



Library Management System

Developing in java and MySQL

TOPICES	Page
Introduction	2
Goal	2
Objective	2
ER-diagram	3-9
Logical Design (Relational Database Schema)	10-16
Data creation	(I attached extra SQL file)
Creating connection with database	17 -
User Interface working diagram and descriptions	18-20
GUI Library Management System homepage	21
Login GUI	22
Admin home GUI	23
Admin Add Menu GUI	23-24
Admin Edit menu GUI	25
Admin update menu GUI	25-26
Admin delete menu GUI	27
Admin show menu GUI	28-29
Librarian Home GUI	30
Librarian Add menu GUI	31-34
Librarian Edit menu GUI	35
Librarian Update menu GUI	35-37
Librarian Delete menu GUI	38-40
Librarian show menu GUI	41-42
Issue Book GUI	43
Return Book	44
Reserve List	44-45
Member Home GUI	45
Reserve Book GUI	46
Search Book GUI	47
Show Book	47

Library Management System

A Library management system is a software that uses to maintain the record of the library. It contains work like the number of available books in the library, the number of books are issued or returning or a book record, etc. Library Management Systems is software that helps to maintain a database that is useful to enter new books & record books borrowed by the members, with the respective submission dates. Moreover, it also reduces the manual record burden of the librarian.

Library management system allows the librarian to maintain library resources in a more operative manner that will help to save their time. It is also convenient for the librarian to manage the process of books allotting. Library management system is also useful for students as well as a librarian to keep the constant track of the availability of all books in a store.

The Library management system is nowadays essential for schools, colleges, private libraries, and other organizations. They can use this software as the purpose of books issuing and returning for renewal. Moreover, this software helps the librarian to maintain information about book issuing and returning the book before the last date.

Project Goal

The goal of coming up with this system was to design, develop, and implement a fully conceptual design, logical design, implementation, operation, and maintenance of a relational database and associated applications library book management system.

Project Objectives

This system has a number of objectives:

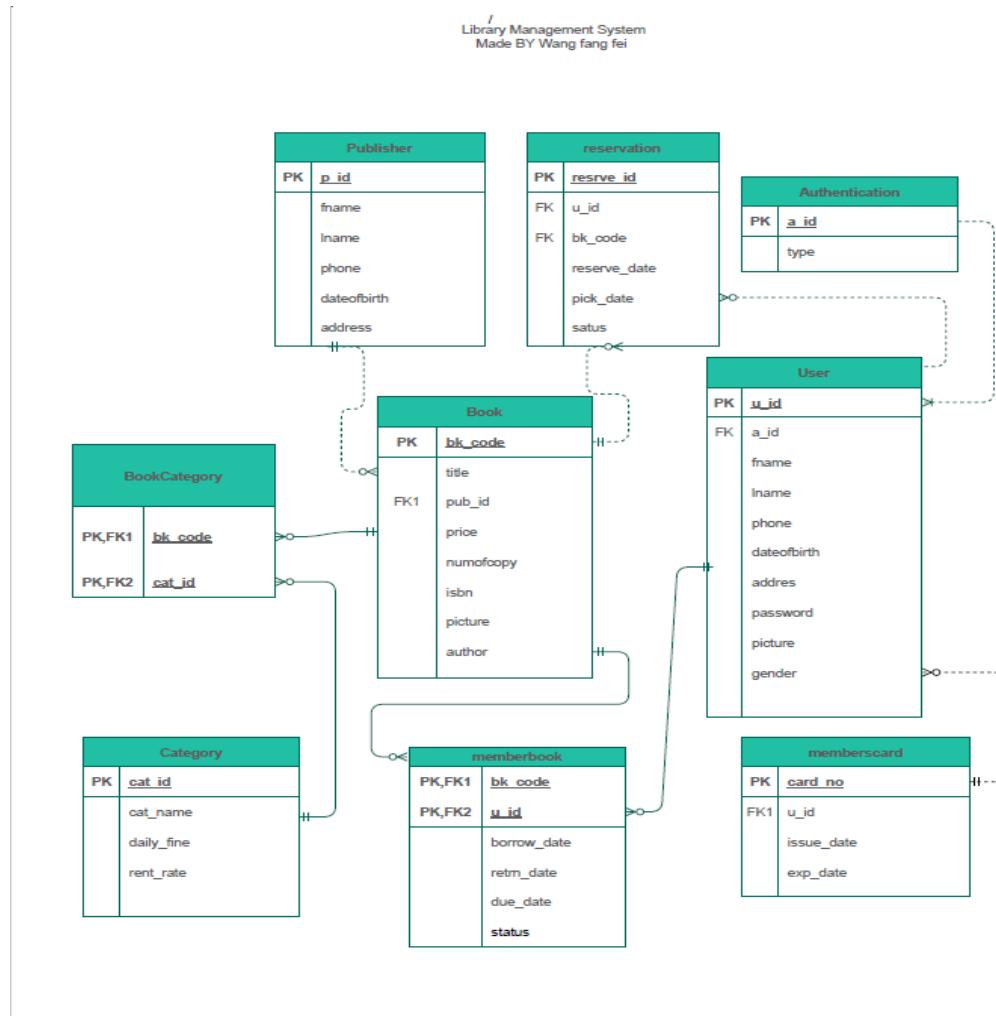
- 1) To develop a system that ensures the privacy of its users and enable them access it remotely.
- 2) To enable easy maintenance of members and book details.
- 3) To enable easy retrieval of books by simply searching the system.
- 4) To design a system that responds in a timely manner.
- 5) To enable easy borrowing and returning of books.
- 6) To enable a secure and portable database system that eliminates duplicate data.
- 7) To ensure a paperless environment.

Benefits of the Proposed System

This system improves services delivered to end users. Information can be searched from the system and the results displayed in a timely manner. Costs incurred are reduced by cutting down the number of library workforce. Data is saved in the most appropriate manner eliminating duplication and redundancy. Multiple users can access the system. The system operates electronically hence ensures less space is occupied and also presents a paperless working environment. The system has access control limiting access to restricted areas. User privacy is ensured by providing login functionality which verifies them to determine their authenticity to access the system. Library staff members are motivated by the system since it makes work easier. The system has a friendly user interface which is attractive and easy to use. Information concerning all the library activities is stored in the most secure manner. The system's database is password protected preventing unauthorized access .

Description of Cardinality

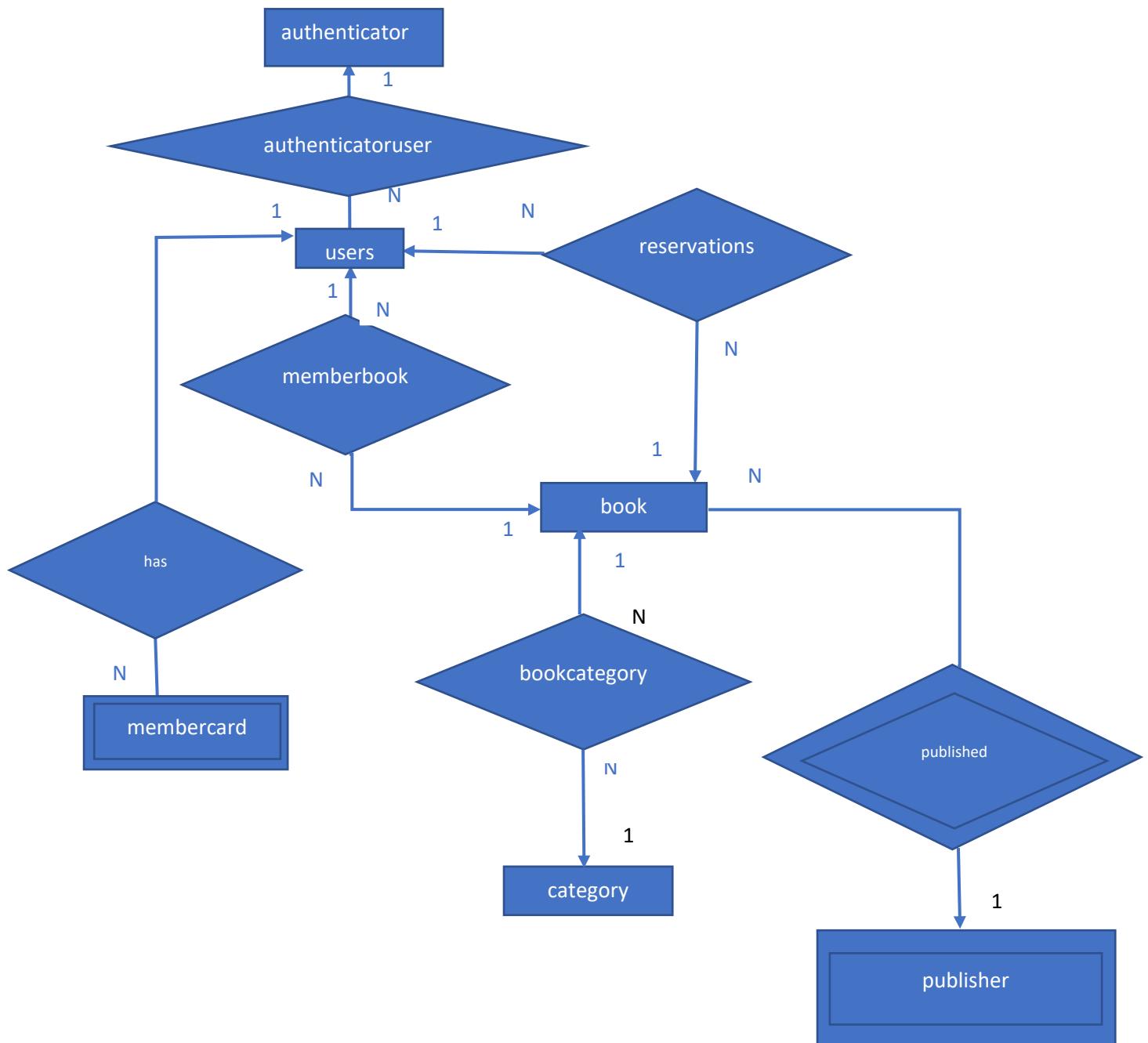
An **identifying** relationship is shown using a dotted line; a **non-identifying** relationship is shown using a solid line. Identifying relationship means that the primary key of the parent entity is also included as the primary key of the child entity forming a compound key. A non-identifying relationship means that the primary key of the parent entity is included in the child entity this time not as part of the primary key but as a foreign key. The following is an illustration of Identifying and Non-identifying relationship with regard to the figure.



Conceptual Design:

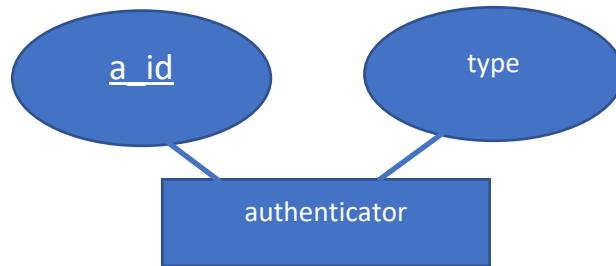
The ER model diagram below will be used to show the various entities in this system and how they relate to each other. I like to identify one relationship by 1 and many relationships I identify by N.

ER-Diagram:

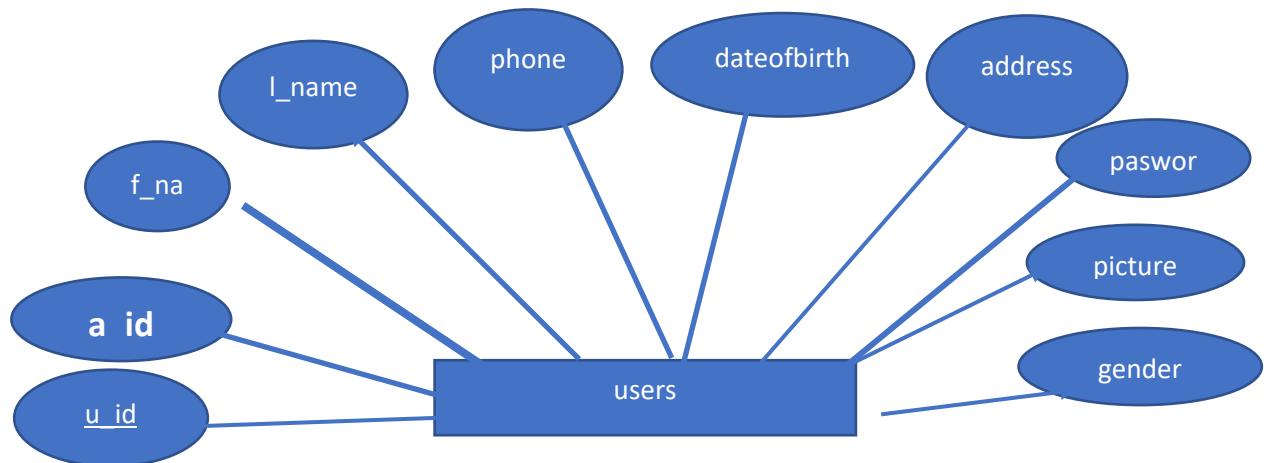


Now I will show all the attributes for each Entity step by steps:

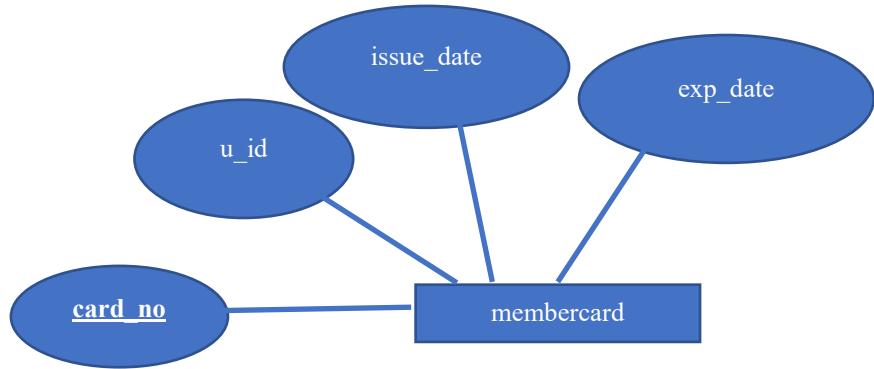
- 1) **authenticator Entity**: here **a_id** means authenticator id and it is **primary key**. And type means authenticator type.



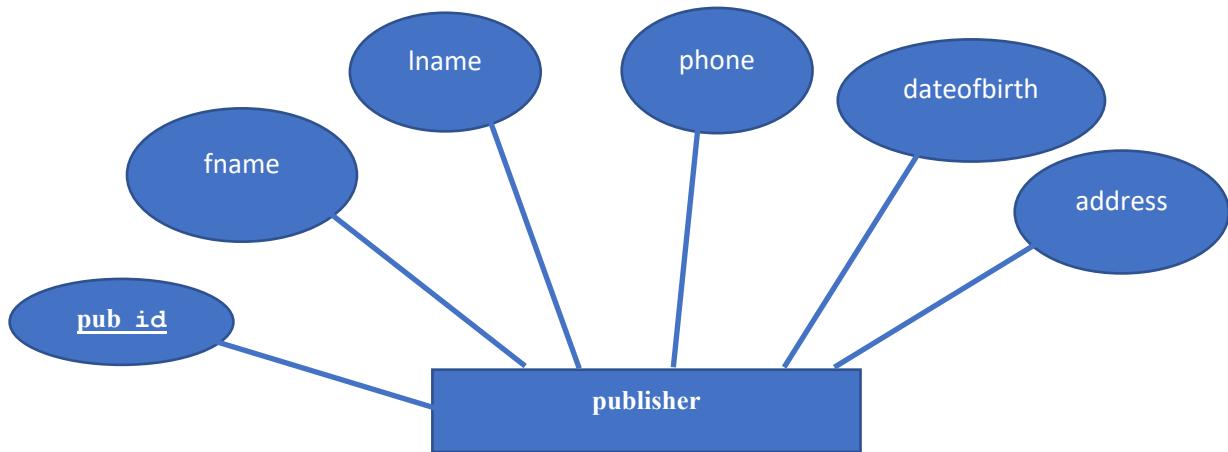
- 2) **users Entity**: In users Entity **u_id** means user id . This is a primary key and **a_id** is a foreign key(authenticator id).



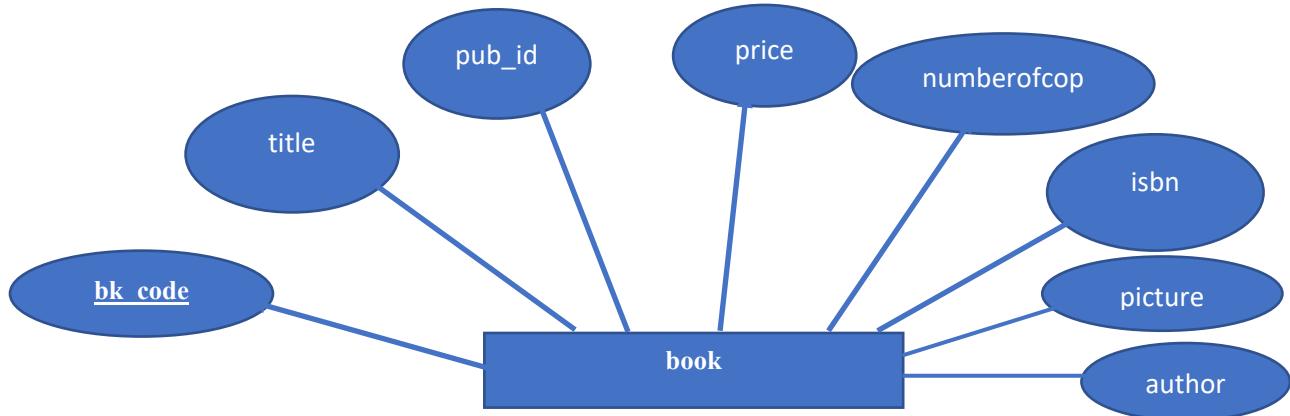
3)membercard Entity: In membercard Entity card_no is primary key. And u_id is foreign key (User Id)



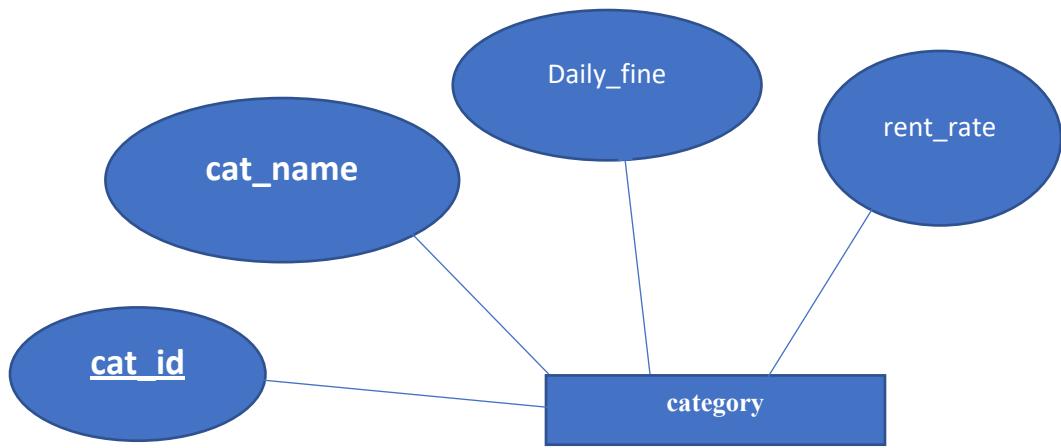
4)publisher Entity: In publisher Entity pub_id is Primary key. That means publisher Id.



