

# Use Case

**Use Case:** Open App

**Id:** UC- 01

## **Description**

This will open the menu to select Library login or admin login

**Level:** Sub-Function

## **Primary Actor**

Library Owner

Admin

## **Supporting Actors**

Database

## **Stakeholders and Interests**

App designer, book librarians

## **Pre-Conditions**

Database must be created

Admin must allow for user input

Library Owner must have books ready to input

## **Post Conditions**

### Success end condition

Library Menu is opened

Admin Menu is opened

### Failure end condition:

App does not open

Wrong password entered

### Minimal Guarantee

App will not change without saving

## **Trigger**

Library Owner clicks button and inputs user name and password

Main Success Scenario

1. Library Owner clicks on the library app

2. App opens
3. Library Owner can proceed to next use case document.

## **Extensions**

none

## **Variations**

No variations

## **Frequency:**

At start of app. Once weekly to monthly

## **Assumptions**

User has a need for the app  
User has access to needed information  
User can read English

## **Special Requirements**

The user must know what the call number, title, author, genre, publisher, number of copies, and date  
Issues

1. What is the maximum characters a title of a book can be?
2. Must have admin access to add Librarian

## **To do**

1. Design app
2. Make sure app is user friendly and clean looking

# Use Case

**Use Case:** Add Librarian

**Id:** UC- 02

**Description**

This will allow Admin to add a librarian

**Level:** Main Function

**Primary Actor**

Admin

**Supporting Actors**

Database

**Stakeholders and Interests**

App designer, book libraries

**Pre-Conditions**

Database must be created

Admin must have Librarian info

**Post Conditions**

Success end condition

Librarian is added

Failure end condition:

Librarian is not added

Minimal Guarantee

App will not change without saving

**Trigger**

Prompt for new Book title

Prompt for new Book Author

Prompt for new Number of Pages

Prompt to save

Main Success Scenario

1. Admin adds Librarian information
2. Librarian is added

## Extensions

Librarian is displayed in Library table

## Variations

No variations

## Frequency:

Any time Owner need to give librarian access

## Assumptions

User has a need for the app

User has access to needed information

User can read English

## Special Requirements

The user must know what the username, password, email, address, city, and contact number is for each librarian

## Issues

1. What is the maximum characters an email can be?
2. Is there a length limit on cities?
3. Do the librarians save correctly?
4. Can I have more than one librarian with similar info?

## To do

1. Add database to app
2. Link database to book java package
3. Connect three databases together
4. Add Hibernate
5. Map data together

# Use Case

**Use Case:** Add Book

**Id:** UC- 03

## **Description**

This will allow Owner to add a book to their library

**Level:** Main Function

## **Primary Actor**

Library Owner

## **Supporting Actors**

Database

## **Stakeholders and Interests**

App designer, book libraries

## **Pre-Conditions**

Database must be created

Library Owner must have books ready to input

## **Post Conditions**

### Success end condition

Book title is added

Book Author is added

Number of pages in book is added

### Failure end condition:

App does not add book title to database

App does not add book author to database

App does not add number of pages to database

### Minimal Guarantee

App will not change without saving

## **Trigger**

Prompt for Book title

Prompt for Book Author

Prompt for Publisher

Prompt for Call Number

Prompt for Genre

Prompt to save

## Main Success Scenario

1. Library Owner adds call number, title, author, genre, publisher, number of copies, and date
2. Success popup
3. Book is displayed

## Extensions

In step 3 if Owner saves the Library should reopen with new book added

## Variations

No variations

## Frequency:

Any time Owner need to add a book

## Assumptions

User has a need for the app  
User has access to needed information  
User can read English

## Special Requirements

The user must know what the call number, title, author, genre, publisher, date, and number of copies is for each book.

## Issues

1. What is the maximum characters a title of a book can be?
3. Is there a length limit on authors?
4. Can Authors have initials?

