**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 1**

**QUẢN LÍ CÁC BÀI BÁO KHOA HỌC**

**SVTH: DIỆP GIA HỮU 17110158**

**SVTH: TRƯƠNG MINH KHOA 17110163**

**SVTH: HỒ SĨ TUẤN 17119111**

**GVPT: TỪ TUYẾT HỒNG**

**Tp. Hồ Chí Minh, tháng 11 năm 2019**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Giáo viên hướng dẫn

(ký và ghi họ tên)

…………………….....

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 cô Từ Tuyết Hồng, 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 cô 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 cô.

Em cũng xin chân thành cảm ơn các quý thầy cô trong khoa Công Nghệ Thông Tin đã giúp đỡ hỗ trợ kiến thức cũng như giải đáp thắc mắc của em. Cùng với đó, em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp cho em hoàn thiện đề tài hơn.

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.

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

# ĐẶC TẢ:

## *Lý do xây dựng phần mềm:*

Đọc báo – một nhu cầu thiết yếu trong cuộc sống hằng ngày. Trong thời đại hiện tại, loại báo phổ biến sẽ là báo điện tử, một chủ đề hết sức nổi bật trong giới báo chí là Khoa học. Cùng với sự phát triển của báo điện tử, không thể thiếu việc kiểm soát, quản lí các bài báo. Để tạo sự thuận tiện cho việc đó, sự cần thiết của phần mềm quản lí các bài báo nói chung và các bài báo khoa học nói riêng là không thế phủ nhận. Do đó nhóm chúng em quyết định chọn đề tài này.

**Ưu điểm của phần mềm quản lý các bài báo Khoa học:**

* Giúp ban quản lý kiểm soát toàn bộ các thông tin về các bài báo, tác giả, nhà xuất bản… một cách dễ dàng hơn.
* Giúp ban quản lí báo tiết kiệm được nhiều thời gian.
* Dễ dàng thêm, xóa, chỉnh sửa,...
* Quản lí một cách hiệu quả hơn các bài báo.

Ngoài những ưu điểm thiết thực trên, thì Quản lí các bài báo khoa học là một trong những đề tài có thể áp dụng lại những nội dung đã học ở những kì trước.

## *Mục tiêu của phần mềm:*

Tạo ra được một phần mềm Quản lí các bài báo Khoa học gần gũi, đơn giản giúp thực hiện các công việc quản lí như xuất ra danh sách dạng Excel, tìm kiếm, thêm, xóa, sửa các đối tượng bài báo, tác giả, nhà xuất bảng, phân quyền riêng biệt cho từng tài khoản của quản trị viên,…

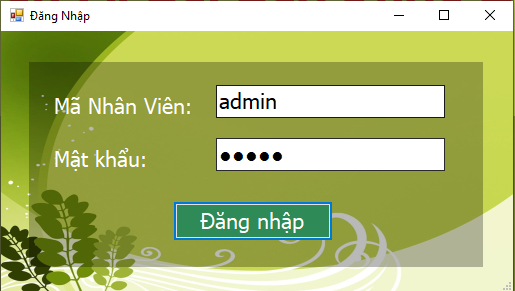
## *Dữ liệu, thông tin đầu vào:*

* Danh sách bài báo Khoa học với trên 400 bài.
* Dang sách Tác giả.
* Danh sách Nhà xuất bản.
* Dang sách người dùng.

