TRƯỜNG ĐẠI HỌC

**SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

HCMC University of Technology and Education

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**XÂY DỰNG CHƯƠNG TRÌNH**

**TẠO LƯU GHI CHÚ**

|  |  |
| --- | --- |
| **Nhóm sinh viên thực hiện:** |  |
| **Võ Hữu Đức** | **17110285** |
| **Trần Nhật Trường** | **15110353** |
| **Phạm Ngọc Hùng** | **18133017** |

****

**GVHD: TS. Huỳnh Xuân Phụng**

**Tp. Hồ Chí Minh, tháng 6 – 2020**

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)***

**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 Công Nghệ Thông Tin 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 20 tháng 6 năm 2020**

**Nhóm sinh viên thực hiện**

MỤC LỤC

[**Danh mục các hình** 5](#_Toc44108043)

[**Danh mục các bảng** 6](#_Toc44108044)

[**Chương 1: Tổng quan chương trình** 7](#_Toc44108045)

[***1. Giới thiệu chung*** 7](#_Toc44108046)

[**1.1.** **Về đồ án phần mềm tạo lưu ghi chú** 7](#_Toc44108047)

[**1.1.1. Yêu cầu đồ án** 7](#_Toc44108048)

[**1.1.2. Phân tích đồ án** 7](#_Toc44108049)

[**1.1.3. Phương hướng thực hiện** 7](#_Toc44108050)

[**1.2.** **Entity Framework và mô hình MVC** 7](#_Toc44108051)

[**1.2.1. Entity Framework** 7](#_Toc44108052)

[**1.2.2. Kiến trúc Entity Framework** 7](#_Toc44108053)

[**1.2.3. Các hướng tiếp cận khi xây dựng EDM** 8](#_Toc44108054)

[**1.2.4. Mô hình MVC** 9](#_Toc44108055)

[**1.2.5. Ưu và nhược điểm khi sử dụng mô hình MVC** 10](#_Toc44108056)

[***2. Đặc tả phần mềm Tạo lưu ghi chú*** 11](#_Toc44108057)

[**2.1.** **Phần mềm Tạo lưu ghi chú** 11](#_Toc44108058)

[**2.1.1. Giới thiệu về phần mềm Tạo lưu ghi chú** 11](#_Toc44108059)

[**2.1.2. Tính năng chính** 11](#_Toc44108060)

[**2.1.3. Ứng dụng** 11](#_Toc44108061)

[**2.2. Yêu cầu kĩ thuật** 12](#_Toc44108062)

[**2.3. Công cụ và công nghệ sử dụng** 12](#_Toc44108063)

[**Chương 2: Kế hoạch thực hiện** 13](#_Toc44108064)

[***1. Kế hoạch*** 13](#_Toc44108065)

[***2. Phân công công việc*** 13](#_Toc44108066)

[**Chương 3: Thiết kế phần mềm** 14](#_Toc44108067)

[***1. Thiết kế giao diện*** 14](#_Toc44108068)

[**1.1.** **Giao diện chương trình** 14](#_Toc44108069)

[**1.2.** **Đặc tả giao diện** 14](#_Toc44108070)

[***2. Thiết kế lớp chức năng*** 15](#_Toc44108071)

[**2.1. Tổng quan** 15](#_Toc44108072)

[**2.2. Đặc tả lớp chức năng** 15](#_Toc44108073)

[**2.3. Đặc tả các phương thức của lớp** 16](#_Toc44108074)

[**Chương 4: Cài đặt và kiểm thử** 21](#_Toc44108075)

[**Chương 5: Kết luận và hướng phát triển** 23](#_Toc44108076)

[***1. Kết luận*** 23](#_Toc44108077)

[***Hướng phát triển*** 23](#_Toc44108078)

[**Tài liệu tham khảo** 24](#_Toc44108079)

# **Danh mục các hình**

**Hình 1: Mô hình Entity Framework** **7**

**Hình 2: Mô hình MVC** **9**

**Hình 3: Giao diện phần mềm.** **14**

**Hình 4: Kiểm thử 1.** **21**

**Hình 5: Kiểm thử 2** **21**

**Hình 6: Kiểm thử 3** **22**

**Hình 7: Kiểm thử 4.** **22**

# **Danh mục các bảng**

**Bảng 1: Kế hoạch theo tuần** **13**

**Bảng 2: Phân công công việc & đóp góp của mỗi sinh viên** **13**

**Bảng 3: Đặc tả giao diện** **14**

**Bảng 4: Đặc tả lớp chức năng** **15**

**Bảng 5: Đặc tả các phương thức của lớp** **16**

# **Chương 1: Tổng quan chương trình**

## ***1. Giới thiệu chung***

## **1.1.** **Về đồ án phần mềm tạo lưu ghi chú**

### **1.1.1. Yêu cầu đồ án** Thiết kế và xây dựng phần mềm cho phép người dung tạo ghi chú và thực hiện các chững năng khác như them, xóa, tìm kiếm ghi chú được sử dụng bằng ngôn ngữ C# và mô hình MVC thông qua thư viện EntityFramework.

