

重 庆 大 学

学 生 实 验 报 告

实验课程名称 JAVA EE 程序设计

开课实验室 DS1501

学 院 大数据与软件学院 年级 2021 级 专业班 软件工程 X 班

学 生 姓 名 XXX 学 号 2021XXXX

开 课 时 间 2022 至 2023 学年第 2 学期

成 绩	
教师签名	

大数据与软件学院制

《JAVA EE 程序设计》实验报告

开课实验室：DS1501

2023 年 4 月 2 日

学院	大数据与软件学 院	年级、专业、班	2021 级软件工 程 X 班	姓名	XXX	成绩	
课程 名称	JAVA EE 程序设计		实验项目 名 称	实验7-8:事件编程、 数据库应用编程。		指导教师	XX

教
师
评

教师签名：

2022 年 月 日

一、实验目的

实验七：基本掌握使用 JAVA 来完成基本的图形界面等调试与编程，事件编程调试。

实验八：掌握利用JAVA完成数据库课程中的基本数据记录操作。

二、实验内容

实验内容：

利用图形界面编程，开发简易C/S模式的管理信息系统，题目自拟，实现数据库访问操作。完成SQL SERVER或MYSQL数据库管理系统的安装，配置，JAVA数据库访问环境的配置；完成数据库表的建立，记录插入等；建立用户表，包含用户名、密码字段；利用JAVA实现图形界面，用户登录验证，进入软件运行后，针对某表记录的增、删、改、查操作。

三、使用仪器、材料

OpenJDK JDK 19.0.2

Tomcat

MySql 8.0.30

MySQL Workbench 8.0 CE

Lenovo Legion R9000P2021H

Windows 11

四、实验过程原始记录(数据、图表、计算等):

本次实验使用 Java 语言链接数据库，设计了一个学生信息管理系统。用户可以通过账号（学号）和密码登录该系统，并可以通过该系统进行添加、删除、查找和修改学生信息等操作。

```
PS F:\CQU\Java\Test\test4\src\EX3> cloc .
    8 text files.
    8 unique files.
    0 files ignored.
```

github.com/AlDanial/cloc v 1.96 T=0.10 s (83.7 files/s, 8287.2 lines/s)

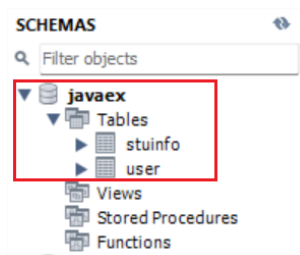
Language	files	blank	comment	code
Java	8	98	118	576
SUM:	8	98	118	576

项目规模：576 行（不计空行）；

运行结果：

Part 1: 实验中使用到的数据库;

1.1 JavaEX(schemas);



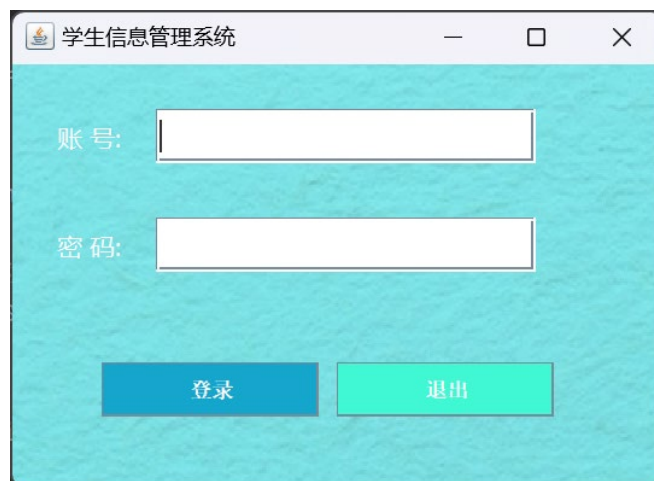
1.2 User: 存储拥有登录系统权限的用户及其密码;

[illegible]

[illegible][illegible]

Part 2: 登录界面：验证用户权限；

2.1 登录界面；



The screenshot shows a window titled "学生信息管理系统" (Student Information Management System). It features a light blue background. There are two input fields: "账号:" (Account) and "密码:" (Password). Below these fields are two buttons: "登录" (Login) in blue and "退出" (Exit) in green.

拥有两个文本框可以分别输入账号（学号）和密码，下有“登录”与“退出”两个按钮；

2.2 成功登录；



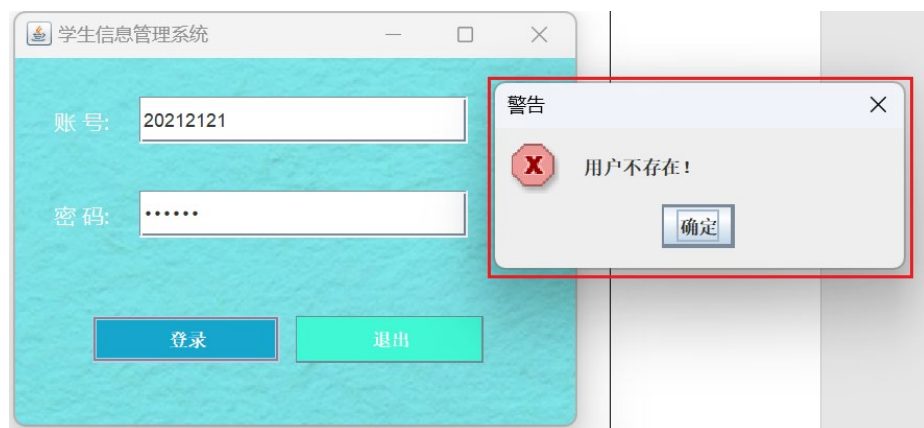
The screenshot shows the same login window as before, but the "账号:" field is now filled with a blurred text and the "密码:" field is filled with dots. The "登录" button is highlighted with a red border, indicating it has been clicked.

输入正确的账号和与之配对的密码，点击“登录”按钮后，进入主界面：

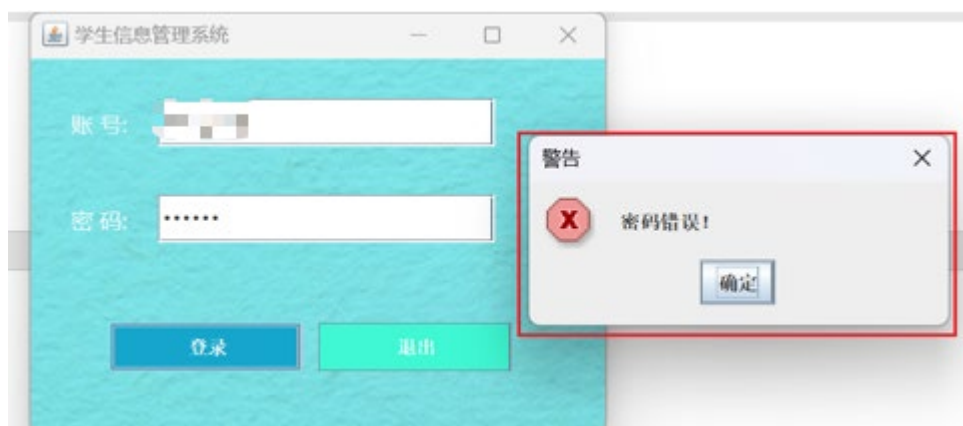


The screenshot shows the main interface of the system. It has a light blue background. At the top, there is a menu bar with the following options: "添加" (Add), "查询" (Query), "修改" (Modify), "删除" (Delete), and "退出" (Exit). In the center, there is a large orange text that says "欢迎登录本系统~" (Welcome to log in to this system~).

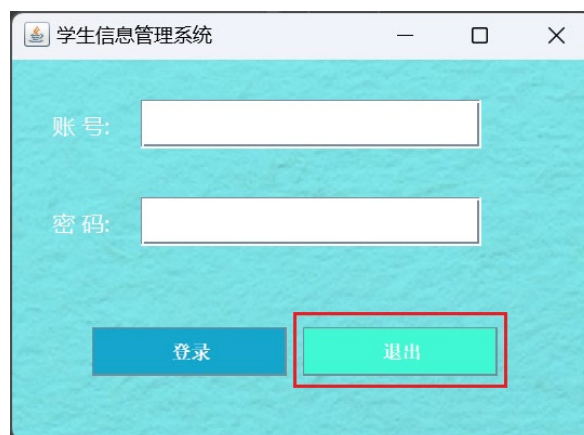
2.3 用户不存在的情况；



2.4 密码错误的情况；



2.5 点击“退出”按钮：关闭窗口；



Part 3 主界面：对数据库进行增删改查操作；

3.1 主界面；



本系统的主界面主要由菜单栏和欢迎词界面构成，菜单栏有五个部件，分别对应添加学生信息、查询学生信息、删除学生信息、修改学生信息、退出系统的功能；

3.1.1 主界面-菜单栏-1



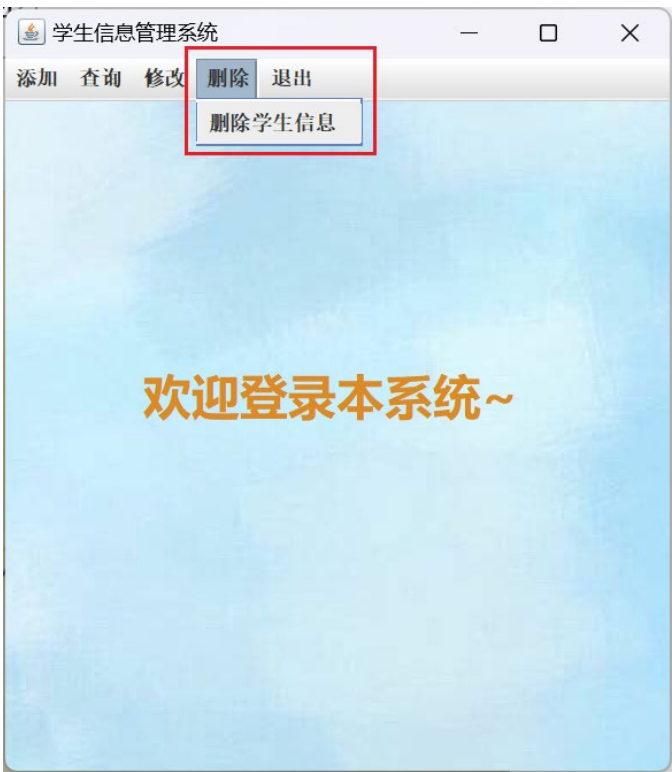
3.1.2 主界面-菜单栏-2



3.1.3 主界面-菜单栏-3



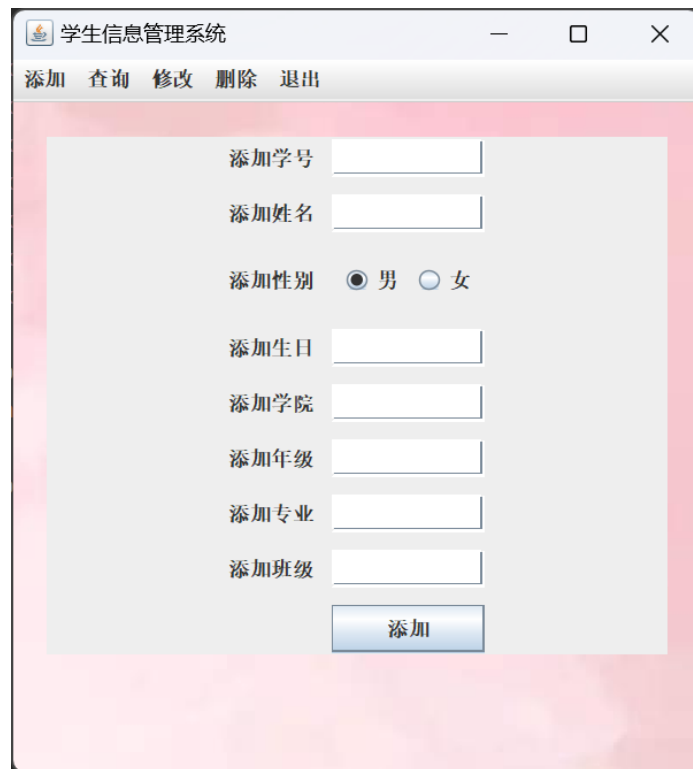
3.1.4 主界面-菜单栏-4



3.1.5 主界面-菜单栏-5;



3.2 添加学生信息：该界面用于添加学生信息，可添加信息包括学号、姓名、性别、生日、学院、年级、专业、班级共八样信息，其中因为学号为主键，所以不能为空且不能与已有学号重复；



The screenshot shows a window titled "学生信息管理系统" (Student Information Management System). The menu bar contains "添加" (Add), "查询" (Query), "修改" (Modify), "删除" (Delete), and "退出" (Exit). The main area has a light gray background with a pink border. It contains the following fields and controls:

- 添加学号 (Add ID): A text input field.
- 添加姓名 (Add Name): A text input field.
- 添加性别 (Add Gender): Radio buttons for "男" (Male) and "女" (Female), with "男" selected.
- 添加生日 (Add Birthday): A text input field.
- 添加学院 (Add College): A text input field.
- 添加年级 (Add Grade): A text input field.
- 添加专业 (Add Major): A text input field.
- 添加班级 (Add Class): A text input field.
- 添加 (Add): A blue button at the bottom right.

添入需要添加的信息后，点击“添加”按钮即可添加成功；

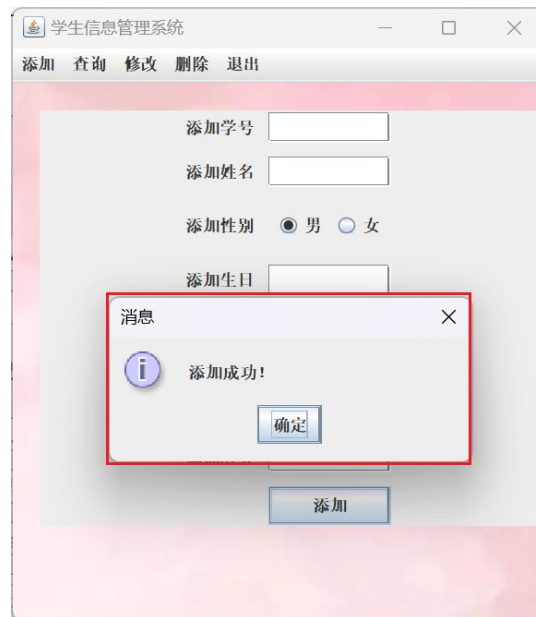
3.2.1 输入需要添加的信息；



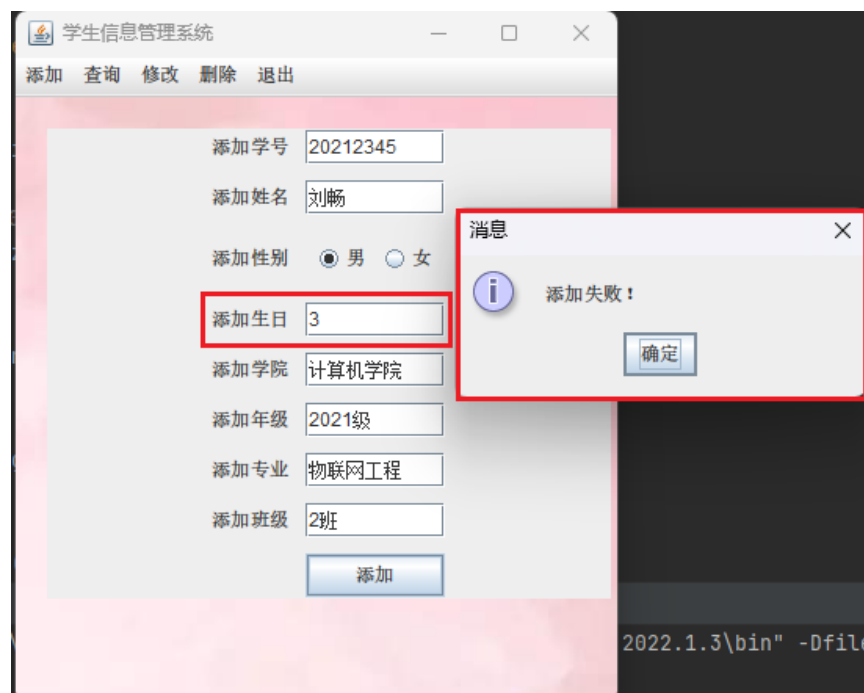
This screenshot shows the same window as the previous one, but with sample data entered into the fields. The "添加" button is highlighted with a red rectangle.

