‘’p

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TPHCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

🙢 🕮 🙠



**BÁO CÁO ĐỒ ÁN WINDOWS**

**CHƯƠNG TRÌNH TẠO, LƯU GHI CHÚ**

**Nhóm sinh viên thực hiện:**

**Nguyễn Hữu Tường 18110235**

**Đặng Nguyễn Hoài Thư 18110209**

**Nguyễn Việt Long 17110177**

**GIẢNG VIÊN:** TS. HUỲNH XUÂN PHỤNG

**Tp. Hồ Chí Minh, tháng 7 năm 2020**

**TPHCM, tháng 12 năm 2019.**

**ĐIỂM SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

…………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Giáo viên hướng dẫn

(***Ký và ghi họ tên***)

**Huỳnh Xuân Phụng**

**LỜI CẢM ƠN**

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, tiến sĩ Huỳnh Xuân Phụng, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn khoa đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa đào tạo chất lượng cao nói chung và ngành công nghệ thông tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báu của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

**TP.HCM, ngày 10 tháng 11 năm 2019**

Nhóm sinh viên thực hiện

MỤC LỤC

[Danh mục các hình 1](#_Toc44227539)

[Danh mục các bảng 2](#_Toc44227540)

[Chương 1: Tổng quan chương trình 3](#_Toc44227541)

[1. Giới thiệu chung 3](#_Toc44227542)

[**1.1. Về đồ án chương tình tạo, lưu ghi chú 3**](#_Toc44227543)

[**1.2. SQL, C#, Windows Form 3**](#_Toc44227544)

[2. Đặc tả phần mềm 6](#_Toc44227545)

[**2.1. Phần mềm tạo, lưu ghi chú 6**](#_Toc44227546)

[**2.2. Yêu cầu kỹ thuật 9**](#_Toc44227547)

[**2.3. Công cụ và công nghệ sử dụng 9**](#_Toc44227548)

[Chương 2: Kế hoạch thực hiện 10](#_Toc44227549)

[1. Kế hoạch 10](#_Toc44227550)

[2. Phân công công việc 11](#_Toc44227551)

[Chương 3: Thiết kế phần mềm 12](#_Toc44227552)

[1. Thiết kế giao diện 12](#_Toc44227553)

[2. Thiết kế lớp 13](#_Toc44227554)

[**2.1. Thiết kế lớp cho Layer 13**](#_Toc44227555)

[**2.2. Thiết kế cơ sở dữ liệu 18**](#_Toc44227556)

[**2.3. Đặc tả nút chức năng 21**](#_Toc44227557)

[Chương 4: Kiểm thử 22](#_Toc44227558)

[Chương 5: Kết luận 26](#_Toc44227559)

[1. Kết luận: 26](#_Toc44227560)

[2. Hướng phát triển: 26](#_Toc44227561)

[Tài liệu tham khảo 27](#_Toc44227562)

# 

# Danh mục các hình

[Hình 1: .NET Framework 5](#_Toc43901553)

[Hình 2: Cơ chế hoạt động của windows form 6](#_Toc43901554)

[Hình 3: Màn hình đăng nhập 7](#_Toc43901555)

[Hình 4: Giao diện chính 8](#_Toc43901556)

[Hình 5: List Tag 8](#_Toc43901557)

[Hình 6: Giao diện phần mềm 12](#_Toc43901558)

[Hình 7: UML Diagram biểu diễn các lớp trong phần mềm 13](#_Toc43901559)

[Hình 8: Kiểm thử 1 22](#_Toc43901560)

[Hình 9: Kiểm thử 2 22](#_Toc43901561)

[Hình 10: Kiểm thử 3 23](#_Toc43901562)

[Hình 11: Kiểm thử 4 23](#_Toc43901563)

[Hình 12: Kiểm thử 5 23](#_Toc43901564)

[Hình 13: Kiểm thử 6 24](#_Toc43901565)

[Hình 14: Kiểm thử 7 24](#_Toc43901566)

[Hình 15: Kiểm thử 8 24](#_Toc43901567)

[Hình 16: Kiểm thử 9 25](#_Toc43901568)