### **1.1.2. Phân tích đồ án**

* Xây dựng phần mềm ghi chú
* Cho phép người dung nhập dữ liệu từ bàn phím để lưu nội dung của Note
* Người dùng có thể them, xóa, tìm kiếm những ghi chú mà mình đã Note

### **1.1.3. Phương hướng thực hiện**

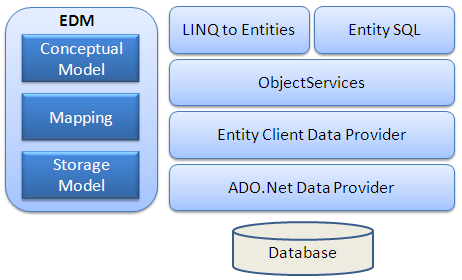
Xây dựng phần mềm bằng ngôn ngữ C# và mô hình MVC

## **1.2.** **Entity Framework và mô hình MVC**

### **1.2.1. Entity Framework**

* Entity Framework là môt thư viện ORM được Microsoft phát triển năm 2008. ORM là viết tắt của Object – Relational – Mapper. ORM là tập hợp các công nghệ cho phép làm việc với các hệ quản trị cơ sở dữ liệu quan hệ.
* Là công cụ làm việc với cơ sở dữ liệu được Microsoft khuyến nghị

**1.2.2. Kiến trúc Entity Framework**

****

***Hình 1: Mô hình Entity Framework***

* **Entity Data Model(EDM)**

Là thành phần lưu trữ thông tin ánh xạ (Mapping) giữa các class và cơ sở dữ liệu. Nó đóng vài trò cầu nối giữa các lớp Model và cơ sở dữ liệu nhưng lại giúp chương trình hoàn toàn tách biệt với cơ sở dữ liệu

EDM có 3 thành phần:

* Conceptual model: Chịu trách nhiệm lưu thông tin giữa các lớp Model và quan hệ giữa chúng
* Storage model: lưu thông tin về cơ sở dữ liệu, bao gồm các bảng, view, stored procedure, khóa, quan hệ;
* Mapping model: lưu thông tin ánh xạ giữa conceptual và storage model
* **LinQ to Entities và Entity SQL**
* Dây là 2 ngôn ngữ dùng để truy vấn dữ liệu
* Entity SQL được tạo ra làm ngôn ngữ truy vấn sau đó LinQ mới ra đời và đội ngũ phát triển Entity Framework đưa LinQ to Entities vào EF
* **Object services**
* Đây là cổng để truy xuất dữ liệu. Tầng này thực hiện quá trình gọi là materialization – quá trình chuyển dổi dữ liệu dạng bảng nhận từ tầng dưới về dạng object.
* Ngoài ra tầng này cũng đảm nhiệm việc quản lí trạng thái
* Entity Client data provider
* Tầng này chịu trách nhiệm tương tác với Data provider của ADO.NET. Tầng này thực hiện chuyển đổi truy vẫn LinQ to Entities học Entity SQL vè truy vấn SQL. Sau đó Data provinder sẽ thực thi truy vấn SQL.
* Một nhiệm vụ khác của Entity Client là chuyển dổi kết quả thu được từ truy vấn SQL về một dạng trung gian. Kết quả này chuyển lên cho Object Services xử lý.

### **1.2.3. Các hướng tiếp cận khi xây dựng EDM**

* **Database-first**

Dây là hướng tiếp cận xuất hiên đầu tiên trong Entity Framework. Hướng tiếp cận náy sẽ giúp sinh ra Model từ cơ sở dữ liệu có sẵn, từ stored procedures hoặc views. Chỉ cần chỉnh lại model (domain classes) theo nhu cầu của người dùng. Việc tinh chỉnh này thực hiện qua môt giao diện đồ họa. Khi bạn xuất phát từ Storage Model có sẵn, EF giúp bạn sinh ra Conceptual Model và Mapping

* **Model-first**

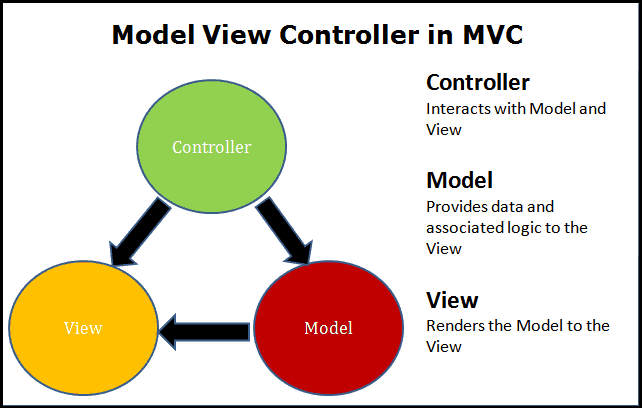
Đây là hướng tiếp cận được sử dụng rộng rãi nhất. Hướng tiếp cận này cho phép bạn sử dụng một giao diện đồ họa để thiết kế model trước. Sau đó trình thiết kế này sẽ sinh ra cái model tương ứng cũng như mã SQL để tao ra database. Bạn xuất phát từ giao diện thiết kế riêng biệt, EF giúp bạn sinh ra 3 thàn phẩn của EDM

* **Code-first**

Đây là hướng tiếp cận được lập trình viên hoan nghênh nhất. Code-first xuất hiện từ EF4.1. Với vai trò lập trình viện, chúng ta không cần quan tâm đến SQL hay giao diện thiết kế nữa. Tất cả những gì cần làm là xây dựng các lớp model theo đúng kiểu hướng đối tượng mà các bạn quen thuộc nhất. Entity Framework sẽ làm tất cả những gì còn lại để bạn có một database phù hợp. Tự mình code Conceptual Model, EF sẽ giúp bạn sinh ra Mapping và Storage Model

### **1.2.4. Mô hình MVC**

MVC là từ viết tắt bởi 3 từ Model – View – Controller. Đây là mô hình thiết kế sử dụng trong kỹ thuật phần mềm. Mô hình source code thành 3 phần, tương ứng mỗi từ. Mỗi từ tương ứng với một hoạt động tách biệt trong một mô hình.



