<Music streaming website>

Use-Case Specification

Version <2.0>

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 10/11/2024 | 1.0 | Use case model and use case specification | Võ Thị Hồng Minh, Đỗ Lương Nhật Minh, Nguyễn Gia Nguyễn |
| 24/11/2024 | 2.0 | Update diagram follow inheritance | Võ Thị Hồng Minh |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Use-case Model 4**](#_heading=h.gjdgxs)

[**2. Use-case Specifications 4**](#_heading=h.30j0zll)

[2.1 Definition: 4](#_heading=h.2ov5gy4xzda)

[2.2 Use-case: Song recommendation 5](#_heading=h.1fob9te)

[2.3 Use-case: Limited Streaming 5](#_heading=h.2et92p0)

[2.4 Use-case: Log In 5](#_heading=h.h81ux5y48m5d)

[2.5 Use-case: Sign Up 6](#_heading=h.tvynsgvx97ob)

[2.6 Use-case: Search song 7](#_heading=h.lk7ru6xg4afd)

[2.7 Use-case: Filter song 7](#_heading=h.k2glt1ftlf7p)

[2.8 Use-case: Edit account info 8](#_heading=h.oscqrum6t7ok)

[2.9 Use-case: Unlimited Streaming 8](#_heading=h.7od6j9yuw4il)

[2.10 Use-case: Create a playlist 9](#_heading=h.4kboytvvwbsz)

[2.11 Use-case: Delete a playlist: 10](#_heading=h.vs8zpm8y2cbw)

[2.12 Use-case: Follow artist-Receive Notification 10](#_heading=h.dcdir11nnat3)

[2.13 Use-case: Social Sharing 11](#_heading=h.klugehjsuh8)

[2.14 Use-case: Personal recommendation 11](#_heading=h.pxck61fm9see)

[2.15 Use case: Offline download 12](#_heading=h.b14yzf30bdfc)

[2.16 Use case: Verified badges 12](#_heading=h.x38h6zrivuwy)

[2.17 Use case: Upload music. 13](#_heading=h.i38ogqlvui07)

[2.18 View analytics 14](#_heading=h.lceayalf2nz5)

