Expenses

GitHub Username: aayusharyan

Expenses

Description

Expenses - Manage your money...

This APP will help you manage your expenses and funds. This will give an overview of all the transactions that are being made. Also this will help to analyse where the expenses are being made. Since the Transactions are grouped, it is easier to understand an overview on which area is having most traction.

This APP is a starter for the final Concept of automatically analysing transactions. The end goal is to provide an overview of Finances without user entering each transaction.

Intended User

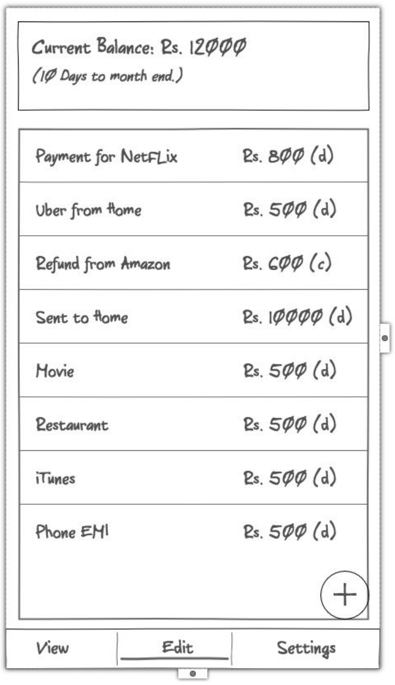
This APP is intended for those who are not able to manage their funds properly and by end of the month, all their money is spent and there is Zero Savings. The user can be a college student as well as a person with proper income.

Features

- Group Types of Transactions together.
- Keep transactions (Such as Cheques, Bills) as deferred.
- Make Recurring Transactions.
- Get an overall view of groups of Transactions.
- Export to CSV for further use *(Still Researching on Export functionality with ROOM)*.
- Log in to take backups and keep data.
- Has a companion Widget which shows Balance on Home Screen.
- App will be written on Java.

User Interface Mocks

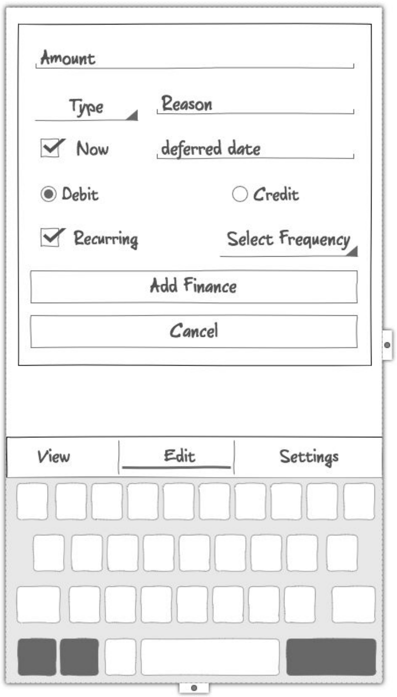
Main Activity


Current Balance: Rs. 12000
(10 Days to month end.)

Payment for NetFLix	Rs. 800 (d)
Uber from Home	Rs. 500 (d)
Refund from Amazon	Rs. 600 (c)
Sent to Home	Rs. 10000 (d)
Movie	Rs. 500 (d)
Restaurant	Rs. 500 (d)
iTunes	Rs. 500 (d)
Phone EMI	Rs. 500 (d)

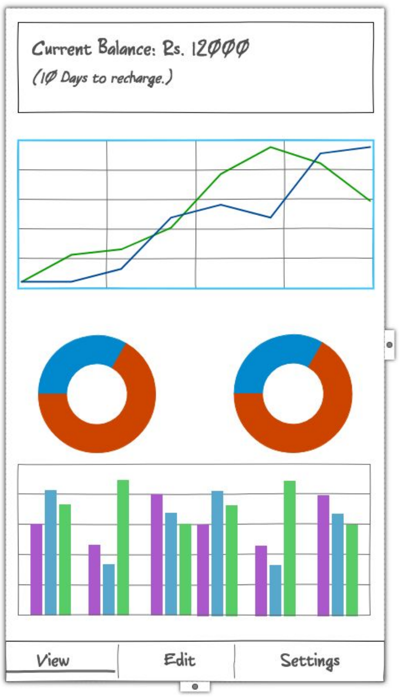
This is the Main Activity user will see on APP Launch. It will contain an overview of available funds and list of Transactions.

Edit Transaction Activity



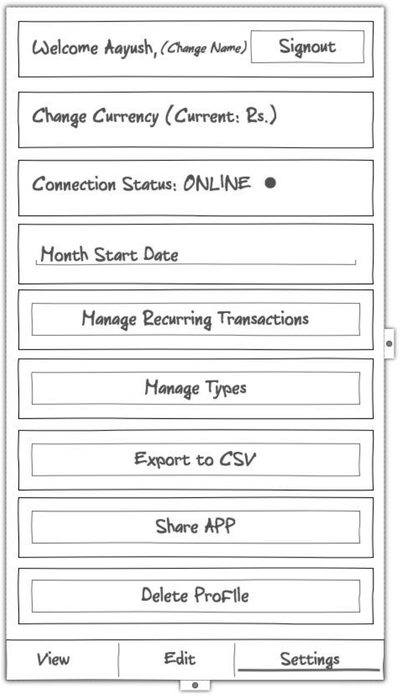
This is the Edit (or Insert) Transaction Activity. Here, the user can edit/insert transaction.

View Transactions Activity


Current Balance: Rs. 12000
(10 Days to recharge.)