# ***Hình 2: Mô hình MVC***

* **Model**:

Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…

* **View**:

Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website.

Thông thường, các ứng dụng web sử dụng MVC View như một phần của hệ thống, nơi các thành phần HTML được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi.

* **Controller**

Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, Controller đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

**1.2.5. Ưu và nhược điểm khi sử dụng mô hình MVC**

* **Ưu điểm:**
* Nhẹ, tiết kiệm băng thông: MVC không sử dụng viewstate nên khá tiết kiệm diện tích băng thông. Khi sử dụng, người dùng có thể sử dụng ứng dụng trên web cần tương tác gửi và nhận dữ liệu một cách liên tục. Do đó, việc giảm băng thông giúp cho website hoạt động tốt và ổn định hơn.
* Kiểm tra dễ dàng: Với MVC, bạn có thể dễ dàng kiểm tra, rà soát lỗi phần mềm trước khi tới tay người tiêu dùng, đảm bảo chất lượng và độ uy tín cao hơn.
* Chức năng control: Trên các nền website thì ngôn ngữ lập trình như CSS, HTML, Javascript có một vai trò vô cùng quan trọng. Việc sử dụng mô hình MVC sẽ giúp bạn có một bộ control ưu việt trên nền tảng các ngôn ngữ hiện đại với nhiều hình thức khác nhau.
* View và size: View sẽ là nơi lưu trữ các dữ liệu. Càng nhiều yêu cầu được thực hiện thì kích thước càng tệp càng lớn. Khi đó, đường truyền mạng cũng giảm tốc độ load. Việc sử dụng mô hình MVC sẽ giúp bạn tiết kiệm được diện tích băng thông một cách tối ưu.
* Chức năng Soc (Separation of Concern): Chức năng này cho phép bạn phân tách rõ ràng các phần như Model, giao diện, data, nghiệp vụ.
* Tính kết hợp: Việc tích hợp ở mô hình MVC cho phép bạn thoải mái viết code trên nền tảng website. Khi đó, server của bạn sẽ được giảm tải khá nhiều.
* Đơn giản: Đây là một mô hình với kết cấu tương đối đơn giản. Dù bạn không có quá nhiều chuyên môn cũng có thể sử dụng được.
* **Nhược điểm:**

MVC thường được sử dụng vào những dự án lớn. Do đó, với các dự án nhỏ, mô hình MVC có thể gây cồng kềnh, tốn thời gian trong quá trình phát triển cũng như thời gian trung chuyển dữ liệu.

# ***2. Đặc tả phần mềm Tạo lưu ghi chú***

## **2.1.** **Phần mềm Tạo lưu ghi chú**

### **2.1.1. Giới thiệu về phần mềm Tạo lưu ghi chú**

Với sự bùng nổ thông tin trong thời đại cách mạng công nghệ 4.0 như hiện này thì việc lưu trữ thông tin và dữ liệu là rất quan trọng. Hiện nay các công cụ lưu trự rất phổ biến cho người dùng, một trong số đó là phần mềm tạo lưu ghi chú. Ứng dung cho phép người dùng tạo và lưu trữ những thông tin quan trọng trong cuộc sống hằng ngày.

### **2.1.2. Tính năng chính**

* Tạo một ghi chú mới
* Thêm văn bản
* Xóa các ghi chú đã tạo
* Tìm kiếm các ghi chú theo các tiêu chí

### **2.1.3. Ứng dụng**

Giúp cho người dùng có thể lưu trữ những thông tin, dữ liệu quan trọng và cần thiết.

## **2.2. Yêu cầu kĩ thuật**

* Thực hiện được yêu cầu mà đồ án đề ra.
* Áp dụng mô hình MVC và các công nghệ phần mềm mới.
* Dung lượng phần mềm nhẹ, chạy ổn định.

## **2.3. Công cụ và công nghệ sử dụng**

* Xây dựng phần mềm trên nền tảng ADO.NET với ngôn ngữ C#
* Thiết kế giao diện và thực thi chương trình trên Visual Studio 2019
* Thiết kế cơ sở dữ liệu bằng SQL Server 2019
* Áp dụng mô hình MVC

|  |  |
| --- | --- |
|  | **Chương 2: Kế hoạch thực hiện** |
| ***1. Kế hoạch*** | |
|  | ***Bảng 1: Kế hoạch theo tuần*** |
|  |  |
| Tuần | Công việc |
|  |  |
| 5 | Tìm hiểu về Entity Framwork |
|  |  |
| 6 | Tìm hiểu về mô hình MVC |
|  |  |
| 7 | Phân tích input, output. Bắt đầu thiết kế giao diện |
|  |  |
| 8 | Xây dựng và cài đặt các lớp chức năng của phần mềm |
|  |  |
| 9 | Xây dựng và cài đặt thuật toán |
|  |  |
| 10 | Xây dựng, cài đặt chức năng tìm kiếm, hoàn thành phần mềm. Soát lỗi, kiểm thử phần mềm. |
|  |  |
| 11 | Viết báo cáo và làm file trình chiếu phục vụ việc báo cáo và thuyết trình đồ án |
|  |  |

