

Design document

Team: Zero Defects



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

Contents

[Architecture Description: 3](#_Toc496034629)

[Solution Structure: 3](#_Toc496034630)

[Classes: 3](#_Toc496034631)

[Methods: 4](#_Toc496034632)

# Architecture Description:

The QBasic front end follows an object oriented approach made up of 4 classes. These 4 classes consist of main, TxnProcess, Utility, and ErrorHandler class. The main 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:

Main

Utility

ErrorHandler

TxnProcess

User Input/Ouput

## 

## Methods:

|  |  |  |
| --- | --- | --- |
| Classe | Function | Intentions |
| Main | Main | To prompt user for user input and call correct TxnProcess functions based on the transaction code entered by the user. |
| TxnProcess | txn\_login | Function to process a user login transaction code.  Sets user type (machine,agent) and reads the valid accounts file. |
| txn\_logout | Function to process a user logout transaction code.  Logs out the current user and creates transaction summary file. |
| txn\_deposit | Function to process a user deposit transaction code.  Checks if account number and deposit amount are valid. |
| txn\_createacct | Function to process a user createacct transaction code.  Checks if account number and account name are valid before creating account. |
| txn\_deleteacct | Function to process a user deleteacct transaction code.  Checks if account number and account name are valid before deleting account. |
| txn\_withdraw | Function to process a user withdraw transaction code.  Checks if account number and withdraw amount are valid before processing withdrawal. |
| txn\_transfer | Function to process a user transfer transaction code.  Checks if "From" and "To" account number and transfer amount are valid before processing transfer. |
| Utility | process\_account\_file | Function to process valid accounts file.  Valid account numbers are stored in TxnProcess.valid\_acc\_list |
| initiaze\_withdraw\_totals | Function to initialize withdrawal amounts for each valid account. |
| create\_txn\_msg | Function to form the transaction message to be added into the transaction summary file. |
| create\_txn\_summary\_file | Function to write all cached transaction messages to transaction summary file. |
| is\_account\_valid | Function to check if passed in account number is valid. |
| is\_account\_unique | Function to check if passed in account number is unique (or new). |
| is\_amount\_valid | Function to check if passed in amount is within limits for machine and agent users.  Used by deposit, withdraw, and transfer transaction commands. |
| is\_name\_valid | Function to check if passed in account name is valid. |
| is\_within\_withdraw\_limit | Function to check if passed in account will surpass daily withdraw limit by completing pending withdrawal.  Returns False if limit will be reached with pending withdrawal, true if withdrawal is valid. |
| ErrorHandler | process\_error | Function to send error string to stdout. |
| \_\_init\_\_ | Initialize error code to error string mapping. |