



OOP LAB PROJECT

SUBMITTED TO: MISS FAOUZIA JABEEN



**MRT WWW.WIN2FARSI.COMNAME: ABDUL SAMAD+FASSI UR
REHMAN**

**REG.NO: FA23-BAI-005+FA23-BAI-043
OBJECT ORIENTED PROGRAMMING**

STUDENTS RESULT MANAGEMENT SYSTEM

CODE

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.util.ArrayList;


public class ResultManagementSystem {

    private JFrame frame;

    private ArrayList<Student> students = new ArrayList<>();

    private static final String ADMIN_ID = "123"; // Hardcoded admin ID
private static final String ADMIN_PASSWORD = "123"; // Hardcoded admin password


    public static void main(String[] args) {

        new ResultManagementSystem();

    }


    public ResultManagementSystem() {

        frame = new JFrame("Student Result Management System");

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(600, 400);

        frame.setLocationRelativeTo(null); // Center the window on the screen

        initializeUI();

    }

}
```

```
        frame.addWindowListener(new WindowAdapter() {  
            @Override  
            public void windowClosing(WindowEvent e) {  
JOptionPane.showMessageDialog(frame, "Thanks For Using our System!");  
                System.exit(0); // Close the application  
            }  
        });  
    }  
  
    private void initializeUI() {  
  
        // JOptionPane.showMessageDialog(frame, "Welcome!");  
  
        JPanel welcomePanel = new JPanel();  
        welcomePanel.add(new JLabel("Welcome!"));  
  
        int option =  
JOptionPane.showOptionDialog(frame, welcomePanel, "Welcome", JOptionPane.DEFAULT_OPTION  
        ,JOptionPane.INFORMATION_MESSAGE, null, new String[]{"Continue"}, "Continue" // Button label  
  
        );  
  
        JButton adminButton = new JButton("Admin");  
        JButton studentButton = new JButton("Student");  
  
        adminButton.addActionListener(e-> adminLogin());
```

```
studentButton.addActionListener(e-> studentActionsUI());
```

```
JPanel panel = new JPanel();
```

```
panel.add(adminButton);
```

```
panel.add(studentButton);
```

```
frame.getContentPane().add(panel);
```

```
frame.setVisible(true);
```

```
}
```

```
private void adminLogin() {
```

```
    JTextField adminIdField = new JTextField(20);
```

```
    JPasswordField adminPasswordField = new JPasswordField(20); // Change to JPasswordField
```

```
        JPanel panel = new JPanel();
```

```
        panel.setLayout(new GridLayout(2, 2));
```

```
        panel.add(new JLabel("Admin ID:"));
```

```
        panel.add(adminIdField);
```

```
        panel.add(new JLabel("Password:"));
```

```
        panel.add(adminPasswordField);
```

```
        int option = JOptionPane.showConfirmDialog(frame, panel, "Admin Login",  
            JOptionPane.OK_CANCEL_OPTION);
```

```
        if (option == JOptionPane.OK_OPTION) {
```

```
            String adminId = adminIdField.getText();
```

```
            String adminPassword = new String(adminPasswordField.getPassword()); // Retrieve  
                password securely
```

```

        if (adminId.equals(ADMIN_ID) && adminPassword.equals(ADMIN_PASSWORD)) {
            adminActions(); // Proceed to admin actions after successful login
        } else {
            JOptionPane.showMessageDialog(frame, "Invalid Admin ID or Password.");
        }
    }
}

private void adminActions() {
    String[] options = {"Enter Marks", "Add Student", "View Profile", "Cancel"};
    int choice = JOptionPane.showOptionDialog(frame, "Choose an action:", "Admin Actions",
        JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null, options,
        options[0]);

    if (choice == 0) {
        enterMarks();
    } else if (choice == 1) {
        addStudent();
    } else if (choice == 2) {
        viewStudentProfile(); // Add this new action
    }
}

private void viewStudentProfile() {
    // Create a list of student names to display
    DefaultListModel<String> studentListModel = new DefaultListModel<>();
    for (Student student : students) {
        studentListModel.addElement(student.getName() + " (" + student.getId() + ")");
    }
}

```

```
}
```

```
JList<String> studentList = new JList<>(studentListModel);  
studentList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
```

```
JScrollPane listScrollPane = new JScrollPane(studentList);
```

```
JPanel panel = new JPanel();  
panel.setLayout(new BorderLayout());  
panel.add(new JLabel("Select a Student to View Profile:"), BorderLayout.NORTH);  
panel.add(listScrollPane, BorderLayout.CENTER);
```

```
int option = JOptionPane.showConfirmDialog(frame, panel, "Select Student",  
JOptionPane.OK_CANCEL_OPTION);  
if (option == JOptionPane.OK_OPTION) {  
    String selectedStudent = studentList.getSelectedValue();  
    if (selectedStudent != null) {  
        String studentId = selectedStudent.substring(selectedStudent.indexOf('(') + 1,  
selectedStudent.indexOf(')'));  
        Student student = findStudentById(studentId);  
        if (student != null) {  
            showStudentProfile(student); // Display the selected student's profile  
        }  
    }  
}  
}
```

```
        private void enterMarks() {  
            JTextField studentIdField = new JTextField(20);  
            JTextField numSubjectsField = new JTextField(5);  
            JPanel panel = new JPanel();  
            panel.setLayout(new GridLayout(3, 2));  
            panel.add(new JLabel("Student ID:"));  
            panel.add(studentIdField);  
            panel.add(new JLabel("Number of Subjects:"));  
            panel.add(numSubjectsField);  
  
            int option = JOptionPane.showConfirmDialog(frame, panel, "Enter Marks",  
                JOptionPane.OK_CANCEL_OPTION);  
            if (option == JOptionPane.OK_OPTION) {  
                String studentId = studentIdField.getText();  
                int numSubjects;  
  
                try {  
                    numSubjects = Integer.parseInt(numSubjectsField.getText());  
                } catch (NumberFormatException e) {  
                    JOptionPane.showMessageDialog(frame, "Please enter a valid number of subjects.");  
                    return;  
                }  
  
                Student student = findStudentById(studentId);  
                if (student != null) {  
                    for (int i = 0; i < numSubjects; i++) {  
                        JTextField courseNoField = new JTextField(10);  
                        JTextField courseTitleField = new JTextField(20);
```

```
    JTextField creditHoursField = new JTextField(5);
```

```
    JTextField marksField = new JTextField(5);
```

```
        JPanel subjectPanel = new JPanel();
```

```
        subjectPanel.setLayout(new GridLayout(4, 2));
```

```
        subjectPanel.add(new JLabel("Course No:"));
```

```
            subjectPanel.add(courseNoField);
```

```
        subjectPanel.add(new JLabel("Course Title:"));
```

```
            subjectPanel.add(courseTitleField);
```

```
        subjectPanel.add(new JLabel("Credit Hours:"));
```

```
            subjectPanel.add(creditHoursField);
```

```
        subjectPanel.add(new JLabel("Marks:"));
```

```
            subjectPanel.add(marksField);
```

```
int subjectOption = JOptionPane.showConfirmDialog(frame, subjectPanel, "Enter  
Subject Details",
```

```
    JOptionPane.OK_CANCEL_OPTION);
```

```
if (subjectOption == JOptionPane.OK_OPTION) {
```

```
    String courseNo = courseNoField.getText();
```

```
    String courseTitle = courseTitleField.getText();
```

```
int creditHours = Integer.parseInt(creditHoursField.getText());
```

```
int marks = Integer.parseInt(marksField.getText());
```

```
    String grade;
```

```
    double gradePoint;
```

```
    if (marks >= 85) {
```

```
        grade = "A";
```



```
        gradePoint = 4.0;
    } else if (marks >= 80) {
        grade = "A-";
        gradePoint = 3.66;
    } else if (marks >= 75) {
        grade = "B+";
        gradePoint = 3.33;
    } else if (marks >= 71) {
        grade = "B";
        gradePoint = 3.00;
    } else if (marks >= 68) {
        grade = "B-";
        gradePoint = 2.66;
    } else if (marks >= 64) {
        grade = "C+";
        gradePoint = 2.33;
    } else if (marks >= 61) {
        grade = "C";
        gradePoint = 2.00;
    } else if (marks >= 58) {
        grade = "C-";
        gradePoint = 1.66;
    } else if (marks >= 54) {
        grade = "D+";
        gradePoint = 1.3;
    } else if (marks >= 50) {
        grade = "D";
```

```
        gradePoint = 1.00;
    } else {
        grade = "F";
        gradePoint = 0.00;
    }

    student.addSubject(courseNo, courseTitle, creditHours, marks, grade, gradePoint);
}
}

double totalGradePoints = 0.0;
int totalCredits = 0;

for (String[] subject : student.getSubjects()) {
    double subjectGradePoint = Double.parseDouble(subject[5]);
    int subjectCredits = Integer.parseInt(subject[2]);
    totalGradePoints += subjectGradePoint * subjectCredits;
    totalCredits += subjectCredits;
}

double gpa = totalGradePoints / totalCredits;
student.setGPA(gpa);

JOptionPane.showMessageDialog(frame, "GPA calculated and updated for " +
    student.getName() + ": " + gpa);
} else {
    JOptionPane.showMessageDialog(frame, "Student ID not found.");
}
```

```
}  
}
```

```
private void addStudent() {  
    JTextField idField = new JTextField(10);  
    JTextField nameField = new JTextField(20);  
    JTextField passwordField = new JTextField(10);  
    JTextField registrationField = new JTextField(15);  
    JTextField fatherNameField = new JTextField(20);  
    JTextField semesterField = new JTextField(5);  
    JTextField programField = new JTextField(15);
```

```
        JPanel panel = new JPanel();  
        panel.setLayout(new GridLayout(7, 2));  
        panel.add(new JLabel("Student ID:"));  
        panel.add(idField);  
        panel.add(new JLabel("Name:"));  
        panel.add(nameField);  
        panel.add(new JLabel("Password:"));  
        panel.add(passwordField);  
        panel.add(new JLabel("Registration No:"));  
        panel.add(registrationField);  
        panel.add(new JLabel("Father's Name:"));  
        panel.add(fatherNameField);  
        panel.add(new JLabel("Semester:"));  
        panel.add(semesterField);  
        panel.add(new JLabel("Program:"));
```

```
        panel.add(programField);

int option = JOptionPane.showConfirmDialog(frame, panel, "Add Student",
        JOptionPane.OK_CANCEL_OPTION);

if (option == JOptionPane.OK_OPTION) {
    String id = idField.getText();
    String name = nameField.getText();
    String password = passwordField.getText();
    String registration = registrationField.getText();
    String fatherName = fatherNameField.getText();
    String semester = semesterField.getText();
    String program = programField.getText();

    Student student = new Student(id, name, password);
    student.setDetails(registration, fatherName, semester, program);
    students.add(student);

    JOptionPane.showMessageDialog(frame, "Student added successfully!");
}

}

private void studentActionsUI() {
    JTextField studentIdField = new JTextField(20);
    JPanel panel = new JPanel();
    panel.add(new JLabel("Enter your Student ID:"));
    panel.add(studentIdField);
```

```

int option = JOptionPane.showConfirmDialog(frame, panel, "Student Login",
JOptionPane.OK_CANCEL_OPTION);

if (option == JOptionPane.OK_OPTION) {

String studentId = studentIdField.getText();

Student student = findStudentById(studentId);

    if (student != null) {
        studentLogin(student);
    } else {
        JOptionPane.showMessageDialog(frame, "Invalid Student ID.");
    }
}

}

private void studentLogin(Student student) {

JPasswordField passwordField = new JPasswordField(20); // Change to JPasswordField