[Hình 17: Kiểm thử 10 25](#_Toc43901569)

# Danh mục các bảng

[Bảng 1: Kế hoạch theo tuần 10](#_Toc43901593)

[Bảng 2: Phân công công việc 11](#_Toc43901594)

[Bảng 3: Danh mục các lớp cho chương trình 13](#_Toc43901595)

[Bảng 4: Đặc tả các phương thức trong lớp Note\_backupcontroller 14](#_Toc43901596)

[Bảng 5: Đặc tả các phương thức trong lớp Notecontrollers 14](#_Toc43901597)

[Bảng 6: Đặc tả các phương thức trong lớp Tag\_Notecontroller 15](#_Toc43901598)

[Bảng 7: Đặc tả các phương thức trong lớp Tagcontroller 15](#_Toc43901599)

[Bảng 8: Đặc tả các phương thức trong lớp User\_Note\_backupcontroller 16](#_Toc43901600)

[Bảng 9: Đặc tả các phương thức trong lớp User\_Notecontrollers 16](#_Toc43901601)

[Bảng 10: Đặc tả các phương thức trong lớp Usercontrollers 17](#_Toc43901602)

[Bảng 11: Các bảng trong cơ sở dữ liệu 18](#_Toc43901603)

[Bảng 12: Các trường trong bảng Note\_Backup 18](#_Toc43901604)

[Bảng 13: Các trường trong bảng Note 19](#_Toc43901605)

[Bảng 14: Các trường trong bảng Note\_Tag 19](#_Toc43901606)

[Bảng 15: Các trường trong bảng Tag 19](#_Toc43901607)

[Bảng 16: Các trường trong bảng User\_Note\_Backup 20](#_Toc43901608)

[Bảng 17: Các trường trong bảng User\_Note 20](#_Toc43901609)

[Bảng 18: Các trường trong bảng User 20](#_Toc43901610)

[Bảng 19: Các nút chức năng 21](#_Toc43901611)

# Chương 1: Tổng quan chương trình

## Giới thiệu chung

### Về đồ án chương tình tạo, lưu ghi chú

#### Yêu cầu đồ án

Thiết kế và xây dựng phần mềm hướng đối tượng giải quyết yêu cầu tạo, lưu ghi chú sử dụng SQL bằng Windowns Form.

#### Phương thức thực hiện

* Xây dựng phần mềm hướng đối tượng bằng C#.
* Ứng dụng cơ sở dữ liệu SQL vào phần mềm.

### SQL, C#, Windows Form

#### Lý thuyết SQL cơ bản

SQL là viết tắt của cụm từ tiếng anh Structured Query Language, là ngôn ngữ truy vấn có cấu trúc. Nó là công cụ được thiết kế nhằm để quản lý dữ liệu sử dụng trong nhiều lĩnh vực, cho phép bạn truy cập và và thực hiện các thao tác lấy các hàng hoặc sửa đổi các hàng, trích xuất, tạo, xóa dữ liệu.

SQL cũng là ngôn ngữ tiêu chuẩn cho các hệ cơ sở dữ liệu quan hệ (RDBMS). Hầu hết các ngôn ngữ bậc cao đều lấy SQL làm tiêu chuẩn như MySQL (hệ quản trị cơ sở dữ liệu thường được sử dụng cho những dịch vụ thiết kế website giá rẻ bởi nó hoàn toàn miễn phí), MS Access, Oracle, Sybase, Visual C, Visual Basic, Postgres, Informix, SQL Server...

Nếu đọc thêm một vài thông tin về SQL Server của Microsoft bạn sẽ nhận thấy rằng ngoài SQL, thì T-SQL cũng được sử dụng như ngôn ngữ địa phương của hệ cơ sở dữ liệu này, Oracle SQL thì dùng kèm thêm PL/SQL…Các chương trình ứng dụng và công cụ của Oracle cho phép mọi người sử dụng truy cập thoải mái tới CSDL mà không cần thông qua SQL trực tiếp. Nhưng những ứng dụng này khi kích hoạt chạy thì phải sử dụng SQL.

