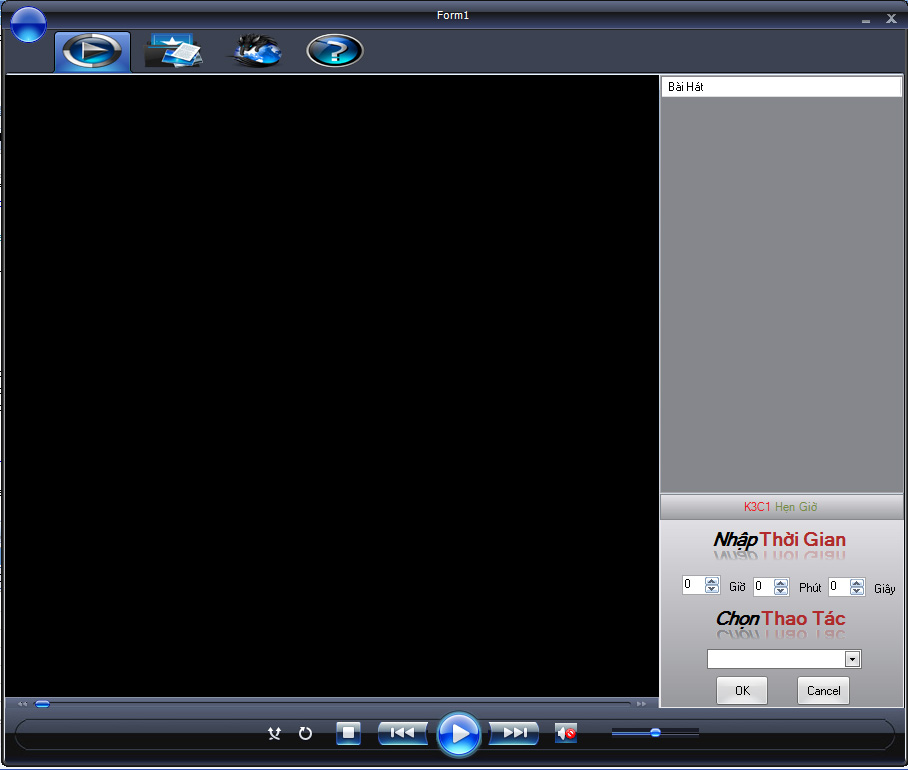
1. **Giới Thiệu Đề Tài:**

* Chương trình nghe nhạc (**K3C1-Mediapro**). Lấy ý tưởng từ chương trình nghe nhạc nổi tiếng Windows Media Player trên HDH Windows
* Sử dụng ngôn ngữ C# để xây dựng chượng trình. Sử dụng thư viện DotnetBar, phần mềm xử lý ảnh Photoshop CS3 để thiết kế giao diện
* Các chức năng cơ bản:
* Phát nhạc nhiều định dạng(bao gồm cả audio và video)
* Lưu danh sách bài hát ưa thích, chơi nhạc ở chế độ random bài hát và repeat danh sách nhạc đang chơi,
* Tìm kiếm nhạc online trên trang: <http://mp3.zing.vn> . Kèm theo tính năng cho phép nghe trực tuyến và download bài hát
* Các playlist(danh sách bài hát yêu thích) được quản lý riêng rất tiện lợi, kèm theo 1 số chức năng tiện lợi như : tìm đến thư mục chứa bài hát, xem thông số bài hát, chuyển đổi qua lại giữa các bài hát trong playlist, ………..
* Chức năng hẹn giờ tắt máy, rất tiện lợi cho người dùng có thói quen nghe nhạc thường xuyên