JPanel panel = new JPanel();

panel.add(new JLabel("Enter your Password:"));

panel.add(passwordField);

int option = JOptionPane.showConfirmDialog(frame, panel, "Student Login",
JOptionPane.OK_CANCEL_OPTION);

if (option == JOptionPane.OK_OPTION) {

String enteredPassword = new String(passwordField.getPassword()); // Retrieve password
securely

    if (student.getPassword().equals(enteredPassword)) {
        studentActions(student); // Proceed to student actions if password matches
    } else {

```

```
JOptionPane.showMessageDialog(frame, "Incorrect Password.");  
  
    }  
  
    }  
  
    }
```

```
private void studentActions(Student student) {  
    String[] columnNames = {"Course No", "Course Title", "Credit Hours", "Marks", "Grade", "Grade  
        Point"};
```

```
    Object[][] rowData = new Object[student.getSubjects().size()][6];
```

```
        int row = 0;  
        for (String[] subject : student.getSubjects()) {  
            rowData[row][0] = subject[0];  
            rowData[row][1] = subject[1];  
            rowData[row][2] = subject[2];  
            rowData[row][3] = subject[3];  
            rowData[row][4] = subject[4];  
            rowData[row][5] = subject[5];  
            row++;  
        }
```

```
    JTable resultTable = new JTable(rowData, columnNames);
```

```
    JScrollPane scrollPane = new JScrollPane(resultTable);
```

```
        JPanel panel = new JPanel();
```

```
        panel.setLayout(new BorderLayout());
```

```
        panel.add(new JLabel("Result Report for " + student.getName(), SwingConstants.CENTER),  
            BorderLayout.NORTH);
```

```
        panel.add(scrollPane, BorderLayout.CENTER);

        // Add GPA display below the result table
        JPanel gpaPanel = new JPanel();
        String gpaText = student.getGPA() == 0.0 ? "GPA: Marks not uploaded yet" : "GPA: " +
            student.getGPA();
        gpaPanel.add(new JLabel(gpaText));

        panel.add(gpaPanel, BorderLayout.SOUTH);

        JOptionPane.showMessageDialog(frame, panel, "Student Result Report",
            JOptionPane.INFORMATION_MESSAGE);
    }

    private Student findStudentById(String studentId) {
        for (Student student : students) {
            if (student.getId().equals(studentId)) {
                return student;
            }
        }
        return null;
    }

    private void showStudentProfile(Student student) {
        StringBuilder profileDetails = new StringBuilder();
        profileDetails.append("Student ID: ").append(student.getId()).append("\n");
        profileDetails.append("Name: ").append(student.getName()).append("\n");
        profileDetails.append("Registration No: ").append(student.registration).append("\n");
    }
}
```

```
profileDetails.append("Father's Name: ").append(student.fatherName).append("\n");
    profileDetails.append("Semester: ").append(student.semester).append("\n");
    profileDetails.append("Program: ").append(student.program).append("\n");

    // Check if GPA is 0.0, indicating marks have not been entered yet
    String gpaText = student.getGPA() == 0.0 ? "GPA: Marks not uploaded yet" : "GPA: " +
        student.getGPA();

    profileDetails.append(gpaText).append("\n");

    JOptionPane.showMessageDialog(frame, profileDetails.toString(), "Student Profile",
        JOptionPane.INFORMATION_MESSAGE);
}
```

```
// Student class

class Student {
    private String id;
    private String name;
    private String password;
    private String registration;
    private String fatherName;
    private String semester;
    private String program;
    private double GPA;
    private ArrayList<String[]> subjects = new ArrayList<>();

    public Student(String id, String name, String password) {
        this.id = id;
```



```
        this.name = name;
        this.password = password;
    }
```

```
public void setDetails(String registration, String fatherName, String semester, String
                        program) {
    this.registration = registration;
    this.fatherName = fatherName;
    this.semester = semester;
    this.program = program;
}
```

```
public void addSubject(String courseNo, String courseTitle, int creditHours, int marks, String
                        grade, double gradePoint) {
    subjects.add(new String[]{courseNo, courseTitle, String.valueOf(creditHours),
                               String.valueOf(marks), grade, String.valueOf(gradePoint)});
}
```

