**LỜI CẢM ƠN**

Sau quá trình học tập và rèn luyện qua môn học cơ sở dữ liệu 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.

Em xin gửi lời cảm ơn chân thành đến thầy cô đã quan tâm hướng dẫn truyền đạt học những kiến thức và kinh nghiệm cho em trong suốt thời gian học học tập bộ môn Cơ sở dữ liệu

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

|  |  |
| --- | --- |
|  | Hà Nội, tháng 11 năm 2022 |

**MỤC LỤC**

[CHƯƠNG 1: KHÁI QUÁT VỀ CƠ SỞ DỮ LIỆU 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)

2.7 Quản lý học lại và học cải thiện ................................................................................46

2.8 Danh sách thi lại........................................................................................................49

2.9 Chức năng tìm điểm..................................................................................................52

[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 giải quyết được những khó khăn trên và tăng tính hiệu quả cho công tác quản lí.

# CHƯƠNG 1: KHÁI QUÁT VỀ CỞ SỞ DỮ LIỆU

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

Trong thời buổi công nghệ số hiện nay, nhiều quy trình, công đoạn hay các hệ thống quản trị đều được mã hóa và vận hành bởi các thiết bị, phần mềm nhằm giúp doanh nghiệp đạt được hiệu suất làm việc tốt nhất. Trên cơ sở đó, các hệ thống quản trị cơ sở dữ liệu ra đời và đóng vai trò quan trọng trong xử lý và kiểm soát nguồn thông tin. Cụ thể, hệ thống quản trị CSDL có các chức năng chính như sau:

Cung cấp môi trường tạo lập cơ sở dữ liệu: Hệ quản trị CSDL đóng vai trò cung cấp cho người dùng một ngôn ngữ định nghĩa dữ liệu để mô tả, khai báo kiểu dữ liệu, các cấu trúc dữ liệu.

Cung cấp cách cập nhật và khai thác dữ liệu: Hệ quản trị CSDL cung cấp cho người dùng ngôn ngữ thao tác dữ liệu để diễn tả các yêu cầu, các thao tác cập nhật và khai thác cơ sở dữ liệu. Thao tác dữ liệu bao gồm: Cập nhật (nhập, sửa, xóa dữ liệu), Khai thác (tìm kiếm, kết xuất dữ liệu).

Cung cấp các công cụ kiểm soát, điều khiển các truy cập vào cơ sở dữ liệu nhằm đảm bảo thực hiện một số yêu cầu cơ bản của hệ cơ sở dữ liệu. Bao gồm:

(1) Đảm bảo an ninh, phát hiện và ngăn chặn các truy cập bất hợp pháp.

(2) Duy trì tính nhất quán của dữ liệu.

(3) Tổ chức và điều khiển các truy cập.

(4) Khôi phục cơ sở dữ liệu khi có sự cố về phần cứng hay phần mềm.

(5) Quản lí các mô tả dữ liệu

# 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 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:

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**

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

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**

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.



