

Mục Lục

| | | |
|-----|--|----|
| I. | Giới thiệu về ứng dụng | 3 |
| 1. | Mô tả chức năng của ứng dụng | 3 |
| 2. | Cấu trúc cơ sở dữ liệu..... | 4 |
| II. | Chi tiết về các chức năng của ứng dụng | 5 |
| 1. | Form đăng nhập..... | 5 |
| a. | Giao diện | 5 |
| b. | Chức năng và mã lệnh..... | 6 |
| 2. | Form tạo tài khoản mới..... | 6 |
| a. | Giao diện | 6 |
| b. | Chức năng và mã lệnh..... | 7 |
| 3. | Form chính | 8 |
| a. | Giao diện | 8 |
| b. | Chức năng và mã lệnh..... | 9 |
| 4. | Form quản lý sinh viên | 11 |
| a. | Giao diện | 11 |
| b. | Chức năng và mã lệnh..... | 11 |
| 5. | Form quản lý môn học..... | 16 |
| a. | Giao diện | 16 |
| b. | Chức năng và mã lệnh..... | 17 |
| 6. | Form quản lý điểm | 21 |
| a. | Giao diện | 21 |
| b. | Chức năng và mã lệnh..... | 21 |
| 7. | Form xem điểm | 23 |
| a. | Giao diện | 23 |
| b. | Chức năng và mã lệnh..... | 24 |
| 8. | Form thông tin người dùng | 25 |
| a. | Giao diện | 25 |

| | |
|----------------------------------|----|
| b. Chức năng và mã lệnh..... | 25 |
| 9. From quản lý người dùng | 27 |
| a. Giao diện | 27 |
| b. Chức năng và mã lệnh..... | 28 |
| III. Kết Luận | 28 |
| 1. Nội dung đã tìm hiểu | 29 |
| 2. Kiến thức đã đạt được..... | 29 |
| IV. Tài liệu tham khảo | 29 |

I. Giới thiệu về ứng dụng.

Ứng dụng này theo dõi quá trình học tập của sinh viên trong suốt quá trình học tập tại trường, vì thời gian cho phép nên ứng dụng này đang ở phiên bản giới hạn chỉ có thể quản lý tất cả sinh viên của 3 lớp định sẵn.

Ngoài việc lưu trữ cập nhập thông tin cho từng sinh viên ứng dụng này giúp cho người dùng đánh giá, cập nhật điểm học tập của sinh viên nhanh nhất và chính xác nhất, nó sẽ giúp cho chúng ta giảm đi công sức lưu trữ sổ sách một số đáng kể.

1. Mô tả chức năng của ứng dụng.

- Hệ thống quản lý sinh viên gồm 3 bộ phận hoạt động liên kết với nhau:
- Bộ phận quản lý sinh viên:
 - ✓ Bộ phận này có nhiệm vụ quản lý hồ sơ của sinh viên trong suốt quá trình học tập tại trường.
 - ✓ Thêm sinh viên khi có sinh viên mới vào trường.
 - ✓ Cập nhật hồ sơ cho sinh viên trong suốt quá trình học tập.
 - ✓ Xóa sinh viên khi bị thôi học hoặc đã rời trường.
- Bộ phận quản lý chương trình đào tạo:
 - ✓ Bộ phận này giúp cho nhà trường có thể xây dựng chương trình đào tạo cho mỗi sinh viên đang học tại trường.
 - ✓ Lưu trữ danh sách các học phần dành cho sinh viên.
 - ✓ Cập nhật môn học.
- Bộ phận quản lý điểm:
 - ✓ Bộ phận này cập nhật điểm cho từng môn học của mỗi sinh viên khi kết thúc học kì.
 - ✓ Quản lý danh sách điểm học phần của từng sinh viên.
 - ✓ In danh sách điểm học phần của mỗi sinh viên.
- Ngoài ra ứng dụng còn có thể cho phép người dùng tìm kiếm thông tin về một sinh viên, một lớp hay một khóa học bất kỳ.

- Bên cạnh đó ứng dụng cho phép người dùng tạo một tài khoản đăng nhập mới, người dùng đăng nhập với tư cách là quản trị viên thì có thể quản lý danh sách thông tin của người sử dụng như thêm, sửa hoặc xóa...

2. Cấu trúc cơ sở dữ liệu.

- Bảng dữ liệu lưu trữ các thông tin của người sử dụng đăng nhập vào hệ thống để làm việc:


| | Column Name | Data Type | Allow Nulls |
|---|------------------|--------------|-------------------------------------|
|  | IDUser | int | <input type="checkbox"/> |
| | UserName | nvarchar(50) | <input checked="" type="checkbox"/> |
| | PhoneNumber | varchar(10) | <input checked="" type="checkbox"/> |
| | Gender | varchar(50) | <input checked="" type="checkbox"/> |
| | Email | varchar(50) | <input checked="" type="checkbox"/> |
| | Account | varchar(50) | <input checked="" type="checkbox"/> |
| | Password | varchar(50) | <input checked="" type="checkbox"/> |
| | Decentralization | varchar(10) | <input checked="" type="checkbox"/> |

Table1- dbo.Login

- Bảng dữ liệu lưu trữ các thông tin của sinh viên đang học tập tại trường:


| | Column Name | Data Type | Allow Nulls |
|---|-------------|--------------|-------------------------------------|
|  | IDStudent | varchar(10) | <input type="checkbox"/> |
| | NameStudent | nvarchar(50) | <input checked="" type="checkbox"/> |
| | Address | nvarchar(50) | <input checked="" type="checkbox"/> |
| | DateOfBirth | date | <input checked="" type="checkbox"/> |
| | Class | varchar(50) | <input checked="" type="checkbox"/> |
| | Email | varchar(50) | <input checked="" type="checkbox"/> |
| | PhoneNumber | varchar(10) | <input checked="" type="checkbox"/> |
| | Gender | varchar(50) | <input checked="" type="checkbox"/> |

Table2- dbo.Student

- Bảng dữ liệu lưu trữ thông tin về các môn học:


| | | | |
|---|---------------|--------------|-------------------------------------|
|  | IDSubject | varchar(10) | <input type="checkbox"/> |
| | SubjectName | nvarchar(50) | <input type="checkbox"/> |
| | NumberCredits | int | <input checked="" type="checkbox"/> |
| | Semester | int | <input checked="" type="checkbox"/> |

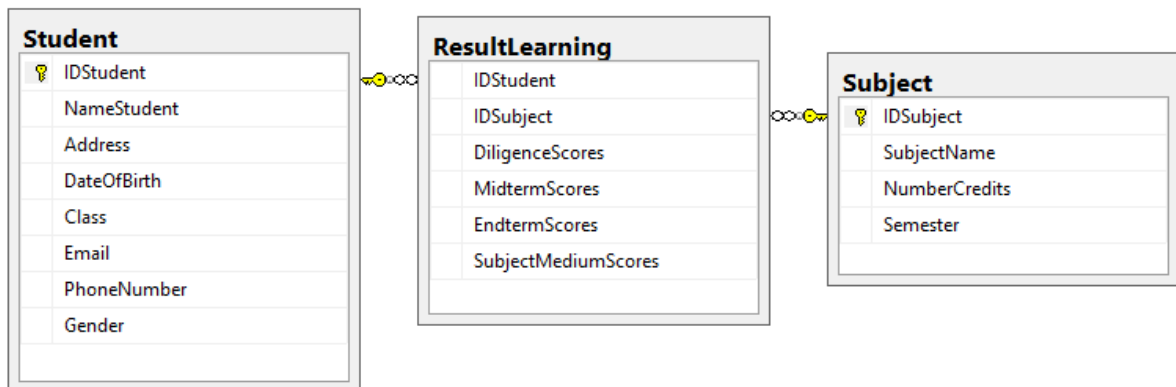
Table3- *dbo.Subject*

- Bảng dữ liệu lưu trữ điểm tất cả các học phần của mỗi sinh viên:

| Column Name | Data Type | Allow Nulls |
|---------------------|-------------|-------------------------------------|
| IDStudent | varchar(10) | <input type="checkbox"/> |
| IDSubject | varchar(10) | <input type="checkbox"/> |
| DiligenceScores | float | <input checked="" type="checkbox"/> |
| MidtermScores | float | <input checked="" type="checkbox"/> |
| EndtermScores | float | <input checked="" type="checkbox"/> |
| SubjectMediumScores | float | <input checked="" type="checkbox"/> |

Table4- *dbo.ResultLearning*

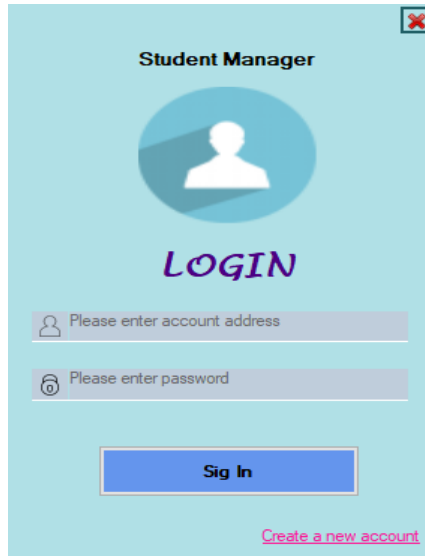
- Biểu đồ quan hệ giữa các bảng:



II. Chi tiết về các chức năng của ứng dụng.

1. Form đăng nhập.

a. Giao diện.



Hình1- frmLogin

b. Chức năng và mã lệnh.

- Button “ Sig In ” đăng nhập vào hệ thống khi người dùng click vào, khi người dùng đã nhập tên tài khoản và mật khẩu vào text box phương thức getID sẽ kiểm tra tài khoản mật khẩu có trong cơ sở dữ liệu hay không nếu khác rỗng nghĩa là tài khoản tồn tại và cho người dùng đăng nhập vào ngược lại thì thông báo tài khoản không hợp lệ.

```
private void btnSigIn_Click(object sender, EventArgs e)
{
    String acc = txtAccount.Text;
    String pass = txtPassWord.Text;
    //lấy ID người dùng
    IDUser = getID(acc, pass);
    //lấy phân quyền của người dùng
    Decentralization = getDecentralization(acc, pass);
    //lấy tên của người dùng
    NameU = getName(acc, pass);
    if (IDUser != "")
    {
        Information.ID = IDUser;
        Information.Decentralization = Decentralization;
        Information.Name = NameU;
        frmStudentManager lienket = new frmStudentManager();
        lienket.Show();
        this.Hide();
    }
    else
    {
        MessageBox.Show("--> Account invalid <--", "Please enter agian",
            MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
    }
}
```

- }
- Link lable “ Create a new account ” dùng để chuyển sang một form mới cho phép người dùng tạo một tài khoản mới khi click vào.

```
private void lnkNAccount_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    frmNewAccount nacc = new frmNewAccount();
    nacc.Show();
    this.Hide();
}
```

2. Form tạo tài khoản mới.

a. Giao diện.

Hình2- frmNewAccount

- ### b. Chức năng và mã lệnh.
- Button “ Sign Up ” cho phép người dùng tạo một tài khoản mới khi click vào, khi người dùng đã nhập dữ liệu vào text box, phương thức getID sẽ kiểm tra xem tài khoản đã tồn tại trong cơ sở dữ liệu hay chưa, bằng rỗng thì chưa có và cho phép thực hiện ngược lại thì thông báo đã tồn tại.

```

private void btnSignUp_Click(object sender, EventArgs e)
{
    string user = txtName.Text;
    string phone = txtPhone.Text;
    string email = txtEmail.Text;
    string name = txtAccount.Text;
    string pass = txtPass.Text;
    string rpass = txtRPass.Text;
    string decentralization = cbbDecentralization.Text;
    try
    {
        conn.Open();
    }
    catch (Exception)
    {
        MessageBox.Show("--> Enable to connect <--", "Sorry",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    if(decentralization != "")
    {
        //lấy tên tài khoản của người dùng vừa nhập vào đã có trong cơ sở dữ
        //liệu hay chưa
        //( bằng rỗng là chưa có và ngược lại)
        IDUser = getID(name);
        if (IDUser == "")
        {
            string sql = "INSERT into Login values(N'" + user + "','" + phone +
                "','" + gender + "','" +
                "'" + email + "','" + name + "','" + pass + "','" +
                decentralization + "')";
            SqlCommand cmd = new SqlCommand(sql, conn);
            //nếu tài khoản và mật khẩu người dùng đã nhập ở ô textbox thì thực
            //hiện
            if (name != "" && pass != "")
            {
                //nếu mật khẩu không trùng khớp với mật khẩu nhập lại lần 2 thì
                //thông báo
                if (pass != rpass)
                {
                    MessageBox.Show("--> Password incorrect <--", "Sorry",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                else
                {
                    cmd.ExecuteNonQuery();
                    MessageBox.Show("--> Created successfully <--",
                        "information", MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
            }
            else
            {
                MessageBox.Show("--> Please enter data <--", "Sorry",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else
        {
            {

```



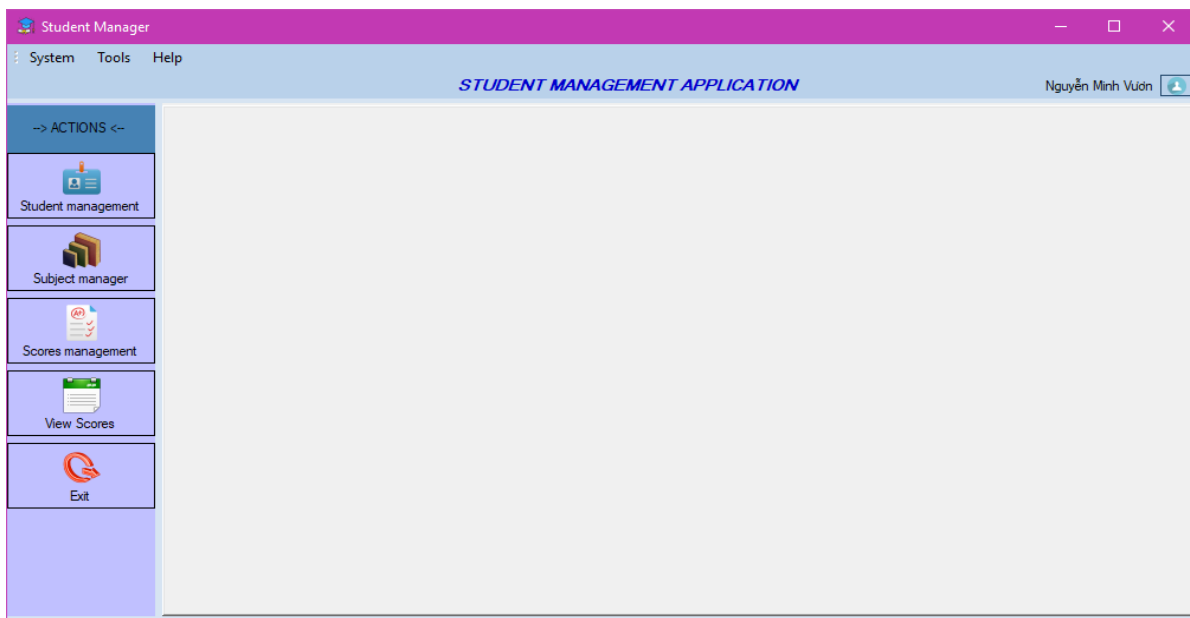
```

        MessageBox.Show("--> Account Already Exists <--", "Sorry",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        MessageBox.Show("--> Please enter decentralization <--", "Sorry",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    conn.Close();
}

```

3. Form chính.

a. Giao diện.



Hình3- frmStudentManager

b. Chức năng và mã lệnh.

- Menu strip “Information user” dùng để chuyển sang một form mới cho phép người dùng xem thông tin của tài khoản đã đăng nhập vào.

```

private void informationUserToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmUserInformation user = new frmUserInformation();
    user.Show();
    this.Hide();
}

```

- Menu strip “Log out” dùng để đăng xuất tài khoản, quay lại frmLogin..

```
private void logOutToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmLogin login = new frmLogin();
    login.Show();
    this.Hide();
}
```

- Menu strip “ Exit ” và button “ Exit ” dùng để dừng chương trình và thoát khỏi ứng dụng.

```
private void mnItemExit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

- Menu strip “ Student management ” và button “Student management ” dùng để mở một form mới cho phép người dùng quản lý danh sách các sinh viên đang học tập tại trường.

```
private void btnStudentManager_Click(object sender, EventArgs e)
{
    addTab(new frmStudent());
}
```

- Menu strip “ Subject management” và button “ Subject management ” dùng để mở một form mới cho phép người dùng quản lý các môn học và chương trình đào tạo của sinh viên.

```
private void btnSubjectManager_Click(object sender, EventArgs e)
{
    addTab(new frmSubject());
}
```

- Menu strip “ Scores management” và button “ Scores management ” dùng để mở một form mới cho phép người dùng quản lý điểm các học phần của từng sinh viên.

```
private void btnScoresManagement_Click(object sender, EventArgs e)
{
    addTab(new frmScores());
}
```

- Menu strip “ View scores” và button “ View scores ” dùng để mở một form mới cho phép người dùng xem điểm học phần của tất cả sinh viên.

```
private void btnViewScores_Click(object sender, EventArgs e)
{
    addTab(new frmViewScores());
}
```

- Menu strip “ User manager ” dùng để mở một form mới cho phép người dùng quản lý danh sách người dùng.

```
private void userManagerToolStripMenuItem_Click(object sender, EventArgs e)
{
    addTab(new frmUser());
}
```

- Menu strip “ About ” dùng để xem thông tin về ứng dụng.

4. Form quản lý sinh viên.

a. Giao diện.

Hình4- frmStudent

b. Chức năng và mã lệnh.

- Button “ Delete ” cho phép người dùng xóa một sinh viên khỏi danh sách khi đã nhập id và ô text box, phương thức getID dùng để lấy id người dùng đã nhập vào so sánh trong cơ sở dữ liệu nếu khác rỗng là đã tìm thấy và cho phép xóa ngược lại thì thông báo lỗi.

```

private void btnDelete_Click(object sender, EventArgs e)
{
    string id = txtID.Text;
    //tạo câu lệnh sql xóa dữ liệu trong bảng Student của cơ sở dữ liệu
    string sql = "DELETE FROM Student WHERE IDStudent = '" + id + "'";
    //tạo câu lệnh sql xóa dữ liệu trong bảng Student của cơ sở dữ liệu
    string sql1 = "delete from ResultLearning where IDStudent = '" + id + "'";
    SqlCommand cmd = new SqlCommand(sql, conn);
    SqlCommand cmd1 = new SqlCommand(sql1, conn);

    conn.Open();
    if (id != "")
    {
        //lấy ID sinh viên trong cơ sở dữ liệu nếu khác rỗng thì cho phép thực
        //hiện
        ID = getID(id);
        if (ID != "")
        {
            try
            {
                cmd1.ExecuteNonQuery();
                cmd.ExecuteNonQuery();
                statusBar2.Panels[0].Text = "Deleted successfully";
                hienThi();
            }
            catch (Exception)
            {
                MessageBox.Show("--> Delete failed <--", "Sorry",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("--> ID invalid <--", "Sorry", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("--> Please enter ID <--", "Sorry", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
    conn.Close();
    btnClear.Enabled = true;
}

```

- Button “Clear” dùng để xóa dữ liệu trên text box.

```

private void btnClear_Click(object sender, EventArgs e)
{
    //hiển thị thông tin lên statusbar
    statusBar2.Panels[0].Text = "Cleared successfully";
    cbbSearchClass.Text = "";
    txtSearch.Text = "";
    txtID.Text = "";
}

```

```

txtName.Text = "";
txtAddress.Text = "";
txtEmail.Text = "";
txtPhone.Text = "";
dtpckBirth.Text = "";
lstClass.Text = "";
rdoFemale.Checked = false;
rdoMale.Checked = false;
btnClear.Enabled = false;
}

```

- Button “ Save ” dùng để lưu thông tin của sinh viên vào cơ sở dữ liệu, phương thức getID dùng để so sánh id người dùng nhập vào đã có trong cơ sở dữ liệu hay chưa nếu không có thì cho phép lưu ngược lại thì thông báo lỗi.

```

private void btnSave_Click(object sender, EventArgs e)
{
    conn.Open();
    string id = txtID.Text;
    string name = txtName.Text;
    string address = txtAddress.Text;
    string dateofbirth = dtpckBirth.Text;
    string classs = lstClass.Text;
    string email = txtEmail.Text;
    string phone = txtPhone.Text;
    String sql = "INSERT into Student values('" + id + "', N'" + name + "', " +
        " N'" + address + "', '" + dateofbirth + "', '" + classs + "', '" +
        email + "', " +
        " '" + phone + "', '" + gender + "')"";
    SqlCommand cmd = new SqlCommand(sql, conn);
    if (id != "")
    {
        //lấy ID trong cơ sở dữ liệu ra để so sánh nếu bằng là đã có, thông báo
        ID = getID(id);
        if (id == ID)
        {
            MessageBox.Show("--> ID already <--", "Sorry", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
        }
        else
        {
            try
            {
                cmd.ExecuteNonQuery();
                statusBar2.Panels[0].Text = "Saved successfully";
                hienThi();
            }
            catch (Exception)
            {
                MessageBox.Show("--> Save failed <--", "Sorry",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

```

```

    }
}
else
{
    MessageBox.Show("--> Please enter ID <--", "Sorry", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
conn.Close();
btnClear.Enabled = true;
}

```

- Button “ Update ” cho phép người dùng cập nhật thông tin của từng sinh viên, khi người dùng đã nhập id vào ô text box, phương thức getID dùng để kiểm tra id người dùng nhập vào có trong cơ sở dữ liệu hay không nếu khác rỗng là có và cho phép cập nhật ngược lại thông báo lỗi.

```

private void btnUpdate_Click(object sender, EventArgs e)
{
    string id = txtID.Text;
    string name = txtName.Text;
    string address = txtAddress.Text;
    string dateofbirth = dtpckBirth.Text;
    string classs = lstClass.Text;
    string email = txtEmail.Text;
    string phone = txtPhone.Text;
    string sqlname = "UPDATE Student Set NameStudent = N'" + name + "' Where IDStudent = '" + id + "'";
    string sqladdress = "UPDATE Student Set Address = N'" + address + "' Where IDStudent = '" + id + "'";
    string sqldate = "UPDATE Student Set DateofBirth = '" + dateofbirth + "' Where IDStudent = '" + id + "'";
    string sqlclass = "UPDATE Student Set Class = '" + classs + "' Where IDStudent = '" + id + "'";
    string sqlemail = "UPDATE Student Set Email = '" + email + "' Where IDStudent = '" + id + "'";
    string sqlphone = "UPDATE Student Set PhoneNumber = '" + phone + "' Where IDStudent = '" + id + "'";
    string sqlgender = "UPDATE Student Set Gender = '" + gender + "' Where IDStudent = '" + id + "'";
    conn.Open();
    if (id != "")
    {
        //lấy ID sinh viên trong cơ sở dữ liệu nếu khác rỗng thì cho phép thực hiện
        ID = getID(id);
        if (ID != "")
        {
            SqlCommand cmdname = new SqlCommand(sqlname, conn);
            SqlCommand cmdaddress = new SqlCommand(sqladdress, conn);
            SqlCommand cmddate = new SqlCommand(sqldate, conn);
            SqlCommand cmdclass = new SqlCommand(sqlclass, conn);
            SqlCommand cmdemail = new SqlCommand(sqlemail, conn);

```

```

SqlCommand cmdphone = new SqlCommand(sqlphone, conn);
SqlCommand cmdgender = new SqlCommand(sqlgender, conn);
try
{
    //nếu người dùng nhập dữ liệu vào ô textbox thì thực hiện update
    if (name != "")
    {
        cmdname.ExecuteNonQuery();
    }
    if (address != "")
    {
        cmdaddress.ExecuteNonQuery();
    }
    if (dateofbirth != "")
    {
        cmddate.ExecuteNonQuery();
    }
    if (classs != "")
    {
        cmdclass.ExecuteNonQuery();
    }
    if (email != "")
    {
        cmdemail.ExecuteNonQuery();
    }
    if (phone != "")
    {
        cmdphone.ExecuteNonQuery();
    }
    if (gender != "")
    {
        cmdgender.ExecuteNonQuery();
    }
    statusBar2.Panels[0].Text = "Updated successfully";
    hienThi();
}
catch (Exception)
{
    MessageBox.Show("--> Update failed <--", "Sorry",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
else
{
    MessageBox.Show("--> ID invalid <--", "Sorry", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
else
{
    MessageBox.Show("--> Please enter ID <--", "Sorry", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
conn.Close();
btnClear.Enabled = true;
}

```

- Button “ Refresh ” dùng để tải lại các dữ liệu từ cơ sở dữ liệu lên data grid view.

```
private void btnRefresh_Click(object sender, EventArgs e)
{
    statusBar2.Panels[0].Text = "Refreshing...";
    btnClear.Enabled = false;
    cbbSearchClass.Text = "";
    txtSearch.Text = "";
    conn.Open();
    hienThi();
    conn.Close();
}
```

- Combo box “ Class ” dùng để hiển thị lên data grid view danh sách sinh viên theo từng lớp.

```
private void cbbSearchClass_SelectedIndexChanged(object sender, EventArgs e)
{
    string scl = cbbSearchClass.Text;
    string sql = "select * from Student where Class = '" + scl + "'";
    conn.Open();
    SqlCommand cmd = new SqlCommand(sql, conn);
    SqlDataReader dar = cmd.ExecuteReader();
    DataTable dat = new DataTable();
    dat.Load(dar);
    dgrStudent.DataSource = dat;
    statusBar2.Panels[0].Text = "Searched successfully";
    conn.Close();
}
```

- Text box “ Search ” dùng để tìm sinh viên trong danh sách.

```
private void txtSearch_TextChanged(object sender, EventArgs e)
{
    string search = txtSearch.Text;
    string sql = "select * from Student where ( NameStudent like N'" + search +
        "' or " +
        " IDStudent like N'" + search + "' or Address like N'" + search + "' " +
        ")";
    conn.Open();
    try
    {
        SqlCommand cmd = new SqlCommand(sql, conn);
        SqlDataReader dar = cmd.ExecuteReader();
        DataTable dat = new DataTable();
        dat.Load(dar);
        dgrStudent.DataSource = dat;
        statusBar2.Panels[0].Text = "Searched successfully";
    }
    catch (Exception)
    {
    }
}
```



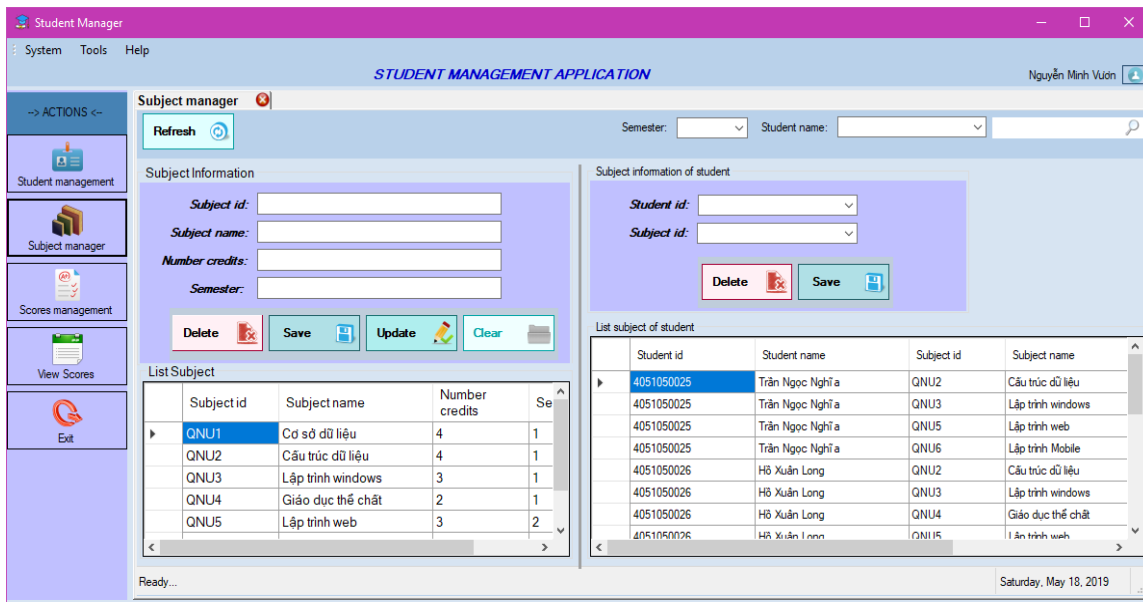
```

        MessageBox.Show("--> The name you searched for does not exist <--",
        "Sorry", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    conn.Close();
}

```

5. Form quản lý môn học.

a. Giao diện.



Hình5- frmSubject

b. Chức năng và mã lệnh.

- Những chức năng này tương tự như ở frmStudent.
- Button “ Refresh ” dùng để tải lại toàn bộ dữ liệu từ cơ sở dữ liệu lên data grid view.
- Text box “ Search ” dùng để tìm kiếm thông tin của sinh viên học thông tin của môn học có trong cơ sở dữ liệu.
- Button thuộc group box “ Subject information ”:
 - o Button “ Delete ” dùng để xóa một môn học khỏi cơ sở dữ liệu.
 - o Button “ Save ” dùng để lưu một môn học mới vào cơ sở dữ liệu.
 - o Button “ Update ” dùng để cập nhật thông tin cho môn học.
 - o Button “ Clear ” dùng để xóa dữ liệu ở ô text box.

- Button thuộc group box “ Subject information of student ”:
 - Button “ Delete ” dùng để xóa học phần của mỗi sinh viên, phương thức getIDStudent, getIDSubject dùng để lấy id của sinh viên và id của môn học trong cơ sở dữ liệu so sánh với id người dùng nhập vào nếu khác rỗng là id đã được tìm thấy cho phép xóa ngược lại thì thông báo lỗi.

```
private void btnDelete1_Click(object sender, EventArgs e)
{
    string idSubject = cbbSubjectID.Text;
    string idStudent = cbbID.Text;
    string sql = "delete from ResultLearning where" +
        " IDStudent = '" + idStudent + "' and IDSubject = '" + idSubject + "'";
    conn.Open();
    if (idStudent != "")
    {
        if (idSubject != "")
        {
            //lấy ID của sinh viên từ cơ sở dữ liệu
            string StudentID = getIDStudent(idStudent, idSubject);
            //lấy ID của môn học từ cơ sở dữ liệu
            string SubjectID = getIDSubject(idStudent, idSubject);
            //nếu khác rỗng là có trong cơ sở dữ liệu, thực hiện xóa
            if (StudentID != "" && SubjectID != "")
            {
                try
                {
                    SqlCommand cmd = new SqlCommand(sql, conn);
                    cmd.ExecuteNonQuery();
                    statusBar2.Panels[0].Text = "Deteled successfully";
                    hienThiSubjectOfStudent();
                }
                catch (Exception)
                {
                    MessageBox.Show("--> Detele failed <--", "Sorry",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                MessageBox.Show("-->ID \"\" + idSubject + "\" of ID \"\" +
                    idStudent + "\" do not exist <--", "Sorry", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("--> Please enter ID subject <--", "Sorry",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else

```

```

{
    MessageBox.Show("--> Please enter ID student <--", "Sorry",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
conn.Close();
}

```

- Button “ Save ” dùng để lưu danh sách các học phần cho mỗi sinh viên đang học tại trường, phương thức checkSubject dùng để kiểm tra học phần người dùng nhập vào đã có trong sinh viên đó chưa, nếu bằng 0 là chưa có thì cho phép lưu ngược lại thông báo lỗi.

```

private void btnSave1_Click(object sender, EventArgs e)
{
    string idSubject = cbbSubjectID.Text;
    string idStudent = cbbID.Text;
    string sql = "insert into ResultLearning(IDStudent, IDSubject)" +
        " values( '"+idStudent+"', '"+idSubject+"')";
    conn.Open();
    if (idStudent != "")
    {
        if(idSubject != "")
        {
            //kiểm tra môn học đã có trong sinh viên này chưa, bằng 0 là chưa có,
            cho thực hiện
            if(checkSubject(idStudent) == 0)
            {
                SqlCommand cmd = new SqlCommand(sql, conn);
                cmd.ExecuteNonQuery();
                statusBar2.Panels[0].Text = "Saved successfully";
                hienThiSubjectOfStudent();
            }
            else
            {
                MessageBox.Show("--> Student already have \"' + idSubject +
                    "\"subjects <--", "Sorry", MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("--> Please enter ID subject <--", "Sorry",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
    else
    {
        MessageBox.Show("--> Please enter ID student <--", "Sorry",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    conn.Close();
}

```

- Combo box “ Semester ” dùng để xem thông tin của sinh viên và môn học theo từng học kì.

