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



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN 1**

**Ứng dụng bảng băm, xây dựng từ điển Anh - Việt**

**GVHD : ThS. THẦY TRẦN CÔNG TÚ**

**NHÓM THỰC HIỆN:**

**NGUYỄN NGỌC HOÀNG PHÚC MSSV:15110099**

**PHAN DƯƠNG PHA MSSV:15110094**

**Tp. Hồ Chí Minh, tháng 12 năm 2017**

**MỤC LỤC**

[1.Mô tả project 1](#_Toc501320575)

[1.1. Phần mềm dùng làm gì? 1](#_Toc501320576)

[1.2. Use case diagram của phần mềm. 1](#_Toc501320577)

[1.3. Bảng mô tả ý nghĩa của từng use case. 2](#_Toc501320578)

[2. Mô tả quá trình làm. 2](#_Toc501320579)

[2.1. Thiết kế giao diện. 2](#_Toc501320580)

[2.1.1. Bảng mô tả thiết kế. 2](#_Toc501320581)

[2.1.2. Nội dung thiết kế. 3](#_Toc501320582)

[2.2. Thiết kế code. 5](#_Toc501320583)

[2.2.1. Mô tả thuật toán 5](#_Toc501320584)

[2.2.2. Các hàm. 7](#_Toc501320585)

[2.3. Thiết kế cơ sở dữ liệu. 10](#_Toc501320586)

[2.3.1. Bảng mô tả các bảng trong cơ sở dữ liệu. 10](#_Toc501320587)

[2.3.2. Bảng mô tả các trường trong bảng. 11](#_Toc501320588)

[2.4. Cài đặt và thử nghiệm. 11](#_Toc501320589)

[3. Mô tả phân công công việc. 11](#_Toc501320590)

[4. Kết luận. 12](#_Toc501320591)

**NỘI DUNG**

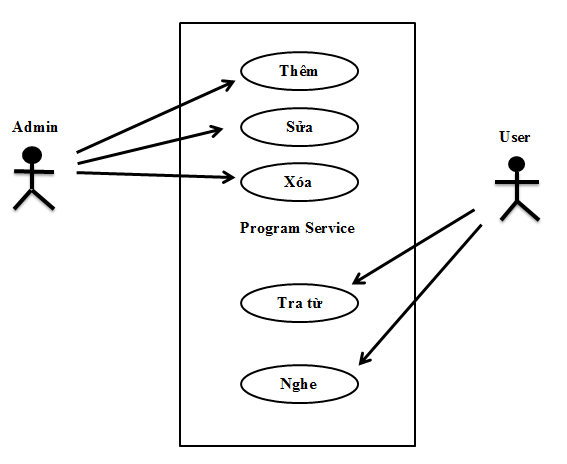
# 1.Mô tả project

## 1.1. Phần mềm dùng làm gì?

Ở Việt Nam, việc học Tiếng Anh vô cùng quan trọng, nhưng trong lúc học thì người học sẽ gặp một vấn đề mà ai cũng gặp phải đó là từ mới. Vì vậy, chương trình của nhóm thực hiện giúp người học giải quyết được vấn đề này. Thông qua chương trình người dùng có thể nhanh chóng tra nghĩa các từ vựng mới cũng như phiên âm hay cách đọc.

Để sử dụng chương trình thì các bạn nên tải thêm chương trình SQL Server 2014 (hoặc 2012) để sử dụng nhằm lưu trữ lượng lớn dữ liệu từ vựng.

## 1.2. Use case diagram của phần mềm.

****

## 1.3. Bảng mô tả ý nghĩa của từng use case.

|  |  |
| --- | --- |
| **Đối tượng** | **Chức năng** |
| Admin (Người quản trị/ Người viết). | * Thêm: Người viết có thể thêm từ (do người dùng gợi ý là từ đó chưa có trong từ điển) khi cần thiết. * Sửa: Trong quá trình sử dụng phần mềm thì người dùng sẽ có những góp ý, thông qua đó người quản trị có thể sửa đổi và cập nhật khi sai sót do dữ liệu từ khá nhiều. * Xóa: Người quản trị có thể xóa một từ khi cần. |
| User (Người dùng). | * Tra từ: Khi người dùng nhập vào từ cần tra thì phần mềm sẽ tra ra nghĩa của từ cùng với đó là phiên âm của từ giúp người dùng dễ phân biệt để phát âm. * Nghe: Người dùng có thể nghe thử nếu phiên âm của từ khá khó để đọc. |

# 2. Mô tả quá trình làm.

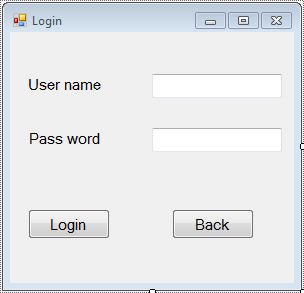
## 2.1. Thiết kế giao diện.

### 2.1.1. Bảng mô tả thiết kế.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Màn hình/ Cửa sổ/ Dialog** | **Người thiết kế và giải thích quyết định.** | **Mục đích chính của màn hình.** |
| 1 | Màn hình chính (form dành cho người dùng). | Phan Dương Pha thiết kế form main để người dùng có thể thao tác các chức năng cơ bản trên từ điển như: tra từ Tiếng Anh sang Tiếng Việt, tra từ Tiếng Việt sang Tiếng Anh, đọc từ Tiếng Anh. | Người dùng thao tác với các chức năng của từ điển. |
| 2 | Màn hình quản lý (form dành cho người quản trị/ người viết). | Nguyễn Ngọc Hoàng Phúc. Quyết định thiết kế màn hình này vì chỉ có người quản trị mới có quyền thao tác trên dữ liệu từ. | Mục đích của màn hình này là giúp cho người quản trị dễ dàng thực hiện các thao tác thêm, sửa, xóa với dữ liệu lớn từ vựng. |

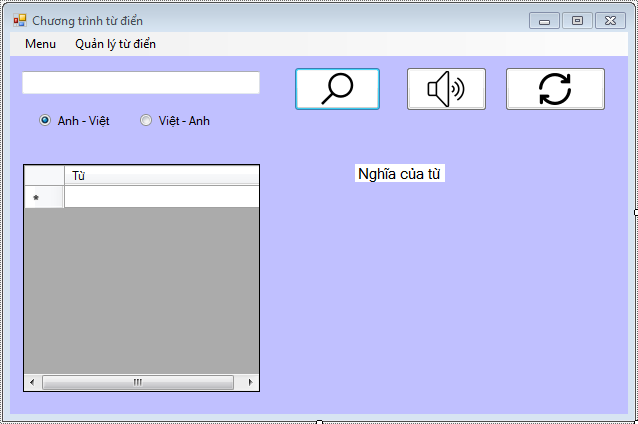
### 2.1.2. Nội dung thiết kế.

#### a)Màn hình đăng nhập



Thực hiện chức năng đăng nhập đơn giản cho người quản lý khi cần đăng nhập vào form quản lý để thực hiện thêm , sửa, xóa một từ.

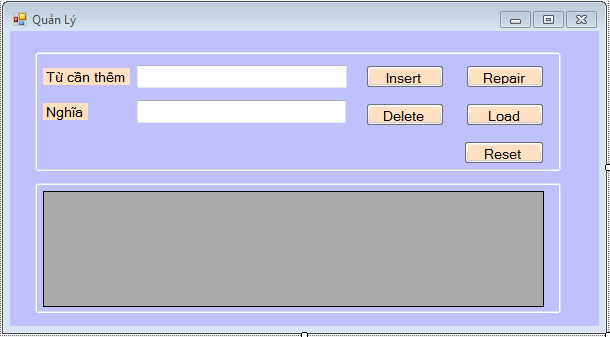
#### b) Màn hình chính.



Màn hình chính gồm:

* Một textbox để người dùng nhập từ cần tìm nghĩa
* 3 buton là tìm nghĩa của từ, phát âm từ, load lại data sau khi thực hiện chức năng bên form quản lý.
* Datagirview hiển thị các từ hiện có trong database
* Và lable hiển thị nghĩa của từ

#### c) Màn hình quản lý.

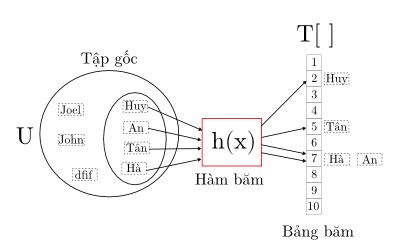


* Màn hình gồm:
* Lable Từ cần thêm, Lable Nghĩa với các Textbox để người quản trị nhập vào các từ cần thao tác.
* Các nút Button với các sự kiện Click để thao tác.
* Datagridview để hiển thị các từ đã có trong dữ liệu giúp người quản trị dễ dàng quan sát.

## 2.2. Thiết kế code.

### 2.2.1. Mô tả thuật toán

Bảng băm là một mảng T[0,1,…,m−1] có kích thước m. Để lưu trữ dữ liệu vào bảng băm, ta sẽ dùng một **hàm băm** (hash function). Một hàm băm: là một ánh xạ gán cho mỗi phần tử của tập U một vị trí trong bảng T. Cụ thể, phần tử x sẽ được lưu tại ô T[h(x)] của bảng và ta nói x được băm vào vị trí h(x) và h(x) được gọi là **mã băm** (hash code) của x. Xem hình minh họa dưới đây.

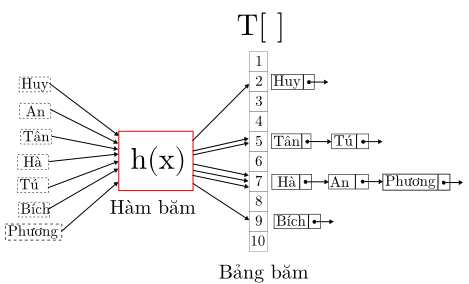


Hai thao tác chính của bảng băm đó là: đưa một phần tử mới vào bảng băm và tìm xem một phần tử có nằm ở trong bảng băm hay không. Mã giả:

PutToTable(key x, h):   
    T[h(x)]←x

Lookup(key x, h):   
    **if** T[h(x)]=x  
        return Yes  
    return No

Do U lớn hơn m, theo nguyên lí [nhốt chim](https://en.wikipedia.org/wiki/Pigeonhole_principle), một hoặc nhiều phần tử của tập U có thể sẽ được băm vào cùng một vị trí của bảng. Ta gọi hiện tượng đó là xung đột(collision). Để giải quyết xung dột ở đây nhóm thực hiện sử dụng một danh sách liên kết để lưu các giá trị có cũng mã băm(hashcode) vào một danh sách liên kết. Xem hình ảnh minh họa dưới đây.



**Nhận xét:**

Tốc độ truy xuất phụ thuộc vào việc lựa chọn hàm băm sao cho băm đều n phần tử vào T danh sách liên kết(T bucket).

Nếu lựa chọn m(kích thước của bảng băm) càng lớn thì tốc độ thực hiện các phép toán trên bảng băm càng nhanh, tuy nhiên sẽ tốn dung lượng bộ nhớ.

Nếu chọn m=n(kích thước bảng băm=số phần tử ) thì thời gian truy xuất tương đương với truy xuất trên mảng(có bậc O(1)), xong tốn bộ nhớ.

Nếu chon m=n/k(k=2,3,4,...) thì ít tốn bộ nhớ hơn k lần, nhưng tốc độ chậm đi k lần.

### 2.2.2. Các hàm.

#### a) Form Main.

* **Khai báo cấu trúc khi lưu trữ một từ, gồm có 4 thông tin.**

public struct Entry

{

public int hashCode; // giá trị được tính dựa vào mã ASCII của từ,

//được sử dụng khi xác định vị trí của phần tử trong bảng băm

public int next; // nếu trong cùng một địa chỉ có 2 phần tử được lưu thì biến này sẽ lưu thứ tự của phần tử được thêm vào trước đó

//và = -1 if nếu là phần tử cuối cùng tức là được thêm vào đầu tiên

public string key; // từ

public string value; // nghĩa tương ứng

}

public int[] buckets; // khai báo buckets chứa giá trị đã băm của từ và thứ tự được thêm vào

public Entry[] entries; // luu trữ các thông tin hashcode, next,key, value của từ

* **Hàm khởi tạo buckets và entry dựa vào số phần tử hiện có trong database.**

public void Initialize(int size)

{

//khoi tao mang bucket

buckets = new int[size];

for (int i = 0; i < buckets.Length; i++) buckets[i] = -1; //lưu trữ vị trí của phần tử trong bảng băm

entries = new Entry[size]; //lưu trữ các giá trị liên quan đến phần tử : key, value, next, hashcode

}

* **Hàm thêm một từ vào bảng băm.**

public void Insert(string keys, string values, int count)

{

//khởi tạo bucket

if (buckets == null) Initialize(0);

//lấy giá trị băm của từ

int hashCode = GetHashCode(keys)& 0x7FFFFFFF;

//hàm băm từ để xác định vị trí của từ trong bảng băm

int targetBucket =hashCode % buckets.Length;

int index;

index = count;

count++;

* **Thêm các thông tin liên quan đến phần tử.**

entries[index].hashCode = hashCode;

//xác định điểm dừng và lưu thông tin về vị trí của entries khi mà 2 entries có cùng địa chỉ mảng trong bucket

entries[index].next = buckets[targetBucket];

entries[index].key = keys;

entries[index].value = values;

//xác định phần tử nằm ở địa thứ mấy và thứ tự được thêm vào buckets,

//nếu 2 phần tử có cùng địa chỉ thì địa chỉ đó sẽ lưu thứ tự của phần tử được thêm vào sau cùng

buckets[targetBucket] = index;

}

* **Hàm tìm nghĩa của phần tử.**

public int FindEntry(string keys)

{

if (keys == null)

{

MessageBox.Show("Bạn chưa nhập từ cần tìm");

}

if (buckets != null)

{

int hashCode = GetHashCode(keys);

// xác định thứ tự của phần tử, qua đó tìm được thông tin của phần tử trong mảng các entries

for (int i = buckets[hashCode % buckets.Length]; i >= 0; i = entries[i].next)

{

//kiểm tra xem hashcode của từ đó với key có giống nhau không

if (entries[i].hashCode == hashCode && Comparer.Equals(entries[i].key, keys)) return i;

}

}

return -1;

}

* Hàm LoadTD, load dữ liệu lên datagirview khi chương trình chạy.
* Sự kiện fmain\_Load khi form được load lên sẽ lấy thông tin từ database đưa lên datagirview.
* Sự kiện show form Quanly khi click vào toolStripMenu Quản lý từ điển.
* Lấy dữ liệu từ dgv đưa lên txtbox.
* Sự kiện button tra từ dược click.
* Xử lý sự kiện khi người dùng chọn chức năng tra từ anh-viet.
* Xử lý sự kiện khi người dùng chọn chức năng tra từ anh-viet.

#### b) Form Quản lý.

* Hàm loaddata() dùng để load dữ liệu từ ở cơ sở dữ liệu lên datagridview giúp dễ quan sát. Hàm này thông qua chuỗi kết nối sql để thực thi một store, store này trả về 1 bảng Dataset nhằm load dữ liệu từ cơ sở dữ liệu lên datagridview.
* Hàm setValue(), hàm này dùng để đưa từ đã chọn từ datagridview lên textbox để thực hiện các thao tác.
* Hàm btlamlai\_Click(), khi thực hiện các thao tác thêm, sửa, xóa thì dữ liệu sẽ bị thay đổi nên ta cần load lại dữ liệu để chắc rằng các thao tác của ta là đúng đắn.
* Hàm Grvdata\_RowEnter(), lấy các giá trị từ datagridview đưa lên textbox thông qua hàm setValue() với sự kiện khi ta click chuột vào bất kì hàng nào trên datagridview.
* Hàm btnNhapLai\_Click(), xóa tất cả dữ liệu trên textbox để nhập lại từ và nghĩa mới khi thao tác.
* Hàm btthem\_Click(), thông qua các biến được truyền vào từ textbox, hàm sẽ gọi store để thực thi câu lệnh SQL và thêm từ, nghĩa từ vào cơ sở dữ liệu. Nếu thêm được thì sẽ hiện thị dòng thông báo “Thêm thành công” còn nếu không thêm được sẽ hiện thị dòng thông báo “Không thêm được, lỗi rồi!”.
* Hàm btxoa\_Click() thông qua biến từ textbox sẽ gọi store để thực thi câu lệnh SQL với từ sẽ làm khóa. Nếu không xóa được sẽ hiển thị thông báo “Không xóa được, lỗi rồi!”.
* Hàm btsua\_Click(), thông qua các biến được truyền vào từ textbox, hàm sẽ gọi store để thực thi câu lệnh SQL và sửa từ, nghĩa từ vào cơ sở dữ liệu. Nếu không sửa được sẽ hiện thị dòng thông báo “Không sửa được, lỗi rồi!”.

## 2.3. Thiết kế cơ sở dữ liệu.

### 2.3.1. Bảng mô tả các bảng trong cơ sở dữ liệu.

|  |  |  |
| --- | --- | --- |
| STT | Tên bảng | Mục đích của bảng trong chương trình |
| 1 | Từ | Bảng chứa dữ liệu các từ và nghĩa của từ. |

### 2.3.2. Bảng mô tả các trường trong bảng.

|  |  |  |
| --- | --- | --- |
| STT | Tên trường | Mục đích của trường trong chương trình |
| 1 | Từ | Chứa dữ liệu các từ. |
| 2 | Nghĩa | Chứa dữ liệu các nghĩa. |

## 2.4. Cài đặt và thử nghiệm.

Cài đặt trên máy người dùng(xem trong file đính kèm).

Tốc độ truy xuất có bậc O(1) nếu không bị dụng độ.

# 3. Mô tả phân công công việc.

|  |  |  |
| --- | --- | --- |
| Tên sinh viên. | Đánh giá chung phần trăm đóng góp. | Mô tả khái quát mảng công việc sinh viên thực hiện trong đồ án. |
| Phan Dương Pha |  | * Thiết kế giao diện. * Viết các hàm khởi tạo ban đầu. * Viết giải thuật bảng băm. * Từ giải thuật viết hàm tra từ. * Chức năng đọc từ. * Tra nghĩa từ Tiếng Anh sang Tiếng Việt. * Viết báo cáo: 50% |
| Nguyễn Ngọc Hoàng Phúc |  | * Tìm file dữ liệu từ. * Viết hàm load dữ liệu từ database lên từ điển. * Các hàm thao tác cơ bản thêm, sửa, xóa. * Tra nghĩa từ Tiếng Việt sang Tiếng Anh. * Viết báo cáo: 50% |

# 4. Kết luận.

* Nhìn chung sau khi trải qua trên dưới 10 tuần để tìm hiểu về đồ án với sự giám sát và giúp đỡ tận tình của Thầy Trần Công Tú thì nhóm thực hiện đã cơ bản hoàn thành được một chương trình từ điển có thể giúp người học dễ dàng tra cứu những từ mới.
* Khó khăn:
* Trước tiên là do lịch học của các bạn khác nhau nên thời gian gặp nhau để họp nhóm của nhóm thực hiện tương đối hạn chế và kế hoạch thực hiện cũng bị trì trệ.
* Thứ hai, trong quá trình làm do không lường trước được kết quả, nhóm thực hiện đã lưu từ và nghĩa vào 2 file riêng biệt nên việc load dữ liệu và tra từ trở nên không khả thi. Do đó nhóm thực hiện đã giải quyết vấn đề bằng cách là lưu dữ liệu trong cùng 1 file đưa vào cơ sở dữ liệu, từ đó việc load dữ liệu và tra từ khả thi hơn trước.
* Thứ ba, do em là Nguyễn Ngọc Hoàng Phúc vẫn chưa học môn Cấu Trúc Dữ Liệu Và Giải Thuật nên hơi khó khăn trong việc tìm hiểu. Nhưng qua sự giúp đỡ của bạn Phan Dương Pha và sự tận tình của Thầy Trần Công Tú nên hiện tại em đã có thể hiểu cơ bản về giải thuật bảng băm (Hash Function) là như thế nào.
* Về phần ý tưởng phát triển đồ án, nếu có nhiều thời gian hơn nữa để nghiên cứu thì nhóm thực hiện nghĩ sẽ thêm chức năng như người dùng chỉ cần nói ra từ mình muốn tìm kiếm thì từ điển sẽ đưa ra các từ gợi ý mà không cần phải nhập tay, hoặc thêm một Hyperlink để tìm hiểu về một từ mới với nhiều hình ảnh và ví dụ sinh động hơn cho người dùng.
* Ưu điểm: Phần mêm của chúng em có giao diện khá gần gũi với người dùng. Với lưu lượng từ khoảng 2500-3000 thì khá phù hợp cho những người mới bắt đầu học Tiếng Anh. Tìm kiếm từ vựng nhanh, đơn giản với các chức năng tra từ, nghe từ.
* Khuyết điểm:
* Lưu lượng từ còn ít khó đáp ứng cho nhu cầu của người dạy ở cấp cao. Chỉ có những chức năng cơ bản như tra từ, nghe từ. Chưa có ví dụ sinh động về từ để người dùng dễ dàng hiểu về từ đó và sử dụng một cách hợp lý.
* Do đồ án chủ yếu nghiên cứu về giải thuật Hash Function ở môn trước nên chương trình từ điển hiện tại vẫn chưa phải là một chương trình hoàn chỉnh.