**LỜI CẢM ƠN**

Sau quá trình học tập và rèn luyện tại khoa Công nghệ thông tin trường Đại học Trần Đại Nghĩa chúng em đã được trang bị các kiến thức cơ bản, các kỹ năng thực tế để có thể hoàn thành đồ án môn học của mình.

Chúng em xin gửi lời cảm ơn chân thành đến thầy Đặng Thế Hùng đã quan tâm hướng dẫn truyền đạt học những kiến thức và kinh nghiệm cho chúng em trong suốt thời gian học học tập bộ môn Thực tập Lập trình Hướng đối tượng

Trong quá trình làm đồ án môn không tránh khỏi được những sai sót, chúng em mong nhận được sự góp ý của quý thầy và các bạn để được hoàn thiện hơn.

|  |  |
| --- | --- |
|  | TP.Hồ Chí Minh, tháng 1 năm 2017 |

**MỤC LỤC**

[CHƯƠNG 1: KHÁI QUÁT VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG 4](#_Toc471778262)

[1. Vị trí và tầm quan trọng của môn học 4](#_Toc471778263)

[2. Các tính chất cơ bản 4](#_Toc471778264)

[CHƯƠNG 2: TỔNG QUAN HỆ THỐNG QUẢN LÍ ĐIỂM SINH VIÊN 6](#_Toc471778265)

[1. Tổng quan hệ thống. 6](#_Toc471778266)

[1.1. Hệ thống tổng quát 6](#_Toc471778267)

[1.2. Nhiệm vụ hệ thống. 6](#_Toc471778268)

[1.3. Chức năng hệ thống. 6](#_Toc471778269)

[1.4. Các biểu mẫu sử dụng trong hệ thống. 7](#_Toc471778270)

[2. Mô hình tiến trình nghiệp vụ. 9](#_Toc471778271)

[2.1. Các kí hiệu sử dụng trong mô hình 9](#_Toc471778272)

[2.2. Vẽ sơ đồ ERD hệ thống. 10](#_Toc471778273)

[CHƯƠNG 3: CÁC CHỨC NĂNG CHÍNH CỦA PHẦN MỀM QUẢN LÝ ĐIỂM SINH VIÊN 11](#_Toc471778274)

[1. Quản lý người dùng 11](#_Toc471778275)

[2. Quản lý thông tin sinh viên 12](#_Toc471778276)

[3. Quản lý điểm sinh viên 13](#_Toc471778277)

[4. Báo cáo danh sách sinh viên 14](#_Toc471778278)

[5. Bảng điểm sinh viên 14](#_Toc471778279)

[CHƯƠNG 4: SƠ ĐỒ QUAN HỆ VÀ CÁC LỚP GIẢI QUYẾT CHỨC NĂNG 15](#_Toc471778280)

[1. Sơ đồ quan hệ 15](#_Toc471778281)

[2. Các lớp giải quyết chức năng 16](#_Toc471778282)

[2.1. Lớp quản lý giảng viên 16](#_Toc471778283)

[2.2. Lớp quản lý thông tin sinh viên 20](#_Toc471778284)

[2.3. Lớp quản lý lớp 25](#_Toc471778285)

[2.4. Lớp quản lý khoa 29](#_Toc471778286)

[2.5. Lớp quản lý thông tin sinh viên 32](#_Toc471778287)

[2.6. Lớp quản lý người dùng 40](#_Toc471778288)

[CHƯƠNG 5: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM 44](#_Toc471778289)

**LỜI NÓI ĐẦU**

Trong những năm gần đây, các chương trình quản lí không còn xa lạ với chúng ta. Quản lý đã xâm nhập vào các lĩnh vực như giáo dục, kinh doanh,….. đặc biệt trong giáo dục và đào tạo, công tác quản lí điểm sinh viên trong quá trình học tập là rất quan trọng, phức tạp và đòi hỏi độ chính xác cao. Đảm bảo mỗi sinh viên khi ra trường hoặc đang trong quá trình rèn luyện có thể theo dõi chính xác được quá trình học tập của mình. Trong suốt quá trình học tập, điểm quá trình của sinh viên là cơ sở để xét duyệt học bổng, điều kiện tốt nghiệp và phân loại xếp hạng sinh viên. Do đó công tác quản lí điểm sinh viên đòi hỏi phải có sự thống nhất và chính xác tuyệt đối giúp cho bộ phận quản lí sửa đổi, điều chỉnh hợp lí, kịp thời về phương pháp quản lí và kế hoạch cho đào tạo.

Hệ thống quản lí điểm sinh viên này sẽ giúp công tác quản lí điểm Sinh viên Trường Đại học Trần Đại Nghĩa giải quyết được những khó khăn trên và tăng tính hiệu quả cho công tác quản lí.

# CHƯƠNG 1: KHÁI QUÁT VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

1. **Vị trí và tầm quan trọng của môn học**

Lập trình hướng đối tượng (gọi tắt là OOP, từ chữ Anh ngữ object-oriented programming), hay còn gọi là lập trình định hướng đối tượng, là kĩ thuật lập trình hỗ trợ công nghệ đối tượng. OOP được xem là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn. Ngoài ra, nhiều người còn cho rằng OOP dễ tiếp thu hơn cho những người mới học về lập trình hơn là các phương pháp trước đó.

Một cách giản lược, đây là khái niệm và là một nỗ lực nhằm giảm nhẹ các thao tác viết mã cho người lập trình, cho phép họ tạo ra các ứng dụng mà các yếu tố bên ngoài có thể tương tác với các chương trình đó giống như là tương tác với các đối tượng vật lý.

Những đối tượng trong một ngôn ngữ OOP là các kết hợp giữa mã và dữ liệu mà chúng được nhìn nhận như là một đơn vị duy nhất. Mỗi đối tượng có một tên riêng biệt và tất cả các tham chiếu đến đối tượng đó được tiến hành qua tên của nó. Như vậy, mỗi đối tượng có khả năng nhận vào các thông báo, xử lý dữ liệu (bên trong của nó), và gửi ra hay trả lời đến các đối tượng khác hay đến môi trường.

1. **Các tính chất cơ bản**

**Đối tượng** (object): Các dữ liệu và chỉ thị được kết hợp vào một đơn vị đầy đủ tạo nên một đối tượng. Đơn vị này tương đương với một chương trình con và vì thế các đối tượng sẽ được chia thành hai bộ phận chính: phần các phương thức (method) và phần các thuộc tính (attribute / Properties). Trong thực tế, các phương thức của đối tượng là các hàm và các thuộc tính của nó là các biến, các tham số hay hằng nội tại của một đối tượng (hay nói cách khác tập hợp các dữ liệu nội tại tạo thành thuộc tính của đối tượng). Các phương thức là phương tiện để sử dụng một đối tượng trong khi các thuộc tính sẽ mô tả đối tượng có những tính chất gì.

Các phương thức và các thuộc tính thường gắn chặt với thực tế các đặc tính và sử dụng của một đối tượng.

Trong thực tế, các đối tượng thường được trừu tượng hóa qua việc định nghĩa của các lớp (class).

Tập hợp các giá trị hiện có của các thuộc tính tạo nên trạng thái của một đối tượng.

Mỗi phương thức hay mỗi dữ liệu nội tại cùng với các tính chất được định nghĩa (bởi người lập trình) được xem là một đặc tính riêng của đối tượng. Nếu không có gì lầm lẫn thì tập hợp các đặc tính này gọi chung là đặc tính của đối tượng.

Lập trình hướng đối tượng là một phương pháp lập trình có 4 tính chất chính sau:

* **Tính trừu tượng** (abstraction): Đây là khả năng của chương trình bỏ qua hay không chú ý đến một số khía cạnh của thông tin mà nó đang trực tiếp làm việc lên, nghĩa là nó có khả năng tập trung vào những cốt lõi cần thiết. Mỗi đối tượng phục vụ như là một "động tử" có thể hoàn tất các công việc một cách nội bộ, báo cáo, thay đổi trạng thái của nó và liên lạc với các đối tượng khác mà không cần cho biết làm cách nào đối tượng tiến hành được các thao tác. Tính chất này thường được gọi là sự trừu tượng của dữ liệu.

Tính trừu tượng còn thể hiện qua việc một đối tượng ban đầu có thể có một số đặc điểm chung cho nhiều đối tượng khác như là sự mở rộng của nó nhưng bản thân đối tượng ban đầu này có thể không có các biện pháp thi hành. Tính trừu tượng này thường được xác định trong khái niệm gọi là lớp trừu tượng hay lớp cơ sở trừu tượng.

* **Tính đóng gói** (encapsulation) và che giấu thông tin (information hiding): Tính chất này không cho phép người sử dụng các đối tượng thay đổi trạng thái nội tại của một đối tượng. Chỉ có các phương thức nội tại của đối tượng cho phép thay đổi trạng thái của nó. Việc cho phép môi trường bên ngoài tác động lên các dữ liệu nội tại của một đối tượng theo cách nào là hoàn toàn tùy thuộc vào người viết mã. Đây là tính chất đảm bảo sự toàn vẹn của đối tượng.
* **Tính đa hình** (polymorphism): Thể hiện thông qua việc gửi các thông điệp (message). Việc gửi các thông điệp này có thể so sánh như việc gọi các hàm bên trong của một đối tượng. Các phương thức dùng trả lời cho một thông điệp sẽ tùy theo đối tượng mà thông điệp đó được gửi tới sẽ có phản ứng khác nhau. Người lập trình có thể định nghĩa một đặc tính (chẳng hạn thông qua tên của các phương thức) cho một loạt các đối tượng gần nhau nhưng khi thi hành thì dùng cùng một tên gọi mà sự thi hành của mỗi đối tượng sẽ tự động xảy ra tương ứng theo đặc tính của từng đối tượng mà không bị nhầm lẫn.

Ví dụ khi định nghĩa hai đối tượng "hinh\_vuong" và "hinh\_tron" thì có một phương thức chung là "chu\_vi". Khi gọi phương thức này thì nếu đối tượng là "hinh\_vuong" nó sẽ tính theo công thức khác với khi đối tượng là "hinh\_tron".

* **Tính kế thừa** (inheritance): Đặc tính này cho phép một đối tượng có thể có sẵn các đặc tính mà đối tượng khác đã có thông qua kế thừa. Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại. Tuy nhiên, không phải ngôn ngữ định hướng đối tượng nào cũng có tính chất này.

# CHƯƠNG 2: TỔNG QUAN HỆ THỐNG QUẢN LÍ ĐIỂM SINH VIÊN

**1. Tổng quan hệ thống.**

**1.1. Hệ thống tổng quát**

Để xây dựng một hệ thống thông tin hoàn chỉnh chúng ta cần trải qua 6 giai đoạn:

* Khảo sát
* Phân tích hệ thống
* Thiết kế hệ thống
* Kiểm thử
* Cài đặt
* Vận hành và bảo trì

Giai đoạn khảo sát là một trong 6 giai đoạn rất quan trọng. Và trong quá trình học tập tôi thấy rằng hệ thống quản lý điểm đang còn nhiều bất cập và khó khăn. Nhập và tính toán điểm bằng Excel, Sinh viên chưa thể tự động xem và theo dõi điểm của mình trong suốt quá trình theo học tại trường. Đặc biệt gặp khó khăn trong quá trình tổng kết điểm và thành tích học tập của sinh viên vào dịp cuối năm học.

Với yêu cầu cấp thiết như vậy nhóm chúng tôi sẽ khắc phục những khó khăn và xây dựng một hệ thống quản lí điểm sinh viên nhằm hỗ trợ cho quá trình quản lí được tốt hơn, tránh sai xót, thuận tiện trong quá trình quản lí điểm.

**1.2. Nhiệm vụ hệ thống.**

Theo dõi toàn bộ quá trình học tập của sinh viên trong suốt quá trình học ở trong trường từ khi nhập học đến khi ra trường, đảm bảo không sai sót trong quá trình nhập điểm và in điểm. Hệ thống này giúp cho phòng đào tạo theo dõi, đánh giá kết quả học tập của sinh viên một cách nhanh nhất và chính xác nhất. Nó sẽ làm giảm đáng kể công sức, giấy tờ, sổ sách lưu trư.

Ngoài việc cập nhật điểm của từng sinh viên, hệ thống cũng có thể tra cứu một số thông tin cần thiết về sinh viên như ngày sinh, quê quán, chỗ ở hiện tại…..

