

# BÁO CÁO BÀI TẬP XÂY DỰNG ỨNG DỤNG TỪ ĐIỂN CÓ GIAO DIỆN NGƯỜI DÙNG

---

Sinh viên: Trần Đình Tôn Hiếu  
Mã số sinh viên: 1712441

## Mục lục

### 1. Cấu trúc chương trình

- 1.1. Framework sử dụng và cấu trúc dữ liệu sử dụng.....
- 1.2. Các tập tin có trong chương trình.....

### 2. Giao diện chương trình

- 2.1. Giao diện chức năng tra cứu từ.....
- 2.2. Giao diện chức năng chỉnh sửa từ.....
- 2.3. Giao diện chức năng thêm từ.....
- 2.4. Giao diện chức năng xóa từ.....
- 2.5. Lưu ý.....

# 1. Cấu trúc chương trình:

## 1.1. Framework sử dụng và cấu trúc dữ liệu sử dụng:

### 1.1.1. Framework sử dụng:

- Để làm ứng dụng từ điển có giao diện người dùng, em đã sử dụng một Application Framework khá nổi tiếng là Qt.



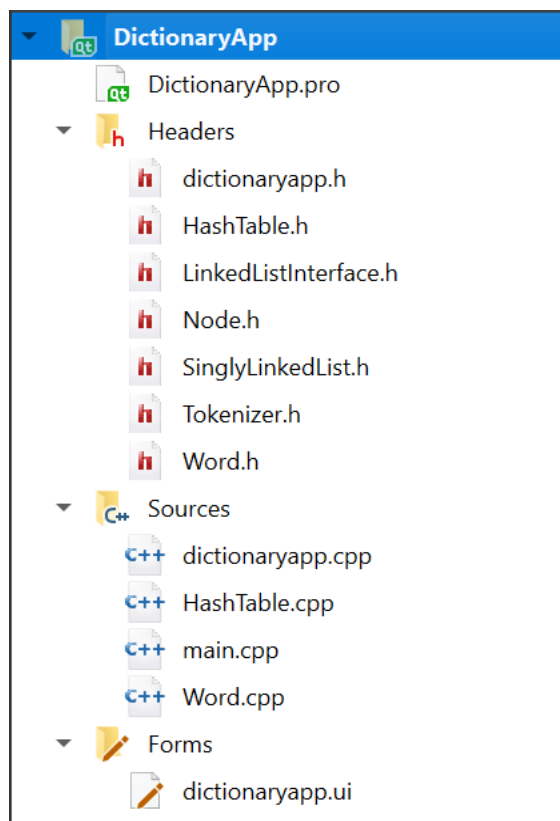
- Mục tiêu của các nhà phát triển nên Qt chính là tạo ra một framework có khả năng thiết kế những phần mềm có thể chạy trên nhiều nền tảng phần mềm lẫn phần cứng khác nhau mà không phải thay đổi nhiều về code.

### 1.1.2. Cấu trúc dữ liệu sử dụng:

- Sau 4 tuần vừa rồi với thiết kế chương trình từ điển bằng 3 cấu trúc dữ liệu khác nhau, em chọn cấu trúc dữ liệu Bảng băm cho ứng dụng từ điển hoàn chỉnh của mình.

- Lý do chủ yếu nhất để chọn bảng băm là tính đơn giản trong mã nguồn của nó. Việc cài đặt bảng băm rất đơn giản mà mang lại hiệu rất tốt.

## 1.2. Các tập tin có trong chương trình:



- Cấu trúc chương trình gồm những file như hình.

- Các file .h và .cpp là các file mã nguồn.

- File DictionaryApp.pro là file chứa các định nghĩa như HEADER, SOURCES là gồm các file nào.

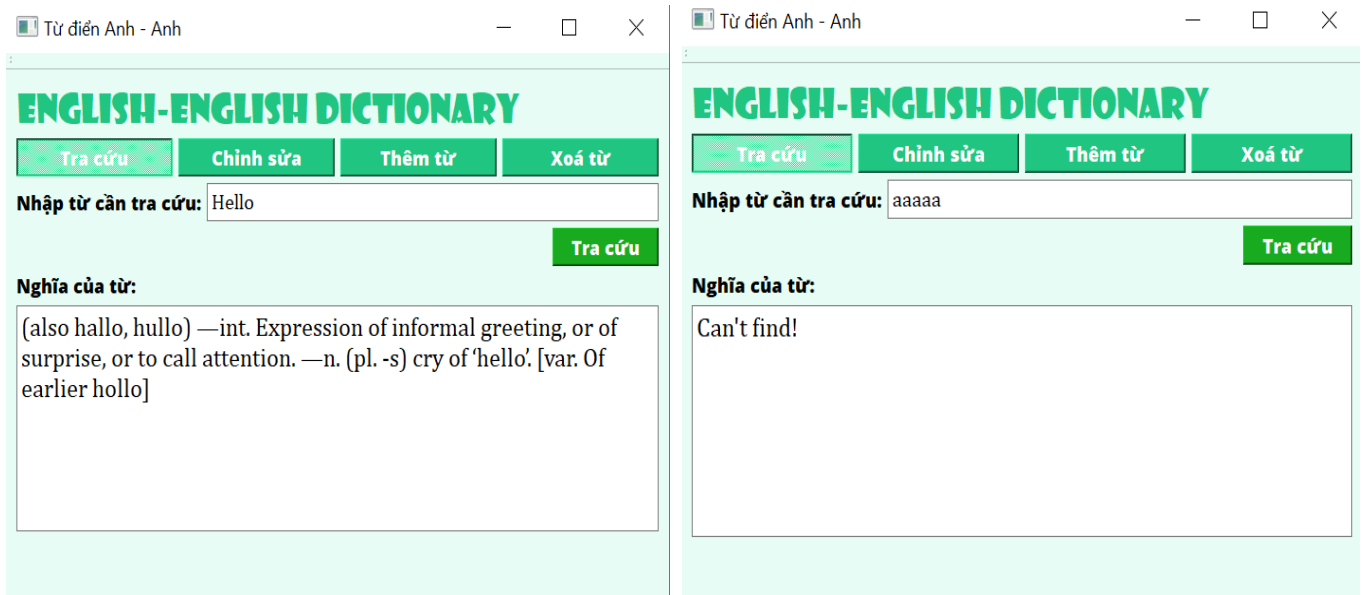
- File dictionaryapp.ui là file giao diện của chương trình.

- dictionaryapp.h và dictionaryapp.cpp dùng để định nghĩa lớp DictionaryApp, là lớp giao diện chính của chương trình. Việc của hàm main chỉ là gọi lớp này. Mọi event của chương trình đều được khai báo và cài đặt trong lớp này.

## 2. Giao diện chương trình:

Giao diện của chương trình rất đơn giản, bao gồm 4 cửa sổ làm việc: Tra cứu, chỉnh sửa, thêm từ, xoá từ.

