Sportifry

Use-Case Specification

Version <1.0>

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 08/11/2021 | 1.0 | operate some basic functions | Tran Kha Tri |
| 22/11/2021 | 2.0 | Modify model and complete document | Tran Kha Tri, T.V.Phat |
|  |  |  |  |
|  |  |  |  |

Table of Contents

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

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

[Use-case: Login](#_x3zi3jnz3lm5) 4

[Use-case: Sign up](#_lg0341hwd1lz) 5

[Use-case: Listen music](#_qqw5dgyjh3dj) 6

[Use-case: View music detail](#_q10j195egpge) 6

[Use-case: Search music](#_399sacq82u17) 6

[Use-case: Create todolist](#_mawmxrkzegf1) 7

[Use-case: Read to-do-list](#_8s56vx10trfl) 7

[Use-case: Delete to-do-list](#_6xkg0s8b192o) 8

[Use-case: Update to-do-list](#_nr548k94ekzr) 8

[Use-case: comment](#_t6iii5ik73q2) 8

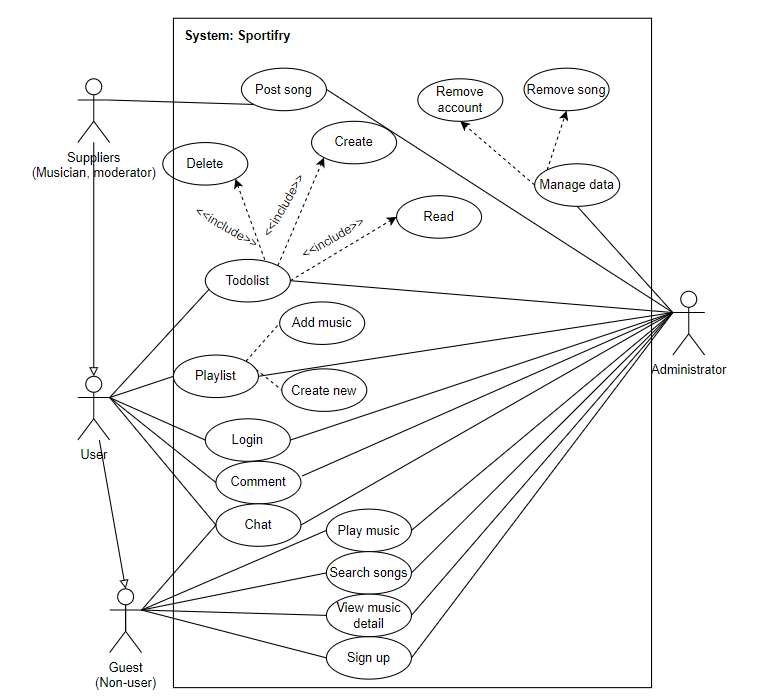
[Use-case: chatbox](#_3gjpgblvocje) 9

[Use-case: add music](#_xlpjpivrfuv5) 9

[Use-case: remove music](#_n3gm6g3oiwm) 10

[Use-case: remove account](#_g8rvrqdngiho) 10

# Use-case Model



# Use-case Specifications:

## Use-case: Login

| Use case Name | Login |
| --- | --- |
| Brief description | This use case describes how the user who registered an account logins software |
| Actors | Audiences (members), suppliers, administrators |
| Basic Flow | At the ‘Login’ page,   1. Website displays 2 fields: ‘Username’ and ‘Password’. Below that have 2 buttons: ‘Sign In’ and ‘Sign up’ 2. User fills those fields and click ‘Sign In’ button 3. System checks your user and password 4. When checking finishes, you are lead to homepage |
| Alternative Flows | **Alternative flow 1: Account does not exist in database**   1. After step 3, system displays ‘Input invalid’ if username is not found in the database 2. User should Click ‘Sign Up’ button   **Alternative flow 2: User input wrong username or password**   1. After step 3, the system displays a notification ”Input invalid”. You have to retype these and click ‘Login’ button again   **Alternative flow 3: User cannot login due to forget account**   1. User should create new account or contact to admin. |
| Pre-conditions | User goes to homepage at **[LINK]**  and user registered users account |
| Post-conditions | The user successfully login website |

