### 

**ĐẠI HỌC UEH**

**TRƯỜNG CÔNG NGHỆ VÀ THIẾT KẾ**

**KHOA CÔNG NGHỆ THÔNG TIN KINH DOANH**



**ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI**

**ỨNG DỤNG CẤU TRÚC DICTIONARY ĐỂ TẠO MỘT BỘ TỪ ĐIỂN ANH – VIỆT**

Học phần: Cấu trúc dữ liệu & giải thuật

Danh sách nhóm:

1.Trương Thị Kiều Phương

2. Đoàn Thị Mai Linh

3. Bùi Anh Minh

4. Bùi Thị Thùy Linh

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

KHÓA: 50

Giảng viên: TS. Đặng Ngọc Hoàng Thành

TP.HCM ngày 4 tháng 4 năm 2025

**Mục lục**

[**Lời mở đầu** 3](#_Toc194755705)

[**Chương 1: Cấu trúc từ điển Dictionary** 4](#_Toc194755706)

[**1.1. Khái niệm Dictionary** 4](#_Toc194755707)

[**1.2. Cấu trúc & Cài đặt từ điển** 4](#_Toc194755708)

[**1.3. Các thuật toán** 5](#_Toc194755709)

[**1. Thuật toán Thêm từ vào từ điển (AddWord)** 5](#_Toc194755710)

[**2. Thuật toán Kiểm tra xem từ có tồn tại không (Contains)** 5](#_Toc194755711)

[**3. Thuật toán Lấy định nghĩa của từ (GetDefinition)** 6](#_Toc194755712)

[**4. Thuật toán Thêm từ vào danh sách yêu thích (AddToFavorites)** 6](#_Toc194755713)

[**5. Thuật toán Lấy danh sách các từ yêu thích (GetFavorites)** 7](#_Toc194755714)

[**6. Thuật toán In toàn bộ từ điển (PrintAllWords)** 7](#_Toc194755715)

[**Chương 2. Phân tích và thiết kế lớp** 8](#_Toc194755716)

[**2.1. Phân tích yêu cầu và ứng dụng cấu trúc Dictionary trong thiết kế từ điển Anh - Việt** 8](#_Toc194755717)

[**2.1.1. Các chức năng trong app từ điển** 8](#_Toc194755718)

[**2.1.2. Tài nguyên cần thiết** 8](#_Toc194755719)

[**2.1.3. Nhập liệu** 9](#_Toc194755720)

[**2.1.4. Khởi tạo biến và chuẩn bị dữ liệu** 9](#_Toc194755721)

[**2.2. Sơ đồ lớp** 10](#_Toc194755722)

[**2.2.1. Class Program** 10](#_Toc194755723)

[**2.2.2. Class Form1** 10](#_Toc194755724)

[**2.2.3. Class DictionaryManager** 11](#_Toc194755725)

[**2.2.4. Class WordEntry** 11](#_Toc194755726)

[**2.3. Cài đặt lớp** 11](#_Toc194755727)

[**2.3.1. Class “ Program”** 12](#_Toc194755728)

[**2.3.2. Class Form1** 12](#_Toc194755729)

[**2.3.3. Class DictionaryManager** 14](#_Toc194755730)

[**2.3.4. Class WordEntry** 15](#_Toc194755731)

[**Chương 3. Thiết kế giao diện** 17](#_Toc194755732)

[**3.1. Giao diện menu chính** 17](#_Toc194755733)

[**3.2. Chi tiết chức năng** 17](#_Toc194755734)

[**Chương 4: Thảo luận & đánh giá** 19](#_Toc194755735)

[**4.1. Các kết quả nhận được** 19](#_Toc194755736)

[***4.1.1. Giao diện khi khởi động chương trình*** 19](#_Toc194755737)

[**4.1.2. Tra nghĩa của từ vựng** 20](#_Toc194755738)

[**4.1.3. Thêm dữ liệu về từ vựng vào danh sách từ điển yêu thích** 20](#_Toc194755739)

[**4.1.4. Xem danh sách từ điển yêu thích** 21](#_Toc194755740)

[**4.2. Khó khăn gặp phải** Trong quá trình làm việc, nhóm gặp một số khó khăn đáng kể như: 22](#_Toc194755741)

[**4.3. Hướng phát triển** Trong tương lai, nhóm định hướng mở rộng ứng dụng theo các hướng sau: 22](#_Toc194755742)

[**KẾT LUẬN** 24](#_Toc194755743)

[**PHỤ LỤC** 25](#_Toc194755744)

[**TÀI LIỆU THAM KHẢO** 26](#_Toc194755745)

# **Lời mở đầu**

Trong bối cảnh hiện đại của kỷ nguyên thông tin, các cấu trúc dữ liệu và thuật toán đóng vai trò then chốt trong việc xử lý và tổ chức thông tin một cách tối ưu. Trong môn học "Cấu trúc dữ liệu và giải thuật", chúng em được làm quen với một loạt các cấu trúc dữ liệu thông dụng, mỗi cấu trúc mang những ưu điểm và ứng dụng riêng biệt.

Trong khuôn khổ báo cáo đồ án lần này, chúng em xin trình bày về một cấu trúc dữ liệu đặc biệt, đó là cấu trúc **Dictionary**, và ứng dụng của nó trong dự án **"Ứng dụng cấu trúc Dictionary để xây dựng từ điển Anh-Việt"**.

