**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

**Báo cáo bài tập lớn**

**Lập trình window**

|  |  |
| --- | --- |
| Giáo viên: | Phạm Văn Hà |
| Lớp: | KTPM3\_K11 |
| Thành viên: | Nguyễn Trọng Hải |
|  | Nguyễn Phú Luật |
|  | Phan Quang Trường |
|  |
|  |
| Hệ thống: | ***Phần mềm tra cứu và tìm kiếm bài hát*** |

Hà nội, Năm 2018

**LỜI MỞ ĐẦU**

Kinh tế ngày càng phát triển, nhu cầu giải trí của con người cũng tăng theo. Để thuận tiện cho người dùng có những công cụ giải trí tốt...Chúng em xây dựng hệ thống tra cứu và tìm kiếm bài hát . Trong phạm vi của bài tập lớn, nhóm em xin trình bày việc quản lý và tra cứu bài hát. Bài thiết kế hoàn thành là kết quả của sự học tập, nghiên cứu của nhóm và hướng dẫn của thầy cô môn “Lập trình windown”.

Do sự hạn chế về kiến thức cũng như hiểu biết về lĩnh vực, nên bài viết không tránh khỏi thiếu xót. Vì vậy chúng em mong nhận được sự chỉ bảo của cô giáo, giúp chúng em có được một bài viết hoàn chỉnh hơn.

Bảng phân việc:

|  |  |
| --- | --- |
| **Tên thành viên** | **Công việc cụ thể** |
| Nguyễn Trọng Hải | - Phân tích thiêt kế hệ thống.  - Admin view:  + Tìm kiếm.  + Nghe nhạc.  + Thêm danh sách bài hát.  + Quản lý bài hát. |
| Nguyễn Phú Luật | - Client View:  + Nghe nhạc.  + Tìm kiếm.  + Tải bài hát.  + Chọn bài hát theo chuyên mục.  + Lựa chọn bài hát yêu thích.  + Xem lời bài hát. |
| Phan Quang Trường | - Thiết kế database.  - Admin view:  + Quản lý người dùng.  + Quản lý thể loại. |

**MỤC LỤC**

[Chương 1. Khảo sát hệ thống 4](#_Toc532555042)

[1.1 Khảo sát sơ bộ 4](#_Toc532555043)

[1.1.1 Mục tiêu 4](#_Toc532555044)

[1.1.2 Các yêu cầu chức năng 4](#_Toc532555045)

[1.1.3 Các yêu cầu phi chức năng 4](#_Toc532555046)

[Chương 2. Mô hình hóa use case 5](#_Toc532555047)

[2.1 Biểu đồ use case 5](#_Toc532555048)

[2.1.1 Các use case của người dùng. 6](#_Toc532555049)

[2.1.2 Các use case của admin. 7](#_Toc532555050)

[2.2 Giao diện. 10](#_Toc532555051)

[2.3 Biểu đồ các lớp Entity của hệ thống(Cơ sở dữ liệu) 20](#_Toc532555052)

[Chương 3. Chương Trình và các kết quả thu được 21](#_Toc532555053)

[3.1 Tổ chức thư mục trong project 21](#_Toc532555054)

[3.2 Các kỹ thuật xử lý 25](#_Toc532555055)

[3.3 Một số hình ảnh minh họa 33](#_Toc532555056)

# Chương 1. Khảo sát hệ thống

## Khảo sát sơ bộ

### Mục đích

Mục tiêu của việc thực hiện bài tập lớn này là để xây dựng phần mềm tìm kiếm và quản lý bài hát “\*.mp3”. Giúp cho việc giải trí của con người trở nên dễ dàng hơn. Ngoài việc thưởng thức những bài hát hay nhất mọi thời đại. Hệ thống còn cho phép quản lý bài hát một cách dễ dàng thao tác đơn giản.

### Các yêu cầu chức năng

* **Người quản trị**
* Đăng nhập.
* Nghe nhạc.
* Quản lý bài hát.
* Thêm danh sách bài hát.
* Quản lý thể loại.
* Quản lý người dùng.
* **Người dùng**
* Nghe nhạc.
* Tìm kiếm bài hát.
* Chọn nhạc theo chuyên mục.
* Tải bài hát.
* Lựa chọn bài hát yêu thích.
* Xem lời bài hát.

### Các yêu cầu phi chức năng

* Hiệu năng hoạt động: ổn định, hiệu suất hoạt động tốt.
* Tính khả dụng: giao diện bắt mắt, dễ sử dụng, phù hợp với tất cả mọi nguời.
* Tính tin cậy:  bảo trì và khắc phục lỗi hiệu quả.
* Chức năng tìm kiếm, phản hồi, đánh giá hoạt động hiệu quả.

# Mô hình hóa use case

## Biểu đồ use case

### Các use case của người dùng.

1. Tên use case: Tìm kiếm bài hát.

* Mô tả vắn tắt: Tìm kiếm bài cho phép người dùng tìm kiếm các bài hát nằm trong CSDL.
* Mô tả chi tiết:
  + Luồng cơ bản:
    - Người dùng nhập tên bài hát hoặc tên ca sĩ vào thanh tìm kiếm.
    - Hệ thống hiện những bài hát có tên gần đúng với tên bài hát hoặc tên ca sĩ lên trên màn hình.
  + Luồng rẽ nhánh:
    - Người dùng xóa hết những kí tự nhập vào thanh tìm kiếm hoặc người nhập không nhập kí tự nào vào thanh tìm kiếm mà click vào biểu tượng tìm kiếm.
    - Hệ thống sẽ hiện toàn bộ bài hát trong cơ sở dữ liệu lên màn hình.

1. Tên use case: Nghe nhạc

* Mô tả vắn tắt: Nghe nhạc cho phép người dùng thưởng thức bài hát từ danh sách bài hát lấy từ CSDL.
* Mô tả chi tiết
  + Luồng cơ bản :
    - Người dùng click vào nút play nhạc .
    - Hệ thống sẽ phát bài nhạc theo lựa chọn play của người dùng. Đồng thời hiện lên giao diện phát nhạc gồm nhạc phổ, play, stop, pause, tua và hệ thống hiện lời bài hát.
  + Luồng rẽ nhánh:
    - Người dùng ấn nút stop nhạc của bài hát đang phát, hoặc ấn nút play bài hát khác.
    - Hệ thống sẽ dừng phát bài nhạc đang phát lại và có thể phát bài hát khác nếu người dùng click vào nút play của bài hát khác.

1. Tên use case: Chọn nhạc theo chuyên mục

* Mô tả vắn tắt: Use case này cho phép người lựa chọn bài hát theo chuyên mục.
* Mô tả chi tiết
  + Luồng cơ bản:
    - Người dùng click vào các danh mục hoặc chủ đề có sẵn trên giao diện người dùng
    - Hệ thống sẽ hiển thị các bài hát theo đúng danh mục hoặc chủ đề người dùng đã chọn.

1. Tên use case: Lựa chọn bài hát yêu thích.

* Mô tả vắn tắt: Use case này cho phép người dùng chọn lựa những bài hát yêu thích vào danh sách yêu thích của mình.
* Mô tả chi tiết
  + Luồng cơ bản:
    - Người dùng click vào biểu tượng trái tim trên giao diện hiện bài hát.
    - Hệ thống sẽ hiển thị lại biểu tượng trái tim yêu thích trên bài hát đó đồng thời cho bào hát đó vào danh mục bài hát yêu thích.
  + Luồng rẽ nhánh:
    - Người dùng click tiếp vào biểu tượng trái tim để bỏ yêu thích bào hát.
    - Hệ thống sẽ hiển thị lại biểu tượng trái tim không yêu thích trên bài hát đồng thời bỏ bài đó khỏi danh mục yêu thích.

1. Tên use case: Quản lý nhạc cá nhân

* Mô tả vắn tắt: Use case này cho phép người dùng tải các bài hát trong cơ sở dữ liệu về cá nhân và quản lý chúng.
* Mô tả chi tiết
  + Luồng cơ bản
    - Người dùng có thể tải các bài hát vào danh mục cá nhân bằng cách ấn vào nút tải trên bào hát . Người dùng có thể xóa những bài hát ra khỏi danh mục cá nhân bằng cách ấn vào nút xóa trên bài hát.
    - Hệ thống sẽ thêm bài hát được người dùng chọn tải xuống và xóa những bài hát người dùng chọn xóa ra khỏi danh mục cá nhân.

### Các use case của admin.

1. Tên use case: Tìm kiếm bài hát.

* Mô tả vắn tắt: Tìm kiếm bài cho phép người dùng tìm kiếm các bài hát nằm trong CSDL.
* Mô tả chi tiết:
  + Luồng cơ bản:
    - Người dùng nhập tên bài hát hoặc tên ca sĩ vào thanh tìm kiếm.
    - Hệ thống hiện những bài hát có tên gần đúng với tên bài hát hoặc tên ca sĩ lên trên màn hình.
  + Luồng rẽ nhánh:
    - Người dùng xóa hết những kí tự nhập vào thanh tìm kiếm hoặc người nhập không nhập kí tự nào vào thanh tìm kiếm mà click vào biểu tượng tìm kiếm.
    - Hệ thống sẽ hiện toàn bộ bài hát trong cơ sở dữ liệu lên màn hình.

