Group 11

Money Drive

YAU Yat Fung, 20371106, yfyau@connect.ust.hk

Introduction

Financial management is very important. Especially in Hong Kong, it is not only used to saving money for buying those expensive houses. And in somehow, it also relates with people's life. We can see people ended their own life time by time because of they were facing some financial difficulties. Therefore, my application idea is a money manager which can remind users pay for their credit card's bill and build up good financial habits. Moreover, the data of the money manager can be shared between a group of people which acts like a money manager version of Google Drive. As the result, I call it "Money Drive".

Design

As Money Drive was planned to have a data sharing function. That means we need to divide it into two parts, client and server. Client side is for the core functions such as Bookkeeping, Reminder and Summary functions. Server side is for the Data sharing.

The Requirements of Client side:

- Input records
- Output some statistics
- Output some reminders
- Log on to server

The Requirements of Server side:

- Receive the records sent by Clients
- Send the records to the Clients with authorization

For meeting the requirements of Client side, Money Drive should have the functions below:

Base Bookkeeping Functions

✧ A calendar as a UI for users input their daily expenditure

Reminder Function

✧ A reminder for remind users to fill in their expenditures

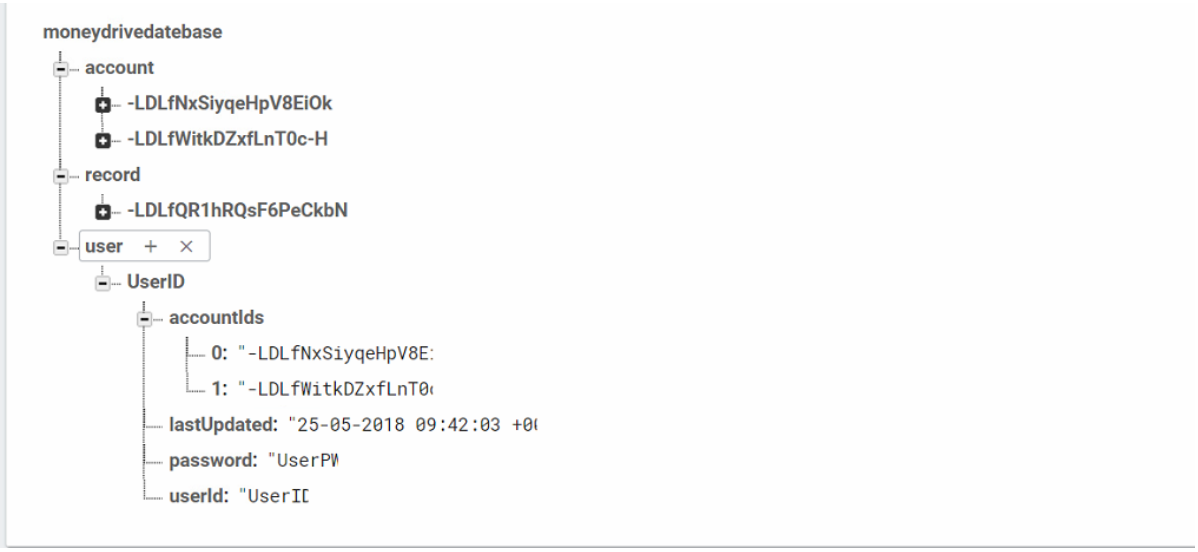
Summary Function

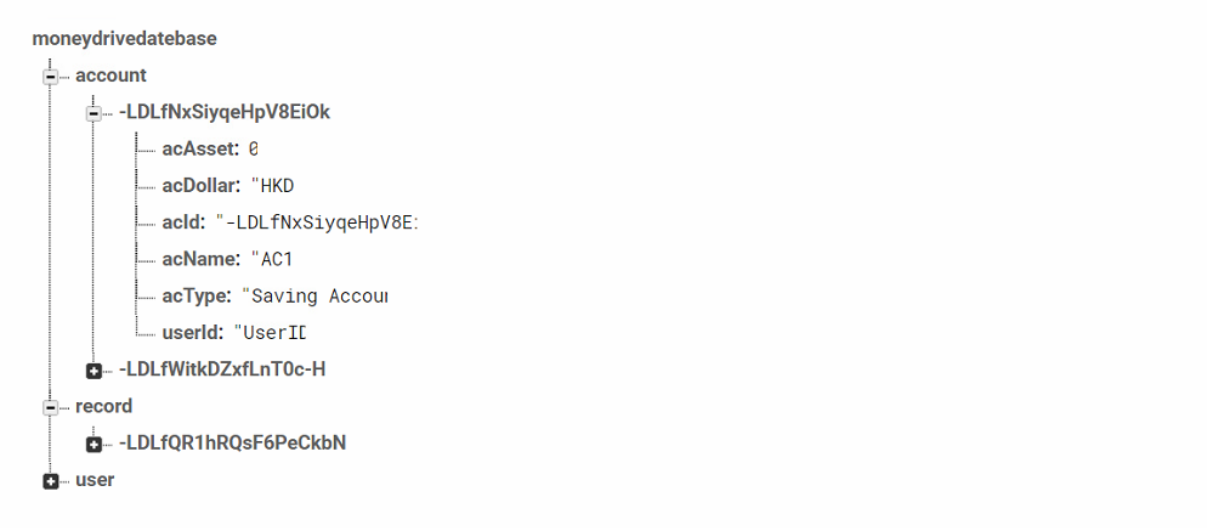
✧ A financial summary for users view their expenses with Charts

Data Sharing Function

✧ A connection between client and server for upload and download data

For the Database. We choose to use Firebase Database as Server side.
Here is the Data Structure:



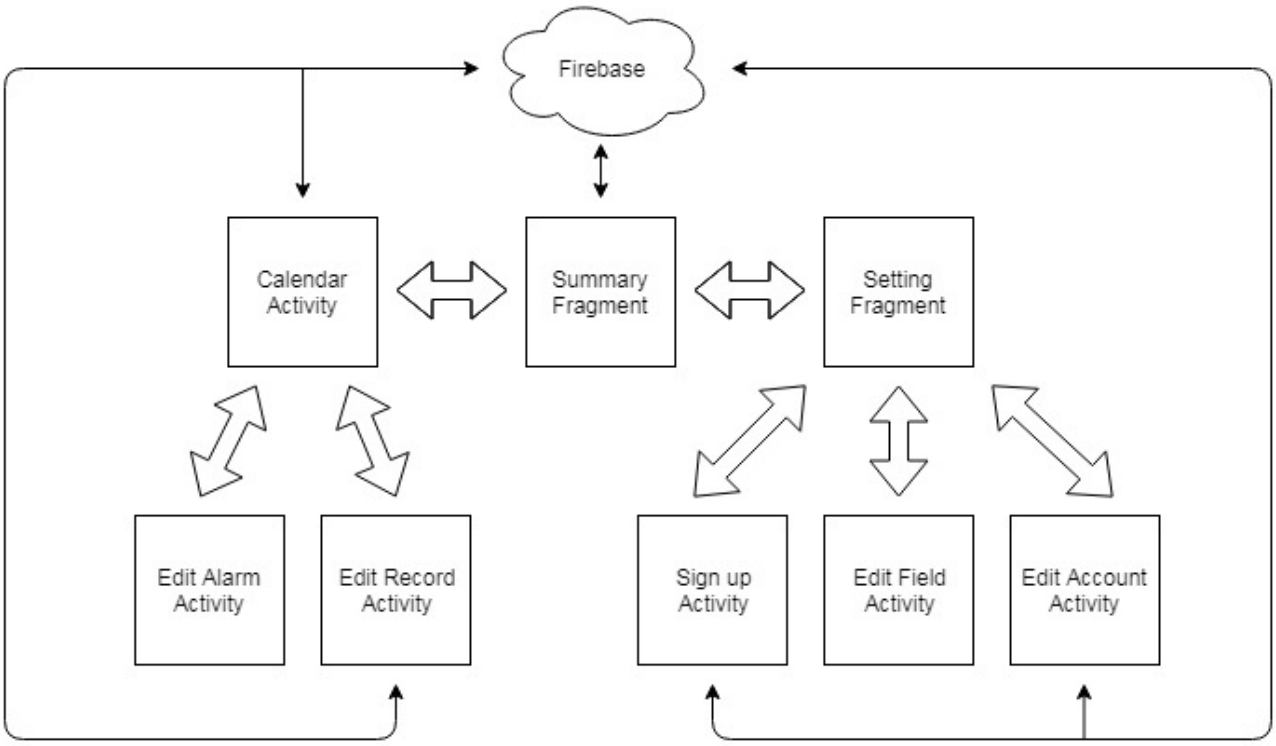


When a new user starts to use Money Drive, he needs to sign up for a user account first. User needs to enter a unique user ID as the key in Firebase for classify the client side. Also, there is a list of account IDs which are the user owned. Therefore, the User table is used for storing the unique user ID and password for log in and after user logged in we would get the list of account IDs for farther usage. As Firebase would automatically having a feedback when using OnDataChange() Method, the lastUpdated is not used at the end.

