A

Project Report

On

LIBRARY MANAGEMENT SYSTEM

Submitted by

Nilesh Mishra

Submitted to

Upskill Campus

In partial fulfillment for the award of the degree of

MCA (Master of Computer Application) Second Semester

In

Awadhesh Pratap Singh University Rewa Madhya Pradesh

2023-2024

Awadhesh Pratap Singh University Rewa Madhya Pradesh

Project Certificate

This is to certify that Mr. Ajeet Mishra & Rekha Tripathi has completed their project on the topic of "LIBRARY MANAGEMENT SYSTEM " Prescribed" by APSU Rewa for MCA (Master of Computer Application) Second Semester.

Date: Name & Signature of the student

Place:

Verified by Supervisor

Name & Signature of the supervisor

Acknowledgement

This Major Project is the result of contribution of many mind. I would like to acknowledge and thank my project guide Mr. Ajeet Mishra for his valuable support and guidance. He guided me through the process from conception and till the completion of this project. I would also like to thanks my Institute Director Mr. P.K. Roy and my all faculties. I thank to lab staff members Ms. Rekha Tripathi and Ms. Shriji Tiwari and other non-teaching members.

I am very thankful for the open handed support extended by many people. While no list would be complete, it is my pleasure to acknowledge the assistance of my friends who provided encouragement, knowledge and constructive suggestions.

Signature of Student

Date:

(Name of Student)

Nilesh Mishra

CONTENTS

INTRODUCTION:

Library Management System consists of list of records about the management of the details of the students and the issues going on and also about some books and all. This is a web-based application. The project has three modules namely- User, Registration, Librarian. According to the Modules the Distributor and Sub Distributors can manage and do their activities in easy manner.

As the modern organizations are automated and computers are working as per the instructions, it becomes essential for the coordination of human beings, commodity and computers in a modern organization. This information helps the distributors to purchase or sale the products very efficiently.

The administrators and all the others can communicate with the system through this project, thus facilitating effective implementation and monitoring of various activities of the distributor of a supermarket.

SYSTEM ANALYSIS:

1. Existing System

Various problems of physical system are described below:-

- If one is not very careful then there is a possibility of issuing more than one book to a user.
- There is a possibility of issuing a book to a user, whose membership is not there.
- When a user requests for the a book, one has to physically check for the presence of a book in the library
- Answering management query is a time consuming process,
- Daily keeping a manual record of changes taking place in the library such as book being issued, book being returned etc. Can become cumbersome if the Library size is bigger.

2.Proposed System

The Library Management System is a software application which avoids more manual hours in taking the book, that need to spend in record keeping and generating reports. Maintaining of user details is complex in manual system in terms of agreements, royalty and activities. This all have to be maintained in ledgers or books. Co-coordinators needs to verify each record for small information also.

- Easy search of book in the online library.
- Avoid the manual work.
- User need not go to the library for Issue any kind of book, he can renewal the book online.

System Specification

Hardware Requirements:-

- Pentium-IV(Processor).
- 256 MB Ram
- 512 KB Cache Memory
- Hard disk 10 GB
- Microsoft Compatible 101 or more Key Board

Software Requirements: -

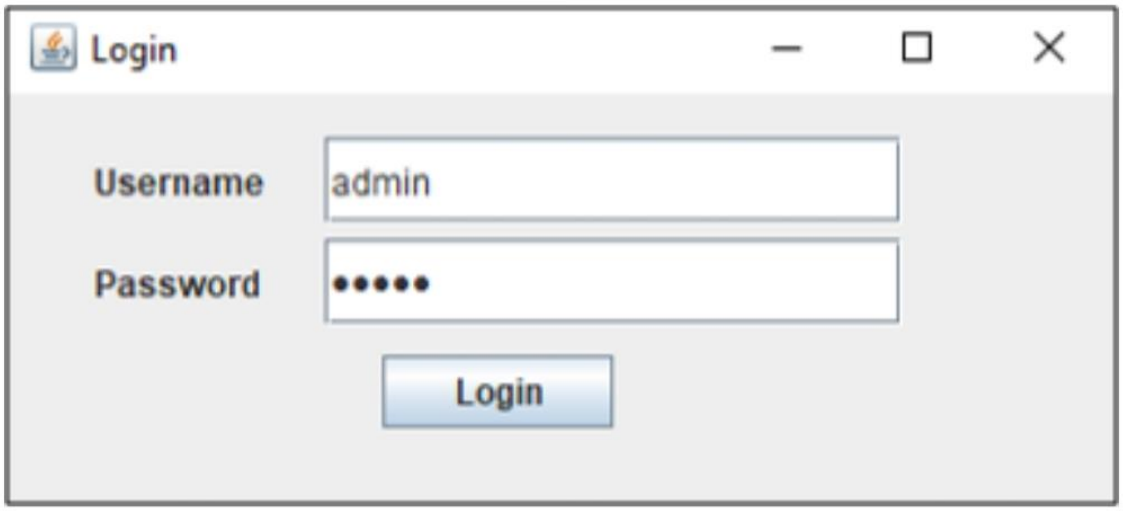
Operating System : Windows 95/98/XP with MS-office

- Programming language: Java swing
- Front-end: Java(JSP)
- Back-End : Java (JDBC,SERVLET)
- Database-MYSQL

System Design

- **Login**

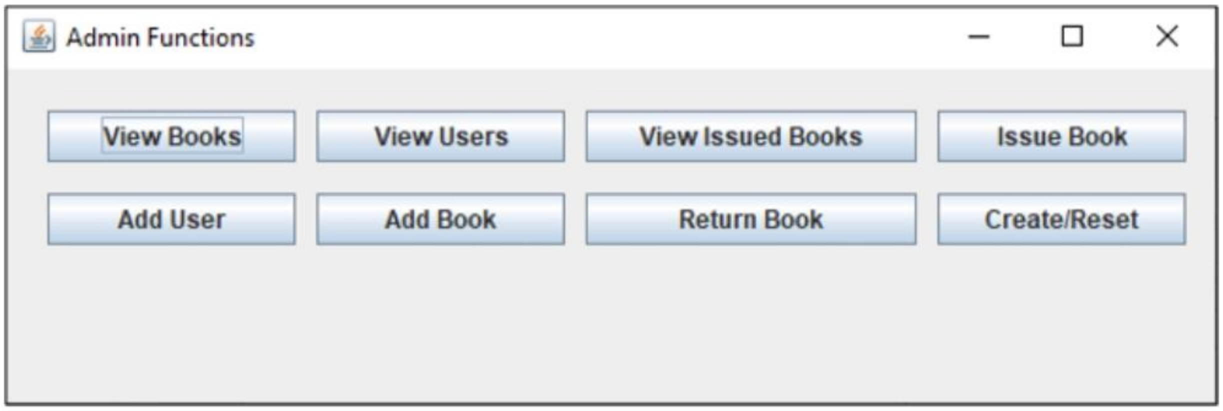
I have created this function to enable the user and the admin login. So, initially when a user logs in for the first time, that user will be an admin by default, and the username and password will be {admin, admin}. Refer below.



