**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**---o0o---**



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**QUẢN LÍ NHÂN VIÊN**

**(DANH SÁCH LIÊN KẾT ĐƠN)**

**SVTH: DIỆP GIA HỮU 17110158**

**SVTH: NGUYỄN KHẮC HOÀNG PHI 17110207**

**GVPT: ThS. TRẦN CÔNG TÚ**

**Tp. Hồ Chí Minh, tháng 12 năm 2018**

MỤC LỤC

[DANH MỤC CÁC HÌNH 3](#_Toc532171219)

[DANH MỤC CÁC BẢNG 3](#_Toc532171220)

[LỜI CẢM ƠN 4](#_Toc532171221)

[I. MÔ TẢ PROJECT: 5](#_Toc532171222)

[1. Lý do xây dựng phần mềm: 5](#_Toc532171223)

[2. Mục tiêu của phần mềm: 6](#_Toc532171224)

[3. Phương pháp và ngôn ngữ: 6](#_Toc532171225)

[II. NỘI DUNG: 7](#_Toc532171226)

[1. Sơ lược về danh sách liên kết: 7](#_Toc532171227)

[2. Thiết kế giao diện: 7](#_Toc532171228)

[3. Thiết kế lớp: 9](#_Toc532171229)

[4. Ý nghĩa của từng phương thức: 10](#_Toc532171230)

[5. Mô tả phân công công việc: 21](#_Toc532171231)

[III.​ KẾT LUẬN: 22](#_Toc532171232)

[1. Đánh giá chung: 22](#_Toc532171233)

[2. Ưu điểm: 22](#_Toc532171234)

[3. Nhược điểm: 22](#_Toc532171235)

[4. Hướng phát triển: 22](#_Toc532171236)

[III. TÀI LIỆU KHAM KHẢO: 23](#_Toc532171237)

# DANH MỤC CÁC HÌNH

* Hình 2.1: Giao diện chính
* Hình 2.2: Giao diện thêm nhân viên
* Hình 2.3: Giao diện xóa nhân viên
* Hình 2.4: Giao diện thay đổi thông tin nhân viên

# DANH MỤC CÁC BẢNG

* Bảng 2.1: Thiết kế giao diện
* Bảng 2.2: Thiết kế các lớp
* Bảng 2.3: Ý nghĩa của từng phương thức
* Bảng 2.4: Phân công công việc

# LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Nay em xin được phép gửi lời cảm ơn chân thành này đến thầy Trần Công Tú, người đã trực tiếp hỗ trợ nhóm em trong suốt quá trình định hướng chọn đề tài, hướng dẫn, nhận xét và góp ý cũng như cung cấp tài liệu tham khảo. Nếu không có những lời hướng dẫn, những kinh nghiệm thực tiễn của thầy thì em nghĩ rằng bài thu hoạch này sẽ khó có thể hoàn thiện và hoàn thành đúng thời hạn được. Một lần nữa, em xin cảm ơn thầy.

Bài thu hoạch được thực hiện trong khoảng thời gian gần 8 tuần. Khoãng thời gian có hạn, cùng với kiến thức còn hạn chế và còn nhiều bỡ ngỡ khác do đó thiếu sót là điều không thể tránh khỏi nên em rất mong nhận được những ý kiến đóng góp quý báo của các quý Thầy Cô để kiến thức của em được hoàn thiện hơn sau này. Em xin chân thành cảm ơn.

|  |  |
| --- | --- |
|  |  |

# MÔ TẢ PROJECT:

## *Lý do xây dựng phần mềm:*

Thời đại 4.0 – Giai đoạn phát triển vượt bậc của công nghệ thông tin, trong giai đoạn này, hầu hết tất cả ngành nghề đều sử dụng sản phẩm từ lĩnh vực Công nghệ thông tin để việc sản xuất cũng như quản lí trở nên dễ dàng. Trong đó, không thể không nhắc đến viếc sử dụng công nghệ thông tin trong quản lí nhân sự. Ở giải đoạn trước, việc quản lí nhân sự của một cửa hàng nhỏ cũng đã hết sức phức tạp với những công cụ chưa toàn vẹn (đòi hỏi người dùng cần có kiến thức về tin học mới có thể sử dụng) nhưng khi áp dụng công nghệ thông tin vào việc này, người dùng không cần có kỹ năng tin học cao, mà chỉ cần nắm rõ cơ cấu tổ chức của cơ quan, tổ chức thì ban lãnh đạo sẽ dễ dàng quản lí nhân sự của cả một công ty lớn với sự giúp đỡ của phần mềm quản lí nhân sự (quản lí nhân viên).