Hệ thống lưu trữ, xử lý kết quả học tập của sinh viên theo quy chế của trường. Kiểm soát thống kê việc khen thưởng, xét học bổng, xét lưu ban, thi lại, học lại của sinh viên một cách nhanh chóng và chính xác. Thông qua công tác quản lý mà có thể có hồ sơ cụ thể về kết quả học tập rèn luyện của sinh viên khi ra trường.

**1.3. Chức năng hệ thống.**

Hệ thống quản lý điểm sinh viên bao gồm 3 bộ phận, hoạt động có mối liên hệ chặt chẽ nhau.

* Bộ phận quản lý đào tạo: Cập nhật môn học, xây dựng chương trình đạo tạo toàn khóa.
* Bộ phận quản lý sinh viên: Bộ phận này có nhiệm vụ quản lý hồ sơ sinh viên trong suốt quá trình học tập tại trường, đảm bảo các công việc có liên quan đến sinh viên như khi có khóa mới nhập học thì bộ phận này tiến hành cập nhật sinh viên khóa mới, danh sách ngành học, khóa học cho sinh viên, thực hiện việc cập nhật để chuyển lớp, tách lớp, dừng học, thôi học. Đồng thời bộ phận này cũng làm nhiệm vụ lưu hồ sơ của nhưng sinh viên đã ra trường vào máy tính. Bộ phận này được sử dụng một PC có cài đặt hệ chương trình gọi là Hồ sơ sinh viên (HSSV) trợ giúp các việc như cập nhật, thêm, xóa, sửa thông tin của sinh viên.
* Ban khảo thí: Cập nhật điểm từng môn khi kết thúc môn học, học kỳ khi cập nhật có kiểm tra môn học đó. Cập nhật điểm rèn luyện của sinh viên. Tính điểm trung bình học tập của mỗi kỳ, của mỗi năm học. Sau đó bộ phận này sẽ thống kê toàn bộ kết quả học tập của sinh viên theo từng học kỳ, toàn khóa học. Đồng thời thống kê báo cáo danh sách sinh viên thi lại, học lại, thôi học, học bổng… dựa trên các quy định của phòng đào tạo và điểm trung bình chung của sinh viên. Chuyển kết quả cho cán bộ khác và sinh viên. Bộ phận này cũng có một PC trên đó cài đặt hệ chương trình Nhập điểm trợ giúp việc cập nhật điểm và in phiếu điểm cho từng lớp và cho từng sinh viên

**1.4. Các biểu mẫu sử dụng trong hệ thống.**

Danh sách sinh viên từng lớp:

**DANH SÁCH LỚP**

**Trường đại học trần đại nghĩa** Khoa:…….

Phòng đào tạo Lớp:………

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Mã SV | Họ và Tên | Ngày sinh | Giới tính | Quê quán | Chỗ ở hiện tại | Ghi chú |
| 1 | …….. | …….. | …….. | …….. | …….. | …….. | …….. |
| 2 | …….. | …….. | …….. | …….. | …….. | …….. | …….. |
| 3 | …….. | …….. | …….. | …….. | …….. | …….. | …….. |

Danh sách môn học

**DANH MÔN HỌC**

**Trường đại học trần đại nghĩa** Khoa:

Phòng đào tạo Lớp:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| STT | Mã môn học | Tên môn học | Bộ môn | Đvhp | Ngành đào tạo | Chuyên ngàn dt | Học kỳ |
| 1 | …….. | …….. | …….. | …….. | …….. | …….. | …….. |
| 2 | …….. | …….. | …….. | …….. | …….. | …….. | …….. |
| 3 | …….. | …….. | …….. | …….. | …….. | …….. | …….. |

Bảng điểm cá nhân

Bộ quốc phòng CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

**Trường đại học trần đại nghĩa** Độc lập-Tự do- Hạnh phúc

**BẢNG ĐIỂM CÁ NHÂN**

Họ tên: Lớp:

Ngày sinh: Khóa:

Giới tính: Chuyên ngành:

Quê quán: Hệ đào tạo:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Mã môn học | Tên môn học | Số ĐVHP | Điểm tổng kết |
| 1 | …….. | …….. | …….. | …….. |
| 2 | …….. | …….. | …….. | …….. |
| 3 | …….. | …….. | …….. | …….. |

Danh sách học bổng

**DANH SÁCH HỌC BỔNG**

**Trường đại học trần đại nghĩa** Khoa:

Phòng đào tạo Năm học:

Hệ đt:

Chuyên ngành:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | MSSV | Họ và tên | Lớp | ĐTBC | Ghi chú |
| 1 | …….. | …….. | …….. | …….. | …….. |
| 2 | …….. | …….. | …….. | …….. | …….. |
| 3 | …….. | …….. | …….. | …….. | …….. |

**2. Mô hình tiến trình nghiệp vụ.**

**2.1. Các kí hiệu sử dụng trong mô hình**

Dùng kí hiệu hình chữ nhật để mô tả chức năng của hệ thống, tên của chức năng ghi bên trong. Ví dụ như: bộ phận quản lý điểm, bộ phận quản lý sinh viên.



Dùng mũi tên chỉ hướng của luồng dữ liệu ghi trên mũi tên.



Dùng kí hiệu hình elip để miêu tả tác nhân bên ngoài của hệ thống, tên tác nhân ghi bên trong.



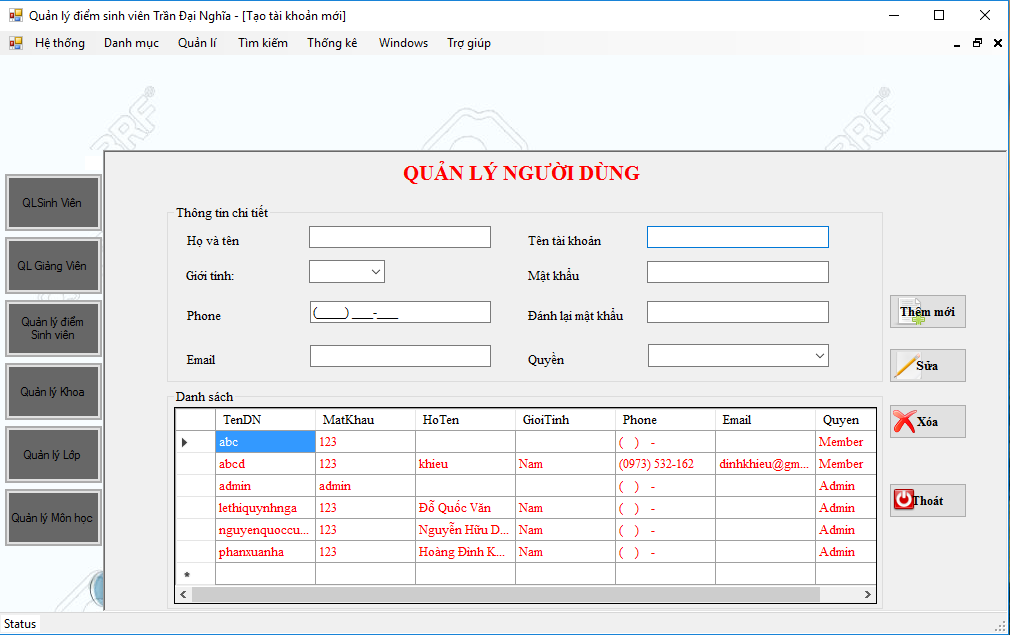
**2.2. Vẽ sơ đồ ERD hệ thống.**



# CHƯƠNG 3: CÁC CHỨC NĂNG CHÍNH CỦA PHẦN MỀM QUẢN LÝ ĐIỂM SINH VIÊN

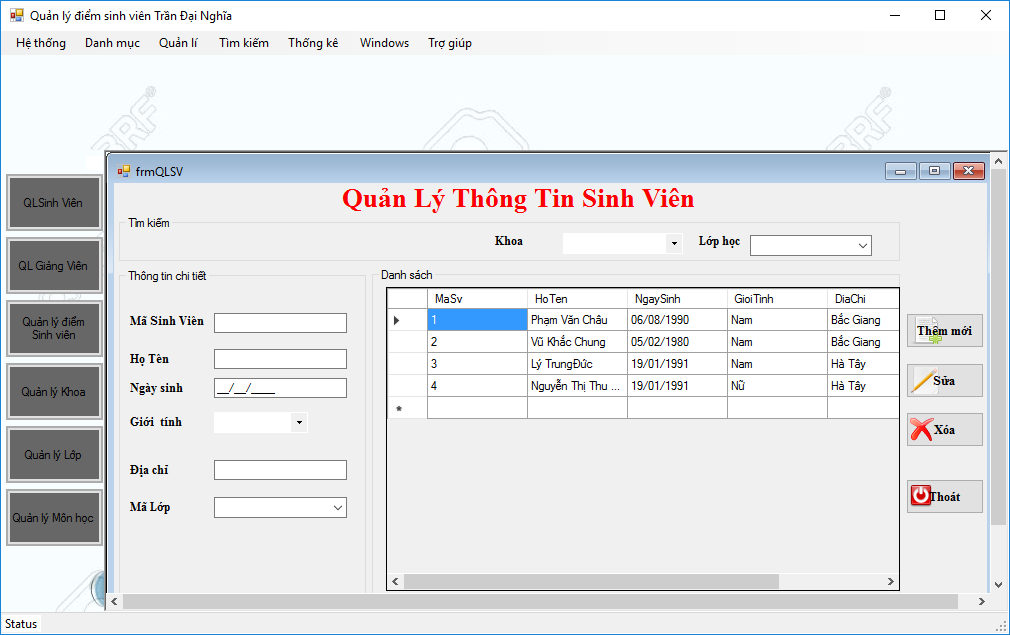
**1. Quản lý người dùng**

Bất kì một hệ thống nào cũng cần phân chia vai trò sử dụng và quản lí. Và đối với hệ thống Quản Lý điểm sinh viên này cũng vậy. Hệ thống chia làm quyền sử dụng là Admin và Member.



Admin có quyền truy cập và sửa chữa cơ sở dữ liệu khi cần thiết. Member chỉ được quyền truy cập và xem thông tin được hiển thị trên hệ thống mà không có quyền chỉnh sửa cơ sở dữ liệu. Form đăng nhập giúp độ bảo mật về sơ sở dữ liệu được tăng lên, tránh và hạn chế tối đa các trường hợp bị đánh cắp và sửa chữa làm sai dữ liệu gốc ban đầu.

**2. Quản lý thông tin sinh viên**

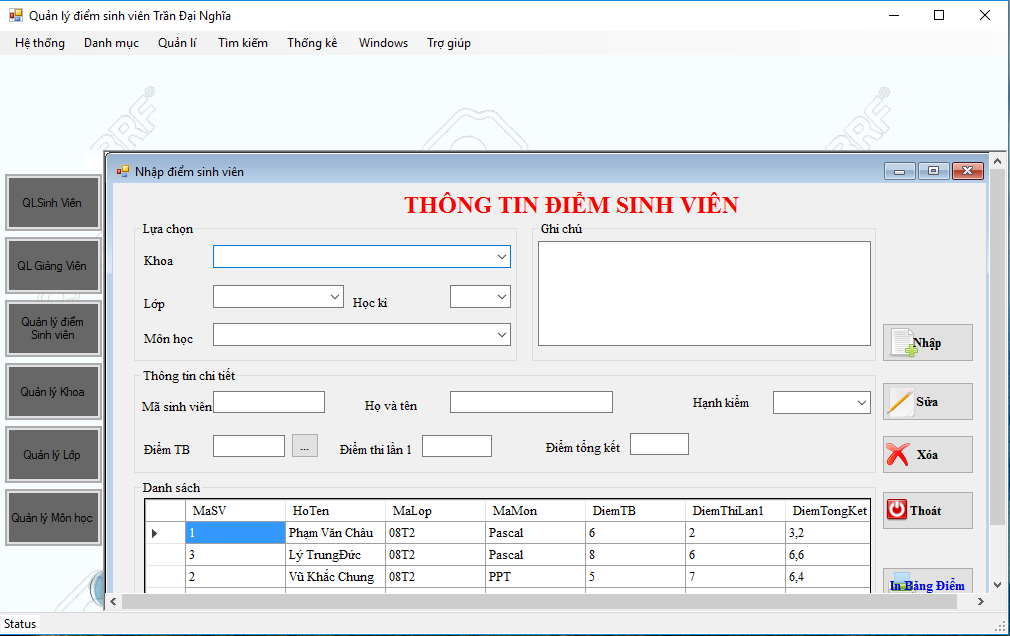


Form nhập thông tin sinh viên chỉ sử dụng bảng dữ liệu sinh viên

Chức năng của giao diên: cập nhật thông tin cần thiết của một sinh viên khi mới vào trường, trong giao diện có các nút lệnh

* Thêm: thêm một bảng ghi mới, trong đó có đầy đủ thông tin sinh viên
* Sửa: khi có sự sai sót trong quá trình điền thông tin nút lệnh này cho phép người nhập có thể sửa bất kỳ một thông tin nào
* Xóa: xóa toàn bộ bảng ghi đã chọn
* Thoát: thoát khỏi form nhập thông tin

**3. Quản lý điểm sinh viên**



Form cập nhật bảng điểm sinh viên sử dụng bảng DM khoa, DM lớp, DM môn học và bảng sinh viên. Trong DM khoa thì lấy thuộc tính mã khoa, trong DM lớp thì lấy thuộc tính mã lớp, trong DM môn học thì lấy thuộc tính tên môn. Còn trong bảng sinh viên thì lấy thuộc tính mã SV và tên SV.

