**CMPS 312 Mobile Application Development – Fall 2021**

**Lab 6 - Navigation**

In this lab, you will practice:

* Interact with lists such as search, sort.
* Add Navigation UI components such as TopAppBar.
* Navigate between the app screens.

**Part A – List Interaction**

1. Use the provided model solution of Qatar2022 app we created in the previous lab.

Note that the json file and the app has been updated to add a new item seating capacity.

1. As shown in Figure 1, add a Top App Bar with Search Box to allow users to search stadiums.

Graphical user interface, application

Description automatically generated

Figure 1.Search stadiums

1. As shown in Figure 2, add a Dropdown list to allow filtering the stadiums based on their status (To be built, Major Renovation or All).

A picture containing website

Description automatically generated

Figure 2. Filter stadiums based on status

1. As shown in Figure 3, Create a sort menu that sorts the stadiums based on Seating capacity, and alphabetically based on city and stadium name.

|  |  |
| --- | --- |
| Graphical user interface, website  Description automatically generated | Graphical user interface, website  Description automatically generated |
| Stadiums are sorted by seating capacity | Stadiums are sorted by name |

Figure 3.Stadium sort options

**Part B – Navigation**

In this part we will create a Banking app wherein we will navigate from one screen to another and pass values between screens.

1. Create a new project place it under the **Lab6-BankingApp** folder on your GitHub repo.

* Name the project BankingApp
* Use cmps312.bankingapp as the package.
* Select *Empty Compose Activity* as the project template.

Add the following dependency in the module’s build.gradle:

implementation "androidx.navigation:navigation-compose:2.4.0-alpha09"

1. Create a composable Navigation function that hosts the *NavHost* and the *navController*
2. Create a sealed Screen class that contains objects for each of **HomeScreen**, **FundTransferScreen**, **DetailScreen** and **ConfirmationScreen**
3. Create composable for all the screens. The name of the composoble should be the same as the name of the object (of that screen) in the Screen class.
4. The NavGraphBuilder of the NavHost includes the composables of all the screens that we have in the application.
5. In the NavBackStackEntry of the composable we call the composable that we created in Step4
6. The HomeScreen UI will be as follows
7. The Check Balance navigates to the DetaislScreen
8. The CheckBalance navigates to the Details Screen Graphical user interface, text, application

   Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Figure 5. Home Screen

1. When Fund Transfer button is clicked it navigates to FundTransfer screen whose UI will be as Graphical user interface

   Description automatically generated

Figure 6.FundTransferScreen

1. After filling the details and On clicking the FundTransfer we will be navigated to a Confirmation page
2. On clicking the Fund Transfer button and navigating to the ConfirmationScreen, we pass all the parameters such as Account Number, Amount, Beneficiary Account Number, and Beneficiary Name to the ConfirmationScreen.
3. The parameters are passed as listOf(navArgument) in the composable of ConfirmationScreen.
4. Each navArgument includes a name, and includes a navArgumentBuilder that contains type and other required details.
5. In the NavBackStactEntry of ConfirmationScreen, we include entries for all the parameters that we will accept.
6. Add the parameters to the route of the Confirmation Screen.
7. route=Screen.ConfirmationScreen.route +"/{parameter1 }/{ parameter2 }/{ parameter3}/{parameter4}"
8. The confirmation page has been passed all the parameters from the FundTransfer page.
9. The confirm button raises a toast

Graphical user interface, application

Description automatically generated

Figure 7.ConfirmationScreen

1. The bottom navigation bar navigates between HomeScreen, FundTransferScreen and DetailsScreen
2. The navigation is as follows
3. Graphical user interface, application, Teams

   Description automatically generated Graphical user interface, application

   Description automatically generated Graphical user interface, application

   Description automatically generated