

BÁO CÁO THỰC HÀNH

Môn học: Lập trình mạng căn bản

Buổi báo cáo: Cuối Kỳ

Tên chủ đề: Tên bài thực hành

GVHD: Lê Minh Khánh Hội

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT106.N22.MMCL.1

STT	Họ và tên	MSSV	Email
1	Lê Hoàng Khánh	21522205	21522205@gm.uit.edu.vn
2	Nguyễn Hữu Hưng	21522118	21522118@gm.uit.edu.vn
3	Đỗ Thế Danh	21520685	21520685@gm.uit.edu.vn
4	Nguyễn Minh Nhật	21522418	21522418@gm.uit.edu.vn

2. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	
Link Video thực hiện (nếu có)	
Link git repo chứa source code	https://github.com/NT106-N22-Group-01/MusicStreamingAPI https://github.com/NT106-N22-Group-01/MusicStreamingAPI
Ý kiến (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá	8.5

A. Tổng quan

1. Giới thiệu

"Music Streaming" là một phần mềm cho phép người dùng nghe nhạc trực tuyến. Ngoài ra phần mềm còn có chức năng tìm kiếm ca sĩ, album và bài hát sẽ hiển thị kết quả tìm kiếm trong vòng vài giây. Phần mềm còn hiển thị lượt xem người dùng, điều này giúp người dùng có thể biết được mức độ phổ biến của bài hát hoặc album đó trên nền tảng.

2. Cơ sở lý thuyết

a) TCP

Để đảm bảo độ tin cậy trong các ứng dụng mạng, ta dùng một giao thức có kết nối: TCP (Transport Control Protocol).

Trên Internet chủ yếu dùng Telnet, HTTP, SMTP, POP3... Để lập trình theo giao thức TCP, .NET cung cấp hai lớp có tên là TCPClient và TCPListener.

Trong đồ án, TCP được sử dụng trong việc giao tiếp giữa Client và Server.

b) API

API (Application Programming Interface): Một tập hợp các quy tắc, giao thức, công cụ.

Dùng để xây dựng các ứng dụng phần mềm và cho phép các ứng dụng khác nhau giao tiếp và tương tác với nhau.

Cho phép người dùng truy cập, sử dụng tài nguyên, dịch vụ từ một ứng dụng hoặc từ hệ thống khác.

Dùng để truy cập dịch vụ web, cơ sở dữ liệu, thư viện mã nguồn mở, phần mềm hệ thống,...

Được cung cấp dưới dạng các giao thức web, như REST, SOAP, JSON-RPC.

c) HTTP

HTTP (Hypertext Transfer Protocol): một giao thức truyền tải siêu văn bản được sử dụng để truyền thông tin trên mạng.

HTTP được sử dụng để tương tác giữa Client và Server, cho phép truyền tải và truy cập các tài liệu và dữ liệu trên World Wide Web.

HTTP hoạt động dựa trên mô hình yêu cầu/phản hồi (request/response), trong đó máy khách gửi một yêu cầu HTTP tới máy chủ, và máy chủ phản hồi bằng một thông điệp phản hồi HTTP chứa thông tin được yêu cầu.

HTTP là một giao thức cơ bản trong việc truyền tải thông tin trên web và đóng vai trò quan trọng trong việc giao tiếp giữa các ứng dụng web và máy chủ.

Trong đồ án, HTTP được sử dụng để gửi các yêu cầu HTTP đến API và nhận phản hồi từ API.

d) IP

IP (Internet Protocol): Một giao thức trên Internet.

Là một địa chỉ đơn nhất.

Đang được sử dụng để nhận diện và liên lạc với nhau trên mạng máy tính bằng cách sử dụng giao thức Internet.

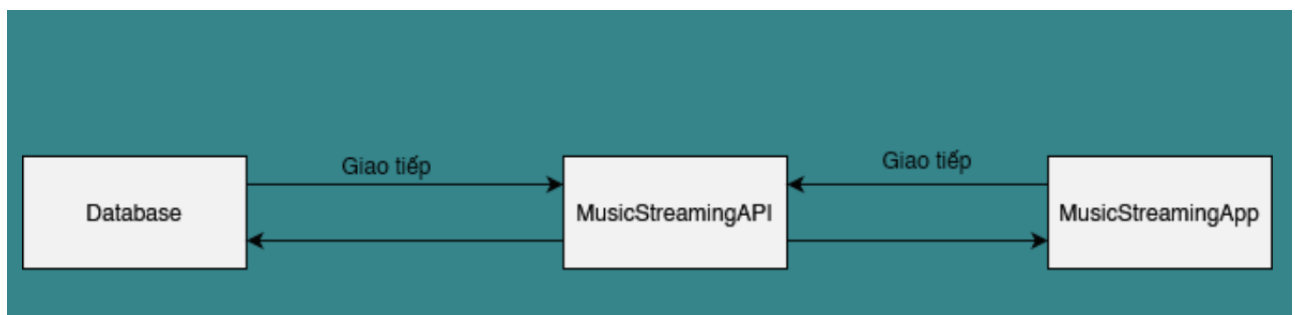
IP có hai loại chính: IPv4 và IPv6.

