| **Test Name** | | | MemberHasOverDueLoansTest | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Tested:** | | | Check if Member has overdue loans, in conditions depending on whether Member have or have not had an overdue loan | | | |
| **Test Description:** | | | This test will check if a dummy or mock Member class with a single loan inside it will be marked as overdue depending on the current date written onto the test script. | | | |
| **Pre-conditions** | | | * The Member class must be initialized * A mock Loan class and Book class must be produced * A Boolean variable checking if the conditions are true or false is produced within each test function | | | |
| **Post-conditions** | | | * The process is successfully completed and the Boolean variable will state whether it is true or not * The TestRunner class will state if there are no failures encountered in the test script. | | | |
| **Notes:** | | **When test is run, it gives out an exception regarding the ELoanState borrowState of the loan object stored in the Member object, despite calling the Commit function prior to or after pushing the loan object into the member object. Bypassing this exception due to the borrowState not changing from PENDING to CURRENT allowed the test to run as normal, providing the correct answer for each – in this test, the exception error that occurs when calling the CheckOverDue(CurrentDate) function is retained so as to demonstrate the problematic function, which otherwise functions when the exception error is bypassed or where the check conditions are moved to first priority.** | | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | **Warning (due to some problematic errors involving the ELoanState not persisting properly. Bypassing the troubled ELoanState exception errors gives a correct answer for each test)** | | | | |
|  | **TEST STEP** | | | **EXPECTED TEST RESULTS** | P | F |
|  | Test if a member with a loan that is not due will result in the test being successful when the member has no overdue loan. | | | The test will state that the member has no overdue loan, and the TestRunner class response after the end of the test will state that it is successful. | X |  |
|  | Test if a member with an overdue loan will result in the test being successful if the member has an overdue loan. | | | The test will state that the member has an overdue loan, and the TestRunner class response after the end of the test will state that it is successful. | X |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Data Table** | | | | | |
|  | **1** | **2** | **3** | **4** | **5** |
| Member member | “Bob” | “Smith” | “012345678” | “email@webmail.com” | 1 |
| Book book1 | “Alice King” | “Ten Thousand Pies Under the Kitchen” | “12341234” | 1 |  |
| testMemberHasNoOverDueLoans().Loan loan1 | book1 | testMember | Date.valueOf(“2015-09-19”) | Date.valueOf(“2015-10-03”) | 1 |
| testMemberHasOverDueLoans().Loan loan1 | book2 | testMember | Date.valueOf(“2015-09-19”) | Date.valueOf(“2015-10-03”) | 1 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |