**TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**MÔN CẤU TRÚC DỮ LIỆU**

---🙡🙣---

Logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

Đề tài: Project 2 - Quản Lý Nhân Sự

**Tp. Hồ Chí Minh - Năm 2022**

**Nhóm gồm các thành viên:**

Bùi Văn Hiệp 44.01.704.021

Nguyễn Nhật Anh 47.01.104.044

Phạm Anh Tùng 47.01.104.231

Nguyễn Phan Trà My 47.01.104.132

Nguyễn Thanh Viết 47.01.104.236

**Giáo viên hướng dẫn:** **Ths. Nguyễn Đỗ Thái Nguyên**

**LỜI CẢM ƠN**

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành nhất đến ThS. Nguyễn Đỗ Thái Nguyên. Trong quá trình học tập và tìm hiểu bộ môn Cấu Trúc Dữ Liệu, chúng em đã nhận được sự quan tâm giúp đỡ, hướng dẫn rất tận tình, tâm huyết của thầy. Những kiến thức mà thầy truyền tải đã giúp chúng em tích luỹ thêm nhiều kiến thức để có cái nhìn sâu sắc và hoàn thiện hơn về học phần và chuyên ngành. Việc làm đồ án đã giúp chúng em phải động não nhiều hơn, năng động và linh hoạt hơn để tìm giải pháp tối ưu, chúng em đã tìm được niềm vui, sự hứng thú và tinh thần đội nhóm khi thực hiện đồ án này, Thông qua bài báo cáo này, chúng em xin trình bày những gì đã nghiên cứu được trong quá trình xây dựng đồ án Quản Lý Nhân Sự gửi đến thầy.

Trong quãng thời gian thực hiện đồ án, dù đã rất cố gắng, một phần cũng do kiến thức còn hạn chế nên bài tiểu luận này khó tránh khỏi những sai sót. Chúng em rất mong được nhận ý kiến đóng góp của thầy để bài báo cáo của chúng em được hoàn thiện hơn.

Chúng em xin trân trọng cảm ơn!

Tp. Hồ Chí Minh, ngày 20 tháng 12 năm 2022

*Nhóm sinh viên thực hiện*

**Bùi Văn Hiệp**

**Nguyễn Nhật Anh**

**Nguyễn Thanh Viết**

**Nguyễn Phan Trà My**

**Phạm Anh Tùng**

**MỤC LỤC**

[1 Giới thiệu 1](#_Toc122464010)

[1.1 Yêu cầu 1](#_Toc122464011)

[1.2 Chức năng 2](#_Toc122464012)

[2 Phân công nhiệm vụ 3](#_Toc122464013)

[3 Sản phẩm 4](#_Toc122464014)

[3.1 Các lớp giải quyết chức năng 4](#_Toc122464015)

[3.2 Các hàm xử lý các chức năng chính 9](#_Toc122464016)

[3.3 Hướng dẫn sử dụng sản phẩm 12](#_Toc122464017)

[3.4 Kết luận 18](#_Toc122464018)

[4 Tham khảo 20](#_Toc122464019)

# Giới thiệu

## Yêu cầu

Hãy xây dựng một ứng dụng quản lý nhân sự với hai loại user: Quản trị (administrator) và nhân viên (employee). User phải đăng nhập và hệ thống để sử dụng các chức năng quản lý nhân viên, hệ thống có thể nhận ra đối tượng đăng nhập là admin hay employee và từ đó trực tiếp tới màn hình menu tương ứng với từng vai trò.

Ứng dụng sử dụng file để lưu dữ liệu của admin và employee:

1. Administrators.txt: lưu danh sách các admin (Có tối thiểu 3 người quản lý) gồm:

- Username.

- Password.

2. Employees.txt: lưu trữ danh sách các employee bao gồm:

- Username.

- Password.

3. [username].txt: lưu trữ thông tin của từng user bao gồm:

- Họ tên.

- Địa chỉ.

- Số điện thoại.

- Địa chỉ email.

Yêu cầu khác:

- Người dùng có thể lựa chọn thực hiện chức năng khác hoặc kết thúc chương trình sau khi hoàn thành mỗi thao tác/chức năng.

- Trong trường hợp có lỗi, hệ thống sẽ thông báo thông tin lỗi đến user, hủy chức năng đang thực hiện, và cho phép user chọn chức năng khác để thực hiện hoặc kết thúc chương trình.

- Mã hóa pass thành dấu \* khi đăng nhập.

- Dùng màu để thiết kế giao diện dễ nhìn.

## Chức năng

* Chức năng của quản trị (administrators) bao gồm:

**Đăng nhập Admin:** Admin chỉ được phép truy cập đến các chức năng bên dưới khi họ nhập đúng thông tin username và password, đang tồn tại trong file Admin.txt.