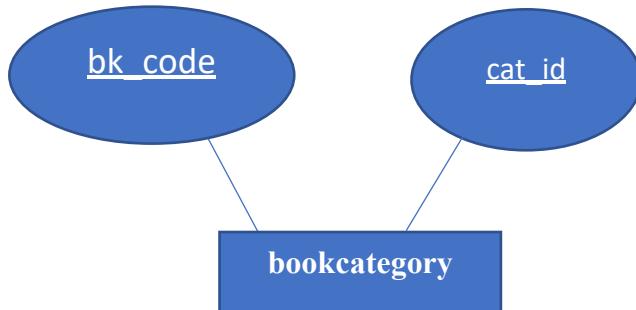
5)book Entity: In book Entity bk_code is Primary key. That means book code. And pub_id is foreign key (publisher Id).



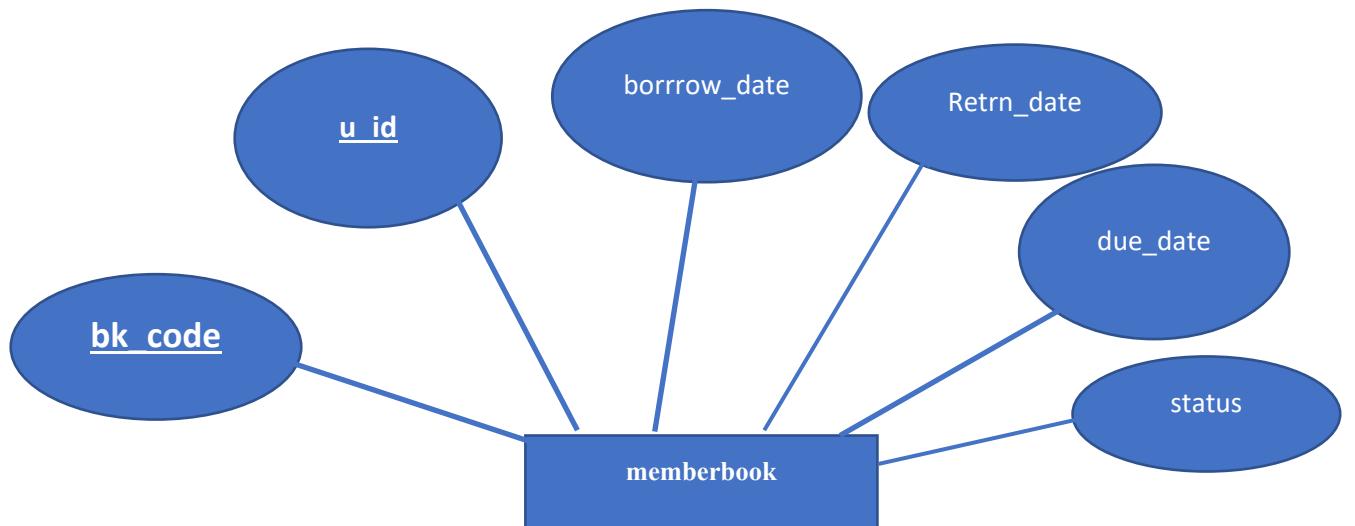
6)category Entity: In category Entity cat_id is primary key. That means category Id.



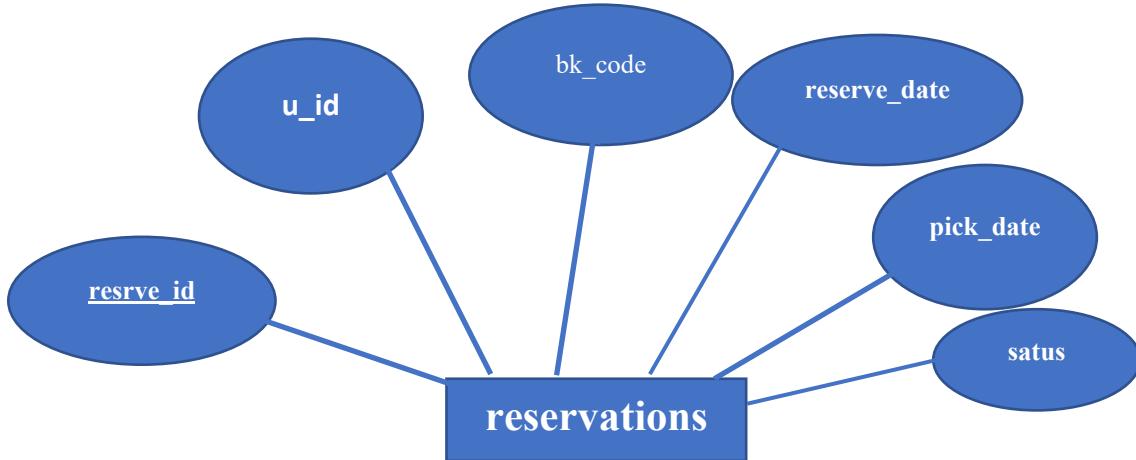
7)BookCategory Entity: In bookcategory Entity **bk_code(book code)** and **cat_id (category id)** is a primary key and also foreign key. Because this two comes from book and category table.



8)memberbook Entity: In memberbook Entity **bk_code(book code)** and **u_id(user id)** are a primary key and also foreignen key.



9) reservation Entity : In reservation resv_id is primary key and u_id and bk_code is a foreign key.



Relationships between Entities –

- A User can have different type of authentication so the relation is User with authenticator is Many to one.
- A Users can add one or many member cards. The relationship is one to many.
- A book can also be in one or more categories. The relationship is one to many.
- A book can be reserved by many members. The relationship is one to many.
- A category can have one or many books. This relationship is one to many.
- A user(member) can borrow many books. This relationship is one to many.
- A user(member) can reserve many books. This is a one-to-many relationships
- A publisher can publish many books but a book is published by only one publisher. The relationship one to many
- Authentication system provides login to multiple users. The relation is one to many.

2. Logical Design (Relational Database Schema)

The logical schema describes the tables in the database in detail

I create a database.

```
CREATE DATABASE LibraryManagementSystem
```

After I create 9 table.

1. Authenticator Table

```
create table authenticator(
    a_id varchar(20) not null ,
    type varchar(30) not null,
    primary key ( a_id)
);
```

LAPTOP-4LKFTJ90\...bo.authenticator [librarySystemD]		
Column Name	Data Type	Allow Null
a_id	varchar(20)	<input type="checkbox"/>
type	varchar(30)	<input type="checkbox"/>

Column Name	Description	DataType	PrimaryKey	Null
a_id	primary key (a_id)	varchar(20)	yes	No
type	Authenticator Type	varchar(30)	No	No

2. Users Table

```
create table users(
    u_id varchar(20) not null,
    a_id varchar(20) not null,
    fname varchar(50) not null,
    lname varchar(50) not null,
    phone varchar(50) not null,
    dateofbirth date not null,
    address varchar(100) not null,
    password varchar(50) not null,
    picture varchar(100) not null,
    gender varchar(20) not null,
    primary key ( u_id ),
    foreign key (a_id) references authenticator ,
);
```

Column Name	Data Type	Allow Nulls
u_id	varchar(20)	<input type="checkbox"/>
a_id	varchar(20)	<input type="checkbox"/>
fname	varchar(50)	<input type="checkbox"/>
lname	varchar(50)	<input type="checkbox"/>
phone	varchar(50)	<input type="checkbox"/>
dateofbirth	date	<input type="checkbox"/>
address	varchar(100)	<input type="checkbox"/>
password	varchar(50)	<input type="checkbox"/>
picture	varchar(100)	<input type="checkbox"/>
gender	varchar(20)	<input type="checkbox"/>

3. Membercard Table

```
create table membercard(
    card_no varchar(20) not null,
    u_id varchar(20) not null,
    issue_date date not null,
    exp_date date not null,
    primary key ( card_no),
    foreign key (u_id) references users,
);
```

Column Name	Data Type	Allow Nulls
card_no	varchar(20)	<input type="checkbox"/>
u_id	varchar(20)	<input type="checkbox"/>
issue_date	date	<input type="checkbox"/>
exp_date	date	<input type="checkbox"/>
		<input type="checkbox"/>

4. publisher Table

```
create table publisher(
    pub_id varchar(20) not null,
    fname varchar(50) not null,
    lname varchar(50) not null,
    phone varchar(50) not null,
    dateofbirth date not null,
    address varchar(100) not null,
    primary key (pub_id),
```

) ;

LAPTOP-4LKFTJ90\... - dbo.publisher		
Column Name	Data Type	Allow Nulls
pub_id	varchar(20)	<input type="checkbox"/>
fname	varchar(50)	<input type="checkbox"/>
lname	varchar(50)	<input type="checkbox"/>
phone	varchar(50)	<input type="checkbox"/>
dateofbirth	date	<input type="checkbox"/>
address	varchar(100)	<input type="checkbox"/>
		<input type="checkbox"/>

5. Book Table

```
create table book(  
  
bk_code varchar(20) not null,  
title varchar(100) not null,  
pub_id varchar(20) not null,  
price varchar(20) not null,  
numofcopy varchar(10) not null,  
isbn varchar(50) not null,  
picture varchar(100) not null,  
[author] [varchar](100)not null,  
primary key ( bk_code),  
foreign key (pub_id) references publisher ,  
  
) ;
```

LAPTOP-4LKFTJ90\...system - dbo.book		
Column Name	Data Type	Allow Nulls
bk_code	varchar(20)	<input type="checkbox"/>
title	varchar(100)	<input type="checkbox"/>
pub_id	varchar(20)	<input type="checkbox"/>
price	varchar(20)	<input type="checkbox"/>
numofcopy	varchar(10)	<input type="checkbox"/>
isbn	varchar(50)	<input type="checkbox"/>
picture	varchar(100)	<input type="checkbox"/>
author	varchar(100)	<input type="checkbox"/>
		<input type="checkbox"/>

6. Category Table

```

create table category(
    cat_id varchar(20) not null,
    cat_name varchar(100) not null,
    daily_fine varchar(20) not null,
    rent_rate varchar(50) not null,
    primary key (cat_id)

);

```

Column Name	Data Type	Allow Nulls
cat_id	varchar(20)	<input type="checkbox"/>
cat_name	varchar(100)	<input type="checkbox"/>
daily_fine	varchar(20)	<input type="checkbox"/>
rent_rate	varchar(50)	<input type="checkbox"/>

7.bookcategory Table

```

create table bookcategory(
    bk_code varchar(20) not null,
    cat_id varchar(20) not null,
    primary key (cat_id, bk_code),
    foreign key (cat_id) references category ,

    foreign key ( bk_code) references book ,

);


```

Column Name	Data Type	Allow Nulls
bk_code	varchar(20)	<input type="checkbox"/>
cat_id	varchar(20)	<input type="checkbox"/>
		<input type="checkbox"/>

8. create table memberbook (

```

bk_code varchar(20) not null,
u_id varchar(20)not null,
borrow_date date not null,
retrn_date date ,
due_date date not null,
status varchar(50) not null,
primary key (u_id, bk_code),

foreign key (u_id) references users,

```

```

foreign key ( bk_code) references book ,
) ;

```

Column Name	Data Type	Allow Nulls
bk_code	varchar(20)	<input type="checkbox"/>
u_id	varchar(20)	<input type="checkbox"/>
borrow_date	date	<input type="checkbox"/>
retrn_date	date	<input checked="" type="checkbox"/>
due_date	date	<input type="checkbox"/>
status	varchar(50)	<input type="checkbox"/>

9. reservations

```

create table reservations(
    resrve_id varchar(20) not null,
    u_id varchar(20) not null,
    bk_code varchar(20) not null,
    reserve_date date not null,
    pick_date date not null,
    satus varchar(50) not null,
    primary key (resrve_id) ,

    foreign key (u_id) references users,

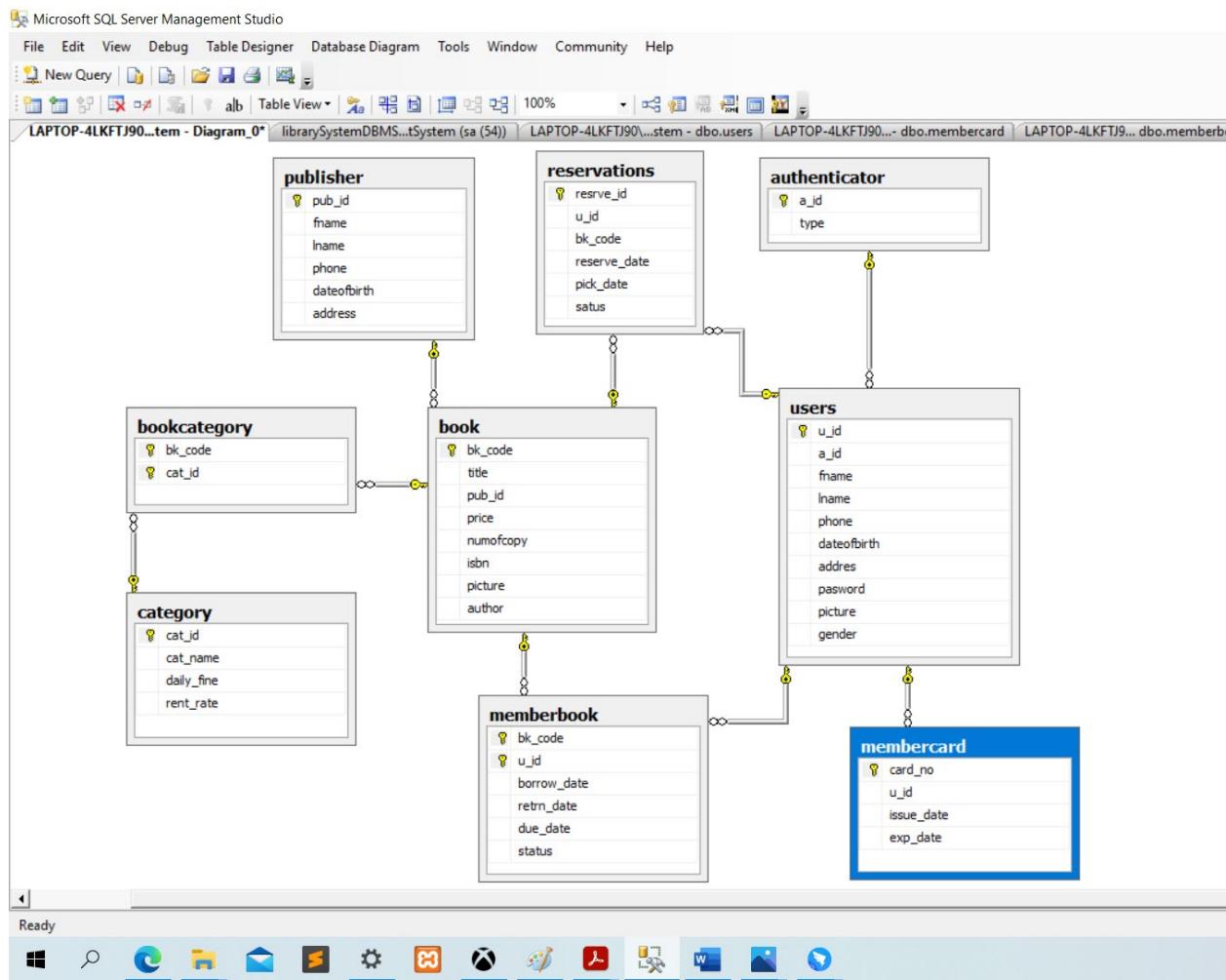
    foreign key ( bk_code) references book,

) ;

```

Column Name	Data Type	Allow Nulls
resrve_id	varchar(20)	<input type="checkbox"/>
u_id	varchar(20)	<input type="checkbox"/>
bk_code	varchar(20)	<input type="checkbox"/>
reserve_date	date	<input type="checkbox"/>
pick_date	date	<input type="checkbox"/>
satus	varchar(50)	<input type="checkbox"/>

Relational Diagram:



3.I some quarry and save it MySQL file. I attached it Under project folder

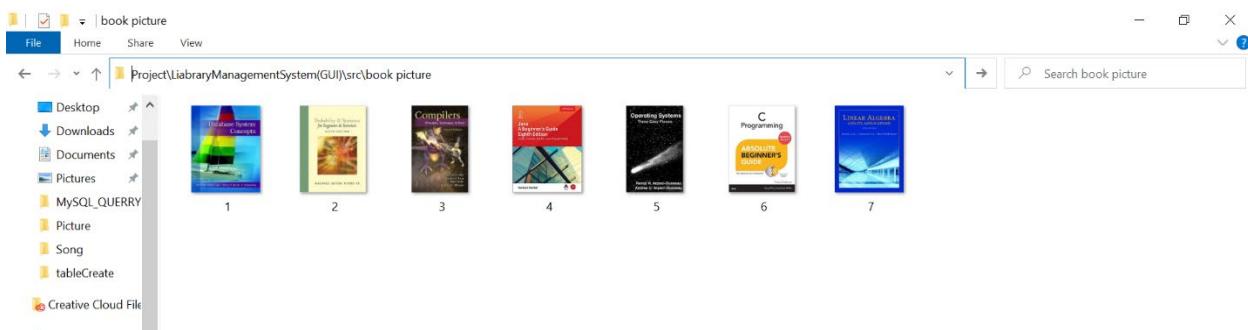
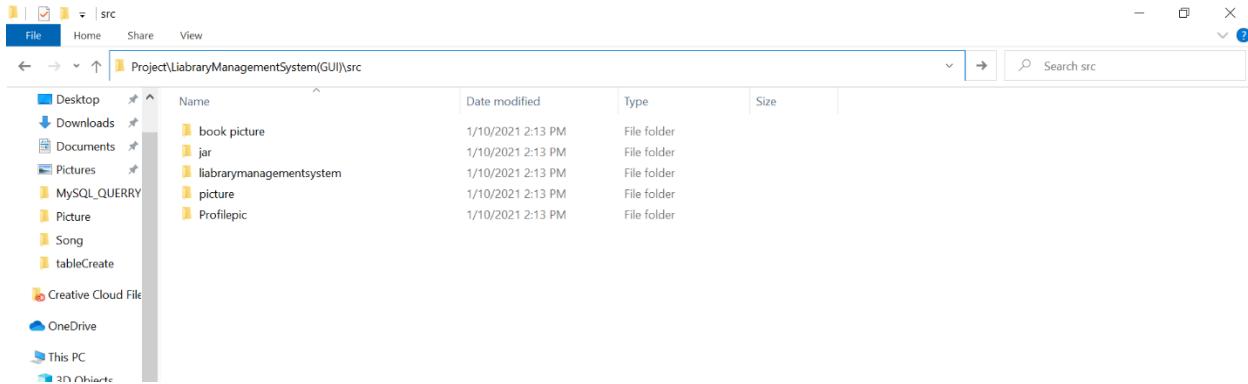
1.Create Database query

2.Create Table query

3.Data Insert query (note the picture path will change your computer because it detect my computer path) if you insert this query please update it.

Under Project\LiabraryManagementSystem(GUI)\src here has two folder book picture and Profilepic where I store Images for book and users. You can select images to there.

But if you insert book and users images by GUI from it will automatic detect the path.

