|  |  |
| --- | --- |
| e-Library Management System  A Database Approach | Abstract  This project aims to develop a sample application for managing library resources in institutions.  Database Systems Lab |

**TABLE OF CONTENTS**

1. **INTRODUCTION**

1. **SYSTEM ANALYSIS AND DESIGN**

1. **DATA BASE DESIGN**
2. **ER-DIAGRAM**

1. **GUI DESIGN & CONNECTIVITY**

1. **FUTURE SCOPE**

1. **CONCLUSION**

1. **BIBLIOGRAPHY**

# 1.0 INTRODUCTION

This e-Library Management System project is aimed at doing away with manual system of tracking of day to day activities of a typical library and making complete automation of manual or semi-automatic process of library management process.

This project is aimed at overcoming the following problems of manual system:

* Fast report generation is not possible
* Tracing a book is difficult
* Information about issue/return of the books are not properly maintained
* No central database can be created as information is not available in database

This application is of immense help for entire eco-system comprising Users, Library Staff and Management as depicted in below table:

|  |  |
| --- | --- |
| **CATEGORY** | **USAGE** |
| Book User | Check availability of books |
| Search the Books |
| Obtain the Books |
| Library Staff | Check availability of books |
| Search the Books |
| Issue the Books |
| Receive the Books |
| Management | Get various Reports on usage and system functioning |

The application is developed using NetBeans IDE & Java programming language for front end and MySQL as back end database. With the help of this application, it is possible to avoid tedious process of using other methods such as MS Excel/Word Document etc.

2.0 SYSTEM ANALYSIS AND DESIGN

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key questions are- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

During analysis, data collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram, interviews, etc. Training, experience and common sense are required for collection of relevant information needed to develop the system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs.

* System analysis can be categorized into four parts.
* System planning and initial investigation
* Information Gathering
* Applying analysis tools for structured analysis  Feasibility study  Cost/ Benefit analysis.

The Library Management System has been designed based on the real world scenario & developed for a receipt and issue of books in the library along with the member’s details.

The books received in the library are entered in Books Entry form and the new member is entered in the member entry form. When the member wants to get the desired book the same is issued on the availability basis to the member. The issuance and due date for the returning of the book is also entered into the Book Issue form under third menu Book Issue. From an end-user perspective, the Library Management System Project consists of three functional elements: Master, Transaction and Report Module.

A library database needs to store information pertaining to its users and assets (i.e books, other electronic media).The library must keep track of the status of each user and asset: its location, status, descriptive attributes, and cost for losses and late returns. Books will be identified by their ID, Name, Author and Publisher. In order to allow multiple copies of the same book each media item will have a unique ID number. Users will provide their name, e-mail id, address and phone number when signing up for a library card. They will then be assigned a unique user name and ID number, plus a temporary password that will have to be changed. Checkout operations will require a library card, as will requests to put media on hold.

# 

# 3.0 DATA BASE DESIGN

Database design involves constructing a suitable model. Since the design process is complicated, especially for large databases, database design is divided into three phases:

* Conceptual database design
* Logical database design
* Physical database design

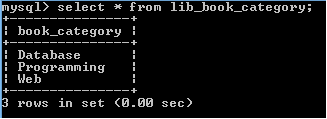
The Database has been designed using MySQL comprising 

MySQL Command Prompt

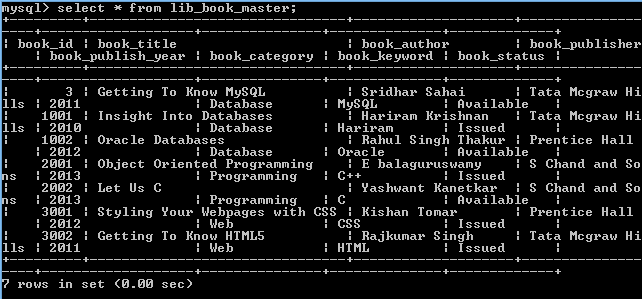
The following tables are created for this project:



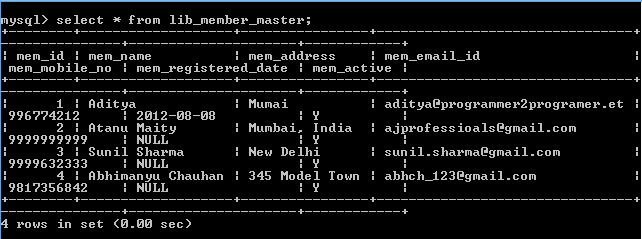
* 1. lib\_book\_category: This table is for various book categories- e.g Database, Programming, Web etc. which can be further customized.



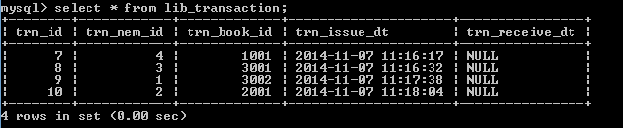
* 1. lib\_book\_master: This table stores the info of all the books



* 1. lib\_member\_master: This table contains the records of members



* 1. lib\_transaction : This table provides details of the issuance and receipt of the books

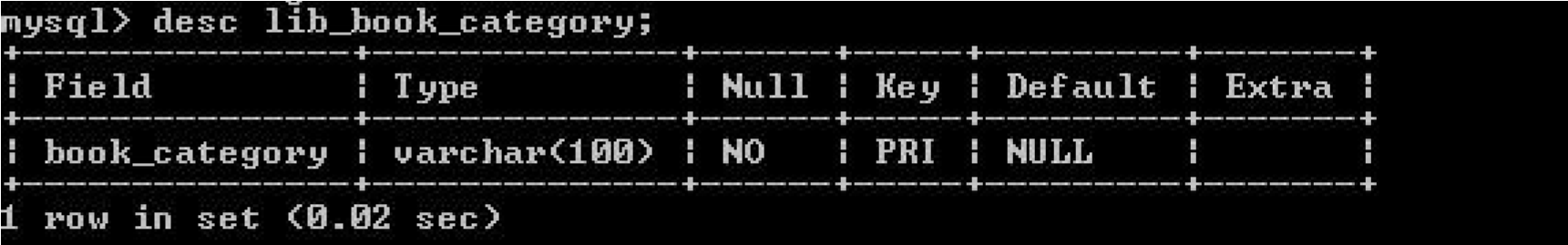


* 1. lib\_user: This table provides the details of the user category: User or Admin

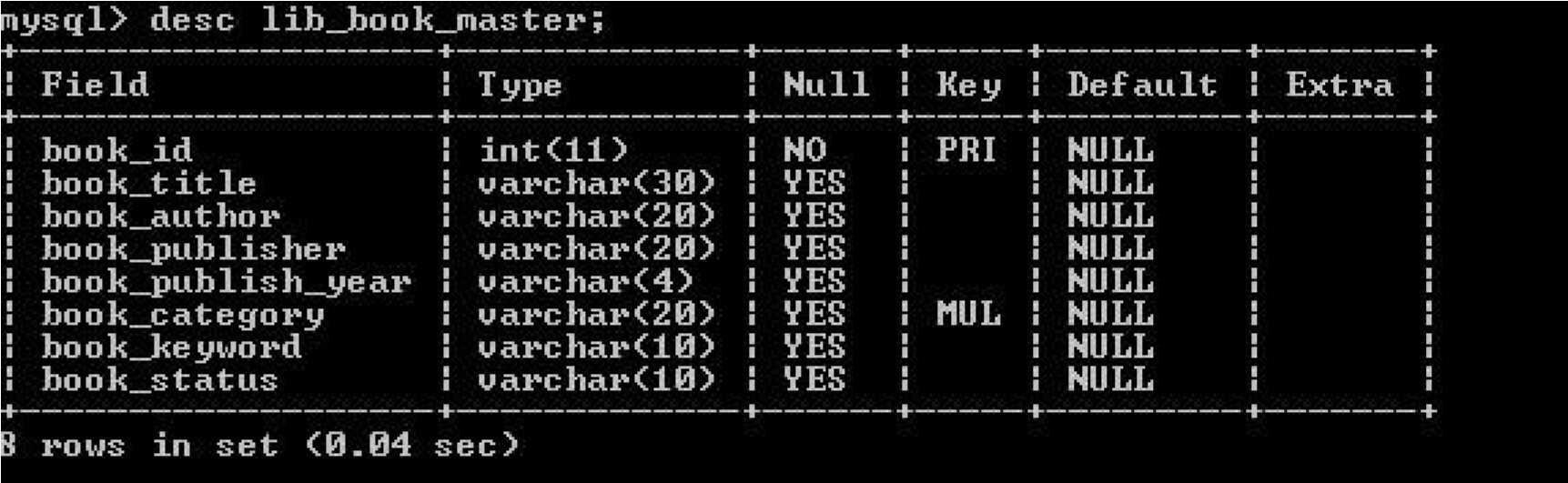


# ->CONSTRAINTS ON TABLES :

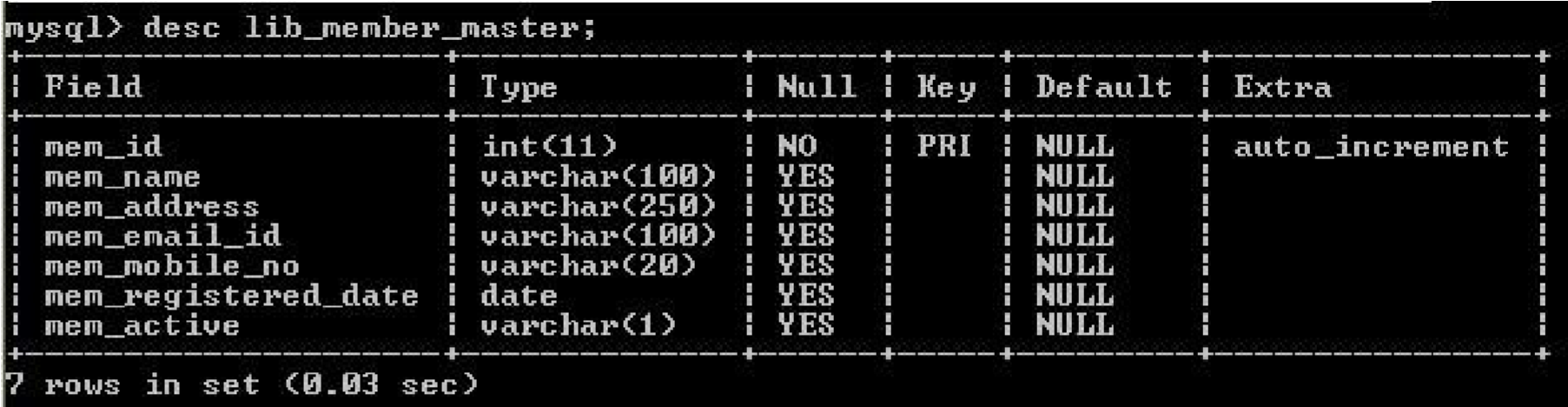
1-) Table lib\_book\_category :