**Các chức năng chính của Admin:**

**Menu Admin sau khi đăng nhập thành công như sau:**

a. Thêm vào user mới bao gồm: username, password, họ tên, địa chỉ, số điện

thoại và địa chỉ email.

*- Nếu usermane trùng thì không thêm.*

*- Pass mặc định là 111111.*

*- Nếu thêm thành công thì ghi update file Employees.txt và tạo file [username].txt lưu trữ thông tin của user bao gồm: họ tên, địa chỉ, số điện thoại và địa chỉ email. (mỗi thông tin lưu một dòng).*

b. Xóa một user ra khỏi danh sách employee.

*- Nếu xóa thành công thì ghi cập nhật lại thông tin của file Employees.txt và*

*xóa file [username].txt.*

c. Tìm và hiển thị thông tin employee theo username.

d. Cập nhật lại thông tin của employee (họ tên, địa chỉ, số điện thoại và địa chỉ

email).

*- Cho người dùng tùy chọn muốn update bất kỳ thuộc tính nào.*

*- Ghi cập nhật lại thông tin file [username].txt*

e. Hiển thị thông tin toàn bộ employee.

* Chức năng của employee bao gồm:

**Đăng nhập:** User chỉ được phép truy cập đến các chức năng bên dưới khi họ

nhập đúng thông tin usename và password đang tồn tại trong file Employees.txt.

*- Nếu đăng nhập sai 3 lần thì thoát.*

*- Nếu là lần đăng nhập đầu tiên thì chương trình bắt buộc đổi mật khẩu sau*

*khi đăng nhập thành công. Ghi cập nhật file Employees.txt.*

**Chức năng của nhân viên là:**

**Menu chương trình chính sau khi đăng nhập thành công như sau:**

a. Hiển thị thông tin của employee: Hiển thị thông tin của file [username].txt

tương ứng.

b. Đổi password: Đổi pass nếu người dùng có nhu cầu từ lần đăng nhập thứ 2.

*- Trước khi nhập pass mới cần nhập mật khẩu để xác nhận chủ tài khoản.*

*- Nhập 2 lần cho pass mới để xác nhận.*

*- Ghi cập nhật file Employees.txt.*

# Phân công nhiệm vụ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **HỌ TÊN** | **CÔNG VIỆC** | **HOÀN THÀNH** |
| 1 | 44.01.704.021 | Bùi Văn Hiệp | * Code chính, chỉnh lý code, hiển thị tiếng Việt cho giao diện, tạo GUI <phần mở rộng>. * Chỉnh sửa báo cáo word. * Quay video phần hướng phát triển. | 99% |
| 2 | 47.01.104.236 | Nguyễn Thanh Viết | Test code, góp ý cải thiện đồ án. | 91% |
| 3 | 47.01.104.132 | Nguyễn Phan Trà My | Test code, quay video báo cáo đồ án. | 92% |
| 4 | 47.01.104.231 | Phạm Anh Tùng | Hỗ trợ Trà My quay video, góp ý cải thiện đồ án. | 90% |
| 5 | 47.01.104.044 | Nguyễn Nhật Anh | Test code, soạn file word báo cáo đồ án, góp ý cải thiện đồ án. | 93% |

# Sản phẩm

## Các lớp giải quyết chức năng

* Class User

Được lưu trữ tại header User.h.

class User {

protected:

string userName, passWord, name, address, phone, email;

public:

string getUserName() { return userName; }

string getPassWord() { return passWord; }

string getName() { return name; }

string getAddress() { return address; }

string getPhone() { return phone; }

string getEmail() { return email; }

void setUserName(string \_userName) { userName = \_userName; }

void setPassWord(string \_passWord) { passWord = \_passWord; }

void setName(string \_name) { name = \_name; }

void setAddress(string \_address) { address = \_address; }

void setPhone(string \_phone) { name = \_phone; }

void setEmail(string \_email) { email = \_email; }

User(string \_userName = "", string \_passWord = "111111", string \_name = "", string \_address = "", string \_phone = "", string \_email = "")

{

userName = \_userName;

passWord = \_passWord;

name = \_name;

address = \_address;

phone = \_phone;

email = \_email;

}

User(const User& u) {

userName = u.userName;

passWord = u.passWord;

name = u.name;

address = u.address;

phone = u.phone;

email = u.email;

}

~User() {}

//đọc file

void docFile( int& nUS, ifstream& fi)

{

string \_userName;

getline(fi, \_userName);

if (\_userName == "") {

return;

}

else {

this->userName = \_userName;

string \_passWord;

getline(fi, \_passWord);

this->passWord = \_passWord;

string \_name;

getline(fi, \_name);

this->name = \_name;

string \_address;

getline(fi, \_address);

this->address = \_address;

string \_phone;

getline(fi, \_phone);

this->phone = \_phone;

string \_email;

getline(fi, \_email);

this->email = \_email;

string key;

getline(fi, key);

if (key == "") nUS++;

}

}

virtual void readFile(ifstream& fileIn) {

getline(fileIn, userName, ',');

getline(fileIn, passWord);

}

//tạo file

virtual void createFile(ofstream& FileOut, char\*& fileName) {

string str = (string)fileName;//tên file

string strFile = "C://" + str + ".TXT";//đường dẫn

fileName = new char[strFile.length() + 1];//tạo một chuỗi mới

strcpy(fileName, strFile.c\_str());//coppy từ sang chuỗi đích từ chuỗi nguồn

if (fileName != NULL)// nếu không NULL sẽ mở file

{

FileOut.open(fileName, ios\_base::out);

}

delete[] fileName;//NULL là delete

}

//xóa file

virtual void deleteFile(string fileName) {

char c = 34;

string Del = "del ";

string FilePath1 = "C://";

string FilePath2 = ".txt";

string strDel = c + Del + FilePath1 + fileName + FilePath2 + c;

char\* chuoiXoa = new char[strDel.length() + 1];

strcpy(chuoiXoa, strDel.c\_str());

system(chuoiXoa);

}

//nhập thông tin

virtual void input() {

cin.ignore();

HuyInput::WriteString(L" Nhập thông tin cá nhân "); cout << endl;

HuyInput::WriteString(L" Họ tên : ");

getline(cin, name);//nhập từ bàn phím

HuyInput::WriteString(L" Địa chỉ: ");

getline(cin, address);

HuyInput::WriteString(L" Số điện thoại: ");

getline(cin, phone);

HuyInput::WriteString(L" Email: ");

getline(cin, email);

}

//đọc thông tin từng dòng

virtual void readInfomation(ifstream& fileIn) {

getline(fileIn, name);

getline(fileIn, address);

getline(fileIn, phone);

getline(fileIn, email);

}

//mở file

virtual void openFile(ifstream& fileIn, char\*& fileName) {//nếu file đã có thì mở file

string str = (string)fileName;// tên file

string strFile = "C://" + str + ".TXT";//đường dẫn

fileName = new char[strFile.length() + 1];

strcpy(fileName, strFile.c\_str());

if (fileName != NULL)

{

fileIn.open(fileName, ios\_base::in);

}

delete[] fileName;

}

};

* Class Admin

Được lưu trữ tại header Admin.h, kế thừa từ class User.

class Admin : public User

{

public:

//khởi tạo

bool operator ==(Admin a) {

return a.getUserName() + a.getPassWord() == getUserName() + getPassWord();

}

Admin(string \_userName = "", string \_Password = "111111", string \_Name = "", string \_Address = "", string \_Phone = "", string \_Email = "")

:User(\_userName, \_Password, \_Name, \_Address, \_Phone, \_Password) {}

//đọc file

void docFile(int& nAD, ifstream& fi) {

string user;

getline(fi, user);

if (user == "") {

return;

}

else {

this->userName = catChuoi(user, 6);

string pass;

getline(fi, pass);

this->passWord = catChuoi(pass, 10);

string key;

getline(fi, key);

if (key == "") nAD++;

}

}

};

* Class Employee

Được lưu trữ tại header Employee.h, kế thừa từ class User.

class Employee : public User

{

public:

bool operator ==(Employee a) {

return a.getUserName() + a.getPassWord() == getUserName() + getPassWord();

}

Employee(string \_userName = "", string \_Password = "111111", string \_Name = "", string \_Address = "", string \_Phone = "", string \_Email = "")

:User(\_userName, \_Password, \_Name, \_Address, \_Phone, \_Password) {}

//đọc file

void docFile(int& nEP, ifstream& fi) {

string user;

getline(fi, user);

if (user == "") {

return;

}

else {

this->userName = user;

string pass;

getline(fi, pass);

this->passWord = pass;

string key;

getline(fi, key);

if (key == "") nEP++;

}

}

//tạo file

void createFile(ofstream& FileOut, char\* FileName) {

User::createFile(FileOut, FileName);

}

//nhập thông tin

void input() {

User::input();

}

};

## Các hàm xử lý các chức năng chính

**Các chức năng của Admin:**

* Chức năng đăng nhập Admin:

Hiển thị màn hình đăng nhập:

+ showAdminLogin(userName, passWord, message);

Kiểm tra tài khoản và mật khẩu:

+ checkAccountAdmin(userName, passWord, nAD, admin);

* Chức năng thêm nhân viên:

+ addEmployee(user);

* Chức năng xóa nhân viên:

+ deleteEmployee(user, \_user);

* Chức năng tìm kiếm nhân viên:

+ findEmployee(user, userName);

* Chức năng cập nhật thông tin nhân viên

Cập nhật tên nhân viên:

+ updateName(user, userName);

Cập nhật địa chỉ:

+ updateAddress(user, userName);

Cập nhật số điện thoại:

+ updatePhone(user, userName);

Cập nhật email:

+ updateEmail(user, userName);

* Hiển thị thông tin toàn bộ Employee

+ showEmployees(user);

**Các chức năng của Employee:**

* Chức năng đăng nhập Employee

Hiển thị giao diện đăng nhập:

+ showEmployeeLogin(userName, passWord, message);

Kiểm tra tài khoản và mật khẩu:

+ checkAccountEmployee(employee, userName, passWord, saveUsername)

* Hiển thị thông tin của nhân viên

+ showEmployeeInfo(user, saveUsername);

* Đổi mật khẩu

Hiển thị giao diện đổi mật khẩu:

+ menuChangePass(user, saveUsername);

Thực hiện việc đổi mật khẩu:

+ changePass(user, currentPass, newPass, userName);

**Các hàm hỗ trợ:**

Hàm kiểm tra file rỗng:

template <class T>

bool isEmpty(int n, T a) {

return (n == 0);

}

Hàm lấy dữ liệu từ file:

template <class T>

void getDS(int& n, T\* array, string fileText) {

ifstream fi(fileText);

while (!fi.eof()) {

array[n].docFile(n, fi);

}

fi.close();

}

Hàm thoát chức năng đang dùng:

void exit() {

char x = \_getch();

while (x != 27) {

if (x != -32) {

SetColor(0, 12);

HuyInput::WriteString(L"Nhấn ESC để quay lại!"); cout << endl;

}

x = \_getch();

}

}

Hàm khởi tạo màu sắc:

void SetColor(int backgound\_color, int text\_color)

{

HANDLE color = GetStdHandle(STD\_OUTPUT\_HANDLE);

int color\_code = backgound\_color \* 16 + text\_color;

SetConsoleTextAttribute(color, color\_code);

}

Hàm hiển thị menu giao diện chính:

void showMainMenu(int k)

Hàm hiển thị menu giao diện Admin:

void showMenuAdmin(int k)

Hàm kiểm tra Username đã tồn tại hay chưa:

bool checkUserNameEmployee(User \* user, string \_User)

Hàm hiển thị menu cập nhật thông tin Employee:

void menuUpdate(int k)

Hàm hiển thị menu giao diện Employee

void showMenuEmployee(int k)

Hàm xoá 1 dòng được chỉ định trong file text

void del\_line(const char\* file\_name, int n)

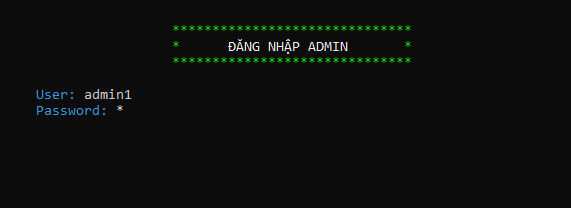
## Hướng dẫn sử dụng sản phẩm

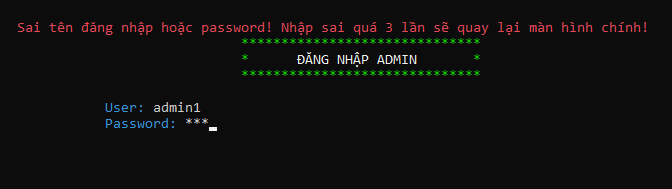
Sau khi khởi chạy chương trình thì giao diện chính sẽ hiện lên như sau:



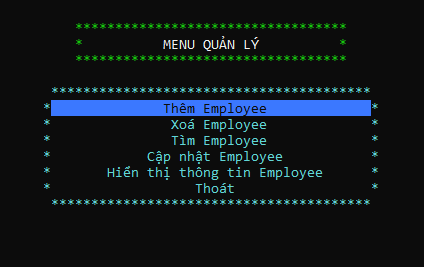
Sử dụng phím mũi tên lên xuống để di chuyển và bấm phím enter để chọn chức năng.

**Các chức năng của Admin:**

Sau khi chọn vào chức năng của quản trị viên thì sẽ hiện ra giao diện đăng nhập.

Người dùng nhập tên đăng nhập và mật khẩu sau đó bấm enter. Nếu sai tên đăng nhập sẽ hiện ra cảnh báo. Nhập sai quá 3 lần sẽ quay lại màn hình chính.

