1. **Testing Strategy**

As the borrowing system is needed to show correct results so issues such as overdue loans restrict the user if necessary. Integration testing will be done on the demo one provided within the resources.

1. Unexpected behaviour occurs on the entity of a well formatted user input
2. The system becomes broken if the user enters incorrect data

Desired functionality:

The interface should process through the, borrow a book use case without any errors.

|  |  |
| --- | --- |
| Field | Requirement |
| memberId | Integer above 0 and not null |
| Barcode | Integer above 0 and not null |

|  |  |
| --- | --- |
| Test steps | Result |
| 1. Click on Borrow books | Pass (initializes memberId scanner) |
| 1. Enter a string into memeberId text field | Pass (cannot enter a string) – returns a message member id must be a integer |
| 1. Enter 0 | Pass returns a message of needing a positive integer |
| 1. Enter 1 | Pass processes through to barcode scanner |
| 1. Enter a string | Pass – returns a message of needing a positive integer |
| 1. Enter a symbol e.g. @ | Pass – returns a message of needing a positive integer |
| 1. Enter 10 | Pass book exists and is not on loan |
| 1. Press cancel | Pass |
| 1. Press complete | Pass |
| 1. Press cancel , reject, confirm | Pass |
| 1. Try borrowing the same book | Pass returns ON\_LOAN |

The overall system works well because it disables the other windows when it transitions through the process. It does not allow room for human error as there is no way to enter something in the previous window.

**Checks:**

1. Member id is a positive integer (not string or negative)
2. Member exists within the system
3. Barcode is a positive integer
4. Book exists in the system
5. Book is available
6. Member is able to borrow (is not restricted)
7. Loan limit is not reached

\*if all conditions have been met it will process through the borrow\*