A white object with a black background

Description automatically generated

Course :Bsc(Hons) Computer Science

Module : Software Development 01

Module Leader : Mr. Guhanathan P

Assignment Type : Individual

Issue Date : 07th April 2024

Hand – in – Date : 29 th April 2024

Deadline : on or before 1.00 PM

Student Id : 20230469

Email : Disandu.20230469@iit.ac.lk

Contents

[**Programme Design** 3](#_Toc165265969)

[**Algorithm** 3](#_Toc165265970)

[**Data Structure Design** 4](#_Toc165265971)

[**Pesudocode Design** 5](#_Toc165265972)

[Function create\_widgets 5](#_Toc165265973)

[Function Display transactions 6](#_Toc165265974)

[Function Search Transactions 6](#_Toc165265975)

[Function Sort By Column 6](#_Toc165265976)

[Function load\_transactions() 6](#_Toc165265977)

[Function save\_transactions() 6](#_Toc165265978)

[Function add\_transactions() 7](#_Toc165265979)

[Function view\_transactions 8](#_Toc165265980)

[Function Update\_transaction 9](#_Toc165265981)

[Function Delete\_transactions 10](#_Toc165265982)

[Function display\_summary 10](#_Toc165265983)

[Function Main() 11](#_Toc165265984)

[**Test cases** 13](#_Toc165265985)

[1. Main Function 13](#_Toc165265986)

[1.2 Screenshot For Test Cases 13](#_Toc165265987)

[2. Add Transaction Function 14](#_Toc165265988)

[2.2 Screenshots For Test Cases 14](#_Toc165265989)

[3. View Transactions Function 16](#_Toc165265990)

[3.1 Screen Shots For Test cases 16](#_Toc165265991)

[4. Update Transactions Function 17](#_Toc165265992)

[5.2 Screen shots for Test Cases 17](#_Toc165265993)

[6. Delete Transactions Function 19](#_Toc165265994)

[6.2 Screen shots for Test Cases 20](#_Toc165265995)

[7. Read Bulk Transactions Function 21](#_Toc165265996)

[7.2 Screen shots for Test Cases 21](#_Toc165265997)

[8. Test Cases For the GUI 22](#_Toc165265998)

[3.1 Screen Shots For Test cases 22](#_Toc165265999)

[Resolved Error Report For Failed Testcases 28](#_Toc165266000)

[Instructions To use the personal Finance tracker 3.0 34](#_Toc165266001)

# **Programme Design**

# **Algorithm**

1.Start

2.Initialize transactions as an Empty Dictionary

3. Define the load Transaction

3.1 Open the JSON file in read Mode

4. Define Save Transactions

4.1 Define global transactions

4.2 Ask from user to save changes

4.3 If user enters “YES”

4.3.1.Open data.JSON File in write Mode

4.4 If user Enters “NO”

4.4.1 Print chabges are not saved

5. Define read\_bulk\_transaction\_from\_file

5.1 Define a empty Dictionary as current\_transaction

5.2 Read Line By line in transaction file

5.3 Split Line into 2 parts

5.4 Append Reavant data into dictionary Relavanatly

6. Define View Transactions Function

6.1 Displaying the the existing transactions in the JSON File  
 Ina GUI Windoe

7. Define Update Transaction Function

7.1 Display all exsist transactions with Prompt

7.2 Allow the user update amount,type or Date for selected transaction

8. Define delete transactions

8.1 Display existing transactions with GUI,prompt the user to select a transaction to delete

8.2 Allow user choose a existing purpose

8.3 Allow user toa choose a transaction under existing purpose

8.4 Deleted the user choosed one

9. Define Display Summary

9.1 Calculate the total incomes,total expenses number incomes,number of expenses &

Net value

9.2 Display following Statistics

10. Define Main Menu

10.1 Continously display the main menu and prompt the user to choose an Option

10.2 Based on the user's choice,Execute the corresponding Functionality

# **Data Structure Design**

transactions={ }

//Every transaction saved in a dictionary into the main transactions Dictionary

Ex:- {"Groceries": [{"amount": 150, "date": "2024-02-03"},

{"amount": 75, "date": "2024-02-15"}]

,"Salary": [{"amount": 1000, "date": "2024-02-01"}]}

# **Pesudocode Design**

Start

Import Tkinter

Import json

Transactions={ }

Class FinancetrackerGUI

Function \_\_init\_\_(self,root)