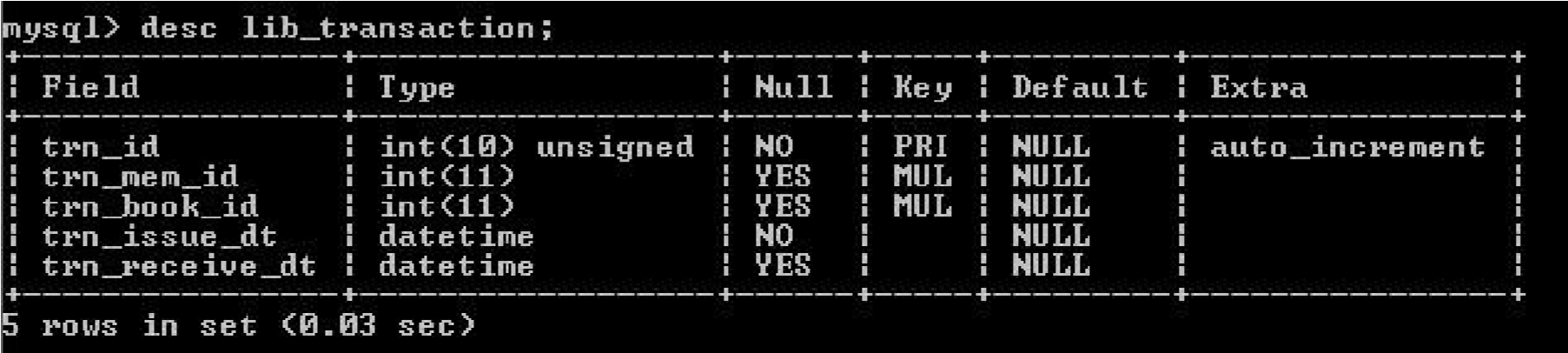
2-) Table lib\_book\_master :



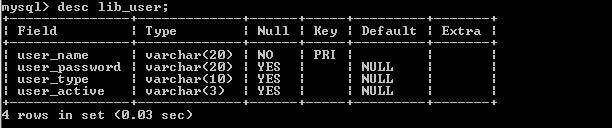
3-) Table lib\_member\_master :



4-) Table lib\_transaction :



5-) Table lib\_user :



# 4.0 ER-DIAGRAM

|  |
| --- |
| lib\_BOOK\_CATEGORY |

|  |
| --- |
| BOOK\_CATEGORY |

|  |
| --- |
| Lib\_book\_master |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Book\_id | Book\_title | Book\_author | Book\_publisher | Book\_publish\_year | Book\_category | Book\_keyword | Book\_status |

|  |
| --- |
| Lib\_member\_master |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mem\_id | Mem\_name | Mem\_address | Mem\_email\_id | Mem\_mobile\_no | Mem\_register | Mem\_active |

|  |
| --- |
| Lib\_transaction |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trn\_id | Trn\_mem\_id | Trn\_book\_id | Trn\_issue\_dt | Trn\_receive\_dt |

|  |
| --- |
| Lib\_user |

|  |  |  |  |
| --- | --- | --- | --- |
| User\_name | **User\_password** | **User\_type** | **User\_active** |

# 5.0 GUI DESIGN & CONNECTIVITY

The Graphical User Interface (GUI) or the Front End has been designed using the rich features of NetBeans IDE for Java Programming.

A good GUI design is based on the following principles:

**The structure principle**

The design should organize the user interface purposefully, in meaningful and useful ways based on clear, consistent models that are apparent and recognizable to users, putting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another. The structure principle is concerned with our overall user interface architecture.

**The simplicity principle**

The design should make simple, common tasks simple to do, communicating clearly and simply in the user’s own language, and providing good shortcuts that are meaningfully related to longer procedures.

**The visibility principle**

The design should keep all needed options and materials for a given task visible without distracting the user with extraneous or redundant information. Good designs don’t overwhelm users with too many alternatives or confuse them with unneeded information.

**The feedback principle**

The design should keep users informed of actions or interpretations, changes of state or condition, and errors or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users.

**The tolerance principle**

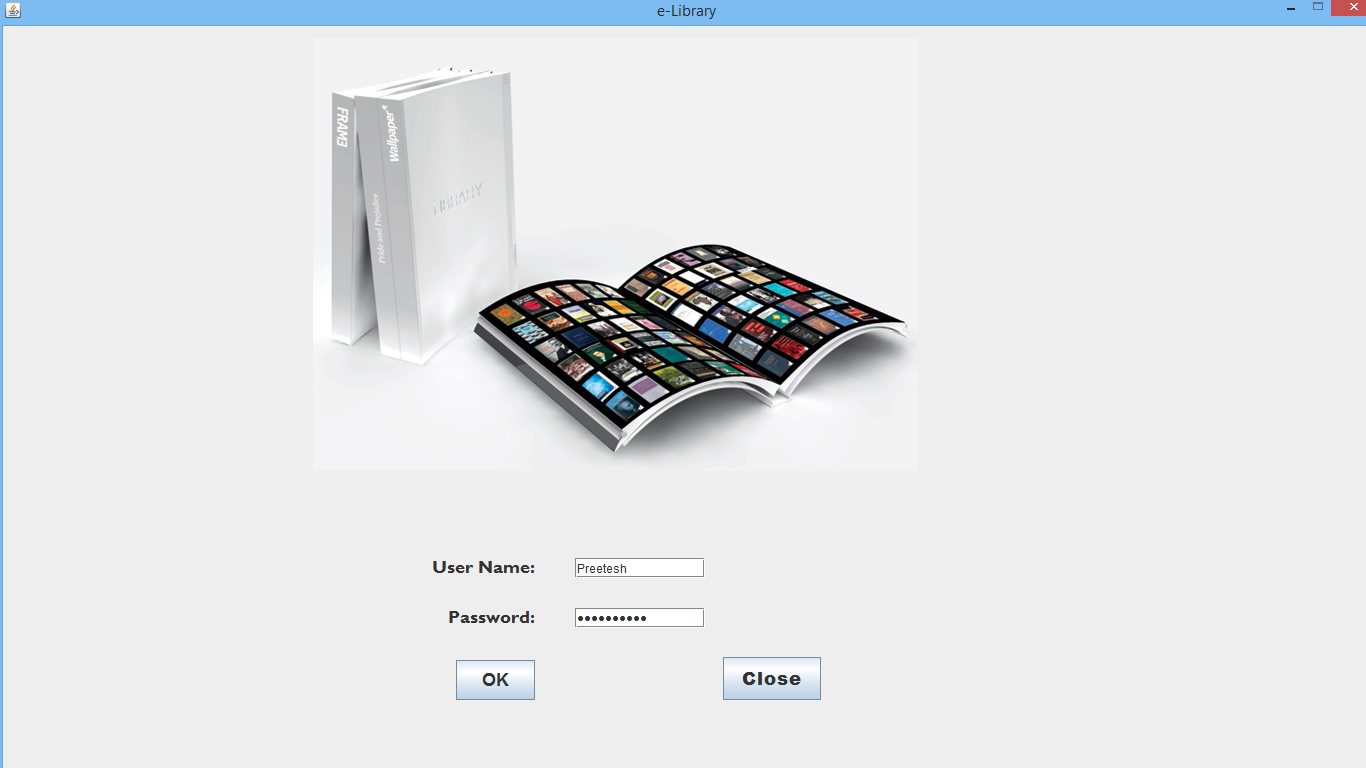
The design should be flexible and tolerant, reducing the cost of mistakes and misuse by allowing undoing and redoing, while also preventing errors wherever possible by tolerating varied inputs and sequences and by interpreting all reasonable actions reasonable.

**The reuse principle**

The design should reuse internal and external components and behaviours, maintaining consistency with purpose rather than merely arbitrary consistency, thus reducing the need for users to rethink and remember.

The following are the screen shots and source code for various GUI forms

## e-Library Login Page (Login.java):



Before going to the login page the **Main Class** is created as per the following code:

public class MainClass

{

//change your MySQL database connection here

public String StrUrl="jdbc:mysql://localhost/p2p\_library";

public String StrUid="root"; public String StrPwd=

"yash96"; public static String StrUser;

}

## Source code for the login page :

import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import

java.sql.ResultSet; import javax.swing.JOptionPane;

/\*Login.java

\*@author Administrator

\*/

public class Login extends javax.swing.JFrame {

/\*\* Creates new form Login \*/

public Login() {

initComponents();

}

private void CmdCloseActionPerformed(java.awt.event.ActionEvent evt) { // Close project

System.exit(1);

}

private void CmdOkActionPerformed(java.awt.event.ActionEvent evt) { //validation

String stru=""; stru=TxtUserName.getText();

String strp=""; strp=TxtPassword.getText(); if

(stru.isEmpty()==true)

{

JOptionPane.showMessageDialog(null,"Enter User Name"); return;

}

if (strp.isEmpty()==true)

{

JOptionPane.showMessageDialog(null,"Enter Password"); return; } try

{

//get database connection details

MainClass mc=new MainClass();

//open connection Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd); String str=""; str="select \* from lib\_user where user\_name =?

and user\_password =?";

PreparedStatement pst=connection.prepareStatement(str);

pst.setString(1, stru);

pst.setString(2, strp); ResultSet rs;

rs=pst.executeQuery(); if

(rs.next())

{

MainClass.StrUser=TxtUserName.getText(); MainForm m=new MainForm(); m.setVisible(true); this.setVisible(false);

} else

{

JOptionPane.showMessageDialog(null,"User name or password are not correct."); return;

}

}

catch (Exception e)

{

System.err.println(e);

System.exit(1);

}

}

private void formWindowOpened(java.awt.event.WindowEvent evt) { // TODO add your handling code here: this.setLocationRelativeTo(null);

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() { public void run() {

new Login().setVisible(true);

}

});

