**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**---------------------------------------------------------**

**BÁO CÁO THỰC HÀNH**

**Môn: Lập trình trực quan (IT008)**

**Đề tài: Trình phát Audio và Video**



**GVHD:** Nguyễn Tấn Toàn

**Nhóm:** 15

21520147 – Phạm Tuấn Anh

21520579 – Nguyễn Tiến Anh

21521682 – Nguyễn Văn Vũ

21521055 – Nguyễn Chí Lâm

**LỜI MỞ ĐẦU**

Lời đầu tiên, em xin chân thành gửi lời cảm ơn đến thầy Nguyễn Tấn Toàn, người đã nhiệt tình giúp đỡ, hướng dẫn và định hướng cho em trong suốt quá trình thực hiện đồ án này.

Cuối cùng, vì còn hạn chế về năng lực và kinh nghiệm làm phần mềm, sản phẩm của tụi em khó có thể tránh khỏi những thiếu sót. Kính mong nhận được những góp ý của thầy.

**Mục lục**

[**Chương 1. Tổng quan 5**](#_Toc125389186)

[**1.1. Giới thiệu đề tài đồ án 5**](#_Toc125389187)

[**1.2. Giới thiệu bài toán cần giải quyết 5**](#_Toc125389188)

[**1.3. Quy trình thực hiện 5**](#_Toc125389189)

[**1.4. Một số chức năng nổi bật 5**](#_Toc125389190)

[**Chương 2. Phân Tích Yêu Cầu Phần Mềm Và Mô Hình Hóa 6**](#_Toc125389191)

[**2.1. Yêu cầu phần mềm 6**](#_Toc125389192)

[**2.1.1. Yêu cầu chức năng 6**](#_Toc125389193)

[**2.1.2 Yêu cầu phi chức năng 6**](#_Toc125389194)

[**2.1.3 Bảng FURPS 6**](#_Toc125389195)

[**2.2. Mô hình hóa 7**](#_Toc125389196)

[**2.2.1 Sơ đồ luồng dữ liệu 7**](#_Toc125389197)

[**2.2.2. Sơ đồ các Class 8**](#_Toc125389198)

[**Chương 3. Kỹ thuật xử lí 22**](#_Toc125389199)

[**3.1 Phân tích 22**](#_Toc125389200)

[**3.1.1 Phân tích hiệu ứng trực quan 22**](#_Toc125389201)

[**3.1.2. Phân tích xử lí chức năng 23**](#_Toc125389202)

[**3.2. Kỹ thuật sử dụng 25**](#_Toc125389203)

[**3.2.1. Xây dụng class MediaController như bộ điều khiển trung tâm của ứng dụng 25**](#_Toc125389204)

[**3.2.2. Xây dựng các interface IMediaItem, IMediaItemList, abstract class AMediaItemList giúp trừu tượng hóa các class TrackItem, TrackItemList, VideoItem, VideoItemList 32**](#_Toc125389205)

[**3.2.3. Xây dựng các interface IIntegration, IIntegrationSearchResultList giúp trừu tượng hóa các class ZingMP3Integration, ZingMP3IntegrationSearchResultList 38**](#_Toc125389206)

[**3.2.4. Sử dụng class FileSystemWatcher để thực hiện động bộ hóa dữ liệu 40**](#_Toc125389207)

[**Chương 4. Thiết kế giao diện 43**](#_Toc125389208)

[**Chương 5. Cài đặt và kiểm thử 49**](#_Toc125389209)

[**5.1. Công nghệ, thư viện sử dụng 49**](#_Toc125389210)

[**5.1.1. Ngôn ngữ lập trình C# 49**](#_Toc125389211)

[**5.1.2. .NET framework 49**](#_Toc125389212)

[**5.1.3. Thư viện Siticone UI 49**](#_Toc125389213)

[**5.1.4. Thư viện TabLib Sharp 49**](#_Toc125389214)

[**5.1.5. Thư viện WMPLib 49**](#_Toc125389215)

[**5.1.6. Thư viện Playlist.NET 50**](#_Toc125389216)

[**5.1.7. Thư viện CSAudioVisualization 50**](#_Toc125389217)

[**5.2 Hướng dẫn cài đặt 50**](#_Toc125389218)

[**5.3. Kiểm thử 53**](#_Toc125389219)

[**5.3.1. Unit Test 53**](#_Toc125389220)

[**5.3.2. Intergration Test 53**](#_Toc125389221)

[**5.3.3. System Test 53**](#_Toc125389222)

[**Chương 6. Tổng kết 54**](#_Toc125389223)

[**6.1 Kết quả đạt được 54**](#_Toc125389224)

[**6.2 Những hạn chế 54**](#_Toc125389225)

[**6.3 Hướng phát triển 54**](#_Toc125389226)

[**6.4 Bảng phân công công việc 54**](#_Toc125389227)

# **Chương 1. Tổng quan**

## 1.1. Giới thiệu đề tài đồ án

Trong thời đại 4.0, công nghệ ngày càng phát triển và được tiếp cận bởi mọi lứa tuổi trong xã hội. Cũng từ đó nhu cầu giải trí của con người thông qua các thiết bị điện tử ngày càng tăng cao, đặc biệt là nhu cầu nghe nhạc và xem video. Mặc dù trên Internet có rất nhiều ứng dụng trình phát nhạc tuy nhiên có không ít ứng dụng có giao diện phức tạp không thân thiện với người dùng. Do đó việc lập trình và phát triển một ứng dụng để chạy và phát các đoạn nhạc video với một giao diện đơn giản, cách sử dụng dễ hiểu là một việc cần thiết và cũng rất thử thách với nhóm chúng em.

**My Media Player** là một ứng dụng được nhóm phát triển với tiêu chí đơn giản, dễ sử dụng nhưng không làm người dùng có trải nghiệm nhàm chán. Tuy đơn giản nhưng **My Media Player** có đầy đủ các tính năng cần có và có thêm một vài tính năng khác như tìm kiếm nhạc trên các nền tảng nhạc online như Zing MP3, SoundCloud, Spotify.

## 1.2. Giới thiệu bài toán cần giải quyết

Yêu cầu cần phải xây dựng một hệ thống có giao diện trực quan dễ sử dụng cho người dùng, kèm theo việc đầy đủ chức năng, nhưng có khả năng mở rộng, nâng cấp trong tương lai nhờ vào cấu trúc triển khai mã nguồn vững chắc.

## 1.3. Quy trình thực hiện

- Thiết kế giao diện từng thành phần của ứng dụng.

- Thêm logic cho các thành phần đã được thiết kế.

- Thêm các tính năng phụ vào ứng dụng.

- Kiểm thử.

- Bảo trì, nâng cấp, sửa chữa.

## 1.4. Một số chức năng nổi bật

- Tự động thêm các bài nhạc và Videos từ thư mục mặc định trên máy tính.

- Hỗ trợ phát nhạc kèm lời(lyrics) nếu có.

- Hỗ trợ tạo danh sách riêng các bài nhạc ưa thích cho người dùng.

- Hỗ trợ tìm kiếm nhạc trên các nền tảng online.