1. Tên use case: Quản lý bài hát.

* Mô tả vắn tắt: Use case này cho phép admin lựa chọn quản lý bài hát trong cơ sở dữ liệu.
* Mô tả chi tiết:
  + Luồng cơ bản:
    - Dùng sau khi đăng nhập vào hệ thống. Hệ thống hiện phần view cho phép người dùng quản lý bài hát.
    - Trên giao diện quản lý bài hát. Người dùng chọn biểu tượng  tương ứng với bài hát để sửa bài hát, Hệ thống sẽ view giao diện sửa bài hát. Người dùng thực hiện sửa và ấn nút sửa để sửa bài hát. Người dùng có thể chọn biểu tượng  để xóa bài hát tương ứng.
    - Trên giao diện quản lý bài hát. Người dùng có thể chọn “Thêm bài hát”, hệ thống sẽ đưa ra giao diện thêm bài hát, người dùng thực hiện nhập các thông tin bài hát và ấn thêm. Hệ thống sẽ lưu thông tin bài hát vào bảng tblSing trong csdl.
  + Luồng rẽ nhánh:
    - Người dùng trong lúc sửa hoặc thêm bài hát ấn tắt form sửa hoặc ấn thoát. Hệ thống quay về giao diện quản lý bài hát.
    - Người dùng ấn xóa nhưng không xác nhận xóa. Hệ thống cũng quay về giao diện quản lý bài hát.

1. Tên use case: Quản lý người dùng.

* Mô tả vắn tắt: Use case này cho phép admin quản lý các tài khoản truy cập vào cơ sở dữ liệu.
* Mô tả chi tiết:
  + Luồng cơ bản:
    - Sau khi đăng nhập vào hệ thống. Người dùng ấn “Quản lý người dùng”. Hệ thống hiển thị giao diện quản lý người dùng.
    - Trên giao diện quản lý người dùng. Người dùng chọn một bản ghi trên danh sách người dùng. Ấn “Xóa” để xóa người dùng khỏi hệ thống. Ấn “Sửa” để thực hiện sửa thông tin người dùng trong hệ thống.
    - Trên giao diện quản lý người dùng. Người dùng có thể chọn “Thêm người dùng”, hệ thống sẽ đưa ra giao diện thêm người dùng, người dùng thực hiện nhập các thông tin người dùng và ấn thêm. Hệ thống sẽ lưu thông tin người dùng vào bảng tblUser trong csdl.
  + Luồng rẽ nhánh:
    - Người dùng trong lúc sửa hoặc thêm bài hát ấn tắt form sửa hoặc ấn thoát. Hệ thống quay về giao diện quản lý người dùng.
    - Người dùng ấn xóa nhưng không xác nhận xóa. Hệ thống cũng quay về giao diện quản lý người dùng.

1. Tên use case: Quản lý thể loại.

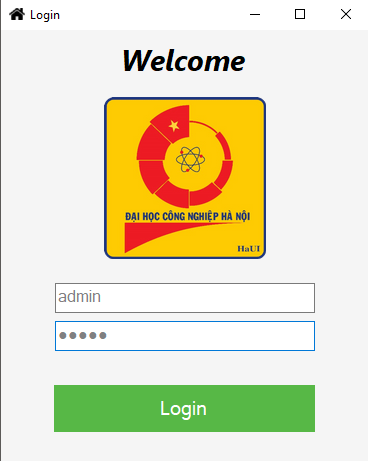
* Mô tả vắn tắt: Use case này cho phép admin quản lý các thể loại bài hát trong CSDL.
* Mô tả chi tiết:
  + Luồng cơ bản:
    - Sau khi đăng nhập vào hệ thống. Người dùng ấn “Quản thể loại”. Hệ thống hiển thị giao diện quản lý thể loại.
    - Trên giao diện quản lý thể loại. Người dùng chọn một bản ghi trên danh sách người dùng. Ấn “Xóa” để xóa thể loại nhạc . Ấn “Sửa” để thực hiện sửa thông tin thể loại nhạc trong hệ thống.
    - Trên giao diện quản lý thể loại. Người dùng có thể chọn “Thêm thể loại”, hệ thống sẽ đưa ra giao diện thêm thể loại, người dùng thực hiện nhập các thông tin thể loại và ấn thêm. Hệ thống sẽ lưu thông tin thể loại vào bảng tblCategory trong csdl.
  + Luồng rẽ nhánh:
    - Người dùng trong lúc sửa hoặc thêm bài hát ấn tắt form sửa hoặc ấn thoát. Hệ thống quay về giao diện quản lý thể loại.
    - Người dùng ấn xóa nhưng không xác nhận xóa. Hệ thống cũng quay về giao diện quản lý thể loại.

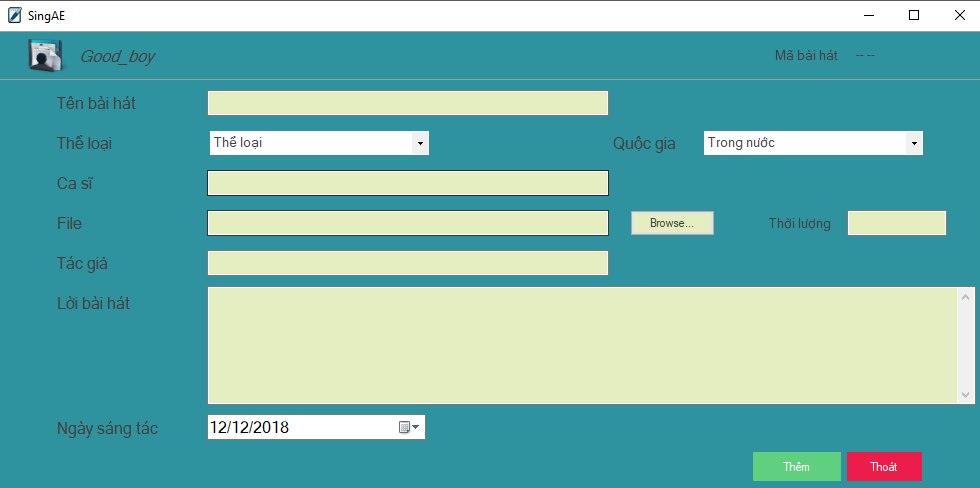
1. Tên use case: Thêm danh sách bài hát.

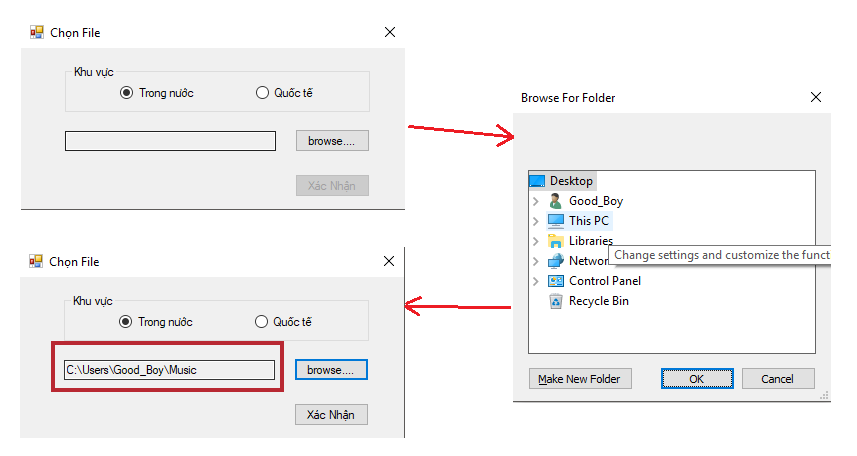
* Mô tả vắn tắt: Use case này cho phép admin thêm danh sách bài hát từ 1 folder trên máy tính.
* Mô tả chi tiết:
  + Luồng cơ bản:
    - Sau khi đăng nhập vào hệ thống. Người dùng ấn “Thêm danh sách bài hát”. Hệ thống hiển thị giao diện thêm danh sách bài hát.
    - Trên giao diện thêm danh sách bài hát. Người dùng ấn “browse” hệ thống đưa ra giao diện chọn đường dẫn thư mục chứa bài hát.
    - Người dùng chọn một thư mục chứa danh sách bài hát cần thêm rồi ấn “xác nhận”. Hệ thống tìm kiếm các bản ghi có đuôi “\*.mp3” và kiểm tra chỉ thêm những bài hát chưa có trong CSDL và báo lại cho người dùng.
    - Người dùng xác nhận báo cáo từ hệ thống. Use case kết thúc.
  + Luồng rẽ nhánh:
    - Thư mục chứa bài hát không có bài hát nào . Hệ thống thông báo không có bài hát. Use case kết thúc.

