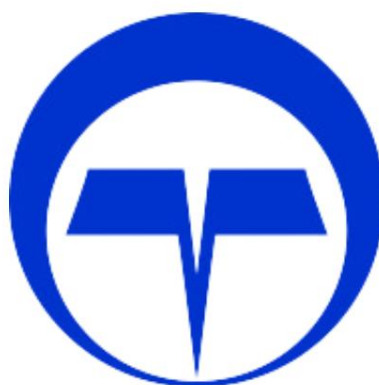


**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**VIỆN KHOA HỌC KỸ THUẬT BƯU ĐIỆN**

-----



**MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**  
**DỰ ÁN: HỆ THỐNG QUẢN LÝ SINH VIÊN**

**Giảng viên: Nguyễn Mạnh Dũng**

**Sinh viên: Nguyễn Đức Thành**

**Lương Toàn Thắng**

**Thiều Minh Duy**

**Hà Nội, tháng 12/2024**

# MỤC LỤC

<b>I. Giới thiệu</b>	<b>3</b>
1. Mục tiêu dự án	3
2. Phạm vi dự án	3
<b>II. Kiến trúc hệ thống</b>	<b>4</b>
1. Cấu trúc tổng thể	4
2. Công nghệ sử dụng	5
3. Quy trình hoạt động	5
4. Bảo mật và phân quyền	6
<b>III. Giao diện người dùng</b>	<b>6</b>
1. Giao diện đăng nhập	6
2. Giao diện quản lý sinh viên (Quản trị viên)	7
3. Giao diện quản lý khóa học (Quản trị viên)	8
4. Giao diện báo cáo (Quản trị viên)	9
5. Giao diện điểm số sinh viên	9
6. Giao diện cho sinh viên	10
<b>IV. Chức năng của hệ thống</b>	<b>11</b>
1. Chức năng của quản trị viên	11
1.1. Quản lý thông tin sinh viên:	11
1.2. Quản lý khóa học	11
1.3. Quản lý điểm số	12
1.4. Báo cáo và phân tích	12
2. Chức năng của sinh viên	12
2.1. Đăng nhập	12
2.2. Xem thông tin cá nhân	12
2.3. Xem các khóa học đã đăng ký	12
2.4. Xem điểm số	12
3. Tính năng bảo mật	12

4. Khả năng mở rộng.....	13
<b>V. Mô tả mã nguồn .....</b>	<b>13</b>
1. Tổng quan về mã nguồn.....	13
2. Các lớp chính .....	13
2.1. Lớp <i>database</i> .....	13
2.2. Lớp <i>studentView</i> .....	14
2.3. Lớp <i>StudentGradeViewController</i> .....	16
<b>VI. Kết luận .....</b>	<b>18</b>
1. Tính năng nổi bật.....	18
2. Khả năng mở rộng.....	18
3. Tương lai của hệ thống.....	19
4. Kết luận chung.....	19

## I. Giới thiệu

### 1. Mục tiêu dự án

Mục tiêu của dự án Hệ thống Quản lý Sinh viên là xây dựng một nền tảng hiệu quả để quản lý thông tin sinh viên và khóa học. Dưới đây là các mục tiêu cụ thể:

- **Quản lý thông tin sinh viên:** Cung cấp khả năng thêm, sửa, đổi, xóa thông tin sinh viên.

- **Quản lý khóa học:** Cho phép quản trị viên quản lý danh sách khóa học, bao gồm việc thêm mới, sửa đổi và xóa thông tin khóa học.

- **Giao diện người dùng thân thiện:** Phát triển giao diện trực quan, dễ sử dụng cho cả quản trị viên và sinh viên.

- **Bảo mật thông tin:** Đảm bảo thông tin cá nhân được bảo vệ thông qua việc yêu cầu đăng nhập và phân quyền truy cập.

- **Tích hợp cơ sở dữ liệu:** Sử dụng MySQL để lưu trữ và quản lý dữ liệu, cho phép truy xuất thông tin nhanh chóng và hiệu quả.

Tóm lại, dự án nhằm tạo ra một công cụ mạnh mẽ giúp quản lý thông tin sinh viên và khóa học một cách hiệu quả, bảo mật và linh hoạt.

### 2. Phạm vi dự án

Phạm vi của dự án Hệ thống Quản lý Sinh viên được xác định để đảm bảo rằng các chức năng và yêu cầu chính được thực hiện một cách hiệu quả. Dưới đây là các khía cạnh cụ thể của phạm vi dự án:

- **Đối tượng người dùng:** Dự án sẽ phục vụ hai nhóm người dùng chính: quản trị viên và sinh viên. Quản trị viên sẽ có quyền truy cập đầy đủ để quản lý thông tin, trong khi sinh viên sẽ chỉ có quyền truy cập vào thông tin cá nhân và các khóa học mà họ tham gia.

