TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC MÁY TÍNH

MÔN HỌC

NGÔN NGỮ LẬP TRÌNH C#

Báo cáo đồ án môn học

ỨNG DỤNG SO SÁNH MÃ NGUỒN

2COMPARE

GVHD: Phạm Nguyễn Trường An**.**

SVTH – Nhóm 2:

1. Nguyễn Trần Minh Tân – 13520747 (NT).
2. Lê Thị Tuyết Mai – 13520489.
3. Đặng Anh Khoa – 13520402.
4. Phạm Hoàng Thịnh – 14520885.

Ngày 7 tháng 1 năm 2016,

MỤC LỤC

Lời cảm ơn

Trước tiên nhóm em xin gửi lời cám ơn chân thành sâu sắc tới thầy Phạm Nguyễn Trường An, đã hướng dẫn nhóm em trong suốt quá trình làm đồ án. Trong thời gian làm việc với thầy, nhóm em không ngừng tiếp thu thêm nhiều kiến thức bổ ích mà còn học tập được tinh thần làm việc, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả. Đây là những điều rất cần thiết cho nhóm em trong quá trình học tập và làm việc sau này.

Sau cùng xin gửi lời cảm ơn chân thành tới các thành viên trong nhóm đã tích cực đóng góp ý kiến và giúp đỡ nhau trong quá trình học tập, nghiên cứu và hoàn thành đồ án.

Giới thiệu đề tài

Mô tả đồ án

Đối với nhu cầu làm việc nhóm hiện nay, các công cụ như Git, SVN đã hỗ trợ các tổ chức rất nhiều trong việc quản lý dự án, phân chia công việc, … Tuy nhiên, đôi lúc sẽ xảy ra tình trạng conflict, nhiều người cùng làm một công việc theo những cách khác nhau, dẫn đến mã nguồn sẽ trở nên lộn xộn, khó tìm được phần hiệu quả nhất để đưa vào project chính.

Đáp ứng nhu cầu cần so sánh các phiên bản mã nguồn để đưa ra phiên bản hoàn thiện nhất, nhóm chọn đề tài xây dựng ứng dụng so sánh hai file mã nguồn, hai folder với các tuỳ chọn tối ưu để tiết kiệm thời gian cho các lập trình viên.

Việc so sánh hai file mã nguồn và đánh dấu sự khác nhau giữa chúng là rất cần thiết đối với lập trình viên khi có thể tiết kiệm được rất nhiều thời gian trong việc xem xét từng file. Các phần khác nhau sẽ được đánh dấu và hiển thị trực quan, dễ dàng sử dụng trong bất kỳ trường hợp nào.

Mục tiêu của đề tài

Xây dựng một công cụ so sánh và đồng bộ hoá mã nguồn của 2 file text, 2 folder tương đối giống nhau để người sử dụng nhận biết được sự khác nhau giữa chúng và chỉnh sửa cho phù hợp.

Đối tượng sử dụng

Công cụ này xuất phát từ nhu cầu của bản thân các thành viên trong nhóm, là những lập trình viên thường xuyên trao đổi mã nguồn với nhau qua github. Do đó đối tượng chính mà phần mềm hỗ trợ là các lập trình viên. Ngoài ra ứng dụng còn phù hợp với những đối tượng có nhu cầu so sánh và đối chiếu.

Giá trị thực tiễn

Trên thị trường hiện nay đã tồn tại khá nhiều sản phẩm có chức năng tương tự với mức độ hoàn thiện rất cao. Nhóm em không có ý định xây dựng một sản phẩm giải pháp thay thế các ứng dụng như thế, mà dừng lại ở mức độ học tập và nghiên cứu, nhằm phát triển kĩ thuật, khả năng tư duy, làm việc nhóm và tuân thủ nội quy nhóm giữa các thành viên. Ngoài ra đây cũng là một món quà đánh dấu sự nỗ lực của các thành viên trong nhóm.

Thiết kế và giao diện chương trình

Sơ đồ hệ thống

Chương trình bao gồm 2 chức năng riêng biệt là so sánh File và so sánh Folder.

Về so sánh File, bao gồm các module sau:

- TextController: Quản lý và điều khiển nội dung của các textbox hiển thị.
- TextLine: cấu trúc của 1 dòng được quản lý bởi controller. Mỗi TextLine bao gồm content (nội dung), hashCode (chuỗi mã hoá dùng cho việc so sánh) và ignoredLine (số dòng trống tiếp theo trong textbox, hỗ trợ việc canh chỉnh và so khớp dữ liệu).
- FileCompareUtils: Tập hợp những hàm hỗ trợ cho việc so sánh file. Các hàm được hiện thực dưới dạng Static, có thể truy cập ở bất cứ vị trí nào trong chương trình.
- FileCompareEngine: bộ máy chính thực hiện việc so sánh 2 file và trả về kết quả. Dữ liệu đầu vào để so sánh là 2 TextController của 2 textbox. Hàm GetResult thuộc engine sẽ trả về kết quả là một ArrayList – DifferentResult.
- FileCompareEngineHelper: Các hàm, lớp hỗ trợ việc so sánh 2 file.
- RichTextboxAdvanced: kế thừa từ RichTextbox, hỗ trợ thêm khả năng đồng bộ hoá thanh cuộn của 2 textbox.

Về so sánh Folder, toàn bộ source code được hiện thực trong file MainForm.cs, không sử dụng các cấu trúc hỗ trợ. Các hàm chính như sau:

- show_folder: liệt kê và hiển thị tất cả các file và folder có trong 1 đường dẫn cho trước. Kết quả thu được sẽ được cập nhật vào TreeNode.
- show_treenode: hiển thị file và folder dưới dạng cây thư mục.
- to_mau: hiển thị trạng thái khác nhau giữa hai folder/file.
- compare: so sánh hai cây thư mục.

Ngoài hai chức năng chinh nêu trên, chương trình còn hỗ trợ thêm chức năng hiển thị lịch sử truy cập. Người sử dụng có thể xem thêm tại thẻ History trong chương trình. Ngoài ra, chương trình được xây dựng dựa trên giao diện MetroFramework, giao diện khá dễ nhìn và thân thiện.

