Project Report of Account Book

Group 6 – Zhang, Jiang, Ma

1. Introduction

As the previous introduction we've submitted, we would like to create an account book which can record the expenses, classify and store the records into different categories, modify the records and give the brief graphs about proportion of different categories. At last, the account book will locally generate a .txt file, whose name is user's name, in a folder of the computer.

In order to implement this project, we would use nested array to store the data, build several classes for expenses and incomes, interact with users by inputting commands and finally generate the graph. For example, we set a path of usernamedata.txt as "E:\VScode\get_userdata.txt"(get_user here is a variable) in the ***InterfaceStructure.py***, we highly suggest you can also create a folder named VScode at your E:disk, or even better, name the path by yourselves in ***InterfaceStructure.py***, in these ways, you will see surprise in the folder after press ***5(quit from the program)***.

Generally speaking, the account book implementation will be divided into two parts, ***InterfaceStructure*** and ***AccountClasses***. Here are the more specific details.

2. *InterfaceStructure*

This structure includes the interface of account book and the execution of our program.

In interface part, it roughly has the same sequence of the execution part. There are some basic function intending to instruct the users like ***welcome()*** and ***begin()***.

Firstly in the ***prompt()*** function, we ask the user if he/she has an account. Depending on the result that user provides, he/she will choose to create a new account or log in his/her account.

For the ***trans_username_wallet()*** function, we create an instance named by user, we will call this function later.

After having the user name and password, account book will automatically generate a .txt file to store the information in the function ***openfile()***.

For now, we just create several categories in ***defaultcategories(),*** like "Shop", "Transport" and "General", the account book also allow users to create their own categories in ***printcategory()***, ***get_choice_2()*** and ***create_category()*** functions. On the contrary, if users want to delete category, function ***get_choice_5()*** and ***deletecategory()*** allow them to do so.

The last part of interface is showing not only the categories but also the proportion of users' data. Function ***get_choice_4()*** ask users to make a choice and we create ***showallcategory()*** and ***showcategory()*** to implement the instructions. Function ***showdistribution()*** provides the graphs so that users can briefly view the configuration of expenses and incomes.

In execution part, program just calls the functions by the order they were defined in ***Interface*** part. Once the user press ***5(quit)***, his/her data will be stored as a local file in time; once the user press ***4(back to log in)***, his/her data will be stored firstly, then another user can log in or create his/her account continually in this running program, ***welcome()***, ***trans_username_wallet()*** etc. will be called again to present a new interface for this new user.

3. *AccountClasses*

There are two classes in ***AccountClasses*** part, class ***wallet*** and class ***expense***.

In the ***expense***, we just simply generate the data of expense.

In the ***wallet***, there are some sub-functions that allow users to modify their account books, such as ***addcategory()***, ***deletecategory()*** and ***findcategory()***. Account book also provides functions to modify the expense in the ***wallet***, like ***addexpense()***, ***findexpense()***, and ***deleteexpense()***.