**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**

Lót

**MỤC LỤC**

[**NỘI DUNG** 1](#_Toc529998308)

[**1. Mô tả project** 1](#_Toc529998309)

[***1.1. Công dụng của phần mềm*** 1](#_Toc529998310)

[***1.2. Use case diagram*** 1](#_Toc529998311)

[**2. Quá trình thực hiện** 2](#_Toc529998312)

[***2.1. Thiết kế các class*** 2](#_Toc529998313)

[***2.2. Thiết kế giao diện*** 3](#_Toc529998314)

[**a. Giao diện chào mừng** 3](#_Toc529998315)

[**b. Giao diện chính** 4](#_Toc529998316)

[**c. User control Search** 5](#_Toc529998317)

[**d. User control AddWord** 6](#_Toc529998318)

[**e. User control EditWord** 7](#_Toc529998319)

[**f. User control DeleteWord** 8](#_Toc529998320)

[***2.3. Thiết kế các class chính*** 9](#_Toc529998321)

[**a. Class Word** 9](#_Toc529998322)

[**b. Class Node** 9](#_Toc529998323)

[**c. Class DSLK** 10](#_Toc529998324)

[**d. Class BangBam** 13](#_Toc529998325)

[**e. Class SpeakText** 17](#_Toc529998326)

[***2.4. Thiết kế các code chính*** 18](#_Toc529998327)

[**a. Hàm tra từ** 18](#_Toc529998328)

[**b. Hàm thêm từ** 19](#_Toc529998329)

[**c. Hàm sửa từ** 20](#_Toc529998330)

[**d. Hàm xóa từ** 22](#_Toc529998331)

[**e. Hàm đọc file** 25](#_Toc529998332)

[**f. Hàm load từ lên combobox từ bảng băm** 26](#_Toc529998333)

[**g. Hàm đếm dòng trong file text** 26](#_Toc529998334)

[**3. Cài đặt và kiểm thử** 27](#_Toc529998335)

[***3.1. Cài đặt*** 27](#_Toc529998336)

[***3.2. Kiểm thử*** 27](#_Toc529998337)

[**4. Phân công công việc** 28](#_Toc529998338)

[**5. Kết luận** 28](#_Toc529998339)

[***5.1. Kết quả*** 28](#_Toc529998340)

[***5.2. Khó khăn*** 28](#_Toc529998341)

[***5.3. Hướng phát triển*** 29](#_Toc529998342)

[***5.4. Ưu điểm*** 29](#_Toc529998343)

[***5.5. Nhược điểm*** 29](#_Toc529998344)

# **NỘI DUNG**

# **1. Mô tả project**

## ***1.1. Công dụng của 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.

## ***1.2. Use case diagram***

Từ điển Anh-Việt

User

|  |  |
| --- | --- |
| **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. |

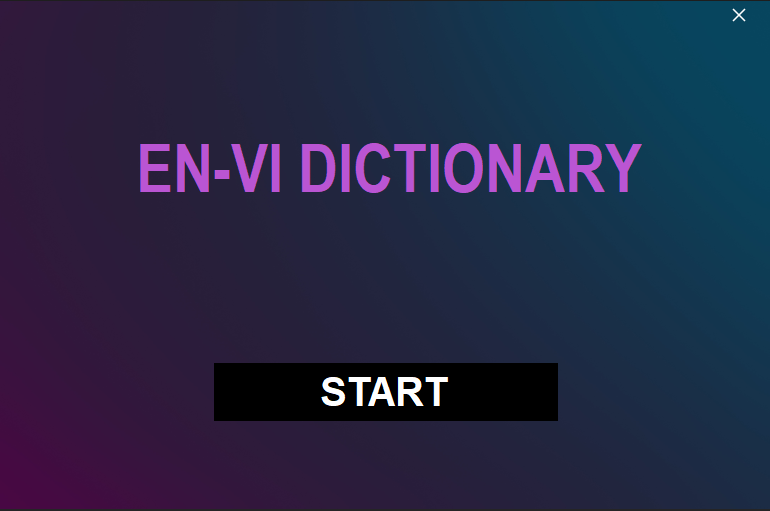
# **2. Quá trình thực hiện**

## ***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 chào mừng**

Hình 1. Giao diện màn hình khóa

Người dùng chỉ cần click START để bắt đầu sử dụng từ điển. Nếu người dùng không muốn tiếp tục sử dụng, đơn giản bấm vào nút ‘x’ ở góc trên bên phải.

