**SSW 567 - Assignment 7 Part 2**

**Group 5**

Ed Chang

Harmony Sullivan

**Assignment Description**

This document contains the projects for weeks 7 and 8.  
I put them together so you can see what you need to do for week 8, which may influence what you do in week 7. Please write the results up in the standard manner we have been using.

AT THE END -- I want you to analyze which kind of tests found what kind of bugs -- was it FMEA? was it scenario? what was the relationship between test technique and finding the defects?????

Objectives:

The objectives of this assignment are for you to

• Create a test architecture

• Create and run scenarios, soap operas, and negative tests.

• See the result from capture/ recapture. .

Assignment:

Use the [ATM Simulation by Russell C. Bjork (Links to an external site.)](http://www.math-cs.gordon.edu/courses/cs211/ATMExample/" \t "_blank) on the web or YOU CAN TEST WHATEVER PROGRAM YOU WANT! You need to have access to the source code and to be able to compile and run it. Feel free to pick a game or a program you previously wrote.

**WEEK 8:**

•Run the test cases, document results, write appropriate bug reports etc.

•Create a negative test strategy. Include a FMEA analysis.

•Include a few of the interesting negative tests (NOT INPUT VALIDATION) in your test architecture.Run them, document results, write appropriate bug reports, etc.

. Now go back to the team that put the bugs into your software. How many did you find? How many other bugs did you find? Using the capture/recapture technique, how many bugs to project are still there?

**Results**

***Test Cases from Week 7 – testing Group 2 code***

At startup – there is 1 Error in SimCardDispenser.java: “The type SimCashDispenser” is already defined.

There are also 45 warnings.