# **Chương 2. Phân Tích Yêu Cầu Phần Mềm Và Mô Hình Hóa**

## 2.1. Yêu cầu phần mềm

### **2.1.1. Yêu cầu chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Định danh** | **Ưu tiên** | **Mô tả** |
| 1 | Play | 1 | Phát file media |
| 2 | Properties | 1 | Hiển thị thông tin file media |
| 3 | Skip | 1 | Tua, skip file media |
| 4 | Shuffle and play | 2 | Phát file audio bất kỳ |
| 5 | Volumebar | 2 | Chỉnh âm lượng |
| 6 | Trackbar | 2 | Hiển thị thời điểm đang phát |
| 7 | Sort by | 3 | Sắp xếp thứ tự file |
| 8 | Play playlist | 3 | Phát playlist |
| 9 | Add to playlist | 3 | Thêm file media vào playlist |
| 10 | Rename | 4 | Chỉnh sửa tên playlist |
| 11 | Remove | 4 | Loại bỏ playlist |
| 12 | Add folder | 5 | Chọn folder để quét tìm file media |
| 13 | Online store | 5 | Tìm kiếm file audio trên ZingMP3 |
| 14 | Streaming | 5 | Phát file audio trên ZingMP3 |

### **2.1.2 Yêu cầu phi chức năng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Định danh** | **Ưu tiên** | **Mô tả** |
| 1 | Stability | 1 | Phát file media liền mạch, ổn định |
| 2 | Accuracy | 1 | Chức năng tìm kiếm file audio trên ZingMP3 chính xác |
| 3 | Effectiveness | 2 | Load file media nhanh |

### **2.1.3 Bảng FURPS**

|  |  |
| --- | --- |
| **Tiêu chí chất lượng** | **Mô tả** |
| Functionality | Hướng tới phục vụ người dùng với mục đích phát file media |
| Usability | Giao diện thiết kế đơn giản, dễ hiểu, dễ sử dụng |
| Reliablity | Hệ thống được thiết kế với khả năng giảm thiểu bug trong quá trình sử dụng |
| Performance | Tối ưu hóa, giảm thiểu thời gian load  Ổn định không giật lag khi sử dụng |
| Supportability | Kịp thời sửa chữa khi có crash |

## 2.2. Mô hình hóa

### **flow2.2.1 Sơ đồ luồng dữ liệu**

### **2.2.2. Sơ đồ các Class**

- Các Interfaces:

Text

Description automatically generated with medium confidenceA picture containing text

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

- Abstract class:

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedA picture containing text

Description automatically generated- Form:

- UserControl:

Graphical user interface

Description automatically generated with low confidenceGraphical user interface, text, application

Description automatically generated+ IMediaItem’s implementation.

Graphical user interface

Description automatically generated with low confidence

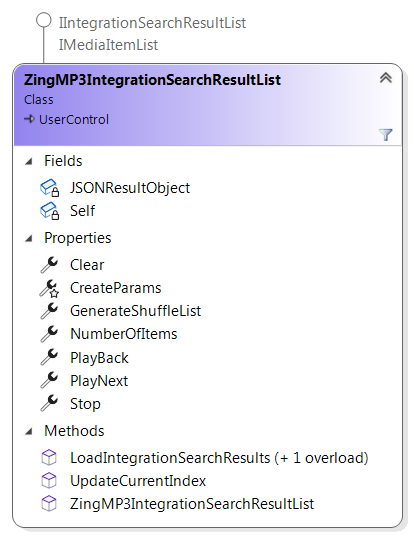
Text

Description automatically generated+ IMediaItemList’s implementation:

Graphical user interface, text, application

Description automatically generated Graphical user interface

Description automatically generated with medium confidence



Graphical user interface, application

Description automatically generated

Text

Description automatically generated with medium confidence Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, application

Description automatically generated

Graphical user interface, text

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

# **Chương 3. Kỹ thuật xử lí**

## 3.1 Phân tích

### **3.1.1 Phân tích hiệu ứng trực quan**

**Music Library**

* Khi click vào một TrackItem:
  + Màu chữ item đó đổi màu sang màu cam để người dùng có thế xác định bài nhạc đang được phát.
  + Panel mới hiện lên với bên trái là Visualizer và thông tin của bài nhạc, bên phải là lyric của bài nhạc.
  + Trackbar bắt đầu chạy tương ứng với thời gian phát hiện tại.
* Khi chọn Sort by:
  + Các bài nhạc sẽ được sắp xếp theo thứ tự alphabet dựa trên thuộc tính đã chọn

**Video Library**

* Khi click vào một video item:
  + Một form sẽ hiện lên phát video được chọn.
  + Label trên media controller hiện tên video.
  + Trackbar bắt đầu chạy tương ứng với thời gian phát hiện tại.

### **3.1.2. Phân tích xử lí chức năng**

**- Track Item**

Diagram

Description automatically generated

**Diagram

Description automatically generated- MediaController**

**- PlaylistItem**

Diagram

Description automatically generated

## 3.2. Kỹ thuật sử dụng

### **3.2.1. Xây dụng class MediaController như bộ điều khiển trung tâm của ứng dụng**

Graphical user interface, text, application, email

Description automatically generated

Class MediaController kế thừa từ class UserControl là bộ điều khiển chính của ứng dụng, được dùng để xử lí, thực thi những file phương tiện (media file) được hỗ trợ như file video và file audio.

Graphical user interface, application

Description automatically generated

***Giao diện của MediaController.***

Class MediaController bao gồm các property chính như sau:

Text

Description automatically generated

Trong đó:

+ CurrentMode (kiểu dữ liệu Mode) được dùng để xác định MediaController đang thực thi, xử lí loại file media nào, nhằm đưa ra quyết định hợp lí cho từng trường hợp.

+ CurrentPlayingMode (kiểu dữ liệu PlayingMode) được dùng để xác định trạng thái đang chơi ở dạng nào của MediaController.

Graphical user interface, text

Description automatically generated

***Định nghĩa của hai kiểu dữ liệu Mode và PlayingMode***

+ VideoForm (Form được dùng để trình chiếu file video) đang liên kết (binding) với MediaController nếu có. (Hai control liên kết với nhau nhằm đảm bảo tính nhất quán khi xử lí, thực thi, kiểm soát các file media)

+ CurrentMediaItemList (kiểu dữ liệu do interface ImediaItemList đại diện) là danh sách các file media đã được nạp vào trong MediaController nếu có. (Ngoài việc kiểm soát riêng lẻ từng file media, MediaController còn có khả năng kiểm soát một danh sách các file media)

+ Watcher (class Watcher do nhóm tự cài đặt có chức năng gần giống như class Timer nhưng mọi hoạt động diễn ra ở một thread khác – mục đích để thực hiện những tác vụ nặng có tính tuần hoàn rồi sau đó cập nhật lại phần UI ở UI thread) được dùng để cập nhật lại UI của MediaController theo thời gian thực.

+ Player (kiểu dữ liệu của class WindowsMediaPlayer – được trích ra từ thư viện WMPLib) là đối tượng (object) chính có khả năng chơi, quản lí các trạng thái hoạt động cũng như cung cấp các thông tin của file media khi file này được nạp vào đối tượng.

Ngoài ra, MediaController còn cung cấp thông tin cho class TrackLyrics thông qua TrackCurrentTimeInSeconds

Graphical user interface, text, application

Description automatically generated

Khi thực hiện streaming file audio, sẽ thường xuyên có trường hợp kết quả trả về khi thực hiện truy vấn bao gồm cả thông tin về Lyrics Synchronization (đồng bộ lời bài hát với bài hát theo thời gian thực) ở dạng JSON hoặc file .lru. Để tận dụng loại tài nguyên này, class TrackLyrics sau khi parse xong thông tin của tài nguyên sẽ liên tục yêu cầu phía MediaController gửi thông tin về thời điểm hiện tại của file audio đang được streaming nhằm hiển thị đúng câu từ của file audio ngay tại thời điểm đó.

Để thực hiện việc nạp file media (audio hoặc video), MediaController có các method sau:

Graphical user interface, text, application

Description automatically generated

Trong đó:

+ LoadLocalTrack được dùng để nạp những file audio có sẵn trong máy tính với tham số đầu vào là đường dẫn tuyệt đối (absolute path) đến file audio được yêu cầu nạp.