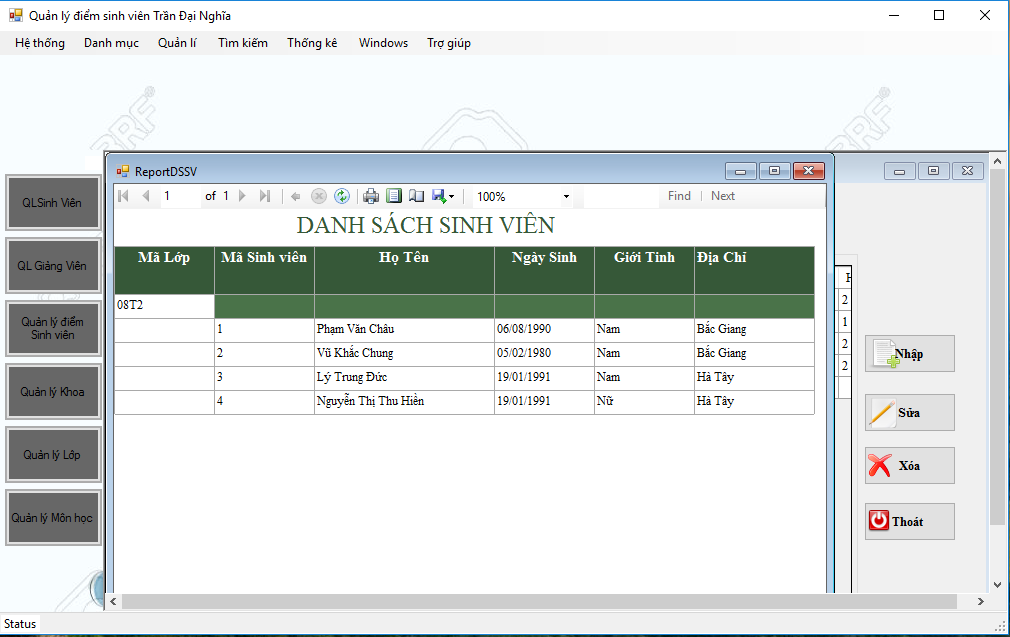
Chức năng của giao diên: Cập nhật điểm của từng môn học trong suốt quá trình học của một sinh viên. Trong form có cac nút lệnh:

* Thêm: thêm điểm của một môn học cho sinh viên của 1 lớp nào đó
* Sửa: sửa điểm của từng môn học
* Xóa: xóa bảng ghi điểm của môn học có mã môn được chọn
* Thoát: thoát khỏi form cập nhật bảng điểm cá nhân

Điểm trung bình được tính theo công thức : (điểm lần 1+ (điểm lần 2 \*2))/3.

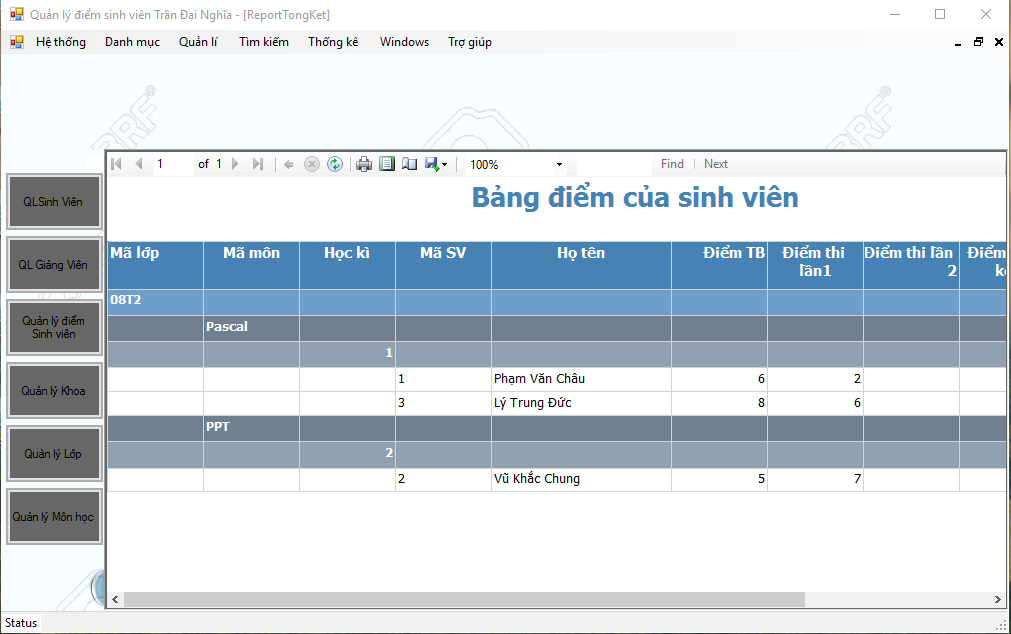
Hạnh kiểm sẽ do bộ phận quản lí đánh giá và xếp loại dựa vào các tiêu chí đánh giá riêng của BCH đoàn ở một hệ thống khác.

**4. Báo cáo danh sách sinh viên**



Báo cáo danh sách sinh viên của từng lớp. Trong báo cáo các bảng sinh viên, DM lớp. Cụ thể là những thuộc tính Hoten\_SV, Ngaysinh, Diachi….

**5. Bảng điểm sinh viên**



Báo cáo cụ thể điểm của sinh viên qua từng môn học. Trong báo cáo các bảng Kết Qủa, Môn Học, Sinh Viên. Cụ thể gồm có các thuộc tính MaLop, MaMon, HocKi, MaSV, HoTen, DiemTB….

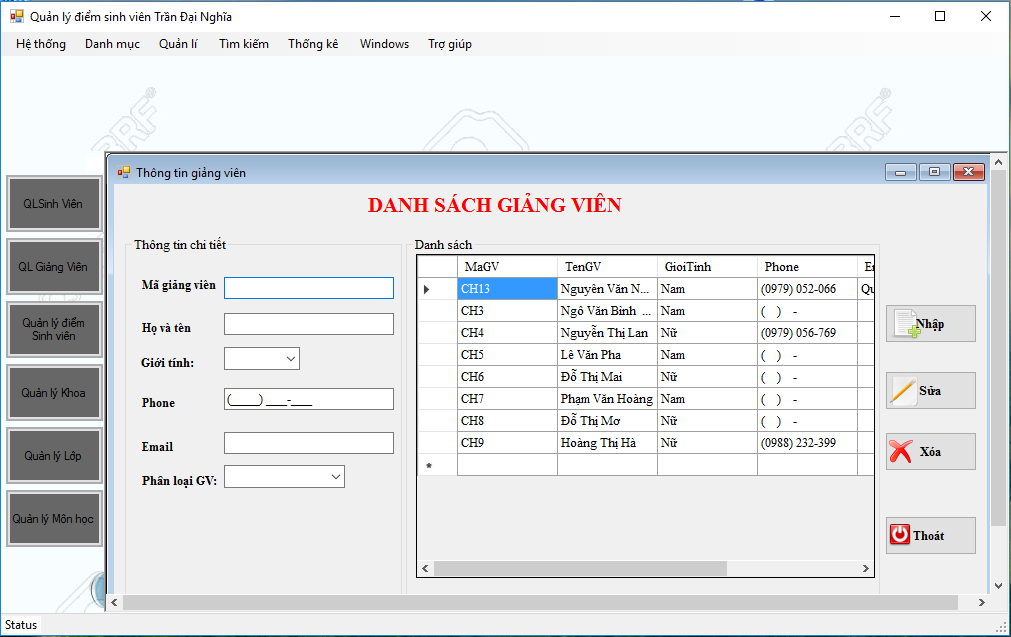
# CHƯƠNG 4: SƠ ĐỒ QUAN HỆ VÀ CÁC LỚP GIẢI QUYẾT CHỨC NĂNG

**1. Sơ đồ quan hệ**



**2. Các lớp giải quyết chức năng**

**2.1. Lớp quản lý giảng viên**



namespace Quản\_lý\_điểm\_sinh\_vien\_CNTT

{

public partial class frmGiangVien : Form

{

private CommonConnect cc = new CommonConnect();

SqlConnection conn = null;

public frmGiangVien()

{

InitializeComponent();

}

private void frmGiangVien\_Load(object sender, EventArgs e)

{

conn = cc.Connected();

if (conn.State == ConnectionState.Open) ;

string sql = "select \* from [Quanlydiem].[dbo].[tblGIANG\_VIEN]";

SqlCommand commandsql = new SqlCommand(sql, conn);

SqlDataAdapter com = new SqlDataAdapter(commandsql);

DataTable table = new DataTable();

com.Fill(table);

dgrDSGV.DataSource = table;

}

// Chức năng nhập

private void button1\_Click(object sender, EventArgs e)

{

//Kiem tra trung ten MaSV

string select2 = "Select \* From tblGIANG\_VIEN where MaGV='" + txtMaGV.Text + "'";

SqlCommand cmd2 = new SqlCommand(select2, conn);

SqlDataReader reader2;

reader2 = cmd2.ExecuteReader();

errorProvider1.Clear();

if (txtMaGV.Text == "")

{

errorProvider1.SetError(txtMaGV, "Mã giảng viên không để trống!");

}

else if (reader2.Read())

{

MessageBox.Show("Bạn đã nhập trùng mã giảng viên", "Thông báo !", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaGV.Focus();

cmd2.Dispose();

reader2.Dispose();

}

else

{

// Trả tài nguyên

cmd2.Dispose();

reader2.Dispose();

// Thực hiện truy vấn

string insert = "Insert Into tblGIANG\_VIEN(MaGV,TenGV,Gioitinh,Phone,Email,PhanloaiGV)" +"Values('" + txtMaGV.Text + "',N'" + txtHoTen.Text + "',N'" + cboGioiTinh.Text + "','" + mskPhone.Text + "','" + txtEmail.Text + "',N'" + cboPhanloai.Text + "')";

SqlCommand cmd = new SqlCommand(insert, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

}

// Trả tài nguyên

cmd2.Dispose();

reader2.Dispose();

//Fill du lieu

FillDataGridView\_SV();

}

private void txtHoTen\_TextChanged(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

public void FillDataGridView\_SV()

{

// Thực hiện truy vấn

string select = "Select \* From tblGIANG\_VIEN ";

SqlCommand cmd = new SqlCommand(select, conn);

// Tạo đối tượng DataSet

DataSet ds = new DataSet();

// Tạo đối tượng điều hợp

SqlDataAdapter adapter = new SqlDataAdapter();

adapter.SelectCommand = cmd;

// Fill dữ liệu từ adapter vào DataSet

adapter.Fill(ds, "SINHVIEN");

// Đưa ra DataGridView

dgrDSGV.DataSource = ds;

dgrDSGV.DataMember = "SINHVIEN";

cmd.Dispose();

}

private void dgrDSGV\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

// Chức năng Xóa

private void button3\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Bạn có chắc chắn muốn xóa ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Thuc hien xoa du lieu

SqlCommand cmd = new SqlCommand("delete from tblGIANG\_VIEN where MaGV='" + txtMaGV.Text + "'", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa dữ liệu thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Load lai du lieu

FillDataGridView\_SV();

}

}

// Chức năng cập nhật

private void button2\_Click(object sender, EventArgs e)

