
Test Plan Document

for

LIBRARY INFORMATION SYSTEM

Authors :

Nakul Aggarwal (19CS10044)

Rupinder Goyal (19CS10050)

Ashwamegh Rathore (19CS30009)

Instructors :

Prof. Sourangshu Bhattacharya

Prof. Abir Das

Owais Iqbal

**Indian Institute Of Technology Kharagpur
27 March 2021**

TEST PLAN OUTLINE

PREFACE

The document contains a detailed test plan for the *Library Information System* application to test all the *use cases*, *features* and *functionalities* offered by the application. The entire *test plan outline* is divided into 5 sections -- *Home Page*, *Librarian's Control Panel*, *Clerk's Control Panel*, *Member's Control Panel* and *Miscellaneous System Features*. Under each *testplan* the *test suites* are written in a *bold-face* and the *test cases* under each *test suite* are written under higher levels of indentation, *without bold-face*.

Read the description for each test case carefully to know the *input* and the *expected output* for that test case, and also the procedure that might have to be followed. The *testing team* is advised to clear the tables in all the databases that do not contain any *accounts* or *catalogue* information before implementing each test case.

1. Test Plan for HOME PAGE

1.1. Test Suite -- GUI DISPLAY

1.1.1. Available Modules

The *home page* should have four modules displayed with their roles/names clearly written on them (*Query Section*, *Log In As Librarian*, *Log In As Clerk*, *Log In As Member*)

1.1.2. Activity

All buttons/modules should be *clickable* and *active*.

1.1.3. Spacing & Aesthetics

Buttons and modules should be *centrally aligned*. Line spacing, padding and margins should be uniform and sufficient.

Font size must be bold and big enough to be read. *Font family/style* should be simple and comprehensive.

The *header* must be present and properly aligned.

1.2. Test Suite -- SIGN IN MODULES

1.2.1. GUI Aspects

Every *sign in* page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. *Password* must not be visible when typed in its respective entry bar.

1.2.2. Log In As Librarian

1.2.2.1. Invalid Login ID

An error message should pop up *without proceeding further*.

1.2.2.2. Member's Login ID

An error message should pop up *without proceeding further*.

1.2.2.3. Clerk's Login ID

An error message should pop up *without proceeding further*.

1.2.2.4. Incorrect Password

An error message should pop up *without proceeding further*.

1.2.2.5. Correct Librarian's Login ID and Password

Login button clickable. Login permitted. Enters the account.

1.2.3. Log In As Clerk

1.2.3.1. Invalid Login ID

An error message should pop up *without proceeding further*.

1.2.3.2. Member's Login ID

An error message should pop up *without proceeding further*.

1.2.3.3. Librarian's Login ID

An error message should pop up *without proceeding further*.

1.2.3.4. Incorrect Password

An error message should pop up *without proceeding further*.

1.2.3.5. Correct Clerk's Login ID and Password

Login button clickable. Login permitted. Enters the account.

1.2.4. Log In As Member

1.2.4.1. Invalid Login ID

An error message should pop up *without proceeding further*.

1.2.4.2. Librarian's Login ID

An error message should pop up *without proceeding further*.

1.2.4.3. Clerk's Login ID

An error message should pop up *without proceeding further*.

1.2.4.4. Incorrect Password

An error message should pop up *without proceeding further*.

1.2.4.5. Correct Member's Login ID and Password

Login button clickable. Login permitted. Enters the account.

1.2.5. Back Navigability

On each of the *login pages*, there should be a *Go Back* button that is *active* and *clickable* and navigates back to the *home page*.

1.3. Test Suite -- QUERY SECTION MODULE

1.3.1. GUI Aspects

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

1.3.2. Search by ISBN

1.3.2.1. Bad ISBN

Select *ISBN* as the search criteria. Enter a *non-existing* ISBN in the catalogue and *search*.

1.3.2.2. Good ISBN

Select *ISBN* as the search criteria. Enter an *ISBN* of the correct format that exists in the catalogue and *search*.

1.3.3. Search by Title

1.3.3.1. Primary Matches

Select *Title* as the search criteria. Enter a *title* that *exactly matches* the title of some book in the catalogue.

1.3.3.2. Secondary Matches

Select *Title* as the search criteria. Enter an arbitrary *title* that *does not match* the title of any book in the catalogue but might be present as a *substring* in some title. (Eg - "a")
Enter an arbitrary *title* to search by *case-insensitive matching* (Eg - "tHe")

1.3.4. Search by Author

1.3.4.1. Primary Matches

Select *Author* as the search criteria. Enter an *author* that *exactly matches* the author of some book in the catalogue.

1.3.4.2. Secondary Matches

Select *Author* as the search criteria. Enter an arbitrary *author* that *does not match* the author of any book in the catalogue but might be present as a *substring* in some author's name. (Eg - "a")
Enter an arbitrary *author* to search by *case-insensitive matching* (Eg - "jAc")

1.3.5. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *home page*.