Đặc điểm:

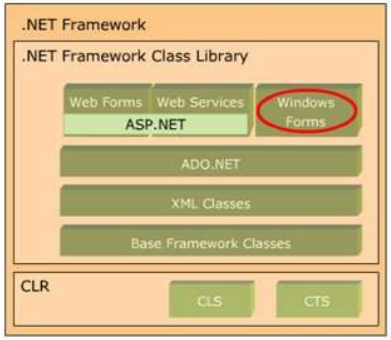
* SQL là ngôn ngữ như tiếng Anh nên các lập trình viên cũng cần có một số vốn ngoại ngữ về chuyên ngành.
* SQL là ngôn ngữ phi thủ tục, không đòi hỏi chúng ta về cách thức truy cập dữ liệu thế nào. Tất cả các thông báo của SQL đều rất dễ sử dụng và hầu như giảm thiểu tối đa khả năng xảy ra lỗi.
* SQL cung cấp các tập lệnh đa dạng cho việc hỏi đáp dữ liệu.
* Chèn, update, xóa các hàng trong 1 quan hệ.
* Tạo, xóa, sửa đổi và thêm các đối tượng trong của cơ sở dữ liệu.
* Điều khiển việc truy cập tới CSDL và các đối tượng của nó để đảm bảo tính năng bảo mật của CSDL.
* Luôn bảo đảm tính chất nhất quán và sự ràng buộc.
* Điều kiện tiên quyết để tiện lợi cho việc sử dụng các hỏi đáp là phải nắm vững được các cấu trúc cơ sở dữ liệu của mình.

Đối tượng làm việc của SQL: Là các bảng (tổng quan là các quan hệ dữ liệu 2 chiều). Các bảng này thường chứa nhiều cột (gọi là trường) và nhiều hàng (gọi là bản ghi). Cột với tên gọi và kiểu dữ liệu xác định tạo nên cấu trúc của bảng (kiểu dữ liệu của cột là dạng duy nhất). Ta có thể dùng lệnh Desc[ribe] TABLE-name để tra xem cấu trúc của bảng. Phần tùy chọn [] có thể được để trong Oracle. Khi bảng đã được hệ thống cho một mục đích nào đó ta có một cơ sở dữ liệu.

Cách thức thực hiện lệnh SQL: Khi bạn thao tác lệnh SQL trên bất kì CSDL quan hệ nào, hệ thống sẽ tự động chọn lọc quyết định cách thức tốt nhất để tiến hành lệnh đó. Và engine SQL sẽ giúp bạn hiểu rõ nhiệm vụ được thông dịch như thế nào.

Có rất nhiều thành phần trong suốt quá trình này phải kể đến như: Classic Query Engine, Query Dispatcher, SQL Query Engine, Optimization Engines.

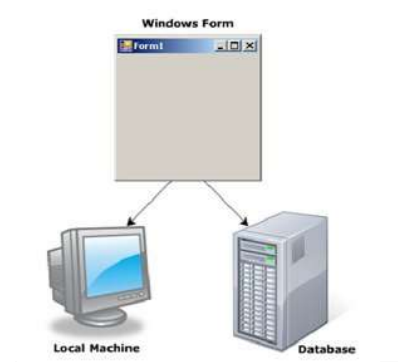
#### Lý thuyết Windows Form cơ bản

Windows Forms hay viết tắt là WinForm là thuật ngữ chỉ việc phát triển các ứng dụng giao diện người dùng bằng cách sử dụng các thành phần xây dựng sẵn (buil in component) còn được gọi là các điều khiển. Hay nói cách khác Windows Forms là một API (Application Programming Interface) cho phép tạo GUI (Graphical User Interface) cho các ứng dụng chạy trên desktop.

##### Hình 1: .NET Framework

Các điều khiển này dùng để hiển thị thông tin cho người dùng cũng như cho người dùng nhập thông tin vào Windows Forms ra đời đáp ứng nhu cầu tạo ứng dụng nhanh (Rapid Application Development).

