Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2023), B.Sc. in CSE (Day)

Student Management System

Project Report

Course Title: Object Oriented Programming Lab
Course Code: CSE-202 Section:
DH

Students Details

Name	ID
Md. Mahim Hossain	221902082

Submission Date: 24 June 2023
Course Teacher's Name: Sagufta Sabah Nakshi

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	Signature:
Comments:	Date:

Contents

Chapter 1

Introduction

1.1 Overview

Object-oriented programming (OOP) is motivated in the development of a student management system to achieve modularity, code reusability, encapsulation, flexibility, extensibility, and maintainability. OOP allows the system to be divided into independent modules, promotes the reuse of code, encapsulates data and functionality within classes, enables customized behavior through polymorphism, facilitates the extension of functionality through inheritance, and simplifies maintenance and updates. By leveraging OOP principles, the student management system can be efficiently designed, implemented, and adapted to meet the specific requirements of educational institutions.

1.2 Motivation

The Student Management System, powered by Java, is your ultimate solution for efficient HR operations. With our comprehensive system, you can streamline employee data management, automate mundane tasks, and enhance overall efficiency. From seamless attendance tracking to accurate payroll calculations, our project simplifies complex processes, saving valuable time and minimizing errors. Gain valuable insights with insightful analytics, enabling data-driven decision-making and strategic resource allocation. Foster a harmonious work environment with transparent communication channels, feedback mechanisms, and performance evaluation tools. Empower your HR team to focus on talent development and organizational growth while ensuring compliance and optimizing your workforce potential. Experience the power of our Java-based Student Management System today [?].

1.3 Problem Definition

The problem addressed by a student management system using object-oriented programming is the need for efficient management of student-related information and processes in educational institutions. Object-oriented programming enables the development of a modular, flexible, and maintainable system that can streamline tasks such as enrollment, attendance tracking, academic records management, communication, and reporting. By leveraging object-oriented principles, the system aims to optimize resource allocation, improve communication, enhance data management, and ultimately enhance overall educational management.

1.4 Objectives

The objective of this work is to give a complete approach to personnel information management. This will be accomplished by developing and deploying an HR management system that will result in a significant shift in the way employee data is managed. This system's objectives include the following:

1. Design of an HR management system to meet needs such as adding and deleting student, viewing and printing student data, and updating student information.
2. Student data is stored in a well-designed database.
3. An easy-to-use interface that will let the user interact with the system.

1.5 Application

Using object-oriented programming (OOP) in a student management system project provides benefits such as modular design, code reusability, encapsulation, inheritance, polymorphism, and flexibility. OOP allows for creating separate modules for different functionalities, promotes code reuse, protects data integrity, enables customization through subclassing, supports flexible behavior, and ensures scalability as the system evolves. Overall, OOP facilitates the development of a robust and adaptable student management system that efficiently handles student-related processes in educational institutions.

Chapter 2

Implementation and Result

2.1 Introduction

The implementation of the Student Management System will be developed using Java is a versatile and widely-used programming language. Java offers numerously advanced tags, such as its object-oriented nature, platform independence, and vast ecosystem of libraries and frameworks.

2.2 Implementation

Following is the Java Class of the Student Management System where we can see all the code of this system in use and you can also see the Database Syntax of the system.

2.2.1 Splash Class

```
package student.management.system;


import javax.swing.*;


import java.awt.*;


import java.awt.event.*;


public class Splash extends JFrame implements ActionListener {

    Splash() {

        getContentPane().setBackground(Color.BLACK);

        setLayout(null);

        JLabel heading = new JLabel("STUDENT MANAGEMENT SYSTEM");
        heading.setBounds(80, 30, 1200, 60);

        heading.setFont(new Font("serif", Font.PLAIN, 60));
        heading.setForeground(Color.GREEN);

        add(heading);

         ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/front.jpg Image i2 =
                  i1.getImage().getScaledInstance(1100, 700, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);

        JLabel image = new JLabel(i3);
        image.setBounds(50, 100, 1050, 500);
        add(image);
```

```
            JButton clickhere = new JButton("CLICK HERE TO CONTINUE");
            clickhere.setBounds(400, 400, 300, 70);
            clickhere.setBackground(Color.BLACK);
            clickhere.setForeground(Color.WHITE);
            clickhere.addActionListener(this);


            image.add(clickhere);


            setSize(1170, 650);

            setLocation(200, 50);

            setVisible(true);

            while(true) { heading.setVisible(false);

            try {
                        Thread.sleep(500); }
                  catch (Exception e){

                  }

                  heading.setVisible(true); try {
                        Thread.sleep(500);

                         } catch

                        (Exception e){

                  }
            }

  }

       public void actionPerformed(ActionEvent ae) {
            setVisible(false);
       new Login();
       } public static void main(String args[]) { new

       Splash();

       }
}
```

2.2.2 Login Class

```
package student.management.system;


import javax.swing.*;


import java.awt.*;


import java.awt.event.*;
import java.sql.*;


public class Login extends JFrame
implements ActionListener{

        JTextField tfusername, tfpassword;

        Login() {

                getContentPane().setBackground(Color.WHITE);
                setLayout(null);

                JLabel lblusername = new JLabel("Username");
                lblusername.setBounds(40, 20, 100, 30);
                add(lblusername);

                tfusername = new JTextField();
                tfusername.setBounds(150, 20, 150, 30);
                add(tfusername);

                JLabel lblpassword = new JLabel("Password");
                lblpassword.setBounds(40, 70, 100, 30);
                add(lblpassword);

                tfpassword = new JTextField();
                tfpassword.setBounds(150, 70, 150, 30);
                add(tfpassword);

                JButton login = new JButton("LOGIN");
                login.setBounds(150, 140, 150, 30);
                login.setBackground(Color.BLACK);
                login.setForeground(Color.WHITE);
                login.addActionListener(this);
        add(login);

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/second.jp
        Image i2 =
        i1.getImage().getScaledInstance(
        200, 200,
        Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);

        JLabel image = new JLabel(i3);
        image.setBounds(350, 0, 200, 200);

        add(image);

        setSize(600, 300);
        setLocation(450, 200);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae)

    { try {
            String username = tfusername.getText();
            String password = tfpassword.getText();

            Conn c = new Conn();
            String query = "select * from login where username = ’"+username+"’ and

            ResultSet rs = c.s.executeQuery(query);

            if (rs.next()) {

                setVisible(false);
                new Home();
            } else {
                JOptionPane.showMessageDialog(null, "Invalid username or password")
                setVisible(false);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } }

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

2.2.3 Home Class

```
package student.management.system;


import javax.swing.*;


import java.awt.*;


import java.awt.event.*;


public class Home extends JFrame implements ActionListener{


    JButton view, add, update, remove;

    Home() { setLayout(null);

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/home.jpg" Image i2 =
                i1.getImage().getScaledInstance(1120, 630, Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);

        JLabel image = new JLabel(i3);

        image.setBounds(0, 0, 1120, 630);

        add(image);

        JLabel heading = new JLabel("Student Management System");
        heading.setBounds(620, 20, 400, 40);
        heading.setFont(new Font("Raleway", Font.BOLD, 25));
        image.add(heading);

        add = new JButton("Add Student");
        add.setBounds(650, 80, 150, 40);
        add.addActionListener(this);

        image.add(add);

        view = new JButton("View Student");
        view.setBounds(820, 80, 150, 40);
        view.addActionListener(this);
        image.add(view);

        update = new JButton("Update Student");
        update.setBounds(650, 140, 150, 40);
        update.addActionListener(this);

        image.add(update);

        remove = new JButton("Remove Student");
        remove.setBounds(820, 140, 150, 40);
        remove.addActionListener(this);

        image.add(remove);

        setSize(1120, 630);
        setLocation(250, 100);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) { if
        (ae.getSource() == add) { setVisible(false); new
        AddStudent();
        } else if (ae.getSource() == view) {
            setVisible(false);
        new ViewStudent();
        } else if (ae.getSource() == update) {
            setVisible(false);
        new ViewStudent();
        } else {
            setVisible(false);
            new RemoveStudent();
        } }


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

2.2.4 AddStudent Class

```
package student.management.system;
```

```
import java.awt.*;


import javax.swing.*;


import
com.toedter.calendar.JDateCh
ooser;


import java.util.*;


import java.awt.event.*;


public class AddStudent extends JFrame implements ActionListener{

    Random ran = new Random();

    int number = ran.nextInt(999999);

    JTextField tfname, tffname, tfaddress, tfphone, tfpostcode, tfemail, tfreligion
    JDateChooser dcdob;
    JComboBox cbbloodgroup;
    JLabel lblstudentId;
    JButton add, back;

    AddStudent() { getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel heading = new JLabel("Add Student Details");
        heading.setBounds(320, 30, 500, 50);

        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));
        add(heading);

        JLabel labelname = new JLabel("Name");

        labelname.setBounds(50, 150, 150, 30);

        labelname.setFont(new Font("serif", Font.PLAIN, 20));

        add(labelname);

        tfname = new JTextField();
```

```
tfname.setBounds(200, 150, 150, 30);

add(tfname);

JLabel labelfname = new JLabel("Father’s Name");
labelfname.setBounds(400, 150, 150, 30);
labelfname.setFont(new Font("serif", Font.PLAIN, 20));

add(labelfname);

tffname = new JTextField();

tffname.setBounds(600, 150, 150, 30);

add(tffname);

JLabel labeldob = new JLabel("Date of Birth");

 labeldob.setBounds(50, 200, 150, 30);

labeldob.setFont(new Font("serif", Font.PLAIN, 20));

add(labeldob);

dcdob = new JDateChooser();

dcdob.setBounds(200, 200, 150, 30);

add(dcdob);

JLabel labelreligion = new JLabel("Religion");

 labelreligion.setBounds(400, 200, 150, 30);

labelreligion.setFont(new Font("serif", Font.PLAIN, 20));

add(labelreligion);

tfreligion = new JTextField();

tfreligion.setBounds(600, 200, 150, 30);

add(tfreligion);

JLabel labeladdress = new JLabel("Address");

labeladdress.setBounds(50, 250, 150, 30);
```

```
 labeladdress.setFont(new Font("serif", Font.PLAIN, 20));
add(labeladdress);

tfaddress = new JTextField();

tfaddress.setBounds(200, 250, 150, 30);

add(tfaddress);

JLabel labelphone = new JLabel("Phone");

labelphone.setBounds(400, 250, 150, 30);

labelphone.setFont(new Font("serif", Font.PLAIN, 20));

add(labelphone);

tfphone = new JTextField();

tfphone.setBounds(600, 250, 150, 30);

add(tfphone);

JLabel labelemail = new JLabel("Email");

labelemail.setBounds(50, 300, 150, 30);

labelemail.setFont(new Font("serif", Font.PLAIN, 20));

add(labelemail);

tfemail = new JTextField();

tfemail.setBounds(200, 300, 150, 30);

add(tfemail);

JLabel labelbloodgroup = new JLabel("Blood Group");
labelbloodgroup.setBounds(400, 300, 150, 30);
labelbloodgroup.setFont(new Font("serif", Font.PLAIN, 20));
add(labelbloodgroup);

String courses[] = {"A+", "O+",
"B+", "AB+", "A-", "O-", "B-",
"AB-"};
```

```
cbbloodgroup = new
JComboBox(courses);
cbbloodgroup.setBackground(Co
lor.WHITE);

cbbloodgroup.setBounds(600, 300, 150, 30);

add(cbbloodgroup);

JLabel labelnationality = new JLabel("Nationality");

labelnationality.setBounds(50, 350, 150, 30);

labelnationality.setFont(new Font("serif", Font.PLAIN, 20));

add(labelnationality);

tfnationality = new JTextField();

tfnationality.setBounds(200, 350, 150, 30);

add(tfnationality);

JLabel labelaadhar = new JLabel("Post Code");

 labelaadhar.setBounds(400, 350, 150, 30);

labelaadhar.setFont(new Font("serif", Font.PLAIN, 20));

add(labelaadhar);

tfpostcode = new JTextField();

tfpostcode.setBounds(600, 350, 150, 30);

add(tfpostcode);

JLabel labelempId = new JLabel("Student id");

 labelempId.setBounds(50, 400, 150, 30);

 labelempId.setFont(new Font("serif", Font.PLAIN, 20));

add(labelempId);

lblstudentId = new JLabel("" + number);

lblstudentId.setBounds(200, 400, 150, 30);
```

```
        lblstudentId.setFont(new Font("serif", Font.PLAIN, 20));

        add(lblstudentId);

        add = new JButton("Add Details");
        add.setBounds(250, 550, 150, 40);
        add.addActionListener(this);
        add.setBackground(Color.BLACK);
        add.setForeground(Color.WHITE); add(add);

        back = new JButton("Back");

        back.setBounds(450, 550, 150, 40);
        back.addActionListener(this);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);

        add(back);

        setSize(900, 700);

        setLocation(300, 50);

        setVisible(true);

    }

    public void actionPerformed(ActionEvent ae) { if
        (ae.getSource() == add) {
            String name = tfname.getText();
            String fname = tffname.getText();
            String dob = ((JTextField) dcdob.getDateEditor().getUiComponent()).getT
            String religion = tfreligion.getText();
            String address = tfaddress.getText();
            String phone = tfphone.getText();
            String email = tfemail.getText();
            String bloodgroup = (String)
            cbbloodgroup.getSelItem();
            String nationality = tfnationality.getText();
            String postcode = tfpostcode.getText();

            String studentId = lblstudentId.getText();

            try {
                Conn conn = new Conn();
```

```
                    String query = "insert into student values('"+name+"', '"+fname+"',
                    conn.s.executeUpdate(query);
                    JOptionPane.showMessageDialog(null, "Details added successfully");
                    setVisible(false);
                    new Home();
                } catch (Exception e) {
                    e.printStackTrace();
                }
            } else {
                setVisible(false); new
                Home();
            } }


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

2.2.5 ViewStudent Class

```
  package student.management.system;


import javax.swing.*;


import java.awt.*;


import java.sql.*;


import net.proteanit.sql.DbUtils;


import java.awt.event.*;


public class ViewStudent extends JFrame implements ActionListener{


    JTable table;
```

```
Choice cstudentId;
JButton search, print, update, back;

ViewStudent() {

        getContentPane().setBackground(Color.WHITE);

        setLayout(null);

        JLabel searchlbl = new JLabel("Search by Student Id");
        searchlbl.setBounds(20, 20, 150, 20);

        add(searchlbl);

        cstudentId = new Choice();

        cstudentId.setBounds(180, 20, 150, 20);

        add(cstudentId);

        try {
                Conn c = new Conn();
                ResultSet rs = c.s.executeQuery("select * from student");
                while(rs.next()) { cstudentId.add(rs.getString("studentId")); }
        } catch (Exception e) {
                e.printStackTrace(); }
        table = new JTable();

        try {
                Conn c = new Conn();
                ResultSet rs =
                c.s.executeQuery("select *
                from student");
                table.setModel(DbUtils.res
                ultSetToTableModel(rs));
        } catch (Exception e) {
                e.printStackTrace();
        }

        JScrollPane jsp = new JScrollPane(table);
        jsp.setBounds(0, 100, 900, 600); add(jsp);

        search = new JButton("Search");

        search.setBounds(20, 70, 80, 20);
        search.addActionListener(this);
        add(search);

        print = new JButton("Print");

        print.setBounds(120, 70, 80, 20);
        print.addActionListener(this);

        add(print);

        update = new JButton("Update");
        update.setBounds(220, 70, 80, 20);
        update.addActionListener(this);

        add(update);

        back = new JButton("Back");
        back.setBounds(320, 70, 80, 20);
        back.addActionListener(this);

        add(back);

        setSize(900, 700);
        setLocation(300, 100);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) { if
        (ae.getSource() == search) {
            String query = "select * from student where studentId = ’"+cstudentId.g try {
                Conn c = new Conn();
                ResultSet rs = c.s.executeQuery(query);
                table.setModel(DbUtils.resultSetToTableModel(rs));
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else if (ae.getSource() == print) { try {
                table.print();
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else if (ae.getSource() == update) { setVisible(false);
           new UpdateStudent(cstudentId.getSelectedItem());
        } else {
            setVisible(false);

            new Home();
            }


    }


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

2.2.6 UpdateStudent Class

```
package student.management.system;

import java.awt.*;

 import javax.swing.*;

import java.awt.event.*;

import java.sql.*;

public class UpdateStudent extends JFrame implements ActionListener{

    JTextField tfbloodgroup, tffname, tfaddress, tfphone, tfpostcode, tfemail, tfre
    JLabel lblstudentId;
    JButton add, back;
    String studentId;

    UpdateStudent(String studentId) { this.studentId =
        studentId;
        getContentPane().setBackground(Color.WHITE);

        setLayout(null);

        JLabel heading = new JLabel("Update Student Details");
        heading.setBounds(320, 30, 500, 50);
        heading.setFont(new Font("SAN_SERIF", Font.BOLD, 25));

        add(heading);

        JLabel labelname = new JLabel("Name");
        labelname.setBounds(50, 150, 150, 30);
labelname.setFont(new Font("serif", Font.PLAIN, 20));

add(labelname);

JLabel lblname = new JLabel();

lblname.setBounds(200, 150, 150, 30);

add(lblname);

JLabel labelfname = new JLabel("Father's Name");

labelfname.setBounds(400, 150, 150, 30);

labelfname.setFont(new Font("serif", Font.PLAIN, 20));

add(labelfname);

tffname = new JTextField();

tffname.setBounds(600, 150, 150, 30);

add(tffname);

JLabel labeldob = new JLabel("Date of Birth");

labeldob.setBounds(50, 200, 150, 30);

labeldob.setFont(new Font("serif", Font.PLAIN, 20));

add(labeldob);

JLabel lbldob = new JLabel();

lbldob.setBounds(200, 200, 150, 30);

add(lbldob);

JLabel labelreligion = new JLabel("Religion");

labelreligion.setBounds(400, 200, 150, 30);

labelreligion.setFont(new Font("serif", Font.PLAIN, 20));

add(labelreligion);

tfreligion = new JTextField();
```

```
tfreligion.setBounds(600, 200, 150, 30);

add(tfreligion);

JLabel labeladdress = new JLabel("Address");

labeladdress.setBounds(50, 250, 150, 30);

labeladdress.setFont(new Font("serif", Font.PLAIN, 20));
add(labeladdress);

tfaddress = new JTextField();

 tfaddress.setBounds(200, 250, 150, 30);

add(tfaddress);

JLabel labelphone = new JLabel("Phone");

labelphone.setBounds(400, 250, 150, 30);

labelphone.setFont(new Font("serif", Font.PLAIN, 20));

add(labelphone);

tfphone = new JTextField();

tfphone.setBounds(600, 250, 150, 30);

add(tfphone);

JLabel labelemail = new JLabel("Email");

labelemail.setBounds(50, 300, 150, 30);

 labelemail.setFont(new Font("serif", Font.PLAIN, 20));

add(labelemail);

tfemail = new JTextField();

 tfemail.setBounds(200, 300, 150, 30);

add(tfemail);

JLabel labelbloodgroup = new JLabel("Blood Group");
labelbloodgroup.setBounds(400, 300, 150, 30);
labelbloodgroup.setFont(new Font("serif", Font.PLAIN, 20));
add(labelbloodgroup);

tfbloodgroup = new JTextField();

tfbloodgroup.setBounds(600, 300, 150, 30);

add(tfbloodgroup);

JLabel labelnationality = new JLabel("Nationality");

labelnationality.setBounds(50, 350, 150, 30);

labelnationality.setFont(new Font("serif", Font.PLAIN, 20));

add(labelnationality);

tfnationality = new JTextField();

tfnationality.setBounds(200, 350, 150, 30);

add(tfnationality);

JLabel labelpostcode = new JLabel("Post Code");
labelpostcode.setBounds(400, 350, 150, 30);

labelpostcode.setFont(new Font("serif", Font.PLAIN, 20));
add(labelpostcode);

JLabel lblpostcode = new JLabel();

lblpostcode.setBounds(600, 350, 150, 30);

add(lblpostcode);

JLabel labelstudentId = new JLabel("Student id");
labelstudentId.setBounds(50, 400, 150, 30);

 labelstudentId.setFont(new Font("serif", Font.PLAIN, 20));
add(labelstudentId);

lblstudentId = new JLabel();

lblstudentId.setBounds(200, 400, 150, 30);

lblstudentId.setFont(new Font("serif", Font.PLAIN, 20));
add(lblstudentId);
```

```
        try {
                Conn c = new Conn();
                String query = "select * from student where studentId = ’"+studentId+"’ ResultSet rs =
                c.s.executeQuery(query);

                while(rs.next()) { lblname.setText(rs.getString("name"));
                tffname.setText(rs.getString("fname"));

                lbldob.setText(rs.getString("dob"));

                tfaddress.setText(rs.getString("address"));

                tfreligion.setText(rs.getString("religion"));

                tfphone.setText(rs.getString("phone"));

                tfemail.setText(rs.getString("email"));

                tfbloodgroup.setText(rs.getString("bloodgroup"));
                lblpostcode.setText(rs.getString("postcode"));
                lblstudentId.setText(rs.getString("studentId"));
                tfnationality.setText(rs.getString("nationality"));

                }
        } catch (Exception e) {
                e.printStackTrace(); }

        add = new JButton("Update Details");
        add.setBounds(250, 550, 150, 40);
        add.addActionListener(this);
        add.setBackground(Color.BLACK);
        add.setForeground(Color.WHITE);

        add(add);

        back = new JButton("Back");

         back.setBounds(450, 550, 150, 40);
        back.addActionListener(this);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE); add(back);

        setSize(900, 700);
        setLocation(300, 50);
        setVisible(true);
}
```

```
        public void actionPerformed(ActionEvent ae) { if
                (ae.getSource() == add) {
                        String fname = tffname.getText();
                        String religion = tfreligion.getText();
                        String address = tfaddress.getText();
                        String phone = tfphone.getText();
                        String email = tfemail.getText();
                        String bloodgroup = tfbloodgroup.getText();

                        String nationality = tfnationality.getText();

                        try {
                                Conn conn = new Conn();
          String query = "update student set fname = ’"+fname+"’, religion = conn.s.executeUpdate(query);
                                JOptionPane.showMessageDialog(null, "Details updated successfully")
                                setVisible(false);
                                new Home();
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                } else {
                        setVisible(false);
                        new Home();
                } }

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


                try {
                        Conn c = new Conn();
                        String query = "select * from employee where studentId = ’"+studentId+"’";
                        ResultSet rs = c.s.executeQuery(query);

                        while(rs.next()) { lblname.setText(rs.getString("name"));
                        tffname.setText(rs.getString("fname"));

                        lbldob.setText(rs.getString("dob"));

                        tfaddress.setText(rs.getString("address"));

                         tfsalary.setText(rs.getString("salary"));

                         tfphone.setText(rs.getString("phone"));
                tfemail.setText(rs.getString("email"));
                tfeducation.setText(rs.getString("education"));

                lblnid.setText(rs.getString("nid"));

                lblempId.setText(rs.getString("studentId"));
                tfdesignation.setText(rs.getString("nationality"));

                }
        } catch (Exception e) {
                e.printStackTrace();

                }

        add = new JButton("Update Details");
        add.setBounds(250, 550, 150, 40);
        add.addActionListener(this);
        add.setBackground(Color.BLACK);
        add.setForeground(Color.WHITE);

        add(add);

        back = new JButton("Back");

        back.setBounds(450, 550, 150, 40);
        back.addActionListener(this);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);

        add(back);

        setSize(900, 700);
        setLocation(300, 50);
        setVisible(true);
}

public void actionPerformed(ActionEvent ae) { if
        (ae.getSource() == add) {
                String fname = tffname.getText();
                String salary = tfsalary.getText();
                String address = tfaddress.getText();
                String phone = tfphone.getText();
                String email = tfemail.getText();
                String education = tfeducation.getText();

                String designation = tfnationality.getText();
                try {
                        Conn conn = new Conn();
                        String query = "update employee set fname =
                        ’"+fname+"’,salary = ’"+salary+"’, address = ’"+address+"’, phone =
                        ’"+phone+"’, email = ’"+email+"’, education = ’"+education+"’,
                        Nationatility= ’"+nationality +"’ where studentId = ’"+studentId+"’";
                        conn.s.executeUpdate(query);
                         JOptionPane.showMessageDialog(null, "Details updated successfully")
                        setVisible(false);
                        new Home();
                } catch (Exception e) {
                        e.printStackTrace();
                }
        } else {
                setVisible(false);
                new Home();
        } }

    public static void main(String[] args) { new
        UpdateEmployee("");
    }
}
```