2. Test Plan for LIBRARIAN'S CONTROL PANEL

2.1. Test Suite -- GUI DISPLAY

2.1.1. Available Modules

The *first page* should have all *nine modules* displayed with their roles/names clearly written on them and besides a *log out button*.

2.1.2. Activity

All buttons/modules should be *clickable* and *active*.

2.1.3. Spacing & Aesthetics

Buttons and modules should be *centrally aligned*. Line spacing, padding and margins should be uniform and sufficient.

Font size must be bold and big enough to be read. *Font family/style* should be simple and comprehensive.

The *header* must be present and properly aligned.

2.2. Test Suite -- LOG OUT

The *log out button* should be active and clickable. It should *successfully* log out the user every time the button is clicked without any *error/exception/warning/confirmation*. Log out should display the *home page*.

2.3. Test Suite -- REGISTRATION

2.3.1. GUI Aspects

2.3.1.1. Back Navigability

There should be a *Go Back button* that is *active* and *clickable* and navigates back to the *first page*.

2.3.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

2.3.2. Erroneous Inputs

2.3.2.1. Unspecified Fields

2.3.2.1.1. Member Type Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

2.3.2.1.2. Institute ID Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

2.3.2.1.3. **Name Not Specified**

An *error* with a suitable message should pop up *without proceeding further*.

2.3.2.1.4. **DOB Not Specified**

An *error* with a suitable message should pop up *without proceeding further*.

2.3.2.2. **Wrongly Specified Fields**

2.3.2.2.1. **Bad Institute ID**

If an account already exists for the specified *institute ID*, an *error* with a suitable message should pop up *without proceeding further*.

2.3.2.2.2. **Bad Date Of Birth**

The *Date of Birth* can never be in future. A *faculty member* must have an age of at least 25. A *non-faculty member* must have an age of at least 16. In these test cases, an *error* with a suitable message should pop up *without proceeding further*.

2.3.3. **Correct Inputs and Successful Registration**

2.3.3.1. **Register Button Working**

If all the *validity checks* are passed, the *register button* should be working and the registration should be successful. The *login ID/LMCN* of the member should appear in a pop up with a *success* message.

2.3.3.2. **Verify Registration**

The new *LIS* user should be able to *login as a member* into her account using the *login ID* and the *default password* (*login id* concatenated with an *underscore* and the *date of birth*).

The *librarian* should be able to see the *new member* in the *View All Members* module.

2.3.3.3. **Verify Databases at the Back-End**

A new record should be *inserted* in the appropriate table of *Accounts* database.

2.4. Test Suite -- DE-REGISTRATION

2.4.1. GUI Aspects

2.4.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.4.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

2.4.2. Erroneous Inputs

2.4.2.1. Unspecified Fields

2.4.2.1.1. LMCN Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

2.4.2.1.2. Name Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

2.4.2.2. Wrongly Specified Fields

2.4.2.2.1. Bad LMCN

If the *LMCN* of the *to-be-deregistered* user is not present in the *LIS's database*, an *error* with a suitable message should pop up *without proceeding further*.

2.4.2.2.2. Bad Name

If the name of the *member* associated with the specified *LMCN* does not match with the specified *name*, an *error* with a suitable message should pop up *without proceeding further*.

2.4.3. Correct Inputs but De-register Unsuccessful

2.4.3.1. Member has Returns Due

If the *member* had issued some books in the past, one or more of which she has not yet returned, then she cannot be *de-registered*. An *error* with a suitable message should pop up *without proceeding further*.

2.4.3.2. Member has Fines Due

If the *member* has some pending fines/penalties, then she cannot be *de-registered*. An *error* with a suitable message should pop up *without proceeding further*.

2.4.4. Correct Inputs and De-register Successful

2.4.4.1. De-Register Button Working

The *de-register button* is clickable without any error.

2.4.4.2. Warning / Confirmation

A warning stating the *consequences of de-registration* should appear before the task is actually performed.

2.4.4.3. Successful De-Registration

If the *warning is accepted* by the user, the member should be *successfully de-registered*.

2.4.4.4. Verify De-Registration

The de-registered *LIS* member should no longer be able to *login as a member* into her account using the *login ID* and the *password*.

The *librarian* would not be able to see the *de-registered member* in the *View All Members* module.

2.4.4.5. Verify Databases at the Back-End

The record of the de-registered member should be present in neither the *FacultyMembers* nor the *NonFacultyMembers* table of the *Accounts* database.

No *notifications* dedicated to the de-registered member are present in the *Messages* table of *Accounts* database.

No books *reserved* by the de-registered member are present in the *ReservedBooks* table of the *Library* database. (*Reservations, if any, for the de-registered member are terminated/expired.*)

No *history of issued books* is found in the *BooksIssuedInPast* table of the *Library* database for the *de-registered member* (*history is deleted*)

2.5. Test Suite -- UNPOPULAR BOOKS ANALYTICS

2.5.1. GUI Aspects

2.5.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.5.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. Input widget should be clearly *visible*. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

2.5.2. Unspecified Fields

If the *time span* is not specified, an error with a suitable message should pop up.