- 添加学号: [blurred]
- 添加姓名: [blurred]
- 添加性别: 男 (selected)
- 添加生日: [blurred]
- 添加学院: 数据与软件学院
- 添加年级: 2021级
- 添加专业: 软件工程
- 添加班级: [blurred]
- 添加: (highlighted with a red rectangle)

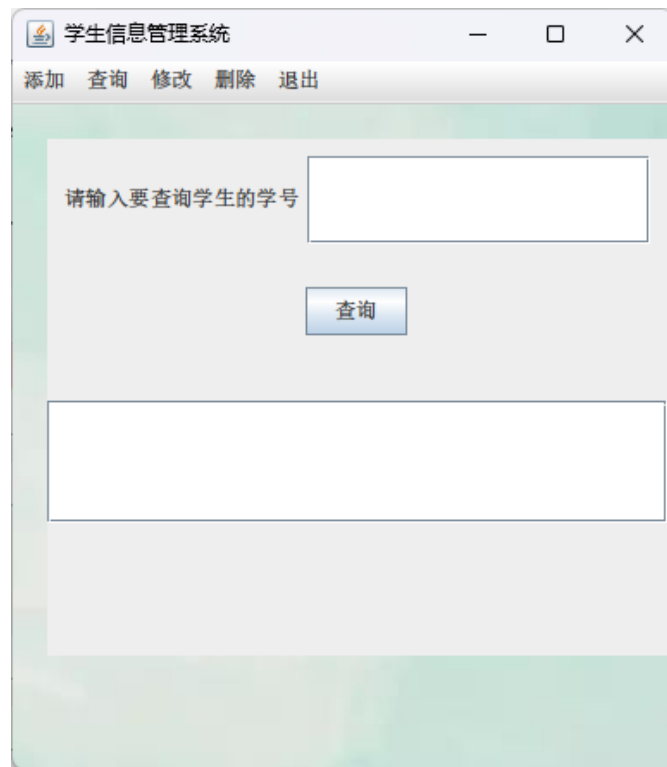
3.2.2 成功添加及其效果；



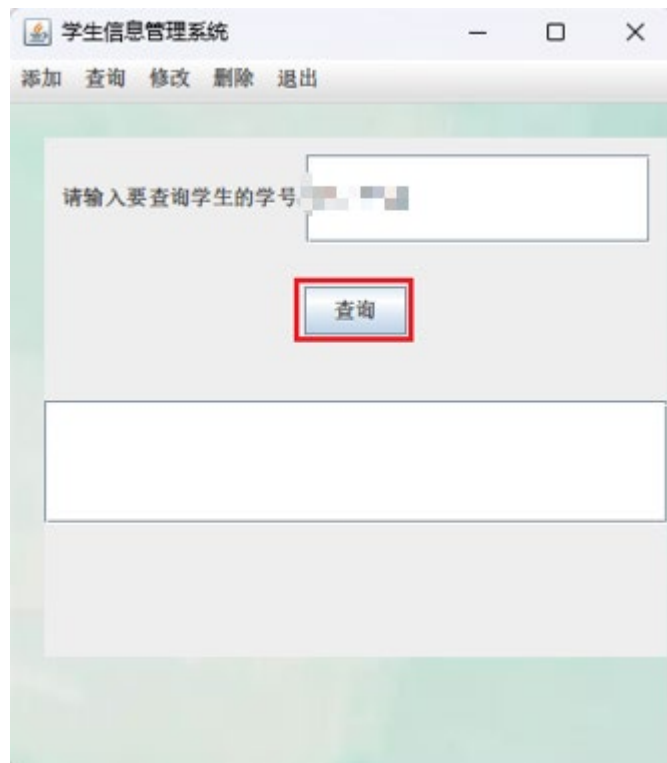
3.2.3 添加信息不满足用户自定义约束；



3.3 查询学生信息：该界面用于查询学生信息，可通过学号查询目标学生存储在数据库中的信息，包括学号、姓名、性别、生日、学院、年级、专业、班级共八样信息；

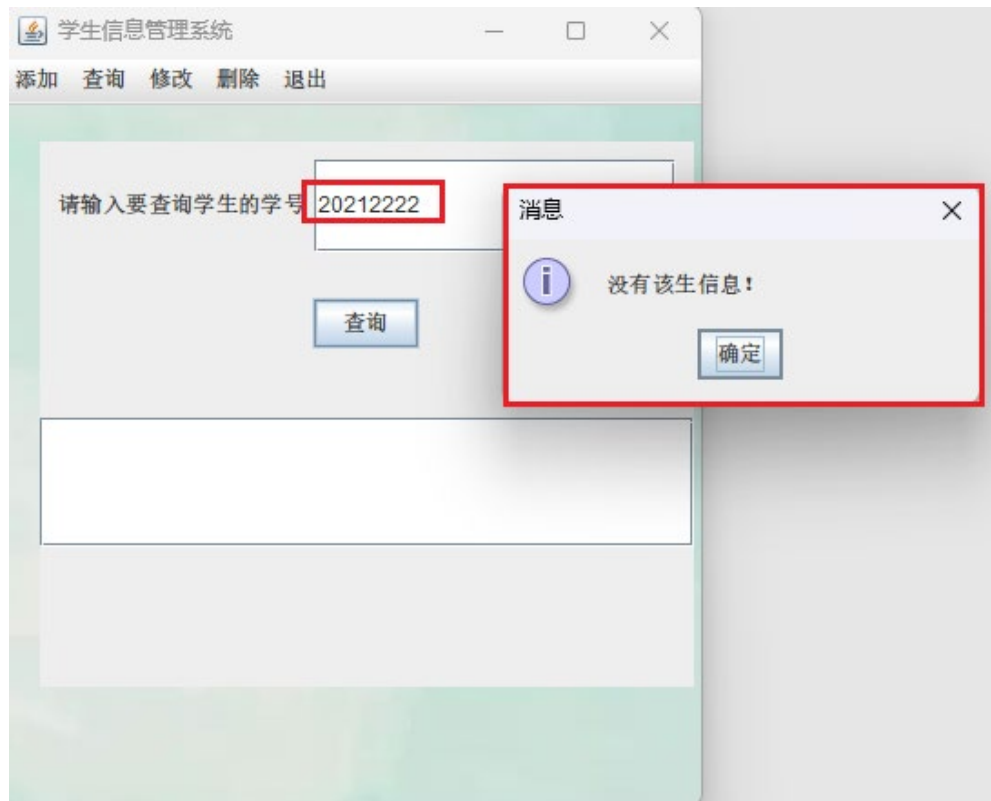


3.3.1 输入要查询的目标学生的学号；

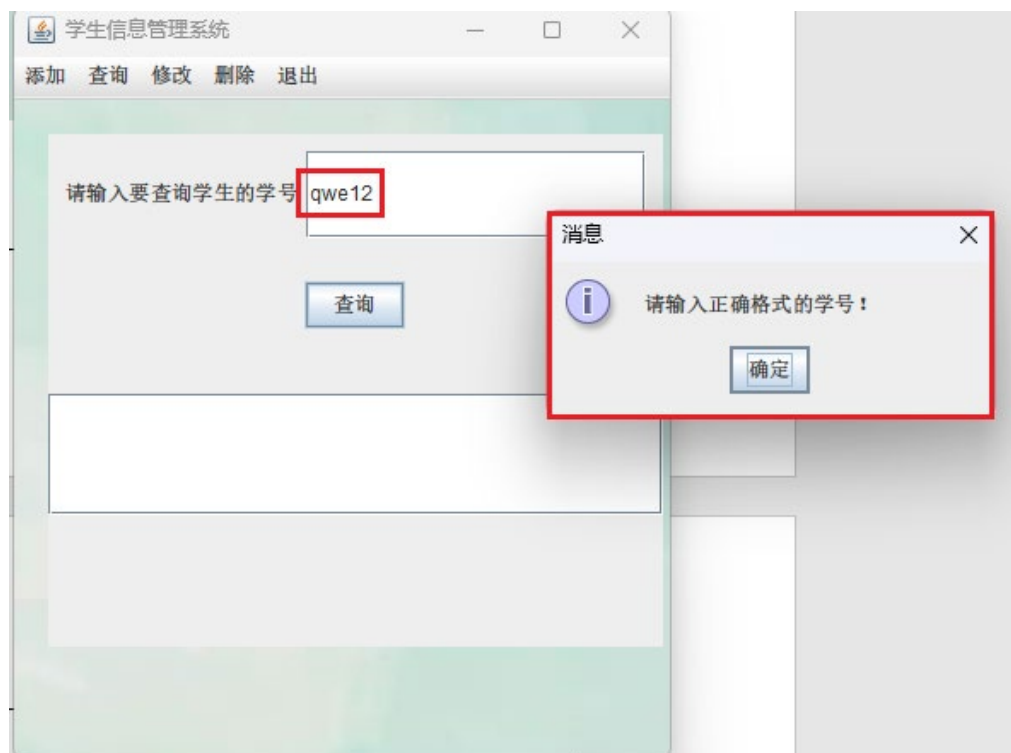


3.3.2 输入正确的学号查询成功及其效果；

3.3.3 查询不存在的学生信息；



3.3.4 输入错误格式的学号；



3.4 修改学生信息：该界面用于修改学生信息，可通过学号唯一修改目标学生存储在数据库中的信息，包括姓名、性别、生日、学院、年级、专业、班级这七样信息；

The screenshot shows a window titled '学生信息管理系统' (Student Information Management System). Below the title bar is a menu bar with '添加' (Add), '查询' (Query), '修改' (Modify), '删除' (Delete), and '退出' (Exit). The main area contains a form with the following fields and controls:

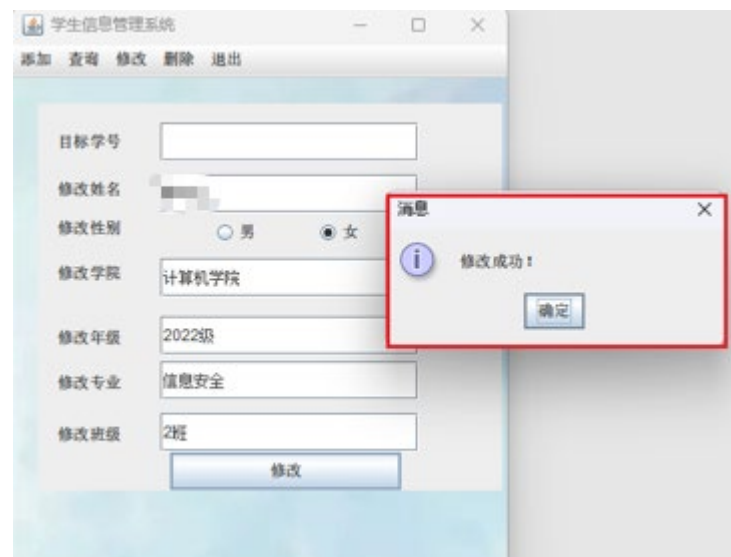
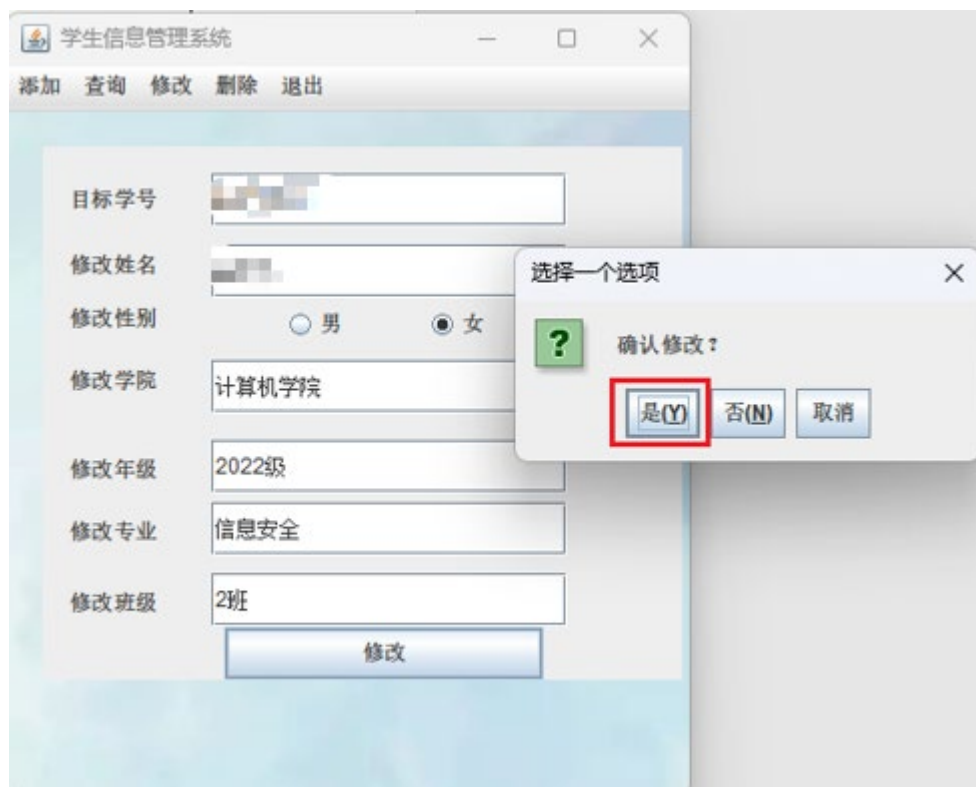
- 目标学号 (Target ID): A text input field.
- 修改姓名 (Modify Name): A text input field.
- 修改性别 (Modify Gender): Radio buttons for '男' (Male) and '女' (Female). The '男' button is selected.
- 修改学院 (Modify College): A text input field.
- 修改年级 (Modify Grade): A text input field.
- 修改专业 (Modify Major): A text input field.
- 修改班级 (Modify Class): A text input field.
- 修改 (Modify): A button at the bottom right of the form.

3.4.1 输入目标学生学号及要修改的信息；

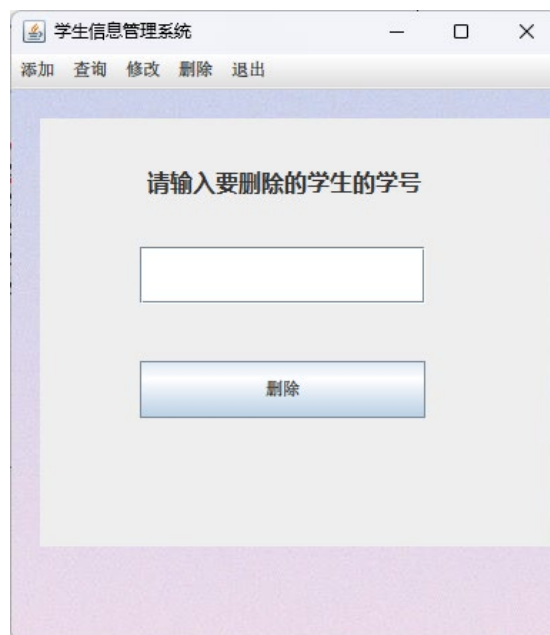
This screenshot shows the same 'Student Information Management System' window, but with data entered into the form fields. The '修改' (Modify) button is highlighted with a red rectangle.

- 目标学号 (Target ID): A text input field containing a blurred value.
- 修改姓名 (Modify Name): A text input field containing a blurred value.
- 修改性别 (Modify Gender): Radio buttons for '男' (Male) and '女' (Female). The '女' button is selected.
- 修改学院 (Modify College): A text input field containing '计算机学院' (Computer College).
- 修改年级 (Modify Grade): A text input field containing '2022级' (2022 Grade).
- 修改专业 (Modify Major): A text input field containing '信息安全' (Information Security).
- 修改班级 (Modify Class): A text input field containing '2班' (Class 2).
- 修改 (Modify): A button at the bottom right of the form, highlighted with a red rectangle.