+ LoadStreamingTrack được dùng để nạp những thông tin của những file audio truy vấn được từ phía remote server, sau đó MediaController sẽ dùng những thông tin này để thực hiện streaming.

+ LoadLocalVideo được dùng để nạp những file video có sẵn trong máy tính với tham số đầu vào là đường dẫn tuyệt đối (absolute path) đến file video được yêu cầu nạp.

Để thực hiện việc nạp danh sách các file media (danh sách audio hoặc danh sách video), MediaController có các method sau:

Graphical user interface, text, application, email

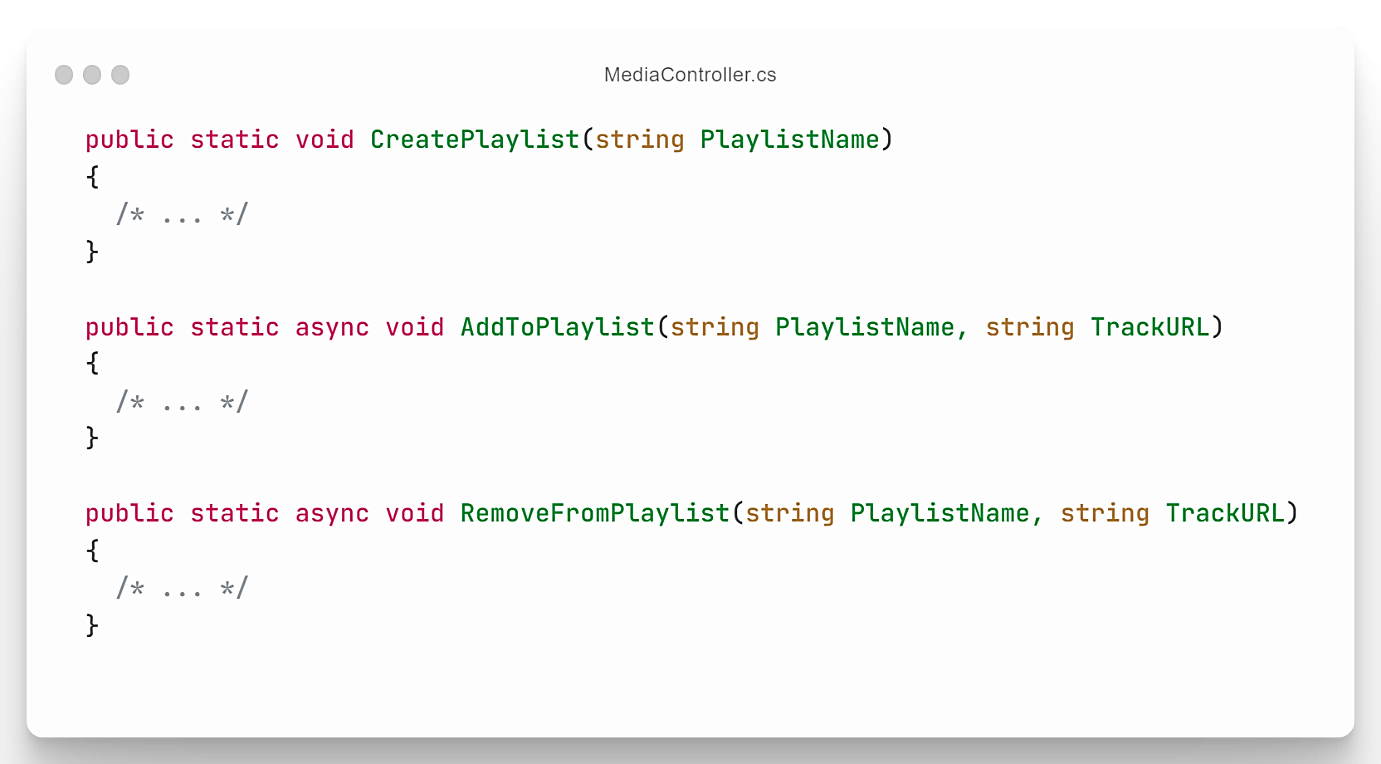
Description automatically generated

Trong đó:

+ LoadMediaItemList với tham số đầu vào là MediaItemList mới với mong muốn thay thế CurrentMediaItemList. Method sẽ kiểm tra sự tồn tại của CurrentMediaItemList, nếu có thì sẽ đặt lại trạng thái ban đầu cho MediaItemList mà CurrentMediaItemList đang tham chiếu tới. Cuối cùng method sẽ gán tham số MediaItemList cho CurrentMediaItemList.

+ LoadMediaItemListWithShuffle thực hiện tương tự như method LoadMediaItemList. Tuy nhiên ở bước cuối cùng, method sẽ gọi method GenerateShuffleList từ IMediaItemList giúp tạo sẵn một danh sách chơi ngẫu nhiên, lúc này khi thực hiện thao tác lên CurrentMediaItemList sẽ sử dụng danh sách được tạo ngẫu nhiên mà không sử dụng danh sách gốc.

Để thực hiện việc kiểm soát các playlist, MediaController có các method sau:



Các playlist hiện tại chỉ được dùng để chứa các file audio, trong đó:

+ CreatePlaylist tạo ra một playlist mới được lưu trong máy tính có tên trùng với giá trị của tham số đầu vào PlaylistName.

+ AddToPlaylist thêm một đường dẫn đến một file audio vào trong playlist đã có ở trong máy tính có tên trùng với giá trị của tham số PlaylistName.

+ RemoveFromPlaylist xóa đi một đường dẫn đến một file video ở trong playlist đã có sẵn trong máy tính có tên trùng với giá trị của tham số PlaylistName.

Text

Description automatically generated

***Tổng quan method UpdateMediaController của class MediaController***

Method UpdateMediaController được gán vào Watcher với mục đích thực thi việc cập nhật lại UI của MediaController một cách tuần hoàn theo thời gian phù hợp với trạng thái hoạt động của ứng dụng.



Sau cùng là MediaControllerArgs cùng với event OnLoadMedia cũng như delegate OnLoadMediaHandler giúp những class khác có khả năng thực thi những tác vụ đi kèm khi event OnLoadMedia được gọi (thông thường được gọi khi thực thi hai method LoadLocalTrack và LoadStreamingTrack).

Ngoài ra còn có những method nhỏ khác giúp điều khiển file media được cài đặt trong MediaController.

### **3.2.2. Xây dựng các interface IMediaItem, IMediaItemList, abstract class AMediaItemList giúp trừu tượng hóa các class TrackItem, TrackItemList, VideoItem, VideoItemList**

Text

Description automatically generated

***Nội dung mã nguồn của interface IMediaItem***

Interface IMediaItem hỗ trợ những control, class có nhu cầu trở thành một “Media Item”. Khi áp dụng interface vào những control, class có nhu cầu sẽ làm cho việc tích hợp (sử dụng pattern Dependency Injection trong OOP) vào các control, class khác muốn sử dụng những control, class này trở nên một cách dễ dàng, thuận tiện hơn, giúp tăng khả năng tái sử dụng mã nguồn. Interface IMediaItem mang đến cho class sở hữu property thuộc kiểu dữ liệu của interface này khả năng truy xuất thông tin của property cũng như thực thi được hành động của property.

Text

Description automatically generated

***Nội dung mã nguồn của interface IMediaItemList***

Tương tự công dụng của interface IMediaItem, IMediaItemList giúp tái sử dụng mã nguồn khi tích hợp vào trong MediaController. Khi MediaController có được một property (CurrentMediaItemList) thuộc kiểu dữ liệu của interface này sẽ dễ dàng điều khiển được property thông qua giao diện các method đã cung cấp.