**GIAO DIỆN CHƯƠNG TRÌNH K3C1-MEDIAPRO**

1. **Tổng Quan Về Chương Trình**
2. **Xây dựng chương trình**

* Để có thể vận hành và chơi 1 file media trong 1 ứng dụng **.NET** thì bắt buộc phải sử dụng 1 thư viện bên ngoài hoặc lời gọi hệ thống**.** Do đó vấn đề trước tiên đối với 1 chương trình nghe nhạc, đó là lựa chọn một thư viện để có thể giải mã và thực thi 1 file media. Hiện nay, có khá nhiều thư viện hỗ trợ cho công việc này như **thư viện wmm.dll, thư viện Quarzt, thư viện wmp.dll, thư viện MCI……………**
* Chương trình nghe nhạc của chúng em sử dụng thư viện wmp.dll do HDH Windows hỗ trợ. Lí do chúng em chọn thư viện wmp.dll cho chương trình của mình là vì thư viện này hỗ trợ các định dạng file media rất đa dạng bao gồm cả audio và video, thao tác nhanh và tiện lợi, không bị ảnh hưởng bởi các file có dấu… Một lí do quan trọng khác nữa là phần mềm nghe nhạc nổi tiếng Windows Media Player hiện cũng đang sử dụng thư viện này. Qua đó cho thấy những tính năng ,ưu điểm của nó đối với việc giải mã và thực thi 1 file media
* Chương trình nghe nhạc **“K3C1-Mediapro”** được xây dựng trên 1 namespace duy nhất là **“dean”,** bao gồm các thành phần cơ bản:
* **Form 1** là form chứa giao diện chính, thực thi các chức năng của chương trình, được thiết kế theo từng Tab riêng với các chức năng riêng biệt
* Form **thongtin** được thiết kế để hiển thị thông tin bài hát
* Class **ham** chứa các hàm tạo và mở 1 danh sách nhạc
* Class **tkonline** thực hiện thao tác tìm kiếm, nghe nhạc và download bài hát trên <http://mp3.zing.vn>
* Class **GeneralValidation** thực hiện tìm kiếm lời bài hát qua mạng

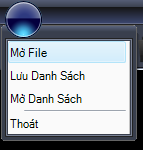
1. **Chương trình gồm các thẻ Tab**C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\playing.pngC:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\danhsach.pngC:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\timkiem1.pngC:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\trogiup.png

TÌM KIẾM

CHƠI NHẠC

TÌM LỜI

DANH SÁCH

* Chơi Nhạc : đây là giao diện chơi nhạc của chương trình
* Danh Sách : chứa các playlist được người dùng tạo hoặc mở, cùng với các chức năng hỗ trợ
* Tìm Kiếm Online : tìm kiếm bài hát trực tuyến và hỗ trợ download thông qua trang web <http://mp3.zing.vn>
* Tìm Lời : tìm lời bài hát bất kỳ trên mạng thông qua nguồn http://www.lyricsplugin.com
* Sau khi cài đặt thư viện wmp.dll, ta thao tác với các file media thông qua đối tượng axWindowsMediaPlayer1
  1. **Tab Chơi Nhạc**
* Menu **Mở\_File :**
* Các biến cần lưu ý :

public string myPlaylist = "UIT\_K3C1";//

public WMPLib.IWMPPlaylist pl;//danh sách file media

public WMPLib.IWMPPlaylistArray plItems;//mảng chứa tất cả các file

public WMPLib.IWMPMedia m1;// biến thể hiện cho mỗi file media

* Hàm CreatePlayLis(OpenFileDialog open,ListView lv) : có tác dụng tạo ra danh sách chơi nhạc dựa trên các bài hát được chọn từ OpenFileDialog. Dữ liệu là các file media đưa vào được xử lý qua đoạn code sau:

int i=0;

foreach (string file in open.FileNames)

{

ListViewItem item = new ListViewItem(open.FileNames[i]); lv.Items.Add(Path.GetFileNameWithoutExtension(item.ToString()));

i++;

m1 = axWindowsMediaPlayer1.newMedia(file);

pl.appendItem(m1);

}

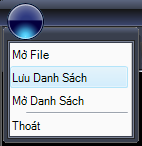
Vòng lặp **foreach** được dùng để duyệt qua từng file được chọn, dữ liệu được đưa lên lv (Listview) để hiện thị trên giao diện chương trình, đồng thời cũng được nạp vào list nhạc **pl**

Cuối cùng là xác định playlist sẽ chơi và play nó:

axWindowsMediaPlayer1.currentPlaylist = pl;

axWindowsMediaPlayer1.Ctlcontrols.play();

