MY INFORMATICS PRACTICES PROJECT

2/10/2016

Library Management System

This is my Informatics Practices Project. It is a management system for libraries in school which is written in java language and data is handled in SQL Database.

ACKNOWLEDGEMENT

In the successful accomplishment of this project, many people have bestowed upon me their blessings and their hearts have pledged me a lot of support. I would like to thank all the people who have guided me and are directly or indirectly connected and concerned with this project. First and foremost I would like to thank my parents for providing me with their invaluable support in the completion of this project. Then I would like to thank my Respected Principal Sir Dr. Saibal Kumar Sanyal and my Physics teacher Mr. Arvind Tiwari, whose valuable guidance has helped me patch with this project and make it a success. Their suggestions and instructions have served as the major contribution towards the completion of this project. Then I would like to thank our creator, God, for helping me. Last but not the least I would like to thank my classmates for their guidance for helping me in the various phases of this project.

-Vijay Lalwani

Class XII Science

B.K Birla Centre for Education, Pune

Affiliated to Central Board of Secondary Education – New Delhi
(A Co-educational World Class Residential School)

Certificate

This is to certify that Master **VIJAY LALWANI** Reg no: **4646716** student of **class 12th Science,**
B.K Birla Centre For education has completed his project title "**Library Management System**" during the academic year 2015 -2016 towards partial fulfilment of credit for **IP Practical evaluation of CBSE AISSCE – 2016** and submitted a satisfactory report, as compiled in the following pages under my supervision.

Mr Manish Gupta.
Department of Computer.
B.K Birla Centre For Education.

Dr Saibal Sanyal.
Principal.
B.K Birla Centre For Education.

External Invigilation
Date:

CENTRAL BOARD OF SECONDARY EDUCATION (CBSE), INDIA

CONTENTS

INTRODUCTION

Once we understand the basics of Swing Java, we are ready to move to newer and bigger challenges. We can now develop full-fledged applications with Swing Java. Now the next question arises here is "What can we do with Swing Java?" I suggest that it will be more appropriate to ask that what we cannot do with Swing Java. And the answer to this question is - not much. From designing innovative user interface to taking advantages of other application objects, from manipulating text and graphics to working with database, Swing Java provides the tool that one will need to get job done right.

The purpose of this project is to make the library management more simple as of now days everything is done on computers from typing letters to taking man in space. So why to still use papers and cards to manage library when it can be done on computer and can be much simpler. This application can issue/return books, keep the information of all 12000 books and can do a lot more which will be described further. If more time was available there could be many more changes done to this project.

THIS PROJECT IS BASED ON THESE LANGUAGES

1. *Java:* Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2015, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

   The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

   The latest version is Java 8, which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis.

2. ***SQL:*** MySQL is an open-source relational database management system (RDBMS); in July 2013, it was the world's second most widely used RDBMS, and the most widely used open-source client–server model RDBMS. It is named after co-founder Michael Widenius's daughter, My. The SQL acronym stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

   MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL. Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr and YouTube.

   On all platforms except Windows, MySQL ships with no GUI tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or install MySQL Workbench via a separate download. Many third party GUI tools are also available.

APPLICATIONS USED TO MAKE THIS PROJECT

1. Netbeans 8.1: *NetBeans is a software development platform written in Java. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform, including the NetBeans integrated development environment (IDE), can be extended by third party developers.[3]*

The NetBeans IDE is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5.[4]

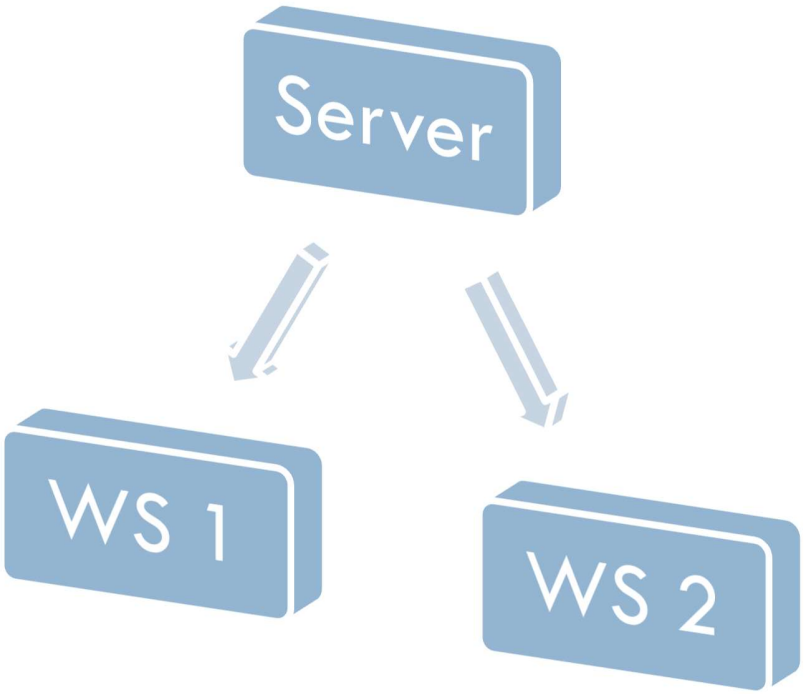
NetBeans is cross-platform and runs on Microsoft Windows, Mac OS X, Linux, Solaris and other platforms supporting a compatible JVM.

The NetBeans Team actively support the product and seek feature suggestions from the wider community. Every release is preceded by a time for Community testing and feedback

2. MySQL Workbench: *MySQL Workbench is the official integrated environment for MySQL. It was developed by MySQL AB, and enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, MySQL GUI Tools. Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).*

MySQL Workbench is available in two editions, the regular free and open source Community Edition which may be downloaded from the MySQL website, and the proprietary Standard Edition which extends and improves the feature set of the Community Edition.

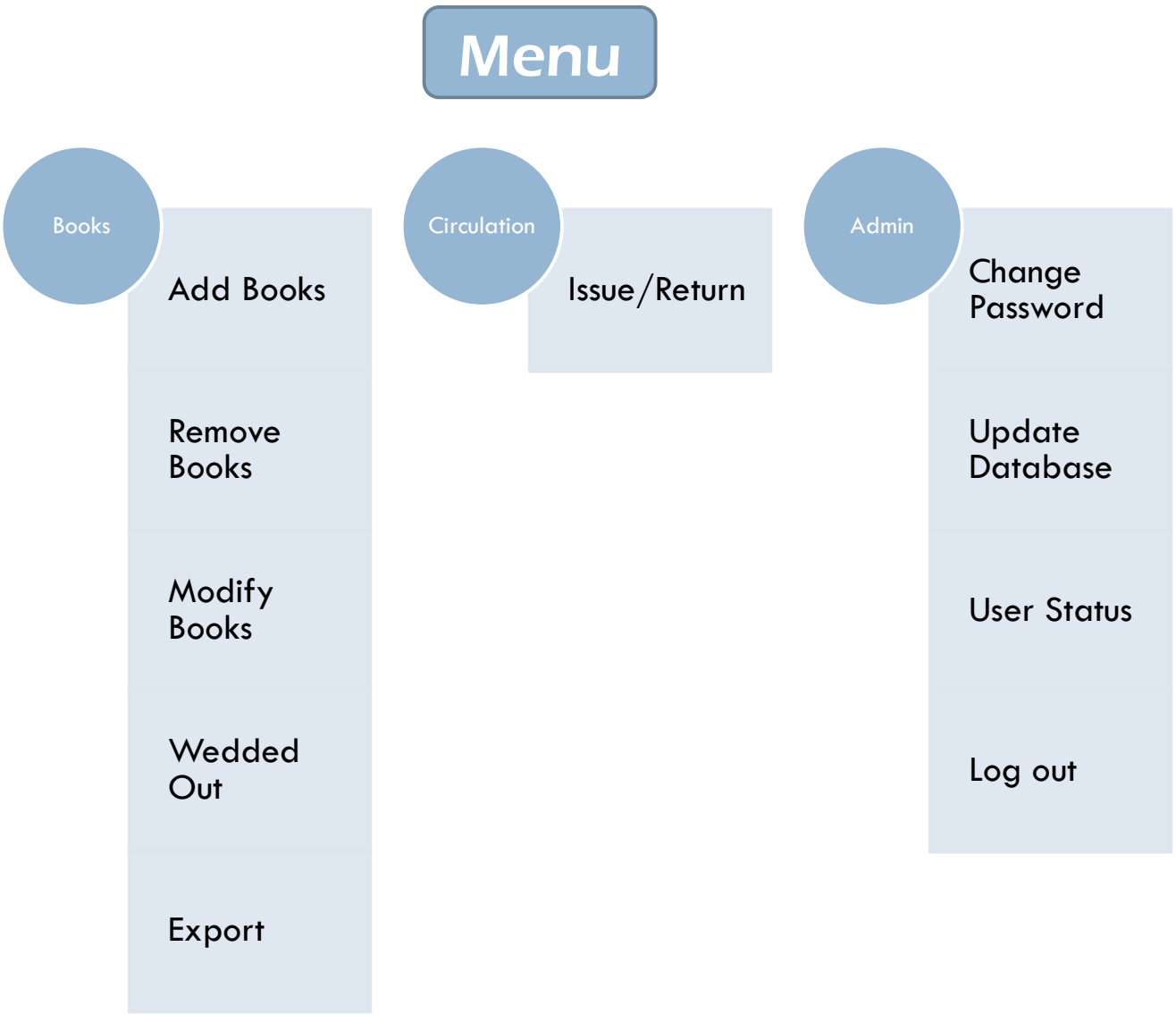
SERVER AND CLIENT INTERFACE



This is an example of star topology where all the work stations are connected to server.

LAYOUT FOR SERVER

APPLICATION LAYOUT



DATABASE LAYOUT

For table books and weddedout

Field	Type	Null	Key	Default	Extra
BOOK_GROUP	varchar(150)	YES		NULL	
BOOK_TITLE	varchar(150)	YES		NULL	
ACCESSION_NO	varchar(30)	NO	PRI	NULL	
AUTHOR_NAME	varchar(100)	YES		NULL	
EDITOR_NAME	varchar(100)	YES		NULL	
PUBLISHER	varchar(100)	YES		NULL	
EDITION	varchar(30)	YES		NULL	
PAGES	varchar(30)	YES		NULL	
ISBN_NO	varchar(30)	YES		NULL	
VENDOR_NAME	varchar(100)	YES		NULL	
LANGUAGE	varchar(30)	YES		NULL	
VOLUMES	varchar(30)	YES		NULL	
PURCHASE_DATE	varchar(30)	YES		NULL	
BILL_NO	varchar(30)	YES		NULL	
BILL_DATE	varchar(30)	YES		NULL	
PRICE_PER_COPY	varchar(30)	YES		NULL	
DDC_CODE	varchar(30)	YES		NULL	
DDC_DESCRIPTION	varchar(30)	YES		NULL	
BOOK_LOCATION	varchar(30)	YES		NULL	
DONATED	varchar(30)	YES		NULL	
SPECIMEN	varchar(30)	YES		NULL	
PUBLICATION_YEAR	varchar(30)	YES		NULL	
CURRENCY_DESC	varchar(30)	YES		NULL	
ISSUED	tinyint(1)	YES		0	
ADM_NO	varchar(7)	YES		NULL	

For table records

Field	Type	Null	Key	Default	Extra
ISSUE_DATE	date	YES		NULL	
RETURN_DATE	date	YES		NULL	
ADM_NO	varchar(7)	YES		NULL	
NAME	varchar(70)	YES		NULL	
BOOK_TITLE	varchar(150)	YES		NULL	
ACCESSION_NO	varchar(30)	YES		NULL	
USER_NO	varchar(2)	YES		NULL	

For table users

Field	Type	Null	Key	Default	Extra
id	varchar(20)	NO	PRI	NULL	
pass	varchar(30)	YES		NULL	

For table students

Field	Type	Null	Key	Default	Extra
ADM_NO	varchar(7)	NO	PRI		
NAME	varchar(70)	YES		NULL	
STD	varchar(10)	YES		NULL	

For table teachers

Field	Type	Null	Key	Default	Extra
SL_NO	varchar(2)	NO	PRI	NULL	
NAME	varchar(100)	YES		NULL	
INITIAL	char(3)	YES		NULL	
SUBJECT	varchar(60)	YES		NULL	

PRIMARY KEYS

Five Primary keys are used in this project they are as follows:

1. *Table Books (ACCESSION_NO)*
2. *Table weddedout (ACCESSION_NO)*
3. *Table Users (ID)*
4. *Table students (ADM_NO)*
5. *Table teachers (SL_NO)*

Primary keys are used in this project to avoid duplicate entries of books, students, teachers and users.

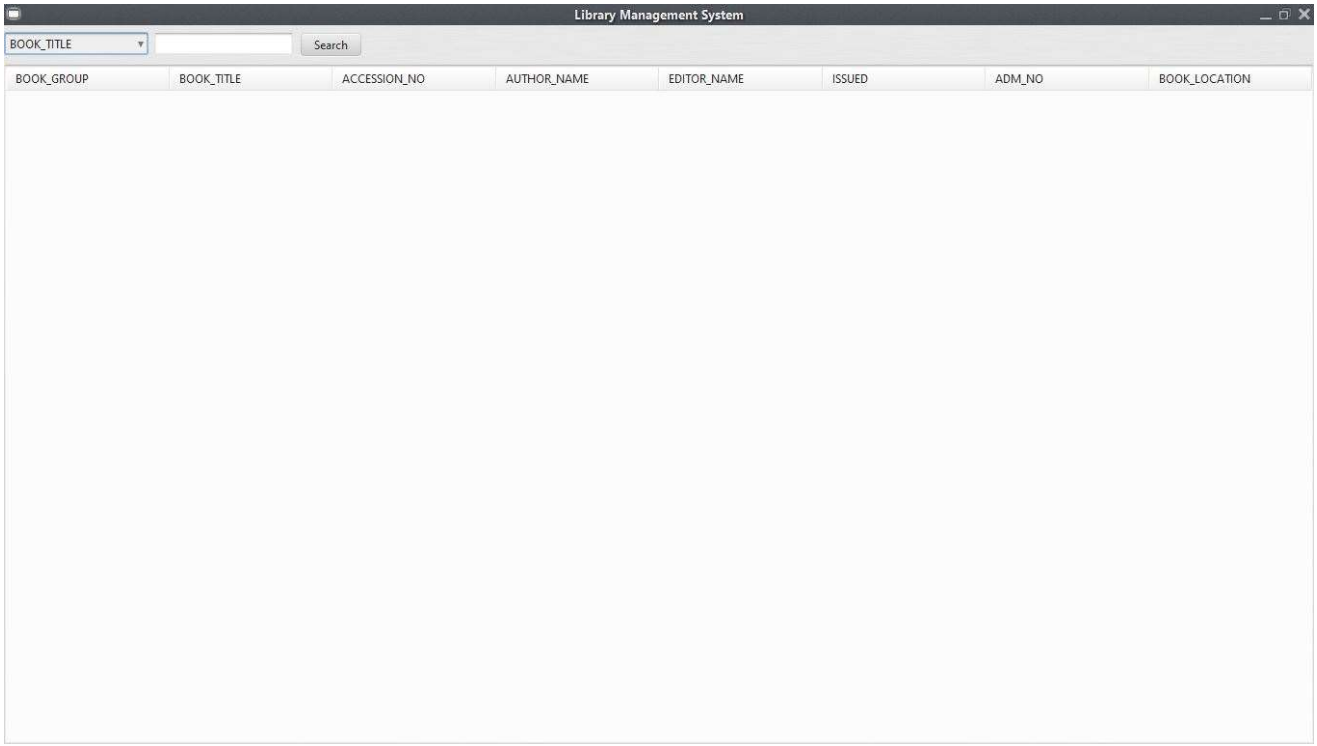
FOREIGN KEYS

There is only one foreign key used in this project and are as follows:

1. *Constraint FK_accession_no between table books and weddedout(ACESSION_NO)*

LAYOUT FOR CLIENT

APPLIATION LAYOUT



Only this frame is available on the application of client.

SERVER LAYOUT

Client Uses the MySQL database of server and can only access the books table.

Only read permission is given to the client which means it cannot modify or delete any entry from database by any means.

Flow diagrams

I. *For Issue/return of books*

II. *For Login*

III. *For removing and adding book again*

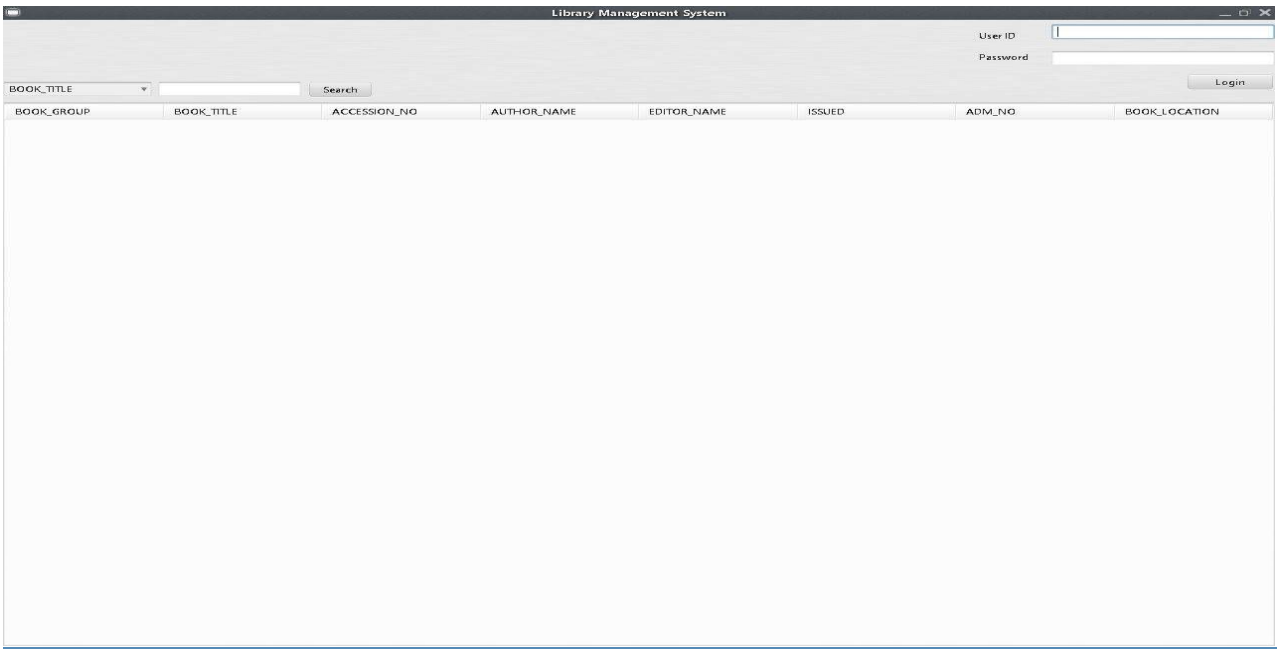
JAVA WORKING FOR SERVER

LIBRARIES

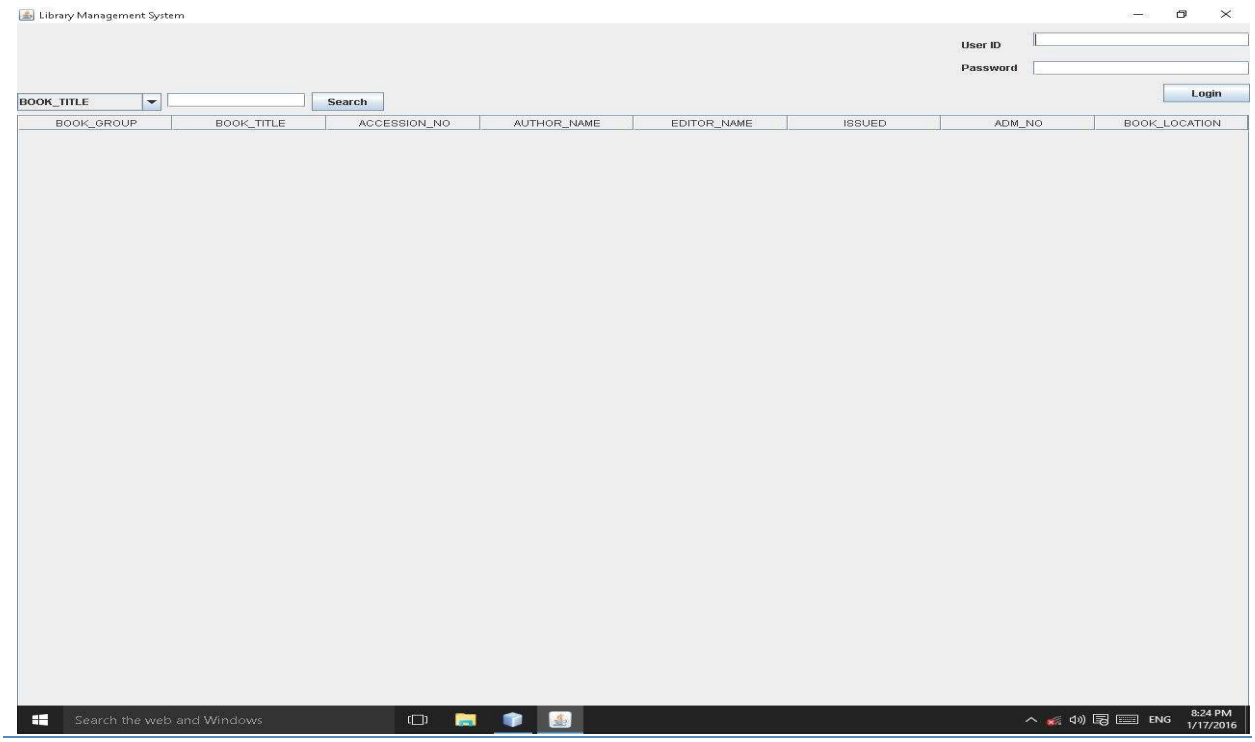
1. MySQL-connector-java - This library is used to connect java to MySQL so that the exchange of the data can take place. Example of this library is given below

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = (Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root", "mysqlip");
    Statement stmt = (Statement)conn.createStatement();
    String sql = "select * from books where " + item + " like'%" + text + "%';";
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        String BOOK_GROUP = rs.getString("BOOK_GROUP");
        String BOOK_TITLE = rs.getString("BOOK_TITLE");
        String ACCESSION_NO = rs.getString("ACCESSION_NO");
        String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
        String EDITOR_NAME = rs.getString("EDITOR_NAME");
        Boolean ISSUED = rs.getBoolean("ISSUED");
        String ADM_NO = rs.getString("ADM_NO");
        String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
        String ISSUE = ISSUED == true ? "Issued" : "Available";
        model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, ADM_NO, BOOK_LOCATION});
    }
    rs.close();
    conn.close();
    stmt.close();
}
catch (Exception e) {
    JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
}
```

2. *Synthetica and syntheticaAluOxide -* This Libraries is used to change the look and feel of the application. It makes the application look much better as you can see below



VS

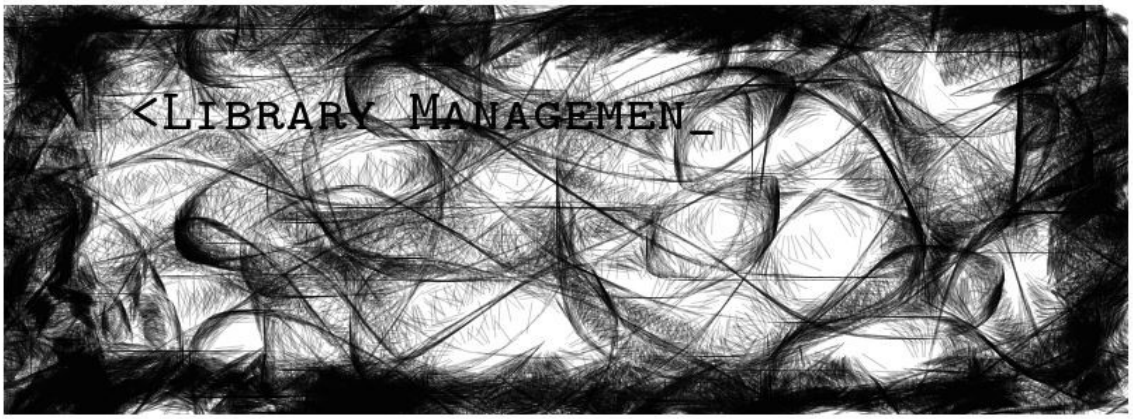


3. *Poi-3.12 and poi-ooxml-3.12 -* This Libraries is used to exchange data from excel files to java. Example of this libraries is given below

```
    private void StudentExportButtonActionPerformed(ActionEvent evt) {
        try {
            String excelFileName = "D:/Students.xls";
            String sheetName = "Sheet1";
            HSSFWorkbook wb = new HSSFWorkbook();
            HSSFSheet sheet = wb.createSheet(sheetName);
            HSSFRow row1 = sheet.createRow(0);
            HSSFCell cell = row1.createCell(0);
            cell.setCellType(1);
            cell.setCellValue("ADM_NO");
            HSSFCell cell1 = row1.createCell(1);
            cell1.setCellType(1);
            cell1.setCellValue("NAME");
            HSSFCell cell2 = row1.createCell(2);
            cell2.setCellType(1);
            cell2.setCellValue("STD");
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = (Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root", "mysql
            Statement stmt = (Statement)conn.createStatement();
            String sql = "select * from students";
            String sql1 = "select count(*) from students;";
            ResultSet rs1 = stmt.executeQuery(sql1);
            rs1.next();
            int TOTAL = rs1.getInt("count(*)");
            ResultSet rs = stmt.executeQuery(sql);
            rs.next();
            for (int j = 1; j <= TOTAL; ++j) {
                HSSFRow row = sheet.createRow(j);
                HSSFCell cell4 = row.createCell(0);
                cell4.setCellType(1);
                cell4.setCellValue(rs.getString("ADM_NO"));
                HSSFCell cell5 = row.createCell(1);
                cell5.setCellType(1);
                cell5.setCellValue(rs.getString("NAME"));
                HSSFCell cell6 = row.createCell(2);
                cell6.setCellType(1);
                cell6.setCellValue(rs.getString("STD"));
                rs.next();
            }
            stmt.close();
            conn.close();
            FileOutputStream fileOut = new FileOutputStream(excelFileName);
            wb.write((OutputStream)fileOut);
```

```
            fileOut.flush();
            fileOut.close();
            JOptionPane.showMessageDialog(null, "Exported Sucessfully");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
```

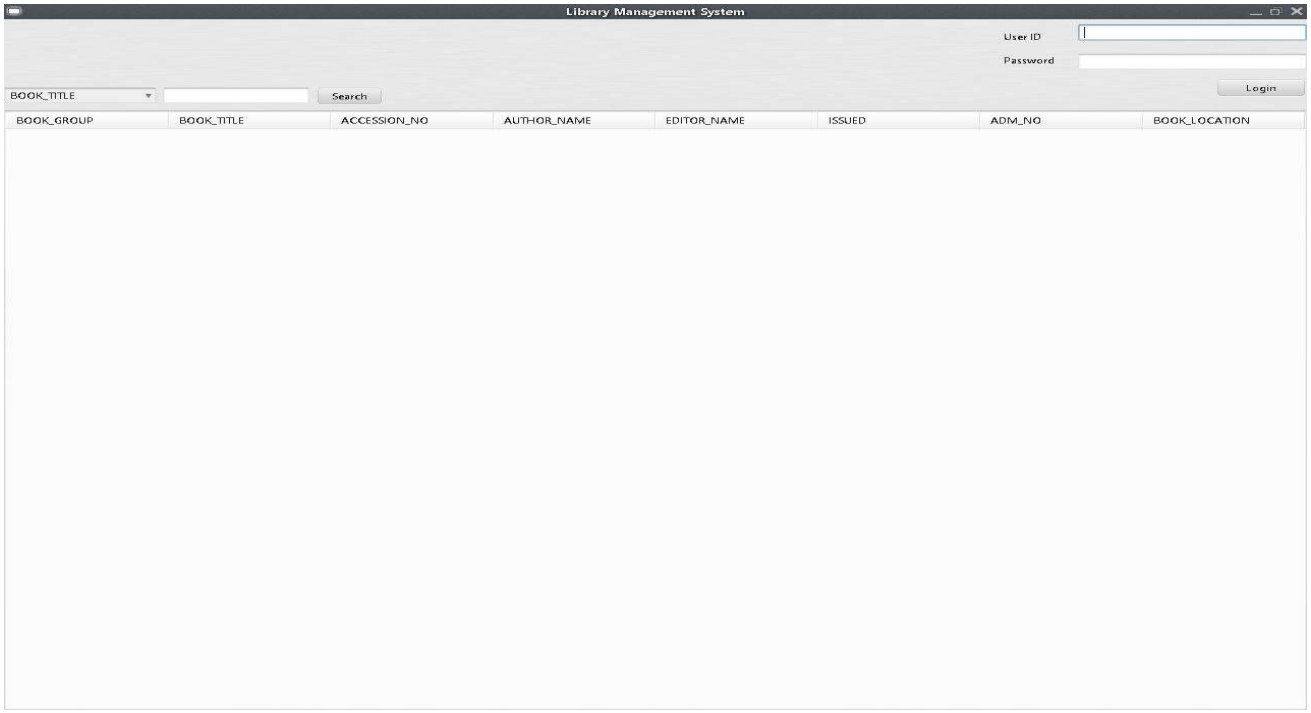
Splash Screen:



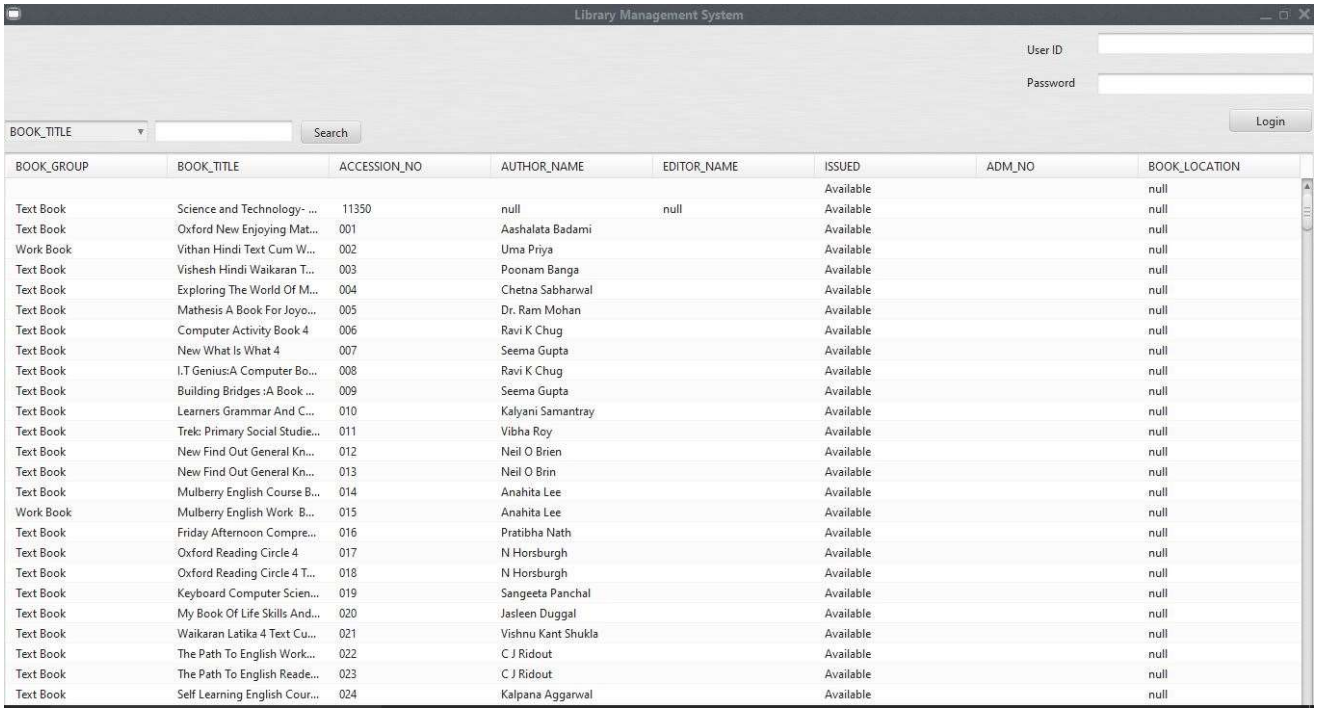
Code:

```
1   package library;
2
3   import de.javasoft.plaf.synthetica.SyntheticaAluOxideLookAndFeel;
4   import javax.swing.JOptionPane;
5   import javax.swing.JPanel;
6   import javax.swing.JWindow;
7   import javax.swing.LookAndFeel;
8   import javax.swing.UIManager;
9
10  public class Splash
11  extends JWindow {
12      private int duration;
13
14      public Splash(int d) {
15          duration = d;
16      }
17
18      public void showSplash() {
19          Thread t = new Thread();
20          JPanel content = (JPanel)getContentPane();
21          try {
22              Thread.sleep(1800);
23          }
24          catch (InterruptedException ex) {
25              JOptionPane.showMessageDialog(null, "Please try restarting the program");
26          }
27      }
28
29      public void showSplashAndExit() {
30          showSplash();
31          new OpenPage().setVisible(true);
32      }
33
34      public static void main(String[] args) {
35          try {
36              UIManager.setLookAndFeel((LookAndFeel)new SyntheticaAluOxideLookAndFeel());
37          }
39          catch (Exception ex) {
40              JOptionPane.showMessageDialog(null, ex.getLocalizedMessage());
41          }
42          Splash splash = new Splash(100);
43          splash.showSplashAndExit();
44      }
45  }
```

Homepage (Openpage):



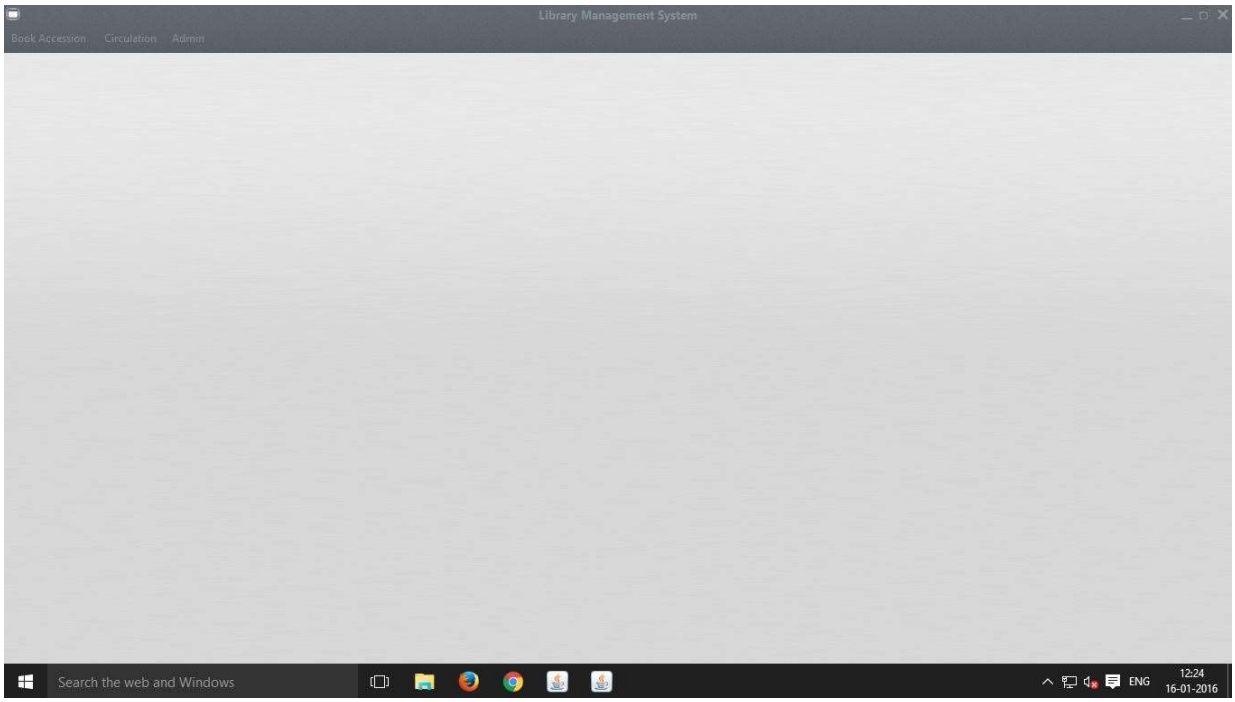
Search Button:



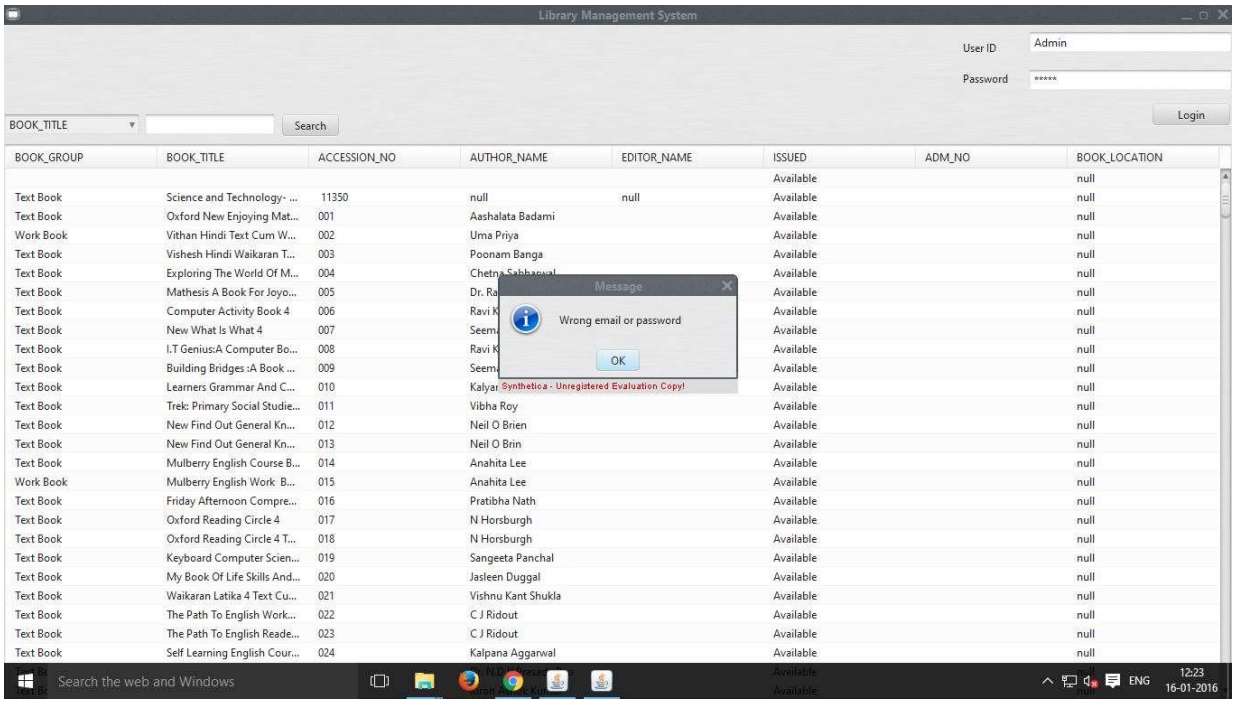
BOOK_GROUP	BOOK_TITLE	ACCESSION_NO	AUTHOR_NAME	EDITOR_NAME	ISSUED	ADM_NO	BOOK_LOCATION
					Available		null
Text Book	Science and Technology- ...	11350	null	null	Available		null
Text Book	Oxford New Enjoying Mat...	001	Aashalata Badami		Available		null
Work Book	Vithan Hindi Text Cum W...	002	Uma Priya		Available		null
Text Book	Vishesh Hindi Waikaran T...	003	Poonam Banga		Available		null
Text Book	Exploring The World Of M...	004	Chetna Sabharwal		Available		null
Text Book	Mathesis A Book For Joyo...	005	Dr. Ram Mohan		Available		null
Text Book	Computer Activity Book 4	006	Ravi K Chug		Available		null
Text Book	New What Is What 4	007	Seema Gupta		Available		null
Text Book	I.T Genius:A Computer Bo...	008	Ravi K Chug		Available		null
Text Book	Building Bridges :A Book ...	009	Seema Gupta		Available		null
Text Book	Learners Grammar And C...	010	Kalyani Samantray		Available		null
Text Book	Trek: Primary Social Studie...	011	Vibha Roy		Available		null
Text Book	New Find Out General Kn...	012	Neil O Brien		Available		null
Text Book	New Find Out General Kn...	013	Neil O Brin		Available		null
Text Book	Mulberry English Course B...	014	Anahita Lee		Available		null
Work Book	Mulberry English Work B...	015	Anahita Lee		Available		null
Text Book	Friday Afternoon Compre...	016	Pratibha Nath		Available		null
Text Book	Oxford Reading Circle 4	017	N Horsburgh		Available		null
Text Book	Oxford Reading Circle 4 T...	018	N Horsburgh		Available		null
Text Book	Keyboard Computer Scien...	019	Sangeeta Panchal		Available		null
Text Book	My Book Of Life Skills And...	020	Jasleen Duggal		Available		null
Text Book	Waikaran Latika 4 Text Cu...	021	Vishnu Kant Shukla		Available		null
Text Book	The Path To English Work...	022	C J Ridout		Available		null
Text Book	The Path To English Reade...	023	C J Ridout		Available		null
Text Book	Self Learning English Cour...	024	Kalpana Aggarwal		Available		null

Login:

If Successful: the frame below opens



If not:



BOOK_GROUP	BOOK_TITLE	ACCESSION_NO	AUTHOR_NAME	EDITOR_NAME	ISSUED	ADM_NO	BOOK_LOCATION
					Available		null
Text Book	Science and Technology- ...	11350	null	null	Available		null
Text Book	Oxford New Enjoying Mat...	001	Aashalata Badami		Available		null
Work Book	Vithan Hindi Text Cum W...	002	Uma Priya		Available		null
Text Book	Vishesh Hindi Waikaran T...	003	Poonam Banga		Available		null
Text Book	Exploring The World Of M...	004	Chetna Sabharwal		Available		null
Text Book	Mathesis A Book For Joyo...	005	Dr. Ra		Available		null
Text Book	Computer Activity Book 4	006	Ravi K		Available		null
Text Book	New What Is What 4	007	Seem		Available		null
Text Book	I.T Genius:A Computer Bo...	008	Ravi K		Available		null
Text Book	Building Bridges :A Book ...	009	Seem		Available		null
Text Book	Learners Grammar And C...	010	Kalyan		Available		null
Text Book	Trek: Primary Social Studie...	011	Vibha Roy		Available		null
Text Book	New Find Out General Kn...	012	Neil O Brien		Available		null
Text Book	New Find Out General Kn...	013	Neil O Brin		Available		null
Text Book	Mulberry English Course B...	014	Anahita Lee		Available		null
Work Book	Mulberry English Work B...	015	Anahita Lee		Available		null
Text Book	Friday Afternoon Compre...	016	Pratibha Nath		Available		null
Text Book	Oxford Reading Circle 4	017	N Horsburgh		Available		null
Text Book	Oxford Reading Circle 4 T...	018	N Horsburgh		Available		null
Text Book	Keyboard Computer Scien...	019	Sangeeta Panchal		Available		null
Text Book	My Book Of Life Skills And...	020	Jasleen Duggal		Available		null
Text Book	Waikaran Latika 4 Text Cu...	021	Vishnu Kant Shukla		Available		null
Text Book	The Path To English Work...	022	C J Ridout		Available		null
Text Book	The Path To English Reade...	023	C J Ridout		Available		null
Text Book	Self Learning English Cour...	024	Kalpana Aggarwal		Available		null

Code:

Search method:

```
152    int Search(String item, String text, DefaultTableModel model) {
153        while (model.getRowCount() > 0) {
154            model.setRowCount(0);
155        }
156        try {
157            Class.forName("com.mysql.jdbc.Driver");
               Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
159            Statement stmt = (Statement)conn.createStatement();
160            String sql = "select * from books where " + item + " like'%" + text + "%';";
161            ResultSet rs = stmt.executeQuery(sql);
162            while (rs.next()) {
163                String BOOK_GROUP = rs.getString("BOOK_GROUP");
164                String BOOK_TITLE = rs.getString("BOOK_TITLE");
165                String ACCESSION_NO = rs.getString("ACCESSION_NO");
166                String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
167                String EDITOR_NAME = rs.getString("EDITOR_NAME");
168                Boolean ISSUED = rs.getBoolean("ISSUED");
169                String ADM_NO = rs.getString("ADM_NO");
170                String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
171                String ISSUE = ISSUED == true ? "Issued" : "Available";
172                model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, ADM_NO, BOOK_LOCATION});
173            }
174            rs.close();
175            conn.close();
176            stmt.close();
177        }
               catch (Exception e) {
179            JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
180            return -1;
181        }
182        return 0;
183    }
```

Search Button:

```
199    private void SearchButtonActionPerformed(ActionEvent evt) {
200        String search = (String)this.SearchBox.getSelectedItem();
201        String text = this.SearchTextField.getText();
202        DefaultTableModel model = (DefaultTableModel)this.SearchResult.getModel();
203        this.Search(search, text, model);
204    }
205
206    private void SearchTextFieldKeyPressed(KeyEvent evt) {
207        if (evt.getKeyCode() == 10) {
208            String search = (String)this.SearchBox.getSelectedItem();
209            String text = this.SearchTextField.getText();
210            DefaultTableModel model = (DefaultTableModel)this.SearchResult.getModel();
211            this.Search(search, text, model);
212        }
213    }
```

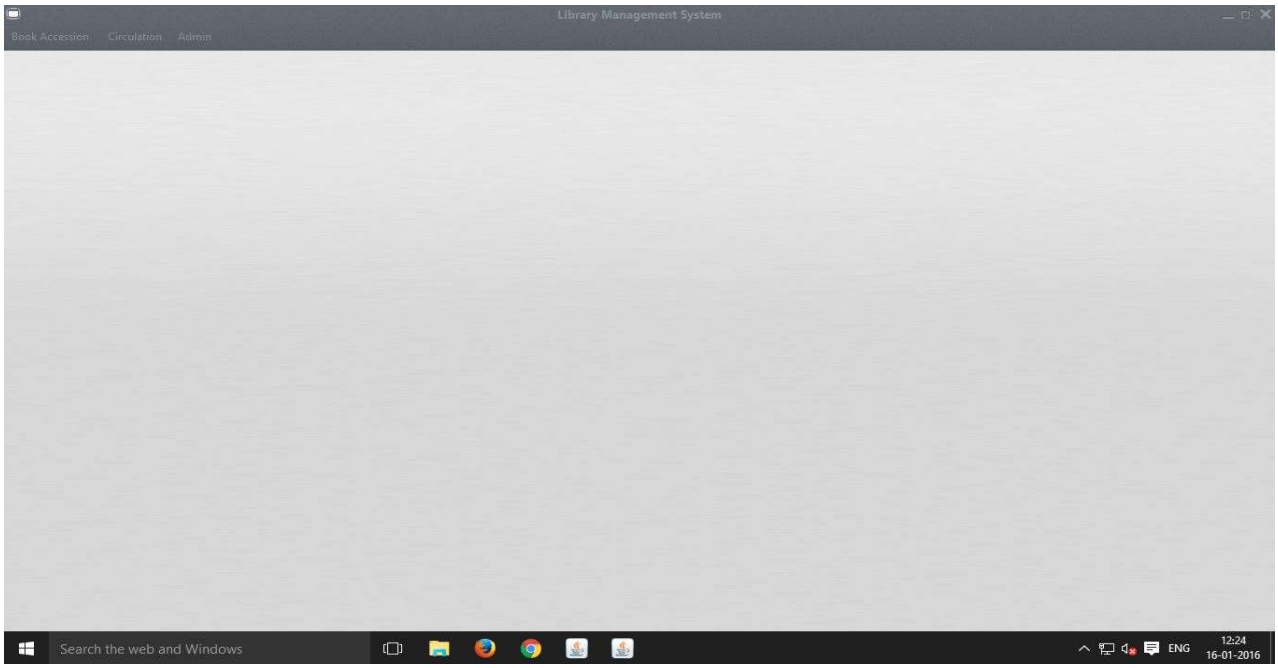
Login Method:

```
124
125     int Login(String id, String pass) {
126         try {
127             Class.forName("com.mysql.jdbc.Driver");
128             Connection con = (Connection)DriverManager.getConnection(db_url,user,pwd);
129             Statement stmt = (Statement)con.createStatement();
130             String sql = "select * from users;";
131             ResultSet rs = stmt.executeQuery(sql);
132             rs.next();
133             String a = rs.getString("id");
134             String b = rs.getString("pass");
135             rs.close();
136             con.close();
137             stmt.close();
138             if (id.equals(a) && pass.equals(b)) {
139                 new Main().setVisible(true);
140                 this.dispose();
141             } else {
142                 JOptionPane.showMessageDialog(null, "Wrong email or password");
143             }
144         }
145         catch (Exception e) {
146             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
147             return -1;
148         }
149         return 0;
150     }
```

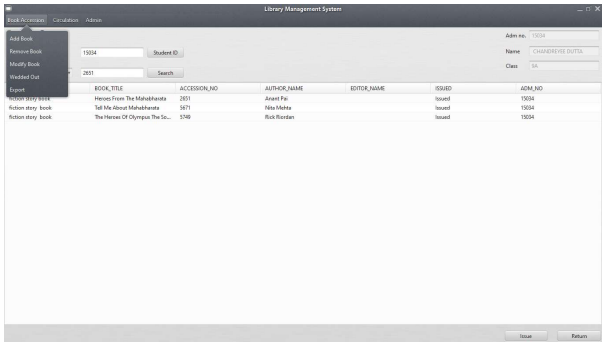
Login Button:

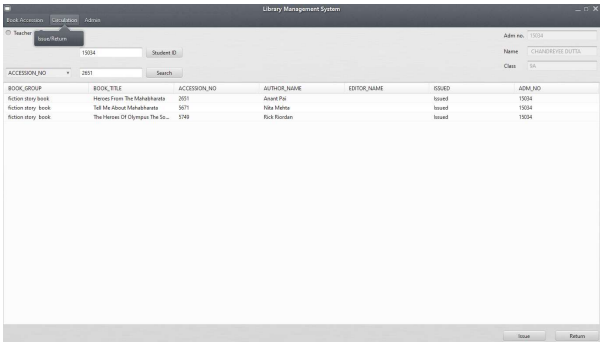
Main page:

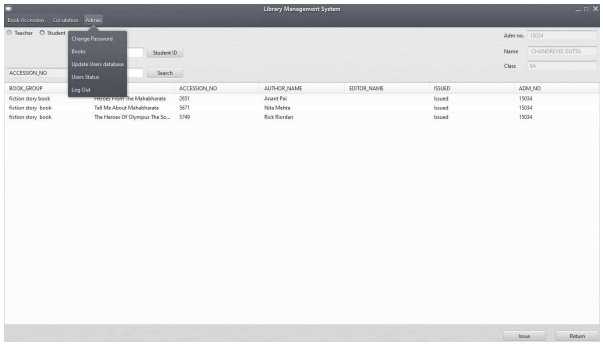
This page actually doesn't contain anything but you can navigate to anywhere through this page-



This are all the menu options:





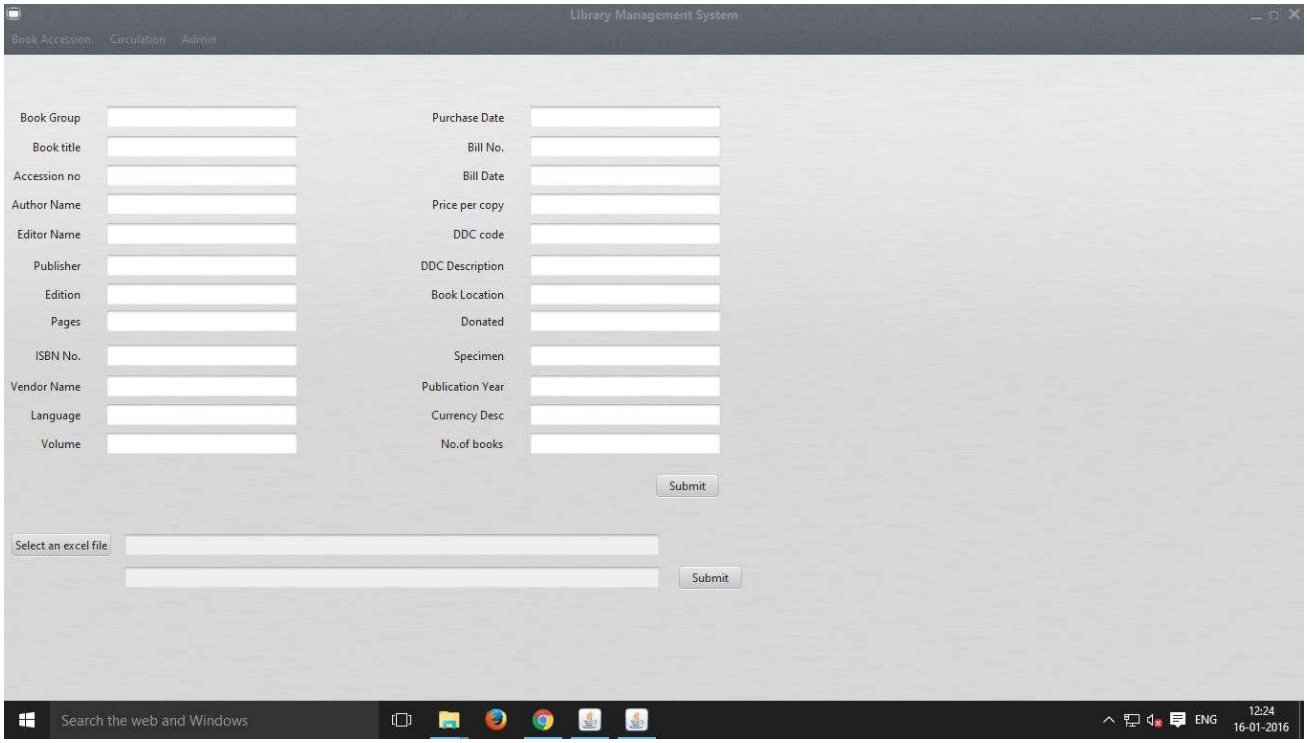


Code:

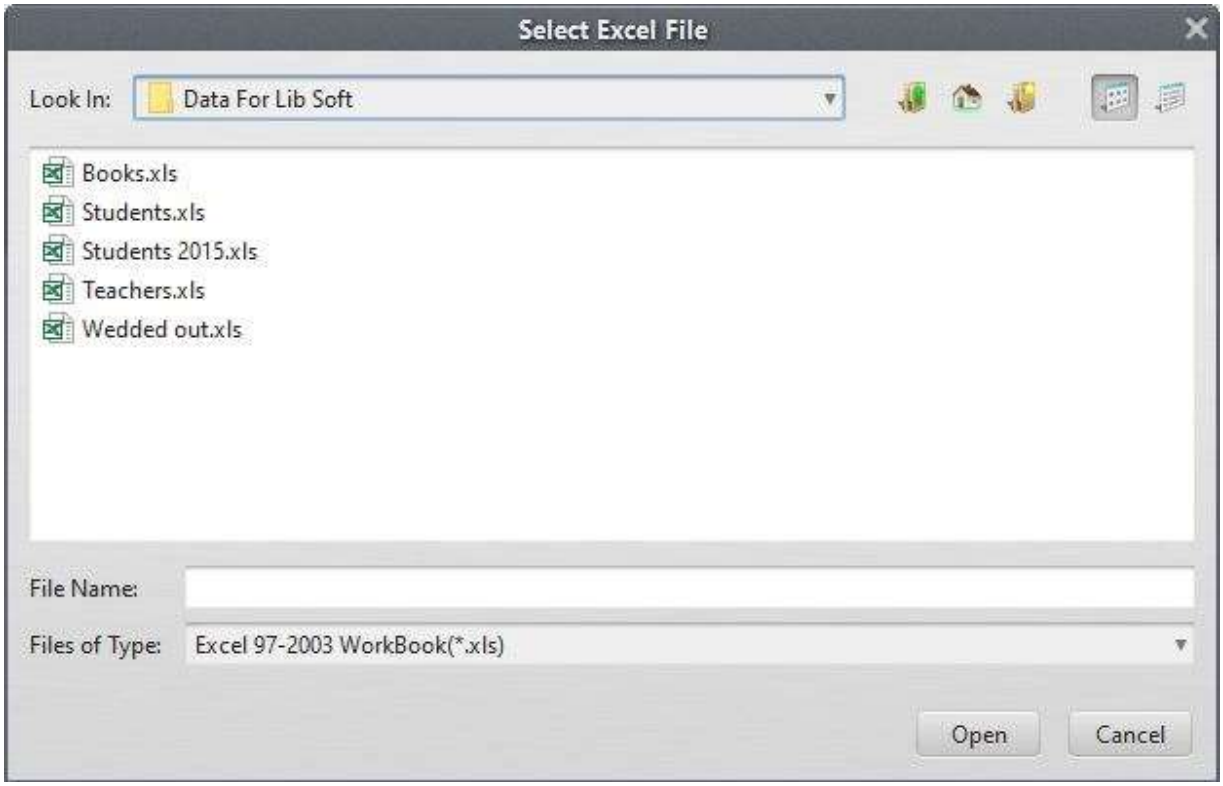
This is the menu bar code and will be common in every frame

