**Task 1: EXPLORATORY TEST CHARTERS - MONEFY APP**

1. **TEST CHARTERS:**

Function of the application - Expense management. Following exploratory test charters will be in the form of below format:

Explore <target>

With <resources>

To discover <information>

Report <expected output vs actual output>

* **TEST CASE 1: Balance calculation.**

**Target**: income and expenditure function

**Resources**: inputs on income, different categories of expenses

**Information**: i) balance = income - (sum of expenditures) where income > sum of expenditures.

**Output**: function to deduct balance works correctly.

* **TEST CASE 2: Credit/loan or negative balance calculation.**

**Target**: income and expenditure function

**Resources**: inputs on income, different categories of expenses

**Information**: i) balance = income - (sum of expenditures) where income < sum of expenditures

ii) balance can be + or - where + balance means saving, - balance means loan/amount you owe to others (Red)

**Output**: function to deduct negative balance works correctly.

* **TEST CASE 3: Payment Card function (2 ways to operate - included/excluded from salary)**

**Target**: Payment Card

**Resources**: i) USD Amount + initial date, included

ii) USD Amount + initial date, excluded

**Information**: i) balance = income - (sum of expenditures) where income < sum of expenditures

ii) balance can be + or - where + balance means saving, - balance means loan/amount you owe to others (Red)

**Output**: function to deduct negative balance works correctly.

* **TEST CASE 4: Pie chart generation – based on expense distribution (not on income distribution)**

**Target:** Income-expense operation

**Resources:** Apply expenditures of different types (start with home expense)

**Information:** Here in app, pie chart % of each expenditure x = (x amount /sum of all expenditures) \* 100

i.e the app is considering distribution of each individual expense to total spent (and not income distribution)

**Output:** suppose income **=** 3000 ; House rent = 800 ; then pie chart percent should be = ( 800/3000) \* 100 = 26% but here in app shows 100% ie (800/800) \* 100 = 100%

second expense suppose is 100 for communication = the pie chart should show 3.33 % dedicated to it, but here in app shows 11%

Hence, we deduct that expected output of app does not matches to actual output.

**MODE OF APP:**

* **TEST CASE 5: Management of week expense if weekly income (or day)**

**Target:** Left menu – week operation

**Resources:** adding weekly expenses and generate weekly report

**Information: i)** any expense added in this mode will take present day’s date and upgrade present week’s report.

ii) swiping right left is possible to see previous or this week’s report

iii) this mode of app works on intension of having weekly salary since each week has own income, expenses and balance.

iv) This mode can be switched to monthly with having combined salaries, combined all week’s expenses and a single balance sheet.

**Editing of Expenses:**

* **TEST CASE 6: + and – signs testing**

**Target:** + sign

**Resources:** click on green + sign, and directly click on category – does blink on input field.

**Information:** It depicts user to fill the amount in currency first, but does not show any error message or to do message.

**Output: Buggy –** since prompt does not specify user message. Although existing amount edition is convenient.

* **TEST CASE 7: Deleting individual entries**

**Target:** each individual record of expense.

**Resource:** click on delete icon

**Information:** Delete function works well but there is no confirmation message **(buggy)** that is user sure to delete it. There is an undo functionality, if user clicks on Cancel, the deletion is undone. And expense chart updates simultaneously.

* **TEST CASE 8: Operating the app in offline mode (No internet)**

**Target:** application basic function in offline

**Resources:** basic operations - +

**Information:** Since app does not require login user, server info, app operates well in offline mode. Although there are app parts which cannot be operated in offline, and app notifies user with proper message. These areas are right menu – add more categories and add more currencies.

