

Báo cáo Project từ điển

Môn học : C Advanced

Thành viên nhóm:

Nguyễn Kiều Thương - 20153728

Dương Ngọc Sơn - 20153177

Trịnh Hữu Vũ - 20154412

Đỗ Thị Hải - 20151181

Mục Lục

1. Tài liệu tham khảo
2. Yêu cầu Project
3. Nhiệm vụ bài toán
4. Quy trình thực hiện
5. Tổ chức chương trình và các hàm chính
6. Kết quả và đánh giá thời gian
7. Nhận xét và kết luận

1. Tài liệu tham khảo

+ Tài liệu hướng dẫn sử dụng B-tree trên trang
<http://www.hydrus.org.uk/doc/bt/html/index.html>

+ Tài liệu hướng dẫn sử dụng GTK+ trên trang

<http://zetcode.com/gui/gtk2/>

+ Tham khảo code của anh chị khóa K57.

2. Yêu cầu Project

Chương trình quản lý từ điển máy tính có chức năng:

- Add/Search/Delete từ (sử dụng B-Tree)
- Hỗ trợ hoàn thiện tìm kiếm. Ví dụ khi gõ “comput và ấn <tab>, từ “computer” sẽ được hoàn thiện (giống trong Bash Shell)
- Gợi ý search => sử dụng soundex library

3. Nhiệm vụ bài toán

- Tìm kiếm dữ liệu từ điển, chuyển đổi sang dữ liệu Btree (find and convert dictionary data)

- Tạo giao diện đồ họa (using GTK+)

- Tìm kiếm, thêm từ, xóa từ

- Gợi ý từ, Auto-complete sử dụng Tab key

4. Quy trình thực hiện

4.1. Phân công nhiệm vụ

- Dương Ngọc Sơn : hàm “tìm kiếm từ” và giao diện chính
- Trịnh Hữu Vũ : hàm “thêm từ”
- Đỗ Thị Hải : hàm “soundex” và các hộp thoại “trợ giúp” và “thông tin”
- Nguyễn Kiều Thương : hàm “xóa từ”

4.2. Nghiên cứu thiết kế

4.3. Kết hợp các công việc đã phân chia

4.4. Hoàn thiện sản phẩm và kiểm thử, rà soát lỗi

4.5. Viết báo cáo và slide

5. Tổ chức chương trình

- Chương trình bao gồm các hàm chính sau :

5.1 Hàm “show_meaning”

5.2 Hàm “display”

5.3 Hàm “searchword”

5.4 Hàm “add_word”

5.5 Hàm “delete_word”

5.1 Hàm *show_meaning*

- Chức năng: Hiện chuỗi “meaning” vào hộp thoại hiện nghĩa.

- CODE:

```
void *show_meaning(char *meaning){
    FILE *stream;
    GtkTextIter iter;
    char line[2000];
    static int created=0;
    // Hàm chuyển 1 string tương đương thành 1 file
    stream=fmemopen(meaning,strlen(meaning)+1,"r");
    gtk_text_buffer_set_text(buffer,"",-1);
    gtk_text_buffer_get_start_iter(buffer,&iter);
    //Tag kiểu chu
    if (!created){
        gtk_text_buffer_create_tag(buffer,"lmarg15","left_margin",15,NULL);
        gtk_text_buffer_create_tag(buffer,"lmarg30","left_margin",30,NULL);
        gtk_text_buffer_create_tag(buffer,"lmarg35","left_margin",35,NULL);

        gtk_text_buffer_create_tag(buffer,"bold","weight",PANGO_WEIGHT_BOLD,NUL
        L);
        created=1;
    }
    //In nghĩa
    while (fgets(line,2000,stream)!=NULL){
        switch(line[0]){
            case '@':
                line[0]=' ';
                gtk_text_buffer_insert(buffer,&iter,"\n",-1);
                gtk_text_buffer_insert_with_tags_by_name(buffer,&iter,line,-1,"bold",NULL);
                gtk_text_buffer_insert(buffer,&iter,"\n",-1);
                break;
            case '*':
                line[0]=' ';
                gtk_text_buffer_insert_with_tags_by_name(buffer,&iter,line,-
                1,"lmarg15",NULL);
                break;
            case '!':
                line[0]=' ';
                gtk_text_buffer_insert_with_tags_by_name(buffer,&iter,line,-
                1,"lmarg15",NULL);
                break;
            case '-':
                line[0]=' ';
```



```
    gtk_text_buffer_insert_with_tags_by_name(buffer,&iter,line,-
1,"lmarg30",NULL);
    break;
case '=':
    line[0]=' ';
    gtk_text_buffer_insert_with_tags_by_name(buffer,&iter,line,-
1,"lmarg35",NULL);
    break;
default:
    gtk_text_buffer_insert(buffer,&iter,line,-1);
    break;
}
}
}
```

5.2 Hàm display

- Chức năng:

+ Kiểm tra xem người dùng đã nhập từ vào ô tìm kiếm hay chưa.

Nếu chưa hiện thị thông báo.

+ Kiểm tra xem từ đó có hay không trong cơ sở dữ liệu. Nếu có thì hiện thị nghĩa từ đó bằng hàm show_meaning, nếu không thì thông báo và đưa ra gợi ý từ gần giống nếu có.

- CODE:

```
gboolean display(GtkWidget *entryword,gpointer database){
    GtkWidget *dialog,*window=mainwindow;
    char word[50],mean[10000];
    int value;
    BTint valnext;
    strcpy(word,(char *)gtk_entry_get_text(GTK_ENTRY(entryword)));
    if (strlen(word) == 0)
    {
        strcpy(mean,"Bạn chưa nhập từ vào ô tìm kiếm!");
    }
    else
    {
        if (bfndky(data,word,(BTint *)&value)==0)
        {
            btsel(data,word,mean,10000,&value);
        }
        else
        {
            //Tim va hien thi suggestion word neu co
            strcpy(mean,"Xin lỗi! Từ này chưa có trong dữ liệu! \n\n Bạn có thể thêm từ này vào từ điển!\n");
            int index = 0, count = 0;
            char next_near_word[100];
            bfndky(data,word,&valnext);
            while ((index <= 100) && (count < 1)){
                bnxtky(data, next_near_word, &valnext);
                if (suggest_word (word, next_near_word) == 0) {
                    strcat (mean, "Gợi ý :\n");
                    strcat (mean, next_near_word);
                    strcat (mean, "\n");
                    count++;
                } else
                    index++;
            }
        }
    }
}
```

```
    }  
  }  
}  
show_meaning(mean);  
}
```

5.3 Hàm searchword

- Chức năng : tìm kiếm hỗ trợ Auto-complete Tab và suggestion search.

- CODE:

```
gboolean searchword(GtkWidget *entryword,GdkEvent *event,gpointer listword){
    GdkEventKey *key=(GdkEventKey *)event;
    GtkListStore *liststore=(GtkListStore *)listword;
    GtkTreeIter iter;
    int count=0,len;
    int value;
    char text[50],near_word[50], *mean = (char *)malloc (10000);

    // Auto-complete Tab
    t_begin = clock();

    if (key->keyval==GDK_KEY_Tab){
        strcpy(text,(char*)gtk_entry_get_text(GTK_ENTRY (entryword)));
        if (btsel(data,text, mean, 10000,&value)==QNOKEY){
            btselfn(data,near_word,mean, 10000, &value);
            if (isPrefix(text,near_word) == 1){
                gtk_entry_set_text(GTK_ENTRY(entryword),near_word);
                gtk_editable_set_position(GTK_EDITABLE(entryword),strlen(near_word));
            }
            else return TRUE;
        }
        display(entryword,NULL);

        return TRUE;
    }
    else{

        //count=0;
        // Suggestion search

        strcpy(text,gtk_entry_get_text((GtkEntry*) entryword));

        if (key->keyval!=GDK_KEY_BackSpace){
            len=strlen(text);
            text[len]=key->keyval;
            text[len+1]='\0';
        }
        gtk_list_store_clear(liststore);
        if (btsel(data,text,mean, 10000, &value) == 0) {
```

```

    gtk_list_store_append(liststore,&iter);
    gtk_list_store_set(liststore,&iter,0,text,-1);
}
btseln(data,near_word,mean, 10000, &value);
while ((isPrefix(text,near_word) == 1) && (count < 8)){
    gtk_list_store_append(liststore,&iter);
    gtk_list_store_set(liststore,&iter,0,near_word,-1);
    btseln(data,near_word,mean, 10000, &value);
    count++;
}

t_end = clock();
g_print ("Thoi gian ham searchword: %lf\n", (double)(t_end -
t_begin)/CLOCKS_PER_SEC);
return FALSE;
}
}

```

5.4 Hàm *add_word*

- Chức năng: Thêm từ vào trong từ điển và cho phép cập nhật nghĩa nếu từ đó đã có trong từ điển.

