



**INFORMATICS
INSTITUTE OF
TECHNOLOGY**

INFORMATICS INSTITUTE OF TECHNOLOGY

In collaboration with

UNIVERSITY OF WESTMINSTER

Test Plan Document

Enhanced Personal Finance Tracker (GUI Implementation with Tkinter and OOP)

Module: Software Development 1

An Individual Assignment by

Mr. G.S.Karnuathilaka

Supervised by

Mr. Guhanathan Poravi

April 29, 2024

Table of Contents

List of Tables	III
List of Figures	IV
Testing.....	8
1.1 Objectives and Goals of testing	8
1.2 Test Cases for Loading JSON file.....	8
1.2.1 Temporary function modifications for test-phase.....	8
1.2.2 Executing Test Cases	9
1.2.3 Test Results Snapshots.....	11
1.3 Test cases for writing to JSON file	12
1.3.1 Temporary function modifications for test-phase.....	12
1.3.2 Executing Test Cases	13
1.3.3 Test Results Snapshots.....	13
1.4 Test Cases for Main Menu.	14
1.4.1 Test Results Snapshots.....	18
1.5 Test cases for Add new transaction.....	22
1.5.1 Test Results Snapshots.....	26
1.6 Test cases for Viewing Transaction Details in GUI	27
1.6.1 Testing Results Snapshots.....	27
1.7 Test cases for Update Transaction	29
1.7.1 Temporary Modifications Done for Test-phase.....	29
1.7.2 Executing Test Cases	29
1.7.3 Testing Results Snapshots.....	32
1.8 Test cases for Delete transaction.....	33
1.8.1 Temporary Modifications Done for Test phase	33
1.8.2 Executing Test Cases	34
1.8.3 Test Results Snapshots.....	36

1.9 Test cases for Display Summary.....	38
1.9.1 Test Results Snapshots.....	40
1.10 Read Bulk Transaction Test Cases.....	40
1.10.1 Temporary Modifications Done for Test-phase.....	40
1.10.2 Executing Test Cases	43
1.10.3 Testing Results Snapshots.....	46
1.11 Test Cases for Searching in the GUI.....	48
1.11.1 Test Cases for Search by Date	48
1.11.2 Test Cases for Search by Amount.....	53
1.11.3 Test Cases for Search By Transaction Type	56
1.11.4 Test Cases for Search By Category Type	59
1.12 Test Cases for Sorting in GUI.....	63
1.12.1 Testing Results Snapshots.....	65
1.13 TEST SUMMARY	67

List of Tables

Table 1: Test Cases for Loading Transactions.....	11
Table 2: Test Cases for writing to JSON file:.....	13
Table 3: Test Cases for Main Menu.....	18
Table 4: Test Cases for Add New Transaction	25
Table 5: Test Cases for Viewing Transactions	27
Table 6: Test Cases for Update Transaction	31
Table 7: Test Cases for Delete Transaction	35
Table 8: Test Cases for Display Summary.....	39
Table 9: Test Cases for Reading Bulk Transactions	46
Table 10: Test Cases for Search by Date	51
Table 11: Test Cases for Search by Amount.....	55
Table 12: Test Cases for Search by Transaction Type.....	59
Table 13: Test Cases for Search by Category Type.....	62
Table 14: Test Cases for Sorting.....	64
Table 15: Test Summary Results	68

List of Figures

Figure 1: Temporary Adjustments for Loading JSON files.....	8
Figure 2: Changing the file path - Loading JSON file.....	9
Figure 3: Incorrectly formatted JSON file	9
Figure 4: Test Result for Test case 01 - Load transactions from JSON file	11
Figure 5: Test Results for Test Case 02- Loading transactions from JSON file.....	11
Figure 6: File Creation - Test Case 02 - Loading transactions from JSON file.....	11
Figure 7: No Existing Transactions - Test Case 03 - Loading Transactions from JSON file:.....	12
Figure 8: Changing the File Path- Writing to JSON file.....	12
Figure 9: Transaction save - Test Case 01 - Writing to JSON file	13
Figure 10: Test Results - Test Case 02 - Writing to a JSON file.....	14
Figure 11: File Creation - Test Case 02- Writing to a JSON file.....	14
Figure 12: Navigation: Main Menu redirecting to Add New Transaction.....	18
Figure 13: Navigation: Main Menu redirecting to View Transaction	18
Figure 14: Navigation: Main Menu redirecting to Update Transaction.....	19
Figure 15: Navigation: Main Menu redirecting to Delete Transaction.....	19
Figure 16: Navigation: Main Menu redirecting to Display Summary	20
Figure 17: Navigation: Main Menu redirecting to Read Bulk Transaction	20
Figure 18: Input Validation - Main Menu Test Cases	21
Figure 19: Empty Input - Test Case 12 - Main Menu Test Cases.....	22
Figure 20: Input Validation - Add New Transaction Test Cases	26
Figure 21: Input Validation and Storing Transaction Details - Add New Transaction Test Cases	26
Figure 22: Transactions displayed on the GUI	27
Figure 23: No Existing Transactions – User Message	28
Figure 24: No Existing Transactions – GUI – Test Case 02 – Viewing Transaction Details in GUI	28
Figure 25: Temporary Modifications - Update Transaction	29
Figure 26: Invalid Inputs - Update Transaction	32
Figure 27: Transaction ID capturing the Correct Transaction	32
Figure 28: Successful Transaction Details Update	33

Figure 29: No Existing Transactions - Update Transaction Test Cases	33
Figure 30: Temporary Modifications for Delete Transaction	33
Figure 31: Invalid Inputs for the Transaction ID - Delete Transaction Test Cases	36
Figure 32: Transaction ID capturing the correct Transaction – Delete Transaction Test Cases ..	37
Figure 33: Successful Transaction Deletion - Delete Transaction Test Cases	37
Figure 34: Summary of overall Transactions - Display Transaction Summary	40
Figure 35: Changing Transaction Category to numeral character - Test Case 02 -Reading Bulk Transactions	41
Figure 36: Changing Transaction Category to special character - Test Case 03 -Reading Bulk Transactions	41
Figure 37: Changing Transaction Amount to textual character - Test Case 04 -Reading Bulk Transactions	41
Figure 38: Changing Transaction Amount to Special character - Test Case 05 -Reading Bulk Transactions	41
Figure 39: Changing Transaction Date to Invalid date - Test Case 06 -Reading Bulk Transactions	42
Figure 40: Changing Transaction Date to Text - Test Case 07 -Reading Bulk Transactions.....	42
Figure 41: Changing Transaction Date to a Number - Test Case 08 -Reading Bulk Transactions	42
Figure 42: Changing Transaction Type to Numeral Character - Test Case 09 -Reading Bulk Transactions	42
Figure 43: Changing Transaction Type to Special Character - Test Case 09 -Reading Bulk Transactions	43
Figure 44: Recording Bulk Transactions - Test Case 11 - Reading Bulk Transactions	43
Figure 45: Input Validation – Transaction Category - Test Case 02 - Read Bulk Transactions ..	46
Figure 46: Input Validation – Transaction Category - Test Case 03 - Read Bulk Transactions ..	47
Figure 47: Input Validation – Transaction Amount - Test Case 04 - Read Bulk Transactions	47
Figure 48: Input Validation – Transaction Amount - Test Case 05 - Read Bulk Transactions	47
Figure 49: Input Validation – Transaction Date - Test Case 06 - Read Bulk Transactions.....	47
Figure 50: Input Validation – Transaction Date - Test Case 07 - Read Bulk Transactions.....	47
Figure 51: Input Validation – Transaction Date - Test Case 08 - Read Bulk Transactions.....	48

Figure 52: Input Validation – Transaction Type - Test Case 09 - Read Bulk Transactions	48
Figure 53: Input Validation – Transaction Type - Test Case 10 - Read Bulk Transactions	48
Figure 54: Successful Implementation of Reading Bulk Transaction - Test Case 11 - Read Bulk Transactions:	48
Figure 55: 'Select By Date' option in 'Search' Menu	49
Figure 56: 'Search By Date' Window	49
Figure 57: Test Result for Test Cases 01, 02, 03, 04, 05, and 06 – Search By Date	52
Figure 58: Test Result for Test Case 07 - Search By Date	52
Figure 59: Test Action for Test Case 08 - Search By Date	52
Figure 60: Test Result for Test Case 08 - Search By Date	53
Figure 61: 'Select By Amount' option in 'Search' Menu	53
Figure 62: 'Search By Amount' Window	54
Figure 63: Test Results for Test Cases 01, 02, 03, and 04 – Search By Amount	55
Figure 64: Test Result for Test Case 05 - Search By Amount	56
Figure 65: 'Search By Transaction Type' option in 'Search' Menu	56
Figure 66: 'Search By Transaction Type' Window	57
Figure 67: Test Result for Test Cases 01,02,03,04 and 05 – Search By Transaction Type	59
Figure 68: Test Result for Test Case 06 – Search By Transaction Type	59
Figure 69: 'Search By Category Type' option in 'Search' Menu	60
Figure 70: 'Search By Category Type' Window	60
Figure 71: Test Result - Test Case 01- Search By Category Type	62
Figure 72: Test Result - Test Case 02 - Search By Category Type	62
Figure 73: Test Result for Test Cases 03, 04, 05- Search By Category Type	63
Figure 74: Test Result for Test Case 06 - Search By Category Type	63
Figure 75: ID and amount sorted in ascending order - Test Result - Test Case 01- Sorting on GUI	65
Figure 76: Date sorted in ascending order - Test Result - Test Case 01- Sorting on GUI	65
Figure 77: Transaction Category and Type Sorted in Ascending Order – Test Result – Test Case 01- Sorting on GUI	66
Figure 78: Transaction ID and Amount Sorted in Descending Order - Test Result - Test Case 02- Sorting on GUI	66

Figure 79: Transaction Category and Type Sorted in Descending Order - Test Result - Test Case 02- Sorting on GUI	67
Figure 80: Transaction Date Sorted in Descending Order - Test Result - Test Case 02- Sorting on GUI	67

Testing

1.1 Objectives and Goals of testing

The main purpose of using the testing phase is to validate the functionality of the code that had been created based on the designed solution and the considered requirements mentioned in the problem statement. The key goals of the testing and their objectives are given below.

1. Satisfying the functional requirements mentioned in the problem statement.
2. Detect the system bugs and loopholes of the created code.
3. To make some further improvements according to the test results that were gathered during the testing phase.

1.2 Test Cases for Loading JSON file.

1.2.1 Temporary function modifications for test-phase

Function adjustments were introduced to confirm test case results and evaluate the program's smooth operation without errors, involving temporary modifications in the following areas:

- Print statements were implemented to verify the loading and storage of existing transactions (JSON data) in the program's data structure to test the outcome in Test Case 01

```
1 usage
def load_transaction(FILE_PATH):
    try:
        with open(FILE_PATH, 'r') as file:
            transactions = json.load(file)
            print(transactions)
            print('Transaction Data Loaded Successfully')
        return transactions
```

Figure 1: Temporary Adjustments for Loading JSON files

- File path was changed to test the expected outcome in Test Case 02

```

4 FILE_PATH = 'test.json'
5 TEXT_FILE_PATH = 'text_transaction.txt'
6
7 DATE_FORMAT = "%Y-%m-%d" # YYYY-MM-DD date format for transactions
8 SPECIAL_CHARS = '\~!#$%^*()_+~={}|[]\:\<>?,./'
9
10 transactions = {}

```

Figure 2: Changing the file path - Loading JSON file

- The JSON file was incorrectly formatted in order to test the expected outcome in Test Case 03.

```

{
  "Sales": [
    {
      "amount": 25000.0,
      "date": "2024-12-05",
      "type": "income",
      "id": 0
    }
  ]
}

```

Figure 3: Incorrectly formatted JSON file

1.2.2 Executing Test Cases

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Loading with JSON file available in the same directory	Load transactions with a JSON file	The transactions should be loaded successfully from the JSON file and stored in the	The transactions successfully loaded and stored in the program's data structure and a message :	Pass

			program's data structure.	displayed on the screen	
				System displayed the existing transactions on to the screen	
02	Loading with no JSON file available in the same directory	Attempt to load transactions from a JSON file that does not exist in the current directory	The program should prompts user for creating an empty file	The program prompted user to create an empty file	Pass
03	Loading with format error in JSON file	Prepare a JSON file with incorrect formatting, such as missing commas, or other syntax errors	Program should detect the format error in the JSON file and It should display an appropriate error message.	The program displayed an error message	Pass
		Attempt to load transactions from the			

		erroneous JSON file			
--	--	------------------------	--	--	--

Table 1: Test Cases for Loading Transactions

1.2.3 Test Results Snapshots

```
Transaction Data: {'Sales': [{'amount': 25000.0, 'date': '2024-12-05', 'type': 'Income', 'id': 0}]}
Transaction Data loaded Successfully
```

Figure 4: Test Result for Test case 01 - Load transactions from JSON file

In the Figure 04,

- System displayed the loaded transactions on the screen.
- A message was displayed by the system to inform user about the successful data load.

```
File Does not exist
Do you wish to create an empty File? (Yes / No): YES
Enter a file_name for the file: Test_transaction_file
File Test_transaction_file created successfully in the directory

Process finished with exit code 0
```

Figure 5: Test Results for Test Case 02- Loading transactions from JSON file

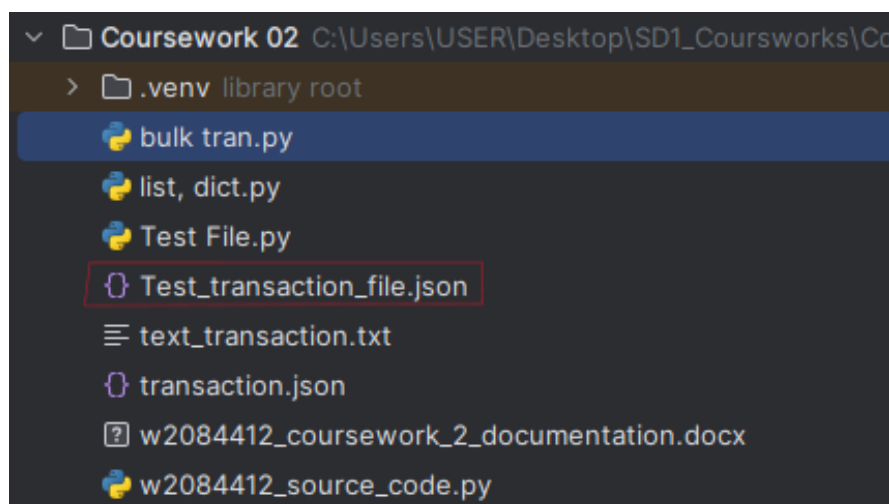


Figure 6: File Creation - Test Case 02 - Loading transactions from JSON file

In the Figure 05 and 06,

- As shown in Figure 02, the file name was changed.
- The system generated an error message: '*File Does Not Exist*'
- The system prompted user to create a new file.
- After the user's confirmation for creating a new file, '*Test_transaction_file*' was given as the File Name.
- As shown in the Figure 6, the system created the File '*Test_transaction_file*' and displayed a message notifying the user.

