iSongs

Use-Case Specification: Search songs by singing or humming , Manage playlists,and Recognize the songs

Version 1.0

Link Github: https://github.com/doannhatquang27/SE1406\_Group06/blob/master/submission/SE1406\_Group6\_Isongs\_Assignment2.docx

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17/07/2021 | 1.0 |  | Le Nhat Hanh Lan  Doan Nhat Quang  Nguyen Xuan Linh Tam  Nguyen Thanh Nhan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Use-case Model 4

2. Use-case Specifications 4

2.1 Use-case: Search songs by singing or humming 4

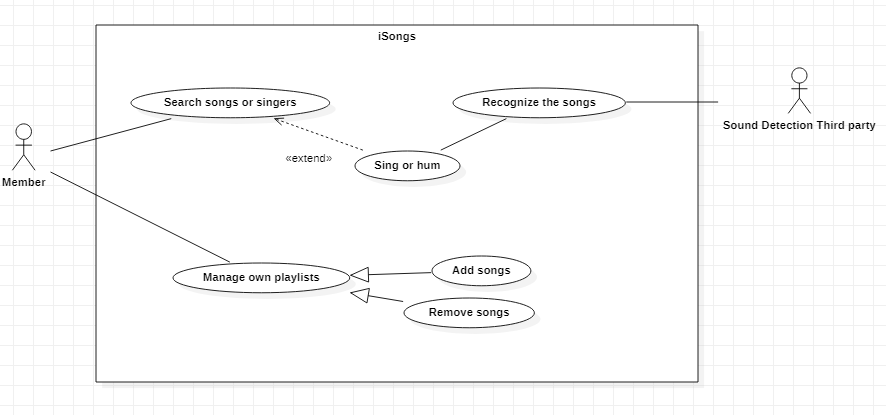
2.2 Use-case: Manage playlists 5

2.3 Use-case: Recognize the songs 6

3. Evaluation architecture style 6

4. UML Design 7

# Use-case Model (Mô hình use-case)



# Use-case Specifications (Đặc tả use-case)

## Use-case: Search song by singing or humming

**Reason to choose this UC:** This is the main highlight feature in this application

|  |  |
| --- | --- |
| Use case Name  (Tên use-case) | Search song by singing or humming |
| Brief description  (Mô tả) | Use your voice when singing or humming to find the songs and its information in case you forget the title of the song |
| Actors  (Actors) | Member |
| Basic Flow  (Luồng cơ bản) | 1. At the homepage, the user selects the icon button so that the application starts recording his voice 2. The user will start singing or humming the song they want to search for 3. The application will use that recording and through a third party to identify the song 4. If the song is found, it will return a response containing some information including the name, artist, album. Then display it to the user |
| Alternative Flows  (Các luồng thay thế) |  |
| Pre-conditions  (Điều kiện đầu vào) | User has an account and already signs in to the application |
| Post-conditions  (Điều kiện đầu ra) | The user successfully receive information of the song which wanted to be found |

## Use-case: Manage own playlists

**Reason to choose this UC:** User can perform CRUD functionality freely in his own playlist without administrator actions

|  |  |
| --- | --- |
| Use case Name  (Tên use-case) | Manage own playlists |
| Brief description  (Mô tả) | This use case describes how the member can create and share his own playlists with other users. |
| Actors  (Actors) | Member |
| Basic Flow  (Luồng cơ bản) | 1. At the homepage, the user clicks “Create Playlist”. 2. System displays new page with the “search” field. 3. User enters the song he wants to search. 4. System displays a list of songs with the same name as the search keyword. 5. User clicks “add” to add the song to his playlists. 6. User double clicks on the created playlists to name it. 7. System displays “Edit Playlist” for user to enter the name. 8. User clicks “Save”. 9. The playlist which the user has just added can be immediately found on the system and updated the playlists at the same time. |
| Alternative Flows  (Các luồng thay thế) | **Alternative flow 1: User cannot find songs searched**   1. From #3 of the basic flow, user enters another keyword 2. Continue step #4 in the basic flow   **Alternative flow 2: User adds another song to playlist**   1. From #6 of the basic flow, user enters another keyword to the “search” field. 2. Back to step #4   **Alternative flow 3: User removes a song from playlist**   1. From #1 of the basic flow, user clicked the existed playlist 2. As the table of the songs displayed, user can search the song that he wants to remove from that playlist 3. User clicks clicked icon “Trash bin” 4. Back to step 2.   **Alternative flow 4: User renames the playlist**   1. From #6 of the basic flow, if the playlist had existed, the user could have renamed this playlist 2. Back to step 6 |
| Pre-conditions  (Điều kiện đầu vào) | User has to log in to the system. |
| Post-conditions  (Điều kiện đầu ra) | New playlist is created.  Other users can search that playlist by name. |

