****

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**🕯✡🕮🕮✡🕯**

**MÔN HỌC: ĐỒ ÁN 1**



**BÁO CÁO**

**CHỦ ĐỀ: Chương Trình Soạn Thảo Văn Bản**

**GVHD: Tử Tuyết Hồng**

**SVTH: MSSV**

**Nguyễn Hoàng Duy 16110037**

**Tp. Hồ Chí Minh, tháng 11 năm 2019**

**ĐIỂM SỐ**

|  |  |  |  |
| --- | --- | --- | --- |
| **TIÊU CHÍ** | **NỘI DUNG** | **TRÌNH BÀY** | **TỔNG** |
| **ĐIỂM** |  |  |  |

**NHẬN XÉT**

*Ký tên*

**Tử Tuyết Hồng**

Mục lục

[**Chương 1: TỔNG QUAN** 4](#_Toc26487787)

[**I.** **Đặc tả hệ thống** 4](#_Toc26487788)

[**1.** **Các chức năng** 4](#_Toc26487789)

[**1.1.** **Soạn thảo** 4](#_Toc26487790)

[**1.2.** **Tìm kiếm** 4](#_Toc26487791)

[**1.3.** **Lưu file, Mở file, New File** 4](#_Toc26487792)

[**2.** **Phân tích thực thể** 4](#_Toc26487793)

[**II.** **Yêu cầu:** 6](#_Toc26487794)

[**III.** **Mục tiêu** 6](#_Toc26487795)

[**Chương 2: NỘI DUNG** 6](#_Toc26487796)

[**I.** **KẾ HOẠCH THỰC HIỆN DỰ KIẾN** 6](#_Toc26487797)

[**II.** **PHÂN CÔNG NHIỆM VỤ** 7](#_Toc26487798)

[**III.** **MÔ TẢ CHỨC NĂNG** 8](#_Toc26487799)

[**IV.** **Thiết kế giao diện** 9](#_Toc26487800)

[**1.** **Giao diện soạn thảo.** 9](#_Toc26487801)

[**2.** **Giao diện Hướng dẫn phím tắt** 9](#_Toc26487802)

[**3.** **Giao diện giới thiệu thông tin nhóm và giáo viên** 10](#_Toc26487803)

[**4.** **Giao diện dùng các chức năng tương tác file** 11](#_Toc26487804)

[**V.** **Các mẫu dùng cho thiết kế code** 15](#_Toc26487805)

[**1.** **Các lớp trong chương trình** 15](#_Toc26487806)

[**2.** **Tài liệu tham khảo:** 44](#_Toc26487807)

# **Chương 1: TỔNG QUAN**

## **Đặc tả hệ thống**

Chương trình soạn thảo văn bản trong đồ án được giả định gồm các yêu cầu sau:

### **Các chức năng**

* 1. **Soạn thảo**
* Chương trình cho phép thực hiện các chức năng soạn thảo cơ bản :
* In đậm, In nghiêng , Gạch chân.
* Thay đổi phông chữ.
* Căn lề : trái, phải, giữa.
* Thay đổi kích cỡ chữ
* Thay đổi màu chữ tùy ý.
  1. **Tìm kiếm**
* Chương trình cho phép tìm kiếm một chuỗi ký tự tùy ý có trong văn bản được viết trước
  1. **Lưu file, Mở file, New File**
* Chương trình cho phép lưu file đã soạn thảo văn bản dưới dạng Rich TextFile(\*rtf)
* Bạn cũng có thể mở file đã lưu sẵn trước đó.

1. **Phân tích thực thể**

Soạn Thảo

Mô tả màn hình Soạn thảo văn bản

Bao gồm các chức năng trong quá trình soạn thảo

**Bảng 1. Soạn thảo**

|  |  |  |
| --- | --- | --- |
| *Tên chức năng* | *Giải thích* | *Ghi chú* |
| Đọc file | Đọc file đã được lưu trước đó(\*rtf) |  |
| Lưu file | Lưu file mà đã soạn thảo dưới dạng Rich TextFile(\*rtf) |  |
| New File | Tạo File mới |  |
| Copy | Sao chép một chuỗi dữ liệu trong phần soạn thảo văn bản |  |
| Cut | Cắt một chuỗi dữ liệu trong phần soạn thảo văn bản |  |
| Undo | Trở về các thao tác trước đó |  |
| Redo | Trả lại những thao tác sau đó |  |
| In đậm | Cho phép in đậm chữ |  |
| Gạch chân | Định dạng chữ dưới dạng gạch chân |  |
| In nghiêng | Cho phép in nghiêng chữ |  |
| Canh giữa | Canh lề chữ ở giữa |  |
| Canh trái | Canh lề chữ ở bên trái của phần soạn thảo |  |
| Chèn hình | Cho phép chèn hình vào vùng soạn thảo văn bản |  |
| Chọn màu chữ | Chọn màu chữ tùy ý |  |
| Search | Cho tìm kiếm một hoặc nhiều chữ có trong phần soạn thảo |  |
| Size | Định dạng chữ với nhiều kích cỡ tùy ý |  |
| Chọn phông chữ | Chọn kiểu chữ tùy ý |  |

## **Yêu cầu:**

* Xây dựng chương trình bằng công cụ phần mềm
* Code: ngôn ngữ C#

## **Mục tiêu**

* Hoàn thành được phần mềm.
* Hoàn thành được giao diện soạn thảo.
* Hoàn thành được các chức năng soạn thảo cơ bản
* Hoàn thành được các chức năng tương tác với file

# **Chương 2: NỘI DUNG**

## **KẾ HOẠCH THỰC HIỆN DỰ KIẾN**

#### **Tuần 9:**

Đề xuất giao diện với các chức năng đã mô tả trong phần yêu cầu hệ thống

#### **Tuần 10:**

chuẩn bị các dữ liệu cần thiết để ode giao diện đã thiết kế từ tuần 5.

#### **Tuần 11:**

Hoàn thiện thiết kế lớp có cơ chế kiểm soát truy cập. Thiết kế giao diện mức ý niệm cho đề tài với các chức năng tương ứng.

Code tính năng + debug…

Viết báo cáo

#### **Tuần 12:**

Code form giao diện kết hợp với thao tác cơ bản hoàn thành bước đầu của đồ án.

Code tính năng + debug …

#### **Tuần 13:**

Code hoàn thiện các tính năng đã đề ra, hoàn thiện giao diện sử dụng cho người dùng + debug, …

Viết báo cáo

#### **Tuần 14:**

Debug và hoàn thành đồ án và nhận phản hồi của GV.

Viết báo cáo.

#### **Tuần 15:**

Tinh chỉnh và hoàn thành đồ án.

Hoàn thiện báo cáo.

#### **Tuần 16:**

GV chấm điểm, nhận xét.