* **TEST CASE 9: Transfer function- cash to Payment Card (and vice versa)**

**Target:** Transfer function

**Description:** added $50 to payment card

**Information:** $50 from salary is deducted as expenses as fund transfer.

* **TEST CASE 10: Delete all transaction**

**Target:** Settings

**Resources:** Clear data

**Information:** saved records in the app and drop box deleted successfully.

* **TEST CASE 11: Create back and restore data**

**Target:** Settings

**Resources**: i) Create backup

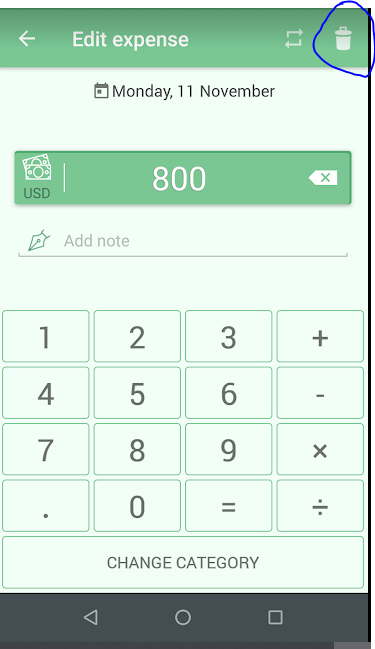
ii) Restore data

**Information:** At any point of time, user can replicate the copy of present data and store in this phone memory. Also, if he has a separate file to change existing app’ s date, he can replace app data from that file.

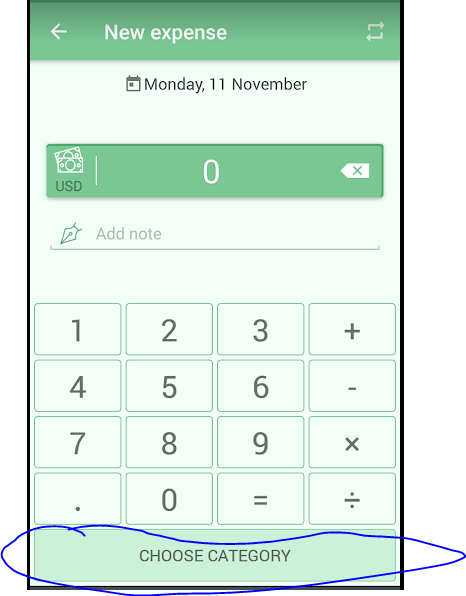
1. **FINDINGS AND BUGS DISCOVERED FROM ABOVE CHARTERS**

Although most of the user functionality and calculation features works well in the app, there are some bugs or UI lacking found in the app:

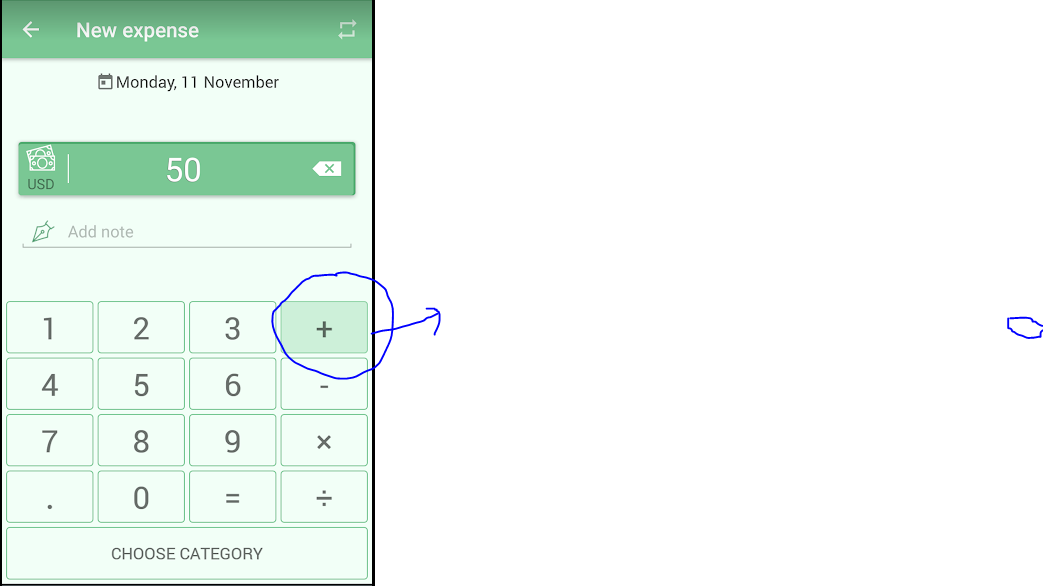
**BUGS:**  
  
1. While deleting an existing record, there is no confirmation message asked from user. This is deletion may affect his overall expense history and balance, it should be asked from user again, that is he sure to delete that.



