

Test Report - CREATEACCT



|  |  |
| --- | --- |
| Name | Student Number |
| Muhammad Usman Majeed | 10086980 |
| Jessica Nahulan | 10029341 |
| Johan Cornelissen | 10098176 |

Contents

[Test method of choice: Decision Coverage Testing 3](#_Toc498880738)

[Source: 3](#_Toc498880739)

[Table of test cases and their inputs: 4](#_Toc498880740)

[Test Case Analysis: 5](#_Toc498880741)

[test\_1 5](#_Toc498880742)

[test\_2 5](#_Toc498880743)

[test\_3 6](#_Toc498880744)

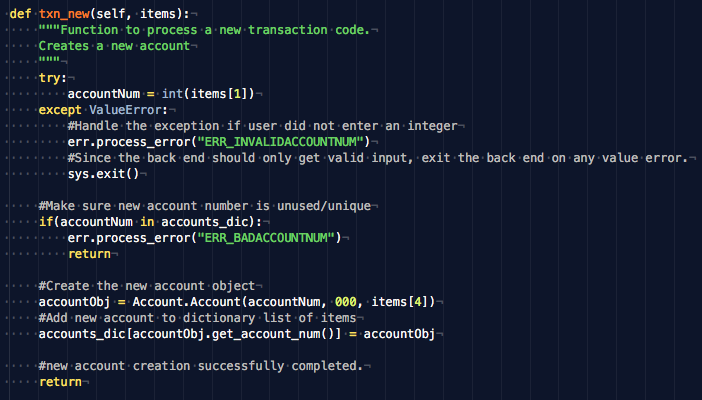
[test\_4 6](#_Toc498880745)

[Test Report: 7](#_Toc498880746)

[Team Contribution: 9](#_Toc498880747)

# Test method of choice: White Box Input Partitioning Testing

# Source:



# Input Partitions:

The method that processes the createacct transaction takes in a list of space separated items that make up the transaction command. Where one createacct command is in the following format:

“NEW <accountToBeCreated> 000 0000000 <accountName>”

\*Assumption was made that BOTH the account number and account name need to remain unique.

The input can be partitioned as follows:

|  |  |  |
| --- | --- | --- |
| Partition | Account Number to be created | Account Name to be created |
| P1 | Invalid Account number | Valid Account Name |
| P2 | Invalid Account number | Invalid Account Name |
| P3 | Valid Account Number | Invalid Account Name |
| P4 | Valid Account Number | Valid Account Name |

# Table of test cases and their inputs:

The following provides a table identifying the test cases necessary to exercise white box input partitioning testing on the create account transaction process for the back office. The decisions below relate to the source image portrayed above and the identified input partitions seen above.

In addition to identifying the various test cases, the table provides the inputs used to exercise each of the various test cases and a summary of the intended behavior.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Decision | Input 1 | Input 2 | Test Case Summary |
| test\_1 | 1 : True | accounts\_dic {1234567,7777777}  *accounts\_dic is populated by input master accounts file* | accountNum:  1111111  *accountNum is provided in transaction summary input* | accountNum to withdraw from is not a valid account provided by master accounts file.  Resulting in decision 1 being “True” |
| test\_2 | 1 : False | accounts\_dic: {1234567,7777777}  *accounts\_dic is populated by input master accounts file* | accountNum:  1234567  *accountNum is provided in transaction summary input* | accountNum to withdraw from is a valid account found in the master accounts file.  Resulting in decision 1 being “False” |
| test\_3 | 2 : True | accounts\_dic[1234567] .getAccountBalance() :  1234  *Balance for account number 1234567 provided by master accounts file* | withdrawAmount:  1235  *Withdraw amount provided in transaction summary input* | withdrawAmount for account number: 1234567 is “1235” which is more than existing balance “1234” resulting in negative balance.  Resulting in decision 2 being “True” |
| test\_4 | 2 : False | accounts\_dic[1234567] .getAccountBalance() :  1234  *balance for account number 1234567 provided by master accounts file* | withdrawAmount:  1000  *Withdraw amount provided in transaction summary input* | withdrawAmount for account number: 1234567 is “1000” which is less than existing balance “1234” resulting in valid withdraw amount.  Resulting in decision 2 being “False” |

# Test Case Analysis:

## test\_1

Relevant transaction summary file line:

**WDR 1111111 1000 0000000 \*\*\***

Master accounts file contents:

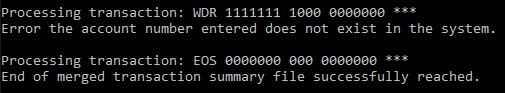
**1234567 1234 JESSICA**

**7777777 3210 USMAN**

Referring to the source image decision [1] , this test case intends to trigger this decision as **true** that the provided account num to be withdrawn from is not a valid account provided by the master accounts file.