* Trong hàm private void MoFile\_Click(object sender, EventArgs e) : gọi hàm CreateLis và truyền tham số. Ngoài ra, dữ liệu về các file media được mở cũng được lưu vào 1 Listbox có tên **lb2** để sử dụng khi cần thiết trong khi viết chương trình(chức năng random bai hát, tìm đến thư mục, tạo node playlist trên Treeview…)
* Menu **Lưu Danh Sách :**

- Sử dụng hàm public void mo\_playlist() của class **ham**

-Hàm mo\_playlist() sẽ kiểm tra danh sách nhạc đang chơi, nếu như

chưa có bài hát nào thì sẽ hiện ra thông báo, ngược lại thì thực thi công việc

- Khai báo hộp thoại SaveFileDialog; biến **“sw”** thuộc kiểu StreamWriter thực hiện thao tác ghi file, dựa trên thông tin là các file media trên danh sách đang chơi nhạc

* Thao tác lấy thông tin đường dẫn file media trong danh sách đang phát :

ax2.currentPlaylist.get\_Item(i).sourceURL.ToString()

* Thực hiện công việc ghi thông tin lên file playlist muốn tạo thông qua hàm WriteLine thuộc class StreamWrite

using (StreamWriter sw = new StreamWriter(filename))

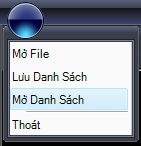
{

for (int i = 0; i < danhsach.Items.Count; i++)

sw.WriteLine(ax2.currentPlaylist.get\_Item(i).sourceURL.ToString());

}

* Menu **Mở Danh Sách :**



* Sử dụng hàm public void mo\_playlist() trong class **ham**
* Hàm này thực hiện mở hộp thoại OpenFileDialog, tìm đến file playlist cần mở. Sử dụng biến **“sr”**  thuộc class StreamReader để đọc dữ liệu từ tệp
* Công việc mở danh sách thực thi qua đoạn code sau:

while ((line = sr.ReadLine()) != null)

{

l1.Items.Add(line);

m11 = ax2.newMedia(line.Trim());

pl1.appendItem(m11);

ax2.currentPlaylist = pl1;

string[] ite = new string[] {ax2.currentPlaylist.get\_Item(kt).getItemInfo("Name").ToString()};

ListViewItem items = new ListViewItem(ite);

danhsach.Items.Add(items);

kt++;

}

* **Các button trên Tab Chơi Nhạc :**

|  |  |
| --- | --- |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\mute_hover.PNG | Có chức năng tắt âm thanh |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\play.png | Play 1 file media |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\pause copy 2 copy1.png | Tạm dừng 1 file media |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\stop copy copy.png | Tắt 1 file media đang mở |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\shuffer_on_hover.PNG | Xáo trộn thứ tự chơi file media ban đầu |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\loop_hover.PNG | Lặp lại danh sách nhạc đang chơi |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\next copy copy.png | Tới bài kế tiếp bài đang chơi trong danh sách |
| C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\back.png | Quay lại bài trước bài đang chơi trong danh sách |

* Các button sử dụng các lời gọi hệ thống do thư viên wmp.dll cung cấp, thông qua biến axWindowsMediaPlayer1
* Button C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\shuffer_on_hover.PNG xây dựng trên ý tưởng dùng 1 biến kiểu Random để cho ra vị trị bất kỳ của 1 file media tiếp theo file media đang phát. Tiếp theo thực hiện xáo trộn vị trí các file trong danh sách đang chơi với vị trí mà hàm Random tạo ra

Khai báo Random rd = new Random(DateTime.Now.Second);

int vt = 0;

* Code thực thi công việc:

for (int i = 0; i < ds.Items.Count; i++)

{

vt = rd.Next(0, ds.Items.Count - 1);

lvi = new ListViewItem(Path.GetFileNameWithoutExtension(lb2.Items[vt].ToString()));

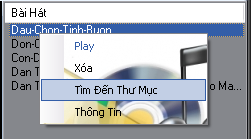
ds.Items[i] = lvi;

m1 = axWindowsMediaPlayer1.newMedia(lb2.Items[vt].ToString());

pl.appendItem(m1);

}

* **Danh Sách Đang Chơi và Menu Chuột Phải:**

****

* Listview có tên **“ds”,** mục đích là hiển thị các bài hát được chọn để play. Với các tính năng thao tác nhanh và tiện lợi, hỗ trợ vấn đề chơi nhạc được tốt hơn
* Công việc đổ dữ liệu vào Listview đã nói ở mục **Mở File**  bên trên
* Tính năng DoubleClick chuột trái để play 1 file media trên Listview:

+) lấy thông tin về bài hát được chọn trên Listview trong playlist đang chơi và gán cho biến **med**

IWMPMedia med = axWindowsMediaPlayer1.currentPlaylist.get\_Item(ds.SelectedIndices[0]);

+) play file media này bằng câu lệnh :

axWindowsMediaPlayer1.Ctlcontrols.playItem(med);

* Menu chuột phải:

|  |  |
| --- | --- |
| Play | Play file media được chọn trên danh sách nhạc đang phát |
| Xóa | Xóa file media được chọn trên danh sách nhạc đang phát |
| Tìm Đến Thư Mục | Tìm đến thư mục chứa file media được chọn |
| Thông Tin | Hiển thị các thông số về file media được chọn |

+) Chức năng **Play** sử dụng hàm DoubleClick chuột vào file media trình bày ở trên

+) Chức năng **Xóa :** lấy vị trí của file media như ở chức năng **Play** , thực hiện 2 thao tác xóa trên playlist nhạc và trên Listview để dảm bảo xóa hoàn toàn khỏi danh sách nhạc đang phát, thao tác xóa:

axWindowsMediaPlayer1.currentPlaylist.removeItem(med);

ds.Items.Remove(sel);

+) Chức năng **Tìm Đến Thư Mục :** lấy vị trí file media và tìm kiếm file đó trong Listbox **lb2**, sử dụng biến kiểu FileInfo để lấy thông tin đường dẫn của file media này. Thực hiện thao tac mở thư mục dựa trên đường dẫn đã có:

int cur = this.ds.FocusedItem.Index;

FileInfo f = new FileInfo(lb2.Items[cur].ToString());

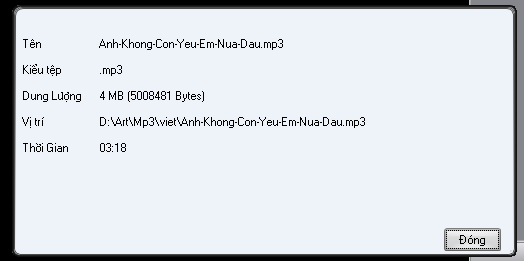
Process.Start(f.Directory.ToString());

+) Chức năng **Thông Tin Bài Hát :**

* Lấy vị trí file media, sử dụng 1 biến kiểu FileInfo để lấy thông tin về file media đó như : tên, đường dẫn, kiểu file, dung lượng……………
* form **thongtin :** sử dụng hàm

public void set(string ten, string kieutep, string dung\_luong, string vitri, string duaration)

hàm này có chức năng là đưa các thông số lấy được từ file media lên form **thongtin**

****

* **Các phím tắt và chức năng kéo thả:**

|  |  |
| --- | --- |
| Phím Enter | Play file media được chọn bằng phím Enter |
| Phím Delete | Xóa file media được chọn bằng phím Delete |
| Chức năng kéo thả | Chơi 1 danh sách nhạc bằng cách kéo thả từ thư mục |

+) Bắt sự kiện nhấn phím Enter : e.KeyCode == Keys.Enter

Thực thi chức năng play 1 file nhạc bất kỳ như chức năng **Play** của menu chuột phải đã nói ở trên

+) Bắt sự kiện nhấn phím Delete : e.KeyCode == Keys.Delete

Thực thi chức năng **Xóa** như trong menu chuột phải của Listview

+) Chức năng kéo thả : sử dụng chức năng Allow Drop của Listview để cho phép kéo thả đối tượng vào Listview.

Hàm thực thi kéo thả: private void ds\_DragDrop(object sender,DragEventArgs e)

Câu lệnh :

string[]s = (string[])e.Data.GetData(DataFormats.FileDrop, false):

có chức năng lấy thông tin dữ liệu của từng đối tượng được kéo thả vào Listview, và đưa dữ liệu này vào 1 mảng kiểu string để được xử lý

Thao tác mở và chơi các file media được kéo thả vào , xử lý code như hàm **CreateLis()**  đã nói ở phần mở file

* Chức năng Hẹn giờ tắt máy:

+) Tạo một đối tượng Timer có tên **hg,** có nhiệm vụ như 1 chiếc đồng hồ đếm ngược cho đến khi thời gian hẹn giờ kết thúc thì sẽ thực thi công việc được chỉ định trong ComboBox

+) Đối tượng ComboBox chứa các Text tương ứng với các chức năng tắt máy tính trên HDH : *Logoff, Restart, Hibernate, Shutdow, Stanby*

+) Các đối tượng NumericUpDown ứng với các đơn vị thời gian giờ, phút, giây

+) Code thực thi:

Tạo một biến **Time** kiểu int có nhiệm vụ lưu thời gian(tính theo giây) người dùng nhập vào để thực thi thao tác tắt máy :

Time = ((int)gio.Value) \* 3600 + ((int)phut.Value) \* 60 + ((int)giay.Value);

Đối tượng Timer thực hiện thao tác đếm ngược, qua mỗi lần đếm biến Time sẽ bị trừ đi 1 cho đến lúc bằng 0 để thực hiện thao tac tắt máy:

if (Time > 0)

{

Time--;

lbComment.Text = "Còn lại " + Time.ToString() + " giây";

}

else

{thao tác tắt máy tương ứng với dòng text trong comboBox được chọn}

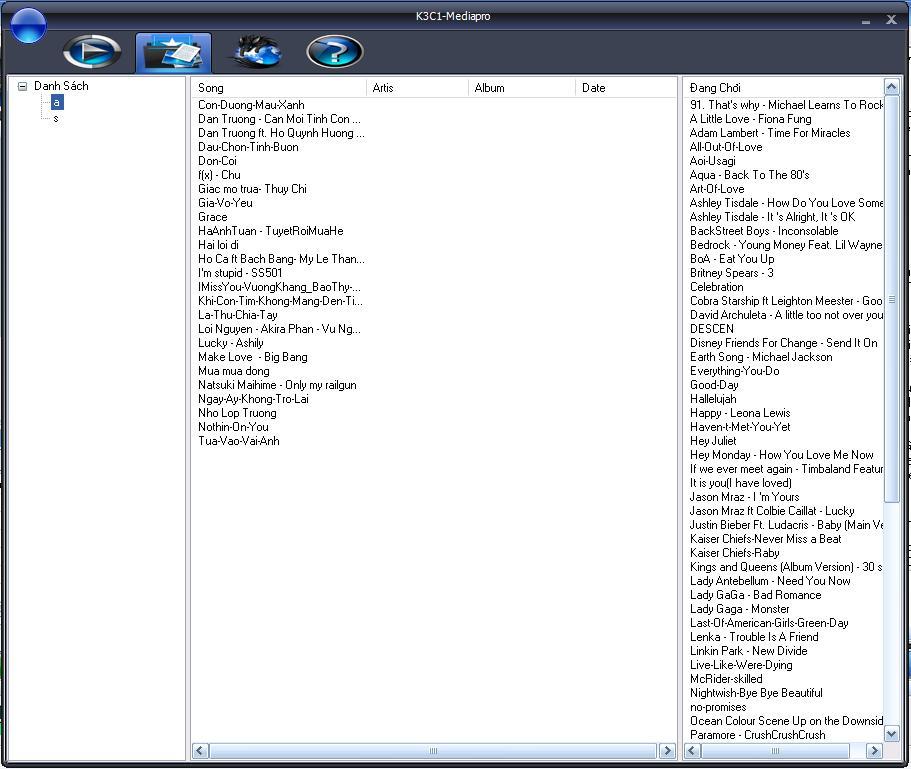
+) Để thực thi chức năng tăt máy tính, cần sử dụng các hàm hệ thống, có trong thư viện “user32.dll”:

Stanby : Application.SetSuspendState(PowerState.Suspend, true, true);

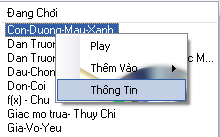
Logoff : ExitWindowsEx(0, 0);

……………………………………..