**Ưu điểm của phần mềm quản lý nhân sự.**

* Giúp ban lãnh đạo có thể quản lý toàn bộ các thông tin và hồ sơ quan trọng về nhân sự theo cách vô cùng dễ dàng nhưng vẫn rất chuyên nghiệp
* Giúp doanh nghiệp tiết kiệm thời gian khi điều hành nhân sự, đưa ra các quyết định về nhân viên cũng đúng hơn.
* Dễ dàng trong việc lập kế hoạch và thời gian biểu công việc phù hợp cho nhân viên cũng như người quản lý.
* Giám sát quá trình quản lý hiệu quả nhất bên cạnh đó đồng thời cũng giúp giám sát về hiệu suất làm việc hàng ngày của mỗi một nhân viên.

Ngoài những ưu điểm thiết thực trên, thì Quản lí nhân viên (quản lí nhân sự) là một trong những đề tài có thể áp dụng tốt Danh sách liên kết – một trong những minh họa tốt về Danh sách liên kết trong môn học Cấu trúc dữ liệu và giải thuật.

## *Mục tiêu của phần mềm:*

Tạo ra được một phần mềm Quản lí nhân viên gần gũi, đơn giản giúp thực hiện các công việc quản lí như xuất ra danh sách nhân viên, tìm kiếm, thêm nhân viên, xóa nhân viên, thay đổi thông tin nhân viên, tính lương, chấm công, …

## *Giải thuật và ngôn ngữ:*

* Về giải thuật: Trong quá trình tham gia học tập môn Cấu trúc dữ liệu và giải thuật, nhóm đã nhận thấy Danh sách liên kết đơn hết sức phụ hợp cho việc thực hiện tạo danh sách quản lí nhân viên nói riêng và các công việc quản lí nói chung. Do đó, nhóm đã thực hiện viết phần mềm quản lí nhân viên ứng dụng từ danh sách liên kết đơn.

Cụ thể cách thức thực hiện như sau:

Bước 1: Cài đặt lớp nhân viên để dễ dàng xử lí thông tin nhân viên.

Bước 2: Cài đặt lớp Node với info kiểu NHANVIEN.

Bước 3: Cài đặt lớp LinkedList để quản lí thông tin của nhiều Node trong danh sách liên kết. Trong lớp này, cài đặt các phương thức thêm nút, xóa nút, đếm nút, hủy danh sách, sắp xếp,…

Bước 4: Liên kết các thao tác với giao diện qua việc tạo các sự kiện trên form.

* Về ngôn ngữ: Để dễ thao tác cũng như tạo giao diện cho phần mềm thì Winform C# là phương tiện thuận lợi cho sinh viên tụi em. Do đó nhóm đã tiến hành cài đặt phần mềm trên ngôn ngữ C#.

# NỘI DUNG:

## *Sơ lược về danh sách liên kết:*

Khái niệm danh sách liên kết: Danh sách liên kết (Linked list) là một kiểu danh sách được hiện thực bằng cách sử dụng kiểu dữ liệu con trỏ. Kiểu danh sách này cho phép thêm, hủy phần tử một cách linh động. (1)

Cấu tạo: Gồm nhiều nút liên kết với nhau qua việc lưu trữ ô nhớ lưu trữ. Mỗi nút có gồm info (để lưu giá trị) và next (lưu vị trí ô nhớ kế tiếp trong danh sách).

Với việc sử dụng danh sách liên kết, ta có thể khắc phục được lưu danh sách trên mảng đó là bộ nhớ linh động, được cấp phát khi cần và hủy bỏ khi thực hiện xong.

