ChillnFree

Use-Case Specification

Version <0.2>

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 |
| 23/11/2021 | 0.2 | Fix use-case model  Add some new use-cases and fix existing use-cases | Võ Hoàng Bảo Duy  Lê Minh Sĩ |
|  |  |  |  |
|  |  |  |  |

Table of Contents

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

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

[2.1 Login.](#_hzbvcscmdpud) 4

[2.2 Search music.](#_idz2pwrrt120) 5

[2.3 Rating.](#_5gme8q71un06) 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.](#_f6w3h097ork1) 10

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

[2.14 Delete the playlist](#_38fbl1o3g7u6) 11

[2.15 Follow other users](#_l500k6f3ehxi) 11

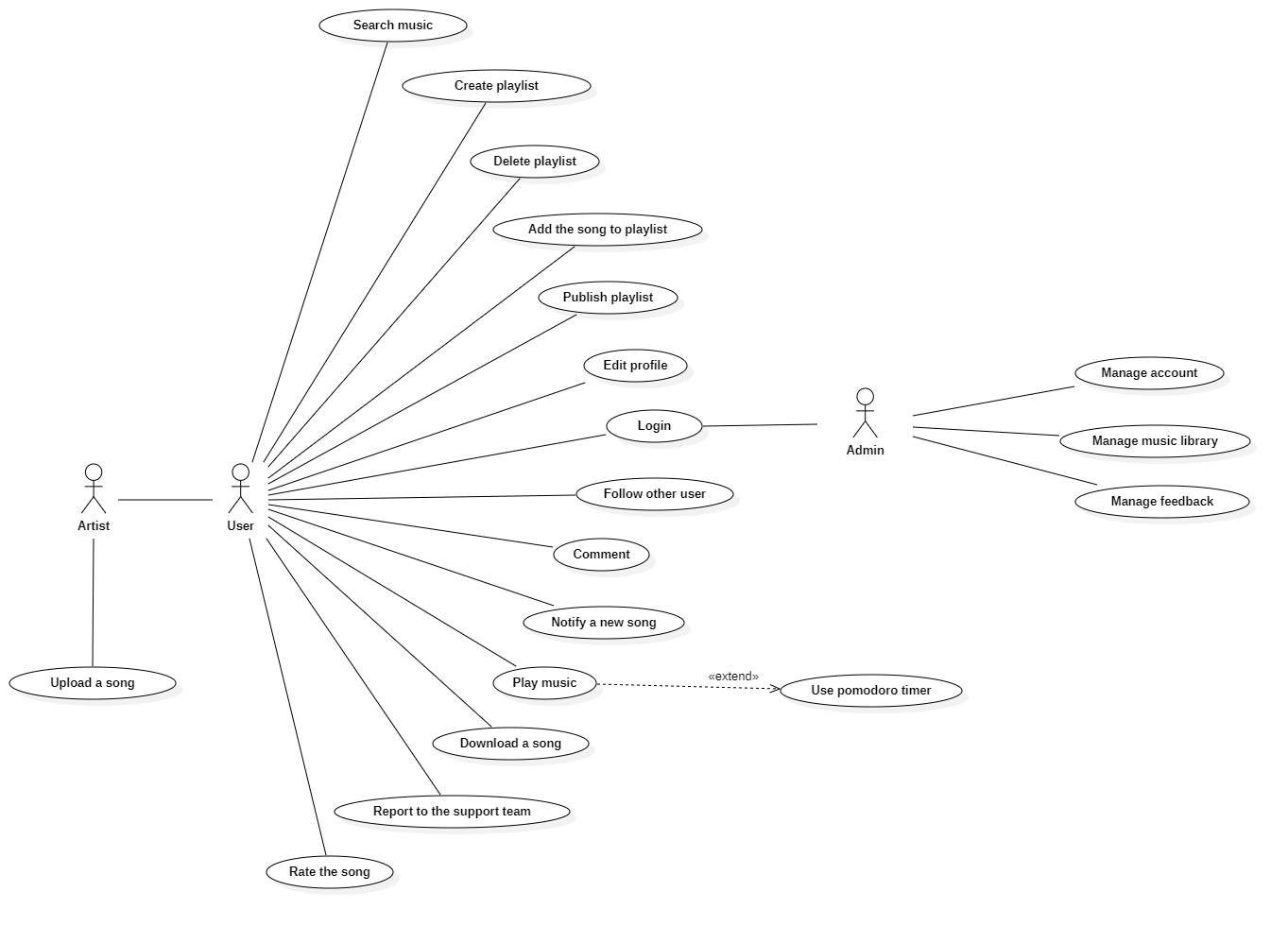
[2.16 Edit profile](#_l511dqmgzq3t) 11

[2.17 Manage user’s account](#_dpr3rm1h9gbl) 12

[2.18 Manage music library](#_zgrpos584vtn) 12

[2.19 Manage report/contact from users](#_pvcvaqbusikd) 13

# Use-case Model



# Use-case Specifications

## 2.1 Login.

| Use case Name | Login through Google’s account. |
| --- | --- |
| Brief description | This use case describes how the user can log in our system. |
| Actors | User. |
| Basic Flow | On the homepage:   * The user clicks on the “Sign in” button. * System will direct the user to Google’s OAuth2 website to log in. * Google account system will let the user logs in or signs up * The user logs in through their gmail account or signs up. |
| Alternative Flows | **Alternative Flow 1: User entering wrong email or password:**   * Google’s OAuth2 will handle it. |
| Pre-conditions | Users need to connect to the website. |
| Post-conditions | * Users successfully log-in to the website. * The user will go back to the homepage with a logged account. |
| Extension points | **Trigger event - First time logging in:**   * The user will be brought to the edit profile page to fill in their information. * The system will then save user information into the database. |

## 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 | On the homepage:   * The system will display a search field for the user. * The user types into that field a name. * The user clicks the “Search” button or press enter after typing their song. * The system gets the name and query all items that are suitable. * The system will display all music that is found in the system. |
| Alternative Flows | **Alternative Flow: No music was found.**   * The system will display “No music found” text instead of song. |
| Pre-conditions | User searches for music. |
| Post-conditions | Found music displayed on the screen. |
| Extension points | None |

## 

## 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 | On the music page:   * The system will display an “upvote” and “downvote” button for the user. * The user clicks the “upvote” or“downvote” button. * The system will get the user action and submit into the system. * The system will change the color of the button of the user’s view. |
| Alternative Flows | **Alternative Flow 1: The user clicks the button again.**   * The system will get the user’s action and submit it into the system. * The system will change the color of the button of the user’s view.   **Alternative Flow 2: The user clicks the button when their counterpart is active.**   * The system will get the user’s action and submit it into the system. * The system will change the color of the counterpart button to the original of the user’s view. * The system will change the color of the clicked button of the user’s view. |
| Pre-conditions | The user has to sign in to the system. |
| Post-conditions | The user successfully rates the song in the system. |
| Extension points | None |

## 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. On "Listen music page", user will see a comment section 2. User writes down own thoughts about music that user is listening on the comment input box 3. User clicks "Post" button to submit the comment 4. System will record that comment and display on comment section |
| Alternative Flows | **Alternative Flow: The user leaves the comment input box blank and submits**   * The system will send a message to warn user |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user will get a success or fail message |
| Extension points | None |

## 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. On "Create playlist page", user clicks "Create new playlist" button 2. System will display an information form of new playlist 3. User fills in the required information (Name, Mode...) and submits 4. System will confirm and store that playlist then send a notification and display the playlist to the user |
| Alternative Flows | **Alternative Flow 1: The user creates new playlist that already exists**   * The system will send a message that asks user to change new playlist’s name   **Alternative Flow 1: The user leave the new playlist’s required information blank**   * The system will send a message that asks user to fill in that blank field |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user will get a success or fail message |
| Extension point | User could select Help to view a demo |

## 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. On "Playlist page", user will see a list of playlists 2. User chooses a playlist and clicks “Share” button 3. System will confirm and send a notification to user with a public link |
| Alternative Flows | **Alternative Flow 1: The shared playlist is created in Private mode**   * The system will ask the user to change the mode of the playlist to “Public” first for sharing function |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user will get a public link of shared playlist |
| Extension point | None |