```
public ArrayList<String[]> getSubjects() {
    return subjects;
}
```

```
public String getPassword() {
    return password;
}
```

```
public String getId() {
    return id;
}
```

```
}
```

```
public String getName() {
```

```
    return name;
```

```
}
```

```
public void setGPA(double GPA) {
```

```
    this.GPA = GPA;
```

```
}
```

```
public double getGPA() {
```

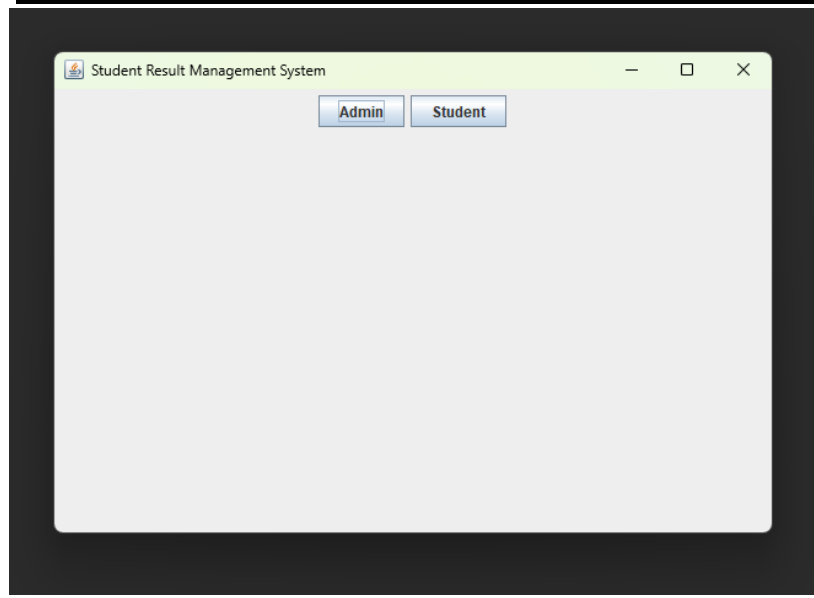
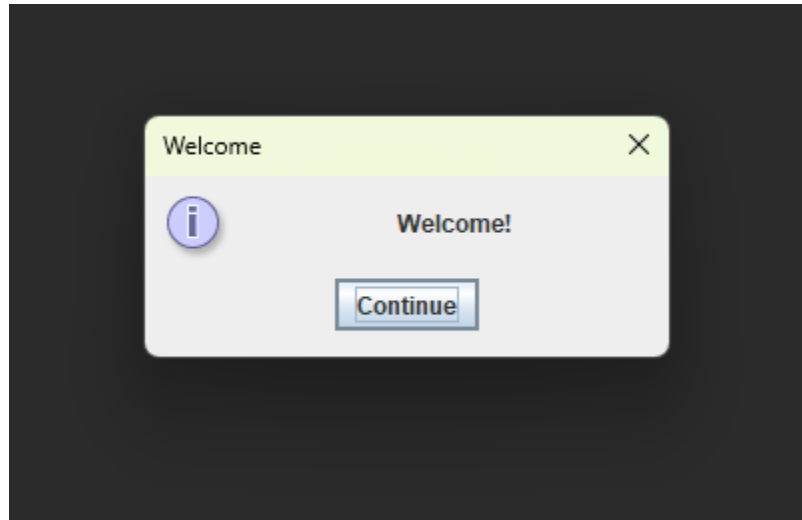
```
    return GPA;
```

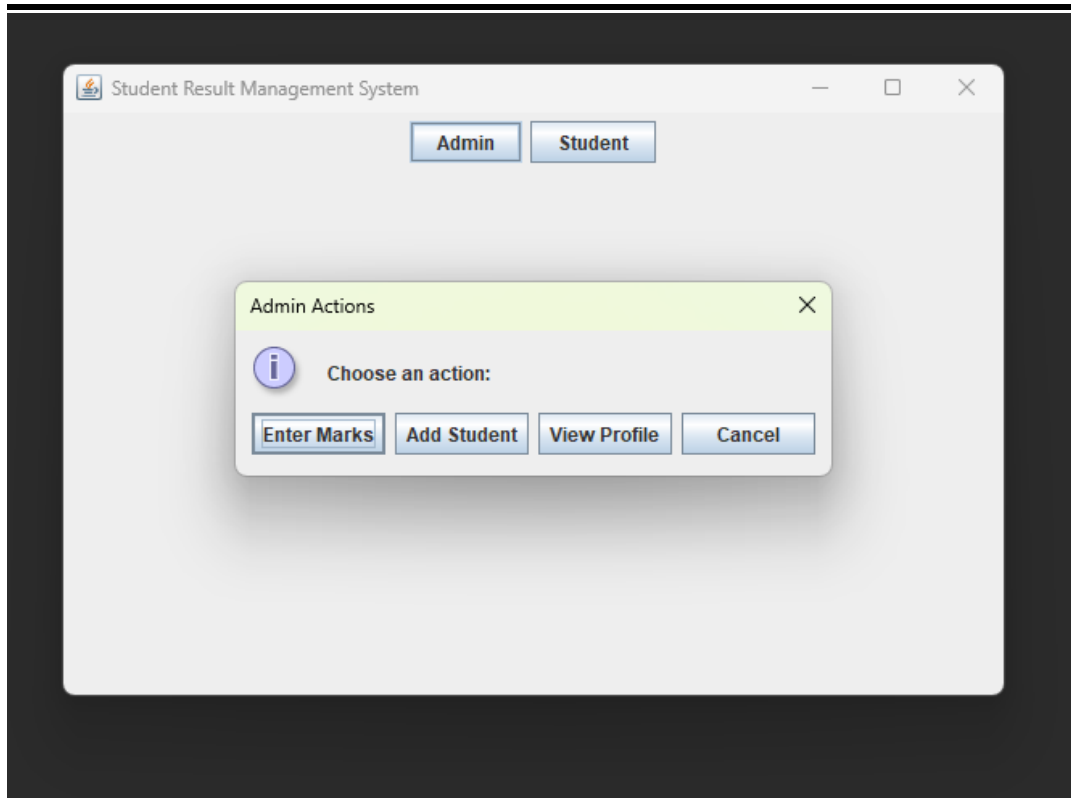
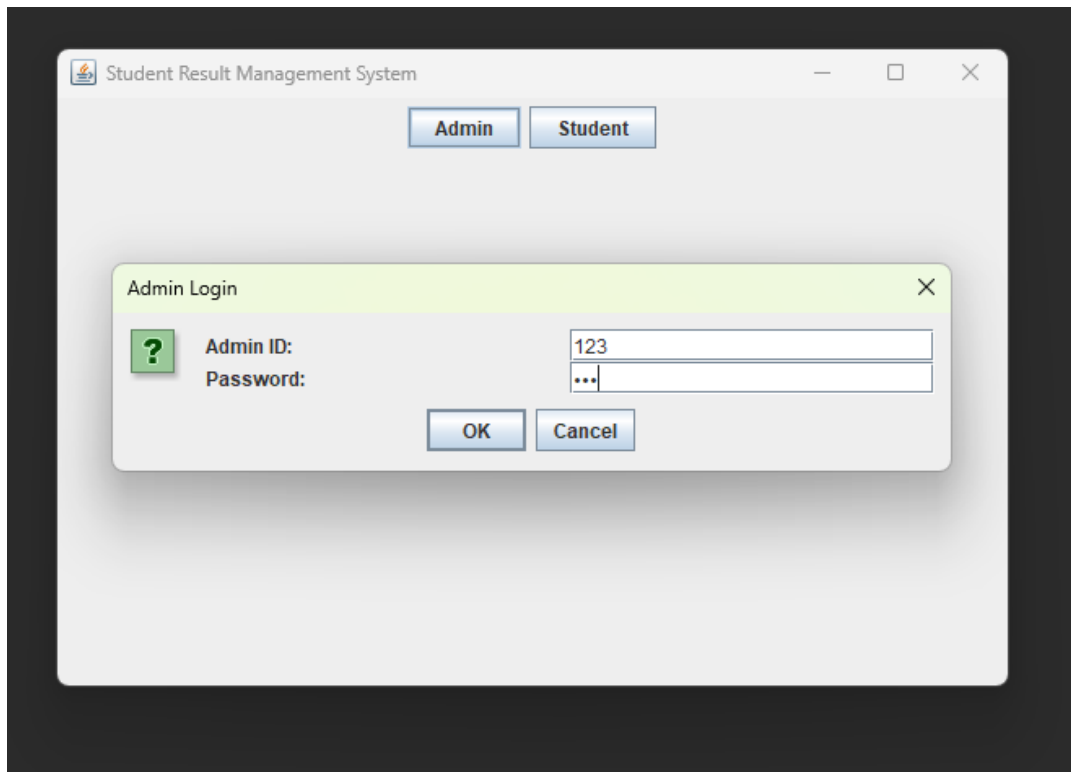
```
}
```