Để làm tăng thêm khả năng tái sử dụng mã nguồn, base/abstract class AMediaItemList được ra đời:

Text

Description automatically generated

Text

Description automatically generated

- Sau đây là phần triển khai các abstract/base class, interface vào trong các control:

Text

Description automatically generated

***Cài đặt interface IMediaItem cho control VideoItem***

Text

Description automatically generated

***Cài đặt abstract/base class AMediaItemList và interface IMediaItemList cho control VideoItemList.***

### **3.2.3. Xây dựng các interface IIntegration, IIntegrationSearchResultList giúp trừu tượng hóa các class ZingMP3Integration, ZingMP3IntegrationSearchResultList**

A picture containing timeline

Description automatically generated

***Nội dung của interface IIntegration.***

Text

Description automatically generated

***Nội dung của interface IIntegrationSearchResultList.***

Text

Description automatically generated with low confidence

***Cài đặt interface IIntegration cho class ZingMP3Integration.***

Graphical user interface, text, application

Description automatically generated

***Cài đặt interface IIntegrationSearchResultList cho control ZingMP3IntegrationSearchResultList.***

### **3.2.4. Sử dụng class FileSystemWatcher để thực hiện động bộ hóa dữ liệu**

Để thực hiện việc đồng bộ hóa dữ liệu các file media giữa ứng dụng và hệ thống tập tin trong máy tính (người dùng có thể quan sát được sử dụng file explorer), ứng dụng đã dùng class FileSystemWatcher được cung cấp từ thư viện System.IO.

Theo định nghĩa từ .NET document của Microsoft:

“FileSystemWatcher Class listens to the file system change notifications and raises events when a directory, or file in a directory, changes.”

Graphical user interface, text, application

Description automatically generated

***Minh họa cách cài đặt FileSystemWatcher cho tính năng thêm, xóa,***

***chỉnh sửa, cập nhật playlist.***

Khi thư mục được chỉ định (ở đây mặc định là thư mục Playlists) có những thay đổi về tập tin sẽ khiến các event như Changed, Created, Deleted được kích hoạt. Mỗi loại event khác nhau sẽ cập nhật lại control ContextMenuStrip cũng như danh sách các control playlist theo cách khác nhau (Ví dụ: Khi tạo ra một playlist mới thì event Created sẽ thêm mới một control ToolStripMenuItem vào trong control ContextMenuStrip, đồng thời danh sách các control playlist cũng sẽ thêm mới một playlist item). Sử dụng chung một ContextMenuStrip cho các control file media sẽ làm giảm bớt số lượng các control phải nạp vào trong mỗi control (giúp giảm bộ nhớ), ngoài ra việc đồng bộ cũng trở nên dễ dàng cài đặt, sửa chữa hơn.

Text

Description automatically generated

***Class FileSystemWatcherService dùng để quản lí các FileSystemWatcher***

Nhược điểm của FileSystemWatcher theo khuyến nghị của Microsoft là với mỗi loại file có đuôi mở rộng khác nhau nên dùng riêng một FileSystemWatcher để theo dõi. Do file audio và file video có quá nhiều định dạng, đồng nghĩa với quá nhiều đuôi mở rộng khác nhau, việc cài đặt FileSystemWatcher cho mỗi loại file sẽ trở nên nhàm chán, không tái sử dụng được mã nguồn. Đây là lí do class FileSystemWatcherService được ra đời, với việc sử dụng một mảng băm hai chiều (với các khóa lần lượt là đường dẫn và đuôi mở rộng) sẽ giúp cho việc quản lí, cấp phát, thu hồi bộ nhớ cho các FileSystemWatcher trở nên thuận tiện hơn, trong đó:

+ Method Subscribe giúp thêm một bộ dữ liệu vào trong mảng băm hai chiều.

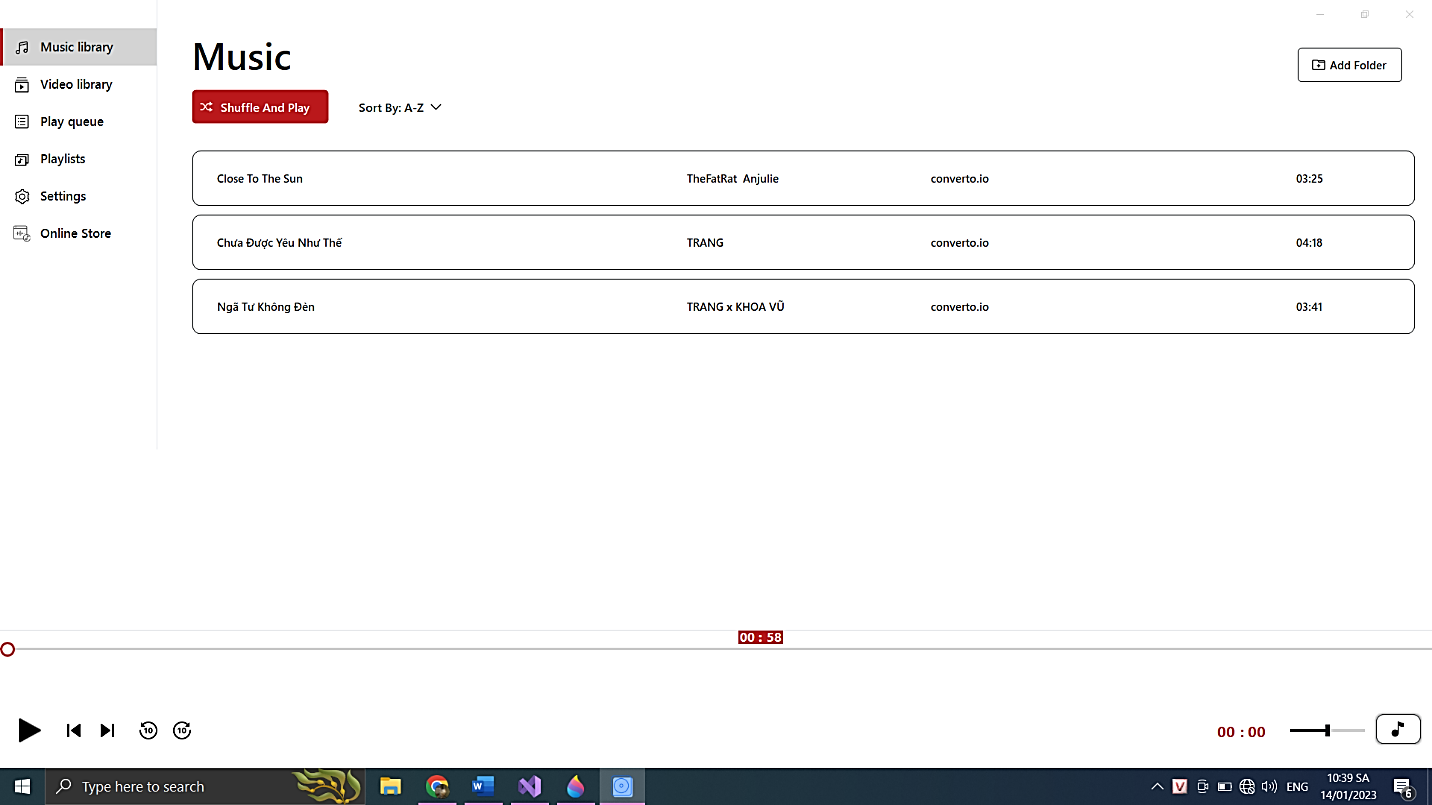
+ Method UnSubscribe giúp xóa một bộ dữ liệu khỏi mảng băm hai chiều.

+ Method Register giúp cài đặt handler cho mỗi loại event của FileSystemWatcher có trong mảng băm hai chiều.

