**CSCI 3060U/SOFE 3980U  
Course Project Assignment #3: Front End Requirements Testing  
Team name:** Untitled  
Albert Fung 100520898  
Karan Chandwaney 100472699  
Calvin Lo 100514352

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **What they are intended to test** | **Nature of failure** | **Error in the code** | **Actions taken to fix it** |
| **loginlogout** | | | | |
| 01\_loginlogout\_std\_session | Test if session asks for account holder’s name in a standard session | test case output file and transaction file missing informations | None | changed output file to reflect correct output  changed the input file to correct the account name  changed the trans file account number from 00001 to 00000 |
| 02\_loginlogout\_begin\_transactions | Test if the session reads in current bank account file and accepts transactions after kind and name are accepted | test case output file and transaction file missing informations  the test case trans file have the logged the transaction that did not complete successfully | missing method to check if the input is equal to transaction name or not. | changed output file to reflect correct output  changed the input file to correct account name  changed the trans file account number from 00001 to 00000  added  isTransaction()  check if the input is transaction every time the program read command  changed the trans file to correct transaction |
| 03\_loginlogout\_not\_logged\_in | Test if other transactions are accepted when user is not logged in and logout transaction is only accepted when logged into a session | test case output file and transaction file missing informations  does not create transaction file in some cases | missing for writing an empty transaction file when not logged in | create a new method a FileWriter that create a new File  create the file once the session start |
| 04\_loginlogout\_subsequent\_login | Test if subsequent logins are accepted when user is not logged out | test case output file and transaction file missing informations  the test case trans file have the logged the transaction that did not complete successfully | while login into a non exist account name, the program still record as logged in | changed output file to reflect correct output  changed the login mode to ‘N’ if the account name is not valid  changed the trans file to correct transaction |
| 05\_loginlogout\_unprivileged | Test if only unprivileged transactions are accepted after standard login | test case output file and transaction file missing informations  the test case trans file have the logged the transaction that did not complete successfully | None | changed output file to reflect correct output  changed the trans file to correct transaction |
| 06\_loginlogout\_privileged\_transactions | Test if after admin is logged in, all transactions are accepted | test case output file and transaction file missing informations  the test case trans file have the logged the transaction that did not complete successfully | None | changed output file to reflect correct output  changed the trans file to correct transaction |
| 07\_loginlogout\_disabled\_account | Test if the session can login to a disabled account but cannot do any transaction | Not a needed test case, login did not check for disable account or not | None | deleted test case. |
| 08\_loginlogout\_logged\_out | Test if bank account transaction file is updated correctly | test case output file and transaction file missing informations  the test case trans file have the logged the transaction that did not complete successfully | None | changed output file to reflect correct output  changed the trans file to correct transaction |
| 09\_loginlogout\_logged\_out\_transactions | Test if the session stops accepting any transactions except login | test case output file and transaction file missing informations  the test case trans file have the logged the transaction that did not complete successfully  does not change input type | the input type keep as “file input” after reading all the rows | changed output file to reflect correct output  changed the trans file to correct transaction  change the input type to terminal when reached last row |
| **withdrawal** | | | | |
| 01\_withdrawal\_prompt | Test if the session ask for account holder’s name if logged in as admin | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | none | corrected grammatical errors |
| 02\_withdrawal\_acc\_name\_validation | Test if the bank account is a valid account for the account holder currently logged in | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | did not check if the account name and number match | changed output file to reflect changes  changed input file to reflect changes  corrected grammatical errors |
| 03\_withdrawal\_acc\_num | Test if the session ask for account number | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | none | corrected grammatical errors |
| 04\_withdrawal\_namenum | Test if the session only allow account holder name and account number entered in admin mode matched. If not matched, the login should be rejected | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | did not check if account name and number match | changed output file to reflect changes  changed input file to reflect changes  corrected grammatical errors |
| 05\_withdrawal\_money | Test if session ask for the amount to withdraw | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | none | corrected grammatical errors |
| 06\_withdrawal\_formatvalidation | Test if session only allow the amount format in ####.## where # is 0-9 | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | did not check if the amount format in ####.## or not | changed output file to reflect changes  changed input file to reflect changes  corrected grammatical errors |
| 07\_withdrawal\_stdmode | Test if the amount to withdraw in standard mode can only be less or equal than $500.00 | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | did not check condition | changed output file to reflect changes  changed input file to reflect changes  corrected grammatical errors |
