INTRODUCTION

1.1 BACKGROUND

The project titled Library Management System is Library management software for monitoring and controlling the transactions in a library .The Minor project "**Library Management System"** is developed in java swing, which mainly focuses on basic operations in a library like adding new member, new books, and updating new information, searching books and members and facility to borrow and return books.

"Library Management System" is a windows application written for Windows operating systems, designed to help users maintain and organize library. Our software is easy to use for both beginners and advanced users. It features a familiar and well thought-out, an attractive user interface, combined with strong searching Insertion and reporting capabilities. The report generation facility of library system helps to get a good idea of which are the books borrowed by the members, makes users possible to generate reports' hard copy.

1.2 PROJECT AIMS AND OBJECTIVES

The project aims and objectives that will be achieved after completion of this project are discussed in this subchapter. The aims and objectives are as follows:

- Online book issue
- Request column for librarian for providing new books
- A separate column for digital library
- Librarian login page where he can find books issued.
- A search column to search availability of books
- Improvement in control and performance.
- The system is developed to cope up with the current issues and problems of library.
- The system is bug free.
- Saves cost
- After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost.
- Saves time
- Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.

1.3 PROJECT PURPOSE, SCOPE & AVAILABILITY

The purpose of Online Library Management system includes :

- Insertion to Database Module— User friendly input screen
- Extracting from Database module —Attractive Output Screen
- Report Generation module— borrowed book list & Available book list
- Search Facility system — search for books and members
- Add / Delete / Update Facility system — Add / Delete / Update member and books and also admin

The scope of Online Library Management System includes:

- Provide the list of books the users can borrow.
- A status page for all users to view books borrowed by them, their individual due dates and their individual penalties if any.
- Provide method for adjusting account settings such as passwords.
- Providing interface to add or delete books to staffs.

REQUIREMENT ANALYSIS DOCUMENT

2.1 INTRODUCTION

Requirement Analysis Document is the starting point of the software requirement activity. Little importance was given to this phases in the early days of software development. The emphasis was first on coding and then shifted to design.
As systems grew more complex, it become evident that the goal of the entire system cannot be easily comprehended. Hence need for the requirements analysis phase arose. Now, for large software systems, requirements analysis is perhaps the most difficult activity and also the most error prone.
Some of the difficulty is due to the scope of this phase. The software project is imitated by the client needs. In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify the requirements by tacking to these people and understanding there needs. In situations where the software is to automated a currently manuals process, most of the needs can be understood by observing the current practice.
The SRS is a means of translating the ideas in the minds of the clients (the output) into formal document (the output of the requirements phase). Thus the output of the phase is a set of formally specified requirements, which hopefully are complete and consistent, while the input has none of these properties.

2.2 EXISTING SYSTEM

In our existing system all the transaction of books are done manually, So taking more time for a transaction like borrowing a book or returning a book and also for searching of members and books. Another major disadvantage is that to preparing the list of books borrowed and the available books in the library will take more time, currently it is doing as a one day process for verifying all records. So after conducting the feasibility study we decided to make the manual Library management system to be computerized.

2.3 PROPOSED SYSTEM

Proposed system is an automated Library Management System. Through our software user can add members, add books, search members, search books, update information, edit information, borrow and return books in quick time.

Our proposed system has the following advantages.

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility

2.4 FEASIBILITY STUDY

All the manual difficulties in managing the Library have been rectified by implementing computerization.
This study plan discusses about that time schedule for the project and it contains the various phases of the project.
The Various Phases of the Project:

S.NO	TASK	TASK DURATION
1	Requirement Specification	10 Day's
2	Requirement document specification	10 Day's
3	Design analysis	10 Day's
4	Design Documentation	10 Day's
5	Design Review	20 Day's
6	Coding	15 Day's

Total 90 Day's

2.4 FEASIBILITY ANALYSIS

Whatever we think need not be feasible .It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

Technical Feasibility:

We can strongly says that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

Economical Feasibility

Development of this application is highly economically feasible .The organization needed not spend much m one for the development of t he system already available. The only thing is to be done is making an environment for the development with an effective supervision. I f we are doing so , we can attain the maximum usability of the corresponding resources .Even after the development , the organization will not be in a condition to invest more in the organization .Therefore , the system is economically feasible.

2.4 FUNCTIONAL REQUIREMENTS

Admin

This feature used by the admin to login into system. They are required to enter user id and password before they are allowed to enter the system .The user id and password will be verified and if invalid id is there user is allowed to not enter the system.
Functional requirements:

- Admin id is provided when they register
- The system must only allow admin with valid id and password to enter the system
- The system performs authorization process which decides what admin level can access to.
- The admin must be able to logout after they finished using system.

Register New Librarian

This feature can be performed by all admin to register new librarian.
Functional requirements

- System must be able to verify information
- System must be able to delete information if information is wrong

Add New Book

This feature allows adding new books to the library Functional requirements

- System must be able to verify information
- System must be able to enter number of copies into table.
- System must be able to not allow two books having same book id.

Search Book

This feature is found in book maintenance part .WE can search book based on book id, book name, publication or by author name.
Functional requirements

- System must be able to search the database based on select search type
- System must be able to filter book based on keyword entered
- System must be able to show the filtered book in table view

Issue and Return Books

This feature allows issuing and returning books and also viewing reports of book issued.

Functional requirements

- System must be able to enter issue information in database.
- System must be able to update number of books.
- System must be able to search if book is available or not before issuing books
- System should be able to enter issue and return date information

2.4 FUNCTIONS OF SYSTEMS

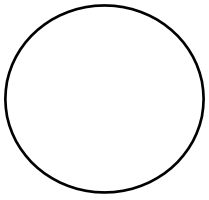
The functional requirement specification is a designed to describe that- What a software system should do?
It declares the intended function of a system and its components.
Based on this requirements, we will get determine the behavior that a software is expected to exhibit in case of certain input.

Data Modeling

In this section discussed about data flow diagram, Entity relationship diagram. These things are represented as diagrams with proper notation.
The data Modeling is one of the most improvement tools used by the system analyst DeMarco (1978) Nad Gand Sarson (1979) popularized the use if the data flow diagram as modeling tools through their structured system analysis methodologies.
A data flow diagram should be the first tool used by system analyst to model system components. These components are the system processes; the data used by this processes and external entities that interact with the system and the information flows in the system.
There are four kinds of system components

Process

Process show what system does. Each process has one or more data inputs and produce one or more data output, Circles in a data flow diagram represent process. Each process has unique name and number. This name and number appear inside the circle that represents the processes in a data flow diagram. It is represented as circle.



Data Stores

File or data store is depositary of data. They contain data that is retained in the system. Processes can enter the data into a data store or retrieve data from the data store. Each data store is represented by thin line in the data flow diagram and each data store has a unique name.

The data store is represented in form of a line

External Entities

External entities are outside the system but they either supply input data into the system or use the system output, they are entities which the designer has no control. Square or rectangle may represent external entities that supply data into a system or sometimes called sources. External entities that use the system data are sometimes called sinks.

Data Flow

Dataflow model the passage of data in the system and are represented lines joining system components. An arrow indicates the direction of the flow and the line labelled by the name of the data flow.

2.4 NON FUNCTIONAL REQUIREMENT

Performance Requirement

The project must the end user requirements. Accuracy and fast must be imposed in the project. It can be development as easy as possible for the sake of end user. It has to be developed with view of satisfying the future requirements and future enhancement. The tool has been finally implemented satisfying the needs specified by the company. As per the performance is concerned this system said is performing well.
This processing as well as tine taken to generate well reports where also even when large amount of data was used.
The system is designed in such a way that even when large amount of data used for processing there would less performance degradation.

Software Requirement

OPERATING SYSTEM	:	Windows XP Professional and
more IDE	:	Eclipse/Net Beans
FRONT END	:	Java Swing
BACK END	:	MYSQL

Pseudo Requirement

SYSTEM	:	Pentium III 700 MHz / More
HARD DISK	:	50 GB or more
MONITOR	:	15 VGA color monitor
MOUSE	:	Any mouse
RAM	:	512 MB or more
KEYBOARD	:	110 keys enhanced

SYSTEM DESIGN

3.1 DATABASE DESIGN

The general theme behind a database is to handle information as an integrated whole. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively.

After designing input and output, the analyst must concentrate on database design or how data should be organized around user requirements. The general objective is to make information access, easy quick, inexpensive and flexible for other users.

During database design the following objectives are concerned:-

- Controlled Redundancy
- Data independence
- Accurate and integrating
- More information at low cost
- Recovery from failure
- Privacy and security
- Performance
- Ease of learning and use

Mapping Objects/ Classes to tables

Table Name: Book

Field Name	Data Type	Description
id	int(10)	Primary
Call no	varchar(100)	
name	varchar(100)	
author	varchar(100)	
publisher	varchar(100)	
quantity	int(10)	
issued	int(10)	
Added date	timestamp	

Table Name: Librarian

Field Name	Data Type	Description
id	int(5)	Primary
name	varchar(100)	
password	varchar(100)	
email	varchar(100)	
address	varchar(100)	
city	varchar(100)	
contact	varchar(20)	

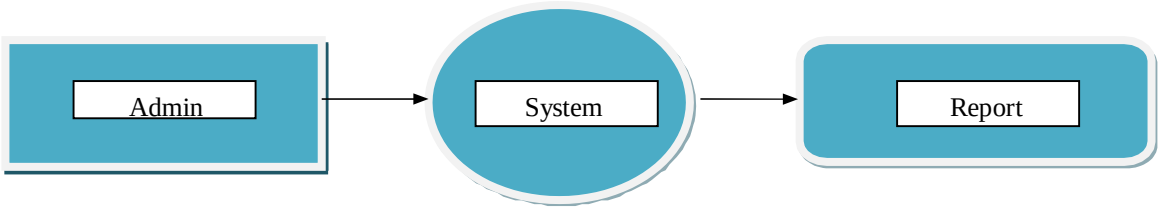
Table Name: Issue Book

Field Name	Data Type	Description
id	int(11)	Primary
bookcallno	varchar(50)	
studentid	int(11)	
studentname	varchar(50)	
studentcontact	varchar(20)	
issuedate	timestamp	

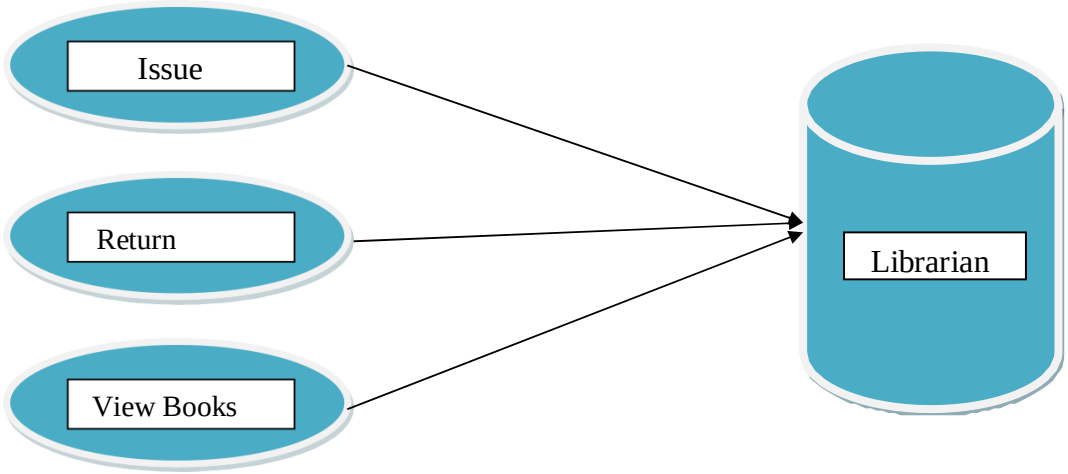
3.2 System Model

Data Flow Diagram

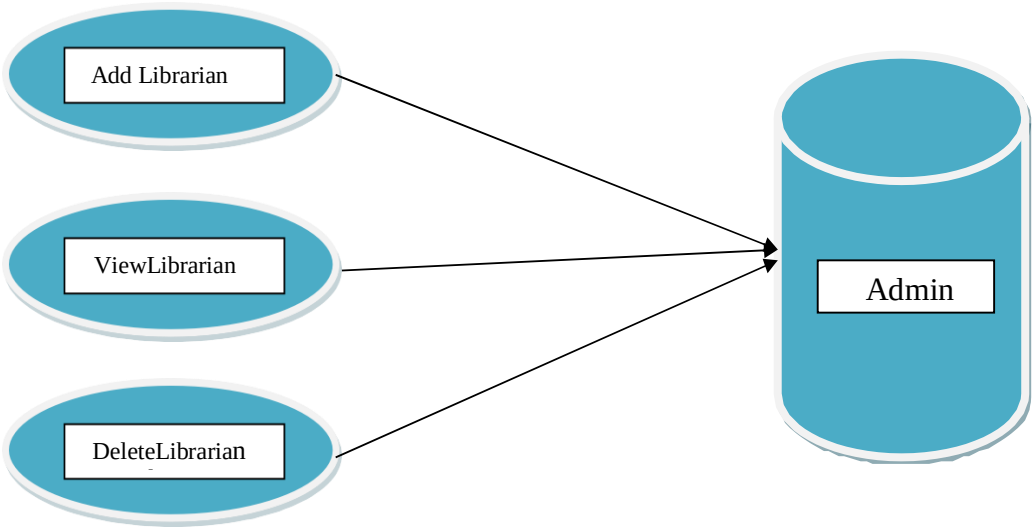
- Context Diagram



- Level 1



- Level 1



3.3 Input Design

Input design is the process of converting user-oriented input to a computer based format. Input design is a part of overall system design, which requires very careful attention Often the collection of input data is the most expensive part of the system. The main objectives of the input design are

- Produce cost effective method of input
- Achieve highest possible level of accuracy
- Ensure that the input is acceptable to and understood by the staff.

Input Data

The goal of designing input data is to make enter easy, logical and free from errors as possible. The entering data entry operators need to know the allocated space for each field; field sequence and which must match with that in the source document. The format in which the data fields are entered should be given in the input form .Here data entry is online; it makes use of processor that accepts commands and data from the operator through a key board. The input required is analyzed by the processor. It is then accepted or rejected.
Input stages include the following processes

- Data Recording
- Data Transcription
- Data Conversion
- Data Verification
- Data Control
- Data Transmission
- Data Correction

One of the aims of the system analyst must be to select data capture method and devices, which reduce the number of stages so as to reduce both the changes of errors and the cost.
Input types, can be characterized as.

- External
- Internal
- Operational
- Computerized
- Interactive

Input files can exist in document form before being input to the computer. Input design is rather complex since it involves procedures for capturing data as well as inputting it to the computer.

3.4 Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of these result for latter consultation .Computer output is the most important and direct source of information to the users. Designing computer output should proceed in an organized well throughout the manner. The right output must be available for the people who find the system easy to use. The outputs have been defined during the logical design stage. If not, they should defined at the beginning of the output designing terms of types of output connect, format, response etc,
Various types of outputs are

- External outputs
- Internal outputs
- Operational outputs
- Interactive outputs
- Turn around outputs

All screens are informative and interactive in such a way that the user can full fill his requirements through asking queries.

IMPLEMENTATION PLANNING AND DETAILS

4.1 Introduction to Front End

What is Java Swing?
Swing is a part of Java Foundation classes (JFC), the other parts of JFC are java2D and Abstract window toolkit (AWT). AWT, Swing & Java 2D are used for building graphical user interfaces (GUIs) in java. In this tutorial we will mainly discuss about Swing API which is used for building GUIs on the top of AWT and are much more light-weight compared to AWT.

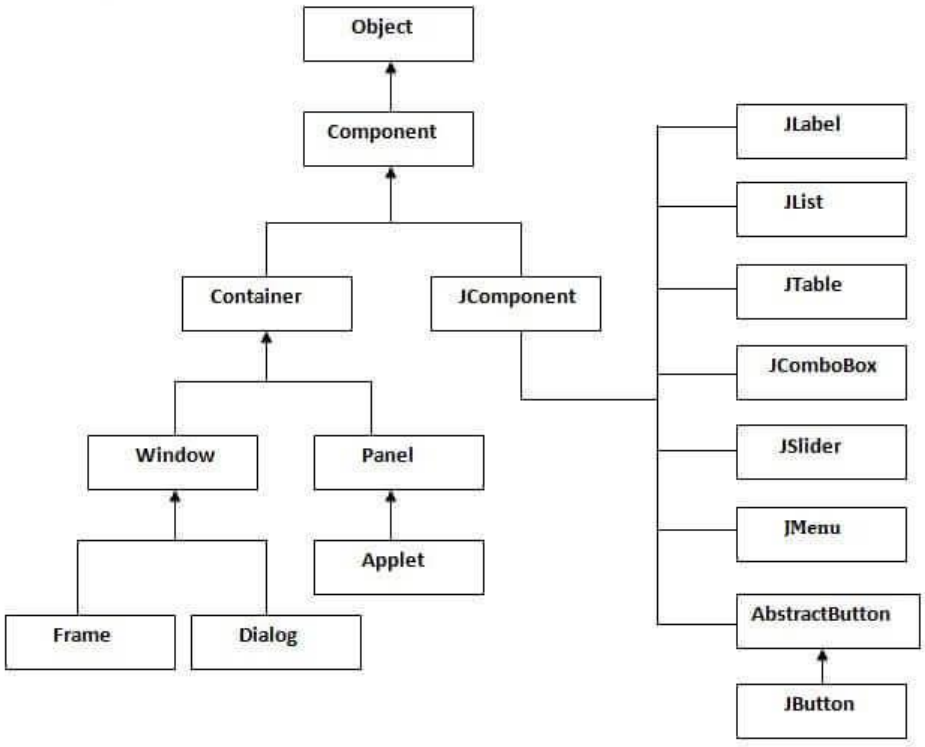
- JFrame – A frame is an instance of JFrame. Frame is a window that can have title, border, menu, buttons, text fields and several other components. A Swing application must have a frame to have the components added to it.

- JPanel – A panel is an instance of JPanel. A frame can have more than one panels and each panel can have several components. You can also call them parts of Frame. Panels are useful for grouping components and placing them to appropriate locations in a frame.
- JLabel – A label is an instance of JLabel class. A label is unselectable text and images. If you want to display a string or an image on a frame, you can do so by using labels. In the above example we wanted to display texts "User" & "Password" just before the text fields , we did this by creating and adding labels to the appropriate positions.
- JTextField – Used for capturing user inputs, these are the text boxes where user enters the data.
- JPasswordField – Similar to text fields but the entered data gets hidden and displayed as dots on GUI.
- JButton – A button is an instance of JButton class.

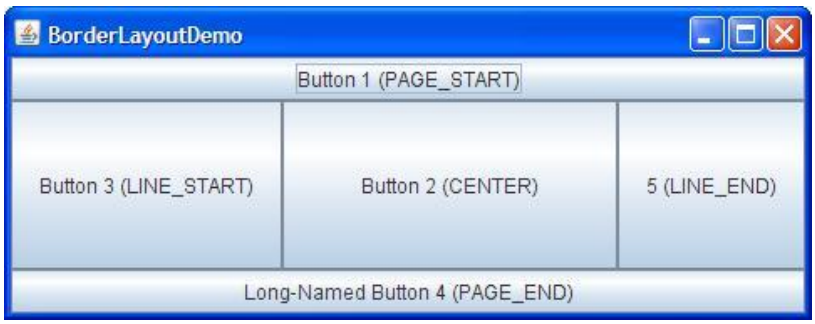
Swing Features:

- Light Weight − Swing components are independent of native Operating System's API as Swing API controls are rendered mostly using pure JAVA code instead of underlying operating system calls.
- Rich Controls − Swing provides a rich set of advanced controls like Tree, Tabbed Pane, slider, color picker, and table controls.
- Highly Customizable − Swing controls can be customized in a very easy way as visual appearance is independent of internal representation.
- Pluggable look-and-feel − SWING based GUI Application look and feel can be changed at run-time, based on available values.

The hierarchy of java swing API is given below.