IPv4 là phiên bản truyền thống được sử dụng rộng rãi, còn IPv6 là phiên bản mới hơn và hỗ trợ cho việc kết nối một số lượng lớn hơn các thiết bị trên mạng.

Trong đồ án, IP được sử dụng để giao tiếp giữa Server và Client.

B. Phân tích và thiết kế hệ thống

1. Kiến trúc tổng quan



Hình 1. Sơ đồ kết nối

2. Network Stack

Ứng dụng sử dụng một số tầng trong mô hình OSI như sau:

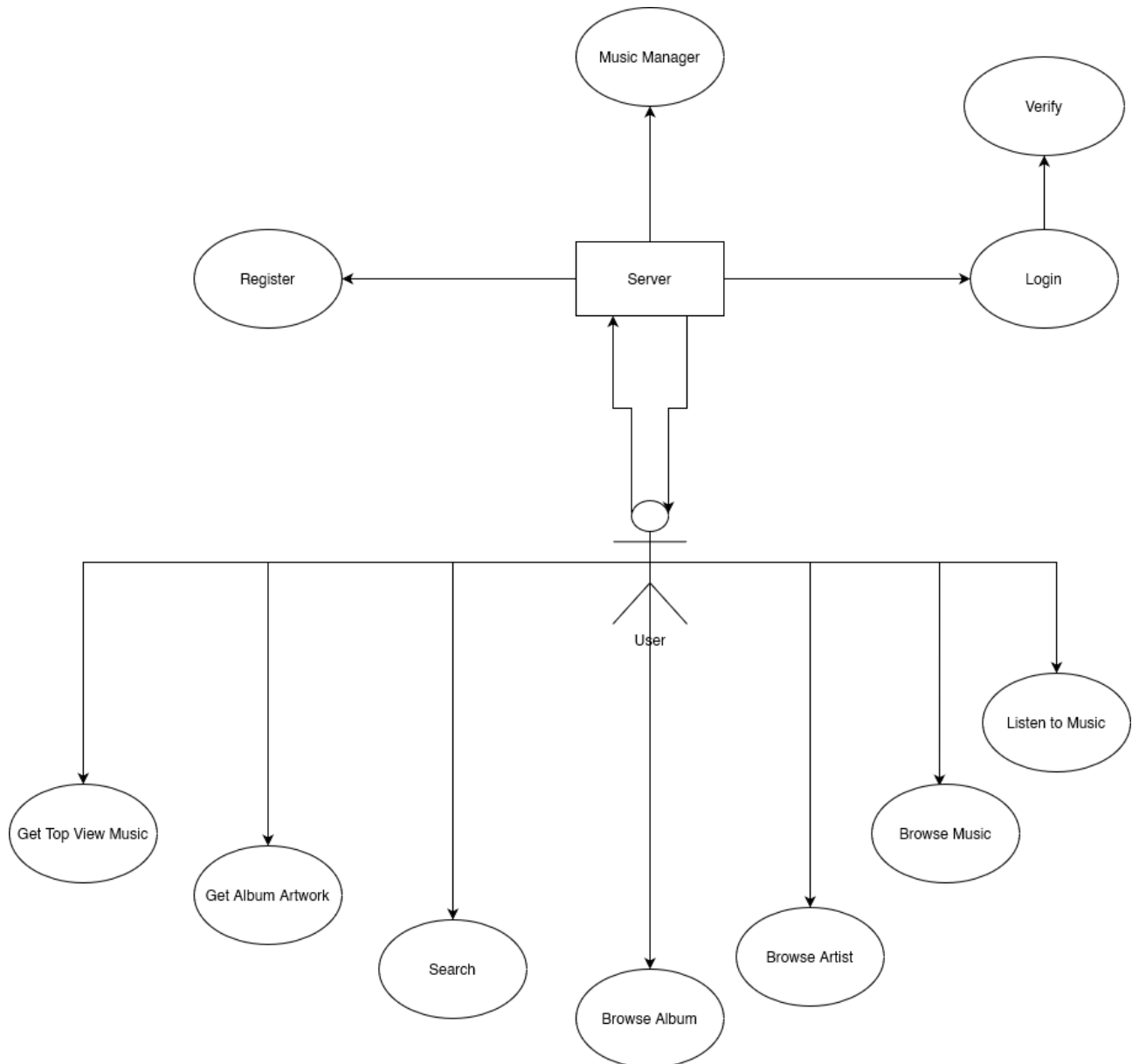
Tầng Application: Sử dụng giao thức HTTP để gửi các yêu cầu HTTP đến API và nhận phản hồi từ API.

Tầng Presentation: Định dạng và mã hóa dữ liệu để đảm bảo tính toàn vẹn và bảo mật của thông tin truyền qua mạng.

Tầng Transport: Sử dụng giao thức TCP để truyền dữ liệu đáng tin cậy giữa Server và Client.

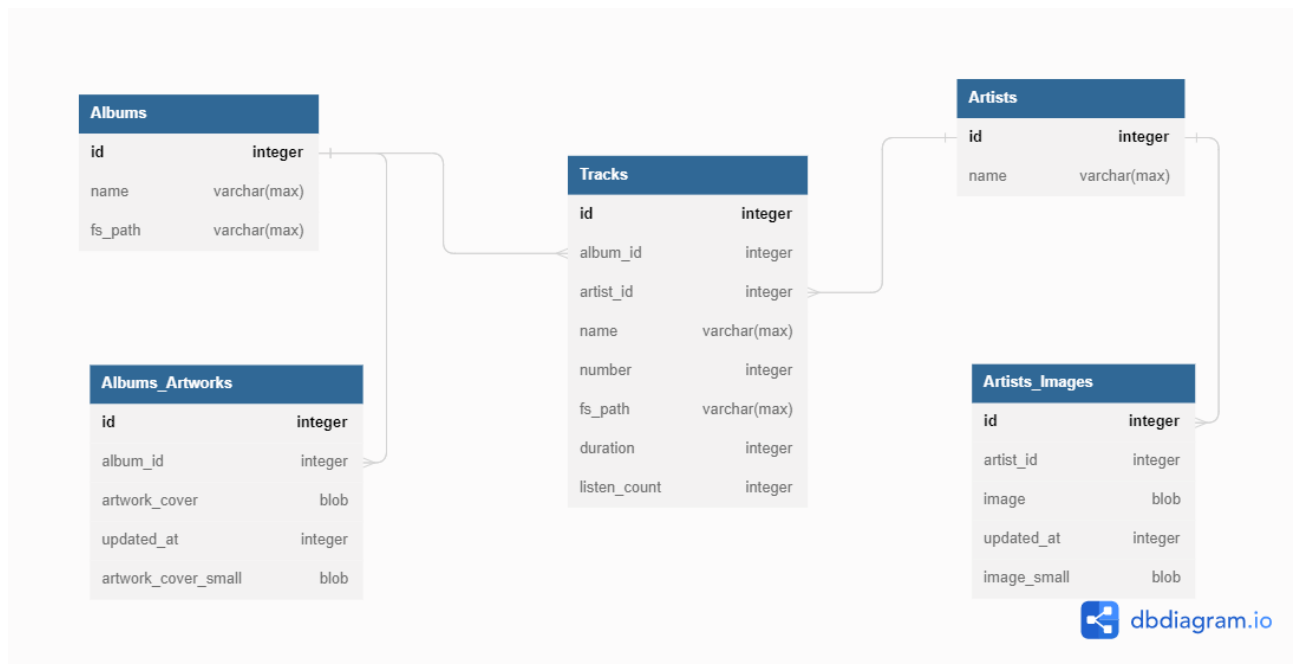
Tầng Network: Sử dụng giao thức IP để định tuyến gói tin và quản lý việc kết nối, giao tiếp giữa Server và Client.

