

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



Báo cáo cuối kỳ môn Lập trình mạng căn bản

ỨNG DỤNG NGHE NHẠC

RHYTHMBOX



Giảng viên bộ môn:
Ths. Lê Minh Khánh Hội

Sinh viên thực hiện:
Đặng Tiến Đạt - 21521929
Nguyễn Hoàng Long - 21522304
Nguyễn Thái Long - 21522305
Nguyễn Văn Huân - 21522108

Lớp NT106.N21.MMCL - Khoá K16 - Nhóm 6

5, 2023

Tóm tắt

Bài báo cáo này sẽ giới thiệu một cách tổng quát ứng dụng nghe nhạc RhythmBox cũng như đề cập tới các định nghĩa về các giao thức được sử dụng trong ứng dụng và các công nghệ kèm theo. Đồng thời trình bày các sơ đồ về hệ thống để người dùng nắm rõ hơn về ứng dụng nghe nhạc. Cuối cùng là giao diện của ứng dụng sau khi hoàn thành.

MỤC LỤC

Chương 1. Tổng quan	5
I. Giới thiệu về đề tài	5
II. Cơ sở lý thuyết	5
a. TCP/IP	6
b. HTTP	6
c. API.....	6
d. TDS.....	7
e. SMB.....	7
III. Các công nghệ được sử dụng.....	8
a. WinForms (client)	8
b. ASP.NET (server).....	8
c. Portal Azure (Cơ sở dữ liệu, File Share và App Service)	9
Chương 2. Phân tích thiết kế hệ thống	9
I. Sơ đồ kiến trúc tổng quan hệ thống.....	9
a. Sơ đồ kiến trúc	9
b. Biểu đồ cơ sở dữ liệu.....	10
II. Network Stack	13
III. Sơ đồ Use Case	14
IV. Luồng của ứng dụng.....	16
Chương 3. Hiện thực hệ thống	17
Chương 4. Tổng Kết.....	23
I. Kết luận	23
II. Hướng phát triển.....	23

MỤC LỤC CÁC HÌNH

Hình 1: Mô hình OSI [2].....	6
Hình 2: Quy trình kết nối của giao thức TDS [6].	7
Hình 3: Cách hoạt động của giao thức SMB [7].....	8
Hình 4: Sơ đồ kiến trúc của ứng dụng.	9
Hình 5: Biểu đồ cơ sở dữ liệu.	11
Hình 6: Mô hình Network Stack.....	14
Hình 7: Sơ đồ Use Case của ứng dụng	15
Hình 8: Sơ đồ luồng của ứng dụng.	16
Hình 9: Form login	18
Hình 10: Form signup	19
Hình 11: Form Forgot password	20
Hình 12: Form nhập OTP	21
Hình 13: Form nhập lại mật khẩu mới.....	22
Hình 14: Form giao diện chính (với người dùng là user)	23

Chương 1. Tổng quan

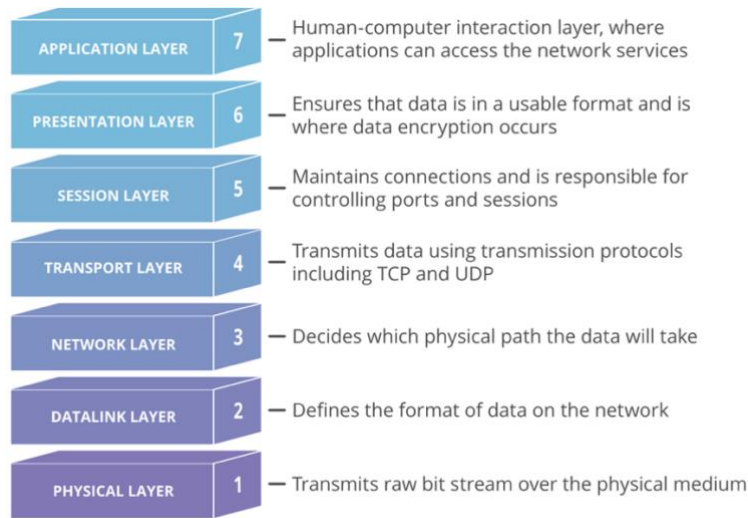
I. Giới thiệu về đề tài

Trong những năm gần đây, số lượng người sử dụng các ứng dụng streaming nhạc để nghe những bài nhạc yêu thích để thư giãn, học tập, làm việc hay tập thể dục đang dần trở nên phổ biến hơn. Với cùng với sự phát triển nhanh chóng của công nghệ, việc nghe nhạc trực tuyến đã trở nên phổ biến hơn bao giờ hết, và ứng dụng streaming nhạc đã trở thành người bạn đồng hành không thể thiếu cho những người yêu âm nhạc. Theo báo vietnamnet đã đề cập: *"Đối với Việt Nam, ước tính hiện có hơn 34 triệu người sở hữu smartphone và dự kiến tới năm 2022 con số này sẽ đạt 53 triệu. Do đó lượng người stream nhạc trong nước cũng được kỳ vọng sẽ tăng trưởng với tốc độ nhanh hơn, khi streaming trở thành hình thức phổ biến và hàng ngàn nghệ sĩ hiện diện trên các nền tảng stream nhạc."* [1]. Có thể thấy rằng, việc các nghệ sĩ phát hành những bài nhạc để tải lên các nền tảng streaming nhạc cũng như lượng người dùng nghe những bài nhạc trực tuyến đã cho thấy mức độ phổ biến của các ứng dụng streaming nhạc. Vì thế, nhóm chúng em đã lựa chọn xây dựng một ứng dụng streaming nhạc on-demand (giúp cho người dùng truy cập vào một thư viện âm nhạc và lựa chọn bất kỳ bài hát, album hoặc playlist nào mà họ muốn nghe và có thể phát ngay lập tức trên máy tính mà không cần phải tải xuống và lưu trữ cục bộ).

Sau quá trình nghiên cứu và phát triển, ứng dụng nghe nhạc trực tuyến RhythmBox đã được hoàn thành. Ứng dụng cho phép người dùng truy cập vào một thư viện rộng lớn với đa dạng các thể loại từ R&B, pop, rock, hip-hop đến nhạc điện tử. Ngoài ra người dùng còn có thể tìm kiếm các bài nhạc, nghe những album mới ra, truy cập vào trang thông tin của những nghệ sĩ cũng như tự tạo cho mình một playlist để có thể nghe những bài hát của mình yêu thích mà không cần phải tìm kiếm lại, và để thực hiện tất cả những điều đó chỉ đơn giản là một cú nhấp chuột. Điểm mạnh của RhythmBox là nghe nhạc ở mọi nơi, mọi lúc và được dùng cho mọi người, miễn là chiếc máy tính được kết nối với internet.

Tóm lại, các ứng dụng streaming nhạc đã thay đổi cách chúng ta nghe nhạc, "thường thức" nhạc bất cứ lúc nào và chúng em mong muốn rằng ứng dụng RhythmBox sẽ đóng góp được một phần nào trong nền công nghiệp streaming nhạc hiện nay.

