

**Test Plan Document**

# *for*

**Food Delivery Service Management System**

## *Authors :* Satvik Bansal (19CS10053) Anish Sofat (19CS10011)

***Aryan Aggarwal (19CS30009)***

***Instructors : Prof. Sourangshu Bhattacharya***

***Prof. Abir Das***

***Uppada Vishnu***

**Indian Institute Of Technology Kharagpur**

**26 March 2021**

**TEST PLAN OUTLINE**

# *MOTIVE*

1. ***Test Plan for HOME PAGE***

## *Test Suite -- GUI DISPLAY*

* + 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. *Activity*

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

* + 1. *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.

## *Test Suite -- SIGN IN MODULES*

* + 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. *Log In As Librarian*
       1. *Invalid Login ID*

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

* + - 1. *Member’s Login ID*

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

* + - 1. *Clerk’s Login ID*

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

* + - 1. *Incorrect Password*

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

* + - 1. *Correct Librarian’s Login ID and Password*

*Login button* clickable*.* Login permitted. Enters the account.

* + 1. *Log In As Clerk*
       1. *Invalid Login ID*

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

* + - 1. *Member’s Login ID*

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

* + - 1. *Librarian’s Login ID*

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

* + - 1. *Incorrect Password*

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

* + - 1. *Correct Clerk’s Login ID and Password*

*Login button* clickable*.* Login permitted. Enters the account.

* + 1. *Log In As Member*
       1. *Invalid Login ID*

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

* + - 1. *Librarian’s Login ID*

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

* + - 1. *Clerk’s Login ID*

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

* + - 1. *Incorrect Password*

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

* + - 1. *Correct Member’s Login ID and Password*

*Login button* clickable*.* Login permitted. Enters the account.

* + 1. *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*.

## *Test Suite -- QUERY SECTION MODULE*

* + 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. *Search by ISBN*
       1. *Bad ISBN*

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

* + - 1. *Good ISBN*

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

* + 1. *Search by Title*
       1. *Primary Matches*

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

* + - 1. *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”*)

* + 1. *Search by Author*
       1. *Primary Matches*

Select *Author* as the search criteria. Enter an *author* that

*exactly matches* the author of some book in the catalogue.

* + - 1. *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”*)

* + 1. *Back Navigability*

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

# *Test Plan for LIBRARIAN’S CONTROL PANEL*

## *Test Suite -- GUI DISPLAY*

* + 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*.

* + 1. *Activity*

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

* + 1. *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.

## *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*.

## *Test Suite -- REGISTRATION*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *Member Type Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Institute ID Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Name Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *DOB Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - 1. *Wrongly Specified Fields*
         1. *Bad Institute ID*

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

* + - * 1. *Bad Date Of Birth*

*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 suitable message should pop up *without proceeding further*.

* + 1. *Correct Inputs and Successful Registeration*
       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.

* + - 1. *Verify Registeration*

The new *LIS* user should be able to *login as 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.

* + - 1. *Verify Databases at the Back-End*

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

## *Test Suite -- DE-REGISTRATION*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*

If the *LMCN* of *to-be-deregistered* user is not specified, an *error* with suitable message should pop up *without proceeding further*.

* + - 1. *Wrongly Specified Fields*

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

If the *LMCN exists* but does not belong to a *member*, an *error* with suitable message should pop up *without proceeding further*.

* + 1. *Correct Inputs but De-register Unsuccessful*
       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 suitable message should pop up *without proceeding further*.

* + - 1. *Member has Fines Due*

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

* + 1. *Correct Inputs and De-register Successful*
       1. *De-Register Button Working*

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

* + - 1. *Warning / Confirmation*

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

* + - 1. *Successful De-Registration*

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

* + - 1. *Verify De-Registeration*

The de-registered *LIS* member should no longer be able to *login as 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 not 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 *Messages* table of *Accounts* database. No books *reserved* by the de-registered member are present in *ReservedBooks* table of *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 *Library* database for the *de-registered member* (*history is deleted*)

## *Test Suite -- UNPOPULAR BOOKS ANALYTICS*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Relevance of Displayed Records*
       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.

* + - 1. *Statistics for 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.

* + - 1. *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.

## *Test Suite -- VIEW CATALOGUE*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *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.

## *Test Suite -- VIEW CURRENTLY ISSUED BOOKS*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Relevance of Displayed Records*

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

## *Test Suite -- VIEW BOOKS ISSUED IN PAST*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Relevance of Displayed Records*

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

## *Test Suite -- VIEW CURRENTLY OVER-DUE BOOKS*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *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, 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.*

## *Test Suite -- VIEW ALL MEMBERS*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *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.

