Test Plan and Execution

Name : Ranuth Dassanayaka

Course : Computer Science

Module : Software Development 1

Date : 28.04.2024

ID : 20234116

* Diplay Transactions
  + Scenario 1 : Loads transactions from an existing file.
    - Test : Calling *display\_transactions( )* function when the file exist.
    - Expected Outcome : Transactions must load and display in the GUI.
    - Actual Outcome : Transactions are loaded successfully in the GUI.
* Save Transactions
  + Scneario 1 : Adding a transaction and see whether it shows when viewing.
    - Test : Calling save\_transactions( ) after adding a transaction
    - Expected Outcome : We should see the transaction we added when we use view\_transactions( ). And in the GUI as well coz of display\_transactions.
    - Actual Outcome : Can see the added transaction when we try view\_transactions( ) and in the GUI as well.
* Add Transactions
  + Scenario 1: Add a new transaction
    - Test : Calling add\_transaction( ) and enter valid inputs to the details.
    - Expected Outcome : New transaction should be added to the list and displayed in the GUI.
    - Actual Outocme : New transaction is added and can be viewed via GUI as well.
* Update Transactions
  + Scenario 1 : Update an existing transaction
    - Test : Calling update\_transaction( ) and inputting valid inputs for everything.
    - Expected Outcome : Transaction in the user entered index should be updated with user entered new inputs. And should be displayed in the GUI.
    - Actual Outcome : “Transaction updated successfully!” comment appeared and it successfully updated. It is displayed in the GUI as well.
* Read from a File
  + Scenario 1 : Reading already existing transactions from an external file.
    - Test : Calling read\_bulk\_transactions\_from\_file( ).
    - Expected Outcome : External file should be imported to our dictionary and the use should see the new transactions that are adding from that external file. In the GUI too.
    - Actual Outcome : The external file transactions are added successfully and shown in the GUI as well.
* Sort Function
  + The transactions should be sorted in ascending order and descending order both and with all three terms. Category, Amount, Date.
    - Category will be sorted according to the Alphabet(A-Z and Z-A)
    - Amount will be sorted after getting the sum of all the transactions of a specific category and then make them show in ascending or descending.
    - Date is in FCFS method.
  + Test : Using all the GUI buttons to do the Asc and Desc sorting.
  + Expected Outcome : All the sorting methods should work as explained in the above descriptions.
  + Actual Outcome : It did work as the descriptions said and works as expected.
* Search Function
  + Test : Searching via category, amount and date.
  + Expected Outcome : When we search through all three a result should come as an output.
  + Actual Outcome : It is displayed as expected.

**Conclusion**

**Tests Done : 07**

**Tests Passed : 07**

**Tests Failed : 00**