**Contents**

[ABSTRACT-----------------------------------------------------------------------------------3](#_Toc12672031)

[Background-----------------------------------------------------------------------------------------3](#_Toc12672032)

[Introduction---------------------------------------------------------------------------------3](#_Toc12672033)

Proposed system of library management system---------------------------------------4

Functional Requirement-------------------------------------------------------------------5

Use case Diagram--------------------------------------------------------------------------------------------12

Sequence diagram---------------------------------------------------------------------------13

Statechart diagram-------------------------------------------------------------------------------------------14

Class Diagram---------------------------------------------------------------------------------------------------15

[Main Class--------------------------------------------------------------------------------------16](#_Toc12672043)

Login class----------------------------------------------------------------------------------------------17

[User class-----------------------------------------------------------------------------------------------](#_Toc12672045)22

[Admin menu class------------------------------------------------------------------------------------2](#_Toc12672046)5

Output-------------------------------------------------------------------------------------------39

**Abstract**

The library management system aims in developing a computerized system to maintain all the daily work of the library. This project has many features that are generally not available in normal library management systems like facility of user login and a facility of teacher’s login. It also has a facility of admin login through which the admin can monitor the whole system. It also has the facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board. It has also a facility where students after logging in their accounts can see a list of books issued and its issue date and return date and also the students can request the librarian to add new books by filling the book request form. The librarian after logging into his account i.e. admin account can generate various reports such as student reports, issue reports, teacher reports, and book reports. Overall this project of ours is being developed to help the students as well as the staff of the library to maintain the library in the best way possible and also reduce the human efforts.

# **Background**

Library management practice is evolving to provide service through computerized system, with an increasing emphasis on technology focused service as compared to being primarily manual focused. Service is not based on a single transaction or episode, but rather includes multiple interactions with a user for several times to select book , register book etc. Each time user want to access library service, he must be there to enjoy the service. But customer in different part of the city may not have enough time to go there and enjoy the service and many libraries has been losing a lot of readers because of following manual based library management system

Introduction to library management system

Library management system is all about organizing, managing the library and library-oriented tasks. It also involves maintaining the database of entering new books and the record of books that have been retrieved or issued, with their respective dates.

The main aim of this project is providing an easy to handle and automated library management system. This project also provides features and interface for maintaining librarian’s records, student’s history of issue and fines.

The owner can easily update, delete and insert data in the database with this project.The following are some of the features provided by this project:

The issue of books by online mode.

Columns provided to search book online.

Requests to the librarian can be sent to provide new books in the column.

Login portal for a student for security purpose of the library.

Homepage for the student which has different buttons to navigate to pages containing the date of issue, date of return, fine charges etc.

Columns for teachers to get book issued if desired.

Requests column for teachers to ask for the introduction of new or essential books in the library.

Maintaining records of the librarian and other library staff.

Feedback page to be viewed and monitored by the college authority, to which student

The teacher can mention any misbehave or feedback for the library.

PROPOSED SYSTEM OF LIBRARY MANAGEMENT SYSTEM

The project Library Management System aims at developing a fully functional computerized system to maintain all the day to day activity of a library

This project has many features which such as the facility of user login and teachers login. Also on the top of all this, there is an admin who will be managing the entire application’s authorization and authentication, not any intruder can login and modify the data, as a login for admin is also available.

The various modules of this system areas:

User Module:

This module is further divided into various sub-modules describing the user in a better way:

New user register:

To sign up a new user to this system

Student Login:

So as to confirm that only an authenticated user is using the project.

Search book:

The user can search book based on book id, book name, or by author name.

Issue Book:

To help the user get the required books issued.

Return Book:

To return the book before the last date without fine, or after the specified time duration with a late fine.

Admin Module:

It is to be operated by the admin with unique id and password. The admin is the person who decides authentication and authorization for all the different users of the application. It further can be subdivided as:

Register user

Issue Book

Maintain books in a stack, means record the availability at regular time interval.

Librarian:

Includes all the library staff who are required to enter the records in the system and keep an eye on the various activities like the issue of the book, the return of the book, non-availability of books etc. through the developed system.

SYSTEM DESIGN OF LIBRARY MANAGEMENT SYSTEM

Hardware requirement of library management system:

**Operating system:** Windows 7 (or above can be used) since it is stable

**Hard disks:** 40GB

**RAM:** 1GB as it will give faster performance throughput

.

Software requirement of library management system:

Java language

Net beans IDE 7.0.1 or eclipse neon

MySQL

PhpMyAdmin: As it can also easily handle server-side queries if any as well as compatible with tools for website designing.

ER DIAGRAM OF LIBRARY MANAGEMENT SYSTEM

The ER diagram provides the description of all the physical objects involved in the application like the library, books, user etc. It also briefs the operations to be done on these entities. This will minimize all the redundancy and attempt to store all the required information in as small a space as possible. The various entities are described as:

Library:

The main part of the organization for which this application has been designed. It has attributes like:

Name:

The name of the library to distinguish it from all the libraries available in any campus, uniquely.

Address:

This defines the address of the library as such the block number or lane number etc.

Books:

These are the basic building block of this system as well as any library. In other words the main purpose of any library and the cause to develop systems like this.

Book\_Name:

The name of the book which is almost unique in some way.

Book\_Code:

A number to use for sorting and arranging the book, as well as identifying it in the library.

Author:

The one who has written the book. As sometimes the book’s series become more popular by the author’s name rather than the book name.

Price:

The market value of the book is also required to maintain in the record, as sometimes it is needed to arrange and sort based on this, secondly, it is also required for compensation in case of loss or damage, as fine charges.