## *Test Suite -- CHANGE ACCOUNT’S PASSWORD*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *Current Password not specified*

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

* + - * 1. *New Password not specified*

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

* + - * 1. *Confirm Password not specified*

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

* + - 1. *Wrongly Specified Fields*
         1. *Current Password incorrect*

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

* + - * 1. *New Password has No Characters* An *error* with suitable message should pop up *without proceeding further*.
        2. *New & Confirm Password Don’t Match*

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

* + 1. *Correct Inputs and Change Password Successful*
       1. *Change Password Button Working*

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

* + - 1. *Verify Changed Password*
         1. *Try Log In with Old Password*

*Log in* should be unsuccessful.

* + - * 1. *Try Log In with New Password*

*Log in* should be successful.

# *Test Plan for CLERK’S CONTROL PANEL*

## *Test Suite -- GUI DISPLAY*

* + 1. *Available Modules*

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

* + 1. *Activity*

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

* + 1. *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.

## *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*.

## *Test Suite -- ISSUE A BOOK (FROM RACK)*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *ISBN Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *LMCN Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - 1. *Wrongly Specified Fields*
         1. *Bad ISBN*

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

* + - * 1. *Bad LMCN*

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

If the *LMCN exists* but does not belong to a *member*, an *error* with suitable message should pop up *without proceeding further*.

* + 1. *Correct Inputs but Issue Unsuccessful*
       1. *Member Already Reached Limit*

An *error* with 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 to issue, an *error* with suitable message should pop up *without proceeding further*.

* + 1. *Correct Inputs and Issue Successful*
       1. *Issue Button Working*

The *issue button* is clickable without any error.

* + - 1. *Verify Changes Through 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.

* + - 1. *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 *Library* database.

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

* + - 1. *Verify Changes Through Member’s Account*
         1. *Check Notifications*

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

* + - * 1. *Check Module(s)*

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

## *Test Suite -- RESERVE A BOOK*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *ISBN Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *LMCN Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - 1. *Wrongly Specified Fields*
         1. *Bad ISBN*

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

* + - * 1. *Bad LMCN*

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

If the *LMCN exists* but does not belong to a *member*, an *error* with suitable message should pop up *without proceeding further*.

* + 1. *Correct Inputs but Reservation Unsuccessful*
       1. *No Copy Issued Out*

If no copy of the book with the entered *ISBN* is issued out, there is no *issued book* to be reserved. In this test case, an error with suitable message should pop up without proceeding further.

If all issued copies of the book with the entered *ISBN* are already reserved, there is no *available* reservation. In this test case, an error with suitable message should pop up without proceeding further.

* + 1. *Correct Inputs and Reservation Successful*
       1. *Reserve Button Working*

The *reserve button* is clickable without any error.

* + - 1. *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 *Library* database.

* + - 1. *Verify Changes Through Member’s Account*
         1. *Check Notifications*

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

## *Test Suite -- ISSUE A RESERVED BOOK*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *Reservation ID not specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Reservation key not specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - 1. *Wrongly Specified Fields*
         1. *Bad Reservation ID*

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

* + - * 1. *Bad Reservation Key*

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

* + 1. *Correct Inputs but Issue Unsuccessful*
       1. *Member Already Reached Limit*

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

* + - 1. *Reserved Book Not Yet Returned*

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

* + 1. *Correct Inputs and Issue Successful*
       1. *Issue Button Working*

The *issue button* is clickable without any error.

* + - 1. *Verify Changes Through 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.

* + - 1. *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 *Library* database.

* + - 1. *Verify Changes Through Member’s Account*
         1. *Check Notifications*

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

* + - * 1. *Check Module(s)*

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

## *Test Suite -- RETURN A BOOK*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *Issue ID Not Specified*

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

* + - 1. *Wrongly Specified Fields*
         1. *Bad Issue ID*

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

* + 1. *Correct Inputs and Return Successful*
       1. *Return Button Working*

The *issue button* is clickable without any error.

* + - 1. *Verify Changes Through 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.

* + - 1. *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 in the *Catalogue* table of *Library* database.

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

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

* + - 1. *Verify Changes Through Member’s Account*
         1. *Check Notifications*

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

* + - * 1. *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.

* + 1. *Ask To Pay Fine*
       1. *Relevant Pop-Up Appearance*
          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*.

* + - * 1. *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.

* + - 1. *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*.

## *Test Suite -- PAY FINE*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *Issue ID Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - 1. *Wrongly Specified Fields*
         1. *Bad Issue ID*

If the specifed *Issue ID* is not present in the *LIS’s database*, an *error* with suitable message should pop up *without proceeding further*.

* + 1. *Correct Inputs but Fine Payment Unsuccessful*
       1. *Fine not pending*

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

* + 1. *Correct Inputs and Fine Payment Successful*
       1. *Pay Fine Button Working*

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

* + - 1. *Verify Databases at the Back-End*

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

* + - 1. *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 fine/penalty.