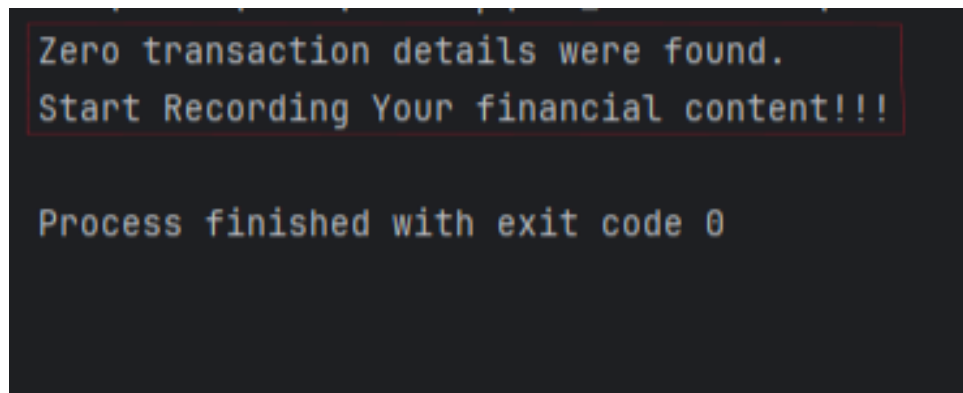


Figure 7: No Existing Transactions - Test Case 03 - Loading Transactions from JSON file:

1.3 Test cases for writing to JSON file

1.3.1 Temporary function modifications for test-phase

- File path was changed to test the expected outcome in Test Case 02

```
4 FILE_PATH = 'test.json'
5 TEXT_FILE_PATH = 'text_transaction.txt'
6
7 DATE_FORMAT = "%Y-%m-%d" # YYYY-MM-DD date format for transactions
8 SPECIAL_CHARS = '\~!#$%^*()_+-={}|[]\:\<>?,./'
9
10 transactions = {}
```

Figure 8: Changing the File Path- Writing to JSON file

1.3.2 Executing Test Cases

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Saving data with JSON file available	Save a dictionary of Transaction data with JSON file available	System is expected to save the transaction data to the JSON file.	The transaction data was successfully saved to the JSON file without any errors	Pass
02	Loading with no JSON file available in the same directory	Attempt to load transactions from a JSON file that does not exist in the current directory	The program should save the transactions in the changed JSON file	The program created a JSON file and transactions were saved in the created JSON file.	Pass

Table 2: Test Cases for writing to JSON file:

1.3.3 Test Results Snapshots

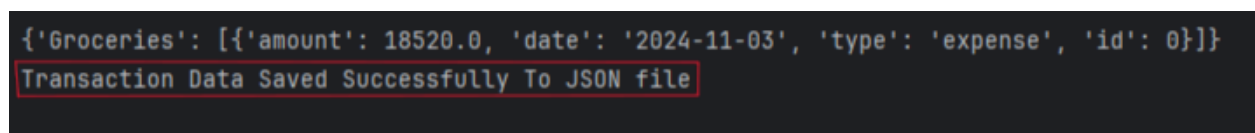


Figure 9: Transaction save - Test Case 01 - Writing to JSON file

In the Figure 9,

- The System successfully saved the transaction details to the JSON file and displayed the saved transactions on the screen.
- User was notified about the successful transaction save.

Transaction Data Saved Successfully To JSON file

Figure 10: Test Results - Test Case 02 - Writing to a JSON file

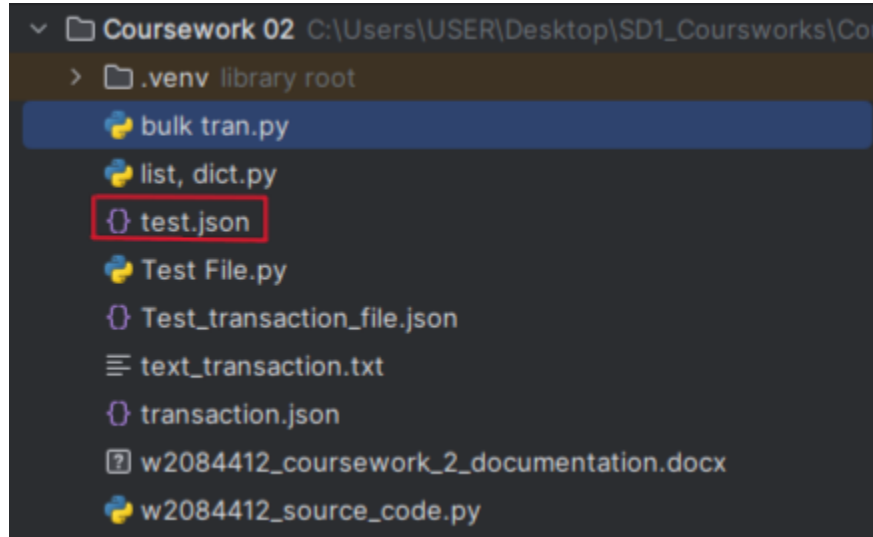


Figure 11: File Creation - Test Case 02- Writing to a JSON file

In the Figure 11,

- As shown in the Figure 8, The file name was changed into '*test.json*'
- As shown in the Figure 11, System created a JSON file under the name '*test.json*' and saved the transactions into the created file.
- A message was displayed by the system to inform the user about the Successful data save.

1.4 Test Cases for Main Menu.

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying the transition from the Main menu function to the 'Add	User inputting '1' as the choice	User is to be redirected to Add New Transaction after prompting '1'	User was redirected to add a new transaction after prompting 1	Pass

	Transaction' function				
02	Verifying the transition from the Main menu function to the 'View Transaction' function	User inputting '2' as the choice	User is to be redirected to View Transaction after prompting '2'	User was redirected to view existing transactions after prompting 2	Pass
03	Verifying the transition from the Main menu function to the 'Update Transaction' function	User inputting '3' as the choice.	User is to be redirected to Update details of the existing transaction after prompting '3'	User was redirected to update details of existing transaction after prompting 3	Pass
04	Verifying the transition from the Main menu function to the 'Delete Transaction' function	User inputting '4' as the choice	User is to be redirected to Delete Transaction after prompting '4'	User was redirected to delete existing transaction after prompting '4'	Pass
05	Verifying the transition from the Main menu function to the 'Display Summary' function	User inputting '5' as the choice	User is to be redirected to Display Summary after prompting '5'	User was redirected to view summary of existing transactions after prompting '5'	Pass

06	Verifying the transition from the Main menu function to the 'Read Bulk transaction' function	User inputting '6' as the choice	User is to be redirected to Read Bulk Transactions after prompting '6'	User was redirected to read bulk transactions from the Text File after prompting '6'	Pass
07	Presenting an error message to the user if a negative integer is entered as their choice	The user inputting a randomly chosen negative integer (-1523) as their selection	The system is expected to exhibit a clear error message to the user, indicating the encountered Issue and prompting the user to re-enter the choice	The system presented a clear error message and prompted the user to re-enter the choice	Pass
08	Presenting an error message to the user in case a text(string input) is provided as the choice	User inputting a randomly chosen text (choice) as the selection	The system is expected to present an error message and prompt the user to enter a choice	The system presented a clear error message and prompted the user to re-enter the choice	Pass
09	Presenting an error message to the user when they	User inputting two specific	The system is expected to exhibit a clear error message to	The system presented a clear error message and	Pass

	input a numeric choice outside the valid range of 1 to 7	values: 0 and 10.	the user, indicating the encountered Issue and prompting the user to re-enter the choice.	prompted the user to re-enter the choice.	
10	Presenting an error message to the user when they input a special character for choice	User inputting a sequence of special characters (@#\$%)	The system is expected to exhibit a clear error message to the user indicating the encountered issue and prompting the user to re-enter the choice	The system displayed an error message and prompted user to re-enter the choice	Pass
11	Terminating the system at the main menu	User inputting '7' as their choice	The user is expected to exit the system, leading to the termination of the program.	The system presented a clear message about the program termination: "Exiting Program" and user was logged out of the system	Pass
12	Verifying an Empty Input for the choice	User pressing 'Enter' key for an input	System is expected to display an error message and	System displayed an error message and user was prompted to enter the choice	Pass

			prompts user for choice		
--	--	--	-------------------------	--	--

Table 3: Test Cases for Main Menu

1.4.1 Test Results Snapshots

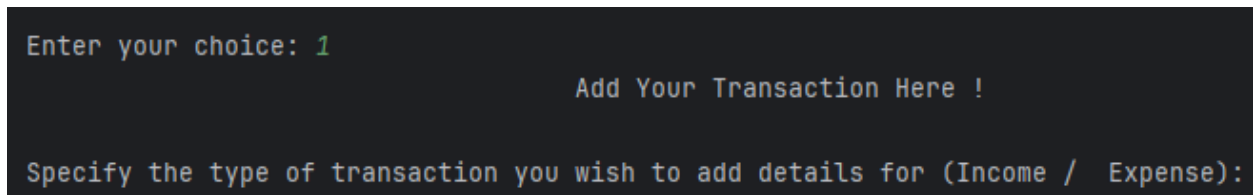


Figure 12: Navigation: Main Menu redirecting to Add New Transaction

- As shown in Figure 12, system re-directed user to add a new transaction to the system.

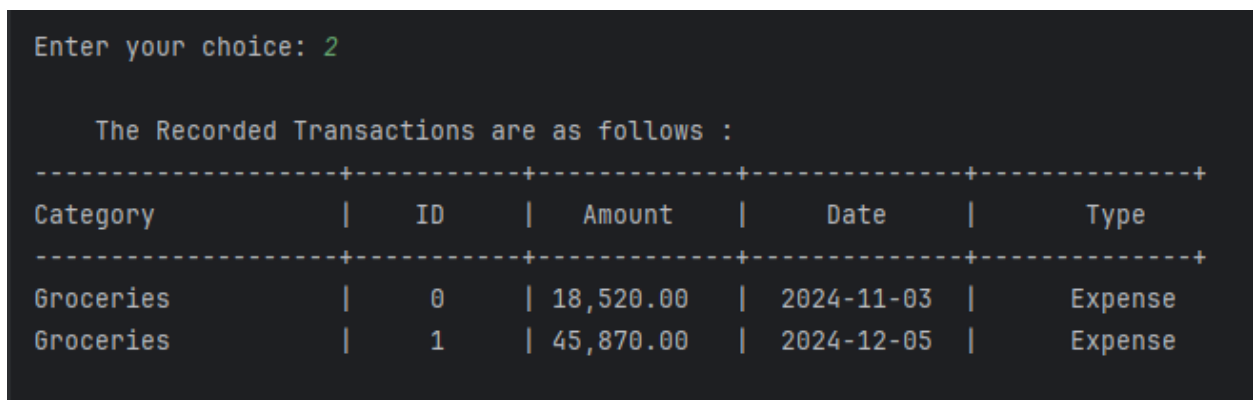


Figure 13: Navigation: Main Menu redirecting to View Transaction

- In Figure 13, system re-directed user to view the existing transactions in the system.

```
Enter your choice: 3

The Recorded Transactions are as follows :
-----+-----+-----+-----+
Category      | ID    | Amount  | Date      | Type
-----+-----+-----+-----+
Groceries     | 0     | 18,520.00 | 2024-11-03 | Expense
Groceries     | 1     | 45,870.00 | 2024-12-05 | Expense

Update your Transaction here!

Please enter the Transaction ID of the transaction you wish to update:
```

Figure 14: Navigation: Main Menu redirecting to Update Transaction

- In Figure 14, system re-directed user to update the details of the existing transactions in the system.

```
Enter your choice: 4

The Recorded Transactions are as follows :
-----+-----+-----+-----+
Category      | ID    | Amount  | Date      | Type
-----+-----+-----+-----+
Groceries     | 0     | 18,520.00 | 2024-11-03 | Expense
Groceries     | 1     | 45,870.00 | 2024-12-05 | Expense

Delete Your Transactions here!

Please enter the Transaction ID of the transaction you wish to delete: |
```

Figure 15: Navigation: Main Menu redirecting to Delete Transaction

- In Figure 15, system re-directed user to delete an existing transaction in the system.

```
Enter your choice: 5
-----+-----+-----+
                        Transaction Summary
-----+-----+-----+
Category      |   Amount   | Category Total
-----+-----+-----+
Groceries
              |  18,520.00 |  18,520.00
              |  45,870.00 |  64,390.00
-----+-----+-----+
Category Total (Expense):                        64,390.00
-----+-----+-----+
Total Income:                                     0.00
Total Expenses:                                  64,390.00
Net Balance:                                    -64,390.00
-----+-----+-----+
```

