<1 > - <Login Case>:

Description: Login to the system with a username and password of a user who is a member of the system.
Pre-conditions: The user is a member and has to act on a computer connected to the library system.
Post-conditions: Display system home page.
Data required: User name (String) and Password (String)

<1.1> - <Login Page with right input>:

Description: User, entered the system to system username and password. If information is true, user will be able to enter the system.
Pre-conditions: User run the program login screen is opened. The user enters the username and password and presses the login button.
Post-conditions: The user logs in and the system displays the main app window.

Data-required: Username: valid string   Password: valid string

<1.2> - <Login Page with wrong input>:

Description: If information is false system will be display error message
Expected Output: Wrong username password combination.
Pre-conditions: User run the program login screen is opened. The user enters the username and password and presses the login button
Post-conditions: The system displays an error message.
Data-required: Username or Password: invalid string

<2 > - <Logout Case>:

Description: A user who is a member of the system has to exit the system.
Pre-conditions: Access to the system via a computer connected to the library system.
Post-conditions: Exiting the system and closing the program.
Data required: A data is not connected.

<3 > - <View my Books Case>:

<3.1> - <View my books when the book list is not empty>:

Description: Display the books that a user who is a member of the system gets on the system.
Pre-conditions: Access to the system via a computer connected to the library system.
Post-conditions: Listing books.
Data required: Book name(String), book ID(int), borrowing date(Date) and return date(Date).

<3.2> <View my books when the book list is empty>:

Description: The user can view the self borrowed books by clicking view my book button.
Expected result: Display the message:"You do not have any borrowed books."
Pre-conditions: User must be logged in
Post-conditions: System shows the empty list.
Data required: User does not any borrowed books.

<4 > - <Search Books Case>:

Description: The user can do searching for books in the system.
Pre-conditions: Access to the system via a computer connected to the library system. And Enter the name of the desired book.
Post-conditions: Displaying the books found in the search result.
Data required: Book name(String), book ID(int) author of book(String), status of book(bool)borrowing date(Date) and return date(Date).

<4.1><Search for Books with correct input>

Description: The user who has logged into the system can search for books
Expected Output: System displays the list of books which are found by searching operation and information about them

Pre-conditions: The search for books window is opened. The search criterias are entered and the search button is clicked.
Post-conditions: The system displays the search result according to the search criterias.
Data-required: Book name(String), book ID(int) author of book(String), status of book(bool)borrowing date(Date) and return date(Date).

<4.2>- <Search for Books with incorrect input>

Description: The user who has logged into the system can search for books
Expected Output: System displays the message “There is no book by that name”
Pre-conditions: The search for books window is opened. The search criterias are entered and the search button is clicked.
Post-conditions: The system displays the message of no book found.

<6 > - <Self Check-Out Case>:

Description: Borrowing a book from the system.
Pre-conditions: The book to be retrieved from the system has been searched, found and available.
Post-conditions: Displaying books on the main screen.
Data required: Book name(String), book ID(int) author of book(String), status of book(bool)borrowing date(Date) and return date(Date).

<7 > - <Self Return Case>:

Description: User tries giving back burrowed book to the system
Pre-conditions: User must had been a book borrowed. There must not be any debts.
Post-conditions: Displaying books on the main screen.
Data required: Book name(String), book ID(int) author of book(String), status of book(bool)borrowing date(Date) and return date(Date).

<7.1> - <Self Return Case for invalid books>:

Description: The user tries giving back books which are not burrowed to the system
Expected result: System displays the message: ”This book is not yours”
Pre-conditions: User must be logged in
Post-conditions: There is no change of state of book.
Data required: Book name(String), book ID(int) author of book(String), status of book(bool)borrowing date(Date) and return date(Date).

<8> - <Add name to book waiting list Case>:

Description: User requests for the defined requested book
Pre-conditions: Someone must not had been borrowed the requested book
Post-conditions: Adding the name to the list and change the book’s state as not available
Data required: Book name(String), book ID(int) author of book(String), status of book(bool),borrowing date(Date), return date(Date) User Information.

<8.1> - <Add name to book waiting list for an undefined book Case>:

Description: User requests for the undefined requested book
Pre-conditions: Someone must not had been borrowed the requested book
Post-conditions: User can not add him/her name to list
Data required: Book name(String), book ID(int) author of book(String), status of book(bool),borrowing date(Date), return date(Date) User Information.

