Department of Computer Science and Engineering

Institute of Engineering and Technology (IET)

JK Lakshmipat University, Jaipur

June 2022

Object Oriented Programming (CS1101)

Course Project Report

On

“Hostel Management System”

Team Members:

1. Chitraksh Gupta (2021Btech063)
2. Anay Sinhal (2021Btech012)
3. Babar Khilji (2021Btech031)
4. Vansh Jaiswal (2021Btech132)

Faculty Guide:

Mr. Devendra Bhavsar
Assistant Professor Institute of Engineering & Technology

Dr. Pranab Roy
Associate Professor
Institute of Engineering & Technology

ACKNOWLEDGEMENT

We have completed this project under the guidance and supervision of **Mr. Devendra Bhavsar (Assistant Professor, Computer Science Engineering Department, Institute of Engineering and Technology JK Lakshmipat University Jaipur) and Dr. Pranab Roy (Associate Professor, Computer Science Engineering Department, Institute of Engineering and Technology JK Lakshmipat University Jaipur)**. We are grateful for the esteemed scholarly guidance, assistance and knowledge, we have received from them towards fruitful and timely completion of this work. We also thank our colleagues who have helped in successful completion of the project.

Table of Content

Introduction

The project on "**Hostel Management System**" has been given to us as a part of the Course project in Object Oriented Programming (CS1101) in 2nd Semester. We have tried our best to make the complicated process of Hostel Management System as simple as possible using Structured and Modular technique and Menu oriented interface.

We have tried to design the software in such a way that user may not have any difficulty in using this software & further expansion is possible without much effort. The main purpose of our exercise is performing each student and warden's activity in computerized way rather than manually which is time consuming.

Administrator can maintain daily updates in the hostel records. Administrator must be an authorized user. He can further change the password. There is the facility for password also. Here we do have also created a page for warden also that can only be accessed by the warden itself. The portal can accept the whole sort of requests like changing room, leaving hostel and many more . we also have a feature that will show your pending will generate a receipt and will be automatically mailed to your email account.

"Hostel Management System" has been designed to computerize the following functions that are performed by the system:

- ✓ Sign Up for Students
- ✓ Sign In for Students
- ✓ Sign In for Warden
- ✓ Send Email on Sign Up
- ✓ View Information for Students
- ✓ View Information of Students for Warden
- ✓ Apply for Room for Students
- ✓ Allocate Room for Warden
- ✓ Room changing Request for Students
- ✓ Accepting Room changing Request for Warden
- ✓ Payment according to Room Type for Students

- ✓ Apply for leaving Hostel for Students
- ✓ Grant Permission for leaving Hostel for Warden
- ✓ Bill Generation
- ✓ Sending Bill by Email

Chapter 1 : Objective

Hostel Management system is the system that manages the student data, students' admission process and create receipt for the fees paid by the student who stay in the hostel and also help in maintaining visitor's messages.

This system is designed in favor of the hostel management which helps them to save the records of the students about their rooms. It helps them from the manual work from which it is very difficult to find the record of the students and the mess bills of the students, and the information of about those ones who had left the hostel.

We design this system on the request of the hostel management, through this they cannot require so efficient person to handle and calculate the things. This system automatically calculates all the bills and issued the notifications for those students who are against some rules. Through this it's possible to check the personal profile of all the current students within fraction of seconds.

Chapter 2 : Technologies Used

Front End:

- Java Swing (To create interactable design)

Back End:

- Java 12.0.1
- MySQL (To connect to Database)

Database Server:

- MySQL Database

IDE Used:

- Apache NetBeans 13

Chapter 3 : Data Tables

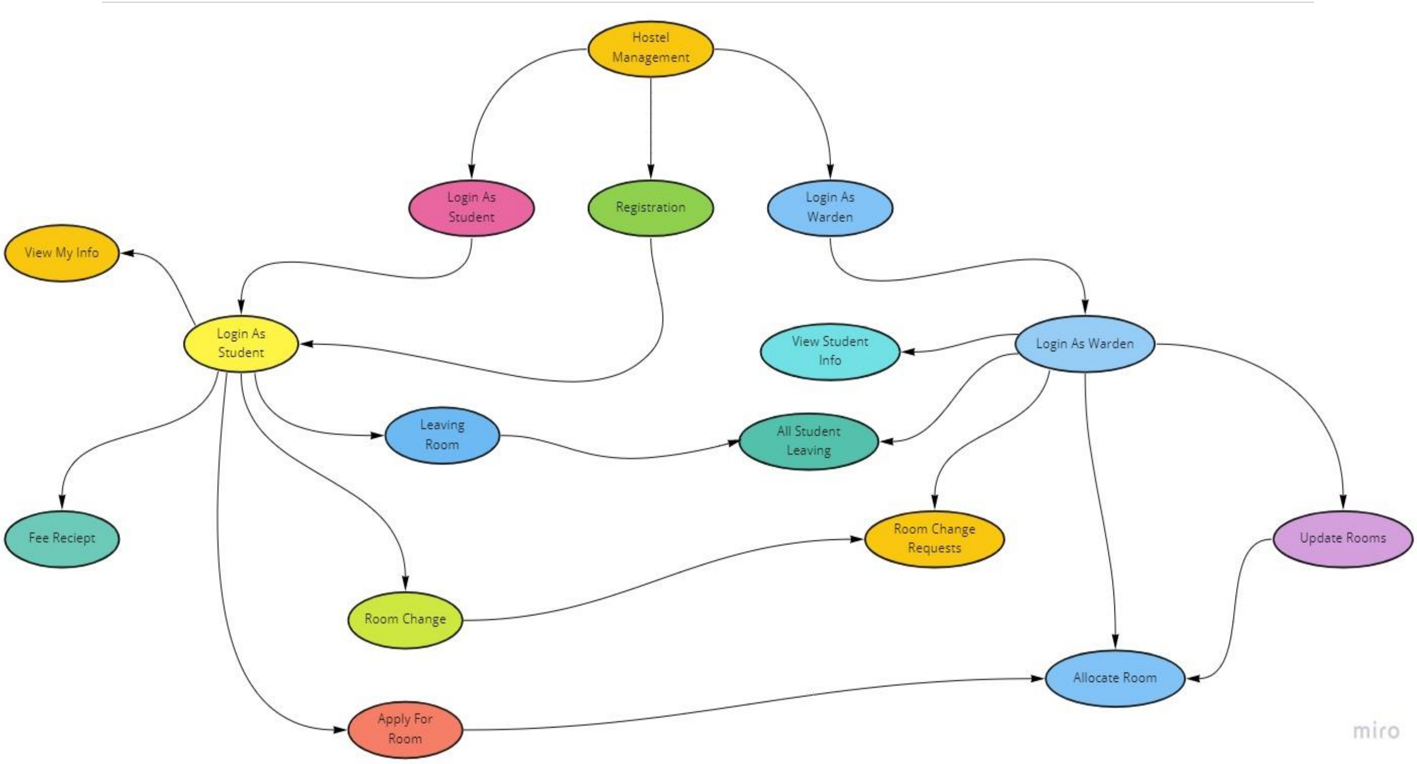
3.1 REGISTER (Stores User Information)

S NO	FIELD NAME	DATA TYPE
1	Name	VARCHAR 20
2	Phone_No	VARCHAR 10
3	Password	VARCHAR 20
4	Gender	VARCHAR 20
5	Roll_No	VARCHAR 20
6	Course	VARCHAR 20
7	Parents_Phone_Number	VARCHAR 10
8	Home_Address	VARCHAR 40
9	Aadhaar	VARCHAR 20
10	Date_Of_Birth	VARCHAR 10
11	Room_Type_Ac_NonAc	VARCHAR 20
12	Sharing	VARCHAR 20
13	Room_Allocated	Float
14	Fee_Details	Int
15	Total_Fee_Paid	Int
16	Room_Allocation_Request	Boolean
17	Apply_For_Leave	Boolean
18	Email_Id	VARCHAR 20
19	Leave_Time_to	VARCHAR 20
20	Leave_Time_from	VARCHAR 20

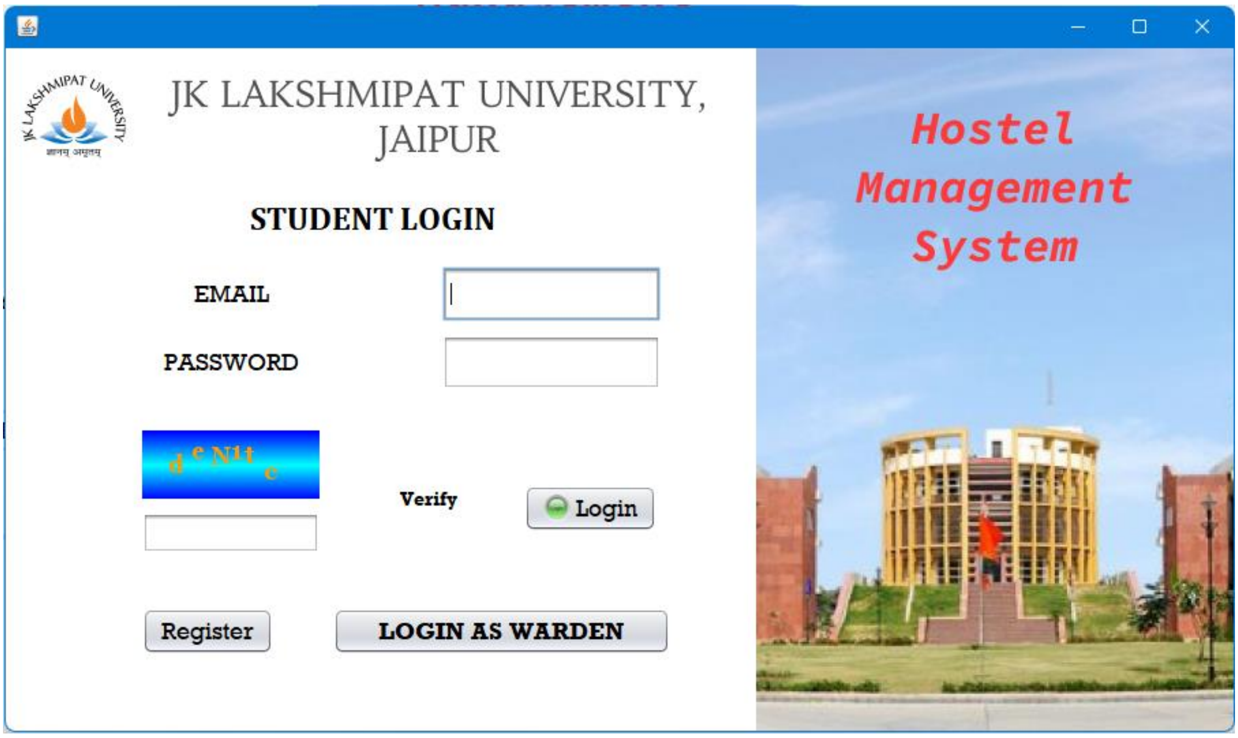
3.2 ROOM (Stores Room Information)

S.NO	FIELD NAME	DATA TYPE
1	Type_Ac_NonAc	VARCHAR 10
2	Sharing	VARCHAR 10
3	Status	VARCHAR 10
4	Email_id	VARCHAR 10
5	Room_No	VARCHAR 10

Chapter 4 : Data Flow Diagram



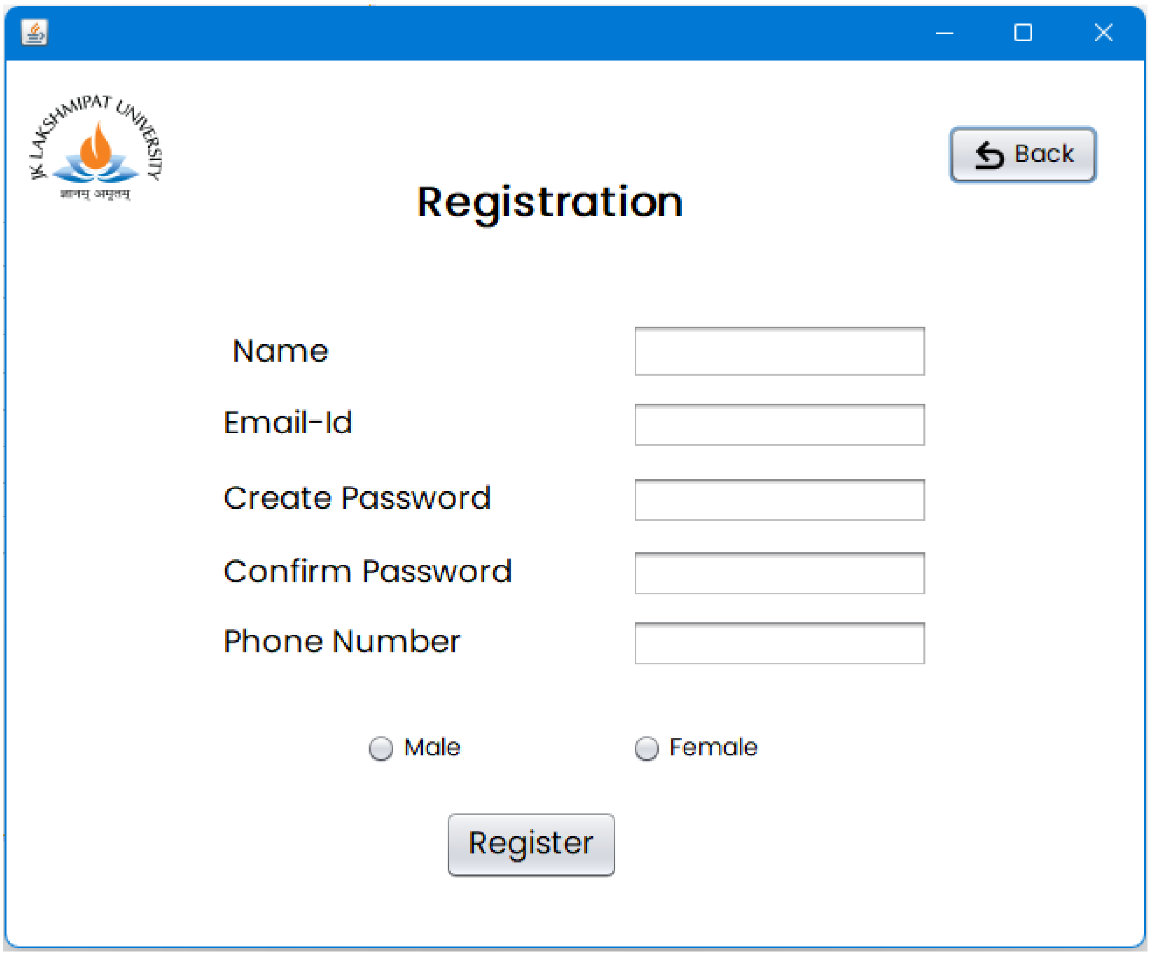
Chapter 5 : Screenshots of Forms



JK LAKSHMIPAT UNIVERSITY, JAIPUR

STUDENT LOGIN

Home Page / Login Page

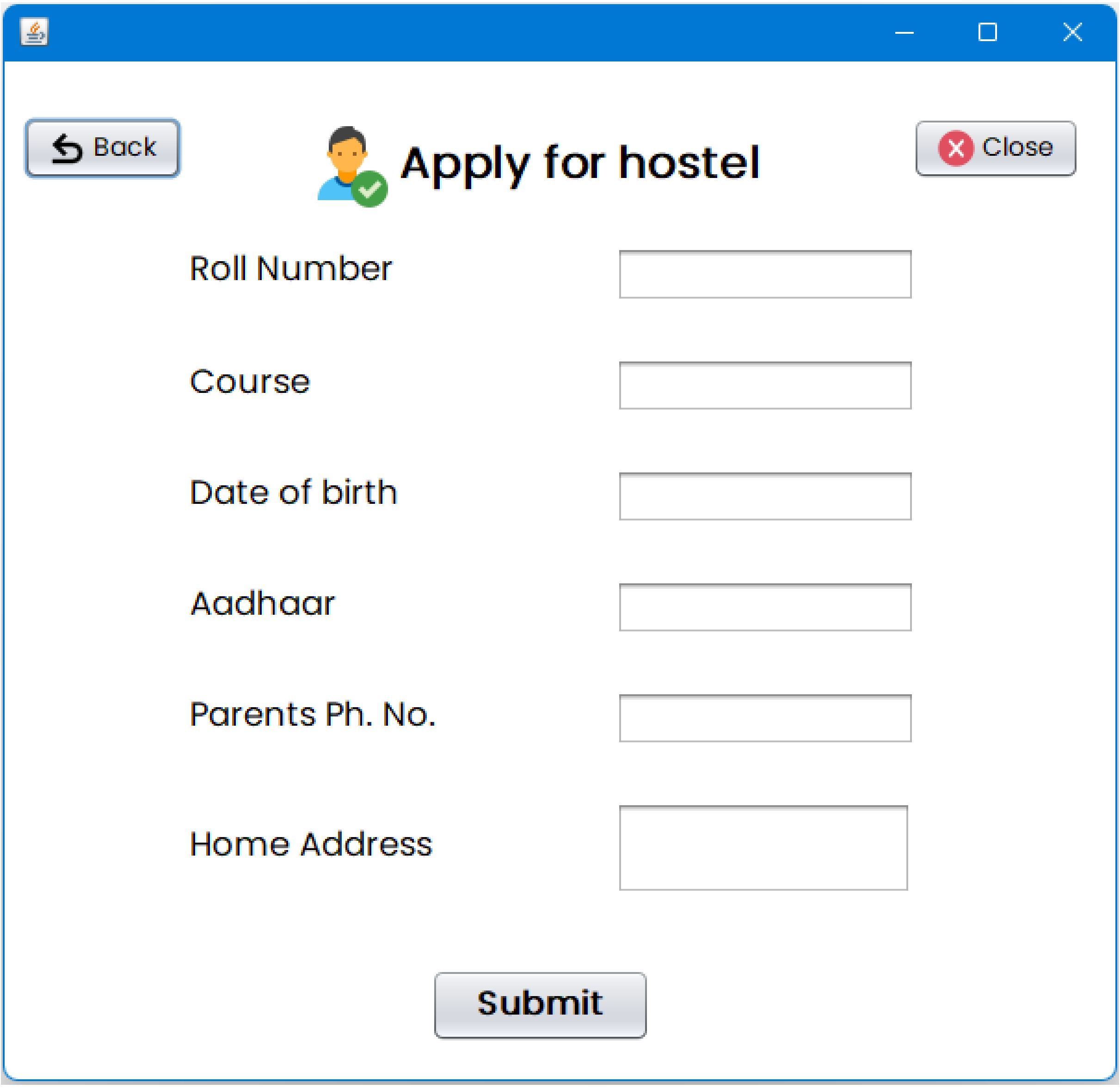


Registration

Name	
Email-Id	
Create Password	
Confirm Password	
Phone Number	

○ Male ○ Female

Registration Page for Students



Roll Number

Course

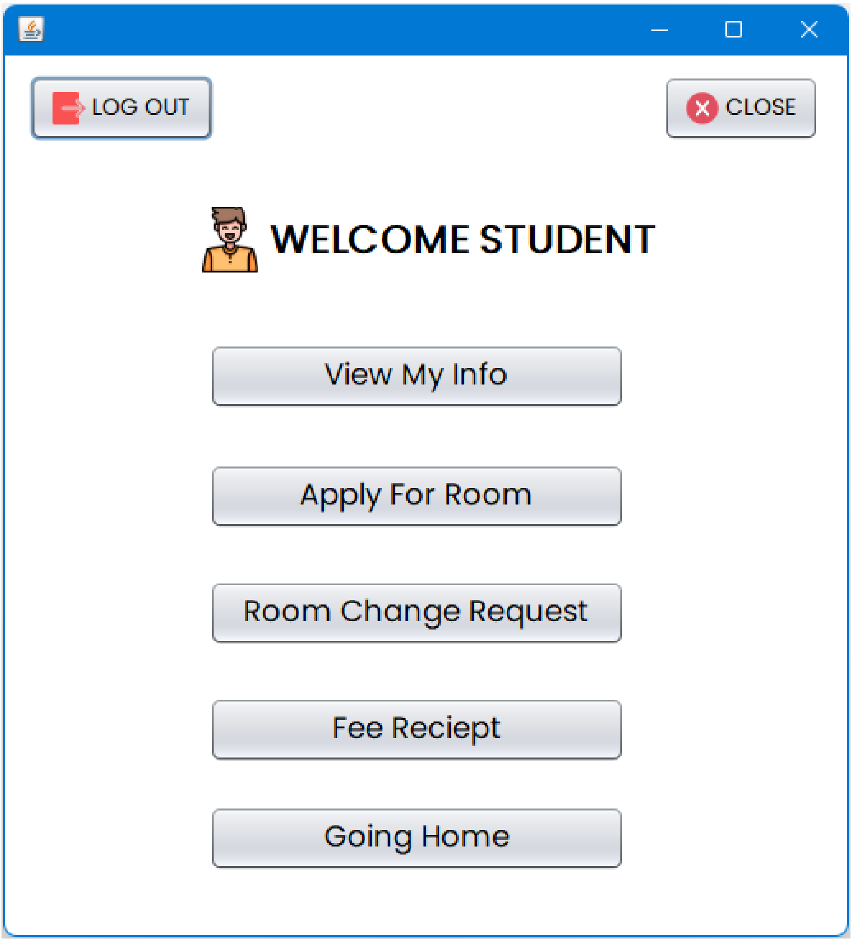
Date of birth

Aadhaar

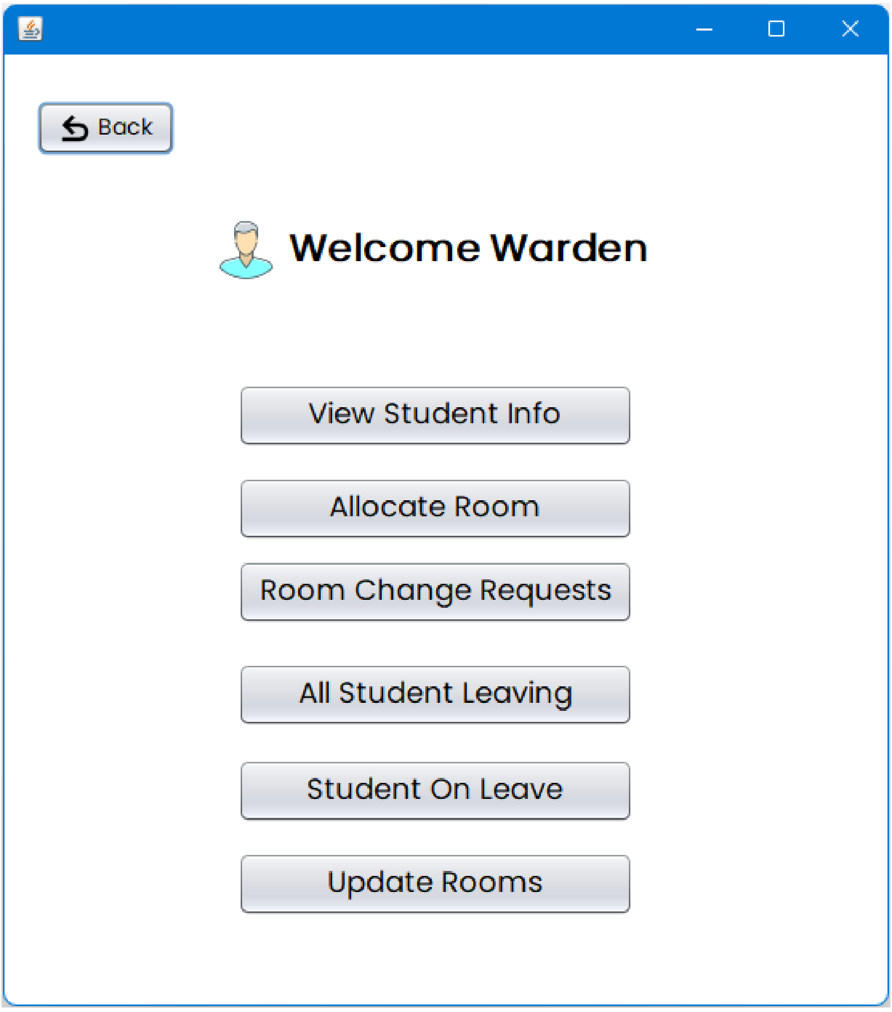
Parents Ph. No.