2.5.3. Relevance of Displayed Records

2.5.3.1. Statistics for Last One Year

Choose 1 as the input and *search*. A list of relevant books should appear with *necessary details* and especially the *last date on which it was issued*. Check the *relevance* of the results.

2.5.3.2. Statistics for the Last Three Years

Choose 3 as the input and *search*. A list of relevant books should appear with *necessary details* and especially the *last date on which it was issued*. Check the *relevance* of the results.

2.5.3.3. Statistics for Last Five Years

Choose 5 as the input and *search*. A list of relevant books should appear with *necessary details* and especially the *last date on which it was issued*. Check the *relevance* of the results.

2.6. Test Suite -- VIEW CATALOGUE

2.6.1. GUI Aspects

2.6.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.6.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. *No input fields should be there*. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

2.6.2. Relevance of Displayed Records

A list of *all books in the catalogue* should appear with all *necessary details* including *ISBN, title, author and language*. No *trigger/input/command* should be required to display the records.

2.7. Test Suite -- VIEW CURRENTLY ISSUED BOOKS

2.7.1. GUI Aspects

2.7.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.7.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. No *input fields* should be *there*. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

2.7.2. Relevance of Displayed Records

A list of *all books currently issued* should appear with all *necessary details* including the *ISBN of the issued book, the issuer's LMCN, date of issue, and due date of return*. No *trigger/input/command* should be required to display the records.

2.8. Test Suite -- VIEW BOOKS ISSUED IN PAST

2.8.1. GUI Aspects

2.8.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.8.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. No *input fields* should be *there*. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

2.8.2. Relevance of Displayed Records

A list of *all books issued in the past* should appear with all *necessary details* including the *ISBN of the issued book, the issuer's LMCN, the date of return, the penalty, and the status of payment of the penalty*. No *trigger/input/command* should be required to display the records.

2.9. Test Suite -- VIEW CURRENTLY OVER-DUE BOOKS

2.9.1. GUI Aspects

2.9.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.9.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. No *input fields* should be *there*. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

2.9.2. Relevance of Displayed Records

A list of *all books currently issued out* for which the *due date of return* is in the past should appear with all *necessary details* including *Issue ID*, *ISBN of the issued book*, *issuer's LMCN*, *date of issue*, *the current count of over-due days* and *current accumulated penalty*. No *trigger/input/command* should be required to display these records.

The current accumulated penalty should be proportional to the difference (in days) between the due date of return and the current date. For a difference of 10 days, the accumulated penalty will be *Rs. 50* and for a difference of 15 days the accumulated penalty will be *Rs. 75*. Check that the *proportionality constant* is 5.

2.10. Test Suite -- VIEW ALL MEMBERS

2.10.1. GUI Aspects

2.10.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.10.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. No *input fields* should be *there*. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

2.10.2. Relevance of Displayed Records

A list of *all members registered in the LIS* should appear with all necessary details including *LMCN, name, and current issue count*. No *trigger/input/command* should be required to display the records.

2.11. Test Suite -- CHANGE ACCOUNT'S PASSWORD

2.11.1. GUI Aspects

2.11.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

2.11.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

2.11.2. Erroneous Inputs

2.11.2.1. Unspecified Fields

2.11.2.1.1. Current Password not specified

An *error* with a suitable message should pop up *without proceeding further*.

2.11.2.1.2. New Password not specified

An *error* with a suitable message should pop up *without proceeding further*.

2.11.2.1.3. Confirm Password not specified

An *error* with a suitable message should pop up *without proceeding further*.

2.11.2.2. Wrongly Specified Fields

2.11.2.2.1. Current Password incorrect

An *error* with a suitable message should pop up *without proceeding further*.

2.11.2.2.2. New Password has No Characters

An *error* with a suitable message should pop up *without proceeding further*.

2.11.2.2.3. New & Confirm Password Don't Match

An *error* with a suitable message should pop up *without proceeding further*.

2.11.3. Correct Inputs and Change Password Successful

2.11.3.1. Change Password Button Working

Change Password button is clickable without raising any error. Password is successfully changed.

2.11.3.2. Verify Changed Password

2.11.3.2.1. Try Log In with Old Password

Log in should be unsuccessful.

2.11.3.2.2. Try Log In with New Password

Log in should be successful.

3. Test Plan for CLERK'S CONTROL PANEL

3.1. Test Suite -- GUI DISPLAY

3.1.1. Available Modules

The *first page* should have all *eight modules* displayed with their roles/names clearly written on them and besides a *log out button*.

3.1.2. Activity

All buttons/modules should be *clickable* and *active*.

3.1.3. Spacing & Aesthetics

Buttons and modules should be *centrally aligned*. Line spacing, padding and margins should be uniform and sufficient.

Font size must be bold and big enough to be read. *Font family/style* should be simple and comprehensive.

The *header* must be present and properly aligned.