3.4.2 修改成功及其效果；



3.5 删除学生信息：该界面用于删除学生信息，可通过学号唯一删除目标学生存储在数据库中的信息，包括学号、姓名、性别、生日、学院、年级、专业、班级共八样信息；



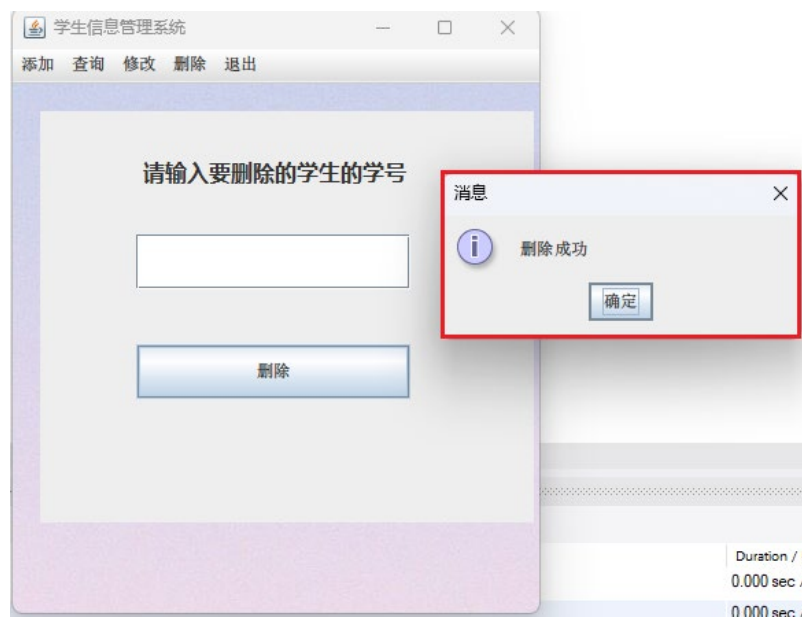
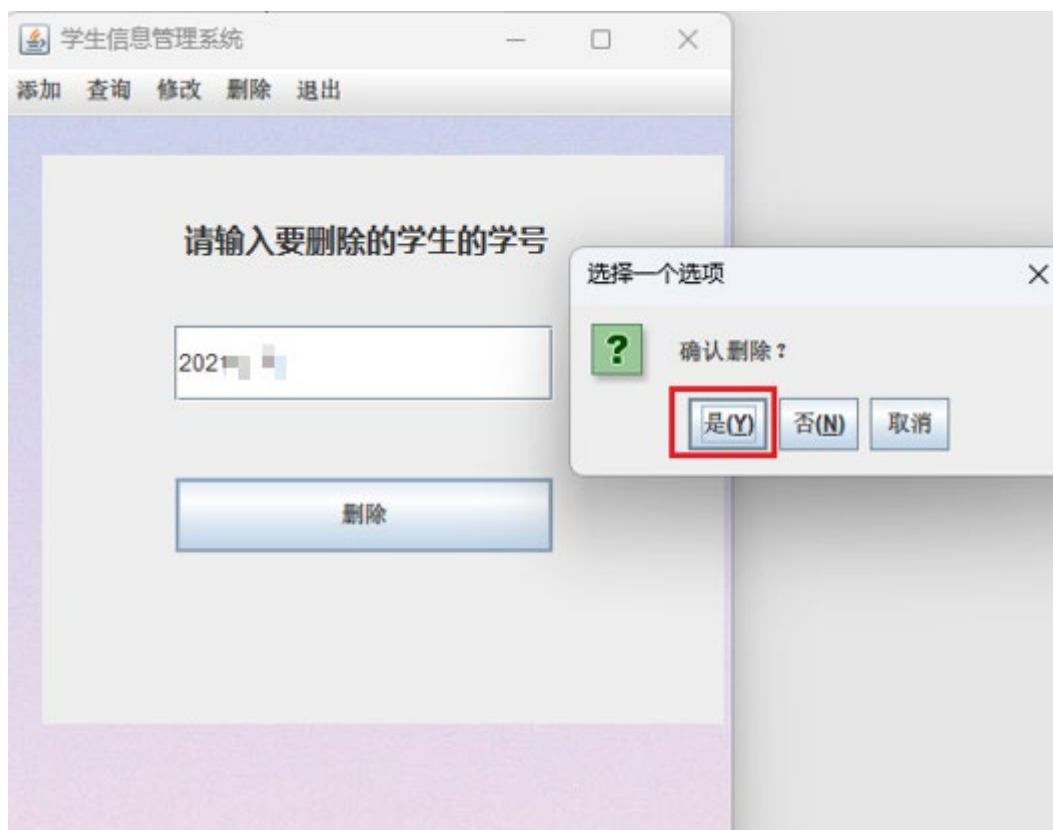
The screenshot shows a window titled '学生信息管理系统' (Student Information Management System). The menu bar contains '添加' (Add), '查询' (Query), '修改' (Modify), '删除' (Delete), and '退出' (Exit). The main area has a prompt '请输入要删除的学生的学号' (Please enter the student ID to be deleted). Below the prompt is a text input field and a blue button labeled '删除' (Delete).

3.5.1 输入要删除的学生的学号；



This screenshot is identical to the previous one, but with a red rectangular box highlighting the '删除' (Delete) button. The text input field now contains a few characters, indicating that a student ID has been entered.

3.5.2 删除成功及其效果；



3.6 退出系统：点击即可关闭该学生信息管理系统主界面；



核心代码：（共 8 个部分，详情请见项目压缩包）

1. Main （主函数）

```
//导入必要的类
package EX3;

import java.awt.*; // 导入 AWT 组件
import java.sql.*; // 导入 JDBC 用于数据库连接和操作的相关类

//定义主类
public class Main {
    public static void main(String[] args) throws SQLException {

        //创建登录对象
        LoginGUI log_in=new LoginGUI();

        //获取屏幕尺寸
        Dimension screenSize=Toolkit.getDefaultToolkit().getScreenSize();

        //设置登录窗口尺寸为屏幕尺寸的0.7倍，保证在大多数屏幕上都能正常显示
        Dimension loginSize = new Dimension((int)(screenSize.width*0.7), (int)(screenSize.height*0.7));

        //将登录窗口位置设为屏幕中心
        log_in.setLocationRelativeTo(null);

        //设置登录窗口尺寸
        log_in.setSize(loginSize);

        //将登录窗口显示出来
        log_in.setVisible(true);
    }
}
```

2. ConnecttoDB （连接至数据库）

```
//导入相关的包
package EX3;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConnecttoDB {

    private static final String DRIVER_CLASS_NAME = "com.mysql.cj.jdbc.Driver";
    private static final String CONNECTION_URL_TEMPLATE = "jdbc:mysql://localhost/%s?serverTimezone=GMT%2B8&useUnicode=true&useSSL=true&characterEncoding=utf8";

    /**
     * connectDB 方法用于连接数据库
     * @param DBName 数据库名称
     * @param id 数据库登录用户名
     * @param pwd 数据库登录密码
     * @return 返回连接对象
     */
    public static Connection connectDB(String DBName, String id, String pwd) {
        Connection connection = null;
        try {
            // 加载MySQL驱动
            Class.forName(DRIVER_CLASS_NAME);

            // 数据库连接URL
            String url = String.format(CONNECTION_URL_TEMPLATE, DBName);

            // 建立数据库连接
            connection = DriverManager.getConnection(url, id, pwd);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }

        // 返回连接对象
        return connection;
    }
}
```

3.LoginGUI （登录界面）

```
//导入相关的包
package EX3;

import javax.swing.*; // 导入 Swing 组件
import java.awt.*; // 导入 AWT 组件
import java.awt.event.*; // 导入事件处理的包
import java.sql.*; // 导入 JDBC 用于数据库连接和操作的相关类

public class LoginGUI extends JFrame implements ActionListener {
    //声明变量
    private JPanel pane;
    private JLabel label1, label2;
    private JTextField textField1;
    private JPasswordField passwordField1;
    private JButton button1, button2;
    private Connection connection;
    private PreparedStatement preparedStatement;

    public LoginGUI() {
        try {
            //连接数据库
            connection = ConnecttoDB.connectDB( DBName: "JavaEX", id: "root", pwd: "123456");
            setDefaultCloseOperation(EXIT_ON_CLOSE);
            init();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
private void init() throws Exception {
    //设置背景图片
    JLabel background = new JLabel(new ImageIcon( filename: "image/pic7.jpg"));
    background.setBounds( x: 0, y: 0, width: 360, height: 235);
    this.getLayeredPane().add(background, new Integer(Integer.MIN_VALUE));

    //设置面板
    pane = (JPanel) getContentPane();
    pane.setOpaque(false);
    pane.setLayout(null);

    //设置窗口属性
    setSize(new Dimension( width: 360, height: 210));
    setTitle("学生信息管理系统");

    //添加标签、文本框、按钮到面板中
    label1 = new JLabel( text: "账 号:");
    label1.setBounds(new Rectangle( x: 25, y: 25, width: 75, height: 30));
    label1.setFont(new Font( name: "微软雅黑", Font.PLAIN, size: 14));
    label1.setForeground(Color.WHITE);
    pane.add(label1);

    textField1 = new JTextField();
    textField1.setBounds(new Rectangle( x: 80, y: 25, width: 210, height: 30));
    pane.add(textField1);
}
```

```

label2 = new JLabel( text: "密 码:");
label2.setBounds(new Rectangle( x: 25, y: 85, width: 75, height: 30));
label2.setFont(new Font( name: "微软雅黑", Font.PLAIN, size: 14));
label2.setForeground(Color.WHITE);
pane.add(label2);

passwordField1 = new JPasswordField();
passwordField1.setBounds(new Rectangle( x: 80, y: 85, width: 210, height: 30));
pane.add(passwordField1);

button1 = new JButton( text: "登录");
button1.setBounds(new Rectangle( x: 50, y: 165, width: 120, height: 30));
button1.setForeground(Color.WHITE);
button1.setBackground(new Color( rgb: 0x15A6CC));
button1.addActionListener( l: this);
pane.add(button1);

button2 = new JButton( text: "退出");
button2.setBounds(new Rectangle( x: 180, y: 165, width: 120, height: 30));
button2.setForeground(Color.WHITE);
button2.setBackground(new Color( rgb: 0x40F8D4));
button2.addActionListener( l: this);
pane.add(button2);

setVisible(true);
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == button1) {
        String user = textField1.getText();
        String pwd = new String(passwordField1.getPassword());
        try {
            //查询数据库，获得用户和密码信息
            String sql = "SELECT ID,Password FROM user WHERE ID=?";
            preparedStatement = connection.prepareStatement(sql);
            preparedStatement.setString( parameterIndex: 1, user);
            ResultSet resultSet = preparedStatement.executeQuery();
            if (resultSet.next()) {
                String id = resultSet.getString( columnLabel: "ID");
                String password = resultSet.getString( columnLabel: "Password");
                if (password.equalsIgnoreCase(pwd)) {
                    //如果用户名和密码都正确，则打开主页面
                    setVisible(false);
                    MainPage add = new MainPage();
                    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
                    Dimension frameSize = add.getSize();
                    if (frameSize.height > screenSize.height) {
                        frameSize.height = screenSize.height;
                    }
                    if (frameSize.width > screenSize.width) {
                        frameSize.width = screenSize.width;
                    }
                    add.setLocation( x: (screenSize.width - frameSize.width) / 2, y: (screenSize.height - frameSize.height) / 2);
                    add.setVisible(true);
                } else {
                    //如果密码错误，则弹出对话框提示
                    JOptionPane.showMessageDialog( parentComponent: this, message: "密码错误!", title: "警告", JOptionPane.ERROR_MESSAGE);
                }
            } else {
                //如果用户名错误，则弹出对话框提示
                JOptionPane.showMessageDialog( parentComponent: this, message: "用户不存在!", title: "警告", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
}

