**Test Cases**

**Akira Aida – 100526064**

**Kathryn McKay – 100524201**

**Alexander Wheadon – 100514985**

**Change Plan Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Files** |
| Verify account holder's name and account number correspond with each other (Incorrect name, correct number). | Admin enters an incorrect account holder's name but correct account number. | The admin is notified the account is not valid. | chng000.in  chng000.out  chng000.trans |
| Verify account holder's name and account number correspond with each other (Correct name, incorrect number). | Admin enters a correct account holder's name but incorrect account number. | The admin is notified the account is not valid. | chng001.in  chng001.out  chng001.trans |
| Verify account holder's name and account number correspond with each other (Correct name, correct number but not matching). | Admin enters a correct account holder and correct account number but they do not correspond to each other. | The admin is notified the account is not valid. | chng002.in  chng002.out  chng002.trans |
| Verify admin can change an account from SP (student) to NP (non-student). | Admin enters a correct account holder, correct account number and they do correspond to each other. | The information is saved to the bank account transaction file saying that the account payment plan is now NP (non-student). | chng003.in  chng003.out  chng003.trans |
| Verify admin can change an account from NP (non-student) to SP (student). | Admin enters a correct account holder, correct account number and they do correspond to each other. | The information is saved to the bank account transaction file saying that the account payment plan is now SP (student). | chng004.in  chng004.out  chng004.trans |
| Verify normal users can not access the changeplan feature. | Normal user attempts to use the changeplan feature. | The command is rejected. | chng005.in  chng005.out  chng005.trans |

**Disable Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Files** |
| Verify account holder's name and account number correspond with each other (Incorrect name, correct number). | Admin enters an incorrect account holder's name but correct account number. | The admin is notified the account is not valid. | dsbl000.in  dsbl000.out  dsbl000.trans |
| Verify account holder's name and account number correspond with each other (Correct name, incorrect number). | Admin enters a correct account holder's name but incorrect account number. | The admin is notified the account is not valid. | dsbl001.in  dsbl001.out  dsbl001.trans |
| Verify account holder's name and account number correspond with each other (Correct name, correct number but not matching). | Admin enters a correct account holder and correct account number but they do not correspond to each other. | The admin is notified the account is not valid. | dsbl002.in  dsbl002.out  dsbl002.trans |
| Verify normal users can not access the disable feature. | Normal user attempts to use the disable feature. | The command is rejected. | dsbl003.in  dsbl003.out  dsbl003.trans |
| Verify admin can change an account from A (active) to D (disabled). | Admin enters a correct account holder, correct account number and they do correspond to each other. | The information is saved to the bank account transaction file saying that the account is now disabled. | dsbl004.in  dsbl004.out  dsbl004.trans |
| Verify user withdraw transaction can no longer be done to a disabled account - user. | Enter the disabled account's correct account holder's name, account number, and an amount of money in the withdraw feature. | Command is denied because the account is disabled. | dsbl005.in  dsbl005.out  dsbl005.trans |
| Verify user deposit transaction can no longer be done to a disabled account - user. | Enter the disabled account's correct account holder's name, account number, and an amount of money in the deposit feature. | Command is denied because the account is disabled. | dsbl006.in  dsbl006.out  dsbl006.trans |
| Verify user paybill transaction can no longer be done to a disabled account - user. | Enter the disabled account's correct account holder's name, account number, company, and an amount of money to pay in the paybill feature. | Command is denied because the account is disabled. | dsbl007.in  dsbl007.out  dsbl007.trans |
| Verify delete transaction can still be done to a disabled account - user. | Enter the disabled account's correct account holder's name and account number in the delete feature. | Disabled account is deleted. | dsbl008.in  dsbl008.out  dsbl008.trans |
| Verify disable transaction can no longer be done to a disabled account - user. | Enter the disabled account's correct account holder's name and account number in the disable feature. | Command is denied because the account is disabled. | dsbl009.in  dsbl009.out  dsbl009.trans |
| Verify user transfer transaction can no longer be done with a disabled account (From an account to a disabled account) - user. | Enter the disabled account's correct account holder's name and account number, another account number and an amount of money in the transfer feature. | Command is denied because the account is disabled (disabled precedes valid account) - user. | dsbl010.in  dsbl010.out  dsbl010.trans |
| Verify user transfer transaction can no longer be done with a disabled account (From a disabled account to another account) - user. | Enter an account holder's name, account number, the disabled account number, and an amount of money in the transfer feature. | Command is denied because the account is disabled (disabled precedes valid account). | dsbl011.in  dsbl011.out  dsbl011.trans |
| Verify user withdraw transaction can no longer be done to a disabled account - admin. | Enter the disabled account's correct account holder's name, account number, and an amount of money in the withdraw feature. | Command is denied because the account is disabled. | dsbl012.in  dsbl012.out  dsbl012.trans |
| Verify user deposit transaction can no longer be done to a disabled account - admin. | Enter the disabled account's correct account holder's name, account number, and an amount of money in the deposit feature. | Command is denied because the account is disabled. | dsbl013.in  dsbl013.out  dsbl013.trans |
| Verify user paybill transaction can no longer be done to a disabled account - admin. | Enter the disabled account's correct account holder's name, account number, company, and an amount of money to pay in the paybill feature. | Command is denied because the account is disabled. | dsbl014.in  dsbl014.out  dsbl014.trans |
| Verify user transfer transaction can no longer be done with a disabled account (From an account to a disabled account) - admin. | Enter the disabled account's correct account holder's name and account number, another account number and an amount of money in the transfer feature.  Test Input: | Command is denied because the account is disabled (disabled precedes valid account) - admin. | dsbl015.in  dsbl015.out  dsbl015.trans |
| Verify user transfer transaction can no longer be done with a disabled account (From a disabled account to another account) - admin. | Enter an account holder's name, account number, the disabled account number, and an amount of money in the transfer feature. | Command is denied because the account is disabled (disabled precedes valid account). | dsbl016.in  dsbl016.out  dsbl016.trans |