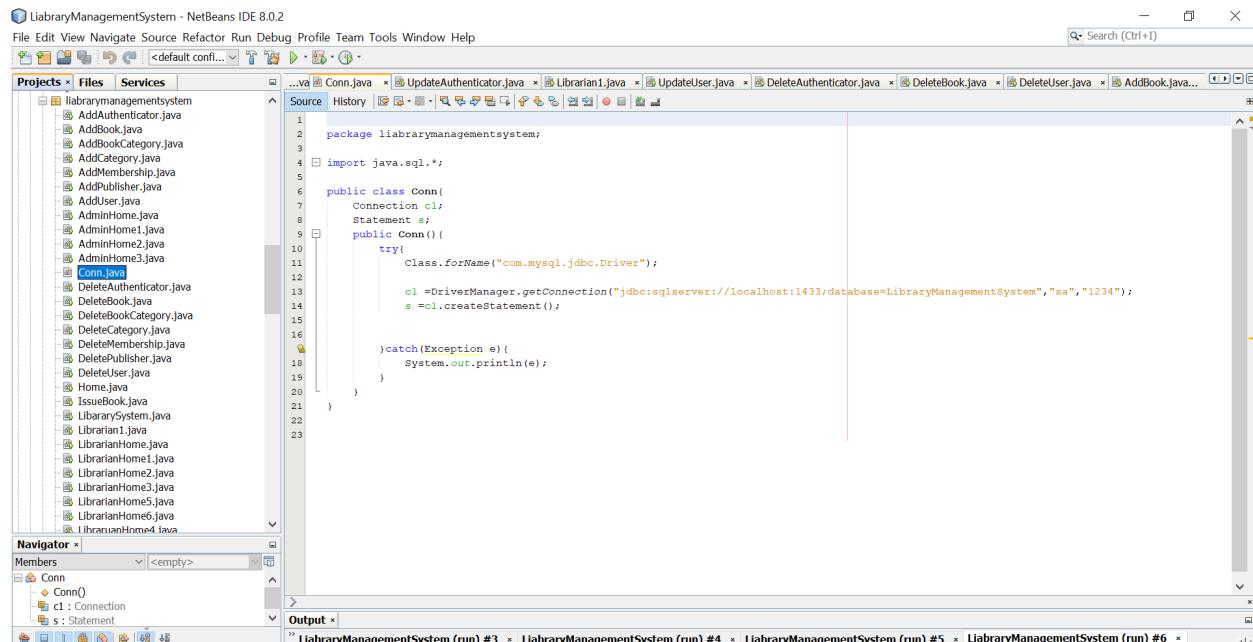


4. Update query

4. IMPLEMENTATION AND RESULTS

I implement my project by using JAVA user interface. And I write all the code manually line by line, without using drop down.

Create Connection with MySQL:



The screenshot shows the NetBeans IDE interface with the following details:

- Title Bar:** LiabraryManagementSystem - NetBeans IDE 8.0.2
- Menu Bar:** File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
- Toolbar:** Standard NetBeans toolbar icons.
- Project Explorer:** Shows the project structure under "librarymanagementsystem".
- Code Editor:** Displays the Conn.java code. The code is as follows:

```
1 package librarymanagementsystem;
2
3 import java.sql.*;
4
5 public class Conn{
6     Connection c1;
7     Statement s;
8     public Conn(){
9         try{
10             Class.forName("com.mysql.jdbc.Driver");
11
12             c1 =DriverManager.getConnection("jdbc:sqlserver://localhost:1433;database=LibraryManagementSystem","sa","1234");
13
14             s =c1.createStatement();
15
16         }catch(Exception e){
17             System.out.println(e);
18         }
19     }
20 }
21
22
23
```