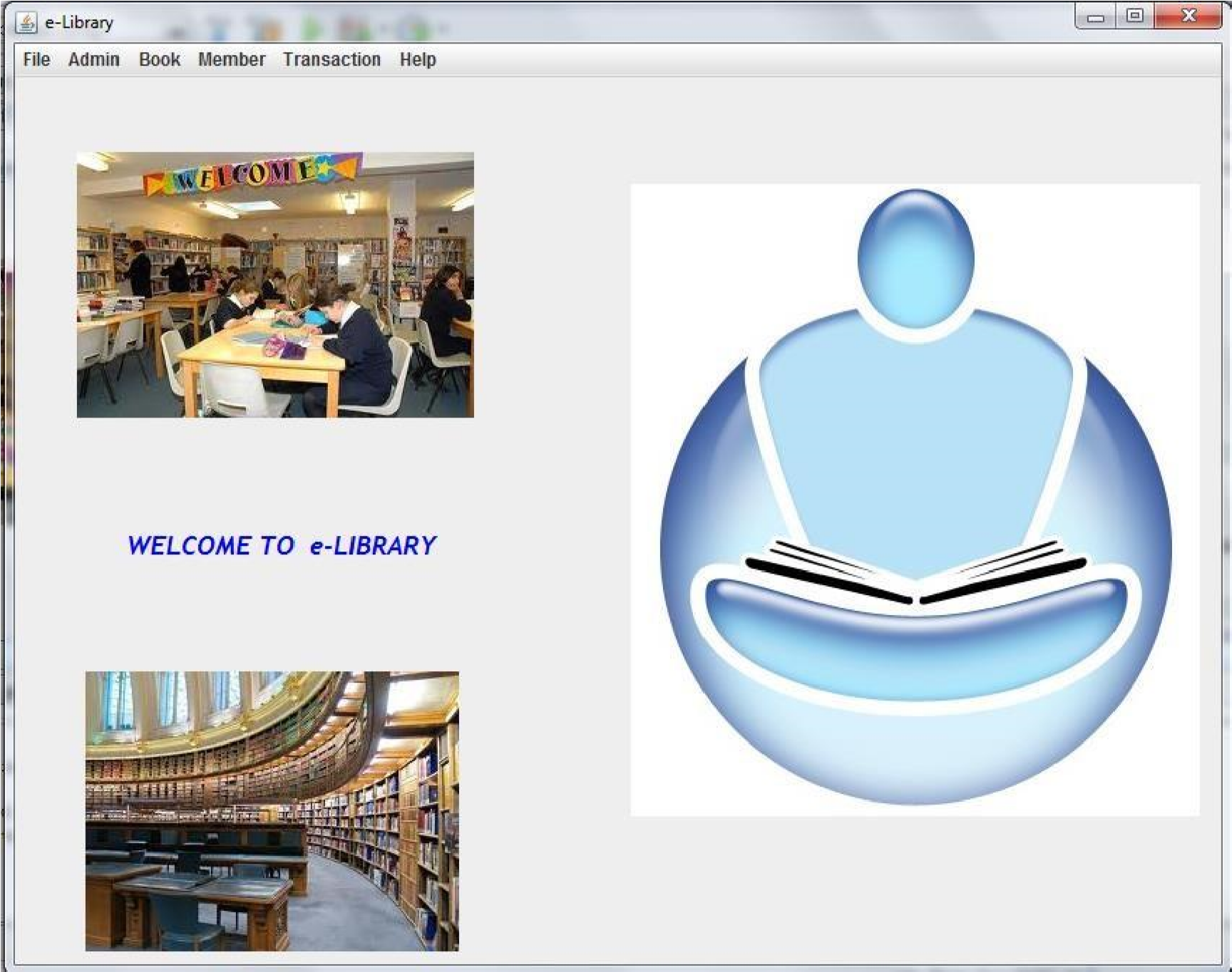
}

// Variables declaration - do not modify private javax.swing.JButton CmdClose; private javax.swing.JButton CmdOk; private javax.swing.JPasswordField TxtPassword; private javax.swing.JTextField TxtUserName; private javax.swing.JLabel jLabel1; private javax.swing.JLabel jLabel2; private javax.swing.JLabel jLabel3; private javax.swing.JPanel jPanel1;

// End of variables declaration

}

## Menu Options jFrame (MainForm.java) :



## Source code for MainFrame.java :

public class MainForm extends javax.swing.JFrame {

/\*\* Creates new form Main \*/ public MainForm() {

initComponents();

}

/\*\* This method is called from within the constructor to \* initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is \* always regenerated by the Form Editor.

\*/

private void MnuExitsActionPerformed(java.awt.event.ActionEvent evt) { System.exit(1);

}

private void MnuBookMasterActionPerformed(java.awt.event.ActionEvent evt) { BookMaster b=new BookMaster(); b.setVisible(true);

}

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

this.setLocationRelativeTo(null);

}

private void MnuSearchBookActionPerformed(java.awt.event.ActionEvent evt) { SearchBook b=new SearchBook(); b.setVisible(true);

}

private void MnuBookStatusActionPerformed(java.awt.event.ActionEvent evt) { BookStatus b=new BookStatus(); b.setVisible(true);

}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {

MemberMaster b=new MemberMaster();

b.setVisible(true);

}

private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {

SearchMember b=new SearchMember();

b.setVisible(true);

}

private void jMenuItem7ActionPerformed(java.awt.event.ActionEvent evt) { UserMaster b=new UserMaster(); b.setVisible(true);

}

private void jMenuItem8ActionPerformed(java.awt.event.ActionEvent evt) {

ChangePassword b=new ChangePassword();

b.setVisible(true);

}

private void jMenuItem5ActionPerformed(java.awt.event.ActionEvent evt) { About b=new About(); b.setVisible(true);

}

private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) { IssueBook b=new IssueBook(); b.setVisible(true);

}

private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) { ReceiveBook b=new ReceiveBook(); b.setVisible(true);

}

private void jMenu4ActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) { TransactionReport b=new TransactionReport(); b.setVisible(true);

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() { public void run() { new MainForm().setVisible(true);

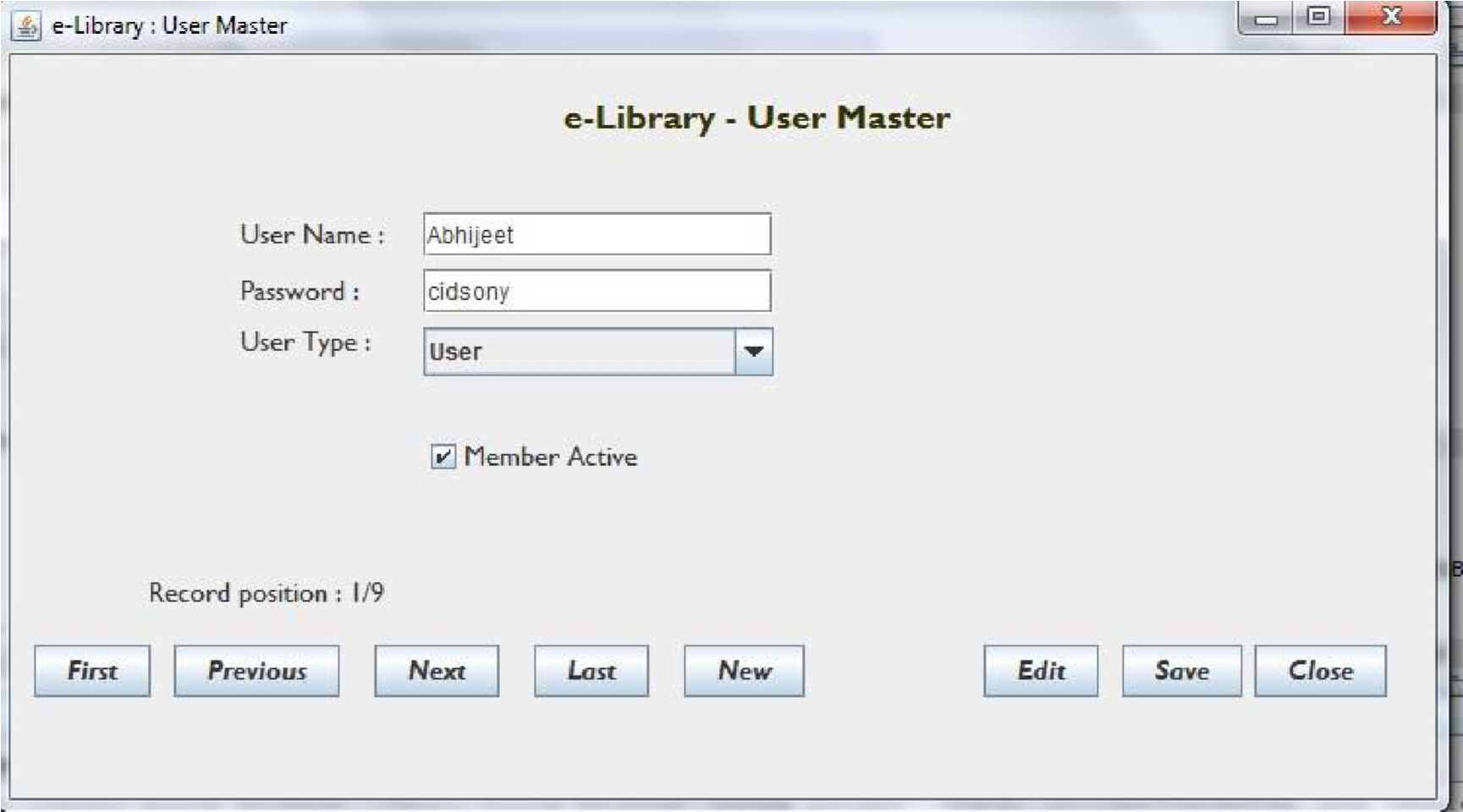
}

});

}

}

## User Master (UserMaster.java) :



## Source code for UserMaster.java :

import java.sql.Statement; import

java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import

java.sql.ResultSet; import

javax.swing.JOptionPane;

public class UserMaster extends javax.swing.JFrame {

MainClass mc=new MainClass();

ResultSet rsuser; int rspos=0; int rscount=0; boolean chadd=false;

/\*\* Creates new form UserMaster \*/

public UserMaster() {

initComponents();

}

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

try {

rsuser.first();

Display();

} catch (Exception e) {

System.err.println(e);

}

} private void

CmdPreviousActionPerform ed(java.awt.event.ActionEve

nt evt) { try {

if (rsuser.isFirst()==true) {

JOptionPane.showMessageDialog(null,"Already in first record."); return;

