Test Plan and Execution

Name : Ranuth Dassanayaka

Course : Computer Science

Module : Software Development 1

Date : 12.04.2024

ID : 20234116

* Load Transactions
  + Scenario 1 : Loads transactions from an existing file.
    - Test : Calling *load\_transactions( )* function when the file exist.
    - Expected Outcome : Transactions must load to the transaction dictionary.
    - Actual Outcome : Transactions are loaded successfully.
  + Scenario 2 : Load transactions when the file doesn’t exist.
    - Test : Calling *load\_functions( )* function when the file doesn’t exist.
    - Expected Outcome : Transactions must not load to the dictionary because the file doesn’t exist.
    - Actual Outcome : Transactions are not loaded.
* 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( ).
    - Actual Outcome : Can see the added transaction when we try view\_transactions( ).
* 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.
    - Actual Outocme : New transaction is added and can be viewed.
  + Scenario 2 : Add a new transaction with a negative value.
    - Test : Calling add\_transaction( ) and entering a negative value for the amount.
    - Expected Outcome : It should not pass a negative amount.
    - Actual Outcome : It did pass a negative value for the amount.
    - Bug detected.
    - Fixed the bug by setting an if statement saying if amount is less than 0 then print(“Please enter a value more than 0”) and ask the user amount again.
    - Final : Works fine as expected.
  + Scenario 3 : Add a new transaction.
    - Test : Calling add\_transaction( ) and entering a numeric value for category.
    - Expected Outcome : It should not pass numeric values for the category field only strings.
    - Actual Outcome : It did pass a negative value for the amount.
    - Bug detected.
    - Fixed the bug by checking if the value is a digit and if it is print an error message and ask the user for a category again. If category.isdigit( )
    - Final : Works fine as expected.
  + Scenario 4 : Add a new transaction.
    - Test : Calling add\_transaction( ) and entering an invalid date.
    - Expected Outcome : It should not pass that date.
    - Actual Outcome : It did pass that date.
    - Bug detected.
    - Fixed the bug by checking if the date entered is in the standard format by using datetime library.
    - Final : Works fine as expected.
  + Scenario 5 : Add a new transaction with a string.
    - Test : Calling add\_transaction( ) and entering a string for the amount.
    - Expected Outcome : It should not pass a string value for the amount.
    - Actual Outcome : ValueError raised.
    - Bug detected.
    - Fixed the bug by handling the error using a try and except.
    - Final : Works fine as expected.
* View Transactions
  + Scenario 1 : View existing transactions.
    - Test : Calling view\_transactions( )
    - Expected Outcome : Existing data will be displayed.
    - Actual Outcome : Existing transactions displayed correctly with the comment “Transactions: ”.
  + Scneario 2 : View transactions when there are no data in the list.
    - Test : Calling view\_transactions( ) when there are no data in the list.
    - Expected Outcome : A “No transactions to view!” comment will appear.
    - Actual Outcome : A “No transactions to view!” comment appeared.
* 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.
    - Actual Outcome : “Transaction updated successfully!” comment appeared and it successfully updated.
  + Scenario 2 : Trying to update a transaction which doesn’t exist.
    - Test : Calling update\_transaction( ) and inputting an invalid index.
    - Expected Outcome : Should appear “Transaction not found!” comment.
    - Actual Outcome : Appeared “Transaction not found!” comment.
  + Scenario 3 : Trying to update a transaction by inputting a negative index.
    - Test : Calling update\_transaction( ) and inputting a negative index.
    - Expected Outcome : Nothing happening and a comment saying “Transaction not Found” will appear.
    - Actual Outcome : Index Error
    - Bug detected
    - Fixed the bug by setting the index range from 0 to length of the transactions dictionary.
    - Final : Works fine as expected.
* Display Summary
  + Scenario 1 : Display summary of transactions.
    - Test : Calling display\_summary( )
    - Expected Outcome : A summary should be displayed with each transaction category and the No. of transactions the user did in that category and sum of the amount transactions happened in that category.
    - Actual Outcome : A summary as expected is displayed.
* 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.
    - Actual Outcome : Transactions added successfully but the transactions didn’t display in the CLI.
    - Bug detected.
    - Fixed it by writing a function (def view\_bulk\_transactions) to display all the transactions of the file we are adding via the CLI. So that the user get an idea of what we are adding.
    - Final : Works as expected!
  + Scenario 2 : Reading already existing transactions from an external file.
    - Test : Calling read\_bulk\_transactions\_from\_file( ).
    - Expected Outcome : Added transactions should be inserted to our global dictionary.
    - Actual Outcome : Transactions added successfully but didn’t display in the JSON file.
    - Bug detected.
    - Fixed it by using save\_transactions( ).
    - Final : Works as expected!
  + Scenario 3 : Reading already existing transactions from an external file.
    - Test : Calling read\_bulk\_transactions\_from\_file( ).
    - Expected Outcome : When calling the same file again. It should update only the updated transaction in the file.
    - Actual Outcome : Transactions added successfully but the already existing transactions got added twice (duplicated).
    - Bug detected.
    - Fixed it by checking if every individual transaction exist in the transactions dictionary already and if yes it won’t get recorded and If not it will get recorded.
    - Final : Works as expected!

**Conclusion**

**Tests Done : 17**

**Tests Passed : 09**

**Tests Failed : 08**

**Bugs Found : 08**

**Adjustments : 08**