ChillnFree

Use-Case Specification

Version <0.1>

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 29/10/2021 | 0.1 | Detail all the use-case | Võ Hoàng Bảo Duy  Trần Ngọc Lam  Lê Minh Sĩ  Đặng Kiệt Hào  Lê Quốc Trọng |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

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

[**Use-case Specifications**](#_30j0zll) **5**

[2.1 Account.](#_hzbvcscmdpud) 5

[2.1.1 Account based on Google's API/Facebook’s API.](#_dno1suiz4mqm) 5

[2.2 Search music.](#_idz2pwrrt120) 5

[2.3 Rating.](#_lz1vw5ew8yu6) 5

[2.4 Comment.](#_ifih3q4ec9mk) 6

[2.5 Create their playlists.](#_ppwwoz9qpe9t) 6

[2.6 Publish the playlists.](#_frthwpdqehaj) 7

[2.7 Upload a song](#_ixenek9g6mlu) 7

[2.8 Download a song.](#_76j920ixmbaa) 8

[2.9 User notification of new music.](#_wfyzp1tm6rcb) 8

[2.10 Add music to the playlist.](#_jga2mvwuyvho) 9

[2.11 Play music.](#_vz695tvf1ynq) 9

[2.12 Use a pomodoro timer.](#_v57ri2155qna) 9

[2.13 Report/Contact to the support team](#_ngdwjgayjomf) 10

[2.14 Delete the playlist](#_38fbl1o3g7u6) 10

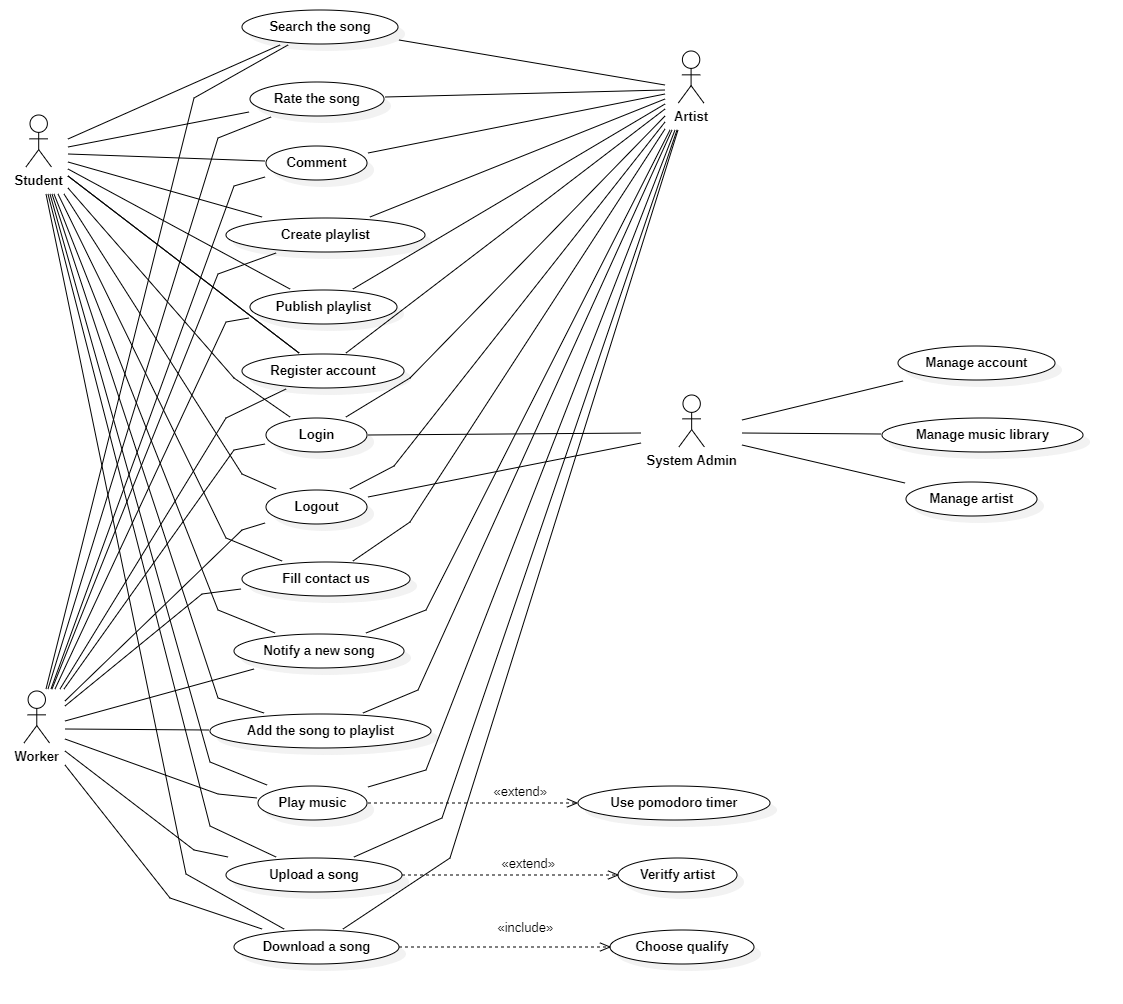
[2.15 Follow other users](#_l500k6f3ehxi) 11