## **PHÂN CÔNG NHIỆM VỤ**

Bảng 2. Phân công nhiệm vụ

|  |  |
| --- | --- |
| TT | Duy |
| Tuần 9 | Thiết kế Form cơ bản với các chức năng cơ bản |
| Tuần 10 | Tùy chỉnh Form cho đẹp với chèn các icon phù hợp với các chức năng |
| Tuần 11 | Hoàn thiện Form soạn thảo . Thiết kế giao diện mức ý niệm cho đề tài với các chức năng tương ứng.  Code tính năng + debug… |
| Tuần 12 | Viết báo cáo. |
| Tuần 13 | Code hoàn thiện các tính năng đã đề ra, hoàn thiện giao diện sử dụng cho người dùng + debug, … |
| Tuần 14 | Viết báo cáo. |
| Tuần 15 | Tinh chỉnh và hoàn thành đồ án.Hoàn thiện báo cáo. |
| Tuần 16 | GV chấm điểm, nhận xét. |

## **MÔ TẢ CHỨC NĂNG**

1. **Form Soạn Thảo**

|  |  |
| --- | --- |
| **Giới thiệu** | **Người Dùng** |
|  | Soạn thảo văn bản |
|  | Sử dụng các chức năng tùy ý  Lưu file.  Thoát. |

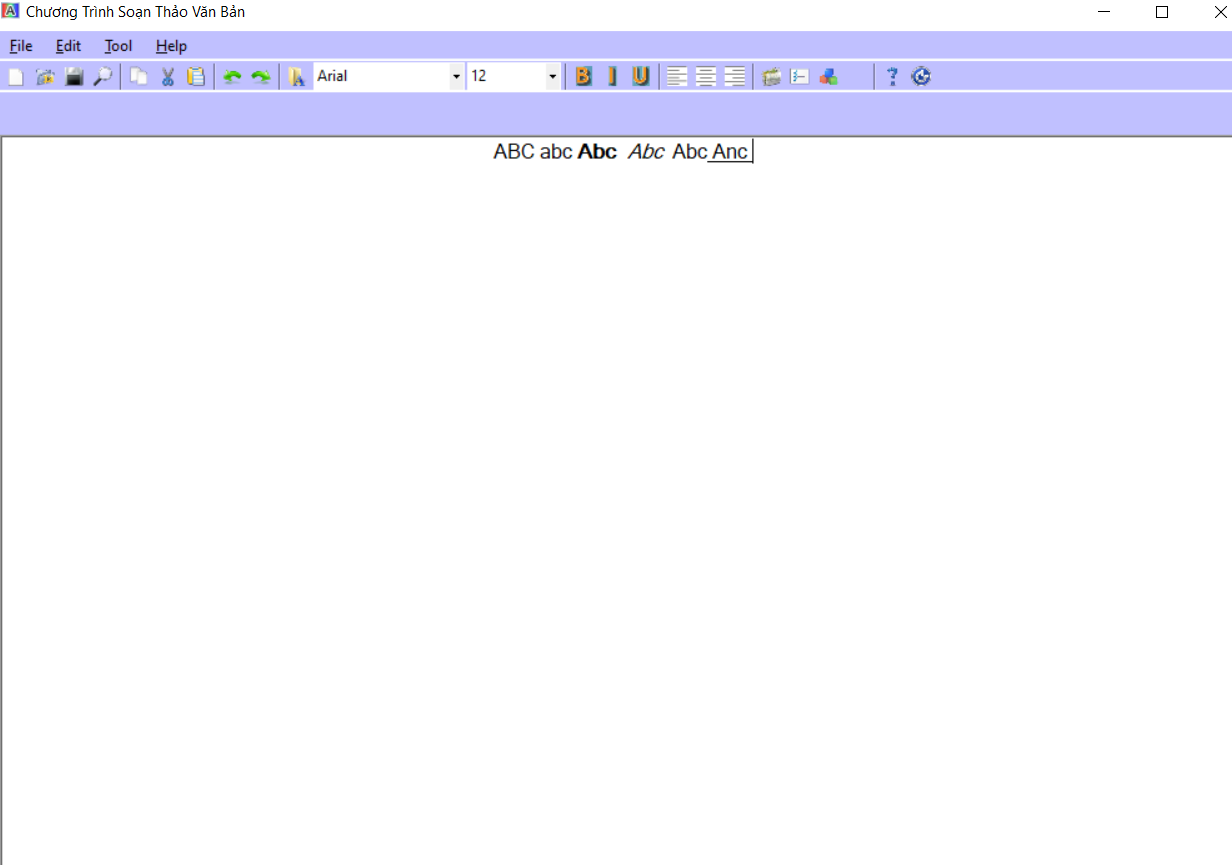
#### **Giới thiệu:**

Nói về thông tin đề tài và thông tin thành viên trong nhóm.

* 1. Chức năng soạn thảo
* Form soạn thảo gồm :
* Menu với các chức năng chính
* Các icon gồm các chức năng tương tác với vùng soạn thảo
  1. Form Hướng Dẫn
* Hướng dẫn các phím tắt được cài đặt trong chương trình cho phép việc soạn thảo văn bản một cách dễ dàng và thuận tiện hơn.
  1. Form Giới thiệu
* Giới thiệu thành viên và giáo viên hướng dẫn cùng với đề tài đã được giao.