```
private void cbbSemester_SelectedIndexChanged(object sender, EventArgs e)
{
    string semester = cbbSemester.Text;
    //tạo câu lệnh select lọc dữ liệu theo từng học kì
    string sql = "select * from Subject where Semester = '" + semester + "'";
    string sql1 = "select ResultLearning.IDStudent, NameStudent,
        ResultLearning.IDSubject, SubjectName," +
        " Subject.NumberCredits, Semester from ResultLearning, Subject, Student
        where" +
        " ResultLearning.IDSubject = Subject.IDSubject and
        ResultLearning.IDStudent = Student.IDStudent" +
        " and Subject.Semester = '" + semester + "'";
    conn.Open();
    if (semester != "")
    {
        SqlCommand cmd = new SqlCommand(sql, conn);
        SqlDataReader dar = cmd.ExecuteReader();
        DataTable dat = new DataTable();
        dat.Load(dar);
        dgrSubject.DataSource = dat;

        SqlCommand cmd1 = new SqlCommand(sql1, conn);
        SqlDataReader dar1 = cmd1.ExecuteReader();
        DataTable dat1 = new DataTable();
        dat1.Load(dar1);
        dgrSubjectOfStudent.DataSource = dat1;
        statusBar2.Panels[0].Text = "Searched successfully";
    }
    else
    {
        hienThi();
    }
    conn.Close();
}
```

- Combo box “ Student name ” dùng để xem danh sách các học phần của mỗi sinh viên.

```
private void cbbStudentName_SelectedIndexChanged(object sender, EventArgs e)
{
    string name = cbbStudentName.Text;
    //tạo câu lệnh sql lọc dữ liệu theo tên sinh viên
    string sql = "select ResultLearning.IDStudent, NameStudent,
        ResultLearning.IDSubject, SubjectName," +
        " Subject.NumberCredits, Semester from ResultLearning, Subject, Student
        where " +
        "ResultLearning.IDSubject = Subject.IDSubject and
        ResultLearning.IDStudent = Student.IDStudent" +
        " and Student.NameStudent = N'" + name + "'";
    conn.Open();
    SqlCommand cmd = new SqlCommand(sql, conn);
```

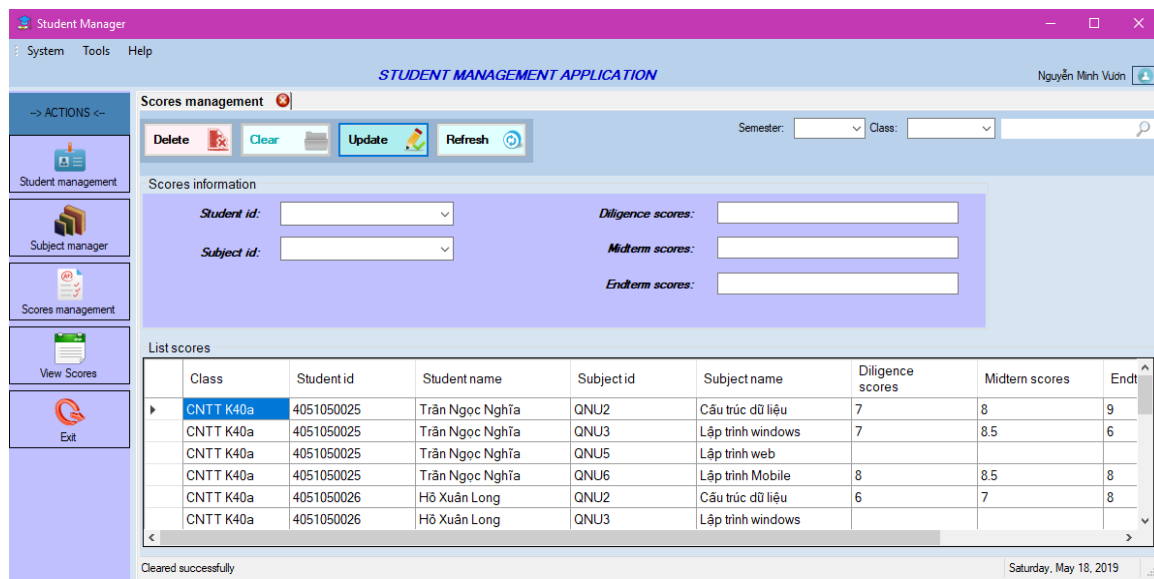
```

SqlDataReader dar = cmd.ExecuteReader();
DataTable dat = new DataTable();
dat.Load(dar);
dgrSubjectOfStudent.DataSource = dat;
statusBar2.Panels[0].Text = "Searched successfully";
conn.Close();
}

```

6. Form quản lý điểm.

a. Giao diện.



Hình6- frmScores

b. Chức năng và mã lệnh.

- Các chức năng này tương tự như ở frmSubject.
- Button “ Delete ” dùng để xóa tất cả các cột điểm của một học phần trong sinh viên nào đó, người dùng nhập id sinh viên và id môn học khi click vào chương trình sẽ tìm và so sánh trong cơ sở dữ liệu nếu khớp thì cho xóa và ngược lại thì thông báo lỗi.
- Button “ Clear ” dùng để xóa tất cả các dữ liệu trên ô text box.
- Button “ Update ” dùng để cập nhật điểm môn học cho mỗi sinh viên và chương trình sẽ tự động cập nhật cột “Subject medium scores ” bằng “ $Diligence\ scores * 0.1 + Midterm\ scores * 0.2 + Endterm\ scores * 0.7$ ”.

```

private void btnUpdate_Click(object sender, EventArgs e)
{
    string id = cbbIDStudent.Text;
    string idSubject = cbbIDSubject.Text;
    string diligence = txtDiligence.Text;
    string midterm = txtMidterm.Text;
    string endterm = txtEndterm.Text;
    conn.Open();
    if (id != "" && idSubject != "")
    {
        //lấy ID sinh viên trong cơ sở dữ liệu nếu khác rỗng thì cho phép thực
        //hiện
        IDStudent = getIDStudent(id);
        //lấy ID của môn học trong cơ sở dữ liệu nếu khác rỗng thì cho phép thực
        //hiện
        IDSubject = getIDSubject(idSubject);
        if (IDStudent != "")
        {
            if (IDSubject != "")
            {
                try
                {
                    //nếu TextBox diligence có dữ liệu thì thực hiện update
                    if (diligence != "")
                    {
                        string sqldiligence = "UPDATE ResultLearning Set
                        DiligenceScores = '" + diligence + "' " +
                        "Where IDStudent = '" + id + "' and IDSubject = '" +
                        idSubject + "'";
                        SqlCommand cmdDiligence = new SqlCommand(sqldiligence,
                        conn);
                        cmdDiligence.ExecuteNonQuery();
                    }
                    if (midterm != "")
                    {
                        string sqlmidterm = "UPDATE ResultLearning Set
                        MidtermScores = '" + midterm + "' " +
                        " Where IDStudent = '" + id + "' and IDSubject = '" +
                        idSubject + "'";
                        SqlCommand cmdMidterm = new SqlCommand(sqlmidterm, conn);
                        cmdMidterm.ExecuteNonQuery();
                    }
                    if (endterm != "")
                    {
                        string sqlendterm = "UPDATE ResultLearning Set
                        EndtermScores = '" + endterm + "' " +
                        "Where IDStudent = '" + id + "' and IDSubject = '" +
                        idSubject + "'";
                        SqlCommand cmdEndterm = new SqlCommand(sqlendterm, conn);
                        cmdEndterm.ExecuteNonQuery();
                    }
                    //tạo câu lệnh sql cập nhật điểm cho cột SubjectMediumScores
                    string sqlmedium = "UPDATE ResultLearning Set
                    SubjectMediumScores = DiligenceScores*0.1 +
                    " + MidtermScores*0.2 +EndtermScores*0.7 Where " +
                    "IDStudent = '" + id + "' and IDSubject = '" + idSubject
                    + "'";
                }
            }
        }
    }
}

```

