Music Mobile App

Use-Case Specification

Version 1.1

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 26/11/22 | 1.0 | Initial version | Nguyễn Phước Gia Huy  Vũ Đình Duy Khánh  Trần Thị Tuyết Nhung  Phạm Trần Minh Ngọc  Trịnh Thế Sơn |
| 10/12/22 | 1.1 | Change Use-case Model, rename a use-case. | Nguyễn Phước Gia Huy  Phạm Trần Minh Ngọc |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Use-case Model**](#_gjdgxs) **4**

[**2. Use-case Specifications**](#_30j0zll) **4**

[2.1 Use-case: Log in](#_1fob9te) 4

[2.2 Use-case: Log out](#_2et92p0) 5

[2.3 Use-case: Upgrade account](#_1q1rvjen9ert) 5

[2.4 Use-case: Delete account](#_wu1636cxpjdu) 6

[2.5 Use-case: Register account](#_9auj552xqjs) 6

[2.6 Use-case: Play music](#_nhdkz5f7r7gd) 7

[2.7 Use-case: Stop song](#_m201xdqmni7w) 7

[2.8 Use-case: View Information](#_h784tvxsrhhj) 8

[2.9 Use-case: Rate Song](#_3gq8ejphjene) 8

[2.10 Use-case: Recommend Song](#_nyfpncjtmazl) 8

[2.11 Use-case: Search (Songs)](#_qw2kqg7jtnf2) 8

[2.12 Use-case: Search (Artists)](#_w5wiobns0s7j) 9

[2.13 Use-case: Create Playlist](#_iiw3masavqs2) 9

[2.14 Use-case: Add song to playlist](#_gp4be27jdctu) 10

[2.15 Use-case: Remove song out of playlist](#_8s21nu6cd9y0) 10

[2.16 Use-case: Delete playlist](#_9pf45ola8abo) 10

[2.17 Use-case: Download songs](#_vdh6zprh4zrk) 11

[2.18 Use-case: Listen premium songs](#_4w79ngyhwcz4) 11

[2.19 Use-case: Upload song](#_1pq0r03t4h9b) 11

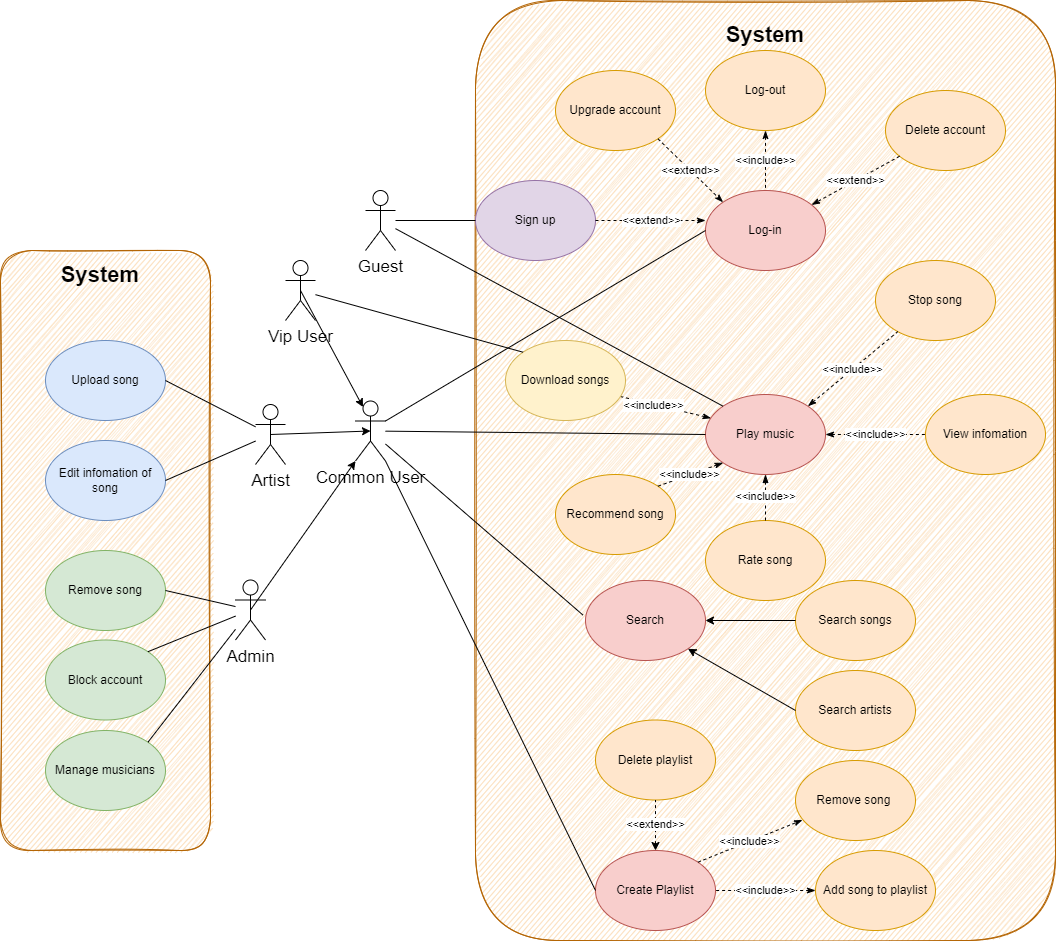
[2.20 Use-case: Edit information of songs](#_ar39oh6w1zvb) 12