Figure 16: Navigation: Main Menu redirecting to Display Summary

- In Figure 16, system re-directed user to view the summary of existing transactions in the system.

```
Enter your choice: 6

Read Bulk Transactions from Text File
```

Figure 17: Navigation: Main Menu redirecting to Read Bulk Transaction

- In the Figure 17, system re-directed user to read bulk transaction details from a text file and save those details to the JSON file.

```
Enter your choice: -1523
Input cannot be a negative value → Test Case 07

Enter your choice: choice
Invalid input format. Please enter a valid input. → Test Case 08

Enter your choice: 0
Input out of range. Please try again. → Test Case 09

Enter your choice: 10
Input out of range. Please try again. → Test Case 09

Enter your choice: @#$%
Invalid input format. Please enter a valid input. → Test Case 10

Enter your choice: 7
Exiting program. → Test Case 11

Process finished with exit code 0
|
```

Figure 18: Input Validation - Main Menu Test Cases

In the Figure 18,

- For a randomly chosen negative integer (-1523), the system generated an error: *'Input cannot be a negative value'*. The user was then prompted to re-enter a choice.
- For a randomly chosen string input (choice), the system generated an error: *'Invalid input. Please enter a valid input'*. The user was then prompted to re-enter a choice.
- The entry of numbers 0 and 10, which fall outside the valid range of 1 to 6, resulted in an error message: *'Input out of range. Please try again'*. Afterward, the user was prompted to re-enter a valid choice.
- For a randomly chosen special characters sequence (@#\$ %), the system generated an error: *'Invalid input format. Please enter a valid input'*. The user was then prompted to re-enter a choice.
- When user enters number 7, system displayed a message *'Exiting program'* and the user was logged out of the application and the program terminated

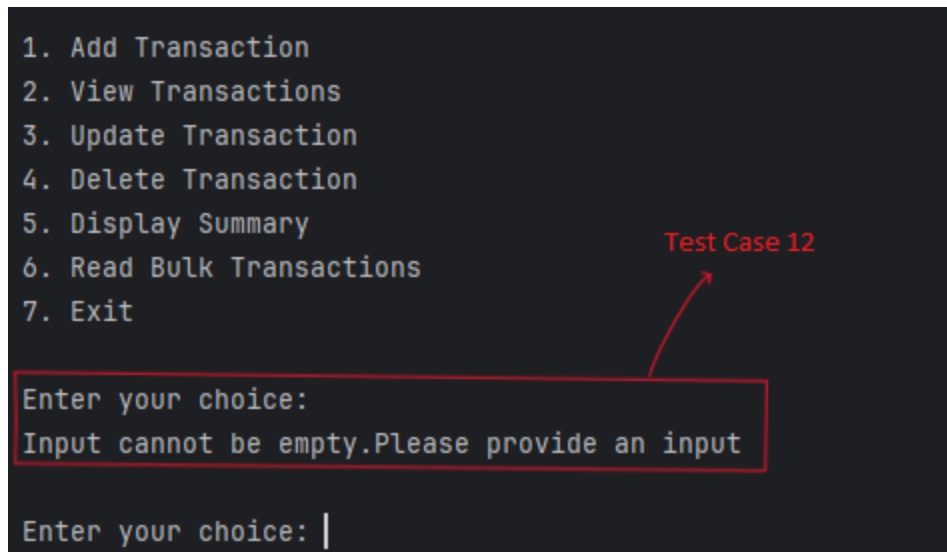


Figure 19: Empty Input - Test Case 12 - Main Menu Test Cases

1.5 Test cases for Add new transaction

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying the input of the transaction type is either "income" or "expense"	User inputting a random text (asset) as the input	The system should display a message and prompt the user to enter the input	System displayed an error message and prompted the user to enter the transaction type	Pass
02	Verifying the input of the transaction type is either "income" or "expense"	User inputting a sequence of special characters (##^&\$@) as the input	The system should display a message and prompt the user to enter the input	System presented an error message and prompted the user to enter the transaction type	Pass

03	Presenting an error message to the user when they input a numeral input for transaction type	User inputting a random number (850) as an input	The system is expected to present an error message to the user, and prompting the user to re-enter the transaction type	System displayed an error message and prompted the user to enter the transaction type	Pass
04	Verifying an Empty Input for the transaction type	User pressing 'Enter' key for an input	System is expected to display an error message and prompts user for transaction type	System displayed an error message and user was prompted to enter the transaction type	Pass
05	Verifying the input of a number (numeral input) for the category type	User inputting a random number (150) as an input	The system is expected to present an error message to the user, and prompting the user to re-enter the category type	System showed an error message and prompted the user to enter the category type	Pass
06	Verifying the input of a special characters for the category type	User inputting a sequence of special characters (##^&\$@)	The system should display an error message and prompt the user	A message was displayed and prompted the user to enter a valid category type	Pass

		as the category type	to enter the category type		
07	Verifying the input of a text (string) for the transaction amount	User inputting a random text (thousand) as the input	The system should display a message and prompt the user to enter the transaction amount	A message was displayed and prompted the user to enter a valid transaction amount	Pass
08	Verifying the input of a negative number for the transaction amount	User inputting a randomly chosen negative number (-1520) as the transaction amount	The system should display an error message to the user and prompt user to enter a valid transaction amount	A message was displayed and prompted the user to enter a valid transaction amount	Pass
09	Verify the transaction date in the correct YYYY-MMDD format	User entering a random date (2024-15-05)	The system is expected to display an error message and prompt the user to enter the transaction_date	The system displayed an error message and user was prompted to enter the transaction_date	Pass
10	Verify the transaction date in the correct	User inputting a randomly chosen	The system is expected to display an error message and	The system displayed an error message and user was	Pass

	YYYY-MMDD format	negative number (- 4580) as the transaction amount	prompt the user to enter the transaction_date	prompted to enter the transaction_date	
11	Verify the transaction date in the correct YYYY-MM- DD format	User entering a random text (15 th June) as the date	The system is expected to display an error message and prompt the user to enter the transaction_date	The system displayed an error message and user was prompted to enter the transaction_date	Pass
12	Capturing and storing transaction details	User inputting precise details for each requested Input and selected 'yes' to save the details.	The system is expected to provide a message to the user indicating that the Transaction has been saved	System displayed a message to the user indicating that the transaction has been saved.	Pass

Table 4: Test Cases for Add New Transaction

1.5.1 Test Results Snapshots

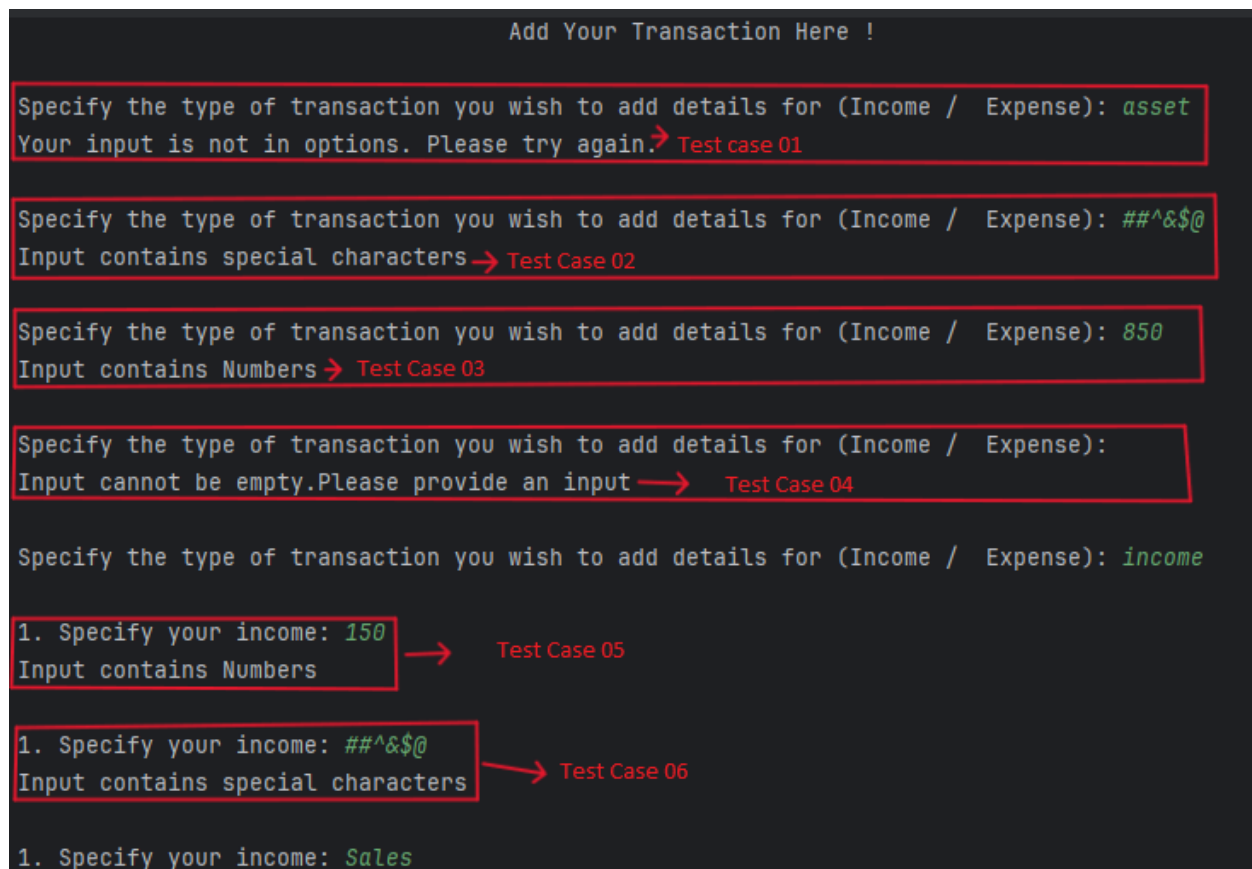


Figure 20: Input Validation - Add New Transaction Test Cases

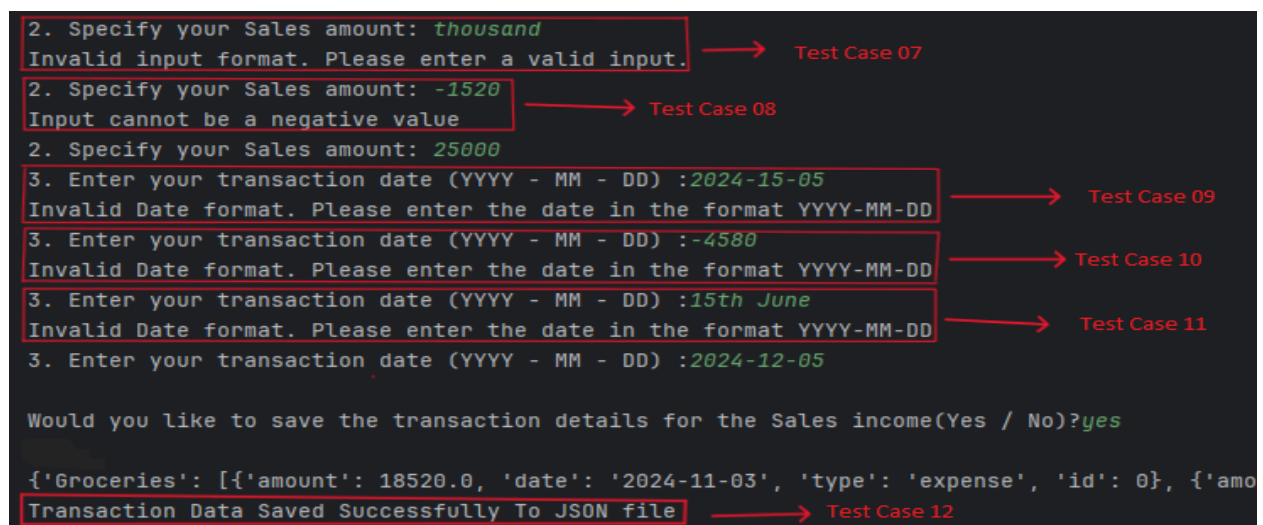


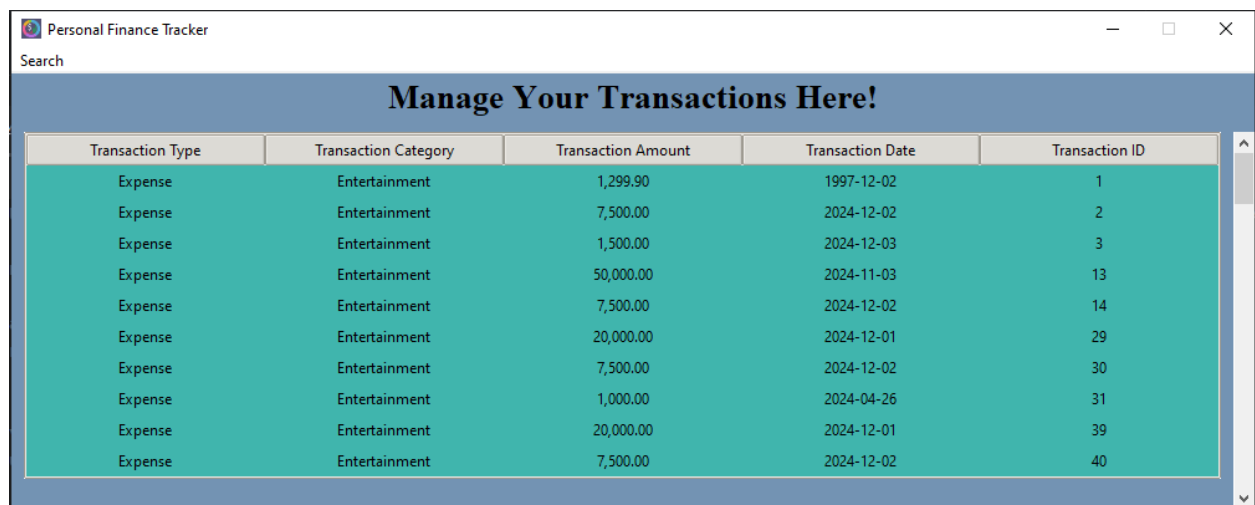
Figure 21: Input Validation and Storing Transaction Details - Add New Transaction Test Cases

1.6 Test cases for Viewing Transaction Details in GUI

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Displaying all the transactions in the JSON file on the GUI	Starts the program	The System is expected to display all the existing transactions in their relevant columns	System displayed the transactions on the GUI	Pass
02	Verifying the system output in the absence of existing transactions	Remove the data in the JSON file and start the program.	The system is expected to display an error message to user	An message was displayed to the user about the absence of existing transactions	Pass

Table 5: Test Cases for Viewing Transactions

1.6.1 Testing Results Snapshots



The screenshot shows a window titled "Personal Finance Tracker" with a search bar and a table of transactions. The table has a header row with columns: Transaction Type, Transaction Category, Transaction Amount, Transaction Date, and Transaction ID. The table contains 10 rows of data, all with "Expense" as the Transaction Type and "Entertainment" as the Transaction Category. The Transaction Amounts range from 1,299.90 to 50,000.00. The Transaction Dates range from 1997-12-02 to 2024-12-02. The Transaction IDs range from 1 to 40.