## **Thiết kế giao diện**

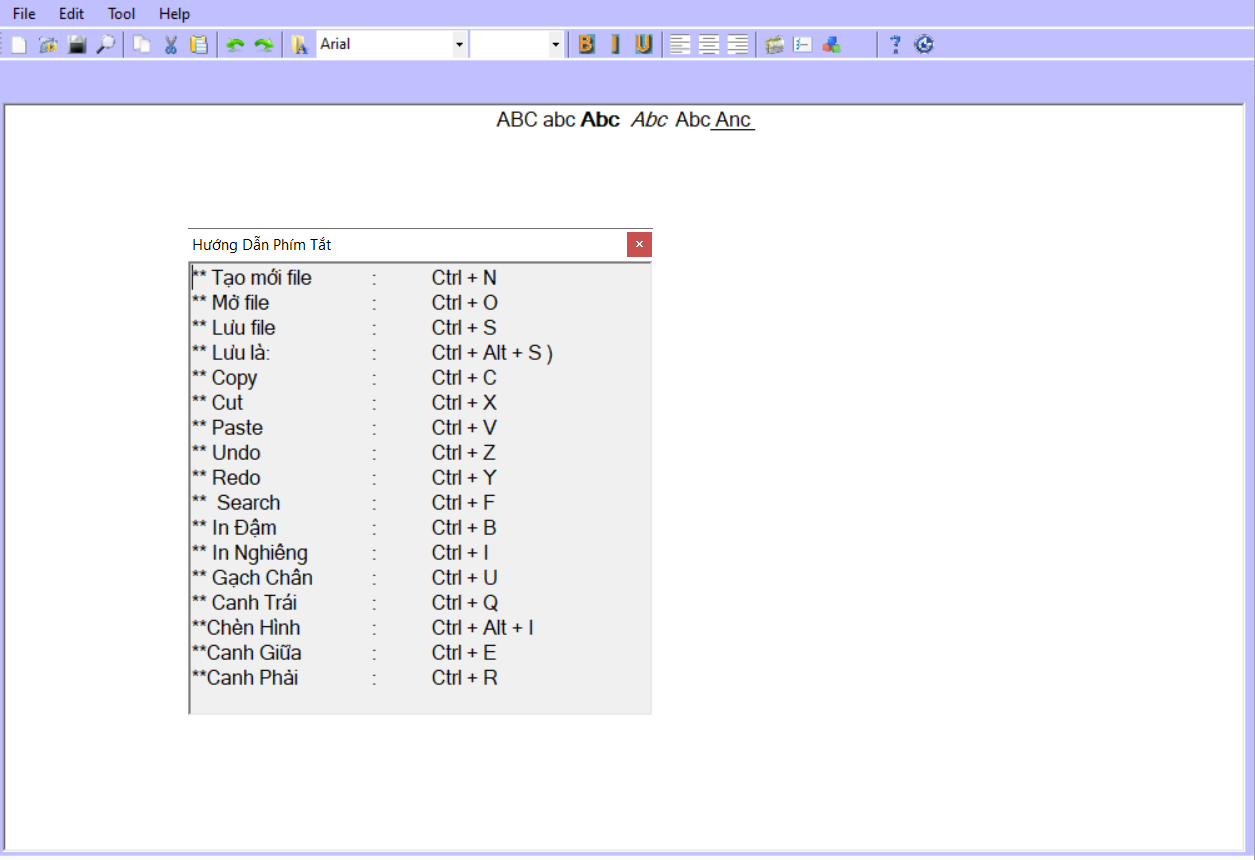
### **Giao diện soạn thảo.**



Hình 2. Giao diện soạn thảo

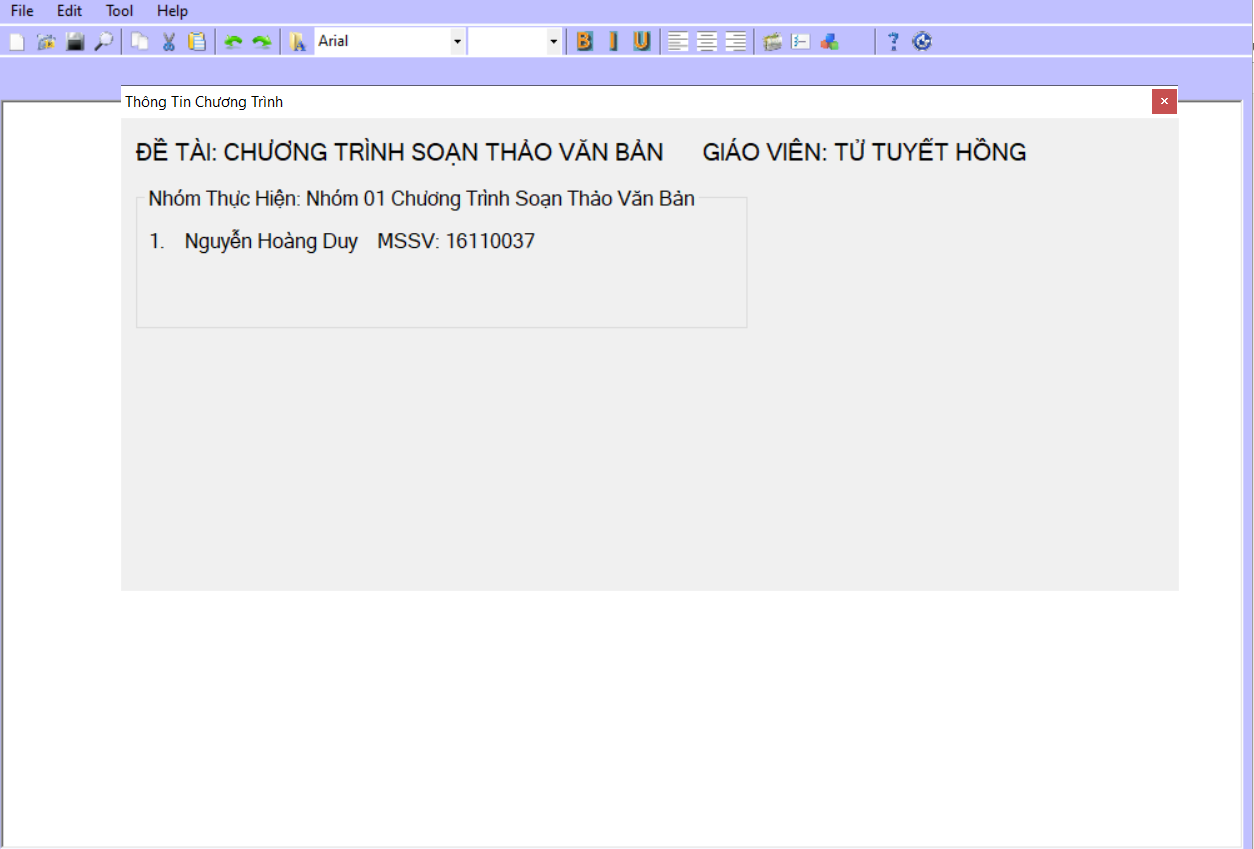
Người dùng có thể click vào các chức năng để định dạng văn bản theo mong muốn

