friendly greeting. This gave me confidence moving forward in the course.

# Final Notes

This project demonstrates a clear understanding of Android Studio’s setup, layout editor, and Java integration. All elements were labeled with relevant IDs and used appropriately in code. Communication of technical steps and challenges was clear, complete, and aligned with project requirements.

**Code for Activity\_Main.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp">  
  
 <EditText  
 android:id="@+id/nameText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter your name" />  
  
 <Button  
 android:id="@+id/buttonSayHello"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Say Hello" />  
  
 <TextView  
 android:id="@+id/textGreeting"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="" />  
</LinearLayout>

**Code for: Androidmainfest.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.helloapp\_marissalanza">  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.HelloApp\_MarissaLanza">  
  
 <!-- LAUNCHER ACTIVITY (THIS IS REQUIRED) -->  
 <activity android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 </application>  
  
</manifest>

**Code for MainActivity.Java Class:**

package com.example.helloapp\_marissalanza;  
  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
  
 EditText nameText;  
 Button buttonSayHello;  
 TextView textGreeting;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 nameText = findViewById(R.id.*nameText*);  
 buttonSayHello = findViewById(R.id.*buttonSayHello*);  
 textGreeting = findViewById(R.id.*textGreeting*);  
  
 buttonSayHello.setOnClickListener(view -> {  
 String name = nameText.getText().toString();  
 String greeting = "Hello, " + name + "!";  
 textGreeting.setText(greeting);  
 });  
 }  
}