2.2.7 RemoveStudent Class

```
package student.management.system;


import javax.swing.*;


import java.awt.*;


import java.sql.*;


import java.awt.event.*;


public class RemoveStudent extends
JFrame implements ActionListener {

    Choice cStudentId;
```

```
JButton delete, back;

RemoveStudent() { getContentPane().setBackground(Color.WHITE);
        setLayout(null);

        JLabel labelstudentId = new JLabel("Student Id");
        labelstudentId.setBounds(50, 50, 100, 30);

        add(labelstudentId);

        cStudentId = new Choice();

        cStudentId.setBounds(200, 50, 150, 30);

        add(cStudentId);

        try {
                Conn c = new Conn();
                String query = "select * from student";
                ResultSet rs = c.s.executeQuery(query);
                while(rs.next()) { cStudentId.add(rs.getString("studentId"));
                }
        } catch (Exception e) {
                e.printStackTrace();
        }

        JLabel labelname = new JLabel("Name");

        labelname.setBounds(50, 100, 100, 30);

         add(labelname);

        JLabel lblname = new JLabel();

        lblname.setBounds(200, 100, 100, 30);

        add(lblname);

        JLabel labelphone = new JLabel("Phone");

        labelphone.setBounds(50, 150, 100, 30);

        add(labelphone);

        JLabel lblphone = new JLabel();

        lblphone.setBounds(200, 150, 100, 30);
```

```
add(lblphone);

JLabel labelemail = new JLabel("Email");

labelemail.setBounds(50, 200, 100, 30);

add(labelemail);

JLabel lblemail = new JLabel();

lblemail.setBounds(200, 200, 100, 30);

add(lblemail);

try {
        Conn c = new Conn();
        String query = "select * from student where studentId = ’"+cStudentId.g ResultSet rs =
        c.s.executeQuery(query);
        while(rs.next()) { lblname.setText(rs.getString("name"));
        lblphone.setText(rs.getString("phone"));
        lblemail.setText(rs.getString("email")); }
} catch (Exception e) {
        e.printStackTrace();
}

cStudentId.addItemListener(new ItemListener() { public void
        itemStateChanged(ItemEvent ie) { try {
                        Conn c = new Conn();
                        String query = "select * from student where studentId = ’"+cStu ResultSet rs =
                        c.s.executeQuery(query);
                        while(rs.next()) { lblname.setText(rs.getString("name"));
                        lblphone.setText(rs.getString("phone"));
                        lblemail.setText(rs.getString("email")); }
                } catch (Exception e) {
                        e.printStackTrace();
                }
        } });

delete = new JButton("Delete");

delete.setBounds(80, 300, 100,30);
delete.setBackground(Color.BLACK);
delete.setForeground(Color.WHITE);
delete.addActionListener(this);

add(delete);
```

```
        back = new JButton("Back");
        back.setBounds(220, 300, 100,30);
        back.setBackground(Color.BLACK);
        back.setForeground(Color.WHITE);
        back.addActionListener(this); add(back);

        ImageIcon i1 = new ImageIcon(ClassLoader.getSystemResource("icons/delete.jp
        Image i2 =
        i1.getImage().getScaledInstance(
        600, 400,
        Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);

        JLabel image = new JLabel(i3);

         image.setBounds(350, 0, 600, 400);

        add(image);

        setSize(1000, 400);
        setLocation(300, 150);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) { if
        (ae.getSource() == delete) { try {
                Conn c = new Conn();
                            String query = "delete from student where studentId = '"+cStudentI
                c.s.executeUpdate(query);
                JOptionPane.showMessageDialog(null, "Student Information Deleted Su
                setVisible(false);
                new Home();
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            setVisible(false);
            new Home();
        } }


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

2.2.8 Conn Class

```
package student.management.system;

import java.sql.*;

public class Conn {

        Connection c;

        Statement s;

        public Conn () { try {
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        c = DriverManager.getConnection("jdbc:mysql:///stu4", "root", "MyselfMa s =
                        c.createStatement();
                } catch (Exception e) {
                        e.printStackTrace();
                }
        }
}
```

2.2.9 SQL Syntax