Công cụ phát triển

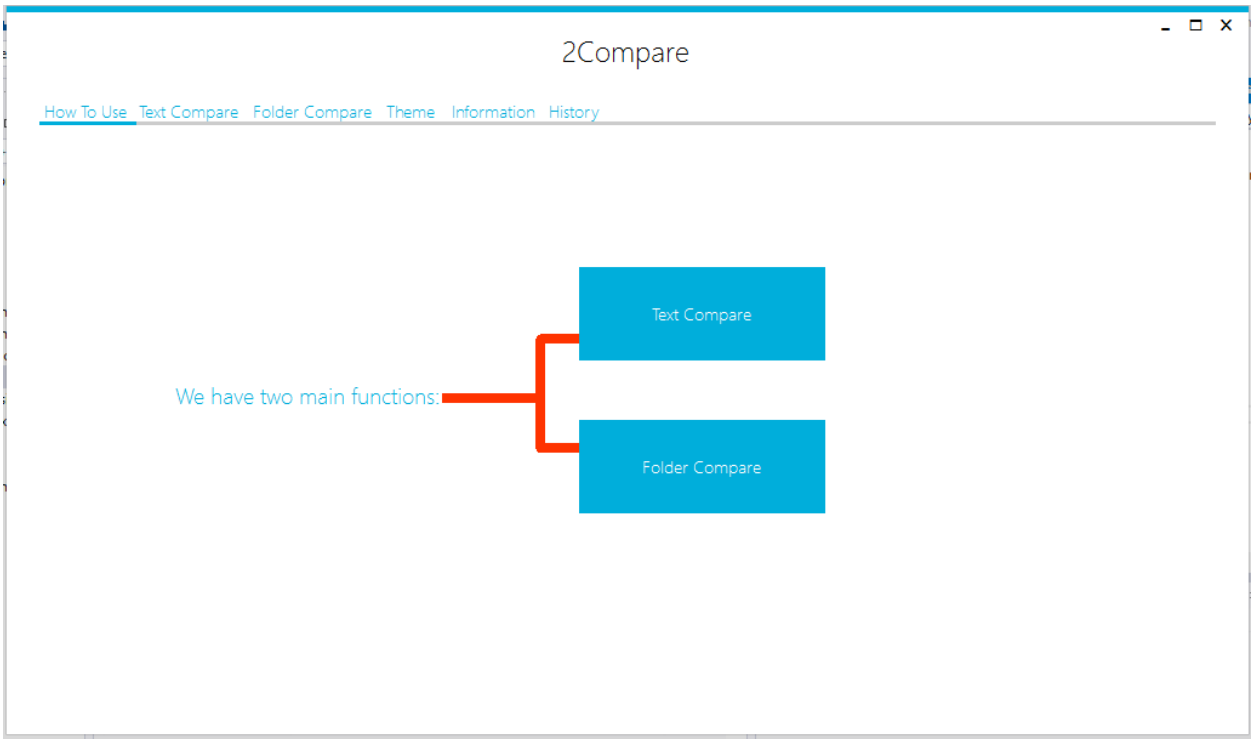
- Ngôn ngữ lập trình C#.
- IDE sử dụng là Visual Studio 2013 Community.
- Framework xây dựng giao diện: MetroFramework.