Cấu trúc Dictionary là một cấu trúc dữ liệu có khả năng lưu trữ các cặp khóa - giá trị một cách hiệu quả. Điểm khác biệt so với các cấu trúc dữ liệu như mảng hay danh sách liên kết là Dictionary cho phép truy xuất giá trị của một phần tử dựa trên khóa tương ứng một cách nhanh chóng, từ đó giảm thiểu thời gian truy vấn dữ liệu.

Trong phạm vi dự án, chúng em đã ứng dụng cấu trúc Dictionary để xây dựng từ điển Anh-Việt. Thông qua việc sử dụng cấu trúc này, chúng em có thể lưu trữ các từ vựng tiếng Anh và nghĩa tiếng Việt đi kèm. Điều này tạo ra một công cụ hữu ích để tra cứu từ vựng và thực hiện dịch thuật một cách nhanh chóng và thuận tiện. Bên cạnh đó, chúng em cũng đã tích hợp công nghệ Windows Forms để thiết kế một giao diện người dùng thân thiện và trực quan.

Chúng em tin rằng báo cáo này sẽ cung cấp một cái nhìn chi tiết và rõ ràng về cấu trúc Dictionary và ứng dụng của nó trong đề tài "Ứng dụng cấu trúc Dictionary để xây dựng từ điển Anh-Việt".

Chúng em xin bày tỏ sự tri ân sâu sắc đến **Thầy Đặng Ngọc Hoàng Thành** vì sự nhiệt tình và định hướng trong suốt thời gian giảng dạy vừa qua. Thầy đã dành rất nhiều tâm huyết và công sức để truyền đạt kiến thức về cấu trúc dữ liệu và giải thuật một cách tường tận, dễ hiểu. Chúng em vô cùng biết ơn sự tận tâm của Thầy trong việc giải đáp thắc mắc và hỗ trợ nhóm trong suốt quá trình nghiên cứu và hoàn thiện đồ án. Những lời nhận xét và định hướng quý báu từ Thầy đã giúp chúng em rèn luyện được nhiều kỹ năng mới cũng như nâng cao khả năng tư duy và giải quyết vấn đề.

Cuối cùng, chúng em xin gửi đến Thầy những lời chúc tốt đẹp nhất. Kính chúc Thầy luôn giữ vững ngọn lửa đam mê trong sự nghiệp giáo dục, tiếp tục truyền cảm hứng cho các thế hệ sinh viên. Chúng em chân thành cảm ơn Thầy vì những đóng góp vô giá và hy vọng sẽ còn nhiều cơ hội được học hỏi thêm từ Thầy trong tương lai.

**Kính chúc Thầy sức khỏe dồi dào, nhiều niềm vui và thành công trên con đường giảng dạy.**

# **Chương 1: Cấu trúc từ điển Dictionary**

## **1.1. Khái niệm Dictionary**

**Dictionary** (*từ điển*) là một cấu trúc dữ liệu dùng để lưu trữ các cặp dữ liệu theo dạng **khóa** (*key*) và **giá trị** (*value*). Mỗi khóa là duy nhất, cho phép truy xuất giá trị tương ứng một cách nhanh chóng và hiệu quả. **Dictionary** hỗ trợ các thao tác cơ bản như thêm mới (*insert*), tìm kiếm (*search*), cập nhật (*update*), và xóa (*delete*).

A diagram of a diagram

Description automatically generated

## **1.2. Cấu trúc & Cài đặt từ điển**

**DictionaryBase** là một lớp **cơ sở** (*base class*) trong .NET Framework, đóng vai trò nền tảng để xây dựng các kiểu dữ liệu từ điển (*dictionary*) trong ngôn ngữ lập trình C#. Lớp này không trực tiếp tạo ra từ điển, mà được kế thừa bởi các lớp con như Dictionary<TKey, TValue> để hiện thực hóa các chức năng lưu trữ và truy xuất dữ liệu.

Việc kế thừa từ **DictionaryBase** giúp các lớp từ điển con sở hữu sẵn những thuộc tính, phương thức cơ bản phục vụ cho thao tác lưu trữ, truy xuất, cập nhật và quản lý các cặp dữ liệu dạng khóa - giá trị một cách tiện lợi và hiệu quả.

A screen shot of a computer

Description automatically generated

## **1.3. Các thuật toán**

Dưới đây là phân tích các thuật toán trong lớp DictionaryManager, là một lớp kế thừa từ DictionaryBase, quản lý các thao tác với từ điển.

### **1. Thuật toán Thêm từ vào từ điển (AddWord)**

**A close-up of a computer screen

Description automatically generated**

**Mô tả:** Thêm một từ mới vào từ điển, với **word** là **khóa** và **meaning** là **giá trị tương ứng.  
Thuật toán:**

* Kiểm tra xem word đã tồn tại trong **InnerHashtable** hay chưa.
* Nếu chưa tồn tại, sử dụng **Add(word, meaning)** để thêm cặp khóa-giá trị vào.
* Nếu đã tồn tại, có thể ghi đè hoặc thông báo tùy theo thiết kế.

