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



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

🞠◊🞠◊🞠



**ĐỀ TÀI: PHẦN MỀM TỪ ĐIỂN**

**Báo cáo đồ án**

**(Môn học: Đồ án 1)**

**GVHD: THẦY TRẦN CÔNG TÚ**

**NHÓM THỰC HIỆN:**

**Nguyễn Ngọc Hoàng Phúc Mssv:15110099**

**Phan Dương Pha Mssv:15110094**

**HỌC KỲ: 1 – NĂM HỌC: 2017-2018**

**TP.HỒ CHÍ MINH-12/2017**

**TRANG LÓT**

**MỤC LỤC**

Contents

[1.Mô tả project 1](#_Toc499282163)

[1.1. Phần mềm dùng làm gì? 1](#_Toc499282164)

[1.2. Use case diagram của phần mềm. 1](#_Toc499282165)

[1.3. Bảng mô tả ý nghĩa của từng use case. 2](#_Toc499282166)

[2. Mô tả quá trình làm. 2](#_Toc499282167)

[2.1. Thiết kế giao diện. 2](#_Toc499282168)

[2.1.1. Bảng mô tả thiết kế. 2](#_Toc499282169)

[2.1.2. Nội dung thiết kế. 3](#_Toc499282170)

[2.2. Thiết kế code. 5](#_Toc499282171)

[2.2.1. Các hàm. 5](#_Toc499282172)

[2.3. Thiết kế cơ sở dữ liệu. 17](#_Toc499282173)

[2.3.1. Bảng mô tả các bảng trong cơ sở dữ liệu. 17](#_Toc499282174)

[2.3.2. Bảng mô tả các trường trong bảng. 17](#_Toc499282175)

[2.4. Cài đặt và thử nghiệm. 18](#_Toc499282176)

[3. Mô tả phân công công việc. 18](#_Toc499282177)

[4. Kết luận. 18](#_Toc499282178)

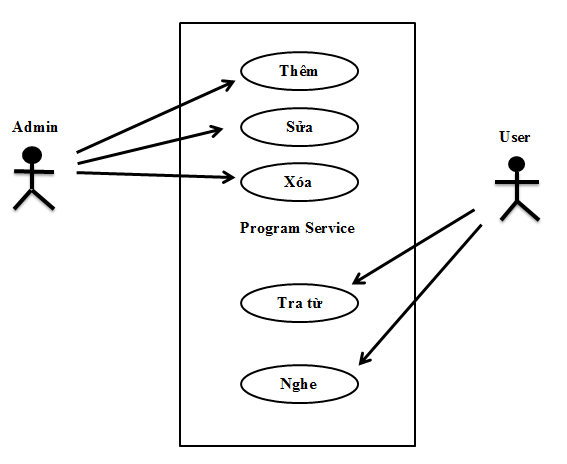
**NỘI DUNG**

# 1.Mô tả project

## 1.1. Phần mềm dùng làm gì?

Trong thực tế, ở các nước nói chung và ở Việt Nam ta nói riêng. Việc học tiếng anh vô cùng quan trọng, nhưng trong lúc học thì người học tất nhiên sẽ gặp một vấn đề mà ai cũng gặp phải đó là từ mới. Vì vậy, phần mềm của chúng em giúp người học giải quyết được vấn đề nan giải này. Thông qua phần mềm người dùng có thể nhanh chóng tra nghĩa các từ vựng mới cũng như phiên âm hay cách đọc.

## 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.

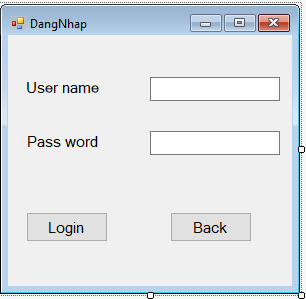
## 2.1. Thiết kế giao diện.

### 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. |

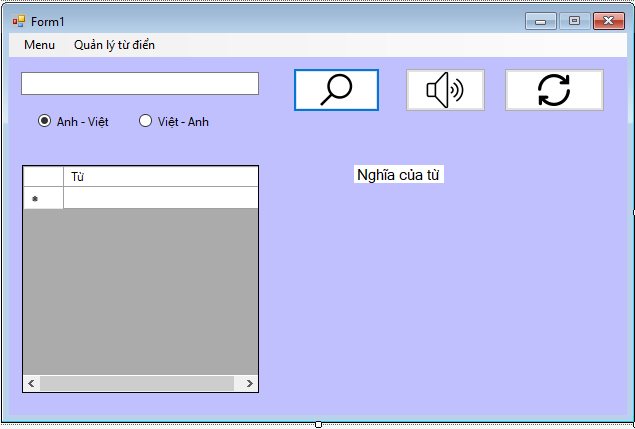
### 2.1.2. Nội dung thiết kế.

#### 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.

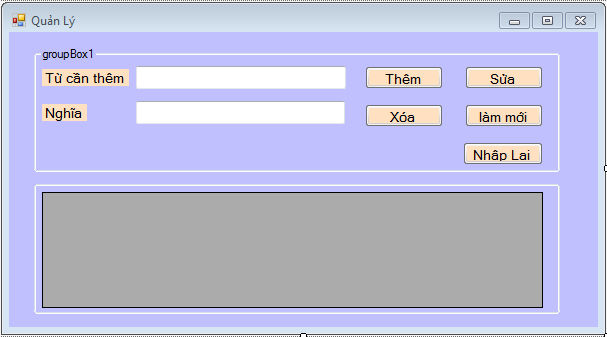
#### b) Màn hình chính.



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. Các hàm.

#### a) Form Main.

* **Hàm LoadTD, load dữ liệu lên datagirview khi chương trình chạy**

public void LoadTD() {

try

{

//khởi dộng connection kết nối đến DB

conn = new SqlConnection(strConnectionString);

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

//lấy dữ liệu từ bảng danh sách từ

//tạo đối tượng SqlAdapter là cầu nối giữa dataset và datasource để thực hiện công việc như đọc hay cập nhật dữ liệu

daTD = new SqlDataAdapter("SELECT \* FROM Tu ", conn);

//// khởi tạo đối tượng datatable

dtTD = new DataTable();

dtTD.Clear();

//// fill dữ liệu vào datatable

daTD.Fill(dtTD);

// Đưa dữ liệu lên dgv

//nếu dữ liệu được hiển thị ra datagridview .Ta cần 1 DataView kết nối đến DataTable .Đối tượng DataView dùng cho việc sắp xếp,lọc, tìm kiếm

dtvTD = new DataView(dtTD);

// gán datasource cho datagridview

dgvTD.DataSource = dtvTD;

if (rdanhviet.Checked == true)

{

dgvTD.Columns[0].Visible = true;

dgvTD.Columns[1].Visible = false;

}

else

{

dgvTD.Columns[0].Visible = false;

dgvTD.Columns[1].Visible = true;

}

}

catch (SystemException er)

{

MessageBox.Show(er.Message);

}

}

* **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 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;

}

* **Sự kiện fmain\_Load khi form được load lên sẽ lấy thông tin từ database đưa lên datagirview**

private void fmain\_Load(object sender, EventArgs e)

{

int c = 0;

LoadTD();

//lấy số lựơng từ để khởi tạo kích thước bảng băm

int size = Convert.ToInt32(dgvTD.RowCount.ToString());

Initialize(size);

string k, v;

//lấy từng phần tử trong dataTable để thêm vào bảng băm

foreach (DataRow dr in dtTD.Rows)

{

k = dr["Tu"].ToString();

v = dr["Nghia"].ToString();

//kiểm tra xem người dùng dang dùng chức năng tra từ nào

//chức năng tra từ tiếng anh sang việt

if (rdanhviet.Checked == true)

{

Insert(k, v, c);

c++;

}

//chức năng tra từ từ viêt sang anh

else

{

Insert(v,k, c);

c++;

}

}

}

* **Sự kiện show form Quanly khi click vào toolStripMenu Quản lý từ diển9**

private void toolStripMenuItem6\_Click(object sender, EventArgs e)

{

new DangNhap().ShowDialog();

}

//hiển thị messagebox chọn yes/no khiclick vào toolStripMenu thoát chương trình

private void toolStripMenuExit\_Click(object sender, EventArgs e)

{

string message = "Bạn có uốn thoát không ?";

string caption = "Từ điển Anh-Việt";