| 08\_withdrawal\_adminmode | Test if the amount to withdraw in admin mode can be unlimited (over $500.00) | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | did not check condition | changed output file to reflect changes  changed input file to reflect changes  corrected grammatical errors |
| 09\_withdrawal\_accbalancevalid | Test if the amount to withdraw must be higher or equal than the account balance | test case input file and output file missing prompts and information | none | corrected grammatical errors |
| 10\_withdrawal\_divby5 | Test if withdrawal amount is divisible by 5, and reject request otherwise | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | did not check condition | changed output file to reflect changes  changed input file to reflect changes  corrected grammatical errors |
| 11\_withdrawal\_SP\_transaction\_fee | Test if the withdrawal amount is balance + 5 cent in student plan. (If the balance is $400.00, withdraw $400 is rejected) | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | none | corrected grammatical errors |
| 12\_withdrawal\_NP\_transaction\_fee | Test if withdrawal amount is balance + 10 cent in non-student plan. (if the balance is $400, withdraw $400 is rejected) | test case input file and output file missing prompts and information  incorrect input  some output prompts missing | none | corrected grammatical errors |
| **transfer** | | | | |
| 01\_transfer\_prompt | Test if the session prompts for the account holder’s name when logged in as admin | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 02\_transfer\_originaccnum | Test if the session prompts for the account number that the money will be coming from | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 03\_transfer\_originaccnamenum | Test if the session only allow account holder name and origin account number entered in admin mode matched. If not matched, the login should be rejected | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check if account number and name match or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check if account number and name match or not |
| 04\_transfer\_destaccnum | Test if the session prompts for the account number that the money will be going to | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 05\_transfer\_destamt | Test if the session prompts for the amount to transfer | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 06\_transfer\_destamtformat | Test if session only allow the amount format in ####.## where # is 0-9 | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check for amount format | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  used method isCorrectFormat to check if the format is correct or not |
| 07\_transfer\_destvalidaccnum | Test if the session asks for a valid account number where the money will be going to | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check for a valid account number where the money will be going to | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check the number is valid or not |
| 08\_transfer\_originvalidaccnum | Test if the session asks for a valid account number where the money will be coming from | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check for a valid account number where the money will coming from | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check the number is valid or not  added code to find the account name of the money go to |
| 09\_transfer\_stdmaxtransfer | Test if the amount of funds in the current session that can be transferred does not exceed $1000.00 in standard mode | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check for the amount cannot exceed $1000.00 | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check if the amount exceed $1000.00 or not |
| 10\_transfer\_posttransfer | Test if the account balance of both accounts are at least $0.00 after transfer | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check if the balance are at least $0 or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check if the balance are at least $0 or not |
| 11\_transfer\_SP\_transaction\_fee | Test if the transfer amount is balance + 5 cent in student plan. (If the balance is $400.00, transfer $400 is rejected) | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check if the transfer amount is balance + 5 cent in the student plan | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check if the transfer amount is balance + 5 cent in the student plan |
| 12\_transfer\_NP\_transaction\_fee | Test if the transfer amount is balance + 10 cent in non-student plan. (if the balance is $400, transfer $400 is rejected) | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check if the transfer amount is balance + 10 cent in the non student plan | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check if the transfer amount is balance + 10 cent in the non student plan |
| 13\_transfer\_disabled | Test if the transfer goes through to a disabled account. | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check if transaction is success or not in a disabled account | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to check if transaction is success or not in a disabled account |
| **paybill** | | | | |
| 01\_paybill\_accname | Test if session asks for account holder’s name when logged in as admin | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check for valid name | changed output file to reflect correct output  changed code to make use of the new functions that verify the bank account holder  changed default valid name to false |
| 02\_paybill\_accnum | Test if session asks for account number | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check for valid number | changed output file to reflect correct output  changed code to make use of the new functions that verify the bank account number are valid  changed default valid number to false  changed the trans file to correct transaction |
