*[The Test Case ID should be unique. In addition, the name of each Test Case should reflect the intent of the test case, ideally expressed as a Boolean condition.]*

**BBUC\_Op3 – Scan Book:**

Description: This use case scans a book then adds the book to the list of intended or pending loans. The result should be true if the barcode is correct, the book is not already on the intended or pending loans list, the book is available, and that the member has not reached the loan limit. IF the loan limit is reached after scanning a book, it will force into the finalize/confirming loan screen.

Pre-conditions:

* BorrowBookCTL class must exist
* scanner class variable must exist
* BorrowBookCTL must be added as listener to scanner
* BorrowBookCTL’s state must be set to SCANNING\_BOOKS

Post-conditions:

(If book is not found due to incorrect id value)

* BorrowBookUI is displayed
* Scanning panel of BorrowBookUI displayed
* Cancel button is enabled
* cardReader is disabled (we don’t want to re-read who is borrowing the books)
* scanner is enabled (to allow scanning of more books)
* book not found error message displayed
* BorrowBookCTL’s state remains as SCANNING\_BOOKS

(If book is not available)

* BorrowBookUI is displayed
* Scanning panel of BorrowBookUI displayed
* Cancel button is enabled
* cardReader is disabled
* scanner is enabled
* book not available error message displayed
* BorrowBookCTL’s state remains as SCANNING\_BOOKS

(If book is already scanned into the pending list)

* BorrowBookUI is displayed
* Scanning panel of BorrowBookUI displayed
* Cancel button is enabled
* cardReader is disabled
* scanner is enabled
* book already scanned error message dispalyed
* BorrowBookCTL’s state remains as SCANNING\_BOOKS

(if scan count of member borrowing is less than loan limit)

* BorrowBookUI is displayed
* Scanning panel of BorrowBookUI displayed
* Cancel button enabled
* cardReader is disabled
* scanner is enabled
* scanCount incremented by 1
* scanned books details displayed
* new (pending) loan created
* pending loan added to pending loan list
* Current pending loan list refreshed and displayed
* BorrowBookCTL’s state remains as SCANNING\_BOOKS

(If scan count is at loan limit)

* BorrowBookUI is displayed
* ConfirmingLoans panel of BorrowBookUI is displayed
* Final pending loan list displayed
* Cancel button enabled
* Reject button enabled
* cardReader is disabled (loans are being finalized)
* scanner is disabled (at loan limit for member)
* scanCount incremented by 1
* BorrowBookCTL’s state changed to CONFIRMING\_LOANS

Data required:

* The book’s barcode or id variable in int
* The memberDAO object that contains the members
* The bookDAO object that contains the books that are available in the library
* The loanDAO object that contains any existing loans that are applied to members
* The BorrowBookCTL class
* The borrower, who is a member class that is obtained from the memberDAO class object

**BBUC\_Op5 – Confirm Loans:**

Description: This test case performs the process of completing the borrowing book process. This is a process where the loans are being completed, and once completed, will return the member back to the program’s main menu where it awaits user input

Pre-conditions:

* BorrowBookCTL class must exist
* Pending loan list must exist
* BorrowBookCTL’s state must be set to CONFIRMING\_LOANS

Post-conditions:

* The Main Menu is displayed
* All pending loans are committed and recorded into the loanDAO database class
* Loan slip of committed loans is printed out from the printer
* cardReader is disabled (we are not scanning any members immediately)
* scanner is disabled (we are finalizing loans – no need to scan more books)
* BorrowBookCTL’s state is changed to COMPLETED

Data required:

* It definitely requires BorrowBookCTL
* A pending loans list containing valid book class objects
* A member to borrow the book
* The MemberDAO, bookDAO, and loanDAO objects