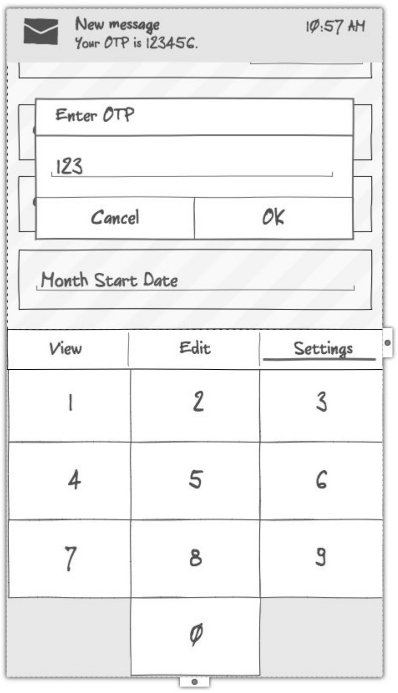
This is the View Transactions Activity, here the User will see an Overview of transactions.

Settings Activity


Welcome Aayush, (Change Name)
Signout
Change Currency (Current: Rs.)
Connection Status: ONLINE ●
Month Start Date
Manage Recurring Transactions
Manage Types
Export to CSV
Share APP
Delete Profile
View
Edit
Settings

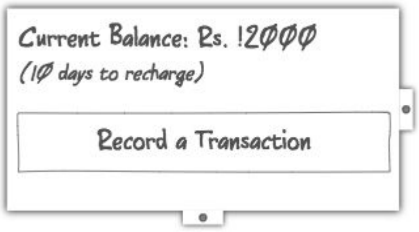
This is the Settings Activity. Here user can perform various actions including signin/signout.

Login Popup View


New message
Your OTP is 123456.
10:57 AM

This is how user will signin… This will use Mobile Number + OTP Method using Firebase Auth.

Helper Widget


Current Balance: Rs. 12000
(10 days to recharge)

This is the helper widget that will be shown on the User's home Screen, It will contain intent to launch the Edit Transaction Activity on Insert Mode to record a transaction.

Key Considerations

How will your app handle data persistence?

Data Persistence will be handled using ROOM Persistence Library. It will work in sync with Firebase to update data in the Cloud Firestore also.

Describe any edge or corner cases in the UX.

When there is no internet connection, and the use tries to sign-in, then the APP might become unresponsive, need to display a Toast.

When exporting to CSV and the amount of data is very large, the APP might become unresponsive, need to handle that export function in another thread and display Progress in Main Thread.

Describe any libraries you'll be using and share your reasoning for including them.

ButterKnife (8.8.1)	For Binding Views
Room (27.1.x) (stable)	For Data Storage
FCM (27.1.x) (stable)	For Notifications
Cloud Firestore (27.1.x) (stable)	To Keep data in the Cloud
Firebase Auth (27.1.x) (stable)	To Sign in the User
AnyChart (stable) (or Equivalent)	To display charts
OpenCSV / FastCSV (stable) (or Equivalent)	To Export the database in CSV format.

The APP will be Build on Android Studio (3.1.x stable) with help of Gradle (3.1.0 stable)

Describe how you will implement Google Play Services or other external services.

I plan to use the following services in their Respective Ways:

Firebase Cloud Messaging	To send Targeted Notifications
Analytics	To check for APP Performances
Cloud Firestore	To store user Data with their account
Firebase Auth	To Login user on the APP.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Configure libraries.
- Setup Cloud Firestore.
- Setup User Identification rules in Firebase.
- Setup ROOM DAOs.
- Setup Analytics on Firebase.
- Setup Resources (Such as Strings, Styles, Values, API Keys, etc. to XML files)

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main Activity.
- Build UI for View Activity.
- Build UI for Edit Activity.
- Build UI for Settings Activity.
- Build UI for Widget.

Task 3: Create Layout and Basic UI

- Create layout for Main Activity.
- Setup RecyclerView on the Main Activity.
- Setup Charts on the View Activity.
- Create Layout for the Settings Activity.
- Create App Bar and Toolbar for All Activities.

Task 4: Dive Deep in UI and Logic

- Link the RecyclerView with ROOM DB.
- Integrate ROOM with Charts.
- Create Layout for the Edit/insert Screen.
- Implement Currency Logic on Settings Screen.
- Implement Manage Types on Settings Screen.
- Implement Share APP Intent on Settings Screen.
- Implement logic on the Edit Screen

Task 5: Connect with Cloud

- Implement User signin / signout on the Settings Screen.
- Implement Firebase Auth.
- Implement Firebase Analytics.
- Implement Firebase to check for internet connection Status.
- Implement Cloud Firestore to backup the Data.
- Implement Delete Profile option on Settings Screen.

Task 6: Create Widget

- Create Widget Layout.
- Integrate Widget with ROOM DB.
- Implement intent to enter transaction.

Task 7: Add Expense Tracker

- Implement Recurring type.
- Implement deferred Payments.
- Implement Manage Recurring types on Settings screen.
- Implement Manage Start date logic on Settings screen.

Task 8: Add Export to CSV

Not sure whether I will be able to implement this or not… (Keeping this Task as Buffer)

- Implement Export to CSV from ROOM DB.

Task 9: Check for Responsiveness and implement Support

- Check the APP on Various layout and emulator environments.
- Check the Widget for various size combinations.
- Implement RTL Support by implementing start/end rules in all Layouts.
- If Margins/Paddings (or any other types of Vectors) have been implemented using Java, then fix that to support RTL.
- Add Accessibility support using Content Descriptors. (Referring to this link).

Task 10: Testing the APP

Testing for Incorrect inputs, Example:

- Negative value in amount.
- Past date in deferred payment.
- Invalid Number/OTP for user auth.
- Trying SQL Injection on Inputs.
- Testing for Accessibility support.
- Testing for RTL support.