# **Chương 4. Thiết kế giao diện**

1. Ảnh có chứa văn bản

   Mô tả được tạo tự động***Giao diện khi mở ứng dụng lần đầu***

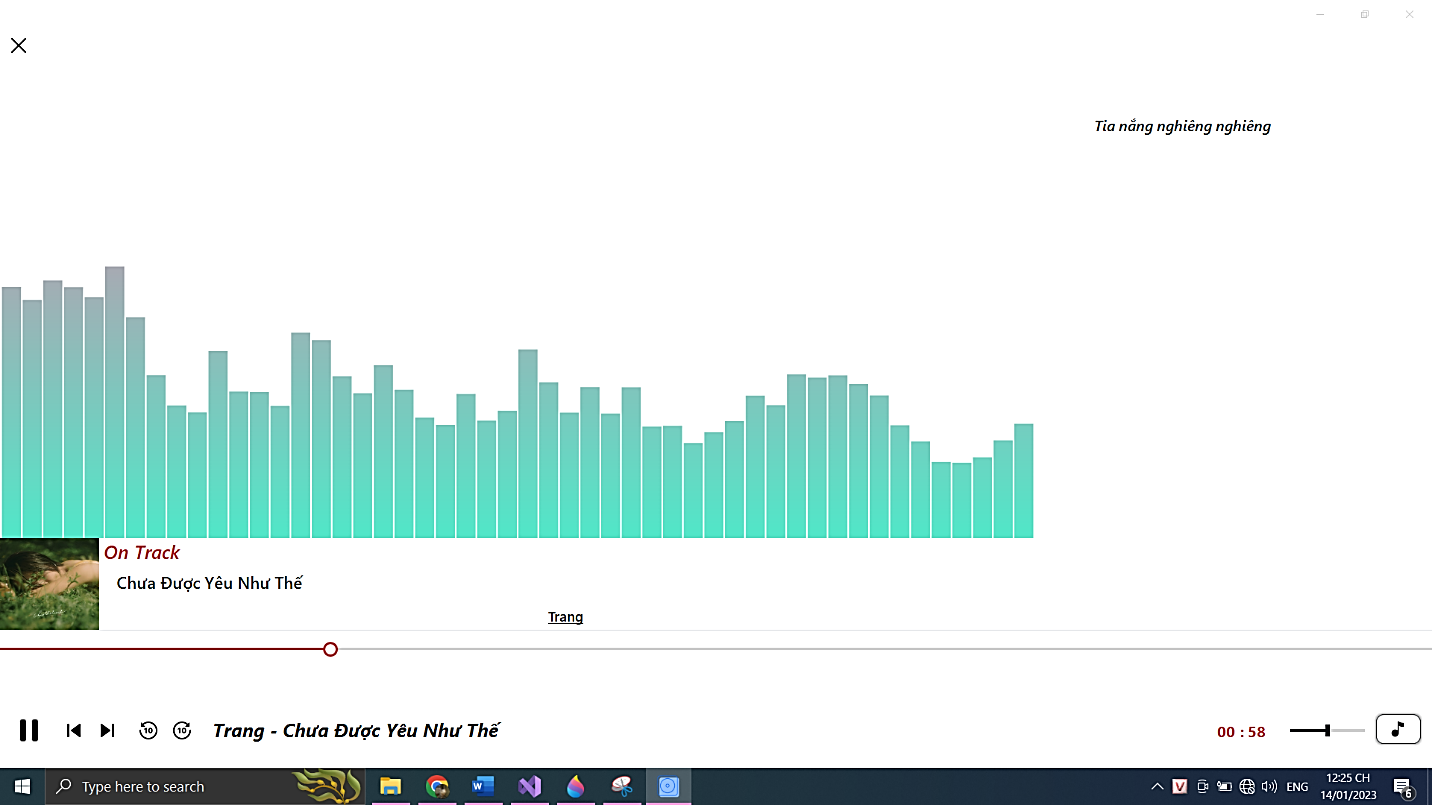


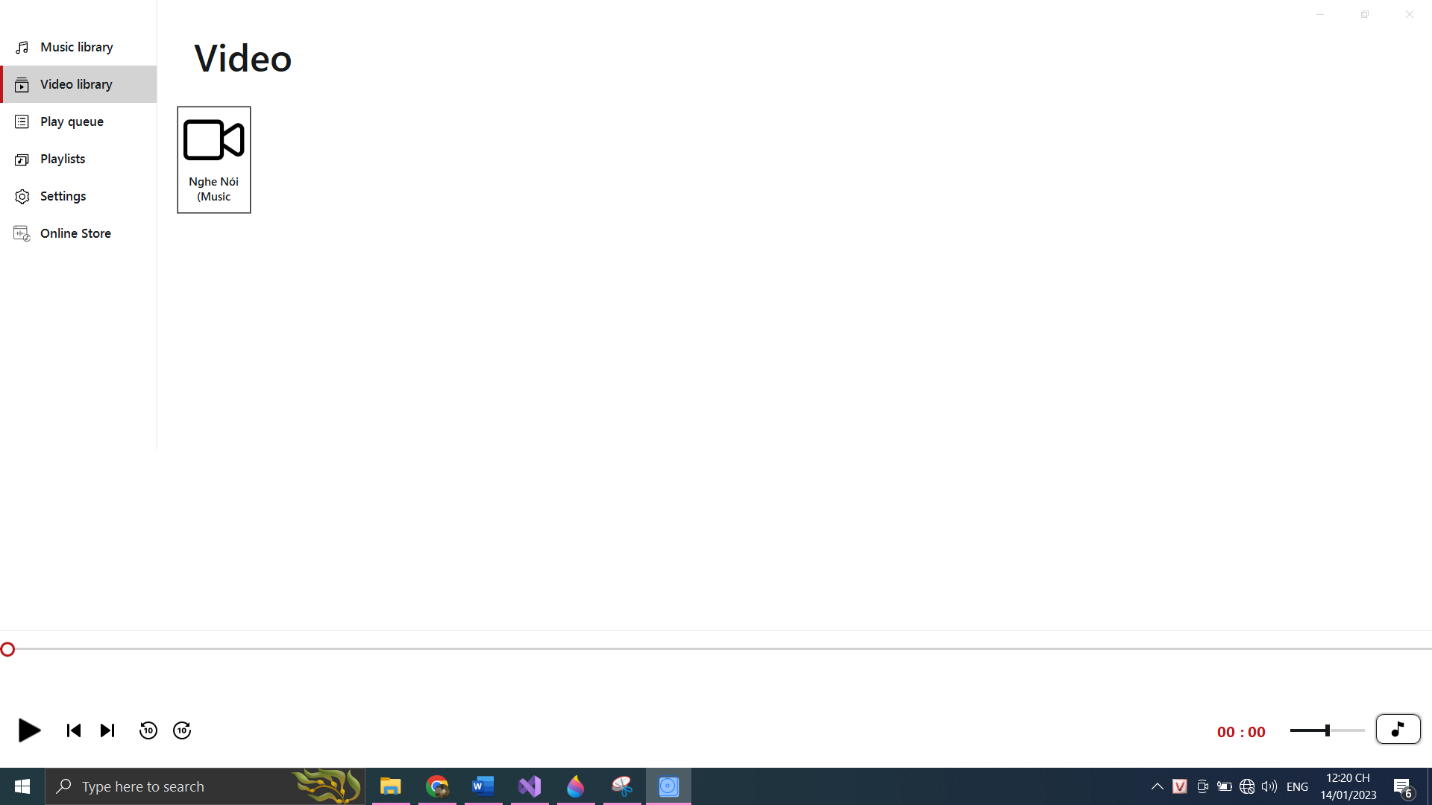
***2. Giao diện tab Music Library sau khi thêm***