[2.16 Manage user’s account](#_l511dqmgzq3t) 11

[2.17 Manage music library](#_zgrpos584vtn) 12

[2.18 Manage artist](#_mm44bhfq5klg) 12

# Use-case Model



# Use-case Specifications

## 2.1 Account.

### **2.1.1 Account based on Google's API/Facebook’s API.**

| Use case Name | Account bases on Google's API/Facebook’s API |
| --- | --- |
| Brief description | This use case describes how the user can manage their account in our system. |
| Actors | User. |
| Basic Flow | 1. From the homepage, the user clicks on the “Sign in” button. 2. The user clicks “Continue with Google/Facebook”. 3. The user will be directed to Google’s/Facebook’s OAuth2 website to sign in. 4. The user signs in through their gmail/facebook account. 5. The user will be brought back to the homepage with a logged account. |
| Alternative Flows | **Alternative Flow 1: User entering wrong email or password:**  OAuth 2 will handle it.  **Alternative Flow 2: First time logging in.**  The user will be brought to the edit profile page to fill in their information. |
| Pre-conditions | Users need to connect to the website. |
| Post-conditions | Users successfully log-in to the website. |

## 2.2 Search music.

| Use case Name | Search for music. |
| --- | --- |
| Brief description | This use case describes how the user can search for music. |
| Actors | User. |
| Basic Flow | 1. From the homepage, the user enters the song name in the “search” field. 2. User clicks on the “Search” button or presses enter. 3. System displays the music found. 4. The user can interact with the displayed music. |
| Alternative Flows | **Alternative Flow: No music was found.**   1. From #3 In the basic flow, there will be “no music was found” displayed instead of music. 2. User can re-enter their music name in the field and retry from #2 from the basic flow. |
| Pre-conditions | User searches for music. |
| Post-conditions | Found music displayed on the screen. |

## 2.3 Rating.

| Use case Name | Rate music. |
| --- | --- |
| Brief description | The use case describes how the user can rate the music. |
| Actors | User. |
| Basic Flow | 1. There will be an “upvote” and “downvote” button in the music that the user is listening to. 2. The user clicks on one of those buttons to rate the song. 3. The button will light up and the user’s rate will be submitted to the system. |
| Alternative Flows | **Alternative Flow 1: The user clicks the button again.**   1. The button will change to its original color. 2. The user’s rate will be removed.   **Alternative Flow 2: The user clicks the button when their counterpart is active.**   1. The counterpart button will change to its original color. 2. The clicked button will light up. 3. The user’s rate will be updated. |
| Pre-conditions | The user has to sign in to the system. |
| Post-conditions | The user successfully rates the song in the system. |