{

errorProvider1.Clear();

if (txtMaGV.Text == "")

errorProvider1.SetError(txtMaGV, "Mã giảng viên không để trống!");

else

{

// Thực hiện truy vấn

string update = "Update tblGIANG\_VIEN Set TenGV=N'" + txtHoTen.Text + "',GioiTinh=N'" +

cboGioiTinh.Text + "',Phone='" + mskPhone.Text + "',Email='" +

txtEmail.Text + "',PhanLoaiGV=N'" + cboPhanloai.Text + "' where MaGV='" + txtMaGV.Text + "'";

SqlCommand cmd = new SqlCommand(update, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật dữ liệu thành công", "Thông báo!");

//Load lai du lieu

FillDataGridView\_SV();

// Trả tài nguyên

cmd.Dispose();

}

}

private void dgrDSGV\_CellContentClick\_1(object sender, DataGridViewCellEventArgs e)

{

txtMaGV.Text = dgrDSGV.CurrentRow.Cells[0].Value.ToString();

txtHoTen.Text = dgrDSGV.CurrentRow.Cells[1].Value.ToString();

cboGioiTinh.Text = dgrDSGV.CurrentRow.Cells[2].Value.ToString();

mskPhone.Text = dgrDSGV.CurrentRow.Cells[3].Value.ToString();

txtEmail.Text = dgrDSGV.CurrentRow.Cells[4].Value.ToString();

cboPhanloai.Text = dgrDSGV.CurrentRow.Cells[5].Value.ToString();

}

private void button4\_Click(object sender, EventArgs e)

{

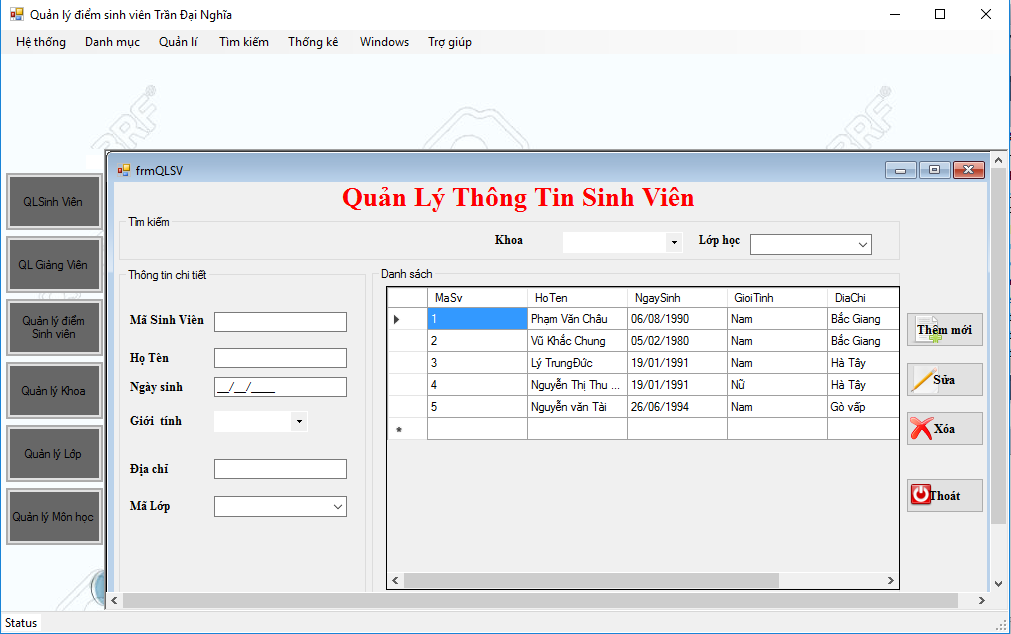
Application.Exit();

}

}

}

**2.2. Lớp quản lý thông tin sinh viên**



namespace Quản\_lý\_điểm\_sinh\_vien\_CNTT

{

public partial class frmQLSV : Form

{

private CommonConnect cc = new CommonConnect();

SqlConnection conn = null;

public frmQLSV()

{

InitializeComponent();

}

private void frmQLSV\_Load(object sender, EventArgs e)

{

conn=cc.Connected();

if (conn.State == ConnectionState.Open) ;

//Add du lieu vao cboKhoa

string select = "Select MaKhoa from tblKHOA ";

SqlCommand cmd = new SqlCommand(select, conn);

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

cboKhoahoc.Items.Add(reader.GetString(0));

}

reader.Dispose();

cmd.Dispose();

//Add du lieu vao MaLop

string selects = "Select MaLop from tblLOP";

SqlCommand cmd1 = new SqlCommand(selects, conn);

SqlDataReader reader1 = cmd1.ExecuteReader();

while (reader1.Read())

{

cboMalop.Items.Add(reader1.GetString(0));

}

reader1.Dispose();

cmd1.Dispose();

//Load lai du lieu

FillDataGridView\_SV();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

txtMaSV.Text = dgrDSSV.CurrentRow.Cells[0].Value.ToString();

txtHoTen.Text = dgrDSSV.CurrentRow.Cells[1].Value.ToString();

mskNgaySinh.Text = dgrDSSV.CurrentRow.Cells[2].Value.ToString();

cboGioiTinh.Text = dgrDSSV.CurrentRow.Cells[3].Value.ToString();

txtDiaChi.Text = dgrDSSV.CurrentRow.Cells[4].Value.ToString();

cboMalop.Text = dgrDSSV.CurrentRow.Cells[5].Value.ToString();

}

// Chức năng thêm

private void btnThemmoi\_Click(object sender, EventArgs e)

{

//Kiem tra trung ten MaSV

string select2 = "Select \* From tblSINH\_VIEN where Masv='" + txtMaSV.Text + "'";

SqlCommand cmd2 = new SqlCommand(select2, conn);

SqlDataReader reader2;

reader2 = cmd2.ExecuteReader();

errorProvider1.Clear();

if (txtMaSV.Text == "")

{

errorProvider1.SetError(txtMaSV, "Mã sinh viên không để trống!");

}

else if (cboMalop.Text == "")

{

errorProvider1.SetError(cboMalop, "Mã lớp không để trống!");

}

else if (reader2.Read())

{

MessageBox.Show("Bạn đã nhập trùng mã sinh viên ", "Thông báo !",MessageBoxButtons.OK,MessageBoxIcon.Warning);

txtMaSV.Focus();

cmd2.Dispose();

reader2.Dispose();

}

else

{

// Trả tài nguyên

cmd2.Dispose();

reader2.Dispose();

// Thực hiện truy vấn

string insert = "Insert Into tblSINH\_VIEN(MaSV,HoTen,Ngaysinh,Gioitinh,DiaChi,Malop)" +"Values('" + txtMaSV.Text + "',N'" + txtHoTen.Text + "','" +mskNgaySinh.Text + "',N'" + cboGioiTinh.Text + "',N'" + txtDiaChi.Text + "','" + cboMalop.Text + "')";

SqlCommand cmd = new SqlCommand(insert, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

}

// Trả tài nguyên

cmd2.Dispose();

reader2.Dispose();

//Load lai du lieu

FillDataGridView\_SV();

}

public void FillDataGridView\_SV()

{

// Thực hiện truy vấn

string select = "Select \* From tblSINH\_VIEN ";

SqlCommand cmd = new SqlCommand(select, conn);

// Tạo đối tượng DataSet

DataSet ds = new DataSet();

// Tạo đối tượng điều hợp

SqlDataAdapter adapter = new SqlDataAdapter();

adapter.SelectCommand = cmd;

// Fill dữ liệu từ adapter vào DataSet

adapter.Fill(ds, "SINHVIEN");

// Đưa ra DataGridView

dgrDSSV.DataSource = ds;

dgrDSSV.DataMember = "SINHVIEN";

cmd.Dispose();

}

// Chức năng xóa

private void btnXoa\_Click(object sender, EventArgs e)

{

//Kiem tra

string select1 = "Select MaSv from tblKET\_QUA where MaSv='" + txtMaSV.Text + "' ";

SqlCommand cmd1 = new SqlCommand(select1, conn);

SqlDataReader reader1 = cmd1.ExecuteReader();

if (reader1.Read())

{

{

MessageBox.Show("Bạn phải xóa Mã Sinh viên " + txtMaSV.Text + "từ bảng Kết quả học tập", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else if (MessageBox.Show("Bạn có chắc chắn muốn xóa ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

reader1.Dispose();

cmd1.Dispose();

// Thuc hien xoa du lieu

SqlCommand cmd = new SqlCommand("delete from tblSINH\_VIEN where MaSv='" + txtMaSV.Text + "'", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa dữ liệu thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Load lai du lieu

FillDataGridView\_SV();

}

reader1.Dispose();

cmd1.Dispose();

}

private void btnThoat\_Click(object sender, EventArgs e)

{

Application.Exit();

}

// Chức năng sửa

private void btnSua\_Click(object sender, EventArgs e)

{

errorProvider1.Clear();

if (txtMaSV.Text == "")

errorProvider1.SetError(txtMaSV, "Mã sinh viên không để trống!");

else if (cboMalop.Text == "")

errorProvider1.SetError(cboMalop, "Mã lớp không để trống!");

else

{

// Thực hiện truy vấn

string update = "Update tblSINH\_VIEN Set HoTen=N'" + txtHoTen.Text + "',NgaySinh='" +

mskNgaySinh.Text + "',GioiTinh=N'" + cboGioiTinh.Text + "',DiaChi=N'" + txtDiaChi.Text + "',MaLop='" + cboMalop.Text + "'where MaSV='" + txtMaSV.Text + "'";

SqlCommand cmd = new SqlCommand(update, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật dữ liệu thành công", "Thông báo!");

//Load lai du lieu

FillDataGridView\_SV();

// Trả tài nguyên

cmd.Dispose();

}

}

private void cboKhoahoc\_SelectedIndexChanged(object sender, EventArgs e)

{

cboLop.Items.Clear();

string select = "Select MaLop from tblLOP where MaKhoa='" + cboKhoahoc.Text + "'";

SqlCommand cmd = new SqlCommand(select, conn);

SqlDataReader reader = cmd.ExecuteReader();

//Add vao cboLop

while (reader.Read())

{

cboLop.Items.Add(reader.GetString(0));

}

//Tra tai nguyen

reader.Dispose();

cmd.Dispose();

}

private void cboLop\_SelectedIndexChanged(object sender, EventArgs e)

{

// Thực hiện truy vấn

string select = "Select \* From tblSINH\_VIEN where MaLop='"+ cboLop.Text +"'";

SqlCommand cmd = new SqlCommand(select, conn);

// Tạo đối tượng DataSet

DataSet ds = new DataSet();

// Tạo đối tượng điều hợp

SqlDataAdapter adapter = new SqlDataAdapter();

adapter.SelectCommand = cmd;

// Fill dữ liệu từ adapter vào DataSet

adapter.Fill(ds, "SINHVIEN");

// Đưa ra DataGridView

dgrDSSV.DataSource = ds;

dgrDSSV.DataMember = "SINHVIEN";

cmd.Dispose();

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

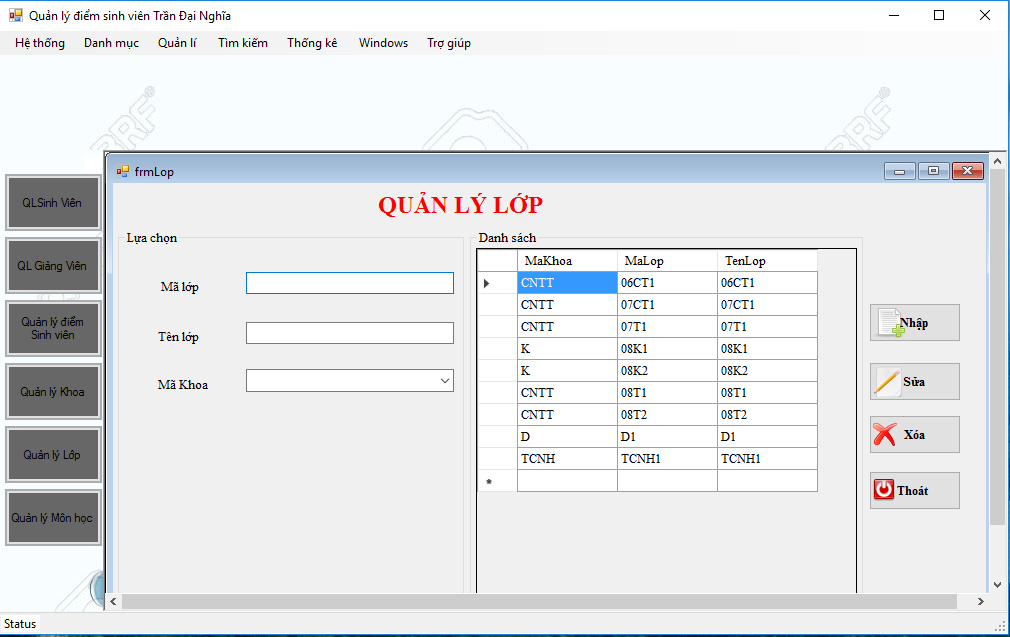
}

private void groupBox3\_Enter(object sender, EventArgs e)