}

rsuser.previous();

Display();

} catch (Exception e) {

System.err.println(e);

}

}

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

try {

if (rsuser.isLast()==true) {

JOptionPane.showMessageDialog(null,"Already in last record."); return; }

rsuser.next();

Display();

} catch (Exception e) {

System.err.println(e);

}

}

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

try {

rsuser.last();

Display();

} catch (Exception e) {

System.err.println(e);

//System.exit(1);

}

}

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

chadd=true;

ClearText();

}

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

chadd=false;

}

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

//validate entry and save record try {

if (TxtUserName.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter user name."); return;

}

if (TxtPassword.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter password."); return; }

String strqr="";

Connection connection;

connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

Statement stmt = connection.createStatement();

ResultSet rs;

System.out.println("Point A0"); //test stub1

if (chadd==true ) {

rsuser.moveToInsertRow(); //test stub2 System.out.println("Point A1"); rsuser.updateString("user\_name",TxtUserName.getText());

rsuser.updateString("user\_password",TxtPassword.getText());

rsuser.updateString("user\_type",CmbUserType.getSelectedItem().toString());

if (ChkUserActive.isSelected()==true) {

rsuser.updateString("user\_active","YES");

} else {

rsuser.updateString("user\_active","NO");

}

rsuser.insertRow();

System.out.println("Point A2"); //test stub3 rsuser.last();

System.out.println("Point A3"); //test stub4

} else {

rsuser.updateString("user\_password",TxtPassword.getText());

rsuser.updateString("user\_type",CmbUserType.getSelectedItem().toString());

if (ChkUserActive.isSelected()==true) {

rsuser.updateString("user\_active","YES");

} else {

rsuser.updateString("user\_active","NO");

}

rsuser.updateRow();

rsuser.refreshRow();

}

stmt=connection.prepareStatement(""); rs =

stmt.executeQuery("select count(\*) from lib\_user");

while(rs.next()){ // rs.first();

rscount=rs.getInt(1);} Display(); chadd=false;

JOptionPane.showMessageDialog(null,"Record updated.");

} catch (Exception e) {

System.err.println(e);

JOptionPane.showMessageDialog(null,"Unable to save record."); return;

}

}

private void CmdCloseActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: this.setVisible(false);

} private void formWindowOpened(java.a

wt.event.WindowEvent evt) { try

{

this.setLocationRelativeTo(null);

ClearText();

CmbUserType.removeAllItems();

CmbUserType.addItem("Admin");

CmbUserType.addItem("User");

Connection connection;

connection=DriverManager.getConnection("jdbc:mysql://localhost/p2p\_library","root","yash96");

ResultSet rs;

PreparedStatement stmt;

//get user count stmt=connection.prepareStatement("select count(\*) from lib\_user");

rs = stmt.executeQuery();

while(rs.next()){

rscount=rs.getInt(1);} //get user details

Statement stmt1 = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

String sqlQuery = "select \* from lib\_user ";

rsuser = stmt1.executeQuery(sqlQuery);

//display first record

rsuser.first();

Display();

}

catch (Exception e)

{

System.err.println(e);

//System.exit(1);

}

}

private void Display()

{ try

{

//clear text of fields for entry/display old data

TxtUserName.setText(rsuser.getString("user\_name"));

TxtPassword.setText(rsuser.getString("user\_password"));

CmbUserType.setSelectedItem(rsuser.getString("user\_type"));

if (rsuser.getString("user\_active").equals("YES"))

{

ChkUserActive.setSelected(true);

} else {

ChkUserActive.setSelected(false);

}

LblPosition.setText("Record position : " + rsuser.getRow() + "/" + String.valueOf(rscount));

}

catch (Exception e)

{

System.err.println(e);

}

}

private void ClearText()

{

//clear text of fields for entry/display old data

TxtUserName.setText("");

TxtPassword.setText("");

ChkUserActive.setSelected(false);

LblPosition.setText("");

}

public static void main(String args[]) { java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new UserMaster().setVisible(true);

}

});

}

## Change Password:

This form is used to change the password



//get database connection details

MainClass mc=new MainClass();

//open connection Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

String str="";

str="select \* from lib\_user where user\_name =? and user\_password =?";

PreparedStatement pst=connection.prepareStatement(str);

pst.setString(1, MainClass.StrUser); pst.setString(2, TxtOldPassword.getText()); ResultSet rs; rs=pst.executeQuery(); if (rs.next()) {

//Update password

String sql = "UPDATE lib\_user SET user\_password='" + TxtNewPassword.getText() + "' WHERE user\_name = '" + MainClass.StrUser +"'";

pst.executeUpdate(sql);

JOptionPane.showMessageDialog(null,"Password updated.");

this.setVisible(false);

} else {

JOptionPane.showMessageDialog(null,"Old password are not correct."); return; }

} catch (Exception e) {

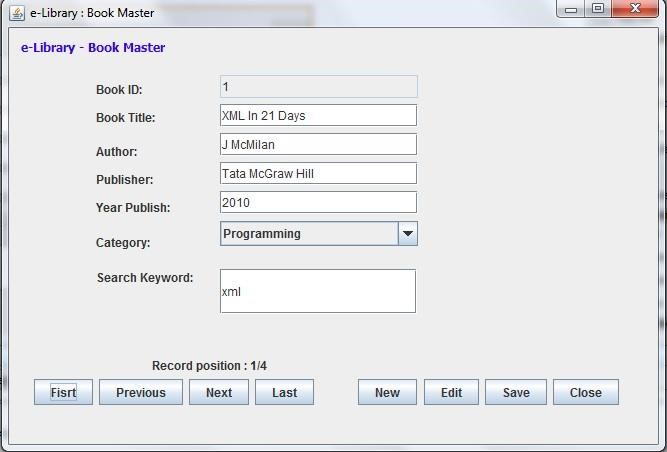
System.err.println(e);

System.exit(1);

}

}

## Book Master Form (BookMaster.java) :



\*/

public class BookMaster extends javax.swing.JFrame {

MainClass mc=new MainClass();

ResultSet rsbook; int rspos=0; int rscount=0; boolean chadd=false;

/\*\* Creates new form BookMaster \*/

public BookMaster() {

initComponents();

}

private void CmdCloseActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: this.setVisible(false);

}

private void formWindowOpened(java.awt.event.WindowEvent evt) { // TODO add your handling code here:

try

{

this.setLocationRelativeTo(null);

ClearText();

CmbCategory.removeAllItems();

Connection connection;

connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs;

//fill book category

PreparedStatement stmt=connection.prepareStatement("select \* from lib\_book\_category order by book\_category"); rs = stmt.executeQuery(); while (rs.next()){

CmbCategory.addItem(rs.getString(1));

}

rs.close();

//get book count stmt=connection.prepareStatement("select ifnull(max(book\_id),0) from lib\_book\_master");

rs = stmt.executeQuery(); while(rs.next()) { //rs.first

rscount=rs.getInt(1);

}

//get book details

Statement stmt1 = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

String sqlQuery = "select \* from lib\_book\_master order by book\_id"; rsbook = stmt1.executeQuery(sqlQuery);

//display first record rsbook.first();

Display();

}

catch (Exception e)

{

System.err.println(e);

//System.exit(1);

}

}

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

chadd=true;

ClearText();

}

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

try

{

if (rsbook.isLast()==true)

{

JOptionPane.showMessageDialog(null,"Already in last record."); return;

}

rsbook.next();

Display();

}

catch (Exception e)

{

System.err.println(e);

}

}

private void CmdFirstActionPerformed(java.awt.event.ActionEvent evt) { try {

rsbook.first();

Display();

}

catch (Exception e)

{

System.err.println(e);

}

}

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

try

{

if (rsbook.isFirst()==true)

{

JOptionPane.showMessageDialog(null,"Already in first record."); return;

}

rsbook.previous(); Display();

}

catch (Exception e)

{

System.err.println(e);

}

}

private void CmdLastActionPerformed(java.awt.event.ActionEvent evt) { try {

rsbook.last();

Display();

}

catch (Exception e)

{

System.err.println(e);

//System.exit(1);

}

}

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

chadd=false;

}

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

//validate entry and save record try

{

if (TxtBookTitle.getText().trim().length()==0 )

{

JOptionPane.showMessageDialog(null,"Enter book title."); return;

}

if (TxtAuthor.getText().trim().length()==0 )

{

JOptionPane.showMessageDialog(null,"Enter author."); return;

}

if (TxtPublisher.getText().trim().length()==0)

{

JOptionPane.showMessageDialog(null,"Enter publisher name."); return;

}

if (CmbCategory.getSelectedItem().equals(""))

{

JOptionPane.showMessageDialog(null,"Select book category."); return;

}

//String strqr="";

Connection connection ; connection = DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

Statement stmt = connection.createStatement(); ResultSet rs;

if (TxtBookID.getText().trim().length()==0 )

{

// String query ="insert into lib\_book\_master (book\_title, book\_author, book\_publisher, book\_publish\_year, book\_category, book\_keyword) values ('" + TxtBookTitle.getText() + "','" +

TxtAuthor.getText() +"','"+ TxtPublisher.getText() +"','" + TxtYear.getText() + "','" + CmbCategory.getSelectedItem().toString() +"','" + TxtKeywords.getText() +"')";

rsbook.moveToInsertRow();

//stmt=connection.prepareStatement("select ifnull(max(book\_id)+1,1) from lib\_book\_master");

rs = stmt.executeQuery("select ifnull(max(book\_id)+1,1) from lib\_book\_master");

while(rs.next()) { //rs.first

rscount=rs.getInt(1);

}

rsbook.updateInt("book\_id",rscount);

rsbook.updateString("book\_title",TxtBookTitle.getText()); rsbook.updateString("book\_author",TxtAuthor.getText()); rsbook.updateString("book\_publisher",TxtPublisher.getText());

rsbook.updateString("book\_publish\_year",TxtYear.getText());

rsbook.updateString("book\_category",CmbCategory.getSelectedItem().toString());

rsbook.updateString("book\_keyword",TxtKeywords.getText());

rsbook.insertRow();

rsbook.last();

Display(); } else {

// String query ="insert into lib\_book\_master (book\_title, book\_author, book\_publisher, book\_publish\_year, book\_category, book\_keyword) values ('" + TxtBookTitle.getText() + "','" +