II. Cơ sở lý thuyết



Hình 1: Mô hình OSI [2].

a. TCP/IP

Transmission Control Protocol/ Internet Protocol là giao thức điều khiển để truyền nhận liên mạng, Đây là một bộ giao thức có chức năng truyền và kết nối các thông tin giữa các thiết bị trong một mạng lưới internet [3] gồm hai thành phần:

- TCP: Transmission Control Protocol hoạt động ở tầng vận chuyển (transport layer) của mô hình OSI (Open Systems Interconnection) (Hình 1), đây là một giao thức hướng kết nối (connection-oriented), có nghĩa là phải thiết lập kết nối tới endpoint để truyền tải dữ liệu. Tiến trình thiết lập kết nối ở TCP được gọi là tiến trình bắt tay 3 bước (three-way handshake) [4]. Thường được sử dụng ở cổng 80, 443, 25 và 21.
- IP: Internet protocol hoạt động ở tầng mạng (network layer) của mô hình OSI (Hình 1). IP sẽ gán các địa chỉ và định tuyến từng gói thông tin. Mỗi mạng sẽ có 1 địa chỉ IP để xác định được chính xác nơi chuyển/nhận thông tin, dữ liệu [3].

b. HTTP

Hyper Text Transfer Protocol: là một giao thức truyền tải siêu văn bản thường được sử dụng trong Word Wide Web (www) để tải các trang web thông qua các đường link. HTTP là một giao thức hoạt động ở tầng ứng dụng (application layer) (Hình 1) để chuyển các thông tin giữa các ứng dụng mạng với nhau, ví dụ trao đổi thông tin giữa server và các trình duyệt web [5]. HTTP hoạt động ở cổng 80 là chủ yếu.

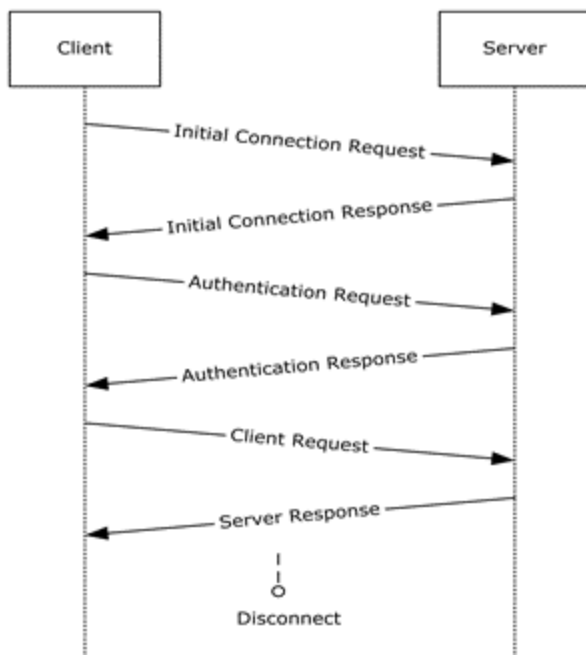
c. API

Application Programming Interface: Là một phần mềm trung gian cho phép hai ứng dụng giao tiếp với nhau. Ở dạng đơn giản nhất, API là giao diện cho phép một ứng dụng giao tiếp với ứng dụng khác thông qua các lệnh đơn giản và cách các lệnh này được gửi và định dạng mà dữ liệu được truy xuất thông qua API có thể khác với API SOAP hoặc REST. Ứng dụng RhythmBox sử dụng REST API nên trong phạm vi bài báo cáo này sẽ chỉ đề cập đến REST API là gì.

REST API: Representational State Transfer được sử dụng khá phổ biến và linh hoạt trong các ứng dụng web hiện nay. Client sẽ gửi dữ liệu đến server, và server sẽ xử lý các dữ liệu này và trả kết quả về lại cho Client. REST định nghĩa các hàm như GET, PUT, POST, DELETE, ... để client có thể truy cập vào dữ liệu của server. Client và server trao đổi dữ liệu với nhau qua HTTP (đã đề cập ở trên).

d. TDS

Tabular Data Stream: Là một giao thức trong ADO.NET được Microsoft phát triển đặc biệt để truyền dữ liệu giữa ứng dụng và SQL Server. Là một giao thức hoạt động ở tầng ứng dụng (*Hình 1*) của mô hình TCP/IP, được dùng để trao đổi những request và response giữa những client và hệ thống server cơ sở dữ liệu [6].



Hình 2: Quy trình kết nối của giao thức TDS [6].

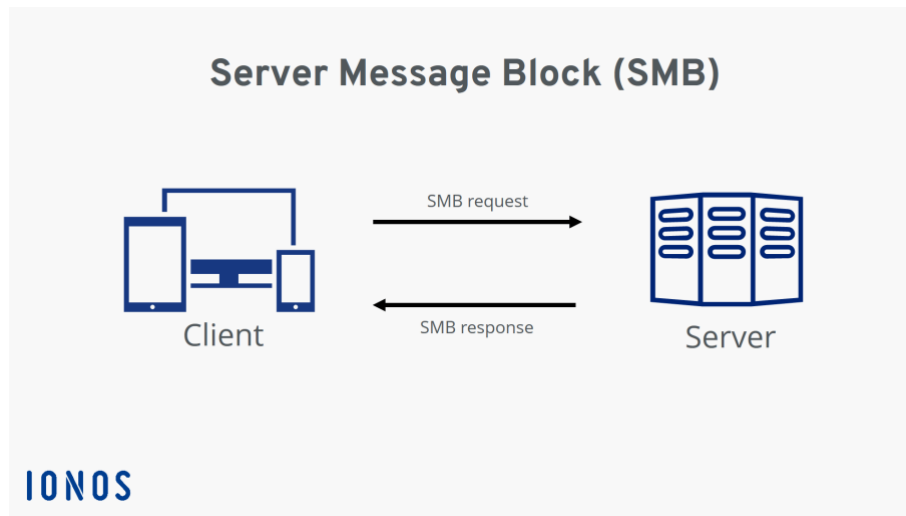
Khi client hình thành một kết nối dài hạn (long-lived) tới server, kết nối sẽ sử dụng giao thức ở tầng vận chuyển là TCP/IP, các thông điệp TDS được dùng để giao tiếp giữa client và server.

Server cơ sở dữ liệu cũng có thể đóng vai trò là một client, trong trường hợp đó phải thiết lập một TDS mới.

Phiên làm việc của TDS gắn liền với phiên của tầng vận chuyển, điều đó có nghĩa là một TDS được thiết lập khi kết nối ở tầng vận chuyển được thiết lập và server nhận được yêu cầu thiết lập một kết nối TDS. Nó sẽ tồn tại đến khi kết nối ở tầng vận chuyển bị đóng.

Đây là kết nối giữa server và cơ sở dữ liệu được lưu trữ trong Azure của ứng dụng RhythmBox.

e. SMB



Hình 3: Cách hoạt động của giao thức SMB [7].

Server Message Block: Là một giao thức client/server hoạt động ở tầng ứng dụng (Hình 1) cho phép truy cập vào các tệp tin và các đường dẫn, cũng như các thiết bị mạng khác như máy in, router. Ngoài ra, các thông tin được trao đổi giữa các tiến trình khác nhau trong cùng một hệ thống máy tính cũng có thể được xử lý bởi giao thức SMB [7].

Ngoài ra, SMB còn sử dụng giao thức truyền dữ liệu TCP/IP để đảm bảo việc truyền dữ liệu và tin nhắn an toàn và đáng tin cậy qua mạng. Ban đầu SMB được phát triển bởi Microsoft và đã trở thành tiêu chuẩn cho việc chia sẻ tệp tin trong mạng Windows.

Đây là kết nối giữa server và File Share trong Azure Storage của ứng dụng RhythmBox.