A Border layout places components in up to five areas: top, bottom, left, right, and center. It is the default layout manager for every java JFrame



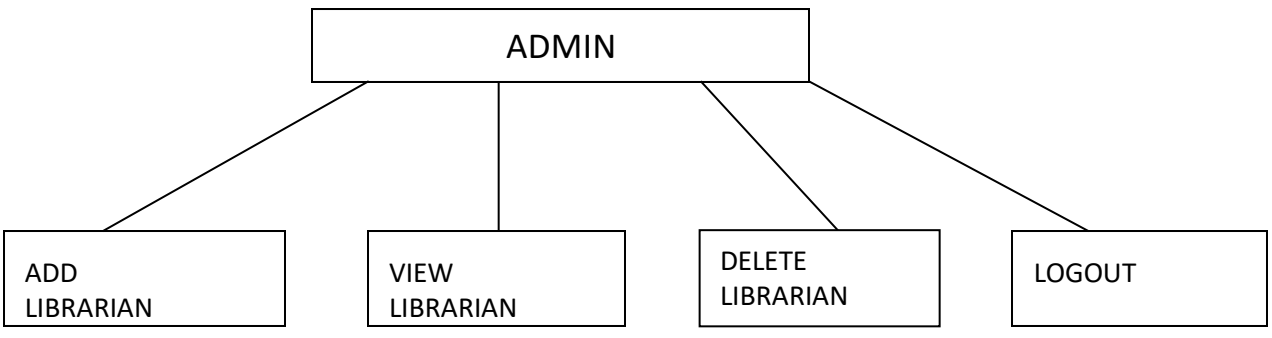
FlowLayout is the default layout manager for every JPanel. It simply lays out components in a single row one after the other.

GridBagLayout is the most sophisticated of all layouts. It aligns components by placing them within a grid of cells, allowing components to span more than one cell.

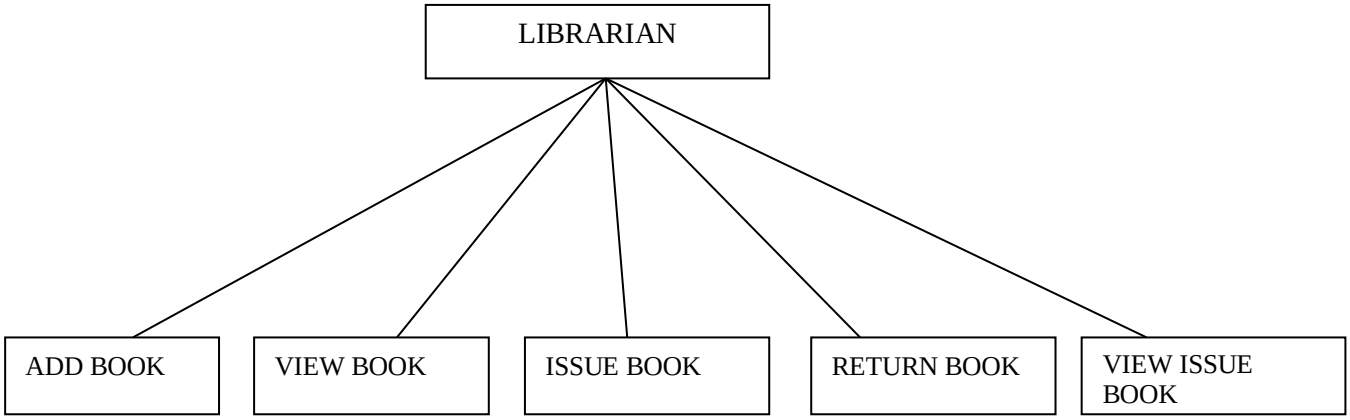
4.2 Program/Module Specification

The library management system is divided into following modules.

- **Admin**



- **Librarian**



4.3 Sample Coding

- **Admin Login**

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JPasswordField;

public class AdminLogin extends JFrame {
	static AdminLogin frame;
	private JPanel contentPane;
	private JTextField textField;
	private JPasswordField passwordField;

	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new AdminLogin();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
```

```
                        }
                    }
                };
        }

        /**
         * Create the frame.
         */
        public AdminLogin() {
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                setBounds(100, 100, 450, 300);
                contentPane = new JPanel();
                contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
                setContentPane(contentPane);

                JLabel lblAdminLoginForm = new JLabel("Admin Login Form");
                lblAdminLoginForm.setForeground(Color.GRAY);
                lblAdminLoginForm.setFont(new Font("Tahoma", Font.PLAIN, 18));

                JLabel lblEnterName = new JLabel("Enter Name:");

                JLabel lblEnterPassword = new JLabel("Enter Password:");

                textField = new JTextField();
                textField.setColumns(10);

                JButton btnLogin = new JButton("Login");
                btnLogin.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                        String name=textField.getText();
                        String password=String.valueOf(passwordField.getPassword());
                        if(name.equals("admin")&&password.equals("admin123")){
                                AdminSuccess.main(new String[]{});
                                frame.dispose();
                        }else{
                                JOptionPane.showMessageDialog(AdminLogin.this,
"Sorry, Username or Password Error","Login Error!",
JOptionPane.ERROR_MESSAGE);
                                textField.setText("");
                                passwordField.setText("");
                        }
                        }
                });

                passwordField = new JPasswordField();
                GroupLayout gl_contentPane = new GroupLayout(contentPane);
                gl_contentPane.setHorizontalGroup(
                        gl_contentPane.createParallelGroup(Alignment.TRAILING)
                                .addGroup(gl_contentPane.createSequentialGroup()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
```

```
        .addGroup(gl_contentPane.createSequentialGroup()
                                                .addGap(124)

        .addComponent(lblAdminLoginForm))

        .addGroup(gl_contentPane.createSequentialGroup()
                                                .addGap(19)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addComponent(lblEnterName)

        .addComponent(lblEnterPassword))
                                                .addGap(47)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)

