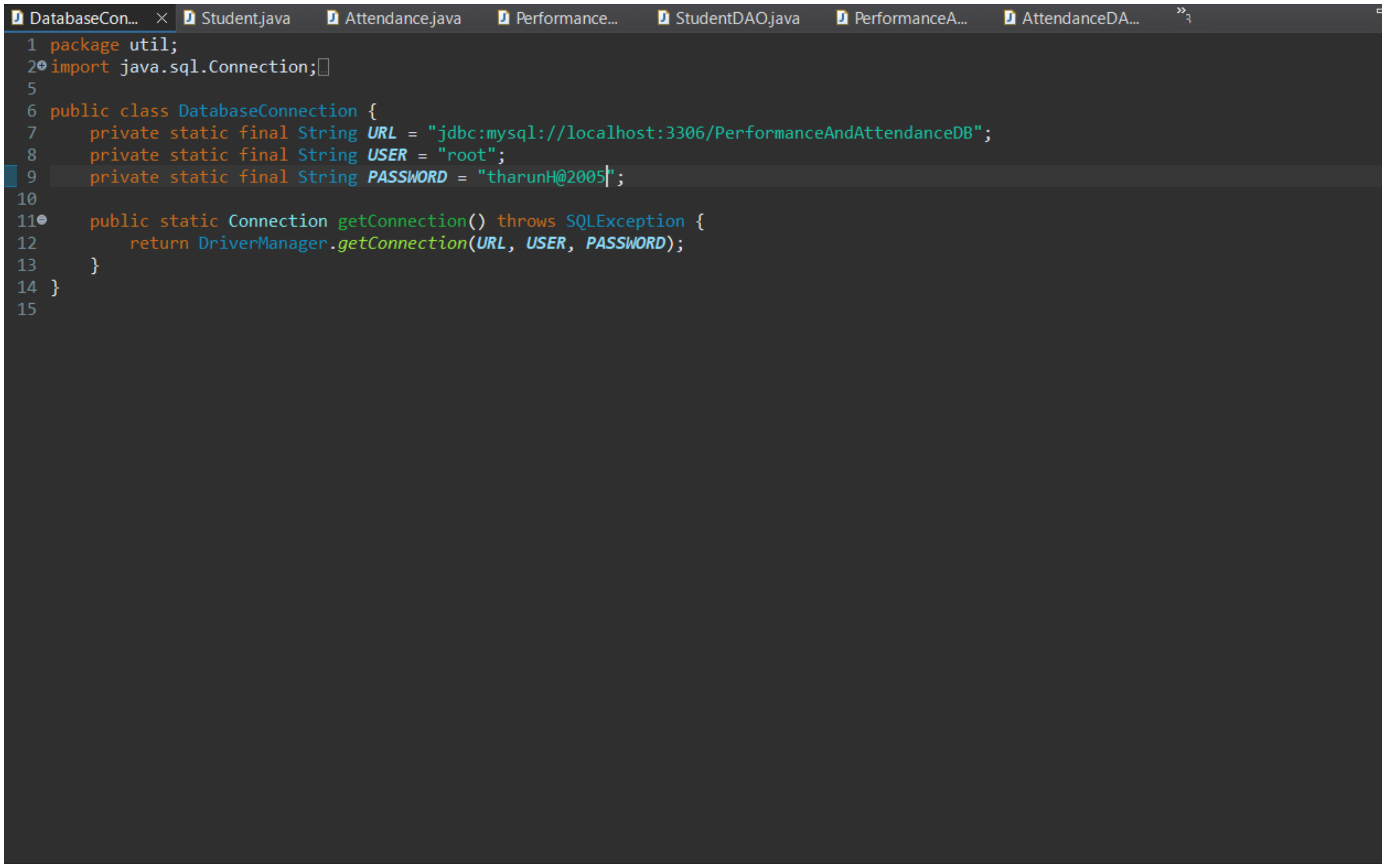
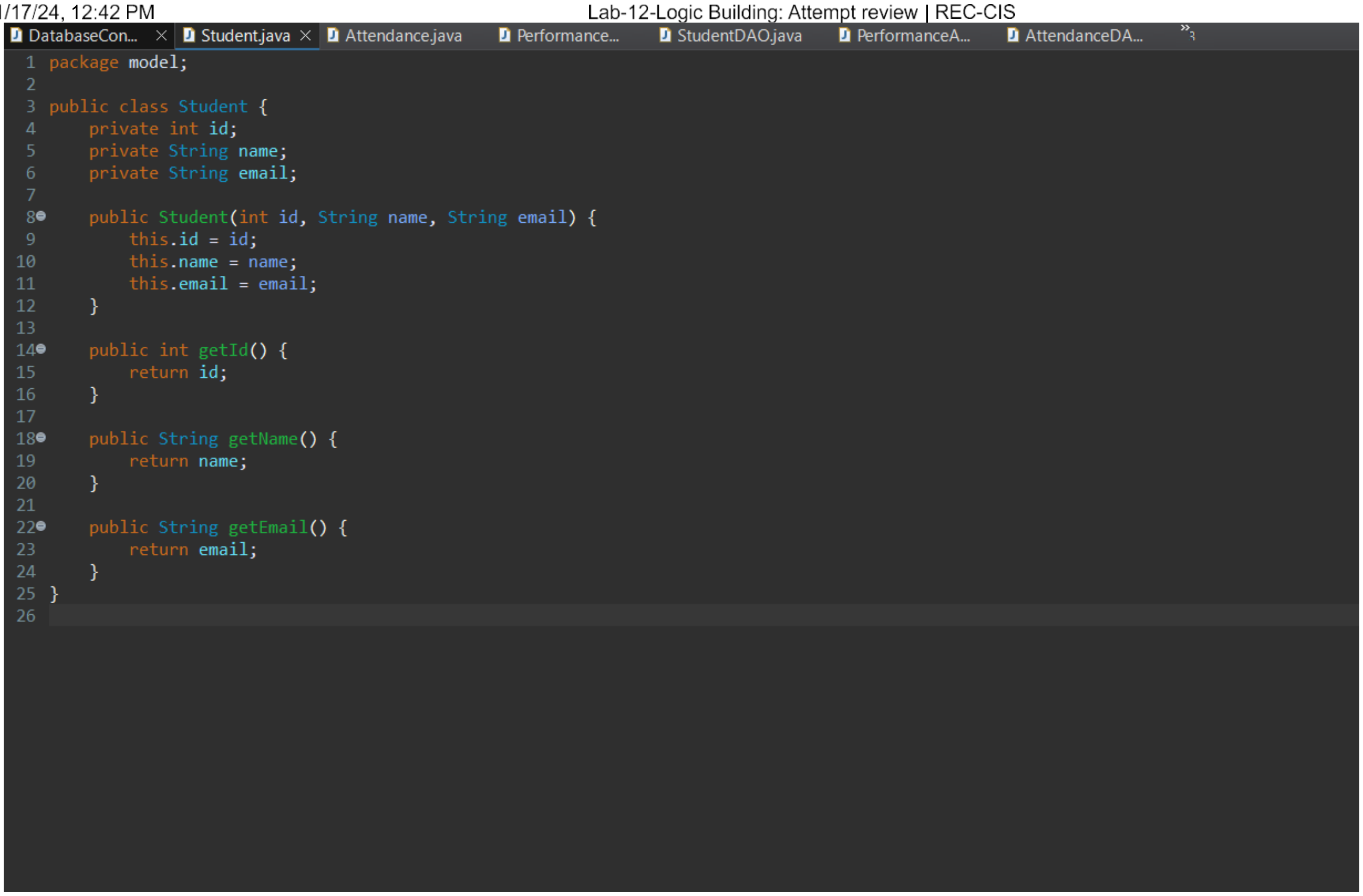
STUDENT ATTENDANCE AND PERFORMANCE SYSTEM