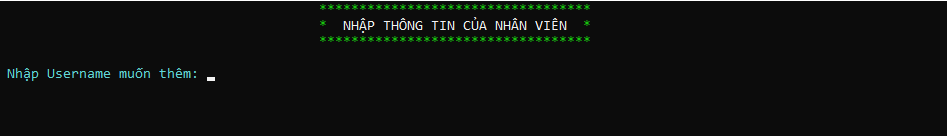
Nếu đăng nhập thành công thì sẽ hiện ra giao diện các chức năng của admin



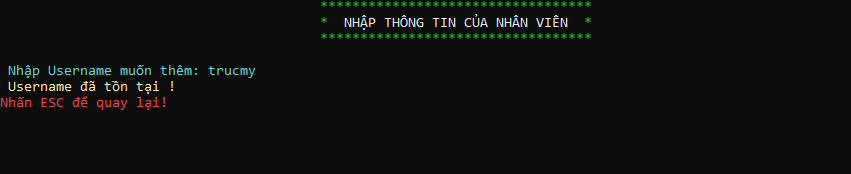
Sử dụng phím mũi tên lên xuống để di chuyển và bấm phím enter để chọn chức năng.

+ Chức năng thêm Employee:

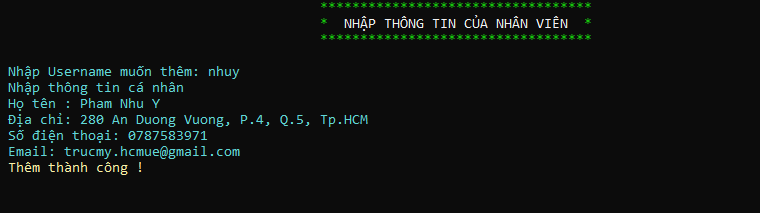
Sau khi chọn chức năng thì màn hình sẽ chuyển qua giao diện thêm Employee.



Nếu Username trùng thì sẽ hiện cảnh báo và cho người dùng quay lại.

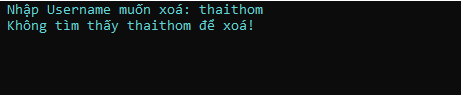


Thêm vào user mới bao gồm: username, họ tên, địa chỉ, số điện thoại và địa chỉ email. Password được mặc định là 111111. Sau khi thêm thành công sẽ hiện thông báo và cho người dùng quay lại menu. Update file Employees.txt và tạo file [username].TXT lưu trữ thông tin của user bao gồm: họ tên, địa chỉ, số điện thoại và địa chỉ email.

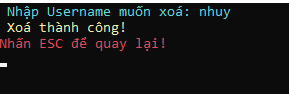


+ Chức năng xóa Enployee:

Sau khi chọn chức năng thì màn hình sẽ chuyển qua giao diện xóa Employee. Nếu Username không đúng thì sẽ hiện cảnh báo và cho người dùng quay lại.

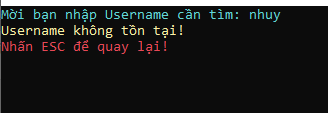


Nếu xóa thành công sẽ hiện thông báo và cho người dùng quay lại menu. Update lại file Employees.txt và xóa file [username].TXT.



+ Chức năng tìm Employee:

Sau khi chọn chức năng thì màn hình sẽ chuyển qua giao diện xóa Employee. Nếu Username không đúng thì sẽ hiện cảnh báo và cho người dùng quay lại.

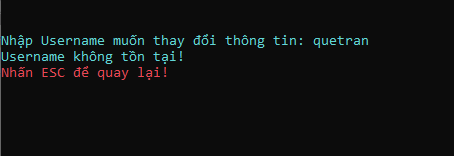


Tìm và hiển thị thông tin được lưu trong file [username].TXT.

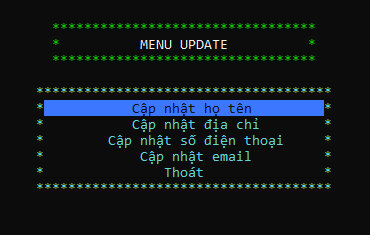


+ Cập nhật Employee:

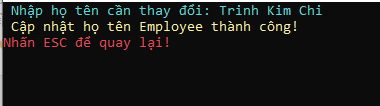
Sau khi chọn chức năng thì màn hình sẽ chuyển qua giao diện nhập Username của Employee cần cập nhật. Nếu Username không đúng thì sẽ hiện cảnh báo và cho người dùng quay lại.



Sau khi nhập đúng Username thì sẽ hiện giao diện menu lựa chọn thông tin muốn cập nhật của Employee. Cho phép người dùng cập nhật lại bất kỳ thông tin nào của một tài khoản Employee.

