TRƯỜNG ĐẠI HỌC PHENIKAA

KHOA CÔNG NGHỆ THÔNG TIN

⸎⸎⸎⸎⸎

**Logo, company name

Description automatically generated**

**Báo cáo bài tập lớn Môn Lập trình hướng đối tượng**

|  |  |
| --- | --- |
| **GVHD:** | **Trần Đăng Hoan** |
| **SVTH:** | **Nguyễn Việt Long** |
|  | **Phí Tài Minh** |
|  | **Đinh Xuân Kiên** |
|  | **Lê Hoàng Ngọc Tú** |

**Đề tài: Quản lý Sinh viên**

HÀ NỘI, THÁNG 06/2022

Table of Contents

[I. Giới thiệu đề tài 3](#_Toc106051622)

[1. Ngôn ngữ lập trình java 3](#_Toc106051623)

[2. Đặt vấn đề 4](#_Toc106051624)

[3. Cơ cấu tổ chức 4](#_Toc106051625)

[II. Mô tả nghiệp vụ 4](#_Toc106051626)

[1. Yêu cầu chung đối với phần mềm 5](#_Toc106051627)

[1.1. Yêu cầu người sử dụng 5](#_Toc106051628)

[1.2. Yêu cầu hệ thống 5](#_Toc106051629)

[2. Tổng quan chức năng phần mềm 6](#_Toc106051630)

[2.1. Đăng nhập 6](#_Toc106051631)

[2.2. Đăng ký 6](#_Toc106051632)

[2.3. Quản lý sinh viên 6](#_Toc106051633)

[III. Thiết kế dữ liệu 7](#_Toc106051634)

[1. Mô hình ER 7](#_Toc106051635)

[2. Các Table 7](#_Toc106051636)

[2.1. Table system\_account 7](#_Toc106051637)

[2.2. Table student 8](#_Toc106051638)

[2.3. Table subject 8](#_Toc106051639)

[2.4. Table subject\_score 9](#_Toc106051640)

[2.5. Table k12,k13,k14,k15\_department 9](#_Toc106051641)

[IV. Nhiệm vụ và mô hình 10](#_Toc106051642)

[1. Nhiệm vụ 10](#_Toc106051643)

[2. Mô hình UML 10](#_Toc106051644)

[2.1. Kết nối chương chình với Database 10](#_Toc106051645)

[2.2. Form Đăng Ký 10](#_Toc106051646)

[2.3. Form Đăng Nhập 11](#_Toc106051647)

[2.4. Form Chính 11](#_Toc106051648)

[2.5. Form Quản lý sinh viên 11](#_Toc106051649)

[2.6. Form quản lí môn học 12](#_Toc106051650)

2.7. Các class trong chương trình biểu diễn trên UML……………………………………………………………………14

[V. Chức năng chính 16](#_Toc106051651)

[VI. Kết luận 21](#_Toc106051652)