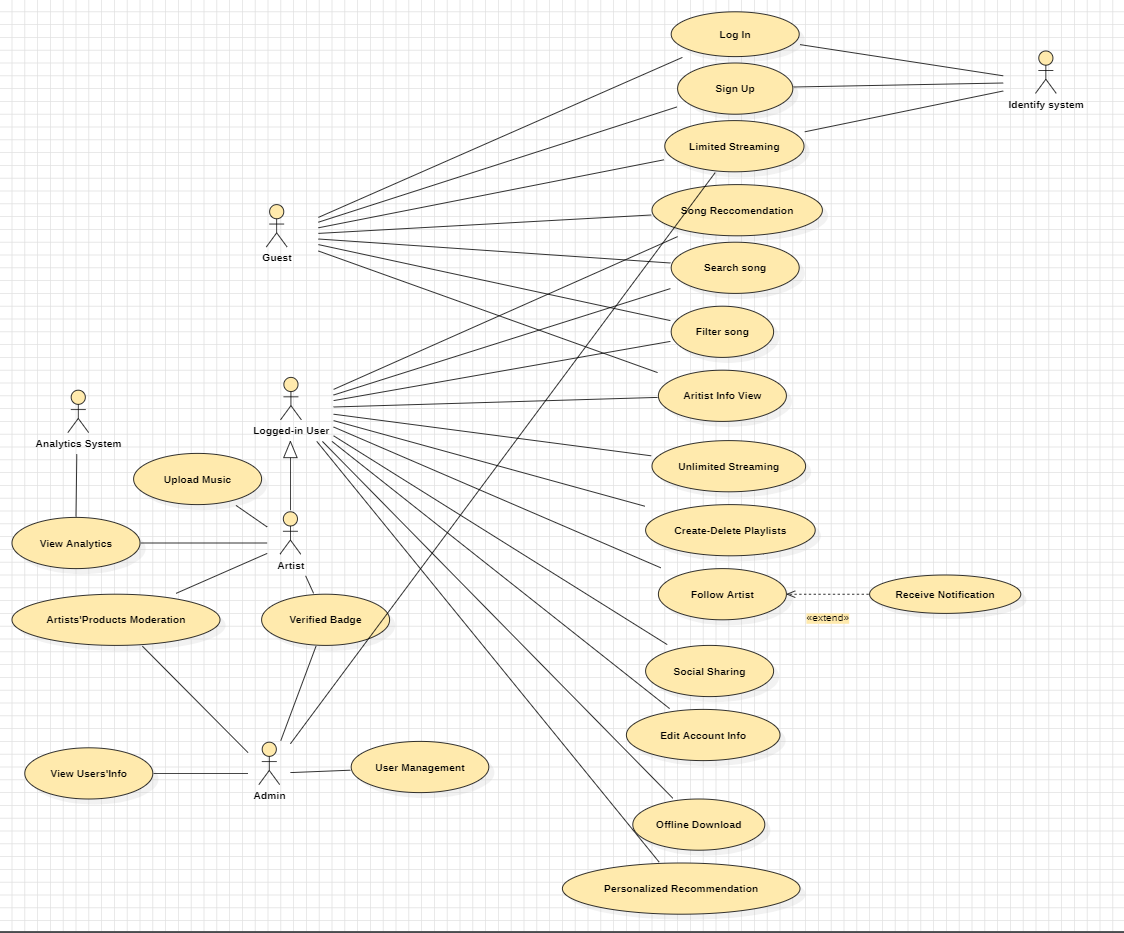
[2.19 Use case: View user info, log 15](#_heading=h.c92hlml6os2b)

[2.20 Use case: User Management 16](#_heading=h.jqxw8qawpp5p)

[2.21 Use case: Review and Approve/Deny Artist Badge Request 17](#_heading=h.ahqggpdegyi9)

[2.22 Use case: Artist's Products Moderation 18](#_heading=h.p0ll7uxcqmr3)

# Use-case Model

**

# Use-case Specifications

## Definition:

We’ll have 6 main actors here:

* Guest: Users who are not logged in.
* Logged-in user: Users who are logged in.
* Artist: Creates music content and uploads it to the system.
* Admin: Manages and maintains the system, ensuring it operates smoothly and securely.
* Analytics System: An external system that collects and analyzes user data.
* Identity System: A system that authenticates the user's identity (e.g., Google, SMS, etc.).

***=> User:*** Refers to all individuals interacting with the system, including guests (non-logged-in users), logged-in users, and artists.

=> ***Artists*** are also logged-in users, but our team will separate them into two groups for easier differentiation when discussing functionalities.

## Use-case: Song recommendation

| Use case Name | Song recommendation |
| --- | --- |
| Brief description | This use case describes how the system displays trending songs and albums to users. |
| Actors | User (Guest, Artist, Logged-in user) |
| Basic Flow | 1. On the homepage, our web will display a curated list of popular songs and albums based on current trends and chart rankings. 2. User clicks on a song or album to preview it or view details. 3. The system allows a short preview of the selected song or album and displays more information. |
| Alternative Flows | **None** |
| Pre-conditions | User goes to the homepage by the link. |
| Post-conditions | The visitor views a selection of popular music, can preview songs, and explore trending albums directly on the homepage. |

## Use-case: Limited Streaming

| Use case Name | Limited Streaming |
| --- | --- |
| Brief description | This use case describes how the system enforces streaming limitations for unregistered users. |
| Actors | Guest, Admin (modify number of song/albums user can listen to), Identify System |
| Basic Flow | 1. The User opens a song or playlist on the website. 2. The Identify System checks the User's status (logged in, subscription level). 3. If User is unregistered, our system allows streaming of a limited number of songs. 4. The system plays a preview or a short clip of the song. 5. The system displays a prompt encouraging the User to sign up or upgrade to a premium account to access full songs. |
| Alternative Flows | From #3 of the basic flow, if the User has been logged-in, the system allows unlimited streaming of songs and albums. |
| Pre-conditions | The User opened the streaming website or app and selected a song or playlist. |
| Post-conditions | The User listens to a limited amount of songs and after reaching the limit, our web will suggest the user to upgrade for full streaming access. |