3. Sơ đồ Use Case



Hình 2. Use Case hệ thống

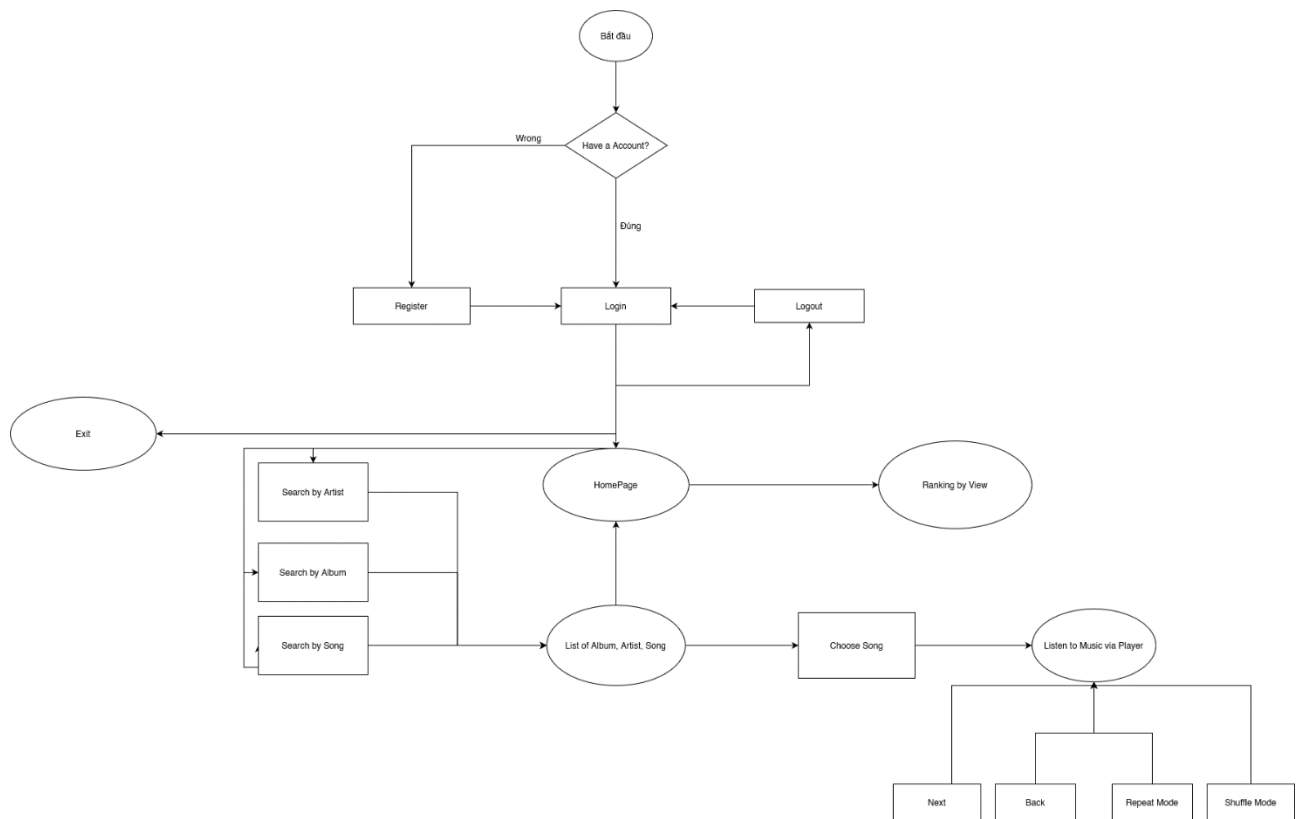
4. Sơ đồ Database



Hình 3. Sơ đồ Database

5. Luồng ứng dụng.

a) Phía Client



Hình 4. Flow Client

b) Phía Server.

Authentication

Khi máy chủ được mở, chkhông cần xác thực các yêu cầu đến API. Các cài đặt được bảo vệ bằng username và password yêu cầu người dùng xác thực các yêu cầu khi sử dụng API. Để làm điều này, hỗ trợ các phương thức sau đây:

Bearer token trong HTTP header `Authorization` (như mô tả trong [RFC 6750](#)):

Authorization: Bearer {token}

Authentication tokens được nhận thông qua `/v1/login/token/` hoặc `/v1/register/token/` endpoint được mô tả dưới đây. Sử dụng tokens là phương pháp được ưu tiên vì nó không tiết lộ username và password ở mỗi request.



Search

Thực hiện tìm kiếm thông qua endpoint sau:

GET /v1/search/?q={query}

Endpoint sẽ trả về mảng JSON chứa thông tin các track. Mỗi đối tượng trong JSON đại diện cho một track duy nhất khớp với **query** được cung cấp. Ví dụ:

```
{
  {
    "id": 22,
    "artist_id": 6,
    "artist": "Aimer",
    "album_id": 3,
    "album": "春はゆく / marie",
    "title": "春はゆく",
    "track": 1,
    "format": "flac",
    "duration": 304000
  },
  {
    "id": 23,
    "artist_id": 6,
    "artist": "Aimer",
    "album_id": 3,
    "album": "春はゆく / marie",
    "title": "marie",
    "track": 2,
    "format": "flac",
    "duration": 307000
  }
}
```

```
}  
)
```

Browse

Cách để duyệt toàn bộ bộ sưu tập là thông qua gọi API **Browse**. Nó cho phép lấy các album hoặc nghệ sĩ trong một trình tự được sắp xếp và phân trang.

GET

/v1/browse/(?by=artist|album)(&perpage={number})(&page={number})(&order-by=id|name)(&order=desc|asc)

JSON trả về chứa dữ liệu cho trang hiện tại, số trang trong tất cả các trang cho phương thức duyệt hiện tại và các URL của trang tiếp theo hoặc trang trước đó.

```
{  
  "data": (  
    {  
      "album_id": 1,  
      "album": "102112",  
      "artist": "Cá Hối Hoang"  
    },  
    {  
      "album_id": 2,  
      "album": "Avid / Hands Up to the Sky",  
      "artist": "Various Artists"  
    },  
    {  
      "album_id": 3,  
      "album": "春はゆく / marie",  
      "artist": "Aimer"  
    }  
  ),  
}
```



```
"next": "",  
"previous": "",  
"pages_count": 1  
}
```

Hiện tại, có hai giá trị có thể cho tham số **by**. Do đó, có hai loại data có thể được trả về: "**artist**" (nghệ sĩ) và "**album**" (đây là giá trị **mặc định**).

by=artist

kết quả sẽ có các giá trị như sau:

```
{  
  "artist": "Jefferson Airplane",  
  "artist_id": 73  
}
```

by=album

kết quả sẽ có các giá trị như sau:

```
{  
  "album": "Battlefield Vietnam",  
  "artist": "Jefferson Airplane",  
  "album_id": 2  
}
```

Các tham số bổ sung:

per-page: điều khiển số lượng mục sẽ có trong trường *data* cho từng trang cụ thể. Giá trị **mặc định là 10**.

page: dữ liệu được tạo sẽ là cho trang này. **Giá trị mặc định là 1**.

order-by: điều khiển cách kết quả sẽ được sắp xếp. Giá trị *id* có nghĩa là sắp xếp sẽ được thực hiện theo ID của album hoặc nghệ sĩ, tùy thuộc vào đối số *by*. Tương tự, điều này cũng áp dụng cho giá trị *name*. **Mặc định là name**.

order: điều khiển xem thứ tự sẽ tăng dần (giá trị *asc*) hay giảm dần (giá trị *desc*). **Mặc định là asc**.

