



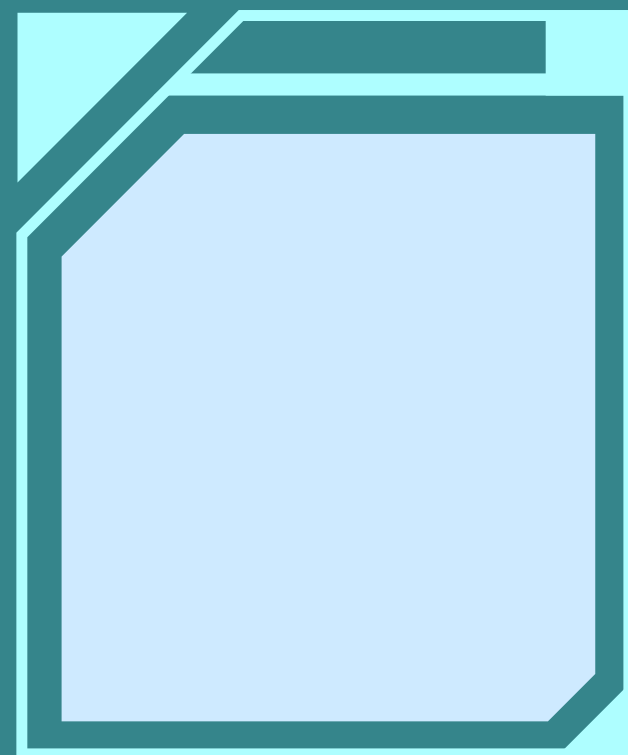
MUSIC STREAMING

FINAL PROJECT

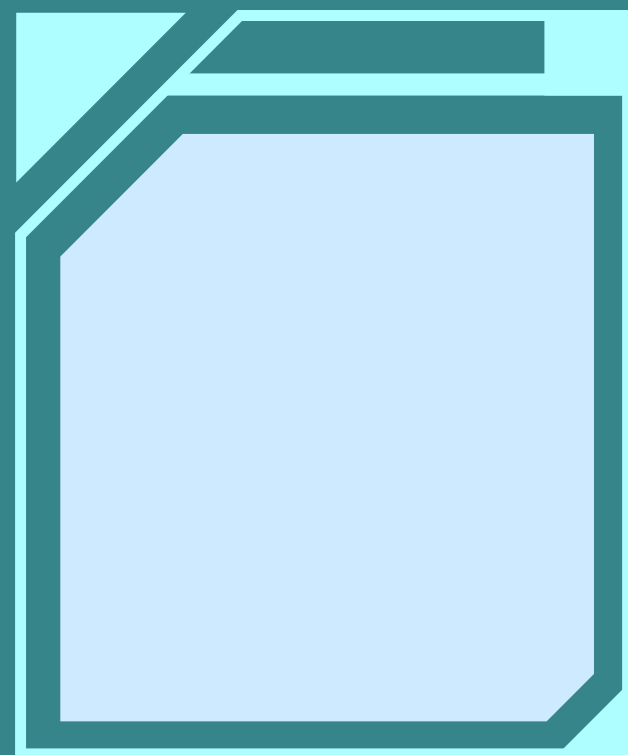
GROUP 01

GVHD: LÊ MINH KHÁNH HỘI

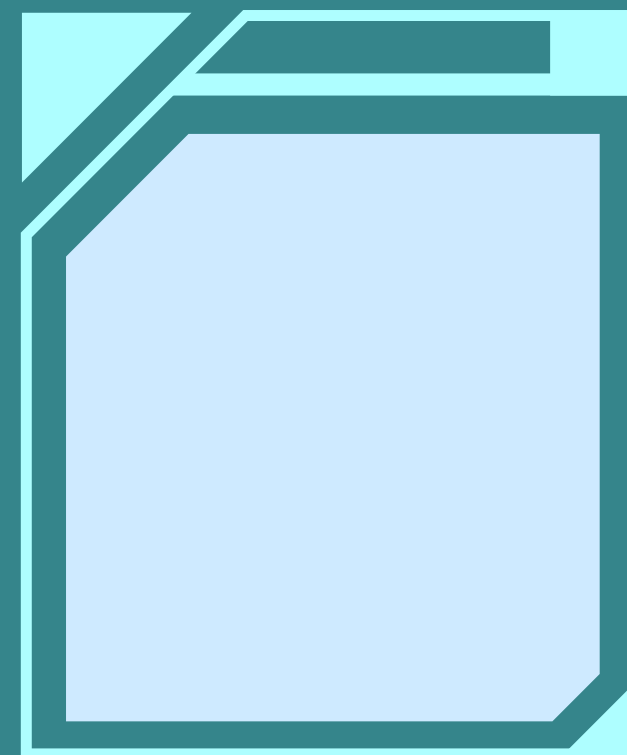
NHÓM 1



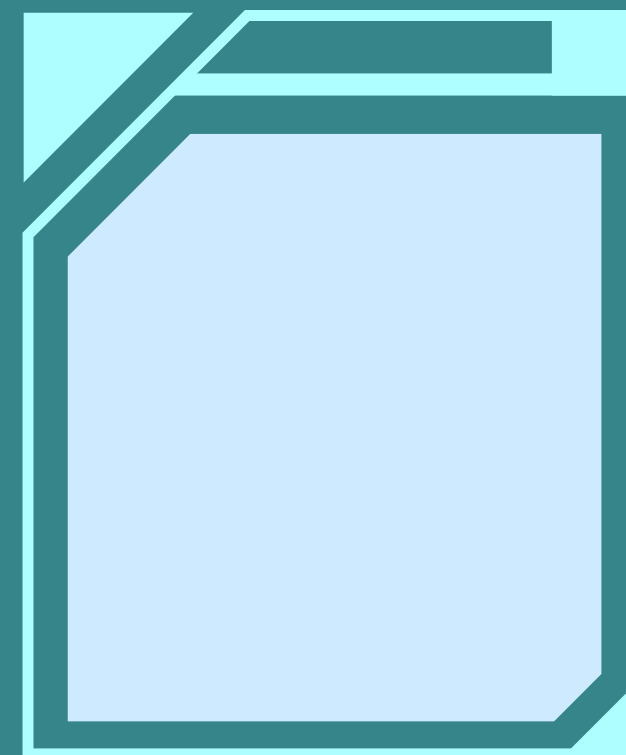
Lê Hoàng Khánh



Đỗ Thế Danh



Nguyễn Minh Nhật



Nguyễn Hữu Hưng

MỤC LỤC

1. Giới thiệu

2. Cơ sở Lý
thuyết

3. Kiến trúc
tổng quan

4. Use case

5. Luồng của
ứng dụng

6. Hiện thực

7. Demo

8. Kết luận

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 lí thuyết

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.

2. Cơ sở lí thuyết

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.

2.Cơ sở lí thuyết

HTTP:

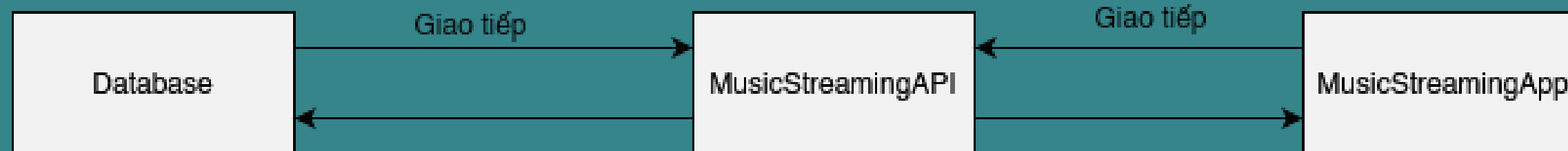
- HTTP (Hypertext Transfer Protocol): một giao thức truyền tải siêu văn bản được sử dụng để truyền thông tin trên mạng.
- HTTP được sử dụng để tương tác giữa Client và Server, cho phép truyền tải và truy cập các tài liệu và dữ liệu trên World Wide Web.
- HTTP hoạt động dựa trên mô hình yêu cầu/phản hồi (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.

2. Cơ sở lí thuyết

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.

