# Use cases

## Use Case 1: Create a Playlist

|  |  |
| --- | --- |
|  |  |
| description | This use case describes the steps a user takes to create a new playlist. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running. 2. The user is on the section where playlists can be managed. |
| Main case | 1. The user selects the option to create a new playlist. 2. The system prompts the user to enter a name for the playlist. 3. The user enters a playlist name. 4. The system validates the name to ensure it is not empty and does not exceed character limits. 5. The system calls CreatePlaylist(name: ‘playlist name’) on the User class. 6. The User class creates a new Playlist object with the name ‘playlist name’. 7. The new playlist is added to the user's list of playlists. 8. The system confirms that the playlist has been created and displays the new playlist in the user's playlist list. |
| Postconditions | A new playlist is created in the user's account and is visible in the list of playlists. |
| Exception path | If the user enters an empty name or a name that exceeds the character limit, the system displays an error message indicating the issue and prompts the user to enter a valid name. |

## Use Case 2: Add a Song to a Playlist

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to add a song to an existing playlist. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running. 2. The user has an existing playlist to which they want to add a song. |
| Main case | 1. The user navigates to their list of playlists. 2. The user selects a playlist. 3. The user selects the option to add a song to the playlist. 4. The system prompts the user to search for a song. 5. The user enters a song name. 6. The system displays a list of results matching the name. 7. The user selects a song they want to add to the playlist. 8. The system calls AddSongToPlaylist(playlistName: ‘song name, song: …) on the User class. 9. The User class finds the playlist and calls AddSong(song: ‘song name’) on the Playlist class. 10. The Playlist class adds the song to the list of songs in the playlist. 11. The system confirms that the song has been added and updates the playlist to show the new song. |
| Postconditions | The song is added to the playlist and is visible in the list of songs within the playlist. |
| Exception path | 1. If the user searches for a song and no results are found, the system displays a message indicating that no matches were found and prompts the user to try another search term. |

## Use Case 3: Remove a Song from a Playlist

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to remove a song from an existing playlist. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running. 2. The user has an existing playlist to which they want to remove a song. |
| Main case | 1. The user navigates to their list of playlists. 2. The user selects a playlist. 3. The user selects the option to remove a song from the playlist. 4. The system calls RemoveSong(Song song) on the playlist class.   The Playlist class removes the song from the list of songs in the playlist.   1. The system confirms that the song has been removed and updates the playlist to show the new playlist. |
| Postconditions | The song is removed from the playlist and is visible in the list of songs within the playlist. |
| Exception path |  |

## 

## Use Case 4: Play a Playlist

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to play a playlist within the music streaming application. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running. 2. The user has at least one playlist with songs in it. |
| Main case | 1. The user navigates to their playlists. 2. The user selects a playlist. 3. The system displays the list of songs in the playlist. 4. The user selects the option to play the playlist. 5. The system calls PlayPlaylist(playlistName: ‘playlist name’) on the User class. 6. The User class retrieves the playlist and calls Play() on the Playlist class. 7. The system begins playing the first song in the playlist. 8. The user listens to the playlist, and the system continues to play the subsequent songs in the playlist order. |
| Postconditions | The playlist is playing, starting with the first song and continuing through the list. |
| Exception path | 1. If the playlist is contains no songs, the system displays an error message indicating that the playlist has no songs to play. |

## 

## Use Case 5: View Friends

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to view their list of friends. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running. 2. The user has at least one friend added to their friends list. |
| Main case | 1. The user navigates to the friends section of the application. 2. The system calls ViewFriends() on the User class. 3. The User class retrieves the list of friends from the user's account. 4. The system displays the list of friends to the user, showing each friend's name. 5. The user views the list of friends and can select a friend to view more details or interact further. |
| Postconditions | The user sees an updated list of their friends, displayed in the friends section of the application. |
| Exception path | 1. If the user has no friends added to their list, the system displays a message indicating that no friends have been added. |

## Use Case 6: Add a Friend

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to add a new friend to their friends list. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running. 2. The user knows the username or other identifying information of the friend they want to add. |
| Main case | 1. The user navigates to the friends section of the application. 2. The user selects the option to add a friend. 3. The system prompts the user to enter the friend's username. 4. The user enters the friend's username. 5. The system searches for the user. 6. The system finds a match and displays the friend's profile summary to the user. 7. The user confirms they want to add the user as a friend. 8. The system calls AddFriend(friend: ‘name’) on the User class. 9. The User class updates the user's friends list. 10. The system confirms that the friend has been added to the friends list and displays the updated friends list to the user. |
| Postconditions | The user is added to the user's (mainUser’s) friends list and is visible in the friends section of the application. |
| Exception path | 1. If the user enters an invalid or non-existent username, the system displays a message indicating that the user was not found and prompts the user to enter a valid username. 2. If the user tries to add a friend who is already in their friends list, the system displays a message indicating that the user is already a friend. |