## 2.4 Use-case: Log In

| Use case Name | Log In |
| --- | --- |
| Brief description | This use case describes how the system enforces streaming limitations for unregistered users. |
| Actors | Guest, Identify system |
| Basic Flow | 1. At the homepage, the user clicks on the 'Log In' button. 2. The system displays the login form. 3. The User enters their login credentials (e.g., email and password) or selects a third-party authentication method (e.g., Google, SMS). 4. The system (Identify System) verifies the credentials provided by the User. 5. If the credentials are valid, the system grants the User access and loads their personalized profile and playlists. 6. The system redirects the user to their personalized homepage. |
| Alternative Flows | **Alternative flow #1:** Incorrect username or password   * From #4 of the basic flow, the system detects that the username or password is incorrect. * The system displays an error message to the user. * The user re-enters their username and password. * Continue step #3 in the basic flow.   **Alternative flow #2:** Users forgets password   * From #2 of the basic flow, the user clicks on the 'Forgot Password' link. * The system displays the password recovery form. * The user enters their email address. * The system sends a password recovery email to the user. * The user follows the instructions in the email to reset their password. * The user returns to the login form. * Continue step #3 in the basic flow. |
| Pre-conditions | Users open the website and click on the “Log In” button.  Users must already register an account on our web. |
| Post-conditions | The User is successfully logged into their account and can access personalized content, playlists, and other user-specific features. If unsuccessful, the User is provided with options to retry or recover their account. |

## 2.5 Use-case: Sign Up

| Use case Name | Sign Up |
| --- | --- |
| Brief description | This use case describes how a new user registers for an account on our web. |
| Actors | User, Identify system |
| Basic Flow | 1. At the homepage, the new user clicks on the 'Sign Up' button. 2. The User enters required information, such as email, password, and possibly other details like name and date of birth. 3. The User may also choose a third-party authentication option, such as Google or SMS verification. 4. The system (Identify System) validates the information provided and checks for any existing account with the same email or phone number. 5. If the information is valid, the system creates a new account and confirms the registration. 6. The system displays a success message and logs the User in automatically or redirects them to the login page. |
| Alternative Flows | **Alternative flow 1:** Username or email already exists   * From #4 of the basic flow, the system detects that the username or email already exists. * The system displays an error message to the user. * The user enters a different username or email. * Continue step #2 in the basic flow.   **Alternative flow 2:** Invalid email format   * From #4 of the basic flow, the system detects that the email format is invalid. * The system displays an error message to the user. * The user enters a valid email address. * Continue step #2 in the basic flow. |
| Pre-conditions | The new user is on the homepage of our website.  User haven’t signed up to our website before. |
| Post-conditions | The User successfully creates a new account and can now log in to access personalized features. |

## 2.6 Use-case: Search song

| Use case Name | Search Song |
| --- | --- |
| Brief description | This use case describes how the user can search for a song on the music streaming website. |
| Actors | User |
| Basic Flow | 1. At the homepage, the user enters keywords in the ‘Search’ field. 2. The user clicks on the ‘Search’ button to start searching for a song. 3. The system displays the songs found based on the entered keywords. 4. The user clicks on a song from the search results to play it. 5. The system plays the selected song. |
| Alternative Flows | **Alternative flow 1:** No songs found   * From #3 of the basic flow, the system does not find any songs matching the entered keywords. * The system displays a message indicating that no songs were found. * The user enters different keywords in the ‘Search’ field. * Continue step #2 in the basic flow.   **Alternative flow 2:** User refines search   * From #3 of the basic flow, the user is not satisfied with the search results. * The user enters additional or different keywords in the ‘Search’ field. * Continue step #2 in the basic flow. |
| Pre-conditions | The user is on the homepage of the website at |
| Post-conditions | The user successfully finds and plays a song based on the search keywords. |

