

Test Plan

for

Library Inventory Control

Anil Thakuri
Blesson Thomas
Maria Vassileva
Robert Whitaker
Timothy Wolf
Donna Young

Revision History

Version	Release Date	Author	Major Changes
1.0	4/19/2015	Group Collaboration	Initial Document

Table of Contents

- [1. Introduction](#)
- [2. Test Items \(Functions\)](#)
- [3. Features to be Tested](#)
 - [3.1 Add a new Book](#)
 - [3.2 Remove a Book](#)
 - [3.3. Update information about a Book](#)
 - [3.4. Check out a Book](#)
 - [3.5. Check-In a Book](#)
 - [3.6. Issue a new Library Card](#)
 - [3.7. Revoke Library Card](#)
 - [3.8. Pay Fine](#)
 - [3.9. Search for a Book](#)
 - [3.10. Generate Report](#)
 - [3.11. Add self to waiting list for a Book](#)
 - [3.12. Remove Patron from waiting list](#)
- [4. Approach](#)
 - [4.1 Component Testing](#)
 - [4.2 Interface Testing](#)
 - [4.3 Performance Testing](#)
- [5. Item Pass/Fail Criteria](#)
 - [5.1 Approval Criteria](#)
- [6. Test Deliverables](#)
- [7. References](#)

1. Introduction

The test plan for the Library Inventory Control application tests the application according to the software requirements specification. The main purpose of the testing is to discover errors, faults, and failures and fix them accordingly.

2. Test Items (Functions)

Items to be tested according to the requirements specifications:

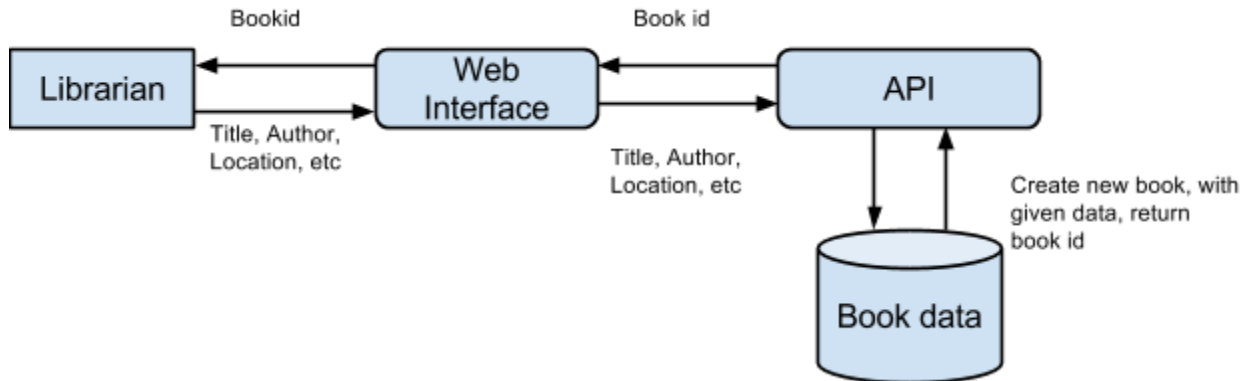
- Add a new book into the system
- Remove a book from the system
- Update Information about a book in the system
- Check-out a book
- Check-in a book
- Issue a new Library Card
- Revoke a Library Card
- Pay a fine for overdue or lost / damaged books
- Search for a book in the system

3. Features to be Tested

For this project we are doing a black box testing. The developed test cases with JUnit, test the input and the output to make sure the requirements are met.