After user logged in and get the list of account IDs, user can access the accounts he owned by using the account IDs. For each user, there are many bank accounts under them. For example, Hang Seng bank and Bank of China. The Account table is used to storing the unique ID, the user id, the name, the type (Cash, Saving Account or Credit card), the dollar type (HKD, RMB, USD...) and the asset. Other variables are used for some future update function such like administration.

Record is under an account which have a unique id, the account id, the field Name, the date, the dollar type, the amount, the description. When user create a new record, user needs to input the information of the record and the record would be saved at Firebase.

Here are the Activities going to be used:



1. CalendarActivity
   - ➔ Provide a Calendar as an input interface for user to choose a day to add, edit or remove record and reminder
   - ➔ Show the daily balance and records when user click on a day
2. EditRecordActivity
   - ➔ Provide an input interface for user to add / edit a record
3. EditAlarmActivity
   - ➔ Provide an input interface for user to add an alarm
4. SummaryFragment
   - ➔ Provide a statistic graph for user review their record
5. SettingFragment
   - ➔ Provide an input interface for user to change their setting such as their user account

Implementation

For the better exterior of Money Drive, three libraries are imported. Firstly, Caldroid is a fragment that display calendar with dates in a month. It can help me implement the Calendar easier and I can change the style of the date cell to fit Money Drive App Theme. Secondly, BottomBar is a custom view component that mimics the new Material Design Bottom Navigation pattern. It helped me build the bottom bar easily that only need to change the xml and the icon. Thirdly, MPAndroidChart is a powerful Android chart view / graph view library. I used it for the Bar Chart and Pie Chart in the summary fragment.

For the data storage, I used SharedPreferences for local and Firebase for online. Local stored the Logged in user ID and the User defined Fields which I don't want to put it on Firebase as it may cause a lot of duplicated data. Firebase stored the data that show above.

For the Firebase, I used Interface for making a call back when the Firebase completed its work. It is because I want to make Firebase's onDataChange() inside a function such like addRecord() but things go trouble when it counted as inner class. It became more user friendly when it returns by a call back.

For the alarm, I used AlarmManager for setting the timer and BroadcastReceiver for notification.

Testing and Evaluation

Mainly debug by using Logcat to display some variables to check for the accurate and its error detection is powerful enough. For Firebase, I used firebase console to check the data.

Conclusions

As what I expected the main structure of Money Drive is completed. Base Bookkeeping Functions, Reminder Function and Summary Function are completed, although may be not powerful enough. And records can be shared when users imported a same account. Fortunately, data can be updated automatically easily by using Firebase onDataChange Method. I hope users will enjoy using Money Drive and Money Drive can be more powerful for improvements in future.

Shortcoming:

1. The amount stored in Firebase should be shift into cent to avoid Firebase automatically correct double to long.
2. The security problem of filter date from Firebase first would receive other users' data. However, the listener will be replaced when filter multiple accounts that affect the automatically update. Should try to use JavaScript for getting the data.
3. The Firebase call back method should increase the error protection as sometime user may already change out of the call back's view. For example, calendar activity and summary fragment both would get the data from Firebase when it is created. If user switching them fast and frequency. Error will be occurred which I think it is because the call back on summary fragment call back and set the value of charts, but the view already changed to the calendar activity.

What I Learned:

1. Base Knowledge of Android App Development
2. Usage of Firebase, which I never heard it before
3. Read for other libraries' documents
4. Importance of a well-prepared design as I change the database from MySQL to Firebase and the data structure and filter of these two databases are different that make a few troubles to me.
5. Think twice before running the code, the installation of APK is slow
6. Thing update frequently, Programmer need to keep learning new version items
7. Interesting experience for building whole project include database by myself

References

1. Caldroid
   https://github.com/roomorama/Caldroid
2. BottomBar
   https://github.com/roughike/BottomBar
3. MPAndroidChart API
   https://github.com/PhilJay/MPAndroidChart
4. Firebase
   https://firebase.google.com/
5. Google, My Best Friend
   https://www.google.com.hk/