RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi, Accredited By NAAC, Bengaluru And NBA, New Delhi

Go, change the world

DEPARTMENT OF

MASTER OF COMPUTER APPLICATIONS

Java Programming Assignment
18MCA25

Submitted in partial fulfillment of the requirements for the award of degree of

MASTER OF COMPUTER APPLICATONS

By STUDENT NAME

CHAITRA BV

(USN: 1RV19MCA21)

Under the Guidance of

Name of the Guide

Mrs. Shaila H Koppad

2019-2020

RV COLLEGE OF ENGINEERING®,

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

Bengaluru– 560059

CERTIFICATE

Certified that the Assignment titled **"Hospital Management System"** carried out by CHAITRA BV**,** **USN:1RV19MCA21**, a Bonafede student of **RV College of Engineering, Bengaluru** submitted in partial fulfilment for the award of **Master of Computer Applications** of **RV College of Engineering, Bengaluru affiliated to Visvesvaraya Technological University, Belagavi** during the year **2019-20**.It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the partial academic requirement in respect of the course Java Programming 18MCA25.

Faculty- In charge:
Mrs. Shaila H Koppad
Department of MCA,RVCE
Bengaluru –59

ACKNOWLEDGEMENT

Before you proceed any further, I would request you to read this “small” acknowledgement. Which I think, it is not enough to express my gratitude towards all the people who have helped me to make this project. “Task Successful” makes everyone happy. But the happiness will be gold without glitter if we didn’t state the person who have supported us to make it a success. I proudly express my respectful thanks to my esteemed education institution RV COLLEGE OF ENGINEERING which has blessed me to continue my further education and has provided me a bright future.We are grateful to all the staff members of computer department for their immense co operation I am also great full to our all teaching staff and non-teaching staff for their valuable guidance novel ideas and excellent support during the course of my study and project.Last but not least I am very thankful to parents and friends who directly or indirectly helped us to complete our project successfully.

CHAITRA BV

1RV19MCA21

TABLE OF CONTENTS

1. ABSTRACT

Hospital Management System provides the benefits of streamlined operation enhanced administration & control, superior patient care, strict cost control and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support.

The project 'Hospital Management System' is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using

MY SQL software which is one of the best and the easiest software to keep our information.

This project uses JAVA as the front-end software which is an Object Oriented Programming and has connectivity with MY SQL.

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Medicine details, Doctor, Wards, , Admin, Store, Patient appointment, bill payment, record modification, discharge details etc.

2. INTRODUCTION

Human Body is a very complex and sophisticated structure and comprises of millions of functions. All these complicated functions have been understood by man him, part-by-part their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of Medical institutions i.e. Hospitals, HOSPITALs etc. research and development institutions and medical colleges. Thus the Health sector aims at providing the best medical facilities to the common man

3. Problem Statement

The project entitled as "HOSPITAL MANAGEMENT SYSTEM" is to computerize the Front Office Management of Hospital and to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient's information, diagnosis details, etc. Traditionally, it was done manually.The main function of my system is to register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully.

The purpose of this project is to plan a solution for problem specified by the requirements. It aims to identify the modules that should be in the system, the specification of those modules and how the interact with each other to produce the results. The goal of the project is to produce a model that can be used later to build that system. The produced model is called design of the system

The Hospital Management System can be used by entering respective username and password. It is accessible either by an administrator or receptionist. Only the respective person can add data in the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected and data processing is very fast, accurate and relevant.

My project Hospital Management system is able to register patients for hospital, stores their disease details into the database. Any of the staff member, doctor & admin is able to add, view, edit, update or delete data. It also contains doctor's information or doctor's profile and any one log in can view doctor's profiles. On the other hand only admin user is able to add, edit, update or delete doctor's profiles. My software is also facilitated to give a unique id for every patient and stores the details of every patient and the staff automatically.

This Hospital Management System also keeps the profile and related details to the database and the permissions to add, view, edit or delete is given only to doctors and admin of the system. It also includes a power search facility by which user can search all information about any doctor, staff and the details of a patient using any keyword.This Hospital Management System contains all basic needs for a hospital including Doctor's Information.Doctor's Profiles, Employee Profiles & Basic Information, Patient's Basic Information, Patient's Disease Information, Inventory Inflow/Outflow, Cost of Services.

These are the list of reports which are generated through this database in this system:Current Inventory, No. of Patient/ Disease in a year, Printable Billing, Hospital Annual, Turnover, Doctor's Rating.This Hospital Management System basically contains three level (role) of users

Admin

Doctor

patient

Admin user is super user this user have all permissions in my system, he is able to do every changes like creating new patient details, users, doctor's profile, staff's profile and many more… This user is also able to edit, update and delete data that stored in database.Doctor user is some sort of middle level user having more power than staff but less power than admin. He is able to do everything other than accepting doctor and user data.Staff level user is lower most user level with minimum permissions. This user is just able to add, edit, delete and search for patient's details.

4. Entity and Attributes

1. Doctor

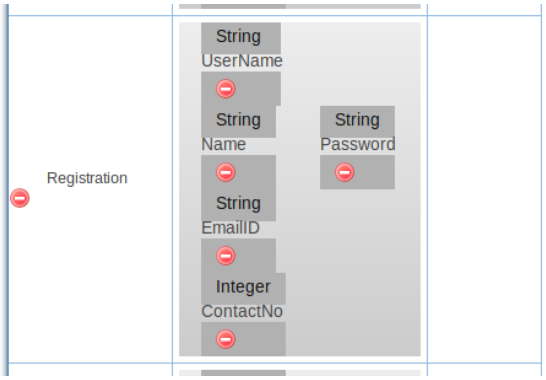
Table #6: Classes identified for solution

Class	Attributes	Operations
Doctor	Integer DoctorID String DoctorName String FatherName String EmailID String ContactNo String Qualification String BloodGroup Double DateOfJoining String Address	void Treates

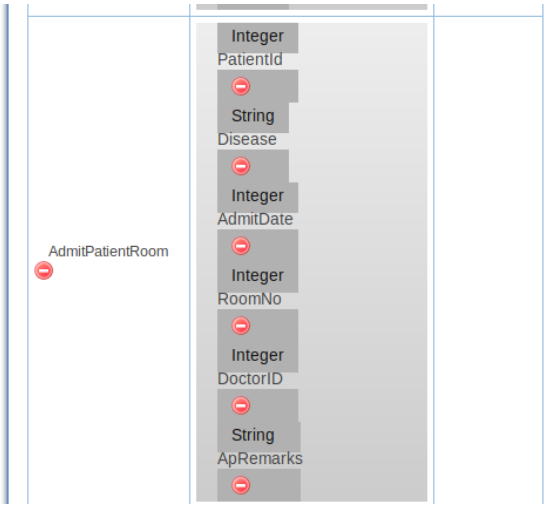
2. Patient

PatientRegistration	Integer PatientID String PatientNmae String FatherName String EmailID Integer ContactNo Integer Age String Remarks String Gender String BloodGroup String Adress	
	Integer PatientId String Disease	

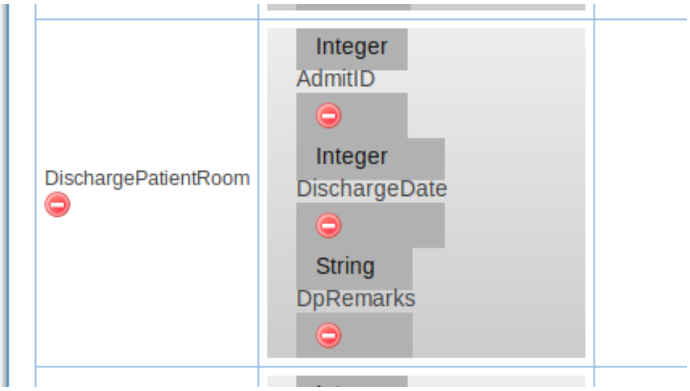
3. Registration



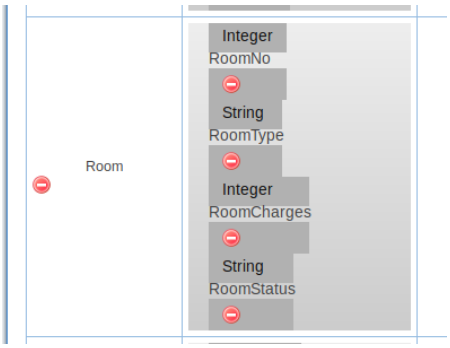
4. AdmitPatientRoom



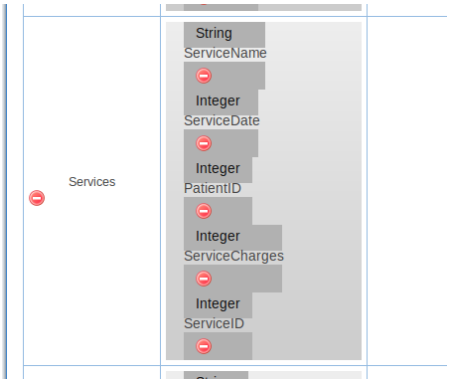
5. DischargePatientRoom



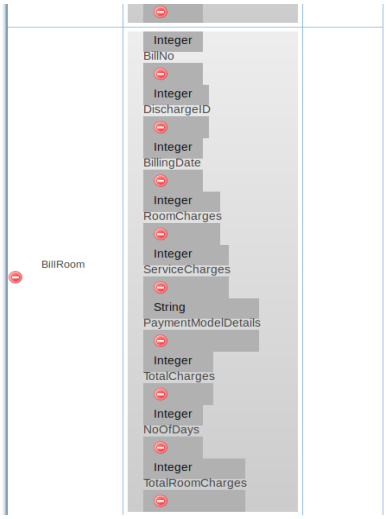
6. Room



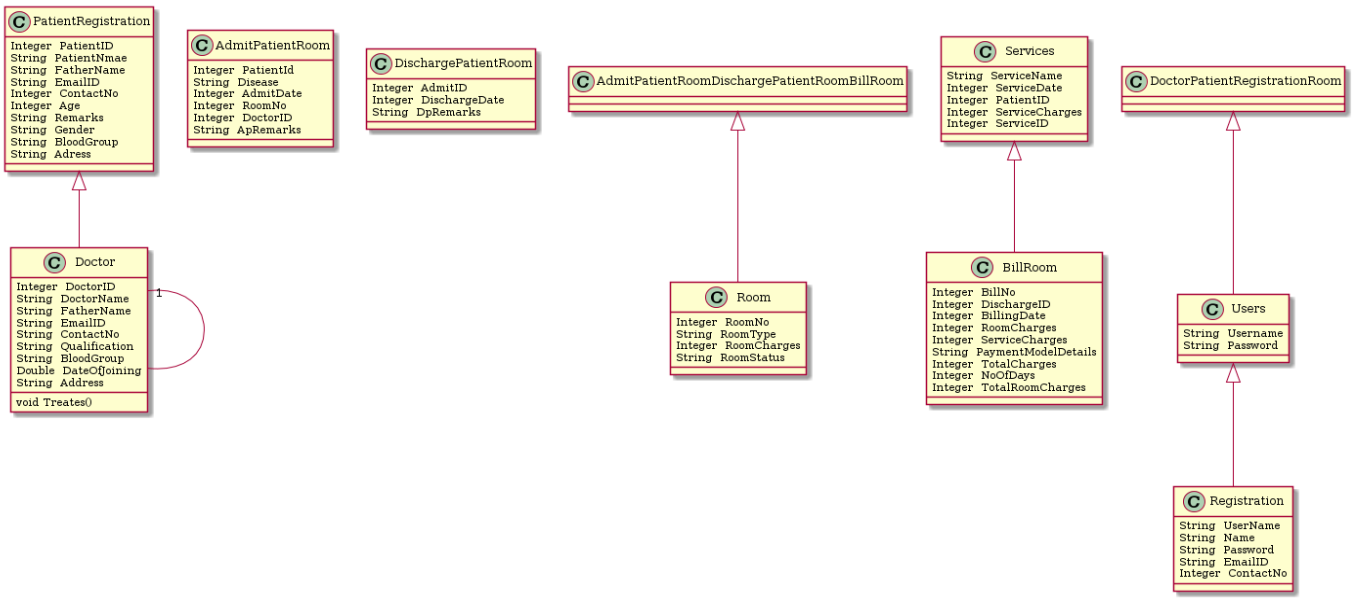
7. Services



8. BillRoom


BillRoom
Integer
BillNo
Integer
DischargeID
Integer
BillingDate
Integer
RoomCharges
Integer
ServiceCharges
String
PaymentModelDetails
Integer
TotalCharges
Integer
NoOfDays
Integer
TotalRoomCharges

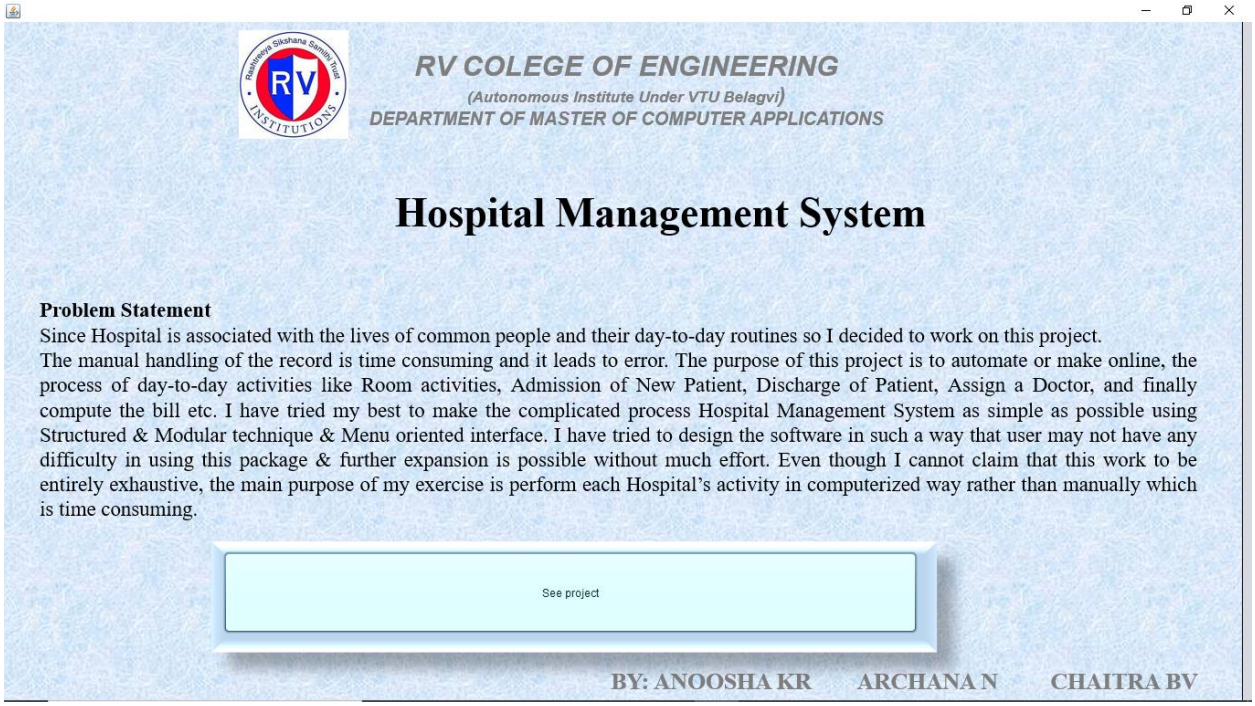
5. CLASS DIAGRAM



6.SCREENSHOTS

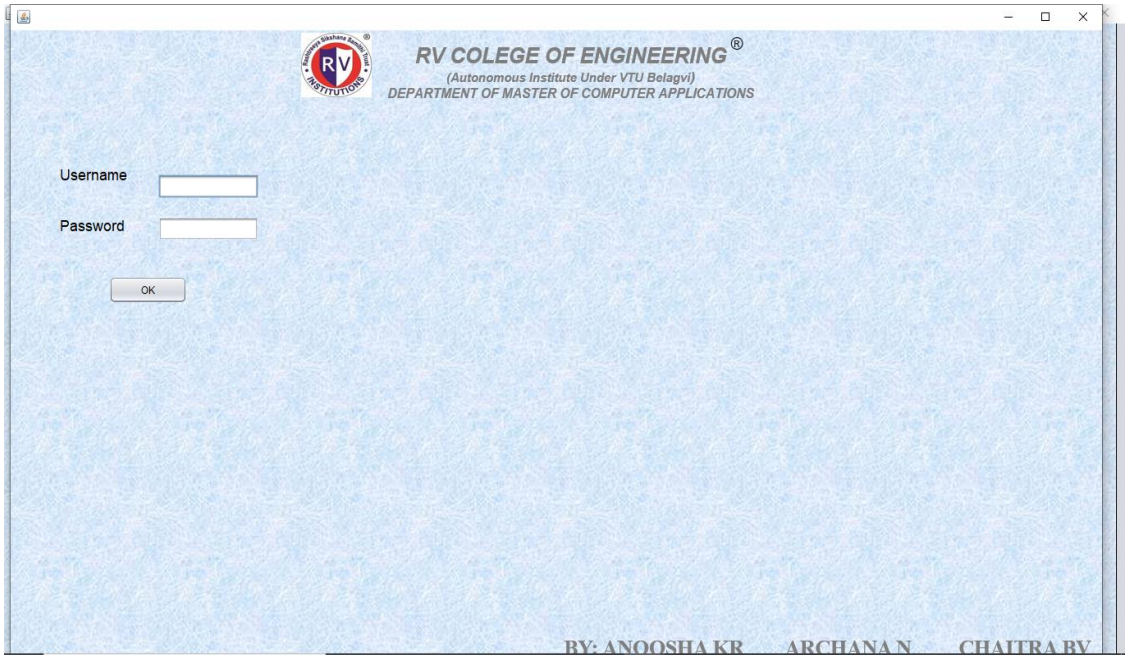
HOME PAGE

This is the first page displayed when the program is run



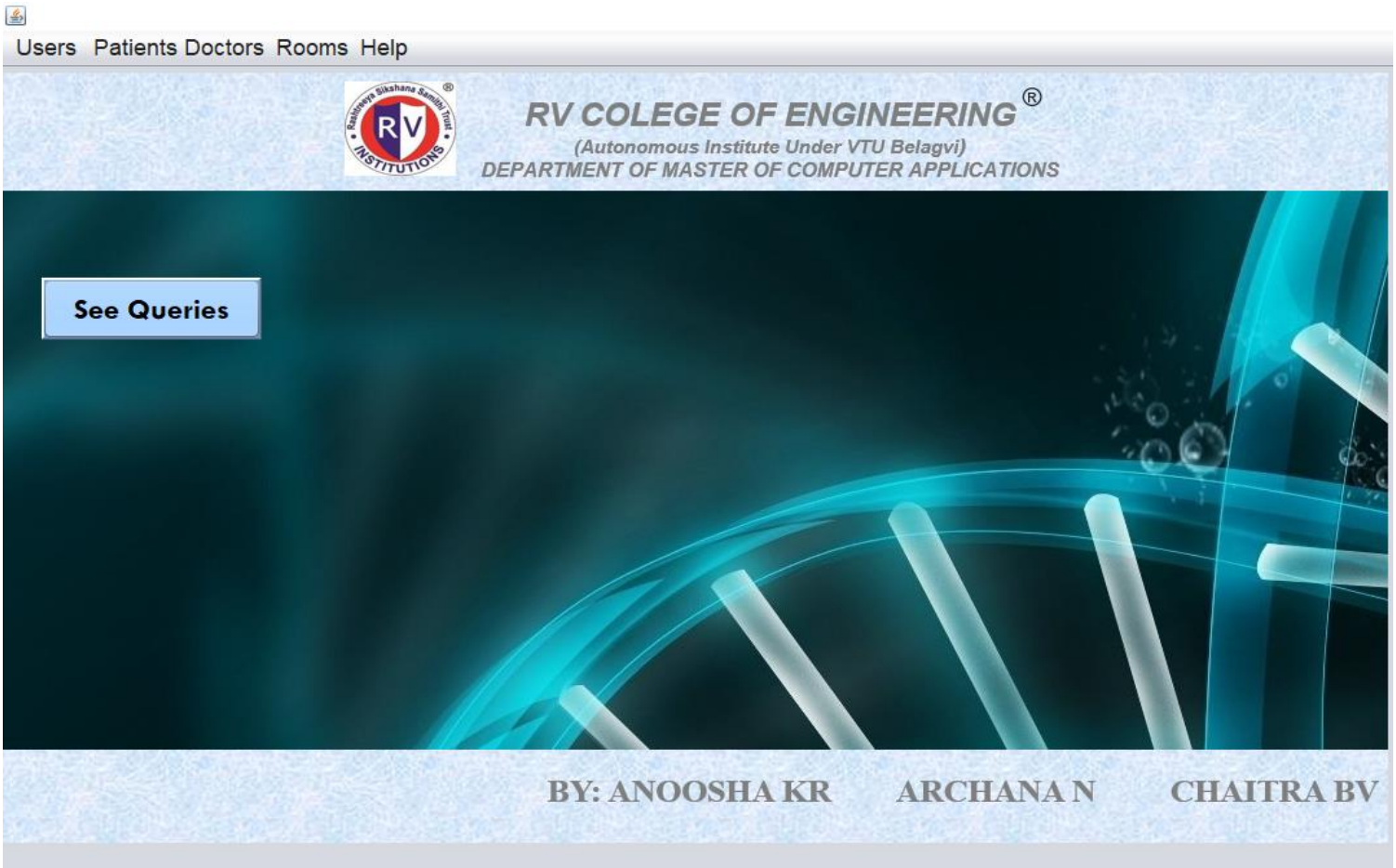
LOGIN FORM

Admin can logged in through correct username or password. If username or password is incorrect then it will give an error message.



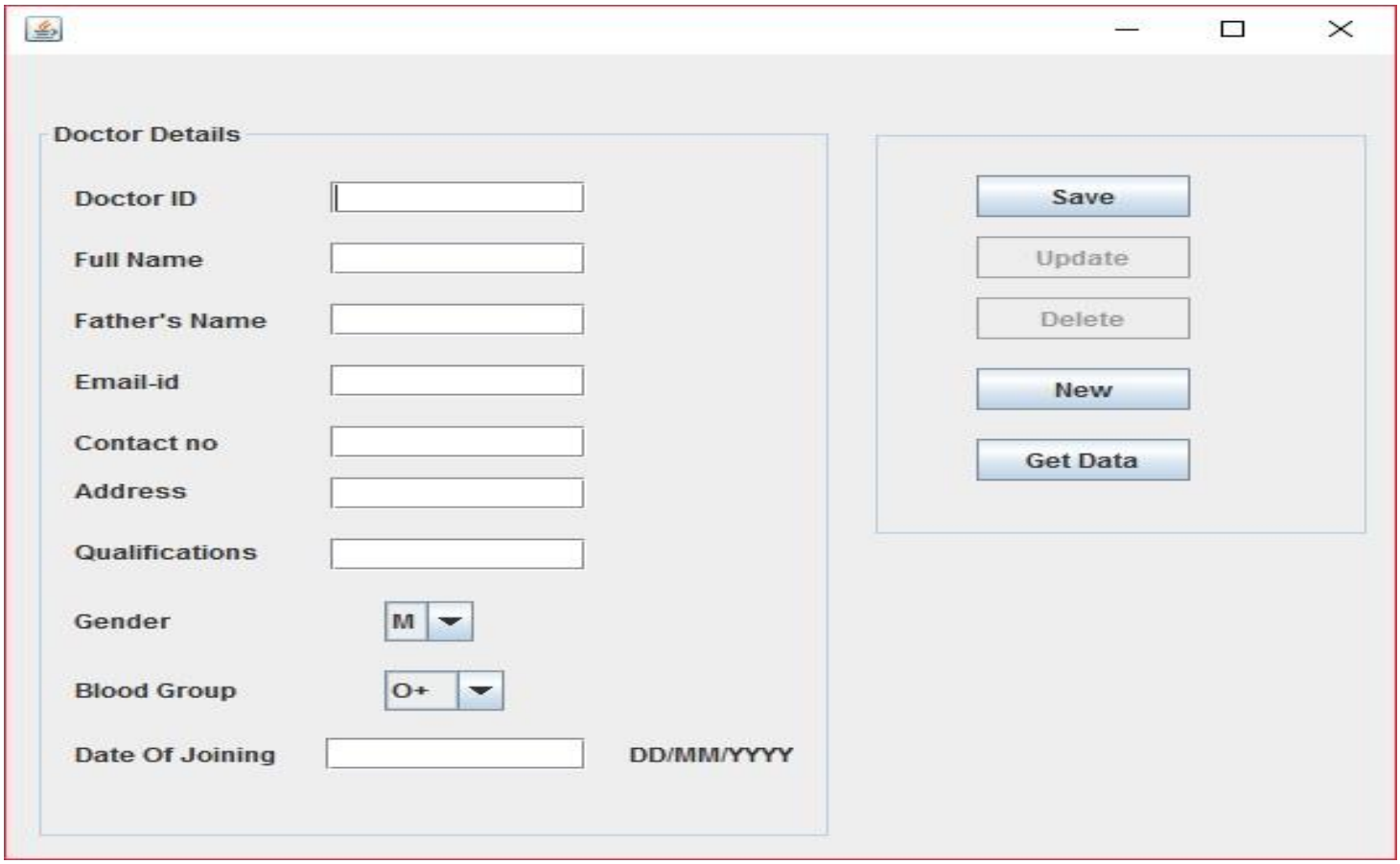
WELCOME PAGE

This is the main page of the program.



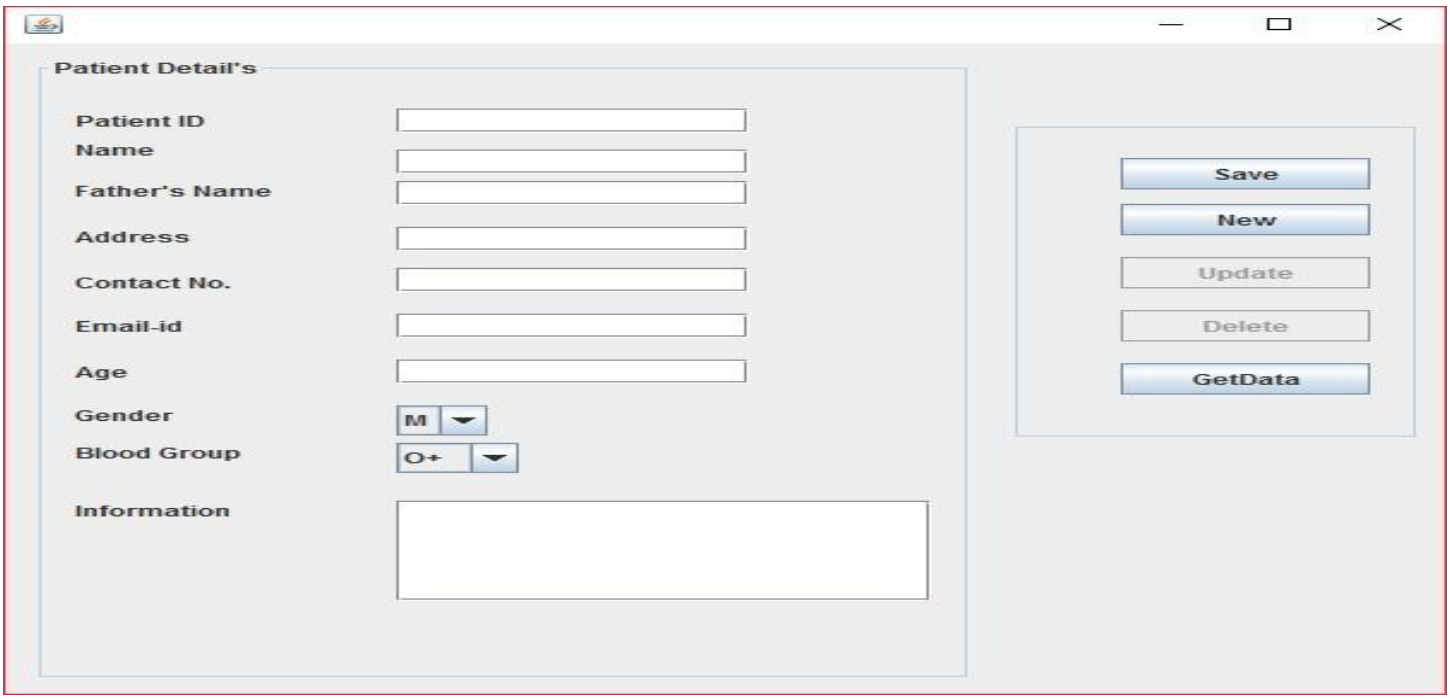
DOCTORES DETAILS

Here we can enter the details of doctor.and also we can save or Delete or update the data. We can display the details of doctors using getdata.



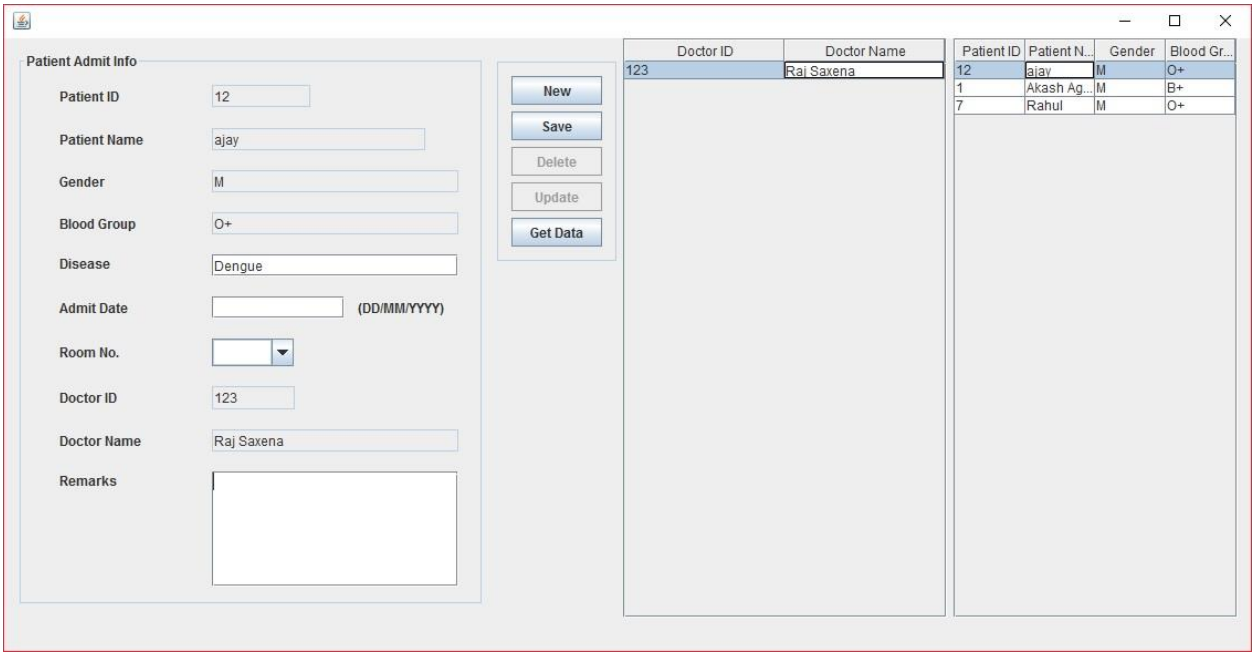
PATIENT DETAILS

here we can enter the details of patient.also we can add the new patient details..and we can delete or update the data..using getdata we can see the already registerd patient details.



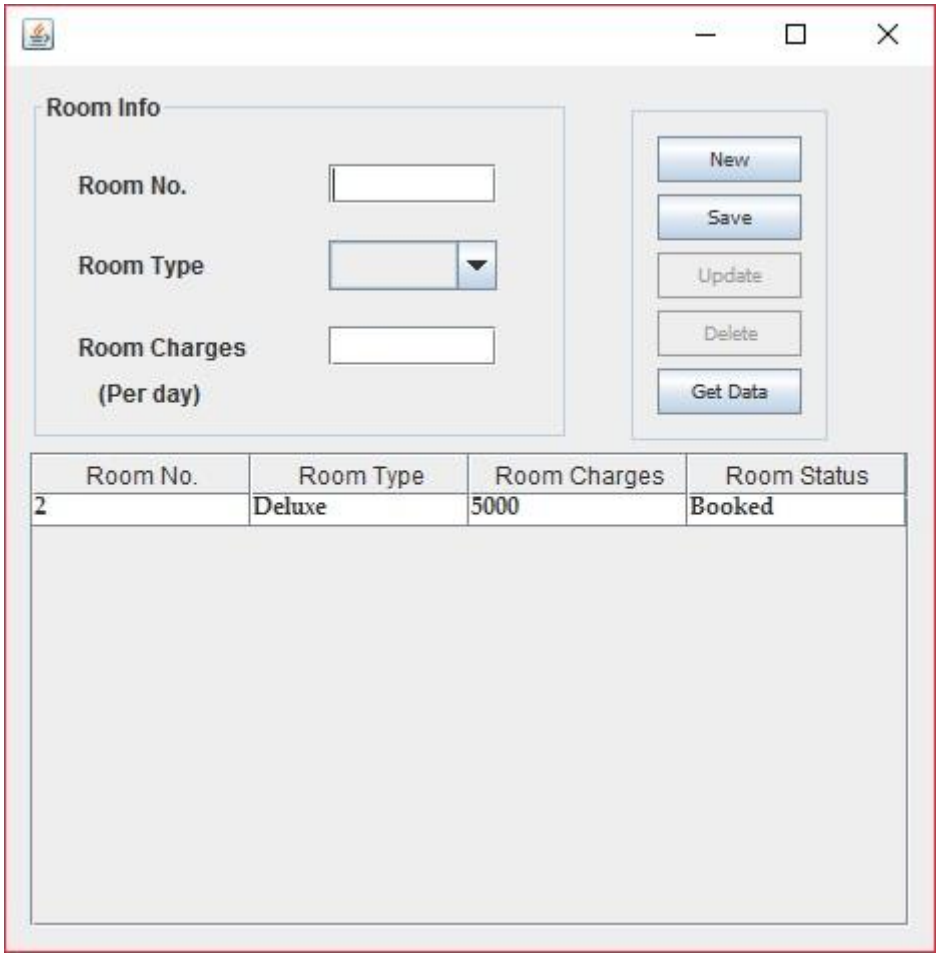
PATIENT ADMIT FORM

This is admission form of patient.



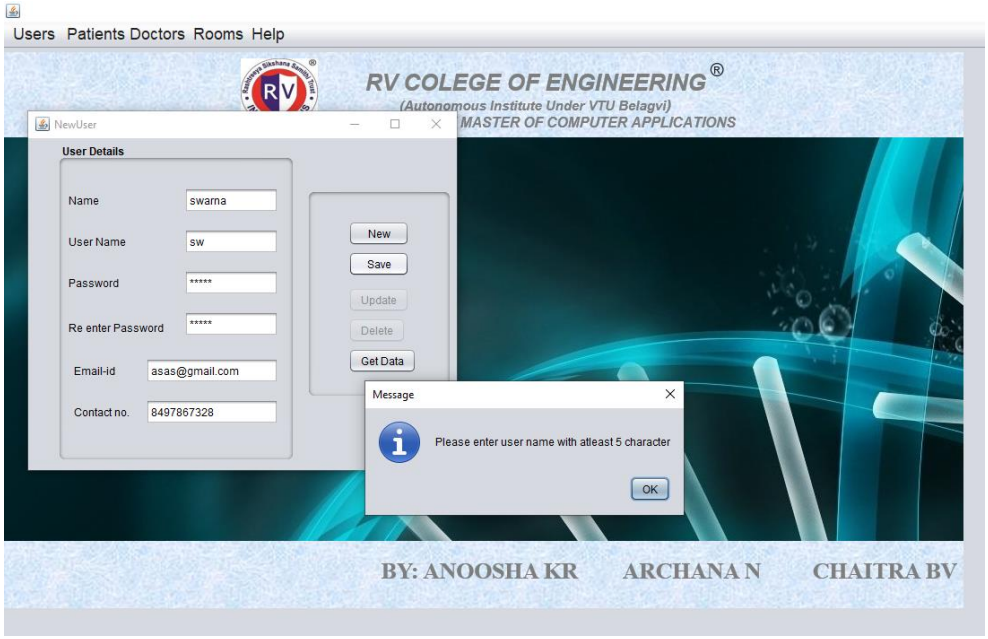
ROOM DETAILS

Room details form

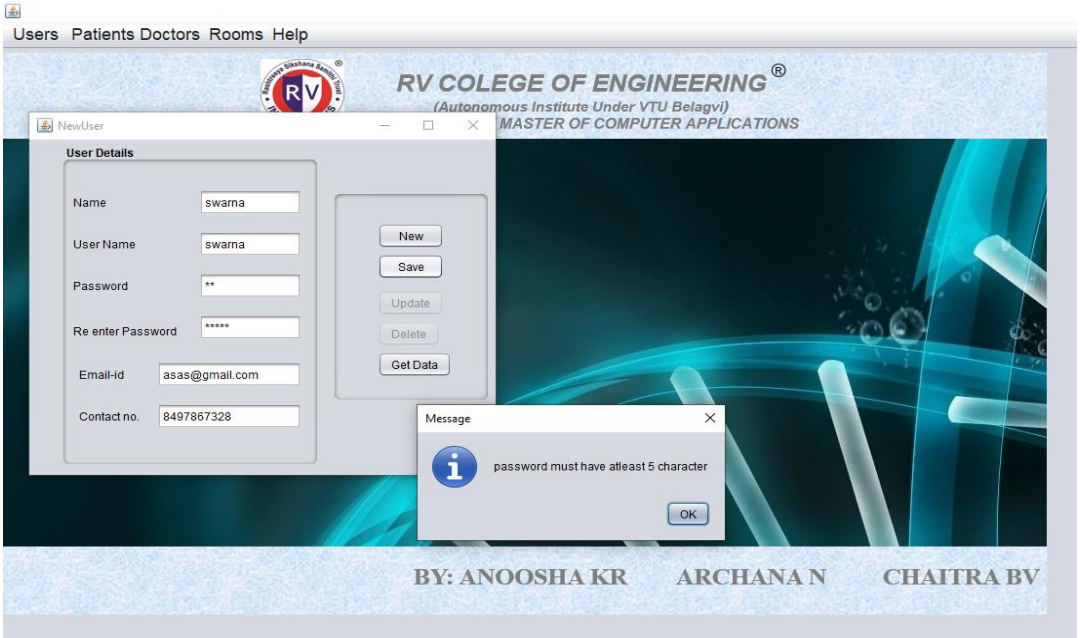


7.Validation screenshots

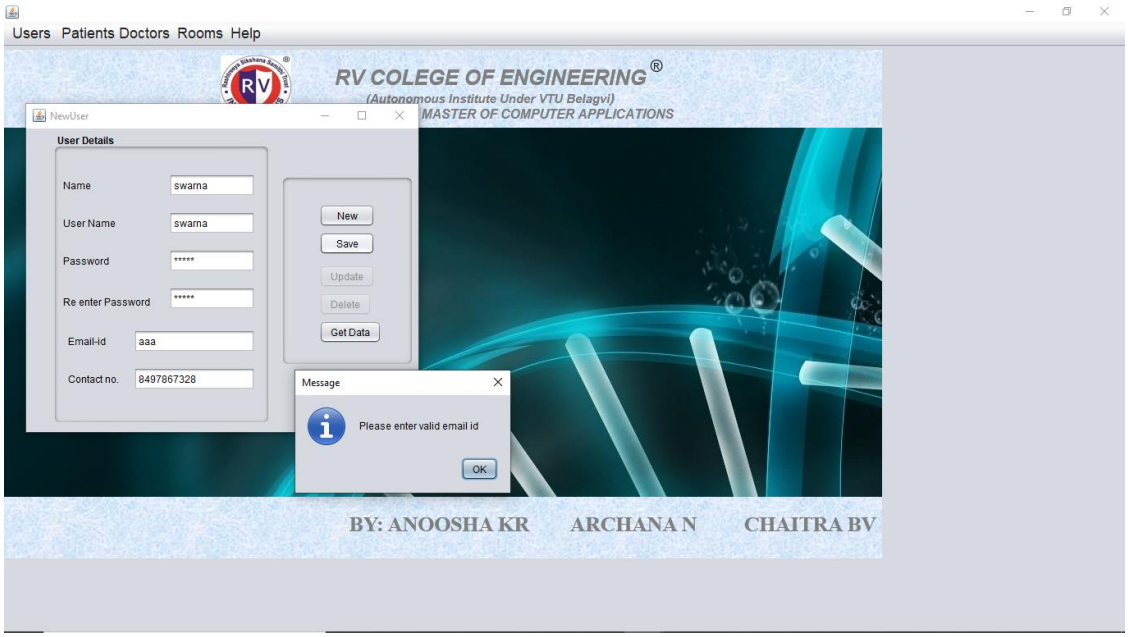
1.UserName Validation



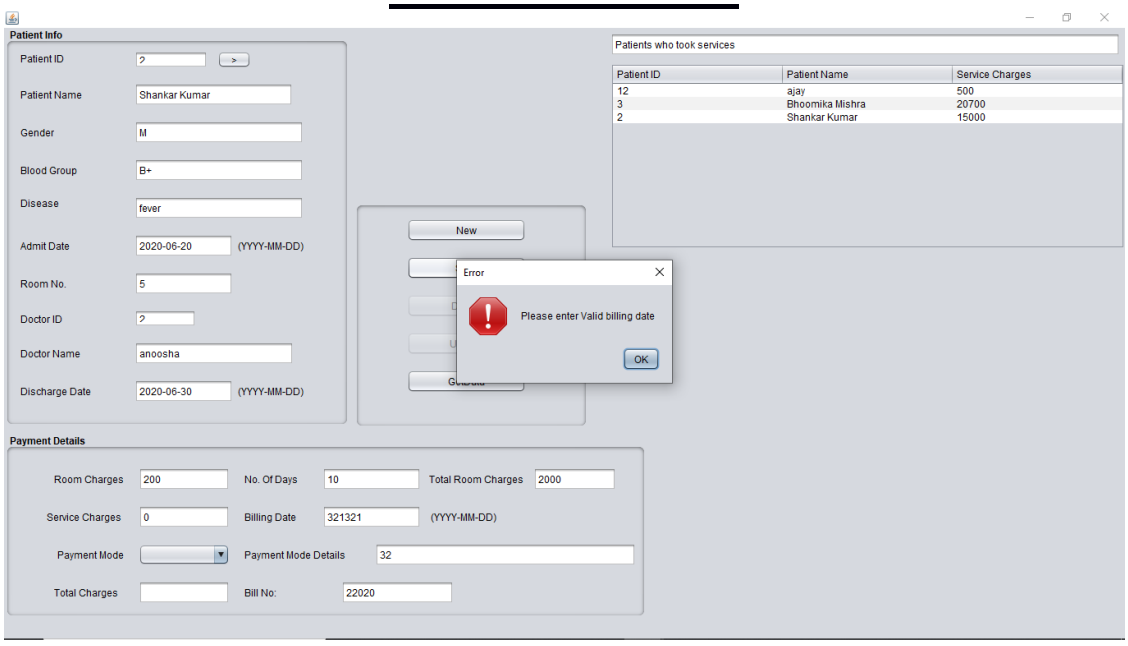
2.Password validation



3.Email validation



4.Date validation



8.Code:

Login Page

```
import java.awt.HeadlessException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import java.sql.*;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

public class login extends javax.swing.JFrame {

    /**
     * Creates new form login
     */
    public login() {
        initComponents();
    }
```

```
    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        txtUsername = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jPassword = new javax.swing.JPasswordField();
        jLabel4 = new javax.swing.JLabel();

        setBackground(new java.awt.Color(255, 153, 153));
        setForeground(new java.awt.Color(255, 153, 153));
        getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

        jLabel1.setFont(new java.awt.Font("Arial", 0, 18)); // NOI18N
```

```
        jLabel1.setText("Username");
        getContentPane().add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60,
160, -1, -1));

        jLabel2.setFont(new  java.awt.Font("Arial",  0,
18)); // NOI18N
        jLabel2.setText("Password");
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60,
220, -1, -1));
        getContentPane().add(txtUsername, new
org.netbeans.lib.awtextra.AbsoluteConstraints(180,
170, 122, -1));

        jButton1.setText("OK");
        jButton1.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        getContentPane().add(jButton1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120,
290, 95, 32));
        getContentPane().add(jPassword, new
org.netbeans.lib.awtextra.AbsoluteConstraints(180,
220, 122, -1));
```

```
        jLabel4.setIcon(new
javax.swing.ImageIcon("C:\\Users\\Nethra\\Pictures
\\Screenshots\\Screenshot (84).png")); // NOI18N
        jLabel4.setText("jLabel4");
        getContentPane().add(jLabel4,                new
org.netbeans.lib.awtextra.AbsoluteConstraints(0,   0,
1330, 760));

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private                                          void
jButton1ActionPerformed(java.awt.event.ActionEv
ent                  evt)                  {//GEN-
FIRST:event_jButton1ActionPerformed
        Connection con=null;
        ResultSet rs=null;
        PreparedStatement pst=null;
        if (txtUsername.getText().equals("")) {
            JOptionPane.showMessageDialog(          this,
"Please enter user name");
            return;
        }

        if (jPassword.getText().equals("")) {
            JOptionPane.showMessageDialog(          this,
"Please enter password");
            return;

            }
```

```
        con=Connect.ConnectDB();
        String sq1= "select * from users where
user_name= '" + txtUsername.getText() + "' and
password ='" + jPassword.getText() + "'";
        try{
            pst=con.prepareStatement(sq1);
            rs= pst.executeQuery();
            if (rs.next()){
             this.hide();
             Main frm=new Main();
             frm.setVisible(true);
