CMPS 312 Mobile App Development
Banking App
Lab Assignment 3
Deadline - Saturday October 24, 2020

Objective

The objective of this assignment is to practice building an android app that communicates with a Web API using retrofit and coroutines. You will also practice using the navigation component, View Models, Databinding and Live Data.

Overview

In the assignment, you will design and implement a **Bank App** that allows the bank employees to add, update, delete accounts. A web version of the app is available at https://employee-bank-app.herokuapp.com. Your task is to implement the mobile version of the web app that handles only the part that deals with accounts, see this prototype

Implementation Instructions

You should write the whole app by yourself no base solution will be provided. You can use the skills acquired in labs **6,7 and 8** to help you deliver the app.

1. Create a new project and name it "**Banking App**" and add all the needed dependencies. You can copy the last lab dependencies and they should be enough for you to complete this project.
2. Add one model classes [**Account**] that can store the following JSON objects that you will get from the server.

Account Object

```
{
  "acctType": "Current",
  "balance": 7,
  "accountNo": "1602887911406"
}
```

- Setup your app to get the data from the server. Below are the web APIs that you need to use to accomplish the functionalities of the app shown in the prototype figs [1,2,3]. You can **test the Web API using Postman** to better understand what each of the following URLs returns and how to call them. https://www.getpostman.com/
- Also you can see the working demo of the web app here https://employee-bank-app.herokuapp.com

HTTP Method	Url	Functionality
Get	/api/accounts/:id	Returns an account by id
Post	/api/accounts	Adds an account
Put	/api/accounts/:id	Updates an account
Delete	/api/accounts/:id	Deletes an account by id

3. You should use the retrofit library to communicate with the server. Make sure you create both the Service Interface and Repository Object kotlin files that will allow you to set up the retrofit library [Refer to Lab 8].
4. Create the app navigation graph, named **nav_graph**, having the **three destinations as** shown in the app **prototype** at figs [1,2,3]. You should at least have the following screens [**list of accounts**, **account details, [update account and add account].**
5. Design the layout for each of the destinations that you have created. The complete design is shown in the prototype images below. All of your layouts should use **data binning** whenever possible. **[Hint** or **Optional]**:
6. Implement the list account fragment and show all the list of accounts.
7. Implement the delete account. When the user clicks on the delete button, you should remove the account from both the list as well as from the server.
8. Implement the add and update account scenarios as shown in fig. 1 fig 3

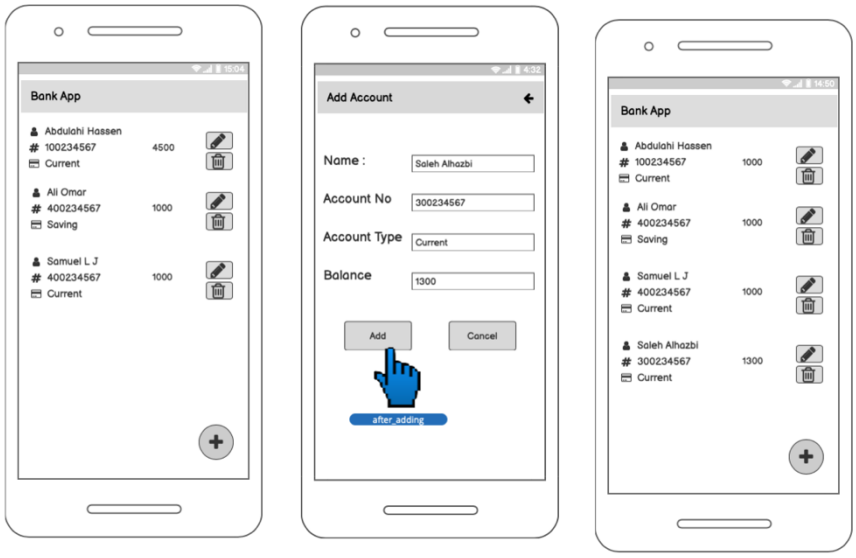

Figure 1 : Add Account

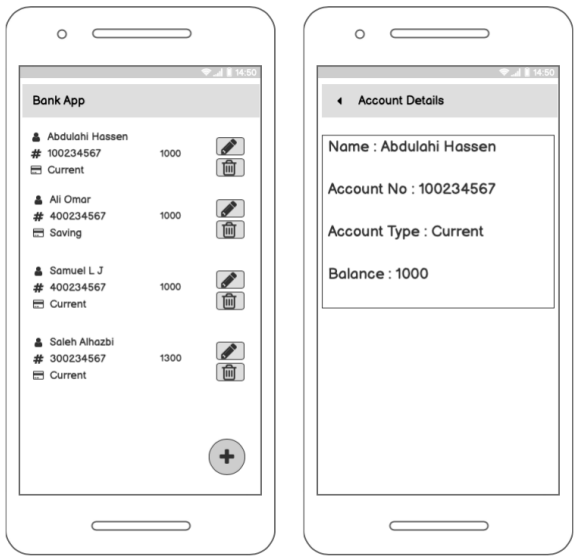

Figure 2 View Account Details

Figure 3 Update Account

Important It is expected that each student to have a **unique app** that accomplishes the tasks. You are free to use any theme/font/colour/icons that you would like to use. Also, discussing with your colleagues is allowed, however, please avoid sharing your code. If any plagiarism is detected then both parties will get zero.

Submit the testing sheet as well as the code under "**your_repository/assignments/assignment3**"