## Giao diện.

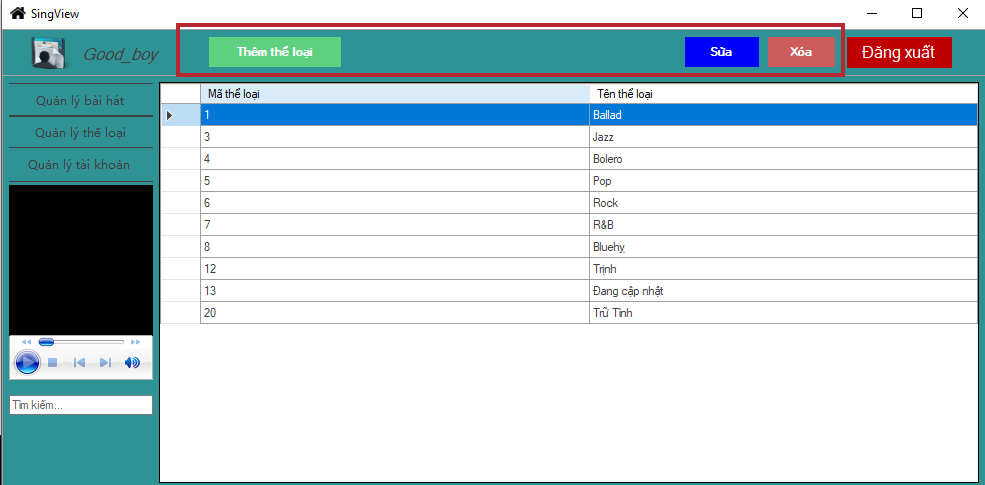
* **Giao diện admin.**
  + Giao diện đăng nhập vào hệ thống.

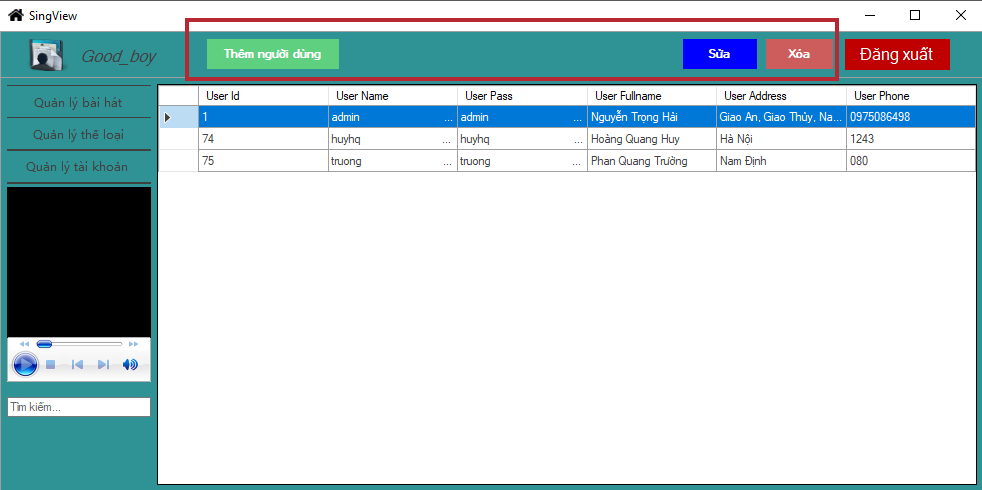
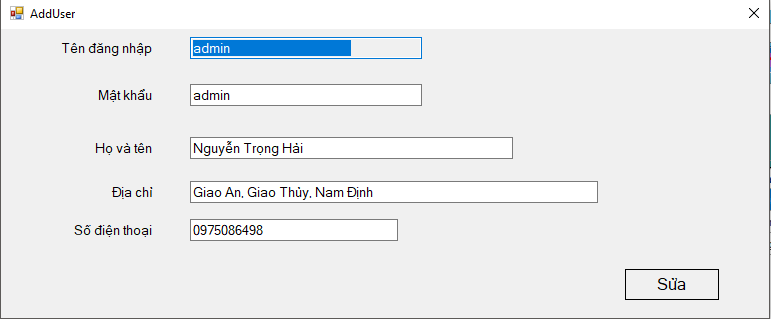
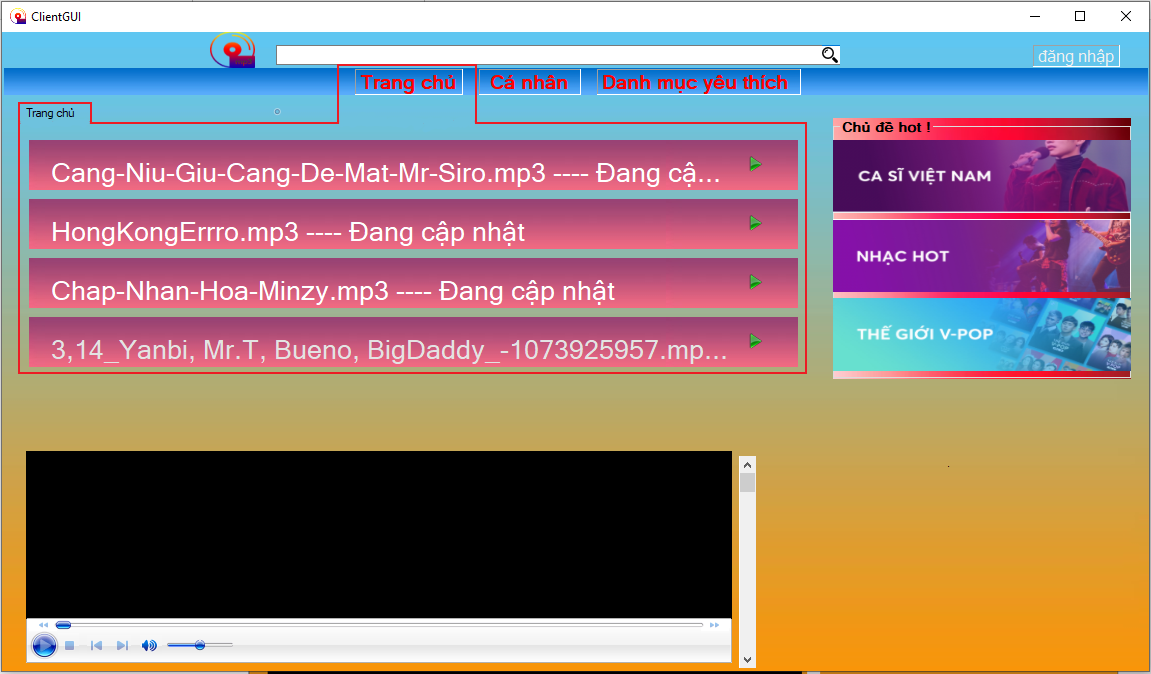
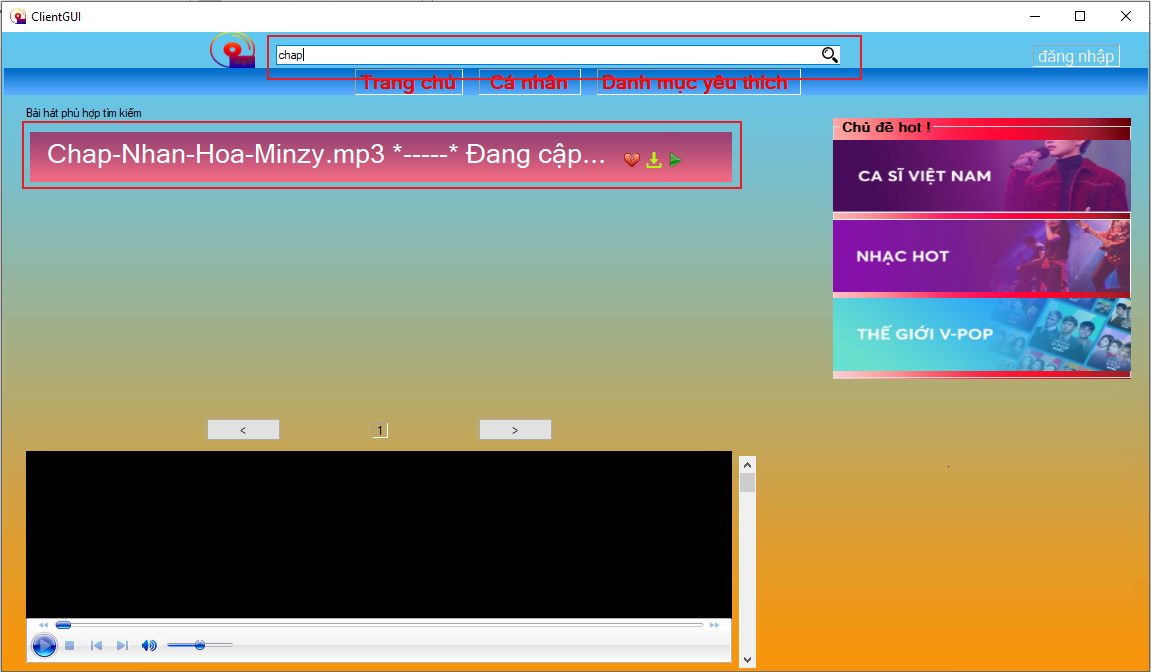


