

BÁO CÁO BÀI TẬP LỚN OOP ỨNG DỤNG QUẢN LÝ SINH VIÊN

Tác giả:

Lê Xuân Sơn (AT180142)

Nguyễn Duy Vượng (AT180151)

Trần Văn Long (AT180131)

Xem trên [GitHub](#)

MỤC LỤC

I. KHẢO SÁT, MÔ TẢ YÊU CẦU BÀI TOÁN	3
1. MÔ TẢ YÊU CẦU	3
2. PHÂN LOẠI NGƯỜI DÙNG	3
II. PHÂN TÍCH THIẾT KẾ	3
1. TỔNG QUAN	3
2. THIẾT KẾ CƠ SỞ DỮ LIỆU	3
3. CẤU TRÚC DỰ ÁN	5
4. LUỒNG HOẠT ĐỘNG	7
5. PHÂN RÃ CHỨC NĂNG	7
III. DEMO SẢN PHẨM	8
1. FORM ĐĂNG NHẬP	8
2. MÀN HÌNH CHÍNH	13
a. Trang chủ	13
b. Danh mục khoa	15
c. Quản lý sinh viên	18
d. Quản lý điểm	23
e. Quản lý môn	26
f. Danh mục lớp	27
IV. KẾT LUẬN	27

NỘI DUNG

I. KHẢO SÁT, MÔ TẢ YÊU CẦU BÀI TOÁN

1. MÔ TẢ YÊU CẦU

Hỗ trợ quản lý khoa, lớp, sinh viên, môn học, điểm học phần

Cung cấp các chức năng cơ bản: xem thông tin, thêm, sửa, xoá các đối tượng

Kết nối cơ sở dữ liệu để lưu trữ dữ liệu và thực hiện các thao tác

Đăng nhập qua tên người dùng và mật khẩu

2. PHÂN LOẠI NGƯỜI DÙNG

Admin: Có quyền xem, thêm, sửa, xoá mọi đối tượng

Guest: Chỉ có quyền xem, không thể thêm, sửa, xoá

II. PHÂN TÍCH THIẾT KẾ

1. TỔNG QUAN

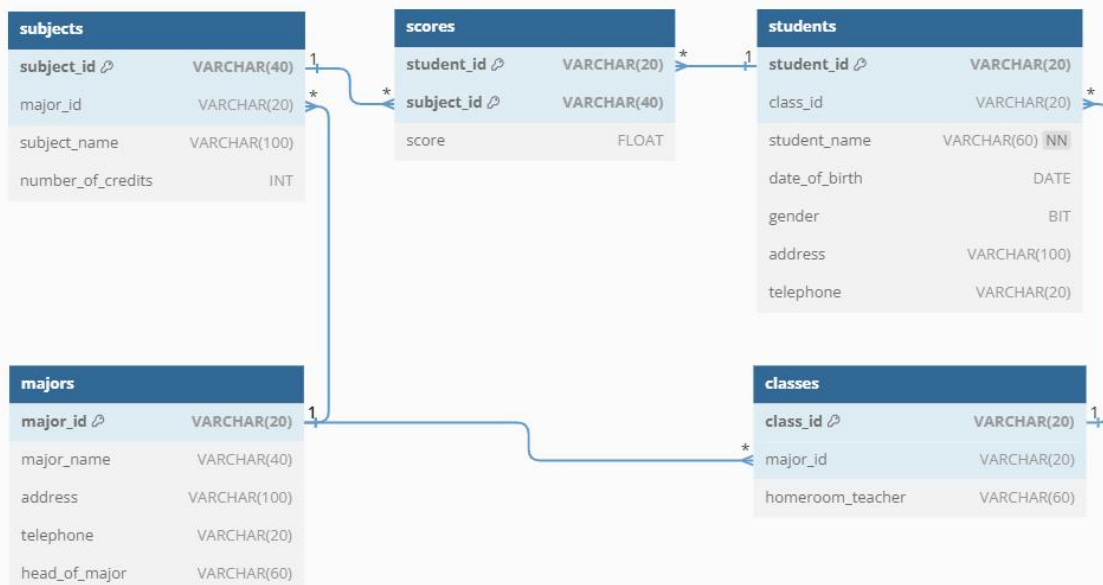
Triển khai Maven project

Lợi thế:

- Gợi ý sẵn cấu trúc tổng thể, dễ dàng module hóa dự án, thuận lợi cho phân công công việc, quản lý dự án
- Dễ dàng mở rộng thêm chức năng
- Tự động tải, thêm, bớt thư viện bằng cách thêm, xoá dependency ở file pom.xml
- Cộng đồng lớn, dễ dàng tìm kiếm hỗ trợ

2. THIẾT KẾ CƠ SỞ DỮ LIỆU

Sơ đồ cơ sở dữ liệu chính, gồm các bảng, các trường, các khóa chính, khóa ngoại và mối quan hệ giữa các bảng:



Cơ sở dữ liệu gồm các bảng: **accounts**, **majors**, **classes**, **subjects**, **students**, **scores**

Trên sơ đồ không biểu diễn bảng **accounts**, bảng này độc lập, không có quan hệ với bảng khác

Trong đó:

- Accounts: lưu trữ thông tin về tài khoản người dùng
- Majors: lưu trữ thông tin về khoa
- Classes: lưu trữ thông tin về lớp
- Subjects: lưu trữ thông tin về môn học
- Students: lưu trữ thông tin về sinh viên
- Scores: lưu trữ thông tin về điểm học phần

Câu lệnh tạo CSLD nằm tại **src/main/resources**:

- DatabaseInit.sql: tạo cơ sở dữ liệu chính, gồm các bảng khoa, lớp, môn học, sinh viên, điểm, thêm thông tin mẫu
- AccountsDatabase.sql: tạo cơ sở dữ liệu lưu thông tin tài khoản người dùng, gồm bảng accounts