- **Chức năng quản lý sinh viên:** Hệ thống sẽ cho phép quản trị viên thực hiện các thao tác như thêm mới, chỉnh sửa và xóa thông tin sinh viên. Điều này bao gồm việc theo dõi quá trình học tập và các hoạt động ngoại khóa của sinh viên.

- **Chức năng quản lý khóa học:** Dự án sẽ cung cấp khả năng cho quản trị viên để tạo và quản lý danh sách các khóa học. Mỗi khóa học sẽ có thông tin chi tiết như thời gian, địa điểm, và giảng viên phụ trách.

- **Giao diện người dùng:** Hệ thống sẽ được thiết kế với giao diện người dùng thân thiện, dễ sử dụng, giúp người dùng dễ dàng tìm kiếm thông tin và thực hiện các thao tác mà không gặp khó khăn.

- **Báo cáo và thống kê:** Dự án sẽ bao gồm các chức năng báo cáo để cung cấp thông tin về tình trạng học tập của sinh viên và hiệu suất của các khóa học. Điều này sẽ giúp quản trị viên có cái nhìn tổng quan và đưa ra quyết định hợp lý.

- **Tích hợp cơ sở dữ liệu:** Hệ thống sẽ sử dụng MySQL để lưu trữ dữ liệu, đảm bảo rằng thông tin được quản lý một cách an toàn và hiệu quả. Dữ liệu sẽ được truy xuất nhanh chóng để phục vụ cho các yêu cầu của người dùng.

Tóm lại, phạm vi dự án Hệ thống Quản lý Sinh viên tập trung vào việc phát triển một ứng dụng quản lý thông tin sinh viên và khóa học, với các chức năng thiết yếu và giao diện thân thiện, nhằm đáp ứng nhu cầu của người dùng một cách hiệu quả và thuận tiện.

## II. Kiến trúc hệ thống

Kiến trúc của Hệ thống Quản lý Sinh viên được thiết kế để đảm bảo tính hiệu quả, khả năng mở rộng và dễ bảo trì. Hệ thống được xây dựng dựa trên mô hình MVC (Model-View-Controller), giúp tách biệt các thành phần của ứng dụng và cải thiện khả năng quản lý mã nguồn. Dưới đây là các thành phần chính của kiến trúc hệ thống:

### 1. Cấu trúc tổng thể

- Model (Mô hình): Đây là phần xử lý dữ liệu và logic nghiệp vụ của ứng dụng. Mô hình sẽ tương tác với cơ sở dữ liệu để thực hiện các thao tác như thêm, sửa, xóa và truy xuất thông tin sinh viên và khóa học. Các lớp như *database*, *courseData* và *studentData* sẽ nằm trong phần này, đảm bảo rằng tất cả các thao tác liên quan đến dữ liệu đều được quản lý một cách tập trung:

- Database: Lớp này chịu trách nhiệm kết nối với cơ sở dữ liệu MySQL, thực hiện các truy vấn và quản lý kết nối.
- StudentData: Lớp này chứa các phương thức để quản lý thông tin sinh viên, bao gồm thêm, sửa, xóa và truy xuất thông tin sinh viên.
- CourseData: Lớp này chứa các phương thức để quản lý các khóa học của sinh viên.

- View (Giao diện): Phần giao diện người dùng được xây dựng bằng JavaFX, sử dụng các tệp FXML để định nghĩa bố cục, giao diện sẽ bao gồm các màn hình

cho việc đăng nhập, quản lý sinh viên, quản lý khóa học và xem báo cáo. Giao diện sẽ được thiết kế để thân thiện và dễ sử dụng, giúp người dùng dễ dàng tương tác với hệ thống. Một số tệp FXML chính:

- Hello-view.fxml: Giao diện đăng nhập
- StudentGradeView.fxml: Giao diện điểm số sinh viên
- StudentView.fxml: Giao diện của sinh viên

- Controller (Điều khiển): Đây là phần xử lý các sự kiện từ người dùng và tương tác với mô hình để cập nhật giao diện. Các lớp như *HelloController* và *dashboardController* sẽ đảm nhiệm vai trò này, nhận các sự kiện từ giao diện và thực hiện các thao tác cần thiết trên mô hình.

## 2. Công nghệ sử dụng

- Ngôn ngữ lập trình: Dự án được phát triển bằng ngôn ngữ Java, một ngôn ngữ lập trình mạnh mẽ và phổ biến, phù hợp cho việc xây dựng ứng dụng desktop.

- JavaFX: Được sử dụng để xây dựng giao diện người dùng, JavaFX cung cấp các thành phần giao diện phong phú và khả năng tạo ra các ứng dụng đồ họa đẹp mắt.

- Cơ sở dữ liệu: MySQL được sử dụng để lưu trữ dữ liệu. Hệ thống sẽ thực hiện các truy vấn SQL để quản lý thông tin sinh viên và khóa học.