Phát nhạc

```
GET /v1/file/{trackID}
```

Endpoint này sẽ trả về tập tin nhạc. `trackID` của một bài hát có thể được tìm thấy bằng cuộc gọi API tìm kiếm.

Lượt nghe

```
GET /v1/file/{trackID}/count
```

Endpoint này sẽ trả về số lượt nghe của track có ID là `trackID`.

Tải Album

```
GET /v1/album/{albumID}
```

Endpoint này sẽ trả về tập tin nén trong đó chứa toàn bộ nhạc của album này.

Album Artwork

Get Artwork

```
GET /v1/album/{albumID}/artwork
```

Trả về bitmap image là artwork cho album này nếu có sẵn. Tìm kiếm artwork hoạt động như sau: thư mục của album sẽ được quét để tìm bất kỳ hình ảnh nào (tệp png/jpeg/gif/tiff), và nếu có bất kỳ hình ảnh nào trông giống hình minh họa, nó sẽ được hiển thị. Nếu thất bại, có thể config để tìm kiếm trong [MusicBrainz Cover Art Archive](https://musicbrainz.org/doc/Cover_Art_Archive). Mặc định, không có external calls được thực hiện.

Artist Image

Get Artist Image

```
GET /v1/artist/{artistID}/image
```

Trả về bitmap image là hình đại diện cho artist nếu có sẵn. Quá trình tìm kiếm hình ảnh hoạt động như sau: nếu hình ảnh của nghệ sĩ được tìm thấy trong cơ sở dữ liệu, nó sẽ được sử dụng. Trong trường hợp không tìm thấy, Server sẽ sử dụng các API MusicBrainz và Discogs để lấy hình ảnh.

Login

```
POST /v1/login/token/
```

```
{
  "username": "your-username",
  "password": "your-password"
}
```

Nếu username và password đúng Endpoint sẽ trả về token để thêm vào header phục vụ cho việc xác thực.

Register

```
POST /v1/register/token/  
{  
  "username": "your-username",  
  "password": "your-password"  
}
```

Endpoint này sẽ tạo tài khoản cho người dùng sau khi tạo thành công
Endpoint sẽ trả về token để thêm vào header phục vụ cho việc xác thực.

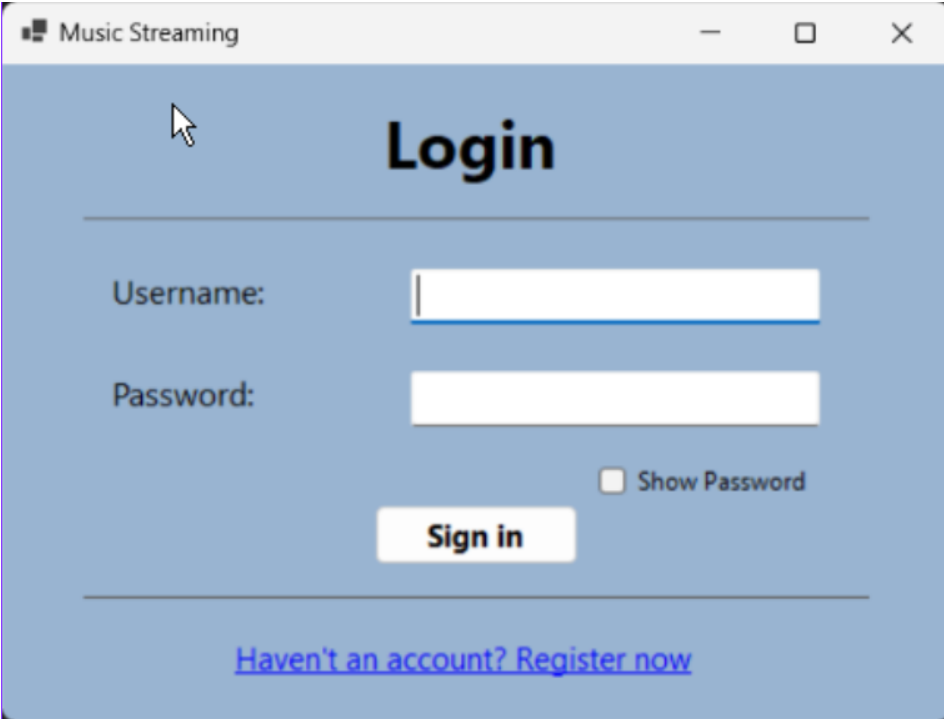
C. Hiện thực hệ thống.

1. Công nghệ sử dụng.

Đối với server sử dụng **Go** để xây dựng **RestFul API** và **SQLite3** dành cho Database.

Đối với client sử dụng **.NET 6** cùng với **Winform** để sử dụng **UI** và sử dụng thêm **mpv** làm thư viện dùng để làm **player** giúp cho việc streaming cũng như phát nhạc.

