Test Plan and Execution

Personal Finance Tracker | 4COSC006C[PRO] | Coursework 01

Name: Ambagahage Shan Emalka Fernando
IIT Student No: 20233126
UoW Student No: w2082285

CONTENT

TESTING

Case No:	Function	Test Case	Input	Expected outcome	Actual Outcome	Status	Image
01	load_transation()	Run the program with “finance.json” file with data within.	Run program	Program runs successfully.	Program runs successfully.	Pass	Figure 1
02		Run the program without “finance.json” file	Run program	Display “*** File not found! New file has been created.***” and run successfully.	*** File not found! New file has been created.*** |runs successfully	Pass	Figure 1
03		Run program with “finance.json” without data in file.	Run program	Program runs successfully.	Json.decoder. JSONDecoderError	Fail	Figure 1
04	Main_menu()	Enter “1” in the menu selection.	1	Asks user to “Enter Amount: “	Enter Amount:	Pass	Figure 2
05	add_transaction()	Enter non float/integer value	Enter a string	Prints “Invalid Input!”	Invalid Input	Pass	Figure 3
06		Enter zero	0	Prints “Enter amount higher than zero.”	Enter amount higher than zero.	Pass	Figure 4
07		Enter an amount higher than zero	1000	Asks user to “Enter category: “	Enter category:	Pass	Figure 5
08		Press Enter without providing an input.	Enter	Prints “Invalid Input!” Asks user to “Enter category: “	Invalid Input!	Pass	Figure 6
09		Enter the category of the amount	Salary	Asks user to “Enter type of transaction(Income{I} or Expense{E}): “	Enter type of transaction(Income{I} or Expense{E}):	Pass	Figure 7
10		Enter an integer/float value.	1	Prints “Invalid Input!” And asks the user to “Enter type of transaction(Income{I} or Expense{E}):”	Invalid Input! Enter type of transaction(Income{I} or Expense{E}):	Pass	Figure 8

11		Enter a wrong letter	a	Prints “Invalid Input!” And asks the user to “Enter type of transaction(Income{I} or Expense{E}):”	Invalid Input! Enter type of transaction(Income{I} or Expense{E}):	Pass	Figure 9
12		Enter word with spelling mistake	icome	Prints “Invalid Input!” And asks the user to “Enter type of transaction(Income{I} or Expense{E}):”	Invalid Input! Enter type of transaction(Income{I} or Expense{E}):	Pass	Figure 10
13		Enter “i/I ” or “e/E” or “income” or “expense”	income	Print “Enter >>> “ and and ask for “Year: ”	Enter >>> Year:	Pass	Figure 11
14		Enter a string value.	Enter a string	Print “Enter a number!” and ask for “Year: “	Enter a number! Year:	Pass	Figure 12
15		Enter less than 4 digits.	224	Print “Invalid Year! Please re-enter.” And ask for “Year: “	Invalid Year! Please re-enter. Year:	Pass	Figure 13
16		Enter valid year	2024	Asks for “Month: “	Month:	Pass	Figure 14
17		Enter a string value.	Enter a string	Print “Enter a number!” Ask “Month: “	Enter a number! Month:	Pass	Figure 15
18		Enter zero	0	Print “Invalid Month! Please re-enter.” Ask “Month: “	Invalid Month! Please re-enter. Month:	Pass	Figure 16
19		Enter thirteen	13	Print “Invalid Month! Please re-enter.” Ask “Month: “	Invalid Month! Please re-enter. Month:	Pass	Figure 17
20		Enter valid month	2	Ask “Day: ”	Day:	Pass	Figure 18
21		Enter a string value.	Enter a string	Print “Enter a number: “ Ask “Day: “	Enter a number! Day:	Pass	Figure 19
22		Enter zero	0	Print “Invalid day! Please re-enter a day between 1 and 29.” Ask “Day: “	Invalid day! Please re-enter a day between 1 and 29. Day:	Pass	Figure 20
23		Enter thirty.	30	Print “Invalid day! Please re-enter a day between 1 and 29.” Ask “Day :”	Invalid day! Please re-enter a day between 1 and 29. Day:	Pass	Figure 21
24		Enter valid day	29	Print “Transaction added successfully!” Ask “<<<Press enter to continue>>>”	Transaction added successfully! <<<Press enter to continue>>>	Pass	Figure 22

25	main_menu()	Press Enter	Enter	Shows main_menu	Shows main_menu	Pass	Figure 23
26	view_transaction()	Enter “2” in the menu selection.	2	Show all Transactions added. <<<Press any key to continue>>>	Shows all Transactions added. <<<Press any key to continue>>>	Pass	Figure 24
27	main_menu()	Press Enter	Enter	Shows main_menu	Shows main_menu	Pass	Figure 25

28	update_transactions()
	delete_transactions()
		Enter “3” in the menu selection. (“4” Delete function the same)	3	Show all Transactions added.
				<<<Press any key to continue>>>
					Show all Transactions added.
					<<<Press any key to continue>>>
						Pass	Figure 26