Defining the root for GUI Window

Giving a title for the GUI

Setting the size of the GUI Window

Adding iconbimap for the Window

Calling the self widgets function

Assigning load transactions with transactions.json to transactions variable

End Function

## Function create\_widgets

Defining a frame for the treeview

Creating the headings of the treeview

Defining the columns of the treeview

Pack the table frame

Pack the tree view

Creating the scrollbar for the treeview

Pack the scrollbar

Configure the maintable

Defining frame for the search

Creating the tk entry

Creating tk search Button

Pack the search frame  
 End Function

## Function Display transactions

Using the for loop to get the children in the Json file

Assing them into transaction List

Append them to suitable columns separately in Tree View

End Function

## Function Search Transactions

Get the user entered the search entry and assign It to the search query

Remove the selected one previously

Using For loop to get the data in the tree view

## Function Sort By Column

Get the columns data separately

If col==amount

Convert data in the column amount to float

Sort data in the column

Else

Sort the data

End If

End Function

## Function load\_transactions()

Set file\_path=data.json

Open file\_path in readmode as json\_file

Set transaction=Json.parse(json\_file)

End Function

## Function save\_transactions()

Prompt user Do you want to save(Yes/No)?

If user enters “Yes” then

Set file\_path=data.json

Open file path in write mode as json\_file

JSON,parse(transactions,json\_file)

Else If user enters “No” then

Display Transaction details are not saved

Else

Disaplay Invalid Option Enter a valid Option

End If

End Function

## Function add\_transactions()

Prompt User to input the purpose of transaction

If user entered purpose Already exsist

Display asking from user save current transaction under existing Programme

If user entered “Yes”

Prompt by asking user the amount

If user entered less than or equals to 0

Display Amount Cannot be Negative

End If

Prompt user to enter the Transaction Type

If transaction type not in “Income” or “Expenses”

Display Invalid Input Enter only Income or expense

Else

Prompt user to enter the date

End If

Append all the user entered inputs to the trnasactions[purpose] list

Elif user enters “No”

Return

Else

Display Please Enter only Yes or No

End If

Else

Prompt User to enter Amount

If amount<=0

Display Amount cannot be negative

Else

Prompt user to Input Transaction Type

End If

If user inputed transaction type not in “income” or “expense”

Display Enter Only “Inocme” Or “Expense”

Else

Prompt user to enter the data in Format DD/MM/YYYY

End If

Append all data entered data into transactions[Purpose] list

End If

End Function

## Function view\_transactions

If not transactions

Display there is No transactions

Else

For purpose,details In transactions

Display The transaction Purpose

For transaction in details

Display amount,type,date one by one

End If

End Function

## Function Update\_transaction

Calling load\_transactions

Calling view transaction

Prompt User to enter the purpose

If user Entered Purpose Not in transactions

Display Enter an Existing Purpose

Else

Prompt user to select the transaction under the user entered Purpose

Prompt user to sleect the transaction

If transaction user entered is <=0 or it is greater than the length of transaction under a purpose

Display a valid transaction number to edit

Else

Subtract one from user inputed transaction Number

Prompt user to edit one from Amount,Type,Date

If user entered Amount

Prompt user to enter the amount

Update in the transactione

Elif user entered Type

Prompt user to enter the type

Update in the transaction

Eif user entered date

Prompt user to enter the Date

Update in the transaction

Else

Display Invalid Input Please enter Type ,Amount or date

End If

End If

End Function

## Function Delete\_transactions

Call Load Transactions Function

Display Existing Transactions

Prompt use asking whether he wants to delete all transactions under a purpose

If user enter “Yes”

Prompt user to enter the purpose to dlete

Delete all transactions under the purpose

Print successful Message

Else

Print Error Message

If User Answers “No”

End If

Prompt to user to enter the purpose

If user entered purpose exsisit in transactions

Disaplay Transactions under that Purpose

Print Success Message

Else

Print Error Message

End If

End If

Else

Print error Message

End If

## Function display\_summary

Load transactions

Initialize dictionary

For each transaction in transactions

If transaction’s Purpose not in the summary dictionary

Add Purpose to the summary sictionary

Else

Increment the count for that purpose in the summary dictionary

End If

Print “Transactions Summary

For each purpose and its count in the summary dictionary

Print the purpose and its Count

End Function

## Function Main()

Call load\_transactions to load transactions from the file

While True Do

Display Personal Finance Tracker

Display 1. Add Transaction

Display 2. View Transactions

Display 3. Update Transactions

Display 4. Delete Transactions

Display 5. Display Summary

Display 6.Exit

Prompt user to choose one out of Following

End While

If users Choice is’ 1’