- CODE :

```
gboolean add_word(GtkWidget *dialog,gpointer entry){
    GtkWidget *dialog1;
    GtkWidget *word,*mean;
    GtkWidget *add,*cancel;
    GtkWidget *word_entry,*textView;
    GtkTextBuffer *buffer;

    dialog=gtk_dialog_new_with_buttons("Thêm
từ",mainwindow,GTK_DIALOG_MODAL,GTK_STOCK_ADD,1,GTK_STOCK_C
ANCEL,2,NULL);
    gtk_window_set_title(GTK_WINDOW(dialog),"Thêm từ");
    gtk_window_set_default_size(GTK_WINDOW(dialog), 500, 300);
    gtk_dialog_set_default_response(GTK_DIALOG(dialog),1);
    word=gtk_label_new("Từ");
    mean=gtk_label_new("Nghĩa");
    word_entry=gtk_entry_new();

    GtkWidget* scrolledWindow=gtk_scrolled_window_new(NULL,NULL);
    gtk_container_set_border_width(GTK_CONTAINER (scrolledWindow),10);

    gtk_scrolled_window_set_policy(GTK_SCROLLED_WINDOW(scrolledWindow),
GTK_POLICY_AUTOMATIC,GTK_POLICY_AUTOMATIC);
    textView=gtk_text_view_new();

    gtk_text_view_set_wrap_mode(GTK_TEXT_VIEW(textView),GTK_WRAP_WOR
D);
    buffer=gtk_text_view_get_buffer(GTK_TEXT_VIEW(textView));
    gtk_container_add(GTK_CONTAINER(scrolledWindow),textView);
    gtk_widget_set_size_request (scrolledWindow, 200, 200);

    GtkWidget *table;
    table=gtk_table_new(5,5,FALSE);
    gtk_table_attach_defaults(GTK_TABLE(table),word,0,1,0,1);
    gtk_table_attach_defaults(GTK_TABLE(table),mean,0,1,1,2);
    gtk_table_attach_defaults(GTK_TABLE(table),word_entry,1,4,0,1);
    gtk_table_attach_defaults(GTK_TABLE(table),scrolledWindow,1,4,1,4);
    gtk_table_set_row_spacings (GTK_TABLE (table),10);
    gtk_table_set_col_spacings (GTK_TABLE (table),3);
```

```

gtk_container_set_border_width(GTK_CONTAINER(textView),1);
gtk_container_set_border_width(GTK_CONTAINER(scrolledWindow),1);
gtk_container_set_border_width (GTK_CONTAINER (table),2);
gtk_box_pack_start (GTK_BOX (gtk_dialog_get_content_area (GTK_DIALOG
(dialog))), table, 0, 0, 0);

```

```

gtk_widget_show_all(dialog);

```

```

GtkTextIter start,end;
char *s = (char*)malloc (200), *meaning = (char*)malloc (10000);
int k;
BTint value;
int result=gtk_dialog_run(GTK_DIALOG(dialog));

```

//Thoi gian thuc hien ham

```

t_begin = clock();

```

```

if (result==1) {
    gtk_text_buffer_get_bounds(buffer,&start,&end);
    strcpy(s, gtk_entry_get_text((GtkEntry*)word_entry));
    strcpy(meaning, gtk_text_buffer_get_text(buffer, &start, &end, FALSE));
    if (strlen(s)==0){

```

```

dialog1=gtk_message_dialog_new(GTK_WINDOW(dialog),GTK_DIALOG_DEST
ROY_WITH_PARENT,GTK_MESSAGE_WARNING,GTK_BUTTONS_OK,"Bạn
chưa nhập từ!");
    gtk_window_set_title(GTK_WINDOW(dialog1),"Lỗi!");
    gtk_dialog_run(GTK_DIALOG(dialog1));
    }
    else if (bfndky(data,s,&value)!=QNOKEY){