3. Kiến trúc tổng quan

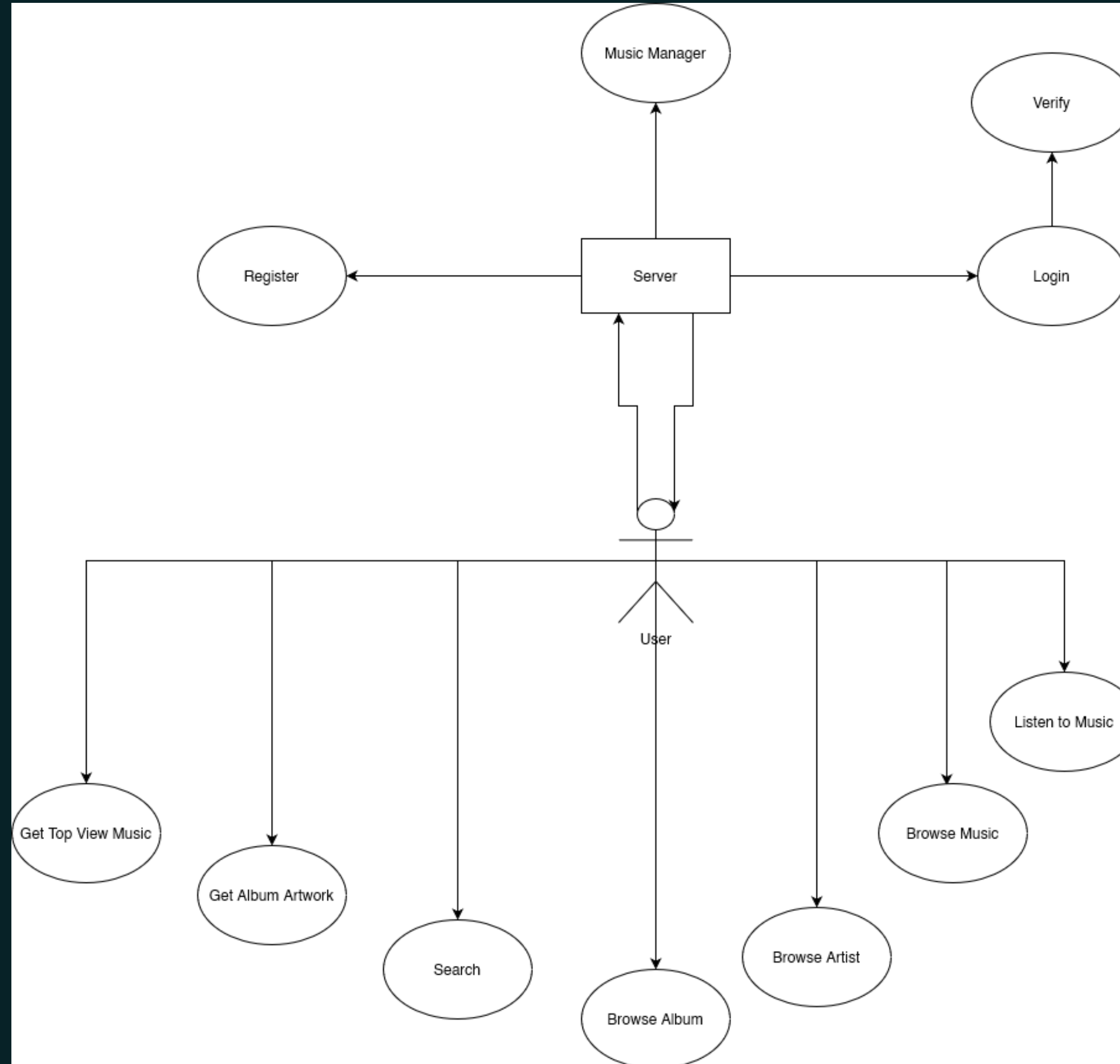


4. Network Stack

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

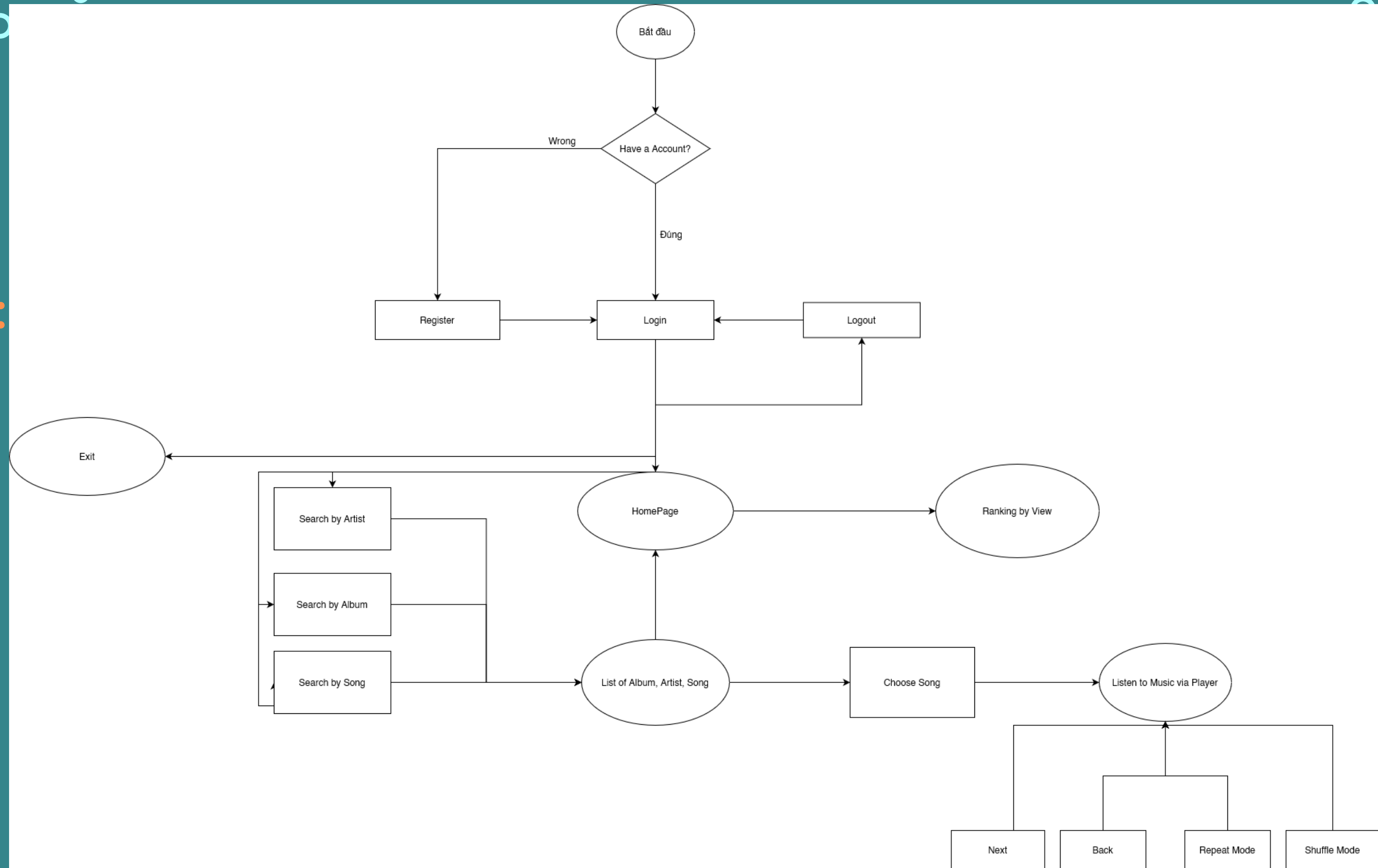
- Tầng Application: Sử dụng giao thức HTTP để gửi các yêu cầu HTTP đến API và nhận phản hồi từ API.
- Tầng Presentation: Định dạng và mã hóa dữ liệu để đảm bảo tính toàn vẹn và bảo mật của thông tin truyền qua mạng.
- Tầng Transport: Sử dụng giao thức TCP để truyền dữ liệu đáng tin cậy giữa Server và 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.