TxtAuthor.getText() +"','"+ TxtPublisher.getText() +"','" + TxtYear.getText() + "','" + CmbCategory.getSelectedItem().toString() +"','" + TxtKeywords.getText() +"')";

rsbook.updateString("book\_title",TxtBookTitle.getText());

rsbook.updateString("book\_author",TxtAuthor.getText()); rsbook.updateString("book\_publisher",TxtPublisher.getText());

rsbook.updateString("book\_publish\_year",TxtYear.getText());

rsbook.updateString("book\_category",CmbCategory.getSelectedItem().toString());

rsbook.updateString("book\_keyword",TxtKeywords.getText());

rsbook.updateRow();

rsbook.refreshRow();

}

JOptionPane.showMessageDialog(null,"Record updated.");

}

catch (Exception e)

{

System.err.println(e);

JOptionPane.showMessageDialog(null,"Unable to save record."); return;

} }

private void Display()

{ try

{

//clear text of fields for entry/display old data

TxtBookID.setText(rsbook.getString("book\_id"));

TxtBookTitle.setText(rsbook.getString("book\_title"));

TxtAuthor.setText(rsbook.getString("book\_author"));

TxtPublisher.setText(rsbook.getString("book\_publisher"));

TxtYear.setText(rsbook.getString("book\_publish\_year"));

CmbCategory.setSelectedItem(rsbook.getString("book\_category"));

TxtKeywords.setText(rsbook.getString("book\_keyword"));

LblPosition.setText("Record position : " + rsbook.getRow() + "/" + String.valueOf(rscount));

}

catch (Exception e)

{ System.err.println(e); }

}

private void ClearText()

{

//clear text of fields for entry/display old data

TxtBookID.setText(""); TxtBookTitle.setText(""); TxtAuthor.setText(""); TxtPublisher.setText(""); TxtYear.setText(""); CmbCategory.setSelectedItem(null); TxtKeywords.setText(""); LblPosition.setText("");

}

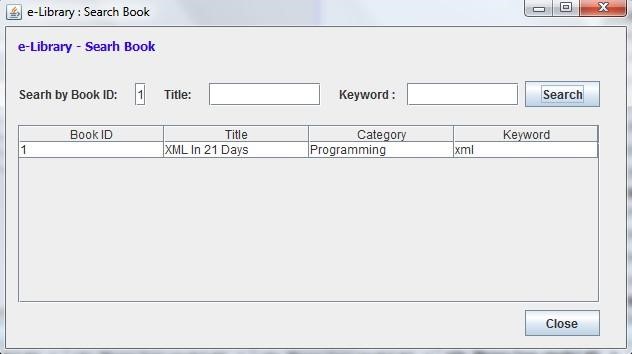
public static void main(String args[]) { java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new BookMaster().setVisible(true); } } }

**Book Search Form (SearchBook.java) :**

**Source code :**



public class SearchBook extends javax.swing.JFrame {

/\*\* Creates new form SearchBook \*/

public SearchBook() {

initComponents();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: try

{

MainClass mc=new MainClass(); Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs;

String StrQr="";

if (TxtBookID.getText().trim().length()>0 )

{

StrQr=StrQr + " and book\_id = " + TxtBookID.getText().trim() + " ";

}

if (TxtTitle.getText().trim().length()>0 )

{

StrQr=StrQr + " and book\_title like '%" + TxtTitle.getText().trim() + "%' ";

}

if (TxtKeyword.getText().trim().length()>0 )

{

StrQr=StrQr + " and book\_keyword like '%" + TxtKeyword.getText().trim() + "%' ";

}

if (StrQr.length()==0)

{

JOptionPane.showMessageDialog(null,"Enter search critaria."); return; }

PreparedStatement stmt=connection.prepareStatement("select book\_id, book\_title, book\_category, book\_keyword from lib\_book\_master where 1=1 " + StrQr + " order by book\_id"); rs = stmt.executeQuery();

jTable1.getColumnModel().getColumn(0).setHeaderValue("Book ID");

jTable1.getColumnModel().getColumn(1).setHeaderValue("Title"); jTable1.getColumnModel().getColumn(2).setHeaderValue("Category"); jTable1.getColumnModel().getColumn(3).setHeaderValue("Keyword"); jTable1.getTableHeader().resizeAndRepaint();

// Removing Previous Data while (jTable1.getRowCount() > 0) {

((DefaultTableModel) jTable1.getModel()).removeRow(0);

}

//Creating Object []rowData for jTable's Table Model

int columns = rs.getMetaData().getColumnCount();

while (rs.next())

{

Object[] row = new Object[columns];

for (int i = 1; i <= columns; i++)

{

row[i - 1] = rs.getObject(i); // 1

}

((DefaultTableModel) jTable1.getModel()).insertRow(rs.getRow() - 1,row);

} }

catch (Exception e)

{

System.err.println(e);

//System.exit(1);

}

}

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

this.setVisible(false);

}

private void formWindowOpened(java.awt.event.WindowEvent evt) { TxtBookID.setText("");

TxtTitle.setText("");

TxtKeyword.setText("");

this.setLocationRelativeTo(null);

jTable1.getColumnModel().getColumn(0).setHeaderValue("Book ID");

jTable1.getColumnModel().getColumn(1).setHeaderValue("Title"); jTable1.getColumnModel().getColumn(2).setHeaderValue("Category"); jTable1.getColumnModel().getColumn(3).setHeaderValue("Keyword"); jTable1.getTableHeader().resizeAndRepaint();

// Removing Previous Data while (jTable1.getRowCount() > 0) {

((DefaultTableModel) jTable1.getModel()).removeRow(0);

}

}

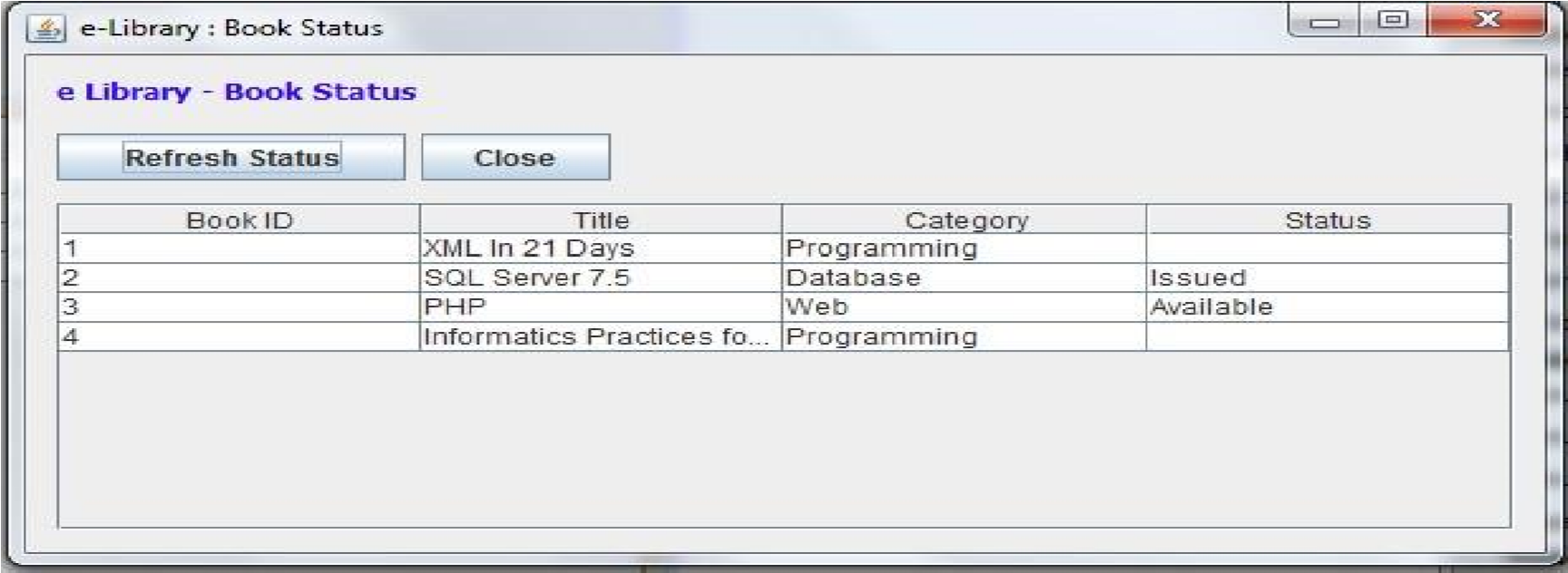
public static void main(String args[]) { java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new SearchBook().setVisible(true);

} }); }

## Book Status Form (BookStatus.java) :



## Source Code :

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

try {

MainClass mc=new MainClass();

Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs;

PreparedStatement stmt=connection.prepareStatement("select book\_id, book\_title, book\_category, book\_status from lib\_book\_master order by book\_id"); rs = stmt.executeQuery();

jTable1.getColumnModel().getColumn(0).setHeaderValue("Book ID");

jTable1.getColumnModel().getColumn(1).setHeaderValue("Title"); jTable1.getColumnModel().getColumn(2).setHeaderValue("Category"); jTable1.getColumnModel().getColumn(3).setHeaderValue("Status"); jTable1.getTableHeader().resizeAndRepaint();

// Removing Previous Data while (jTable1.getRowCount() > 0) {

((DefaultTableModel) jTable1.getModel()).removeRow(0);

}

//Creating Object []rowData for jTable's Table Model

int columns = rs.getMetaData().getColumnCount();

while (rs.next()) {

Object[] row = new Object[columns]; for (int i = 1; i <= columns; i++) {

row[i - 1] = rs.getObject(i); // 1

}

((DefaultTableModel) jTable1.getModel()).insertRow(rs.getRow() - 1,row);

}

} catch (Exception e) {

System.err.println(e);

//System.exit(1);

}

}

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

this.setVisible(false);

}

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

this.setLocationRelativeTo(null);

jTable1.getColumnModel().getColumn(0).setHeaderValue("Book ID"); jTable1.getColumnModel().getColumn(1).setHeaderValue("Title"); jTable1.getColumnModel().getColumn(2).setHeaderValue("Category"); jTable1.getColumnModel().getColumn(3).setHeaderValue("Status"); jTable1.getTableHeader().resizeAndRepaint();

// Removing Previous Data while (jTable1.getRowCount() > 0) {

((DefaultTableModel) jTable1.getModel()).removeRow(0);