Exception text immediately following each test case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Objective | Inputs | Expected Results | Actual Results |
| 1.1.1.1 | Perform balance inquiry of account # 1 | Choose Balance Inquiry option, then account # 1 (Checking) | $100 | Fail - Blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.inquiry(SimulatedBank.java:201)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:56)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 1.1.1.2 | Access account # 1 and withdrawal valid amount | Choose to do another transaction, Choose withdrawal, Choose account # 1 (checking), Choose option 1 ($20) | $20 in green text gets larger at bottom left of screen; Balance shows as $80 | Fail - After choosing account # 1, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.NullPointerException  at banking.Message.toString(Message.java:91)  at atm.physical.Log.logSend(Log.java:33)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:60)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 1.1.1.3 | Perform balance inquiry of account # 2 | Choose to do another transaction, Choose Balance Inquiry option, then account # 2 (savings) | $1000 | Fail - After choosing account # 2, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.inquiry(SimulatedBank.java:201)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:56)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 1.1.1.4 | Access account # 2 and withdrawal valid amount | Choose to do another transaction, Choose withdrawal, Choose account # 2 (Savings), Choose option 1 ($20) | $20 in green text gets larger at bottom left of screen; Balance shows as $980 | Fail - After choosing account # 2, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.NullPointerException  at banking.Message.toString(Message.java:91)  at atm.physical.Log.logSend(Log.java:33)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:60)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 1.1.2.1 | Access account # 1 and attempt to withdrawal invalid amount (greater amount than account # 1 balance) [NEG] | Choose to do another transaction, Choose withdrawal, Choose account # 1, Choose option # 4 ($100) | Message: Insufficient cash available | Fail - After choosing account # 1, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.NullPointerException  at banking.Message.toString(Message.java:91)  at atm.physical.Log.logSend(Log.java:33)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:60)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 1.1.2.2 | Access account # 2 and attempt to withdrawal invalid amount (greater amount than account # 2 balance) [NEG] | [Choose $20]Choose to do another transaction, Choose withdrawal, Choose account # 2 (Savings), Choose option # 5 ($200) (Repeat 4 more times) | [$20 in green text gets larger at bottom left of screen; Balance shows as $60]  Message: Insufficient cash available **\* Note that this seems to not be working in current application. Insufficient funds message occurs after 1st withdrawal** | Fail - After choosing account # 1, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.NullPointerException  at banking.Message.toString(Message.java:91)  at atm.physical.Log.logSend(Log.java:33)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:60)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| **CLOSE APPLICATION AND REOPEN** | | | | |
| 2.1.1.1 | Perform balance inquiry of account # 1 | Choose Balance Inquiry option, then account # 1 (Checking) | $100 | Fail - After choosing account # 1, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.inquiry(SimulatedBank.java:201)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:56)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 2.1.1.2 | Access account # 1 and deposit valid amount | Choose to do another transaction, Choose deposit (# 2), Choose account # 1 (checking), Enter $30.00, Click to insert envelope | Total Balance shows as $130, Available balance shows as $100 | Fail - After entering $30.00, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.initiateDeposit(SimulatedBank.java:114)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:44)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 2.1.1.3 | Perform balance inquiry of account # 2 | Choose to do another transaction, Choose Balance Inquiry option, then account # 2 (Savings) | $1000 | Fail - After choosing account # 2, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.inquiry(SimulatedBank.java:201)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:56)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 2.1.1.4 | Access account # 2 and deposit valid amount | Choose to do another transaction, Choose deposit (# 2), Choose account # 2 (savings), Enter $30.00, Click to insert envelope | Total Balance shows as $1030, Available balance shows as $1000 | Fail - After entering $30.00, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.initiateDeposit(SimulatedBank.java:114)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:44)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 2.1.2.1 | Access account # 1 and attempt to deposit invalid amount (0) [NEG] | Choose to do another transaction, Choose deposit (#2), Choose account # 1, Hit enter | A tone will sound, no change to the UI | Pass - Tone sounds, no change to UI |
