MINI PROJECT REPORT

on

"Student Monitoring System

By

Tanvi Hirlekar, Tanvi Nehete, Vaishnavi Patil

SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF BACHELOR OF ENGINEERING

In

Computer Engineering

SIES Graduate School of Technology

RISE WITH EDUCATION

DEPARTMENT OF COMPUTER ENGINEERING

SIES GRADUATE SCHOOL OF TECHNOLOGY

NERUL, NAVI MUMBAI – 400706

ACADEMIC YEAR

2023 – 2024

CONTENTS

ABSTRACT

The "Student Monitoring System" is a Java Swing-based application designed to streamline and enhance the management of student data in an educational institution. This comprehensive system offers an intuitive and user-friendly interface, making it an invaluable tool for teachers.With a dedicated login window, this system provides a secure and tailored platform for teachers to access and update student information.The system is designed exclusively for teachers, ensuring that only authorized educators can access the platform.Teachers can easily access and update student details, including personal information, contact details, offering streamlined administrative support.

The system empowers teachers to record and update subject-wise marks for each student. This feature is crucial for tracking and managing student academic performance.Teachers can monitor student attendance, keeping a close watch on presence and absence, aiding in effective classroom management.Utilizing the Java Swing library, the system offers an intuitive graphical user interface (GUI), ensuring ease of use and accessibility for educators.The system establishes user roles, granting teachers specific privileges and access rights while ensuring data security and privacy. A robust search and filtering mechanism simplifies the process of locating specific student records, particularly beneficial in larger educational institutions.By implementing this system, educational institutions can expect improved data management and administrative efficiency while empowering teachers to take an active role in managing and updating student information, all within a secure and teacher-focused platform.

INTRODUCTION

In the dynamic landscape of modern education, the effective management of student data and academic progress is a pivotal task. Educational institutions constantly seek innovative solutions to streamline administrative processes, enhance teacher-student communication, and provide educators with tools to monitor and guide their students effectively. In response to this demand, we introduce the " Student Monitoring System," a Java Swing-based application specifically designed to meet the unique needs of teachers and instructors within educational settings.

This project aims to address the multifaceted challenges faced by educators by offering a dedicated platform where teachers can access, update, and oversee student information effortlessly. The heart of this system lies in its user-centric design, ensuring that only authorized teachers gain access via a secure login window, creating a private and efficient space for educators to engage in data management. This system provides a comprehensive array of features, encompassing student information management, subject and marks recording, attendance tracking, and report generation. These features are strategically integrated to empower educators, streamline administrative tasks, and enhance the overall quality of education within an institution.

The "Student Monitoring System" is not merely an administrative tool; it's a pedagogical companion. With its user-friendly interface, teachers can effortlessly record subject-wise marks, monitor attendance, and maintain student records. The system generates essential reports for assessing academic progress, allowing educators to guide their students effectively.

Through this project, we aspire to empower educators at all educational levels, from primary schools to higher education institutions. By offering teachers the means to actively manage and update student data, we aim to enhance data accuracy, communication, and efficiency within educational organizations. This project represents a step forward in redefining the relationship between technology and education, making data management a seamless and valuable aspect of the educational journey.

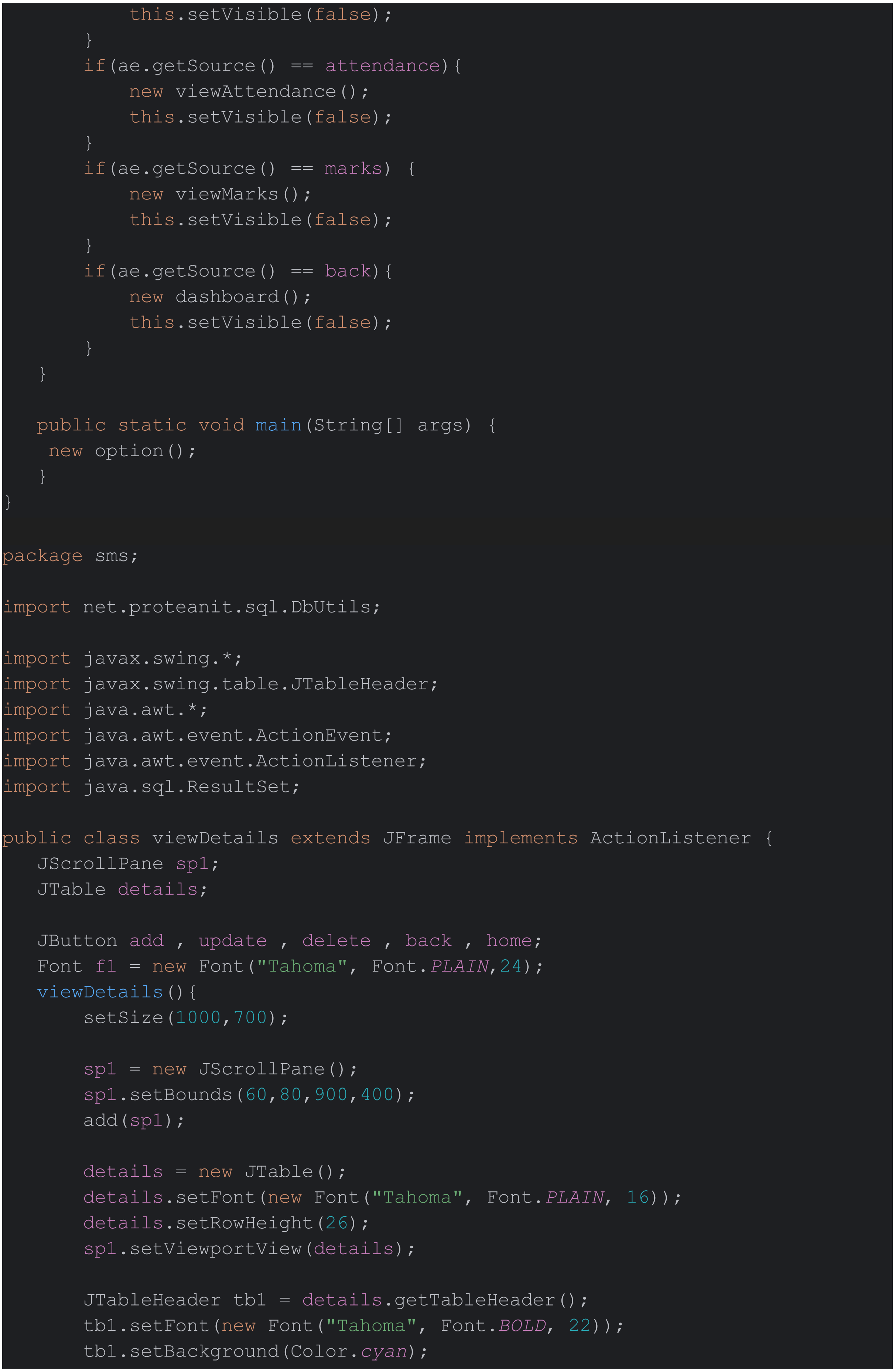
DETAILED SYSTEM DESIGN

Input Code :

```
package sms;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class starter extends JFrame implements ActionListener {

   JButton login;
   starter(){
       setSize(800,700);
       JLabel l1=new JLabel("SIES");
       l1.setBounds(325,50,500,100);
       l1.setForeground(Color.white);
       l1.setFont(new Font("TAHOMA", Font.BOLD,50));
       add(l1);

       login=new JButton("LOGIN");
       login.setBounds(650,550,100,70);
       login.setFont(new Font("Ariel",Font.ITALIC,15));
       login.addActionListener(this);
       add(login);

       ImageIcon i2 = new
ImageIcon(ClassLoader.getSystemResource("images/b9.jpg"));
       JLabel i3 = new JLabel(i2);
       i3.setBounds(0,0,900,800);
       add(i3);

       setLayout(null);
       setLocationRelativeTo(null);
       setVisible(true);
   }

   public void actionPerformed(ActionEvent ae){
       if(ae.getSource()==login){
           new login();
           this.setVisible(false);
       }
   }

   public static void main(String[] args) {
       new starter();
```

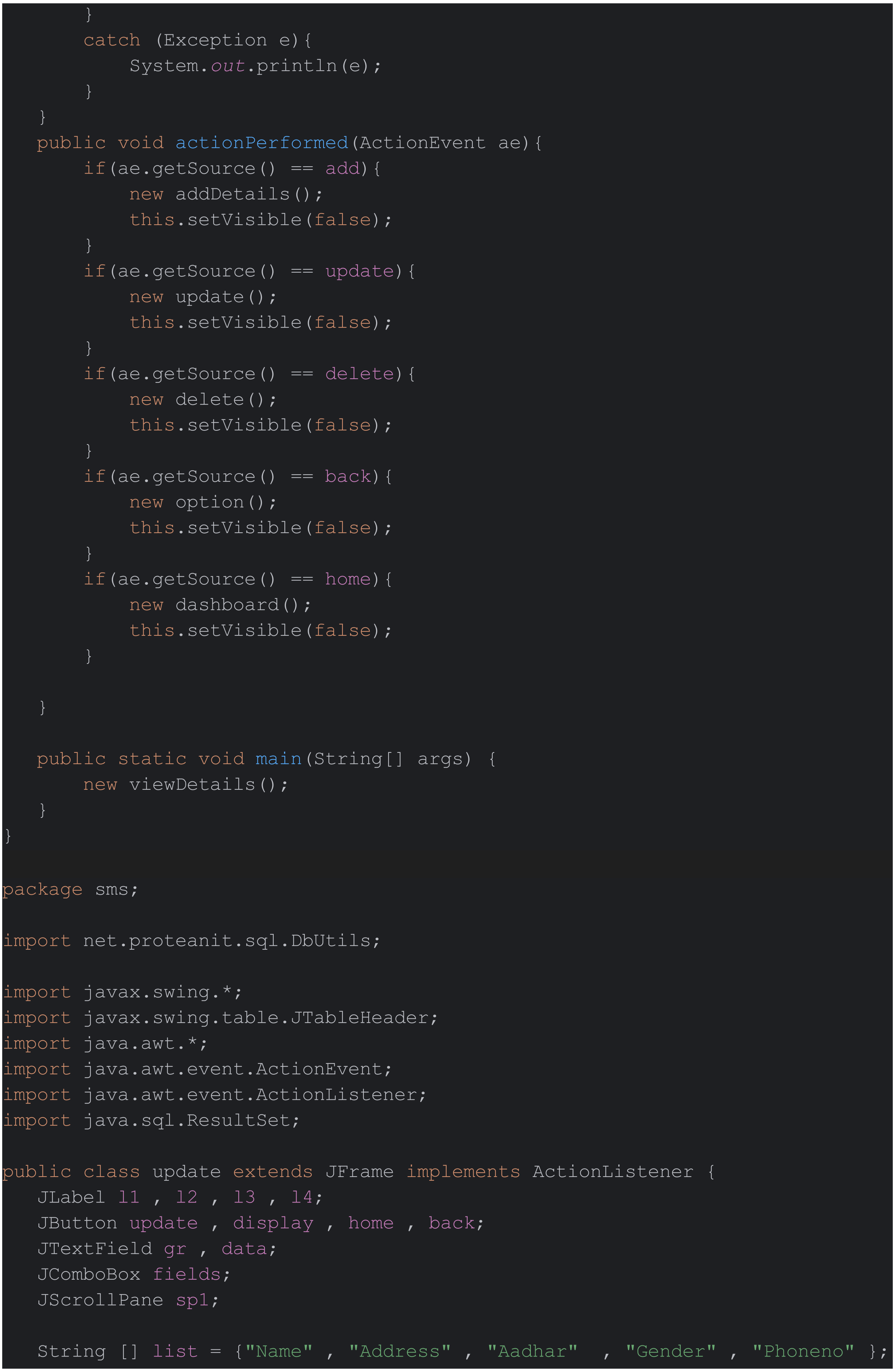
```
    }
}
package sms;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;

public class login extends JFrame implements ActionListener {
    JTextField username;
    JPasswordField password;
    JButton signin;
    JButton cancel;

    login() {
        setSize(500, 350);


        JLabel l1 = new JLabel("USERNAME");
        l1.setBounds(80, 60, 150, 50);
        l1.setFont(new Font("Tahoma", Font.BOLD, 24));
        l1.setForeground(Color.WHITE);
        add(l1);

        JLabel l2 = new JLabel("PASSWORD");
        l2.setBounds(80, 120, 150, 50);
        l2.setFont(new Font("Tahoma", Font.BOLD, 24));
        l2.setForeground(Color.WHITE);
        add(l2);

        username = new JTextField();
        username.setBounds(250, 60, 150, 40);
        username.setFont(new Font("Times New Roman", Font.PLAIN, 18));
        add(username);

        password = new JPasswordField();
        password.setBounds(250, 120, 150, 40);
        add(password);

        signin = new JButton("SIGN IN");
        signin.setBounds(320, 220, 110, 20);
        signin.addActionListener(this);
        add(signin);

        cancel = new JButton("CANCEL");
        cancel.setBounds(320, 260, 110, 20);
        cancel.addActionListener(this);
        add(cancel);
        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("images/b15.jpg"));
        JLabel i3 = new JLabel(i1);
        i3.setBounds(0, 0, 500, 350);
        add(i3);


        setLayout(null);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == signin) {
            String username = this.username.getText();
            String password = String.valueOf(this.password.getPassword());
            conn c = new conn();
            try {
                String str = "select * from login where username = '" +
username + "' and password = '" + password + "';";
                ResultSet rs = c.s.executeQuery(str);
                if(rs.next()){
                    JOptionPane.showMessageDialog(null,"Login Successfull");

                    this.setVisible(false);
                    new dashboard();
                }
                else{
                    JOptionPane.showMessageDialog(null,"Invalid Login
Credentials");
                }


            }
            catch (Exception e) {
                System.out.println(e);
                JOptionPane.showMessageDialog(null,"Some Error Occurred");
            }
        }

        if(ae.getSource() == cancel){
            System.exit(0);
        }

    }

        public static void main (String[]args){

            new login();
        }
}

package sms;
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class option extends JFrame implements ActionListener  {
   JButton details , attendance , marks , back;
   Font f1 = new Font("Tahoma", Font.BOLD,24);
   option(){
       setSize(700,400);


       details = new JButton("Details");
       details.setBounds(50,140,180,50);
       details.setFont(f1);
       details.addActionListener(this);
       add(details);

       attendance = new JButton("Attendance");
       attendance.setBounds(250,140,180,50);
       attendance.setFont(f1);
       attendance.addActionListener(this);
       add(attendance);

       marks = new JButton("Marks");
       marks.setBounds(450,140,180,50);
       marks.setFont(f1);
       marks.addActionListener(this);
       add(marks);

       ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("images/back.png"));
       Image i2 = i1.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
       ImageIcon i3 = new ImageIcon(i2);
       back = new JButton(i3);
       back.setBounds(10,10,50,50);
       back.addActionListener(this);
       add(back);
       ImageIcon i12 = new
ImageIcon(ClassLoader.getSystemResource("images/b9.jpg"));
       JLabel i13 = new JLabel(i12);
       i13.setBounds(0,0,900,800);
       add(i13);

       setLayout(null);
       setLocationRelativeTo(null);
       setVisible(true);

   }
   public void actionPerformed(ActionEvent ae){
       if(ae.getSource() == details){
           new viewDetails();
```

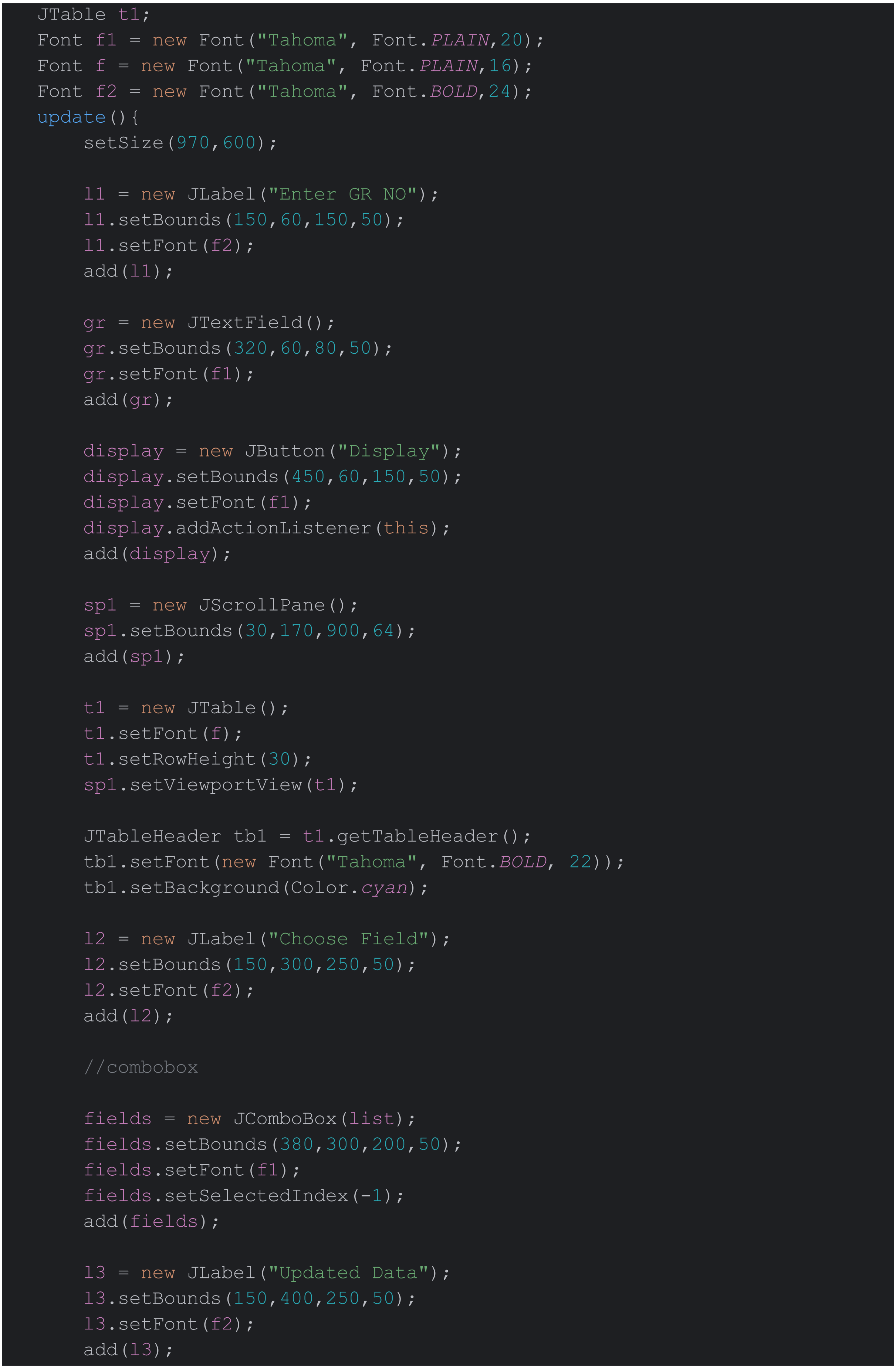


```
           this.setVisible(false);
       }
       if(ae.getSource() == attendance){
           new viewAttendance();
           this.setVisible(false);
       }
       if(ae.getSource() == marks) {
           new viewMarks();
           this.setVisible(false);
       }
       if(ae.getSource() == back){
           new dashboard();
           this.setVisible(false);
       }
   }

   public static void main(String[] args) {
    new option();
   }
}

package sms;

import net.proteanit.sql.DbUtils;

import javax.swing.*;
import javax.swing.table.JTableHeader;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;

public class viewDetails extends JFrame implements ActionListener {
   JScrollPane sp1;
   JTable details;

   JButton add , update , delete , back , home;
   Font f1 = new Font("Tahoma", Font.PLAIN,24);
   viewDetails(){
       setSize(1000,700);

       sp1 = new JScrollPane();
       sp1.setBounds(60,80,900,400);
       add(sp1);

       details = new JTable();
       details.setFont(new Font("Tahoma", Font.PLAIN, 16));
       details.setRowHeight(26);
       sp1.setViewportView(details);

       JTableHeader tb1 = details.getTableHeader();
       tb1.setFont(new Font("Tahoma", Font.BOLD, 22));
       tb1.setBackground(Color.cyan);
```

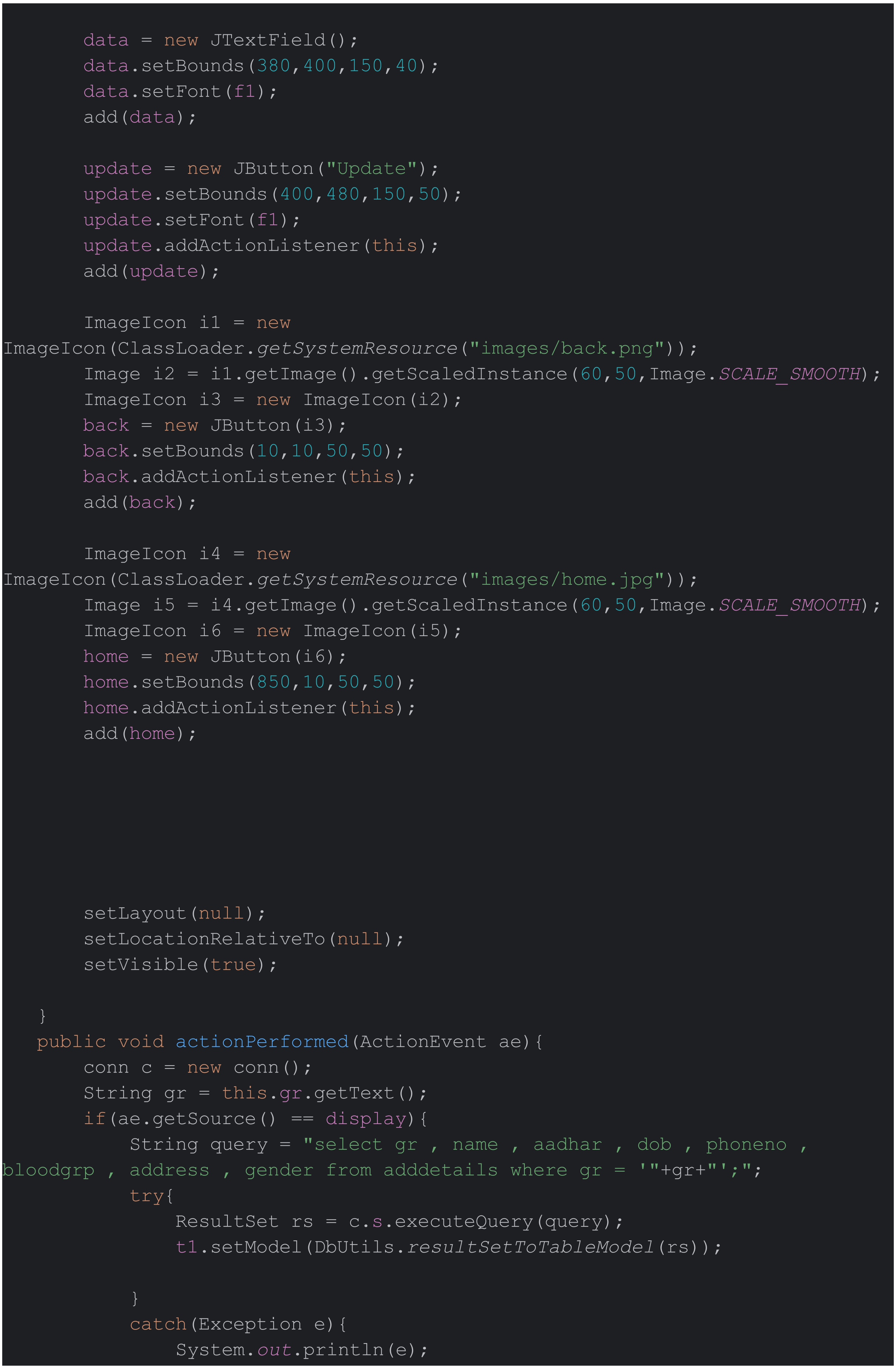
```
        getDetails();

        add = new JButton("Add");
        add.setBounds(200,550,150,50);
        add.setFont(f1);
        add.addActionListener(this);
        add(add);

        update = new JButton("Update");
        update.setBounds(400,550,150,50);
        update.setFont(f1);
        update.addActionListener(this);
        add(update);

        delete = new JButton("Delete");
        delete.setBounds(600,550,150,50);
        delete.setFont(f1);
        delete.addActionListener(this);
        add(delete);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("images/back.png"));
        Image i2 = i1.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i3 = new ImageIcon(i2);
        back = new JButton(i3);
        back.setBounds(10,10,50,50);
        back.addActionListener(this);
        add(back);

        ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("images/home.jpg"));
        Image i5 = i4.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i6 = new ImageIcon(i5);
        home = new JButton(i6);
        home.setBounds(930,10,50,50);
        home.addActionListener(this);
        add(home);




        setLayout(null);
        setLocationRelativeTo(null);
        setVisible(true);
    }
    public void getDetails(){
        conn c = new conn();
        String query = "select gr , name , aadhar , phoneno , bloodgrp ,
address , gender , dob from adddetails";
        try{
            ResultSet rs = c.s.executeQuery(query);
            details.setModel(DbUtils.resultSetToTableModel(rs));
```



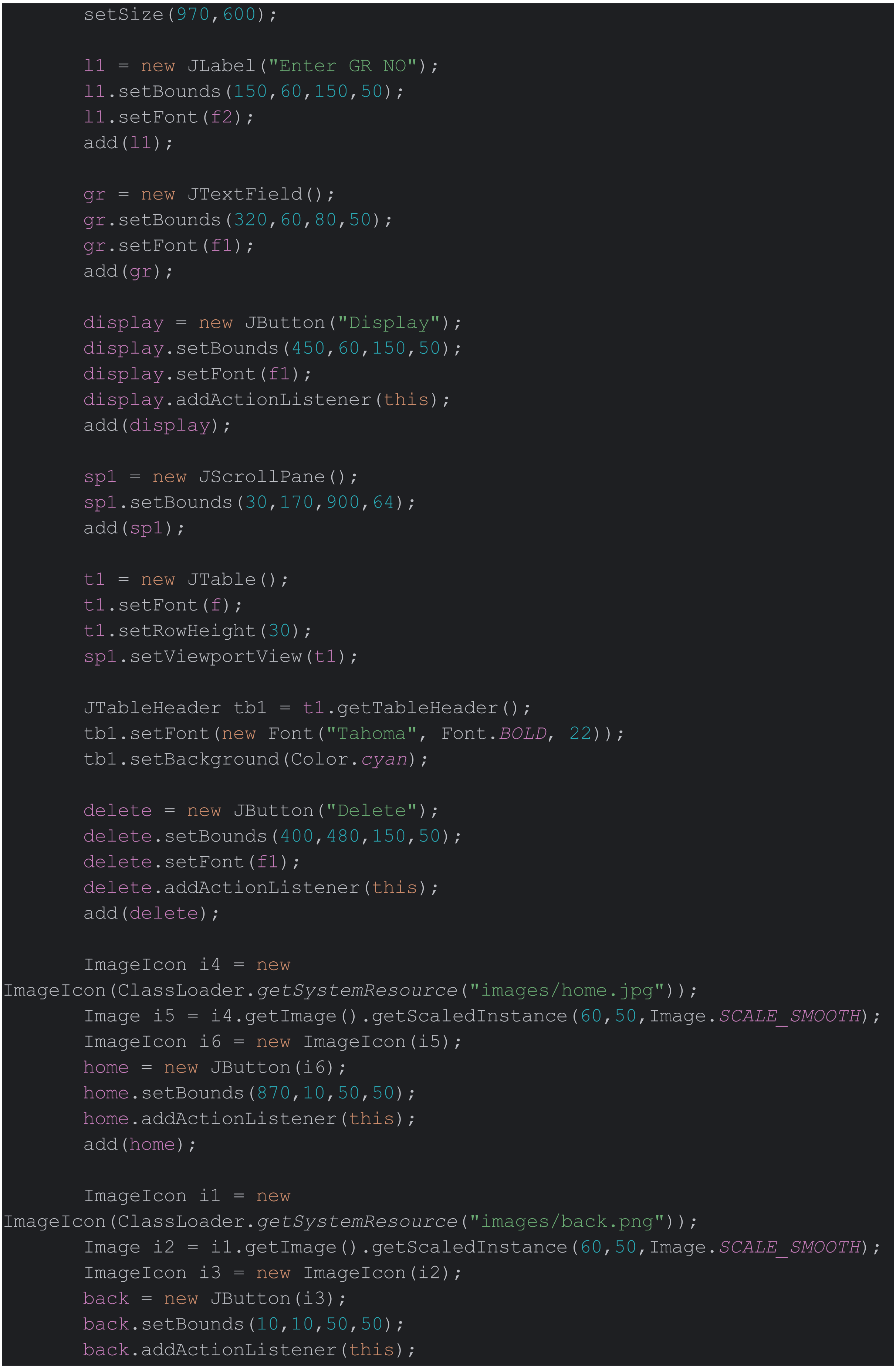
```
        }
        catch (Exception e){
            System.out.println(e);
        }
    }
    public void actionPerformed(ActionEvent ae){
        if(ae.getSource() == add){
            new addDetails();
            this.setVisible(false);
        }
        if(ae.getSource() == update){
            new update();
            this.setVisible(false);
        }
        if(ae.getSource() == delete){
            new delete();
            this.setVisible(false);
        }
        if(ae.getSource() == back){
            new option();
            this.setVisible(false);
        }
        if(ae.getSource() == home){
            new dashboard();
            this.setVisible(false);
        }

    }

    public static void main(String[] args) {
        new viewDetails();
    }
}

package sms;

import net.proteanit.sql.DbUtils;

import javax.swing.*;
import javax.swing.table.JTableHeader;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;

public class update extends JFrame implements ActionListener {
    JLabel l1 , l2 , l3 , l4;
    JButton update , display , home , back;
    JTextField gr , data;
    JComboBox fields;
    JScrollPane sp1;

    String [] list = {"Name" , "Address" , "Aadhar"  , "Gender" , "Phoneno" };
```



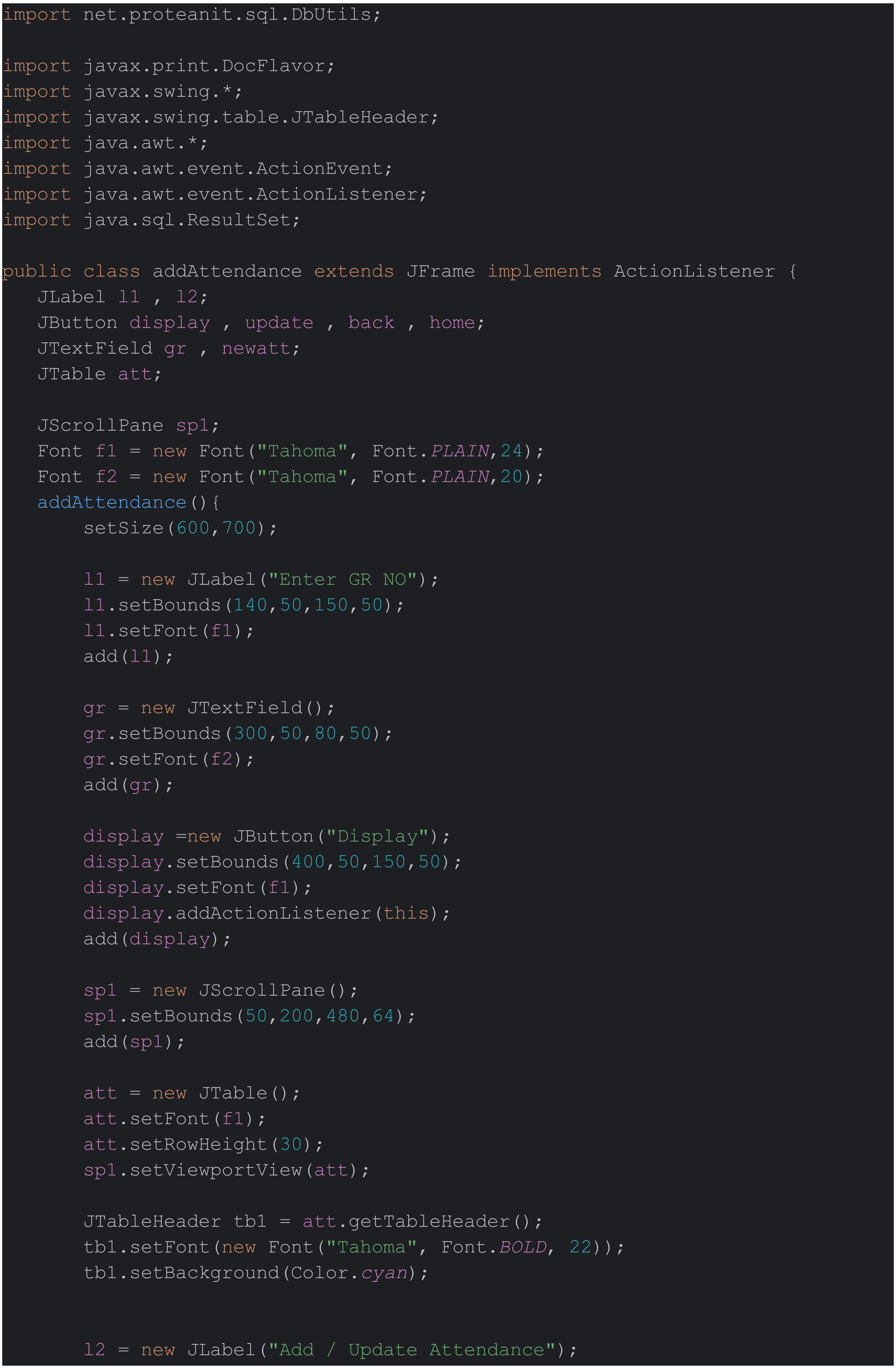
```
    JTable t1;
    Font f1 = new Font("Tahoma", Font.PLAIN,20);
    Font f = new Font("Tahoma", Font.PLAIN,16);
    Font f2 = new Font("Tahoma", Font.BOLD,24);
    update(){
        setSize(970,600);

        l1 = new JLabel("Enter GR NO");
        l1.setBounds(150,60,150,50);
        l1.setFont(f2);
        add(l1);

        gr = new JTextField();
        gr.setBounds(320,60,80,50);
        gr.setFont(f1);
        add(gr);

        display = new JButton("Display");
        display.setBounds(450,60,150,50);
        display.setFont(f1);
        display.addActionListener(this);
        add(display);

        sp1 = new JScrollPane();
        sp1.setBounds(30,170,900,64);
        add(sp1);

        t1 = new JTable();
        t1.setFont(f);
        t1.setRowHeight(30);
        sp1.setViewportView(t1);

        JTableHeader tb1 = t1.getTableHeader();
        tb1.setFont(new Font("Tahoma", Font.BOLD, 22));
        tb1.setBackground(Color.cyan);

        l2 = new JLabel("Choose Field");
        l2.setBounds(150,300,250,50);
        l2.setFont(f2);
        add(l2);

        //combobox

        fields = new JComboBox(list);
        fields.setBounds(380,300,200,50);
        fields.setFont(f1);
        fields.setSelectedIndex(-1);
        add(fields);

        l3 = new JLabel("Updated Data");
        l3.setBounds(150,400,250,50);
        l3.setFont(f2);
        add(l3);
```



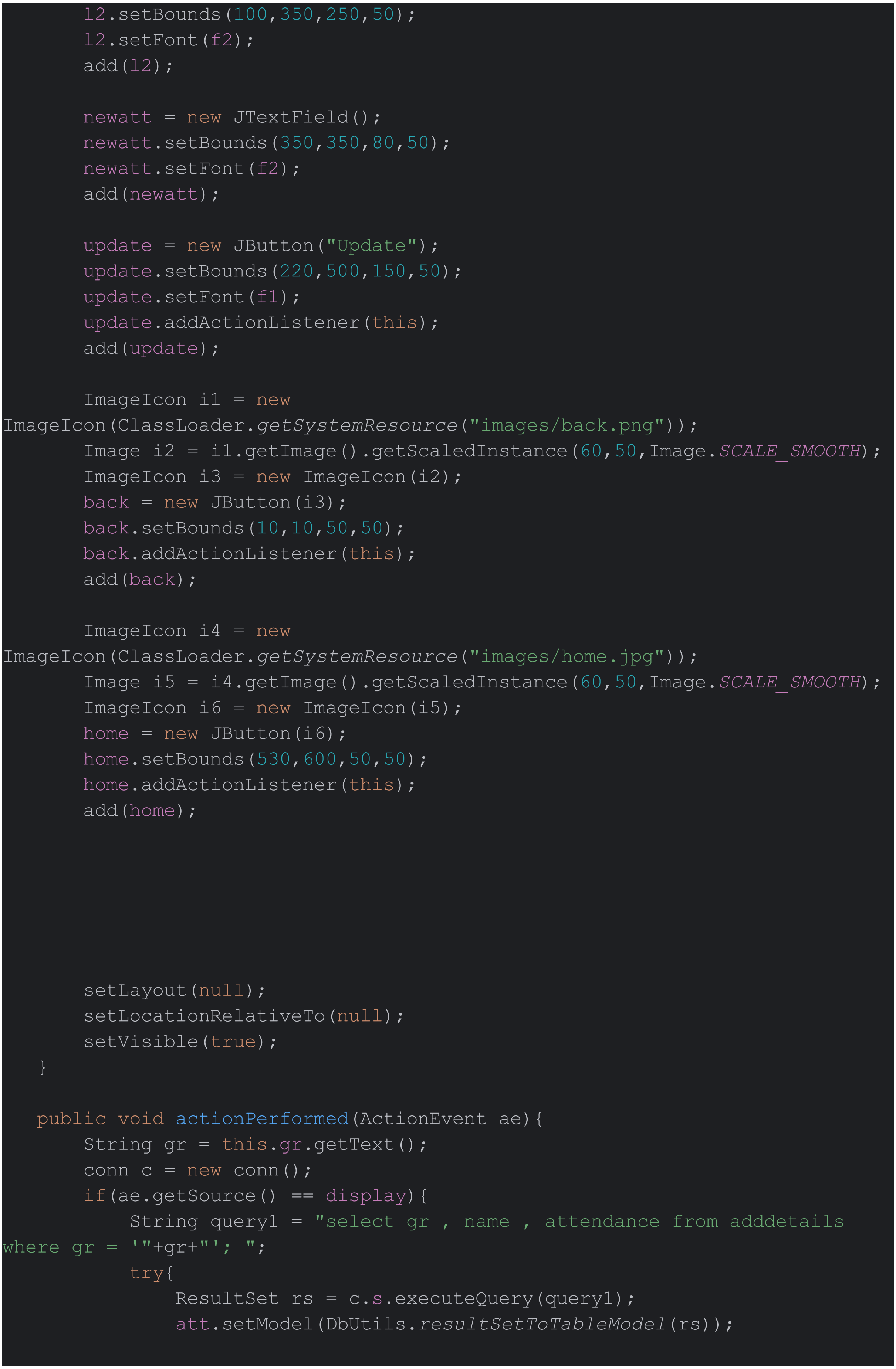
```
        data = new JTextField();
        data.setBounds(380,400,150,40);
        data.setFont(f1);
        add(data);

        update = new JButton("Update");
        update.setBounds(400,480,150,50);
        update.setFont(f1);
        update.addActionListener(this);
        add(update);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("images/back.png"));
        Image i2 = i1.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i3 = new ImageIcon(i2);
        back = new JButton(i3);
        back.setBounds(10,10,50,50);
        back.addActionListener(this);
        add(back);

        ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("images/home.jpg"));
        Image i5 = i4.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i6 = new ImageIcon(i5);
        home = new JButton(i6);
        home.setBounds(850,10,50,50);
        home.addActionListener(this);
        add(home);







        setLayout(null);
        setLocationRelativeTo(null);
        setVisible(true);

    }
    public void actionPerformed(ActionEvent ae){
        conn c = new conn();
        String gr = this.gr.getText();
        if(ae.getSource() == display){
            String query = "select gr , name , aadhar , dob , phoneno ,
bloodgrp , address , gender from adddetails where gr = '"+gr+"';";
            try{
                ResultSet rs = c.s.executeQuery(query);
                t1.setModel(DbUtils.resultSetToTableModel(rs));

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

        }
        if(ae.getSource() == update){
            String value = this.data.getText();
            String choice = fields.getSelectedItem().toString();
            String query = "update adddetails set "+choice+" = '"+value+"'
where gr = '"+gr+"';";
            try{
                c.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null,"Data Updated
Successfully!!");
            }
            catch (Exception e){
                System.out.println(e);
                JOptionPane.showMessageDialog(null,"Some Error Occurred!");
            }
        }
        if(ae.getSource() == back){
            new viewDetails();
            this.setVisible(false);
        }
        if(ae.getSource() == home){
            new dashboard();
            this.setVisible(false);
        }
    }

    public static void main(String[] args) {
        new update();
    }
}
package sms;

import net.proteanit.sql.DbUtils;

import javax.swing.*;
import javax.swing.table.JTableHeader;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;

public class delete extends JFrame implements ActionListener {
    JButton delete , display , home , back;
    JLabel l1;
    JTextField gr;
    JScrollPane sp1;
    JTable t1;
    Font f1 = new Font("Tahoma", Font.PLAIN,20);
    Font f = new Font("Tahoma", Font.PLAIN,16);
    Font f2 = new Font("Tahoma", Font.BOLD,24);
    delete(){
```



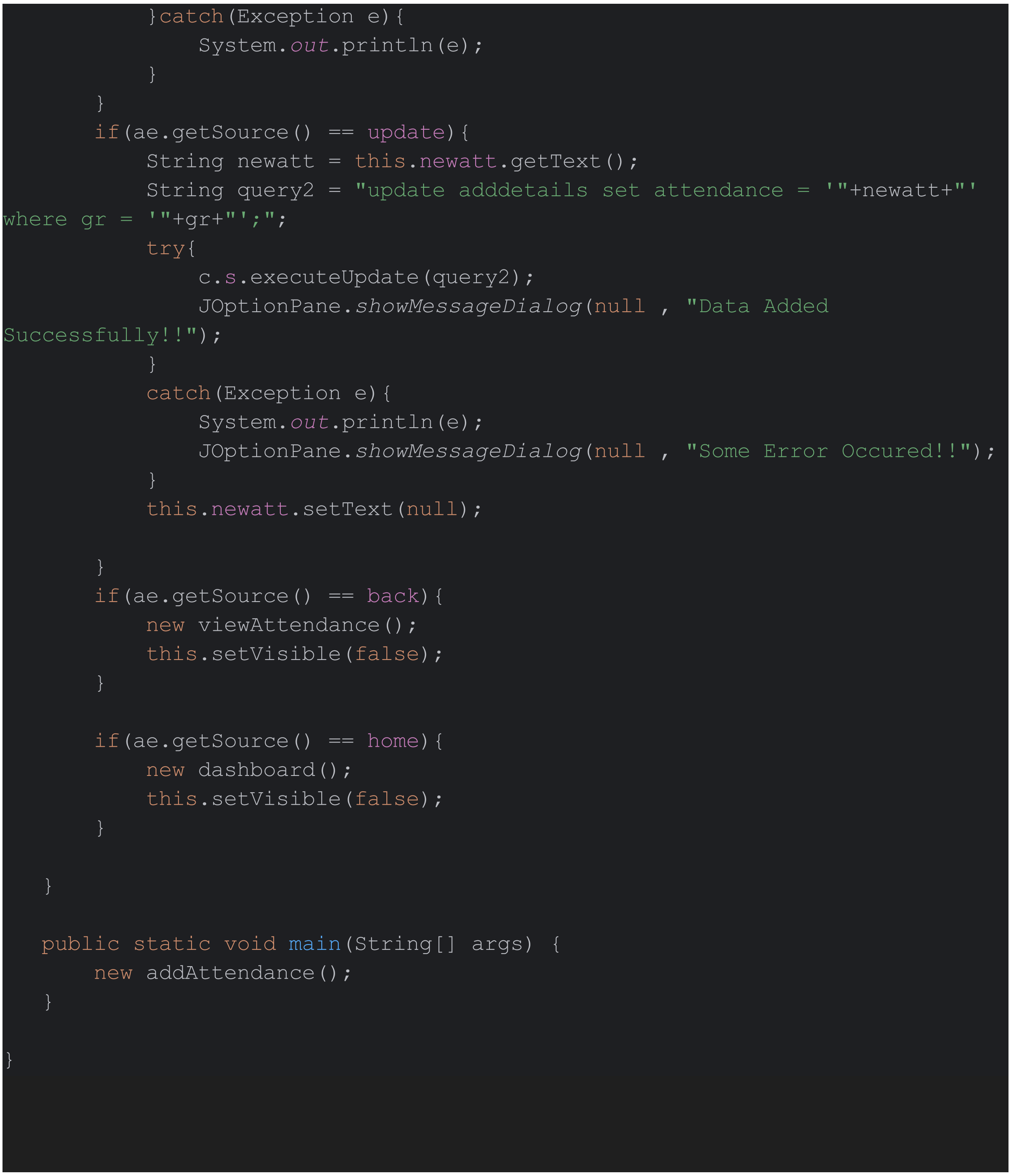
```
        setSize(970,600);

        l1 = new JLabel("Enter GR NO");
        l1.setBounds(150,60,150,50);
        l1.setFont(f2);
        add(l1);

        gr = new JTextField();
        gr.setBounds(320,60,80,50);
        gr.setFont(f1);
        add(gr);

        display = new JButton("Display");
        display.setBounds(450,60,150,50);
        display.setFont(f1);
        display.addActionListener(this);
        add(display);

        sp1 = new JScrollPane();
        sp1.setBounds(30,170,900,64);
        add(sp1);

        t1 = new JTable();
        t1.setFont(f);
        t1.setRowHeight(30);
        sp1.setViewportView(t1);

        JTableHeader tb1 = t1.getTableHeader();
        tb1.setFont(new Font("Tahoma", Font.BOLD, 22));
        tb1.setBackground(Color.cyan);

        delete = new JButton("Delete");
        delete.setBounds(400,480,150,50);
        delete.setFont(f1);
        delete.addActionListener(this);
        add(delete);

        ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("images/home.jpg"));
        Image i5 = i4.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i6 = new ImageIcon(i5);
        home = new JButton(i6);
        home.setBounds(870,10,50,50);
        home.addActionListener(this);
        add(home);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("images/back.png"));
        Image i2 = i1.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i3 = new ImageIcon(i2);
        back = new JButton(i3);
        back.setBounds(10,10,50,50);
        back.addActionListener(this);
        add(back);



        setLayout(null);
        setLocationRelativeTo(null);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent ae) {
        conn c = new conn();
        String gr = this.gr.getText();
        if (ae.getSource() == display) {
            String query = "select gr , name , aadhar , dob , phoneno ,
bloodgrp , address , gender from adddetails where gr = '"+gr+"';";
            try {
                ResultSet rs = c.s.executeQuery(query);
                t1.setModel(DbUtils.resultSetToTableModel(rs));

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

        }
        if(ae.getSource() == delete){
            String query = "delete from adddetails  where gr = '"+gr+"';";
            try {
                c.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null,"Data Deleted
Successfully");

            } catch (Exception e) {
                System.out.println(e);
                JOptionPane.showMessageDialog(null,"Some Error Occured");
            }
        }

        if(ae.getSource() == back){
            new viewDetails();
            this.setVisible(false);
        }
        if(ae.getSource() == home){
            new dashboard();
            this.setVisible(false);
        }

    }

    public static void main(String[] args) {
        new delete();
    }
}
package sms;
```



```java
import net.proteanit.sql.DbUtils;

import javax.print.DocFlavor;
import javax.swing.*;
import javax.swing.table.JTableHeader;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;

public class addAttendance extends JFrame implements ActionListener {
    JLabel l1 , l2;
    JButton display , update , back , home;
    JTextField gr , newatt;
    JTable att;

    JScrollPane sp1;
    Font f1 = new Font("Tahoma", Font.PLAIN,24);
    Font f2 = new Font("Tahoma", Font.PLAIN,20);
    addAttendance(){
        setSize(600,700);

        l1 = new JLabel("Enter GR NO");
        l1.setBounds(140,50,150,50);
        l1.setFont(f1);
        add(l1);

        gr = new JTextField();
        gr.setBounds(300,50,80,50);
        gr.setFont(f2);
        add(gr);

        display =new JButton("Display");
        display.setBounds(400,50,150,50);
        display.setFont(f1);
        display.addActionListener(this);
        add(display);

        sp1 = new JScrollPane();
        sp1.setBounds(50,200,480,64);
        add(sp1);

        att = new JTable();
        att.setFont(f1);
        att.setRowHeight(30);
        sp1.setViewportView(att);

        JTableHeader tb1 = att.getTableHeader();
        tb1.setFont(new Font("Tahoma", Font.BOLD, 22));
        tb1.setBackground(Color.cyan);


        l2 = new JLabel("Add / Update Attendance");
```



```
        l2.setBounds(100,350,250,50);
        l2.setFont(f2);
        add(l2);

        newatt = new JTextField();
        newatt.setBounds(350,350,80,50);
        newatt.setFont(f2);
        add(newatt);

        update = new JButton("Update");
        update.setBounds(220,500,150,50);
        update.setFont(f1);
        update.addActionListener(this);
        add(update);

        ImageIcon i1 = new
ImageIcon(ClassLoader.getSystemResource("images/back.png"));
        Image i2 = i1.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i3 = new ImageIcon(i2);
        back = new JButton(i3);
        back.setBounds(10,10,50,50);
        back.addActionListener(this);
        add(back);

        ImageIcon i4 = new
ImageIcon(ClassLoader.getSystemResource("images/home.jpg"));
        Image i5 = i4.getImage().getScaledInstance(60,50,Image.SCALE_SMOOTH);
        ImageIcon i6 = new ImageIcon(i5);
        home = new JButton(i6);
        home.setBounds(530,600,50,50);
        home.addActionListener(this);
        add(home);






        setLayout(null);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae){
        String gr = this.gr.getText();
        conn c = new conn();
        if(ae.getSource() == display){
            String query1 = "select gr , name , attendance from adddetails
where gr = '"+gr+"'; ";
            try{
                ResultSet rs = c.s.executeQuery(query1);
                att.setModel(DbUtils.resultSetToTableModel(rs));
```



```
            }catch(Exception e){
                System.out.println(e);
            }
        }
        if(ae.getSource() == update){
            String newatt = this.newatt.getText();
            String query2 = "update adddetails set attendance = '"+newatt+"'
where gr = '"+gr+"';";
            try{
                c.s.executeUpdate(query2);
                JOptionPane.showMessageDialog(null , "Data Added
Successfully!!");
            }
            catch(Exception e){
                System.out.println(e);
                JOptionPane.showMessageDialog(null , "Some Error Occured!!");
            }
            this.newatt.setText(null);

        }
        if(ae.getSource() == back){
            new viewAttendance();
            this.setVisible(false);
        }

        if(ae.getSource() == home){
            new dashboard();
            this.setVisible(false);
        }

    }

    public static void main(String[] args) {
        new addAttendance();
    }

}
```