} }

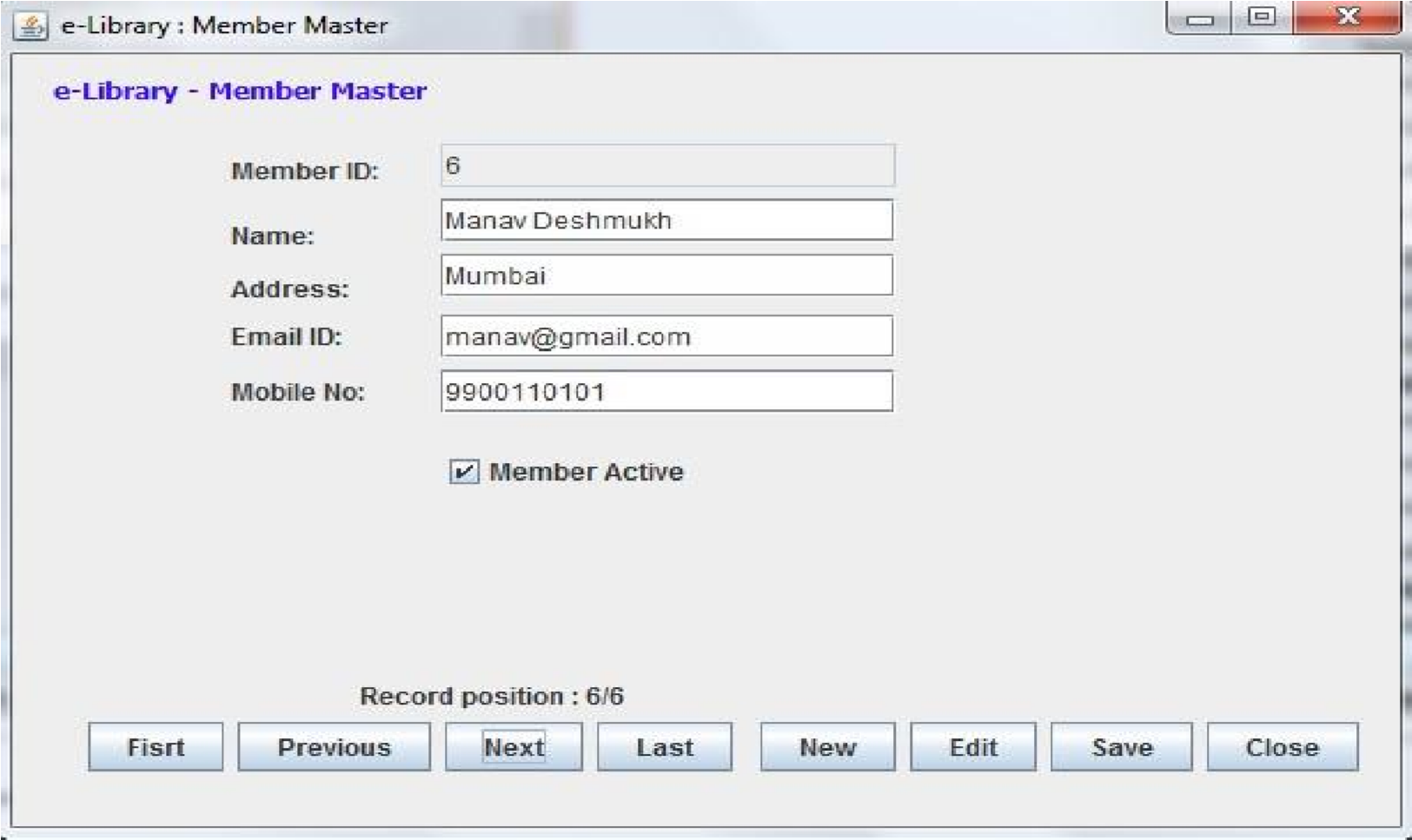
public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() { public void run() {

new BookStatus().setVisible(true);

} }); }

## Member Master Form (MemberMaster.java) :



## Source Code :

public class MemberMaster extends javax.swing.JFrame {

MainClass mc=new MainClass();

ResultSet rsmem; int rspos=0; int rscount=0; boolean chadd=false;

/\*\* Creates new form MemberMaster \*/ public MemberMaster() {

initComponents();

} { try

{

//clear text of fields for entry/display old data

TxtMemID.setText(rsmem.getString("mem\_id"));

TxtName.setText(rsmem.getString("mem\_name"));

TxtAddress.setText(rsmem.getString("mem\_address"));

TxtEmailID.setText(rsmem.getString("mem\_email\_id"));

TxtMobileNo.setText(rsmem.getString("mem\_mobile\_no"));

if (rsmem.getString("mem\_active").equals("Y"))

{

ChkMemActive.setSelected(true);

} else

{

ChkMemActive.setSelected(false);

}

LblPosition.setText("Record position : " + rsmem.getRow() + "/" + String.valueOf(rscount));

}

catch (Exception e)

{

System.err.println(e);

}

}

private void ClearText()

{

//clear text of fields for entry/display old data

TxtMemID.setText("");

TxtName.setText("");

TxtAddress.setText("");

TxtEmailID.setText("");

TxtMobileNo.setText("");

ChkMemActive.setSelected(false);

LblPosition.setText("");

}

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

try {

rsmem.first();

Display();

} catch (Exception e) {

System.err.println(e);

}

}

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

try {

if (rsmem.isFirst()==true) {

JOptionPane.showMessageDialog(null,"Already in first record."); return;

}

rsmem.previous();

Display();

} catch (Exception e) {

System.err.println(e);

}

}

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

try {

if (rsmem.isLast()==true) {

JOptionPane.showMessageDialog(null,"Already in last record."); return;

}

rsmem.next();

Display();

} catch (Exception e) {

System.err.println(e);

} }

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

try {

rsmem.last();

Display();

} catch (Exception e) {

System.err.println(e);

//System.exit(1);

}

}

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

chadd=true;

ClearText();

}

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

chadd=false;

}

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

//validate entry and save record

try {

if (TxtName.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter member name."); return;

}

if (TxtAddress.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter address."); return;

}

if (TxtEmailID.getText().trim().length()==0) {

JOptionPane.showMessageDialog(null,"Enter email id."); return;

}

if (TxtMobileNo.getText().trim().length()==0) {

JOptionPane.showMessageDialog(null,"Enter mobile no."); return;

}

Connection connection;

connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs; if (TxtMemID.getText().trim().length()==0 ) { rsmem.moveToInsertRow();

rsmem.updateString("mem\_name",TxtName.getText());

rsmem.updateString("mem\_address",TxtAddress.getText()); rsmem.updateString("mem\_email\_id",TxtEmailID.getText()); rsmem.updateString("mem\_mobile\_no",TxtMobileNo.getText());

if (ChkMemActive.isSelected()==true)

{

rsmem.updateString("mem\_active","Y");

}

else

{

rsmem.updateString("mem\_active","N");

}

rsmem.insertRow();

rsmem.last();

Display();

} else {

rsmem.updateString("mem\_name",TxtName.getText());

rsmem.updateString("mem\_address",TxtAddress.getText()); rsmem.updateString("mem\_email\_id",TxtEmailID.getText()); rsmem.updateString("mem\_mobile\_no",TxtMobileNo.getText());

if (ChkMemActive.isSelected()==true)

{

rsmem.updateString("mem\_active","Y");

} else

{

rsmem.updateString("mem\_active","N");}

rsmem.updateRow();

rsmem.refreshRow();

}

PreparedStatement stmt; stmt=connection.prepareStatement("select count(\*) from lib\_member\_master");

rs = stmt.executeQuery();

while(rs.next()){ // rs.first();

rscount=rs.getInt(1);}

Display();

JOptionPane.showMessageDialog(null,"Record updated.");

} catch (Exception e) {

System.err.println(e);

JOptionPane.showMessageDialog(null,"Unable to save record."); return;

}

}

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

this.setVisible(false);

}

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

try

{

this.setLocationRelativeTo(null);

ClearText();

Connection connection;

connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs;

//get mem count PreparedStatement stmt; stmt=connection.prepareStatement("select count(\*) from lib\_member\_master"); rs = stmt.executeQuery();

while(rs.next()){ // rs.first();

rscount=rs.getInt(1);}

//get mem details

Statement stmt1 = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

String sqlQuery = "select \* from lib\_member\_master order by mem\_id"; rsmem = stmt1.executeQuery(sqlQuery);

//display first record rsmem.first();

Display();

}

catch (Exception e)

{

System.err.println(e);

//System.exit(1);

} }

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() { public void run() {

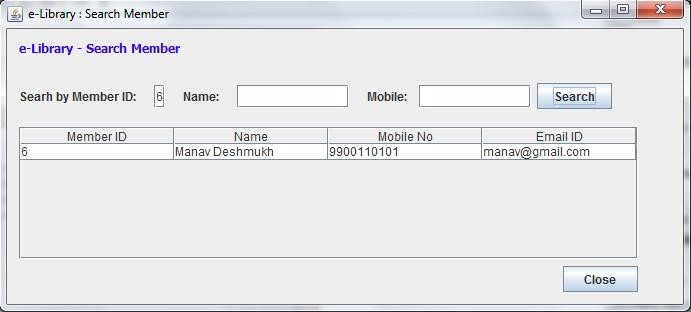
new MemberMaster().setVisible(true);

}

});

}

**Search Member Form (SearchMember.java) :**



**Source Code :**



public class MemberMaster extends javax.swing.JFrame {

MainClass mc=new MainClass();

ResultSet rsmem; int rspos=0; int rscount=0; boolean chadd=false;

/\*\* Creates new form MemberMaster \*/ public MemberMaster() {

initComponents();

} { try {

//clear text of fields for entry/display old data

TxtMemID.setText(rsmem.getString("mem\_id"));

TxtName.setText(rsmem.getString("mem\_name"));

TxtAddress.setText(rsmem.getString("mem\_address"));

TxtEmailID.setText(rsmem.getString("mem\_email\_id"));

TxtMobileNo.setText(rsmem.getString("mem\_mobile\_no"));

if (rsmem.getString("mem\_active").equals("Y"))

{

ChkMemActive.setSelected(true);

} else

{

ChkMemActive.setSelected(false);

}

LblPosition.setText("Record position : " + rsmem.getRow() + "/" + String.valueOf(rscount));

}

catch (Exception e)

{

System.err.println(e);

}

}

private void ClearText()

{

//clear text of fields for entry/display old data

TxtMemID.setText("");

TxtName.setText("");

TxtAddress.setText("");

