Simple Countdown Single-Timer App

## Requirements

Develop a sample timer app using Java and/or Kotlin given the requirements below.

**Your solution shall be evaluated based on following directions, app functionality, coding style, and a clean/maintainable implementation.**

The app shall launch to Screen2 (Show Timer Screen).

minSdk = 23

### Screen1: New Timer Screen

1. Create a new **countdown timer**.
   1. The user can specify a label associated with the timer.
   2. The newly created timer shall be visible on a “timer” screen.
   3. The timer shall automatically start after being created.

### Screen2: Show Timer Screen

1. The timer shall be displayed on the Show Timer screen
2. The user shall be able to pause, play, and delete the created timer:
   1. Pause/Play will be the same button.
   2. Timer paused shall show the play icon <https://www.flaticon.com/free-icon/play-button_149657?term=play&page=1&position=10>
   3. Timer running shall show the pause icon

<https://www.flaticon.com/free-icon/pause-button_61219?term=pause&page=1&position=2>

* 1. Delete icon top right hand corner <https://www.flaticon.com/free-icon/close_151882?term=close&page=1&position=2>

1. When clicking on the **FAB**, if there is already a running timer, the app shall display a toast indicating that a timer is already running.

### Running Timer Logic

1. On the show timer screen, the timer shall continue to countdown and update every second when the app is in the foreground. When the app is in the background (ie home screen pressed or another app launched), the timer shall continue to run such that the correct value is shown when the app is foregrounded (assuming the background process is not killed).
2. When the app is killed while in the background the timer shall be reset to the original value. Upon opening the app the timer shall not be running, but ready to be started at its original value.
3. The timer shall continue to run upon screen rotation (configuration change).
4. When the timer is completed, the app shall display a system notification with the label of the elapsed timer.
   1. If the user is on Screen2, the UI shall be updated by resetting the timer back to the original time and display the play icon.
5. You DO NOT need to account for the system’s doze mode.

## Design

NOTE: These are reference designs and DO NOT have to be implemented to precisely match the reference.

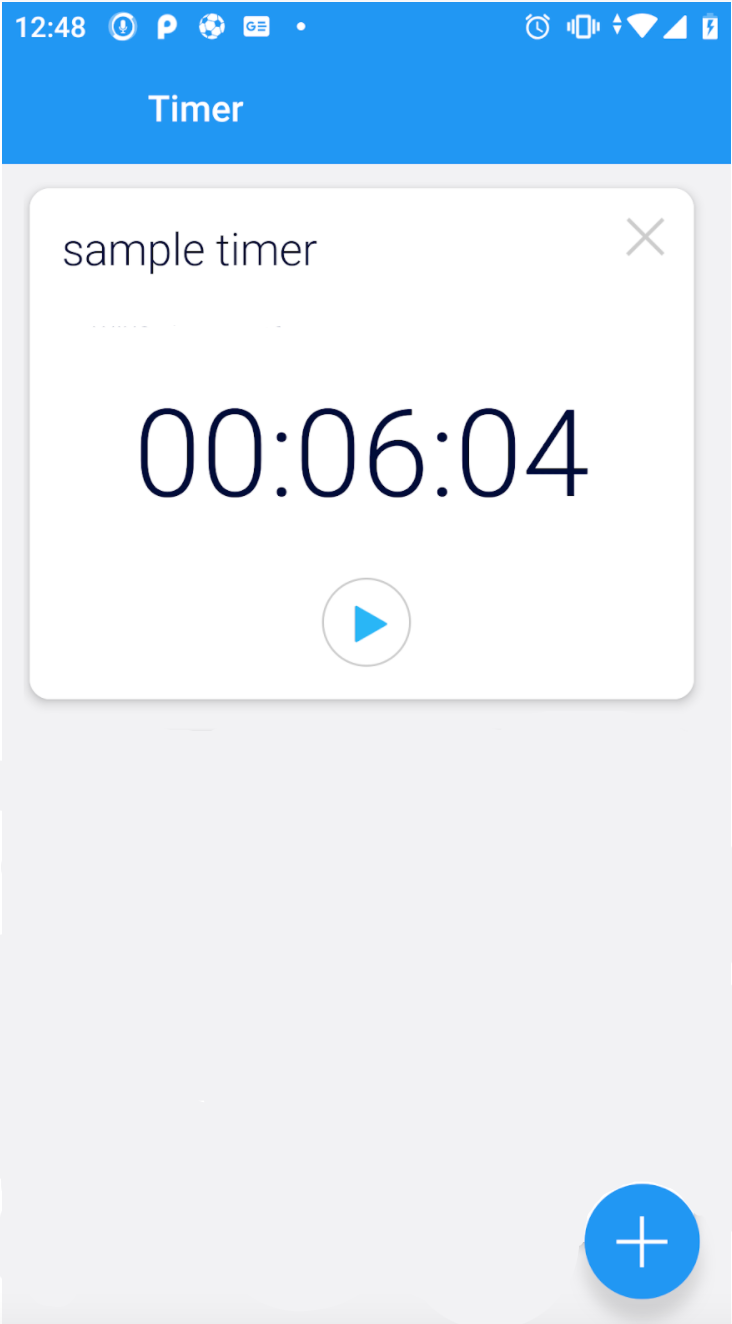
### Screen1: New Timer Screen



Notes:

1. This screen is launched by clicking the **+ FAB** button from Screen2.
   1. Creating a new timer or closing this screen will take the user back to Screen2.
2. **Timer** in the screen above is the label/title that is editable by the user.

### Screen2: Show Timer Screen



Notes:

1. **Please note that even though multiple timers are shown in the reference screen here, we would expect only a single timer for this solution.**
2. The app launches to this screen.
   1. Clicking the **+ FAB** button will open Screen1.
3. The labels above**, 6 minutes 9 seconds** and **56 minutes** are the initial values of example timers. **For this exercise, you only need to support a single timer.**
4. The title of the timer is the **label** that is specified when creating the timer in **Screen1:1.a**
   1. Example labels shown above are **another one** and **sample timer**