| 03\_paybill\_accnameandnum | Test if the session only allow account holder name and account number entered in admin mode matched. If not matched, the login rejected | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check for match account name and number | changed input file to test the intention  changed output file to reflect correct output  changed code to make use of the new functions that verify the bank account number and account holder name are match  changed the trans file to correct transaction |
| 04\_paybill\_companyname | Test if session asks for company name to whom bill is being paid | test case output file and transaction file missing informations  incorrect input | None | changed output file to reflect correct output  changed input file to test the intention  changed the trans file to correct transaction |
| 05\_paybill\_companyamt | Test if session asks for amount to be paid to company | test case output file and transaction file missing informations  incorrect input | None | changed output file to reflect correct output  changed input file to test the intention  changed the trans file to correct transaction |
| 06\_paybill\_amtformat | Test if session only allow the amount format in ####.## where # is 0-9 | test case output file and transaction file missing informations  incorrect input | code did not check for the format | changed output file to reflect correct output  changed input file to test the intention  changed the trans file to correct transaction  create a new method isCorrectFormat() to check the format  include regex library to header file |
| 07\_paybill\_validaccnum | Test if account number **does not** matches customer’s name | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed input file to test the “not match” account number  changed the trans file to correct transaction |
| 08\_paybill\_validcompany | Test is bill is being paid to a valid company - EC, CQ or TV | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check the input company name equal to valid company | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added method to check the valid company |
| 09\_paybill\_stdmaxamt | Test if bill payment amount in current session does not exceed $2000.00 in standard mode | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check the amount is below $2000 | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added method to check the amout is below $2000 |
| 10\_paybill\_post | Test is the account balance is at least $0.00 after bill is paid | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check the account balance is at least $0.00 after bill is paid | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added method to check the amount is at least $0 after bill is paid |
| 11\_paybill\_SP\_transaction\_fee | Test if the paybill amount is balance + 5 cent in student plan. (If the balance is $400.00, pay $400 bill is rejected) | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check the account balance + $0.05 in student plan | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  create a new method enoughTransactionFee() to check the balance |
| 12\_paybill\_NP\_transacation\_fee | Test if paybill amount is balance + 10 cent in non-student plan. (if the balance is $400, pay $400 bill is rejected) | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check the account balance + $0.10 in non student plan | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  used the new method enoughTransactionFee() to check the balance |
| **deposit** | | | | |
| 01\_deposit\_accholdernameprompt | Test if session asks for account holder’s name when logged in as admin | test case output file and transaction file missing informations  incorrect input | wrong order of prompting | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  changed the order of asking information |
| 02\_deposit\_accnumprompt | Test if session asks for account number | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  print out a message amount the deposit account number entered  changed the trans file to correct transaction |
| 03\_deposit\_acc\_name\_num\_match | Test if the session only allow account holder name and account number entered in admin mode matched. If not matched, the login should be rejected | test case output file and transaction file missing informations  incorrect input  did not check for condition | the valid account number setted initialize as true  did not check if the account name and number match or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  changed the initial value of valid account number to false  added method to check for matching account name and number |
| 04\_deposit\_amt | Test if session asks for the amount to be deposited | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 05\_deposit\_amtformatvalid | Test if session only allow the amount format in ####.## where # is 0-9 | test case output file and transaction file missing informations  incorrect input  did not check for condition | valid account number setted initialize as true  did not check if the amount format in ####.## or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  changed the initial value of valid amount to false  added method to check for matching account name and number |
| 06\_deposit\_validacc | Test if the account is valid for the account holder that is logged in | tested in 03\_deposit\_acc\_name\_num\_match | None | None |
| 07\_deposit\_SP\_transaction\_fee | Test if the deposit amount had - the transaction fee 5 cent in student plan. (If the deposit is $400.00, the final deposit amount will be $399.95) | test case output file and transaction file missing informations  incorrect input  did not check for condition | valid account number setted initialize as true  did not check if the amount format in ####.## or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  create a new method minusTransactionFee() to minus the transaction fee  included <iomanip> to convert float to string with precision  added method to make sure the amount after minus the transaction fee is created than 0 |