Home Address

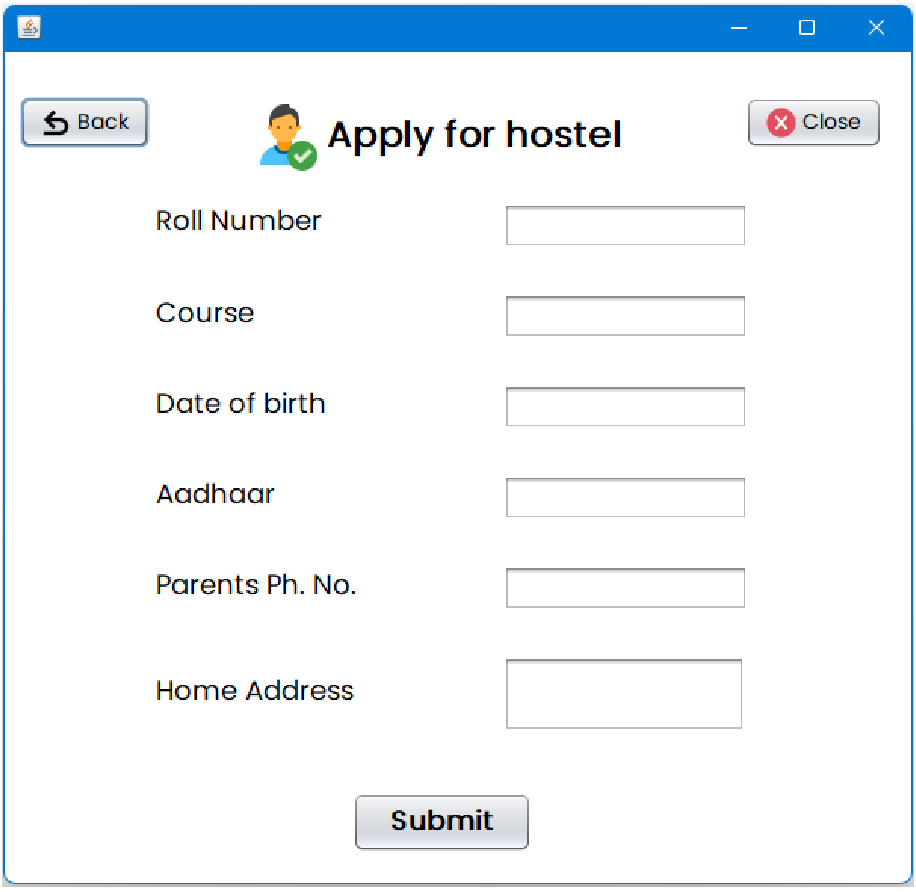
Apply For Hostel Page



Student Home Page



Warden Home Page



Apply for hostel

Roll Number

Course