# Giới thiệu đề tài

## Ngôn ngữ lập trình java

* Java là một ngôn ngữ lập tình hướng đối tượng (OOP) và dựa vào các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiệt kế để biên dịch mã nguồn thành bytecode, sau đó được môi trường thực thi ( runtime environment)
* Java trước đây chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C/C++, sau khi nhờ công nghệ “biên dịch tại chỗ” – Just in time compilation, khoảng cách nay đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá tình dịch/chạy.
* Cú pháp Java được vay mượn nhiều từ C/C++ nhưng cú pháp hướng đối tượng đơn giản và ít tính năng xử lý cấp thấp hơn. Do đó việc viét chương trình bằng Java dễ , đơn giản, đỡ tốn công sửa lỗi hơn.
* Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động “dọn dẹp rác”. Programmer không quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà Programmer không đóng các stream thì rò rỉ dữ liệu vẫn có thể xảy ra.
* Ưu điểm
  + Java: Cross – Platform là đặc trưng. Cross – Platform có nghĩa là bạn có thể viết mã nguồn trên một IDE ở một máy tính và thực thi chương trình ở bất kì máy tính sử dụng hệ điều hành nào. Với điều kiện máy tính thực thi chương trình phải có hệ thống trình biên – thông dịch riêng được gọi là Java Virtual Machine, JVM có nhiều phiên bản dành riêng cho từng hệ điều hành ( Windows, Linus, MAC OS)
  + Công nghệ mở và phát triển không ngừng. Cạnh tranh với các công nghệ .NET, cộng đồng công nghệ Java cũng không kém cạnh và miễn phí đường chính mình
  + Nâng cao hiệu năng phát triển phần mềm, liên quan tới 3 yếu tố trong OOP:
    - Tính Mô- đun: Tách biệt các nhiệm vụ trong quá trình phát triển phần mềm dựa trên các đối tượng cụ thể, hay mỗi đối tượng có một nhiệm vụ riêng
    - Tính mở rộng: Các đối tượng có thể mở rộng thêm các thuộc tính mới, các hành vi mới trong tương lai
    - Tính tái sử dụng: Các đối tượng có thể tái sử dụng một ứng dụng hoặc nhiều ứng dụng khác nhau. Nhờ vào các yếu tố trên OOP giúp tăng hiệu năng phát triển phần mềm so với lập trình hướng thử tục
  + Nâng cao khả năng bảo trì phần mềm: Chính nhờ các yếu tố trên mà việc bảo trì cũng trở nên dễ dàng hơn, vì thiết kế theo mô-đun nên việc thay đổi một phần của chương trình có thể không làm ảnh hưởng đến những phần còn lại, rất phù hợp với dự án lớn, đòi việc phải bảo trì và thay đổi
  + Phát triển phần mềm nhanh hơn: Nhờ vào tính tái sử dụng mà các phần mềm được phát triển nhanh hơn, OOP thường có thư viện đối tượng phong phú, hay những đối tượng do chính bạn tự định nghĩa, các đoạn code được tối ưu hóa đều có thể tái sử dụng trong tương lai, giảm thiểu chi phí phát triển
* Nhược điểm
  + Khá phức tạp, có thể khó nhằn cho beginner, nó đòi hỏi sự tư duy dựa trên sự tương tác giữa các đối tượng, do đó chúng ta cần nắm được bản chất : lớp , đối tượng, thuộc tính, phương thức, nắm được 4 tính chất của hướng đối tượng : tính trừu tượng (abstract), tính kế thừa (inheritance), tính đóng gói (encapsulation), tính đa hình (polymorphism)
  + Chương trình có thể chậm và kích thước lớn hơn so với các chương trình lập trình hướng thủ tục. Do các phần mềm này thường yêu cầu nhiều câu lệnh hơn để thực thi, programmer cần viết ra nhiều dòng mã hơn để đảm bảo các thuộc tính, phương thức của đối tượng => kíck thước chương trình lớn
  + Lập trình hướng đối tượng không phải chìa khóa vạn năng cho nhiều vấn đề
  + Bảo mật sourcecode không tốt

## Đặt vấn đề

* Các công tác quản lí sinh viên đóng vai trò cực kì quan trọng đối với hoạt động của một Khoa trong một ngôi trường Đại học hay Cao đẳng.
* Vậy bài toán “Quản lí sinh viên” được nói đến những vấn đề cơ bản như là: Quản lí sinh viên theo khóa, theo lớp, theo các loại hình đào tạo. Quản lí môn học của các lớp theo kỳ đối với mỗi sinh viên.
* Hệ thống còn phải quản lí sinh viên theo Mã sinh viên
* Những hệ thống trên còn có đăng kí tài khoản cho sinh viên đó

## Cơ cấu tổ chức

* Một trường đại học mỗi năm tiếp nhận khá nhiều sinh viên đến nhập học và trong quá trình học như vậy nhà trường sẽ phải quản lý các kết quả học tập của từng sinh viên.
* Trong trường có nhiều khoa , ngành, mỗi khoa sẽ có một phòng giáo là nơi cập nhật thông tin của sinh viên, lớp , các ngành học ...
* Mỗi khoa sẽ có nhiều lớp học , thông tin lớp học sẽ có lớp, khóa học, năm học, năm kết thúc mỗi lớp sẽ có mã lớp. Mỗi lớp sẽ có nhiều sinh viên, mỗi sinh viên sẽ cung cấp thông tin khi nhập học như họ tên , ngày sinh, nơi sinh, địa chỉ, số điện thoại và sẽ cấp một mã sinh cho sinh viên đó
* Quá trình đào tạo , sinh viên học môn học mà khoa phân lớp, thông tin về môn học ..... Sau khi hoàn thành sinh viên sẽ có lịch thi kết thúc học phần.

# Mô tả nghiệp vụ

## 1. Yêu cầu chung đối với phần mềm

### 1.1. Yêu cầu người sử dụng

- Các chức năng của phần mềm: phải tuân theo quy chế, quy trình đào tạo của trường.

- Phần mềm phải có giao diện thân thiện: để mọi người đều có thể sử dụng được, không nhất thiết phải là người trong ngành công nghệ thông tin.

- Hệ thống phải dễ sử dụng, quản lý: đảm bảo tốt cho việc sử dụng phần mềm để quản lý cũng như tra cứu cùng thời điểm với số lượng lớn người sử dụng.

- Hệ thống phải có khả năng bảo mật tốt: tất cả mọi thông tin cá nhân chỉ có người được phân quyền mới được phép xem và chinh sửa.

- Hệ thống phải có chức năng phục hồi, sao lưu dữ liệu thường xuyên: tránh tình trạng mất, hỏng, sai lệch dữ liệu.

- Hệ thống cần có khả năng mở rộng, nâng cấp trong tương lai: để có thể thay đổi cho phù hợp với yêu cầu công tác quản lý.

