

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA



BÁO CÁO BÀI TẬP LỚN
HỌC PHẦN LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
TÊN ĐỀ TÀI: QUẢN LÝ SINH VIÊN

| | |
|----------------------|----------------------------|
| Họ và tên sinh viên: | Nguyễn Trung Đức |
| Mã sinh viên: | 21012397 |
| Ngành học: | Công nghệ thông tin |
| Khóa: | K15 |
| Tên lớp: | KHMT |

Hà Nội, Tháng 10 Năm 2023

MỤC LỤC

| | | |
|------|---------------------------------------|----|
| I. | LỜI MỞ ĐẦU | 1 |
| 1. | Lý do chọn đề tài..... | 1 |
| 2. | Mục tiêu nghiên cứu | 1 |
| 3. | Phạm vi nghiên cứu | 1 |
| II. | NỘI DUNG | 3 |
| | CHƯƠNG 1. Cơ Sở Lý Thuyết..... | 3 |
| 1.1. | MySQL | 3 |
| 1.2. | Ngôn ngữ Java | 4 |
| | CHƯƠNG 2. Phân Tích Và Thiết Kế | 6 |
| 1. | Phân tích..... | 6 |
| 1.1. | <i>admin</i> | 6 |
| 1.2. | <i>student</i> | 6 |
| 1.3. | <i>course</i> | 7 |
| 1.4. | <i>score</i> | 7 |
| 2. | Thiết kế..... | 8 |
| 2.1. | <i>Login</i> | 8 |
| 2.2. | <i>Home</i> | 14 |
| III. | KẾT LUẬN | 34 |
| 1. | Ưu điểm..... | 34 |
| 2. | Nhược điểm | 34 |
| 3. | Hướng phát triển | 34 |

I. LỜI MỞ ĐẦU

1. Lý do chọn đề tài

Trong một thế giới ngày càng số hóa và liên quan đến công nghệ thông tin, việc quản lý thông tin trở nên quan trọng hơn bao giờ hết. Điểm số và thông tin liên quan đến học tập luôn luôn là một phần quan trọng của cuộc sống học tập. Chúng ta cần một cách hiệu quả để ghi, theo dõi, và quản lý thông tin điểm số của học sinh hoặc sinh viên. Đây là lý do chúng tôi quyết định phát triển một ứng dụng sử dụng lập trình hướng đối tượng để quản lý điểm số.

2. Mục tiêu nghiên cứu

Mục tiêu của dự án này là phát triển một ứng dụng quản lý điểm số đơn giản và mạnh mẽ, dựa trên lập trình hướng đối tượng. Ứng dụng sẽ cho phép người dùng thêm, sửa, xóa thông tin điểm số của học sinh hoặc sinh viên. Chúng ta cũng sẽ xây dựng một giao diện đồ họa để làm cho việc quản lý điểm số trở nên dễ dàng và thú vị.

3. Phạm vi nghiên cứu

Nghiên cứu sinh viên theo hệ thống tín chỉ, trong đó quản lý thông tin sinh viên theo ngành, ngày sinh, giới tính, số điện thoại, email, tên cha mẹ sinh viên, thông tin về điểm các môn để đưa ra gpa của sinh viên đó .

BẢNG CÁC TỪ VIẾT TẮT

| STT | TỪ VIẾT TẮT | VIẾT ĐẦY ĐỦ |
|------------|--------------------|--------------------|
| 1 | lb | jLabel |
| 2 | bt | jButton |
| 3 | tf | textField |
| 4 | pf | jPasswordField |
| 5 | tb | jTable |
| 6 | dc | jDateChooser |
| 7 | cb | jComboBox |
| 8 | pl | jPanel |
| 9 | R | Right |
| 10 | L | Left |

II. NỘI DUNG

CHƯƠNG 1. Cơ Sở Lý Thuyết

1.1. MySQL

Là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mạnh mẽ và phổ biến. Dưới đây là một số tính linh hoạt của MySQL:

Hỗ trợ nhiều loại dữ liệu: MySQL hỗ trợ nhiều loại dữ liệu bao gồm số nguyên, số thực, chuỗi, ngày tháng, và nhiều loại dữ liệu đặc biệt khác. Bạn có thể tạo bảng với các trường có kiểu dữ liệu tùy chỉnh.

Ngôn ngữ truy vấn SQL mạnh mẽ: MySQL hỗ trợ SQL, là ngôn ngữ truy vấn mạnh mẽ cho các thao tác truy xuất, cập nhật, thêm, và xóa dữ liệu.

Hỗ trợ giao dịch: MySQL hỗ trợ giao dịch, cho phép bạn thực hiện nhiều câu lệnh SQL trong một giao dịch và bảo đảm tính nhất quán và an toàn cho dữ liệu.

Đa nền tảng: MySQL hỗ trợ nhiều hệ điều hành và môi trường, cho phép bạn chạy trên nhiều nền tảng khác nhau.

Tính linh hoạt về lựa chọn lưu trữ: MySQL cho phép bạn chọn lựa chế độ lưu trữ dựa trên nhu cầu của bạn. Có nhiều lựa chọn lưu trữ như InnoDB, MyISAM, Memory, và nhiều lựa chọn khác.

Thư viện mở rộng: Cộng đồng MySQL phát triển nhiều thư viện mở rộng để bổ sung chức năng cho MySQL, ví dụ như JSON, GIS, và nhiều tính năng mở rộng khác.

Hỗ trợ các phiên bản mới của SQL: MySQL liên tục cải thiện và cập nhật để hỗ trợ các phiên bản mới của SQL và các tính năng nâng cao.

Khả năng chia dữ liệu và sao lưu: MySQL hỗ trợ các tính năng chia dữ liệu (sharding) và sao lưu dự phòng, giúp tăng hiệu suất và đảm bảo an toàn dữ liệu.

Tương thích với nhiều ngôn ngữ lập trình: MySQL có nhiều giao tiếp và thư viện kết nối cho nhiều ngôn ngữ lập trình, bao gồm Java, PHP, Python, Ruby, và nhiều ngôn ngữ khác.

Sự phát triển liên tục: MySQL là một dự án mã nguồn mở và có cộng đồng lớn, nên nó liên tục được cải thiện và phát triển với các phiên bản mới và bản vá bảo mật.

Tính linh hoạt của MySQL cho phép bạn sử dụng nó trong nhiều ứng dụng và tình huống khác nhau, từ ứng dụng web đơn giản đến hệ thống doanh nghiệp phức tạp.

1.2. Ngôn ngữ Java

Máy ảo Java (JVM – Java Virtual Machine)

Tất cả các chương trình muốn thực thi được thì phải được biên dịch ra mã máy. Mã máy của từng kiến trúc CPU của mỗi máy tính là khác nhau (tập lệnh mã máy của CPU Intel, CPU Solarix, CPU Macintosh... là khác nhau), vì vậy trước đây là một trướng chình sau khi được biên dịch xong thì có thể chạy được trên một kiến trúc CPU cụ thể nào đó

Chương trình thực thi được trên Windows được biên dịch dưới dạng file có đuôi ‘.exe’ còn trên Linux thì được biên dịch dưới dạng file có đuôi là ‘.elf’ vì vậy trước đây chạy trướng trình trên Windows muốn chạy được trên Linux cũng như các hệ điều hành khác thì ta phải chỉnh sửa và biên dịch lại.

Ngôn ngữ lập trình Java ra đời nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một trướng chình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java(mã java bytecode). Sau đó máy ảo Java chỉ trách nhiệm chuyển mã java bytecode thành mã máy tương ứng.

Sun Microsystem chịu trách nhiệm phát triển các máy ảo java chạy trên hệ điều hành trên các kiến trúc CPU khác nhau.

Thông dịch

Java là một ngôn ngữ lập trình vừa được biên dịch và thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi “.java” đầu tiên được biên dịch thành tập tin có đuôi “.class” và sau đó được trình thông dịch thông dịch thành mã máy.