5. Sơ đồ Use Case



6. FlowChart

• Phía Client:



6. Flowchart

- **Phía Server:**

- GET /v1/search/?q={query}
- GET /v1/browse/[?by=artist|album][&per-page={number}][&page={number}][&order-by=id|name][&order=desc|asc]
- GET /v1/file/{trackID}
- GET /v1/album/{albumID}/artwork
- GET /v1/artist/{artistID}/image
- POST /v1/login/token/
 - {
 - "username": "your-username",
 - "password": "your-password"
 - }
- POST /v1/register/token/
 - {
 - "username": "your-username",
 - "password": "your-password"
 - }

6. Database



8. Hiện thực hệ thống

Công nghệ sử dụng



Server

.NET

Client



Windows Forms



8. Hiện thực hệ thống

Validation

```
public static class Common
{
    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;
        }

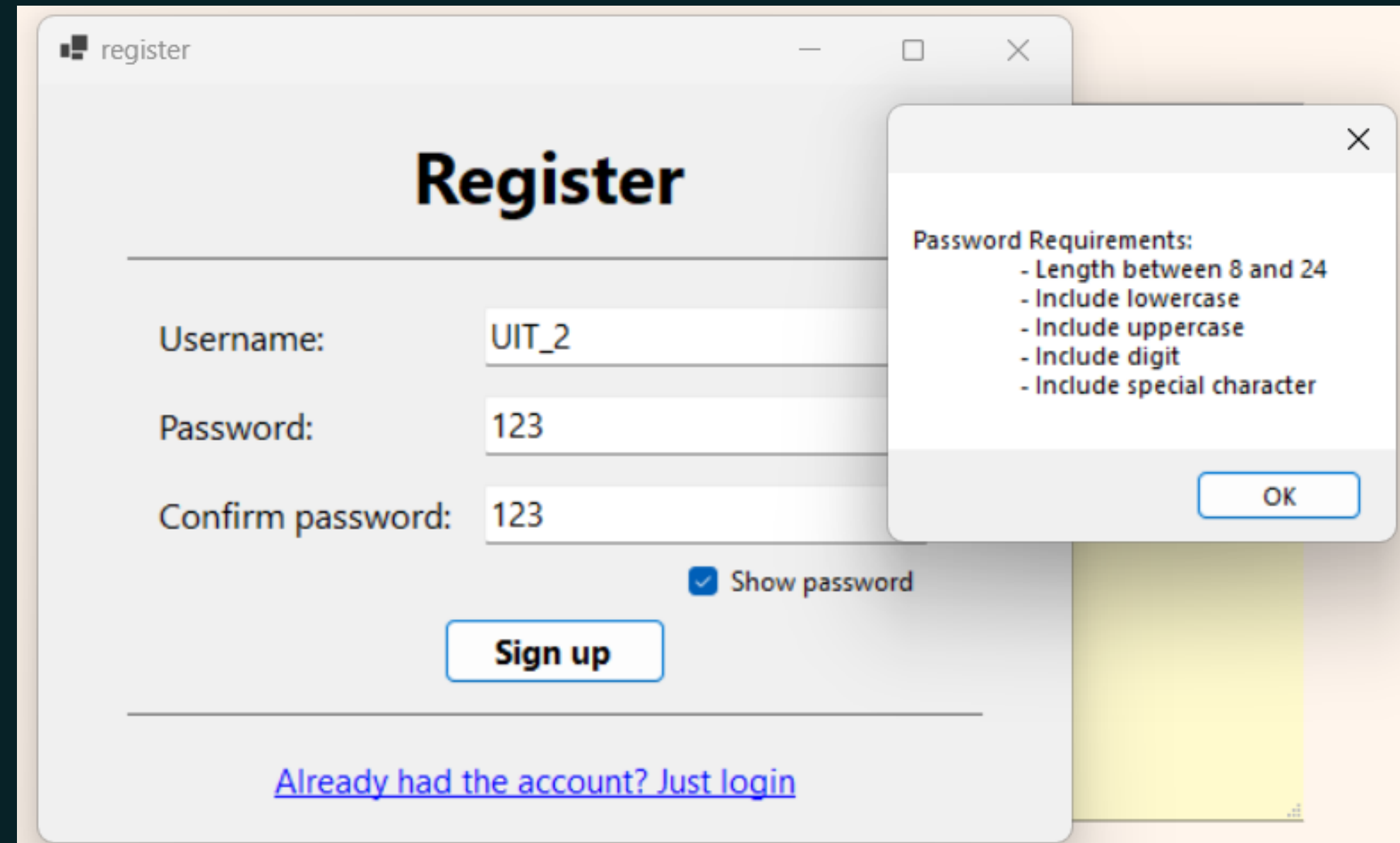
        return true;
    }

    1 reference
    public static bool IsValidPassword(string password)
    {
        string pattern = @"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[\W_]).{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;
        }
    }
}
```

8. Hiện thực hệ thống

Phía Client

Register Form:



The image shows a web browser window titled "register" displaying a "Register" form. The form has three input fields: "Username:" with the value "UIT_2", "Password:" with the value "123", and "Confirm password:" with the value "123". Below the password field is a checkbox labeled "Show password" which is checked. A "Sign up" button is located below the form fields. At the bottom of the form, there is a link that says "Already had the account? Just login". A modal dialog box is open over the form, titled "Password Requirements:", listing the following rules:

- Length between 8 and 24
- Include lowercase
- Include uppercase
- Include digit
- Include special character

An "OK" button is present at the bottom right of the modal dialog.