## 2.4 Comment.

| Use case Name | Comment music |
| --- | --- |
| Brief description | The use case describes how the user can comment on the music |
| Actors | User |
| Basic Flow | 1. There will be a comment section in the music that the user is listening to 2. User types his/her comment into the form 3. User clicks the button “Post” to submit his/her comment to the website |
| Alternative Flows | None |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user successfully comments on the music on the system |

## 2.5 Create their playlists.

| Use case Name | Create playlists |
| --- | --- |
| Brief description | The use case describes how the user can create own playlists |
| Actors | User |
| Basic Flow | 1. At the Library page, user goes to the Playlists tab 2. There will be a button “Create new playlist” and user has to fill up the information about new playlist(Name, Mode) 3. The new playlist created successfully will appear in the Playlists tab |
| Alternative Flows | **Alternative Flow 1: The user creates new playlist that already exists**   1. After #2 of the basic flow, there will be an alert showing up on the website 2. The system will ask the user to change the name of new playlist or delete the existing playlist |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user successfully creates the playlist on the system |

## 2.6 Publish the playlists.

| Use case Name | Publish playlists |
| --- | --- |
| Brief description | The use case describes how the user can share own playlists |
| Actors | User |
| Basic Flow | 1. At the Library page, user goes to the Playlists tab 2. There will be a list of user’s playlists showing up 3. User clicks into a playlist that he/she wants to share 4. There will be a button “Share” and user can get a link to share to everyone |
| Alternative Flows | **Alternative Flow 1: The shared playlist is created in Private mode**   1. After #4 of the basic flow, there will be an alert showing up on the website 2. The system will ask the user to change the mode of the playlist to Public for sharing |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user successfully gets the link of the shared playlist |

## 2.7 Upload a song

| Use case Name | Upload a song |
| --- | --- |
| Brief description | This use case describes how the user can upload a song from their devices. |
| Actors | User |
| Basic Flow | 1. At the homepage (or any page that has the ‘Upload’ feature), the user can upload a song to the system by clicking ‘Upload’. 2. User chooses a song from their devices and clicks ‘Select’/’Open’ from windows. 3. A song will be uploaded to the system temporarily. 4. The system displays the original name of the song, user can change the name of the song. 5. User clicks ‘Upload’ (to upload to the server and publish to the website). 6. Songs will be uploaded to the database of the system. |
| Alternative Flows | **Alternative flow 1: User does not choose songs to upload.**   1. From #2 of the basic flow, if the user does not choose a song to upload. 2. Back to #1 of the basic flow.     **Alternative flow 2: Song is uploaded to the temporary memory but the user does not click ‘Upload’ (to confirm) and move to another page.**   1. From #4 of the basic flow, users uploaded a song from devices to the system and users don’t confirm ‘Upload’. 2. Browser displays a warning (having two selections: Cancel and OK) if the user wants to move to another page. 3. If the user chooses ‘Cancel’, the user will stay on the current page. 4. If the user chooses ‘OK’, the system will delete the song from temporary memory and the user will move to the page that the user chose. |
| Pre-conditions | User has to sign in to the system. |
| Post-conditions | The user successfully uploads a song to the system (website). |

## 2.8 Download a song.

| Use case Name | Download a song |
| --- | --- |
| Brief description | This use case describes how the user can download a song from the website. |
| Actors | User |
| Basic Flow | 1. User searches and chooses a song to download. 2. User clicks the 'Download’ button on the expected song. 3. System displays sound available quality for the user to choose. 4. User chooses the desired sound quality. 5. System starts to download the song and stores it in the download location saved in the browser. |
| Alternative Flows | **Alternative flow 1: User downloads a playing track.**   1. User clicks the 'Download’ button at the tool bar of the song. 2. Continue step #2 in the basic flow.     **Alternative flow 2: Song is not allowed to download**   1. From #2 of the basic flow, the song has the ‘Downloads’ button disabled, which means this song is not allowed to be downloaded by the poster. 2. User can choose to download from other publishers or search for another song. |
| Pre-conditions | User goes to the homepage. |
| Post-conditions | The user successfully downloads a song to his/her device. |