Call add\_transactions

Else If Users Choice is’2’

Call view\_transactions

Else If Users Choice is’3’

Call Update\_transactions

Else If users Choice is’4’

Call delete\_Transactions

Else If users choice is ‘5’

Call display\_summary

Else If users choice is ‘6’

Call read\_bulk\_transactions\_from\_a \_file

Else If users Choice is’7’

Display Exiting the programme…

Break

Else

Display Invalid Choice Try Again

End If

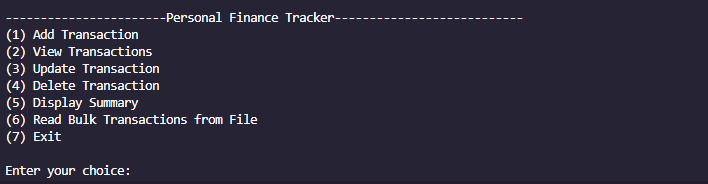
End Function

# **Test cases**

## Main Function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Enter choice option 1 | Run the add\_transaction function | Run the add\_transaction function | Pass |
| 02 | Enter choice option 2 | Runt the view\_transactions  function | Runt the view\_transactions  function | Pass |
| 03 | Enter choice option 3 | Run the update transaction Function | Run the update transaction Function | Pass |
| 04 | Enter choice option 4 | Run the delete transaction function | Run the delete transaction function | Pass |
| 05 | Enter choice option 5 | Run the display summary Function | Run the display summary Function | pass |
| 06 | Enter choice option 6 | Asking user to enter the file name to read  transactions | Read transaction from file | Pass |
| 07 | Enter Choice Option 7 | Exit from the programme | Exit from the programme | Pass |
| 08 | Enter a choice without 1 to 7 | Printing Error message invalid choice | Printing Error message invalid choice | Pass |

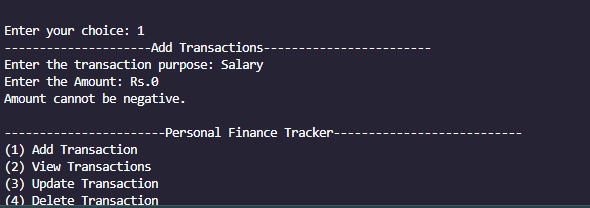
### 1.2 Screenshot For Test Cases

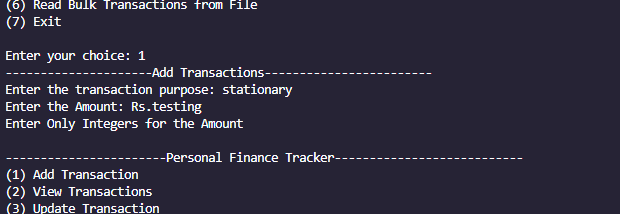


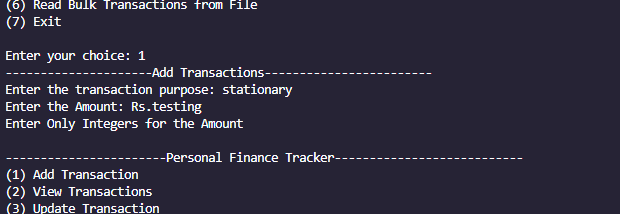
## Add Transaction Function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case | Input | Expected Output | Actual Output | Remarks |
| 01 | Entering Negative amount or 0 for the Amount | Error message | Error Message  Amount canot be negative | Pass |
| 02. | Entering Strings for the amount | Handling The Value Error | Enter amount only in integers | Pass |
| 03. | Spelling mistakes when typing Income or Expense | Handling the error | Crashing The Programme | Fail |
| 04. | When Purpose Write in capital or simple Letters | Transactions are saved Under same purpose | Transactions are saved in 2 different purposes | Fail |

### 2.2 Screenshots For Test Cases







## View Transactions Function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Calling View Transaction function without  Having any data in the data File | Error Message which prints there is not any transactions to view | Error Message which prints there is not any transactions to view | Pass |