## Use-case: Sign up

| Use case Name | Sign up |
| --- | --- |
| Brief description | This use case describes how the user registers an account |
| Actors | Guest (non-user) |
| Basic Flow | At the ‘Sign up’ page,   1. Website displays fields: ‘Username’, ‘Password’. Below these are the ‘Sign up’ button. 2. User fills all these informations, then click ‘Sign up’ 3. The system checks the database whether such an account name exists before and the system adds you to the system and you have a new account 4. When completely, you are lead to homepage again. |
| Alternative Flows | **Alternative flow 1: username exists**   1. From #3 of the basic flow, system denies and you have to retype another username 2. Continue step #3 in the basic flow |
| Pre-conditions | User goes to homepage at **[LINK]**  and user don’t have any account |
| Post-conditions | User becomes a member with all the services that the site provides. |

## Use-case: Listen music

| Use case Name | Listen music |
| --- | --- |
| Brief description | This use case describes how can play a song |
| Actors | All users |
| Basic Flow | 1. At the ‘music player’ page, the user clicks on a song which you want to listen to. 2. The song is played right now |
| Alternative Flows |  |
| Pre-conditions | User wants to listen to music |
| Post-conditions | Users hear the song they want to enjoy |

## Use-case: Music control

| Use case Name | Music control |
| --- | --- |
| Brief description | This use case describes how users can go previous/next song, play/pause, loop |
| Actors | All users |
| Basic Flow | 1. At the ‘music player’ page, under music player every song always has control area includes previous, play/pause, next, and loop button 2. User clicks on that. 3. System handle event and execute it. |
| Alternative Flows |  |
| Pre-conditions | User want to control music |
| Post-conditions | Event is executed |

## Use-case: Search music

| Use case Name | Search music |
| --- | --- |
| Brief description | This use case describes how can search a song |
| Actors | All users |
| Basic Flow | 1. In the ‘music player’ page, at the top page, there’s always a search bar. User clicks on that to enable that. 2. User enters a music’s name or artist name 3. System check your music name or artist name in database 4. System shows the song you are looking for |
| Alternative Flows | **Alternative flow 1: The song’s not found**   1. From #2, after entering a search, the system displays nothing 2. Users should probably listen to another song. |
| Pre-conditions | Users are in website and they want to search a song |
| Post-conditions | Users find the song that they want |

## Use-case: Create todolist

| Use case Name | Create todolist |
| --- | --- |
| Brief description | This use case describes how user create a to-do-list, combine work right on site |
| Actors | Users except guest |
| Basic Flow | 1. In homepage, there’s a ‘To do list’ button in navigation bar, user clicks on that button to go to To do list area 2. Website displays 2 fields: ‘Title’ and ‘Content’. Below that have ‘+’ button 3. User fills all these informations, then click ‘+’ button 4. Click on ‘Save’ button to save that list 5. The system checks the database whether such an todolist name exists before. 6. System notices 'Successful’ to user |
| Alternative Flows | **Alternative flow 1: No information, but activate ‘+’ button**   1. Nothing happens |
| Pre-conditions | Users who have an account, are in website and they want to note a to-do-list |
| Post-conditions | User create to-do-list successfully and save it to use next time |

## Use-case: Read to-do-list

| Use case Name | Read to-do-list |
| --- | --- |
| Brief description | This use case describes how user revise a previously saved to-do-list |
| Actors | All users exception non-users |
| Basic Flow | 1. In homepage, there’s a ‘To do list’ button in navigation bar, user clicks on that button to go to To do list area 2. A list shows on display, user can check it |
| Alternative Flows |  |
| Pre-conditions | Users who have an account, are in website and they want to revise a to-do-list |
| Post-conditions | Users can re-read the to-do-list they previously saved |

## Use-case: Delete to-do-list

| Use case Name | Delete to-do-list |
| --- | --- |
| Brief description | This use case describes how user delete a to-do-list |
| Actors | All users except guest |
| Basic Flow | 1. In ‘Todolist’ page, all saved todolists are showed, there is a line “click to remove this task” 2. Users click on that line. 3. System sends a request to database, removes it in database 4. The system updates the list immediately, shows lists to the users. |
| Alternative Flows |  |
| Pre-conditions | Users who have an account, are in website and they want to delete a to-do-list |
| Post-conditions | The system will notify that the deletion is successful and the deleted todo list will be lost |

