

## INFO 448A

## Homework 1 – Layouts, Widgets, Event Handling

**Goal:**

The goal of this assignment is to get familiar with layouts using views/widgets, event handling with Kotlin, and become familiar with Android Studio's IDE like the code & layout editor.

*Android Studio can take a while to download, install, and set up so make sure to start early. Set up instructions can be found on the course Canvas.*

**Homework Description:**

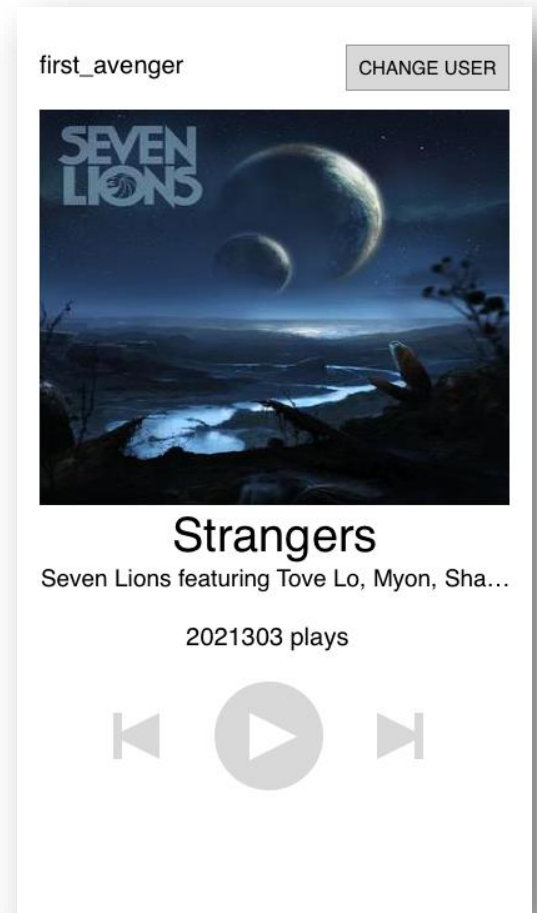
You've just joined a startup! They want you to make the next big music app! They are projecting a billion dollar company! But wait... They are a sleezy company. They are copycats! They want to rip off everyone else's design. But you don't care, you want the big \$\$\$\$. Let's hear it boss. ~\\_(\ツ)\_/~

They have this great app idea called **Dotify** where people listen to music on the cloud (sound familiar? 😞). Because they are still working on making their backend service, they left you in charge of making the first proof of concept screen on their Android app. Unfortunately it won't play any music (yet!)

**Requirements:**

You are to create a screen that shows a music player. They want to show a user's username, song's album art, song title, artist name, number of plays, and some media control buttons. Your app should have the following:

1. Your app's/project's name is EXACTLY called **Dotify**
2. The entire container must have a padding of 16 density pixels on all sides
3. Text that displays a username
  - a. The username should be constrained to the left side of the screen.
  - b. It should be vertically centered to the "Change User" button
4. A button with the text "Change User"
  - a. This button is constrained to the top right of the screen
5. A single large image of the album cover art that expands the full width of the screen and has a fixed height
  - a. The image source must come from your project's res/drawable directory
  - b. image must be below the username and "Change User" button
6. Text displaying the song's title with large font size
  - a. The text must be centered horizontally
7. Text of the artist's name whose font size is smaller than the song title
  - a. Must be below the artist & centered horizontally
  - b. The artist's name must only have a max number of lines of 1. (Try testing it with different text lengths)
  - c. If the artist's text is longer than what can fit on a single line within the screen bounds, the text should ellipsize (...) at the end.