* + Giao diện quản lý bài hát
  + Thêm bài hát
  + Thêm danh sách bài hát



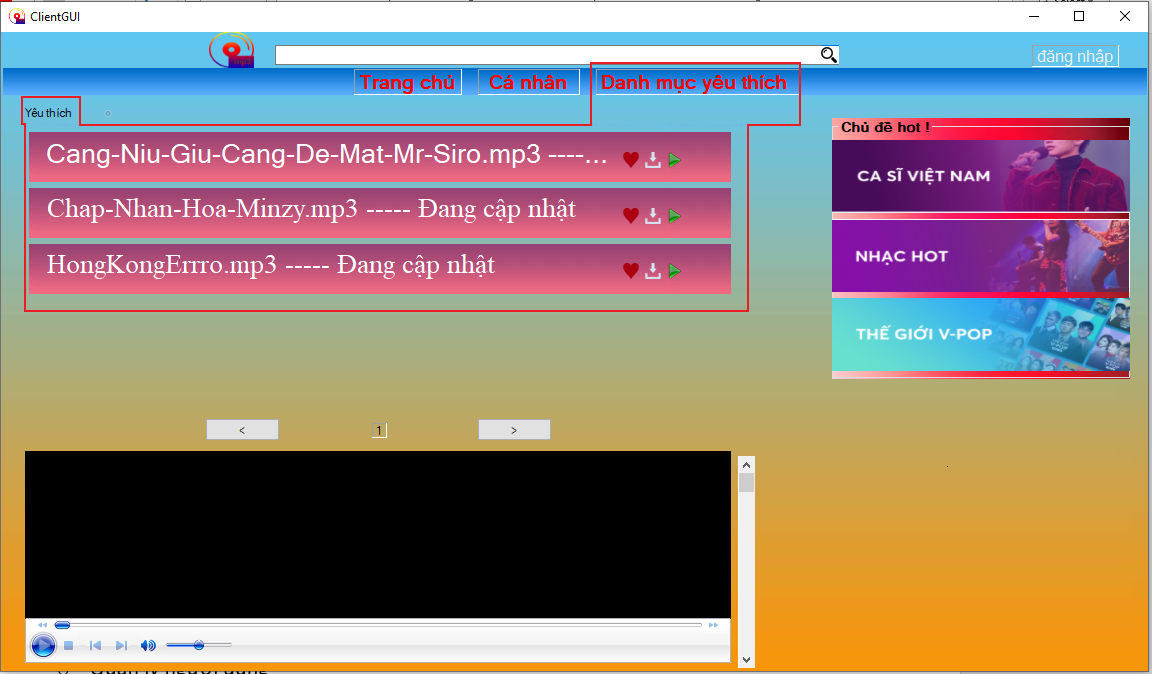
* + Quản lý thể loại

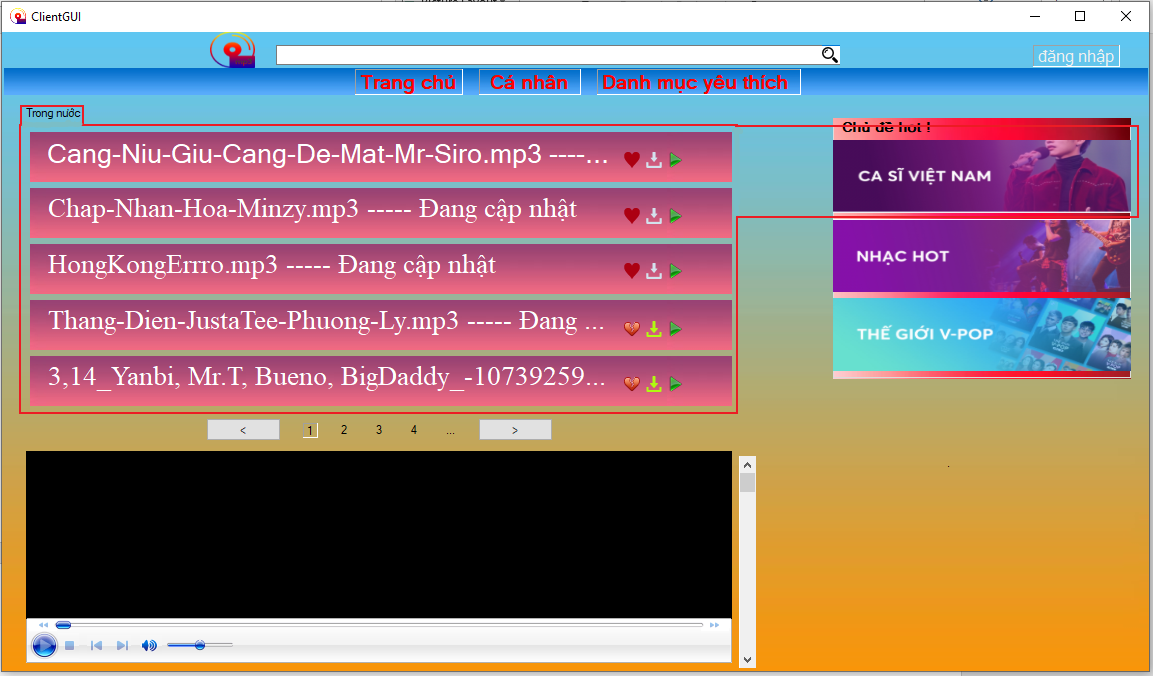
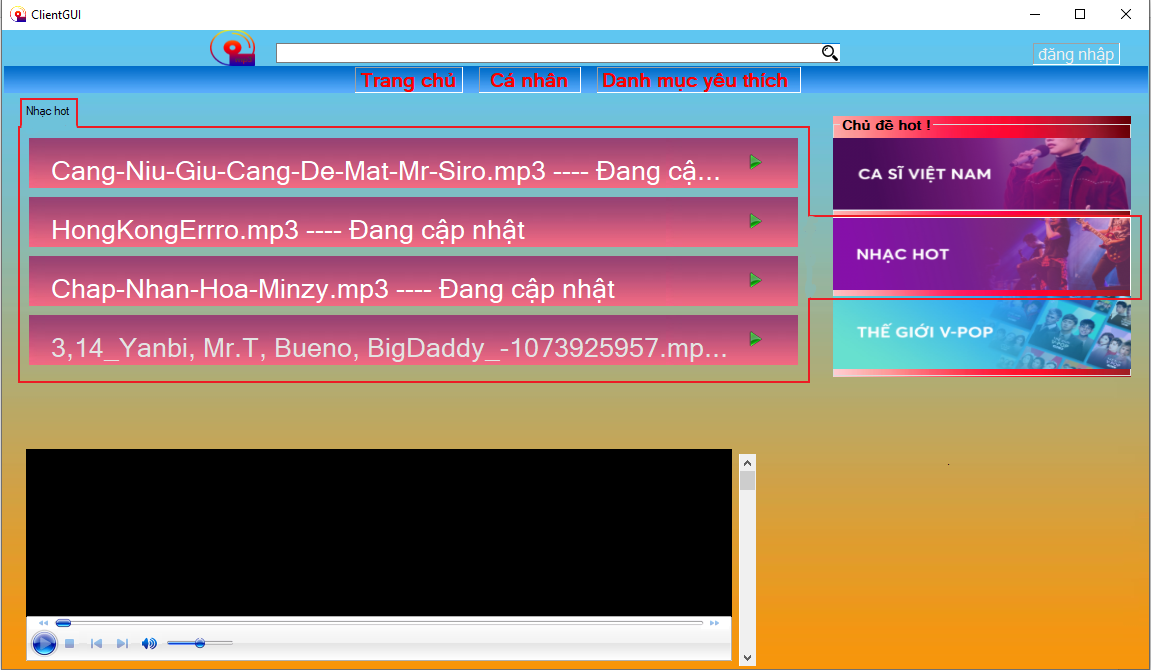
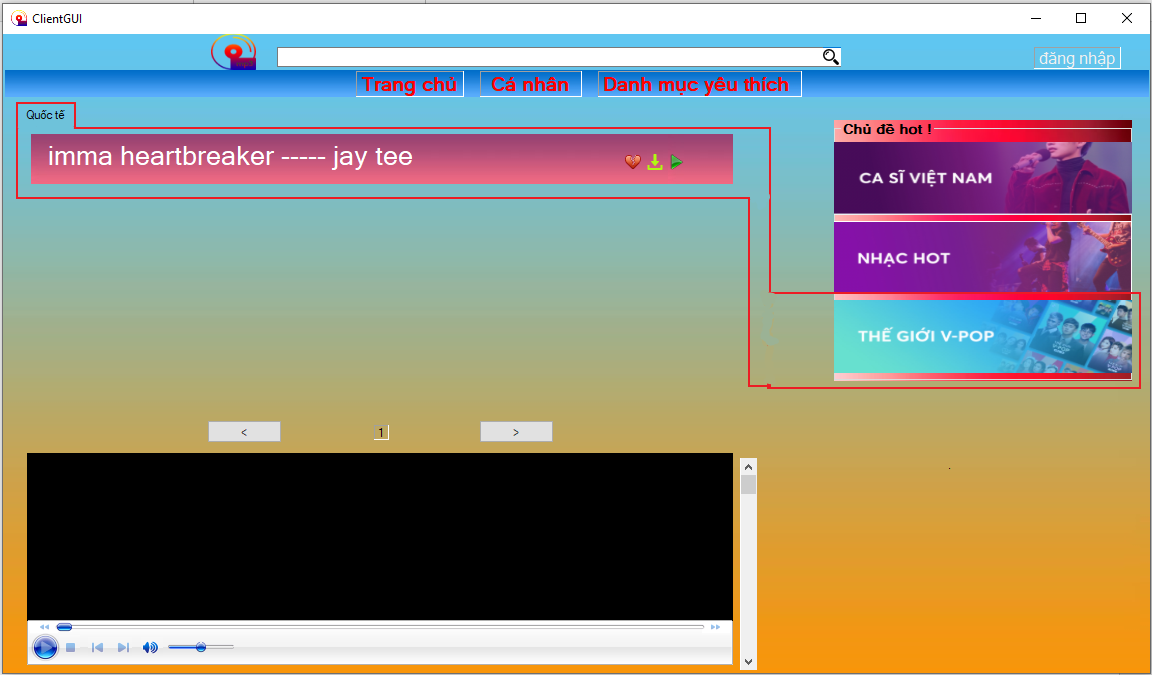
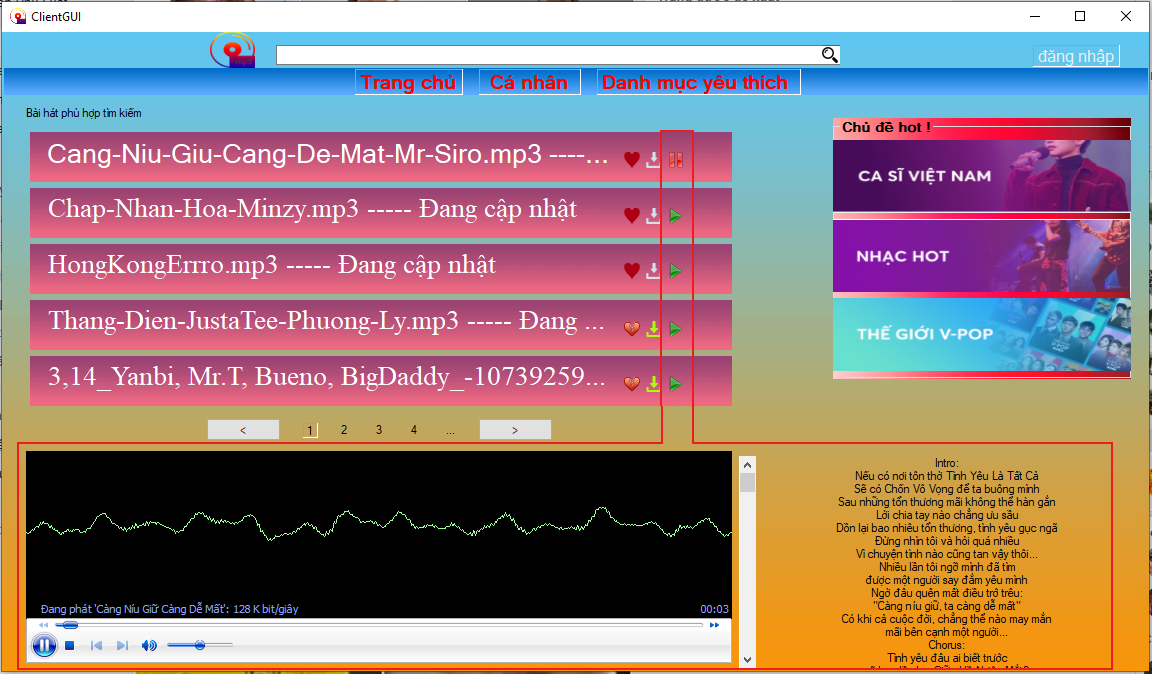
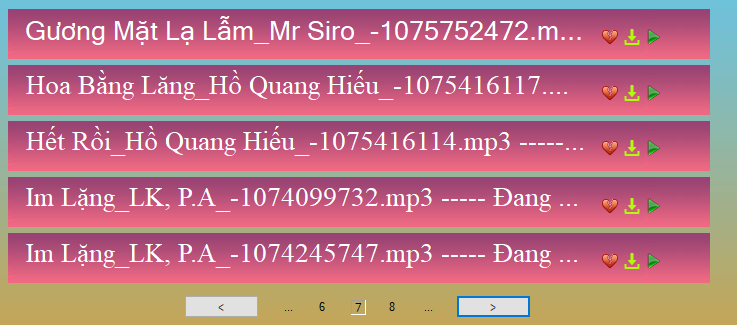