| 2.1.2.2 | Access account # 1 and attempt to deposit exceeding maximum (what is maximum?) [NEG] | [Enter 1.00, hit enter, click to insert envelope]  Choose to do another transaction, Choose deposit (# 2), Choose account # 1, Enter 1000000000000.00 | [Total balance shows as $131, Available balance shows as $100]  Green screen is blank | Fail - After entering large amount, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.NumberFormatException: For input string: "10000000000000000000"  at java.lang.NumberFormatException.forInputString(Unknown Source)  at java.lang.Integer.parseInt(Unknown Source)  at java.lang.Integer.parseInt(Unknown Source)  at atm.physical.CustomerConsole.readAmount(CustomerConsole.java:110)  at atm.transaction.Deposit.getSpecificsFromCustomer(Deposit.java:46)  at atm.transaction.Transaction.performTransaction(Transaction.java:108)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| **CLOSE APPLICATION AND REOPEN** | | | | |
| 2.1.2.3 | Access account # 2 and attempt to deposit invalid amount (0) [NEG] | Choose deposit (#2), Choose account # 2 (savings), Hit enter | A tone will sound, no change to the UI | Pass - Tone sounds, no change to UI |
| 2.1.2.4 | Access account # 2 and attempt to deposit exceeding maximum (what is maximum?) [NEG] | [Enter 1.00, hit enter, click to insert envelope]  Choose to do another transaction, Choose deposit (# 2), Choose account # 2, Enter 1000000000000.00 | [Total balance shows as $1001, Available balance shows as $1000]  Green screen is blank | Fail - After entering large amount, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.NumberFormatException: For input string: "1000000000000"  at java.lang.NumberFormatException.forInputString(Unknown Source)  at java.lang.Integer.parseInt(Unknown Source)  at java.lang.Integer.parseInt(Unknown Source)  at atm.physical.CustomerConsole.readAmount(CustomerConsole.java:110)  at atm.transaction.Deposit.getSpecificsFromCustomer(Deposit.java:46)  at atm.transaction.Transaction.performTransaction(Transaction.java:108)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| **CLOSE APPLICATION AND REOPEN** | | | | |
| 3.1.1.1 | Perform balance inquiry for account # 1 | Choose Balance Inquiry option, then account # 1 (Checking) | $100 | Fail - After choosing account # 1, blank green screen, blank receipt screen. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.inquiry(SimulatedBank.java:201)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:56)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 3.1.1.2 | Perform balance inquiry for account # 2 | Choose to do another transaction, Choose Balance Inquiry, Choose account #2 (savings) | $1000 | Fail - After choosing account # 2, blank green screen, blank receipt screen. Forced to close application. [see above for error message] |
| 3.1.1.3 | Transfer valid amount from account # 1 to account # 2 | Choose to do another transaction, Choose Transfer, Choose account # 1 (checking), Choose account # 2 (savings), Enter 20.00, hit enter | Total Balance $1020, Available Balance $1020 | Pass - Total Bal: 100.20  Available: 100.20  (understandable that the amounts are different since the preceding test cases did not run correctly)  Green screen asks if I want another transaction |
| 3.1.1.4 | Perform balance inquiry for account # 1 (verify) | Choose to do another transaction, Choose Balance Inquiry, Choose checking (# 1) | $80.00 | Fail - Blank green screen, receipt screen same as 3.1.1.3. Forced to close application. |
| Exception in thread "Thread-2" java.lang.ArrayIndexOutOfBoundsException: -1  at simulation.SimulatedBank.inquiry(SimulatedBank.java:201)  at simulation.SimulatedBank.handleMessage(SimulatedBank.java:56)  at simulation.Simulation.sendMessage(Simulation.java:200)  at atm.physical.NetworkToBank.sendMessage(NetworkToBank.java:65)  at atm.transaction.Transaction.performTransaction(Transaction.java:122)  at atm.Session.performSession(Session.java:89)  at atm.ATM.run(ATM.java:124)  at java.lang.Thread.run(Unknown Source) | | | | |
| 3.1.1.5 | Transfer valid amount from account # 2 to account # 1 | Choose to do another transaction, Choose Transfer, Choose account #2 (savings), Choose account #1 (checking), Enter 20.00 | Total Balance shows as $100, Available balance shows as $100 | Pass - Total Bal: 1000.20  Available: 1000.20  (understandable that the amounts are different since the preceding test cases did not run correctly)  Green screen asks if I want another transaction |
| 3.1.1.6 | Perform balance inquiry for account # 2 (verify) | Choose to do another transaction, Choose Balance Inquiry, Choose savings(# 2) | $1000.00 | Fail - After choosing account # 2, blank green screen, blank receipt screen. Forced to close application. [see above for error message] |
| 3.1.2.1 | Transfer invalid amount (exceeding balance) from account # 1 to account # 2 [NEG] | Choose to do another transaction, Choose Transfer, Choose account #1 (checking), choose account #2 (savings), Enter 111.10, hit enter | Message: Insufficient available balance | Fail - (entered 10000.00)  Transfer amount stated on receipt is $100. Total Bal: $230.80, Available: $230.80 |
| 3.1.2.2 | Transfer invalid amount (exceeding balance) from account # 2 to account # 1 [NEG] | Choose to do another transaction, Choose Transfer, Choose account #2 (savings), choose account #1 (checking), Enter 1030.00, hit enter | Message: Insufficient available balance | Fail - (entered 10000.00)  Receipt states transfer amount is $100.00. Total Bal: $969.20, Available: $969.20 |
| 3.1.3.1 | Attempt to transfer from invalid account type (account type not on card) to valid account type (account type on card) [NEG] | Choose to do another transaction, Choose Transfer, Choose 3 (Money Market), Choose 1 (Checking), Enter 20.00 | Message: Invalid from account type | Fail - Message: Invalid to account type |
| 3.1.3.2 | Attempt to transfer to invalid account type (account type not on card) from valid account type (account type on card) (also valid amount) [NEG] | Choose to do another transaction, Choose Transfer, Choose 1 (Checking), Choose 3 (Money Market), Enter 20.00 | Message: Invalid to account type | Fail - Message: invalid from account type |