Quantity:

This is to indicate the availability of each book individually, so as to know whether last copy should be issued or kept as a reference piece. Also to maintain the number of books.

Rack\_No:

To get the exact location of the book, so as it becomes easy to search it and sort it at the time of binding up work.

Subject\_Code:

As there are various further division and subcategories of any subject. So, in that case, this is the unique id to distinguish the books, arrange them, and sort them. Like in computer science there are further many specialities like core java, advanced java, HTML, html5 etc.

User:

The next is the beneficiary, by whom the library is being accessed and who serves as a purpose for this system. Its attributes include:

Name:

The name of the student or teacher, who will get the book issued, or who will return the book.

Id:

The user’s unique college or university roll number i.e. the id. The same is applicable to teachers also, with their unique id.

Address:

This refers to the user’s physical area of residence. It is a composite attribute. As it further contains the house number and lane number.

Fine\_Amount:

To indicate the amount of fine he/she has to deposit and keep it up to date so that he/she is aware of the payment to be made at the end of the year or session.

Issue Status:

It makes to the notice of the librarian as well as to the student or teacher that ow many books they have already got issued and how much more can they get at the current point of time. It includes attributes as:

Book\_Name:

The name of the book which is almost unique in some way.

Book\_Code:

A number to use for sorting and arranging the book, as well as identifying it in the library.

Id:

The user’s unique college or university roll number i.e. the id. The same is applicable to teachers also, with their unique id. To know which user has been issued the book and for what time limit, that is what time the user is supposed to return the book, and if not will be charged fine.

Date\_Issue:

The date on which user got the book issued to read from it.

Return\_Date:

It indicates the date on which user is supposed to be returning the book, that is it is the date after the duration completed for which the user has been issued the book.

Return Status:

This tells the library management authority about the status of returned books per user. Whether a particular user has returned the book or not, on or before the last date. If not, in that case, the fine will be charged from him/her as a penalty for late submission.

Book\_Name:

The name of the book which is almost unique in some way.

Book\_Code:

A number to use for sorting and arranging the book, as well as identifying it in the library.

Id:

The user’s unique college or university roll number i.e. the id. The same is applicable to teachers also, with their unique id. To know which user has been issued the book and for what time limit, that is what time the user is supposed to return the book, and if not will be charged fine.

Date\_Issue:

The date on which user got the book issued to read from it.

Return\_Date:

It indicates the date on which user is supposed to be returning the book, that is it is the date after the duration completed for which the user has been issued the book.

USE CASE DIAGRAM OF LIBRARY MANAGEMENT SYSTEM

Use Case Diagram Description:

The above-mentioned use case diagram depicts the functionality of the library management system in a brief and satisfactory way.

In this diagram, the user has been shown more specifically as a student who will the first login to the system, to get access to the application. After getting the authentication and being authorized, the user will use the system with ease and security.

There is a database maintained for storing the records of books that are available in the stock, books that have been issued to some user, then the return date of each issued book.

After logging in the user will search for the books in the library using the subject code, book code, and therefore access to the rack number becomes easy, so ultimately using the rack number user finds the book, if it is available or not.

Once the book has been found, if wanted the book, the user can get it issued easily with their unique college or university id.

Now since the book has been issued it needs to be returned after a defined duration of time as a part of the system procedure.

At a regular interval of time user can check if there is any fine charged on him/her or not.

The librarian will also monitor all these activities at its end.

And along with this, there is some additional work for the librarian to be done like, adding new books in the rack, updating the book availability status time to time, checking the user’s identity is valid or not for a particular time period.

There may be some cases at times when for example say a student has been suspended due to some disciplinary action and therefore has been blocked from using any service provided by the university normally.

In this case, it becomes the duty of the librarian to check the identity is valid to issue books or not. Also similar is the case when a student has already passed out from the university.

SYSTEM REQUIREMENT OF LIBRARY MANAGEMENT SYSTEM

To show the comprehensive information for the intended purpose and about the system to be developed.

Non-Functional Requirement of library management system

These are those that specify some criteria that can be used to evaluate the performance of a system in some particular conditions.

Efficiency Requirement:

Through this system, the students or teachers and the librarian gets a way to ease their work. Through this system, the student can search and get the book issued easily.

Also, less time will be needed to spend by the librarian to handle this. Therefore the throughput is faster processing of library management system.

Reliability Requirement:

The system does its work with more accuracy like user registration to the system, user validation and authorization, book search and issue operation, return status, and updating the database by synchronizing between database and application.

Usability Requirement:

The proposed library management system provides a user-friendly environment to the users so that the librarians, as well as the students, can utilize the system in an effective manner for ease of work.

Delivery Requirement:

There is always some time duration specified to develop a project. Similarly, this system is expected to be complete within 6 months of time. This launch will be used for improving the performance, as it will be evaluated by the users and then the problems that are occurring with the system will be solved.

System Implementation Requirement:

To develop this system PHP, the server side scripting language has been used along with HTML 5 for designing the system layout. Also, PHP has been used since it is easy and effective for database connectivity. For the backend part which includes the database itself, MySQL has been used.

Functional Requirement of library management system

On the other side, there are those that deal with all type of technical functioning of the system.

Login :

he is to authenticate a user, that is to know whether he or she can get access to the system. At the time of login, the user will be required to enter their user id and password.

If for any user these fields don’t match, then the user will not be allowed to use the system. For this, the user id is stored at the time of registration.

This library managememt system must only allow a user with valid id and password to become the beneficiary.After this authorization takes place, to know what all are the levels a particular user can access to.

