TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN LẬP TRÌNH TRỰC QUAN



ĐỀ TÀI:

XÂY DỰNG CHƯƠNG TRÌNH TỪ ĐIỂN ANH – VIỆT CÓ HÌNH ẢNH VÀ ÂM THANH

**TÊN NHÓM : LAST GROUP**

SINH VIÊN 1 : NGUYỄN MẠNH TÙNG

MSSV: 17521236

SINH VIÊN 2 : NGUYỄN LƯƠNG DUY

MSSV: 17520004

**LỜI MỞ ĐẦU**

Sự phát triển như vũ bão của ngành công nghệ thông tin kéo theo đó là các dịch vụ mạng ngày càng phát triển, mở rộng và hoàn thiện hơn, tuy vẫn tồn tại nhiều khuyết điểm song không ít tiện lợi từ công nghệ đem lại cho xã hội loài người sự nhanh chóng và chính xác... Thư điện tử dần thay thế thư tay, các cuộc hội nghị truyền hình với âm thanh hình ảnh trực tiếp giúp linh hoạt hóa công việc của tổ chức, đoàn thể. Người bác sĩ không phải trực tiếp đứng tại phòng mổ vẫn có thể thực hiện ca phẫu thuật với sự chính xác cao, và độ an toàn tuyệt đối… còn rất nhiều lợi ích mà công nghệ thông tin đem lại cho chúng ta, vậy dựa vào đâu mà công nghệ có thể thực hiện được những điều tưởng chừng không thể trong một tích tắc như vậy?

Ngôn ngữ lập trình là một phần không thể thiếu trong việc xây dựng nên một thế giới công nghệ linh hoạt và mạnh mẽ. Không gian làm việc Microsoft .Net tổng hợp bởi bốn bộ ngôn ngữ lập trình: C#, VB.NET, Managed C++, and J# .NET. ở đó có sự chồng gối lên nhau của các ngôn ngữ, và được định nghĩa trong FCL (framework class library). Hỗ trợ cho lập trình viên phát triển các ứng dụng mạng với kích thước nhẹ và mạnh mẽ trong xử lý.

Dựa trên kiến thức lập trình mạng với C#, sự đa dạng của các dịch vụ mạng với nhiều tính năng và đòi hỏi ngày càng cao. Từ ý tưởng áp dụng công nghệ thông tin vào việc quản lý, bớt phần gánh nặng về sổ sách cũng như đầu óc con người, đồ án này hướng đến xây dựng một phần mềm quản lý bán hàng máy tính, áp dụng cho các công ty hay cửa hàng máy tính.Giúp các công ty có thể dễ dàng hơn trong công việc quản lý hàng hóa của công ty hay cửa hàng mình.

Trong quá trình xây dựng chương trình không tránh khỏi sai sót kính mong các thầy cô và các bạn đóng góp ý kiến xây dựng để chương trình có thể hoàn thiện hơn.

# CHƯƠNG 1: TỔNG QUAN

# 1. MỤC TIÊU VÀ NỘI DUNG ĐỀ TÀI:

# 1. MỤC TIÊU – HƯỚNG GIẢI QUYẾT VÀ KẾ HOẠCH THỰC HIỆN

# 1.1 Mục tiêu cần đạt được:

Chúng ta cần phải thiết kế và cài đặt một chương trình quản lý có các chức năng sau:

1. Tìm kiếm với từ khóa ( tra cứu thủ công )
2. Tra cứu hình ảnh
3. Âm thanh

## HƯỚNG GIẢI QUYẾT :

### Hướng giải quyết:

* Sử dụng combobox để điền key và so sánh với csdl ( file .xml)
* Thiết kế ứng dụng: các chức năng, Forms .
* Cài đặt chương trình ứng dụng, nhập liệu, chạy thử và kiểm tra lỗi.
* Viết một bài báo cáo về công việc đã thực hiện theo mẫu qui định.
* **MÔI TRƯỜNG CÀI ĐẶT:**

Để thực hiện được yêu cầu phải cần có một ngôn ngữ lập trình có thể kết nối đến cơ sở dữ liệu và một hệ quản trị cơ sở dữ liệu.

* Ngôn ngữ lập trình C#: Visual Studio 2017.
* Hệ quản trị cơ sở dữ liệu: XML database

### Kế hoạch thực hiện:

Thời gian thực hiện chương trình là 12 tuần với kế hoạch như sau:

|  |  |
| --- | --- |
| **Thời gian** | **Công việc** |
| Tuần 1 | Tìm hiểu đề tài |
| Tuần 2-3 | Lên ý tưởng các chức năng chính |
| Tuần 4 | Phân chia công việc |
| Tuần 5-10 | Viết và kiểm thử chương trình |
| Tuần 11 | Viết báo cáo |
| Tuần 12 | Hiệu chỉnh và hoàn tất |

# CHƯƠNG 2: NGÔN NGỮ LẬP TRÌNH – CƠ SỞ DỮ LIỆU

### Giới thiệu về Microsoft .NET và Kiến trúc .NET

**1.1 Giới thiệu về Microsoft .NET**

Microsoft .Net không phải là một ngôn ngữ lập trình, đó là một không gian làm việc tổng hợp bởi bốn bộ ngôn ngữ lập trình: C#, VB.NET, Managed C++, and J# .NET. ở đó có sự chồng gối lên nhau của các ngôn ngữ, và được định nghĩa trong FCL (framework class library).

Microsoft .Net bao gồm 2 phần chính: Framework và Intergrated Development Enviroment (IDE). Framework cung cấp những gì cần thiết và căn bản, là khuôn dạng hay môi trường hỗ trợ các hạ tầng cơ sở theo một quy ước nhất định để công việc được thuận tiện. IDE cung cấp một môi trường giúp chúng ta triển khai dễ dàng và được nhanh chóng các ứng dụng dựa trên nền tảng .Net.

Thành phần Framework là quan trọng nhất .NET là cốt lõi và tinh hoa của môi trường, còn IDE chỉ là công cụ để phát triển dựa trên nền tảng đó thôi. Trong .NET toàn bộ các ngôn ngữ C#, Visual C++ hay Visual Basic.NET đều dùng cùng một IDE.

Microsoft .NET là nền tảng cho việc xây dựng và thực thi các ứng dụng phân tán thế hệ kế tiếp. Bao gồm các ứng dụng từ client đến server và các dịch vụ khác. Một số tính năng của Microsoft .NET cho phép những nhà phát triển sử dụng như sau:

* Một mô hình lập trình cho phép nhà phát triển xây dựng các ứng dụng dịch vụ web và

ứng dụng client với Extensible Markup Language (XML).

* Tập hợp dịch vụ XML Web, như Microsoft .NET My Services cho phép nhà phát

triển đơn giản và tích hợp người dùng kinh nghiệm.

* Cung cấp các server phục vụ bao gồm: Windows 2000, SQL Server, và BizTalk Server, tất cả điều tích hợp, hoạt động, và quản lý các dịch vụ XML Web và các ứng dụng.
* Các phần mềm client như Windows XP và Windows CE giúp người phát triển phân

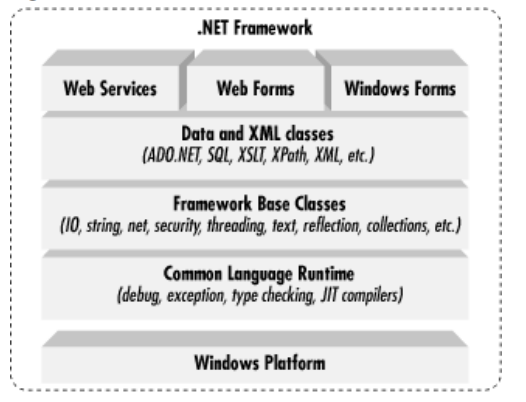
phối sâu và thuyết phục người dùng kinh nghiệm thông qua các dòng thiết bị.

* Nhiều công cụ hỗ trợ như Visual Studio .NET, để phát triển các dịch vụ Web XML,ứng dụng trên nền Windows hay nền web một cách dể dàng và hiệu quả.

### 

### 1.2 Kiến trúc .Net

Mô hình kiến trúc của .Net:



*Hình 1: Mô hình kiến trúc của .NET*

.NET Framework là một platform mới làm đơn giản việc phát triển ứng dụng trong môi trường phân tán của Internet. .NET Framework được thiết kế đầy đủ để đáp ứng theo quan điểm sau:

Để cung cấp một môi trường lập trình hướng đối tượng vững chắc, trong đó mã nguồn đối tượng được lưu trữ và thực thi một cách cục bộ. Thực thi cục bộ nhưng được phân tán trên Internet, hoặc thực thi từ xa.

Để cung cấp một môi trường thực thi mã nguồn mà tối thiểu được việc đóng gói phần mềm và sự tranh chấp về phiên bản.

Để cung cấp một môi trường thực thi mã nguồn mà đảm bảo việc thực thi an toàn mã nguồn, bao gồm cả việc mã nguồn được tạo bởi hãng thứ ba hay bất cứ hãng nào mà tuân thủ theo kiến trúc .NET.

Để cung cấp một môi trường thực thi mã nguồn mà loại bỏ được những lỗi thực hiện các script hay môi trường thông dịch.

Để làm cho những người phát triển có kinh nghiệm vững chắc có thể nắm vững nhiều kiểu ứng dụng khác nhau. Như là từ những ứng dụng trên nền Windows đến những ứng dụng dựa trên web.

Để xây dựng tất cả các thông tin dựa triên tiêu chuẩn công nghiệp để đảm bảo rằng mã nguồn trên .NET có thể tích hợp với bất cứ mã nguồn khác.

.NET Framework có hai thành phần chính: Common Language Runtime (CLR) và thư viện lớp .NET Framework. CLR là nền tảng của .NET Framework.

Phát triển ứng dụng Client

Những ứng dụng client cũng gần với những ứng dụng kiểu truyền thống được lập trình dựa trên Windows. Đây là những kiểu ứng dụng hiển thị những cửa sổ hay những form trên desktop cho phép người dùng thực hiện một thao tác hay nhiệm vụ nào đó. Những ứng dụng client bao gồm những ứng dụng như xử lý văn bản, xử lý bảng tính, những ứng dụng trong lĩnh vực thương mại như công cụ nhập liệu, công cụ tạo báo cáo...Những ứng dụng client này thường sử dụng những cửa sổ, menu, toolbar, button hay các thành phần GUI khác, và chúng thường truy cập các tài nguyên cục bộ như là các tập tin hệ thống, các thiết bị ngoại vi như máy in.

Một loại ứng dụng client khác với ứng dụng truyền thống như trên là ActiveX control (hiện nay nó được thay thế bởi các Windows Form control) được nhúng vào các trang web trên Internet. Các ứng dụng này cũng giống như những ứng dụng client khác là có thể truy cập tài nguyên cục bộ.

Những lớp .NET Framework chứa trong .NET Framework được thiết kế cho việc sử dụng phát triển các GUI. Điều này cho phép người phát triển nhanh chóng và dễ dàng tạo các cửa sổ, button, menu, toolbar, và các thành phần khác trong các ứng dụng được viết phục vụ cho lĩnh vực thương mại.

### 

### 2. Ngôn ngữ lập trình C#

Ngôn ngữ C# khá đơn giản, chỉ khoảng 80 từ khóa và hơn mười mấy kiểu dữ liệu được xây dựng sẵn. Tuy nhiên, ngôn ngữ C# có ý nghĩa cao khi nó thực thi những khái niệm lập trình hiện đại. C# bao gồm tất cả những hỗ trợ cho cấu trúc, thành phần component, lập trình hướng đối tượng.

Phần cốt lõi hay còn gọi là trái tim của bất cứ ngôn ngữ lập trình hướng đối tượng là sự hỗ trợ của nó cho việc định nghĩa và làm việc với những lớp. Những lớp thì định nghĩa những kiểu dữ liệu mới, cho phép người phát triển mở rộng ngôn ngữ để tạo mô hình tốt hơn để giải quyết vấn đề. Ngôn ngữ C# chứa những từ khóa cho việc khai báo những kiểu lớp đối tượng mới và những phương thức hay thuộc tính của lớp, và cho việc thực thi đóng gói, kế thừa, và đa hình, ba thuộc tính cơ bản của bất cứ ngôn ngữ lập trình hướng đối tượng.

