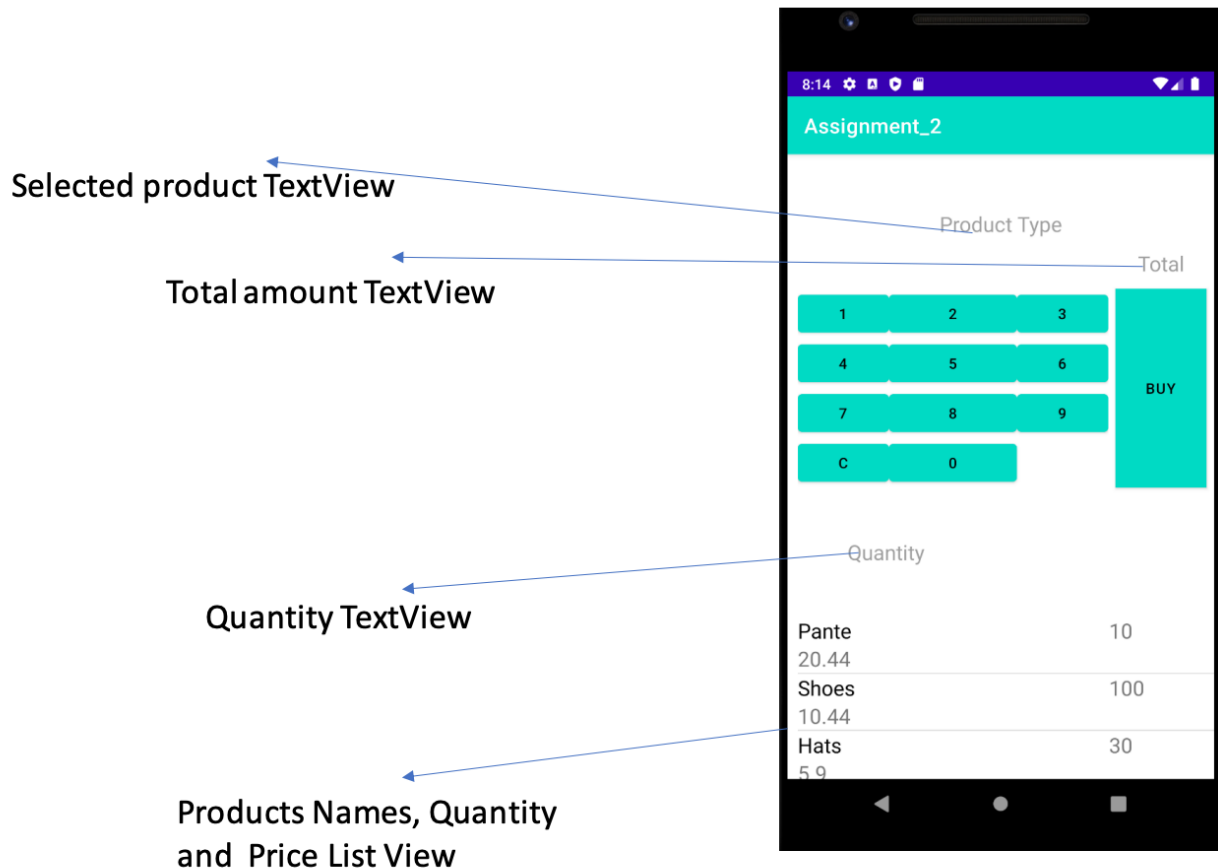


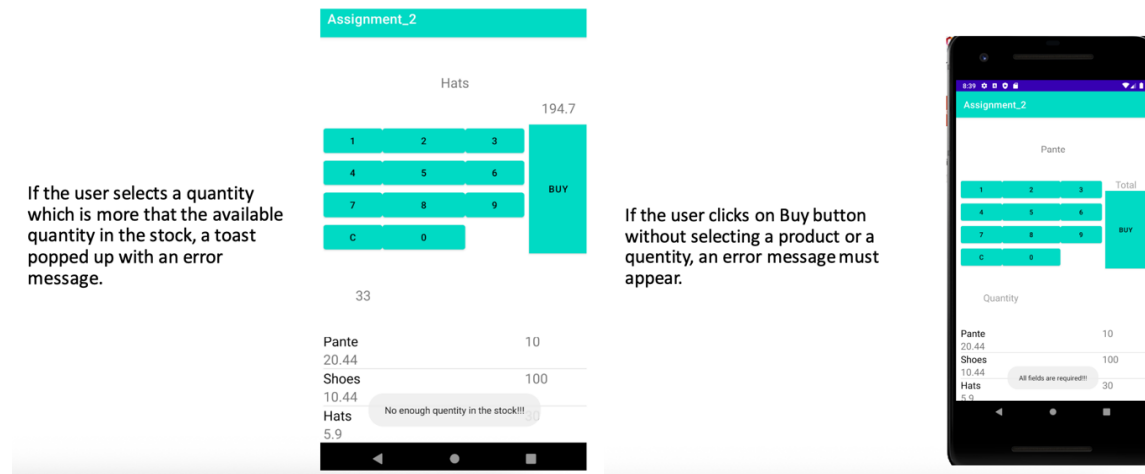
Assignment: Cash Register App Using Kotlin, Jetpack Compose, Navigation, and MVVM

In this assignment, you will create a small Cash Register application using Kotlin and Jetpack Compose. The purpose of the project is to help you practice building user interfaces with Compose, managing application state using the MVVM pattern, and implementing multi-screen navigation using Navigation-Compose. The application will consist of two main screens: a cash register screen where the user selects products and enters quantities, and a purchase history screen that displays all completed transactions.

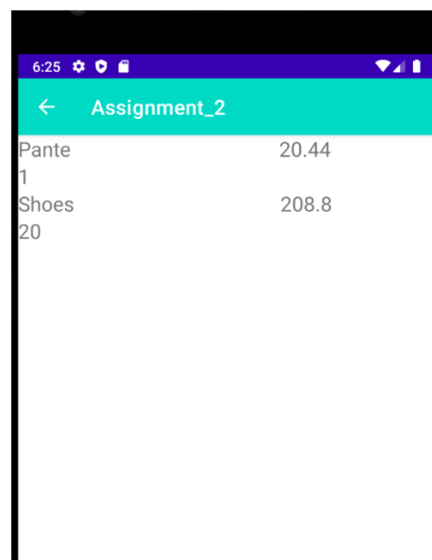


The first screen of the app is the Cash Register Screen. This screen must display a list of products, where each product includes a name, a price and a quantity. When the user taps on a product, that product becomes the currently selected item, and the screen must show the selected product's name and price. The user must be able to enter the quantity using an on-screen number pad implemented entirely in Jetpack Compose. The number pad should allow the user to enter digits from 0 to 9, as well as clear the current input. As the quantity changes, the screen should automatically calculate and display the total amount by multiplying the selected product's price by the entered quantity. Once the user confirms the purchase by tapping the Pay button, the app must create a new purchase record containing the product name, the quantity, and the total amount. This purchase must be added to a purchase history list maintained in the ViewModel.

After saving the purchase, the app must navigate to the second screen. (Don't forget to update the available quantity after each purchase)



The second screen is the Purchase History Screen. This screen must show a list of all completed purchases made by the user. Each entry in the list should include at a minimum the product name, the quantity purchased, and the total price. You may optionally include a timestamp indicating when the purchase was made. The list must update automatically every time the user completes a new payment on the cash register screen.



Your application must use a single-activity architecture, with Navigation-Compose managing the transition between the cash register screen and the history screen. You are required to structure your code using the MVVM pattern.

You will need to define a needed model for this app. The navigation graph must define at least two destinations, one for the cash register screen and one for the history screen, and the Pay button must trigger navigation to the history screen after storing the purchase.

This assignment is designed to help you understand state management in Compose, multi-screen navigation, UI interaction patterns, and the basics of structuring an app using the MVVM architecture.