Also after finishing the work user must log out of the system to prevent the transaction from any intruder.

User Sign Up:

This is performed when the user is new to the system and wants to become a part, that wants to create a new account. For this system must verify all the user’s information. And also the information must be removed if found invalid or irrelevant.

Adding New book in the library:

This feature is used to add new books to the library by the authority. The system must enter and maintain the number of copies of each individual book.Also, the system must allocate unique id to individual books carefully.

Search operation:

The system must provide the facility of searching books based on their unique identity, the name of the book, author name. There must be some filters available to search with keywords. Some table view of the searches must be available.

Issue and Return book:

This is for issuing and returning books and also maintaining the issue and return status in the database timely. The system must be performing well with storing the issue data into the database. The update number of books feature must be working fine.

Before issuing any book, the system must firstly check for its availability, if it is available in stock or not. Along with issuing the books, the return dates must be shared with the students and must be entered into the database also.

Fine charge:

In case of failure in return on or before the last date, the student will be charged penalty fine, based on the per day rate multiplied by the number of days. The system must update this status time to time.

Also once the fine has been paid, the status must be updated again shortly.

ONLINE LIBRARY MANAGEMENT SYSTEM PROJECT

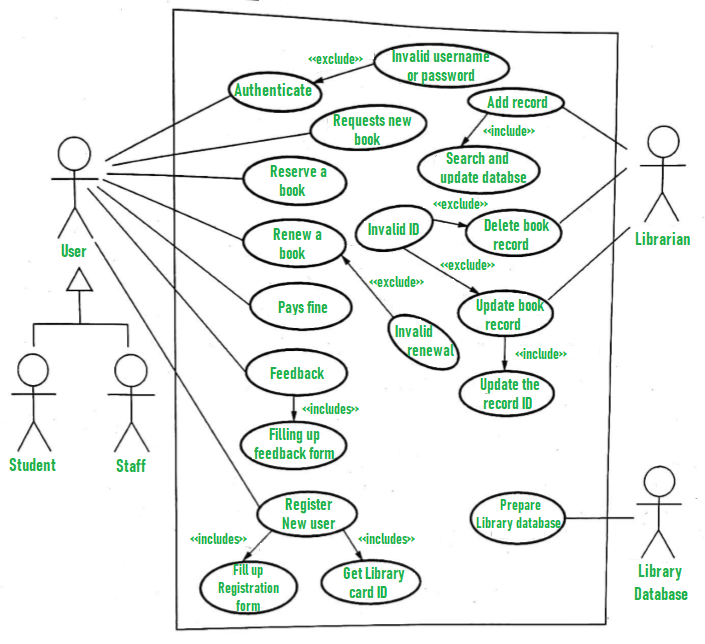
One add-on feature for this application is the option of providing online notice board, this feature can be used by the librarian for uploading announcements related to an event going on in the university or any book fair which is about to held in the nearby future.

Also, information like the sale of old books can be shared.

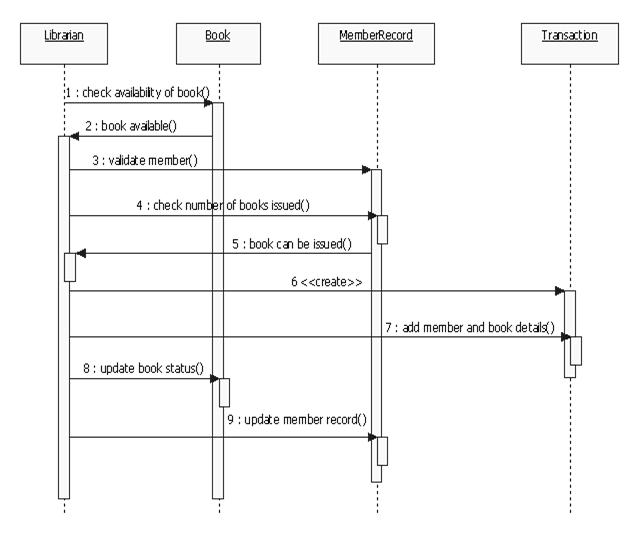
Another feature can be adding lecture notes, the teacher can create some lecture notes and upload it in pdf format in the application. So as the students will find the application more interesting and beneficial.

The lecture notes part will play a major role in gaining the popularization for this system.

