**main**

This method takes in both the accountListFile location and transacionSummaryFile location via parameters, and then calls the startUp function which would begin the session.

**startUp**

This method introduces the program and begins the session by calling the login method and asking for a command. The user’s command will then be validated, and the method corresponding to the command will be called. This menu is placed in a while loop to ensure the program does not end once a transaction is completed (unless logout is entered).

**login**

This method asks the user to enter either agent or machine mode when logging in. It is set to loop if the input is invalid, until the user enters a valid mode. This then assigns the agent variable a Boolean value which will be used in a variety of functions to check the session mode.

**logout**

This method, once called, notifies the user that the session will close, and then adds the logout transaction to tsfQueue. After this, the transaction summary file will be written into using all thedata gathered in the tsfQueue, and the session will have formally ended.

**createAcct**

This method takes in the user input for account creation, and then validates the input by checking the mode they’re in (agent/machine) and validity of the account # and name (if it is valid and already exists or not). if all is valid and the account # does not yet exist, the function will add the appropriate account creation line into the tsfData queue, which will then be fed into the back-end transaction summary file at the end of the session. Else, it will display an error message and return users to the main selection page.

**deleteAcct**

This method takes in the user input for account deletion, and then validates the input by checking the mode they’re in (agent/machine) and validity of the account # and name (if it is valid and currently exists or not).If all is valid and the account # currently exists, the function will add the appropriate account deletion line into the tsfData queue, which will then be fed into the back-end transaction summary file at the end of the session. Else, it will display an error message and return users to the main selection page.

**deposit**

This method takes in the user input for depositing into an account, and then validates the input by checking the account # (if it is valid and currently exists or not) and the amount to deposit (if it is a number within the allowed range specified by the mode). the function will add the appropriate deposit transaction line into the tsfData queue, which will then be fed into the back-end transaction summary file at the end of the session. Else, it will display an error message and return users to the main selection page.

**withdraw**

This method takes in the user input for withdrawing from an account, and then validates the input by checking the account # (if it is valid and currently exists or not) and the amount to withdraw (if it is a number within the allowed range specified by the mode). the function will add the appropriate withdraw transaction line into the tsfData queue, which will then be fed into the back-end transaction summary file at the end of the session. Else, it will display an error message and return users to the main selection page.

**transfer**

This method takes in the user input for transfering to/from an account, and then validates the input by checking the account #’s (if they’re both valid and currently exist or not) and the amount to transfer (if it is a number within the allowed range specified by the mode). the function will add the appropriate transfer transaction line into the tsfData queue, which will then be fed into the back-end transaction summary file at the end of the session. Else, it will display an error message and return users to the main selection page.

**isAllDigits**

This method is a common helper function that is called by multiple larger methods. This takes in a string as a parameter, and verifies that the string consists of only integers. It will then return a Boolean value depending on the outcome of the process (true=numerical string, false=non-numerical string).

**accountNumberValid**

This method is a common helper function that is called by multiple larger methods. This takes in a string as a parameter, and verifies that the string follows all rules and constraints for account numbers (such as numerical, length of 7, cannot start with 0).

**accountNameValid**

This method is a common helper function that is called by multiple larger methods. This takes in a string as a parameter, and verifies that the string follows all rules and constraints for account names (such as no leading/trailing spaces, alphanumeric only, length >=3 and length <=30).

**writeTransactionsToSummaryFile**

This method is called by the logout method, and enters all the data from tsfQueue into the transaction summary file.