{

} } }

**2.3. Lớp quản lý lớp**



namespace Quản\_lý\_điểm\_sinh\_vien\_CNTT

{

public partial class frmLop : Form

{

private CommonConnect cc = new CommonConnect();

SqlConnection conn = null;

public frmLop()

{

InitializeComponent();

}

private void txtMaKhoa\_TextChanged(object sender, EventArgs e)

{

}

private void frmLop\_Load(object sender, EventArgs e)

{

conn = cc.Connected();

if (conn.State == ConnectionState.Open) ;

string sql = " select \* from [Quanlydiem].[dbo].[tblLOP]";

SqlCommand commandsql = new SqlCommand(sql, conn);

SqlDataAdapter com = new SqlDataAdapter(commandsql);

DataTable table = new DataTable();

com.Fill(table);

dgrLop.DataSource = table;

//Add du lieu vao cboKhoaHoc

string select = "Select MaKhoa from tblKHOA ";

SqlCommand cmd = new SqlCommand(select, conn);

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

cboKhoa.Items.Add(reader.GetString(0));

}

reader.Dispose();

cmd.Dispose();

//Load lai du lieu

FillDataGridView\_Lop();

}

private void button1\_Click(object sender, EventArgs e)

{

string select1 = "Select MaKhoa from tblKHOA where MaKhoa='" + cboKhoa.Text + "' ";

SqlCommand cmd1 = new SqlCommand(select1, conn);

SqlDataReader reader1 = cmd1.ExecuteReader();

errorProvider1.Clear();

if (txtMaLop.Text == "")

{

errorProvider1.SetError(txtMaLop, "!");

txtMaLop.Focus();

}

else if (txtMaLop.Text == dgrLop.CurrentRow.Cells[1].Value.ToString())

{

errorProvider1.SetError(txtMaLop, "!");

MessageBox.Show("Mã lớp đã tồn tại !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaLop.Focus();

}

else if (reader1.Read())

{

reader1.Dispose();

cmd1.Dispose();

// Thực hiện truy vấn

string insert = "Insert Into tblLOP(MaLop,TenLop,Makhoa)" + "Values('" + txtMaLop.Text + "',N'" + txtTenlop.Text + "','" + cboKhoa.Text + "')";

SqlCommand cmd = new SqlCommand(insert, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Nhập thông tin thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Fill du lieu vao Database

FillDataGridView\_Lop();

}

else

{

MessageBox.Show("Nhập mã khoa không chính xác !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

cboKhoa.Focus();

reader1.Dispose();

cmd1.Dispose();

}

reader1.Dispose();

cmd1.Dispose();

}

public void FillDataGridView\_Lop()

{

// Thực hiện truy vấn

string select = "Select \* From tblLop ";

SqlCommand cmd = new SqlCommand(select, conn);

// Tạo đối tượng DataSet

DataSet ds = new DataSet();

// Tạo đối tượng điều hợp

SqlDataAdapter adapter = new SqlDataAdapter();

adapter.SelectCommand = cmd;

// Fill dữ liệu từ adapter vào DataSet

adapter.Fill(ds, "SINHVIEN");

// Đưa ra DataGridView

dgrLop.DataSource = ds;

dgrLop.DataMember = "SINHVIEN";

cmd.Dispose();

}

private void dgrLop\_CellContentClick(object sender, DataGridViewCellEventArgs {

cboKhoa.Text = dgrLop.CurrentRow.Cells[0].Value.ToString();

txtMaLop.Text = dgrLop.CurrentRow.Cells[1].Value.ToString();

txtTenlop.Text = dgrLop.CurrentRow.Cells[2].Value.ToString();

}

private void button3\_Click(object sender, EventArgs e)

{

//Kiem tra

string select1 = "Select MaLop from tblSINH\_VIEN where MaLop='" + txtMaLop.Text + "' ";

SqlCommand cmd1 = new SqlCommand(select1, conn);

SqlDataReader reader1 = cmd1.ExecuteReader();

if (reader1.Read())

{

{

MessageBox.Show("Bạn phải xóa Mã Lớp "+ txtMaLop.Text+"từ bảng Sinh Viên", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else if (MessageBox.Show("Bạn có chắc chắn muốn xóa ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Thuc hien xoa du lieu

cmd1.Dispose();

reader1.Dispose();

SqlCommand cmd = new SqlCommand("delete from tblLOP where MaLop='" + txtMaLop.Text + "'", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa dữ liệu thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Load lai du lieu

FillDataGridView\_Lop();

}

cmd1.Dispose();

reader1.Dispose();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

// Thực hiện truy vấn

string update = "Update tblLOP Set MaKhoa='" + cboKhoa.Text + "',MaLop='" + txtMaLop.Text + "',TenLop='" + txtTenlop.Text + "' where MaLop='" + txtMaLop.Text + "' ";

SqlCommand cmd = new SqlCommand(update, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật dữ liệu thành công", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Load lai du lieu

FillDataGridView\_Lop();

// Trả tài nguyên

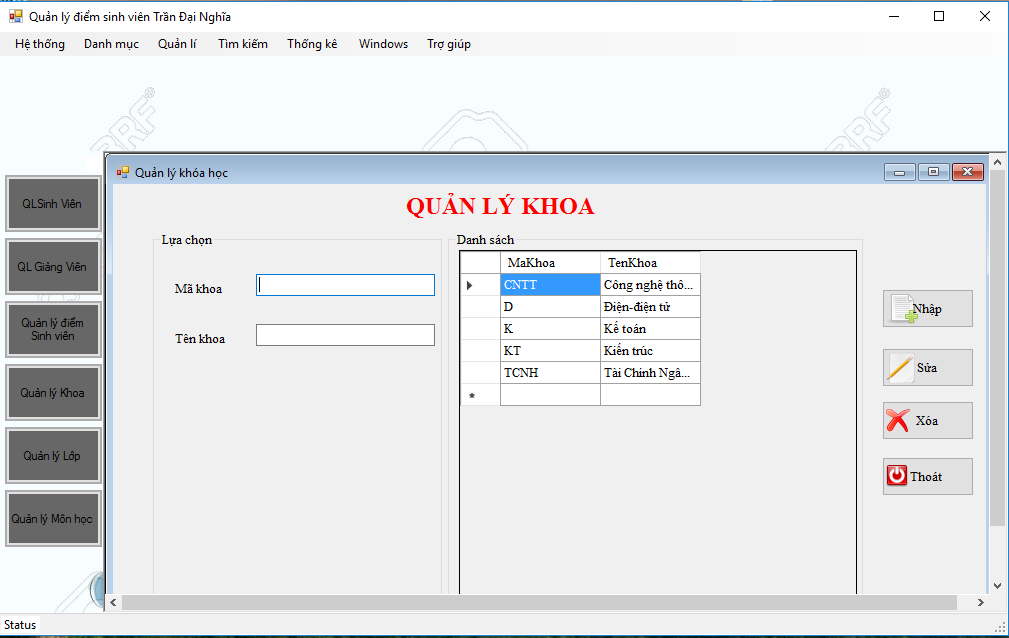
cmd.Dispose();

}

}

}

**2.4. Lớp quản lý khoa**



namespace Quản\_lý\_điểm\_sinh\_vien\_CNTT

{

public partial class frmKhoa : Form

{

private CommonConnect cc = new CommonConnect();

SqlConnection conn = null;

public frmKhoa()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string select1 = "Select MaKhoa from tblKHOA where MaKhoa='" + txtKhoa.Text + "' ";

SqlCommand cmd1 = new SqlCommand(select1, conn);

SqlDataReader reader1 = cmd1.ExecuteReader();

errorProvider1.Clear();

if (txtKhoa.Text == "")

{

errorProvider1.SetError(txtKhoa, "Khóa học không để trống!");

txtTenKhoa.Focus();

}

else if (reader1.Read())

{

{

MessageBox.Show("Thông tin đã tồn tại !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtKhoa.Focus();

}

//Tra tai nguyen

reader1.Dispose();

cmd1.Dispose();

}

else

{

reader1.Dispose();

cmd1.Dispose();

// Thực hiện truy vấn

string insert = "Insert Into tblKHOA(MaKhoa,TenKhoa)" +

"Values('" + txtKhoa.Text + "',N'" + txtTenKhoa.Text + "')";

SqlCommand cmd = new SqlCommand(insert, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Nhập thông tin thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Fill du lieu vao Database

FillDataGridView\_Khoa();

}

}

public void FillDataGridView\_Khoa()

{

// Thực hiện truy vấn

string select = "Select \* From tblKHOA ";

SqlCommand cmd = new SqlCommand(select, conn);

// Tạo đối tượng DataSet

DataSet ds = new DataSet();

// Tạo đối tượng điều hợp

SqlDataAdapter adapter = new SqlDataAdapter();

adapter.SelectCommand = cmd;

// Fill dữ liệu từ adapter vào DataSet

adapter.Fill(ds, "SINHVIEN");

// Đưa ra DataGridView

dgrKhoa.DataSource = ds;

dgrKhoa.DataMember = "SINHVIEN";

cmd.Dispose();

}

private void frmLop\_Khoa\_Load(object sender, EventArgs e)

{

conn = cc.Connected();

if (conn.State == ConnectionState.Open) ;

string sqlkhoa = "select \* from [Quanlydiem].[dbo].[tblKHOA]";

SqlCommand comand = new SqlCommand(sqlkhoa, conn);

SqlDataAdapter sqlcom = new SqlDataAdapter(comand);

DataTable table = new DataTable();

sqlcom.Fill(table);

dgrKhoa.DataSource = table;

}

private void button3\_Click(object sender, EventArgs e)

{

//Kiem tra

string select1 = "Select MaKhoa from tblLOP where MaKhoa='" + txtKhoa.Text + "' ";

SqlCommand cmd1 = new SqlCommand(select1, conn);

SqlDataReader reader1 = cmd1.ExecuteReader();

if (reader1.Read())

{

{

MessageBox.Show("Bạn phải xóa Mã Khoa "+ txtKhoa.Text +"từ bảng Lớp","Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else if (MessageBox.Show("Bạn có chắc chắn muốn xóa ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Thuc hien xoa du lieu

reader1.Dispose();

cmd1.Dispose();

SqlCommand cmd = new SqlCommand("delete from tblKHOA where MaKhoa='" + txtKhoa.Text + "'", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa dữ liệu thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Load lai du lieu

FillDataGridView\_Khoa();

}

reader1.Dispose();

cmd1.Dispose();

}

private void button2\_Click(object sender, EventArgs e)

{

// Thực hiện truy vấn

string update = "Update tblKHOA Set TenKhoa=N'" + txtTenKhoa.Text + "' where MaKhoa='" + txtKhoa.Text + "' ";

SqlCommand cmd = new SqlCommand(update, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật dữ liệu thành công", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Load lai du lieu

FillDataGridView\_Khoa();

// Trả tài nguyên

cmd.Dispose();

}

private void dgrKhoa\_CellContentClick(object sender, DataGridViewCellEventArgs

{

txtKhoa.Text = dgrKhoa.CurrentRow.Cells[0].Value.ToString();

txtTenKhoa.Text = dgrKhoa.CurrentRow.Cells[1].Value.ToString();

}

private void button4\_Click(object sender, EventArgs e)