### 2.1. Giao diện tra cứu từ:

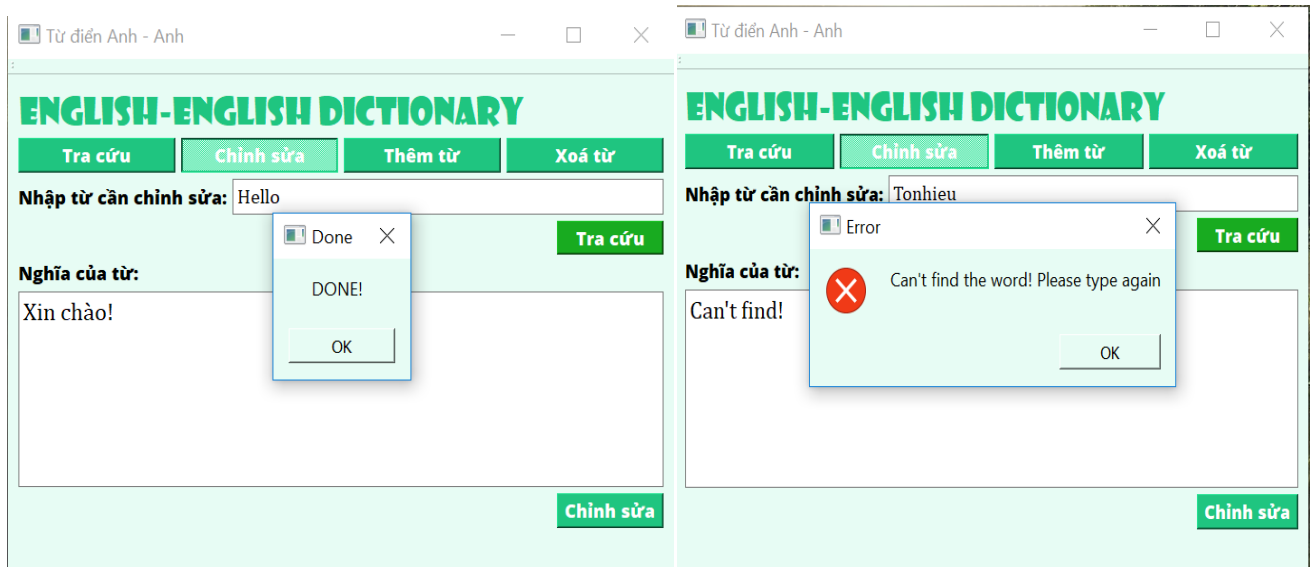


- Khi mở chương trình thì mặc định tab được chọn sẽ là “Tra cứu” và có giao diện như trên. Vì giao diện đã quá đơn giản và rõ ràng nên sẽ không cần thiết phải có hướng dẫn sử dụng. Theo em thì, nếu một User interface mà cần người dùng phải đọc hướng dẫn sử dụng thì coi như UI đó đã thất bại. Nếu như tìm kiếm từ không thành công thì sẽ hiển thị như hình bên phải.

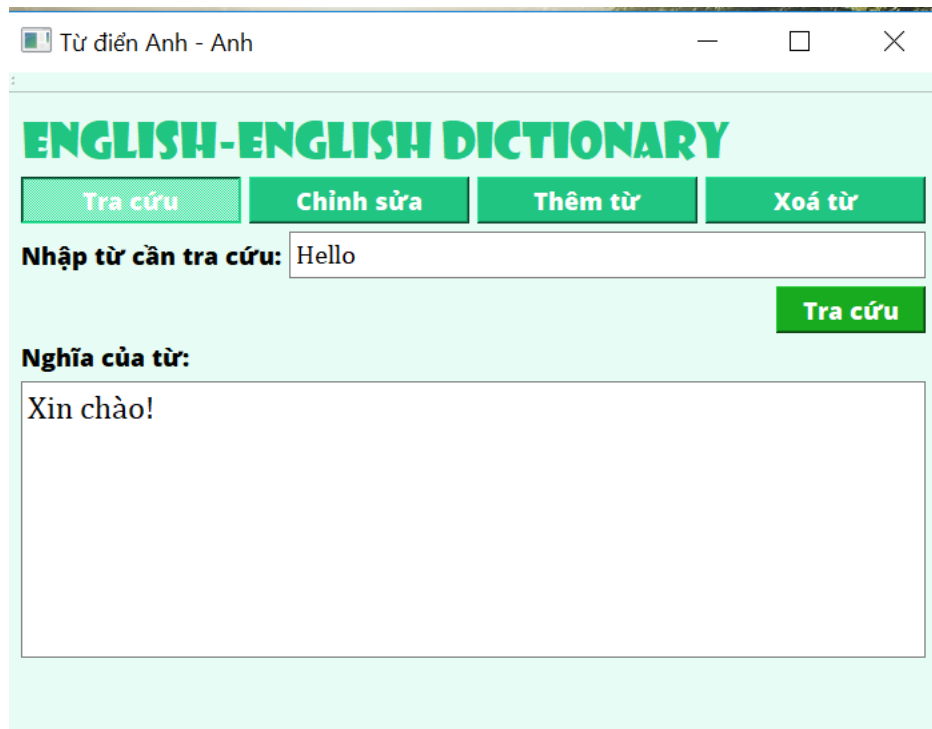
- Trong class DictionaryApp sẽ có thêm một thuộc tính là HashTable \*hashTable, và được khởi tạo cùng lúc với DictionaryApp. Vì vậy, mọi event của chương trình có trong DictionaryApp đều có thể gọi được hashTable thực hiện.