Windows Form cho phép người phát triển tạo ra các giao diện người dùng sử dụng các thành phần khác nhau (components). Các thành phần này còn được gọi là các điều khiển (controls). Những điều khiển này cho phép chúng ta thu thập thông tin từ người dùng cũng như trình bày các thông tin để người dùng có thể xem.

 Một Form được chạy trên một máy tính cục bộ (local machine) và một form có thể truy cập đến các tài nguyên khác nhau như bộ nhớ, các thư mục, các tệp tin, các cơ sở dữ liệu…

##### Hình 2: Cơ chế hoạt động của windows form

Do đó Windows Form phù hợp cho các ứng dụng desktop như các ứng dụng quản lý thông tin, các ứng dụng tương tác trực tiếp với người dùng.

Vai trò của Windows Form:

• Các Form có thể chứa các điều khiển (các thành phần) khác nhau.

• Xử lý dữ liệu được nhập bởi người dùng.

• Hiển thị (trình bày) các thông tin tới người dùng.

• Kết nối đến các nguồn CSDL khác nhau trên các máy tính cục bộ hoặc máy tính khác

## Đặc tả phần mềm

### Phần mềm tạo, lưu ghi chú

#### Giới thiệu về phần mềm

Phần mềm tạo, lưu ghi chú là một phần mềm đơn giản, dễ sử dụng không cần kết nối Internet, có thể sử dụng nó ở mọi nơi. Tìm ghi chú nhanh chóng với tìm kiếm ngay lập tức và các thẻ đơn giản. Ghi chú của bạn được sao lưu khi bạn thay đổi chúng. Chỉ cần kéo thanh trượt phiên bản để quay ngược thời gian. Và đặc biệt nó hoàn toàn miễn phí.

#### Dữ liệu đầu vào – đầu ra

* Dữ liệu đầu vào: Database chứa thông tin User và Note.
* Dữ liệu đầu ra: Hiển thị thông tin của Note.

#### Tính năng chính

* Đăng nhập bằng tài khoản.
* Tạo ghi chú.
* Sửa thông tin ghi chú.
* Tìm kiếm ghi chú theo tên và Tag.
* Xóa và lấy lại những ghi chú đã xóa.
* Thay đổi Font chữ, màu chữ.
* Xuất và in ra bản ghi chú.
* Lưu trữ thông tin ghi chú của từng tài khoản.

#### Giao diện dự kiến

##### Hình 3: Màn hình đăng nhập

##### Hình 4: Giao diện chính

##### Hình 5: List Tag

### Yêu cầu kỹ thuật

* Thực hiện được yêu cầu mà đồ án đề ra.
* Áp dụng lập trình hướng đối tượng và cơ sở dữ liệu.
* Dung lượng phần mềm nhẹ, chạy ổn định.

### Công cụ và công nghệ sử dụng

* Xây dựng phần mềm bằng Windowns Form trên nền .NET Framework
* Thiết kế giao diện người dùng (GUI) bằng Windowns Form
* Thiết kế View Model và Data Model bằng Visual Studio 2019 (Code C#)

# Chương 2: Kế hoạch thực hiện

## Kế hoạch

###### Bảng 1: Kế hoạch theo tuần

|  |  |
| --- | --- |
| Tuần | Công việc |
| 5 | Tìm hểu về Entity Framework và Database-first, cách xây dựng database |
| 6 | Phân tích dữ liệu đầu vào, đầu ra, xây dựng database |
| 7 | Thiết kế giao diện ban đầu |
| 8 | Viết thuật toán kết nối Database vào chương trình, xây dựng mô hình MVC |
| 9 | Viết thuật toán chỉnh sửa thông tin trực tiếp trên giao diện và lưu vào database |
| 10 | Soát lỗi, kiểm thử chương trình |
| 11 | Kiểm tra chương trình lần cuối |
| 12 | Báo cáo lần 1 |
| 13 | Chỉnh sửa theo yêu cầu lần 1. Báo cáo lần 2 |
| 14 | Hoàn thiện chương trình. Báo cáo lần 3. Viết báo cáo 25% |
| 15 | Hoàn thành báo cáo. Kiểm tra bài báo cáo. Báo cáo cuối kỳ |