```

```

dialog1=gtk_message_dialog_new(GTK_WINDOW(dialog),GTK_DIALOG_DEST
ROY_WITH_PARENT,GTK_MESSAGE_QUESTION,GTK_BUTTONS_YES_NO
,"Từ đã có trong cơ sở dữ liệu!Bạn muốn cập nhập nghĩa của từ này?");
    gtk_window_set_title(GTK_WINDOW(dialog1),"Trùng từ");
    if (gtk_dialog_run(GTK_DIALOG(dialog1))==GTK_RESPONSE_YES)
        btupd(data,s,meaning,strlen(meaning)+1);
    }
    else{

```

```

dialog1=gtk_message_dialog_new(GTK_WINDOW(dialog),GTK_DIALOG_DEST
ROY_WITH_PARENT,GTK_MESSAGE_QUESTION,GTK_BUTTONS_YES_NO
,"Bạn muốn thêm từ này vào cơ sở dữ liệu?");
    gtk_window_set_title(GTK_WINDOW(dialog1),"Xác nhận");

```

```
    if (gtk_dialog_run(GTK_DIALOG(dialog1))==GTK_RESPONSE_YES) {  
        btins(data,s ,meaning,strlen(meaning)+1);  
    }  
}  
}  
t_end = clock();  
g_print ("Thoi gian ham add_word: %lf\n", (double)(t_end -  
t_begin)/CLOCKS_PER_SEC);  
free (s);  
free (meaning);  
gtk_widget_destroy(dialog);  
}
```


5.5 Hàm *delete_word*

- Chức năng: Xóa từ

- CODE:

```
gboolean delete_word(GtkWidget *widget, gpointer entry)
{
    GtkWidget *dialog,*window=mainwindow;
    char word[50];
    BTint value;

    //Tính thời gian

    t_begin = clock();

    strcpy(word,(char*)gtk_entry_get_text(GTK_ENTRY(entry)));
    if (strlen(word) == 0)
    {

        dialog=gtk_message_dialog_new(GTK_WINDOW(window),GTK_DIALOG_DESTROY_WITH_PARENT,GTK_MESSAGE_WARNING,GTK_BUTTONS_OK,"Bạn chưa nhập từ!");
        gtk_window_set_title(GTK_WINDOW(dialog),"Lỗi!");
        gtk_dialog_run(GTK_DIALOG(dialog));
    }
    else if (bfndky(data,word,&value)==0)
    {

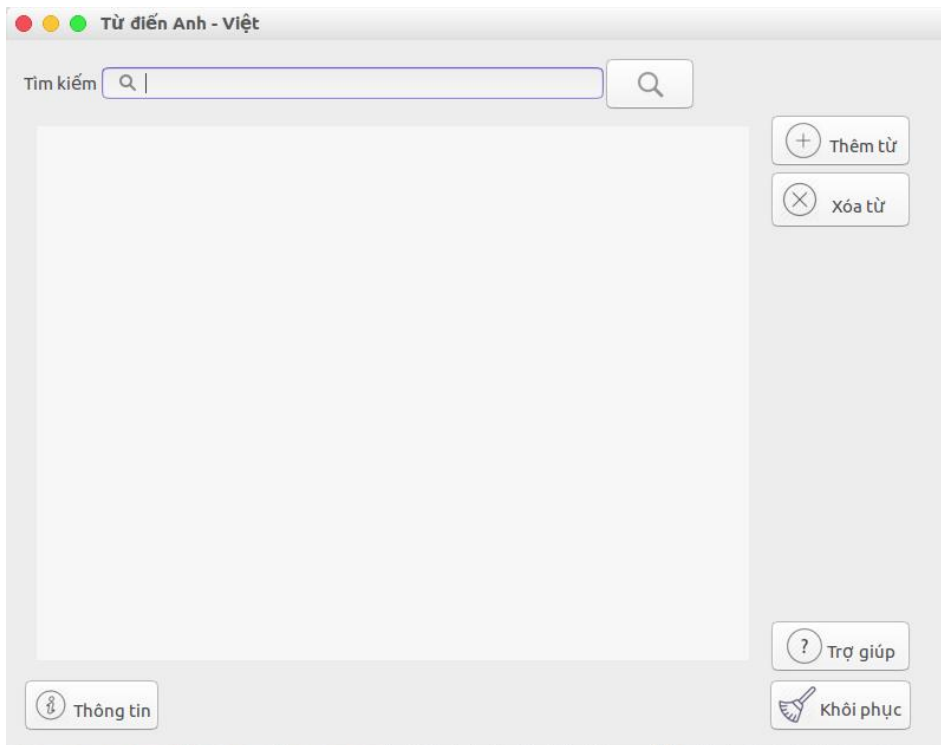
        dialog=gtk_message_dialog_new(GTK_WINDOW(window),GTK_DIALOG_DESTROY_WITH_PARENT,GTK_MESSAGE_QUESTION,GTK_BUTTONS_YES_NO,"Bạn muốn xóa từ này khỏi cơ sở dữ liệu?");
        gtk_window_set_title(GTK_WINDOW(dialog),"Xác nhận");
        if (gtk_dialog_run(GTK_DIALOG(dialog))==GTK_RESPONSE_YES)
        {
            btdel(data, word);
            gtk_text_buffer_set_text(buffer,"",-1);
            gtk_entry_set_text(GTK_ENTRY(entry),"");
        }
    }
    else
    {

        dialog=gtk_message_dialog_new(GTK_WINDOW(window),GTK_DIALOG_DESTROY_WITH_PARENT,GTK_MESSAGE_WARNING,GTK_BUTTONS_OK,"Từ không có trong cơ sở dữ liệu!");
        gtk_window_set_title(GTK_WINDOW(dialog),"Lỗi!");
```