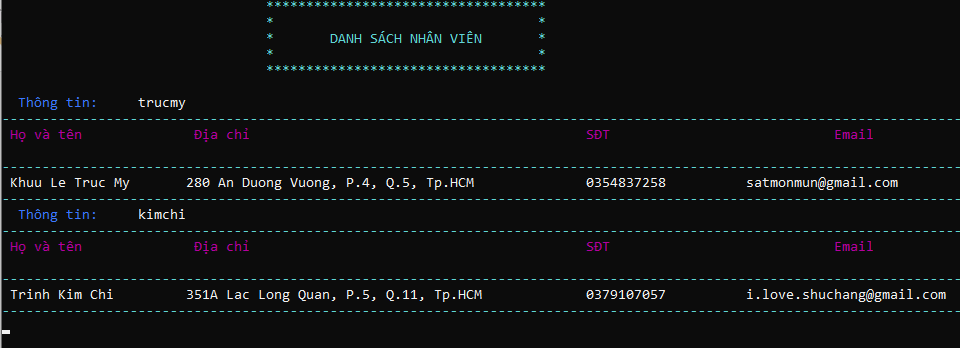


Nhập thông tin muốn thay đổi sau đó nhấn enter. Dữ liệu sẽ được update lại trong file [username.TXT].



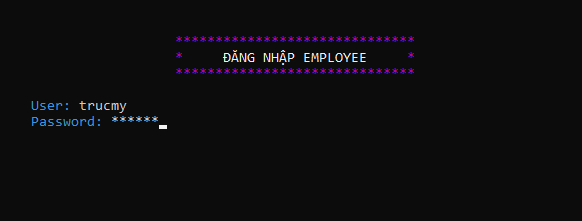
+ Chức năng hiển thị thông tin toàn bộ Employee:

Sau khi chọn chức năng thì màn hình sẽ chuyển qua giao diện hiển thị thông tin toàn bộ các Employee.

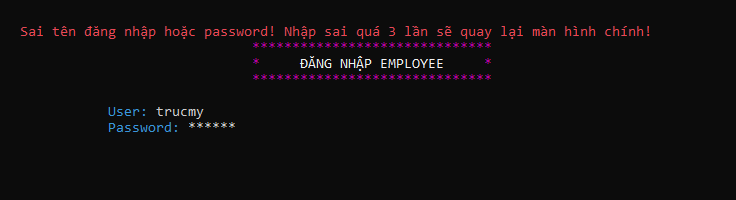


**Các chức năng của Employee:**

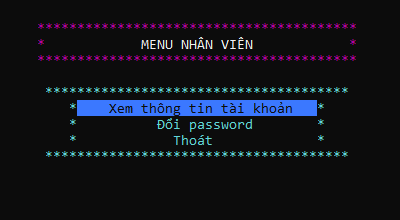
Sau khi chọn vào chức năng của nhân viên thì sẽ hiện ra giao diện đăng nhập.



Người dùng nhập tên đăng nhập và mật khẩu sau đó bấm enter. Nếu sai tên đăng nhập sẽ hiện ra cảnh báo. Nhập sai quá 3 lần sẽ quay lại màn hình chính.

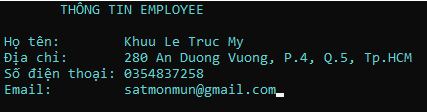


Nếu đăng nhập thành công thì sẽ hiện ra giao diện các chức năng của nhân viên



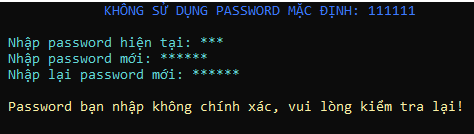
+ Chức năng xem thông tin tài khoản:

Sau khi chọn vào chức năng của nhân viên thì sẽ hiện ra giao diện hiện thị thông tin của employee từ file [username.TXT].

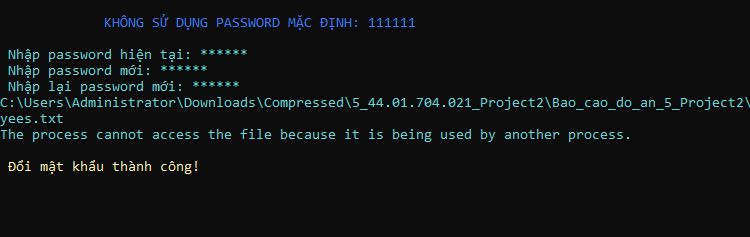


+ Chức năng đổi mật khẩu:

Sau khi chọn vào chức năng của nhân viên thì sẽ hiện ra giao diện đổi mật khẩu. Người dùng nhập mật khẩu hiện tại, mật khẩu mới, xác nhận mật khẩu mới. Mật khẩu mới không được là mật khẩu mặc định. Nếu nhập sai một trong 3 nội dung trên sẽ hiện cảnh báo và cho người dùng quay lại.