### **2. Thuật toán Kiểm tra xem từ có tồn tại không (Contains)**

**A close-up of a computer screen

Description automatically generated**

**Mô tả:** Kiểm tra xem từ có tồn tại trong từ điển hay không. **Thuật toán:** Dùng phương thức **Contains()** của InnerHashtable để kiểm tra sự tồn tại của khóa (word).

### **3. Thuật toán Lấy định nghĩa của từ (GetDefinition)**

**A computer screen shot of text

Description automatically generated**

**Mô tả:** Lấy định nghĩa của từ từ từ điển.

**Thuật toán:**

* Kiểm tra xem từ có tồn tại trongInnerHashtablebằng phương thức **Contains().**
* Nếu từ có trong từ điển, trả về giá trị tương ứng với khóa đó.
* Nếu không tồn tại, trả về thông báo **'Từ không có trong từ điển'** hoặc giá trị **null** tuỳ cách xử lý.

### **4. Thuật toán Thêm từ vào danh sách yêu thích (AddToFavorites)**

**A white background with red and blue text

Description automatically generated**

**Mô tả:** Thêm từ vào danh sách yêu thích bằng cách thay đổi giá trị của từ trong từ điển, thêm tiền tố "*Yêu thích*".  
**Thuật toán:**

* Truy xuất giá trị của từ từ InnerHashtable và thay đổi giá trị của nó (nếu từ đã tồn tại trong từ điển).
* Nếu từ chưa có, thêm vào từ điển với giá trị là **"Yêu thích: [trống]"** hoặc cho người dùng nhập giá trị yêu thích.

### **5. Thuật toán Lấy danh sách các từ yêu thích (GetFavorites)**

**A screen shot of a computer code

Description automatically generated**

**Mô tả:** Liệt kê các từ yêu thích ra màn hình hoặc trả về danh sách từ yêu thích dưới dạng chuỗi/tập hợp.

**Thuật toán:**

* Duyệt qua tất cả các phần tử trong InnerHashtable bằng cách sử dụng vòng lặp foreach.
* Kiểm tra nếu giá trị bắt đầu bằng "Yêu thích" (sử dụng phương thức StartsWith()).
* In ra các từ yêu thích.

### **6. Thuật toán In toàn bộ từ điển (PrintAllWords)**

**A screen shot of a computer code

Description automatically generated**

**Mô tả:** In tất cả các từ và định nghĩa trong từ điển.

**Thuật toán:**

* Duyệt qua tất cả các phần tử trong InnerHashtable bằng vòng lặp foreach.
* Với mỗi phần tử, in ra **khóa** (*word*) và **giá trị** (*meaning*) theo định dạng rõ ràng (**ví dụ:** word: meaning).

# **Chương 2. Phân tích và thiết kế lớp**

## **2.1. Phân tích yêu cầu và ứng dụng cấu trúc Dictionary trong thiết kế từ điển Anh - Việt**

### **2.1.1. Các chức năng trong app từ điển**

1. **Tra từ vựng**

* Nghĩa
* Loại từ
* Phiên âm

1. **Thêm vào danh sách yêu thích**

A screenshot of a video game

Description automatically generated

*Các chức năng trong từ điển*

### **2.1.2. Tài nguyên cần thiết**

Để xây dựng ứng dụng từ điển như mô tả, nhóm cần chuẩn bị một tập dữ liệu chứa danh sách các từ vựng tiếng Anh, trong đó mỗi mục bao gồm đầy đủ các thông tin: **nghĩa tiếng Việt**, **loại từ** (*danh từ, động từ, tính từ, v.v*.), và **phiên âm** theo chuẩn quốc tế (*IPA*).

Danh sách này đóng vai trò là nguồn dữ liệu đầu vào chính, giúp hệ thống có thể thực hiện các thao tác tra cứu, tìm kiếm và hiển thị thông tin một cách chính xác. Dữ liệu sẽ được lưu trữ ở định dạng văn bản đơn giản (*txt*), thuận tiện cho việc đọc – ghi trong quá trình lập trình, đồng thời dễ dàng cập nhật và mở rộng.

### **2.1.3. Nhập liệu**

Để có dữ liệu đưa vào list và các Dictionary, nhóm đã tham khảo nhiều nguồn và tạo ra 1 list từ điển ở Notepad với các cú pháp như sau:

A white background with black text

Description automatically generated

*Từ điển ở Notepad*

### **2.1.4. Khởi tạo biến và chuẩn bị dữ liệu**

Đầu tiên, nhóm tiến hành **khai báo các biến thành phần** trong lớp Form1 để quản lý dữ liệu từ điển và danh sách yêu thích. Cụ thể, đường dẫn đến file từ điển (anhviet.txt) được gán vào biến dictionaryPath với định dạng **chuỗi**.  
Các cấu trúc dữ liệu Dictionary<string, WordEntry> và HashSet<string> cũng được khởi tạo để lưu trữ dữ liệu chính và danh sách từ yêu thích.