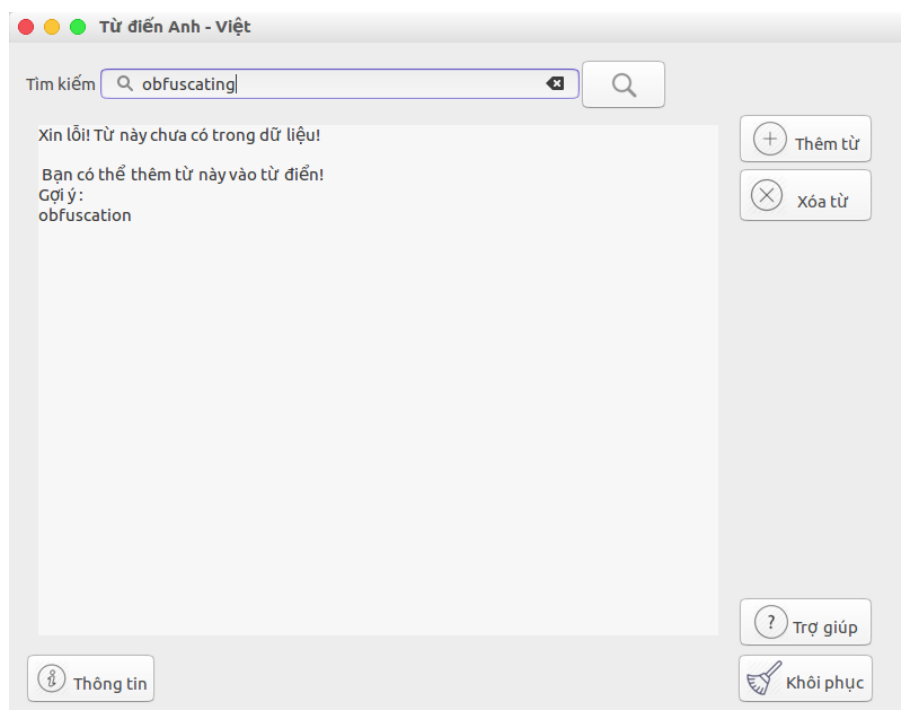
```
    gtk_dialog_run(GTK_DIALOG(dialog));  
}  
  
t_end = clock();  
g_print ("Thoi gian ham delete_word: %lf\n", (double)(t_end -  
t_begin)/CLOCKS_PER_SEC);  
  
gtk_widget_destroy(dialog);  
return FALSE;  
}
```