- Maven: Công cụ quản lý dự án và phụ thuộc, giúp dễ dàng quản lý các thư viện và phiên bản cần thiết cho dự án.

## 3. Quy trình hoạt động

- Đăng nhập: Người dùng (quản trị viên hoặc sinh viên) sẽ nhập thông tin đăng nhập vào giao diện. Hệ thống sẽ kiểm tra thông tin này thông qua lớp điều khiển và xác thực với cơ sở dữ liệu.

- Quản lý thông tin: Sau khi đăng nhập thành công, quản trị viên có thể truy cập các chức năng quản lý sinh viên và khóa học. Hệ thống sẽ sử dụng các phương thức trong mô hình để thực hiện các thao tác như thêm, sửa, xóa và truy xuất thông tin.

- Báo cáo và phân tích: Quản trị viên có thể yêu cầu các báo cáo về tình trạng học tập của sinh viên và hiệu suất của các khóa học. Hệ thống sẽ truy xuất dữ liệu từ cơ sở dữ liệu và hiển thị thông tin trên giao diện.

- Đăng xuất: Người dùng có thể đăng xuất khỏi hệ thống, và giao diện sẽ trở về màn hình đăng nhập.

#### **4. Bảo mật và phân quyền**

Hệ thống sẽ áp dụng các biện pháp bảo mật để bảo vệ thông tin người dùng. Điều này bao gồm việc mã hóa mật khẩu, sử dụng kết nối an toàn đến cơ sở dữ liệu và phân quyền truy cập cho các loại người dùng khác nhau. Quản trị viên sẽ có quyền truy cập đầy đủ, trong khi sinh viên chỉ có thể xem thông tin cá nhân và các khóa học mà họ đã đăng ký.

### **III. Giao diện người dùng**

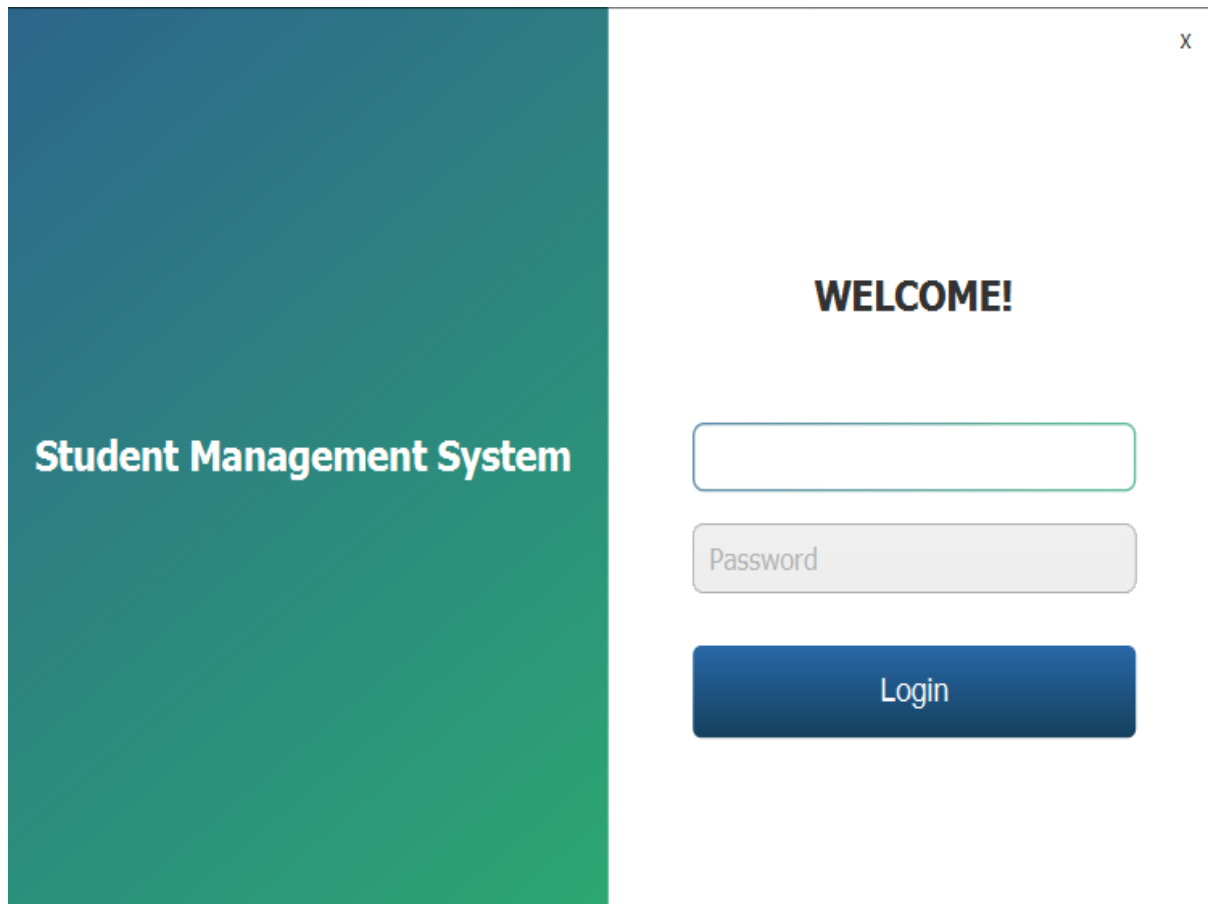
Giao diện người dùng của Hệ thống Quản lý Sinh viên được thiết kế để phục vụ cho cả quản trị viên và sinh viên, đảm bảo rằng mỗi nhóm người dùng có thể dễ dàng truy cập và sử dụng các chức năng cần thiết. Dưới đây là mô tả chi tiết về giao diện cho cả hai loại người dùng, bao gồm các thành phần chính và hình ảnh minh họa.