TxtEmailID.setText("");

TxtMobileNo.setText("");

ChkMemActive.setSelected(false);

LblPosition.setText("");

}

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

try {

rsmem.first();

Display();

} catch (Exception e) {

System.err.println(e);

}

}

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

try {

if (rsmem.isFirst()==true) {

JOptionPane.showMessageDialog(null,"Already in first record."); return;

}

rsmem.previous();

Display();

} catch (Exception e) {

System.err.println(e);

}

}

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

try {

if (rsmem.isLast()==true) {

JOptionPane.showMessageDialog(null,"Already in last record."); return;

}

rsmem.next();

Display();

} catch (Exception e) {

System.err.println(e); }

}

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

try {

rsmem.last();

Display();

} catch (Exception e) {

System.err.println(e);

}

}

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

chadd=true;

ClearText();

}

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

chadd=false;

}

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

//validate entry and save record try {

if (TxtName.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter member name."); return;

}

if (TxtAddress.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter address."); return;

}

if (TxtEmailID.getText().trim().length()==0) {

JOptionPane.showMessageDialog(null,"Enter email id."); return;

}

if (TxtMobileNo.getText().trim().length()==0) {

JOptionPane.showMessageDialog(null,"Enter mobile no."); return;

}

Connection connection;

connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs; if (TxtMemID.getText().trim().length()==0 ) {

rsmem.moveToInsertRow();

rsmem.updateString("mem\_name",TxtName.getText());

rsmem.updateString("mem\_address",TxtAddress.getText()); rsmem.updateString("mem\_email\_id",TxtEmailID.getText());

rsmem.updateString("mem\_mobile\_no",TxtMobileNo.getText());

if (ChkMemActive.isSelected()==true)

{

rsmem.updateString("mem\_active","Y");

} else

{

rsmem.updateString("mem\_active","N");

}

rsmem.insertRow();

rsmem.last();

Display();

} else {

rsmem.updateString("mem\_name",TxtName.getText());

rsmem.updateString("mem\_address",TxtAddress.getText()); rsmem.updateString("mem\_email\_id",TxtEmailID.getText());

rsmem.updateString("mem\_mobile\_no",TxtMobileNo.getText());

if (ChkMemActive.isSelected()==true) rsmem.updateString("mem\_active","Y");

else

rsmem.updateString("mem\_active","N");

rsmem.updateRow(); rsmem.refreshRow();

}

PreparedStatement stmt; stmt=connection.prepareStatement("select count(\*) from lib\_member\_master"); rs = stmt.executeQuery(); while(rs.next()){ // rs.first();

rscount=rs.getInt(1);}

Display();

JOptionPane.showMessageDialog(null,"Record updated.");

} catch (Exception e) {

System.err.println(e);

JOptionPane.showMessageDialog(null,"Unable to save record."); return;

}

}

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

this.setVisible(false);

} private void formWindowOpened(java.a

wt.event.WindowEvent evt) { try{

this.setLocationRelativeTo(null);

ClearText();

Connection connection;

connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs;

//get mem count PreparedStatement stmt; stmt=connection.prepareStatement("select count(\*) from lib\_member\_master");

rs = stmt.executeQuery();

while(rs.next()){ // rs.first(); rscount=rs.getInt(1);}

//get mem details

Statement stmt1 = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

String sqlQuery = "select \* from lib\_member\_master order by mem\_id"; rsmem = stmt1.executeQuery(sqlQuery);

//display first record rsmem.first();

Display();

}

catch (Exception e)

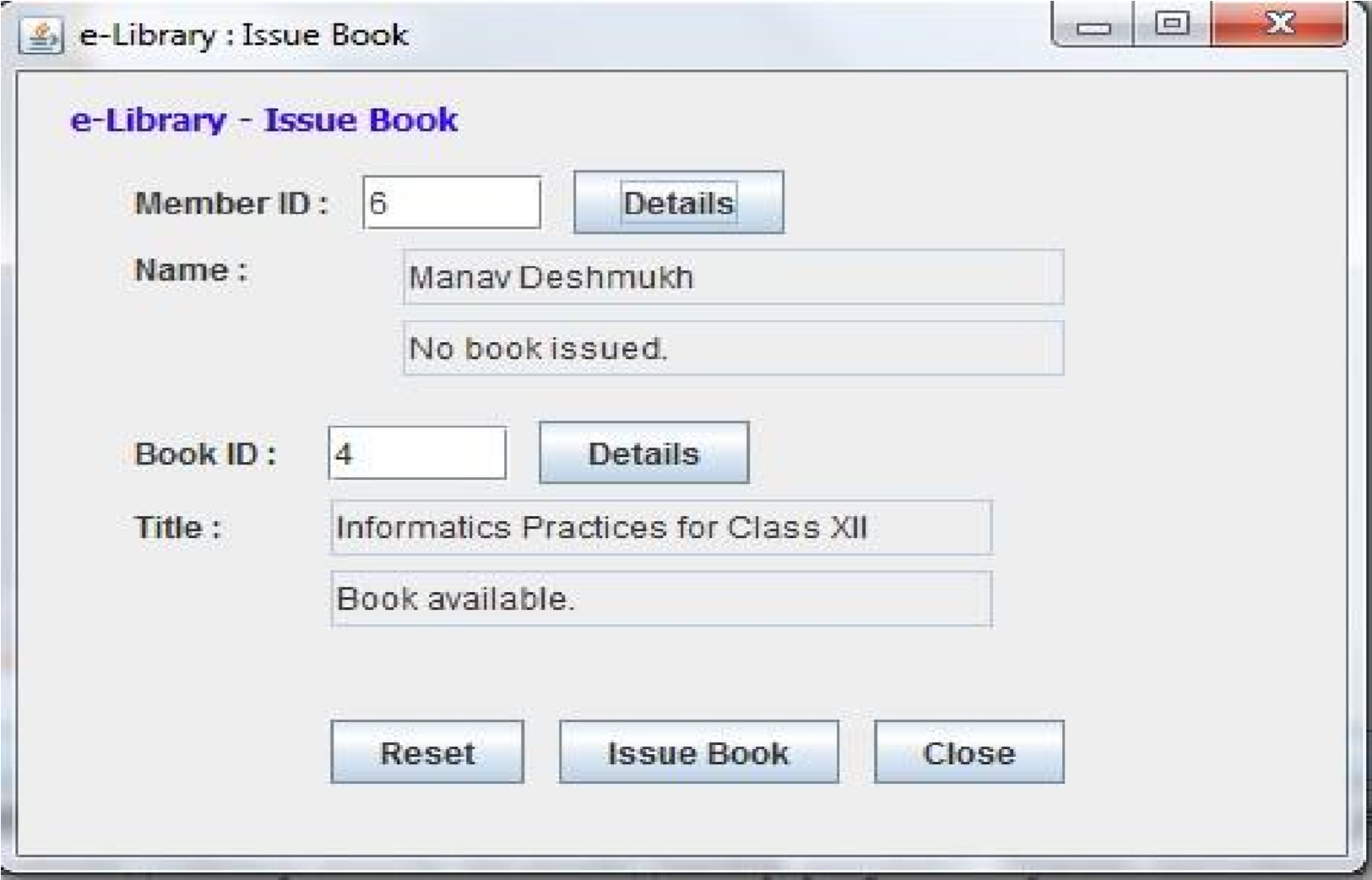
{

System.err.println(e);

//System.exit(1);

} }

## Issue Book Form (IssueBook.java) :



## Source Code :

public class IssueBook extends javax.swing.JFrame {

/\*\* Creates new form IssueBook \*/

public IssueBook() {

initComponents();

}

private void CmdResetActionPerformed(java.awt.event.ActionEvent evt) { ClearText();

}

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

//issue book try

{

if (TxtMemberID.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter member id."); return;

}

if (TxtBookID.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter member id."); return;

}

MainClass mc=new MainClass(); Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

String sql = "insert into lib\_transaction (trn\_mem\_id,trn\_book\_id,trn\_issue\_dt) values (" +

TxtMemberID.getText() + "," + TxtBookID.getText() + ",sysdate())";

PreparedStatement pst=connection.prepareStatement(sql);

pst.executeUpdate(sql);

sql = "update lib\_book\_master set book\_status='Issued' where book\_id=" +

TxtBookID.getText(); pst=connection.prepareStatement(sql); pst.executeUpdate(sql);

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

ClearText();

} catch (Exception e) { System.err.println(e);

System.exit(1); } } private void CmdCloseActionPerformed(java.awt.event.ActionEvent evt) {

this.setVisible(false);

}

private void formWindowOpened(java.awt.event.WindowEvent evt) { ClearText(); this.setLocationRelativeTo(null); }

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

//get member deatils try

{

//get database connection details

MainClass mc=new MainClass();

//open connection Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

String str=""; str="select \* from lib\_member\_master where mem\_id =? ";

PreparedStatement pst=connection.prepareStatement(str); pst.setString(1, TxtMemberID.getText());

ResultSet rs; rs=pst.executeQuery(); if (rs.next()) {

TxtMemberName.setText(rs.getString("mem\_name"));

//get mem status str="select \* from lib\_transaction where trn\_mem\_id =? and (trn\_receive\_dt is null or

trn\_receive\_dt='')";

PreparedStatement pst1=connection.prepareStatement(str); pst1.setString(1, TxtMemberID.getText());

ResultSet rs1; rs1=pst1.executeQuery();

if (rs1.next())

{

TxtMemberStatus.setText("Book already issued.");

CmdIssueBook.setEnabled(false);

CmdBookDetails.setEnabled(false);

} else {

TxtMemberStatus.setText("No book issued.");

CmdIssueBook.setEnabled(true);

CmdBookDetails.setEnabled(true);

}

} else {

JOptionPane.showMessageDialog(null,"Invalid member id.");

TxtMemberID.setText("");

TxtMemberName.setText("");

TxtMemberStatus.setText("");

CmdIssueBook.setEnabled(false); return;

}

}

catch (Exception e)

{

System.err.println(e);

System.exit(1);

}

}

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

//get member deatils

try

{

//get database connection details

MainClass mc=new MainClass();

//open connection Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

String str=""; str="select \* from lib\_book\_master where book\_id =? ";

PreparedStatement pst=connection.prepareStatement(str); pst.setString(1, TxtBookID.getText());

ResultSet rs; rs=pst.executeQuery(); if

(rs.next())

{

TxtBookTitle.setText(rs.getString("book\_title"));

