**BÁO CÁO BÀI TẬP LỚN**

Môn học: Kỹ thuật lập trình

Mã lớp: 95107

Năm học: 20162

Đề tài: 1

Nhóm: 19

Thành viên: Phạm Sỹ Bằng – 2015 5119

Lê Sĩ Bích – 2015 5125

Trịnh Bá Huy – 2015 5712

Phạm Hùng Sơn – 2015 6403

Nguyễn Hữu Thắng – 2015 6500

**CHƯƠNG 1: TỔNG QUAN**

1. **Giới thiệu về hệ thống**

Chúng ta đang sống trong một thời đại mới, thời đại phát triển rực rỡ của CNTT. CNTT đã ở một bước phát triển mạnh mẽ ở mọi nơi, mọi lĩnh vực trong đời sống xã hội. Hòa chung với xu thế đó, các doanh nghiệp luôn không ngừng áp dụng các công nghệ mới nhất, tiên tiến nhất vào quản lí, cũng như sản xuất nhằm đạt hiệu quả tốt nhất về năng suất cũng như chất lượng cho doanh nghiệp của mình.

Trước đây khi CNTT chưa phát triển, một bài toán lớn đặt ra với các doanh nghiệp là

làm sao để quản lí doanh nghiệp một cách dễ dàng mà lại đạt hiệu quả cao. Để trả lời cho bài toán trên, nhóm 19 chúng em đã nghiên cứu và cho ra một phần mềm quản lí nhân viên nhằm giúp các doanh nghiệp tháo gỡ được những khó khăn đã và đang đối mặt về quản lí.

1. **Định hướng công nghệ, công cụ**

**Ngôn ngữ lập trình**

* C
* C++

**Công cụ phát triển:**

* IDE Dev C++
* Text editor Atom, Sublime Text
* GitHub (quản lí project của nhóm):

Link project: https://github.com/moonlight8978/project-ktlt-20162

* Facebook Messenger (nhóm chat, liên lạc giữa các thành viên)

**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ**

1. **Khảo sát**

**Mô tả yêu cầu:**

Đầu vào: File input.txt với một cấu trúc cho trước:

Số thứ tự (mã số nhân viên, bắt đầu bằng ký tự NV)

Họ

Tên

Đơn vị

Chức vụ

Ngày tháng năm sinh

Quê quán

Địa chỉ

Email

Số điện thoại

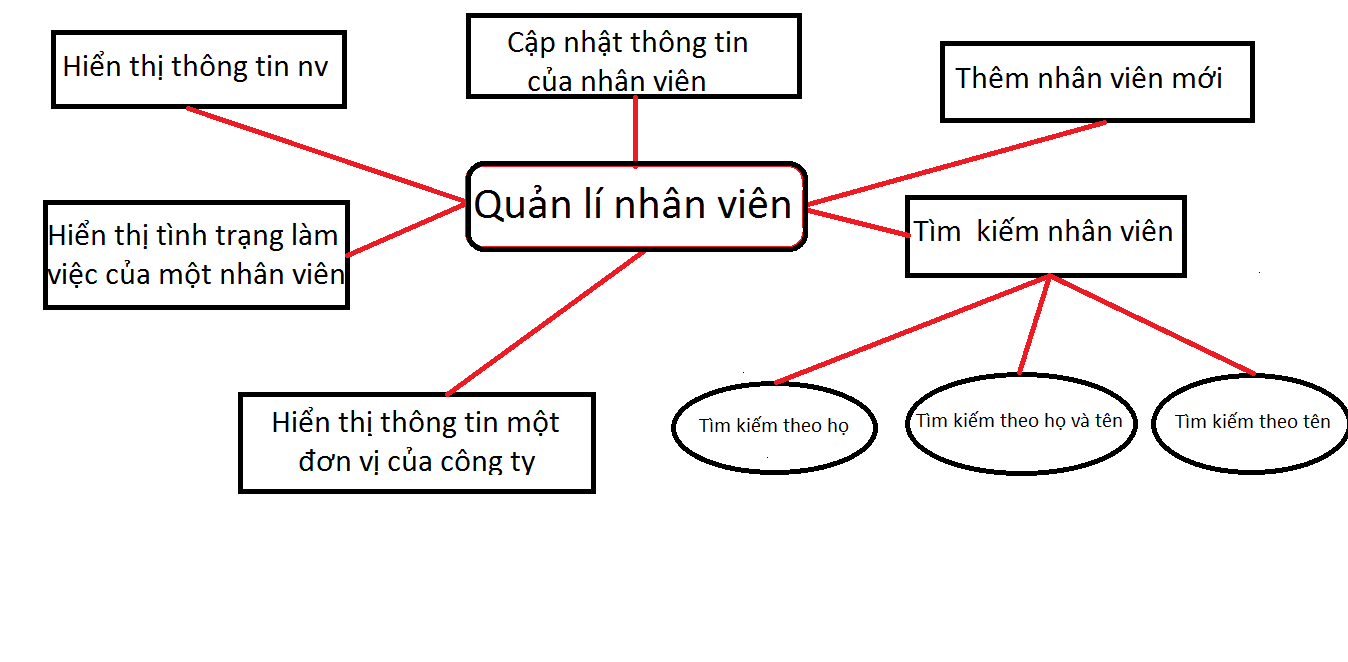
Ngày bắt đầu làm việc

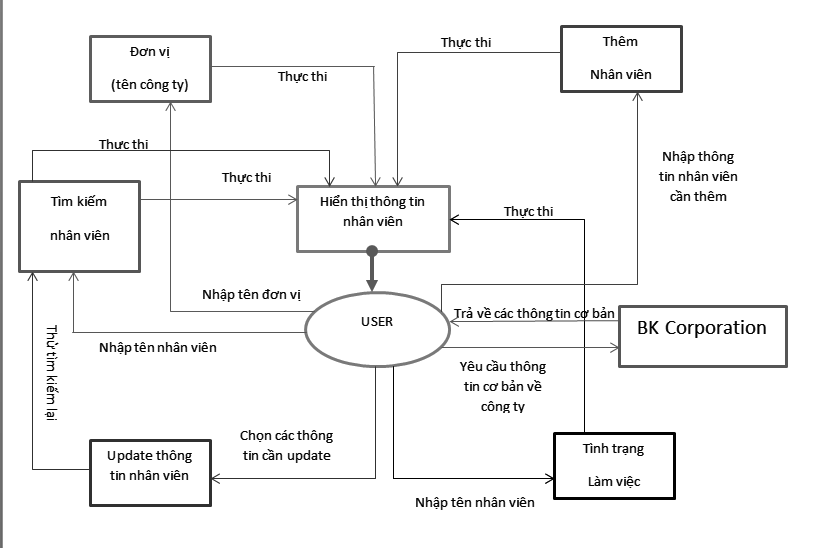
Ngày làm việc, thời gian đến, thời gian về