***Test Results with Group 1 Injected Bugs***

For the project with Group 1’s injected bugs:

A bug was discovered while trying to ‘insert card’ to initiate the transaction. This required our team to debug and fix this bug before we can proceed with our testing. During the debugging, we discovered that a bug was injected in ATM.java where the ‘state’ variable was never set to ‘SERVING\_CUSTOMER\_STATE’ even after the card has been inserted.

The test results below are shown after this bug has been remedied.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case ID | Test Objective | Inputs | Expected Results | Actual Results |
| 1.1.1.1 | Perform balance inquiry of account # 1 | Choose Balance Inquiry option, then account # 1 (Checking) | $100 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output |
| 1.1.1.2 | Access account # 1 and withdrawal valid amount | Choose to do another transaction, Choose withdrawal, Choose account # 1 (checking), Choose option 1 ($20) | $20 in green text gets larger at bottom left of screen; Balance shows as $80 | Failed – Insufficient cash available. No observable error / exception  **Note:** The failure caused by Test 1.1.1.1 prevented continuation of the test. Require restarting and proceeding to Test Case 1.1.1.2 directly. |
| 1.1.1.3 | Perform balance inquiry of account # 2 | Choose to do another transaction, Choose Balance Inquiry option, then account # 2 (savings) | $1000 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output |
| 1.1.1.4 | Access account # 2 and withdrawal valid amount | Choose to do another transaction, Choose withdrawal, Choose account # 2 (Savings), Choose option 1 ($20) | $20 in green text gets larger at bottom left of screen; Balance shows as $980 | Failed – Insufficient cash available. No observable error / exception.  **Note:** The failure caused by Test 1.1.1.3 prevented continuation of the test. Require restarting and proceeding to Test Case 1.1.1.2 directly |
| 1.1.2.1 | Access account # 1 and attempt to withdrawal invalid amount (greater amount than account # 1 balance) | Choose to do another transaction, Choose withdrawal, Choose account # 1, Choose option # 4 ($100) | Message: Insufficient cash available | Passed |
| 1.1.2.2 | Access account # 2 and attempt to withdrawal invalid amount (greater amount than account # 2 balance) | [Choose $20]Choose to do another transaction, Choose withdrawal, Choose account # 2 (Savings), Choose option # 5 ($200) (Repeat 4 more times) | [$20 in green text gets larger at bottom left of screen; Balance shows as $60]  Message: Insufficient cash available **\* Note that this seems to not be working in current application. Insufficient funds message occurs after 1st withdrawal** | Failed – Insufficient cash available message appears during the 1st withdrawn request. |
| **CLOSE APPLICATION AND REOPEN** | | | | |
| 2.1.1.1 | Perform balance inquiry of account # 1 | Choose Balance Inquiry option, then account # 1 (Checking) | $100 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output |
| 2.1.1.2 | Access account # 1 and deposit valid amount | Choose to do another transaction, Choose deposit (# 2), Choose account # 1 (checking), Enter $30.00, Click to insert envelope | Total Balance shows as $130, Available balance shows as $100 | Failed – Once ‘Deposit’ is chosen as the transaction, the display went blank and all buttons and selects are no longer usable.  **Note:** The failure caused by Test 2.1.1.1 prevented continuation of the test. Require restarting and proceeding to Test Case 2.1.1.2 directly |
| 2.1.1.3 | Perform balance inquiry of account # 2 | Choose to do another transaction, Choose Balance Inquiry option, then account # 2 (Savings) | $1000 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output  **Note:** The failure caused by Test 2.1.1.2 prevented continuation of the test. Require restarting and proceeding to Test Case 2.1.1.3 directly |
| 2.1.1.4 | Access account # 2 and deposit valid amount | Choose to do another transaction, Choose deposit (# 2), Choose account # 2 (savings), Enter $30.00, Click to insert envelope | Total Balance shows as $1030, Available balance shows as $1000 | Failed – once ‘Deposit’ is chosen as the transaction, the display went blank and all buttons and selects are no longer usable.  **Note:** The failure caused by Test 2.1.1.3 prevented continuation of the test. Require restarting and proceeding to Test Case 2.1.1.4 directly |
| 2.1.2.1 | Access account # 1 and attempt to deposit invalid amount (0) | Choose to do another transaction, Choose deposit (#2), Choose account # 1, Hit enter | A tone will sound, no change to the UI | Failed – once ‘Deposit’ is chosen as the transaction, the display went blank and all buttons and selects are no longer usable.  **Note:** The failure caused by Test 2.1.1.4 prevented continuation of the test. Require restarting and proceeding to Test Case 2.1.2.1 directly |
| 2.1.2.2 | Access account # 1 and attempt to deposit exceeding maximum (what is maximum?) | [Enter 1.00, hit enter, click to insert envelope]  Choose to do another transaction, Choose deposit (# 2), Choose account # 1, Enter 1000000000000.00 | [Total balance shows as $131, Available balance shows as $100]  Green screen is blank | Failed – once ‘Deposit’ is chosen as the transaction, the display went blank and all buttons and selects are no longer usable.  **Note:** The failure caused by Test 2.1.2.1 prevented continuation of the test. Require restarting and proceeding to Test Case 2.1.2.2 directly |