**Delete Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Files** |
| Verify account holder's name and account number correspond with each other (Incorrect name, correct number). | Admin enters an incorrect account holder's name but correct account number. | The admin is notified the account is not valid. | delt000.in  delt000.out  delt000.trans |
| Verify account holder's name and account number correspond with each other (Correct name, incorrect number). | Admin enters a correct account holder's name but incorrect account number. | The admin is notified the account is not valid. | delt001.in  delt001.out  delt001.trans |
| Verify account holder's name and account number correspond with each other (Correct name, correct number but not matching). | Admin enters a correct account holder and correct account number but they do not correspond to each other. | The admin is notified the account is not valid. | delt002.in  delt002.out  delt002.trans |
| Verify normal users can not access the disable feature. | Normal user attempts to use the delete feature. | The command is rejected. | delt003.in  delt003.out  delt003.trans |
| Verify admin can delete an account. | Admin enters a correct account holder, correct account number and they do correspond to each other. | The information is saved to the bank account transaction file saying that the account is now deleted. | delt004.in  delt004.out  delt004.trans |
| Verify user can not do transactions to a deleted account. | Enter the deleted account's correct account holder's name, account number, and an amount of money in the withdraw feature. | Command is denied because the account is deleted. | delt005.in  delt005.out  delt005.trans |
| Verify admin can not do transactions to a deleted account. | Enter the deleted account's correct account holder's name, account number, and an amount of money in the withdraw feature. | Command is denied because the account is deleted. | delt006.in  delt006.out  delt006.trans |

**Create Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Files** |
| Verify account creation (with valid input). | Admin enters an account holder's name that is under 20 characters and initial balance less than $99999.99 into the create feature. | Information is saved to the bank account transaction file with this new information and a uniquely generated account number. | crte000.in  crte000.out  crte000.trans |
| Verify account creation (with invalid name input - to long). | Admin enters an account holder's name that is greater than 20 characters and initial balance less than $99999.99 into the create feature. | Transaction is allowed but the name is truncated to 20 characters. | crte001.in  crte001.out  crte001.trans |
| Verify account creation (with invalid name input - used already). | Admin enters an account holder's name that is already used. | Transaction is not allowed because the name is already taken. | crte002.in  crte002.out  crte002.trans |
| Verify account creation (with invalid inital balance input). | Admin enters an account holder's name that is less than or equal to 20 characters and initial balance greater than $99999.99 into the create feature. | Transaction is denied and the admin is notified that the initial balance is to much. | crte003.in  crte003.out  crte003.trans |
| Verify account creation (with both inputs being invalid). | Admin enters an account holder's name that is greater than 20 characters and initial balance greater than $99999.99 into the create feature. | Transaction is denied and the admin is notified that the initial balance is to much. | crte004.in  crte004.out  crte004.trans |
| Verify normal users can not access the create feature. | Normal user attempts to use the create feature. | The command is rejected. | crte005.in  crte005.out  crte005.trans |
| Verify account is not available in the same session. | A transaction is done with the newly created account holder's name and account number. | Transaction is denied because the file hasn't been sent to the back end yet for processing (same session) | crte006.in  crte006.out  crte006.trans |

**Login Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Files** |
| Verify that user cannot login if another login session has been created. | User attempts to login when another person is logged in. | The user is notified that they must first logout before they attempt another login. | logn000.in  logn000.out  logn000.trans |
| Verify that username exists when the user logs in. | User attempts to login when don’t currently have an account. | The user is notified that they do not have a valid account and must speak to an admin before being able to login in. | logn001.in  logn001.out  logn001.trans |
| Verify that users and admins have different privileges | Use admin command as user and as admin. | User is unable to perform admin operation, and admin is able to perform operation | logn002.in  logn002.out  logn002.trans |
| Verify login must be perform before other commands can be used | Use commands before user login. | User is unable to perform operation since they are not logged in | logn003.in  logn003.out  logn003.trans |

**Logout Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Files** |
| Verify logout is successful. | User is unable to use operations after logout | User is unable to perform operations after logout | logt000.in  logt000.out  logt000.trans |

**Transfer Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Files** |
| Verify account number for transfer. | Use incorrect account number when transferring. | User is unable to transfer from a different account | tran000.in  tran000.out  tran000.trans |
| Verify user cannot transfer to the same account | Transfer to the same account. | User is unable to transfer to the same account | tran001.in  tran001.out  tran001.trans |
| Verify user can transfer at the transfer zero dollars | Attempt to transfer zero dollars. | User is unable to transfer 0 dollars. | tran002.in  tran002.out  tran002.trans |
| Verify user transfers incurs fees where the admin does not | Attempt to transfer 100 in student user account and then 100 in admin account. | Account reduces 100.50 because of fees when user transfers, account reduces by 100 when admin transfers | tran003.in  tran003.out  tran003.trans |
| Verify max transfer | Test 1000 dollar transfer | Account reduces 1000.50 because of fees when user transfers | tran004.in  tran004.out  tran004.trans |
| Verify working transfer limit | Test 1005 dollar transfer | Account does not reduce any money. | tran005.in  tran005.out  tran005.trans |
| Verify summing of transfers | Test a 505 dollar transfer after 550 dollar transfer | Account reduces 500.50 because of fees when user transfers | tran006.in  tran006.out  tran006.trans |
| Verify amount only allow multiples of five | Test invalid inputs. | Account denies all invalid inputs for transfer. | tran007.in  tran007.out  tran007.trans |