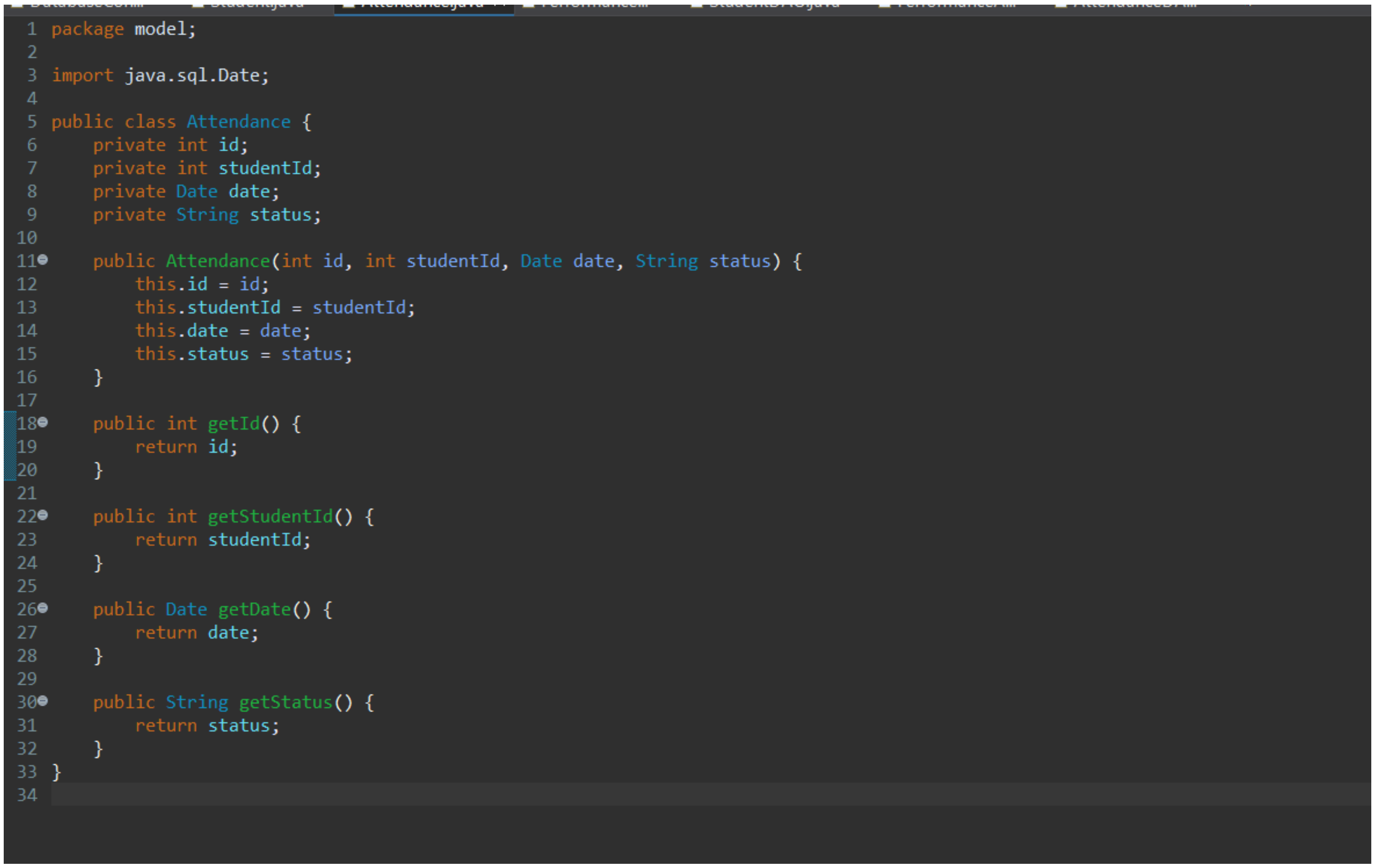


DatabaseCon... Student.java Attendance.java Performance... StudentDAO.java PerformanceA... AttendanceDA...