Ảnh có chứa văn bản

Mô tả được tạo tự động***nhạc vào ứng dụng.***

***3. Các chức năng trong tab Music Library.***

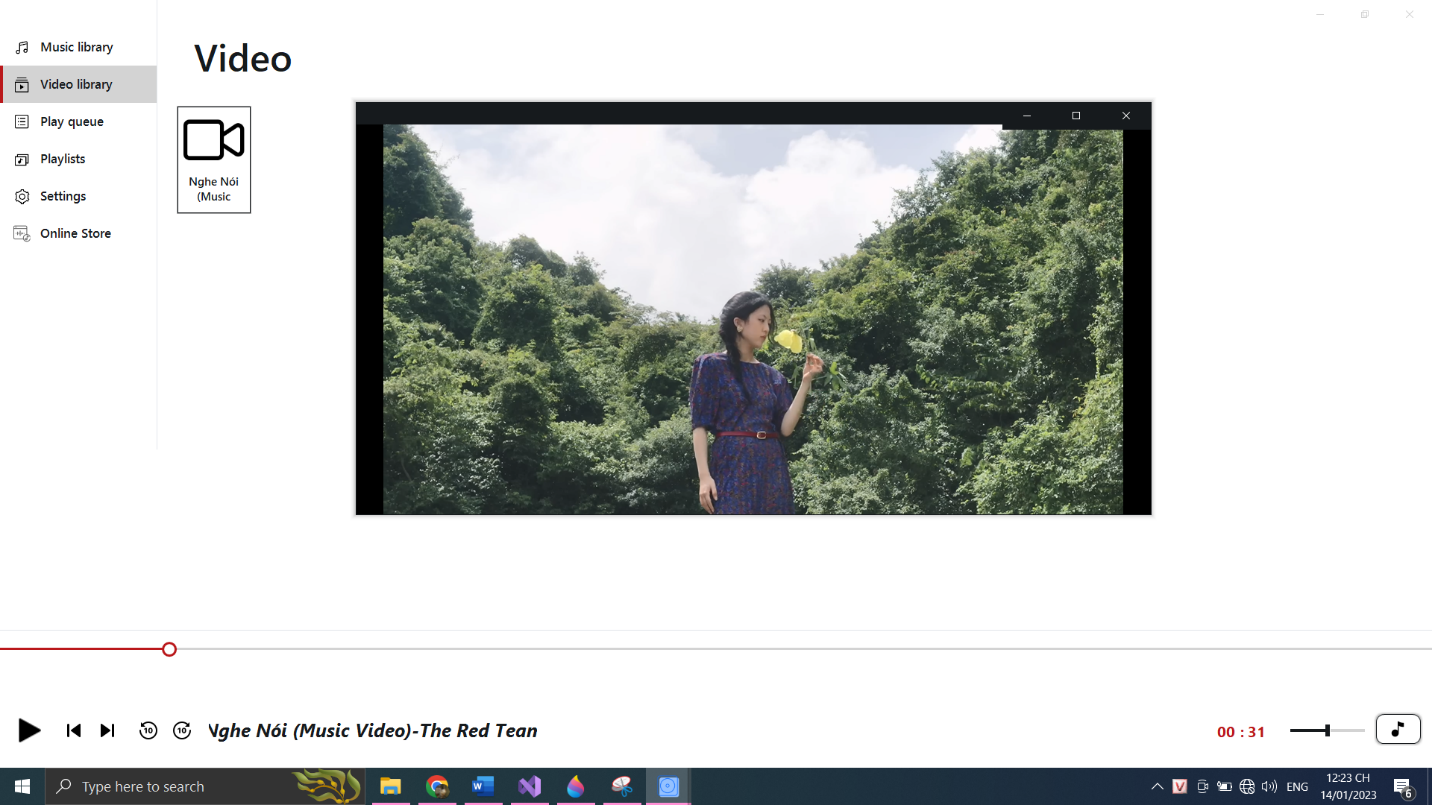
***4. Giao diện ứng dụng khi phát nhạc***

******

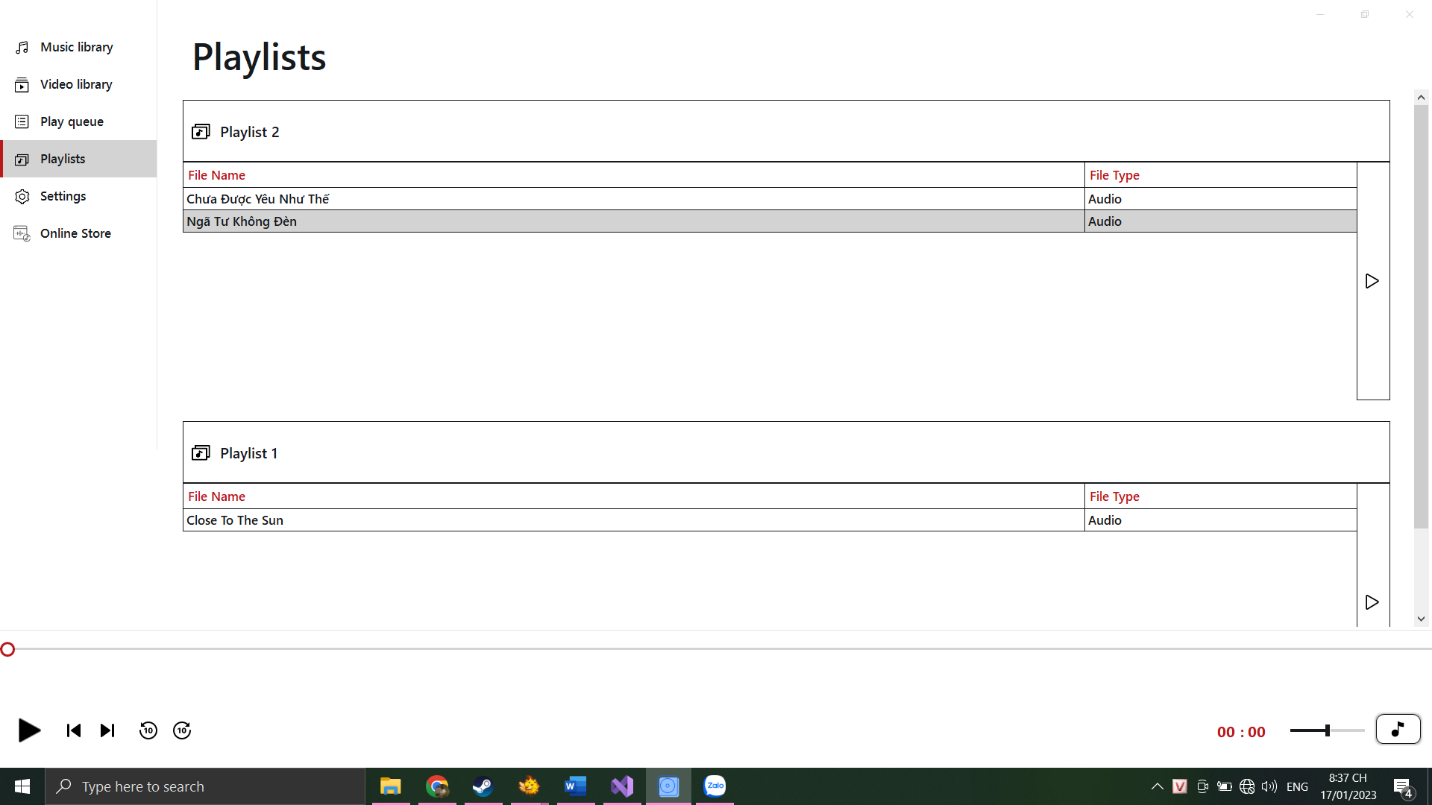
***5. Giao diện tab Video Library.***

***6. Giao diện ứng dụng khi phát Video.***

Ảnh có chứa văn bản

Mô tả được tạo tự động- Thêm những bài nhạc người dùng muốn ưu tiên nghe vào hàng đợi trong tab Play Queue.

