Capstone Stage 1

GitHub Username: https://github.com/kristijanSpirkoski

Dough

Description

It is a money manager app. It helps users keep track of their balance, transactions and paid subscriptions in a simple and elegant way. It automates the process of keeping track of monthly income/payments.
All libraries will be stable
The app will be written in Java. It uses Gradle version 4.0.1, and Google Services version 4.3.3. Minimum Android API for the app: 21.

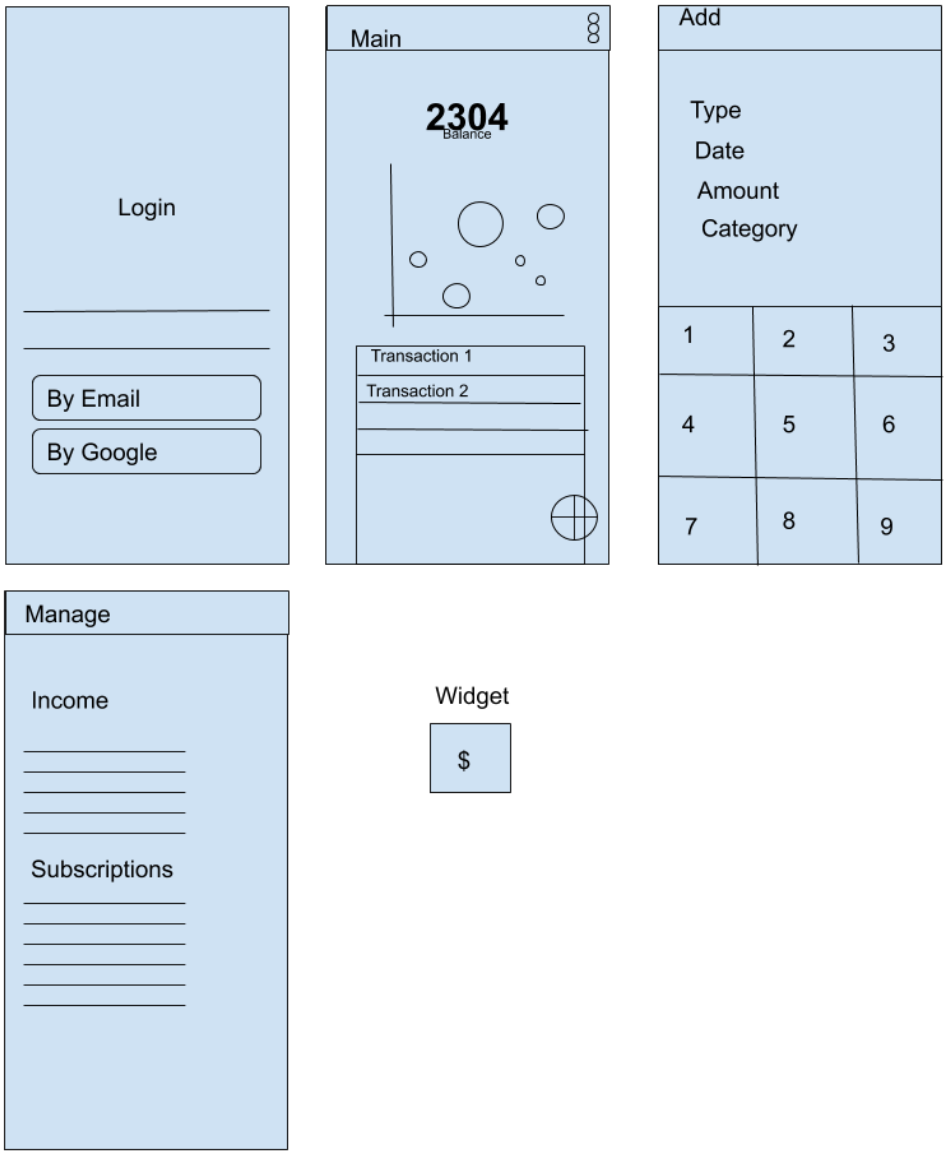
Intended User

There is no particular demographic, it would be beneficial to anyone who would like to organize their spending a bit more: from students, travellers, to the elderly, or even little kids with a weekly allowance.

Features

- Log in as a user on the app
- Keeps and displays history of transactions(ex. Backpack purchase for $50)
- Stores current balance
- Keeps track of subscriptions and updates balance accordingly(ex. Spotify charged 15th of each month)
- Keeps track of income and updates balance accordingly(ex. Paycheck on 25th of each month)
- Provides a scatter plot for better visualization
- Provides a widget to quickly add a new transaction

User Interface Mocks



1) Login page with option to register, log in with email or google account, provided with firebase authentication.
2) Main page where the user can look at his current balance, most recent transactions in a recycler/list view, and a scatter plot for better visualization of his monthly activity(transactions including spending for individual purchases, monthly subscriptions, income through salary...). A fab which starts a new activity to add a transaction. An options menu which has two options: Manage(starts the activity), and Log out(logs out user and starts Login Activity)
3) Add activity:

- Type is a dropdown menu with following options: Expense, Income, Periodic Income, Subscription
- Date of expense or monthly/weekly.. charge date
- Amount in desired currency(dollar, euro)
- Category: specific to type of transaction(ex. Salary, food, rent...) with the option of the user to add his own. This will be a new fragment which is inflated on the screen

The screen appears quite empty on the mock, however it will be filled up with input fields(ex. Numbers to enter amount, or categories to choose type of expense)

4) Manage page which allows the user to edit, or remove his saved periodic transactions(salaries, monthly subscriptions...)

Key Considerations

Adding extra to previous section:

- RTL will be handled by the change listeners from the Firebase Realtime Database
- Strings will be stored in the strings.xml resource file, and some specific ones in classes like FirebaseConstants(for remembering reference points of the DB)
- Dimensions will be specified in the views themselves, and in the dimen.xml resource folder for ones which appear multiple times
- The widget will open up the Add Transaction screen
- App will use content description on appropriate views for accessibility
- App has an AppBar/Toolbar

How will your app handle data persistence?

I will use a Firebase Realtime Database

Describe any edge or corner cases in the UX.

- If a user is already logged in, and has asked to be remembered, the Main Activity will start immediately instead of the Log In Activity
- After adding a new transaction, the user is brought back to the Main Activity with the updated information there

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

I will use a library for my scatter plot graph(still have not decided on which one between these, it depends on which one is better for what I want in the app during implementation):

- https://github.com/lecho/hellocharts-android
- https://github.com/PhilJay/MPAndroidChart
- https://github.com/halfhp/androidplot/blob/master/docs/quickstart.md/
- https://jaxenter.com/effort-free-graphs-on-android-with-achartengine-105649.html

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

I will use a Firebase Realtime Database and Firebase Authorization

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 graphing library
Create a Firebase Project

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for Log In, Add, Manage Activity
- Build Fragments for Add Activity

Task 3: Implement Firebase

- Realtime Database: add/update/remove transaction
- Authentication

Task 4: Implement Scatter Plot

- Research library
- Add minimal data

Task 5: Implement widget

- Open the Add Activity on the widget click