ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện CNTT & Truyền thông



**BÁO CÁO BÀI TẬP LỚN C ADVANCE**

1. Nguyễn Tài Thao - 20176875

2. Lương Đức Minh - 20176821

3. Vũ Nguyễn Khánh - 20176795

4. Nguyễn Thị Ngân - 20176834

**LỜI MỞ ĐẦU**

Bản báo cáo khái quát quá trình thực hiện bài tập lớn, gồm có 5 phần:

* Phần I trình bày yêu cầu của bài tập lớn.
* Phần II định hướng cách giải quyết bài tập và phân công công việc cho từng người trong nhóm.
* Phần III gồm code của các hàm chính (tìm kiếm, gợi ý, thêm, sửa, xóa).
* Phần IV là hình ảnh demo chương trình do nhóm thực hiện.

Chương trình từ điển được nhóm viết tham khảo các anh chị khóa 60, tìm hiểu về GTK+ từ trang web [https://developer.gnome.or](https://developer.gnome.or/)g .

Chương trình từ điển Anh – Việt và bản báo cáo chắc chắn sẽ không tránh khỏi những sai sót. Cả nhóm rất mong nhận được ý kiến của thầy giáo và các bạn.

Xin chân thành cảm ơn.

**Hà nội, ngày 15 tháng 5 năm 2019**

**PHẦN I: NHỮNG YÊU CẦU CỦA MINI PROJECT**

Viết một chương trình quản lý từ điển Anh – Việt có giao diện đồ họa gồm các chức năng:

1. Thêm/Tìm kiếm/Xóa (sử dụng B-tree)
2. Hỗ trợ tìm kiếm giống với tính năng “auto – complete”
3. Tìm kiếm có gợi ý
4. Kiểm thử chương trình với số lượng từ lớn

*\*\*File description.txt mô tả hướng dẫn chạy code*

**PHẦN II: ĐỊNH HƯỚNG GIẢI QUYẾT BÀI TOÁN VÀ PHÂN CHIA CÔNG VIỆC**

1. **Định hướng giải quyết bài toán:**

* Tìm kiếm và tạo một file dữ liệu từ điển từ với số lượng từ lớn.
* Sử dụng GTK+ để làm giao diện đồ họa cho từ điển.
* Sử dụng thư viện B – Tree để viết các hàm tìm kiếm/thêm/sửa/xóa.

*(\*)Tại sao lại dùng B-tree:*

*- Trong quá trình làm việc, bọn em nghĩ là cần một loại cấu trúc dữ liệu có tính chất đối xứng và có thứ tự để cho truy cập tới mọi định nghĩa từ đều có thời gian giống nhau. Một trong những giải pháp là B-tree.*

*- Dùng B-tree có thể truy cập trực tiếp một lướng lớn từ trên đĩa mềm mà không tốn RAM,nhờ cấu trúc đặc trưng của B-tree nên thời gian truy cập đến các từ nhanh,đều như nhau hoặc chênh nhau rất ít.*

* Phải có hỗ trợ hoàn thành tìm kiếm.
* Sử dụng soundex library để gợi ý tìm kiếm.
* Test chương trình với các kích thước từ điển khác nhau.
* Viết báo cáo .

1. **Phân chia công việc:**

* Nguyễn Tài Thao: Viết giao diện và làm báo cáo
* Lương Đức Minh: Viết hàm thao tác với file và build B-tree
* Vũ Nguyễn Khánh: Viết hàm gợi ý và tính năng tự động hoàn thành
* Nguyễn Thị Ngân: Viết hàm Thêm/Chỉnh sửa/Xóa

**PHẦN III: CÁC HÀM CHÍNH ĐƯỢC SỬ DỤNG**

1. **Hàm tra từ trong từ điển:**

void display\_meaning(GtkWidget \*widget, gpointer soundexTree){

int rsize, beExisted, i;

char word[50], meaning[5000], suggest[100][100];

strcpy(word, gtk\_entry\_get\_text(GTK\_ENTRY(textSearch)));

beExisted = btsel(data, word, meaning, 5000, &rsize);

if(beExisted == 0){

gtk\_text\_buffer\_set\_text(buffer, meaning, -1);

}

else{

gtk\_text\_buffer\_set\_text(buffer, "Opps! We have no idea about this word", -1);

}

}

1. **Hàm thêm từ vào trong từ điển:**

void add\_new\_word(GtkWidget \*widget, gpointer pointer){

char word[50], meantext[10000], soundex[5];

int rsize, insertDictResult, insertSoundexResult;

data\_struct \*pt = (data\_struct\*)pointer;

GtkWidget \*inputtext = (GtkWidget\*\*)(pt->data\_array[0]);

GtkWidget \*mean = (GtkWidget\*\*)(pt->data\_array[1]);