## Use-case: comment

| Use case Name | Comment |
| --- | --- |
| Brief description | This use case describes how user comment a song |
| Actors | All user except guest |
| Basic Flow | 1. In ‘music player’ page, user choose a song which they want to comment (in this step, that song also plays in same time) 2. The screen displays a box to view comments, there is an input box to enter comments. 3. Users type their reaction or feelings in that bar and press Enter. 4. System sends it and saves on the database and that comment is displayed in that song. |
| Alternative Flows |  |
| Pre-conditions | Users who have an account, are in website and they want to comment a song |
| Post-conditions | Users comment successfully |

## Use-case: chatbox

| Use case Name | chatbox |
| --- | --- |
| Brief description | This use case describes how users back-and-forth texting |
| Actors | All user |
| Basic Flow | 1. In ‘music’ page, there is a “Message” button at the bottom right corner, User chooses it. 2. Write massage, then press Enter 3. Message sends to database server, saves it 4. System displays it in chat box with username or ‘null’ if user is guest |
| Alternative Flows |  |
| Pre-conditions | Users are in website and they want to chat with others |
| Post-conditions | Users chat successfully |

## Use-case: add music

| Use case Name | Add music |
| --- | --- |
| Brief description | This use case describes how user add a link song |
| Actors | Musician and administrator |
| Basic Flow | 1. In ‘manage’ page, there is a form for user adding music 2. Users fill it: title, author, song’s link, lyrics. Click Add Music 3. System sends all that information, saves it in database, when user request play that song, system will connect the link |
| Alternative Flows | **Alternative flow 1: song already exists**   1. After #5, System display “This song already exists” 2. User should check information again or retype another name. 3. Continue #3 |
| Pre-conditions | Users who have roles such as admin or musician who owns the song , are in website and they want to add that song |
| Post-conditions | Users add the song successfully |

## Use-case: remove music

| Use case Name | Remove music |
| --- | --- |
| Brief description | This use case describes how user remove a song |
| Actors | administrator |
| Basic Flow | 1. In ‘manage’ page, there is a form for adding and removing music 2. Admin enters the song’s name, select ‘Delete music’ option 3. System check in database, and delete the song with that name |
| Alternative Flows | **Alternative flow 1: wrong song’s name**   1. After #3, System doesn’t remove that song and the song still in list 2. User should retype the song's name correctly . 3. Continue #2   **Alternative flow 2: Song not exist in database**   1. After #3, nothing changes |
| Pre-conditions | Users who have roles such as admin are in website and they want to remove that song |
| Post-conditions | Users remove the song successfully |

## Use-case: remove account

| Use case Name | Remove an account |
| --- | --- |
| Brief description | This use case describes how user remove an account |
| Actors | Administrator |
| Basic Flow | 1. In ‘manage’ page, there is a form for adding and removing user (beside manage music form) 2. Admin enters the username, select ‘Delete’ option 3. System check in database, and delete the account with that username |
| Alternative Flows | **Alternative flow 1: wrong song’s name**   1. After #3, System doesn’t remove any account   **Alternative flow 2: username don’t exist**   1. After #3, nothing changes |
| Pre-conditions | User who is ‘Admin’ role, there is an user who violate the code of ethical standards, admin need to remove its account as a form of ban |
| Post-conditions | That user account permanently deleted from the website. |

## Use-case: add account

| Use case Name | add account |
| --- | --- |
| Brief description | This use case describes how admin add an account and set role for a user |
| Actors | Administrator |
| Basic Flow | 1. In ‘manage’ page, there is a form for adding and removing user (beside manage music form) 2. Admin fills all information,it is required 3. System sends these to database, and saved it 4. user can log in with that username and password |
| Alternative Flows | **Alternative flow 1: username already exist**   1. After #3, New user is not added |
| Pre-conditions | User who is ‘Admin’ role, there is an user who want sign in, especially is another admin or new musician |
| Post-conditions | User becomes a member with all the services that the site provides. |