- Chi phí cho hệ thống (phần mềm, phần cứng, nhân sự vận hành) phải hợp lý: không vượt quá ngân sách của trường ngưng vẫn đáp ứng được yêu cầu công việc.

- Phần mềm phải có hướng dẫn sử dụng (video, hình ảnh,...): hướng dẫn chi tiết từng chức năng để người mới sử dụng có thể làm việc tốt với phần mềm.

### 1.2. Yêu cầu hệ thống

**Yêu cầu chức năng:**

- Chức năng quản trị hệ thống: cho phép người quản trị hệ thống có thể quản lý người sử dụng, phân quyền, quản lý danh mục và vận hành hệ thống.

- Chức năng quản lý thông tin: cho phép các bộ phận, phòng ban thực hiện cập nhật và quản lý thông tin hoạt động của đơn vị mình.

- Chức năng tra cứu thông tin: cho phép người truy cập hệ thống có thể xem các thông tin mà đã được người quản trị phân quyền cho mình.

**Yêu cầu phi chức năng:**

- Giao diện thân thiện, dễ sử dụng.

- Truy xuất dữ liệu nhanh, lưu trữ dữ liệu tốt.

- Tìm kiếm nhanh, thuận tiện.

- Hệ thống bảo mật cao.

- Đáp ứng được các yêu cầu nghiệp vụ.

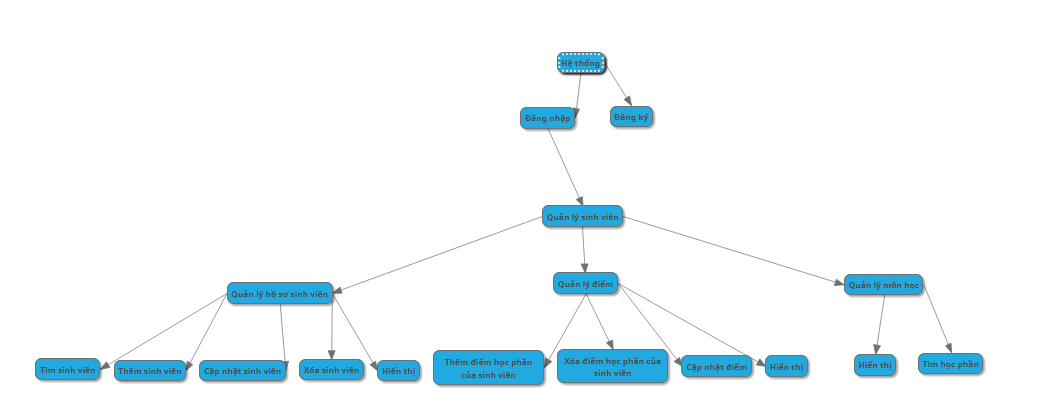
**Yêu cầu miền ứng dụng:**

- Chạy được trên các hệ điều hành khác nhau.

- Hệ quản trị cơ sở dữ liệu (mySQL).

- Giao diện thiết kế theo một chuẩn nhất định.

## 2. Tổng quan chức năng phần mềm



Tổng quan chức năng phần mềm qlsv

### 2.1. Đăng nhập

- Đăng nhập vào chương trình quản lý sinh viên bằng tài khoản của hệ thống

### 2.2. Đăng ký

- Đăng ký tài khoản hệ thống để có thể vào được chương trình

### 2.3. Quản lý sinh viên

Bao gồm:

* Quản lý hồ sơ sinh viên
* Quản lý điểm
* Quản lý môn học

#### 2.3.1. Quản lý hồ sơ sinh viên

- Thêm thông tin cá nhân sinh viên vào hồ sơ

- Cập nhật lại thông tin sinh viên dựa trên MSSV

- Xóa sinh viên dựa trên MSSV

- Hiển thị toàn bộ sinh viên trong hồ sơ

- Tìm sinh viên dựa trên MSSV

#### 2.3.2. Quản lý điểm

- Thêm điểm một học phần của một sinh viên

- Cập nhật lại điểm cho sinh viên

- Xóa điểm học phần của sinh viên

- Hiển thị toàn bộ điểm trong CSDL

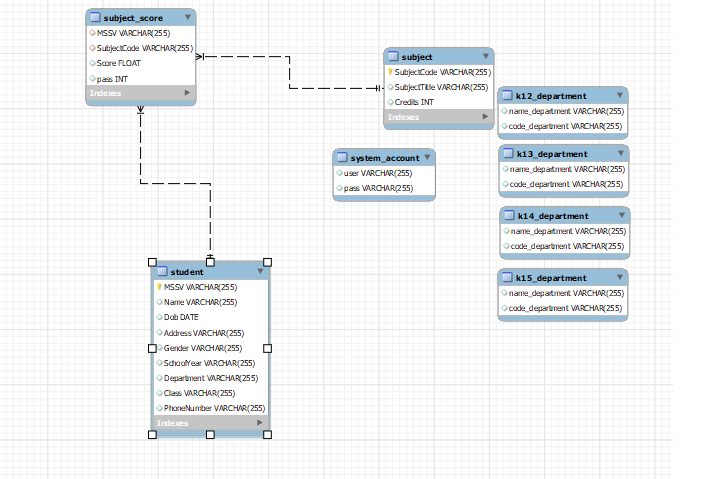
##### 2.3.3. Quản lý môn

- Hiển thị mã học phần, tên học phần và số tín chỉ

- Tìm học phần dựa trên mã học phần

# Thiết kế dữ liệu

## Mô hình ER



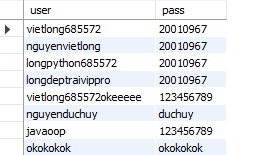
## Các Table

### 2.1. Table system\_account

Các trường và kiểu dữ liệu:

* user Varchar(255) // Tên tài khoản đăng nhập
* pass Varchar(255) // Mật khẩu đăng nhập

Dùng để lưu tên tài khoản và mật khẩu đăng nhập hệ thống

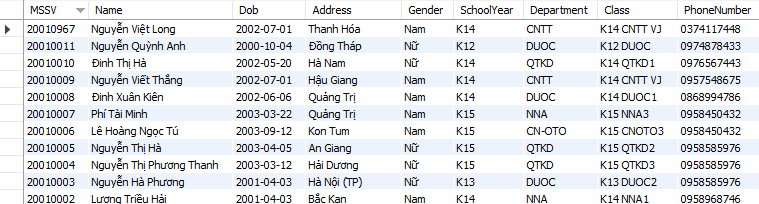


### Table student

Các trường và kiểu dữ liệu:

* MSSV Varchar Khóa chính // mã số sinh viên
* Name Varchar // Tên sinh viên
* Dob Date // ngày sinh
* Address Varchar // Quê quán
* Gender Varchar // Giới tính
* SchoolYear Varchar // Khóa
* Department Varchar // Khoa
* Class Varchar // Lớp
* PhoneNumber Varchar // số điện thoại

Chưa hồ sơ thông tin cá nhân của sinh viên



### Table subject

Các trường và kiểu dữ liệu:

* SubjectCode Varchar //Mã học phần
* SubjectTitle Varchar //Tên học phần
* Credits int // Số tín chỉ

Chứa thông tin của các học phần

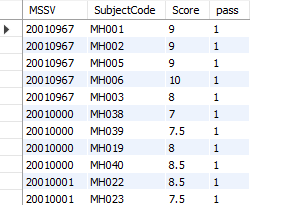


### Table subject\_score

Các trường và kiểu dữ liệu:

* MSSV Varchar (Khóa ngoại liên kết với khóa chính MSSV Table Student) // mã số sinh viên
* SubjectCode Varchar (Khóa ngoại liên kết với khóa chính SubjectCode Table subject) // Mã học phần
* Score float // điểm học phần
* Pass int // pass==1 ? đạt : trượt

Chứa tất cả các điểm của sinh viên những học phần đã thi

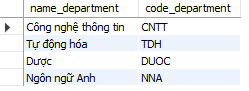
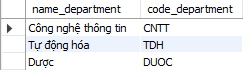


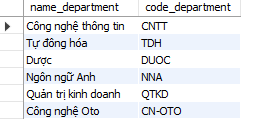
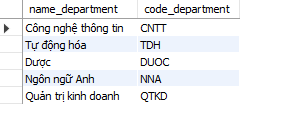
### Table k12,k13,k14,k15\_department

Các trường và kiểu dữ liệu:

* Name\_department varchar // Tên khoa
* Code\_department varchar // Viết tắt của tên khoa

Chứa các ngành học của từng khóa





# Nhiệm vụ và mô hình

## Nhiệm vụ

* Theo dõi toàn bộ quá trình học tập của sinh viên trong suốt quá trình học ở trường từ khi nhập học đến khi ra trường đảm bảo sẽ không có sai sót trong quá trình nhập điểm. Hệ thông này sẽ giúp bên phong đào tạo theo dõi, đánh giá kết quả học tập của sinh viên nhanh nhất.

## Mô hình UML

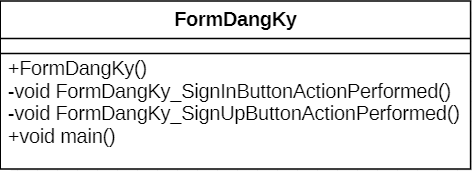
### 2.1. Kết nối chương chình với Database

Table

Description automatically generated

* Với mô hình UML trên thì ta sẽ viết source code dựa trên đó
* Đầu tiên ta cần viết những thuộc tính để kết nối đến Database như Link dẫn, tài khoảng, mật khẩu để truy cập vào database
* Những thuộc tính đó đươc viết theo kiểu Private hay nói cách khác ta cần bảo mật đến Database
* Tiếp đến ta cần vận hành dòng code để truy cập theo kiểu Public

### 2.2. Form Đăng Ký



* Với Form Đăng ký ta sẽ vận hành nó với các hàm
* Ta sử dụng hàm Void theo kiểu Private để viết Form Đăng Nhập hay cũng viết những chức năng cho Giao diện đăng kí

### Form Đăng Nhập

A picture containing text

Description automatically generated

* Tiếp đến FormDangNhap ta vận hành nó bằng cách
* Sử dung hàm Void theo kiểu Pivate để viết các chức năng cho Giao diện Đăng Nhập

### Form Chính

Text

Description automatically generated

* Ta viết code theo mô hình UML: FormChinh
* Với cách vận hành ta đầu tiên ta có kiểu Public FormChinh
* Tiếp theo sẽ dùng hàm Void theo kiểu Private để viết chương chính
* Dùng các hàm để viết chương chính cũng như viết và thiết kế giao diện cho Form Chính

### Form Quản lý sinh viên

Text

Description automatically generated with medium confidence

* Đầu tiên ta có những thuộc tính Student sv, Connection...
* Tiếp đến cách vận hành:
  + Sử dụng hàm void như void initTable, void loadDatatoTable.. dùng theo kiểu Public
  + Tiếp theo sử dụng hàm void như void QLSV...... theo kiểu Private
  + Sử dụng các hàm hay các thuộc tính để thiết kế giao diện

## Form quản lí môn học

Text

Description automatically generated

* Tương tự quản lí Sinh viên thì quản lí môn học cũng như vậy
* Ta cũng cần có những thuộc tính để viết chương trình hay để khởi tạo những thao tác cho giao diện
* Ta cũng sử dụng các hàm để viết giao diện hay những thao tác cần có trong chương chình quản lí sinh viên hay môn học và cách sắp xếp nó thật hợp lí
* Tóm lại với mô hình UML giúp ta dễ dàng viết chương trình một cách hiệu quả
* UML cũng giúp ta dễ dàng hiểu chương chình đang viết gì và cần gì
* Với những mô hình trên giúp ta thuận tiện hơn rất nhiều
  1. Các class có trong chương trình biểu diễn trên UML
* Class Student:

Text

Description automatically generated with medium confidence

* Class SubjectExcute:

Table

Description automatically generated

* Class Subject:

Text

Description automatically generated

* Class StudentExcute

A picture containing table

Description automatically generated

* Class SubjectScore

Text

Description automatically generated

* Class SubjectScoreExcute:

A picture containing text

Description automatically generated

# Chức năng chính

* Đầu tiền đến với chức năng và giao diện của chương trình quản lí sinh viên ta cần phải đến với giao diện đăng nhập tài khoản

1. Giao diện đăng nhập tài khoản

Graphical user interface, application

Description automatically generated

* Như hình trên là giao diện ta sẽ thấy đầu tiên khi chạy chương trình hay là phần mềm quản lí sinh viên
* Đây là giao diện “ Đăng nhập ” bao gồm
  + Form ghi “Tài khoản” và “Mật khẩu”
  + Lệnh Ghi nhớ tài khoản: Ghi nhớ tài khoản mỗi lần vào phần mềm không cần điền
  + Có 2 nút : Đăng nhập và Đăng ký
    - Đăng nhập:
      * Trường hợp Tài khoản, Mật khẩu không đúng sẽ báo lỗi
      * Trường hợp đúng sẽ đăng nhập vào giao diện chính
    - Đăng ký: Sẽ chuyển sang giao diện đăng ký
* Nếu đã có tài khoản ta đăng nhập sẽ hiển thị ra Giao diện chính của chương trình
* Nếu chưa có tài khoản, nút “Đăng ký” là nơi ta tạo tài khoản để nhập thông tin cá nhân