1. **Giao diện Hướng dẫn phím tắt**



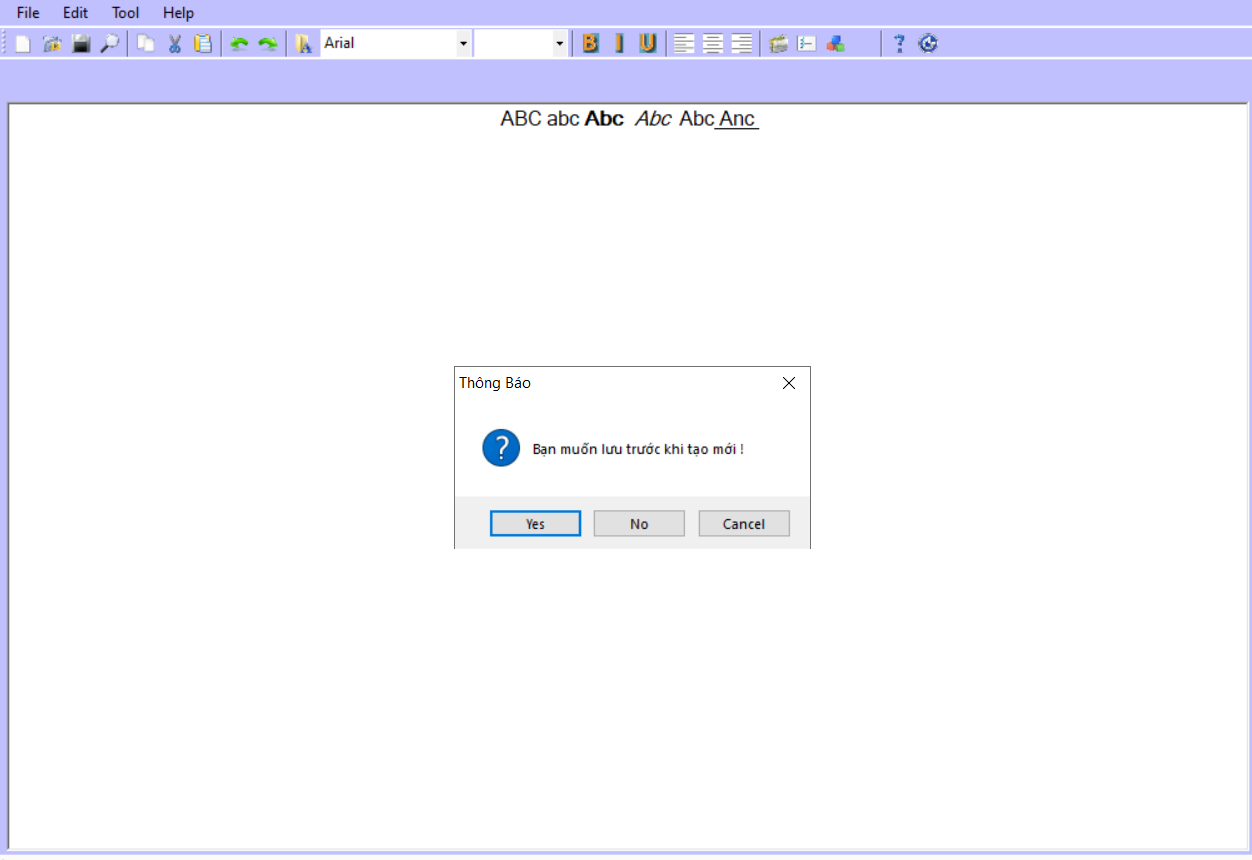
Hình 3. Giao diện Hướng dẫn phím tắt

1. **Giao diện giới thiệu thông tin nhóm và giáo viên**

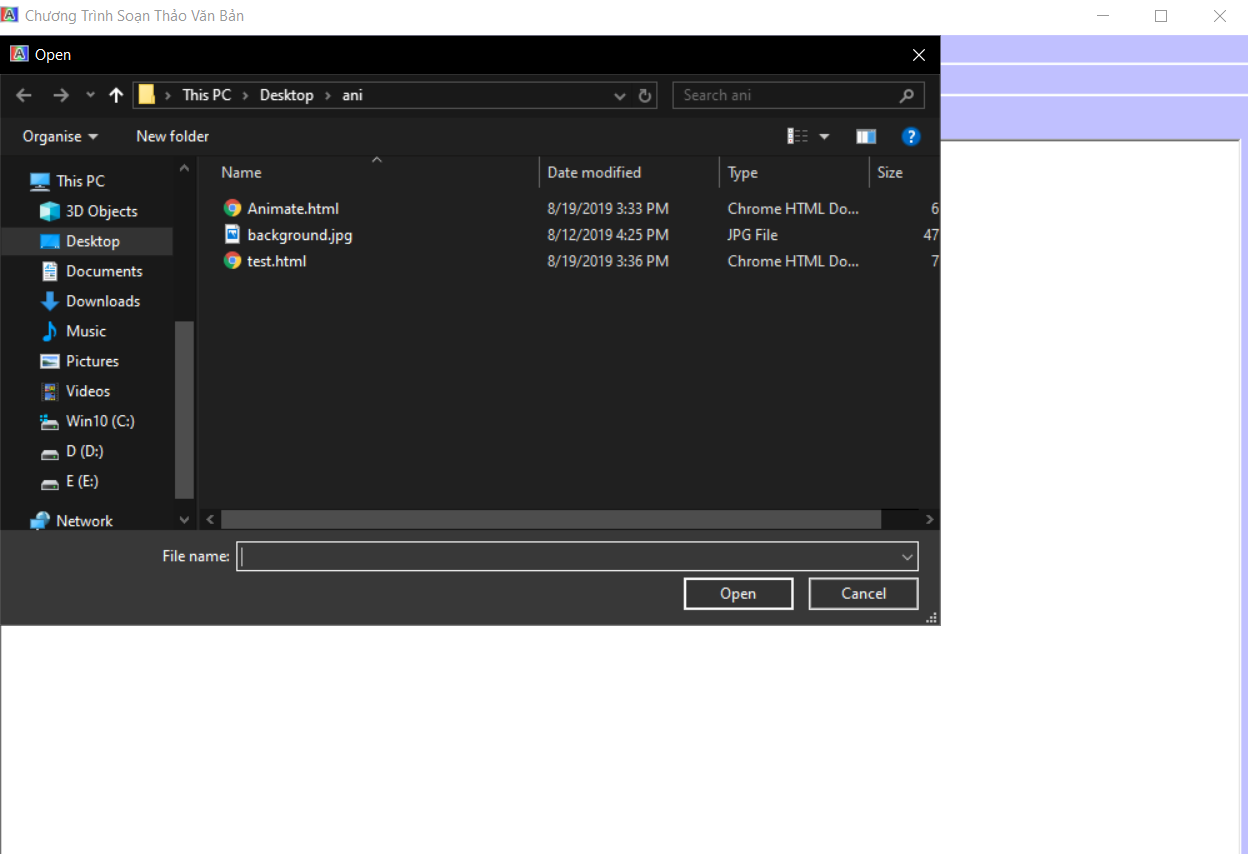


1. **Giao diện dùng các chức năng tương tác file**

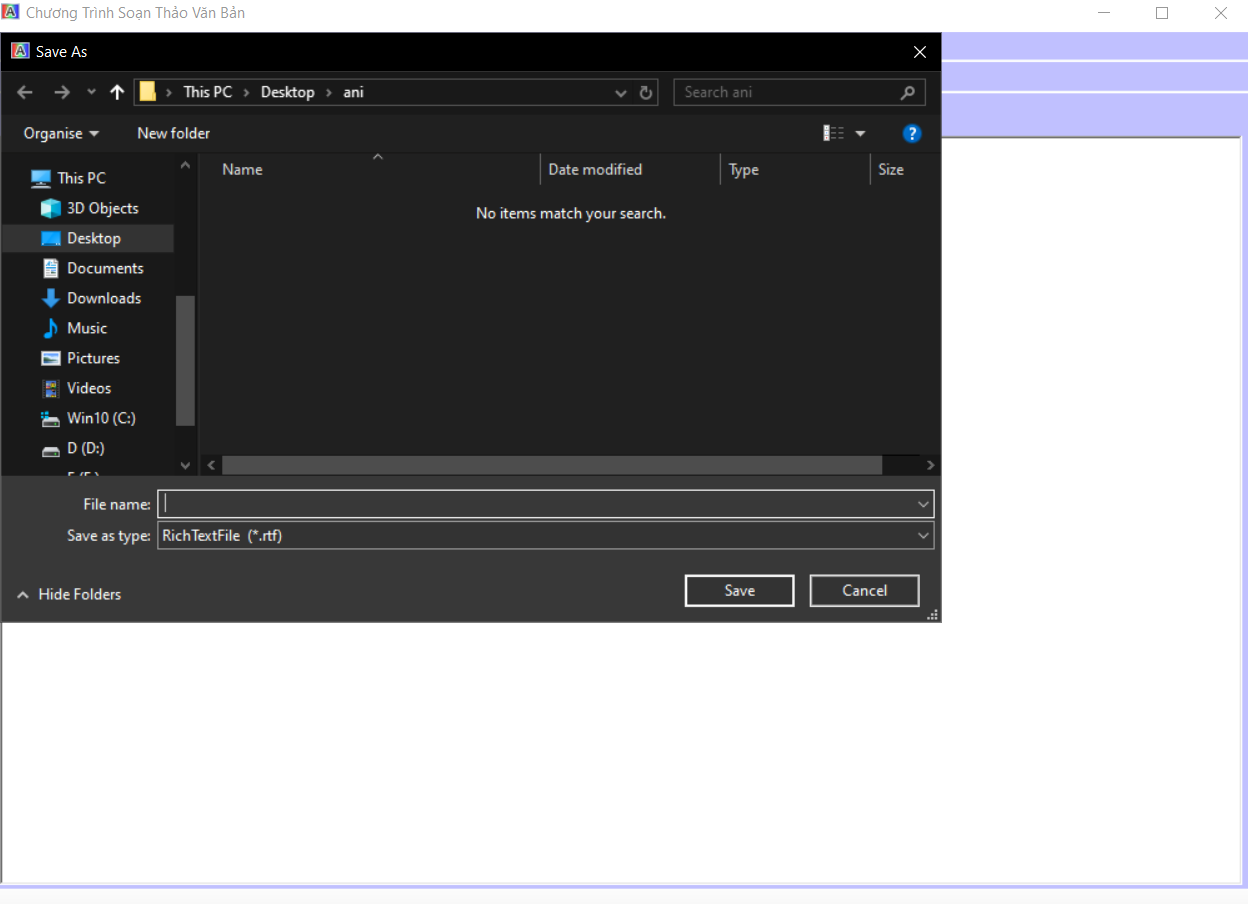
Sau khi soạn thảo bản có thể new file để chuyển sang file soạn thảo khác và bạn có thể lưu file đã soạn thảo trước đó



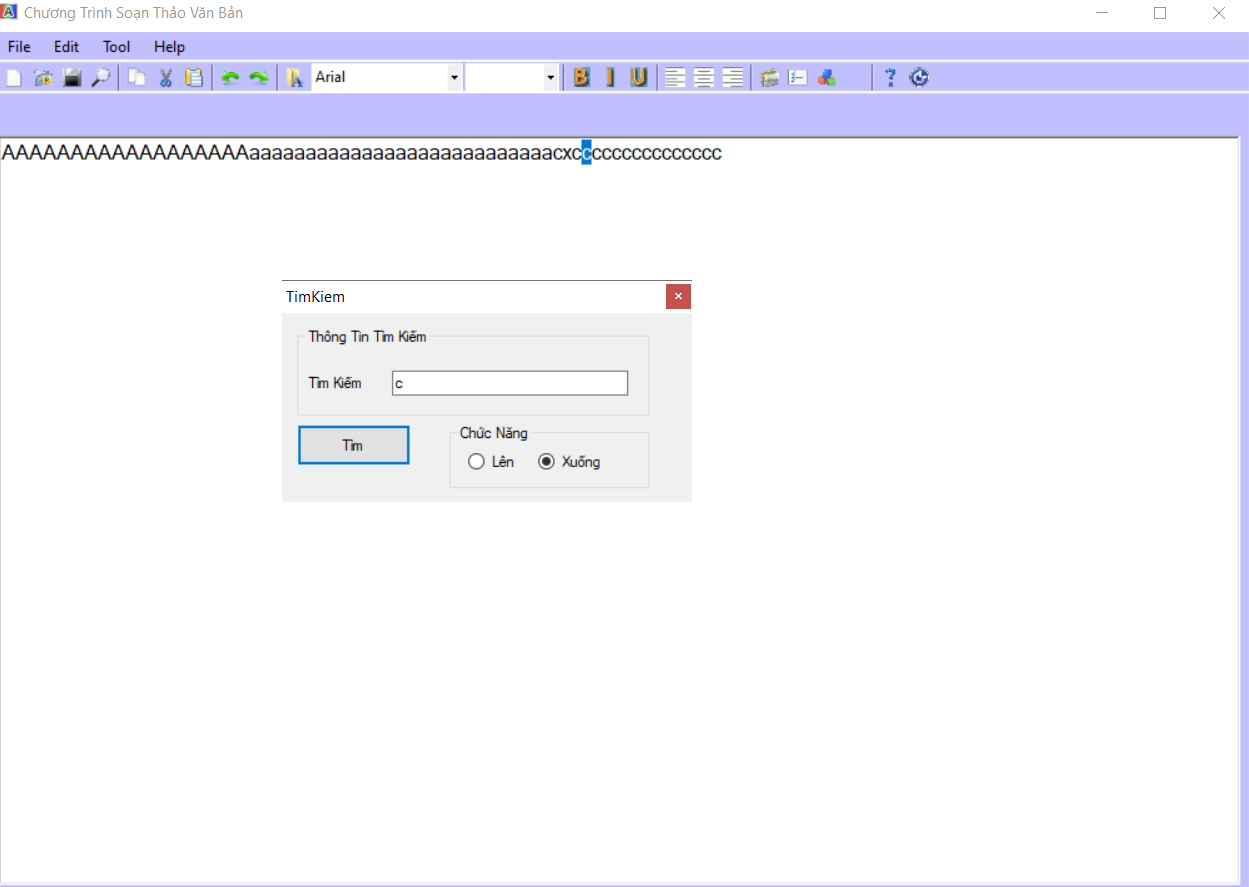
Người dùng có thể mở file đã lưu bằng chức năng Open file



Sau khi soạn thảo xong người dùng có thể lưu file dưới dạng Rich TextFile(\*rtf)



Người dùng có thể tìm kiếm các ký có trong văn bản bằng cách dùng chức năng tìm kiếm



## **Các mẫu dùng cho thiết kế code**

1. **Các lớp trong chương trình**

* **Form Soạn Thảo**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Text;

using System.IO;

using System.Diagnostics;

using System.Windows.Forms;

namespace Chuong\_Trinh\_Soan\_Thao\_Van\_Ban

{

public partial class frmworkpad : Form

{

private string duongdan = ""; //Tạo đường dẫn lưu hoặc mở tập tin.

private int kiemTraSave = 0; //0 tức là chưa lưu. 1 là đã lưu.

private SaveFileDialog save; //Tạo Sự kiện Dialog save để hiện thị cửa sổ save có sẵn trong hệ thống.

private OpenFileDialog open; //Tạo sự kiện Dialog open để hiển thị cửa sổ open có săn trong hệ thống.

private ColorDialog color; //Tạo Sự kiện Dialog color để hiện thị màu chữ có sẵn trong hệ thống.

private FontDialog dialogFonts; //Tạo Sự kiện Dialog Fonts để hiện thị màu chữ có sẵn trong hệ thống.

private int check = 0; //Tạo biến check để kiểm tra cho các sự kiện in đậm, in nghiêng, gạch chân, canh trái, phải, giữa...

frmtimkiem fsearch;

public frmworkpad()

{

InitializeComponent();

}

private void frmworkpad\_Load(object sender, EventArgs e)

{

loadFonts();

//toolStrip\_mau.BackColor =

}

#region Các Phương Thức Làm Việc

public void dinhDangFonts()

{

float fsize = 10;

if (tsbcbfontsize.SelectedIndex != -1)

{

fsize = (float)float.Parse(tsbcbfontsize.SelectedItem.ToString());

}

string fname = "Arial";

if (tsbcbfonts.SelectedIndex != -1)

{

fname = tsbcbfonts.SelectedItem.ToString();

}

try

{

Font font = new Font(new FontFamily(fname), fsize);

rtbinfo.SelectionFont = font;

}

catch

{

MessageBox.Show("Font này không hỗ trợ kiểu hiển thị hiện tại", "Lỗi", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

//rtbinfo.Dispose();

}

public void readDocFile()

{

//VS 2010 ta dùng Application chứ không dùng ApplicationClass.

// Tạo một thể hiện của ứng dụng MS Word.

Microsoft.Office.Interop.Word.Application wordApp = new Microsoft.Office.Interop.Word.Application();

// Các tham số được sử dụng trong hàm Open được hỗ trợ bởi thư viện API của MS Word

object fileName = duongdan; //path là đường dẫn đến file cần đọc

object missing = System.Reflection.Missing.Value;

object vk\_read\_only = false;

object vk\_visible = true;

object vk\_false = false;

// không sử dụng các thông số không cần thiết ngoai trừ đường dẫn đến file cần mở

Microsoft.Office.Interop.Word.Document doc = wordApp.Documents.Open(ref fileName, ref missing,

ref vk\_read\_only, ref missing,

ref missing, ref missing,

ref missing, ref missing,

ref missing, ref missing,

ref missing, ref vk\_visible,

ref missing, ref missing,

ref missing, ref missing);

doc.ActiveWindow.Selection.WholeStory();

doc.ActiveWindow.Selection.Copy();

IDataObject data = Clipboard.GetDataObject();

rtbinfo.Text = data.GetData(DataFormats.UnicodeText).ToString();//hiển thị dữ liệu lên RichTextBox

if (doc != null)

{

doc.Close(ref vk\_false, ref missing, ref missing);

}

wordApp.Quit(ref vk\_false, ref missing, ref missing);

}

public void taoDinhDangSave()

{

save = new SaveFileDialog();

save.DefaultExt = "rtf"; // Mặc định khi mở của sổ lưu là định dang \*.rtf.

save.Filter = "RichTextFile |\*.rtf|Doc file (\*.doc)|\*.doc|All files (\*.\*)|\*.\*";

//Các định dang khác được cố định khi lưu file.

}

public void kiemTraThoat()

{

//kiem tra file đã được lưu chưa. nếu = 0, tức là chưa lưu, =1 : tức là đã lưu.

if (this.kiemTraSave == 0)

{// nếu chưa lưu. Làm

if (!rtbinfo.Text.Equals("")) //kiểm tra nội dung bên trong.

{//nếu khác rỗng. Làm

if (this.duongdan.Equals("")) //kiểm tra đường dẫn.

{//đường dẫn rỗng. Làm

save = new SaveFileDialog();

save.DefaultExt = "rtf";

save.Filter = "RichTextFile |\*.rtf";

DialogResult result = save.ShowDialog();

/\*

\* mặc định khi show ra cửa sổ OpenFileDialog và SaveFileDialog.

\* sẽ có 3 button OK / NO / Cancel.

\*/

if (result == DialogResult.Cancel)

{//nếu là Cancel. Làmd

return; //trả về và không làm gì. thoát khỏi sự kiện.

}

//gán đường dẫn.

this.duongdan = save.FileName;

try

{

rtbinfo.SaveFile(duongdan); //lưu file theo đường dẫn đã chọn

}

catch (System.Exception ex)

{

MessageBox.Show(ex.Message.ToString(), "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

}

}

else //Ngược lại trường hợp đường dẫn rỗng.

{

try

{

//có đường dẫn rồi. cho phép lưu chồng file theo đường dẫn.

rtbinfo.SaveFile(duongdan);

}

catch (System.Exception ex)

{

MessageBox.Show(ex.Message.ToString(), "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

}

}

}

}

}

public void kiemTraNew()

{

//xuất hiện thông báo với Icon Question

if (!rtbinfo.Text.Equals(""))

{

DialogResult chon = MessageBox.Show("Bạn muốn lưu trước khi tạo mới !", "Thông Báo", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);

// Nếu chọn ok

if (chon == DialogResult.Yes)

{

/\* \* Làm.

\* Nếu đường dẫn rỗng. trong lập tình hướng đối tượng không thể so sánh chuỗi bằng '=='. ta phải dùng từ khóa 'Equals'

\* \*/

if (this.duongdan.Equals(""))

{

// Khởi tạo sự kiện lưu file.

save = new SaveFileDialog();

save.DefaultExt = "rtf";

save.Filter = "RichTextFile |\*.rtf"; //Gán cho file lưu xuống mặc định là \*.rtf.

/\* \*

\* Gọi chức năng lưu của hệ thống Window

\* \*/

DialogResult result = save.ShowDialog();

// Nếu chọn cancel.

if (result == DialogResult.Cancel)

{

//kiểu trả về, rời khỏi sự kiện thoát.

return;

}

/\*\* Nếu không chọn cancel.

\* Gán duongdan = với save.Filename;

\*/

this.duongdan = save.FileName;

// Lưu đường dẫn vào biến.

try // Thử.

{

rtbinfo.SaveFile(duongdan); // Lưu tập tin từ RichTextBox định dạng mặc định [\*.rtf]

}

catch (System.Exception ex) // Trường hợp ngoại lệ

{

MessageBox.Show(ex.Message.ToString(), "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

// Thông báo lỗi và rời khỏi sự kiện.

}

}

else

{

//Tức là có đường dẫn không trống hay là đã có đường dẫn.

// Tiến hành lưu tập tin mà không cần gọi cửa sổ đê lưu

try

{

rtbinfo.SaveFile(duongdan); // Lưu tập tin.

}

catch (System.Exception ex)

{

MessageBox.Show(ex.Message.ToString(), "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

}

}

rtbinfo.Text = ""; // Khởi tạo RichTextBox.

}

// Ngược lại trường hợp chọn Yes

else if (chon == DialogResult.No)

{

rtbinfo.Text = ""; // Thoát không cần lưu.

}

else // Ngược lại nếu chọn nút thứ 3 là nút Cancel.

{

return; // Trả về, không lưu gì, thoát khỏi sự kiện.

}

}

else

{

return;

}

}

public void loadFonts()

{

//tạo danh sách cỡ chữ....

for (int i = 8; i <= 72; i++)

{

tsbcbfontsize.Items.Add(i.ToString());

}

//set mặc định cho Size.

tsbcbfontsize.SelectedIndex = 4;

System.Drawing.Text.InstalledFontCollection fonts = new System.Drawing.Text.InstalledFontCollection();

//tao danh sach Fonts lấy từ hệ thống chính máy tính bạn.

foreach (FontFamily f in fonts.Families)

{

tsbcbfonts.Items.Add(f.Name.ToString());

}

tsbcbfonts.SelectedItem = "Arial";

//set mặc định cho Fonts.

}

public void kiemTraOpen()

{

open = new OpenFileDialog();

save = new SaveFileDialog();

if (!this.rtbinfo.Text.Equals(""))

{

DialogResult Result2 = MessageBox.Show("Bạn có muốn lưu lại trước khi mở tệp mới ?", "Thông báo", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);

//tạo một biến kết quả trả về từ MessageBox

if (Result2 == DialogResult.Yes) //nếu chọn Yes

{

//tương tự menu item Thoát

if (this.duongdan.Equals(""))

{

DialogResult resVal = save.ShowDialog();

if (resVal == DialogResult.Cancel)

{

return;

}

this.duongdan = save.FileName;

try

{

rtbinfo.SaveFile(duongdan);

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

}

}

else

{

try

{

rtbinfo.SaveFile(duongdan);

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

}

}

DialogResult Result = open.ShowDialog();

//tạo một biến kết quả từ việc mở tập tin

//lúc đó hộp thoại mở tập tin sẽ có 2 nút là Open và Cancel

if (Result == DialogResult.Cancel) //nếu chọn Cancel

{

return; //rời khỏi sự kiện

}

else //hoặc chọn mở

{

try //thử

{

this.duongdan = open.FileName; //lưu đường dẫn tập tin

readDocFile();

}

catch (Exception Ex) //nếu có ngoại lệ

{

MessageBox.Show(Ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

//thông báo lổi

}

}

}

else if (Result2 == DialogResult.No)

{

DialogResult Result = open.ShowDialog();

//tạo một biến kết quả từ việc mở tập tin

//lúc đó hộp thoại mở tập tin sẽ có 2 nút là Open và Cancel

if (Result == DialogResult.Cancel) //nếu chọn Cancel

{

return; //rời khỏi sự kiện

}

else //hoặc chọn mở

{

try //thử

{

this.duongdan = open.FileName; //lưu đường dẫn tập tin

readDocFile();

}

catch (Exception Ex) //nếu có ngoại lệ

{

MessageBox.Show(Ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

//thông báo lổi

}

}

}

else

{

return; //rời khỏi sự kiện

}

}

else

{

DialogResult Result = open.ShowDialog();

//tạo một biến kết quả từ việc mở tập tin

//lúc đó hộp thoại mở tập tin sẽ có 2 nút là Open và Cancel

if (Result == DialogResult.Cancel) //nếu chọn Cancel

{

return; //rời khỏi sự kiện

}

else //hoặc chọn mở

{

try //thử

{

this.duongdan = open.FileName; //lưu đường dẫn tập tin

readDocFile();

}

catch (Exception Ex) //nếu có ngoại lệ

{

MessageBox.Show(Ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

//thông báo lổi

}

}

}

}

public void saveData()

{

//giống như trên nhưng không cần biết đã lưu tập tin hay chưa

if (this.duongdan.Equals(""))

{

save = new SaveFileDialog();

save.DefaultExt = "rtf";

save.Filter = "RichTextFile |\*.rtf";

DialogResult Result = save.ShowDialog();

if (Result == DialogResult.Cancel)

{

return;

}

duongdan = save.FileName;

try

{

rtbinfo.SaveFile(duongdan);

}

catch (Exception Ex)

{

MessageBox.Show(Ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

}

}

this.kiemTraSave = 1; //đặt lại trạng thái cho biến kiểm tra việc lưu

}

#endregion

#region Bắt Các Sự Kiện CLick Trong MenuStrip

//MenuStrip Exit.

private void menustrip\_Exit\_Click(object sender, EventArgs e)

{

//kiem tra

if (this.kiemTraSave == 0) //Tức là đoạn văn bản chưa được lưu.

{

Application.Exit();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\* Khi thoát chương trình.

\* Mặc định hệ thống sẽ gọi sự kiện FormClosing hoặc FormClosed.

\* Để tránh trường hợp này, ta nên lập trình cho sự kiện thoát của FORM trước.

\* Sau đó vào phần code cho Menu chỉ cần khọi hàm thoát, khi đó chương trình sẽ tự động gọi hàm.

\* FormClosing hoặc FormClosed.

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

}

// Ngược lại trường hợp kiemTraSave = 0

// tức là đã lưu rồi.

else

{

Application.Exit();

}

}

//MenuStrip Save As

private void menustrip\_SaveAs\_Click(object sender, EventArgs e)

{

taoDinhDangSave();

DialogResult chon = save.ShowDialog();

this.duongdan = save.FileName;

try

{

if (chon == DialogResult.OK)

{

rtbinfo.SaveFile(duongdan);

}

else

{

return;

}

}

catch (System.Exception ex)

{

MessageBox.Show(ex.Message, "Thông Báo", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//MenuStrip Save.

private void menustrip\_Save\_Click(object sender, EventArgs e)

{

if (kiemTraSave == 0)

{

saveData();

}

else

{

rtbinfo.SaveFile(duongdan);

}

}

//MenuStrip Open.

private void menustrip\_Open\_Click(object sender, EventArgs e)

{

if (kiemTraSave == 0)

{

kiemTraOpen();

}

}

//MenuStrip New.

private void menustrip\_New\_Click(object sender, EventArgs e)

{

if (kiemTraSave == 0)

{

kiemTraNew();

}

else

{

rtbinfo.Text = ""; // Tạo mới

}

}

//MenuStrip Copy.

private void menustrip\_Copy\_Click(object sender, EventArgs e)

{

rtbinfo.Copy();

}

//MenuStrip Cut.

private void menustrip\_Cut\_Click(object sender, EventArgs e)

{

rtbinfo.Cut();

}

//MenuStrip Undo.

private void menustrip\_Undo\_Click(object sender, EventArgs e)

{

rtbinfo.Undo();

}

//MenuStrip Redo.

private void menustrip\_Re\_Undo\_Click(object sender, EventArgs e)

{

rtbinfo.Redo();

}

//MenuStrip In Đậm.

private void menustrip\_InDam\_Click(object sender, EventArgs e)

{

if (this.check == 0)

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Bold);

check++;

}

else

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Regular);

check--;

}

}

//MenuStrip In Nghiêng.

private void menustrip\_InNghieng\_Click(object sender, EventArgs e)

{

if (this.check == 0)

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Italic);

check++;

}

else

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Regular);

check--;

}

}

//MenuStrip Gạch Chân.

private void menustrip\_Gach\_Chan\_Click(object sender, EventArgs e)

{

if (this.check == 0)

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Underline);

check++;

}

else

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Regular);

check--;

}

}

//MenuStrip Canh Giữa.

private void menustrip\_Canh\_Giua\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Center;

}

//MenuStrip Canh Trái.

private void menustrip\_Canh\_Trai\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Left;

}

//MenuStrip Canh Phải.

private void menustrip\_Canh\_Phai\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Right;

}

//MenuStrip Chèn Hình.

private void menustrip\_Chen\_Hinh\_Click\_1(object sender, EventArgs e)

{

open = new OpenFileDialog();

DialogResult Result = open.ShowDialog();

//khai báo biến kết quả cho việc mở tập tin

if (Result == DialogResult.Cancel) //nếu chọn Cancel

{

return; //rời khỏi sự kiện

}

else //hoặc chọn Open

{

try //thử

{

string ImagePath = open.FileName; //lấy đường dẫn

//của tập tin

Bitmap myBitmap = new Bitmap(ImagePath); //tạo một Bitmap

Clipboard.SetDataObject(myBitmap); //đặt đối tượng dử liệu vào Clipboard

DataFormats.Format myFormat = DataFormats.GetFormat(DataFormats.Bitmap);

//lấy định dạng của hình

if (rtbinfo.CanPaste(myFormat)) //nếu có thể chèn vào RTBox

{

rtbinfo.Paste(myFormat); //chèn hình vào

}

else //nếu không thể chèn

{

MessageBox.Show("Không thể chèn !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

//hiển thị thông báo

}

}

catch (Exception ex) //nếu có ngoại lệ (mở tập tin không phải hình)

{

MessageBox.Show(ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

//thông báo và rời khỏi sự kiện

}

}

}

//MenuStrip Đánh Dấu.

private void menustrip\_Danh\_Dau\_Click(object sender, EventArgs e)

{

if (check == 0)

{

rtbinfo.SelectionBullet = true; //đánh dấu

check++;

}

else

{

rtbinfo.SelectionBullet = false; //hủy đánh dấu

check--;

}

}

//MenuStrip Chọn Màu Chữ.

private void menustrip\_Mau\_Chu\_Click(object sender, EventArgs e)

{

color = new ColorDialog();

color.ShowDialog(); //hiển thị hộp thoại chọn màu

rtbinfo.SelectionColor = color.Color; //đặt lại màu

}

//MenuStrip Hướng Dẫn

private void menustrip\_huongdan\_Click(object sender, EventArgs e)

{

frmhuongdan huongdan = new frmhuongdan();

huongdan.Show();

}

//MenuStrip Thông Tin

private void menustrip\_thongtin\_Click(object sender, EventArgs e)

{

ThongTin thongtin = new ThongTin();

thongtin.Show();

}

//MenuStrip Search.

private void menustrip\_Search\_Click(object sender, EventArgs e)

{

if (fsearch == null || fsearch.IsDisposed)

fsearch = new frmtimkiem(rtbinfo);

fsearch.ShowFind();

}

#endregion

#region Bắt Sự Kiện CLick ToolStrip

//ToolStrip Combobox Chọn Fonts.

private void tsbcbfonts\_Click(object sender, EventArgs e)

{

dinhDangFonts();

}

//ToolStrip Combobox Chọn Size.

private void tsbcbfontsize\_Click(object sender, EventArgs e)

{

dinhDangFonts();

}

//ToolStrip Icon New (tạo file mới trong RichTextBox)

private void tsbnew\_Click(object sender, EventArgs e)

{

if (kiemTraSave == 0)

{

kiemTraNew();

}

else

{

rtbinfo.Text = ""; // Tạo mới

}

}

//ToolStrip Icon Save.

private void tsbsave\_Click(object sender, EventArgs e)

{

if (kiemTraSave == 0)

{

saveData();

}

else

{

rtbinfo.SaveFile(duongdan);

}

}

//ToolStrip Icon Open.

private void tsbopen\_Click(object sender, EventArgs e)

{

if (kiemTraSave == 0)

{

kiemTraOpen();

}

}

//ToolStrip Icon Undo

private void tsbundo\_Click(object sender, EventArgs e)

{

rtbinfo.Undo();

}

//ToolStrip Icon Re Undo

private void tsbreundo\_Click(object sender, EventArgs e)

{

rtbinfo.Redo();

}

//ToolStrip Icon Cut

private void tsbcut\_Click(object sender, EventArgs e)

{

rtbinfo.Cut();

}

//ToolStrip Icon Copy

private void tsbcopy\_Click(object sender, EventArgs e)

{

rtbinfo.Copy();

}

//ToolStrip Icon Paste

private void tsbpaste\_Click(object sender, EventArgs e)

{

rtbinfo.Paste();

}

//ToolStrip Icon In Đậm.

private void tsbindam\_Click(object sender, EventArgs e)

{

if (this.check == 0)

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Bold);

check++;

}

else

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Regular);

check--;

}

}

//ToolStrip Icon In Nghiêng.

private void tsbinnghieng\_Click(object sender, EventArgs e)

{

if (this.check == 0)

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Italic);

check++;

}

else

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Regular);

check--;

}

}

//ToolStrip Icon Gạch Chân.

private void tsbgachchan\_Click(object sender, EventArgs e)

{

if (this.check == 0)

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Underline);

check++;

}

else

{

rtbinfo.SelectionFont = new Font(rtbinfo.SelectionFont, FontStyle.Regular);

check--;

}

}

//ToolStrip Icon Canh Trái.

private void tsbcanhtrai\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Left;

}

//ToolStrip Icon Canh Giữa.

private void tsbcanhgiua\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Center;

}

//ToolStrip Icon Canh Phải.

private void tsbcanhphai\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Right;

}

//ToolStrip Icon Chèn Hình.

private void tsbchonhinh\_Click(object sender, EventArgs e)

{

open = new OpenFileDialog();

DialogResult Result = open.ShowDialog();

//khai báo biến kết quả cho việc mở tập tin

if (Result == DialogResult.Cancel) //nếu chọn Cancel

{

return; //rời khỏi sự kiện

}

else //hoặc chọn Open

{

try //thử

{

string ImagePath = open.FileName; //lấy đường dẫn

//của tập tin

Bitmap myBitmap = new Bitmap(ImagePath); //tạo một Bitmap

Clipboard.SetDataObject(myBitmap); //đặt đối tượng dử liệu vào Clipboard

DataFormats.Format myFormat = DataFormats.GetFormat(DataFormats.Bitmap);

//lấy định dạng của hình

if (rtbinfo.CanPaste(myFormat)) //nếu có thể chèn vào RTBox

{

rtbinfo.Paste(myFormat); //chèn hình vào

}

else //nếu không thể chèn

{

MessageBox.Show("Không thể chèn !", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

//hiển thị thông báo

}

}

catch (Exception ex) //nếu có ngoại lệ (mở tập tin không phải hình)

{

MessageBox.Show(ex.Message.ToString(), "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Stop);

return;

//thông báo và rời khỏi sự kiện

}

}

}

//ToolStrip Icon Đánh Dấu.

private void tsbdanhdau\_Click(object sender, EventArgs e)

{

if (check == 0)

{

rtbinfo.SelectionBullet = true; //đánh dấu

check++;

}

else

{

rtbinfo.SelectionBullet = false; //hủy đánh dấu

check--;

}

}

//ToolStrip Icon Màu Chữ.

private void tsbchonmauchu\_Click(object sender, EventArgs e)

{

color = new ColorDialog();

DialogResult mau = color.ShowDialog(); //hiển thị hộp thoại chọn màu

if (mau == DialogResult.OK)

{

rtbinfo.SelectionColor = color.Color; //đặt lại màu

toolStrip\_mau.BackColor = color.Color;

}

else

{

return;

}

}

//ToolStrip Icon Fonts Full.

private void tsbfontstyle\_Click(object sender, EventArgs e)

{

dialogFonts = new FontDialog();

DialogResult font = dialogFonts.ShowDialog();

if (this.rtbinfo.SelectedText.Equals(""))

{

if (font == DialogResult.OK)

{

rtbinfo.Font = dialogFonts.Font;

}

else

{

return;

}

}

else

{

if (font == DialogResult.OK)

{

rtbinfo.SelectionFont = dialogFonts.Font;

}

else

{

return;

}

}

}

//ToolStrip Icon Help.

private void tsbhelp\_Click(object sender, EventArgs e)

{

frmhuongdan huongdan = new frmhuongdan();

huongdan.Show();

}

//ToolStrip Icon Thông Tin.

private void tsbthongtin\_Click(object sender, EventArgs e)

{

ThongTin info = new ThongTin();

info.Show();

}

#endregion

private void tsbtimkiem\_Click(object sender, EventArgs e)

{

if (fsearch == null || fsearch.IsDisposed)

fsearch = new frmtimkiem(rtbinfo);

fsearch.ShowFind();

}

#region Control Right CLick Mouse

private void rightclick\_cut\_Click(object sender, EventArgs e)

{

rtbinfo.Cut();

}

private void rightclick\_copy\_Click(object sender, EventArgs e)

{

rtbinfo.Copy();

}

private void rightclick\_paste\_Click(object sender, EventArgs e)

{

rtbinfo.Paste();

}

private void rightclick\_fonts\_Click(object sender, EventArgs e)

{

dialogFonts = new FontDialog();

DialogResult font = dialogFonts.ShowDialog();

if (this.rtbinfo.SelectedText.Equals(""))

{

if (font == DialogResult.OK)

{

rtbinfo.Font = dialogFonts.Font;

}

else

{

return;

}

}

else

{

if (font == DialogResult.OK)

{

rtbinfo.SelectionFont = dialogFonts.Font;

}

else

{

return;

}

}

}

private void rightclick\_danhdau\_Click(object sender, EventArgs e)

{

if (check == 0)

{

rtbinfo.SelectionBullet = true; //đánh dấu

check++;

}

else

{

rtbinfo.SelectionBullet = false; //hủy đánh dấu

check--;

}

}

private void rightclick\_mauchu\_Click(object sender, EventArgs e)

{

color = new ColorDialog();

color.ShowDialog(); //hiển thị hộp thoại chọn màu

rtbinfo.SelectionColor = color.Color; //đặt lại màu

}

private void rightclick\_canhtrai\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Left;

}

private void rightclick\_canhgiua\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Center;

}

private void rightclick\_canhphai\_Click(object sender, EventArgs e)

{

rtbinfo.SelectionAlignment = HorizontalAlignment.Right;

}

private void rightclick\_selectall\_Click(object sender, EventArgs e)

{

rtbinfo.SelectAll();

}

#endregion

private void frmworkpad\_FormClosing(object sender, FormClosingEventArgs e)

{

DialogResult thoat = MessageBox.Show("Bạn muốn thoát khỏi chương trinh", "Thông Báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);

if (thoat == DialogResult.Cancel)

{

e.Cancel = true;

}

else

{

kiemTraThoat();

}

}

private void menuStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

}

}

}

* **Form Tìm kiếm**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace Chuong\_Trinh\_Soan\_Thao\_Van\_Ban

{

public partial class frmtimkiem : Form

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* lưu ý trước khi tạo class này ta cần thức hiện thao tác click vào RichTextBox bên Form Chính.

\* cài đặt lại thuộc tính RichTextBox là Hideselection = false.

\* để khi bấm vào nút tìm kiếm bên class này, thì class Form chính trong RichTex sẽ không bị ẩn đi vùng tìm kiếm.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

private RichTextBox \_richTextBox; //khởi tạo biến bool \_close = 'True' or 'false' để kiểm tra.

// Khởi tạo có tham số. Để truyền RichTextBox là rtbinfo trong class chương trình demo.

public frmtimkiem(RichTextBox richTextBox)

{

InitializeComponent();

\_richTextBox = richTextBox;

}

public void ShowFind()

{

//Hiển thị Form Search với tham số là RichTextBox được truyền vào.

this.Show(\_richTextBox);

txtsearch.Focus();

txtsearch.SelectAll();

}

// Bắt sự kiện Click cho button Search.

public void btnsearch\_Click(object sender, EventArgs e)

{

//RichTextBox hỗ trợ phương Thức int Find(....);

Find(\_richTextBox, txtsearch.Text, radup.Checked);

}

public void Find(RichTextBox richtext, string nhap, bool check\_up)

{

//Khởi tạo giá trị RichTextBoxFinds = none.

RichTextBoxFinds chose = RichTextBoxFinds.None;

//nếu đã check vào radup. hay raduo.checked

if (check\_up)

{

//tìm kiếm chạy ngược từ dưới lên trên

chose |= RichTextBoxFinds.Reverse;

}

// tạo biến int. như đã nói trên RichTextBox hỗ trợ phương thức int Find(...). giá trị trả về là int.

int index;

//neu check\_up = true

if (check\_up)

{

index = richtext.Find(nhap,0, richtext.SelectionStart, chose);

}

else

{

index = richtext.Find(nhap, richtext.SelectionStart + richtext.SelectionLength, chose);

}

if (index >= 0)

{

richtext.SelectionStart = index;

richtext.SelectionLength = nhap.Length;

}

else

{

MessageBox.Show(Application.ProductName + " has finished searching the document.",

Application.ProductName, MessageBoxButtons.OK,

MessageBoxIcon.Information);

}

}

}

}

### **Tài liệu tham khảo:**

* 1. <http://info24h.vn/lap-trinh-xu-ly-giao-dien-trong-winform-voi-c-183.html>
  2. <https://www.guru99.com/c-sharp-windows-forms-application.html>