GtkWidget \*dialog = (GtkWidget\*\*)(pt->data\_array[2]);

BTA \*soundexTree = (BTA\*\*)(pt->tree);

GtkTextIter st\_iter;

GtkTextIter ed\_iter;

gtk\_text\_buffer\_get\_start\_iter (gtk\_text\_view\_get\_buffer(GTK\_TEXT\_VIEW(mean)), &st\_iter);

gtk\_text\_buffer\_get\_end\_iter (gtk\_text\_view\_get\_buffer(GTK\_TEXT\_VIEW(mean)), &ed\_iter);

gtk\_text\_buffer\_get\_text(gtk\_text\_view\_get\_buffer(GTK\_TEXT\_VIEW(mean)), &st\_iter, &ed\_iter, FALSE);

strcpy(meantext, gtk\_text\_buffer\_get\_text(gtk\_text\_view\_get\_buffer(GTK\_TEXT\_VIEW(mean)), &st\_iter, &ed\_iter, FALSE));

strcpy(word, gtk\_entry\_get\_text(GTK\_ENTRY(inputtext)));

if(word[0] == '\0' || meantext[0] == '\0')

Show\_message(dialog, GTK\_MESSAGE\_WARNING, "Cảnh báo!", "Không được bỏ trống phần nào.");

else if(btsel(data, word, meantext, 5000, &rsize) == 0)

Show\_message(dialog, GTK\_MESSAGE\_WARNING, "Xảy ra lỗi!", "Từ vừa nhập đã có trong từ điển.");

else{

insertDictResult = btins(data, word, meantext, strlen(meantext) + 1);

SoundEx(soundex, word, 4, 1);

insertSoundexResult = btins(soundexTree, word, soundex, 5\*sizeof(char));

if(insertDictResult == 0 && insertSoundexResult == 0)

Show\_message(dialog, GTK\_MESSAGE\_INFO, "Thành công!", "Đã thêm từ vừa nhập vào từ điển.");

else

Show\_message(dialog, GTK\_MESSAGE\_INFO, "Xảy ra lỗi!", "Có lỗi bất ngờ xảy ra.");

}

}

1. **Hàm sửa từ có sẵn trong từ điển:**

data\_struct\* pt = (data\_struct\*)pointer;

GtkWidget \*inputtext = (GtkWidget\*\*)(pt->data\_array[0]);

GtkWidget \*mean = (GtkWidget\*\*)(pt->data\_array[1]);

GtkWidget \*dialog = (GtkWidget\*\*)(pt->data\_array[2]);

BTA \*soundexTree = (BTA\*\*)(pt->tree);

GtkTextIter st\_iter;

GtkTextIter ed\_iter;

int result, rsize;

char word[50], meantext[10000];

gtk\_text\_buffer\_get\_start\_iter (gtk\_text\_view\_get\_buffer(GTK\_TEXT\_VIEW(mean)), &st\_iter);

gtk\_text\_buffer\_get\_end\_iter (gtk\_text\_view\_get\_buffer(GTK\_TEXT\_VIEW(mean)), &ed\_iter);

strcpy(word, gtk\_entry\_get\_text(GTK\_ENTRY(inputtext)));

if (word[0] == '\0' || meantext[0] == '\0')

Show\_message(dialog, GTK\_MESSAGE\_WARNING, "Cảnh báo!",

"Không được bỏ trống phần nào.");

else if(btsel(data, word, meantext, 5000, &rsize) != 0)

Show\_message(dialog, GTK\_MESSAGE\_ERROR, "Xảy ra lỗi!",

"Không tìm thấy từ này trong từ điển.");

else {

strcpy(meantext, gtk\_text\_buffer\_get\_text(gtk\_text\_view\_get\_buffer(GTK\_TEXT\_VIEW(mean)), &st\_iter, &ed\_iter, FALSE));

result = btupd(data, word, meantext, strlen(meantext) + 1);

if (result == 0){

Show\_message(dialog, GTK\_MESSAGE\_INFO, "Thành công!",

"Đã cập nhật lại nghĩa của từ trong từ điển.");

}

else

Show\_message(dialog, GTK\_MESSAGE\_ERROR, "Xảy ra lỗi!", "Có lỗi bất ngờ xảy ra.");

}

}

1. **Hàm xóa từ:**

void delete\_word\_from\_dict(char \*word, BTA\* soundexTree){

int delInDictResult, delInSoundexResult;

char a[100] = "Đã xóa từ ";

delInDictResult = btdel(data, word);

delInSoundexResult = btdel(soundexTree, word);

if (delInDictResult == 0 && delInSoundexResult == 0)