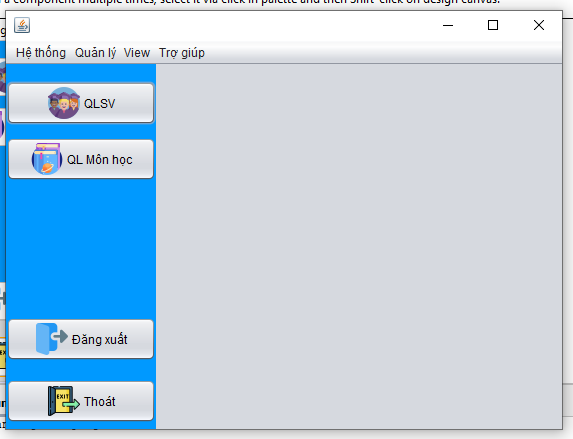
1. Giao diện đăng kí tài khoản

Graphical user interface, text, application, email

Description automatically generated

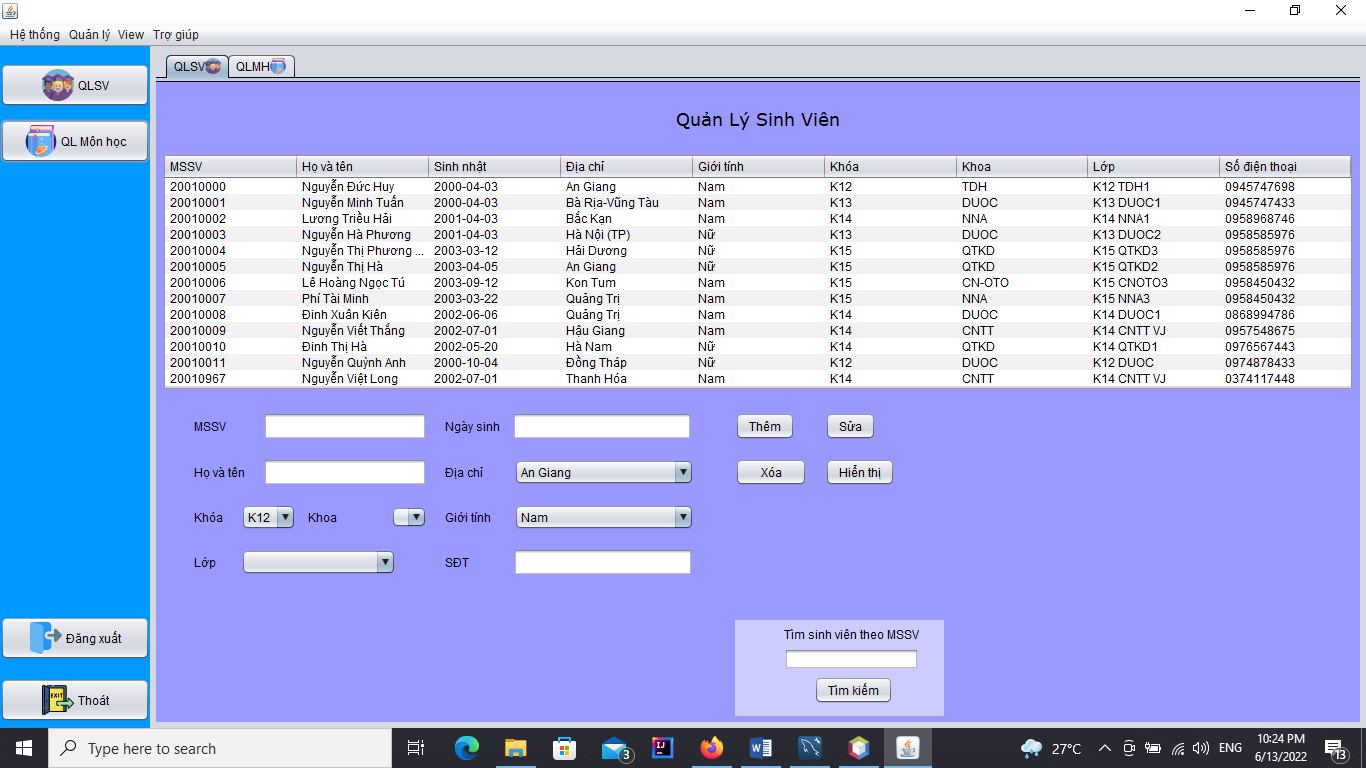
* Đây là giao diện đăng ký tài khoản khi chưa có tài khoản
* Ta thấy có một form để điền thông tin tạo tài khoản
* Giao diện có 2 nút:
  + Đã có tài khoản, đăng nhập?? : Có tài khoản khi chọn nút này nó sẽ trở về giao diện đăng nhập
  + Đăng ký: Tạo tài khoản theo mẫu vừa diền
    - Trường hợp mất khẩu không khớp nó sẽ bảo lỗi
    - Trường hợp mất khẩu hợp lệ tài khoản đăng kí thành công và chuyển sang giao diện Đăng nhập