## Phân công công việc

###### Bảng 2: Phân công công việc

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên SV | Mô tả công việc | Đóng góp |
| 1 | Đặng Nguyễn Hoài Thư | * Xây dựng database * Viết code cho các tính năng của chương trình * Viết code cho các lớp trong controller * Cài đặt và kiểm thử * Chỉnh sửa báo cáo | 40% |
| 2 | Nguyễn Hữu Tường | * Thiết kế giao diện ban đầu * Viết code cho các tính năng của chương trình * Viết code cho các lớp trong controller * Cài đặt và kiểm thử * Chỉnh sửa báo cáo | 40% |
| 3 | Nguyễn Việt Long | * Viết báo cáo * Cài đặt và kiểm thử | 20% |

# Chương 3: Thiết kế phần mềm

## Thiết kế giao diện

Thanh cuộn all Note và Tag

Thanh công cụ

Nút xóa Note

Nút hiển thị thông tin Note

Tìm kiếm



Tiêu đề Note

##### Hình 6: Giao diện phần mềm

Các tag

Nơi hiển thị danh sách Note

Nơi hiển thị nội dung Note

## Thiết kế lớp

### Thiết kế lớp cho Layer

#### Chi tiết

##### Hình 7: UML Diagram biểu diễn các lớp trong phần mềm

#### Đặc tả lớp

###### Bảng 3: Danh mục các lớp cho chương trình

|  |  |  |
| --- | --- | --- |
| TT | Tên lớp | Mục đích |
| 1 | Note\_backupcontroller | Xử lý cơ sở dữ liệu của bảng Note\_Backup |
| 2 | Notecontrollers | Xử lý cơ sở dữ liệu của bảng Note |
| 3 | Tag\_Notecontroller | Xử lý cơ sở dữ liệu của bảng Note\_Tag |
| 4 | Tagcontroller | Xử lý cơ sở dữ liệu của bảng Tag |
| 5 | User\_Note\_backupcontroller | Xử lý cơ sở dữ liệu của bảng User\_Note\_Backup |
| 6 | User\_Notecontrollers | Xử lý cơ sở dữ liệu của bảng User\_Note |
| 7 | Usercontrollers | Xử lý cơ sở dữ liệu của bảng User |

#### Đặc tả phương thức trong lớp

###### Bảng 4: Đặc tả các phương thức trong lớp Note\_backupcontroller

|  |  |  |
| --- | --- | --- |
| TT | Phương thức | Mục đích |
| 1 | public static bool AddNote\_Backup(Note\_Backup note\_backup) | Thêm Note đã xóa vào bảng Note\_Backup |
| 2 | public static Note\_Backup LoadNote\_Backup(int ID) | Lấy Note đã xóa theo ID của Note |
| 3 | public static bool DeleteNote\_Backup(Note\_Backup note\_backup) | Xóa Note vĩnh viễn |

###### Bảng 5: Đặc tả các phương thức trong lớp Notecontrollers

|  |  |  |
| --- | --- | --- |
| TT | Phương thức | Mục đích |
| 1 | public static int getIDNotes() | Lấy ID lớn nhất trong bảng Note |
| 2 | public static List<int> getListNoteID() | Lấy danh sách ID trong bảng Note |
| 3 | public static bool AddNote(Note note) | Thêm Note mới vào trong bảng Note |
| 4 | public static bool UpdateNote(Note note) | Cập nhật thông tin chỉnh sửa của một Note |
| 5 | public static Note LoadNote(int ID) | Lấy Note theo ID của Note |
| 6 | public static bool DeleteNote(Note note) | Xóa Note trong bảng Note |
| 7 | public static List<Note> getListNote\_searchHeader(string header) | Lấy danh sách những Note có chứa header trong bảng Note |