            }
            else{
              JOptionPane.showMessageDialog(null,
"Login Failed..Try again !");
            }
        }catch(SQLException | HeadlessException e){
         JOptionPane.showMessageDialog(null, e);
                // TODO add your handling code here:
    }//GEN-LAST:event_jButton1ActionPerformed
    }
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold  defaultstate="collapsed"  desc="
Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not
available, stay with the default look and feel.
```

```
        * For details see
http://download.oracle.com/javase/tutorial/uiswing/l
ookandfeel/plaf.html
        */
       try {
          for
(javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels(
)) {
              if ("Metal".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getCl
assName());
                  break;
              }
          }
       } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(login.class.getN
ame()).log(java.util.logging.Level.SEVERE,     null,
ex);
       } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(login.class.getN
ame()).log(java.util.logging.Level.SEVERE,     null,
ex);
       } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(login.class.getN
ame()).log(java.util.logging.Level.SEVERE,     null,
```

```
ex);
        } catch
(javax.swing.UnsupportedLookAndFeelException
ex) {

java.util.logging.Logger.getLogger(login.class.getN
ame()).log(java.util.logging.Level.SEVERE, null,
ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new
Runnable() {
            public void run() {
                new login().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify//GEN-
BEGIN:variables
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPasswordField jPassword;
    private javax.swing.JTextField txtUsername;
    // End of variables declaration//GEN-
END:variables
}
```

8.Conclusion

This project has been a rewarding experience in more than one way. The entire project work has enlightened us in the following areas.

a) We have gained an insight into the working of the HOSPITAL. This represents a typical real world situation.

b) Our understanding of database design has been strengthened this is because in order to generate the final reports of database designing has to be properly followed.

c) Scheduling a project and adhering to that schedule creates a strong sense of time management.

d) Sense of teamwork has developed and confidence of handling real life project has increased to a great extent.

13.References

[1] Deepak Thomas ”Beginning Java Programmings”, Wrox Press Ltd. Paperback-17, October, 2002.70-130 pp.

[2] Matt Doyle, “DBS 5.3, 2nd edition”, October 2009. 150-270 pp.

[3] Luke Welling, Laura Thomson. Sams MySQL Web Development, 2nd edition, Paperback- 20 February, 2003. 105-209 pp.

[4] W. Jason Gilmore “Beginning JAVA and MySQL 5 from Novice to Professional SECOND EDITION”, Jul 9, 2008.100-150 pp.

[5] Abraham Silberschatz, Henry F. Korth and S. Sudarshan “Sixth Edition Java Programming System Conceptsreleased”, January 28, 2010. 206-253 pp.

[6] Server-Side Scriptinghttp://Mysql.net/manual/en/index.php,

[7] Java Programming System https://www.w3schools.com/,

[8] Netbeans.com

[9] https://stackoverflow.com/,