- Navigator:** Shows the members of the Conn class.
- Output:** Shows multiple runs of the project.

```
package librarymanagementsystem;

import java.sql.*;
public class Conn{
    Connection c1;
    Statement s;
    public Conn(){
        try{
            Class.forName("com.mysql.jdbc.Driver");
            c1=DriverManager.getConnection("jdbc:sqlserver://localhost:1433;database=LibraryManagementSystem","sa","1234");
            s =c1.createStatement();
        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

Here I create a connection with my database. My database name is LibraryManagementSystem. User is sa , Password is 1234.

NOTE: You can run only 4 class because that 4 class only extends of JFrame and other class extends of JPanel. Those panel I adjust this 4 frame.

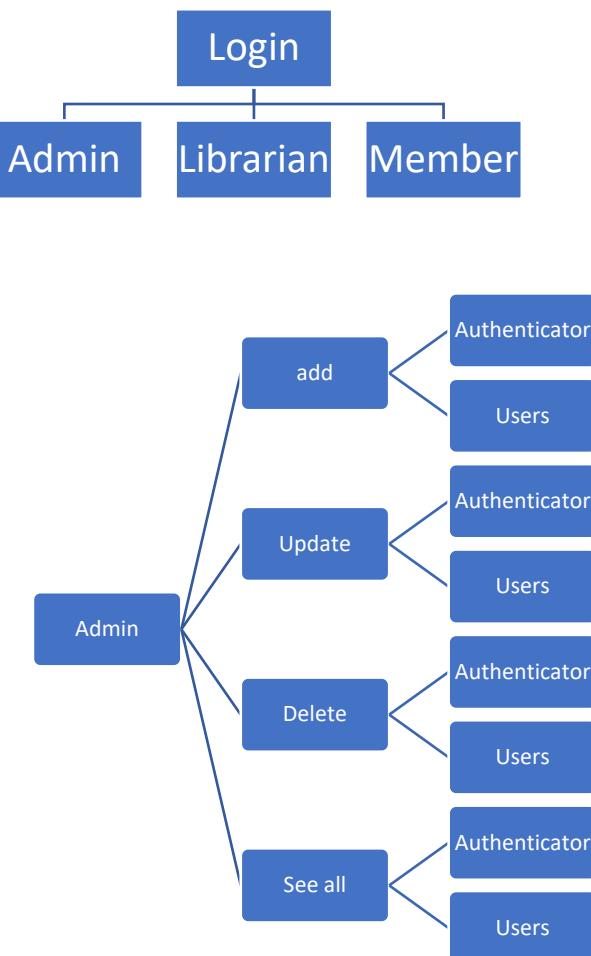
Those 4 class are. LibarySystem.java

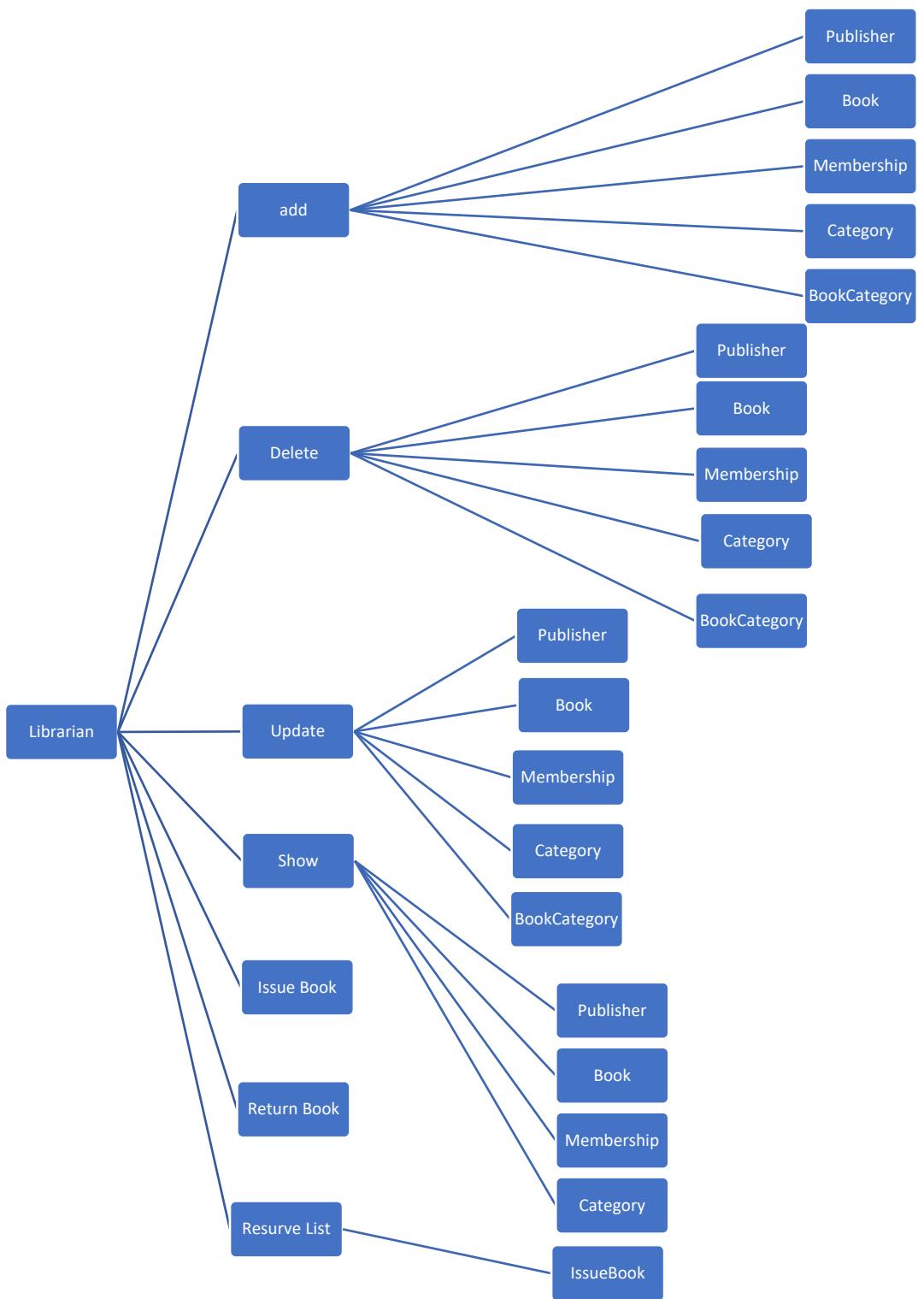
1. admin1.java
2. Librarian1.java
3. Member1.java

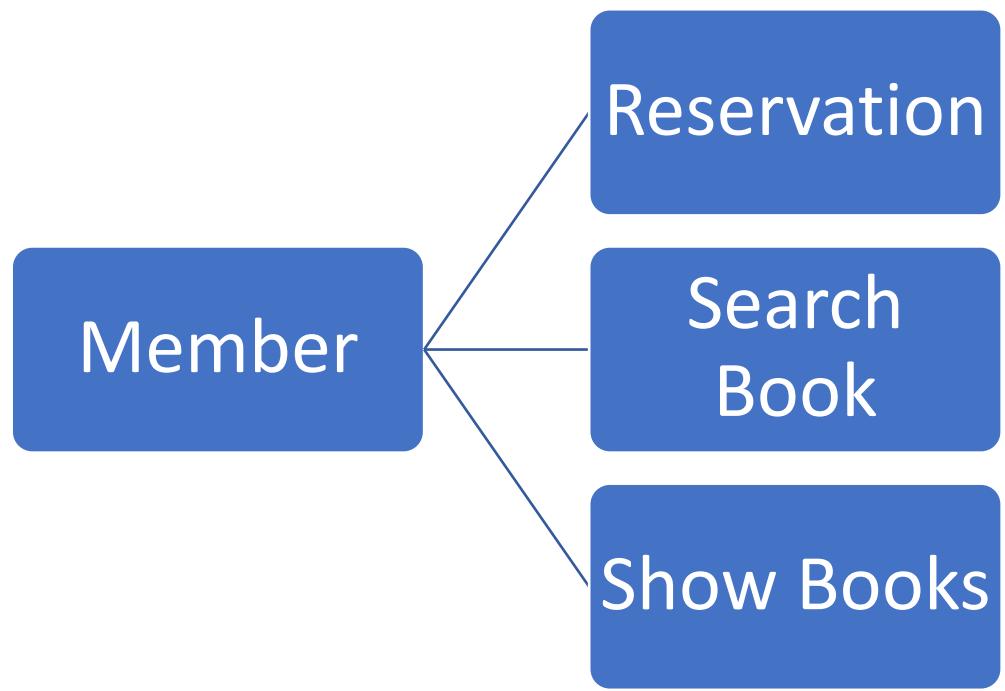
Here has There type of Authentication One for admin, one for Librarian and one for member.

LBMS has been designed and developed with high security standards. A number of security levels have been designed to limit privileges according to a user type. Users can access a page depending on the security levels given to them. These levels determine the various rights and operations a user can carry out on the system. The levels include; the admin access levels; librarian access levels and the members access levels.

In bellow I draw a diagram how can this system work. May be its make more easy to understand.







In My project LibaraySystem.java is **Library Management System homepage**.

shows the **Library Management System homepage**.



Jiangxi Normal University Library

As the public infrastructure of a well-known university in Jiangxi Province, the library of Jiangxi Normal University was initially established in 1940, after that, the library has been renamed repeatedly with the development of the university. The university has been successively named as National Chung Cheng University, National Nan Chang University, Jiangxi Normal College, Jiangxi Normal University, the library name was renamed correspondingly, in 1983, the library was renamed as Library of Jiangxi Normal University.

It has 3 option

- **Home:** It will show home page.
- **Login:** By click login option, user will get login page. And can login the system.
- **Exit:** the system will be exit.

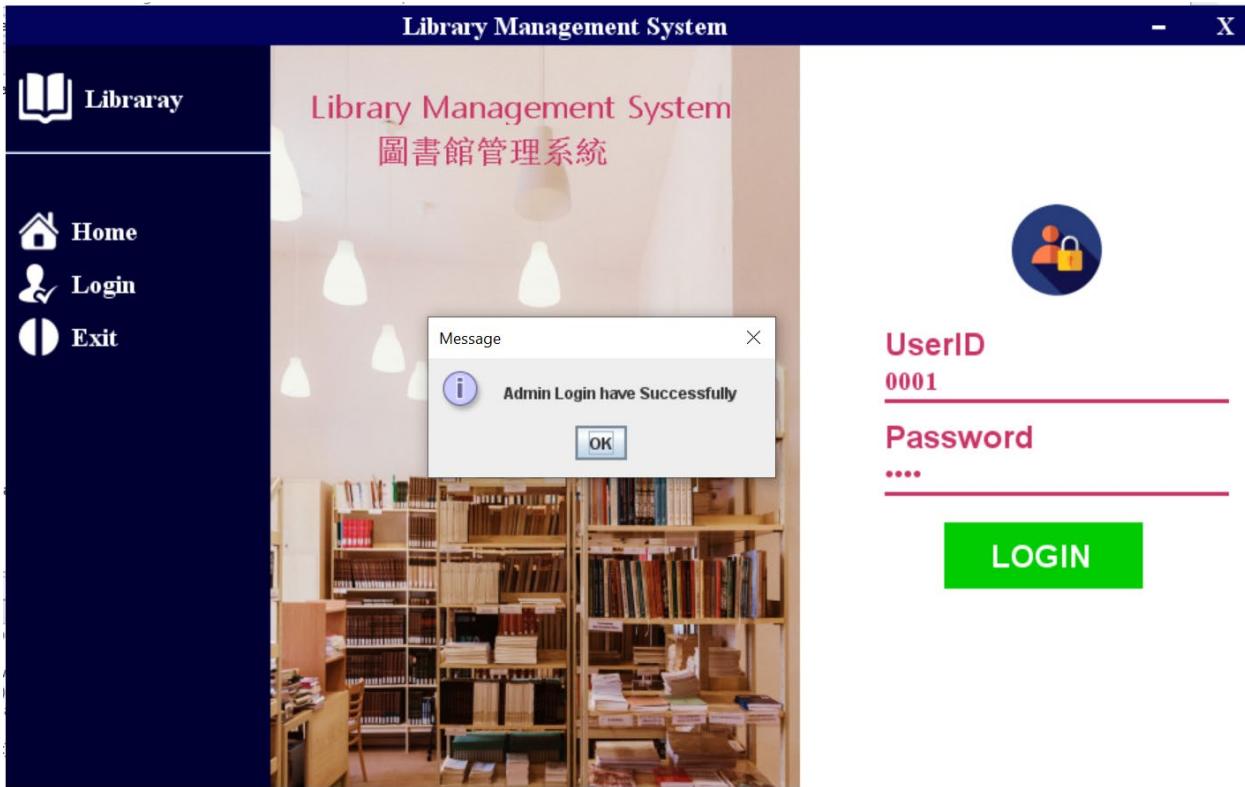
Login Page: In my project Login.java is login page.

By write the User Id and Password a user can login the system. It will detect automatic the user type.

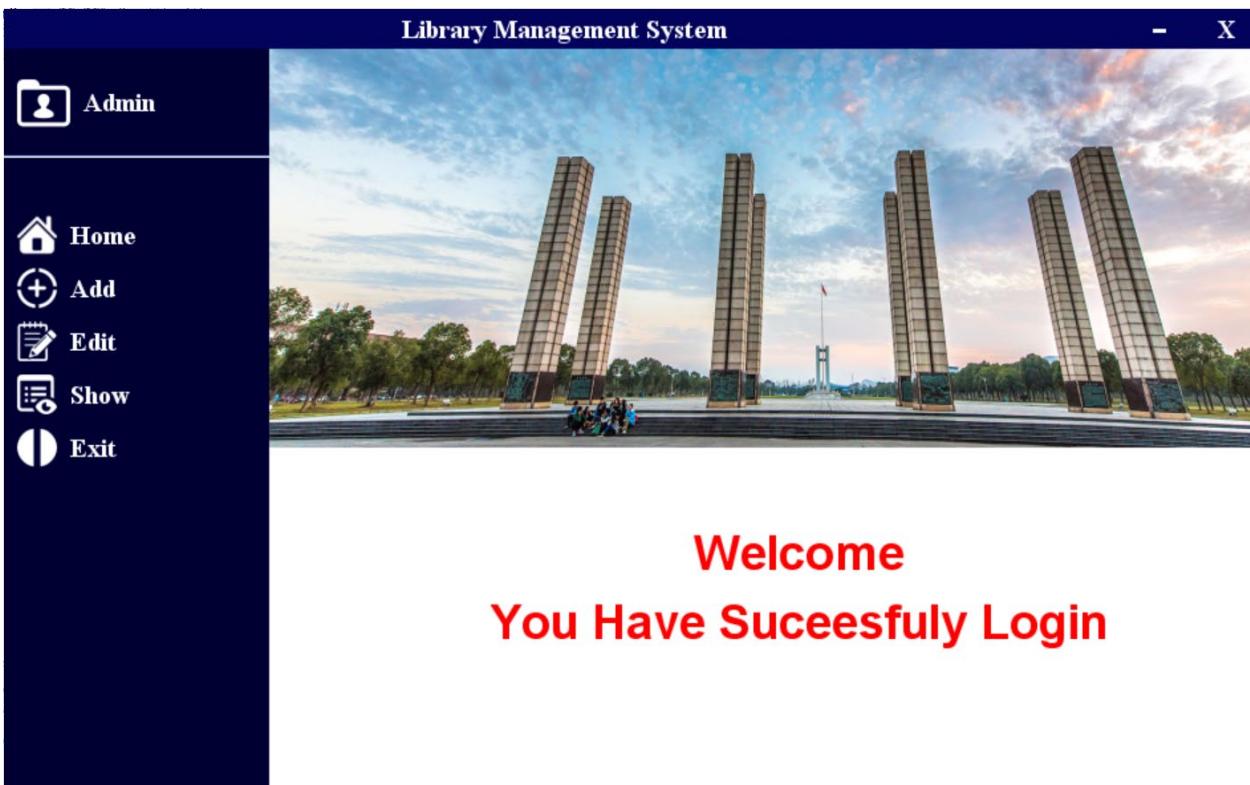
1. If User is an admin, it will go to admin homepage.
2. If User is a Librarian, it will go Librarian homepage.
3. If User is a member, it will go Librarian homepage.

We will at first See the Admin Window. (**And note one thing, first admin Information I insert into database because I want go admi home page by login option**)

Here admin id is 0001 and 1234. IF I put this and click login button. It will successfully login.



After admin Login successfully it will go Admin home page In my project file **admin1.java** is admin home page.



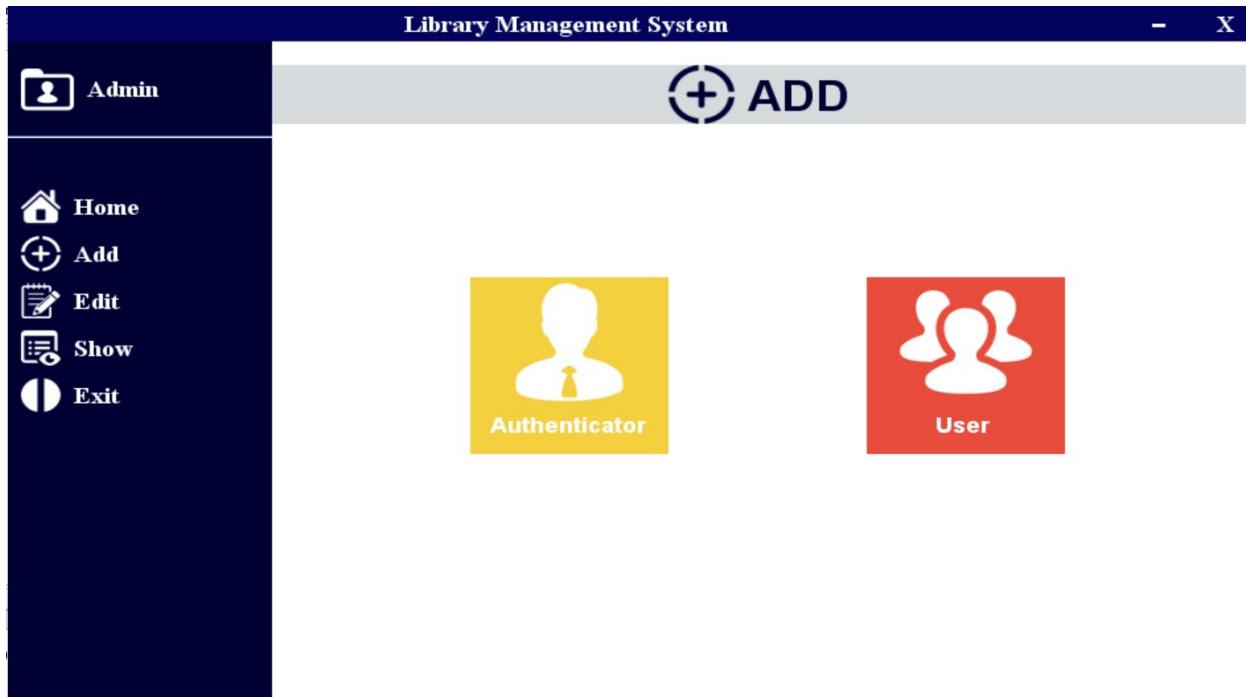
Here has 5 option.

1. Home page
2. Add Page: Here has add authenticator and add user's option.
3. Edit: Here has delete and update authenticator and user's option.
4. Show: Here has Shown authenticator, user option
5. Exit

What kind of Work Admin can do?

1. An Admin can Add authenticator and Users