## *Test Suite -- ADD BOOK TO CATALOGUE*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *ISBN Not Specifed*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Title Not Specifed*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Author Not Specifed*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Language Not Specifed*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Rack No. Not Specifed*

An *error* with suitable message should pop up

*without proceeding further*.

* + - * 1. *Number of Copies Not Specifed*

An *error* with suitable message should pop up

*without proceeding further*.

* + - 1. *Wrongly Specified Fields*
         1. *Bad ISBN*

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

* + 1. *Correct Inputs and Insertion Successful*
       1. *Add Button Working*

The *add button* is clickable without any error.

* + - 1. *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*.

* + - 1. *Verify Databases at the Back-End*

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

## *Test Suite -- REMOVE BOOK FROM CATALOGUE*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *ISBN Not Specified*

An *error* with suitable message should pop up

*without proceeding further*.

* + - 1. *Wrongly Specified Fields*
         1. *Bad ISBN*

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

* + 1. *Correct Inputs but Removal Unsuccessful*
       1. *Copy(ies) Are Issued Out*

If there is atleast 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 suitable message should pop up *without proceeding further*.

* + - 1. *Copy(ies) Are Reserved*

If there is atleast 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 suitable message should pop up *without proceeding further*.

* + 1. *Correct Inputs and Removal Successful*
       1. *Remove Button Working*

The *issue button* is clickable without any error.

* + - 1. *Warning / Confirmation*

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

* + - 1. *Successful De-Registration*

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

* + - 1. *Verify Changes Through 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.

* + - 1. *Verify Databases at the Back-End*

The record of the book should be deleted from the

*Catalogue* table of *Library* database.

## *Test Suite -- CHANGE ACCOUNT’S PASSWORD*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *Current Password not specified*

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

* + - * 1. *New Password not specified*

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

* + - * 1. *Confirm Password not specified*

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

* + - 1. *Wrongly Specified Fields*
         1. *Current Password incorrect*

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

* + - * 1. *New Password has No Characters* An *error* with suitable message should pop up *without proceeding further*.
        2. *New & Confirm Password Don’t Match*

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

* + 1. *Correct Inputs and Change Password Successful*
       1. *Change Password Button Working*

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

* + - 1. *Verify Changed Password*
         1. *Try Log In with Old Password*

*Log in* should be unsuccessful.

* + - * 1. *Try Log In with New Password*

*Log in* should be successful.

# *Test Plan for MEMBER’S CONTROL PANEL*

## *Test Suite -- GUI DISPLAY*

* + 1. *Available Modules*

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

* + 1. *Activity*

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

* + 1. *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.

## *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*.

## *Test Suite -- VIEW NOTIFICATIONS*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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 is large in size/number.

* + 1. *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.

* + - 1. *Book Issued*

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

* + - 1. *Book Returned*

Return this book for the *member M* from a *clerk’s*

account. Check the notifications of *M*.

* + - 1. *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*.

* + - 1. *Reserved Book Returned by Former Issuer*

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

* + - 1. *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.

* + - 1. *Reservation Expired*

Insert a *test record* in the *Catalogue* table of *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*, *date of former return ‘x’* days before *current* date and placeholder values for other fields.

* + - * 1. *Boundary Case*

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

* + - * 1. *Remote Case*

Choose *‘x’* as *15*. Check notifications of

*M*. Delete *test records* and the

*notification* from the database.

* + 1. *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*.

## *Test Suite -- VIEW CURRENTLY ISSUED BOOKS*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Relevance of Displayed Records*

A list of *all books currently issued* by that particular *member* should appear with all *necessary details* including *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.

## *Test Suite -- VIEW BOOKS ISSUED IN PAST*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Relevance of Displayed Records*

A list of *all books issued in the passed* by that particular *member* should appear with all *necessary details* including *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.

## *Test Suite -- CHANGE ACCOUNT’S PASSWORD*

* + 1. *GUI Aspects*
       1. *Back Navigability*

There should be a *Go Back* button that is *active* and

*clickable* and navigates back to the *first page*.

* + - 1. *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.

* + 1. *Erroneous Inputs*
       1. *Unspecified Fields*
          1. *Current Password not specified*

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

* + - * 1. *New Password not specified*

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

* + - * 1. *Confirm Password not specified*

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

* + - 1. *Wrongly Specified Fields*
         1. *Current Password incorrect*

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

* + - * 1. *New Password has No Characters* An *error* with suitable message should pop up *without proceeding further*.
        2. *New & Confirm Password Don’t Match*

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

* + 1. *Correct Inputs and Change Password Successful*
       1. *Change Password Button Working*

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

* + - 1. *Verify Changed Password*
         1. *Try Log In with Old Password*

*Log in* should be unsuccessful.

* + - * 1. *Try Log In with New Password*

*Log in* should be successful.

# *Test Plan for AUTOMATIC FEATURES*