        .addComponent(passwordField)
                                                        .addComponent(textField,
GroupLayout.DEFAULT_SIZE, 172, Short.MAX_VALUE))))
                                        .addContainerGap(107, Short.MAX_VALUE))
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addContainerGap(187, Short.MAX_VALUE)
                                        .addComponent(btnLogin,
GroupLayout.PREFERRED_SIZE, 86, GroupLayout.PREFERRED_SIZE)
                                        .addGap(151))
                );
                gl_contentPane.setVerticalGroup(
                        gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addComponent(lblAdminLoginForm)
                                        .addGap(26)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblEnterName)
                                                .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                        .addGap(28)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblEnterPassword)
                                                .addComponent(passwordField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                        .addGap(18)
                                        .addComponent(btnLogin,
GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
                                        .addContainerGap(80, Short.MAX_VALUE))
                );
                contentPane.setLayout(gl_contentPane);
        }
```

}

- **Books Form**

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.LayoutStyle.ComponentPlacement;

public class BooksForm extends JFrame {
	static BooksForm frame;
	private JPanel contentPane;
	private JTextField textField;
	private JTextField textField_1;
	private JTextField textField_2;
	private JTextField textField_3;
	private JTextField textField_4;


	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new BooksForm();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
```

```
	 * Create the frame.
	 */
	public BooksForm() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 450, 404);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);

		JLabel lblAddBooks = new JLabel("Add Books");
		lblAddBooks.setForeground(Color.GRAY);
		lblAddBooks.setFont(new Font("Tahoma", Font.PLAIN, 18));

		JLabel lblCallNo = new JLabel("Call No:");

		JLabel lblName = new JLabel("Name:");

		JLabel lblAuthor = new JLabel("Author:");

		JLabel lblPublisher = new JLabel("Publisher:");

		JLabel lblQuantity = new JLabel("Quantity:");

		textField = new JTextField();
		textField.setColumns(10);

		textField_1 = new JTextField();
		textField_1.setColumns(10);

		textField_2 = new JTextField();
		textField_2.setColumns(10);

		textField_3 = new JTextField();
		textField_3.setColumns(10);

		textField_4 = new JTextField();
		textField_4.setColumns(10);

		JButton btnAddBooks = new JButton("Add Books");
		btnAddBooks.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			String callno=textField.getText();
			String name=textField_1.getText();
			String author=textField_2.getText();
			String publisher=textField_3.getText();
			String squantity=textField_4.getText();
			int quantity=Integer.parseInt(squantity);
			int i=BookDao.save(callno, name, author, publisher, quantity);
			if(i>0){

	JOptionPane.showMessageDialog(BooksForm.this,"Books added
successfully!");
```

```
                        LibrarianSuccess.main(new String[]{});
                        frame.dispose();

                }else{

        JOptionPane.showMessageDialog(BooksForm.this,"Sorry, unable to save!");
                }
                }
        });

        JButton btnBack = new JButton("Back");
        GroupLayout gl_contentPane = new GroupLayout(contentPane);
        gl_contentPane.setHorizontalGroup(
                gl_contentPane.createParallelGroup(Alignment.TRAILING)
                        .addGroup(gl_contentPane.createSequentialGroup()

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()
                                                .addGap(150)
                                                .addComponent(lblAddBooks))

        .addGroup(gl_contentPane.createSequentialGroup()
                                                .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
                                                        .addComponent(lblName,
GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
                                                        .addComponent(lblCallNo)
                                                        .addComponent(lblAuthor,
GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)

        .addComponent(lblPublisher, GroupLayout.DEFAULT_SIZE, 67,
Short.MAX_VALUE)

        .addComponent(lblQuantity, GroupLayout.PREFERRED_SIZE, 55,
GroupLayout.PREFERRED_SIZE))
                                                .addGap(47)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addComponent(textField_4, GroupLayout.PREFERRED_SIZE, 167,
GroupLayout.PREFERRED_SIZE)

        .addComponent(textField_3, GroupLayout.PREFERRED_SIZE, 167,
GroupLayout.PREFERRED_SIZE)

        .addComponent(textField_2, GroupLayout.PREFERRED_SIZE, 167,
GroupLayout.PREFERRED_SIZE)

        .addComponent(textField_1, GroupLayout.PREFERRED_SIZE, 167,
GroupLayout.PREFERRED_SIZE)
```

```
                                                .addComponent(textField,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE))))
                                        .addContainerGap(125, Short.MAX_VALUE))
                                .addGroup(Alignment.LEADING,
gl_contentPane.createSequentialGroup()
                                        .addGap(161)
                                        .addComponent(btnAddBooks,
GroupLayout.PREFERRED_SIZE, 101, GroupLayout.PREFERRED_SIZE)
                                        .addContainerGap(162, Short.MAX_VALUE))
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addContainerGap(359, Short.MAX_VALUE)
                                        .addComponent(btnBack)
                                        .addContainerGap())
                );
                gl_contentPane.setVerticalGroup(
                        gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addComponent(lblAddBooks)
                                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblCallNo)
                                                .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblName)
                                                .addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblAuthor)
                                                .addComponent(textField_2,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblPublisher)
                                                .addComponent(textField_3,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblQuantity)
                                                .addComponent(textField_4,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
```

```
GroupLayout.PREFERRED_SIZE))
                                        .addGap(30)
                                        .addComponent(btnAddBooks,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(ComponentPlacement.RELATED)
                                        .addComponent(btnBack)
                                        .addContainerGap(53, Short.MAX_VALUE))
                );
                contentPane.setLayout(gl_contentPane);
        }

}
```

- **Issue Book**

```
import java.sql.*;
public class IssueBookDao {

public static boolean checkBook(String bookcallno){
        boolean status=false;
        try{
                Connection con=DB.getConnection();
                PreparedStatement ps=con.prepareStatement("select * from books
where callno=?");
                ps.setString(1,bookcallno);
           ResultSet rs=ps.executeQuery();
                status=rs.next();
                con.close();
        }catch(Exception e){System.out.println(e);}
        return status;
}

public static int save(String bookcallno,int studentid,String studentname,String
studentcontact){
        int status=0;
        try{
                Connection con=DB.getConnection();

                status=updatebook(bookcallno);//updating quantity and issue

                if(status>0){
                PreparedStatement ps=con.prepareStatement("insert into
issuebooks(bookcallno,studentid,studentname,studentcontact) values(?,?,?,?)");
                ps.setString(1,bookcallno);
                ps.setInt(2,studentid);
                ps.setString(3,studentname);
                ps.setString(4,studentcontact);
                status=ps.executeUpdate();
                }
```

```
            con.close();
        }catch(Exception e){System.out.println(e);}
        return status;
}
public static int updatebook(String bookcallno){
        int status=0;
        int quantity=0,issued=0;
        try{
            Connection con=DB.getConnection();

            PreparedStatement ps=con.prepareStatement("select quantity,issued
from books where callno=?");
            ps.setString(1,bookcallno);
            ResultSet rs=ps.executeQuery();
            if(rs.next()){
                quantity=rs.getInt("quantity");
                issued=rs.getInt("issued");
            }

            if(quantity>0){
            PreparedStatement ps2=con.prepareStatement("update books set
quantity=?,issued=? where callno=?");
            ps2.setInt(1,quantity-1);
            ps2.setInt(2,issued+1);
            ps2.setString(3,bookcallno);

            status=ps2.executeUpdate();
            }
            con.close();
        }catch(Exception e){System.out.println(e);}
        return status;
}
}
```

- **Librarian Form**

```
import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.JButton;
```

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class LibrarianForm extends JFrame {
	static LibrarianForm frame;
	private JPanel contentPane;
	private JTextField textField;
	private JTextField textField_1;
	private JTextField textField_2;
	private JTextField textField_3;
	private JTextField textField_4;
	private JPasswordField passwordField;

