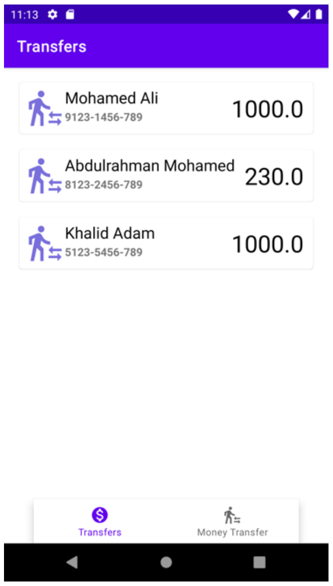
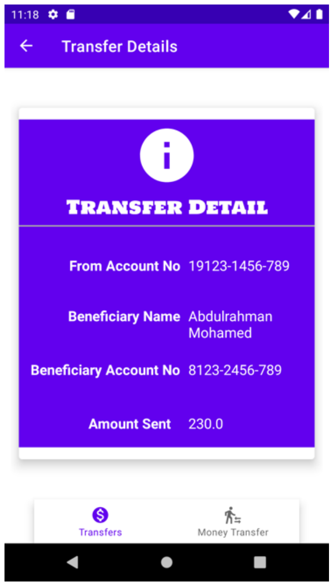
# CMPS 312 Mobile Application Development
# Lab 7 – Model-View-ViewModel (MVVM) Architecture

## Objective

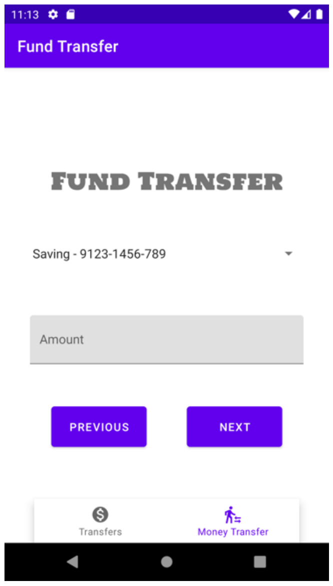
In this Lab, you will **continue building the Banking App** following MVVM Architecture. In particular, you will practice using ViewModel, LiveData and Data Binding to objects.

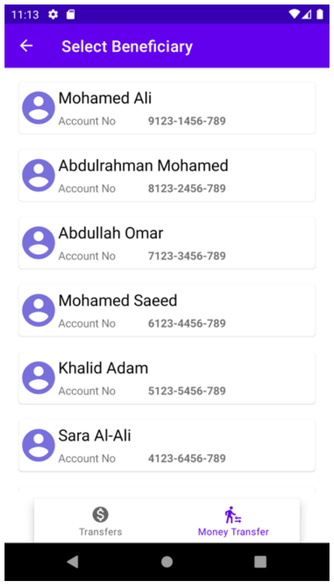


| | | |
|---|---|---|
| Mohamed Ali | 9123-1456-789 | 1000.0 |
| Abdulrahman Mohamed | 8123-2456-789 | 230.0 |
| Khalid Adam | 5123-5456-789 | 1000.0 |

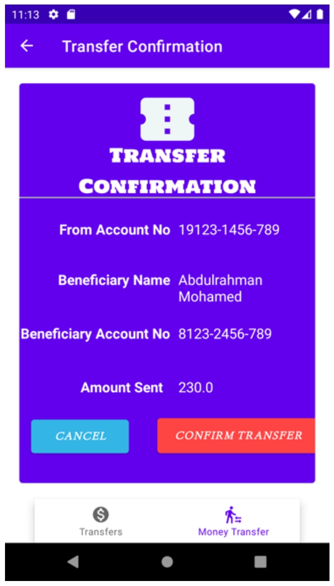


| | |
|---|---|
| From Account No | 19123-1456-789 |
| Beneficiary Name | Abdulrahman Mohamed |
| Beneficiary Account No | 8123-2456-789 |
| Amount Sent | 230.0 |





| | | |
|---|---|---|
| Mohamed Ali | Account No | 9123-1456-789 |
| Abdulrahman Mohamed | Account No | 8123-2456-789 |
| Abdullah Omar | Account No | 7123-3456-789 |
| Mohamed Saeed | Account No | 6123-4456-789 |
| Khalid Adam | Account No | 5123-5456-789 |
| Sara Al-Ali | Account No | 4123-6456-789 |