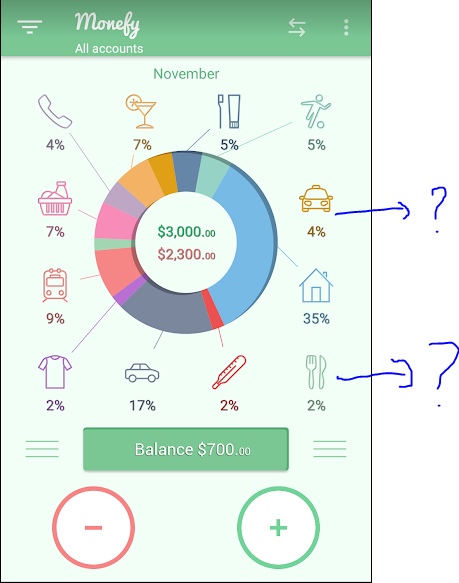
2. Choosing category before entering the amount- blinks the screen but does not give a proper message to user to fill amount first.



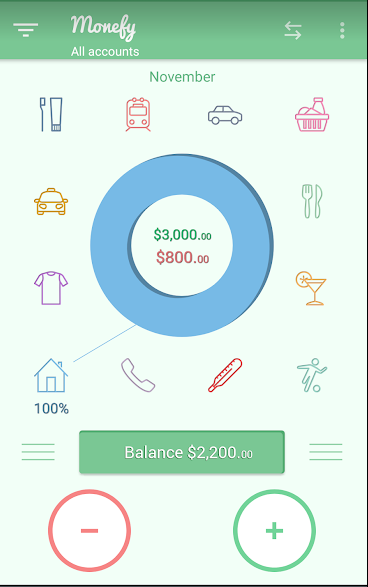
3. Addition, subtraction and other mathematical operations can be done on present quantities, but while clicking any symbol (+ or – etc), user has no idea -what is going on the screen. There should be **a color change or press key impression**, which can make the user aware that a particular operation is going on present amount.



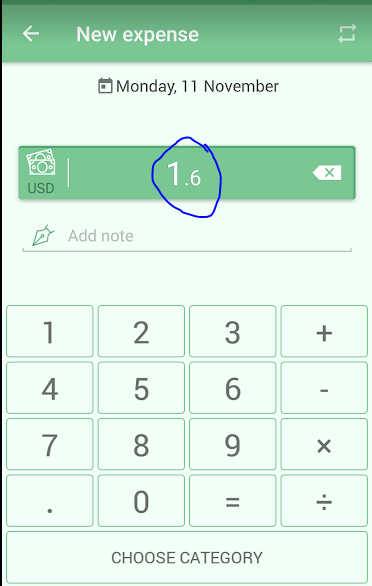
4. Pie chart pointing error – in case of many expenses on the display, the expense distribution lacks some pointing / may be because of lack of space on screen.