29		Press Enter	Enter	Ask “Enter the ID of the transaction you would like to update:	Enter the ID of the transaction you would like to update:	Pass	Figure 27
30		Enter zero	0	Print “Transaction ID of that value does not exist! Please Input a Valid index.” Ask “Enter the ID of the transaction you would like to update:”	Transaction ID of that value does not exist! Please Input a Valid index. Enter the ID of the transaction you would like to update:	Pass	Figure 28
31		Enter seven (Final ID is six)	7	Print “Transaction ID of that value does not exist! Please Input a Valid index.” Ask “Enter the ID of the transaction you would like to update:”	Transaction ID of that value does not exist! Please Input a Valid index. Enter the ID of the transaction you would like to update:	Pass	Figure 29
32		Enter valid ID	6	All prompts are asked again.	All prompts are asked again.	Pass	Figure 30
33		Enter invalid choice in main_menu	0	Print “Invalid choice. Please try again.” Shows main_menu	Invalid choice. Please try again.	Pass	
34		Enter six	6	Print “Exiting program” Exits code	Exiting program. Exits code	Pass	

Summary

- Total tests: 35
- Tests passed: 34
- Tests failed: 01
  - The issue in the code is an error when data is missing from the JSON file or the file is corrupted.

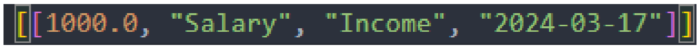
Bugs and De-Bugging

load_transactions()

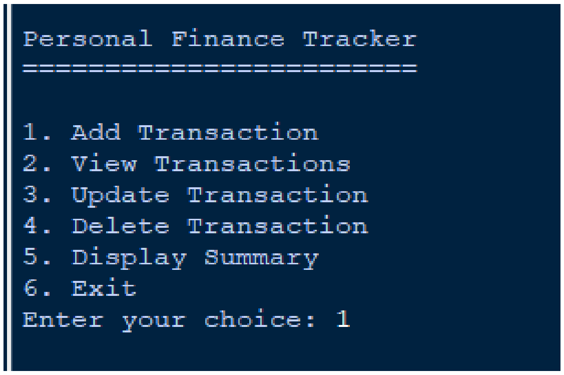
function Before Error Handling

Optimal scenario:

- “finance.json” exists
- Data within “finance.json” exists.
- Test steps:
  - Run program



```
[[1000.0, "Salary", "Income", "2024-03-17"]]
```



```
Personal Finance Tracker
========================

1. Add Transaction
2. View Transactions
3. Update Transaction
4. Delete Transaction
5. Display Summary
6. Exit
Enter your choice: 1
```

Expected Outcome:

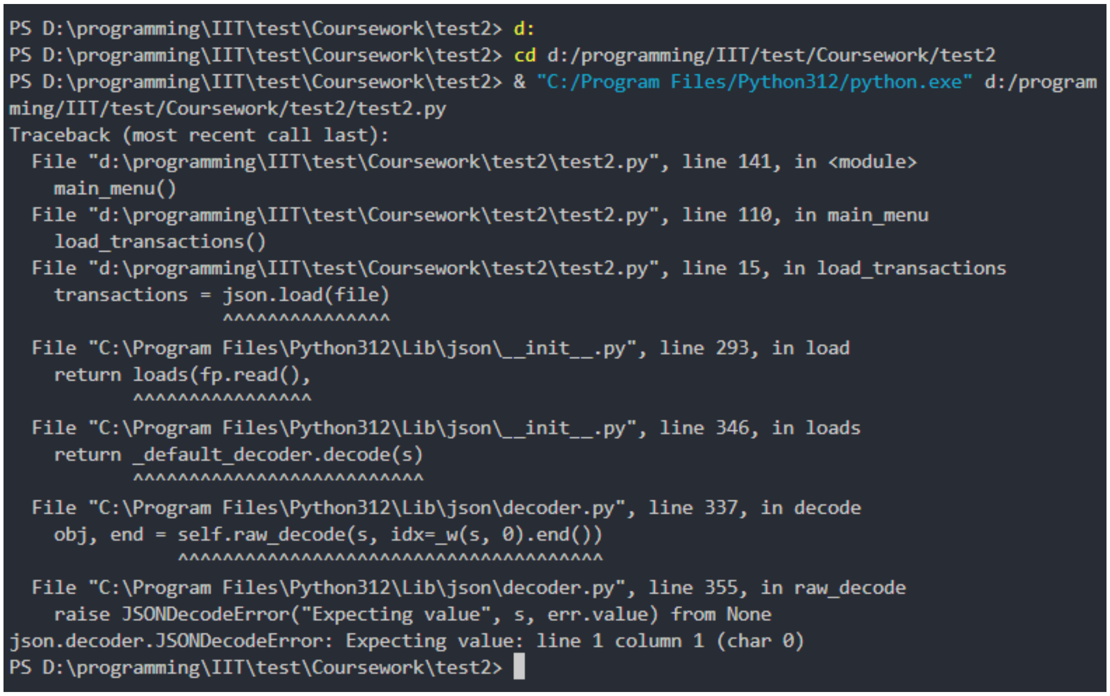
- Transaction should be loaded from file without errors.