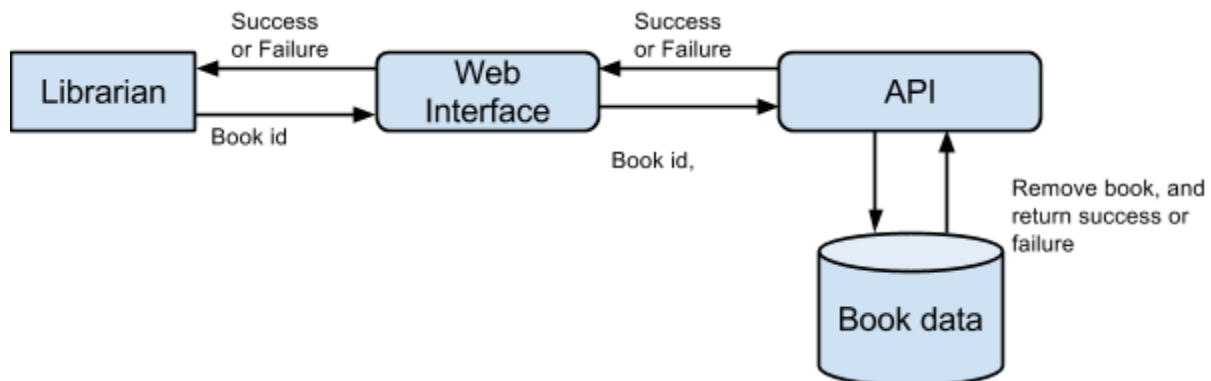
3.1 Add a new Book

- Preconditions
 - Librarian Class user successfully logged in. Add book screen.
- Test Case Input
 - Book Id, All Book data
- Test Expected Output
 - Success Message
- Verification
 - Book Detail View should show should show updated data.
 - Check if the system allows you to enter title
 - Check if the system allows you to enter ISBN
 - Check if the system allows you to select an author
 - Check if the new book is in database



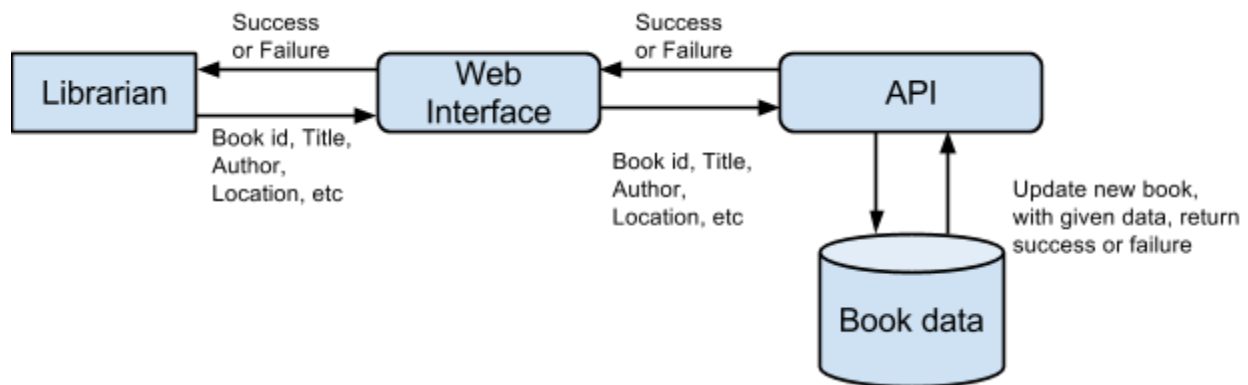
3.2 Remove a Book

- Preconditions
 - Librarian Class user successfully logged in. Book detail screen.
- Test Case Input
 - Book Id, delete method
- Test Expected Output
 - Success Message
- Verification
 - Searching for the book should no longer return a result.