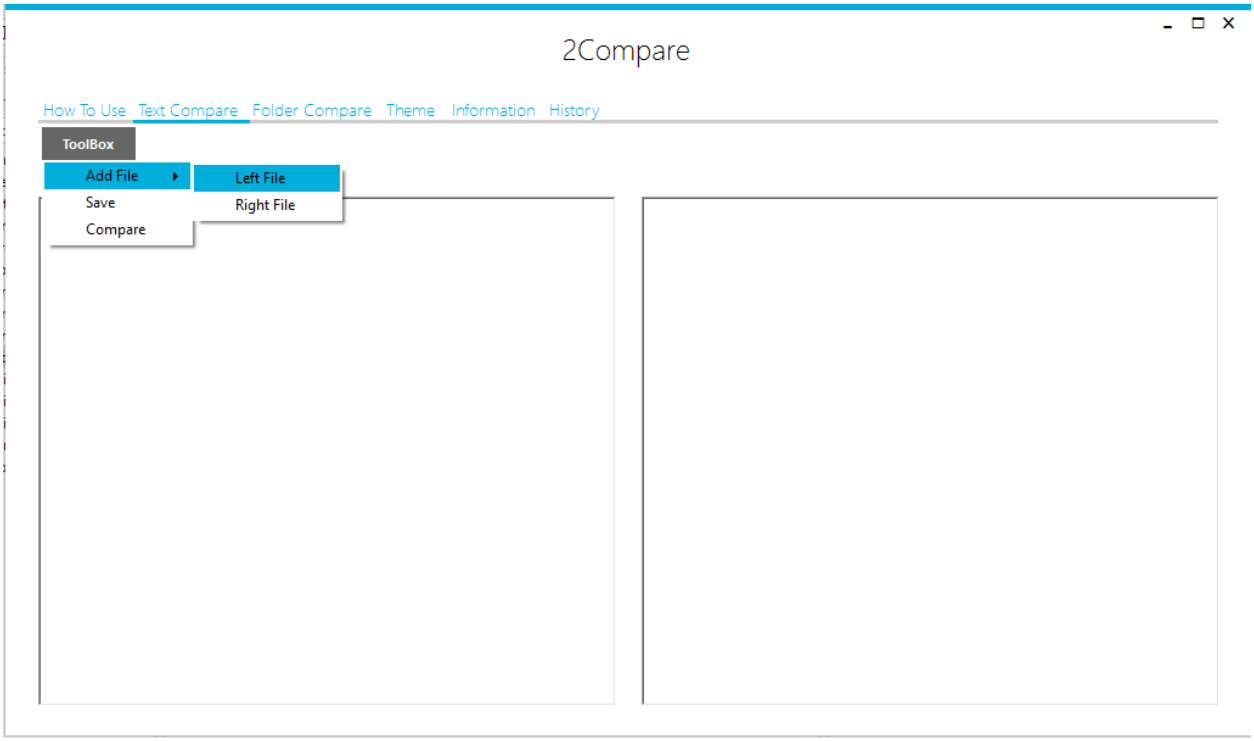
Mô tả chương trình

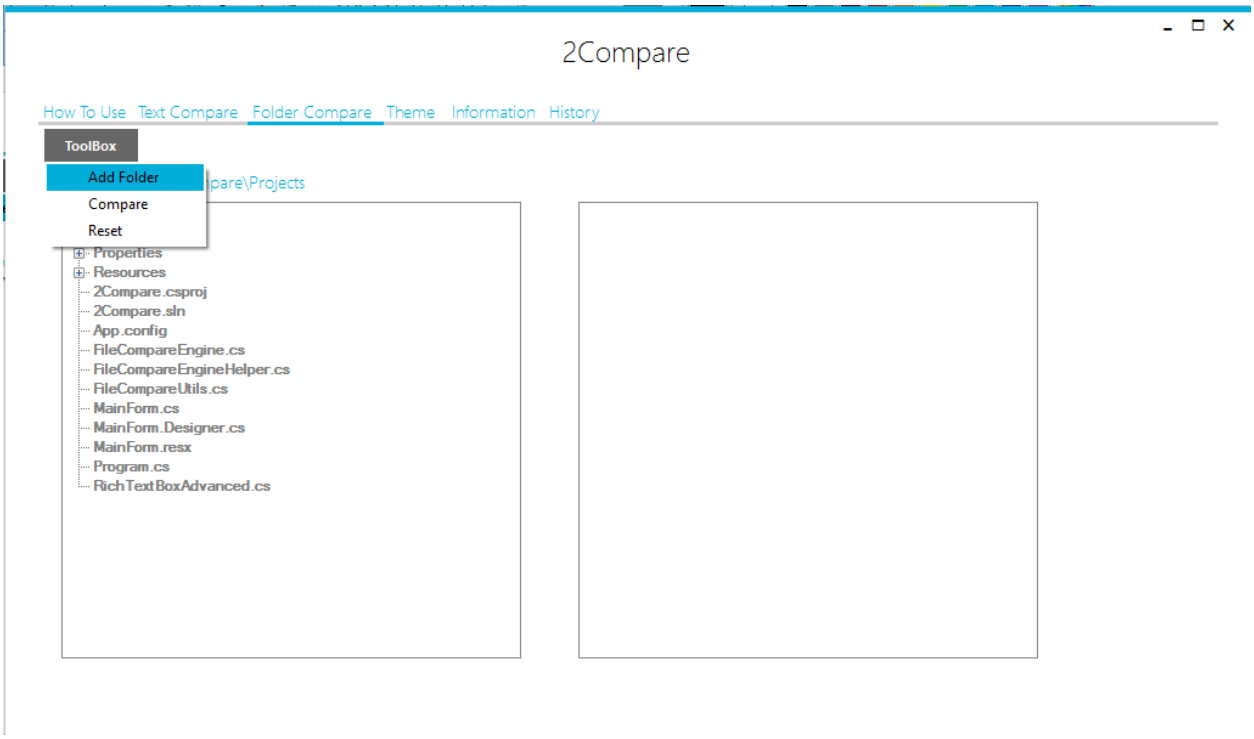
Giao diện chương trình bao gồm 6 tab, bao gồm:

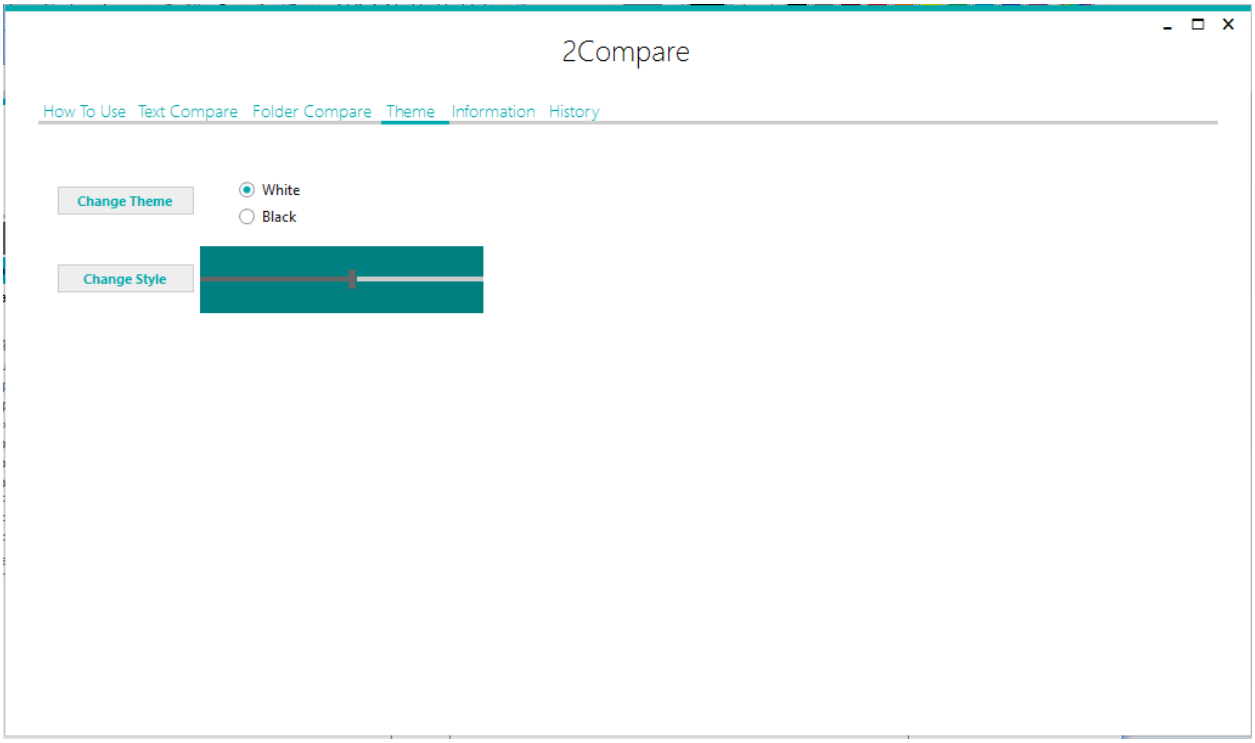
- How To Use: Liệt kê các chức năng chính của chương trình.
- Text Compare: Chức năng so sánh 2 file text.
- Folder Compare: Chức năng so sánh 2 folder.
- Theme: Các tuỳ chọn thay đổi giao diện chương trình.
- Information: Thông tin các thành viên của nhóm phát triển.
- History: Lịch sử sử dụng chương trình.

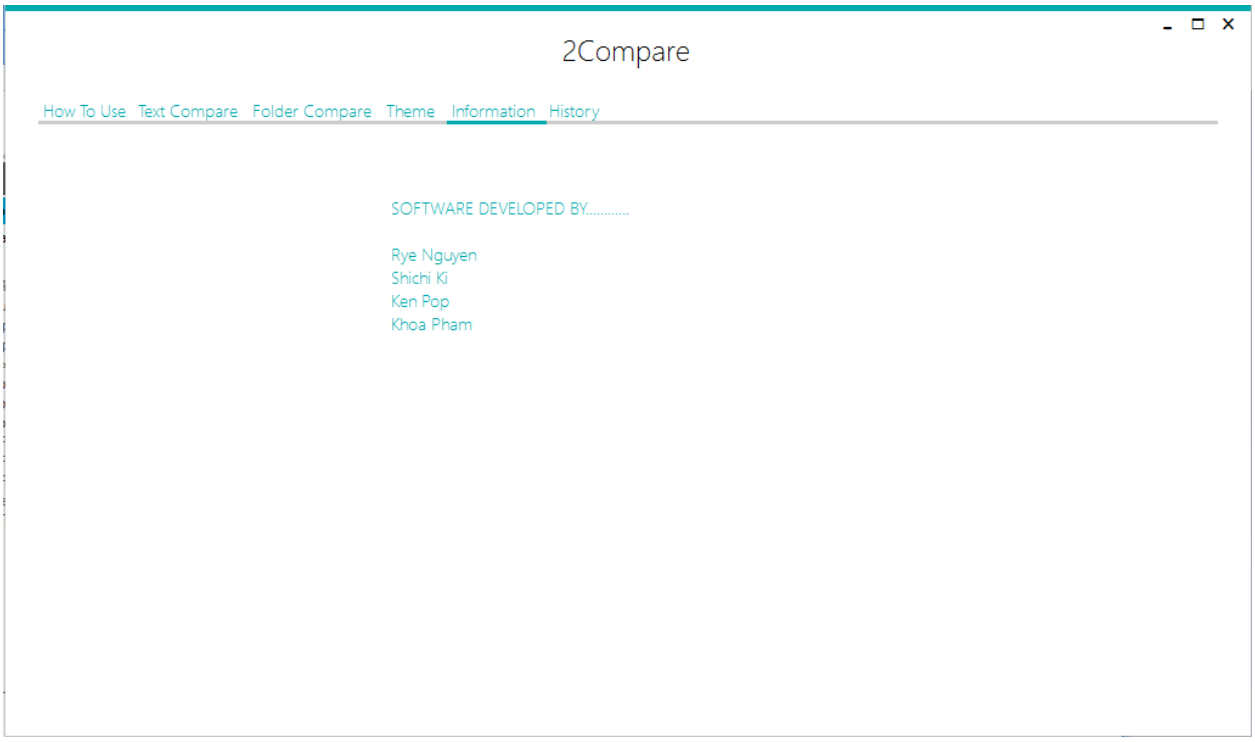
Một số hình ảnh tổng quan về giao diện chương trình:

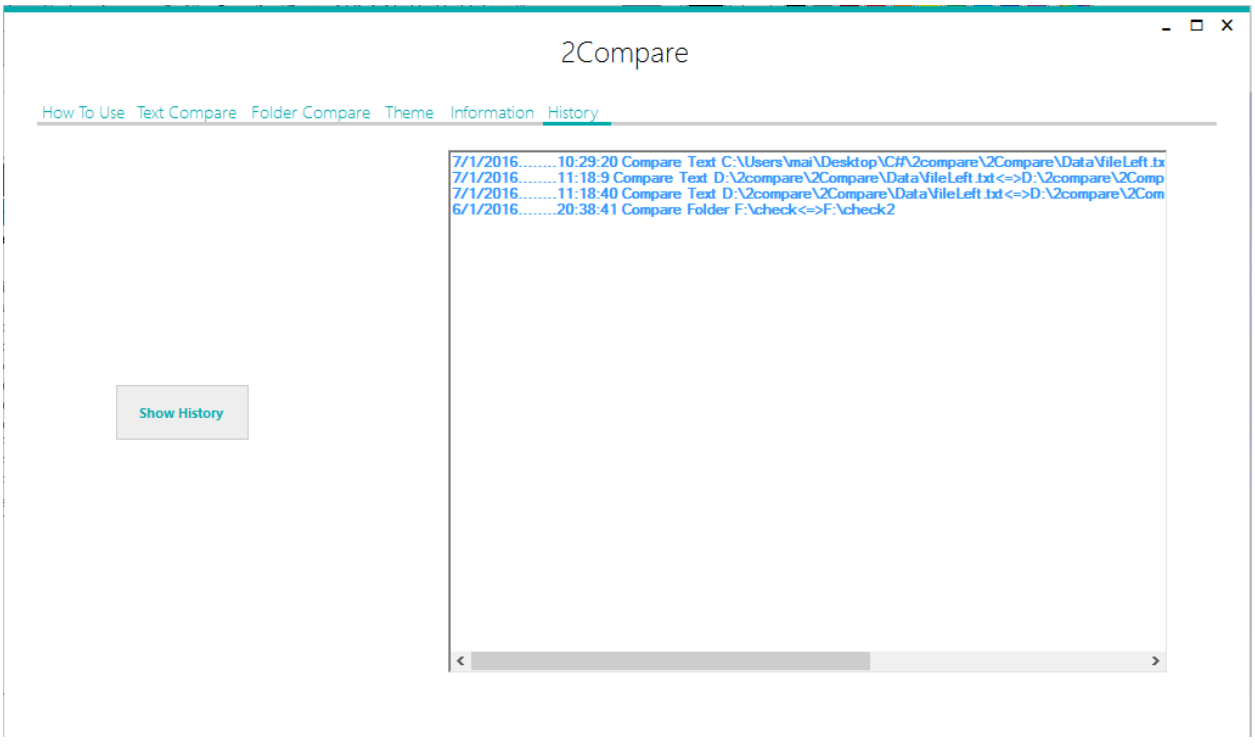





2Compare
How To Use
Text Compare
Folder Compare
Theme
Information
History
ToolBox
Add Folder
Compare
Reset
pare\Projects
Properties
Resources
2Compare.csproj
2Compare.sln
App.config
FileCompareEngine.cs
FileCompareEngineHelper.cs
FileCompareUtils.cs
MainForm.cs
MainForm.Designer.cs
MainForm.resx
Program.cs
RichTextBoxAdvanced.cs

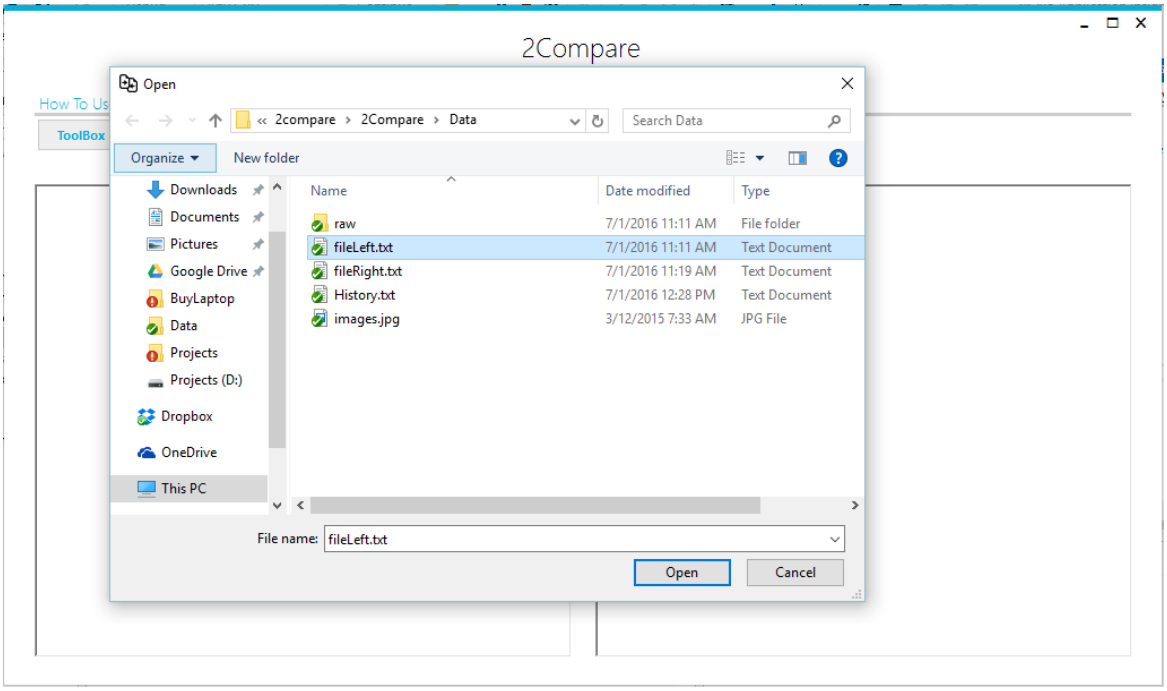

2Compare
How To Use
Text Compare
Folder Compare
Theme
Information
History
Change Theme
White
Black
Change Style


2Compare
How To Use
Text Compare
Folder Compare
Theme
Information
History
SOFTWARE DEVELOPED BY.....
Rye Nguyen
Shichi Ki
Ken Pop
Khoa Pham

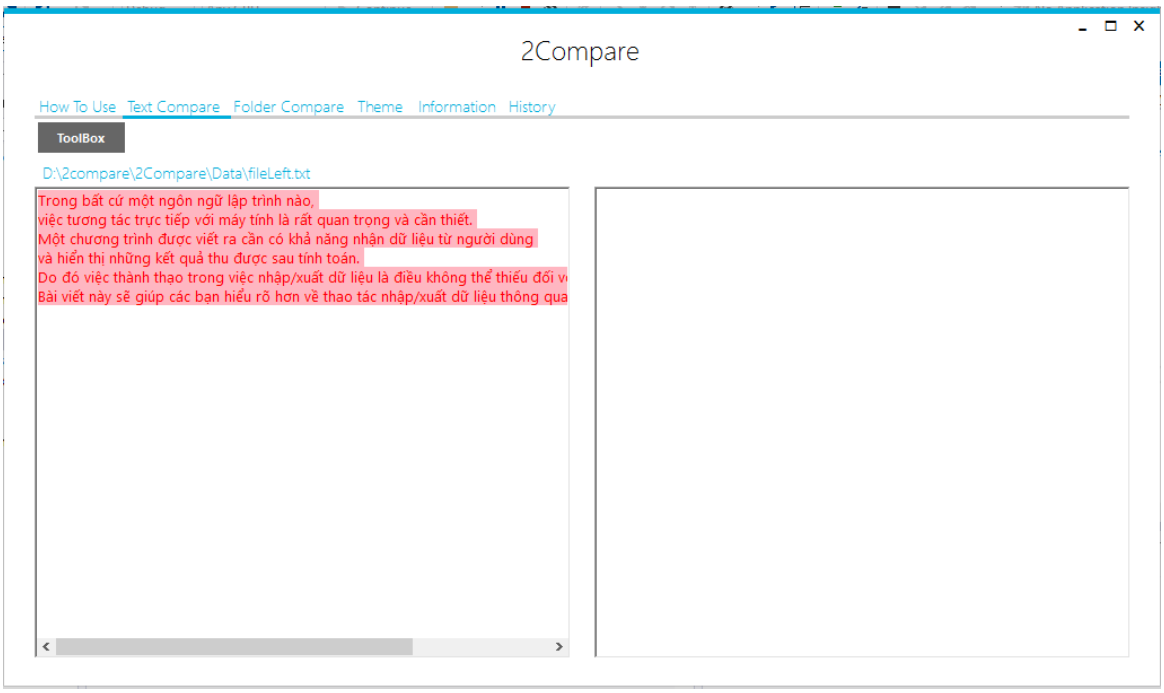

2Compare
How To Use
Text Compare
Folder Compare
Theme
Information
History
Show History
7/1/2016.....10:29:20 Compare Text C:\Users\mai\Desktop\C#\2compare\2Compare\Data\fileLeft.tx
7/1/2016.....11:18:9 Compare Text D:\2compare\2Compare\Data\fileLeft.txt<=>D:\2compare\2Comp
7/1/2016.....11:18:40 Compare Text D:\2compare\2Compare\Data\fileLeft.txt<=>D:\2compare\2Com
6/1/2016.....20:38:41 Compare Folder F:\check<=>F:\check2

