Exploratory testing charter

Testers: Andres Daniel

Timebox

Date: 07/11/2022
Duration execution: 1h 30m

Objectives, environment and testing techniques

Objectives:
Discover the features of the app, how to use it and identify the paths that an user or client would use, another objective of this session is to make observations of what the app is doing right and what could do better, if in this session any issue or bug is detected on the app it would be described in the observations column and at the end on the defects / issues section.

Env:
The application would be running in its iOS version.
iOS version: 15.5
Monefy version: 1.3.30

Testing techniques:
- Black box testing
- Exploratory testing

Test ideas & test data

Test ideas:
- Try to test the paths that a normal user could use
- Try to validate the types of data that user could input
- Try to do a “bad use” of the app to watch the behavior
- Try to discover how much freedom the app gives to the user

Test data:
- No test data

Scope

Features to be tested:
- Tutorials (for first usage or any other available tutorial)
- Add incomes
- Add expenses
- Add transfers
- Update records
- Order records
- Search records
- Use filters
- Accounts configuration
- Currencies
- Settings

Features not to be tested:

Exploratory testing charter
- Premium account features

Test Log

Input action(s)	Expected output	Actual output	Observation(s), Defect-id, etc.
First usage of the app: following the steps	Should give the context to the user of how to start using the app	Gives an explanation of what the user could do with the app	N/A
Add incomes	Should add a new income to the app	The income is added, the user must choose a category and the currency	N/A
Add incomes: test the calculator feature	Should do basic math operations before adding a new expense	The calculator works correctly, the user can perform math operations such as addition, subtract, multiply and divide	N/A
Add expenses	Should subtract the entered value to the current balance, if balance is 0 should return a negative balance	The value is subtracted from the balance, if the balance is 0 it returns the box in a red color	The value is not returned as negative if the balance was 0 after the subtract
Add transfers	Should transfer the money to another account added previously	The transaction is successfully, the balance shows correctly	The balance is the sum of the money that user has in all of his accounts
Add transfers: from different accounts	Should transfer the money to another account added previously	The transaction is successfully, the balance shows correctly	N/A
Update records added previously	Should let the user update an expense or income added previously	The user is able to update incomes or expenses	The categories shows to the user the percentage of the expense
Order records	Should let the user order the record added by expense or income	The user can order records by date or by the category	N/A

Search records	Should let the user to search a record by the note added or the category	The user is only able to search a record by the note added when its created or updated	If there isn't any note added user won't be able to find the data
Use filters	Should be able to list the incomes and expenses that were added in the time of the filter	It shows the data of the day, week, month, year, all and interval or user can choose an specific date and all accounts o an specific account	N/A
Accounts configuration	Should let user to create or update an account added	User is able to create or update any account	N/A

Debriefing information; (Filled at the end of the session)

Defects / Issues
No issues or bugs found
Conclusion / Tests notes / Summary about product quality
In the big picture the application works correctly, maybe there are some functionalities that I'm not pretty sure with how they work, but maybe there could be resolved reading the documentation of the app. It's a small application and it guides the user to not input incorrect values or too long values so it could be a hard task to break the application.
Prioritization
I should suggest prioritizing test cases that are related to the main functionalities like add expenses, incomes, the calculator on the previously mentioned and transfers, because without those functionalities there is not so much to do in the app. Next the validations of searches and order by and some settings, like add new account, update or delete.
Risks
- Not all the functionalities can be tested because it needs a premium account