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new LibrarianForm();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */
	public LibrarianForm() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 450, 450);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);

		JLabel lblAddLibrarian = new JLabel("Add Librarian");
		lblAddLibrarian.setForeground(Color.DARK_GRAY);
		lblAddLibrarian.setFont(new Font("Tahoma", Font.PLAIN, 22));

		JLabel lblName = new JLabel("Name:");

		JLabel lblPassword = new JLabel("Password:");

		JLabel lblEmail = new JLabel("Email:");

		JLabel lblAddress = new JLabel("Address:");

		JLabel lblCity = new JLabel("City:");
```

```
		JLabel lblContactNo = new JLabel("Contact No:");

		textField = new JTextField();
		textField.setColumns(10);

		textField_1 = new JTextField();
		textField_1.setColumns(10);

		textField_2 = new JTextField();
		textField_2.setColumns(10);

		textField_3 = new JTextField();
		textField_3.setColumns(10);

		textField_4 = new JTextField();
		textField_4.setColumns(10);

		passwordField = new JPasswordField();

		JButton btnNewButton = new JButton("Add Librarian");
		btnNewButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			String name=textField.getText();
			String password=String.valueOf(passwordField.getPassword());
			String email=textField_1.getText();
			String address=textField_2.getText();
			String city=textField_3.getText();
			String contact=textField_4.getText();

			int i=LibrarianDao.save(name, password, email, address, city,
contact);
			if(i>0){

	JOptionPane.showMessageDialog(LibrarianForm.this,"Librarian added
successfully!");
				AdminSuccess.main(new String[]{});
				frame.dispose();

			}else{

	JOptionPane.showMessageDialog(LibrarianForm.this,"Sorry, unable to save!");
			}
			}
		});
		btnNewButton.setForeground(Color.DARK_GRAY);

		JButton btnBack = new JButton("Back");
		btnBack.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			AdminSuccess.main(new String[]{});
			frame.dispose();
```

```
		}
	});
	GroupLayout gl_contentPane = new GroupLayout(contentPane);
	gl_contentPane.setHorizontalGroup(
		gl_contentPane.createParallelGroup(Alignment.TRAILING)
			.addGroup(gl_contentPane.createSequentialGroup()
				.addGap(20)

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
					.addComponent(lblPassword,
GroupLayout.DEFAULT_SIZE, 62, Short.MAX_VALUE)
					.addComponent(lblName)
					.addComponent(lblEmail,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)
					.addComponent(lblAddress,
GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
					.addComponent(lblCity,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)
					.addComponent(lblContactNo,
GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
				.addGap(58)

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
					.addComponent(textField_4,
GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
					.addComponent(textField_3,
GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
					.addComponent(textField_2,
GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
					.addComponent(textField_1,
GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
					.addComponent(textField,
GroupLayout.DEFAULT_SIZE, 177, Short.MAX_VALUE)
					.addComponent(passwordField))
				.addContainerGap(107, Short.MAX_VALUE))
			.addGroup(gl_contentPane.createSequentialGroup()
				.addContainerGap(151, Short.MAX_VALUE)
				.addComponent(lblAddLibrarian)
				.addGap(144))
			.addGroup(gl_contentPane.createSequentialGroup()
				.addContainerGap(160, Short.MAX_VALUE)
				.addComponent(btnNewButton,
GroupLayout.PREFERRED_SIZE, 131, GroupLayout.PREFERRED_SIZE)
				.addGap(133))
			.addGroup(gl_contentPane.createSequentialGroup()
				.addContainerGap(200, Short.MAX_VALUE)
				.addComponent(btnBack)
				.addGap(169))
	);
	gl_contentPane.setVerticalGroup(
```

```
                gl_contentPane.createParallelGroup(Alignment.LEADING)
                    .addGroup(gl_contentPane.createSequentialGroup()
                        .addComponent(lblAddLibrarian)
                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()
                                .addComponent(lblName)
                                .addGap(18)
                                .addComponent(lblPassword))

        .addGroup(gl_contentPane.createSequentialGroup()
                                .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(ComponentPlacement.UNRELATED)
                                .addComponent(passwordField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)))
                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                            .addComponent(lblEmail)
                            .addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                            .addComponent(lblAddress)
                            .addComponent(textField_2,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                            .addComponent(lblCity)
                            .addComponent(textField_3,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                            .addComponent(lblContactNo)
                            .addComponent(textField_4,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                        .addGap(18)
                        .addComponent(btnNewButton,
GroupLayout.PREFERRED_SIZE, 36, GroupLayout.PREFERRED_SIZE)
```

```
        .addPreferredGap(ComponentPlacement.RELATED, 57, Short.MAX_VALUE)
                                        .addComponent(btnBack)
                                        .addGap(19))
                );
                contentPane.setLayout(gl_contentPane);
        }