Working :

Certainly, let's outline the general workflow and functionality for your "Student Monitoring System" based on the components you've mentioned: starter window, login window, and option window. This will provide an overview of how these components work together:

Starter Window:

1. The starter window serves as the entry point of your application. It may include branding, a school logo, or other introductory information.

Login Window:

1. Upon launching the application, users are presented with a login window.
2. Teachers will enter their credentials, typically a username and password, to access the system.
3. The system should validate the login credentials against a database of authorized teachers.
4. If the login is successful, the user is granted access to the system; otherwise, they receive an error message.

Option Window:

1. Once logged in, teachers are directed to the option window.
2. The option window typically provides a menu or navigation panel with options such as "Add Details," "Attendance," and "Marks."
3. Teachers can click on these options to access specific functionalities.

Add Details Section:

1. In the "Add Details" section, teachers can input and update student information, including personal details, contact information, and enrollment data.

2. The system should provide user-friendly forms for data entry and validation to ensure accurate information.

3. Teachers can save the data, and it should be stored in the database.

Attendance Section:

1. The "Attendance" section allows teachers to mark student attendance.

2. Teachers can select a date and mark students as present or absent for that date.

3. The system should provide feedback on the successful recording of attendance.

Marks Section:

1. In the "Marks" section, teachers can enter and update subject-wise marks for students.

2. Teachers select the student, subject, and enter the marks for each student.

3. The system should ensure data accuracy and proper storage in the database.

User Access Control:

1. The system should enforce user access control to ensure that teachers can only access features and data relevant to their role.

Error Handling:

1. The system should handle errors gracefully, providing feedback to users in case of login failures, data validation issues, or other errors.

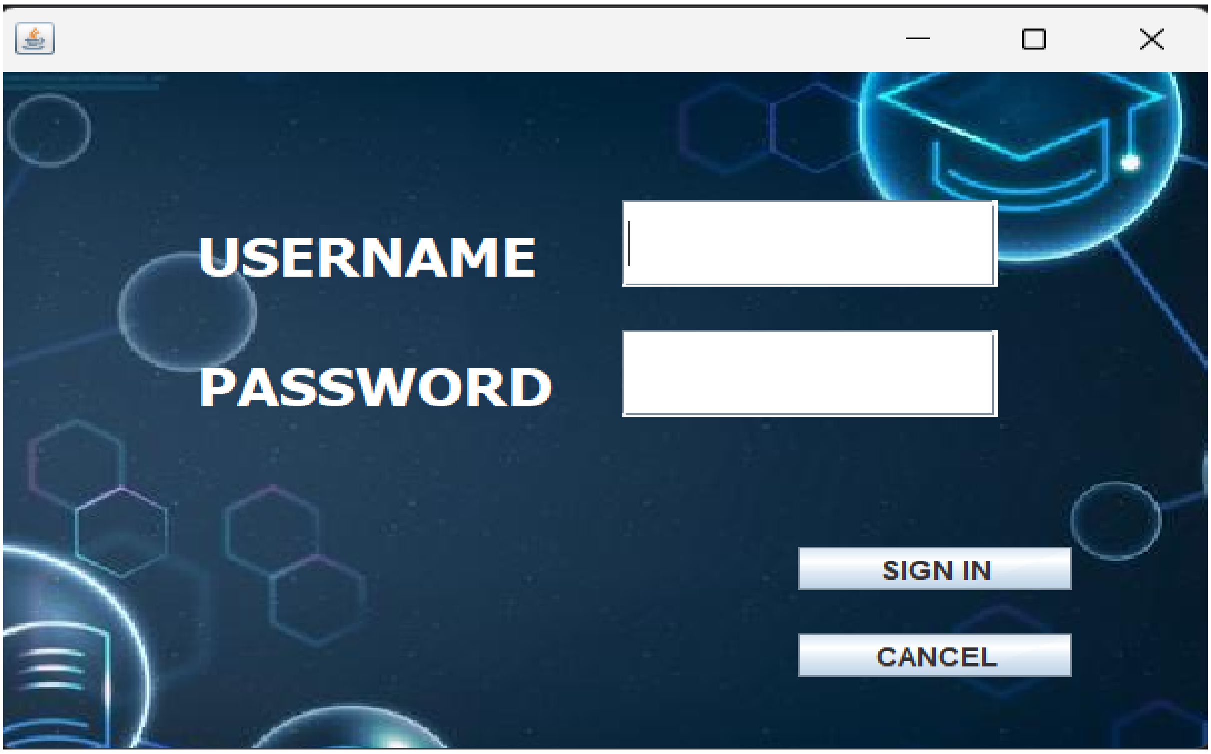
Data Persistence:

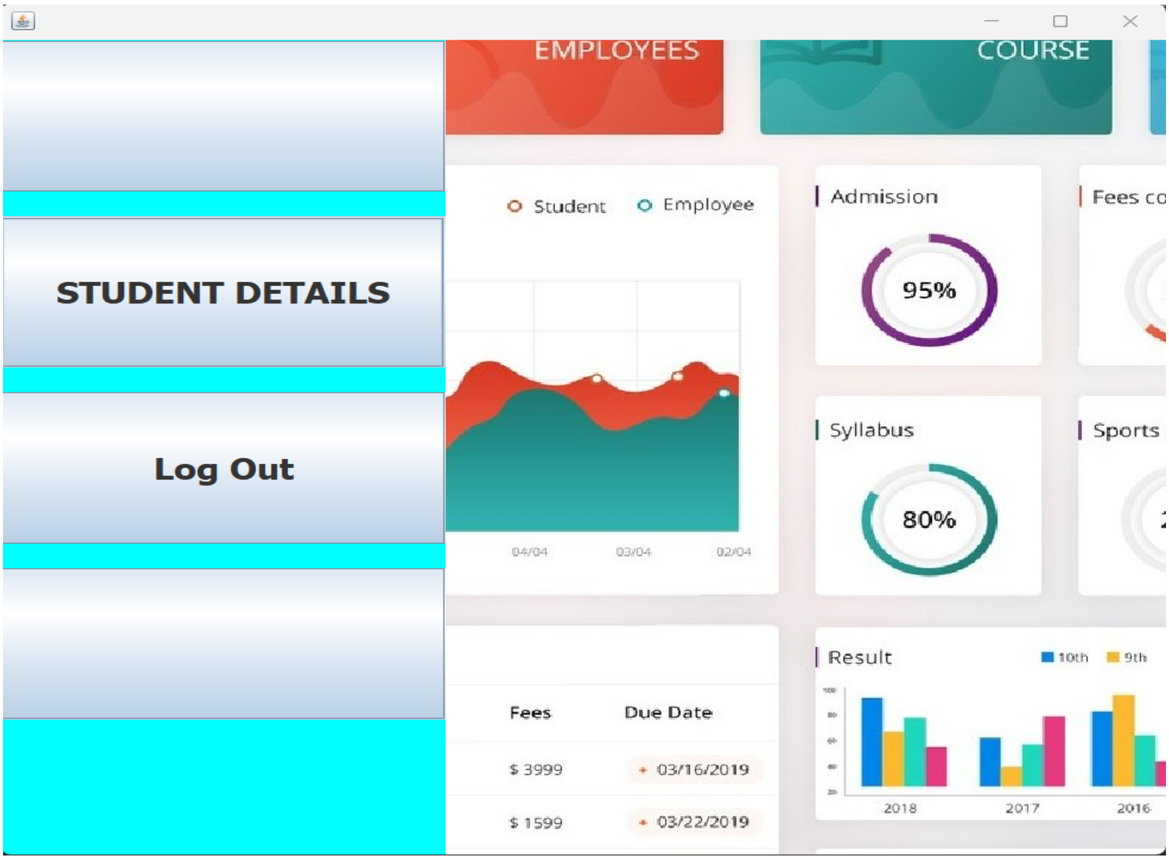
1. All data entered by teachers, including student details, attendance records, and marks, should be stored persistently in a database.

Data Retrieval:

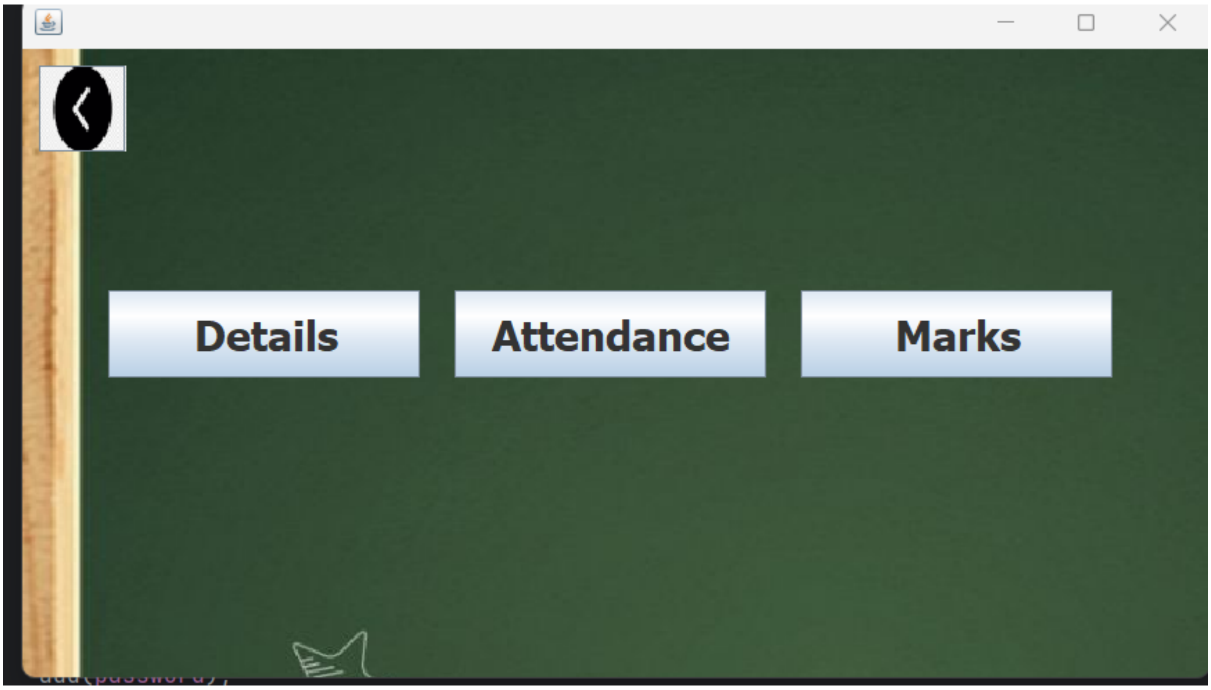
1. Teachers should have the ability to retrieve and view previously entered data, such as student details, attendance history, and subject-wise marks.

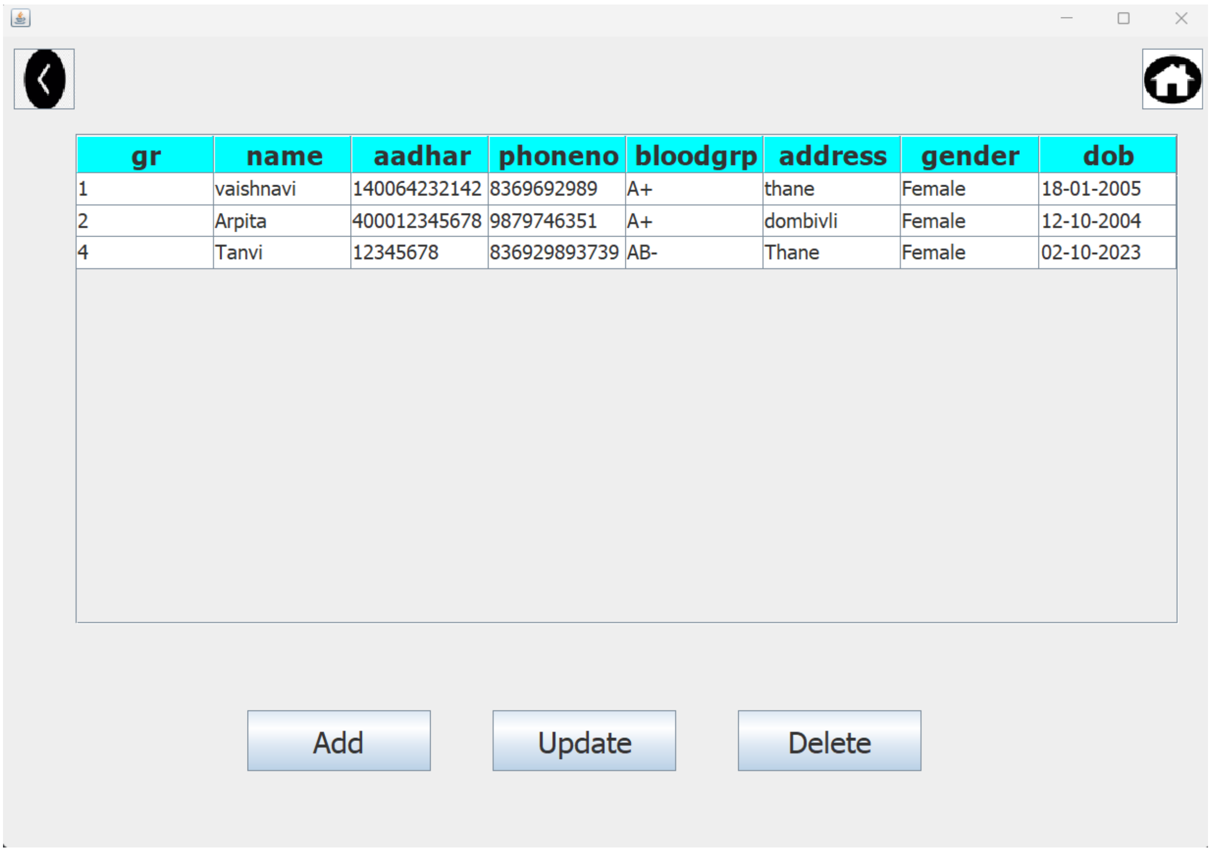
SNAPSHOTS OF OUTPUT :