Actual Outcome

- Transactions are loaded without errors.

Test scenario 1:

- "finance.json" exists.
- Data within "finance.json" does not exist.
- Test steps:
    - Run program



```
PS D:\programming\IIT\test\Coursework\test2> d:
PS D:\programming\IIT\test\Coursework\test2> cd d:/programming/IIT/test/Coursework/test2
PS D:\programming\IIT\test\Coursework\test2> & "C:/Program Files/Python312/python.exe" d:/program
ming/IIT/test/Coursework/test2/test2.py
Traceback (most recent call last):
  File "d:\programming\IIT\test\Coursework\test2\test2.py", line 141, in <module>
    main_menu()
  File "d:\programming\IIT\test\Coursework\test2\test2.py", line 110, in main_menu
    load_transactions()
  File "d:\programming\IIT\test\Coursework\test2\test2.py", line 15, in load_transactions
    transactions = json.load(file)
                   ^^^^^^^^^^^^^^^
  File "C:\Program Files\Python312\Lib\json\__init__.py", line 293, in load
    return loads(fp.read(),
           ^^^^^^^^^^^^^^^^
  File "C:\Program Files\Python312\Lib\json\__init__.py", line 346, in loads
    return _default_decoder.decode(s)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Program Files\Python312\Lib\json\decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Program Files\Python312\Lib\json\decoder.py", line 355, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
PS D:\programming\IIT\test\Coursework\test2> 
```

Expected Outcome:

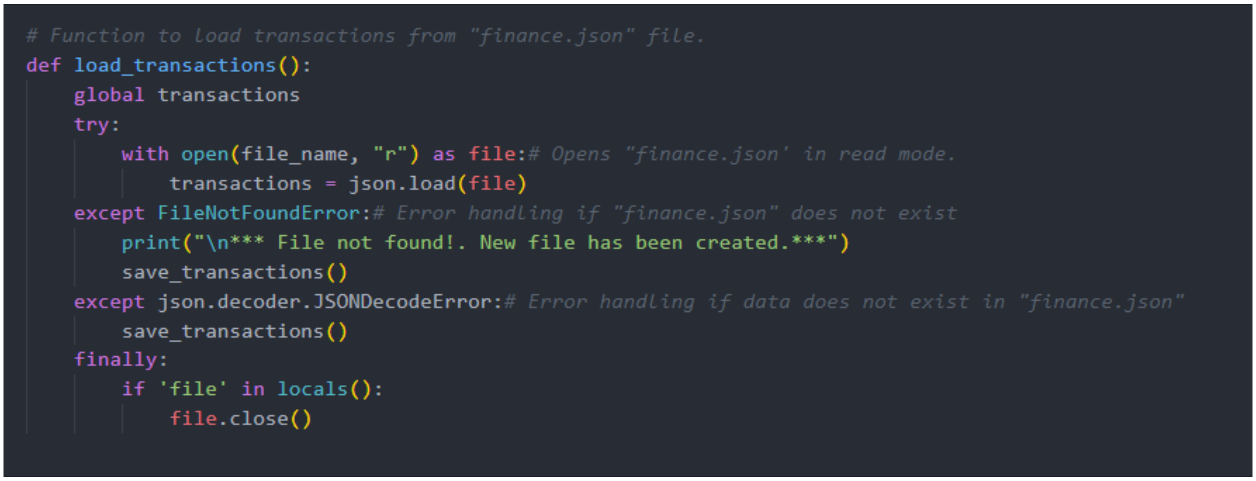
- Transaction should be loaded from file without errors.

Actual Outcome

- JSONDecodeError.