Cơ sở dữ liệu được lưu trên máy chủ MySQL, thông tin kết nối lưu tại **src/main/resources/database.properties**

3. CẤU TRÚC DỰ ÁN

```
StudentManagementSystem
├── pom.xml
├── src
│   ├── main
│   │   ├── java/com/kma/sms
│   │   │   ├── StudentManagementSystem.java
│   │   │   ├── authen
│   │   │   │   ├── UserSession.java
│   │   │   │   └── UserSessionRequest.java
│   │   │   ├── ui
│   │   │   │   ├── LoginForm.form
│   │   │   │   ├── MainForm.form
│   │   │   │   ├── studentpopup
│   │   │   │   │   ├── StudentAdder.form
│   │   │   │   │   └── StudentEditor.form
│   │   │   │   └── ...
│   │   │   ├── controller
│   │   │   │   ├── LoginFormController.java
│   │   │   │   ├── MajorManagePanelController.java
│   │   │   │   ├── ScoreManagePanelController.java
│   │   │   │   └── ...
│   │   │   ├── model
│   │   │   │   ├── Class.java
│   │   │   │   ├── ClassRequestObject.java
│   │   │   │   └── ...
│   │   │   ├── dao
│   │   │   │   ├── ClassDAO.java
│   │   │   │   ├── MajorDAO.java
│   │   │   │   ├── ScoreDAO.java
│   │   │   │   └── ...
│   │   │   └── util
│   │   │       └── DatabaseConnection.java
│   └── resources
│       ├── AccountsDatabase.sql
│       ├── DatabaseInit.sql
│       ├── database.properties
│       ├── background.properties
│       └── images
│           ├── logo.png
│           └── header.png
```

pom.xml: Project Object Model, bao gồm:

- Tên project, version, mô tả project
- Danh sách các thư viện, frameworks cần dùng. Maven sẽ tự động tải, include các thư viện này vào project
- Thông tin về build config
- Các thông tin khác

src/main/resources:

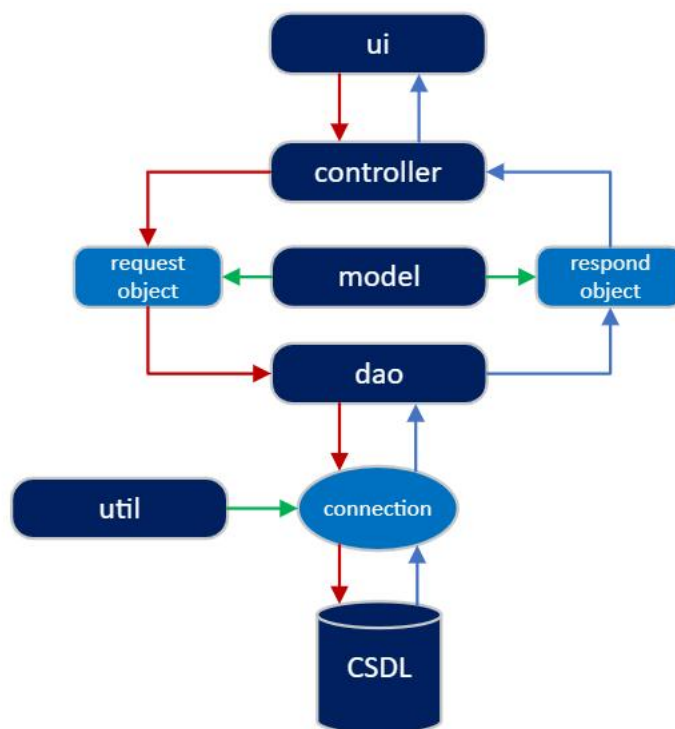
- Câu lệnh tạo cơ sở dữ liệu, tài khoản người dùng (.sql)
- File cấu hình, thông tin về một số đối tượng (.properties)
- Images: folder chứa các file ảnh được dùng trong dự án

src/main/java/com/kma/sms:

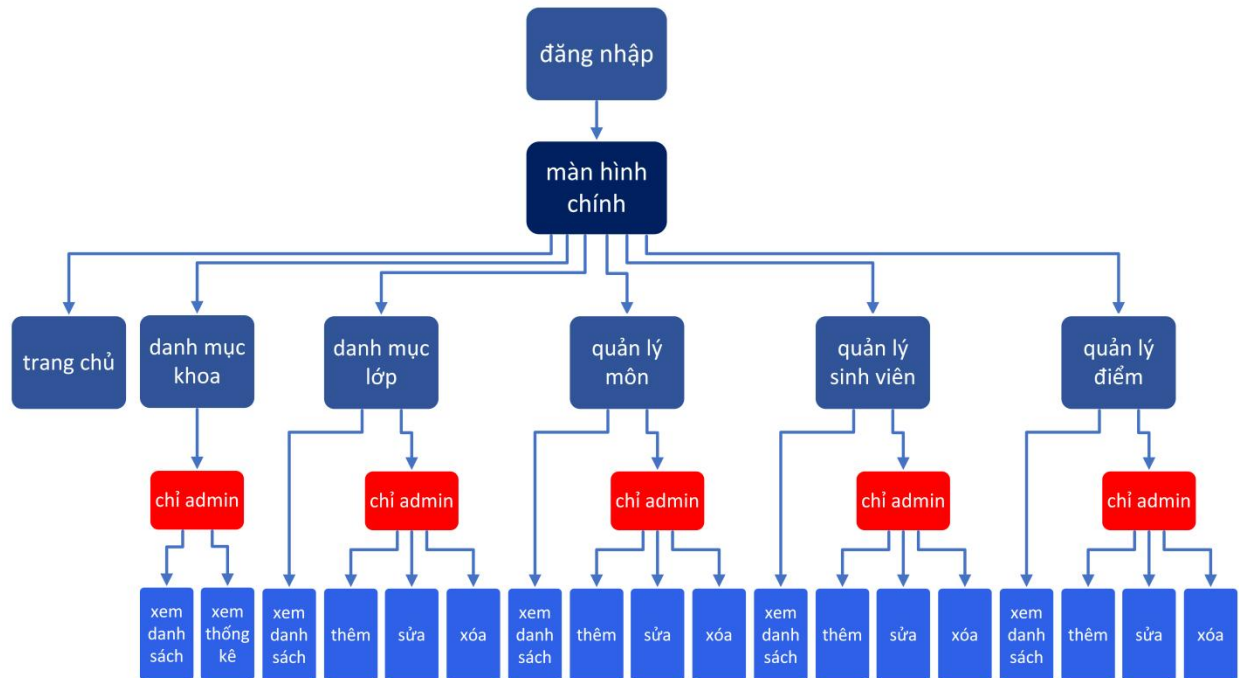
- StudentManagementSystem.java: Entry point của dự án
- Authen:
 - UserSessionRequest: lưu trữ, truyền đạt thông tin đăng nhập của người dùng đến lớp xử lý
 - UserSession: lưu trữ, trả về phiên đăng nhập của người dùng (hợp lệ hay không, kèm quyền của người dùng đó)
- Model:
 - Đóng vai trò như cấu trúc dữ liệu, lưu trữ các trường thông tin về đối tượng (Student, Class, Score, ...), truyền trong các lớp.
 - RequestObject: cấu trúc lưu trữ thông tin cung cấp bởi người dùng, sẽ được gửi đến các lớp xử lý
 - Non-RequestObject (RespondObject): Cấu trúc lưu trữ thông tin trả về từ chương trình, sau khi xử lý RequestObject
- Ui:
 - Giao diện người dùng, gồm các JForm, JPanel
 - Nhiệm vụ tương tác với người dùng, nhận thông tin nhập vào bởi người dùng, gửi đến controller
 - Nhận thông tin từ controller trả về người dùng
- Controller:
 - Lớp trung gian giữa giao diện vào các lớp xử lý thông tin bên dưới (dao)