| **CLOSE APPLICATION AND REOPEN** | | | | |
| 2.1.2.3 | Access account # 2 and attempt to deposit invalid amount (0) | Choose deposit (#2), Choose account # 2 (savings), Hit enter | A tone will sound, no change to the UI | Failed – once ‘Deposit’ is chosen as the transaction, the display went blank and all buttons and selects are no longer usable. |
| 2.1.2.4 | Access account # 2 and attempt to deposit exceeding maximum (what is maximum?) | [Enter 1.00, hit enter, click to insert envelope]  Choose to do another transaction, Choose deposit (# 2), Choose account # 2, Enter 1000000000000.00 | [Total balance shows as $1001, Available balance shows as $1000]  Green screen is blank | Failed – once ‘Deposit’ is chosen as the transaction, the display went blank and all buttons and selects are no longer usable.  **Note:** The failure caused by Test 2.1.2.3 prevented continuation of the test. Require restarting and proceeding to Test Case 2.1.2.4 directly |
| **CLOSE APPLICATION AND REOPEN** | | | | |
| 3.1.1.1 | Perform balance inquiry for account # 1 | Choose Balance Inquiry option, then account # 1 (Checking) | $100 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output |
| 3.1.1.2 | Perform balance inquiry for account # 2 | Choose to do another transaction, Choose Balance Inquiry, Choose account #2 (savings) | $1000 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output  **Note:** The failure caused by Test 3.1.1.1 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.1.2 directly |
| 3.1.1.3 | Transfer valid amount from account # 1 to account # 2 | Choose to do another transaction, Choose Transfer, Choose account # 1 (checking), Choose account # 2 (savings), Enter 20.00, hit enter | Total Balance $1020, Available Balance $1020 | Failed – once the amount to be transferred (20.00) is entered, the display went blank and all buttons and selects are no longer usable  **Note:** The failure caused by Test 3.1.1.2 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.1.3 directly |
| 3.1.1.4 | Perform balance inquiry for account # 1 (verify) | Choose to do another transaction, Choose Balance Inquiry, Choose checking (# 1) | $80.00 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output  **Note:** The failure caused by Test 3.1.1.3 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.1.4 directly |
| 3.1.1.5 | Transfer valid amount from account # 2 to account # 1 | Choose to do another transaction, Choose Transfer, Choose account #2 (savings), Choose account #1 (checking), Enter 20.00 | Total Balance shows as $100, Available balance shows as $100 | Failed – once the amount to be transferred (20.00) is entered, the display went blank and all buttons and selects are no longer usable  **Note:** The failure caused by Test 3.1.1.4 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.1.5 directly |
| 3.1.1.6 | Perform balance inquiry for account # 2 (verify) | Choose to do another transaction, Choose Balance Inquiry, Choose savings(# 2) | $1000.00 | Failed – Balance was not shown. Null exception thrown in Transaction.java as part of console output  **Note:** The failure caused by Test 3.1.1.5 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.1.6 directly |
| 3.1.2.1 | Transfer invalid amount (exceeding balance) from account # 1 to account # 2 | Choose to do another transaction, Choose Transfer, Choose account #1 (checking), choose account #2 (savings), Enter 111.10, hit enter | Message: Insufficient available balance | Failed – once the amount to be transferred (20.00) is entered, the display went blank and all buttons and selects are no longer usable  **Note:** The failure caused by Test 3.1.1.6 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.2.1 directly |
| 3.1.2.2 | Transfer invalid amount (exceeding balance) from account # 2 to account # 1 | Choose to do another transaction, Choose Transfer, Choose account #2 (savings), choose account #1 (checking), Enter 1030.00, hit enter | Message: Insufficient available balance | Failed – once the amount to be transferred (1030.00) is entered, the display went blank and all buttons and selects are no longer usable  **Note:** The failure caused by Test 3.1.2.1 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.2.2 directly |
| 3.1.3.1 | Attempt to transfer from invalid account type (account type not on card) to valid account type (account type on card) | Choose to do another transaction, Choose Transfer, Choose 3 (Money Market), Choose 1 (Checking), Enter 20.00 | Message: Invalid from account type | Failed – once the amount to be transferred (20.00) is entered, the display went blank and all buttons and selects are no longer usable  **Note:** The failure caused by Test 3.1.2.2 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.3.1 directly |
| 3.1.3.2 | Attempt to transfer to invalid account type (account type not on card) from valid account type (account type on card) (also valid amount) | Choose to do another transaction, Choose Transfer, Choose 1 (Checking), Choose 3 (Money Market), Enter 20.00 | Message: Invalid to account type | Failed – once the amount to be transferred (20.00) is entered, the display went blank and all buttons and selects are no longer usable  **Note:** The failure caused by Test 3.1.3.1 prevented continuation of the test. Require restarting and proceeding to Test Case 3.1.3.2 directly |

