

## Exercise #02

IT University of Copenhagen (ITU)  
Mobile App Development, BSc. (MOAPD)  
(Spring 2024)

February 06, 2024

**Introduction** Throughout the semester, you will create an Android app named “CopenhagenBuzz.” This app is about sharing events in the Copenhagen area, like festivals and concerts. You will build it step by step, enhancing the app’s look each week and adding new features based on what you learn in class. You will turn in your finished app for a mandatory assignment at the end of the course. For Assignment #01, you must do the following tasks:

- ☐ The `Event` class implemented as a Kotlin *data class*.
- ☐ The use of *View Binding* to manage the UI components.
- ☐ A complete *Docstring* documentation in the `MainActivity.kt` file.

**CopenhagenBuzz App Version 2** This week, your task is to refactor the `MainActivity.kt` and `Event.kt` files, focusing on implementing and testing key Kotlin features. Check out **Slides #02** for inspiration to elevate your project. The learning objectives for this exercise encompass:

- Refactoring the components of your Android app.
- Employing `data classes` to manage the information displayed in the UI efficiently.
- Using `ViewBinding` to streamline code and simplify the handling of UI components.
- Creating Kotlin documentation using comments in the `docstring` style.

**Exercise 02.01. Design Patterns and Package Structure** – As your project grows in complexity, this exercise guides you in refining your CopenhagenBuzz project by adopting a design pattern like Model-View-Controller (MVC) or Model-View-ViewModel (MVVM). Alternatively, you can opt for a restructuring approach that organizes your app’s package structure based on *categories* or *application features* (refer to <https://guides.codepath.com/android/Organizing-your-Source-Files>). To streamline this process, create distinct packages within the primary project package `dk.itu.moapd.copenhagenbuzz.<USER>`. Subsequently, relocate your Kotlin source code into these specific folders according to the chosen organizational method. Going forward, ensure all new Kotlin files are created or added in their designated packages.

**Exercise 02.02. Implementing data classes** – In Exercise 01.03, you created an `Event` class structured like Java code. Your existing version includes a secondary constructor, encapsulation methods for all attributes, and the `toString()` method. For this exercise, your goal is to refactor the `Event` class and transform it into a Kotlin `data class`. You can read more information about `data classes` in the official Kotlin documentation website.

**Exercise 02.03. Using ViewBinding** – In CopenhagenBuzz app V1, you employed the `findViewById()` method to link UI components with Kotlin variables. Consider leveraging `ViewBinding`, a beneficial Android feature that automates code creation for secure and efficient interaction with UI components. Your task in this exercise is to refactor the `MainActivity.kt` file by refactoring it to utilize `ViewBinding` to obtain references to your UI components. You can read more information about `ViewBinding` in the official Android developers documentation website.

**Exercise 02.04. Documenting Kotlin source-code** – A `docstring` is a string literal placed as the initial statement in a class, module, function, attribute, or method definition. It serves the purpose of automatically generating documentation for your project. In this exercise, your task is to compose documentation for the `MainActivity.kt` and `Event.kt` files:

- You must initiate each Kotlin file with a software license to establish the rules for using or redistributing your source code. Utilize the Choose a License website to determine the suitable open-source license.
- You must formulate the documentation for your Kotlin source code using KDoc. Adhere to the *official Android code style guide* and *Kotlin style guide* throughout this process.

**Exercise 02.05. App progress messaging** – In this exercise, incorporate a `Snackbar` to present concise messages regarding your app's progress at the bottom of the screen, replacing the `Log.d()` method. Upon pressing the `Add Event` button, your app should display a message as shown in Figure 1. For further insights on `Snackbars`, refer to the official Android developers documentation website or the Material Design documentation website.

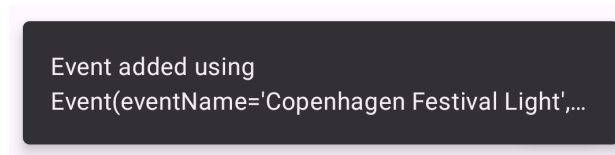


Figure 1: Illustration showcasing the usage of a Snackbar in the CopenhagenBuzz app.