Trong ngôn ngữ C# mọi thứ liên quan đến khai báo lớp điều được tìm thấy trong phần khai báo của nó. Định nghĩa một lớp trong ngôn ngữ C# không đoi hỏi phải chia ra tập tin header và tập tin nguồn giống như trong ngôn ngữ C++. Hơn thế nữa, ngôn ngữ C# hỗ trợ kiểu XML, cho phép chèn các tag XML để phát sinh tự động các document cho lớp.

Ngôn ngữ C# cung cấp những đặc tính hướng thành phần (component-oriented), như là những thuộc tính, những sự kiện. Lập trình hướng thành phần được hỗ trợ bởi CLR cho phép lưu trữ metadata với mã nguồn cho một lớp.

# 3. XML DATABASE

Ngày nay, việc ứng dụng công nghệ thông tin vào đời sống được phát triển vô cùng mạnh mẽ. Ngành công nghệ thông tin nói chung và lĩnh vực công nghệ mạng nói riêng đã cho ra đời nhiều ứng dụng, dịch vụ đặc biệt kể đến là dịch vụ “aplication as service”. Các giao dịch, tương tác đều thông qua môi trường internet. Cùng với sự phát triển đó là sự ra đời của các ngôn ngữ và các giao thức giao tiếp để phục vụ những mục đích và dịch vụ nhất định. Có thể kể đến một số ngôn ngữ HTML, Java, PHP, .Net ... Với dự phát triển của ngôn ngữ XML mấy năm gần đây và đã được sử dụng rỗng rãi trên nhiều ngôn ngữ khác nhau. Cùng với nó là sự phát triển của các cơ sở dữ liệu đặc biệt là cơ sở dữ liệu XML để thao tác dữ liệu với các ngôn ngữ đó một cách dễ dàng và tiện lợi.

**3.1 XML DB là gì?**

* là một hệ thống phần mềm dữ liệu không đổi cho phép quy định dữ liệu, lưu trữ trong định dạng XML.
* Dữ liệu này có thể truy vấn, chuyển đổi, exported và trả lại kết quả để gọi hệ thống.
* Là một ‘hương vị’ của CSDL hướng đối tượng – là một phần của NoSQL DB.
* Các dữ liệu lưu trữ trong DB có thể truy vấn bằng Xquery và exported sang các định dạng mong muốn

**3.2 Đặc điểm XML DB**

* XML phù hợp cho dữ liệu thưa, dữ liệu lồng nhau và dữ liệu có nội dung hỗn hợp (như một văn bản với các thẻ đánh dấu).
* Là một ‘hương vị’ của CSDL hướng đối tượng – là một phần của NoSQL DB.
* Một doanh nghiệp có thể có rất nhiều định dạng XML trong tài liệu hiện hành.
* Dữ liệu web có sẵn như RDF/XML

**3.3 XML DB có hai loại**

* XML-enabled: không có gì ngoài việc cung cấp phần mở rộng để chuyển đổi tài liệu XML. Đây là CSDLQH - nơi dữ liệu được lưu trữ trong các bảng gồm các hàng và các cột. Các bảng được thiết lập gồm các bản ghi trong đó gồm các trường.
* Native XML (NXD):Dựa trên ý tưởng thùng chứa chứ không phải định dạng bảng. Nó có thể lưu trữ số lượng lớn các tài liệu XML và dữ liệu. NXD được truy vấn bằng các câu lệnh truy vấn XQuery. Xử lý tài liệu XML và các dữ liệu có cấu trúc, không phải là các bảng, bản ghi hay các truòng.

**3.4 XML là gì?**

* là một ngôn ngữ đánh dấu mà bạn có thể sử dụng để tạo ra thẻ riêng của mình.
* XML ra đời nhằm khắc phục những nhược điểm của HTML.
* XML có khả năng mô tả nhiều loại dữ liệu khác nhau và đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet.
* Dễ dàng xử lý, chuyển tải và trao đổi dữ liệu.
* Mô tả dữ liệu và cách dữ liệu muốn thể hiện thông qua tên các thẻ.
* Tổ chức dữ liệu có cấu trúc và phân cấp.