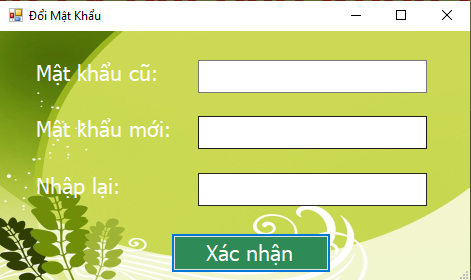
## *Giao diện dự kiến:*

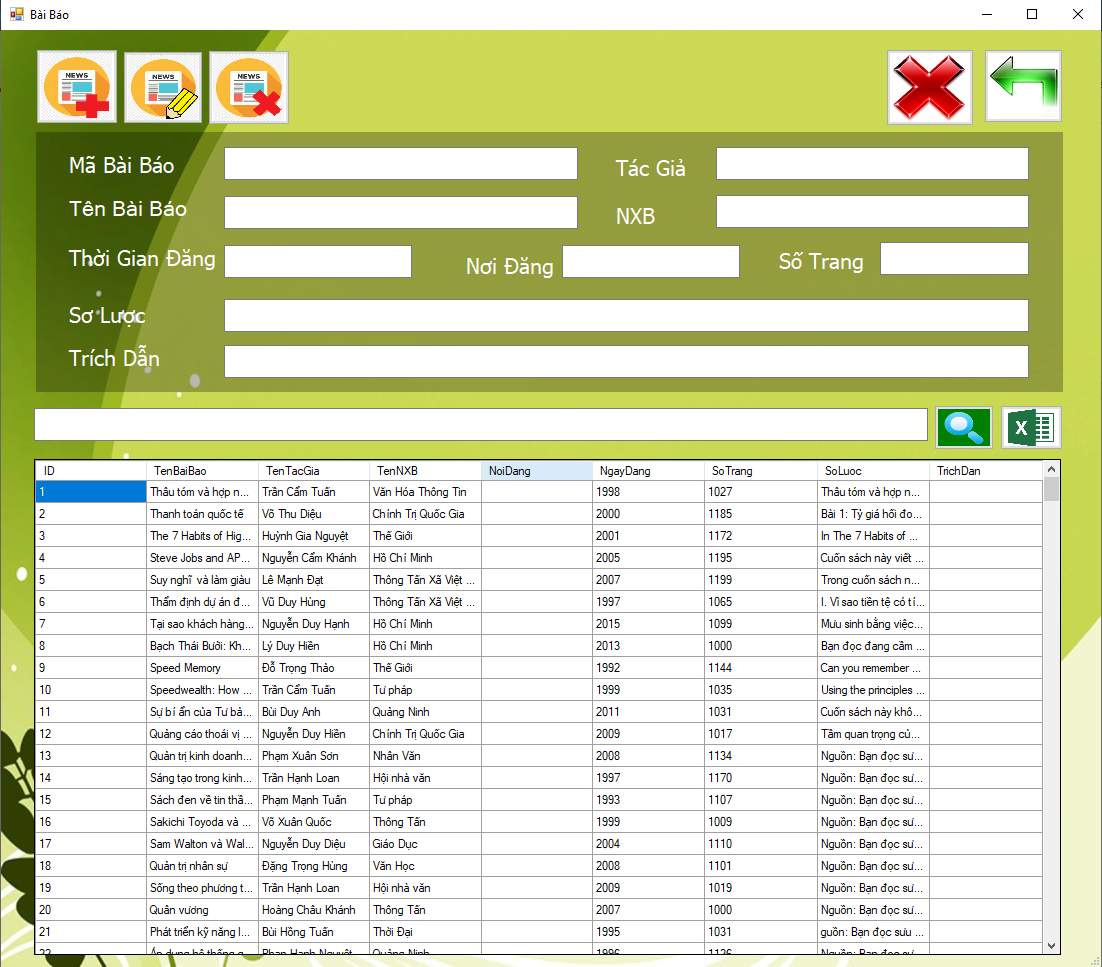
Giao diện chính:

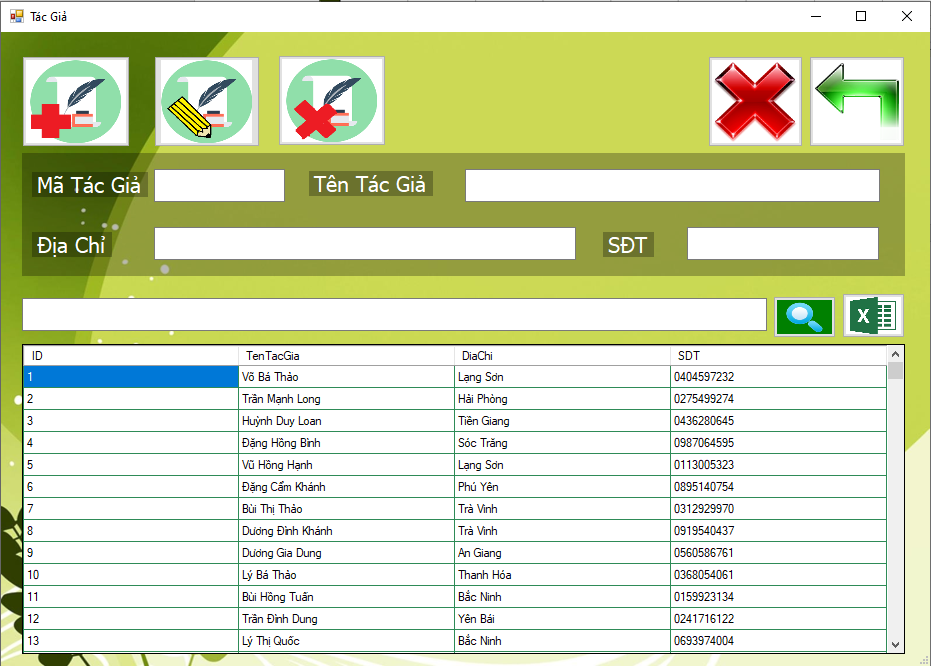
Giao diện đăng nhập:



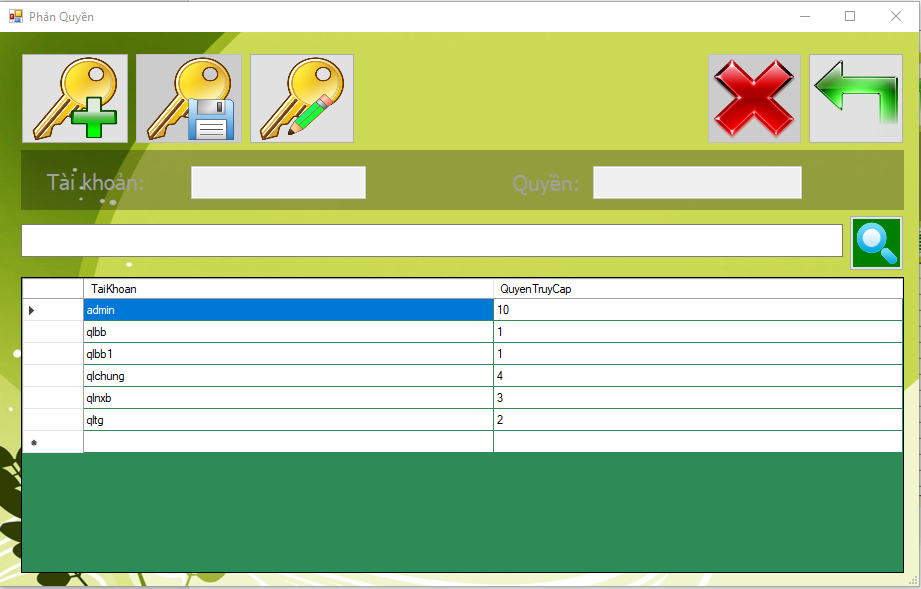
Giao diện đổi mật khẩu:



Giao diện quản lí Bài báo:

Giao diện quản lí Tác giả:

Giao diện quản lí Nhà xuất bản:

Giao diện quản lí Tài khoản truy cập:

# PHÂN CÔNG CÔNG VIỆC:

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên Sinh viên | Mô tả khải quát mảng công việc SV thực hiện  trong đồ án | Ước tính phần trăm đóng góp |
| 1 | Diệp Gia Hữu |  |  |
| 2 | Trương Minh Khoa |  |  |
| 3 | Hồ Sĩ Tuấn |  |  |