## 2.7 Use-case: Filter song

| Use case Name | Filter Song |
| --- | --- |
| Brief description | This use case describes how a user filters songs based on specific criteria such as genre, artist, album, or release date to narrow down search results. |
| Actors | User |
| Basic Flow | 1. The User navigates to the "Filter" section after performing a search or browsing songs. 2. The User selects the filter criteria they want to apply (e.g., genre, artist, album, release date, or popularity). 3. The system applies the selected filters and updates the song list to reflect the chosen criteria. 4. The User views the filtered song results. 5. The User can further interact with the filtered results, such as playing a song or adding it to a playlist. |
| Alternative Flows | **Alternative flow 1:** No songs match the filter criteria   * From #3 of the basic flow, if no songs match the applied filters, the system displays a message indicating no results and allows the User to modify the filter criteria. * The User can modify the filter settings and continue from step #2 in the basic flow. |
| Pre-conditions | The user has performed a search or is browsing through the song catalog. |
| Post-conditions | The user sees a list of songs that match the selected filter criteria and can further interact with the results. |

## 2.8 Use-case: Edit account info

| Use case Name | Edit Account Info |
| --- | --- |
| Brief description | This use case lets users update their personal info |
| Actors | Logged-in users, Artists |
| Basic Flow | 1. At the “homepage”, users go to the “account info page”. 2. Users click “Update info”. 3. Info that can be update will be shown in the boxes 4. Users update their info in given boxes. 5. Users click “save” to save the info they typed in. |
| Alternative Flows | None |
| Pre-conditions | User goes to the “homepage” |
| Post-conditions | The users update their info successfully and they will be shown at the “account info page” |

## 2.9 Use-case: Unlimited Streaming

| Use case Name | Unlimited Streaming |
| --- | --- |
| Brief description | This use case describes how users can access and stream an unlimited selection of songs, albums, and playlists. |
| Actors | Logged-in user, Artist |
| Basic Flow | 1. Users log into their account to the streaming section. 2. The system displays a library of available songs, albums, and playlists. 3. User selects a song, album, or playlist to stream. 4. The system plays the selected item in the media player with no interruptions. |
| Alternative Flows | None |
| Pre-conditions | Users must be logged in to access unlimited streaming. |
| Post-conditions | Users can enjoy uninterrupted streaming |

## 2.10 Use-case: Create a playlist

| Use case Name | Create a playlist |
| --- | --- |
| Brief description | This use case will describe how user create a playlist |
| Actors | Logged-in users, Artist |
| Basic Flow | 1. At the “homepage”, click at the “create a playlist” 2. A “create new playlist” box will pop-up 3. Users type playlist’s name to that box 4. Users click to “create” 5. New playlist will be shown at homepage |
| Alternative Flows | **Alternative flow #1:**   * From the “homepage”, users click to any song * The song’s page user click to will be shown * Users choose to “add to playlist” icon * A box will pop-up and show choice(s) and the “create a playlist” choice will be shown at the top * Users click to “create a playlist”. * Follow steps #2 to #5 of Basic flow   **Alternative flow #2:**   * From the “homepage”, users click to “user’s playlists” page. * Users will be changed to the “user’s playlists” page. * User click to the “create a playlist” * Follow steps #2 to #5 |
| Pre-conditions | User goes to “homepage” |
| Post-conditions | The users created their playlists successfully and they will be shown at “homepage” and “user’s playlists” |