## *Thiết kế giao diện:*

Bảng 2.1: Thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Màn hình/Cửa sổ/Dialog** | **Giải thích ngắn gọn** | **Mục đích chính của màn hình** |
| 1 | Màn hình giao diện chính:    *Hình 2.1: Giao diện chính* | Để người dùng có thể dễ dàng tương tác cũng như thao tác trên phần mềm với giao diện đơn giản, dễ hiểu. | Hiển thị ra giao diện chính của phần mềm. |
| 2 | Màn hình thêm nhân viên :    *Hình 2.2: Giao diện thêm nhân viên* | Lấy thông tin của một nhân viên mới để truyển về để vào danh liên kết chứa. | Hiển thị giao diện thêm nhân viên vào danh sách liên kết. |
| 3 | Màn hình xóa nhân viên :    *Hình 2.3: Giao diện xóa nhân viên* | Tìm thông tin của một nhân viên cần xóa bằng MSNVđể xóa trong danh sách liên kết. | Hiển thị giao diện xóa nhân viên trong danh sách liên kết. |
| 4 | Màn hình thay đổi thông tin nhân viên :    *Hình 2.4: Giao diện thay đổi thông tin nhân viên* | Tìm thông tin của một nhân viên cần thay đổi bằng MSNV và thay đổi thông tin của nhân viên bằng cách điền vào giao diện. | Hiển thị giao diện thay đổi thông tin nhân viên trong danh sách liên kết. |

## *Thiết kế lớp:*

*Bảng 2.2: Thiết kế các lớp*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên lớp** | **Tên các SV phụ trách viết** | **Mục đích chính của lớp trong chương trình** |
| 1 | fNhanVien | Diệp Gia Hữu | Lớp hiển thị giao diện chính của chương trình. |
| 2 | fThem | Diệp Gia Hữu | Lớp hiển thị giao thêm nhân viên vào danh sách liên kết. |
| 3 | fXoa | Nguyễn Khắc Hoàng Phi | Lớp hiển thị giao diện xóa nhân viên khỏi danh sách liên kết. |
| 4 | fSua | Nguyễn Khắc Hoàng Phi | Lớp hiển thị giao diện thay đổi thông tin nhân viên trong danh sách liên kết. |
| 5 | NHANVIEN | Nguyễn Khắc Hoàng Phi | Lớp tổ chức thông tin của một nhân viên. |
| 6 | Node | Diệp Gia Hữu | Lớp tổ chữ dữ liệu của một nút trong danh sách liên kết. |
| 7 | LinkedList | Diệp Gia Hữu  Nguyễn Khắc Hoàng Phi | Lớp tổ chức cấu trúc dữ liệu danh sách liên kết đơn. |