***Negative Testing***

For negative testing, the following tests were proposed:

* Loss of network connectivity
* Concurrent withdrawn from the same banking account that would cause the balance to go below the minimum balance requirement if both transactions were to executed
* Select cancel while other transaction is being performed

Since the ‘Loss of network connectivity’ and ‘Concurrent withdrawn’ can’t be execute with the current system, we decided to execute ‘select cancel while performing other transaction’ as the negative test.

For the project with Group 1’s injected bug, the only time that the negative testing can be executed when we tried to perform a ‘Transfer’ transaction since all other transaction cause the system to failed already. When ‘Cancel’ is selected while executing transfer action caused the system to stop the transaction, i.e. the desired behavior. If ‘Cancel’ is selected again when the transaction has been stopped, then system ejects the Card inserted – once again, this is the desired behavior. Thus the Group 1 project behaves the way as we expected.

We found a weakness in Group 2 code when testing invalid transfers on card 1. The to/from message was backwards.

Added the following test cases for card 2 to see if it made any difference:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.1.3.1 [card 2] | Attempt to transfer from invalid account type (account type not on card) to valid account type (account type on card) [NEG] | Choose to do another transaction, Choose Transfer, Choose 2 (Savings), Choose 1 (Checking), Enter 20.00 | Message: Invalid from account type | Fail - Message: Invalid to account type |
| 3.1.3.2 [card 2] | Attempt to transfer to invalid account type (account type not on card) from valid account type (account type on card) (also valid amount) [NEG] | Choose to do another transaction, Choose Transfer, Choose 1 (Checking), Choose 2 (Savings), Enter 20.00 | Message: Invalid to account type | Fail - Message: Invalid from account type |

The same errors were found.

***FMEA Analysis:***

* Identify and list all components and their failure modes (how they can fail), considering all possible operating modes.
* For each failure mode, determine the possible effects (e.g., impact) on all other components as well as the effect on the overall system.
* Then calculate the probabilities and the seriousness of the results of the failure modes. (1 = very serious, 5 = inconvenient)

