**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỆ ĐIỀU HÀNH**

**Đề tài:**Xây dựng chương trình nghe nhạc mp3

**Lớp 58TH3**

**Thành viên:**

Họ và tên: Lâm Tuấn Khải

Mã sinh viên :1651061138

Họ và tên: Nguyễn Văn Khánh

Mã sinh viên :1651061105

Họ và tên: Nguyễn Ngọc Phương Huy

Mã sinh viên : 1651061037

Họ và tên: Đỗ Minh Huy

Mã sinh viên : 1651060851

Họ và tên: Phạm Văn Nhật

Mã sinh viên : 1651061196

Hà Nội, tháng 4 năm 2019

MỤC LỤC

Trang

Lời nói đầu …………………………………………………………………

I. Code chương trình………………………………………………………..

II. Hình ảnh mô phỏng quá trình chương trình đang chạy…………………

LỜI NÓI ĐẦU

Báo cáo thực hành sử dụng hàm API của windows để xây dựng chương sử dụng ngôn ngữ C# trong đó API là Application Programming Interface (giao diện lập trình ứng dụng) để gọi hàm bên trong thư viện liên kết động DLL. Thực hiên các tác vụ của windows tạo các control, truy xuất các tác vụ hệ thống bàn phím màn hình và các chức năng khác. Nó chính là một phần mềm giao tiếp được với các ứng dụng khác nhau giữa chương trình và hệ điều hành. Trong bài này sử dụng 2 hàm API của windows nằm trong thư viện user32.dll và winmm.dll.

I. Code chương trình

1. Khai báo hàm API sử dụng ngôn ngữ C#

using System.Runtime.InteropServices;

[DllImport("user32.dll", EntryPoint = "MessageBox")]

public static extern int MessageBox(int a, string b, string c, uint type);

[DllImport("winmm.dll")]

private static extern long mciSendString(string strCommand, StringBuilder

strReturn, int iReturnLength, IntPtr hwndCallback);

2. Phần xử lý hàm API trong thư viện user32.dll và winmm.dll

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

namespace Mp3Player

{

class Mp3Player

{

[DllImport("user32.dll", EntryPoint = "MessageBox")]

public static extern int MessageBox(int a, string b, string c, uint type);

[DllImport("winmm.dll")]

private static extern long mciSendString(string strCommand, StringBuilder

strReturn, int iReturnLength, IntPtr hwndCallback);

public bool Repeat { get; set; }

public Mp3Player(string szFileName)

{

string szCommad = "open \"" + szFileName + "\" type mpegvideo alias MediaFile";

MessageBox(0, "Ban da chon thanh cong", "Thông báo", 0);

mciSendString(szCommad, null, 0, IntPtr.Zero);

}

public void Mp3Play()

{

string szCommand = "play MediaFile";

if (Repeat) szCommand += " REPEAT";

mciSendString(szCommand, null, 0, IntPtr.Zero);

}

public void Mp3Pause()

{

string szCommand = "pause MediaFile";

mciSendString(szCommand, null, 0, IntPtr.Zero);

}

public void Mp3Close()

{

string szCommand = "close MediaFile";

mciSendString(szCommand, null, 0, IntPtr.Zero);

}

}

}

2. Code xử lý sự kiện form

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Mp3Player

{

public partial class Form1 : Form

{

private Mp3Player \_mp3Player;

public Form1()

{

InitializeComponent();

}

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

{

using (OpenFileDialog op = new OpenFileDialog())

{

op.Filter = "MP3 File|\*.mp3";

op.InitialDirectory =

Environment.GetFolderPath(Environment.SpecialFolder.MyMusic);

if (op.ShowDialog() == DialogResult.OK)

{

\_mp3Player = new Mp3Player(op.FileName);

\_mp3Player.Repeat = true;

txt\_fileName.Text = op.FileName;

WindowsMediaPlayer.URL = op.FileName;