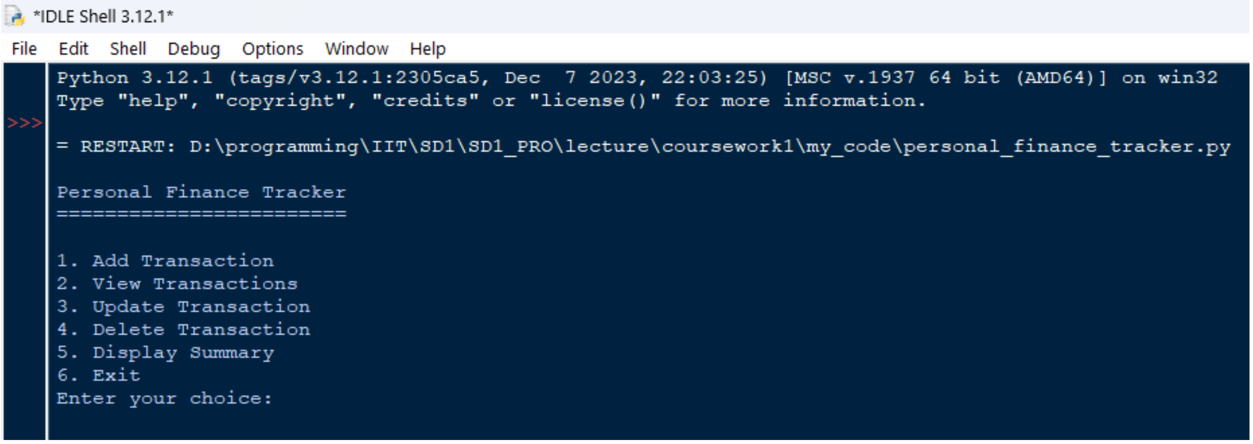
Adjustments made to the program

- Added a try except statement with two excepts to handle each error.
- Each except has called in save_transaction(): it creates a new JSON file with an empty list because it opens the file in write mode and writes the empty list to the file.



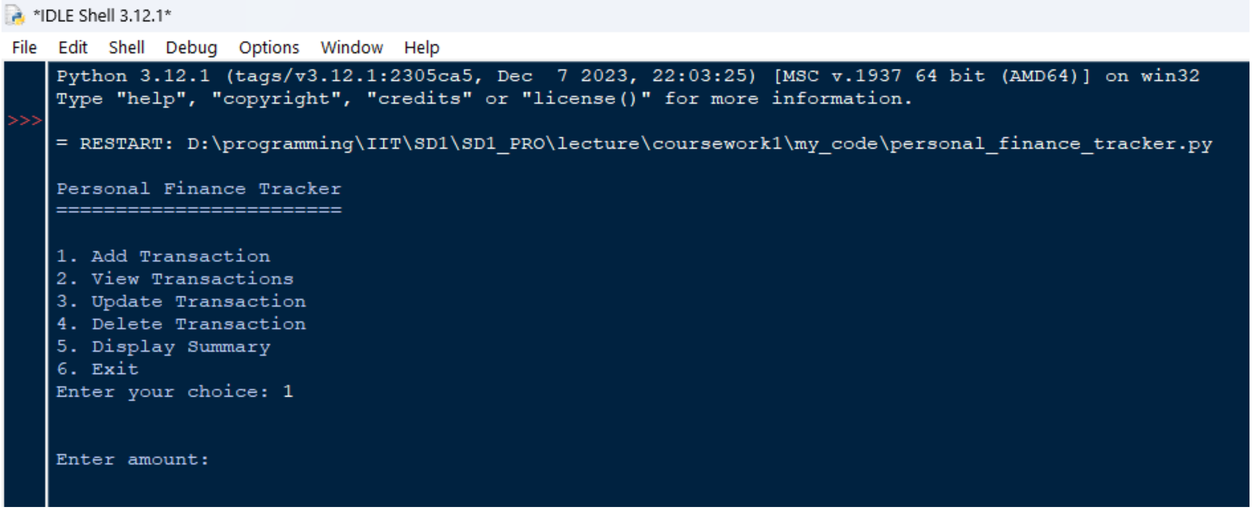
```
# Function to load transactions from "finance.json" file.
def load_transactions():
    global transactions
    try:
        with open(file_name, "r") as file:# Opens "finance.json' in read mode.
            transactions = json.load(file)
    except FileNotFoundError:# Error handling if "finance.json" does not exist
        print("\n*** File not found!. New file has been created.***")
        save_transactions()
    except json.decoder.JSONDecodeError:# Error handling if data does not exist in "finance.json"
        save_transactions()
    finally:
        if 'file' in locals():
            file.close()
```