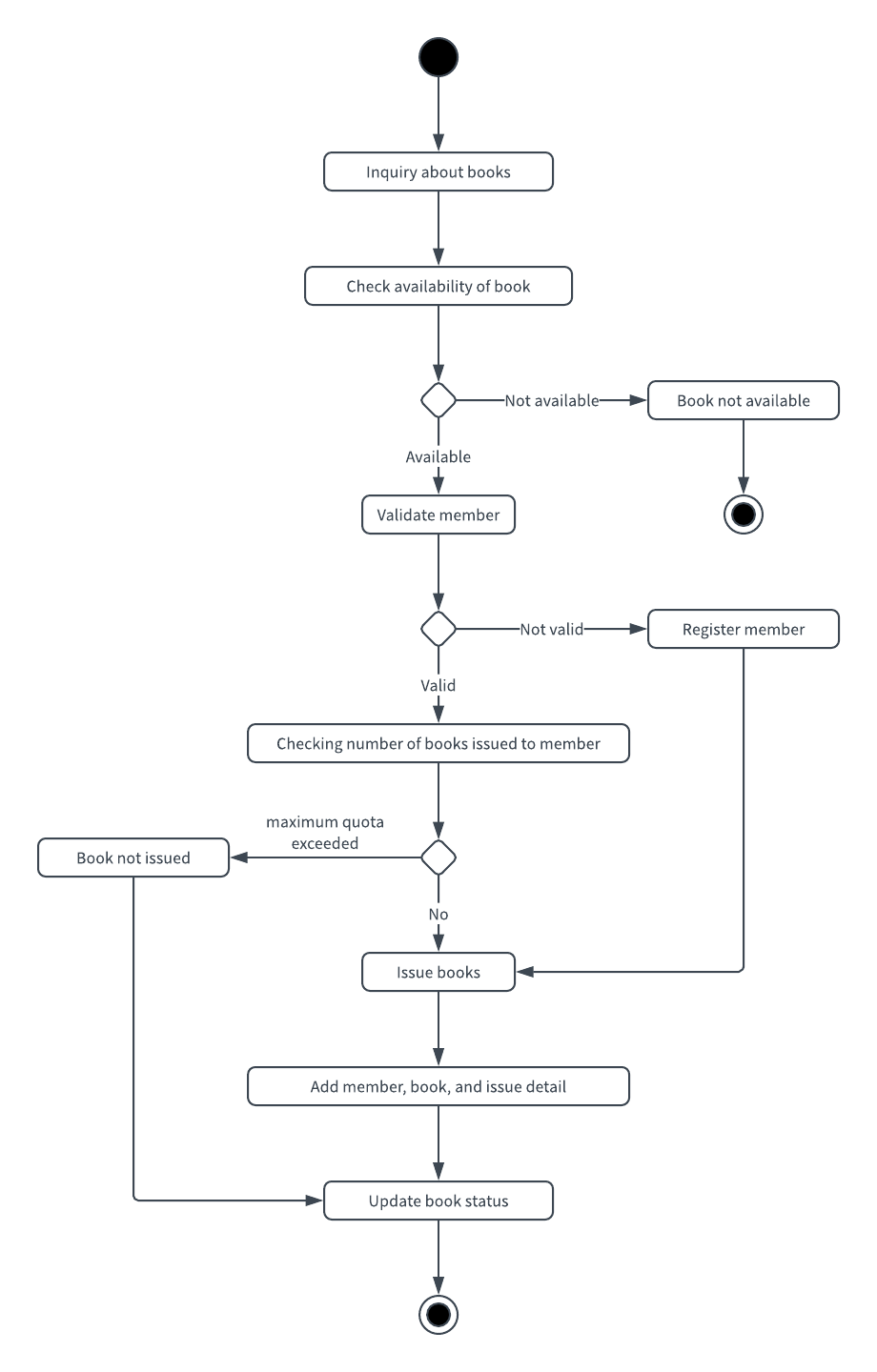
**Use case diagram**

****

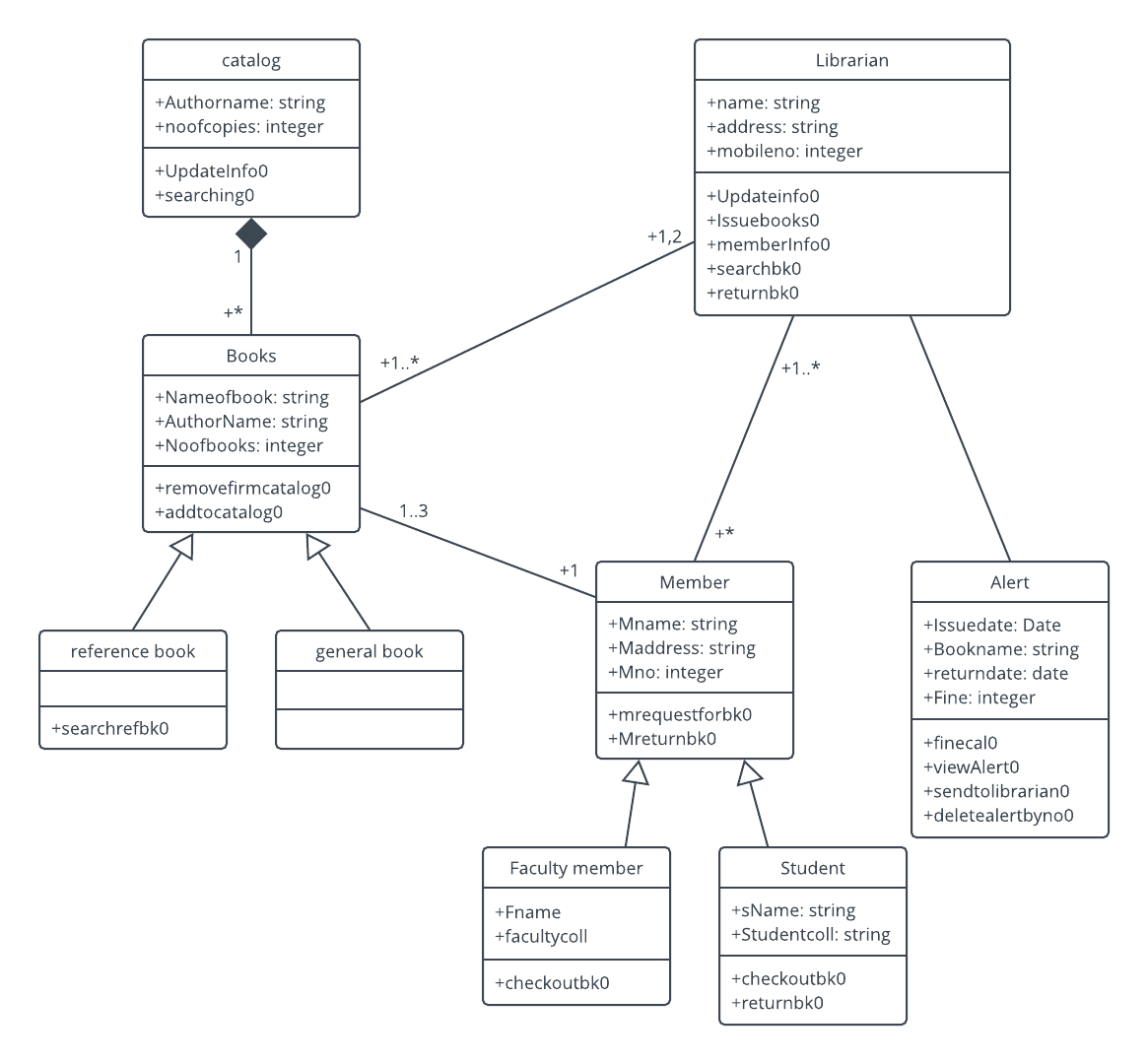
Sequence diagram



state chart diagram



Class diagram



Main class

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.Locale;

import java.util.concurrent.TimeUnit;

import javax.swing.\*;

import net.proteanit.sql.DbUtils;

public class main {

public static class ex{

public static int days=0;

}

public static void main(String[] args) {

login();

//create();

}

Login class

public static void login() {

JFrame f=new JFrame("Login");//creating instance of JFrame

JLabel l1,l2;

l1=new JLabel("Username"); //Create label Username

l1.setBounds(30,15, 100,30); //x axis, y axis, width, height

l2=new JLabel("Password"); //Create label Password

l2.setBounds(30,50, 100,30);

JTextField F\_user = new JTextField(); //Create text field for username

F\_user.setBounds(110, 15, 200, 30);

JPasswordField F\_pass=new JPasswordField(); //Create text field for password

F\_pass.setBounds(110, 50, 200, 30);

JButton login\_but=new JButton("Login");//creating instance of JButton for Login Button

login\_but.setBounds(130,90,80,25);//Dimensions for button

login\_but.addActionListener(new ActionListener() { //Perform action

public void actionPerformed(ActionEvent e){

String username = F\_user.getText(); //Store username entered by the user in the variable "username"

String password = F\_pass.getText(); //Store password entered by the user in the variable "password"

if(username.equals("")) //If username is null

{

JOptionPane.showMessageDialog(null,"Please enter username"); //Display dialog box with the message

}

else if(password.equals("")) //If password is null

{

JOptionPane.showMessageDialog(null,"Please enter password"); //Display dialog box with the message

}

else { //If both the fields are present then to login the user, check wether the user exists already

//System.out.println("Login connect");

Connection connection=connect(); //Connect to the database

try

{

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY"); //Use the database with the name "Library"

String st = ("SELECT \* FROM USERS WHERE USERNAME='"+username+"' AND PASSWORD='"+password+"'"); //Retreive username and passwords from users

ResultSet rs = stmt.executeQuery(st); //Execute query

if(rs.next()==false) { //Move pointer below