}
```

- **Return Book**

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.LayoutStyle.ComponentPlacement;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class ReturnBook extends JFrame {
        static ReturnBook frame;
        private JPanel contentPane;
        private JTextField textField;
        private JTextField textField_1;

        /**
         * Launch the application.
         */
        public static void main(String[] args) {
                EventQueue.invokeLater(new Runnable() {
                        public void run() {
                                try {
                                        frame = new ReturnBook();
                                        frame.setVisible(true);
                                } catch (Exception e) {
                                        e.printStackTrace();
                                }
                        }
                });
        }
```

```
	/**
	 * Create the frame.
	 */

	public ReturnBook() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 516, 413);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		JLabel lblReturnBook = new JLabel("Return Book");
		lblReturnBook.setForeground(Color.GRAY);
		lblReturnBook.setFont(new Font("Tahoma", Font.PLAIN, 18));

		JLabel lblBookCallno = new JLabel("Book Callno:");
		JLabel lblStudentId = new JLabel("Student Id:");
		textField = new JTextField();
		textField.setColumns(10);
		textField_1 = new JTextField();
		textField_1.setColumns(10);

		JButton btnReturnBook = new JButton("Return Book");
		btnReturnBook.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				String bookcallno=textField.getText();
				int studentid=Integer.parseInt(textField_1.getText());
				int i=ReturnBookDao.delete(bookcallno, studentid);
				if(i>0){

	JOptionPane.showMessageDialog(ReturnBook.this,"Book returned
successfully!");
					LibrarianSuccess.main(new String[]{});
					frame.dispose();

				}else{

	JOptionPane.showMessageDialog(ReturnBook.this,"Sorry, unable to return
book!");
				}
			}
		});

		JLabel lblNewLabel = new JLabel("Note: Check the book properly!");
		lblNewLabel.setForeground(Color.RED);
		lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 13));

		JButton btnBack = new JButton("Back");
		btnBack.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				LibrarianSuccess.main(new String[]{});
				frame.dispose();
			}
```

```
        });
        GroupLayout gl_contentPane = new GroupLayout(contentPane);
        gl_contentPane.setHorizontalGroup(
                gl_contentPane.createParallelGroup(Alignment.TRAILING)
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addGap(36)

    .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING, false)
                                        .addComponent(lblStudentId,
Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                        .addComponent(lblBookCallno,
Alignment.LEADING, GroupLayout.DEFAULT_SIZE, 78, Short.MAX_VALUE))
                                .addGap(56)

    .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                        .addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, 181, GroupLayout.PREFERRED_SIZE)
                                        .addComponent(textField,
GroupLayout.PREFERRED_SIZE, 181, GroupLayout.PREFERRED_SIZE))
                                .addContainerGap(139, Short.MAX_VALUE))
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addContainerGap(210, Short.MAX_VALUE)
                                .addComponent(btnReturnBook,
GroupLayout.PREFERRED_SIZE, 104, GroupLayout.PREFERRED_SIZE)
                                .addGap(176))
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addContainerGap(205, Short.MAX_VALUE)
                                .addComponent(lblReturnBook)
                                .addGap(187))
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addGap(19)
                                .addComponent(lblNewLabel)
                                .addContainerGap(294, Short.MAX_VALUE))
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addContainerGap(355, Short.MAX_VALUE)
                                .addComponent(btnBack)
                                .addGap(46))
        );
        gl_contentPane.setVerticalGroup(
                gl_contentPane.createParallelGroup(Alignment.LEADING)
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addContainerGap()
                                .addComponent(lblReturnBook)
                                .addGap(32)

    .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                        .addComponent(lblBookCallno)
                                        .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                .addGap(34)
```

```
        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                        .addComponent(lblStudentId)
                                        .addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                    .addGap(29)
                                    .addComponent(btnReturnBook,
GroupLayout.PREFERRED_SIZE, 34, GroupLayout.PREFERRED_SIZE)
                                    .addGap(23)
                                    .addComponent(btnBack)

        .addPreferredGap(ComponentPlacement.RELATED, 28, Short.MAX_VALUE)
                                    .addComponent(lblNewLabel)
                                    .addGap(72))
                );
                contentPane.setLayout(gl_contentPane);
        }

}
```

SYSTEM TESTING

Testing:

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. System testing makes logical assumption that if all the parts of the system are correct, then the goal will be successfully achieved. A series of testing are done for the proposed system before thesystem is ready for the user acceptance testing.

The following are the types of Testing:

- Unit Testing
- Integration Testing
- Validation Testing

Unit Testing:

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Then the web form level testing is made. For example storage of data to the table inthe correct manner.
In the company as well as seeker registration form, the zero length username and password are given and checked. Also the duplicate username is given and checked. In the job and question entry, the button will send data to the server only if the client side validations are made. The dates are entered in wrong manner and checked. Wrong email-id and web site URL (Universal Resource Locator) is given and checked.

Integration Testing:

Testing is done for each module. After testing all the modules, the modules are integrated and testing of the final system is done with the test data, specially designed to show that the system will operate successfully in all its aspects conditions. Thus the system testing is a confirmation that all is correct and an opportunity to show the user that the system works.

Validation Testing:

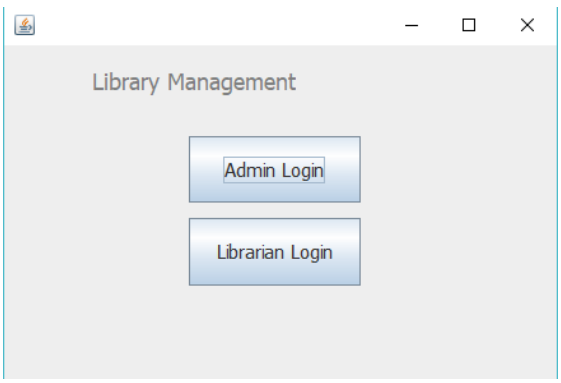
The final step involves Validation testing, which determines whether the software function as the user expected. The end-user rather than the system developer conduct this test most software developers as a process called "Alpha and Beta Testing" to uncover that only the end user seems able to find.
The compilation of the entire project is based on the full satisfaction of the end users. In the project, validation testing is made invarious forms. In question entry form, the correct answer only will be accepted in the answer box. The answers other than the four given choices will not be accepted.

MAINTENANCE:

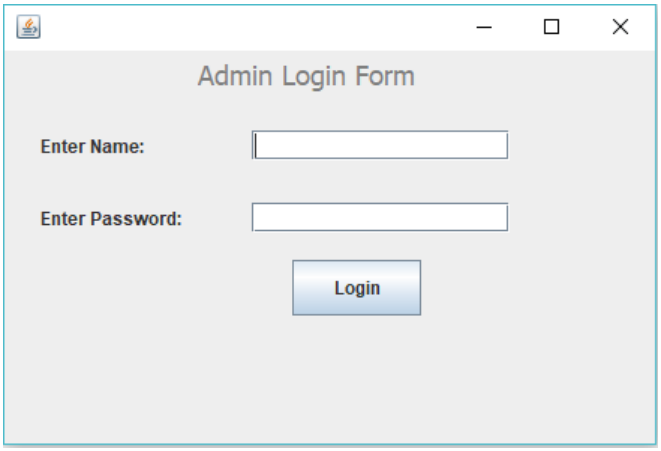
The objectives of this maintenance work are to make sure that the system gets into work all time without any bug. Provision must be for environmental changes which may affect the computer or software system. This is called the maintenance of the system. Nowadays there is the rapid change in the software world. Due to this rapid change, the system should be capable of adapting these changes. In our project the process can be added without affecting other parts of the system.
Maintenance plays a vital role. The system liable to accept any modification after its implementation. This system has been designed to favor all new changes. Doing this will not affect the system's performance or its accuracy.