Transaction Type	Transaction Category	Transaction Amount	Transaction Date	Transaction ID
Expense	Entertainment	1,299.90	1997-12-02	1
Expense	Entertainment	7,500.00	2024-12-02	2
Expense	Entertainment	1,500.00	2024-12-03	3
Expense	Entertainment	50,000.00	2024-11-03	13
Expense	Entertainment	7,500.00	2024-12-02	14
Expense	Entertainment	20,000.00	2024-12-01	29
Expense	Entertainment	7,500.00	2024-12-02	30
Expense	Entertainment	1,000.00	2024-04-26	31
Expense	Entertainment	20,000.00	2024-12-01	39
Expense	Entertainment	7,500.00	2024-12-02	40

Figure 22: Transactions displayed on the GUI

As shown in Figure 22, the existing transactions are showed on the GUI.

- The income type transactions and expense type transactions were represented in two separate colors for the user's ease of identification.

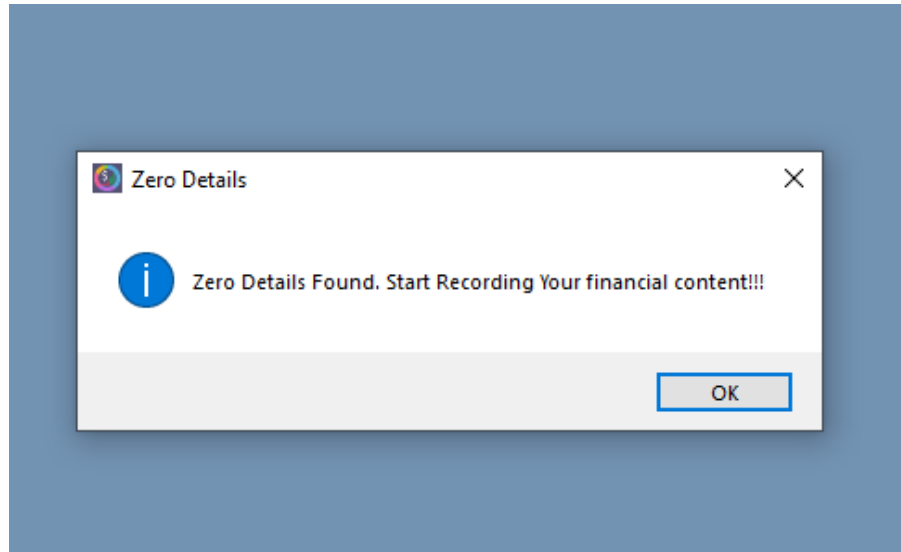


Figure 23: No Existing Transactions – User Message

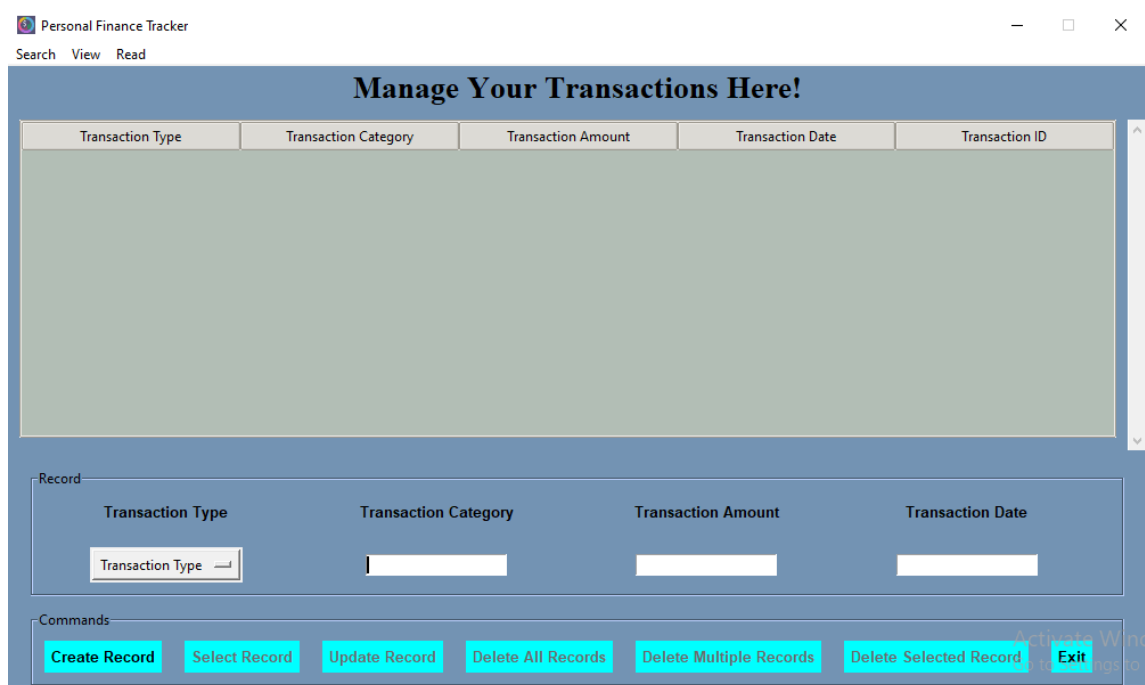


Figure 24: No Existing Transactions – GUI – Test Case 02 – Viewing Transaction Details in GUI

In Figure 23 and 24,

- A message was displayed by the system to inform the user about no existing transactions.
- The transaction table on the GUI became empty.

1.7 Test cases for Update Transaction

1.7.1 Temporary Modifications Done for Test-phase

Function adjustments were introduced to confirm test case results and evaluate the program's smooth operation without errors, involving temporary modifications in the following areas:

- A print statement was applied to verify the displaying the correct details of the transaction relevant to user entered Transaction ID in Test Case 04

```
def update_transaction():
    if empty_transaction(content):
        print("No transactions found. Please record transactions before attempting to update.")
        return

    view_transaction()
    print("\n\tUpdate your Transaction here!".center(100))
    while True:
        tran_id = validate_input( message: '\nPlease enter the Transaction ID of the transaction you wish to update: ', int)
        for category in content:
            for transaction in content[category]:
                if tran_id == transaction['id']:
                    print(f"\nTransaction with the provided Transaction ID: {tran_id} has been found.")
                    print(transaction)
                    new_amount = validate_input( message: f'\n1. Enter the New amount for {category}: ', float)
                    update_date = validate_input( message: '\nDo you want to update the transaction date ? (Yes/No):', str,
                                                  options_list: ['yes', 'no'])
```

Figure 25: Temporary Modifications - Update Transaction

1.7.2 Executing Test Cases

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying the input of a text as the transaction ID	User inputting randomly chosen text (ID) as the transaction ID	System is expected to Display an error message and prompt user to re-enter the transaction ID	An error message was displayed and user was asked to re-enter the transaction ID	Pass

02	Checking the scenario when invalid transaction ID is entered	User inputting a randomly chosen number (10) which is not shown in the transaction details	The system should display an error message and redirect the user to the main menu	System displayed an error message and user was redirected to the main menu	Pass
03	Verifying an Empty Input for the ID	User pressing 'Enter' key for an input	System is expected to display an error message and prompts user for Transaction ID	System displayed an error message and user was prompted to enter the Transaction ID	Pass
04	Verifying that the transaction ID captures the correct transaction	User inputting an already existing transaction ID	System should print the details of the transaction relevant to transaction ID	System displayed a message indicating the Transaction details relevant to transaction ID have been found and displayed the details of the transaction	Pass
05	Verifying a Textual input for	User inputting a random text	System is expected to Display an error	User was asked again to enter the transaction	Pass

	Transaction amount	(Five Hundred) as the transaction amount	message and prompting user for Transaction amount	amount after displaying an error message	
06	Verifying the input of a special characters for the transaction amount	User inputting a sequence of special characters (##@\$) as the transaction amount	System should be displaying an error message and prompt for Transaction amount	An error message displayed and prompted for the Input again.	Pass
07	Verifying the correct update of transaction details	User inputting the precise details for each input and confirms to update the transaction details	System should update the transaction details and write the updated details back to the file	Transaction details were updated and the updated details were saved in the file	Pass
08	Checking the scenario when there are no existing transactions	Remove the data in the JSON file User will input '3' as the choice in main menu	System should Display a message to the user after it check for existing transactions	An error message was displayed and user was redirected to main menu	Pass

Table 6: Test Cases for Update Transaction

1.7.3 Testing Results Snapshots

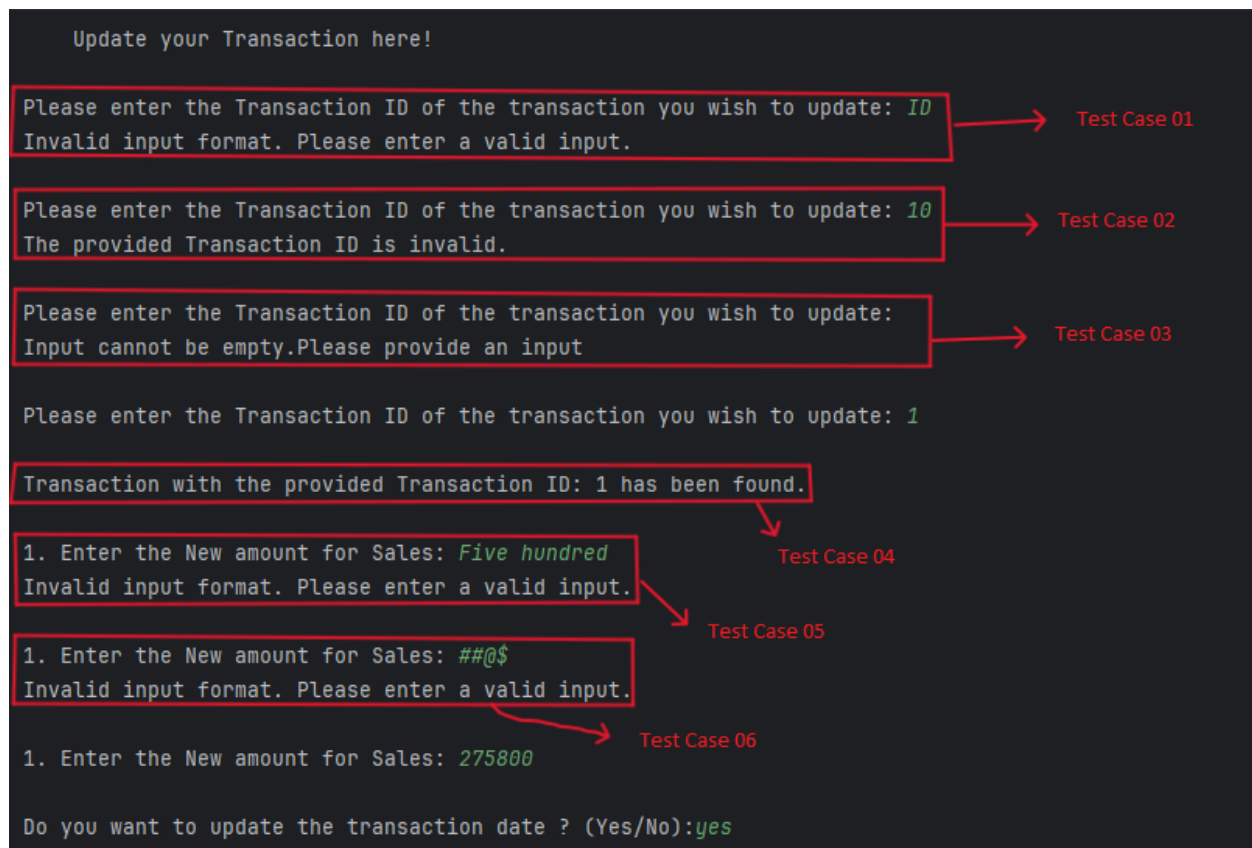


Figure 26: Invalid Inputs - Update Transaction

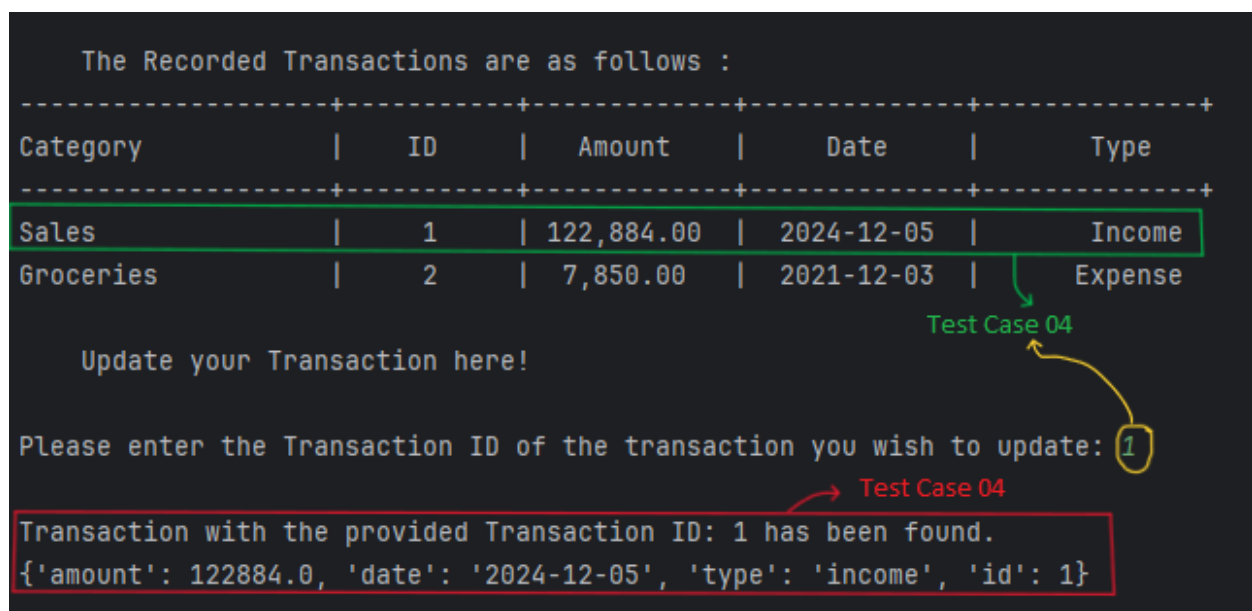


Figure 27: Transaction ID capturing the Correct Transaction

In the Figure 27,

- According to the Test Case 04, System displayed the transactions relevant to the user-entered transaction ID.

```
Do you want to update the transaction details ? (Yes/No):yes
{'Sales': [{'amount': 275800.0, 'date': '2024-12-05', 'type': 'income', 'id': 1}],
Transaction details for Transaction ID: 1 have been successfully updated.
Transaction Data Saved Successfully To JSON file
```