## 2.11 Use-case: Delete a playlist:

| Use case Name | Delete a playlist |
| --- | --- |
| Brief description | This use case will describe how user delete a playlist |
| Actors | Logged-in users, artists |
| Basic Flow | 1. At the “homepage”, users click to “user’s playlists” 2. Users choose the playlist they want to delete 3. The playlist page will pop-up 4. User click to “delete playlist” 5. A box will pop up and tell users to confirm to delete the playlist 6. Users click “confirm” to delete the playlist 7. System will show a notification that tell “Playlist … had been deleted” |
| Alternative Flows | None |
| Pre-conditions | User goes to “homepage” |
| Post-conditions | The playlist user want to delete is deleted and they will be disappeared |

## 2.12 Use-case: Follow artist-Receive Notification

| Use case Name | Follow artist —Receive Notification |
| --- | --- |
| Brief description | This use case describe how users follow their favorite artist and receive notification from them |
| Actors | Logged-in users, Artists |
| Basic Flow | 1. At the “homepage”, users click to the search bar. 2. Users type the artist’s name they want to find in the bar. 3. The searching result will pop up. 4. Users click to the search result they want. 5. Users will be directed to the page of the artist they chose. 6. Users will see a heart icon at the artist profile page. 7. Users click the heart icon to follow the artist. 8. After clicking the icon, the system notification will pop-up and tell the users have followed the artist. |
| Alternative Flows | **Alternative flow #1: User can’t find the artist on search bar**   1. At the “homepage”, users search for a song of the artist they want to follow on the search bar 2. Click on the song. 3. Users will be directed to the song’s page. 4. Users will find artist information at the credit of the song. 5. Users click to the artist profile. 6. Follow steps #5 to #8 of the basic flow |
| Pre-conditions | Users go to the “homepage” |
| Post-conditions | Users followed their favorite artists successfully and users will receive every notification when their followed artist updates any news. |

## 2.13 Use-case: Social Sharing

| Use case Name | Social Sharing |
| --- | --- |
| Brief description | This use case describes how logged-in users can share their favorite song or artist with others by generating a shareable link. |
| Actors | Logged-in users |
| Basic Flow | 1. Users navigate to the “homepage.” 2. Users search for the desired song or artist using the search bar. 3. Users click on the song or artist profile in the search results. 4. The system directs the user to the song or artist's profile page. 5. Users locate the share icon on the profile page. 6. Users click the share icon. 7. The system displays a pop-up box with a link to the song or artist's profile. 8. Users click the copy button next to the link to copy it. 9. Users paste the link to share it with others. |
| Alternative Flows | no alternative flow |
| Pre-conditions | Users go to “homepage” |
| Post-conditions | Users got the link that direct to the song/artist profile they want to share |

## 2.14 Use-case: Personal recommendation

| Use case Name | Personal Recommendation |
| --- | --- |
| Brief description | This use case describes how the system provides personalized song, album, and playlist recommendations based on the user’s listening history and preferences. |
| Actors | Logged-in User |
| Basic Flow | 1. Users log into their account. 2. The system analyzes the user’s listening history, liked songs, saved albums, and playlists. 3. Based on this data, the system generates a personalized list of recommended songs, albums, and playlists. 4. Users view the personalized recommendations on the homepage or within a dedicated "For You" section. 5. Users can click on any recommended song, album, or playlist to start streaming. |
| Alternative Flows | **For New Users:** If there is limited listening history, the system recommends popular songs and trending playlists until sufficient data is available for personalized recommendations. |
| Pre-conditions | Users must be logged in, and the system must have access to their listening history or preferences. |
| Post-conditions | Users receive personalized recommendations and can discover new music that aligns with their tastes, enhancing their overall listening experience. |

## 2.15 Use case: Offline download

| Use case Name | Offline Download |
| --- | --- |
| Brief description | This use case describes how a logged-in user downloads a song to their device for offline listening. |
| Actors | Logged-in users |
| Basic Flow | 1. User navigates to the “homepage.” 2. User searches for the desired song using the search bar. 3. User clicks on the song in the search results. 4. The system directs the user to the song's profile page. 5. User locates the download icon next to the play button. 6. User clicks the download icon. 7. The system downloads the song to the user’s device for offline access. |
| Alternative Flows | no alternatives flow |
| Pre-conditions | User is logged in and has navigated to the “homepage.” |
| Post-conditions | The selected song is successfully downloaded and saved to the user’s device, available for offline listening. |