III. Các công nghệ được sử dụng

a. WinForms (client)

Windows Form là một phần của Microsoft .NET Framework, cung cấp một bộ công cụ để phát triển ứng dụng giao diện người dùng (GUI) trên nền tảng Windows. Nền tảng phát triển Windows Form hỗ trợ một loạt các tính năng dùng để phát triển các ứng dụng như là bộ điều khiển, đồ họa, liên kết dữ liệu và nhận đầu vào từ người dùng. WinForms có trình thiết kế trực quan kéo - thả (drag - and - drop) trong Visual Studio để dễ dàng tạo các ứng dụng Windows Forms.

RhythmBox sử dụng WinForms để tạo giao diện và xử lý thông tin bên phía client.

b. ASP.NET (server)

ASP.NET là một framework web mã nguồn mở được tạo bởi Microsoft để xây dựng các ứng dụng web hiện đại và các dịch vụ web với công nghệ .NET [8]. ASP.NET cung cấp cơ sở hạ tầng phần mềm toàn diện và các dịch vụ khác nhau để xây dựng các ứng dụng web mạnh mẽ cho PC cũng như thiết bị di động [9].

ASP.NET hoạt động dựa trên giao thức HTTP, và sử dụng các lệnh và chính sách của HTTP để thiết lập liên kết giữa trình duyệt và server [9].

RhythmBox sử dụng ASP.NET cụ thể là Blazor Server App để xử lý các thông tin gửi từ client và xây dựng liên kết tới cơ sở dữ liệu của server.

c. Portal Azure (Cơ sở dữ liệu, File Share và App Service)

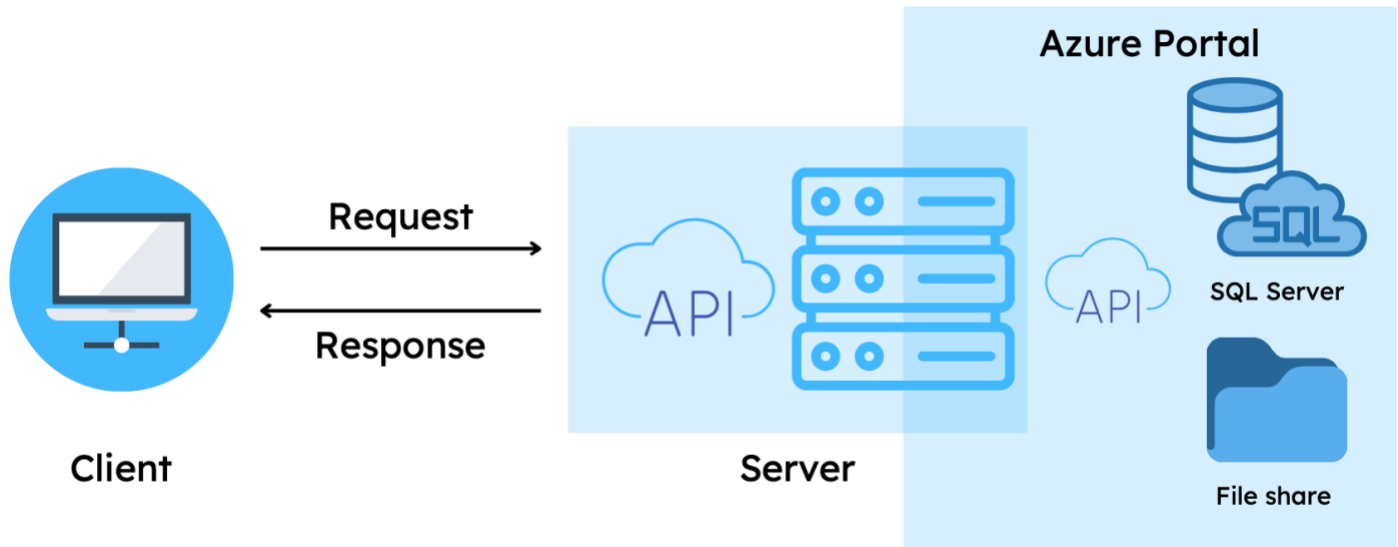
Portal Azure là một nền tảng điện toán đám mây được cung cấp bởi Microsoft Azure để quản lý và điều khiển các tài nguyên và dịch vụ trên nền tảng Azure. Được hoạt động dựa trên ứng dụng web, Portal Azure cung cấp những giải pháp thay thế công cụ dòng lệnh và có thể xây dựng, quản lý và giám sát mọi thứ từ ứng dụng web đơn giản đến triển khai đám mây phức tạp trong công thông tin [10].

RhythmBox sử dụng Portal Azure để triển khai cơ sở dữ liệu SQL Server, tổ chức lưu các tệp tin trong File Share và triển khai ứng dụng server bằng App Service.

Chương 2. Phân tích thiết kế hệ thống

I. Sơ đồ kiến trúc tổng quan hệ thống

a. Sơ đồ kiến trúc



Hình 4: Sơ đồ kiến trúc của ứng dụng.

Vì là ứng dụng hoạt động dựa trên việc truyền tải các thông tin qua internet nên RhythmBox sẽ được chia làm hai hướng phát triển chính là client và server.

Đối với phía client:

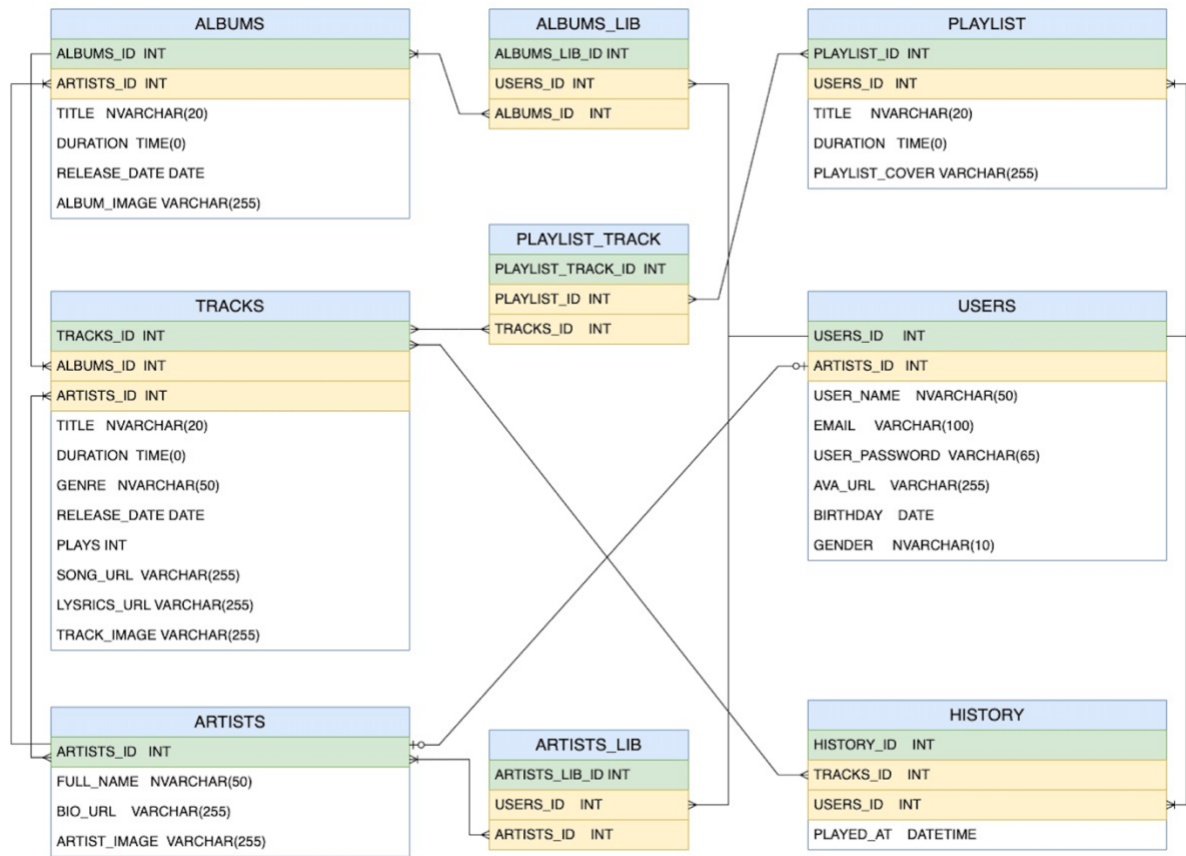
- Ứng dụng sẽ xây dựng giao diện dựa bằng WinForms cùng với thư viện Bunifu Framework được cài thêm.
- Xử lý các thông tin mà người dùng nhập vào từ các form của giao diện hay những cái nhấp chuột.
- Sau khi xử lý các thông tin, ứng dụng sẽ chuyển đổi thành dữ liệu dạng json và gửi các dữ liệu đó tới server thông qua các API được server tạo ra.
- Khi nhận dữ liệu từ server dưới dạng json, ứng dụng sẽ chuyển thành một JObject hoặc JArray để xử lý các dữ liệu đó và in lên màn hình hoặc thực hiện một chức năng cụ thể cho người dùng.

