**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**⬩⬩🙠🕮🙢⬩⬩**



**BÁO CÁO ĐỒ ÁN**

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**

**ĐỀ TÀI: ỨNG DỤNG BẢNG BĂM ĐỂ**

**XÂY DỰNG TỪ ĐIỂN**

**SVTH: TRẦN TẤN QUÝ**

**MSSV: 16110196**

**SVTH: VÕ GIA HUY**

**MSSV: 16110092**

**TP. Hồ Chí Minh, tháng 12 năm 2018**

**MỤC LỤC**

[**1. Giới thiệu** 1](#_Toc531080812)

[***1.1. Bảng băm là gì?*** 1](#_Toc531080813)

[***1.2. Ứng dụng bảng băm*** 1](#_Toc531080814)

[***1.3. Giới thiệu phần mềm*** 1](#_Toc531080815)

[***1.4. Mục tiêu*** 1](#_Toc531080816)

[***1.5. Chức năng chính*** 2](#_Toc531080817)

[**2. Quá trình thực hiện** 3](#_Toc531080818)

[***2.1. Thiết kế các class*** 3](#_Toc531080819)

[***2.2. Thiết kế giao diện*** 4](#_Toc531080820)

[**a. Giao diện chính** 4](#_Toc531080821)

[**b. User control Search** 5](#_Toc531080823)

[**c. User control AddWord** 6](#_Toc531080824)

[**d. User control EditWord** 7](#_Toc531080826)

[**e. User control DeleteWord** 8](#_Toc531080828)

[***2.3. Thiết kế bảng băm*** 9](#_Toc531080830)

[**a. Hàm băm** 9](#_Toc531080831)

[**b. Thêm từ vào bảng băm** 9](#_Toc531080832)

[**c. Xóa từ trong bảng băm** 10](#_Toc531080833)

[**d. Sửa từ trong bảng băm** 10](#_Toc531080834)

[***2.4. Thiết kế các code chính*** 12](#_Toc531080835)

[**a. Hàm tra từ** 12](#_Toc531080836)

[**b. Hàm thêm từ** 13](#_Toc531080837)

[**c. Hàm sửa từ** 14](#_Toc531080838)

[**d. Hàm xóa từ** 16](#_Toc531080839)

[**e. Hàm đọc file** 19](#_Toc531080840)

[**3. Cài đặt và kiểm thử** 21](#_Toc531080841)

[***3.1. Cài đặt*** 21](#_Toc531080842)

[***3.2. Kiểm thử*** 21](#_Toc531080843)

[**4. Phân công công việc** 22](#_Toc531080844)

[**5. Kết luận** 22](#_Toc531080845)

[***5.1. Kết quả*** 22](#_Toc531080846)

[***5.2. Khó khăn*** 22](#_Toc531080847)

[***5.3. Ưu điểm*** 23](#_Toc531080848)

[***5.4. Nhược điểm*** 23](#_Toc531080849)

[***5.5. Hướng phát triển*** 23](#_Toc531080850)

**DANH MỤC CÁC HÌNH**

[Hình 1. Giao diện chính của ứng dụng 4](#_Toc530473552)

[Hình 2. Giao diện tra từ 5](#_Toc530473553)

[Hình 3. Giao diện lịch sử tra từ 5](file:///D:\CTDL\FinalProject\Dictionary_HashTable\Dictionary_HashTable\CTDLReport.docx#_Toc530473554)

[Hình 4. Giao diện thêm từ 6](#_Toc530473555)

[Hình 5. Giao diện sửa từ 7](#_Toc530473556)

[Hình 6. Giao diện xóa từ 8](#_Toc530473557)

# **1. Giới thiệu**

## ***1.1. Bảng băm là gì?***

Bảng băm là một cấu trúc dữ liệu cho phép chúng ta có thể nhanh chóng xác định phần tử nào đó có tồn tại trong bảng băm hay không. Các phép toán trên bảng băm sẽ giúp hạn chế số lần so sánh, và vì vậy sẽ cố gắng giảm thiểu được thời gian truy xuất. Độ phức tạp của các phép toán trên bảng băm thường có bậc là O(1) và không phụ thuộc vào kích thước của bảng băm.

## ***1.2. Ứng dụng bảng băm***

Bảng băm được ứng dụng để xây dựng từ điển do tính hữu dụng và có tốc độ tìm kiếm cao của các hàm băm, phù hợp khi tổ chức dữ liệu có kích thước lớn. Trong đề tài được giao, nhóm sẽ ứng dụng bảng băm được xây dựng trên hàm băm theo phương pháp kết nối trực tiếp.

## ***1.3. Giới thiệu phần mềm***

Trong thời đại đang ngày nay việc học hỏi, hiểu biết thêm các ngôn ngữ khác bên cạnh tiếng mẹ đẻ đang ngày càng trở nên phổ biến. Trong đó tiếng Anh không phải là một ngoại lệ, thậm chí tiếng Anh nắm vị trí số 1 và là ngôn ngữ phổ biến nhất. Việc bắt đầu học một ngôn ngữ mới lạ luôn gây ra ít nhiều khó khăn trong việc hiểu nghĩa của từ, do vậy phần mềm này được xây dựng nhằm giải quyết khó khăn đó. Chức năng chính của phần mềm là tra cứu từ, dịch từ từ tiếng Anh sang tiếng Việt.

## ***1.4. Mục tiêu***

Mục tiêu của đề tài là xây dựng được ứng dụng từ điển dựa trên bảng băm theo phương pháp kết nối trực tiếp. Ứng dụng phải đáp ứng yêu cầu chính của bảng băm là khả năng tìm kiếm với tốc độ cao kèm theo đó giao diện của ứng dụng phải đẹp mắt, tương thích với người dùng.

## ***1.5. Chức năng chính***

|  |  |
| --- | --- |
| **Tên bảng** | **Chức năng** |
| Giao diện chính | - Menu chứa các chức năng. - Các chức năng chính của chương trình. |
| Bảng tra từ | Tra từ có trong từ điển và nghe từ. |
| Bảng thêm từ | Thêm một từ mới vào từ điển. |
| Bảng sửa từ | Sửa từ và nghĩa của một từ có sẵn trong từ điển. |
| Bảng xóa từ | Xóa một từ có sẵn trong từ điển. |
| Bảng thông tin | Thông tin của các sinh viên thực hiện đề tài. |

# **2. Quá trình thực hiện**

## ***2.1. Thiết kế các class***

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên class** | **Chức năng** |
| 1 | Word | Dùng để lưu từ được truy xuất từ tập tin, có 2 thuộc tính là: - Name: từ trong tiếng Anh. - Meaning: nghĩa của từ trong tiếng Việt. |
| 2 | Node | Class tạo ra các node để lưu từ từ class Word. |
| 3 | DSLK | Đưa các node từ class Node và liên kết lại thành danh sách liên kết đơn. |
| 4 | BangBam | Liên kết nhiều danh sách liên kết đơn lại với nhau thông qua hàm băm để lưu các Node một cách có tổ chức. |

## ***2.2. Thiết kế giao diện***

### **a. Giao diện chính**

# 

Hình 1. Giao diện chính của ứng dụng

Giao diện chính gồm:

- Thanh công cụ chứa các chúc năng chính của chương trình, người dùng có thể bấm vào nút  trên góc trái để sử dụng các chức năng hoặc có thể chọn trực tiếp lên các icon để sử dụng.

- Gồm các nút:

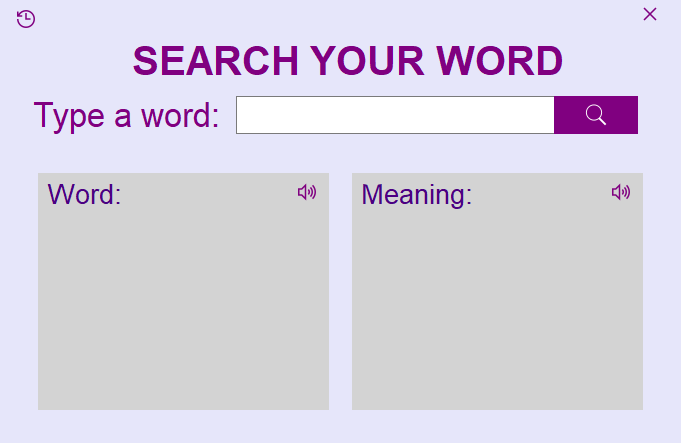
+ để tra từ, để thêm từ.

+ để sửa từ, để xóa từ.

+ để quay trở về giao diện chào mừng, để thoát ứng dụng.

+ để hiển thị các chúc năng, để hiển thị thông tin người thực hiện.

### **b. User control Search**

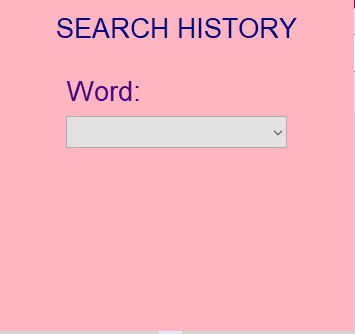
Hình 2. Giao diện tra từ

User control Search gồm:

- Một textbox để nhập từ cần tra.

- Có các nút như: nút tra từ, nút đóng user control, 2 nút phát âm và nút xem lịch sử tra từ.

- 2 panel: một panel chứa từ trong tiếng Anh và một panel chứa nghĩa của từ đó.

 - 1 panel ẩn chính là lịch sử tra từ với 1 combobox hiển thị các từ đã tra.

Hình 3. Giao diện lịch sử tra từ

### **c. User control AddWord**

# 

Hình 4. Giao diện thêm từ

User control AddWord gồm:

- Hai textbox để chứa từ và nghĩa của từ.

- Một button để tắt usercontrol ở góc phải bên trên, cặp button với nút dùng để thêm từ và nút dùng để xóa text của 2 textbox nếu cần thiết.

### **d. User control EditWord**

# 

Hình 5. Giao diện sửa từ

User control EditWord gồm:

- Một combobox dùng để chọn từ cần sửa.

 - Textbox ‘New word’ dùng để sửa từ tiếng Anh, textbox ‘Meaning’ dùng để sửa từ tiếng Việt.

- Nút dùng để lưu từ cần sửa, nút dùng để xóa text trong hai textbox và nút đóng giao diện trên góc phải.

### **e. User control DeleteWord**

# 

Hình 6. Giao diện xóa từ

User control DeleteWord gồm:

 - Một combobox và một textbox, người dùng có thể chọn hoặc gõ từ cần xóa từ 1 trong 2 lựa chọn này.  
 - Nút dùng để xóa từ được chọn, nút dùng để xóa text trong cả textbox và combobox, nút đóng giao diện trên góc phải.

## ***2.3. Thiết kế bảng băm***

### **a. Hàm băm**

Bảng băm được xây dựng theo phương pháp kết nối trực tiếp. Hàm băm có dạng:

public int Bam(char c)

{

if (c >= 'A' && c <= 'Z')

c = (char)((int)c + 32);

return (int)c % 97;

}

Với mỗi từ khi được thêm vào bảng băm sẽ phải thông qua hàm băm trên. Hàm băm sẽ lấy chữ cái đầu tiên của từ đó chuyển sang chữ thường trong trường hợp người dùng vô tình nhập chữ hoa đầu từ sau đó sẽ tiến hành hash với key là 97. Lý do key 97 được chọn là chữ cái ‘a’ trong bảng chữ cái có mã ASCII là 97, như vậy ta có thể hiểu đơn giản là các từ bắt đầu với chữ ‘a’ sẽ nằm ở đầu bảng băm và tăng dần đến chữ ‘z’. Khi băm bằng phương pháp trên đụng độ sẽ xảy ra khi có 2 từ trở lên có ký tự đầu giống nhau. Giải pháp để xử lý đụng độ là thêm các từ bị trùng vào cuối danh sách liên kết của từ đó.

