**1 – Test Application Initialization Successfully Completes:**

Description: Tests that the java application begins correctly. Should result in the Main menu, Printer, Scanner, and Card Reader being displayed with the Main menu “Borrow Books” button being enabled.

Pre-conditions:

* BorrowBookCTL class exists
* memberDAO exists
* loanDAO exists
* bookDAO exists
* display exists
* cardReader exists
* scanner exists
* printer exists
* BorrowBookCTL added as listener to cardReader and scanner
* BorrowBookCTL state == ***CREATED***

Post-conditions:

* BorrowBookUI is displayed
* SwipeCard panel of BorrowBookUI displayed
* Cancel button enabled
* cardReader is enabled
* scanner is disabled
* BorrowBookCTL state == ***INITIALIZED***

Data required: None

**2 – Test Card Swipe Successfully Completes:**

Description: Tests the successful entering of an existing member whom has no restrictions. Member identification and ability to scan books for specified member should result.

Pre-conditions:

* cardReader exists, is visible and enabled
* BorrowBookCTL added as listener to cardReader
* memberDAO exists
* BorrowBookCTL state == ***INITIALIZED***

Post-conditions:

* Scanning panel of BorrowBookUI displayed
* Complete, Cancel buttons enabled
* cardReader is disabled
* scanner is enabled
* borrower details displayed
* existing loans displayed
* scan count initialized as number of
* existing loans
* existing fine message displayed if relevant
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Data required: Member object and MemberID

**3 – Test Card Swipe Fails Due to Restrictions:**

Description: Tests the event that a member exists but has restrictions. Member identification and a borrowing restriction error message should result, but the ability to scan books is not enabled.

Pre-conditions:

* cardReader exists, is visible and enabled
* BorrowBookCTL added as listener to cardReader
* memberDAO exists
* BorrowBookCTL state == ***INITIALIZED***

Post-conditions:

* Restricted panel of BorrowBookUI displayed
* Cancel button enabled
* cardReader is disabled
* scanner is disabled
* borrower details displayed
* existing loans displayed
* existing fine message displayed if relevant
* overdue message displayed if relevant
* atLoanLimit message displayed if relevant
* borrowing restricted error message displayed
* BorrowBookCTL state == ***BORROWING\_RESTRICTED***

Data required: Member object and MemberID

**4 – Test Card Swipe Fails Due to Member Not Existing:**

Description: Tests the event that a member doesn’t exist. Should result in a member not found error message and the ability to scan books not being enabled.

Pre-conditions:

* cardReader exists, is visible and enabled
* BorrowBookCTL added as listener to cardReader
* memberDAO exists
* BorrowBookCTL state == ***INITIALIZED***

Post-conditions:

* Display error message “MemberID (ID) not found”

Data required: Member object and MemberID

**5 – Test Book Scan Successfully Completes (One Book):**

Description: Tests the successful completion of scanning one book. Should result in the book details being shown in the current book and pending loan list sections.

Pre-conditions:

* BorrowBookCTL class exists
* scanner exists
* BorrowBookCTL added as listener to scanner
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Post-conditions:

* + BorrowBookUI is displayed
  + Scanning panel of BorrowBookUI displayed
  + Cancel button enabled
  + cardReader is disabled
  + scanner is enabled
  + book details displayed
  + new pending loan created
  + pending loan added to pending loan list
  + Current pending loan list displayed
  + BorrowBookCTL state == ***SCANNING\_BOOKS***

Data required: Book object, member object

**6 – Test Book Scan Successfully Completes (Two Books):**

Description: Tests the successful completion of scanning two books. Should result in both book details being shown in the current book (when it is the most recently scanned book) and pending loan list sections.

Pre-conditions:

* BorrowBookCTL class exists
* scanner exists
* BorrowBookCTL added as listener to scanner
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Post-conditions:

* + BorrowBookUI is displayed
  + Scanning panel of BorrowBookUI displayed
  + Cancel button enabled
  + cardReader is disabled
  + scanner is enabled
  + Both book details displayed
  + new pending loans created
  + pending loans added to pending loan list
  + Current pending loan list displayed after each book is scanned
  + BorrowBookCTL state == ***SCANNING\_BOOKS***

Data required: 2 Book objects, member object

**7 – Test Book Scan Fail Due to Book Not Found:**

Description: Tests the event that a book isn’t found in the system when scanned. Should result in a book not found error message.

Pre-conditions:

* BorrowBookCTL class exists
* scanner exists
* BorrowBookCTL added as listener to scanner
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Post-conditions:

* + BorrowBookUI is displayed
  + Scanning panel of BorrowBookUI displayed
  + Cancel button enabled
  + cardReader is disabled
  + scanner is enabled
  + book not found error message displayed
  + BorrowBookCTL state == ***SCANNING\_BOOKS***

Data required: Book object, member object

**8 – Test Book Scan Fail Due to Book Not Available:**

Description: Tests the event that a book isn’t available because its state is not available. Should result in a book not available error message followed by the books state.