Đối với phía server:

- Ứng dụng sẽ xây dựng các API tương ứng với các chức năng bên phía client.
- Sau khi nhận dữ liệu từ client, ứng dụng sẽ xử lý các dữ liệu đó và trả kết quả lại cho client.
- Ứng dụng sẽ được triển khai ở môi trường public bằng dịch vụ điện toán đám mây Portal Azure. Nhờ đó, không chỉ những thiết bị cùng mạng LAN sử dụng được dịch vụ từ server và các ứng dụng khác mạng cũng có thể sử dụng được.
- Ngoài ra, thông tin người dùng, nghệ sĩ, danh sách các bài hát hay các album đều được lưu trữ ở cơ sở dữ liệu SQL Server. SQL Server cũng được lưu trữ tập trung ở Portal Azure để tiện hơn trong quá trình phát triển cũng như cung cấp các dữ liệu cho server.
- Nếu lưu trữ các tệp tin âm nhạc, tệp tin hình ảnh trong cơ sở dữ liệu sẽ làm giảm hiệu suất truy vấn cũng như tốn nhiều dung lượng hơn cho cơ sở dữ liệu, vì thế nhóm đã sử dụng thêm dịch vụ Azure Storage để lưu trữ các tệp tin đó.

Client và server sẽ giao tiếp thông qua các API, server sẽ giao tiếp với SQL Server bằng giao thức TDS (Đã đề cập ở trên) và giao tiếp với File Share thông qua giao thức SMB (Đã đề cập ở trên). Cả hai giao thức TDS và SMB đều hoạt động thông qua API.

b. Biểu đồ cơ sở dữ liệu



Hình 5: Biểu đồ cơ sở dữ liệu.

Ứng dụng RhythmBox sử dụng 9 bảng gồm: PLAYLIST, USERS, HISTORY, ALBUMS, TRACKS, ARTISTS, ALBUMS_LIB, PLAYLIST_TRACK VÀ ARTIST_LIB.

- **PLAYLIST:** Bảng này sẽ được dùng để lưu trữ các playlist của người dùng như mã người dùng (USER_ID), tên của playlist đó (TITLE) sẽ được lưu trữ mặc định là "My Playlist #x" với x là thứ tự playlist có tên là mặc định tồn tại trong bảng PLAYLIST, ngoài ra nếu người dùng có thể sửa thành tên mình muốn. Ngoài ra bảng PLAYLIST còn lưu trữ tổng thời lượng (DURATION) của các bài hát có trong playlist cụ thể của người dùng và cuối cùng đường dẫn đến ảnh của playlist (PLAYLIST_COVER), ảnh của playlist sẽ là ảnh của bài hát (track) đầu tiên có trong playlist đó.
- **USERS:** Bảng này sẽ được dùng để lưu thông tin của người dùng như mã người dùng (USERS_ID), tên của người dùng (USER_NAME), email của người dùng (EMAIL), password để đăng nhập vào ứng dụng của người dùng (USER_PASSWORD) được lưu dưới dạng mã băm SHA-256, đường dẫn đến ảnh của người dùng (AVA_URL), khi tạo tài khoản mới ảnh của người dùng sẽ là ảnh mặc định và người dùng có thể chỉnh sửa lại ảnh mình mong muốn. Ngoài ra bảng còn lưu trữ ngày sinh (BIRTHDAY) và giới tính của người dùng (GENDER).

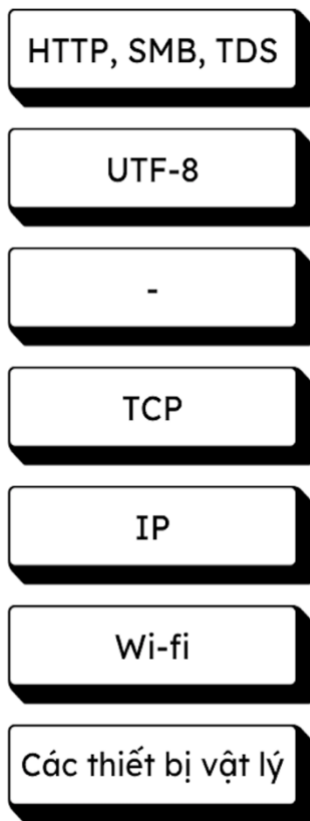
- **HISTORY:** Bảng này sẽ được dùng để lưu trữ lịch sử nghe nhạc của người dùng như mã lịch sử (HISTORY_ID), mã bài hát mà người dùng nghe (TRACKS_ID) và cuối cùng là thời gian khi người dùng bắt đầu nghe (PLAYED_AT).
- **ALBUMS:** Bảng này sẽ được dùng để lưu trữ thông tin các album của nghệ sĩ. Đầu tiên là mã album (ALBUMS_ID), mã nghệ sĩ (ARTISTS_ID) để chỉ định album này của nghệ sĩ nào, tiêu đề của album (TITLE), tổng thời lượng của album (DURATION), ngày phát hành của album (RELEASE_DATE) và cuối cùng là đường dẫn đến hình ảnh của album (ALBUM_IMAGE).
- **TRACKS:** Bảng này sẽ được dùng để lưu trữ thông tin của các bài hát như mã bài hát (TRACKS_ID), mã album (ALBUMS_ID) mà chứa bài hát đó, trường hợp bài hát không thuộc album nào thì giá trị là NULL, tiếp theo là mã nghệ sĩ (ARTIST_ID) thể hiện bản nhạc này thuộc quyền sở hữu của nghệ sĩ nào, tên của bản nhạc (TITLE), thời lượng của bản nhạc (DURATION), thể loại của bản nhạc (GENRE), ngày phát hành (RELEASE_DATE), số lượng người nghe (PLAYS), đường dẫn đến tệp tin chứa âm thanh của bản nhạc đó (SONG_URL), đường dẫn đến lyrics (lời bài hát) của bản nhạc đó (LYRICS_URL) và cuối cùng là đường dẫn đến tệp tin chứa hình ảnh của bản nhạc đó (TRACK_IMAGE), hình ảnh của bản nhạc sẽ chính là hình ảnh của album chứa bản nhạc đó, trong trường hợp bản nhạc không thuộc album nào thì ảnh của bản nhạc sẽ chính là ảnh của nghệ sĩ sở hữu bản nhạc đó.
- **ARTISTS:** Bảng này sẽ được dùng để lưu trữ các thông tin của một nghệ sĩ, đầu tiên là mã nghệ sĩ (ARTISTS_ID), nghệ danh của nghệ sĩ đó (FULL_NAME), đường dẫn tới tệp tin chứa thông tin giới thiệu về nghệ sĩ (BIO_URL), đường dẫn tới tệp tin chứa hình ảnh của nghệ sĩ (ARTIST_IMAGE). Bảng nghệ sĩ sẽ được tạo tự động khi một người dùng đăng kí làm nghệ sĩ.
- **ALBUMS_LIB:** Bảng này được dùng để lưu trữ các album mà người dùng theo dõi trong mục Library. Đầu tiên sẽ chứa mã của bảng (ALBUMS_LIB_ID), tiếp theo sẽ là mã của người dùng (USERS_ID) cùng với mã của album (ALBUMS_ID) mà người dùng theo dõi.
- **PLAYLIST_TRACK:** Bảng này sẽ được sử dụng để lưu trữ thông tin các track nào ở playlist nào. Cụ thể, bảng sẽ lưu trữ mã của bảng (PLAYLIST_TRACK_ID, cùng với là mã của playlist (PLAYLIST_ID) và mã bản nhạc (TRACKS_ID).
- **ARTISTS_LIB:** Bảng này sẽ lưu trữ các nghệ sĩ mà người dùng theo dõi trong mục Library. Đầu tiên sẽ chứa mã của bảng (ARTISTS_LIB_ID), tiếp theo sẽ là mã của người dùng (USERS_ID) cùng với mã của nghệ sĩ (ARTISTS_ID) mà người dùng theo dõi.

Các bảng này sẽ có các quan hệ một - một, một - nhiều, nhiều - nhiều. Cụ thể hơn như sau:

- **USERS** sẽ có quan hệ nhiều - nhiều với bảng **ALBUMS_LIB** và bảng **ALBUMS_LIB** sẽ có quan hệ nhiều - nhiều với bảng **ALBUMS**: một user có thể có đăng ký nhiều album trong mục Library và một album có thể được nhiều user đăng ký.

- USERS sẽ có quan hệ nhiều - nhiều với bảng ARTIST_LIB bảng ARTISTS_LIB sẽ có quan hệ nhiều - nhiều với bảng ARTISTS: một user có thể có đăng ký nhiều nghệ sĩ trong mục Library và một nghệ sĩ có thể được nhiều user đăng ký.
- USERS sẽ có quan hệ một - nhiều với bảng PLAYLIST: một user có thể tạo nhiều playlist và chỉ một playlist thuộc sở hữu của một user.
- USERS có quan hệ một - nhiều với bảng HISTORY: một user có thể có nhiều lịch sử nghe nhạc và chỉ một lịch sử nghe nhạc thuộc sở hữu của user.
- USERS sẽ có quan hệ một - một với bảng ARTIST: một user có thể là 1 nghệ sĩ hoặc không và 1 nghệ sĩ có thể là 1 user hoặc không.
- PLAYLIST có quan hệ nhiều - nhiều với bảng PLAYLIST_TRACK và bảng PLAYLIST_TRACK sẽ có quan hệ nhiều - nhiều với bảng TRACKS: một playlist sẽ chứa nhiều bản nhạc và một bản nhạc có thể thuộc nhiều playlist khác nhau.
- TRACKS có quan hệ một - nhiều với bảng ALBUMS: một bản nhạc chỉ có thể thuộc một album và một album có thể có nhiều bản nhạc.
- TRACKS có quan hệ nhiều - nhiều với bảng ARTISTS: một bản nhạc sẽ thuộc sở hữu của nhiều nghệ sĩ và một nghệ sĩ sẽ sở hữu nhiều bản nhạc.
- ALBUMS sẽ có quan hệ một - nhiều với bảng ARTISTS: một album sẽ thuộc một nghệ sĩ và một nghệ sĩ sẽ có nhiều album.

II. Network Stack



Hình 6: Mô hình Network Stack

Tầng 7: Tầng ứng dụng - Ứng dụng sẽ sử dụng các giao thức HTTP, SMB, TDS để giao tiếp thông qua API cho client trao đổi các dữ liệu và server truy vấn cơ sở dữ liệu cũng như truy cập vào File Share.

Tầng 6: Tầng trình diễn - Ứng dụng sẽ sử dụng cách thức mã hoá từ một tệp tin chứa âm thanh, hình ảnh, văn bản thành một chuỗi byte bằng UTF-8.

Tầng 5: Tầng phiên - Ứng dụng sẽ không trực tiếp sử dụng tầng phiên nên sẽ không đề cập ở đây.

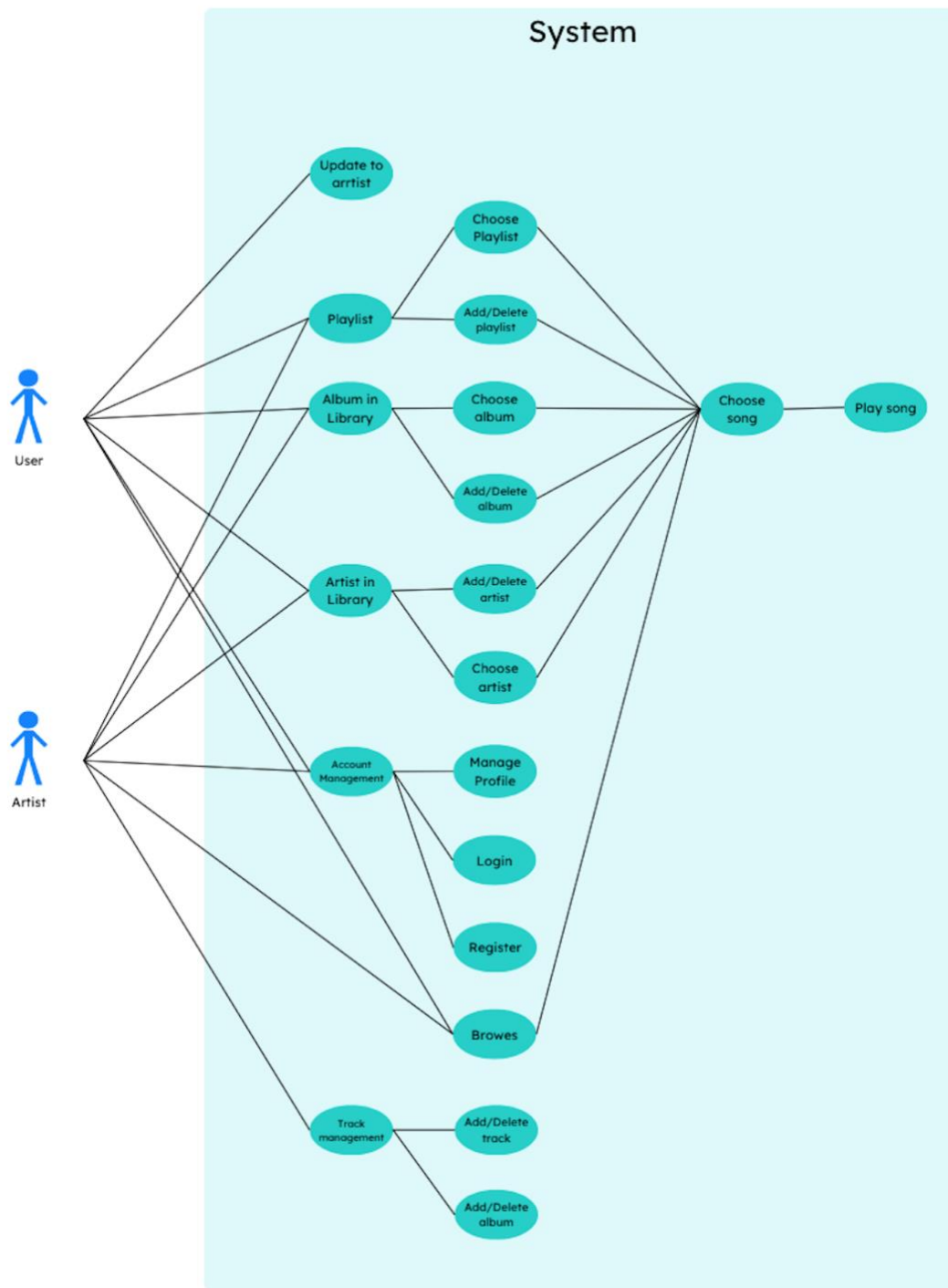
Tầng 4: Tầng vận chuyển - Ứng dụng sẽ sử dụng giao thức TCP để trao đổi các gói tin một cách tin cậy và an toàn.