###### Bảng 6: Đặc tả các phương thức trong lớp Tag\_Notecontroller

|  |  |  |
| --- | --- | --- |
| TT | Phương thức | Mục đích |
| 1 | public static bool AddNote\_Tag(Note\_Tag note\_tag) | Thêm Note\_Tag(Tag và ID của Note) vào bảng Note\_Tag |
| 2 | public static bool DeleteNote\_Tag(Note\_Tag note\_tag) | Xóa Note\_Tag(Tag và ID của Note) trong bảng Note\_Tag |
| 3 | public static List<Note\_Tag> getListNote\_searchTag(string tag) | Lấy danh sách Tag có chứa tag trong bảng Note\_Tag |
| 4 | public static List<Note\_Tag> LoadNoteofTag(string tag) | Lấy danh sách tất cả các Note\_Tag trong bảng Note\_Tag |
| 5 | public static List<Note\_Tag> GetTag(int ID) | Lấy Note\_Tag theo ID của Note |

###### Bảng 7: Đặc tả các phương thức trong lớp Tagcontroller

|  |  |  |
| --- | --- | --- |
| TT | Phương thức | Mục đích |
| 1 | public static bool AddTag(Tag tag) | Thêm Tag vào bảng Tag |
| 2 | public static List<Tag> LoadTag() | Lấy danh sách tất cả các Tag trong bảng Tag |
| 3 | public static bool DeleteTag(Tag tag) | Xóa Tag trong bảng Tag |

###### Bảng 8: Đặc tả các phương thức trong lớp User\_Note\_backupcontroller

|  |  |  |
| --- | --- | --- |
| TT | Phương thức | Mục đích |
| 1 | public static bool AddUser\_Note\_Backup(User\_Note\_Backup user\_note\_backup) | Thêm User\_Note (Tên User và ID của Note) đã xóa vào bảng User\_Note\_Backup |
| 2 | public static List<User\_Note\_Backup> LoadNoteofUser\_Backup(string username) | Lấy danh sách User\_Note đã xóa trong bảng  User\_Note\_Backup |
| 3 | public static bool DeleteUser\_Note\_Backup(User\_Note\_Backup user\_note\_backup) | Xóa User\_Note vĩnh viễn trong cơ sở dữ liệu |

###### Bảng 9: Đặc tả các phương thức trong lớp User\_Notecontrollers

|  |  |  |
| --- | --- | --- |
| TT | Phương thức | Mục đích |
| 1 | public static bool AddUser\_Note(User\_Note user\_note) | Thêm User\_Note (Tên User và ID của Note) vào bảng User\_Note |
| 2 | public static bool UpdateUser\_Note(User\_Note user\_note) | Cập nhật thông tin của User\_Note (Tên User và ID của Note) đã chỉnh sửa vào bảng User\_Note |
| 3 | public static List<User\_Note> LoadNoteofUser(string username) | Lấy danh sách tất cả các User\_Name có cùng username trong bảng User\_Note |
| 4 | public static bool DeleteUser\_Note(User\_Note user\_note) | Xóa user\_note trong bảng User\_Note |
| 5 | public static bool checkUser\_Note(User\_Note user\_note) | Kiểm tra user\_note có tồn tại trong bảng User\_Note hay không? |

###### Bảng 10: Đặc tả các phương thức trong lớp Usercontrollers

|  |  |  |
| --- | --- | --- |
| TT | Phương thức | Mục đích |
| 1 | public static bool AddUser(User user) | Thêm một user vào bảng User |
| 2 | public static User getUser(string username) | Lấy một User theo username trong bảng User |
| 3 | public static bool checkPassword(string username,string password) | Kiểm tra mật khẩu của User đó có đúng hay không? |

### 

### Thiết kế cơ sở dữ liệu

#### Các bảng của cơ sở dữ liệu

###### Bảng 11: Các bảng trong cơ sở dữ liệu

|  |  |  |
| --- | --- | --- |
| TT | Tên bảng | Mục đích |
| 1 | Note\_Backup | Chứa thông tin của Note đã xóa tạm thời |
| 2 | Note | Chứa thông tin của Note |
| 3 | Note\_Tag | Chứa thông tin của Tag và Note tương ứng |
| 4 | Tag | Chứa thông tin của Tag trong cơ sở dữ liệu |
| 5 | User\_Note\_Backup | Chứa thông tin của User và Note tương ứng đã xóa tạm thời |
| 6 | User\_Note | Chứa thông tin của User và Note tương ứng |
| 7 | User | Chứa thông tin của User |