* 1. **Tab Danh Sách**



**Giao diện chính của Tab khi phát nhạc**

* **Các đối tượng trên Tab**
* TreeView hiển thị các list nhạc mà người dùng tạo hoặc mở trong quá trình nghe nhạc, mỗi Node trên TreeView biểu thị cho 1 danh sách nhạc(playlist)
* listView1 dùng để hiển thị cac bài hát nằm trong playlist khi click vào Node tương ứng trên TreeView.
* listView2 hiển thị danh sách bài hát đang phát. Menu chuột phải hỗ trợ các thao tác trên listView2
* Menu chuột phải cho listView1

|  |  |
| --- | --- |
| Play | Play file media |
| Thêm Vào | Chuyển file media từ danh sách nhạc này sang danh sách nhạc khác |
| Thông Tin | Xem thông tin file media |

* **Các hàm dùng thao tác trên Tab**
* Hàm **taoNode** :
* Tạo Node trên TreeView, tương ứng với danh sách nhạc được mở hoặc được tạo . Dùng một mảng listBox[] để lưu các danh sách được mở hoặc được tạo, sử dụng biến **lb2** để lưu các file media trong từng listBox
* Biến

temp[n] = new ToolStripMenuItem();

temp[n].Text = Path.GetFileNameWithoutExtension(path);

lưu tên của mỗi danh sách nhạc và gán cho menu chuột phải “**Thêm Vào**”(đối tượng “addToToolStripMenuItem”). Sử dụng hàm **add\_click** để thêm các file media vào mỗi danh sách nhạc trên menu chuột phải này

addToToolStripMenuItem.DropDownItems.Add(temp[n]);

temp[n].MergeIndex = n;

temp[n].Click += new EventHandler(add\_click);

* Hàm **add\_click** :
* Có chức năng như đã nói ở trên. Thao tác thực thi bằng cách lấy các file media chứa trong **lb2** tương ứng với vị trí của danh sách nhạc trong listBox[], rồi thêm vào danh sách nhạc tại menu chuột phải “**Thêm Vào**”

ToolStripMenuItem \_temp = (ToolStripMenuItem)sender;

for (int i = 0; i <= n; i++)

{

if (\_temp.MergeIndex == i)

list[i].Items.Add(lb2.Items[so1].ToString());

}

* Hàm **Info(string f)**:
* Thêm 1 Item vào listView1 dựa trên thông tin đưa vào từ biến **f**

listView1.Items.Add(new ListViewItem(new string[] { Path.GetFileNameWithoutExtension(f.ToString()) }));

* Hàm private void treeView1\_AfterSelect
* Thực hiện thao tác xuất các file media ra listView1 khi click vào 1 Node có trên TreeView
* Hàm sẽ lấy vị trí của Node được chọn, tìm đến danh sách nhạc tương ứng nằm trong listBox[] và thực thi thao tác thêm Item lên listView1, thông qua việc sử dụng hàm **Info**
* Hàm private void treeView1\_NodeMouseDoubleClick
* Thực hiện thao tác play 1 danh sách nhạc khi Double\_Click vào Node tương ứng của danh sách nhạc đó trên TreeView
* Công việc đầu tiên hàm thực thi là ngưng phát nhạc nếu chương trình đang chạy, làm trống các danh sách chứa nhạc trên các listView
* Lấy vị trí của Node được chọn như đã trình bày
* Thực thi thao tác play 1 danh sách nhạc . Lấy vị trí của danh sách nhạc trong listBox[], thực hiện thêm bài hát trong danh sách tới listView2

for (int j = 0; j < list[i].Items.Count; j++)