1. Giao diện chính của chương trình



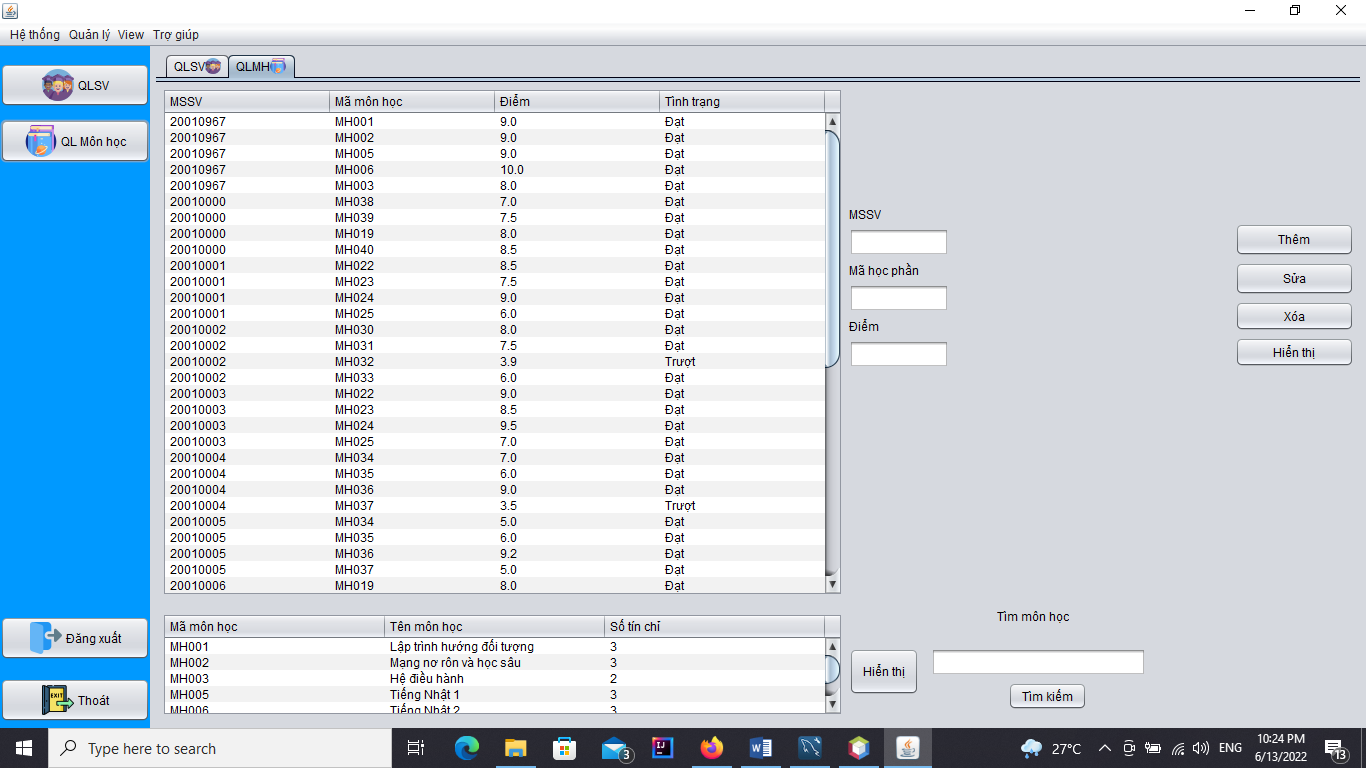
* Khi đã đăng nhập thành công ta sẽ chuyển đến giao diện chính của phần mềm
* Giao dien gồm hệ thống , quản lí, view, trợ giúp ở phần trên
  + Hệ thống:
    - Đăng xuất: Đăng xuất ra khỏi hệ thống về giao diện đăng nhập , được tích hợp trên bàn phím Crtl + S
    - Thoát: Thoát khỏi phần mềm, tích hợp trên bàn phím Crtl + E
  + Quản lí:
    - Sinh viên: Truy cập vào giao diện quản lí sinh viên, tích hợp trên bàn phím Crtl +Alt + S
    - Môn học: Truy cập vào giao diện quản lí môn học, tích hợp trên bàn phím Crtl +Alt + M
  + View: Sửa form của giao diện hiện tính năng này chưa tích hợp đươc
  + Trợ giúp: Trợ giúp trong chương trình
* Tiếp đến các nút QLSV,QL Môn học, Đăng Xuất ,Trợ giúp
  + QLSV: truy cập vào giao diện quản lí sinh viên
  + QL Môn Học: Truy cập vào giao diên quản lí môn học
  + Đăng xuất: Đăng xuất về giao diện đăng nhập
  + Trợ giúp: Trợ giúp trong Chương trình

1. Giao diện quản lí sinh viên



* Nếu muốn xem thông tin cá nhân của sinh viên hãy ấn vào nút QLSV hoặc Crtl + Alt + S đề truy cập giao diện
* Trong giao diện này ta sẽ thấy một danh sách trống nếu chưa có 1 thông tin nào được đưa vào nếu đã có thông tin nó sẽ hiển thị Tên, MSSV, Địa chỉ ....của sinh viên đó
* Đầu tiên ta sẽ có một vài tính năng: Thêm, Sửa, Xóa, Hiển thị, Tìm kiếm
  + Thêm: Thêm thông tin cá nhân của sinh viên như MSSV, ngày sinh, khóa
  + Sửa: Sửa thông tin sinh viên như MSSV...
  + Xóa: Xóa sinh đó khi đã chuyển trường, khoa....
  + Tìm kiếm: Tìm kiếm theo mã số sinh viên và hiển thị nó ở giao diện
    - Trường hợp không tìm thấy mã số sinh viên sẽ báo lỗi không tìm thấy
    - Trường hợp có sẽ hiển thị thông tin sinh viên đó
* Trong giao diện này ta vẫn có nút Đăng Xuất hay Trợ giúp hay QL Môn học, có thể chuyển sang giao diện khác hoặc Đăng xuất ra giao diện Đăng nhập
* Có thể dùng những phím tắt như : Ctrl + Alt + S chuyển sang giao diện khá