## *Ý nghĩa của từng phương thức:*

Bảng 2.3: Ý nghĩa của từng phương thức

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên và nội dung phương thức,**  **mục đích** | **Giải thích phương thức** | **Tên file, số thứ tự dòng chứa khai báo** |
| 1 | Tên: Node  Input: Dữ liệu một nhân viên, ví trí của nút kế tiếp.  Output: không có.  Mục đích: Tạo nút mới truyền giá trị trực tiếp.  Code:  public Node(NHANVIEN x, Node next)  {  this.info = x;  this.Next = next;  } | Tạo giá trị và liên kết nút đó với một nút khác. | LinkedList.cs  (21) |
| 2 | Tên: CreateNode  Input: Thông tin của một nhân viên  Output: không có  Mục đích: Khai báo nút mới với next bằng null.  Code:  public Node CreateNode(NHANVIEN x)  {  Node p = new Node();  p.info = new NHANVIEN();  p.info.MSNV = x.MSNV;  p.info.HoTenDem = x.HoTenDem;  p.info.Ten = x.Ten;  p.info.CMND = x.CMND;  p.info.NgaySinh = x.NgaySinh;  p.info.CongViec = x.CongViec;  p.info.Luong = x.Luong;  p.Next = null;  if (p.info == null)  {  return null;  }  return p;  } | Tạo một nút mới có giá trị kiểu nhân viên, nếu giá trị bằng null thì trả về null. | LinkedList.cs  (26) |
| 3 | Tên: Nhap  Input: Thông tin của nhân viên  Output: không có  Mục đích: Nhập giá trị cho một đối tượng nhân viên.  Code:  public void Nhap(string msnv, string hotendem, string ten, string cmnd, string ngaysinh, string congviec, string luong)  {  this.MSNV = msnv;  this.HoTenDem = hotendem;  this.Ten = ten;  this.CMND = cmnd;  this.NgaySinh = ngaysinh;  this.CongViec = congviec;  this.Luong = luong;  } | Nhập thông tin của một nhân viên. | LinkedList.cs  (59) |
| 4 | Tên: LinkedList  Input: không có  Output: không có  Mục đích: Khởi tạo một danh sách liên kết rỗng.  Code:  public LinkedList()  {  Head = Tail = null;  } | Cho nút đầu bằng nút cuối và có giá trị bằng null. | LinkedList.cs  (77) |
| 5 | Tên: IsEmptyList  Input: không có  Output: không có  Mục đích: Kiểm tra danh sách rỗng.  Code:  public int IsEmptyList()  {  if (this.Head == null)  return 1;  return 0;  } | Kiểm tra danh sách có rỗng hay không qua việc so sánh nút đầu. | LinkedList.cs  (82) |
| 6 | Tên: addTail  Input: Biến kiểu nhân viên chứa thông tin nhân viên.  Output: không có  Mục đích: Thêm vào cuối danh sách.  Code:  public void addTail(NHANVIEN x)  {  Node p = new Node(x, null);  if (this.Head == null)  this.Head = this.Tail = p;  else  {  this.Tail.Next = p;  this.Tail = p;  }  } | Khai báo một nút mới có giá trị kiểu nhân viên và Next là null. Kiểm tra nếu danh sách rỗng thì gán nút đầu và nút cuối của danh sách liên kết bằng nút vừa tạo. Nếu sai thì gán Next của nút cuối bằng vị trí lưu trữ nút vừa khai báo và giá trị nút cuối bằng giá trị nút khai báo. | LinkedList.cs  (89) |
| 7 | Tên: addHead  Input: Biến kiểu nhân viên chứa thông tin nhân viên  Output: không có  Mục đích: Thêm vào đầu danh sách.  Code:  public void addHead(NHANVIEN x)  {  Node p = new Node(x, Head);  Head = p;  if (Tail == null)  Tail = Head;  } | Khai báo nút mới có giá trị bằng thông tin nhân viên nhập vào và Next của nút này bằng vị trí lưu trữ của nút đầu trong danh sách liên kết. Sau đó gán nút đầu bằng nút vừa tạo. Kiểm tra nếu nút cuối bằng null thì gán nút cuối bằng nút đầu. | LinkedList.cs  (101) |