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

DialogResult result;

// Displays the MessageBox.

result = MessageBox.Show(this, message, caption, buttons);

if (result == DialogResult.Yes)

{

// Closes the parent form.

this.Close();

}

}

* **Lấy dữ liệu từ dgv đưa lên txtbox**

private void dgvTD\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

//thao tác cho radiobutton anh-viet

if (rdanhviet.Checked == true)

{

// Thứ tự dòng hiện hành

int r = dgvTD.CurrentCell.RowIndex;

//Chuyển thông tin lên textbox

//lấy giá trị ở vị trí hiện tại trong cell[0] dưa lên tntratu

this.tbtratu.Text =

dgvTD.Rows[r].Cells[0].Value.ToString();

//lấy giá trị ở vị trí hiện trong cell[1] tại dưa lên lable nghĩa

this.label1.Text =

dgvTD.Rows[r].Cells[1].Value.ToString();

}

//thao tác cho radiobutton viet-anh

else if (rdvietanh.Checked == true)

{

// Thứ tự dòng hiện hành

int r = dgvTD.CurrentCell.RowIndex;

//Chuyển thông tin lên textbox

//lấy giá trị ở vị trí hiện tại trong cell[1] dưa lên tntratu

this.tbtratu.Text =

dgvTD.Rows[r].Cells[1].Value.ToString();

//lấy giá trị ở vị trí hiện trong cell[0] tại dưa lên tntratu

this.label1.Text =

dgvTD.Rows[r].Cells[0].Value.ToString();

}

}

catch (SystemException er)

{

MessageBox.Show(er.Message);

}

}

* **Sự kiện button tra từ dược click**

private void btntratu\_Click(object sender, EventArgs e)

{

if (tbtratu.Text == "")

{

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

}

else

{

//lấy được thứ tự của từ được thêm vào bảng băm

int i = FindEntry(tbtratu.Text);

//nếu i trả về lớn ơn 0 thì d49 tìm thấy

if (i >= 0)

{

label1.Text = entries[i].value;

}

//không tìm thấy

else

{

MessageBox.Show("Từ bạn tìm hiện không có ");

}

}

}

* **Xử lý sự kiện khi người dùng chọn chức năng tra từ anh-viet**

private void rdanhviet\_Click(object sender, EventArgs e)

{

int c = 0;

//gọi hàm load để lấy dữ liệu

LoadTD();

//lấy số lựơng từ để khởi tạo kích thước bảng băm

int size = Convert.ToInt32(dgvTD.RowCount.ToString());

Initialize(size);

string k, v;

//lấy từng phần tử trong dataTable để thêm vào bảng băm

foreach (DataRow dr in dtTD.Rows)

{

k = dr["Tu"].ToString();

v = dr["Nghia"].ToString();

//thêm phần tử vào bảng băm

Insert(k, v, c);

c++;

}

}

* **Xử lý sự kiện khi người dùng chọn chức năng tra từ anh-viet**

private void rdvietanh\_Click(object sender, EventArgs e)

{

int c = 0;

LoadTD();

//lấy số lựơng từ để khởi tạo kích thước bảng băm

int size = Convert.ToInt32(dgvTD.RowCount.ToString());

Initialize(size);

string k, v;

//lấy từng phần tử trong dataTable để thêm vào bảng băm

foreach (DataRow dr in dtTD.Rows)

{

k = dr["Tu"].ToString();

v = dr["Nghia"].ToString();

//thêm phần tử vào bảng băm

Insert(v, k, c);

c++;

}

}

#### 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.**

string strConnectionString = "Data Source=(local);Initial Catalog=DA1;Integrated Security=True";

private void loaddata()

{

Grvdata.DataSource = SqlHelper.ExecuteDataset(strConnectionString, "Tu\_Select").Tables[0];

}

Đây là store lấy dữ liệu bên SQL.