### 3.1 Screen Shots For Test cases

A screenshot of a computer screen

Description automatically generated

## Update Transactions Function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Enter Not Existing Purpose to Update | Error Message | Transaction Not Existing Please Enter a existing transaction | Pass |
| 02 | Entering a existing Purpose | Display the Number of Transactions | Displaying the Number of Transactions | Pass |
| 03 | Entering a transaction Number not Exsisting | Displaying a error message Without Crashing Programme | Displaying a Error Message | Pass |
| 04 | Entering String Value For Trnsactions Number | Displaying error message while Handling the Value Error | Displaying Error Message | Pass |
| 05 | Entering a word other than amount,type or Date | Displaying a error  message | Diplaying Please Enter word out of amount,type,Date | Pass |

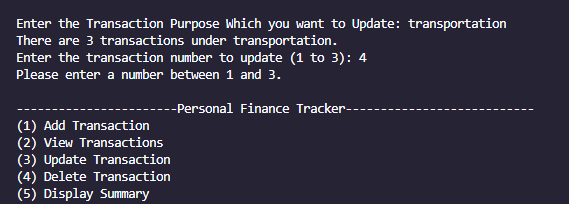
### Screen shots for Test Cases

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

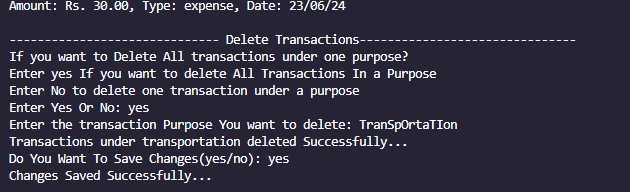
A screenshot of a computer program

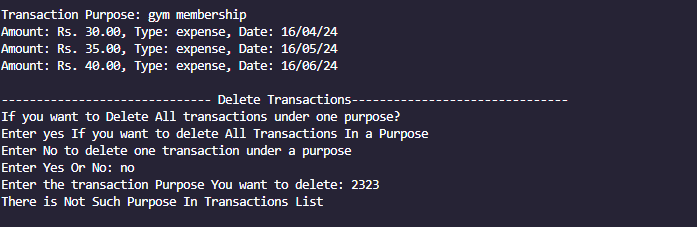
Description automatically generated

## Delete Transactions Function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Choice 04 | Returning to the main main Menu | When opening the delete transaction Without any Json file It Runs The code Without returning To main menu | Fail |
| 02 | Entering Purpose with Mix Cases | Choosing the correct  Transactions Purpose | Choosing the correct  Transactions Purpose | Pass |
| 03 | Deleting Specified Transaction | Deleting the purpose name when we delete all transactions | Not Deleting Transaction Purpose After deleting all transactions one by one | Fail |
| 04 | Entering  Integers | Displaying Error | Displaying Error | Pass |

### Screen shots for Test Cases

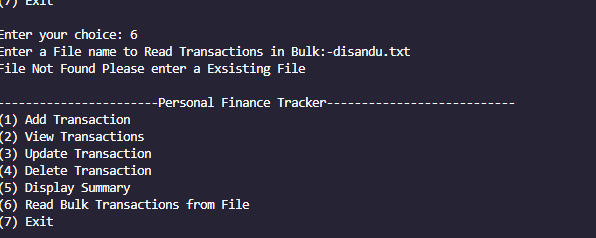




## Read Bulk Transactions Function

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Inputing wrong file | Error message displaying File Not Found | Error message displaying File Not Found | Pass |

### Screen shots for Test Cases



## Test Cases For the GUI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case** | **Input** | **Expected Output** | **Actual Output** | **Remarks** |
| 01 | Calling View Transaction function without  Having any data in the data File | Error Message which prints there is not any transactions to view | Error Message which prints there is not any transactions to view | Pass |
| 02. | Option 2 | Opening the Gui Window when we call the transaction Function | Opening the Gui Window when we call the transaction Function | Pass |
| 03 | Sorting the amount when the amount is a string | Sorting Correctly | Not Sorting the data correctly | Fail |
| 04 | Search again after one search result | Delete previously selected data | Delete previously selected data | Pass |
| 05 | Moving the Scrollbar after adding more data | Show data in the botton | Show data in the botton | Pass |
| 06 | Click the header to sort | Sorting the ascending or descending order when clicking | Sorting the ascending or descending order when clicking | Pass |
| 07 | Entering a data which not exsist | Not highlight any transactions | Not highlight any transactions | Pass |