# ***2. Phân công công việc***

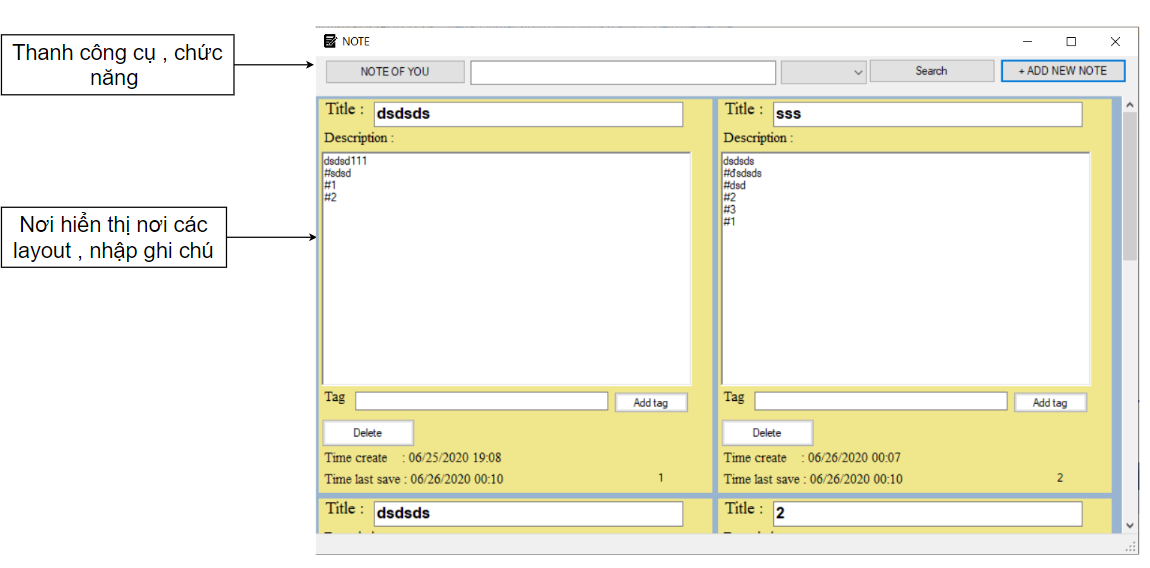
***Bảng 2: Phân công công việc & đóp góp của mỗi sinh viên***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Tên sinh viên |  | Miêu tả công việc | Đóng góp |
|  |  |  |  |  |
| 1 |  |  | * Thiết kế chính các lớp của NoteApp |  |
|  | Võ Hữu Đức |  | * Thiết kế chính các lớp chức năng cho phần mềm |  |
|  |  |  | * Thiết kế thuật toán | 60% |
|  |  |  |  |  |
|  |  |  | |  |
|  |  |  |  |  |
| 2 | Trần Nhật Trường |  | * Thiết kế giao chính giao diện phần mềm và báo cáo | 20% |
|  |  |  |  |  |
| 3 | Phạm Ngọc Hùng |  | * Báo cáo và cài đặt và kiểm thử. * Hỗ trợ thiết kế thuật toán và giao diện. | 20% |
|  |  |  |  |  |

# **Chương 3: Thiết kế phần mềm**

# ***1. Thiết kế giao diện***

* 1. **Giao diện chương trình**



***Hình 3: Giao diện phần mềm***

* 1. **Đặc tả giao diện**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Phân Loại** | **Make-Up** | **Chức Năng – Chú Thích** |
| 1 | btShowNote | Button | fpnButton.Controls.Clear();  using (var ctx = new NoteAppEntities1())  {t = ctx.Notes .SqlQuery  ("Select \* from Note") .ToList<Note>();  }  foreach (var item in t)  {  fpn z = new fpn();  z.h = z.fpnnote(item); fpnButton.Controls.Add(z.h);  } | Hiển thị tất cả các ghi chú đã được lưu lại lên form |
| 2 | txtSearch | Textbox | Được cài đặt ở phần design. | Nơi người dùng nhập thông tin cần tìm kiếm . |
| 3 | cbSearch | Combobox | Được cài đặt ở phần design gồm các item ( ID , TITLE,DESCRIPTION,TAG ) | Chọn thuộc tính cho chức năng search |
| 4 | btaddNote | Button | Sonote();  int g = 1;  for (int i = 0; i < sonote; i++)  {  if (g != t[i].id)  {  break;  }  else  g++;  }  fpn p = new fpn();  Note z = new Note();  p = x.Getfpn(g); fpnButton.Controls.Add(p.h);  z = x.Getnote(p); db.Notes.Add(z); db.SaveChanges(); p.txtTitle.Focus(); |  |
| **5** | fpnButton | Flow Layout Panel | Được cài đặt ở phần design. | Nơi hiển thị Flow layout panel ghi chú. |

***Bảng 3: Đặc tả giao diện***

# ***2. Thiết kế lớp chức năng***

## **2.1. Tổng quan**

Dựa theo yêu cầu đồ án , thiết kế giao diện và các chức năng cần thiết với sự vận hành của chương trình , do sự hạn chế về thời gian và kiến thức của chúng em. Phần mềm này chúng em cài đặt một số chức năng chính của đồ án và hoàn thiện nó.