A black and blue text

Description automatically generated

*Minh họa đoạn khai báo các biến phục vụ cho quá trình xử lý dữ liệu trong ứng dụng*

## **2.2. Sơ đồ lớp**

Hệ thống ứng dụng từ điển được thiết kế gồm bốn lớp chính, mỗi lớp đảm nhận một vai trò riêng biệt trong tổ chức mã nguồn và xử lý chức năng. Dưới đây là sơ đồ các lớp và mô tả ngắn gọn:

### **2.2.1. Class Program**

Lớp Program là điểm khởi đầu của ứng dụng Windows Forms. Nó thực hiện các công việc sau:

1. **[STAThread]**: Chỉ định ứng dụng sử dụng mô hình luồng **STA**.
2. **Static void Main()**: Phương thức bắt đầu khi ứng dụng chạy.
3. **Application.EnableVisualStyles()**: Kích hoạt giao diện hiện đại cho ứng dụng.
4. **Application.SetCompatibleTextRenderingDefault(false)**: Cấu hình cách vẽ văn bản.
5. **Application.Run(new Form1())**: Khởi tạo và chạy cửa sổ chính (Form1).

Nhìn chung, lớp này khởi động ứng dụng và hiển thị cửa sổ chính.

### **2.2.2. Class Form1**

Lớp xử lý giao diện người dùng và các thao tác chính của ứng dụng.

**Biến thành phần:**

1. **dictionary:** lưu danh sách từ điển.
2. **favoriteWords:** lưu danh sách từ yêu thích.
3. **dictionaryPath:** đường dẫn tới file dữ liệu.

**Phương thức:**

1. **LoadDictionary():**đọc dữ liệu từ file và load vào dictionary.
2. **SetBackgroundImage():** hiển thị hình nền giao diện.
3. **btnSearch\_Click():** xử lý sự kiện tìm kiếm từ.
4. **btnAddToFavorites\_Click():** thêm từ vào danh sách yêu thích.
5. **btnShowFavorites\_Click():** hiển thị các từ yêu thích đã lưu.
6. **txtSearch\_TextChanged():** xóa kết quả khi nhập từ mới.

Ứng dụng cung cấp các tính năng cơ bản để tra cứu từ điển, lưu và quản lý các từ yêu thích, và hiển thị giao diện đẹp với hình nền.

### **2.2.3. Class DictionaryManager**

Lớp DictionaryManager kế thừa từ DictionaryBase và cung cấp các chức năng quản lý từ điển.

1. **AddWord(string word, string meaning)**: Thêm một từ và nghĩa của nó vào từ điển.
2. **Contains(string word)**: Kiểm tra xem từ có tồn tại trong từ điển hay không.
3. **GetDefinition(string word)**: Lấy nghĩa của từ nếu từ đó tồn tại trong từ điển, nếu không thì trả về thông báo "*Không tìm thấy từ này*".
4. **AddToFavorites(string word)**: Đánh dấu từ là yêu thích bằng cách thêm tiền tố "*Yêu thích*" vào nghĩa của từ.
5. **GetFavorites()**: Hiển thị danh sách các từ đã được đánh dấu là yêu thích.
6. **PrintAllWords()**: In ra tất cả các từ và nghĩa trong từ điển.

Lớp này cho phép người dùng thêm từ, tra cứu nghĩa, đánh dấu và xem các từ yêu thích, cũng như in ra toàn bộ từ điển.

### **2.2.4. Class WordEntry**

Lớp WordEntry đại diện cho một từ trong từ điển với các thuộc tính và phương thức sau:

1. **Thuộc tính**:

* English: Từ tiếng Anh.
* Meaning: Nghĩa của từ (bằng tiếng Việt).
* Pronunciation: Phiên âm của từ.
* Type: Loại từ (ví dụ: danh từ, động từ, tính từ, v.v.).

1. **Phương thức**:

* **ToString()**: Trả về một chuỗi mô tả đầy đủ về từ, bao gồm từ tiếng Anh, phiên âm, loại từ và nghĩa.

Chức năng của lớp này là lưu trữ thông tin về một từ và cung cấp cách hiển thị thông tin này dưới dạng một chuỗi dễ đọc.

## **2.3. Cài đặt lớp**

Phần này trình bày một số đoạn mã tiêu biểu thể hiện cách cài đặt các lớp và phương thức đã nêu trong mục 2.2.

### **2.3.1. Class “ Program”**

Lớp Program chỉ chứa phương thức Main() – nơi khởi động ứng dụng:

A screenshot of a computer program

Description automatically generated

### **2.3.2. Class Form1**

Lớp Form1 khai báo các biến như dictionary, favoriteWords, dictionaryPath, đồng thời hiện thực các phương thức tương tác giao diện.

**A screenshot of a computer program

Description automatically generated**

**A screenshot of a computer program

Description automatically generatedA screen shot of a computer

Description automatically generated**

### **2.3.3. Class DictionaryManager**

Lớp mở rộng từ DictionaryBase, quản lý thao tác dữ liệu với từ điển.

**A screenshot of a computer program

Description automatically generated**

### **2.3.4. Class WordEntry**

Lớp đại diện cho một mục từ.

**A screenshot of a computer program

Description automatically generated**

# **Chương 3. Thiết kế giao diện**

Chương này trình bày giao diện người dùng (*UI*) của ứng dụng từ điển, được xây dựng bằng *Windows Forms* trong môi trường *Visual Studio*.