| 8 | Tên: SearchNode  Input: Danh sách cần tìm bên trong, dữ liệu cần tìm (kiểu string)  Output: không có  Mục đích: Tìm kiếm thông tin trong danh sách liên kết.  Code:  public void SearchNode(LinkedList myList, string x)  {  Node p = myList.Head;  while (p!=null)  {  NHANVIEN temp = new NHANVIEN();  if ((x == p.info.MSNV)||  (x == p.info.HoTenDem) ||  (x == p.info.Ten) ||  (x == p.info.CMND) ||  (x == p.info.NgaySinh) ||  (x == p.info.CongViec) ||  (x == p.info.Luong))  {  temp.Nhap(p.info.MSNV, p.info.HoTenDem, p.info.Ten, p.info.CMND, p.info.NgaySinh, p.info.CongViec, p.info.Luong);  found.Add(temp);  }  p = p.Next;  }  } | Khai báo nút mới bằng giá trị nút đầu trong danh sách liên kết để làm biến chạy. Gọi vòng lặp while với điều kiện biến chạy khác null. Lần lượt so sánh chuỗi kí tự cần tìm với các thuộc tính của nhân viên được lưu trong danh sách liên kết, đẩy next của biến chạy sáng ô nhớ kế tiếp. Nếu tìm được sẽ thêm nhân viên tìm được vào một danh sách mới. | LinkedList.cs  (109) |
| 9 | Tên: removeHead  Input: Danh sách liên kết thao tác  Output: không có  Mục đích: Xóa phần tử đầu trong danh dách.  Code:  public void removeHead(LinkedList myList)  {  if (myList.Head == null)  {  }  else if (myList.Head == myList.Tail)  {  myList.Head = null;  myList.Head = myList.Tail = null;  }  else  {  Node temp = new Node();  temp = myList.Head;  myList.Head = myList.Head.Next;  temp = null;  }  } | Kiểm tra xem danh sách truyền vào có rỗng hay không, nếu danh sách không rỗng thì kiểm tra xem danh sách có một phần tử duy nhất không.  Nếu danh sách chỉ có một phần tử, thì gán nút đầu bằng null, nút cuối bằng null. Nếu hai trường hợp trên đều sai, thì khai báo một nút tạm và gán nó bằng nút đầu danh sách, gán nút đầu danh sách bằng nút ở ô kế, sau cùng là hủy nút tạm. | LinkedList.cs  (124) |
| 10 | Tên: removeAfter  Input: danh sách thao tác, nút cần xóa nút phía sau nó.  Output: Không có.  Múc đích: Xóa một phần tử sau node chỉ định.  Code:  public void removeAfter(LinkedList myList, Node node)  {  if (myList.Head == null)  { }  else  {  Node temp = new Node();  temp = node.Next;  if (temp != null)  {  node.Next = temp.Next;  temp = null;  }  }  } | Nếu danh sách không rỗng, khai báo nút tạm và gán nó bằng ô kế của nút mà muốn xóa nút phía sau nó.  Nếu biến tạm vừa gán khác null, ô nhớ của nút truyền vào gán bằng nút có kế của biến tạm, hủy biến tạm. Tức là ngắt ra và bỏ một nút sau nút truyền vào. | LinkedList.cs  (143) |
| 11 | Tên: removeKey  Input: danh sách thao tác, mã số nhân viên muốn xóa.  Output: không có.  Mục đích: Xóa nút có khóa nhập vào, cụ thể là Mã số nhân viên.  Code:  public void removeKey(LinkedList myList, string x)  {  if (myList.Head == null)  { }  else  {  Node result = new Node();  result = myList.Head;  Node before = new Node();  before = null;  while (result.info.MSNV != x)  {  before = result;  result = result.Next;  }  if (result != null)  {  if (result == myList.Head)  removeHead(myList);  else  removeAfter(myList, before);  }  }  } | Nếu danh sách không rỗng, khai báo nút result và gán nó bằng nút đầu danh sách. Khai báo nút before bằng null.  Duyệt qua danh sách liên kết với vòng lặp While điều kiện nút giá trị mã số nhân viên nút result khác giá trị cần tìm, trong vòng lặp liên tục gán nút before bằng result để lưu lại vị trí nút cần xóa (nếu thỏa điều kiện ngưng, nút cần xóa là nút kế tiếp tiếp nút before). Nếu result ở đầu danh sách thì gọi phương thức removeHead, ngược lại gọi phương thức removeAfter truyền vào nút before. | LinkedList.cs  (159) |
| 12 | Tên: CountNode  Input: danh sách thao tác  Output: Không có.  Mục đích: đếm số nút có trong danh sách liên kết.  Code:  public int CountNode(LinkedList myList)  {  int dem = 0;  Node p = new Node();  p = myList.Head;  while (p!=null)  {  dem++;  p = p.Next;  }  return dem;  } | Khai báo biến đếm bằng 0, khởi tạo nút và gán bằng nút đầu danh sách để làm biến chạy. Duyệt qua danh sách với biến chạy vừa khai báo bằng vòng lặp While điều kiện biến chạy khác null, cứ một lần tăng giá trị biến đếm lên 1 và đầy qua ô kế. Phương thức sẽ trả về số nút có trong danh sách liên kết. | LinkedList.cs  (184) |
