# Scenario 1: Test Returned Borrowed Book Use Case

## Scenario Description

* This test scenario will test the functionality of the Returned Borrowed Book use case.
* The use case revolves around the concept that a borrowed book can be returned to the library system either in an undamaged or damaged state. Depending on how the book is returned to the library (damaged or undamaged), the *IBook* object state should reflect this.

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 0.1 | 24/08/2014 | Justin McKay | Initial Draft |
| 1.0 | 27/08/2014 | Author Name | Initial Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Return Borrowed Book

## Use Case

* This test scenario will test the functionality of the **Returned Borrowed Book** use case.

## Test Components/Requirements

This test scenario covers the following high-level test requirements (see scripts below for specific requirements covered by each test script):

* *ILoanDAO* stub object
* *ILoan* mock object
* *IBook* mock object
* *IMember* mock object

## User Groups

* Library Admin staff

## Script 1.1: Return Borrowed Book

### Script Description

* This test will utilize the IBOOK2 – Test\_Can\_Return\_Book unit test to ensure that a book that has been loaned out can be returned. If the book is returned in a damaged state, the system must reflect this state when returning the book. Otherwise the book becomes available once more.

### Testing Requirements

This test script covers the following specific testing requirements:

* This test script will test the use case requirement to return a borrowed book, either damaged or undamaged.

### Setup

* To run this test script, an *IBook* mock object will be required to created and loaned out to a member. We can then call the *IBook.returnBook* method to return the book. This will require the following:
* An *IMember* mock object used to create the *ILoan* object.
* Two *IBook* mock objects initially in the ON\_LOAN state (so that we can test the *returnBook* method) are created. One to be returned undamaged, the other to be returned damaged.
* Use an *ILoanDAO* object to add, retrieve and update the *ILoan* object once the books have been returned.

### Teardown

* Remove the *ILoan* object from the *ILoanDAO* object.
* Each of the mock objects will be destroyed upon the completion of the test.

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Create mock *IMember* object | A valid *IMember* mockobject is created. |  |
| 2 | Create 2 mock *IBook* objects | 2 valid *IBook* mockobjects are created. |  |
| 3 | Create 2 mock *ILoan* objects based on the mock *IMember* & *IBook* objects | 2 valid *ILoan* objects are created |  |
| 4 | Add the *ILoan* objects to the repository using an *ILoanDAO* object | The *ILoan* object is added to the *ILoanDAO* repository successfully |  |
| 5 | Retrieve the *ILoan* objects from the *ILoanDAO* repository based on the id of the loans | The correct *ILoan* objects are returned |  |
| 6 | Get the *IBook* objects from the *ILoan objects* | The correct *IBook* objects are returned |  |
| 7 | For one of the *IBook* objects, call the *returnBook* method with the damaged flag set to false | The *IBook* object state should be set to AVAILABLE. |  |
| 8 | For the returned *IBook* call the *getState* method to ensure the correct state has been applied to the book. | The *IBook* object state should be set to AVAILABLE. |  |
| 9 | For the second *IBook* object, call the *returnBook* method with the damaged flag set to true | The *IBook* object state should be set to DAMAGED. |  |
| 10 | For the damaged returned *IBook* call the *getState* method to ensure the correct state has been applied to the book. | The *IBook* object state should be set to DAMAGED. |  |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date/Time | Tester | Test ID | Test Phase | Status |
| 27/08/14 11:35 am | Justin McKay | Script 1.1 | System Cycle 1 |  |
|  |  |  |  |  |

## Unit Tests for Return Borrowed Book Use Case

### IBOOK2 – Test\_Can\_Return\_Book

**Use Case Tested:** Return Borrowed Book Use Case

**Purpose:**To ensure that a book that has been borrowed can be returned.

**Pre-conditions:**- The book must be in the ON\_LOAN state.

**Post-conditions:**- The book will be in the AVAILABLE state if the book is returned not damaged or the book will be in the DAMAGED state if the book is returned damaged.  
- The ILoan object will be removed from the IBook object.

**Data required:**- A valid IBook object that is in the ON\_LOAN state.

|  | **TEST STEP** | **EXPECTED TEST RESULTS** | P | F |
| --- | --- | --- | --- | --- |
|  | Create mock IBook object in the ON\_LOAN state | IBook object is created |  |  |
|  | Call the *returnBook*  method with the damaged flag set to false | The *IBook* object is set returned and in the AVAILABLE state |  |  |
|  |  |  |  |  |

**Test Results:**

### **IBOOKDAO2 – Test\_Can\_Get\_Book\_By\_Id**

**Use Case Tested:** Return Borrowed Book Use Case

**Purpose:** A test to ensure that an IBook can be returned from the IBookDAO object with a specific known Id.

**Pre-conditions:**- The IBookDAO object must contain at least one book in the collection and the ID of that book must be known.

**Post-conditions:**- The IBook will be returned from the method.

**Data required:**- A valid IBook object with a known Id value.

|  | **TEST STEP** | **EXPECTED TEST RESULTS** | P | F |
| --- | --- | --- | --- | --- |
|  | Create a mock *IBook* object | New *IBook* object is created |  |  |
|  | Add the *IBook* object to the *IBookDAO* collection | The *IBook* object is added to the collection |  |  |
|  | Call the *getBookId* method to get the book from the collection | The *IBook* object is returned from the collection and the Id matches. |  |  |

**Test Results:**

### **ILOANDAO2 – Test\_Can\_Create\_Pending\_Loan**

**Use Case Tested:** Return Borrowed Book Use Case

**Purpose:**A test to ensure that an *ILoan* object can be created and to the *ILoanDAO* object based on the *IMember* and *IBook* objects.

**Pre-conditions:   
-** The *ILoanDAO* object must have the pending loan list for the specified member. **-** The *IMember* object must exist.  
- The *IBook* object must exist.

**Post-conditions:**- A new *ILoan* object is returned with a valid id and in the PENDING state.

**Data required:**- The *IBook* and *IMember* objects.  
- A borrow date and a due date.

|  | **TEST STEP** | **EXPECTED TEST RESULTS** | P | F |
| --- | --- | --- | --- | --- |
|  | Create mock *IMember* object. | New *IMember* object is created |  |  |
|  | Call the *createNewPendingList* method to create pending list for the *IMember* object. | *ILoanDAO* object has new pending loan list for the member |  |  |
|  | Call the *createPendingLoan* method with the *IBook* and *IMember* object to create the new pending loan. | The new *ILoan* object is created in the PENDING state |  |  |

**Test Results:**

### IBOOK5 – Test\_Can\_Get\_Correct\_Book\_State

**Use Case Tested:** Return Borrowed Book Use Case

**Purpose:**To ensure that the state of the book returned from the *IBook.getState()* method is the current state of the book.

**Pre-conditions:**- The book must be in either the AVAILABLE, ON\_LOAN, DISPOSED, DAMAGED or LOST state.

**Post-conditions:**- The book will be in the original state.

**Data required:**- A valid IBook object that is in one of the AVAILABLE, ON\_LOAN, DISPOSED, DAMAGED or LOST states.

|  | **TEST STEP** | **EXPECTED TEST RESULTS** | P | F |
| --- | --- | --- | --- | --- |
|  | Create mock *IBook* object. | New *IBook* object is created |  |  |
|  | Call the *getState* method to ensure the state is returned. | The state returned should be AVAILABLE |  |  |
|  |  |  |  |  |

**Test Results:**