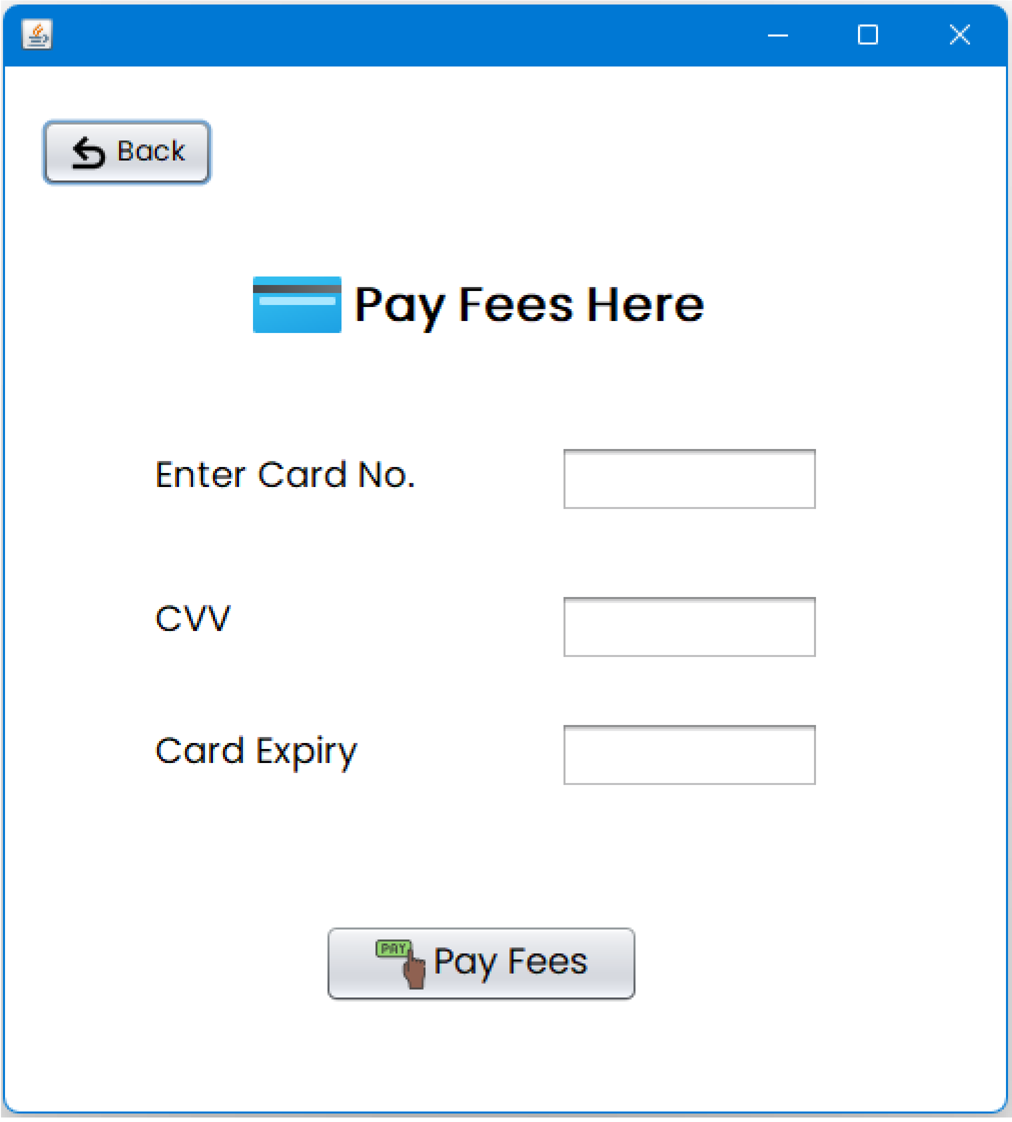
Date of birth

Aadhaar

Parents Ph. No.

Home Address

Apply For Room Page



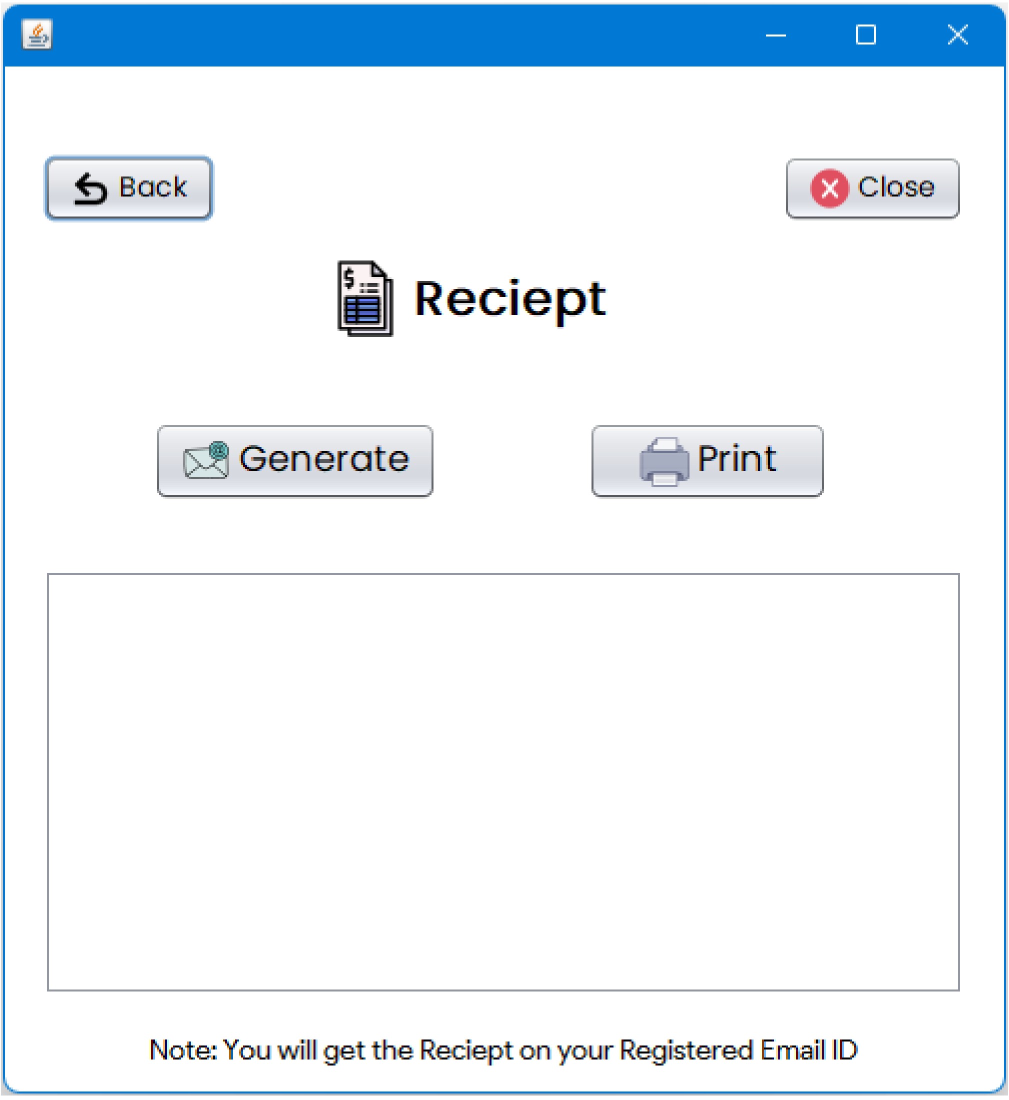
Pay Fees Here

Enter Card No.