2. Login Form



The screenshot shows a Windows Form titled "Music Streaming". The form has a blue background and a white border. At the top, the title bar shows the application name and standard window controls. The main content area has a large "Login" title in bold black text. Below the title, there are two input fields: "Username:" and "Password:". The "Password:" field has a "Show Password" checkbox to its right. Below the input fields is a "Sign in" button. At the bottom of the form, there is a link that says "Haven't an account? Register now".

Hình 5. Login Form

```
private async void loginButton_Click(object sender, EventArgs e)
{
    var username = txtUsername.Text;
    var password = txtPassword.Text;

    if (!Common.ValidateInput(username, password))
        return;

    var loginRequest = new
    {
        username,
        password
    };

    var json = JsonConvert.SerializeObject(loginRequest);
    var content = new StringContent(json, Encoding.UTF8, "application/json");

    try
    {
        var response = await httpClient.PostAsync($"{Config.Config.ApiBaseUrl}/v1/login/token/", content);
        if (response.IsSuccessStatusCode)
        {
            var responseContent = await response.Content.ReadAsStringAsync();
            var responseObj = JsonConvert.DeserializeObject<dynamic>(responseContent);

            string tokenValue = responseObj.token;

            var MainForm = new Player.Main(tokenValue);
            this.Hide();
            MainForm.Show();
        }
        else
        {
            MessageBox.Show("Login failed. Please check your credentials.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"An error occurred: {ex.Message}");
    }
}
```

Hình 6. Code Login.

3. Register Form

```
2 references
public static bool ValidateInput(string username, string password)
{
    if (string.IsNullOrEmpty(username))
    {
        MessageBox.Show("Please enter a username.");
        return false;
    }

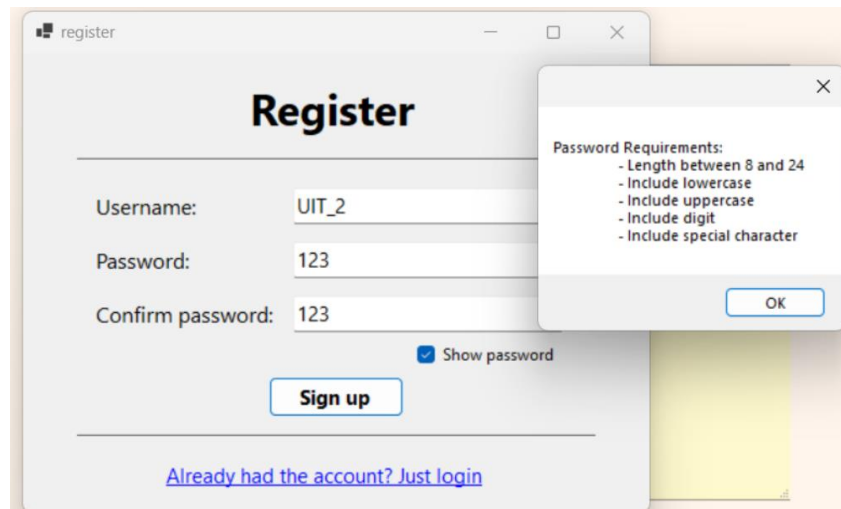
    if (username.Contains(" "))
    {
        MessageBox.Show("Username cannot contain spaces.");
        return false;
    }

    if (string.IsNullOrEmpty(password))
    {
        MessageBox.Show("Please enter a password.");
        return false;
    }

    return true;
}

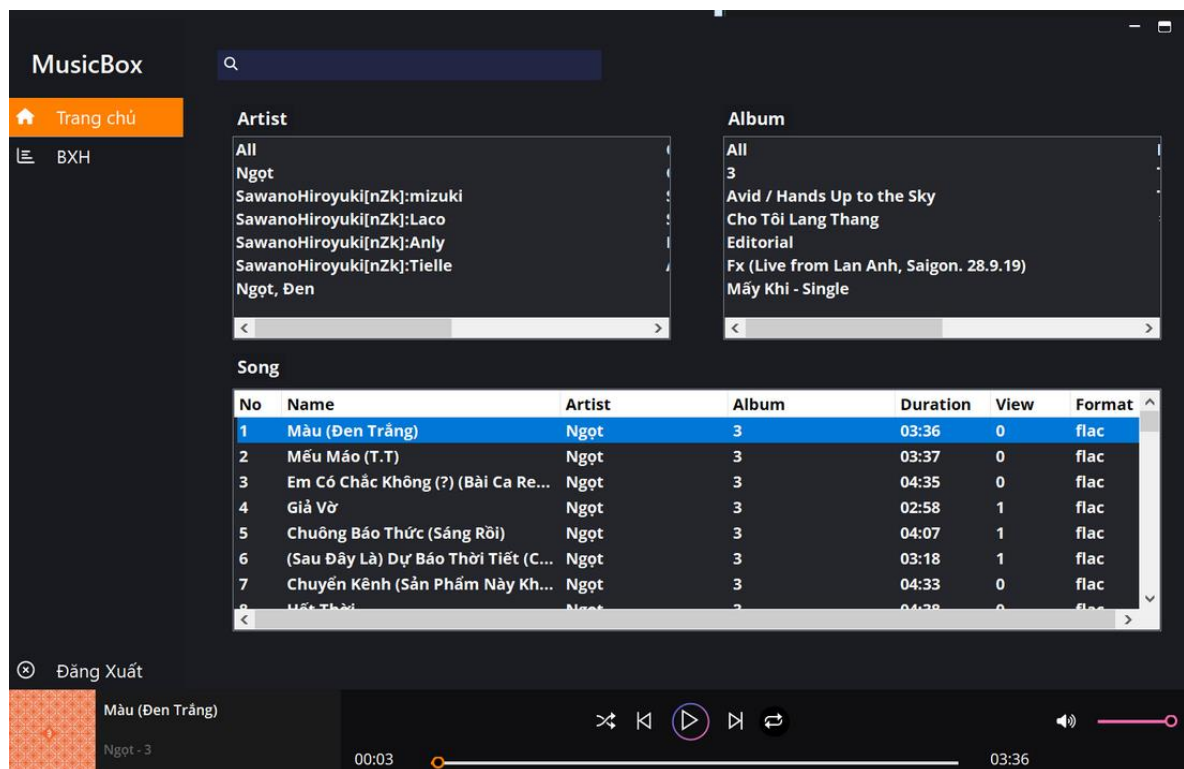
1 reference
public static bool IsValidPassword(string password)
{
    string pattern = @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[!@#$%^&*~`|,./;:~]).{8,24}$";
    if (Regex.IsMatch(password, pattern))
    {
        return true;
    }
    else
    {
        MessageBox.Show("Password Requirements:\r\n\t" +
            "- Length between 8 and 24\r\n\t" +
            "- Include lowercase\r\n\t" +
            "- Include uppercase\r\n\t" +
            "- Include digit\r\n\t" +
            "- Include special character\r\n");
        return false;
    }
}
```

