Exercise #04

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

February 20, 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 use of `Fragments` in at least three user interface screens.
- ☐ The use of `Android Jetpack Navigation` to manage the app's navigation flow.
- ☐ The integration between `Bottom Navigation Bar` and `Jetpack Navigation`.

CopenhagenBuzz App Version 4 This week, you will work on the CopenhagenBuzz app V4. This version introduces a shift from Activities to Fragments to manage user interfaces. Your main task will involve adapting the MainActivity class to incorporate Fragments. Figure 1 illustrates the fragments you will develop during this week.

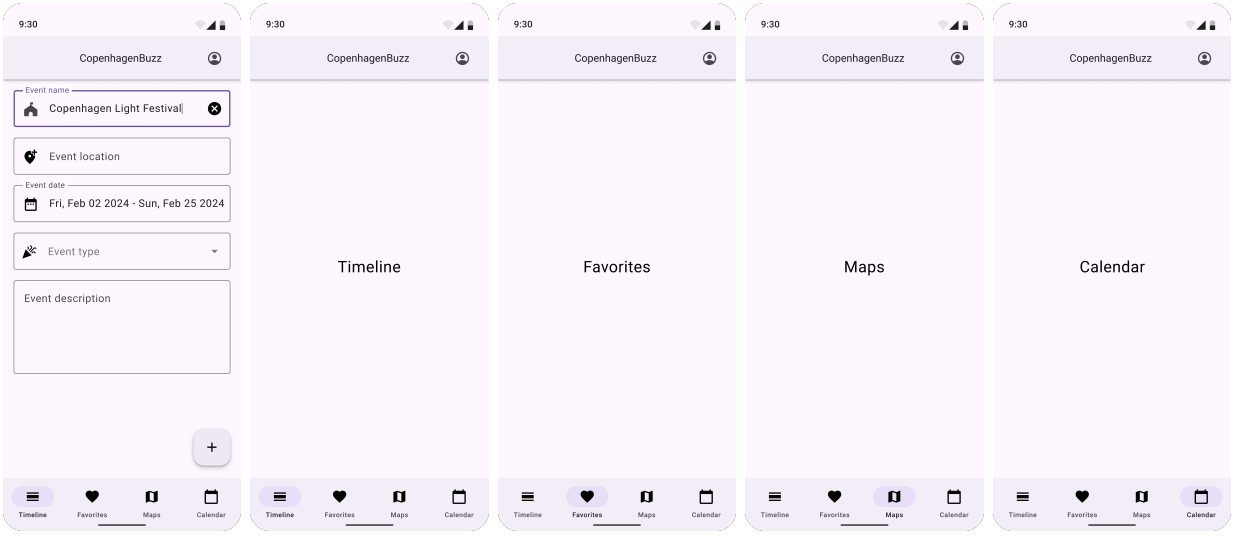

Figure 1: You will create these user interfaces for the CopenhagenBuzz app V4.

The goals you will achieve through this exercise include:

- Integrating `Fragments` with `Navigation` components, ensuring that each fragment represents a distinct destination within the app and can be navigated to and from using the `Navigation Bar`.
- Implementing `Navigation` components, including `NavHostFragment` and `NavController`, to facilitate seamless navigation between different app sections.
- Enhancing the user experience by incorporating a `Bottom Navigation Bar` and make it easier for users to navigate and access different features or content.

Exercise 04.01. *Creating advanced user interfaces* – In this exercise, you will create at least three fragments for your CopenhagenBuzz app: (1) one for displaying the next events in the Copenhagen area (`TimelineFragment`), (2) another for showing user-favorite events (`FavoritesFragment`), and (3) a third fragment with Google Maps to display event locations in a map (`MapsFragment`). In the following weeks, you will implement the behavior these user interfaces. However, today you can *mock* them with a static image (JPG or PNG) as a background if you want to. Optionally, you can add a fourth fragment (`CalendarFragment`) for a calendar view of events (see the item "'Calendar" in Figure 1), but it is NOT mandatory.

Exercise 04.02. *Refactoring the add event component* – In this exercise, you will refactor the code you developed during Exercise #01, Exercise #02, and Exercise #03 in the `MainActivity.kt` and `content_main.xml` files. Instead of implementing the user interface directly in these files, you will copy and paste the existing user interface, specifically the one used to add new events to the app, into a new `Dialog`, `Activity`, `Fragment`, `BottomSheet`[1], or any other UI component of your preference. By doing so, you will be able to streamline the development process and prepare your project to implement the `Navigation` component in the next exercise, leveraging the `MainActivity.kt` and `content_main.xml` files as the primary user interface components.

Exercise 04.03. *Using a fragment navigator* – In this exercise, you will implement the navigator component in your CopenhagenBuzz app to facilitate navigation between the fragments created in Exercise 4.01. The `Navigation` contains three essential components that collaborate seamlessly:

1. **Navigation Graph** (`res/navigation/nav_graph.xml`): A centralized resource containing all navigation-related details, including destinations (`places`) within your app and the possible user paths.

[1] `https://m3.material.io/components/bottom-sheets/overview`

2. **NavHostFragment** (`res/layout/content_main.xml`): A unique widget added to your layout, responsible for displaying various destinations from the Navigation Graph.

3. **NavController** (`MainActivity.kt`): An object that manages the current position within the navigation graph, facilitating the swapping of destination content within the `NavHostFragment` as users navigate through the app.

Important: Before using Jetpack Navigation, make sure to install the necessary dependencies to ensure smooth integration and functionality within your project, as shown in Listing 1. Also, make sure to read the official documentation[2] to understand how to use the `Navigation` component.

Listing 1: Using the `LoginActivity` as the application's starting point.

```
dependencies {
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.core:core-ktx:1.12.0")
    implementation("androidx.navigation:navigation-fragment-ktx:2.7.6")
    implementation("androidx.navigation:navigation-ui-ktx:2.7.6")
    implementation("com.google.android.material:material:1.11.0")
}
```

Exercise 04.04. *Adding a Bottom Navigation Bar* – In this exercise, you will integrate a `Navigation Bar` into your CopenhagenBuzz app with the previously created fragments, as shown in Figure 2. A `Navigation Bar`, also known as a `Bottom Navigation Bar`, is a visual guide for users to navigate through different sections or features of your app easily. It allows users to switch between primary destinations quickly, reducing the effort required to explore your app's content. Integrating a Navigation Bar ensures consistency in navigation across your app. It enhances user engagement by providing a seamless browsing experience.

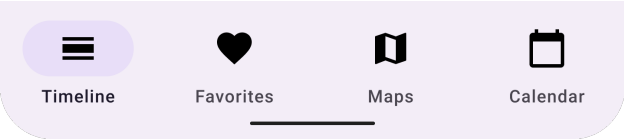

Figure 2: Example of a `Bottom Navigation Bar` to access four distinct fragments.

[2] `https://developer.android.com/guide/navigation/navigation-getting-started`