System.out.print("No user");

JOptionPane.showMessageDialog(null,"Wrong Username/Password!"); //Display Message

}

else {

f.dispose();

rs.beforeFirst(); //Move the pointer above

while(rs.next())

{

String admin = rs.getString("ADMIN"); //user is admin

//System.out.println(admin);

String UID = rs.getString("UID"); //Get user ID of the user

if(admin.equals("1")) { //If boolean value 1

admin\_menu(); //redirect to admin menu

}

else{

user\_menu(UID); //redirect to user menu for that user ID

}

}

}

}

catch (Exception ex) {

ex.printStackTrace();

}

}

}

});

f.add(F\_pass); //add password

f.add(login\_but);//adding button in JFrame

f.add(F\_user); //add user

f.add(l1); // add label1 i.e. for username

f.add(l2); // add label2 i.e. for password

f.setSize(400,180);//400 width and 500 height

f.setLayout(null);//using no layout managers

f.setVisible(true);//making the frame visible

f.setLocationRelativeTo(null);

}

Connection to sql

public static Connection connect()

{

try {

Class.forName("com.mysql.cj.jdbc.Driver");

//System.out.println("Loaded driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost/mysql?user=root&password=edureka");

//System.out.println("Connected to MySQL");

return con;

}

catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

Create database

public static void create() {

try {

Connection connection=connect();

ResultSet resultSet = connection.getMetaData().getCatalogs();

//iterate each catalog in the ResultSet

while (resultSet.next()) {

// Get the database name, which is at position 1

String databaseName = resultSet.getString(1);

if(databaseName.equals("library")) {

//System.out.print("yes");

Statement stmt = connection.createStatement();

//Drop database if it pre-exists to reset the complete database

String sql = "DROP DATABASE library";

stmt.executeUpdate(sql);

}

}

Statement stmt = connection.createStatement();

String sql = "CREATE DATABASE LIBRARY"; //Create Database

stmt.executeUpdate(sql);

stmt.executeUpdate("USE LIBRARY"); //Use Database

//Create Users Table

String sql1 = "CREATE TABLE USERS(UID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, USERNAME VARCHAR(30), PASSWORD VARCHAR(30), ADMIN BOOLEAN)";

stmt.executeUpdate(sql1);

//Insert into users table

stmt.executeUpdate("INSERT INTO USERS(USERNAME, PASSWORD, ADMIN) VALUES('admin','admin',TRUE)");