```
}
```

```
}
```

OUTPUT





Student Result Management System

Admin Student

Add Student

? Student ID: 1

Name: Abdul Samad

Password: 123

Registration No: 005

Father's Name: Muhammad Ilyas

Semester: 3

Program: BSAI

OK Cancel

Student Result Management System

Admin Student

Add Student

? Student ID: 2

Name: Fassi

Password: 123

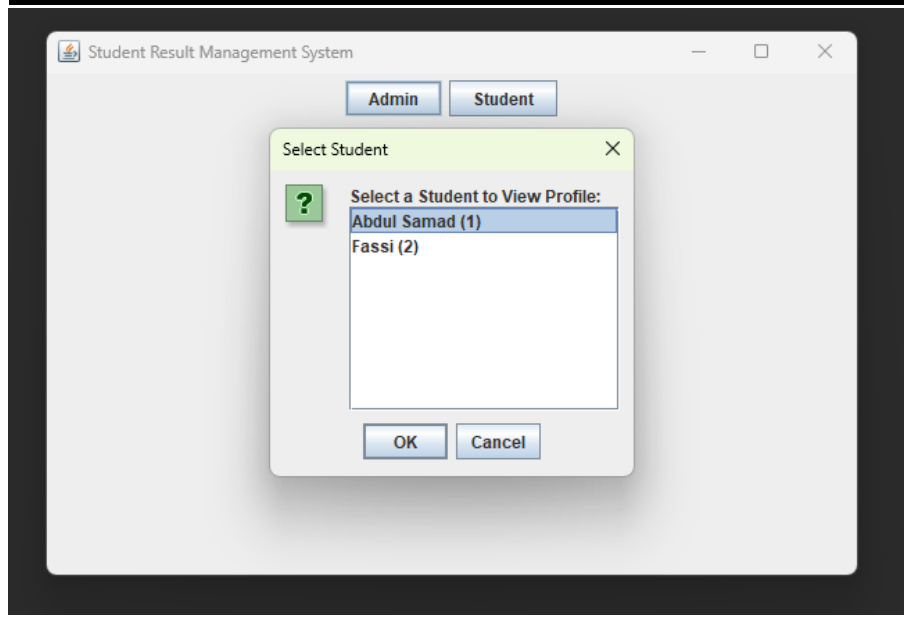