2. An admin can Update authenticator and Users information
3. An admin can delete authenticator and user information
4. An admin can see all user and authenticator list*

If I click **Add option** it will open add page in my project **AdminHome1.java**

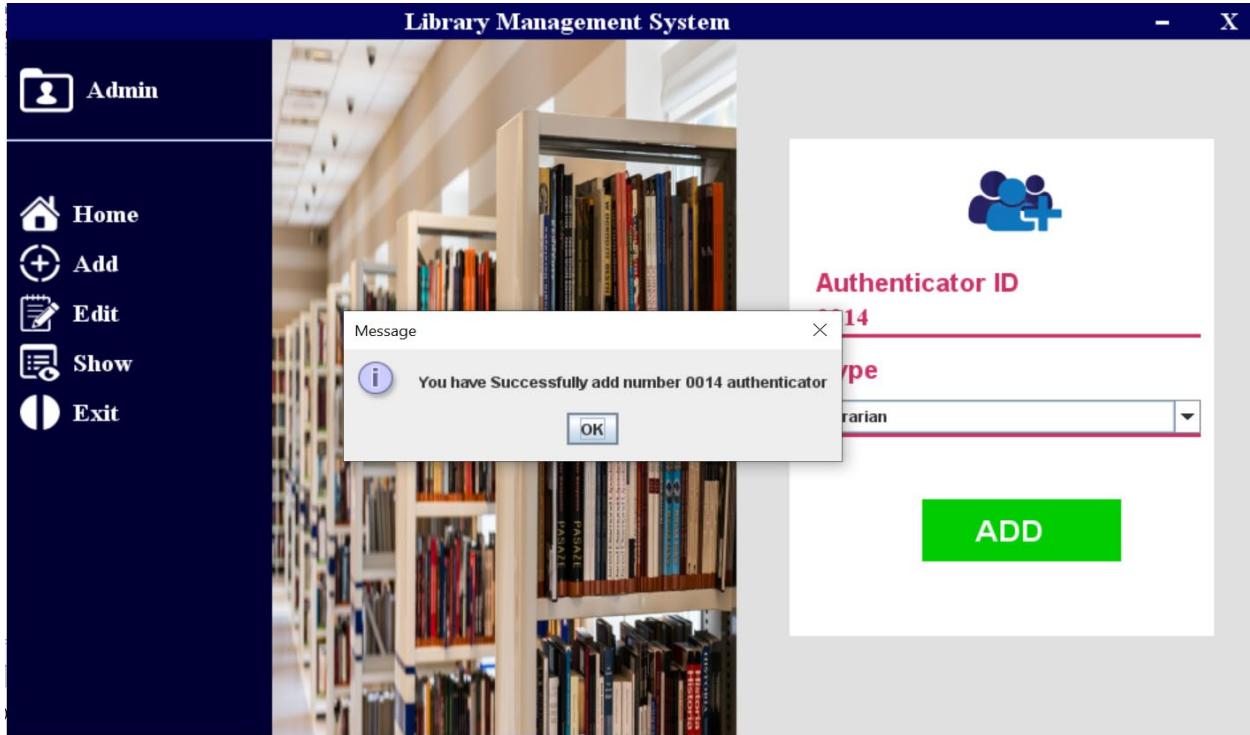


If I click **Authenticator** option, then Add authenticator page will be open. In my database **AddAuthenticator.java** is that page. Here has Authenticator ID and type. Id will automatic increment.

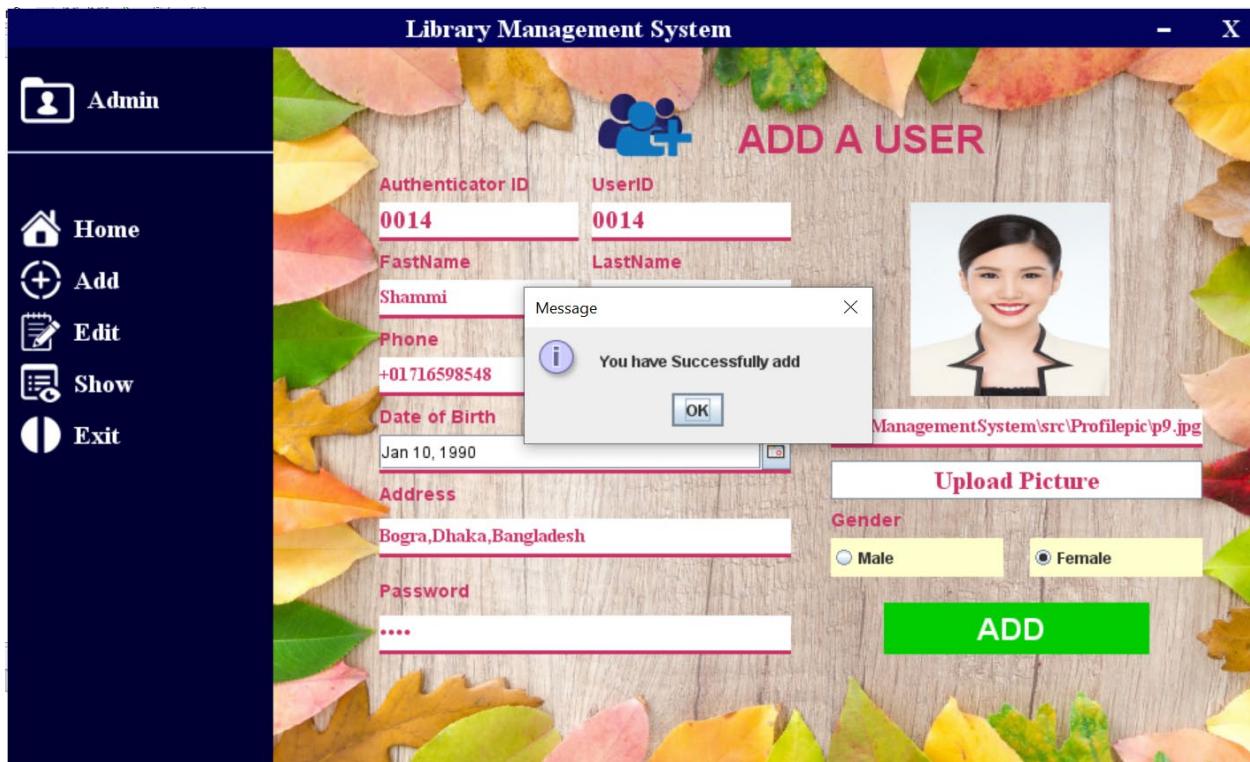
Has 3 type authenticator Librarian, Admin, Member. We have to select one type.then if click add option authenticator will successfully create.

AddUser: If I select the **User** option, then user add page will be open. In my project **AddUser.java** is ass user page. Here we must have to put authenticator id, so we have to insert a authenticator id that we already create. If we put a authenticator id that is not create yet it will detect automatic and tell the authenticator id is not valid. If we use a authenticator id that already use, It will print a message that

already this user ID use. And UserID will auto Increment. IF we not fill all the field it will give a message please fill up all the field.



After fill all the fill properly and also follow the rules(that I already discuss) then If we click in add button the user will successfully create and it will print a successfully add messages.



Edit page:

If I click **Edit** option It will open **edit page**. In my project **AdminHome2.java** is that page.

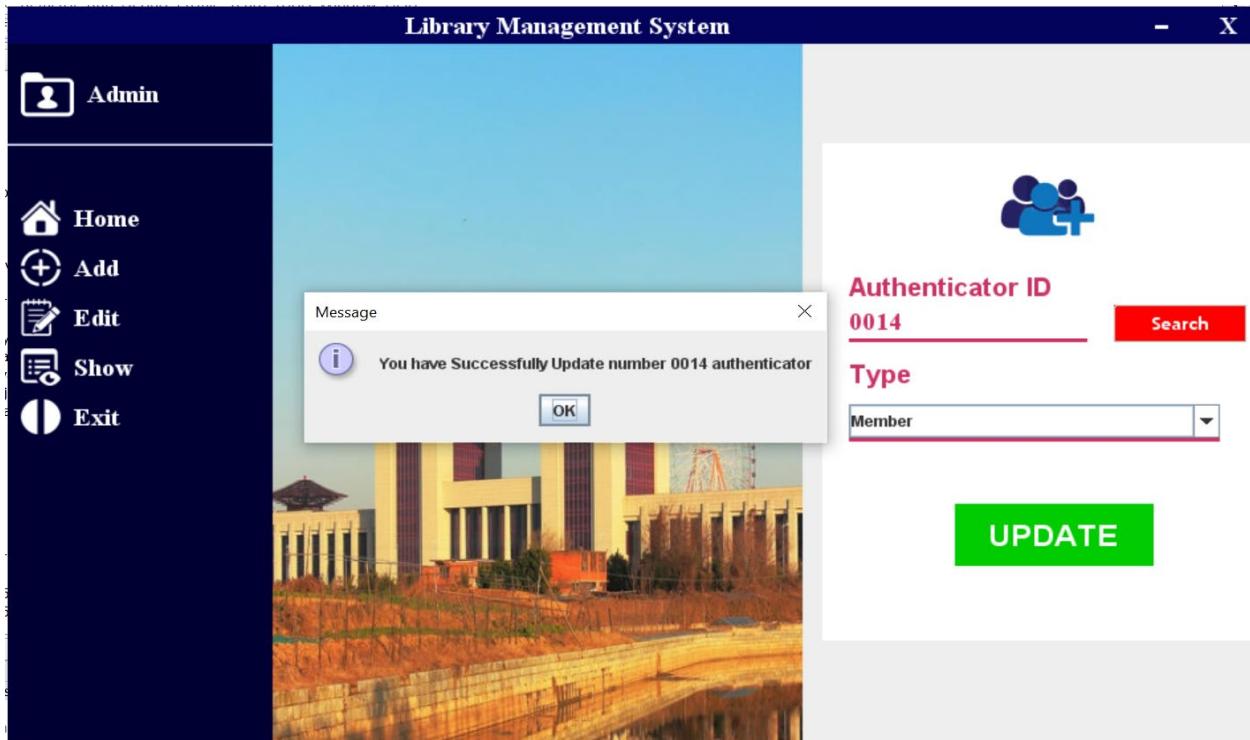
In this page here has tow part, One for Update authenticator and users and one for delete authenticator and users.

1. IF I click in update sector authenticator it will open update authenticator page.
2. If I click in update sector users it will open update users page.
3. IF I click in delete sector authenticator it will open delete authenticator page
4. IF I click in delete sector users it will open delete users page.



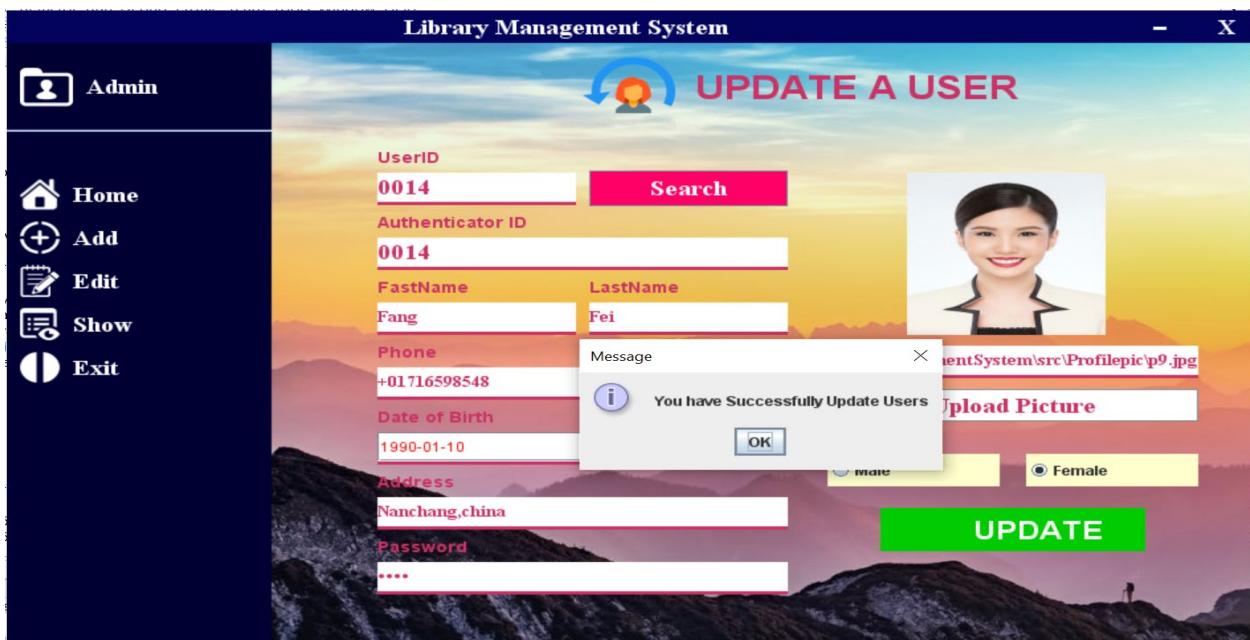
Update Authenticator: In my project file **UpdateAuthenticator.java** is **Update Authenticator page**.

At first, we can write our authenticator and click search option. It will auto detect that authenticator type. After we can change the type. And click update option. Then the authenticator type will be change.



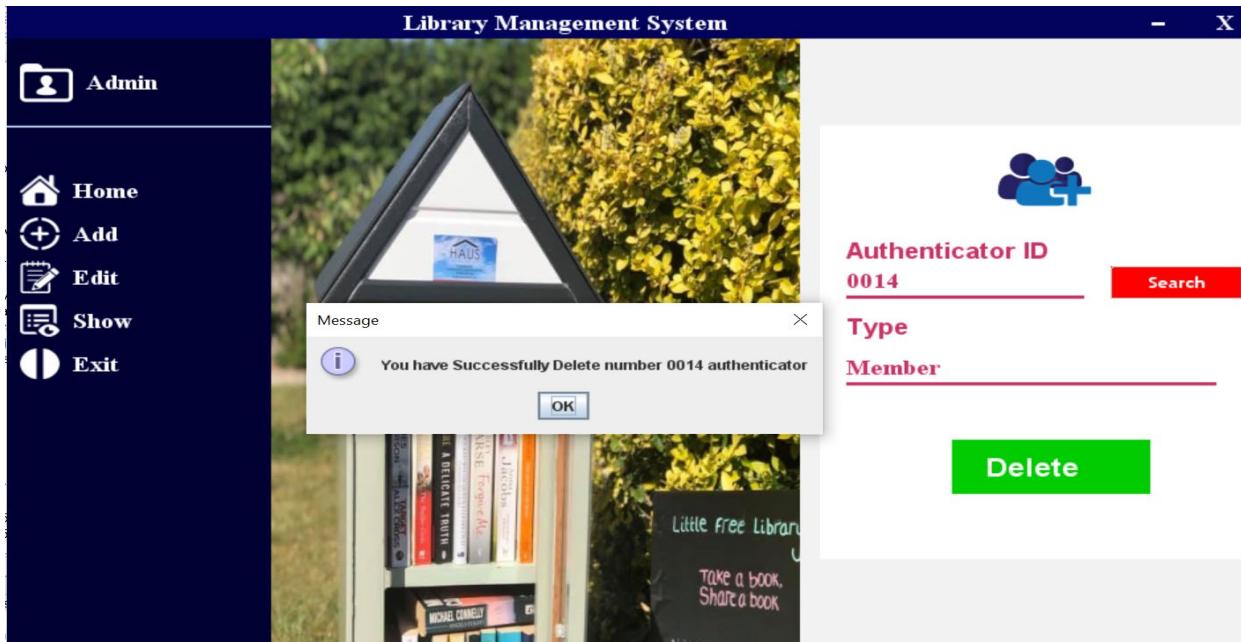
Update User : In my project **UpdateUser.java** is update user page.

By using this page admin can update user information. If insert the user id and click search the all filed will automatically fill. And if the user Id is not available It will print a message that the user id is not available. In here I update only name. but using this admin can update all the user information.
Note(Without user id.)

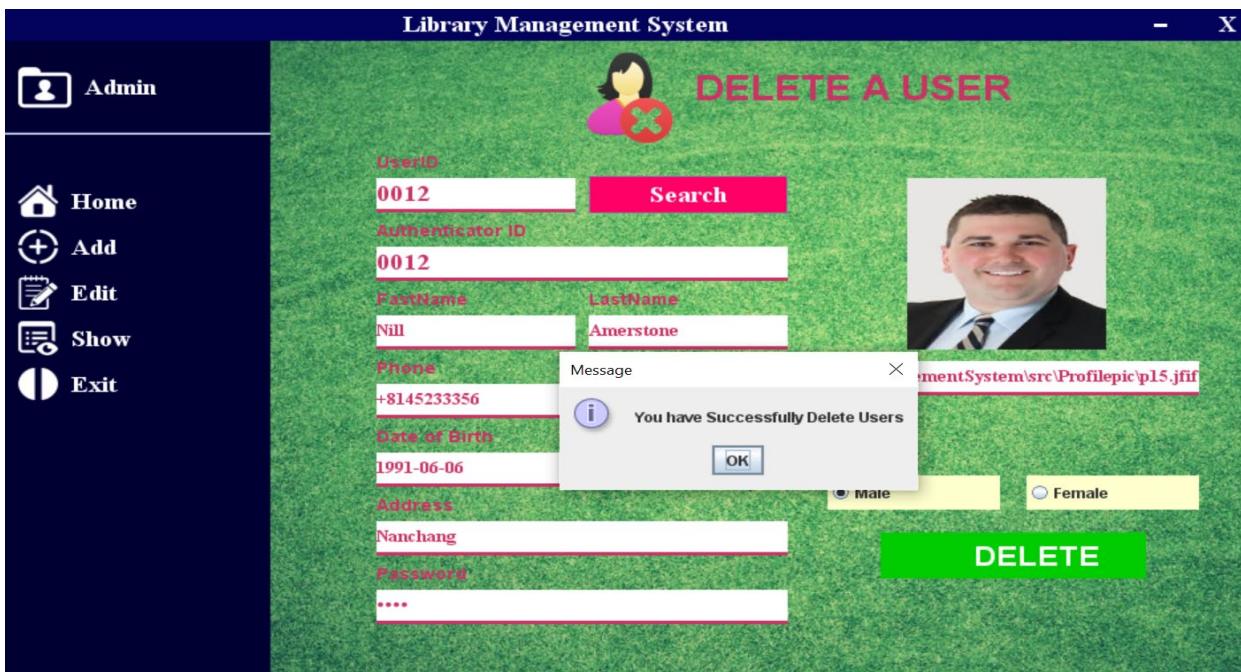


Delete Authenticator:

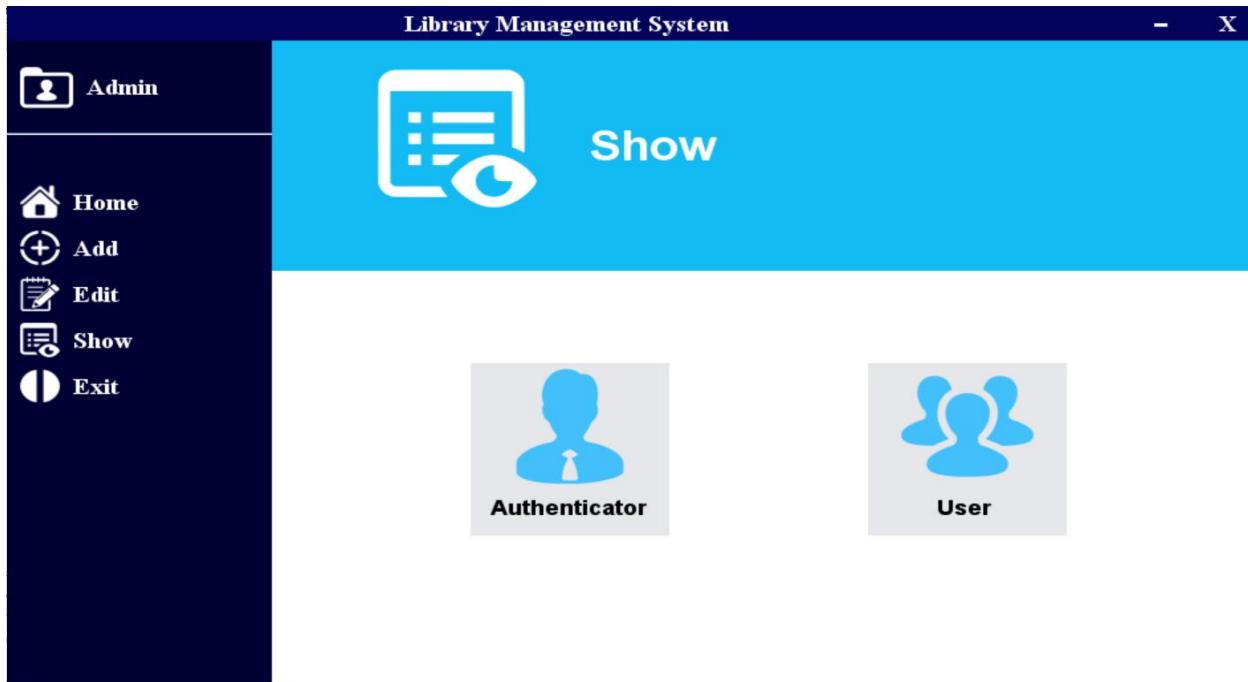
By using this option admin can delete authenticator. In my project `DeleteAuthenticator.java` is that page. Insert authenticator id and click search.it will fill the type automatic not available It will print a error messages.



Delete Users: By using this option admin can delete users. In my project `DeleteUser.java` is that page. Admin write UserId click search button it will show that userId users all the information, If the id is available. Now if admin click delete button the user will be delete. And also, one thing if the user has member card, the user will borrow any book or reserve book, it will remove the user from all the table. Because this user id has used, they're as a foreign _key.