* + Quản lý người dùng
  + 
* **Giao diện người dùng.**
  + Giao diện người dùng
  + Trang chủ
  + Tìm kiếm
  + Bài hát đã tải vào mục cá nhân

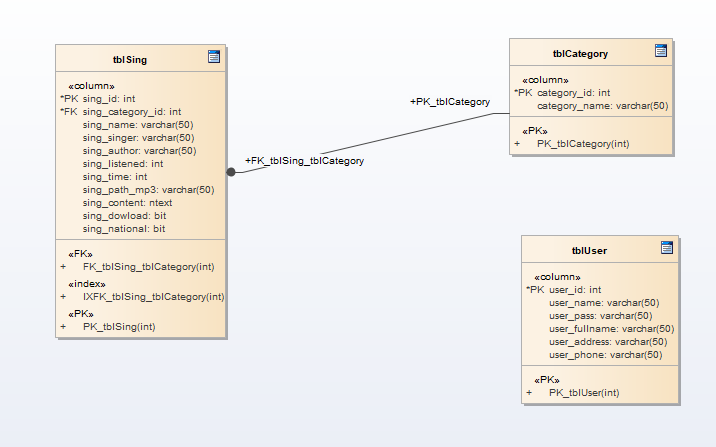


* + Bài hát trong danh mục yêu thích



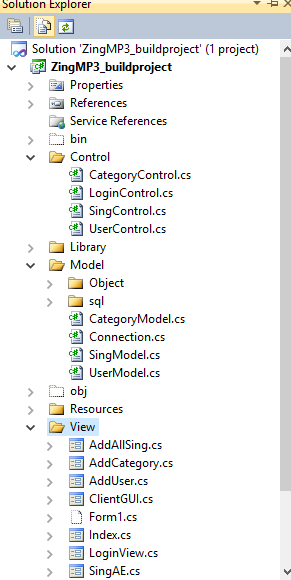
* + Bài hát chủ đề nhạc trong nước
  + Bài hát trong chủ đề nhạc hot
  + Bài hát chủ đề nhạc thế giới
  + Giao diện phát nhạc
  + Phân trang bài hát

## Biểu đồ các lớp Entity của hệ thống(Cơ sở dữ liệu)



# Chương Trình và các kết quả thu được

## Tổ chức thư mục trong project

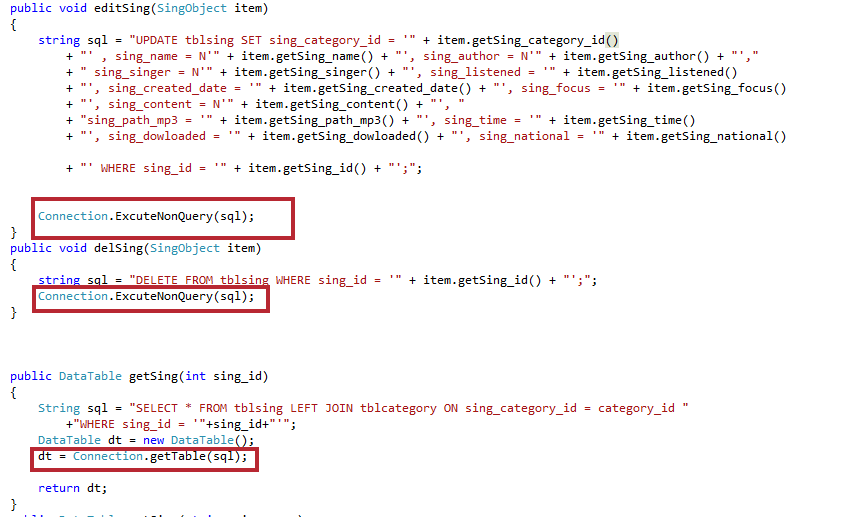


* **Tổng quan :** Project được tổ chức theo mô hình 3 layer.

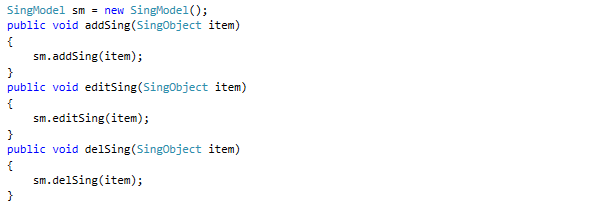
Giúp project dễ dàng trong việc quản lý (Update và Bảo trì)

* **Namespace Model**
* Sử dùng một lớp chỉ thao tác với cơ sở dữ liệu và trả về dữ liệu để xử lý.