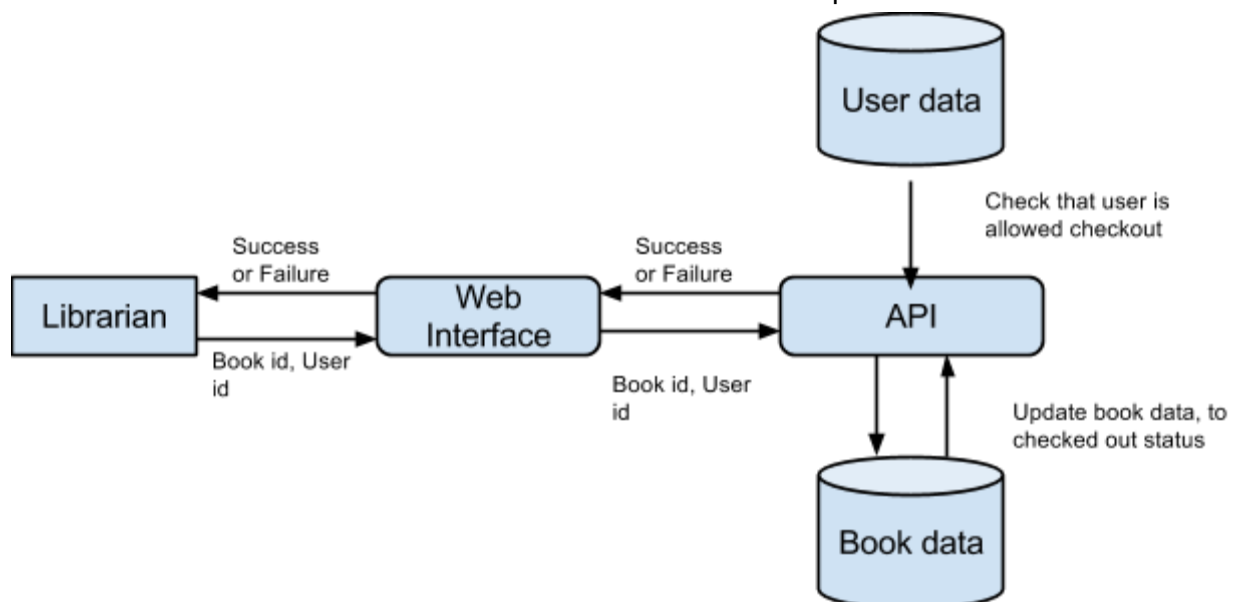
3.3. Update information about a Book

- Preconditions
 - Librarian Class user successfully logged in. Book detail screen.
- Test Case Input
 - Book Id, all data about the book
- Test Expected Output
 - Success Message
- Verification
 - Book Detail View should show updated data.



3.4. Check out a Book

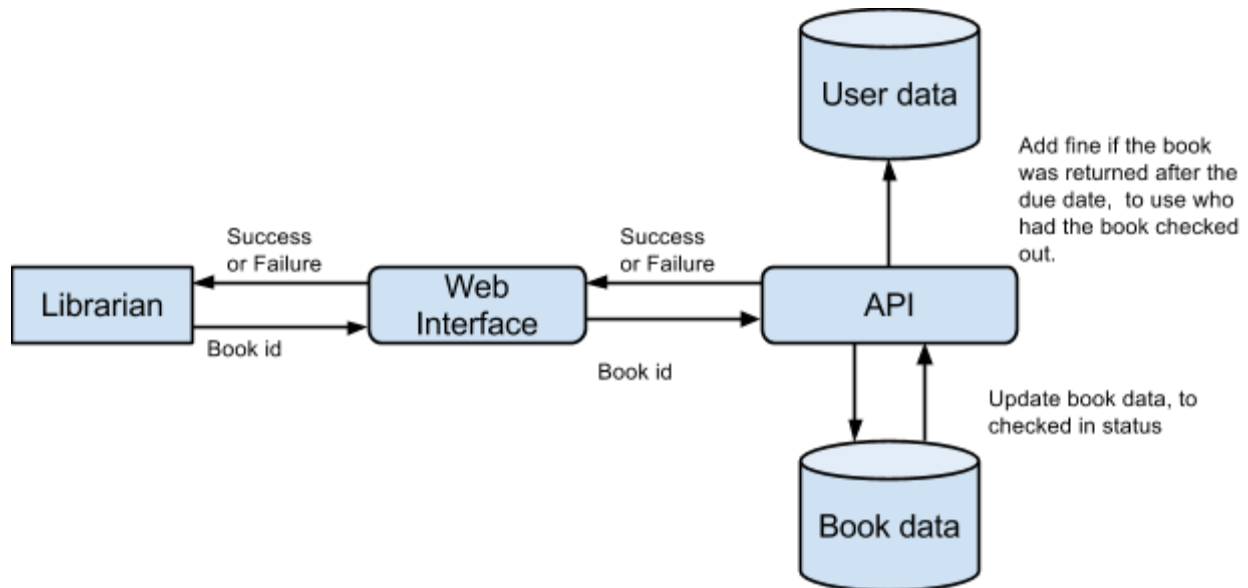
- Preconditions
 - Librarian Class user successfully logged in. Checkout Screen.
- Test Case Input
 - Book Id, Patron Id
- Test Expected Output
 - Success Message
- Verification
 - Book Detail View should show
 - “CHECKEDOUT” status
 - Checkout User should be patron with id above