Tầng 3: Tầng mạng - Ứng dụng sử dụng giao thức IP để xác định đích đến và địa chỉ mạng của API.

Tầng 2: Tầng liên kết - dữ liệu âm thanh có thể được đóng gói vào các gói tin và được truyền qua các giao diện mạng như Wi-Fi.

Tầng 1: Tầng vật lý - Tầng này bao gồm các thành phần vật lý như dây cáp, thiết bị mạng và các giao diện mạng để truyền dữ liệu âm thanh giữa máy nghe nhạc và máy chủ.

III. Sơ đồ Use Case



Hình 7: Sơ đồ Use Case của ứng dụng

Ứng dụng RhythmBox sẽ cung cấp các tính năng cho người dùng (user) và nghệ sĩ (artist) tương tự nhau.

Các tính năng chung:

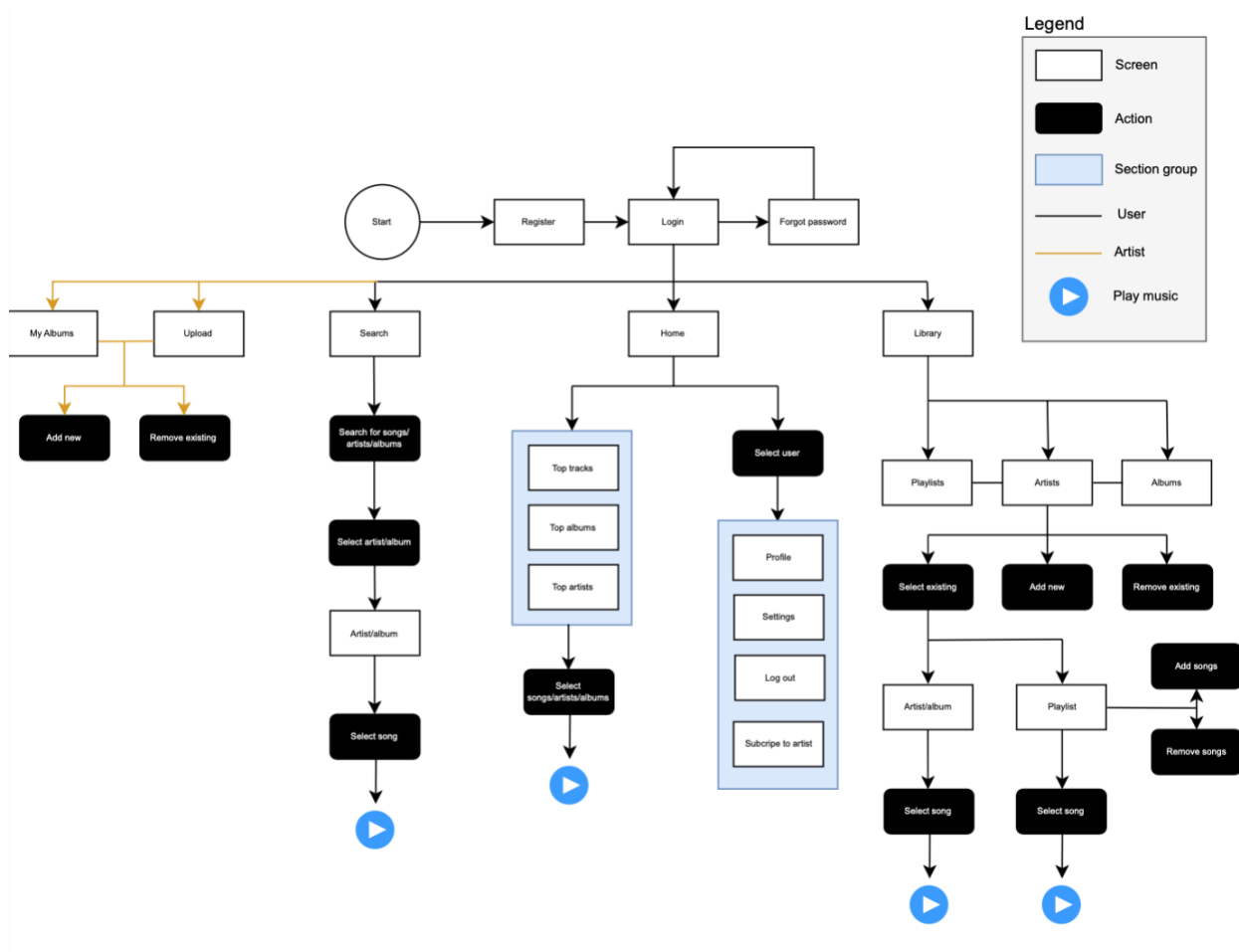
- Tính năng quản lý Playlist: cả người dùng và nghệ sĩ đều có thể tự quản lý playlist của riêng mình như tạo mới, xoá, sửa và thêm các bài hát vào playlist đó.

- Tính năng quản lý album trong thư viện: cả người dùng và nghệ sĩ đều có thể tự quản lý album trong thư viện như theo dõi mới một album, xoá album đã theo dõi.
- Tính năng quản lý nghệ sĩ trong thư viện: cả người dùng và nghệ sĩ đều có thể tự quản lý các nghệ sĩ mình theo dõi trong thư viện như thêm nghệ sĩ mới hay xoá nghệ sĩ đã theo dõi.
- Tính năng quản lý tài khoản: cả người dùng và nghệ sĩ đều có thể tự quản lý tài khoản của mình như đăng nhập, đăng ký, chỉnh sửa Profile.

Tính năng khác nhau:

- Người dùng sẽ có thêm tính năng đăng lý làm nghệ sĩ, sau khi đăng ký, vai trò của người dùng bây giờ là nghệ sĩ.
- Nghệ sĩ sẽ có thêm tính năng quản lý các bài nhạc cũng như album của mình.

IV. Luồng của ứng dụng



Hình 8: Sơ đồ luồng của ứng dụng.

Đây chính là sơ đồ luồng ứng dụng dựa trên sơ đồ Use Case ở trên và sơ đồ này cũng sẽ thể hiện hai vai trò là vai trò người dùng và vai trò nghệ sĩ.