3.2. Test Suite -- LOG OUT

The *log out* button should be active and clickable. It should *successfully* log out the user every time the button is clicked without any *error/exception/warning/confirmation*. Log out should display the *home page*.

3.3. Test Suite -- ISSUE A BOOK (FROM RACK)

3.3.1. GUI Aspects

3.3.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

3.3.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*.
Entry boxes in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.3.2. Erroneous Inputs

3.3.2.1. Unspecified Fields

3.3.2.1.1. ISBN Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.3.2.1.2. LMCN Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.3.2.2. Wrongly Specified Fields

3.3.2.2.1. Bad ISBN

If the *ISBN* of the book is not present in the *LIS's database*, an *error* with a suitable message should pop up *without proceeding further*.

3.3.2.2.2. Bad LMCN

If the *LMCN* of the *issuer* is not present in the *LIS's database*, an *error* with a suitable message should pop up *without proceeding further*.

3.3.3. Correct Inputs but Issue Unsuccessful

3.3.3.1. Member Already Reached Limit

An *error* with a suitable message should pop up *without proceeding further*.

3.3.3.2. No Copies Available Presently

If there are no copies of that book currently available on the rack to issue, an *error* with a suitable message should pop up *without proceeding further*.

3.3.4. Correct Inputs and Issue Successful

3.3.4.1. Issue Button Working

The *issue button* is clickable without any error.

3.3.4.2. Verify Changes Through the Librarian's Account

When the *Librarian* logs into her account, she should be able to see the newly issued book in the *Show All Currently Issued Books* module.

3.3.4.3. Verify Databases at the Back-End

The *count of available copies* of the issued book should be decremented by 1 in the *Catalogue* table of the *Library* database.

The *count of issued books by the member* should be incremented by 1 in the *concerned* table of *Accounts* database.

A new record should be inserted in the *CurrentlyIssuedBooks* table of *Library* database.

3.3.4.4. **Verify Changes Through Member's Account**

3.3.4.4.1. **Check Notifications**

The *member* should have received a notification stating the details of the *issued book* along with the *due date of return*.

3.3.4.4.2. **Check Module(s)**

In *View Currently Issued Books* module, the *member* should be able to see the details of the newly issued book.

3.4 **Test Suite -- RESERVE A BOOK**

3.4.1. **GUI Aspects**

3.4.1.1. **Back Navigability**

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

3.4.1.2. **Spacing & Aesthetics**

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.4.2. **Erroneous Inputs**

3.4.2.1. **Unspecified Fields**

3.4.2.1.1. **ISBN Not Specified**

An *error* with a suitable message should pop up *without proceeding further*.

3.4.2.1.2. **LMCN Not Specified**

An *error* with a suitable message should pop up *without proceeding further*.

3.4.2.2. **Wrongly Specified Fields**

3.4.2.2.1. **Bad ISBN**

If the *ISBN* of the book is not present in the *LIS's database*, an *error* with a suitable message should pop up *without proceeding further*.

3.4.2.2.2. Bad LMCN

If the *LMCN* of the *issuer* is not present in the *LIS's database*, an *error* with a suitable message should pop up *without proceeding further*.

3.4.3. Correct Inputs but Reservation Unsuccessful

3.4.3.1. No Relevant Issued Book

If either all the issued books with the specified *ISBN* are already reserved or if there are no issued books at all with the specified *ISBN*, an *error* with a suitable message should pop up *without proceeding further*.

3.3.4. Correct Inputs and Reservation Successful

3.3.4.1. Reserve Button Working

The *reserve button* is clickable without any error.

3.3.4.3. Verify Databases at the Back-End

The status of the record in the *CurrentlyIssuedBooks* table of *Library* database corresponding to the issued book that is reserved should be stated as *reserved*.

The new reservation should be inserted in the *ReservedBooks* table of the *Library* database.

3.3.4.4. Verify Changes Through Member's Account

3.3.4.4.1. Check Notifications

The *member* should have received a notification stating the details of the *reservation* along with the *Reservation ID* and *Reservation Key*.

3.5. Test Suite -- ISSUE A RESERVED BOOK

3.5.1. GUI Aspects

3.5.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

3.5.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*.
Entry boxes in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.5.2. Erroneous Inputs

3.5.2.1. Unspecified Fields

3.5.2.1.1. Reservation ID not specified

An *error* with a suitable message should pop up *without proceeding further*.

3.5.2.1.2. Reservation key not specified

An *error* with a suitable message should pop up *without proceeding further*.

3.5.2.2. Wrongly Specified Fields

3.5.2.2.1. Bad Reservation ID

If the provided *Reservation ID* is not present in the *LIS's database*, an *error* with a suitable message should pop up *without proceeding further*.

3.5.2.2.2. Bad Reservation Key

If the provided *Reservation Key* does not match with the key corresponding to the specified *Reservation ID*, an *error* with a suitable message should pop up *without proceeding further*.

3.5.3. Correct Inputs but Issue Unsuccessful

3.5.3.1. Member Already Reached Limit

An *error* with a suitable message should pop up *without proceeding further*.

3.5.3.2. Reserved Book Not Yet Returned

If the *issued book* that is reserved is not yet returned by the issuer, an *error* with a suitable message should pop up *without proceeding further*.

3.5.4. Correct Inputs and Issue Successful

3.5.4.1. Issue Button Working

The *issue button* is clickable without any error.

3.5.4.2. Verify Changes Through the Librarian's Account

When the *Librarian* logs into her account, she should be able to see the newly issued book in the *Show All Currently Issued Books* module.

3.5.4.3. Verify Databases at the Back-End

The *count of issued books by the member* should be incremented by 1 in the *concerned* table of *Accounts* database.

The *reservation* should be deleted from the *ReservedBooks* table in the *Library* database.

A new record should be inserted in the *CurrentlyIssuedBooks* table of *Library* database.

3.5.4.4. Verify Changes Through Member's Account

3.3.4.4.1. Check Notifications

The *member* should have received a notification stating the details of the *issued book* along with the *due date of return*.

3.3.4.4.2. Check Module(s)

In *View Currently Issued Books* module, the *member* should be able to see the details of the newly issued book.

3.6. Test Suite -- RETURN A BOOK

3.6.1. GUI Aspects

3.6.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

3.6.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.6.2. Erroneous Inputs

3.6.2.1. Unspecified Fields

3.6.2.1.1. Issue ID Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.6.2.2. Wrongly Specified Fields

3.6.2.2.1. Bad Issue ID

If the specified *Issue ID* is not present in the database, an *error* with a suitable message should pop up *without proceeding further*.

3.6.3. Correct Inputs and Return Successful

3.6.3.1. Return Button Working

The *issue button* is clickable without any error.

3.6.3.2. Verify Changes Through the Librarian's Account

When the *Librarian* logs into her account, she should *not* be able to see the *returned* in the *Show All Currently Issued Books* module.

When the *Librarian* logs into her account, she should be able to see the *returned* book in the *Show All Books Issued In Past* module.

3.6.3.3. Verify Databases at the Back-End

The *count of issued books by the member* should be decremented by 1 in the *concerned* table of *Accounts* database.

The *count of available copies* of the returned book should be incremented by 1 (if the book is not reserved) in the *Catalogue* table of the *Library* database.

The respective record of the *returned book* should be deleted from the *CurrentlyIssuedBooks* table in the *Library* database.

The details of the *returned book* should be inserted in the *BooksIssuedInPast* table of the *Library* database.

If the book is reserved, the corresponding *reservation record* in *ReservedBooks* table of *Library* database should be updated and a notification should be sent to the *reserver member*.

3.6.3.4. Verify Changes Through Member's Account

3.6.3.4.1. Check Notifications

The *member* should have received a notification confirming the successful return of the book.

If the book that is returned was *reserved* by a *member*, she should be notified about the return of the book.

3.6.3.4.2. Check Module(s)

In *View Currently Issued Books* module, the *member* should not be able to see the details of the returned book.

In *View Books Issued In Past* module, the *member* should be able to see the details of the returned book.

3.6.4. Ask To Pay Fine

3.6.4.1. Relevant Pop-Up Appearance

3.6.4.1.1. Book Returned Before Due Date

If the book was returned before the *due date of return*, no pop-up should appear and the module should be automatically *exited* to the *first page*.

3.6.4.1.2. Book Returned After Due Date

If the book was returned after the *due date of return* then a pop-up should appear that asks to *navigate* directly to the *Pay Fine* module.

3.6.4.2. Navigation to Pay Fine Module

If the *pop-up* question is responded in affirmation then the *user* should be navigated to the *Pay Fine* section. Otherwise, the module should be *exited* to the *first page*.

3.7. Test Suite -- PAY FINE

3.7.1. GUI Aspects

3.7.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

3.7.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.7.2. Erroneous Inputs

3.7.2.1. Unspecified Fields

3.7.2.1.1. Issue ID Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.7.2.2. Wrongly Specified Fields

3.7.2.2.1. Bad Issue ID

If the specified *Issue ID* is not present in the *LIS's database*, an *error* with a suitable

message should pop up *without proceeding further*.

3.7.3. Correct Inputs but Fine Payment Unsuccessful

3.7.3.1. Fine not pending

If the penalty associated with the specified *Issue ID* is already paid (not pending) or was zero, an *information pop-up* with a suitable message should appear *without proceeding further*.

3.7.4. Correct Inputs and Fine Payment Successful

3.7.4.1. Pay Fine Button Working

The *pay fine button* is clickable without any pop-up.

3.7.4.2. Verify Databases at the Back-End

The status of the *pending fine* corresponding to the specified *Issue ID* in the *BooksIssuedInPast* table of the *Library* database should be changed to *paid*.

3.7.4.3. Verify Changes Through Member's Account

3.6.3.4.1. Check Notifications

The *member* should have received a notification confirming the successful payment of the fine/penalty.