3.5. Check-In a Book

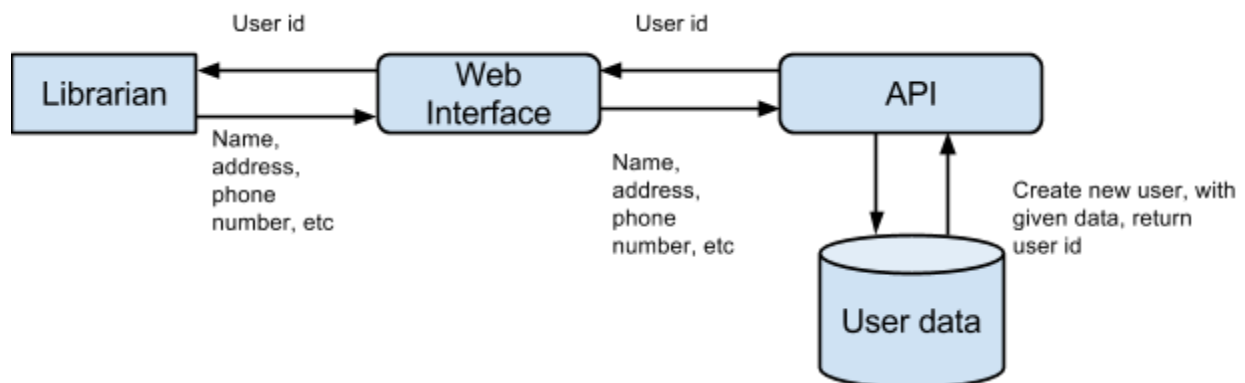
- Preconditions
 - Librarian Class user successfully logged in. Checkin Screen
- Test Case Input

- Book Id
- Test Expected Output
 - Success Message
- Verification
 - Book Detail View should show
 - “CHECKEDIN” status



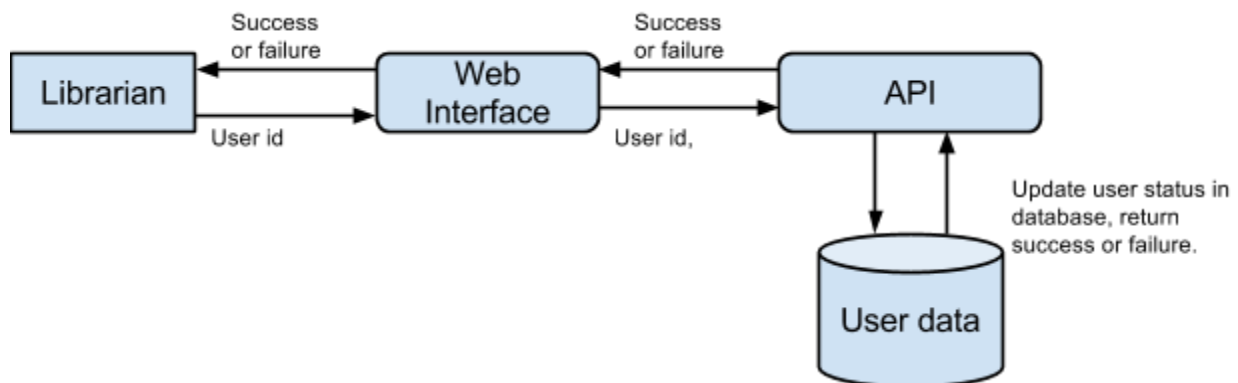
3.6. Issue a new Library Card

- Preconditions
 - Librarian Class user successfully logged in. New User Screen
- Test Case Input
 - All Library Patron Data
- Test Expected Output
 - Success Message
- Verification
 - Patron Detail View should show
 - All Data entered