## 2.9 User notification of new music.

| Use case Name | User notification of new music |
| --- | --- |
| Brief description | This use case describes how the user receives a notification from the website. |
| Actors | User |
| Basic Flow | 1. User searches and chooses another user to follow and receive notification. 2. User clicks the 'Follow’button on the expected artist. 3. System notifies the users through their email whenever the artist they follow has a new update. |
| Alternative Flows | None |
| Pre-conditions | User needs to sign in to the website. |
| Post-conditions | User is able to access notifications. |

## 2.10 Add music to the playlist.

| Use case Name | Add music to the playlist |
| --- | --- |
| Brief description | This use case describes how the user adds a song to his/her playlist. |
| Actors | User |
| Basic Flow | 1. User clicks the 'Add to playlist’ button on the expected songs. 2. The system displays all the playlist created by the user. 3. User chooses the playlist to add the song. 4. System notifies the users about adding successfully. |
| Alternative Flows | **Alternative flow : No playlists have been created yet.**   1. From #5 of the basic flow, the system supports the user fastly create a playlist by entering the playlist name and choosing private/public mode. 2. Continue step #6 in the basic flow. |
| Pre-conditions | User needs to sign in to the website. |
| Post-conditions | User is able to add a song to their playlist. |

## 2.11 Play music.

| Use case Name | Play music |
| --- | --- |
| Brief description | This use case describes how the user plays a song. |
| Actors | User |
| Basic Flow | 1. At the homepage, the user clicks on a song they want to listen to. 2. System displays a song they want to play. 3. User can click the ‘play’ button again to stop. |
| Alternative Flows | **Alternative flow 1: Play a song at another page.**   1. From #1 of the basic flow, user can choose a song to play after searching for a song want to play 2. Continue step #2 in the basic flow.   **Alternative flow 2: Play a playlist.**   1. From #1 of the basic flow, user can choose a playlist want to play 2. Continue step #2 in the basic flow.   **Alternative flow 3: Click another button.**   1. From #3 of the basic flow, user can click the ‘next’ button to play a next song or ‘previous' button to play a previous song. |
| Pre-conditions | User selects a song they want to play. |
| Post-conditions | User can listen to the song they choose. |

## 2.12 Use a pomodoro timer.

| Use case Name | Use a pomodoro timer. |
| --- | --- |
| Brief description | This use case describes how the user uses a pomodoro timer. |
| Actors | User |
| Basic Flow | 1. At the listening page, the user clicks the ‘clock’ button in the player control. 2. System displays a pomodoro timer at this page. 3. User clicks the 'pomodoro’ button to set a countdown time. 4. User clicks the ‘play’ button in the middle of the clock to start the countdown. |
| Alternative Flows | **Alternative flow 1: Set another time.**   1. From #3 of the basic flow, user can click the ‘short break’ or ‘long break’ button to set a countdown break time. 2. Continue step #4 in the basic flow.   **Alternative flow 2: Set a custom time.**   1. From #3 of the basic flow, user can customize their countdown time by clicking on the ‘plus’ or ‘minus’ button to increase/decrease the countdown time. 2. Continue step #4 in the basic flow. |
| Pre-conditions | At the listening page. |
| Post-conditions | The user can use a pomodoro timer. |

## 2.13 Report/Contact to the support team

| Use case Name | Report/Contact to the support team |
| --- | --- |
| Brief description | This use case describes how the user can contact or report to the support team |
| Actors | User |
| Basic Flow | 1. Having the form at the end of each page, users can report or contact the support team here. 2. User fills in their email and message of the form (both aren’t null). 3. User clicks the button ‘Send’. 4. User will receive notifications (successfully or not). 5. The system sends the message of users to the support team. 6. The support team will respond to users soon. |
| Alternative Flows | **Alternative flow 1: User does not fill in email or message of the form.**   1. From #1 of the basic flow, the user must fill in the email and message of the form. 2. Notify the error until the user fills in the full information of the form. |
| Pre-conditions | User goes to pages of systems (signed in or not). User wants to report or contact the support team about the error of the system or something. |
| Post-conditions | The user successfully sends the message to the support team and the user will receive a response from the support team. |