- Nhiệm vụ kiểm tra thông tin gửi đến từ ui xem có hợp lệ, đúng cú pháp, thay đổi định dạng, format thông tin, construct RequestObject gửi đến dao
- Nhận thông tin trả về từ dao, format lại gửi lên ui
- Dao:
 - Data Access Object: cung cấp các phương thức làm việc với cơ sở dữ liệu (thêm, sửa, xóa, ...)
 - Nhận request từ controller, kết nối từ util
 - Trả về kết quả (là một RespondObject) đến controller
- Util:
 - Utility: nhiệm vụ tạo lập kết nối đến csdl

4. LUỒNG HOẠT ĐỘNG



5. PHÂN RÃ CHỨC NĂNG



III. DEMO SẢN PHẨM

1. FORM ĐĂNG NHẬP (LoginForm)

Gồm:

```

private javax.swing.JPasswordField PasswordInputField;
private javax.swing.JButton loginButton;
private javax.swing.JLabel passwordPanel;
private javax.swing.JTextField userNameInputField;
private javax.swing.JLabel userNamePanel;
  
```


Khi nhấn Login, gọi **LoginFormController** lấy kết quả đăng nhập:

```
// Get username and password
String username = userNameInputField.getText();
String password = new String(PasswordInputField.getPassword());
String message = LoginFormController.getLoginMessage(username, password);
```

getLoginMessage kiểm tra sơ bộ, các trường phải được điền:

```
public static String getLoginMessage(String username, String password) {
    // Check if username and password is empty
    if (username.isEmpty() || password.isEmpty()) {
        // Return error message
        return "Vui lòng nhập tên đăng nhập và mật khẩu";
    }
    int result = sendLoginRequestAndReceiveResponse(username, password);
    String message = "";
    switch (result) {
        case 1:
            message = "Đăng nhập thành công";
            break;
        default:
            message = "Đăng nhập thất bại";
            break;
    }
    return message;
}
```

sendLoginRequestAndReceiveResponse thực hiện hash mật khẩu theo thuật toán SHA-256.

Tạo Object **UserSessionRequest** gửi đến **UserDAO** để tạo phiên đăng nhập

```
private static int sendLoginRequestAndReceiveResponse(String username, String password) {
    // hash password using SHA-256
    String hashedPassword = "";
    try {
```

```

        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] hash = md.digest(password.getBytes(StandardCharsets.UTF_8));
        hashedPassword = String.format("%064x", new BigInteger(1, hash));
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }

    UserSessionRequest userSessionRequest = new UserSessionRequest(username,
hashedPassword);

    int result = UserDao.login(userSessionRequest);

    return result;
}

```

UserSessionRequest:

```

public class UserSessionRequest {
    String username;
    String hashedPassword;

    public UserSessionRequest(String username, String hashedPassword) {
        this.username = username;
        this.hashedPassword = hashedPassword;
    }

    // gettter
    public String getUsername() {
        return username;
    }

    public String getHashedPassword() {
        return hashedPassword;
    }
}

```

UserDAO:

```
public class UserDAO {

    public static int login(UserSessionRequest userSessionRequest) {
        try (Connection connection = DatabaseConnection.connect()){
            // SQL injection prevention
            String query = "SELECT * FROM accounts WHERE username = ? AND hashed_password = ?";

            PreparedStatement preparedStatement = connection.prepareStatement(query);
            preparedStatement.setString(1, userSessionRequest.getUsername());
            preparedStatement.setString(2, userSessionRequest.getHashedPassword());

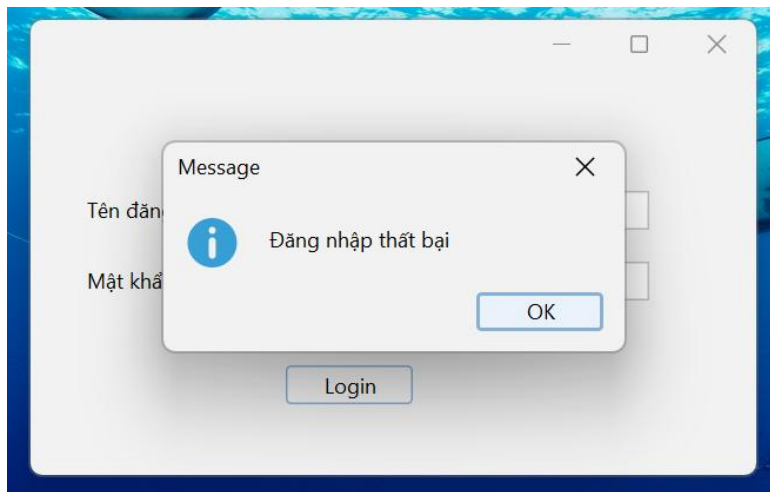
            System.out.println(preparedStatement);

            java.sql.ResultSet resultSet = preparedStatement.executeQuery();
            if (resultSet.next()) {
                Boolean isAdmin = resultSet.getBoolean("is_admin");
                // create user session
                UserSession.setIsAdmin(isAdmin);
                return 1;
            }
            else {
                return 0;
            }
        }
        catch(SQLException e){
            e.printStackTrace();
        }
        return 0;
    }
}
```

Sau khi nhận kết nối từ `util.DatabaseConnection`, thực hiện tạo câu lệnh query kiểm tra thông tin, kiểm tra quyền (`is_admin`).

Nếu `username` và `hashed_password` đúng, tạo `UserSession`, trả về 1, sau đó người dùng có thể vào màn hình chính.

Nếu không đúng, hiện lỗi:



Util:

```
public class DatabaseConnection {

    private static final String PROPERTIES_FILE = "database.properties";
    private static String url;
    private static String user;
    private static String password;

    // Static block to initialize the database properties
    static {
        try (InputStream input =
DatabaseConnection.class.getClassLoader().getResourceAsStream(PROPERTIES_FILE)) {
            Properties prop = new Properties();
            if (input == null) {
                throw new IOException("Unable to find " + PROPERTIES_FILE);
            }
            prop.load(input);
        }
    }
}
```

```

        url = prop.getProperty("db.url");
        user = prop.getProperty("db.user");
        password = prop.getProperty("db.password");

    } catch (IOException e) {
        e.printStackTrace();
        throw new RuntimeException("Failed to load database properties", e);
    }
}

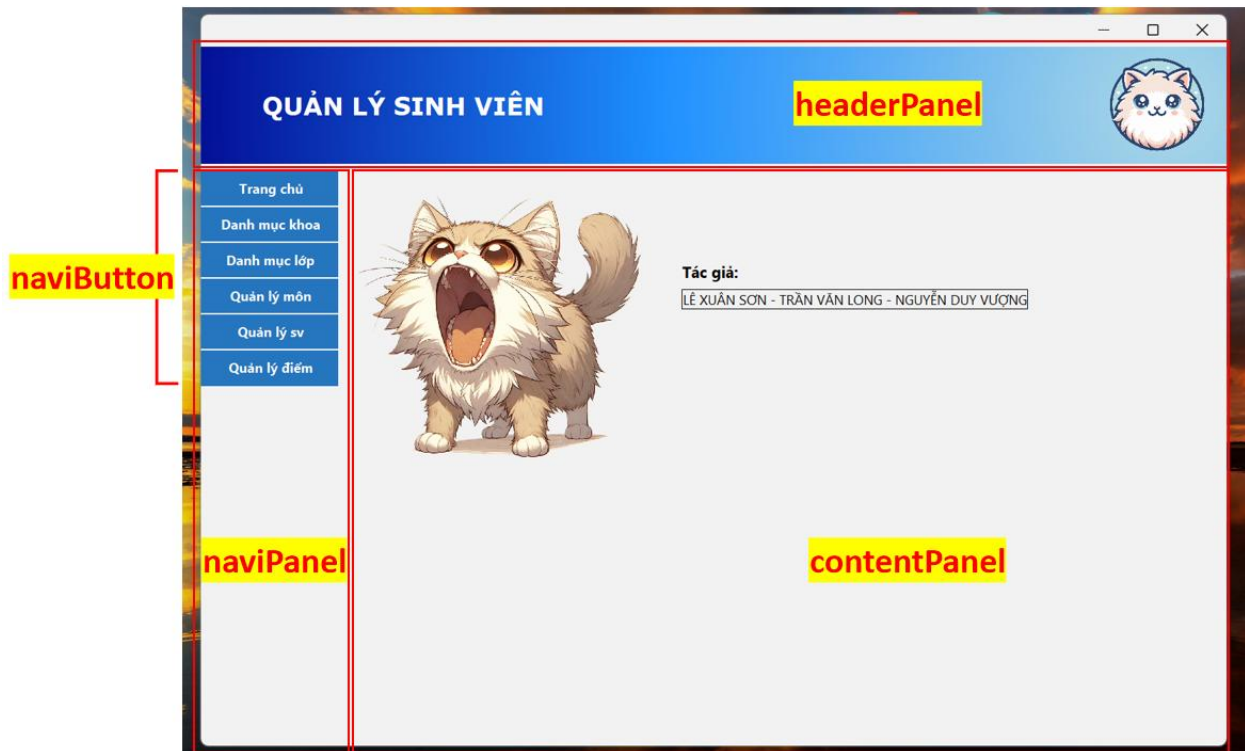
public static Connection connect() throws SQLException {
    return DriverManager.getConnection(url, user, password);
}
}

```

Các chức năng sau này đều hoạt động với nguyên lý cơ bản tương tự như Login.