### **b. Thêm từ vào bảng băm**

public void Add(Word wd)

{

char c = GetFirstChar(wd.Name);

int hash = Bam(c);

bool check = Search\_Check(wd.Name);

try

{

if (check == true) throw new Exception("This word has been added!");

Node p = new Node();

p.CreateNode(wd);

Table[hash].AddTail(p);

}

catch (Exception e)

{

}

}

Bảng băm được xây dựng từ nhiều danh sách liên kết đơn và hàm băm dựa trên chữ cái đầu tiên của mỗi từ thì việc đụng độ khi thực hiện hàm là điều không thể tránh khỏi. Dựa theo phương pháp kết nối trực tiếp, cách giải quyết cho việc đụng độ là thêm các từ có giá trị bị trùng vào cuối danh sách liên kết đơn đang được thao tác. Nếu không có đụng độ thì khởi tạo node và thêm vào bình thường.

### **c. Xóa từ trong bảng băm**

public void RemoveWord (string s)

{

char c = GetFirstChar(s);

int hash = Bam(c);

if (s == Table[hash].Head.GetWord().Name)

Table[hash].RemoveHead();

else

{

Node p = new Node();

p = Table[hash].SearchTruoc(s);

try

{

if (Table[hash].Head == null || p == null || Table[hash].Head == Table[hash].Tail || p.pNext == null)

throw new Exception();

Table[hash].RemoveAfter(p);

}

catch (Exception e)

{

Console.WriteLine(e);

}

}

}

Khi cần xóa một từ không cần thiết hoặc từ bị lỗi. Ta cần phải có hàm để xóa từ đó ra khỏi bảng băm. Hàm được thực hiện thông qua hàm băm sau đó sử dụng hàm xóa đầu RemoveHead(); nếu là từ đầu tiên trong danh sách liên kết. Nếu không phải thì khởi tạo một node p mới mang giá trị là vị trí đứng trước từ cần xóa. Hàm SeachTruoc(s); sẽ thực hiện nhiệm vụ là tìm vị trí ở trước từ cần xóa sau đó sử dụng hàm xóa sau RemoveAfter(p) để thực hiện xóa từ.

### **d. Sửa từ trong bảng băm**

public void EditWord(string name, string meaning)

{

char c = GetFirstChar(name);

int hash = Bam(c);

bool check = Search\_Check(name);

try

{

if (check == false)

{

throw new Exception("This word doesn't exist in HashTable");

}

Node p = new Node();

p = Table[hash].Head;

while (p != null)

{

if (p.GetWord().Name == name)

p.word.Meaning = meaning;

p = p.pNext;

}

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

}

Hàm sửa từ tiếp tục được thực hiện dựa trên hàm băm và hàm Search\_Check(name) hỗ trợ cho việc xác định xem từ có tồn tại trong bảng băm hay không. Nếu từ có tồn tại trong bảng băm thì sẽ tiến hành duyệt từ trong danh sách liên kết để tìm đúng từ cần sửa, sau đó cho với meaning được truyền vào ta sẽ gán lại nghĩa của từ trong danh sách liên kết.

## ***2.4. Thiết kế các code chính***

### **a. Hàm tra từ**

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

{

BB = new BangBam();

LoadDataFile();

string s1 = txtWord.Text.Trim();

try

{

if (s1 == "") throw new Exception("Pleaes type something!");

bool check = BB.Search\_Check(s1);

if (check == true)//từ tồn tại

{

lblWordHere.Text = s1;

lblMeaningHere.Text = BB.Search\_Lookup(s1);

cbHistoryWord.Items.Add(s1);

}

else

{

cbHistoryWord.Items.Add(s1);

lblWordHere.Text = "This word doesn't exist!";

lblMeaningHere.Text = "Tu khong ton tai";

}

}

catch (Exception exception)

{

MessageBox.Show(exception.Message, "Error!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

Để tìm 1 từ, ta sẽ khởi tạo class BangBam với các hàm đã được xây dựng sẵn. Dữ liệu lấy từ textbox sau đó truyền vào hàm BB.Search\_Check(s1); để kiểm tra. Nếu từ tồn tại, tiến hành đưa từ và nghĩa vào các label được thiết kế sẵn. Ngược lại, đưa vào label để thông báo đến người dùng rằng từ này không tồn tại.

