

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



**XÂY DỰNG ỨNG DỤNG PHÁT AUDIO
VỚI PYTHON
PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ**

Người thực hiện: Nhóm 25

Phan Thanh Duy 3119410075

Trần Quang Vinh 3119410493

Vũ Đình Đại 3119410079

Giảng viên hướng dẫn:

Từ Lăng Phiêu

Thành phố Hồ Chí Minh, 2024.

LỜI CẢM ƠN

Em xin được gửi lời cảm ơn đến giảng viên Từ Lăng Phiêu đã tận tình giảng dạy và giúp đỡ em trong việc hoàn thành bài tập, cũng như hiểu vấn đề của môn học đề ra. Với những kiến thức em tích lũy được qua những ngày học tập, đây là kết quả của quá trình học tập của em. Tuy vẫn còn nhiều mặt còn hạn chế, nhưng em sẽ cố gắng để đạt được kết quả tốt nhất có thể.

Mục lục

1	Tóm tắt & giới thiệu	1
1.1	Giới thiệu đề tài	3
1.1.1	Vấn đề cần giải quyết	3
1.1.2	Mục tiêu	3
1.1.3	Giải pháp	3
1.1.4	Nội dung cần làm ở dự án	3
1.1.5	Tài nguyên hỗ trợ	3
1.2	Tìm hiểu về Python trong xây dựng ứng dụng	4
1.2.1	Giới thiệu về Python	4
1.2.2	Môi trường cài đặt và công cụ hỗ trợ	5
1.3	Tìm hiểu về 2 thư viện chính trong dự án này, cách thức hoạt động và áp dụng vào dự án	6
1.3.1	Giới thiệu về Thư viện Tkinter trong Python	6
1.3.2	Giới thiệu về Thư viện VLC trong Python cho Ứng dụng Phát Audio	7
1.4	Xây dựng mô hình	8
1.4.1	Sơ đồ ERD	8
1.4.2	Sơ đồ tiến trình hoạt động	9
2	Hiện thực hoá	11
2.1	Các thư viện sử dụng trong dự án	11
2.1.1	Phía Client	11
2.1.2	Phía Server	12
2.2	Chức năng	13
2.2.1	Giao diện server chứa bài hát	13
2.2.2	Giao diện server thêm xóa sửa bài hát	14
2.2.3	Giao diện người dùng phát bài hát	16
2.2.4	Giao diện khi chọn bài hát	17

2.2.5	Giao diện tìm kiếm bài hát	18
2.2.6	Giao diện thêm bài hát mới	19
2.2.7	Giao diện xóa bài hát	19
2.3	Đánh giá	19
2.3.1	Ưu nhược điểm	19
2.3.2	Bài học nhận được	20
2.3.3	Phương hướng phát triển	20

Chương 1

Tóm tắt & giới thiệu

1. Mục Tiêu - Mục Tiêu Chính: Xây dựng một ứng dụng phát audio đơn giản nhưng mạnh mẽ bằng ngôn ngữ lập trình Python.

- Mục Tiêu Phụ:

- Hiểu cách sử dụng thư viện âm thanh và đa phương tiện của Python như PyDub, PyAudio.

- Xây dựng giao diện người dùng đơn giản và dễ sử dụng.

- Tích hợp các tính năng cơ bản như play, pause, stop, và điều chỉnh âm lượng. **2.**

Phân Tích Yêu Cầu - Yêu Cầu Hệ Thống:

- Python cài đặt trên máy tính.

- Các thư viện cần thiết như PyDub, PyAudio.

- Yêu Cầu Người Dùng:

- Giao diện thân thiện và dễ sử dụng.

- Chức năng phát, tạm dừng, và dừng lại một tệp âm thanh.

- Thay đổi âm lượng.

3. Thiết Kế Ứng Dụng - Thiết Kế Giao Diện Người Dùng (GUI): Sử dụng thư viện Tkinter để tạo giao diện đồ họa.

- Thiết Kế Backend: Sử dụng PyDub để xử lý và phát tệp âm thanh, và PyAudio để điều khiển âm lượng.

4. Triển Khai và Kiểm Thử - Triển Khai Ứng Dụng: Chạy và kiểm thử ứng dụng trên nhiều nền tảng hệ điều hành.

- Kiểm Thử Tính Năng: Đảm bảo các tính năng như play, pause, stop, và điều chỉnh âm lượng hoạt động một cách chính xác và ổn định. **5. Kết Luận và Hướng Phát**

Triển Tiếp Theo - Kết Luận: Thành công trong xây dựng một ứng dụng phát audio đơn giản nhưng hiệu quả bằng Python.

- Hướng Phát Triển Tiếp Theo:

- Tích hợp tính năng mở và phát các định dạng tệp âm thanh khác nhau.
- Cải thiện giao diện và trải nghiệm người dùng.
- Phát triển tính năng tìm kiếm và quản lý thư viện âm nhạc. **6. Tài Nguyên Tham Khảo** - [PyDub Documentation](<https://github.com/jiaaro/pydub>)
- [PyAudio Documentation](<https://people.csail.mit.edu/hubert/pyaudio/docs/>)
- [Tkinter Documentation](<https://docs.python.org/3/library/tkinter.html>)

1.1 Giới thiệu đề tài

1.1.1 Vấn đề cần giải quyết

Trong thế giới số hóa ngày nay, nhu cầu sử dụng ứng dụng phát audio ngày càng tăng cao. Tuy nhiên, một số ứng dụng phát audio hiện có thường có giao diện phức tạp hoặc không linh hoạt đối với người dùng. Nhiều người dùng cảm thấy khó khăn trong việc tìm hiểu cách sử dụng và tùy chỉnh các tính năng của ứng dụng để phù hợp với nhu cầu cá nhân của họ. Điều này tạo ra một nhu cầu rõ ràng cho một ứng dụng phát audio đơn giản, dễ sử dụng và linh hoạt.