```

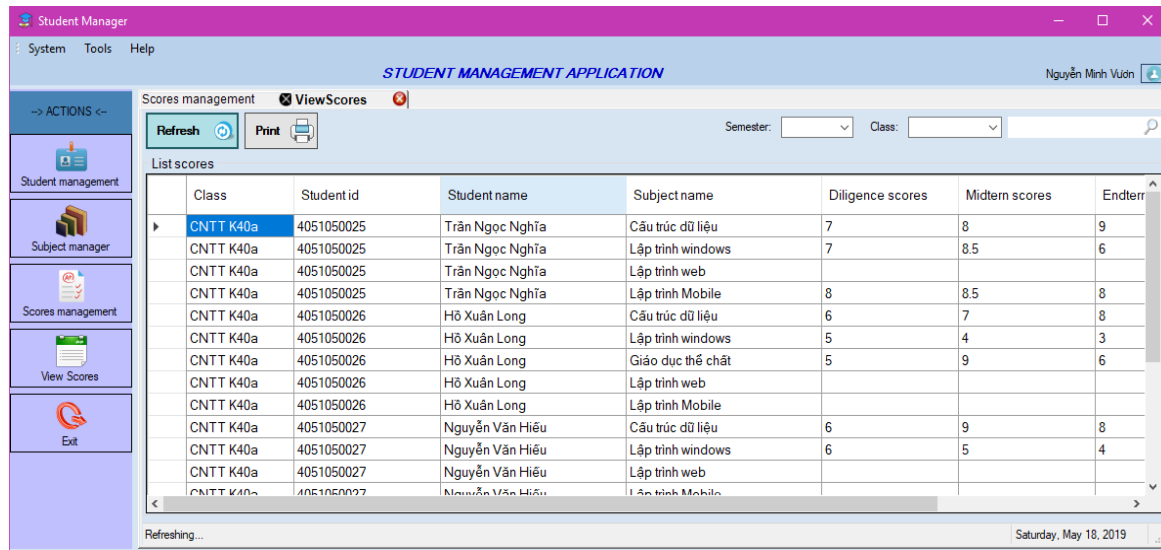
        SqlCommand cmdmedium = new SqlCommand(sqlmedium, conn);
        cmdmedium.ExecuteNonQuery();
        statusBar2.Panels[0].Text = "Updated successfully";
        hienThi();
    }
    catch (Exception)
    {
        MessageBox.Show("--> Update failed <--", "Sorry",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
else
{
    MessageBox.Show("--> ID subject invalid <--", "Sorry",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else
{
    MessageBox.Show("--> ID student invalid <--", "Sorry",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else
{
    MessageBox.Show("--> Please enter ID <--", "Sorry", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
conn.Close();
btnClear.Enabled = true;
}

```

- Button “ Refresh ” dùng để tải lại dữ liệu từ cơ sở dữ liệu lên data grid view.
- Combo box “ Semester ” dùng để xem điểm các học phần của sinh viên theo từng học kì.
- Combo box “ Class ” dùng để xem điểm của sinh viên theo từng lớp.
- Text box “ Search ” dùng tìm thông tin điểm của sinh viên trong cơ sở dữ liệu.

7. Form xem điểm.

a. Giao diện.



Hình7- frmViewScores

b. Chức năng và mã lệnh.

- Các chức năng này cũng tương tự như ở frmScores.
- Button “ Refresh ” dùng để tải lại dữ liệu lên data grid view.
- Combo box “ Semester ” dùng để xem điểm các học phần của sinh viên theo từng học kì.
- Combo box “ Class ” dùng để xem điểm các học phần của sinh viên theo từng lớp.
- Text box “ Search ” dùng để tìm kiếm sinh viên có trong cơ sở dữ liệu.
- Button “ Print ” khi click vào chương trình sẽ mở một frm mới cho phép người dùng in thông tin điểm học phần cá nhân của mỗi sinh viên, người dùng nhập id của sinh viên cần in vào ô text box “ Search ”, phương thức getID dùng để lấy id của người dùng nhập vào so sánh trong cơ sở dữ liệu nếu khác rỗng là đã tìm thấy và cho phép người dùng chuyển qua form mới ngược lại thông báo lỗi.

```
private void btnPrint_Click(object sender, EventArgs e)
{
    //lấy id của sinh viên từ cơ sở dữ liệu
    String idStudent = getID(txtSearch.Text);
    if(txtSearch.Text != "")
    {
```



```

//nếu id khác rỗng là có trong cơ sở dữ liệu, cho phép thực hiện
if (idStudent != "")
{
    Information.IDStudent = idStudent;
    frmReport rp = new frmReport();
    rp.Show();
}
else
{
    MessageBox.Show("--> ID student you searched for does not exist <--",
        "Sorry", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
else
{
    MessageBox.Show("--> Please enter ID student you want print <--",
        "Sorry", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

8. Form thông tin người dùng.

a. Giao diện.

Hình8- frmUserInformation

b. Chức năng và mã lệnh.