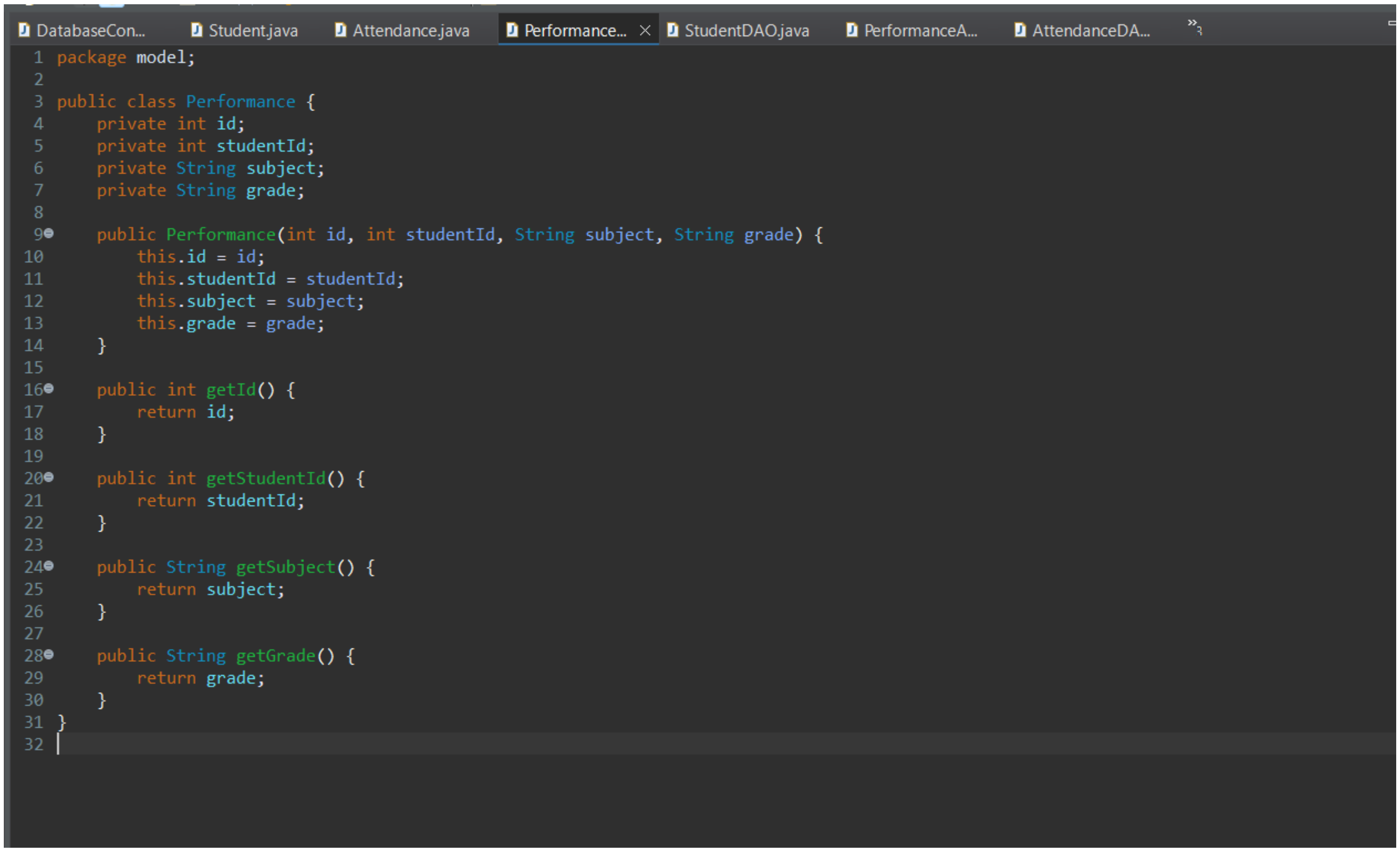
```java
package util;
import java.sql.Connection;

public class DatabaseConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/PerformanceAndAttendanceDB";
    private static final String USER = "root";
    private static final String PASSWORD = "tharunH@2005";

    public static Connection getConnection() throws SQLException {
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```



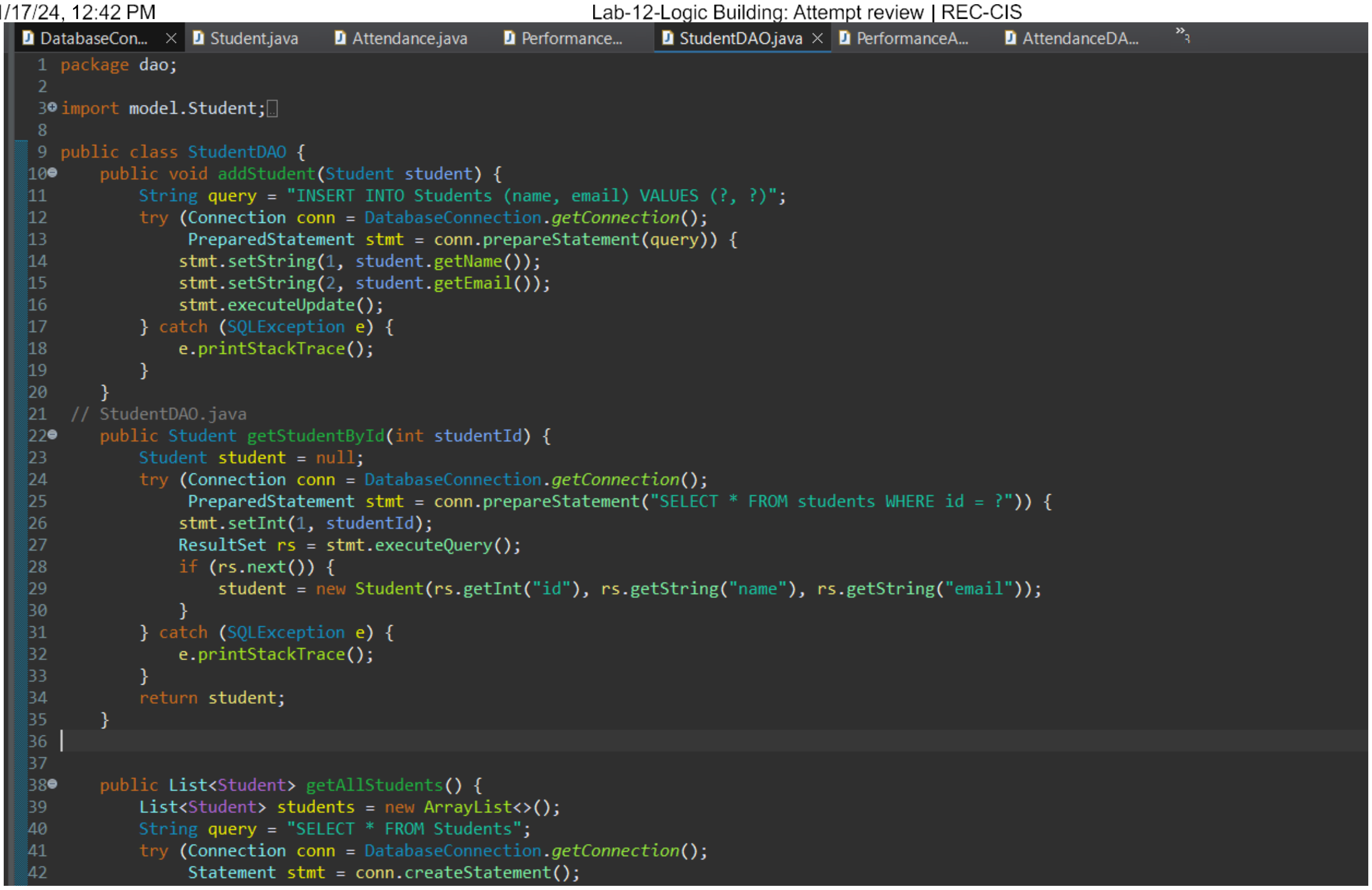
```java
package model;

public class Student {
    private int id;
    private String name;
    private String email;

    public Student(int id, String name, String email) {
        this.id = id;
        this.name = name;
        this.email = email;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }
}
```



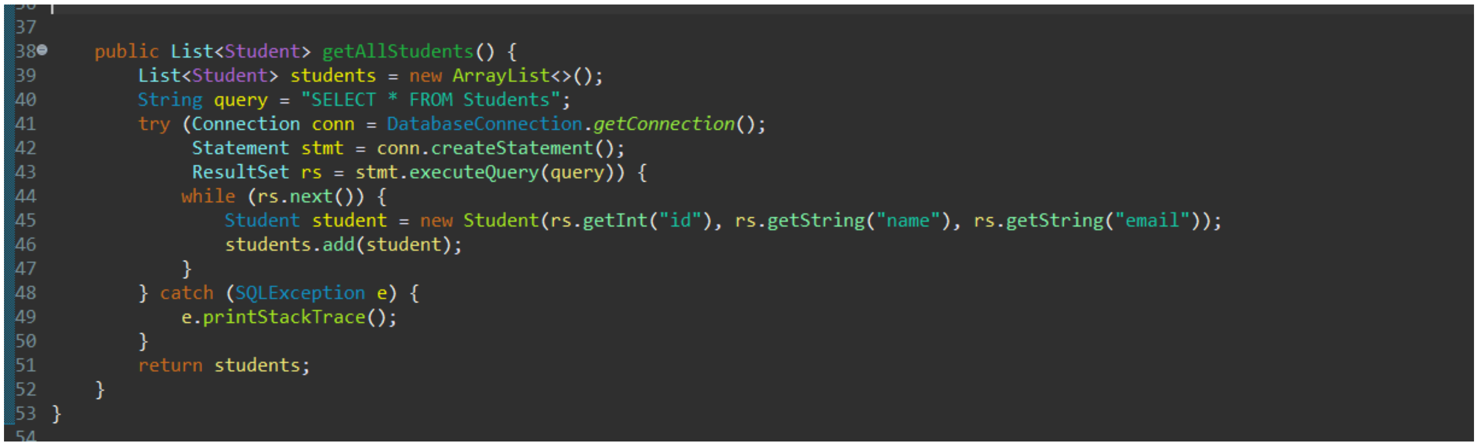
```java
package model;

import java.sql.Date;

public class Attendance {
    private int id;
    private int studentId;
    private Date date;
    private String status;

    public Attendance(int id, int studentId, Date date, String status) {
        this.id = id;
        this.studentId = studentId;
        this.date = date;
        this.status = status;
    }

