

Back Office Design document

Team: Zero Defects



|  |  |
| --- | --- |
| Name | Student Number |
| Muhammad Usman Majeed | 10086980 |
| Jessica Nahulan | 10029341 |
| Johan Cornelissen | 10098176 |

Contents

[QBasic Application Usage: 3](#_Toc496037025)

[Architecture Description: 3](#_Toc496037026)

[Solution Structure: 4](#_Toc496037027)

[Classes: 4](#_Toc496037028)

[Methods: 5](#_Toc496037029)

# QBasic Back Office Application Usage:

The following provides a brief overview of how to use the QBasic Back Office python application.

The back office takes four arguments: the previous day Master Accounts File, the Merged Transaction Summary File, and file to use for the new Master Accounts File created, and the file to use for the new Valid Accounts File created.

The previous day Master Accounts File is created by the last run of the back-office application. The Merged Transaction Summary File will be provided by an intermediate script that concatenates the Front End’s Transaction Summary Files. Lastly the new Master Accounts File created will be used by the next day’s back office, and the new Valid Accounts File will be used by the next day’s Front End sessions.

**How to run the back-office application:**

./BackOffice.py masteraccount.txt transactionsummary.txt newmasteraccount.txt newvalidaccount.txt

# Architecture Description:

**Update for back-office!**

The QBasic front end follows an object-oriented approach made up of 4 classes. These 4 classes consist of QBasic, TxnProcess, Utility, and ErrorHandler classes. The QBasic class is responsible for collecting user input, processing provided input files, and produces output via standard out as well as through the transaction summary file. The implementation of each transaction command is done through the TxnProcess class. Each transaction code corresponds to a TxnProcess method which configures the commands constraints and behavior. Logic that is required in multiple places, like checking what kind of user is logged in, is done through methods in the Utility class. The Utility class members are used by multiple TxnProcess methods and ensures the architecture is modular. Lastly, the ErrorHandler class is used throughout all the classes to allow for clean and concise error reporting. A simple ErrorHandler method can be invoked using a unique error code to allow for a more detailed error message to be displayed to the user. A class relationship diagram is provided below in the “Solution Structure” section to illustrate how the classes interact with one another.

# Solution Structure:

## Classes:

Log Output

BackOffice

MasterAccounts.txt

NewMasterAccounts.txt

TransactionSummary.txt

ValidAccountsFile.txt

Account Object(s)

TxnProcess

Utility

## 

ErrorHandler

## Methods:

|  |  |  |
| --- | --- | --- |
| Class | Function | Intentions |
| BackOffice | Main | To prompt user for user input and call correct TxnProcess functions based on the transaction code entered by the user. |
| TxnProcess | txn\_new | Function to process a user login transaction code.  Sets user type (machine,agent) and reads the valid accounts file. |
| txn\_del | Function to process a user logout transaction code.  Logs out the current user and creates transaction summary file. |
| txn\_dep | Function to process a user deposit transaction code.  Checks if account number and deposit amount are valid. |
| txn\_wdr | Function to process a user createacct transaction code.  Checks if account number and account name are valid before creating account. |
| txn\_xfr | Function to process a user deleteacct transaction code.  Checks if account number and account name are valid before deleting account. |
| txn\_eos | Function to process a user withdraw transaction code.  Checks if account number and withdraw amount are valid before processing withdrawal. |
| Account | \_\_init\_\_ |  |
| get\_account\_num |  |
| get\_account\_name |  |
| get\_account\_balance |  |
| set\_account\_balance |  |
| Utility | process\_master\_account | Function to process valid accounts file.  Valid account numbers are stored in TxnProcess.valid\_acc\_list |
| create\_master\_valid\_account\_files | Function to initialize withdrawal amounts for each valid account. |
| ErrorHandler | process\_error | Function to send error string to stdout. |
| \_\_init\_\_ | Initialize error code to error string mapping. |