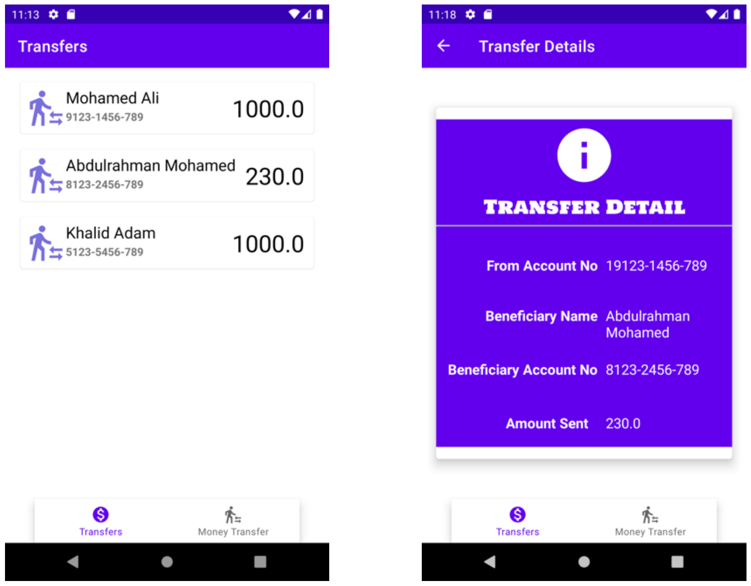
| | |
|---|---|
| From Account No | 19123-1456-789 |
| Beneficiary Name | Abdulrahman Mohamed |
| Beneficiary Account No | 8123-2456-789 |
| Amount Sent | 230.0 |

## Preparation

1. Sync the Lab GitHub repo and copy the **Lab7-MVVM** folder into your repository.
2. Open the project ***Bank App*** in Android Studio. The project has the following folders and files:
    - **layout**: the layouts for the fragments and the main activity.
    - **adapters**: two adapters, one for the beneficiaries list and one for transfers list.
    - **model:** Account, Transfer and Beneficiary.
    - **repository:** implements reading from accounts.json, transfers.json and beneficiaries.json.
    - **drawable:** images needed for this app.
    - **assets:** contains the json files
    - **navigation:** navigation Graph

## PART A: Implementing the Transfer List and Transfer Details

In Part A, your task is to implement the Transfer List and Transfer Details use case. As shown in Figure 1, when the user selects from the 'Transfers' menu option, the app should navigate to the transfer lists screen. When the user clicks a transfer from the list then the app should navigate to the transfer details.



| | |
|---|---|
| Mohamed Ali 9123-1456-789 | 1000.0 |
| Abdulrahman Mohamed 8123-2456-789 | 230.0 |
| Khalid Adam 5123-5456-789 | 1000.0 |

| | |
|---|---|
| From Account No | 19123-1456-789 |
| Beneficiary Name | Abdulrahman Mohamed |
| Beneficiary Account No | 8123-2456-789 |
| Amount Sent | 230.0 |

Figure 1. Transfer List and Transfer Details use case

1. Create a new package named viewmodel inside the ui.transfer package.
2. Implement **TransferViewModel** class to hold the list of transfers.
   - Inside the class declare a mutable live data object named **_transfers**

```
private var _transfers = MutableLiveData<MutableList<Transfer>>()
```

- Expose the _transfers mutable live data variable through a function called transfers().

```
val transfers: LiveData<List<Transfer>> = _transfers
```

- Initialize the _transfers using the BankingRepository object.

Tip: unit int function for initialization

3. Implement the TransferListFragment

   - Declare and instantiate a viewModel object of type TransferViewModel

   - Observe the viewModel.transfers LiveData and display the received data on the console. Test the app and check the displayed data.

   - Pass the received transfers list to the **TransferListAdapter** to display them on the RecyclerView.

4. Implement the RecyclerView adapter and connect with the TransferList RecyclerView. You need to use binding to connect the Transfer List Item layout with a transfer object.

   - To enable data binding add the following to app / build.gradle .

```
apply plugin: 'kotlin-kapt'
android {    ...
  buildTypes {  ...   }
  android.buildFeatures.dataBinding true
}
```

   - Modify **list_item_transfer** layout file and add the necessary data binding to display a transfer.

   - Implement **TransferListAdapter** to allow displaying transfers in the RecyclerView.

5. Implement display Transfer details:

- In TransferViewModel add a **selectedTransfer** object.
- When the user clicks a transfer on the transfers list recycler view, the event handler (i.e, displayDetails function) should assign the clicked transfer object to viewModel.selectedTransfer so that it will made available to the TransferDetailsFragment. Then the app should navigate to the TransferDetailsFragment.

  Note that the TransferListFragment and TransferDetailsFragment will communicate via the ViewModel.
- Do the necessary data binding needed for the **TransferDetailsFragment** to display the viewModel.selectedTransfer.

