**Use Case Name : authentic**

Primary Actor -> Person , Admin , Librarian

Goal in Context -> check if user authorized or not

Pre -> user has been register a username and password

Successfully end condition -> user is authorized to access the system

Failed end condition -> user is not authorized

Main flow Step Action

1-the data are provided to the system

2-check database verified

3- return database verified

Extension : Login

**Use case name : Login**

Primary Actor -> Person , Admin , Librarian

Goal in Context -> user can access the system with him permission

Pre : authorized username and pass

Successfully end condition -> login to the system

Failed end condition -> not authorized

Main Flow Step Action

1-Open The main Page of system

2-Enter Username and Pass

3-click login button

4-Validation

5-Included:: authorized username and pass

6-open him profile

Include Case : authentic

**Use Case Name : Logout**

Primary Actor -> Person , Admin , Librarian

Goal in Context -> user have been logged out and him credentials has been destroyed from memory

Pre : user has been logged in

Successfully end condition -> go to main page and him credentials has destroyed from memory

Failed end condition -> system can`t destroy him credentials

Main Flow Step Action

1-click logout button

2-destroy credentials from memory

3-transmit him to main page

**Use Case Name : reserve book**

Primary Actor -> System

Goal in Context -> the book is reserved

Pre : the book is available in the database

Successfully end condition -> the book is borrow by librarian

Failed end condition -> the book is not available and librarian can`t order it

Main Flow Step Action

1-get book details from the librarian

2-Validation

3-check if book available or not

4-reserve book

Extension : get book

**Use Case Name : Generate Invoice**

Primary Actor -> System

Goal in Context -> invoice is generated

Pre : librarian ask to generate invoice

Successfully end condition -> invoice is present to librarian

Failed end condition -> system can`t generate the invoice

Main Flow Step Action

1-get invoice details from librarian

2-Validation

3-create invoice with data

4-present invoice to librarian

Extension : show invoice

**Use Case Name : Add book**

Primary Actor -> Administrator

Goal in Context -> book is add to the system database

Pre : logged in

Successfully end condition -> book is added successfully to the system database

Failed end condition -> book is not added

Main Flow Step Action

1-click add book button

2-enter book details

3-Validation

4-click save button

5- the data inserted into the system database

6- return message to the administrator

**Use Case Name : delete book**

Primary Actor -> Administrator

Goal in Context -> book is deleted from the system database

Pre : logged in , the book is found in system database

Successfully end condition -> book is deleted successfully to the system database

Failed end condition -> book is found in the system database

Main Flow Step Action

1-click delete book button

2-enter book data

3-Validation

4-check if book in system database or not

5-delete the book from system database

6-return message to the administrator

**Use Case Name : Edit Book**

Primary Actor -> Administrator

Goal in Context -> book is modified in the system database

Pre -> logged in , book is founded in system database

Successfully end condition -> book is modified successfully

Failed end condition -> book is not modified successfully

Main Flow Step Action

1-click edit book button

2-enter new book data

3-validation

4-click save button

5-modified details in system database

6-return successfully message

**Use Case Name : Add section**

Primary Actor -> Administrator

Goal in Context -> section is add to the system database

Pre : logged in

Successfully end condition -> section is added successfully to the system database

Failed end condition -> section is not added

Main Flow Step Action

1-click add section button

2-enter section details

3-Validation

4-click save section

5- the data inserted into the system database

6- return message to the administrator

**Use Case Name : delete section**

Primary Actor -> Administrator

Goal in Context -> section is deleted from the system database

Pre : logged in , the section is found in system database

Successfully end condition -> section is deleted successfully to the system database

Failed end condition -> section is found in the system database

Main Flow Step Action

1-click delete section button

2-enter section data

3-Validation

4-check if section in system database or not

5-delete the section from system database

6-return message to the administrator

**Use Case Name : Edit Section**

Primary Actor -> Administrator

Goal in Context -> section is modified in system database

Pre -> logged in , section is founded in system database

Successfully end condition -> section is modified in system database

Failed end condition -> section is not modified successfully

Main Flow Step Action

1-click edit section button

2-enter new section data

3-click save button

4-validation

5-modified section data in system database

6-return successfully message

**Use Case Name : get book**

Primary Actor -> Administrator, Librarian

Goal in Context -> get message from “System”

Pre : logged in

Successfully end condition -> book is reserved

Failed end condition -> book is not reserved

Main Flow Step Action

1-click get book button

2-include::reserved book

3-receive message from the system

Include Case : reserve book

**Use Case Name : show invoice**

Primary Actor -> Administrator, Librarian

Goal in Context -> receive invoice from the system

Pre : logged in

Successfully end condition -> invoice is presented

Failed end condition -> invoice is not presented

Main Flow Step Action

1-click show invoice

2-include::generate invoice

3- receive invoice from the system

Include Case :generate invoice

**Use Case Name : add member**

Primary Actor -> Administrator, Librarian

Goal in Context -> member is added successfully in the system database

Pre : logged in

Successfully end condition -> member is added successfully

Failed end condition -> member is not added successfully

Main Flow Step Action

1-click add member button

2-enter member data

3-Validation

4-validation

5-insert data into system database

**Use Case Name : delete member**

Primary Actor -> Administrator, Librarian

Goal in Context -> member is delete successfully from the system database

Pre : logged in

Successfully end condition -> member is deleted successfully

Failed end condition -> member is not deleted successfully

Main Flow Step Action

1-click delete member button

2-enter member id

3-Validation

4-check if member is found or not

5-deleted from system database

**Use Case Name : edit member**

Primary Actor -> Administrator, Librarian

Goal in Context -> member data is updated successfully in the system database

Pre : logged in

Successfully end condition -> member data is updated successfully

Failed end condition -> member data is not updated successfully

Main Flow Step Action

1-click edit button

2-enter new member data

3-Validation

4-click save button

5-update data in the system database

**Use Case Name : search book**

Primary Actor -> Administrator, Librarian

Goal in Context -> present the book if is found

Pre : logged in , book is available in the system database

Successfully end condition -> book is available in system database

Failed end condition -> book is not available in system database

Main Flow Step Action

1-click search book button

2-enter book data

3-Validation

4-search book data in system database

5-show the book if founded

**Use Case Name : search section**

Primary Actor -> Administrator, Librarian

Goal in Context -> present the section if is found

Pre : logged in , section is available in the system database

Successfully end condition -> section is available in system database

Failed end condition -> section is not available in system database

Main Flow Step Action

1-click search section button

2-enter section data

3-Validation

4-search section data in system database

5-show the section if founded