2. MÀN HÌNH CHÍNH (MainForm)

a. Trang chủ (homePagePanel)



Màn hình chính chia làm 3 khu vực:

```
private javax.swing.JPanel contentPanel;
private javax.swing.JPanel headerPanel;
private javax.swing.JPanel naviPanel;
```

- **headerPanel:** Hiện thị banner, logo
- **naviPanel:** chứa các nút điều hướng

```
private javax.swing.JButton classDirNaviButton;
private javax.swing.JButton departmentDirNaviButton;
private javax.swing.JButton homePageNaviButton;
private javax.swing.JButton scoreMangeNaviButton;
private javax.swing.JButton studentManageNaviButton;
private javax.swing.JButton subjectManageNaviButton;
```

- **contentPanel:** show các JPanel tương ứng khi người dùng nhấn nút điều hướng, sử dụng **CardLayout**

```
public class MainForm extends javax.swing.JFrame {
```

```

/* Card layout */
private void initCardLayout() {
    CardLayout cardLayout = new CardLayout();
    contentPanel.setLayout(cardLayout);
    contentPanel.add(new HomePagePanel(), "homePagePanel");
    contentPanel.add(new MajorManagePanel(), "majorManagePanel");
    contentPanel.add(new ClassManagePanel(), "classManagePanel");
    contentPanel.add(new SubjectManagePanel(), "subjectManagePanel");
    contentPanel.add(new StudentManagePanel(), "studentManagePanel");
    contentPanel.add(new ScoreManagePanel(), "scoreManagePanel");

    // always fit the content panel to the parent panel
    this.pack();
    contentPanel.revalidate();
}

/** Creates new form MainForm */
public MainForm() {
    initComponents();
    initCardLayout();
}

// ..... //
}

```

Chuyển đổi giữa các JPanel trong CardLayout

```

private void studentManageNaviButtonActionPerformed(java.awt.event.ActionEvent
evt) {// GEN-FIRST:event_studentManageNaviButtonActionPerformed

    // show student manage panel and hide other panels
    CardLayout cardLayout = (CardLayout) contentPanel.getLayout();
    cardLayout.show(contentPanel, "studentManagePanel");
} // GEN-LAST:event_studentManageNaviButtonActionPerformed

```

b. Danh mục khoa (majorManagePanel)

Chỉ admin có quyền xem Danh mục khoa và các thống kê liên quan. Kiểm tra quyền của người dùng hiện tại (xảy ra tại Controller)

```
if (!UserSession.getIsAdmin()) {  
    return null;  
}
```

majorManagePanel gồm **JTable**, các **(jTextField)**,...

```
private javax.swing.JTextField avgScoreField;  
private javax.swing.JTextField excellentField;  
private javax.swing.JTextField intermediateField;  
private javax.swing.JTable majorTable;  
private javax.swing.JScrollPane majorTableScrollPane;  
private javax.swing.JTextField numOfFemalesField;  
private javax.swing.JTextField numOfMalesField;  
private javax.swing.JTextField numOfStudentsField;
```


QUẢN LÝ SINH VIÊN

Trang chủ

Danh mục khoa

Danh mục lớp

Quản lý môn

Quản lý sv

Quản lý điểm

MajorManagePanel

Tổng số: 987

Điểm tb: 6.75

Xuất sắc: 1

Nam: 802

Giới: 80

Nữ: 185

Khá: 778

STT	Mã ngành	Tên ngành	Địa chỉ	SĐT	Trưởng khoa
1	AT	An toàn thông tin	Phòng 1101, Nhà A1	376984887	Nguyễn Văn A
2	CB	Khoa cơ bản	Phòng 1104, Nhà B2	234584887	Lê Thị D
3	CT	Công nghệ thông tin	Phòng 1102, Nhà A1	546984887	Nguyễn Văn B
4	DT	Điện tử viễn thông	Phòng 1103, Nhà A1	876984887	Nguyễn Văn C

majorTable

Các trường thông tin được cập nhật mỗi khi người dùng chọn khoa trên bảng.

Về cơ bản vẫn là construct request, tạo câu lệnh query, trả kết quả. Luồng hoạt động tương tự Login.

Ví dụ về các lấy điểm trung bình:

MajorManagePanel

```
// update average score in text field
double averageScore =
MajorManagePanelController.getAverageScoreByMajorId(majorId);
// get first 2 digits after decimal point
averageScore = Math.round(averageScore * 100.0) / 100.0;
avgScoreField.setText(Double.toString(averageScore));
```

MajorManagePanelController

```
// get average score by majorId
public static double getAverageScoreByMajorId(String majorId) {
```

```

        return MajorDAO.getAverageScore(majorId);
    }

```

MajorDAO

```

public static float getAverageScore(String majorId) {
    try (Connection connection = DatabaseConnection.connect()) {
        Statement statement = connection.createStatement();
        ResultSet resultSet = statement.executeQuery("SELECT AVG(scores.score) AS
average_score"
            + " FROM scores JOIN students ON scores.student_id =
students.student_id JOIN classes ON students.class_id = classes.class_id"
            + " WHERE scores.score IS NOT NULL AND classes.major_id = '" +
majorId + "'");
        if (resultSet.next()) {
            return resultSet.getFloat(1);
        } else {
            return 0;
        }
    } catch (SQLException e) {
        e.printStackTrace();
        return 0;
    }
}

```

c. Quản lý sinh viên (studentManagePanel)

Thành phần:

```

private javax.swing.JButton addStudentButton;
private javax.swing.JTextField addressInput;
private javax.swing.JLabel addressLabel;
private javax.swing.JTextField classIdInput;
private javax.swing.JLabel classIdLabel;
private javax.swing.JMenuItem deleteItem;
private javax.swing.JMenuItem editItem;

```

```

private javax.swing.JLabel footerLabel;
private javax.swing.JComboBox<String> genderInput;
private javax.swing.JLabel genderLabel;
private javax.swing.JPopupMenu popupMenu;
private javax.swing.JButton searchStudentButton;
private javax.swing.JTextField studentIdInput;
private javax.swing.JLabel studentIdlabel;
private javax.swing.JTextField studentNameInput;
private javax.swing.JLabel studentNameLabel;
private javax.swing.JTable studentTable;
private javax.swing.JScrollPane tableScrollPane;

