Test Report Document
for the
Back Office Unit Testing of Simple Interactive Banking System
(SimBank)

Submitted By:

Tech Testers

Zainab Bello - 10147946

Wanyu Zhang - 10141510

Jia Yue Sun (Selena) - 10152968

TABLE OF CONTENTS

1. ACCOUNT CREATION TRANSACTIONS

Systematic white box unit testing -- Statement coverage testing

1.1. Methods to be tested:

```
// This method adds a newly created account to the arrayList with the contents
// the master accounts file
// Parameters: line - the transaction summary line to be added to the master accounts file
public void createMaster(String[] line) {
   1 String[] masterline = new String[3];
   2 acc = line[1];
   3 int mastIndex = getIndex(acc);
  10 if (mastIndex == -1){
  11     masterline[0] = line[1];
  12     masterline[1] = "000";
  13     masterline[2] = line[4];
         //add this string into the master accounts
  14     addMaster(masterline);
  16 } else {
         //log error and ignore the create transaction
  17     System.out.println("Create Invalid - Account already exists!");
     }
}

// This method gets the index of the specified account number from the arrayList
// containing the master accounts file
// Parameter: acctNum - the account number being searched for
// Returns: the index of the specified account number if it exists, if not exit the program.
public int getIndex(String acctNum){
   4 int len = masterFile.size();
   5 for (int i = 0; i < len; i++){
   6     String[] trans = masterFile.get(i);
   7     if (trans[0].equals(acctNum))
   8         return i;
     }
   9 return -1;
}

// This method adds a new line to the masterFile arrayList
// Parameters: trans - a newline to be added to the arrayList
protected void addMaster(String[] trans){
  15 masterFile.add(trans);
}// end addMaster
```

1.2. Test Cases and Test Inputs analysis

<u>Note</u>: For this table, the Master Accounts file being used is : 11111111 5000 John

Decision	Input -String[] line	Test	Input -String[] line	Output
1	CR 11111111 00000000 000 Joe	**T1**	CR 11111111 00000000 000 Joe	An error message is logged on the console and the transaction is ignored since the account number already exists

2	CR 11111111 00000000 000 Joe			
3	CR 11111111 00000000 000 Joe			
4	CR 11111111 00000000 000 Joe			
5	CR 11111111 00000000 000 Joe			
6	CR 11111111 00000000 000 Joe			
7	CR 11111111 00000000 000 Joe			
8	CR 11111111 00000000 000 Joe			
9	CR 12345678 00000000 000 Fred	**T2**	CR 12345678 00000000 000 Fred	The internal master accounts file gets updated with the newly created account
10	CR 12345678 00000000 000 Fred			
11	CR 12345678 00000000 000 Fred			
12	CR 12345678 00000000 000 Fred			
13	CR 12345678 00000000 000 Fred			
14	CR 12345678 00000000 000 Fred			
15	CR 12345678 00000000 000 Fred			
16	CR 11111111 00000000 000 Joe	**T1**	CR 11111111 00000000 000 Joe	An error message is logged on the console and the transaction is ignored since the account number already exists

17	CR 11111111 00000000 000 Joe			

1.3. Report on the results of the test runs on the Back End

After the first run of each test on the back end, the first test case passed, while the second test case failed. The failure of the second test case was due to a syntax error in the program. After fixing this syntax error, the results of the test runs were as expected. The failure report is as shown below:

ACCOUNT CREATION TRANSACTION						
Test Case Name	Test Case Description	Nature of the failure	Error in the Code	Actions taken to Fix Error	Before Modifications	After Modifications
CRtransactionT1	Checks that invalid create transaction is flagged and then logs an error message to the console	Syntax error in the code. The program was calling a variable that had not been initialized	The program should log an error message to the console and ignore the transaction. However due to the syntax error in the code, the program was not built properly	Changed the variable name of the arrayList being used in the method from "arrays" to "masterList"	[11111111, 5000, John] [11111111, 000, Joe]	Create Invalid - Account already exists! [11111111, 5000, John]