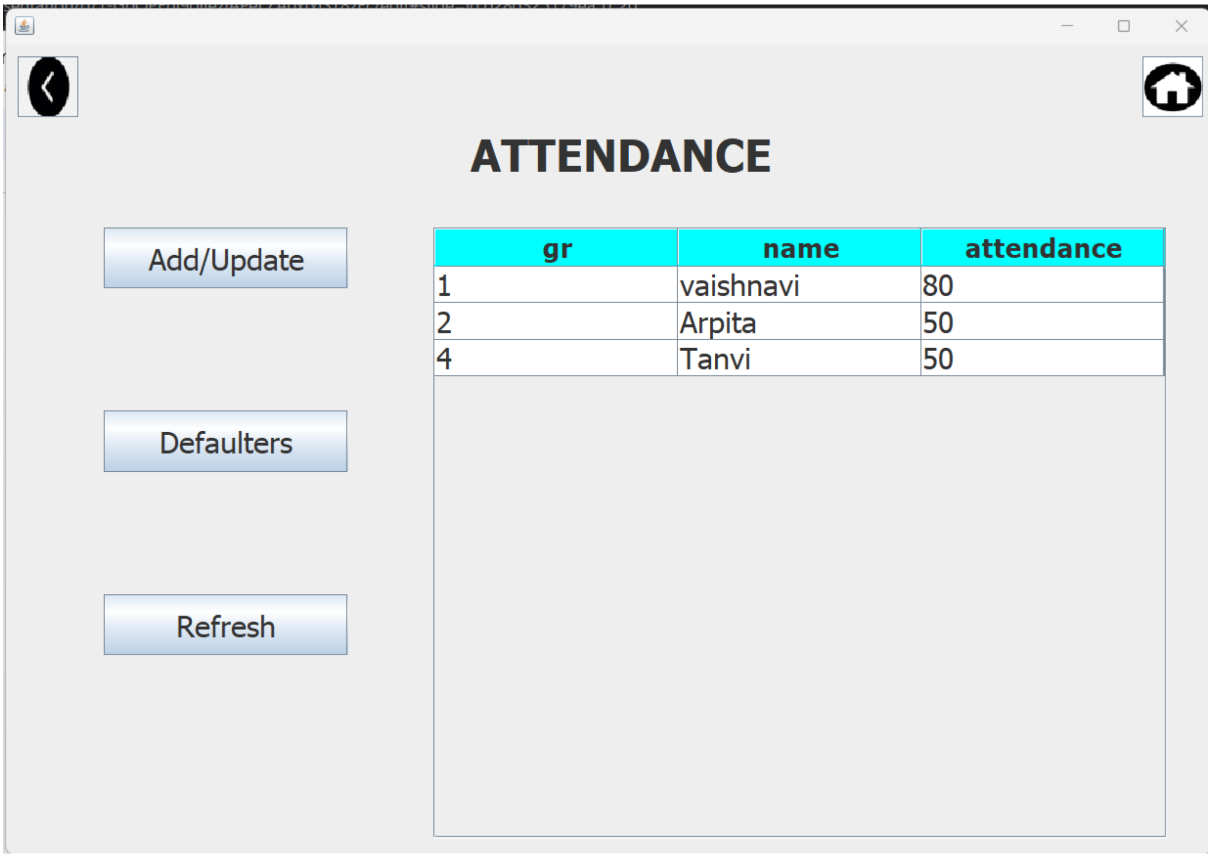


Fees	Due Date
$ 3999	03/16/2019
$ 1599	03/22/2019



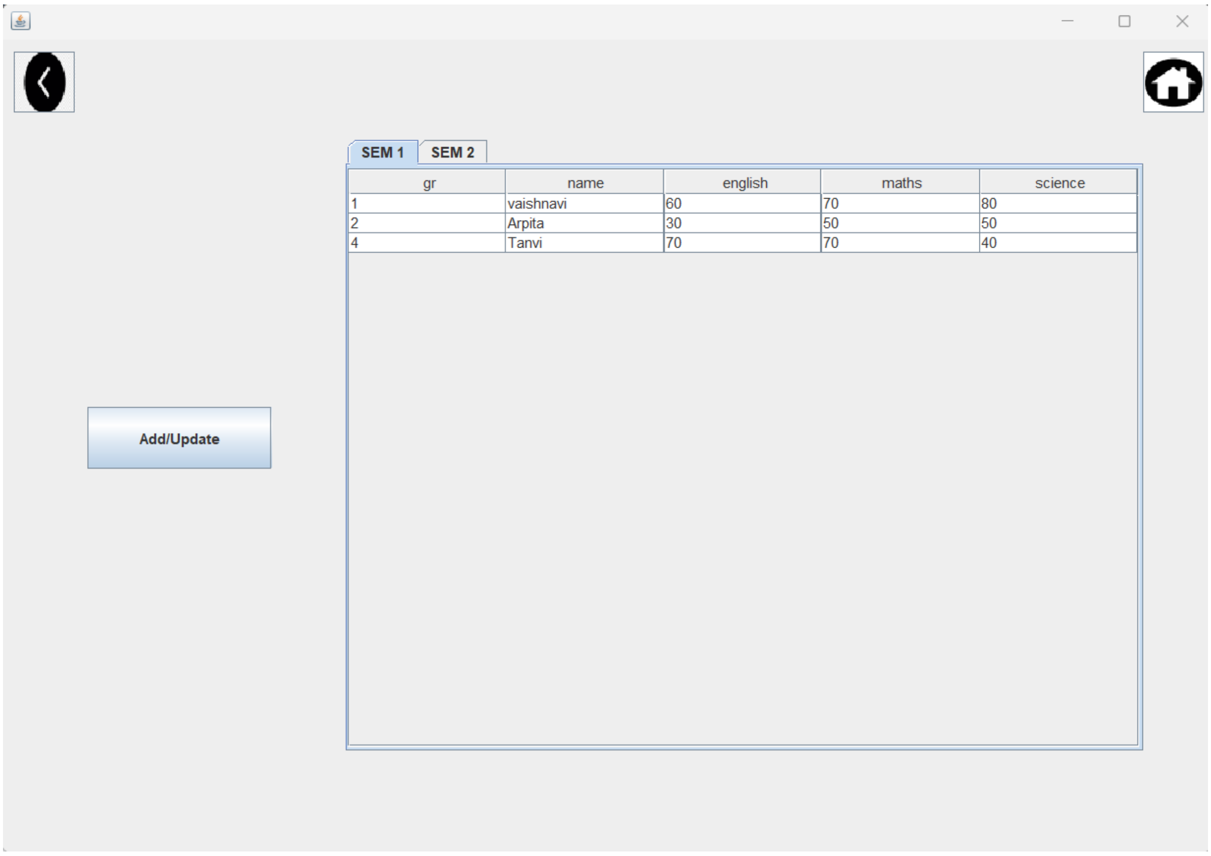


gr	name	aadhar	phoneno	bloodgrp	address	gender	dob
1	vaishnavi	140064232142	8369692989	A+	thane	Female	18-01-2005
2	Arpita	400012345678	9879746351	A+	dombivli	Female	12-10-2004
4	Tanvi	12345678	836929893739	AB-	Thane	Female	02-10-2023



ATTENDANCE

gr	name	attendance
1	vaishnavi	80
2	Arpita	50
4	Tanvi	50



gr	name	english	maths	science
1	vaishnavi	60	70	80
2	Arpita	30	50	50
4	Tanvi	70	70	40

FUTURE SCOPE :

The "Teacher's Student Monitoring System" project offers several avenues for future development and expansion. Here are some potential areas for future scope and enhancement:

1. Student and Parent Portals: Consider creating dedicated portals for students and their parents. Students can view their own academic progress, while parents can keep track of their child's performance and attendance. This enhances communication between all stakeholders.

2. Notification System: Implement a notification system that can send alerts to teachers, students, and parents regarding important updates, such as exam results, attendance issues, and school announcements.

3. Integration with Other Systems: Explore the integration of your system with other educational tools and platforms, such as Learning Management Systems (LMS) or school management software, to create a comprehensive educational ecosystem.

4. Analytics and Insights: Enhance the system by incorporating data analytics and reporting features. Provide teachers with actionable insights into student performance trends and areas that need improvement.

5. Mobile Application: Develop a mobile application version of the system to make it more accessible to teachers who are on the go.

6. Machine Learning and Predictive Analytics: Implement machine learning models to predict student performance, helping teachers and administrators identify at-risk students and intervene early.

7. Scalability: Optimize the system for scalability, ensuring it can handle an increasing number of students, teachers, and data as the educational institution grows.

8. Support for Multiple Schools: Extend the system to support multiple schools or campuses within a district or educational network, offering a centralized platform for data management.

CONCLUSION :

As we conclude this project, we reflect on the transformative impact it can have within educational institutions. It empowers teachers to actively manage and monitor student data, facilitating personalized academic guidance. The ability to record marks, track attendance, and generate insightful reports simplifies the administrative workload, allowing educators to concentrate on what truly matters: teaching and nurturing the next generation.

The "Student Monitoring System" is more than just a software application; it is a pivotal tool that embodies the future of efficient and data-driven education. This project has successfully addressed the complex and dynamic needs of educational institutions by providing teachers with a dedicated and secure platform for student data management.

REFERENCES

https://www.w3schools.com/java/default.asp

https://www.javatpoint.com/java-swing

https://www.slideshare.net/amit_gandhi/student-management-system-35796214