Độc lập nền

Một chương trình viết bằng ngôn ngữ Java có thể chạy trên nhiều máy tính có hệ điều hành khác nhau (Windows, Unix, Linux...) với điều kiện ở đó có cài đặt máy ảo java (Java Virtual Machine)

Hướng đối tượng

Hướng đối tượng trong Java tương tự như C++ nhưng java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí chính một chương trình viết bằng Java (đó là main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ đa kế thừa

Đa nhiệm – đa luồng

Java hỗ trợ lập trình đa nhiệm , đa luồng cho phép nhiều tiến trình, tiểu tiến trình có thể chạy song song cùng một thời điểm và tương tác với nhau.

CHƯƠNG 2. Phân Tích Và Thiết Kế

1. Phân tích

Xác định các lớp của bài toán:

Ở phần tạo các class cho bài toán, tôi không tạo các lớp của các đối tượng với ngôn ngữ Java mà tôi tạo các lớp này trong MySQL để dễ cho việc truy xuất và quản lý dữ liệu.

Thiết kế cơ sở dữ liệu

1.1.admin

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
|-----|------------|-------------------|---------------|
| 1 | id | int (primary key) | số thứ tự |
| 2 | username | varchar(45) | tên đăng nhập |
| 3 | password | varchar(45) | mật khẩu |

Bảng admin 1

1.2.student

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
|-----|---------------|-------------------|---------------------|
| 1 | <u>id</u> | int (primary key) | số thứ tự |
| 2 | name | varchar(120) | tên sinh viên |
| 3 | date_of_birth | date | ngày tháng năm sinh |
| 4 | gender | varchar(10) | giới tính |
| 5 | email | varchar(100) | email |
| 6 | phone | varchar(12) | số điện thoại |
| 7 | farther_name | varchar(150) | họ tên bố |
| 8 | mother_name | varchar(150) | họ tên mẹ |
| 9 | address1 | text | địa chỉ sinh sống 1 |
| 10 | address2 | text | địa chỉ sinh sống 2 |
| 11 | imge_path | varchar(200) | ảnh |

Bảng Student 1

1.3.course

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
|-----|-------------------|-------------------|-----------------|
| 1 | <u>id</u> | int (primary key) | số thứ tự |
| 2 | <u>student_id</u> | int (| mã số sinh viên |
| 3 | course1 | vachar(200) | Môn học 1 |
| 4 | course2 | vachar(200) | Môn học 2 |
| 5 | course3 | vachar(200) | Môn học 3 |
| 6 | course4 | vachar(200) | Môn học 4 |
| 7 | course5 | vachar(200) | Môn học 5 |

Bảng Course 1

1.4.score

| STT | Tên trường | Kiểu dữ liệu | Chú thích |
|-----|-------------------|-------------------|-----------------|
| 1 | <u>id</u> | int (primary key) | Số thứ tự |
| 2 | <u>student_id</u> | int (| Mã số sinh viên |
| 3 | semester | int | Mã giảng viên |
| 4 | course1 | vachar(200) | Môn học 1 |
| 5 | score1 | double | Điểm môn 1 |
| 6 | course2 | vachar(200) | Môn học 2 |
| 7 | score2 | double | Điểm môn 2 |
| 8 | course3 | vachar(200) | Môn học 3 |
| 9 | score3 | double | Điểm môn 3 |
| 10 | course4 | vachar(200) | Môn học 4 |
| 11 | score4 | double | Điểm môn 4 |
| 12 | course5 | vachar(200) | Môn học 5 |
| 13 | score5 | double | Điểm môn 5 |
| 14 | average | double | Điểm trung bình |

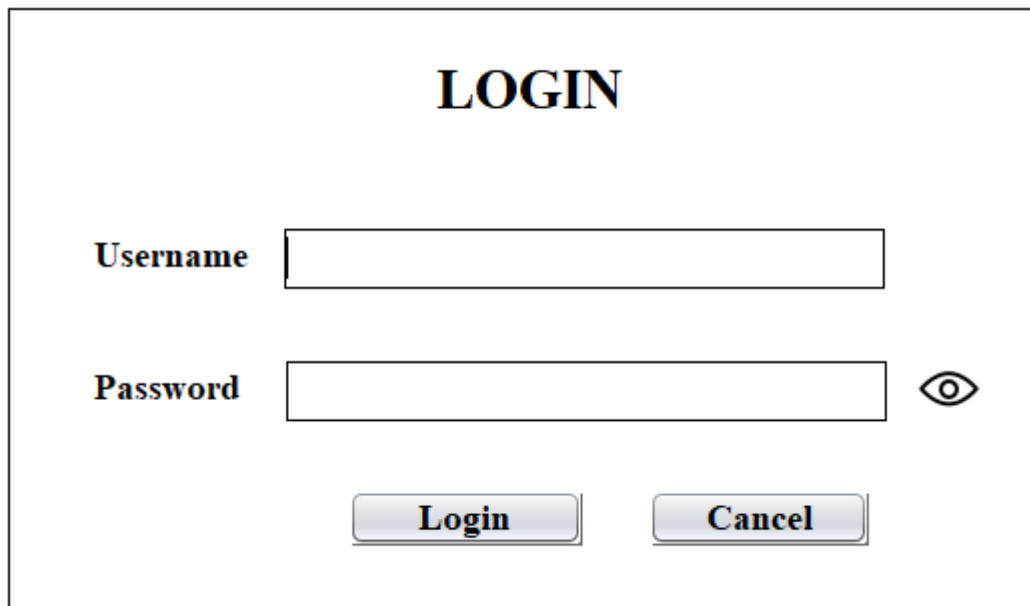
Bảng Score 1

2. Thiết kế

2.1.Login

2.1.1. Tổng quan

Phần mềm có login gồm:username và password, người dùng phải đăng nhập đúng username và password với vào được hệ thống.



The diagram shows a login window titled "LOGIN". It contains two input fields: "Username" and "Password". The "Password" field has an eye icon to its right, indicating a toggle for password visibility. Below the input fields are two buttons: "Login" and "Cancel".

Login 1

Người dùng có thể kiểm tra lại mật khẩu của mình bằng các kích và biểu tượng hình con mắt để xem mật khẩu.

2.1.2. Code Login

Sau khi ta phân tích ở trên dựa theo dữ kiện đó để thực hiện chúng dưới code. Chúng ta phân tích càng rõ ràng chi tiết vậy thì khi thực hiện viết mã ít bị lỗi hơn.

Code:

```
create database student_management;  
use student_management;
```

Login 2


Chúng ta đang quản lý sinh viên vậy ta cần tạo một database có tên “student_management” trong mysql để lưu trữ dữ liệu. Database này sẽ các bảng về “admin”, “student”, “course”, “score”. Phần này ta đang cần tạo một bảng dữ liệu để lưu thông tin đăng nhập lên ta chắc chắn phải bảng “admin”

```
create table admin(  
    id int not null auto_increment,  
    username varchar(45) not null,  
    password varchar(45) not null,  
    primary key(`id`)  
);
```

Login 3

Sau khi ta đã code song với với mysql bây giờ ta sang bên app Netsbean để làm giao diện cho phần này

LOGIN

| | |
|--------------------------------------|--|
| Username | <input type="text"/> |
| Password | <input type="password"/>  |
| <input type="button" value="Login"/> | <input type="button" value="Cancel"/> |

Login 4

Đầu tiên ta cần phải tạo package ‘login’ và ‘db’

Để kết nối được với mysql bạn cần phải có code kết nối từ netsbean. Dữ liệu ta đang được lưu trên mysql việc kết nối này giúp ta truy xuất được dữ liệu từ mysql về để sử dụng. Đoạn mã kết nối như sau

```

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

public class MyConnection {

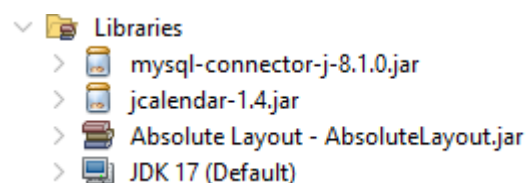
    public static Connection getConnection() {
        String username = "root";
        String password = "mysql";
        String url = "jdbc:mysql://localhost:3306/student_management";
        try {
            return DriverManager.getConnection(url, user: username, password);
        } catch (SQLException ex) {
            System.out.println(x: ex);
        }
        return null;
    }
}

```

Login 5

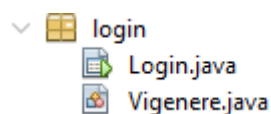
Với ‘username, password’ bạn lên xem ‘username, password’ của bạn trong mysql là gì để kết nối cho chính xác. Nếu bạn dùng mysql thường thì cổng mặc định là 3306

Chú ý: muốn kết nối được bạn cần phải tải thêm thư viện để kết nối. Thư viện kết nối “mysql-connector-j-8.1.0.jar”



Login 6

Trong package ‘login’ đó ta new mới một JFrame Form đặt tên là ‘Login’ để thiết kế giao diện đăng nhập như trên và một class mã hoá có tên ‘Vigenere’.