Use-case

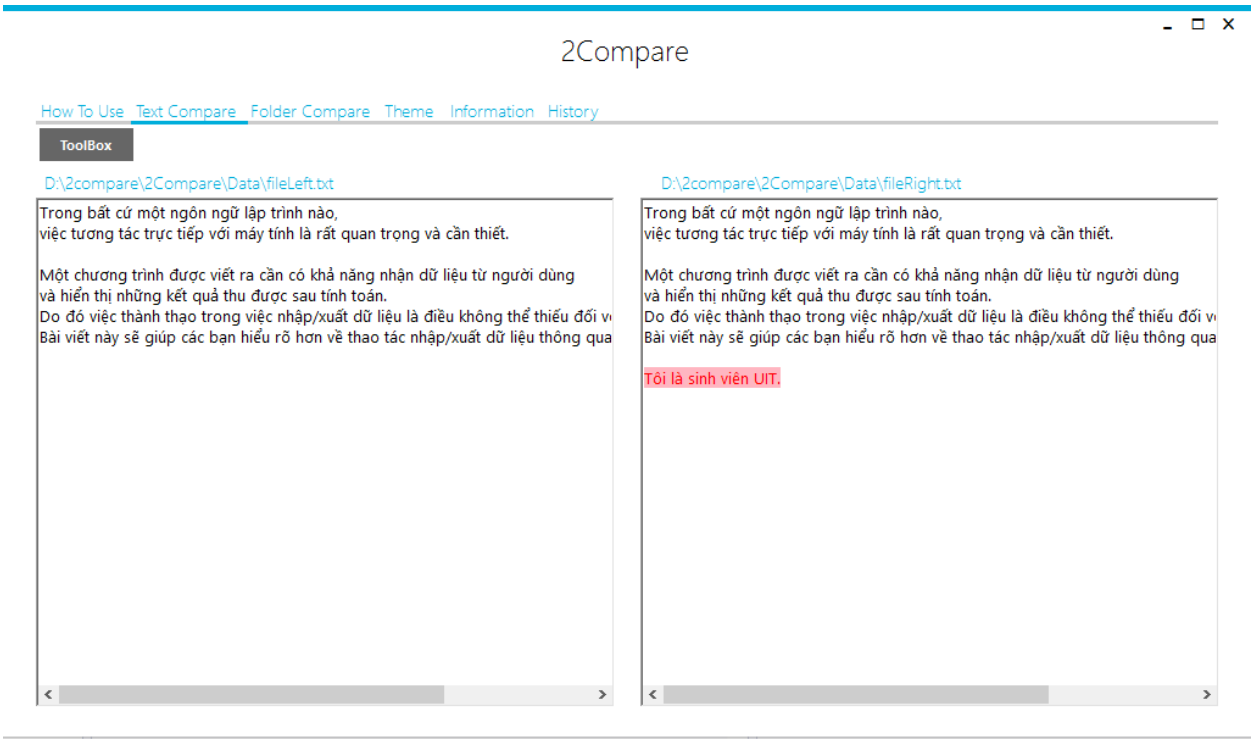
Đối với Text Compare, khi người dùng chọn chức năng Add File (Left hoặc Right), một hộp thoại mở lên yêu cầu người dùng chọn file cần đọc để load lên textbox.



Sau khi lựa chọn được file, bấm Open và chương trình sẽ tiến hành so sánh tự động, bất kể file còn lại đã được load hay chưa.



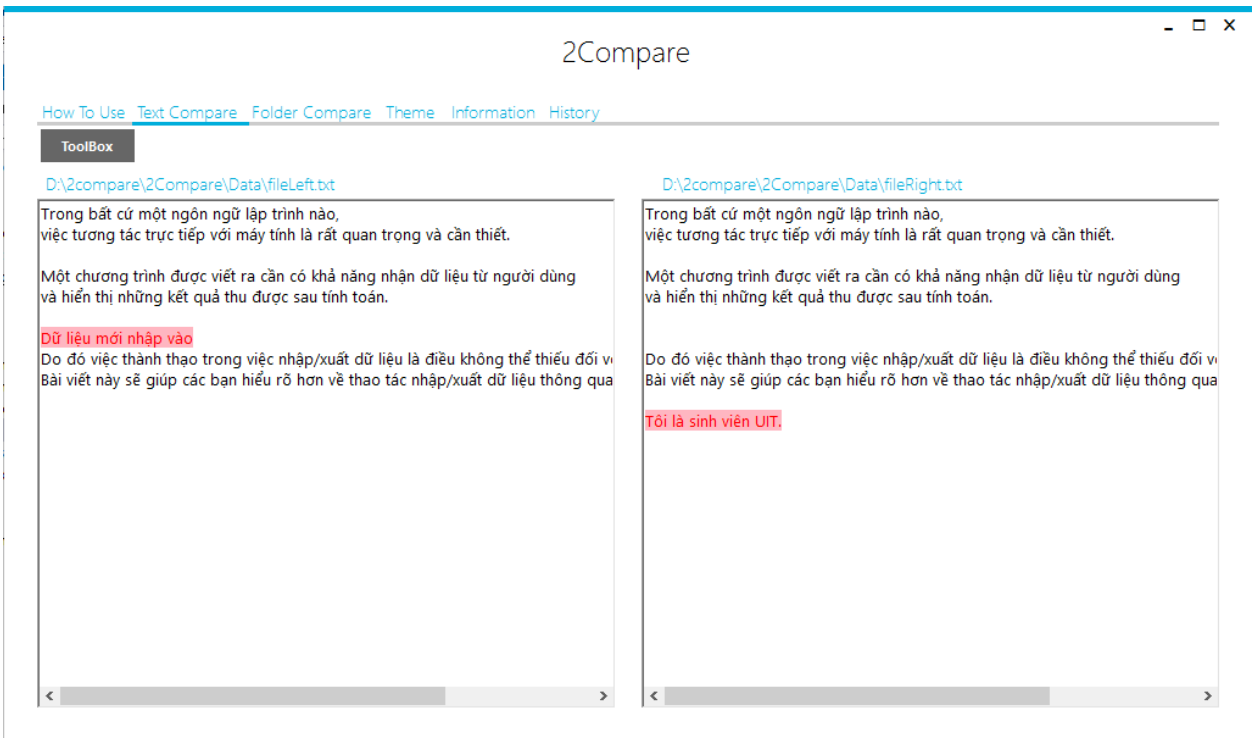
Sau khi load đủ 2 file, chương trình sẽ tự động so sánh lại một lần nữa. Lần so sánh này mới có giá trị khi người dùng có thể thấy được sự khác nhau giữa 2 file một cách trực quan.