### **b. Giao diện chính**

Hình 2. 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:

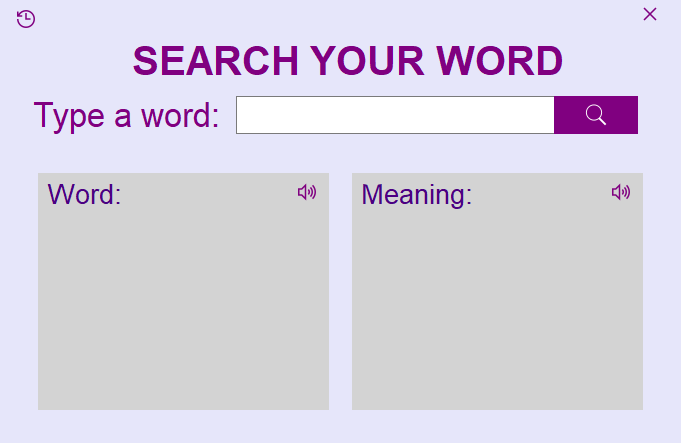
+ để tra từ, để thêm từ.

+ 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.

### **c. User control Search**

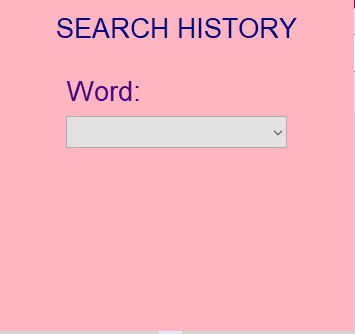
Hình 3. Giao diện tra từ

User control Search gồm:

- Một textbox để nhập từ cần tra.

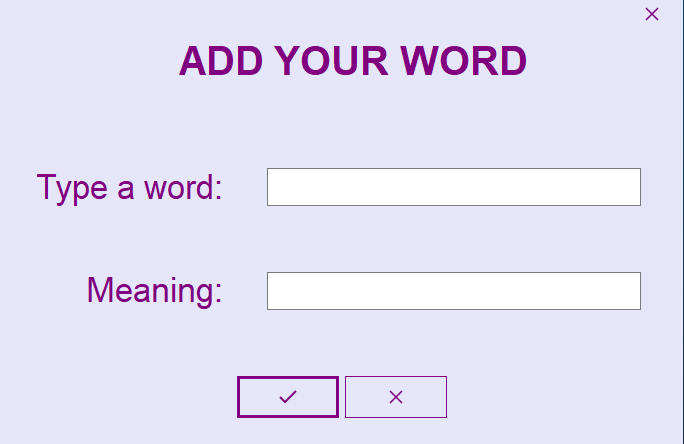
- Có các nút như: nút tra từ, nút đóng user control, 2 nút phát âm và nút xem lịch sử tra từ.

- 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 3.1. Giao diện lịch sử tra từ

### **d. 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.

### **e. User control EditWord**

  
Hình 5. Giao diện sửa từ

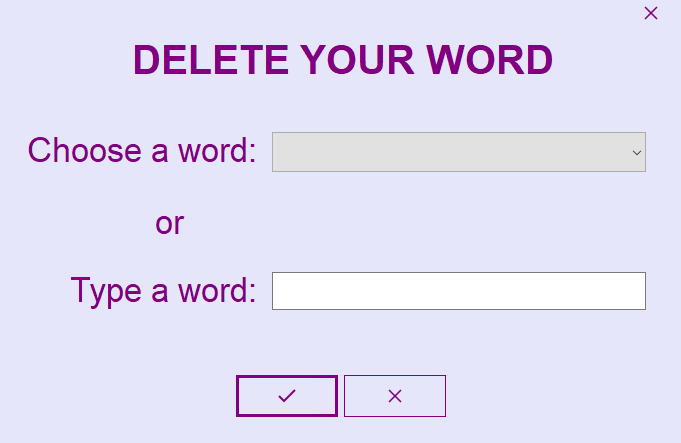
User control EditWord gồm:

- 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.

### **f. 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 các class chính***

### **a. Class Word**

public class Word

{

private string name;

private string meaning;

public Word(string name, string meaning)

{

this.name = name;

this.meaning = meaning;

}

public string Name

{

set { name = value; }

get { return name; }

}

public string Meaning

{

set { meaning = value; }

get { return meaning; }

}

}

### **b. Class Node**

public class Node

{

public Word word;

public Node pNext;

public Node()

{

word = new Word(null, null);

pNext = null;

}

public void CreateNode(Word wd)

{

word.Name = wd.Name;

word.Meaning = wd.Meaning;

pNext = null;

}

public void Next(Node x)

{

pNext = x; // ham thay doi pNext

}

public Word GetWord()

{

return word;

}

public Node GetNext()

{

return pNext;

}

}

### **c. Class DSLK**

public class DSLK

{

public Node Head;

public Node Tail;

public DSLK()

{

Head = Tail = null;

}

public void AddHead(Node p)

{

if (Head == null)

Head = Tail = p;

else

{

p.Next(Head);

// Cach 2:

// p.pNext = Head;

Head = p;

}

}

public void AddTail(Node p)

{

if (Head == null)

Head = Tail = p;

else

{

Tail.Next(p);

Tail = p;

}

}

public void Duyet()

{

Node p = Head;

try

{

if (p == null)

throw new Exception("Empty List");

while (p != null)

{

Console.WriteLine("{0} {1}", p.word.Name, p.word.Meaning);

p = p.pNext;