**3.5 Tài liệu XML**

* Một tài liệu XML có hai phần:
  + Cấu trúc logic: định nghĩa các phần tử, các thuộc tính, kiểu dữ liệu ...
  + Cấu trúc vật lý: chứa dữ liệu và các phần tử chứa dữ liệu như text, hình ảnh, media ...
* Tài liệu XML có thể được soạn thảo bởi các phần mềm như: notepad, editPlus, XML notepad ...

**3.6 Tại sao XMLDB lại tối ưu hơn**

* CSDL thường có thể lưu trữ tài liệu có cấu trúc và không có cấu trúc.cả hai đều yêu cầu các cấu trúc dữ liệu mà không thường xuyên thay đổi. Tuy nhiên điểm yếu của CSDLQH là việc lưu trữ bán cấu trúc.bạn có thể lưu dữ liệu bán cấu trúc trong các cấu trúc quan hệ nhưng kết quả mong muốn trả về là một tài liệu cấu trúc thường xuyên thay đổi.
* Trong khi đó XML phù hợp với định dạng bán cấu trúc. Thêm nữa bạn có thể duy trì mô hình dữ liệu một cách dễ dàng, thêm tên của các thành phần không làm thay đổi cấu trúc dữ liệu- vì nó là cáu trúc cây.
* XML phù hợp với các tài liệu như hồ sơ xin việc, các mô tả sản phẩm, các đơn hàng của khách hàng ...
* NXD phù hợp với để chứa tài liệu XML hoặc các phân mảnh của tài liệu XML.
* NXD: là cách dùng các công nghệ có liên quan đến XML mà không pha tạp với các công nghệ phi XML- tức là khả năng sử dụng XQuery và Xpath không có chút SQL nào.
* NXD NXD cũng có thể được phân loại như là một kho chứa tài liệu như CSDLQH
* NXD: thực hiện câu lệnh truy vấn select nhanh. Với insert và update thì chậm hơn.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ CHƯƠNG TRÌNH

1. Thiết kế cơ sở dữ liệu

* Mô tả cấu trúc XML database:

Cú pháp XML cơ bản cho một phần tử là:

**<tên** thuộc\_tính="giá trị"**>**nội dung**</tên>**

* Cấu trúc file data.xml:

<DictItem>

<Items>

<DictData>

<Key>key</Key>

<Mean>mean</Mean>

</DictData>

</Items>

</DictItem>

Key: từ khóa

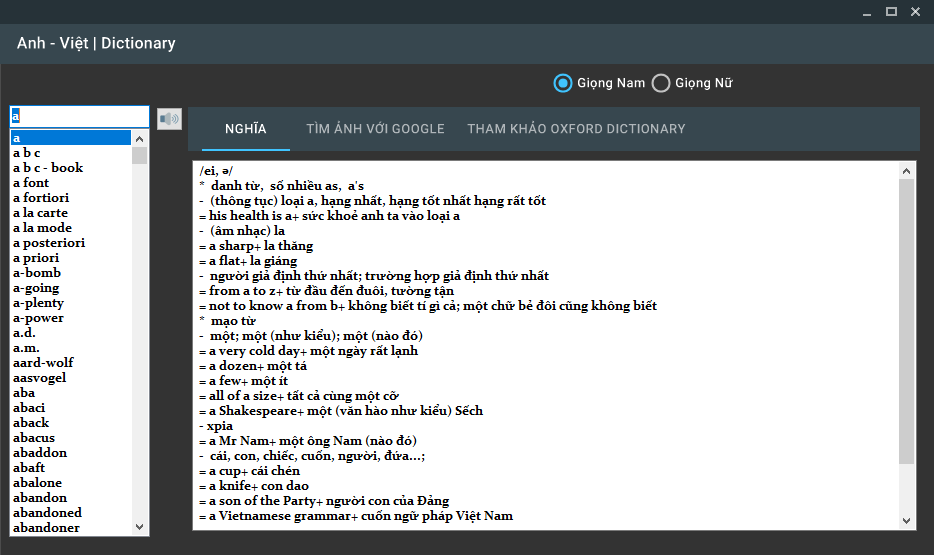
Mean: nghĩa

* Lý do lựa chọn xml database:

Nếu sử dụng sql database người dung sẽ cần cài đặt sql sever gây khó khăn trong việc sử dụng phần mềm.