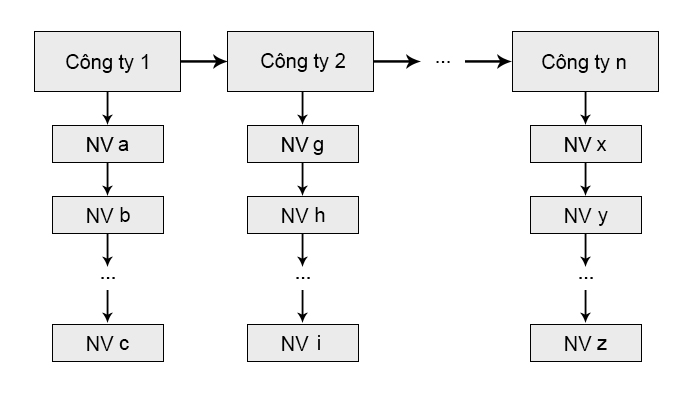
Đầu ra: Thực hiện được các thao tác

* Hiển thị các thông tin cơ bản về BKCorporation (Tên chủ tịch, tên các công ty con và giám đốc, phó giám đốc tương ứng, tên các phòng ban và trưởng, phó phòng tương ứng, tổng số nhân viên của công ty).
* Tìm kiếm thông tin của nhân viên: Nhập vào một chuỗi tìm kiếm, đưa ra tất cả các nhân viên có họ và tên tương ứng và các thông tin cơ bản về người đó.
* Hiển thị tình trạng làm việc của nhân viên: Nhập vào mã nhân viên, đưa ra thông tin làm việc của nhân viên trong tháng hiện tại. Tính số giờ thiếu hụt của nhân viên (đi muộn, về sớm. Làm tròn lên 1 giờ. Không tính thứ 7 và chủ nhật và các ngày lễ)
* Hiển thị thông tin của một đơn vị: Nhập vào một đơn vị (công ty con hoặc phòng, ban), liệt kê tất cả các nhân viên trong đơn vị đó.
* Thêm nhân viên mới.
* Cập nhật thông tin cơ bản của nhân viên đang làm việc tại công ty.

1. **Phân tích yêu cầu**

**Vẽ biểu đồ mô hình hệ thống:**

**Biểu đồ các chức năng hệ thống:**

1. **Cấu trúc dữ liệu**

Gốc là một danh sách liên kết các công ty.

Mỗi công ty sẽ có tên công ty để phân biệt, và một danh sách các nhân viên, phân biệt bởi mã nhân viên (các nhân viên khác công ty cũng có mã số khác nhau).

**Nêu lí do vì sao**

Nhóm sử dụng cấu trúc dữ liệu trên nhằm quản lí nhân viên của một công ty (trong tập đoàn) dễ dàng hơn, rút ngắn được thời gian tìm kiếm khi biết trước tên công ty hoặc khi đọc dữ liệu của một công ty.

Tuy nhiên khi tìm kiếm nhân viên dựa theo thông tin của nhân viên đó thì không tránh khỏi việc duyệt toàn bộ.

**CHƯƠNG 3: CÀI ĐẶT KIỂM THỬ**

1. **Cài đặt**

**Mã nguồn các Cấu trúc dữ liệu (và giải thuật) mà nhóm đã sử dụng.**

Struct lưu trữ thông tin một nhân viên:

struct Employee {

char manv[10]; // ma nhan vien

char ho[30];

char ten[10];

char congty[30];

char chucvu[30];

Date sinhnhat;

char que[20]; // que quan

char diachi[50];

char email[50];

char sdt[50]; // so dien thoai

Date ngaybd; // ngay bat dau

vector<NgayLamViec> ngaylv; // ngay lam viec

};

Struct lưu trữ các nhân viên thuộc cùng một công ty:

struct Company {

char label[30]; // ten cong ty

vector<Employee\*> dsnv; // danh sach nhan vien

};

Các Struct phụ khác (Ngày, Giờ, Ngày làm việc):

struct Date {

int ngay;

int thang;

int nam;

};

struct Time {

int gio;

int phut;

};

struct NgayLamViec {

Date ngay;

Time gioden;

Time giove;

};

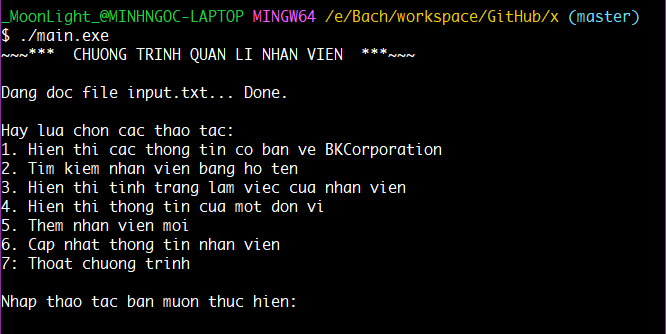
**Trình bày các kỹ thuật lập trình phòng thủ.**

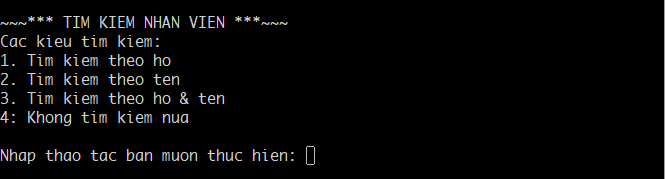
Nhóm đã sử dụng kĩ thuật **CHECK BAD INPUT** :

