

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1 - Phone Main Screen](#)

[Screen 2 - Phone - Add Transaction](#)

[Screen 3 - Tablet - Main Screen](#)

[Screen 4 - Tablet - Add Transaction Screen](#)

[Screen 5 - Widget](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Content Provider](#)

[Task 3: Main Activity](#)

[Task 4: Add Transaction Activity](#)

[Task 4: Add Widget](#)

[Task 5: Prepare to release](#)

GitHub Username: [pedrolopesme](#)

MoneyLog

Description

With the app MoneyLog you can manage and track your expenses. At any time, you can check all your money transactions ordered by creation date and check your finance situation.

Intended User

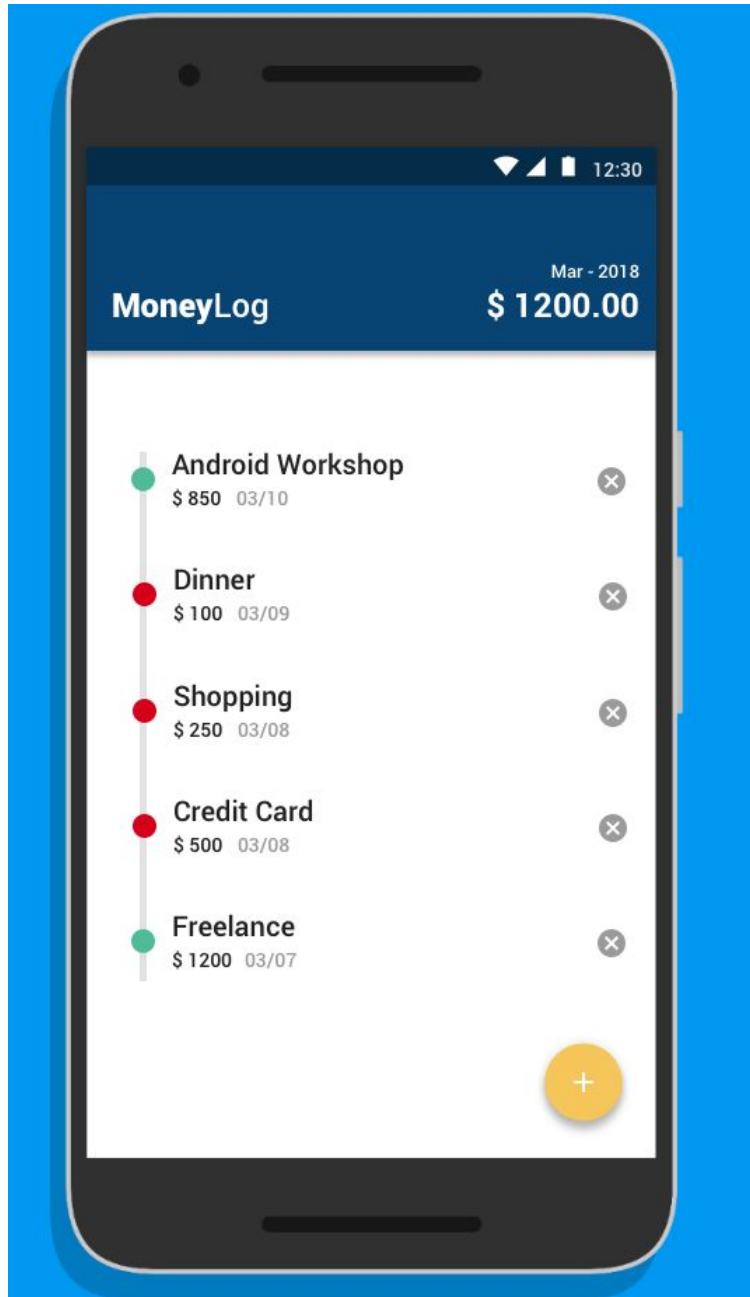
MoneyLog App comes to help people who need to track their expenses on a daily basis.