CVV

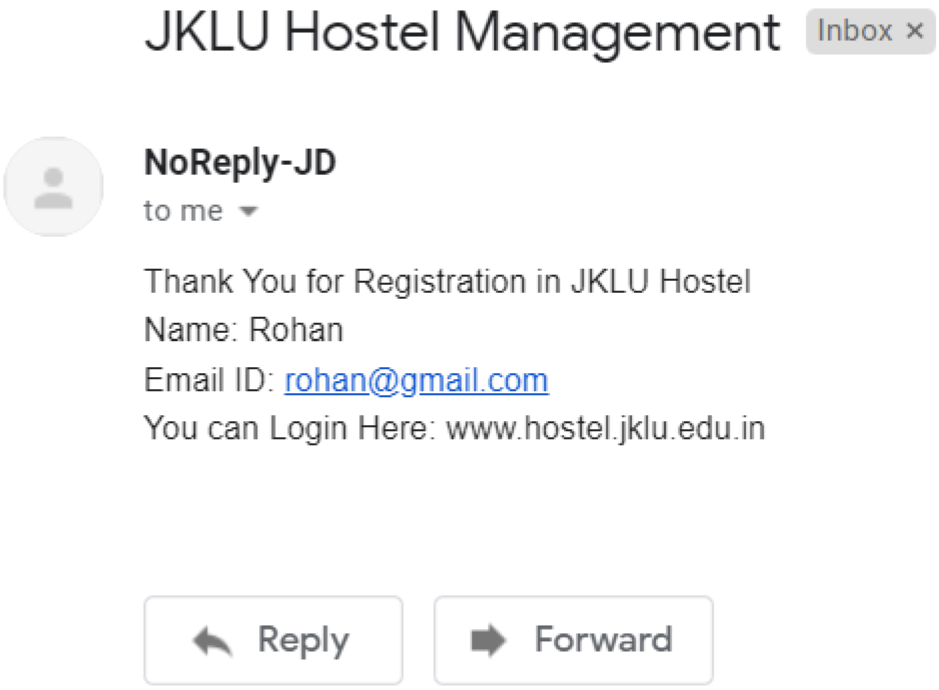
Card Expiry

Pay Fees Page



Note: You will get the Reciept on your Registered Email ID

Receipt Generation Page



JKLU Hostel Management

NoReply-JD

to me

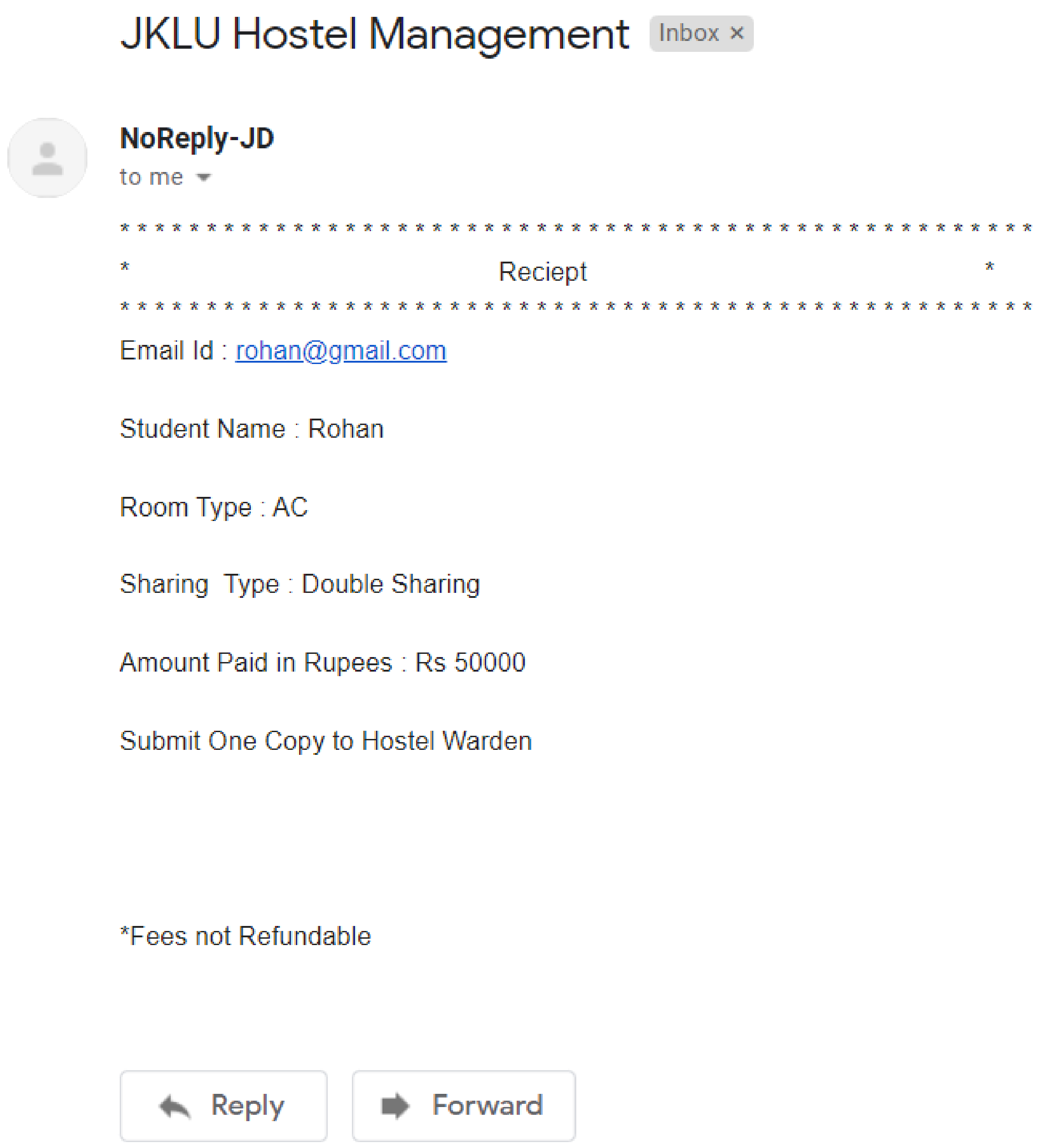
Thank You for Registration in JKLU Hostel