{

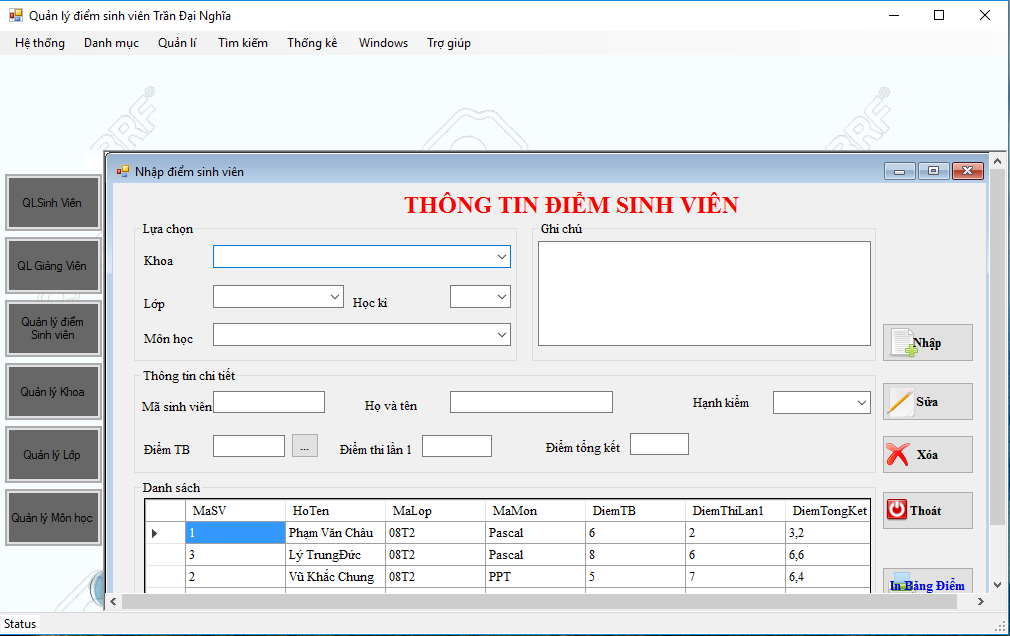
this.Close();

}

}

}

**2.5. Lớp quản lý thông tin sinh viên**



namespace Quản\_lý\_điểm\_sinh\_vien\_CNTT

{

public partial class frmQLDiem : Form

{

private CommonConnect cc = new CommonConnect();

SqlConnection conn = null;

public frmQLDiem()

{

InitializeComponent();

}

private void groupBox2\_Enter(object sender, EventArgs e)

{

}

private void groupBox4\_Enter(object sender, EventArgs e)

{

}

private void cboKhoaHoc\_SelectedIndexChanged(object sender, EventArgs e)

{

cboLop.Items.Clear();

cboLop.Text = "";

string select = "Select MaLop from tblLOP where MaKhoa='" + cboKhoaHoc.Text + "'";

SqlCommand cmd = new SqlCommand(select, conn);

SqlDataReader reader = cmd.ExecuteReader();

//Add vao cboLop

while (reader.Read())

{

cboLop.Items.Add(reader.GetString(0));

}

//Tra tai nguyen

reader.Dispose();

cmd.Dispose();

}

private void frmQLDiem\_Load(object sender, EventArgs e)

{

conn = cc.Connected();

if (conn.State == ConnectionState.Open) ;

string sql = " select \* from [Quanlydiem].[dbo].[tblKET\_QUA2]";

SqlCommand commandsql = new SqlCommand(sql, conn);

SqlDataAdapter com = new SqlDataAdapter(commandsql);

DataTable table = new DataTable();

com.Fill(table);

dgrDiem.DataSource = table;

//Add du lieu vao cboKhoaHoc

string select = "Select MaKhoa from tblKHOA ";

SqlCommand cmd = new SqlCommand(select, conn);

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

cboKhoaHoc.Items.Add(reader.GetString(0));

}

reader.Dispose();

cmd.Dispose();

//Load lai du lieu

FillDataGridView\_Diem();

}

private void cboNamHoc\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void cboHocKi\_SelectedIndexChanged(object sender, EventArgs e)

{

cboMonHoc.Items.Clear();

string select = "Select MaMon from tblMON where HocKi='" + cboHocKi.Text + "'";

SqlCommand cmd = new SqlCommand(select, conn);

SqlDataReader reader = cmd.ExecuteReader();

//Add vao cboLop

while (reader.Read())

{

cboMonHoc.Items.Add(reader.GetString(0));

}

//Tra tai nguyen

reader.Dispose();

cmd.Dispose();

}

private void button1\_Click(object sender, EventArgs e)

{

//Kiem tra trung ten MonHoc va MASV

string select1 = "Select MaSV from tblSINH\_VIEN where MaSV=N'" + txtMaSV.Text + "' and Hoten=N'"+ txtHoTen.Text +"' ";

SqlCommand cmd1 = new SqlCommand(select1, conn);

SqlDataReader reader1 = cmd1.ExecuteReader();

errorProvider1.Clear();

if (txtMaSV.Text == "")

{

errorProvider1.SetError(txtMaSV,"Mã sinh viên không để trống!");

txtMaSV.Focus();

}

else if (txtMaSV.Text == dgrDiem.CurrentRow.Cells[0].Value.ToString() && cboMonHoc.Text == dgrDiem.CurrentRow.Cells[3].Value.ToString())

{

{

MessageBox.Show("Sinh viên này đã được nhập điểm môn: " + cboMonHoc.Text,"Thông báo", MessageBoxButtons.OK,MessageBoxIcon.Warning);

txtMaSV.Focus();

}

}

else if (cboLop.Text == "")

{

errorProvider1.SetError(cboLop, "Mã lớp không để trống!");

cboLop.Focus();

}

else if (cboHocKi.Text == "")

{

errorProvider1.SetError(cboHocKi, "Học kỳ không để trống!");

cboHocKi.Focus();

}

else if (cboMonHoc.Text == "")

{

errorProvider1.SetError(cboMonHoc, "Mã môn không để trống!");

cboMonHoc.Focus();

}

else if (reader1.Read())

{

//Tra tai nguyen

reader1.Dispose();

cmd1.Dispose();

// Thực hiện truy vấn

string insert = "Insert Into tblKET\_QUA(MaSV,HoTen,MaLop,MaMon,DiemThiLan1,DiemTB,DiemTongket,HanhKiem,HocKi,GhiChu)" +

"Values('" + txtMaSV.Text + "',N'" + txtHoTen.Text + "','" + cboLop.Text + "',N'" + cboMonHoc.Text + "','" + txtDiemThi1.Text + "','" +

txtDiemTB.Text + "','" + txtDiemTK.Text + "',N'" + cboHanhKiem.Text + "',N'" +

cboHocKi.Text + "',N'" + txtGhiChu.Text + "')";

SqlCommand cmd = new SqlCommand(insert, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Nhập thông tin thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

}

else

{

{

MessageBox.Show("Nhập mã sinh viên không chính xác !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtMaSV.Focus();

}

//Tra tai nguyen

reader1.Dispose();

cmd1.Dispose();

}

//Tra tai nguyen

reader1.Dispose();

cmd1.Dispose();

//Load lai du lieu

FillDataGridView\_Diem();

}

public void FillDataGridView\_Diem()

{

// Thực hiện truy vấn

string select = "Select \* From tblKET\_QUA ";

SqlCommand cmd = new SqlCommand(select, conn);

// Tạo đối tượng DataSet

DataSet ds = new DataSet();

// Tạo đối tượng điều hợp

SqlDataAdapter adapter = new SqlDataAdapter();

adapter.SelectCommand = cmd;

// Fill dữ liệu từ adapter vào DataSet

adapter.Fill(ds, "SINHVIEN");

// Đưa ra DataGridView

dgrDiem.DataSource = ds;

dgrDiem.DataMember = "SINHVIEN";

cmd.Dispose();

}

private void cboLop\_SelectedIndexChanged(object sender, EventArgs e)

{

//// Thực hiện truy vấn

//string select = "Select \* From tblKET\_QUA Where MaLop='"+cboLop .Text +"'";

//SqlCommand cmd = new SqlCommand(select, conn);

//// Tạo đối tượng DataSet

//DataSet ds = new DataSet();

//// Tạo đối tượng điều hợp

//SqlDataAdapter adapter = new SqlDataAdapter();

//adapter.SelectCommand = cmd;

//// Fill dữ liệu từ adapter vào DataSet

//adapter.Fill(ds, "SINHVIEN");

//// Đưa ra DataGridView

//dgrDiem.DataSource = ds;

//dgrDiem.DataMember = "SINHVIEN";

//cmd.Dispose();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Bạn có chắc chắn muốn xóa ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Thuc hien xoa du lieu

string delete = "delete from tblKET\_QUA where MaSV='" + txtMaSV.Text + "' and MaMon='" + cboMonHoc.Text + "' ";

SqlCommand cmd = new SqlCommand(delete, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa dữ liệu thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Load lai du lieu

FillDataGridView\_Diem();

}

}

private void dgrDiem\_CellContentClick(object sender, DataGridViewCellEventArgs

{

txtMaSV.Text = dgrDiem.CurrentRow.Cells[0].Value.ToString();

txtHoTen.Text = dgrDiem.CurrentRow.Cells[1].Value.ToString();

cboLop.Text = dgrDiem.CurrentRow.Cells[2].Value.ToString();

cboMonHoc.Text = dgrDiem.CurrentRow.Cells[3].Value.ToString();

txtDiemTB.Text = dgrDiem.CurrentRow.Cells[4].Value.ToString();

txtDiemThi1.Text = dgrDiem.CurrentRow.Cells[5].Value.ToString();

txtDiemTK.Text = dgrDiem.CurrentRow.Cells[6].Value.ToString();

cboHanhKiem.Text = dgrDiem.CurrentRow.Cells[7].Value.ToString();

cboHocKi.Text = dgrDiem.CurrentRow.Cells[8].Value.ToString();

txtGhiChu.Text = dgrDiem.CurrentRow.Cells[9].Value.ToString();

}

private void button2\_Click(object sender, EventArgs e)

{

//Kiem tra trung ten MonHoc va MASV

//string select1 = "Select MaSV,TenMon from tblKET\_QUA where TenMon=N'" + cboMonHoc.Text + "' ";

//SqlCommand cmd1 = new SqlCommand(select1, conn);

//SqlDataReader reader1 = cmd1.ExecuteReader();

//errorProvider1.Clear();

if (txtMaSV.Text == "")

{

errorProvider1.SetError(txtMaSV, "Mã sinh viên không để trống!");

}

//else if (reader1.Read())

//{

// {

// MessageBox.Show("Sinh viên này đã được nhập điểm môn: " + cboMonHoc.Text, "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

// txtMaSV.Focus();

// }

////Tra tai nguyen

//reader1.Dispose();

//cmd1.Dispose();

//}

else

{

////Tra tai nguyen

//reader1.Dispose();

//cmd1.Dispose();

// Thực hiện truy vấn

string update = "Update tblKET\_QUA Set HoTen=N'" + txtHoTen.Text + "',MaMon=N'" +cboMonHoc.Text + "',MaLop='" +

cboLop.Text + "',DiemThiLan1='" + txtDiemThi1.Text + "',DiemTB='" +txtDiemTB.Text + "' ,DiemTongket='" +

txtDiemTK.Text + "',HanhKiem=N'" + cboHanhKiem.Text + "',HocKi=N'" + cboHocKi.Text + "',GhiChu=N'" +

txtGhiChu.Text + "' where MaSV='" + txtMaSV.Text + "' and MaMon=N'" + cboMonHoc.Text + "'";

SqlCommand cmd = new SqlCommand(update, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật dữ liệu thành công", "Thông báo!", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Load lai du lieu

FillDataGridView\_Diem();

// Trả tài nguyên

cmd.Dispose();

}

}

private void button4\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void txtDiemTK\_KeyPress(object sender, KeyPressEventArgs e)

{

}

private void txtDiemThi2\_TextChanged(object sender, EventArgs e)

{

}

private void txtDiemThi1\_TextChanged(object sender, EventArgs e )

{

double DIEMTHI, DIEMTB, DIEMTK;

if (txtDiemThi1.Text == "")

{

this.txtDiemThi1.Text = "0";

DIEMTB = double.Parse(this.txtDiemTB.Text);

//Tính điểm TK

DIEMTK = (0.3 \* DIEMTB + 0.7 \* 0);

this.txtDiemTK.Text = Convert.ToString(DIEMTK);

}

else if (txtDiemTB.Text == "")

{

this.txtDiemTB.Text = "0";

DIEMTHI = double.Parse(this.txtDiemThi1.Text);

//Tính điểm TK

DIEMTK = (0.3 \* 0 + 0.7 \* DIEMTHI);

this.txtDiemTK.Text = Convert.ToString(DIEMTK);

