# Scenario 2: Happy Day Scenario

## Scenario Description

* This scenario is for the user case Borrow Book and will focus on the normal flow of borrowing the book in a regional library and alternative flow while scanning books.
* It covers all the processes and steps, which were, need to be implemented in the use case and the exception that occurs while scanning books
* For example, how the printing borrow details work.

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date | Author | Description |
| 1.0 | 01/10/2016 | Amar Adhikari | Initial Version |

## Test Scripts

The following scripts will cover this scenario:

* 1.1 Normal Flow of book borrowing
* 1.2 book not found
* 1.3 book not available
* 1.4 book already scanned

## Use Case

* This scenario covers the use case **Borrow Book**.

**Description:**

The use case starts with the scanning of library card. However the burrower must be registered in the database initially. The burrower now takes the book with scanning in the scanner. And finally the loan slip is shown and printed.

**Actor:**

Burrower is the key person who must be the member of that library and must have library card to scan and hence can burrow books.

**Precondition:**

1. The burrower must be on library system database.
2. The burrower must scan library card initially.
3. The burrower should scan the book properly.

**Normal flow of activities:**

| **Step #** | **Actor** | **system** |
| --- | --- | --- |
| 1 | Click on borrow books | 1.1 Activate cursor to enter id or swipe card |
| 2 | Enters id | 2.1 Display member current loan and show whether s/he can borrow books or not. If yes then allow to scan book or enter book id |
| 3 | Scan book | 3.1 Display the book details  3.2 Displays if burrower have overdue, loan limit, has fines and fine limit |
| 4 | Click on complete | System ask to confirm. |
| 5 | Click confirm | 4.1 All the borrowing details are sent to printing options |
| 6 | Click on Print the receipt | 4.2 Printer prints the receipts |

**Alternative flows:**

3. After scanning book if system shows error message

3.1 book not found

| **Step #** | **Actor** | **system** |
| --- | --- | --- |
| 3 | Scan book | 3.1.1 book not found |
| 4 | Click cancel | 3.1.2 return to initial state |

3.2 book not available

| **Step #** | **Actor** | **system** |
| --- | --- | --- |
| 3 | Scan book | 3.2.1 book not available |
| 4 | Click cancel | 3.2.2 return to initial state |

3.3 book already scanned

| **Step #** | **Actor** | **system** |
| --- | --- | --- |
| 1 | Scan book | 3.3.1 book already scanned |
| 2 | C lick ok | 2.1 Display member current loan and show whether s/he can borrow books or not. If yes then allow to scan book or enter book id |

**Post condition:**

Burrower takes the book.

Burrower prints the loan details.

## Test Components/Requirements

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

* Java GUI platform
* 5 borrower and 15 books should be registered
* Functional Requirements Group 1

## User Groups

* Burrowers
* System tester

## Script 1.1: Normal flow

### Script Description

* This script test the normal function of the system where the burrower scan id, scan book, confirm it and print loan details.

### Testing Requirements

This test script covers the following specific testing requirements:

* Java GUI
* 5 borrower and 15 books should be registered.

### Setup

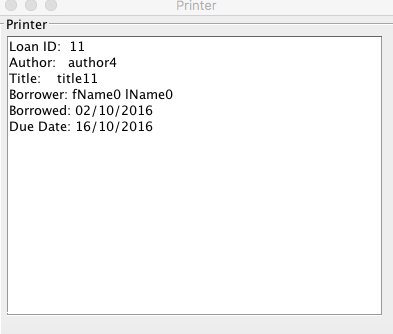
* GUI was setup-using IntelliJ and the repository was added to the members before doing the following test.
* Required test actions were performed accordingly.
* The GUI was set to initial state and actions performed from there.
* Examples might include required test data, other scripts that should be run before this one, initial states that should be set, specific required equipment/software, etc…

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Click on borrow books | Allow to enter id or swipe card | Pass |
| 2 | Enter id | Display member current loan and show whether s/he can borrow books or not. If yes then allow to scan book or enter book id | Pass |
| 3 | Enter book id | Display the book details and ask to complete. | Pass |
| 4 | Click complete | System ask to confirm | pass |
| 5 | Enter confirm | All the borrowing details are sent to printing options | Pass |
| 6 | Click on Print the loan details | Printer prints the loan details. | Pass |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| First step | Member id | Book ID | confirm | Status |
| Click on borrow | 1 | 10 | Click on yes | Passed |
| Click on borrow | 2 | 14 | Click on cancel | Failed |



All the activities are normally performed and the loan details are displayed to be print.