//Create Books table

stmt.executeUpdate("CREATE TABLE BOOKS(BID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, BNAME VARCHAR(50), GENRE VARCHAR(20), PRICE INT)");

//Create Issued Table

stmt.executeUpdate("CREATE TABLE ISSUED(IID INT NOT NULL AUTO\_INCREMENT PRIMARY KEY, UID INT, BID INT, ISSUED\_DATE VARCHAR(20), RETURN\_DATE VARCHAR(20), PERIOD INT, FINE INT)");

//Insert into books table

stmt.executeUpdate("INSERT INTO BOOKS(BNAME, GENRE, PRICE) VALUES ('War and Peace', 'Mystery', 200), ('The Guest Book', 'Fiction', 300), ('The Perfect Murder','Mystery', 150), ('Accidental Presidents', 'Biography', 250), ('The Wicked King','Fiction', 350)");

resultSet.close();

}

catch (Exception ex) {

ex.printStackTrace();

}

}

User class

public static void user\_menu(String UID) {

JFrame f=new JFrame("User Functions"); //Give dialog box name as User functions

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); //Exit user menu on closing the dialog box

JButton view\_but=new JButton("View Books");//creating instance of JButton

view\_but.setBounds(20,20,120,25);//x axis, y axis, width, height

view\_but.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

JFrame f = new JFrame("Books Available"); //View books stored in database

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Connection connection = connect();

String sql="select \* from BOOKS"; //Retreive data from database

try {

Statement stmt = connection.createStatement(); //connect to database

stmt.executeUpdate("USE LIBRARY"); // use librabry

stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

JTable book\_list= new JTable(); //show data in table format

book\_list.setModel(DbUtils.resultSetToTableModel(rs));

JScrollPane scrollPane = new JScrollPane(book\_list); //enable scroll bar

f.add(scrollPane); //add scroll bar

f.setSize(800, 400); //set dimensions of view books frame

f.setVisible(true);

f.setLocationRelativeTo(null);

} catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

}

);

JButton my\_book=new JButton("My Books");//creating instance of JButton

my\_book.setBounds(150,20,120,25);//x axis, y axis, width, height

my\_book.addActionListener(new ActionListener() { //Perform action

public void actionPerformed(ActionEvent e){

JFrame f = new JFrame("My Books"); //View books issued by user

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

int UID\_int = Integer.parseInt(UID); //Pass user ID

//.iid,issued.uid,issued.bid,issued.issued\_date,issued.return\_date,issued,

Connection connection = connect(); //connect to database

//retrieve data

String sql="select distinct issued.\*,books.bname,books.genre,books.price from issued,books " + "where ((issued.uid=" + UID\_int + ") and (books.bid in (select bid from issued where issued.uid="+UID\_int+"))) group by iid";

String sql1 = "select bid from issued where uid="+UID\_int;

try {

Statement stmt = connection.createStatement();

//use database

stmt.executeUpdate("USE LIBRARY");

stmt=connection.createStatement();

//store in array

ArrayList books\_list = new ArrayList();

ResultSet rs=stmt.executeQuery(sql);

JTable book\_list= new JTable(); //store data in table format

book\_list.setModel(DbUtils.resultSetToTableModel(rs));

//enable scroll bar

JScrollPane scrollPane = new JScrollPane(book\_list);

f.add(scrollPane); //add scroll bar

f.setSize(800, 400); //set dimensions of my books frame

f.setVisible(true);

f.setLocationRelativeTo(null);

} catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

}

);

f.add(my\_book); //add my books

f.add(view\_but); // add view books

f.setSize(300,100);//400 width and 500 height

f.setLayout(null);//using no layout managers

f.setVisible(true);//making the frame visible

f.setLocationRelativeTo(null);

}

Admin menu

public static void admin\_menu() {

JFrame f=new JFrame("Admin Functions"); //Give dialog box name as admin functions

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); //

JButton create\_but=new JButton("Create/Reset");//creating instance of JButton to create or reset database

create\_but.setBounds(450,60,120,25);//x axis, y axis, width, height

create\_but.addActionListener(new ActionListener() { //Perform action

public void actionPerformed(ActionEvent e){

create(); //Call create function

JOptionPane.showMessageDialog(null,"Database Created/Reset!"); //Open a dialog box and display the message

}

});

JButton view\_but=new JButton("View Books");//creating instance of JButton to view books

view\_but.setBounds(20,20,120,25);//x axis, y axis, width, height

view\_but.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

JFrame f = new JFrame("Books Available");

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Connection connection = connect(); //connect to database

String sql="select \* from BOOKS"; //select all books

try {

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY"); //use database

stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

JTable book\_list= new JTable(); //view data in table format

book\_list.setModel(DbUtils.resultSetToTableModel(rs));

//mention scroll bar

JScrollPane scrollPane = new JScrollPane(book\_list);

f.add(scrollPane); //add scrollpane

f.setSize(800, 400); //set size for frame

f.setVisible(true);

f.setLocationRelativeTo(null);

} catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

}

);

JButton users\_but=new JButton("View Users");//creating instance of JButton to view users

users\_but.setBounds(150,20,120,25);//x axis, y axis, width, height

users\_but.addActionListener(new ActionListener() { //Perform action on click button

public void actionPerformed(ActionEvent e){

JFrame f = new JFrame("Users List");

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Connection connection = connect();

String sql="select \* from users"; //retrieve all users

try {

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY"); //use database

stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

JTable book\_list= new JTable();

book\_list.setModel(DbUtils.resultSetToTableModel(rs));