Đầu tiên khởi chạy ứng dụng, tiếp theo là đăng ký nếu chưa có tài khoản, sau khi đăng ký sẽ đăng nhập. Ở bước đăng nhập nếu người dùng quên mật khẩu thì sẽ được chuyển qua trang quên mật khẩu để tiến hành đặt mật khẩu.

Sau khi đăng nhập thành công, người dùng sẽ có 3 trang chính để thực hiện các thao tác:

- Trang Home: Ở trang Home, người dùng sẽ thấy 3 panel chứa top tracks, top albums và top artists, người dùng có chọn cụ thể một bài nhạc, một album hay một nghệ sĩ tùy theo sở thích. Khi người dùng chọn một bản nhạc, thì bản nhạc đó sẽ được phát ngay lập tức, nếu người dùng chọn album thì sẽ mở ra một trang chứa thông tin của album đó và người dùng có thể chọn nhạc trực tiếp trong album. Khi người dùng chọn nghệ sĩ, người dùng sẽ thấy được thông tin của nghệ sĩ đó, các bài nhạc và các album của nghệ sĩ. Khi chọn vào nút User, người dùng sẽ được chuyển hướng đến trang cho phép thay đổi các thông tin của người dùng, cài đặt, đăng xuất hoặc đăng ký là một nghệ sĩ.
- Trang Search: Ở trang Search, người dùng sẽ nhập từ khoá muốn tìm kiếm vào khung search, sau khi trả kết quả theo nhạc, album, nghệ sĩ, người dùng có thể tùy ý chọn theo mong muốn và sở thích.
- Trang Library: Ở trang Library, người dùng sẽ được quản lý các playlist, album và nghệ sĩ của mình. Được thêm, xoá các playlist, album, nghệ sĩ và sửa tên đối với playlist.

Riêng đối với vai trò là nghệ sĩ:

- Trang Upload: Ở trang Upload nghệ sĩ sẽ được xem các danh sách bản nhạc mà mình phát hành, xoá những bản nhạc, thêm bản nhạc mới.
- Trang My Albums: Ở trang My Albums, nghệ sĩ sẽ được quản lý các album của mình như thêm mới, xoá và sửa các album.

Chương 3. Hiện thực hệ thống

I. Giao diện hệ thống

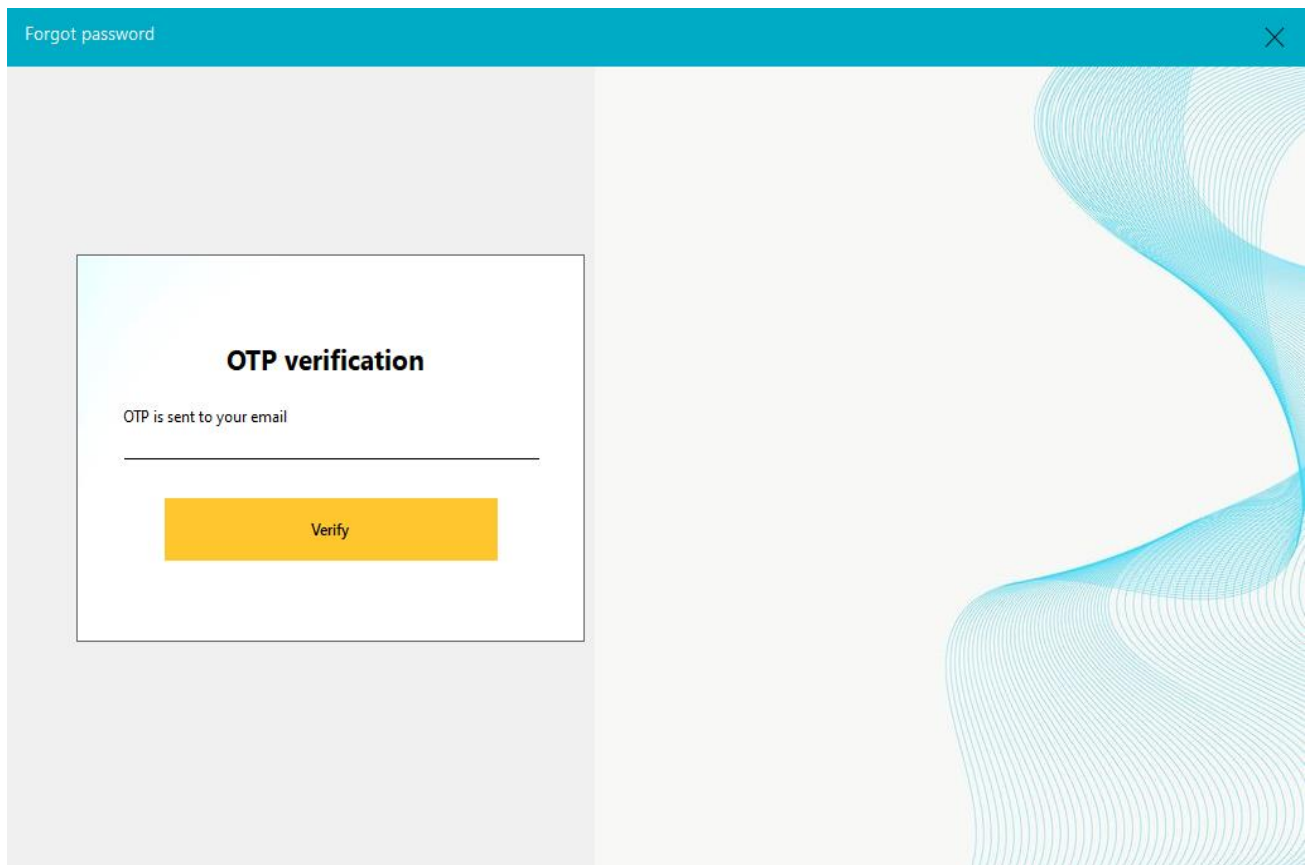
a. Trang đăng nhập (Login page)

A screenshot of a web application window titled "Forgot password" with a close button (X) in the top right corner. The main content area has a light gray background with a blue abstract wave pattern on the right side. In the center, there is a white rectangular box containing the text "Forgot password" in bold. Below this text is an "Email" label followed by a horizontal input field. At the bottom of the box is a yellow button with the text "Forgot password".

Hình 11: Form Forgot password

Trong trường hợp người dùng quên mật khẩu, hệ thống sẽ yêu cầu người dùng cung cấp email đã đăng ký trước đó để hệ thống tiến hành làm các thủ tục cấp mật khẩu mới.

Nếu email hợp lệ (tức hệ thống sẽ kiểm tra tính tồn tại của email ứng với tài khoản đó), hệ thống sẽ gửi mã OTP đến cho email của người dùng. Người dùng lúc này cần kiểm tra email để lấy OTP.