***7. Giao diện tab Play Queue.***

- Tạo các danh sách riêng để lưu những bài nhạc yêu thích trong tab Playlist.

***8. Giao diện tab Playlists.***

- Chỉnh sửa các thư mục nhạc/video đã thêm vào ứng dụng trong tab Settings.

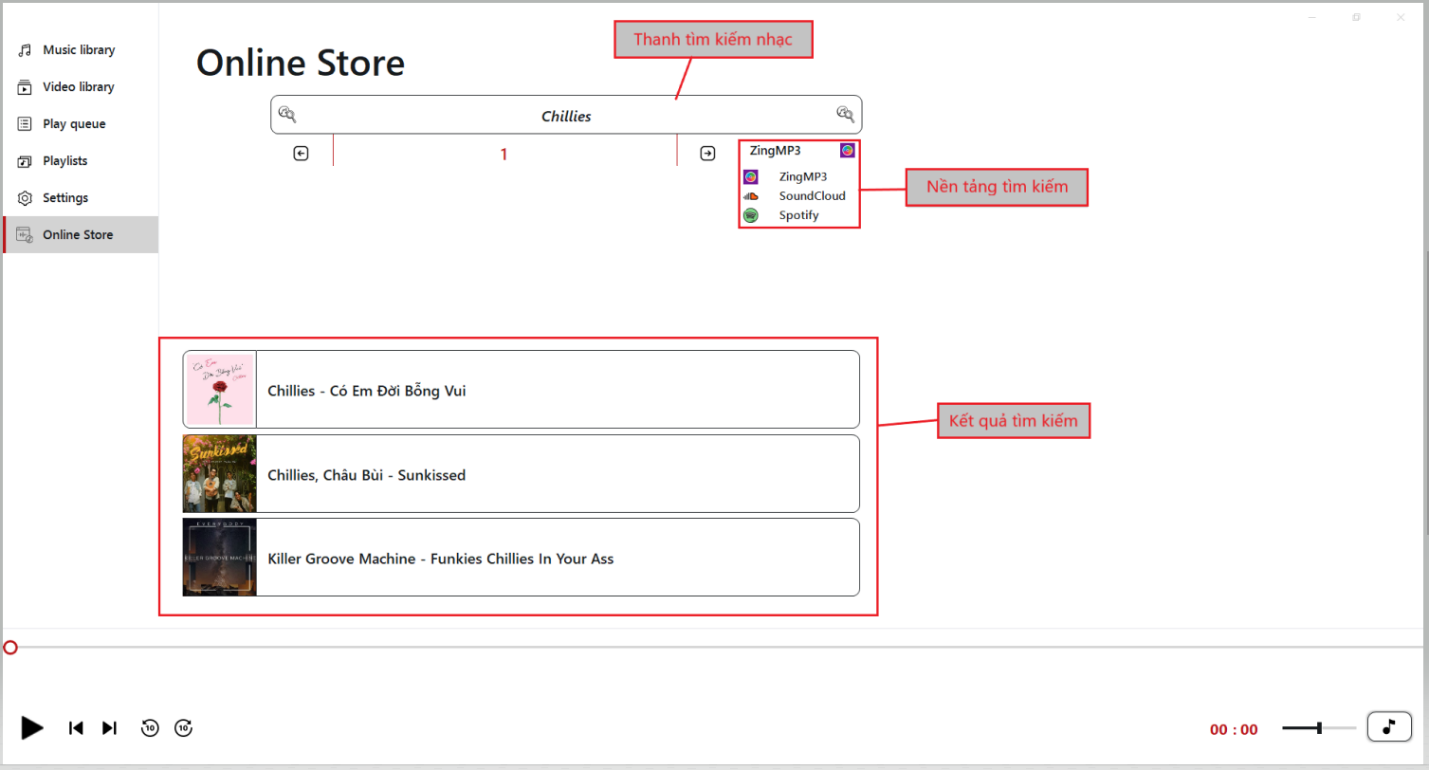
***9. Giao diện và các tính năng của tab***

***Settings.***

Ảnh có chứa văn bản

Mô tả được tạo tự động- Tìm kiếm nhạc trên các nên tảng Online

***10. Giao diện tab Online Store***

- Giao diện và chức năng của bộ điều khiển

***11. Các chức năng của bộ điều khiển.***

# **Chương 5. Cài đặt và kiểm thử**

## 5.1. Công nghệ, thư viện sử dụng

### **5.1.1. Ngôn ngữ lập trình C#**

C# là một ngôn ngữ cấp cao và thuần hướng đối tượng, hỗ trợ rất mạnh trong việc xây dựng các ứng dụng trên môi trường Windows. Đối với những lập trình viên đã từng sử dụng qua C, C++, hoặc Java, việc nắm bắt và sử dụng C# một cách hiệu quả là khá dễ dàng và ít tốn thời gian.

### **5.1.2. .NET framework**

Net Framework là một công cụ hay một nền tảng lập trình hoạt động trên hệ điều hành Windows và được sáng tạo bởi nhà Microsoft. Net Framework chịu trách nhiệm điều khiển cho hệ thống phần mềm với chức năng chính là đảm bảo an toàn, quản lý bộ nhớ và xử lý một số lỗi.  Net Framework đó là tạo ra một môi trường để các nhà phát triển thoải mái sáng tạo code của riêng họ. Code này sẽ có thể dùng cho tương thích với rất nhiều phần cứng mà Net Framework hỗ trợ.

### **5.1.3. Thư viện Siticone UI**

Siticone UI (là một phần của Siticone Framework Suite) là framework hàng đầu dành cho các developer trong mảng thiết kế UI/UX của .NET và Windows Forms. Nó là tập hợp của tất cả các thư viện và framework dùng để phát triển UI của .NET. Nó giúp cho các developer để thiết kể ra các UI sinh động, hiệu quả, nhanh chóng nhằm mang lại trải nghiệm người dùng cực kỳ ấn tượng mà không cần quá nhiều kinh nghiệm lập trình.

### **5.1.4. Thư viện TabLib Sharp**

Là một thư viện nền tảng độc lập của .NET để đọc và ghi dữ liệu trong các file media, bao gồm các file video, âm thanh và ảnh. Nó giúp đơn giản hóa việc trình bày hoặc gắn thẻ tất cả bộ sưu tập media của người dùng, bất kể định dạng/vùng chứa nào mà chúng có thể sử dụng. Người dùng có thể đọc/ghi các thẻ/thuộc tính phổ biến của media hoặc cũng có thể tạo và truy xuất các thẻ tùy chỉnh của riêng mình.

### **5.1.5. Thư viện WMPLib**

WMPLib là thư viện âm thanh của Windows Media Player, thư viện này có sẵn trong hệ điều hành Windows, hỗ trợ xử lý nhiều loại file âm thanh như WAV, MP3… và có nhiều phương thức thao tác với âm thanh như Play, Pause, Stop, Volume, Playlist.