TextBox sẽ tự động thêm các dòng trống để đẩy các thành phần giống nhau đến cùng một vị trí tương đối.

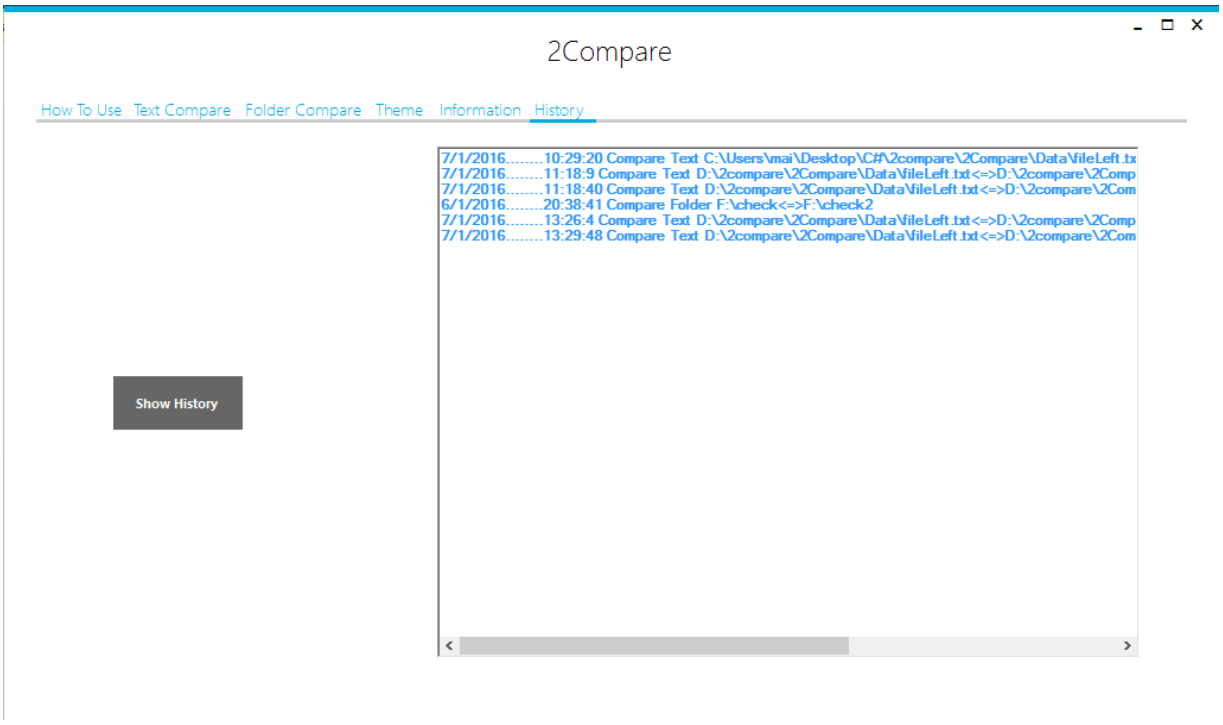
Với Text Compare, chương trình hỗ trợ các thao tác nhập dữ liệu bằng tay, enter, delete và backspace để thao tác với textbox. Các dữ liệu này sẽ được cập nhật realtime vào controller tương ứng.

Sau khi thay đổi dữ liệu, người dùng nhấn chọn nút ToolBox -> Compare để tiến hành so sánh lại. Chương trình chưa hỗ trợ chức năng cập nhật realtime cho textbox, do đó người dùng sử dụng button Compare để tái so sánh.



Về Folder Compare, các chức năng và giao diện cũng tương tự như TextCompare.

Với chức năng xem lịch sử, khi người dùng click vào button Show History, danh sách các lần so sánh trước đó sẽ được hiển thị. Danh sách này được lưu trữ trong file History.txt đính kèm



Vấn đề và giải pháp

Các vấn đề đã giải quyết được

Synchronized Scrolling

Đối với một ứng dụng đồng bộ, ngoài logic thì việc hiển thị cũng quan trọng không kém bởi điều đó giúp ta sử dụng công cụ một cách hiệu quả nhất. Việc đồng bộ thanh cuộn vì thế là một chức năng vô cùng quan trọng.

Để hiện thực tính năng này, nhóm đã xây dựng lớp đối tượng RichTextBoxAdvanced, mở rộng linh hoạt RichTextBox truyền thống.

File RichTextBoxAdvanced.cs:

```
public partial class RichTextBoxAdvanced : RichTextBox
{
	private const int WM_VSCROLL = 0x115;
	private const int WM_HSCROLL = 0x114;
	private const int WM_MOUSEWHEEL = 0x20a;

	// …

	public void BindScroll(RichTextBoxAdvanced arg)
	{
		if (peers.Contains(arg) || arg == this) { return; }
		peers.Add(arg);
		arg.BindScroll(this);
	}
}
```

Hiển thị các dòng tương ứng ở cùng một vị trí

Với 2 file text có nội dung khác nhau, việc so sánh và sắp xếp các dòng tương tự nhau vào cùng 1 vị trí sẽ giúp người dùng thao tác nhanh hơn.