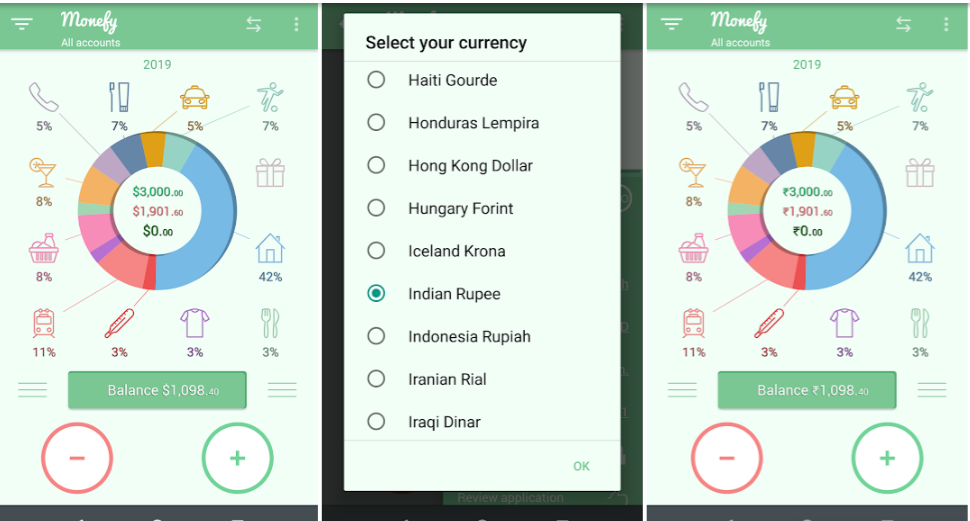


5. The expense distribution strategy used does not indicate ratio of income used in a particular expenditure. suppose income = 3000 ; rent = 800 ; ratio should be = 800/3000 \* 100 = 26% , but below is 100 % ( ratio of expense to total sum of all expenses)

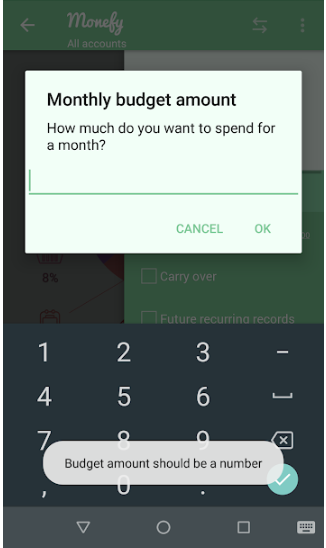


6. Non- Uniform font – after using decimal in USD amount



7. After changing the currency to a different format, the amount in USD is not convertible to any other currency. The digits remain same. 

8. The budget amount cannot be set to null once initialized.



1. **PRIORTISATION OF CHARTERS**

🡪Since app’s main objective is expense management so balance calculation, merging of categories, creation of expenses and expense distribution are the key factors of function. The functionality of the app lies in basic calculations and display of results.

🡪Secondly, factors such as maintaining a week’s data, months transaction, or converting them to year are also second priority to be considered under test.

🡪Other factors such as currency change, maintaining data across app throughout usage, restoring backups and export function are other secondary priorities which helps in maintaining user record and copies of data.

1. **TIME ESTIMATION FOR EACH CHARTER**

For careful analysis of the application, I have divided the app into below modules and dedicated an amount of time in each functionality.

* BALANCE AND PIE CHART –1 HOUR
* CATEGORY EDITION, DELETION, UNDO, MERGING, NOTES –1 HOUR
* LOCKED FEATURES – 2O MINS
* SETTINGS – 25 MINS
* WEEK/YEAR/ DAY EXPENSES –40 MIN

1. **RISKS TO BE MITIGATED FOR THIS TYPE OF APPLICATION.**

Since the application is not server based, does not require login – logout authentication of users. Hence, the main app’s functions are static in nature. The app is based on mathematical operations and logic to display calculated results. Still there are risks or factors to be considered for better user experience.

1. **Alerts and confirmation messages** for upcoming decisive operations- to make user aware.
2. Operations regarding single entity should be kept together – example here **merging, disabling of records** is kept under settings section. But editing amount, adding second amount in same category, deletion of record is kept in list of expense (horizontal bars beside balance button)
3. **Locked features should be marked (or highlighted).** User has to explore each and every setting option and dig into it, only to find out that the feature is locked. This should be improved, as it is annoying.
4. There should be **a help section** – which is a kind of user manual explaining all process. For example: the function of carry over or other hidden parts can be highlighted there.