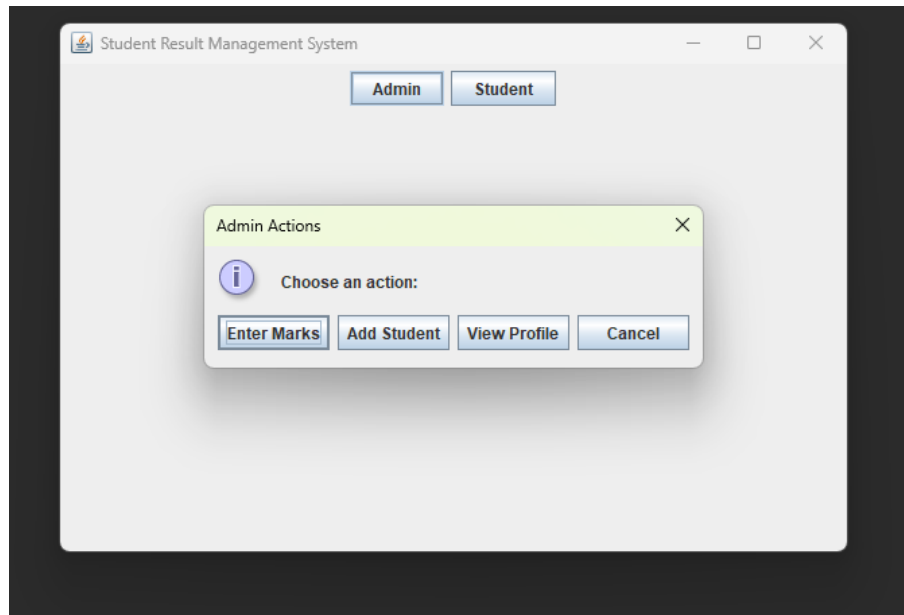
Registration No: 042

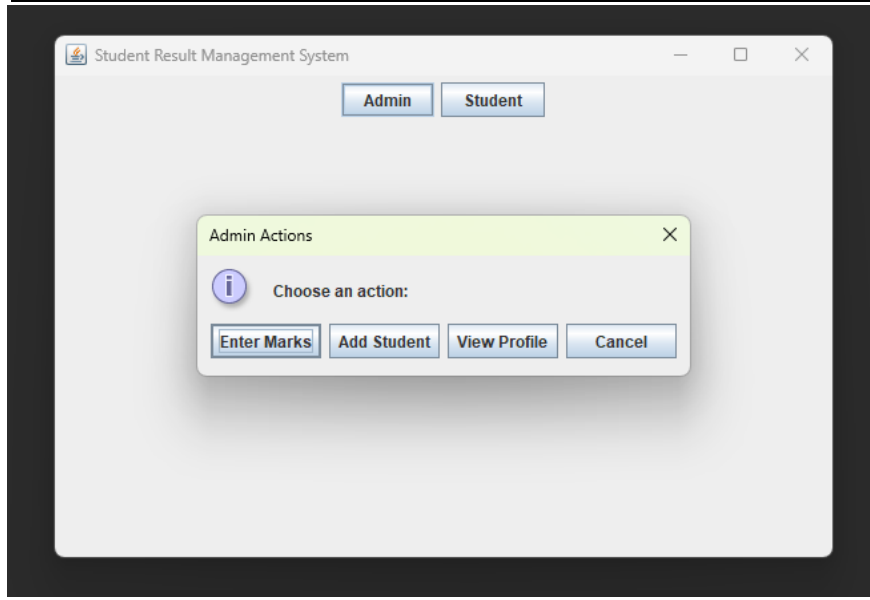
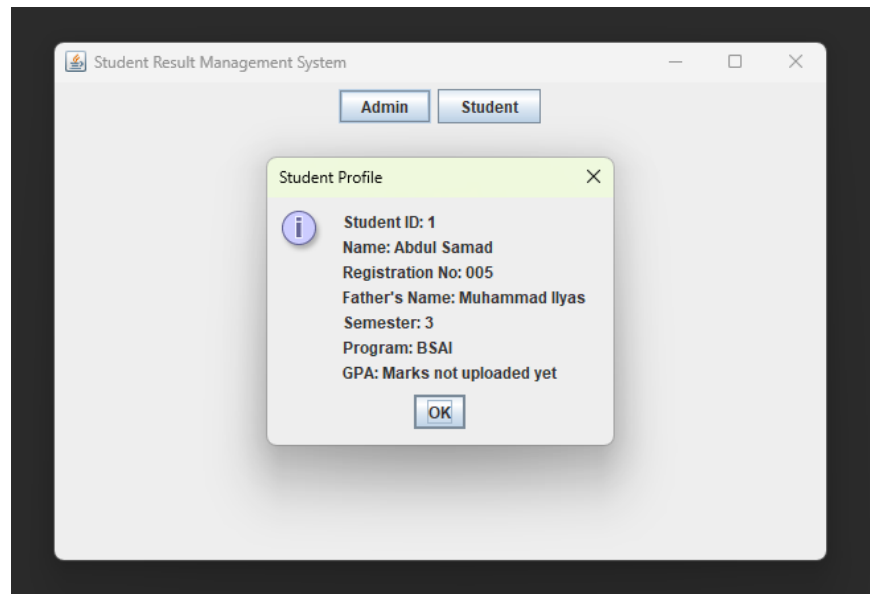
Father's Name: Muhammad

Semester: 3

Program: BSAI

OK Cancel





Student Result Management System

Admin Student

Enter Marks

? Student ID: 1

Number of Subjects: 3

OK Cancel

Student Result Management System

Admin Student

Enter Subject Details