//mention scroll bar

JScrollPane scrollPane = new JScrollPane(book\_list);

f.add(scrollPane); //add scrollpane

f.setSize(800, 400); //set size for frame

f.setVisible(true);

f.setLocationRelativeTo(null);

} catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

}

);

JButton issued\_but=new JButton("View Issued Books");//creating instance of JButton to view the issued books

issued\_but.setBounds(280,20,160,25);//x axis, y axis, width, height

issued\_but.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

JFrame f = new JFrame("Users List");

//f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Connection connection = connect();

String sql="select \* from issued";

try {

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY");

stmt=connection.createStatement();

ResultSet rs=stmt.executeQuery(sql);

JTable book\_list= new JTable();

book\_list.setModel(DbUtils.resultSetToTableModel(rs));

JScrollPane scrollPane = new JScrollPane(book\_list);

f.add(scrollPane);

f.setSize(800, 400);

f.setVisible(true);

f.setLocationRelativeTo(null);

} catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

}

);

JButton add\_user=new JButton("Add User"); //creating instance of JButton to add users

add\_user.setBounds(20,60,120,25); //set dimensions for button

add\_user.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

JFrame g = new JFrame("Enter User Details"); //Frame to enter user details

//g.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//Create label

JLabel l1,l2;

l1=new JLabel("Username"); //label 1 for username

l1.setBounds(30,15, 100,30);

l2=new JLabel("Password"); //label 2 for password

l2.setBounds(30,50, 100,30);

//set text field for username

JTextField F\_user = new JTextField();

F\_user.setBounds(110, 15, 200, 30);

//set text field for password

JPasswordField F\_pass=new JPasswordField();

F\_pass.setBounds(110, 50, 200, 30);

//set radio button for admin

JRadioButton a1 = new JRadioButton("Admin");

a1.setBounds(55, 80, 200,30);

//set radio button for user

JRadioButton a2 = new JRadioButton("User");

a2.setBounds(130, 80, 200,30);

//add radio buttons

ButtonGroup bg=new ButtonGroup();

bg.add(a1);bg.add(a2);

JButton create\_but=new JButton("Create");//creating instance of JButton for Create

create\_but.setBounds(130,130,80,25);//x axis, y axis, width, height

create\_but.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

String username = F\_user.getText();

String password = F\_pass.getText();

Boolean admin = false;

if(a1.isSelected()) {

admin=true;

}

Connection connection = connect();

try {

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY");

stmt.executeUpdate("INSERT INTO USERS(USERNAME,PASSWORD,ADMIN) VALUES ('"+username+"','"+password+"',"+admin+")");

JOptionPane.showMessageDialog(null,"User added!");

g.dispose();

}

catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

});

g.add(create\_but);

g.add(a2);

g.add(a1);

g.add(l1);

g.add(l2);

g.add(F\_user);

g.add(F\_pass);

g.setSize(350,200);//400 width and 500 height

g.setLayout(null);//using no layout managers

g.setVisible(true);//making the frame visible

g.setLocationRelativeTo(null);

}

});

JButton add\_book=new JButton("Add Book"); //creating instance of JButton for adding books

add\_book.setBounds(150,60,120,25);

add\_book.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

//set frame wot enter book details

JFrame g = new JFrame("Enter Book Details");

//g.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// set labels

JLabel l1,l2,l3;

l1=new JLabel("Book Name"); //lebel 1 for book name

l1.setBounds(30,15, 100,30);

l2=new JLabel("Genre"); //label 2 for genre

l2.setBounds(30,53, 100,30);

l3=new JLabel("Price"); //label 2 for price

l3.setBounds(30,90, 100,30);

//set text field for book name

JTextField F\_bname = new JTextField();

F\_bname.setBounds(110, 15, 200, 30);

//set text field for genre

JTextField F\_genre=new JTextField();

F\_genre.setBounds(110, 53, 200, 30);

//set text field for price

JTextField F\_price=new JTextField();

F\_price.setBounds(110, 90, 200, 30);

JButton create\_but=new JButton("Submit");//creating instance of JButton to submit details

create\_but.setBounds(130,130,80,25);//x axis, y axis, width, height

create\_but.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

// assign the book name, genre, price

String bname = F\_bname.getText();

String genre = F\_genre.getText();

String price = F\_price.getText();

//convert price of integer to int

int price\_int = Integer.parseInt(price);

Connection connection = connect();

try {

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY");

stmt.executeUpdate("INSERT INTO BOOKS(BNAME,GENRE,PRICE) VALUES ('"+bname+"','"+genre+"',"+price\_int+")");

JOptionPane.showMessageDialog(null,"Book added!");

g.dispose();

}

catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

});

g.add(l3);

g.add(create\_but);

g.add(l1);

g.add(l2);

g.add(F\_bname);

g.add(F\_genre);

g.add(F\_price);

g.setSize(350,200);//400 width and 500 height

g.setLayout(null);//using no layout managers

g.setVisible(true);//making the frame visible

g.setLocationRelativeTo(null);

}

});

JButton issue\_book=new JButton("Issue Book"); //creating instance of JButton to issue books

issue\_book.setBounds(450,20,120,25);

issue\_book.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

//enter details

JFrame g = new JFrame("Enter Details");

//g.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//create labels

JLabel l1,l2,l3,l4;

l1=new JLabel("Book ID(BID)"); // Label 1 for Book ID

l1.setBounds(30,15, 100,30);

l2=new JLabel("User ID(UID)"); //Label 2 for user ID

l2.setBounds(30,53, 100,30);

l3=new JLabel("Period(days)"); //Label 3 for period

l3.setBounds(30,90, 100,30);

l4=new JLabel("Issued Date(DD-MM-YYYY)"); //Label 4 for issue date

l4.setBounds(30,127, 150,30);