### 3.1 Screen Shots For Test cases

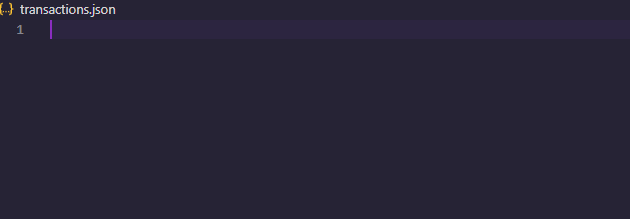


Diagram 1.1(Diagram For Test Case 01)

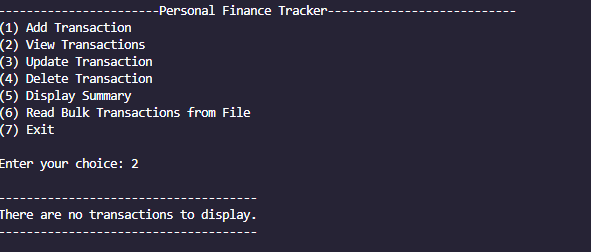


Diagram 1.2(Diagram For Test Case 01)

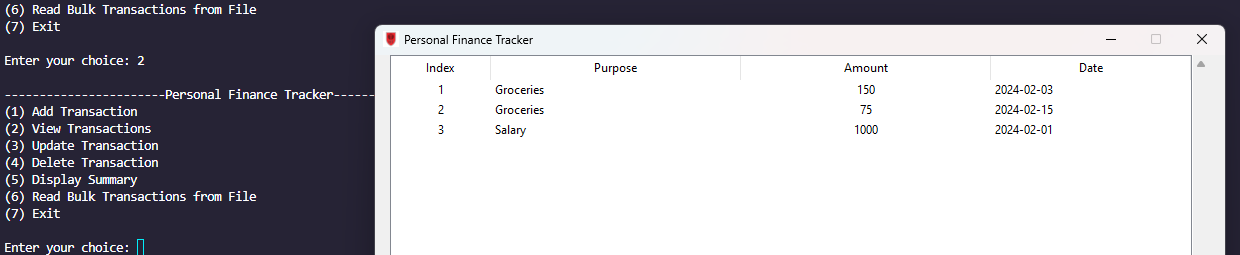


Diagram 2.1(Diagram For Test Case 02)

A screenshot of a computer

Description automatically generated

Diagram 4.1(Diagram For Test Case 04)

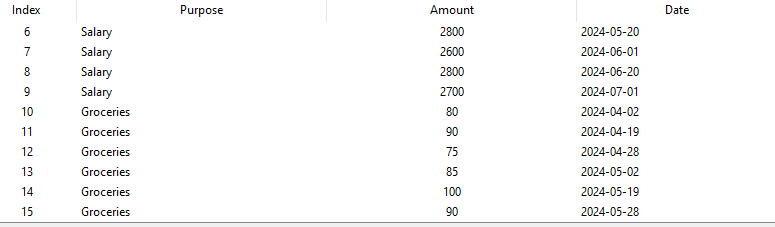


Diagram 5.1(Diagram For Test Case 05)

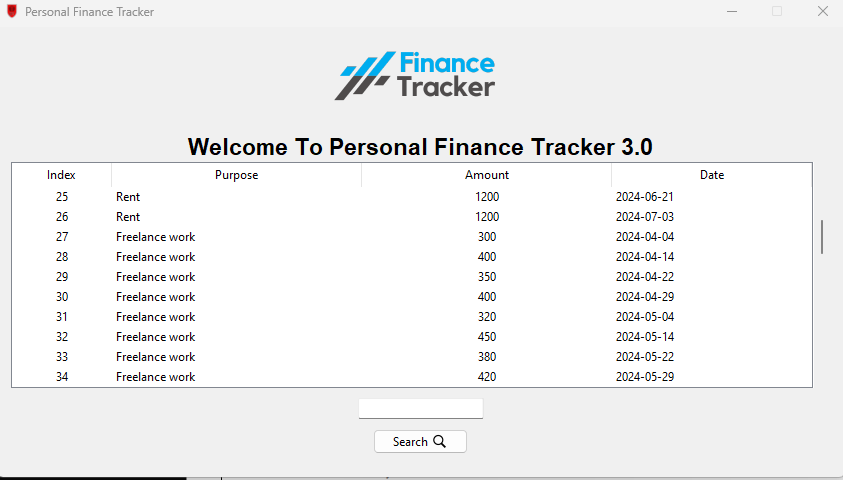


Diagram 5.2(Diagram For Test Case 05)