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **How it can fail** | **Possible effects** | **Probabilities / Seriousness of the results of the failure modes** |
| Green entry pane | Blank | User unable to see what is going on | 1 - System failure |
| not accept entry | No transaction can be performed | 1. System failure |
| White receipt pane | Blank | User unable to confirm transaction performed | 1. System failure |
| display incorrect amounts | Un-satisfied user caused by incorrect information | 3-User doesn’t trust the system and/or institutional reputation damage |
| Number pad | Not operate at all | No transaction can be performed | 1 - System failure |
| not operate correctly | Un-satisfied user caused by unable to perform transaction or incorrect transaction | 3 - User doesn’t trust the system and/or institutional reputation damage |
| ON/OFF button | Not operate at all | System not available or can’t perform operation as needed | 2 - Loss of usage of the system by the customer |
| not operate correctly | Operator can’t perform operation on the system | 3 - Operator can’t trust the system status |
| Show Log / Hide Log button | Does not show log | Transaction log not available | 3 - Unable to confirm transaction |
| Does not hide log | Transaction log not available | 3 - Unable to confirm transaction |
| Entry screen for $20 cash dispenser | Does not accept entered value and does not move on to ATM window | User can’t complete transaction | 3 - User doesn’t trust the system and/or institutional reputation damage |
| Insert Card Click Button | Does not allow user to enter card # when clicked | User can’t complete transaction | 3 - User doesn’t trust the system and/or institutional reputation damage |
| Withdrawal view of $ in green | Does not appear | User can’t complete transaction | 3 - User doesn’t trust the system and/or institutional reputation damage |
| Eject card | Unable to eject card as needed | User can’t retrieve card | 3 - User doesn’t trust the system and/or institutional reputation damage |

***Discussion / comparison with other team:***

***Bug Projection based on test results -***

Using the capture/recapture technique, the defects found for Group 1 during the initial test run were:

* Unable to perform any transaction
* Unable to perform inquiry transaction
* Unable to perform deposit transaction
* Unable to perform withdraw transaction
* Unable to perform transfer transaction
* Unable to display any info

Since there was 2 bugs found on the second test run and they were both discovered during the initial pass as well, the total number of bugs projected for Group 1 is:

N = 6 \*2 / 2 = 6

Since there were 6 defects found in the Group 1 Project during the initial test run. The remaining defects would be:

6 – 6 = 0

For Group 2, the defects found were:

* Blank green and receipt screen
* Incorrect transfer amount on receipt screen
* To/from is backwards for invalid transfer message

Our calculation for Group 2 is:

3/5 = 1/x. 5/3-1 = 0.66 🡪 round up to 1 remaining defect to be found.

**Lessons Learned**

As in real life, we had to work around some of issues discovered during the operation testing of our ATM systems. For Group 1’s project, due to the severity and the failure cause by one of the bugs injected, we had to debug the system and fix the bug before any of the tests can be executed. As we working through our test cases, additional workaround had to be performed in order to complete these test cases. In some cases, we can proceed with the test cases with documented faults; others times, we had to restart the test cases from the start in order to complete the test case. We tried to debug the problem as much as we can, but there are times that we could not resolve the issue so we could only document these failures encountered.

For Group 2’s code, the symptom that was encountered over and over was a blank green screen and blank receipt screen when trying to perform withdrawal, deposit or balance inquiry. The code was inspected thoroughly but we could not determine the bug in the code. The eclipse debugger was used but this did not point out where the bugs were either. In a real-life situation, this may be a developer task as opposed to a tester task. I considered comparing the original code with every file in the code with bugs inserted but that seems to miss the point of debugging from a failed test case. The issue was documented and hopefully a developer who wrote the code would have a better idea of where to look.

Two different methods were used to project the total number of defects for the projects fro the two groups since there isn’t a consensus as to which one is the ‘right’ technique to use. For Group 1’s project, we used the two different test runs and the defects found on each run to calculate the total/remaining defects. For Group 2’s project, we used the technique provided in Week 7 quiz for estimating remaining defects.

One additional observation during the testing of these project; it is our experience that whenever a show-stopper occurs, the best way to address the issue would be having the team familiar with the software help troubleshooting the issue and resolve it before additional resource is consumed with the testing. That way, valuable resource isn’t waste on testing that would have to repeat again in later time. This benefit the test team as well as the overall team morale that focus can be placed on fixing the problem and enhance collaboration between the teams.

Lastly, other than the 1st defect found for Group 1, the remaining tests conducted were ‘black box’, i.e. the outcomes of the test cases determines if a ‘defect’ was found or not. When the same defects are found multiple times, we group these defects as one instead of multiple defects. In the real world, we may not have the luxury to understand how the system should work and these defects may need to document separately; thus the number of defects found could be quite a bit higher.

**Honor Pledge**

We pledge on our honor that we have not given or received any unauthorized assistance on this assignment/examination. We further pledge that we have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source.