| 13 | Tên: clear  Input: danh sách liên kết thao tác.  Output: Không có  Mục đích: Hủy cả danh sách liên kết.  Code:  public void clear(LinkedList myList)  {  Node temp = new Node();  temp = myList.Head;  while (temp != null)  {  myList.removeHead(myList);  temp = temp.Next;  }  } | Hủy bằng các duyệt qua danh sách và liên tục gọi phương thức removeHead để xóa phần tử đầu cho đến khi không còn phần tử nào. | LinkedList.cs  (197) |
| 14 | Tên: Sort  Input: Không có.  Output: không có.  Mục đích: Sắp xếp danh sách liên kết theo Mã số nhân viên.  Mã giả:  public void Sort()  {  LinkedList lnv1 = new LinkedList();  LinkedList lnv2 = new LinkedList();  Node pivot, p;  if (Head == Tail)  return;  pivot = Head;  p = Head.Next;  string[] msnvPivot = pivot.info.MSNV.Split('V');  while (p != null)  {  Node q = p;  p = p.Next;  q.Next = null;  string[] msnvQ = q.info.MSNV.Split('V');  if (Convert.ToInt32(msnvQ[1]) < Convert.ToInt32(msnvPivot[1]))  lnv1.addTail(q.info);  else  lnv2.addTail(q.info);  };  lnv1.Sort();  lnv2.Sort();  if (lnv1.IsEmptyList() == 0)  {  this.Head = lnv1.Head;  lnv1.Tail.Next = pivot;  }  else  this.Head = pivot;  pivot.Next = lnv2.Head;  if (lnv2.IsEmptyList() == 0)  this.Tail = lnv2.Tail;  else  this.Tail = pivot;  }  } | Sử dụng phương pháp Quick Sort. | LinkedList.cs  (208) |
| 15 | Tên: DanhSach\_Click  Input: Không có  Output: Xuất ra danh sách nhân viên trên dataGridView.  Mục đích: Tạo sự kiện cho button XUẤT DANH SÁCH trên form fNhanVien. | Tạo datatable để chứa dữ liệu được truyền vào từ danh sách liên kết. Chuyển nó lên dataGridView để hiển thị. | fNhanVien.cs  (29) |
| 16 | Tên: bThem\_Click  Input: Thông tin nhân viên cần thêm.  Output: Gọi form fThem để điền thông tin.  Mục đích: Tạo sự kiện cho button bThem trên form fNhanVien để thêm nhân viên vào danh sách liên kết. | Gọi form fThem để lấy dữ liệu, khai báo một nhân viên mới. Kiểm tra điểu kiện, nếu thõa gọi phương thức addHead để thêm vào danh sách liên kết. | fNhanVien.cs  (71) |
| 17 | Tên: bXoa\_Click  Input: Mã số nhân viên cần xóa.  Output: Gọi form fXoa để đền MSNV cần xóa.  Mục đích: Tạo sự kiện cho button bXoa trên form fNhanVien để xóa nhân viên khỏi danh sách liên kết. | Gọi form fXoa để lấy dữ liệu. Duyệt qua danh sách nếu thõa điều kiện gọi phương thức removeKey để xóa. | fNhanVien.cs  (101) |
| 18 | Tên: bTimKiem\_Click  Input: Thông tin cần tìm.  Output: Nhân viên có thông tin thõa.  Mục đích: Tạo sự kiện cho button bTimKiem trên form fNhanVien để xuất ra dánh sách cần tìm. | Lấy dữ liệu cần tim ở ô textbox tTimKiem sau đó gọi phương thức SearchNode. Tạo datatable chứa danh sách tìm được. Xuất dữ liệu từ datatable sang dataGridView. | fNhanVien.cs  (124) |
| 19 | Tên: bSua\_Click  Input: Thông tin nhân viên.  Output: form sửa thông tin nhân viên.  Mục đích: Tạo sự kiện cho button SỬA, để thay đổi thông tin nhân viên. | Gọi form fSua để lấy thông tin. Duyệt qua danh sách liên kết để so sánh MSNV, nếu thõa gán lại giá trị mới. | fNhanVien.cs  (172) |
| 20 | Tên: bLuu\_Click  Input: Không có.  Output: Thông báo.  Mục đích: Tạo sự kiện cho button nhằm lưu thông tin từ danh sách liên kết vào file “DanhSachNhanVien.txt”. | Gọi hàm CountNode để lấy số nút trong danh sách, ghi số nút ra file. Duyệt qua dánh sách, lấy dữ liệu từng nút, thực hiện ghép chuỗi kí tự rồi ghi ra file. | fNhanVien.cs  (192) |
| 21 | Tên: bDoc\_Click  Input: file DanhSachNhanVien.txt  Output: Thông báo.  Mục đích: Tạo sự kiện cho button nhằm lấy dữ liệu từ file txt. | Gọi hàm clear để đảm bảo danh sách đg rỗng. Đọc từng dòng trong file và thêm vào List. Lấy từng phần tử trong list để cắt chuỗi sau đó đưa vào danh sách liên kết. | fNhanVien.cs  (216) |