1.1.2 Mục tiêu

Mục tiêu của dự án là xây dựng một ứng dụng phát audio đơn giản nhưng mạnh mẽ, giúp người dùng dễ dàng phát các file âm thanh mà không gặp phải sự phức tạp từ giao diện. Ứng dụng sẽ tập trung vào việc cung cấp các chức năng cơ bản như play, pause, stop, và điều chỉnh âm lượng một cách dễ dàng và trực quan.

1.1.3 Giải pháp

Để đạt được mục tiêu này, chúng tôi sẽ sử dụng các thư viện Python mạnh mẽ như PyDub và PyAudio. PyDub sẽ được sử dụng để xử lý và phát các file âm thanh, trong khi PyAudio sẽ được sử dụng để điều khiển và điều chỉnh âm lượng. Bằng cách này, chúng tôi sẽ tạo ra một ứng dụng phát audio mạnh mẽ và dễ sử dụng cho người dùng.

1.1.4 Nội dung cần làm ở dự án

Các thành phần đóng góp vào dự án bao gồm: Thiết kế và phát triển giao diện người dùng dễ sử dụng và thân thiện. Xây dựng các chức năng chính của ứng dụng như play, pause, stop. Đảm bảo tính ổn định và hiệu suất của ứng dụng thông qua kiểm thử và sửa lỗi.

1.1.5 Tài nguyên hỗ trợ

Chúng tôi sẽ sử dụng các tài nguyên hỗ trợ như: Tài liệu hướng dẫn sử dụng PyDub và PyAudio. Tài liệu tham khảo và ví dụ về thiết kế giao diện người dùng bằng Python. Cộng đồng Python và các nguồn tài nguyên trực tuyến khác để giải quyết vấn đề và tìm kiếm giải pháp cho các thách thức trong quá trình phát triển.

1.2 Tìm hiểu về Python trong xây dựng ứng dụng

1.2.1 Giới thiệu về Python

Cấu trúc của Python là cách mà các thành phần của ngôn ngữ được tổ chức và kết hợp lại với nhau để tạo thành các chương trình hoặc dự án Python. Dưới đây là một cái nhìn tổng quan về cấu trúc của Python:

- Dòng lệnh (Statements):
 - Python sử dụng dấu gạch chéo (‘) để chỉ định một dòng lệnh nằm trên nhiều dòng.
 - Mỗi dòng lệnh thường kết thúc bằng dấu chấm phẩy (;) hoặc không cần dấu chấm phẩy nếu mỗi dòng lệnh được đặt trên một dòng riêng biệt.
- Khối mã (Code Blocks):
 - Python sử dụng dấu thụt lề để phân biệt các khối mã (code blocks). - Các khối mã thường được định nghĩa bằng các cặp dấu mở ngoặc và đóng ngoặc, ví dụ: ‘if’, ‘else’, ‘for’, ‘while’, ‘def’ (định nghĩa hàm), và các cấu trúc điều khiển khác.
- Chú thích (Comments):
 - Python cho phép sử dụng ký hiệu “ để viết chú thích trên cùng một dòng lệnh. - Đối với các chú thích dài hơn, bạn có thể sử dụng dấu ngoặc kép ba lần """ hoặc """ để bao quanh văn bản chú thích.
- Biến và Kiểu dữ liệu (Variables and Data Types):
 - Biến trong Python không cần khai báo kiểu dữ liệu và có thể thay đổi kiểu dữ liệu trong quá trình thực thi.
 - Một số kiểu dữ liệu cơ bản trong Python bao gồm: số nguyên (int), số thực (float), chuỗi (string), danh sách (list), bộ (tuple), từ điển (dictionary), và nhiều kiểu dữ liệu khác.
- Hàm (Functions):
 - Hàm trong Python được định nghĩa bằng từ khóa ‘def’, sau đó là tên hàm và danh sách các tham số (nếu có).
 - Một hàm có thể trả về một giá trị bằng từ khóa ‘return’, hoặc không trả về gì bằng cách không có câu lệnh ‘return’.
- Modules (Modules):
 - Python cho phép bạn phân chia mã của mình thành các modules, mỗi module là một tập hợp các định nghĩa hàm, lớp và biến.
 - Để sử dụng một module trong mã của mình, bạn có thể sử dụng từ khóa ‘import’.
- Lớp và Đối tượng (Classes and Objects):
 - Python là một ngôn ngữ lập trình hướng đối tượng, vì vậy nó hỗ trợ việc định nghĩa

và sử dụng lớp và đối tượng.

- Lớp được định nghĩa bằng từ khóa ‘class’, trong đó có các phương thức (method) và thuộc tính (attribute).

- Để tạo một đối tượng từ một lớp, bạn sử dụng cú pháp ‘tên-lớp()’.

Đó là một cái nhìn tổng quan về cấu trúc của Python. Cấu trúc này cung cấp một cơ sở vững chắc để bạn bắt đầu xây dựng các chương trình Python phức tạp hơn. Cấu trúc của Python là cách mà các thành phần của ngôn ngữ được tổ chức và kết hợp lại với nhau để tạo thành các chương trình hoặc dự án Python. Dưới đây là một cái nhìn tổng quan về cấu trúc của Python:

1.2.2 Môi trường cài đặt và công cụ hỗ trợ

Để phát triển các dự án Python một cách hiệu quả, bạn cần thiết lập một môi trường phát triển Python và cài đặt các công cụ hỗ trợ cần thiết. Dưới đây là các bước cơ bản để thiết lập môi trường và cài đặt các công cụ:

- Cài đặt Python - Đầu tiên, bạn cần cài đặt Python trên máy tính của mình. Bạn có thể tải xuống bản cài đặt Python mới nhất từ trang web chính thức của Python (<https://www.python.org/downloads/>) và làm theo hướng dẫn để cài đặt.