## 2.16 Use case: Verified badges

| Use case Name | Verified Badge for Artist Account |
| --- | --- |
| Brief description | This use case describes how an artist account is verified and awarded a verified badge to signify its authenticity and credibility to users. |
| Actors | Artist, Admin |
| Basic Flow | 1. Artist navigates to the "Account Settings" page. 2. Artist locates and clicks the "Request Verification" button. 3. The system displays a verification form requiring the artist to provide identification documents and additional credentials for verification. 4. Artist completes and submits the form. 5. System sends a notification to the system administrator with the artist's verification request details. 6. System administrator reviews the documents and credentials submitted by the artist. 7. If the information is verified as authentic, the system administrator approves the request. 8. System updates the artist's profile to display a verified badge next to their name. 9. System notifies the artist that their account has been successfully verified. |
| Alternative Flows | **Alternative Flow #1:** Invalid Documents Provided   * If the artist submits incomplete or invalid documentation, the system notifies the artist of the issue and prompts them to resubmit the required documents. |
| Pre-conditions | The artist is logged in and has an artist account type.  The artist has navigated to the "Account Settings" page. |
| Post-conditions | If verified, the artist’s account displays a verified badge, indicating the authenticity of the account to users. |

## 2.17 Use case: Upload music.

| Use case Name | Upload Music |
| --- | --- |
| Brief description | This use case describes how an artist uploads a new song to their profile for users to listen to and download. |
| Actors | Artist |
| Basic Flow | 1. Artist navigates to the “Profile” or “Dashboard” page. 2. Artist clicks on the “Upload New Song” button. 3. System displays an upload form that includes fields for song title, genre, album (if applicable), description, cover art, and audio file. 4. Artist completes the form, selecting the audio file from their device and filling out other relevant fields. 5. Artist clicks the “Upload” button to submit the song. 6. Our web validates the audio file format, size, and other input fields. 7. If validation succeeds, the system uploads the song, creates a song profile page, and links it to the artist's profile. 8. Our web confirms the upload was successful by displaying a success message to the artist and an option to view or edit the song profile. |
| Alternative Flows | **Alternative Flow #1:** **File Format or Size Invalid:**   * If the selected audio file is not in a supported format or exceeds the maximum size limit, the system displays an error message and prompts the artist to upload a valid file. |
| Pre-conditions | The artist is logged in and has an artist account type.  The artist has navigated to the "Account Settings" page. |
| Post-conditions | If verified, the artist’s account displays a verified badge, indicating the authenticity of the account to users. |

## 2.18 View analytics

| Use case Name | View Analytics |
| --- | --- |
| Brief description | This use case describes how an artist can view detailed analytics on their uploaded songs, including plays, downloads, likes, etc.. |
| Actors | Artist, Analytics System |
| Basic Flow | 1. Artist navigates to the “Dashboard” or “Profile” page. 2. Artist clicks on the “View Analytics” option. 3. System displays an analytics dashboard, which includes key metrics like total plays, downloads, likes, and shares for each uploaded song. 4. Artist selects a specific song to view more detailed analytics. 5. System displays a breakdown of analytics for the selected song, including:    1. Daily, weekly, and monthly plays    2. Age and gender demographics of listeners 6. Artists review the analytics and can navigate back to view analytics for other songs. |
| Alternative Flows | **No Analytics Data Available**:   * If no data is available (e.g., the song is newly uploaded), the system displays a message stating that analytics will be available once more user interaction has occurred. |
| Pre-conditions | The artist is logged in and has at least one song uploaded.  The artist has navigated to the “Dashboard” or “Profile” page. |
| Post-conditions | The artist successfully views analytics data, gaining insights into song performance and audience engagement. |