Giao diện được thiết kế theo hướng tối giản, thân thiện với người dùng, đảm bảo hiển thị đầy đủ thông tin từ vựng (nghĩa, loại từ, phiên âm) và cung cấp các nút chức năng thao tác như tìm kiếm, thêm vào danh sách yêu thích, và xem lại từ đã lưu.

## **3.1. Giao diện menu chính**

**A screenshot of a video game

Description automatically generated**

## **3.2. Chi tiết chức năng**

|  |  |  |
| --- | --- | --- |
| **Loại Control** | **Tên biến** | **Chức năng** |
| **Label** | lblTitle | Hiển thị tiêu đề "Từ điển Anh-Việt" |
| **TextBox** | txtSearch | Nhập từ cần tìm kiếm |
| **Button** | btnSearch | Nút tìm kiếm nghĩa của từ |
| **Label** | lblResult | Hiển thị kết quả tra cứu |
| **Button** | btnAddToFavorites | Thêm từ vào danh sách yêu thích |
| **Button** | btnShowFavorites | Hiển thị danh sách từ yêu thích |
| **TextBox** | txtWord | Ô hiển thị danh sách từ yêu thích |
| **Button** | btnRemoveFromFavorites | Xóa từ khỏi danh sách yêu thích |

# **C****hương 4: Thảo luận & đánh giá**

## **4.1. Các kết quả nhận được**

Sau quá trình thực hiện, nhóm đã hoàn thành một ứng dụng từ điển Anh–Việt với các chức năng cơ bản:

* Tìm kiếm từ vựng, hiển thị nghĩa, loại từ và phiên âm.
* Thêm từ vào danh sách yêu thích và hiển thị lại khi cần.
* Giao diện có hình nền trực quan, dễ sử dụng.
* Dữ liệu được nạp từ file .txt, dễ cập nhật và mở rộng.

Ứng dụng hoạt động ổn định, xử lý các thao tác nhanh, đúng mục tiêu ban đầu của đề tài.

### ***4.1.1. Giao diện khi khởi động chương trình***

**A screenshot of a video game

Description automatically generated**

*Giao diện khi khởi động chương trình*

## **4.1.2. Tra nghĩa của từ vựng**

Nhập từ vựng vào textbox “txtSearch” rồi ấn button “Tìm kiếm”.

A screenshot of a computer screen

Description automatically generated

*Tra nghĩa của từ vựng*

### **4.1.3. Thêm dữ liệu về từ vựng vào danh sách từ điển yêu thích**

Nhập từ vựng vào textbox “txtSearch” rồi ấn button “yêu thích”.

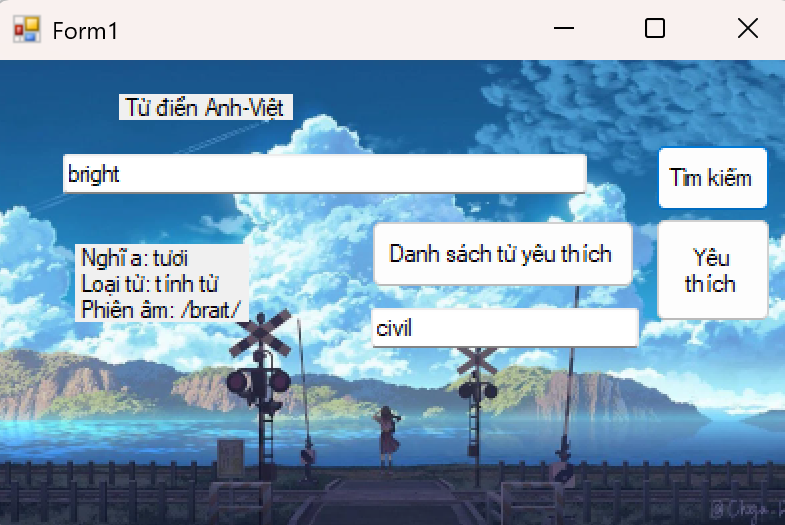
**A screenshot of a computer screen

Description automatically generated**

*Thêm từ vựng vào danh sách yêu thích*

### **4.1.4. Xem danh sách từ điển yêu thích**

Nhấn vào button "danh sách từ yêu thích" để xem danh sách yêu thích.

****

## **4.2. Khó khăn gặp phải** Trong quá trình làm việc, nhóm gặp một số khó khăn đáng kể như:

* Khi phóng to, thu nhỏ màn hình Form thì các control bên trong không thay đổi size (kích thước các control không phù hợp khi Form thay đổi).
* Khi nhấn vào các button thì chưa nổi bật (nghĩa là khi ấn một button nào đó thì nên có một dấu hiệu để cho người sử dụng biết rằng mình đang sử dụng chức năng của button đó).
* Chưa có những ví dụ về cách sử dụng của từ vựng.
* Chưa có khả năng phát âm từ vựng.