Show Option: In bellow I show you show menu , In my project **AdminHome3.java** this is that page. Here has user and authenticator. If we click users, It will users list. If click authenticator It will show authenticator all list that admin create,



Authenticator List: In my project **ShowAllAuthenticator.java** is this class.

If admin click show All button. It will show all the authenticator that already have created. If admin do not create any authenticator it will print, "You do not create any Authenticator" message.



Users List: If admin click Show All button, It will show all the user list, If admin create any user. In my project `ShowAllUser.java` is that class.



Now I will discuss librarian work



Here has 5 option.

1. Home page
2. Add Page: Here has add publisher, book, category, Bookcategory, membership option.
3. Edit: Here has delete and update publisher, book, category, Bookcategory, membership option.
4. Show: Here has Shown publisher, book, category, Bookcategory, membership option.
5. Issue Book
6. Return Book
7. Reserve list
8. Exit

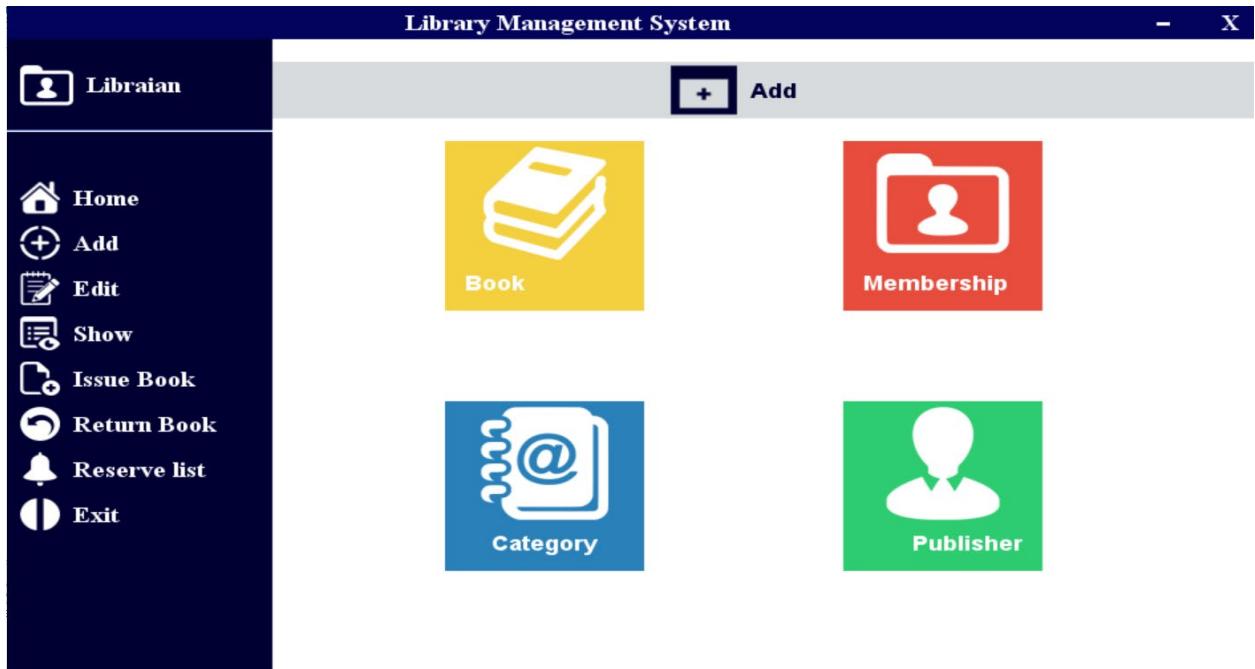
What kind of Work Librarian can do?

1. A librarian can Add Publisher, book, category, bookcategory, membership.
2. A librarian can Update Publisher, book, category, bookcategory, membership.
3. A librarian can delete Publisher, book, category, bookcategory, membership.
4. A librarian can issue book, return book.
5. A librarian can see Publisher, book, category, bookcategory, membership list.
6. A librarian can see reserve book list.

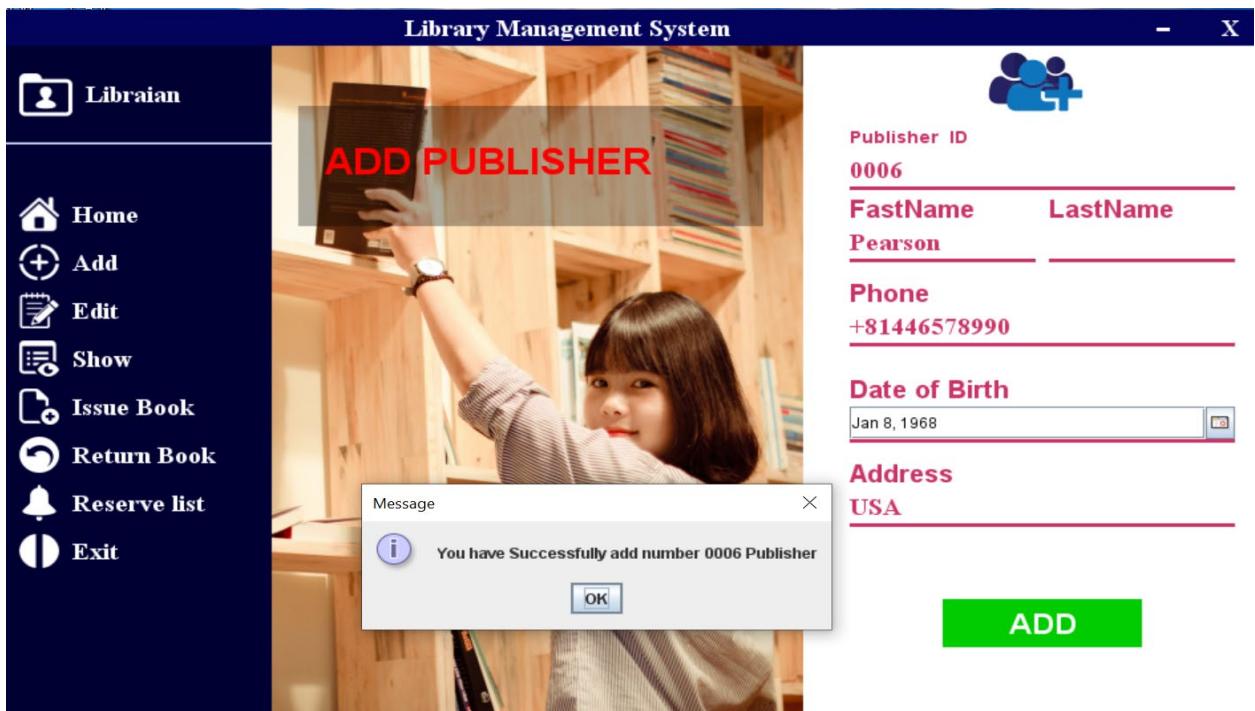
Librarian Home: In my project **Librarian1.java** class is Librarian home page. And This class is extended on JFrame. And other class is extended in JPanel that use in here. And we add those in card layout.

Librarian Add access sector:

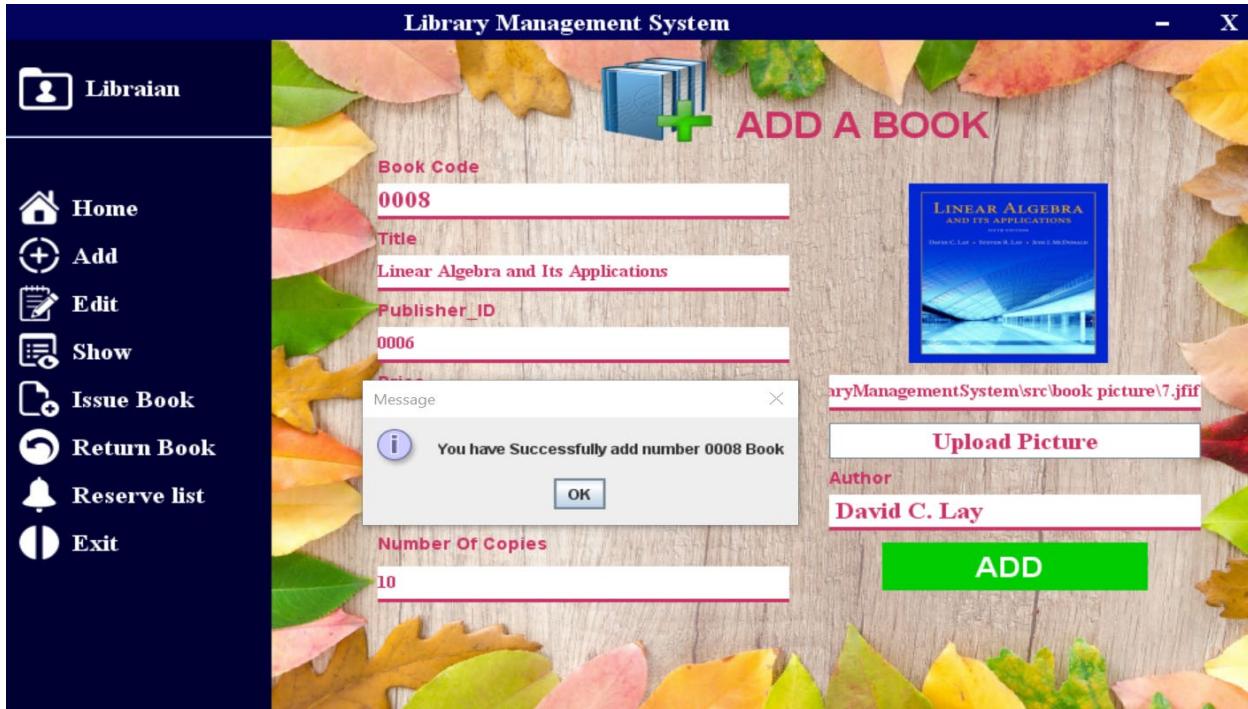
Add Menu: In my project LibrarianHome1.java is this page. Under this page has 4 option.



Add Publisher: by Click Publisher Librarian can add publisher. It will check the publisher already have or not in database. If do not have, it will successfully create users. And remember before add any publisher book, you have to that publisher. Because it will use book table again. In my project **AddPublisher.java** is that page.



Add Book: By click book menu a librarian can add book. And here has publisher id that is foreign keySo, add book, please notice this publisher have in publisher table. Book code will auto increment.If click add button book will add successfully. In my project `AddBook.java` is that page.



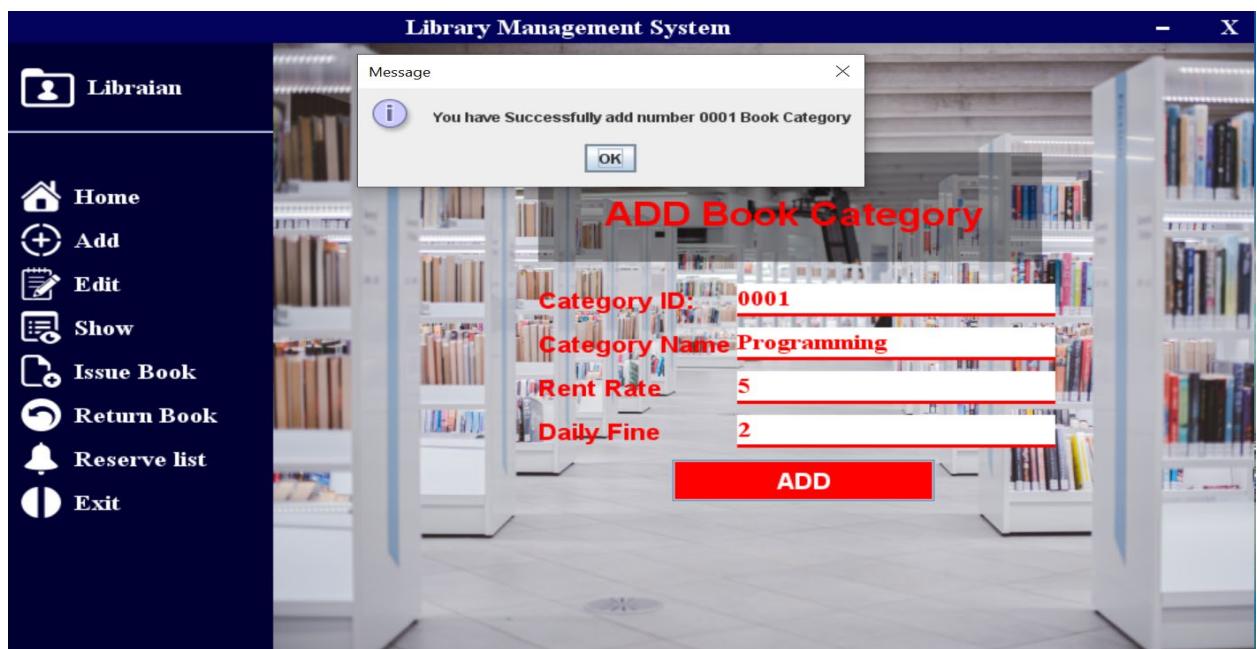
Add Membership: librarian can add membership. The card number will auto increment. And after fill all the information it will check the user id already have or not. If the user has then it will check the member card already create or not. If it no creates yet, it will successfully create. In My project `AddMembership.java` is that page.



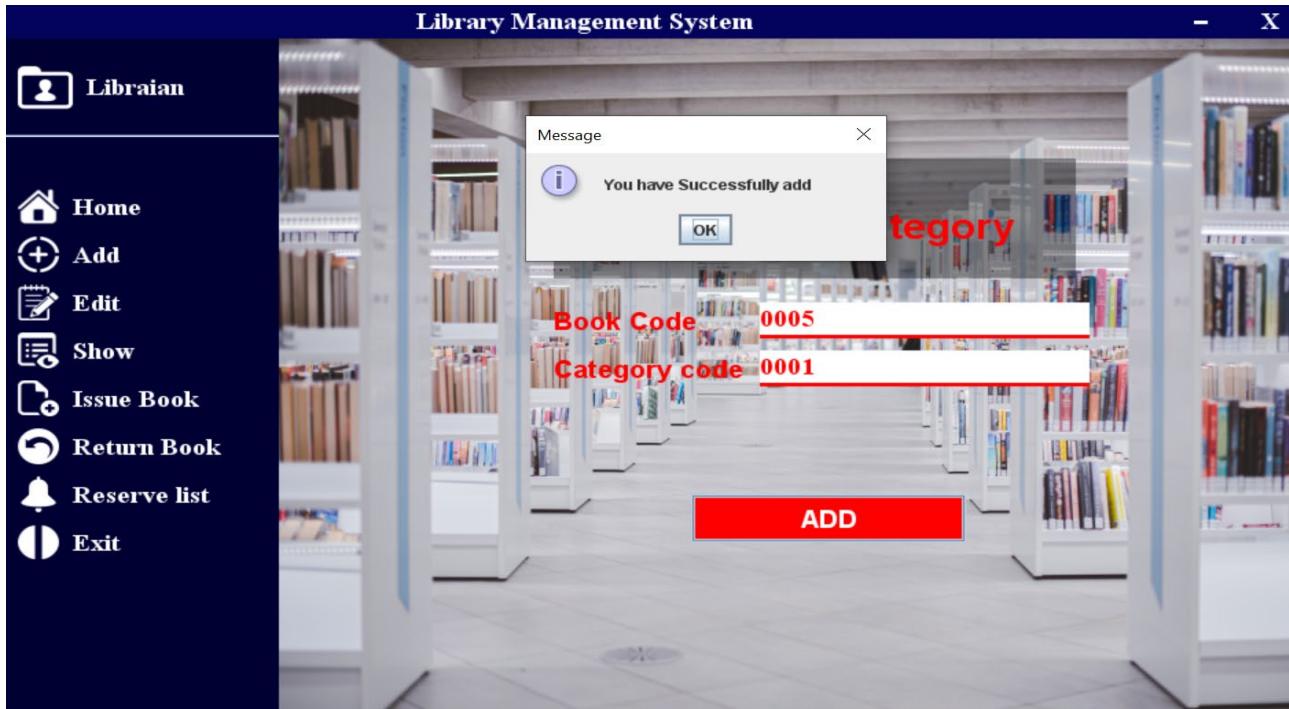
Category: If click category now it will open new window here has add category and add book category option. LibrarianHome4.java class in project is show that.



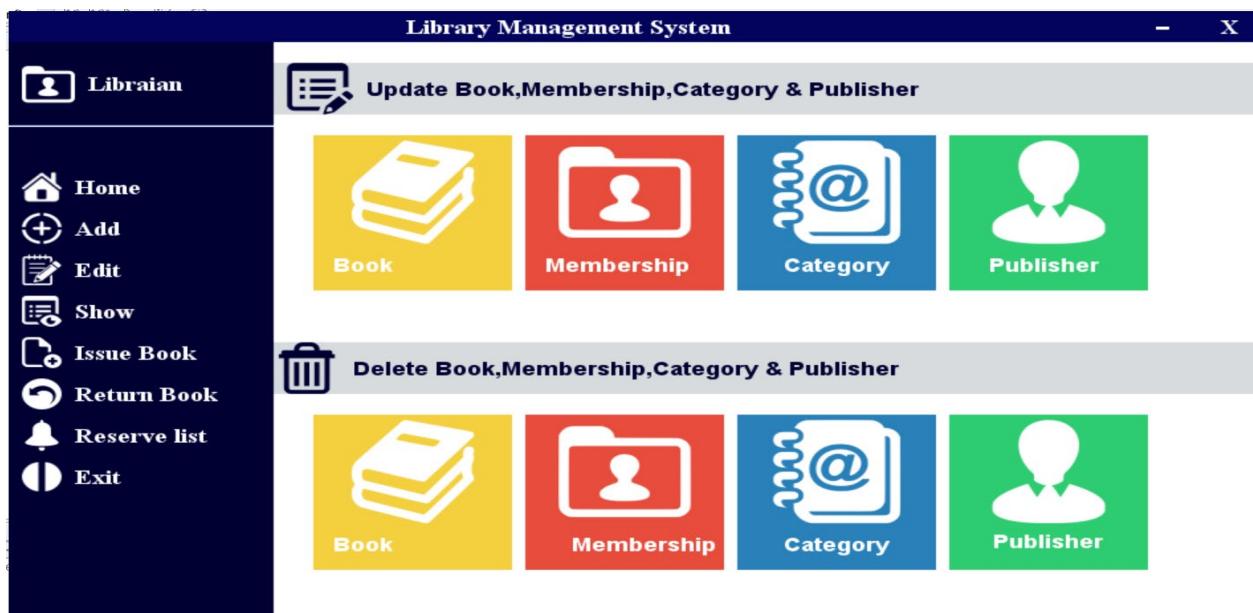
Add Category: By using this option a user can add a category. In my project AddCategory.java is that page.



Add Book Category: If the book is available and the category code also available the book category will successfully create. `AddBookCategory.java` is show add book category.

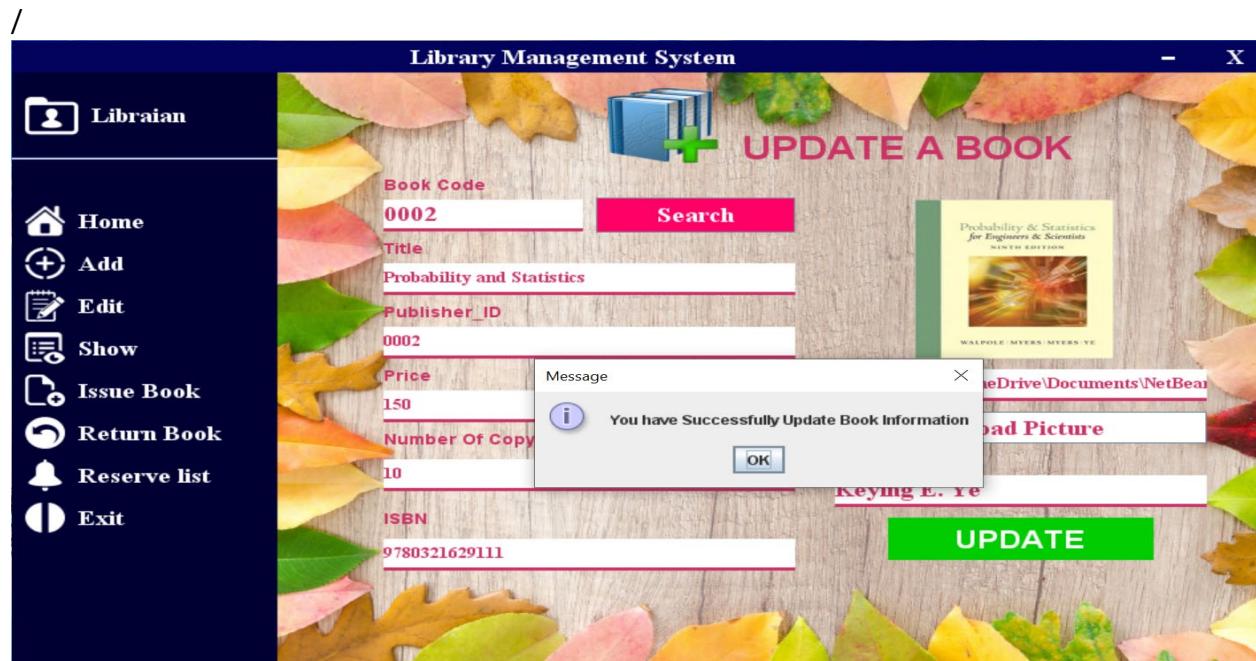