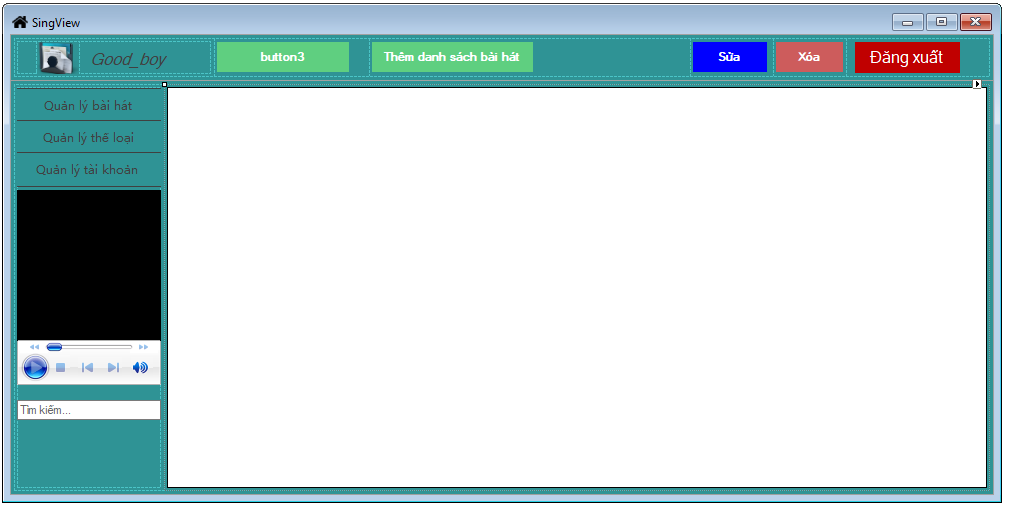


* Các lớp model khác xử dụng lại hai phương thức của lớp Connection để truy xuất dữ liệu. 
* **Namespace Control**
* Sử dụng giữ liệu từ namespace model để xử lý trả về đối tượng





* **Namespace View**
* Gồm các form cho người dùng thao tác.(hình ảnh được lấy từ form quản lý của admin)



* Xử dụng các đối tượng và phương thức cung cấp bởi namespace Control để thao tác(hình ảnh được lấy từ use case “tìm kiêm”)



## Các kỹ thuật xử lý

* Coppy file “\*.mp3” và chuyển thành đường dẫn tương đối.

class File\_Me

{

public static string CoppyFile(string file\_mp3\_temp, string file\_mp3)

{

//---------------Coppy file------------------

string[] temp;// = file\_mp3\_temp.Split('\\');

string fileName = "";

string sourcePath = file\_mp3\_temp;

temp = file\_mp3.Split('\\');

string directoryPath = "";

for (int i = 0; i < temp.Length - 1; i++)

{

directoryPath += temp[i] + "\\";

}

directoryPath = directoryPath.Substring(0, directoryPath.Length - 1);

// Use Path class to manipulate file and directory paths.

string sourceFile = System.IO.Path.Combine(sourcePath, fileName);

// To copy a folder's contents to a new location:

// Create a new target folder, if necessary.

try

{

if (!System.IO.Directory.Exists(directoryPath))

{

System.IO.Directory.CreateDirectory(directoryPath);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

// To copy a file to another location and

//overwrite the destination file if it already exists.

if (System.IO.File.Exists(sourcePath))

{

if (!System.IO.File.Exists(file\_mp3))

{

System.IO.File.Copy(sourceFile, file\_mp3, true);

}

}

else

{

MessageBox.Show("Source path does not exist!");

file\_mp3 = "";

}

return file\_mp3;

}

}

* Thêm danh sách bài hát từ thư mục của người dùng.

//Từ đường dẫn của thư mục kiểm tra các tệp tin có đuôi mở rộng “\*.mp3”

foreach (string file in Directory.EnumerateFiles(txtLinkFile.Text, "\*.mp3"))

{

info = new FileInfo(file);

file\_mp3\_temp = info.FullName;

temp = file\_mp3\_temp.Split('\\');

file\_name = temp[temp.Length - 1];

file\_name = file\_name.Substring(0, file\_name.Length - 4);

file\_mp3 = @"C:\Csharp\ZingMp3\Path\_mp3\" + file\_name + ".mp3";

//Copy file

file\_mp3 = Library.File\_Me.CoppyFile(file\_mp3\_temp, file\_mp3);

if (!file\_mp3.Equals(""))

{

string[] files = file\_mp3.Split('\\');

file\_mp3 = @"..\..\..\..\";

for (int i = files.Length - 2; i < files.Length; i++)

{

file\_mp3 += files[i] + "\\";

}

file\_mp3 = file\_mp3.Substring(0, file\_mp3.Length - 1);

}

so.setSing\_name(info.Name);

WMPLib.IWMPMedia media = this.Media.newMedia(info.FullName);

so.setSing\_time((int)media.duration);

so.setSing\_singer("Đang cập nhật");

so.setSing\_content("Đang cập nhật");

so.setSing\_author("Đang cập nhật");

so.setSing\_path\_mp3(file\_mp3);

so.setSing\_category\_id(13);

so.setSing\_created\_date(info.CreationTime.ToString().Split(' ')[0]);

so\_template = sc.getSing(so.getSing\_name());

if (so\_template == null)

{

try

{

sc.addSing(so);

}

catch (Exception ex)

{

MessageBox.Show(so.getSing\_path\_mp3()+"\n" + so.getSing\_id());

}

mes += so.getSing\_name() + " --- Đã được thêm.\n";

}

else

{

mes += so.getSing\_name() + " --- Đã tồn tại.\n";

}

}

}

* Xử lý sửa, xóa và nghe nhạc tại giao diện quản lý bài hát

private void dtgSing\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

var senderGrid = (DataGridView)sender;

int clIndex = e.ColumnIndex;

if (senderGrid.Columns[clIndex] is DataGridViewImageColumn &&

e.RowIndex >= 0)

{

Int32 row = e.RowIndex;

int sing\_id = Int32.Parse(dtgSing.Rows[row].Cells[1].Value.ToString());

SingObject so = new SingObject();

SingControl sc = new SingControl();

so = sc.getSing(sing\_id);

//-----------------------Run or pause a sing-------------------------

if (clIndex == 8)

{

try

{

if (rIndex != row)

{

string RelativePath = so.getSing\_path\_mp3();

string AbsolutePath = Path.GetFullPath(Path.Combine(System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetEntryAssembly().Location), RelativePath));

MediaZing.URL = AbsolutePath;

senderGrid.Rows[rIndex].Cells[8].Value = Image.FromFile(@"..\..\..\..\Img\Run.png");

rIndex = row;

//player.Play();

senderGrid.Rows[row].Cells[8].Value = Image.FromFile(@"..\..\..\..\Img\Pause.png");

running = true;

}

else

{

if (!running)

{

rIndex = row;

string RelativePath = so.getSing\_path\_mp3();

string AbsolutePath = Path.GetFullPath(Path.Combine(System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetEntryAssembly().Location), RelativePath));

MediaZing.URL = AbsolutePath;

//player.Play();

senderGrid.Rows[row].Cells[8].Value = Image.FromFile(@"..\..\..\..\Img\Pause.png");

running = true;

}

else

{

MediaZing.URL = "";

//player.Stop();

senderGrid.Rows[row].Cells[8].Value = Image.FromFile(@"..\..\..\..\Img\Run.png");

running = false;

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

//------------------------------edit a sing-----------------------

else if (clIndex == 9)

{

SingAE edit = new SingAE();

edit.SingEdit(so);

edit.ShowDialog();

}

//------------------------------del a sing------------------------

else if (clIndex == 10)

{

DialogResult rs = MessageBox.Show("Bạn có muốn xóa bài hát này", "Cảnh bảo", MessageBoxButtons.OKCancel);

if (rs == DialogResult.OK)

{

MediaZing.URL = "";

sc.delSing(so);

this.SingView\_Load(sender, e);

}

}

}

}

* Kĩ thuật phân trang

//phân trang theo tổng bài hát và mảng tên bài hát

public void phanTrang(int count, string[] q)

{

tongPage = count / 5;

tongPage++;

lePage = count % 5;

if (lePage == 0)

{

if (hienTaiPage == tongPage)

{

hienTaiPage--;

}

tongPage--;

}

if (count > 0)

{

loadSearch(tongPage, lePage, hienTaiPage, q);

}

else

{

hienGrSearch(0, 0, 0, 0, 0);

}

}

public void loadSearch(int tongPage, int lePage, int hienTaiPage, string[] q)

{

if (lePage == 0)

{

hienGrSearch(1, 1, 1, 1, 1);

layString(1, 1, 1, 1, 1, q);

}

else

{

if (hienTaiPage < tongPage)

{

hienGrSearch(1, 1, 1, 1, 1);

layString(1, 1, 1, 1, 1, q);

}

else

{

if (lePage == 1)

{

hienGrSearch(1, 0, 0, 0, 0);

layString(1, 0, 0, 0, 0, q);

}

if (lePage == 2)

{

hienGrSearch(1, 1, 0, 0, 0);

layString(1, 1, 0, 0, 0, q);

}

if (lePage == 3)

{

hienGrSearch(1, 1, 1, 0, 0);

layString(1, 1, 1, 0, 0, q);

}

if (lePage == 4)

{

hienGrSearch(1, 1, 1, 1, 0);

layString(1, 1, 1, 1, 0, q);

}

if (lePage == 5)

{

hienGrSearch(1, 1, 1, 1, 1);

layString(1, 1, 1, 1, 1, q);

}

if (lePage == 6)

{

hienGrSearch(1, 1, 1, 1, 1);

layString(1, 1, 1, 1, 1, q);