[2.21 Use-case: Manage musicians](#_4uttxg44cedb) 12

[2.22 Use-case: Block account](#_l7hdtpxkxbyg) 13

[2.23 Use-case: Remove song](#_y3hohvnwjofv) 13

# Use-case Model



# Use-case Specifications

## Use-case: Log in

| **Use case Name** | Log in |
| --- | --- |
| **Brief description** | This use case describes how a user logs into the Music Mobile App. |
| **Actors** | Common user, Vip user, Artist, Admin. |
| **Basic Flow** | 1. The use case begins when the actor types name and password on the login form. 2. The system validates the actor’s password and logs him/her into the system. 3. The system displays a home page and the use case ends. |
| **Alternative Flows** | **Alternative flow 1: Invalid Name / Password**  In #1, the system cannot find the name or the password is invalid.   1. An error message is displayed. 2. The actor can type in a new name or password or choose to cancel the operation, at which point the use case ends. |
| **Pre-conditions** | The user has account |
| **Post-conditions** | 1. The user has access to the functions of the system. 2. The system displays a home page based on the user type. |

## Use-case: Log out

| **Use case Name** | Log out |
| --- | --- |
| **Brief description** | This use case allows users to logout to the Music Mobile App.  The user clicks on “Logout” and their session is terminated. |
| **Actors** | Common user, Vip user, Artist, Admin. |
| **Basic Flow** | 1. Use case begins when the user/admin clicks on the ‘Logout’ button. 2. The system will move the user session and invalidate the cookie/session. 3. The system redirects to the default page. |
| **Alternative Flows** | There are no alternative flow associated with this use case |
| **Pre-conditions** | 1. The user is logged in. 2. The user no longer wants to be logged in. |
| **Post-conditions** | The user is logged out. |

## Use-case: Upgrade account

| **Use case Name** | Upgrade account |
| --- | --- |
| **Brief description** | For a Common User, they want to be a VIP User so that they can download and listen to premium songs.  This use-case describes how a Common User can be a VIP User by paying money. |
| **Actors** | Common User |
| **Basic Flow** | 1. On the home page, click ‘upgrade’ . 2. System displays a list of VIP’s packages (a month, 3 months or a year, …). 3. Guests choose a package . 4. System displays the package and a paying board. 5. User pays the money. 6. System displays a notification that is successful. 7. Common User becomes a VIP User. |
| **Alternative Flows** | **Alternative flow 1: Payment is fail**   1. From #4 of the basic flow system sends a message to the user to ask to pay again. 2. Get back to step #5 of basic flow. |
| **Pre-conditions** | At the homepage and must be a Common User and login. |
| **Post-conditions** | In the profile, ‘Common User’ becomes ‘VIP User’. |

## Use-case: Delete account

| **Use case Name** | Delete account |
| --- | --- |
| **Brief description** | For users to delete their account |
| **Actors** | Common User |
| **Basic Flow** | 1. At the home page, click ‘delete’ . 2. System display notification to confirm. 3. User clicks ‘yes’. 4. System will remove the account from the database and display notification that is successful. 5. User return to the homepage |
| **Alternative Flows** | **Alternative flow 1: System send error message**   1. From #1 of the basic flow, click again ‘delete’ 2. Continue step #2 |
| **Pre-conditions** | The user was at the home page and had logged in. |
| **Post-conditions** | The user successfully deletes, signs out and to be at homepage. |