7. Hiện thực hệ thống

Điều kiện:

- Kiểm tra điều kiện của username và password.
- Kiểm tra password thích hợp và xác thực lại.

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

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

    if (!Common.IsValidPassword(password))
        return;

    if (confirmPassword != password)
    {
        MessageBox.Show("Confirm Password and Password is different");
        return;
    }
}
```

7. Hiện thực hệ thống

Khởi tạo gói tin:

- Tạo JSON Object
- Đóng gói HTTP Request

```
var registerRequest = new
{
    username,
    password
};

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

7. Hiện thực hệ thống

Tiến hành đăng ký:

- Gửi HTTP POST lên Server
- Sử dụng token được phản hồi để sử dụng ứng dụng

```
try
{
    var response = await httpClient.PostAsync($"{Config.Config.ApiBaseUrl}/v1/register/token/", content);

    if (response.IsSuccessStatusCode)
    {
        var responseContent = await response.Content.ReadAsStringAsync();

        var responseObj = JsonConvert.DeserializeObject<dynamic>(responseContent);

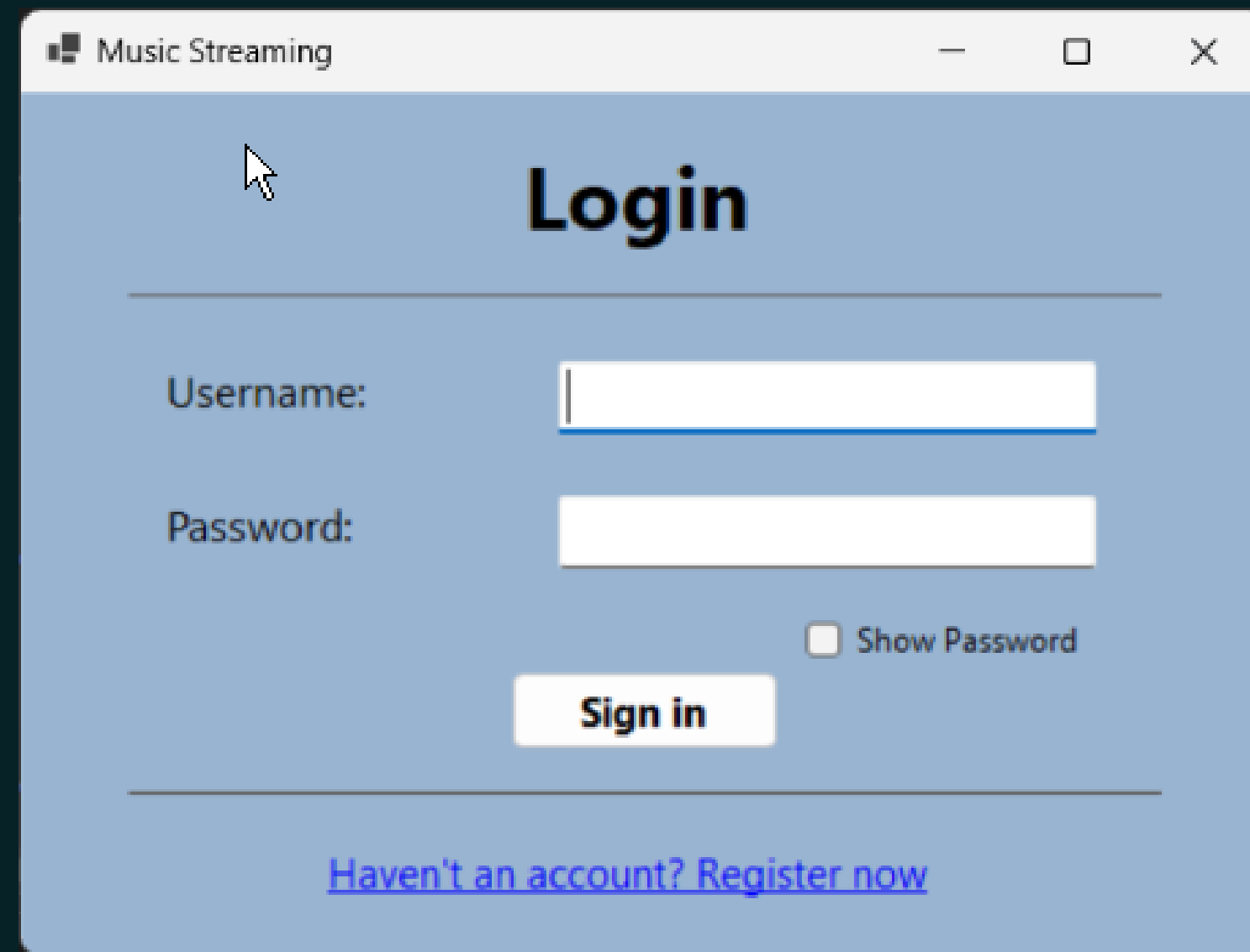
        string tokenValue = responseObj.token;

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


7. Hiện thực hệ thống

Phía Client

Login Form:



The image shows a screenshot of a web application window titled "Music Streaming". The window displays a "Login" form. The form has a light blue background and a white border. It contains the following elements:

- A title "Login" in bold black text, centered at the top of the form area.
- A horizontal line separating the title from the input fields.
- A label "Username:" followed by a white text input field with a blue border.
- A label "Password:" followed by a white password input field with a blue border.
- A checkbox labeled "Show Password" to the right of the password field.
- A "Sign in" button with a blue background and white text, centered below the input fields.
- A horizontal line separating the "Sign in" button from the registration link.
- A link "Haven't an account? Register now" in blue text at the bottom of the form.

7. Hiện thực hệ thống

Đăng nhập:

```
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}");
    }
}
```

7. Hiện thực hệ thống

Phía Server

Register Form:

- Import packages

```
package webserver

import (
    "encoding/json"
    "net/http"
    "time"

    "github.com/gbrlscnchs/jwt/v3"
    "golang.org/x/crypto/bcrypt"
    "gorm.io/gorm"
)
```

7. Hiện thực hệ thống

Phía Server

Register Form:

- Khởi tạo

```
type registerTokenHandler struct {  
    db      *gorm.DB  
    secect  string  
}  
  
func NewRigisterTokenHandler(db *gorm.DB, secect string) http.Handler {  
    return &registerTokenHandler{  
        db:      db,  
        secect:  secect,  
    }  
}
```

7. Hiện thực hệ thống

Phía Server

Register Form:

- Giải mã (Decode) HTTP Request

```
func (register *registerTokenHandler) ServeHTTP(w http.ResponseWriter, r *http.Request) {  
    w.Header().Set("Content-Type", "application/json; charset=utf-8")  
  
    reqBody := struct {  
        User string `json:"username"`  
        Pass string `json:"password"`  
    }{}  
  
    dec := json.NewDecoder(r.Body)  
    if err := dec.Decode(&reqBody); err != nil {  
        respondWithJSONError(  
            w,  
            http.StatusBadRequest,  
            "Error parsing JSON request: %s.",  
            err,  
        )  
        return  
    }  
}
```

7. Hiện thực hệ thống

Phía Server

Register Form:

- Tiến hành băm (hashing)

```
bytes, err := bcrypt.GenerateFromPassword([]byte(reqBody.Pass), 14)
if err != nil {
    respondWithJSONError(
        w,
        http.StatusInternalServerError,
        "Error hash Password: %s.",
        err,
    )
    return
}

// Perform the registration logic here
// You can use the reqBody.User and reqBody.Password to create a new user record in the database
// and generate a registration token if needed

// Example registration logic
// create a new user in the database
user := User{
    Username: reqBody.User,
    Password: string(bytes),
}

if err := register.db.Create(&user).Error; err != nil {
    respondWithJSONError(
        w,
        http.StatusInternalServerError,
        "Error saving user to database: %s.",
        err,
    )
    return
}
```


7. Hiện thực hệ thống

Phía Server

Register Form:

- Tiến hành băm (hashing)

```
now := time.Now()
pl := jwt.Payload{
    IssuedAt:      jwt.NumericDate(now),
    ExpirationTime: jwt.NumericDate(time.Now().Add(rememberMeDuration)),
}

if len(register.sect) == 0 {
    respondWithJSONError(
        w,
        http.StatusInternalServerError,
        "Error generating JWT: secret is empty.",
    )
    return
}
```

7. Hiện thực hệ thống

Phía Server

Register Form:

- Tiến hành băm (hashing)

```
token, err := jwt.Sign(pl, jwt.NewHS256([]byte(register.sect)))
if err != nil {
    respondWithJSONError(
        w,
        http.StatusInternalServerError,
        "Error generating JWT: %s.",
        err,
    )
    return
}

enc := json.NewEncoder(w)
err = enc.Encode(&struct {
    Token string `json:"token"`
}{
    Token: string(token),
})

if err != nil {
    respondWithJSONError(
        w,
        http.StatusInternalServerError,
        "Error writing token response: %s.",
        err,
    )
    return
}
}
```

7. Hiện thực hệ thống

Phía Client

Đăng nhập:

```
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}");
    }
}
```

7. Hiện thực hệ thống

Phía Server

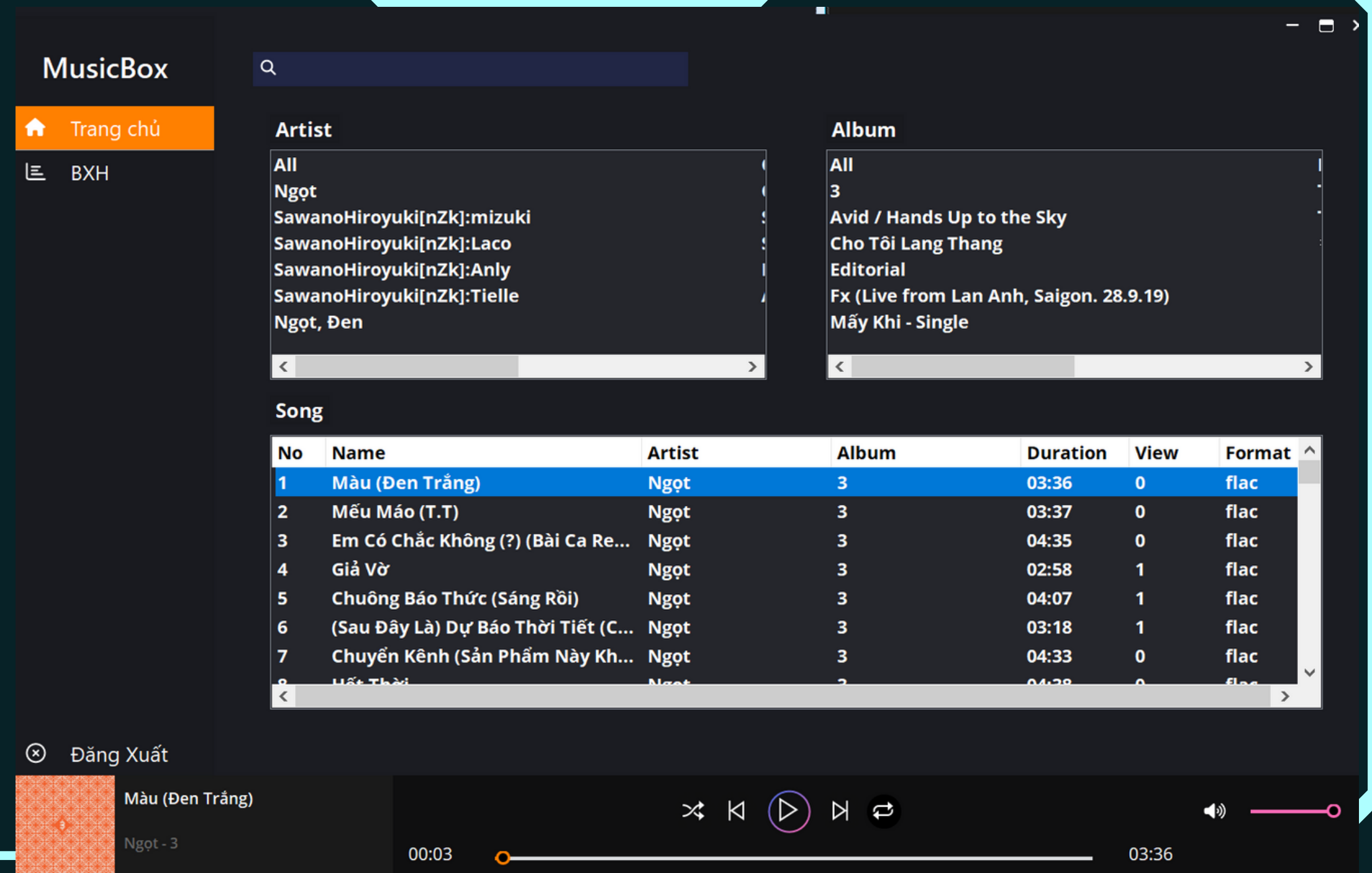
Kiểm tra thông tin:

```
func checkLoginCreds(user, pass string, db *gorm.DB) bool {  
    var userModel User  
    if err := db.Where("username = ?", user).First(&userModel).Error; err != nil {  
        return false  
    }  
  
    err := bcrypt.CompareHashAndPassword([]byte(userModel.Password), []byte(pass))  
    if err != nil {  
        return false  
    }  
  