ALTER procedure [dbo].[Tu\_Select]

as

SELECT [Tu]

,[Nghia]

FROM [dbo].[Tu]

* **Hàm setValue(), hàm này dùng để đưa từ đã chọn từ datagridview lên textbox để thực hiện các thao tác.**

private void setValue(int index)

{

try

{

//khởi tạo biến row bằng hàng được chọn trong datagridview với biến index

DataGridViewRow row = Grvdata.Rows[index];

//lấy giá trị ô đầu tiên trong hàng đưa vào textbox từ

tbthemtu.Text = row.Cells[0].Value.ToString();

//lấy giá trị ô thứ hai trong hàng đưa vào textbox nghĩa

tbnghia.Text = row.Cells[1].Value.ToString();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

* **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.**

private void btlamlai\_Click(object sender, EventArgs e)

{

loaddata();

}

* **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.**

private void Grvdata\_RowEnter(object sender, DataGridViewCellEventArgs e)

{

setValue(e.RowIndex);

}

* **Hàm btnNhapLai\_Click(), xóa tất cả dữ liệu trên textbox để nhập lại từ và nghĩa mới khi thao tác.**

private void btnNhaplai\_Click(object sender, EventArgs e)

{

tbthemtu.Clear();

tbnghia.Clear();

}

* **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!”.**

private void btthem\_Click(object sender, EventArgs e)

{

try

{

//khai báo các biến truyền vào

string Tu = tbthemtu.Text.Trim();

string Nghia = tbnghia.Text.Trim();

//truyền dữ liệu lên sql

SqlHelper.ExecuteNonQuery(strConnectionString, "Tu\_Them", Tu, Nghia);

MessageBox.Show("Thêm thành công!");

}

catch (Exception ex)

{

MessageBox.Show("Không thêm được, lỗi rồi!");

}

}

Store thêm bên SQL.

ALTER procedure [dbo].[Tu\_Them]

(@Tu nvarchar(250)

,@Nghia nvarchar(max))

as

INSERT INTO [dbo].[Tu]

([Tu]

,[Nghia])

VALUES

(@Tu

,@Nghia)

* **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!”.**

private void btxoa\_Click(object sender, EventArgs e)

{

try

{

string Tu = tbthemtu.Text.Trim();

SqlHelper.ExecuteNonQuery(strConnectionString, "Tu\_Xoa", Tu);

loaddata();

}

catch (Exception ex)

{

MessageBox.Show("Không xóa được, lỗi rồi!");

}

}

Store Xóa bên SQL.

ALTER procedure [dbo].[Tu\_Xoa]

@Tu nvarchar(250)

as

DELETE FROM [dbo].[Tu]

WHERE [Tu] = @Tu

* **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!”.**

private void btsua\_Click(object sender, EventArgs e)

{

try

{

string Tu = tbthemtu.Text.Trim();

string Nghia = tbnghia.Text.Trim();

SqlHelper.ExecuteNonQuery(strConnectionString, "Tu\_Sua", Tu, Nghia);

loaddata();

}

catch (Exception ex)

{

MessageBox.Show("Không sửa được, lỗi rồi!");

}

}

Store Sửa bên SQL.

ALTER procedure [dbo].[Tu\_Sua]

@Tu nvarchar(250)

,@Nghia nvarchar(max)

as

UPDATE [dbo].[Tu]

SET [Tu] = @Tu

,[Nghia] = @Nghia

WHERE [Tu] = @Tu

## 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.

# 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 | 55% | 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. |
| Nguyễn Ngọc Hoàng Phúc | 45% | 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 sanh tiếng anh.  Viết báo cáo. |

# 4. Kết luận.

* Nhìn chung sau khi trải qua trên dưới 10 tuần để tìm hiểu về phần mềm với sự giám sát và giúp đỡ tận tình của Thầy Trần Công Tú thì chúng em đã cơ bản hoàn thành được một phần mềm từ điển điện tử có thể giúp người học dễ dàng tra cứu những từ mới. Tuy nhiên, bên cạnh đó thì chúng em gặp khá nhiều khó khăn:
* Trước tiên là do lịch học của các bạn khác nhau nên thời gian gặp nhau để họp nhóm của chúng em tương đối hạn chế và đương nhiên là kế hoạch thực hiện cũng bị trì trệ.
* Thứ hai, trong quá trình làm do không lường trước được kết quả, chúng em 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 chúng em 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 chúng em 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 bạn mới bắt đầu học tiếng anh hoặc những bạn chưa biết gì về 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.