

Pseudo Code Design

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

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

CONTENT

File handling functions.....	2
load_transactions().....	2
save_transactions().....	2
Feature Implementations.....	3
add_transaction().....	3
view_transactions().....	4
update_transaction().....	4
delete_transation().....	6
display_summery().....	7
Helper Functions.....	8
prompt().....	8
main_menu().....	9

File handling functions

1. load_transactions()

```
BEGIN FUNCTION load_transaction()  
    declare transactions list as global  
    IF (file exists) THEN  
        OPENFILE 'finance.json' FOR read AS file  
        LOADFILE file TO transactions  
        CLOSEFILE file  
    ELSE  
        DISPLAY ("File not found. creating new file")  
        CALL save_transactions()  
    END IF  
END FUNCTION
```

2. save_transactions()

```
BEGIN FUNCTION save_transactions()  
    OPENFILE 'finance.json' FOR write as file  
    save transactions list to file  
    CLOSEFILE file  
END FUNCTION
```

Feature Implementations

3. add_transaction()

```
BEGIN FUNCTION add_transaction ()
    declare transactions list as global
    temporary = []
    WHILE (True) DO
        IF (no errors) THEN
            FOR i = 0 TO 4 DO
                add (CALL FUNCTION prompt(i)) to temporary list
            END FOR
            add temporary list to transactions list
            DISPLAY ("Transaction added successfully")
            BREAK statement
        ELSE
            DISPLAY ("Invalid input")
            CONTINUE
        END IF
        CALL save_transaction()
    END WHILE
END FUNCTION
```

4. view_transactions()

```
BEGIN FUNCTION view_transaction()
    i = 0
    FOR entry in transactions DO
        amount = entry[0]
        category = entry[1]
        transaction_type = entry[2]
        date = entry[3]
        i = i + 1
        DISPLAY "Transaction ID: ", i
        Display "Amount: ", amount
        Display "Category: ", category
        Display "Type: ", transaction_type
        Display "Date: ", date
    END FOR
END FUNCTION
```

5. update_transaction()

```
FUNCTION update_transaction()
    declare transactions list as global
    IF (transactions available in list) THEN
        CALL view_transactions
        IF (no errors) THEN
            GET index_value
            IF 0 < index_value <= length of the number of
sub_lists in global list
                temporary = []
```

```

                                WHILE True DO
                                    IF (no errors) THEN
                                        FOR i = 1 TO 4 DO
                                            temporary = CALL prompt(i)
                                        END FOR
                                        transactions[index_value - 1] =
temporary
                                            BREAK
                                    ELSE
                                        DISPLAY "Invalid Input."
                                    END IF
                                    save_transactions()
                                END WHILE
                            ELSE
                                DISPLAY "Index not valid. Please re-enter."
                            END IF
                        ELSE
                            DISPLAY "Invalid input."
                        END IF
                    ELSE
                        CALL transaction_availability()
                    END IF
                END FUNCTION

```

6. delete_transation()

```
FUNCTION delete_transaction()  
    declare transactions list as global  
    IF (transactions available in list) THEN  
        CALL view_transactions  
        IF (no errors) THEN  
            GET index_value  
            IF 0 < index_value <= length of the number of  
sub_lists in global list  
                remove transactions[index_value - 1]  
                CALL save_transactions()  
            ELSE  
                DISPLAY "Index not valid. Please re-enter."  
            END IF  
        ELSE  
            DISPLAY "Invalid input."  
        END IF  
    ELSE  
        CALL transaction_availability()  
    END IF  
END FUNCTION
```

7. display_summery()

```
FUNCTION display_summery()
    income = 0
    expense = 0
    balance = 0
    FOR entry in transactions DO
        IF entry[2]  "Income" THEN
            income = entry[0]
        ELSE IF entry[2] is equal to "Expense" DO
            expense = entry[0]
        END IF
        balance = income - expense
        DISPLAY "Your total income is {income}"
        DISPLAY "Your total expense is {expense}"
        DISPLAY "Your financial balance is {balance}"
    END FOR
END FUNCTION
```


Helper Functions

8. prompt()

```
FUNCTION prompt(i)
    IF i equal to 0 THEN
        WHILE True DO
            GET amount
            IF amount <= 0 THEN
                DISPLAY "Enter amount higher than zero."
                CONTINUE
            ELSE
                RETURN amount
            END IF
        END WHILE
    ELSE IF i equal to 1 THEN
        RETURN GET category
    ELSE IF i equal to 2 THEN
        WHILE True DO
            GET type
            IF type equal to "i" or "income"
                RETURN "Income"
            ELSE IF type equal to "e" or "expense"
                RETURN "Expense"
            ELSE
                DISPLAY "Invalid Input."
            END WHILE
        END WHILE
    ELSE IF i equal to 3 THEN
        WHILE True DO
            CREATE temporary list as date
```

```
        months = ((months with 31 days), (months with 30
days), (february))

    IF (no errors) THEN
        GET year
        GET month
        IF 0 < month <= 12 THEN
            BREAK
        ELSE
            DISPLAY "Invalid month"
        END IF
    WHILE True DO
        GET day
        IF month IN months[0] THEN
            IF 0 < day <= 30 THEN
                BREAK
            ELSE
                DISPLAY "Invalid day"
                CONTINUE
            END IF
        ELSE IF month IN months[1] THEN
            IF 0 < day <= 31 THEN
                BREAK
            ELSE
                DISPLAY "Invalid day"
                CONTINUE
            END IF
        ELSE IF month IN months[2] THEN
            IF year is a leap year THEN
                IF 0 < day <= 29 THEN
                    BREAK
                ELSE

```

```

                                DISPLAY "Invalid day"
                                CONTINUE
                            ELSE
                                IF 0 < day <= 28 THEN
                                    BREAK
                                ELSE
                                    DISPLAY "Invalid day"
                                    CONTINUE
                                END IF
                            END IF
                        END WHILE
                        APPEND year to date list
                        APPEND month to date list
                        APPEND day to date list
                        RETURN date
                    ELSE
                        DISPLAY "Enter a number."
                        CONTINUE
                    END IF
                END WHILE
            END IF
        END FUNCTION
```

9. main_menu()

```
FUNCTION main():  
    CALL load_functions  
    display menu  
    GET input  
    while True do  
        if input is equal to 1 then  
            CALL add_transaction()  
  
        else if input is equal to 2 then  
            CALL view_transactions()  
  
        else if input is equal to 3 then  
            CALL update_transaction ()  
  
        else if input is equal to 4 then  
            CALL delete_transaction ()  
  
        else if input is equal to 5 then  
            CALL display_summery()  
  
        else if input is equal to 6 then  
            display "Exiting program"  
            BREAK loop  
  
        else do  
            display "Invalid choice. Please try again"  
        END if  
    END while  
END FUNCTION
```