6. Kết quả và đánh giá thời gian

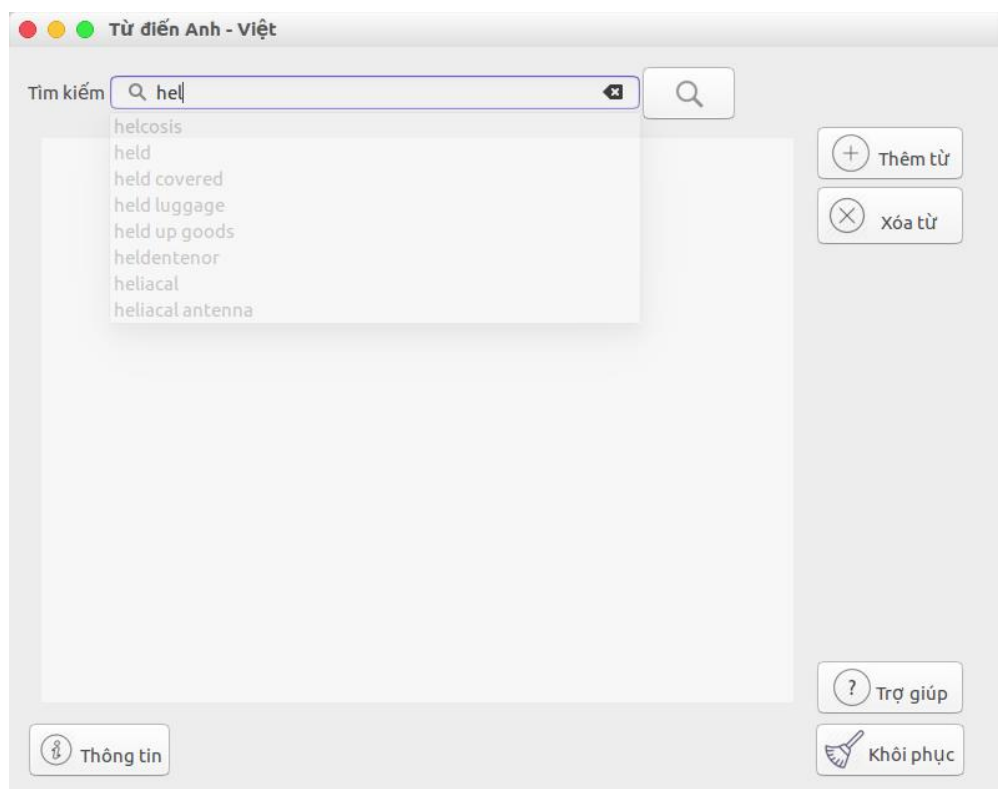
- Màn hình chính



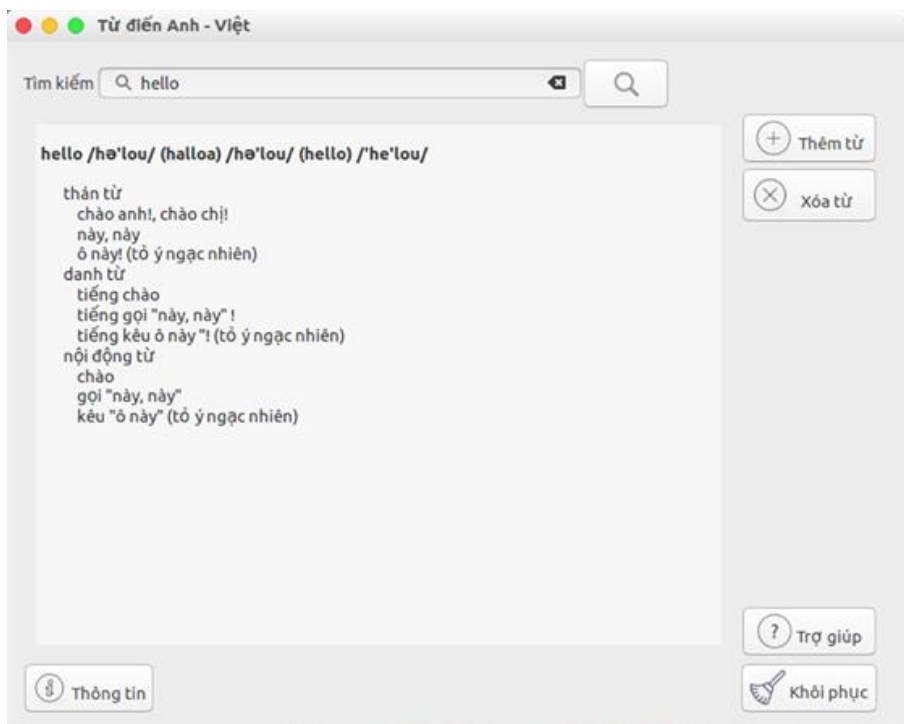
- Chức năng suggestion_word :