#### Các trường trong bảng

###### Bảng 12: Các trường trong bảng Note\_Backup

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | Header | nvarchar(100) | Chứa tiêu đề của Note đã xóa tạm thời |
| 2 | Context | nvarchar(MAX) | Chứa nội dung của Note đã xóa tạm thời |
| 3 | Time | nchar(100) | Chứa thời gian tạo của Note đã xóa tạm thời |
| 4 | TimeEdit | nchar(100) | Chứa thời gian chỉnh sửa của Note đã xóa tạm thời |

###### Bảng 13: Các trường trong bảng Note

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | Header | nvarchar(100) | Chứa tiêu đề của Note |
| 2 | Context | nvarchar(MAX) | Chứa nội dung của Note |
| 3 | Time | nchar(100) | Chứa thời gian tạo của Note |
| 4 | TimeEdit | nchar(100) | Chứa thời gian chỉnh sửa của Note |

###### Bảng 14: Các trường trong bảng Note\_Tag

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | ID | int | Chứa ID của Note |
| 2 | MiniTag | nvarchar(100) | Chứa nội dung Tag của Note đó |
| 3 | Description | nchar(10) | Chứa mô tả của Note đó |

###### Bảng 15: Các trường trong bảng Tag

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | Tag | nvarchar(100) | Chứa nội dug cùa Tag |

###### Bảng 16: Các trường trong bảng User\_Note\_Backup

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | ID | int | Chứa ID của Note |
| 2 | Username | nchar(25) | Chúa tên người đăng nhập ứng với Note đã xóa tạm thời |
| 3 | Description | nchar(10) | Chứa mô tả ứng với Note đã xóa tạm thời |

###### Bảng 17: Các trường trong bảng User\_Note

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | ID | int | Chứa ID của Note |
| 2 | Username | nchar(25) | Chúa tên người đăng nhập ứng với Note |
| 3 | Description | nchar(10) | Chứa mô tả ứng với Note |

###### Bảng 18: Các trường trong bảng User

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên trường | Kiểu dữ liệu | Mục đích |
| 1 | Username | nchar(25) | Chứa tên người dùng đã đăng ký |
| 2 | Password | nchar(50) | Chứa mật khẩu của người dùng |
| 3 | Email | nchar(100) | Chứa emai của người dùng |
| 4 | Phone | nchar(50) | Chứa số điện thoại của người dùng |

### Đặc tả nút chức năng

###### Bảng 19: Các nút chức năng

|  |  |  |
| --- | --- | --- |
| TT | Nút chức năng | Mục đích |
| 1 |  | Hiển thị thanh Menu: Gồm có chế độ All Notes, Trash và danh sách tất cả các Tag có trong cơ sở dữ liệu |
| 2 |  | Thêm một Note mới |
| 3 |  | Thay đổi chế độ xem của Note đang chọn |
| 4 |  | Thay đổi Font chữ hoặc màu chữ của Note |
| 5 |  | Xóa một Note tạm thời, Note được thêm vào danh sách Trash |
| 6 |  | Hiển thị thêm thông tin của Note đang chọn (Ngày tạo, chỉnh sửa Note,…) |
| 7 |  | Xóa một Note khỏi danh sách Trash vĩnh viễn |
| 8 |  | Hồi phục lại Note đã xóa trong danh sách Trash |