{

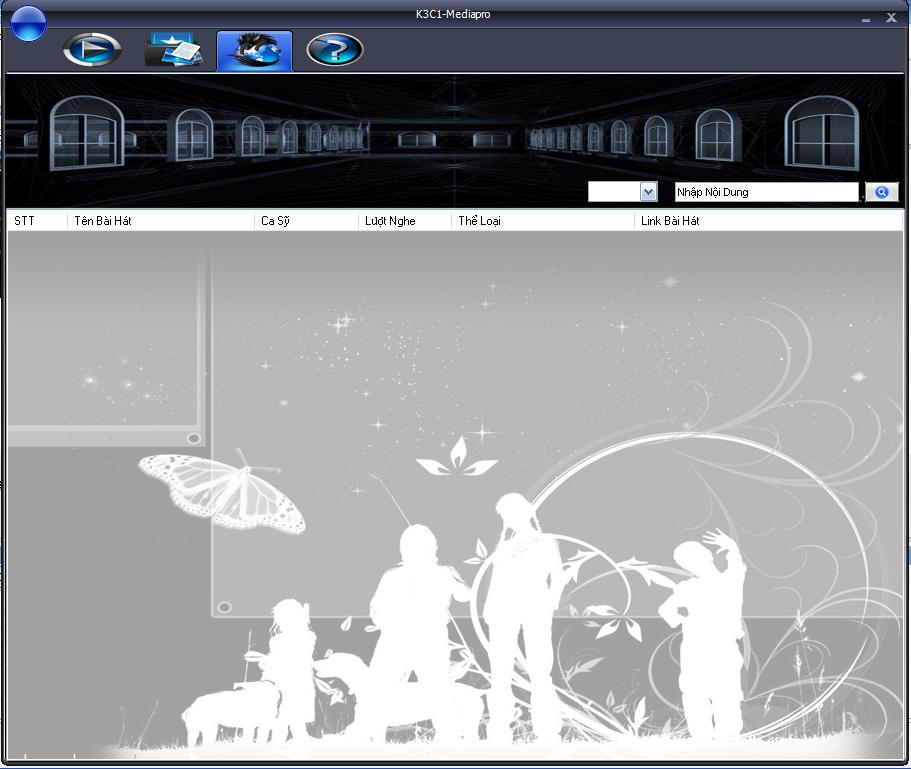
s[j] = list[i].Items[j].ToString();

lb2.Items.Add(s[j]);

listView2.Items.Add(Path.GetFileNameWithoutExtension(s[j]));

}

* Tạo WMPLib.IWMPPlaylist lưu danh sách nhạc, và play
  1. **Tab Tìm Kiếm**



**Giao diện Tab Tìm Kiếm**

* **Các chức năng và thuộc tính trên Tab**
* Lấy thông tin do người dùng nhập vào, sau đó gửi lên trang <http://mp3.zing.vn> . Dữ liệu được trang web trả về nếu yêu cầu tìm kiếm được chấp nhận, thông tin tìm kiếm được mô tả trong listViewEx1
* Thành phần Tab gồm
  + +) ComboBox : nhập thể loại tìm kiếm(Bài Hát , Ca Sĩ)
  + +) TextBox: nhập nội dung cần tìm
  + +) Button tìm kiếm C:\Documents and Settings\Windows XP\Desktop\dean(hoantat)\dean\Resources\search.png
  + +) listViewEx1 : hiển thị thông tin các bài hát tìm được, hỗ trợ play file

+) Menu chuột phải :

|  |  |
| --- | --- |
| Tải Về Máy | Download file về máy, sử dụng hàm **download** được xây dựng trong class **tkonline** |
| Xóa Khỏi Danh Sách | Xóa bài hát khỏi danh sách hiển thị |

* Toàn bộ chức năng tìm kiếm, download file của Tab được xây trên 1 class riêng biệt **tkonline.cs**
* Các hàm trong class **tkonline.cs**
* Hàm public string InitPlayUrl(string s): lấy dữ liệu từ trang web trả về
* Hàm public void FindFromUrl(string path): thực thi công việc tìm kiếm trên web, với thông tin được cung cấp bởi biến **path.** Hàm này sử dụng hàm **InitPlayUrl** để lấy dữ liệu trả về từ web
* Hàm public void Searchnhac() : tổng hợp thông tin về yêu cầu tìm kiếm, sử dụng hàm **FindFromUrl** để thực hiện nhiệm vụ tìm kiếm dựa trên thông tin đã tổng hợp được
* Hàm public void download() : download dữ liệu tìm được từ web, sử dụng phương thức tĩnh download file của lớp System.Net.Webclient
* Tổng quan về code:
* Trước tiên cần có mã HTML của trang web <http://mp3.zing.vn> . Hiểu về mã HTML, để có thể can thiệp vào dữ liệu của web. Dữ liệu gửi trả về dưới dạng mã HTML. Trong quá trình đọc cần sử dụng các hàm thuộc lớp **String** như : StartWith(kiểm tra chuỗi bắt đầu với ký tự nào), Split(cắt chuỗi)
* Để gửi yêu cầu thực thi tới 1 trang web, cần sử dụng đến class HttpWebRequest nằm trong namespace System.Net