## 2.7 Upload a song

| Use case Name | Upload a song |
| --- | --- |
| Brief description | This use case describes how the artist can upload a song from their devices. |
| Actors | Artist |
| Basic Flow | 1. At the ‘Upload page’, the artist can upload a song to the system. 2. The Artist 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, artist, genre and picture of the song (there are defaults of the song). 5. Artist can change the name, artist, genre of the song. 6. Artist clicks ‘Upload’ (to upload to the server and publish to the website). 7. Songs will be uploaded to the database of the system. |
| Alternative Flows | **Alternative flow 1: Artist does not choose songs to upload.**   1. From #2 of the basic flow, if the artist 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 artist does not click ‘Upload’ (to confirm) and moves to another page.**   1. From #4 of the basic flow, the artist uploads a song from devices to the system and the artist does not 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 artist 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 artist chose.   **Alternative flow 3: Artist chooses the song that does not have the field: original name, picture, artist and genre from #3 of the basic flow**   1. From #4 of the basic flow, the system will display text ‘**undefined**’ with the nothing field. 2. Back to #5 of the basic flow |
| Pre-conditions | Artist has to sign in to the system with an artist account. |
| Post-conditions | The Artist successfully uploads a song to the system (website)  Trigger an event to send a notification to Artist to say that a song was sent to the system |
| Extension point | None |

## 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. |
| Extension point | None |

## 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. |
| Extension point | None |

## 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. |
| Extension point | None |

## 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. |
| Alternative Flows | **Alternative flow: Select a song that is currently playing.**   1. From #2 of the basic flow, the system continues to play the current song. |
| Pre-conditions | User selects a song they want to play. |
| Post-conditions | Display the music play screen along with playing the selected song. |
| Extension points | * User can play a song on another page. * User can click the ‘play’ button again to stop, the ‘next’ button to play a next song or ‘previous' button to play a previous song. * User can select a playlist to listen to all the songs in it. |

## 

## 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: Go to another page**   * A pomodoro timer will be deactivated and the time will return to default. |
| Pre-conditions | At the listening page. |
| Post-conditions | The user can use a pomodoro timer. |
| Extension points | * User can set another time by clicking the ‘short break’ or ‘long break’ button to set a countdown break time. * User can set a custom time by clicking on the ‘plus’ or ‘minus’ button to increase/decrease the countdown time. |

## 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. In the form of report/contact, 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 the user 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.  Trigger an event sends a notification to the user to say that the message of the user was sent to the support team. |
| Extension point | None |

## 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. On "Playlist page", user will see a list of playlists 2. User chooses a playlist and clicks “Delete” button 3. The system will send a notification to user and refresh the playlist page |
| Alternative Flows | **Alternative Flow: No playlist is already created**   * No playlist will display for user to choose |
| Pre-conditions | The user has to sign in to the system |
| Post-conditions | The user will get a success notification |
| Extension point | None |

## 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. |
| Extension point | None |

## 2.16 Edit profile

| Use case Name | Edit profile |
| --- | --- |
| Brief description | This use case describes how users edit their profile |
| Actors | User/artist. |
| Basic Flow | 1. At the profile page, the user clicks the button “Edit profile” at the bottom of the page. 2. The system shows a form for user to enter new information. 3. After entering all the information, user clicks the “Save” button to save the new profile. 4. All changes will be saved by the system. |
| Alternative Flows | **Alternative Flow 1: User exits the edit profile page before clicking the “Save” button.**  A message will pop up to inform that all the changes would not be saved if the user exits the page.  **Alternative Flow 2: User leaves required information blank.**  Notify the error until the user fills in the required information of the form. |
| Pre-conditions | Administrator has to sign in to the system by admin account. |
| Post-conditions | Users can successfully change their profile. |
| Extension point | None |

## 2.17 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 ban or restrict the users’ activities on the website 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. |
| Extension point | None |

## 2.18 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. |
| Extension points | None |

## 2.19 Manage report/contact from users

| Use case Name | Manage reports/contact from users. |
| --- | --- |
| Brief description | This use case describes how administrators can manage reports/contact for the user.. |
| Actors | Administrator |
| Basic Flow | On the manage feedback page:   * The system will direct the admin to the “Manage Reports” page, this page contains a list of users’ feedback. * The admin clicks the user’s email and chooses the “Reply” button, there will be a text field for the administrator to reply to the user's issues. |
| Alternative Flows | None |
| Pre-conditions | Administrator has to sign in to the system by admin account. |
| Post-conditions | Administrator successfully manages feedback/report/contact from users. |
| Extension point | None |