### **b. Hàm thêm từ**

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

{

BB = new BangBam();

LoadDataFile();

string s1 = txtWord.Text.Trim();

string s2 = txtMeaning.Text;

try

{

if (s1 == "" || s2 == "")

throw new Exception("Please type something!");

bool check = BB.Search\_Check(s1);

if (check == true)

MessageBox.Show("This word existed!", "Warning", MessageBoxButtons.OK,

MessageBoxIcon.Warning);

else

{

string[] names = new string[] { txtWord.Text.Trim() + "@" + txtMeaning.Text };

var file = new FileStream(@"E:\Words\input.txt",

FileMode.Append, FileAccess.Write, FileShare.ReadWrite);

using (StreamWriter sw = new StreamWriter(file))

{

foreach (string s in names)

{

sw.WriteLine(s);

}

sw.Close();

}

BangBam BB;

BB = new BangBam();

LoadDataFile();

MessageBox.Show("Added succesfully!", "Congratulations", MessageBoxButtons.OK,

MessageBoxIcon.Information);

Word wd = new Word(s1, s2);

BB.Add(wd);

txtWord.ResetText();

txtMeaning.ResetText();

file.Close();

}

txtWord.Focus();

}

catch (Exception exception)

{

MessageBox.Show(exception.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

Khởi tạo class BangBam, người dùng nhập từ và nghĩa vào 2 textbox được định sẵn. Thông qua hàm BB.Search\_Check(s1) sẽ cho người dùng biết từ đã tồn tại hay chưa. Nếu chưa sẽ tiến hành thêm từ vào bảng băm kèo theo đó tiến hành việc ghi từ và nghĩa được nhập vào file text.

### **c. Hàm sửa từ**

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

{

BB = new BangBam();

LoadDataFile();

string s1 = cbEdit.Text;

if (s1 == "")

{

s1 = "ccccc";

}

string s2 = txtNewWord.Text.Trim();

if (s2 == "")

{

s2 = "aaaaa";

}

string s3 = txtMeaning.Text;

if (s3 == "")

{

s3 = "bbbbb";

}

string s4 = s1 + "@" + BB.Search\_Lookup(s1);

string tempFile = Path.GetTempFileName();

if (s1 == "ccccc" || s2 == "aaaaa" || s3 == "bbbbb")

MessageBox.Show("Please type something!", "Error", MessageBoxButtons.OK,

MessageBoxIcon.Error);

else

{

try

{

bool check = BB.Search\_Check(s1);

if (check == false)

MessageBox.Show("This word doesn't exist", "Warning", MessageBoxButtons.OK,

MessageBoxIcon.Warning);

else

{

using (var sr1 = new StreamReader(@"E:\Words\input.txt"))

using (var sw2 = new StreamWriter(tempFile))

{

string line;

while ((line = sr1.ReadLine()) != null)

{

if (line != s4)

sw2.WriteLine(line);

}

sr1.Close();

sw2.Close();

}

BB.RemoveWord(s1);

File.Delete(@"E:\Words\input.txt");

File.Move(tempFile, @"E:\Words\input.txt");

string[] names = new string[] { txtNewWord.Text.Trim() + "@" + txtMeaning.Text };

var file = new FileStream(@"E:\Words\input.txt",

FileMode.Append, FileAccess.Write, FileShare.ReadWrite);

using (StreamWriter sw1 = new StreamWriter(file))

{

foreach (string s in names)

{

sw1.WriteLine(s);

}

sw1.Close();

}

file.Close();

MessageBox.Show("Edited succesfully!", "Congratulations", MessageBoxButtons.OK,

MessageBoxIcon.Information);

Word wd = new Word(s2, s3);

BB.Add(wd);

cbEdit.ResetText();

txtNewWord.ResetText();

txtMeaning.ResetText();

}

}

catch (Exception exception)

{

MessageBox.Show(exception.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

Khởi tạo class BangBam, người dùng chọn từ cần sửa từ combobox và điền từ mới và nghĩa mới vào 2 textbox được định sẵn. Thông qua hàm BB.Search\_Check(s1) sẽ cho người dùng biết từ đã tồn tại hay chưa(thường thì từ sẽ không tồn tại nhưng vẫn cần thiết để tránh lỗi xảy ra). Nếu chưa sẽ tiến hành chỉnh sửa từ vào bảng băm bằng cách thêm 1 từ mới vào bằng hàm BB.Add(wd); và xóa từ chỉnh sửa đi bằng hàm BB.RemoveWord(s1) đồng thời file text cũng được xử lí tương tự.

### **d. Hàm xóa từ**

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

{

BB = new BangBam();

LoadDataFile();

string s1 = txtMeaning.Text.Trim();

if (s1 == "")

{

s1 = "zzzzz";

}

string s3 = cbDelete.Text.Trim();

if (s3 == "")

{

s3 = "cccccc";

}

string s2 = s1 + "@" + BB.Search\_Lookup(s1);

string s4 = s3 + "@" + BB.Search\_Lookup(s3);

string tempFile = Path.GetTempFileName();

if (s3 == "cccccc")

{

if (s1 == "zzzzz")

{

MessageBox.Show("Please type or choose something!", "Error", MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

else

{

try

{

bool check = BB.Search\_Check(s1);

if (check == false)

MessageBox.Show("This word doesn't exist!", "Warning", MessageBoxButtons.OK,

MessageBoxIcon.Warning);

else

{

using (var sr2 = new StreamReader(@"E:\Words\input.txt"))

using (var sw3 = new StreamWriter(tempFile))

{

string line;

while ((line = sr2.ReadLine()) != null)

{

if (line != s2)

sw3.WriteLine(line);

}

sr2.Close();

sw3.Close();

}

File.Delete(@"E:\Words\input.txt");

File.Move(tempFile, @"E:\Words\input.txt");

MessageBox.Show("Deleted succesfully!", "Congratulations", MessageBoxButtons.OK,

MessageBoxIcon.Information);

BB.RemoveWord(s1);

cbDelete.ResetText();

txtMeaning.ResetText();

}

}

catch (Exception exception)

{

MessageBox.Show(exception.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

else if (s1 == "zzzzz")

{

try

{

bool check = BB.Search\_Check(s3);

if (check == false)

MessageBox.Show("This word doesn't exist!", "Warning", MessageBoxButtons.OK,

MessageBoxIcon.Warning);

else

{

using (var sr = new StreamReader(@"E:\Words\input.txt"))

using (var sw = new StreamWriter(tempFile))

{

string line;

while ((line = sr.ReadLine()) != null)

{

if (line != s4)

sw.WriteLine(line);

}

sr.Close();

sw.Close();

}

File.Delete(@"E:\Words\input.txt");

File.Move(tempFile, @"E:\Words\input.txt");

MessageBox.Show("Deleted succesfully!", "Congratulations", MessageBoxButtons.OK,

MessageBoxIcon.Information);

BB.RemoveWord(s3);

cbDelete.ResetText();

txtMeaning.ResetText();

}

}

catch (Exception exception)

{

MessageBox.Show(exception.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

cbDelete.Enabled = true;

txtMeaning.Enabled = true;

lblOr.Enabled = true;

lblType.Enabled = true;

lblChoose.Enabled = true;

cbDelete.Items.Clear();

AddComboBox(BB, cbDelete);

}

Khởi tạo class BangBam, người dùng có hai lựa chọn một là chọn từ cần xóa từ combobox hai là điền từ cần xóa vào ô textbox để xóa. Thông qua hàm BB.Search\_Check(s1) sẽ cho người dùng biết từ đã tồn tại hay chưa (chỉ áp dụng với ô textbox). Nếu chưa sẽ tiến hành xóa từ đi bằng hàm BB.RemoveWord(s1) đồng thời file text cũng được xử lí tương tự.

### **e. Hàm đọc file**

public void LoadDataFile()

{

var file = new FileStream(@"E:\Words\input.txt", FileMode.Open, FileAccess.Read, FileShare.ReadWrite);

var sr = new StreamReader(file);

string s;

for (int i = 0; i < TotalLines(@"E:\Words\input.txt"); i++)

{

string name = null, meaning = null;

s = sr.ReadLine();

int j = 0;

while (s[j] != '@')

{

name += s[j];

j++;

}

j++;

while (j < s.Length)

{

meaning += s[j];

j++;

}

Word wd = new Word(name, meaning);

BB.Add(wd);

}

file.Close();

sr.Close();

}

### 

# **3. Cài đặt và kiểm thử**

## ***3.1. Cài đặt***

- Ứng dụng được xây dựng trên nền tảng Visual Studio 2015 bởi ngôn ngữ lập trình C# cùng với Windows Form và User Control.