## To do

1. Add database to app
2. Link database to book \_app java package
3. Connect three databases together
4. Add Hibernate
5. Map data together

# Use Case

**Use Case:** Return Book

**Id:** UC- 04

**Description**

This will allow Owner to remove a book from their library

**Level:** Main Function

**Primary Actor**

Library Owner

**Supporting Actors**

Database

**Stakeholders and Interests**

App designer, book libraries

**Pre-Conditions**

Database must be created

Library Owner must have books ready to input

**Post Conditions**

Success end condition

Book is returned

Failure end condition:

App does not remove book from student account

Minimal Guarantee

App will not change without saving

**Trigger**

Prompt to delete

Prompt to save

Main Success Scenario

1. Library Owner removes book from student account
2. Book is removed
3. Library Owner is prompted to save information or cancel

## **Extensions**

In step 3 if Owner saves the Library should reopen with book removed

## **Variations**

No variations

## **Frequency:**

Any time Owner need to remove a book

## **Assumptions**

User has a need for the app

User has access to needed information

User can read English

## **Special Requirements**

The user must know what the id, call number, title, student id, student name, and date are for each return

## **Issues**

1. What if I delete the wrong book?
2. Can I undo changes?
3. What if I don't know the book Id?

## **To do**

1. Add ability to remove book from student



# Use Case

**Use Case:** Issue Book

**Id:** UC- 05

## **Description**

This will allow Librarian to loan a book

**Level:** Main Function

## **Primary Actor**

Librarian

## **Supporting Actors**

Database

## **Stakeholders and Interests**

App designer, book libraries

## **Pre-Conditions**

Database must be created

Admin must have Librarian info

## **Post Conditions**

### Success end condition

Book is added to student account

### Failure end condition:

Book is not added to student account

### Minimal Guarantee

App will not change without saving

## **Trigger**

Choose to issue book

Main Success Scenario

1. Librarian loans book to student
2. Book is added to issued student

## **Extensions**

Librarian is displayed in Library table

## **Variations**

No variations

## **Frequency:**

Any time Owner need to give librarian access

## **Assumptions**

User has a need for the app

User has access to needed information

User can read English

## **Special Requirements**

The user must know what the id, call number, title, student id, student name, and date are for each student

## **Issues**

1. Where can the call number be located?
2. Can I have more than one copy of a book?
3. Do the students get the book issued correctly?
4. What if they return same day?

## **To do**

1. Add issuer java
2. Add time to date

# Use Case

**Use Case:** List Librarian

**Id:** UC- 06

**Description**

This will allow Admin to see all librarians

**Level:** Main Function

**Primary Actor**

Admin

**Supporting Actors**

Database

**Stakeholders and Interests**

App designer, book libraries

**Pre-Conditions**

Database must be created

Admin must have Librarian info

**Post Conditions**

Success end condition

Librarians are listed

Failure end condition:

Librarian list is missing or incomplete

Minimal Guarantee

App will not change without saving

**Trigger**

Click list librarian

Main Success Scenario

3. Admin adds Librarian information
4. Librarian is added

## **Extensions**

Librarians are listed from database table

## **Variations**

No variations

## **Frequency:**

Any time Owner needs to see who the librarians are

## **Assumptions**

User has a need for the app

User has access to needed information

User can read English

## **Special Requirements**

The user must know what the username, password, email, address, city, and contact number is for each librarian

## **Issues**

1. Do the librarians all list?
2. Do the librarians save correctly?
3. Can I have more than one librarian with similar info?

## **To do**

1. Add database to app
2. Link database to book\_app package
3. Connect databases together
4. Add Hibernate
5. Map data together

# Use Case

**Use Case:** Delete Librarian

**Id:** UC- 07

**Description**

This will allow Admin to delete a librarians

**Level:** Main Function

**Primary Actor**

Admin

**Supporting Actors**

Database

**Stakeholders and Interests**

App designer, book libraries

**Pre-Conditions**

Database must be created

Admin must have Librarian info

**Post Conditions**

Success end condition

Librarian is deleted successfully

Failure end condition:

Librarian is not deleted

Minimal Guarantee

App will not change without saving

**Trigger**

Choose a librarian to delete

Main Success Scenario

5. Admin adds Librarian information
6. Librarian is added

## Extensions

Librarians are listed from database table

## Variations

No variations

## Frequency:

Any time Owner needs to see who the librarians are

## Assumptions

User has a need for the app

User has access to needed information

User can read English

## Special Requirements

The user must know what the username, password, email, address, city, and contact number is for each librarian

## Issues

1. Do the librarians all list?
2. Do the librarians save correctly?
3. Did the librarian delete completely?
4. Can I read librarian?

## To do

1. Add database to app
2. Link database to book\_app package
3. Connect databases together
4. Add Hibernate
5. Map data together