    public int getId() {
        return id;
    }

    public int getStudentId() {
        return studentId;
    }

    public Date getDate() {
        return date;
    }

    public String getStatus() {
        return status;
    }
}
```



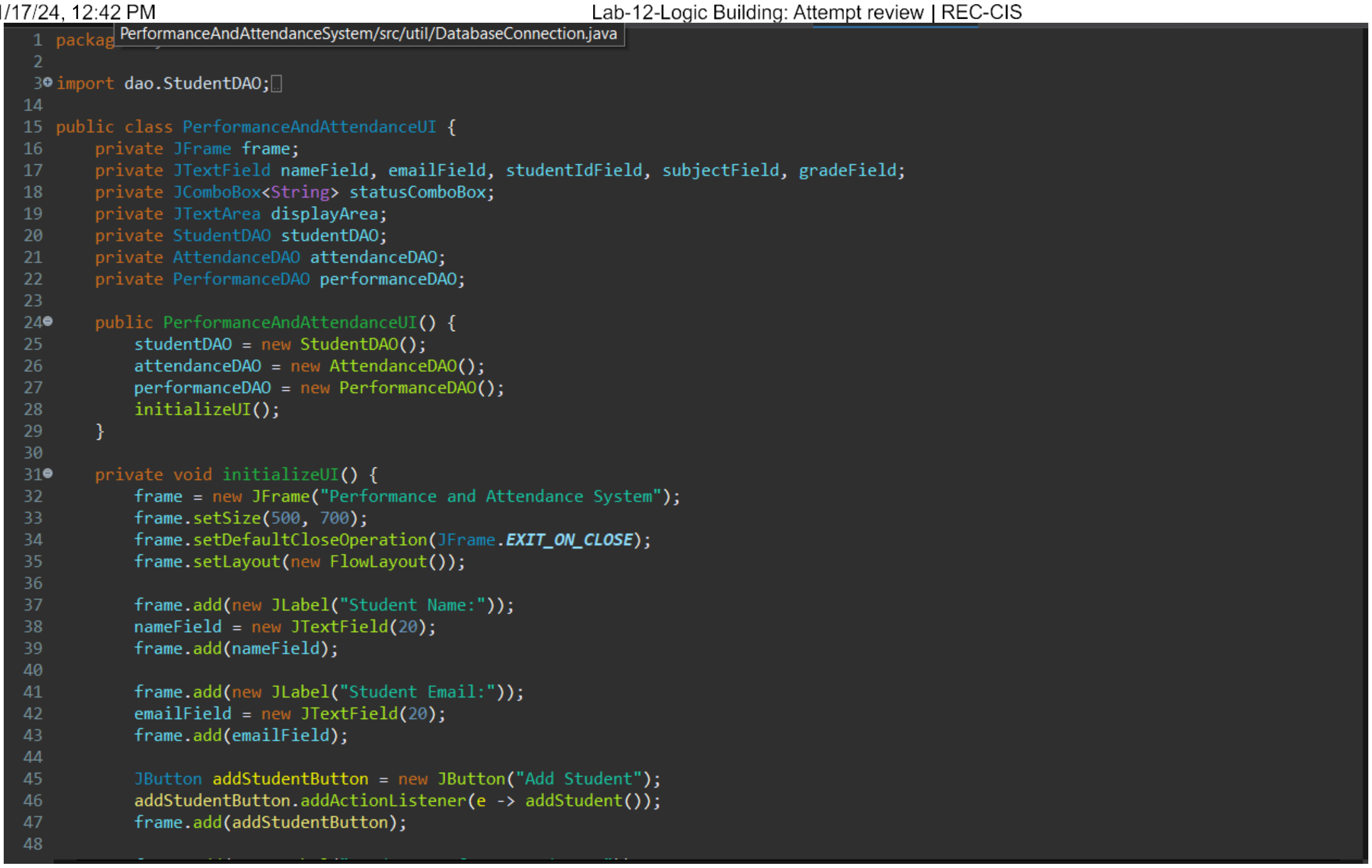
```
package model;

public class Performance {
    private int id;
    private int studentId;
    private String subject;
    private String grade;

    public Performance(int id, int studentId, String subject, String grade) {
        this.id = id;
        this.studentId = studentId;
        this.subject = subject;
        this.grade = grade;
    }

    public int getId() {
        return id;
    }

    public int getStudentId() {
        return studentId;
    }

    public String getSubject() {
        return subject;
    }

    public String getGrade() {
        return grade;
    }
}
```



```java
package dao;

import model.Student;

public class StudentDAO {
    public void addStudent(Student student) {
        String query = "INSERT INTO Students (name, email) VALUES (?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setString(1, student.getName());
            stmt.setString(2, student.getEmail());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
 // StudentDAO.java
    public Student getStudentById(int studentId) {
        Student student = null;
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement("SELECT * FROM students WHERE id = ?")) {
            stmt.setInt(1, studentId);
            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                student = new Student(rs.getInt("id"), rs.getString("name"), rs.getString("email"));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return student;
    }


    public List<Student> getAllStudents() {
        List<Student> students = new ArrayList<>();
        String query = "SELECT * FROM Students";
        try (Connection conn = DatabaseConnection.getConnection();
             Statement stmt = conn.createStatement();
```

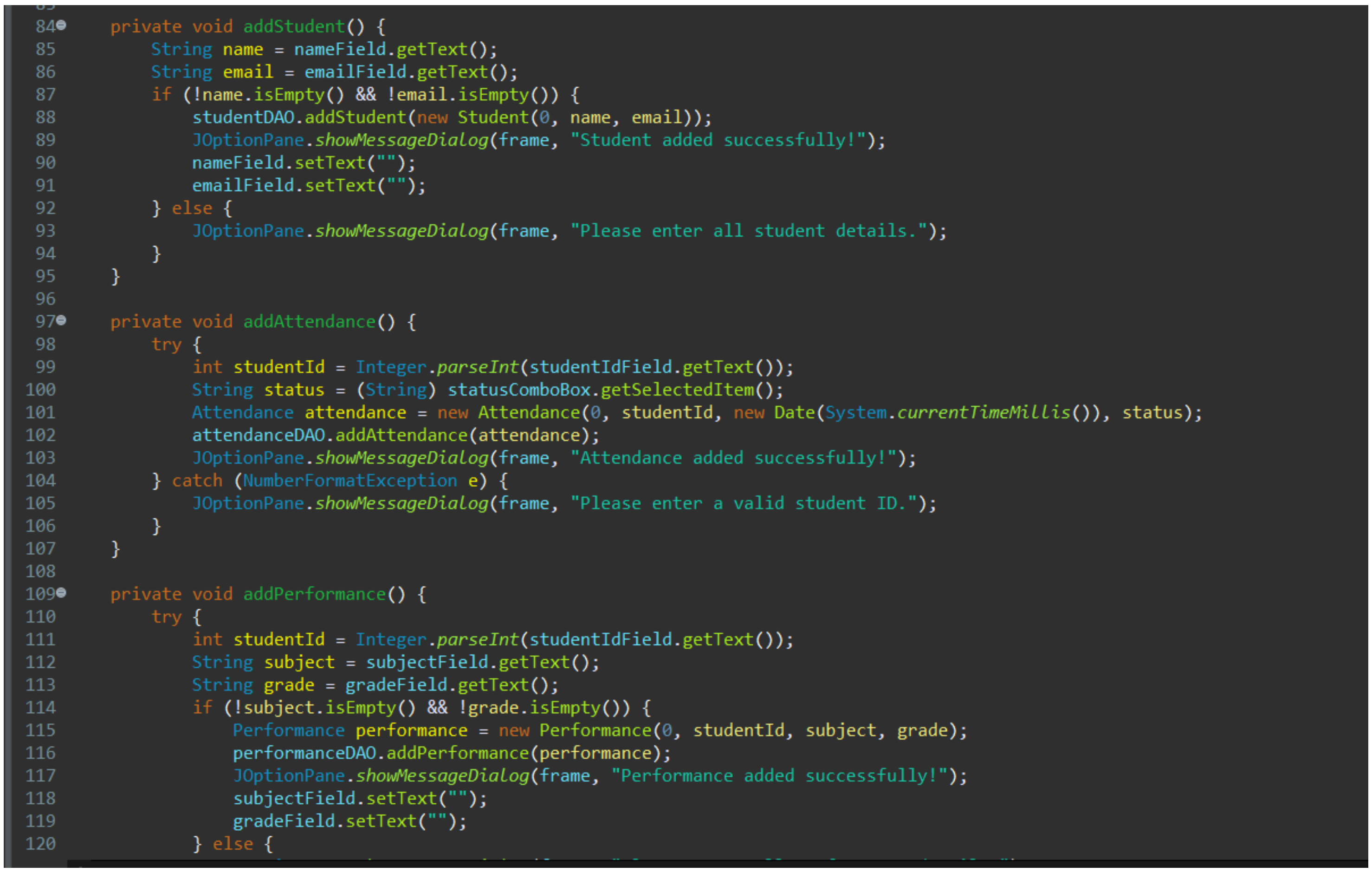