Edit Menu: If Librarian click in edit menu then this page will be opened. Here has two part update and delete. By using update sector option Librarian can update category, book, publisher, membership, bookcategory. And by using delete user can delete category, book, publisher, membership, bookcategory. In my project `LibrarianHome2.java` is this page.

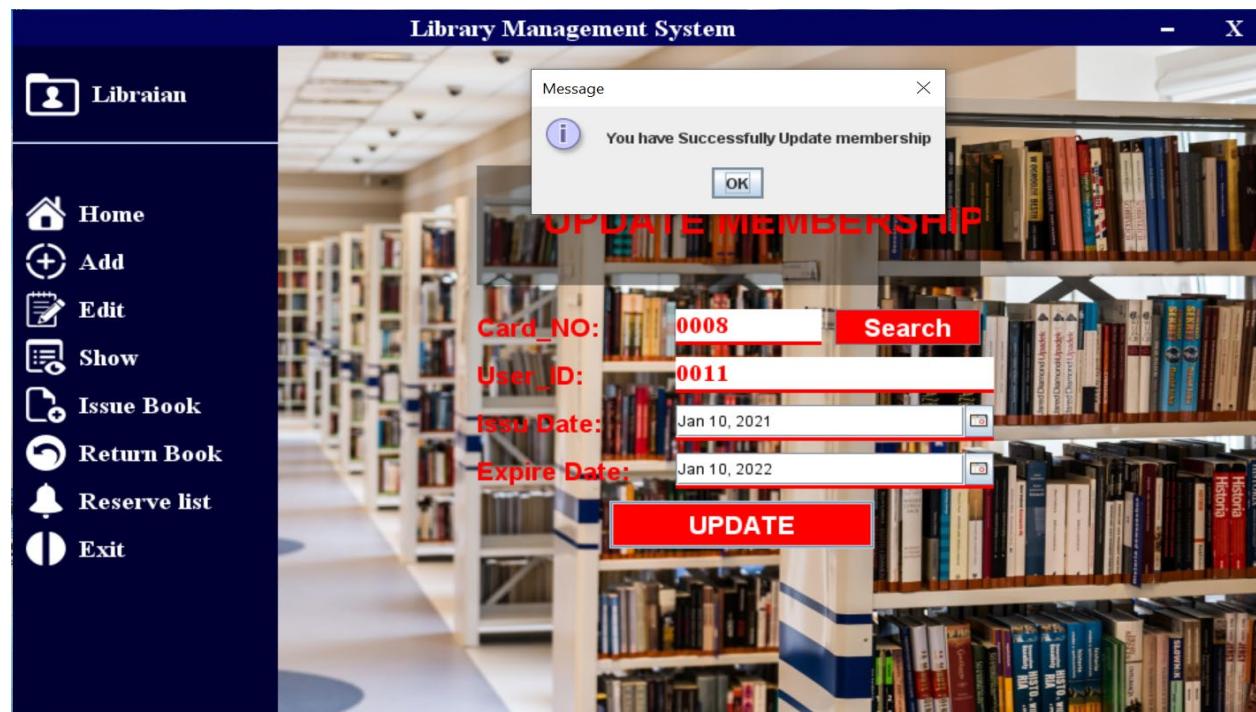


Librarian Update access sector:

UpdateBook: By using this option a Librarian can update book details. If he just write book code and click search the book all information will be fill automatic. If the book code does not exists it will print error messages. That "This Book does not have". In my project **UpdateBook.java** is that page.



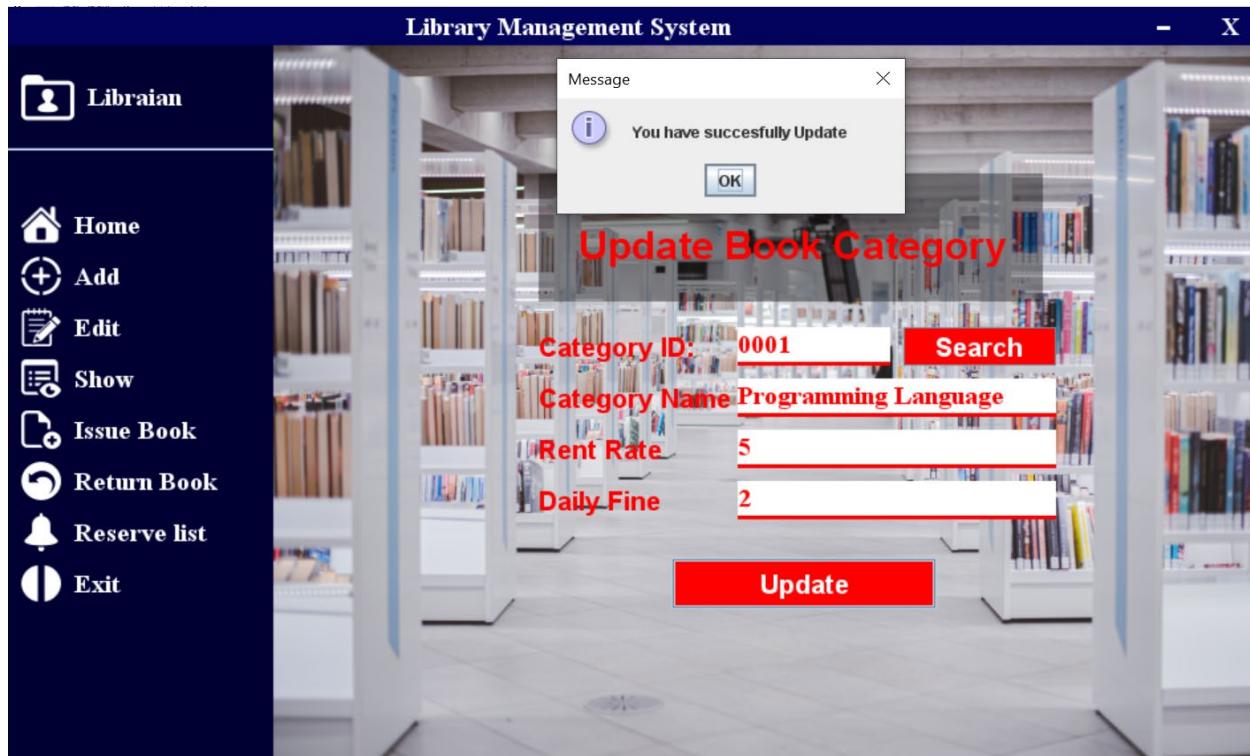
Update Membership: by using this option a Librarian can update Membership information. In my project **UpdateMembership.java** is this page.



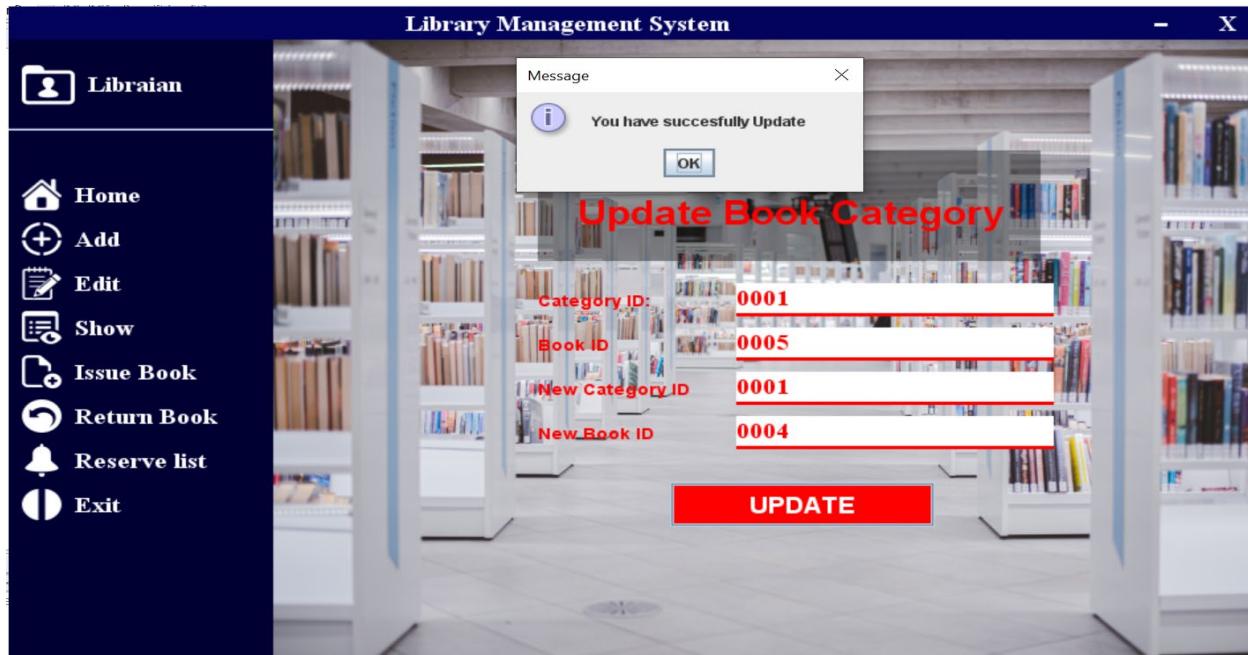
Update Category: If click this option It will open another page. That is `LibrarianHome6.java` will open.



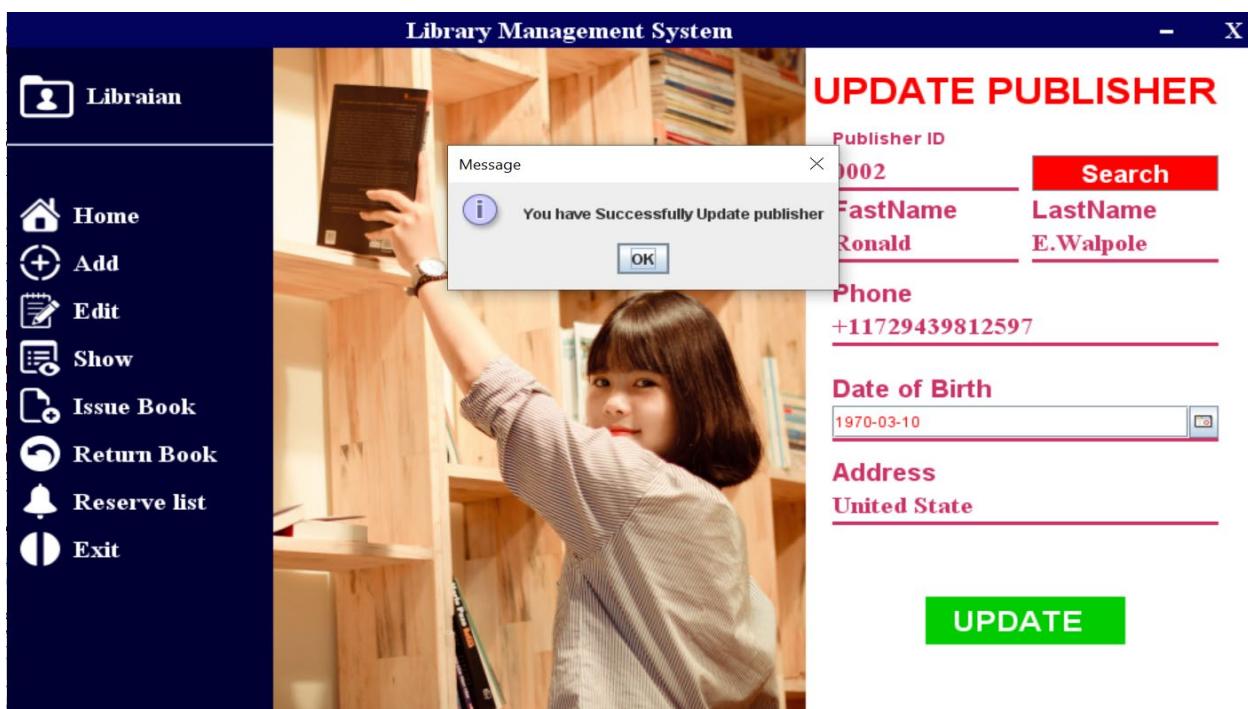
Update Category: By using this option a librarian can update category name, rent rate daily fine.In my project `UpdateCategory.java` is that page.



Update BookCategory: here librarian should have to write old category ID and book code first. If this book category exists, then it will check the new category have or not in category table from the database if it also have then it check the book Id that means book code has or not in book table. If it also have then a librarian can update book category details. In my project **UpdateBookCategory.java** is that pages.

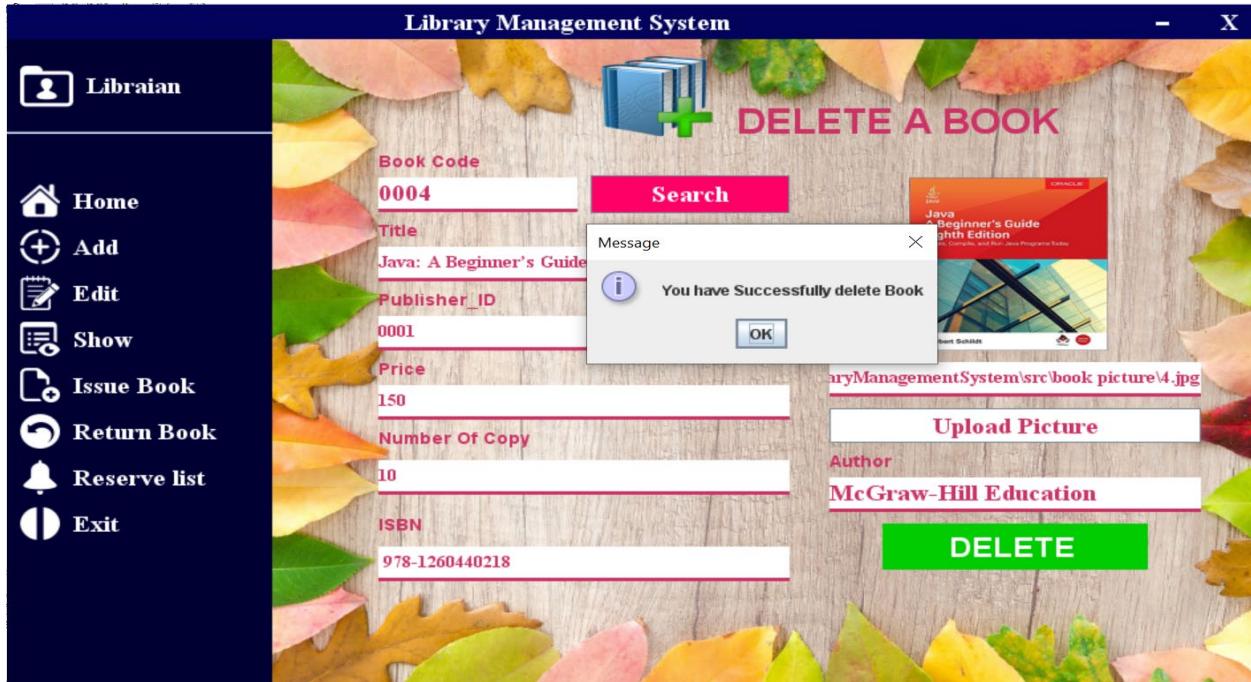


Update Publisher: By using this option a Librarian can update publisher details.

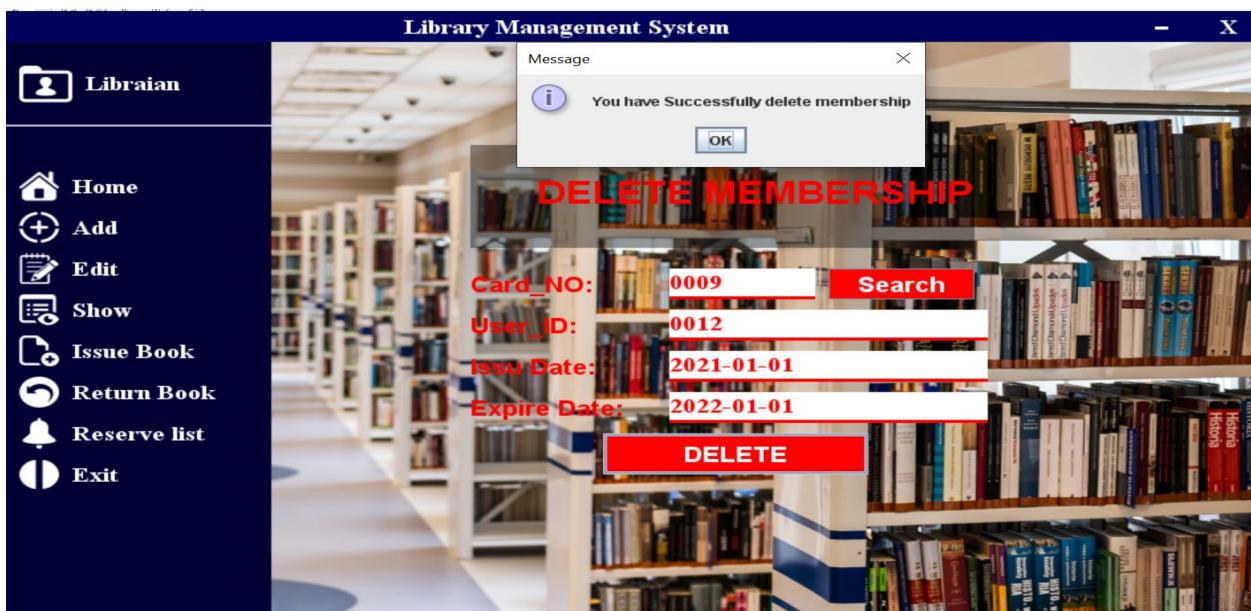


Librarian Delete access sector:

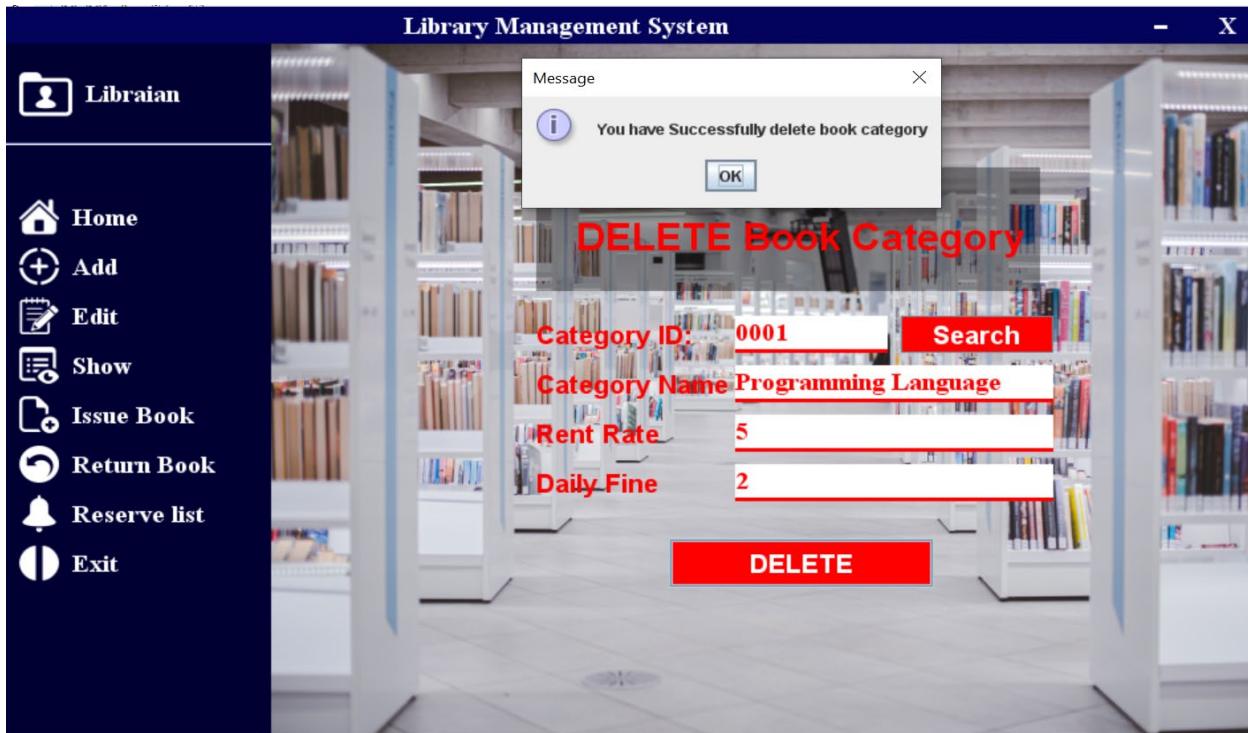
Delete Book: Here librarian can delete book. Here at first check the book have or not in bookcategory Table, if have it will delete from this table. After If will check memberbook table, If there has this book, It also delete from there, after it will check reservations table, if there also has, It will delete from there, at the end it will check book table, and delete from it. We have to delete book from multiple tables because it uses there as a **foreign key**. In my project `DeleteBook.java` is that page.



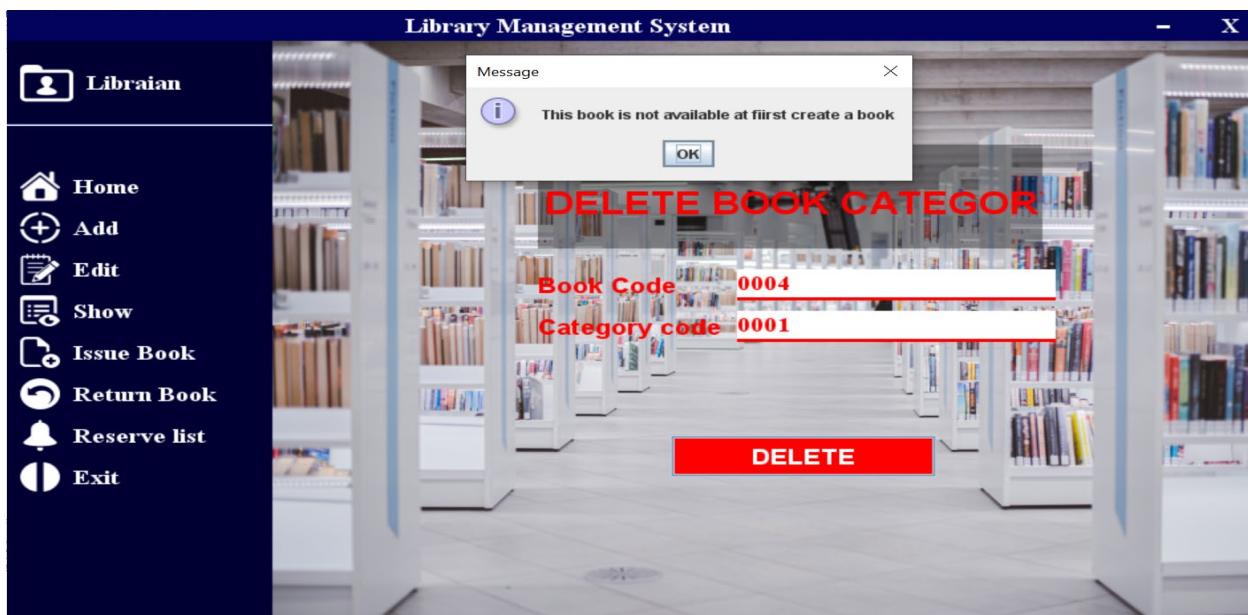
Delete Membership: By this option a Librarian can delete membership. In my project `DeleteMembership.java` is this class.



Delete Category: by this option a Librarian can delete category. After write the category id if click search button. it checks category table this category id is available or not. If have all the field will fill automatically. If click delete button, Then it will check bookcategory table the category id have or not . If have it will delete from there. After it checks category table this category id is available or not. If have it will delete from category. In my project DeleteCategory.java is that page.

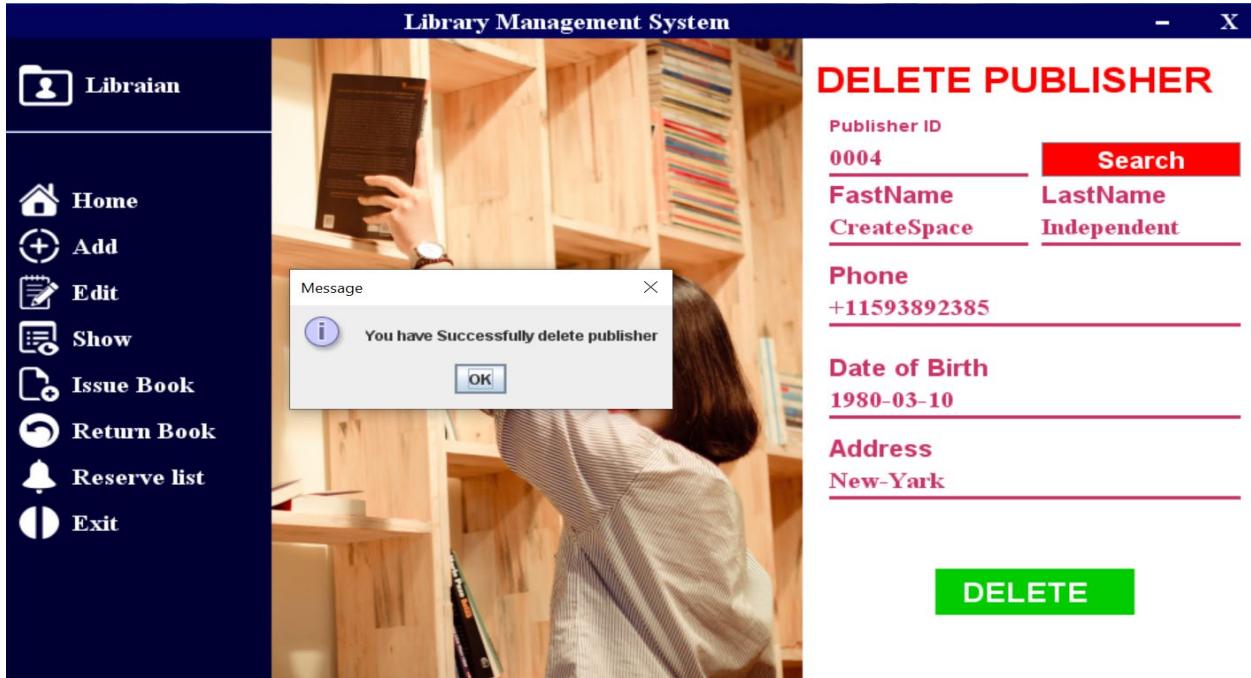


Delete BookCategory: At first Librarian have to fill category id and book code. After if click delete button the system will check the BookCategory is available or not. IF it is available, It will delete successfully. In my project DeleteBookCategory.java



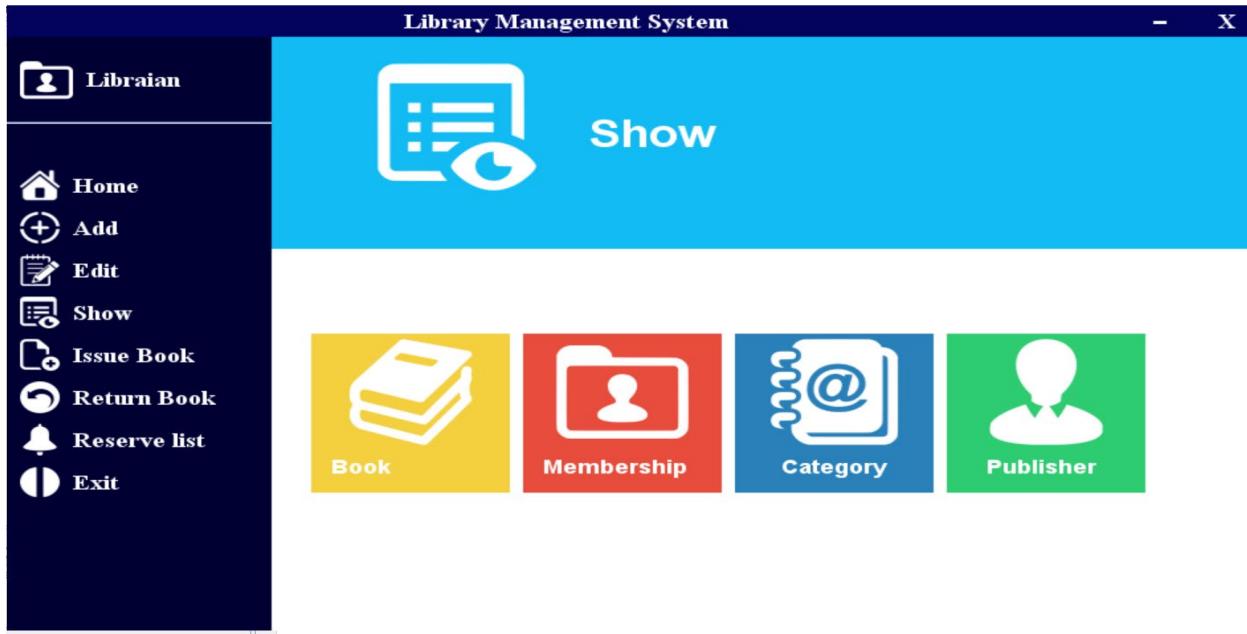