```
508     //Menu actions
509     private void AddBookMenuItem2ActionPerformed(ActionEvent evt) {
510         new AddBook().setVisible(true);
511         this.dispose();
512     }
513     private void RemoveBookMenuItemActionPerformed(ActionEvent evt) {
514         new RemoveBook().setVisible(true);
515         this.dispose();
516     }
517     private void ModifyBookMenuItemActionPerformed(ActionEvent evt) {
518         new ModifyBook().setVisible(true);
519         this.dispose();
520     }
521     private void WeddedoutMenuItemActionPerformed(ActionEvent evt) {
522         new WeddedOut().setVisible(true);
523         this.dispose();
524     }
525     private void ExportMenuItemActionPerformed(ActionEvent evt) {
526         new Export().setVisible(true);
527         this.dispose();
528     }
529     private void IssueMenuItemActionPerformed(ActionEvent evt) {
530         new Issue().setVisible(true);
531         this.dispose();
532     }
533     private void CirculationMenuActionPerformed(ActionEvent evt) {
534         new Issue().setVisible(true);
535         this.dispose();
536     }
537     private void ChangePasswordMenuItemActionPerformed(ActionEvent evt) {
538         new ChangePassword().setVisible(true);
539         this.dispose();
540     }
541     private void BooksMenuItemActionPerformed(ActionEvent evt) {
542         new Books().setVisible(true);
543         this.dispose();
544     }
545     private void UpdateDatabaseMenuItemActionPerformed(ActionEvent evt) {
546         new UpdateDatabases().setVisible(true);
547         this.dispose();
548     }
549     private void UserStatusMenuItemActionPerformed(ActionEvent evt) {
550         new UsersData().setVisible(true);
551         this.dispose();
552     }
552     }
553     private void LogOutMenuItemActionPerformed(ActionEvent evt) {
554         new OpenPage().setVisible(true);
555         this.dispose();
556     }
557     //menu actions ends
```

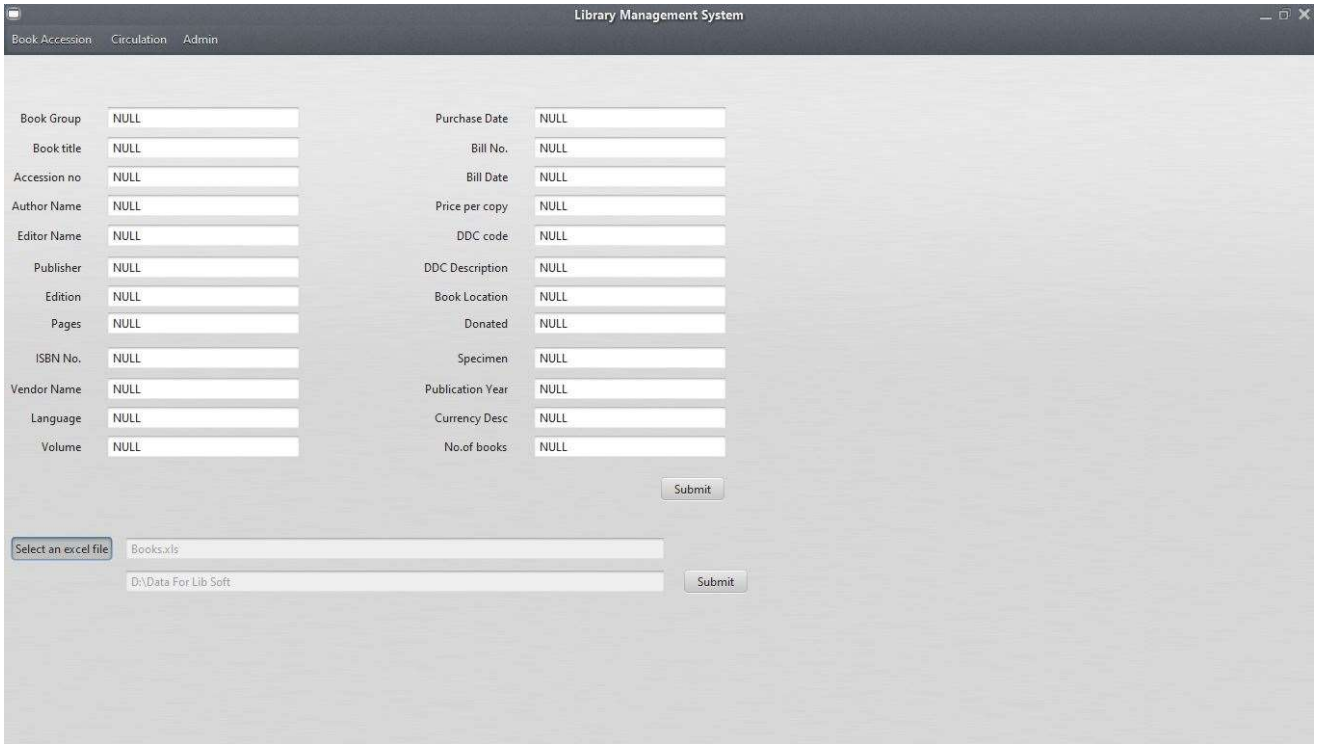
AddBook:



Import Dialog:

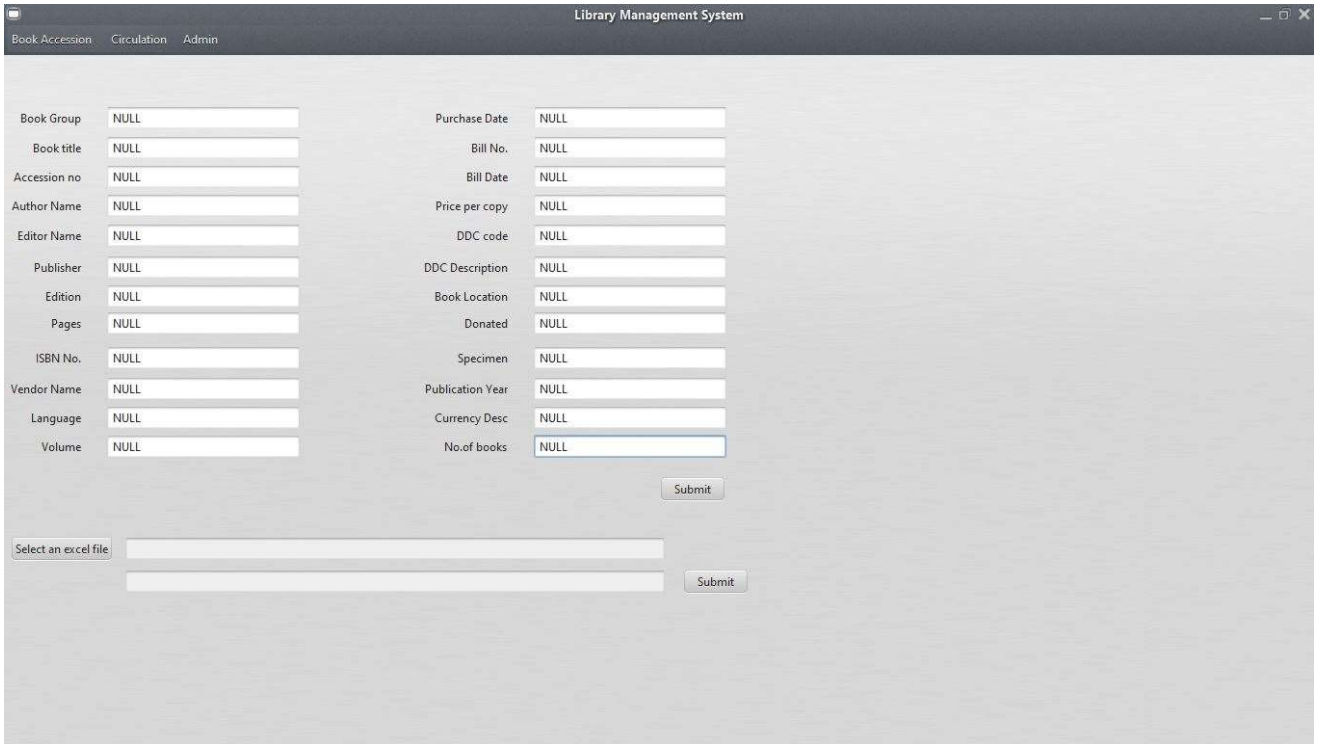


Imported Successfully:



Excel file Added Successfully:

Adding Single Book:



Single Book Added:

CODE:

Imports:

```
3  import com.mysql.jdbc.Connection;
4  import com.mysql.jdbc.Statement;
5  import java.awt.Cursor;
6  import java.awt.EventQueue;
7  import java.awt.event.ActionEvent;
8  import java.io.File;
9  import java.io.FileInputStream;
10 import java.io.InputStream;
11 import java.sql.DriverManager;
12 import java.util.Iterator;
13 import javax.swing.GroupLayout;
14 import javax.swing.JButton;
15 import javax.swing.JFileChooser;
16 import javax.swing.JFrame;
17 import javax.swing.JLabel;
18 import javax.swing.JMenu;
19 import javax.swing.JMenuBar;
20 import javax.swing.JMenuItem;
21 import javax.swing.JOptionPane;
22 import javax.swing.JTextField;
23 import javax.swing.JToggleButton;
24 import javax.swing.LayoutStyle;
25 import javax.swing.filechooser.FileNameExtensionFilter;
26 import org.apache.poi.hssf.usermodel.HSSFCell;
27 import org.apache.poi.hssf.usermodel.HSSFRow;
28 import org.apache.poi.hssf.usermodel.HSSFSheet;
29 import org.apache.poi.hssf.usermodel.HSSFWorkbook;
30 import org.apache.poi.xssf.usermodel.XSSFCell;
31 import org.apache.poi.xssf.usermodel.XSSFRow;
32 import org.apache.poi.xssf.usermodel.XSSFSheet;
33 import org.apache.poi.xssf.usermodel.XSSFWorkbook;
```

Submit Button:

```
private void SubmitButtonActionPerformed(ActionEvent evt) {
        String[] ARRAY = new String[22];
        ARRAY[0]=jTextField1.getText();
        ARRAY[1]=jTextField2.getText();
        ARRAY[2]=jTextField3.getText();
        ARRAY[3]=jTextField4.getText();
        ARRAY[4]=jTextField5.getText();
        ARRAY[5]=jTextField6.getText();
        ARRAY[6]=jTextField7.getText();
        ARRAY[7]=jTextField8.getText();
        ARRAY[8]=jTextField9.getText();
        ARRAY[9]=jTextField10.getText();
        ARRAY[10]=jTextField11.getText();
        ARRAY[11]=jTextField12.getText();
        ARRAY[12]=jTextField13.getText();
        ARRAY[13]=jTextField14.getText();
        ARRAY[14]=jTextField15.getText();
        ARRAY[15]=jTextField16.getText();
        ARRAY[16]=jTextField17.getText();
        ARRAY[17]=jTextField18.getText();
        ARRAY[18]=jTextField19.getText();
        ARRAY[19]=jTextField20.getText();
        ARRAY[20]=jTextField21.getText();
        ARRAY[21]=jTextField22.getText();
        ARRAY[22]=jTextField23.getText();
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
            Statement stmt = (Statement)conn.createStatement();
            String sql = "insert into books (BOOK_GROUP,BOOK_TITLE,ACCESSION_NO,AUTHOR_NAME,EDITOR_NAME,PUBLISHER,EDITION,PAGES,"
                    + "ISBN_NO,VENDOR_NAME,LANGUAGE,VOLUMES,PURCHASE_DATE,BILL_NO,BILL_DATE,PRICE_PER_COPY,DDC_CODE,DDC_DESCRIPTION,"
                    + "BOOK_LOCATION,DONATED,SPECIMEN,PUBLICATION_YEAR,CURRENCY_DESC) values('" + ARRAY[0] + "','" + ARRAY[1] + "','"
                    + "'" + ARRAY[2] + "','" + ARRAY[3] + "','" + ARRAY[4] + "','" + ARRAY[5] + "','" + ARRAY[6] + "','" + ARRAY[7] + "','"
                    + "'" + ARRAY[8] + "','" + ARRAY[9] + "','" + ARRAY[10] + "','" + ARRAY[11] + "','" + ARRAY[12] + "','" + ARRAY[13] + ""
                    + "','" + ARRAY[14] + "','" + ARRAY[15] + "','" + ARRAY[16] + "','" + ARRAY[17] + "','" + ARRAY[18] + "','"
                    + "'" + ARRAY[19] + "','" + ARRAY[20] + "','" + ARRAY[21] + "','" + ARRAY[22] + "');";
            stmt.executeUpdate(sql);
            stmt.close();
            conn.close();
            JOptionPane.showMessageDialog(null, "Added Sucessfully");
        }
        catch (Exception e) {
            JOptionPane.showMessageDialog(null,e.getLocalizedMessage());
        }
    }
```

Select Excel File Button:

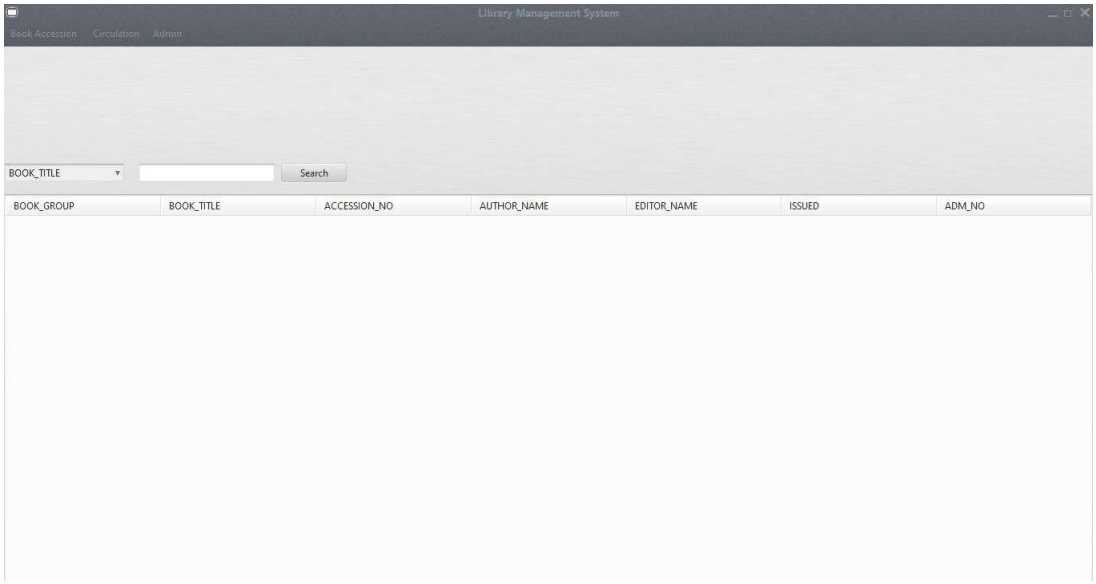
```
465     private void SelectEXCELButtonActionPerformed(ActionEvent evt) {
466         JFileChooser fileChooser = new JFileChooser("c://");
467         fileChooser.setDialogTitle("Select Excel File");
468         fileChooser.setFileSelectionMode(0);
469         fileChooser.setFileFilter(new FileNameExtensionFilter("All Files(*.)",""));
470         fileChooser.setFileFilter(new FileNameExtensionFilter("Excel WorkBook(*.xlsx)", "xlsx"));
471         fileChooser.setFileFilter(new FileNameExtensionFilter("Excel 97-2003 WorkBook(*.xls)", "xls"));
472         fileChooser.setAcceptAllFileFilterUsed(true);
473         int result = fileChooser.showOpenDialog(this);
474         if (result == 0) {
475             this.filename.setText(fileChooser.getSelectedFile().getName());
476             this.dir.setText(fileChooser.getCurrentDirectory().toString());
477             this.abcd = fileChooser.getCurrentDirectory().toString() + "\\" + fileChooser.getSelectedFile().getName();
478             this.ext = fileChooser.getTypeDescription(fileChooser.getSelectedFile());
479         }
480         if (result == 1) {
481             this.filename.setText("You pressed cancel");
482             this.dir.setText("");
483         }
484     }
```

Submit Excel File Button:

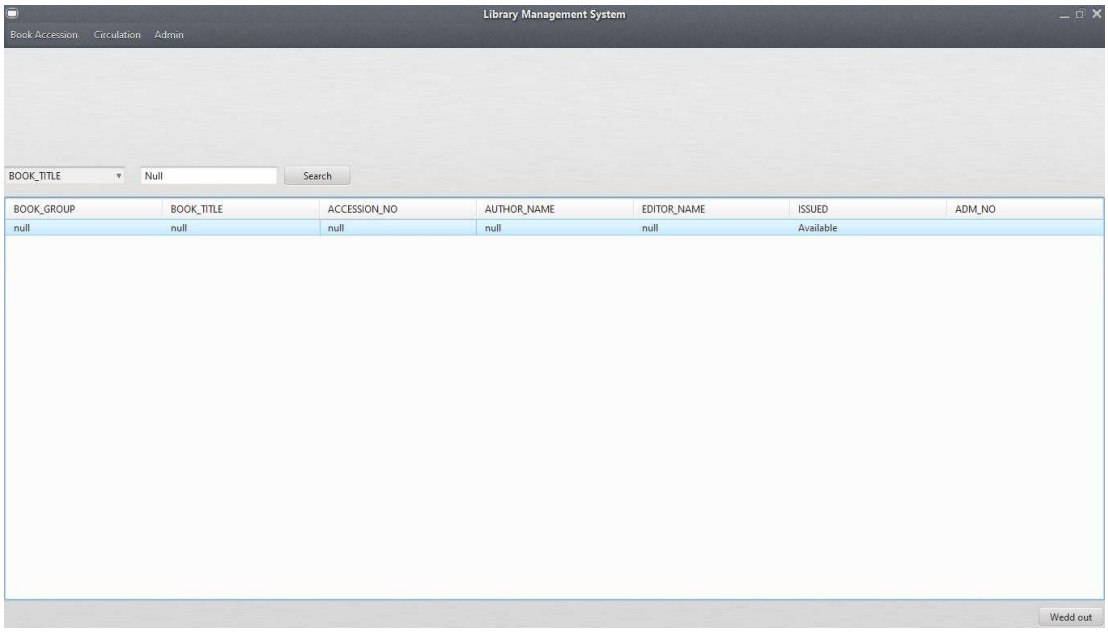
```
352     private void SubmitEXCELButtonActionPerformed(ActionEvent evt) {
353         if (ext.equals("Microsoft Excel Worksheet")) {
354             try {
355                 FileInputStream file = new FileInputStream(new File(this.abcd));
356                 XSSFWorkbook yourworkbook = new XSSFWorkbook((InputStream)file);
357                 XSSFSheet sheet1 = yourworkbook.getSheetAt(0);
358                 Iterator rows = sheet1.rowIterator();
                    XSSFRow row = (XSSFRow)rows.next();
360                 int rowNum = sheet1.getLastRowNum();
361                 short colNum = sheet1.getRow(0).getLastCellNum();
362                 int j = 1;
363                 Class.forName("com.mysql.jdbc.Driver");
364                 Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
365                 Statement stmt = (Statement)conn.createStatement();
366                 String Sql1 = "truncate books;";
367                 stmt.executeUpdate(Sql1);
368                 for (int i = 0; i < rowNum; ++i) {
369                     String[] array = new String[colNum];
370                     boolean[] array1 = new boolean[colNum];
371                     int a = 0;
372                     try {
373                         while (rows.hasNext())
374                         {
375                             row = sheet1.getRow(j);
376                             XSSFCell column = row.getCell(a);
                                column.setCellType(column.CELL_TYPE_STRING);
                                if (column.getCellType() == column.CELL_TYPE_STRING)
379                             {
380                                 array[a] = column.getStringCellValue();
381                             }
                                else if (column.getCellType() == column.CELL_TYPE_BOOLEAN)
383                             {
384                                 array1[a] = column.getBooleanCellValue();
385                             }
                                else if (column.getCellType() == column.CELL_TYPE_BLANK)
387                             {
388                                 array[a] = "";
389                             }
390                             ++a;
391                         }
392                     }
393                     catch (NullPointerException column) {
394                         // empty catch block
395                     }
396                     String sql = "Insert into books(BOOK_GROUP,BOOK_TITLE,ACCESSION_NO,AUTHOR_NAME,EDITOR_NAME,PUBLISHER,EDITION,"
397                             + "PAGES,ISBN_NO,VENDOR_NAME,LANGUAGE,VOLUMES,PURCHASE_DATE,BILL_NO,BILL_DATE,PRICE_PER_COPY,DDC_CODE,"
398                             + "DDC_DESCRIPTION,BOOK_LOCATION,DONATED,SPECIMEN,PUBLICATION_YEAR,CURRENCY_DESC,ISSUED,ADM_NO) values"
399                             + " ('" + array[0] + "','" + array[1] + "','" + array[2] + "','" + array[3] + "','" + array[4] + "','"
400                             + "'" + array[5] + "','" + array[6] + "','" + array[7] + "','" + array[8] + "','" + array[9] + "','"
401                             + "'" + array[10] + "','" + array[11] + "','" + array[12] + "','" + array[13] + "','" + array[14] + "','"
402                             + "'" + array[15] + "','" + array[16] + "','" + array[17] + "','" + array[18] + "','" + array[19] + "','"
403                             + "'" + array[20] + "','" + array[21] + "','" + array[22] + "'," + array1[23] + ",'" + array[24] + "');";
404                     stmt.executeUpdate(sql);
405                 }
406             }
                catch (Exception e) {
408                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
409             }
410         } else if (ext.equals("Microsoft Excel 97-2003 Worksheet")) {
411             try {
412                 FileInputStream file = new FileInputStream(new File(this.abcd));
413                 HSSFWorkbook yourworkbook = new HSSFWorkbook((InputStream)file);
414                 HSSFSheet sheet1 = yourworkbook.getSheetAt(0);
415                 Iterator rows = sheet1.rowIterator();
                    HSSFRow row = (HSSFRow)rows.next();
417                 int rowNum = sheet1.getLastRowNum();
418                 short colNum = sheet1.getRow(0).getLastCellNum();
419                 int j = 1;
420                 Class.forName("com.mysql.jdbc.Driver");
                    Connection conn = (Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root", "mysqlip");
422                 Statement stmt = (Statement)conn.createStatement();
423                 String Sql1 = "truncate books;";
424                 stmt.executeUpdate(Sql1);
425                 for (int i = 0; i < rowNum; ++i) {
426                     String[] array = new String[colNum];
427                     int a = 0;
428                     try {
429                         while (rows.hasNext()) {
430                             row = sheet1.getRow(j);
431                             HSSFCell column = row.getCell(a);
432                             column.setCellType(1);
433                             if (column.getCellType() == 1) {
434                                 array[a] = column.getStringCellValue();
435                             } else if (column.getCellType() == 3) {
436                                 array[a] = "";
437                             }
438                             ++a;
439                         }
440                     }
```

```
444                     String sql = "Insert into books(BOOK_GROUP,BOOK_TITLE,ACCESSION_NO,AUTHOR_NAME,EDITOR_NAME,PUBLISHER,EDITION,PAGES"
445                             + ",ISBN_NO,VENDOR_NAME,LANGUAGE,VOLUMES,PURCHASE_DATE,BILL_NO,BILL_DATE,PRICE_PER_COPY,DDC_CODE"
446                             + ",DDC_DESCRIPTION,BOOK_LOCATION,DONATED,SPECIMEN,PUBLICATION_YEAR,CURRENCY_DESC) values "
447                             + "('" + array[0] + "','" + array[1] + "','" + array[2] + "','" + array[3] + "','" + array[4] + "','"
448                             + "" + array[5] + "','" + array[6] + "','" + array[7] + "','" + array[8] + "','" + array[9] + "','"
449                             + "" + array[10] + "','" + array[11] + "','" + array[12] + "','" + array[13] + "','" + array[14] + "','"
450                             + "" + array[15] + "','" + array[16] + "','" + array[17] + "','" + array[18] + "','" + array[19] + "','"
                                + "" + array[20] + "','" + array[21] + "','" + array[22] + "');";
452                     stmt.executeUpdate(sql);
453                     ++j;
454                 }
455                 stmt.close();
456                 conn.close();
457                 JOptionPane.showMessageDialog(null, "Database Updated");
458             }
459             catch (Exception e) {
460                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
461             }
462         }
463     }
```

RemoveBook:



Book selected:



After Book removed:

CODE:

Imports:

```
3  import com.mysql.jdbc.Connection;
4  import com.mysql.jdbc.Statement;
5  import java.awt.Cursor;
6  import java.awt.EventQueue;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9  import java.awt.event.KeyAdapter;
10 import java.awt.event.KeyEvent;
11 import java.awt.event.MouseAdapter;
12 import java.awt.event.MouseEvent;
13 import java.sql.DriverManager;
14 import java.sql.SQLException;
15 import javax.swing.DefaultComboBoxModel;
16 import javax.swing.GroupLayout;
17 import javax.swing.JButton;
18 import javax.swing.JComboBox;
19 import javax.swing.JFrame;
20 import javax.swing.JMenu;
21 import javax.swing.JMenuBar;
22 import javax.swing.JMenuItem;
23 import javax.swing.JOptionPane;
24 import javax.swing.JScrollPane;
25 import javax.swing.JTable;
26 import javax.swing.JTextField;
27 import javax.swing.LayoutStyle;
28 import javax.swing.table.DefaultTableModel;
```

Search Button:

```
301     private void SearchButtonActionPerformed(ActionEvent evt) {
302         String item = (String)SearchBox.getSelectedItem();
303         String text = SearchTextField.getText();
304         DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
305         while (model.getRowCount() > 0) {
306             model.setRowCount(0);
307         }
308         new OpenPage().Search(item, text, model);
309     }
```

Search Enter Key Pressed:

```
311     private void SearchTextFieldKeyPressed(KeyEvent evt) {
312         if (evt.getKeyCode() == 10) {
313             String item = (String)SearchBox.getSelectedItem();
314             String text = SearchTextField.getText();
315             DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
316             while (model.getRowCount() > 0) {
317                 model.setRowCount(0);
318             }
319             new OpenPage().Search(item, text, model);
320         }
321     }
```

Table Mouse Clicked: (For selecting book)

```
291     private void SearchResultMouseClicked(MouseEvent evt) {
292         DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
293         column = SearchResult.getSelectedColumn();
294         row = SearchResult.getSelectedRow();
295         acess = model.getValueAt(row, 2);
296         if (evt.getClickCount() == 2) {
297             weddout();
298         }
299     }
```

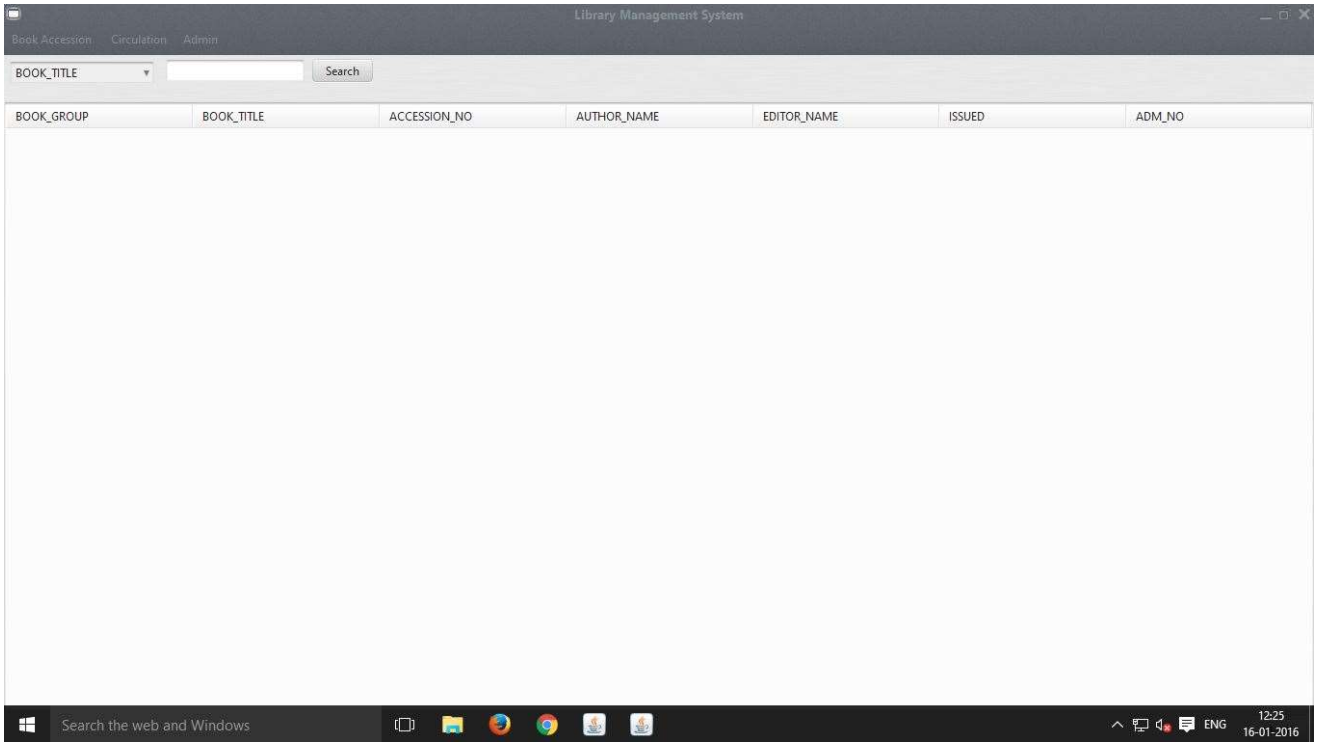
WeddOut Method:

```
254     void weddout() {
255         try {
256             Class.forName("com.mysql.jdbc.Driver");
                Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
258             Statement stmt = (Statement)conn.createStatement();
259             Statement stmt1 = (Statement)conn.createStatement();
260             String sql = "INSERT INTO weddedout (BOOK_GROUP,BOOK_TITLE,ACCESSION_NO,AUTHOR_NAME,EDITOR_NAME,PUBLISHER"
261                     + ",EDITION,PAGES,ISBN_NO,VENDOR_NAME,LANGUAGE,VOLUMES,PURCHASE_DATE,BILL_NO,BILL_DATE,PRICE_PER_COPY"
262                     + ",DDC_CODE,DDC_DESCRIPTION,BOOK_LOCATION,DONATED,SPECIMEN,PUBLICATION_YEAR,CURRENCY_DESC)Select BOOK_GROUP"
263                     + ",BOOK_TITLE,ACCESSION_NO,AUTHOR_NAME,EDITOR_NAME,PUBLISHER,EDITION,PAGES,ISBN_NO,VENDOR_NAME,LANGUAGE,"
264                     + "VOLUMES,PURCHASE_DATE,BILL_NO,BILL_DATE,PRICE_PER_COPY,DDC_CODE,DDC_DESCRIPTION,BOOK_LOCATION,DONATED,"
265                     + "SPECIMEN,PUBLICATION_YEAR,CURRENCY_DESC FROM books WHERE accession_no ='" + acess + "' ;";
266             String sql1 = "DELETE FROM books WHERE accession_no ='" + acess + "';";
267             stmt.executeUpdate(sql);
268             stmt1.executeUpdate(sql1);
269             conn.close();
270             stmt.close();
271             stmt1.close();
272             DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
273             model.removeRow(row);
274             JOptionPane.showMessageDialog(null, "Added to Wedded Out");
275         }
276         catch (ArrayIndexOutOfBoundsException e) {
277             JOptionPane.showMessageDialog(null, "Please Select a book");
278         }
            catch (SQLException e) {
280             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
281         }
282         catch (ClassNotFoundException e) {
283             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
284         }
285     }
```

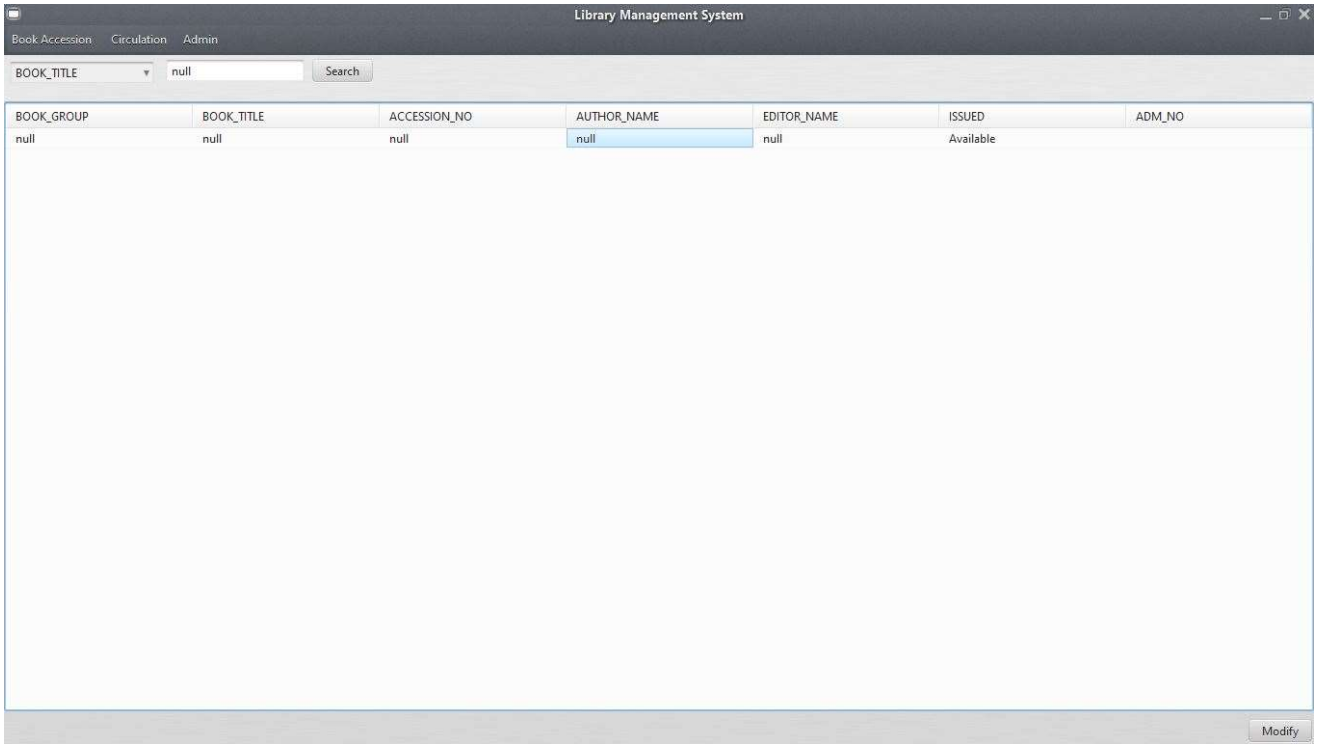
Weddout Button:

```
287     private void WeddOutButtonActionPerformed(ActionEvent evt) {
288         weddout();
289     }
```

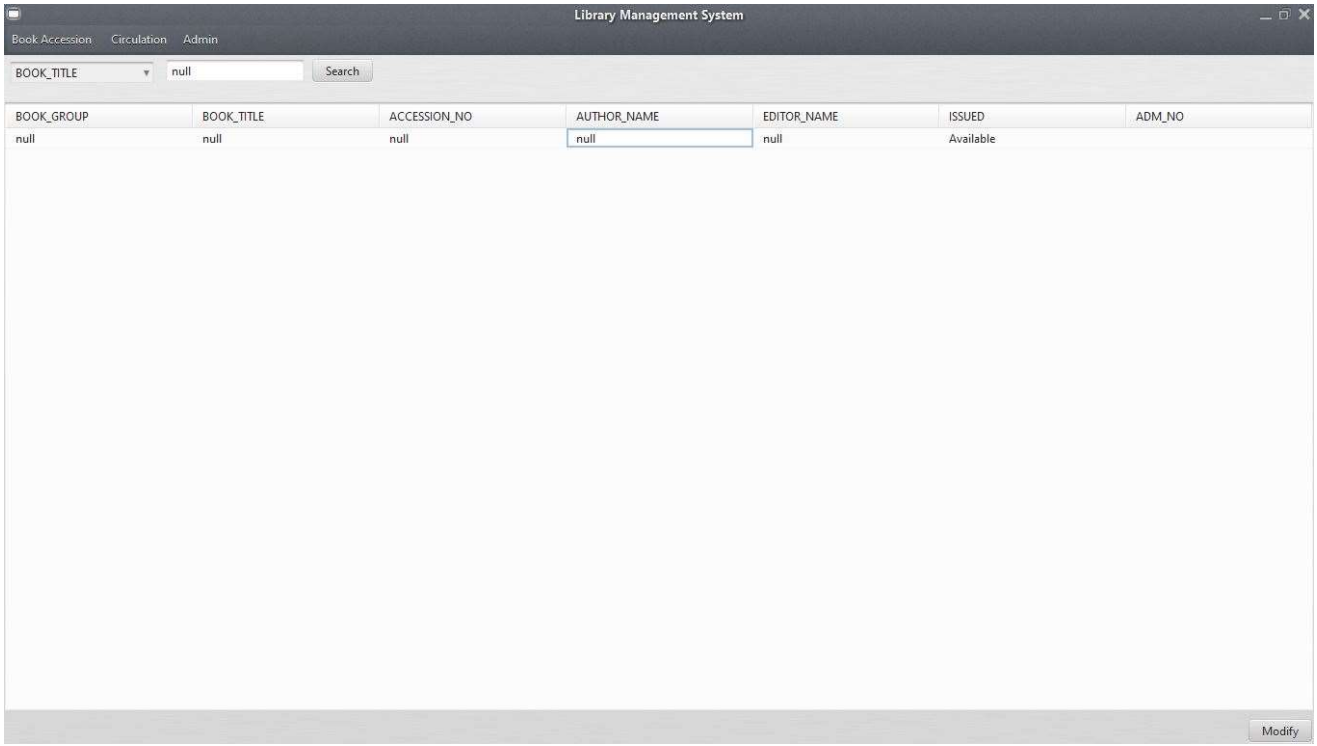
ModifyBook:



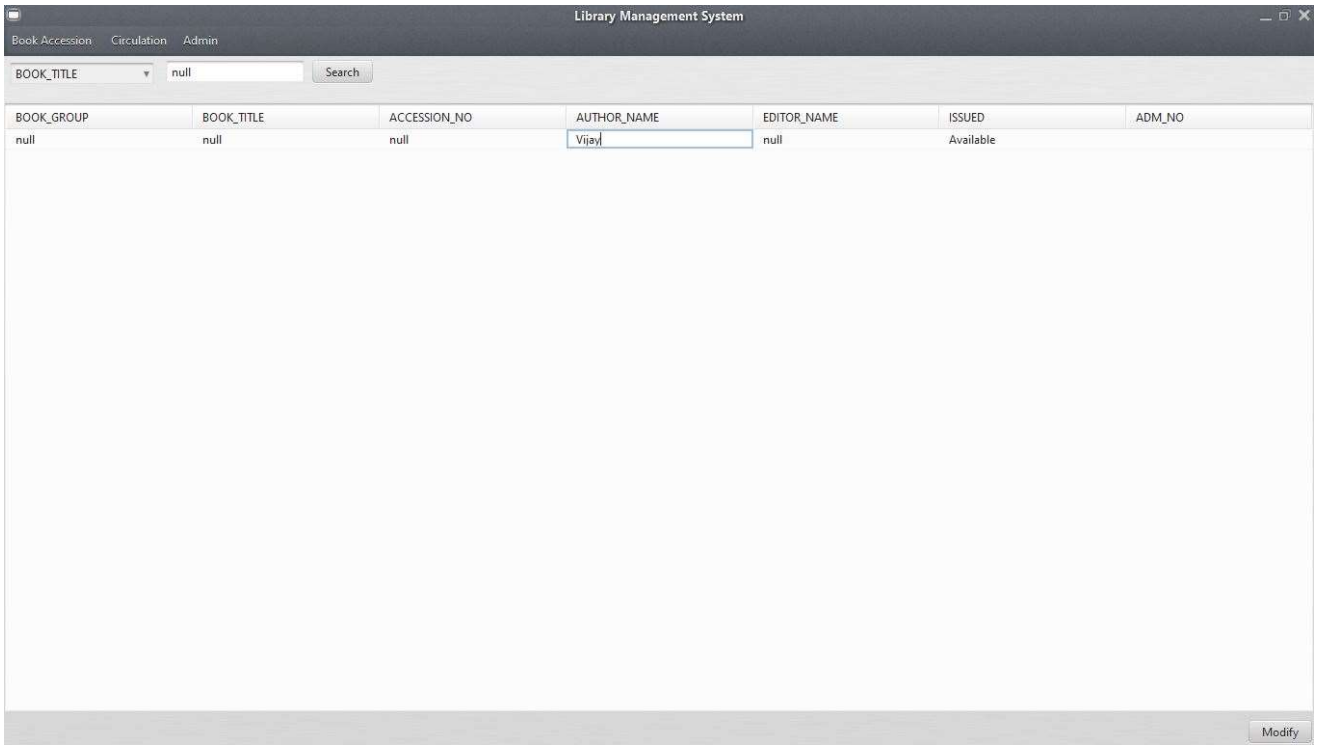
Select Field:



Edit Field:

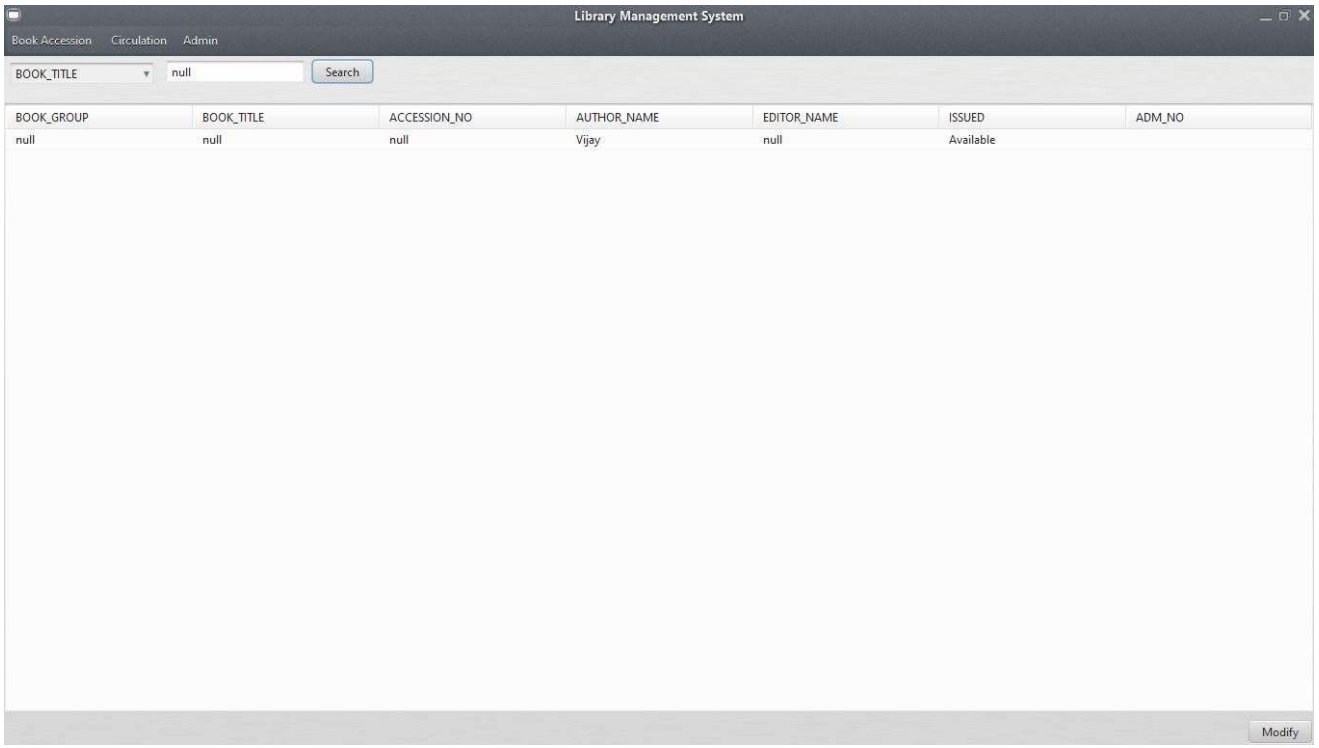

Library Management System
Book Accession
Circulation
Admin
BOOK_TITLE
null
Search
BOOK_GROUP
BOOK_TITLE
ACCESSION_NO
AUTHOR_NAME
EDITOR_NAME
ISSUED
ADM_NO
null
null
null
null
null
Available
Modify

EditValue:


Library Management System
Book Accession
Circulation
Admin
BOOK_TITLE
null
Search
BOOK_GROUP
BOOK_TITLE
ACCESSION_NO
AUTHOR_NAME
EDITOR_NAME
ISSUED
ADM_NO
null
null
null
Vijay
null
Available
Modify

Modified Sucessfully:

Modified Value:



CODE:

Imports:

```
3  import com.mysql.jdbc.Connection;
4  import com.mysql.jdbc.Statement;
5  import java.awt.Cursor;
6  import java.awt.EventQueue;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9  import java.awt.event.KeyAdapter;
10 import java.awt.event.KeyEvent;
11 import java.awt.event.MouseAdapter;
12 import java.awt.event.MouseEvent;
13 import java.sql.DriverManager;
14 import javax.swing.DefaultComboBoxModel;
15 import javax.swing.GroupLayout;
16 import javax.swing.JButton;
17 import javax.swing.JComboBox;
18 import javax.swing.JFrame;
19 import javax.swing.JMenu;
20 import javax.swing.JMenuBar;
21 import javax.swing.JMenuItem;
22 import javax.swing.JOptionPane;
23 import javax.swing.JScrollPane;
24 import javax.swing.JTable;
25 import javax.swing.JTextField;
26 import javax.swing.LayoutStyle;
27 import javax.swing.table.DefaultTableModel;
```

Search Button:

```
303     private void SearchButtonActionPerformed(ActionEvent evt) {
304         String item = (String)SearchBox.getSelectedItem();
305         String text = SearchTextField.getText();
306         DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
307         while (model.getRowCount() > 0) {
308             model.setRowCount(0);
309         }
310         new OpenPage().Search(item, text, model);
311     }
```

Search Enter Key Pressed:

```
321     private void SearchTextFieldKeyPressed(KeyEvent evt) {
322         if (evt.getKeyCode() == 10) {
323             String item = (String)SearchBox.getSelectedItem();
324             String text = SearchTextField.getText();
325             DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
326             while (model.getRowCount() > 0) {
327                 model.setRowCount(0);
328             }
329             new OpenPage().Search(item, text, model);
330         }
331     }
```

Table Mouse Clicked Action: (To select the value to modify)

```
313     private void SearchResultMouseClicked(MouseEvent evt) {
314         row = SearchResult.getSelectedRow();
315         column = SearchResult.getSelectedColumn();
316         DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
317         modified = model.getValueAt(row,column);
318         acess = model.getValueAt(row, 2);
319     }
```

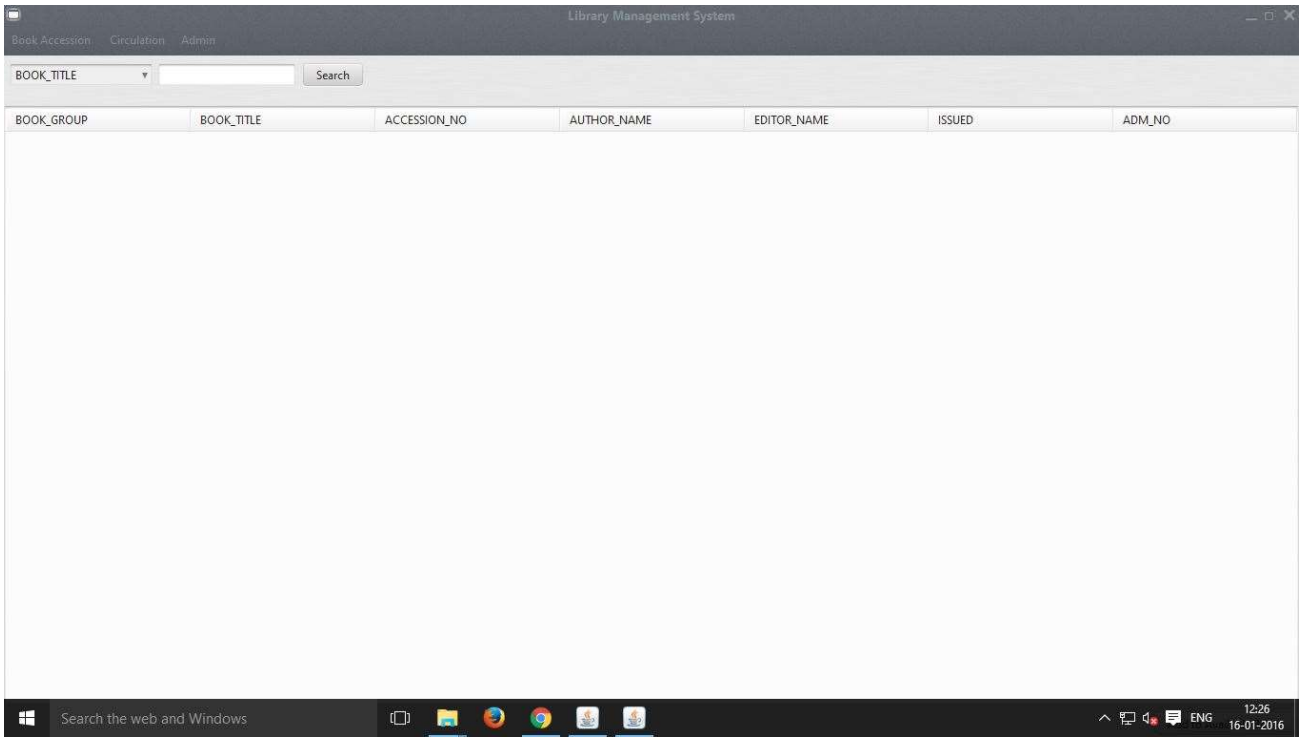
Modify Method:

```
259     void Modify() {
260         try {
261             switch (column) {
262                 case 0: {
263                     colum = "Book_group";
264                     break;
265                 }
266                 case 1: {
267                     colum = "book_title";
268                     break;
269                 }
270                 case 2: {
271                     colum = "accession_no";
272                     break;
273                 }
274                 case 3: {
275                     colum = "author_name";
276                     break;
277                 }
278                 case 4: {
279                     colum = "Editor_Name";
280                     break;
281                 }
282                 case 6: {
283                     colum = "Adm_No";
284                 }
285             }
286             Class.forName("com.mysql.jdbc.Driver");
287             Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
288             Statement stmt = (Statement)conn.createStatement();
289             String sql = "Update books set " + colum + " = '" + modified + "' where accession_no = '" + acess + "';";
290             stmt.executeUpdate(sql);
291             conn.close();
292             stmt.close();
293             JOptionPane.showMessageDialog(null, "Modified Sucessfully");
294         }
295         catch (Exception e) {
296             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
297         }
298     }
```

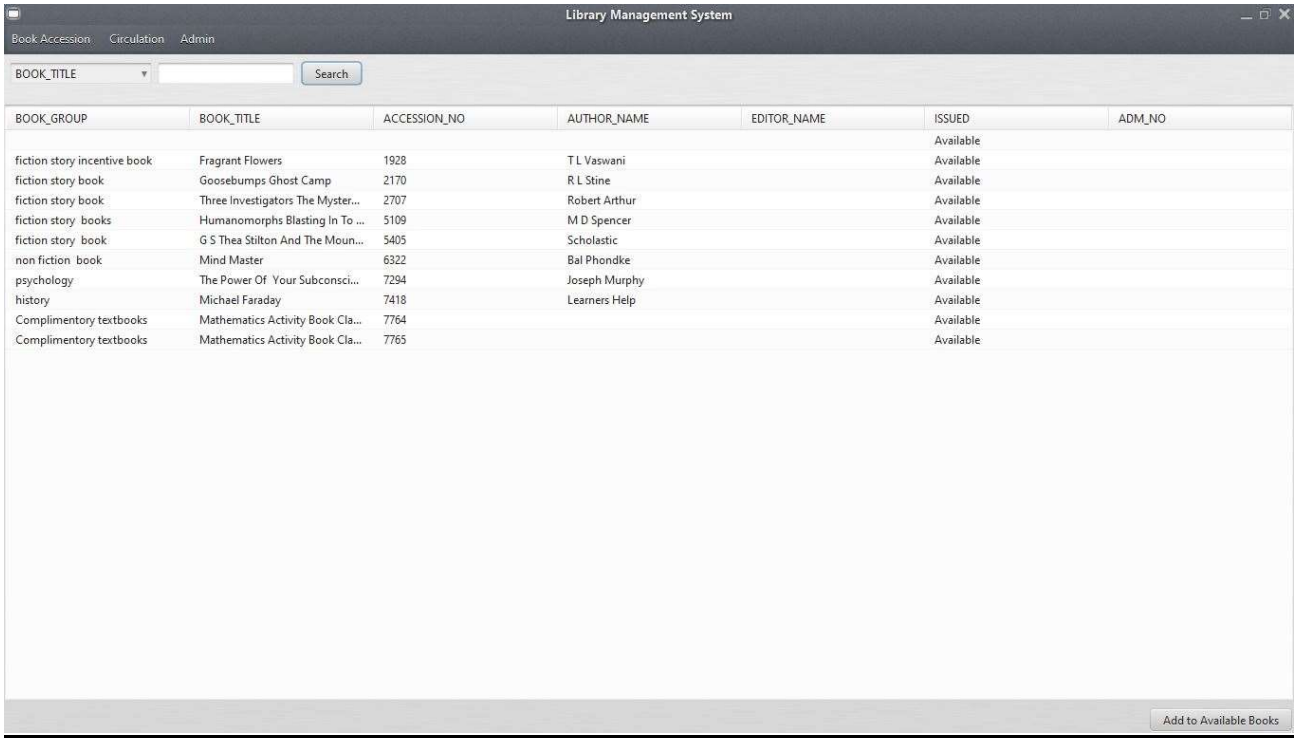
Method Button:

```
299     private void ModifyButtonActionPerformed(ActionEvent evt) {
300         Modify();
301     }
```

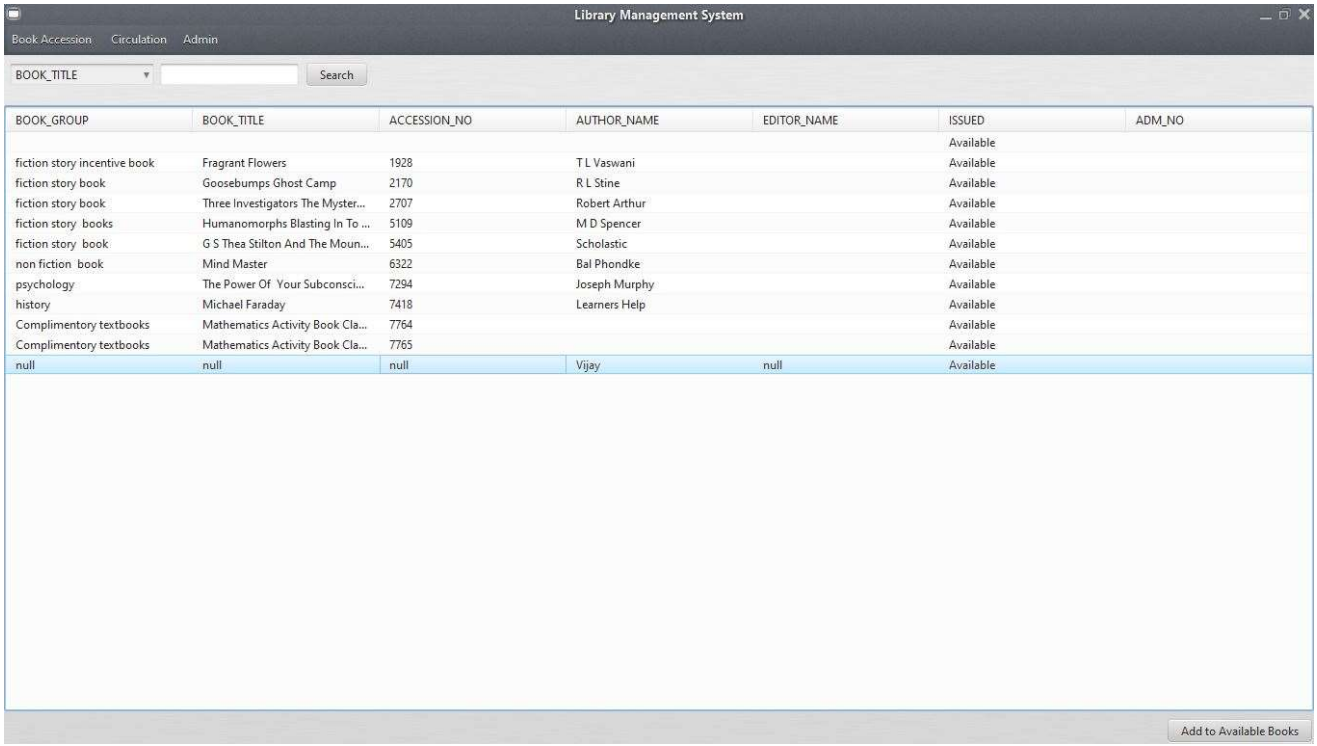
Weddedout:



Search Book:



Select Book:



Added To available books:

Imports:

```
3  import com.mysql.jdbc.Connection;
4  import com.mysql.jdbc.Statement;
5  import java.awt.Cursor;
6  import java.awt.EventQueue;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9  import java.awt.event.KeyAdapter;
10 import java.awt.event.KeyEvent;
11 import java.awt.event.MouseAdapter;
12 import java.awt.event.MouseEvent;
13 import java.sql.DriverManager;
14 import java.sql.ResultSet;
15 import javax.swing.DefaultComboBoxModel;
16 import javax.swing.GroupLayout;
17 import javax.swing.JButton;
18 import javax.swing.JComboBox;
19 import javax.swing.JFrame;
20 import javax.swing.JMenu;
21 import javax.swing.JMenuBar;
22 import javax.swing.JMenuItem;
23 import javax.swing.JOptionPane;
24 import javax.swing.JScrollPane;
25 import javax.swing.JTable;
26 import javax.swing.JTextField;
27 import javax.swing.LayoutStyle;
28 import javax.swing.table.DefaultTableModel;
```

Search Button:

```
305    private void SearchButtonActionPerformed(ActionEvent evt) {
306        String item = (String)SearchBox.getSelectedItem();
307        String text = SearchTextField.getText();
308        DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
309        try {
310            while (model.getRowCount() > 0) {
311                model.setRowCount(0);
312            }
313            Class.forName("com.mysql.jdbc.Driver");
314            Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
315            Statement stmt = (Statement)conn.createStatement();
316            String sql = "select * from weddedout where " + item + " like'%" + text + "%';";
317            ResultSet rs = stmt.executeQuery(sql);
318            while (rs.next()) {
319                String BOOK_GROUP = rs.getString("BOOK_GROUP");
320                String BOOK_TITLE = rs.getString("BOOK_TITLE");
321                String ACCESSION_NO = rs.getString("ACCESSION_NO");
322                String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
323                String EDITOR_NAME = rs.getString("EDITOR_NAME");
324                Boolean ISSUED = rs.getBoolean("ISSUED");
325                String ADM_NO = rs.getString("ADM_NO");
326                String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
327                String ISSUE = ISSUED == true ? "Issued" : "Available";
328                model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, ADM_NO, BOOK_LOCATION});
329            }
330            rs.close();
331            conn.close();
332            stmt.close();
333        }
334        catch (Exception e) {
335            JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
336        }
337    }
```

Search Enter key:

```
351     private void SearchTextFieldKeyPressed(KeyEvent evt) {
352         if (evt.getKeyCode() == 10) {
353             String item = (String)SearchBox.getSelectedItem();
354             String text = SearchTextField.getText();
355             DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
356             try {
357                 while (model.getRowCount() > 0) {
358                     model.setRowCount(0);
359                 }
360                 Class.forName("com.mysql.jdbc.Driver");
                    Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
362                 Statement stmt = (Statement)conn.createStatement();
363                 String sql = "select * from weddedout where " + item + " like'%" + text + "%';";
364                 ResultSet rs = stmt.executeQuery(sql);
365                 while (rs.next()) {
366                     String BOOK_GROUP = rs.getString("BOOK_GROUP");
367                     String BOOK_TITLE = rs.getString("BOOK_TITLE");
368                     String ACCESSION_NO = rs.getString("ACCESSION_NO");
369                     String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
370                     String EDITOR_NAME = rs.getString("EDITOR_NAME");
371                     Boolean ISSUED = rs.getBoolean("ISSUED");
372                     String ADM_NO = rs.getString("ADM_NO");
373                     String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
374                     String ISSUE = ISSUED == true ? "Issued" : "Available";
375                     model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, ADM_NO, BOOK_LOCATION});
376                 }
377                 rs.close();
378                 conn.close();
379                 stmt.close();
380             }
381             catch (Exception e) {
382                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
383             }
384         }
385     }
```

Add Button:

```
333     private void AddButtonActionPerformed(ActionEvent evt) {
334         DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
335         String item = (String)SearchBox.getSelectedItem();
336         String text = SearchTextField.getText();
337         weddout(item, text, model);
338     }
```

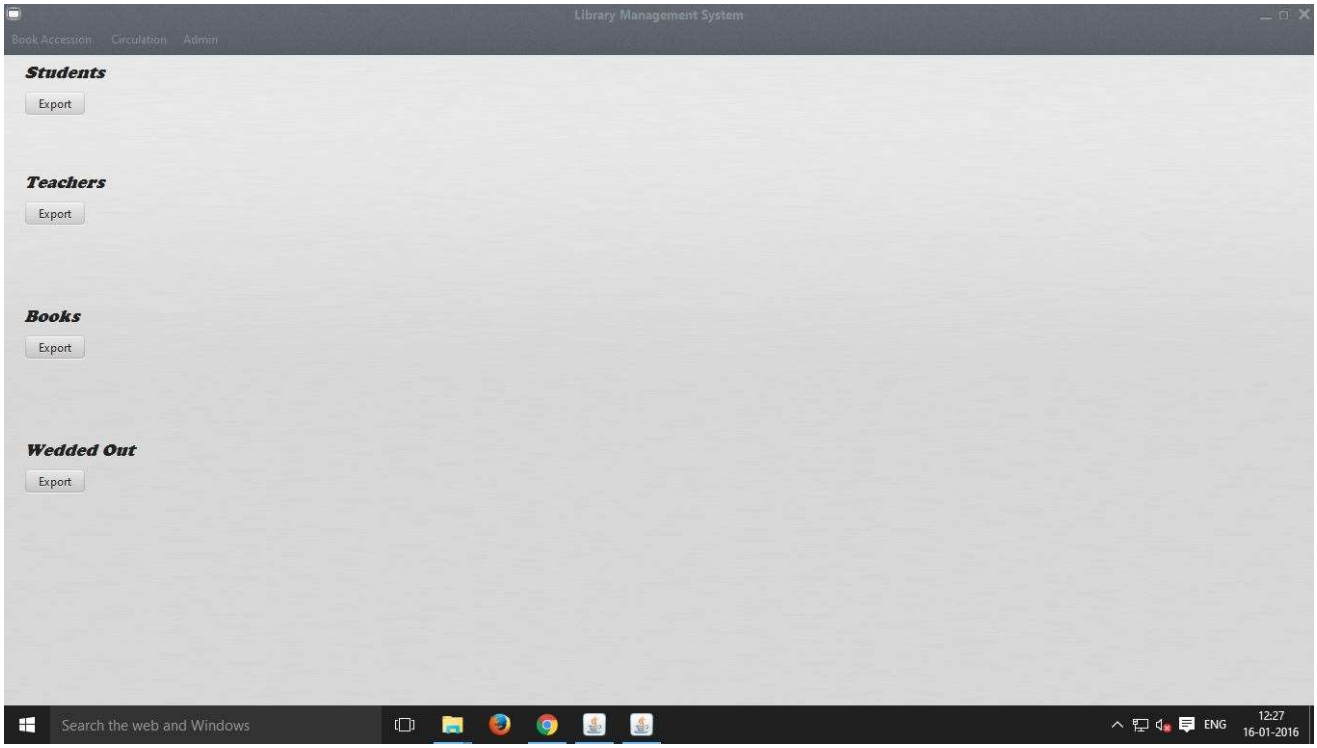
Table Mouse Clicked

```
340     private void SearchResultMouseClicked(MouseEvent evt) {
341         DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
342         row = SearchResult.getSelectedRow();
343         acess = model.getValueAt(row, 2);
344         if (evt.getClickCount() == 2) {
345             String item = (String)SearchBox.getSelectedItem();
346             String text = SearchTextField.getText();
347             weddout(item, text, model);
348         }
349     }
```

Weddedout Method:

```
260     int weddout(String item, String text, DefaultTableModel model) {
261         while (model.getRowCount() > 0) {
262             model.setRowCount(0);
263         }
264         try {
265             Class.forName("com.mysql.jdbc.Driver");
                Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
267             Statement stmt = (Statement)conn.createStatement();
268             String sql = "INSERT INTO books (BOOK_GROUP,BOOK_TITLE,ACCESSION_NO,AUTHOR_NAME,EDITOR_NAME,"
269                     + "PUBLISHER,EDITION,PAGES,ISBN_NO,VENDOR_NAME,LANGUAGE,VOLUMES,PURCHASE_DATE,BILL_NO,"
270                     + "BILL_DATE,PRICE_PER_COPY,DDC_CODE,DDC_DESCRIPTION,BOOK_LOCATION,DONATED,SPECIMEN,PUBLICATION_YEAR,"
271                     + "CURRENCY_DESC)Select BOOK_GROUP,BOOK_TITLE,ACCESSION_NO,AUTHOR_NAME,EDITOR_NAME,PUBLISHER,EDITION,PAGES,"
272                     + "ISBN_NO,VENDOR_NAME,LANGUAGE,VOLUMES,PURCHASE_DATE,BILL_NO,BILL_DATE,PRICE_PER_COPY,DDC_CODE,DDC_DESCRIPTION,"
273                     + "BOOK_LOCATION,DONATED,SPECIMEN,PUBLICATION_YEAR,CURRENCY_DESC FROM weddedout WHERE accession_no "
274                     + "='" + acess + "' ;";
275             String sql1 = "DELETE FROM weddedout WHERE ACCESSION_NO ='" + acess + "';";
276             stmt.executeUpdate(sql);
277             stmt.executeUpdate(sql1);
278             while (model.getRowCount() > 0) {
279                 model.setRowCount(0);
280             }
281             String sql2 = "select * from weddedout where " + item + " like'%" + text + "%';";
282             ResultSet rs = stmt.executeQuery(sql2);
283             while (rs.next()) {
284                 String BOOK_GROUP = rs.getString("BOOK_GROUP");
285                 String BOOK_TITLE = rs.getString("BOOK_TITLE");
286                 String ACCESSION_NO = rs.getString("ACCESSION_NO");
287                 String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
288                 String EDITOR_NAME = rs.getString("EDITOR_NAME");
289                 Boolean ISSUED = rs.getBoolean("ISSUED");
290                 String ADM_NO = rs.getString("ADM_NO");
291                 String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
292                 String ISSUE = ISSUED == true ? "Issued" : "Available";
293                 model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, ADM_NO, BOOK_LOCATION});
294             }
295             conn.close();
296             stmt.close();
297             JOptionPane.showMessageDialog(null, "Added to Available Books");
298         }
            catch (Exception e) {
300             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
301         }
302         return 0;
303     }
```

Export:



After Export Successful:

CODE:

Imports:

```
3  import com.mysql.jdbc.Connection;
4  import com.mysql.jdbc.Statement;
5  import java.awt.Cursor;
6  import java.awt.Dimension;
7  import java.awt.EventQueue;
8  import java.awt.Font;
9  import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11 import java.io.FileOutputStream;
12 import java.io.OutputStream;
13 import java.sql.DriverManager;
14 import java.sql.ResultSet;
15 import javax.swing.GroupLayout;
16 import javax.swing.JButton;
17 import javax.swing.JFrame;
18 import javax.swing.JLabel;
19 import javax.swing.JMenu;
20 import javax.swing.JMenuBar;
21 import javax.swing.JMenuItem;
22 import javax.swing.JOptionPane;
23 import javax.swing.LayoutStyle;
24 import org.apache.poi.hssf.usermodel.HSSFCell;
25 import org.apache.poi.hssf.usermodel.HSSFRow;
26 import org.apache.poi.hssf.usermodel.HSSFSheet;
27 import org.apache.poi.hssf.usermodel.HSSFWorkbook;
```

Export for Students Button:

```
251     private void StudentExportButtonActionPerformed(ActionEvent evt) {
252         try {
253             String excelFileName = "D:/Students.xls";
254             String sheetName = "Sheet1";
255             HSSFWorkbook wb = new HSSFWorkbook();
256             HSSFSheet sheet = wb.createSheet(sheetName);
257             HSSFRow row1 = sheet.createRow(0);
258             HSSFCell cell = row1.createCell(0);
259             cell.setCellType(1);
260             cell.setCellValue("ADM_NO");
261             HSSFCell cell1 = row1.createCell(1);
262             cell1.setCellType(1);
263             cell1.setCellValue("NAME");
264             HSSFCell cell2 = row1.createCell(2);
265             cell2.setCellType(1);
266             cell2.setCellValue("STD");
267             Class.forName("com.mysql.jdbc.Driver");
                Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
269             Statement stmt = (Statement)conn.createStatement();
270             String sql = "select * from students";
271             String sql1 = "select count(*) from students;";
272             ResultSet rs1 = stmt.executeQuery(sql1);
273             rs1.next();
274             int TOTAL = rs1.getInt("count(*)");
275             ResultSet rs = stmt.executeQuery(sql);
276             rs.next();
277             for (int j = 1; j <= TOTAL; ++j) {
278                 HSSFRow row = sheet.createRow(j);
279                 HSSFCell cell4 = row.createCell(0);
280                 cell4.setCellType(1);
281                 cell4.setCellValue(rs.getString("ADM_NO"));
282                 HSSFCell cell5 = row.createCell(1);
283                 cell5.setCellType(1);
284                 cell5.setCellValue(rs.getString("NAME"));
285                 HSSFCell cell6 = row.createCell(2);
286                 cell6.setCellType(1);
287                 cell6.setCellValue(rs.getString("STD"));
288                 rs.next();
289             }
290             stmt.close();
291             conn.close();
292             FileOutputStream fileOut = new FileOutputStream(excelFileName);
293             wb.write((OutputStream)fileOut);
294             fileOut.flush();
295             fileOut.close();
296             JOptionPane.showMessageDialog(null, "Exported Sucessfully");
297         }
            catch (Exception e) {
                e.printStackTrace();
300         }
301     }
```

Export for Teachers Button:

```
303     private void TeacherExportButtonActionPerformed(ActionEvent evt) {
304         try {
305             String excelFileName = "D:/Teachers.xls";
306             String sheetName = "Sheet1";
307             HSSFWorkbook wb = new HSSFWorkbook();
308             HSSFSheet sheet = wb.createSheet(sheetName);
309             HSSFRow row1 = sheet.createRow(0);
310             HSSFCell cell = row1.createCell(0);
311             cell.setCellType(1);
312             cell.setCellValue("SL_NO");
313             HSSFCell cell1 = row1.createCell(1);
314             cell1.setCellType(1);
315             cell1.setCellValue("NAME");
316             HSSFCell cell2 = row1.createCell(2);
317             cell2.setCellType(1);
318             cell2.setCellValue("INITIAL");
319             HSSFCell cell3 = row1.createCell(3);
320             cell3.setCellType(1);
321             cell3.setCellValue("SUBJECT");
322             Class.forName("com.mysql.jdbc.Driver");
                Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
324             Statement stmt = (Statement)conn.createStatement();
325             String sql = "select * from teachers";
326             String sql1 = "select count(*) from teachers;";
327             ResultSet rs1 = stmt.executeQuery(sql1);
328             rs1.next();
329             int TOTAL = rs1.getInt("count(*)");
330             ResultSet rs = stmt.executeQuery(sql);
331             rs.next();
332             for (int j = 1; j < TOTAL; ++j) {
333                 HSSFRow row = sheet.createRow(j);
334                 HSSFCell cell4 = row.createCell(0);
335                 cell4.setCellType(1);
336                 cell4.setCellValue(rs.getString("SL_NO"));
337                 HSSFCell cell5 = row.createCell(1);
338                 cell5.setCellType(1);
339                 cell5.setCellValue(rs.getString("NAME"));
340                 HSSFCell cell6 = row.createCell(2);
341                 cell6.setCellType(1);
342                 cell6.setCellValue(rs.getString("INITIAL"));
343                 HSSFCell cell7 = row.createCell(3);
344                 cell7.setCellType(1);
345                 cell7.setCellValue(rs.getString("SUBJECT"));
346                 rs.next();
347             }
348             stmt.close();
349             conn.close();
350             FileOutputStream fileOut = new FileOutputStream(excelFileName);
351             wb.write((OutputStream)fileOut);
352             fileOut.flush();
353             fileOut.close();
354             JOptionPane.showMessageDialog(null, "Exported Sucessfully");
355         }
            catch (Exception e) {
357             e.getLocalizedMessage();
358         }
359     }
```

Export for Books Button:

```
361     private void BookExportButtonActionPerformed(ActionEvent evt) {
362         try {
363             String excelFileName = "D:/Books.xls";
364             String sheetName = "Sheet1";
365             HSSFWorkbook wb = new HSSFWorkbook();
366             HSSFSheet sheet = wb.createSheet(sheetName);
367             HSSFRow row1 = sheet.createRow(0);
368             HSSFCell cell = row1.createCell(0);
369             cell.setCellType(1);
370             cell.setCellValue("BOOK_GROUP");
371             HSSFCell cell1 = row1.createCell(1);
372             cell1.setCellType(1);
373             cell1.setCellValue("BOOK_TITLE");
374             HSSFCell cell2 = row1.createCell(2);
375             cell2.setCellType(1);
376             cell2.setCellValue("ACCESSION_NO");
377             HSSFCell cell3 = row1.createCell(3);
378             cell3.setCellType(1);
379             cell3.setCellValue("AUTHOR_NAME");
380             HSSFCell cell8 = row1.createCell(4);
381             cell8.setCellType(1);
382             cell8.setCellValue("EDITOR_NAME");
383             HSSFCell cell9 = row1.createCell(5);
384             cell9.setCellType(1);
385             cell9.setCellValue("PUBLISHER");
386             HSSFCell cell10 = row1.createCell(6);
387             cell10.setCellType(1);
388             cell10.setCellValue("EDITION");
389             HSSFCell cell11 = row1.createCell(7);
390             cell11.setCellType(1);
391             cell11.setCellValue("PAGES");
392             HSSFCell cell12 = row1.createCell(8);
393             cell12.setCellType(1);
394             cell12.setCellValue("ISBN_NO");
395             HSSFCell cell13 = row1.createCell(9);
396             cell13.setCellType(1);
397             cell13.setCellValue("VENDOR_NAME");
398             HSSFCell cell14 = row1.createCell(10);
399             cell14.setCellType(1);
400             cell14.setCellValue("LANGUAGE");
401             HSSFCell cell15 = row1.createCell(11);
402             cell15.setCellType(1);
403             cell15.setCellValue("VOLUMES");
404             HSSFCell cell16 = row1.createCell(12);
405             cell16.setCellType(1);
406             cell16.setCellValue("PURCHASE_DATE");
```

```
            HSSFCell cell17 = row1.createCell(13);
            cell17.setCellType(1);
            cell17.setCellValue("BILL_NO");
            HSSFCell cell18 = row1.createCell(14);
            cell18.setCellType(1);
            cell18.setCellValue("BILL_DATE");
            HSSFCell cell19 = row1.createCell(15);
            cell19.setCellType(1);
            cell19.setCellValue("PRICE_PER_COPY");
            HSSFCell cell20 = row1.createCell(16);
            cell20.setCellType(1);
            cell20.setCellValue("DDC_CODE");
            HSSFCell cell21 = row1.createCell(17);
            cell21.setCellType(1);
            cell21.setCellValue("DDC_DESCRIPTION");
            HSSFCell cell22 = row1.createCell(18);
            cell22.setCellType(1);
            cell22.setCellValue("BOOK_LOCATION");
            HSSFCell cell23 = row1.createCell(19);
            cell23.setCellType(1);
            cell23.setCellValue("DONATED");
            HSSFCell cell24 = row1.createCell(20);
            cell24.setCellType(1);
            cell24.setCellValue("SPECIMEN");
            HSSFCell cell25 = row1.createCell(21);
            cell25.setCellType(1);
            cell25.setCellValue("PUBLICATION_YEAR");
            HSSFCell cell26 = row1.createCell(22);
            cell26.setCellType(1);
            cell26.setCellValue("CURRENCY_DESC");
            HSSFCell cell27 = row1.createCell(23);
            cell27.setCellType(1);
            cell27.setCellValue("ISSUED");
            HSSFCell cell28 = row1.createCell(24);
            cell28.setCellType(1);
            cell28.setCellValue("ADM_NO");
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
            Statement stmt = (Statement)conn.createStatement();
            String sql = "select * from books";
            String sql1 = "select count(*) from books;";
            ResultSet rs1 = stmt.executeQuery(sql1);
            rs1.next();
            int TOTAL = rs1.getInt("count(*)");
            ResultSet rs = stmt.executeQuery(sql);
            rs.next();
            for (int j = 1; j <= TOTAL; ++j) {
                HSSFRow row = sheet.createRow(j);
                HSSFCell cell4 = row.createCell(0);
                cell4.setCellType(1);
                cell4.setCellValue(rs.getString("BOOK_GROUP"));
                HSSFCell cell5 = row.createCell(1);
                cell5.setCellType(1);
                cell5.setCellValue(rs.getString("BOOK_TITLE"));
                HSSFCell cell6 = row.createCell(2);
                cell6.setCellType(1);
                cell6.setCellValue(rs.getString("ACCESSION_NO"));
                HSSFCell cell7 = row.createCell(3);
                cell7.setCellType(1);
                cell7.setCellValue(rs.getString("AUTHOR_NAME"));
                HSSFCell cell81 = row.createCell(4);
                cell81.setCellType(1);
                cell81.setCellValue(rs.getString("EDITOR_NAME"));
                HSSFCell cell91 = row.createCell(5);
                cell91.setCellType(1);
                cell91.setCellValue(rs.getString("PUBLISHER"));
                HSSFCell cell101 = row.createCell(6);
                cell101.setCellType(1);
                cell101.setCellValue(rs.getString("EDITION"));
                HSSFCell cell111 = row.createCell(7);
                cell111.setCellType(1);
                cell111.setCellValue(rs.getString("PAGES"));
                HSSFCell cell121 = row.createCell(8);
                cell121.setCellType(1);
                cell121.setCellValue(rs.getString("ISBN_NO"));
                HSSFCell cell131 = row.createCell(9);
                cell131.setCellType(1);
                cell131.setCellValue(rs.getString("VENDOR_NAME"));
                HSSFCell cell141 = row.createCell(10);
                cell141.setCellType(1);
                cell141.setCellValue(rs.getString("LANGUAGE"));
                HSSFCell cell151 = row.createCell(11);
                cell151.setCellType(1);
                cell151.setCellValue(rs.getString("VOLUMES"));
                HSSFCell cell161 = row.createCell(12);
                cell161.setCellType(1);
                cell161.setCellValue(rs.getString("PURCHASE_DATE"));
                HSSFCell cell171 = row.createCell(13);
                cell171.setCellType(1);
                cell171.setCellValue(rs.getString("BILL_NO"));
                HSSFCell cell181 = row.createCell(14);
                cell181.setCellType(1);
```

```
                cell181.setCellValue(rs.getString("BILL_DATE"));
                HSSFCell cell191 = row.createCell(15);
                cell191.setCellType(1);
                cell191.setCellValue(rs.getString("PRICE_PER_COPY"));
                HSSFCell cell201 = row.createCell(16);
                cell201.setCellType(1);
                cell201.setCellValue(rs.getString("DDC_CODE"));
                HSSFCell cell211 = row.createCell(17);
                cell211.setCellType(1);
                cell211.setCellValue(rs.getString("DDC_DESCRIPTION"));
                HSSFCell cell221 = row.createCell(18);
                cell221.setCellType(1);
                cell221.setCellValue(rs.getString("BOOK_LOCATION"));
                HSSFCell cell231 = row.createCell(19);
                cell231.setCellType(1);
                cell231.setCellValue(rs.getString("DONATED"));
                HSSFCell cell251 = row.createCell(20);
                cell251.setCellType(1);
                cell251.setCellValue(rs.getString("SPECIMEN"));
                HSSFCell cell261 = row.createCell(21);
                cell261.setCellType(1);
                cell261.setCellValue(rs.getString("PUBLICATION_YEAR"));
                HSSFCell cell271 = row.createCell(22);
                cell271.setCellType(1);
                cell271.setCellValue(rs.getString("CURRENCY_DESC"));
                HSSFCell cell281 = row.createCell(23);
                cell281.setCellType(4);
                cell281.setCellValue(rs.getBoolean("ISSUED"));
                HSSFCell cell291 = row.createCell(24);
                cell291.setCellType(1);
                cell291.setCellValue(rs.getString("ADM_NO"));
                rs.next();
            }
            stmt.close();
            conn.close();
            FileOutputStream fileOut = new FileOutputStream(excelFileName);
            wb.write((OutputStream)fileOut);
            fileOut.flush();
            fileOut.close();
            JOptionPane.showMessageDialog(null, "Exported Sucessfully");
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
```

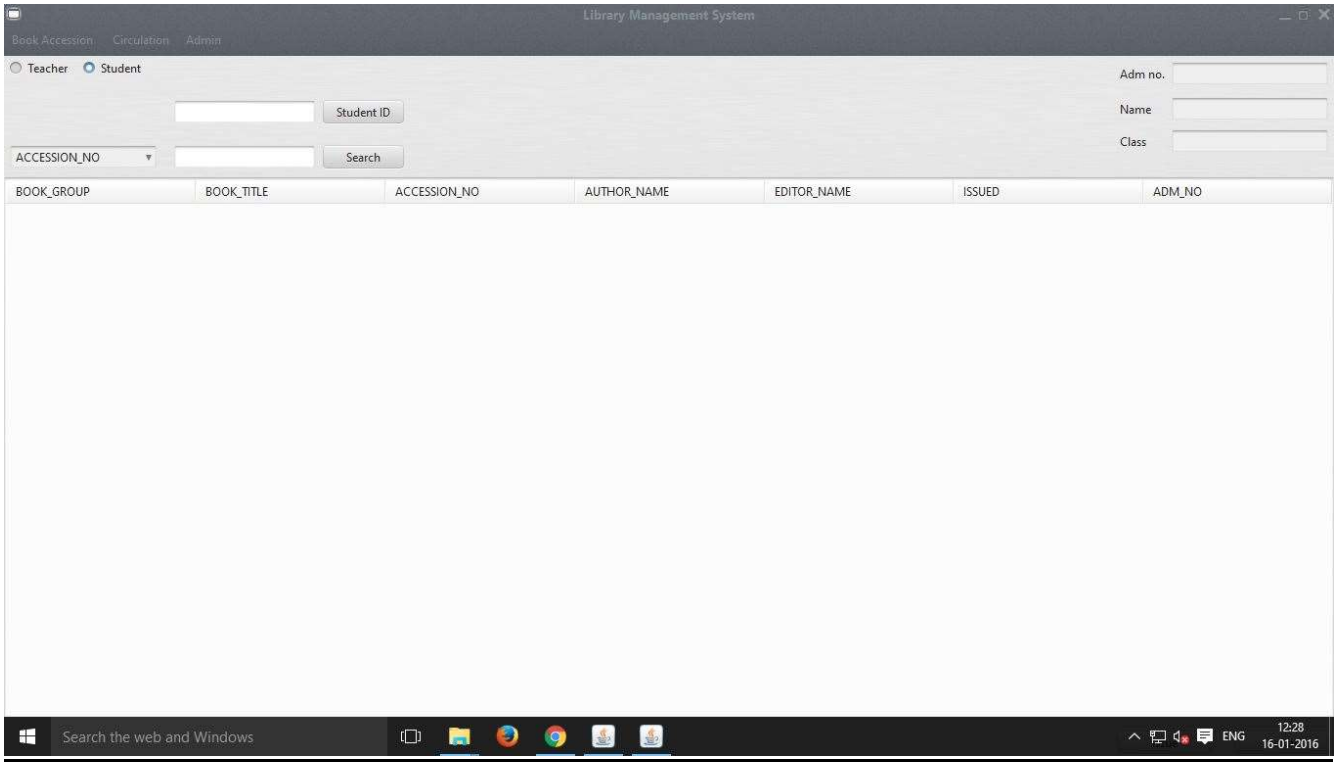
Export for Weddedout Button:

```
605     private void BookExportButton1ActionPerformed(ActionEvent evt) {
606         try {
607             String excelFileName = "D:/Wedded out.xls";
608             String sheetName = "Sheet1";
609             HSSFWorkbook wb = new HSSFWorkbook();
610             HSSFSheet sheet = wb.createSheet(sheetName);
611             HSSFRow row1 = sheet.createRow(0);
612             HSSFCell cell = row1.createCell(0);
613             cell.setCellType(1);
614             cell.setCellValue("BOOK_GROUP");
615             HSSFCell cell1 = row1.createCell(1);
616             cell1.setCellType(1);
617             cell1.setCellValue("BOOK_TITLE");
618             HSSFCell cell2 = row1.createCell(2);
619             cell2.setCellType(1);
620             cell2.setCellValue("ACCESSION_NO");
621             HSSFCell cell3 = row1.createCell(3);
622             cell3.setCellType(1);
623             cell3.setCellValue("AUTHOR_NAME");
624             HSSFCell cell8 = row1.createCell(4);
625             cell8.setCellType(1);
626             cell8.setCellValue("EDITOR_NAME");
627             HSSFCell cell9 = row1.createCell(5);
628             cell9.setCellType(1);
629             cell9.setCellValue("PUBLISHER");
630             HSSFCell cell10 = row1.createCell(6);
631             cell10.setCellType(1);
632             cell10.setCellValue("EDITION");
633             HSSFCell cell11 = row1.createCell(7);
634             cell11.setCellType(1);
635             cell11.setCellValue("PAGES");
636             HSSFCell cell12 = row1.createCell(8);
637             cell12.setCellType(1);
638             cell12.setCellValue("ISBN_NO");
639             HSSFCell cell13 = row1.createCell(9);
640             cell13.setCellType(1);
641             cell13.setCellValue("VENDOR_NAME");
642             HSSFCell cell14 = row1.createCell(10);
643             cell14.setCellType(1);
644             cell14.setCellValue("LANGUAGE");
645             HSSFCell cell15 = row1.createCell(11);
646             cell15.setCellType(1);
647             cell15.setCellValue("VOLUMES");
648             HSSFCell cell16 = row1.createCell(12);
649             cell16.setCellType(1);
650             cell16.setCellValue("PURCHASE_DATE");
```

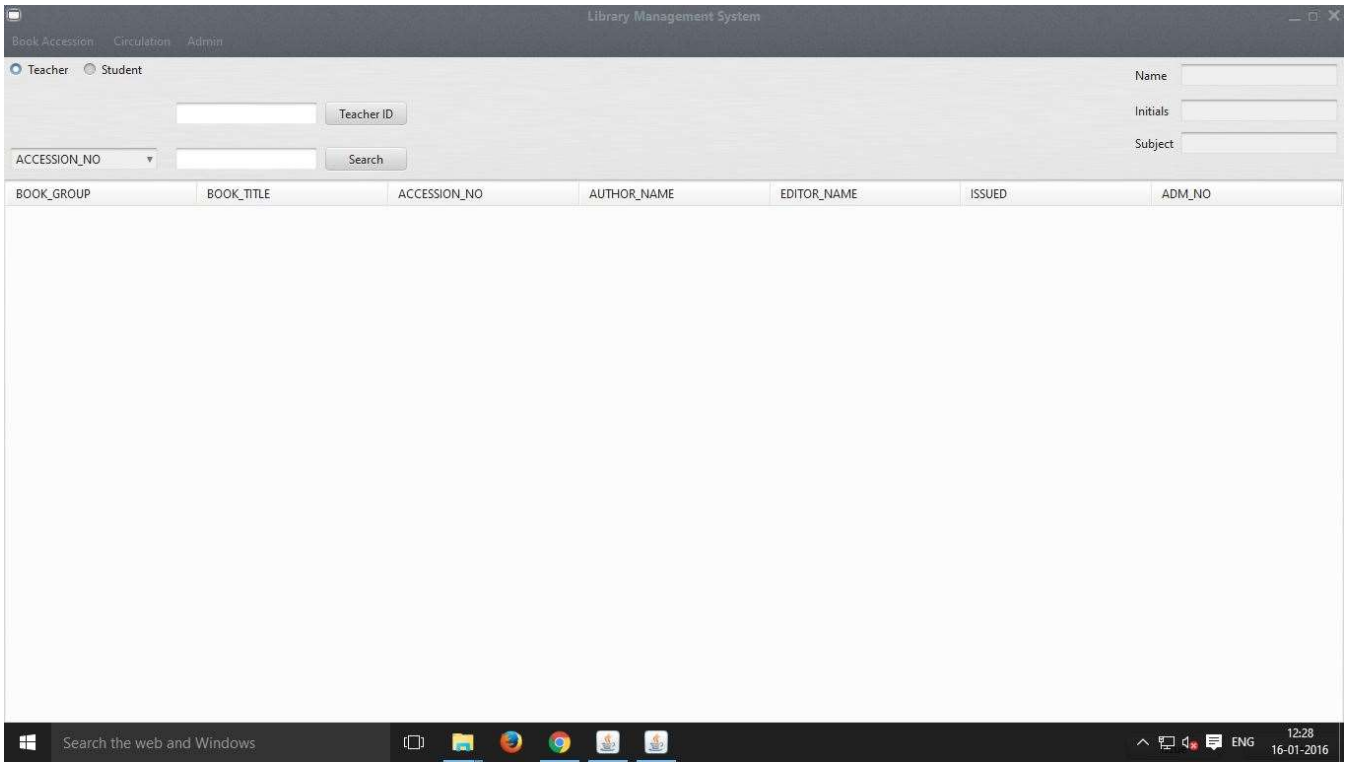
```
651             HSSFCell cell17 = row1.createCell(13);
652             cell17.setCellType(1);
653             cell17.setCellValue("BILL_NO");
654             HSSFCell cell18 = row1.createCell(14);
655             cell18.setCellType(1);
656             cell18.setCellValue("BILL_DATE");
657             HSSFCell cell19 = row1.createCell(15);
658             cell19.setCellType(1);
659             cell19.setCellValue("PRICE_PER_COPY");
660             HSSFCell cell20 = row1.createCell(16);
661             cell20.setCellType(1);
662             cell20.setCellValue("DDC_CODE");
663             HSSFCell cell21 = row1.createCell(17);
664             cell21.setCellType(1);
665             cell21.setCellValue("DDC_DESCRIPTION");
666             HSSFCell cell22 = row1.createCell(18);
667             cell22.setCellType(1);
668             cell22.setCellValue("BOOK_LOCATION");
669             HSSFCell cell23 = row1.createCell(19);
670             cell23.setCellType(1);
671             cell23.setCellValue("DONATED");
672             HSSFCell cell24 = row1.createCell(20);
673             cell24.setCellType(1);
674             cell24.setCellValue("SPECIMEN");
675             HSSFCell cell25 = row1.createCell(21);
676             cell25.setCellType(1);
677             cell25.setCellValue("PUBLICATION_YEAR");
678             HSSFCell cell26 = row1.createCell(22);
679             cell26.setCellType(1);
680             cell26.setCellValue("CURRENCY_DESC");
681             HSSFCell cell27 = row1.createCell(23);
682             cell27.setCellType(1);
683             cell27.setCellValue("ISSUED");
684             HSSFCell cell28 = row1.createCell(24);
685             cell28.setCellType(1);
686             cell28.setCellValue("ADM_NO");
687             Class.forName("com.mysql.jdbc.Driver");
                Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
689             Statement stmt = (Statement)conn.createStatement();
690             String sql = "select * from weddedout";
691             String sql1 = "select count(*) from weddedout;";
692             ResultSet rs1 = stmt.executeQuery(sql1);
693             rs1.next();
694             int TOTAL = rs1.getInt("count(*)");
695             ResultSet rs = stmt.executeQuery(sql);
696             rs.next();
```