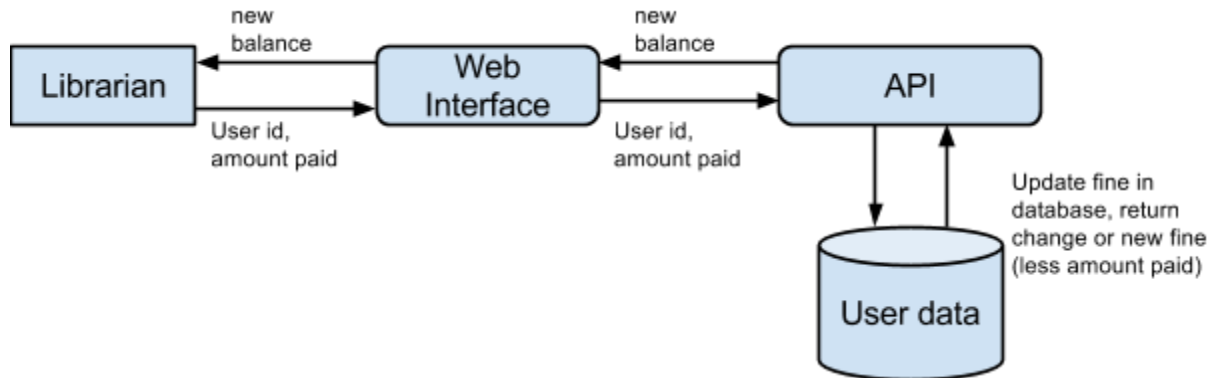
3.7. Revoke Library Card

- Preconditions
 - Librarian Class user successfully logged in. User Detail Screen
- Test Case Input
 - User Disable
- Test Expected Output
 - Success Message
- Verification
 - Patron Detail View should show
 - User Disabled



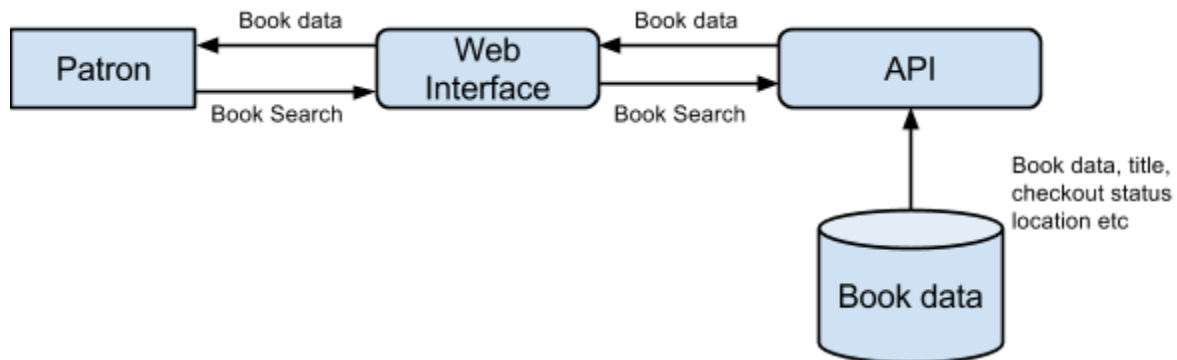
3.8. Pay Fine

- Preconditions
 - Librarian Class user successfully logged in. User Detail Screen
- Test Case Input
 - Amount Tendered
- Test Expected Output
 - Success Message, new fine less fine amount.
 - If Fine < Amount Tendered, return a Fine - Amount Tendered as change.
- Verification
 - Patron Detail View should show
 - New fine amount.