## Use-case: Register account

| **Use case Name** | Register account |
| --- | --- |
| **Brief description** | This use case describes how a user creates an account for the Music Mobile App. |
| **Actors** | Guest |
| **Basic Flow** | This use case starts when a system user is not logged in to the system and goes to the login page.   1. The System prompts the user for a username and password or to register a new account. 2. The user selects the registration option. 3. The System prompts user for registration information, Username, password, etc 4. The user enters their information. 5. System verifies information. 6. Creates an account. 7. The use case ends |
| **Alternative Flows** | **Alternative flow 1: Cancel Registration**  In #2:   1. The user selects the cancel option. 2. The system returns the user to the home page without the user being logged in and any information entered has been erased.   **Alternative flow 2: Invalid Information Entered**  In #4:   1. System displays information with appropriate messages to correct invalid information. 2. User re-enters information.   In #5:   1. The system detects false information or information that has been used. 2. The system offers to provide information again. |
| **Pre-conditions** | The user is not logged in to the system. |
| **Post-conditions** | Returned to the home page as a Logged In User. |

## Use-case: Play music

| **Use case Name** | Play music |
| --- | --- |
| **Brief description** | This use case works when the user wants to listen to a song in the Music Mobile App. |
| **Actors** | Guest, Common User, VIP User, Artist and Admin |
| **Basic Flow** | 1. User opens the Music Mobile App. 2. Select the song you want to listen to. 3. System plays music. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Open the Music Mobile App. |
| **Post-conditions** | 1. **Stop song**   Users can stop playing music whenever they don't want to listen anymore.   1. **View information song**   Users can view song information such as singer, songwriter, release date...   1. **Recommend song**   Users can recommend the music they are listening to.   1. **Back-ward**   Users can listen to the music they just listened to before.   1. **For-ward**   Users can skip to the next song.   1. **Rate song**   Users can rate the song. |

## Use-case: Stop song

| **Use case Name** | Stop song |
| --- | --- |
| **Brief description** | This use-case works when the user wants to listen to a song in the Music Mobile App. |
| **Actors** | Guest, Common User, VIP User, Artist, Admin |
| **Basic Flow** | 1. At the song board, click ‘Pause’ .  2. System stops the song being played. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | There must be a song being played. |
| **Post-conditions** | Users can ‘play’ the song to continue listening. |

## Use-case: View Information

| **Use case Name** | View Information |
| --- | --- |
| **Brief description** | This use-case works when common users, vip users want to view information about a song in the Music Mobile App. |
| **Actors** | Common User, VIP User. |
| **Basic Flow** | 1. Users choose a song. 2. User clicks on the ‘Information’ button. 3. System displays information about a song. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Must have a user's account to login. |
| **Post-conditions** | System displays information about a song being chosen. |

## Use-case: Rate Song

| **Use case Name** | Rate Song |
| --- | --- |
| **Brief description** | This use case describes how common users, vip users rate and express their feeling about the song. |
| **Actors** | Common User, Vip User. |
| **Basic Flow** | 1. Users choose a song. 2. User clicks on the ‘Rate’ button. 3. System display comment rate frame. 4. Users write their feelings about the song in the comment box and choose from 1 to 5 stars. 5. System adds the rate data in the comment rate frame, increasing the quantity of the rate comment. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Must have a user's account to login. |
| **Post-conditions** | System successfully adds the rate to the comment rate frame and increases the quantity of the rate comment. |

## Use-case: Recommend Song

| **Use case Name** | Recommend Song |
| --- | --- |
| **Brief description** | This use case describes how common users, vip users rate and express their feeling |
| **Actors** | Common User, Vip User. |
| **Basic Flow** | 1. Users play a song. 2. System suggests similar songs for the user. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Must have a user's account to login. |
| **Post-conditions** | The user can play recommended songs. |

## Use-case: Search (Songs)

| **Use case Name** | Search Song |
| --- | --- |
| **Brief description** | This use case describes how guest, common user, vip user search the song based on keywords. |
| **Actors** | Guest, Common User, Vip User. |
| **Basic Flow** | 1. User clicks on the ‘Search’ bar at the homepage. 2. User enters keywords on the ‘Search’ bar. 3. User clicks on the ‘Search’ button to start searching for a song. 4. System displays the songs found. |
| **Alternative Flows** | **Alternative flow 1: The user cannot find songs searched.**   1. From #2 of the basic flow, the user enters another term. 2. Continue step #3 in the basic flow. |
| **Pre-conditions** | User goes to the homepage. |
| **Post-conditions** | System displays the list of songs based on keywords users enter. |