8. Text displaying the number of song plays
  - a. The text should be in the format of “# plays” where # is the number of plays i.e. “3674 plays”
  - b. The initial number of plays should be a random integer between a fixed interval
9. 3 image buttons for media controls (previous, play, & next buttons)
  - a. Images must come from your project’s res/drawable directory
  - b. Each media button must be aligned horizontally next to each other
  - c. The media buttons must be distributed evenly across the screen’s full width horizontally.
  - d. Play Button
    - i. When a user clicks on play, it increases the number of plays in the text displayed from **requirement #8** by +1. (400 plays → 401 plays → 402 plays, etc.)
  - e. Previous Button
    - i. When a user clicks on the previous button, it should Toast a brief message that says “Skipping to previous track”
  - f. Next Button
    - i. When a user clicks on the next button, it should Toast a brief message that says “Skipping to next track”
10. Has the following behavior for the username & the “Change user” button from **requirement #3 & #4**
  - a. When a user clicks on “Change user”,
    - i. it should hide the username
    - ii. show an EditText view in place of the username for a user to type in a new username
    - iii. change the “Change user” button’s text to “Apply”
  - b. When a user clicks the “Apply” button
    - i. Hide the EditText view, show the username text
    - ii. Display the new text from the EditText field as the username

The song’s name, artist, cover art, media control button images can be anything you like! Could be your favorite song, you’re most hated, or even something completely random. Certain criteria is vague on purpose. Be creative! Your app is meant to be shown off on your GitHub, portfolio, and even potential employers. Don’t be like everyone else. **Write clean readable code.** There are many ways to achieve this assignment. There’s no single right answer.

(You are more than welcome to use the image assets provided on the Course canvas for starting off)

### Restrictions:

You are NOT allowed to write code in Java. Kotlin only.

You are NOT allowed to copy anyone else’s code. Reach out to me or your TA if you need help.

You are NOT allowed to have this screen play actual song/sound (we’ll get to this later in the course).

You are NOT allowed to build any additional screens/Activities.

You are NOT allowed to have remote image fetching. All Images must be stored locally in res/drawable folder.

### Tips:

- Start with the basic layout positioning. Minor details should be done last like margins, paddings, color, text sizing, etc. last
- Commit early and often! The moment you create your project, immediately set up your project to git. In Android Studio select **VCS > Enable Version Control Integration > Select Git and OK**. Committing frequently will allow you to revert back in history to a semi working state in case you get stuck.
  - Try to avoid committing the entire project in one commit.

- While not required, I encourage using ConstraintLayout as your main ViewGroup over LinearLayout/RelativeLayout. You won't be judged/docked if you don't use it. Just understand ConstraintLayout will come in handy later in the course.

### Extra Credit (Optional):

If you are going to do any of these, make sure you explicitly mention which ones in your README.md file

1. A user is not allowed to apply a new username if the edit text field is empty. (+ 0.25)
2. Long pressing on the cover image changes the color of all the text views to a different color. (+ 0.25)
3. If using ConstraintLayout, utilize a Barrier or Guideline somewhere with a view constrained to it (+0.25)
4. All hardcoded dimensions & colors are extracted into res/values/dimens.xml & res/values/colors.xml respectively (+ .5)
5. Instead of **requirement #5**, the album cover image must be a perfect 1:1 square size whose width matches the screen width and height matches the height as well. (+ 1)
6. Create another xml file that uses the a different ViewGroup type than your original. (+ 1)
  - a. i.e. If you used a ConstraintLayout for your homework, this second xml uses LinearLayouts, and vice versa (I recommend at least one of your xml should be ConstraintLayout)

### Submission:

- You must submit your project on GitHub account which you declared from Homework 0
- Your repository name should be EXACTLY called **Dotify**
- Your code MUST be committed on a branch EXACTLY called **hw1**
  - Try executing these git commands in a different folder to test it
 

```
$ git clone <github url>/Dotify
```

```
$ git checkout hw1
```

 And see if the project builds
- Your repository should have a README.md file at the root that:
  - Title of Homework, your full name
  - briefly describes the app
  - list of extra credit you attempted
  - Shows an photo or screenshot of your app running on a device/emulator
  - Optional: Any special set up/installation instructions or how to use the app
- Your app must use at least 4 different widgets/views
- Your app must be able to run on an emulator & device
- If you need more help with submission, please check out the **Assignment Submission Help & Tips** page on Canvas.