Hình 7. Validate Password



Hình 8. Register Form

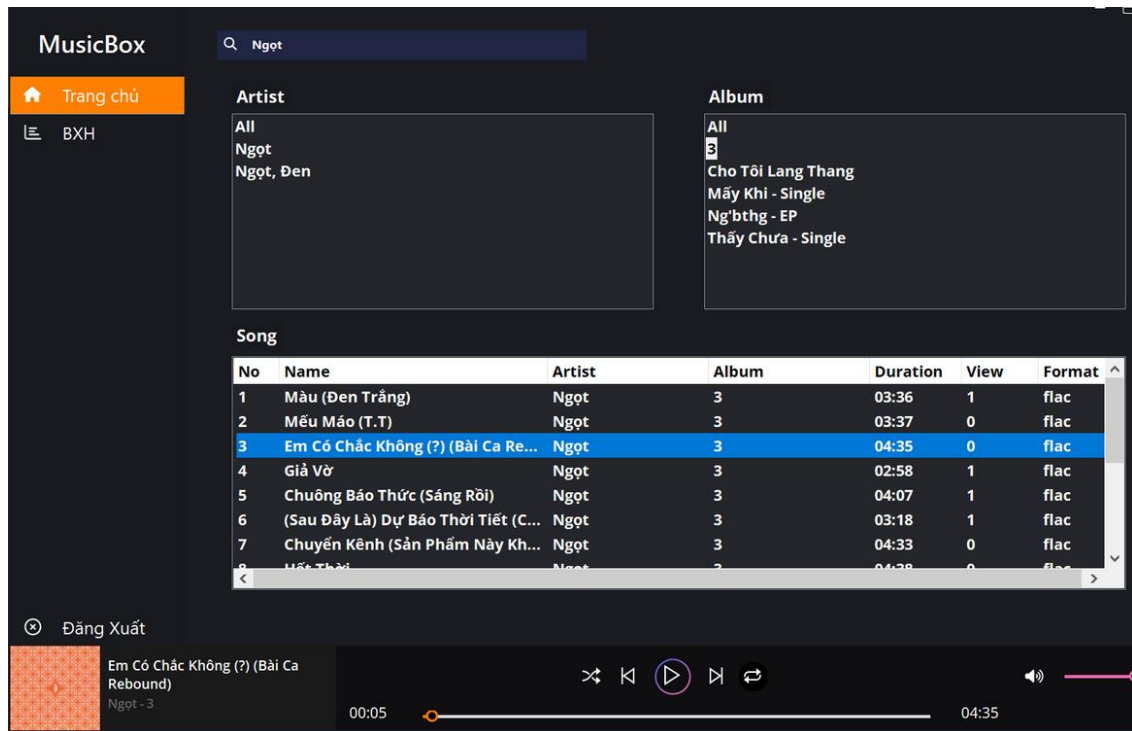
4. Ứng dụng nghe nhạc.



Hình 9. Ứng dụng nghe nhạc.