- Quản lý Môi trường ảo (Virtual Environment):

- Môi trường ảo là một cách để tạo ra một môi trường phát triển Python cô lập, độc lập với môi trường Python toàn cục trên hệ thống của bạn. Điều này giúp bạn quản lý các dependencies và phiên bản của các gói Python cho từng dự án một cách dễ dàng.

- Bạn có thể sử dụng công cụ ‘virtualenv’ hoặc ‘venv’ (được tích hợp sẵn trong Python 3) để tạo một môi trường ảo. Ví dụ:

```
““ python -m venv myenv ““
```

- Kích hoạt Môi trường ảo:

- Sau khi tạo một môi trường ảo, bạn cần kích hoạt nó trước khi bắt đầu làm việc trên dự án. Trên Windows, bạn có thể sử dụng:

```
““ myenv““ Trên macOS và Linux: ““ source myenv/bin/activate ““
```

- Cài đặt Công cụ quản lý gói (Package Manager):

- Công cụ quản lý gói giúp bạn quản lý các dependencies của dự án Python. ‘pip’ là công cụ quản lý gói phổ biến nhất trong cộng đồng Python và được sử dụng để cài đặt, cập nhật và gỡ cài đặt các gói Python.

- Bạn không cần phải cài đặt pip một cách riêng lẻ vì nó được cài đặt tự động khi bạn

cài đặt Python.

- Cài đặt Công cụ Hỗ trợ Phát triển (Development Tools):
- Một số công cụ phát triển hữu ích cho Python bao gồm:
- IDEs (Integrated Development Environments): Như PyCharm, Visual Studio Code, Sublime Text, và IDLE (phần mềm tích hợp sẵn trong Python).
- Editors và Plugins: Như Vim, Emacs, và các plugins cho các editors như LSP (Language Server Protocol) để hỗ trợ phát triển Python.
- Debugger: Ví dụ như PDB (Python Debugger), hoặc các tính năng debugger tích hợp trong các IDE như PyCharm.

- Quản lý dependencies của Dự án:
 - Khi bạn làm việc trên một dự án Python, bạn cần quản lý các dependencies của nó. Bạn có thể sử dụng một tập tin ‘requirements.txt’ để liệt kê các gói cần thiết và sử dụng pip để cài đặt chúng.
- ```
““ pip install -r requirements.txt ““
```

Với các bước trên, bạn sẽ có một môi trường phát triển Python sạch sẽ và sẵn sàng để bắt đầu phát triển các dự án Python của mình.

## **1.3 Tìm hiểu về 2 thư viện chính trong dự án này, cách thức hoạt động và áp dụng vào dự án**

### **1.3.1 Giới thiệu về Thư viện Tkinter trong Python**

Tkinter là một thư viện GUI (Graphical User Interface - Giao diện người dùng đồ họa) được tích hợp sẵn trong ngôn ngữ lập trình Python. Thư viện này cung cấp các công cụ và tài nguyên cho việc phát triển các ứng dụng có giao diện người dùng trực quan trên nhiều nền tảng hệ điều hành.

Đặc điểm chính của Tkinter:

- Dễ học và sử dụng: Tkinter cung cấp một giao diện dễ hiểu và sử dụng, làm cho việc phát triển ứng dụng GUI trở nên đơn giản đối với người mới bắt đầu.
- Tích hợp sẵn trong Python: Không cần phải cài đặt bất kỳ thư viện bổ sung nào, Tkinter đi kèm với Python mà không gây ra sự phức tạp trong quá trình cài đặt.
- Hỗ trợ đa nền tảng: Ứng dụng được phát triển bằng Tkinter có thể chạy trên nhiều

hệ điều hành phổ biến như Windows, macOS và Linux.

- Phong phú trong tài liệu và nguồn lực học tập: Tkinter có sẵn nhiều tài liệu và ví dụ trực tuyến giúp người dùng dễ dàng học và sử dụng.

Mặc dù Tkinter không có nhiều tính năng như các thư viện GUI mạnh mẽ khác như PyQt hoặc wxPython, nhưng với tính đơn giản và tích hợp sẵn trong Python, nó vẫn là một công cụ phổ biến và hữu ích cho việc phát triển các ứng dụng GUI trong Python.

### 1.3.2 Giới thiệu về Thư viện VLC trong Python cho Ứng dụng Phát Audio

Thư viện VLC (LibVLC) là một thư viện phổ biến và mạnh mẽ được sử dụng cho việc phát các định dạng phương tiện đa phương tiện khác nhau, bao gồm âm thanh và video. Trong Python, bạn có thể sử dụng thư viện python-vlc để tích hợp các tính năng phát audio từ VLC vào dự án của mình.

Cách sử dụng thư viện python-vlc trong dự án Python:

Cài đặt thư viện python-vlc: Bạn cần cài đặt thư viện python-vlc bằng cách sử dụng pip, một công cụ quản lý gói Python. Bạn có thể cài đặt nó bằng lệnh sau trong terminal:// — pip install python-vlc

Tích hợp thư viện vào dự án của bạn: Sau khi cài đặt xong, bạn có thể bắt đầu sử dụng thư viện trong mã Python của mình. Dưới đây là một ví dụ đơn giản về cách sử dụng thư viện python-vlc để phát một file âm thanh:

```
import vlc
```

Khởi tạo một instance của VLC media player

```
player = vlc.MediaPlayer()
```

Load file âm thanh cần phát

```
media = vlc.Media("path/to/your/audio/file.mp3")
```