## Use Case 7: Remove a Friend

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to Remove a friend from their friends list. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running. 2. The user knows the username or other identifying information of the friend they want to remove. |
| Main case | 1. The user navigates to the friends section of the application. 2. The user selects the option to remove a friend. 3. The system prompts the user to enter the friend's username. 4. The user enters the friend's username. 5. The system searches for the user. 6. The user confirms they want to remove the user as a friend. 7. The system calls RemoveFriend(friend: ‘name’) on the User class. 8. The User class updates the user's friends list. 9. The system confirms that the friend has been removed from the friends list and displays the updated friends list to the user. |
| Postconditions | The user is removed from the user's (mainUser’s) friends. |
| Exception path | 1. If the user enters an invalid or non-existent username, the system displays a message indicating that the user was not found and prompts the user to enter a valid username. 2. If the user tries to remove a friend who is not in their friends list, the system displays a message indicating that the user is not a friend. |

## 

## Use Case 8: View Friend’s Playlists

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to view the playlists of a friend. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running 2. The user has at least one friend who has public playlists. |
| Main case | 1. The user navigates to their friends list. 2. The user selects a friend from the list. 3. The system calls ViewFriendPlaylists(friend: ‘friend name’) on the User class. 4. The User class retrieves the list of public playlists from the friend's account. 5. The system displays the list of playlists created by the friend to the user. 6. The user views the list of playlists. |
| Postconditions | The user views the list of playlists created by their friend. |
| Exception path | 1. If the friend has no public playlists, the system displays a message indicating that the friend has no public playlists available. |

## Use Case 9: View Albums

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to view the albums available. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running |
| Main case | 1. The user navigates to the albums section of the application. 2. The system calls ViewAlbums() on the Client class. 3. The Client class retrieves a list of available albums from the system’s database. 4. The system displays the list of albums to the user, including details such as album title, artist. 5. The user browses through the list of albums. |
| Postconditions | The user views a list of albums available on the platform. |
| Exception path | 1. If there are no albums available in the system, the system displays a message indicating that no albums are available at the moment. |

## Use Case 10: View Artists

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to view the list of artists available. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running |
| Main case | 1. The user navigates to the artists section of the application. 2. The system calls ViewArtists() on the Client class. 3. The Client class retrieves a list of available artists from the system’s database. 4. The system displays the list of artists to the user, including details such as artist name. 5. The user browses through the list of artists. |
| Postconditions | The user views a list of artists available on the platform. |
| Exception path | 1. If there are no artists available in the system, the system displays a message indicating that no artists are available at the moment. |

## Use Case 11: Show Queue

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to view the queue of the user. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running |
| Main case | 1. The user navigates to the queue menu of the application. 2. The user navigates to the show queue menu of the application. 3. The system displays the list of songs in the queue. |
| Postconditions | The user is presented with the current list of songs in the queue. |
| Exception path | 1. If there are no songs in the queue the system displays a message that no songs are in the queue at the moment. |

## Use Case 12: Remove song from Queue

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to remove a song from queue. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running |
| Main case | 1. The user navigates to the queue menu of the application. 2. The user navigates to the remove song from queue menu of the application. 3. The system displays the list of songs in the queue. 4. The user select the song he want to remove. |
| Postconditions | The song is removed from the queue. |
| Exception path | 1. If there are no songs in the queue the system displays a message to enter, because they are no songs in the queue. |

## Use Case 13: Play Queue

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to play the queue. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running |
| Main case | 1. The user navigates to the queue menu of the application. 2. The user navigates to the play queue menu of the application. 3. The system displays the list of songs in the queue. 4. The user can start the queue. |
| Postconditions | The user start playing this queue |
| Exception path | 1. If there are no songs in the queue, the system displays a message that the queue is empty. |

## Use Case 14: Add playlist to Queue

|  |  |
| --- | --- |
|  |  |
| Description | This use case describes the steps a user takes to add a playlist to the queue. |
| Actor | Client (mainUser) |
| Version | 1.0 |
| Preconditions | 1. The user is has the application running |
| Main case | 1. The user navigates to the queue menu of the application. 2. The user navigates to the add playlist to queue menu of the application. 3. The system displays the list of playlists. 4. The user selects a playlist he want to add. |
| Postconditions | The playlist is added tot he queue |
| Exception path | 1. If there are no playlists available. The system displays a message that there aree no available playlists. |