|  |  |
| --- | --- |
| **Bug Number** | 2 |
| **Description** | Patron can borrow more items than the set limit for borrowing in one session and can borrow more items if they already have active loans equal to the loan limit. |
| **Pre-conditions** | Remove library.obj, build library from scratch |

# Step 1 : Replication

* This bug can be replicated by running ‘Bug 2 FAT.docx’. Below is an example with sample output (before the bug is resolved):

|  |  |  |
| --- | --- | --- |
| **Step** | **Expected Output/Result** | **Actual** |
| Attempt to borrow 3 Items | Loan limit = 2 Loans.  User cannot borrow 3 items | **Text  Description automatically generated**Patron can borrow 3 items: |
| Attempt to borrow more items while having 3 active loans | Loan limit = 2 Loans.  User cannot borrow more items | Patron can borrow more items:  **Text  Description automatically generated** |

# Step 2 : Simplification

|  |  |
| --- | --- |
| **Automated Test** | Bug2Test |
| **Description** | The patron has 2 loans already, attempt to scan another item. The patron should not be able to as they have reached the loan limit. |
| **Required File** | Bug2Test.java |
| **Set up** | Create a library, patron and 3 items.  Borrow 2 items and issue 2 loans. |
| **Test** |  |
| **Result** | No exception is thrown, meaning the loan limit is not enforced upon scanning the third item: |

# Step 3 : Tracing

|  |  |
| --- | --- |
| **Debugging Log** | Bug 2 using Bug2Test |
| **Initial Observation** | Bug2Test confirms the bug. The program does not prevent the patron from borrowing more than 2 items at a time, upon scanning the third item. |

## Hypothesis 1:

|  |  |
| --- | --- |
| **Description** | Line 59 getNumberofCurrentLoans returns incorrect value  Patrons number of current loans does not report/return the correct value. Causing the program to fault and allow more than limit. |
| **Class** | Patron.java |
| **Line / Variable** | Line 59 getNumberofCurrentLoans  Variable: currentLoans.size(); |
| **Prediction** | currentLoans.size incorrect |
| **Test** | Enter breakpoint:    Then:   1. Take out a loan 2. Borrow another item, complete that loan 3. Borrow another item. |
| **Result** | The currentLoans.size method returns the **correct value**.  After first borrow:    Second borrow: |
| **Conclusion** | Hypothesis is rejected |

## Hypothesis 2:

|  |  |
| --- | --- |
| **Description** | We now know that the number of current loans reported is correct. Now we will check to see if the flag for checking if a patron can borrow is working correctly. |
| **Class** | Library.java |
| **Line/Variable** | 49 canPatronBorrow |
| **Prediction** | Library.canPatronBorrow(patron) returns true after borrowing 2 items, and attempting to borrow a third |
| **Test** | Enter break points:  Line 49: |
| **Result** | Attempt to borrow first item:  canPatronBorrow returns true as it is the first item:  Complete borrowing:    Attempt to borrow 2nd Item. canPatronBorrowReturns true again.    Patron now has 2 loans:    Attempt a third borrow:    canPatronBorrow returns false == **sane.** |
| **Conclusion** | Hypothesis rejected. |

## Hypothesis 3:

|  |  |
| --- | --- |
| **Description** | From the above debug trace, we now know that the canPatronBorrow method *works* when we try to borrow items using separate transactions.  This means that there is a bug in the program when we try to borrow multiple items within the same transaction.  Thus, line 79 library.getNumberOfLoansRemainingForPatron(patron) - pendingList.size() < 0 must contain a logic error. |
| **Class** | BorrowItemControl.java |
| **Line / variable** | 79 |
| **Prediction** | library.getNumberOfLoansRemainingForPatron(patron) returns correct value  pendingList.size() returns correct value  if (library.getNumberOfLoansRemainingForPatron(patron) - pendingList.size() < 0) contains logic error. |
| **Test** | Enter breakpoint line 79:    Then run Bug 2 FAT. |
| **Result** | Scan first item and pendingList.size = 1:    Scan second item and pendingList.size = 2:    Scan third item  pendingList.size = 3 while loansRemaining = 2:      The statement library.getNumberOfLoansRemainingForPatron(patron) - pendingList.size() < 0 will only return true once the pending list of loans is HIGHER than the loan limit. |
| **Conclusion** | Hypothesis is accepted. |

# Step 3 : Resolution

|  |  |
| --- | --- |
| **Bug Source** | From the debugging trace above the statement:  will only return true once the pending list of loans is HIGHER than the loan limit. Meaning the patron can borrow more than the loan limit. |
| **Solution** | Edit the statement to the following:    This will correctly check if the pending loans is the same as the loan limit. |
| **Confirmation via automated test** | Bug2Test:  Result: |
| **Confirmation via FAT** | Loan Limit of 2 items is enforced PER transaction:    Then if the same patron tries to borrow another item while having 2 loans already they cannot:    (Loans list for patron 1): |