

CS 399: Mobile Application Development Summer 2020

Homework-04

Due Friday June 12
30 Points Total

Hanyue Wang

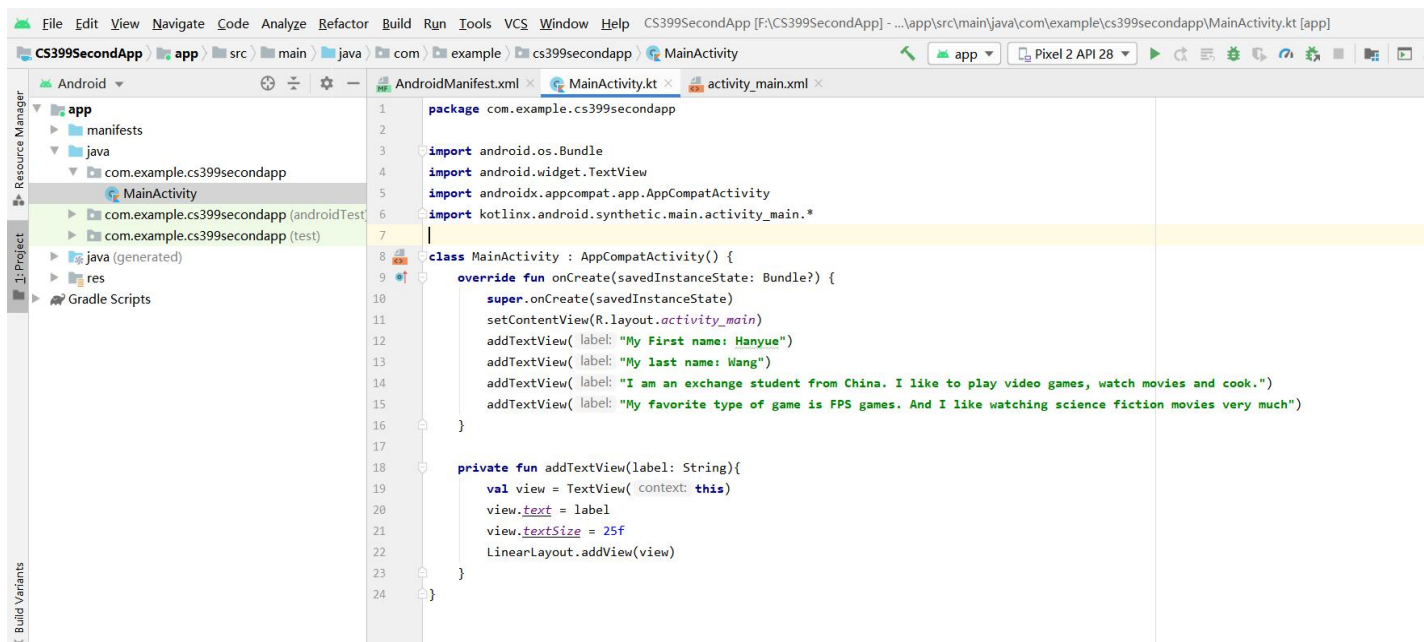
CS-399(1204-2860) SPECIAL TOPICS 002 - MOBILE
APPLICATION DEVELOPMENT

Deliverables:

[20 Points] Application Code and Result:

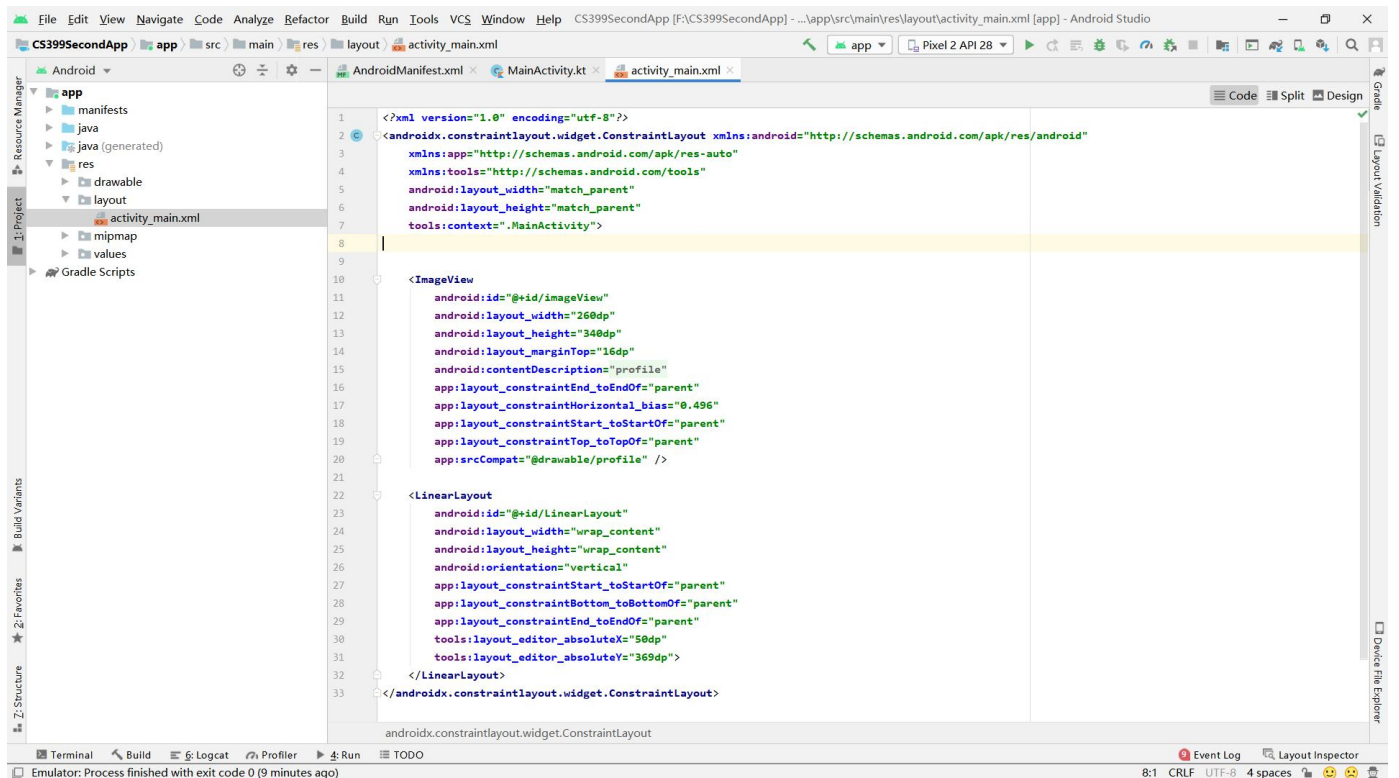
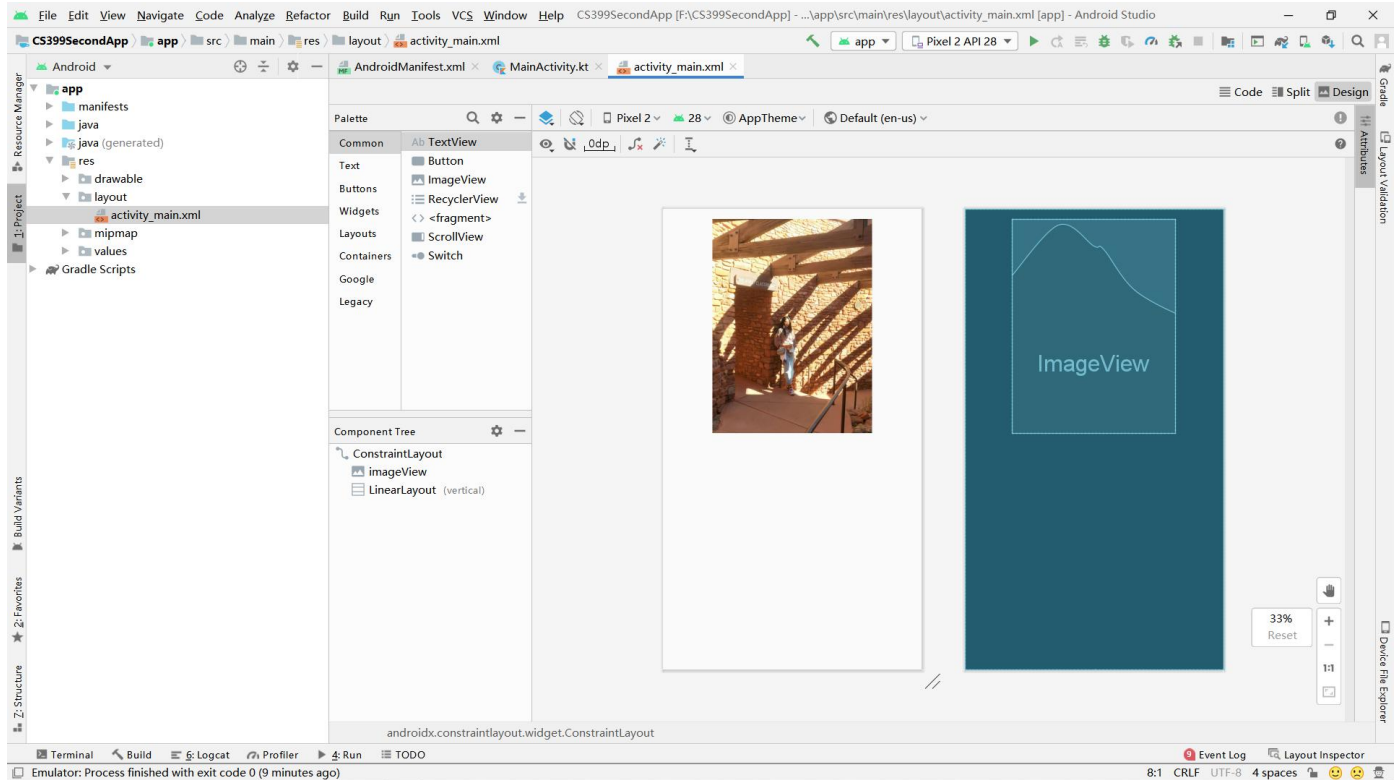
Include snapshots of the following artifacts shown in the Android Studio. Paste the snapshots in a copy of this document.