## Use-case: Search (Artists)

| **Use case Name** | Search Artist |
| --- | --- |
| **Brief description** | This use case describes how guest, common user, vip user search the Artist based on keywords. |
| **Actors** | Guest, Common User, Vip User, Admin |
| **Basic Flow** | 1. User clicks on the ‘Search’ bar at the homepage and choose ‘Artist’. 2. User enters keywords on the ‘Search’ bar. 3. User clicks on the ‘Search’ button to start searching for an Artist. 4. System displays the Artist found. |
| **Alternative Flows** | **Alternative flow 1: The user cannot find Artists searched.**  1. From #2 of the basic flow, the user enters another term.  2. Continue step #3 in the basic flow. |
| **Pre-conditions** | User goes to the homepage. |
| **Post-conditions** | System displays the list of songs based on keywords users enter. |

## Use-case: Create Playlist

| **Use case Name** | Create Playlist |
| --- | --- |
| **Brief description** | This use case describes how the common user, vip user creates a playlist. |
| **Actors** | Common user, vip user. |
| **Basic Flow** | 1. User clicks on "Playlist". 2. System change from current page to playlist page. 3. User clicks the icon to create a playlist. 4. System displays a box to enter the name playlist. 5. User enter name playlist. 6. User clicks the 'Create’ button. 7. System creates a new playlist based on the name which the user entered. |
| **Alternative Flows** | **Alternative flow 1: User enters a name that already exists in the playlist.**   1. System displays a message “this name already exists, Please re-enter.” 2. Continue step #5 in the basic flow.   **Alternative flow 2: User enters a blank name.**   1. System displays a message “Empty Name, Please Enter.” 2. Continue step #5 in the basic flow. |
| **Pre-conditions** | Must have a user's account to login. |
| **Post-conditions** | The user successfully created a playlist. |

## Use-case: Add song to playlist

| **Use case Name** | Add song to playlist |
| --- | --- |
| **Brief description** | This use case describes how the common user, vip user add a song to playlist |
| **Actors** | Common user, vip user. |
| **Basic Flow** | 1. At the song bar, the user clicks a “+” icon to add this song to the playlist. 2. System display list of user playlist. 3. User clicks the playlist which the user wants to add to the song. 4. System adds the song into this playlist. |
| **Alternative Flows** | **Alternative flow 1: User adds the song which existed in the playlist.**   1. System displays a message “this song already exists in the playlist.” 2. System cancel adds this song. |
| **Pre-conditions** | Must have a user's account to login, playlist is valid. |
| **Post-conditions** | The user successfully adds songs into the playlist. |

## Use-case: Remove song out of playlist

| **Use case Name** | Remove songs from the playlist. |
| --- | --- |
| **Brief description** | This use case describes how the common user, vip user removes a song from the playlist. |
| **Actors** | Common user, vip user. |
| **Basic Flow** | 1. User clicks on ‘Playlist’. 2. System change from current page to playlist page. 3. User clicks the playlist which has the song the user wants to remove. 4. System displays a list of songs in this playlist the user chooses. 5. User clicks the icon ‘-’ to remove the song in the playlist. 6. System will ask the user if you are sure you want to remove this song from this playlist. 7. If the user clicks “Yes”, continue to the next step. If the user clicks “No”, cancel the remove and go back to step. 8. System removes the song from the playlist. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Must have a user's account to login. |
| **Post-conditions** | The user successfully removes the song from the playlist. |

## Use-case: Delete playlist

| **Use case Name** | Delete playlist |
| --- | --- |
| **Brief description** | This use case describes how the common user, vip user deletes a playlist. |
| **Actors** | Common user, vip user. |
| **Basic Flow** | 1. User is at the playlist page, user clicks ‘delete’. 2. System display notification to confirm. 3. User clicks ‘yes’. 4. System will remove the playlist from the database. 5. System displays a notification that is successful. 6. User returns to the users page. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Must have a user's account to login and have a playlist. |
| **Post-conditions** | The user can not access the playlist anymore. |

## Use-case: Download songs

| **Use case Name** | Download Song |
| --- | --- |
| **Brief description** | This use case describes how the Vip User downloads songs. |
| **Actors** | Vip User. |
| **Basic Flow** | 1. At the song bar, the user clicks on the ‘Download’ button. 2. System will ask the user if you are sure you want to download this song. 3. If the user clicks “Yes”, continue to the next step. If the user clicks “No”, cancel the download and go back to step. 4. System will get data of this song and add it to the user's download folder. 5. System displays a successful download message to the user. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Must have a vip user’s account to login. |
| **Post-conditions** | The user successfully downloads songs. |

