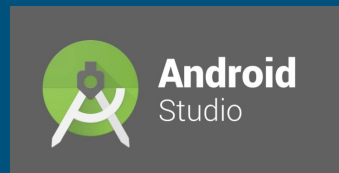




Shift Calendar

Shifting time and tasks to help you get stuff done

Planning



- ❖ Java
- ❖ Android Operating System and Software Development Kit (SDK)
- ❖ Gradle
- ❖ Markdown
- ❖ eXtensible Markup Language (XML)
- ❖ Android Studio
- ❖ IntelliJ platform



- ❖ Learning process using Coursera, FreeCodeCamp, and books.

The screenshot shows the Android Studio IDE with the following components:

- Top Bar:** Registration, activity_main.xml, MainActivity.java, and a toolbar with icons for running, debugging, and other IDE functions.
- Left Sidebar:** Project, Resource Manager, and a navigation pane with icons for Project, Structure, Run, and other tools.
- Main Editor:** Displays the MainActivity.java file with the following code:


```

1 package com.example.registration;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6
7 import androidx.appcompat.app.AppCompatActivity;
8
9 import android.os.Bundle;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 import android.os.Bundle;
14
15 import androidx.appcompat.app.AppCompatActivity;
16
17 public class MainActivity extends AppCompatActivity {
18
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23     }
24
25     //
26
27     Before you start creating, here are a few quick notes from the 1st hour of the FreeCodeCamp
28     Tutorial:
29
30     - After you install Android Studio, make sure you click on the option tab at the bottom right,
31     then select "Preferences" > "System", and change the updates channel to "Stable", not "Beta."
32
33     - Once you create a new application, always select Empty Activity, because this offers the most
34     amount of options to customize.
35
36     - Make sure you rename the project name to something you please, and you can leave the package
37     name alone, no changes needed.
38
39     - You can adjust the "activity_main.xml" file through "Design," "Split," and "Code" view. After
40     you drag an object onto the screen, make sure you adjust its id, and rename it to something
41     like camel-case, such as editText.
42
43     - Afterwards, adjust the constraints, not to reposition, but to make sure the buttons don't get
44     bunched up on together, and instead bounce off each other like security bubbles on different
45     screen sizes. Afterwards, scroll down to common attributes to change stuff like the hints and
46     the default text.
47
48     - Change the margins to find the space between objects on the UI (User Interface).
49
50     - Add different action listeners there, and make sure you create a corresponding function
51     right here on this file, like "onRegistrationClick()", as seen below.
52
53     - Before you work with any button, really, make sure you get its id, like:
54
55     TextView exampleButton = findViewById(R.id.exampleButton);
56
57     where the name of the button is "exampleButton".
58
59 
```
- Bottom Bar:** TOOD, Terminal, Database Inspector, Profiler, and Logcat. On the right, there are tabs for Event Log, Layout Inspector, and a status bar showing 11 LP, UTT-8, 4 spaces.

[courseur.org/learn/android-programming/home/welcome](#)

Apps

Gmail

YouTube

Maps

Yahoo Mail

Translate

Github

Articles

Internet Speed Te...

Other Bookmarks

courseura

Explore

What do you want to learn?

Q

🔔

William Zhang

▼

Overview

Week 1

Week 2

Week 3

Week 4

Week 5

Grades

Notes

1

Get 50% off when you invite a friend to learn on Coursera. [Terms and conditions](#)

Learn more

✕

Programming Mobile Applications for Android Handheld Systems: Part 1

by University of Maryland, College Park

START

WEEK 1

WEEK 2

WEEK 3

WEEK 4

WEEK 5

➔

My Weekly Goal

Mar 1 - Mar 7

Learn 5 days a week

Edit

Mo

Tu

We

Th

Fr

Sa

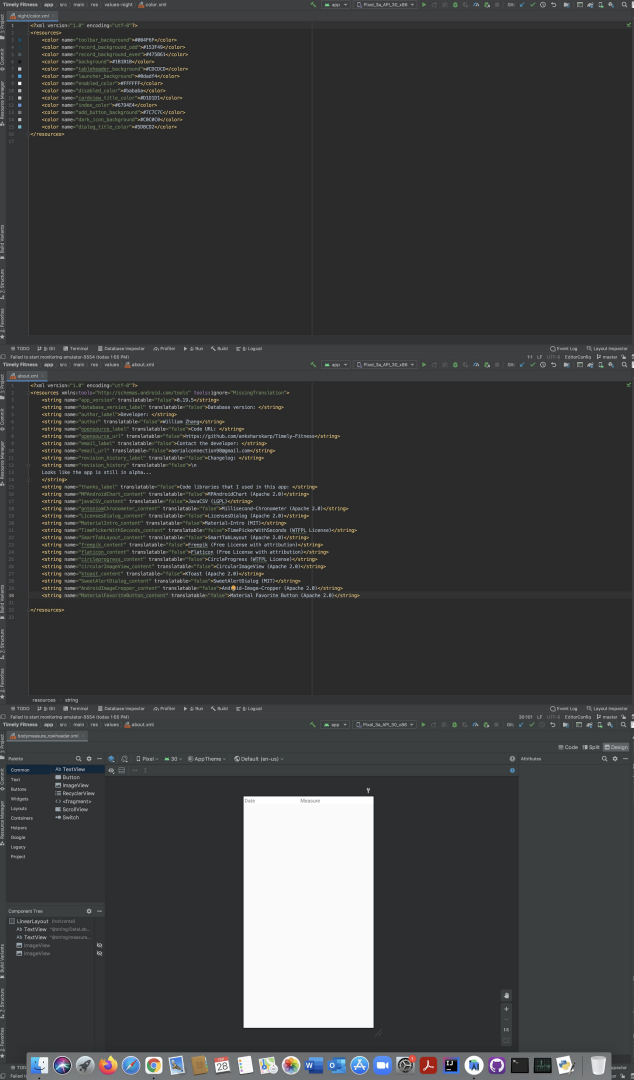
Su

1 day

Was this helpful?

