Technical Design Document 03

Winson Ma

**Description**: Prompts users to enter their monthly expenses, including the type and amount of each expense. It then uses the reduced function to calculate the total expenses, identifying the highest, and identifying the lowest. Then it displays a summary of the total amount of expenses.

**Function:** get\_expenses()

* **Description**: Collects expense information from user.
* **Variables**:
* Expense (list of tuples): Stores the expenses type and amount
* Expenses\_type (str): Stores the type of expenses entered by user.
* Amount (float): Stores the amount entered by user.
* **Logical Steps**:

1. Initialize an empty list.
2. Enters a while loop that continues until user types ‘done’.
3. Prompts the user to enter expense type.
4. If user types ‘done’, the loop stops.
5. Prompts the user to enter amount.
6. Try-except block to handle potential error if the user enters a non-number amount.
7. If the amount is valid, appends a tuple to the expenses list.
8. Returns expenses list.

* **Return**:
* Expenses (list of tuples): List of tuples, where each tuple contains the expense type and amount.

**Function**: analyze\_expenses(expenses)

* **Description**: Analyzes the expenses to calculate the total, highest, and lowest expenses using the reduce function.
* **Parameters**:
* Expenses (list of tuples): List of tuples, where each tuple contains the type and amount.
* Variables:
  + Total\_expense (float): Stores the total expenses.
  + Highest\_expense (tuple): Stores expense type and amount of highest.
  + Lowest\_expense (tuple): Stores expense type and amount of lowest.
* **Logical steps**:

1. Handles the case of an empty expenses list. If empty, returns 0, None, None.
2. Calculates the total expenses and a lambda function that sums the amount.
3. Finds the highest using reduce and a lambda function that compare amount and returns the tuple highest amount.
4. Finds the lowest using reduce and a lambda function that compare amount and returns the tuple with lower amount.
5. Returns total\_expense, highest\_expense, and lowest\_expense

* **Returns**:
  + total\_expense (float): Total
  + highest\_expense (tuple): Expense type and amount of highest.
  + lowest\_expense (tuple: Expense type and amount of lowest.

Function: Main()

* Description: Main function that commands the programs execution.
* Variables:
  + Expense (list of tuples): Stores expense returned by get\_expenses().
  + Total (float): Stores total expenses returned by analyze\_expenses().
  + Highest (tuple): Stores highest returned by analyze\_expenses().
  + Lowest (tuple): Stores lowest returned by analyze\_expenses().
* Logical Steps:

1. Calls get\_expenses() to get user’s answer.
2. Calls analyze\_expenses() to analyze expenses.
3. Prints summary of expenses.
4. Handles the case of no expenses entered by user.

* Returns:
  + None

GitHub: <https://github.com/FamiliarotherW/COP2373-Assignments-03>