Username: admin

Password: •••••

For this schema, I have considered only one admin. So, once a user logs in as an admin, he or she will be redirected to the admin menu as below. I will discuss the functions of the admin in the admin menu section.



- **Create**

The create function is used to create the database, tables and add data into these tables. So, to do that, SQL statements will be used as below.

Now, that we have created the database, connected with GUI and enables the login function, next in this article on Library Management System Project in Java, let us now discuss the functions of the User Menu.

- **User Menu**

The User Menu is designed to show details of all the books present in the library and the books issued by the user.

Next, in this article on Library Management System Project in Java, let us discuss the code for Admin Menu function.

- **Admin Menu**

The Admin Menu is designed to show details of users, books, issued books, add books, return books, add user, and create or reset the database.

Now that you have understood all the functions, let us execute our library management system project in Java and see the outputs.

Coding

- Login

```
            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);
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
T{
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);
}
```

- **Create/ Reset**

```
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 Menu

```
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);

E);

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 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();
}
```

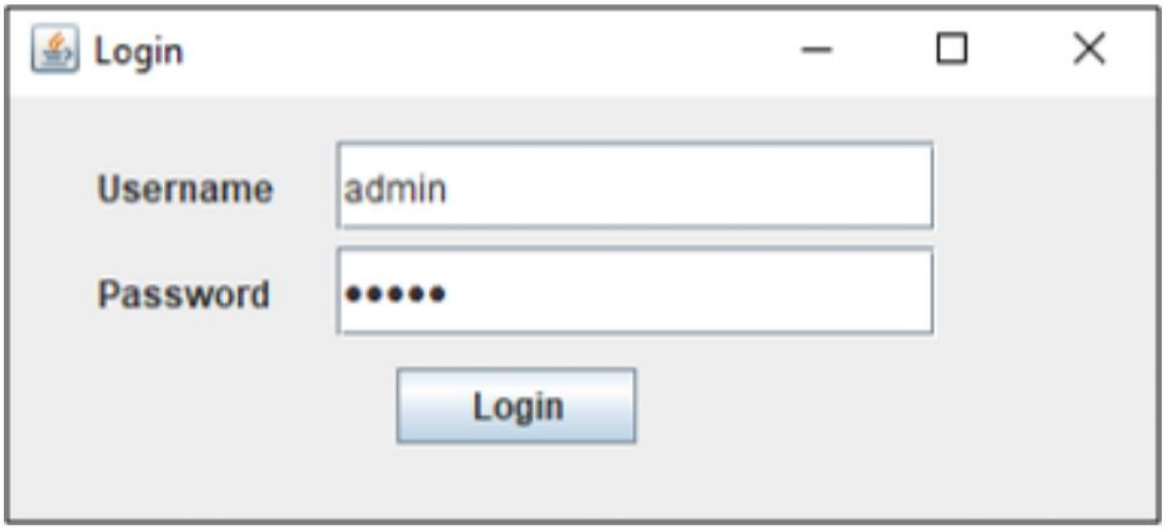
```
//update return date

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;a
mp;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
```

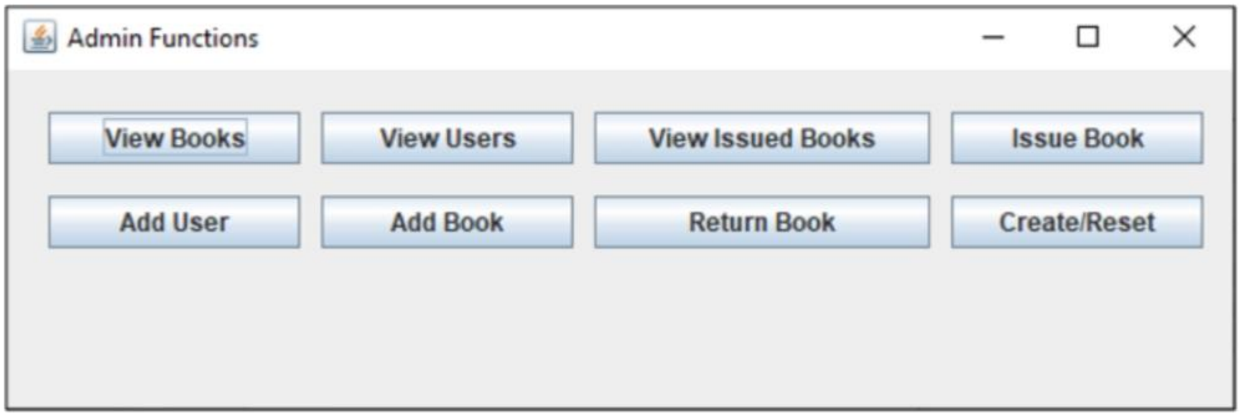
```
f.setVisible(true);//making the frame visible
f.setLocationRelativeTo(null);