```java

    public List<Student> getAllStudents() {
        List<Student> students = new ArrayList<>();
        String query = "SELECT * FROM Students";
        try (Connection conn = DatabaseConnection.getConnection();
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(query)) {
            while (rs.next()) {
                Student student = new Student(rs.getInt("id"), rs.getString("name"), rs.getString("email"));
                students.add(student);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return students;
    }
}
```

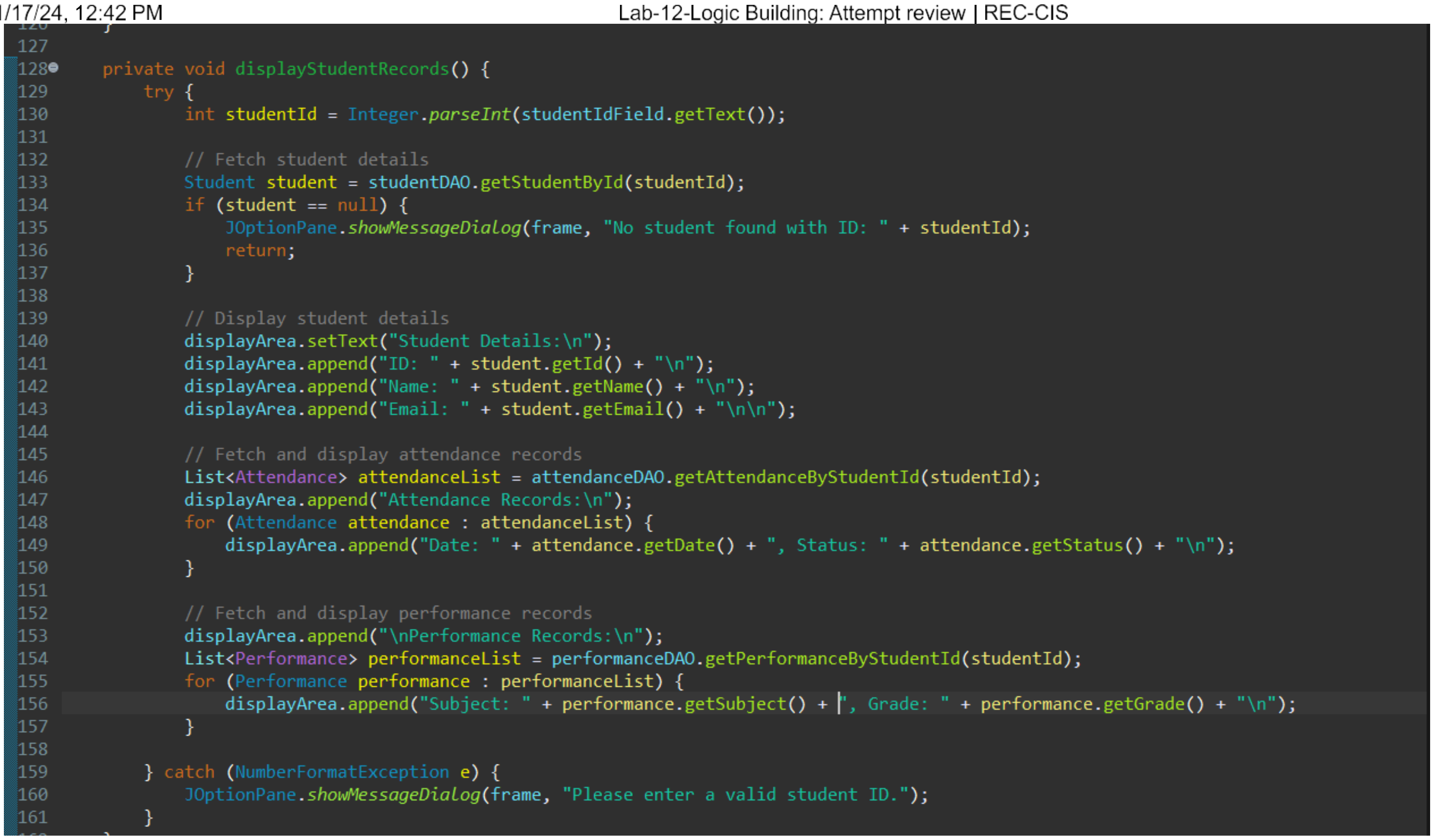


PerformanceAndAttendanceSystem/src/util/DatabaseConnection.java

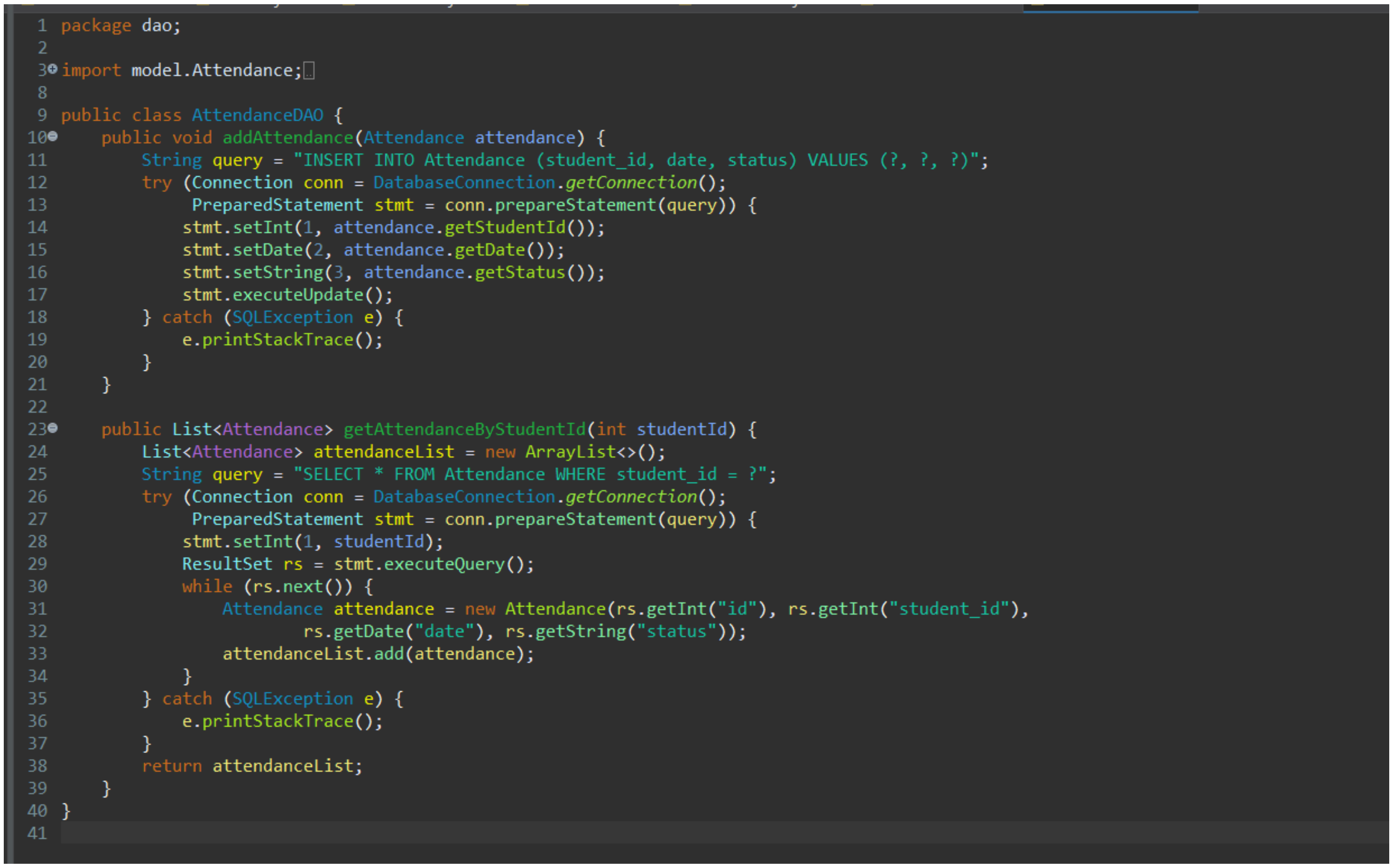
```java
packag

