

Peer-to-Peer Music Sharing System

CSCI3280 course project specification- Phase 1

Feb. 8 - Mar. 16

Project Introduction

In this project, you are required to achieve **two** main goals.

- Firstly, you are required to implement **a graphical music player** to play WAV audio files, control the playback, display music lyrics and manage a music library.
- Secondly, you are required to build a Peer-to-Peer (P2P) system for playing the music (in the form of streaming) from remote computers. Each computer works as both client and server, which means you may get the audio data from other computers and share audio data for others to be downloaded.

Note that you are welcome to use any programming languages, such as C/C++, C#, Java, python or use different language to handle different parts. You have to explicitly state what packages or libraries your programs are based on when presenting your work. We do not limit the operating system (OS) for the application to work on, but we suggest Microsoft Windows.

The project should be carried out by a group of **exact four** students. Your system should fulfill the basic requirements and have enough enhanced features. After implementing the system, you are required to demonstrate your system as well as to give a **12-minute** demonstration of your project.

▶ 0:00 / 0:26 ⏪ 🔍 🔍 ⏴

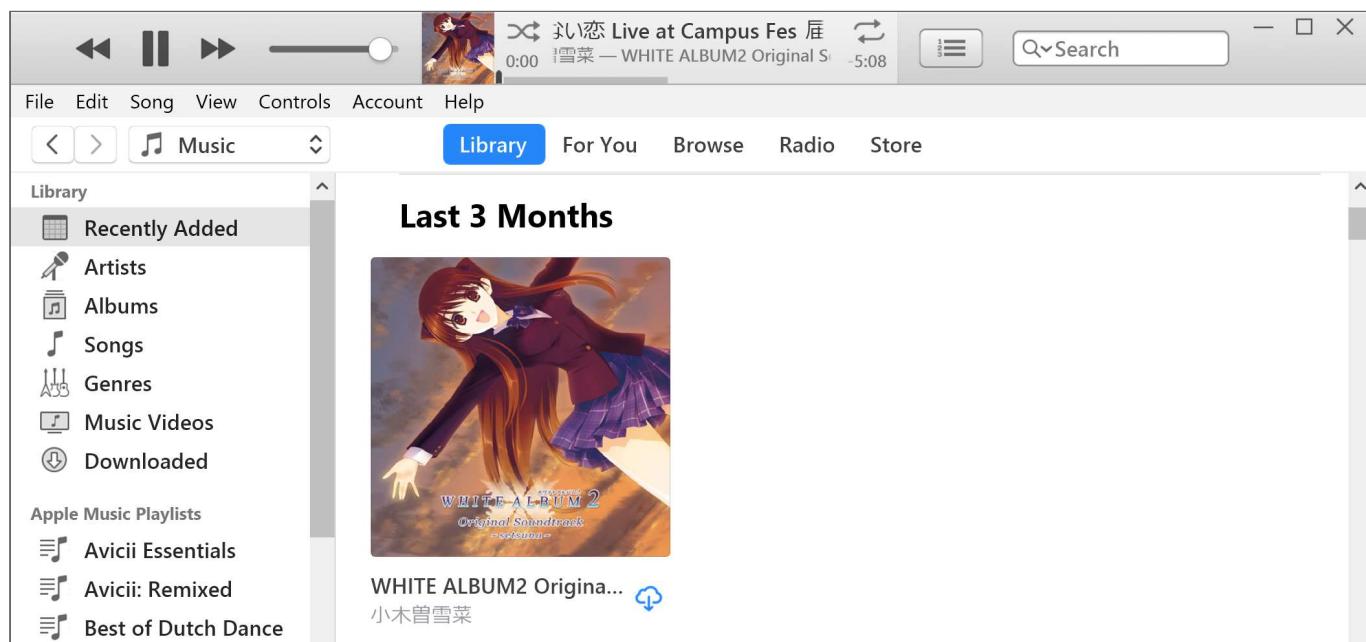
Search...

Title	Author	Album	Duration	Location
CONTROL	CAPSULE	CAPS LOCK	4:36.584	local
真夏のSounds good!	AKB48	真夏のSounds good!	4:35.33	network
君という光	GARNET CROW	Crystallize ~君という光~	5:10.493	network
Poacher	Oona Räisänen	None	00:26.123	local

Lyrics

-None-

A very simple web-based player.



iTunes, a well designed music player.