* Kiểm tra tên và mã nhân viên (bắt đầu bằng NVx <trong đó x là chữ số>) nhập vào hợp lệ hay chưa (isNameInvalid(input), isManvInvalid(input) ) ở tất cả các câu.
* Kiểm tra ngày tháng năm nhập vào đã hợp lí hay chưa ( isDayInvalid(input), isMonthInvalid(input), isYearInvalid(input) ) ở các câu 5 và 6.
* Kiểm tra xâu nhập có phải là xâu rỗng? (isInputEmpty(input)) ở tất cả các câu.
* Kiểm tra email đã hợp lệ chưa (IsEmailInvalid(input)) ở câu 6.
* Menu Options (choose\_feature(choice)) ở các chương trình chính và các câu 2 và 6.

1. **Kết quả**

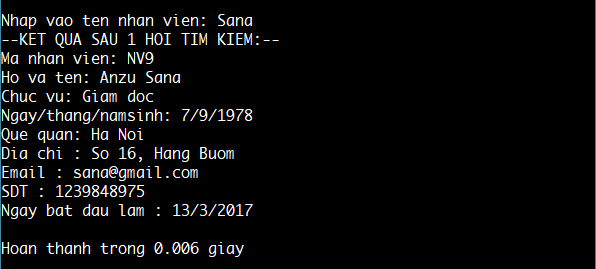
Screenshot một số chức năng

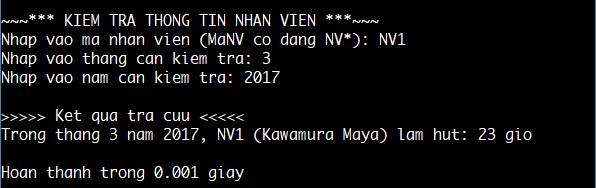
Chương trình chính:

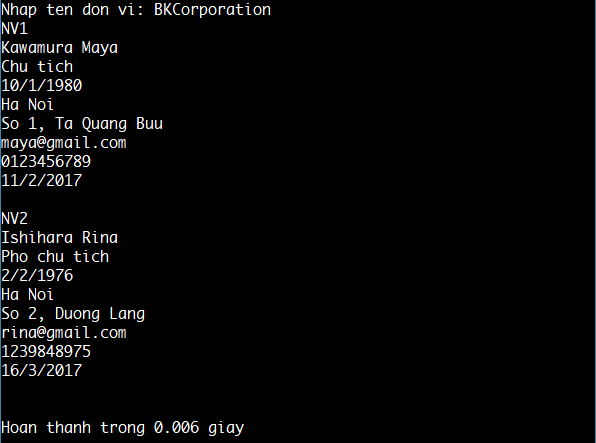
Câu 2:

1. **Kiểm thử**

Câu 1:

Câu 2

Câu 3

Câu 4

Chương trình đã vận hành tương đối ổn định, chính xác.

Tuy nhiên vẫn còn một số hạn chế:

* Phòng thủ còn yếu, sạn vẫn lọt vào chương trình.
* Giải thuật đơn giản, không hiệu quả khi đầu vào lớn.

Một số bug đã biết nhưng chưa xử lý được, nhóm sẽ rút kinh nghiệm vào các lần tới:

* Đầu vào quá lớn, chương trình không chạy được
* Chưa xử lí được dấu cách (space) trong input

**KẾT LUẬN**

1. **Phân công**

Phạm Sỹ Bằng:

* Câu 2
* Kiểm thử ứng dụng

Lê Sĩ Bích:

* Câu 1, câu 3
* Xây dựng cấu trúc dữ liệu
* Viết hàm đọc file, lưu vào cấu trúc dữ liệu

Trịnh Bá Huy:

* Câu 4
* Xây dựng Struct

Phạm Hùng Sơn:

* Câu 5
* Viết hàm in thông tin một nhân viên

Nguyễn Hữu Thắng:

* Câu 6
* Xây dựng hàm phòng thủ: check mail, check tên, check ngày, tháng, năm…
* Xây dựng hàm tìm nhân viên bằng mã số.

Viết báo cáo: cả nhóm