Features

- Main screen show the user's expenses ordered by date of creation. Also, how much money the user has based on his transactions
- An easy way to register new transactions, categorizing as Debt or Income
- An Widget to show the user's financial situation
- The user will be notified via Push Notification whenever his account turns to Negative or Positive

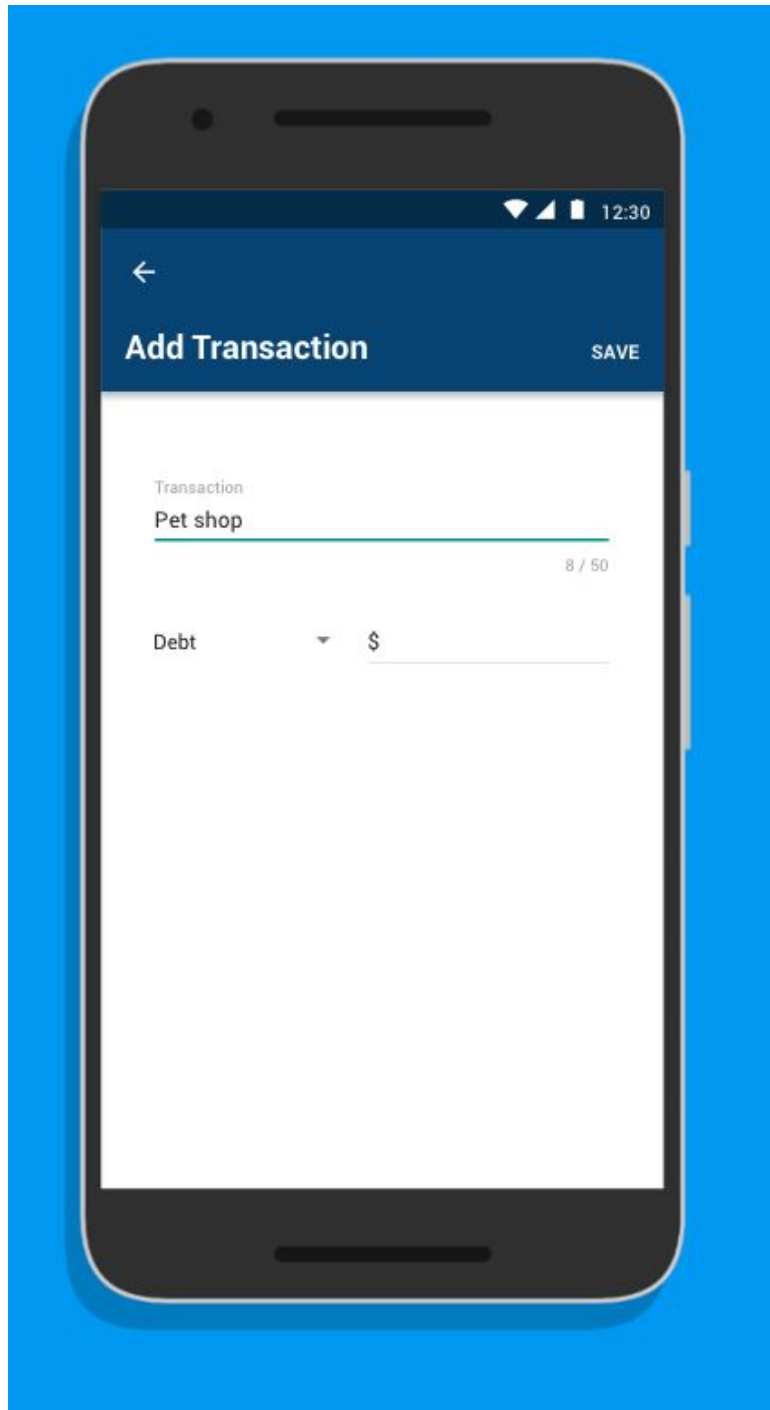
User Interface Mocks

Screen 1 - Phone Main Screen



User's transactions history and his account summary.

Screen 2 - Phone - Add Transaction



The image shows a mobile application interface for adding a transaction. The screen is displayed on a smartphone with a black bezel and a blue background. The status bar at the top shows the time as 12:30 and signal icons. The app's header is a dark blue bar with a back arrow on the left, the title "Add Transaction" in the center, and a "SAVE" button on the right. Below the header, the form has a "Transaction" label followed by a text input field containing "Pet shop" and a character count "8 / 50". Below this is a "Debt" label with a dropdown arrow and a dollar sign "\$" followed by an empty input field.