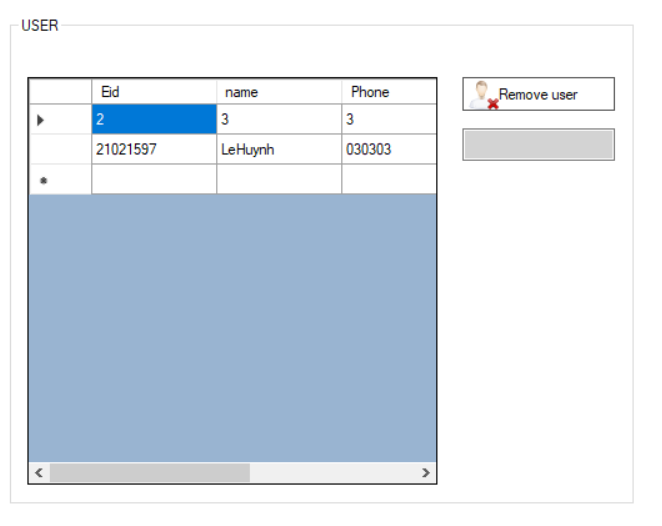
**2.2. Vẽ sơ đồ ERD hệ thống.**



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

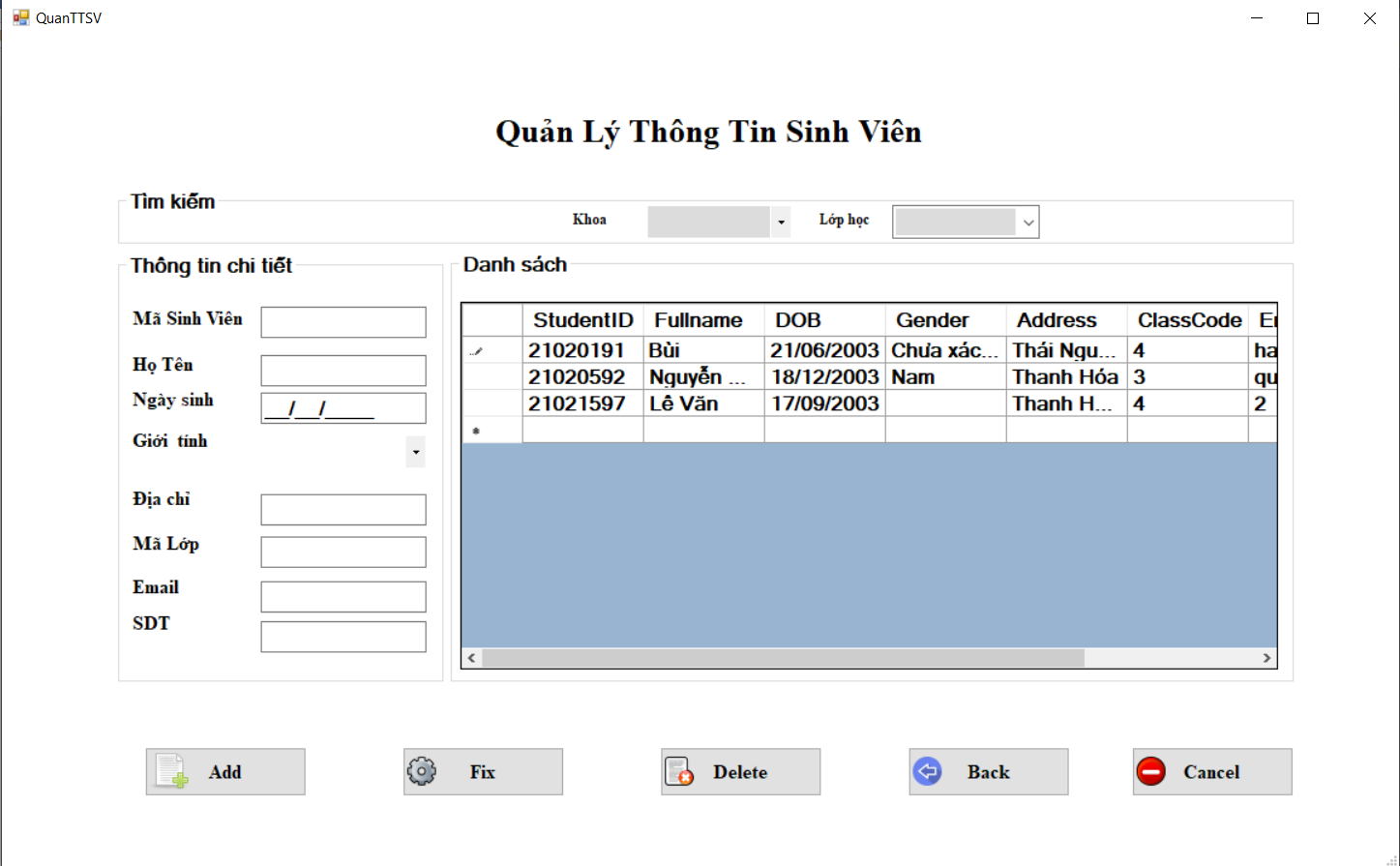
**1. Quản lý người dùng**

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**

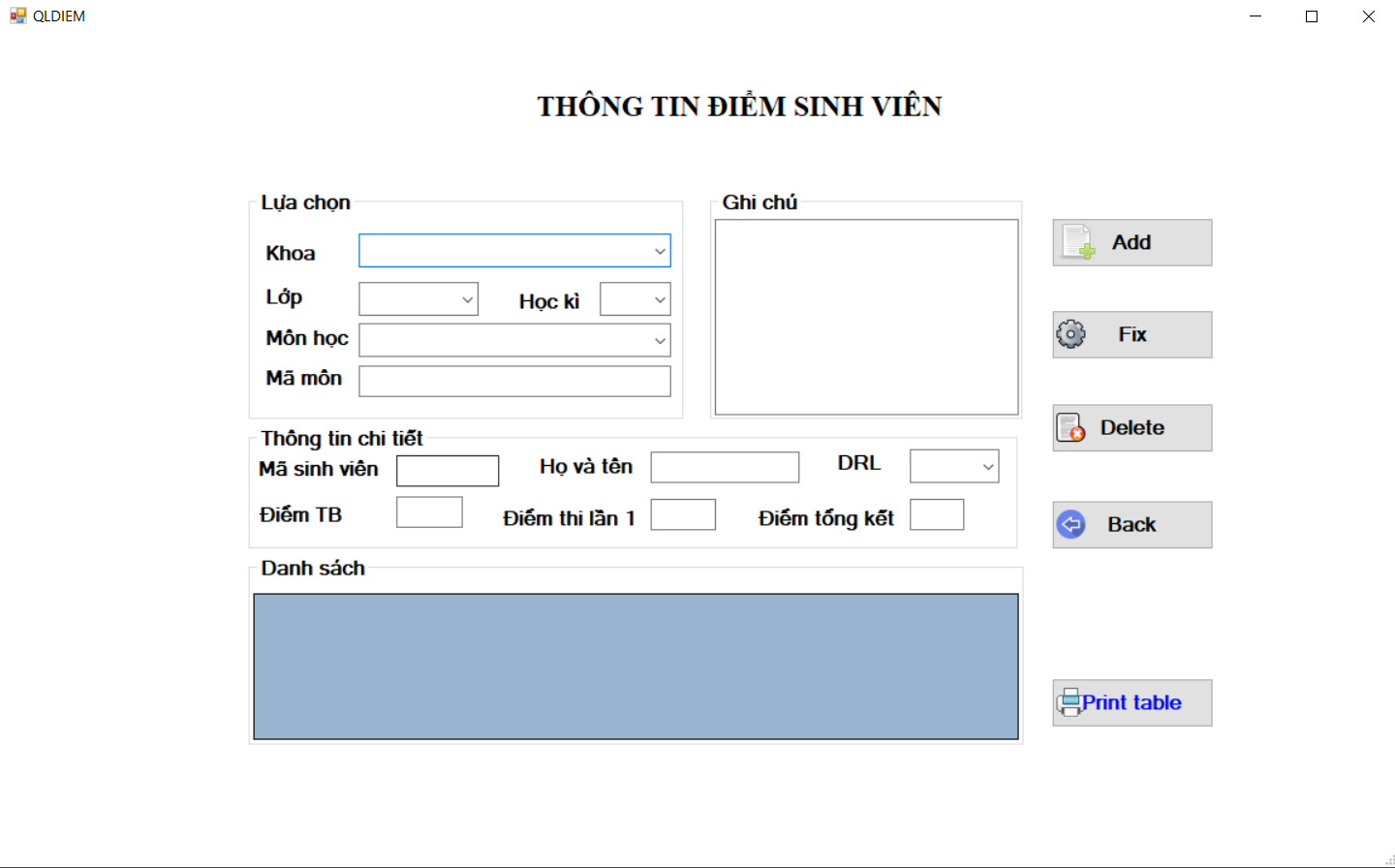


Form nhập thông tin sinh viên chỉ sử dụng bảng dữ liệu sinh viên

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\*0.4+ điểm lần 2 \*0.6

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**

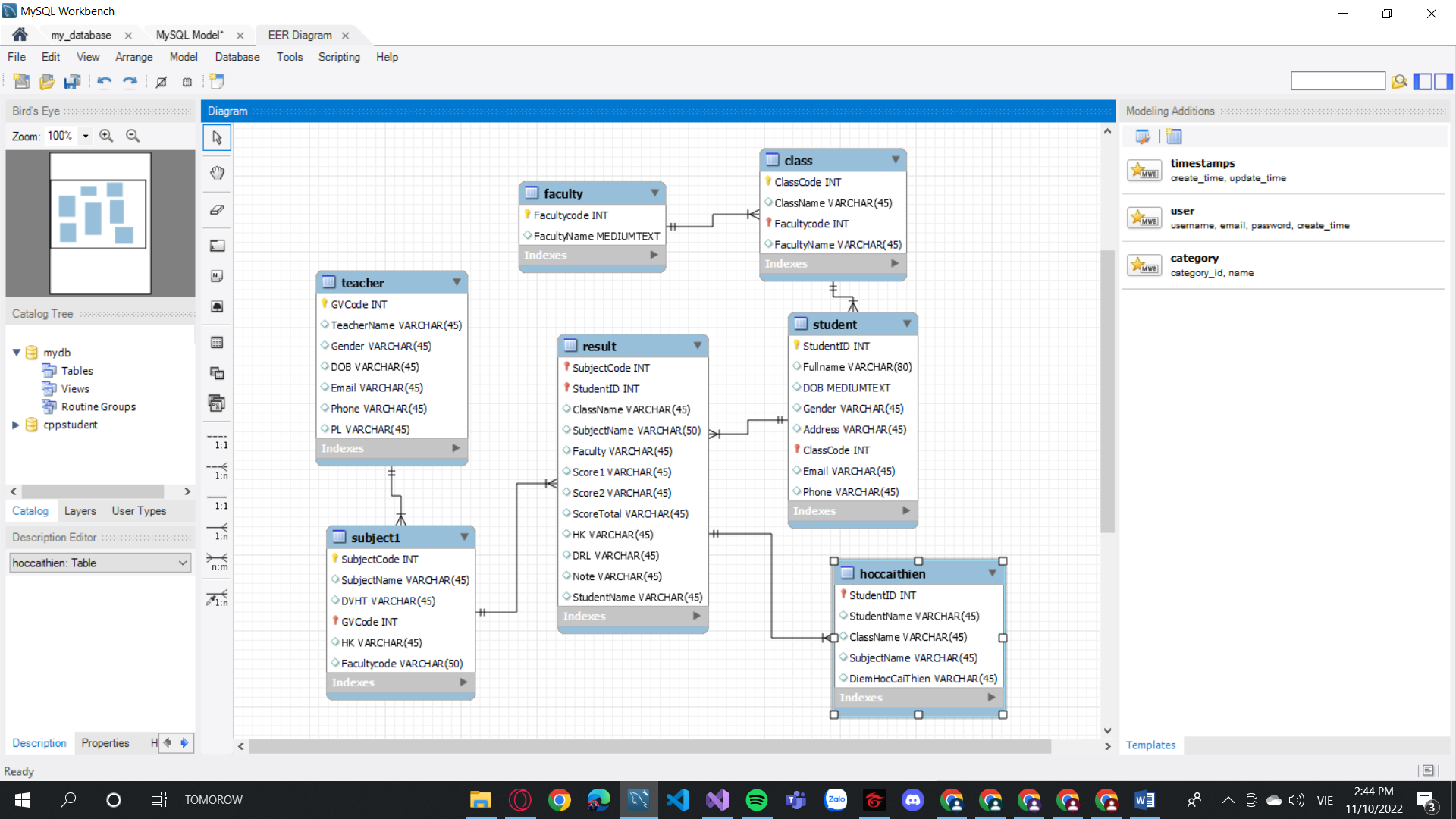
Báo cáo danh sách sinh viên của từng lớp. Trong báo cáo các bảng sinh viên, DM lớp. Cụ thể là những thuộc tính Hoten\_SV, Ngaysinh, Diachi….

**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….

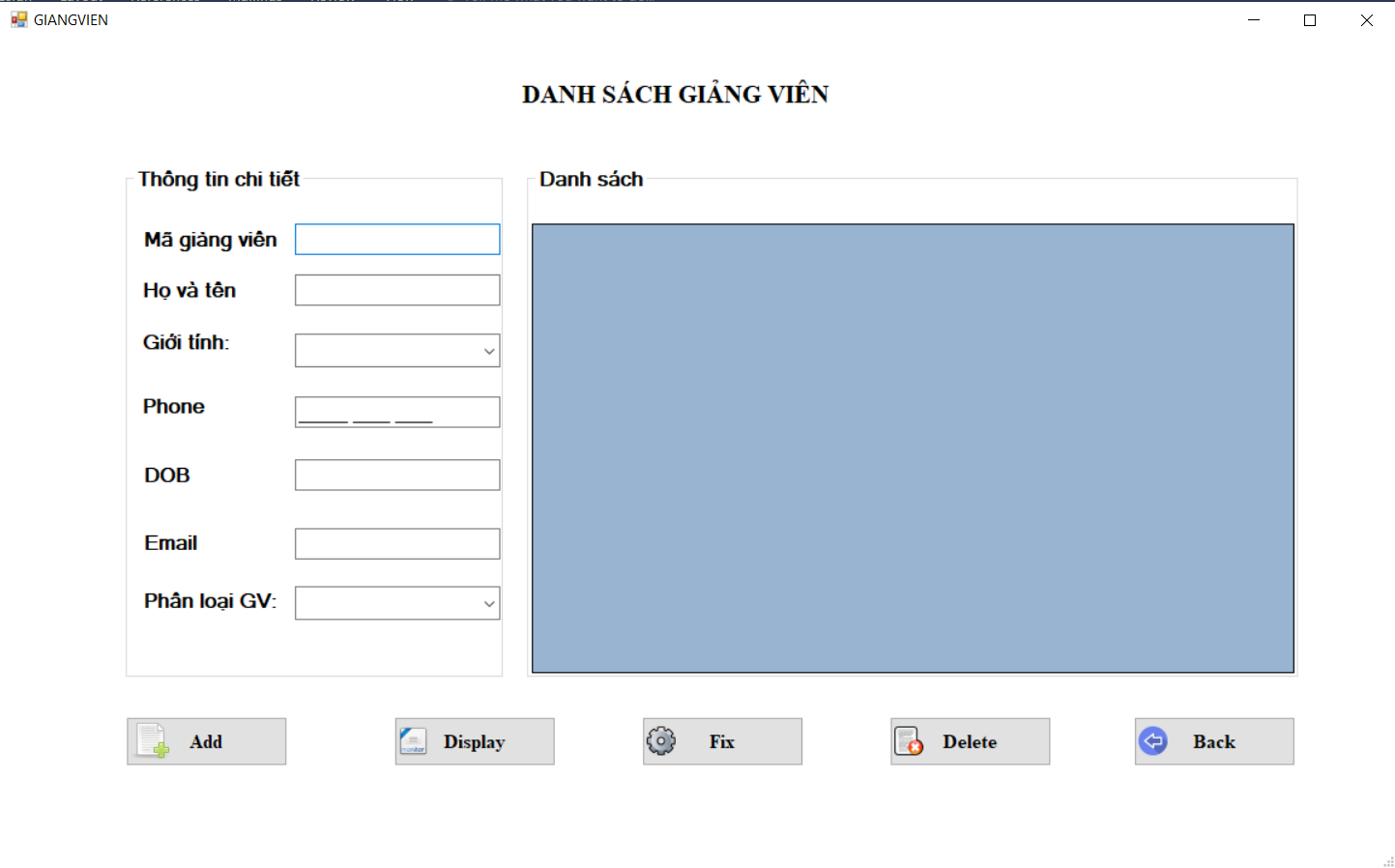
# CHƯƠNG 4: SƠ ĐỒ QUAN HỆ VÀ CÁC LỚP GIẢI QUYẾT CHỨC NĂNG

**1. Sơ đồ quan hệ**



**2. Các lớp giải quyết chức năng**

**2.1. Lớp quản lý giảng viên**



private: System::Void Load22() {

DataTable^ sqlDt5 = gcnew DataTable();

dataGridviewDSGV->DataSource = sqlDt5;

}

private: System::Void GVUpload() {

Load22();

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from teacher";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewDSGV->DataSource = sqlDt;

}

private: System::Void btnAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

sqlQuery = "insert into teacher( GVCode, TeacherName, Gender, DOB, Email, Phone, PL)" + "value('" + txtMaGV->Text + "','" + txtHoTen->Text +"','" +cboGioiTinh->Text +"','"+txtDOB->Text+"','"+txtEmail->Text+"','"+mskPhone->Text +"','"+cboPhanloai->Text+ "')";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Student Management System Update");

GVUpload();

}

private: System::Void btnDelete\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

String^ delete2 = txtMaGV->Text;

if (delete2->Length == 0) {

MessageBox::Show("Error", " Error", MessageBoxButtons::OK);

return;

}

//Delete student

try {

sqlConn->Open();

System::Windows::Forms::DialogResult iDelete;

iDelete = MessageBox::Show("Comfirm if you want to delete teacher data", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iDelete == System::Windows::Forms::DialogResult::Yes) {

sqlQuery = "delete from teacher where GVCode = '" + txtMaGV->Text + "'";

}

sqlQuery = "delete from teacher where GVCode = '" + txtMaGV->Text + "'";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show("Erorr", "Student Management System", MessageBoxButtons::OK);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Deleted faculty data!");

Load22();

GVUpload();

}

private: System::Void btnExit\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Hide();

}

private: System::Void btnXem\_Click(System::Object^ sender, System::EventArgs^ e) {

GVUpload();

}

private: System::Void GIANGVIEN\_Load(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void label8\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void btnSua\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

//GVCode, TeacherName, Gender, DOB, Email, Phone, PL

sqlConn->Open();

sqlQuery = "Update teacher Set GVCode=N'" + txtMaGV->Text + "',TeacherName='" +

txtHoTen->Text + "',Gender='" + cboGioiTinh->Text + "',DOB='" +

txtDOB->Text + "',Email='" + txtEmail->Text + "',Phone='"+mskPhone->Text +"',PL='"+cboPhanloai->Text +"' where GVCode='" + txtMaGV->Text + "' ";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

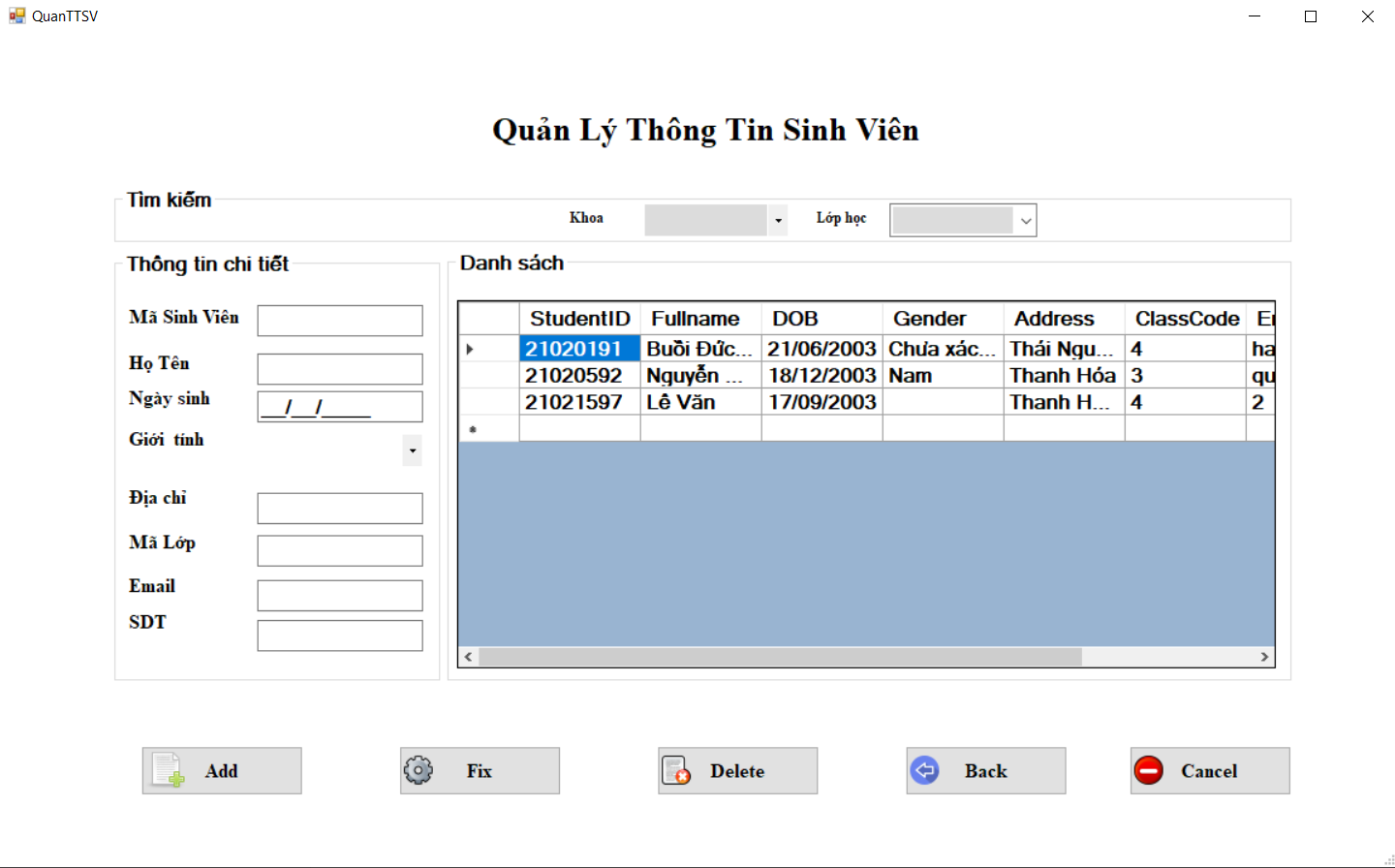
sqlConn->Close();

}

GVUpload();

}

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

****

private: System::Void userUpload() {

DataTable^ sqlDt = gcnew DataTable();

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from student";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewDSSV->DataSource = sqlDt;

}

private: System::Void Upload() {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from student";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewDSSV->DataSource = sqlDt;

}

private: System::Void QuanTTSV\_Load(System::Object^ sender, System::EventArgs^ e) {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

//Add khoa vaof cbo

sqlConn1->Open();

cboKhoahoc->Items->Clear();

cboKhoahoc->Text = "";

String^ aaa = "Select FacultyName from faculty";

sqlCmd1 = gcnew MySqlCommand(aaa, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

while (sqlDd1->Read()) {

cboKhoahoc->Items->Add(sqlDd1->GetString(0));

}

sqlConn1->Close();

Upload();

}

private: System::Void cboKhoahoc\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

//Add lop trong khoa

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

cboLop->Items->Clear();

cboLop->Text = "";

String^ aa = "Select ClassName from class where FacultyName = '" + cboKhoahoc->Text + "'";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

cboLop->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

}

private: System::Void cboLop\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn1->Open();

txtMaLop->Clear();

txtMaLop->Text = "";

String^ mon = "Select ClassCode from class where ClassName ='" + cboLop->Text + "' ";

sqlCmd1 = gcnew MySqlCommand(mon, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

sqlDd1->Read();

String^ du = sqlDd1->GetString(0);

txtMaLop->Text = du;

sqlConn1->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn1->Close();

}

}

private: System::Void btnAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

sqlQuery = "insert into student (StudentID, Fullname," + " DOB, Gender, Address, ClassCode, Email, Phone)" + "value('" + txtMaSV->Text + "',N'" + txtHoTen->Text + "',N'" + mskNgaySinh->Text + "',N'" + cboGioiTinh->Text + "',N'" + txtDiaChi->Text + "', '" + txtMaLop->Text + "','" + txtEmail->Text + "','" + txtSDT->Text + "')";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

Upload();

}

private: System::Void btnXoa\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

String^ delete2 = txtMaSV->Text;

if (delete2->Length == 0) {

MessageBox::Show("Error", " Error", MessageBoxButtons::OK);

return;

}

//Delete student

try {

sqlConn->Open();

System::Windows::Forms::DialogResult iDelete;

iDelete = MessageBox::Show("Comfirm if you want to delete class data", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iDelete == System::Windows::Forms::DialogResult::Yes) {

sqlQuery = "delete from student where StudentID = '" + txtMaSV->Text + "'";

}

sqlQuery = "delete from student where StudentID = '" + txtMaSV->Text + "'";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show("Erorr", "Student Management System", MessageBoxButtons::OK);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Deleted student data!");

Upload();

userUpload();

}

private: System::Void btnThoat\_Click(System::Object^ sender, System::EventArgs^ e) {

System::Windows::Forms::DialogResult iExit;

iExit = MessageBox::Show("Comfirm if you want to cancel", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iExit == System::Windows::Forms::DialogResult::Yes) {

Application::Exit();

}

}

private: System::Void btnBAck\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Hide();

}

private: System::Void groupBox3\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void groupBox2\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void groupBox1\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void btnSua\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

//GVCode, TeacherName, Gender, DOB, Email, Phone, PL

sqlConn->Open();

sqlQuery = "Update student Set StudentID=N'" + txtMaSV->Text + "',Fullname='" + txtHoTen->Text + "',DOB='" +

mskNgaySinh->Text + "',Gender='" + cboGioiTinh->Text + "',Address='" + txtDiaChi->Text + "',ClassCode='" + txtMaLop->Text + "',Email='" + txtEmail->Text + "',Phone='" + txtSDT->Text + "' where StudentID='" + txtMaSV->Text + "' ";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

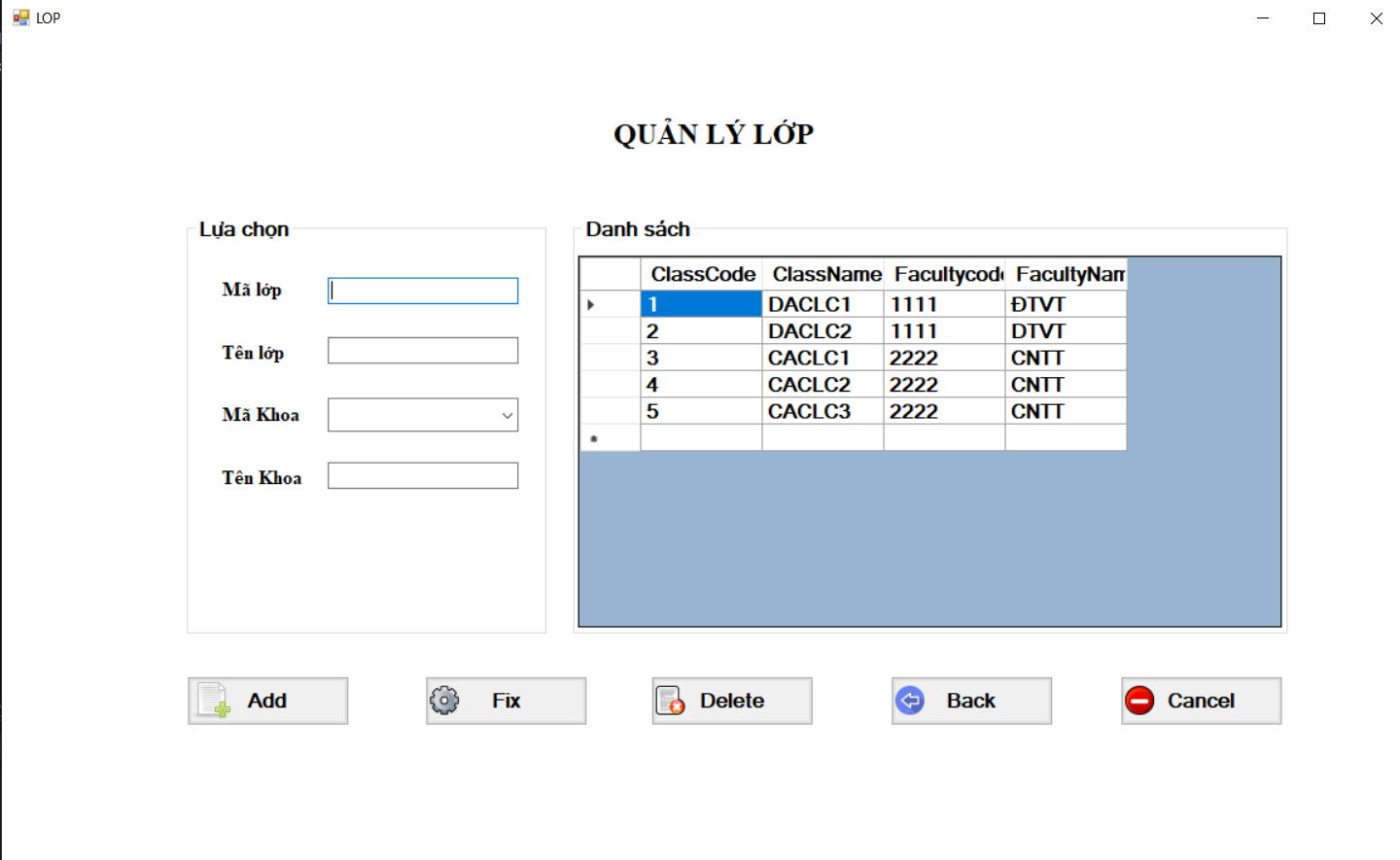
sqlConn->Close();

}

Upload();

}

**2.3. Lớp quản lý lớp**



private: System::Void Load22() {

DataTable^ sqlDt5 = gcnew DataTable();

dataGridviewLop->DataSource = sqlDt5;

}

private: System::Void Upload() {

Load22();

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from class";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewLop->DataSource = sqlDt;

}

private: System::Void btnAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

sqlQuery = "insert into class(ClassCode, ClassName, Facultycode, FacultyName)" + "value('" +txtMaLop->Text+"',N'"+txtTenlop->Text+"','"+cboKhoa->Text+ "','"+txtFacultyName->Text+"')";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Student Management System Update");

Upload();

}

private: System::Void LOP\_Load(System::Object^ sender, System::EventArgs^ e) {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn1->Open();

cboKhoa->Items->Clear();

cboKhoa->Text = "";

String^ aa = "Select Facultycode from faculty";

sqlCmd1 = gcnew MySqlCommand(aa, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

while (sqlDd1->Read()) {

cboKhoa->Items->Add(sqlDd1->GetString(0));

}

sqlConn1->Close();

Upload();

}

private: System::Void cboKhoa\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

txtFacultyName->Clear();

txtFacultyName->Text = "";

String^ aa = "Select FacultyName from faculty where Facultycode ='"+cboKhoa->Text +"' ";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlDd->Read();

String^ s = sqlDd->GetString(0);

txtFacultyName->Text = s;

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

}

private: System::Void btnDelete\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

String^ delete2 = txtMaLop->Text;

if (delete2->Length == 0) {

MessageBox::Show("Error", " Error", MessageBoxButtons::OK);

return;

}

//Delete student

try {

sqlConn->Open();

System::Windows::Forms::DialogResult iDelete;

iDelete = MessageBox::Show("Comfirm if you want to delete class data", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iDelete == System::Windows::Forms::DialogResult::Yes) {

sqlQuery = "delete from class where ClassCode = '" + txtMaLop->Text + "'";

}

sqlQuery = "delete from class where ClassCode = '" + txtMaLop->Text + "'";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show("Erorr", "Student Management System", MessageBoxButtons::OK);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Deleted class data!");

Upload();

}

private: System::Void btnExit\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Hide();

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

System::Windows::Forms::DialogResult iExit;

iExit = MessageBox::Show("Comfirm if you want to cancel", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iExit == System::Windows::Forms::DialogResult::Yes) {

Application::Exit();

}

}

private: System::Void btnSua\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

//GVCode, TeacherName, Gender, DOB, Email, Phone, PL

sqlConn->Open();

sqlQuery = "Update class Set ClassCode=N'" + txtMaLop->Text + "',ClassName='" +

txtTenlop->Text + "',Facultycode='"+cboKhoa->Text+"',FacultyName='"+txtFacultyName->Text+"' where ClassCode='" + txtMaLop->Text + "' ";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

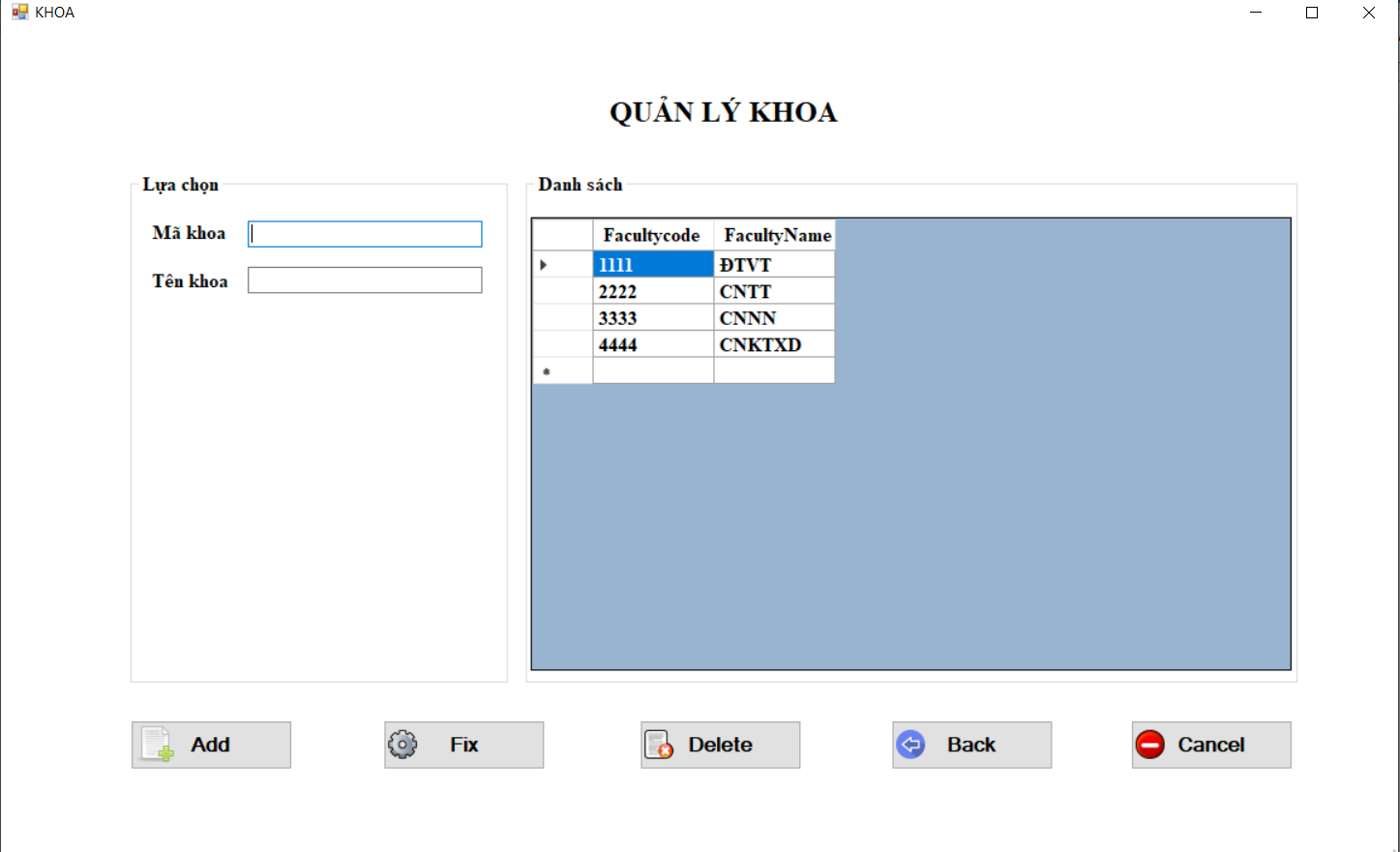
sqlConn->Close();

}

Upload();

}

**2.4. Lớp quản lý khoa**



private: System::Void Load22() {

DataTable^ sqlDt5 = gcnew DataTable();

dataGridviewKhoa->DataSource = sqlDt5;

}

private: System::Void KhoaUpload() {

Load22();

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from faculty";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewKhoa->DataSource = sqlDt;

}

private: System::Void btnAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

sqlQuery = "insert into faculty( Facultycode, FacultyName)" + "value('" +txtKhoa->Text+"','"+txtTenKhoa->Text + "')";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Student Management System Update");

KhoaUpload();

}

private: System::Void KHOA\_Load(System::Object^ sender, System::EventArgs^ e) {

KhoaUpload();

}

private: System::Void btnExit\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Hide();

}

private: System::Void btnDelete\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

String^ delete2 = txtKhoa->Text;

if (delete2->Length == 0) {

MessageBox::Show("Error", " Error", MessageBoxButtons::OK);

return;

}

//Delete student

try {

sqlConn->Open();

System::Windows::Forms::DialogResult iDelete;

iDelete = MessageBox::Show("Comfirm if you want to delete student data", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iDelete == System::Windows::Forms::DialogResult::Yes) {

sqlQuery = "delete from faculty where Facultycode = '" + txtKhoa->Text + "'";

}

sqlQuery = "delete from faculty where Facultycode = '" + txtKhoa->Text + "'";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show("Erorr", "Erorr", MessageBoxButtons::OK);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Deleted faculty data!");

Load22();

KhoaUpload();

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

System::Windows::Forms::DialogResult iExit;

iExit = MessageBox::Show("Comfirm if you want to cancel", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iExit == System::Windows::Forms::DialogResult::Yes) {

Application::Exit();

}

}

private: System::Void btnSua\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

//GVCode, TeacherName, Gender, DOB, Email, Phone, PL

sqlConn->Open();

sqlQuery = "Update faculty Set Facultycode=N'" + txtKhoa->Text + "',FacultyName='" +

txtTenKhoa->Text + "' where Facultycode='" + txtKhoa->Text + "' ";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

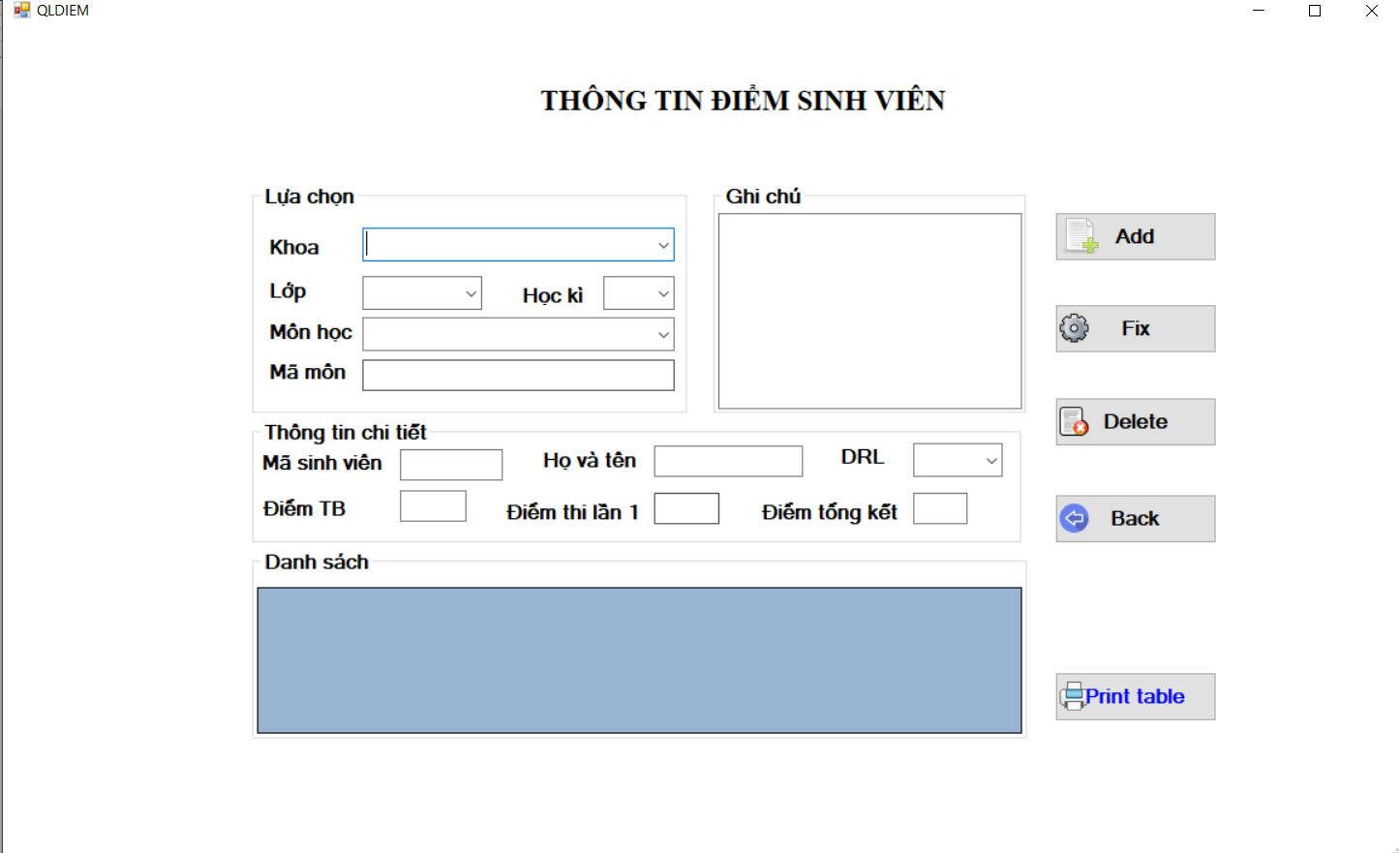
sqlConn->Close();

}

KhoaUpload();

}

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



private: System::Void userUpload() {

DataTable^ sqlDt = gcnew DataTable();

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from result";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewDiem->DataSource = sqlDt;

}

private: System::Void Upload() {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from result";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewDiem->DataSource = sqlDt;

}

private: System::Void QLDIEM\_Load(System::Object^ sender, System::EventArgs^ e) {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

//Add khoa vaof cbo

sqlConn1->Open();

cboKhoaHoc->Items->Clear();

cboKhoaHoc->Text = "";

String^ aaa = "Select FacultyName from faculty";

sqlCmd1 = gcnew MySqlCommand(aaa, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

while (sqlDd1->Read()) {

cboKhoaHoc->Items->Add(sqlDd1->GetString(0));

}

sqlConn1->Close();

}

private: System::Void btnAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

sqlQuery = "insert into result (SubjectCode, StudentID, ClassName,"+" SubjectName, Faculty, Score1, Score2,ScoreTotal, HK,DRL, Note, StudentName)" + "value('" + txtScode->Text + "',N'" + txtMaSV->Text + "',N'" +cboLop->Text + "',N'" + cboMonHoc->Text + "',N'" + cboKhoaHoc->Text + "', '" + txtDiemTB->Text +"','"+txtDiemThi1->Text+"','"+txtDiemTK->Text+"','"+cboHocKi->Text+"','"+cboDRL->Text+"','"+txtGhiChu->Text+"','" +txtHoTen->Text+ "')";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

Upload();

userUpload();

String^ ok = txtDiemTK->Text;

if (ok == "0") {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn1->Open();

sqlQuery1 = "insert into hlai ( StudentID, StudentName,ClassName, SubjectName, HK,FacultyName )" + "value('" + txtMaSV + "','" + txtHoTen->Text + "','" + cboLop->Text + "','" + cboMonHoc->Text + "','" + cboHocKi->Text + "', '" + cboKhoaHoc->Text + "')";

sqlCmd1 = gcnew MySqlCommand(sqlQuery1, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

sqlConn1->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn1->Close();

}

}

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void cboKhoaHoc\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

//Add lop trong khoa

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

cboLop->Items->Clear();

cboLop->Text = "";

String^ aa = "Select ClassName from class where FacultyName = '"+cboKhoaHoc->Text+"'";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

cboLop->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

//Add mon vaof cbo

sqlConn2->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn2->Open();

cboMonHoc->Items->Clear();

cboMonHoc->Text = "";

String^ aaaq = "Select SubjectName from subject1";

sqlCmd2 = gcnew MySqlCommand(aaaq, sqlConn2);

sqlDd2 = sqlCmd2->ExecuteReader();

while (sqlDd2->Read()) {

cboMonHoc->Items->Add(sqlDd2->GetString(0));

}

sqlConn2->Close();

Upload();

userUpload();

}

private: System::Void btnExit\_Click(System::Object^ sender, System::EventArgs^ e) {

cboKhoaHoc->Items->Clear();

this->Close();

}

private: System::Void txtMaSV\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

//Add ten vaof txt

sqlConn3->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn3->Open();

cboMonHoc->Items->Clear();

cboMonHoc->Text = "";

String^ i = "Select Fullname from student where StudentID = '" + txtMaSV->Text + "'";

sqlCmd3 = gcnew MySqlCommand(i, sqlConn3);

sqlDd3 = sqlCmd3->ExecuteReader();

while (sqlDd3->Read()) {

String^ sa = sqlDd3->GetString(0);

txtHoTen->Text = sa;

sqlConn3->Close();

}

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn3->Close();

}

}

private: System::Void cboMonHoc\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

//Add ma mon

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn1->Open();

txtScode->Clear();

txtScode->Text = "";

String^ mon = "Select SubjectCode from subject1 where SubjectName ='" + cboMonHoc->Text + "' ";

sqlCmd1 = gcnew MySqlCommand(mon, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

sqlDd1->Read();

String^ du = sqlDd1->GetString(0);

txtScode->Text = du;

sqlConn1->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn1->Close();

}

}

private: System::Void txtDiemThi1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

txtGhiChu->Text = "";

float tb, ck, tk;

try {

tb = Convert::ToDouble(txtDiemTB->Text);

ck = Convert::ToDouble(txtDiemThi1->Text);

tk = tb \* 0.4 + ck \* 0.6;

if (tk < 4) {

txtGhiChu->Text = "Hoc lai";

}

if (tk >= 9) {

tk = 4;

}

else if (tk >= 8.5 && tk < 9) {

tk = 3.7;

}

else if (tk >= 8 && tk < 8.5) {

tk = 3.5;

}

else if (tk >= 7 && tk < 8) {

tk = 3;

}

else if (tk >= 6.5 && tk < 7) {

tk = 2.5;

}

else if (tk >= 5.5 && tk < 6.5) {

tk = 2;

}

else if (tk >= 5 && tk < 5.5) {

tk = 1.5;

}

else if (tk >= 4 && tk < 5) {

tk = 1;

}

else {

tk = 0;

}

txtDiemTK->Text = Convert::ToString(tk);

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

}

private: System::Void txtDiemTB\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

txtGhiChu->Text = "";

}

private: System::Void btnDelete\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

String^ delete2 = txtScode->Text;

if (delete2->Length == 0) {

MessageBox::Show("Error", " Error", MessageBoxButtons::OK);

return;

}

//Delete student

try {

sqlConn->Open();

System::Windows::Forms::DialogResult iDelete;

iDelete = MessageBox::Show("Comfirm if you want to delete class data", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iDelete == System::Windows::Forms::DialogResult::Yes) {

sqlQuery = "delete from result where SubjectCode = '" + txtScode->Text + "'";

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

//GVCode, TeacherName, Gender, DOB, Email, Phone, PL

sqlConn1->Open();

sqlQuery1 = "delete from hoccaithien where SubjectName = '" + cboMonHoc->Text + "'";

sqlCmd1 = gcnew MySqlCommand(sqlQuery1, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

sqlConn1->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn1->Close();

}

}

sqlQuery = "delete from result where SubjectCode = '" + txtScode->Text + "'";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show("Erorr", "Student Management System", MessageBoxButtons::OK);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Deleted result data!");

Upload();

userUpload();

}

private: System::Void btnXuat\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void cboLop\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void label12\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void btnSua\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

//GVCode, TeacherName, Gender, DOB, Email, Phone, PL

sqlConn->Open();

sqlQuery = "Update result Set SubjectCode=N'" + txtScode->Text + "',StudentID='" +txtMaSV->Text + "',StudentName='" +

txtHoTen->Text +"',ClassName='" +cboLop->Text +"',SubjectName='" +cboMonHoc->Text + "',Faculty='" + cboKhoaHoc->Text + "',Score1='" + txtDiemTB->Text + "',Score2='" +txtDiemThi1->Text + "',ScoreTotal='" +txtDiemTK->Text + "',HK='" + cboHocKi->Text + "',DRL='" + cboDRL->Text + "',Note='" +txtGhiChu->Text + "' where SubjectCode='" + txtScode->Text + "' ";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

Upload();

try {

//GVCode, TeacherName, Gender, DOB, Email, Phone, PL

sqlConn->Open();

sqlQuery = "Update hoccaithien Set StudentID=N'" + txtMaSV->Text + "',StudentName='" + txtHoTen->Text + "',ClassName='" +

cboLop->Text + "',SubjectName='" + cboMonHoc->Text + "',SubjectName='" + cboMonHoc->Text + "',DiemHocCaiThien='" + txtDiemTK->Text + "' where StudentID='" + txtMaSV->Text + "' ";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

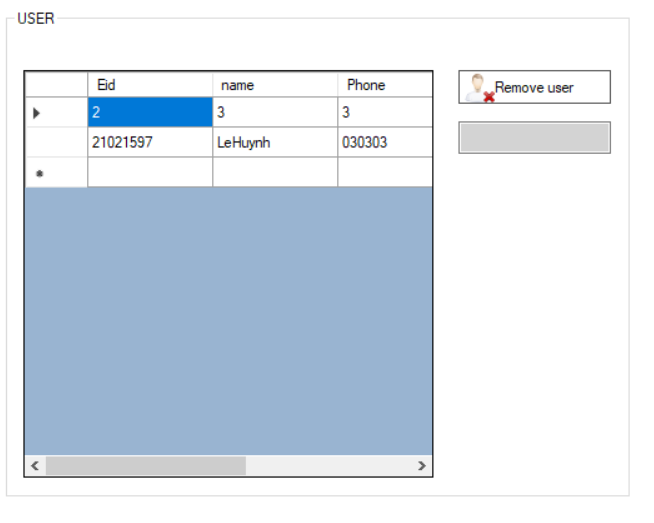
finally {

sqlConn->Close();

}

}

**2.6. Lớp quản lý người dùng**



//Dang ky

private: System::Void btnOk\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

String^ Id = txtID->Text;

String^ name = txtName->Text;

String^ phone = txtPhone->Text;

String^ username = txtUsername->Text;

String^ password = txtPassword->Text;

String^ comfirmPassword = txtComfirm->Text;

if (Id->Length == 0 || name->Length == 0 || phone->Length == 0 || username->Length == 0 || password->Length == 0 || comfirmPassword->Length == 0) {

MessageBox::Show("Please enter all the fields", "On or more empty fields", MessageBoxButtons::OK);

return;

}

if (String::Compare(password, comfirmPassword) != 0) {

MessageBox::Show("Password and Comfirm Password do not match", "Password Error", MessageBoxButtons::OK);

return;

}

try {

sqlConn->Open();

sqlQuery = "INSERT INTO edata(Eid, name, Phone, user\_name, password)" + "value('" + txtID->Text + "', '" + txtName->Text + "', '" + txtPhone->Text + "', '" + txtUsername->Text + "', '" + txtPassword->Text + "')";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show("Failer to register new user", "Register Failure", MessageBoxButtons::OK);

}

finally {

sqlConn->Close();

}

MessageBox::Show("Registered a new user");

}

//Login

private: System::Void btnLogin\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ conString = L"datasource=localhost;port=3306;username=root;password=123456789";

MySqlConnection^ conDataBase = gcnew MySqlConnection(conString);

MySqlCommand^ cmdDataDase = gcnew MySqlCommand("select \*from database.edata where user\_name='" + this->txtUsername->Text + "'and password='" + this->txtPassword->Text + "'; ", conDataBase);

MySqlDataReader^ myReader;

try {

conDataBase->Open();

myReader = cmdDataDase->ExecuteReader();

int count = 0;

while (myReader->Read()) {

count++;

}

if (count == 1) {

MyForm2^ f2 = gcnew MyForm2();

f2->ShowDialog();

}

else if (count > 1) {

MessageBox::Show("Unknown");

}

else {

MessageBox::Show("Wrong password or username!");

}

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

}

//Remove user

private: System::Void btnDelete\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn2->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database2;

String^ delete1 = txtRemove->Text;

if (delete1->Length == 0 ) {

MessageBox::Show("Error", "Student Management System", MessageBoxButtons::OK);

return;

}

try {

sqlConn2->Open();

System::Windows::Forms::DialogResult iRemove;

iRemove = MessageBox::Show("Comfirm if you want to remove user", "Student Management System", MessageBoxButtons::YesNo, MessageBoxIcon::Question);

if (iRemove == System::Windows::Forms::DialogResult::Yes) {

sqlQuery2 = "delete from edata where Eid = '" + txtRemove->Text + "'";

}

sqlCmd2 = gcnew MySqlCommand(sqlQuery2, sqlConn2);

sqlDd2 = sqlCmd2->ExecuteReader();

sqlConn2->Close();

}

catch (Exception^ ex) {

MessageBox::Show("Erorr","Student Management System", MessageBoxButtons::OK);

}

finally {

sqlConn2->Close();

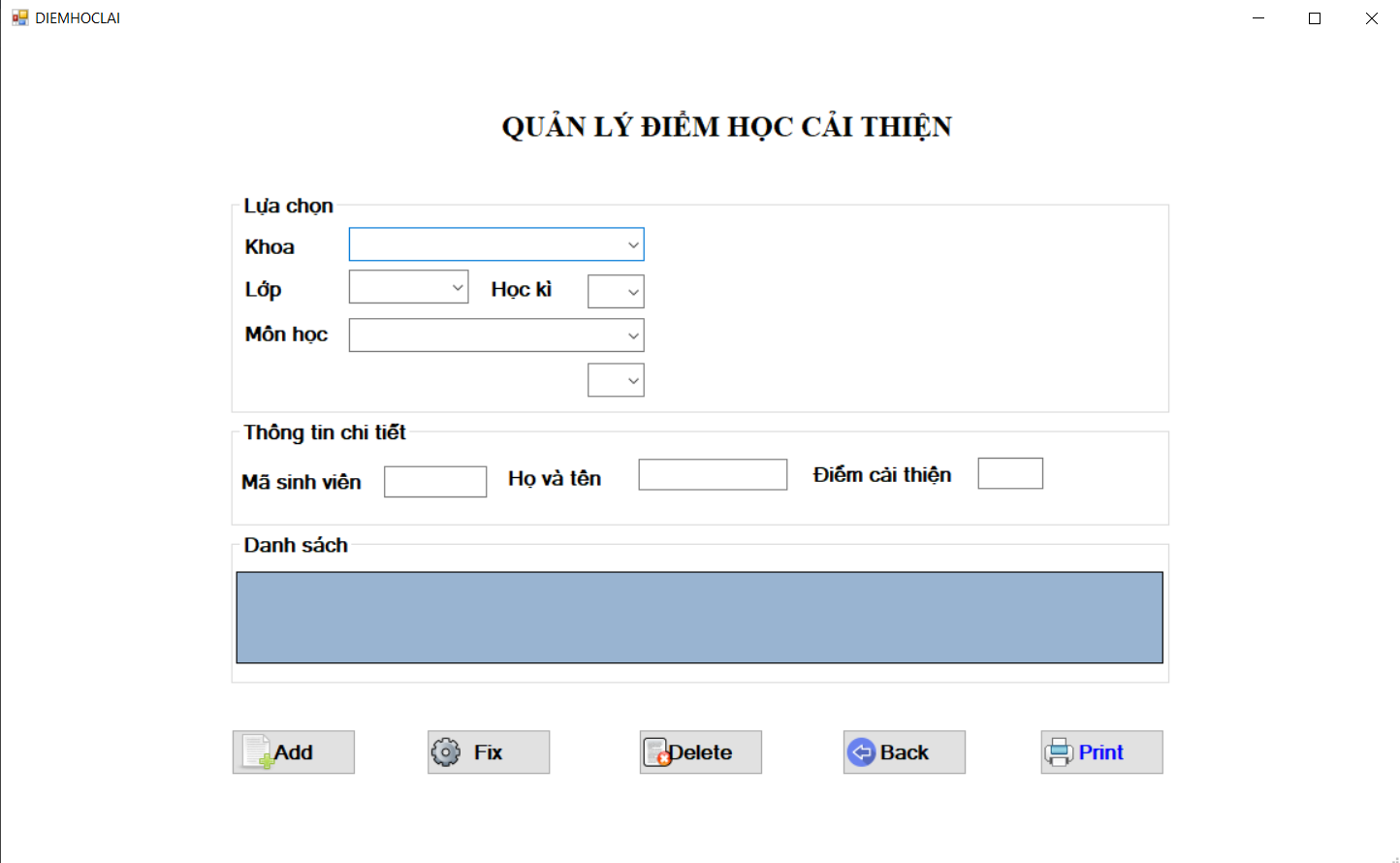
}

MessageBox::Show("Access Permission removed access permission! ");

userUpload();

}

* 1. **Quản lý danh sách học lại và học cải thiện**



private: System::Void Upload() {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from hoccaithien";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewDiem->DataSource = sqlDt;

}

private: System::Void DIEMHOCLAI\_Load(System::Object^ sender, System::EventArgs^ e) {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

//Add khoa vaof cbo

sqlConn1->Open();

cboKhoaHoc->Items->Clear();

cboKhoaHoc->Text = "";

String^ aaa = "Select FacultyName from faculty";

sqlCmd1 = gcnew MySqlCommand(aaa, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

while (sqlDd1->Read()) {

cboKhoaHoc->Items->Add(sqlDd1->GetString(0));

}

sqlConn1->Close();

}

private: System::Void cboKhoaHoc\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

//Add lop trong khoa

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

cboLop->Items->Clear();

cboLop->Text = "";

String^ aa = "Select ClassName from class where FacultyName = '" + cboKhoaHoc->Text + "'";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

cboLop->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

//Add mon vaof cbo

sqlConn2->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn2->Open();

cboMonHoc->Items->Clear();

cboMonHoc->Text = "";

String^ aaaq = "Select SubjectName from subject1";

sqlCmd2 = gcnew MySqlCommand(aaaq, sqlConn2);

sqlDd2 = sqlCmd2->ExecuteReader();

while (sqlDd2->Read()) {

cboMonHoc->Items->Add(sqlDd2->GetString(0));

}

sqlConn2->Close();

}

private: System::Void cboMonHoc\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

cboHocKi->Items->Clear();

cboHocKi->Text = "";

String^ aa = "Select HK from result where SubjectName = '" + cboMonHoc->Text + "'";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

cboHocKi->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

try {

sqlConn->Open();

txtok->Items->Clear();

txtok->Text = "";

String^ aa = "Select HK from result where SubjectName = '" + cboMonHoc->Text + "'";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

txtok->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn1->Open();

txtDiemThi3->Text = "";

String^ ii = "Select ScoreTotal from result where StudentID = '" + txtMaSV->Text + "' and SubjectName='" + cboMonHoc->Text + "'";

sqlCmd1 = gcnew MySqlCommand(ii, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

sqlDd1->Read();

String^ saa = sqlDd1->GetString(0);

txtDiemThi3->Text = saa;

sqlConn1->Close();

}

catch (Exception^ ex) {

//MessageBox::Show(ex->Message);

}

finally {

sqlConn1->Close();

}

}

private: System::Void txtMaSV\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

//Add ten vaof txt

sqlConn3->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn3->Open();

cboMonHoc->Items->Clear();

cboMonHoc->Text = "";

String^ i = "Select Fullname from student where StudentID = '" + txtMaSV->Text + "'";

sqlCmd3 = gcnew MySqlCommand(i, sqlConn3);

sqlDd3 = sqlCmd3->ExecuteReader();

while (sqlDd3->Read()) {

String^ sa = sqlDd3->GetString(0);

txtHoTen->Text = sa;

sqlConn3->Close();

}

}

catch (Exception^ ex) {

//MessageBox::Show(ex->Message);

}

finally {

sqlConn3->Close();

}

//Cập nhật điểm qua mã sinh viên và môn học

}

private: System::Void btnAdd\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

sqlQuery = "insert into hoccaithien(StudentID, StudentName, ClassName,SubjectName, DiemHocCaiThien)" + "value('" +txtMaSV->Text+ "','" + txtHoTen->Text + "','" + cboLop->Text + "','" + cboMonHoc->Text + "','"+txtDiemThi3->Text + "')";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

Upload();

}

private: System::Void groupBox2\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void btnDelete\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void btnExit\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Hide();

}

private: System::Void btnSua\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

sqlQuery = "Update hoccaithien Set StudentID=N'" + txtMaSV->Text + "',StudentName='" +

txtHoTen->Text + "',ClassName='" +cboLop->Text + "',SubjectName='" +

cboMonHoc->Text + "',DiemHocCaiThien='" + txtDiemThi3->Text + "' where StudentID='" + txtMaSV->Text + "' ";

sqlCmd = gcnew MySqlCommand(sqlQuery, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

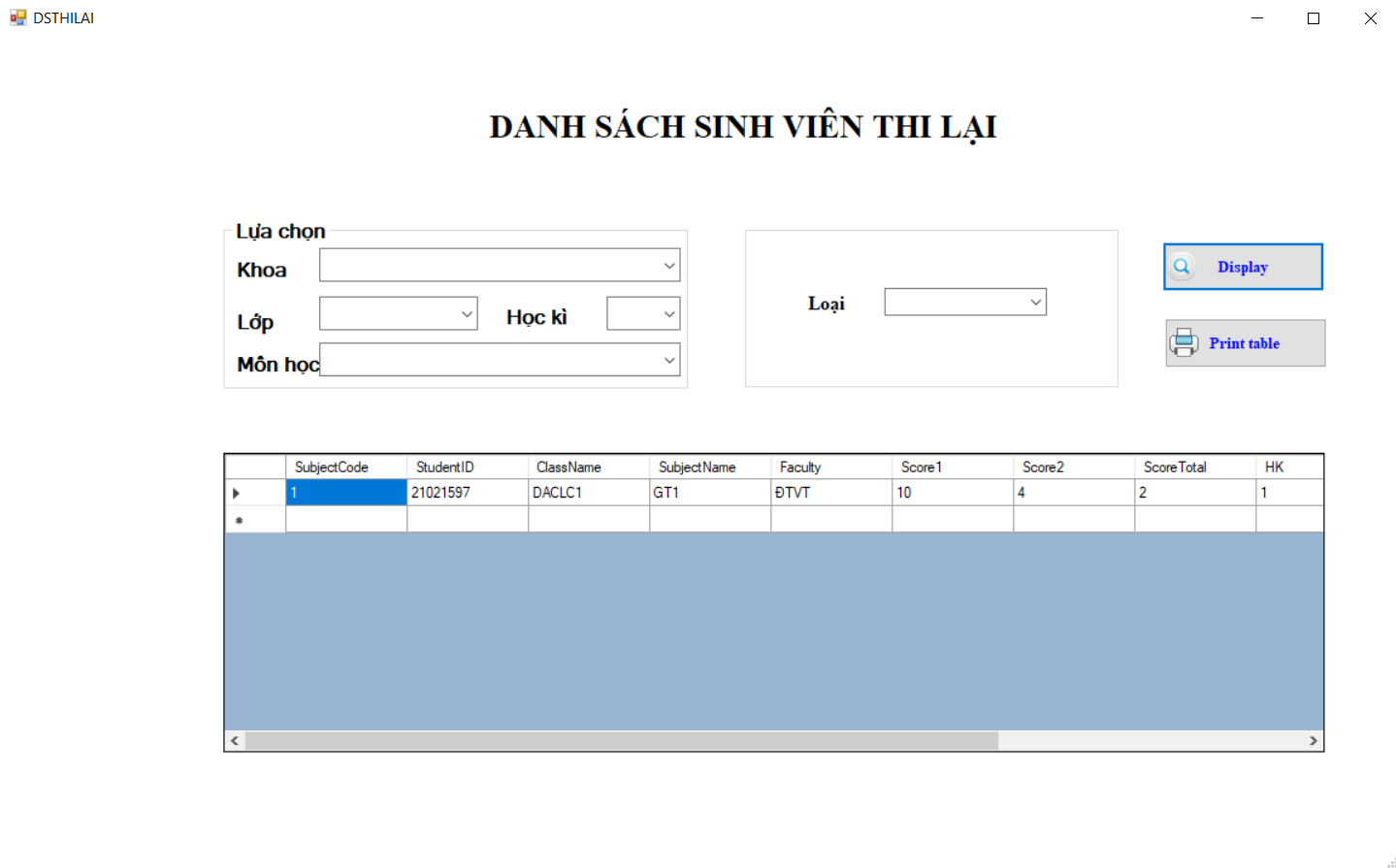
sqlConn->Close();

}

Upload();

}

* 1. **Danh sách thi lại**

****

private: System::Void Upload1() {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from hoccaithien";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewThiLai->DataSource = sqlDt;

}

private: System::Void Upload2() {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from result";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewThiLai->DataSource = sqlDt;

}

private: System::Void DSTHILAI\_Load(System::Object^ sender, System::EventArgs^ e) {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

//Add khoa vaof cbo

sqlConn1->Open();

cboKhoaHoc->Items->Clear();

cboKhoaHoc->Text = "";

String^ aaa = "Select FacultyName from faculty";

sqlCmd1 = gcnew MySqlCommand(aaa, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

while (sqlDd1->Read()) {

cboKhoaHoc->Items->Add(sqlDd1->GetString(0));

}

sqlConn->Close();

}

private: System::Void cboKhoaHoc\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

//Add lop trong khoa

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

cboLop->Items->Clear();

cboLop->Text = "";

String^ aa = "Select ClassName from class where FacultyName = '" + cboKhoaHoc->Text + "'";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

cboLop->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

//Add khoa vaof cbo

//sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

cboMonHoc->Items->Clear();

cboMonHoc->Text = "";

String^ b = "Select SubjectName from subject1";

sqlCmd = gcnew MySqlCommand(b, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

cboMonHoc->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

}

private: System::Void cboMonHoc\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn->Open();

cboHocKi->Items->Clear();

cboHocKi->Text = "";

String^ aa = "Select HK from result where SubjectName = '" + cboMonHoc->Text + "'";

sqlCmd = gcnew MySqlCommand(aa, sqlConn);

sqlDd = sqlCmd->ExecuteReader();

while (sqlDd->Read()) {

cboHocKi->Items->Add(sqlDd->GetString(0));

}

sqlConn->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn->Close();

}

}

private: System::Void btnTimKiem\_Click(System::Object^ sender, System::EventArgs^ e) {

if (cboLoai->Text == "Hoc cai thien") {

Upload1();

}

Upload2();

}

private: System::Void btnXuat\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void groupBox2\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

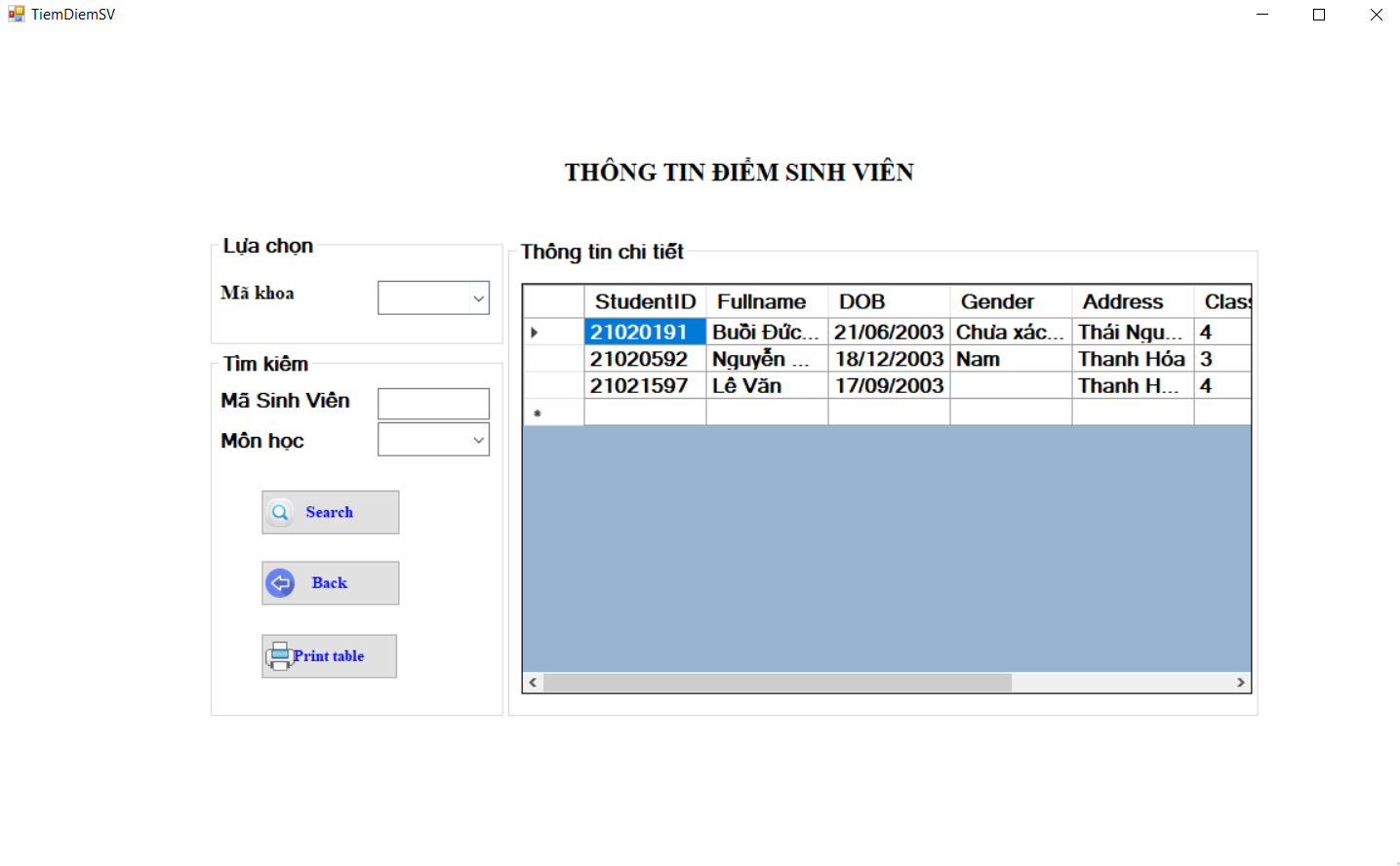
private: System::Void groupBox3\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void cboLoai\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

}

* 1. **Chức năng tìm điểm**

****

private: System::Void Upload() {

sqlConn->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn->Open();

sqlCmd->Connection = sqlConn;

sqlCmd->CommandText = "select \*from student";

sqlDd = sqlCmd->ExecuteReader();

sqlDt->Load(sqlDd);

sqlDd->Close();

sqlConn->Close();

dataGridviewDIEMSV->DataSource = sqlDt;

}

private: System::Void SearchUpload() {

sqlConn2->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

sqlConn2->Open();

sqlCmd2->Connection = sqlConn2;

sqlCmd2->CommandText = "Select \* From result where StudentID='" + txtMaSV->Text + "' and SubjectName='" + cboMonHoc->Text + "'";;

sqlDd2 = sqlCmd2->ExecuteReader();

sqlDt2->Load(sqlDd2);

sqlDd2->Close();

sqlConn2->Close();

dataGridviewDIEMSV->DataSource = sqlDt2;

}

private: System::Void TiemDiemSV\_Load(System::Object^ sender, System::EventArgs^ e) {

sqlConn1->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

//Add khoa vaof cbo

sqlConn1->Open();

cboKhoa->Items->Clear();

cboKhoa->Text = "";

String^ aaa = "Select FacultyName from faculty";

sqlCmd1 = gcnew MySqlCommand(aaa, sqlConn1);

sqlDd1 = sqlCmd1->ExecuteReader();

while (sqlDd1->Read()) {

cboKhoa->Items->Add(sqlDd1->GetString(0));

}

sqlConn1->Close();

//Add ma mon

sqlConn2->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

sqlConn2->Open();

cboMonHoc->Items->Clear();

cboMonHoc->Text = "";

String^ mon = "Select SubjectName from subject1";

sqlCmd2 = gcnew MySqlCommand(mon, sqlConn2);

sqlDd2 = sqlCmd2->ExecuteReader();

while (sqlDd2->Read()) {

cboMonHoc->Items->Add(sqlDd2->GetString(0));

}

sqlConn2->Close();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn1->Close();

}

Upload();

}

private: System::Void btnTimKiem\_Click(System::Object^ sender, System::EventArgs^ e) {

sqlConn2->ConnectionString = "server=" + server + ";" + "username=" + username + ";" + "password=" + password + ";" + "database=" + database;

try {

SearchUpload();

}

catch (Exception^ ex) {

MessageBox::Show(ex->Message);

}

finally {

sqlConn2->Close();

}

}

private: System::Void btnThoat\_Click(System::Object^ sender, System::EventArgs^ e) {

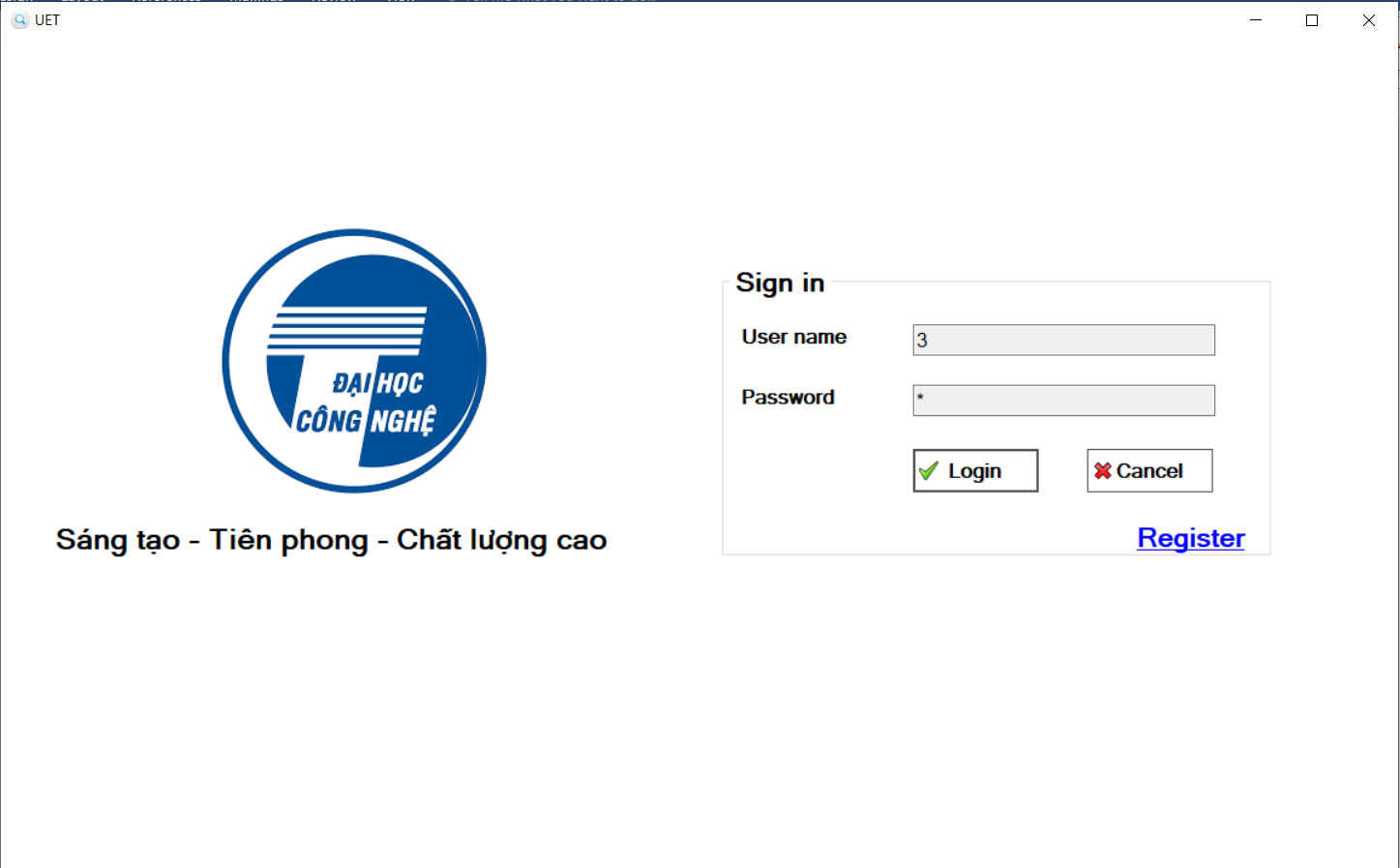
this->Hide();