Show\_message(window, GTK\_MESSAGE\_INFO, "Thành công!", strcat(strcat(a, word), " khỏi từ điển"));

else

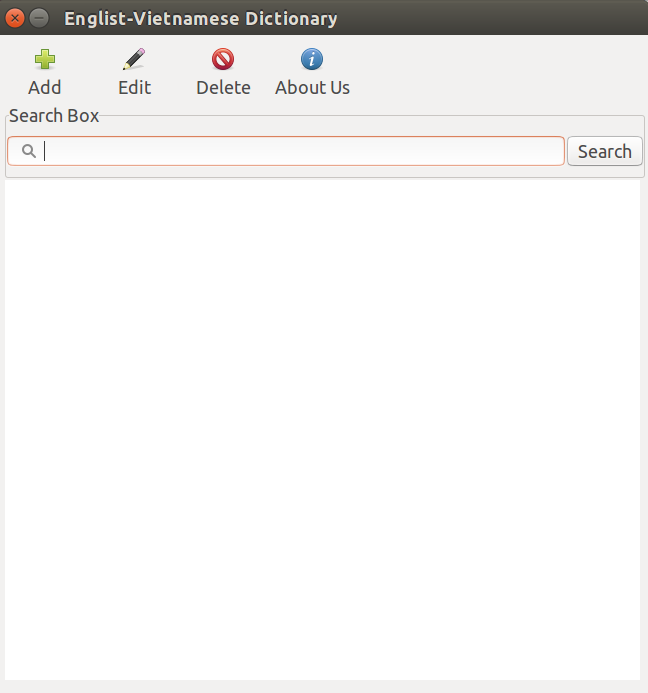
Show\_message(window, GTK\_MESSAGE\_ERROR, "Xảy ra lỗi!", "Có lỗi bất ngờ xảy ra.");

gtk\_entry\_set\_text(GTK\_ENTRY(textSearch), "");