## Use-case: Recognize the song based on Third Party

**Reason to choose this UC:**   This is the function for building Knowledge Base to recognize the songs

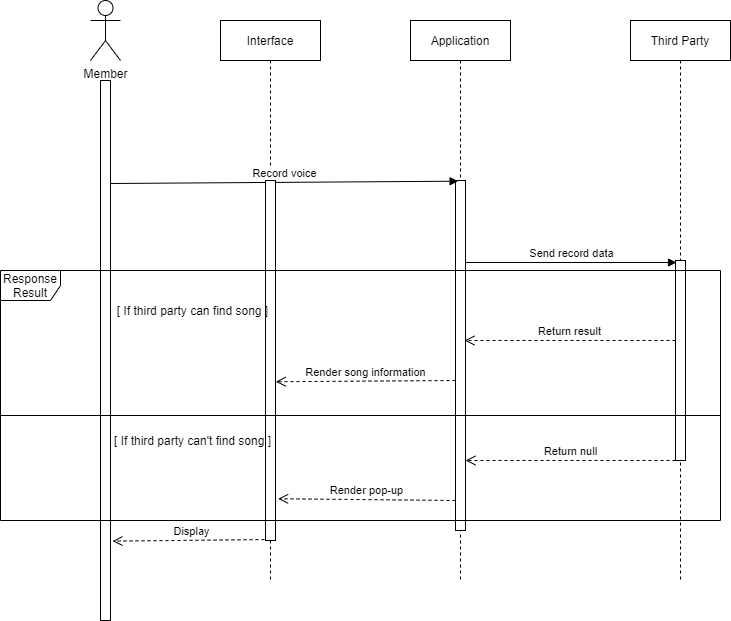
|  |  |
| --- | --- |
| Use case Name  (Tên use-case) | Recognize the song on the behalf of Third Party |
| Brief description  (Mô tả) | This use case describes how the recognize functionality works |
| Actors  (Actors) | Third Party |
| Basic Flow  (Luồng cơ bản) | 1. The Third Party captures read the input sound like input stream and writes to a byte array 2. This byte array is signal recorded in the time domain and frequency domain 3. This array will be converted from the time domain to the frequency domain by the [Discrete Fourier Transform](https://en.wikipedia.org/wiki/Discrete_Fourier_transform) (DFT) that we use the [Fast Fourier transform](https://en.wikipedia.org/wiki/Fast_Fourier_transform) (FFT) numerical algorithms 4. In order to fingerprint a song, we can simply identify the frequency with the highest magnitude each interval of the frequency 5. Then we can start searching the database for matching hash tags. |
| Alternative Flows  (Các luồng thay thế) |  |
| Pre-conditions  (Điều kiện đầu vào) | There must be an initial sound input |
| Post-conditions  (Điều kiện đầu ra) | The application returns the exact song’s result from voice recorded |