import dao.StudentDAO;

public class PerformanceAndAttendanceUI {
    private JFrame frame;
    private JTextField nameField, emailField, studentIdField, subjectField, gradeField;
    private JComboBox<String> statusComboBox;
    private JTextArea displayArea;
    private StudentDAO studentDAO;
    private AttendanceDAO attendanceDAO;
    private PerformanceDAO performanceDAO;

    public PerformanceAndAttendanceUI() {
        studentDAO = new StudentDAO();
        attendanceDAO = new AttendanceDAO();
        performanceDAO = new PerformanceDAO();
        initializeUI();
    }

    private void initializeUI() {
        frame = new JFrame("Performance and Attendance System");
        frame.setSize(500, 700);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new FlowLayout());

        frame.add(new JLabel("Student Name:"));
        nameField = new JTextField(20);
        frame.add(nameField);

        frame.add(new JLabel("Student Email:"));
        emailField = new JTextField(20);
        frame.add(emailField);

        JButton addStudentButton = new JButton("Add Student");
        addStudentButton.addActionListener(e -> addStudent());
        frame.add(addStudentButton);

```



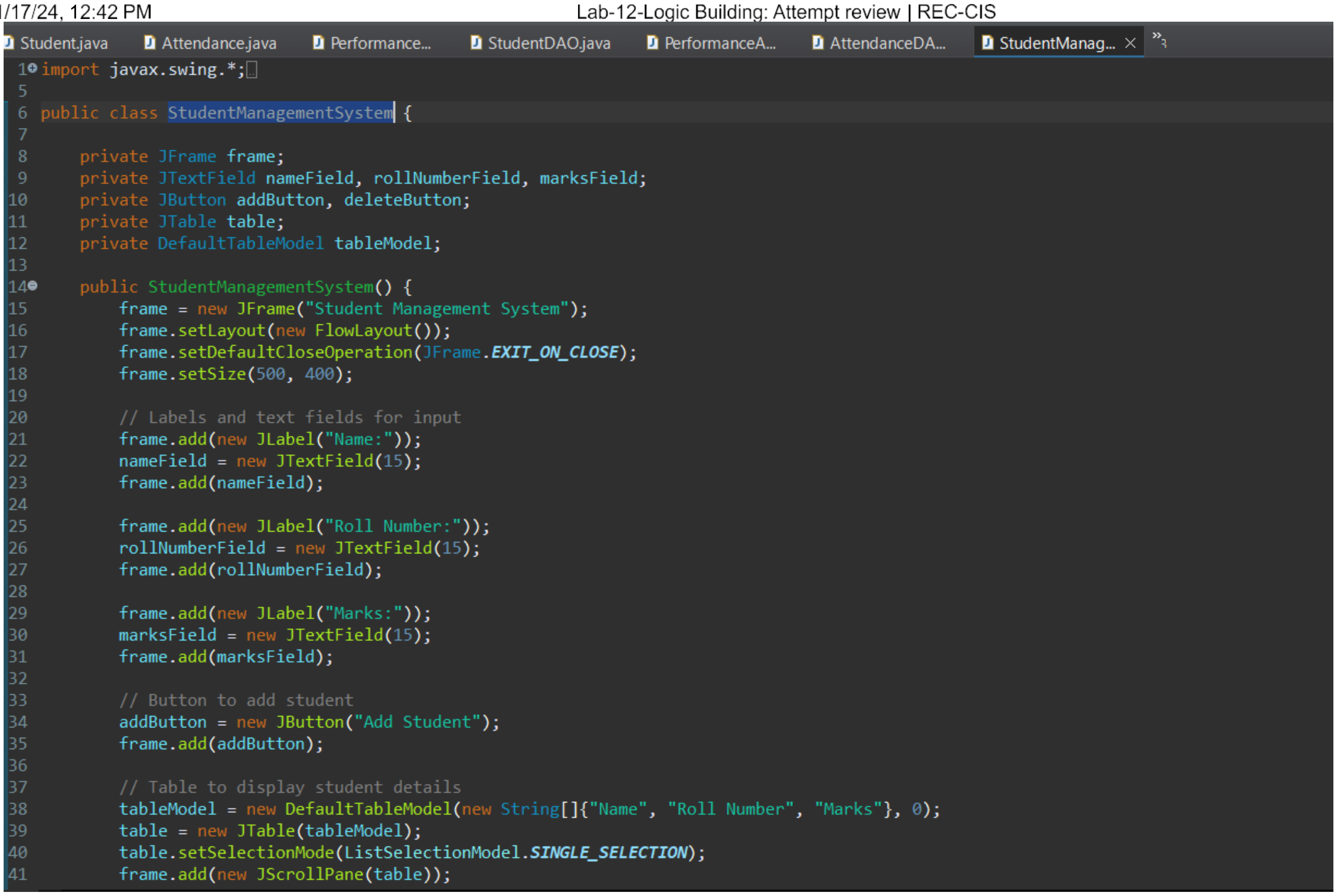
```java
    private void addStudent() {
        String name = nameField.getText();
        String email = emailField.getText();
        if (!name.isEmpty() && !email.isEmpty()) {
            studentDAO.addStudent(new Student(0, name, email));
            JOptionPane.showMessageDialog(frame, "Student added successfully!");
            nameField.setText("");
            emailField.setText("");
        } else {
            JOptionPane.showMessageDialog(frame, "Please enter all student details.");
        }
    }

    private void addAttendance() {
        try {
            int studentId = Integer.parseInt(studentIdField.getText());
            String status = (String) statusComboBox.getSelectedItem();
            Attendance attendance = new Attendance(0, studentId, new Date(System.currentTimeMillis()), status);
            attendanceDAO.addAttendance(attendance);
            JOptionPane.showMessageDialog(frame, "Attendance added successfully!");
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(frame, "Please enter a valid student ID.");
        }
    }