## 2.19 Use case: View user info, log

| Use case Name | View User Info and Activity Log |
| --- | --- |
| Brief description | This use case describes how a system administrator can view detailed information about a user and access their activity log to monitor account actions and ensure compliance with platform policies. |
| Actors | Admin. |
| Basic Flow | 1. System administrator logs into the admin dashboard. 2. System administrator navigates to the “User Management” section.. 3. System administrator searches for a specific user by entering the username, email, or user ID in the search bar. 4. System displays a list of matching users, and the system administrator selects the desired user. 5. System displays the user’s profile, which includes details such as:    1. Account information (username, email, account type, status)    2. Date of account creation and last login    3. Verification status (e.g., verified artist badge) 6. System administrator clicks on “View Activity Log” to access a chronological list of actions taken by the user on the platform, including:    1. Logins and logouts    2. Uploaded content (songs, profile updates)    3. Interactions (likes, comments, downloads)    4. Any reported violations or flagged content 7. System administrator reviews the user’s activity log for any policy violations or suspicious activity. |
| Alternative Flows | **Alternative Flow #1:** No matching user found   * If no matching user is found, the system displays a message indicating that the user does not exist or suggesting that the administrator refine the search. |
| Pre-conditions | The system administrator is logged in with appropriate permissions.  The system administrator has access to the “User Management” section. |
| Post-conditions | The system administrator successfully views user information and activity logs, allowing for monitoring and action if necessary to maintain platform integrity. |

## 2.20 Use case: User Management

| Use case Name | User Management |
| --- | --- |
| Brief description | This use case describes how a system administrator manages user accounts on the platform, including actions like viewing, editing, suspending, and deleting accounts. |
| Actors | Admin |
| Basic Flow | 1. System administrator logs into the admin dashboard. 2. System administrator navigates to the “User Management” section. 3. System displays a list of all user accounts with options to search, filter, and sort by attributes like username, email, account type, status, and last active date. 4. System administrator searches for a specific user by entering keywords or applying filters. 5. System displays matching users, and the administrator selects a user to manage their account. 6. System administrator views user details, including profile information, account status, verification status, and recent activity. 7. System administrator has the following options:    1. **Edit Account Information**: Modify user details like username, email, and account type.    2. **Change Account Status**: Suspend, unsuspend, or reactivate the user account as needed.    3. **Send Warning**: Send a warning message to the user for policy violations or inappropriate behavior. The message will appear in the user’s inbox or as a notification upon login.    4. **Permanently Ban Account**: Issue a permanent ban on the user account, which prevents the user from logging in or accessing the platform. The system marks the account as “Banned” and restricted access.    5. **Revoke Verification Badge**:Remove the verification badge for artist accounts based on eligibility.    6. **Delete Account**: Permanently delete the user’s account and all associated data. 8. System administrator selects the desired action, and the system performs the requested action, updating the account information accordingly. 9. System confirms the success of the action and updates the user list to reflect any changes. |
| Alternative Flows | **Alternative Flow #1:** User Not Found:   * If no matching user is found, the system displays a message indicating that the user does not exist or suggesting that the administrator refine the search. |
| Pre-conditions | The system administrator is logged in with appropriate permissions.  The system administrator has access to the “User Management” section. |
| Post-conditions | The system administrator successfully views user information and activity logs, allowing for monitoring and action if necessary to maintain platform integrity. |