## **4.3. Hướng phát triển** Trong tương lai, nhóm định hướng mở rộng ứng dụng theo các hướng sau:

* Nghiên cứu sâu hơn về cách xử lý **thuộc tính của control** (Label, Button, TextBox,...) và các **sự kiện (Event)** liên quan, nhằm **tự động điều chỉnh kích thước giao diện khi Form thay đổi kích thước**, khắc phục hạn chế hiện tại về tính linh hoạt UI.
* Tìm hiểu cách thay đổi **giao diện động cho nút bấm**, cụ thể là làm cho Button đổi màu khi được nhấn để tăng tính trực quan và cải thiện trải nghiệm người dùng.
* Nhóm cần thêm thời gian để **thu thập và chọn lọc các nhóm từ đồng nghĩa – trái nghĩa**, giúp nâng cao tính ứng dụng của từ điển và hỗ trợ người học mở rộng vốn từ.
* Có thể mở rộng hệ thống bằng cách:

+ Tạo thêm một **từ điển phụ** chứa những từ nằm trong danh sách từ chính → dùng như **nguồn tham khảo nội bộ**.

+ Hoặc chèn thêm **ví dụ minh họa ngắn** ngay sau từng mục từ trong file anhviet.txt để người dùng dễ hình dung cách dùng từ.

* Cần thêm thời gian để **thu thập phát âm chuẩn** cho từng từ vựng và tích hợp các đoạn audio tương ứng vào ứng dụng, giúp người học luyện nghe và phát âm chính xác hơn.

# **KẾT LUẬN**

Thông qua việc thực hiện đồ án này, nhóm đã có cơ hội tiếp cận và áp dụng kiến thức về cấu trúc dữ liệu và giải thuật một cách cụ thể, rõ ràng hơn so với những gì từng học lý thuyết trước đó.

Nhóm đã hiểu rõ hơn về cách lựa chọn cấu trúc dữ liệu phù hợp với từng chức năng, và thấy được tầm quan trọng của việc tổ chức code sao cho dễ hiểu, dễ mở rộng. Dù vẫn còn nhiều thứ chưa làm được như mong muốn, nhưng các thành viên đã cố hết sức để có thể hoàn thành bài tập một cách tốt nhất trong khả năng của mình.

Qua đồ án này, nhóm cũng nhận ra rằng việc viết ra một chương trình không chỉ là vấn đề về cú pháp, mà còn là quá trình tư duy, tổ chức và thử nghiệm. Việc được thử sức qua đồ án này rất quan trọng và ý nghĩa trên con đường học tập, giúp chúng em có thể biến những lý thuyết suông thành những thứ có thể áp dụng vào thực tiễn. Bài làm của nhóm sẽ không chỉ là một kết quả nộp môn, mà còn là một bước khởi đầu nho nhỏ giúp chúng em tự tin hơn khi tiếp cận những đề tài kỹ thuật trong tương lai.