3.8. Test Suite -- ADD BOOK TO CATALOGUE

3.8.1. GUI Aspects

3.8.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

3.8.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.8.2. Erroneous Inputs

3.8.2.1. Unspecified Fields

3.8.2.1.1. ISBN Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.8.2.1.2. Title Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.8.2.1.3. Author Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.8.2.1.4. Language Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.8.2.1.5. Rack No. Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.8.2.1.6. Number of Copies Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.8.2.2. Wrongly Specified Fields**3.8.2.2.1. Bad ISBN**

If the book having the specified *ISBN* is already present in the catalogue/database, an *error* with a suitable message should pop up *without proceeding further*.

3.8.2.2.1. Bad Number Of Copies

If the specified test has a *non-digit character* or the number to which the test evaluates is *non-positive*, an *error* with a suitable message should pop up *without proceeding further*.

3.8.3. Correct Inputs and Insertion Successful**3.3.4.1. Add Button Working**

The *add button* is clickable without any error.

3.3.4.2. Verify Changes In Catalogue

When the *Librarian* logs into her account, she should be able to see the newly added book in the *Show Catalogue* module.

The newly added book should be visible in the *Query Section* module on the *home page*.

3.3.4.3. Verify Databases at the Back-End

The record corresponding to the newly added book should be inserted into the *Catalogue* table of the *Library* database.

3.9. Test Suite -- REMOVE BOOK FROM CATALOGUE

3.9.1. GUI Aspects

3.9.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

3.9.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.9.2. Erroneous Inputs

3.9.2.1. Unspecified Fields

3.9.2.1.1. ISBN Not Specified

An *error* with a suitable message should pop up *without proceeding further*.

3.9.2.2. Wrongly Specified Fields

3.9.2.2.1. Bad ISBN

If the *ISBN* of the book is not present in the *LIS's database*, an *error* with a suitable message should pop up *without proceeding further*.

3.9.3. Correct Inputs but Removal Unsuccessful

3.9.3.1. Copy(ies) Are Issued Out

If there is at least one copy of the to-be-removed book that is issued out, it cannot be removed from the catalogue. In this test case, an *error* with a suitable message should pop up *without proceeding further*.

3.9.3.2. Copy(ies) Are Reserved

If there is at least one copy of the to-be-removed book that is reserved by a *member*, it cannot be removed from the catalogue. In this test case, an *error* with a suitable message should pop up *without proceeding further*.

3.9.4. Correct Inputs and Removal Successful

3.9.4.1. Remove Button Working

The *issue button* is clickable without any error.

3.9.4.2. Warning / Confirmation

A warning stating the *consequences of removing a book from the catalogue* should appear before the task is actually performed.

3.9.4.3. *Successful Removal*

If the *warning* is accepted by the *clerk*, the book should be successfully removed from the catalogue.

3.9.4.4. *Verify Changes Through the Librarian's Account*

When the *Librarian* logs into her account, she should no longer be able to see the *removed* book in the *View Catalogue* module.

3.9.4.5. *Verify Databases at the Back-End*

The record of the book should be deleted from the *Catalogue* table of the *Library* database.

3.10. Test Suite -- **CHANGE ACCOUNT'S PASSWORD**

3.10.1. *GUI Aspects*

3.10.1.1. *Back Navigability*

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first* page.

3.10.1.2. *Spacing & Aesthetics*

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

3.10.2. *Erroneous Inputs*

3.10.2.1. *Unspecified Fields*

3.10.2.1.1. *Current Password not specified*

An *error* with a suitable message should pop up *without proceeding further*.

3.10.2.1.2. *New Password not specified*

An *error* with a suitable message should pop up *without proceeding further*.

3.10.2.1.3. *Confirm Password not specified*

An *error* with a suitable message should pop up *without proceeding further*.

3.10.2.2. *Wrongly Specified Fields*

3.10.2.2.1. *Current Password incorrect*

An *error* with a suitable message should pop up *without proceeding further*.

3.10.2.2.2. *New Password has No Characters*

An *error* with a suitable message should pop up *without proceeding further*.

3.10.2.2.3. *New & Confirm Password Don't Match*

An error with a suitable message should pop up *without proceeding further*.

3.10.3. *Correct Inputs and Change Password Successful*

3.10.3.1. *Change Password Button Working*

Change Password button is clickable without raising any error. Password is successfully changed.

3.10.3.2. *Verify Changed Password*

3.10.3.2.1. *Try Log In with Old Password*

Log in should be unsuccessful.

3.10.3.2.2. *Try to Log In with New Password*

Log in should be successful.

4. **Test Plan for MEMBER'S CONTROL PANEL**

4.1. **Test Suite -- GUI DISPLAY**

4.1.1. *Available Modules*

The *first page* should have all *four modules* displayed with their roles/names clearly written on them and besides a *log out button*.

4.1.2. *Activity*

All buttons/modules should be *clickable* and *active*.