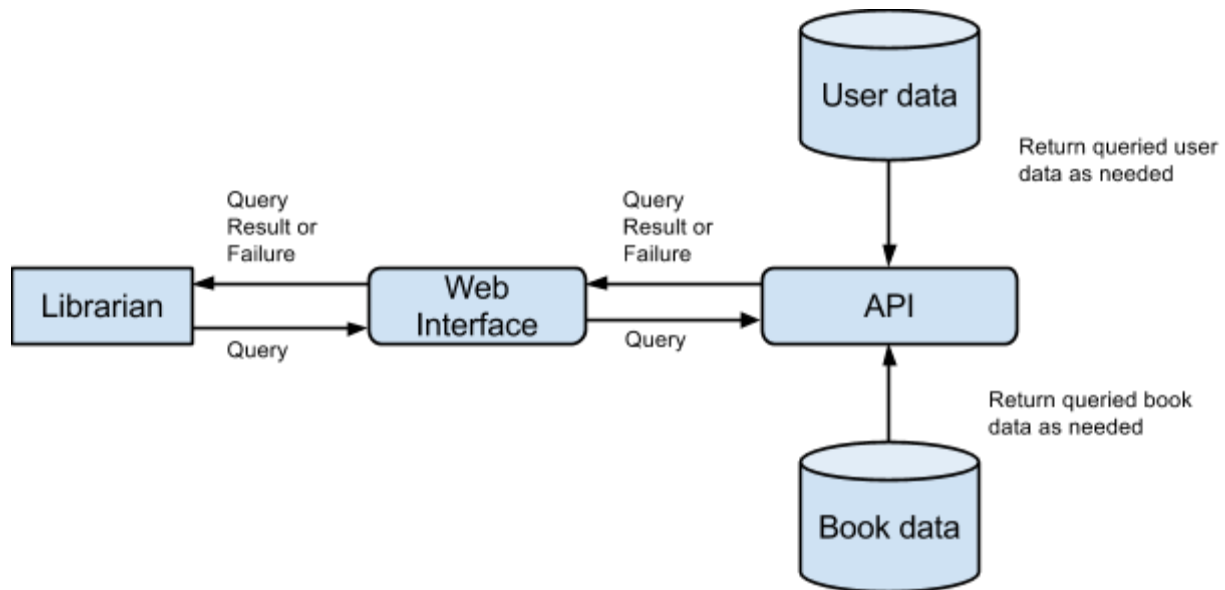
3.9. Search for a Book

- Preconditions
 - Book Search Screen
- Test Case Input
 - Any book data they want to search for
- Test Expected Output
 - A list of possible matches
- Verification
 - Check if displayed details match the search criteria.