Set media cho player

```
player.set-media(media)
```

Phát media

```
player.play()
```

Dừng phát sau 10 giây

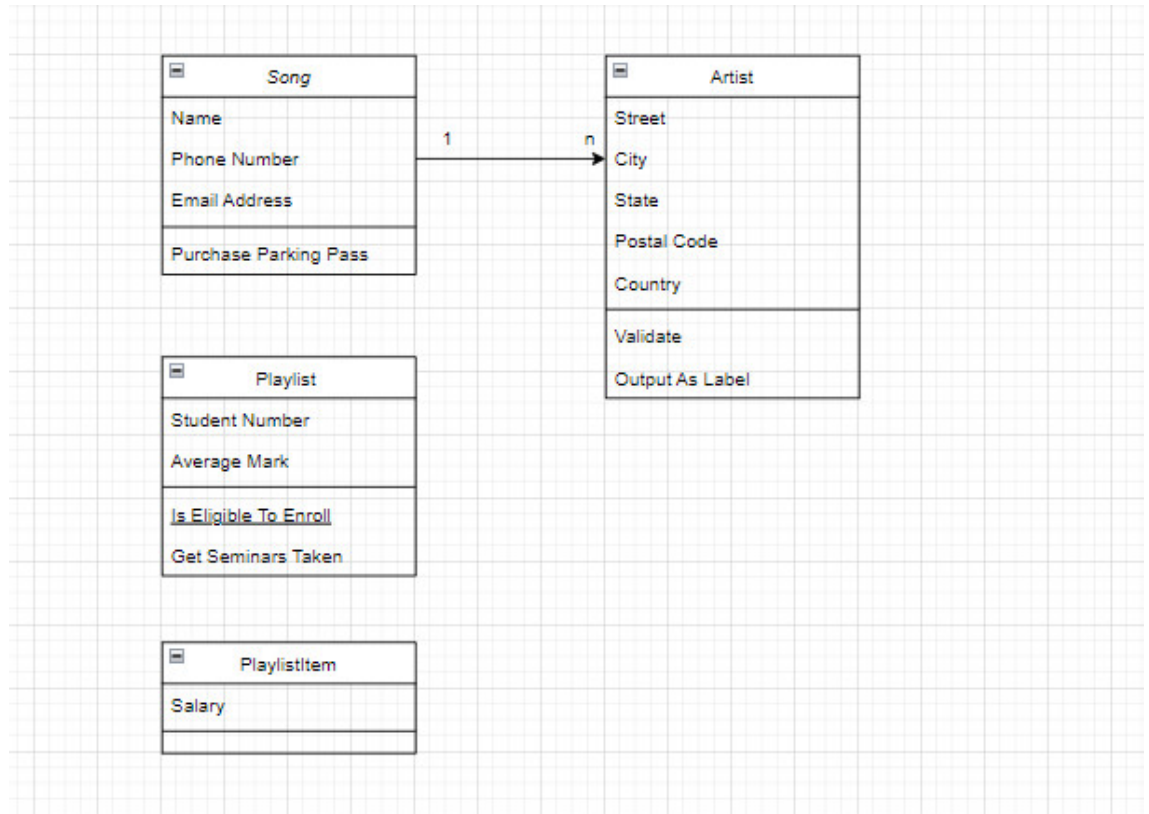
```
import time
```

```
time.sleep(10)
```

```
player.stop()
```

## 1.4 Xây dựng mô hình

### 1.4.1 Sơ đồ ERD



Thực thể Song đại diện cho mỗi bài hát trong hệ thống, với mỗi bài hát có một SongID là khóa chính (Primary Key) để xác định bài hát đó.

Mỗi bài hát cũng có một Title, ArtistID là khóa ngoại (Foreign Key) trỏ tới nghệ sĩ tạo ra bài hát đó, FilePath để lưu trữ đường dẫn đến file âm thanh, và Duration để lưu trữ thời lượng của bài hát

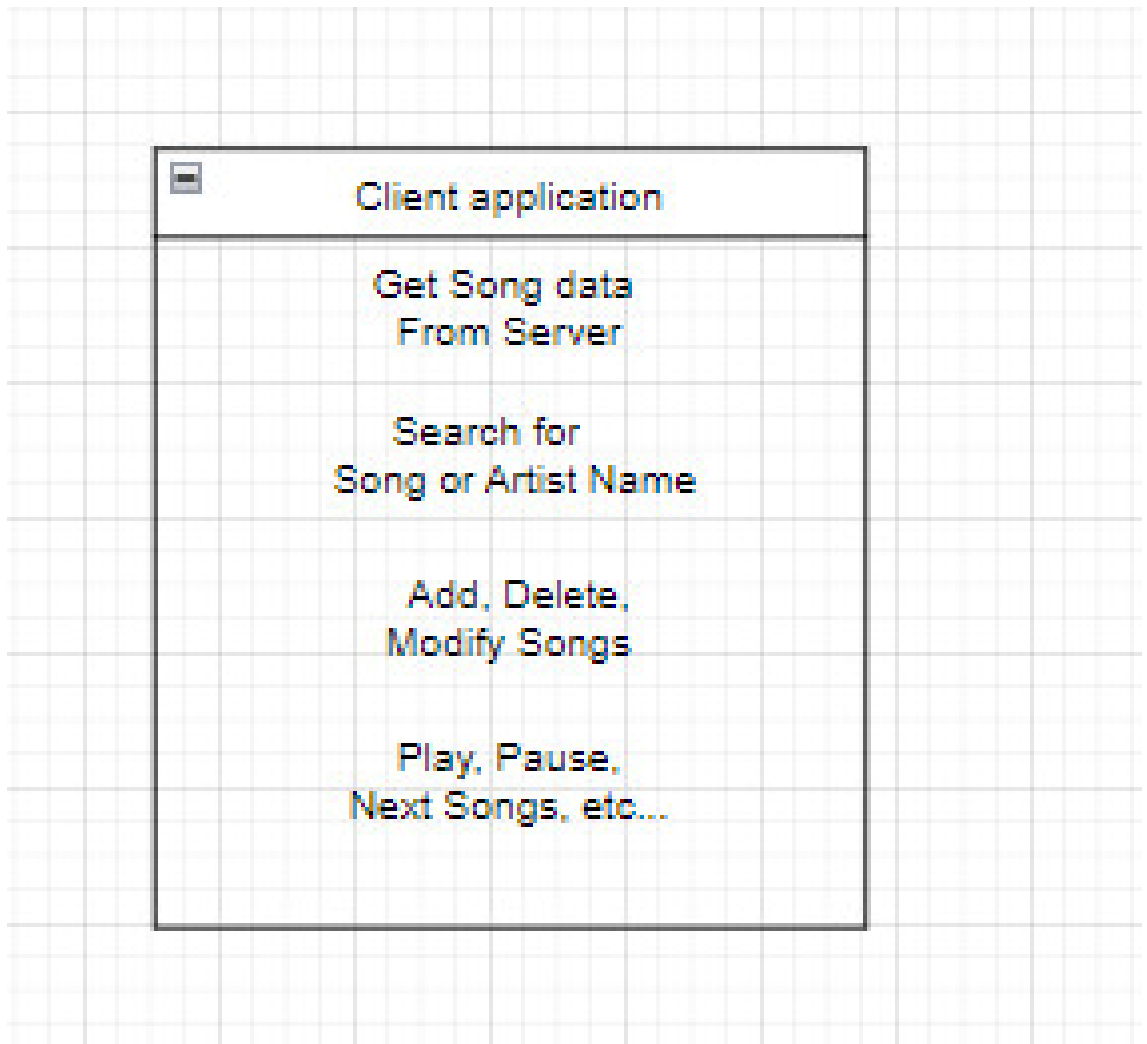
Thực thể Artist đại diện cho mỗi nghệ sĩ, với mỗi nghệ sĩ có một ArtistID là khóa chính để xác định nghệ sĩ đó. Mỗi nghệ sĩ cũng có một Name để xác định tên của nghệ sĩ.

Thực thể Playlist đại diện cho danh sách phát, với mỗi danh sách phát có một PlaylistID là khóa chính để xác định danh sách phát đó. Mỗi danh sách phát cũng có một Name và Description để mô tả danh sách phát đó.

Thực thể PlaylistItem đại diện cho mỗi mục trong danh sách phát, với mỗi mục có một PlaylistItemID là khóa chính để xác định mục đó. Mỗi mục cũng có một PlaylistID

là khóa ngoại trỏ tới danh sách phát mà nó thuộc về, một SongID là khóa ngoại trỏ tới bài hát mà mục đó đại diện cho, và một Position để xác định vị trí của mục trong danh sách phát.

#### 1.4.2 Sơ đồ tiến trình hoạt động



Lấy dữ liệu bài hát từ máy chủ: Ứng dụng khách có khả năng gửi yêu cầu đến máy chủ để lấy dữ liệu về các bài hát.

Tìm kiếm theo tên bài hát và tên ca sĩ: Người dùng có thể tìm kiếm bài hát hoặc ca sĩ bằng cách nhập tên vào ứng dụng khách.

Thêm, xóa, sửa bài hát: Ứng dụng khách cho phép người dùng thêm mới, xóa hoặc chỉnh sửa thông tin của các bài hát trong danh sách.

Chức năng phát nhạc: Người dùng có thể phát, tạm dừng, chuyển bài hát tiếp theo hoặc trở lại bài hát trước đó từ danh sách bài hát đã lấy từ máy chủ.

Những hoạt động này được thực hiện trên ứng dụng khách một cách tuần tự và tương tác với người dùng thông qua giao diện người dùng của ứng dụng.

# Chương 2

## Hiện thực hoá

### 2.1 Các thư viện sử dụng trong dự án

#### 2.1.1 Phía Client

Dưới đây là danh sách các thư viện được sử dụng ở phía Client:

- ‘vlc’: Thư viện VLC Python, cho phép bạn tương tác với trình phát đa phương tiện VLC từ Python.
- ‘time’: Thư viện thời gian tiêu chuẩn của Python, cung cấp các hàm cho việc xử lý và đo thời gian.
- ‘tkinter’: Thư viện giao diện người dùng (GUI) cơ bản của Python, cung cấp các công cụ để tạo và quản lý cửa sổ, widget và sự kiện GUI.
- ‘filedialog’: Mô-đun con của tkinter, cung cấp các hộp thoại để chọn và lưu tệp.
- ‘PIL’: Python Imaging Library, thư viện xử lý hình ảnh trong Python.
- ‘os’: Thư viện hệ điều hành của Python, cung cấp các hàm để tương tác với hệ điều hành, ví dụ như tạo và xóa thư mục, điều chỉnh đường dẫn, v.v.
- ‘requests’: Thư viện HTTP cho Python, cho phép bạn gửi các yêu cầu HTTP đến máy chủ và xử lý phản hồi.
- ‘Gradian’: Một module tự tạo, có thể là một phần của dự án của bạn, không có thông tin cụ thể được cung cấp về chức năng của nó.
- ‘Constants’: Một module tự tạo, có thể là một phần của dự án của bạn, không có thông tin cụ thể được cung cấp về chức năng của nó.
- ‘urlopen’: Một hàm từ mô-đun ‘urllib.request’ cho phép mở một URL và lấy dữ liệu từ nó.
- ‘json’: Thư viện JSON cho Python, cung cấp các hàm để chuyển đổi giữa các đối tượng Python và định dạng JSON.