```

```

    } catch (SQLException ex) {
        JOptionPane.showMessageDialog( parentComponent: this, message: "登陆失败!", title: "警告", JOptionPane.ERROR_MESSAGE);
    } finally {
        try {
            if (preparedStatement != null) {
                preparedStatement.close();
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
} else if (e.getSource() == button2) {
    //点击退出按钮，退出程序
    System.exit( status: 0);
}
}
}

```

4. MainPage （主界面）

```

// 导入相关的类
package EX3;

import javax.swing.*; // 导入 Swing 组件
import java.awt.*; // 导入 AWT 组件
import java.awt.event.*; // 导入事件处理的包
import java.sql.*; // 导入 JDBC 用于数据库连接和操作的相关类

// 定义一个类 MainPage，继承 JFrame 类
public class MainPage extends JFrame {
    // 定义成员变量
    JPanel pane; // 内容面板
    JMenuBar menuBar; // 导航栏
    JLabel label; // 标签

    // 定义构造方法，实例化对象时自动调用
    public MainPage() {
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        init(); // 调用 Init 方法
    }
}

```



```

// 定义 Init 方法，用来进行界面的初始化
private void init() {
    // 设置背景
    JLabel background = new JLabel(new ImageIcon( filename: "image/pic13.jpg"));
    background.setBounds( x: 0, y: 0, width: 400, height: 450);
    this.getLayeredPane().add(background, new Integer(Integer.MIN_VALUE));

    // 设置内容面板
    pane = (JPanel) getContentPane();
    pane.setOpaque(false); // 设置透明
    pane.setLayout(null); // 设置布局管理器为空

    // 设置窗口大小和标题
    setSize(new Dimension( width: 400, height: 450));
    setTitle("学生信息管理系统");

    // 创建导航栏
    menuBar = new JMenuBar();
    setJMenuBar(menuBar);

    // 添加菜单及菜单项
    JMenu menu1 = new JMenu( s: "添加");
    menuBar.add(menu1);
    JMenuItem menuItem1 = new JMenuItem( text: "添加学生信息");
    menuItem1.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            InfoAdd add = null;
            try {
                add = new InfoAdd();
            } catch (SQLException ex) {
                throw new RuntimeException(ex);
            }
            changePanel(add, imagePath: "image/pic14.jpg", n: 1);
        }
    });
    menu1.add(menuItem1);

    JMenu menu2 = new JMenu( s: "查询");
    menuBar.add(menu2);
    JMenuItem menuItem2 = new JMenuItem( text: "查询学生信息");

```

```

menuItem2.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        InfoSearch search = new InfoSearch();
        changePanel(search, imagePath: "image/pic10.jpg", n: 2);
    }
});
menu2.add(menuItem2);

```

```

JMenu menu3 = new JMenu( s: "修改");
menuBar.add(menu3);
JMenuItem menuItem3 = new JMenuItem( text: "修改学生信息");
menuItem3.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        InfoChange change = new InfoChange();
        changePanel(change, imagePath: "image/pic11.jpg", n: 3);
    }
});
menu3.add(menuItem3);

JMenu menu4 = new JMenu( s: "删除");
menuBar.add(menu4);
JMenuItem menuItem4 = new JMenuItem( text: "删除学生信息");
menuItem4.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        InfoDelete delete = new InfoDelete();
        changePanel(delete, imagePath: "image/pic12.jpg", n: 4);
    }
});
menu4.add(menuItem4);

JMenu menuFile = new JMenu( s: "退出");
menuBar.add(menuFile);
JMenuItem menuFile_exit = new JMenuItem( text: "退出系统");
menuFile_exit.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) { System.exit( status: 0); }
});
menuFile.add(menuFile_exit);

// 添加标签
label = new JLabel( text: "欢迎登录本系统~");
label.setFont(new Font( name: "微软雅黑", Font.BOLD, size: 28));
label.setForeground(new Color( rgb: 0xDC8B27));
label.setBounds(new Rectangle( x: 80, y: 120, width: 240, height: 100)); // 设置位置、大小
pane.add(label);

```

```

// 设置窗口居中
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
int centerX = screenSize.width / 2;
int centerY = screenSize.height / 2;
setLocation( x: centerX - getWidth() / 2, y: centerY - getHeight() / 2);

// 显示窗口
setVisible(true);
}

```



```

// 定义方法用于更换内容面板和背景
private void changePanel(JPanel panel, String imagePath, int n) {
    pane.removeAll();
    pane.add(panel);

    // 更换背景
    JLabel background = new JLabel(new ImageIcon(imagePath));
    background.setBounds(x: 0, y: 0, width: 400, height: 450);
    this.getLayeredPane().add(background, new Integer(value: Integer.MIN_VALUE + n));
    panel.setBounds(new Rectangle(x: 20, y: 20, width: 360, height: 300)); // 设置位置、大小
    setVisible(true);
}
}

```

5. InfoAdd （添加学生信息）

```

//InfoAdd类用于向数据库中添加学生信息
package EX3;

import javax.swing.*; // 导入 Swing 组件
import java.awt.*; // 导入 AWT 组件
import java.awt.event.*; // 导入事件处理的包
import java.sql.*; // 导入 JDBC 用于数据库连接和操作的相关类

public class InfoAdd extends JPanel implements ActionListener {

    // 定义文本框/标签的长宽
    private static final int TEXT_WIDTH = 200;
    private static final int TEXT_HEIGHT = 30;
    private static final int LABEL_WIDTH = 55;
    private static final int LABEL_HEIGHT = 20;

    // 数据库连接及语句相关变量
    private Connection connection;
    private Statement statement;

    // 定义标签以及对应的文本框
    private JLabel labelId = new JLabel(text: "添加学号");
    private JLabel labelName = new JLabel(text: "添加姓名");
    private JLabel labelGender = new JLabel(text: "添加性别");
    private JLabel labelBirthDay = new JLabel(text: "添加生日");
    private JLabel labelCollege = new JLabel(text: "添加学院");
    private JLabel labelGrade = new JLabel(text: "添加年级");
    private JLabel labelMajor = new JLabel(text: "添加专业");
    private JLabel labelCl = new JLabel(text: "添加班级");
}

```

```

private JTextField textFieldId = new JTextField();
private JTextField textFieldName = new JTextField();
private JTextField textFieldBirthday = new JTextField();
private JTextField textFieldCollege = new JTextField();
private JTextField textFieldGrade = new JTextField();
private JTextField textFieldMajor = new JTextField();
private JTextField textFieldCl = new JTextField();

// 定义性别单选框及对应的ButtonGroup
private JRadioButton radioButtonMale = new JRadioButton( text: "男");
private JRadioButton radioButtonFemale = new JRadioButton( text: "女");
private ButtonGroup buttonGroupGender = new ButtonGroup();

// 定义保存按钮
private JButton buttonSave = new JButton( text: "添加");

/**
 * 构造函数，初始化所有组件
 * @throws SQLException
 */
public InfoAdd() throws SQLException {
    initComponents();
    initLayout();
    initEvent();
}

/**
 * 初始化组件
 * @throws SQLException
 */
private void initComponents() throws SQLException {
    // 连接数据库
    connection = ConnecttoDB.connectDB( DBName: "JavaEX", id: "root", pwd: "123456");
    // 创建Statement
    statement = connection.createStatement();
    // 添加性别单选框到ButtonGroup，设置男性为默认选项
    buttonGroupGender.add(radioButtonMale);
    buttonGroupGender.add(radioButtonFemale);
    radioButtonMale.setSelected(true);
}