## **2.2. Đặc tả lớp chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lớp** | **Thuộc tính** | **Kế thừa** | **Mô tả** |
| **fpn** | FlowLayoutPanel h ,  Label lbTimeCreate,  Label lbtimeLastSave,  Label lbTitle ,  TextBox txtTitle,  Label lbDescription,  RichTextBox rtDescription,  Label lbtag,  TextBox txttag,  Button btaddtag,  Button btdelete ,  Label lbstt ,  DateTime timelastsave,  DateTime timecreate |  | Tạo khung Flow layout panel cho note. Bao gồm các label, textbox, richtextbox,button để tương tác trực tiếp với ghi chú khi tạo hoặc đã được lưu. |
| **Note** | int id  string tieude  string noidung  DateTime createdatetime  DateTime lasttimesave  string tag |  | Lấy dữ liệu từ lớp fpn khi tương tác trên giao diện để lưu xuống CSDL và lấy dữ liệu từ CSDL load vào fpn. |

***Bảng 4: Đặc tả lớp chức năng***

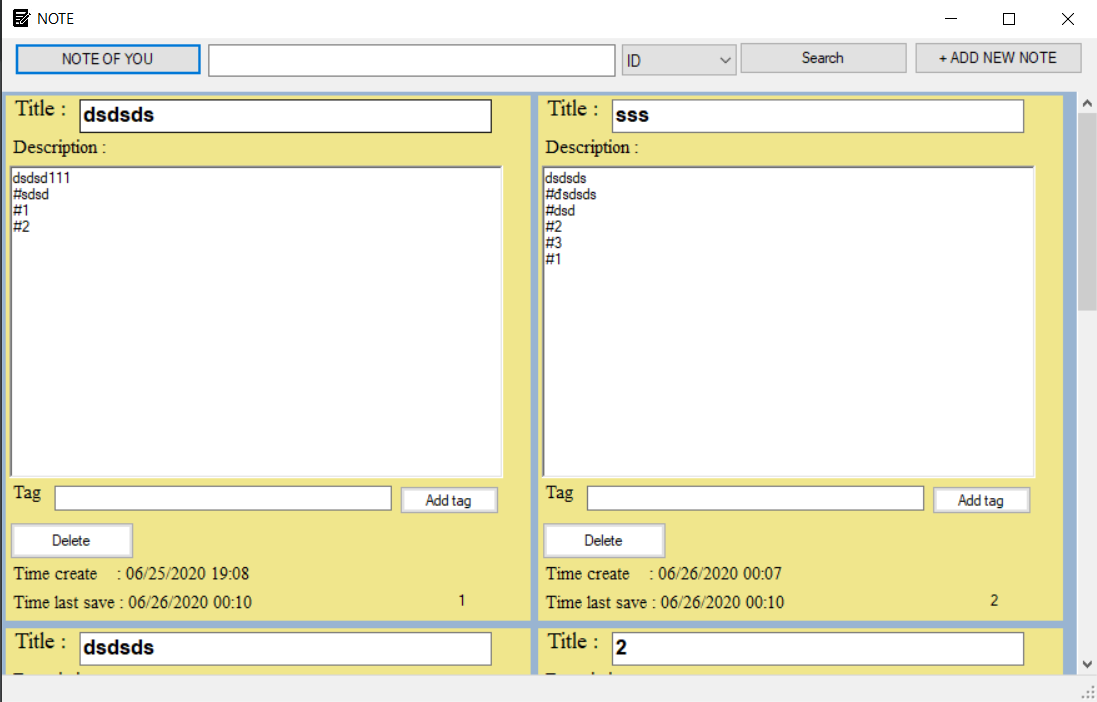
## **2.3. Đặc tả các phương thức của lớp**