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(path);

request.Referer = "http://mp3.zing.vn";

request.UserAgent = "1234556";

* . Để nhận thông tin trả về tử web, cần sử dụng đến class HttpWebResponse. Thông tin phản hồi từ web được lưu vào 1 biến kiểu Stream. Công việc còn lại là thực thi thao tác đọc bằng thư viện StreamReader

Stream responseStream = ((HttpWebResponse)request.GetResponse()).GetResponseStream();

StreamReader reader = new StreamReader(responseStream);

* Thông tin trả về từ Web, sau khi thực hiện thạo tác đọc sẽ trả về 1 chuỗi kiểu string và được gán biến **str10**

str10 = reader.ReadLine().Trim().ToString();

* Dữ liệu trả về từ web hay chính là các file media, sẽ đư**ợ**c đưa vào **pl1** kiểu WMPLib.IWMPPlaylist để quản lý
* Dữ liệu trả về sau khi được lọc bằng hàm StartWith và hàm Split để lấy các thông tin : tên bài hát , ca sỹ, lượt nghe, thể loại, đường dẫn ……. sẽ được gán các biến kiểu **string** và xuất ra listViewEx1.
* Đoạn code dưới đây lấy thông tin bài hát(str2) và đường dẫn đến dữ liệu file trên web(str5)

if (str10.StartsWith("<div class=\"leftInfo\">"))

{

str10 = reader.ReadLine().Trim().ToString();

string[] strArray5 = new string[] { ">", "</a>","<a href=\"", "\"title" };

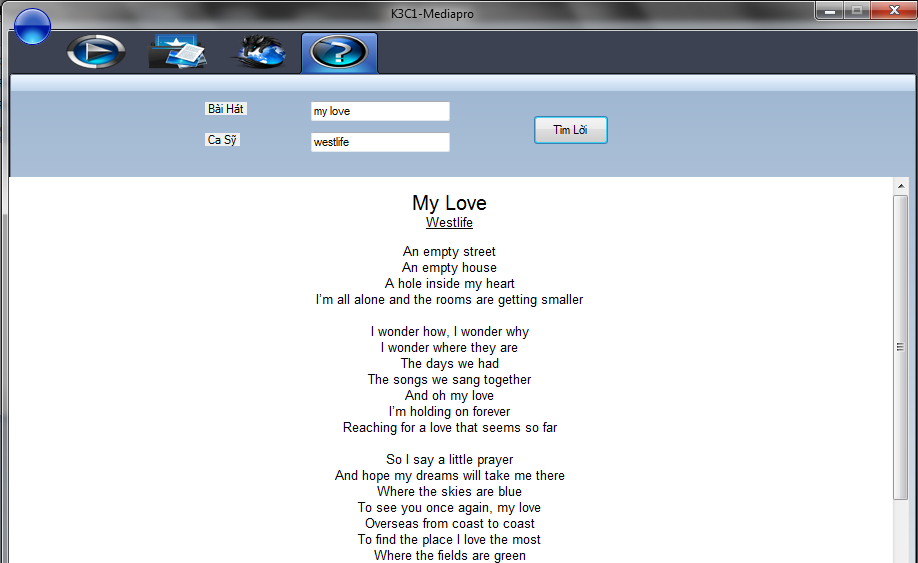
string[] strArray6 = str10.Split(strArray5, StringSplitOptions.RemoveEmptyEntries);

str2 = strArray6[3];

str5 = "http://mp3.zing.vn" + strArray6[1];

}

* 1. **Tab Tìm Lời**

****

* Dữ liệu nhập vào 2 Text : Bài Hát, Ca Sỹ. Các ký tự khoảng trắng được chuyển đổi thành “%20”. Dữ liệu trả về sau khi tìm kiếm được đặt trên WebBrowser
* Thao tác tìm kiếm lời trên Web được thực thi trong class GeneralValidation,
* Các hàm trong class
* Hàm IsUrl : lấy dữ liệu nhập vào và tiến hành xử lý, trả về kết quả nếu tìm được là lời bài hát