}
}
```

<u>Output</u>:

Execute the application by clicking on the run button. Once, you execute you will see the below dialog box. In the below dialog box, mention username and password as {admin, admin}. Then click on the Login button.



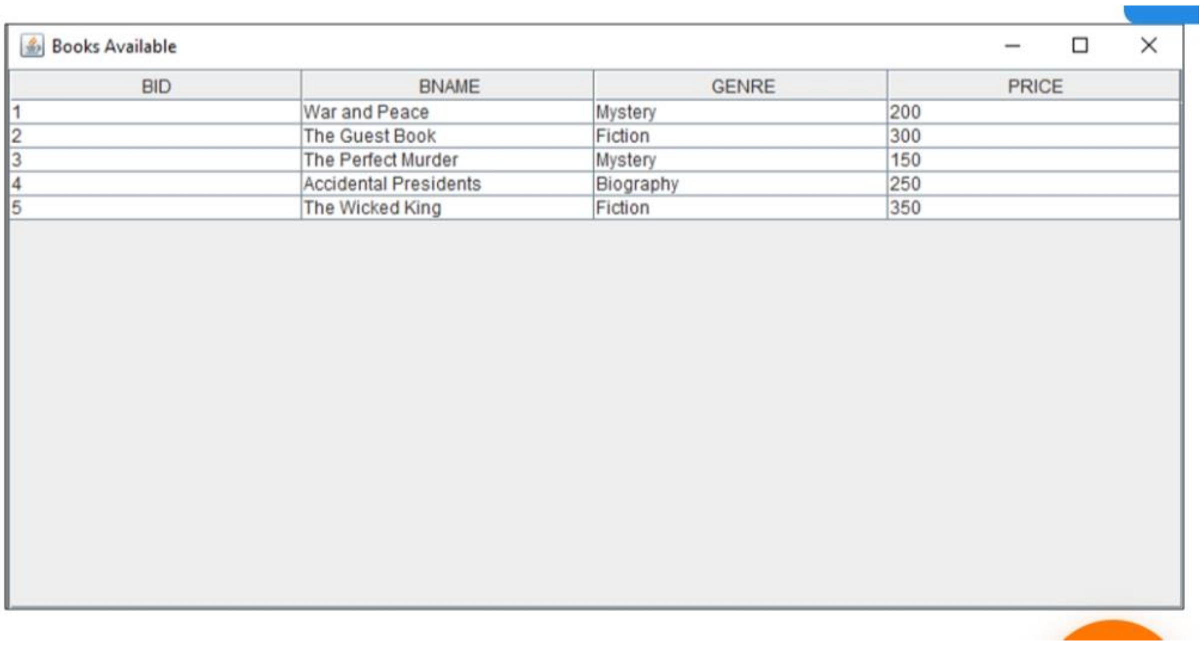
Once you click on the Login button, you will see the below dialog box opening up



Here you have various options which you can explore. So, let us start with the first one:

View Books

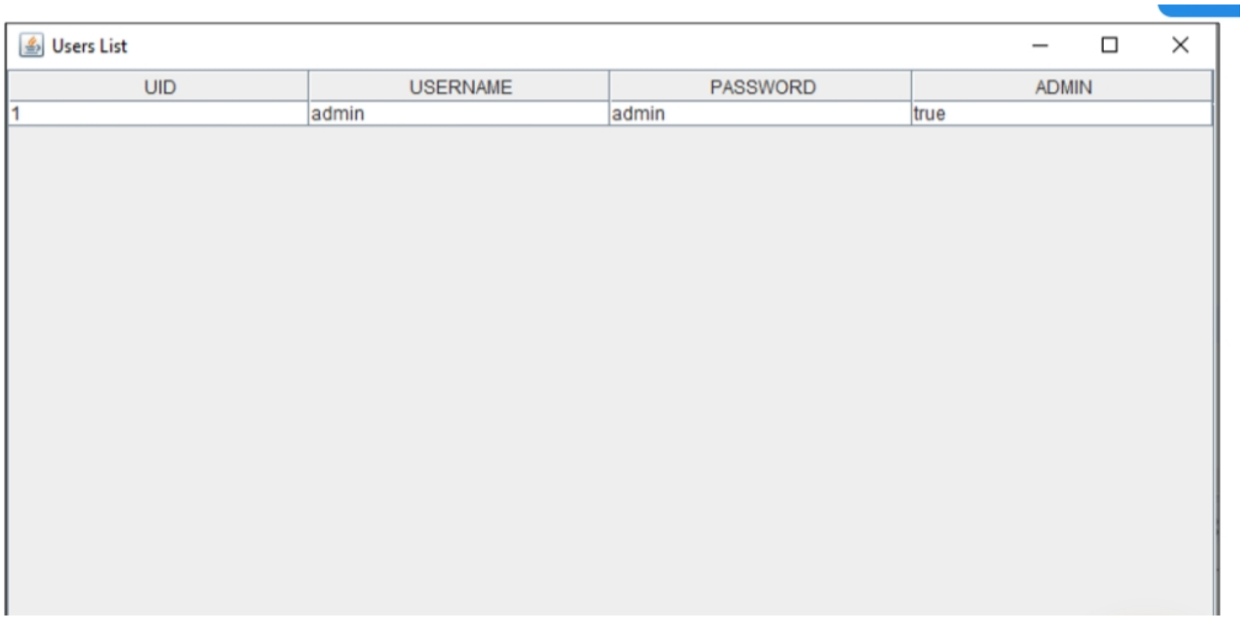
Once, you click on View Books button, you will see the below frame displaying all the books present in the database, with their details.



BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350

View Users

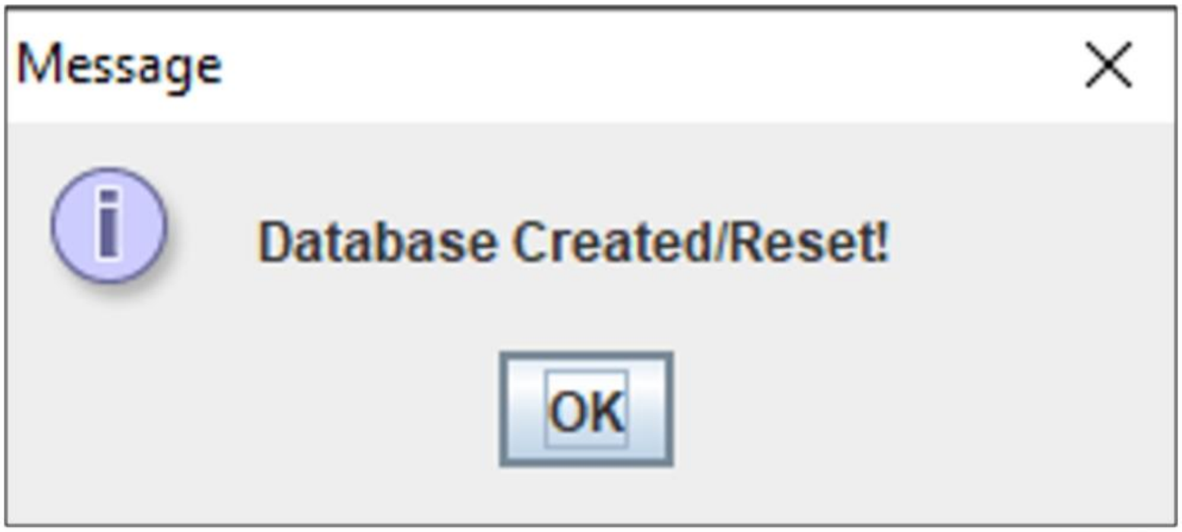
The View Users button is used to view the current users on the system. Since we just have only one user present i.e the admin, it will show you output as below:


Users List

UID	USERNAME	PASSWORD	ADMIN
1	admin	admin	true

Create/Reset

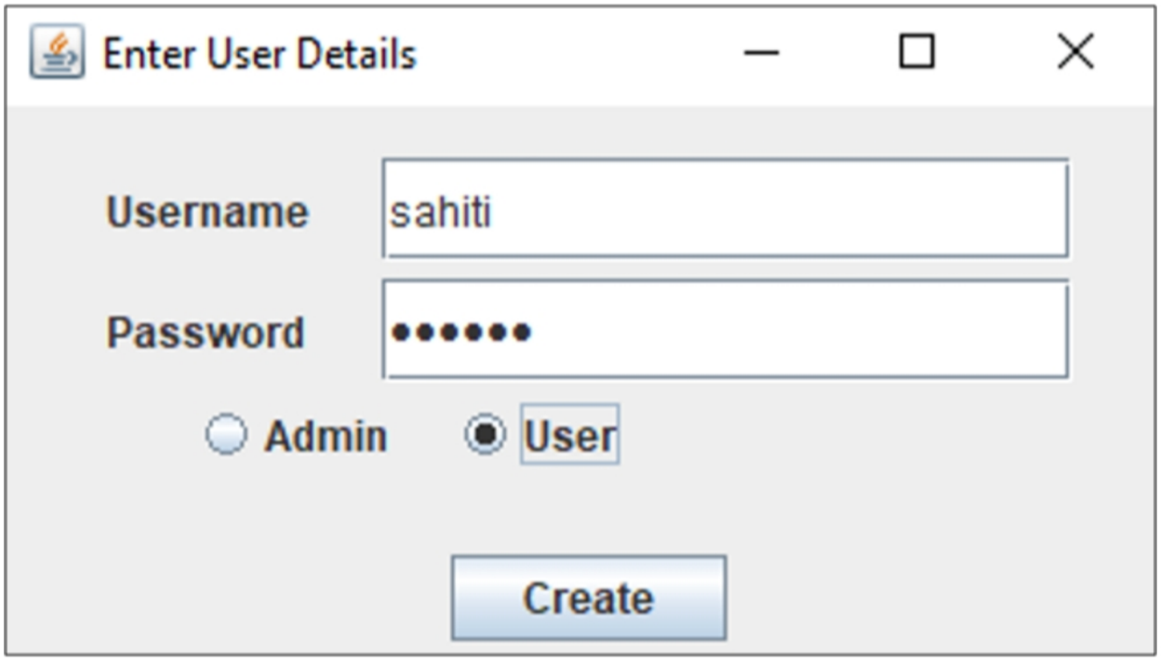
This functionality is used to create or reset a database. So, once you click on the button Create/Rest, you will see the below output:


Message

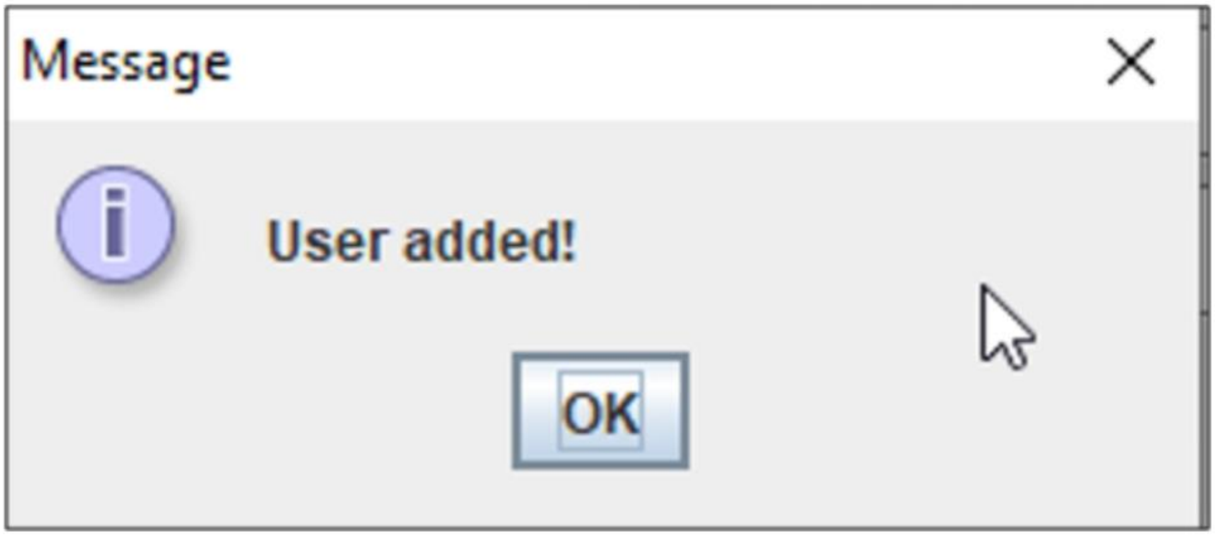
Database Created/Reset!

Add User

To add a user, click on the option “Add User” and mention details such as username, password and choose the radio button user or admin. By default, it will be the user. Then, click on Create.

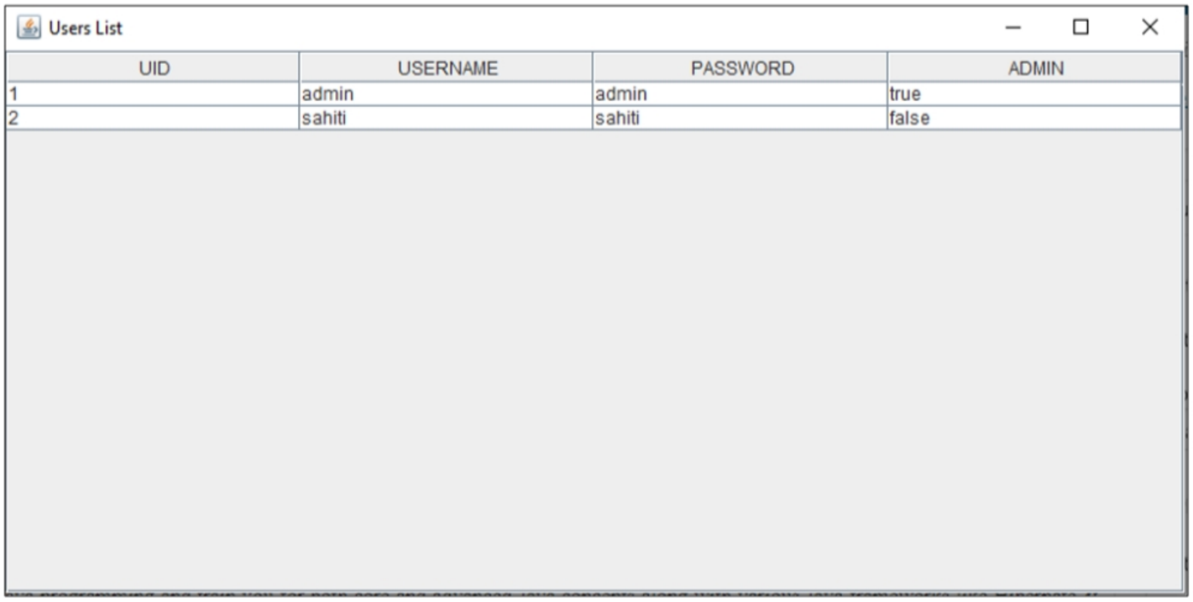


Once the user is created, you will see an output as below:



User added!

Now, again if you click on View Users button, you will see the below output:

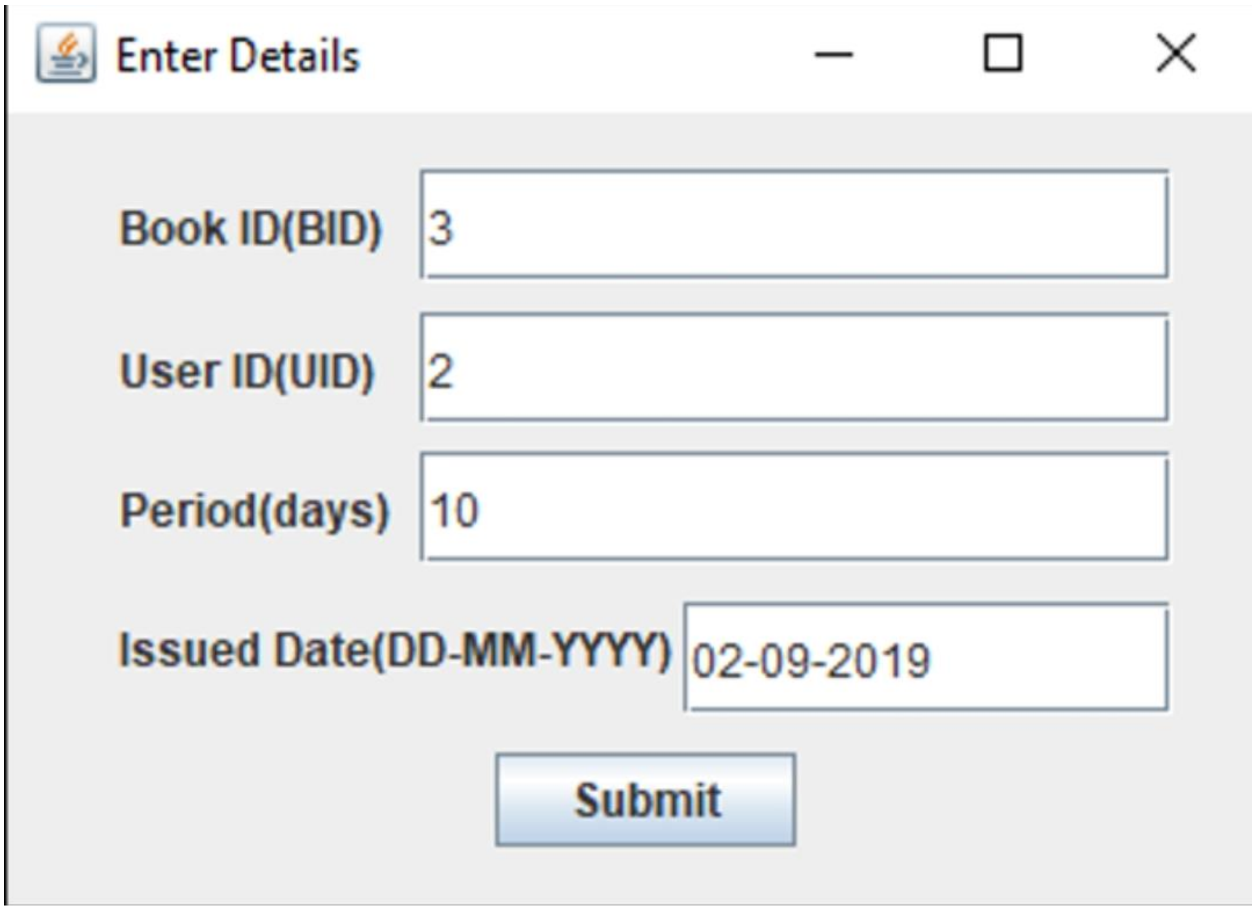


UID	USERNAME	PASSWORD	ADMIN
1	admin	admin	true
2	sahiti	sahiti	false

Alright, so now that we have added a user. Let us say, that particular user wants to issue books. To do that, the user has to choose the option of Issue Book.

Issue Book

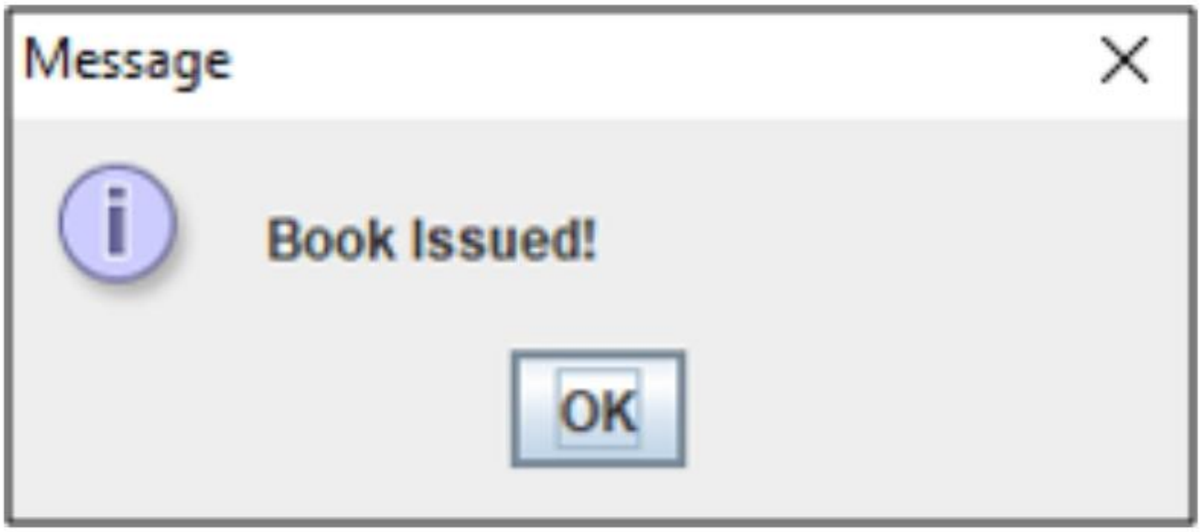
Suppose, if you are the user, once you click on the Issue Book button, you have to mention the Book ID, User ID, Period(Number of days for issuing the book), and the Issue Date as follows:



Enter Details

Book ID(BID)	3
User ID(UID)	2
Period(days)	10
Issued Date(DD-MM-YYYY)	02-09-2019

Then click on Submit. Once, you click on Submit, you will see the below dialog box:

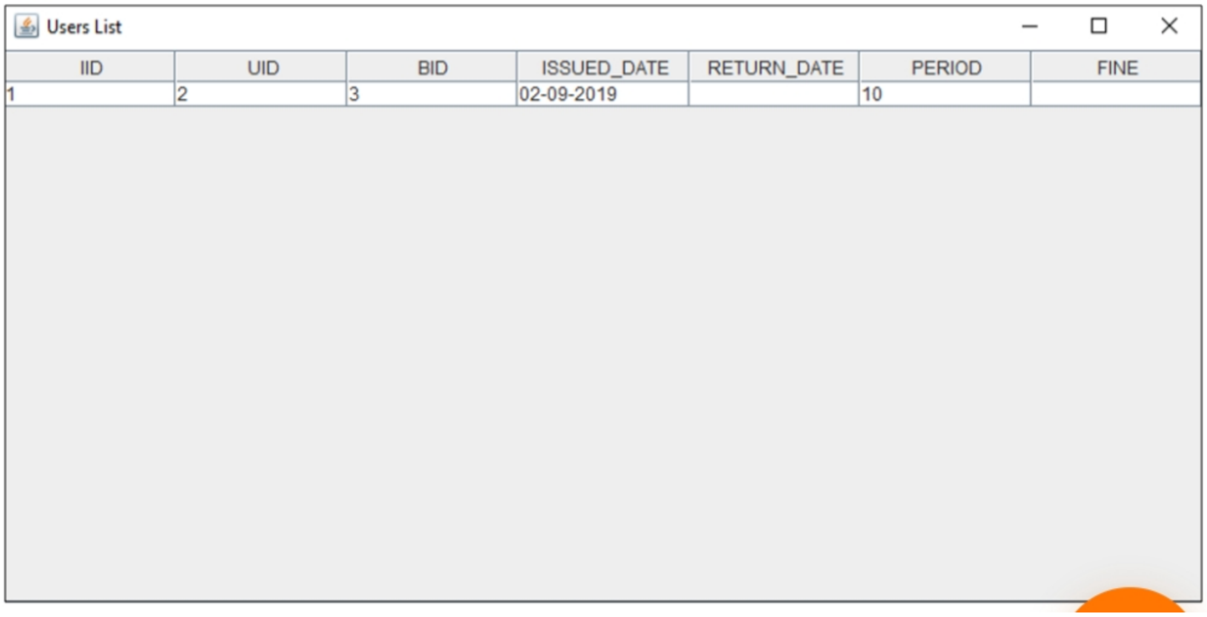

Message

Book Issued!

Now, if you want to see the issued books details, you can use the View Issued Books functionality.

View Issued Books

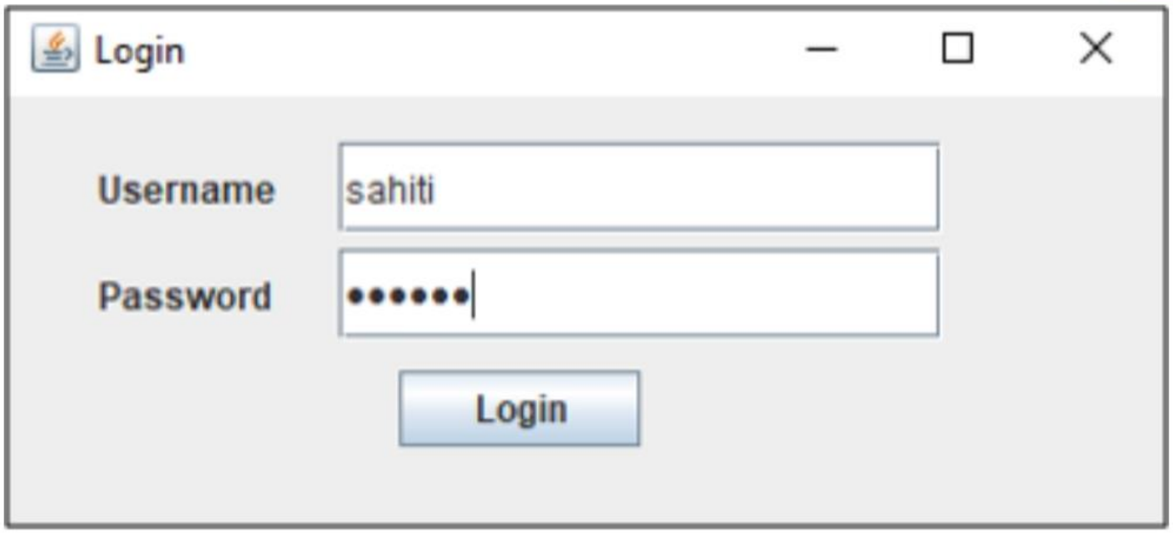
Once you click on this button, you will see the following output:



Users List

IID	UID	BID	ISSUED_DATE	RETURN_DATE	PERIOD	FINE
1	2	3	02-09-2019		10	

Alright, so, now if the user logs in to the system, using the login function, as below:

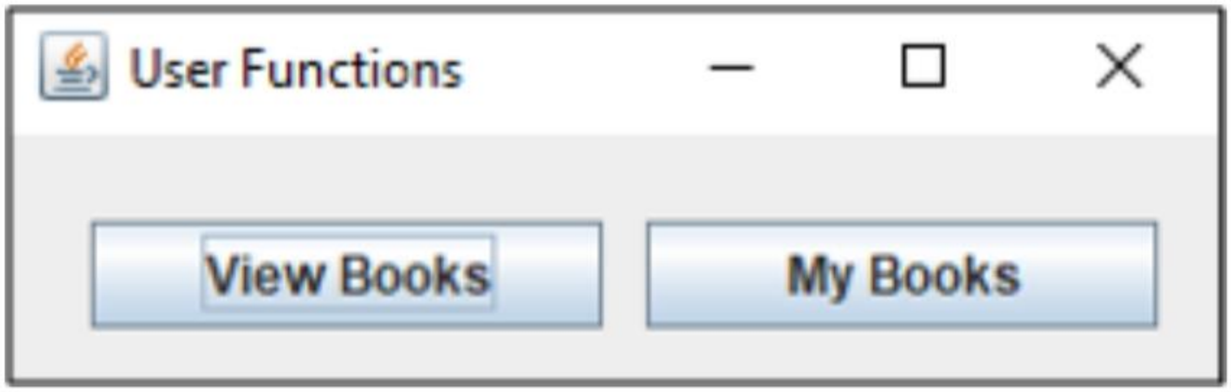


Login

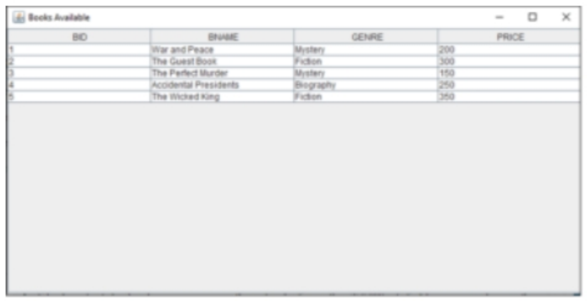
Username: sahiti

Password: ••••••

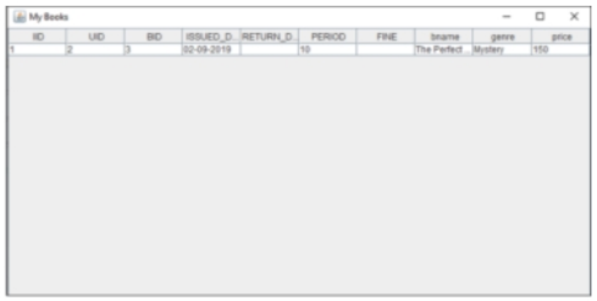
Then the user will see the below User Menu.



Here, the user can view all the books in the database by using the View Books option and the books issued by the user in the My Books section as below:



BID	BNAME	GENRE	PRICE
1	War and Peace	Mystery	200
2	The Guest Book	Fiction	300
3	The Perfect Murder	Mystery	150
4	Accidental Presidents	Biography	250
5	The Wicked King	Fiction	350

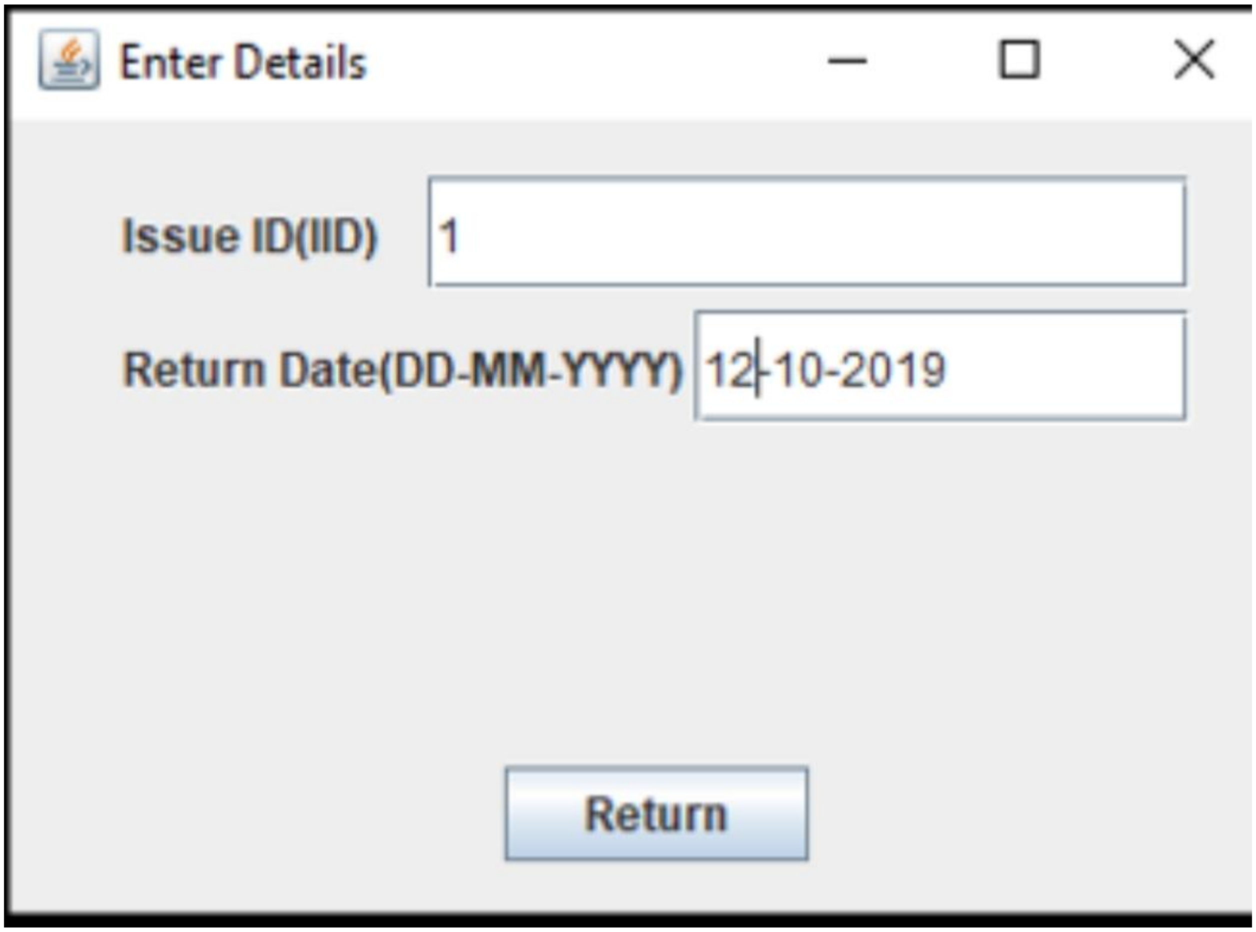


IID	UID	BID	ISSUED_D...	RETURN_D...	PERIOD	FINE	bname	genre	price
1	2	3	02-09-2019		10		The Perfect ...	Mystery	150

Now, if you wish to return the book, then you have to choose the option of Return Book.

Return Book

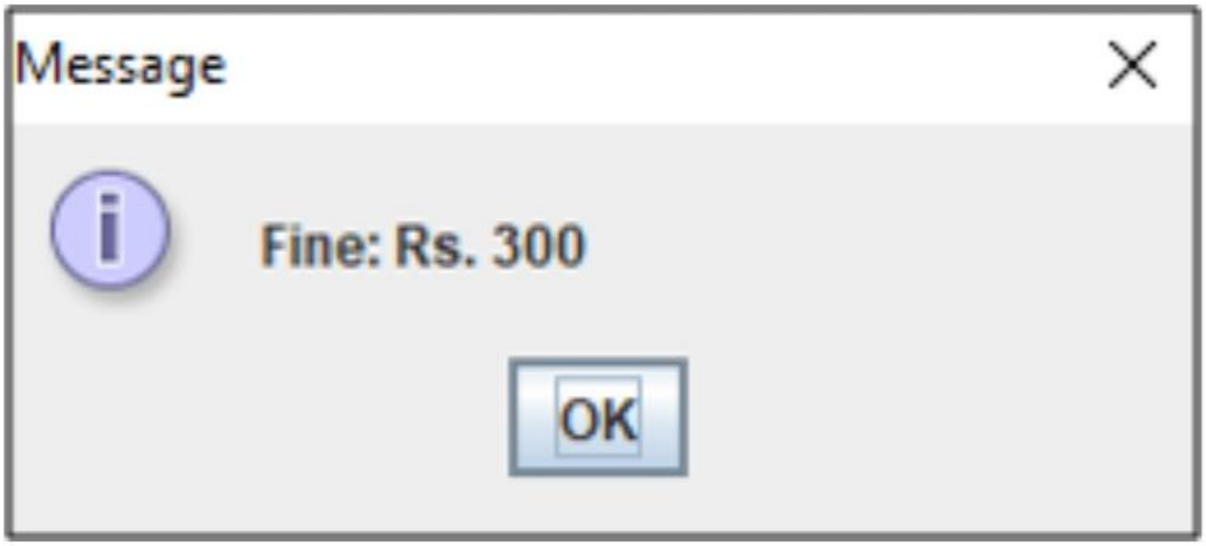
Once, you click on the Return Book, mention the Issue ID and the return date as below. Then click on Return.


Enter Details

Issue ID(IID) 1

Return Date(DD-MM-YYYY) 12-10-2019

Then, you see a message box displaying the fine.



Fine: Rs. 300

After that, you again see a dialog box, showing the message “Book Returned“. Refer below.

Add Book

Click on the Add Book button, and mention the book name, genre and price. Then, click on the Submit button. Refer below.