? Course No: CSC123

Course Title: OOP

Credit Hours: 3

Marks: 75

OK Cancel

Student Result Management System

Admin Student

Enter Subject Details

? Course No: CSC973

Course Title: AICT

Credit Hours: 3

Marks: 80

OK Cancel

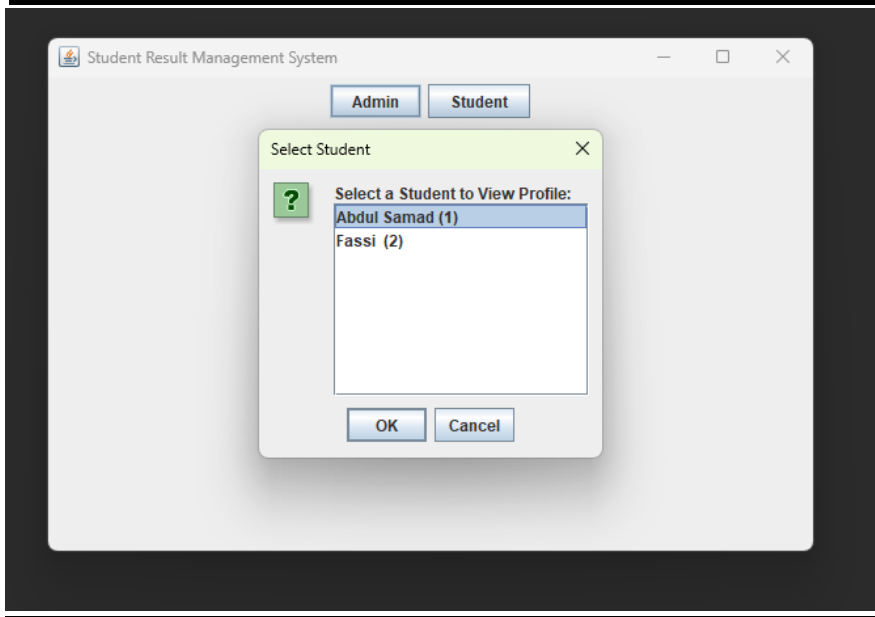
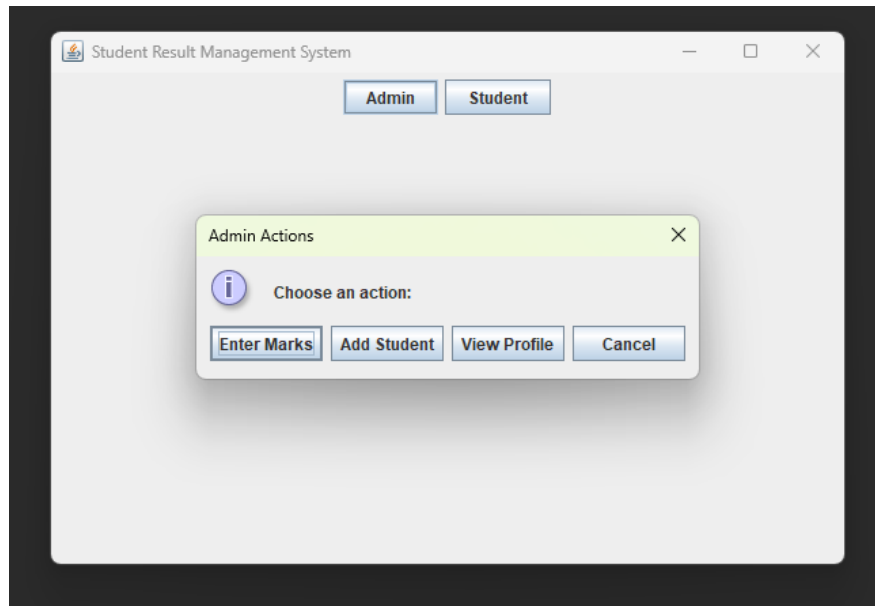
Student Result Management System

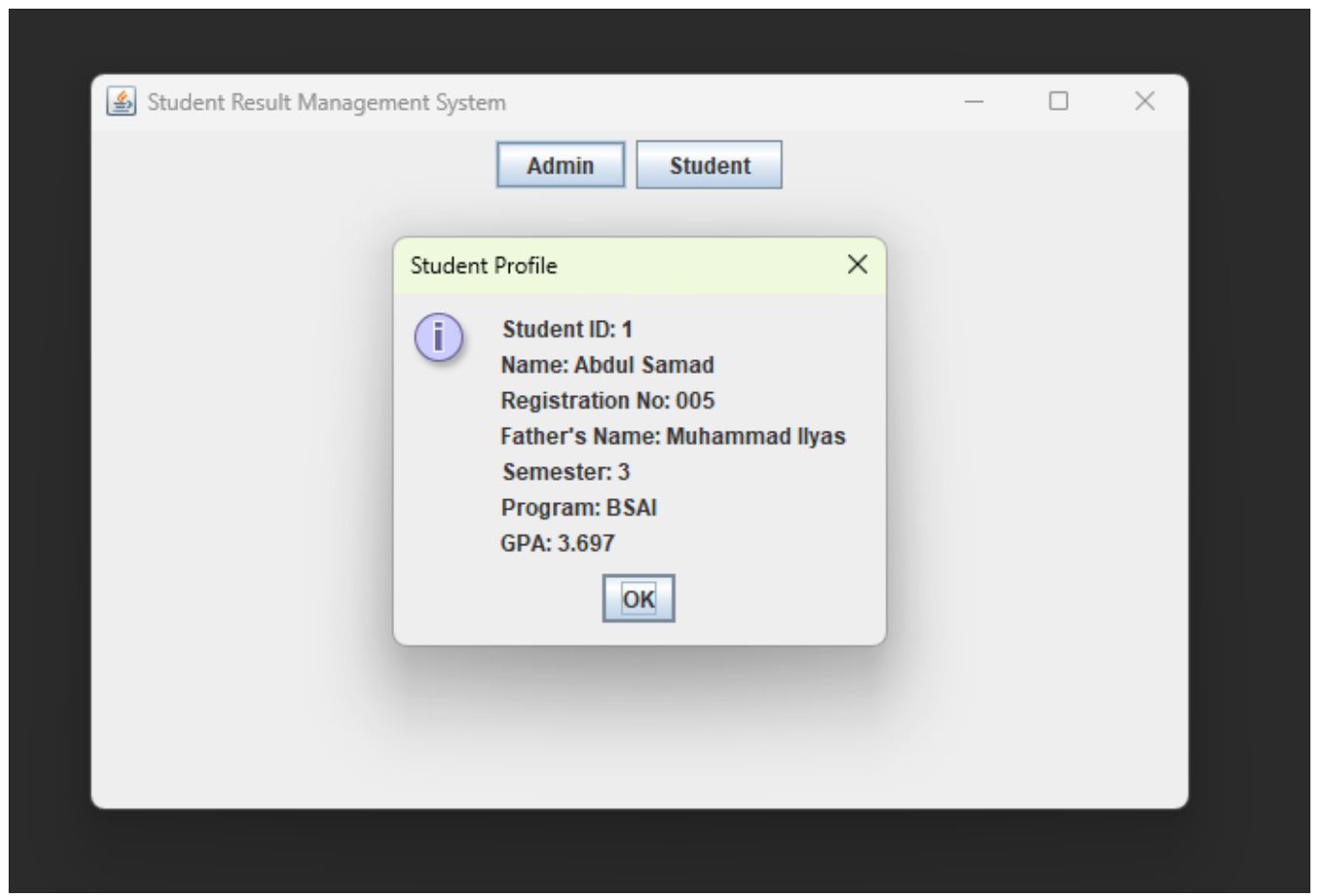
Admin Student