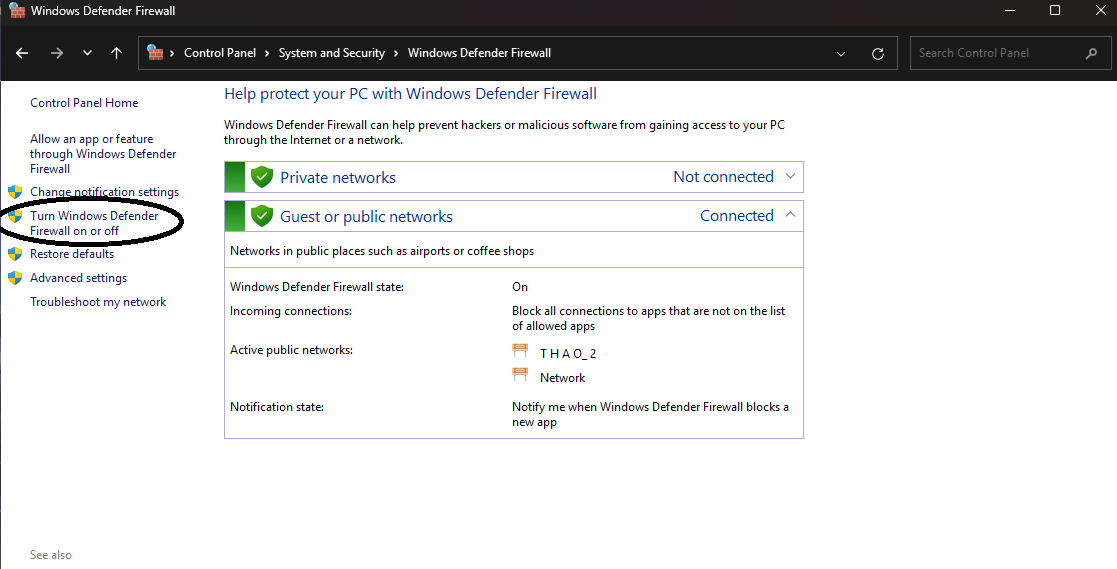
**PHỤ LỤC**

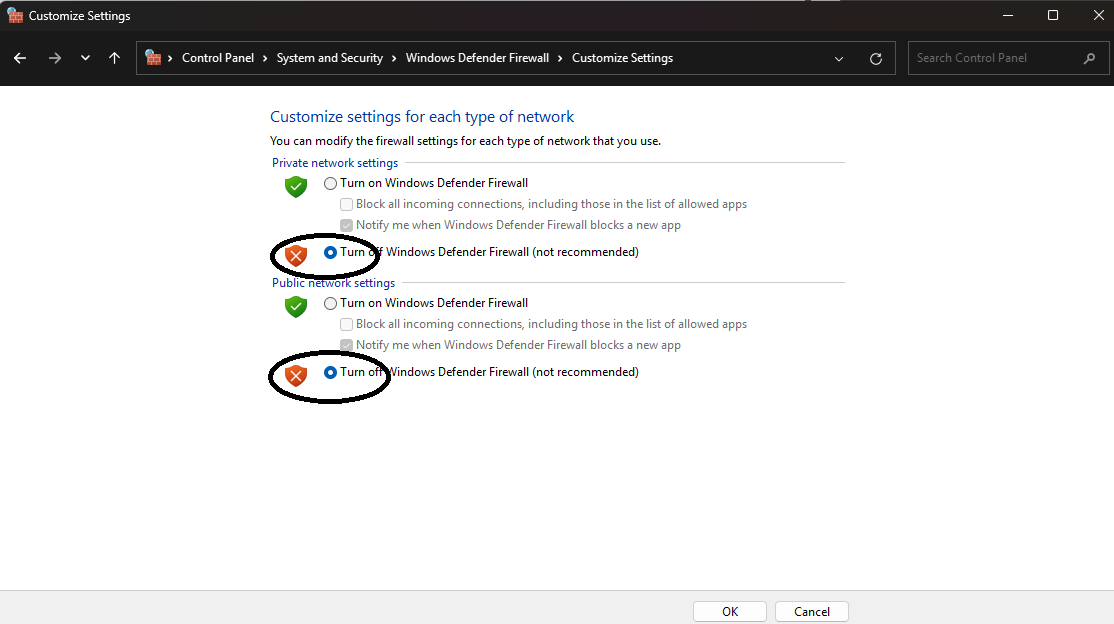
Toàn bộ mã nguồn trên GitHub: [**https://github.com/PhuongT-T/App-T-i-n-Anh\_Vi-t/tree/master**](https://github.com/PhuongT-T/App-T-i-n-Anh_Vi-t/tree/master)

1. **Hướng dẫn cài đặt:**

**Bước 1: Tắt FireWall ( tường lửa )** Vì sẽ có vài file trên GitHub tải về sẽ bị chặn nên cần phải tắt tường lửa.

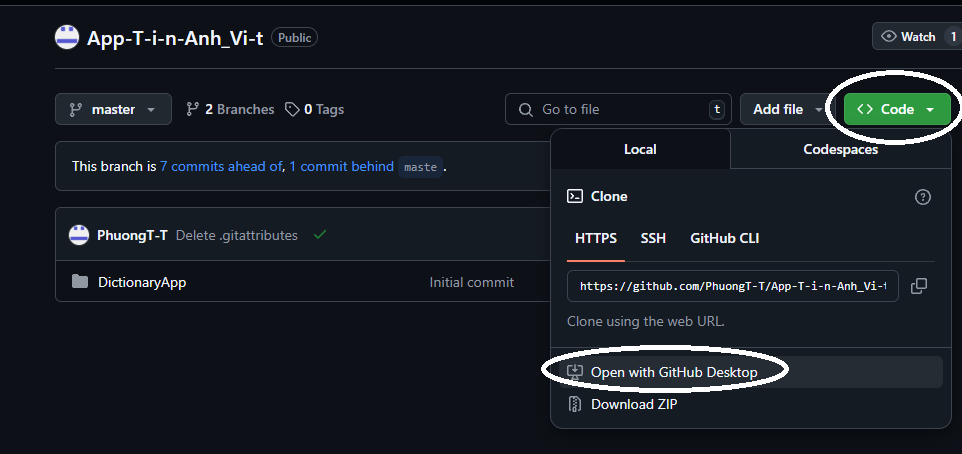
=> Mở Windows tìm Windows Defender Firewall, Nhấp vào Turn Windows Defender Firewall on or off (Bật hoặc tắt tường lửa Windows). – Chọn Turn off Windows Defender Firewall cho cả hai mục Private network settings và Public network settings.