### 2.1.2 Phía Server






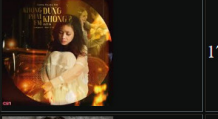


Dưới đây là danh sách các thư viện được sử dụng ở phía Server:

- Flask: Flask là một framework web cho Python, được sử dụng để xây dựng các ứng dụng web và API. Nó cung cấp một cách tiếp cận dễ dàng để xử lý các yêu cầu HTTP và tạo các trang web động.
- jsonify: Flask cung cấp hàm jsonify để tạo ra một phản hồi JSON từ một đối tượng Python.
- request: Module request của Flask được sử dụng để truy cập thông tin của yêu cầu HTTP, như dữ liệu form hoặc tham số truy vấn.
- render\_template: Flask cho phép tạo và render các mẫu HTML, cho phép chúng ta tạo các trang web động.
- "sendfile": sendfile được sử dụng để gửi một file dưới dạng phản hồi từ máy chủ đến trình duyệt của người dùng.
- Response: Response là một lớp được sử dụng để tạo ra một phản hồi HTTP tùy chỉnh từ Flask.
- Flask-Restful: Flask-Restful là một extension cho Flask giúp viết các API RESTful một cách dễ dàng hơn bằng cách sử dụng các class Python thường.
- reqparse: reqparse là một công cụ trong Flask-Restful được sử dụng để phân tích và kiểm tra dữ liệu yêu cầu.
- os: Module os cung cấp các chức năng để tương tác với hệ thống tệp và môi trường hệ thống.
- . Store (services.db.connect): Store có thể là một module tự tạo trong dự án của bạn, có thể liên quan đến kết nối và tương tác với cơ sở dữ liệu.
- datetime: datetime là một module trong Python cung cấp các lớp và hàm để làm việc với thời gian và ngày tháng. Trong trường hợp này, nó có thể được sử dụng để xử lý dữ liệu thời gian trong ứng dụng của bạn.



## 2.2 Chức năng

### 2.2.1 Giao diện server chứa bài hát

| List of Songs |                           |                                                                                     |                       |                        |  |
|---------------|---------------------------|-------------------------------------------------------------------------------------|-----------------------|------------------------|--|
| ID            | Name                      | Image                                                                               | Path                  | Artists                |  |
| 21            | Đứng người đứng thời điểm |    | 1714128850.920477.mp3 | Thanh Hưng             |  |
| 22            | Như Người Xa Lạ           |    | 1714129233.024552.mp3 | Vương Anh Tú           |  |
| 23            | Màu Nước Mắt              |    | 1714129273.797331.mp3 | Nguyễn Trần Trung Quân |  |
| 24            | Càng Niu Giữ Càng Dễ Mất  |   | 1714129360.251528.mp3 | Mr Siro                |  |
| 24            | Càng Niu Giữ Càng Dễ Mất  |  | 1714129360.251528.mp3 | Mr Siro                |  |
| 25            | Không Phải Em Đứng Không  |  | 1714129417.788314.mp3 | Dương Hoàng Yến        |  |
| 26            | Từ Khi Gặp Em             |  | 1714129466.011177.mp3 | Trịnh Thăng Bình       |  |
| 27            | Bán Duyên                 |  | 1714129569.742814.mp3 | Đinh Dũng              |  |

Trong giao diện này Server có nhiệm vụ chứa các bài hát và phía client sẽ lấy danh sách bài hát từ đây để hiển thị và phát.

## 2.2.2 Giao diện server thêm xóa sửa bài hát

### List of Songs

| ID | Name                      | Image | Path                  | Artists          |      |     |
|----|---------------------------|-------|-----------------------|------------------|------|-----|
| 38 | Đứng người đứng thời điểm |       | 1715174405.289592.mp3 | Thanh Hưng       | Edit | Xóa |
| 44 | Bài mới                   |       | 1715176629.150387.mp3 | Tác giả          | Edit | Xóa |
| 46 | Anh Ta Là Sao             |       | 1715178483.741297.mp3 | Erik             | Edit | Xóa |
| 26 | Từ Khi Gặp Em             |       | 1714129466.011177.mp3 | Trịnh Thăng Bình | Edit | Xóa |

cốc cốc

Bot Store - Coze

Coze

Copilot - Coze AI Bot

Codeman - Coze AI Bo

List of Songs

← → ↺

127.0.0.1:5000/list-songs

A Cổng Thông Tin Đào...

A Cổng Thông Tin Đào...

(1) Facebook

T

🕒

AI

📱

📧

+

📺

📺

📺

📘

🎵

### List of Songs

| ID | Name                      | Image |                       |            |      |     |
|----|---------------------------|-------|-----------------------|------------|------|-----|
| 38 | Đứng người đứng thời điểm |       | 1715174405.289592.mp3 | Thanh Hưng | Edit | Xóa |
| 44 | Ngọt Ngào Bên Em          |       | 1715176629.150387.mp3 | Tác giả    | Edit | Xóa |
| 46 | Anh Ta Là Sao             |       | 1715178483.741297.mp3 | Erik       | Edit | Xóa |
|    |                           |       |                       |            |      |     |

127.0.0.1:5000 cho biết

Bạn có chắc muốn xóa bài hát này không?