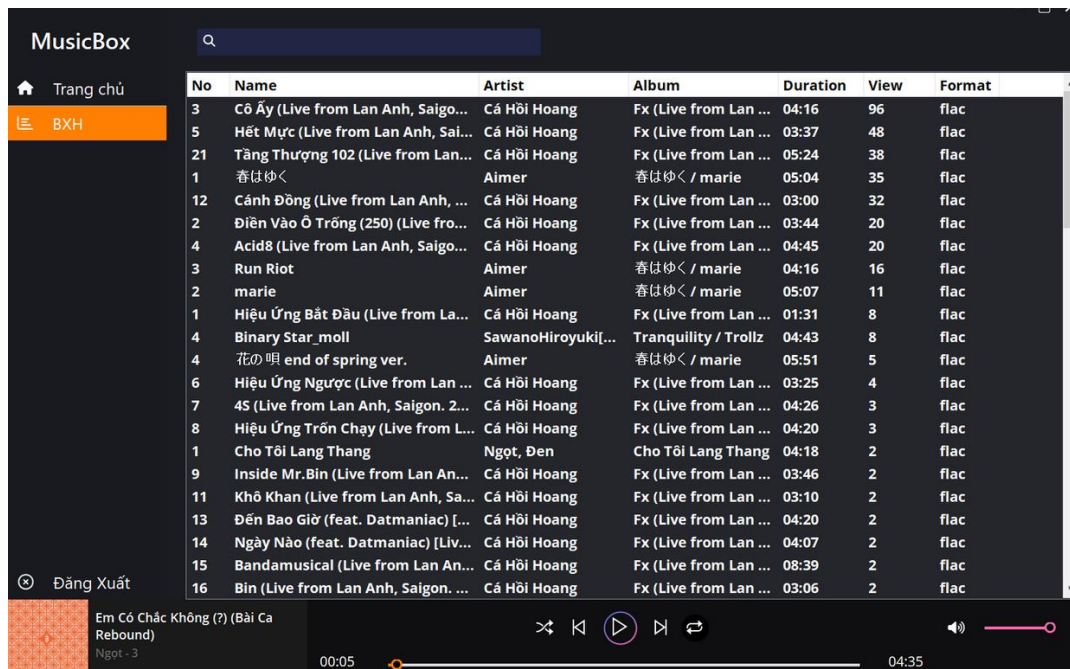
Trở về danh sách Artist, Album và Song có thể filter theo Artist và Album bằng cách bấm vào từng mục của ListView của 2 mục trên. Ấn 2 lần vào nhạc muốn nghe để phát. Sau khi phát nhạc ta có thể điều khiển player: Tạm dừng phát, tiếp tục lại, phát bài kế tiếp, lùi lại, điều chỉnh âm lượng, tua nhạc. Ngoài ra còn có các chế độ phát như lặp lại và ngẫu nhiên. Ngoài

ra ta có thể kiểm thông nhạc qua thành tìm kiếm và nhập thông tin liên quan như tên bài, nghệ sĩ, album.



Hình 10. Kết quả tìm kiếm nhạc.

Cùng theo đó ta có thể xem bảng xếp hạng lượt nghe của toàn bộ bài hát bằng cách ấn vào tab BXH.



Hình 11. BXH bài hát.

5. Tổng Kết

a) Kết luận

Trong quá trình xây dựng và phát triển sản phẩm, nhóm chúng em còn gặp nhiều khó khăn, hạn chế về kiến thức cũng như các công cụ, nhưng nhóm đã cố gắng hết mình để có được một sản phẩm hoàn thiện nhất có thể.

b) Hướng phát triển

Đây chỉ là ứng dụng nghe nhạc cơ bản nên chắc chắn sẽ còn nhiều tính năng cần phát triển.

Tính năng dành cho nghệ sĩ có thể upload nhạc lên dễ dàng hơn.

Tính năng phân loại nhạc theo từng thể loại.

Tính năng like, share nhạc.

Tính năng tạo playlist.

Phát triển thêm các ứng dụng nền web, điện thoại...

Có các tung tab profile cho các user và các user khác có thể flow user profile nào đó.

Thêm nhiều tính năng để người dùng sử dụng dễ dàng hơn.

Nghiên cứu áp dụng các công nghệ hiện đại hơn vào ứng dụng nhằm tối ưu hiệu năng, code, dễ dàng trong việc nâng cấp sau này.

Trên đây là một số hướng phát triển mà nhóm đề ra đối với ứng dụng nghe nhạc của nhóm.

Bảng phân công công việc

STT	Tên	MSSV	Nội dung công việc phụ trách	Mức độ hoàn thành	Tự đánh giá (theo thang điểm 10)
1	Lê Hoàng Khánh	21522205	Xây dựng API và xử lý logic Client	Tốt	8.5
2	Nguyễn Hữu Hưng	21522118	Phát triển UI cho Client.	Khá	7
3	Đỗ Thế Danh	21520685	Phát triển Database, phần bảo mật đăng nhập tạo tài khoản.	Tốt	8
4	Nguyễn Minh Nhật	21522418	Phát triển Client xử lý Logic cho Client. Deploy server lên Internet	Tốt	7.5

**Nhóm chúng em xin chân thành cảm ơn cô đã hướng dẫn tận tình
để có hoàn thành đồ án một cách hoàn thiện nhất.**

HẾT