Design Document

Jordan McGregor – 10052770

Nicholas Petrielli – 10107308

Andrew Storus – 10103737

Class FrontEnd – main class of the program. When a valid login is commanded from Std In, a session object is created. The class also takes in the accountsFile and transactionFile to be written by the session object.

Class Session – A class that is used to create a session after a successful login, the session has a Boolean so that the program can know if it is logged in as an agent or atm. Commands are read in through Std In and if they are a valid command the corresponding method within the class will be invoked. The class contains a class object for ValidAccounts as well as a Transactions class for the transactionList.

Session.runCommand() – invoked by the front end to process commands on Std In. Other transaction methods are then invoked from this method.

Session.create() – the create account transaction is added to transactionList. Error message is given if not in atm mode.

Session.delete() – account is removed from accountList. Transaction added to transaction list. Error message is printed if not in atm mode.

Session.withdraw() - invoked whenever the withdraw command is read in. This method uses the FrontEndValidator object to check if the account requested is valid. The transaction is then added to the transactionList.

Session.deposit() - invoked whenever the deposit command is read in. This method uses the FrontEndValidator object to check if the account requested is valid. The transaction is then added to the transactionList.

Session.Transfer() – invoked when the transfer command is read in. The user is prompted for the destination account and the origin account numbers. These are checked for validity with the FrontEndValidator and then the transaction is added to transactionList.

Session.logout() - when this method id invoked the transaction summary file is written, and then the current session is destroyed.

Class ValidAccounts – a class that contains all of the valid accounts read in from the valid account file in an array of strings. The object is instantiated with the accountsFile. The class also contains a list to keep track of the amount withdrawn on each account in a session.

ValidAccount.createAccout(int accountNumber) – adds an account to the list of valid accounts. Writes the transaction to the summary file. The account number to be created is first checked against the list of existing accounts.

ValidAccounts.withdrawAmount(int accountNumber, int amount) – adds an amount to the total amount withdrawn on an account number during a session.

ValidAccount.deleteAccout(int accountNumber) – deletes an account from the list of validAccounts.

Class: Transactions – Each session will instantiate a transaction object. This object will be used to keep track of all transactions to be written to the transaction summary file after a logout. The class member transactionList is a list of strings of transactions completed in the session.

Transactions.appendToTransactionFile(string TransactionName) – Adds a transaction to the existing transaction list.

Transactions.writeTransactionFile() – writes everything in the transactionList to the transaction summary file.

Class FrontEndValidator – used to validate amounts withdrawn and deposited and account numbers to be deleted and created.

FrontEndValidator.checkValidAccount(int accountChecking, list validAccounts, invalidAccounts) – checks if the given account number is legal for creation.

FrontEndValidator.checkValidAmount(int amount, bool isAdmin) – checks if the given amount is of legal size.

FrontEndValidor.checkWithdrawAmount(int amount, int accoutNumber, list withdrawalAmount, bool isAdmin) – checks if a withdrawal amount is legal and if an account has gone over the withdrawal limit in a given session. Limits change depending on admin status.

FrontEndValidator.isValidAccountNumberToCreate(int accountNumber) – checks if the account number to be created is valid.

FrontEndValidator.isValidAccountNameToCreate(string accountName) – checks if the account name to be created is of valid size and contains valid characters.