| 08\_deposit\_NP\_transaction\_fee | Test if deposit amount had - the transaction fee 10 cent in non-student plan. (if the deposit is $400, the final deposit amount will be $399.90) | test case output file and transaction file missing informations  incorrect input  did not check for condition | valid account number setted initialize as true  did not check if the amount format in ####.## or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 09\_deposit\_same\_day\_usage | Test if the funds are not available for use in same day | Need to test with withdrawal | None | None |
| 10\_deposit\_accepted\_amt | Test if deposit amount is divisible by 5, and reject request otherwise | test case output file and transaction file missing informations  incorrect input  did not check for condition | did not check if the amount is divisible or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  create a new method minusTransactionFee() to minus the transaction fee |
| **create** | | | | |
| 01\_create\_valid\_new\_acc\_name | Test if session asks for new account holder name, and truncates name if characters exceed 20 | test case output file and transaction file missing informations  incorrect input  did not check condition | did not check if name exceed 20 characters | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  create a new method fixName() to truncates name if exceed 20 characters and fill in space if under 20 characters |
| 02\_create\_initbal | Test if session asks for initial balance of account | test case output file and transaction file missing informations  incorrect input  did not ask for information | did not ask for initial balance | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  add code to ask for initial balance and store it |
| 03\_create\_record\_transaction\_file | Test if session successfully records new account information to bank account transaction file | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 04\_create\_test\_admin\_mode | Test if creation of account only accepted while logged in as admin | test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 05\_create\_unique\_account\_num | Test if new account number does not already exist in Bank System database, and reject if exists | test case output file and transaction file missing informations  incorrect input  did not create information | ask for an account number instead of create a random number | changed input file to test the intention  there should be any cases that account number is already exist  changed output file to reflect correct output  changed the trans file to correct transaction  added code to create a random account number that is unique |
| 06\_create\_max\_acc\_bal | Test if creation of account is rejected if account balance is over $99999.99 | test case output file and transaction file missing informations  incorrect input  did not check condition | did not check if the initial amount is over $99999.99 or not | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added method to check if the amount is over $99999.99 or not  added method to check if the entered amount in this #####.## |
| 07\_create\_end\_session | Test if account logged out after account creation and session restricts further transactions | test case output file and transaction file missing informations  incorrect input |  | changed input file to test the intention (added login to newly create account after logout)  changed output file to reflect correct output  changed the trans file to correct transaction |
| 08\_create\_truncate\_long\_acc\_name | Test if account name is truncated if it is greater than 20 characters | already fixed in 01\_create\_valid\_new\_acc\_name  test case output file and transaction file missing informations  incorrect input | None | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction |
| 09\_create\_default\_acc\_type | Test if account type is set to non-student plan as default | test case output file and transaction file missing informations  incorrect input  did not set information | did not set account type to non-student plan | changed input file to test the intention  changed output file to reflect correct output  changed the trans file to correct transaction  added code to set the account type to non-student |
| **delete** | | | | |
| 01\_delete\_admin\_mode | Test if the delete transaction is accepted when logged into admin mode and prompt for account name and number | input file, output file missing prompts and inputs | did not search bank accounts file for account name and number, to verify if it exists | changed input file to reflect changes  changed output file to reflect changes  added method to main code to search bank accounts file  fixed grammatical errors |
| 02\_delete\_standard\_mode | Test if the delete transaction is rejected when logged into standard mode and doesn’t prompt for account name and number | input file, output file missing prompts and inputs | none | fixed grammatical errors |
| 03\_delete\_valid | Test if the delete transaction is accepted when entered a valid holder’s name in admin mode with valid account name and number and saved to bank account transaction file | input file, output file missing prompts and inputs  TransactionProcessing.cpp missing code | did not search bank accounts file for account name and number, to verify if it exists | changed input file to reflect changes  changed output file to reflect changes  added method to main code to search bank accounts file  fixed grammatical errors |
| 04\_delete\_invalid\_name | Test if the delete transaction is rejected when entered an invalid holder’s name in admin mode | input file, output file missing prompts and inputs | did not search bank accounts file for account name and number, to verify if it exists | changed input file to reflect changes  changed output file to reflect changes  added method to main code to search bank accounts file  fixed grammatical errors |