Images



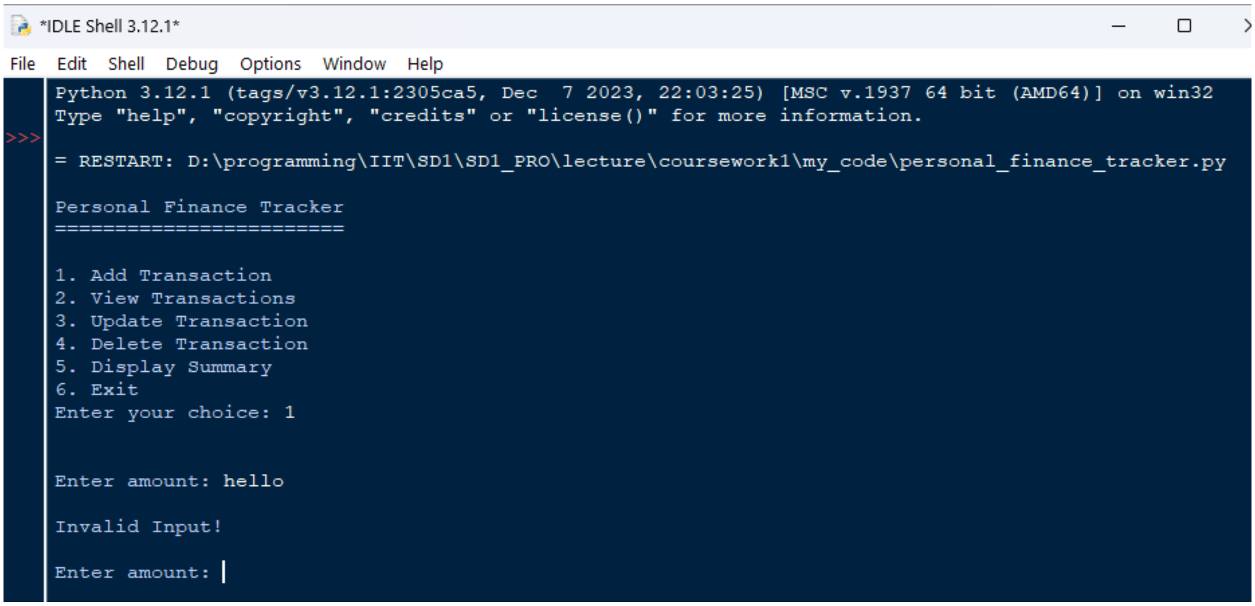
```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\programming\IIT\SD1\SD1_PRO\lecture\coursework1\my_code\personal_finance_tracker.py

Personal Finance Tracker
========================

1. Add Transaction
2. View Transactions
3. Update Transaction
4. Delete Transaction
5. Display Summary
6. Exit
Enter your choice:
```

Figure 1



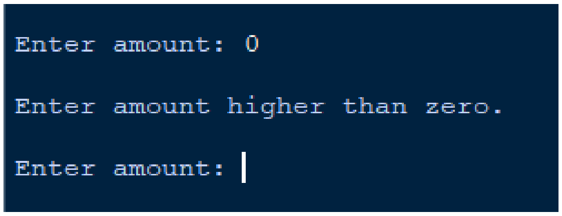
```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\programming\IIT\SD1\SD1_PRO\lecture\coursework1\my_code\personal_finance_tracker.py

Personal Finance Tracker
========================

1. Add Transaction
2. View Transactions
3. Update Transaction
4. Delete Transaction
5. Display Summary
6. Exit
Enter your choice: 1


Enter amount:
```

Figure 2



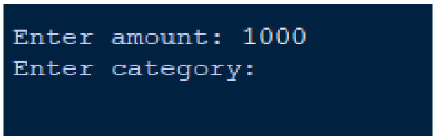
```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\programming\IIT\SD1\SD1_PRO\lecture\coursework1\my_code\personal_finance_tracker.py

Personal Finance Tracker
========================

1. Add Transaction
2. View Transactions
3. Update Transaction
4. Delete Transaction
5. Display Summary
6. Exit
Enter your choice: 1


Enter amount: hello

Invalid Input!

Enter amount: 
```

Figure 3



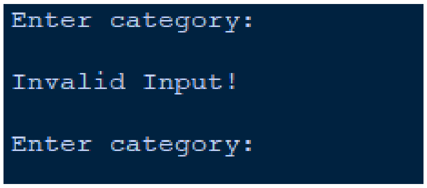
```
Enter amount: 0

Enter amount higher than zero.

Enter amount: 
```

Figure 4



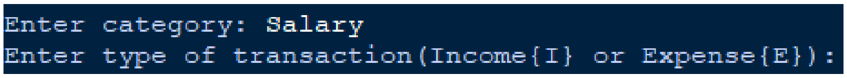
```
Enter amount: 1000
Enter category: 
```

Figure 5



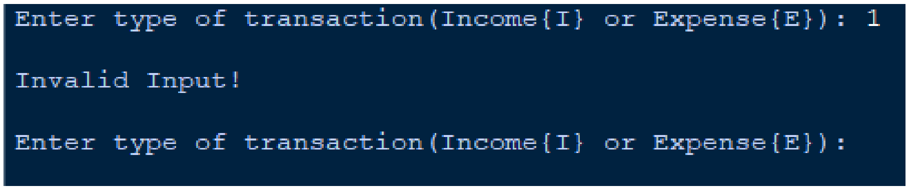
```
Enter category:

Invalid Input!

Enter category:
```

Figure 6



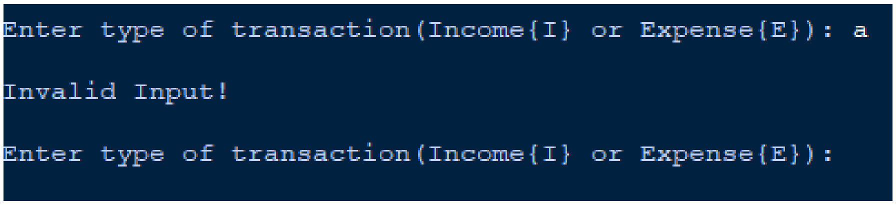
```
Enter category: Salary
Enter type of transaction(Income{I} or Expense{E}):
```