JTextField F\_bid = new JTextField();

F\_bid.setBounds(110, 15, 200, 30);

JTextField F\_uid=new JTextField();

F\_uid.setBounds(110, 53, 200, 30);

JTextField F\_period=new JTextField();

F\_period.setBounds(110, 90, 200, 30);

JTextField F\_issue=new JTextField();

F\_issue.setBounds(180, 130, 130, 30);

JButton create\_but=new JButton("Submit");//creating instance of JButton

create\_but.setBounds(130,170,80,25);//x axis, y axis, width, height

create\_but.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

String uid = F\_uid.getText();

String bid = F\_bid.getText();

String period = F\_period.getText();

String issued\_date = F\_issue.getText();

int period\_int = Integer.parseInt(period);

Connection connection = connect();

try {

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY");

stmt.executeUpdate("INSERT INTO ISSUED(UID,BID,ISSUED\_DATE,PERIOD) VALUES ('"+uid+"','"+bid+"','"+issued\_date+"',"+period\_int+")");

JOptionPane.showMessageDialog(null,"Book Issued!");

g.dispose();

}

catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

});

g.add(l3);

g.add(l4);

g.add(create\_but);

g.add(l1);

g.add(l2);

g.add(F\_uid);

g.add(F\_bid);

g.add(F\_period);

g.add(F\_issue);

g.setSize(350,250);//400 width and 500 height

g.setLayout(null);//using no layout managers

g.setVisible(true);//making the frame visible

g.setLocationRelativeTo(null);

}

});

JButton return\_book=new JButton("Return Book"); //creating instance of JButton to return books

return\_book.setBounds(280,60,160,25);

return\_book.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

JFrame g = new JFrame("Enter Details");

//g.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

//set labels

JLabel l1,l2,l3,l4;

l1=new JLabel("Issue ID(IID)"); //Label 1 for Issue ID

l1.setBounds(30,15, 100,30);

l4=new JLabel("Return Date(DD-MM-YYYY)");

l4.setBounds(30,50, 150,30);

JTextField F\_iid = new JTextField();

F\_iid.setBounds(110, 15, 200, 30);

JTextField F\_return=new JTextField();

F\_return.setBounds(180, 50, 130, 30);

JButton create\_but=new JButton("Return");//creating instance of JButton to mention return date and calculcate fine

create\_but.setBounds(130,170,80,25);//x axis, y axis, width, height

create\_but.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e){

String iid = F\_iid.getText();

String return\_date = F\_return.getText();

Connection connection = connect();

try {

Statement stmt = connection.createStatement();

stmt.executeUpdate("USE LIBRARY");

//Intialize date1 with NULL value

String date1=null;

String date2=return\_date; //Intialize date2 with return date

//select issue date

ResultSet rs = stmt.executeQuery("SELECT ISSUED\_DATE FROM ISSUED WHERE IID="+iid);

while (rs.next()) {

date1 = rs.getString(1);

}

try {

Date date\_1=new SimpleDateFormat("dd-MM-yyyy").parse(date1);

Date date\_2=new SimpleDateFormat("dd-MM-yyyy").parse(date2);

//subtract the dates and store in diff

long diff = date\_2.getTime() - date\_1.getTime();

//Convert diff from milliseconds to days

ex.days=(int)(TimeUnit.DAYS.convert(diff, TimeUnit.MILLISECONDS));

} catch (ParseException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

//update return date

stmt.executeUpdate("UPDATE ISSUED SET RETURN\_DATE='"+return\_date+"' WHERE IID="+iid);

g.dispose();

Connection connection1 = connect();

Statement stmt1 = connection1.createStatement();

stmt1.executeUpdate("USE LIBRARY");

ResultSet rs1 = stmt1.executeQuery("SELECT PERIOD FROM ISSUED WHERE IID="+iid); //set period

String diff=null;

while (rs1.next()) {

diff = rs1.getString(1);

}

int diff\_int = Integer.parseInt(diff);

if(ex.days&amp;amp;amp;amp;amp;amp;amp;amp;amp;amp;amp;gt;diff\_int) { //If number of days are more than the period then calculcate fine

//System.out.println(ex.days);

int fine = (ex.days-diff\_int)\*10; //fine for every day after the period is Rs 10.

//update fine in the system

stmt1.executeUpdate("UPDATE ISSUED SET FINE="+fine+" WHERE IID="+iid);

String fine\_str = ("Fine: Rs. "+fine);

JOptionPane.showMessageDialog(null,fine\_str);

}

JOptionPane.showMessageDialog(null,"Book Returned!");

}

catch (SQLException e1) {

// TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, e1);

}

}

});

g.add(l4);

g.add(create\_but);

g.add(l1);

g.add(F\_iid);

g.add(F\_return);

g.setSize(350,250);//400 width and 500 height

g.setLayout(null);//using no layout managers

g.setVisible(true);//making the frame visible

g.setLocationRelativeTo(null);

}

});

f.add(create\_but);

f.add(return\_book);

f.add(issue\_book);

f.add(add\_book);

f.add(issued\_but);

f.add(users\_but);

f.add(view\_but);

f.add(add\_user);

f.setSize(600,200);//400 width and 500 height

f.setLayout(null);//using no layout managers

f.setVisible(true);//making the frame visible

f.setLocationRelativeTo(null);

}

}