## Use-case: Listen premium songs

| **Use case Name** | Listen premium songs |
| --- | --- |
| **Brief description** | There are some premium songs that other users can not listen to. These can be listened to by VIP Users. |
| **Actors** | VIP User |
| **Basic Flow** | 1. VIP User clicks a special song. 2. System plays the song. |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | Must login and be VIP User. |
| **Post-conditions** | VIP User successfully listen this song, and can add this song to playlist |

## Use-case: Upload song

| **Use case Name** | Upload songs |
| --- | --- |
| **Brief description** | For artist to up their songs to app’s server |
| **Actors** | Artist |
| **Basic Flow** | 1. At the homepage, the Artist clicks the button to upload. 2. System displays the list files of Artist phones. 3. Artists choose their sound file that they want to upload and click ‘Upload’. 4. Their phone sends the file to the server and the system displays a menu of song information. 5. Artists fill information in the menu and click ‘apply’. 6. System will display notification that is successful or not |
| **Alternative Flows** | **Alternative flow 1: User cannot up file to server**   1. From #3 of the basic flow, choose again sound file and click ‘Upload’ 2. Continue step #4   **Alternative flow 2: System send notification that is not successful**   1. From #5 of the basic flow, system display again menu that user has filled 2. Continue step #6 |
| **Pre-conditions** | User need to login the app |
| **Post-conditions** | The user successfully uploads a new song. |

## Use-case: Edit information of songs

| **Use case Name** | Edit information of songs |
| --- | --- |
| **Brief description** | For artist to edit their information of songs |
| **Actors** | Artist |
| **Basic Flow** | 1. At the user page, click the song to be edited and click ‘edit’. 2. System display menu of song information. 3. Artists enter the new information and click ‘apply’. 4. System will display notification that is successful or not |
| **Alternative Flows** | **Alternative flow 1: System send notification that is not successful**   1. From #3 of the basic flow, system display again menu that user has filled 2. Continue step #4 |
| **Pre-conditions** | User needs to login to the app and be at the user page. |
| **Post-conditions** | The user successfully edits new information for the song. |

## Use-case: Manage musicians

| **Use case Name** | Manage musicians |
| --- | --- |
| **Brief description** | Can see the profile of each musician, see notification if they upload files, receive reports from users, and change information. |
| **Actors** | Admin |
| **Basic Flow** | 1. At the home page, the system shows notification if there is some news. 2. Admin click link in the report. 3. System move admin to the page that was notified. 4. Admin can edit information . |
| **Alternative Flows** | There are no alternative flows associated with this use case. |
| **Pre-conditions** | The user was at the home page and had logged in. |
| **Post-conditions** | The user can see all information about the songs and artists that are public. |

## Use-case: Block account

| **Use case Name** | Block account |
| --- | --- |
| **Brief description** | Admin can block the account of the user that has been suitable (like spamming in comment, using inappropriate language or upload unacceptable files, …) |
| **Actors** | Admin |
| **Basic Flow** | 1. At the user page, click ‘block’ . 2. System display notification to confirm. 3. Admin click ‘yes’. 4. System will remove account into ‘blacklist’ that is not allow user to comment, upload songs, download music,… (user become a ‘guest’) 5. System displays a notification that is successful. 6. Admin return to the users page |
| **Alternative Flows** | **Alternative flow 1: User have been already blocked**   1. From #2 of the basic flow system sends a message that the user has been blocked. 2. Move to step #6 |
| **Pre-conditions** | The Admin was at the user page and had logged in. |
| **Post-conditions** | The Admin successfully blocks the user and is at the user page. |

## Use-case: Remove song

| **Use case Name** | Remove song |
| --- | --- |
| **Brief description** | For Admins, that allows them to remove songs that are not suitable away from the server. |
| **Actors** | Admin |
| **Basic Flow** | 1. At the song board, click ‘remove’ . 2. System display notification to confirm. 3. Admin click ‘yes’. 4. System will remove songs from the database. 5. System displays a notification that is successful. 6. Admin return to the users page |
| **Alternative Flows** | **Alternative flow 1: there is admin that removed the song earlier**  1. From #5 of the basic flow system sends a message for the user to notify the song was already removed.  2. Move to step #6 of basic flow |
| **Pre-conditions** | Admin must be at the song board. |
| **Post-conditions** | The user can not access the song any more. |

....