Figure 28: Successful Transaction Details Update

In the Figure 28,

- System displayed the updated transactions.
- A message was displayed by the system to inform the user about the successful update.

```
Enter your choice: 3
No transactions found. Please record transactions before attempting to update.
```

Test Case 08

Figure 29: No Existing Transactions - Update Transaction Test Cases

1.8 Test cases for Delete transaction

1.8.1 Temporary Modifications Done for Test phase

Function adjustments were introduced to confirm test case results and evaluate the program's smooth operation without errors, involving temporary modifications in the following areas:

- A print statement was applied to verify the displaying the correct details of the transaction relevant to user entered Transaction ID in Test Case 04.

```
# Check if the transaction is not empty or if its ID matches the provided transaction ID
elif tran_id == transaction['id']:
    print(f"\nTransaction with the provided Transaction ID: {tran_id} has been found.")
    print(transaction)
```

Figure 30: Temporary Modifications for Delete Transaction

1.8.2 Executing Test Cases

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying the input of a text as the transaction ID	User will input randomly chosen text (delete) as the transaction ID	System is expected to Display an error message and prompt user to reenter the transaction ID	An error message was displayed and user was asked to re-enter the transaction ID	Pass
02	Checking the scenario when invalid transaction ID is entered	User inputting a randomly chosen number (10) as ID which is not shown in the transaction details	The system should display an error message and redirect the user to the main menu	System displayed an error message was displayed and user was redirected to the main menu	Pass
03	Verifying an Empty Input for the ID	User pressing 'Enter' key for an input	System is expected to display an error message and prompts user for Transaction ID	System displayed an error message and user was prompted to enter the Transaction ID	Pass

04	Verifying that the transaction ID captures the correct transaction	User inputting an already existing transaction ID	System should print the details of the transaction relevant to transaction ID	System displayed a message indicating the Transaction details relevant to transaction ID have been found and displayed the details of the transaction	Pass
05	Removing transactions	User will input valid transaction ID and user confirms the transaction deletion	System should delete the transaction details relevant to the user entered Transaction ID	System deletes the Transaction details relevant to Transaction ID and displays a message to the user indicating the successful deletion	Pass
06	Checking the scenario when there are no existing transactions	Remove the data in the JSON file User will input '6' as the choice in main menu	System should Display a message to the user after it check for existing transactions	An error message was displayed and user was redirected to main menu	Pass

Table 7: Test Cases for Delete Transaction

1.8.3 Test Results Snapshots

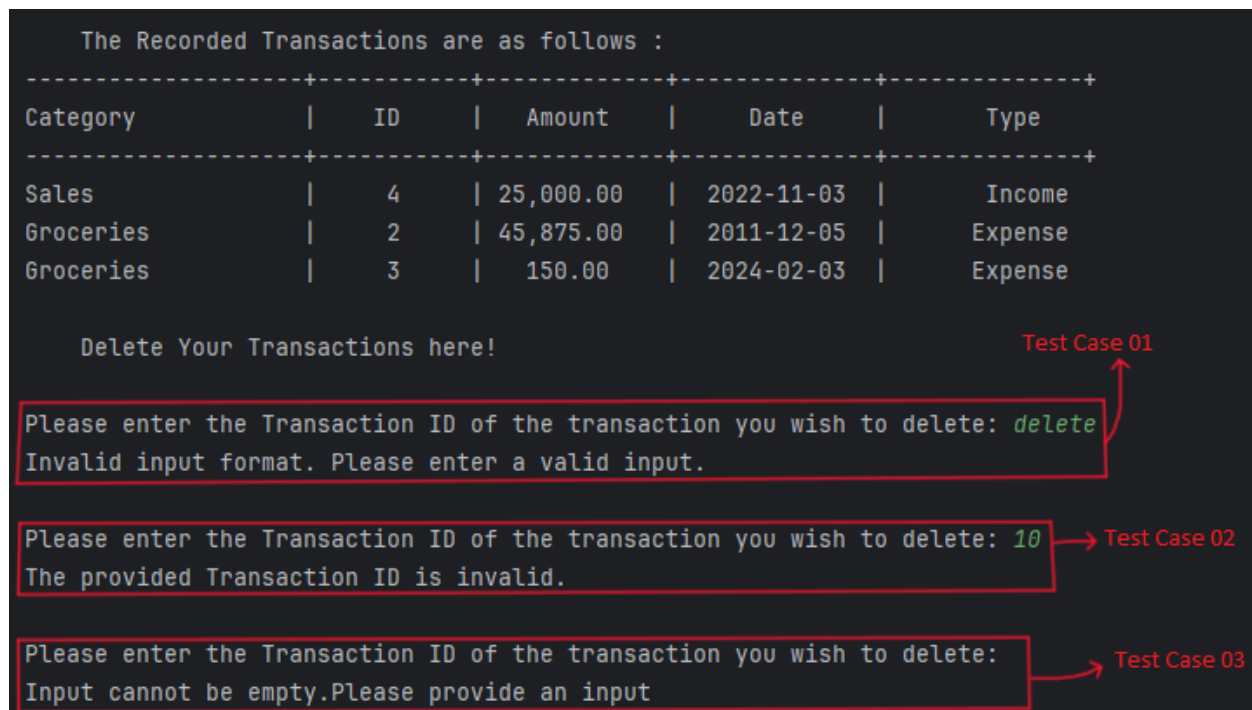


Figure 31: Invalid Inputs for the Transaction ID - Delete Transaction Test Cases

In the Figure 31,

- For a randomly chosen string input (delete), the system generated an error: *'Invalid input. Please enter a valid input'*. The user was then prompted to re-enter a choice.
- For a randomly chosen numeral input (10) as the ID, the system displayed an error: *'The provided Transaction ID is invalid'*. The user was then prompted to re-enter a choice.
- For an empty input, System generated an error: *'Input cannot be empty. Please provide an input.'* The user was then prompted to re-enter a choice.

```
The Recorded Transactions are as follows :
-----+-----+-----+-----+-----+
Category      | ID    | Amount  | Date      | Type
-----+-----+-----+-----+-----+
Sales         | 1     | 122,884.00 | 2024-12-05 | Income
Groceries     | 2     | 7,850.00  | 2021-12-03 | Expense

Delete Your Transactions here!

Please enter the Transaction ID of the transaction you wish to delete: 2
Test Case 04

Transaction with the provided Transaction ID: 2 has been found.
{'amount': 7850.0, 'date': '2021-12-03', 'type': 'Expense', 'id': 2}

Do you want to delete the transaction details ? (Yes/No):|
```

Figure 32: Transaction ID capturing the correct Transaction – Delete Transaction Test Cases

In the Figure 32,

- System displayed a message ‘Transaction with the provided Transaction ID: 2 has been found’ after capturing the transaction which is relevant to user entered Transaction ID.

```
The Recorded Transactions are as follows :
-----+-----+-----+-----+-----+
Category      | ID    | Amount  | Date      | Type
-----+-----+-----+-----+-----+
Sales         | 4     | 25,000.00 | 2022-11-03 | Income
Groceries     | 3     | 150.00   | 2024-02-03 | Expense
Entertainment | 5     | 6,000.00 | 2021-12-05 | Expense

Delete Your Transactions here!

Please enter the Transaction ID of the transaction you wish to delete: 3
Test Case 04

Transaction with the provided Transaction ID: 3 has been found.

Do you want to delete the transaction details ? (Yes/No):yes
Transaction details for Transaction ID 3 have been successfully deleted.
```

Figure 33: Successful Transaction Deletion - Delete Transaction Test Cases

In the Figure 33,

- After user's confirmation on transaction deletion, a message was displayed by the system to inform the user about successful transaction deletion.

1.9 Test cases for Display Summary

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Checking the scenario when there are no existing transactions	Remove the data in the JSON file. User will input '5' as the choice in main menu.	System should Display a message to the user after it check for existing transactions	An error message was displayed and user was redirected to main menu.	Pass
02	Verifying that the system catches all the relevant transactions for each transaction category	User Calling the Display Summary Function by Inputting '5' as the choice	The system is expected to display all the transaction total of each Transaction Type	System displayed summary of transaction total for each of the transaction type	Pass
03	Verifying that the system catches all the transactions relevant to 'income' category	User will call the function responsible for filtering transactions based on their category,	System should display all the transactions relevant to category 'income'	System displayed all the transactions relevant to category 'income'	Pass

		specifically targeting transactions categorized as 'income'			
04	Verifying that the system catches all the transactions relevant to 'expense' category	User will call the function responsible for filtering transactions based on their category, specifically targeting transactions categorized as 'expense'	System should display all the transactions relevant to category 'expense'	System displayed all the transactions relevant to Category 'expense'	Pass
05	Verifying the correct summary of existing transactions	User will input '5' as the choice in main menu	System should display a summary of transactions including total income, total expense and the net balance	System displayed the overall summary of transactions including total income, total expense, and net balance	Pass

Table 8: Test Cases for Display Summary

1.9.1 Test Results Snapshots

Transaction Summary		
Category	Amount	Category Total
Sales		
	75,800.00	75,800.00
	2,500,000.00	2,575,800.00
Category Total (Income):		
		2,575,800.00
Entertainment		
	6,000.00	6,000.00
	10,000.00	16,000.00
Category Total (Expense):		
		16,000.00
Groceries		
	9,800.00	9,800.00
	14,500.00	24,300.00
Category Total (Expense):		
		24,300.00
Total Income:	2,575,800.00	
Total Expenses:	40,300.00	
Net Balance:	2,535,500.00	

Test Case 02

Test Case 03

Test Case 04

Test Case 05

Figure 34: Summary of overall Transactions - Display Transaction Summary

In the Figure 34,

- The system displayed Sum of transactions in each of the Transaction category type.
- The system displayed the total summary of Transactions of Total Income, Total Expenses and the Net Balance.

1.10 Read Bulk Transaction Test Cases

1.10.1 Temporary Modifications Done for Test-phase

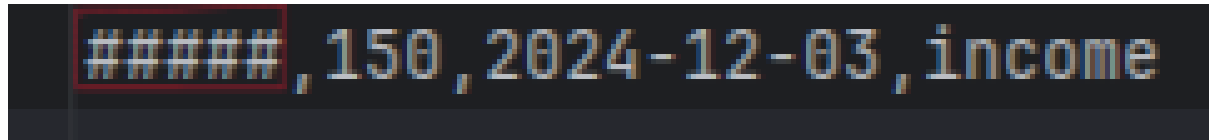
- Default Transaction category was changed into a randomly-chosen sequence of numbers '9999999' to test the expected Outcome in Test Case 02.



A screenshot of a transaction record displayed on a dark background. The text is '99999999,150,2024-12-03,income'. The first eight digits '99999999' are enclosed in a red rectangular box.

Figure 35: Changing Transaction Category to numeral character - Test Case 02 -Reading Bulk Transactions

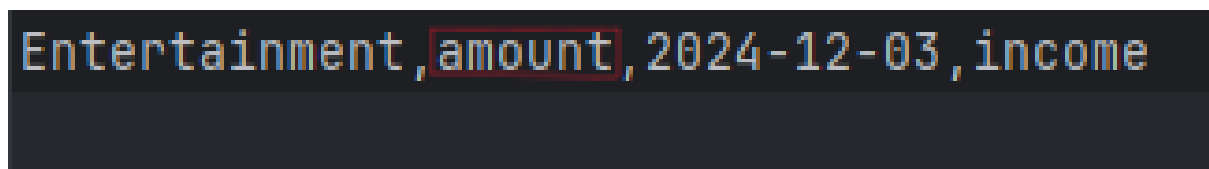
- Default Transaction category was changed into a randomly-chosen sequence of special characters '#####' to test the Expected Outcome in Test Case 03.



A screenshot of a transaction record displayed on a dark background. The text is '#####,150,2024-12-03,income'. The seven hash characters '#####' are enclosed in a red rectangular box.