}

if (lePage == 7)

{

hienGrSearch(1, 1, 1, 1, 1);

layString(1, 1, 1, 1, 1, q);

}

}

}

int[] idi = { Convert.ToInt32(labelSearch1.Text), Convert.ToInt32(labelSearch2.Text), Convert.ToInt32(labelSearch3.Text), Convert.ToInt32(labelSearch4.Text), Convert.ToInt32(labelSearch5.Text)};

//hàm xem đã tải chưa nếu tải rồi thì đổi màu

xuLyTaiRoi(idi);

}

* Kĩ thuật tìm kiếm gần đúng bài hát theo tên hoặc ca sĩ

//search sing

public DataTable getSings(string search)

{

String sql = "SELECT \* FROM tblsing LEFT JOIN tblcategory ON sing\_category\_id = category\_id ";

sql += "where sing\_name like '%" + search + "%' OR sing\_singer like '%" + search + "%' OR sing\_author like '%" + search

+ "%' OR sing\_content like '%" + search + "%' OR category\_name like '%" + search + "%'";

DataTable dt = new DataTable();

dt = Connection.getTable(sql);

return dt;

}

* Kĩ thuật hiện bài hát nổi bật được nghe nhiều lên trên

//sắp xếp bài hát nổi bật được nghe nhiều lên trên và giảm dần

public void trend(string[] s, int[] a ,int [] id)

{

for (int i = 0; i < a.Length; i++)

{

int max = a[i];

for (int j = i + 1; j < a.Length; j++)

{

if (max < a[j])

{

string tem = s[i];

s[i] = s[j];

s[j] = tem;

int tem2 = a[i];

a[i] = a[j];

a[j] = tem2;

int tem3 = id[i];

id[i] = id[j];

id[j] = tem3;

}

}

}

}

* Kĩ thuật sử lý khi người dùng ấn play bài hát khác trong khi đang nghe nhạc

private void playMusic(PictureBox pb,Label la)

{

SingObject singObject = new SingObject();

singObject = singControl.getSing(Convert.ToInt32(la.Text));

try

{

if (!running)

{

string RelativePath = singObject.getSing\_path\_mp3();

string AbsolutePath = Path.GetFullPath(Path.Combine(System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetEntryAssembly().Location), RelativePath));

MediaZing.URL = AbsolutePath;

pb.Image = Image.FromFile(@"..\..\..\..\Img\Pause.png");

running = true;

singObject.setSing\_listened(singObject.getSing\_listened() + 1);

singControl.editSing(singObject);

this.PictureBox = pb;

txtSingContent.Text = singObject.getSing\_content();

trangDangPhat = hienTaiPage;

nhan = title.Text;

}

else

{

if (pb != PictureBox)

{

MediaZing.URL = "";

PictureBox.Image = Image.FromFile(@"..\..\..\..\Img\Run.png");

running = false;

playMusic( pb,la);

}

else

{

MediaZing.URL = "";

pb.Image = Image.FromFile(@"..\..\..\..\Img\Run.png");

running = false;

}

}

//xuLyHienThiPlay();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

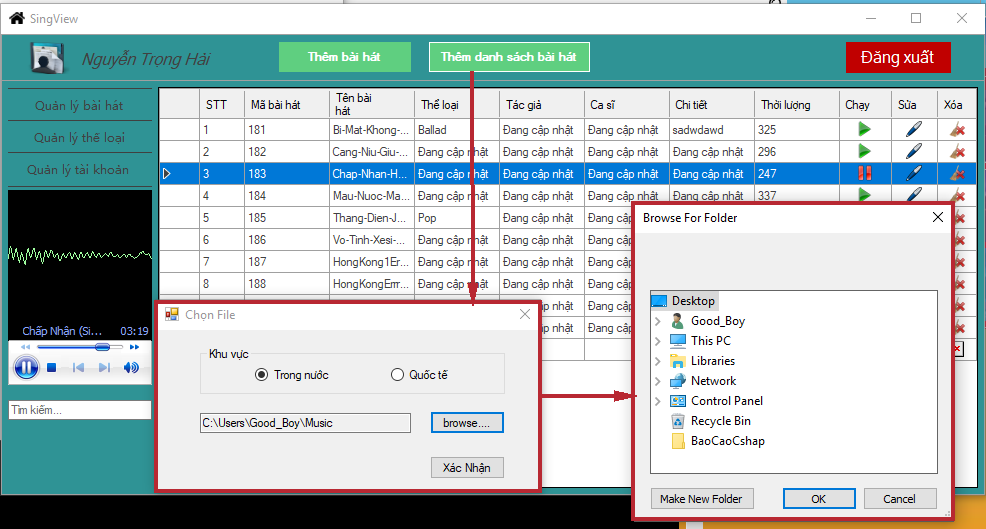
}

## Một số hình ảnh minh họa

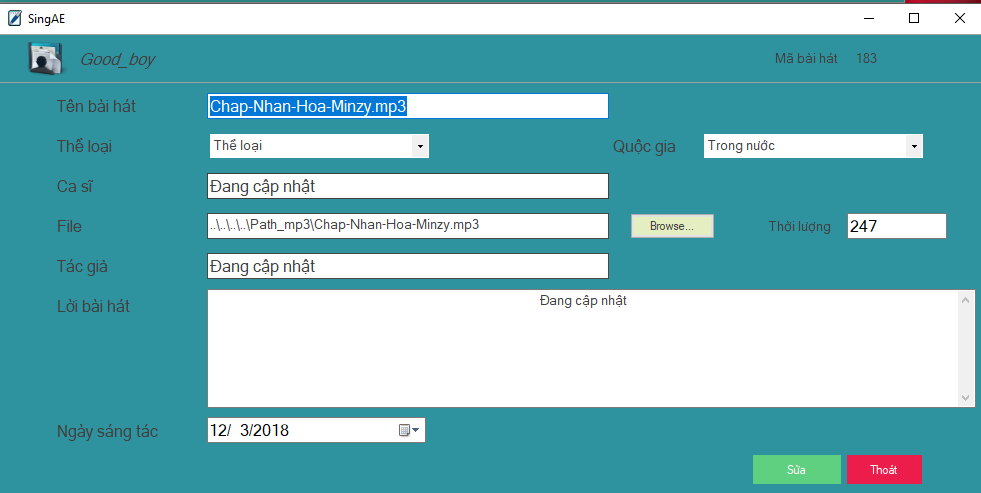
* Cho phép admin thưởng thức nhạc.



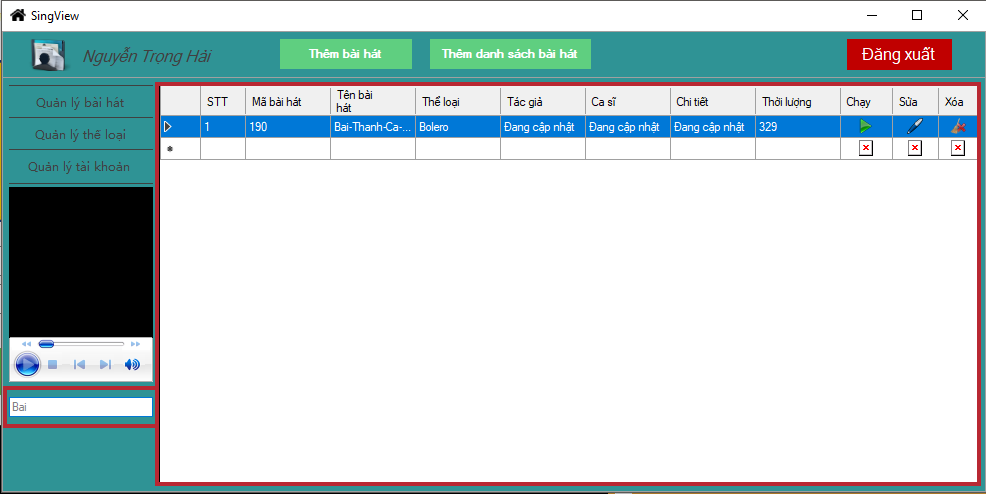
* Cho phép thêm danh sách bài hát, hỗ trợ admin trong việc quản lý



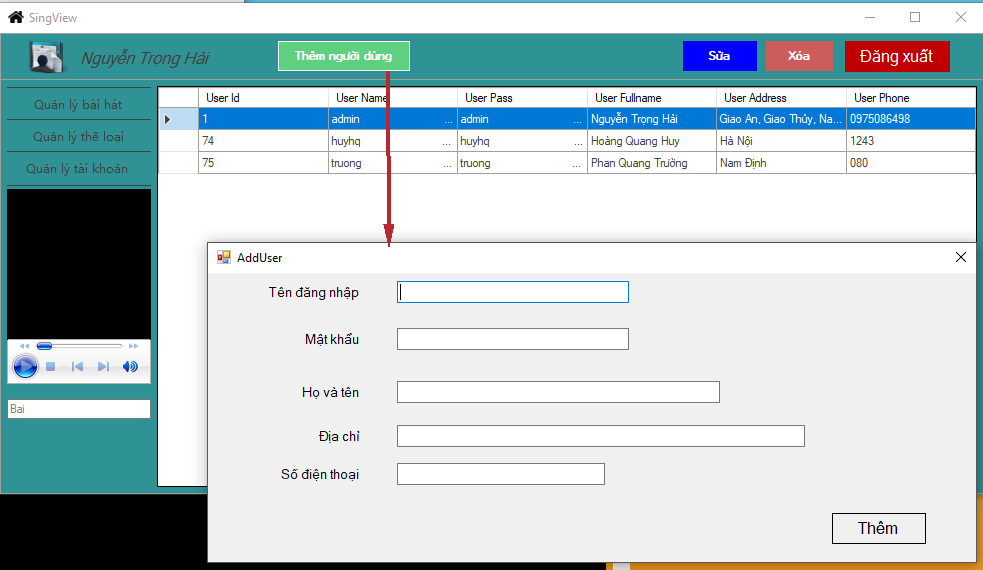
* Giao diện thêm và sửa bài hát.