You will see a dialog box displaying the below message:

Apart from this, you can also, see the added books in the View Books section as below:

Validation check

The process of evaluating web-based application during the development process or at the end of the development process to determine whether it satisfied information requirement. Validation testing ensures that the product actually meets the user needs. It can also have defined as to demonstrate that the information fulfills its intended use when deployed on appropriate environment.

Validation testing can be best demonstrated. The web-based application under test is evaluated during this type of testing.

- **VALIDATION INPUT TRANSACTION:-**

Validation input data is largely done through website which is the programmer's responsibility but it is important that system analyst must know what a common problem might in validation a transaction.

Business committed to quality will include validation checks a part of their routine website.

1. Submitting the wrong data to system.
2. Submitting the data by an unauthorized person.
3. Asking the system to perform an unacceptable function.

- **VALIDATION INPUT DATA: -**

  It is essential that the input data themselves along with the transaction requested are valid. Several texts can be incorporated into website to ensure the validity. We consider many possible ways to validate input and they are as follows:

  1. Test for missing data.
  2. Test for correct field length.
  3. Test for range or reasonable.
  4. Test for comparison with stored data.

Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the “integrated” software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the “inter-assemblages” and also within the system as a whole. Our Project went through two levels of testing:

1. Unit testing

2.integration testing

UNIT TESTING:

Unit testing is undertaken when a module has been created and successfully reviewed. In order to test a single module, we need to provide a complete environment i.e.. besides the module we would require:

1.The procedures belonging to other modules that the module under test calls..
2.Non local data structures that module accesses.
3.A procedure to call the functions of the module under test with appropriate parameters.

1. **Test For the admin module:**

- **Testing admin login form-**This form is used for log in of administrator of the system. In this we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for username and password.

- **Student account addition-** In this section the admin can verify student details from student academic info and then only add student details to main library database it contains add and delete buttons if user click add button data will be added to student database and if he clicks delete button the student data will be deleted.

- **Book Addition-** Admin can enter details of book and can add the details to the main book table also he can view the books requests.

2. **Test for Student login module:**

- **Test for Student login Form-** This form is used for log in of Student. In this we enter the library id, username and password if all these are correct student login page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask for library id, username and password.

- **Test for account creation-** This form is used for new account creation when student does not fill the form completely it asks again to fill the whole form when he fill the form fully it gets redirected to page which show waiting for conformation message as his data will be only added by administrator after verification.

3. **Test for teacher login module:**

   - **Test for teacher login form-** This form is used for log in of teacher In this we enter the username and password if all these are correct teacher login page will

Open otherwise, if any of data is wrong it will get redirected back to the login page and again ask for username and password.

INTEGRATION TESTING:

In this type of testing, we test various integration of the project module by providing the input. The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.