    return true  
}
```

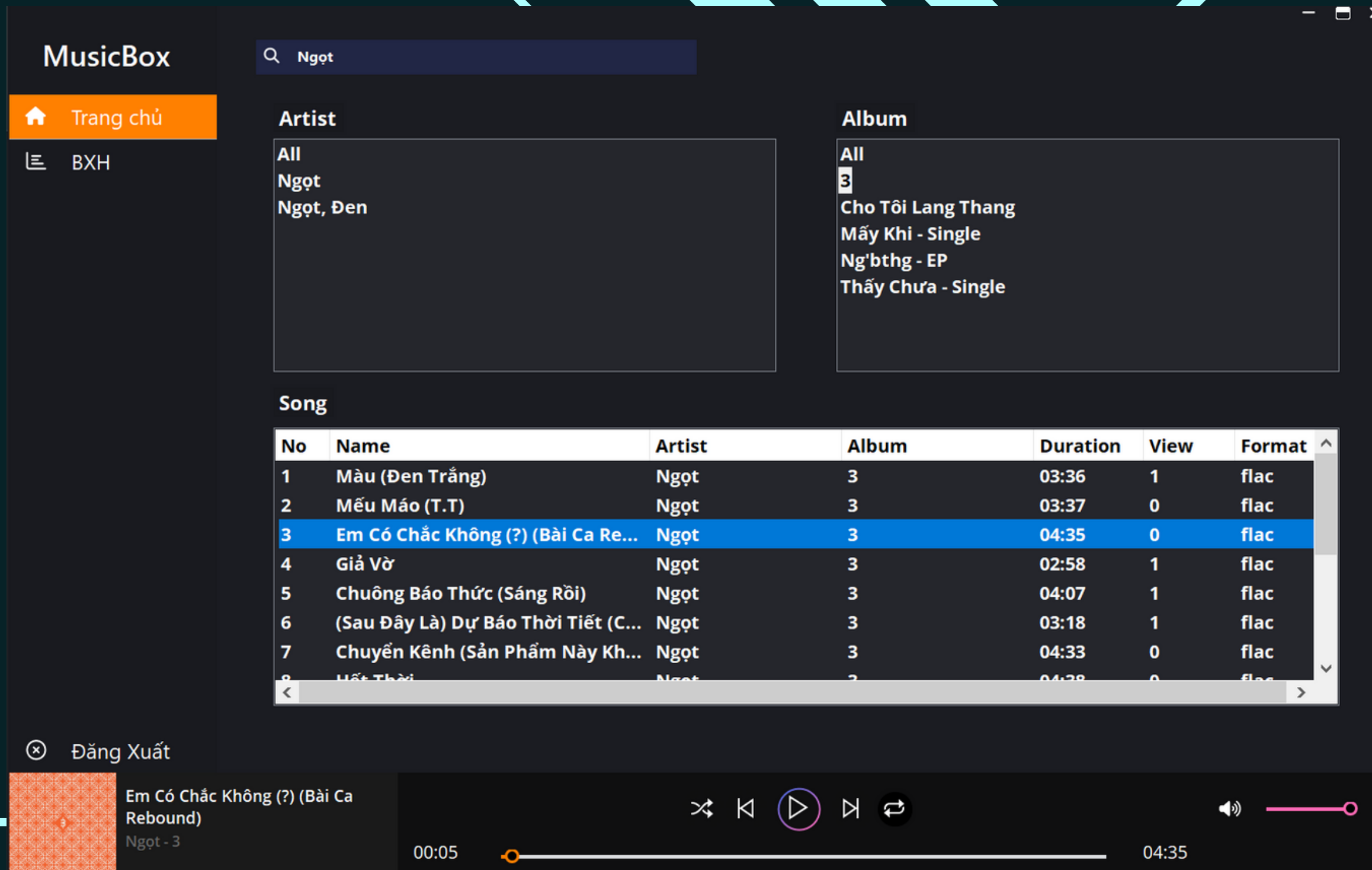
```
if !checkLoginCreds(reqBody.User, reqBody.Pass, h.db) {  
    respondWithJSONError(w, http.StatusUnauthorized, wrongLoginText)  
    return  
}
```

7. Hiện thực hệ thống

Giao diện sau khi đăng nhập:



7. Hiện thực hệ thống



Kết quả được trả về khi tìm kiếm và giao diện nghe nhạc

7. Hiện thực hệ thống

MusicBox

🏠 Trang chủ

BXH

ⓧ Đăng Xuất

Em Có Chắc Không (?) (Bài Ca Rebound)

Ngọt - 3

00:05

04:35

No	Name	Artist	Album	Duration	View	Format
3	Cô Ấy (Live from Lan Anh, Saigo...	Cá Hồi Hoang	Fx (Live from Lan ...	04:16	96	flac
5	Hết Mực (Live from Lan Anh, Sai...	Cá Hồi Hoang	Fx (Live from Lan ...	03:37	48	flac
21	Tầng Thượng 102 (Live from Lan...	Cá Hồi Hoang	Fx (Live from Lan ...	05:24	38	flac
1	春はゆく	Aimer	春はゆく / marie	05:04	35	flac
12	Cánh Đồng (Live from Lan Anh, ...	Cá Hồi Hoang	Fx (Live from Lan ...	03:00	32	flac
2	Điền Vào Ô Trống (250) (Live fro...	Cá Hồi Hoang	Fx (Live from Lan ...	03:44	20	flac
4	Acid8 (Live from Lan Anh, Saigo...	Cá Hồi Hoang	Fx (Live from Lan ...	04:45	20	flac
3	Run Riot	Aimer	春はゆく / marie	04:16	16	flac
2	marie	Aimer	春はゆく / marie	05:07	11	flac
1	Hiệu Ứng Bắt Đầu (Live from La...	Cá Hồi Hoang	Fx (Live from Lan ...	01:31	8	flac
4	Binary Star_moll	SawanoHiroyuki[...	Tranquility / Trollz	04:43	8	flac
4	花の唄 end of spring ver.	Aimer	春はゆく / marie	05:51	5	flac
6	Hiệu Ứng Ngược (Live from Lan ...	Cá Hồi Hoang	Fx (Live from Lan ...	03:25	4	flac
7	4S (Live from Lan Anh, Saigon. 2...	Cá Hồi Hoang	Fx (Live from Lan ...	04:26	3	flac
8	Hiệu Ứng Trốn Chạy (Live from L...	Cá Hồi Hoang	Fx (Live from Lan ...	04:20	3	flac
1	Cho Tôi Lang Thang	Ngọt, Đen	Cho Tôi Lang Thang	04:18	2	flac
9	Inside Mr.Bin (Live from Lan An...	Cá Hồi Hoang	Fx (Live from Lan ...	03:46	2	flac
11	Khô Khan (Live from Lan Anh, Sa...	Cá Hồi Hoang	Fx (Live from Lan ...	03:10	2	flac
13	Đến Bao Giờ (feat. Datmaniac) [...	Cá Hồi Hoang	Fx (Live from Lan ...	04:20	2	flac
14	Ngày Nào (feat. Datmaniac) [Liv...	Cá Hồi Hoang	Fx (Live from Lan ...	04:07	2	flac
15	Bandamusical (Live from Lan An...	Cá Hồi Hoang	Fx (Live from Lan ...	08:39	2	flac
16	Bin (Live from Lan Anh, Saigon. ...	Cá Hồi Hoang	Fx (Live from Lan ...	03:06	2	flac

Bảng xếp hạng nhạc theo lượt nghe

8.DEMO

9. Kết luận

- Ứng dụng này giúp chúng ta nghe nhạc trực tuyến.
- Qua đồ án này , cả nhóm đã được học tập, áp dụng và phát triển kĩ năng lập trình.
- Kĩ năng làm việc nhóm.
- Phân chia công việc cho từng thành viên