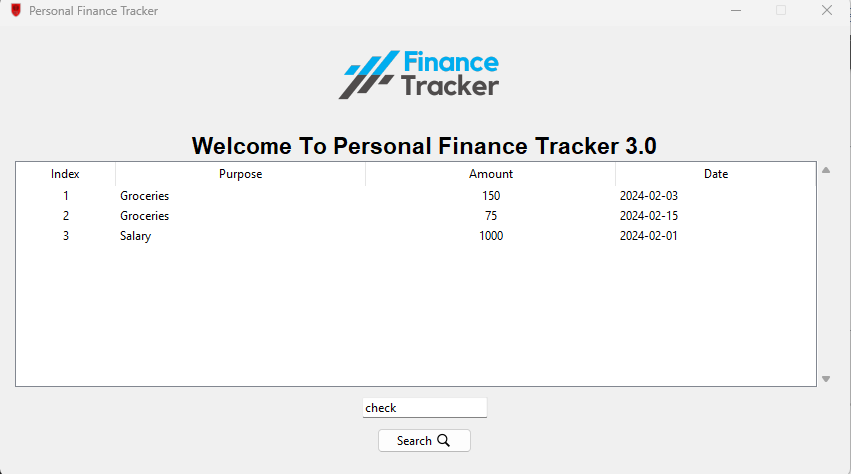


Diagram 6.1(Sorting the the tree view according to purpose)

A screenshot of a computer

Description automatically generated

Diagram 6.2(Sorting the the tree view according to purpose)

A screenshot of a computer

Description automatically generated

Diagram 6.3(Sorting the the tree view according to amount)

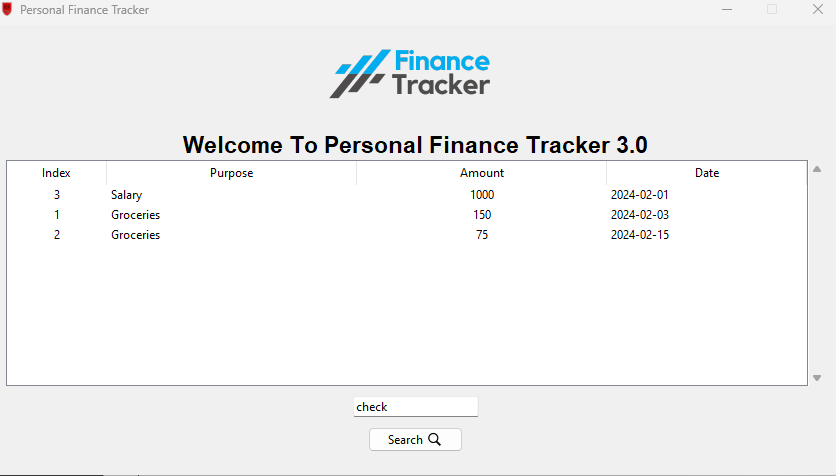


Diagram 6.4(Sorting the the tree view according to amount)

A screenshot of a computer

Description automatically generated

Diagram 6.5(Sorting the the tree view according to amount)

A screenshot of a computer

Description automatically generated

Diagram 6.6(Sorting the the tree view according to date)

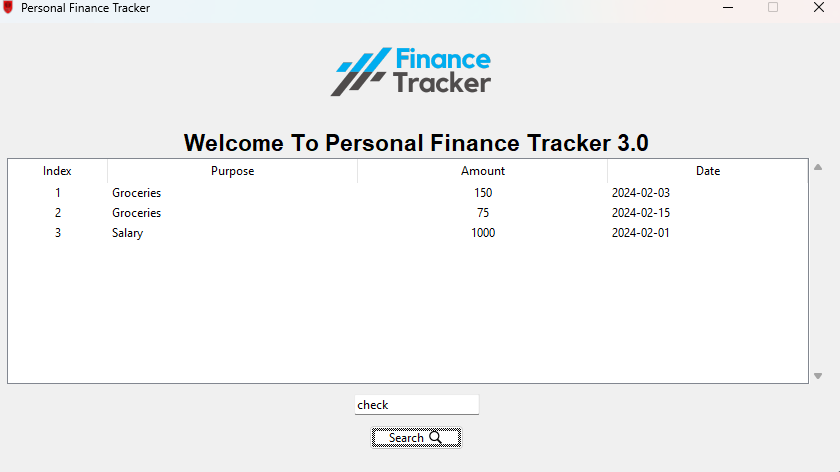


Diagram 8.1(Entering data to which is not in JSON to search)