2. WITHDRAW TRANSACTIONS

Systematic white box unit testing -- Basic block coverage

2.1. Methods to be tested:

```
// This method checks if the account balance is positive after withdrawing a certain amount.
// If it is, it updates the master accounts arrayList
// Parameter: line - the transaction summary line stating the account number a certain amount is to be withdrawn from
public int withdrawMaster(String[] line) {
    String[] masterline = new String[3];
    acc = line[2];
1   int mastIndex = getIndex(acc);
    if (mastIndex != -1) {
        int balance = Integer.parseInt(masterFile.get(mastIndex)[1]);
5       int amount = Integer.parseInt(line[3]);
        int newBalance = balance - amount;
        if (newBalance < 0) { // balance is negative - ignore transaction
6           System.out.println("Invalid Transaction - negative balance!");
            return -1;
        }else{
            masterline[0] = line[2];
7           if (newBalance == 0)
8               masterline[1] = "000";
            else if (newBalance < 10) {
9               String balStr = "00" + String.valueOf(newBalance);
                masterline[1] = balStr;
            } else if (newBalance < 100) {
10              String balStr = "0" + String.valueOf(newBalance);
                masterline[1] = balStr;
            }
            else
11              masterline[1] = String.valueOf(newBalance);
            masterline[2] = masterFile.get(mastIndex)[2];
            masterFile.remove(mastIndex);   //remove old entry
            addMaster(masterline);
            return 0;
        }
    }
    else {
        //account number does not exist
13      System.out.println("Withdraw Invalid - Account number does not exist!");
        return -1;
    }
}

// This method gets the index of the specified account number from the arrayList
// containing the master accounts file
// Parameter: acctNum - the account number being searched for
// Returns: the index of the specified account number if it exists, if not exit the program.
public int getIndex(String acctNum){
    int len = masterFile.size();
    for (int i = 0; i < len; i++){
2       String[] trans = masterFile.get(i);
3       if (trans[0].equals(acctNum))
4           return i;
    }
    return -1;
}

// This method adds a new line to the masterFile arrayList
// Parameters: trans - a newline to be added to the arrayList
protected void addMaster(String[] trans){
12      masterFile.add(trans);
}// end addMaster
```

2.2. Test Cases and Test Inputs analysis

<u>Note</u>: For this table, the Master Accounts file being used is : 11111111 5000 John

Blocks	Input - String[] line	Test	Input - String[] line	Output
1	WD 00000000 11111111 5001 ***	**T1**	WD 00000000 11111111 5001 ***	An error message is logged on the console and the transaction is ignored as it causes the account's balance to be negative
2	WD 00000000 11111111 5001 ***			
3	WD 00000000 11111111 5001 ***			
4	WD 00000000 11111111 5001 ***			
5	WD 00000000 11111111 5001 ***			
6	WD 00000000 11111111 5001 ***			
7	WD 00000000 11111111 5000 ***	**T2**	WD 00000000 11111111 5000 **	The internal master accounts file gets updated with the new account balance for the account specified
8	WD 00000000 11111111 5000 ***			
9	WD 00000000 11111111 4999 ***	**T3**	WD 00000000 11111111 4999 ***	The internal master accounts file gets updated with the new account balance for the account specified
10	WD 00000000 11111111 4990 ***	**T4**	WD 00000000 11111111 4990 ***	The internal master accounts file gets updated with the new account balance for the account specified
11	WD 00000000 11111111 4900 ***	**T5**	WD 00000000 11111111 4900 ***	The internal master accounts file gets updated with the new account balance for the account specified
12	WD 00000000 11111111 4900 ***			
13	WD 00000000 12341234 1000 ***	**T6**	WD 00000000 12341234 1000 ***	An error message is logged

				on the console and the transaction is ignored since the account number does not exist

2.3. Report on the results of the test runs on the Back End

After the first run of each test on the back end, all the test cases failed. This was also due to the same syntax error in the program that caused the second test case of the create transaction to fail. After fixing this syntax error, the results of the test runs were as expected.The failure report is as shown below:

WITHDRAW TRANSACTION

Test Case Name	Test Case Description	Nature of the failure	Error in the Code	Actions taken to Fix Error	Before Modifications	After Modifications
WDransactionT1	Checks that a negative balance after the withrawal transaction is flagged and then logs an error message to the console	Syntax error in the code. The program was calling a variable that had not been initialized	The program should log an error message to the console and ignore the transaction. However due to the syntax error in the code, the program was not built properly	Changed the variable name of the arrayList being used in the method from "arrays" to "masterList"	In Withdraw! Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1 at java.util.ArrayList.elementData(Unknown Source) at java.util.ArrayList.get(Unknown Source) at MergedTransactions.withdrawMaster(MergedTransactions.java:86) at MergedTransactions.mergeStart(MergedTransactions.java:181) at BackOffice.main(BackOffice.java:67)	Invalid Transaction - negative balance! [11111111, 5000, John]
WDransactionT2	Checks that a valid withdraw transaction is accepted	Syntax error in the code. The program was calling a variable that had not been initialized	The program should perform the withdraw transaction since it is valid.However due to the syntax error in the code, the program was not built properly	Changed the variable name of the arrayList being used in the method from "arrays" to "masterList"	In Withdraw! Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1 at java.util.ArrayList.elementData(Unknown Source) at java.util.ArrayList.get(Unknown Source) at MergedTransactions.withdrawMaster(MergedTransactions.java:86) at MergedTransactions.mergeStart(MergedTransactions.java:181) at BackOffice.main(BackOffice.java:67)	[11111111, 000, John]
WDransactionT3	Checks that a valid withdraw transaction is accepted	Syntax error in the code. The program was calling a variable that had not been initialized	The program should perform the withdraw transaction since it is valid.However due to the syntax error in the code, the program was not built properly	Changed the variable name of the arrayList being used in the method from "arrays" to "masterList"	In Withdraw! Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1 at java.util.ArrayList.elementData(Unknown Source) at java.util.ArrayList.get(Unknown Source) at MergedTransactions.withdrawMaster(MergedTransactions.java:86) at MergedTransactions.mergeStart(MergedTransactions.java:181) at BackOffice.main(BackOffice.java:67)	[11111111, 001, John]
WDransactionT4	Checks that a valid withdraw transaction is accepted	Syntax error in the code. The program was calling a variable that had not been initialized	The program should perform the withdraw transaction since it is valid.However due to the syntax error in the code, the program was not built properly	Changed the variable name of the arrayList being used in the method from "arrays" to "masterList"	In Withdraw! Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1 at java.util.ArrayList.elementData(Unknown Source) at java.util.ArrayList.get(Unknown Source) at MergedTransactions.withdrawMaster(MergedTransactions.java:86) at MergedTransactions.mergeStart(MergedTransactions.java:181) at BackOffice.main(BackOffice.java:67)	[11111111, 010, John]
WDransactionT5	Checks that a valid withdraw transaction is accepted	Syntax error in the code. The program was calling a variable that had not been initialized	The program should perform the withdraw transaction since it is valid.However due to the syntax error in the code, the program was not built properly	Changed the variable name of the arrayList being used in the method from "arrays" to "masterList"	In Withdraw! Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1 at java.util.ArrayList.elementData(Unknown Source) at java.util.ArrayList.get(Unknown Source) at MergedTransactions.withdrawMaster(MergedTransactions.java:86) at MergedTransactions.mergeStart(MergedTransactions.java:181) at BackOffice.main(BackOffice.java:67)	[11111111, 100, John]
WDransactionT6	Checks that the account number from which a certain amount is to be withdrawn from is flagged and then logs an error message to the console	Syntax error in the code. The program was calling a variable that had not been initialized	The program should log an error message to the console and ignore the transaction. However due to the syntax error in the code, the program was not built properly	Changed the variable name of the arrayList being used in the method from "arrays" to "masterList"	In Withdraw! Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1 at java.util.ArrayList.elementData(Unknown Source) at java.util.ArrayList.get(Unknown Source) at MergedTransactions.withdrawMaster(MergedTransactions.java:86) at MergedTransactions.mergeStart(MergedTransactions.java:181) at BackOffice.main(BackOffice.java:67)	Withdraw Invalid - Account number does not exist! [11111111, 5000, John]

3. TEST CASE FILES USED TO RUN THE BACK END

We used the following Master Accounts file in running the test cases on the Back End.

MasterAccounts.txt - 11111111 5000 John

The following are the Merged Transactions files used in running the test cases on the Back End of our program.

Merged Transactions Filename	Contents
CRtransactionT1.txt	CR 11111111 00000000 000 Joe
CRtransactionT2.txt	CR 12345678 00000000 000 Fred
WDtransactionT1.txt	WD 00000000 11111111 5001 ***
WDtransactionT2.txt	WD 00000000 11111111 5000 ***
WDtransactionT3.txt	WD 00000000 11111111 4999 ***
WDtransactionT4.txt	WD 00000000 11111111 4990 ***
WDtransactionT5.txt	WD 00000000 11111111 4900 ***
WDtransactionT6.txt	WD 00000000 12341234 1000 ***