```

STT	Mã sv	Mã lớp	Họ tên	Ngày sinh	Giới tính	Địa chỉ	SĐT
1	AT170120	AT17AK	Vũ Như Hoa	2002-09-20	Nam	Đồng Tháp	303230101
2	AT170121	AT17AT	Vũ Phú Hòa	2002-08-02	Nam	Lâm Đồng	735337209
3	AT170142	AT17AK	Vũ Tùng Sơn	2002-05-14	Nam	Quảng Ninh	919656939
4	AT170156	AT17AK	Kim Lâm Vũ	2002-10-18	Nam	Quảng Nam	526280722
5	AT170238	AT17BK	Vũ Quốc Phi	2002-10-09	Nam	Quảng Trị	175876491
6	AT170240	AT17BK	Vũ Duy Quân	2002-02-15	Nam	Bến Tre	761479133
7	AT170244	AT17BK	Vũ Trường Sơn	2003-12-15	Nam	Đồng Nai	139255748
8	AT170304	AT17CK	Vũ Quốc Anh	2002-07-16	Nam	Đắk Lắk	500685545
9	AT170357	AT17CT	Bùi Tuấn Vũ	2002-02-17	Nam	Hải Phòng	443923356
10	AT170420	AT17DP	Vũ Huy Hoàng	2002-09-06	Nam	Hải Dương	639112270
11	AT170437	AT17DK	Vũ Mạnh Phúc	2001-09-23	Nam	Thái Bình	160000433
12	AT170509	AT17EK	Vũ Thành Đạt	2002-09-03	Nam	Đồng Nai	835147947
13	AT170555	AT17EK	Vũ Quang Tuấn	2002-07-17	Nam	Bắc Ninh	176123955
14	AT170557	AT17EP	Nguyễn Minh Vũ	2002-06-26	Nam	Lào Cai	496001896
15	AT170609	AT17GK	Vũ Tiến Đạt	2002-02-23	Nam	Bạc Liêu	803754216
16	AT170635	AT17GK	Vũ Giang Nam	2002-09-12	Nam	Hà Nội	

Guest chỉ có quyền xem, không thể thêm, sửa, xóa.

Kiểm tra quyền trước khi thực hiện hành động:

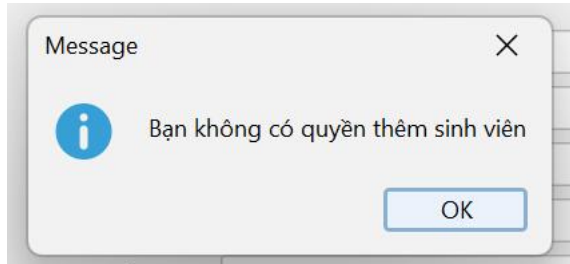
```

if (!UserSession.getIsAdmin()) {

```

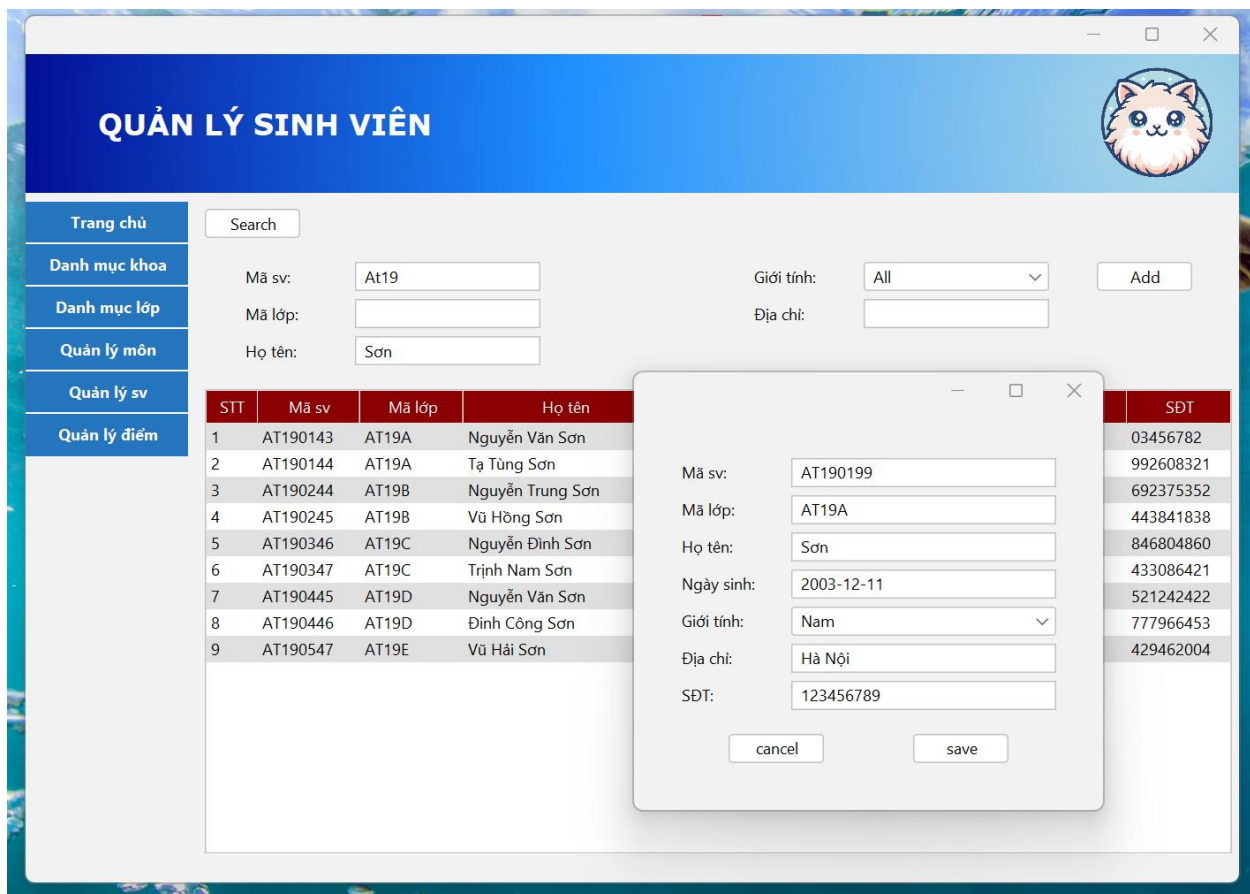
```
return null;
}
```

Thông báo

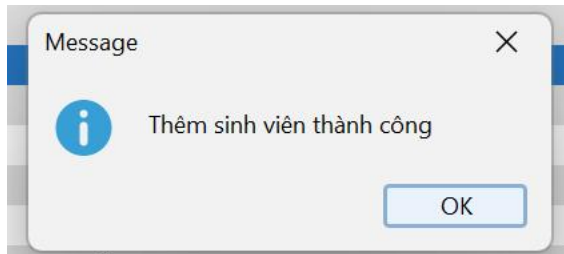


Thêm sinh viên:

Nhấn nút Add, có cửa sổ bật lên, điền thông tin
save để lưu, cancel để hủy



Thông báo



Bảng sinh viên được cập nhật ngay sau đó

STT	Mã sv	Mã lớp	Họ tên	Ngày sinh	Giới tính	Địa chỉ	SĐT
1	AT190143	AT19A	Nguyễn Văn Sơn	2004-04-12	Nam	Bạc Liêu	03456782
2	AT190144	AT19A	Tạ Tùng Sơn	2003-06-13	Nam	Tuyên Quang	992608321
3	AT190199	AT19A	Sơn	2003-12-11	Nam	Hà Nội	123456789
4	AT190244	AT19B	Nguyễn Trung Sơn	2003-10-27	Nam	Bắc Ninh	692375352
5	AT190245	AT19B	Vũ Hồng Sơn	2004-09-28	Nam	Bến Tre	443841838
6	AT190346	AT19C	Nguyễn Đình Sơn	2004-02-18	Nam	Hà Nam	846804860
7	AT190347	AT19C	Trịnh Nam Sơn	2003-12-25	Nam	Tiền Giang	433086421
8	AT190445	AT19D	Nguyễn Văn Sơn	2004-11-27	Nam	Hưng Yên	521242422
9	AT190446	AT19D	Đinh Công Sơn	2004-03-23	Nam	Hà Nam	777966453
10	AT190547	AT19E	Vũ Hải Sơn	2003-04-04	Nam	Gia Lai	429462004

Sửa sinh viên

Nhấn chuột phải vào sinh viên cần sửa, chọn edit

STT	Mã sv	Mã lớp	Họ tên	Ngày sinh	Giới tính	Địa chỉ	SĐT
1	AT190143	AT19A	Nguyễn Văn Sơn	2004-04-12	Nam	Bạc Liêu	03456782
2	AT190144	AT19A	Tạ Tùng Sơn	2003-06-13	Nam	Tuyên Quang	992608321
3	AT190199	AT19A	Sơn	2003-12-11	Nam	Hà Nội	123456789
4	AT190244	AT19B	Nguyễn Trung Sơn	2003-10-27	Nam	Bắc Ninh	692375352
5	AT190245	AT19B	Vũ Hồng Sơn	2004-09-28	Nam	Bến Tre	443841838
6	AT190346	AT19C	Nguyễn Đình Sơn	2004-02-18	Nam	Hà Nam	846804860

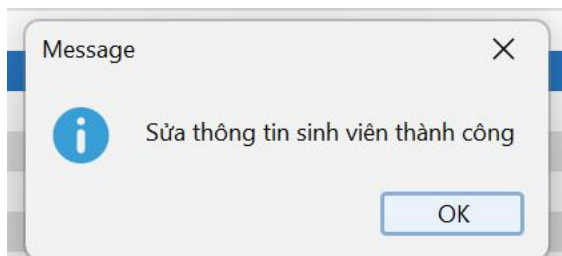
Sửa các trường cần thiết

Form for updating student information:

- Mã sv: AT190199
- Mã lớp: AT19B
- Họ tên: Sơn
- Ngày sinh: 2004-12-11
- Giới tính: Nam
- Địa chỉ: Thanh Hóa
- SĐT: 0000000000

Buttons: cancel, save

Save để lưu thay đổi, cancel để hủy

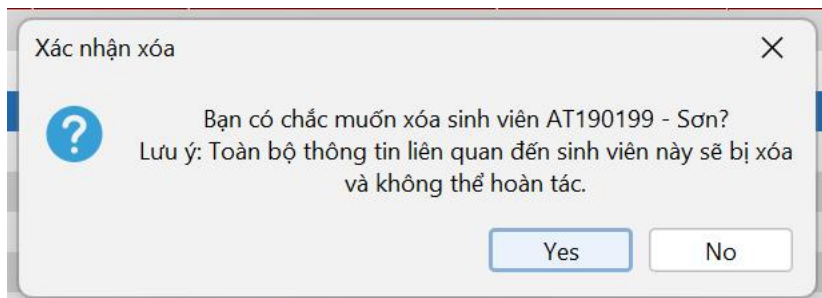


Bảng sinh viên được cập nhật ngay

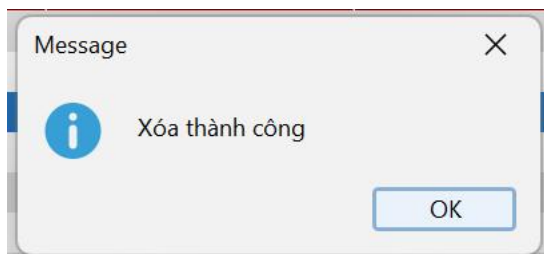
STT	Mã sv	Mã lớp	Họ tên	Ngày sinh	Giới tính	Địa chỉ	SĐT
1	AT190143	AT19A	Nguyễn Văn Sơn	2004-04-12	Nam	Bạc Liêu	03456782
2	AT190144	AT19A	Tạ Tùng Sơn	2003-06-13	Nam	Tuyên Quang	992608321
3	AT190199	AT19B	Sơn	2004-12-11	Nam	Thanh Hóa	0000000000
4	AT190244	AT19B	Nguyễn Trung Sơn	2003-10-27	Nam	Bắc Ninh	692375352

Xóa sinh viên

Nhấp chuột phải vào sinh viên cần xóa, chọn delete. Xác nhận



Thông báo



Bảng được cập nhật, không còn sinh viên đó nữa.