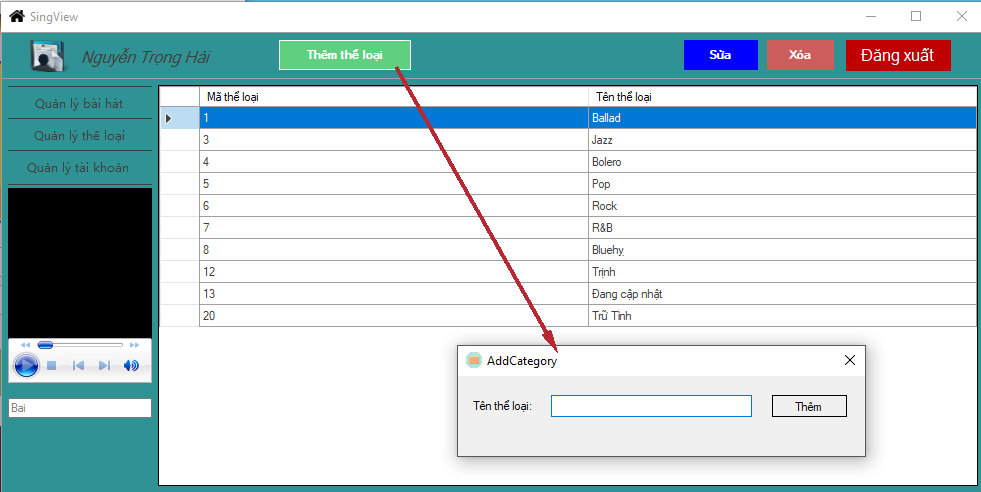
* Tìm kiếm bài hát.



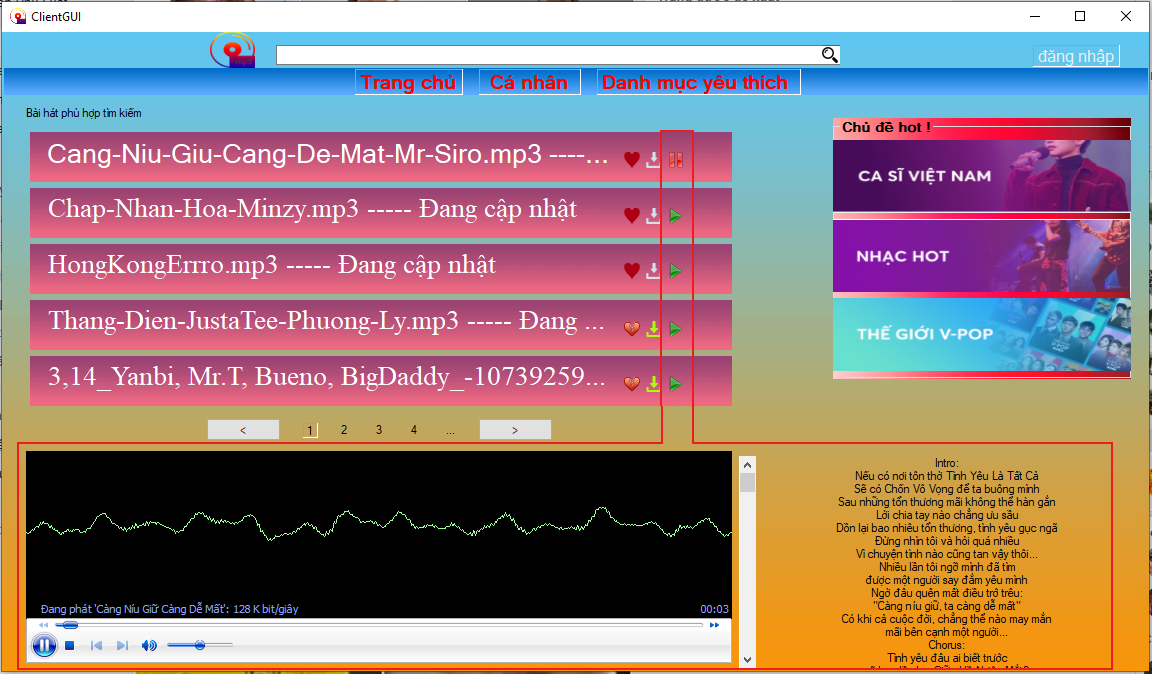
* Quản lý tài khoản.



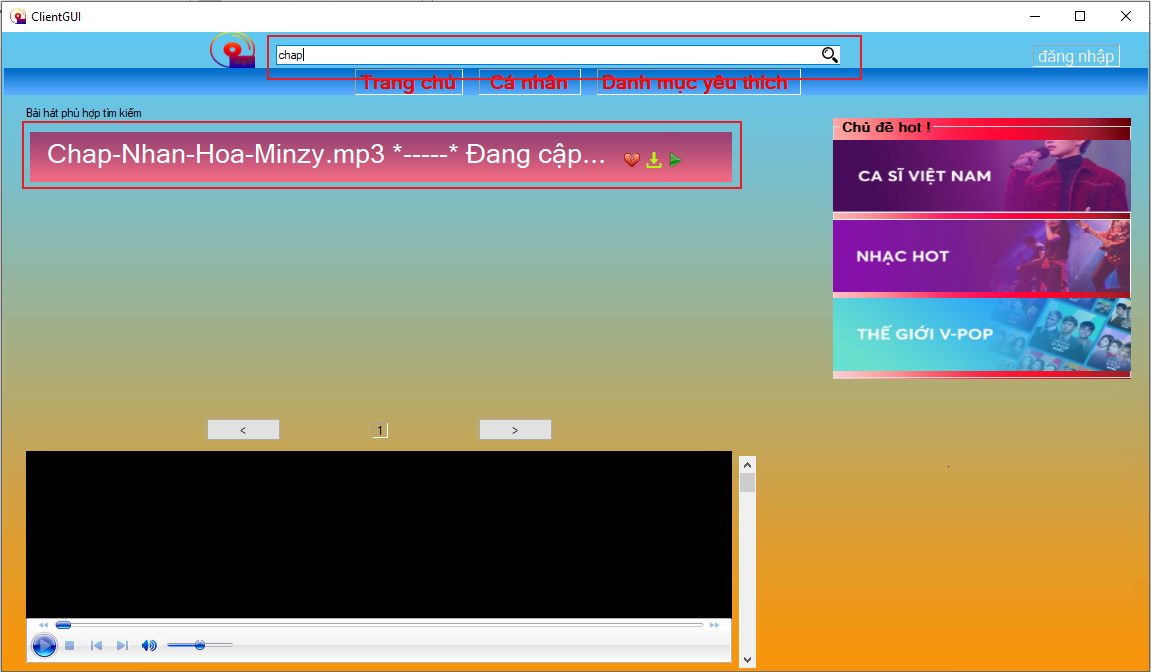
* Quản lý thể loại.



* Giao diện phát nhạc của người dùng



* Giao diện tìm kiếm bài hát



* Giao diện bài hát nổi bật



**Kết luận**

Trong quá trình xây dựng và phát triển phần mềm, nhóm đã gặp rất nhiều khó khăn trong việc thiết kế giao diện, cấu trúc.

Tuy nhiên hệ thống cũng áp ứng được những chức năng cơ bản của một phần mêm tìm kiếm và quản lý bài hát.

Chương trình được kiểm tra rà soát sao cho việc xuất hiện lỗi là tối thiểu, ít nhất. Đảm bảo chương trình hoạt động một cách mượt mà nhất. Tối ưu giao diện và đơn giản hóa giúp cho việc sử dụng dễ dàng hơn.

Đã hoàn thành được những mục tiêu đặt ra.

Hệ thống đi theo mô hình MVC nên dễ update và bảo trì

Chưa đạt được: Giao diện cổ điển, chưa bắt mắt, linh hoạt.

Kết thúc việc xây dựng phần mềm, nhóm đã đúc kết được nhiều kinh nghiệm, học hỏi thêm được nhiều kiến thức trong việc phát triển phần mềm bằng ngôn ngữ C#, góp phần hoàn thiện kỹ năng, tuy duy trong xây dựng phần mềm và củng cố kiến thức đã học.

Nhóm xin được cảm ơn sự hỗ trợ, chỉ dẫn của Thầy **Phạm Văn Hà** trong quá trình xây dựng phần mềm!

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tài liệu tham khảo: Giáo trình: Lập trình Windows.