}

else

{

DIEMTHI = double.Parse(this.txtDiemThi1.Text);

DIEMTB = double.Parse(this.txtDiemTB.Text);

//Tính điểm TK

DIEMTK = (0.3 \* DIEMTB + 0.7 \* DIEMTHI);

this.txtDiemTK.Text = Convert.ToString(DIEMTK);

}

DIEMTK = double.Parse(this.txtDiemTK.Text);

if (DIEMTK <= 4.5)

{

this.txtGhiChu.Text = "Thi lại";

}

else

{

this.txtGhiChu.Text = "";

}

}

private void txtDiemTB\_TextChanged(object sender, EventArgs e)

{

double DIEMTHI, DIEMTB, DIEMTK;

if (txtDiemTB.Text == "")

{

this.txtDiemTB.Text = "0";

DIEMTHI = double.Parse(this.txtDiemThi1.Text);

//Tính điểm TK

DIEMTK = (0.3 \* 0 + 0.7 \*DIEMTHI);

this.txtDiemTK.Text = Convert.ToString(DIEMTK);

}

else if (txtDiemThi1.Text == "")

{

this.txtDiemThi1.Text = "0";

DIEMTB = double.Parse(this.txtDiemTB.Text);

//Tính điểm TK

DIEMTK = (0.3 \* DIEMTB + 0.7 \* 0);

this.txtDiemTK.Text = Convert.ToString(DIEMTK);

}

else

{

DIEMTHI = double.Parse(this.txtDiemThi1.Text);

DIEMTB = double.Parse(this.txtDiemTB.Text);

//Tính điểm TK

DIEMTK = (0.3 \* DIEMTB + 0.7 \* DIEMTHI);

this.txtDiemTK.Text = Convert.ToString(DIEMTK);

}

DIEMTK = double.Parse(this.txtDiemTK.Text);

if ((DIEMTK <= 4.5))

{

this.txtGhiChu.Text = "Thi lại";

}

else

{

this.txtGhiChu.Text = "";

}

}

private void cboMonHoc\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void btnXuat\_Click(object sender, EventArgs e)

{

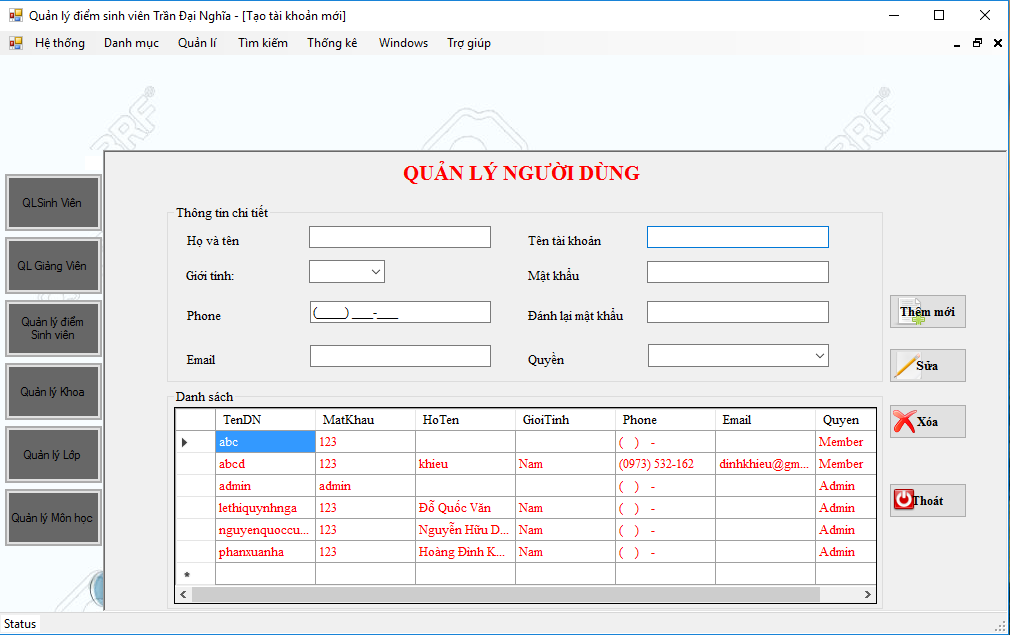
Thaotac.Export2Excel(dgrDiem);

}

}

}

**2.6. Lớp quản lý người dùng**



namespace Quản\_lý\_điểm\_sinh\_vien\_CNTT

{

public partial class frmQuanLyNguoiDung : Form

{

private CommonConnect cc = new CommonConnect();

SqlConnection conn = null;

public frmQuanLyNguoiDung()

{

InitializeComponent();

}

private void frmQuanLyNguoiDung\_Load(object sender, EventArgs e)

{

}

private void btnThemmoi\_Click(object sender, EventArgs e)

{

//Kiem tra trung TenDN

string select2 = "Select \* From tblLOGIN where TenDN='" + txtTaikhoan.Text + "'";

SqlCommand cmd2 = new SqlCommand(select2, conn);

SqlDataReader reader2;

reader2 = cmd2.ExecuteReader();

errorProvider1.Clear();

if (txtTaikhoan.Text == "")

{

errorProvider1.SetError(txtTaikhoan, "Tên tài khoản không để trống !");

txtTaikhoan.Focus();

}

else if (txtMK.Text == "")

{

errorProvider1.SetError(txtMK, "Bạn chưa nhập mật khẩu !");

txtMK.Focus();

}

else if (txtConfimMk.Text == "")

{

errorProvider1.SetError(txtConfimMk, "Bạn chưa nhập lại mật khẩu !");

txtConfimMk.Focus();

}

else if (txtConfimMk.Text != txtMK.Text)

MessageBox.Show("Bạn nhập lại mật khẩu không trùng khớp", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

else if (reader2.Read())

{

MessageBox.Show("Tài khoản " + txtTaikhoan.Text + " đã tồn tại", "Thông báo !", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtTaikhoan.Focus();

cmd2.Dispose();

reader2.Dispose();

}

else

{

// Trả tài nguyên

cmd2.Dispose();

reader2.Dispose();

// Thực hiện truy vấn

string insert = "Insert Into tblLOGIN(TenDN,MatKhau,HoTen,Gioitinh,Phone,Email,Quyen)" + "Values('" + txtTaikhoan.Text + "','" + txtMK.Text + "',N'" + txtHoTen.Text + "',N'" + cboGioiTinh.Text + "','" + mskPhone.Text + "','" + txtEmail.Text + "',N'" + cboQuyen.Text + "')";

SqlCommand cmd = new SqlCommand(insert, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Thêm mới thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

}

// Trả tài nguyên

cmd2.Dispose();

reader2.Dispose();

//Fill du lieu

FillDataGridView\_Login();

}

private void frmQuanLyNguoiDung\_Load\_1(object sender, EventArgs e)

{

this.tblLOGINTableAdapter.Fill(this.quanlydiemDataSet25.tblLOGIN);

conn = cc.Connected();

if (conn.State == ConnectionState.Open) ;

//Fill du lieu

FillDataGridView\_Login();

}

public void FillDataGridView\_Login()

{

// Thực hiện truy vấn

string select = "Select \* From tblLOGIN ";

SqlCommand cmd = new SqlCommand(select, conn);

// Tạo đối tượng DataSet

DataSet ds = new DataSet();

// Tạo đối tượng điều hợp

SqlDataAdapter adapter = new SqlDataAdapter();

adapter.SelectCommand = cmd;

// Fill dữ liệu từ adapter vào DataSet

adapter.Fill(ds, "SINHVIEN");

// Đưa ra DataGridView

dgrLogin.DataSource = ds;

dgrLogin.DataMember = "SINHVIEN";

cmd.Dispose();

}

private void dgrLogin\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

txtTaikhoan.Text = dgrLogin.CurrentRow.Cells[0].Value.ToString();

txtMK.Text = dgrLogin.CurrentRow.Cells[1].Value.ToString();

txtHoTen.Text = dgrLogin.CurrentRow.Cells[2].Value.ToString();

cboGioiTinh.Text = dgrLogin.CurrentRow.Cells[3].Value.ToString();

mskPhone.Text = dgrLogin.CurrentRow.Cells[4].Value.ToString();

txtEmail.Text = dgrLogin.CurrentRow.Cells[5].Value.ToString();

cboQuyen.Text = dgrLogin.CurrentRow.Cells[6].Value.ToString();

}

private void btnXoa\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Bạn có chắc chắn muốn xóa ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

{

// Thuc hien xoa du lieu

SqlCommand cmd = new SqlCommand("delete from tblLOGIN where TenDN='" + txtTaikhoan.Text + "'", conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Xóa dữ liệu thành công", "Thông báo!");

// Trả tài nguyên

cmd.Dispose();

//Fill du lieu

FillDataGridView\_Login();

}

}

private void btnSua\_Click(object sender, EventArgs e)

{

errorProvider1.Clear();

if (txtTaikhoan.Text == "")

errorProvider1.SetError(txtTaikhoan, "Tên tài khoản không để trống!");

else

{

// Thực hiện truy vấn

string update = "Update tblLOGIN Set MatKhau=N'" + txtMK.Text + "',HoTen=N'" + txtHoTen.Text + "',GioiTinh=N'" + cboGioiTinh.Text + "',Phone='" + mskPhone.Text + "',Email='" + txtEmail.Text + "',Quyen=N'" + cboQuyen.Text + "' where TenDN='" + txtTaikhoan.Text + "'";

SqlCommand cmd = new SqlCommand(update, conn);

cmd.ExecuteNonQuery();

MessageBox.Show("Cập nhật dữ liệu thành công", "Thông báo!");

//Load lai du lieu

FillDataGridView\_Login();

// Trả tài nguyên

cmd.Dispose();

}

}

private void btnThoat\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

# CHƯƠNG 5: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

Sau đây nhóm sẽ hướng dẫn sử dụng phần mềm Quản lý điểm sinh viên hệ dân sự như sau:

Trong phần mềm hiện tại đặt hai quyền hoạt động : admin và member. Khi người sử dụng đăng nhập hệ thống thì bắt buộc phải chọn đúng quyền của mình. Mỗi một member sẽ được hệ thống cung câp một user và password riêng:



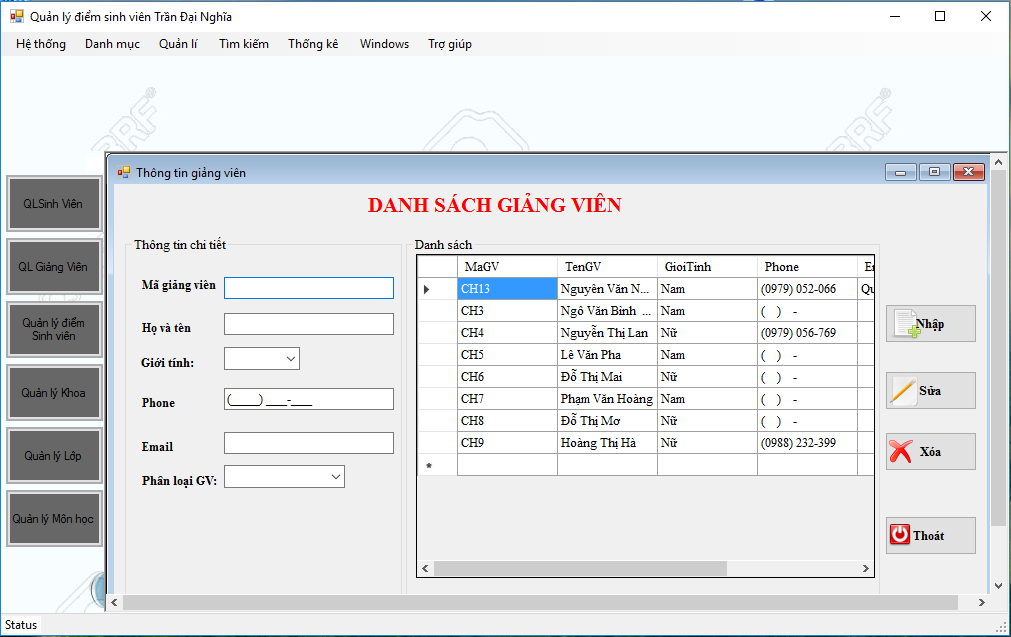
Sau khi đăng nhập thành công, giao diện chính hệ thống sẽ xuất hiện:

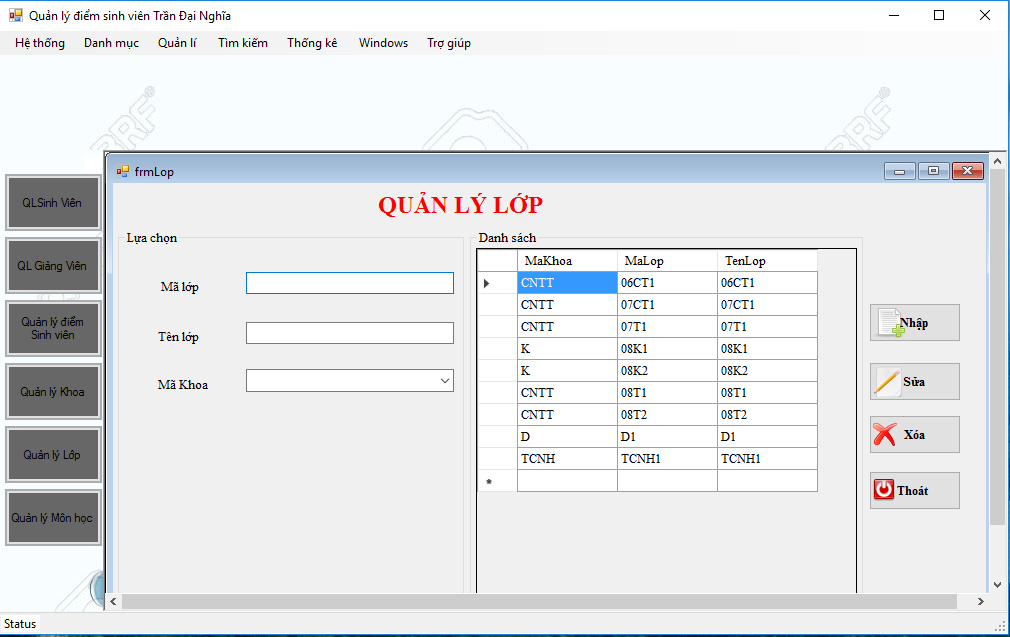


Các cửa sổ chức năng hệ thống được mở ra:

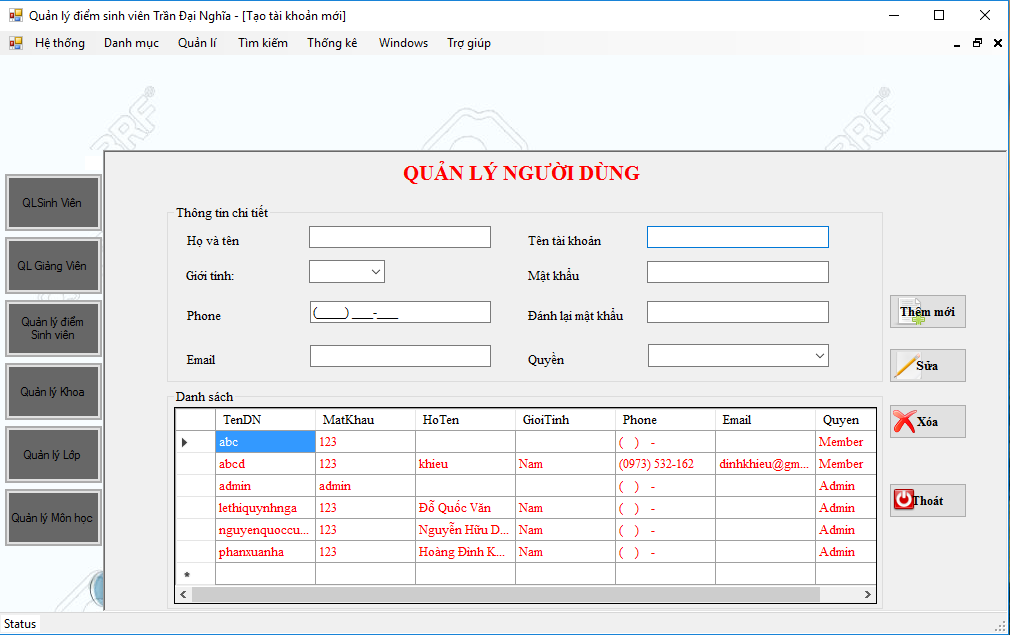


Nếu muốn truy cập danh sách, tìm kiếm, thêm hoặc sửa , xóa lớp, giảng viên, khoa, sinh viên thì click chuột vào QL Sinh Viên, QL Khoa, QL Lớp giao diện như hình sau:

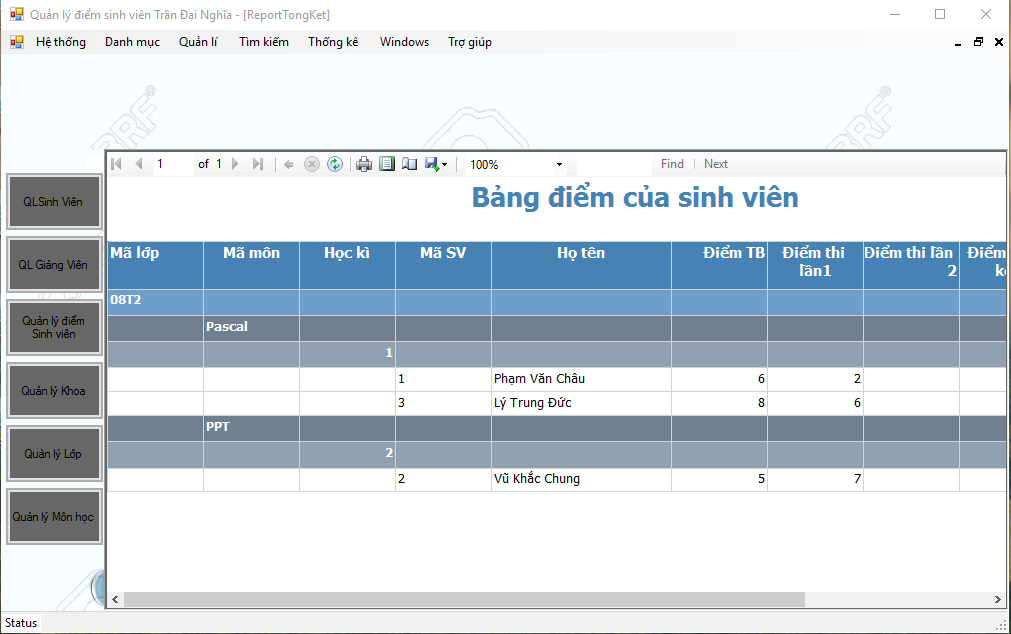




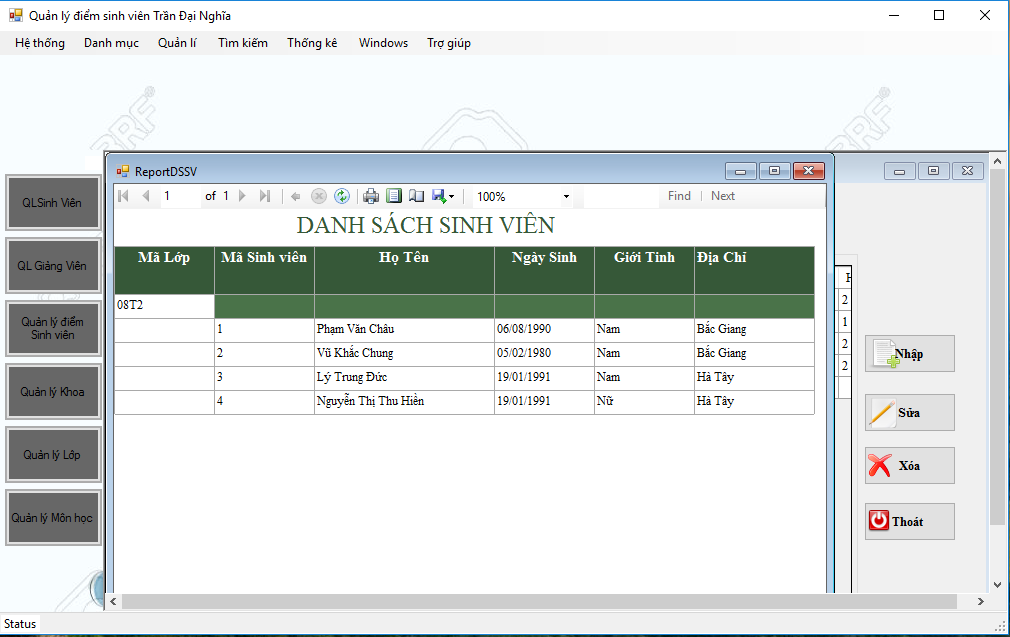
Trong quá trình sử dụng hệ thống, nếu bộ phận quản lí muốn thay đổi người sử dụng và vận hành hệ thống này thì có thể thay đổi như sau:



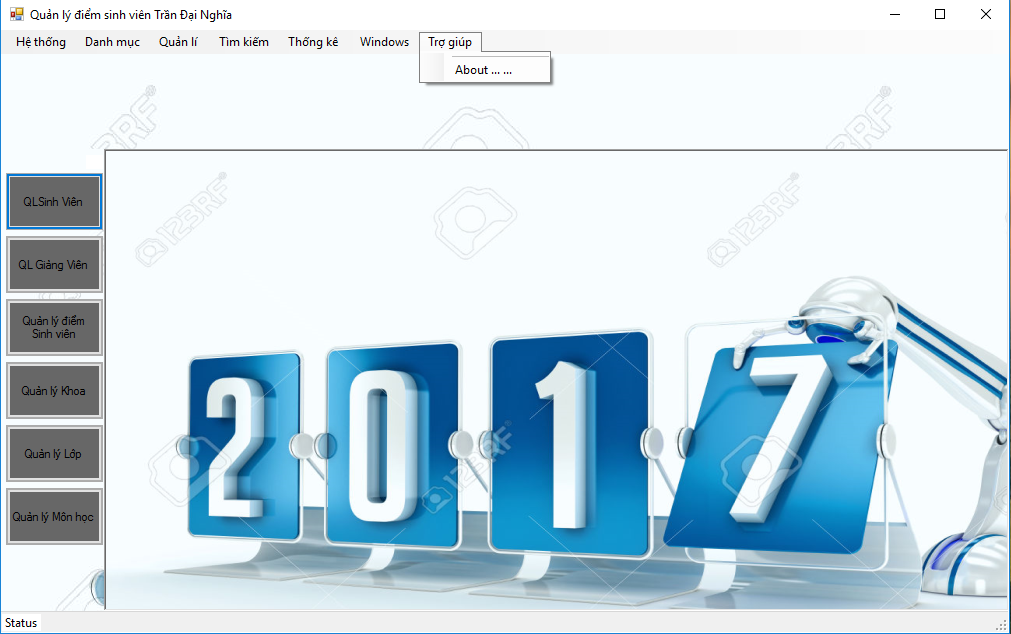
Phần mềm hỗ trợ thống kê và in ra kết quả học tập cũng như danh sách sinh viên các lớp:



Trong giao diện sử dụng có chức năng thống kê. Click chuột vào Thống kê và chọn mục muốn in ra.



Trong quá trình sử dụng hệ thống, nếu có vấn đề khó khăn hoặc chưa hiểu rõ về cơ chế hoạt động thì có thể Click chuột vào mục Trợ giúp. Hệ thống sẽ hỗ trợ tối đa cho người dùng.



**KẾT LUẬN**

Qua 3 chương phân tích và tìm hiểu hệ thống thông tin về bài toán quản lí điểm sinh viên dân sự Trường đại học Trần Đại Nghĩa. Nhóm chúng em đã đưa ra những phân tích về hệ thống ở cả hai phuong 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 sơ đồ 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 trùng lặp và hạn chế. Thiết kế hệ thống còn chưa đúng với thực tế.

Nhưng trong quá trình làm bài, nhóm đã nhận được sự hỗ trợ nhiệt tình từ giáo viên hướng dẫn. Nhóm rất mong nhận được hơn nữa những ý kiến đóng góp để hoàn thành đề tài được tốt hơn. Chúng em xin chân thành cảm ơn Thầy.

**TÀI LIỆU THAM KHẢO**

* Phân tích và thiết kế hệ thống thông tin – Khoa CNTT Đại Học Trần Đại Nghĩa.
* Lập trình hướng đối tượng – Khoa CNTT Đại học Trần Đại Nghĩa.