# THIẾT KẾ:

## *Thiết kế lớp:*

*Bảng 2.1: Thiết kế các lớp*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên lớp** | **Tên các SV phụ trách viết** | **Mục đích chính của lớp trong chương trình** |
| 1 | frmMain |  | Lớp hiển thị giao diện chính của chương trình. |
| 2 | frmDangNhap |  | Lớp hiển thị giao diện đăng nhập. |
| 3 | frmDoiMk |  | Lớp hiển thị giao diện đổi mật khẩu. |
| 4 | frmBaiBao |  | Lớp hiển thị giao diện quản lí bài báo khoa học. |
| 5 | frmTacGia |  | Lớp hiển thị giao diện quản lí tác giả. |
| 6 | frmNXB |  | Lớp hiển thị giao diện quản lí Nhà xuất bản. |
| 7 | frmPhanQuyen |  | Lớp hiển thị giao diện quản lí Tài khoản truy cập. |
| 8 | table\_BaiBao |  | Lớp chứa các câu lệnh để truy vấn SQL Server phục vụ cho các phương thức của frmBaiBao. |
| 9 | table\_TacGia |  | Lớp chứa các câu lệnh để truy vấn SQL Server phục vụ cho các phương thức của frmTacGia. |
| 10 | table\_NXB |  | Lớp chứa các câu lệnh để truy vấn SQL Server phục vụ cho các phương thức của frmNXB. |
| 11 | table\_user1 |  | Lớp chứa các câu lệnh để truy vấn SQL Server phục vụ cho các phương thức của frmPhanQuyen, frmDangNhap, frmDoiMK. |

## *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** | **Tên SV phụ trách viết** |
| 1 | Tên: public frmMain()  Input: Không có.  Output: không có.  Mục đích: Load form.  Code:  public frmMain()  {  InitializeComponent();  } | Gọi hàm load form. | frmMain.cs  (20) |  |
| 2 | Tên: private void lDN\_Click(object sender, EventArgs e)  Input: Event click chuột vào button  Output: không có  Mục đích: Cùng một label có thể đăng nhập lẫn đăng xuất.  Mã giả:  Nếu testlogin (Biến kiểm tra đăng nhập) là đúng:  Hiện thông báo, đổi giá trị của testlogin về false, đổi các thuộc tính của label lDN.  Ngược lại: Hiện form frmDangNhap. | Biến testlogin được lấy giá trị từ frmDangNhap nhằm xét tình trạng đăng nhập để thực hiện một số chức năng. | frmMain.cs  (25) |  |
| 3 | Tên: private void btnBaiBao\_Click(object sender, EventArgs e)  Input: Event click vào button  Output: không có  Mục đích: Kiểm tra quyền truy cập và hiện frmBaiBao.  Mã giả:  Nếu testlogin đúng, quyền truy cập đúng: Hiện form frmBaiBao.  Ngược lại: Hiện thông báo. | Bằng việc kiểm tra các biến thể hiện các quyền truy cập để hiện form Quản lí bài báo. | frmMain.cs  (44) |  |
| 4 | Tên: private void btnTacGia\_Click(object sender, EventArgs e)  Input: Event click vào button  Output: không có  Mục đích: Kiểm tra quyền truy cập và hiện frmTacGia.  Mã giả: Tương tự phương thức thứ 3 | Bằng việc kiểm tra các biến thể hiện các quyền truy cập để hiện form Quản lí Tác giả. | frmMain.cs  (54) |  |
| 5 | Tên: private void btnNXB\_Click(object sender, EventArgs e)  Input: Event click vào button  Output: không có  Mục đích: Kiểm tra quyền truy cập và hiện frmNXB  Mã giả: Tương tự phương thức thứ 3 | Bằng việc kiểm tra các biến thể hiện các quyền truy cập để hiện form Quản lí Nhà xuất bản. | frmMain.cs  (64) |  |
| 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) |  |

## *Thiết kế giao diện:*

Bảng 2.2: 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. |

# 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 xử 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:

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