```

```

/**
 * 初始化布局
 */
private void initLayout() {
    setLayout(new GridBagLayout());
}

```

```

// 添加组件
addComponent(labelId, x: 0, y: 0);
addComponent(textFieldId, x: 1, y: 0);
addComponent(labelName, x: 0, y: 1);
addComponent(textFieldName, x: 1, y: 1);
addComponent(labelGender, x: 0, y: 2);
JPanel panelGender = new JPanel(new FlowLayout(FlowLayout.LEFT));
panelGender.add(radioButtonMale);
panelGender.add(radioButtonFemale);
addComponent(panelGender, x: 1, y: 2);
addComponent(labelBirthday, x: 0, y: 3);
addComponent(textFieldBirthday, x: 1, y: 3);
addComponent(labelCollege, x: 0, y: 4);
addComponent(textFieldCollege, x: 1, y: 4);
addComponent(labelGrade, x: 0, y: 5);
addComponent(textFieldGrade, x: 1, y: 5);
addComponent(labelMajor, x: 0, y: 6);
addComponent(textFieldMajor, x: 1, y: 6);
addComponent(labelCl, x: 0, y: 7);
addComponent(textFieldCl, x: 1, y: 7);
addComponent(buttonSave, x: 1, y: 8);
}

```

//初始化事件监听器

```

private void initEvent() { buttonSave.addActionListener(this); }

/**
 * 添加组件到Panel中的方法
 * @param component 待添加组件
 * @param x 区域横坐标
 * @param y 区域纵坐标
 */
private void addComponent(JComponent component, int x, int y) {
    GridBagConstraints constraints = new GridBagConstraints();
    constraints.gridx = x;
    constraints.gridy = y;
    constraints.fill = GridBagConstraints.HORIZONTAL;
    constraints.insets = new Insets(5, 5, 5, 5);
    add(component, constraints);
}

```



```

/**
 * 保存按钮监听器
 * @param e 按钮事件
 */
@Override
public void actionPerformed(ActionEvent e) {
    // 获取所有字段内容
    String id = textFieldId.getText();
    String name = textFieldName.getText();
    String gender = radioButtonMale.isSelected() ? "男" : "女";
    String birthday = textFieldBirthday.getText();
    String college = textFieldCollege.getText();
    String grade = textFieldGrade.getText();
    String major = textFieldMajor.getText();
    String cl = textFieldCl.getText();

    try {
        // 构造插入SQL语句
        String sqlstr = "insert into stuinfo values(" + id + "," + name + "," + gender + "," + birthday
            + "," + college + "," + grade + "," + major + "," + cl + ")";
        // 执行插入语句
        statement.executeUpdate(sqlstr);
        // 清空所有字段
        textFieldId.setText("");
        textFieldName.setText("");
        textFieldBirthday.setText("");
        textFieldCollege.setText("");
        textFieldGrade.setText("");
        textFieldMajor.setText("");
        textFieldCl.setText("");
        // 显示成功信息
        JOptionPane.showMessageDialog( parentComponent: this, message: "添加成功!");
    } catch (SQLException ex) {
        // 显示失败信息
        JOptionPane.showMessageDialog( parentComponent: this, message: "添加失败!");
        // 打印出错信息
        ex.printStackTrace();
    }
}
}

```

6. InfoSearch （查询学生信息）

```

//导入相关的包
package EX3;

import javax.swing.*; // 导入 Swing 组件
import java.awt.*; // 导入 AWT 组件
import java.awt.event.*; // 导入事件处理的包
import java.sql.*; // 导入 JDBC 用于数据库连接和操作的相关类

public class InfoSearch extends JPanel implements ActionListener {
    Connection connection;
    Statement statement;
    JLabel Label1=new JLabel( text: "请输入要查询学生的学号");
    JTextField TextField1=new JTextField();
    JButton Button1=new JButton( text: "查询");
    JTextArea TextArea1=new JTextArea();
}

```

```

public InfoSearch(){
    try{
        Init();
    }
    catch (Exception e){
        e.printStackTrace();
    }
}

//重构Init()函数, 细化Swing组件的处理
private void Init() throws Exception{
    connection= ConnecttoDB.connectDB( DBName: "JavaEX", id: "root", pwd: "123456");
    statement=connection.createStatement();
    //使用网格布局
    this.setLayout(new GridLayout( rows: 4, cols: 1, hgap: 5, vgap: 5));
    JPanel panel1 = new JPanel();
    //使用边界布局
    panel1.setLayout(new BorderLayout( hgap: 5, vgap: 5));
    panel1.add(Label1,BorderLayout.WEST);
    panel1.add(TextField1,BorderLayout.CENTER);
    panel1.setBorder(BorderFactory.createEmptyBorder( top: 10, left: 10, bottom: 10, right: 10));

    JPanel panel2 = new JPanel(new FlowLayout(FlowLayout.CENTER, hgap: 10, vgap: 10));
    panel2.add(Button1);

    this.add(panel1);
    this.add(panel2);
    this.add(new JScrollPane(TextArea1));

    Button1.addActionListener( this);
}

```

```

@Override
public void actionPerformed(ActionEvent e){
    String ID=TextField1.getText();
    TextArea1.setText("");
    //查询语句应使用参数化查询, 避免SQL注入攻击
    String sqlstr="select * from stuinfo where 学号=?";
    try{
        //使用prepareStatement函数执行参数化查询
        PreparedStatement pstmt=connection.prepareStatement(sqlstr);
        pstmt.setInt( parameterIndex: 1, Integer.parseInt(ID));
        ResultSet resultSet=pstmt.executeQuery();
        if(resultSet.next()){
            TextArea1.append("学号: " + resultSet.getInt( columnLabel: "学号") + "\n");
            TextArea1.append("姓名: " + resultSet.getString( columnLabel: "姓名") + "\n");
            TextArea1.append("性别: " + resultSet.getString( columnLabel: "性别") + "\n");
            TextArea1.append("生日: " + resultSet.getString( columnLabel: "生日") + "\n");
            TextArea1.append("学院: " + resultSet.getString( columnLabel: "学院") + "\n");
            TextArea1.append("年级: " + resultSet.getString( columnLabel: "年级") + "\n");
            TextArea1.append("专业: " + resultSet.getString( columnLabel: "专业") + "\n");
            TextArea1.append("班级: " + resultSet.getString( columnLabel: "班级") + "\n");
        }
        else{
            JOptionPane.showMessageDialog( parentComponent: this, message: "没有该生信息!");
        }
        TextField1.setText("");
        pstmt.close();
    }
    catch (NumberFormatException ex){
        JOptionPane.showMessageDialog( parentComponent: this, message: "请输入正确格式的学号!");
    }
    catch (SQLException ex){
        JOptionPane.showMessageDialog( parentComponent: this, message: "查询失败!");
    }
}
}