- Button “ Update ” dùng để cập nhật thông tin cho người dùng, ban đầu thì text box hiển thị thông tin người dùng chỉ được xem, khi người dùng click vào edit thì cho phép người dùng cập nhật.

```

private void btnUpdate_Click(object sender, EventArgs e)
{
    string name = txtName.Text;
    string phone = txtPhone.Text;
    string gender = txtGender.Text;
    string email = txtEmail.Text;
    string decentralization = txtDecentralization.Text;
    string acc = txtAccount.Text;
    string pass = txtPass.Text;
    //tạo câu lệnh sql để update cho cột UserName
    string sqlname = "UPDATE Login Set UserName = N'" + name + "' Where IDUser = '" + id + "'";
    //
    string sqlphone = "UPDATE Login Set PhoneNumber = '" + phone + "' Where IDUser = '" + id + "'";
    string sqlgender = "UPDATE Login Set Gender = '" + gender + "' Where IDUser = '" + id + "'";
    string sqlemail = "UPDATE Login Set Email = '" + email + "' Where IDUser = '" + id + "'";
    string sqldecentralization = "UPDATE Login Set Decentralization = '" + decentralization + "' Where IDUser = '" + id + "'";
    string sqlacc = "UPDATE Login Set Account = '" + acc + "' Where IDUser = '" + id + "'";
    string sqlpass = "UPDATE Login Set Password = '" + pass + "' Where IDUser = '" + id + "'";
    try
    {
        conn.Open();
    }
    catch (Exception)
    {
        MessageBox.Show("--> Enable to connect <--", "Sorry",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    SqlCommand cmdname = new SqlCommand(sqlname, conn);
    SqlCommand cmdphone = new SqlCommand(sqlphone, conn);
    SqlCommand cmdgender = new SqlCommand(sqlgender, conn);
    SqlCommand cmdemail = new SqlCommand(sqlemail, conn);
    SqlCommand cmddecentralization = new SqlCommand(sqldecentralization, conn);
    SqlCommand cmdacc = new SqlCommand(sqlacc, conn);
    SqlCommand cmdpass = new SqlCommand(sqlpass, conn);
    try
    {
        //nếu textbox đã mở thì cho thực hiện update
        if (txtName.Enabled == true)
        {
            cmdname.ExecuteNonQuery();
            MessageBox.Show("--> Updated successfully <--", "Information",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            NameUser();
        }
        if (txtPhone.Enabled == true)
        {
            cmdphone.ExecuteNonQuery();
            MessageBox.Show("--> Updated successfully <--", "Information",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            Phone();
        }
    }
}

```

```

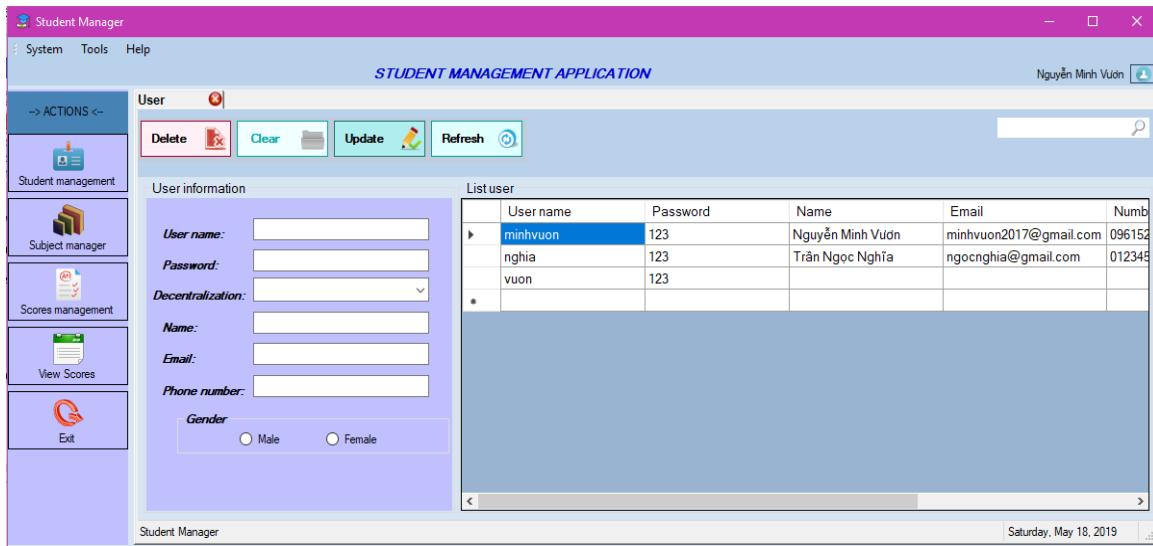
    }
    if (txtGender.Enabled == true)
    {
        cmdgender.ExecuteNonQuery();
        MessageBox.Show("--> Updated successfully <--", "Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        Gender();
    }
    if (txtEmail.Enabled == true)
    {
        cmdemail.ExecuteNonQuery();
        MessageBox.Show("--> Updated successfully <--", "Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        Email();
    }
    if (txtDecentralization.Enabled == true)
    {
        cmddecentralization.ExecuteNonQuery();
        MessageBox.Show("--> Updated successfully <--", "Information",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        Decentralization();
    }
    if (txtPass.Enabled == true)
    {
        if(txtPass.Text != "")
        {
            cmdpass.ExecuteNonQuery();
            MessageBox.Show("--> Updated successfully <--", "Information",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            Password();
        }
        else
        {
            MessageBox.Show("--> Please enter password <--", "Sorry",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
catch (Exception)
{
    MessageBox.Show("--> Not update <--", "Sorry", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
conn.Close();
}

```

- Button “Log out” đăng xuất tài khoản khỏi chương trình và quay lại frmLogin.

9. Form quản lý người dùng.

a. Giao diện.



Hình9- frmUserManger

b. Chức năng và mã lệnh.

- Các chức năng tương tự như ở frmStudent.
- Button “ Delete ” dùng để xóa một tài khoản khỏi cơ sở dữ liệu, khi người dùng nhập vào text box “ User name” cần xóa chương trình sẽ tìm và so sánh trong cơ sở dữ liệu nếu có thì cho phép người dùng xóa ngược lại thì thông báo lỗi.
- Button “ Clear ” dùng để xóa tất cả dữ liệu trên text box.
- Button “ Update ” dùng để cập nhật thông tin cho người dùng, khi người dùng đã nhập vào “ User name ”.
- Button “ Refresh ” dùng để tải lại dữ liệu từ cơ sở dữ liệu lên data grid view.
- Text box “ Search ” dùng để tìm kiếm thông tin của người dùng có trong cơ sở dữ liệu.

III. Kết luận.

- Sau thời gian học tập và tìm hiểu em đã hoàn thành bài tập cuối kì của mình đúng thời hạn. Tuy nhiên, thời gian cho phép nên ứng dụng của em còn nhiều hạn chế trong lập trình và thiếu sót trong thiết kế, em rất mong có sự góp ý của thầy để

ứng dụng được hoàn thiện và tiếp thu thêm nhiều kinh nghiệm giúp ích cho công việc sau này.

1. Nội dung đã tìm hiểu.

- Tìm hiểu về lập trình với cơ sở dữ liệu ADO.NET.
- Tìm hiểu về lập trình hướng đối tượng và không tên trong c#.
- Tìm hiểu cơ bản về Winform, ADO.NET.

2. Kiến thức đã đạt được.

- Qua bài tập lớn em trang bị cho mình được một số kiến thức về ngôn ngữ c# cũng như các kiến thức về ngôn ngữ lập trình hướng đối tượng.
- Tạo được một ứng dụng windows và thực hiện các nhiệm vụ của ứng dụng quản lý như thêm, xóa, update ...

IV. Tài liệu tham khảo.

1. Beginning Visual C# 2015 Programming with Visual Studio 2015
2. Sybex – C.Sharp Database Programming
3. Database Programming with ADO.NET – Lê Quang Hùng