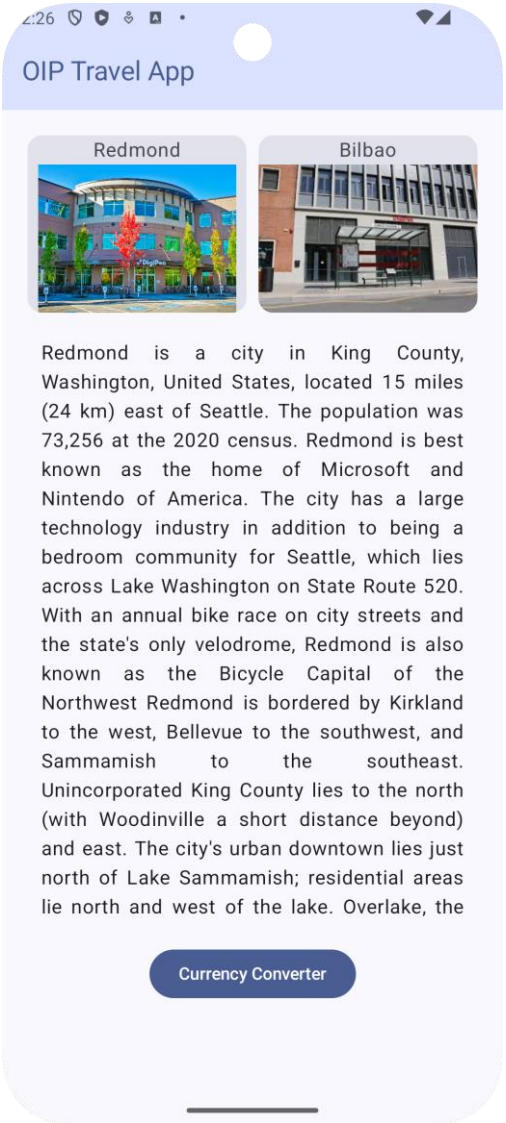
OIP Travel App: Navigation and TextField

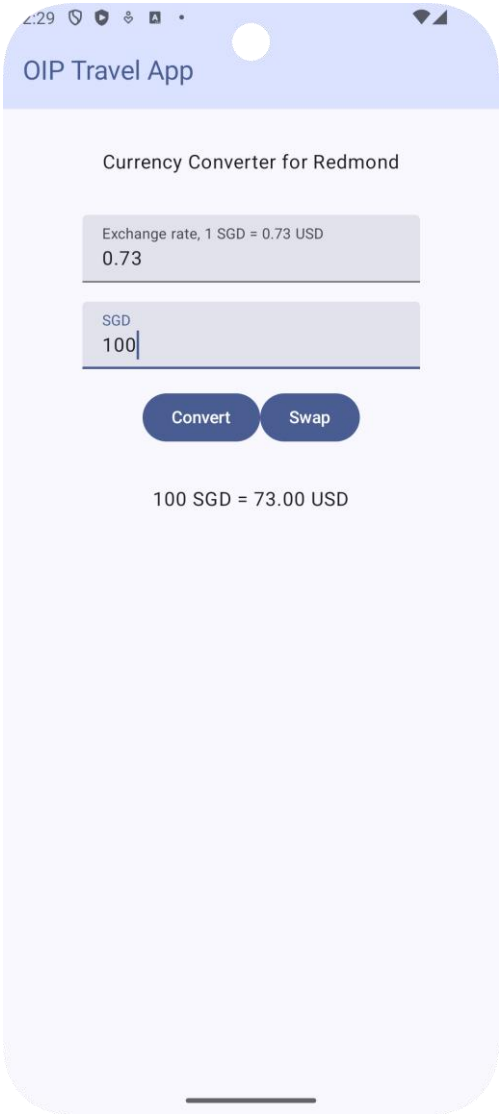
CSD3156 Mobile and Cloud Computing Spring 2025

Overview

This lab provides exercises and guidelines to gain familiarity with Navigation and TextField.

You are going to build an app to facilitate the OIP travel using Jetpack Compose with basic Compose Navigation and TextField.





Jetpack Compose Navigation

https://developer.android.com/codelabs/basic-android-kotlin-compose-navigation

https://developer.android.com/codelabs/jetpack-compose-navigation

A preferred way is using safe args:

https://developer.android.com/guide/navigation/design/type-safety

TextField

https://developer.android.com/develop/ui/compose/text/user-input

Outcomes

Upon completion of the session, you should be able to:

- Get familiar with Navigation and TextField
- More familiar with the basic UI elements and layout in Jetpack Compose

Description

The app consists of two main screens:

- First Screen: On this screen, you can select an OIP campus by tapping on it. Additionally, you can browse scrollable description texts for those two cities. Once you've chosen a campus, tap the "Currency Converter" button to proceed to the next screen, with the selected campus passed along.
- Second Screen: Here, you can enter a currency amount and the exchange rate to convert between SGD and the respective currency associated with the selected campus. You can press the "Swap" button to toggle the direction of conversion (SGD to/from the selected currency) and use the "Convert" button to display the conversion result. To return to the first screen, simply press the back button of the emulator/device.

Creating a basic navigation in Jetpack Compose

The goal of this exercise is to practice basic navigation and TextField in Jetpack Compose.

- Fork the repo csd3156-lab02-2025.
- The images are in R.drawable and texts are in R.string, they are in general very long texts.

- The app has only one Activity: **MainActivity**, and two screens.
- Note that the top part is a Row of 2 Columns (campuses) (testTag: **Redmond, Bilbao** respectively, use Modifier.testTag to set the testTag) with Image and Text. Ensure that each entire column is clickable including the Image and Text.
- The middle part is a Text that is vertically scrollable, you can use .*verticalScroll*(rememberScrollState()) to realize that.
- The bottom is a Button with text "**Currency Converter**".
- Hit the campus to select one, the default selection will be Redmond.
- On the second screen, the based on the selected campus, “Currency Converter for Redmond” or “Currency Converter for Bilbao” will be displayed and the currency would be **USD** or **EUR** respectively.
- There are two TextField (testTag: **ExchangeRate**, **InputValue** respectively) that you can enter exchange rate and input amount.
- The Buttons are “Convert” and “Swap”. Once hit “Convert”, a text of the conversion result is displayed as: **100 SGD = 73.00 USD,** if the input amount is 100 and exchange rate is 0.73, this the **default exchange direction, from SGD to USD (or EUR).** Please note the **format** and **spaces**, the input is an integer, and the output is rounded to two decimal places.
- Once hit “Swap” the exchange direction swaps, it becomes

  you can then re-input exchange rate/amount to convert.
- You need to recreate a UI similar to the screenshot above.

IMPORTANT: Ensure that the activity is named as MainActivity:

- The Campus Columns are with the testTag “Redmond”, “Bilbao”

- The TextFields are with the testTag “ExchangeRate”, “InputValue”

- The Button names are “Currency Converter”, “Convert”, “Swap”

- The currencies are SGD, USD and EUR, and the format is xx SGD = yy.zz USD

- Don’t change the build.gradle, setting, and Manifest files, otherwise you might not be able to use the test apk to test by yourself

Lab Exercise 2

Due Date: Wed, Jan 22 2025 2359 hrs

Fork the repo **csd3156-lab02-2025** and then clone it.

1. Implement the UI, UI logic and Navigation.
2. Test your program using UIInstrumentedTest and the test apk.
3. Commit and push all changes to your forked repository.

END OF DOCUMENT