}

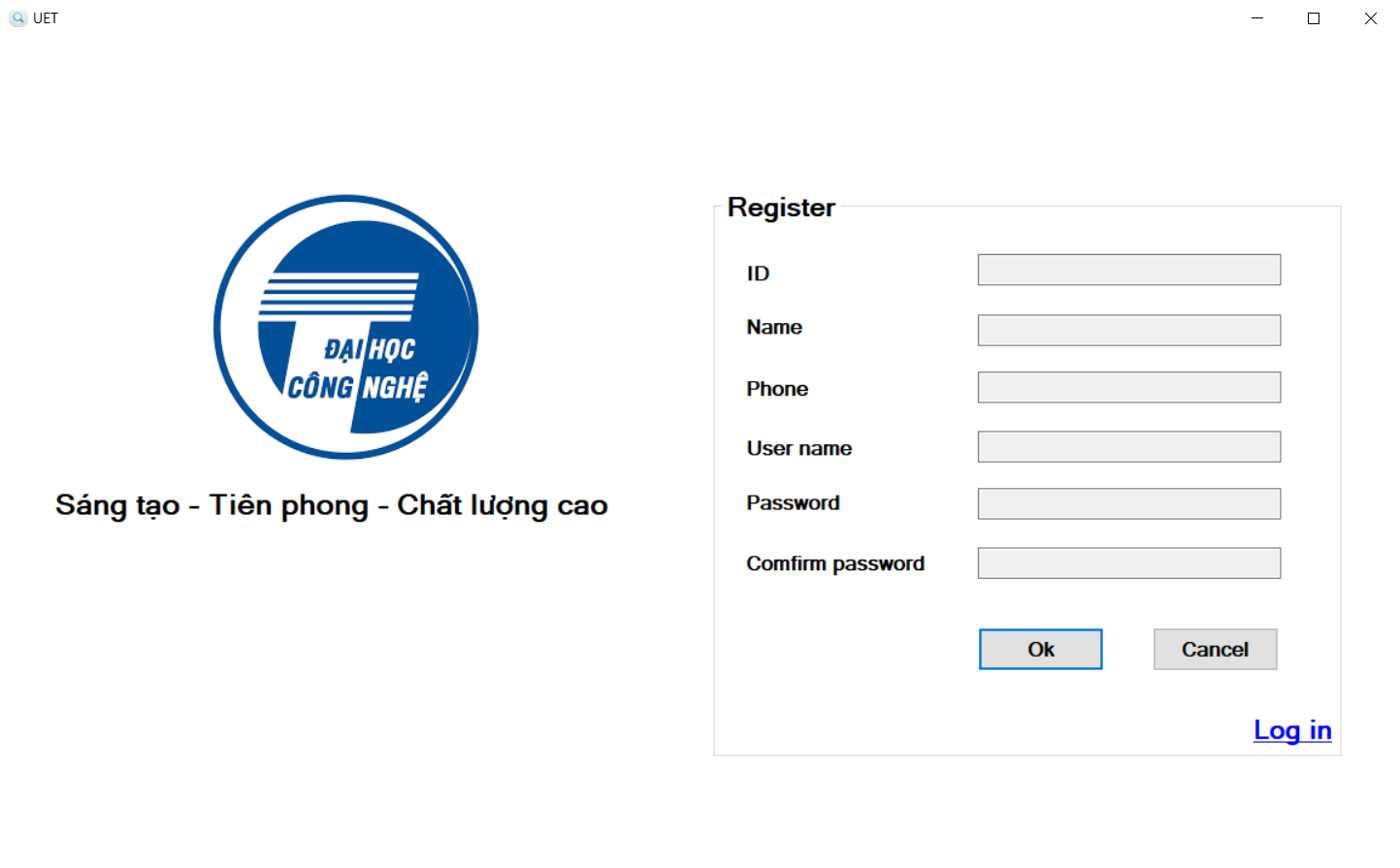
# CHƯƠNG 5: HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

Sau đây em sẽ hướng dẫn sử dụng phần mềm Quản lý điểm sinh viên như sau:

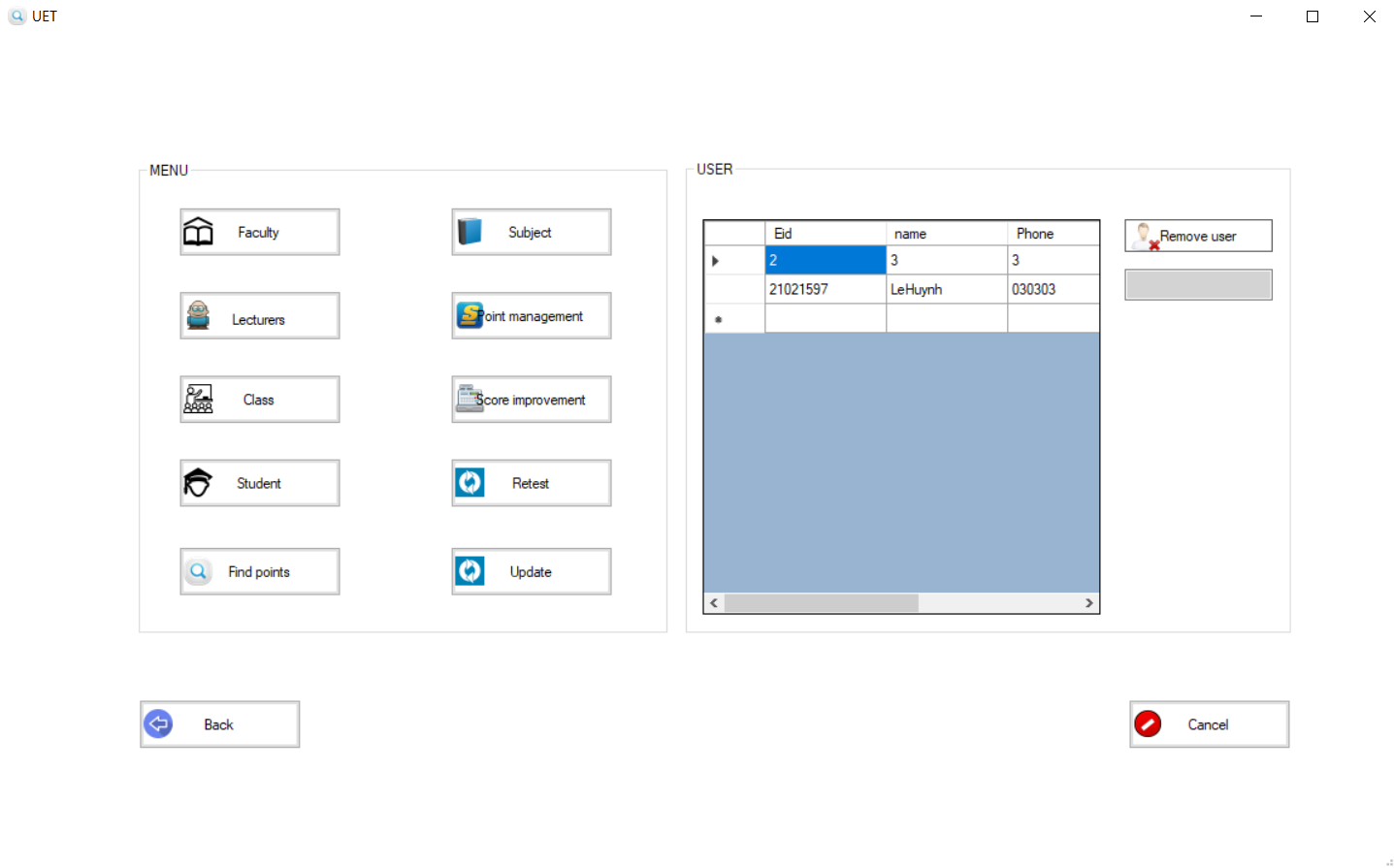
Trong phần mềm hiện tại đặt hai quyền hoạt động : admin và member. Khi người sử dụng đăng nhập hệ thống thì bắt buộc phải chọn đúng quyền của mình. Mỗi một member sẽ được hệ thống cung câp một user và password riêng:



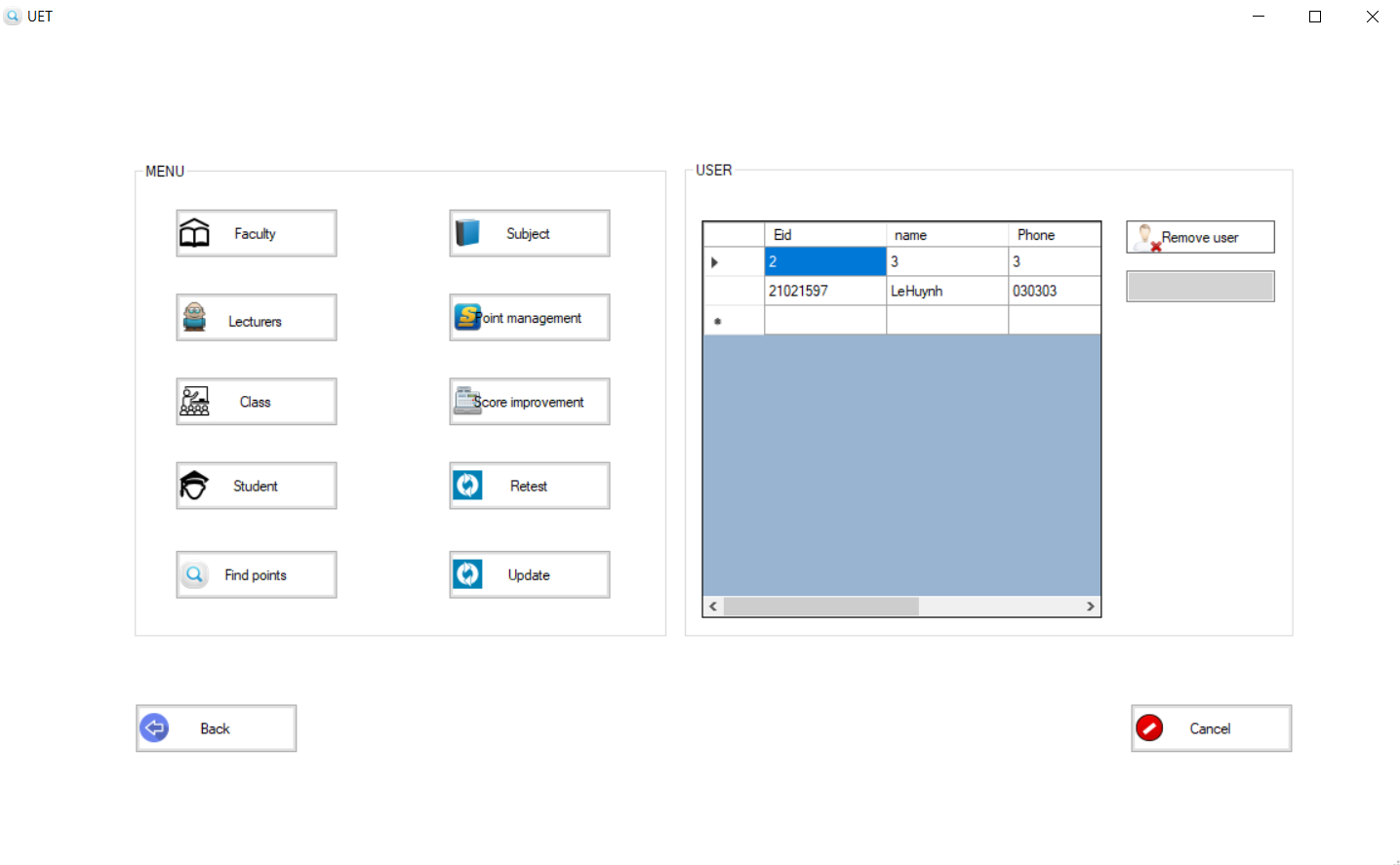
Phần đăng ký tài khoản



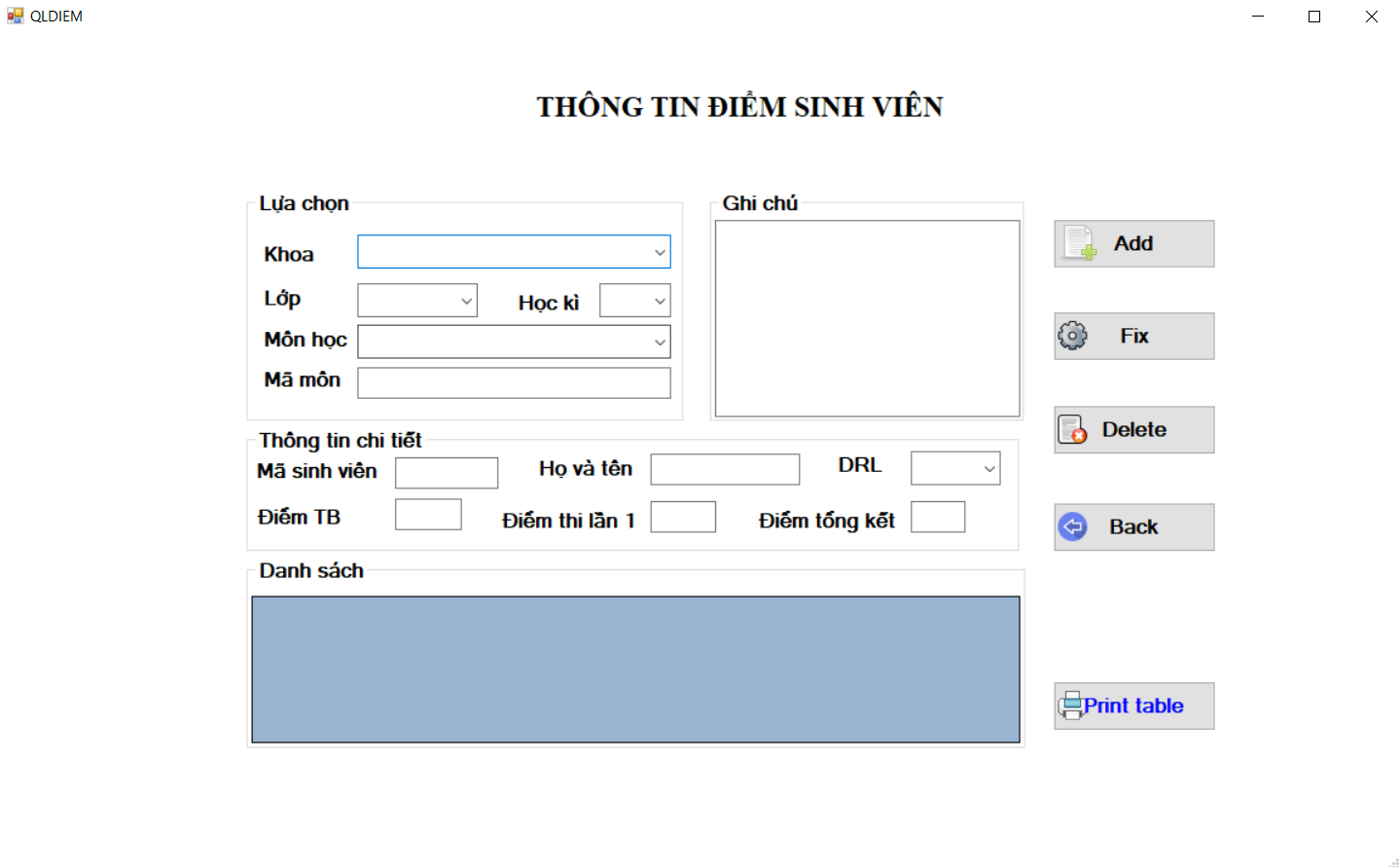
Sau khi đăng nhập thành công, giao diện chính hệ thống sẽ xuất hiện:

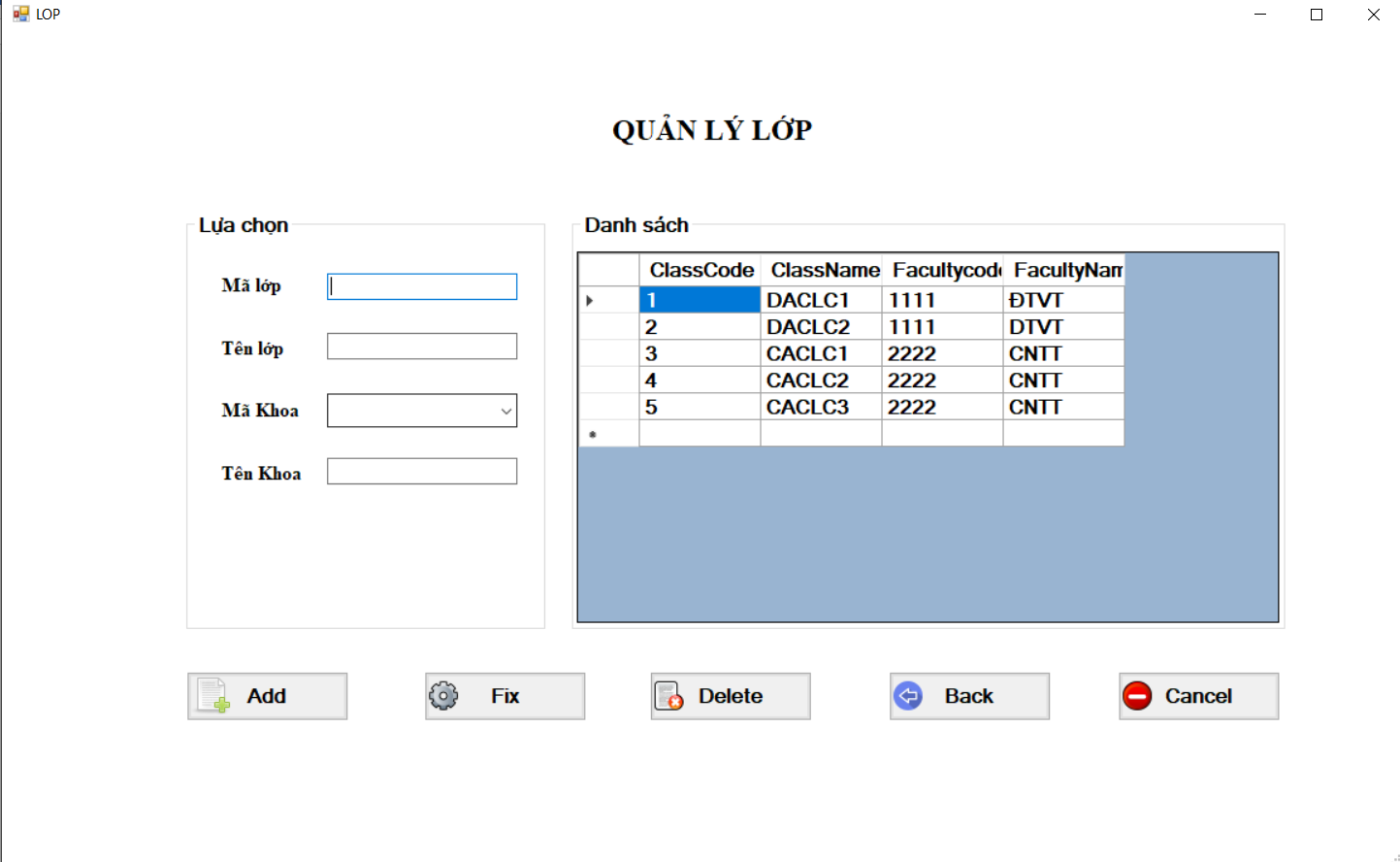


Các cửa sổ chức năng hệ thống được mở ra:



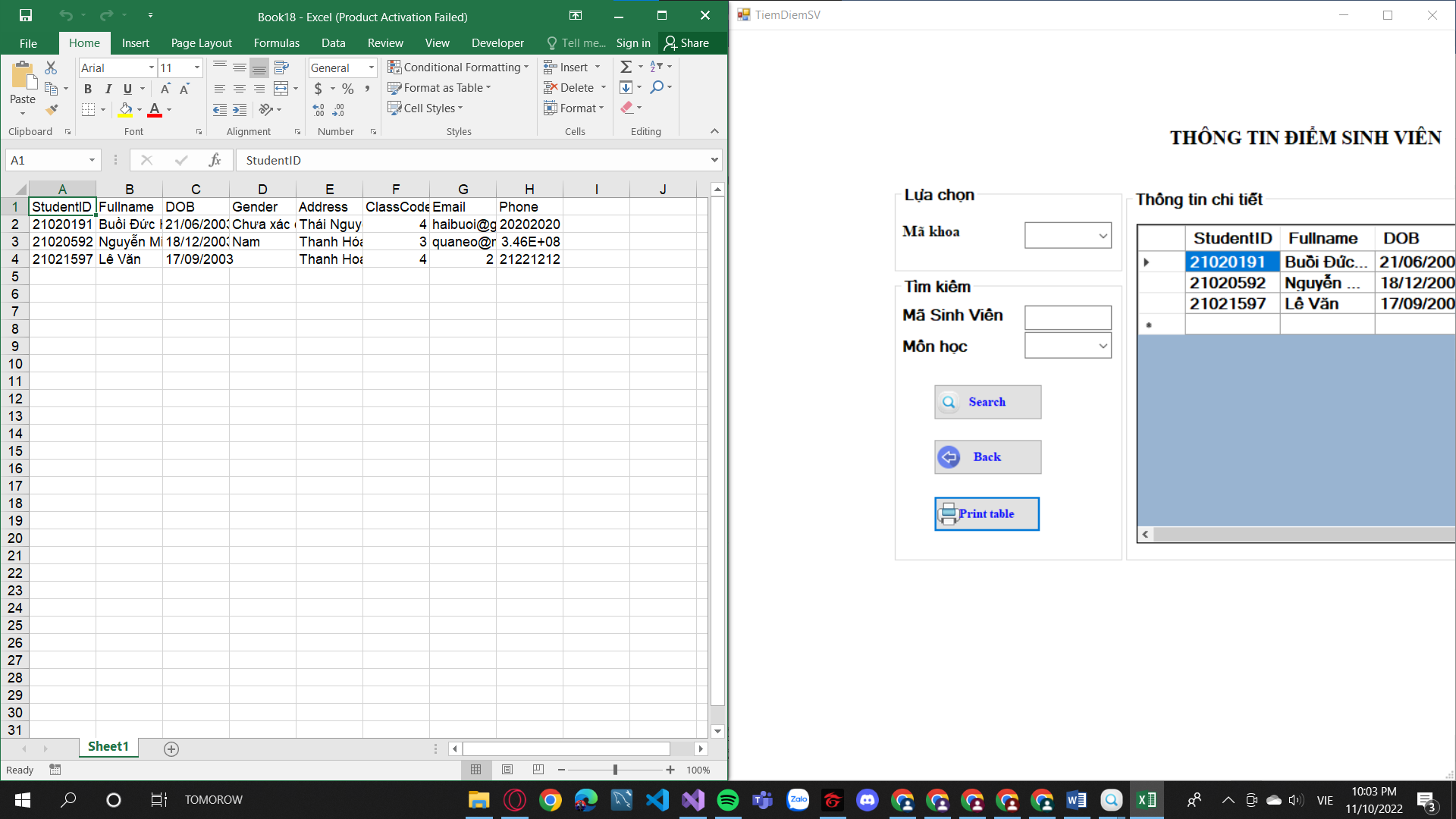
Nếu muốn truy cập danh sách, tìm kiếm, thêm hoặc sửa , xóa lớp, giảng viên, khoa, sinh viên thì click chuột vào QL Sinh Viên, QL Khoa, QL Lớp giao diện như hình sau:



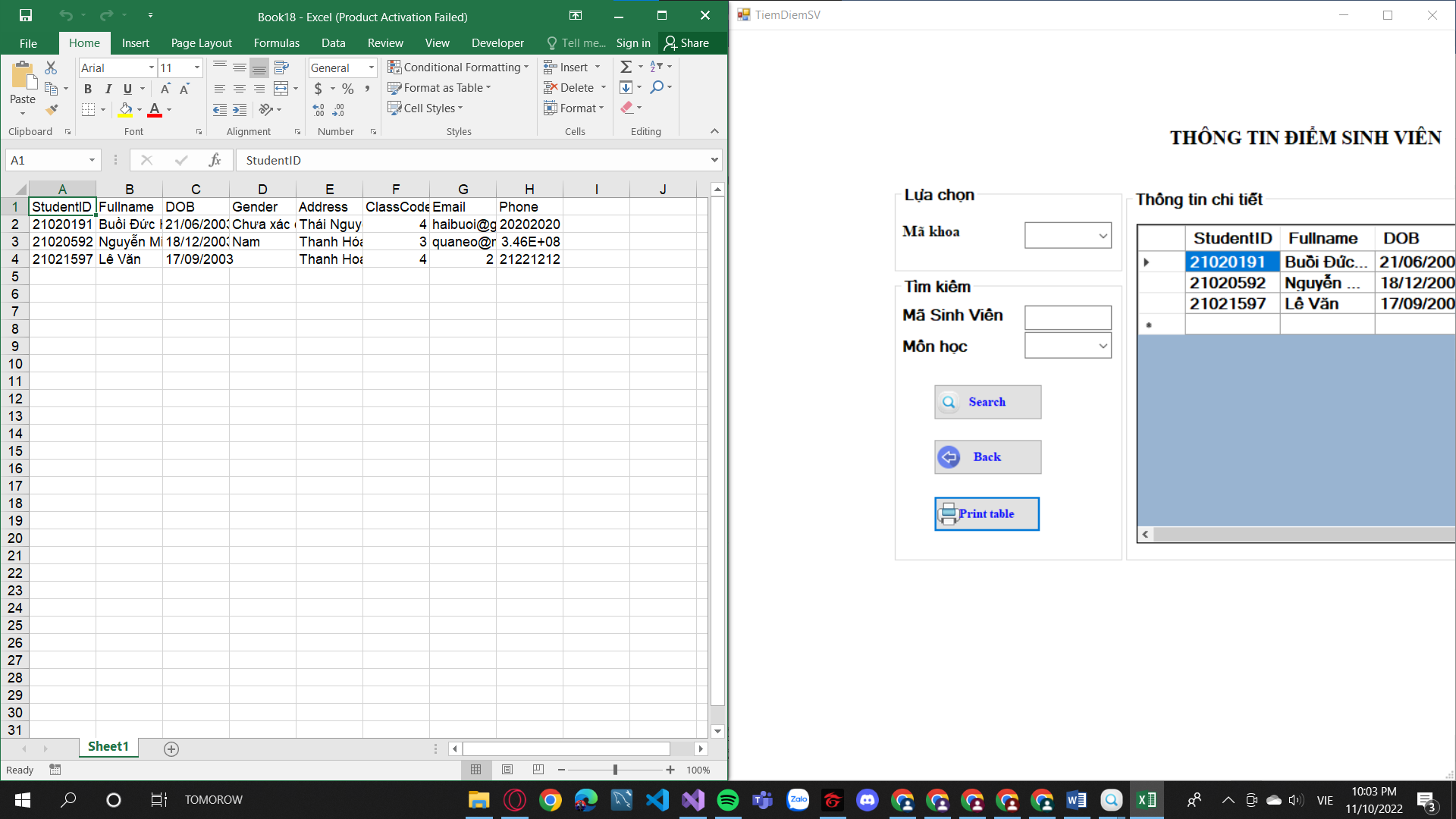


Trong quá trình sử dụng hệ thống, nếu bộ phận quản lí muốn thay đổi người sử dụng và vận hành hệ thống này thì có thể thay đổi như sau:

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 môn cơ sở dũ liệu về bài toán quản lí điểm sinh viên. 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 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, Em đã nhận được sự hỗ trợ nhiệt tình từ giáo viên hướng dẫn. Em rất mong nhận được hơn nữa những ý kiến đóng góp để hoàn thành đề tài được tốt hơn. Em xin chân thành cảm ơn Cô.

**TÀI LIỆU THAM KHẢO**

1. BÀI GIẢNG THỰC HÀNH CƠ SỞ DỮ LIỆU – KHOA CÔNG NGHỆ THÔNG TIN – TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
2. Kênh youtube freeCodeCamp.org: <https://www.youtube.com/c/Freecodecamp>.
3. Kênh youtube: <https://www.youtube.com/watch?v=huCyyUVYk3g>.
4. <https://social.msdn.microsoft.com/Forums/en-US/a9529502-6304-4aa6-90ee-0757ab258d87/create-c-windows-forms-application-in-visual-studio-2017?forum=winforms>
5. DJ Oamen: <https://www.youtube.com/c/DJOamen>