## Script 1.2: Book not found

### Script Description

* This script test the alternative flow while scanning book the system displays book not found.

### Testing Requirements

This test script covers the following specific testing requirements:

* Java GUI
* 5 borrower and 15 books should be registered.

### Setup

* GUI was setup-using IntelliJ and the repository was added to the members before doing the following test.
* Required test actions were performed accordingly.
* The GUI was set to initial state and actions performed from there.
* Examples might include required test data, other scripts that should be run before this one, initial states that should be set, specific required equipment/software, etc…

### Script Steps

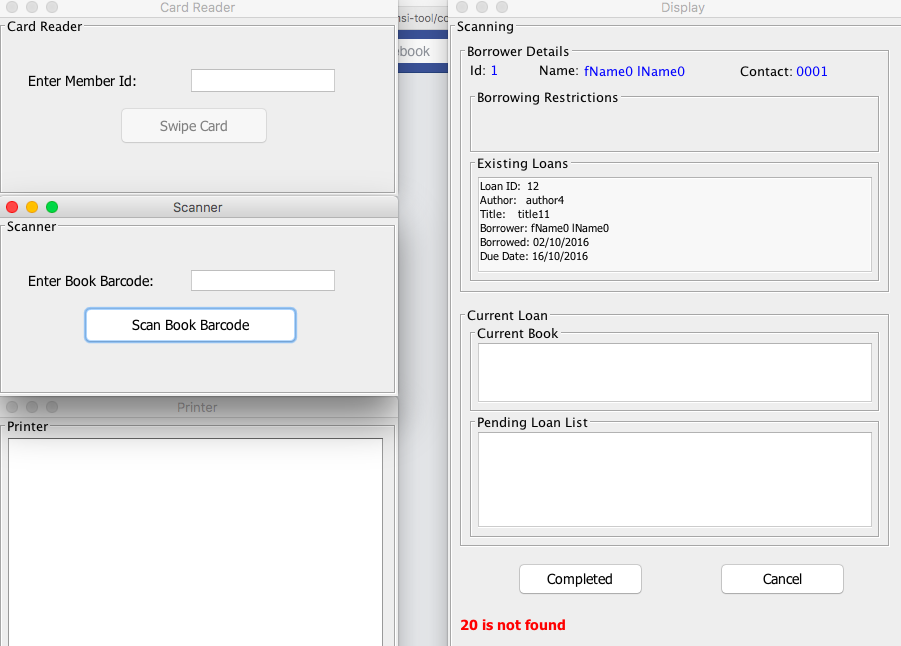
| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Click on borrow books | Allow to enter id or swipe card | Pass |
| 2 | Enter id | Display member current loan and show whether s/he can borrow books or not. If yes then allow scanning book or entering book id. | Pass |
| 3 | Enter book id | Display the book not found | Pass |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| test | Member id | Book ID | confirm | Status |
| 1. | 3 | 10 | Click on yes | Failed |
| 2. | 5 | 20 | Click on cancel | Passed |

In test 1 , book ID 10 exists so the burrower can burrow so it failed the testing.

In test 2, book ID 20 does not exist and display book not found so system passed.



## Script 1.3: Book not available

### Script Description

* This script test the alternative step while scanning book and the system display book not available

### Testing Requirements

This test script covers the following specific testing requirements:

* Java GUI
* 5 borrower and 15 books should be registered.