Figure 7



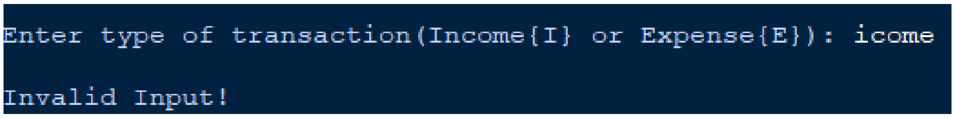
```
Enter type of transaction(Income{I} or Expense{E}): 1

Invalid Input!

Enter type of transaction(Income{I} or Expense{E}):
```

Figure 8



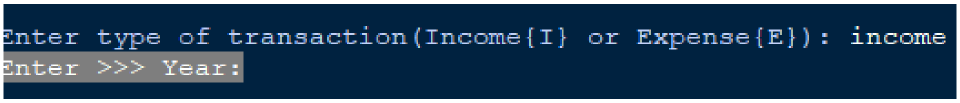
```
Enter type of transaction(Income{I} or Expense{E}): a

Invalid Input!

Enter type of transaction(Income{I} or Expense{E}):
```

Figure 9



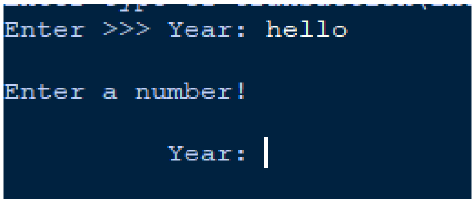
```
Enter type of transaction(Income{I} or Expense{E}): icome

Invalid Input!
```

Figure 10



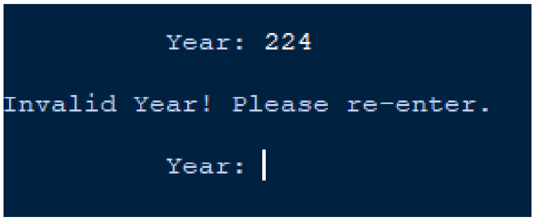
```
Enter type of transaction(Income{I} or Expense{E}): income
Enter >>> Year:
```

Figure 11



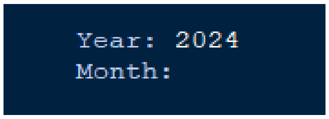
```
Enter >>> Year: hello

Enter a number!

          Year:
```

Figure 12



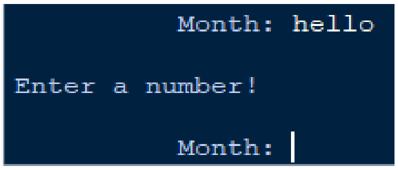
```
          Year: 224

Invalid Year! Please re-enter.

          Year:
```

Figure 13



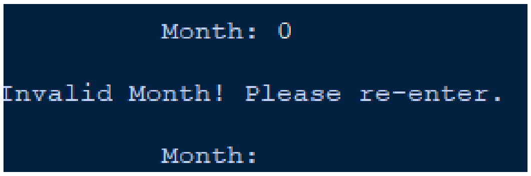
```
     Year: 2024
     Month:
```

Figure 14



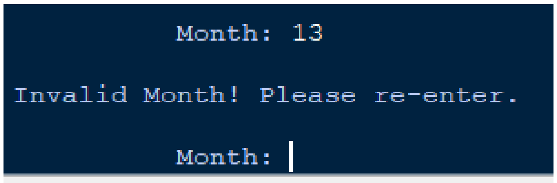
```
        Month: hello

Enter a number!

        Month:
```

Figure 15



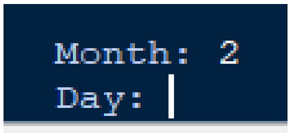
```
        Month: 0

Invalid Month! Please re-enter.

        Month:
```

Figure 16



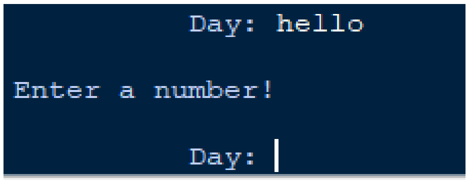
```
        Month: 13

Invalid Month! Please re-enter.

        Month:
```