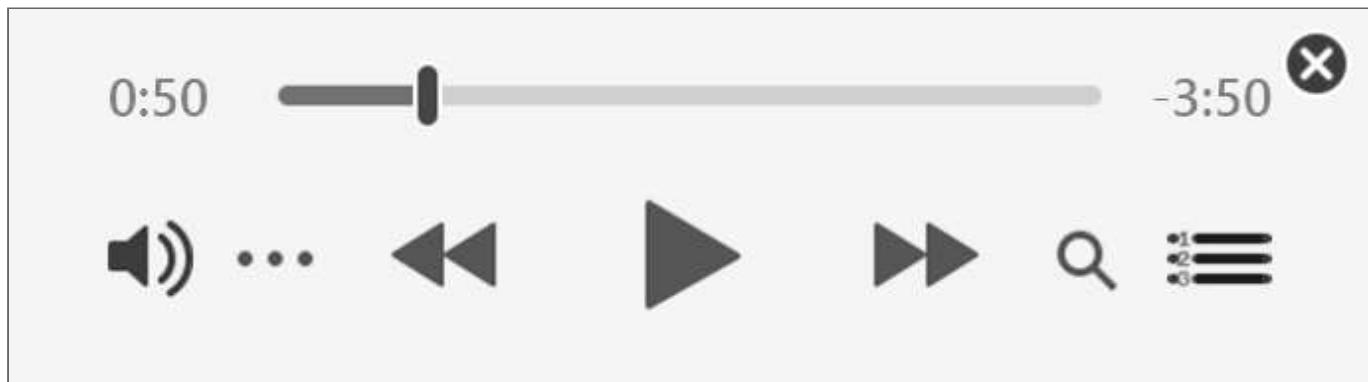
Introduction to Phase I

For the part one, the music player part, you are required to implement an graphical music player to play WAV format audio files, display music information, and search music.

Basic Requirements (Phase I)

Basic User Interface

Your program shall have a basic user interface. The interface should at least include a play/stop button, and a list control of audio files. The user can select the music in the list control to play the audio file. You also need to provide an interface for user to edit the information of audio files.



A demonstration of playback controls. You can add any function as you wish, but the controls should contain at least a play/stop (or a play/pause) button.

Music Decoding and Playback

You are required to understand the inside structure of wave format and write your own codes that can open, analyze, and playback a WAV file. The fmt sub-chunk and the data sub-chunk of the WAV file must be read and extracted manually, which means **you cannot use any third-party libraries/programs**. The sound data should be played fluently and bring the users beauty of music.

Music Management

Your program should have a database which stores music information (e.g., album, title, length) such that the program can detect the music files in the database then display them in the play list. Your program should also have a database (or a text file) to store the information of the audio

files. The information of the audio files should be manually input and removed by the user or automatically generated.

```
'1.wav' 'Years' 'Eason Chan' 'Black , White & Grey'  
'2.wav' 'Unfortunately, not you' 'Jasmine Liang' 'Silk Road'  
'3.wav' 'Our Song' 'Leehom Wang' 'Change their'  
'4.wav' 'Unfortunately, not you' 'HU XIA' 'NONE'
```

A minimal database based on text.

Music Information Display

You program shall be able to display the information of the music including the music title, singer, album name according to your database. When the user plays a song in the list control, the corresponding information should be displayed. If the certain information is not available, the program should display “None” on certain places.

Name	Time	Artist	Album
WHITE ALBUM Live at Campus Fes	4:40	小木曾雪菜	WHITE ALBUM2 Or...
White As Snow ...	4:12	CAPSULE	WAVE RUNNER
White As Snow	4:12	CAPSULE	Wave Runner (Delu...
White As Snow (extended mix)	5:11	CAPSULE	WAVE RUNNER
White Garden	3:08	Another Infinity fea...	Sakura Luminance
White Is Right	2:00	Pink Guy	Pink Season
WHITE LOVE	6:55	JUJU	Request
White Love / MOMENT	5:38	Speed	Xiami Compilations
White Peak	4:13	xi	RADIAL

A demonstration of music files list.

Music Searching

User can type in keywords to search music based on your data base. Your program can search the music from the database of music according to the keywords. The results should be displayed in the list control of your program. What's more, the user could search from any properties of the music, including music title, singer, and album as the keywords.

Lyrics Display

Your program should be able to play music and show lyrics. The lyrics file can be simple text files or in LRC format. The location of the lyric file can be maintained by the database, or simply place the file to the same folder of the music file with similar names.

Suggested Enhanced Features (Phase I)

Support for other music formats

You can use any 3rd-party libs/programs or implement by yourself to extend your music player to support other audio formats, such as mp3, aac, ogg and others.

Support for video playback

Progressbar

You may implement a progressbar on the GUI to support fast seeking for playback.

Synchronized lyric display

LRC format contains lyrics with the addition of timing information. You may play the music and show the lyrics synchronously. You should download LRC file of your favorite song from the internet or type the lyrics text without Time-info, and then edit it by yourself.

Visualization

You may create beautiful music visualization effects based on the music. For example, display music spectrum based on Fourier analysis. You may grab some inspirations from here:

willianjusten/awesome-audio-visualization



A bar visualization of music. (Based on WMPlayer)

And... any other creative ideas to enhance the system