- Dữ liệu được tổ chức dưới dạng file txt, thực hiện lấy dữ liệu bên trong ứng dụng.

- Số lượng danh sách liên kết đơn phụ thuộc vào số lượng từ có trong file text.

- Tốc độ tìm kiếm từ là O(1).

## ***3.2. Kiểm thử***

- Với việc tra từ, nhập từ vào textbox sau đó nhấn nút tìm kiếm. Nếu việc tìm kiếm thành công (từ có trong từ điển) sẽ trả về kết quả là từ được nhập vào và nghĩa của từ.

- Với việc thêm từ, nhập từ và nghĩa vào các textbox và thêm. Nếu việc thêm thành công thì từ sẽ được nằm trong bảng băm và khi tra từ thì từ mới được thêm vào cũng sẽ xuất hiện, khi xóa hoặc sửa từ thì từ đó cũng sẽ xuất hiện để thao tác.

- Để nghe cách phát âm ta có thể click vào biểu tượng loa để nghe (yêu cầu mạng internet).

# **4. Phân công công việc**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Mô tả công việc** | **Võ Gia Huy** | **Trần Tấn Quý** | **Chi tiết** |
| 1 | Thiết kế class Word |  | x |  |
| 2 | Thiết kế class DSLK | x |  |  |
| 3 | Thiết kế class BangBam |  | x |  |
| 4 | Thiết kế các form, user control |  | x |  |
| 5 | Thiết kế phần nghe từ | x |  |  |
| 6 | Thực hiện hàm thêm, sửa, xóa từ | x |  |  |
| 7 | Thực hiện hàm tra từ |  | x |  |

# **5. Kết luận**

## ***5.1. Kết quả***

Nhóm đã xây dựng thành công đề tài là chương trình từ điển Anh-Việt ứng dụng bảng băm của môn học Cấu trúc dữ liệu & Giải thuật.

## ***5.2. Khó khăn***

- Do lần đầu nhóm được tiếp cận GitHub nên cũng rất bỡ ngỡ và gặp không ít lỗi cũng như thắc mắc về cơ chế hoạt động của GitHub nhưng nhờ sự giải đáp của thầy thì nhóm cũng đã hiểu hơn về GitHub và đã có thể thao tác một cách hiệu quả.

- Việc tiếp cận để hiểu về cách hoạt động của hàm băm cũng như bảng băm lúc đầu tiêu tốn không ít thời gian và một lần nữa với sự trợ giúp của thầy, nhóm đã nắm rõ hơn về nó.

## ***5.3. Ưu điểm***

Giao diện đẹp mắt, hiện đại, tiện lợi và thân thiện với người dùng, tốc độ tra cứu từ nhanh chóng được tích hợp với tính năng nghe từ cực kỳ hữu dụng. Ngoài ra người dùng cũng có thể tùy chỉnh từ điển theo ý thích.

## ***5.4. Nhược điểm***

- Với việc tổ chức dữ liệu trên file, tốc độ truy xuất dữ liệu sẽ phần nào chậm hơn so với các cách tổ chức dữ liệu khác. Đặc biệt không tiện lợi trong việc lưu trữ thêm các chi tiết về từ ngữ (VD: loại từ, cách phát âm,…).

- Để nghe được từ vẫn còn phải yêu cầu mạng Internet.

- Phần nghe từ tiếng Việt hoạt động kém ổn định.

## ***5.5. Hướng phát triển***

- Ta có thể dùng cơ sở dữ liệu để lưu trữ nhiều thông tin cho từ trong từ điển hơn (VD: loại từ, cách phát âm, các trường hợp ứng dụng,…) thay vì chỉ đơn giản là từ và nghĩa trên file text.

- Tích hợp hệ thống speech to text vào ứng dụng để tăng tính tiện lợi, có thể nghe từ mọi lúc mọi nơi mà không yêu cầu mạng Internet.

- Xây dựng hệ thống gợi ý từ.