## PART B: Implementing the Transfer Component

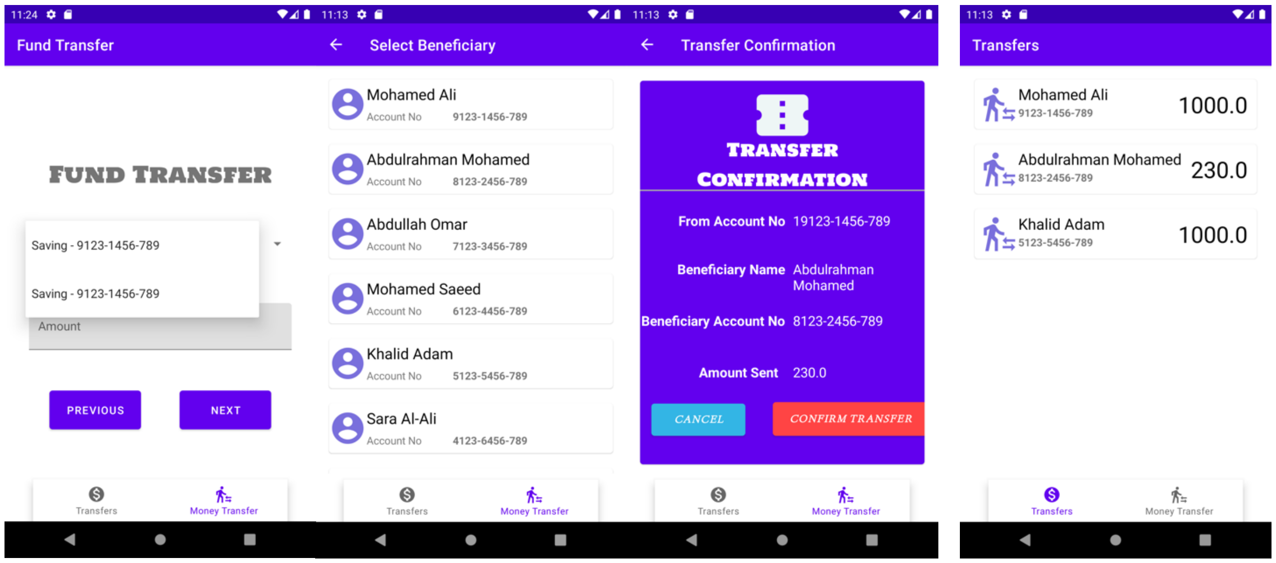
In PART B your task is to implement the *money transfer* use case. As shown in Figure 2, first the user selects the from account and specify the transfer amount. Then the user selects a beneficiary. Upon confirmation the transfer is added to the transfers list.

1. Implement the Fund Transfer Fragment. First fill the spinner with the accounts to allow the user to select the From Account.

   Add a **transfer** object to **TransferViewModel** to hold the transfer data needed for the three screens of the use case [Fund Transfer, Select Beneficiary, and Transfer Confirmation].

   When the user clicks next, store the fromAccount and amount in viewModel.transfer. Then navigate to Select Beneficiary.

2. Implement BeneficiaryFragment to allow the user to select a beneficiary.
   First, add BeneficiaryViewModel to provide the beneficiaries list to the BeneficiaryFragment.
   Implement the BeneficiaryAdapter and add the necessary binding to the layout to display the list of beneficiaries.
   When the selects a beneficiary, store the select beneficiary in transferViewModel.transfer. Then navigate to Confirm Transfer.
3. Implement ConfirmTransferFragment to display the transfer details and allow the user to confirm the transfer.
   Finally, when the confirms, add the transfer object to the list of transfers then navigate to the app home (i.e., TransferListFragment), make sure you pop off all loaded fragments from the backstack.



11:24

Fund Transfer

FUND TRANSFER

Saving - 9123-1456-789

Saving - 9123-1456-789

Amount

PREVIOUS NEXT

Transfers Money Transfer

11:13

Select Beneficiary

Mohamed Ali
Account No 9123-1456-789

Abdulrahman Mohamed
Account No 8123-2456-789

Abdullah Omar
Account No 7123-3456-789

Mohamed Saeed
Account No 6123-4456-789

Khalid Adam
Account No 5123-5456-789

Sara Al-Ali
Account No 4123-6456-789

Transfers Money Transfer

11:13

Transfer Confirmation

TRANSFER CONFIRMATION

From Account No 19123-1456-789

Beneficiary Name Abdulrahman Mohamed

Beneficiary Account No 8123-2456-789

Amount Sent 230.0

CANCEL CONFIRM TRANSFER

Transfers Money Transfer

11:13

Transfers

Mohamed Ali 9123-1456-789 1000.0

Abdulrahman Mohamed 8123-2456-789 230.0

Khalid Adam 5123-5456-789 1000.0

Transfers Money Transfer

Figure 2. Money Transfer use case