OK

Hủy

g Thông Tin Đào...
A Cổng Thông Tin Đào...
(1) Facebook
T

127.0.0.1:5000 cho biết

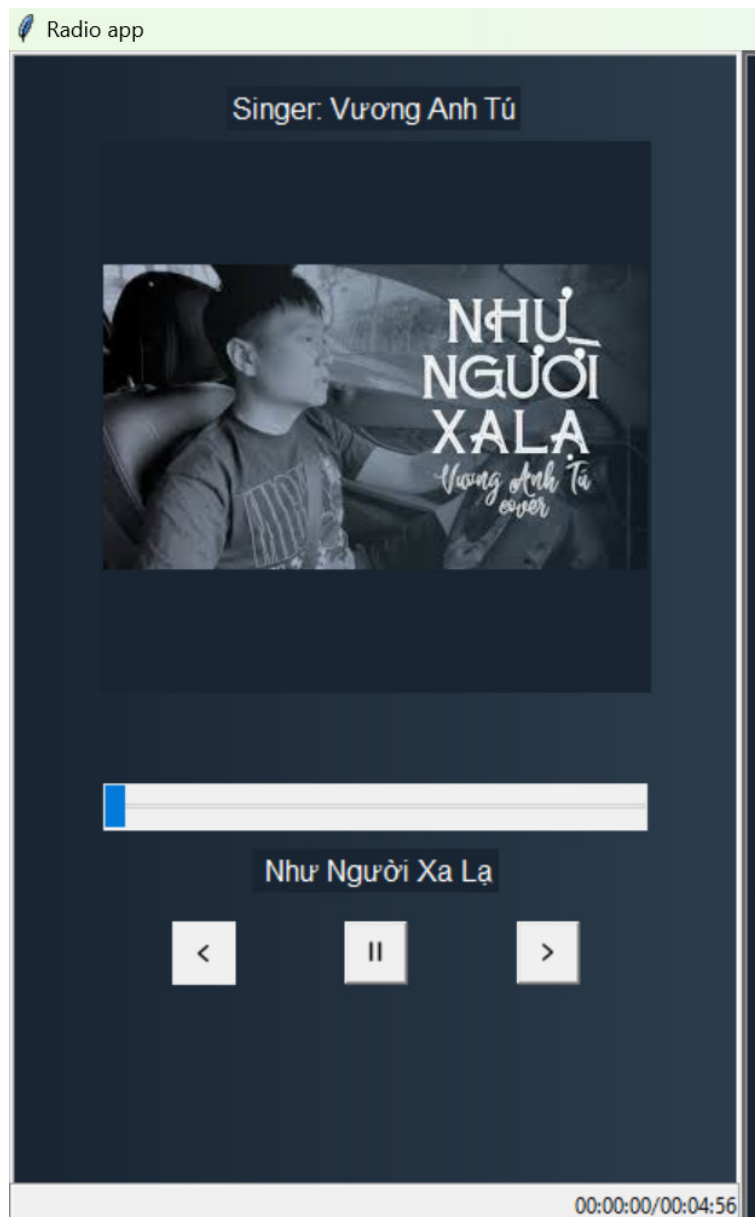
Bài hát đã được xóa thành công

OK

# List of Songs

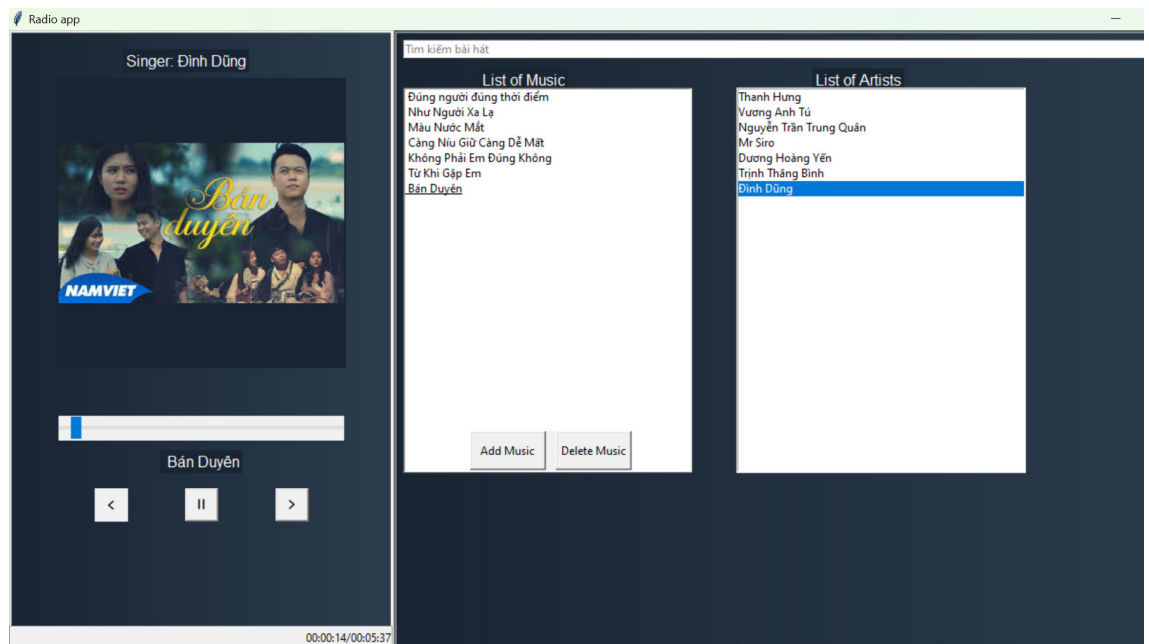
| ID | Name                      | Image |                       |                        |      |     |
|----|---------------------------|-------|-----------------------|------------------------|------|-----|
| 38 | Đúng người đúng thời điểm |       | 1715174405.289592.mp3 | Thanh Hưng             | Edit | Xóa |
| 44 | Ngọt Ngào Bên Em          |       | 1715176629.150387.mp3 | Tác giả                | Edit | Xóa |
| 26 | Từ Khi Gặp Em             |       | 1714129466.011177.mp3 | Trịnh Thăng Bình       | Edit | Xóa |
| 27 | Bán Duyên                 |       | 1714129569.742814.mp3 | Đình Dũng              | Edit | Xóa |
| 22 | Như Người Xa Lạ           |       | 1714129233.024552.mp3 | Vương Anh Tú           | Edit | Xóa |
| 23 | Màu Nước Mắt              |       | 1714129273.797331.mp3 | Nguyễn Trần Trung Quân | Edit | Xóa |
| 24 | Càng Niu Giữ Càng Dễ Mất  |       | 1714129360.251528.mp3 | Mr Siro                | Edit | Xóa |

### 2.2.3 Giao diện người dùng phát bài hát



Trong giao diện này, khi người dùng phát nhạc thì thời lượng bài hát được phát theo thời gian sẽ được hiển thị trên thanh seekbar như ảnh.

## 2.2.4 Giao diện khi chọn bài hát



Sau khi người dùng nhấp vào bài hát mình muốn phát thì phía bên trái màn ảnh sẽ hiển thị lên hình ảnh bìa và sẽ xuất hiện cái nút play stop next ở phía dưới.