DeletePublisher: If librarian want to delete publisher at first he have to check book table, is there has any book that published by this user. If it have at first he have to delete it from book table. After he can delete the publisher from publisher table. Because the publisher id is a foreign key in book table.

In my project **DeletePublisher.java** is that class.



Librarian Show access sector:

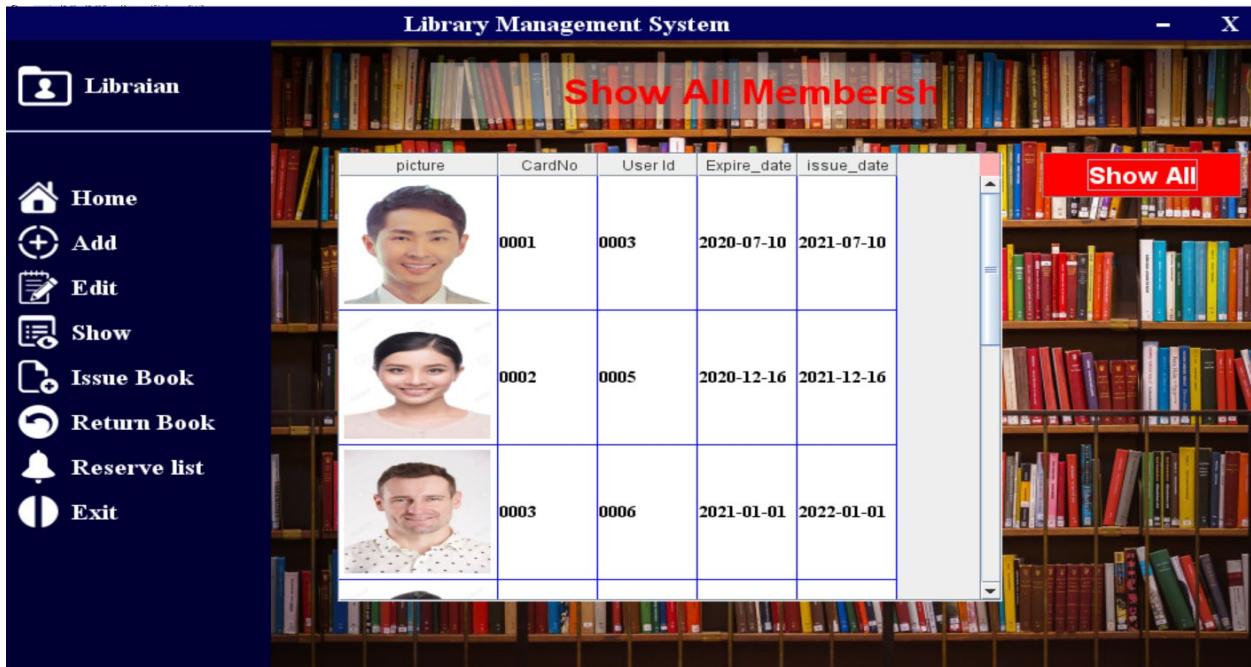
Librarian can see all the book list, Publisher list, category, bookcategory and membership. In my project **LibrarianHome3.java**



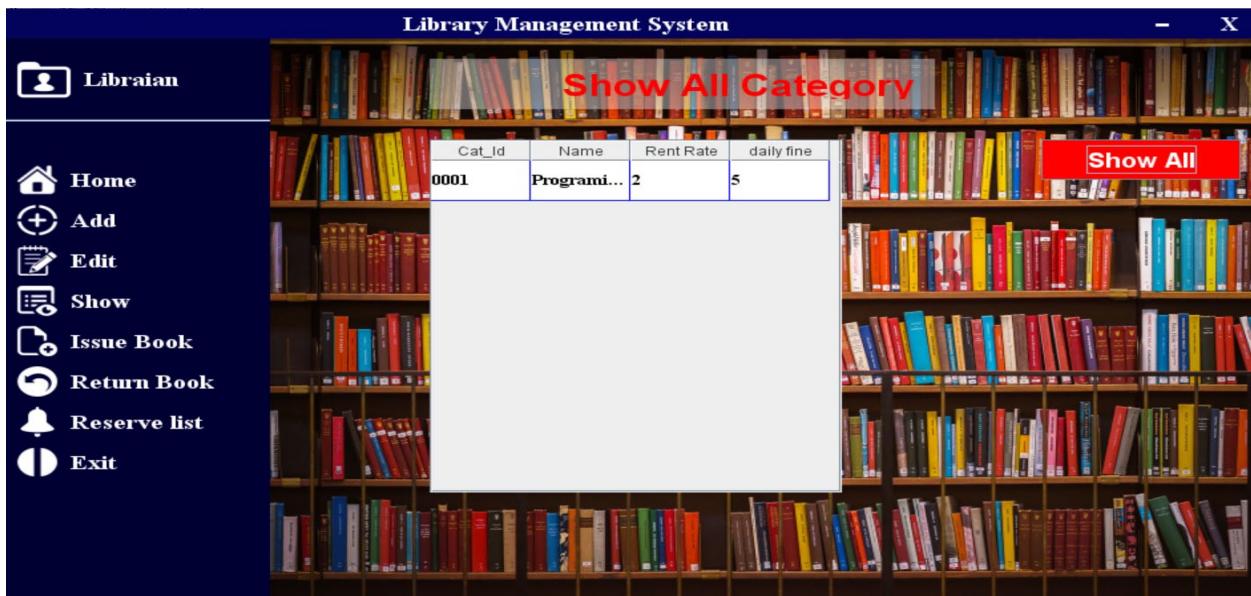
Show all Book: If click show all book all the book will show. If there has any book in book table. In my project ShowBook.java is this class.

Show All Book								
Picture	ID	title	publisher_...	Catrgory N...	price	numberofc...	isbn	
	0001	Database ...	McGraw ...		100	10	97800735...	
	0002	Probabilit...	Ronald E....		150	10	97803216...	
	0003	Compiler...	Pearson ...		150	10	978-0201...	

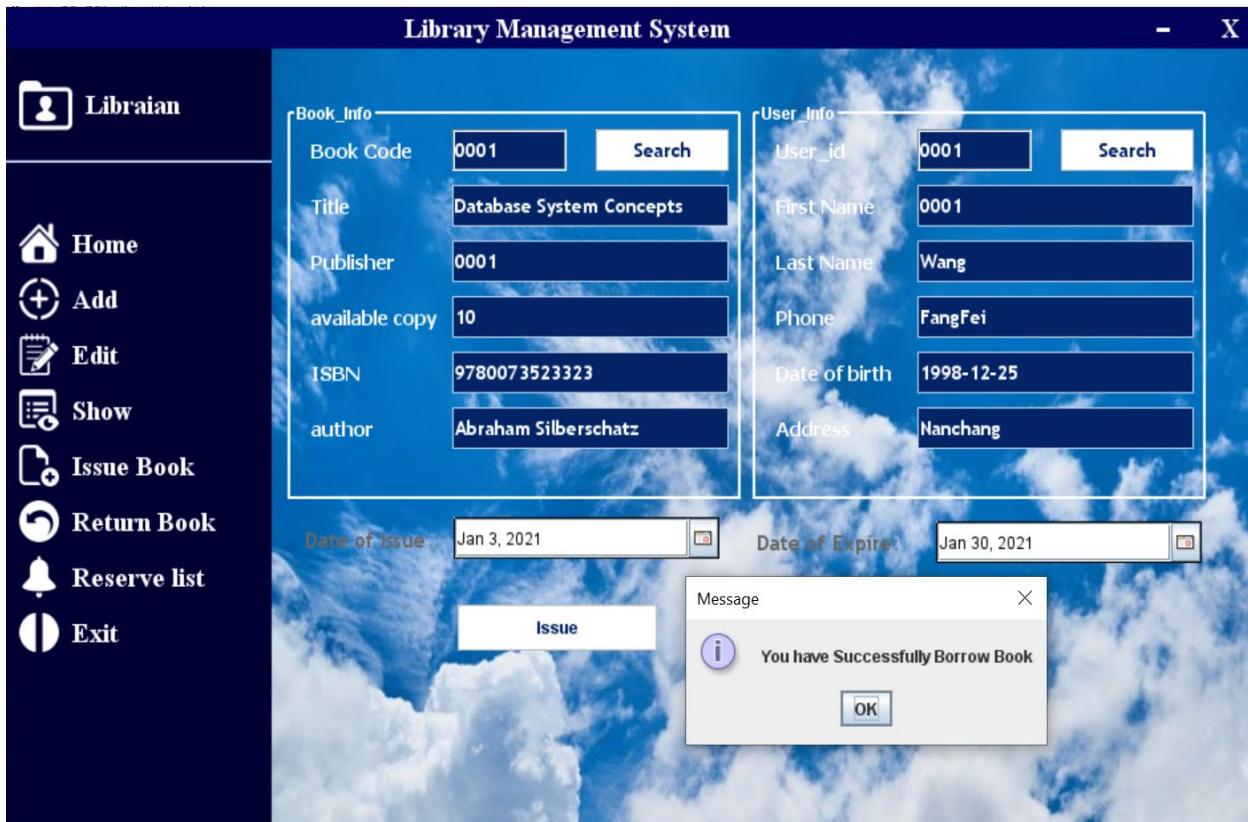
Show Membership: If click show all, all the membership will show. If there has any membership in membership table. In my project **ShowMembership.java** is this class.



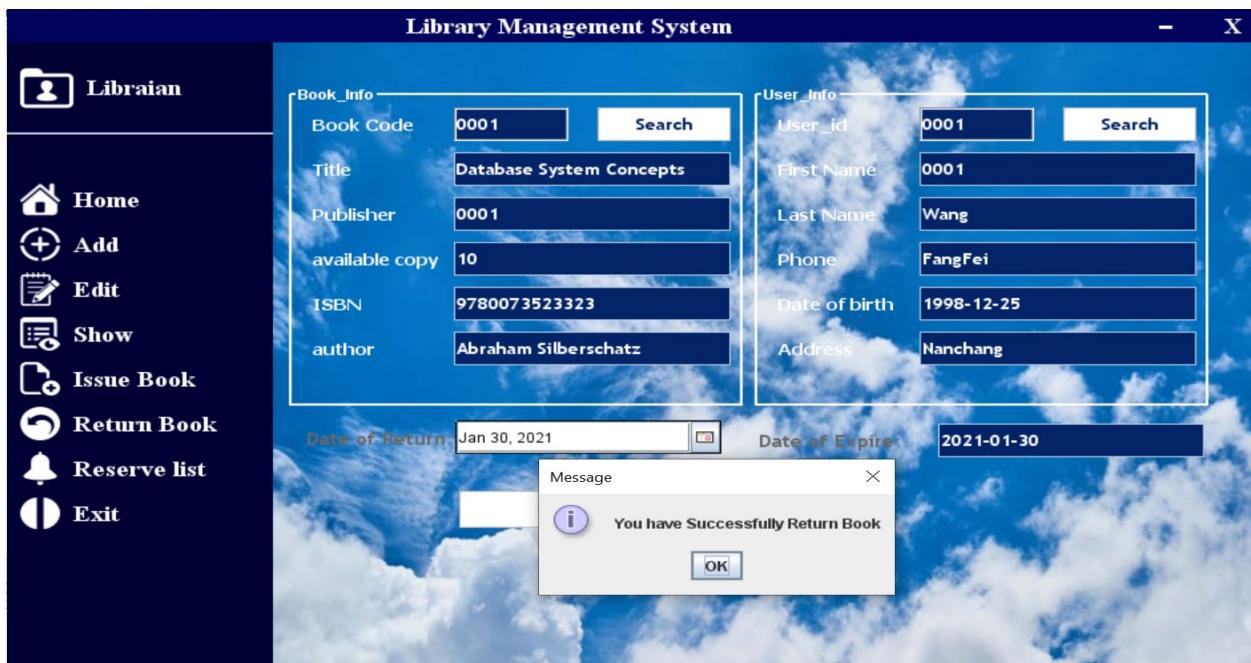
Show Category: If click show all, all the category will show. If there has any category in category table. In my project **ShowCategory.java** is this class.



Issue Book: Librarian can issue book. by search it will check the user id and book code available or not. If it available then it will check the book already take this user or not. If not take then it will successfully issue. In my project **IssueBook.java** is this class.



Return Book: If any user take book and come to return it. Then the librarian can return BOOK BY THIS window. At first it will check the book status is it really borrow or not. If it borrows by this user then I will successfully return.



Reserve list: Librarian can see all the book reserve list. And he can approve it, and make it issue. Librarian have to select table row that he wants to approve. And click approve he can issue it by click approve.

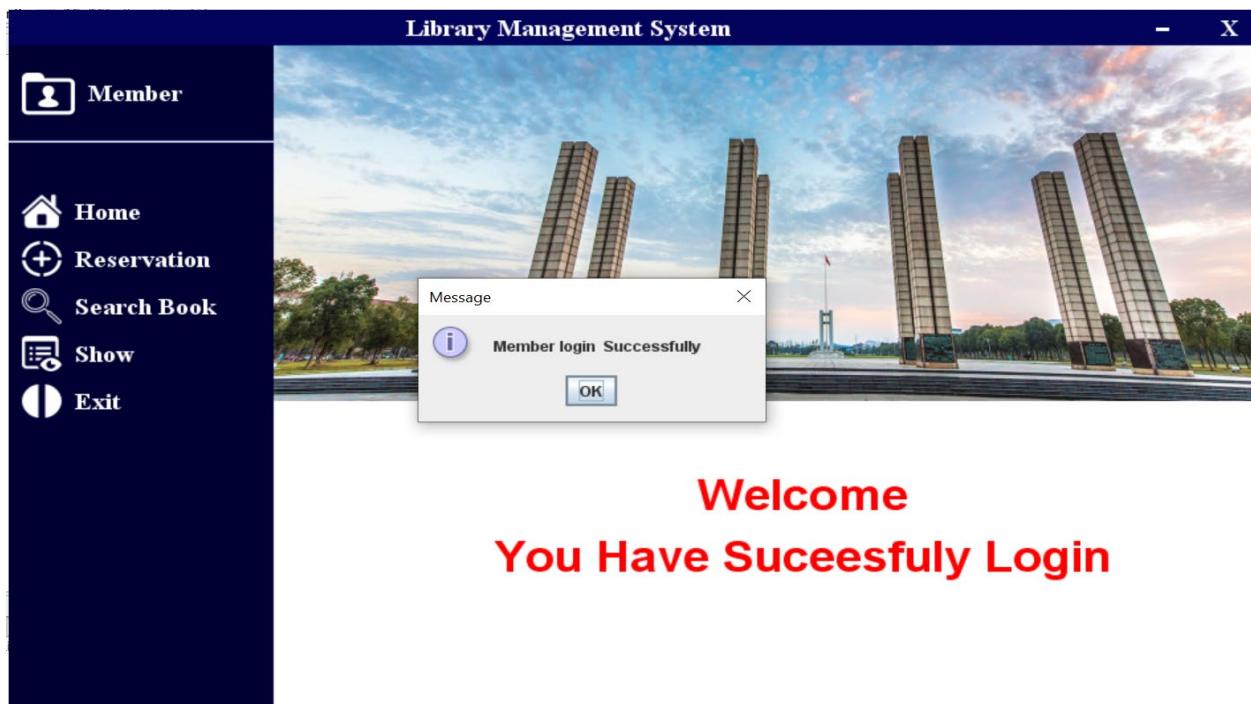
Reserve List						Aprove	
Reserve_Id	U_ID	Book_code	Reserve_d...	Pick_date	Status		
0001	0003	0001	2021-02-01	2021-01-01	Not Aprove		
0002	0005	0002	2021-02-05	2021-01-05	Not Aprove		
0003	0006	0003	2021-02-04	2021-01-04	Not Aprove		

Reserve Successfully



Member Access

IF member successfully login then the bellow page will open. In my project this name is **Member1.java** And this is extended on JFrame.



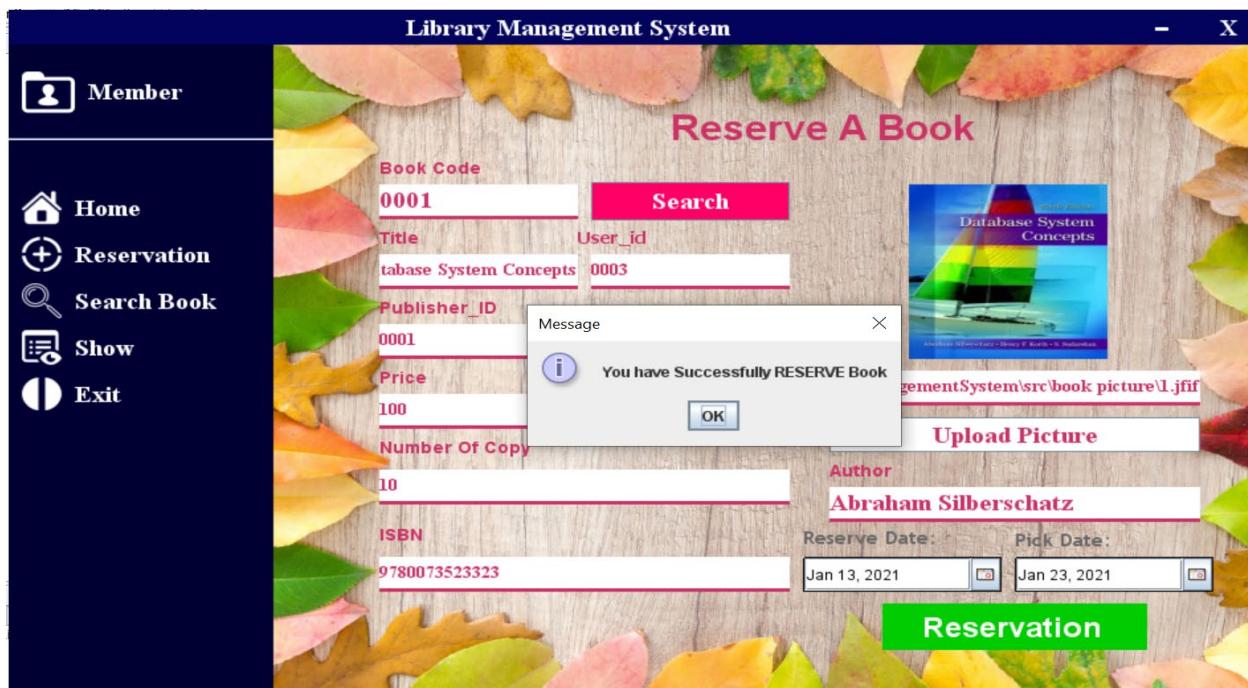
Here has 5 option.

1. Home page
2. Reservation Page: Here member can reserve book.
3. Search Book: Member can search book.
4. Show: Member can see all the that have in this library.
5. Exit

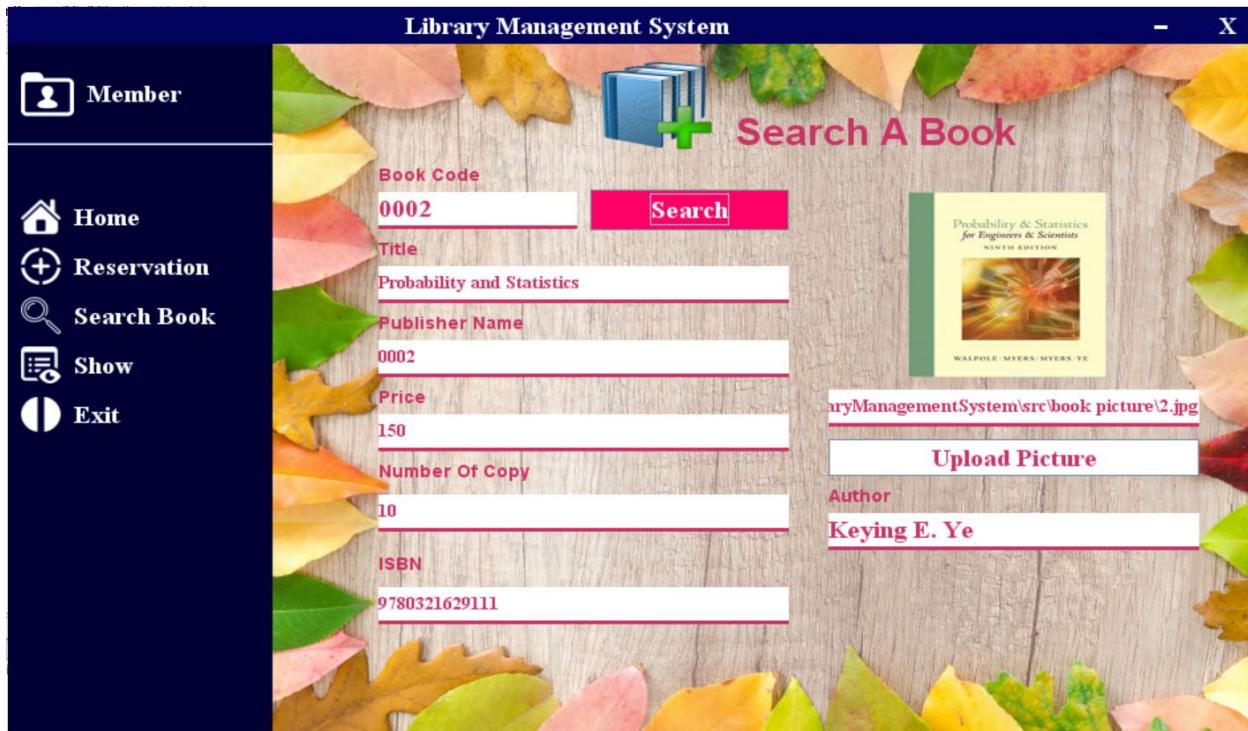
What kind of Work Member can do?

1. A member can reserve book.
2. A Member can search book.
3. Member can see all the that have in this library

Reservation: a member can reserve book. Member have to write book code then click search button. If the book code is valid then The book information will automatic. IF click reservation then the book reservation will successfully.



Search Book: Member can search the book-by-book code. If the book code is valid it will show all the information that related in book code.



Show Book: Member can see all the book that have in library list.

Show All Book								
Picture	ID	title	publisher_...	Catrgory N...	price	numberofc...	isbn	
	0001	Database ...	McGraw ...		100	10	97800735...	
	0002	Probabilit...	Ronald E....		150	10	97803216...	
	0003	Compiler...	Pearson ...		150	10	978-0201...	