-

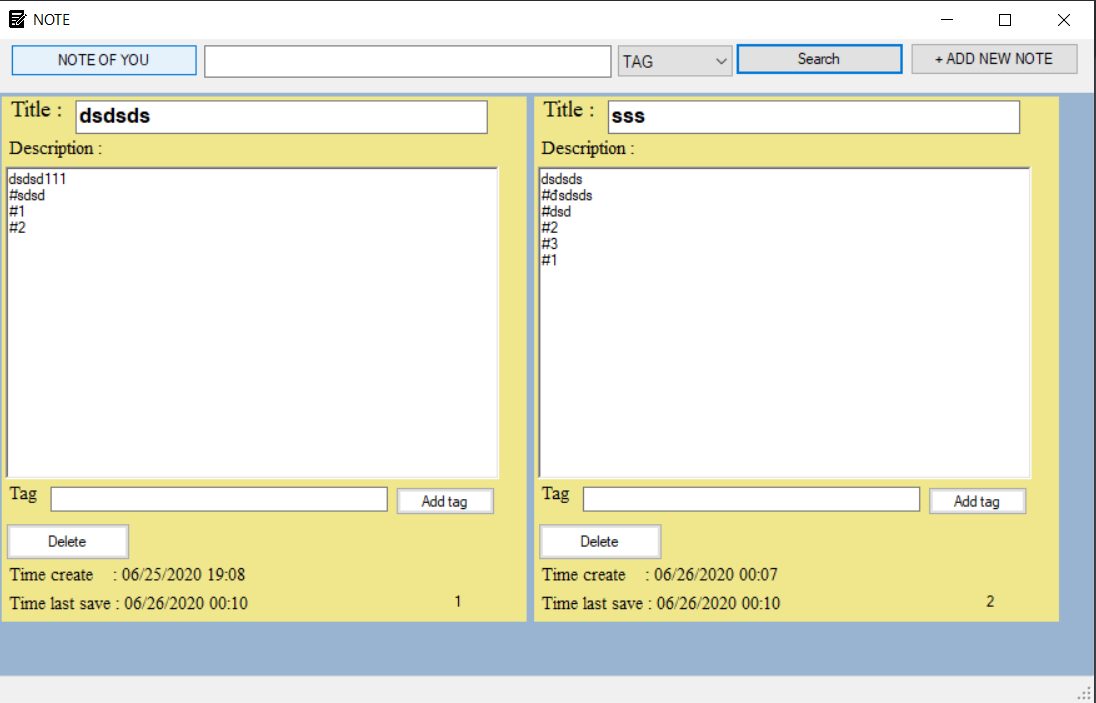
|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Phương thức** | **Mục đích** | **Ghi chú** |
| 1 | public void Taolbstt(Note t)  {  lbstt = new Label();  lbstt.Width = 20;  lbstt.Text = t.id.ToString();  } | Tạo label hiển thị số thứ tự (cũng chính là id của note) |  |
| 2 | public void Taolbcreatetime(Note t)  {  lbTimeCreate = new Label();  lbTimeCreate.Text = "Time create : " + t.createdatetime.ToString("MM/dd/yyyy HH:mm");  lbTimeCreate.Font = new Font("Times New Roman", 10, FontStyle.Regular);  lbTimeCreate.Width = 400;  } | Tạo label hiển thị thời gian khởi tạo note |  |
| 3 | public void Taolbtimelastsave(Note t)  {    #region Label time last save  lbtimeLastSave = new Label();  lbtimeLastSave.Text = "Time last save : " + t.lasttimesave.ToString("MM/dd/yyyy HH:mm");  lbtimeLastSave.Width = 350;  lbtimeLastSave.Font = new Font("Times New Roman", 10, FontStyle.Regular);    #endregion  } | Tạo label hiển thị thời gian chỉnh sửa cuối của note |  |
| 4 | public void Taotimelastsave(Note t)  {  timelastsave = DateTime.Now;  } | Tạo thời gian và gán vào thuộc tính timelast save của biến Note để lưu lại vào csdl |  |
| 5 | public void Taotimecreate(Note t)  {  timecreate = DateTime.Now;  } | Tạo thời gian và gán vào thuộc tính timecreate của biến Note để lưu lại vào csdl |  |
| 6 | public void Taotitle(Note t)  {  //Label Tiêu đề  lbTitle = new Label();  lbTitle.Text = "Title :";  lbTitle.Font = new Font("Times New Roman", 13, FontStyle.Regular);  lbTitle.Width = 50;  lbTitle.Height = 20;  // Textbox nhập tiêu đề  txtTitle = new TextBox();  txtTitle.Text = t.tieude;  txtTitle.Width = 330;  txtTitle.Font = new Font("Arial", 13, FontStyle.Bold);  txtTitle.TextChanged += (o, ep) =>  {  int dem = 0;  for (int i = 0; i < x.Count(); i++)  {  if (x[i].id == (int.Parse(lbstt.Text)))  {  dem = i;  break;  }  }  x[dem].tieude = txtTitle.Text;  x[dem].noidung = rtDescription.Text;  x[dem].lasttimesave = DateTime.Now;  lbtimeLastSave.Text = "Time last save : " + x[dem].lasttimesave.ToString("MM/dd/yyyy HH:mm");  db.SaveChanges();  };  } | Tạo label hiển thị cho title của layout ghi chú , tạo textbox để nhập title , cài đặt tính năng tự động lưu lại note mỗi khi có tương tác. |  |
| 7 | public void Taodescription(Note t)  {  // Label nội dung  lbDescription = new Label();  lbDescription.Text = "Description :";  lbDescription.Font = new Font("Times New Roman", 11, FontStyle.Regular);  lbDescription.Width = 400;  lbDescription.Height = 20;  // Richtextbox nhập nội dung  rtDescription = new RichTextBox();  rtDescription.Width = 395;  rtDescription.Height = 250;  rtDescription.Text = t.noidung;  rtDescription.TextChanged += (o, ep) =>  {  int dem = 0;  for (int i = 0; i < x.Count(); i++)  {  if (x[i].id == (int.Parse(lbstt.Text)))  {  dem = i;  break;  }  }  x[dem].tieude = txtTitle.Text;  x[dem].noidung = rtDescription.Text;  x[dem].lasttimesave = DateTime.Now;  lbtimeLastSave.Text = "Time last save : " + x[dem].lasttimesave.ToString("MM/dd/yyyy HH:mm");  db.SaveChanges();  };  } | Tạo label hiển thị cho description của layout ghi chú , tạo richtextbox để nhập title , cài đặt tính năng tự động lưu lại note mỗi khi có tương tác. |  |