Figure 17



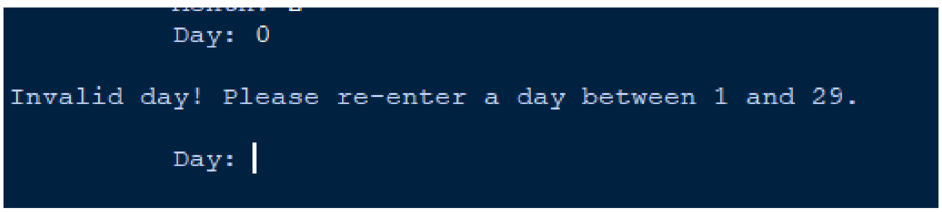
```
  Month: 2
  Day:
```

Figure 18



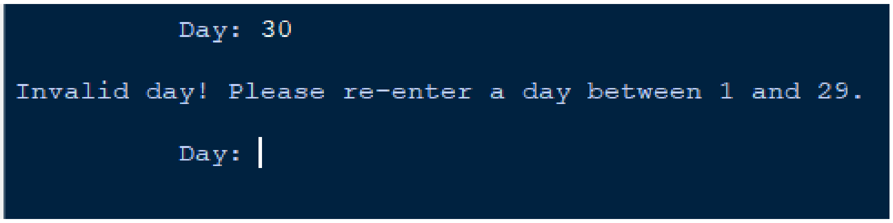
```
        Day: hello

Enter a number!

        Day:
```

Figure 19



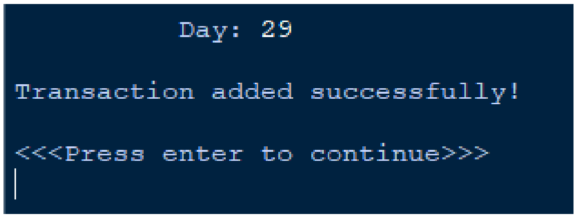
```
        Month: 2
        Day: 0

Invalid day! Please re-enter a day between 1 and 29.

        Day:
```

Figure 20



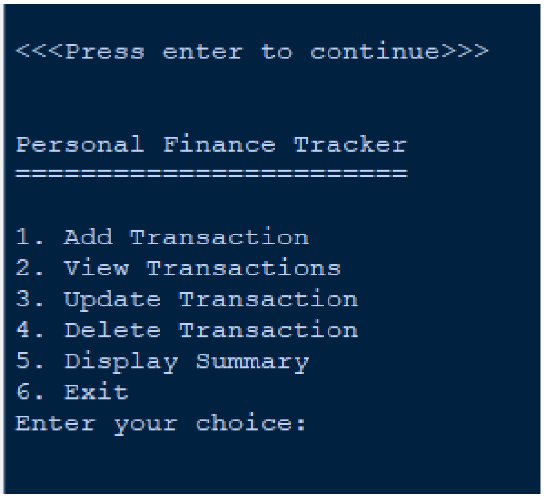
```
          Day: 30

Invalid day! Please re-enter a day between 1 and 29.

          Day:
```

Figure 21



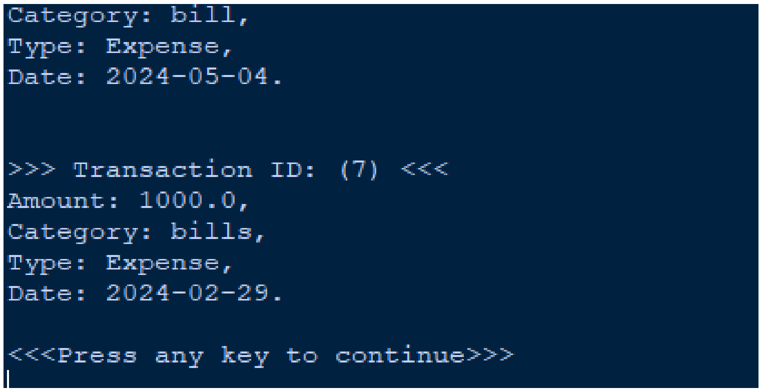
```
          Day: 29

Transaction added successfully!

<<<Press enter to continue>>>
```

Figure 22



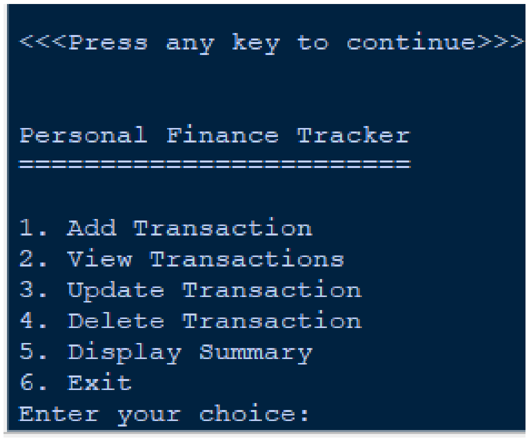
```
<<<Press enter to continue>>>


Personal Finance Tracker
========================

1. Add Transaction
2. View Transactions
3. Update Transaction
4. Delete Transaction
5. Display Summary
6. Exit
Enter your choice:
```