### 2.2.5 Giao diện tìm kiếm bài hát

The image displays two screenshots of a web application interface for searching music. The interface is divided into a header, a search bar, and two main panels.

**Top Screenshot:**

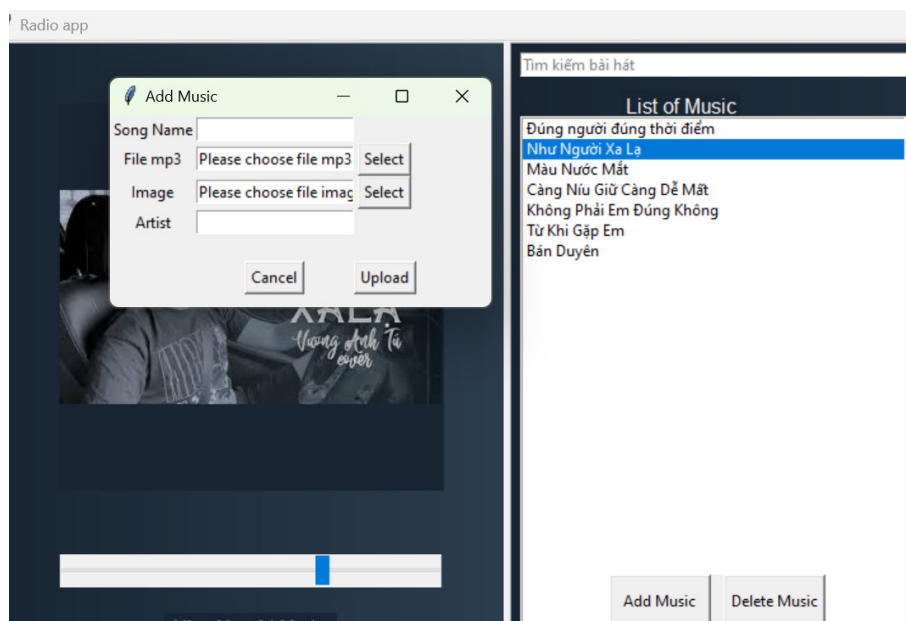
- Search Bar:** Contains the text "Bản Duyệt".
- List of Music:** Contains the text "Bản Duyệt".
- List of Artists:** Contains the text "Đinh Dũng".
- Buttons:** "Add Music" and "Delete Music" are located at the bottom of the "List of Music" panel.

**Bottom Screenshot:**

- Search Bar:** Contains the text "Đinh Dũng".
- List of Music:** Contains the text "Bản Duyệt".
- List of Artists:** Contains the text "Đinh Dũng".
- Buttons:** "Add Music" and "Delete Music" are located at the bottom of the "List of Music" panel.

Ở giao diện này, khi người dùng muốn tìm kiếm một bài hát mong muốn thì sẽ nhập tên bài hát vào ô tìm kiếm. Nếu người dùng quên tên bài hát thì có thể nhập vào tên ca sĩ (tác giả) để tìm kiếm thì sẽ hiển thị bài hát đó.

## 2.2.6 Giao diện thêm bài hát mới



Ở phần này có thể thêm vào bài hát mới bằng cách chọn vào 1 file audio bài hát, sau đó người dùng cần chọn ảnh và đặt tên cho bài hát cũng như thêm tên tác giả vào.

## 2.2.7 Giao diện xóa bài hát

Ở giao diện này người dùng muốn xóa bài hát nào thì nhấp vào bài hát đó và nhấp Delete Music thì ứng dụng sẽ hỏi có muốn xóa bài hát không thì bấm có là hoàn thành xóa bài hát.

## 2.3 Đánh giá

### 2.3.1 Ưu nhược điểm

#### Điểm mạnh

Ứng dụng phát audio đã được xây dựng với giao diện người dùng đơn giản và dễ sử dụng, giúp người dùng dễ dàng tương tác và thao tác với các chức năng. Tích hợp các tính năng cơ bản như play, pause, stop đáp ứng được nhu cầu cơ bản của người dùng.

#### Điểm yếu

Hiện vẫn còn một số hạn chế về khả năng mở rộng và tính năng của ứng dụng. Giao diện người dùng còn đơn giản và có thể cải thiện để tạo ra trải nghiệm người

dùng tốt hơn. Hiệu suất và tương thích có thể cần được tối ưu hóa để đảm bảo hoạt động mượt mà trên các nền tảng và môi trường khác nhau.

### **2.3.2 Bài học nhận được**

Trong quá trình phát triển dự án, chúng tôi đã học được nhiều điều quan trọng như:

Quan trọng của việc thiết kế giao diện người dùng đơn giản và dễ sử dụng. Phát triển các tính năng cơ bản trước, sau đó tăng cường và mở rộng chúng dựa trên phản hồi từ người dùng. Quản lý và tối ưu hóa tài nguyên để đảm bảo hiệu suất và tương thích tốt nhất.

### **2.3.3 Phương hướng phát triển**

Để cải thiện và phát triển ứng dụng phát audio, chúng tôi đề xuất một số phương hướng sau:

Tích hợp các tính năng nâng cao như tìm kiếm, tạo danh sách phát, và điều chỉnh cài đặt âm thanh. Cải thiện giao diện người dùng để tạo ra trải nghiệm người dùng thú vị hơn và dễ sử dụng hơn. Kiểm tra và tối ưu hóa hiệu suất của ứng dụng để đảm bảo hoạt động mượt mà trên các nền tảng và môi trường khác nhau. Tích hợp tính năng đồng bộ hóa với các dịch vụ lưu trữ và phát nhạc trực tuyến để mở rộng phạm vi và tiện ích của ứng dụng.



# Tài liệu tham khảo

- [1] PyDub Documentation
- [2] PyAudio Documentation
- [3] Tkinter Documentation
- [4] Flask Documentation