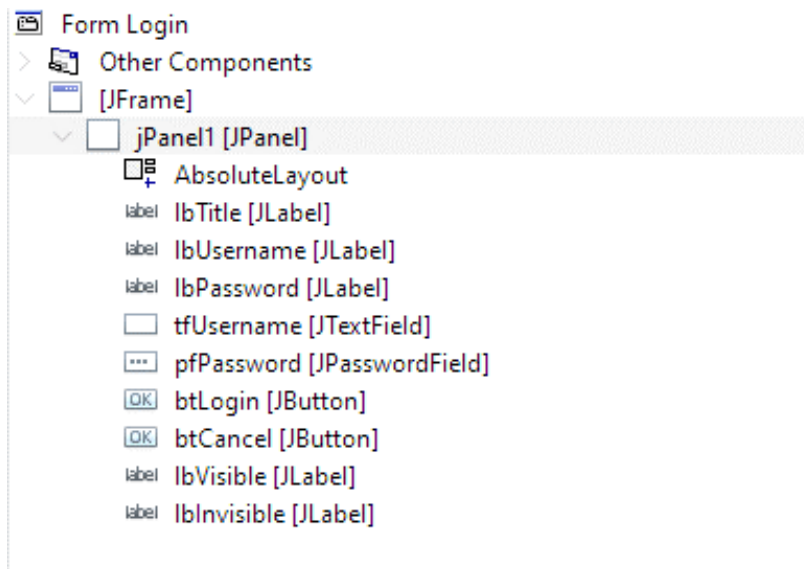
Login 7

Trường thông tin được sử dụng như sau:

“LOGIN, Username, Password, icon” : JLabel

Dữ liệu được lấy từ hình chữ nhật bên cạnh phải của Username, Password là: jTextField

“Login, Cancel”: JButton



Login 8

Ta sẽ bắt điều kiện khi dữ không nhập thông tin sẽ thông báo cho người dùng họ nhập thông tin vào để đăng nhập nếu dữ liệu không trống thì trả về True còn ngược lại False.

```
public boolean isEmpty() {  
    if (tfUsername.getText().isEmpty()) {  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Username is missing");  
        return false;  
    }  
    if (String.valueOf(data: pfPassword.getPassword()).isEmpty()) {  
        JOptionPane.showMessageDialog(parentComponent: this, message: "Passworld is missing");  
        return false;  
    }  
    return true;  
}
```

Login 9

Với trường pfPassword khi người dùng kích vào hình con mắt tôi sẽ xét như sau:

```
private void lbVisibleMouseClicked(java.awt.event.MouseEvent evt) {  
    pfPassword.setEchoChar((char) 0);  
    lbVisible.setVisible(aFlag: false);  
    lbInvisible.setVisible(aFlag: true);  
}  
  
private void lbInvisibleMouseClicked(java.awt.event.MouseEvent evt) {  
    pfPassword.setEchoChar(c: '*');  
    lbVisible.setVisible(aFlag: true);  
    lbInvisible.setVisible(aFlag: false);  
}
```

Login 10

khi ấn biểu tượng hình con mắt thì mật khẩu hiện ra cho người dùng xem lại và nhấp chuột lại thì mật khẩu được che dấu đi bởi ký tự ‘*’.

Sau đó để đăng nhập được người dùng phải có đăng nhập đúng tài khoản mật khẩu. Ta sử dụng java để kết nối với mysql để trích xuất dữ liệu và so sánh với dữ liệu người dùng nhập nếu đúng thì sẽ đưa người dùng vào giao diện home còn không thì người dùng không vào được. Người dùng có thể xem lại mật khẩu của mình bằng kích chuột vào biểu tượng hình con mắt.

```
private void btLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    if (isEmpty()) {  
        String username = vg.encryptVigenere(plaintext: tfUsername.getText(), key);  
        String password = vg.encryptVigenere(plaintext: String.valueOf(data: pfPassword.getPassword()), key);  
        try {  
            ps = con.prepareStatement(sql: "select * from admin where username = ? and password = ?");  
            ps.setString(parameterIndex: 1, x: username);  
            ps.setString(parameterIndex: 2, x: password);  
            ResultSet rs = ps.executeQuery();  
            if (rs.next()) {  
                JOptionPane.showMessageDialog(parentComponent: this, message: "Login success");  
                Home home = new Home();  
                home.setVisible(b: true);  
                this.dispose();  
            } else {  
                JOptionPane.showMessageDialog(parentComponent: this, message: "Invalid username or password!");  
            }  
        } catch (SQLException ex) {  
            Logger.getLogger(Login.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);  
        }  
    }  
}
```

Login 11

Để tránh việc bị mất thông tin mật khẩu và tài khoản cần phải được mã hoá ở trong trong này tôi dùng mã hoá “Vigenere” có nhiều loại mã hoá hay hơn có thể sử dụng như RSA, MD5...

2.1.2.1.Vigenere

Mã hoá

```
public String encryptVigenere(String plaintext, String key) {
    String encode = "";
    int textLength = plaintext.length();
    int keyLength = key.length();
    int count = 0;
    for (int i = 0; i < textLength; i++) {
        char textChar = plaintext.charAt(index: i);
        char keyChar = key.charAt(index: count);
        if (textChar > 96 && textChar < 123) {
            char c = (char) (((textChar - 97 + keyChar - 97) % 26) + 97);
            encode += c;
        }
        if (textChar > 64 && textChar < 91) {
            char c = (char) (((textChar - 65 + keyChar - 65) % 26) + 65);
            encode += c;
        }
        count++;
        if (count == keyLength) {
            count = 0;
        }
    }
    return encode;
}
```

Login 12

Giải mã

```

public String decryptVigenere(String plaintext, String key) {
    String decode = "";
    int textLength = plaintext.length();
    int keyLength = key.length();
    int count = 0;
    for (int i = 0; i < textLength; i++) {
        char textChar = plaintext.charAt(index: i);
        char keyChar = key.charAt(index: count);
        if (textChar > 96 && textChar < 123) {
            char c = (char) (((textChar - 97) - (keyChar - 97) + 26) % 26) + 97);
            decode += c;
        }
        if (textChar > 64 && textChar < 91) {
            char c = (char) (((textChar - 65) - (keyChar - 65) + 26) % 26) + 65);
            decode += c;
        }
        count++;
        if (count == keyLength) {
            count = 0;
        }
    }
    return decode;
}

```

Login 13

2.2.Home

2.2.1. Tổng quan

Ở dao diện đầu tiên là thông tin chung vì mặc đề là Student lên khi đăng nhập thành công sẽ hiện ra dao diện của Student .

STUDENT MANAGEMENT SYSTEM

| Student | Course | Score | Marks Sheet | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|--|---------------|-------------|--------------|----------------|---------------|---------------|----------------|----------------|----------------|---------------|----------------|----------------|------------|---|------------|------------|--------|--------------|------------|-----|------|----------|----------|----------------|---|---------|------------|------|--------------|------------|------|------|---------|--------|----------------|---|----------|------------|------|--------------|------------|---------|---------|--------|--------|----------------|---|---------|------------|------|-------------|------------|---------|-----------|-----------|--------|----------------|---|----------|------------|------|--------------|------------|---------|-----------|-----------|--------|----------------|
| <p>Student's ID: 7</p> <p>Student's name: <input type="text"/></p> <p>Date Of Birth: <input type="text"/></p> <p>Gender: <input type="text"/></p> <p>Email: <input type="text"/></p> <p>Phone: <input type="text"/></p> <p>Father's Name: <input type="text"/></p> <p>Mother's Name: <input type="text"/></p> <p>Address Line 1: <input type="text"/></p> <p>Address Line 2: <input type="text"/></p> <p>Image <input type="text"/></p> <p><input type="button" value="Browes"/></p> | <p>Search Student <input type="text"/></p> <p><input type="button" value="Search"/> <input type="button" value="Refresh"/></p> <table border="1"> <thead> <tr> <th>Students ID</th> <th>Students na...</th> <th>Date of Birth</th> <th>Gender</th> <th>Email</th> <th>Phone</th> <th>Father's Name</th> <th>Mother's Name</th> <th>Address Lin...</th> <th>Address Lin...</th> <th>Image path</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>Thanh Ngan</td> <td>2023-10-01</td> <td>Female</td> <td>thanh@gma...</td> <td>1234567890</td> <td>Nam</td> <td>Hanh</td> <td>Hoa Binh</td> <td>Hoa Binh</td> <td>C:\Users\Ad...</td> </tr> <tr> <td>3</td> <td>Hai Anh</td> <td>2023-08-13</td> <td>Male</td> <td>haianh@ga...</td> <td>4113411333</td> <td>Hieu</td> <td>Thao</td> <td>Nghe An</td> <td>Ha Noi</td> <td>C:\Users\Ad...</td> </tr> <tr> <td>4</td> <td>Duc Minh</td> <td>2023-05-10</td> <td>Male</td> <td>ducminh@g...</td> <td>0345345113</td> <td>Van Nam</td> <td>Thi Hoa</td> <td>Ha Nam</td> <td>Ha Noi</td> <td>C:\Users\Ad...</td> </tr> <tr> <td>5</td> <td>Quy Anh</td> <td>2023-10-02</td> <td>Male</td> <td>quyanh@g...</td> <td>1111000111</td> <td>Ba Hien</td> <td>Huynh Yen</td> <td>Quang Tri</td> <td>Ha Noi</td> <td>C:\Users\Ad...</td> </tr> <tr> <td>6</td> <td>Thuy Hoa</td> <td>2023-06-06</td> <td>Male</td> <td>hoa@gmail...</td> <td>4445556668</td> <td>Ba Hieu</td> <td>Thi Nhung</td> <td>Hai Duong</td> <td>Ha Noi</td> <td>C:\Users\Ad...</td> </tr> </tbody> </table> <p><input type="button" value="Add"/> <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="Print"/> <input type="button" value="Clear"/> <input type="button" value="Logout"/></p> | | | Students ID | Students na... | Date of Birth | Gender | Email | Phone | Father's Name | Mother's Name | Address Lin... | Address Lin... | Image path | 2 | Thanh Ngan | 2023-10-01 | Female | thanh@gma... | 1234567890 | Nam | Hanh | Hoa Binh | Hoa Binh | C:\Users\Ad... | 3 | Hai Anh | 2023-08-13 | Male | haianh@ga... | 4113411333 | Hieu | Thao | Nghe An | Ha Noi | C:\Users\Ad... | 4 | Duc Minh | 2023-05-10 | Male | ducminh@g... | 0345345113 | Van Nam | Thi Hoa | Ha Nam | Ha Noi | C:\Users\Ad... | 5 | Quy Anh | 2023-10-02 | Male | quyanh@g... | 1111000111 | Ba Hien | Huynh Yen | Quang Tri | Ha Noi | C:\Users\Ad... | 6 | Thuy Hoa | 2023-06-06 | Male | hoa@gmail... | 4445556668 | Ba Hieu | Thi Nhung | Hai Duong | Ha Noi | C:\Users\Ad... |
| Students ID | Students na... | Date of Birth | Gender | Email | Phone | Father's Name | Mother's Name | Address Lin... | Address Lin... | Image path | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Thanh Ngan | 2023-10-01 | Female | thanh@gma... | 1234567890 | Nam | Hanh | Hoa Binh | Hoa Binh | C:\Users\Ad... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Hai Anh | 2023-08-13 | Male | haianh@ga... | 4113411333 | Hieu | Thao | Nghe An | Ha Noi | C:\Users\Ad... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Duc Minh | 2023-05-10 | Male | ducminh@g... | 0345345113 | Van Nam | Thi Hoa | Ha Nam | Ha Noi | C:\Users\Ad... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Quy Anh | 2023-10-02 | Male | quyanh@g... | 1111000111 | Ba Hien | Huynh Yen | Quang Tri | Ha Noi | C:\Users\Ad... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Thuy Hoa | 2023-06-06 | Male | hoa@gmail... | 4445556668 | Ba Hieu | Thi Nhung | Hai Duong | Ha Noi | C:\Users\Ad... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Home 1

Muốn chuyển đến các giao diện khác người dùng có thể kích chọn vào:

2.2.1.1.Student

Khi chọn giao diện người dùng có thể nhập thông tin của sinh viên, khi nhập sai người dùng có thể chỉnh sửa lại, lưu, thêm, xóa thông qua giao diện

STUDENT MANAGEMENT SYSTEM

Student

Course

Score

Marks Sheet

Student's ID

7

Student's name:

Date Of Birth

Gender

Male

Email

Phone

Father's Name

Mother's Name

Address Line 1

Address Line 2

Image

Browse

Search Student

Search

Refresh

| Students ID | Students na... | Date of Brith | Gender | Email | Phone | Father Name | Mother Name | Address Lin... | Address Lin... | Image path |
|-------------|----------------|---------------|--------|--------------|------------|-------------|-------------|----------------|----------------|----------------|
| 2 | Thanh Ngan | 2023-10-01 | Female | thanh@gma... | 1234567890 | Nam | Hanh | Hoa Binh | Hoa Binh | C:\Users\Ad... |
| 3 | Hai Anh | 2023-08-13 | Male | haianh@ga... | 4113411333 | Hieu | Thao | Nghe An | Ha Noi | C:\Users\Ad... |
| 4 | Duc Minh | 2023-05-10 | Male | ducminh@g... | 0345345113 | Van Nam | Thi Hoa | Ha Nam | Ha Noi | C:\Users\Ad... |
| 5 | Quy Anh | 2023-10-02 | Male | quyanh@g... | 1111000111 | Ba Hien | Huynh Yen | Quang Tri | Ha Noi | C:\Users\Ad... |
| 6 | Thuy Hoa | 2023-06-06 | Male | hoa@gmail... | 4445556668 | Ba Hieu | Thi Nhung | Hai Duong | Ha Noi | C:\Users\Ad... |

+ Add

Update

✖ Delete

Print

Clear

Logout

Student 1

2.2.1.2.Course

Khi được chọn vào giao diện sẽ hiện ra các khoá học để cho người dùng đăng ký thông qua giao diện

STUDENT MANAGEMENT SYSTEM

Student
Course
Score
Marks Sheet

Student's ID

Search

ID

Student's ID

Semester

Course 1

Course 2

Course 3

Course 4

Course 5

Search Student

| ID | Students ID | Semester | Course 1 | Course 2 | Course 3 | Course 4 | Course 5 |
|----|-------------|----------|----------|---------------------|--------------|----------------|------------------|
| 9 | 6 | 2 | C# | Computer Science | Linux | Java OOP | Machine Learning |
| 8 | 6 | 1 | Python | Digital Image Pr... | Big Data | Web Programing | Deep Learning |
| 7 | 5 | 2 | C++ | Computer Science | Big Data | Web Programing | Could Computing |
| 6 | 5 | 1 | C | Computer NetW... | Database | Java OOP | Machine Learning |
| 5 | 4 | 1 | C | Computer NetW... | Database | Java OOP | Machine Learning |
| 4 | 3 | 2 | C++ | Computer NetW... | Linux | Java OOP | Machine Learning |
| 3 | 3 | 1 | C# | Digital Image Pr... | Big Data | Web Programing | Could Computing |
| 2 | 2 | 1 | C++ | Computer Science | Data Science | JSP | Deep Learning |

Save
 Print
 Clear
 Logout

Course 1

2.2.1.3.Score

Khi được lựa chọn sẽ hiện ra các khoá học đã đăng ký và người dùng có thể nhập điểm cho các môn đã đăng ký, nếu nhập sai có thể kích chọn vào sinh viên trên bảng và chỉnh sửa lại thông tin sinh viên đó thông qua giao diện.

STUDENT MANAGEMENT SYSTEM

Student
Course
Score
Marks Sheet

Student's ID

Search

Semester

ID

Student's ID

Semester

Course 1

Course 2

Course 3

Course 4

Course 5

Search Student

| ID | StudentID | Semester | Course1 | Score1 | Course2 | Score2 | Course3 | Score3 | Course4 | Score4 | Course5 | Score5 | Average |
|----|-----------|----------|---------|--------|--------------|--------|-------------|--------|------------|--------|------------|--------|---------|
| 5 | 2 | 1 | C++ | 3.0 | Comput... | 3.0 | Data Sci... | 3.7 | JSP | 3.6 | Deep Le... | 3.8 | 3.66 |
| 8 | 3 | 1 | C# | 4.0 | Digital I... | 3.0 | Big Data | 3.5 | Web Pro... | 3.6 | Could C... | 3.8 | 3.58 |
| 9 | 2 | 1 | C++ | 3.5 | Comput... | 3.4 | Data Sci... | 3.0 | JSP | 3.6 | Deep Le... | 3.8 | 3.46 |
| 10 | 2 | 1 | C++ | 3.8 | Comput... | 3.6 | Data Sci... | 3.0 | JSP | 3.0 | Deep Le... | 3.3 | 3.34 |

Save
 Update
 Print
 Sort
 Clear
 Logout

Score 1

2.2.1.4.Marks Sheet

Phần dao diện còn lại này để tìm kiếm sinh viên dựa trên id của họ giao diện cho tao thấy rõ nhất về điểm trung bình của sinh viên thông qua giao diện.

STUDENT MANAGEMENT SYSTEM

Student Course Score Marks Sheet

Student's ID

| ID | StudentID | Semester | Course1 | Score1 | Course2 | Score2 | Course3 | Score3 | Course4 | Score4 | Course5 | Score5 | Average |
|----|-----------|----------|---------|--------|--------------|--------|----------|--------|------------|--------|------------|--------|---------|
| 8 | 3 | 1 | C# | 4.0 | Digital I... | 3.0 | Big Data | 3.5 | Web Pro... | 3.6 | Could C... | 3.8 | 3.58 |

Print

Clear

Logout

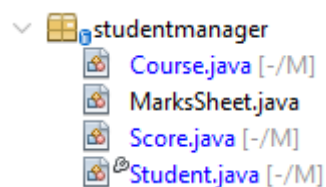
GPA: 3.58

Marks Sheet 1

2.2.2. Code Home

Việc đầu tiên ta vẫn cần phải tạo một package mới có tên là “student_manager” và “home”.

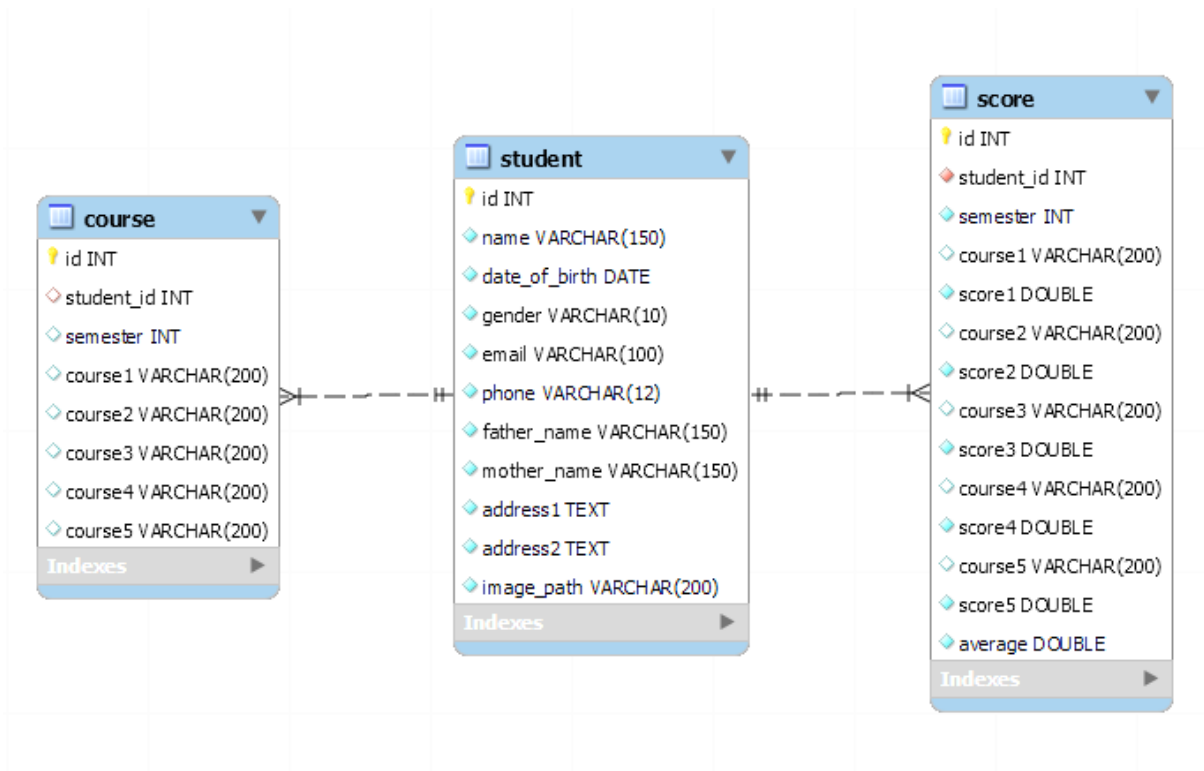
Trong package ‘student_manager’ tạo 4 class “Student”, “Course”, “Score”, “Marks Sheet”.



studentmangager 1

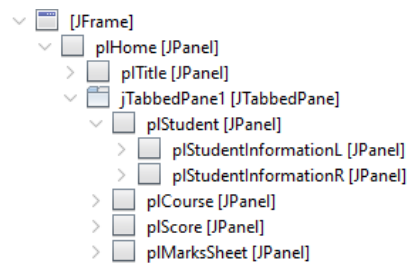
Package này là nơi để ta truy xuất và dự trữ thông tin của sinh viên từ Java với MySQL. Code ở trong package này giúp cho ta nhanh chóng truy xuất, sửa đổi, bổ sung thông tin của sinh viên khi ta lựa chọn các button trong package “home”

Sang bên MySQL ta cần phải tạo thêm các bảng dữ liệu “student”, “course”, “score” để lưu trữ thông tin này, các bảng này được liên kết với nhau như hình vẽ



studentmanager 2

Trong package “home” được thiết kế như hình dưới



Home 2

Ta tạo đầu tiên là jTablePanel nhằm mục đích để lưu trữ các JPanel các JPanel này là nơi để thiết kế ‘Student’, ‘Course’, ‘Score’, ‘Marks Sheet’.