}

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

public Node Search(string s)

{

Node p = Head;

while (p != null)

{

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

return p;

p = p.pNext;

}

return null;

}

public Node SearchTruoc(string s)

{

Node p = Head.pNext;

Node q = Head;

if (Head != Tail)

{

while (p != null)

{

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

return q;

p = p.pNext;

q = q.pNext;

}

}

return null;

}

public void AddAfter(Node p, Node q)

{

try

{

if (Head == null || Head == Tail || p == null)

throw new Exception("Can not Add");

q.Next(p.GetNext());

p.Next(q);

if (p == Tail)

Tail = q;

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

public void RemoveHead()

{

try

{

if (Head == null)

throw new Exception("Empty List");

Node p = Head;

Head = p.GetNext();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

public void RemoveAfter(Node p)

{

try

{

if (p.pNext == null || Head == null || Head == Tail || p == null)

throw new Exception("Can not Remove");

Node q = p.GetNext();

p.Next(q.GetNext());

if (p.GetNext() == null)

Tail = p;

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

}

}

### **d. Class BangBam**

public class BangBam

{

public DSLK[] Table;

public BangBam()

{

Table = new DSLK[100];

for (int i = 0; i < 100; i++)

Table[i] = new DSLK(); //khởi tạo một danh sách lk với n phần tử

}

public int Bam(char c)

{

if (c >= 'A' && c <= 'Z') //nếu ký tự đầu là chữ in hoa từ A->Z, cộng 32 để chuyển

c = (char)((int)c + 32); //sang chữ thường, sau đó ép kiểu int cho ký tự

return (int)c % 97; //và hash với key là 97

}

public char GetFirstChar(string s)

{

return s[0]; //lấy ký tự đầu tiên nhằm cho mục đích hash

}

public void Duyet()

{

for (int i = 0; i < 100; i++)

if (Table[i].Head != null)

Table[i].Duyet(); //Duyệt các phần tử trong dslk

}

//kiểm tra từ có tồn tại hay không

public bool Search\_Check(string s)

{

char c = GetFirstChar(s);

int hash = Bam(c);

Node p = new Node();

p = Table[hash].Head;

while (p != null)

{

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

return true;

p = p.pNext;

}

return false;

}

public string Search\_Lookup(string s)

{

char c = GetFirstChar(s);

int hash = Bam(c);

bool check = Search\_Check(s);

string k = null;

try

{

if (check == false)

{

throw new Exception("HashTable doesn't have this word");

}

Node p = new Node();

p = Table[hash].Head;

while (p != null)

{

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

k = p.GetWord().Meaning;

p = p.pNext;

}

return k;

}

catch (Exception e)

{

Console.WriteLine(e.Message);

return k;

}

}

//thêm từ

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)

{

}

}

//xóa từ

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);

}

}

}

//sửa từ

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);

}

}

}

### **e. Class SpeakText**

public class SpeakText

{

private WebBrowser wbWeb;

public WebBrowser WbWeb

{

get { return wbWeb; }

set { wbWeb = value; }

}

public SpeakText(WebBrowser wb)

{

this.WbWeb = wb;

}

private void SetText(string data)

{

HtmlElement element = WbWeb.Document.GetElementById("text");

element.SetAttribute("value", data);

}

private void Speak()

{

HtmlElement element = WbWeb.Document.GetElementById("playbutton");

element.InvokeMember("click");

}

public void Spreak(string data)

{

//SetText("");

SetText(data);

Speak();

}

}

## ***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);

}

}

### **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);

}

}

### **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);

}

}

}

### **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);

}

### **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();

}

### **f. Hàm load từ lên combobox từ bảng băm**

public static void AddComboBox(BangBam BB, ComboBox cbdelete)

{

int n = 0;

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

string line;

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

{

n++;

}

sr.Close();

Node p = new Node();

for (int i = 0; i < n; i++)

{

p = BB.Table[i].Head;

while (p != null)

{

cbdelete.Items.Add(p.word.Name);

p = p.pNext;

}

}

}

### **g. Hàm đếm dòng trong file text**

int TotalLines(string filePath)

{

using (StreamReader r = new StreamReader(filePath))

{

int i = 0;

while (r.ReadLine() != null) { i++; }

return i;

}

}

# **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.

- Phép băm được thực hiện bằng cách lấy ký tự đầu của mỗi từ chuyển sang chữ in thường (nếu input là chữ in hoa) sau đó chia lấy dư với khóa là 97 (do chữ ‘a’ bắt đầu với mã ASCII là 97).

- Độ phức tạp của phép toán trên bảng băm 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.

- 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***

Sau 8 tuần học tập và thực hiện đề tài cùng với sự giúp đỡ nhiệt tình của thầy Trần Công Tú, nhóm chúng em 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 chúng em 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 chúng em 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, chúng em đã nắm rõ hơn về nó.

## ***5.3. 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.

## ***5.4. 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.

## ***5.5. 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.