}

}

}

private void btn\_Play\_Click(object sender, EventArgs e)

{

if(\_mp3Player != null)

{

WindowsMediaPlayer.Ctlcontrols.play();

\_mp3Player.Mp3Play();

}

}

private void btn\_Pause\_Click(object sender, EventArgs e)

{

if (\_mp3Player != null)

{

\_mp3Player.Mp3Pause();

WindowsMediaPlayer.Ctlcontrols.pause();

}

}

private void btn\_Thoat\_Click(object sender, EventArgs e)

{

this.Close();

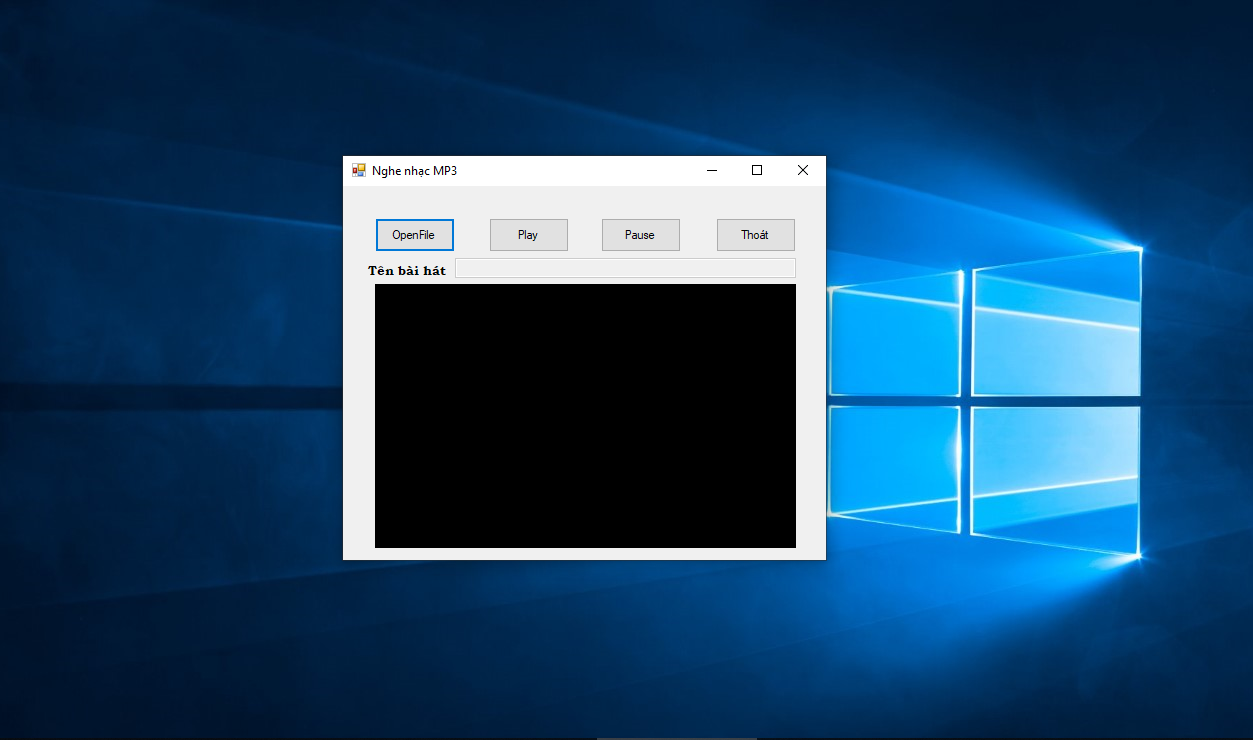
}

}

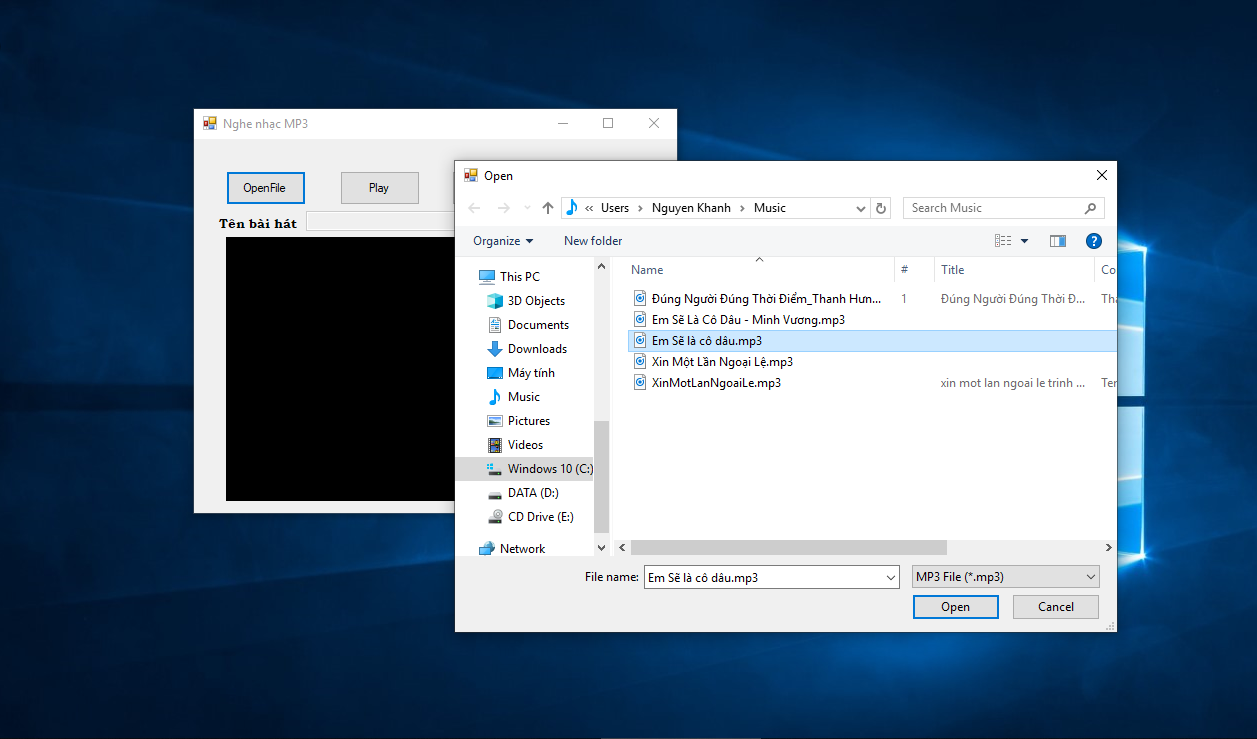
}

II. Hình ảnh mô phỏng quá trình chương trình đang chạy

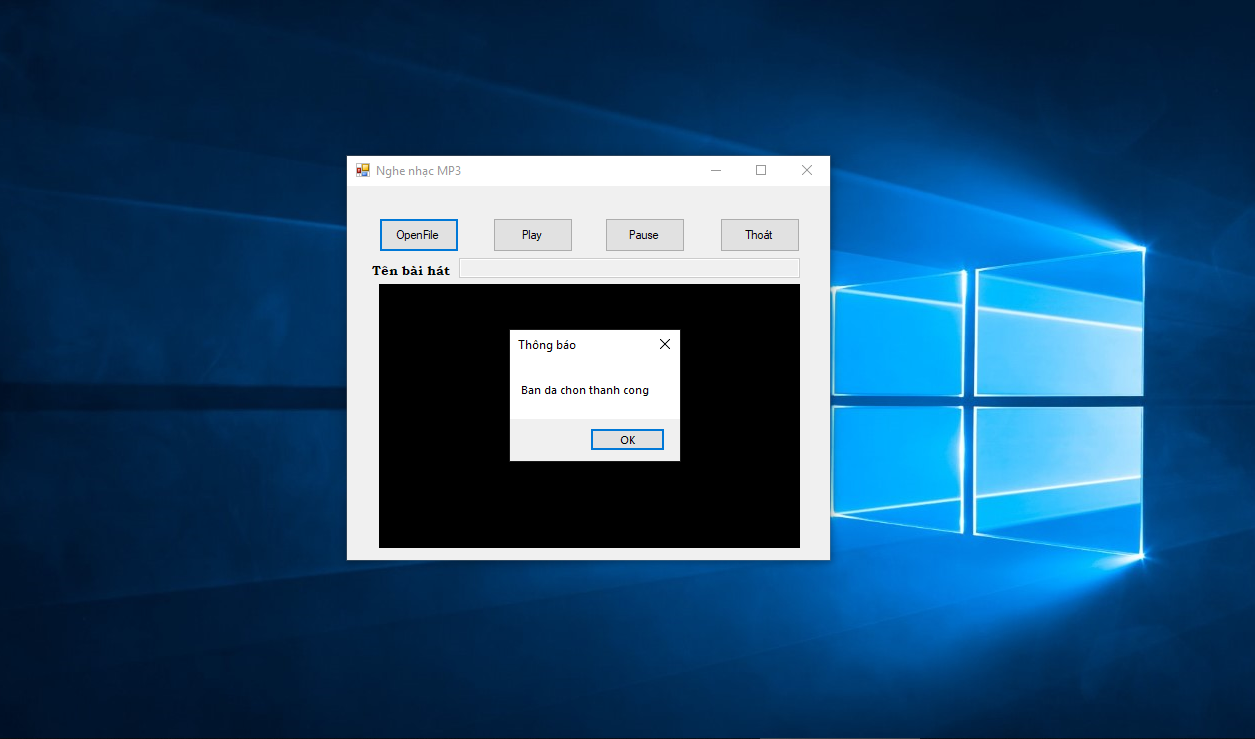