```
create database stu4;

use stu4;

show databases;
```

```
create table login(username
varchar(20),

password varchar(20));



insert into login values('admin',
'12345');



select * from login;



create table student (name
varchar(20), fname varchar(20),
dob varchar(30), religio select *
from student;
```

2.3 Result

In the following section, we can see the output of the above implementation.

2.3.1 First GUI

This is the First GUI of the Employee Management System.

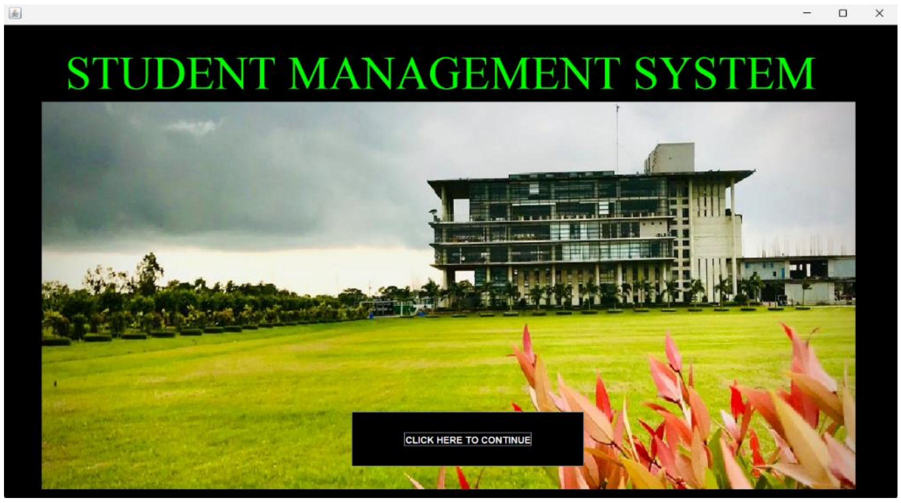


Figure 2.1: First GUI

2.3.2 Login Frame

This is the login frame of this system where the user has to enter the required credentials to have access to the main dashboard.

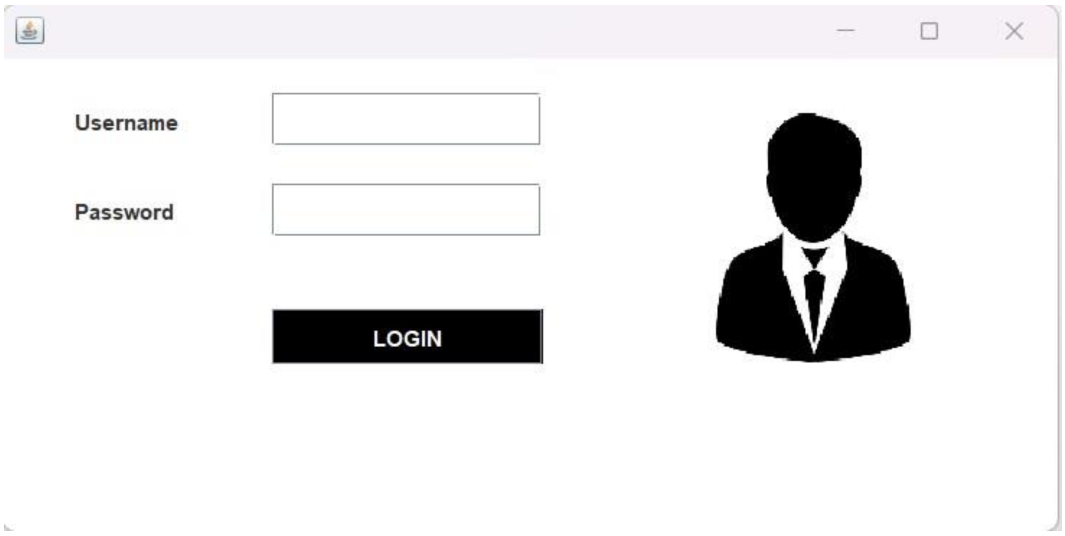


Figure 2.2: Admin Login Panel

2.3.3 Main Dashboard

After login in, the user is directed to the main dashboard of this system where users can perform various operations like adding an student, deleting an student

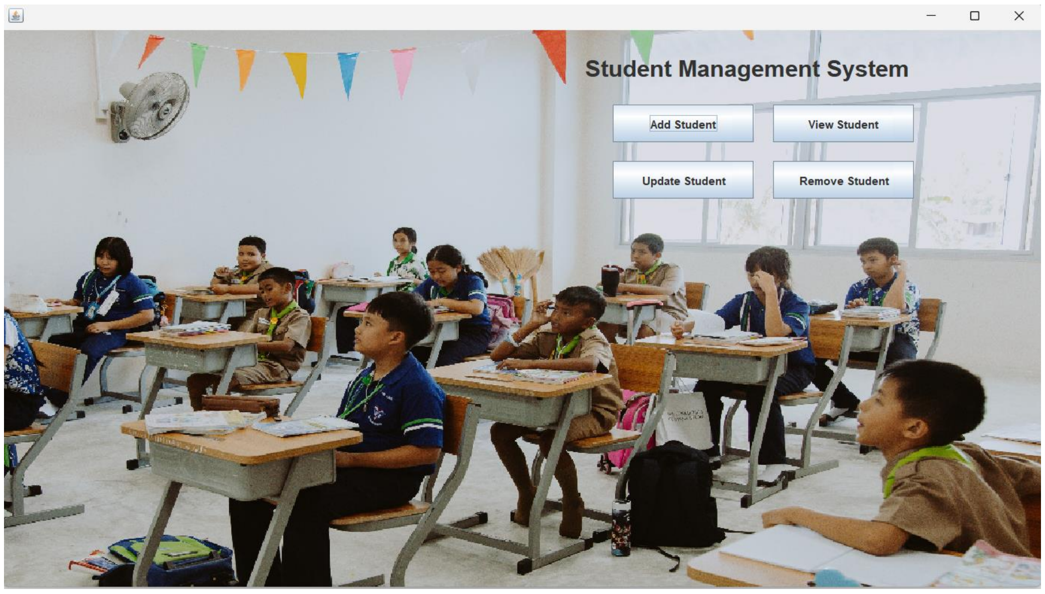
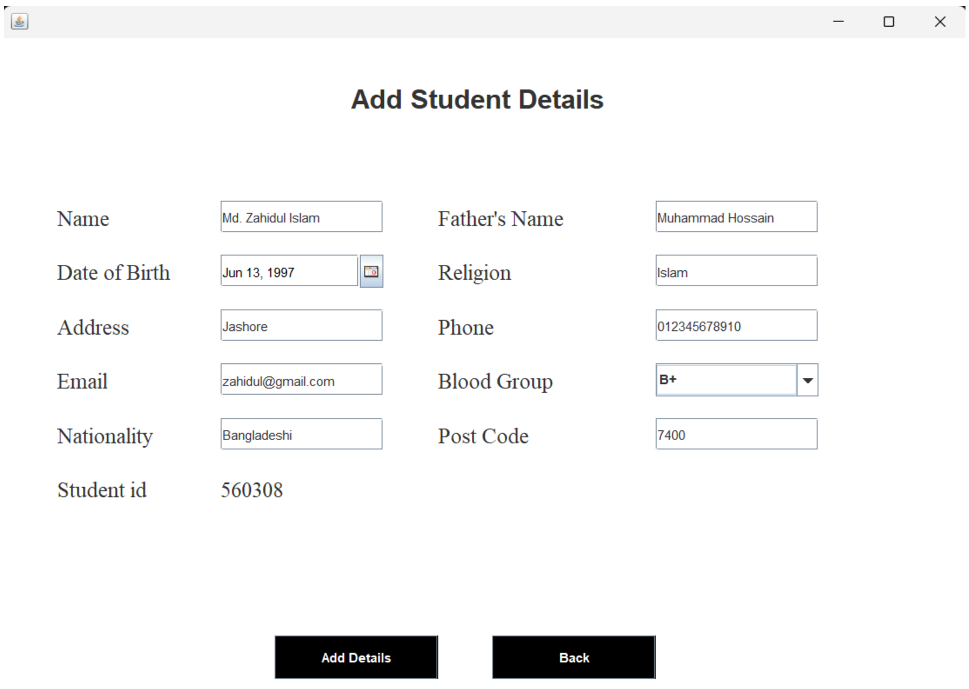


Figure 2.3: Main Dashboard

2.3.4 Add Student

Here users have to enter all the required credentials to add a new student to the system.

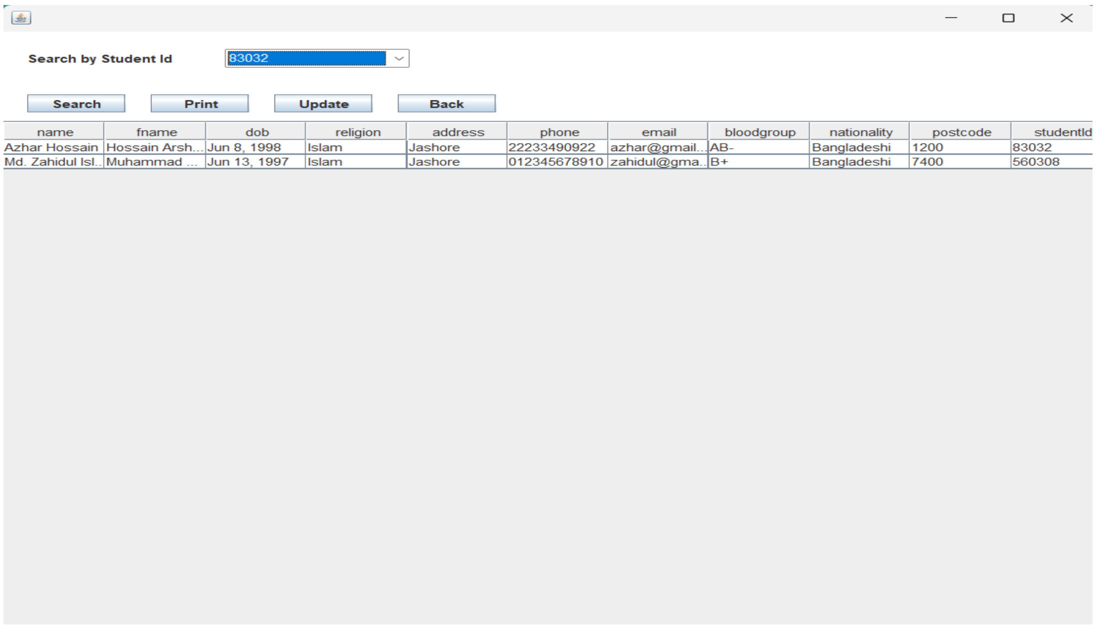

Add Student Details

Name	Md. Zahidul Islam	Father's Name	Muhammad Hossain
Date of Birth	Jun 13, 1997	Religion	Islam
Address	Jashore	Phone	012345678910
Email	zahidul@gmail.com	Blood Group	B+
Nationality	Bangladeshi	Post Code	7400
Student id	560308		

Figure 2.4: Add Student Details

