

Queen's University

**Assignment:
Quinterac part 4**

**Course:
CMPE 327**

**Prepared for:
Lecturer: Ding, Steven**

**Group Number:
23**

**Prepared by:
Adams Liu: 10187602
Eric Chau: 10141119
Ian Chang: 10190262**

**Due Date:
November 10th, 2019**

Structure Overview

The backend takes in the merged transaction summary file and the master accounts file as inputs, and outputs the updated version of the master accounts file and the new valid accounts list. The master accounts file has all its account balances defaulted to 500000 cents. To start, information on each account was read from the master accounts file and was stored in a nested dictionary, where the key of each dictionary was set to an account number, and the balance and name as the dictionary attributes. After the dictionary was set up, each line on the transaction summary file was processed, where each transaction was led to its respective function (its definition can be seen in the table below) to check for fatal errors and to update the master accounts dictionary. Once the EOS line on the transaction summary file was encountered, the contents of the nested dictionary is written on the master accounts input file, where the existing contents on the file is overwritten. The valid accounts list file is also updated, by writing the account numbers on the master accounts dictionary onto the file, where the existing contents on the file is also overwritten. Finally, the updated version of the master accounts file and the valid account list file is outputted.

Function Documentation

Functions	Function Description
deposit()	Adds specified amount on top of existing customer balance only if it does not exceed over \$999,999.99.
withdraw()	Takes away specified amount from existing customer balance, if balance is insufficient then it will print an error.
transfer()	If the funds are sufficient from account2 then it will add the specified amount to account1 and take away that amount for account2. (Note: Error will occur if balance added exceeds over \$999,999.99).
create_acc()	Create an account with a non-existing account number, and any account name. If account number already exists, the back office will throw an error. Else, the account will be successfully created with a default balance of \$500000.
delete_acc()	Delete an existing account with matching account name. If exists account name does not match with the account number provided or if the balance of the current account is not 0 then an error will be thrown.
end_sesh()	Creates a "valid accounts list" and a "new master accounts file" output given the updates provided in the transaction summary file.