Figure 36: Changing Transaction Category to special character - Test Case 03 -Reading Bulk Transactions

- Default Transaction Amount was changed into a randomly-chosen textual characters 'amount' to test the expected Outcome in Test Case 04.



A screenshot of a transaction record displayed on a dark background. The text is 'Entertainment,amount,2024-12-03,income'. The word 'amount' is enclosed in a red rectangular box.

Figure 37: Changing Transaction Amount to textual character - Test Case 04 -Reading Bulk Transactions

- Default Transaction Amount was changed into a randomly-chosen special characters '@#\$%' to test the expected Outcome in Test Case 05.



A screenshot of a transaction record displayed on a dark background. The text is 'Entertainment,@#\$%,2024-12-03,income'. The special characters '@#\$%' are enclosed in a red rectangular box.

Figure 38: Changing Transaction Amount to Special character - Test Case 05 -Reading Bulk Transactions

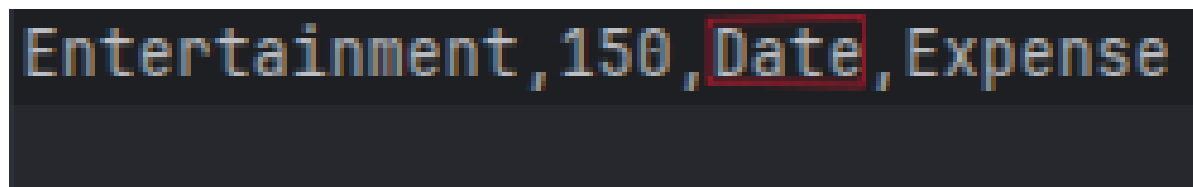
- Default Transaction Date was changed into an invalid date '2024-15-03' to test the expected Outcome in Test Case 06.



Entertainment, 150, 2024-15-03, income

Figure 39: Changing Transaction Date to Invalid date - Test Case 06 -Reading Bulk Transactions

- Default Transaction Date was changed into a text ‘Date’ to test the expected Outcome in Test Case 07.



Entertainment, 150, Date, Expense

Figure 40: Changing Transaction Date to Text - Test Case 07 -Reading Bulk Transactions

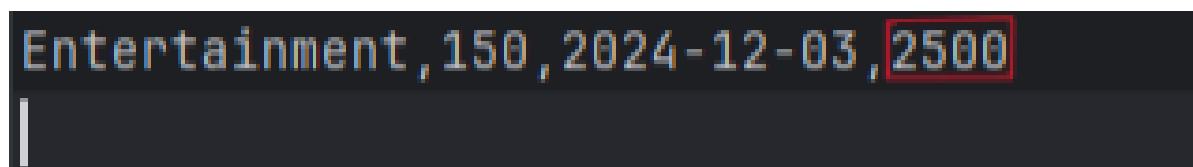
- Default Transaction Date was changed into a sequence of numbers ‘20241203’ to test the expected Outcome in Test Case 08.



Entertainment, 150, 20241203, Expense

Figure 41: Changing Transaction Date to a Number - Test Case 08 -Reading Bulk Transactions

- Default Transaction Type was changed into a number ‘2500’ to test the expected Outcome in Test Case 09.



Entertainment, 150, 2024-12-03, 2500

Figure 42: Changing Transaction Type to Numeral Character - Test Case 09 -Reading Bulk Transactions

- Default Transaction Type was changed into a randomly chosen special characters ‘%^&*>’ to test the expected Outcome in Test Case 10.

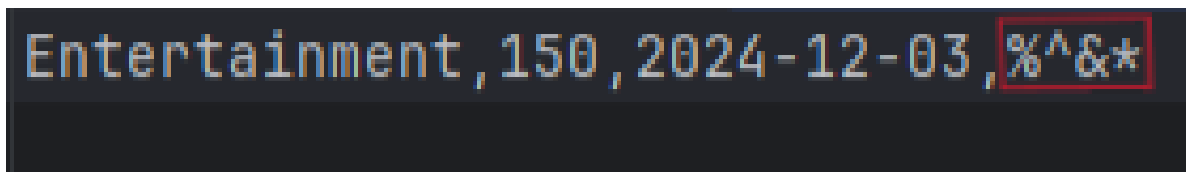


Figure 43: Changing Transaction Type to Special Character - Test Case 09 -Reading Bulk Transactions

- A Significant number of transactions were recorded in the Text File to evaluate the expected Outcome in Test Case 11.

```
Entertainment,20000,2024-12-01,Expense|
Entertainment,7500,2024-12-02,Expense
Entertainment,1500,2024-12-03,Expense
Sales,20000,2024-12-04,Income
Groceries,8000,2024-12-05,Expense
Groceries,750,2024-12-06,Expense
Sales,55000,2024-12-07,Income
Rent Income,500,2024-12-08,Income
Rent Income,1500,2024-12-09,Income
Travel,200,2024-12-10,Expense
Travel,400,2024-12-11,Expense
Travel,300,2024-12-15,Expense
```

Figure 44: Recording Bulk Transactions - Test Case 11 - Reading Bulk Transactions

1.10.2 Executing Test Cases

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Checking the scenario where One of the detail is missing	Remove one of the transaction detail from a	An error message should be displayed	System displayed an error message	Pass

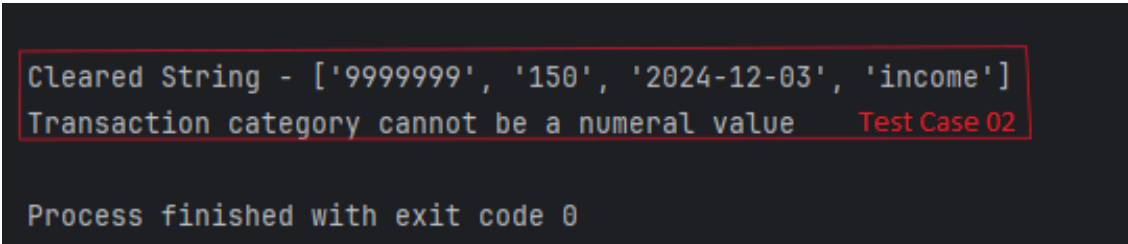
		one transaction in the Text file	by the system indicating the error encountered		
		Calling the read bulk transaction function	while reading from a text file.		
02	Checking the scenario where Transaction category contains numbers	Replacing the transaction category with a sequence of numbers (9999999)	An error message should be displayed by the system indicating the error encountered while reading from a text file.	System displayed an error message	Pass
		Calling the read bulk transaction function			
03	Checking the scenario where Transaction category contains special characters	Replacing the transaction category with a sequence of special characters (#####)	The system is expected to display an error message.	The system displayed an error message indicating the error encountered.	Pass

		Calling the read bulk transaction function			
04	Checking the scenario where Transaction amount contains textual characters	Calling the read bulk transaction function	The system is expected to display an error message.	The system displayed an error message indicating the error encountered.	Pass
05	Checking the scenario where Transaction amount contains special characters	Calling the read bulk transaction function	The system is expected to display an error message.	The system displayed an error message indicating the error encountered.	Pass
06	Checking the scenario where Transaction date contains special invalid date	Calling the read bulk transaction function	The system is expected to display an error message.	The system displayed an error message indicating the error encountered.	Pass
07	Checking the scenario where Transaction date contains textual characters	Calling the read bulk transaction function	The system is expected to display an error message.	The system displayed an error message indicating the error encountered.	Pass
08	Checking the scenario where Transaction date	Calling the read bulk	The system is expected to display	The system displayed an error message indicating	Pass

	contains numeral characters	transaction function.	an error message.	the error encountered.	
09	Checking the scenario where Transaction type contains numeral characters	Calling the read bulk transaction function.	The system is expected to display an error message.	The system displayed an error message indicating the error encountered.	Pass
10	Checking the scenario where Transaction type contains special characters	Calling the read bulk transaction function.	The system is expected to display an error message.	The system displayed an error message indicating the error encountered.	Pass
11	Verifying the transactions addition to the JSON file successfully	Calling the read bulk transaction function.	A message should be displayed by the system indicating the successful transaction save	The system displayed a message	Pass

Table 9: Test Cases for Reading Bulk Transactions

1.10.3 Testing Results Snapshots



```

Cleared String - ['9999999', '150', '2024-12-03', 'income']
Transaction category cannot be a numeral value Test Case 02

Process finished with exit code 0

```

Figure 45: Input Validation – Transaction Category - Test Case 02 - Read Bulk Transactions

```
Cleared String - ['####', '150', '2024-12-03', 'income']
Category contains special characters      Test Case 03

Process finished with exit code 0
```

Figure 46: Input Validation – Transaction Category - Test Case 03 - Read Bulk Transactions

```
Cleared String - ['Entertainment', 'amount', '2024-12-03', 'income']
Amount must be a numeric value          Test Case 04

Process finished with exit code 0
```

Figure 47: Input Validation – Transaction Amount - Test Case 04 - Read Bulk Transactions

```
Cleared String - ['Entertainment', '@#$', '2024-12-03', 'income']
Amount must be a numeric value          Test Case 05

Process finished with exit code 0
```

Figure 48: Input Validation – Transaction Amount - Test Case 05 - Read Bulk Transactions

```
Cleared String - ['Entertainment', '150', '2024-15-03', 'income']
Invalid Date format                     Test Case 06

Process finished with exit code 0
```

Figure 49: Input Validation – Transaction Date - Test Case 06 - Read Bulk Transactions

```
Cleared String - ['Entertainment', '150', 'Date', 'income']
Invalid Date format                     Test Case 07

Process finished with exit code 0
```

Figure 50: Input Validation – Transaction Date - Test Case 07 - Read Bulk Transactions


```
Cleared String - ['Entertainment', '150', '20241203', 'income']
Invalid Date format
Test Case 08

Process finished with exit code 0
```

Figure 51: Input Validation – Transaction Date - Test Case 08 - Read Bulk Transactions

```
Cleared String - ['Entertainment', '150', '2024-12-03', '2500']
Invalid income/expense type
Test Case 09

Process finished with exit code 0
```

Figure 52: Input Validation – Transaction Type - Test Case 09 - Read Bulk Transactions

```
Cleared String - ['Entertainment', '150', '2024-12-03', '%^&*']
Invalid income/expense type
Test Case 10

Process finished with exit code 0
```

Figure 53: Input Validation – Transaction Type - Test Case 10 - Read Bulk Transactions

```
Transactions Added to the JSON file from the text file
Test Case 11

Process finished with exit code 0
```

Figure 54: Successful Implementation of Reading Bulk Transaction - Test Case 11 - Read Bulk Transactions:

1.11 Test Cases for Searching in the GUI

1.11.1 Test Cases for Search by Date

- Select ‘Search By Date’ from ‘Search’ Menu

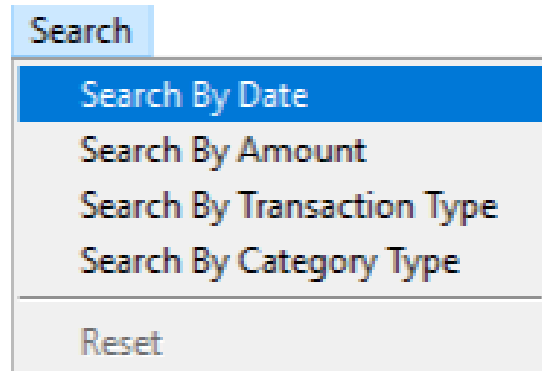


Figure 55: 'Select By Date' option in 'Search' Menu

- Click on 'Search Records' button after providing an input for the 'Date' .

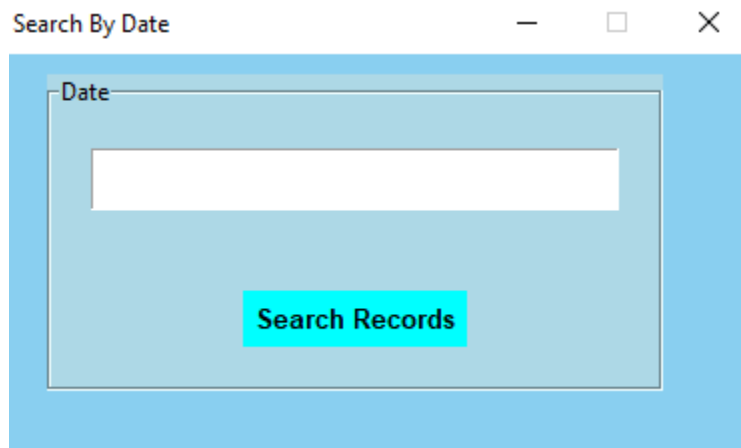


Figure 56: 'Search By Date' Window

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying for an empty input for searching	Clicking on the 'Search Records' button without providing an input for the Date	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass

02	Verifying for an invalid date for searching	Providing invalid date (2024-15-03) as the input	System is expected to show an error message and prompts user for an input again	An Error message was displayed and user was prompted again to input a date	Pass
03	Verifying for a textual value as the date for searching	Providing a text (Date) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
04	Verifying the input of a special character sequence as the date for searching	Providing invalid date (@#%) as the input	System is expected to show an error message and prompts user for an input again	An Error message was displayed and user was prompted again to input a date	Pass
05	Verifying for a numeral value as the date	Providing a number (2850) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message to the user and prompted again for an input	Pass
06	Verifying the input of a negative	Providing a negative (-	System is expected to show an error	System displayed an error message	Pass

	number as the date for searching	15820) as the input	message and prompts user for an input again	and user was prompted for an input	
07	Verifying for a date which is not in the records as the date for searching	Providing a date which is not in the records (2080-12-05) as the input	System is expected to show a message indicating the user that the Entered Date was not found	A message was displayed to the user	Pass
08	Verifying the input of a valid date as the date for searching	Providing a valid date (2024-12-02) as the input	System is expected to show the records relevant to the entered date	System displayed all the records relevant to the date on the GUI	Pass
09	Verifying the reset option after search process.	Selecting 'Reset' option in the 'Search' Menu	System is expected to remove the data of the search process and display all the records in the JSON file	System removed the data of the search process and displayed all records in the JSON file	Pass