Đối với công cụ RichTextBox thô sơ, nhóm em đã nảy ra ý tưởng chèn thêm 1 số dòng và quản lý bởi controller. Các dòng này được gọi là "ignoreLine" và được lưu trữ ở TextLine trước đó.

File FileCompareUtils.cs:

```
	public class TextLine : IComparable
	{
		public string m_content;
		public int m_hashCode;

		public int m_ignoredLine;

		// …

	}
```

File MainForm.cs, dòng 487 + 513:

```
((TextLine)_leftController.GetLineByIndex(_index)).m_ignoredLine = _item.Length;
((TextLine)_rightController.GetLineByIndex(_index)).m_ignoredLine = _item.Length;
```

Sử dụng thư viện hỗ trợ xây dựng giao diện

Giao diện của chương trình được xây dựng dựa trên thư viện MetroFramework, tích hợp trong source code của chương trình.

Các vấn đề chưa giải quyết được

History và khả năng linh hoạt

Mỗi dòng của History là 1 lần so sánh file hoặc folder. Chức năng click chọn và lặp lại lịch sử so sánh là chưa có sẵn. Tính năng này tuy hiện thực đơn giản, nhưng do giới hạn thời gian hoàn thiện đồ án nên nhóm em chưa hoàn thiện được.

Highlight một dòng trên RichTextBox và TreeView

Vấn đề này là khá khó khăn và tốn nhiều thời gian nghiên cứu. Nhóm em vẫn chưa hoàn thiện được tính năng này, RichTextBox chỉ hỗ trợ highlight vùng có kí tự hiển thị nên việc highlight toàn bộ dòng là không khả thi, do đó cần xây dựng phần mở rộng.

So sánh sâu vào từng dòng

Giải thuật so sánh chuỗi của nhóm chỉ dừng lại đơn giản ở mức so sánh dòng. Việc đi sâu vào kí tự khá phức tạp và không khả thi với mức độ đồ án môn học.

Copy To

Các ứng dụng đồng bộ hoá hiện nay như Beyond Compare hay các công cụ online phần lớn đều có chức năng Copy To, tăng tính ứng dụng và ý nghĩa của việc sử dụng phần mềm. Tuy nhiên việc xây dựng khá phức tạp và tốn thời gian nên nhóm em chưa hoàn thiện được chức năng này.

Giải thuật và cài đặt

Text Compare

```
function Compare(destStart, destEnd, sourceStart, sourceEnd)
{
	// Setup for first matchList
	curBestIndex = 1, curBestLength = 1, maxPossible = 0

	for(destIndex = destStart; destIndex <= destEnd; destIndex++)
	{
		// Gán maxPossible là tối ưu nhất (có nhiều Line trùng nhau nhất và giảm dần
theo logic)

		// Tính toán trạng thái hiện tại: Matched, NotMatched
		// Tính toán độ dài lớn nhất của trạng thái (số lượng khác nhau hai bên)

		if(Matched)
		{
			if (curItem.Length > curBestLength)
			{
				curBestIndex = destIndex;
				curBestLength = curItem.Length;
				bestItem = curItem;
			}
			//Jump over the match
			destIndex += curItem.Length - 1;
		}
	}

	if(curBestIndex < 0)
	{
		// Không có đoạn trùng khớp
	}
	else
	{
		// Tạo 1 matchState và thêm vào matchList

		if(destStart < curBestIndex && sourceStart < sourceIndex)
		{
			Đệ quy Compare(destStart, curBestIndex - 1, sourceStart, sourceIndex -
1)
		}

		if(destEnd > curBestIndex + curBestLength && sourceEnd > sourceIndex +
curBestLength)
		{
			Đệ quy Compare(curBestIndex + curBestLength, destEnd, sourceIndex +
curBestLength, sourceEnd)
		}
	}
}
```

Folder Compare

```
CompareFolder( folder a, foder b)
{
        i=0;
        bước 1: folder a.a[i];// foder con đầu tiên trong folder a;
        Bước 2: duyệt hết các folder của y
                If có foder a.a[0] trong y
                i++
        Bước 3: nếu có thì quay lại bước 1
        Bước 4 : nếu kết quả trả về false tức folder đó không tồn tại trong
}
```

Kết luận

- Trên cơ sở thực hiện được so sánh 2 file text, so sánh được 2 folder và một số chức năng cơ bản đã hoàn thiện.
- Còn nhiều điểm hạn chế khi so sánh với các ứng dụng tương tự.
- Hướng phát triển:
    - Đi sâu vào thuật giải tìm ra chỗ khác nhau.
    - Thực hiện so sánh trên nhiều file khác nhau tìm ra sự khác nhau.
    - Hiện thực thêm các chức năng nhỏ và tiện dụng.

Tài liệu tham khảo

[1] MetroFramework: https://github.com/thielj/MetroFramework

[2] A Generic, Reusable Diff Algorithm in C#: http://www.codeproject.com/Articles/6943/A-Generic-Reusable-Diff-Algorithm-in-C-II

[3] TextBoxSynchronizedScroll: https://gist.github.com/jkingry/593809

[4] Color different parts of a RichTextBox string: http://stackoverflow.com/questions/1926264/color-different-parts-of-a-richtextbox-string

[5] RichTextBox references: https://msdn.microsoft.com/en-us/library/system.windows.forms.richtextbox%28v=vs.110%29.aspx

Phụ lục

Video clip demo sử dụng chương trình

https://www.youtube.com/watch?v=HSCtmVc22gU

Quá trình làm việc nhóm

Trong quá trình làm việc nhóm, nhóm chúng em có sử dụng một số công cụ để trao đổi tài liệu, mã nguồn cũng như chia sẻ kĩ thuật.

- Facebook chat, Skype: Chia sẻ tài liệu, các file báo cáo, … Group còn là nơi tán gẫu khi rảnh rỗi.
- Github: quản lý mã nguồn của project.

Link github: https://github.com/uit-cs-level1/2compare.

Các contributor:

- Nguyễn Trần Minh Tân – ryenguyen7411.
- Lê Thị Tuyết Mai – shichiki.
- Phạm Hoàng Thịnh – kenpop.
- Đặng Anh Khoa – Jameshana.