1. Components of your program including:
 - a. The Kotlin file under the *main > java* directory. Expand all the text to show the complete text of the Kotlin code including the *import* instructions.

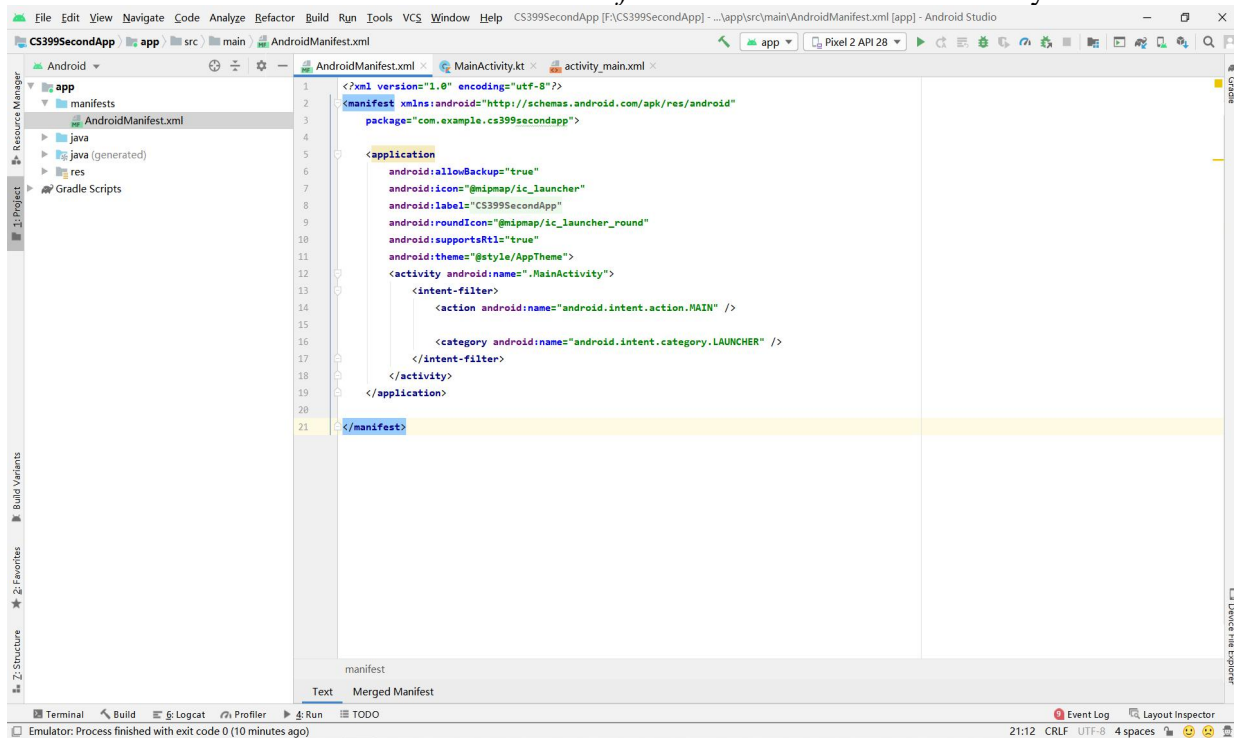


```
1 package com.example.cs399secondapp
2
3 import android.os.Bundle
4 import android.widget.TextView
5 import androidx.appcompat.app.AppCompatActivity
6 import kotlinx.android.synthetic.main.activity_main.*
7
8 class MainActivity : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12         addTextView(label: "My First name: Hanyue")
13         addTextView(label: "My last name: Wang")
14         addTextView(label: "I am an exchange student from China. I like to play video games, watch movies and cook.")
15         addTextView(label: "My favorite type of game is FPS games. And I like watching science fiction movies very much")
16     }
17
18     private fun addTextView(label: String){
19         val view = TextView(context = this)
20         view.text = label
21         view.textSize = 25f
22         LinearLayout.addView(view)
23     }
24 }
```

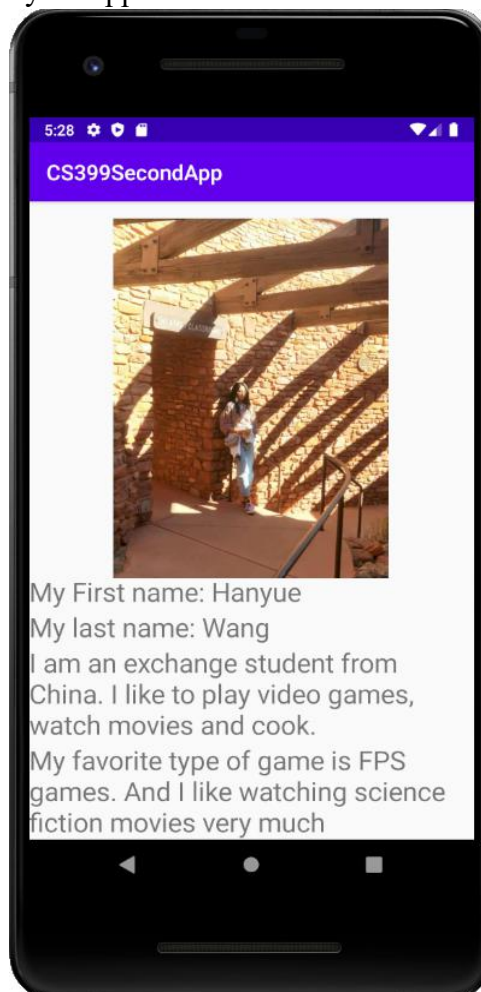
b. The layout xml file under *main > res* directory in both *design* and *text* editor views.



c. The *AndroidManifest.xml* file under *main* directory.