Table 10: Test Cases for Search by Date

1.11.1.1 Testing Results Snapshots

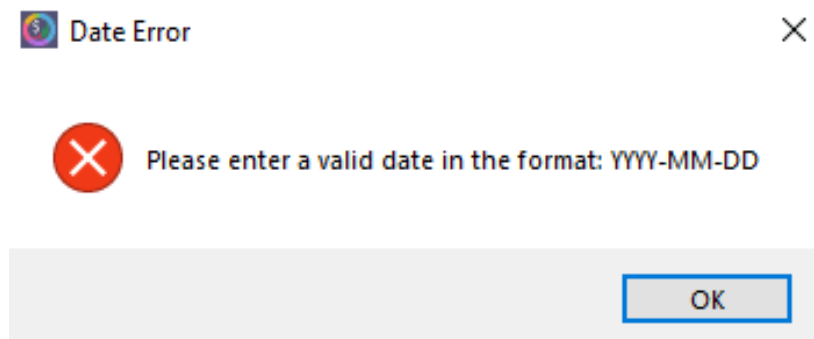


Figure 57: Test Result for Test Cases 01, 02, 03, 04, 05, and 06 – Search By Date

- In Figure 57, System displayed the above error message for invalid inputs for the Date

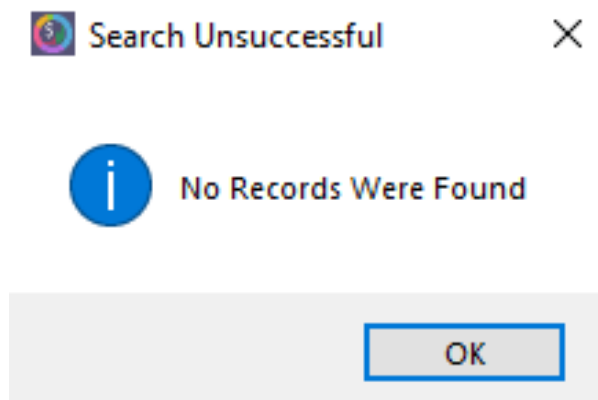


Figure 58: Test Result for Test Case 07 - Search By Date

- In Figure 58, System displayed the above message after searching the provided invalid transaction date.

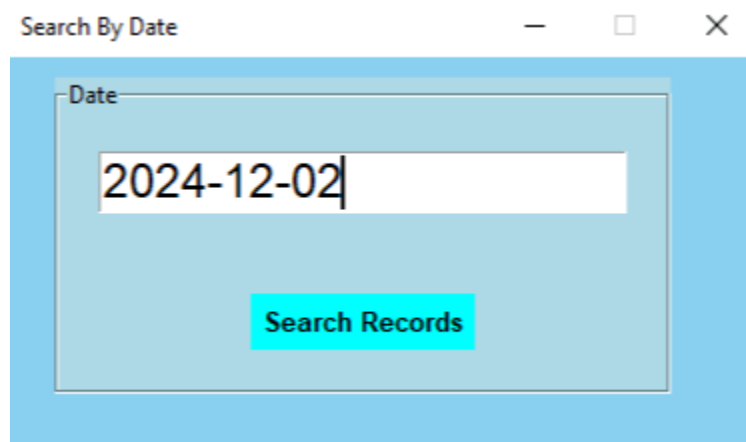


Figure 59: Test Action for Test Case 08 - Search By Date

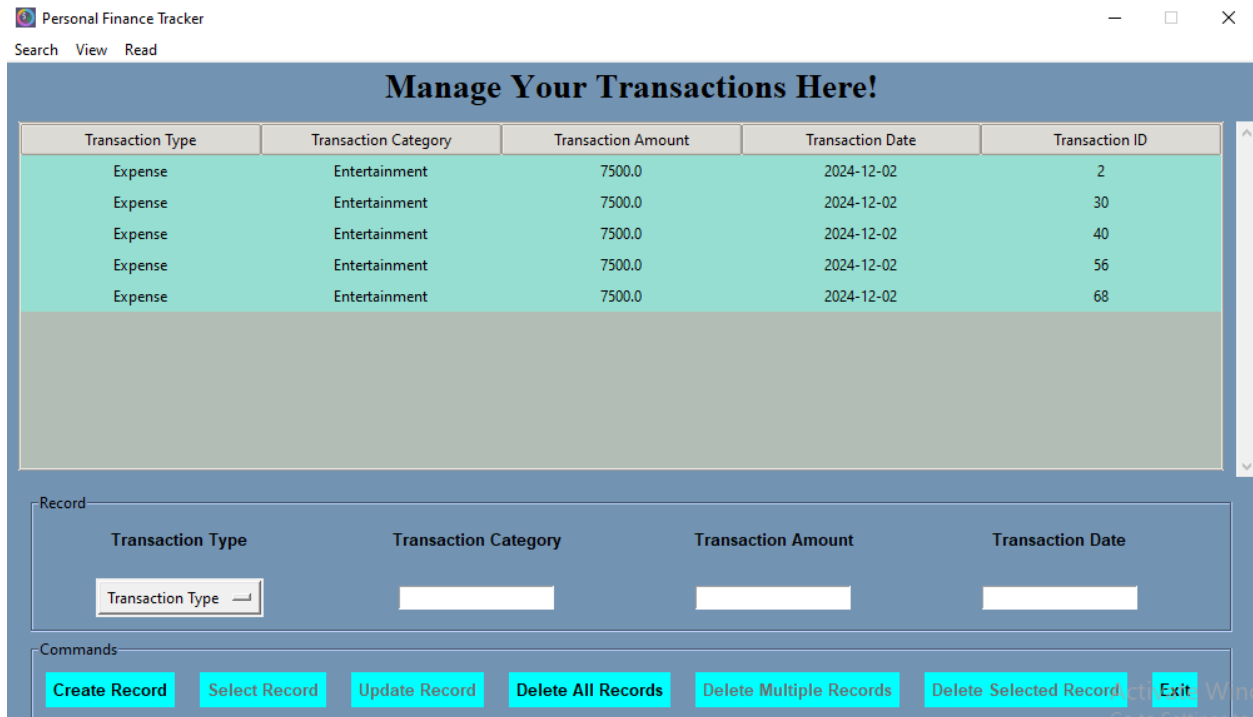


Figure 60: Test Result for Test Case 08 - Search By Date

In Figures 59, and 60,

- System searched through all records and displayed the records relevant to provided validated date

1.11.2 Test Cases for Search by Amount

- Select 'Search By Amount' from 'Search' Menu

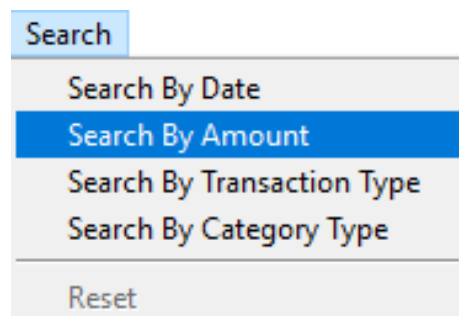


Figure 61: 'Select By Amount' option in 'Search' Menu

- Click on 'Search Records' button after providing an input for the 'Amount'

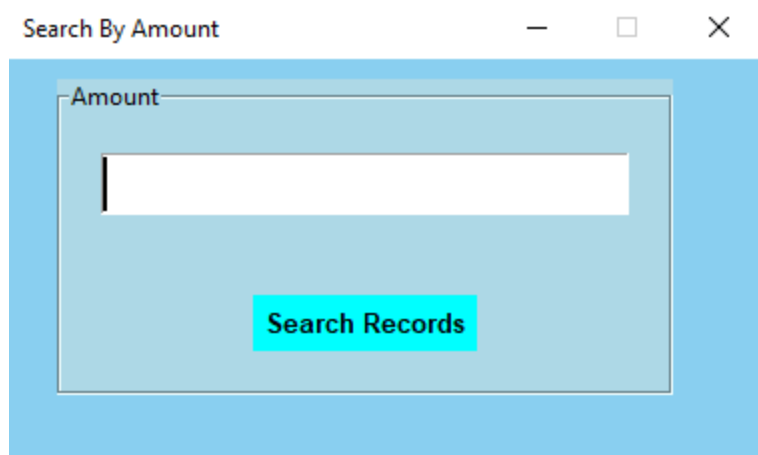


Figure 62: 'Search By Amount' Window

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying for an empty input for searching	Clicking on the 'Search Records' button without providing an input for the 'Amount'	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
02	Verifying for a textual value as the 'Amount' for searching	Providing a text (Amount) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
03	Verifying the input of a special character	Providing invalid date (!@#\$) as the input	System is expected to show an error message and	An Error message was displayed and user was	Pass

	sequence as the 'Amount' for searching		prompts user for an input again	prompted again to input a 'Amount'	
04	Verifying the input of a negative number as the 'Amount' for searching	Providing a negative (-4520) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
05	Verifying the input of a valid amount as the 'Amount' for searching	Providing a valid date (7500) as the input	System is expected to show the records relevant to the entered Amount	System displayed all the records relevant to the date on the GUI	Pass

Table 11: Test Cases for Search by Amount

1.11.2.1 Testing Results Snapshots

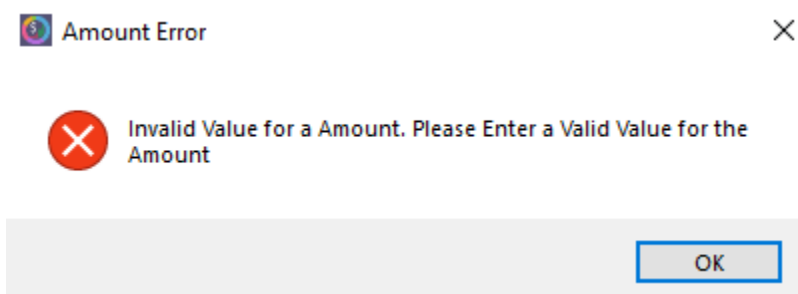
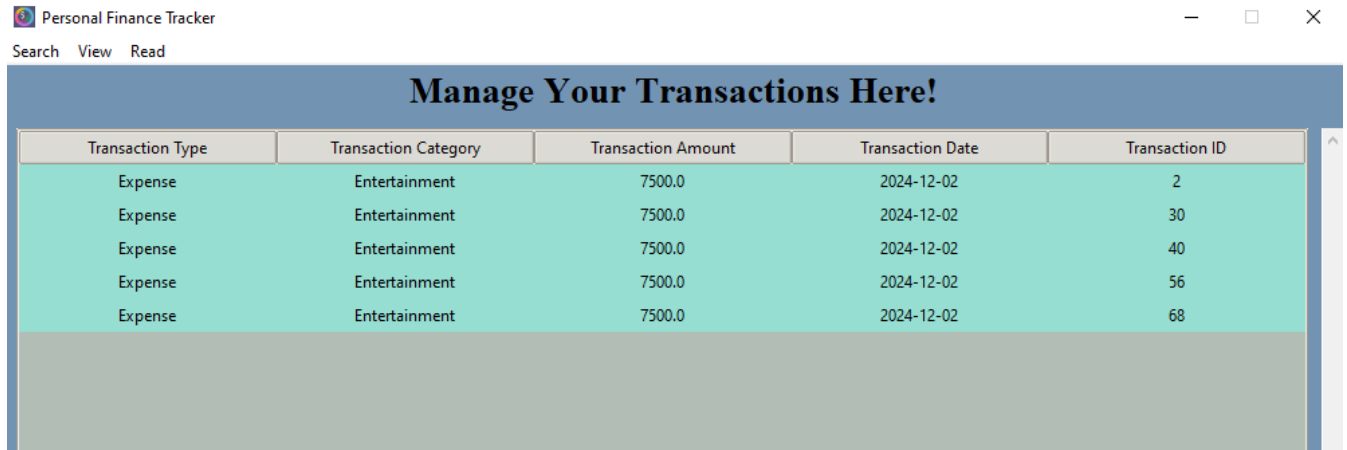


Figure 63: Test Results for Test Cases 01, 02, 03, and 04 – Search By Amount

- In Figure 63, system displayed the above message after searching the provided invalid transaction amount



Transaction Type	Transaction Category	Transaction Amount	Transaction Date	Transaction ID
Expense	Entertainment	7500.0	2024-12-02	2
Expense	Entertainment	7500.0	2024-12-02	30
Expense	Entertainment	7500.0	2024-12-02	40
Expense	Entertainment	7500.0	2024-12-02	56
Expense	Entertainment	7500.0	2024-12-02	68

Figure 64: Test Result for Test Case 05 - Search By Amount

- In Figure 64, System searched through all records and displayed the records relevant to provided validated amount

1.11.3 Test Cases for Search By Transaction Type

- Select 'Search By Transaction Type' from 'Search' Menu

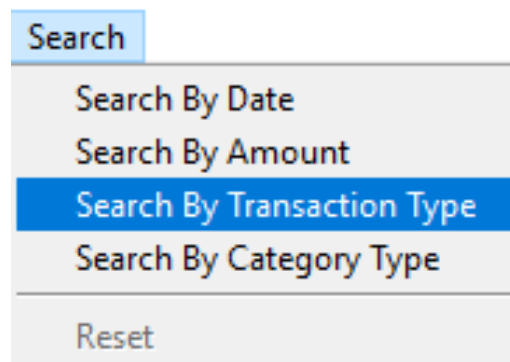


Figure 65: 'Search By Transaction Type' option in 'Search' Menu

- Click on 'Search Records' button after providing an input for the 'Transaction Type'