2.3.5 View Student

In order to view student information, the user has to enter the student ID. In This Section, Users can search for Information by Student id and users can use Update, Print and they can back to the main dashboard.



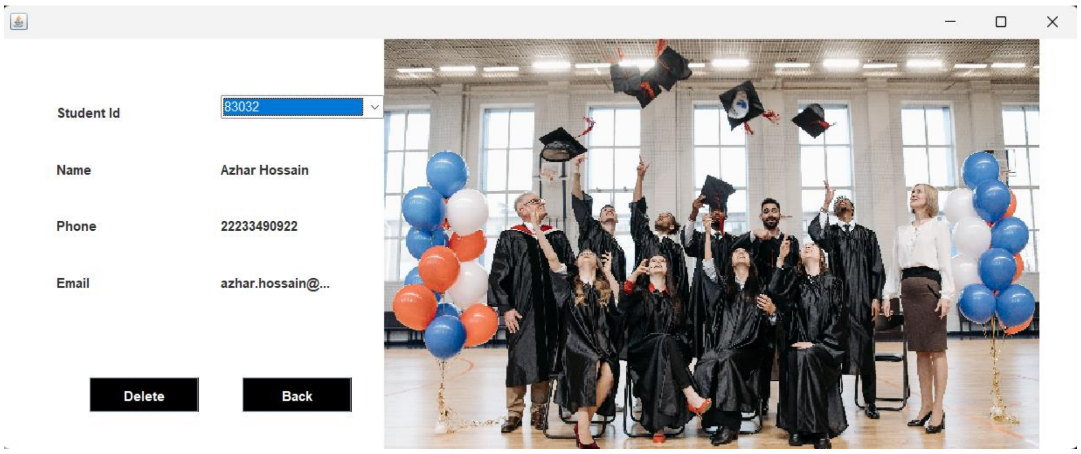
Search by Student Id: 83032

name	fname	dob	religion	address	phone	email	bloodgroup	nationality	postcode	studentId
Azhar Hossain	Hossain Arsh...	Jun 8, 1998	Islam	Jashore	22233490922	azhar@gmail...	AB-	Bangladeshi	1200	83032
Md. Zahidul Isl...	Muhammad ...	Jun 13, 1997	Islam	Jashore	012345678910	zahidul@gma...	B+	Bangladeshi	7400	560308

Figure 2.5: View Student

2.3.6 Remove Students

The user has to enter the student id in order to delete his/her information from the system.

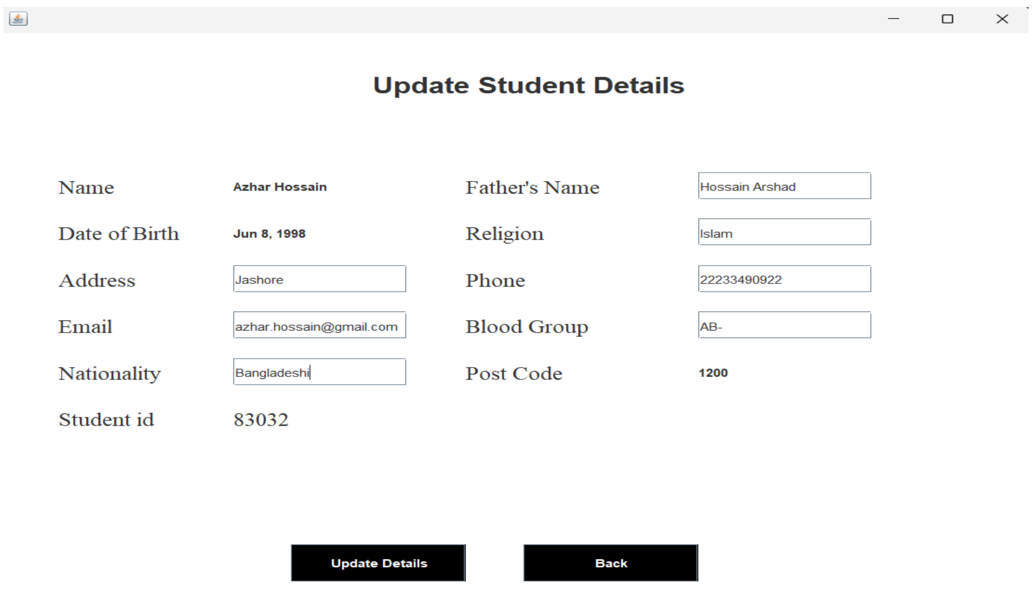


Student Id: 83032
Name: Azhar Hossain
Phone: 22233490922
Email: azhar.hossain@...

Figure 2.6: Remove Student

2.3.7 Update Student Details

If any Corrections need. Here we can update Student Information.



Update Student Details

Name	Azhar Hossain	Father's Name	Hossain Arshad
Date of Birth	Jun 8, 1998	Religion	Islam
Address	Jashore	Phone	22233490922
Email	azhar.hossain@gmail.com	Blood Group	AB-
Nationality	Bangladeshi	Post Code	1200
Student id	83032		

Figure 2.7: Update Student Details

Chapter 3

Conclusion

3.1 Discussion

Our student management system in Java project is designed to be effective, efficient, and user-friendly. It has been developed using the Java language with a MySQL database to manage students' academic information and provide a seamless experience for teachers and administrators. The system has the potential to accommodate vast data, providing a useful tool for analyzing past student data and enabling early intervention where necessary.

3.2 Limitations

While developing the Student Management System project in Java, we need to consider certain limitations. Firstly, data security and privacy must be ensured to protect sensitive student information from unauthorized access or data breaches. Secondly, the system should be scalable and capable of handling a growing number of student and organizational requirements. Thirdly, integration with existing HR systems or third party applications might pose compatibility challenges. Additionally, user-friendliness and ease of navigation should be prioritized to ensure seamless adoption and efficient usage by HR personnel. Lastly, proper documentation and training materials should be provided to facilitate system maintenance and user support.

3.3 Scope of Future Work

The GUI and the features added to this system are the basic ones. In the future, there will be a better Graphical User Interface and there will be more features added to this system. If the Graphical User interface is improved then this system will be more userfriendly and more features added will make this system a lot better and HR will be able to perform more operations.

3.4 References

- Google.
- YouTube.
- GitHub.
- Codeforces.
- Oracle.
- HackerRank.