### Setup

* GUI was setup-using IntelliJ and the repository was added to the members before doing the following test.
* Required test actions were performed accordingly.
* The GUI was set to initial state and actions performed from there.
* Examples might include required test data, other scripts that should be run before this one, initial states that should be set, specific required equipment/software, etc…

### Script Steps

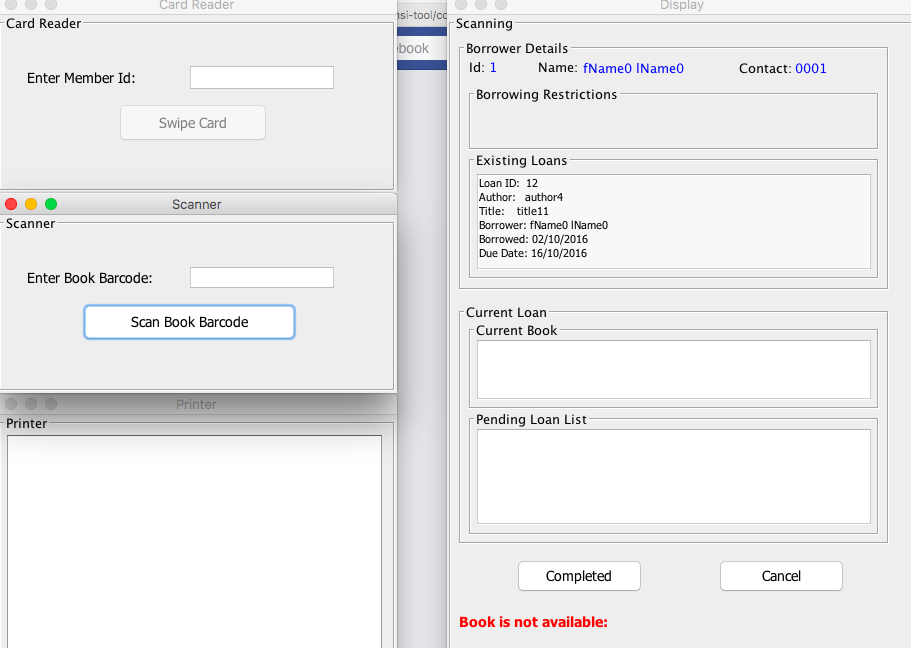
| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Click on borrow books | Allow to enter id or swipe card | Pass |
| 2 | Enter id | Display member current loan and show whether s/he can borrow books or not. If yes then allow to scan book or enter book id | Pass |
| 3 | Enter book id | Display the book not available | Pass |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| test | Member id | Book ID | confirm | Status |
| 1 | 2 | 10 | Click on yes | Failed |
| 2 | 4 | 12 | Click on cancel | Passed |

In test 1, book 10 exists and is available so the burrower can burrow and it failed the testing.

In test 2, book 11 exists but still burrower cant burrow because other has already burrowed it.



## Script 1.4: Book already scanned

### Script Description

* This script tests the alternative flow of the normal flow where the burrower scans the book repeatedly.

### Testing Requirements

This test script covers the following specific testing requirements:

* Java GUI
* 5 borrower and 15 books should be registered.

### Setup

* GUI was setup-using IntelliJ and the repository was added to the members before doing the following test.
* Required test actions were performed accordingly.
* The GUI was set to initial state and actions performed from there.
* Examples might include required test data, other scripts that should be run before this one, initial states that should be set, specific required equipment/software, etc…

### Script Steps

| **Step #** | **Test Action** | **Expected Results** | **Pass/ Fail** |
| --- | --- | --- | --- |
| 1 | Click on borrow books | Allow to enter id or swipe card | Pass |
| 2 | Enter id | Display member current loan and show whether s/he can borrow books or not. If yes then allow to scan book or enter book id | Pass |
| 3 | Enter book id | Display the book is already scanned | Pass |

### Test Execution

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| test | Member id | Book ID | confirm | Status |
| 1 | 1 | 10 | Click on yes | Passed |
| 2 | 2 | 12 | Click on cancel | passed |

In test 1 and 2 , the burrower had already scanned the book so the system display as the book has been already scanned.