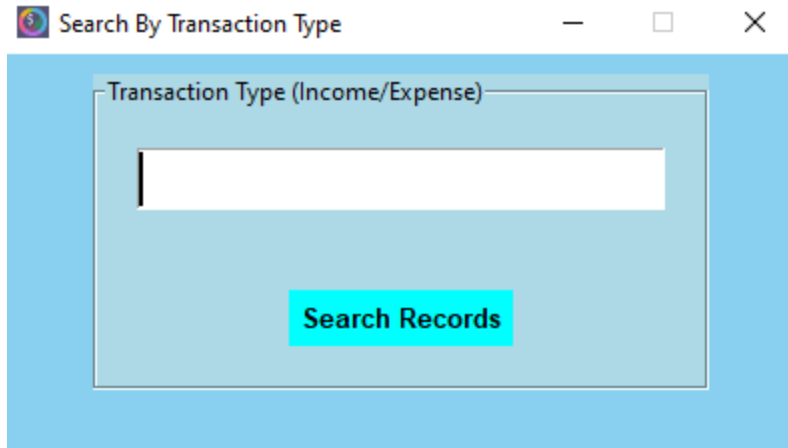


Figure 66: 'Search By Transaction Type' Window

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying for an empty input for searching	Clicking on the 'Search Records' button without providing an input for the 'Transaction Type'	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
02	Verifying for a textual value which is not 'Income' or 'Expense' as the 'Transaction Type' for searching	Providing a text (Liability) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass

03	Verifying the input of a special character sequence as the 'Transaction Type' for searching	Providing invalid date (!@#\$) as the input	System is expected to show an error message and prompts user for an input again	An Error message was displayed and user was prompted again to input a 'Transaction Type'	Pass
04	Verifying the input of a number as the 'Transaction Type' for searching	Providing a number (2500) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
05	Verifying the input of a negative number as the 'Transaction Type' for searching	Providing a negative number (-4520) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
06	Verifying the input of a valid amount as the 'Transaction Type' for searching	Providing a valid Transaction Type (Expense) as the input	System is expected to show the records relevant to the entered Transaction Type	System displayed all the records relevant to the Transaction Type on the GUI	Pass

Table 12: Test Cases for Search by Transaction Type

1.11.3.1 Testing Results Snapshots

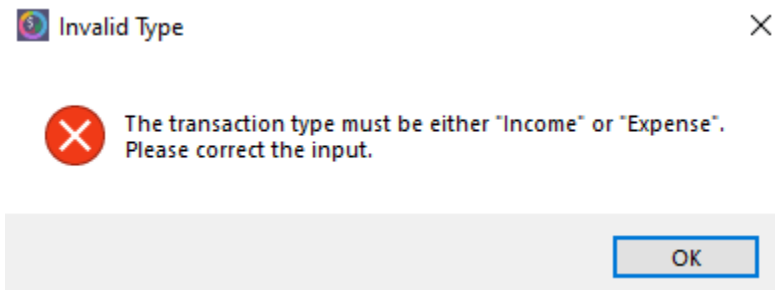


Figure 67: Test Result for Test Cases 01,02,03,04 and 05 – Search By Transaction Type

- In Figure 67, system displayed the above message after searching the provided invalid transaction type.

Personal Finance Tracker

Search View Read

Manage Your Transactions Here!

Transaction Type	Transaction Category	Transaction Amount	Transaction Date	Transaction ID
Expense	Entertainment	6000.0	2024-12-10	1
Expense	Entertainment	7500.0	2024-12-02	2
Expense	Entertainment	1500.0	2024-12-03	3
Expense	Entertainment	50000.0	2024-11-03	13
Expense	Entertainment	20000.0	2024-12-01	29
Expense	Entertainment	7500.0	2024-12-02	30
Expense	Entertainment	1000.0	2024-04-26	31
Expense	Entertainment	20000.0	2024-12-01	39
Expense	Entertainment	7500.0	2024-12-02	40
Expense	Entertainment	1500.0	2024-12-03	41

Figure 68: Test Result for Test Case 06 – Search By Transaction Type

- In Figure 68, System searched through all records and displayed the records relevant to provided transaction type.

1.11.4 Test Cases for Search By Category Type

- Select 'Search By Transaction Type' from 'Search' Menu

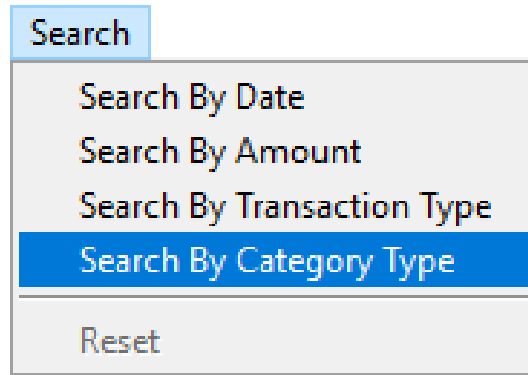


Figure 69: 'Search By Category Type' option in 'Search' Menu

- Click on 'Search Records' button after providing an input for the 'Category Type'

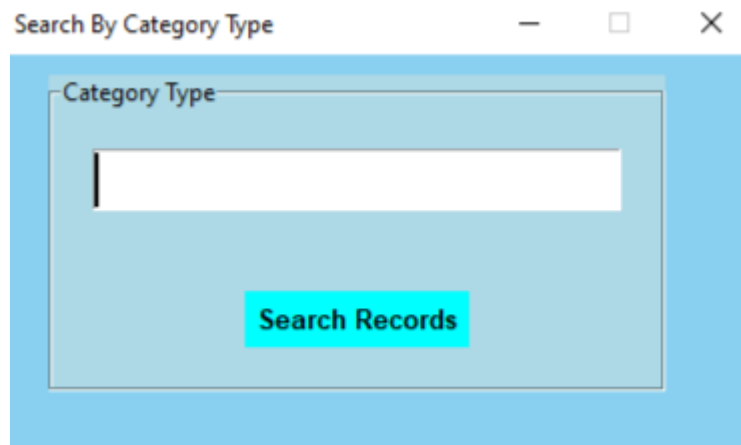


Figure 70: 'Search By Category Type' Window

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status
01	Verifying for an empty input for searching	Clicking on the 'Search Records' button without providing an input for the 'Category Type'	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass

02	Verifying for a textual value as the 'Category Type' for searching	Providing a text (Category) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
03	Verifying the input of a special character sequence as the 'Category Type' for searching	Providing special character sequence (!@#)\$ as the input	System is expected to show an error message and prompts user for an input again	An Error message was displayed and user was prompted again to input a 'Category Type'	Pass
04	Verifying the input of a negative number as the 'Category Type' for searching	Providing a negative number (-4520) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass
05	Verifying the input of a decimal number as the 'Category Type' for searching	Providing a decimal number (25.13) as the input	System is expected to show an error message and prompts user for an input again	System displayed an error message and user was prompted for an input	Pass

06	Verifying the input of a valid value as the 'Category Type' for searching	Providing a valid value (Groceries) as the input	System is expected to show the records relevant to the entered Category Type	System displayed all the records relevant to the Category Type on the GUI	Pass
----	---	--	--	---	------

Table 13: Test Cases for Search by Category Type

1.11.4.1 Testing Results Snapshots

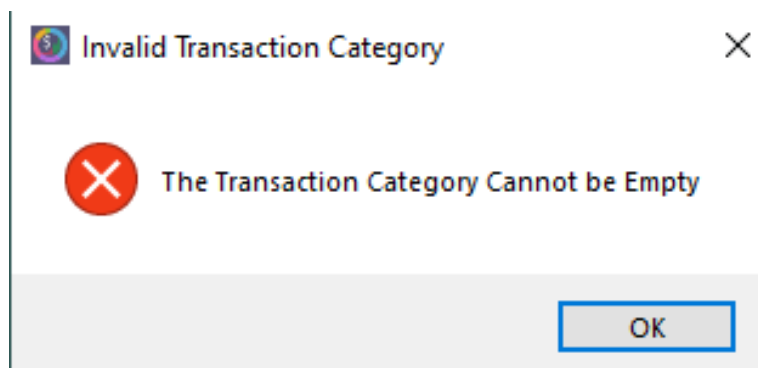


Figure 71: Test Result - Test Case 01- Search By Category Type

- In Figure 71, system generated above error for an empty input

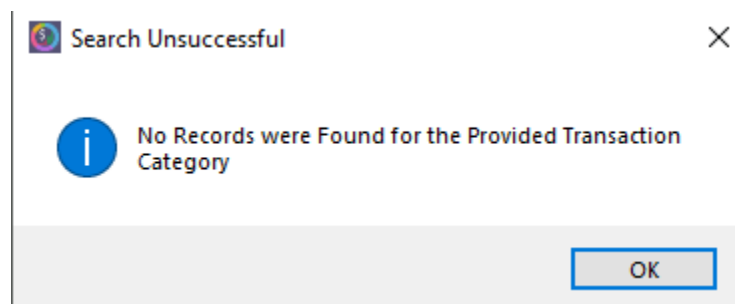


Figure 72: Test Result - Test Case 02 - Search By Category Type

- In Figure 72, system generated above error when there are no records under the provided transaction category name

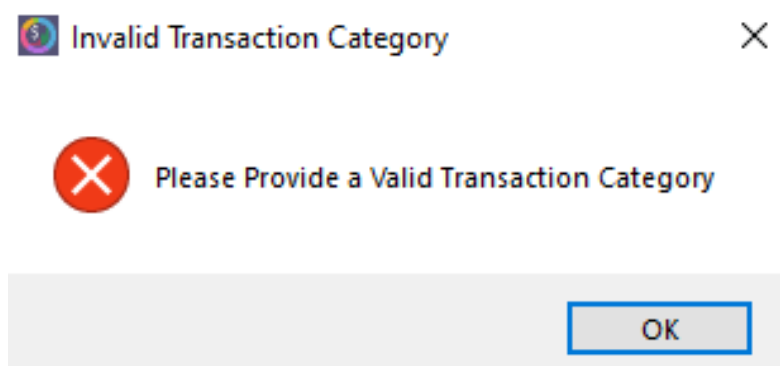


Figure 73: Test Result for Test Cases 03, 04, 05- Search By Category Type

- In Figure 73, system generated above error for an invalid input

Manage Your Transactions Here!				
Transaction Type	Transaction Category	Transaction Amount	Transaction Date	Transaction ID
Expense	Groceries	8000.0	2024-12-05	5
Expense	Groceries	750.0	2024-12-06	6
Expense	Groceries	8000.0	2024-12-05	17
Expense	Groceries	750.0	2024-12-06	18
Income	Groceries	5000.0	2024-12-05	26
Expense	Groceries	25000.0	2024-12-05	28
Expense	Groceries	8000.0	2024-12-05	33
Expense	Groceries	750.0	2024-12-06	34
Expense	Groceries	8000.0	2024-12-05	43
Expense	Groceries	750.0	2024-12-06	44

Figure 74: Test Result for Test Case 06 - Search By Category Type

- System searched through all records and displayed the records relevant to provided transaction category

1.12 Test Cases for Sorting in GUI

Test Case Number	Description of the test case	Action	Expected Outcome/s	Actual Outcome/s	Pass/Fail Status

01	Verifying for sorting in all columns in ascending order	Clicking on each columns.	System is expected to show all records in an ascending order	System showed all the records in an ascending order	Pass
02	Verifying for sorting in all columns in descending order	Clicking on each columns.	System is expected to show all records in an descending order	System showed all the records in an descending order	Pass

Table 14: Test Cases for Sorting

1.12.1 Testing Results Snapshots

Transaction ID
1
2
3
4
5
6
7
8
9
10

Transaction Amount
200.00
200.00
200.00
200.00
200.00
300.00
300.00
300.00
300.00
300.00

Figure 75: ID and amount sorted in ascending order - Test Result - Test Case 01- Sorting on GUI

Transaction Date
2024-04-26
2024-04-26
2024-11-03
2024-12-01
2024-12-01
2024-12-01
2024-12-01
2024-12-02
2024-12-02
2024-12-02

Figure 76: Date sorted in ascending order - Test Result - Test Case 01- Sorting on GUI

Transaction Category	Transaction Type
Entertainment	Expense
Entertainment	Expense
Entertainment	Expense
Entertainment	Expense
Groceries	Expense
Groceries	Expense
Groceries	Expense
Groceries	Income
Groceries	Income
Groceries	Income

Figure 77: Transaction Category and Type Sorted in Ascending Order – Test Result – Test Case 01- Sorting on GUI

Transaction ID	Transaction Amount
78	50,000.00
77	25,000.00
76	25,000.00
75	20,000.00
74	20,000.00
73	20,000.00
72	20,000.00
71	20,000.00
70	20,000.00
69	20,000.00

Figure 78: Transaction ID and Amount Sorted in Descending Order - Test Result - Test Case 02- Sorting on GUI

Transaction Category	Transaction Type
Travel	Income
Travel	Income
Travel	Income
Travel	Income
Sales	Income
Sales	Income
Sales	Expense
Sales	Expense
Sales	Expense
Sales	Expense

Figure 79: Transaction Category and Type Sorted in Descending Order - Test Result - Test Case 02- Sorting on GUI

Transaction Date
2024-12-11
2024-12-11
2024-12-11
2024-12-10
2024-12-10
2024-12-10
2024-12-10
2024-12-10
2024-12-10
2024-12-09

Figure 80: Transaction Date Sorted in Descending Order - Test Result - Test Case 02- Sorting on GUI

1.13 TEST SUMMARY

This document includes the various testing activities conducted as a part of Personal Finance Tracker Application. Multiple Testing activities were performed in order to ensure the smooth operation of the personal finance tracker

The personal finance tracker is a python-based application which was developed focusing on basic programming principles including GUI Implementation with Tkinter and OOP, input validation and file operations.

Through this testing phase, the bugs and errors which could not be identified before was able to identify and necessary adjustments were made to minimize those errors and this testing phase was a critical step in developing a smooth and executable program

Total No of Test Cases Planned	Total No of Test Cases Executed	Total No of Passed Test Cases	Total No of Failed Test Cases
90	89	89	0

Table 15: Test Summary Results