# Chương 4: Kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tình huống | Mục đích | Mô tả |
| Kiểm thử 1: Đăng ký User | Hình 8: Kiểm thử 1 | Đăng ký tài khoản | Giúp người dùng có thể đăng ký tài khoản, để đăng nhập và sử dụng ứng dụng, màn hình thông báo đã đăng ký thành công |
| Kiểm thử 2: | Hình 9: Kiểm thử 2 | Đăng nhập | Sau khi đăng ký tài khoản, người dùng đăng nhập tài khoản đã đăng ký để sử dụng phần mềm, màn hình thông báo đã đăng nhập thành công |
| Kiểm thử 3: | Hình 10: Kiểm thử 3 | Hiển thị danh sách các Note đã có | Sau khi đăng nhập, chương trình sẽ tiến hành tải lại các Note mà người dùng đã tạo trước đó và hiển thị lên ListView |
| Kiểm thử 4 | Hình 11: Kiểm thử 4 | Hiển thị nội dung khi click vào Note đó | Khi click vào một Note, nội dung của Note đó sẽ hiển thị, ta có thể chỉnh sửa, thay đổi Font chữ, màu chữ hoặc xóa Note đó |
| Kiểm thử 5 | Hình 12: Kiểm thử 5 | Tìm kiếm Note | Người dùng có thể tìm kiếm Note theo Tag hoặc theo header của Note đó |
| Kiểm thử 6 | Hình 13: Kiểm thử 6 | Thay đổi chế độ xem | Người dùng có thể thay đổi chế độ xem của một Note, có thể phóng to hoặc thu nhỏ |
| Kiểm thử 7 | Hình 14: Kiểm thử 7 | Tạo Note mới | Người dùng có thể tạo một Note mới |
| Kiểm thử 8 | Hình 15: Kiểm thử 8 | Xóa một Note | Sau khi xóa một Note, Note đã xóa hiển thị trong danh sách Trash, người dùng có thể xóa vĩnh viễn hoặc hồi phục lại Note |
| Kiểm thử 9 | Hình 16: Kiểm thử 9 | Hiển thị danh sách các Tag đã có khi click vào Menu | Sau khi click vào nút Menu, chương trình sẽ hiển thị danh sách tất cả các Tag đã được người dùng tạo khi thêm Note. |
| Kiểm thử 10 | Hình 17: Kiểm thử 10 | Hiển thị các Note chứa Tag đã tạo trước đó | Khi click vào các Tag trong danh sách Tag, người dùng có thể xem được các Note của Tag đã chọn, có thể chỉnh sửa và xóa được. |

# Chương 5: Kết luận

## Kết luận:

Về cơ bản, nhóm tự nhận xét phần mềm đã giải quyết được 95% so với yêu cầu của đồ án đặt ra, phần mềm được thiết kế theo mô hình MVC. Trong quá trình thực hiện đồ án, nhóm em gặp phải một số khó khăn như: Khó khăn trong việc thiết kế dữ liệu cho đồ án, khó khăn trong quá trình viết code để hiển thị và quản lí note trên listview. Nhưng may mắn, nhờ sự giải đáp, hướng dẫn tận tình của thầy, cũng như là sự tham khảo trên internet, nhóm chúng em đã khắc phục được các khó khăn nói trên. Và sau đây là một số ưu điểm cũng như nhược điểm của phần mềm mà nhóm em đã làm:

* Ưu điểm:
  + Giao diện thân thiện với người dùng, hệ thống được xây dựng gọn nhẹ, dễ sử dụng.
  + Chương trình chạy ổn định, cho ra kết quả chính xác.
  + Hệ thống ràng buộc dữ liệu được đảm bảo.
* Nhược điểm:
* Chương trình tiêu tốn nhiều tài nguyên của hệ thống khi hoạt động.
* Bắt lỗi còn hạn chế, chưa hoàn thiện.

## Hướng phát triển:

* Thêm nhiều tính năng mới để đáp ứng yêu cầu sử dụng của người dùng.
* Hoàn thiện tốt hơn về lập trình C# và ràng buộc dữ liệu.
* Bắt lỗi tốt hơn, hoàn thiện hơn.
* Tối ưu hóa thuật toán giúp chương trình tiêu tốn ít tài nguyên hệ thống

# Tài liệu tham khảo

[1]. <https://simplenote.com/>

[2].<https://stackoverflow.com/questions/21327931/get-the-index-of-item-selected-in-listview>

[3]. Loạt video thực hành của thầy Huỳnh Xuân Phụng

[3.1]. <https://www.youtube.com/watch?v=xJlJKvbYtXU&t=6s>

[3.2]. <https://www.youtube.com/watch?v=i-Gzq5c-BNM>

[3.3]. <https://www.youtube.com/watch?v=lyOJPvWPHFM>