### **5.1.6. Thư viện Playlist.NET**

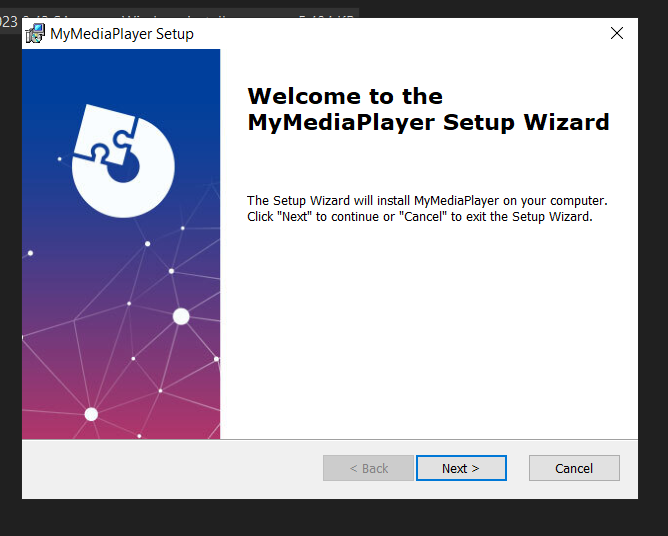
Là một thư viện để đọc và ghi các tệp danh sách phát (Playlist). Thư viện này hỗ trợ các định dạng: m3u, pls, wpl, zpl.

### **5.1.7. Thư viện CSAudioVisualization**

Là một thư viện hỗ trợ hiển thị ra các sóng âm thanh của các thiết bị đầu ra.

## 5.2 Hướng dẫn cài đặt

- Mở file cài đặt **MyMediaPlayerInstaller.msi** , một của sổ cài đặt hiện ra:



- Chọn **Next**:

Ảnh có chứa văn bản

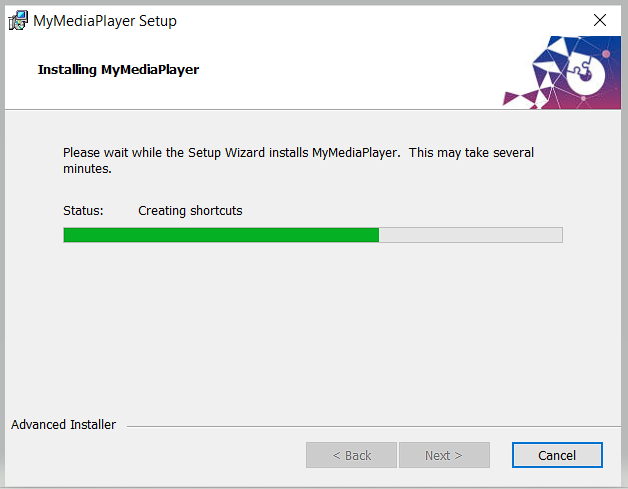
Mô tả được tạo tự động

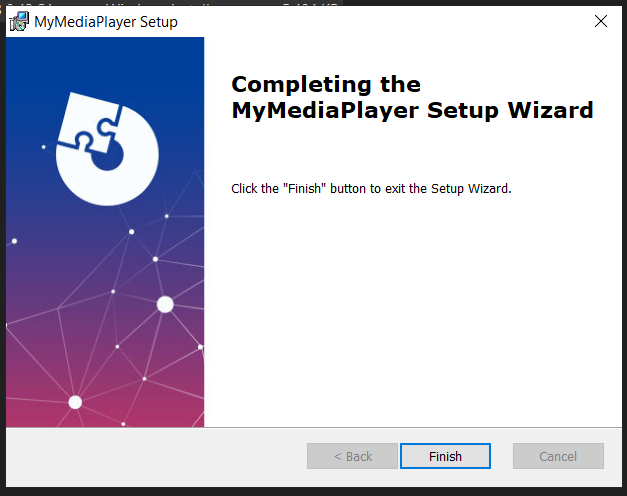
- Chọn vị trí để cài đặt ở nút **Browse…** sau đó chọn **Next:**

Ảnh có chứa văn bản

Mô tả được tạo tự động

- Chọn **Install:**





- Cài đặt thành công, nhấn **Finish** để hoàn tất cài đặt.

## 5.3. Kiểm thử

### **5.3.1. Unit Test**

Để đảm bảo chất lượng của chương trình, từng lớp sẽ được kiểm tra một cách riêng rẽ bằng phương pháp hộp trắng. Người tiến hành sẽ là những người trực tiếp cài đặt các lớp và phương thức ấy.

### **5.3.2. Intergration Test**

Kiểm tra sự liên kết và giao diện giữa các lớp trong phần mềm.

### **5.3.3. System Test**

Hệ thống sẽ được kiểm tra theo phương pháp hộp đen. Tức là dựa vào những đặc tả và yêu cầu của hệ thống đã thu thập được từ đầu để kiểm tra output chung của hệ thống mà không cần quan tâm đến việc thực hiện bên trong của chúng.

# **Chương 6. Tổng kết**

## 6.1 Kết quả đạt được

- Hoàn thành tất cả các mục tiêu nhóm đã đề ra:

+ Mở và phát được các file âm thanh, video.

+ Thiết kể và hoàn thiện giao diện trực quan, sinh động.

+ Hoàn thiện các tính năng phụ (Tạo hàng đợi phát nhạc, Tạo danh sách nhạc yêu thích, Stream nhạc từ các nền tảng online…).

- Học được cách thức, quá trình thực hiện một dự án phần mềm.

- Cải thiện kĩ năng làm việc nhóm của mỗi thành viên.

## 6.2 Những hạn chế

- Vẫn còn một số lỗi vặt chưa khắc phục:

+ Thanh thời gian có một số lỗi khi người dùng tương tác kéo thả.

+ Hiển thị sóng nhạc vẫn chưa tương thích với các thiết bị tai nghe.

- Tính năng stream nhạc Spotify với SoundCloud chưa hoàn thiện do kiến thức còn hạn hẹp nên chưa khai thác được API của những server này.

- Giao diện dễ nhìn nhưng chưa đạt được yêu cầu cuốn hút, hấp dẫn với người dùng.

## 6.3 Hướng phát triển

+ Phát triển thêm tính năng Streaming Video từ các nền tảng như Youtube, …

+ Phát triển tính năng cá nhân hóa giao diện, giúp người dùng tạo đặc trưng của cá nhân.

+ Thêm vào Settings các cài đặt thay đổi giao diện (hình thức, bố cục, màu sắc,…).

+ Tiếp tục khai khác sâu API của ZingMP3.

## 6.4 Bảng phân công công việc

**Tổng quan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Công việc** | **Người thực hiện** | **MSSV** | **Mức độ hoàn thành** |
| **1** | Tìm hiểu và đăng kí đề tài đồ án | Phạm Tuấn Anh | 21520147 | 100% |
| **Lập trình** | | | | |
| **2** | Thiết kế giao diện | Nguyễn Tiến Anh | 21520579 | 80% |
| **3** | Thiết kế hệ thống, chức năng | Phạm Tuấn Anh | 21520147 | 100% |
| **4** | Cài đặt class MediaController | Nguyễn Văn Vũ | 21521682 | 50% |
| **5** | Cài đặt class PlaylistItem | Nguyễn Chí Lâm | 21521055 | 50% |
| **6** | Cài đặt tổng quan, các class TrackItem, TrackItemList, FolderLocation | Nguyễn Tiến Anh | 21520579 | 80% |
| **7** | Cài đặt các class, interface còn lại. Cài đặt bổ sung, hỗ trợ cho các thành viên trong nhóm | Phạm Tuấn Anh | 21520147 | 100% |
| **Báo cáo** | | | | |
| **8** | Bố trí, trình bày bài báo cáo | Nguyễn Văn Vũ | 21521682 | 80% |
| **9** | Viết chương 1 + 4 + 5 + 6 | Nguyễn Văn Vũ | 21521682 | 100% |
| **10** | Viết chương 2 cho đến phần 2.2.1 | Nguyễn Chí Lâm | 21521055 | 100% |
| **11** | Viết phần 2.2.2 cho đến phần 3.1 | Nguyễn Tiến Anh | 21520579 | 100% |
| **12** | Viết phần 3.2 | Phạm Tuấn Anh | 21520147 | 100% |