<9 > - <Issue late fine Case>:

Description: Send warnings for upcoming or past books on the date of return.
Pre-conditions: The book must had been borrowed and the due date of book must be nearing or past.
Post-conditions: Send warning message to user's mail account.
Data required: Book name(String), book ID(int) author of book(String), borrowing date(Date), return date(Date) User Information.

<10 > - <Self Return Case>:

Description: Notification that the requested book is available.
Pre-conditions: It has been added to the wish list and the book has been returned.

Post-conditions: Send notification message to user's mail account.
Data required: Book name(String), book ID(int) author of book(String), borrowing date(Date), return date(Date) User Information.

<11 > - <Manipulate book Case>:

Description: The administrator adds a new book to the system, updates or deletes existing book information.
Pre-conditions: Entrance into the system as an administrator.
Post-conditions: Book information changes.
Data required: : Book name(String), book ID(int) author of book(String), User Information.

<11.1> - < Create Book >:

Description: When the administrator click “add new book” button and fill the book informations. system adds a new book,
Expected output: User receives “book created successfully” message
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Book is created
Data required: : Book name(String), book ID(int) author of book(String), User Information.

<11.1.1> - < Create Book with already existing book >:

Description: When the admisinistrator clicks “add new book” button and write a book name which does not exist. system gives an error message
Pre-conditions: User which who in must be logged as administrator and the book name which is wanted to create must already exist in the system
Expected output: User receives “this book is already created” message
Post-conditions: Program gives an error message.
Data required: : Book name(String), book ID(int) author of book(String), User Information.

<11.2> - < Update Book >:

Description: When the administrator click “update book” button and the book which is wanted to update is updated
Expected output :User receives “book updated successfully” message.
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Book is updated in the system.
Data required: : Book name(String), book ID(int) author of book(String), User Information.

<11.2.1> - < Update Book with already existing bookname >:

Description: When the administrator click “update book” button and the book which is wanted to update has a name that already exist in the system. program gives an error message
Expected output :User receives “this name is already used” message.
Pre-conditions: User wh logged in must be logged as administrator.
Post-conditions: Program gives an error message.
Data required: : Book name(String), book ID(int) author of book(String), User Information.

<11.3> - < Delete Book >:

Description: When the administrator click “delete book” button and fill the book informations; delete book from the system
Expected output :User receives “book deleted successfully” message
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Book is deleted.
Data required: : Book name(String), book ID(int) author of book(String), User Information.

<11.3.1> - < Delete Book with bookname that does not exist in the system>:

Description: When the administrator click “delete book” button and fill the book informations which include a book name that does not exst in the system
Expected output : User receives “This book name is not exist.” message

Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Program gives an error message.
Data required: : Book name(String), book ID(int) author of book(String), User Information.

<12 > - <Manipulate custemer Case>:

Description: The administrator can add a new customer to the system, update or delete existing customer information
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Customer information changes
Data required: User Information.

<12.1> - <Create customer>:

Description: The administrator can add a new customer to the system
Expected Output: User receives “Customer created successfully.” message
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Customer added.
Data required: User Information.

<12.1.1> - <Create customer with existing user name>:

Description: The administrator can not add a new customer whose name is already exist in the system
Expected Output: User receives “This Customer has already created” message
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Program gives an error message.
Data required: User Information.

<12.2> - <Update customer>:

Description: The administrator can update a customer’s information
Expected Output: User receives “Customer information updated successfully.” message
Pre-conditions: User which logged in must be logged as administrator.
Post-conditions: Customer updated.
Data required: User Information.

<12.2.1> - <Update customer name with existing user name>:

Description: The administrator can not change customer's name with the same name
Expected Output: User receives “Customer has already this name” message
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Program gives an error message.
Data required: User Information.

<12.3> - <Delete customer>:

Description: The administrator can delete customer from the system
Expected Output: User receives “Customer deleted successfully.” message
Pre-conditions: User which logged in must be logged as administrator.
Post-conditions: Customer deleted.
Data required: User Information.

<12.3.1> - <Delete customer whose user name does not exist>:

Description: The administrator can not delete a customer whose name does not exist in the system
Expected Output: User receives “Customer does not exist in the system” message
Pre-conditions: User who logged in must be logged as administrator.
Post-conditions: Program gives an error message.
Data required: User Information.

<13 > - <Pay fine Case>:

Description: The administrator is required to collect the prices of books with penalty and return them to the library.
Pre-conditions: User who logged in must be logged as administrator and he/she must find the punishment of the book
Post-conditions: The book is returned to the library and the debt is erased.
Data required: Book name(String), book ID(int) author of book(String), User Information.