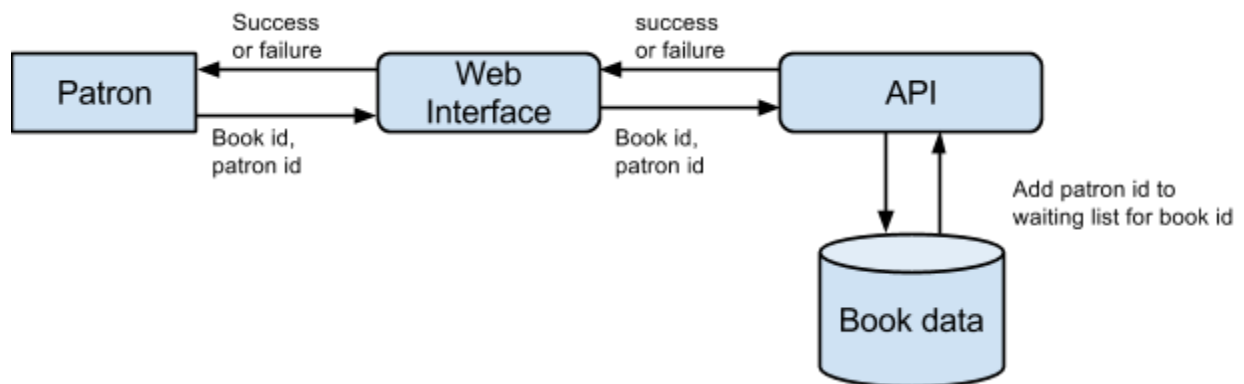
3.10. Generate Report

- Preconditions
 - Reports Screen
- Test Case Input
 - Parameter(s) for Report
- Test Expected Output
 - Reporting results for parameter given
- Verification
 - Make sure proper results are displayed for report



3.11. Add self to waiting list for a Book

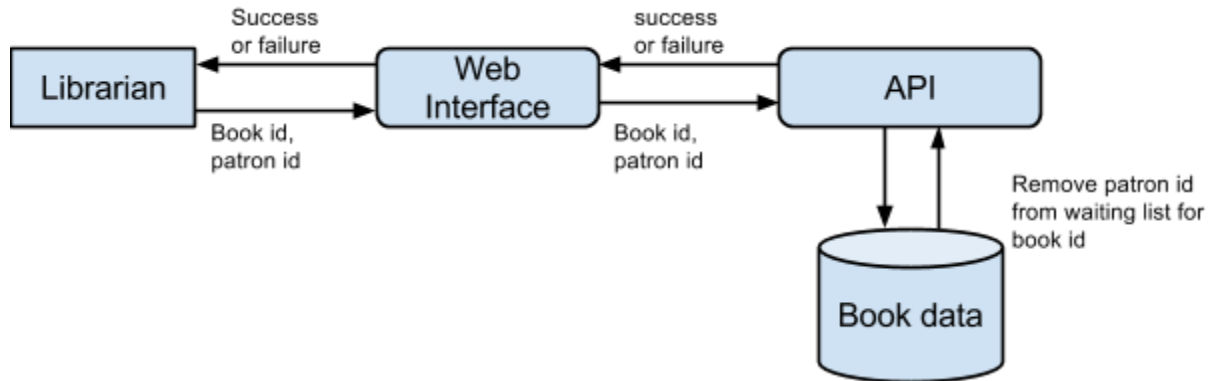
- Preconditions
 - Library User Logged in, Book Detail Screen
- Test Case Input
 - book id
- Test Expected Output
 - Success
- Verification
 - View User waiting list



3.12. Remove Patron from waiting list

- Preconditions
 - Library User Logged in, User Detail Screen
- Test Case Input
 - book id, User Id

- Test Expected Output
 - Success
- Verification
 - View User waiting list, book should be missing



4. Approach

We have developed several test cases that test the above functionalities of the application.

- BookControllerTest
- AuthenticationControllerTest
- ReportControllerTest
- BookMapperTest
- AuthorMapperTest
- UserMapperTest

4.1 Component Testing

The purpose of the component testing is to ensure that all components work together as designed.

4.2 Interface Testing

Application was tested on two operating systems - Mac OS and Windows, to ensure it is running successfully on different platforms.

4.3 Performance Testing

Performance testing was done to ensure application performs to software requirements and expectations

5. Item Pass/Fail Criteria

5.1 Approval Criteria

In order for a component to be approved, it has to meet the requirements for each module as described in part 3. of this test plan.

6. Test Deliverables

Testing report log as appendix “A” to this test plan.

7. References

Documents that support this test plan:

Document	Version	Release Date	Author	Major Changes
Software Management Project Plan	1.0	3/1/2015	Group Collaboration	Initial Document
Software Requirements Specifications	1.0	3/15/2015	Group Collaboration	Initial Document
Software Design Specifications	1.0	3/29/2015	Group Collaboration	Initial Document
Appendix “A” Testing Scenarios	1.0	04/19/2015	Group Collaboration	Initial Document
Appendix “B” Test Log	1.0	04/19/2015	Group Collaboration	Initial Document