```

7. InfoChange （查询信息）

```
// 导入需要的包
package EX3;

import javax.swing.*; // 导入 Swing 组件
import java.awt.*; // 导入 AWT 组件
import java.awt.event.*; // 导入事件处理的包
import java.sql.*; // 导入 JDBC 用于数据库连接和操作的相关类

public class InfoChange extends JPanel implements ActionListener {
    // 声明数据库连接和声明
    private Connection connection;
    private Statement statement;

    // 声明需要使用到的 UI 组件
    private JLabel Label1=new JLabel( text: "目标学号");
    private JTextField TextField1=new JTextField();
    private JLabel Label2=new JLabel( text: "修改姓名");
    private JLabel Label3=new JLabel( text: "修改性别");
    private JLabel Label4=new JLabel( text: "修改学院");
    private JLabel Label5=new JLabel( text: "修改年级");
    private JLabel Label6=new JLabel( text: "修改专业");
    private JLabel Label7=new JLabel( text: "修改班级");
    private JTextField TextField2=new JTextField();
    private JTextField TextField3=new JTextField();
    private JTextField TextField4=new JTextField();
    private JTextField TextField5=new JTextField();
    private JTextField TextField6=new JTextField();
    private JTextField TextField7=new JTextField();
    private ButtonGroup ButtonGroup1=new ButtonGroup();
    private JRadioButton RadioButton1=new JRadioButton( text: "男");
    private JRadioButton RadioButton2=new JRadioButton( text: "女");
    private JButton Button1=new JButton( text: "修改");

    // 构造函数，初始化
    public InfoChange() {
        try{
            init(); // 调用 Init 方法
        } catch (Exception e){
            e.printStackTrace(); // 打印异常
        }
    }
}
```



```

// 初始化连接数据库和 UI 界面
private void init() throws Exception {
    connection= ConnecttoDB.connectDB( DBName: "JavaEX", id: "root", pwd: "123456"); // 连接到数据库
    statement=connection.createStatement(); // 创建声明

    // 设定 UI 界面元素布局
    setLayout(null);
    Label1.setBounds(new Rectangle( x: 15, y: 15, width: 150, height: 30));
    TextField1.setBounds(new Rectangle( x: 95, y: 15, width: 200, height: 30));
    Label2.setBounds(new Rectangle( x: 15, y: 55, width: 60, height: 25));
    Label3.setBounds(new Rectangle( x: 15, y: 85, width: 60, height: 25));
    Label4.setBounds(new Rectangle( x: 15, y: 120, width: 60, height: 25));
    Label5.setBounds(new Rectangle( x: 15, y: 170, width: 60, height: 25));
    Label6.setBounds(new Rectangle( x: 15, y: 205, width: 60, height: 25));
    Label7.setBounds(new Rectangle( x: 15, y: 245, width: 60, height: 25));
    TextField2.setBounds(new Rectangle( x: 95, y: 55, width: 200, height: 30));
    TextField4.setBounds(new Rectangle( x: 95, y: 120, width: 200, height: 30));
    TextField5.setBounds(new Rectangle( x: 95, y: 165, width: 200, height: 30));
    TextField6.setBounds(new Rectangle( x: 95, y: 200, width: 200, height: 30));
    TextField7.setBounds(new Rectangle( x: 95, y: 240, width: 200, height: 30));
    RadioButton1.setBounds(new Rectangle( x: 135, y: 85, width: 65, height: 30));
    RadioButton2.setBounds(new Rectangle( x: 215, y: 85, width: 85, height: 30));
    Button1.setBounds(new Rectangle( x: 102, y: 270, width: 180, height: 30));
    RadioButton1.setSelected(true);

    // 添加按钮监听事件
    Button1.addActionListener( k this);

    // 将 UI 元素添加到面板上
    add(TextField1);
    add(Label1);
    add(TextField2);
    add(Label2);
    add(RadioButton1);
    add(RadioButton2);
    add(Label3);
    add(Label4);
    add(TextField4);
    add(Label5);
    add(TextField5);
    add(Label6);
    add(TextField6);

```

```

    add(Label7);
    add(TextField7);
    add(Button1);

    // 将单选按钮添加到按钮组
    ButtonGroup1.add(RadioButton1);
    ButtonGroup1.add(RadioButton2);
}

```

```

// 实现 ActionListener 接口，处理 "修改" 按钮点击事件
@Override
public void actionPerformed(ActionEvent e){
    // 从 UI 上获取要修改的学生信息
    String id=TextField1.getText().trim();
    String name=TextField2.getText().trim();
    String gender= RadioButton1.isSelected() ? "男" : "女";
    String college=TextField4.getText().trim();
    String grade=TextField5.getText().trim();
    String major=TextField6.getText().trim();
    String cl=TextField7.getText().trim();

    // 构造 SQL 语句
    String sqlstr="update stuinfo set 姓名 = '"+name+"',性别 = '"+gender+"',学院 = '"+college+"',年级 = '"+
        grade+"',专业 = '"+major+"',班级 = '"+cl+"' where 学号 = '"+id+"'";

    // 弹出确认对话框，确认是否修改学生信息
    if(JOptionPane.showConfirmDialog( parentComponent: this, message: "确认修改？")==JOptionPane.YES_OPTION){
        try{
            int result = statement.executeUpdate(sqlstr); // 执行 SQL 更新语句，执行修改学生信息
            if (result > 0){
                TextField1.setText(""); // 将编辑框清空
                JOptionPane.showMessageDialog( parentComponent: this, message: "修改成功！"); // 弹出提示框，提示修改成功
            } else {
                JOptionPane.showMessageDialog( parentComponent: this, message: "修改失败！"); // 弹出提示框，提示修改失败
            }
        } catch (Exception ex){
            JOptionPane.showMessageDialog( parentComponent: this, message: "修改失败！"); // 弹出提示框，提示修改失败
        }
    }
}
}

```

8. InfoDelete （删除学生信息）

```

//InfoDelete类，继承自JPanel类，用于实现学生信息的删除功能
package EX3;

import javax.swing.*; // 导入 Swing 组件
import java.awt.*; // 导入 AWT 组件
import java.awt.event.*; // 导入事件处理的包
import java.sql.*; // 导入 JDBC 用于数据库连接和操作的相关类

public class InfoDelete extends JPanel implements ActionListener {

    private static final String DATABASE_NAME = "JavaEX";
    private static final String DATABASE_USER = "root";
    private static final String DATABASE_PASSWORD = "123456";

    private static final String LABEL_TEXT = "请输入要删除的学生的学号";
    private static final String BUTTON_TEXT = "删除";
    private static final String CONFIRM_MESSAGE = "确认删除？";
    private static final String DELETE_SUCCESS_MESSAGE = "删除成功";
    private static final String DELETE_FAILED_MESSAGE = "删除失败";
}

```



```

// 数据库连接和语句对象
private Connection connection;
private Statement statement;

// 标签、文本框、按钮
private JLabel label = new JLabel(LABEL_TEXT);
private JTextField textField = new JTextField();
private JButton button = new JButton(BUTTON_TEXT);

public InfoDelete() {
    try {
        init();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

private void init() throws Exception {
    // 连接数据库
    connection = ConnecttoDB.connectDB(DATABASE_NAME, DATABASE_USER, DATABASE_PASSWORD);
    statement = connection.createStatement();

    // 设置布局为null
    setLayout(null);

    // 设置组件的位置和大小
    label.setFont(new Font("微软雅黑", Font.BOLD, size: 16));
    label.setBounds(new Rectangle(x: 75, y: 30, width: 200, height: 30));
    textField.setBounds(new Rectangle(x: 70, y: 90, width: 200, height: 40));
    button.setBounds(new Rectangle(x: 70, y: 170, width: 200, height: 40));

    // 添加按钮的监听器
    button.addActionListener(this);

    // 将组件添加到面板中
    add(textField);
    add(label);
    add(button);
}

```

```

@Override
public void actionPerformed(ActionEvent e) {
    // 获取用户输入的学号
    String id = textField.getText();

    // 消息框，询问是否删除该学生
    if (JOptionPane.showConfirmDialog( parentComponent: this, CONFIRM_MESSAGE) == JOptionPane.YES_OPTION) {

        try {
            // 执行删除语句
            String sql = "delete from stuinfo where 学号 = ?";
            PreparedStatement preparedStatement = connection.prepareStatement(sql);
            preparedStatement.setString( parameterIndex: 1, id);
            preparedStatement.executeUpdate();

            // 清空文本框，并弹出提示框
            textField.setText("");
            JOptionPane.showMessageDialog( parentComponent: this, DELETE_SUCCESS_MESSAGE);
        } catch (Exception ex) {
            // 删除失败，弹出提示框，并输出异常信息
            JOptionPane.showMessageDialog( parentComponent: this, DELETE_FAILED_MESSAGE);
            ex.printStackTrace();
        }
    }
}
}

```

实验总结：

本次实验旨在让学生掌握使用 Java 编写图形界面程序、事件编程和基本数据库操作的能力。

在实验过程中，我首先了解了图形界面编程的基本概念和方法，包括 Swing 组件库的使用以及界面元素的排版和布局。通过课程教学和实践练习，我逐渐掌握了 Swing 组件库的常用控件和属性的使用方法，并能够基于这些控件和属性实现一个简单的图形界面应用程序。

接着，我学习了事件编程的思想和技巧，深入理解了事件监听器和回调函数的概念，学会了利用事件机制实现 GUI 界面的响应和交互。在实验中，我通过丰富的实践练习，加深了对事件编程原理的理解和应用能力。

最后，我学习了如何利用 Java 对数据库进行基本的数据操作，了解了数据库连接、数据查询、数据插入、更新和删除等操作的基本流程和原理。在实验中，我编写了一段 Java 程序，通过 JDBC 的方式连接到本地的 MySQL 数据库，实现了对 StuInfo 表中的数据进行查询、插入和更新等操作。通过此实验，我巩固了对数据库操作的理解和实际应用的能力。

总的来说，本次实验是一次非常有意义的学习和实践活动。通过这次实验，我不仅掌握了基本的 Java 图形界面编程技巧和事件编程思想，还进一步了解了 Java 与数据库的交互机制，为今后的学习和工作打下了坚实的基础。