1. Các chức năng chính :

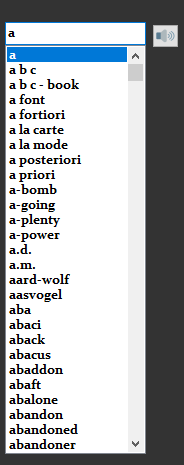
* Tra cứu cơ bản:



Mục đích: Dùng 1 combobox để tìm kiếm key, sau đó so sánh với csdl và hiện nghĩa của từ lên textbox bên phải

Cách thực hiện: sử dụng even SelectedIndexChanged và so sánh với database sau đó hiện nghĩa tương ứng vào textbox

* Phát âm từ được chọn:



Mục đích: Phát âm được từ khóa người dùng đã nhập

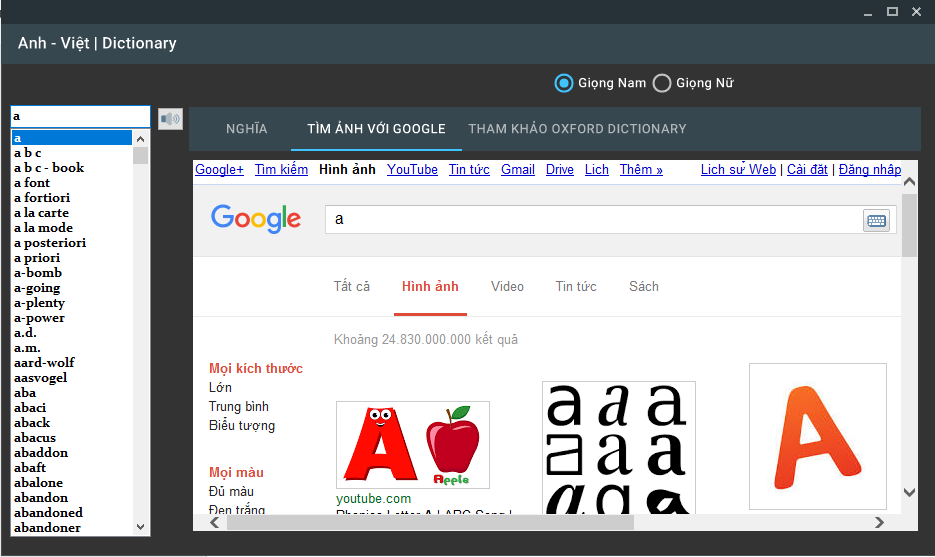
Cách thực hiện: sử dụng class SpeechSynthesizer được cung cấp trong c#

Thư viện sử dụng:

using System.Speech;

using System.Speech.Synthesis;

* Tìm kiếm hình ảnh với google:



Mục đích: Hiện kết quả tìm kiếm hình ảnh từ google cho key

Cách thực hiện: sử dụng class WebBrowser có sẵn của c# winform

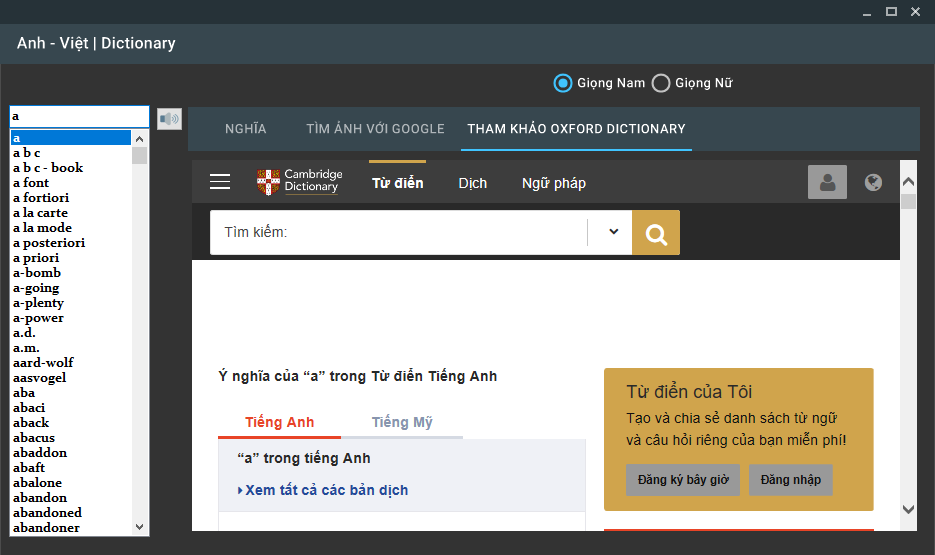
Thư viện sử dụng:

using Google;

using Google.API;

using Google.API.Search;

* Tham khảo Cambridge Dictionary:



Mục đích: Hiện kết quả tra cứu key với nghĩa đầy đủ hơn tại Cambridge Dictionary

Cách thực hiện: sử dụng class WebBrowser có sẵn của c# winform

Thư viện sử dụng:

using Google;

using Google.API;

using Google.API.Search;

# CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

**1. KẾT LUẬN:**

**1.1 Thuận lợi và khó khăn trong lúc thực hiện đồ án:**

**1.1.1 Thuận lợi:**

**-** Bản thân đã được trang bị các kiến thức liên quan về thiết kế phần mềm. Đã có kinh nghiệm trong việc tìm hiểu, phân tích thiết kế hệ thống thông tin.

- Từ điển là một ứng dụng khá quen thuộc và dễ dàng trong cách tiếp cận các chức năng bởi vì khá phổ biến.

- Được sự giúp đỡ của tận tình của GVHD và sự trợ giúp kiến thức từ Internet.

**1.1.2 Khó khăn:**

**-** Số lượng người trong nhóm ít 2 dẫn đến khối lượng công việc nhiều.

- Với Cơ sở dữ liệu là XML database kiến thức vẫn nắm chưa thật kỹ về hệ quản trị này, áp dụng chỉ dẫn thiết kế và kết nối trên Internet.

- Tìm kiếm database dạng XML trên Internet không có dẫn đến việc phải tự tìm hiểu việc chuyển đổi csdl từ SQL sang XML

- Do chưa có kinh nghiệm về thiết kế một phần mềm nên còn rất khó khăn khi triển khai công việc.

**1.2 Kết quả đạt được:**

Qua thời gian nghiên cứu và học tập, nhìn chung đồ án đã hoàn thành các mục tiêu, nhiệm vụ đề ra của một chương trình từ điển anh-việt với các chức năng như hệ thống phân tích thiết kế đã mô tả như ở trên.

Xây dựng được một chương trình hoàn chỉnh có khả năng áp dụng vào thực tế.

**2. ƯU VÀ NHƯỢC ĐIỂM CỦA HỆ THỐNG**

2.1 Ưu điểm:

- Hệ thống được xây dựng gọn nhẹ, dễ sử dụng.

- Giao diện thân thiện với người sử dụng.

2.2 Nhược điểm:

- Kỹ thuật lập trình chưa tốt.

- Bắt lỗi hạn chế, chưa hoàn thiện.

- Các chức năng của chương trình chưa linh động, có thể gây khó khăn khi sử dụng.

**3. HƯỚNG PHÁT TRIỂN:**

**-** Khả năng xử lý được tất cả các sự kiện, các lỗi ngoài ý muốn tốt hơn của chương trình và dùng thao tác lên chương trình.

- Nâng cao tính linh động của chương trình.

- Thêm các chức năng mới để đáp ứng điều kiện của người dùng.

- Nâng cao kỷ thuật lập trình và hoàn chỉnh các thành phần còn thiếu theo hướng chuyên nghiệp, chạy thử, khả năng đưa vào áp dụng thức tế khả quan.

**TÀI LIỆU THAM KHẢO**

* 1. https://www.howkteam.vn
  2. https://www.Caulacbovb.com
  3. https://www.Dictionary.cambridge.org
  4. https://www.O7planning.org