1. Giao diện quản lý môn học



* Khi đã có thông tin sinh viên, điều ta cần làm là theo dõi quá trình học của sinh viên thông qua điểm mà sinh viên đó đạt được
* Vậy giao diện quản lí môn học cho sinh viên là rất cần thiết
* Trong giao diện này ta có một danh sách quản lí điểm và tình trạng học lực của sinh viên theo mã sinh viên và điểm
* Danh sách thứ hai là quản lí môn học có mã môn học, tên môn học và số tín chỉ
* Nếu muốn xem danh sách môn học thì
  + Tìm kiếm: Tìm kiếm môn học, có thể tìm kiếm theo tên hoặc Mã môn học
    - Trường hợp không có môn học đó thì chương trình sẽ báo là lỗi không tìm thấy
    - Trường hợp có môn học chương trình sẽ hiển thị
  + Cũng như quản lí sinh viên thì cũng có Thêm, Sửa, Xóa, Hiển thị
    - Thêm: Thêm điểm cho sinh
    - Sửa: Sửa điểm cho sinh viên
    - Xóa: Xóa điểm của sinh viên
    - Hiển thị: Hiển thị thông tin điểm số của sinh viên
* Ở giao diện này cũng có các tính năng Đăng xuất ,Trợ giúp, QLSV.... có thể chuyển sang giao diện khác

# Kết luận

1. Ưu điểm

* Rút ngắn thời gian chờ đợi của sinh viên
* Sử dụng máy tính vào các công việc tìm kiếm các thông tin chi tiết về điểm sinh viên sẽ dễ dàng nhanh chóng và thuận tiện
* Việc lưu trữ sẽ đơn giản, không cần phải có nơi lưu trữ lớn, các thông tin về sinh viên sẽ chính xác và nhanh chóng
* Việc thống kê quản lí điểm của sinh viên sẽ nhanh hơn
* Rút ngắn thời gian quản lý và giảm số lượng nhân viên quản lí

1. Nhược điểm

* Giao diện chưa được bắt mắt
* Hệ thống chưa đủ sức chứa cho lượng sinh viên lớn
* Kiến thức vẫn chưa có nhiều nên tính năng chưa được hoàn thiện

1. Hướng phát triển

* Cần tối ưu việc phần quyền sử dụng
* Hệ thống sẽ nâng cấp thêm nhiều tính năng để đáp ứng như cầu người sử dụng
* Tối ưu hệ thống và xây dựng hệ thống có thể thực hiện online tiện cho việc tiếp cận giảng viên, sinh viên

1. Kết luận

* Qua 3 chương phân tích và tìm hiểu hệ thống thông tin về bài toán quản lí điểm sinh viên . Nhóm chúng em đã đưa ra những phân tích về hệ thống ở cả hai phương diện chức năng và dữ liệu. Về chức năng nhóm đã đưa ra được các sơ đồ chức năng của hệ thống. Về dữ liệu đã đưa ra cơ sở dữ liệu. Tuy nhiên do còn hạn chế về nghiên cứu cũng như tiếp cận thực tế, chính vì vậy việc phân tích không tránh khỏi việc sai sót. Việc phân chia các bộ phận còn bất cập. Chức năng còn hạn chế. Thiết kế hệ thống còn chưa đẹp và sơ sài
* Trong suốt quá trình học tập và làm việc nhóm đã học và tiếp thu được rất nhiều kiến thức mới từ thầy. Nhóm rất mong nhận được những ý kiến đóng góp của thầy để có thể hoàn thành tốt hơn ở những đề tài tiếp theo. Chúng em cảm ơn thầy

1. **Tài liệu tham khảo**

* <https://freetuts.net/chuong-trinh-quan-ly-sinh-vien-java-2748.html>
* <https://viettuts.vn/bai-tap-java/bai-tap-quan-ly-sinh-vien-trong-java>
* <https://www.youtube.com/watch?v=2Ktm950lVP4&list=PLyxSzL3F748401hWFgJ8gKMnN6MM8QQ7F&index=102&ab_channel=TITV>
* <https://viettuts.vn/bai-tap-java/bai-tap-quan-ly-sinh-vien-trong-java-swing>
* <https://codelearn.io/sharing/tao-ung-dung-quan-ly-sinh-vien-voi-jdbc-nhu-the-nao>
* <https://gokisoft.com/phan-mem-quan-ly-sinh-vien-mysql--java-chuong-trinh-quan-ly-sinh-vien-mysql--java-lap-trinh-java.html>
* <https://tailieu.vn/doc/de-tai-he-thong-quan-ly-diem-sinh-vien-he-dan-su-nguyen-thi-hoa-1657027.html>
* <https://sharecode.vn/?fbclid=IwAR3yhBrDbbHM2DJs7UwSVb86aYn0zIUsErEFHOrGp1zt9aeieIFBTondIPA>