2022

Android Task Report

APPLICATION - RBCACCOUNTS

DHRUMIL AMISH SHAH

1. Introduction & Development Approach

The RBCAccounts application is developed with Kotlin & Java programming languages using Android Studio as Integrated Development Environment (IDE). It is developed using Model-View-ViewModel (MVVM) software architectural pattern, and uses RBCAccountLibrary.aar file provided in the assignment task questionnaire. It also uses Kotlin Coroutines and Splash Screen API offered by Android.

The project enclosing the developed application is structured in a way that every unit of the application follows the Separation of Concerns (SoC) principle. Additionally, the code quality of the application is given paramount importance as it follows the constraints listed below:

- ✓ Implemented principles of cohesion and coupling.
- ✓ Developed a clean, readable, structured, and maintainable codebase.
- ✓ Implementation using design patterns.
- ✓ Refactored the code to reduce code smells.
- ✓ Followed consistent naming convention throughout the project.

2. Acceptance Criteria

The task demands the following criteria to be met:

2.1 Use the library to build the app:

- ✓ I have cloned the library from the remote repository to my local system and stored it as RBCAccountLibrary.aar under the app/libs folder of the project.

2.2 Display the list of accounts, grouped by type (display name, account number, and balance).

- ✓ The first screen followed by the splash screen, on the launch of the application, displays the list of accounts following the required constraints. The data from the given library has been sorted by type and displayed using RecyclerView and Adapter.

2.3 Each account is clickable to display details that include transactions.

- ✓ This task is achieved using the Interface pattern that helps to connect Activity and Adapter together. On click of any account, the user is routed to the transactions page for that account.

2.4 The account details the page will display the current account name, account number and account balance along with all the transactions for that account sorted by newest first.

- ✓ The account name, account number, and account balance is passed from account activity to transactions activity using Intent.
- ✓ Multiple APIs are parallelly invoked using async/await and Kotlin coroutines.
- ✓ The transactions are grouped by date and sorted in descending order using Kotlin functions.

3. Grading Criteria & Requirements

The application is developed using Kotlin & Java, however, to demonstrate my ability to program in both languages I have provided the Kotlin version of all the Java files. I have not used any third-party libraries. The project is solely dependent on Android SDK. The implementation follows software design architecture, design patterns, and OOPC. The application's UI is user-friendly. The UI when viewed in portrait or landscape mode will show responsiveness without any glitches. Additionally, I have also provided Dark Theme support and a Splash Screen for a rich user experience.

4. GitHub Link

The code for the application is available at link and the video demonstration of the application is available at link.

5. Application User Interface

Below are the images that show the application's user interfaces:

5.1 Splash Screen

Figure 1 - Splash Screen

5.2 Account and Transaction Activities

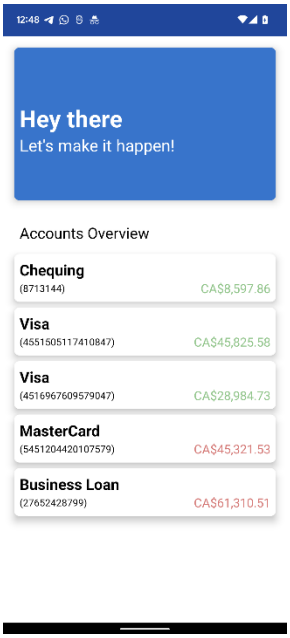

Figure 2 - Accounts Activity (Light Theme)

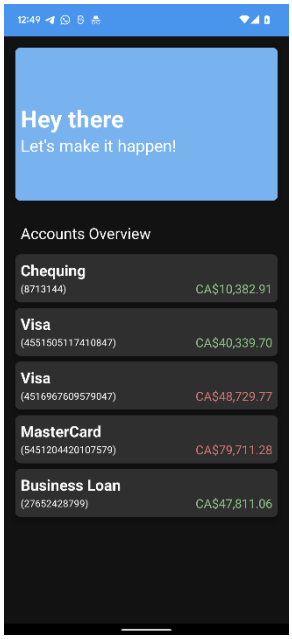

Figure 3 - Accounts Activity (Dark Theme)

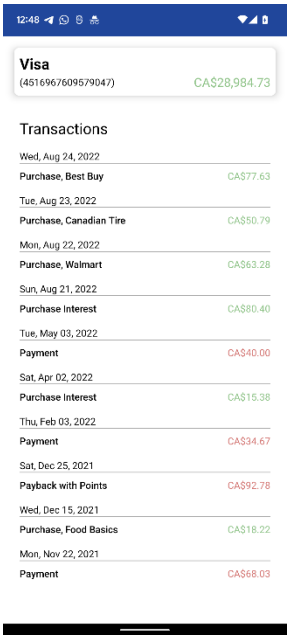

Figure 4 - Transactions Activity (Light Theme)

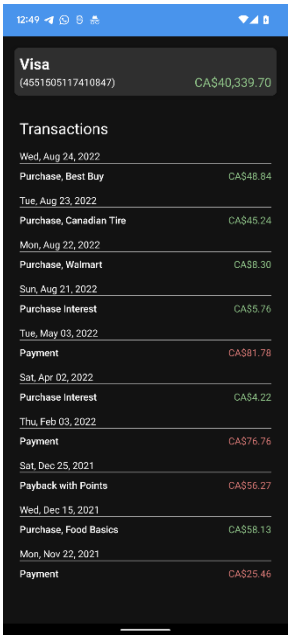

Figure 5 - Transactions Activity (Dark Theme)

5.3 Differrent States of Transaction Activity

Figure 6 - Loading Transactions (Light Theme)

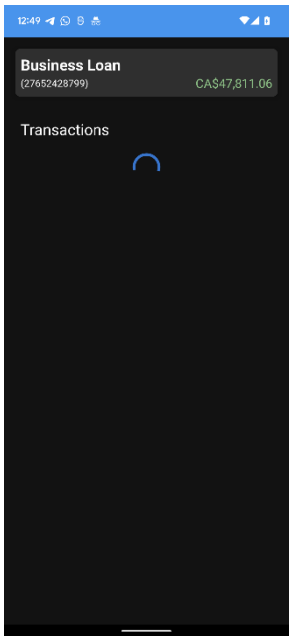

Figure 7 - Loading Transactions (Dark Theme)

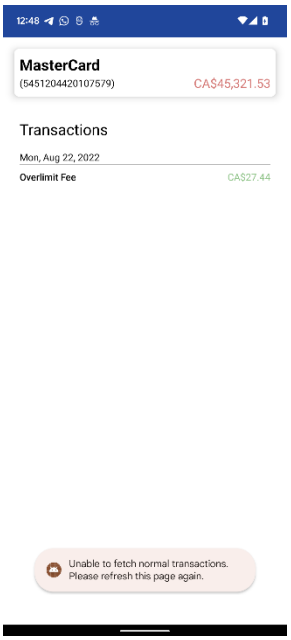

Figure 8 - Error Message (Light Theme)

Figure 9 - Error Message (Dark Theme)