## Resolved Error Report For Failed Testcases

1. Spelling Mistakes When Typing Income or Expense
2. **Description**

When User do an spelling mistake when entering the words Income or Expense It save in the dictionary without error message

1. **Steps To Reproduce**

Using If condition wheather user entered Income or expense

1. **Screenshot Of the Error**

A computer screen shot of a computer

Description automatically generated

1. **Code Snippet To Resolve Error**

trs\_type = str(input("Enter the Transaction Type (Income/Expense): ")).strip().lower()

*#Check whether the user input Income, expense word correctly*

if trs\_type not in ("income", "expense"):

    print("Invalid type. Please enter 'Income' or 'Expense'.")

*#if user input wrong it return s to enter it again*

    return

1. Writing Purpose In Mix Cases
2. **Description**

When user enters same purpose in 2 times to add transactions

In different cases

1. **Steps To Reproduce**

User .lower() Method to convert user entering purpose into lowercase in every instance

1. **Screenshot of the error**

A screenshot of a computer screen

Description automatically generated

1. **Code Snippet to Resolve Error**

*#Asking user to input the purpose*

purpose = input("Enter the transaction purpose: ").lower()

*#Check whether the user inputed purpose exsist in the transactions*

if purpose in transactions:

    print("This transaction already exists. Do you want to add details to it?")

1. Selecting Choice Number 04
2. **Description**

When user choose choice Number 4 Without there any JSON File The Function Delete transactions Runs without Error Message

1. **Steps To Reproduce**

Use If Condition

1. **ScreenShot Of the Error**

A screenshot of a computer screen

Description automatically generated

1. **Code Snippet To Resolve the Error**

 print("------------------------------ Delete Transactions---------------------“)

if not transactions:

return

print("If you want to Delete All transactions under one purpose?")

print("Enter yes If you want to delete All Transactions In a Purpose")

print("Enter No to delete one transaction under a purpose")

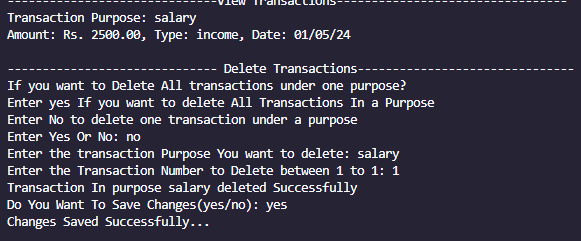
1. Deleting Transactions One by One in One Purpose
2. **Description**

When deleting transactions Under a Purpose One by one,after deleting the Final transaction it remains the purpose without deleting it

1. **Steps To Reproduce**

Using If Condition to check is there is a purpose remaining purpose after that

1. **ScreenShot of the Error**



A screenshot of a computer

Description automatically generated

1. **Code Snippet To Resolve Error**

if transactions[delete\_purpose]==[]:

    del transactions[delete\_purpose]

    save\_transactions()

else:

    save\_transactions()

1. Spelling Mistakes When Typing Income or Expense
2. **Description**

When User do an spelling mistake when entering the words Income or Expense It save in the dictionary without error message

1. **Steps To Reproduce**

Using If condition wheather user entered Income or expense

1. **Screenshot Of the Error**

A computer screen shot of a computer

Description automatically generated

1. **Code Snippet To Resolve Error**

trs\_type = str(input("Enter the Transaction Type (Income/Expense): ")).strip().lower()

*#Check whether the user input Income, expense word correctly*

if trs\_type not in ("income", "expense"):

    print("Invalid type. Please enter 'Income' or 'Expense'.")