Sau khi đổi mật khẩu thành công sẽ hiện thông báo và update lại dữ liệu trong file Employees.txt.



## Kết luận

* **Ưu điểm:**

Các chức năng cơ bản của chương trình đã hoàn thành. Giao diện tiếng Việt thân thiện giúp người dùng dễ sử dụng.

* **Hạn chế:**

Do hạn chế về kiến thức chuyên môn và kỹ năng lập trình nên việc nghiên cứu và giải quyết các vấn đề phát sinh làm tốn nhiều thời gian, dẫn đến chương trình tuy đáp ứng các yêu cầu cơ bản nhưng vẫn gặp nhiều lỗi nhỏ, nhiều tính năng chưa được tối ưu, chưa kịp cài đặt các tính năng mới theo ý đồ của người lập trình.

* **Kinh nghiệm thu được:**

Có thêm kinh nghiệm trong việc kiểm thử phần mềm. Nắm vững kiến thức lý thuyết cơ bản phục vụ tốt cho việc thiết kế chương trình.

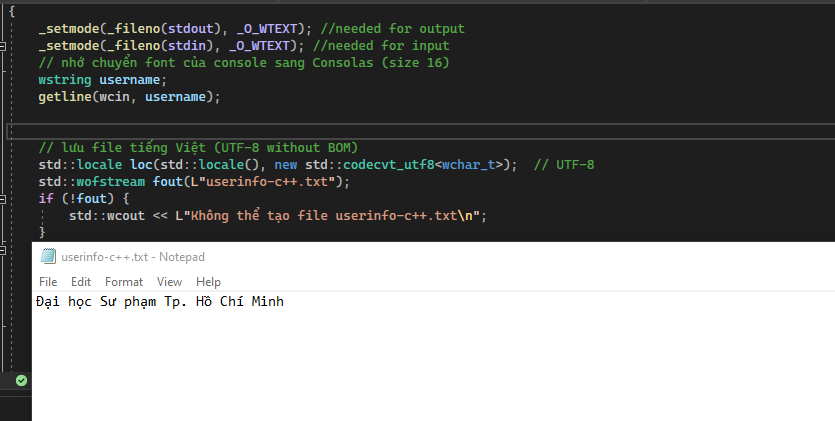
Cải tiến thành công cách xoá thông tin nhân viên theo dòng, với cách thông thường được nhiều nhóm sử dụng (cũng được chia sẻ nhiều trên mạng) thì nếu không cài code khéo sẽ dẫn tới vấn đề chỉ xoá username nhưng password vẫn được giữ lại ở dòng đó, sẽ gây xung đột khi đọc file ở những lần sau, dẫn đến phần mềm không thể đọc đúng thông tin nhân viên. Nhóm đã chọn hướng tiếp cận là viết thêm 1 hàm xoá dòng, khi xác định được username cần xoá ở dòng thì thì chạy hàm xoá dòng và dừng chương trình liền, vì bản chất username sẽ không có 2 username giống hệt nhau, nên chỉ cần tìm thấy 1 cái là dừng tìm kiếm.

Giải quyết được vấn đề bị lỗi tạo thêm dòng chứa password mặc định ở dòng cuối cùng của file Employees.txt khi thay đổi password, nhóm chọn cách tìm kiếm substring của từng dòng trong file text, nếu phát hiện dòng nào có 2 ký tự đầu là “, “ thì sẽ xoá dòng đó. Tuy nhiên khi xoá dòng như vậy thì dòng cuối của file text sẽ không có khoảng trắng, dẫn tới xung đột khi tạo tài khoản mới ở những lần sử dụng sau. Nhóm chọn giải pháp thêm 1 dòng trắng vào dòng cuối cùng của file text.