//get mem status str="select \* from lib\_transaction where trn\_book\_id =? and (trn\_receive\_dt is null or

trn\_receive\_dt='')";

PreparedStatement pst1=connection.prepareStatement(str); pst1.setString(1, TxtBookID.getText());

ResultSet rs1; rs1=pst1.executeQuery(); if

(rs1.next())

{

TxtBookStatus.setText("Book not available.");

CmdIssueBook.setEnabled(false);

} else

{

TxtBookStatus.setText("Book available.");

CmdIssueBook.setEnabled(true);

}

} else

{

JOptionPane.showMessageDialog(null,"Invalid book id.");

TxtBookID.setText("");

TxtBookTitle.setText("");

TxtBookStatus.setText("");

CmdIssueBook.setEnabled(false); return;

}

}

catch (Exception e)

{

System.err.println(e);

System.exit(1);

}

}

private void ClearText()

{

TxtMemberID.setText("");

TxtMemberName.setText("");

TxtMemberStatus.setText("");

TxtBookID.setText("");

TxtBookTitle.setText("");

TxtBookStatus.setText("");

CmdIssueBook.setEnabled(false);

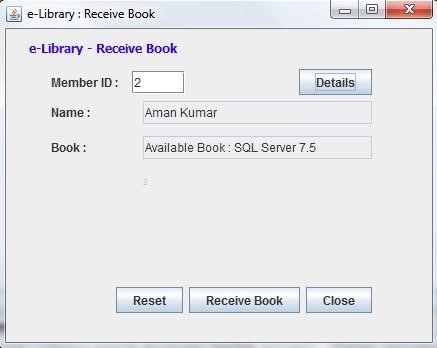
CmdBookDetails.setEnabled(false);

}

**Source Code :**



**Receive Book Form (ReceiveBook.java) :**



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

//get member deatils

try {

//get database connection details

MainClass mc=new MainClass();

//open connection Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

String str=""; str="select \* from lib\_member\_master where mem\_id =? ";

PreparedStatement pst=connection.prepareStatement(str); pst.setString(1, TxtMemberID.getText());

ResultSet rs; rs=pst.executeQuery(); if

(rs.next()) {

TxtMemberName.setText(rs.getString("mem\_name"));

//get mem status str="select \* from lib\_transaction, lib\_book\_master where book\_id=trn\_book\_id and

trn\_mem\_id =? and (trn\_receive\_dt is null or trn\_receive\_dt='')";

PreparedStatement pst1=connection.prepareStatement(str); pst1.setString(1, TxtMemberID.getText());

ResultSet rs1;

rs1=pst1.executeQuery();

if (rs1.next()) {

TxtBookDetails.setText("Available Book : " + rs1.getString("book\_title"));

CmdReceiveBook.setEnabled(true);

LblBookID.setText(rs1.getString("book\_id"));

} else {

TxtBookDetails.setText("No book available.");

CmdReceiveBook.setEnabled(false);

LblBookID.setText("");

}

} else {

JOptionPane.showMessageDialog(null,"Invalid member id.");

TxtMemberID.setText("");

TxtMemberName.setText("");

TxtBookDetails.setText("");

CmdReceiveBook.setEnabled(false);

LblBookID.setText("");

return;

}

} catch (Exception e) {

System.err.println(e);

System.exit(1);

}

}

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

ClearText();

}

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

//issue book try {

if (TxtMemberID.getText().trim().length()==0 ) {

JOptionPane.showMessageDialog(null,"Enter member id."); return;

}

MainClass mc=new MainClass(); Connection connection; connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

String sql = "update lib\_transaction set trn\_receive\_dt = sysdate() where trn\_mem\_id= " +

TxtMemberID.getText() + " and trn\_receive\_dt is null and trn\_book\_id=" + LblBookID.getText();

PreparedStatement pst=connection.prepareStatement(sql); pst.executeUpdate(sql);

sql = "update lib\_book\_master set book\_status='Available' where book\_id=" + LblBookID.getText(); pst=connection.prepareStatement(sql); pst.executeUpdate(sql);

JOptionPane.showMessageDialog(null,"Book Received");

ClearText();

} catch (Exception e) {

System.err.println(e);

System.exit(1);

}

}

private void ClearText()

{

TxtMemberID.setText("");

TxtMemberName.setText("");

TxtBookDetails.setText("");

LblBookID.setText("");

CmdReceiveBook.setEnabled(false);

}

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

this.setVisible(false);

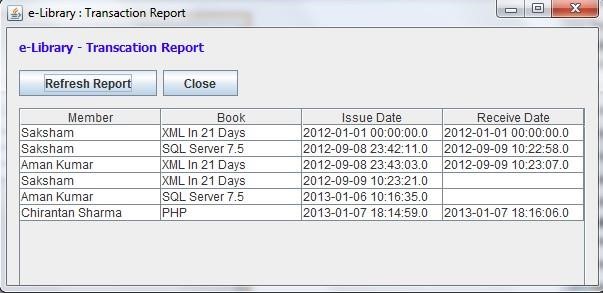
}

private void formWindowOpened(java.awt.event.WindowEvent evt) { ClearText(); this.setLocationRelativeTo(null);

}

**Transaction Report Form (TransactionReport.java) :**

public class TransactionReport extends javax.swing.JFrame {



**Source Cod**



**e :**



/\*\* Creates new form TransactionReport \*/ public TransactionReport() {

initComponents();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { // TODO add your handling code here: try {

MainClass mc=new MainClass();

Connection connection;

connection=DriverManager.getConnection(mc.StrUrl,mc.StrUid,mc.StrPwd);

ResultSet rs;

PreparedStatement stmt=connection.prepareStatement("SELECT m.mem\_name,b.book\_title, l.trn\_issue\_dt, l.trn\_receive\_dt FROM lib\_transaction l, lib\_book\_master b, lib\_member\_master m where l.trn\_book\_id=b.book\_id and l.trn\_mem\_id=m.mem\_id order by l.trn\_id"); rs = stmt.executeQuery();

jTable1.getColumnModel().getColumn(0).setHeaderValue("Member"); jTable1.getColumnModel().getColumn(1).setHeaderValue("Book"); jTable1.getColumnModel().getColumn(2).setHeaderValue("Issue Date"); jTable1.getColumnModel().getColumn(3).setHeaderValue("Receive Date"); jTable1.getTableHeader().resizeAndRepaint();

// Removing Previous Data while (jTable1.getRowCount() > 0) {

((DefaultTableModel) jTable1.getModel()).removeRow(0);

}

//Creating Object []rowData for jTable's Table Model int columns = rs.getMetaData().getColumnCount();

while (rs.next()) {

Object[] row = new Object[columns]; for (int i = 1; i <= columns; i++) {

row[i - 1] = rs.getObject(i); // 1

}

((DefaultTableModel) jTable1.getModel()).insertRow(rs.getRow() - 1,row);

}

} catch (Exception e) {

System.err.println(e);

//System.exit(1);

}

}

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

this.setVisible(false);

}

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

this.setLocationRelativeTo(null);

jTable1.getColumnModel().getColumn(0).setHeaderValue("Member");

jTable1.getColumnModel().getColumn(1).setHeaderValue("Book"); jTable1.getColumnModel().getColumn(2).setHeaderValue("Issue Date"); jTable1.getColumnModel().getColumn(3).setHeaderValue("Receive Date"); jTable1.getTableHeader().resizeAndRepaint();

// Removing Previous Data while (jTable1.getRowCount() > 0) {

((DefaultTableModel) jTable1.getModel()).removeRow(0);

} }

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() { public void run() {

new TransactionReport().setVisible(true);

}

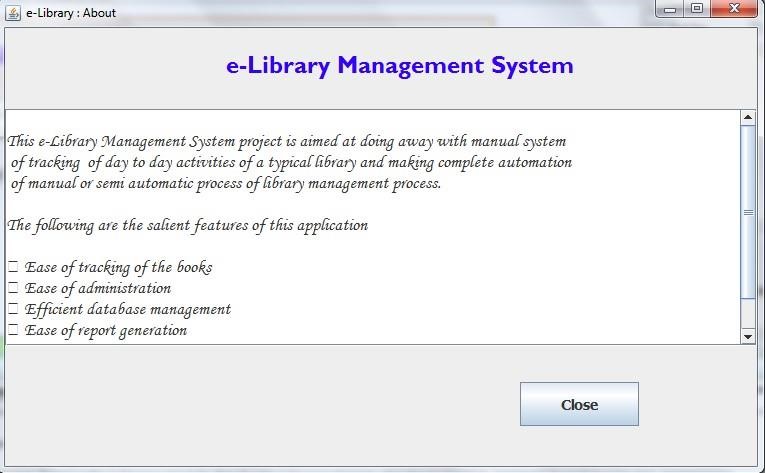
});

}

**ABOUT**

**FORM :**

**FUTURE SCOPE**



## FUTURE SCOPE OF APPLICATION

This application has flexibility to further enhance various functionalities. We can add new features as and when required. There is flexibility in various modules and re-usability of code is available.

## SOFTWARE SCOPE:

## Extensibility

The software is extensible through principles which enhance extensibility such as- hide data structures, avoid traversing multiple links and methods, avoid case statements on objects types and distinguish private and public operators

## Re-Usability

Reusability is possible in this application as and when required. Re-usable software reduces effort in design, coding and testing. Reducing the amount of code also simplifies the understanding, which increases the likelihood that code is correct

## Understandability

A method is understandable if a person other than the creator of the code can understand the code at any point in time in future.

## Cost Effectivness

The application has been developed in cost effective manner for the functionalities considered in the design.

# CONCLUSION

## ADVANTAGES OF e-LIBRARY MANAGEMENT SYSTEM

* Innovative and productive library operations

* Experience the efficiency and convenience of library management with the help of proven technology.

* Modular yet scalable deployment at preferred pace and schedule.

* Improved service provided to library patrons as a result of reducing time to borrow library materials, cutting down queue and queuing time, and ease of identifying the library materials required from the library.

* Cuts down laborious tasks.

* Smoothens business workflow and upgrades the library image to its library patrons.

* Doing More with Less

# BIBLIOGRAPHY

## BOOKS

1. Thinking in Java- Bruce Eckel

1. System Analysis And Design For Software Engineers- NIIT

1. The Essential Guide To User Interface Design: An Introduction To

GUI Design Principles And Techniques- Wilbert O Galitz

## WEB-SITES

1. netbeans.org/kb/docs/web/mysql-webapp.html