## *Mô tả phân công công việc:*

Bảng 2.4: Phân công công việc

|  |  |  |
| --- | --- | --- |
| **Phương thức cài đặt** | **Diệp Gia Hữu** | **Nguyễn Khắc Hoàng Phi** |
| Tìm hiểu về đề tài | X | X |
| Tìm hiểu cách làm giao diện | X | X |
| Tổ chức dữ liệu lớp Node | X |  |
| Tổ chức dữ liệu lớp NHANVIEN |  | X |
| Thiết kế giao diện chính fNhanVien | X |  |
| Thiết kế giao diện thêm fThem | X |  |
| Thiết kế giao diện xóa fXoa |  | X |
| Thiết kế giao diện sửa fSua |  | X |
| Tạo chức năng lấy dữ liệu từ file txt | X |  |
| Tạo chức năng ghi dữ liệu ra file txt |  | X |
| Tạo chức năng xuất danh sách | X |  |
| Tạo chức năng thêm nhân viên  cùng các phương thức liên quan |  | X |
| Tạo chức năng xóa nhân viên  cùng các phương thức liên quan | X |  |
| Tạo chức năng sửa thông tin |  | X |
| Tạo chức năng tìm kiếm cùng các  phương thức liên quan | X |  |
| Phương thức Sort |  | X |
| Viết báo cáo | X | X |
| Làm Power Point | X |  |
| Thuyết trình | X | X |

# III.​ KẾT LUẬN:

## *Đánh giá chung:*

* Đồ án đã được hoàn thành được hơn 80% mục tiêu đề ra.

## *Ưu điểm:*

* Giao diện dễ hiểu.
* Sử dụng hợp lí các phương thức trên danh sách liên kết để quản lí nhân viên.

## *Nhược điểm:*

* Chưa tối ưu hóa phần mềm.
* Phần giao diện thiết kế còn đơn giản có phần chưa hợp lí vì đây là lần đầu tiên nhóm sử dụng winform và chưa từng được học trước đó.
* Khả năng làm việc nhóm còn kém, chưa linh hoạt dẫn đên thời gian hoàn thành công việc chậm.
* Vì sử dụng file txt để lưu dữ liệu nên không thể quản lí nhiều thuộc tính (việc cắt chuỗi, xử lí dữ liệu trên txt khá rườm rà). Do đó, nhóm chỉ làm vài thuộc tính cơ bản để tượng trưng. Dẫn đến, phần mềm chưa đủ điều kiện để sử dụng thực tế.

## *Hướng phát triển:*

* Chuyển qua Cơ sở dữ liệu bằng việc liên kết SQL server khi đó có thể thêm nhiều chức năng mới như tính lương, chấm ca, …
* Thiết kế lại giao diện một cách hớp lí nhất có thể, bắt mắt người dùng.

1. **TÀI LIỆU KHAM KHẢO:**

Đồ án của nhóm có tham khảo một số tài liệu tại các Website sau:

* (1) Giáo trình cấu trúc dữ liệu và giải thuật.
* https://anonyviet.com/tong-hop-ma-nguon-c-sharp/