    private void addPerformance() {
        try {
            int studentId = Integer.parseInt(studentIdField.getText());
            String subject = subjectField.getText();
            String grade = gradeField.getText();
            if (!subject.isEmpty() && !grade.isEmpty()) {
                Performance performance = new Performance(0, studentId, subject, grade);
                performanceDAO.addPerformance(performance);
                JOptionPane.showMessageDialog(frame, "Performance added successfully!");
                subjectField.setText("");
                gradeField.setText("");
            } else {
```



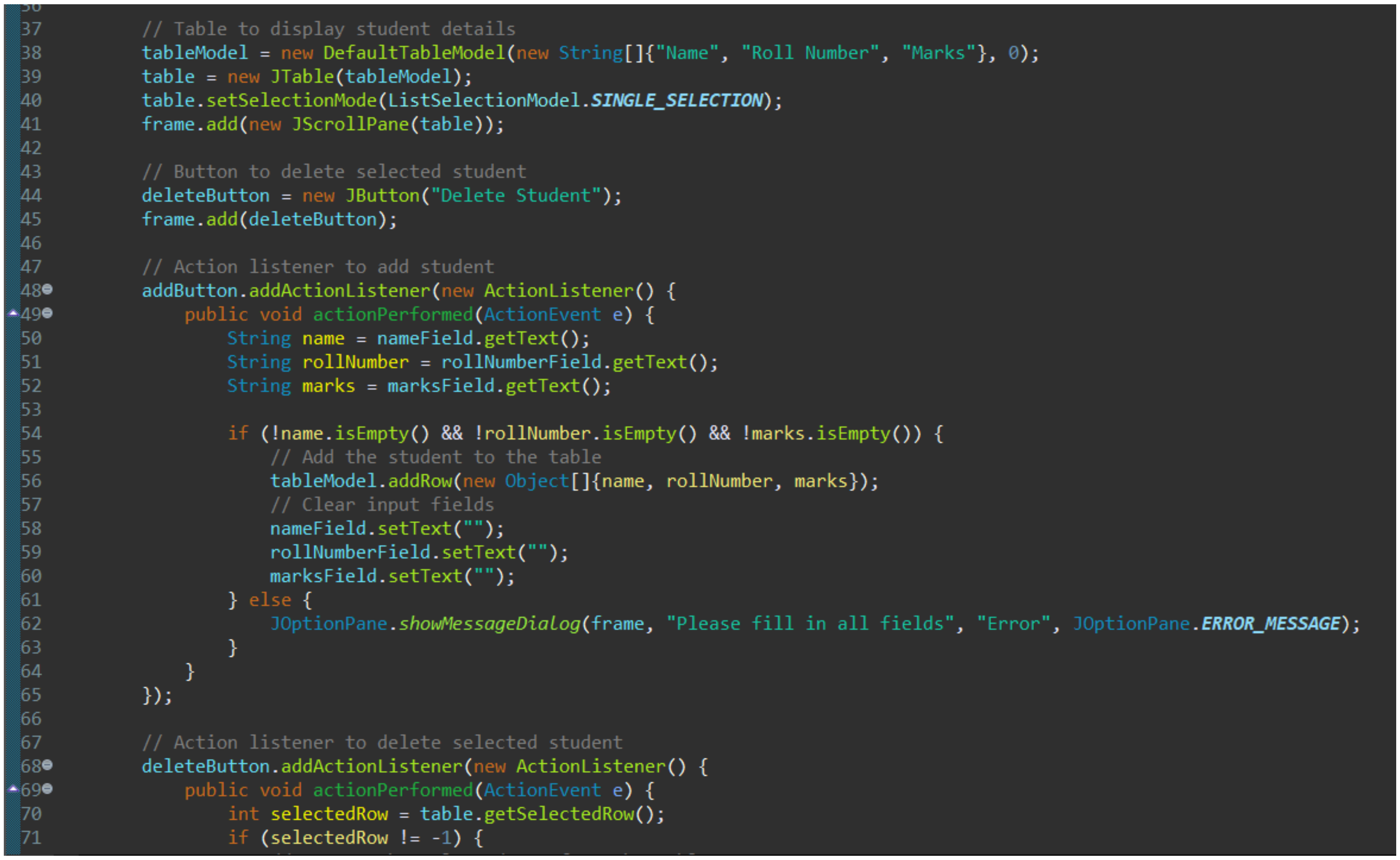
```java
    private void displayStudentRecords() {
        try {
            int studentId = Integer.parseInt(studentIdField.getText());

            // Fetch student details
            Student student = studentDAO.getStudentById(studentId);
            if (student == null) {
                JOptionPane.showMessageDialog(frame, "No student found with ID: " + studentId);
                return;
            }

            // Display student details
            displayArea.setText("Student Details:\n");
            displayArea.append("ID: " + student.getId() + "\n");
            displayArea.append("Name: " + student.getName() + "\n");
            displayArea.append("Email: " + student.getEmail() + "\n\n");

            // Fetch and display attendance records
            List<Attendance> attendanceList = attendanceDAO.getAttendanceByStudentId(studentId);
            displayArea.append("Attendance Records:\n");
            for (Attendance attendance : attendanceList) {
                displayArea.append("Date: " + attendance.getDate() + ", Status: " + attendance.getStatus() + "\n");
            }

            // Fetch and display performance records
            displayArea.append("\nPerformance Records:\n");
            List<Performance> performanceList = performanceDAO.getPerformanceByStudentId(studentId);
            for (Performance performance : performanceList) {
                displayArea.append("Subject: " + performance.getSubject() + ", Grade: " + performance.getGrade() + "\n");
            }

        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(frame, "Please enter a valid student ID.");
        }
```



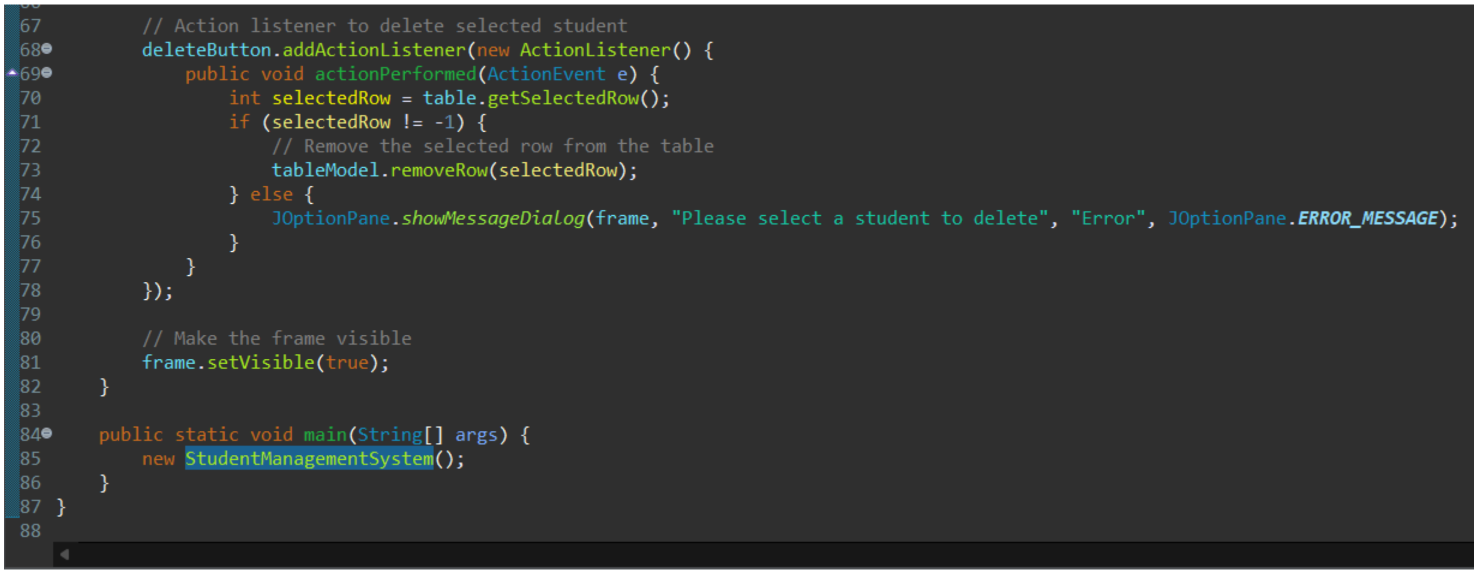
```java
package dao;

import model.Attendance;