STT	Mã sv	Mã lớp	Họ tên	Ngày sinh	Giới tính	Địa chỉ	SĐT
1	AT190143	AT19A	Nguyễn Văn Sơn	2004-04-12	Nam	Bạc Liêu	03456782
2	AT190144	AT19A	Tạ Tùng Sơn	2003-06-13	Nam	Tuyên Quang	992608321
3	AT190244	AT19B	Nguyễn Trung Sơn	2003-10-27	Nam	Bắc Ninh	692375352
4	AT190245	AT19B	Vũ Hồng Sơn	2004-09-28	Nam	Bến Tre	443841838
5	AT190346	AT19C	Nguyễn Đình Sơn	2004-02-18	Nam	Hà Nam	846804860
6	AT190347	AT19C	Trịnh Nam Sơn	2003-12-25	Nam	Tiền Giang	433086421
7	AT190445	AT19D	Nguyễn Văn Sơn	2004-11-27	Nam	Hưng Yên	521242422
8	AT190446	AT19D	Đinh Công Sơn	2004-03-23	Nam	Hà Nam	777966453
9	AT190547	AT19E	Vũ Hải Sơn	2003-04-04	Nam	Gia Lai	429462004

Lưu ý: do tham chiếu khóa ngoại giữa các bảng, không thể thêm sinh viên vào lớp. Thay vào đó, cần tạo lớp trước. Tương tự với các trường hợp khác.

d. Quản lý điểm (scoreManagePanel)



Do đây là project học tập, nên nhóm muốn triển khai thêm các tiếp cận khác. Ở đây giao diện làm việc bố trí hơi khác so với phần sinh viên, nhưng chức năng, phần xử lý “backend” vẫn tương tự.

Vậy nhóm xin phép chỉ giới thiệu giao diện.

Các trường để thêm, sửa, xóa sẽ được cập nhật theo hàng người dùng chọn.

Người dùng với quyền admin có thể sửa các trường đó, nhấn Insert để thêm điểm, Update để sửa điểm hoặc Delete để xóa điểm theo Mã môn và mã sinh viên.

Update điểm:

ATCBTT3 Update

Toán cao cấp A3 Delete

9 Insert

Message

Sửa thông tin thành công

OK

Sau đó, bảng điểm được cập nhật

4	ATCBLH1	Vật lý đại cương A1	5.6
5	ATCBLH2	Vật lý đại cương A2	8.0
6	ATCBTT2	Toán cao cấp A2	4.3
7	ATCBTT3	Toán cao cấp A3	9.0
8	ATCBTT7	Toán cao cấp A1	5.7

Insert điểm:

ATCBTT3 Update

Toán cao cấp A3 Delete

9 Insert

Message

Thêm thông tin thành công

OK

Delete điểm

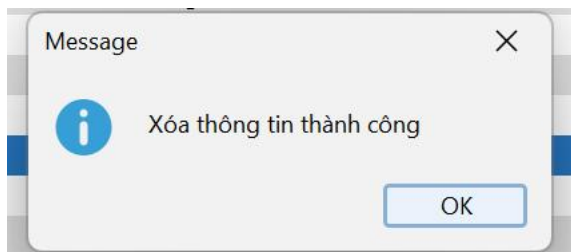
Xác nhận delete điểm:

Xóa điểm

Bạn có chắc chắn muốn xóa điểm môn Toán cao cấp A3 của sinh viên có mã số AT190111 không?

Yes No

Xóa thành công



Và bảng điểm được cập nhật

e. Quản lý môn (subjectManagePanel)

QUẢN LÝ SINH VIÊN

Trang chủ

Danh mục khoa

Danh mục lớp

Quản lý môn

Quản lý sv

Quản lý điểm

Search

Mã hp: AT

Số tín: 3

Add

Tên hp: Toán

Mã ngành: All

STT	Mã học phần	Mã ngành	Tên học phần	Số tín chỉ
1	ATATKH1	AT	Cơ sở an toàn thông tin	3
2	ATATPM3	AT	Công nghệ web an toàn	3
3	ATCBTT2	CB	án cao cấp A2	3
4	ATCBTT3	CB	án cao cấp A3	3
5	ATCBTT5	CB	án chuyên đề	3
6	ATCBTT7	CB	Toán cao cấp A1	3
7	CLC1ATATAT2	AT	An toàn mạng máy tính	3

f. Danh mục lớp (classManagePanel)

STT	Lớp	Khoa	Sĩ số	GVN
1	AT17CT	AT	19	Nguyễn Văn Thắng
2	CT06C	CT	4	Nguyễn Văn Quang
3	CT5A	CT	55	Lê Văn An
4	CT5B	CT	56	Lê Văn An
5	CT6B	CT	46	Nguyễn Văn Thắng
6	CT6C	CT	39	Nguyễn Văn Thắng
7	CT6D	CT	45	Lê Văn An
8	CT7A	CT	65	Nguyễn Văn Thắng
9	CT7C	CT	62	Trần Thị Thanh

IV. KẾT LUẬN

Hoàn thành chức năng cơ bản của ứng dụng.

Có thể mở rộng thêm các chức năng khác.

HẾT