# Evaluation architecture style

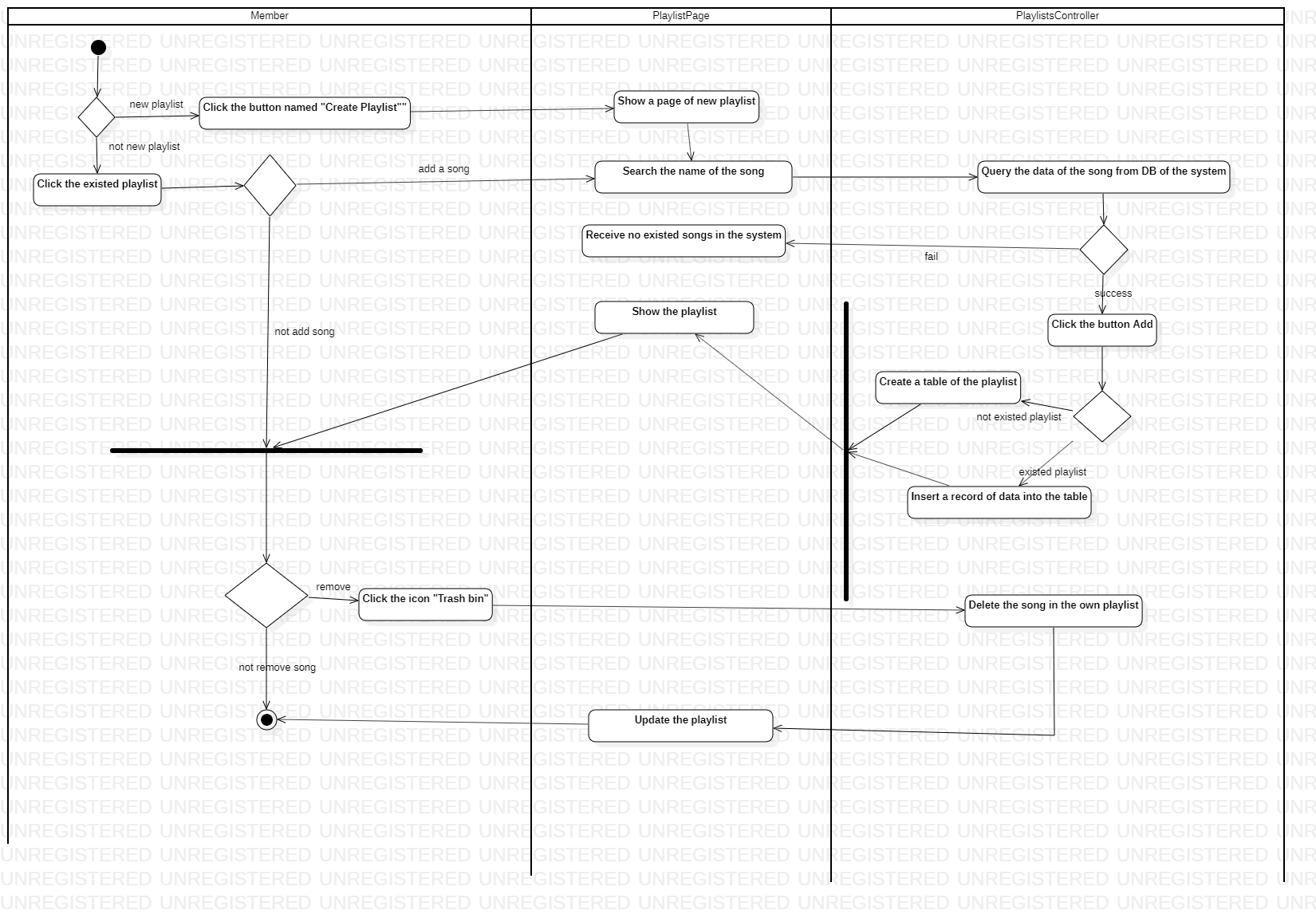
Architecture style that we use for this application is Pipe-Filter. Because we receive the input steam sound that can be broken into a series of processing steps over data streams, and at each step filters consume and move data incrementally. It provides high overall throughput for excessive data processing. Therefore, this architecture style will apply to the UC3: Recognize the song based on Third Party

# UML Design

## UC1: Search song by singing or humming (Sequence Diagram)



## UC2: Manage own playlists (Activity Diagram)



## UC3: Recognize the song based on Third Party (Component Diagram)