IMPLEMENTATION, EVALUATION AND MAINTENANCE

Implementation

The design of a management information system may seem to management to be an expensive project, the cost of getting the MIS on line satisfactorily may often be comparable to that of its design, and the implementation has been accomplished when the outputs of the MIS are continuously utilized by decision makers. Once the design has been completed, there are four basic methods for implementing the MIS.

These are following:

1.Install the system in a new operation or organization.

2.Cut off the old system and install the new:

This produces a time gap during which no system is in operation. Practically, installation requires one or two days for small companies or small systems.

3.Cut over by segments:

This method is also referred as” phasing in” the new system. Small parts or subsystems are substituted for the old. In the case of upgrading old systems, this may be a very desirable method.

4.Operate in parallel and cut over:

The new system is installed and operated in parallel with the current system until it has been checked out, then only the current system is cut out. This method is expensive

Because of personal and related costs. Its big advantages are that the system is fairly well debugged when it becomes the essential information system.

Evolution

After the MIS has been operating smoothly for a short period of time, an evaluation of each step in the design and of the final system performance should be made. Evaluation should not be delayed beyond the time when the system’s analysts have completed most of the debugging. The longer the delay, the more difficult it will be for designer to remember important details. The evaluation should be made by the customer as well as by the designers.

Maintenance