Name: Rohan

Email ID: rohan@gmail.com

You can Login Here: www.hostel.jklu.edu.in

Email for Registration



JKLU Hostel Management

NoReply-JD

* *

* Reciept *

* *

Email Id : rohan@gmail.com

Student Name : Rohan

Room Type : AC

Sharing Type : Double Sharing

Amount Paid in Rupees : Rs 50000

Submit One Copy to Hostel Warden

*Fees not Refundable

Email for Fee Receipt

Chapter 6 : Scope of Future Application

This project can be in the College Hostels after adding some more features and capabilities and solving some problems which might occur in real life scenario.

It can be catered to a Hostel/Organization’s personal needs and more features maybe added to it for services that the hostel may be providing to its students.

Utmost care and back-up procedures must be established to ensure 100% successful implementation of the computerized banking system. In case of system failure, the organization should be in a position to process the transaction with another organization or if the worst comes to the worst, it should be in a position to complete it manually.

Utmost care and back-up procedures must be established to ensure 100% successful implementation of the computerized hostel system. In case of system failure, the organization should be able to process the transaction with another organization or if the worst comes to the worst, it should be able to complete it manually.

With some changes and improved functionality this software can be used in real life as well.

Chapter 7 : Conclusion

A system has been developed to manage and automate the over-all processing of any large student hostel.

Hostel Management System is a user-friendly and customize software for providing support for hostel admin.

This project a very flexible software and can be upgraded according to the individual hostel needs.

Chapter 8 : Bibliography

- Official Apache NetBeans Website: https://netbeans.apache.org/
- Official NetBeans Documentation: https://netbeans.org/kb/index.html
- Official Java Documentation: https://docs.oracle.com/en/java/javase/13/
- Schildt, Herbert. 2019. Java: the complete reference.
- Prepared Statement: https://www.javatpoint.com/PreparedStatement-interface
- Java Mail API: https://www.javatpoint.com/example-of-sending-email-using-java-mail-api
- Stack Overflow: https://stackoverflow.com/

Chapter 9 : Appendix

Email Sending API

```
package com.mycompany.hostelmanage;
import java.util.Date;
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class EmailUtil {

        public static void sendEmail(Session session, String toEmail, String subject, String body){
                try
            {
              MimeMessage msg = new MimeMessage(session);
              //set message headers
              msg.addHeader("Content-type", "text/HTML; charset=UTF-8");
              msg.addHeader("format", "flowed");
              msg.addHeader("Content-Transfer-Encoding", "8bit");

              msg.setFrom(new InternetAddress("no_reply@example.com", "NoReply-JD"));

              msg.setReplyTo(InternetAddress.parse("no_reply@example.com", false));

              msg.setSubject(subject, "UTF-8");

              msg.setText(body, "UTF-8");

              msg.setSentDate(new Date());

              msg.setRecipients(Message.RecipientType.TO, InternetAddress.parse(toEmail, false));
              System.out.println("Message is ready");
          Transport.send(msg);

              System.out.println("EMail Sent Successfully!!");
            }
            catch (Exception e) {
              e.printStackTrace();
            }
        }
}
```

Students Leaving from Hostel

```
package com.mycompany.hostelmanage;
```

```
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

public class leaving extends javax.swing.JFrame {

    public leaving() {
        initComponents();
        DefaultTableModel model = (DefaultTableModel) table1.getModel();
        table1.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        try
        {
            Connection con =db.mycon();
            Statement st=con.createStatement();
            ResultSet rs=st.executeQuery("select * from register where Leav_not= 'ON LEAVE' ");
            while(rs.next())
            {
               model.addRow(new Object[]{
rs.getString(1),rs.getString(7),rs.getString(9),rs.getString(19),rs.getString(20),rs.getString(21)});
            }
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
```

Room Allocation

```
package com.mycompany.hostelmanage;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author chitr
 */
public class roomAllocation extends javax.swing.JFrame {
    public roomAllocation() {
        initComponents();
        DefaultTableModel model = (DefaultTableModel) table1.getModel();
        table1.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        model.setRowCount(0);
        try {
            Connection con = db3.mycon3();
            Statement st = con.createStatement();
            ResultSet rs = st.executeQuery("select * from register where Room_allocated='not' and
Fee_details='Paid'");
            while (rs.next()) {
                model.addRow(new Object[] { rs.getString(1), rs.getString(3), rs.getString(13),
rs.getString(14) });
            }

        } catch (Exception e) {
            System.out.println(e);
        }
  }
```