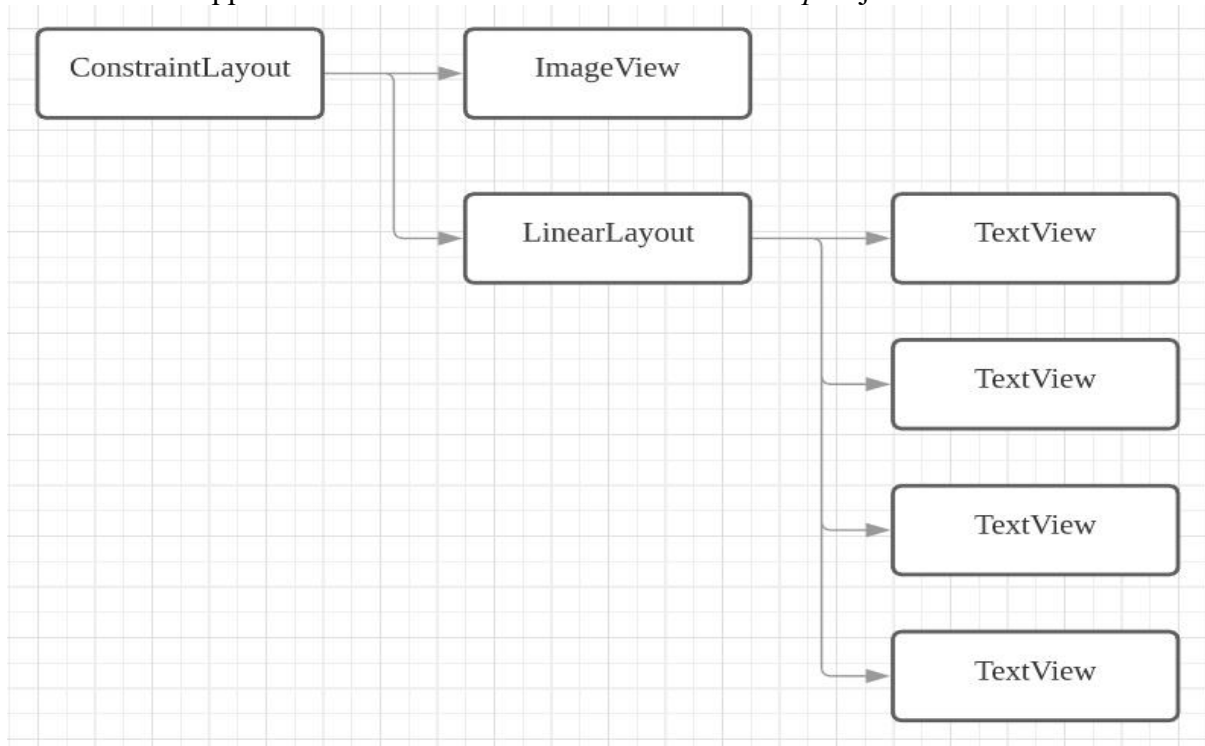


2. View of your application when it is run on the Pixel 2 virtual device.



[5 Points] Object Hierarchy:

Draw a hierarchical instance diagram of the *ConstraintLayout* object created in your Profile application. Include all the *View* and *ViewGroup* objects.



[5 Points] Class Hierarchy:

Draw a hierarchical class diagram showing the *inheritance* (class/subclass) relationships among all the *View* and *ViewGroup* classes. Start the root of your class hierarchy at `java.lang.Object` class. Don't include *implements* relationships with any *interface* in your model. Use the qualified name of the classes in your class diagram.