| 8 | public void TaoTag(Note t)  {  // Tag  lbtag = new Label();  lbtag.Text = "Tag";  lbtag.Width = 30;  lbtag.Font = new Font("Times New Roman", 11, FontStyle.Regular);  txttag = new TextBox();  txttag.Width = 270;  btaddtag = new Button();  btaddtag.Text = "Add tag";  btaddtag.Width = 80;  if (t.tag != null)  {  rtDescription.Text += "\n#" + t.tag.ToString();  }  // btaddtag.Height = 20;  btaddtag.BackColor = Color.White;  btaddtag.Click += (sender, args) =>  {  if (txttag.TextLength != 0)  {  // add tag vào cuối  rtDescription.Text += "\n#" + txttag.Text.Trim();  MessageBox.Show("Addtag Success", "Notification", MessageBoxButtons.OK, MessageBoxIcon.Information);  this.txttag.Clear();  int dem = 0;  for (int i = 0; i < x.Count(); i++)  {  if (x[i].id == (int.Parse(lbstt.Text)))  {  dem = i;  break;  }  }  x[dem].tieude = txtTitle.Text;  x[dem].noidung = rtDescription.Text;  x[dem].lasttimesave = DateTime.Now;  lbtimeLastSave.Text = "Time last save : " + x[dem].lasttimesave.ToString("MM/dd/yyyy HH:mm");  db.SaveChanges();  }  else  {  MessageBox.Show("Addtag fail", "Notification", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  t.tag += " " + txttag.Text;  timelastsave = DateTime.Now;  db.SaveChanges();  };  } | Tạo label hiển thị , tạo textbox để nhập tag , tạo button add tag vào trong nội dung , tính năng lưu lại note khi có tương tác. |  |
| 9 | public void TaoDelete(Note t)  {  #region Button delete  //Button delete , thêm chức năng hỏi trc khi xóa  btdelete = new Button();  btdelete.Text = "Delete";  btdelete.Width = 100;  btdelete.Height = 30;  btdelete.BackColor = Color.White;  btdelete.Click += (sender, args) =>  {  /// chức năng xác nhận trc khi xóa  DialogResult dialogResult = MessageBox.Show("Sure", "Delete Note?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (dialogResult == DialogResult.Yes)  {  MessageBox.Show("Delete Success", "Notification", MessageBoxButtons.OK, MessageBoxIcon.Information);  // lấy vị trị trong list để xóa  int dem = 0;  for (int i = 0; i < x.Count(); i++)  {  if(x[i].id==(int.Parse(lbstt.Text)))  {  dem = i;  break;  }  }  db.Notes.Remove(x[dem]);  h.Dispose();  db.SaveChanges();  //h.Dispose();  }  else if (dialogResult == DialogResult.No)  {  MessageBox.Show("Delete Fail", "Notification", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  };  #endregion  } | Tạo button xóa note , và cập nhập lại list note khi xóa. |  |
| 10 | public FlowLayoutPanel fpnnote(Note t)  {  FlowLayoutPanel h = new FlowLayoutPanel();  h.Width = 420;  h.Height = 420;  h.BackColor = Color.Khaki;  Taotimelastsave(t);  Taotimecreate(t);  Taolbstt(t);  Taolbcreatetime(t);  Taotitle(t);  TaoDelete(t);  Taolbtimelastsave(t);  Taodescription(t);  TaoTag(t);  #region Tạo fpn  h.Controls.Add(lbTitle);  h.Controls.Add(txtTitle);  h.Controls.Add(lbDescription);  h.Controls.Add(rtDescription);  h.Controls.Add(lbtag);  h.Controls.Add(txttag);  h.Controls.Add(btaddtag);  h.Controls.Add(btdelete);  h.Controls.Add(lbTimeCreate);  h.Controls.Add(lbtimeLastSave);  h.Controls.Add(lbstt);  x = db.Notes.SqlQuery("Select \* from Note").ToList<Note>();  #endregion  return h;  } | Tạo form cho flow layout panel , thứ tự hiển thị trên flow layout của các label , textbox , richtextbox trên flow layout panel . |  |