4.1.3. *Spacing & Aesthetics*

Buttons and modules should be *centrally aligned*. Line spacing, padding and margins should be uniform and sufficient.

Font size must be bold and big enough to be read. *Font family/style* should be simple and comprehensive.

The *header* must be present and properly aligned.

4.2. **Test Suite -- LOG OUT**

The *log out* button should be active and clickable. It should *successfully* log out the user every time the button is clicked without any *error/exception/warning/confirmation*. Log out should display the *home page*.

4.3. **Test Suite -- VIEW NOTIFICATIONS**

4.3.1. GUI Aspects

4.3.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

4.3.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. The screen must be appropriately partitioned into the *panel that shows the messages* and the *panel that displays the full selected message*. The *font size* of the displayed message should be easily readable. Both the panels should be *scrollable* when the displayed content/notifications are large in size/number.

4.3.2. Test Notification Activity

Each of the following tasks should result in a new notification in the account of the member with whom the task is concerned.

4.3.2.1. Book Issued

Issue a book for a *member M* from a *clerk's* account. Check the notifications of *M*.

4.3.2.2. Book Returned

Return this book for the *member M* from a *clerk's* account. Check the notifications of *M*.

4.3.2.3. Book Reserved

Issue a book for a *member N* from a *clerk's* account. Reserve this book for the *member M* from a *clerk's* account. Check the notifications of *M*.

4.3.2.4. Reserved Book Returned by Former Issuer

Return the book issued by *N* from a *clerk's* account. Check the notifications of *M*.

4.3.2.5. Penalty Cleared

Introduce an external stimulus by inserting a *test record* in the *BooksIssuedInPast* table (*Library* database) with the details of *member M*, placeholder values for *ISBN* and *Issue ID*, *non-zero penalty* and *unpaid* status for the penalty. Pay penalty from a *clerk's* account for this particular record. Check notification of *M*. Delete *test record* and the *notification* from the database.

4.3.2.6. Reservation Expired

Insert a *test record* in the *Catalogue* table of the *Library* database through *Add New Book* module in a *clerk's* account, using placeholders as inputs. Introduce an external stimulus by inserting a *test record* in the *ReservedBooks* table (*Library* database) with the details of *member M*, *ISBN* of the *test book*, *date of former return* 'x' days before the *current* date and placeholder values for other fields.

4.3.2.6.1. Boundary Case

Choose 'x' as 8. Check notifications of *M*. Delete *test records* and the *notification* from the database.

4.3.2.6.2. Remote Case

Choose 'x' as 15. Check notifications of *M*. Delete *test records* and the *notification* from the database.

4.3.3. Select And Display

The *notification* messages should be *clickable*. Upon selecting and clicking, the content inside the message should be displayed on the *side display panel*.

4.4. Test Suite -- VIEW CURRENTLY ISSUED BOOKS

4.4.1. GUI Aspects

4.4.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

4.4.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. No *input fields* should be there. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

4.4.2. Relevance of Displayed Records

A list of *all books currently issued* by that particular *member* should appear with all *necessary details* including the *ISBN* of the *issued book*, *date of issue*, and *due date of return*. The *issuer's LMCN* should not appear. No *trigger/input/command* should be required to display the records.

4.5. Test Suite -- VIEW BOOKS ISSUED IN PAST

4.5.1. GUI Aspects

4.5.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

4.5.1.2. Spacing & Aesthetics

Every *widget* on the page must be *neatly aligned*. No *input fields* should be *there*. Enough space is there for the search results. Proper spacing is imperative. The search results are *scrollable* if they are large in number.

4.5.2. Relevance of Displayed Records

A list of *all books issued in the past* by that particular *member* should appear with all *necessary details* including the *ISBN of the issued book*, *date of return*, *penalty*, and *the status of payment of the penalty*. The *issuer's LMCN* should not appear. No *trigger/input/command* should be required to display the records.

4.6. Test Suite -- CHANGE ACCOUNT'S PASSWORD

4.6.1. GUI Aspects

4.6.1.1. Back Navigability

There should be a *Go Back* button that is *active* and *clickable* and navigates back to the *first page*.

4.6.1.2. Spacing & Aesthetics

Every *widget* on the page must be *centrally aligned*. *Entry boxes* in the form should be wide enough to accept the entire input. Proper spacing is imperative.

4.6.2. Erroneous Inputs

4.6.2.1. Unspecified Fields

4.6.2.1.1. Current Password not specified

An *error* with a suitable message should pop up *without proceeding further*.

4.6.2.1.2. New Password not specified

An *error* with a suitable message should pop up *without proceeding further*.

4.6.2.1.3. Confirm Password not specified

An *error* with a suitable message should pop up *without proceeding further*.

4.6.2.2. *Wrongly Specified Fields*

4.6.2.2.1. *Current Password incorrect*

An error with a suitable message should pop up *without proceeding further*.

4.6.2.2.2. *New Password has No Characters*

An error with a suitable message should pop up *without proceeding further*.