#### **1. Giao diện đăng nhập**

Mô tả: Màn hình đăng nhập là điểm khởi đầu cho cả quản trị viên và sinh viên. Giao diện này bao gồm các trường nhập liệu cho tên người dùng và mật khẩu, cùng với nút đăng nhập.

Thành phần: trường nhập liệu cho tên người dùng; trường nhập liệu cho mật khẩu; nút đăng nhập; thông báo lỗi (nếu có).

Giao diện đăng nhập:



Đoạn mã chính:

```
<AnchorPane>
    <TextField fx:id="username" promptText="Tên người dùng" />
    <PasswordField fx:id="password" promptText="Mật khẩu" />
    <Button fx:id="loginBtn" text="Đăng nhập" onAction="#loginAdmin" />
    <Label fx:id="errorMessage" textFill="RED" />
</AnchorPane>
```

## 2. Giao diện quản lý sinh viên (Quản trị viên)

Mô tả: Sau khi đăng nhập, quản trị viên sẽ được chuyển đến giao diện quản lý sinh viên. Giao diện này cho phép quản trị viên xem, thêm, sửa và xóa thông tin sinh viên.

Thành phần: Bảng danh sách sinh viên; nút thêm, sửa, xóa sinh viên; khu vực hiển thị thông tin chi tiết sinh viên.

Giao diện quản lý sinh viên:



Student Management System

Welcome,  
**Admin**

Home

Add Students

Available Courses

Grades of Students

Sign out

Search

Student	Year	Course	First Name	Last Name	Gender	Birth Date	Status
2	Second year	INT1332	Pham	Nam	Male	2005-03-16	Enrolled
3	Fourth year	BAS1150	Pham	Linh	Female	2003-06-26	Enrolled
4	Third year	SKD1103	Nguyen	Tuan	Male	2004-06-16	Enrolled
5	First year	BAS1150	Nguyen	Yen	Female	2006-08-17	Enrolled
6	Second year	INT1332	Nguyen	An	Female	2006-10-12	Enrolled
7	First year	BAS1150	Vu	Nhung	Female	2006-12-12	Enrolled

Student: 6

Year:

Course:

First Name: Nguyen

Last Name: An

Gender:

Birth Date: 10/12/2006

Status:

Clear

Delete

Update

Add

### 3. Giao diện quản lý khóa học (Quản trị viên)

Mô tả: Giao diện này cho phép quản trị viên quản lý danh sách các khóa học, bao gồm việc thêm mới, sửa đổi và xóa thông tin khóa học.

Thành phần: bảng danh sách khóa học; nút thêm, sửa, xóa khóa học; khu vực hiện thị thông tin chi tiết của khóa học.

Giao diện quản lý khóa học:

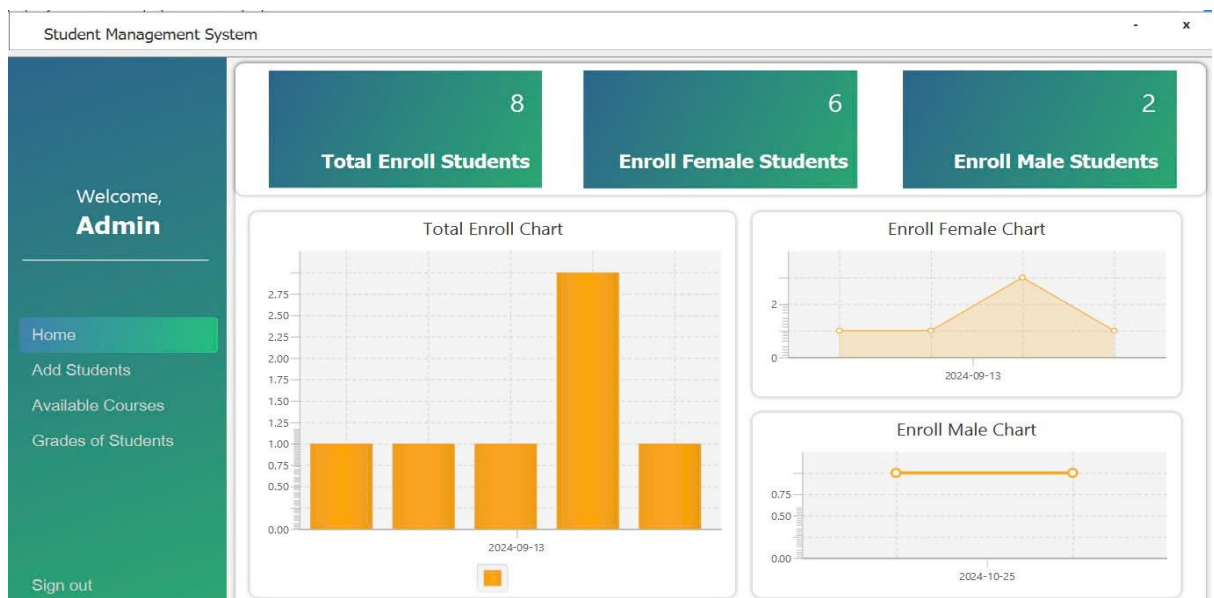
[illegible]

#### 4. Giao diện báo cáo (Quản trị viên)

Mô tả: Giao diện báo cáo cung cấp cho quản trị viên cái nhìn tổng quan về tình trạng học tập của sinh viên và hiệu suất của các khóa học.

Thành phần: biểu đồ và đồ thị hiển thị dữ liệu; bảng thống kê về số lượng sinh viên và tỷ lệ hoàn thành khóa học

Giao diện báo cáo:



## 5. Giao diện điểm số sinh viên

Mô tả: Quản trị viên có thể thêm, sửa, xóa điểm số của mỗi sinh viên, đồng thời hệ thống cũng có thể cung cấp các báo cáo tổng hợp về điểm số của sinh

viên, giúp quản trị viên theo dõi hiệu suất học tập và đưa ra các quyết định hợp lý.

Giao diện điểm số sinh viên:


[illegible]

## 6. Giao diện cho sinh viên

Mô tả: Sinh viên sẽ có một giao diện riêng để xem thông tin cá nhân, các khóa học đã đăng ký và điểm số. Giao diện này giúp sinh viên dễ dàng theo dõi quá trình học tập của mình.

Thành phần: thông tin cá nhân của sinh viên; danh sách các khóa học đã đăng ký; điểm số và tình trạng học tập.

Giao diên cho sinh viên:

Student Grade View					Logout	X
<div>  <div>Student: Bui Yen (ID: 1)</div> </div>						
Year	Course	First Sem	Second Sem	Final		
Second year	INT1358	7.0	9.0	8.0		

#### IV. Chức năng của hệ thống

Hệ thống Quản lý Sinh viên được thiết kế với nhiều chức năng nhằm đáp ứng nhu cầu quản lý thông tin sinh viên, khóa học và điểm số một cách hiệu quả. Dưới đây là các chức năng chính của hệ thống, được chia thành hai nhóm người dùng: quản trị viên và sinh viên.

##### 1. Chức năng của quản trị viên

Quản trị viên là người có quyền truy cập đầy đủ vào hệ thống và có thể thực hiện các chức năng sau:

###### 1.1. Quản lý thông tin sinh viên:

- Thêm sinh viên: Quản trị viên có thể thêm thông tin mới cho sinh viên, bao gồm họ tên, ngày sinh,...
- Sửa thông tin sinh viên: Quản trị viên có thể chỉnh sửa thông tin của sinh viên đã có trong hệ thống.
- Xóa sinh viên: Quản trị viên có thể xóa thông tin của sinh viên không còn cần thiết.

###### 1.2. Quản lý khóa học

- Thêm khóa học: Quản trị viên có thể thêm thông tin về các khóa học mới, bao gồm tên khóa học, mô tả, số tín chỉ.
- Sửa thông tin khóa học: Quản trị viên có thể chỉnh sửa thông tin của các khóa học đã có.

- Xóa khóa học: Quản trị viên có thể xóa các khóa học không còn hoạt động hoặc không cần thiết

### 1.3. Quản lý điểm số

- Thêm điểm số: Quản trị viên có thể nhập thêm điểm số cho từng sinh viên trong các khóa học.

- Sửa điểm số: Quản trị viên có thể chỉnh sửa điểm số của sinh viên nếu có sự thay đổi.

- Xóa điểm số: Quản trị viên có thể xóa điểm số không còn cần thiết nếu có sự sai sót.

### 1.4. Báo cáo và phân tích

- Tạo báo cáo: Quản trị viên có thể tạo các báo cáo tổng hợp về tình trạng học tập của sinh viên, hiệu suất của các khóa học và điểm số.

- Phân tích dữ liệu: Hệ thống cung cấp các công cụ để phân tích dữ liệu, giúp quản trị viên đưa ra quyết định hợp lý.

## 2. Chức năng của sinh viên

Sinh viên có quyền truy cập vào các chức năng cơ bản của hệ thống, bao gồm:

### 2.1. Đăng nhập

Sinh viên có thể đăng nhập vào hệ thống bằng tài khoản và mật khẩu của mình.

### 2.2. Xem thông tin cá nhân

Sinh viên có thể xem thông tin cá nhân của mình, bao gồm họ tên, mã sinh viên,...

### 2.3. Xem các khóa học đã đăng ký

Sinh viên có thể xem danh sách các khóa học mà họ đã đăng ký, bao gồm tên khóa học, mô tả và số tín chỉ.

### 2.4. Xem điểm số

Sinh viên có thể xem điểm số của mình trong các khóa học, giúp họ theo dõi tiến độ học tập và hiệu suất của bản thân.

## 3. Tính năng bảo mật

Hệ thống sẽ áp dụng các biện pháp bảo mật để bảo vệ thông tin người dùng, bao gồm:

- Mã hóa mật khẩu: Mật khẩu của người dùng sẽ được mã hóa để đảm bảo an toàn.

- Phân quyền truy cập: Chỉ những người dùng có quyền mới có thể truy cập vào các chức năng nhạy cảm của hệ thống.

#### **4. Khả năng mở rộng**

Hệ thống được thiết kế với khả năng mở rộng, cho phép thêm các chức năng mới trong tương lai mà không cần thay đổi cấu trúc hiện tại. Điều này giúp hệ thống có thể thích ứng với các yêu cầu mới và cải tiến theo thời gian.

### **V. Mô tả mã nguồn**

Mã nguồn của hệ thống được chia thành các lớp và gói khác nhau, mỗi lớp đảm nhận một chức năng cụ thể trong ứng dụng. Hệ thống sử dụng JavaFX cho giao diện người dùng và JDBC để kết nối với cơ sở dữ liệu MySQL. Các lớp chính bao gồm:

#### **1. Tổng quan về mã nguồn**

Mã nguồn của hệ thống được chia thành các lớp và gói khác nhau, mỗi lớp đảm nhận một chức năng cụ thể trong ứng dụng. Hệ thống sử dụng JavaFX cho giao diện người dùng và JDBC để kết nối với cơ sở dữ liệu MySQL. Các lớp chính bao gồm:

- Lớp kết nối cơ sở dữ liệu: Chịu trách nhiệm thiết lập và quản lý kết nối với cơ sở dữ liệu.

- Lớp điều khiển: Xử lý logic nghiệp vụ và tương tác giữa giao diện người dùng và mô hình dữ liệu.

- Lớp giao diện người dùng: Định nghĩa các thành phần giao diện và cách thức người dùng tương tác với hệ thống.

#### **2. Các lớp chính**

##### *2.1. Lớp database*

Lớp này quản lý kết nối đến cơ sở dữ liệu MySQL. Nó cung cấp phương thức *connectDb()* để thiết lập kết nối và trả về đối tượng *Connection*.

```

package com.example.project_end_term;

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

public class database {
    private static final String URL = "jdbc:mysql://localhost:3306/studentdata";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public static Connection connectDb() {
        Connection connection = null;
        try {
            connection = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("Kết nối thành công!");
        } catch (SQLException e) {
            System.out.println("Kết nối thất bại!");
            e.printStackTrace();
        }
        return connection;
    }
}

```

## 2.2. Lớp *studentView*

Lớp `StudentView` là một phần quan trọng trong Hệ thống Quản lý Sinh viên, chịu trách nhiệm quản lý giao diện đăng nhập cho sinh viên. Lớp này không chỉ xử lý việc hiển thị giao diện mà còn thực hiện các thao tác xác thực thông tin người dùng, đảm bảo rằng chỉ những sinh viên đã đăng ký và có trạng thái "Enrolled" mới có thể truy cập vào hệ thống.

### \* Các thành phần giao diện

Lớp *studentView* sử dụng JavaFX để xây dựng giao diện người dùng. Các thành phần chính bao gồm:

- TextField *studentId*: Trường nhập liệu cho sinh viên nhập mã sinh viên của họ. Đây là thông tin cần thiết để xác thực danh tính của sinh viên.
- PasswordField *password*: Trường nhập liệu cho sinh viên nhập mật khẩu. Mật khẩu sẽ được mã hóa và kiểm tra để đảm bảo tính bảo mật.
- Button *loginBtn*: Nút đăng nhập, khi được nhấn sẽ kích hoạt phương thức `loginStudent()` để thực hiện xác thực.

```

@FXML
private TextField studentId;

@FXML
private PasswordField password;

@FXML
private Button loginBtn;

```

#### \* Phương thức *loginStudent()*

Phương thức *loginStudent()* là phần quan trọng nhất của lớp này. Nó thực hiện các bước sau:

- Kết nối đến cơ sở dữ liệu: Sử dụng lớp *database* để thiết lập kết nối với cơ sở dữ liệu MySQL.

- Xác thực thông tin: Thực hiện truy vấn SQL để kiểm tra xem mã sinh viên và trạng thái của họ có hợp lệ hay không. Nếu sinh viên không tồn tại hoặc không có trạng thái "Enrolled", hệ thống sẽ hiển thị thông báo lỗi.

- Chuyển hướng: Nếu thông tin đăng nhập hợp lệ, lớp sẽ tải giao diện điểm số của sinh viên (*studentGradeView.fxml*), cho phép sinh viên xem điểm số và thông tin học tập của họ.

```

public void loginStudent() {
    String sql = "SELECT * FROM student WHERE studentId = ? AND status = 'Enrolled'";
    connect = database.connectDb();

    try {
        prepare = connect.prepareStatement(sql);
        prepare.setString(1, studentId.getText());
        result = prepare.executeQuery();

        Alert alert;

        if (studentId.getText().isEmpty() || password.getText().isEmpty()) {
            alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error");
            alert.setContentText("Please fill all blank fields");
            alert.showAndWait();
        } else if (!result.next()) {
            alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error");
            alert.setContentText("Student ID not found or not enrolled");
            alert.showAndWait();
        } else if (!password.getText().equals("1")) {
            alert = new Alert(Alert.AlertType.ERROR);
            alert.setTitle("Error");
            alert.setContentText("Wrong password");
            alert.showAndWait();
        } else {
            // Load student grade view
            FXMLLoader loader = new FXMLLoader(getClass().getResource("studentGradeView.fxml"));
            Parent root = loader.load();
            // Set up the new scene and stage
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```



#### \* Quản lý thông báo

Lớp *StudentView* cũng quản lý việc hiển thị thông báo cho người dùng. Sử dụng lớp *Alert* của JavaFX, lớp này có thể hiển thị các thông báo lỗi khi người dùng nhập thông tin không hợp lệ, giúp cải thiện trải nghiệm người dùng. Điều này không chỉ giúp người dùng nhận biết các lỗi mà còn hướng dẫn họ cách khắc phục.

#### \* Tính năng bảo mật

Lớp này cũng chú trọng đến tính bảo mật. Mật khẩu không được hiển thị trên giao diện và được kiểm tra một cách an toàn. Điều này giúp bảo vệ thông tin nhạy cảm của sinh viên. Hệ thống cũng có thể được mở rộng để mã hóa mật khẩu trước khi lưu trữ trong cơ sở dữ liệu, tăng cường bảo mật cho thông tin người dùng.

#### \* Khả năng mở rộng

Lớp *StudentView* có thể được mở rộng trong tương lai để bao gồm các tính năng như: quên mật khẩu; đăng ký tài khoản; tùy chỉnh giao diện.

### 2.3. Lớp *StudentGradeViewController*

Lớp *StudentGradeViewController* là một phần quan trọng trong Hệ thống Quản lý Sinh viên, chịu trách nhiệm quản lý giao diện hiển thị điểm số của sinh viên. Lớp này không chỉ lấy thông tin điểm số từ cơ sở dữ liệu mà còn hiển thị chúng một cách trực quan trên giao diện người dùng, giúp sinh viên dễ dàng theo dõi hiệu suất học tập của mình.

#### \* Các thành phần giao diện

Lớp *StudentGradeViewController* sử dụng JavaFX để xây dựng giao diện người dùng. Các thành phần chính bao gồm:

- TableView *gradeTable*: Bảng hiển thị điểm số của sinh viên.
- Label *studentInfo*: Hiển thị thông tin cá nhân của sinh viên.
- ImageView *studentImage*: Hiển thị hình ảnh của sinh viên.

```
@FXML
private TableView<studentData> gradeTable;
@FXML
private Label studentInfo;
@FXML
private ImageView studentImage;
```

### \* Phương thức *displayGrade()*

Phương thức *displayGrade()* là phần quan trọng nhất của lớp này. Nó thực hiện các bước sau:

- Kết nối đến cơ sở dữ liệu: Sử dụng lớp *database* để thiết lập kết nối với cơ sở dữ liệu MySQL.

- Lấy thông tin sinh viên: Thực hiện truy vấn SQL để lấy thông tin cá nhân của sinh viên dựa trên *studentId*. Nếu tìm thấy, thông tin sẽ được hiển thị trên giao diện.

- Lấy điểm số: Thực hiện truy vấn SQL khác để lấy điểm số của sinh viên từ bảng *student\_grade*. Dữ liệu này sẽ được chuyển đổi thành danh sách và hiển thị trong *gradeTable*.

```
public void displayGrade() {
    try {
        connect = database.connectDb();
        String studentQuery = "SELECT firstName, lastName, image FROM student WHERE studentId = ?";
        prepare = connect.prepareStatement(studentQuery);
        prepare.setInt(1, studentId);
        result = prepare.executeQuery();

        if (result.next()) {
            studentInfo.setText("Student: " + result.getString("firstName") + " " + result.getString("lastName") + " (ID: " + studentId + ")");
            String imagePath = result.getString("image");
            if (imagePath != null && !imagePath.isEmpty()) {
                Image image = new Image("file:" + imagePath);
                studentImage.setImage(image);
            } else {
                Image defaultImage = new Image(getClass().getResourceAsStream("/com/example/project_end_term/images/default-avatar.png"));
                studentImage.setImage(defaultImage);
            }
        }

        String gradeQuery = "SELECT * FROM student_grade WHERE studentId = ?";
        prepare = connect.prepareStatement(gradeQuery);
        prepare.setInt(1, studentId);
        result = prepare.executeQuery();

        ObservableList<studentData> gradelist = FXCollections.observableArrayList();
        while (result.next()) {
            gradelist.add(new studentData(
                studentId,
                result.getString("year"),
                result.getString("course"),
                result.getDouble("first_sem"),
                result.getDouble("second_sem"),
                result.getDouble("final")
            ));
        }
        gradeTable.setItems(gradelist);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

### \* Quản lý dữ liệu

Lớp *StudentGradeViewController* không chỉ hiển thị dữ liệu mà còn quản lý cách thức dữ liệu được lấy và cập nhật. Bằng cách sử dụng các câu lệnh SQL, lớp

này đảm bảo rằng thông tin hiển thị luôn chính xác và cập nhật theo thời gian thực.

**\* Tính năng bảo mật và hiệu suất**

Lớp này cũng chú trọng đến tính bảo mật và hiệu suất. Việc kiểm tra và xử lý dữ liệu từ cơ sở dữ liệu được thực hiện một cách an toàn, đảm bảo rằng thông tin nhạy cảm của sinh viên không bị lộ ra ngoài. Ngoài ra, việc sử dụng *ObservableList* giúp cập nhật bảng điểm một cách hiệu quả mà không cần phải làm mới toàn bộ giao diện.

## **VI. Kết luận**

Hệ thống Quản lý Sinh viên được thiết kế nhằm đáp ứng nhu cầu quản lý thông tin sinh viên, khóa học và điểm số một cách hiệu quả và thân thiện với người dùng. Qua việc áp dụng mô hình MVC (Model-View-Controller), hệ thống đã tách biệt rõ ràng các thành phần của ứng dụng, giúp cho việc phát triển, bảo trì và mở rộng trở nên dễ dàng hơn.

### **1. Tính năng nổi bật**

Hệ thống cung cấp nhiều tính năng hữu ích cho cả quản trị viên và sinh viên, bao gồm:

- Quản lý thông tin sinh viên: Cho phép quản trị viên thêm, sửa, xóa và truy xuất thông tin sinh viên một cách dễ dàng.
- Quản lý khóa học: Hỗ trợ quản trị viên trong việc quản lý các khóa học, từ việc thêm mới đến việc cập nhật thông tin.
- Quản lý điểm số: Cung cấp cho sinh viên khả năng theo dõi điểm số và hiệu suất học tập của mình thông qua giao diện trực quan.
- Bảo mật thông tin: Hệ thống áp dụng các biện pháp bảo mật để bảo vệ thông tin nhạy cảm của người dùng, đảm bảo rằng chỉ những người có quyền mới có thể truy cập vào các chức năng nhạy cảm.

### **2. Khả năng mở rộng**

Hệ thống được thiết kế với khả năng mở rộng cao, cho phép dễ dàng thêm các chức năng mới trong tương lai mà không cần thay đổi cấu trúc hiện tại. Điều này giúp hệ thống có thể thích ứng với các yêu cầu mới và cải tiến theo thời gian, đáp ứng tốt nhất nhu cầu của người dùng.

### **3. Tương lai của hệ thống**

Trong tương lai, hệ thống có thể được mở rộng để tích hợp thêm các tính năng như quản lý tài chính, hỗ trợ học trực tuyến, và các công cụ phân tích dữ liệu để giúp quản trị viên đưa ra quyết định tốt hơn. Việc lắng nghe phản hồi từ người dùng cũng sẽ là một phần quan trọng trong quá trình phát triển, giúp cải thiện trải nghiệm người dùng và nâng cao hiệu quả của hệ thống.

### **4. Kết luận chung**

Tóm lại, Hệ thống Quản lý Sinh viên không chỉ là một công cụ quản lý thông tin mà còn là một giải pháp toàn diện giúp nâng cao hiệu quả học tập và quản lý trong môi trường giáo dục. Với thiết kế hiện đại, tính năng phong phú và khả năng mở rộng, hệ thống hứa hẹn sẽ đáp ứng tốt nhất nhu cầu của người dùng và góp phần vào sự phát triển bền vững của các cơ sở giáo dục.