Figure 23



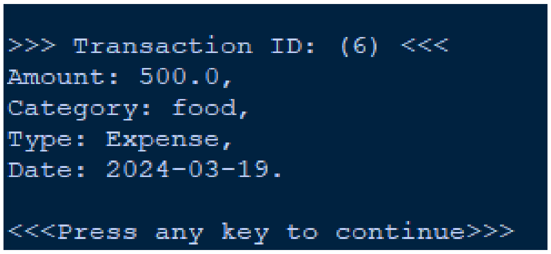
```
Category: bill,
Type: Expense,
Date: 2024-05-04.


>>> Transaction ID: (7) <<<
Amount: 1000.0,
Category: bills,
Type: Expense,
Date: 2024-02-29.

<<<Press any key to continue>>>
```

Figure 24



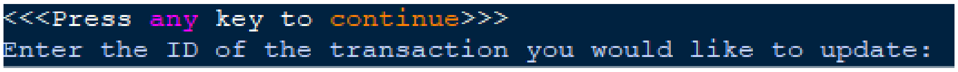
```
<<<Press any key to continue>>>


Personal Finance Tracker
========================

1. Add Transaction
2. View Transactions
3. Update Transaction
4. Delete Transaction
5. Display Summary
6. Exit
Enter your choice:
```

Figure 25



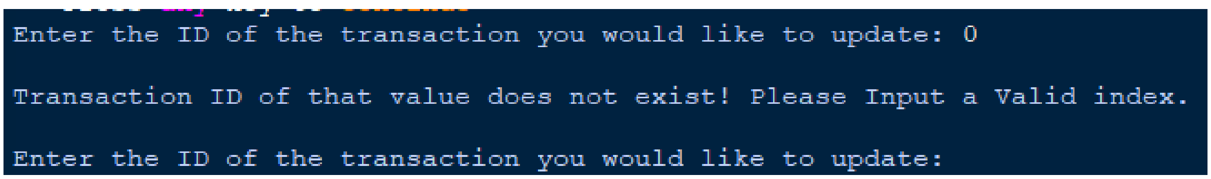
```
>>> Transaction ID: (6) <<<
Amount: 500.0,
Category: food,
Type: Expense,
Date: 2024-03-19.

<<<Press any key to continue>>>
```

Figure 26



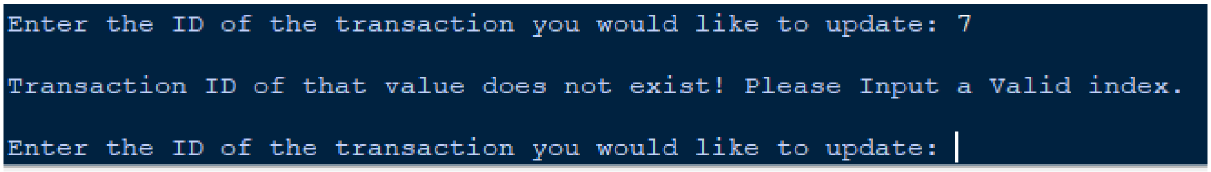
```
<<<Press any key to continue>>>
Enter the ID of the transaction you would like to update:
```

Figure 27



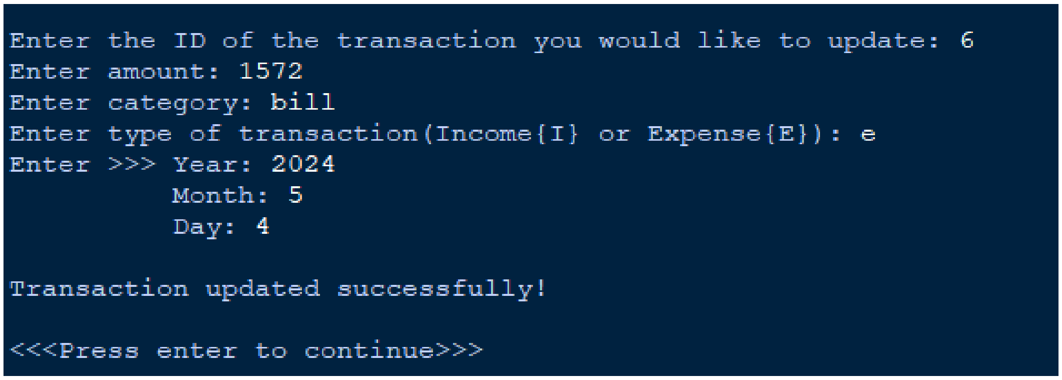
```
Enter the ID of the transaction you would like to update: 0

Transaction ID of that value does not exist! Please Input a Valid index.

Enter the ID of the transaction you would like to update:
```

Figure 28



```
Enter the ID of the transaction you would like to update: 7

Transaction ID of that value does not exist! Please Input a Valid index.

Enter the ID of the transaction you would like to update:
```

Figure 29



```
Enter the ID of the transaction you would like to update: 6
Enter amount: 1572
Enter category: bill
Enter type of transaction(Income{I} or Expense{E}): e
Enter >>> Year: 2024
          Month: 5
          Day: 4

Transaction updated successfully!

<<<Press enter to continue>>>
```

Figure 30