## 2.21 Use case: Review and Approve/Deny Artist Badge Request

| Use case Name | Review and Approve/Deny Artist Badge Request |
| --- | --- |
| Brief description | This use case describes how a system administrator reviews verification badge requests submitted by artists, determines their legitimacy, and either approves, denies, or requests additional information from the artist. |
| Actors | System Administrator, Artist (logged-in). |
| Basic Flow | 1. Artists submit a verification badge request through their “Account Settings” page, including required documents and credentials. 2. System notifies the system administrator of a new verification badge request. 3. System administrator logs into the admin dashboard and navigates to the “Verification Requests” section. 4. System displays a list of pending verification requests, including each artist's name, account details, and uploaded documents. 5. System administrator selects a specific artist’s request to review. 6. System displays the artist’s account information and verification documents for evaluation. 7. System administrator evaluates the request and has the following options:    1. **Approve Verification**:       1. If the documentation is legitimate, the system administrator selects “Approve,” and the system grants the artist a verified badge.       2. System updates the artist’s profile to display the verified badge and notifies the artist of the successful verification.    2. **Request Additional Information**:       1. If documentation is incomplete or insufficient, the administrator selects “Request Additional Information.”       2. System prompts the artist to provide further details or documents and notifies them of the request.    3. **Deny Verification**:       1. If the documentation does not meet verification criteria or appears illegitimate, the administrator selects “Deny.”       2. System notifies the artist that their verification request has been denied, including a reason if applicable. |
| Alternative Flows | **Alternative Flow #1:** Resubmission   * If the artist receives a “Request Additional Information” notification, they can resubmit the verification request with the required documents. * System sends the resubmission notification to the system administrator, who can review it as a new request.   **Alternative Flow #2:** Appeal for Denial   * If the verification request is denied, the artist can submit an appeal with additional justification or documents. The system flags this as an appeal for administrator review. |
| Pre-conditions | * Artist has submitted a badge request with necessary documents. * System administrator is logged in with access to the “Verification Requests” section. |
| Post-conditions | * If approved, the artist’s profile displays a verified badge, indicating account authenticity. * If additional information is requested, the artist is notified to resubmit. * If denied, the artist is informed of the decision with an option to appeal or resubmit. |

## 2.22 Use case: Artist's Products Moderation

| Use case Name | Artist's Products Moderation |
| --- | --- |
| Brief description | This use case describes how a system administrator reviews and moderates content uploaded by artists, including songs, albums, and other media. The administrator can approve, deny, or request modifications to ensure all content adheres to platform guidelines. |
| Actors | Admin (System Administrator), Artist |
| Basic Flow | 1. System administrator logs into the admin dashboard and navigates to the “Content Moderation” section. 2. System displays a list of newly uploaded or flagged content by artists, including songs, albums, or other media that require moderation. 3. System administrator selects a specific content item from an artist to review. 4. System displays content details, including:    1. Title, genre, and description    2. Upload date and artist’s name    3. Associated media files (e.g., audio, cover art)    4. Any user-generated flags or reports associated with the content (if applicable) 5. System administrator evaluates the content and has the following options:    1. **Approve Content**:       1. If the content adheres to platform guidelines, the administrator approves it.       2. System updates the content status to “Approved,” making it publicly available.    2. **Request Modifications**:       1. If the content needs adjustments (e.g., inappropriate cover art or explicit language without proper tagging), the administrator selects “Request Modifications.”       2. System notifies the artist to modify the content according to specific feedback before resubmitting for approval.    3. **Deny/Remove Content**:       1. If the content violates guidelines or is deemed inappropriate, the administrator selects “Deny” (for new uploads) or “Remove” (for existing content).       2. System removes or prevents the content from being visible on the platform and notifies the artist of the denial with reasons if applicable. |
| Alternative Flows | **Alternative Flow #1:** Resubmission After Modification Request:   * If the artist receives a modification request, they can adjust the content and resubmit it. * System notifies the administrator of the resubmission, prompting another review cycle.   **Alternative Flow #2:** User Reports Flagged Content:   * If content is flagged by users for review, the system administrator sees it in a separate “Flagged Content” section, which follows the same review process |
| Pre-conditions | * Artists have uploaded content that is pending review, or existing content has been flagged by users. * System administrator is logged in with access to the “Content Moderation” section. |
| Post-conditions | * Content is approved, modified, or denied based on the platform guidelines maintaining a safe and compliant environment for users. |