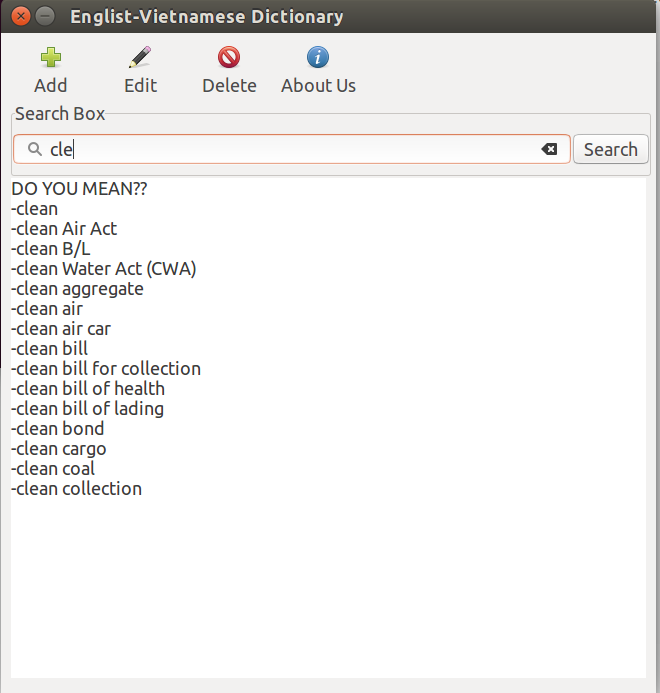
}

**PHẦN IV: DEMO CHƯƠNG TRÌNH**

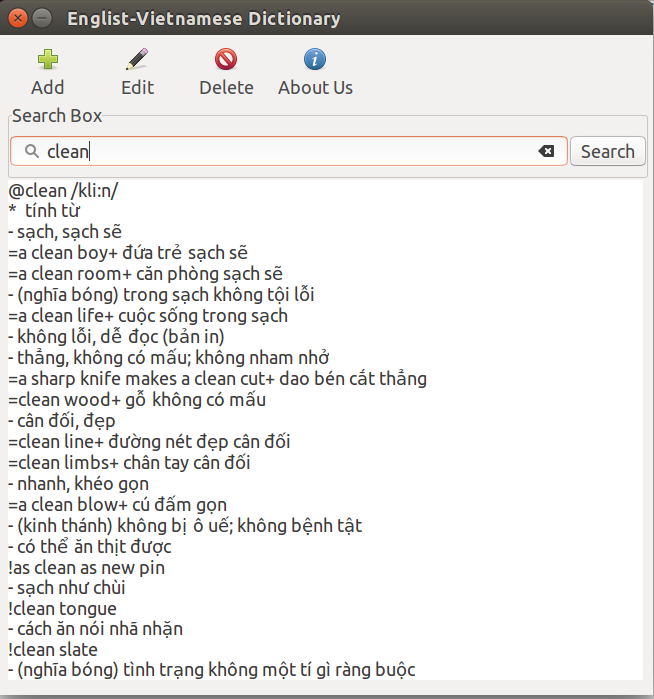
1. **Giao diện đồ họa tổng quan**

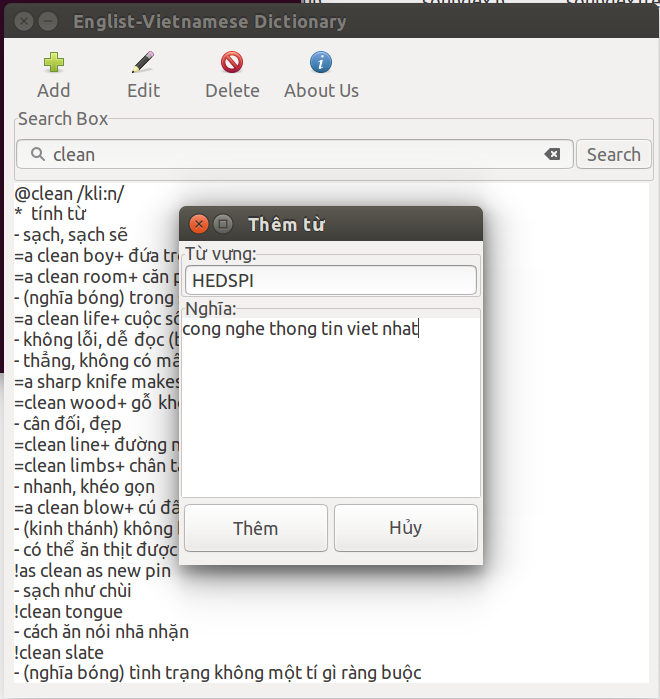
****

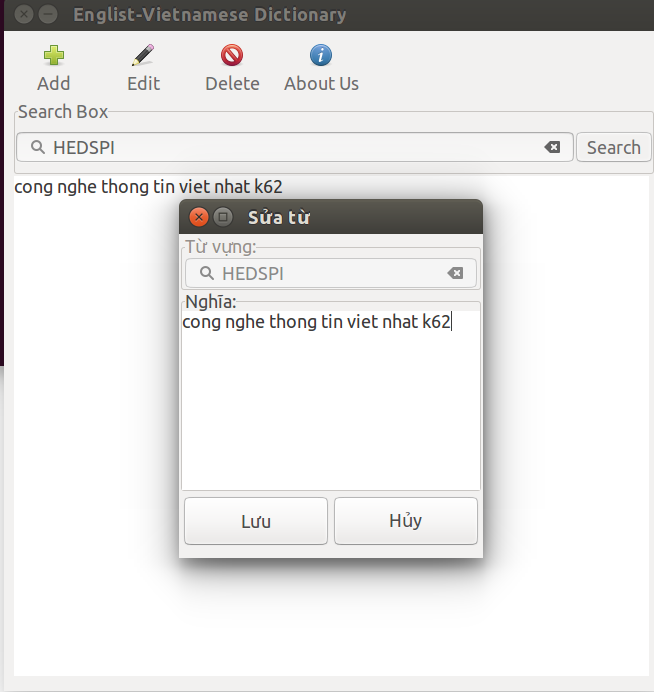
1. **Gợi ý tìm kiếm**

****

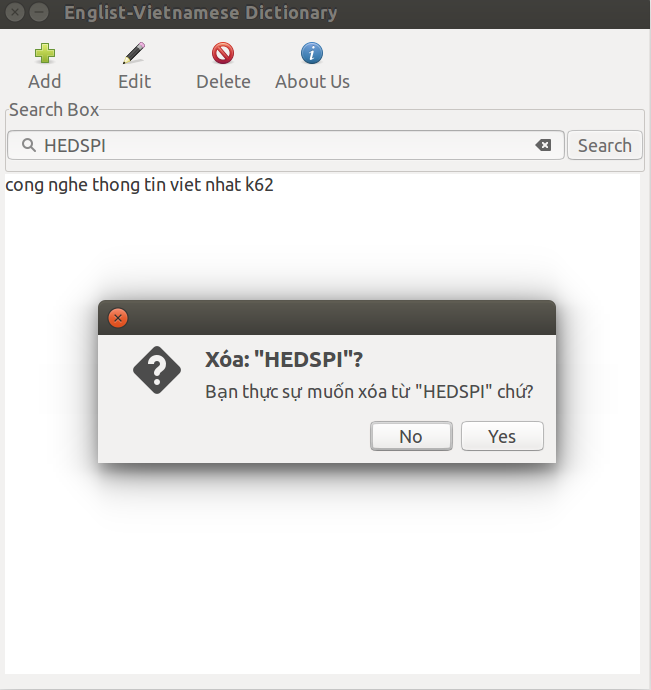
1. **Tra nghĩa**

****

1. **Thêm từ**
2. **Sửa từ**

****

1. **Xóa từ**

****