Phần JFrame cài đặt Events → Window → WindowOpened

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    for (double i = 0.1; i < 1.0; i += 0.1) {
        String s = i + "";
        float f = Float.parseFloat(s);
        this.setOpacity(opacity:f);
        try {
            Thread.sleep(millis: 30);
        } catch (InterruptedException ex) {
            Logger.getLogger(name: Home.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
        }
    }
}
```

Home 3

mục đích là tạo hiệu ứng chuyển đổi độ mờ của theo từng giai đoạn và tạm dừng mỗi giai đoạn để hiển thị hiệu ứng trước đó.

plHome cài đặt Events -> Mouse -> mousePressed

```
int xx, xy;
```

```
private void plHomeMousePressed(java.awt.event.MouseEvent evt) {
    xx = evt.getX();
    xy = evt.getY();
}
```

Home 4

Khi người dùng nhấn chuột, đoạn mã này sẽ lấy điểm x và y của nút bấm và gán chúng vào các biến xx và xy. Sau đó, bạn có thể sử dụng giá trị này để thực hiện các công cụ hành động, ví dụ như vẽ một hình ảnh tại điểm này hoặc xử lý các đặc biệt tương tác liên quan đến vị trí mà người dùng nhấn chuột.

Events -> MouseMotion -> mouseDragged

```
private void plHomeMouseDragged(java.awt.event.MouseEvent evt) {
    int x = evt.getXOnScreen();
    int y = evt.getYOnScreen();
    this.setLocation(x - xx, y - xy);
}
```

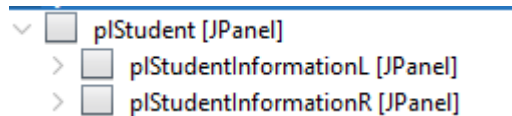
Home 5

Đoạn mã trong sự kiện mouseDragged được sử dụng để thực hiện việc di chuyển cửa sổ hoặc thành phần giao diện người dùng (GUI) dựa trên sự di chuyển của chuột. Cửa sổ hoặc thành phần sẽ được di chuyển đến vị trí mới. Điều này tạo ra

hiệu ứng kéo thả (drag and drop) khi người dùng di chuyển chuột và cửa sổ hoặc thành phần được di chuyển theo đó.

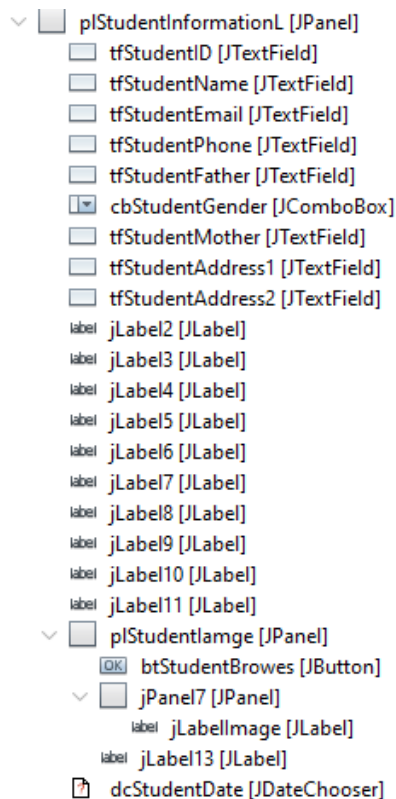
2.2.2.1.Student

Trong plStudent ta tạo ra thêm hai JPanel mới và được đặt tên là ‘plStudentInforamtionL’ và ‘plStudentInforamtionR’



Student 2

Các trường thông tin trong plStudentInforamtionL nơi nhập thông tin của sinh viên được thiết kế như sau



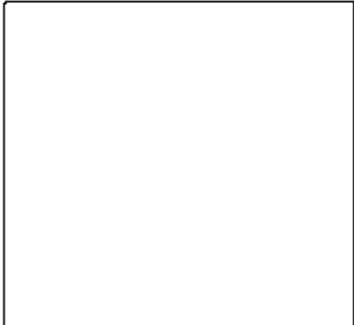


Student 3

Đối với ‘tfStudentID’ được cài đặt không thể thay đổi dữ liệu được ID của sinh viên khi được thêm sinh viên mới vào sẽ tự động tăng lên.

tfStudentID -> properties -> không tích chọn editable

JDateChooser không có trong app netsbean, muốn sử dụng được phải tải thư viện này về và thêm chúng vào sử dụng. Gói thư viện cần tải là ‘jcalendar-1.4’

| | |
|---|---|
| Student's ID | <input type="text"/> |
| Student's name: | <input type="text"/> |
| Date Of Birth | <input type="text"/>  |
| Gender | <input type="text" value="Male"/>  |
| Email | <input type="text"/> |
| Phone | <input type="text"/> |
| Father's Name | <input type="text"/> |
| Mother's Name | <input type="text"/> |
| Address Line 1 | <input type="text"/> |
| Address Line 2 | <input type="text"/> |
| <div><div><i>Image</i> <input type="button" value="Browse"/></div><div></div></div> | |

Student 4

‘**dcStudentDate**’ khi nhập dữ liệu năm sinh của sinh viên ở Date Of Birth sẽ được định dạng “yyyy-MM-dd”

‘**tfStudentPhone**’ khi người dùng nhập ký tự không phải là chữ số thì sẽ không nhập được và thông báo cho người dùng phải nhập số. Cài đặt như sau Events → Key → keyType

```
private void tfStudentPhoneKeyTyped(java.awt.event.KeyEvent evt) {
    if (!Character.isDigit(ch: evt.getKeyChar())) {
        evt.consume();
        JOptionPane.showMessageDialog(parentComponent: this, message: "The phone number must be a number!");
    }
}
```

Student 5

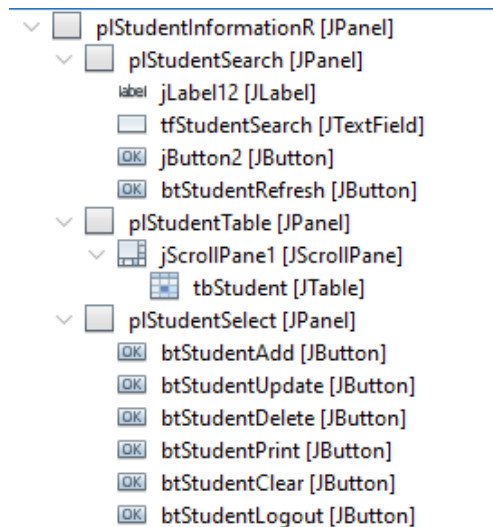
‘tfStudentEmail’ nếu người dùng nhập email không đúng thì thông tin sinh viên sẽ không được thêm vào, thông báo lỗi sẽ được hiện ra.

‘image’ ở đây ta cho người dùng nhập ảnh của mình, đường dẫn của ảnh sẽ được lưu lại. Khi người dùng không nhập ảnh thì sẽ thông báo cho người dùng là không có ảnh

```
private void btStudentBrowesActionPerformed(java.awt.event.ActionEvent evt) {
    JFileChooser file = new JFileChooser();
    file.setCurrentDirectory(new File(pathname: System.getProperty(key: "user.home")));
    FileNameExtensionFilter filter = new FileNameExtensionFilter(description: "*image",
        extensions: "jpg", extensions: "gif", extensions: "png");
    file.addChoosableFileFilter(filter);
    int output = file.showSaveDialog(parent: file);
    if (output == JFileChooser.APPROVE_OPTION) {
        File selectFile = file.getSelectedFile();
        String path = selectFile.getAbsolutePath();
        jLabelImage.setIcon(icon: imageAdjust(path, pic: null));
        imagePath = path;
    } else {
        JOptionPane.showMessageDialog(parentComponent: this, message: "No image selected");
    }
}
```

Student 6

Đối với plStudentInforamtionR được thiết kế như sau



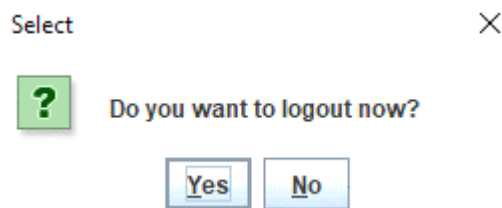
Student 7

btStudentLogout được thiết kế như sau:

```
private void btStudentLogoutActionPerformed(java.awt.event.ActionEvent evt) {  
    int a = JOptionPane.showConfirmDialog(parentComponent: this, message: "Do you want to logout now?",  
        title: "Select", optionType: JOptionPane.YES_NO_OPTION);  
    if (a == 0) {  
        this.dispose();  
    }  
}
```

Student 8

Khi người dùng chọn logout sẽ hiện thông báo để chắc chắn rằng lựa chọn



Student 9

nếu lựa chọn là Yes thì sẽ thoát khỏi giao diện ngược lại người dùng có thể chọn “No” hoặc “x” để không thoát khỏi.

Phần này là phần giao diện để người dùng quyết định lưu thông tin hay sửa đổi, bổ sung, tìm kiếm... thông qua các button.

| | | | |
|-----------------------|--|---------------|----------------|
| Search Student | <input style="width: 95%;" type="text"/> | Search | Refresh |
| | | | |
| Add | Update | Delete | Print |
| Clear | | Logout | |

Student 10

Trong bảng “tbStudent” tôi sẽ cài đặt Event -> Mouse -> mouseClicked. Mục đích làm như vậy là khi chọn vào một sinh viên nào đó thì thông tin sinh viên sẽ được hiện ra bên **plStudentInforamtionL** để cho ta có thể sửa đổi, bổ sung...

btStudentDelete người dùng lựa chọn sinh viên cần xoá trên bảng thông tin, sinh viên này xoá đi thông qua id sinh viên đó với lệnh truy vấn với sql

```

public void delete(int id) {
    int yesOrNo = JOptionPane.showConfirmDialog(parentComponent: null,
        message: "Course and score records will also be deleted",
        title: "Student Dlete", optionType:JOptionPane.OK_CANCEL_OPTION);
    if (yesOrNo == JOptionPane.OK_OPTION) {
        try {
            ps = con.prepareStatement(sql:"delete from student where id = ?");
            ps.setInt(parameterIndex:1, x: id);
            if (ps.executeUpdate() > 0) {
                JOptionPane.showMessageDialog(parentComponent: null,
                    message: "Student deleted sucessfully");
            }
        } catch (SQLException ex) {
            Logger.getLogger(name: Student.class.getName()).log(level: Level.SEVERE, msg:null, thrown: ex);
        }
    }
}

```

Student 11

btStudentPrint tất cả các thông tin sinh viên sẽ được in ra dưới file như :pdf,..
cú pháp để xây dựng button này

```
private void btStudentPrintActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        MessageFormat header = new MessageFormat(pattern: "Students Information");
        MessageFormat footer = new MessageFormat(pattern: "Page{0, number, integer}");
        tbStudent.print(printMode: JTable.PrintMode.FIT_WIDTH, headerFormat: header, footerFormat: footer);
    } catch (PrinterException ex) {
        Logger.getLogger(name: Home.class.getName()).log(level: Level.SEVERE, msg: null, thrown: ex);
    }
}
```

Student 12

btStudentAdd khi lựa chọn button Add thông tin sinh viên nhập đầy đủ sẽ được lưu trữ lên database và sau đó thông tin được hiện sinh viên lưu thành công sẽ được hiện ra trên bảng, có thông báo lưu thành công thông tin sinh viên. Khi thông tin bị rỗng hoặc điền không đúng sẽ không thêm được thông tin.

```
private void btStudentAddActionPerformed(java.awt.event.ActionEvent evt) {
    if (isEmptyStudent()) {
        if (!student.isEmailExist(email: tfStudentEmail.getText())) {
            if (!student.isPhoneExist(phone: tfStudentPhone.getText())) {
                int id = student.getMax();
                String name = tfStudentName.getText();
                SimpleDateFormat dateFormat = new SimpleDateFormat(pattern: "yyyy-MM-dd");
                String date = dateFormat.format(date: dcStudentDate.getDate());
                String gender = cbStudentGender.getSelectedItem().toString();
                String email = tfStudentEmail.getText();
                String phone = tfStudentPhone.getText();
                String father = tfStudentFather.getText();
                String mother = tfStudentMother.getText();
                String address1 = tfStudentAddress1.getText();
                String address2 = tfStudentAddress2.getText();
```

Student 13

```
                student.insert(id, sname: name, date, gender, email, phone, father,
                    mother, address1, address2, imagePath);
                tbStudent.setModel(new DefaultTableModel(data: null, new Object[]{"Student ID", "Student Name",
                    "Date of Birth", "Gender", "Email", "Phone Number",
                    "Father Name", "Mother Name", "Address Line 1", "Address Line 2", "Image Path"}));
                student.getStudentValue(table: tbStudent, searchValue: "");
                clearStudent();
            } else {
                JOptionPane.showMessageDialog(parentComponent: this, message: "This phone already exists");
            }
        } else {
            JOptionPane.showMessageDialog(parentComponent: this, message: "This email already exists");
        }
    }
}
```

Student 14

btStudentUpdate thông tin sinh viên bị sai người dùng kích chọn vào sinh viên sai thông tin đó, sau khi chọn thông tin sinh viên bị sai trên bảng dữ liệu thông tin sinh viên này được hiện qua bên **plStudentInforamtionT** như hình sau

[illegible]

Student 15

ta có thể chỉnh sửa lại thông tin và lựa chọn Update để cập nhật thông tin.

btStudentSearch khi nhập đúng ID thì thông tin của ID đó được hiện ra trên bảng dữ liệu

```

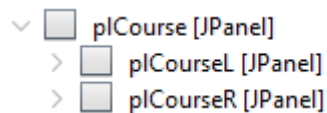
public void searchStudent(JTable table, String searchValue) {
    String sql = "select * from student where id = ?";
    try {
        ps = con.prepareStatement(sql);
        ps.setInt(parameterIndex:1, x: Integer.parseInt(s: searchValue));
        ResultSet rs = ps.executeQuery();
        DefaultTableModel model = (DefaultTableModel) table.getModel();
        Object[] row;
        while (rs.next()) {
            row = new Object[11];
            row[0] = rs.getInt(columnIndex:1);
            row[1] = rs.getString(columnIndex:2);
            row[2] = rs.getString(columnIndex:3);
            row[3] = rs.getString(columnIndex:4);
            row[4] = rs.getString(columnIndex:5);
            row[5] = rs.getString(columnIndex:6);
            row[6] = rs.getString(columnIndex:7);
            row[7] = rs.getString(columnIndex:8);
            row[8] = rs.getString(columnIndex:9);
            row[9] = rs.getString(columnIndex:10);
            row[10] = rs.getString(columnIndex:11);
            model.addRow(rowData: row);
        }
    } catch (SQLException ex) {
        Logger.getLogger(name: Student.class.getName()).log(level:Level.SEVERE, msg: null, thrown: ex);
    }
}

```

Student 16

2.2.2.2.Course

plCourse JPanel được chia làm hai JPanel đó là **plCourseL** và **plCourseR**.



Course 2


Phần **plCourseL** là phần điền các thông khoá học của sinh viên được thiết kế như sau:

```

>  plCourseSearchL [JPanel]
  label jLabel14 [JLabel]
 tfCourseID [JTextField]
 tfCourseStudentID [JTextField]
  label jLabel16 [JLabel]
 cbCourseSemester [JComboBox]
 cbCourseCourse1 [JComboBox]
 cbCourseCourse2 [JComboBox]
 cbCourseCourse3 [JComboBox]
 cbCourseCourse4 [JComboBox]
  label jLabel17 [JLabel]
  label jLabel18 [JLabel]
  label jLabel19 [JLabel]
  label jLabel20 [JLabel]
  label jLabel21 [JLabel]
 cbCourseCourse5 [JComboBox]
  label jLabel22 [JLabel]