4.6.2.2.3. *New & Confirm Password Don't Match*

An error with a suitable message should pop up *without proceeding further*.

4.6.3. *Correct Inputs and Change Password Successful*

4.6.3.1. *Change Password Button Working*

Change Password button is clickable without raising any error. Password is successfully changed.

4.6.3.2. *Verify Changed Password*

4.6.3.2.1. *Try Log In with Old Password*

Log in should be unsuccessful.

4.6.3.2.2. *Try to Log In with New Password*

Log in should be successful.

5. *Test Plan for MISCELLANEOUS SYSTEM FEATURES*

5.1. *Test Suite -- REMINDERS*

Reminders should be received by the *members* in the same inbox where the *notifications* are received. Stimulate the following *test cases* and check if the reminders appear.

5.1.1. *Approaching Due Date of Return*

Issue a book for the member *M* from the *clerk's* account. Stimulate the test case by adjusting the *due date of return* for the issued book to 2 days *after* the current date in the *CurrentlyIssuedBooks* table of the *Library* database.

5.1.1.1. *Appearance of Reminder*

Check the inbox of *member M*. A *reminder of the approaching due date of return* should appear.

5.1.1.2. *Disappearance of Reminder*

Return this book for *member M* through the *clerk's* account. The same reminder should no longer be there in the inbox of *member M*.

Delete *test record(s)* / *notification(s)* from the database.

5.1.2. Pending Return of an Issued Book

Issue a book for the member *M* from the *clerk's* account. Stimulate the test case by adjusting the *due date of return* for the issued book to 2 days *before* the current date in the *CurrentlyIssuedBooks* table of the *Library* database.

5.1.2.1. Appearance of Reminder

Check the inbox of *member M*. A *reminder of the pending return of the book* should appear.

5.1.2.2. Disappearance of Reminder

Return this book for *member M* through the *clerk's* account. The same reminder should no longer be there in the inbox of *member M*.

Delete *test record(s)* / *notification(s)* from the database.

5.1.3. Reserved Book Waiting

Issue a book for a *member N* from a *clerk's* account. Reserve this book for the *member M* from a *clerk's* account. Return the book issued by *N* from a *clerk's* account. Stimulate the test case by adjusting the *date of former return* in the *ReservedBooks* table of the *Library* database to 2 days before the current date.

5.1.3.1. Appearance of Reminder

Check the inbox of *member M*. A *reminder of the reserved book waiting to be officially issued* should appear.

5.1.3.2. Disappearance of Reminder

Issue this reserved book for *member M* through the *clerk's* account. The same reminder should no longer be there in the inbox of *member M*.

Delete *test record(s)* / *notification(s)* from the database.

5.1.4. Fine Payment Pending

Introduce an external stimulus by inserting a *test record* in the *BooksIssuedInPast* table (*Library* database) with the details of *member M*, placeholder values for *ISBN* and *Issue ID*, *non-zero penalty* and *unpaid* status for the penalty.

5.1.4.1. Appearance of Reminder

Check the inbox of *member M*. A reminder of pending payment of penalty should appear.

5.1.4.2. Disappearance of Reminder

Pay penalty from a *clerk's* account for this particular record. The same reminder should no longer be there in the inbox of *member M*.

Delete *test record(s)* / *notification(s)* from the database.

5.2. Test Suite -- AUTOMATIC RESERVATION CANCELLATION

5.2.1. Stimulate the Test Case

Scale the period of the *thread that checks for expiration of reservations* down to a shorter span, like 20-30 minutes. Insert a *test record* in the *Catalogue* table of the *Library* database through *Add New Book* module in a *clerk's* account, using placeholders as inputs. Introduce an external stimulus by inserting a *test record* in the *ReservedBooks* table (*Library* database) with the details of *member M*, *ISBN* of the *test book*, *date of former return* 'x' days before the *current* date and placeholder values for other fields. Introduce these test records when a new period of the thread has just started.

5.2.2. Post Stimulation State of Reminders

Check the *inbox* of *member M* and ensure that there are no recent *notifications* for cancellation of reservation for the *test book*.

5.2.3. Automatic Cancellation

5.2.3.1. Boundary Case

Choose the value of 'x' as 6. When the period of the thread gets over, the reservation should get automatically expired.

5.2.3.2. Remote Case

Choose the value of 'x' as 10. When the period of the thread gets over, the reservation should get automatically expired.

5.2.4. Verification

In both the *test cases*, verify the *automatic expiry* through the following test.

5.2.3.1. Verify Through Member's Inbox

Member *M* should have received a notification regarding the cancellation of reservation.

5.2.3.2. Verify Databases at the Back-End

The number of available copies of the *test book* reflected in the *Catalogue* table of the *Library* database should be *incremented by 1* post the cancellation of reservation.

The record corresponding to the reservation of the *test book* by the *test member M* in the *ReservedBooks* table of the *Library* database should be deleted.

Delete *test record(s) / notification(s)* from the *database(s)*.