***Bảng 5: Đặc tả các phương thức của lớp***

# **Chương 4: Cài đặt và kiểm thử**

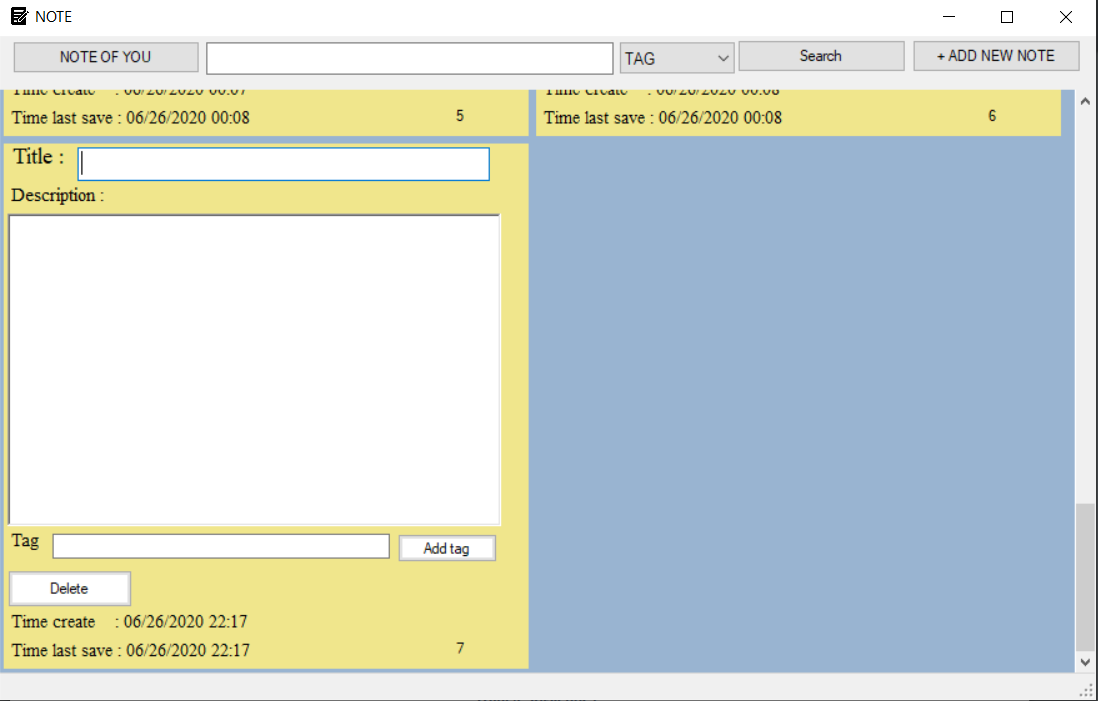
***Kiểm thử 1 : Load form***

***Hình 4: Kiểm thử 1***

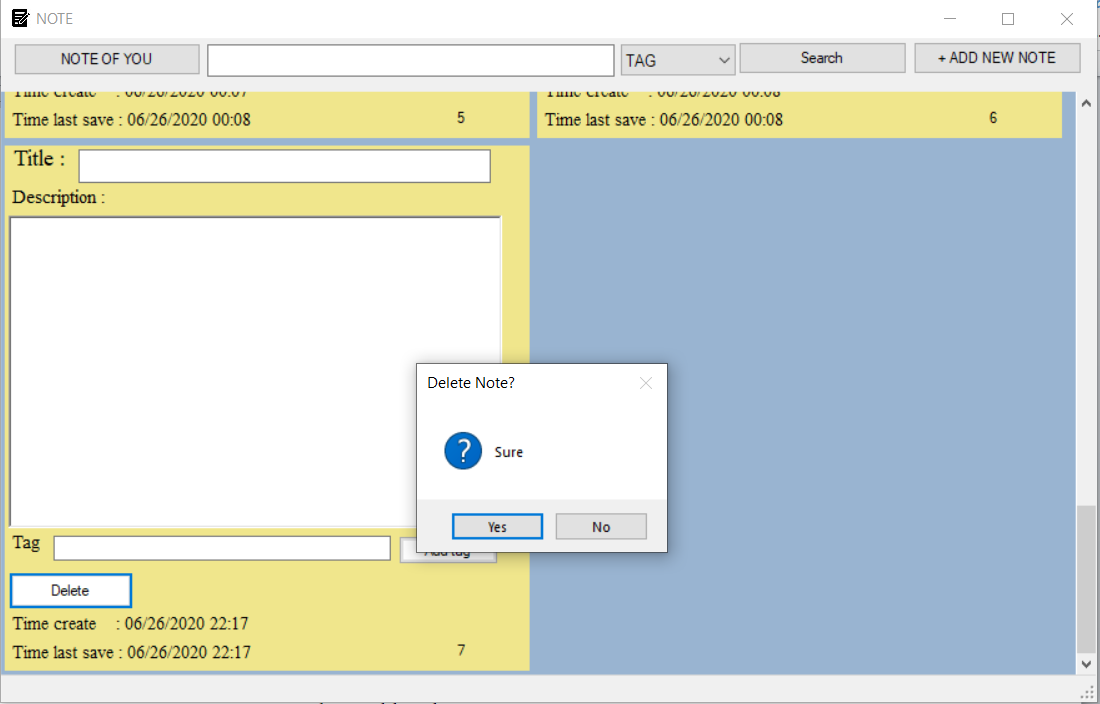
***Kiểm thử 2 : Search note có tag #2***

***Hình 5: Kiểm thử 2***

***Kiểm thử 3 : Add note mới.***



***Hình 6: Kiểm thử 3***

***Kiểm thử 4 : Xóa note***

***Hình 7: Kiểm thử 4***

# **Chương 5: Kết luận và hướng phát triển**

# ***1. Kết luận***

Về cơ bản, nhóm tự nhận xét phần mềm của nhóm đã giải quyết được được 90% yêu cầu mà đồ án đặt đặt ra. Sau đây là ưu điểm cũng như tồn tại của phần mềm .

* Ưu điểm:

1. Giao diện gọn gàng, dễ tiếp cận, dễ làm quen.
2. Dung lượng khá nhẹ.
3. Chương trình tốn rất ít tài nguyên hệ thống khi hoạt động.
   1. Chương trình chạy ổn định, cho ra kết quả chính xác, không bị crash trong quá trình thực thi yêu cầu người dùng.

* Nhược điểm:

1. Chương trình còn khá đơn giản, chưa có nhiều chức năng nâng cao
   * 1. Chưa có chức năng kiểm lỗi cho dữ liệu đầu vào.

# ***Hướng phát triển***

* + Thêm tính năng đọc dữ liệu đầu vào từ file.
  + Tối ưu hoá thuật toán đối với dữ liệu đầu vào từ lớn đến rất lớn.
  + Viết thêm tính năng kiểm lỗi dữ liệu đầu vào.
  + Cải thiện giao diện người dùng.

***/// sửa báo cáo , fix lỗi crash***

# **Tài liệu tham khảo**

[1]. Trang web: youtube.com, google.com

[2] Link: <https://stackoverflow.com/?fbclid=IwAR1xfmC2kZhKxMHeiKCDEKaSUzpGRdjd_vm6EcjGk1Vy25SofYg88yEx6Y0>

[3]. Loạt video bài học về Entity Framework <https://www.youtube.com/watch?v=9-hAhYMWj9I>