Yes

No

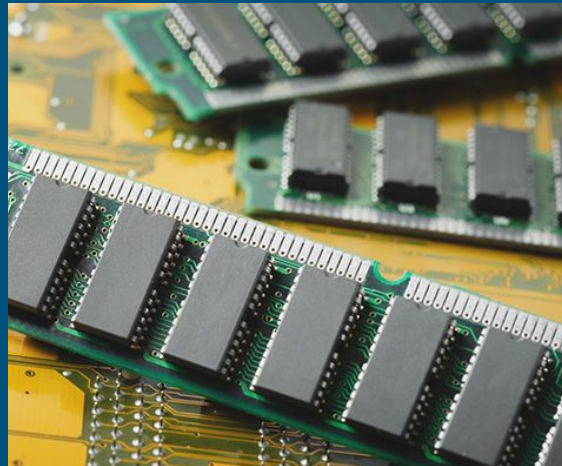


Designing

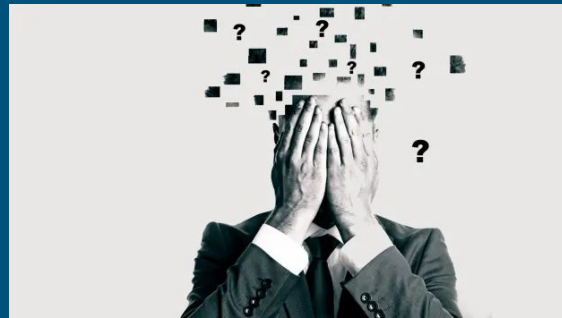
- ❖ XML problems and difficulties.
- ❖ Material Design
- ❖ Android Studio “drag and drop” interface
- ❖ Database creation using SQLite

Problems and Solutions

- ❖ Learning to use SQL
- ❖ Difference/conflicts between planning and implementation.
- ❖ Software bugs.



Additional Problems and Difficulties.



- ❖ Confusing API and documentation.
- ❖ External dependencies and code libraries (Gradle fails?)
- ❖ Software bugs/Git issues

Backend Application Logic

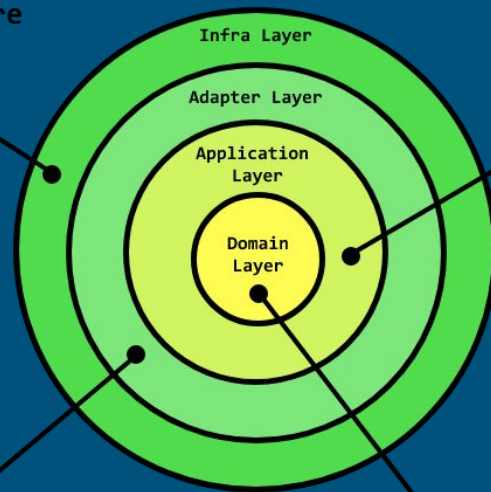
- ❖ Interactive User Interface
- ❖ SQL server
- ❖ Google Calendar API
- ❖ Display certain messages under detected events from the user.
- ❖ XML framework (“skeleton” for the app)

Infrastructure Details

Controllers
Routes
Databases
Caches
ORMs

Adapter Logic

Responsible for access to infrastructure and external APIs



Application Features

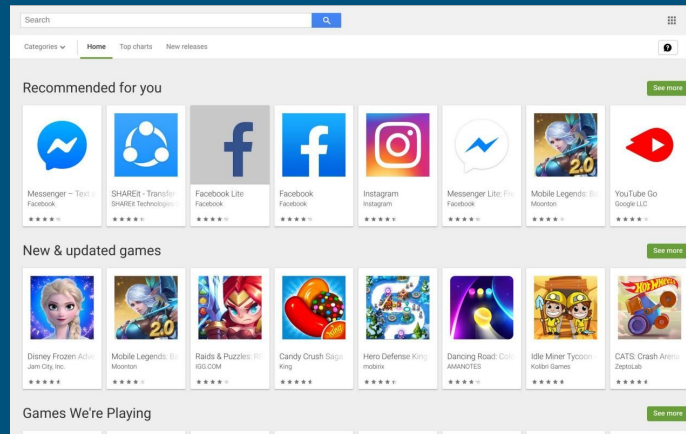
Use Cases / Application Services
The features of our application

Core Domain Logic

Entities
Value Objects
Domain Events

Packaging the Final Product

- ❖ Host project website on GitHub pages, and apply for the Google Play Store
- ❖ <https://www.github.com/Anksharskarp/Shift-Calendar>
- ❖ https://play.google.com/store?hl=en_US&gl=US



Reflections



- ❖ Problem solving/difficulties and strategies to overcome the issues.
- ❖ “Creating as you learn” is a process.
- ❖ Inquiry-based learning is the most efficient process of acquiring new knowledge (a useful discovery that I made during my passion project).
- ❖ Anticipate what you will need; don’t get it while you are already in the process of the creating the product.
- ❖ Research, watch, read, and most importantly, apply!