SCREENSHOTS

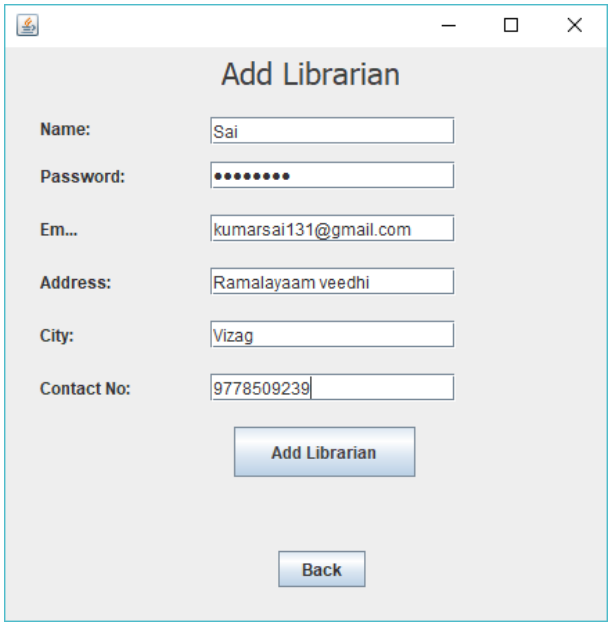
- **End User Page**



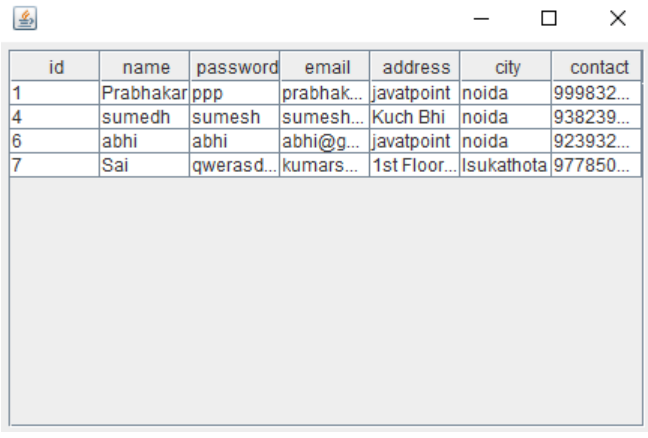
- **Admin Login Page**



- **Add Librarian**

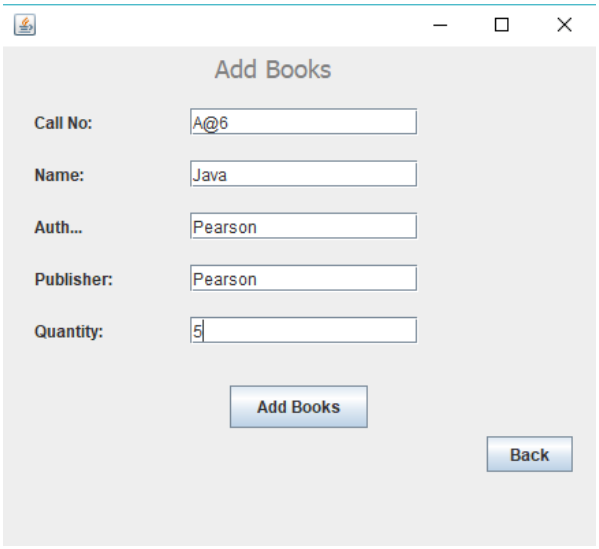


- **View Librarian**

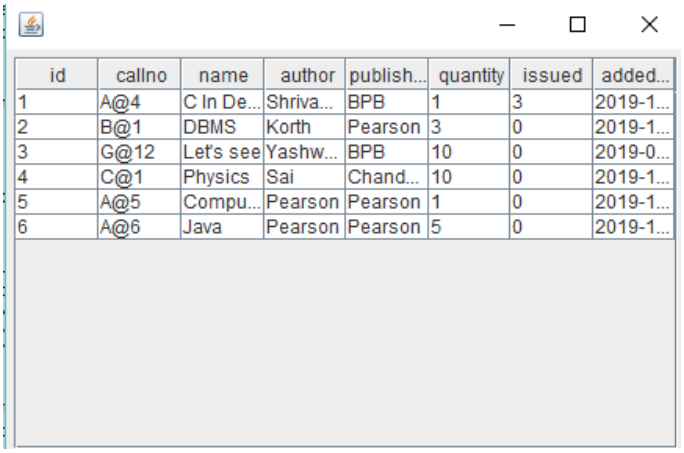


id	name	password	email	address	city	contact
1	Prabhakar	ppp	prabhak...	javatpoint	noida	999832...
4	sumedh	sumesh	sumesh...	Kuch Bhi	noida	938239...
6	abhi	abhi	abhi@g...	javatpoint	noida	923932...
7	Sai	qwerasd...	kumars...	1st Floor...	Isukathota	977850...

- **Add Book**

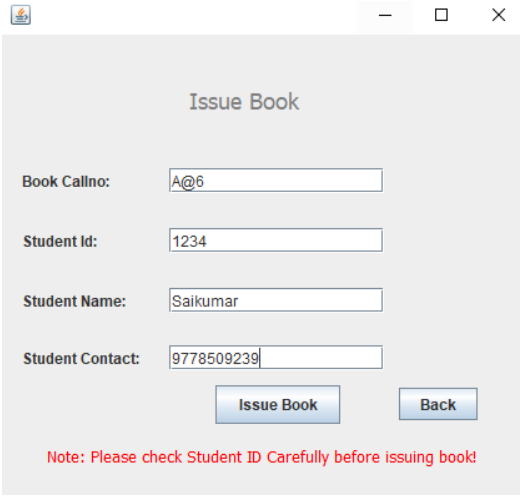


- **View Book**

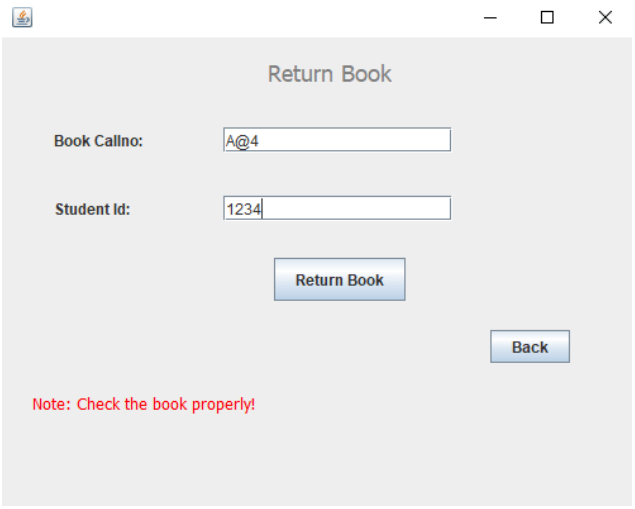


id	callno	name	author	publish...	quantity	issued	added...
1	A@4	C In De...	Shriva...	BPB	1	3	2019-1...
2	B@1	DBMS	Korth	Pearson	3	0	2019-1...
3	G@12	Let's see	Yashw...	BPB	10	0	2019-0...
4	C@1	Physics	Sai	Chand...	10	0	2019-1...
5	A@5	Compu...	Pearson	Pearson	1	0	2019-1...
6	A@6	Java	Pearson	Pearson	5	0	2019-1...

- **Issue Book**



- **Return Book**



CONCLUSION

Our project is only a humble venture to satisfy the needs in a library. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the organization.

The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

Last but not the least it is not the work that played the ways to success but ALMIGHTY.

BIBLIOGRAPHY

Websites:

https://www.mysql.com/

https://www.w3schools.com/

http://www.science.smith.edu/dftwiki/index.php/Tutorial:_Accessing_a_MySql_database_in_Java_(Eclipse)

https://www.javatpoint.com/