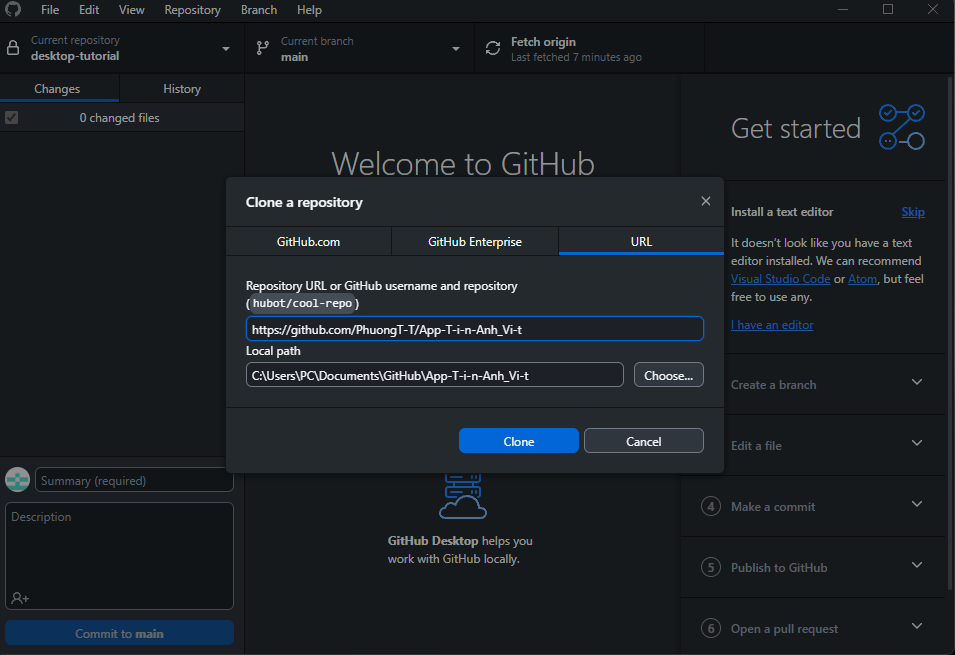


**Bước 2**: Truy cập đường dẫn: [**https://github.com/PhuongT-T/App-T-i-n-Anh\_Vi-t/tree/master**](https://github.com/PhuongT-T/App-T-i-n-Anh_Vi-t/tree/master)

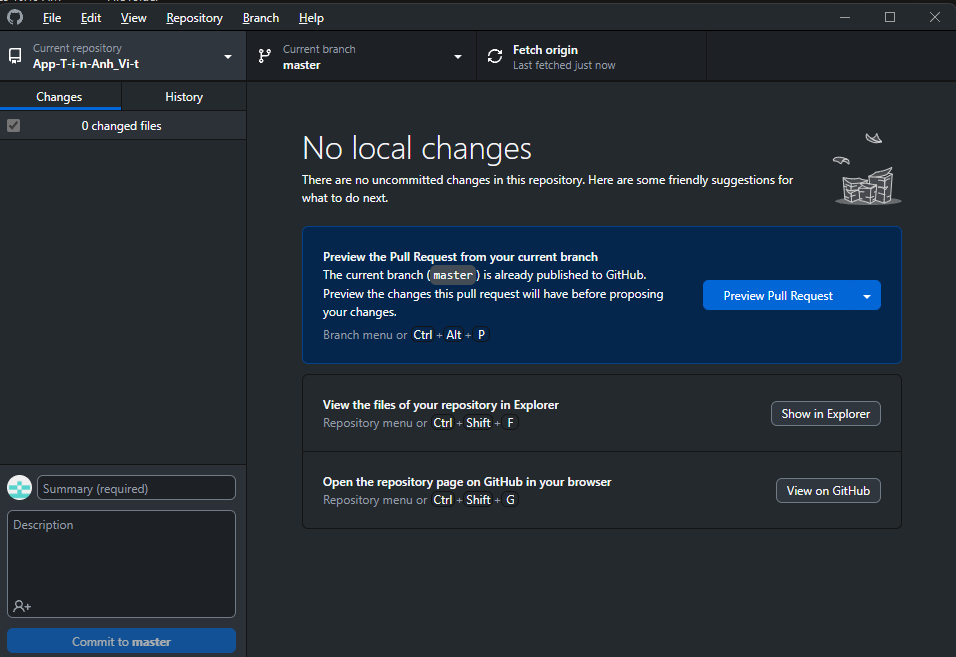
**Bước 3**: Click vào “Code” sau đó click vào “Open with GitHub Desktop”   
( Mọi người nên tải GitHub Desktop và có tài khoản GitHub. Link tải GitHub Desktop:<https://desktop.github.com/> ).



**Bước 4**: Sau khi click vào trang sẽ đưa bạn vào GitHub Desktop, sau đó click vào “Clone”.



**Bước 5**: Cuối cùng click vào ”Show in Explorer” sẽ hiện file của bọn mình ra.



Lưu ý: Vì vì code trên có địa chỉ file từ điển nằm trên máy tính bạn viết code nên người dùng muốn sử dụng app phải tạo 1 file từ điển trên Notepad và địa chỉ file đó thay thế vào đoạn code sau ở class form1:



**II. Hướng dẫn sử dụng ứng dụng**

1. **Khởi chạy ứng dụng**
2. Mở file thực thi hoặc chạy từ Visual Studio thông qua Program.cs.
3. **Tra từ**
4. Nhập từ tiếng Anh vào ô tìm kiếm.
5. Bấm “Tìm kiếm”.
6. Nghĩa tiếng Việt của từ sẽ hiển thị ở phần kết quả.
7. **Xử lý không tìm thấy**
8. Nếu từ không có trong cơ sở dữ liệu, ứng dụng sẽ thông báo “Không tìm thấy từ”.
9. **Thoát ứng dụng**
10. Đóng cửa sổ chính (Form1) để kết thúc chương trình.

**Bảng phân công công việc:**

|  |  |
| --- | --- |
| Thành viên | Nhiệm vụ |
| Trương Thị Kiều Phương (Leader) | Định hướng bài làm  Code chính |
| Đoàn Thị Mai Linh | Viết báo cáo |
| Bùi Thị Thùy Linh | Viết báo cáo |
| Bùi Anh Minh | Viết báo cáo |

# **TÀI LIỆU THAM KHẢO**

1. Bài giảng của thầy Đặng Ngọc Hoàng Thành

2. Howkteam.com

3. Github.com