**Trường Cao Đẳng Công Nghệ Thông Tin**

**Đại học Đà Nẵng**

**Khoa : Công nghệ thông tin**

**BÁO CÁO PHÂN TÍCH DỰ ÁN**

****

**Môn học :** Dự Án Java

**Giáo viên hướng dẫn :** Trần Minh Thắng

**Sinh viên** : Trần Văn Hoàng Hải

**Lớp** : 18IF1

Đà Nẵng, thứ 4 ngày 19 tháng 6 năm 2019

**Lời nói đầu**

Để hoàn thành bài tập thực hiện dự án phần mềm quản lý, em đã chọn làm phần mềm quản lý trường học với những mục đích như sau :

+ Có thể lưu lại thông tin các sinh viên theo từng lớp, cũng như thông tin các giáo viên.

+ Cung cấp một số thao tác như : thêm sinh viên (hoặc giáo viên, môn học, lớp học), cập nhật thông tin sinh viên (hoặc giáo viên, môn học, lớp học) , hoặc là xóa sinh viên (hoặc giáo viên, môn học, lớp học) khỏi danh sách trong trường hợp học viên chuyển trường.

Để hoàn thành dự án này, em đã tham khảo nhiều nguồn khác nhau như qua bạn bè, các bài học trên lớp, sự hướng dẫn thêm của thầy Thắng – giáo viên bộ môn.

**Mục lục :**

Lời nói đầu 2

Mục lục 2

Phần I : Phân tích cấu trúc chung 3

1. Giới thiệu chung 3
2. Các đối tượng xuất hiện trong dự án và mối quan hệ 3

Phần II : Triển khai dự án 4

1. Java Core 4
2. Java Desktop 4
3. Kết hợp Java Desktop và MySQL 4

Phần III : Hoàn thiện dự án 3

Phần IV : Ưu, nhược điểm và hướng phát triển 3

**Phần I : Phân tích cấu trúc chung:**

1. Giới thiệu chung:

Phần mềm quản lý trường học sẽ có các trang như sau:

+ Thông tin học sinh

+ Thông tin giáo viên

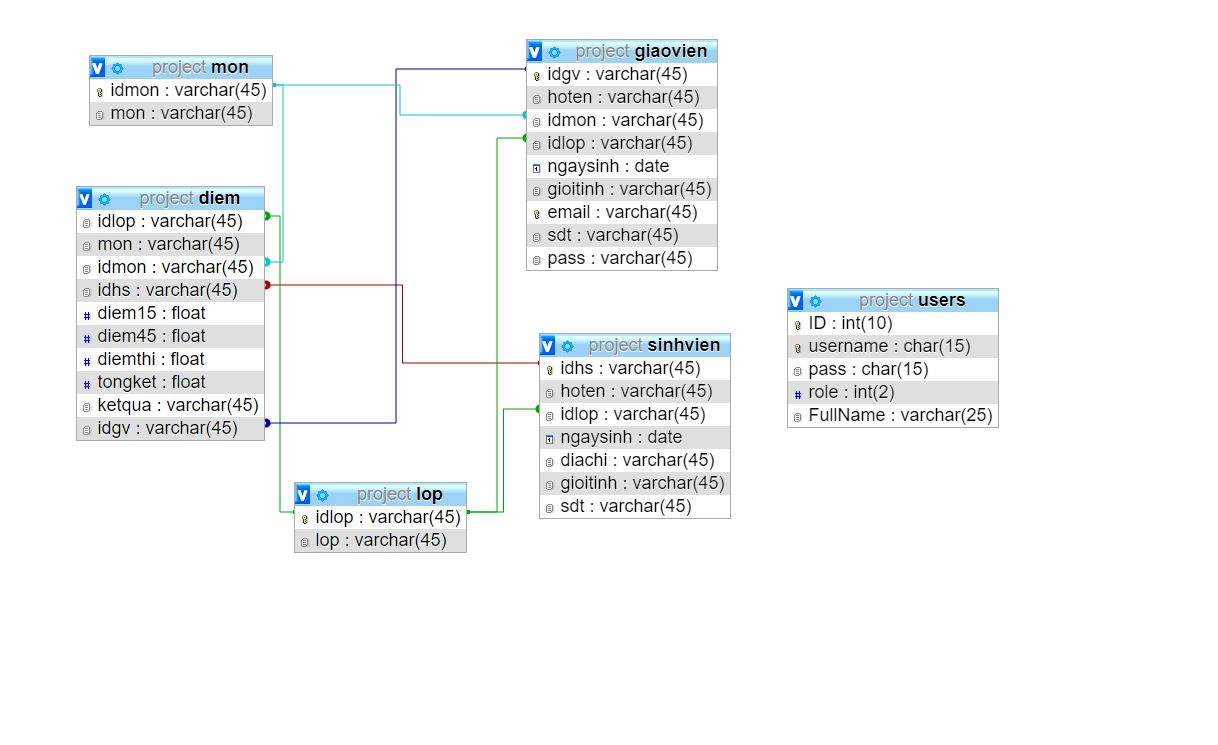
+ Điểm

+ Môn học

+ Lớp

Trong mỗi trang sẽ có các giao diện để người dùng có thể thao tác với cơ sở dữ liệu bao gồm các button, textfield, table,….

1. Sơ đồ mối quan hệ giữa các đối tượng:



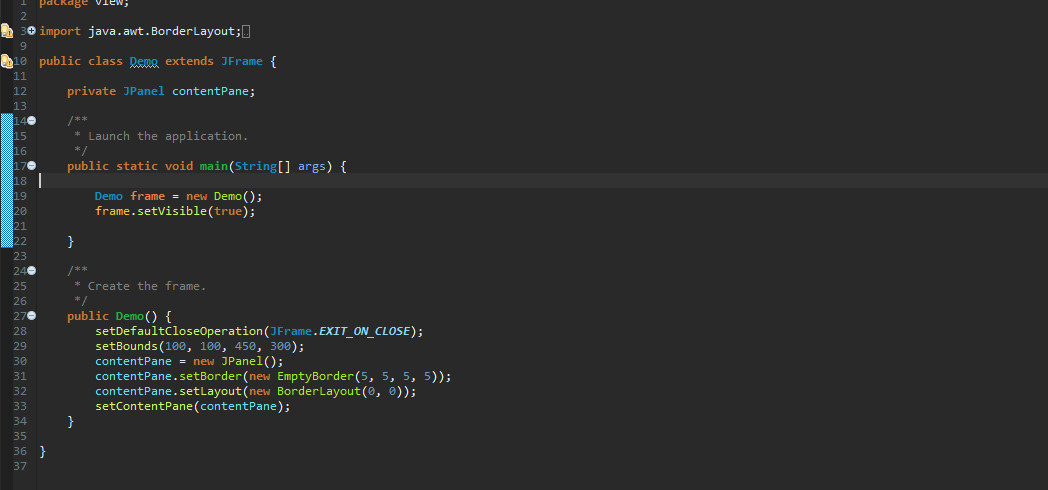
**Phần II : Triển khai dự án**

1. **Java Core**

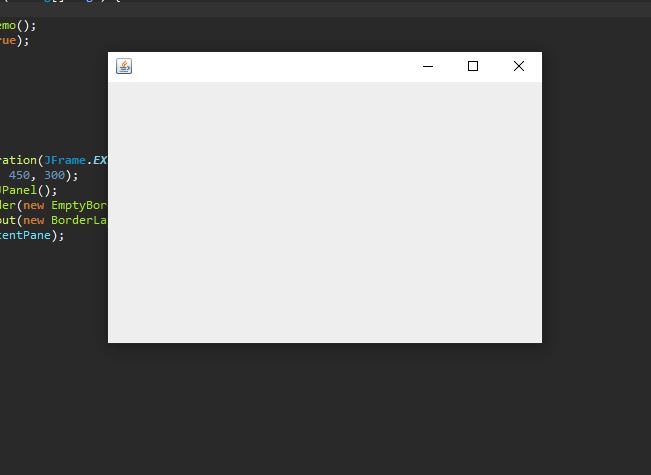


Tạo mới một class rồi tạo các hàm tạo có tham số và không có tham số, sử dụng các phương thức getter và setter để lấy và cũng như xét giá trị cho các trường

**2. Java Destop**

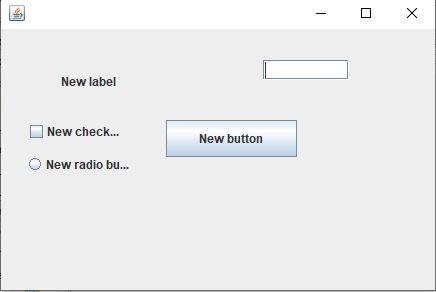


Khi muốn làm việc với java swing (destop) thì ta cần phải extends class đó với Jframe, sau đó setSize, xét layout rồi add vào ContentPane để có được 1 giao diện swing cơ bản.



Khi muốn thêm một số Components cho giao diện t chỉ cần tạo mới rồi add vào ContentPane

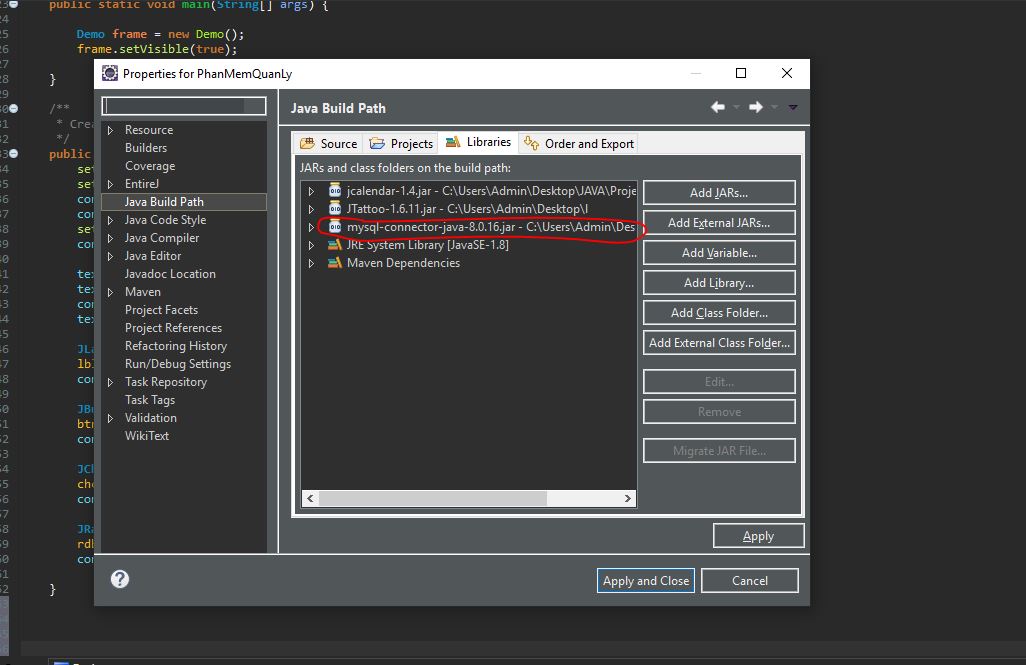




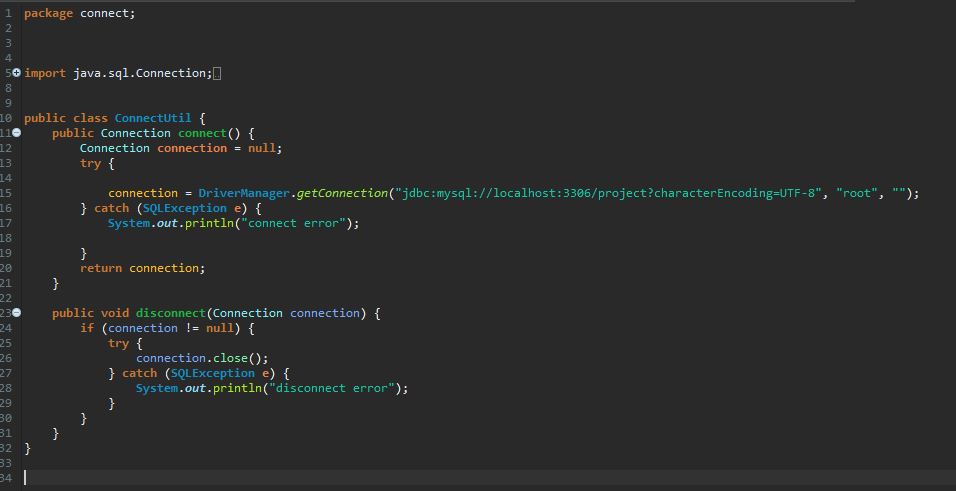
*\* Một vài Components đơn giản \* .*

1. **Java Destop & MySQl & Database**

Đầu tiên để làm việc được với database thì cần phải kết nối với cơ sở dữ liệu MySQL bằng cách cung cấp thư viện Driver phù hợp.

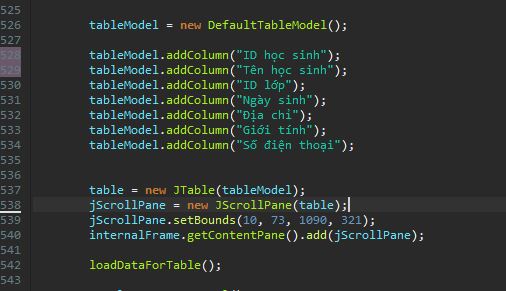


Sau đó viết tạo một class để kiểm tra kết nối



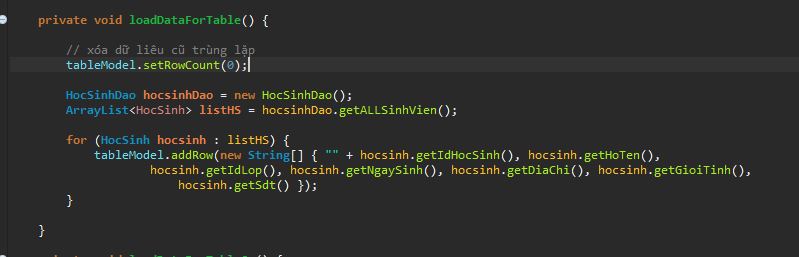
Dùng khối lệnh try catch để bắt lỗi và đưa ra thông báo nếu việc kết nối với databasse không thành công .

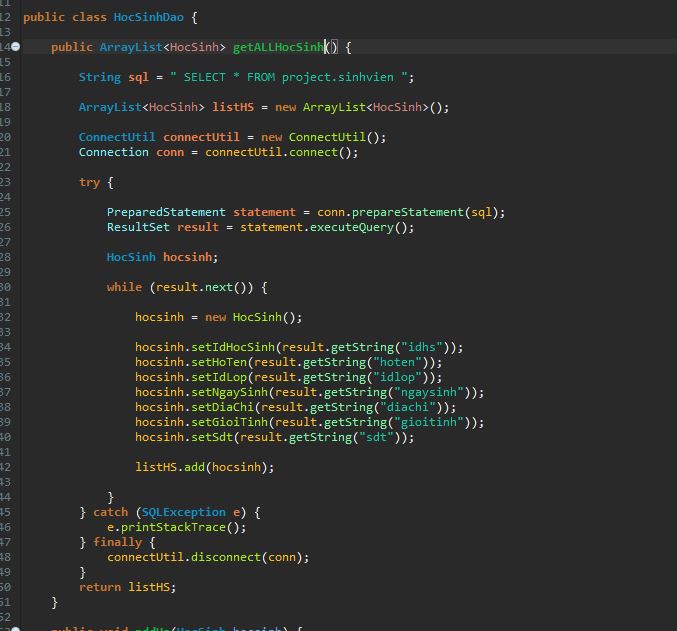
Sau khi đã kết nối thành công thì ta dđi tạo bảng để load dữ liệu từ databasse lên bảng.



Dùng DefaultTableModel để tạo tên cho các cột của bảng rồi add vào JTable dùng JscrollPanel để chứa table để phòng khi dữ liệu bị đầy quá bảng

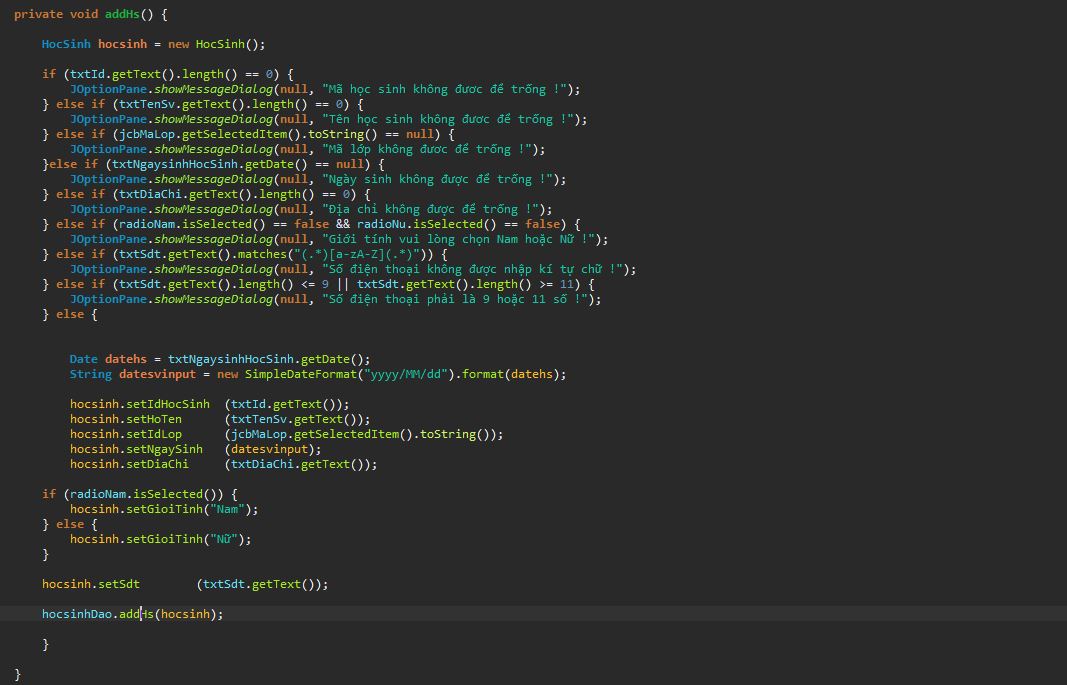
Sử dụng hàm LoadDataForTable() kết hợp với hàm getALLHocSinh() để lấy dữ liệu từ database lên table.



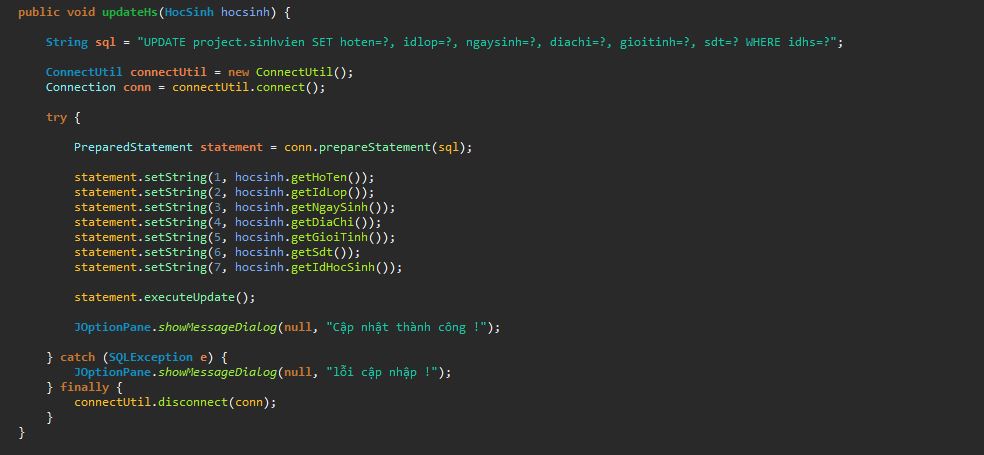


Rồi dùng cách hàm thêm sửa xóa để thao tác với bảng

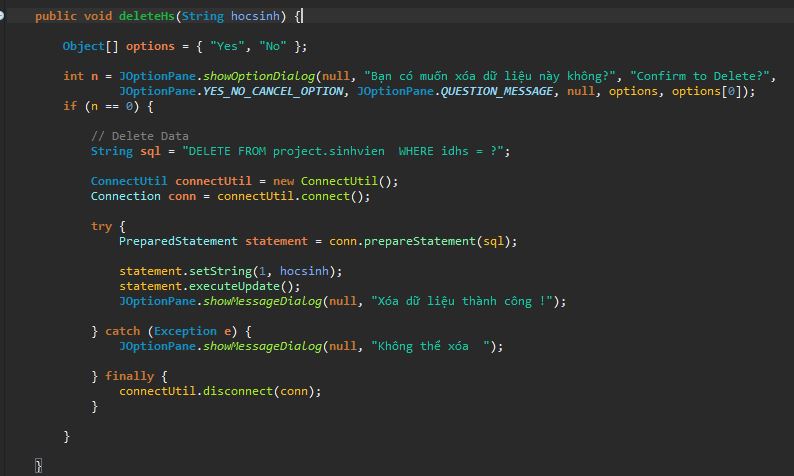
* Hàm thêm ( tạm gọi là ADD để phân biệt) ( Học sinh, giáo viên, môn học,… ) : hàm này có vai trò thêm đối tượng vào database. Ví dụ dưới đây là thêm sinh viên, sử dụng các phương thức có trong đối tượng là Student.



* Hàm sửa ( UPDATE ) ( Học sinh, giáo viên, môn học,… ) : hàm này được sử dụng với mục đích sửa lại các thông tin đã có từ trước của các đối tượng trên database.

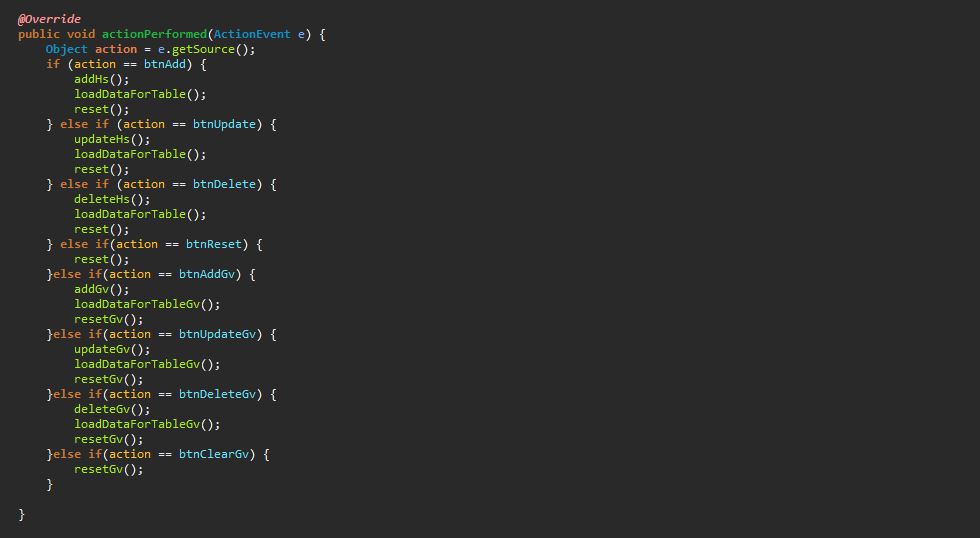


* Hàm xóa ( DELETE ) ( Sinh viên, giáo viên, môn học,… ) : hàm này được dùng để xóa thông tin các đối tượng đã có trước trên database trong trường hợp sinh viên chuyển trường hoặc giáo viên chuyển nơi công tác,…



Đây là các hàm cơ bản dùng để tương tác trực tiếp với database. Sau đây là các hàm lấy thông tin từ giao diện để chuẩn bị các yếu tố cần thiết để sử dụng các hàm ở trên. ( Dưới đây đều là các hàm liên quan đến đối tượng sinh viên để làm đại diện cho các đối tượng khác)

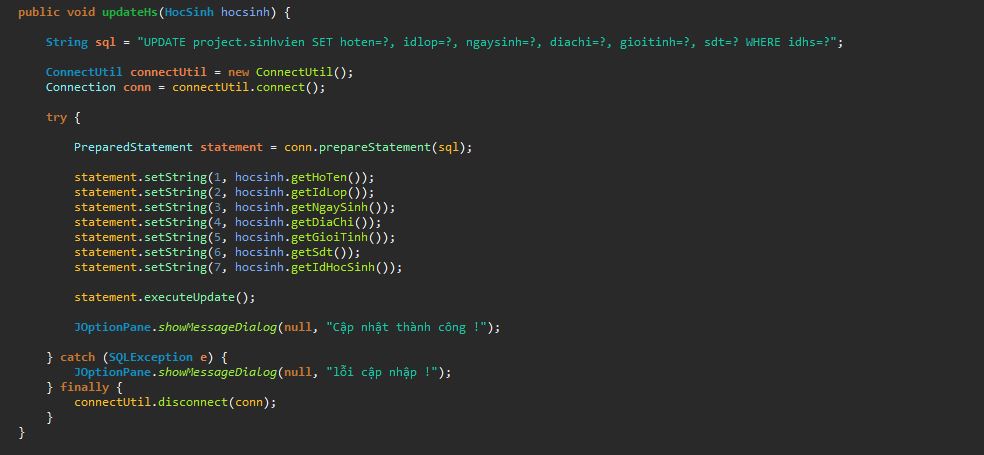
* Hàm ghi đè ActionPerformed : hàm được sử dụng với vai trò như điều hướng các tính năng của giao diện ví dụ như kick nút ADD => gọi đến hàm ADD,…



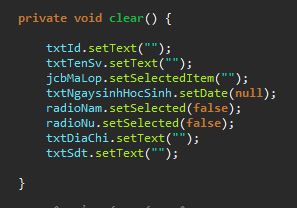
* Hàm addHs : lấy các thông tin từ giao diện cung cấp cho hàm ADD.



* Hàm updateHs : dùng kết hợp với thao tác mouseClick để cập nhật thông tin đối tượng.

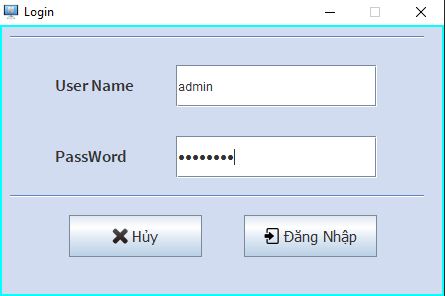
+

- Hàm clean : làm trống vùng nhập.



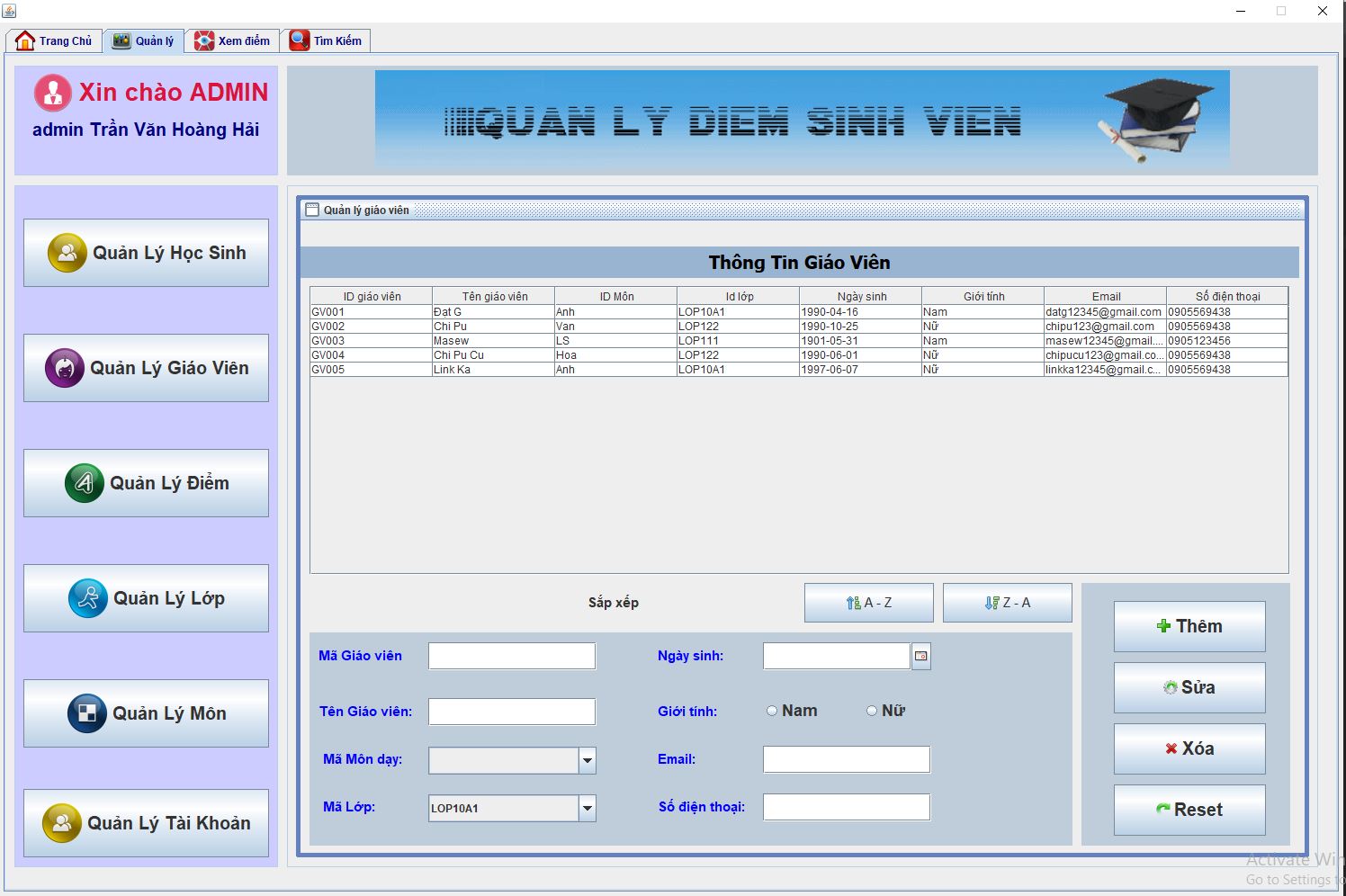
1. Các giao diện quản lý :

+ Đăng nhập :

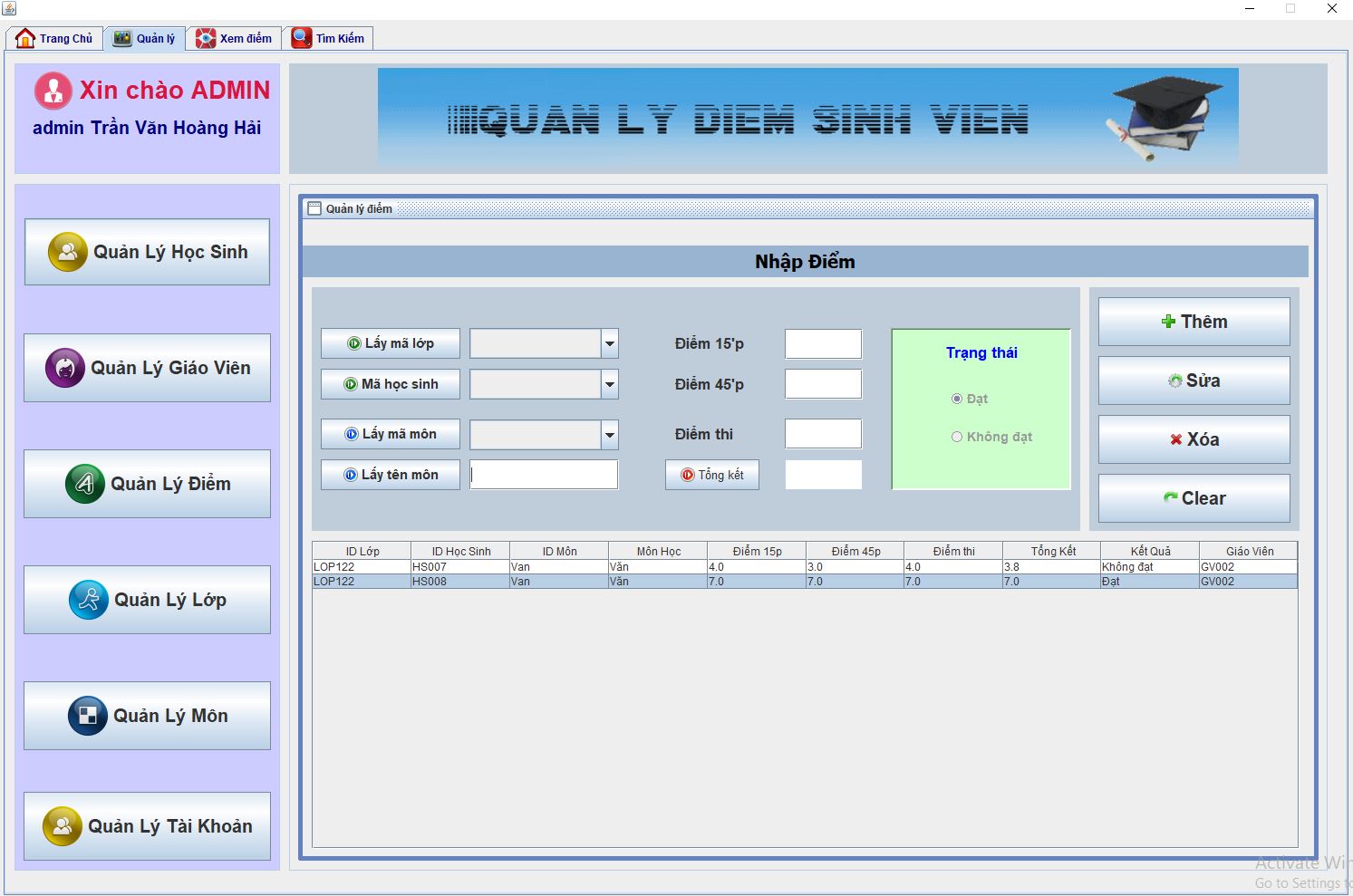


+ Quản lý sinh viên :

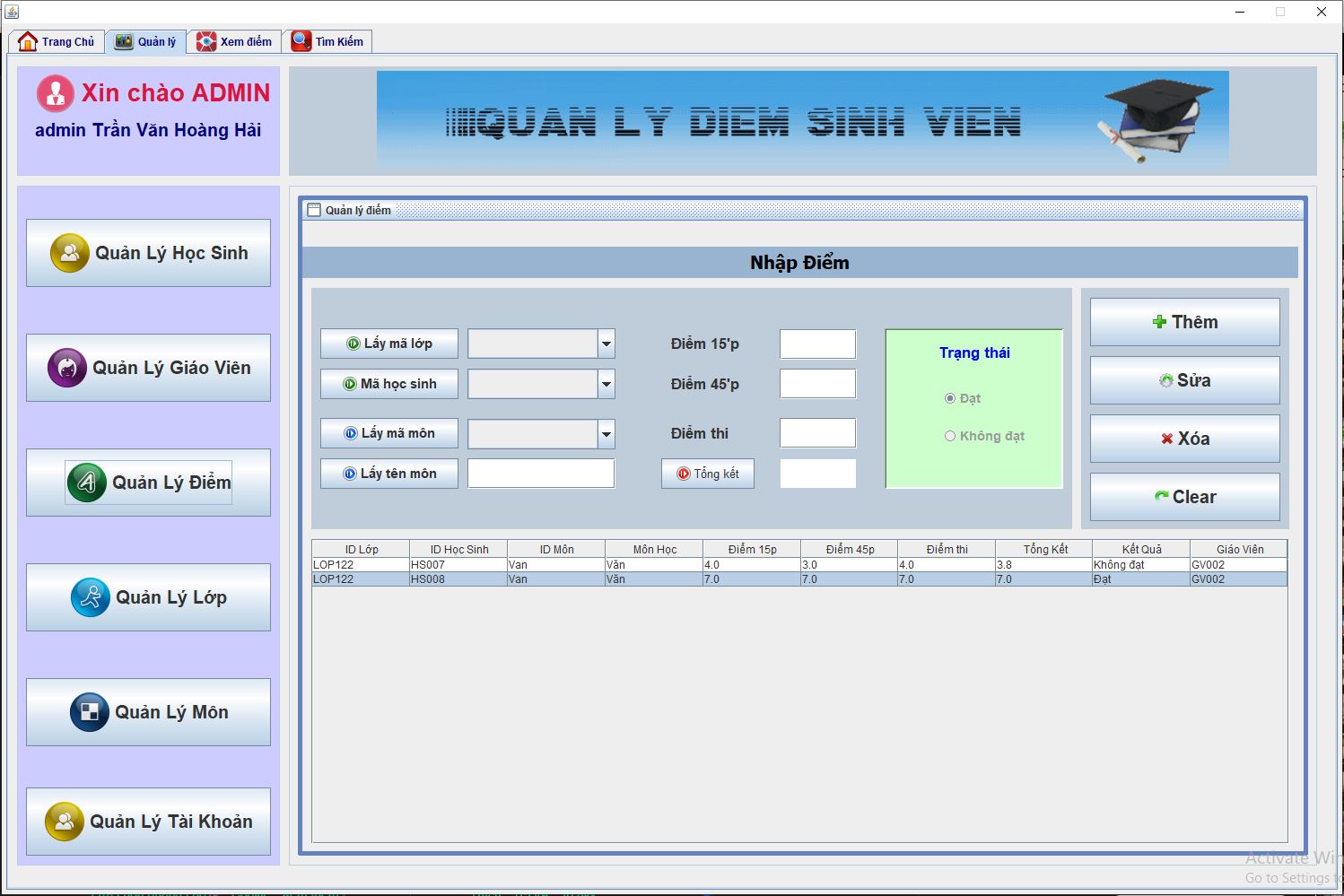


+ Quản lý giáo viên: 

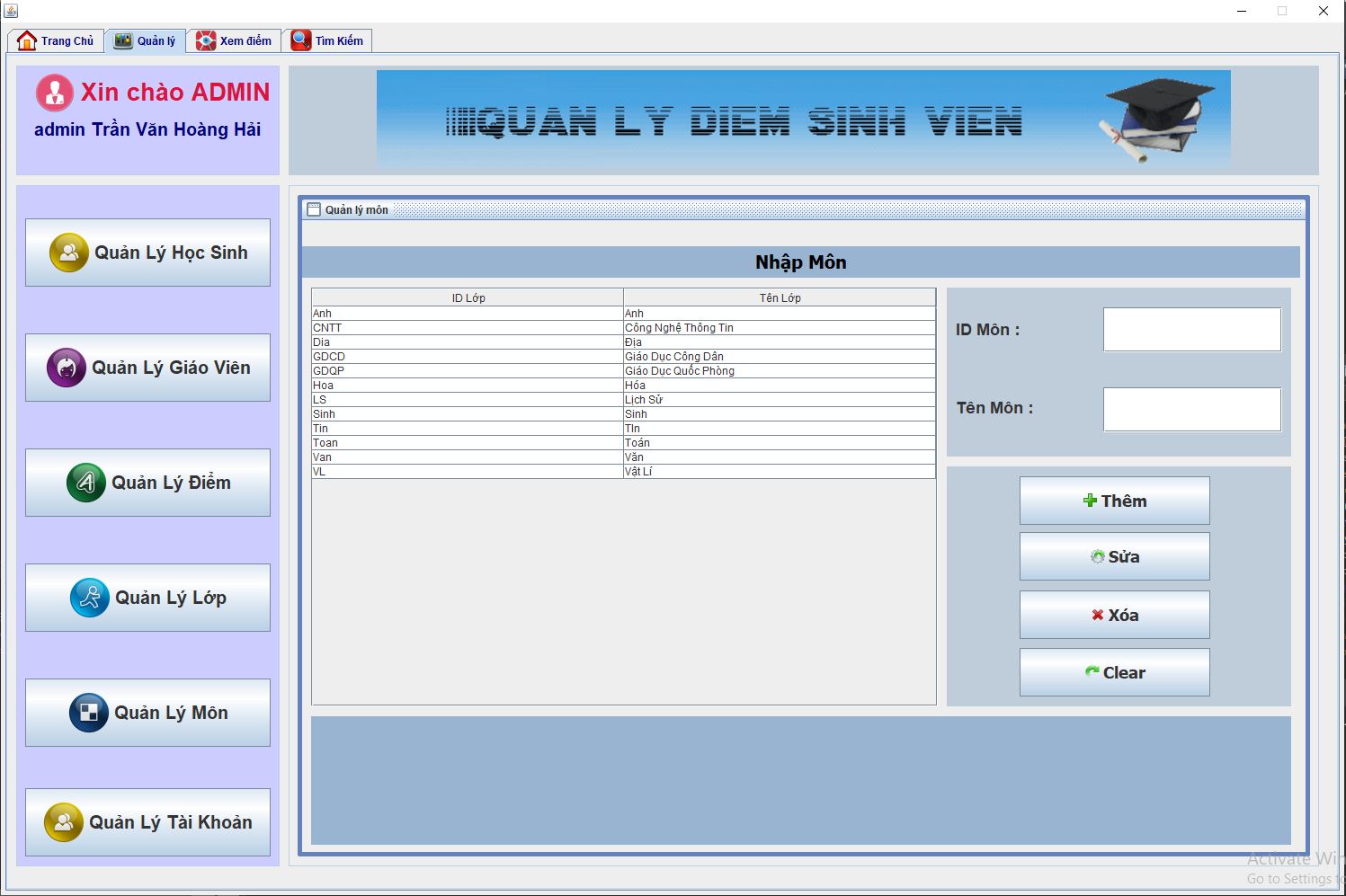
+ Quản lý lớp học :



+ Quản Lý Điểm:



+ Quản Lý Môn Học:



\* Ngoài ra còn có phân quyền cho hệ thống:

- Có 4 phân quyền cơ bản cho hệ thống là :

+ ADMIN của nhà trường sẽ nhập toàn quyền.

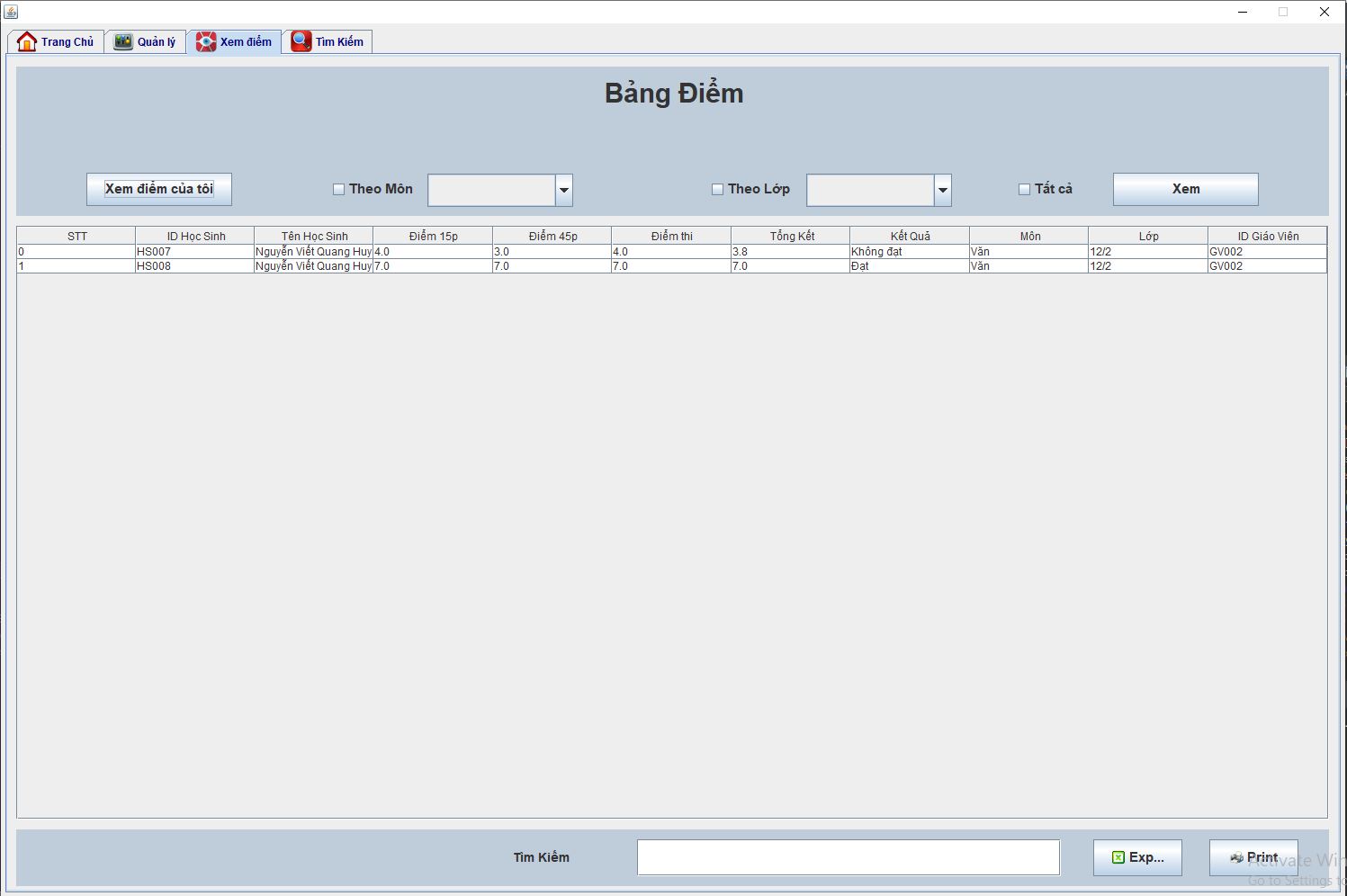
+ Giáo vụ của nhà trường sẽ quản lí học sinh, giáo viên, môn và lớp



+ Giáo viên của nhà trường sẽ được phép xem thông tin của các học sinh, thêm, sửa, xóa điểm.



+ Học sinh sẽ được quyền xem điểm.



**Phần IV : Ưu, nhược điểm của hệ thống và hướng phát triển**

1. Ưu và nhược điểm :

+ Ưu điểm :

- Hệ thống xây dựng gọn nhẹ.

- Hệ thống có sự ràng buộc về dữ liệu.

- Giao diện khá thân thiện với người dùng.

+ Nhược điểm :

* Cách tổ chức chưa chặt chẽ.
* Bắt lỗi còn hạn chế, chưa hoàn thiện
* Các chức năng còn đơn giản, chưa linh động.

1. Hướng phát triển:

* Khả năng xử lý các sự kiện, các lỗi ngoài ý muốn tốt hơn của chương trình và dùng thao tác lên chương trình.
* Hoàn thiện tốt hơn về lập trình Java core, thiết kế Desktop và các ràng buột dữ liệu.
* Nâng cao sự linh hoạt của chương trình.
* Nâng cao kỹ thuật, hoàn thiện các phần còn thiếu của chương trình.

***Trân trọng cảm ơn quý thầy cô đã giúp đỡ em trong suốt quá trình hoàn thành dự án.***