public class AttendanceDAO {
    public void addAttendance(Attendance attendance) {
        String query = "INSERT INTO Attendance (student_id, date, status) VALUES (?, ?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setInt(1, attendance.getStudentId());
            stmt.setDate(2, attendance.getDate());
            stmt.setString(3, attendance.getStatus());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public List<Attendance> getAttendanceByStudentId(int studentId) {
        List<Attendance> attendanceList = new ArrayList<>();
        String query = "SELECT * FROM Attendance WHERE student_id = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setInt(1, studentId);
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                Attendance attendance = new Attendance(rs.getInt("id"), rs.getInt("student_id"),
                        rs.getDate("date"), rs.getString("status"));
                attendanceList.add(attendance);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return attendanceList;
    }
}
```



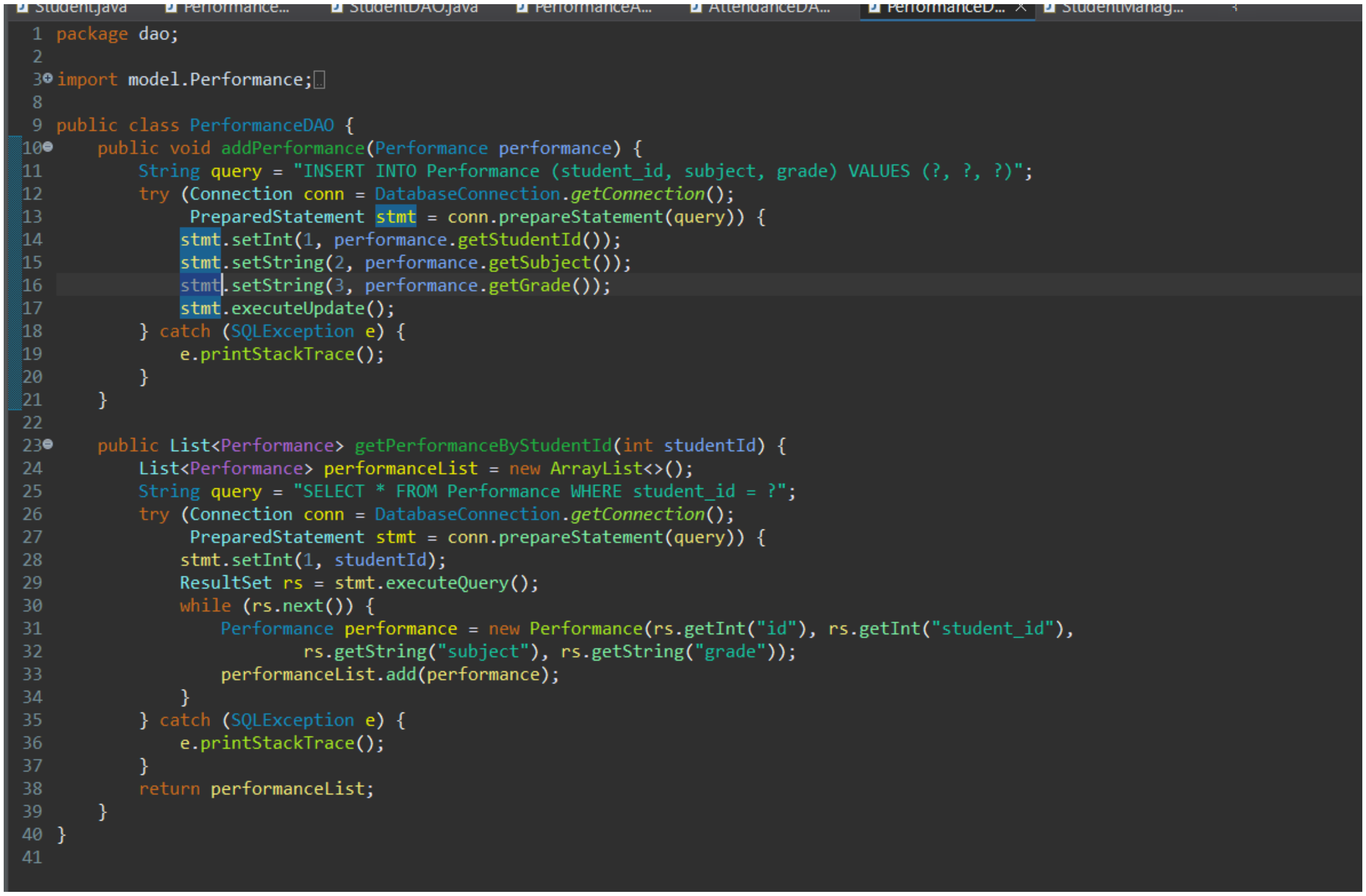
```java
import javax.swing.*;

public class StudentManagementSystem {

    private JFrame frame;
    private JTextField nameField, rollNumberField, marksField;
    private JButton addButton, deleteButton;
    private JTable table;
    private DefaultTableModel tableModel;

    public StudentManagementSystem() {
        frame = new JFrame("Student Management System");
        frame.setLayout(new FlowLayout());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 400);

        // Labels and text fields for input
        frame.add(new JLabel("Name:"));
        nameField = new JTextField(15);
        frame.add(nameField);

        frame.add(new JLabel("Roll Number:"));
        rollNumberField = new JTextField(15);
        frame.add(rollNumberField);

        frame.add(new JLabel("Marks:"));
        marksField = new JTextField(15);
        frame.add(marksField);

        // Button to add student
        addButton = new JButton("Add Student");
        frame.add(addButton);

        // Table to display student details
        tableModel = new DefaultTableModel(new String[]{"Name", "Roll Number", "Marks"}, 0);
        table = new JTable(tableModel);
        table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        frame.add(new JScrollPane(table));
```



```java
        // Table to display student details
        tableModel = new DefaultTableModel(new String[]{"Name", "Roll Number", "Marks"}, 0);
        table = new JTable(tableModel);
        table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        frame.add(new JScrollPane(table));

        // Button to delete selected student
        deleteButton = new JButton("Delete Student");
        frame.add(deleteButton);

        // Action listener to add student
        addButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String name = nameField.getText();
                String rollNumber = rollNumberField.getText();
                String marks = marksField.getText();

                if (!name.isEmpty() && !rollNumber.isEmpty() && !marks.isEmpty()) {
                    // Add the student to the table
                    tableModel.addRow(new Object[]{name, rollNumber, marks});
                    // Clear input fields
                    nameField.setText("");
                    rollNumberField.setText("");
                    marksField.setText("");
                } else {
                    JOptionPane.showMessageDialog(frame, "Please fill in all fields", "Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        });

        // Action listener to delete selected student
        deleteButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int selectedRow = table.getSelectedRow();
                if (selectedRow != -1) {
```



```
        // Action listener to delete selected student
        deleteButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int selectedRow = table.getSelectedRow();
                if (selectedRow != -1) {
                    // Remove the selected row from the table
                    tableModel.removeRow(selectedRow);
                } else {
                    JOptionPane.showMessageDialog(frame, "Please select a student to delete", "Error", JOptionPane.ERROR_MESSAGE);
                }
            }
        });

        // Make the frame visible
        frame.setVisible(true);
    }

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



```
package dao;

import model.Performance;

public class PerformanceDAO {
    public void addPerformance(Performance performance) {
        String query = "INSERT INTO Performance (student_id, subject, grade) VALUES (?, ?, ?)";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setInt(1, performance.getStudentId());
            stmt.setString(2, performance.getSubject());
            stmt.setString(3, performance.getGrade());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public List<Performance> getPerformanceByStudentId(int studentId) {
        List<Performance> performanceList = new ArrayList<>();
        String query = "SELECT * FROM Performance WHERE student_id = ?";
        try (Connection conn = DatabaseConnection.getConnection();
             PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setInt(1, studentId);
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                Performance performance = new Performance(rs.getInt("id"), rs.getInt("student_id"),
                        rs.getString("subject"), rs.getString("grade"));
                performanceList.add(performance);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return performanceList;
    }
}
```