



Name : KARTIK TYAGI

Class : XII

Certificate

This is to certify that *Kartik Tyagi* of Class XII of Delhi Public School Ghaziabad has worked under my supervision on the project titled “*Library Management*” and has completed it to my total satisfaction.

Acknowledgement

It gives us immense pleasure to express our sincere and deep feelings of gratitude to our Informatics teacher Mrs. Anubha Seth, who has given us his invaluable moral support from time to time, and helped & co-operated with us in preparation of this Project.

I am also grateful to My Parents and My Siblings for their constant help and cooperation.

Kartik Tyagi

XII

Index

SNo.	TOPIC	PAGE No.
1.	System Description at a Glance	5
2.	Problem Description	6
3.	System Requirement	6
4.	Database Design	7-9
5.	Forms, Buttons & Codes <ul style="list-style-type: none">• Login Form• Sign In Form• Front Page• Registered Books• Registered Visitors• New Registration• Transactions<ul style="list-style-type: none">➤ Issue a book➤ Return a book• Records Archives• Update database<ul style="list-style-type: none">➤ Update Book records<ul style="list-style-type: none">❖ Add a book❖ Dispose a book➤ Update Visitor records<ul style="list-style-type: none">❖ Dispose a Visitor➤ Reset Database Button• About Us link on Front page• Logout button on Front page	9-113
6.	Bibliography	114

SYSTEM DESCRIPTION AT A GLANCE

A Library is a place where one can access any information through books, newspapers and periodicals. It plays an important role in the growth of young individuals and has a vast impact on the life of people in general.

Any person can issue a book using a library card. Books of varied genres are available in a library. A person can borrow any book according to his suitable needs. Every person is allotted a library card with a specific number. This number can be used to view the person's records. Each book in a library is allotted an accession number which can be used to view specific details of the book.

The Librarian can view the transaction records and can get the details of each record. He can add new members/books to the library, the Library card number or the Accession number of each new record is provided by the software itself. He can dispose a member/ book from the library records. For safety measures the librarian is given an opportunity to retain a disposed record.

The Librarian can reset the database according to his will. To use this tool each librarian is allotted an administrator code.

Limitations of the project

- Mass transactions cannot be done.
- Once the database is reset, it cannot be undone.

Problem Description

This project is made on the topic of “Library Management” and is built for the use of a librarian. As it is difficult to deal with a lot of data and transactions, this project has been made to deal with all the problems and provide an efficient management system for a library.

All the requirements in a library as deemed fit by us are dealt with using this project. We have provided all basic functionalities which a librarian might require while carrying out his job. Each and every process being carried out in a library has been closely studied and analysed while making this project.

It has been our prime motive to make this project a user friendly one and we hope that it suitably deals with all the necessities.

System Requirement

- INTEL PROCESSOR SPEED : 500MH_z
- RAM: 128 MB FOR CLIENT AND 512 MB FOR SERVER.
- WINDOWS 98/XP/2000/NT.
- J2SE PLATFORM 6.0, 5.0 UPDATE 9/13.
- JAVA DEVELOPMENT KIT.
- INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)-NET BEANS.
- MySQL VERSION FOR WINDOWS (x86)(MySQL 4.0 OR 5.0 OR HIGHER).
- MySQL JDBC DRIVER.

Database design

```
drop database library;

create database library;

use library;

create table Book_information

(Accession_Number varchar(5),Title varchar(50) NOT NULL,
Author varchar(20) NOT NULL,Genre varchar(15) NOT NULL,
Price integer(4)DEFAULT '0', Stock_available integer(2) DEFAULT '0',
PRIMARY KEY(Accession_Number) );

create table User_Id

(Name varchar(30), Phone_No varchar(15), UserID varchar(50) NOT NULL,
Password varchar(20),Question varchar(100),Answer varchar(100),
Primary Key(Password));

create table Registered_Issuers

(LibraryCard_Number varchar(10) NOT NULL, Name_of_issuer varchar(20) NOT NULL,
Phone_numbers varchar(20), Address varchar(50),
Date_Registration date,
PRIMARY KEY(LibraryCard_Number) );

create table Transaction_Records

(Record_Number int(5) Auto_increment, LibraryCard_Number varchar(10),
Accession_Number varchar(5),Date_of_issue date, Date_of_return date,
Duration_of_issue varchar(10), Fine_Charged varchar(7),
Foreign Key(LibraryCard_Number) references Registered_Issuers (LibraryCard_Number) on
update cascade on delete restrict,
Foreign Key(Accession_Number) references Book_Information (Accession_Number) on
update cascade on delete restrict,
Primary Key(Record_Number) );

create table disposed
```

```
( Accession_number varchar(5), LibraryCard_number varchar(7));  
  
insert into Book_information values('B01', 'Fundamentals of Physics', 'Resnick & Halliday',  
'Physics', 250,1);  
  
insert into Book_information values('B02', 'Concepts of Physics Volume 1', 'H.C.  
Verma', 'Physics', 250,1);  
  
insert into Book_information values('B03', 'Concepts of Physics Volume 2', 'H.C.  
Verma', 'Physics', 230,1);  
  
insert into Book_information values('B04', 'Physics MCQ', 'D Mukherji', 'Physics', 290,1);  
  
insert into Book_information values('B05', 'Problems In Physics', 'M.K. Agarwal', 'Physics',  
280,1);  
  
insert into Book_information values('B06', 'Understanding Physics Mechanics', 'D.C. Pandey',  
'Physics', 170,1);  
  
insert into Book_information values('B07', 'Understanding Physics Optics Part 1', 'D.C.  
Pandey', 'Physics', 190,1);  
  
insert into Book_information values('B08', 'Understanding Physics Optics Part 2', 'D.C.  
Pandey', 'Physics', 200,1);  
  
insert into Book_information values('B09', 'Numerical Bank', 'G. Jaggi & K. Sharma',  
'Physics', 280,1);  
  
insert into Book_information values('B10', 'Fundamentals of Physics', 'Yavorsky & Pinsky',  
'Physics', 240,1);  
  
insert into Book_information values('B11', 'Calculus in One Variable', 'Thomas & Finney',  
'Maths', 300,1);  
  
insert into Book_information values('B12', 'Mathematics Class XII Vol 1', 'R.D.  
Sharma', 'Maths', 220,1);  
  
insert into Book_information values('B13', 'Mathematics Class XII Vol 2', 'R.D.  
Sharma', 'Maths', 260,1);  
  
insert into Book_Information values('B14', 'Higher Algebra', 'Hall & Knight', 'Maths', 190,1);  
insert into Book_Information values('B15', 'Mathematics', 'Mohit Khanna', 'Maths', 235,1);  
insert into registered_issuers values('L01', 'Vansh Passi', '9972565690', 'SH 123 Shastri  
Nagar', curdate());
```

Table description

Tables	Description
1. Book_information	To store information of books
2. Registered_issuers	To store information of members
3. Transaction_Records	To store records of transactions
4. User_Id	To store information of librarian
5. Disposed	To retain disposed records

Forms, Buttons & Code

Login Form :

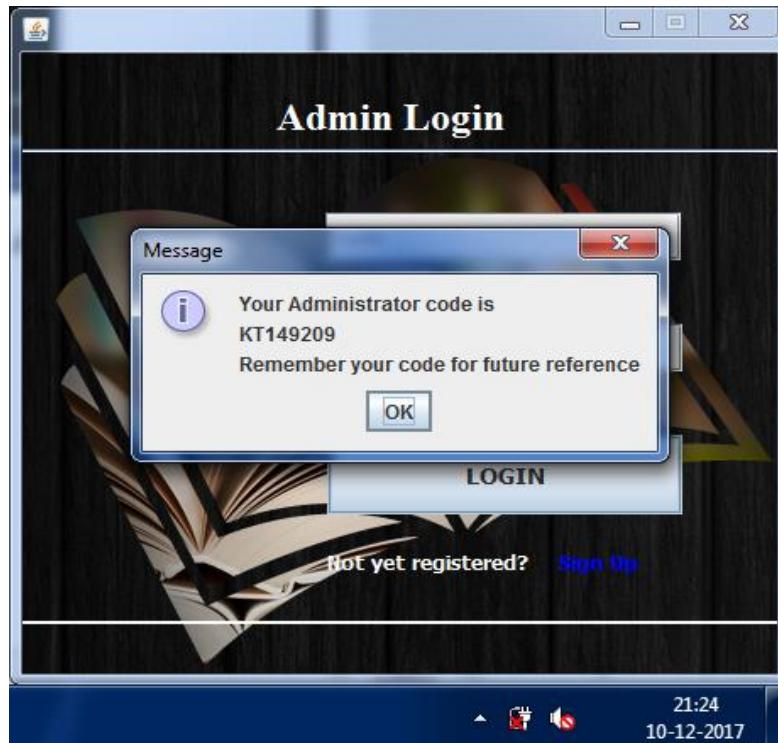
The Login form is the first page of the project and is used to access the main page. It requires a login ID and password for anyone to access it. It is a security measure to prevent unauthorised entry in the project.



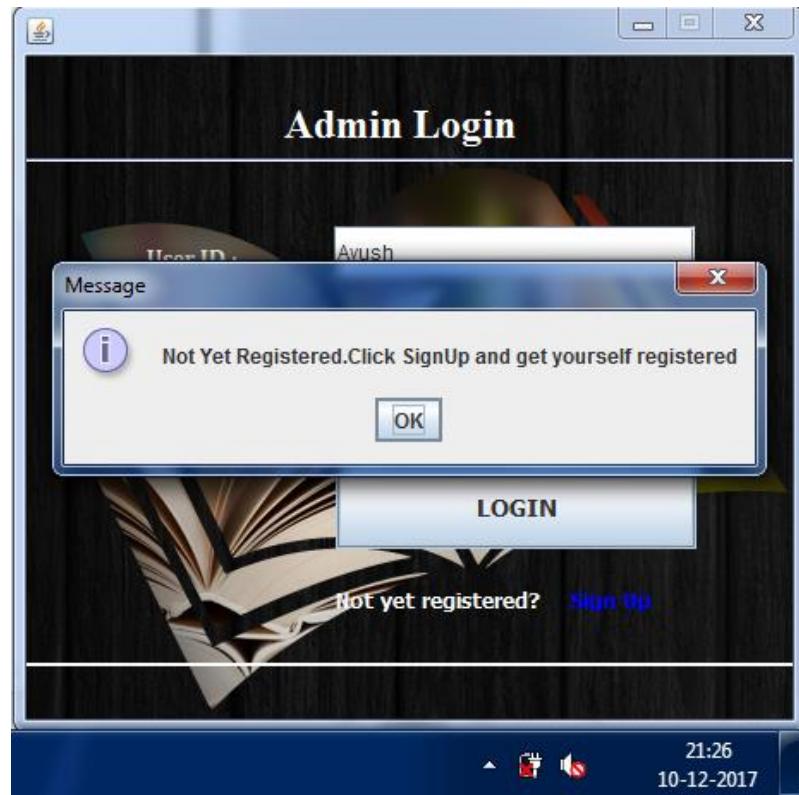
If an authorised User ID and Password is given the program checks the validity of the ID and Password and confirms registration.



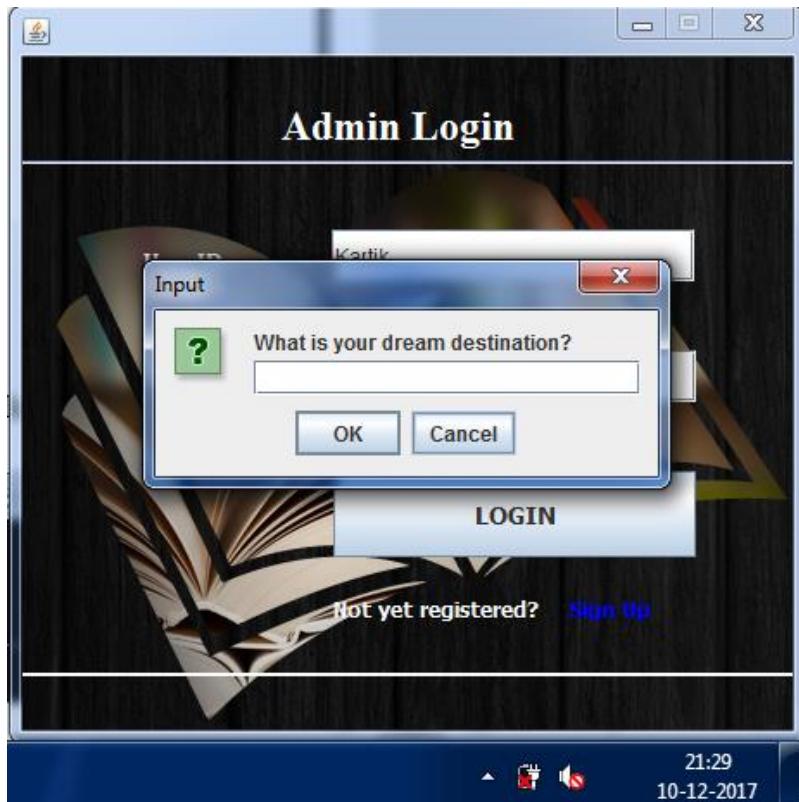
If the User ID or Password confirmed the login page gives the Administrator code to the authorised person.



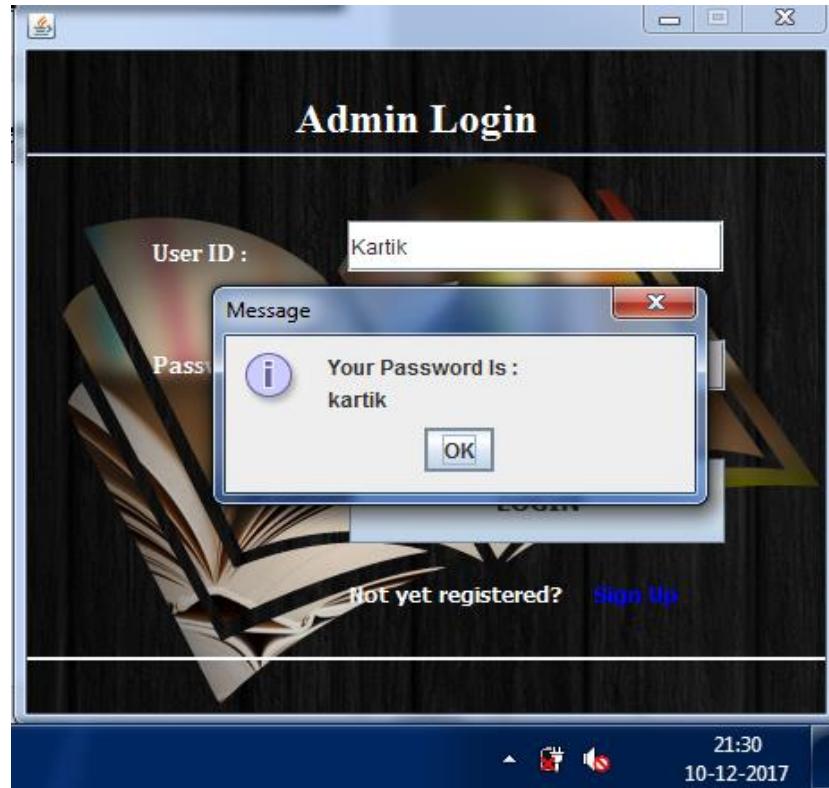
If the User ID or Password is incorrect the page prevents the entry of such unauthorised personnel and gives a suitable message.



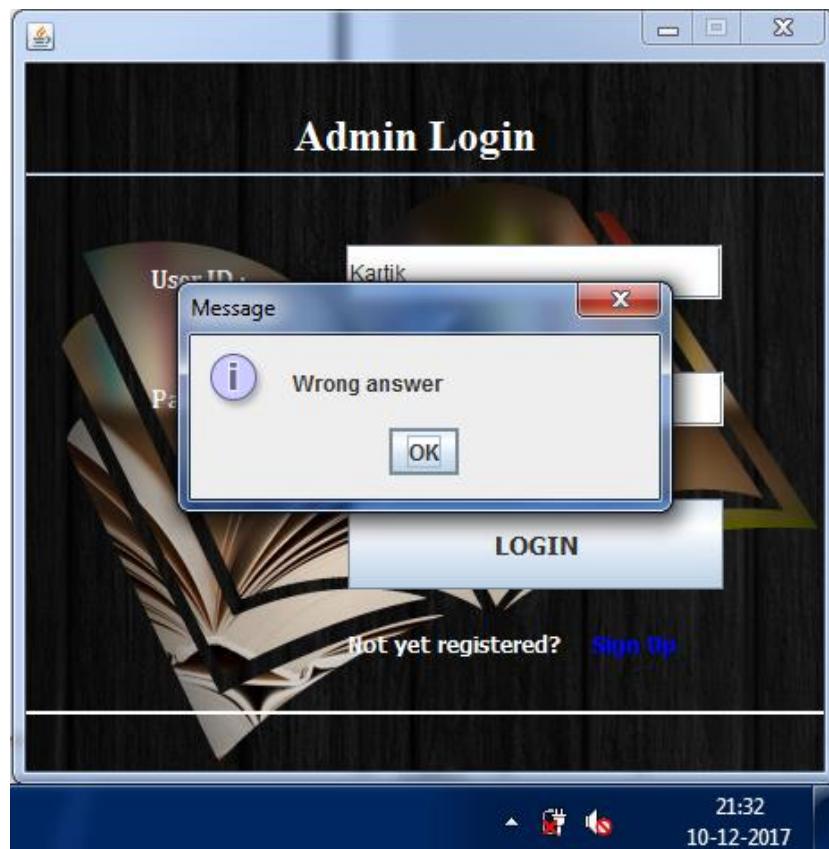
If the person forgets his password the can click on forgot password and give the answer to a security question as set by the person himself.



If the answer is correct the person gets his password along with access to the main page.



If the answer to the security question is wrong an appropriate message is displayed.



Coding for Login Form :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String a=jTextField1.getText();  
    try  
{  
        Class.forName("java.sql.Driver");  
        Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement)con.createStatement();
        String query="Select Question,Answer,Password,Name from User_id where UserID='"+a+"'";
        ResultSet rs=stmt.executeQuery(query);
        rs.first();
        String secques=rs.getString(1);
        String ans=rs.getString(2);
        String ansbyuser=JOptionPane.showInputDialog(this,secques);
        String libinch=rs.getString(4);
        ansbyuser.toLowerCase();
        ans.toLowerCase();
        if(ansbyuser.equals(ans))  
        {  
            String pass=rs.getString(3);
            JOptionPane.showMessageDialog(this,"Your Password Is :\n"+pass);
            Main_Page ob=new Main_Page();
            ob.setVisible(true);
            ob.setTitle("Librarian Incharge : "+libinch);
            this.dispose();
            this.setEnabled(false);
        }  
    else  
    {  
        JOptionPane.showMessageDialog(this,"Wrong answer");
        login_page ob=new login_page();
        ob.setVisible(true);
        this.dispose();
    }  
    stmt.close();
}
```

```

con.close();
}

catch(Exception e)
{
    if(e.getMessage().equals("Illegal operation on empty result set."))
    {
        JOptionPane.showMessageDialog(this,"Please enter valid username");
        jTextField1.setText("");
        jPasswordField1.setText("");
    }
    else
    {
        JOptionPane.showMessageDialog(this,e.getMessage());
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    int count=0;
    String uid,password;
    uid=jTextField1.getText();
    password=jPasswordField1.getText();
    if(uid.equals("")||password.equals ""))
    {
        JOptionPane.showMessageDialog(this,"Please enter the required information");
    }
    else
    {
        try
        {
            Class.forName("java.sql.Driver");
            Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
            Statement stmt=(Statement)con.createStatement();
            String query="Select userID,password,Name from user_id";
            ResultSet rs=stmt.executeQuery(query);
            while(rs.next())
        
```

```

{
    String c1=rs.getString(1);
    String c2=rs.getString(2);
    if((uid.equals(c1))&&(password.equals(c2)))
    {
        String c3=rs.getString(3);
        JOptionPane.showMessageDialog(this,"Registration Confirmed");
        JOptionPane.showMessageDialog(this,"Your Administrator code is \n"+ "KT149209 \n"+
                                         "Remember your code for future reference");

        Main_Page ob=new Main_Page();
        ob.setVisible(true);
        ob.setTitle("Librarian Incharge : "+c3);
        this.dispose();
        this.setEnabled(false);
        count=1;
        break;
    }
    else
    {
        count=0;
        continue;
    }
}
if(count==0)
{
    JOptionPane.showMessageDialog(this,"Not Yet Registered.Click SignUp and get yourself
registered");
    jTextField1.setText("");
    jPasswordField1.setText("");
}
}

catch(Exception e)
{
    if(e.getMessage().equals("Illegal operation on empty result set."))
    {

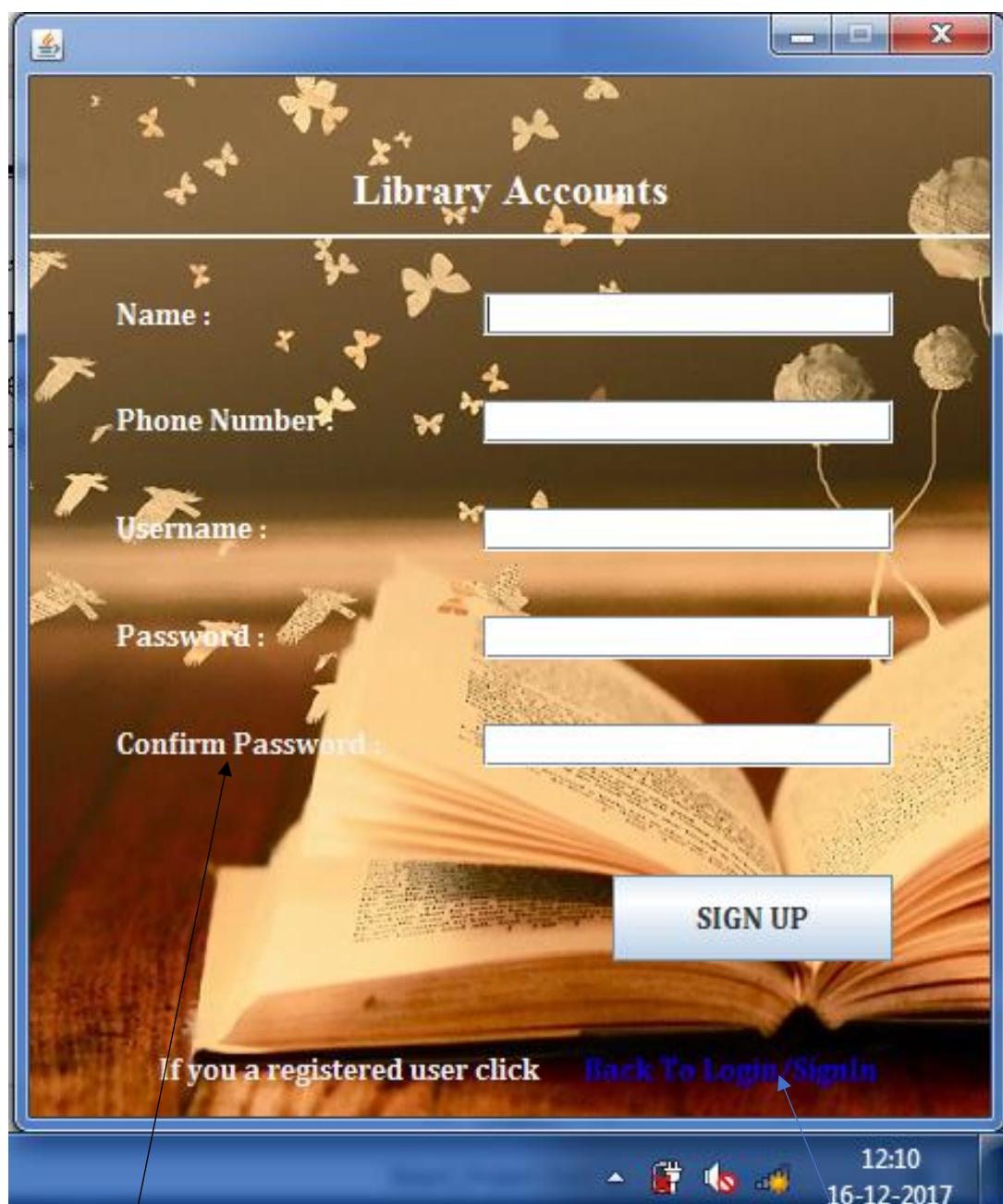
```

```
JOptionPane.showMessageDialog(this,"Not registered");
jTextField1.setText("");
jPasswordField1.setText("");
}
else
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
}
}
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
Sign_Up_Page ob=new Sign_Up_Page();
ob.setVisible(true);
this.dispose( );
```

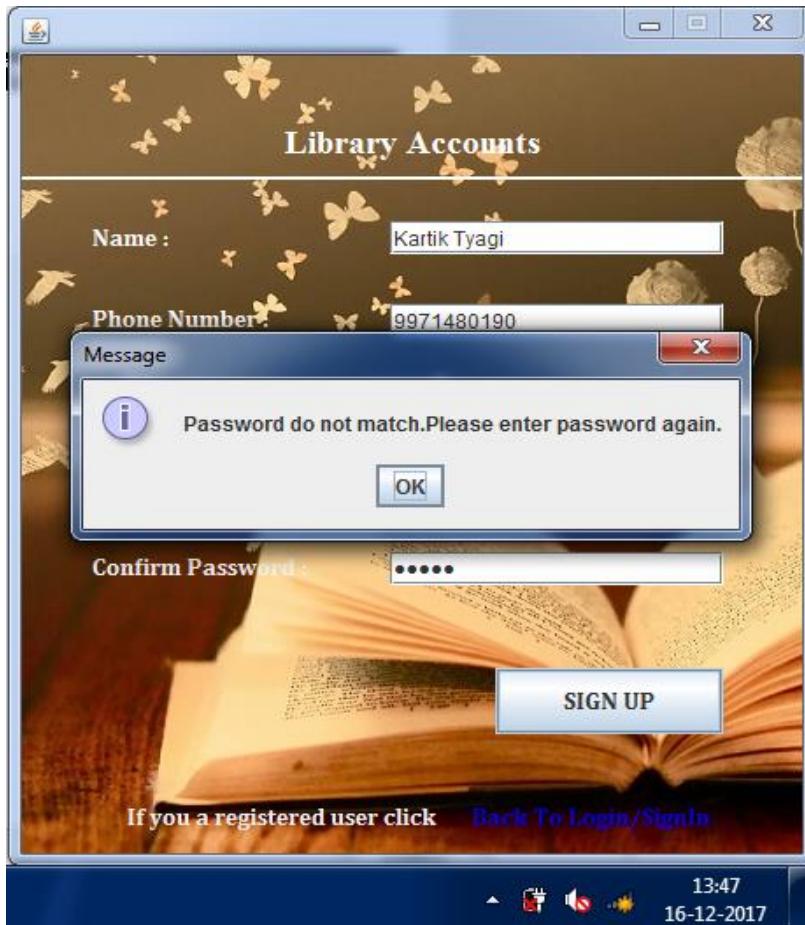
Sign Up Form :

Sign In from opens when the user clicks on the “Sign Up” label on the login page. It is used to create a new library account for a librarian. It requires some information of the person being registered. It also provides an opportunity to set **Security question** and **Security answer** for the person’s convenience in case he forgets his password.

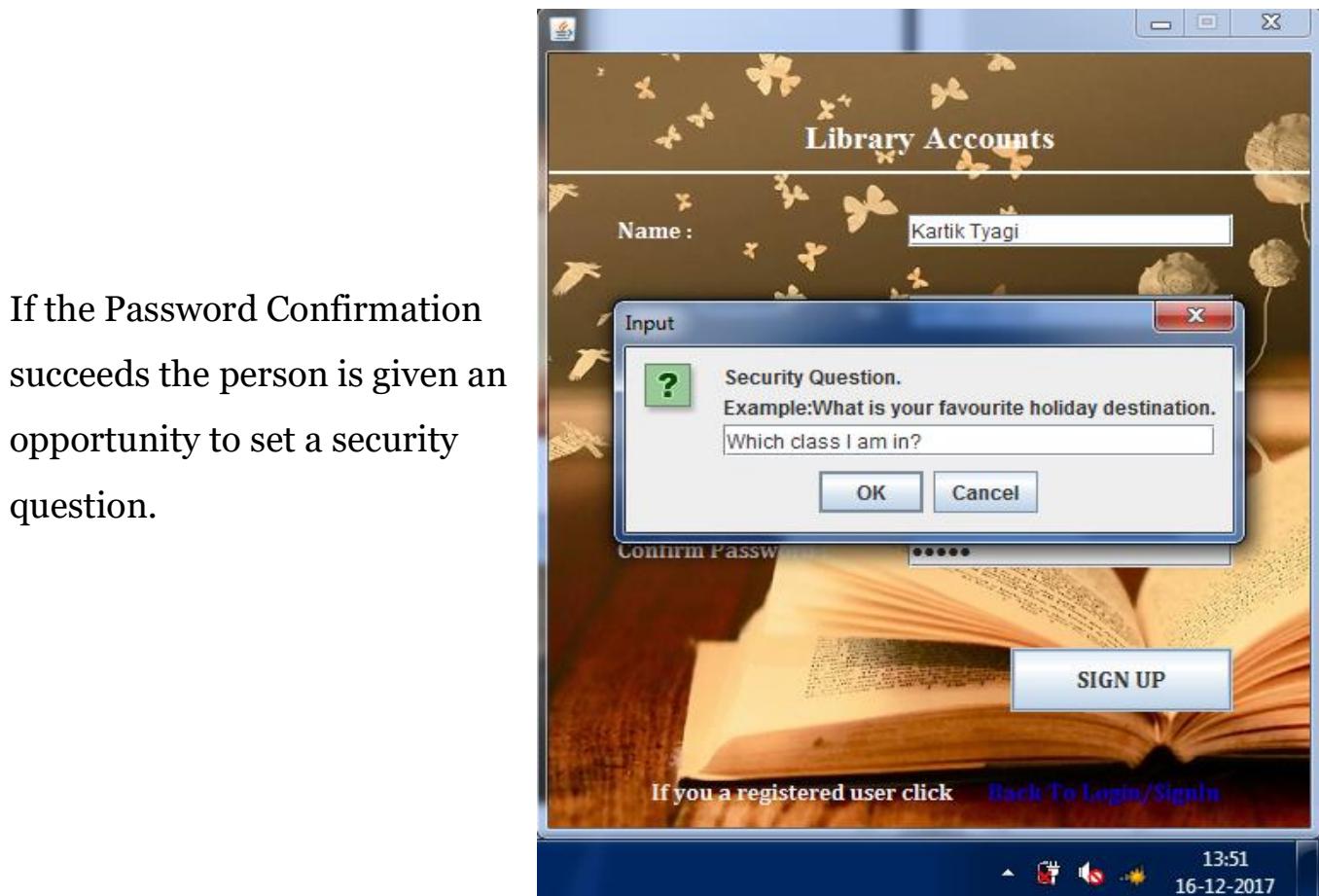


Password confirmation

Can be used to go back to Login form.

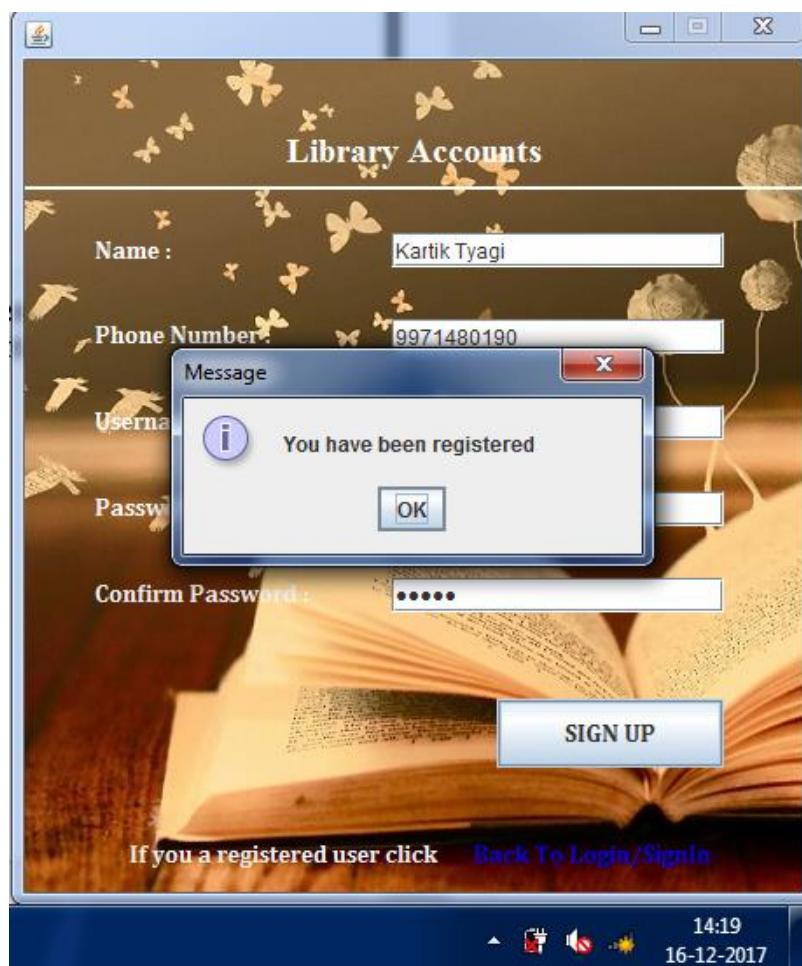
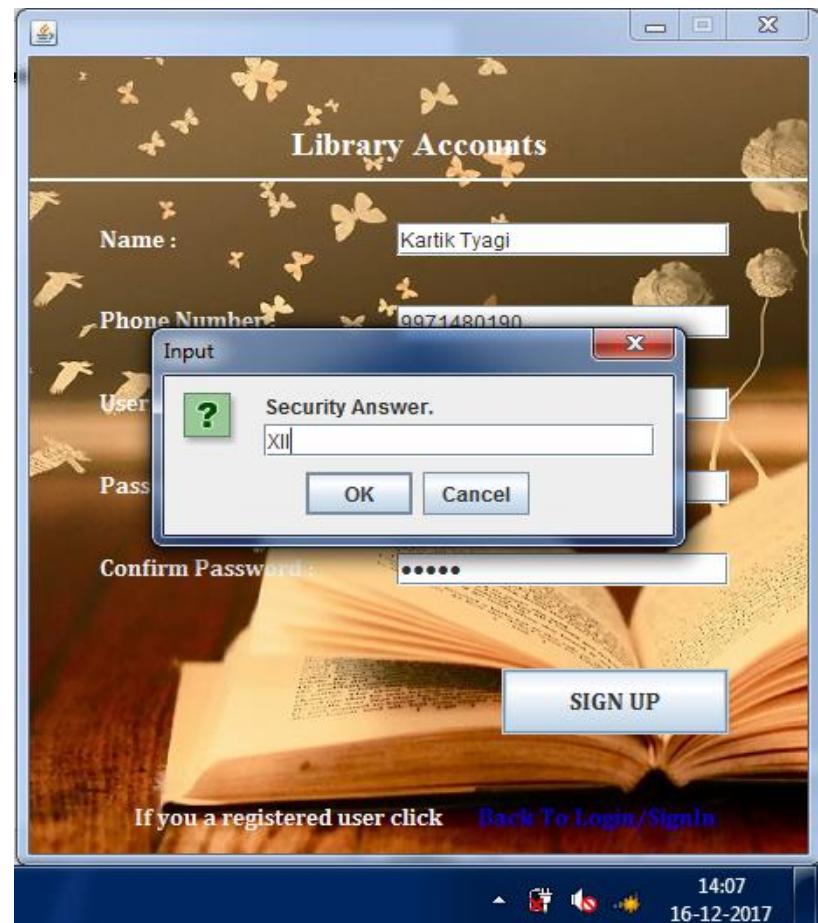


If Password Confirmation fails
the window displays a suitable
message.



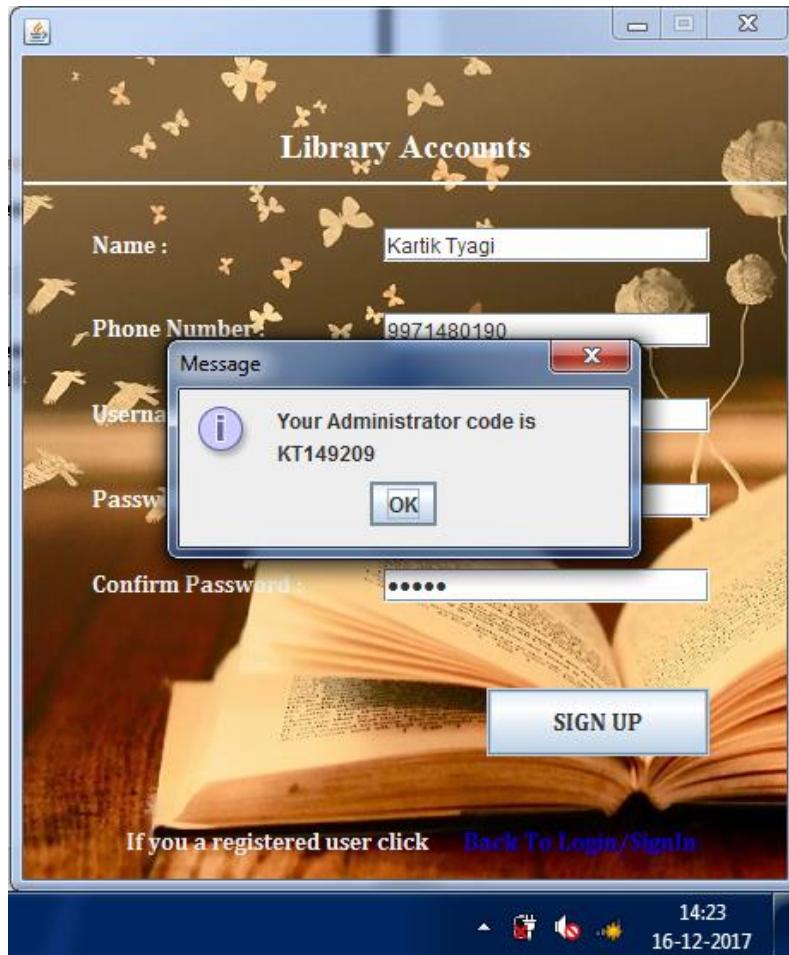
If the Password Confirmation
succeeds the person is given an
opportunity to set a security
question.

Also to set its Security Answer.



If we get a successful registration an appropriate message is displayed.

For the first time the newly registered librarian gets his Administrator Code.



Coding for Sign Up Form :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
login_page ob=new login_page();  
ob.setVisible(true);  
this.dispose();  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
String name,pno,uname,password,cpassword;  
name=jTextField1.getText();  
pno=jTextField2.getText();  
uname=jTextField3.getText();  
password=jPasswordField1.getText();  
cpassword=jPasswordField2.getText();  
if(name.equals("")||pno.equals("")||uname.equals("")||password.equals("")||cpassword.equals(""))  
{
```

```

JOptionPane.showMessageDialog(this,"Please enter the required information.");
}

else if(password.equals(cpassword)!=true)
{
    JOptionPane.showMessageDialog(this,"Password do not match.Please enter password again.");
    jPasswordField1.setText("");
    jPasswordField2.setText("");
}
else
{
    String ques=JOptionPane.showInputDialog(this,"Security Question. \n"+ "Example:What is your favourite
holiday destination.");

    String ans="";
    while(ques.equals(""))
    {
        ques=JOptionPane.showInputDialog(this,"Security Question. \n"+ "Example:What is your favourite holiday
destination?");

    }
    if(ques.equals("")==false)
    {
        ans=JOptionPane.showInputDialog(this,"Security Answer.");
        while(ans.equals(""))
        {
            ans=JOptionPane.showInputDialog(this,"Security Answer.");
        }
    }
    try
    {
        Class.forName("java.sql.Driver");
        Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement)con.createStatement();
        String query="Insert into user_id values
        ("+name+","+pno+","+uname+","+password+","+ques+","+ans+");";
        int i=stmt.executeUpdate(query);
        if(i==1)
        {

```

```

JOptionPane.showMessageDialog(this, "You have been registered");
JOptionPane.showMessageDialog(this, "Your Administrator code is \n"+ "KT149209");
Main_Page ob=new Main_Page();
ob.setVisible(true);
this.dispose();
}
else if(i==0)
{
JOptionPane.showMessageDialog(this, "You could not be registered");
}
stmt.close();
con.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(this,"Sorry!Internal Error.");
}
}
}

```

```

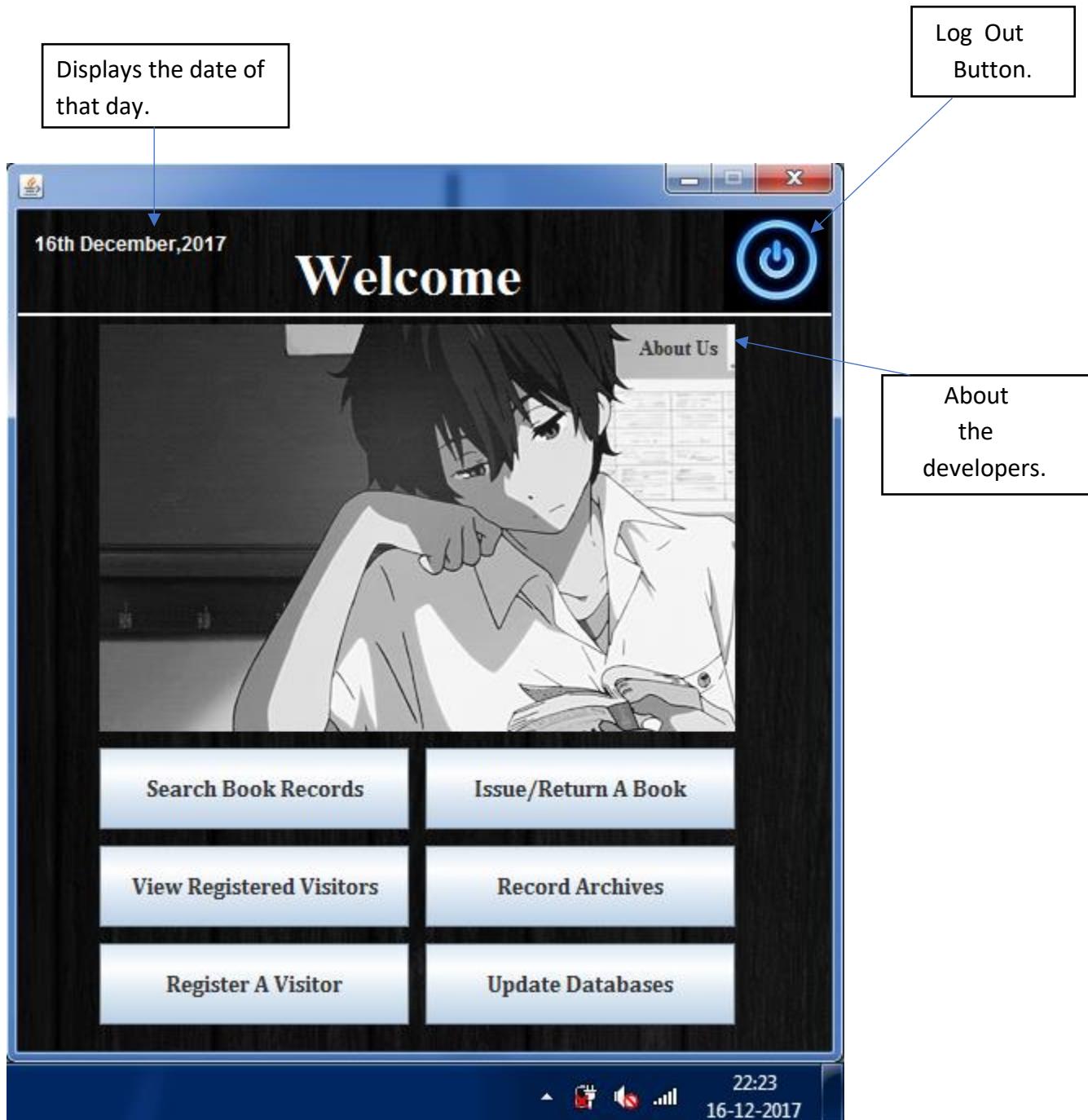
private void jTextField2KeyTyped(java.awt.event.KeyEvent evt) {
char c=evt.getKeyChar();
if(c>='0'&&c<='9')
{
}
else
{
    evt.consume();
}
}

```

Main Page Form :

Main Page (Welcome Page) is the most important page in the project. It is the page which provides various functionalities to the user. It displays the name of the user who has logged in. It contains various buttons providing specific functions. The user can :

- Search book records of registered books.
- Search member records of registered members.
- Register a new visitor.
- Issue / Return a book for a member.
- View transaction records through records archives.
- Update database.



Coding for Main Page Form :

```
private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {  
    About_Us ob=new About_Us();  
    ob.setVisible(true);  
    this.dispose();  
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    int i=JOptionPane.showConfirmDialog(this,"Do you wish to  
    logout?","Logout",JOptionPane.YES_NO_CANCEL_OPTION);  
    if(i==0)  
    {  
        login_page ob=new login_page();  
        ob.setVisible(true);  
        this.dispose();  
    }  
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Search_Book_Records ob=new Search_Book_Records();  
    ob.setVisible(true);  
    this.dispose();  
}
```

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {  
    Transactions ob=new Transactions();  
    ob.setVisible(true);  
    this.dispose();  
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    registered_visitors ob=new registered_visitors();  
    ob.setVisible(true);
```

```

this.dispose();
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
record_archives ob=new record_archives();
ob.setVisible(true);
this.dispose();
}

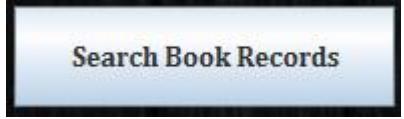
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
register_new_visitor ob = new register_new_visitor();
ob.setVisible(true);
this.dispose();
}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
database_updation ob=new database_updation();
ob.setVisible(true);
this.dispose();
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
try
{
Class.forName("java.sql.Driver");
Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt=(Statement)con.createStatement();
Statement stmt1=(Statement)con.createStatement();
String s="select Day(curdate()),MonthName(curdate()),Year(curdate());";
ResultSet rs=stmt.executeQuery(s);
rs.first();
String s1=rs.getString(1);
}

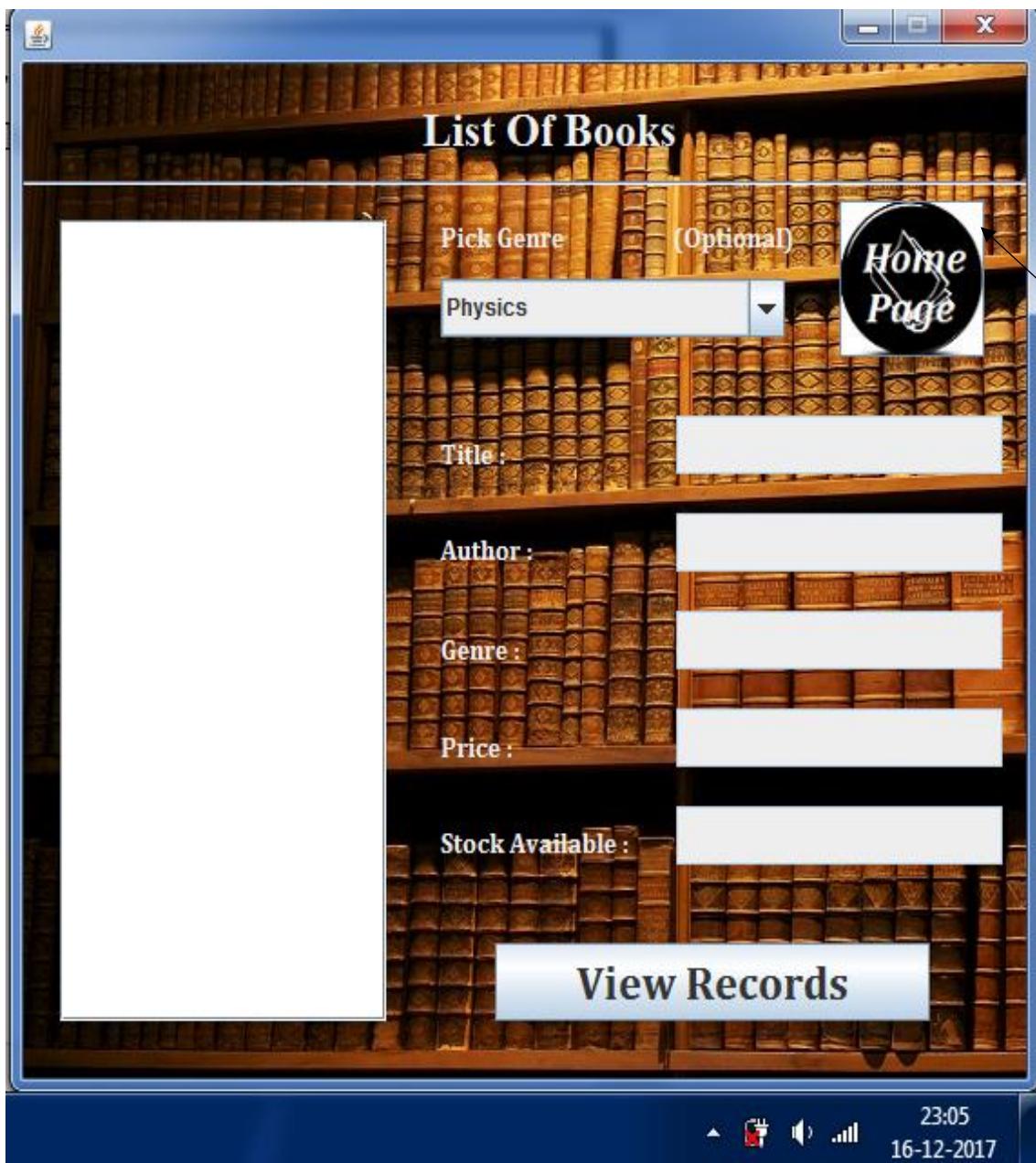
```

```
String s2=rs.getString(2);
String s3=rs.getString(3);
if(Integer.parseInt(s1)==1)
{
    jLabel5.setText(s1+"st "+s2+","+s3);
}
else if(Integer.parseInt(s1)==2)
{
    jLabel5.setText(s1+"nd "+s2+","+s3);
}
else if(Integer.parseInt(s1)==3)
{
    jLabel5.setText(s1+"rd "+s2+","+s3);
}
else if(Integer.parseInt(s1)>=4)
{
    jLabel5.setText(s1+"th "+s2+","+s3);
}
}
catch(Exception e)
{
    jLabel5.setText("");
}
```

If a person clicks on 

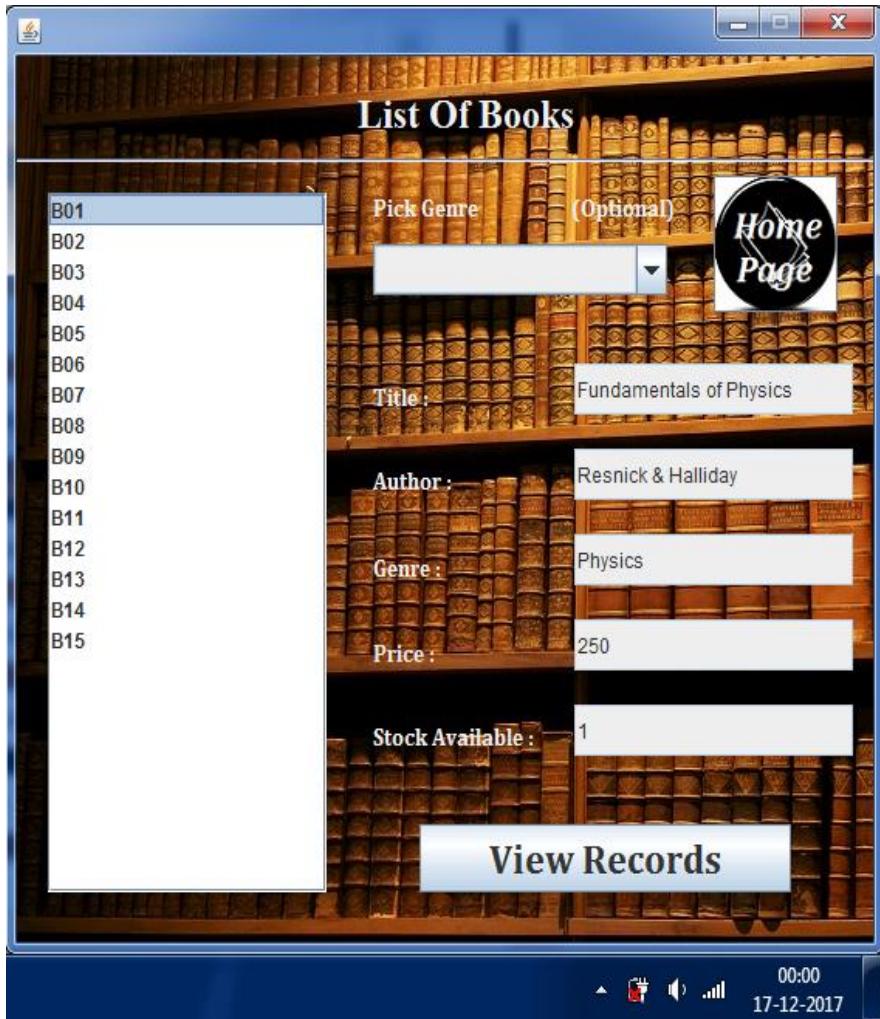
button.

The following window opens it provides the user an opportunity to view the books already registered in the library.



The Registered Book form gives us access to the details of books already registered in the library. It contains a combo box to pick genre of the book. If no genre is picked it displays all the records. Each record contains the book's accession number which is displayed in the list on the left hand side and if the user selects any record, details of the specific record are displayed in the text fields.

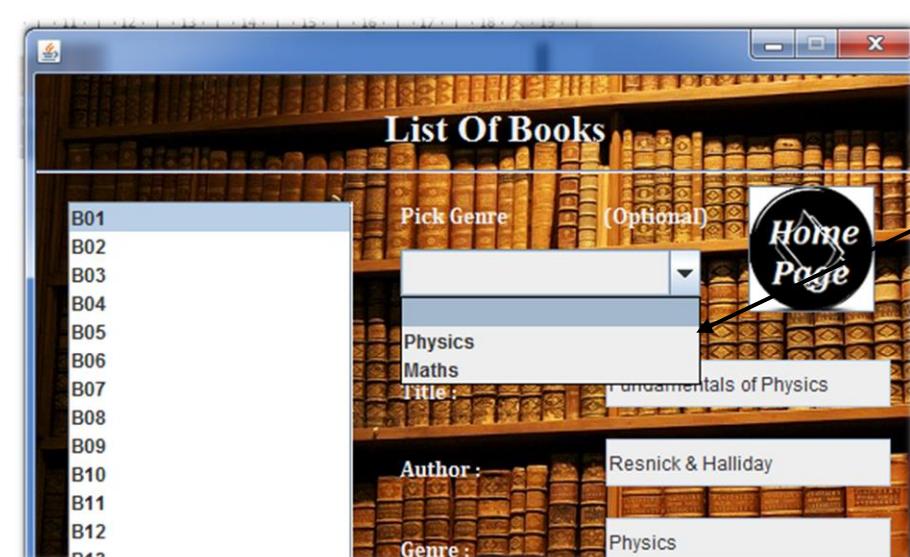
Picking no genre and selecting a book no/accession no.



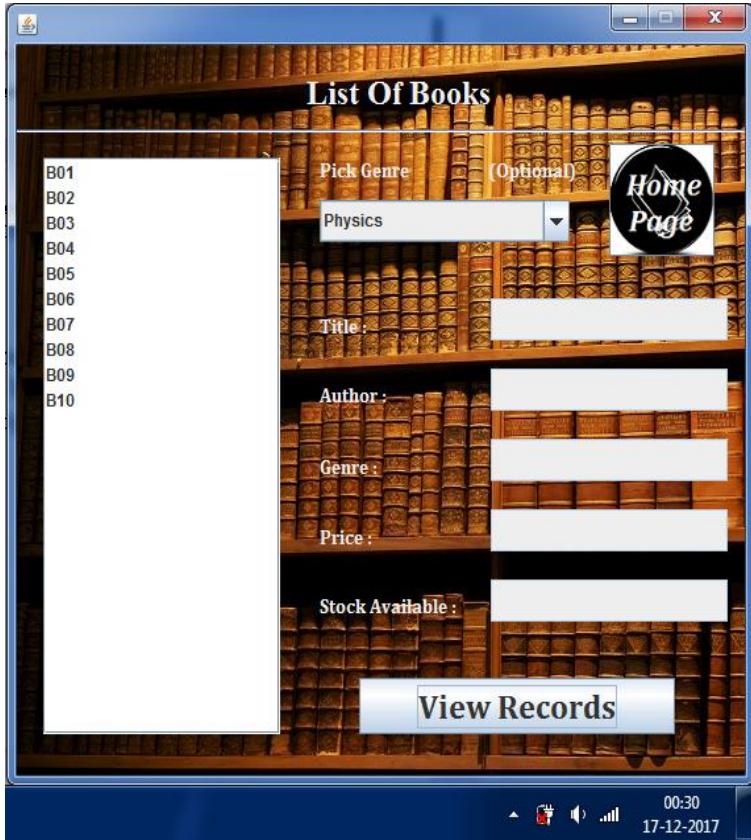
Here the stock available is 1 which means that the book has not been issued



Picking a genre

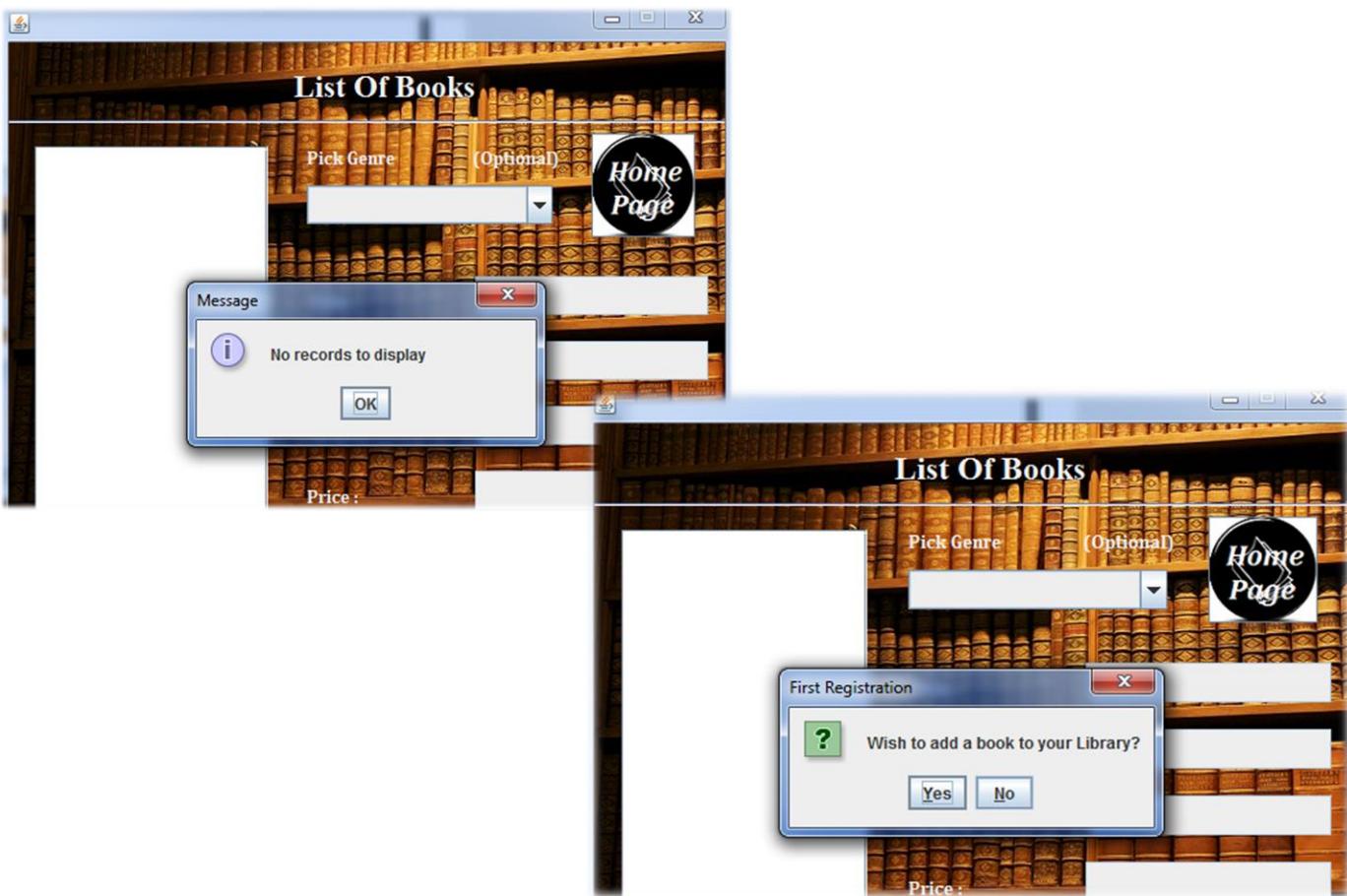


Genre of books that are already registered are displayed in the Combo Box.

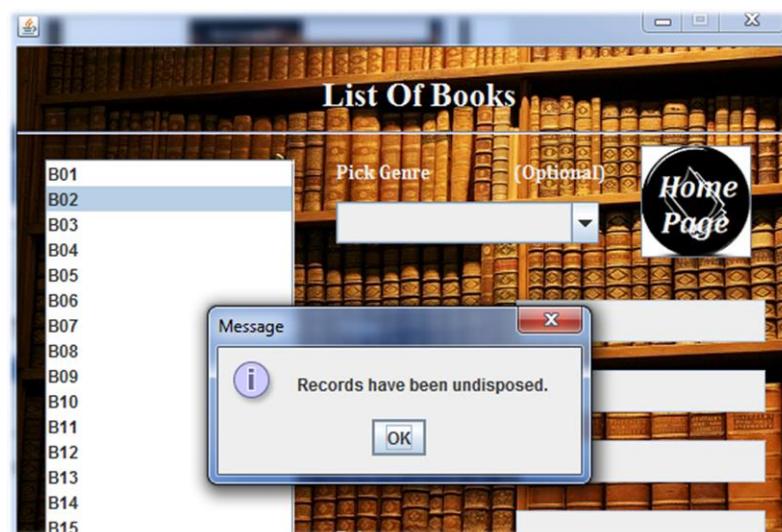
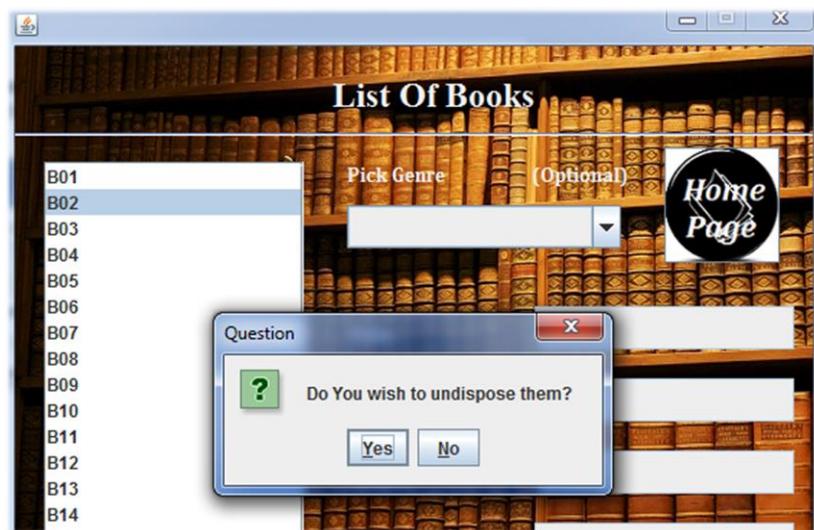
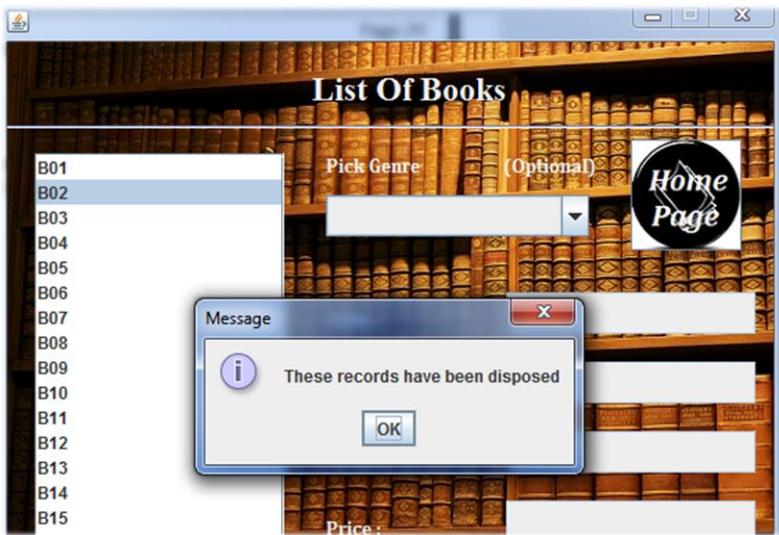


Picking a genre only accession numbers of books of that genre are displayed.

If there are no books records to display, then the window displays the message accordingly. The program even asks he user if he wishes to register a new book to the library. The user's choice leads him to the right form.



If the records of a book are disposed then we cannot view the book details. But he can retain the information.



Coding for Search Book Form :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    Main_Page ob=new Main_Page();  
    ob.setVisible(true);  
    this.dispose();  
}  
  
  
private void jList1MouseClicked(java.awt.event.MouseEvent evt) {  
  
try  
{  
    Class.forName("java.sql.Driver");  
    Connection  
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    Object l1=jList1.getSelectedValue();
    String query="select * from disposed";
    ResultSet rs=stmt.executeQuery(query);
    int count=0;
    if(rs.next())
    {
        while(rs.next())
        {
            if(l1.equals(rs.getString(1)))
            {
                count=1;
                break;
            }
        }
    }
    if(count==1)
    {
        JOptionPane.showMessageDialog(this,"These records have been disposed");
        int i=JOptionPane.showConfirmDialog(this,"Do You wish to undispose them?","Question",JOptionPane.YES_NO_OPTION);
    }
}
```

```

        if(i==0)
        {
            Statement stmts=(Statement)con.createStatement();
            String query1="Delete from disposed where accession_number='"+l1+"'";
            stmts.executeUpdate(query1);
            JOptionPane.showMessageDialog(this,"Records have been undisposed.");
        }
    }

    if(count==0)
    {
        String query2="Select Title,Author,Genre,Price,Stock_Available from book_information
where accession_number='"+l1+"';

        ResultSet rs1=stmt.executeQuery(query2);
        while(rs1.next())
        {
            jTextField1.setText(rs1.getString(1));
            jTextField2.setText(rs1.getString(2));
            jTextField3.setText(rs1.getString(3));
            jTextField4.setText(rs1.getString(4));
            jTextField5.setText(rs1.getString(5));
        }
    }

    stmt.close();
    con.close();
}

catch(Exception e)
{
    if(jList1.getSelectedValue()==null)
    {
        JOptionPane.showMessageDialog(this,"No book Selected");
    }
    else
    {

```

```

JOptionPane.showMessageDialog(this,e.getMessage());
}

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField5.setText("");

DefaultListModel l=(DefaultListModel)jList1.getModel();
l.clear();

if(jComboBox1.getSelectedItem()!="")
{
    try
    {
        Class.forName("java.sql.Driver");
        Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement)con.createStatement();
        Statement stmt1=(Statement)con.createStatement();
        String query="select accession_number from book_information order by Accession_number asc;";
        String query1="Select count(accession_number) from book_information;";
        ResultSet rs=stmt.executeQuery(query);
        ResultSet rs1=stmt1.executeQuery(query1);
        rs1.next();
        int c=Integer.parseInt(rs1.getString(1));
        if(c==0)
        {
            JOptionPane.showMessageDialog(this,"No records to display");
            int i=JOptionPane.showConfirmDialog(this,"Wish to add a book to your Library?","First
Registration",JOptionPane.YES_NO_OPTION);
            if(i==0)
            {

```

```

new_book_addition ob= new new_book_addition();
ob.setVisible(true);
this.dispose();
}
else
{
Search_Book_Records ob=new Search_Book_Records();
ob.setVisible(true);
this.dispose();
}
}

while(rs.next())
{
String a=rs.getString(1);
l.addElement(a);
}
stmt.close();
stmt1.close();
con.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
}
else
{
try
{
Class.forName("java.sql.Driver");
Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt=(Statement)con.createStatement();

```

```

Statement stmt1=(Statement)con.createStatement();
String val=(String)jComboBox1.getSelectedItem();
String query="Select accession_number from book_information where Genre='"+val+"' order by
Accession_number asc;";
String query1="Select count(accession_number) from book_information;";
ResultSet rs=stmt.executeQuery(query);
ResultSet rs1=stmt1.executeQuery(query1);
rs1.next();
int c=Integer.parseInt(rs1.getString(1));
if(c==0)
{
    JOptionPane.showMessageDialog(this,"No records to display");
    int i=JOptionPane.showConfirmDialog(this,"Wish to add a book to your Library?","First
Registration",JOptionPane.YES_NO_OPTION);
    if(i==0)
    {
        new_book_addition ob= new new_book_addition();
        ob.setVisible(true);
        this.dispose();
    }
    else
    {
        Search_Book_Records ob=new Search_Book_Records();
        ob.setVisible(true);
        this.dispose();
    }
}
while(rs.next())
{
    String accessno=rs.getString(1);
    l.addElement(accessno);
}
stmt.close();
stmt1.close();

```

```

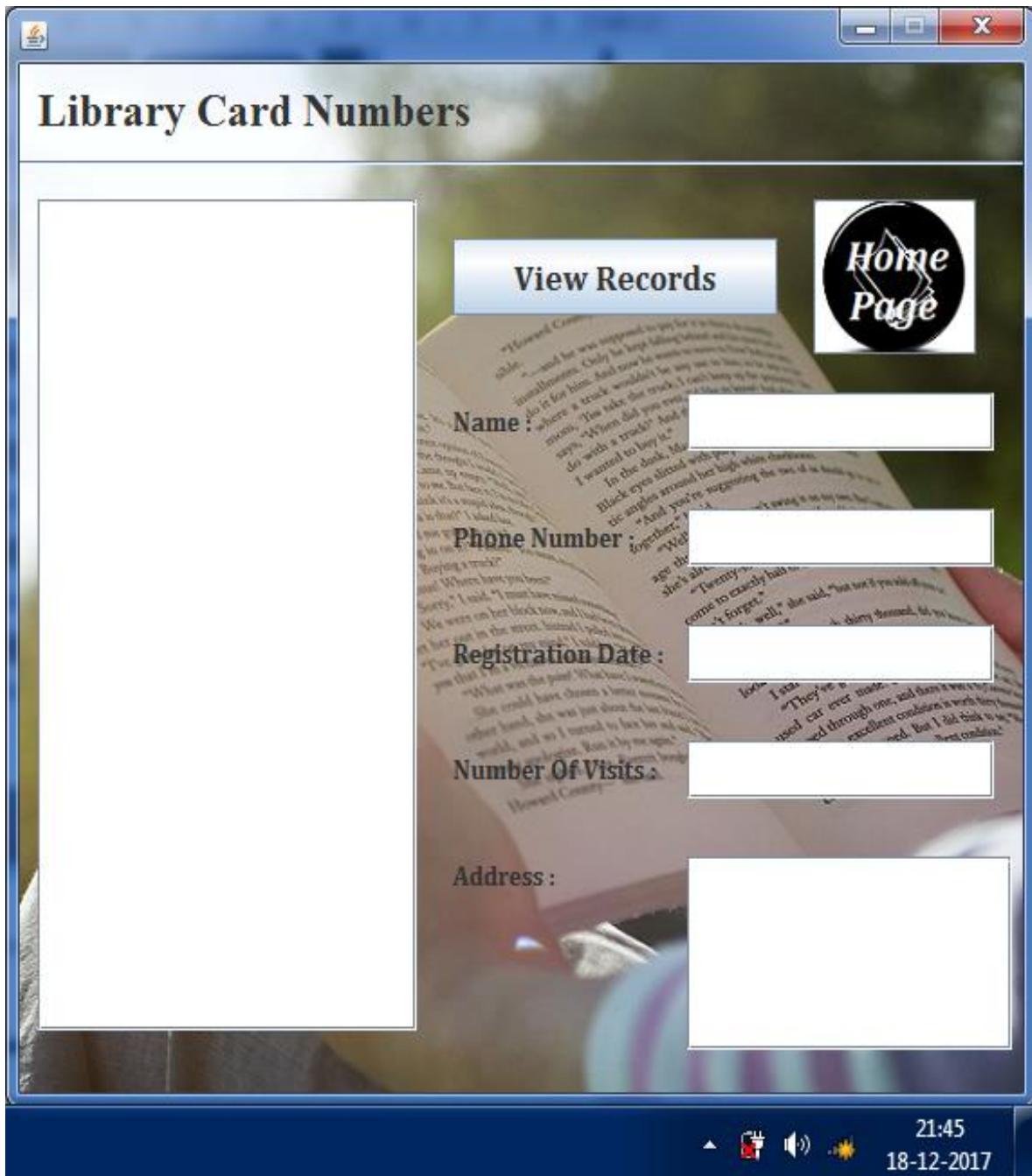
        con.close();
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e.getMessage());
    }
}
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
DefaultComboBoxModel cb=(DefaultComboBoxModel)jComboBox1.getModel();
cb.addElement("");
try
{
    Class.forName("java.sql.Driver");
    Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    String query="Select distinct genre from book_information;";
    ResultSet rs=stmt.executeQuery(query);
    while(rs.next())
    {
        String genre=rs.getString(1);
        cb.addElement(genre);
    }
    stmt.close();
    con.close();
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}
}

```

If a person clicks on **View Registered Visitors** button.

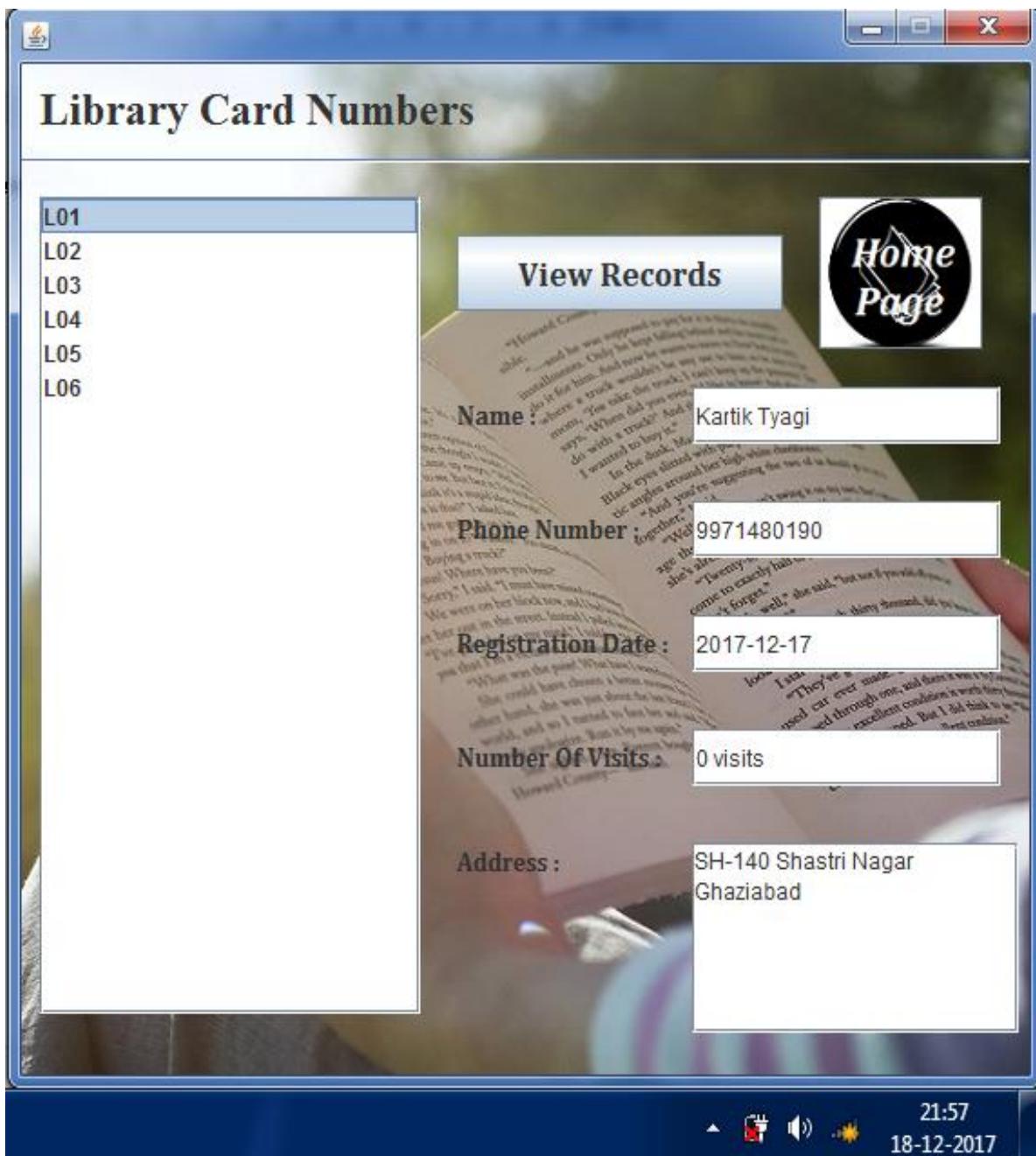
The following window opens it provides the user an opportunity to view the members already registered in the library.



The Registered Visitors form gives us access to the details of visitors already registered in the library. It displays records of all members. Each record contains the person's library card number which is displayed in the list on the left hand side and if the user selects any record, details of the specific record are displayed in the text fields.

Viewing the records.

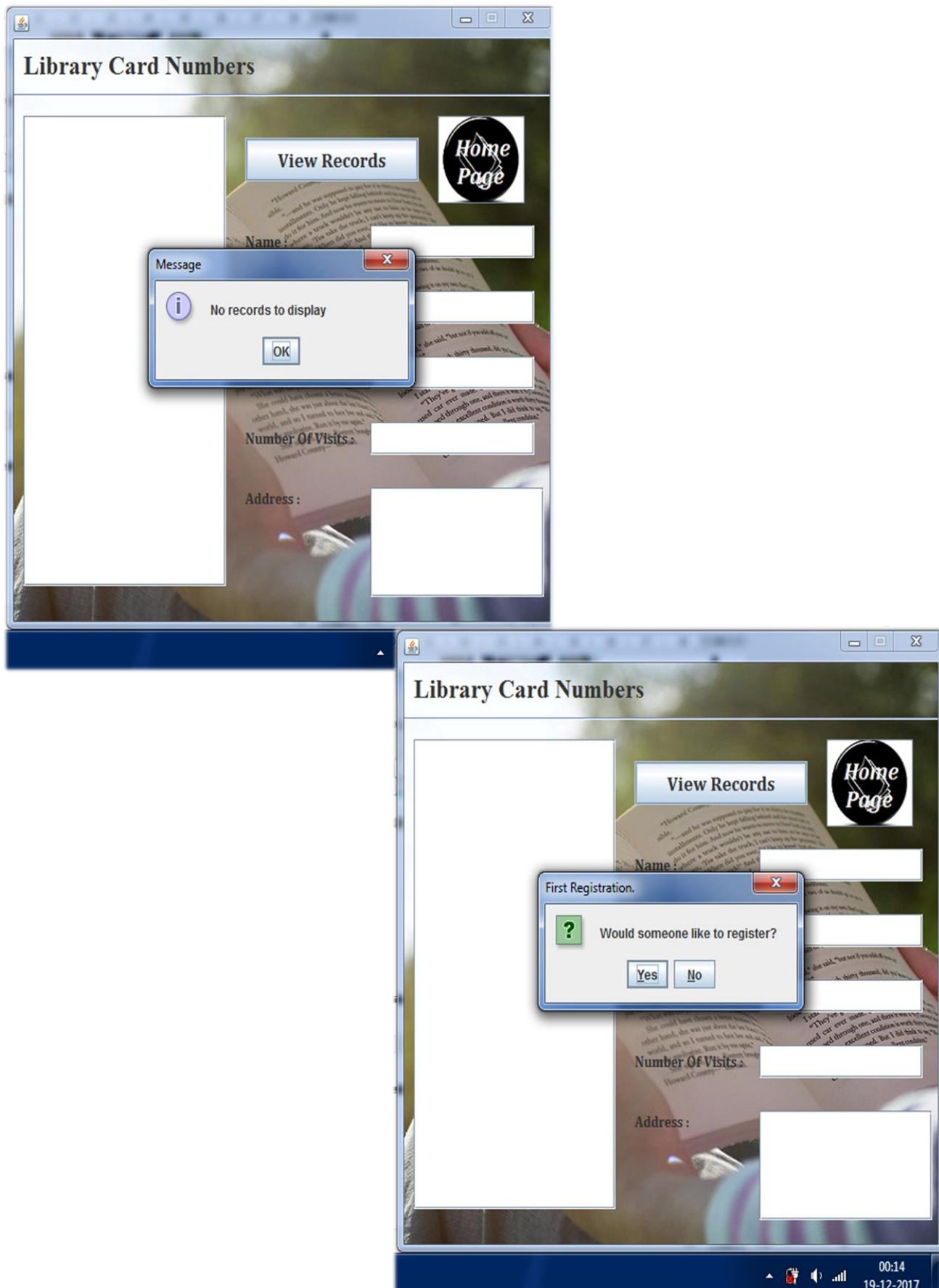
All the library card numbers are displayed in the list, and if the user selects any library card number the details of the person are displayed as shown above.



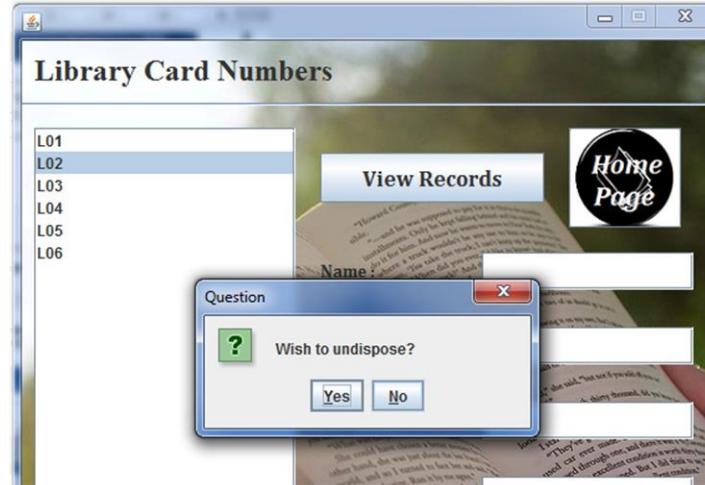
Here the number of visits is zero it means that the person has visited the library zero times.



If there are no member records to display, then the window displays the message accordingly. The program even asks the user if he wishes to register a new member to the library. The user's choice leads him to the right form.



If the records of a member are disposed then we cannot view his details. But he can retain the information.



Coding for Search Visitor Form :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
Main_Page ob=new Main_Page();  
ob.setVisible(true);  
this.dispose();  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
DefaultListModel l=(DefaultListModel)jList1.getModel();  
l.clear();  
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
jTextField4.setText("");  
jTextArea1.setText("");  
  
try  
{  
    Class.forName("java.sql.Driver");  
    Connection  
    con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    Statement stmt1=(Statement)con.createStatement();
    String query="Select LibraryCard_Number from Registered_issuers";
    String query1="Select count(LibraryCard_Number) from Registered_issuers;";
    ResultSet rs=stmt.executeQuery(query);
    ResultSet rs1=stmt1.executeQuery(query1);
    rs1.next();
    System.out.println(rs1);
    int c=Integer.parseInt(rs1.getString(1));
    if(c==0)
    {
        JOptionPane.showMessageDialog(this,"No records to display");
        int i=JOptionPane.showConfirmDialog(this,"Would someone like to register?","First
Registration.",JOptionPane.YES_NO_OPTION);
    }
}
```

```

if(i==0)
{
    register_new_visitor ob= new register_new_visitor();
    ob.setVisible(true);
    this.dispose();
}

else
{
    registered_visitors ob=new registered_visitors();
    ob.setVisible(true);
    this.dispose();
}

}

while(rs.next())
{
    String s=rs.getString(1);
    l.addElement(s);
}

stmt.close();
stmt1.close();
con.close();
}

catch(Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}

```

private void jList1MouseClicked(java.awt.event.MouseEvent evt) {

```

int count=0;
if(jList1.getSelectedValue()==null)
{
    JOptionPane.showMessageDialog(this,"No Record Present.");
}

```

```

}

else
{
try
{
    Class.forName("java.sql.Driver");
    Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    Object s1=jList1.getSelectedValue();
    String query="select * from disposed;";
    ResultSet rs=stmt.executeQuery(query);
    int count1=0;
    while(rs.next())
    {
        if(s1.equals(rs.getString(2)))
        {
            count1=1;
            break;
        }
    }
    if(count1==1)
    {
        JOptionPane.showMessageDialog(this,"These records have been disposed.");
        int i=JOptionPane.showConfirmDialog(this,"Wish to
undispose?","Question",JOptionPane.YES_NO_OPTION);
        if(i==0)
        {
            Statement stmts=(Statement)con.createStatement();
            String query1="Delete from disposed where LibraryCard_number='"+s1+"'";
            stmts.executeUpdate(query1);
            JOptionPane.showMessageDialog(this,"Undisposed.");
        }
    }
}

```

```

else if(count1==0)
{
    String s="Select name_of_issuer,Phone_numbers,Date_Registration,address from
Registered_issuers where LibraryCard_Number='"+s1+"';

    ResultSet rs1=stmt.executeQuery(s);

    while(rs1.next())
    {
        jTextField1.setText(rs1.getString(1));
        jTextField2.setText(rs1.getString(2));
        jTextField3.setText(rs1.getString(3));
        jTextArea1.setText(rs1.getString(4));
    }

    String as="Select Record_Number from Transaction_records where
LibraryCard_Number='"+s1+"';

    ResultSet rs2=stmt.executeQuery(as);
    while(rs2.next())
    {
        count++;
    }

    int visits=count;
    if(count==1)
    {
        jTextField4.setText(Integer.toString(visits)+" visit");
    }
    else
    {
        jTextField4.setText(Integer.toString(visits)+" visits");
    }
}

stmt.close();
con.close();
}

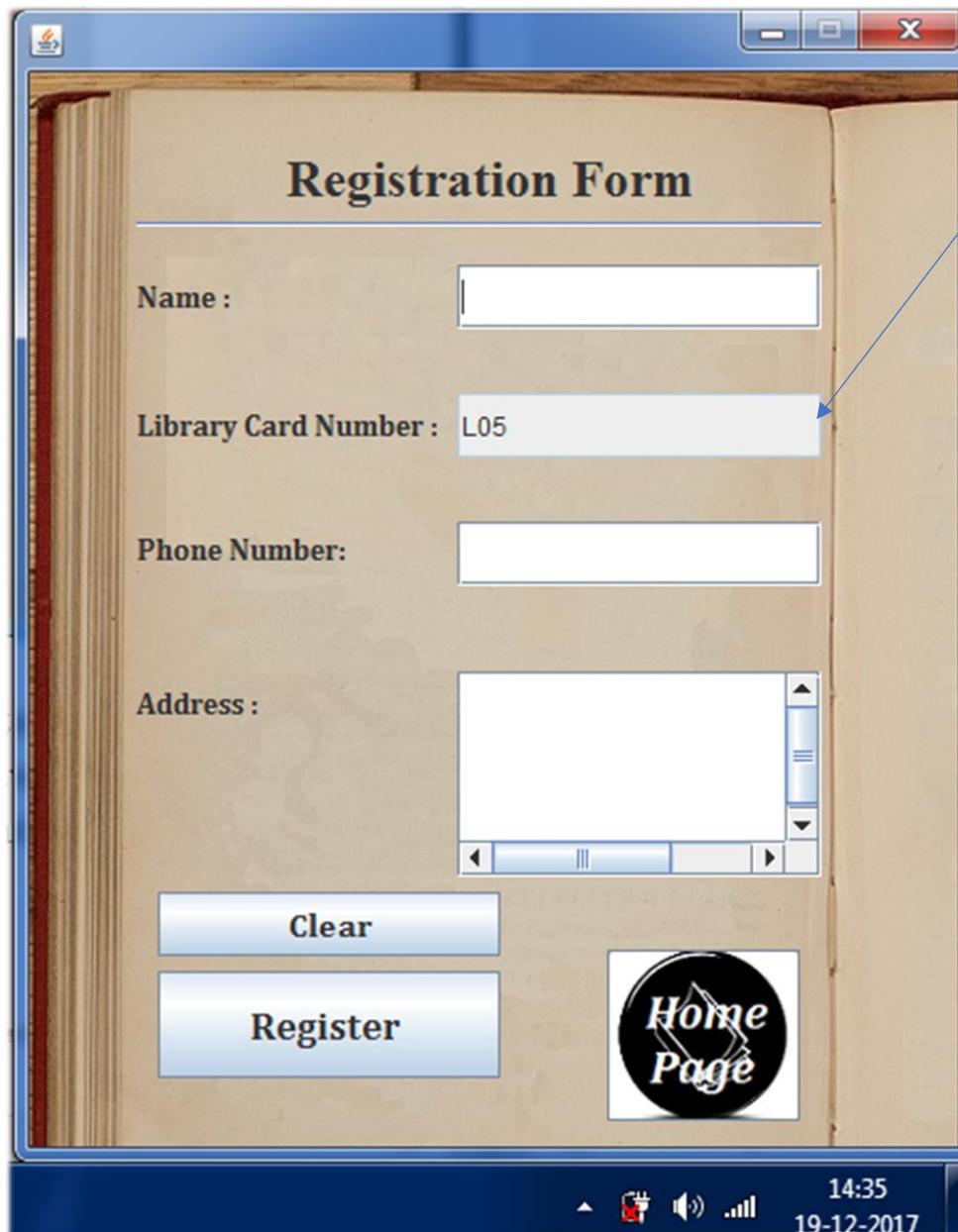
catch(Exception e)
{

```

```
if(jList1.getSelectedValue()==null)
{
    JOptionPane.showMessageDialog(this,"No Library Card Number Selected");
}
else
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}
```

If a person clicks on  button.

The following window opens it provides the user an opportunity to register a new member to the library.



Allotted by the software cannot be changed.

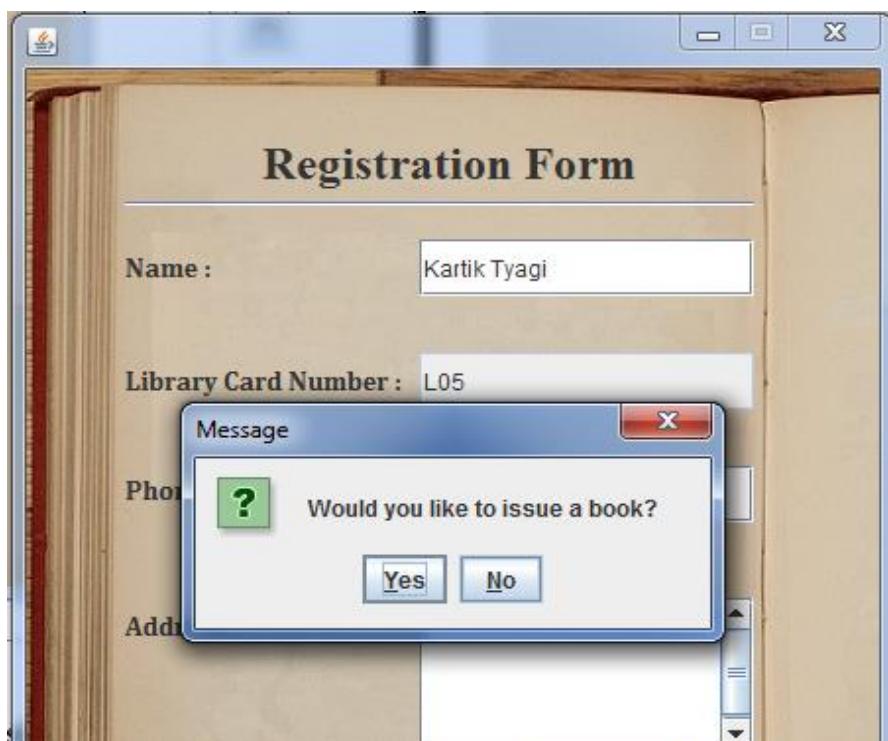
This window provides an opportunity for the user to register a new member to the library. The library card number is allotted by the software itself. A few details regarding the member are required for the registration. Also the date of registration is registered by the program itself.

Registering a new Visitor.

If there is any validation error with the data entered the program displays a suitable message.



The program asks if the newly registered visitor would like to issue a book.



The choice of the user leads him to the suitable form. If he wishes to issue a book the window leads him to the **Issue a book form**. Otherwise he is sent back to the front page.

Coding for Register A Visitor Form :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    Main_Page ob = new Main_Page();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextArea1.setText("");
}
```

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
```

```
    try
```

```
{
```

```
    Class.forName("java.sql.Driver");
```

```
    Connection con=(Connection)
```

```
    DriverManager.getConnection("jdbc:mysql://localhost/library","root","");

```

```
    Statement stmt=(Statement) con.createStatement();
```

```
    Statement stmt1=(Statement) con.createStatement();
```

```
    String s="select LibraryCard_Number from Registered_issuers";
```

```
    ResultSet rs=stmt1.executeQuery(s);
```

```
    ResultSet rs1=stmt.executeQuery(s);
```

```
    rs1.first();
```

```
    String k1=rs1.getString(1);
```

```
    while(rs.next())
```

```
{
```

```
    if(rs.last())
```

```
{
```

```
    String a=rs.getString(1);
```

```
    String a1=a.substring(1);
```

```

int k=Integer.parseInt(a1);
k=k+1;
if(k<10)
{
    jTextField2.setText("L0"+Integer.toString(k));
}
else
{
    jTextField2.setText("L"+Integer.toString(k));
}
stmt.close();
con.close();
}
catch(Exception e2)
{
    if(e2.getMessage().equals("Illegal operation on empty result set."))
    {
        jTextField2.setText("L01");
    }
    else
    {
        JOptionPane.showMessageDialog(null,e2.getMessage());
    }
}
}
}

```

```

private void jTextField1FocusLost(java.awt.event.FocusEvent evt) {

String s=jTextField1.getText();
int l=s.length();
if(s.isEmpty()==false)
{

```

```

for(int i=0;i<l;i++)
{
    char ch=s.charAt(i);
    if(Character.isLetter(ch)==false)
    {
        if(ch==' ')
        {
            continue;
        }
        JOptionPane.showMessageDialog(this,"Name must be in alphabets");
        jTextField1.setText("");
        break;
    }
}
}
}

```

```

private void jTextField3FocusLost(java.awt.event.FocusEvent evt) {
String s=jTextField3.getText();
int l=s.length();
if(l!=10)
{
    JOptionPane.showMessageDialog(this,"Incorrect Phone Number");
    jTextField3.setText("");
}
else
{
    if(s.isEmpty()==false)
    {
        for(int i=0;i<l;i++)
        {
            char ch=s.charAt(i);
            if(Character.isDigit(ch)==false)

```

```
{  
    JOptionPane.showMessageDialog(this,"Phone Number must be in digits");  
    jTextField3.setText("");  
    break;  
}  
}  
}  
}  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
String a1=jTextField1.getText();  
String b=jTextField2.getText();  
String c=jTextField3.getText();  
String d=jTextArea1.getText();  
if(a1.isEmpty()||c.isEmpty()||d.isEmpty())  
{  
    JOptionPane.showMessageDialog(this,"Information required");  
}  
else  
{  
    try  
{  
        Class.forName("java.sql.Driver");  
        Connection con=(Connection)  
DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement) con.createStatement();
        String s="insert into
Registered_issuers(Name_of_issuer,LibraryCard_Number,Phone_numbers,Address,Date_Registration)
values ('"+a1+"','"+b+"','"+c+"','"+d+"',curdate());";
        int p=stmt.executeUpdate(s);
        if(p==1)
        {
            JOptionPane.showMessageDialog(this,"Successful Registration");
        }
    }
}
```

```
int i=JOptionPane.showConfirmDialog(this,"Would you like to issue a  
book?","Message",JOptionPane.YES_NO_OPTION);  
  
if(i==0)  
{  
    Book_Issue ob= new Book_Issue();  
    ob.setVisible(true);  
    this.dispose();  
}  
  
else if(i==1)  
{  
    Main_Page ob= new Main_Page();  
    ob.setVisible(true);  
    this.dispose();  
}  
}  
  
stmt.close();  
con.close();  
}  
  
catch(Exception e)  
{  
    JOptionPane.showMessageDialog(this,e.getMessage());  
}  
}  
}
```

If a person clicks on **Issue/Return A Book** button.

The following window opens it provides the user an opportunity to issue/return a book for a member.

It is an important form of the program as it deals with the prime of making this project. It includes two options to **issue a new book** or to **return an issued book**.

The librarian has to deal with these two options quite often, therefore it has been necessary to make them easily available on the front page. A home button has been provided so that the user can shift back to the main form.



Coding for Transactions Form :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    Main_Page ob=new Main_Page();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

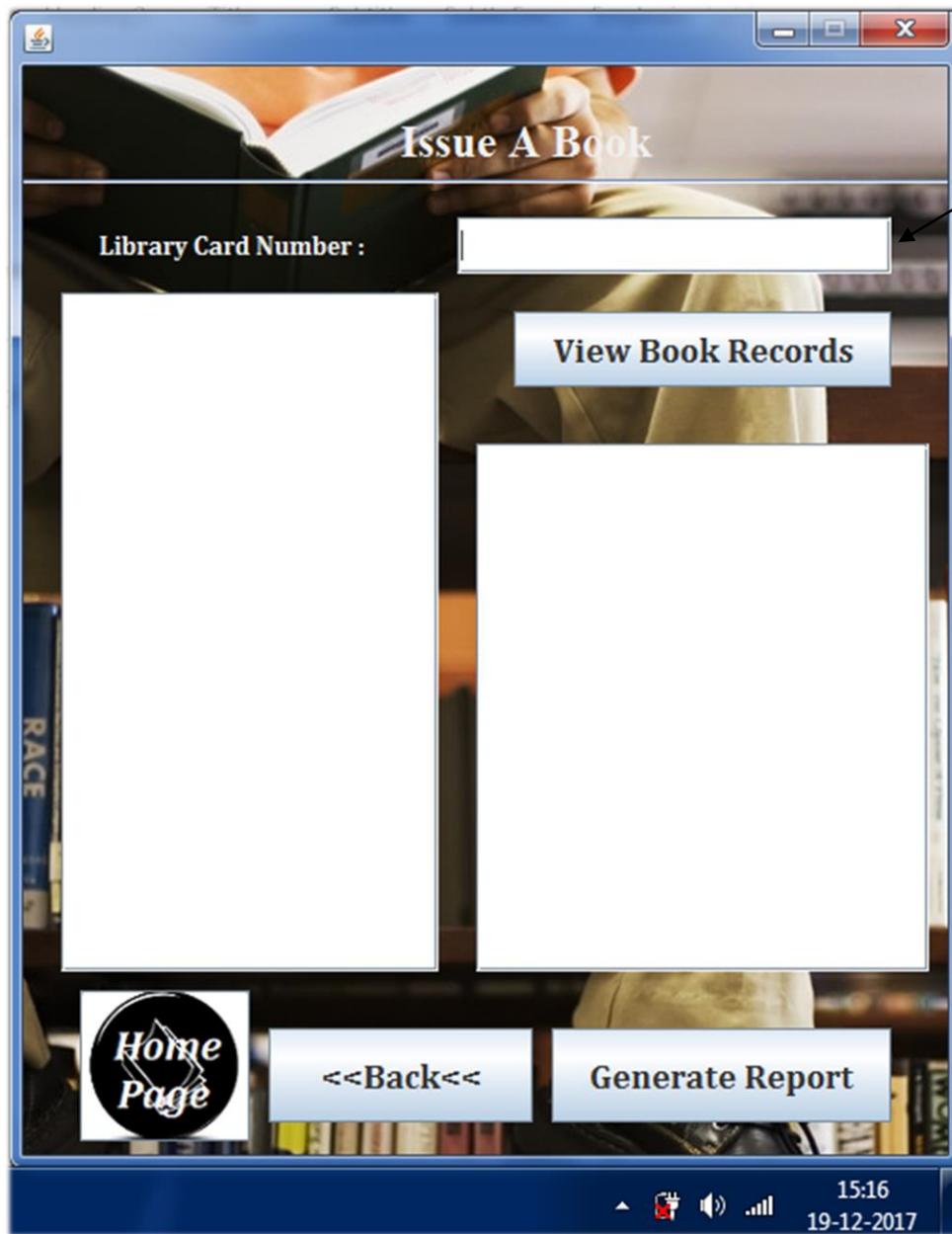
```
    Book_Issue ob=new Book_Issue();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    Return_Book ob=new Return_Book();
    ob.setVisible(true);
    this.dispose(); }
```

If a person clicks on **Issue A Book** button.

The following window opens up, it provides the user an opportunity to issue a book for a member.



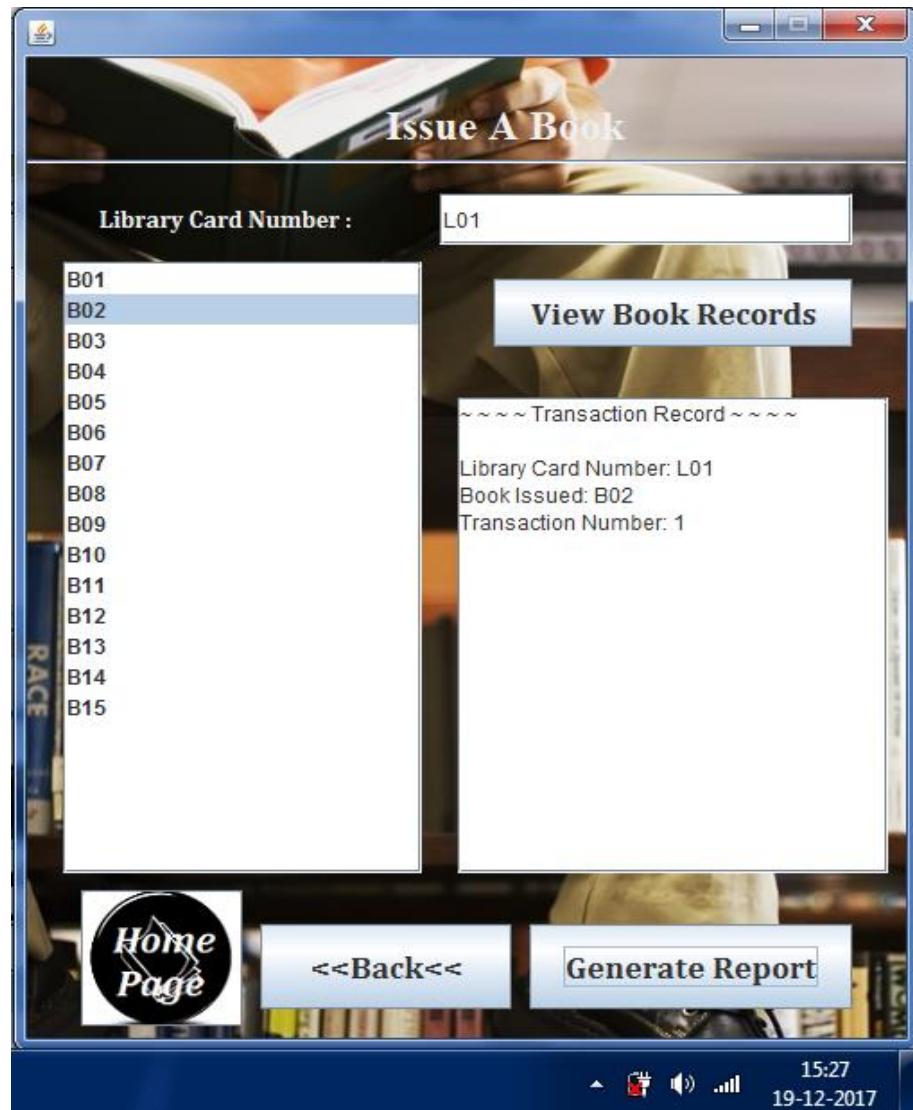
Allotted by the software
at the time of
registration.

Issuing a book is one of the basic characteristics of a library and hence this operation has an important role in this project. If a person wishes to issue a book he must have a library card number which is required by this form. He is given a transaction number which is required while returning a book. This form deals with the accession numbers of books which is to be issued by the member.

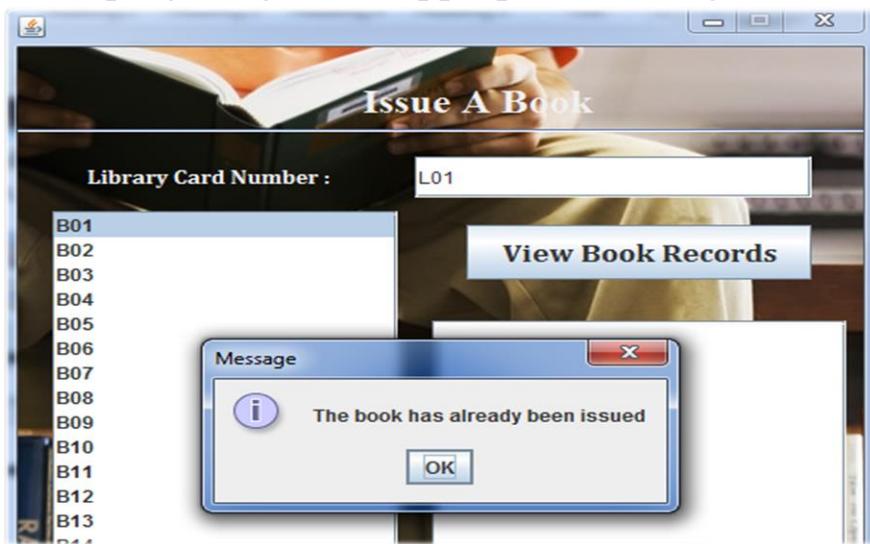
Once the Library card number is entered, click "View Book Records" button. This will result in a list of accession numbers in the list given on left hand side. The user can select the specific accession number given on book.

Once this is done the user must click on “ Generate Report ” button.

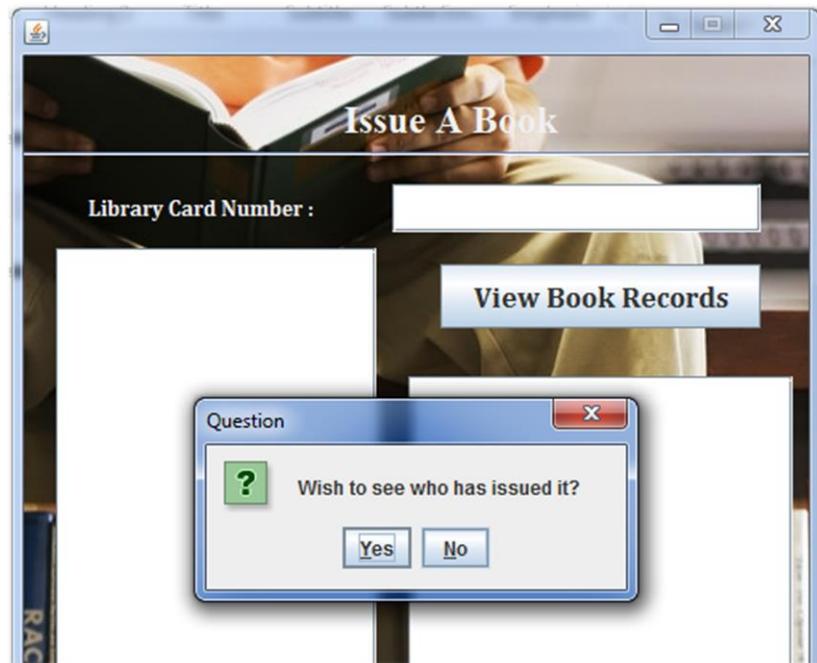
- This checks the issuer’s registration.
- Then completes transaction.
- And gives the transaction number.



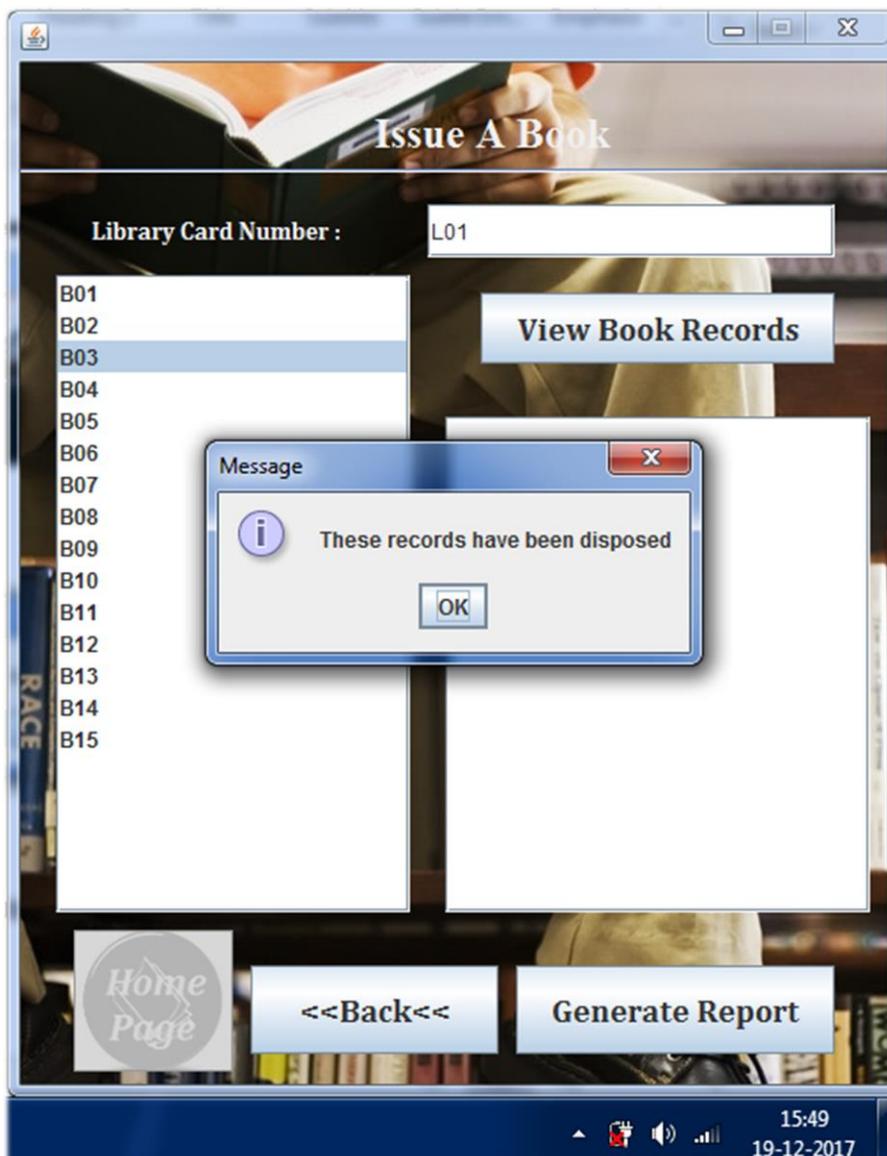
If by mistake the librarian selects the wrong book number which has already been issued then the program gives an appropriate message.



We even get a chance to see who has issued the book.



The user cannot issue a disposed book.



If the user wishes to issue a disposed book, he must first retain the information of the book.

Coding for Issue A Book Form :

```
public class Book_Issue extends javax.swing.JFrame {  
    String z;  
    public Book_Issue() {  
        z="0";  
        initComponents();  
    }  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    Main_Page ob=new Main_Page();  
    ob.setVisible(true);  
    this.dispose();  
}  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    DefaultListModel l=(DefaultListModel)(jList1.getModel());  
    l.clear();  
    jTextArea1.setText("");  
    try  
    {  
        Class.forName("java.sql.Driver");  
        Connection  
        con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement)con.createStatement();
        Statement stmt1=(Statement)con.createStatement();
        String query="select accession_number from book_information";
        ResultSet rs=stmt.executeQuery(query);
        ResultSet rs1=stmt1.executeQuery(query);
        rs1.first();
        String k=rs1.getString(1);
        while(rs.next())
        {
            String a=rs.getString(1);
            if(a.isEmpty())
            {  
}
```

```

        JOptionPane.showMessageDialog(this,"No records to display");
        break;
    }
    l.addElement(a);
}
stmt.close();
con.close();
}
catch(Exception e)
{
    if(e.getMessage().equals("Illegal operation on empty result set."))
    {
        JOptionPane.showMessageDialog(this,"No records to display");
    }
    else
    {
        JOptionPane.showMessageDialog(this,e.getMessage());
    }
}
}

```

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
int count=1;
if(jTextField1.getText().equals(""))
{
    JOptionPane.showMessageDialog(this,"Library Card Number Needed");
    int s=JOptionPane.showConfirmDialog(this,"Are you registered?","Confirming
Registration",JOptionPane.YES_NO_OPTION);
    if(s==1)
    {
        JOptionPane.showMessageDialog(this,"Opening Registration.....");
        register_new_visitor ob=new register_new_visitor();
        ob.setVisible(true);
}

```

```

        this.dispose();
    }
else
{
    DefaultListModel l=(DefaultListModel)(jList1.getModel());
    l.clear();
}
}

else if(jTextField1.getText().equals("mysql")==false)
{
    if(jList1.getSelectedValue()==null)
    {
        JOptionPane.showMessageDialog(this,"No Record Selected");
    }
else
{
    int count1=0;
    String Lib=jTextField1.getText();
    try
    {
        Class.forName("java.sql.Driver");
        Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement)con.createStatement();
        Statement stmtks=(Statement)con.createStatement();
        String s="select LibraryCard_Number from Registered_issuers";
        ResultSet rs=stmt.executeQuery(s);
        ResultSet rs12=stmtks.executeQuery(s);
        rs12.first();
        String kk=rs12.getString(1);
        while(rs.next())
        {
            String a1=rs.getString(1);

```

```

if(Lib.equals(a1))
{
    JOptionPane.showMessageDialog(this,"Registration Confirmed");
    count1=2;
    break;
}
else
{
    count1 =1;
    continue;
}
}

if(count1==1)
{
    JOptionPane.showMessageDialog(this,"Not Registered");
    jTextField1.setText("");
}

DefaultListModel m=(DefaultListModel)(jList1.getModel());
m.clear();

int sk=JOptionPane.showConfirmDialog(this,"Would you like to register?","Confirming
Registration",JOptionPane.YES_NO_OPTION);

if(sk==0)
{
    register_new_visitor ob=new register_new_visitor();
    ob.setVisible(true);
    this.dispose();
    this.setEnabled(false);
}
}

if(count1==2)
{
    String k=jTextField1.getText();
    Object k1=jList1.getSelectedValue();
    try{

```

```

Class.forName("java.sql.Driver");

Connection
con1=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt1=(Statement)con1.createStatement();

String skk="insert into Transaction_records(LibraryCard_Number,Accession_Number,Date_of_Issue)
values('"+k+"','"+k1+"',curdate());";

String s1="select Record_Number from transaction_Records";
int l=stmt.executeUpdate(skk);

if(l==1)

{
    JOptionPane.showMessageDialog(this,"Transaction Complete");
    count=0;
}

if(count==0)
{
    try{

Class.forName("java.sql.Driver");

Connection
conk=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmtk=(Statement)conk.createStatement();

String sql="update book_information set stock_available = 0 where Accession_number = '"+k1+"'";
stmtk.executeUpdate(sql);

}
catch(Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}

ResultSet rs1=stmt1.executeQuery(s1);
int ka=0;
int counter=0;
while(rs1.next())
{
    counter++;
}

```

```

}

ka=counter;

JOptionPane.showMessageDialog(this,"Transaction Number: "+ka);

jTextArea1.setText("");

jTextArea1.append("~~~~ Transaction Record ~~~~\n\n");

jTextArea1.append("Library Card Number: "+k+"\n");

jTextArea1.append("Book Issued: "+k1+"\n");

jTextArea1.append("Transaction Number: "+ka+"\n");

}

}

catch(Exception e)

{

    JOptionPane.showMessageDialog(this,e.getMessage());

}

}

}

catch(Exception e1)

{

if(e1.getMessage().equals("Illegal operation on empty result set."))

{

    JOptionPane.showMessageDialog(this,"Library Card not registered");

    jTextField1.setText("");

    DefaultListModel m=(DefaultListModel)(jList1.getModel());

    m.clear();

    int sk=JOptionPane.showConfirmDialog(this,"Would you like to register?","Confirming Registration",JOptionPane.YES_NO_OPTION);

    if(sk==0)

    {

        register_new_visitor ob=new register_new_visitor();

        ob.setVisible(true);

        this.dispose();

        this.setEnabled(false);

    }

}

```

```
        }
    else
    {
        JOptionPane.showMessageDialog(this, e1.getMessage());
    }
}
}
}
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
Transactions ob=new Transactions();
ob.setVisible(true);
this.dispose();
}

private void jList1MouseClicked(java.awt.event.MouseEvent evt) {
if(jList1.getSelectedValue()==null)
{
    JOptionPane.showMessageDialog(this,"No Record Present");
}
try
{
    Class.forName("java.sql.Driver");
    Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    Statement stmtk=(Statement)con.createStatement();
    Object a1=jList1.getSelectedValue();
    String sql="select stock_available from book_information where accession_number= '"+a1+"'";
    ResultSet rsk=stmtk.executeQuery(sql);
    rsk.first();
    String s1="select * from disposed;";
    System.out.println(s1);
}
```

```

ResultSet rs1=stmt.executeQuery(s1);
String a=rsk.getString(1);
if(a.equals("0"))
{
    JOptionPane.showMessageDialog(this,"The book has already been issued");
    jTextField1.setText("");
    DefaultListModel m=(DefaultListModel)(jList1.getModel());
    m.clear();
    int sk=JOptionPane.showConfirmDialog(this,"Wish to see who has issued
it?","Question",JOptionPane.YES_NO_OPTION);
    if(sk==0)
    {
        record_archives ob=new record_archives();
        ob.setVisible(true);
        this.dispose();
        this.setEnabled(false);
    }
    else
    {
        jButton2.setEnabled(false);
    }
}
int count=0;
while(rs1.next())
{if(a1.equals(rs1.getString(1)))
{
    count=1;
    break;
}
}
if(count==1)
{
    JOptionPane.showMessageDialog(this,"These records have been disposed");
}

```

```

jButton2.setEnabled(false);
}
else
{
    jButton2.setEnabled(true);
}
stmt.close();
con.close();
}
catch(Exception e)
{
if(jList1.getSelectedValue()==null)
{
    JOptionPane.showMessageDialog(this,"No book Selected");
}
else
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}
}
}

```

```

private void jTextField1FocusLost(java.awt.event.FocusEvent evt) {

String s=jTextField1.getText();
try
{
    Class.forName("java.sql.Driver");
    Connection
    con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    String s1="select * from disposed;";
    ResultSet rs1=stmt.executeQuery(s1);
    int count=0;

```

```

while(rs1.next())
{if(s.equals(rs1.getString(2)))
{
    count=1;
    break;
}
}
if(count==1)
{
    JOptionPane.showMessageDialog(this,"The Issuer records have been disposed");
    jTextField1.setText("");
    DefaultListModel m=(DefaultListModel)(jList1.getModel());
    m.clear();
}
stmt.close();
con.close();
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}

```

```

private void jTextField1FocusGained(java.awt.event.FocusEvent evt) {

DefaultListModel m=(DefaultListModel)(jList1.getModel());
m.clear();
jTextArea1.setText("");
}

```

```

private void formWindowOpened(java.awt.event.WindowEvent evt) {

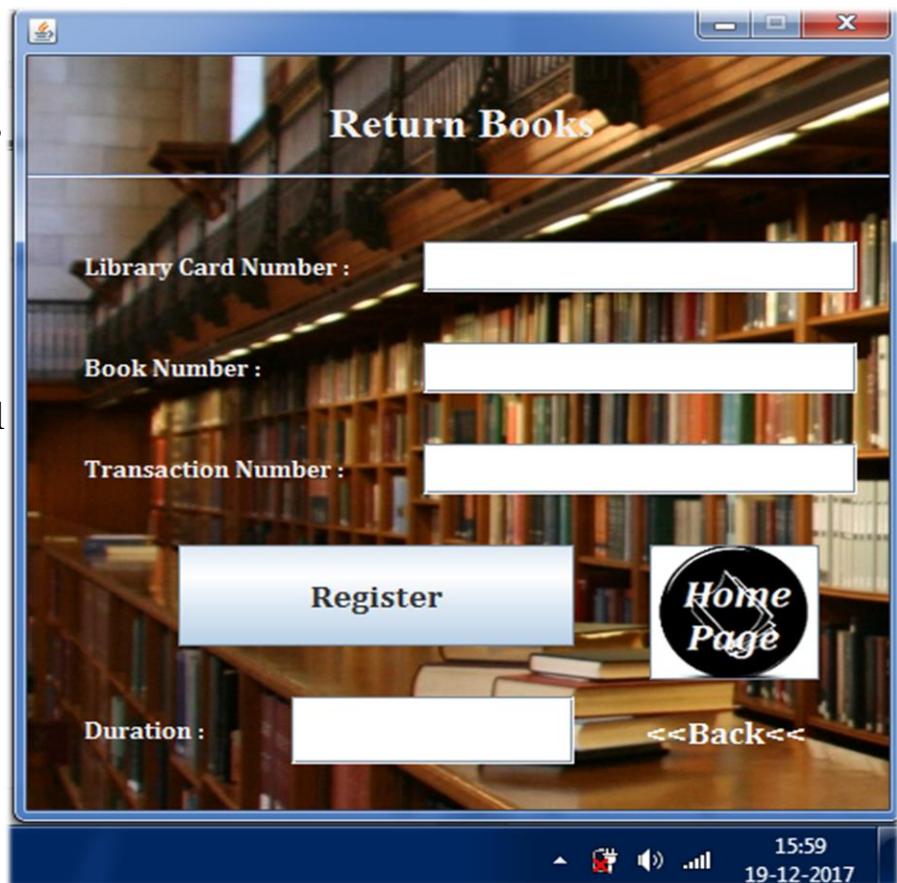
jTextField1.setText(z);
}

```

If a person clicks on **Return A Book** button.

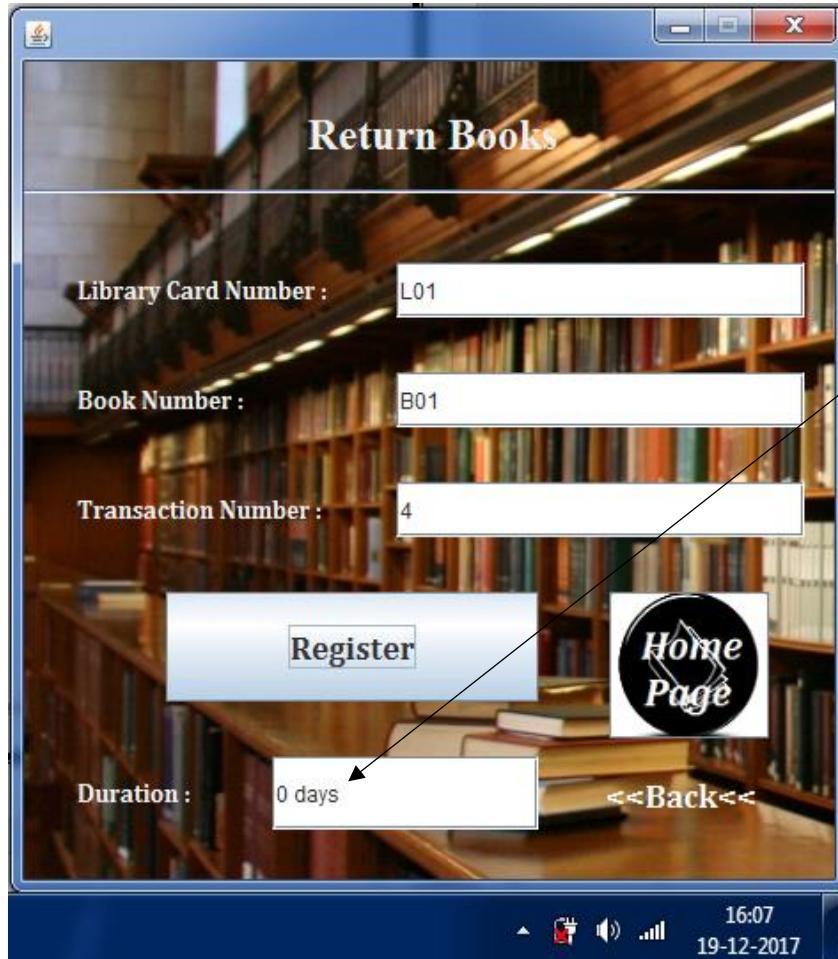
The following window opens up, it provides the user an opportunity to return a book for a member.

Returning a book is as important as issuing a book and hence this operation has an important role in this project. If a person wishes to return a book he must have library card number, accession number and transaction number required by this form. This form also displays the duration of the issue.



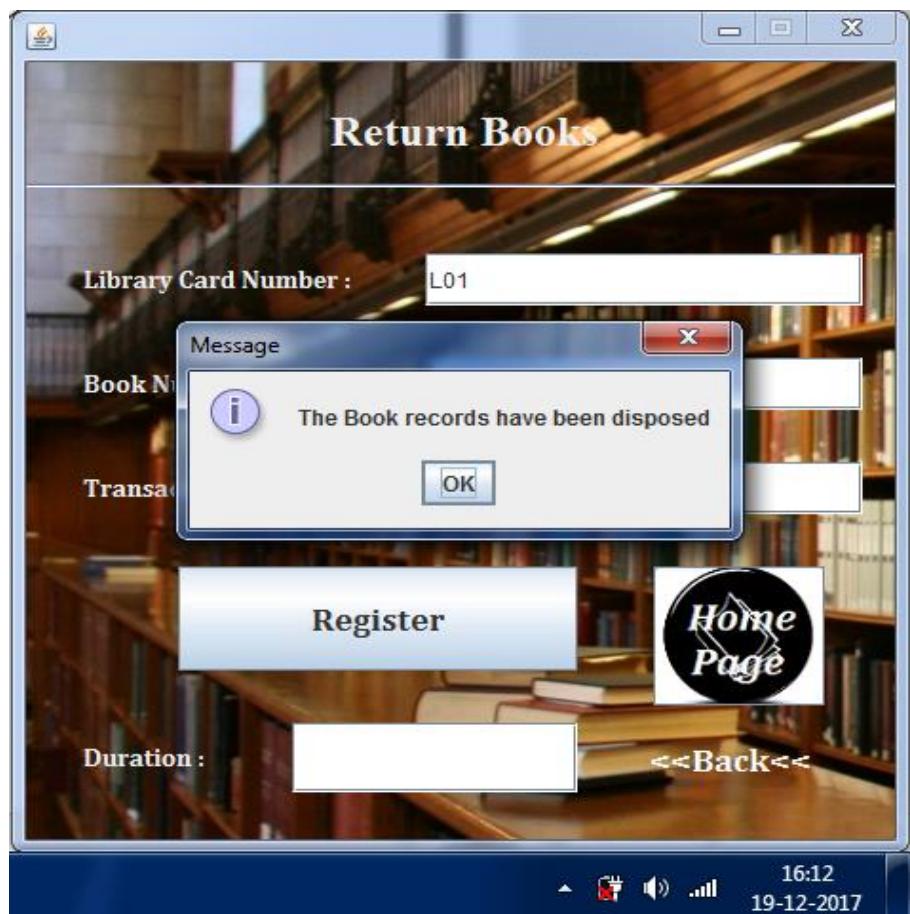
If a person wishes to return a book, the librarian must enter his Library card number, the accession number of the book being returned and the transaction number.





The book is returned on the same day.

The user cannot return a disposed book.



Coding for Return A Book Form :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
Main_Page ob=new Main_Page();  
ob.setVisible(true);  
this.dispose();  
}  
  
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
Transactions ob=new Transactions();  
ob.setVisible(true);  
this.dispose();  
}  
  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
String tno=jTextField3.getText();  
String bookno=jTextField2.getText();  
int c=0;  
if(tno.isEmpty())  
{  
JOptionPane.showMessageDialog(this,"Please enter a record number.");  
}  
else if(bookno.isEmpty())  
{  
JOptionPane.showMessageDialog(this,"Please enter the book number.");  
}  
else if(tno.isEmpty()==false)  
{  
try  
{  
Class.forName("java.sql.Driver");  
Connection  
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt=(Statement)con.createStatement();
```

```

String query="Select date_of_return from transaction_records where record_number='"+tno+"'";
ResultSet rs=stmt.executeQuery(query);
String dor="";
rs.first();
dor=rs.getString(1);
if(dor.isEmpty()==false)
{
    JOptionPane.showMessageDialog(this,"This transaction has already been made.");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField1.setText("");
    jTextField4.setText("");
    c=0;
}
}
catch(Exception e)
{
    c=1;
}
}
if(c==1)
{
    try
    {
        Class.forName("java.sql.Driver");
        Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement)con.createStatement();
        Statement stmt1=(Statement)con.createStatement();
        Statement stmt2=(Statement)con.createStatement();
        String query1="Select accession_number from transaction_records where
record_number='"+tno+"'";
        String query2="Update book_information set Stock_available = 1 where accession_number =
"+bookno+";";

```

```

String query3="Update Transaction_Records set Date_of_return = curdate() where
Record_Number='"+tno+"';

ResultSet rs1=stmt2.executeQuery(query1);

rs1.first();

int count1=0;

String acc=rs1.getString(1);

if(bookno.equals(acc))

{

    count1=1;

}

else

{

    JOptionPane.showMessageDialog(this,"Incorrect Book Number.");

    jTextField2.setText("");

    jTextField3.setText("");

    jTextField1.setText("");

    jTextField4.setText("");

    count1=0;

}

if(count1==1)

{

    int i=stmt.executeUpdate(query3);

    int count=0;

    if(i==1)

    {

        JOptionPane.showMessageDialog(this,"Registration Complete.");

        count=1;

    }

    else

    {

        JOptionPane.showMessageDialog(this,"Registration Failed.");

        count=0;

    }

}

```

```

if(count==1)
{
    stmt1.executeUpdate(query2);
}
else
{
    JOptionPane.showMessageDialog(this,"No such record exists");
    jTextField3.setText("");
    jTextField2.setText("");
    jTextField1.setText("");
    jTextField4.setText("");
}

String query4="select DateDiff(Date_of_return, Date_of_issue) from Transaction_Records where
Record_Number='"+tno+"'";
ResultSet rs=stmt.executeQuery(query4);
rs.first();
String a=rs.getString(1);
String a1=a+" days";
jTextField4.setText(a+" days");

String sa="update Transaction_Records set Duration_of_issue='"+a1+"' where
Record_Number='"+tno+"'";
stmt.executeUpdate(sa);

String f;
if(Integer.parseInt(a)>15)
{
    double fine;
    fine=5*(Integer.parseInt(a)-15);
    f=Double.toString(fine)+" Rs";
}
else
{
    f="0 Rs";
}

```

```
String sm="update Transaction_records set Fine_Charged='"+f+"' where  
Record_Number='"+tno+"';  
  
stmt.executeUpdate(sm);  
  
stmt.close();  
  
con.close();  
  
}  
  
}  
  
catch(Exception e)  
{  
  
String a=e.getMessage();  
  
if(a.equals("Illegal operation on empty result set."))  
{  
  
JOptionPane.showMessageDialog(this,"No such record exists.");  
  
jTextField3.setText("");  
  
jTextField2.setText("");  
  
jTextField1.setText("");  
  
jTextField4.setText("");  
  
}  
  
else  
{  
  
JOptionPane.showMessageDialog(this,e.getMessage());  
  
}  
  
}  
  
}
```

```
private void jTextField1FocusLost(java.awt.event.FocusEvent evt) {  
String s=jTextField1.getText();  
try  
{  
    Class.forName("java.sql.Driver");  
    Connection  
    con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
}
```

```

Statement stmt=(Statement)con.createStatement();
String s1="select * from disposed;";
ResultSet rs1=stmt.executeQuery(s1);
int count=0;
while(rs1.next())
{
if(s.equals(rs1.getString(2)))
{
count=1;
break;
}
}
if(count==1)
{
JOptionPane.showMessageDialog(this,"The Issuer records have been disposed");
jTextField1.setText("");
}
stmt.close();
con.close();
}

catch(Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
}

private void jTextField2FocusLost(java.awt.event.FocusEvent evt) {

String s=jTextField2.getText();
try
{
Class.forName("java.sql.Driver");
Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
}

```

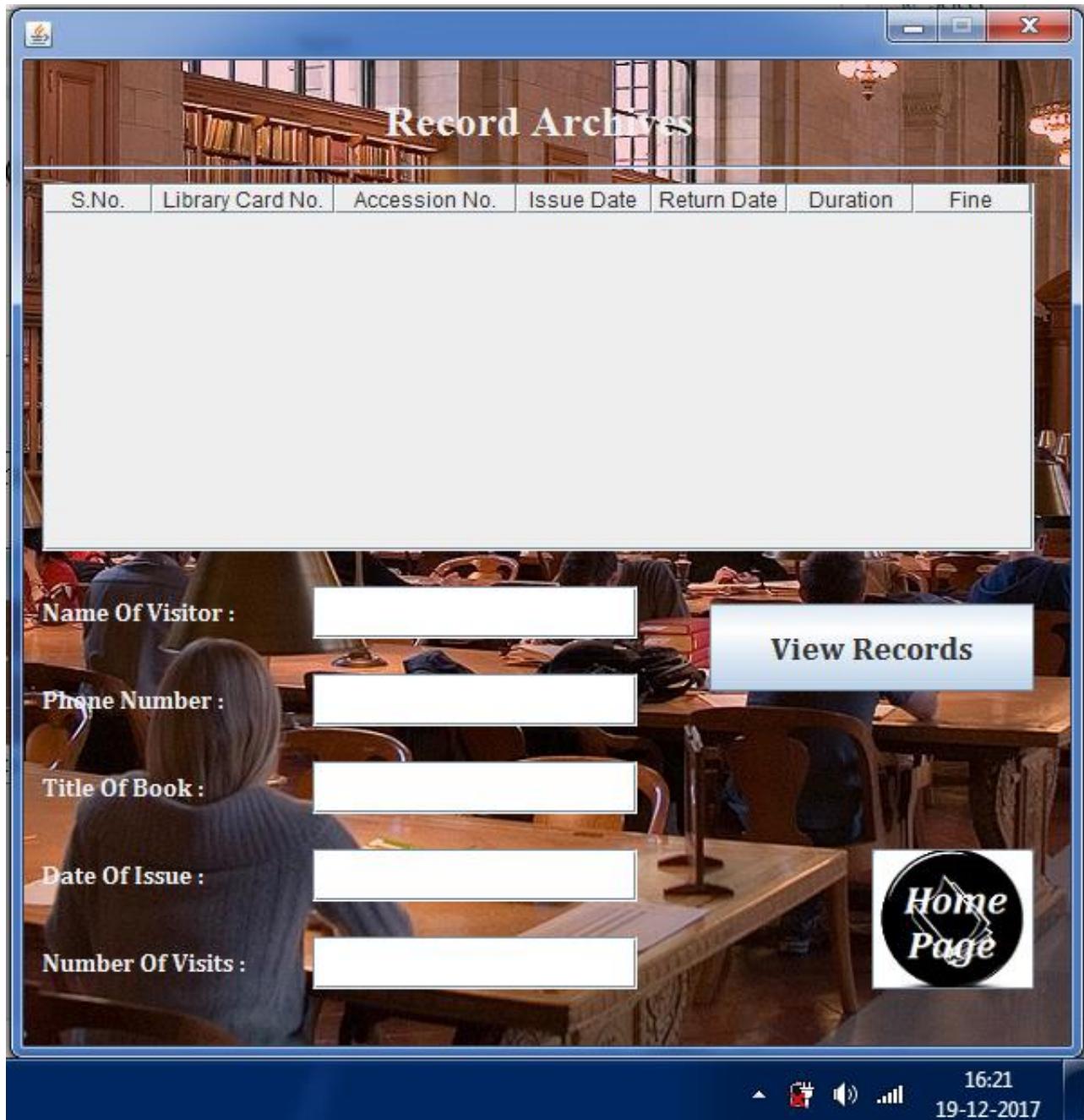
```

Statement stmt=(Statement)con.createStatement();
String s1="select * from disposed;";
ResultSet rs1=stmt.executeQuery(s1);
int count=0;
while(rs1.next())
{
    if(s.equals(rs1.getString(1)))
    {
        count=1;
        break;
    }
}
if(count==1)
{
    JOptionPane.showMessageDialog(this,"The Book records have been disposed");
    jTextField2.setText("");
    jTextField1.setText("");
}
stmt.close();
con.close();
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}

```

If a person clicks on **Record Archives** button.

The following window opens it provides the user an opportunity to see the transactions till date.



The user can also view specific details of a particular transaction by clicking on the row in the table.

Viewing transactions.

The form displays the above details for each specific record which is selected in the table.

S.No.	Library Card No.	Accession No.	Issue Date	Return Date	Duration	Fine
1	L01	B02	2017-12-19			
2	L01	B01	2017-12-19			
3	L01	B01	2017-12-19			
4	L02	B01	2017-12-19	2017-12-19	0 days	0 Rs
5	L01	B04	2017-12-19			

Name Of Visitor : Kartik Tyagi

Phone Number : 9971480190

Title Of Book : Concepts of Physics Volume 1

Date Of Issue : 2017-12-19

Number Of Visits : 4 visits

View Records

Home Page

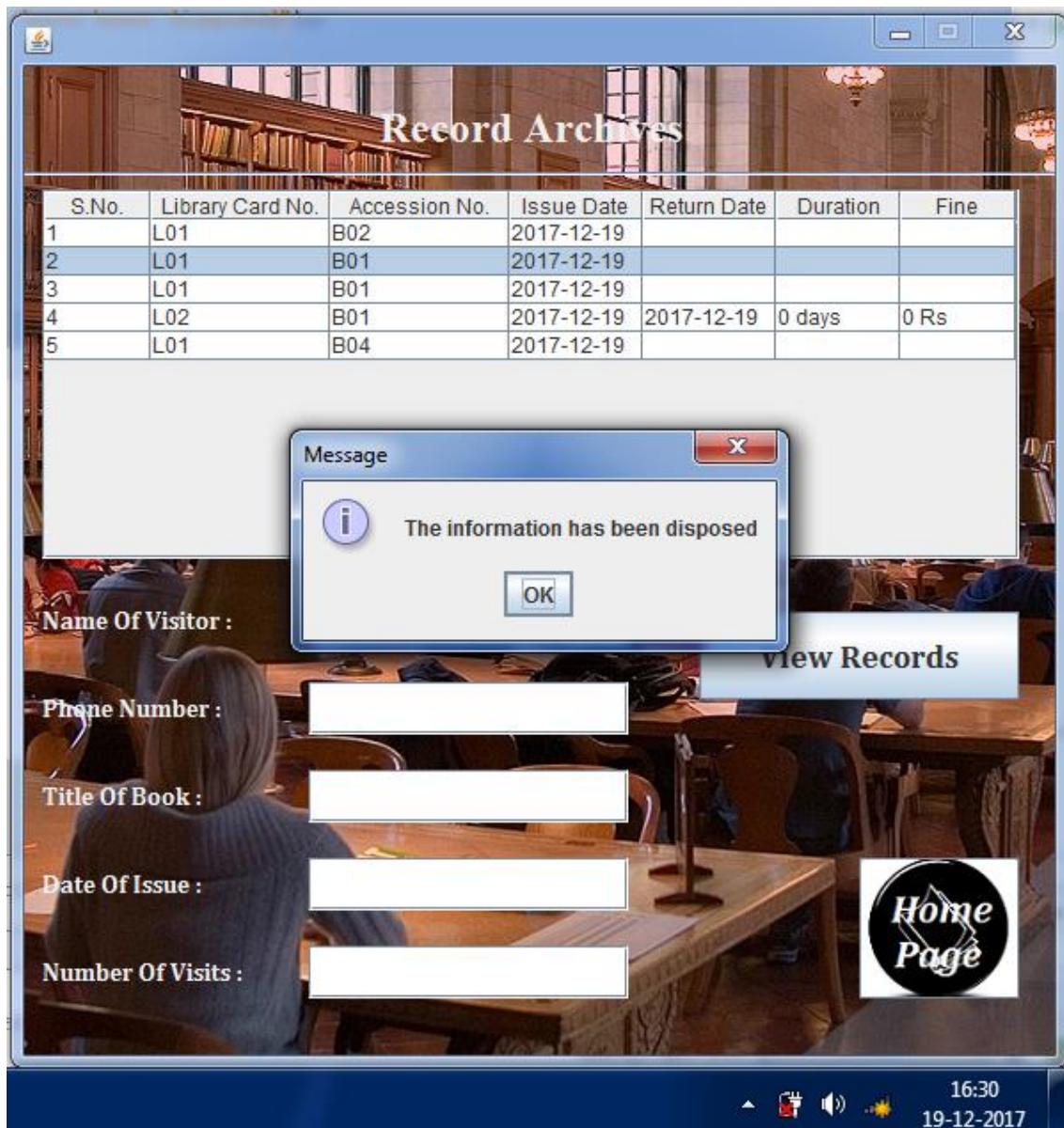
16:23 19-12-2017

The user cannot view transactions of disposed book or disposed member.

Here number of visits is 5 it means that the person has visited the library 5 times since his registration.

The user cannot view the details of the transaction until he retains the data of the disposed quantity.





Coding for Record Archives Form :

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    Main_Page ob = new Main_Page();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
```

```

jTextField5.setText("");

DefaultTableModel t = (DefaultTableModel)jTable1.getModel();

int r=t.getRowCount();

if(r>=0)

{

for(int i=0;i<r;i++)

{



t.removeRow(0);

}

}

try

{



Class.forName("java.sql.Driver");

Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library",
"root","");

Statement stmt=(Statement)con.createStatement();

String query="Select Record_Number,LibraryCard_Number,Accession_Number,
Date_of_issue,Date_of_return,Duration_of_issue,Fine_Charged from Transaction_Records";

ResultSet rs = stmt.executeQuery(query);

while (rs.next())

{



String a11, b, c, d, e, f, g;

a11=rs.getString(1);

b=rs.getString(2);

c=rs.getString(3);

d=rs.getString(4);

e=rs.getString(5);

f=rs.getString(6);

g=rs.getString(7);

t.addRow(new Object[]{a11, b, c, d, e, f, g});

}

if (t.getRowCount()==0)

{



JOptionPane.showMessageDialog(this, "No records to display.");

```

```

        }

        stmt.close();
        con.close();
    }

    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

```

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

    int s=jTable1.getSelectedRow();
    s=s+1;

    int count=0,count1=0;

    try
    {
        Class.forName("java.sql.Driver");

        Connection con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library",
        "root","");
        Statement stmt=(Statement)con.createStatement();
        Statement stmt1=(Statement)con.createStatement();

        String sa="select * from disposed";

        String s1="select LibraryCard_Number,Accession_number,Date_of_issue from Transaction_records
        where Record_number=" + s + ";" ;

        ResultSet rs=stmt.executeQuery(s1);

        rs.first();

        ResultSet rs1=stmt1.executeQuery(sa);

        String l, ac;

        String x, y, z;

        x=rs.getString(1);
        y=rs.getString(2);

        while(rs1.next())
        {
            l=rs1.getString(2);

```

```

ac=rs1.getString(1);

if(x.equals(l))
{
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");

    JOptionPane.showMessageDialog(this, "The information has been disposed");

    count1 = 1;
    break;
}

if(y.equals(ac))
{
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");

    JOptionPane.showMessageDialog(this, "The information has been disposed");

    count1 = 1;
    break;
}

}

if(count1 == 0)
{
    Statement stmt2 = (Statement) con.createStatement();

    String s2 = "select LibraryCard_Number,Accession_number,Date_of_issue from
Transaction_records where Record_number='" + s + "'";

    ResultSet rs3 = stmt2.executeQuery(s2);

    rs3.first();

    String x1=rs3.getString(1);

    String y1=rs3.getString(2);
}

```

```

z=rs3.getString(3);

String S="select Name_of_issuer,Phone_numbers,Title from
Registered_issuers,Book_information"
+ " where LibraryCard_Number = '" + x1 + "' && Accession_Number = '" + y1 + "' ;";

ResultSet Rs=stmt2.executeQuery(S);

String a,b,c;

Rs.first();

a=Rs.getString(1);

b=Rs.getString(2);

c=Rs.getString(3);

jTextField1.setText(a);

jTextField2.setText(b);

jTextField3.setText(c);

jTextField4.setText(z);

String ak="select Record_Number from transaction_records where LibraryCard_Number='"
+x + "' ;"

ResultSet r1=stmt1.executeQuery(ak);

while(r1.next())
{
    count++;

}

int k=count;

if (k==1)
{
    jTextField5.setText(Integer.toString(k) + " visit");

}
else
{
    jTextField5.setText(Integer.toString(k) + " visits");
}

}

stmt.close();
con.close();

```

```
}

catch(Exception e)

{

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

jTextField4.setText("");

jTextField5.setText("");

String sk;

sk=e.getMessage();

if(sk.equals("Illegal operation on empty result set."))

{

JOptionPane.showMessageDialog(this, "Book information has been disposed \n"

+ "Or \n"

+ "Issuer Information has been disposed");

}

else

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

}

}
```

If a person clicks on **Update Databases** button.

The following window opens up, it provides the user an opportunity to issue a book for a member.

Update database form is one of the most important forms of the project. It helps a user to manipulate data at the back end. It consists of three buttons, namely:

- Update Visitor records
- Update Book records
- Reset Database

Reset Database is a powerful tool which can delete all records from the back-end therefore it must be handled with care.



Coding for Update Databases Form :

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    Main_Page ob=new Main_Page();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    update_book_records ob=new update_book_records();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    update_visitors_record ob=new update_visitors_record();
```

```

        ob.setVisible(true);

        this.dispose();
    }

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    try
    {
        String s=JOptionPane.showInputDialog(this,"Enter Administrator code :");
        if(s.equals("KT149209"))
        {
            int i=JOptionPane.showConfirmDialog(this,"Do you wish to delete all records?","Reset
Data",JOptionPane.YES_NO_OPTION);
            if(i==0)
            {
                try{
                    Class.forName("java.sql.Driver");
                    Connection con=(Connection)
                    DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
                    Statement stmt=(Statement) con.createStatement();
                    String s1="delete from transaction_records;";
                    String s2="alter table transaction_records auto_increment=1;";
                    String s3="delete from book_information";
                    String s4="delete from disposed";
                    String s5="delete from registered_issuers";
                    stmt.executeUpdate(s1);
                    stmt.executeUpdate(s2);
                    stmt.executeUpdate(s3);
                    stmt.executeUpdate(s4);
                    stmt.executeUpdate(s5);
                    JOptionPane.showMessageDialog(this,"All data has been safely deleted");
                    stmt.close();
                    con.close();
                }
            }
        catch(Exception e)
        {
    }
}

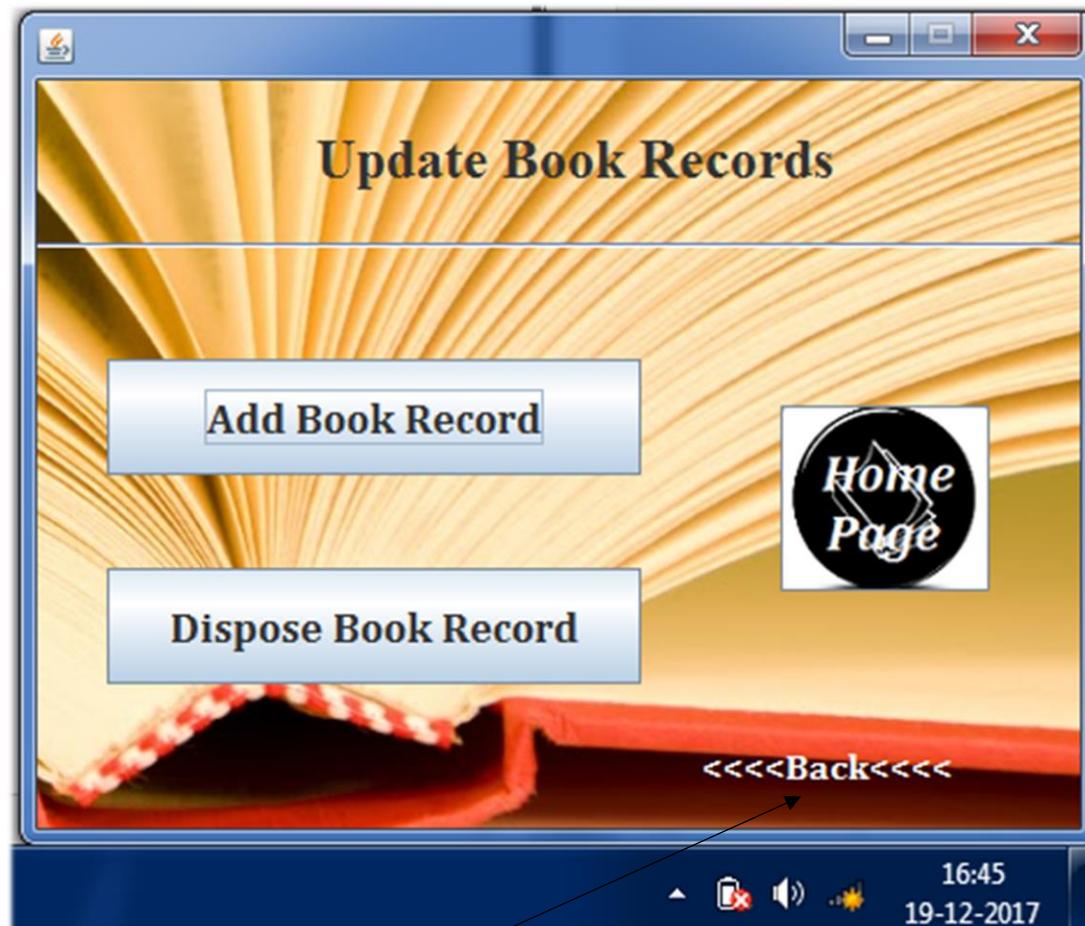
```

```
JOptionPane.showMessageDialog(this,e.getMessage());  
    }  
}  
else  
{  
    database_updation ob=new database_updation();  
    ob.setVisible(true);  
    this.dispose();  
}  
}  
else  
{  
    JOptionPane.showMessageDialog(this,"Incorrect Administrator code");  
}  
}  
}  
catch(Exception e)  
{  
    database_updation ob=new database_updation();  
    ob.setVisible(true);  
    this.dispose();  
}  
}
```

If a person clicks on
opens.

Update Book Records

button, the following window



Button to go back one page.

Here we have not included the coding of the Reset Database button, it is explained independently.

This form contains two options :

- Add a new book.
- Dispose a book.

Coding for Update Book Records Form :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt){
```

```
    Main_Page ob=new Main_Page();
    ob.setVisible(true);
    this.dispose();
}
```



```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    new_book_addition ob=new new_book_addition();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose_book_record ob=new dispose_book_record();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    database_updation ob=new database_updation();
    ob.setVisible(true);
    this.dispose();
}
```

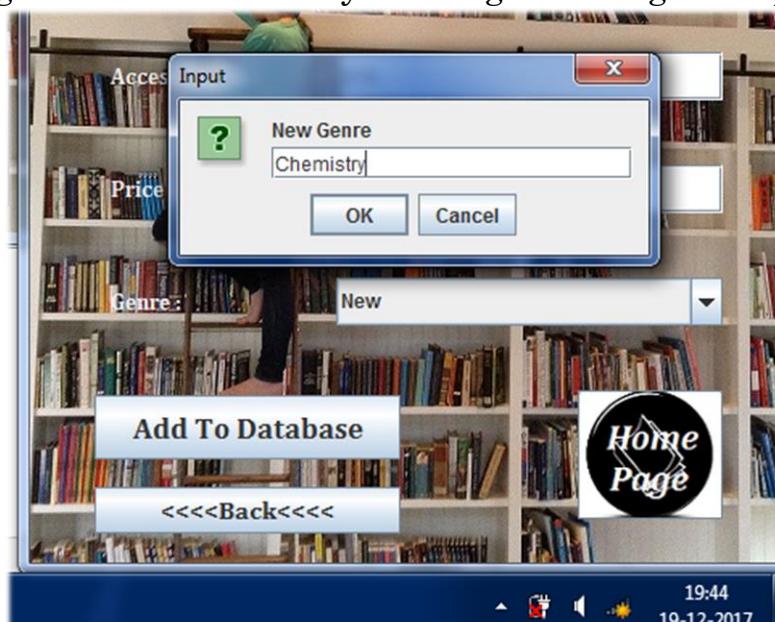
If a person goes with **Add Book Record** option, the following window

opens where the user can register a new book.



Allotted by software
and cannot be changed.

Genre is selected from the set of already registered genres, if the person wishes to add a new genre he can do so by clicking on new genre option.



Coding for Add A Book Form :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
String a1=jTextField1.getText();  
String b=jTextField2.getText();  
String c=jTextField3.getText();  
String d=jTextField4.getText();  
Object a=jComboBox1.getSelectedItem();  
  
if(a1.isEmpty()||b.isEmpty()||c.isEmpty()||d.isEmpty()||a.equals(""))  
{  
    JOptionPane.showMessageDialog(this,"Information Required");  
}  
else  
{  
int k=1;  
try  
{  
    Class.forName("java.sql.Driver");  
    Connection con=(Connection)  
    DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement) con.createStatement();  
  
    String s="insert into Book_information(Title,Author,Accession_number,Price,Genre,stock_available)  
values ('"+a1+"','"+b+"','"+c+"','"+d+"','"+a+"','"+k+"');";  
  
    int p=stmt.executeUpdate(s);  
  
    if(p==1)  
    {  
        JOptionPane.showMessageDialog(this,"Data Updated");  
    }  
    stmt.close();  
    con.close();  
}  
catch(Exception e)  
{  
    JOptionPane.showMessageDialog(this,e.getMessage());  
}
```

```

}

update_book_records ob=new update_book_records();
ob.setVisible(true);
this.dispose();
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
Main_Page ob=new Main_Page();
ob.setVisible(true);
this.dispose();
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
update_book_records ob=new update_book_records();
ob.setVisible(true);
this.dispose();
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
try
{
Class.forName("java.sql.Driver");
Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt=(Statement)con.createStatement();
Statement stmt1=(Statement)con.createStatement();
String s="select Accession_Number from Book_information";
ResultSet rs1=stmt1.executeQuery(s);
ResultSet rs=stmt.executeQuery(s);
rs1.first();
String k1=rs1.getString(1);
while(rs.next())
{

```

```

if(rs.last())
{
    String a=rs.getString(1);
    String a1=a.substring(1);
    int k=Integer.parseInt(a1);
    k=k+1;
    if(k>10)
    {
        jTextField3.setText("B"+Integer.toString(k));
    }
    else
    {
        jTextField3.setText("B0"+Integer.toString(k));
    }
}
}

rs.close();
stmt.close();
con.close();
}

catch(Exception e2)
{
    if(e2.getMessage().equals("Illegal operation on empty result set."))
    {
        jTextField3.setText("B01");
    }
    else
    {
        JOptionPane.showMessageDialog(null,e2.getMessage());
    }
}
}

DefaultComboBoxModel m1=(DefaultComboBoxModel)(jComboBox1.getModel());
try

```

```

{
    Class.forName("java.sql.Driver");
    Connection
    con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    String s="select distinct genre from book_information;";
    ResultSet rs=stmt.executeQuery(s);
    while(rs.next())
    {
        String a=rs.getString(1);
        m1.addElement(a);
    }
    rs.close();
    stmt.close();
    con.close();
    m1.addElement("New");
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}

```

```

private void jComboBox1ActionPerformed(java.awt.event.ActionEvent evt) {
Object s=jComboBox1.getSelectedItem();
if(s.equals("New"))
{
    Object s1=JOptionPane.showInputDialog(this,"New Genre");
    jComboBox1.addItem(s1);
    jComboBox1.setSelectedItem(s1);
}
}

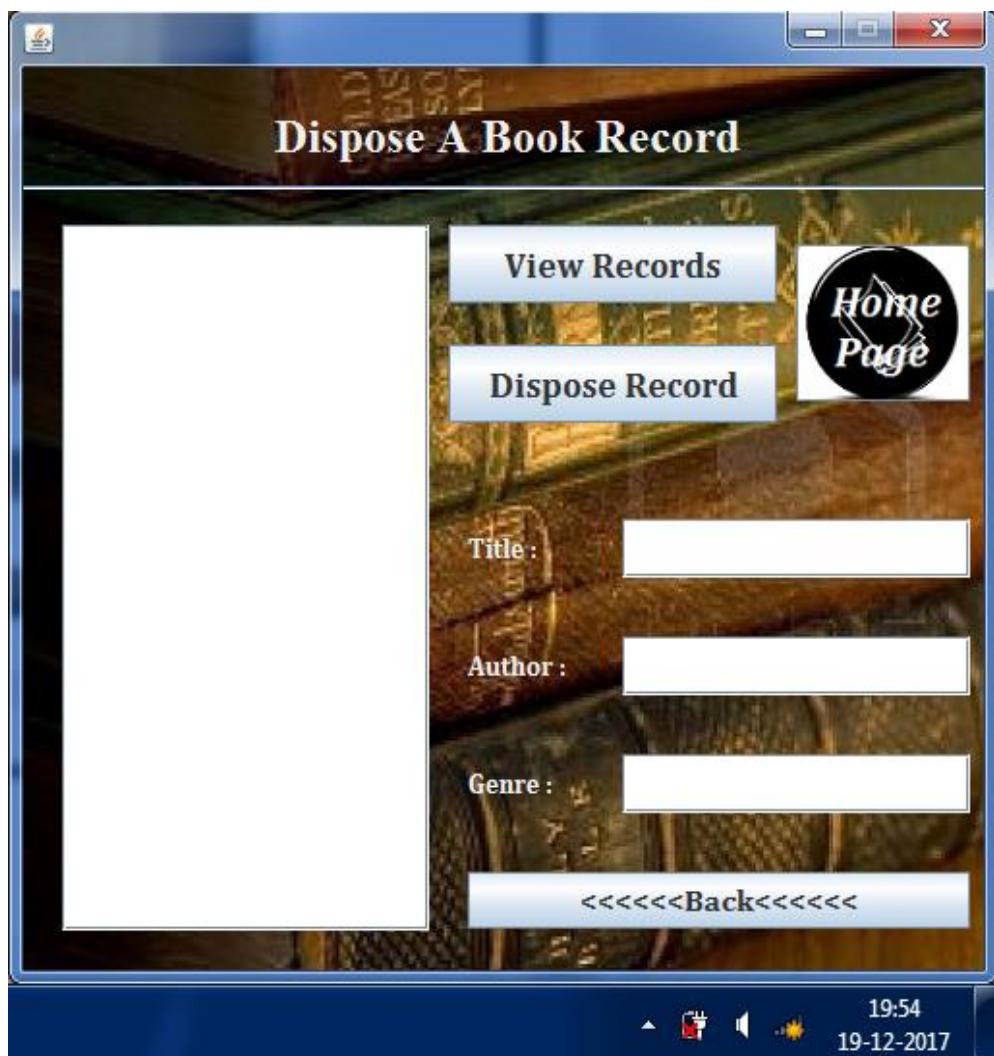
```

```
private void jTextField1FocusLost(java.awt.event.FocusEvent evt) {  
String s=jTextField1.getText();  
int l=s.length();  
if(s.isEmpty()==false)  
{  
for(int i=0;i<l;i++)  
{  
char ch=s.charAt(i);  
if(Character.isLetter(ch)==false)  
{  
if(ch==' ')  
{  
continue;  
}  
JOptionPane.showMessageDialog(this,"Title must be in alphabets");  
jTextField1.setText("");  
break;  
}  
}  
}  
}  
}  
}
```

```
private void jTextField2FocusLost(java.awt.event.FocusEvent evt) {  
String s=jTextField2.getText();  
int l=s.length();  
if(s.isEmpty()==false)  
{  
for(int i=0;i<l;i++)  
{  
char ch=s.charAt(i);  
if(Character.isLetter(ch)==false)  
{  
if(ch==' ')  
}{  
}  
}  
}  
}
```

```
{  
    continue;  
}  
JOptionPane.showMessageDialog(this,"Author name must be in alphabets");  
jTextField2.setText("");  
break;  
}  
}  
}  
}  
  
private void jTextField4FocusLost(java.awt.event.FocusEvent evt) {  
String s=jTextField4.getText();  
int l=s.length();  
if(s.isEmpty()==false)  
{  
    for(int i=0;i<l;i++)  
    {  
        char ch=s.charAt(i);  
        if(Character.isDigit(ch)==false)  
        {  
            JOptionPane.showMessageDialog(this,"Price must be in digits");  
            jTextField4.setText("");  
            break;  
        }  
    }  
}  
}
```

If a person goes with **Dispose Book Record** option, the following window opens where the user can dispose a registered book.



This form is used to dispose a book, in case it is torn, stolen or harmed in any way.



Coding for Dispose A Book Form :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
  
Main_Page ob=new Main_Page();  
ob.setVisible(true);  
this.dispose();  
  
}private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
  
update_book_records ob=new update_book_records();  
ob.setVisible(true);  
this.dispose();  
  
}private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
jButton2.setEnabled(true);  
jTextField1.setText("");  
jTextField2.setText("");  
jTextField3.setText("");  
  
DefaultListModel m=(DefaultListModel)(jList1.getModel());  
m.clear();  
  
try  
{  
    Class.forName("java.sql.Driver");  
    Connection con=(Connection)  
    DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement) con.createStatement();
    Statement stmt1=(Statement) con.createStatement();
    String s="select accession_number from book_information order by Accession_number asc;";
    ResultSet rs=stmt.executeQuery(s);
    ResultSet rs1=stmt1.executeQuery(s);
    rs1.first();
    String k=rs1.getString(1);
    while(rs.next())
    {
        String a=rs.getString(1);
        m.addElement(a);
    }
}
```

```

        rs.close();
        stmt.close();
        con.close();
    }

    catch(Exception e)
    {
        if(e.getMessage().equals("Illegal operation on empty result set."))
        {
            JOptionPane.showMessageDialog(this,"No records to display");
        }
        else
        {
            JOptionPane.showMessageDialog(this,e.getMessage());
        }
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
Object d=jList1.getSelectedValue();
if(d==null)
{
    JOptionPane.showMessageDialog(this,"No Record Selected");
}
else
{
    try
    {
        Class.forName("java.sql.Driver");
        Connection con=(Connection)
        DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement) con.createStatement();
        String s1="insert into disposed(Accession_number) values('"+d+"');";
        int i=stmt.executeUpdate(s1);
        if(i==1)
        {

```

```

JOptionPane.showMessageDialog(this,"Records have been safely disposed");
jButton2.setEnabled(false);
}
stmt.close();
con.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
}

private void jList1MouseClicked(java.awt.event.MouseEvent evt) {
jButton2.setEnabled(true);
try
{ Class.forName("java.sql.Driver");
Connection con=(Connection)
DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt=(Statement) con.createStatement();
Object a1=jList1.getSelectedValue();
String s1="select * from disposed;";
ResultSet rs1=stmt.executeQuery(s1);
int count=0;
while(rs1.next())
{
if(a1.equals(rs1.getString(1)))
{
count=1;
}
}
if(count==1)
{ jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
}
}
}

```

```

jButton2.setEnabled(false);

JOptionPane.showMessageDialog(this,"These records have been disposed");

}

else if(count==0)

{

jButton2.setEnabled(true);

String s="select Title,Author,Genre from book_information where accession_number ='"+a1+"';";

ResultSet rs=stmt.executeQuery(s);

while(rs.next())

{

String a=rs.getString(1);

String b= rs.getString(2);

String c= rs.getString(3);

jTextField1.setText(a);

jTextField2.setText(b);

jTextField3.setText(c);

}

}

stmt.close();

con.close();

}

catch(Exception e)

{

if(jList1.getSelectedValue()==null)

{

JOptionPane.showMessageDialog(this,"No book Selected");

}

else

{

JOptionPane.showMessageDialog(this,e.getMessage());

}

}

}

```

If a person clicks on **Update Visitors Records** button, the following window opens.

We have already seen this form on the front page.

Button to back one form.



This form contains two options :

- Add a new member.
- Dispose a member.

Coding for Update Visitors Form :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
register_new_visitor ob=new register_new_visitor();
```

```
ob.setVisible(true);
```

```
this.dispose();
```

```
}
```

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
Main_Page ob=new Main_Page();
```

```
ob.setVisible(true);
```

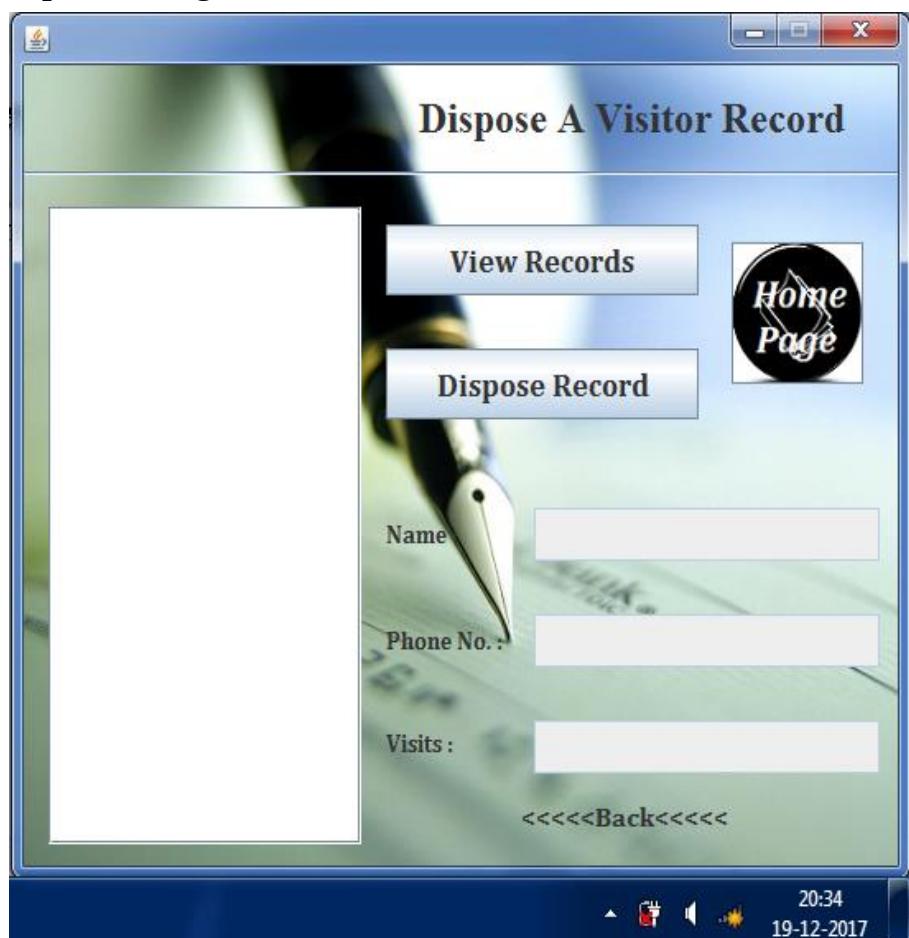
```
this.dispose();
```

```
}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose_visitor_record ob=new dispose_visitor_record();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    database_updation ob=new database_updation();
    ob.setVisible(true);
    this.dispose();
}
```

If a person goes with **Dispose A Record** option, the following window opens where the user can dispose a registered member.



This form is used to dispose a member, in case the person is no longer a part of the library.

Coding for Dispose A Visitor Form :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    Main_Page ob=new Main_Page();
    ob.setVisible(true);
    this.dispose();
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    jButton2.setEnabled(true);
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    DefaultListModel l=(DefaultListModel)(jList1.getModel());
    l.clear();
    try
```

```
{
```

```
    Class.forName("java.sql.Driver");
    Connection
    con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
    Statement stmt=(Statement)con.createStatement();
    Statement stmt1=(Statement)con.createStatement();
    String s="select LibraryCard_number from Registered_issuers order by LibraryCard_number asc";
    ResultSet rs=stmt.executeQuery(s);
    ResultSet rs1=stmt1.executeQuery(s);
    rs1.first();
    String k=rs1.getString(1);
    while(rs.next())
    {
        String a=rs.getString(1);
        l.addElement(a);
    }
    rs.close();
    stmt.close();
```

```
con.close();  
}  
catch(Exception e)  
{  
    if(e.getMessage().equals("Illegal operation on empty result set."))  
    {  
        JOptionPane.showMessageDialog(this,"No records to display");  
    }  
    else  
    {  
        JOptionPane.showMessageDialog(this,e.getMessage());  
    }  
}
```

```
private void jList1MouseClicked(java.awt.event.MouseEvent evt) {  
  
int count=0;  
  
if(jList1.getSelectedValue()==null)  
{  
    JOptionPane.showMessageDialog(this,"No Record Present");  
}  
else  
{  
    try  
{  
        Class.forName("java.sql.Driver");  
        Connection  
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
        Statement stmt=(Statement)con.createStatement();
        Object a1=jList1.getSelectedValue();
        String s1="select * from disposed;";
        ResultSet rs1=stmt.executeQuery(s1);
        int count1=0;
```

```

while(rs1.next())
{
    if(a1.equals(rs1.getString(2)))
    {
        count1=1;
    }
}
if(count1==1)
{
    jButton2.setEnabled(false);
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    JOptionPane.showMessageDialog(this,"These records have been disposed");
}
else if(count1==0)
{
    jButton2.setEnabled(true);
    String s="select Name_of_issuer,Phone_numbers from registered_issuers where
LibraryCard_number ="+a1+";";
    ResultSet rs=stmt.executeQuery(s);
    while(rs.next())
    {
        String a=rs.getString(1);
        String b= rs.getString(2);
        jTextField1.setText(a);
        jTextField2.setText(b);
    }
    String a="select Record_Number from Transaction_records where
LibraryCard_Number="+a1+";";
    ResultSet r1=stmt.executeQuery(a);
    while(r1.next())
    {
        count++;
    }
}

```

```

    }

    int k=count;

    if(k==1)

    {

        jTextField3.setText(Integer.toString(k)+" visit");

    }

    else

    {

        jTextField3.setText(Integer.toString(k)+" visits");

    }

}

stmt.close();

con.close();

}

catch(Exception e)

{

    if(jList1.getSelectedValue()==null)

    {

        JOptionPane.showMessageDialog(this,"No Library Card Number Selected");

    }

    else

    {

        JOptionPane.showMessageDialog(this,e.getMessage());

    }

}

}

}

}

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

Object d=jList1.getSelectedValue();

if(d==null)

{

JOptionPane.showMessageDialog(this,"No Record Selected");

```

```

jTextField3.setText("");
}

else
{
try
{
Class.forName("java.sql.Driver");
Connection
con=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt=(Statement)con.createStatement();
String s1="insert into disposed(LibraryCard_number) values('"+d+"');";
int i=stmt.executeUpdate(s1);
if(i==1)
{
JOptionPane.showMessageDialog(this,"Records have been safely disposed");
jButton2.setEnabled(false);
}
stmt.close();
con.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
}
}
}
}

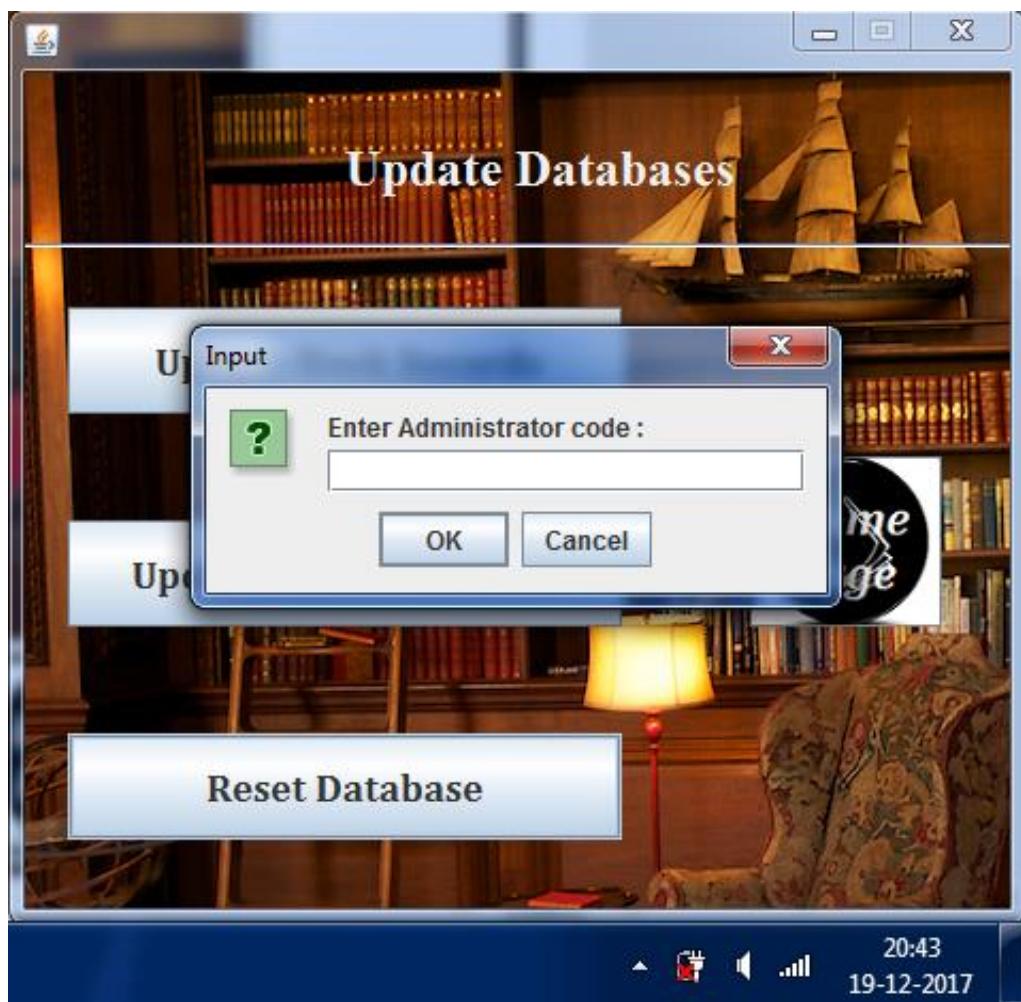
```

```

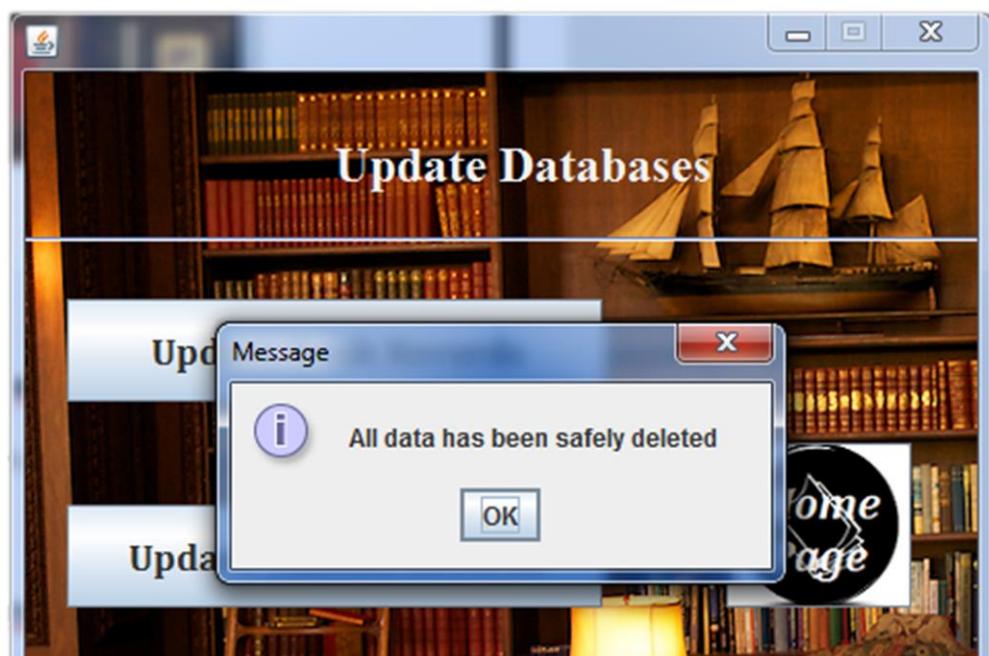
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
update_visitors_record ob=new update_visitors_record();
ob.setVisible(true);
this.dispose();
}

```

If the person selects **Reset Database** option he must contain an Administrator code to carry out the task of resetting the database.



If Administrator code is correct.



Coding for Reset Database Button :

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
try  
{    String s=JOptionPane.showInputDialog(this,"Enter Administrator code :");  
if(s.equals("KT149209"))  
{  
    int i=JOptionPane.showConfirmDialog(this,"Do you wish to delete all records?","Reset  
Data",JOptionPane.YES_NO_OPTION);  
    if(i==0)  
    {  
        try{  
            Class.forName("java.sql.Driver");  
            Connection con=(Connection)  
DriverManager.getConnection("jdbc:mysql://localhost/library","root","");
Statement stmt=(Statement) con.createStatement();
String s1="delete from transaction_records;";
String s2="alter table transaction_records auto_increment=1;";
String s3="delete from book_information";
String s4="delete from disposed";
String s5="delete from registered_issuers";
stmt.executeUpdate(s1);
stmt.executeUpdate(s2);
stmt.executeUpdate(s3);
stmt.executeUpdate(s4);
stmt.executeUpdate(s5);
JOptionPane.showMessageDialog(this,"All data has been safely deleted");
stmt.close();
con.close();
}  
catch(Exception e)
{
    JOptionPane.showMessageDialog(this,e.getMessage());
}
}
}
```

```
else
{
    database_updation ob=new database_updation();
    ob.setVisible(true);
    this.dispose();
}

}
else
{
    JOptionPane.showMessageDialog(this,"Incorrect Administrator code");
}
}

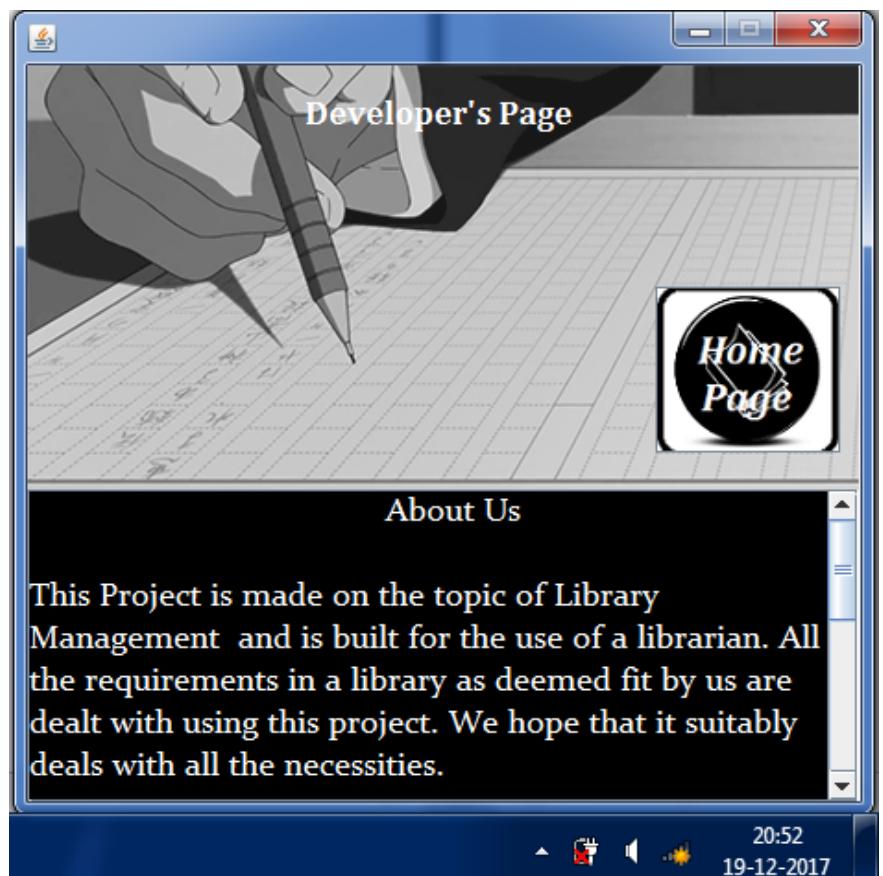
catch(Exception e)
{
    database_updation ob=new database_updation();
    ob.setVisible(true);
    this.dispose();
}

}
```

If a person clicks on “ About Us ” link on upper right hand corner of the Front Page form he will reach the developers page.



The person can scroll to read the problem description of this project.



If a person clicks on  button on the Front Page form he will get this option.



The user's choice will either log out the program or make it continue to function.

Bibliography

- Classroom notes
- Informatics practices book (CBSE publication)
- Stackoverflow.com