1. Giao diện khi chạy chương trình



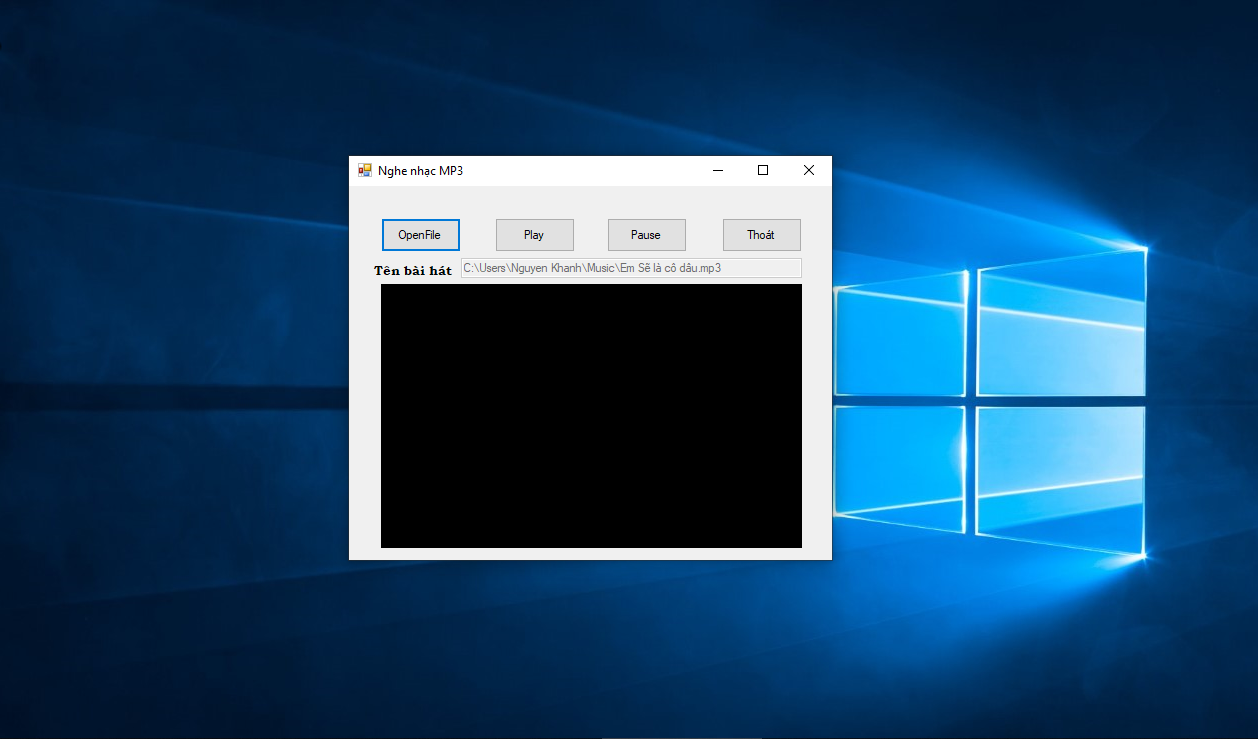
1. Giao diện chọn file nhạc



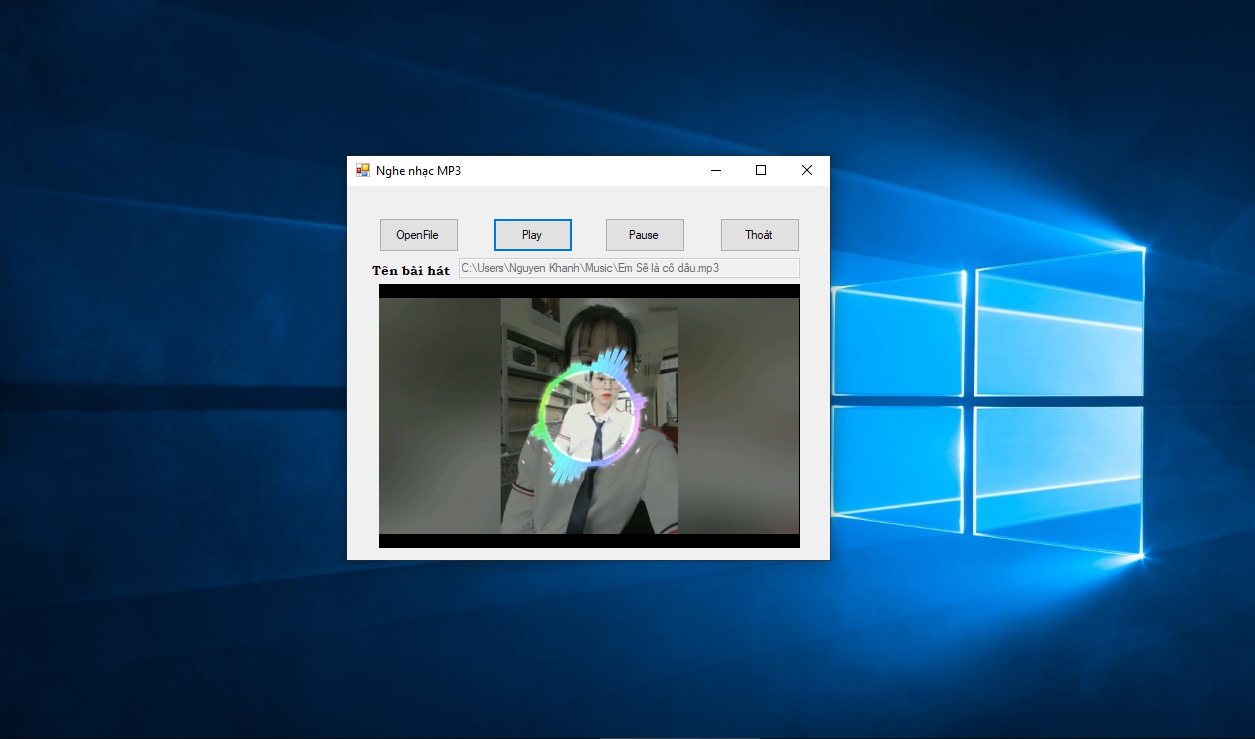
1. Giao diện thông báo đã chọn file thành công