*#if user input wrong it return s to enter it again*

    return

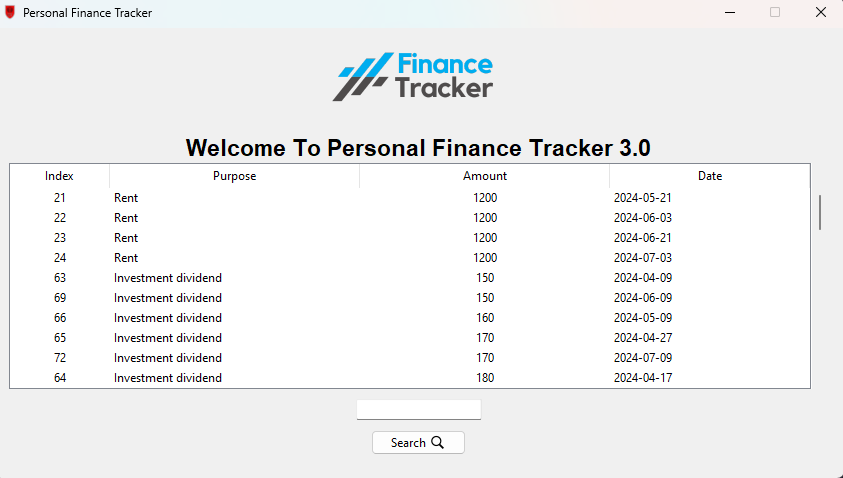
1. The Amount column is not sorting correctly
2. **Description**

When a user sorts the amount column it is not sort the data in a correct way

1. **Steps To Reproduce**

Converting the data type of amount into float without keeping it as an string

1. **ScreenShot of the Error**

****

1. **Code Snippet To Resolve Error**

if col == "Amount":

**self**.data.sort(*key*=lambda *x*: float(x[0]), *reverse*=reverse)

else:

**self**.data.sort(*reverse*=reverse)

    for index, item in enumerate(**self**.data):

**self**.main\_table.move(item[1], "", index)

**self**.main\_table.heading(col, *command*=lambda: **self**.sort\_by\_column(col, not reverse))

## Instructions To use the personal Finance tracker 3.0

1. Install Python and Required Libraries

* Ensure you have the or Higher Installed in your laptop
* If you must a version lower than it you must install the

Tkinter by pip

* Json library is already included in the python if not you must

Install it also.

1. Download the project Files

* Download the project files in the zip file and move all of the files in the zip in to a one folder

1. Navigate to the project directory

* Open Your command prompt and navigate to the directory where you downloaded Or cloned the project files

1. Run the application

* Open the Financial Tracker With GUI(Final One).py and run it on the terminal

1. Using the Main Menu

* The main menu provides several options for managing your personal finance transactions.
* Enter the corresponding number for the desired option and press Enter.
* Follow the on-screen instructions to perform the selected operation.

1. Adding a Transaction

* Select option 1 from the main menu to add a new transaction.
* Enter the transaction purpose, amount, type (income or expense), and date when prompted.
* The transaction will be added to the transactions dictionary and saved in the transactions.json file.

1. Viewing Transactions

* Select option 2 from the main menu to view the transactions in a GUI window.
* The GUI window will display the transactions in a Treeview, with columns for Index, Purpose, Amount, and Date.
* You can search for transactions by entering a query in the search bar and clicking the "Search" button.
* You can sort the transactions by clicking on the column headers (Purpose, Amount, or Date).

1. Updating a Transaction

* Select option 3 from the main menu to update an existing transaction.
* Enter the transaction purpose for which you want to update a transaction.
* Select the specific transaction number you want to update.
* Choose the field you want to update (amount, type, or date) and enter the new value.
* The updated transaction will be saved in the transactions.json file.

1. Deleting a Transaction

* Select option 4 from the main menu to delete a transaction.
* You will be prompted to choose whether you want to delete all transactions under a specific purpose or a single transaction.
* If deleting a single transaction, enter the transaction purpose and the transaction number you want to delete.
* The transaction(s) will be removed from the transactions dictionary and the transactions.json file.

1. Displaying Summary

* Select option 5 from the main menu to display a summary of the transactions.
* The summary will show the total amount of incomes, total amount of expenses, the number of incomes and expenses, and the net value (total incomes - total expenses).

1. Reading Bulk Transactions from a File

* Select option 6 from the main menu to read bulk transactions from a file.
* Enter the filename (including the file extension) when prompted.
* The transactions from the specified file will be added to the transactions dictionary and saved in the transactions.json file.

1. Exiting the Application

* Select option 7 from the main menu to exit the application.
* Any unsaved changes to the transactions will be prompted to be saved before exiting.

Note: The application uses a file named transactions.json to store and load the transaction data. If the file does not exist, it will be created automatically when you add or save transactions.