Pre-conditions:

* BorrowBookCTL class exists
* scanner exists
* BorrowBookCTL added as listener to scanner
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Post-conditions:

* + BorrowBookUI is displayed
  + Scanning panel of BorrowBookUI displayed
  + Cancel button enabled
  + cardReader is disabled
  + scanner is enabled
  + book not available error message displayed
  + BorrowBookCTL state == ***SCANNING\_BOOKS***

Data required: Book object, member object

**9 – Test Book Scan Fail Due to Book Already Scanned:**

Description: Tests the event that a book can’t be scanned because it already has been scanned during this session. Should result in an error message stating the bookID and notifying that it is already scanned.

Pre-conditions:

* BorrowBookCTL class exists
* scanner exists
* BorrowBookCTL added as listener to scanner
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Post-conditions:

* + BorrowBookUI is displayed
  + Scanning panel of BorrowBookUI displayed
  + Cancel button enabled
  + cardReader is disabled
  + scanner is enabled
  + book already scanned error message displayed
  + BorrowBookCTL state == ***SCANNING\_BOOKS***

Data required: 2 Book objects, member object

**10 – Test Book Scan Fail Due to Loan Limit Exceeded:**

Description: Test the event that books can’t be scanned because the member’s loan limit has been reached. Should result in the scanner not being enabled and borrowing restriction error messages being displayed in the borrowing restriction section and button of display window.

Pre-conditions:

* BorrowBookCTL class exists
* scanner exists
* BorrowBookCTL added as listener to scanner
* BorrowBookCTL state == ***SCANNING\_BOOKS***
* Member details provided

Post-conditions:

* Display borrowing restrictions notice
* Display error that member cannot borrow.

Data required: Member 4

**11 – Test Complete Scans Successfully Completes:**

Description: Test the successful completion of completing scans. After the confirm button is pressed it should result in in the Main menu changing to the confirm loans pane, the confirm, reject and cancel buttons being enabled and scanner being disabled.

Pre-conditions:

* BorrowBookCTL class exists
* Pending loan list exists
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Post-conditions:

* BorrowBookUI is displayed
* Confirming Loan panel of BorrowBookUI is displayed
* List of pending loans displayed
* Cancel button enabled
* cardReader is disabled
* scanner is disabled
* BorrowBookCTL state == ***CONFIRMING\_LOANS***

Data required: Current loan details, book details, member details

**12 – Test Complete Scans Successfully Completes via Automatic Activation Due to Loan Limit Met:**

Description: Test the successful completion of completing scans due to the members loan limit being met. Should result in the Main menu changing to the confirm loans pane, the confirm, reject and cancel buttons being enabled and scanner being disabled, all automatically without having pressed the confirm button.

Pre-conditions:

* BorrowBookCTL class exists
* Pending loan list exists
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Post-conditions:

* BorrowBookUI is displayed
* Confirming Loan panel of BorrowBookUI is displayed
* List of pending loans displayed
* Cancel button enabled
* cardReader is disabled
* scanner is disabled
* BorrowBookCTL state == ***CONFIRMING\_LOANS***

Data required: Current loan details, book details, member details

**13 – Test Loan Confirmation Successfully Completes:**

Description: Test the successful completion of the current loan being confirmed. Should result in the loan being added to the member, the printer activating and displaying loan details and the Main menu returning to original state.

Pre-conditions:

* BorrowBookCTL class exists
* Pending loan list exists
* BorrowBookCTL state == ***CONFIRMING\_LOANS***

Post-conditions:

* Main Menu is displayed
* All pending loans are committed and recorded
* Loan slip of committed loans printed
* cardReader is disabled
* scanner is disabled

BorrowBookCTL state == ***COMPLETED***

Data required: Loan details, member details

**14 – Test Loan Confirmation Fails Due to Cancelling:**

Description: Test the event that the loan is not confirmed because the cancel button is pressed. Should result in the current loan details being deleted and the Main menu returning to original state.

Pre-conditions:

* BorrowBookCTL class exists
* Pending loan list exists
* BorrowBookCTL state == ***CONFIRMING\_LOANS***

Post-conditions:

* Main Menu is displayed
* All pending loans deleted
* BorrowBookCTL state == ***INITIALIZED***

Data required: Loan details, member details

**15 – Test Loans Rejected Successfully Completes:**

Description: Test the successful completion of the loans being rejected. Should result in the scanner being enabled and the Display window returning to previous state not showing the previously scanned book.

Pre-conditions:

* BorrowBookCTL class exists
* Pending loan list exists
* BorrowBookCTL state == ***CONFIRMING\_LOANS***

Post-conditions:

* BorrowBookUI is displayed
* Scanning panel of BorrowBookUI displayed
* Borrower details displayed
* Existing loan details displayed
* pending loan list is empty
* scan count == number of existing loans
* Cancel button enabled
* cardReader is disabled
* scanner is enabled
* BorrowBookCTL state == ***SCANNING\_BOOKS***

Data required: Loan details

**16 – Test Loans Rejected Successfully Completes Due to Cancelling:**

Description: Tests the successful completion of the loans being rejected by clicking the cancel button. Should result in the Display window returning to the Main menu pane.

Pre-conditions:

* BorrowBookCTL class exists
* Pending loan list exists
* BorrowBookCTL state == ***CONFIRMING\_LOANS***

Post-conditions:

* Main Menu is displayed
* All pending loans deleted
* BorrowBookCTL state == ***INITIALIZED***

Data required: Loan details