- Chức năng suggestion_search:

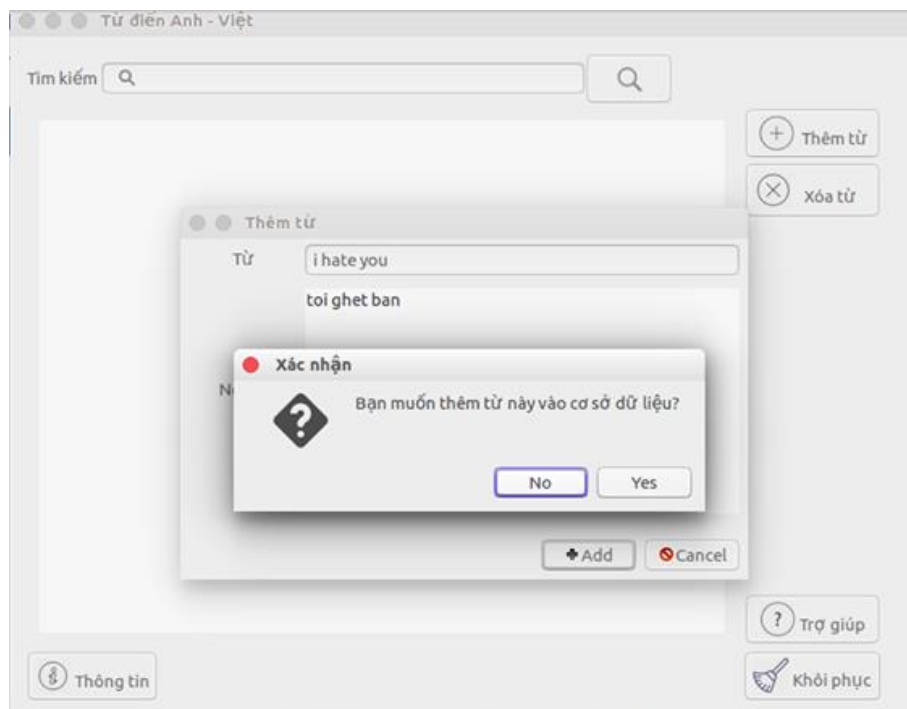


- Chức năng search:



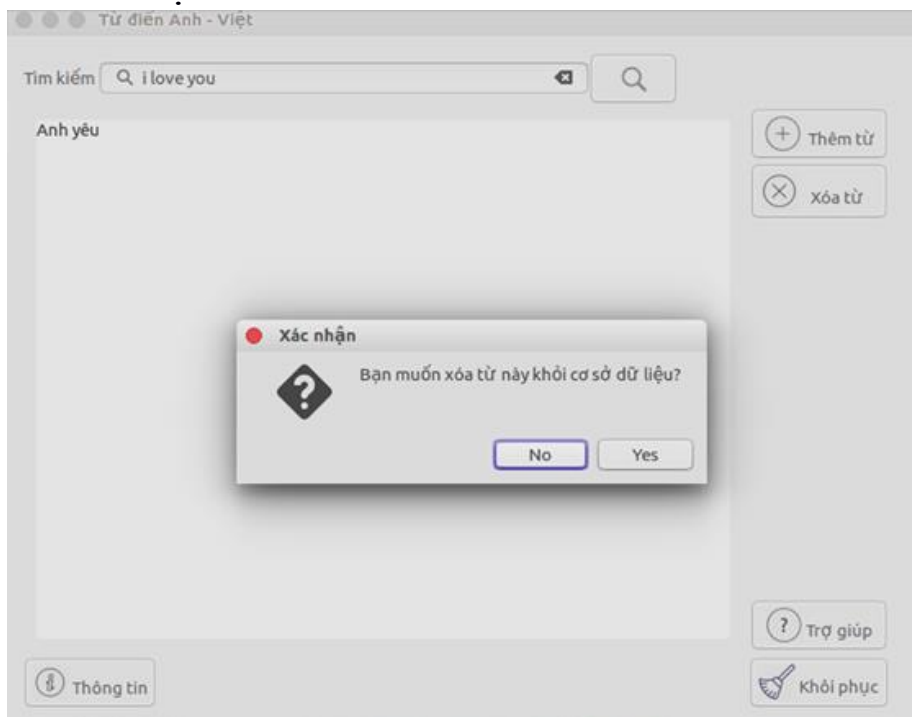
- Chức năng add word:

+Giao diện:

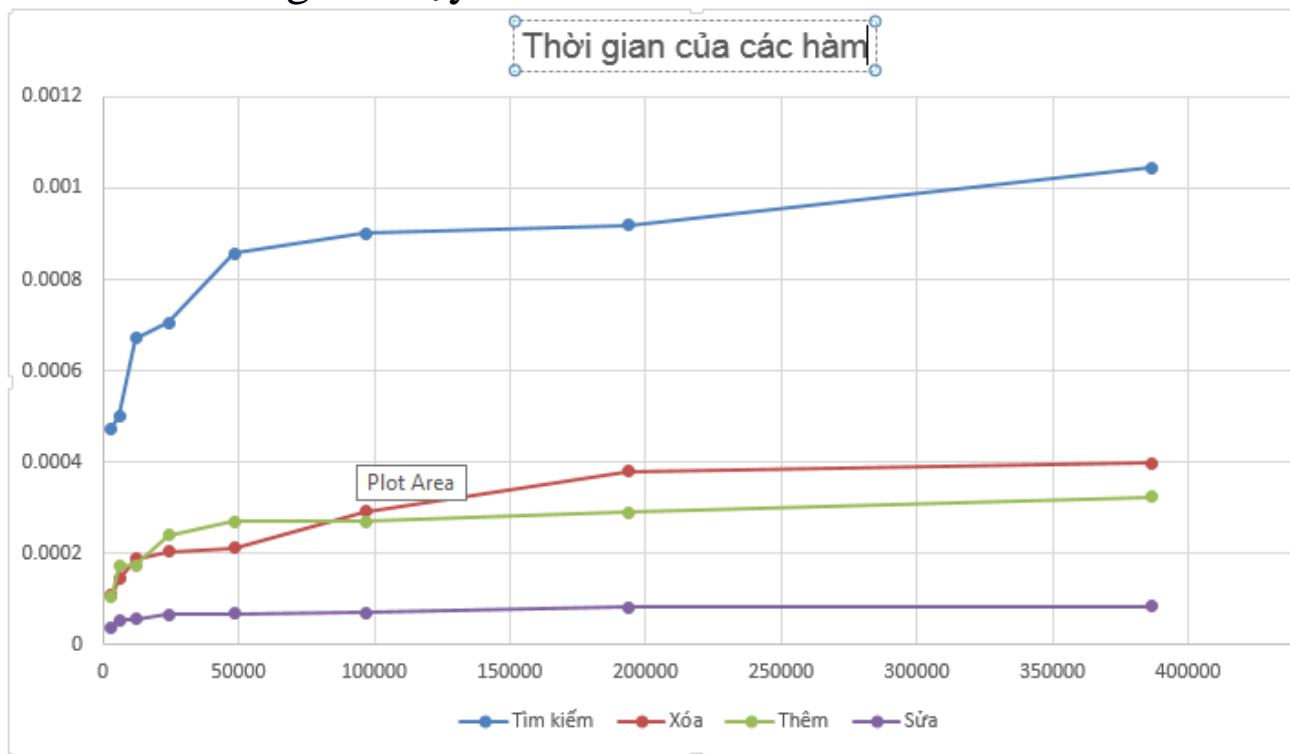


- Chức năng delete word:

+ Giao diện:



- Biểu đồ thời gian chạy:



7. Nhận xét và kết luận

+ Ưu điểm:

- Phần mềm có đầy đủ chức năng như yêu cầu
- Giao diện đơn giản dễ sử dụng

+ Nhược điểm:

- Thời gian thực hiện chưa tối

+ Hướng phát triển trong tương lai:

- Bổ sung thêm cơ sở dữ liệu
- Thêm chức năng phát
- Giao diện đẹp hơn