## 2.14 Delete the playlist

| Use case Name | Delete playlists |
| --- | --- |
| Brief description | The use case describes how the user can delete own playlists |
| Actors | User |
| Basic Flow | 1. At the Library page, the user goes to the Playlists tab. 2. There will be a list of user’s playlists created. 3. User clicks into a button “X” which is next to the playlist that the user wants to remove. 4. The chosen playlist will disappear from the Playlists tab |
| Alternative Flows | **Alternative Flow: No playlist is already created**  After #1 of the basic flow, there will be not a list of playlists displayed |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user successfully delete the playlist on the system |

## 2.15 Follow other users

| Use case Name | Follow other users |
| --- | --- |
| Brief description | This use case describes how the user can follow other users (artist, ...) |
| Actors | User |
| Basic Flow | 1. At each song/album/playlist, it has the author of the song/playlist/album, user can follow the author here. 2. User clicks ‘Follow’ to follow the author to receive a notification when the author publishes a song/playlist/album. 3. Now, the system displays the ‘Unfollow’ button. 4. The system adds the author to the following list of users. |
| Alternative Flows | **Alternative flow: User clicks ‘Unfollow’**   1. From #3 of the basic flow, the user clicks ‘Unfollow’. 2. Now, the system displays the ‘Follow’ button. 3. The system removes the author from the following list of users. |
| Pre-conditions | User has to sign in to the website. |
| Post-conditions | The user successfully adds the author to the following list of the user. |

## 2.16 Manage user’s account

| Use case Name | Manage user’s account |
| --- | --- |
| Brief description | This use case describes how administrators can manage users’ accounts. |
| Actors | Administrator |
| Basic Flow | 1. At the control dashboard of the administrator, the administrator clicks ‘Manage user’s account’. 2. The system shows all of users’ accounts, the administrator can search by the account name to find the user. 3. The administrator can remove or restrict the users’ account when they violate terms of the website . 4. All changes will be updated by the system. |
| Alternative Flows | None |
| Pre-conditions | Administrator has to sign in to the system by admin account. |
| Post-conditions | Administrator successfully manages users’ accounts. |

## 2.17 Manage music library

| Use case Name | Manage music library. |
| --- | --- |
| Brief description | This use case describes how administrators can manage music. |
| Actors | Administrator |
| Basic Flow | 1. At the control dashboard of the administrator, the administrator clicks ‘Manage music library’. 2. The system displays all songs. 3. The administrator can view song details by clicking the ‘view’ button next to the song. 4. The administrator can remove songs that violate copyright or don’t match community standards. 5. All changes will be updated by the system. |
| Alternative Flows | None |
| Pre-conditions | Administrator has to sign in to the system by admin account. |
| Post-conditions | Administrator successfully manages users’ accounts. |

## 2.18 Manage artist

| Use case Name | Manage artist |
| --- | --- |
| Brief description | This use case describes how administrators can manage artists. |
| Actors | Administrator |
| Basic Flow | 1. At the administrator’s dashboard, the admin clicks the “Manage Artists” button. 2. The admin will be directed to the “Manage Artists” page, this page contains a list of users that are granted “Artist” badge and add/remove button. 3. The admin clicks the “add”/”remove” button, there will be a field to enter the username. 4. The admin enters a username and press enter, there will be a confirmation box appear. 5. The admin presses the “Yes” button of the box, the user that admin entered will be granted/revoke “Artist” status. |
| Alternative Flows | **Alternative Flow 1: Admin clicks no at #5.**  The confirm box closes and nothing will be executed.  **Alternative Flow 2: Admin enters a username at #4 that is already an Artist if the “add” button is pressed or not an Artist if the “remove” button is pressed.**  A notice box appears that says “This user is already an Artist”/”This user is not an Artist” and fades away after a few seconds. |
| Pre-conditions | Administrator has to sign in to the system by admin account. |
| Post-conditions | Administrator successfully manages artists. |