- Thời gian tải ứng dụng lên hay thời gian tìm kiếm một từ gần như bằng 0, ta không hề phải chờ đợi khi sử dụng ứng dụng.

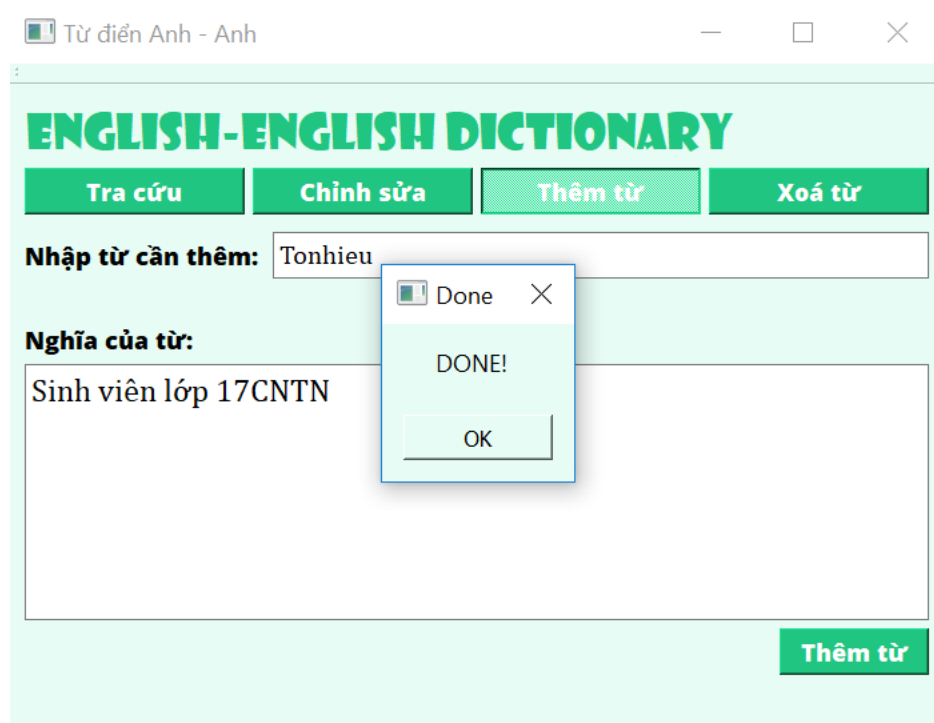
## 2.2. Giao diện chỉnh sửa từ:



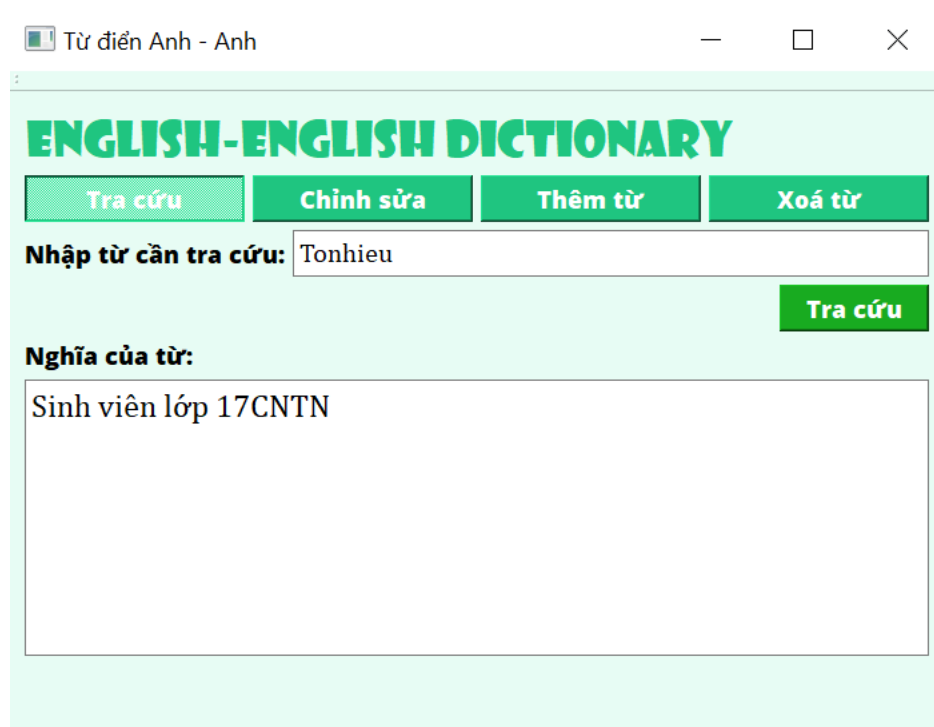
- Giao diện tab chỉnh sửa từ sẽ như trên. Hình ở bên trái là đối với chỉnh sửa một từ đã có trong từ điển. Hình ở bên phải là đối với từ không có trong từ điển, ta sẽ nhận được thông báo như trên.
- Sau khi chỉnh sửa như ở hình đầu tiên, khi tra cứu lại, ta sẽ thu được kết quả như sau:



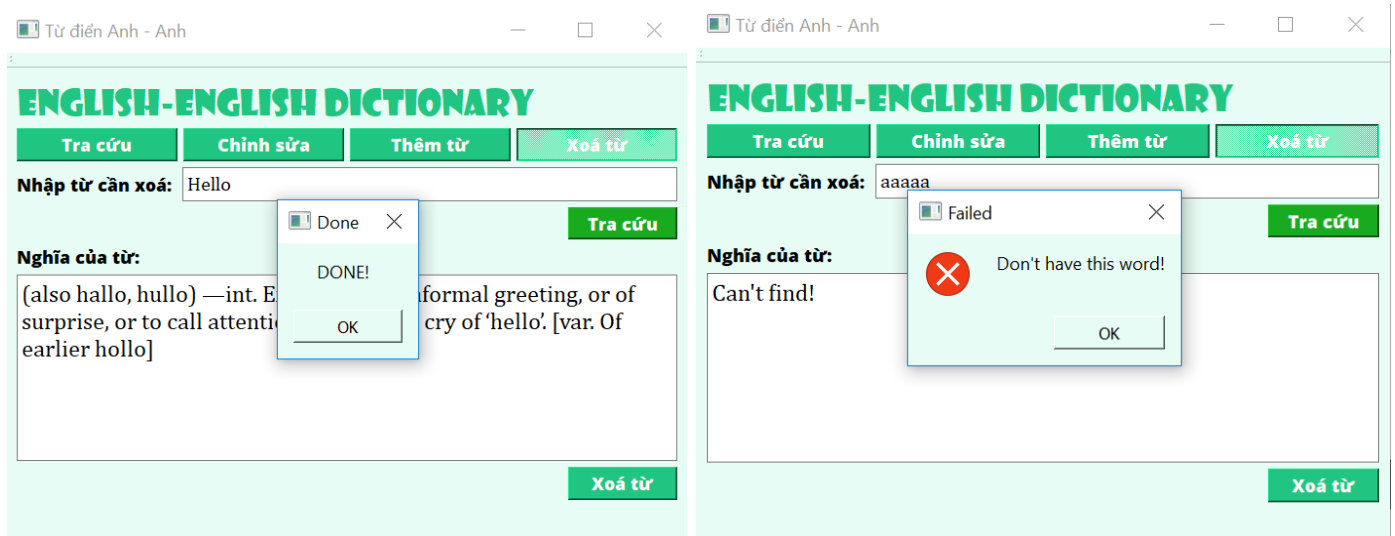
## 2.3. Giao diện thêm một từ:



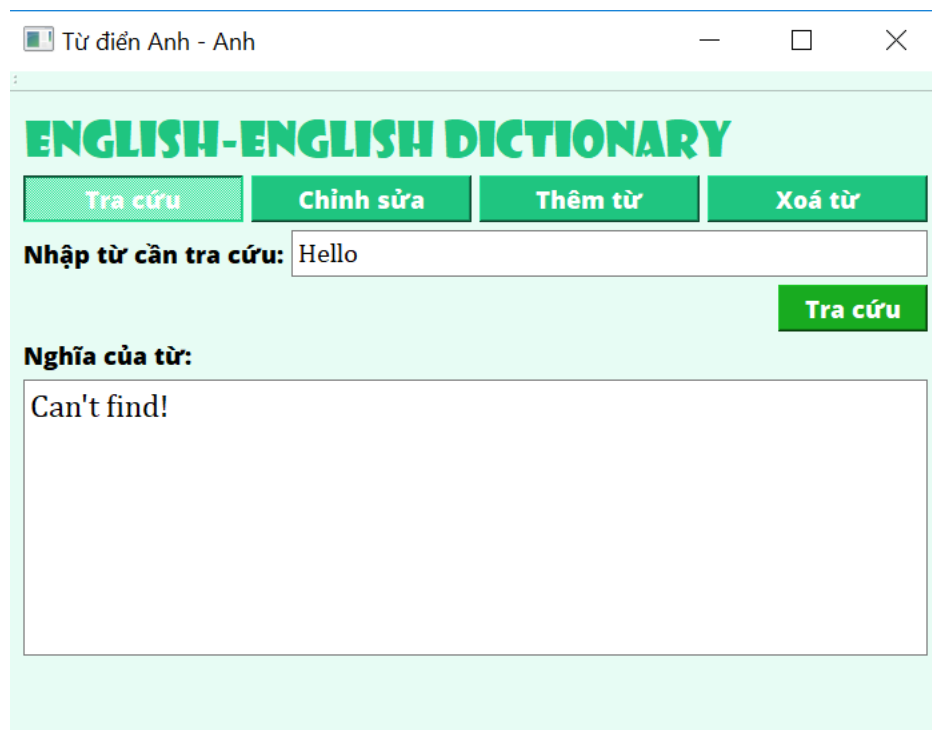
- Sau khi thêm như hình trên, ta tra cứu sẽ có kết quả như sau:



## 2.4. Giao diện xoá một từ:



- Giao diện xoá từ sẽ như hình trên. Ở hình bên trái là xoá từ thành công đối với một từ có trong từ điển. Hình bên phải là xoá từ không thành công vì từ đó không có trong từ điển. Sau khi thực hiện thao tác xoá như hình bên trái, ta tra cứu lại từ Hello sẽ có kết quả như sau:



## 2.5. Lưu ý:

- Chương trình sẽ đọc dữ liệu từ file DictionaryFile.txt nằm cùng thư mục với file .exe như hình sau:

 DictionaryApp	11/26/2018 9:51 PM	Application
 DictionaryFile	11/21/2018 8:35 PM	Text Document

- Nếu như không có file DictionaryFile.txt nằm cùng thư mục thì sẽ báo lỗi:

