

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**



**MÔN LẬP TRÌNH TRỰC QUAN**  
**BÀI TẬP THỰC HÀNH 4**

GVHD: Nguyễn Ngọc Quý

Sinh viên thực hiện: Nguyễn Gia Hưng – 24520604

## This image shows a full page of white paper with horizontal dotted lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

**Người nhận xét**  
(Ký tên và ghi rõ họ tên)

**MỤC LỤC**

<b>Bài 1. Viết chương trình chặn các thông điệp bàn phím và thông điệp chuột</b>	<b>6</b>
.....	6
Mô tả chương trình.....	6
Nội dung code .....	6
Kiểm thử chương trình.....	8
<b>Bài 2. Viết ứng dụng hiển thị ngày giờ hệ thống.....</b>	<b>11</b>
Mô tả chương trình.....	11
Nội dung code .....	11
Kiểm thử chương trình.....	12
<b>Bài 3. Xây dựng chương trình mô phỏng điều khiển Windows media Player để phát các tập tin Audio và Video.....</b>	<b>15</b>
Mô tả chương trình.....	15
Nội dung code .....	15
Kiểm thử chương trình.....	18
<b>Bài 4. Xây dựng chương trình soạn thảo văn bản cơ bản.....</b>	<b>20</b>
Mô tả chương trình.....	20
Nội dung code .....	22
Kiểm thử chương trình.....	30
<b>Bài 5. Xây dựng chương trình quản lý thông tin sinh viên cơ bản .....</b>	<b>34</b>
Mô tả chương trình.....	34
Nội dung code .....	35
Kiểm thử chương trình.....	43
<b>Bài 6. Viết chương trình sao chép tập tin .....</b>	<b>47</b>
Mô tả chương trình.....	47
Nội dung code .....	47
Kiểm thử chương trình.....	54

**DANH MỤC BẢNG**

Bảng 1.1. Nội dung code bài 1 .....	6
Bảng 1.2. Kiểm thử bài 1.....	8
Bảng 2.1. Nội dung code bài 2 .....	11
Bảng 2.2. Kiểm thử bài 2.....	13
Bảng 3.1. Nội dung code bài 3 .....	15
Bảng 3.2. Kiểm thử bài 3.....	18
Bảng 4.1. Nội dung code bài 4 .....	22
Bảng 4.2. Kiểm thử bài 4.....	30
Bảng 5.1. Nội dung code bài 5 .....	35
Bảng 5.2. Kiểm thử bài 5.....	44
Bảng 6.1. Nội dung code bài 6 .....	47
Bảng 6.2. Kiểm thử bài 6.....	54

## **DANH MỤC HÌNH ẢNH**

Hình 1.1. Giao diện chính bài 1 .....	8
Hình 2.1. Giao diện chính bài 2 .....	13
Hình 3.1. Giao diện chính bài 3 .....	18
Hình 4.1. Giao diện chính bài 4 .....	30
Hình 5.1. Giao diện chính bài 5 .....	43
Hình 5.2. Giao diện thêm mới bài 5 .....	44
Hình 6.1. Giao diện chính bài 6 .....	54

## NỘI DUNG BÀI LÀM

### Bài 1. Viết chương trình chặn các thông điệp bàn phím và thông điệp chuột

#### Mô tả chương trình

- Yêu cầu: Xử lý sự kiện khi người dùng click chuột vào form hoặc nhấn phím khi form đang mở
  - Viết chương trình chặn các thông điệp bàn phím và thông điệp chuột
  - Khi nhấn phím bất kỳ thì xuất các thông tin: Mã ASCII, key code,...
- Hướng xử lý:
  - Tạo một label để thông báo sự kiện xảy ra
  - Xử lý sự kiện *MouseClick* và *KeyPress* của form, lấy thông tin từ *MouseEventArgs e* và *KeyPressEventArgs e* để in ra thông tin ra theo yêu cầu

#### Nội dung code

Bảng 1.1. Nội dung code bài 1

##### MouseKey.cs

```
using System.Windows.Forms;

namespace Bai1
{
    public partial class MouseKey : Form
    {
        public MouseKey()
        {
            InitializeComponent();
        }

        private void ShowMessage(string msg)
        {
            labelNotice.Text = msg;
        }

        private void MouseKey_MouseClick(object sender, MouseEventArgs e)
        {
            ShowMessage($"Mouse Click on ({e.X.ToString()}, {e.Y.ToString()})");
        }

        private void MouseKey_KeyPress(object sender, KeyPressEventArgs e)
        {
            ShowMessage($"Key Pressed: '{e.KeyChar.ToString()}' with ASCII code {(int)e.KeyChar}");
        }
    }
}
```

##### MouseKey.Design.cs

```
using System.Drawing;

namespace Bai1
{
    partial class MouseKey
    {
        //
    }
}
```

```

    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise,
    false.</param>
    protected override void Dispose(bool disposing)
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
        this.title = new System.Windows.Forms.Label();
        this.labelNotice = new System.Windows.Forms.Label();
        this.SuspendLayout();
        //
        // title
        //
        this.title.BackColor = System.Drawing.Color.LightSkyBlue;
        this.title.Dock = System.Windows.Forms.DockStyle.Top;
        this.title.Font = new System.Drawing.Font("Segoe UI", 22.2F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.title.ForeColor = System.Drawing.Color.AliceBlue;
        this.title.Location = new System.Drawing.Point(0, 0);
        this.title.Name = "title";
        this.title.Size = new System.Drawing.Size(643, 60);
        this.title.TabIndex = 0;
        this.title.Text = "Mouse and Key";
        this.title.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
        //
        // labelNotice
        //
        this.labelNotice.BackColor = System.Drawing.Color.LightSteelBlue;
        this.labelNotice.Dock = System.Windows.Forms.DockStyle.Bottom;
        this.labelNotice.Location = new System.Drawing.Point(0, 404);
        this.labelNotice.Name = "labelNotice";
        this.labelNotice.Size = new System.Drawing.Size(643, 48);
        this.labelNotice.TabIndex = 2;
        this.labelNotice.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
        //
        // MouseKey
        //
        this.AutoScaleDimensions = new System.Drawing.SizeF(11F, 28F);
        this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
        this.BackColor = System.Drawing.Color.AliceBlue;
        this.ClientSize = new System.Drawing.Size(643, 452);
        this.Controls.Add(this.labelNotice);
        this.Controls.Add(this.title);
        this.Font = new System.Drawing.Font("Segoe UI", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
        this.Margin = new System.Windows.Forms.Padding(4, 5, 4, 5);
        this.Name = "MouseKey";
        this.Text = "Mouse and Key";
        this.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.MouseKey_KeyPress);
        this.MouseClick += new
System.Windows.Forms.MouseEventHandler(this.MouseKey_MouseClick);
        this.ResumeLayout(false);
    }

    private System.Windows.Forms.Label title;

```

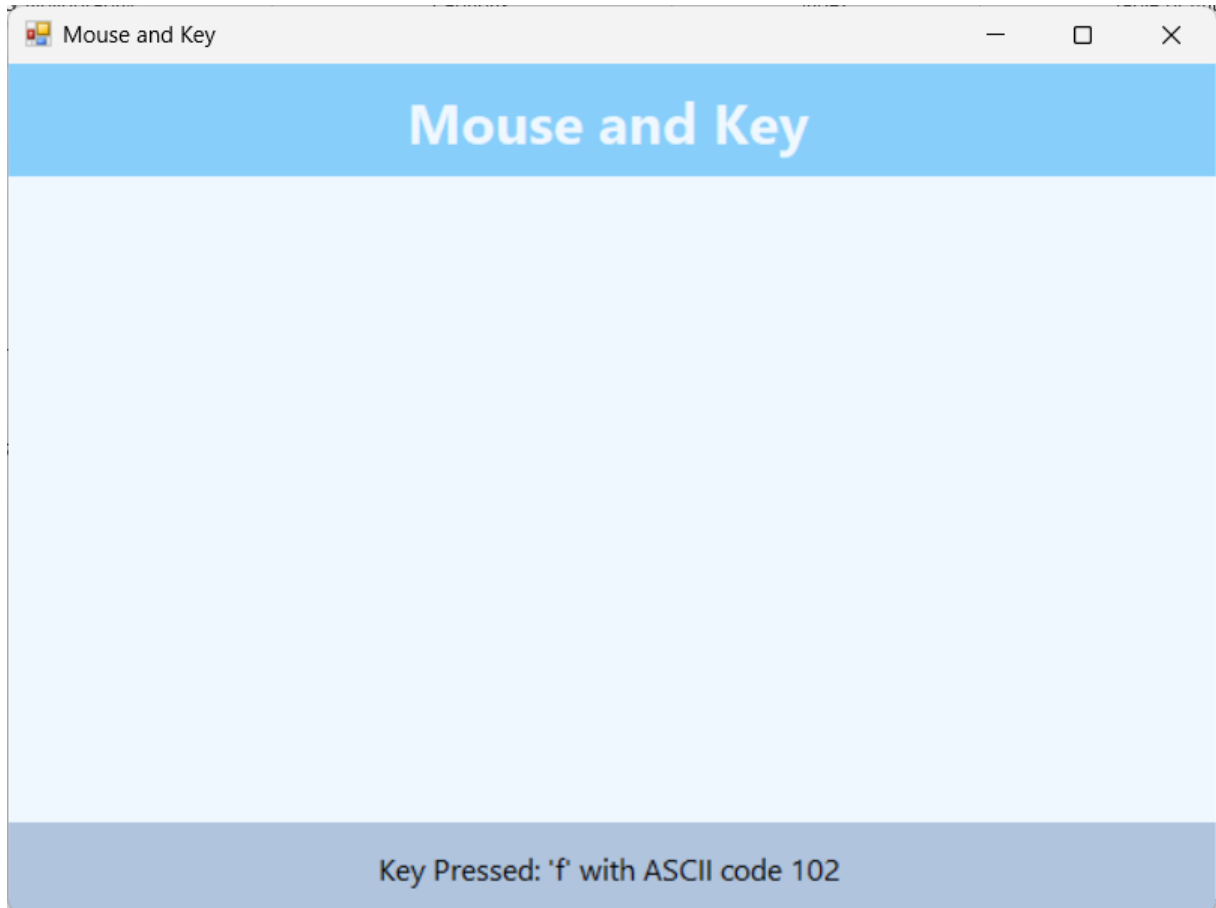
```
#endregion

    private System.Windows.Forms.Label labelNotice;
}
}
```

### Kiểm thử chương trình

- Giao diện chương trình:

Hình 1.1. Giao diện chính bài 1



- Kịch bản kiểm thử:

Bảng 1.2. Kiểm thử bài 1

STT	Tiêu chí	Hành động	Kết quả
1	Kiểm tra phím thường	Mở chương trình, nhấn các ký tự (chữ và số) thông thường trên bàn phím.	In ra ký tự đúng với phím đã gõ cùng với mã ASCII.



## IT008 – LẬP TRÌNH TRỰC QUAN

2	Kiểm tra phím Hoa và ký tự đặc biệt	Giữ Shift và nhấn các ký tự chữ và số.  Bật Capslock và nhấn các ký tự.	In ra kí tự in hoa và đặc biệt đúng những gì đã gõ, mã ASCII phải khác với kí tự thường.
3	Nhiều phím liên tục và ký tự tiếng Việt	Bật trình gõ tiếng Việt, gõ 2 (nhiều) ký tự liên tiếp tạo thành chữ cái tiếng Việt (aa → â,...)	Có thể in ra ký tự tiếng Việt với mã ASCII theo Unicode
4	Kiểm tra phím đặc biệt	Nhấn các phím chức năng trên bàn phím (Shift, Ctrl, Alt, F1,...)	Không thay đổi thông báo do những phím này không tạo dữ liệu
5	Kiểm tra click chuột trên form	Click chuột Trái, Phải, Con lăn trên các vị trí bất kì trên form trừ phần Title và Label thông báo	Thể hiện vị trí đã click
6	Kiểm tra click tại các vị trí chặn click	Click vào phần Title của form, phần label thông báo	Không thay đổi thông báo do các label này đã che phần Form
7	Kiểm tra Click kéo chuột	Nhấn chuột tại một vị trí, sau đó kéo sang vị trí khác rồi thả ra.  Click lại tại vị trí cuối để kiểm tra	Khi nhấn chuột không thay đổi thông báo, chỉ thay đổi khi nhả chuột

## IT008 – LẬP TRÌNH TRỰC QUAN

8	Kết hợp chuột và phím	<p>Nhấn giữ chuột, nhấn phím bất kỳ, sau đó mới nhả chuột</p> <p>Nhấn giữ một phím, click chuột, sang đó nhả phím</p>	<p>Khi giữ chuột, vẫn có thể nhận sự kiện bàn phím, khi nhả chuột, sự kiện chuột mới được gọi.</p> <p>Khi giữ phím, sự kiện click chuột hiện lên, sau đó trở về thông báo bàn phím lập tức</p>
---	-----------------------	---	--

– Video minh họa: <https://youtu.be/yNJM-C3Ms1U>

**Bài 2. Viết ứng dụng hiển thị ngày giờ hệ thống****Mô tả chương trình**

- Yêu cầu: Tạo chương trình hiển thị giờ trên hệ thống theo thời gian hiện tại
- Hướng xử lý:
  - Tạo một timer clock và sự kiện tick của clock để cập nhật thời gian theo giây. Chạy timer ngay khi Load form
  - Dùng Datetime.Now để lấy thời gian hiện tại, thông báo trên label mỗi khi sự kiện tick xảy ra

**Nội dung code***Bảng 2.1. Nội dung code bài 2***SimpleClock.cs**

```
using Microsoft.VisualBasic.Devices;

namespace Bai2
{
    public partial class SimpleClock : Form
    {
        public SimpleClock()
        {
            InitializeComponent();
        }

        private void clock_Tick(object sender, EventArgs e)
        {
            labelMain.Text = DateTime.Now.ToString("dddd, MMMM dd, yyyy HH:mm:ss tt");
        }

        private void SimpleClock_Load(object sender, EventArgs e)
        {
            clock.Start();
        }
    }
}
```

**SimpleClock.Design.cs**

```
namespace Bai2
{
    partial class SimpleClock
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```

```

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    components = new System.ComponentModel.Container();
    title = new Label();
    labelMain = new Label();
    clock = new System.Windows.Forms.Timer(components);
    SuspendLayout();
    //
    // title
    //
    title.BackColor = SystemColors.ActiveCaptionText;
    title.Dock = DockStyle.Top;
    title.Font = new Font("Segoe UI Black", 22.2F, FontStyle.Bold, GraphicsUnit.Point,
0);
    title.ForeColor = SystemColors.Control;
    title.Location = new Point(0, 0);
    title.Name = "title";
    title.Size = new Size(800, 60);
    title.TabIndex = 0;
    title.Text = "Simple Clock";
    title.TextAlign = ContentAlignment.MiddleCenter;
    //
    // labelMain
    //
    labelMain.Dock = DockStyle.Fill;
    labelMain.Font = new Font("Segoe UI", 12F, FontStyle.Regular, GraphicsUnit.Point,
0);
    labelMain.Location = new Point(0, 60);
    labelMain.Name = "labelMain";
    labelMain.Size = new Size(800, 390);
    labelMain.TabIndex = 1;
    labelMain.Text = "Ngày giờ";
    labelMain.TextAlign = ContentAlignment.MiddleCenter;
    //
    // clock
    //
    clock.Tick += clock_Tick;
    //
    // SimpleClock
    //
    AutoScaleDimensions = new.SizeF(8F, 20F);
    AutoScaleMode = AutoScaleMode.Font;
    ClientSize = new Size(800, 450);
    Controls.Add(labelMain);
    Controls.Add(title);
    Name = "SimpleClock";
    Text = "SimpleClock";
    Load += this.SimpleClock_Load;
    ResumeLayout(false);
}

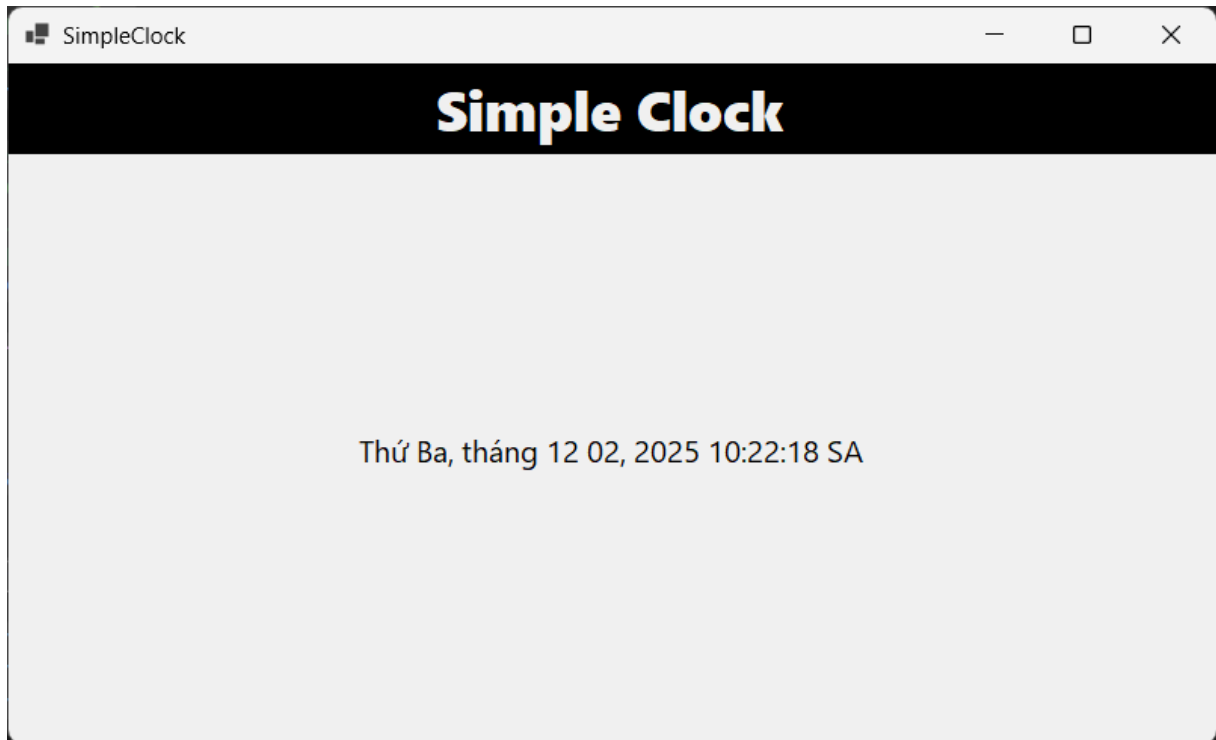
#endregion

private Label title;
private Label labelMain;
private System.Windows.Forms.Timer clock;
}

```

### Kiểm thử chương trình

- Giao diện chương trình:



– Kịch bản kiểm thử:

Bảng 2.2. Kiểm thử bài 2

STT	Tiêu chí	Hành động	Kết quả
1	Độ chính xác	Mở song song chương trình và Clock của hệ thống, quan sát độ trùng khớp về thời gian của hai chương trình	Thời gian thay đổi theo giây của chương trình và hệ thống khớp nhau
2	Cập nhật thời gian khi form không Active	Tạo một chương trình khác làm form chương trình không active nhưng vẫn thấy được thời gian trên form	Thời gian vẫn thay đổi
3	Cập nhật thời gian khi Minimize form	Thu nhỏ (minimize) form lại, xem trạng thái của form trên thanh taskbar.	Thời gian ngừng chạy, cho đến khi click mở lại form thì thời gian cập

## IT008 – LẬP TRÌNH TRỰC QUAN

		Click để mở lại chương trình tại thanh taskbar	nhật lại theo hiện tại
4	Thay đổi giờ hệ thống	Khi chương trình đang chạy, vào Settings của Windows thay đổi giờ	Chương trình phải cập nhật ngay lập tức theo giờ mới của hệ thống (không được đếm tiếp từ giờ cũ).

– Video minh họa: <https://youtu.be/nY4s40Qw5IA>

### Bài 3. Xây dựng chương trình mô phỏng điều khiển Windows media Player để phát các tập tin Audio và Video

#### Mô tả chương trình

- Yêu cầu:
  - Ứng dụng chứa Windows Media Player control cho phép Play các file video/sound theo nhiều dạng format: (\*.avi \*.mpeg \*.wav \*.midi \*.mp4 \*.mp3)
  - Menu Bar có 1 Menu File và 2 Sub Menu Open và Exit
  - Khi chọn Sub Menu Exit sẽ thoát ứng dụng
  - Khi chọn Sub Menu Open sẽ mở hộp thoại Open File để chọn mở file Media
  - Control Multi Media sẽ phát file Media đã chọn.
  - StatusStrip hiển thị: Ngày giờ hệ thống và thay đổi giờ theo mỗi giây.
- Hướng xử lý:
  - Tạo Timer và sự kiện Tick để cập nhật ngày giờ hiện tại lên StatusStrip mỗi giây. Kích hoạt Timer ngay khi khởi tạo Form
  - Xử lý sự kiện Open: Dùng OpenFileDialog để chọn file media với bộ lọc định dạng, gán đường dẫn file vào thuộc tính URL của Windows Media Player control để phát
  - Xử lý sự kiện Exit: Gọi lệnh Application.Exit() để thoát ứng dụng

#### Nội dung code

Bảng 3.1. Nội dung code bài 3

MediaPlayer.cs
<pre> namespace Bai3 {     public partial class MediaPlayer : Form     {         public MediaPlayer()         {             InitializeComponent();             timer1.Start();         }          private void openToolStripMenuItem_Click(object sender, EventArgs e)         {             OpenFileDialog dialog = new OpenFileDialog();             dialog.Filter = "Video/Sound File   *.avi; *.mpeg; *.wav; *.midi; *.mp4; *.mp3";              if (dialog.ShowDialog() == DialogResult.OK)             {                 this.Text = Path.GetFileName(dialog.FileName);                 axWindowsMediaPlayer1.URL = dialog.FileName;                 axWindowsMediaPlayer1.Ctlcontrols.play();             }         }          private void exitToolStripMenuItem_Click(object sender, EventArgs e)         { </pre>

```

        Application.Exit();
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        toolStripStatusLabel1.Text = "Hôm nay là ngày " +
        DateTime.Now.ToString("dd/MM/yyyy") + " - Bây giờ là " + DateTime.Now.ToString("HH:mm:ss
        tt").ToUpper();
    }
}

```

### MediaPlayer.Designer.cs

```

namespace Bai3
{
    partial class MediaPlayer
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            components = new System.ComponentModel.Container();
            System.ComponentModel.ComponentResourceManager resources = new
            System.ComponentModel.ComponentResourceManager(typeof(MediaPlayer));
            label1 = new Label();
            menuStrip1 = new MenuStrip();
            fileToolStripMenuItem = new ToolStripMenuItem();
            openToolStripMenuItem = new ToolStripMenuItem();
            exitToolStripMenuItem = new ToolStripMenuItem();
            statusStrip1 = new StatusStrip();
            toolStripStatusLabel1 = new ToolStripStatusLabel();
            axWindowsMediaPlayer1 = new AxWMPLib.AxWindowsMediaPlayer();
            timer1 = new System.Windows.Forms.Timer(components);
            menuStrip1.SuspendLayout();
            statusStrip1.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)axWindowsMediaPlayer1).BeginInit();
            SuspendLayout();
            //
            // label1
            //
            label1.BackColor = Color.MediumSeaGreen;
            label1.Dock = DockStyle.Top;
            label1.Font = new Font("Segoe UI Black", 22.2F, FontStyle.Bold,
            GraphicsUnit.Point, 0);
            label1.ForeColor = Color.Honeydew;
            label1.Location = new Point(0, 28);
            label1.Name = "label1";
            label1.Size = new Size(800, 60);
            label1.TabIndex = 0;
            label1.Text = "Media Player";
            label1.TextAlign = ContentAlignment.MiddleCenter;
            //
            // menuStrip1

```



```

//
menuStrip1.BackColor = Color.SeaGreen;
menuStrip1.ImageScalingSize = new Size(20, 20);
menuStrip1.Items.AddRange(new ToolStripItem[] { fileToolStripMenuItem });
menuStrip1.Location = new Point(0, 0);
menuStrip1.Name = "menuStrip1";
menuStrip1.Size = new Size(800, 28);
menuStrip1.TabIndex = 1;
menuStrip1.Text = "menuStrip1";
//
// fileToolStripMenuItem
//
fileToolStripMenuItem.DropDownItems.AddRange(new ToolStripItem[] {
openToolStripMenuItem, exitToolStripMenuItem });
fileToolStripMenuItem.Name = "fileToolStripMenuItem";
fileToolStripMenuItem.Size = new Size(46, 24);
fileToolStripMenuItem.Text = "&File";
//
// openToolStripMenuItem
//
openToolStripMenuItem.Name = "openToolStripMenuItem";
openToolStripMenuItem.Size = new Size(128, 26);
openToolStripMenuItem.Text = "&Open";
openToolStripMenuItem.Click += openToolStripMenuItem_Click;
//
// exitToolStripMenuItem
//
exitToolStripMenuItem.Name = "exitToolStripMenuItem";
exitToolStripMenuItem.Size = new Size(128, 26);
exitToolStripMenuItem.Text = "&Exit";
exitToolStripMenuItem.Click += exitToolStripMenuItem_Click;
//
// statusStrip1
//
statusStrip1.BackColor = Color.SeaGreen;
statusStrip1.ImageScalingSize = new Size(20, 20);
statusStrip1.Items.AddRange(new ToolStripItem[] { toolStripStatusLabel1 });
statusStrip1.Location = new Point(0, 424);
statusStrip1.Name = "statusStrip1";
statusStrip1.Size = new Size(800, 26);
statusStrip1.TabIndex = 2;
statusStrip1.Text = "statusStrip1";
//
// toolStripStatusLabel1
//
toolStripStatusLabel1.Name = "toolStripStatusLabel1";
toolStripStatusLabel1.Size = new Size(151, 20);
toolStripStatusLabel1.Text = "toolStripStatusLabel1";
//
// axWindowsMediaPlayer1
//
axWindowsMediaPlayer1.Dock = DockStyle.Fill;
axWindowsMediaPlayer1.Enabled = true;
axWindowsMediaPlayer1.Location = new Point(0, 88);
axWindowsMediaPlayer1.Name = "axWindowsMediaPlayer1";
axWindowsMediaPlayer1.OcxState =
(AxHost.State)resources.GetObject("axWindowsMediaPlayer1.OcxState");
axWindowsMediaPlayer1.Size = new Size(800, 336);
axWindowsMediaPlayer1.TabIndex = 3;
//
// timer1
//
timer1.Tick += timer1_Tick;
//
// MediaPlayer
//
AutoScaleDimensions = new.SizeF(8F, 20F);
AutoScaleMode = AutoScaleMode.Font;
ClientSize = new Size(800, 450);
Controls.Add(axWindowsMediaPlayer1);
Controls.Add(statusStrip1);
Controls.Add(label1);
Controls.Add(menuStrip1);
MainMenuStrip = menuStrip1;
Name = "MediaPlayer";
menuStrip1.ResumeLayout(false);
menuStrip1.PerformLayout();
statusStrip1.ResumeLayout(false);

```

```

        statusStrip1.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)axWindowsMediaPlayer1).EndInit();
        ResumeLayout(false);
        PerformLayout();
    }

    #endregion

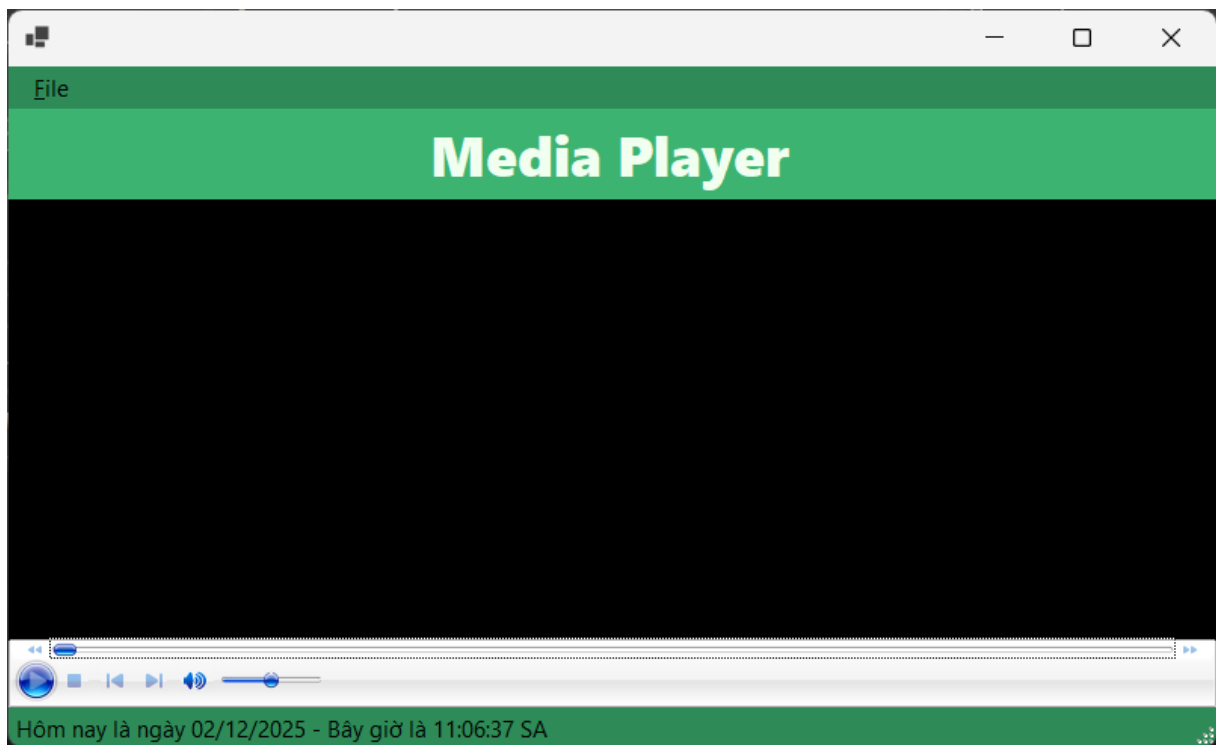
    private Label label1;
    private MenuStrip menuStrip1;
    private ToolStripMenuItem fileToolStripMenuItem;
    private ToolStripMenuItem openToolStripMenuItem;
    private ToolStripMenuItem exitToolStripMenuItem;
    private StatusStrip statusStrip1;
    private ToolStripStatusLabel toolStripStatusLabel1;
    private AxWMPLib.AxWindowsMediaPlayer axWindowsMediaPlayer1;
    private System.Windows.Forms.Timer timer1;
}

```

### Kiểm thử chương trình

- Giao diện chương trình:

Hình 3.1. Giao diện chính bài 3



- Kịch bản kiểm thử:

Bảng 3.2. Kiểm thử bài 3

STT	Tiêu chí	Hành động	Kết quả
1	Thời gian	Mở chương trình, quan sát thời gian trên thanh trạng thái và thời gian hệ thống	StatusStrip hiện đúng định dạng ngày giờ hệ thống.

## IT008 – LẬP TRÌNH TRỰC QUAN

			Giây thay đổi liên tục mỗi giây.
2	Mở file	Click vào chức năng mở file, kiểm tra các file được lọc đúng định dạng có thể mở.  Kiểm tra trường hợp khi click cancel	Các file được lọc theo các đuôi *.avi *.mpeg *.wav *.midi *.mp4 *.mp3  Form vẫn giữ trạng thái cũ (trống hoặc chạy video cũ) khi cửa sổ mở file đóng lại.
3	Phát video	Mở một file đuôi *.mp4.  Kiểm tra các chức năng tua (seek), âm lượng (volume), tắt âm (mute), tạm dừng (pause), dừng (stop), tiếp (play), tua lại (rewind) và tua nhanh (fast forward)	Các chức năng thực hiện đúng như các chương trình điều khiển Video
4	Phát audio	Mở một file âm thanh đuôi *.mp3.  Kiểm tra các chức năng như video	Thực hiện tốt được các chức năng.
5	Thoát chương trình	Đóng cửa sổ hoặc chọn File > Exit để đóng chương trình	Chương trình đóng thành công.

– Video minh họa: [https://youtu.be/M\\_QD1Gg46KI](https://youtu.be/M_QD1Gg46KI)

***Mô tả chương trình***

– Yêu cầu:

- Chương trình soạn thảo văn bản có chức năng soạn thảo, lưu – mở file, định dạng font chữ
- Sử dụng RichTextBox để thiết kế điều khiển hiển thị và nhập nội dung văn bản
- Sử dụng công cụ MenuStrip để tạo Hệ Thống và Định Dạng
  - + Trong menu Hệ thống có chức năng tạo văn bản (Ctrl + N), mở file, lưu nội dung (Ctrl + S) và Thoát chương trình.
  - + Click vào menu Định dạng sẽ gọi form FontDialog có sẵn của windows
- Sử dụng công cụ **ToolStrip** để tạo thanh công cụ chứa các button và image
- Khi mới mở Form Soạn thảo văn bản, thực hiện:
  - + Tạo dữ liệu cho ComboBox Font: chứa tất cả các Font chữ của hệ thống.
  - + Tạo dữ liệu cho ComboBox Size: chứa các giá trị từ 8 → 72 ( 8, 9, 10, 11, 12, 14, 16, 18, 20, 22, 24, 26, 28, 36, 48, 72)
  - + Tạo giá trị mặc định là Font Tahoma, Size 14
- Khi chọn Tạo văn bản mới (hoặc nhấn nút Tạo mới): Xóa nội dung hiện có trên RichTextBox và khởi tạo các giá trị mặc định như Font, Size, ...
- Khi chọn Mở tập tin (hoặc nhấn nút Mở): Hiển thị hộp thoại mở tập tin (OpenFileDialog) cho phép người dùng chọn tập tin văn bản (\*.txt hoặc \*.rtf) để mở.
- Khi chọn Lưu nội dung văn bản (hoặc nhấn nút Lưu): Lưu nội dung văn bản trên RichTextBox xuống tập tin. Nếu là văn bản mới và trước đó chưa lưu lần nào thì hiển thị hộp thoại lưu tập tin (SaveFileDialog) cho phép người dùng chọn thư mục cần lưu tập tin với kiểu tập tin cần lưu là \*.rtf.
- Nếu là văn bản đã được mở trước đó thì thông báo cho người dùng lưu văn bản thành công.
- Xử lý các chức năng **B**, ***I***, **U**

– Hướng xử lý:

- Thiết kế giao diện theo yêu cầu, thiết lập các giá trị mặc định khi khởi tạo form
- Chức năng định dạng:
  - + Khi chọn văn bản, sự kiện Selection Change của Rich Text Box được gọi, xử lý hiển thị font hiện tại của chữ được chọn lên tool bar.
  - + Xử lý định dạng chữ khi chọn bằng FontDialog

## IT008 – LẬP TRÌNH TRỰC QUAN

- + Xử lý sự kiện TextChange cho combo box Font và Size cùng thực hiện định dạng font nếu 1 trong 2 thay đổi.
- + Thêm và xóa định dạng khi Click vào nút B, I và U
- Chức năng hệ thống:
  - + Trước khi tạo mới file (New), kiểm tra nếu có văn bản đã soạn thảo thì hỏi cần lưu văn bản đã soạn không. Nếu người dùng chọn "Yes" thì thực hiện lưu, sau đó làm sạch khung soạn thảo (RichTextBox), đặt lại tiêu đề form về mặc định ("Untitled.txt") và reset các biến trạng thái
  - + Xử lý sự kiện Save:
    - Kiểm tra biến lưu đường dẫn file (textFile). Nếu biến rỗng (chưa từng lưu) thì gọi hàm SaveAs.
    - Nếu đã có đường dẫn, khởi tạo StreamWriter để ghi đè nội dung từ RichTextBox vào file đó.
  - + Xử lý hàm SaveAs:
    - Hiện thị hộp thoại SaveFileDialog để người dùng chọn nơi lưu và đặt tên.
    - Nếu người dùng ấn OK, cập nhật đường dẫn file, cập nhật tiêu đề Form và thực hiện ghi nội dung vào file mới.
  - + Xử lý sự kiện Open:
    - Hiện thị hộp thoại OpenFileDialog.
    - Nếu người dùng chọn file và ấn OK: Trước khi mở, kiểm tra xem văn bản hiện tại có thay đổi không để hỏi lưu (tương tự chức năng New).
    - Sau đó, đọc toàn bộ nội dung file được chọn bằng File.ReadAllText, hiển thị lên RichTextBox và cập nhật tiêu đề Form. Có bắt lỗi (try-catch) khi đọc file.
  - + Xử lý sự kiện Exit:
    - Kiểm tra khung soạn thảo có dữ liệu không. Nếu có, hiển thị MessageBox với 3 lựa chọn (Yes/No/Cancel):
      - Yes: Lưu file rồi thoát.
      - No: Thoát không lưu.
      - Cancel: Hủy lệnh thoát, quay lại chương trình.
    - Cách xử lý tương tự khi chọn chức năng Thoát trong Hệ thống, hoặc nhất X trực tiếp để đóng form

*Bảng 4.1. Nội dung code bài 4***TextEditor.cs**

```

namespace Bai4
{
    public partial class TextEditor : Form
    {
        static string textFile = "";
        static bool isOpen = false;
        bool isUpdatingUI = false;

        public TextEditor()
        {
            InitializeComponent();
            InitializeCombobox();
            richTextBox.SelectionChanged += new EventHandler(richTextBox_SelectionChanged);
        }

        private void InitializeCombobox()
        {
            foreach (FontFamily font in System.Drawing.FontFamily.Families)
            {
                toolStripComboBoxFont.Items.Add(font.Name);
            }

            toolStripComboBoxSize.Items.AddRange(new object[] { 8, 9, 10, 11, 12, 14, 16, 18,
20, 22, 24, 26, 28, 36, 48, 72 });

            toolStripComboBoxFont.Text = "Tahoma";
            toolStripComboBoxSize.Text = "14";
        }

        private void richTextBox_SelectionChanged(object sender, EventArgs e)
        {
            isUpdatingUI = true;

            try
            {
                if (richTextBox.SelectionFont != null)
                {
                    toolStripComboBoxFont.Text = richTextBox.SelectionFont.Name;
                    toolStripComboBoxSize.Text =
((int)richTextBox.SelectionFont.Size).ToString();

                    toolStripButtonBold.Checked = richTextBox.SelectionFont.Bold;
                    toolStripButtonItalic.Checked = richTextBox.SelectionFont.Italic;
                    toolStripButtonUnderline.Checked = richTextBox.SelectionFont.Underline;
                }
                else
                {
                    toolStripComboBoxFont.Text = "";
                    toolStripComboBoxSize.Text = "";
                    toolStripButtonBold.Checked = false;
                    toolStripButtonItalic.Checked = false;
                    toolStripButtonUnderline.Checked = false;
                }
            }
            finally
            {
                isUpdatingUI = false;
            }
        }

        private void formatToolStripMenuItem_Click(object sender, EventArgs e)
        {
            FontDialog fontdialog = new FontDialog();
            fontdialog.ShowColor = true;
            fontdialog.ShowApply = true;
            fontdialog.ShowEffects = true;
            fontdialog.ShowHelp = true;
            if (fontdialog.ShowDialog() == DialogResult.OK)

```

```

        {
            richTextBox.SelectionFont = fontdialog.Font;
            richTextBox.SelectionColor = fontdialog.Color;
        }
    }

    private void changeFontStyle(object sender, EventArgs e)
    {
        if (isUpdatingUI) return;

        try
        {
            string fontName = toolStripComboBoxFont.Text;
            string fontSizeStr = toolStripComboBoxSize.Text;

            if (!string.IsNullOrEmpty(fontName) && !string.IsNullOrEmpty(fontSizeStr))
            {
                float fontSize = float.Parse(fontSizeStr);

                FontStyle currentStyle = FontStyle.Regular;
                if (richTextBox.SelectionFont != null)
                {
                    currentStyle = richTextBox.SelectionFont.Style;
                }

                richTextBox.SelectionFont = new Font(fontName, fontSize, currentStyle);
            }
        }
        catch
        {
            return;
        }
    }

    private void toolStripButtonBold_Click(object sender, EventArgs e)
    {
        if (richTextBox.SelectionFont != null)
        {
            Font font = richTextBox.SelectionFont;
            FontStyle fs = font.Style;

            if (font.Bold)
                fs &= ~FontStyle.Bold;
            else
                fs |= FontStyle.Bold;

            richTextBox.SelectionFont = new Font(font.FontFamily, font.Size, fs);

            toolStripButtonBold.Checked = !toolStripButtonBold.Checked;
        }
    }

    private void toolStripButtonItalic_Click(object sender, EventArgs e)
    {
        if (richTextBox.SelectionFont != null)
        {
            Font font = richTextBox.SelectionFont;
            FontStyle fs = font.Style;

            if (font.Italic)
                fs &= ~FontStyle.Italic;
            else
                fs |= FontStyle.Italic;

            richTextBox.SelectionFont = new Font(font.FontFamily, font.Size, fs);

            toolStripButtonItalic.Checked = !toolStripButtonItalic.Checked;
        }
    }

    private void toolStripButtonUnderline_Click(object sender, EventArgs e)
    {
        if (richTextBox.SelectionFont != null)
        {
            Font font = richTextBox.SelectionFont;
            FontStyle fs = font.Style;

            if (font.Underline)

```

```

        fs &= ~FontStyle.Underline;
    else
        fs |= FontStyle.Underline;

    richTextBox.SelectionFont = new Font(font.FontFamily, font.Size, fs);
    toolStripButtonUnderline.Checked = !toolStripButtonUnderline.Checked;
}

private void toolStripButtonNew_Click(object sender, EventArgs e)
{
    if (richTextBox.Text != "")
    {
        DialogResult dr = MessageBox.Show("Bạn có muốn lưu các thay đổi?", "Các thay
đổi chưa được lưu lại", MessageBoxButtons.YesNo);
        switch (dr)
        {
            case DialogResult.Yes:
                if (isOpen)
                    toolStripButtonSave_Click(sender, e);
                else SaveAs();
                break;
            case DialogResult.No:
                break;
        }
    }
    this.Text = "Untitled.rtf";
    richTextBox.Text = "";
    toolStripComboBoxFont.Text = "Tahoma";
    toolStripComboBoxSize.Text = "14";
    textFile = "";
    isOpen = false;
}

private void toolStripButtonSave_Click(object sender, EventArgs e)
{
    if (textFile == "")
        SaveAs();
    else
    {
        try
        {
            // Kiểm tra đuôi file để chọn chế độ lưu phù hợp
            if (Path.GetExtension(textFile).ToLower() == ".txt")
            {
                richTextBox.SaveFile(textFile, RichTextBoxStreamType.PlainText);
            }
            else
            {
                // Mặc định lưu bao gồm định dạng (RTF)
                richTextBox.SaveFile(textFile, RichTextBoxStreamType.RichText);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lỗi lưu file: " + ex.Message);
        }
    }
}

private void SaveAs()
{
    SaveFileDialog saveTextFile = new SaveFileDialog();
    saveTextFile.Filter = "Rich Text Format (*.rtf)|*.rtf|Text file (*.txt)|*.txt|All
files (*.*)|*.*";

    if (saveTextFile.ShowDialog() == DialogResult.OK)
    {
        textFile = saveTextFile.FileName;
        isOpen = true;
        this.Text = textFile;

        toolStripButtonSave_Click(null, null);
    }
}

//System menu strip event handlers

```



```

private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    toolStripButtonNew_Click(sender, e);
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    toolStripButtonSave_Click(sender, e);
}

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    OpenFileDialog openTextFile = new OpenFileDialog();
    openTextFile.Filter = "Rich Text Format (*.rtf)|*.rtf|Text file (*.txt)|*.txt|Document file (*.doc)|*.doc|All files (*.*)|*.*";
    if (openTextFile.ShowDialog() == DialogResult.OK)
    {
        if (richTextBox.Text != "")
        {
            DialogResult dr = MessageBox.Show("Bạn có muốn lưu các thay đổi của tệp tin cũ?", "Các thay đổi chưa được lưu lại", MessageBoxButtons.YesNo);
            if (dr == DialogResult.Yes)
            {
                if (isOpen)
                    toolStripButtonSave_Click(sender, e);
                else SaveAs();
            }
        }

        textFile = openTextFile.FileName;
        this.Text = textFile;
        isOpen = true;

        try
        {
            if (Path.GetExtension(textFile).ToLower() == ".txt")
            {
                richTextBox.LoadFile(textFile, RichTextBoxStreamType.PlainText);
            }
            else
            {
                richTextBox.LoadFile(textFile, RichTextBoxStreamType.RichText);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Lỗi đọc file hoặc định dạng không hợp lệ: " +
ex.Message);
        }
    }
}

private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void TextEditor_FormClosing(object sender, FormClosingEventArgs e)
{
    if (richTextBox.Text != "")
    {
        DialogResult dr = MessageBox.Show("Bạn có muốn lưu các thay đổi trước khi thoát?", "Thoát", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
        if (dr == DialogResult.Yes)
        {
            toolStripButtonSave_Click(sender, e);
        }
        else if (dr == DialogResult.Cancel)
        {
            e.Cancel = true;
        }
    }
}
}

```

}

## TextEditor.Designer.cs

```

namespace Bai4
{
    partial class TextEditor
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources = new
            System.ComponentModel.ComponentResourceManager(typeof(TextEditor));
            menuStrip1 = new MenuStrip();
            systemToolStripMenuItem = new ToolStripMenuItem();
            newToolStripMenuItem = new ToolStripMenuItem();
            openToolStripMenuItem = new ToolStripMenuItem();
            toolStripSeparator2 = new ToolStripSeparator();
            saveToolStripMenuItem = new ToolStripMenuItem();
            exitToolStripMenuItem = new ToolStripMenuItem();
            formatToolStripMenuItem = new ToolStripMenuItem();
            toolStrip1 = new ToolStrip();
            toolStripButtonNew = new ToolStripButton();
            toolStripButtonSave = new ToolStripButton();
            toolStripSeparator1 = new ToolStripSeparator();
            toolStripComboBoxFont = new ToolStripComboBox();
            toolStripComboBoxSize = new ToolStripComboBox();
            toolStripButtonBold = new ToolStripButton();
            toolStripButtonItalic = new ToolStripButton();
            toolStripButtonUnderline = new ToolStripButton();
            labelTitle = new Label();
            richTextBox = new RichTextBox();
            menuStrip1.SuspendLayout();
            toolStrip1.SuspendLayout();
            SuspendLayout();
            //
            // menuStrip1
            //
            menuStrip1.BackColor = Color.SteelBlue;
            menuStrip1.ImageScalingSize = new Size(20, 20);
            menuStrip1.Items.AddRange(new ToolStripItem[] { systemToolStripMenuItem,
            formatToolStripMenuItem });
            menuStrip1.Location = new Point(0, 0);
            menuStrip1.Name = "menuStrip1";
            menuStrip1.Size = new Size(800, 28);
            menuStrip1.TabIndex = 0;
            menuStrip1.Text = "menuStrip1";
            //
            // systemToolStripMenuItem
            //
        }
    }

```

```

        systemToolStripMenuItem.DropDownItems.AddRange(new ToolStripItem[] {
newToolStripMenuItem, openToolStripMenuItem, toolStripSeparator2, saveToolStripMenuItem,
exitToolStripMenuItem });
        systemToolStripMenuItem.Name = "systemToolStripMenuItem";
        systemToolStripMenuItem.Size = new Size(85, 24);
        systemToolStripMenuItem.Text = "Hệ thống";
        //
        // newToolStripMenuItem
        //
        newToolStripMenuItem.Image =
(Image)resources.GetObject("newToolStripMenuItem.Image");
        newToolStripMenuItem.Name = "newToolStripMenuItem";
        newToolStripMenuItem.ShortcutKeys = Keys.Control | Keys.N;
        newToolStripMenuItem.Size = new Size(285, 26);
        newToolStripMenuItem.Text = "Tạo văn bản mới ";
        newToolStripMenuItem.Click += newToolStripMenuItem_Click;
        //
        // openToolStripMenuItem
        //
        openToolStripMenuItem.Image =
(Image)resources.GetObject("openToolStripMenuItem.Image");
        openToolStripMenuItem.Name = "openToolStripMenuItem";
        openToolStripMenuItem.Size = new Size(285, 26);
        openToolStripMenuItem.Text = "Mở tập tin";
        openToolStripMenuItem.Click += openToolStripMenuItem_Click;
        //
        // toolStripSeparator2
        //
        toolStripSeparator2.Name = "toolStripSeparator2";
        toolStripSeparator2.Size = new Size(282, 6);
        //
        // saveToolStripMenuItem
        //
        saveToolStripMenuItem.Image =
(Image)resources.GetObject("saveToolStripMenuItem.Image");
        saveToolStripMenuItem.Name = "saveToolStripMenuItem";
        saveToolStripMenuItem.ShortcutKeys = Keys.Control | Keys.S;
        saveToolStripMenuItem.Size = new Size(285, 26);
        saveToolStripMenuItem.Text = "Lưu nội dung văn bản";
        saveToolStripMenuItem.Click += saveToolStripMenuItem_Click;
        //
        // exitToolStripMenuItem
        //
        exitToolStripMenuItem.Name = "exitToolStripMenuItem";
        exitToolStripMenuItem.Size = new Size(285, 26);
        exitToolStripMenuItem.Text = "Thoát";
        exitToolStripMenuItem.Click += exitToolStripMenuItem_Click;
        //
        // formatToolStripMenuItem
        //
        formatToolStripMenuItem.Name = "formatToolStripMenuItem";
        formatToolStripMenuItem.Size = new Size(92, 24);
        formatToolStripMenuItem.Text = "&Định dạng";
        formatToolStripMenuItem.Click += formatToolStripMenuItem_Click;
        //
        // toolStrip1
        //
        toolStrip1.BackColor = Color.SteelBlue;
        toolStrip1.ImageScalingSize = new Size(20, 20);
        toolStrip1.Items.AddRange(new ToolStripItem[] { toolStripButtonNew,
toolStripButtonSave, toolStripSeparator1, toolStripComboBoxFont, toolStripComboBoxSize,
toolStripButtonBold, toolStripButtonItalic, toolStripButtonUnderline });
        toolStrip1.Location = new Point(0, 28);
        toolStrip1.Name = "toolStrip1";
        toolStrip1.RenderMode = ToolStripRenderMode.System;
        toolStrip1.Size = new Size(800, 28);
        toolStrip1.TabIndex = 1;
        toolStrip1.Text = "toolStrip1";
        //
        // toolStripButtonNew
        //
        toolStripButtonNew.DisplayStyle = ToolStripItemDisplayStyle.Image;
        toolStripButtonNew.Image = (Image)resources.GetObject("toolStripButtonNew.Image");
        toolStripButtonNew.ImageTransparentColor = Color.Magenta;
        toolStripButtonNew.Name = "toolStripButtonNew";
        toolStripButtonNew.Size = new Size(29, 25);
        toolStripButtonNew.Text = "toolStripButton1";
        toolStripButtonNew.Click += toolStripButtonNew_Click;

```

```

//
// toolStripButtonSave
//
toolStripButtonSave.DisplayStyle = ToolStripItemDisplayStyle.Image;
toolStripButtonSave.Image =
(Image)resources.GetObject("toolStripButtonSave.Image");
toolStripButtonSave.ImageTransparentColor = Color.Magenta;
toolStripButtonSave.Name = "toolStripButtonSave";
toolStripButtonSave.Size = new Size(29, 25);
toolStripButtonSave.Text = "toolStripButton2";
toolStripButtonSave.Click += toolStripButtonSave_Click;
//
// toolStripSeparator1
//
toolStripSeparator1.Name = "toolStripSeparator1";
toolStripSeparator1.Size = new Size(6, 28);
//
// toolStripComboBoxFont
//
toolStripComboBoxFont.Name = "toolStripComboBoxFont";
toolStripComboBoxFont.Size = new Size(121, 28);
toolStripComboBoxFont.TextChanged += changeFontStyle;
//
// toolStripComboBoxSize
//
toolStripComboBoxSize.Name = "toolStripComboBoxSize";
toolStripComboBoxSize.Size = new Size(80, 28);
toolStripComboBoxSize.OwnerChanged += changeFontStyle;
toolStripComboBoxSize.TextChanged += changeFontStyle;
//
// toolStripButtonBold
//
toolStripButtonBold.DisplayStyle = ToolStripItemDisplayStyle.Image;
toolStripButtonBold.Image =
(Image)resources.GetObject("toolStripButtonBold.Image");
toolStripButtonBold.ImageTransparentColor = Color.Magenta;
toolStripButtonBold.Name = "toolStripButtonBold";
toolStripButtonBold.Size = new Size(29, 25);
toolStripButtonBold.Text = "toolStripButton1";
toolStripButtonBold.Click += toolStripButtonBold_Click;
//
// toolStripButtonItalic
//
toolStripButtonItalic.DisplayStyle = ToolStripItemDisplayStyle.Image;
toolStripButtonItalic.Image =
(Image)resources.GetObject("toolStripButtonItalic.Image");
toolStripButtonItalic.ImageTransparentColor = Color.Magenta;
toolStripButtonItalic.Name = "toolStripButtonItalic";
toolStripButtonItalic.Size = new Size(29, 25);
toolStripButtonItalic.Text = "toolStripButton1";
toolStripButtonItalic.Click += toolStripButtonItalic_Click;
//
// toolStripButtonUnderline
//
toolStripButtonUnderline.DisplayStyle = ToolStripItemDisplayStyle.Image;
toolStripButtonUnderline.Image =
(Image)resources.GetObject("toolStripButtonUnderline.Image");
toolStripButtonUnderline.ImageTransparentColor = Color.Magenta;
toolStripButtonUnderline.Name = "toolStripButtonUnderline";
toolStripButtonUnderline.Size = new Size(29, 25);
toolStripButtonUnderline.Text = "toolStripButton1";
toolStripButtonUnderline.Click += toolStripButtonUnderline_Click;
//
// labelTitle
//
labelTitle.BackColor = Color.LightSkyBlue;
labelTitle.Dock = DockStyle.Top;
labelTitle.Font = new Font("Segoe UI Black", 22.2F, FontStyle.Bold,
GraphicsUnit.Point, 0);
labelTitle.Location = new Point(0, 56);
labelTitle.Name = "labelTitle";
labelTitle.Size = new Size(800, 60);
labelTitle.TabIndex = 3;
labelTitle.Text = "Trình soạn thảo văn bản";
labelTitle.TextAlign = ContentAlignment.MiddleCenter;
//
// richTextBox
//

```

```

        richTextBox.BorderStyle = BorderStyle.None;
        richTextBox.Dock = DockStyle.Fill;
        richTextBox.Location = new Point(0, 116);
        richTextBox.Name = "richTextBox";
        richTextBox.Size = new Size(800, 334);
        richTextBox.TabIndex = 4;
        richTextBox.Text = "";
        richTextBox.SelectionChanged += richTextBox_SelectionChanged;
        //
        // TextEditor
        //
        AutoScaleDimensions = new.SizeF(8F, 20F);
        AutoScaleMode = AutoScaleMode.Font;
        ClientSize = new Size(800, 450);
        Controls.Add(richTextBox);
        Controls.Add(labelTitle);
        Controls.Add(toolStrip1);
        Controls.Add(menuStrip1);
        MainMenuStrip = menuStrip1;
        Name = "TextEditor";
        Text = "Soạn thảo văn bản";
        FormClosing += TextEditor_FormClosing;
        menuStrip1.ResumeLayout(false);
        menuStrip1.PerformLayout();
        toolStrip1.ResumeLayout(false);
        toolStrip1.PerformLayout();
        ResumeLayout(false);
        PerformLayout();
    }

#endregion

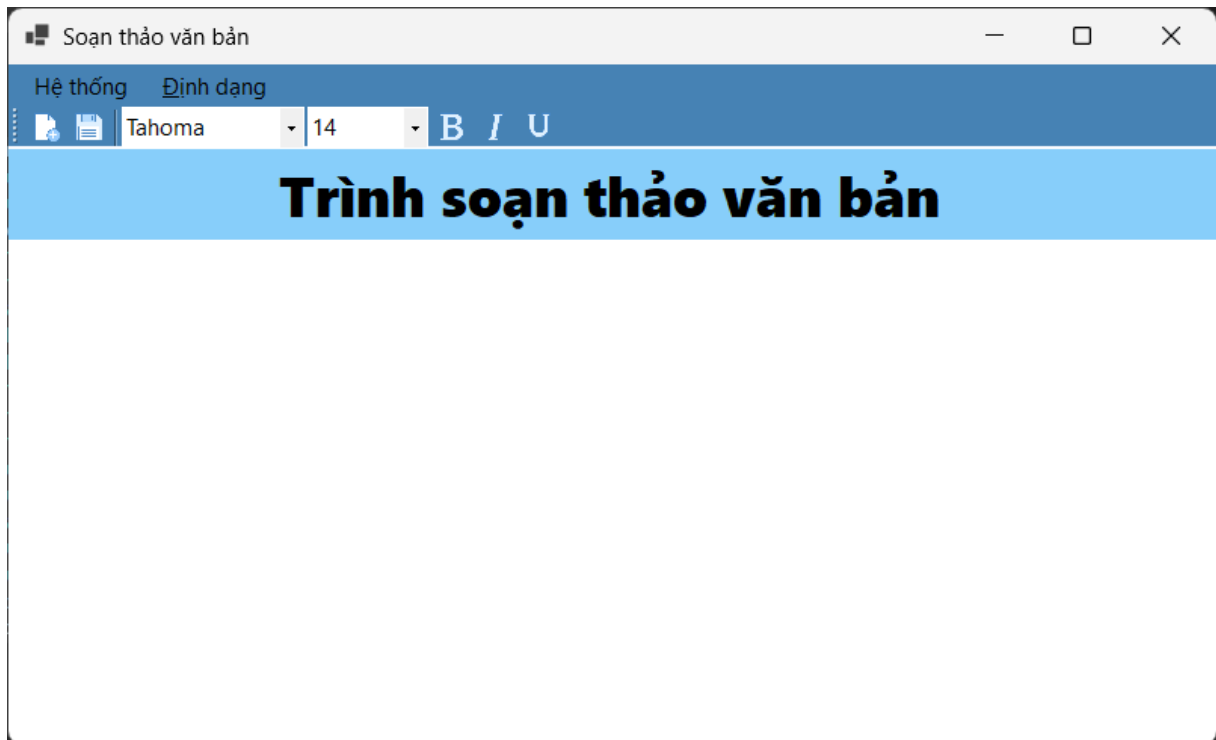
private MenuStrip menuStrip1;
private ToolStripMenuItem systemToolStripMenuItem;
private ToolStripMenuItem formatToolStripMenuItem;
private ToolStrip toolStrip1;
private ToolStripButton toolStripButtonNew;
private ToolStripButton toolStripButtonSave;
private ToolStripSeparator toolStripSeparator1;
private ToolStripComboBox toolStripComboBoxFont;
private ToolStripComboBox toolStripComboBoxSize;
private ToolStripButton toolStripButtonBold;
private ToolStripButton toolStripButtonItalic;
private ToolStripButton toolStripButtonUnderline;
private ToolStripMenuItem newToolStripMenuItem;
private ToolStripMenuItem openToolStripMenuItem;
private ToolStripSeparator toolStripSeparator2;
private ToolStripMenuItem saveToolStripMenuItem;
private ToolStripMenuItem exitToolStripMenuItem;
private Label labelTitle;
private RichTextBox richTextBox;
}
}

```

IT008 – LẬP TRÌNH TRỰC QUAN  
*Kiểm thử chương trình*

- Giao diện chương trình:

*Hình 4.1. Giao diện chính bài 4*



- Kịch bản kiểm thử:

*Bảng 4.2. Kiểm thử bài 4*

STT	Tiêu chí	Hành động	Kết quả
1	Soạn thảo	Tạo chương trình và viết văn bản bằng các kí tự thường và đặc biệt khác nhau	Có thể hiển thị các kí tự đặc biệt theo font mặc định
2	Đổi font chữ bằng toolbar	Chọn các kí tự cùng kiểu định dạng, thay đổi font, size, click vào các button <b>B</b> , <i>I</i> , <u>U</u> .	Văn bản được chọn thay đổi hình thức hiển thị (Font, Cỡ, Kiểu) tương ứng ngay lập tức. Các thuộc tính không bị thay đổi được giữ nguyên.

## IT008 – LẬP TRÌNH TRỰC QUAN

3	Cập nhật định dạng hiện tại trên tool bar	Chọn các văn bản đã thay đổi định dạng, có định dạng giống nhau và khác nhau	Các ô ComboBox (Font, Size) và nút bấm ( <b>B</b> , <i>I</i> , <u>U</u> ) tự động cập nhật để hiển thị đúng thông số của văn bản đang được bôi đen. Nếu chọn vùng văn bản hỗn hợp, ô Font/Size để trống, nút B/I/U sáng lên nếu toàn bộ vùng chọn đều có kiểu đó, nếu không cùng kiểu thì tắt tất cả.
4	Đổi định dạng chữ khi chọn nhiều kí tự khác định dạng	Chọn nhiều kí tự có định dạng khác nhau, sau đó định dạng lại kiểu mới giống nhau	Toàn bộ văn bản được chọn sẽ đồng nhất về thuộc tính vừa thay đổi (ví dụ: tất cả về size 14), bất kể định dạng gốc của chúng khác nhau thế nào
5	Tạo mới (có và chưa có dữ liệu)	Click nút <b>New</b> (Ctrl + N) khi khung soạn thảo trống.  Nhập văn bản, sau đó click nút <b>New</b> . Hộp thoại hỏi lưu xuất hiện, chọn các button Yes hoặc No	Khi khung trống, không có gì thay đổi, form được làm mới về các giá trị mặc định.  Hộp thoại hỏi lưu xuất hiện. Nếu chọn Yes, thực hiện lưu (Save hoặc SaveAs), nếu No thì xóa trắng nội dung

			ngay, form làm mới về các giá trị mặc định.
6	Lưu (lưu mới SaveAs và lưu cũ Save)	Click <b>Save</b> khi đang soạn thảo văn bản mới (chưa từng lưu).  Mở một file có sẵn, sửa nội dung, rồi click <b>Save</b> .	Hiển thị hộp thoại <b>SaveFileDialog</b> . Sau khi chọn nơi lưu và tên, file được tạo trên ổ cứng, tiêu đề Form cập nhật theo đường dẫn file.  Nếu đã lưu (mở file mới, đã SaveAs một lần), không hiện hộp thoại, dữ liệu mới được ghi đè trực tiếp vào file cũ.
7	Mở file	Click nút <b>Open</b> , chọn một file định dạng .rtf hoặc .txt từ máy tính	Nếu văn bản hiện tại chưa lưu, hiện thông báo hỏi lưu.  Nội dung file được load chính xác vào khung soạn thảo (giữ nguyên màu sắc, font chữ nếu là file .rtf).  Tiêu đề Form cập nhật thành đường dẫn của file vừa mở.
8	Thoát (có thay đổi và không thay đổi)	Click vào nút <b>Thoát</b> (trên hệ thống) hoặc nút <b>X</b> (góc phải trên cùng form).	Nếu văn bản trống hoặc đã lưu: Chương



## IT008 – LẬP TRÌNH TRỰC QUAN

			<p>trình đóng ngay lập tức.</p> <p>Nếu văn bản có thay đổi chưa lưu: Hiện thị hộp thoại Yes/No/Cancel. Nếu Yes, thực hiện lưu rồi đóng, nếu No thì đóng và không lưu, còn Cancel thì hủy lệnh đóng</p>
--	--	--	--

- Video minh họa: [https://youtu.be/Fs\\_iycQN2g8](https://youtu.be/Fs_iycQN2g8)

**Mô tả chương trình**

– Yêu cầu:

- Form Quản lý sinh viên có Menu Chức năng gồm 2 SubMenu là Thêm mới và Thoát
- Sử dụng MenuStrip để tạo menu
- Sử dụng ToolStrip để tạo hình và nút bấm Thêm mới, Tìm kiếm theo tên
- Khi click vào menu thêm mới (Ctrl + N) hoặc ở icon hình thêm mới trên toolStrip thì gọi Form nhập liệu sinh viên
- Ở phần tìm kiếm theo tên sinh viên, khi Textbox tìm kiếm thay đổi thì luôn luôn tìm lại dữ liệu chứa tên tìm kiếm (không phân biệt hoa thường)

– Hướng xử lý:

- Thiết kế giao diện và khởi tạo:
  - + Thiết kế Form chính với MenuStrip, ToolStrip và DataGridView.
  - + Thiết kế Form nhập liệu (AddNewForm) để nhận thông tin sinh viên.
- Chức năng Thêm mới (Quản lý luồng dữ liệu giữa 2 Form):
  - + **Tại Form chính:** Khi click menu "Thêm mới" hoặc nút Add trên ToolStrip, khởi tạo và gọi AddNewForm dưới dạng hộp thoại (ShowDialog).
  - + Tại Form nhập liệu (AddNewForm):
    - *Hàm kiểm tra (ValidInfo):* Duyệt từng ký tự để đảm bảo MSSV chỉ chứa số, Tên không chứa số, và Điểm phải là số thực từ 0-10.
    - *Nút Thêm:* Gọi hàm kiểm tra. Nếu dữ liệu đúng, khởi tạo đối tượng Student và đóng form trả về kết quả. Nếu sai, hiển thị thông báo lỗi.
    - *Nút Thoát (Form con):* Kiểm tra các ô nhập liệu, nếu đang có dữ liệu thì hiển thị MessageBox hỏi xác nhận trước khi đóng.
  - + **Nhận dữ liệu về Form chính:** Sau khi Form con đóng, kiểm tra nếu có đối tượng sinh viên mới trả về thì thêm dòng vào DataGridView và gọi hàm cập nhật số thứ tự (UpdateTT).
- Chức năng Tìm kiếm:
  - + Xử lý sự kiện TextChanged trên ô tìm kiếm của ToolStrip.
  - + Lấy từ khóa, chuyển về chữ thường (ToLower) và tạm dừng vẽ lưới (SuspendLayout).
  - + Duyệt qua từng dòng của DataGridView:
    - Lấy giá trị cột Tên (Cells[2]), so sánh chứa chuỗi ký tự tìm kiếm.

## IT008 – LẬP TRÌNH TRỰC QUAN

- Thiết lập thuộc tính Visible = true nếu tìm thấy, ngược lại gán false để ẩn dòng.
- + Gọi lại hàm UpdateTT để đánh lại số thứ tự liên tục cho các kết quả tìm được.
- Chức năng Hệ thống & Tiện ích:
  - + **Hàm UpdateTT:** Duyệt lại lưới, đánh số thứ tự (STT) tăng dần cho các dòng đang hiển thị (Visible = true).
  - + **Sắp xếp:** Khi người dùng sắp xếp cột (Sorted), tự động gọi lại UpdateTT để giữ đúng thứ tự hiển thị.
  - + **Menu Thoát:** Gọi lệnh Application.Exit() để tắt chương trình.

### Nội dung code

Bảng 5.1. Nội dung code bài 5

#### QLSV.cs

```
namespace Bai5
{
    public partial class QLSV : Form
    {
        public QLSV()
        {
            InitializeComponent();
        }

        private void UpdateTT()
        {
            int stt = 1;
            for (int i = 0; i < dataGridView.Rows.Count; i++)
            {
                if (dataGridView.Rows[i].Visible && !dataGridView.Rows[i].IsNewRow)
                {
                    dataGridView.Rows[i].Cells[0].Value = stt.ToString();
                    stt++;
                }
            }
        }

        private void toolStripButtonAdd_Click(object sender, EventArgs e)
        {
            addToolStripMenuItem_Click(sender, e);
        }

        private void addToolStripMenuItem_Click(object sender, EventArgs e)
        {
            AddNewForm addStudent = new AddNewForm();
            addStudent.ShowDialog();

            if (addStudent.newStudent != null)
            {
                dataGridView.Rows.Add(
                    "",
                    addStudent.newStudent.ID,
                    addStudent.newStudent.name,
                    addStudent.newStudent.apartment,
                    addStudent.newStudent.score.ToString()
                );
                UpdateTT();
            }
        }

        private void exitToolStripMenuItem_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}
```

```

    }

    private void toolStripTextBoxSearch_TextChanged(object sender, EventArgs e)
    {
        string keyword = toolStripTextBoxSearch.Text.ToLower().Trim();
        dataGridView.SuspendLayout();

        foreach (DataGridViewRow row in dataGridView.Rows)
        {
            if (!row.IsNewRow)
            {
                var cellValue = row.Cells[2].Value;

                if (cellValue != null && cellValue.ToString().ToLower().Contains(keyword))
                {
                    row.Visible = true;
                }
                else
                {
                    try
                    {
                        row.Visible = false;
                    }
                    catch (Exception)
                    {
                        dataGridView.CurrentCell = null;
                        row.Visible = false;
                    }
                }
            }
        }

        dataGridView.ResumeLayout();

        UpdateTT();
    }

    private void dataGridView_Sorted(object sender, EventArgs e)
    {
        UpdateTT();
    }
}

```

## QLSV.Designer.cs

```

namespace Bai5
{
    partial class QLSV
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
    }
}

```

```

private void InitializeComponent()
{
    System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(QLSV));
    DataGridViewCellStyle dataGridViewCellStyle1 = new DataGridViewCellStyle();
    menuStrip1 = new MenuStrip();
    chứcNăngToolStripMenuItem = new ToolStripMenuItem();
    addToolStripMenuItem = new ToolStripMenuItem();
    exitToolStripMenuItem = new ToolStripMenuItem();
    toolStrip1 = new ToolStrip();
    toolStripButtonAdd = new ToolStripButton();
    toolStripLabel1 = new ToolStripLabel();
    toolStripTextBoxSearch = new ToolStripTextBox();
    dataGridView = new DataGridView();
    STT = new DataGridViewTextBoxColumn();
    MSSV = new DataGridViewTextBoxColumn();
    Ten = new DataGridViewTextBoxColumn();
    Khoa = new DataGridViewTextBoxColumn();
    Diem = new DataGridViewTextBoxColumn();
    menuStrip1.SuspendLayout();
    toolStrip1.SuspendLayout();
    ((System.ComponentModel.ISupportInitialize)dataGridView).BeginInit();
    SuspendLayout();
    //
    // menuStrip1
    //
    menuStrip1.BackColor = Color.ForestGreen;
    menuStrip1.ImageScalingSize = new Size(20, 20);
    menuStrip1.Items.AddRange(new ToolStripItem[] { chứcNăngToolStripMenuItem });
    menuStrip1.Location = new Point(0, 0);
    menuStrip1.Name = "menuStrip1";
    menuStrip1.Size = new Size(800, 28);
    menuStrip1.TabIndex = 0;
    menuStrip1.Text = "menuStrip1";
    //
    // chứcNăngToolStripMenuItem
    //
    chứcNăngToolStripMenuItem.DropDownItems.AddRange(new ToolStripItem[] {
addToolStripMenuItem, exitToolStripMenuItem });
    chứcNăngToolStripMenuItem.Name = "chứcNăngToolStripMenuItem";
    chứcNăngToolStripMenuItem.Size = new Size(93, 24);
    chứcNăngToolStripMenuItem.Text = "Chức năng";
    //
    // addToolStripMenuItem
    //
    addToolStripMenuItem.Name = "addToolStripMenuItem";
    addToolStripMenuItem.ShortcutKeys = Keys.Control | Keys.N;
    addToolStripMenuItem.Size = new Size(212, 26);
    addToolStripMenuItem.Text = "Thêm mới";
    addToolStripMenuItem.Click += addToolStripMenuItem_Click;
    //
    // exitToolStripMenuItem
    //
    exitToolStripMenuItem.Name = "exitToolStripMenuItem";
    exitToolStripMenuItem.Size = new Size(212, 26);
    exitToolStripMenuItem.Text = "Thoát";
    exitToolStripMenuItem.Click += exitToolStripMenuItem_Click;
    //
    // toolStrip1
    //
    toolStrip1.AutoSize = false;
    toolStrip1.BackColor = Color.Honeydew;
    toolStrip1.ImageScalingSize = new Size(20, 20);
    toolStrip1.Items.AddRange(new ToolStripItem[] { toolStripButtonAdd,
toolStripLabel1, toolStripTextBoxSearch });
    toolStrip1.Location = new Point(0, 28);
    toolStrip1.Name = "toolStrip1";
    toolStrip1.Size = new Size(800, 50);
    toolStrip1.TabIndex = 1;
    toolStrip1.Text = "toolStrip1";
    //
    // toolStripButtonAdd
    //
    toolStripButtonAdd.AutoSize = false;
    toolStripButtonAdd.BackColor = Color.LightGreen;
    toolStripButtonAdd.Font = new Font("Segoe UI", 12F);
    toolStripButtonAdd.Image = (Image)resources.GetObject("toolStripButtonAdd.Image");
    toolStripButtonAdd.ImageAlign = ContentAlignment.MiddleLeft;

```

```

        toolStripButtonAdd.ImageTransparentColor = Color.Magenta;
        toolStripButtonAdd.Name = "toolStripButtonAdd";
        toolStripButtonAdd.Size = new Size(200, 50);
        toolStripButtonAdd.Text = "Thêm mới";
        toolStripButtonAdd.Click += toolStripButtonAdd_Click;
        //
        // toolStripLabel1
        //
        toolStripLabel1.Font = new Font("Segoe UI", 12F);
        toolStripLabel1.Name = "toolStripLabel1";
        toolStripLabel1.Size = new Size(169, 47);
        toolStripLabel1.Text = "Tìm kiếm theo tên";
        //
        // toolStripTextBoxSearch
        //
        toolStripTextBoxSearch.BorderStyle = BorderStyle.FixedSingle;
        toolStripTextBoxSearch.Font = new Font("Segoe UI", 12F);
        toolStripTextBoxSearch.Name = "toolStripTextBoxSearch";
        toolStripTextBoxSearch.Size = new Size(250, 50);
        toolStripTextBoxSearch.TextChanged += toolStripTextBoxSearch_TextChanged;
        //
        // dataGridView
        //
        dataGridView.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
        dataGridView.BackgroundColor = Color.DarkSeaGreen;
        dataGridView.BorderStyle = BorderStyle.None;
        dataGridViewCellStyle1.Alignment = DataGridViewContentAlignment.MiddleLeft;
        dataGridViewCellStyle1.BackColor = SystemColors.Control;
        dataGridViewCellStyle1.Font = new Font("Segoe UI", 9F, FontStyle.Bold,
GraphicsUnit.Point, 0);
        dataGridViewCellStyle1.ForeColor = SystemColors.WindowText;
        dataGridViewCellStyle1.SelectionBackColor = SystemColors.Highlight;
        dataGridViewCellStyle1.SelectionForeColor = SystemColors.HighlightText;
        dataGridViewCellStyle1.WrapMode = DataGridViewTriState.True;
        dataGridView.ColumnHeadersDefaultCellStyle = dataGridViewCellStyle1;
        dataGridView.ColumnHeadersHeightSizeMode =
DataGridViewColumnHeadersHeightSizeMode.AutoSize;
        dataGridView.Columns.AddRange(new DataGridViewColumn[] { STT, MSSV, Ten, Khoa,
Diem });

        dataGridView.Dock = DockStyle.Fill;
        dataGridView.Location = new Point(0, 78);
        dataGridView.Name = "dataGridView";
        dataGridView.RowHeadersWidth = 51;
        dataGridView.RowTemplate.DividerHeight = 1;
        dataGridView.RowTemplate.ReadOnly = true;
        dataGridView.Size = new Size(800, 372);
        dataGridView.TabIndex = 2;
        dataGridView.Sorted += dataGridView_Sorted;
        //
        // STT
        //
        STT.HeaderText = "Số TT";
        STT.MinimumWidth = 6;
        STT.Name = "STT";
        STT.ReadOnly = true;
        //
        // MSSV
        //
        MSSV.HeaderText = "Mã Số SV";
        MSSV.MinimumWidth = 6;
        MSSV.Name = "MSSV";
        //
        // Ten
        //
        Ten.HeaderText = "Tên Sinh Viên";
        Ten.MinimumWidth = 6;
        Ten.Name = "Ten";
        //
        // Khoa
        //
        Khoa.HeaderText = "Khoa";
        Khoa.MinimumWidth = 6;
        Khoa.Name = "Khoa";
        //
        // Diem
        //
        Diem.HeaderText = "Điểm TB";
        Diem.MinimumWidth = 6;

```

```

        Diem.Name = "Diem";
        //
        // QLSV
        //
        AutoScaleDimensions = new SizeF(8F, 20F);
        AutoScaleMode = AutoScaleMode.Font;
        ClientSize = new Size(800, 450);
        Controls.Add(dataGridView);
        Controls.Add(toolStrip1);
        Controls.Add(menuStrip1);
        MainMenuStrip = menuStrip1;
        Name = "QLSV";
        Text = "Quản lý sinh viên";
        menuStrip1.ResumeLayout(false);
        menuStrip1.PerformLayout();
        toolStrip1.ResumeLayout(false);
        toolStrip1.PerformLayout();
        ((System.ComponentModel.ISupportInitialize)dataGridView).EndInit();
        ResumeLayout(false);
        PerformLayout();
    }

#endregion

private MenuStrip menuStrip1;
private ToolStripMenuItem chứcNăngToolStripMenuItem;
private ToolStripMenuItem addToolStripMenuItem;
private ToolStripMenuItem exitToolStripMenuItem;
private ToolStrip toolStrip1;
private ToolStripButton toolStripButtonAdd;
private ToolStripLabel toolStripLabel1;
private ToolStripTextBox toolStripTextBoxSearch;
private DataGridView dataGridView;
private DataGridViewTextBoxColumn STT;
private DataGridViewTextBoxColumn MSSV;
private DataGridViewTextBoxColumn Ten;
private DataGridViewTextBoxColumn Khoa;
private DataGridViewTextBoxColumn Diem;
}
}

```

## AddNewForm.cs

```

namespace Bai5
{
    public partial class AddNewForm : Form
    {
        public AddNewForm()
        {
            InitializeComponent();
        }
        public Student newStudent;

        private bool ValidInfo()
        {
            bool isValid = true;

            if (textBoxMSSV.Text == "" || textBoxName.Text == "" || textBoxDiem.Text == "")
                isValid = false;

            foreach (char c in textBoxMSSV.Text)
            {
                if ((int)c < (int)'0' || (int)c > (int)'9')
                {
                    isValid = false;
                    textBoxMSSV.Text = "";
                }
            }

            foreach (char c in textBoxName.Text)
            {
                if ((int)c >= (int)'0' && (int)c <= (int)'9')
                {
                    isValid = false;
                    textBoxName.Text = "";
                }
            }
        }
    }
}

```

```

        double temp;
        if (!double.TryParse(textBoxDiem.Text, out temp))
        {
            textBoxDiem.Text = "";
            isValid = false;
        }
        else if (temp > 10 || temp < 0)
        {
            textBoxDiem.Text = "";
            isValid = false;
        }

        return isValid;
    }

    private void buttonAdd_Click(object sender, EventArgs e)
    {
        if (ValidInfo())
        {
            newStudent = new Student(textBoxMSSV.Text, textBoxName.Text,
            comboBoxKhoa.Text, double.Parse(textBoxDiem.Text));
            this.Close();
        }
        else MessageBox.Show("Vui lòng điền lại các thông tin bị sai hoặc chưa được
        điền");
    }

    private void buttonExit_Click(object sender, EventArgs e)
    {
        if (textBoxMSSV.Text == "" && textBoxName.Text == "" && textBoxDiem.Text == "")
            this.Close();
        else
        {
            DialogResult dialogResult = MessageBox.Show("Bạn có chắc chắn muốn thoát?",
            "XÁC NHẬN", MessageBoxButtons.YesNo);
            if (dialogResult == DialogResult.Yes)
            {
                this.Close();
            }
        }
    }
}

```

## AddNewForm.Designer.cs

```

namespace Bai5
{
    partial class AddNewForm
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
    }
}

```



```

private void InitializeComponent()
{
    label1 = new Label();
    labelTitle = new Label();
    label2 = new Label();
    label3 = new Label();
    label4 = new Label();
    textBoxMSSV = new TextBox();
    textBoxName = new TextBox();
    textBoxDiem = new TextBox();
    comboBoxKhoa = new ComboBox();
    buttonExit = new Button();
    buttonAdd = new Button();
    SuspendLayout();
    //
    // label1
    //
    label1.AutoSize = true;
    label1.Font = new Font("Segoe UI", 12F, FontStyle.Regular, GraphicsUnit.Point, 0);
    label1.Location = new Point(137, 94);
    label1.Margin = new Padding(10);
    label1.Name = "label1";
    label1.Size = new Size(154, 28);
    label1.TabIndex = 0;
    label1.Text = "Mã Số Sinh Viên";
    //
    // labelTitle
    //
    labelTitle.BackColor = Color.PaleTurquoise;
    labelTitle.Dock = DockStyle.Top;
    labelTitle.Font = new Font("Segoe UI Black", 22.2F, FontStyle.Bold,
GraphicsUnit.Point, 0);
    labelTitle.Location = new Point(0, 0);
    labelTitle.Name = "labelTitle";
    labelTitle.Size = new Size(800, 60);
    labelTitle.TabIndex = 1;
    labelTitle.Text = "Thêm sinh viên";
    labelTitle.TextAlign = ContentAlignment.MiddleCenter;
    //
    // label2
    //
    label2.AutoSize = true;
    label2.Font = new Font("Segoe UI", 12F, FontStyle.Regular, GraphicsUnit.Point, 0);
    label2.Location = new Point(137, 142);
    label2.Margin = new Padding(10);
    label2.Name = "label2";
    label2.Size = new Size(125, 28);
    label2.TabIndex = 2;
    label2.Text = "Tên Sinh viên";
    //
    // label3
    //
    label3.AutoSize = true;
    label3.Font = new Font("Segoe UI", 12F, FontStyle.Regular, GraphicsUnit.Point, 0);
    label3.Location = new Point(137, 190);
    label3.Margin = new Padding(10);
    label3.Name = "label3";
    label3.Size = new Size(57, 28);
    label3.TabIndex = 3;
    label3.Text = "Khoa";
    //
    // label4
    //
    label4.AutoSize = true;
    label4.Font = new Font("Segoe UI", 12F, FontStyle.Regular, GraphicsUnit.Point, 0);
    label4.Location = new Point(137, 238);
    label4.Margin = new Padding(10);
    label4.Name = "label4";
    label4.Size = new Size(84, 28);
    label4.TabIndex = 4;
    label4.Text = "Điểm TB";
    //
    // textBoxMSSV
    //
    textBoxMSSV.BorderStyle = BorderStyle.FixedSingle;
    textBoxMSSV.Font = new Font("Segoe UI", 12F);
    textBoxMSSV.Location = new Point(304, 91);
    textBoxMSSV.Name = "textBoxMSSV";

```

```

textBoxMSSV.Size = new Size(326, 34);
textBoxMSSV.TabIndex = 1;
//
// textBoxName
//
textBoxName.BorderStyle = BorderStyle.FixedSingle;
textBoxName.Font = new Font("Segoe UI", 12F);
textBoxName.Location = new Point(304, 139);
textBoxName.Name = "textBoxName";
textBoxName.Size = new Size(326, 34);
textBoxName.TabIndex = 2;
//
// textBoxDiem
//
textBoxDiem.BorderStyle = BorderStyle.FixedSingle;
textBoxDiem.Font = new Font("Segoe UI", 12F);
textBoxDiem.Location = new Point(304, 235);
textBoxDiem.Name = "textBoxDiem";
textBoxDiem.Size = new Size(326, 34);
textBoxDiem.TabIndex = 4;
//
// comboBoxKhoa
//
comboBoxKhoa.DropDownStyle = ComboBoxStyle.DropDownList;
comboBoxKhoa.Font = new Font("Segoe UI", 12F);
comboBoxKhoa.FormattingEnabled = true;
comboBoxKhoa.Items.AddRange(new object[] { "Công nghệ phần mềm", "Khoa học máy
tính", "Kỹ thuật máy tính", "Công nghệ thông tin", "Mạng máy tính và truyền thông", "Hệ thống
thông tin" });
comboBoxKhoa.Location = new Point(304, 187);
comboBoxKhoa.Name = "comboBoxKhoa";
comboBoxKhoa.Size = new Size(326, 36);
comboBoxKhoa.TabIndex = 3;
//
// buttonExit
//
buttonExit.BackColor = Color.Salmon;
buttonExit.FlatStyle = FlatStyle.Flat;
buttonExit.Font = new Font("Segoe UI", 12F);
buttonExit.Location = new Point(480, 306);
buttonExit.Name = "buttonExit";
buttonExit.Size = new Size(150, 50);
buttonExit.TabIndex = 6;
buttonExit.Text = "Thoát";
buttonExit.UseVisualStyleBackColor = false;
buttonExit.Click += buttonExit_Click;
//
// buttonAdd
//
buttonAdd.BackColor = Color.PaleGreen;
buttonAdd.FlatStyle = FlatStyle.Flat;
buttonAdd.Font = new Font("Segoe UI", 12F);
buttonAdd.Location = new Point(304, 306);
buttonAdd.Name = "buttonAdd";
buttonAdd.Size = new Size(170, 50);
buttonAdd.TabIndex = 5;
buttonAdd.Text = "Thêm mới";
buttonAdd.UseVisualStyleBackColor = false;
buttonAdd.Click += buttonAdd_Click;
//
// AddNewForm
//
AutoScaleDimensions = new.SizeF(8F, 20F);
AutoScaleMode = AutoScaleMode.Font;
ClientSize = new Size(800, 450);
Controls.Add(buttonAdd);
Controls.Add(buttonExit);
Controls.Add(comboBoxKhoa);
Controls.Add(textBoxDiem);
Controls.Add(textBoxName);
Controls.Add(textBoxMSSV);
Controls.Add(label4);
Controls.Add(label3);
Controls.Add(label2);
Controls.Add(labelTitle);
Controls.Add(label1);
MaximizeBox = false;
Name = "AddNewForm";

```

```

        Text = "Thêm Sinh Viên ";
        ResumeLayout(false);
        PerformLayout();
    }

    #endregion

    private Label label1;
    private Label labelTitle;
    private Label label2;
    private Label label3;
    private Label label4;
    private TextBox textBoxMSSV;
    private TextBox textBoxName;
    private TextBox textBoxDiem;
    private ComboBox comboBoxKhoa;
    private Button buttonExit;
    private Button buttonAdd;
}

```

### Kiểm thử chương trình

- Giao diện chương trình:

Hình 5.1. Giao diện chính bài 5

Số TT	Mã Số SV	Tên Sinh Viên	Khoa	Điểm TB
1	24520604	Nguyễn Gia Hưng	Công nghệ phần m...	10



**Thêm sinh viên**

Mã Số Sinh Viên

Tên Sinh viên

Khoa

Điểm TB

**Thêm mới** **Thoát**

– Kịch bản kiểm thử:

Bảng 5.2. Kiểm thử bài 5

STT	Tiêu chí	Hành động	Kết quả
1	Chức năng	<p>Khởi động chương trình</p> <p>Click vào Thêm mới, hoặc Chức năng &gt; Thêm mới, hoặc nhấn Ctrl + N</p> <p>Click Menu Hệ thống &gt; Thoát.</p>	<p>Form QLSV hiện lên, DataGridView trống, Menu và Toolbar hiển thị đầy đủ.</p> <p>Khi mở chức năng thêm, form nhập liệu (AddNewForm) được mở ra dưới dạng Dialog.</p> <p>Khi chọn Thoát và đóng form, chương trình đóng hoàn toàn</p>

## IT008 – LẬP TRÌNH TRỰC QUAN

2	Thêm mới thông tin không hợp lệ	<p>Mở chức năng thêm, nhập thông tin sinh viên cần thêm.</p> <p>Để trống vài thông tin và nhấn Thêm.</p> <p>Nhập MSSV có ký tự không phải số, nhập tên chứa số, nhập điểm là chữ, nhập điểm là số không hợp lệ (<math>&lt; 0</math> hoặc <math>&gt; 10</math>), sau đó nhấn thêm</p>	<p>Hiện thị thông báo lỗi, ô lỗi bị xóa trắng và yêu cầu nhập lại các ô trống</p>
3	Thoát khi đang nhập	<p>Nhập vài thông tin và bấm Thoát, chọn Yes hoặc No.</p> <p>Để form trống và bấm thoát</p>	<p>Nếu chọn Yes, đóng form nhập liệu và trở về form gốc, nếu chọn No thì hủy Thoát, trở về nhập liệu.</p> <p>Nếu form trống, thoát luôn không hỏi Yes No</p>
4	Thêm thông tin hợp lệ	Nhập các thông tin hợp lệ	<p>Form nhập liệu đóng.</p> <p>Dữ liệu mới xuất hiện ở dòng cuối DataGridView. STT tự động điền đúng thứ tự.</p>
5	Tìm tên	Nhập đúng tên sinh viên trong danh sách, kể cả đúng Hoa thường và không đúng.	<p>DataGridView chỉ hiển thị các dòng có tên chứa từ khóa. Cột STT được đánh lại từ 1, 2...</p>
6	Tìm không thấy	Nhập tên không tồn tại	DataGridView ẩn hết các dòng dữ liệu

## IT008 – LẬP TRÌNH TRỰC QUAN

7	Xóa từ khóa	Xóa hết chữ trong ô tìm kiếm	DataGridView hiển thị lại toàn bộ danh sách sinh viên ban đầu. STT cập nhật lại đầy đủ.
8	Sắp xếp cột	Click vào tiêu đề cột để sắp xếp.	Các hàng thay đổi vị trí theo thứ tự tăng/giảm dần. Cột STT <b>không</b> bị đảo lộn.

– Video minh họa: <https://youtu.be/5g2AUJsRPHk>

**Mô tả chương trình**

- Yêu cầu:
  - Chương trình có thể chọn đường dẫn thư mục nguồn và đích
  - Khi nhấn Sao chép sẽ copy toàn bộ nội dung trong thư mục nguồn vào thư mục đích
  - Tiến trình sao chép thể hiện trên process bar, các file đang sao chép thể hiện trên thanh statusStrip và tooltip khi hover vào thanh process bar
  - Chức năng các button được thể hiện bởi tooltip khi hover chuột vào
- Hướng xử lý:
  - Thiết kế giao diện và cấu hình theo yêu cầu
  - Tạo và cấu hình một **BackgroundWorker** để thực hiện tác vụ sao chép file trên luồng phụ, giúp giao diện không bị treo
    - + Khi người dùng nhấn nút Sao chép, chương trình kiểm tra đường dẫn nguồn/đích hợp lệ, khóa nút bấm để tránh xung đột, và gọi lệnh RunWorkerAsync để bắt đầu luồng xử lý nền.
    - + Worker thực hiện tính toán tổng dung lượng thư mục nguồn trước. Sau đó, nó chạy thuật toán **đệ quy** để duyệt vào từng thư mục con. Quá trình copy file sử dụng FileStream để đọc/ghi từng khối dữ liệu (buffer). Sau mỗi khối dữ liệu được ghi, Worker tính toán phần trăm hoàn thành và gọi ReportProgress để gửi kèm **tên file hiện tại** về luồng chính.
  - Tại luồng giao diện chính, chương trình nhận thông tin từ Worker để thực hiện: Tăng giá trị của **ProgressBar**, Hiển thị tên file đang chép lên thanh **StatusStrip**, Thực hiện gán **Tooltip động** cho ProgressBar, giúp hiển thị tên file ngay tại vị trí chuột khi người dùng hover vào thanh tiến trình.
  - Khi quá trình kết thúc hoặc bị hủy, chương trình hiển thị thông báo (MessageBox), đặt lại thanh tiến trình và mở lại (Enable) các nút chức năng để người dùng tiếp tục thao tác.

**Nội dung code***Bảng 6.1. Nội dung code bài 6*

CopyFile.cs
<pre>using System.ComponentModel;  namespace Bai6 {     public partial class CopyFile : Form     {         string sourcePath = "";         string destinationPath = "";     } }</pre>

```

// Biến để tính toán Progress Bar
long totalSize = 0;
long totalCopied = 0;

BackgroundWorker worker = new BackgroundWorker();

public CopyFile()
{
    InitializeComponent();

    // Cấu hình giao diện ban đầu
    textBoxSource.ReadOnly = true;
    textBoxDes.ReadOnly = true;
    toolStripStatusLabel.Text = "";

    // Cấu hình Worker
    worker.WorkerReportsProgress = true;
    worker.WorkerSupportsCancellation = true;

    worker.DoWork += Worker_DoWork;
    worker.ProgressChanged += Worker_ProgressChanged;
    worker.RunWorkerCompleted += Worker_RunWorkerCompleted;

    // Cấu hình Tooltip
    tooltip1.SetToolTip(buttonSource, "Chọn thư mục nguồn");
    tooltip1.SetToolTip(buttonDes, "Chọn thư mục đích");
    tooltip1.SetToolTip(textBoxSource, "Đường dẫn thư mục nguồn");
    tooltip1.SetToolTip(textBoxDes, "Đường dẫn thư mục đích");
    tooltip1.SetToolTip(buttonCopy, "Bắt đầu sao chép thư mục");
    tooltip1.SetToolTip(progressBar1, "Tiến trình sao chép");
}

// 1. Chọn Folder Nguồn
private void buttonSource_Click(object sender, EventArgs e)
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    fbd.Description = "Thư mục nguồn";

    if (fbd.ShowDialog() == DialogResult.OK)
    {
        sourcePath = fbd.SelectedPath;
        textBoxSource.Text = sourcePath;
    }
}

// 2. Chọn Folder Đích
private void buttonDes_Click(object sender, EventArgs e)
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    fbd.Description = "Thư mục đích";

    if (fbd.ShowDialog() == DialogResult.OK)
    {
        destinationPath = fbd.SelectedPath;
        textBoxDes.Text = destinationPath;
    }
}

// 3. Nút Copy (Đã cập nhật logic đổi tên trùng)
private void buttonCopy_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(sourcePath) || string.IsNullOrEmpty(destinationPath))
    {
        MessageBox.Show("Vui lòng chọn đủ đường dẫn!");
        return;
    }
    if (!Directory.Exists(sourcePath))
    {
        MessageBox.Show($"Thư mục nguồn không tồn tại hoặc đã bị đổi tên!\nĐường dẫn cũ: {sourcePath}\nVui lòng chọn lại.", "Lỗi đường dẫn", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    if (!Directory.Exists(destinationPath))
    {

```



```

        MessageBox.Show($"Thư mục đích không tồn tại hoặc đã bị xóa!\nĐường dẫn cũ: {destinationPath}\nVui lòng chọn lại.", "Lỗi đường dẫn", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    // Lấy tên thư mục gốc (Ví dụ: "Data")
    string folderName = new DirectoryInfo(sourcePath).Name;
    string finalDest = Path.Combine(destinationPath, folderName);

    int count = 1;
    while (Directory.Exists(finalDest))
    {
        // Nếu trùng, tạo tên mới: "Data (1)", "Data (2)"...
        string newName = $"{folderName} ({count})";
        finalDest = Path.Combine(destinationPath, newName);
        count++;
    }
    // Kết thúc vòng lặp này, finalDest sẽ là một đường dẫn duy nhất chưa tồn tại

    // Kiểm tra tránh copy lồng nhau (Source nằm trong Dest)
    // Lưu ý: finalDest lúc này có thể là "Data (1)", vẫn phải check
    if (finalDest.StartsWith(sourcePath))
    {
        MessageBox.Show("Thư mục đích không được nằm bên trong thư mục nguồn!");
        return;
    }

    buttonCopy.Enabled = false;
    progressBar1.Value = 0;

    // Truyền đường dẫn đích (đã được xử lý trùng tên) vào Worker
    worker.RunWorkerAsync(finalDest);
}

private void Worker_DoWork(object sender, DoWorkEventArgs e)
{
    string finalDestDir = (string)e.Argument;

    if (!Directory.Exists(sourcePath))
    {
        throw new DirectoryNotFoundException("Thư mục nguồn đã bị mất hoặc đổi tên trong quá trình xử lý.");
    }

    //Tính tổng dung lượng thư mục nguồn
    totalSize = CalculateDirSize(sourcePath);
    totalCopied = 0;

    //Bắt đầu Copy đệ quy
    CopyDirectoryRecursive(sourcePath, finalDestDir);
}

// Hàm Đệ quy: Copy thư mục và toàn bộ nội dung con
private void CopyDirectoryRecursive(string sourceDir, string destDir)
{
    if (!Directory.Exists(sourcePath))
    {
        throw new DirectoryNotFoundException("Thư mục nguồn đã bị mất hoặc đổi tên trong quá trình xử lý.");
    }
    // 1. Tạo thư mục đích
    Directory.CreateDirectory(destDir);

    // 2. Lấy danh sách File và copy
    string[] files = Directory.GetFiles(sourceDir);
    foreach (string file in files)
    {
        string name = Path.GetFileName(file);
        string destFile = Path.Combine(destDir, name);
        CopySingleFile(file, destFile);
    }

    // 3. ĐỆ QUY: Lấy danh sách Thư mục con
    string[] dirs = Directory.GetDirectories(sourceDir);
    foreach (string dir in dirs)
    {
        string name = new DirectoryInfo(dir).Name;

```

```

        string destSubDir = Path.Combine(destDir, name);
        CopyDirectoryRecursive(dir, destSubDir);
    }
}

// Hàm copy 1 file
private void CopySingleFile(string source, string dest)
{
    byte[] buffer = new byte[1024 * 1024]; // Buffer 1MB
    int bytesRead;

    if (totalSize > 0)
    {
        int percent = (int)((double)totalCopied / totalSize * 100);
        worker.ReportProgress(percent, source);
    }

    using (FileStream fsSource = new FileStream(source, FileMode.Open,
        FileAccess.Read))
    using (FileStream fsDest = new FileStream(dest, FileMode.Create,
        FileAccess.Write))
    {
        while ((bytesRead = fsSource.Read(buffer, 0, buffer.Length)) > 0)
        {
            fsDest.Write(buffer, 0, bytesRead);
            totalCopied += bytesRead;

            if (totalSize > 0)
            {
                int percent = (int)((double)totalCopied / totalSize * 100);
                worker.ReportProgress(percent, source);
            }
        }
    }
}

// Hàm tính tổng dung lượng
private long CalculateDirSize(string path)
{
    long size = 0;
    try
    {
        string[] files = Directory.GetFiles(path);
        foreach (string file in files)
        {
            FileInfo info = new FileInfo(file);
            size += info.Length;
        }

        string[] dirs = Directory.GetDirectories(path);
        foreach (string dir in dirs)
        {
            size += CalculateDirSize(dir);
        }
    }
    catch { } // Bỏ qua lỗi truy cập nếu có file hệ thống
    return size;
}

private void Worker_ProgressChanged(object sender, ProgressChangedEventArgs e)
{
    if (e.ProgressPercentage > progressBar1.Value)
    {
        progressBar1.Value = Math.Min(e.ProgressPercentage, 100);

        string currentFile = e.UserState as string;

        if (currentFile != null)
        {
            string fileName = Path.GetFileName(currentFile);
            string statusMessage = $"Đang sao chép: {fileName}";

            toolStripStatusLabel.Text = statusMessage;
            toolTip1.SetToolTip(progressBar1, statusMessage);
        }
    }
}

```

```

private void Worker_RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)
{
    buttonCopy.Enabled = true;
    progressBar1.Value = 100;
    toolStripStatusLabel.Text = "Đã hoàn tất sao chép.";
    toolTip1.SetToolTip(progressBar1, "Sao chép hoàn tất.");

    if (e.Error != null) MessageBox.Show("Lỗi: " + e.Error.Message);
    else
    {
        MessageBox.Show("Sao chép thư mục hoàn tất!");
        progressBar1.Value = 0;
        toolTip1.SetToolTip(progressBar1, "Tiến trình sao chép");
    }
}
}
}

```

### CopyFile.Designer.cs

```

namespace Bai6
{
    partial class CopyFile
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
        false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            components = new System.ComponentModel.Container();
            groupBox1 = new GroupBox();
            buttonCopy = new Button();
            buttonDes = new Button();
            buttonSource = new Button();
            textBoxDes = new TextBox();
            textBoxSource = new TextBox();
            label2 = new Label();
            label1 = new Label();
            groupBox2 = new GroupBox();
            progressBar1 = new ProgressBar();
            statusStrip1 = new StatusStrip();
            toolStripStatusLabel = new ToolStripStatusLabel();
            toolTip1 = new ToolTip(components);
            groupBox1.SuspendLayout();
            groupBox2.SuspendLayout();
            statusStrip1.SuspendLayout();
            SuspendLayout();
            //
            // groupBox1
            //
            groupBox1.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Right;
            groupBox1.BackColor = SystemColors.ButtonFace;
            groupBox1.Controls.Add(buttonCopy);

```

```

groupBox1.Controls.Add(buttonDes);
groupBox1.Controls.Add(buttonSource);
groupBox1.Controls.Add(textBoxDes);
groupBox1.Controls.Add(textBoxSource);
groupBox1.Controls.Add(label2);
groupBox1.Controls.Add(label1);
groupBox1.Font = new Font("Segoe UI", 12F);
groupBox1.Location = new Point(39, 39);
groupBox1.Margin = new Padding(30);
groupBox1.Name = "groupBox1";
groupBox1.Size = new Size(698, 209);
groupBox1.TabIndex = 0;
groupBox1.TabStop = false;
groupBox1.Text = "Sao chép tập tin";
//
// buttonCopy
//
buttonCopy.Anchor = AnchorStyles.Top | AnchorStyles.Right;
buttonCopy.BackColor = SystemColors.ControlDark;
buttonCopy.FlatStyle = FlatStyle.Flat;
buttonCopy.Location = new Point(313, 139);
buttonCopy.Margin = new Padding(10, 10, 30, 10);
buttonCopy.Name = "buttonCopy";
buttonCopy.Size = new Size(102, 49);
buttonCopy.TabIndex = 6;
buttonCopy.Text = "Sao chép";
buttonCopy.UseVisualStyleBackColor = false;
buttonCopy.Click += buttonCopy_Click;
//
// buttonDes
//
buttonDes.Anchor = AnchorStyles.Top | AnchorStyles.Right;
buttonDes.BackColor = SystemColors.ControlDark;
buttonDes.FlatStyle = FlatStyle.Flat;
buttonDes.Location = new Point(605, 85);
buttonDes.Margin = new Padding(10, 10, 30, 10);
buttonDes.Name = "buttonDes";
buttonDes.Size = new Size(60, 34);
buttonDes.TabIndex = 5;
buttonDes.Text = "...";
buttonDes.UseVisualStyleBackColor = false;
buttonDes.Click += buttonDes_Click;
//
// buttonSource
//
buttonSource.Anchor = AnchorStyles.Top | AnchorStyles.Right;
buttonSource.BackColor = SystemColors.ControlDark;
buttonSource.FlatStyle = FlatStyle.Flat;
buttonSource.Location = new Point(605, 37);
buttonSource.Margin = new Padding(10, 10, 30, 10);
buttonSource.Name = "buttonSource";
buttonSource.Size = new Size(60, 34);
buttonSource.TabIndex = 4;
buttonSource.Text = "...";
buttonSource.UseVisualStyleBackColor = false;
buttonSource.Click += buttonSource_Click;
//
// textBoxDes
//
textBoxDes.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Right;
textBoxDes.BorderStyle = BorderStyle.FixedSingle;
textBoxDes.Location = new Point(313, 85);
textBoxDes.Margin = new Padding(10);
textBoxDes.Name = "textBoxDes";
textBoxDes.Size = new Size(272, 34);
textBoxDes.TabIndex = 3;
//
// textBoxSource
//
textBoxSource.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Right;
textBoxSource.BorderStyle = BorderStyle.FixedSingle;
textBoxSource.Location = new Point(313, 37);
textBoxSource.Margin = new Padding(10);
textBoxSource.Name = "textBoxSource";
textBoxSource.Size = new Size(272, 34);
textBoxSource.TabIndex = 2;
//
// label2

```

```

//
label2.AutoSize = true;
label2.Location = new Point(33, 88);
label2.Margin = new Padding(30, 10, 10, 10);
label2.Name = "label2";
label2.Size = new Size(238, 28);
label2.TabIndex = 1;
label2.Text = "Đường Dẫn Thư Mục Đích";
//
// label1
//
label1.AutoSize = true;
label1.Location = new Point(33, 40);
label1.Margin = new Padding(30, 10, 10, 10);
label1.Name = "label1";
label1.Size = new Size(260, 28);
label1.TabIndex = 0;
label1.Text = "Đường Dẫn Thư Mục Nguồn";
//
// groupBox2
//
groupBox2.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Right;
groupBox2.BackColor = SystemColors.Control;
groupBox2.Controls.Add(progressBar1);
groupBox2.Font = new Font("Segoe UI", 12F);
groupBox2.Location = new Point(39, 267);
groupBox2.Margin = new Padding(30);
groupBox2.Name = "groupBox2";
groupBox2.Size = new Size(698, 119);
groupBox2.TabIndex = 7;
groupBox2.TabStop = false;
groupBox2.Text = "Tiến trình sao chép";
//
// progressBar1
//
progressBar1.Anchor = AnchorStyles.Top | AnchorStyles.Left | AnchorStyles.Right;
progressBar1.Location = new Point(33, 46);
progressBar1.Name = "progressBar1";
progressBar1.Size = new Size(632, 46);
progressBar1.TabIndex = 6;
//
// statusStrip1
//
statusStrip1.ImageScalingSize = new Size(20, 20);
statusStrip1.Items.AddRange(new ToolStripItem[] { toolStripStatusLabel });
statusStrip1.Location = new Point(0, 491);
statusStrip1.Name = "statusStrip1";
statusStrip1.Size = new Size(776, 22);
statusStrip1.TabIndex = 8;
statusStrip1.Text = "statusStrip1";
//
// toolStripStatusLabel
//
toolStripStatusLabel.Name = "toolStripStatusLabel";
toolStripStatusLabel.Size = new Size(0, 16);
//
// CopyFile
//
AutoScaleDimensions = new.SizeF(8F, 20F);
AutoScaleMode = AutoScaleMode.Font;
ClientSize = new Size(776, 513);
Controls.Add(statusStrip1);
Controls.Add(groupBox2);
Controls.Add(groupBox1);
Name = "CopyFile";
Text = "Sao chép tập tin";
groupBox1.ResumeLayout(false);
groupBox1.PerformLayout();
groupBox2.ResumeLayout(false);
statusStrip1.ResumeLayout(false);
statusStrip1.PerformLayout();
ResumeLayout(false);
PerformLayout();
}

#endregion

private GroupBox groupBox1;

```

```

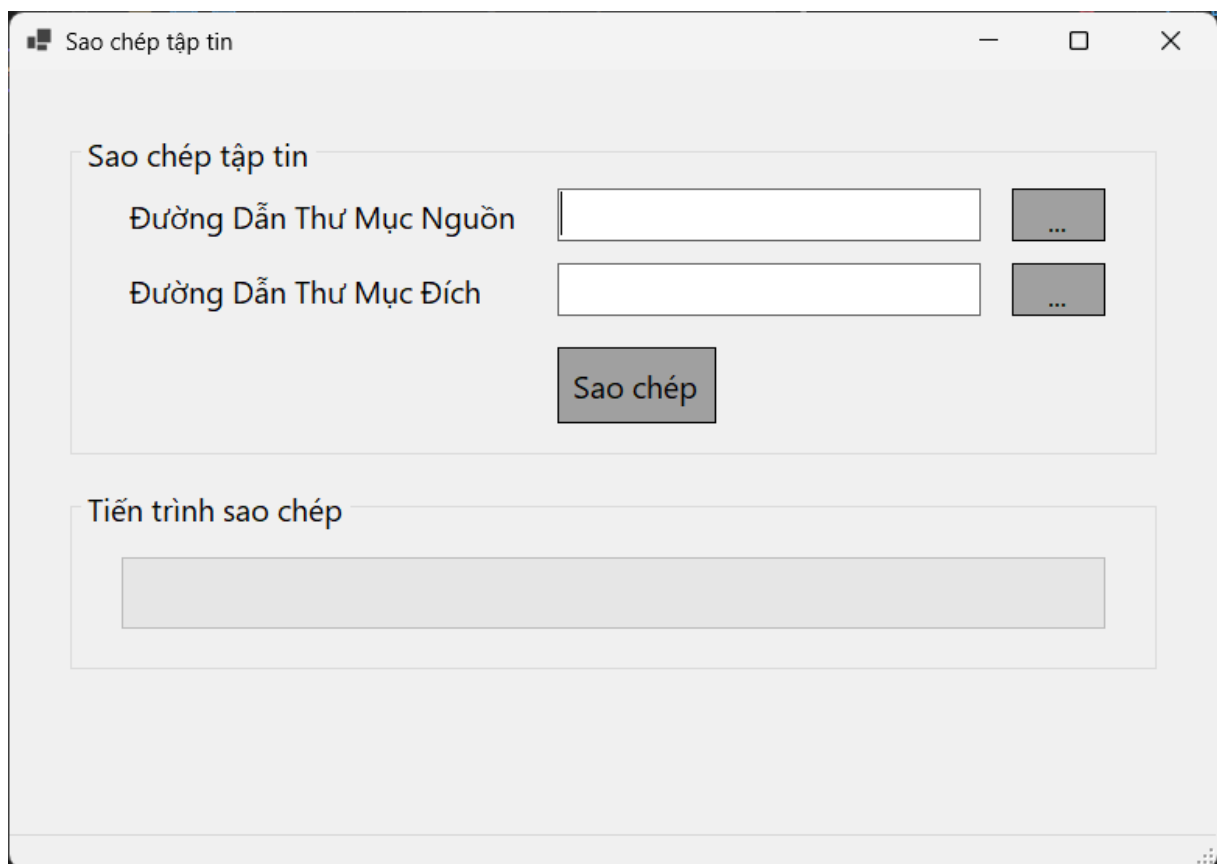
private Label label2;
private Label label1;
private TextBox textBoxSource;
private Button buttonDes;
private Button buttonSource;
private TextBox textBoxDes;
private Button buttonCopy;
private GroupBox groupBox2;
private ProgressBar progressBar1;
private StatusStrip statusStrip1;
private ToolStripStatusLabel toolStripStatusLabel;
private ToolTip toolTip1;
    }
}

```

### Kiểm thử chương trình

- Giao diện chương trình:

Hình 6.1. Giao diện chính bài 6



- Kịch bản kiểm thử:

Bảng 6.2. Kiểm thử bài 6

STT	Tiêu chí	Hành động	Kết quả
1	Giao diện và tooltip tĩnh	Tạo chương trình, di chuột vào các textbox, button trên giao diện	Hiển thị tooltip hướng dẫn chức năng của button, textbox

## IT008 – LẬP TRÌNH TRỰC QUAN

2	Đường dẫn hợp lệ	Dùng chức năng open folder để chọn đường dẫn hợp lệ, folder đích không nằm trong nguồn. Nhấn Sao chép.  Kiểm tra trạng thái các thư mục khi hoàn tất	Sao chép thành công. Thư mục đích có 1 folder thư mục nguồn.
3	Đường dẫn không hợp lệ	Để đường dẫn trống, nhấn Sao chép.  Chọn folder đích nằm trong folder nguồn	Thông báo nhập đầy đủ đường dẫn.  Cảnh báo không sao chép để tránh lặp vô tận.
4	Status Strip và Tool Tip khi copy dữ liệu lớn	Chọn một file nguồn có dữ liệu lớn, thực hiện sao chép hợp lệ. Di chuột và thanh tiến trình, quan sát thanh trạng thái và tooltip	Thanh trạng thái và tool tip thể hiện file đang copy
5	Ngắt tiến trình	Thực hiện sao chép, khi chưa hoàn tất thì tắt chương trình	Thư mục đích có một vài file từ thư mục đích (chưa sao chép hoàn chỉnh)
6	Đường dẫn không tồn tại	Chọn địa chỉ nguồn và đích. Sau đó đổi tên thư mục nhưng không chọn lại, lúc này các biến địa chỉ trong chương trình đang lưu một địa chỉ không tồn tại. Thực hiện sao chép.	Hiển thị thông báo lỗi đường dẫn không tồn tại hoặc đổi tên. Yêu cầu chọn lại

## IT008 – LẬP TRÌNH TRỰC QUAN

7	Thay đổi thư mục trong quá trình sao chép	Thực hiện sao chép hợp lệ thư mục lớn để có thời gian, thực hiện đổi tên thư mục nguồn và đích	Không thực hiện được do window chặn, do đang được sử dụng ở một tiến trình khác
---	---	--	---

– Video minh họa: <https://youtu.be/dlF7zQTZGtg>

Link code:

[https://github.com/nguyenhun11/IT008.Q14\\_BTH4\\_NguyenGiaHung\\_24520604](https://github.com/nguyenhun11/IT008.Q14_BTH4_NguyenGiaHung_24520604)

Playlist các video minh họa:

<https://www.youtube.com/playlist?list=PLdcN3QW0mt2ztlI-Q2WIMQAlpz6zYrxj8>