Back End Testing Methodology and Report for Withdrawal Transactions

For the back-end testing on assignment 5 we are to practice partial white box testing of the Back-Office portion of Quinterac we programmed in Assignment #4.

For the Testing of our back Office we implemented white box testing for the new account creation transactions as well as the withdrawal transactions. For these two sections of the program we are to use two separate white box testing methodologies in order to test out program. For the new account transactions, we chose to perform the white box testing methodology of Path coverage. For the withdrawal transactions, we chose to perform the white box testing methodology of decision coverage. Both of our testing procedures only covered the section of code needed to perform these actions.

Down below outlines the breakdown of how we derived our test cases we have to write in order to perform the tests for the withdrawal transaction. For our test cases we are only testing the method/ section of code that performs the actual transaction and verifies that these sections of code are actually functioning as intended.

The two main files we are testing for this section of our code are BackendObj (in specific the processTrans() as well as processWDR() methods inside this file) and the AccountObj file (in specific the validate() method in this file).

Withdrawal Transaction (Decision Coverage)

- Line that holds WDR at the start
    - accBal value of the account is < 0
        - don't need to check as it throws an exception if less than 0
    - accBal of the account is > 0
        - deleted is true and accBal is not 0
        - deleted is true and accBal is 0
        - deleted is false and accBal is not 0
        - deleted is false and accBal is 0
- Line that Holds DOG at the start
    - Goes Nowhere (throws exception) (Test already covered in the New Account transaction tests)

Since we are performing decision coverage on the withdrawal transaction method/section of our Back-Office code we have 3 decision statements that we must create test cases for in order to make sure each state of the if statement is provoked in testing. The first decision statement that needs to be provoked is the case statement that reads the transaction code and select the correct case in order to perform the correct action required. We are not worrying about if the test case picks up on a non recognized case as we have already tested that in the path coverage of the New account creation test cases. The second decision statement is an if statement to check if the account we are wanting to withdrawal from has an account balance that is greater than 0. For this if statement we need to perform

2 tests one where an account has a balance less than 0 and another test for where an account has a balance greater than 0. Finally, we have our last if statement that this section of code will run through and since we have two variables that this if statement is checking to be true we will need to perform 4 tests for this one if statement.

All of our testing files for the back Office can be found at the following directory path in our repository on GitHub:

\Quinterac\Assignment_4_5\src\test\resources\R4

Chart of Test Cases for Withdrawal Transaction

Requirement	Test Num	Test Name	Purpose	Input Files	Output	Output Files
Test the Functionality of the **withdrawal** transaction Performed **DECISION** coverage on this portion of our Back-Office Program	R2T1	withdrawT1	Withdraw from account when account balance is < 0	BackR2T1Master.txt TransactionSummarFiles/ backR2T1/ TransactionSummary.txt	Balance cannot be negative. Error during processWDR	backR2T1Expected.txt
	R2T2	withdrawT2	Withdraw from an account normally	BackR2T1Master.txt TransactionSummarFiles/ backR2T2/ TransactionSummary.txt	Done	backR2T2Expected.txt
	R2T3	withdrawT3	Trying to delete and account when account balance is > 0	BackR2T1Master.txt TransactionSummarFiles/ backR2T3/ TransactionSummary.txt	Error during processDEL	backR2T3Expected.txt
	R2T4	withdrawT4	Deleted account when account balance is 0	BackR2T1Master.txt TransactionSummarFiles/ backR2T4/ TransactionSummary.txt	Done	backR2T4Expected.txt
	R2T5	withdrawT5	Withdraw from account when balance is greater than 0	BackR2T1Master.txt TransactionSummarFiles/ backR2T5/ TransactionSummary.txt	Done	backR2T5Expected.txt
	R2T6	withdrawT6	Withdraw from account when balance is 0	BackR2T1Master.txt TransactionSummarFiles/ backR2T6/ TransactionSummary.txt	Error during processWDR	backR2T6Expected.txt

Test and Failure Report

Test Number	Issue	Resolution
backR2T3	[TEST CASE] Expected output did not match actual due to spelling error	[Solved] Fixed spelling error