```
697             for (int j = 1; j <= TOTAL; ++j) {
698                 HSSFRow row = sheet.createRow(j);
699                 HSSFCell cell4 = row.createCell(0);
700                 cell4.setCellType(1);
701                 cell4.setCellValue(rs.getString("BOOK_GROUP"));
702                 HSSFCell cell5 = row.createCell(1);
703                 cell5.setCellType(1);
704                 cell5.setCellValue(rs.getString("BOOK_TITLE"));
705                 HSSFCell cell6 = row.createCell(2);
706                 cell6.setCellType(1);
707                 cell6.setCellValue(rs.getString("ACCESSION_NO"));
708                 HSSFCell cell7 = row.createCell(3);
709                 cell7.setCellType(1);
710                 cell7.setCellValue(rs.getString("AUTHOR_NAME"));
711                 HSSFCell cell81 = row.createCell(4);
712                 cell81.setCellType(1);
713                 cell81.setCellValue(rs.getString("EDITOR_NAME"));
714                 HSSFCell cell91 = row.createCell(5);
715                 cell91.setCellType(1);
716                 cell91.setCellValue(rs.getString("PUBLISHER"));
717                 HSSFCell cell101 = row.createCell(6);
718                 cell101.setCellType(1);
719                 cell101.setCellValue(rs.getString("EDITION"));
720                 HSSFCell cell111 = row.createCell(7);
721                 cell111.setCellType(1);
722                 cell111.setCellValue(rs.getString("PAGES"));
723                 HSSFCell cell121 = row.createCell(8);
724                 cell121.setCellType(1);
725                 cell121.setCellValue(rs.getString("ISBN_NO"));
726                 HSSFCell cell131 = row.createCell(9);
727                 cell131.setCellType(1);
728                 cell131.setCellValue(rs.getString("VENDOR_NAME"));
729                 HSSFCell cell141 = row.createCell(10);
730                 cell141.setCellType(1);
731                 cell141.setCellValue(rs.getString("LANGUAGE"));
732                 HSSFCell cell151 = row.createCell(11);
733                 cell151.setCellType(1);
734                 cell151.setCellValue(rs.getString("VOLUMES"));
735                 HSSFCell cell161 = row.createCell(12);
736                 cell161.setCellType(1);
737                 cell161.setCellValue(rs.getString("PURCHASE_DATE"));
738                 HSSFCell cell171 = row.createCell(13);
739                 cell171.setCellType(1);
740                 cell171.setCellValue(rs.getString("BILL_NO"));
741                 HSSFCell cell181 = row.createCell(14);
742                 cell181.setCellType(1);
743                 cell181.setCellValue(rs.getString("BILL_DATE"));
744                 HSSFCell cell191 = row.createCell(15);
745                 cell191.setCellType(1);
746                 cell191.setCellValue(rs.getString("PRICE_PER_COPY"));
747                 HSSFCell cell201 = row.createCell(16);
748                 cell201.setCellType(1);
749                 cell201.setCellValue(rs.getString("DDC_CODE"));
750                 HSSFCell cell211 = row.createCell(17);
751                 cell211.setCellType(1);
752                 cell211.setCellValue(rs.getString("DDC_DESCRIPTION"));
753                 HSSFCell cell221 = row.createCell(18);
754                 cell221.setCellType(1);
755                 cell221.setCellValue(rs.getString("BOOK_LOCATION"));
756                 HSSFCell cell231 = row.createCell(19);
757                 cell231.setCellType(1);
758                 cell231.setCellValue(rs.getString("DONATED"));
759                 HSSFCell cell251 = row.createCell(20);
760                 cell251.setCellType(1);
761                 cell251.setCellValue(rs.getString("SPECIMEN"));
762                 HSSFCell cell261 = row.createCell(21);
763                 cell261.setCellType(1);
764                 cell261.setCellValue(rs.getString("PUBLICATION_YEAR"));
765                 HSSFCell cell271 = row.createCell(22);
766                 cell271.setCellType(1);
767                 cell271.setCellValue(rs.getString("CURRENCY_DESC"));
768                 HSSFCell cell281 = row.createCell(23);
769                 cell281.setCellType(4);
770                 cell281.setCellValue(rs.getBoolean("ISSUED"));
771                 HSSFCell cell291 = row.createCell(24);
772                 cell291.setCellType(1);
773                 cell291.setCellValue(rs.getString("ADM_NO"));
774                 rs.next();
775             }
776             stmt.close();
777             conn.close();
778             FileOutputStream fileOut = new FileOutputStream(excelFileName);
779             wb.write((OutputStream)fileOut);
780             fileOut.flush();
781             fileOut.close();
782             JOptionPane.showMessageDialog(null, "Exported Sucessfully");
783         }
            catch (Exception e) {
                e.printStackTrace();
786         }
787     }
```

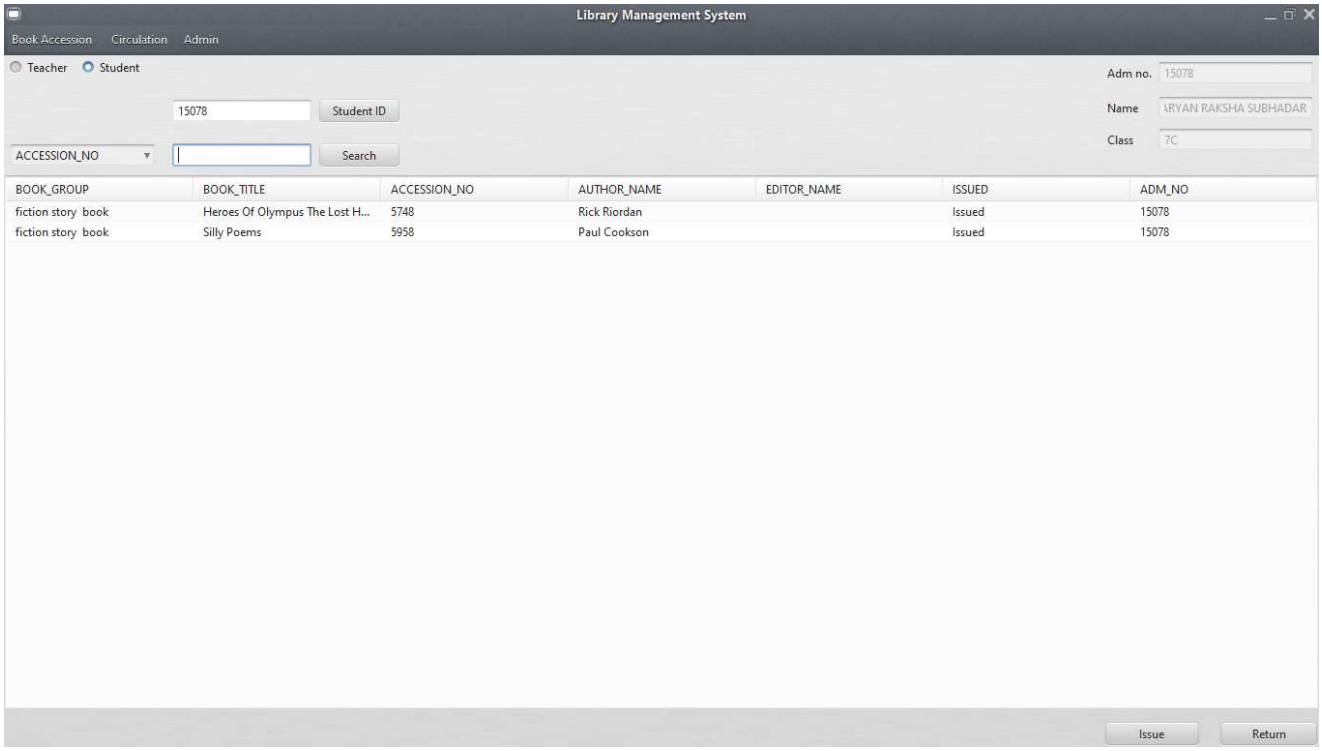
Issue / Return:



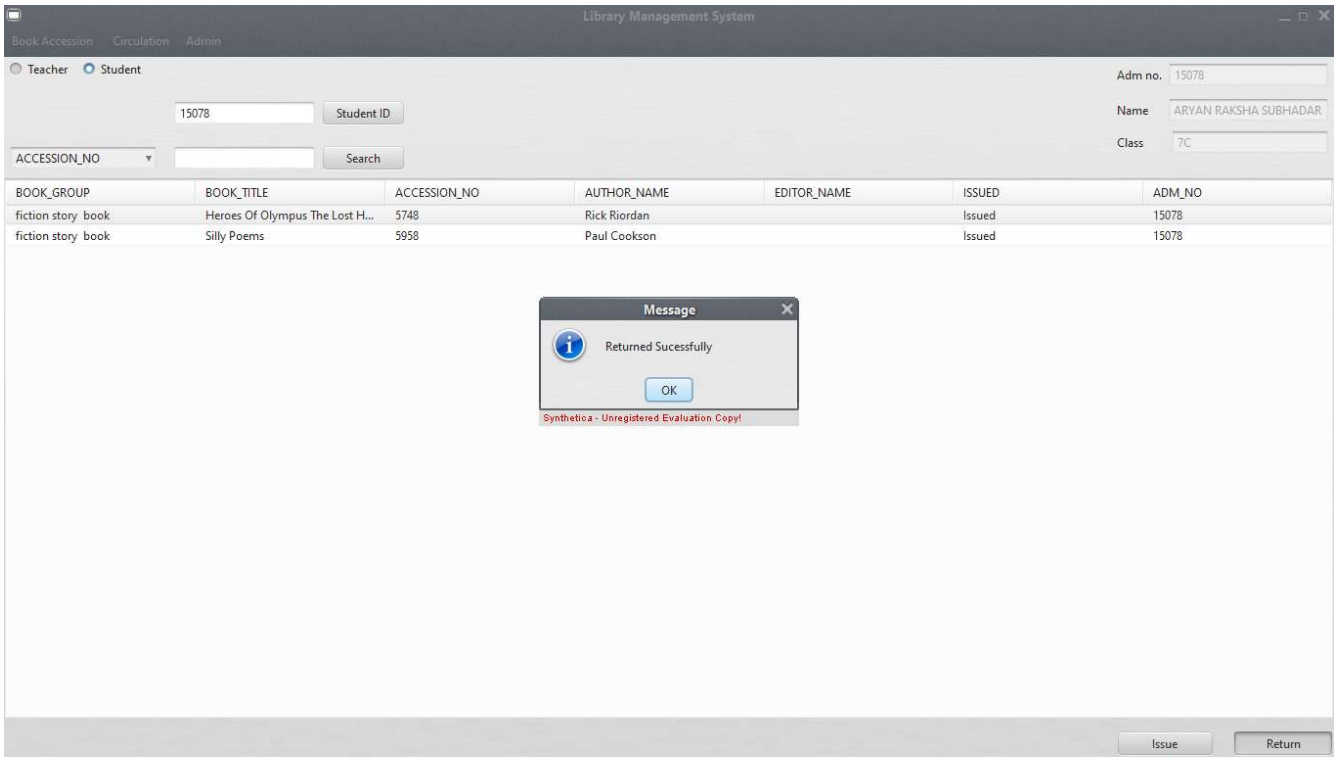
For Teachers:



If adm.no or User no. is searched the result is shown with the books issued by him:

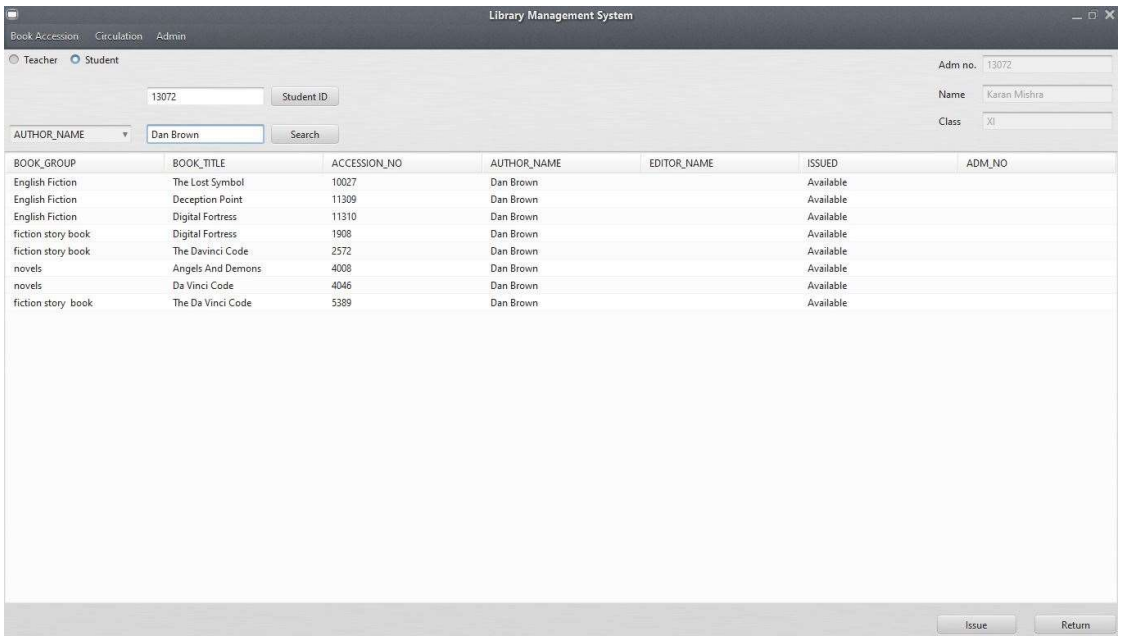


Book Returned Sucessfully :

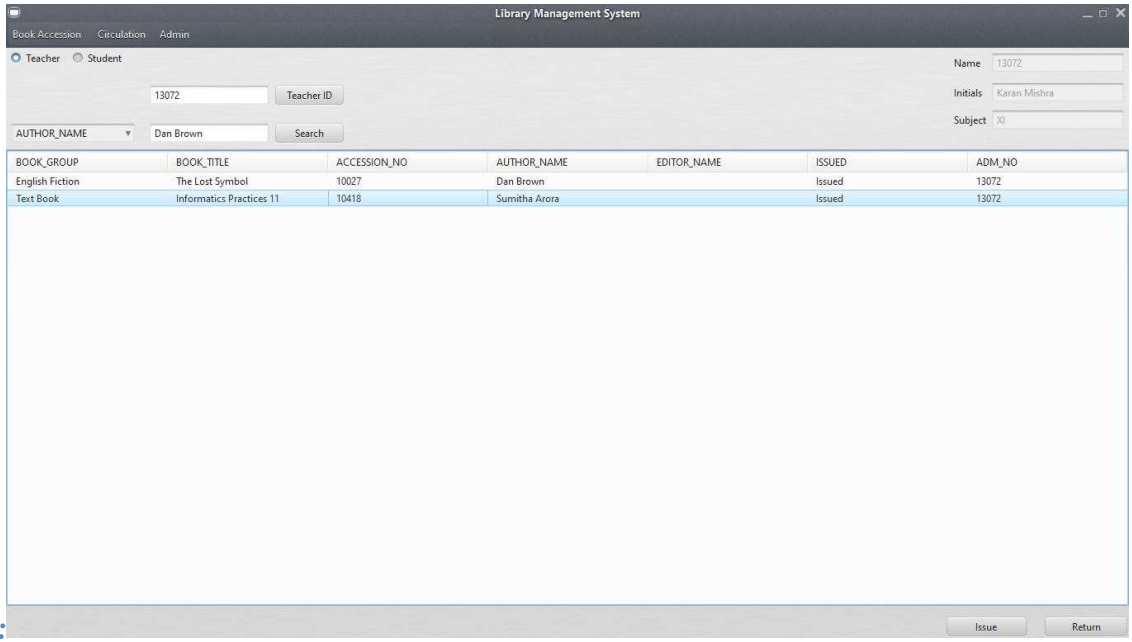


If user wants to return book issued by someone else:

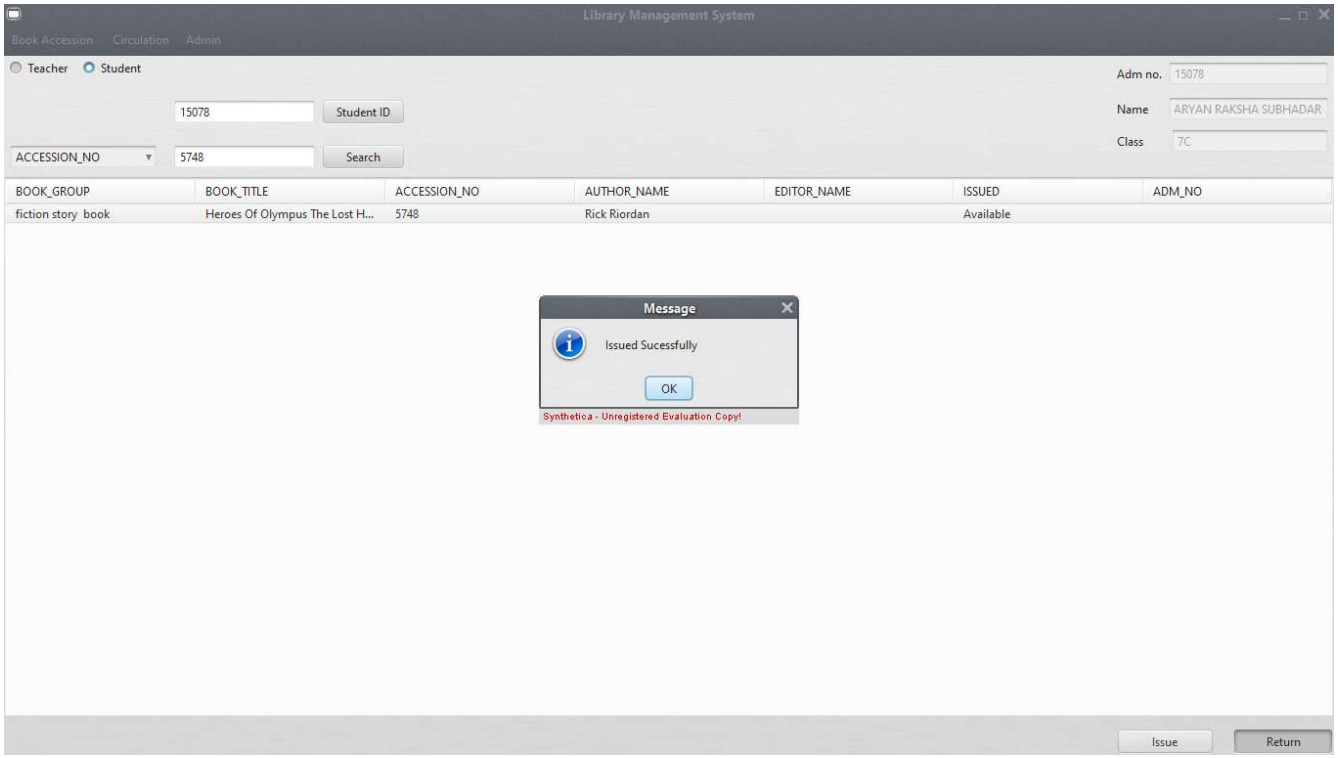
Search Book:



Select book:



Book Issued Sucessfully:



Note:-*The user can even issue book by just doubble clicking it.*

If user wants to issue Book which is issued by someone else:

CODE:

Imports:

```
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.Statement;
import java.awt.Cursor;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.ButtonGroup;
import javax.swing.DefaultComboBoxModel;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.JToggleButton;
import javax.swing.LayoutStyle;
import javax.swing.table.DefaultTableModel;
```

Students Radio Button:

```
595
596     private void StudentRadioButtonMousePressed(MouseEvent evt) {
597         this.IDButton.setText("Student ID");
598         this.InfoLabel1.setText("Adm no.");
599         this.InfoLabel2.setText("Name");
600         this.InfoLabel3.setText("Class");
601     }
602
```

Teachers Radio Button:

```
603     private void TeacherRadioButtonMousePressed(MouseEvent evt) {
604         this.IDButton.setText("Teacher ID");
605         this.InfoLabel1.setText("Name");
606         this.InfoLabel2.setText("Initials");
607         this.InfoLabel3.setText("Subject");
608     }
```

StudentID Method: (This is used to serach database with adm.no)

```
342     void StudentID(String id) {
343         try {
344             Class.forName("com.mysql.jdbc.Driver");
345             Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
346             Statement stmt = (Statement)conn.createStatement();
347             String sql = id.contains("D") || id.contains("d") ? "select * from students where adm_no "
348                     + "= '" + id + "';" : "select * from students where adm_no = " + id + ";";
349             ResultSet rs = stmt.executeQuery(sql);
350             rs.next();
351             this.InfoTextField1.setText(rs.getString("adm_no"));
352             this.InfoTextField2.setText(rs.getString("name"));
353             this.InfoTextField3.setText(rs.getString("std"));
354             rs.close();
355             stmt.close();
356             conn.close();
357         }
358         catch (SQLException e) {
359             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
360         }
361         catch (ClassNotFoundException e) {
362             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
363         }
364     }
```

TeacherID Method: (This is used to search database from UserID)

```
366     void TeacherID(String id) {
367         try {
368             Class.forName("com.mysql.jdbc.Driver");
369             Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
370             Statement stmt = (Statement)conn.createStatement();
371             String sql = "select * from teachers where sl_no = " + id + ";";
372             ResultSet rs = stmt.executeQuery(sql);
373             rs.next();
374             this.InfoTextField1.setText(rs.getString("Name"));
375             this.InfoTextField2.setText(rs.getString("Initial"));
376             this.InfoTextField3.setText(rs.getString("Subject"));
377             rs.close();
378             stmt.close();
379             conn.close();
380         }
381         catch (SQLException e) {
382             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
383         }
384         catch (ClassNotFoundException e) {
385             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
386         }
387     }
```

ID Search Button: (Used to search by mouse)

```
558     private void IDButtonActionPerformed(ActionEvent evt) {
559         this.id = this.IDTextField.getText();
560         DefaultTableModel model = (DefaultTableModel)this.SearchTable.getModel();
561         if (this.StudentRadioButton.isSelected()) {
562             this.StudentID(this.id);
563         } else if (this.TeacherRadioButton.isSelected()) {
564             this.TeacherID(this.id);
565         }
566         this.BooksIssued(this.id, model);
567     }
```

ID Search Enter Key: (To search by clicking enter)

```
622     private void IDTextFieldKeyPressed(KeyEvent evt) {
623         if (evt.getKeyCode() == 10) {
624             this.id = this.IDTextField.getText();
625             DefaultTableModel model = (DefaultTableModel)this.SearchTable.getModel();
626             if (this.StudentRadioButton.isSelected()) {
627                 this.StudentID(this.id);
628             } else if (this.TeacherRadioButton.isSelected()) {
629                 this.TeacherID(this.id);
630             }
631             this.BooksIssued(this.id, model);
632         }
633     }
```

BooksIssued Method: (Used to see issue books as soon as ID is searched)

```
460     int BooksIssued(String ADM_NO, DefaultTableModel model) {
461         while (model.getRowCount() > 0) {
462             model.setRowCount(0);
463         }
464         try {
465             Class.forName("com.mysql.jdbc.Driver");
466             Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
467             Statement stmt = (Statement)conn.createStatement();
468             String sql = "select * from books where adm_no = " + ADM_NO + ";";
469             ResultSet rs = stmt.executeQuery(sql);
470             while (rs.next()) {
471                 String BOOK_GROUP = rs.getString("BOOK_GROUP");
472                 String BOOK_TITLE = rs.getString("BOOK_TITLE");
473                 String ACCESSION_NO = rs.getString("ACCESSION_NO");
474                 String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
475                 String EDITOR_NAME = rs.getString("EDITOR_NAME");
476                 Boolean ISSUED = rs.getBoolean("ISSUED");
477                 String adm_no = rs.getString("ADM_NO");
478                 String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
479                 String ISSUE = ISSUED == true ? "Issued" : "Available";
480                 model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, adm_no, BOOK_LOCATION});
481             }
482             rs.close();
483             conn.close();
484             stmt.close();
485         }
486         catch (Exception e) {
487             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
488         }
489         return 0;
490     }
```

Search Button:

```
576
577     private void SearchButtonActionPerformed(ActionEvent evt) {
578         String item = (String)this.SearchBox.getSelectedItem();
579         String text = this.SearchTextField.getText();
580         DefaultTableModel model = (DefaultTableModel)this.SearchTable.getModel();
581         while (model.getRowCount() > 0) {
582             model.setRowCount(0);
583         }
584         new OpenPage().Search(item, text, model);
585     }
```

Search Enter Key:

```
610     private void SearchTextFieldKeyPressed(KeyEvent evt) {
611         if (evt.getKeyCode() == 10) {
612             String item = (String)this.SearchBox.getSelectedItem();
613             String text = this.SearchTextField.getText();
614             DefaultTableModel model = (DefaultTableModel)this.SearchTable.getModel();
615             while (model.getRowCount() > 0) {
616                 model.setRowCount(0);
617             }
618             new OpenPage().Search(item, text, model);
619         }
620     }
```

Mouse action at table: (To get data of selected book)

```
587     private void SearchTableMouseClicked(MouseEvent evt) {
588         this.row = this.SearchTable.getSelectedRow();
589         DefaultTableModel model = (DefaultTableModel)this.SearchTable.getModel();
590         this.acess = (String)model.getValueAt(this.row, 2);
591         if (evt.getClickCount() == 2) {
592             this.issue();
593         }
594     }
```

Issue Method:

```
389     void issue() {
390         admno = InfoTextField1.getText();
391         if (this.admno.length() == 0) {
392             JOptionPane.showMessageDialog(null, "please enter a adm_no");
393         } else if (this.StudentRadioButton.isSelected()) {
394             try {
395                 Class.forName("com.mysql.jdbc.Driver");
396                 Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
397                 Statement stmt = (Statement)conn.createStatement();
398                 String Sql = "select book_title,issued,adm_no from books where ACCESSION_NO = '" + acess + "';";
399                 String sql = "select name from students where adm_no = '" + admno + "';";
400                 ResultSet rs = stmt.executeQuery(Sql);
401                 rs.next();
402                 booktitle = rs.getString("Book_title");
403                 status = rs.getBoolean("issued");
404                 adm_no = rs.getString("adm_no");
405                 ResultSet rs1 = stmt.executeQuery(sql);
406                 rs1.next();
407                 name = rs1.getString("NAME");
408                 rs.close();
409                 if (status) {
410                     JOptionPane.showMessageDialog(null, "the book is already issued by admmission number " + adm_no);
411                 }
412                 String Sql1 = "update books set issued = true,adm_no='" + admno + "' where accession_no='" + acess + "';";
413                 String Sql2 = "insert into records(ISSUE_DATE,ADM_NO,NAME,BOOK_TITLE,ACCESSION_NO) values "
414                         + "(curdate(),'" +admno+ "','"+name+"','"+booktitle+"','"+acess+"');";
415                 stmt.executeUpdate(Sql1);
416                 stmt.executeUpdate(Sql2);
417                 stmt.close();
418                 conn.close();
419                 JOptionPane.showMessageDialog(null, "Issued Sucessfully");
420             }
421             catch (Exception ex) {
422                 JOptionPane.showMessageDialog(null, ex.getLocalizedMessage());
423             }
424         } else if (this.TeacherRadioButton.isSelected()) {
425             try {
426                 Class.forName("com.mysql.jdbc.Driver");
427                 Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
428                 Statement stmt = (Statement)conn.createStatement();
429                 String Sql = "select book_title,issued,adm_no from books where ACCESSION_NO = '" +acess + "';";
430                 String sql = "select name from teachers where sl_no = '" + admno + "';";
431                 ResultSet rs = stmt.executeQuery(Sql);
432                 rs.next();
433                 booktitle = rs.getString("Book title");
434                 status = rs.getBoolean("issued");
435                 adm_no = rs.getString("adm_no");
436                 ResultSet rs1 = stmt.executeQuery(sql);
437                 rs1.next();
438                 name = rs1.getString("NAME");
439                 rs.close();
440                 if (status) {
441                     JOptionPane.showMessageDialog(null, "the book is already issued by USER " +adm_no);
442                 }
443                 String Sql1 = "update books set issued = true,adm_no='" +admno + "' where accession_no='" +acess + "';";
444                 String Sql2 = "insert into records(ISSUE_DATE,NAME,BOOK_TITLE,ACCESSION_NO,USER_NO) values "
445                         + "(curdate(),'" + name + "','" + booktitle + "','" + acess + "','" + admno + "');";
446                 stmt.executeUpdate(Sql1);
447                 stmt.executeUpdate(Sql2);
448                 stmt.close();
449                 conn.close();
450                 JOptionPane.showMessageDialog(null, "Issued Sucessfully");
451             }
452             catch (Exception ex) {
453                 JOptionPane.showMessageDialog(null, ex.getLocalizedMessage());
454             }
455         } else {
456             JOptionPane.showMessageDialog(null, "Please select a Button");
457         }
458     }
```

Issue Button:

```
568
569     private void IssueButtonActionPerformed(ActionEvent evt) {
570         String item = (String)this.SearchBox.getSelectedItem();
571         String text = this.SearchTextField.getText();
572         DefaultTableModel model = (DefaultTableModel)this.SearchTable.getModel();
573         this.issue();
574         new OpenPage().Search(item, text, model);
575     }
576
```

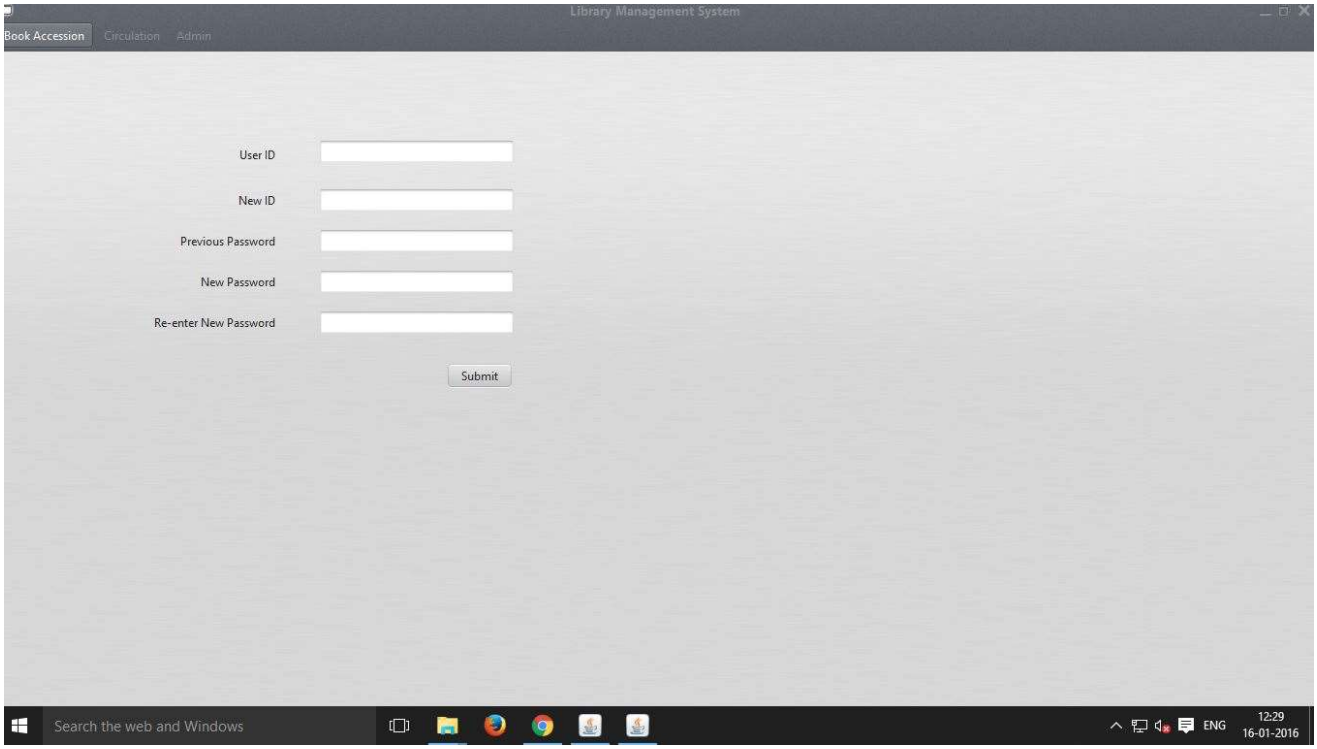
Return Method:

```
492     void RETURN() {
493         admno = InfoTextField1.getText();
494         if (admno.length() == 0) {
495             JOptionPane.showMessageDialog(null, "please enter a adm. no");
496         } else if (this.StudentRadioButton.isSelected()) {
497             try {
498                 Class.forName("com.mysql.jdbc.Driver");
                  Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
500                 Statement stmt = (Statement)conn.createStatement();
501                 String Sql = "select issued,adm_no from books where ACCESSION_NO = '" +acess + "';";
502                 ResultSet rs = stmt.executeQuery(Sql);
503                 rs.next();
504                 status = rs.getBoolean("issued");
505                 adm_no = rs.getString("adm_no");
506                 rs.close();
507                 if (status) {
508                     JOptionPane.showMessageDialog(null, "the book is already present in library");
509                 }
510                 if (adm_no.equals(admno)) {
511                     JOptionPane.showMessageDialog(null, "The book is issued by admmission number " + adm_no);
512                 }
513                 String Sql1 = "update books set issued = false,adm_no=null where accession_no='" + acess + "';";
514                 String Sql2 = "update records set RETURN_DATE = curdate() where adm_no = '"+ admno +"' and accession_no = '" +acess +"';";
515                 stmt.executeUpdate(Sql1);
516                 stmt.executeUpdate(Sql2);
517                 stmt.close();
518                 conn.close();
519                 JOptionPane.showMessageDialog(null, "Returned Sucessfully");
520                 }
              catch (Exception e) {
522                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
523             }
524         } else if (this.TeacherRadioButton.isSelected()) {
525             try {
526                 Class.forName("com.mysql.jdbc.Driver");
                  Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
528                 Statement stmt = (Statement)conn.createStatement();
529                 String Sql = "select issued,ADM_NO from books where ACCESSION_NO = '" +acess + "';";
530                 ResultSet rs = stmt.executeQuery(Sql);
531                 rs.next();
532                 status = rs.getBoolean("issued");
533                 adm_no = rs.getString("ADM_NO");
534                 rs.close();
535                 if (status) {
536                     JOptionPane.showMessageDialog(null, "the book is already present in library");
537                 }
538                 if (adm_no.equals(admno)) {
539                     JOptionPane.showMessageDialog(null, "The book is issued by USER " +adm_no);
540                 }
541                 String Sql1 = "update books set issued = false,ADM_NO=null where accession_no='" + acess + "';";
542                 String Sql2 = "update records set RETURN_DATE = curdate() "
543                         + "where user_id = '" + admno + "' and accession_no = '" + acess + "';";
544                 stmt.executeUpdate(Sql1);
545                 stmt.executeUpdate(Sql2);
546                 stmt.close();
547                 conn.close();
548                 JOptionPane.showMessageDialog(null, "Returned Sucessfully");
549             }
              catch (Exception e) {
551                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
552             }
553         } else {
554             JOptionPane.showMessageDialog(null, "Please select a Button");
555         }
556     }
557
```

Return Button:

```
635     private void ReturnButtonActionPerformed(ActionEvent evt) {
636         String item = (String)this.SearchBox.getSelectedItem();
637         String text = this.SearchTextField.getText();
638         DefaultTableModel model = (DefaultTableModel)this.SearchTable.getModel();
639         this.RETURN();
640         new OpenPage().Search(item, text, model);
641     }
```

Change Password:



if all Fields are not filled:

if previous id or password is not correct:

If new password and re-typed new password is not same:

password modified suessfully:

CODE:

Imports:

```
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.Statement;
import java.awt.Cursor;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.sql.DriverManager;
import java.sql.ResultSet;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPasswordField;
import javax.swing.JTextField;
```

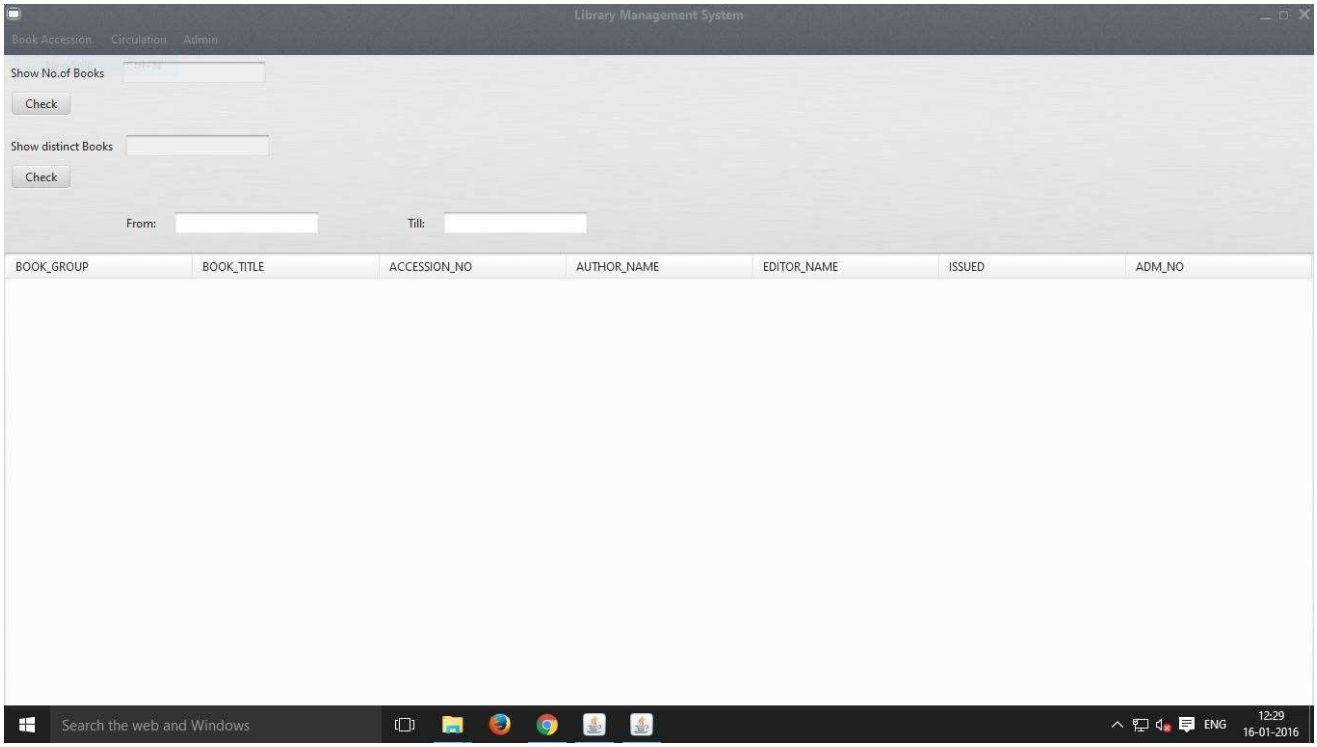
Submit Enter Key:

```
    private void RePassTextFieldKeyPressed(KeyEvent evt) {
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
            String UID = OldIDTextField.getText();
            String nuid = NewIDTextField.getText();
            String PWD = OldPassTextField.getText();
            String nPWD = NewPassTextFileld.getText();
            String rnPWD = RePassTextField.getText();
            if (UID.equals("") || nuid.equals("") || PWD.equals("") || nPWD.equals("") || rnPWD.equals("")) {
                JOptionPane.showMessageDialog(null, "Please fill all the fields");
            } else {
                try {
                    Class.forName("com.mysql.jdbc.Driver");
                    Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
                    Statement stmt = (Statement)conn.createStatement();
                    String sql = "select * from users;";
                    ResultSet rs = stmt.executeQuery(sql);
                    rs.next();
                    a = rs.getString("id");
                    b = rs.getString("pass");
                    rs.close();
                }
                catch (Exception e) {
                    JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
                }
                if (nPWD.equals(rnPWD)) {
                    if (UID.equals(a) && PWD.equals(b)) {
                        try {
                            Class.forName("com.mysql.jdbc.Driver");
                            Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
                            Statement stmt = (Statement)conn.createStatement();
                            String sql = "update users set id ='" + nuid + "',pass = '" + nPWD + "' where id = '" + a + "'; ";
                            stmt.executeUpdate(sql);
                            stmt.close();
                            conn.close();
                            JOptionPane.showMessageDialog(null, "modified Sucessfully");
                        }
                        catch (Exception e) {
                            JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
                        }
                    } else {
                        JOptionPane.showMessageDialog(null, "your id or password is in correct");
                    }
                } else {
                    JOptionPane.showMessageDialog(null, "password and re-enter password are not same");
                }}}}
```

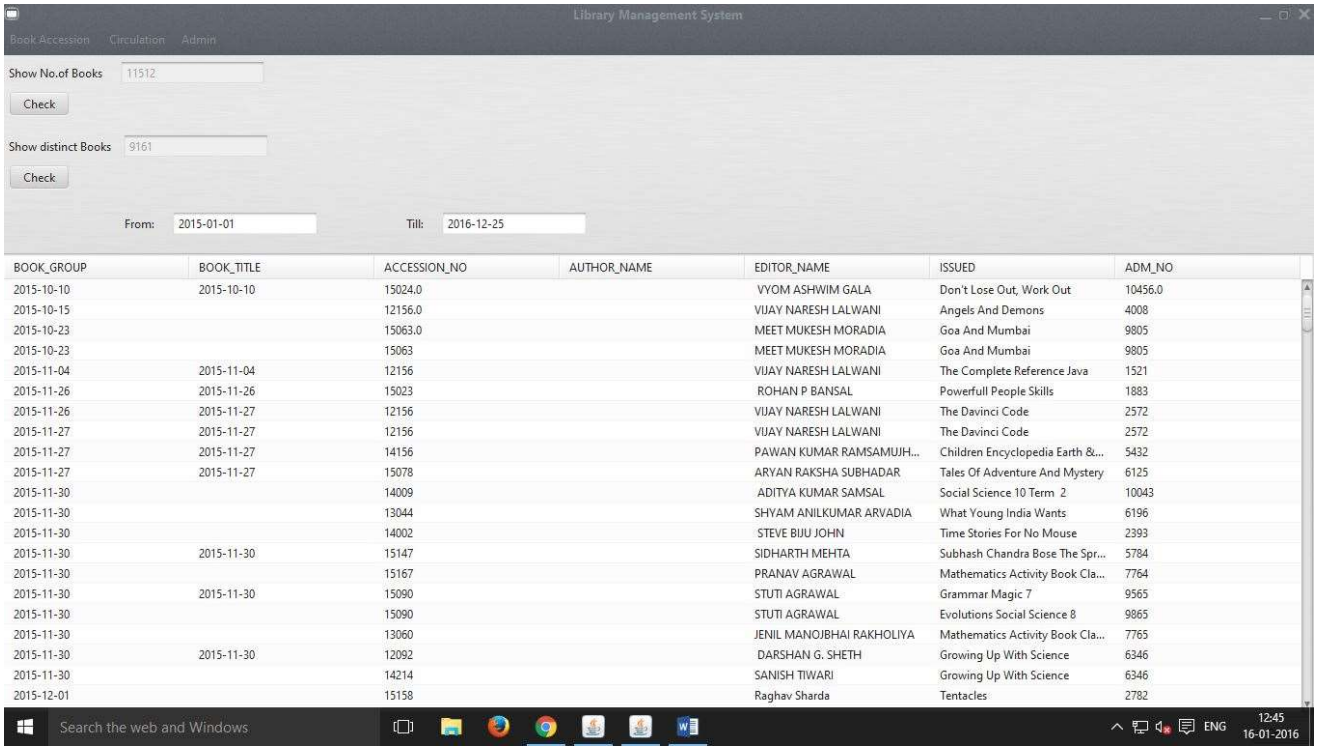
Submit Button:

```
236     private void SubmitButtonActionPerformed(ActionEvent evt) {
237         String UID = OldIDTextField.getText();
238         String nuid = NewIDTextField.getText();
239         String PWD = OldPassTextField.getText();
240         String nPWD = NewPassTextFileld.getText();
241         String rnPWD = RePassTextField.getText();
242         if (UID.equals("") || nuid.equals("") || PWD.equals("") || nPWD.equals("") || rnPWD.equals("")) {
243             JOptionPane.showMessageDialog(null, "Please fill all the fields");
244         } else {
245             try {
246                 Class.forName("com.mysql.jdbc.Driver");
247                 Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
248                 Statement stmt = (Statement)conn.createStatement();
249                 String sql = "select * from users;";
                    ResultSet rs = stmt.executeQuery(sql);
251                 rs.next();
252                 a = rs.getString("id");
253                 b = rs.getString("pass");
254                 rs.close();
255             }
                catch (Exception e) {
257                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
258             }
259             if (nPWD.equals(rnPWD)) {
260                 if (UID.equals(a) && PWD.equals(b)) {
261                     try {
262                         Class.forName("com.mysql.jdbc.Driver");
                            Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
264                         Statement stmt = (Statement)conn.createStatement();
265                         String sql = "update users set id ='" + nuid + "',pass = '" + nPWD + "' where id = '" + a + "'; ";
266                         stmt.executeUpdate(sql);
267                         stmt.close();
268                         conn.close();
269                         JOptionPane.showMessageDialog(null, "modified Sucessfully");
270                     }
                        catch (Exception e) {
272                         JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
273                     }
274                 } else {
275                     JOptionPane.showMessageDialog(null, "your id or password is in correct");
276                 }
277             } else {
278                 JOptionPane.showMessageDialog(null, "password and re-enter password are not same");
279             }
280         }
281     }
```

Books:



Show Books from(Date)-till(Date):



CODE:

Imports:

```
3  import java.awt.Cursor;
4  import java.awt.EventQueue;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7  import java.awt.event.KeyAdapter;
8  import java.awt.event.KeyEvent;
9  import java.sql.Connection;
10 import java.sql.DriverManager;
11 import java.sql.ResultSet;
12 import java.sql.SQLException;
13 import java.sql.Statement;
14 import javax.swing.GroupLayout;
15 import javax.swing.JButton;
16 import javax.swing.JFrame;
17 import javax.swing.JLabel;
18 import javax.swing.JMenu;
19 import javax.swing.JMenuBar;
20 import javax.swing.JMenuItem;
21 import javax.swing.JOptionPane;
22 import javax.swing.JScrollPane;
23 import javax.swing.JTable;
24 import javax.swing.JTextField;
25 import javax.swing.LayoutStyle;
26 import javax.swing.table.DefaultTableModel;
```

Show No.Of Books Button:

```
281     private void NoOfBooksButtonActionPerformed(ActionEvent evt) {
282         try {
283             Class.forName("com.mysql.jdbc.Driver");
284             Connection conn = DriverManager.getConnection(db_url, user, pwd);
285             Statement stmt = conn.createStatement();
286             String Sql = "select count(*)from books";
287             ResultSet rs = stmt.executeQuery(Sql);
288             rs.next();
289             NoOfBooksTextField.setText(rs.getString("count(*)"));
290             rs.close();
291             stmt.close();
292             conn.close();
293         }
294         catch (Exception e) {
295             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
296         }
297     }
```

Show No.Of Distinct Books Button:

```
299     private void DistinctBooksButtonActionPerformed(ActionEvent evt) {
300         try {
301             Class.forName("com.mysql.jdbc.Driver");
                Connection conn = DriverManager.getConnection(db_url,user,pwd);
303             Statement stmt = conn.createStatement();
304             String Sql = "select count(distinct book_title) from books;";
305             ResultSet rs = stmt.executeQuery(Sql);
306             rs.next();
307             DistinctBooksTextField.setText(rs.getString("count(distinct book_title)"));
308             rs.close();
309             stmt.close();
310             conn.close();
311         }
                catch (Exception e) {
313             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
314         }
315     }
```

Result:

```
317     private void ResultButtonActionPerformed(ActionEvent evt) {
318         String from = FromTextField.getText();
319         String till = TillTextField.getText();
320         DefaultTableModel model = (DefaultTableModel)ResultTable.getModel();
321         while (model.getRowCount() > 0) {
322             model.setRowCount(0);
323         }
324         if (from.equals("") || till.equals("")) {
325             JOptionPane.showMessageDialog(null, "Please enter all the fields");
326         } else {
327             try {
328                 Class.forName("com.mysql.jdbc.Driver");
                    Connection conn = DriverManager.getConnection(db_url,user, pwd);
330                 Statement stmt = conn.createStatement();
331                 String Sql = "select*from records where issue_date between '" + from + "' and '" + till + "';";
332                 ResultSet rs = stmt.executeQuery(Sql);
333                 while (rs.next()) {
334                     String ISSUE_DATE = rs.getString("ISSUE_DATE");
335                     String RETURN_DATE = rs.getString("RETURN_DATE");
336                     String ADM_NO = rs.getString("ADM_NO");
337                     String NAME = rs.getString("NAME");
338                     String BOOK_TITLE = rs.getString("BOOK_TITLE");
339                     String ACCESSION_NO = rs.getString("ACCESSION_NO");
340                     String USER_NO = rs.getString("USER_NO");
341                     model.addRow(new Object[]{ISSUE_DATE, RETURN_DATE, ADM_NO, USER_NO, NAME, BOOK_TITLE, ACCESSION_NO});
342                 }
343                 rs.close();
344                 conn.close();
345                 stmt.close();
346             }
                    catch (Exception e) {
348                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
349             }
350         }
351     }
```

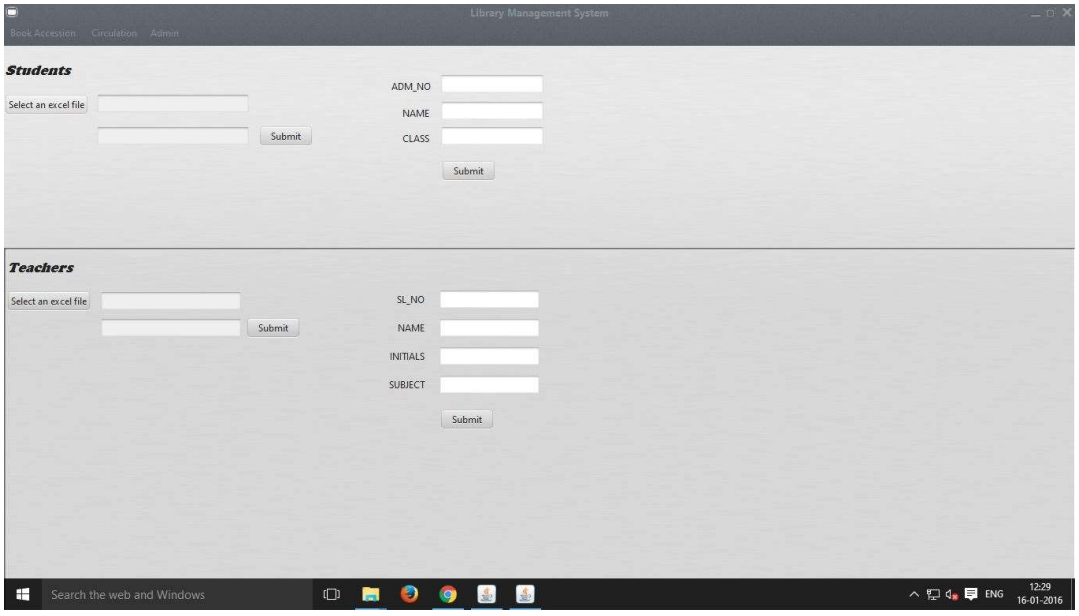
From Enter Key Pressed:

```
394     private void FromTextFieldKeyPressed(KeyEvent evt) {
395         if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
396             String from = FromTextField.getText();
397             String till = TillTextField.getText();
398             if (from.equals("") || till.equals("")) {
399                 JOptionPane.showMessageDialog(null, "Please enter all the fields");
400             } else {
401                 DefaultTableModel model = (DefaultTableModel)ResultTable.getModel();
402                 while (model.getRowCount() > 0) {
403                     model.setRowCount(0);
404                 }
405                 try {
406                     Class.forName("com.mysql.jdbc.Driver");
                      Connection conn = DriverManager.getConnection(db_url,user,pwd);
408                     Statement stmt = conn.createStatement();
409                     String Sql = "select*from records where issue_date between '" + from + "' and '" + till + "';";
410                     ResultSet rs = stmt.executeQuery(Sql);
411                     while (rs.next()) {
412                         String ISSUE_DATE = rs.getString("ISSUE_DATE");
413                         String RETURN_DATE = rs.getString("RETURN_DATE");
414                         String ADM_NO = rs.getString("ADM_NO");
415                         String NAME = rs.getString("NAME");
416                         String BOOK_TITLE = rs.getString("BOOK_TITLE");
417                         String ACCESSION_NO = rs.getString("ACCESSION_NO");
418                         String USER_NO = rs.getString("USER_NO");
419                         model.addRow(new Object[]{ISSUE_DATE, RETURN_DATE, ADM_NO, USER_NO, NAME, BOOK_TITLE, ACCESSION_NO});
420                     }
421                     rs.close();
422                     conn.close();
423                     stmt.close();
424                 }
425                 catch (SQLException e) {
426                     JOptionPane.showMessageDialog(null, e.getLocalizedMessage() + "The order is Year-Month-Date");
427                 }
428                 catch (ClassNotFoundException e) {
429                     JOptionPane.showMessageDialog(null, "Not able to load mysql probally lib folder deleted");
430                 }
431             }
432         }
433     }
```

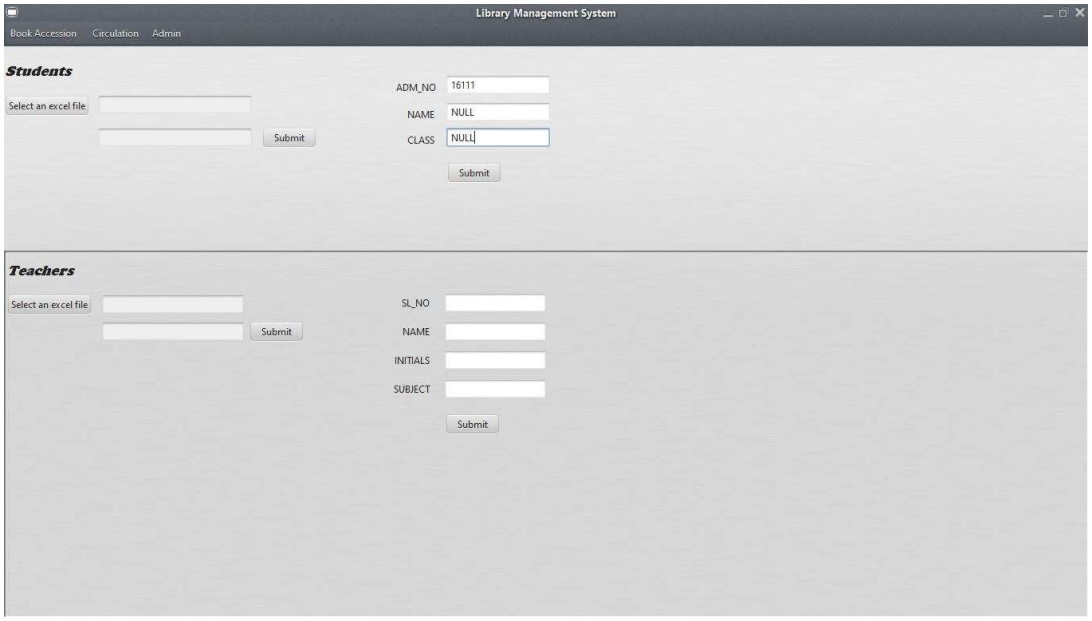
Till Enter Key Pressed:

```
353     private void TillTextFieldKeyPressed(KeyEvent evt) {
354         if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
355             String from = FromTextField.getText();
356             String till = TillTextField.getText();
357             if (from.equals("") || till.equals("")) {
358                 JOptionPane.showMessageDialog(null, "Please enter all the fields");
359             } else {
360                 DefaultTableModel model = (DefaultTableModel)ResultTable.getModel();
361                 while (model.getRowCount() > 0) {
362                     model.setRowCount(0);
363                 }
364                 try {
365                     Class.forName("com.mysql.jdbc.Driver");
                     Connection conn = DriverManager.getConnection(db_url,user,pwd);
367                     Statement stmt = conn.createStatement();
368                     String Sql = "select*from records where issue_date between '" + from + "' and '" + till + "';";
369                     ResultSet rs = stmt.executeQuery(Sql);
370                     while (rs.next()) {
371                         String ISSUE_DATE = rs.getString("ISSUE_DATE");
372                         String RETURN_DATE = rs.getString("RETURN_DATE");
373                         String ADM_NO = rs.getString("ADM_NO");
374                         String NAME = rs.getString("NAME");
375                         String BOOK_TITLE = rs.getString("BOOK_TITLE");
376                         String ACCESSION_NO = rs.getString("ACCESSION_NO");
377                         String USER_NO = rs.getString("USER_NO");
378                         model.addRow(new Object[]{ISSUE_DATE, RETURN_DATE, ADM_NO, USER_NO, NAME, BOOK_TITLE, ACCESSION_NO});
379                     }
380                     rs.close();
381                     conn.close();
382                     stmt.close();
383                 }
384                 catch (SQLException e) {
385                     JOptionPane.showMessageDialog(null, e.getLocalizedMessage() + "The order is Year-Month-Date");
386                 }
387                 catch (ClassNotFoundException e) {
388                     JOptionPane.showMessageDialog(null, "Not able to load mysql probally lib folder deleted");
389                 }
390             }
391         }
392     }
```

Update Databases:

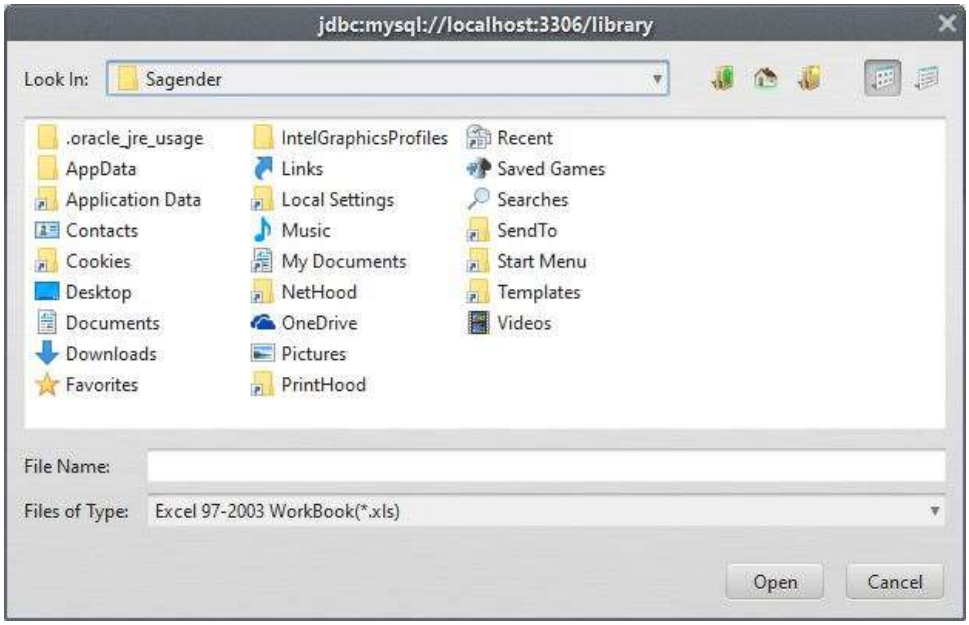


Adding indudia student:

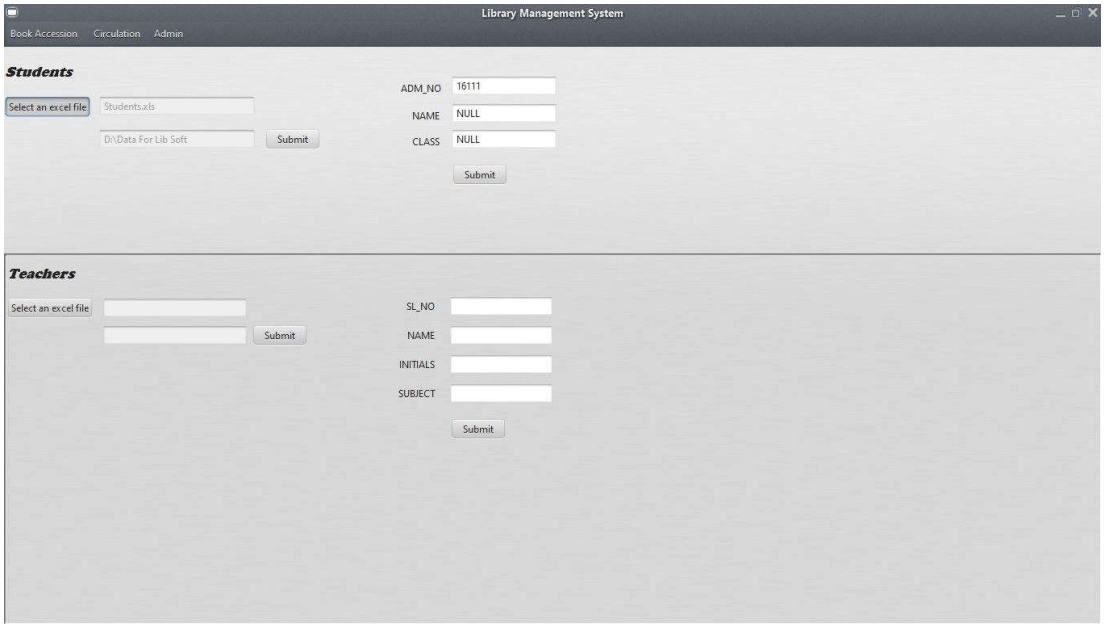


After Update Successful (indidual):

Selecting Excel file for updating:



File selected:



Database Upated:

CODE:

Imports:

```
import java.awt.Cursor;
import java.awt.EventQueue;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Iterator;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JToggleButton;
import javax.swing.LayoutStyle;
import javax.swing.border.SoftBevelBorder;
import javax.swing.filechooser.FileNameExtensionFilter;
import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.xssf.usermodel.XSSFCell;
import org.apache.poi.xssf.usermodel.XSSFRow;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
```

Student Select Excel File Button: \

```
private void StudentEXCELSelectButtonActionPerformed(ActionEvent evt) {
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setCurrentDirectory(new File(System.getProperty("user.home")));
    fileChooser.setDialogTitle("jdbc:mysql://localhost:3306/library");
    fileChooser.setFileSelectionMode(0);
    fileChooser.setFileFilter(new FileNameExtensionFilter("Excel WorkBook(*.xlsx)", "xlsx"));
    fileChooser.setFileFilter(new FileNameExtensionFilter("Excel 97-2003 WorkBook(*.xls)", "xls"));
    fileChooser.setAcceptAllFileFilterUsed(true);
    int result = fileChooser.showOpenDialog(this);
    if (result == 0) {
        filename.setText(fileChooser.getSelectedFile().getName());
        dir.setText(fileChooser.getCurrentDirectory().toString());
        abcd = fileChooser.getCurrentDirectory().toString() + "\\" + fileChooser.getSelectedFile().getName();
        ext = fileChooser.getTypeDescription(fileChooser.getSelectedFile());
    }
    if (result == 1) {
        filename.setText("You pressed cancel");
        dir.setText("");
    }
}
```

Teacher Select Excel File Button:

Submit Excel File For Students:

```
564     private void StudentSubmitButtonActionPerformed(ActionEvent evt) {
565         String ADM_NO = StudentAdm_NoTextField.getText();
566         String NAME = StudentNameTextField.getText();
567         String STD = StudentClassTextField.getText();
568         if (ADM_NO.equals("") || NAME.equals("") || STD.equals("")) {
569             JOptionPane.showMessageDialog(null, "Please enter all the fields");
570         } else {
571             try {
572                 Class.forName("com.mysql.jdbc.Driver");
                    Connection conn = DriverManager.getConnection(db_url,user,pwd);
574                 Statement stmt = conn.createStatement();
575                 String sql = "Insert into Students(ADM_NO,NAME,STD) values ('" + ADM_NO + "','" + NAME + "','" + STD + "');";
576                 stmt.executeUpdate(sql);
577                 JOptionPane.showMessageDialog(null, "Added Successfully");
578                 stmt.close();
579                 conn.close();
580             }
                catch (SQLException e) {
582                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
583             }
584             catch (ClassNotFoundException e) {
585                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
586             }
587         }
588     }
```

Submit Excel File for Teachers:

```
758     private void TeacherSubmitTextFieldKeyPressed(KeyEvent evt) {
759         if (evt.getKeyCode() == 10) {
760             String SL_NO = TeacherSl_NoTextField.getText();
761             String NAME = TeacherNameTextField.getText();
762             String INITIAL = TeacherInitalsTextField.getText();
763             String SUBJECT = TeacherSubmitTextField.getText();
764             if (SL_NO.equals("") || NAME.equals("") || INITIAL.equals("") || SUBJECT.equals("")) {
765                 JOptionPane.showMessageDialog(null, "Please enter all the fields");
766             } else {
767                 try {
768                     Class.forName("com.mysql.jdbc.Driver");
                        Connection conn = DriverManager.getConnection(db_url,user,pwd);
770                     Statement stmt = conn.createStatement();
771                     String sql = "Insert into Teachers(NAME,SL_NO,INITIAL,SUBJECT) values ('" + NAME + "','" + SL_NO + "','" + INITIAL + "'"
                                + ",'" + SUBJECT + "');";
773                     JOptionPane.showMessageDialog(null, "Added Successfully");
774                     stmt.executeUpdate(sql);
775                     stmt.close();
776                     conn.close();
777                 }
                    catch (Exception e) {
779                     JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
780                 }
781             }
782         }
783     }
784
```

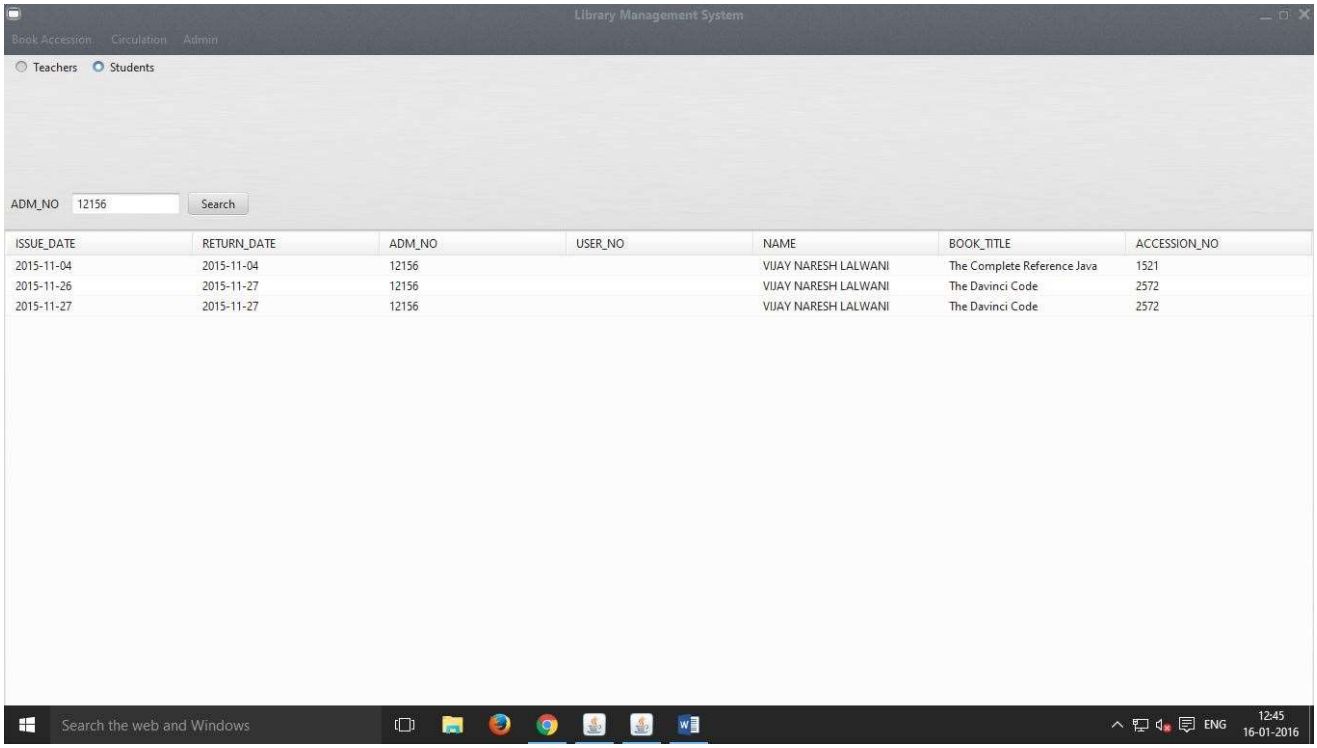
Submit excel file for Students:

```
389  private void StudentEXCELSubmitButtonActionPerformed(ActionEvent evt) {
390
391          try {
392                  FileInputStream file = new FileInputStream(new File(abcd));
393                  XSSFWorkbook yourworkbook = new XSSFWorkbook((InputStream)file);
394                  XSSFSheet sheet1 = yourworkbook.getSheetAt(0);
395                  Iterator rows = sheet1.rowIterator();
                     XSSFRow row = (XSSFRow)rows.next();
397                  int rowNum = sheet1.getLastRowNum();
398                  short colNum = sheet1.getRow(0).getLastCellNum();
399                  int j = 1;
400                  Class.forName("com.mysql.jdbc.Driver");
                     Connection conn = DriverManager.getConnection(db_url,user,pwd);
402                  Statement stmt = conn.createStatement();
403                  String Sql1 = "truncate Students;";
404                  stmt.executeUpdate(Sql1);
405                  for (int i = 0; i < rowNum; ++i) {
406                      String[] array = new String[colNum];
407                      int a = 0;
408                      try {
409                          while (rows.hasNext()) {
410                              row = sheet1.getRow(j);
411                              XSSFCell column = row.getCell(a);
412                              column.setCellType(1);
413                              if (column.getCellType() == 1) {
414                                  array[a] = column.getStringCellValue();
415                              } else if (column.getCellType() == 3) {
416                                  array[a] = "";
417                              }
418                              ++a;
419                          }
420                      }
421                      catch (NullPointerException column) {
422                          // empty catch block
423                      }
424                      try {
                             String sql = "Insert into students(ADM_NO,NAME,STD) values ('" + array[0] + "','" + array[1] + "','" + array[2] + "');";
426                          stmt.executeUpdate(sql);
427                      }
428                      catch (SQLException e) {
429                          JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
430                      }
431                      ++j;
432                  }
433                  stmt.close();
434                  conn.close();
435                  JOptionPane.showMessageDialog(null, "Database Updated");
436              }
              catch(Exception e) {
438               JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
439          }
440  }
```

Submit excel file for teachers:

```
463     private void TeachersEXCELSubmitButtonActionPerformed(ActionEvent evt) {
464             try {
465                     FileInputStream file = new FileInputStream(new File(abcd));
466                     XSSFWorkbook yourworkbook = new XSSFWorkbook((InputStream)file);
467                     XSSFSheet sheet1 = yourworkbook.getSheetAt(0);
468                     Iterator rows = sheet1.rowIterator();
                        XSSFRow row = (XSSFRow)rows.next();
470                     int rowNum = sheet1.getLastRowNum();
471                     short colNum = sheet1.getRow(0).getLastCellNum();
472                     int j = 1;
473                     Class.forName("com.mysql.jdbc.Driver");
                        Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root", "mysqlip");
475                     Statement stmt = conn.createStatement();
476                     String Sql1 = "truncate Teachers;";
477                     stmt.executeUpdate(Sql1);
478                     for (int i = 0; i < rowNum; ++i) {
479                         String[] array = new String[colNum];
480                         int a = 0;
481                         try {
482                             while (rows.hasNext()) {
483                                 row = sheet1.getRow(j);
484                                 XSSFCell column = row.getCell(a);
485                                 column.setCellType(1);
486                                 if (column.getCellType() == 1) {
487                                     array[a] = column.getStringCellValue();
488                                 } else if (column.getCellType() == 3) {
489                                     array[a] = "";
490                                 }
491                                 ++a;
492                             }
493                         }
494                         catch (NullPointerException column) {
495                             // empty catch block
496                         }
497                         try {
498                             String sql = "Insert into Teachers(SL_NO,NAME,INITIAL,SUBJECT) values ('" + array[0] + "',"
                                        + "'" + array[1] + "','" + array[2] + "','" + array[3] + "');";
500                             stmt.executeUpdate(sql);
501                         }
502                         catch (SQLException e) {
503                             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
504                         }
505                         ++j;
506                     }
507                     stmt.close();
508                     conn.close();
509                     JOptionPane.showMessageDialog(null, "Database Updated");
510
511                 }
513             catch (Exception e) {
514                 JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
515             }
516
517     }
```

Users Data:


Library Management System
Book Accession
Circulation
Admin
Teachers
Students
ADM_NO
12156
Search
ISSUE_DATE
RETURN_DATE
ADM_NO
USER_NO
NAME
BOOK_TITLE
ACCESSION_NO
2015-11-04
2015-11-04
12156
VIJAY NARESH LALWANI
The Complete Reference Java
1521
2015-11-26
2015-11-27
12156
VIJAY NARESH LALWANI
The Davinci Code
2572
2015-11-27
2015-11-27
12156
VIJAY NARESH LALWANI
The Davinci Code
2572
Search the web and Windows
ENG
12:45
16-01-2016

CODE:

Imports:

```
3  import com.mysql.jdbc.Connection;
4  import com.mysql.jdbc.Statement;
5  import java.awt.Cursor;
6  import java.awt.EventQueue;
7  import java.awt.event.ActionEvent;
8  import java.awt.event.ActionListener;
9  import java.awt.event.MouseAdapter;
10 import java.awt.event.MouseEvent;
11 import java.sql.DriverManager;
12 import java.sql.ResultSet;
13 import java.sql.SQLException;
14 import javax.swing.ButtonGroup;
15 import javax.swing.GroupLayout;
16 import javax.swing.JButton;
17 import javax.swing.JFrame;
18 import javax.swing.JLabel;
19 import javax.swing.JMenu;
20 import javax.swing.JMenuBar;
21 import javax.swing.JMenuItem;
22 import javax.swing.JOptionPane;
23 import javax.swing.JRadioButton;
24 import javax.swing.JScrollPane;
25 import javax.swing.JTable;
26 import javax.swing.JTextField;
27 import javax.swing.LayoutStyle;
28 import javax.swing.table.DefaultTableModel;
```

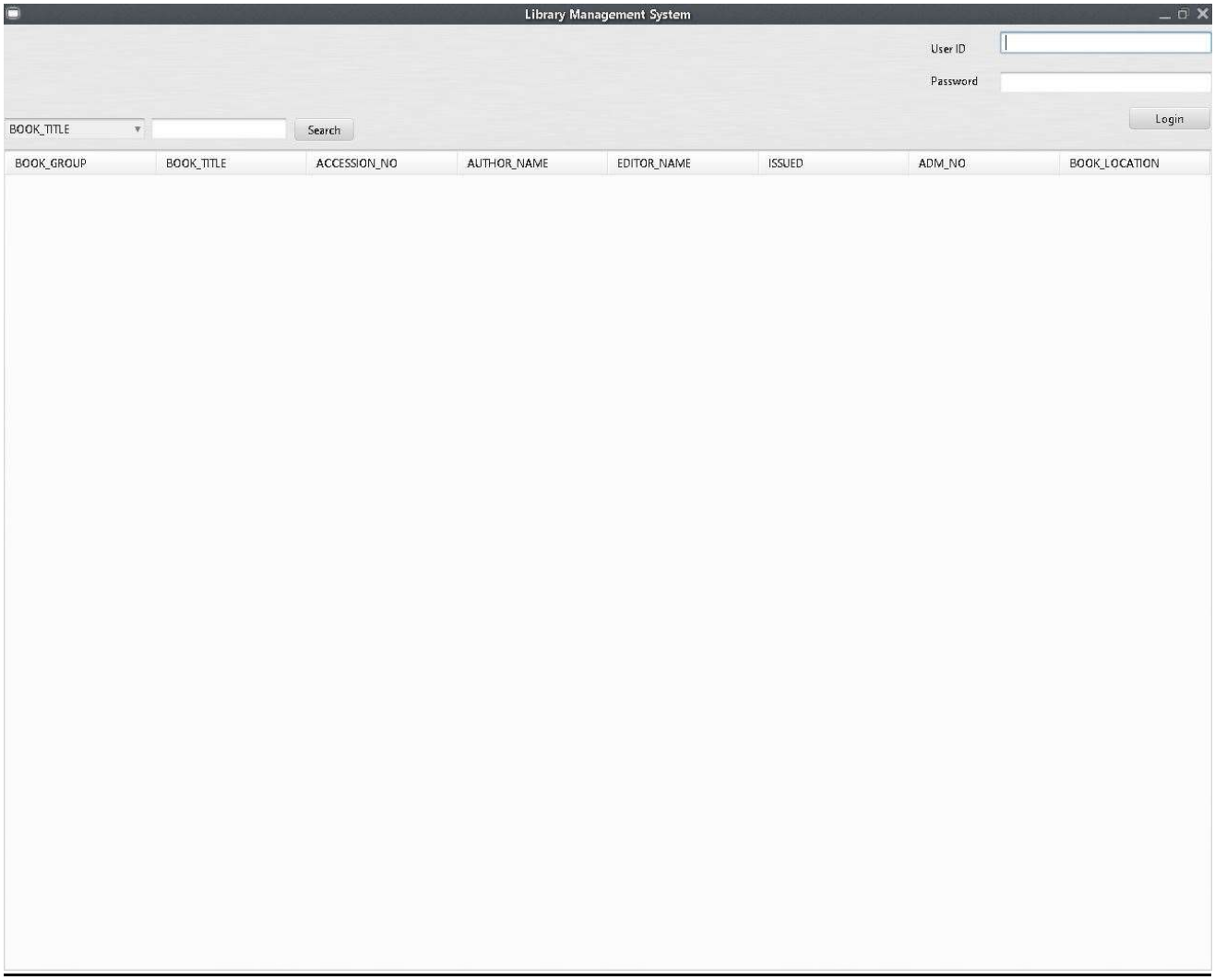
Radio Button:

```
289     private void TeacherRadioButtonMouseClicked(MouseEvent evt) {
290         IDLabel.setText("User no.");
291     }
292
293     private void StudentRadioButtonMouseClicked(MouseEvent evt) {
294         IDLabel.setText("ADM_NO");
295     }
```

Search ID:

```
249     private void IDSearchButtonActionPerformed(ActionEvent evt) {
250         String search = IDTextField.getText();
251         DefaultTableModel model = (DefaultTableModel)SearchResult.getModel();
252         while (model.getRowCount() > 0) {
253             model.setRowCount(0);
254         }
255         try {
256             Class.forName("com.mysql.jdbc.Driver");
            Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
258             Statement stmt = (Statement)conn.createStatement();
259             if (TeacherRadioButton.isSelected()) {
260                 sql = "select * from records where USER_NO = '" + search + "';";
261             } else if (StudentRadioButton.isSelected()) {
262                 sql = "select * from records where ADM_NO = '" + search + "';";
263             } else {
264                 JOptionPane.showMessageDialog(null, "Please Select a user first");
265             }
266             ResultSet rs = stmt.executeQuery(sql);
267             while (rs.next()) {
268                 String ISSUE_DATE = rs.getString("ISSUE_DATE");
269                 String RETURN_DATE = rs.getString("RETURN_DATE");
270                 String ADM_NO = rs.getString("ADM_NO");
271                 String NAME = rs.getString("NAME");
272                 String BOOK_TITLE = rs.getString("BOOK_TITLE");
273                 String ACCESSION_NO = rs.getString("ACCESSION_NO");
274                 String USER_NO = rs.getString("USER_NO");
275                 model.addRow(new Object[]{ISSUE_DATE, RETURN_DATE, ADM_NO, USER_NO, NAME, BOOK_TITLE, ACCESSION_NO});
276             }
277             rs.close();
278             conn.close();
279             stmt.close();
280         }
        catch (SQLException e) {
282             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
283         }
284         catch (ClassNotFoundException e) {
285             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
286         }
287     }
```

LogOut:



CODE:

```
552     }
553     private void LogOutMenuItemActionPerformed(ActionEvent evt) {
554         new OpenPage().setVisible(true);
555         this.dispose();
556     }
557 //menu actions ends
```

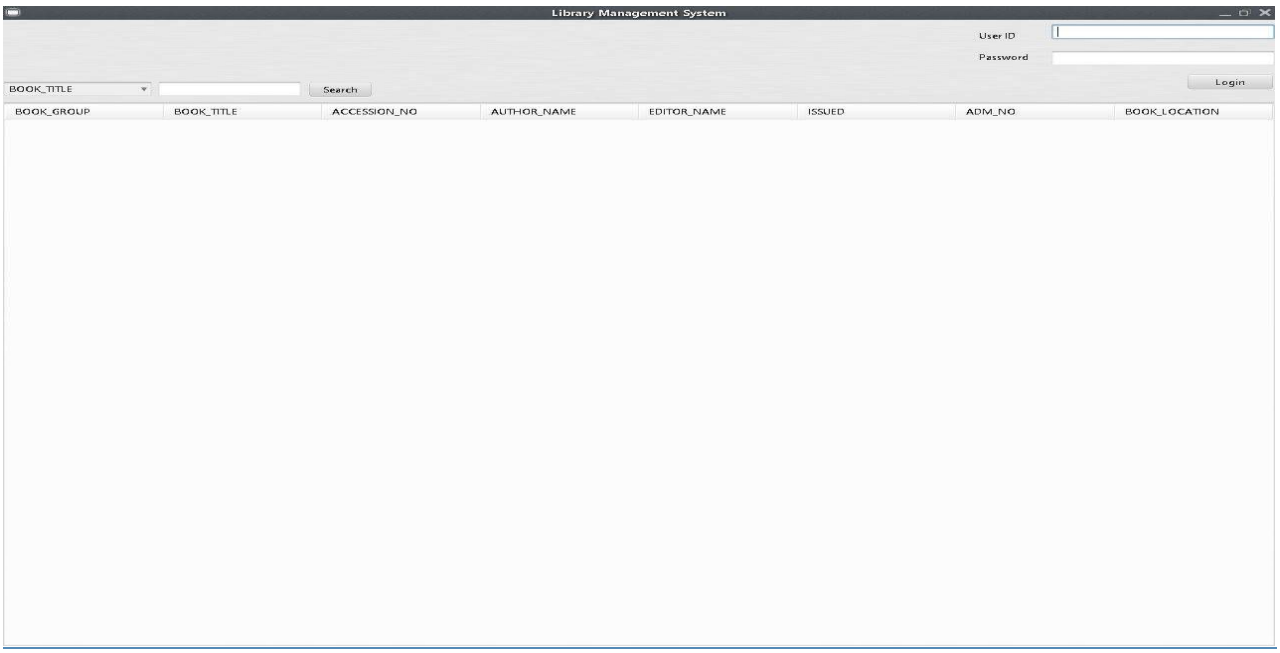
JAVA WORKING FOR CLIENT

LIBRARIES

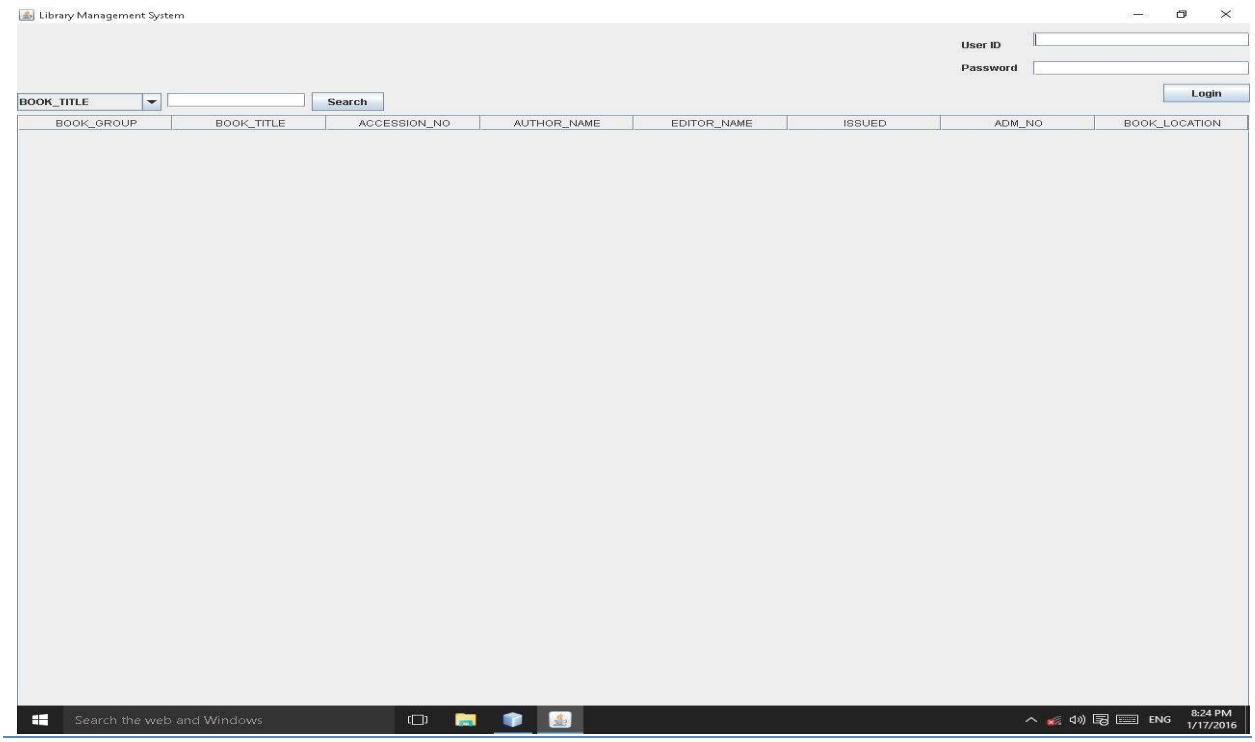
1. MySQL-connector-java - This library is used to connect java to MySQL so that the exchange of the data can take place. Example of this library is given below

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = (Connection)DriverManager.getConnection("jdbc:mysql://localhost:3306/library", "root", "mysqlip");
    Statement stmt = (Statement)conn.createStatement();
    String sql = "select * from books where " + item + " like'%" + text + "%';";
    ResultSet rs = stmt.executeQuery(sql);
    while (rs.next()) {
        String BOOK_GROUP = rs.getString("BOOK_GROUP");
        String BOOK_TITLE = rs.getString("BOOK_TITLE");
        String ACCESSION_NO = rs.getString("ACCESSION_NO");
        String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
        String EDITOR_NAME = rs.getString("EDITOR_NAME");
        Boolean ISSUED = rs.getBoolean("ISSUED");
        String ADM_NO = rs.getString("ADM_NO");
        String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
        String ISSUE = ISSUED == true ? "Issued" : "Available";
        model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, ADM_NO, BOOK_LOCATION});
    }
    rs.close();
    conn.close();
    stmt.close();
}
catch (Exception e) {
    JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
}
```

2. *Synthetica and syntheticaAluOxide -* This Libraries is used to change the look and feel of the application. It makes the application look much better as you can see below

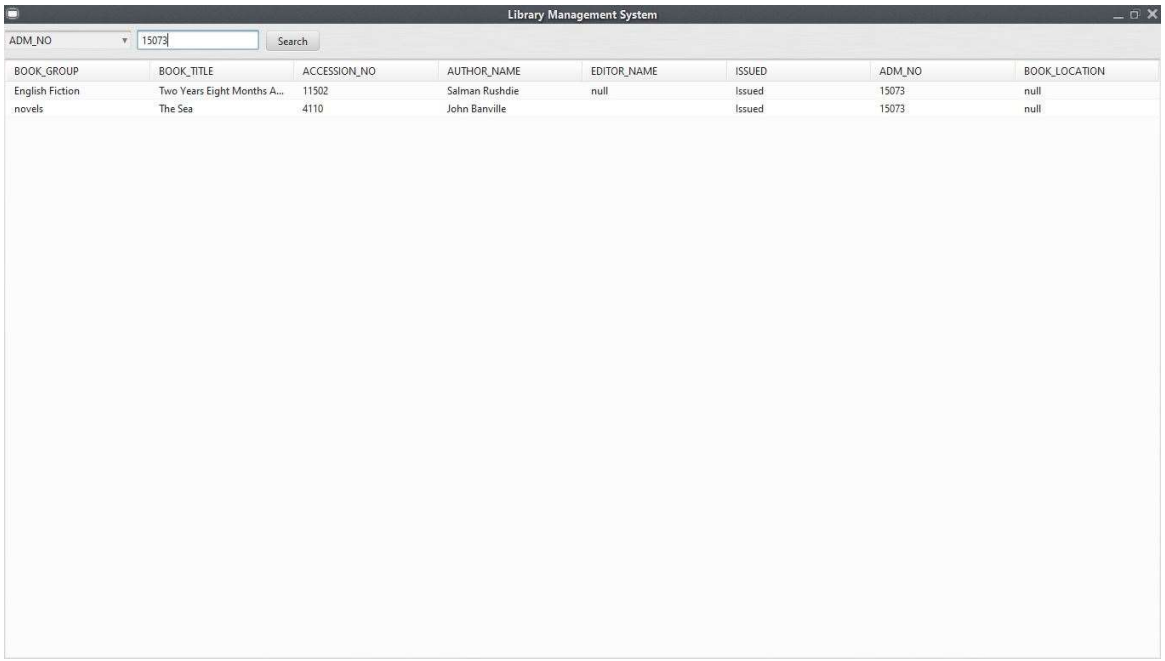


VS

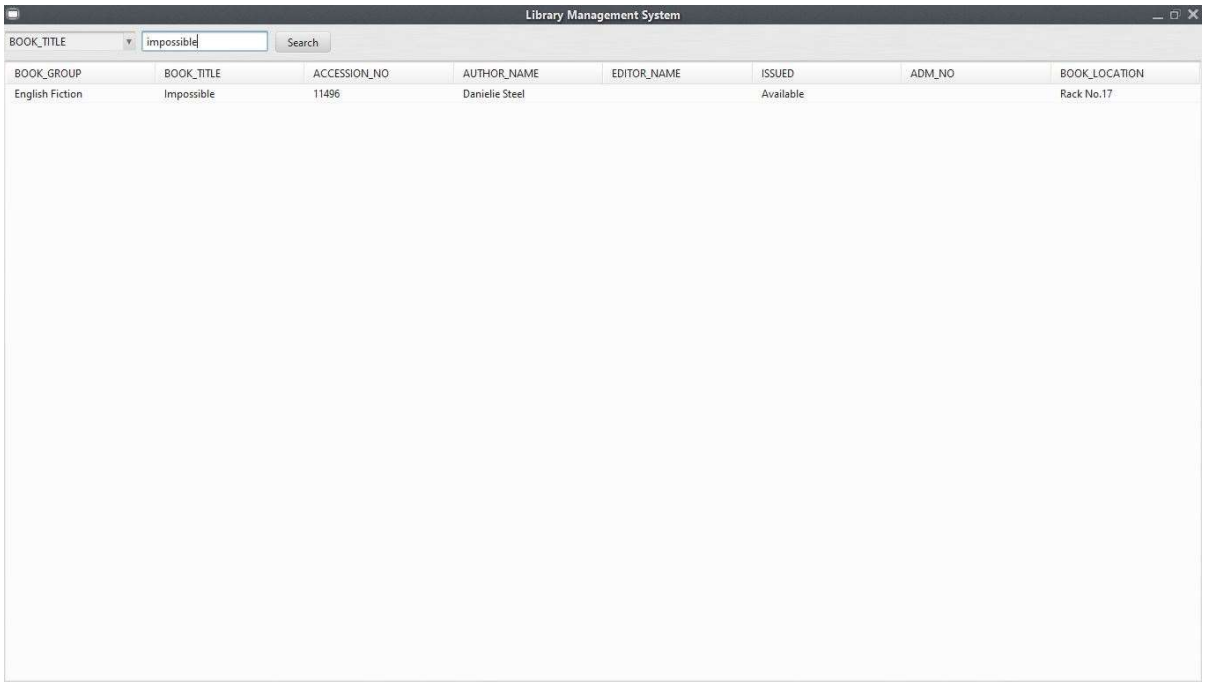


FRAME:

Search your own Status:



Search for Books:



CODE:

Search Button:

```
199     private void SearchButtonActionPerformed(ActionEvent evt) {
200         String search = (String)this.SearchBox.getSelectedItem();
201         String text = this.SearchTextField.getText();
202         DefaultTableModel model = (DefaultTableModel)this.SearchResult.getModel();
203         this.Search(search, text, model);
204     }
205
206     private void SearchTextFieldKeyPressed(KeyEvent evt) {
207         if (evt.getKeyCode() == 10) {
208             String search = (String)this.SearchBox.getSelectedItem();
209             String text = this.SearchTextField.getText();
210             DefaultTableModel model = (DefaultTableModel)this.SearchResult.getModel();
211             this.Search(search, text, model);
212         }
213     }
```

Searh Method:

```
152     int Search(String item, String text, DefaultTableModel model) {
153         while (model.getRowCount() > 0) {
154             model.setRowCount(0);
155         }
156         try {
157             Class.forName("com.mysql.jdbc.Driver");
158             Connection conn = (Connection)DriverManager.getConnection(db_url,user,pwd);
159             Statement stmt = (Statement)conn.createStatement();
160             String sql = "select * from books where " + item + " like'%" + text + "%';";
161             ResultSet rs = stmt.executeQuery(sql);
162             while (rs.next()) {
163                 String BOOK_GROUP = rs.getString("BOOK_GROUP");
164                 String BOOK_TITLE = rs.getString("BOOK_TITLE");
165                 String ACCESSION_NO = rs.getString("ACCESSION_NO");
166                 String AUTHOR_NAME = rs.getString("AUTHOR_NAME");
167                 String EDITOR_NAME = rs.getString("EDITOR_NAME");
168                 Boolean ISSUED = rs.getBoolean("ISSUED");
169                 String ADM_NO = rs.getString("ADM_NO");
170                 String BOOK_LOCATION = rs.getString("BOOK_LOCATION");
171                 String ISSUE = ISSUED == true ? "Issued" : "Available";
172                 model.addRow(new Object[]{BOOK_GROUP, BOOK_TITLE, ACCESSION_NO, AUTHOR_NAME, EDITOR_NAME, ISSUE, ADM_NO, BOOK_LOCATION});
173             }
174             rs.close();
175             conn.close();
176             stmt.close();
177         }
179         catch (Exception e) {
179             JOptionPane.showMessageDialog(null, e.getLocalizedMessage());
180             return -1;
181         }
182         return 0;
183     }
```

CONCLUSION

This project labelled as "Library Management System" was made especially for library of B.K Birla Centre for Education, Pune. It is used to keep records of all 10,000-12,000 books (currently) present in the library.

Before this application library used papers to keep the records of the books which is very difficult and old fashioned. By using this application the works becomes very easy and very efficient. If this software has to be used somewhere else it has to be modified. Because the system of library is different in every school. This application could be modified even more, but due to lack of time and resources it was not possible.

This application helped me immensely to learn about lot of new things in java and MySQL which is not mentioned in cbse NCERT. This project made me realised how big computer world is and made me appreciate it.

BIBLIOGRAPHY

- Stackoverflow.com
- Wikipedia.org
- Sumita Arora for class 12th cbse.
- Images.google.com