Control and maintenance of the system are the responsibilities of the line managers. Control of the systems means the operation of the system as it was designed to operate. Sometimes, well-intentioned people or operators may make unauthorized changes to improve the system, changes that are not approved or documented. Maintenance is closely related to control. Maintenance is that ongoing activity that keeps the MIS at the highest levels of effectiveness and efficiency within cost constraints. Maintenance is directed towards reducing errors due to design, reducing errors due to environmental changes and improving the system's scope and services.

FUTURE SCOPE OF PROJECT

Our web-based application “Library Management System” which provides complete information about Users like Student, Admin and Lecturer. We will add more content on them in future. In our web-based application right now, only Books and Users with their information available but in future we will add Online Lectures, Links, etc.

We will also provide more images in GUI related to our web-based application in future. We will try to find out more about this topic and add in future. We will try to make this application more attractive so that visitor cannot get bored while using it. We will provide login id to each and every user so that he can access our website from anywhere through log in id and password. In future we add some major facilities like Reservation of Book.

We will also provide more images in GUI related to our web-based application in future. We will try to find out more about this topic and add in future. We will try to

make this application more attractive so that visitor cannot get bored while using it. We will provide login id to each and every user so that he can access our website from anywhere through log in id and password. In future we add some major facilities like Reservation of Book..

CONCLUSION :

Library Management System allows the user to store the book details and the customer details. This software package allows storing the details of all the data related to library. The system is strong enough to withstand regressive yearly operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

BIBLIOGRAPHY

Websites References:

https://www.tutorialspoint.com/index.htm

https://www.javatpoint.com

https://www.w3schools.com

https://html.com