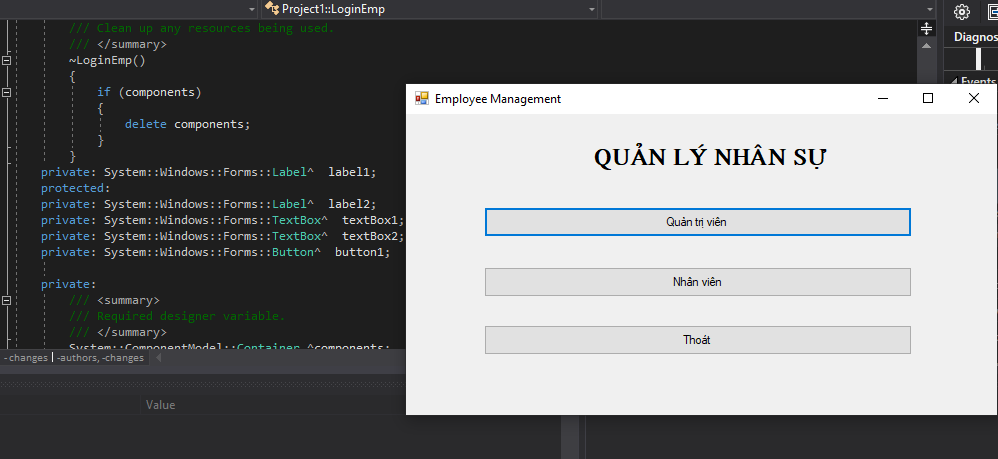
Giải quyết được vấn đề giao diện tiếng Việt có dấu khi hiển thị trên màn hình console, thông thường mọi người thường bị bó hẹp trong tư duy là console của C++ khó hiển thị tiếng Việt và cũng không bỏ thời gian đi tìm hiểu, cho nên đa phần các nhóm sẽ lựa chọn là hiển thị tiếng Việt không dấu để đỡ tốn thời gian. Nhóm chúng em đã tìm thấy trên mạng 1 thư viện hỗ trợ hiển thị tiếng Việt có dấu trên màn hình console. Thông qua quan sát và cảm nhận chủ quan cá nhân, chúng em tự tin rằng trong số các nhóm nộp đồ án của học kỳ này và kể cả trước đó, nhóm chúng em sẽ là 1 trong số những nhóm hiếm hoi sử dụng giao diện tiếng Việt có dấu, thậm chí là nhóm duy nhất (trừ khi source code của nhóm bị lộ ra trong quá trình phát triển, hoặc nhóm khác cũng tìm hiểu vấn đề tiếng Việt có dấu). Trong quá trình phát triển, nhóm nhận thấy việc gõ và hiển thị tiếng Việt có dấu sẽ thuận lợi hơn khi sử dụng Visual Studio để soạn thảo và biên dịch.

Củng cố lại các kiến thức đã được học, đặc biệt là kỹ năng phân tích, giải quyết vấn đề. Biết cách áp dụng các kiến thức lý thuyết vào ứng dụng thực tế. Rèn luyện được kĩ năng làm việc nhóm, phân chia, sắp xếp công việc để hoàn thành nhiệm vụ được giao một cách hiệu quả nhất.

* **Hướng phát triển**

****

Thêm tính năng thêm thông tin nhân viên bằng tiếng Việt. Hiện nhóm đã có thể gõ tiếng Việt và ghi ra file txt ở 1 project riêng rẽ, tuy nhiên khi tích hợp vào project hiện tại thì chưa đạt được mục đích mong muốn, nếu có thêm thời gian thì có lẽ sẽ giải quyết được.



Ngoài ra nhóm còn nghĩ đến hướng thiết kế giao diện đồ hoạ (GUI) vẫn sử dụng C++ trên Visual Studio với tuỳ chọn C++/CLR. Tuy nhiên trong quá trình tìm hiểu, nhóm phát hiện ra đây không phải là 1 ngôn ngữ thuần C++, nó có cú pháp khác với C++, dường như được lai giữa C++ và C#, có thể vẫn tiếp cận và code danh sách liên kết kiểu C++ được nhưng cần nhiều thời gian để nghiên cứu, mà với vốn thời gian eo hẹp của nhóm thì không đủ để làm việc đó. Tuy nhiên, chúng em cũng đã đạt được 1 số thành quả nho nhỏ khi tiếp cận hướng nghiên cứu này, chúng em đã tạo được giao diên đăng nhập cơ bản và có thể đọc file text khi click vào 1 button cụ thể. Trong tương lai, khi có thời gian hoặc cần giải quyết 1 vấn đề tương tự, có lẽ nhóm sẽ tiếp tục nghiên cứu sâu hơn hướng tiếp cận này.

# Tham khảo

Overview of Data Structures - GeeksForGeeks: <https://www.geeksforgeeks.org/data-structures/>

Data Structure Tutorial - W3schools:

<https://www.w3schools.in/data-structures/tutorials/>

C++ OOP (Object-Oriented Programming) - W3Schools: <https://www.w3schools.com/cpp/cpp_oop.asp>

Thư viện nhập xuất tiếng Việt trong C++

<https://github.com/codelaghien/HuyInput>

Tạo giao diện đồ hoạ (GUI) cho C++

<https://www.youtube.com/watch?v=LF1cI7zeFm4>

Code xuất chuỗi tiếng Việt vào file text

<https://github.com/tntxtnt/VietnameseIO>