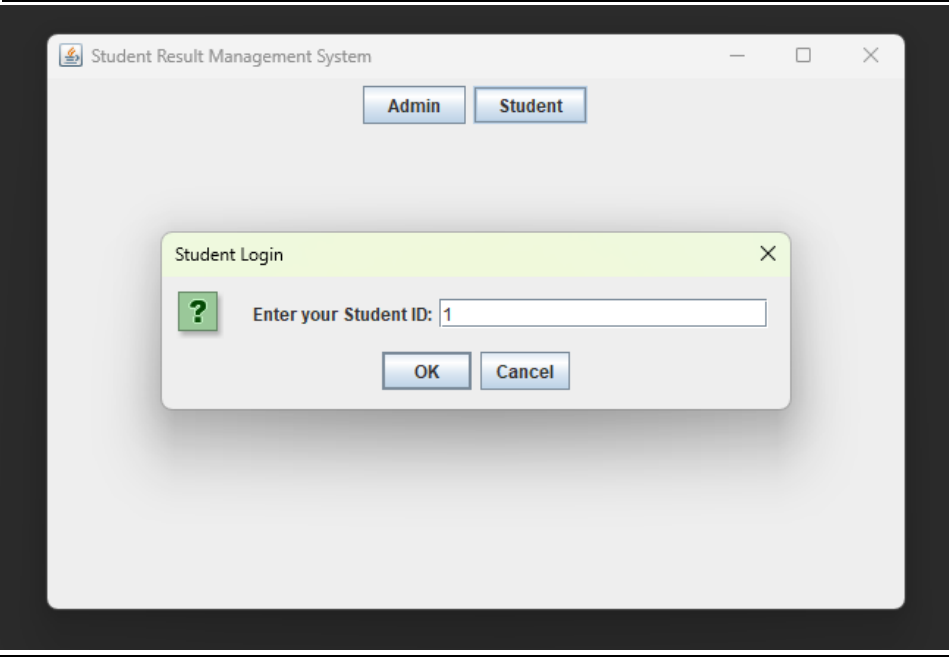
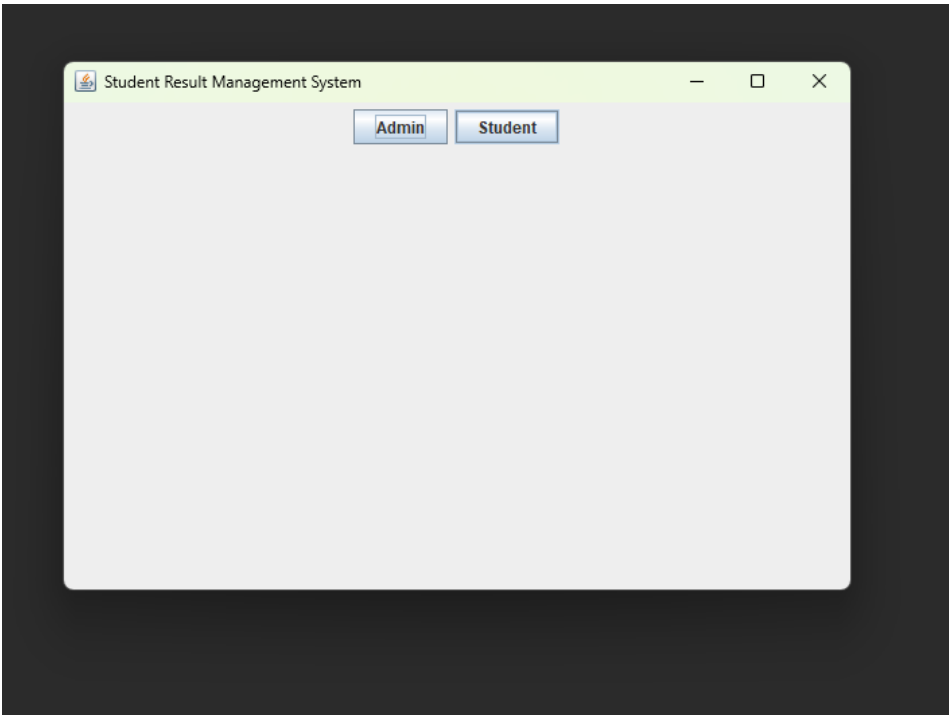
Message

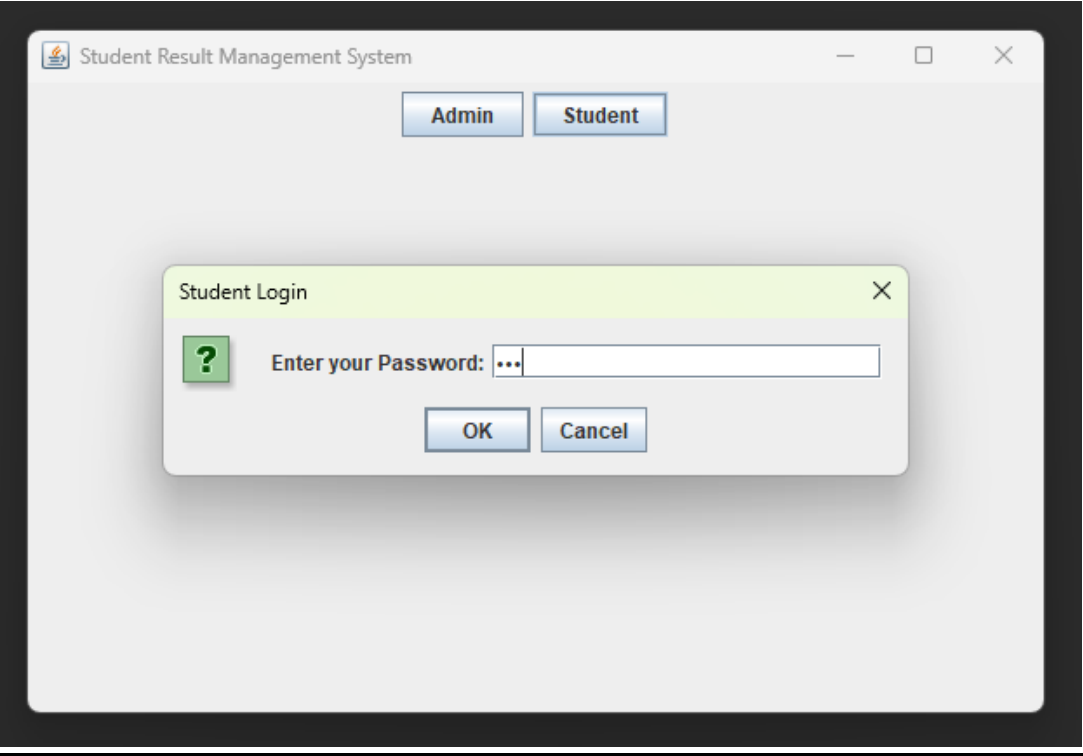
i GPA calculated and updated for Abdul Samad: 3.697

OK









Student Result Report



Result Report for Abdul Samad

Course No	Course Title	Credit Hours	Marks	Grade	Grade Point
CSC123	OOP	3	75	B+	3.33
CSC321	PF	4	90	A	4.0
CSC973	AICT	3	80	A-	3.66

GPA: 3.697

OK