Transaction

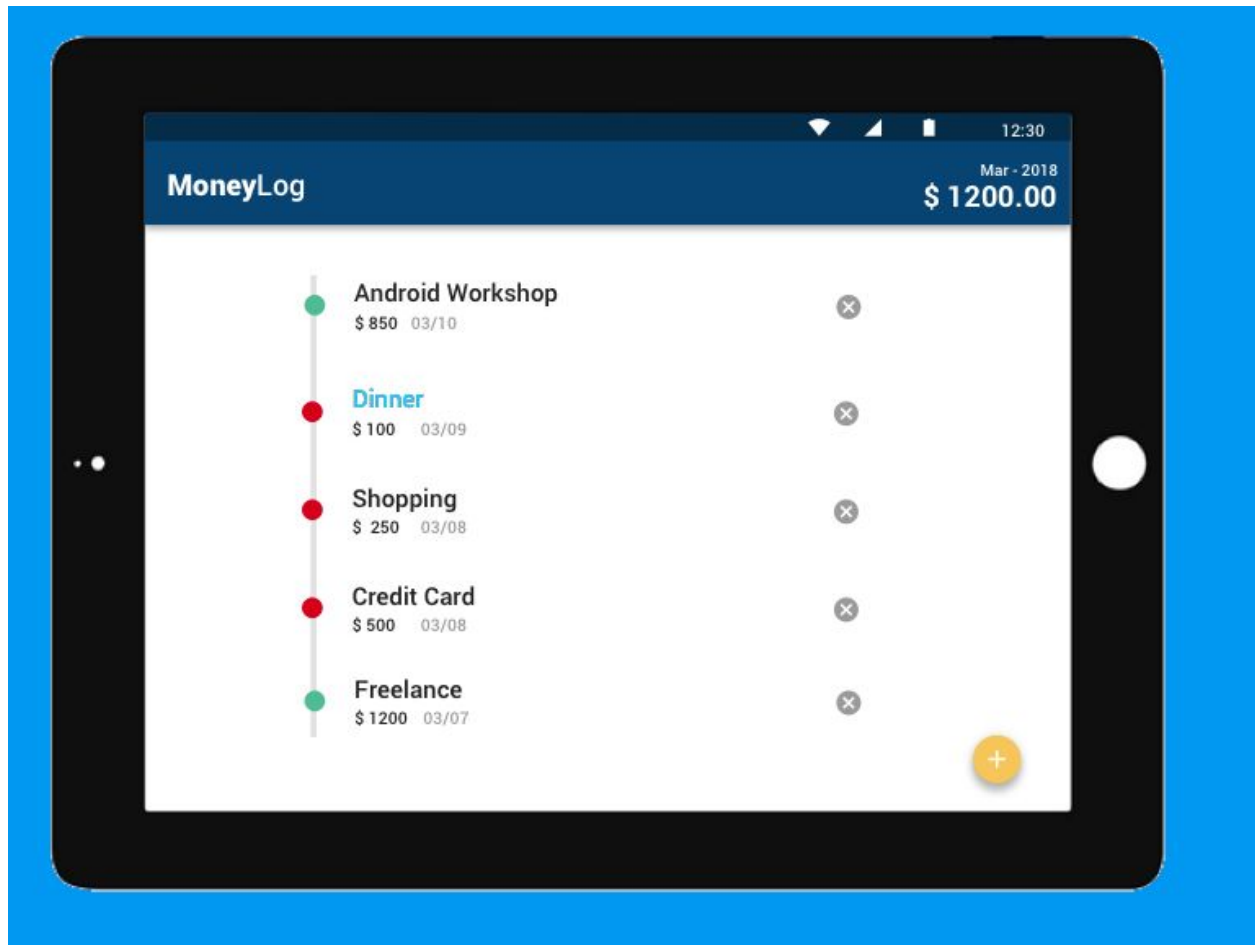
Pet shop

8 / 50

Debt ▼ \$

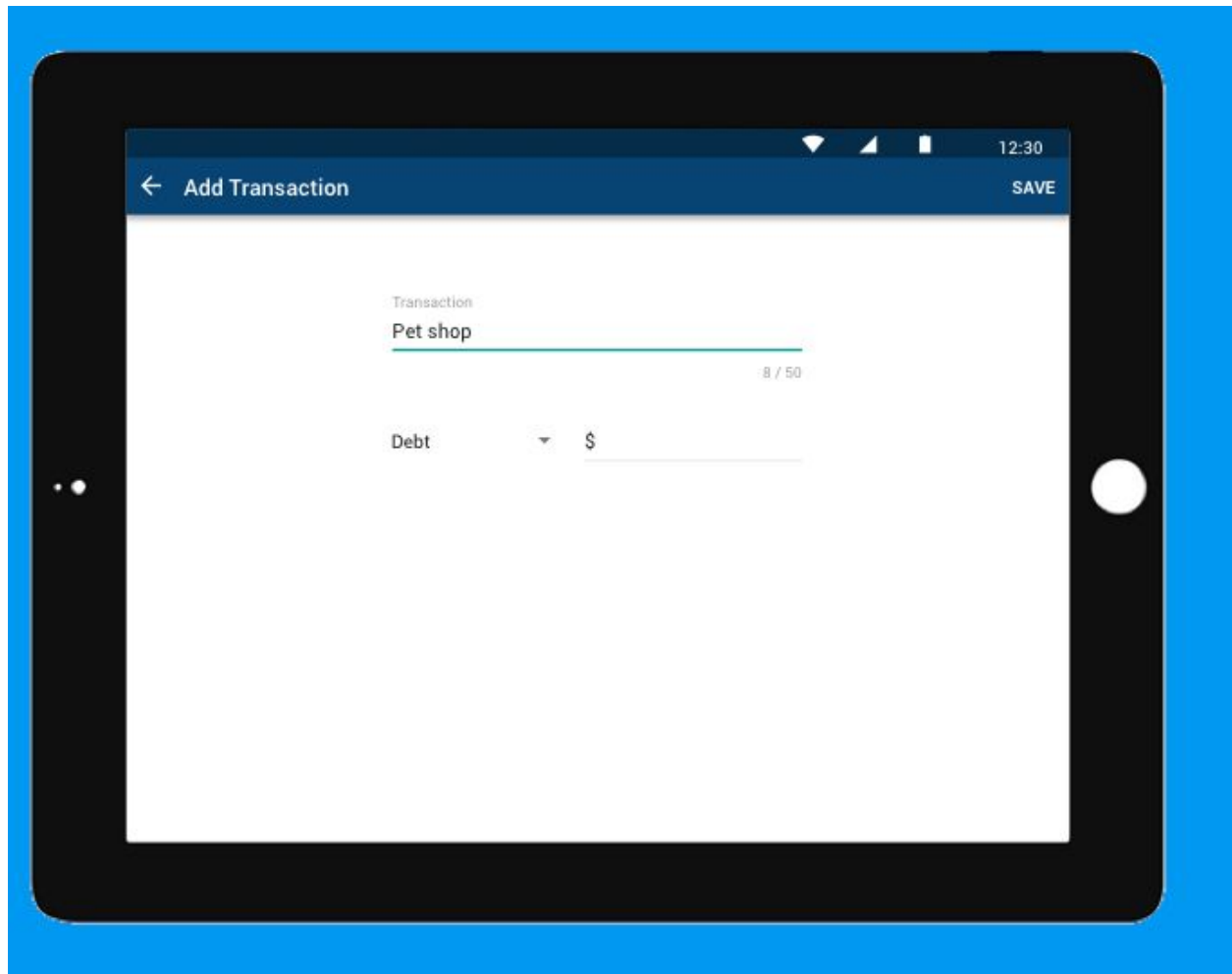
Screen designed to add new Transactions. It'll be accessed from the app main screen and the widget.