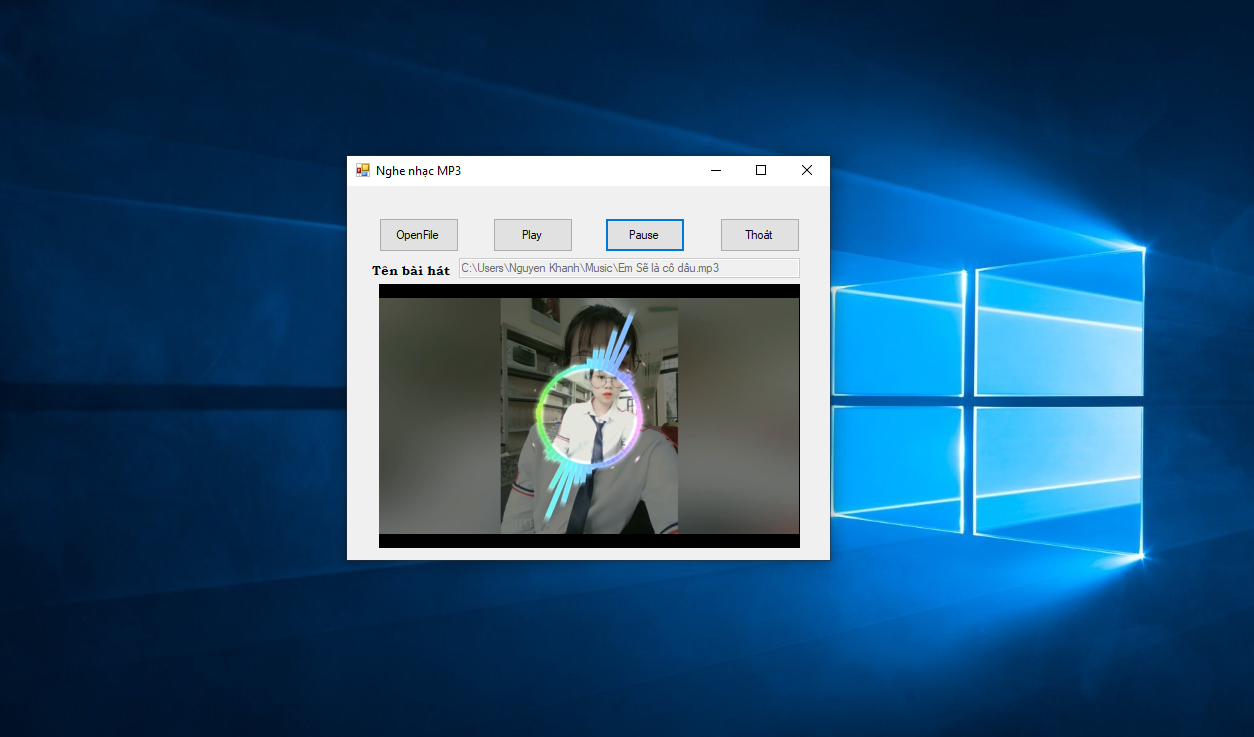
1. Giao diện sau khi chọn bài



1. Giao diện khi nghe nhạc



1. Giao diện khi tạm ngưng



1. Giao diện khi thoát chương trình và có thông báo xác nhận