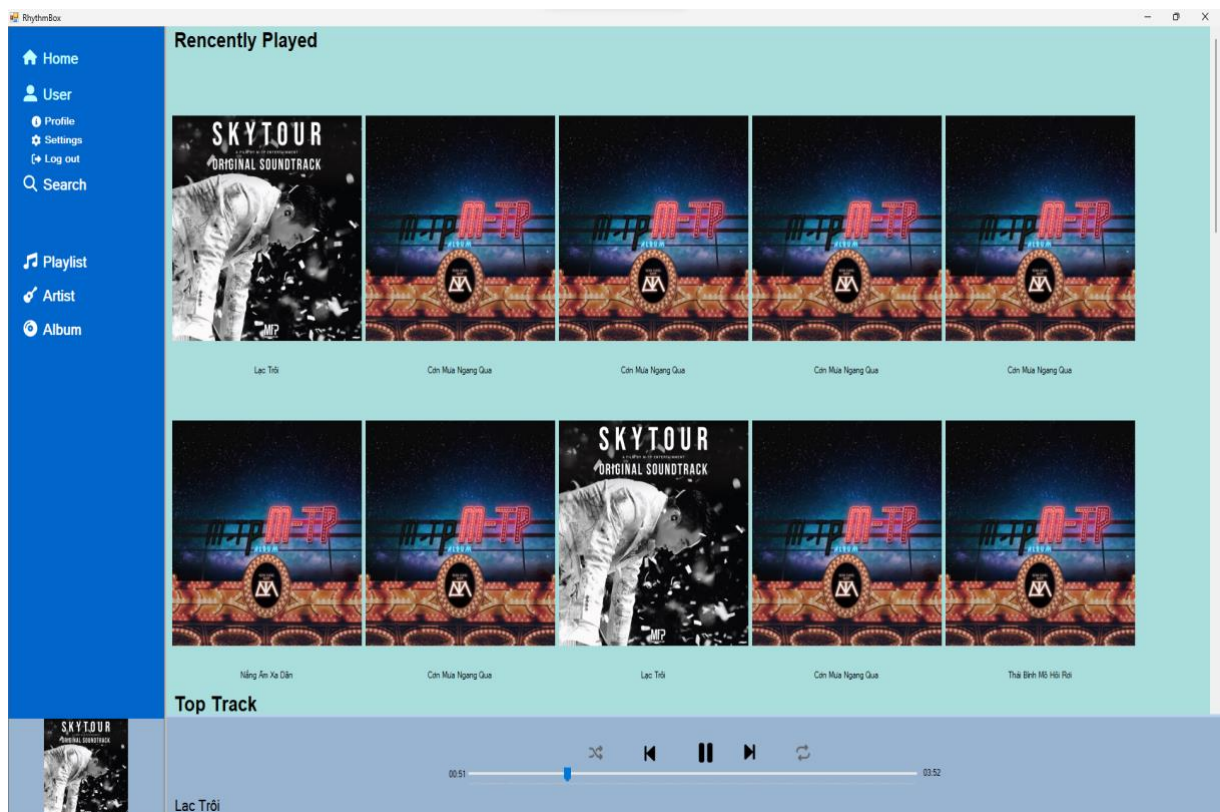


Hình 12: Form nhập OTP

Sau khi người dùng nhập OTP vào, nếu trùng với mã mà hệ thống gửi, người dùng có thể lấy lại mật khẩu sau đó.

[illegible]

Đây là giao diện dành cho người dùng nhập mật khẩu mới. Sau khi thành công, hệ thống sẽ chuyển người dùng đến trang đăng nhập.



Chương 4. Tổng Kết

I. 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.

Tuy sản phẩm sau khi hoàn thành không được đầy đủ các chức năng mà nhóm hướng tới nhưng đây cũng là tâm huyết và công sức mà nhóm em bỏ ra, và mong muốn có thêm thời gian để hoàn thiện đồ án một cách tốt nhất.

II. Hướng phát triển

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

Phát triển phía client:

- Xây dựng thêm ứng dụng web, ứng dụng trên điện thoại.
- Nghiên cứu và thử nghiệm các công nghệ xây dựng giao diện khác để xây dựng một giao diện tốt hơn hiện tại.
- Thêm nhiều tính năng để người dùng sử dụng dễ dàng hơn.

Phát triển phía server:

- Thêm mô hình máy học để có thể gợi ý các bài hát cho client.
- Sửa đổi lại cơ sở dữ liệu hoàn thiện hơn.
- Tìm hiểu thêm các công cụ hay thư viện giúp cho việc truyền dữ liệu cho client nhanh hơn.
- Backup cơ sở dữ liệu và File Share.
- Tìm hiểu về cân bằng tải và áp dụng vào server.

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

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.

Bảng phân công công việc

STT	MSSV	Tên	Nội dung công việc phụ trách	Mức độ hoàn thành	Tự đánh giá (theo thang điểm 10)
1	21521929	Đặng Tiến Đạt	Xây dựng API cho server và xử lý bên phía client	Xây dựng API (90%) Xử lý phía client (40%)	8/10
2	21522304	Nguyễn Hoàng Long	Xây dựng CSDL, tải các file lên File Share, triển khai ứng dụng lên Azure, xây dựng API cho server, xử lý phía client	Xây dựng CSDL, tải lên các file nhạc-ảnh, triển khai ứng dụng(100%) Xây dựng API (90%) Xử lý phía client (50%)	8/10
3	21522305	Nguyễn Thái Long	Xử lý bên phía client, cải thiện giao diện đồ họa.	Xử lý phía client (60%) Cải thiện đồ họa (70%)	7/10
4	21522108	Nguyễn Văn Huân	Làm giao diện đồ họa, cung cấp các tệp tin cho File Share, thiết kế các sơ đồ của hệ thống, hỗ trợ xử lý phía client	Tạo giao diện đồ họa mới (100%) Cung cấp các tệp tin (70%) Thiết kế các sơ đồ (100%)	8/10

Tài liệu tham khảo

- [1] Vietnamnet, “Thị trường phát nhạc trực tuyến đang lên ngôi tại Việt Nam,” [Trực tuyến]. Available: <https://vietnamnet.vn/thi-truong-phat-nhac-truc-tuyen-dang-len-ngoi-tai-viet-nam-i29933.html>. [Đã truy cập 13 6 2023].
- [2] “What Is The OSI Model?,” [Trực tuyến]. Available: <https://firewall.firm.in/what-is-the-osi-model/>. [Đã truy cập 13 6 2023].
- [3] F. Cloud, “TCP/IP là gì? Toàn tập kiến thức về giao thức tcp/ip từ A-Z,” [Trực tuyến]. Available: <https://fptcloud.com/tcp-ip-la-gi/>. [Đã truy cập 13 6 2023].
- [4] H. Nguyễn, “Giao thức TCP (Transmission Control Protocol),” 1 14 2020. [Trực tuyến]. Available: <https://viblo.asia/p/tim-hieu-giao-thuc-tcp-va-udp-jvEla11x1kw>. [Đã truy cập 13 6 2023].
- [5] “What is HTTP?,” [Trực tuyến]. Available: <https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/>. [Đã truy cập 13 6 2023].
- [6] Microsoft, “Overview,” 10 4 2023. [Trực tuyến]. Available: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-tds/893fcc7e-8a39-4b3c-815a-773b7b982c50. [Đã truy cập 13 6 2023].
- [7] D. Guide, “What is SMB (Server Message Block)?,” 23 9 2020. [Trực tuyến]. Available: <https://www.ionos.com/digitalguide/server/know-how/server-message-block-smb/>. [Đã truy cập 13 6 2023].
- [8] Microsoft, “What is ASP.NET?,” [Trực tuyến]. Available: <https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet>. [Đã truy cập 13 6 2023].
- [9] “ASP.NET - Introduction,” [Trực tuyến]. Available: https://www.tutorialspoint.com/asp.net/asp.net_introduction.htm. [Đã truy cập 13 6 2023].
- [10] Microsoft, “What is the Azure portal?,” [Trực tuyến]. Available: <https://learn.microsoft.com/en-us/azure/azure-portal/azure-portal-overview>. [Đã truy cập 13 6 2023].

HẾT