Screen 3 - Tablet - Main Screen



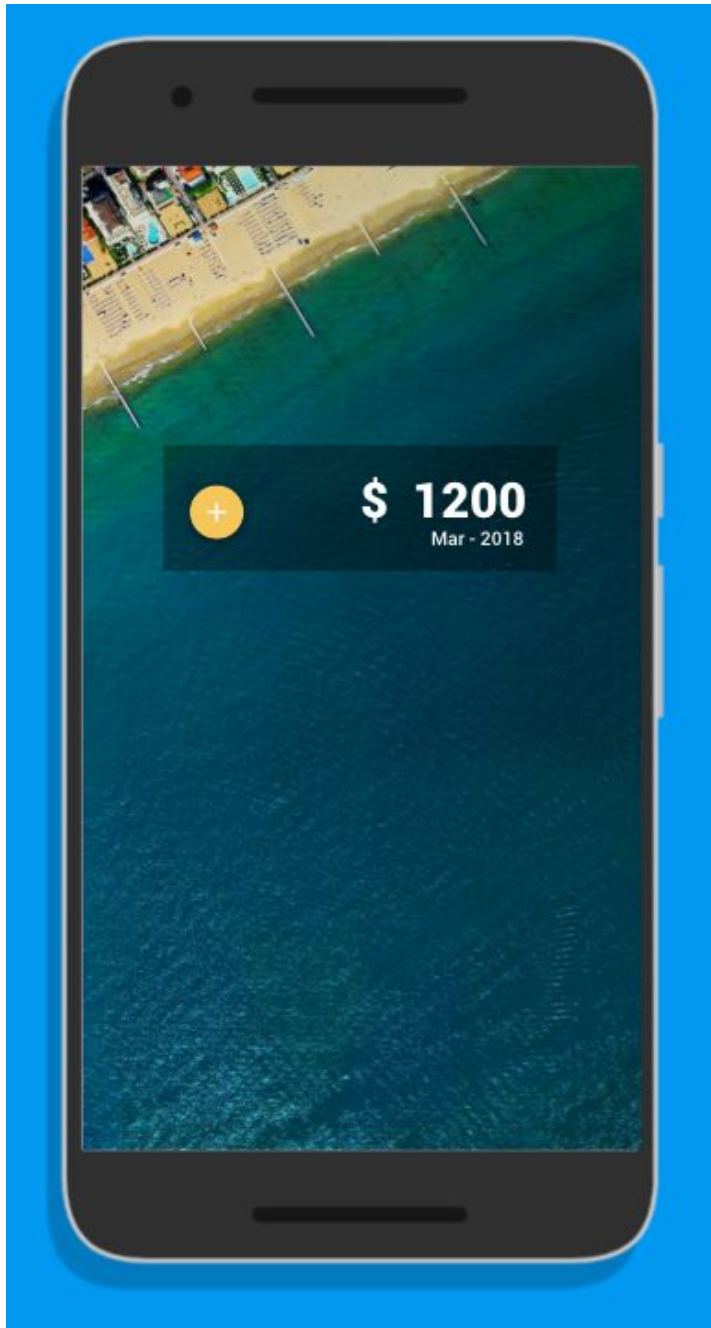
User's transactions history and his account summary - tablet version.

Screen 4 - Tablet - Add Transaction Screen

The image shows a tablet device with a black bezel and a white home button, set against a blue background. The screen displays the 'Add Transaction' interface. At the top, a dark blue header bar contains a back arrow, the text 'Add Transaction', and a 'SAVE' button. The status bar at the very top shows the time '12:30' and standard Android icons. The main content area is white and contains a text input field with the label 'Transaction' and the value 'Pet shop'. Below this is a character count '8 / 50'. Underneath the input field is a 'Debt' label, a dropdown arrow, and a currency symbol '\$' followed by an empty input field for the amount.

Tablet version of the screen designed to add new Transactions. It'll be accessed from the app main screen and the widget.

Screen 5 - Widget



Widget designed to show user's account summary directly from his home screen.

Key Considerations

How will your app handle data persistence?

MoneyLog will store user's transactions locally, using a Content Provider to add, list and delete entries.

Describe any edge or corner cases in the UX.

- Upon launch, users will get a list of their last transactions
- Users touch "Add Button" from Widget: They need to be redirected to Add Transaction screen, skipping Main Screen
- Users touch Push Notification : They need open the app on the Main Screen, showing their transactions.

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

- ButterKnife, to write less code whenever I need to find a view.
- Timber, for better logging
- Google's Support Library, specially Design Support Lib and NotificationCompat

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
- Create a code skeleton, creating placeholders for the future code
- Create build variants

Task 2: Content Provider

- Create Transactions Content Provider, allowing:
 - Add new Transaction
 - List Transactions
 - Remove a specific transaction via ID

Task 3: Main Activity

- Implement UI for MainActivity and TransactionsFragment for both Phones and Tablets
- Integrate UI with the Content Provider, listing transactions manually inserted in DB
- Integrate UI with the Content Provider, allowing to remove a Transaction

Task 4: Add Transaction Activity

- Implement UI for AddTransactionActivity and AddTransactionFragment
- Integrate UI with the Content Provider, allowing to insert a transaction
- Create notifications whenever a new transaction turns user's account to Positive or Negative

Task 4: Add Widget

- Implement UI for MoneyLog Widget
- Retrieve users account summary every minute
- Add a button to launch MoneyLog App in add transaction activity

Task 5: Prepare to release

- Create app flavors
- Create an APK and publish it to Google Play