| 05\_delete\_invalid\_number | Test if the delete transaction is rejected when entered an invalid account number in admin mode | input file, output file missing prompts and inputs | did not search bank accounts file for account name and number, to verify if it exists | changed input file to reflect changes  changed output file to reflect changes  added method to main code to search bank accounts file  fixed grammatical errors |
| 06\_delete\_further\_trans | Test if further transactions are not accepted on a deleted account | input file, output file missing prompts and inputs | none | fixed grammatical errors |
| 07\_delete\_disable | Test if a disabled account can be deleted in admin mode and saved to bank account transaction file | input file, output file missing prompts and inputs | none | fixed grammatical errors |
| **disable** | | | | |
| 01\_disable\_admin\_mode | Test if the disable is accepted when logged into admin mode and prompt for account name and number | input file, code had wrong if condition statements | changed code to make use of new functions that verify if bank account holder and bank account number are valid  changed code when checking the condition of the account, success and failure were reversed  changed code to correct grammar and keep variable names consistent | changed code and input file format |
| 02\_disable\_standard\_mode | Test if the disable is rejected when logged into standard mode and doesn’t prompt for account name and number | input file, output file | none | grammatical errors corrected in input file |
| 03\_disable\_status\_changed | Test if the status changed from active to disabled in admin mode with valid account name and number and saved to bank account transaction file | input file, output file | none | grammatical errors corrected in input file |
| 04\_disable\_invalid\_name | Test if the disable transaction is rejected when entered an invalid holder’s name in admin mode | input file, output file | none | grammatical errors corrected in input file |
| 05\_disable\_invalid\_number | Test if the disable transaction is rejected when entered an invalid account number in admin mode | input file, output file | none | grammatical errors corrected in input file |
| 06\_disble\_further\_trans | Test if further transactions are not accepted on a disabled account | input file, output file | none | grammatical errors corrected in input file |
| 07\_disable\_disabled\_account | Test if the disable transaction is rejected when entered an disabled account number in admin mode | input file, output file | none | grammatical errors corrected in input file |
| 08\_disable\_recent\_disabled\_account | Test if further transaction are not accepted on a recent disabled account in same session. | input file, output file | none | grammatical errors corrected in input file |
| **changeplan** | | | | |
| 01\_changeplan\_admin\_mode | Test if the changeplan transaction is accepted when logged into admin mode and prompt for account name and number | input file, output file and transaction file | code changed to produce correct transaction file, changed prompts in code to match expected outputs | changed input file to provide correct input to the system |
| 02\_changeplan\_standard\_mode | Test if the changeplan transaction is rejected when logged into standard mode and doesn’t prompt for account name and number | input file, output file and transaction file | none | changed input file to provide correct input to the system |
| 03\_changeplan\_SP\_to\_NP | Test if session is successfully able to change plan from SP to SN and saved to bank account transaction file in admin mode | input file, output file | added if statement to detect non student and student account numbers | changed output file to reflect correct output |
| 04\_changeplan\_NP\_to\_SP | Test if session is successfully able to change plan from NP to SP and saved to bank account transaction file admin mode | input file, output file, transaction file | none | changed input and output file to reflect correct output |
| 05\_changeplan\_invalid\_name | Test if the changeplan transaction is rejected when entered an invalid holder’s name in admin mode | input file, code | implemented 2 new functions to search the bank accounts file to verify name | changed input file to reflect capitalization of input user |
| 06\_changeplan\_invalid\_number | Test if the changeplan transaction is rejected when entered an invalid account number in admin mode | input file | changed spacing of concatenating strings and variables | changed input file to reflect correct formatting |
| **enable** | | | | |
| 01\_enable\_admin\_mode | Test if enabling of account accepted while logged in as admin and prompt for account name and number | input file, code had wrong if condition statements | changed code to make use of the new functions that verify the bank account holder and bank account number are valid  changed code when checking the condition of the account, success and failure were reversed  changed code to correct grammar | changed code and input file format |
| 02\_enable\_standard\_mode | Test if enabling of account rejected while logged in as standard user and doesn’t prompt for account name and number | input file | none | input file changed with capital account holder names |
| 03\_enable\_status\_changed | Test if session successfully changes account from disabled (D) to active (A) with valid account name and number and saved to bank account transaction file | input file | none | input file changed with capital account holder names |
| 04\_enable\_invalid\_name | Test if the enable transaction is rejected when entered an invalid holder’s name in admin mode | input file | changed to reflect correct spacing when concatenating strings | input file changed with capital account holder names |
| 05\_enable\_invalid\_number | Test if the enable transaction is rejected when entered an invalid account number in admin mode | input file | changed to reflect correct spacing when concatenating strings | input file changed with capital account holder names |
| 06\_enable\_enabled\_account | Test if the enable transaction is rejected when enable an enabled account in admin mode. | input file |  | input file changed with capital account holder names |