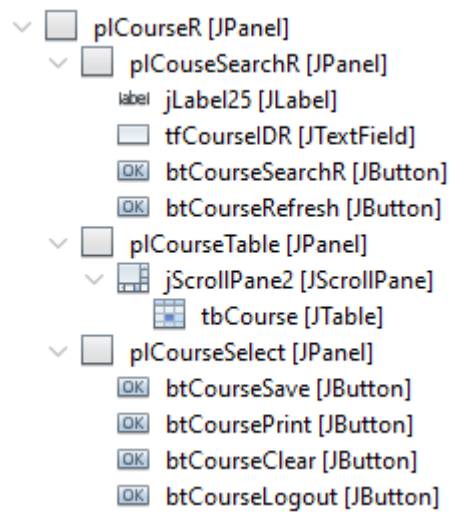
```

Course 3

| | |
|---|--|
| Student's ID <input type="text"/>  Search | |
| ID | <input type="text"/> |
| Student's ID | <input type="text"/> |
| Semester | <input type="text"/> |
| Course 1 | <input type="text" value="C"/> |
| Course 2 | <input type="text" value="Computer NetWorking"/> |
| Course 3 | <input type="text" value="Database"/> |
| Course 4 | <input type="text" value="Java OOP"/> |
| Course 5 | <input type="text" value="Machine Learning"/> |

Course 4

plCourseR bên đây sẽ là phần hiển thị thông tin khoá học được đăng ký từ sinh viên và hiện thông tin đó ra bảng



Course 5

| |
|--|
| <div style="display: flex; justify-content: space-between; align-items: center;"> Search Student <div style="flex-grow: 1; border: 1px solid #ccc; height: 20px;"></div> <div style="display: flex; gap: 10px;"> <div style="display: flex; align-items: center; gap: 5px;"> Search </div> <div style="display: flex; align-items: center; gap: 5px;"> Refresh </div> </div> </div> |
| |
| <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="display: flex; align-items: center; gap: 10px;"> <div style="display: flex; align-items: center; gap: 5px;"> Save </div> <div style="display: flex; align-items: center; gap: 5px;"> Print </div> <div style="display: flex; align-items: center; gap: 5px;"> Clear </div> <div style="display: flex; align-items: center; gap: 5px;"> Logout </div> </div> </div> |

Course 6

2.2.2.3.Score

plScore vẫn gồm hai phần chính là plScoreL và plScoreR phần bên trái vẫn là nơi để nhập thông tin và phần bên phải là hiện thông tin đó ra


plScoreL thiết kế như sau

```

    - plScoreL [JPanel]
      - jLabel23 [JLabel]
      - tfScoreID [JTextField]
      - jLabel24 [JLabel]
      - jLabel50 [JLabel]
      - jLabel51 [JLabel]
      - jLabel52 [JLabel]
      - jLabel53 [JLabel]
      - tfScoreStudentID [JTextField]
      - jLabel54 [JLabel]
      - jLabel55 [JLabel]
      - plScoreSearchL [JPanel]
        - jLabel56 [JLabel]
        - tfScoreStudentIDL [JTextField]
        - tfScoreSemesterSearch [JTextField]
        - jLabel57 [JLabel]
        - jPanel29 [JPanel]
          - btScoreSearchL [JButton]
          - tfScoreSemester [JTextField]
          - tfScoreCourse1 [JTextField]
          - tfScoreCourse2 [JTextField]
          - tfScoreCourse3 [JTextField]
          - tfScoreCourse4 [JTextField]
          - tfScoreCourse5 [JTextField]
          - tfScoreScore1 [JTextField]
          - tfScoreScore2 [JTextField]
          - tfScoreScore3 [JTextField]
          - tfScoreScore4 [JTextField]
          - tfScoreScore5 [JTextField]

```

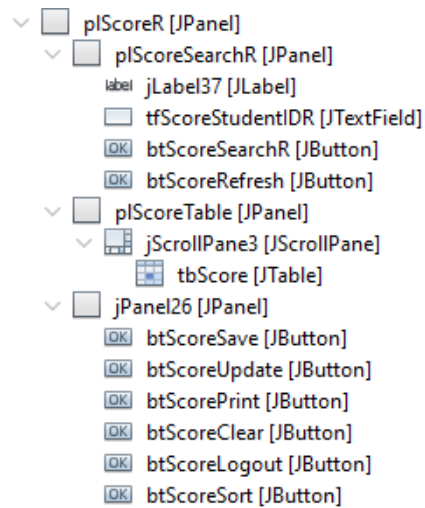
Score 2

| | |
|----------------------|--|
| Student's ID | |
| <input type="text"/> |  Search |
| Semester | |
| <input type="text"/> | |

| | | |
|---------------------|----------------------|------------|
| ID | <input type="text"/> | |
| Student's ID | <input type="text"/> | |
| Semester | <input type="text"/> | |
| Course 1 | <input type="text"/> | 0.0 |
| Course 2 | <input type="text"/> | 0.0 |
| Course 3 | <input type="text"/> | 0.0 |
| Course 4 | <input type="text"/> | 0.0 |
| Course 5 | <input type="text"/> | 0.0 |

Score 3

plScoreR thiết kế như sau



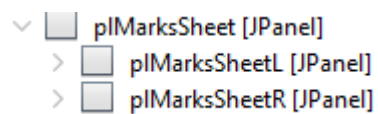
Score 4

| |
|---|
| <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">Search Student</div> <input style="flex-grow: 1;" type="text"/> <div style="margin-left: 10px;"> <div style="display: flex; gap: 10px;"> <div> Search</div> <div> Refresh</div> </div> </div> </div> |
| |
| <div style="display: flex; justify-content: space-between; gap: 10px;"> <div> Save</div> <div> Update</div> <div> Print</div> <div>Sort</div> <div> Clear</div> <div> Logout</div> </div> |

Score 5

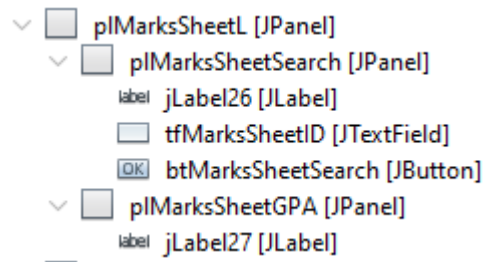
2.2.2.4.Marks Sheet

Ta chia làm hai phần là plMraksSheetL ,plMraksSheetL



Marks Sheet 2

plMraksSheetL



Marks Sheet 3

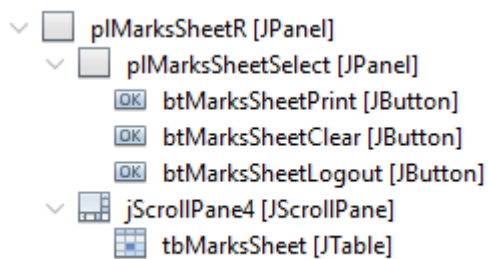
Student's ID

GPA: 0.0

Marks Sheet 4

plMraksSheetR

Phần bên phải này khi người dùng nhập tìm kiếm ID thì thông tin về điểm khoá học của sinh viên đó được hiện ra ở đây.



Marks Sheet 5

| |
|--|
| |
| <div><div> Print</div><div> Clear</div><div> Logout</div></div> |

Marks Sheet 6

Đối với các button ở các bảng khác cũng như các button ở phần thiết kế của Studet lên tôi sẽ không viết lại nữa. Ý tưởng các button tương tự, chỉ việc sao chép lại và sử dụng lại tên cho các button của bảng là chúng ta có thể sử dụng lại chúng.

III. KẾT LUẬN

1. Ưu điểm

Phần mềm quản lý sinh viên cơ bản đã hoàn thành được những công việc sau:

- Cấu trúc tương đối đầy đủ và gần gũi với thực tế giúp việc nắm bắt một cách dễ dàng
- Cập nhập, thêm, sửa, xoá, sắp xếp thông tin

2. Nhược điểm

- Phần mềm còn nhiều hạn chế chưa đáp ứng được đầy đủ một mô hình quản sinh viên
- Trong quá trình xây dựng còn nhiều hạn chế chưa đáp ứng hết cho người dùng.

3. Hướng phát triển

- Bổ sung thêm nhiều chức năng nữa cho phù với công việc quản lý sinh viên như thời khoá biểu, lịch học, lịch thi...
- Tối ưu chương trình để có thể sử dụng Online tiếp cận hệ thống giảng viên, sinh viên.