The expected output is that the withdraw transaction will not take place as invalid data has been provided and the output master accounts file will still depict the original balance for the relevant account number (Balance of 1234 in input master accounts file remains 1234 in output master accounts file for account number: 1234567).

**Stdoutput**:



## test\_2

Relevant transaction summary file line:

**WDR 1234567 1000 0000000 \*\*\***

Master accounts file contents:

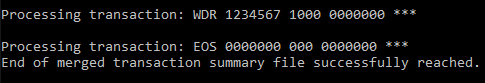
**1234567 1234 JESSICA**

**7777777 3210 USMAN**

Referring to the source image decision [1] , this test case intends to trigger this decision as **false** that the provided account num to be withdrawn from is a valid account provided by the master accounts file.

The expected output is that the withdraw transaction will take place as the account number provided is valid and the output master accounts file will depict the updated balance for the relevant account number (Balance of 1234 in input master accounts file becomes 234 in output master accounts file for account number: 1234567 as 1000 was successfully withdrawn from the account).

**Stdoutput:**



## test\_3

Relevant transaction summary file line:

**WDR 1234567 1235 0000000 \*\*\***

Master accounts file contents:

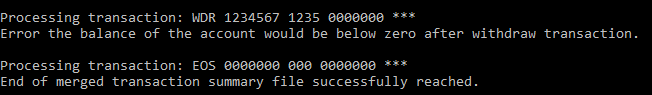
**1234567 1234 JESSICA**

**7777777 3210 USMAN**

Referring to the source image decision [2] , this test case intends to trigger this decision as **true** that the amount to be withdrawn is more than the existing balance in the relevant account (that if the withdraw were to take place, a negative balance would result).

The expected output is that the withdraw transaction will not take place as the amount to be withdrawn “1235”, specified by the transaction summary input, is more than the existing balance for the account, balance specified by the master accounts file. The expected output is that due to the potential negative balance, the withdraw transaction will not take place and the output master accounts file will still depict the original balance amount for account number 1234567 as was in the original input master accounts file.

**Stdoutput:**



## test\_4

Relevant transaction summary file line:

**WDR 1234567 1000 0000000 \*\*\***

Master accounts file contents:

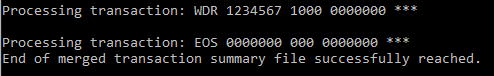
**1234567 1234 JESSICA**

**7777777 3210 USMAN**

Referring to the source image decision [2] , this test case intends to trigger this decision as **false** that the amount to be withdrawn is less than the existing balance in the relevant account.

The expected output is that the withdraw transaction will successfully take place as the amount to be withdrawn “1000”, specified by the transaction summary input, is less than the existing balance for the account, balance specified by the master accounts file. The expected output is that the withdraw transaction will successfully take place for account number 1234567 and the new master accounts fill depict a new balance of 1234 – 1000 = 234.

**Stdoutput:**



# Test Report:

#####################TESTING STATISTICS#####################

Tests Passed: 4

Tests Failed: 0

Tests Run: 4

###########################################################

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test# | Test Area | Test Name | Result | Acceptance Criteria | Error in output, Error in code, and fix applied | Re-Test details |
| 1 | Withdraw | test\_1 | Pass | When input transaction file for withdraw transaction includes an account number not present in the master accounts file, the transaction does not take place and no change is made to the account’s balance in the output master accounts file.  Std output should indicate error message for account number not found. | N/A | N/A |
| 2 | Withdraw | test\_2 | Pass | When input transaction file for withdraw transaction includes an account number present in the master accounts file, the transaction should successfully take place and the new updated balance should be present in the output master accounts file. | N/A | N/A |
| 3 | Withdraw | test\_3 | Fail | When input transaction file indicates an amount to be withdrawn that is larger than the balance present in the account, the transaction should not take place. No change should be made to the balance in the output master accounts file.  Std output should indicate error message for a bad/negative balance. | **Error in output**:  The output master accounts file reflected a changed balance amount when the transaction should not have occurred, and no change should have been made.  Error message for std output was not observed but should have been  **Error in code:**  **Line number: 142**  The withdraw amount was not being subtracted from the original balance, so the amount was always greater than 0 and the decision was always true.  **Fix applied:**  The withdraw amount was subtracted from the original balance before checking if this new amount is less than 0. | Performed manual unit test on several WITHDRAW transactions to ensure an being withdrawn that is greater than the existing balance results in an error and an incomplete/unsuccessful transaction.  Re-ran tests to ensure test failure was no longer observed. |
| 4 | Withdraw | test\_4 | Pass | When input transaction file indicates an amount to be withdrawn that is less than the balance present in the account, the transaction should successfully take place. The output master accounts file should reflect an updated balance with the withdraw amount subtracted from the original balance. | N/A | N/A |

# Team Contribution:

|  |  |  |
| --- | --- | --- |
| Name | Hours Spent | Tasks |
| Muhammad Usman Majeed |  |  |
| Jessica Nahulan |  |  |
| Johan Cornelissen |  |  |