Software Requirements Specification

for

Music Streaming System

Version 2.0 approved

Prepared by Nguyễn Đức Khánh

Ngô Lê Hoàng

Đỗ Minh Quang

Group 5, 2324II INT2208E 23, VNU-UET

March 26, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Initial Draft	March 12th 2024	First draft with section details	0.1
Detail Enhancement	March 19th 2024	Subsection details updated Example figure given for each section.	1.0
More Detail Specification	March 26th 2024	Appendix section updated. Use Case specifications template updated.	1.1
Reformat Functional Requirement	April 2nd 2024	Change format for Use Case specification Add new actor for the system and new UC diagram	2.0
Diagram Modified	April 16th 2024	Change some diagrams of document	2.1
Diagram Modified 2	April 23th 2024	Edit overview use case diagram	2.2

1. Introduction

The digital revolution has transformed the way we experience music, bringing the vast universe of songs from around the globe to our fingertips. MusicStreaming stands at the forefront of this transformation, offering a seamless and personalized music streaming service to millions of users. This document delineates the requirements for MusicStreaming's software system, ensuring that it continues to meet the evolving needs of its users with efficiency, reliability, and innovation.

1.1. Purpose

The purpose of this Software Requirements Specification (SRS) is to establish a clear and detailed set of requirements for the MusicStreaming application. It serves as a contractual basis between stakeholders and the development team, ensuring a mutual understanding of functionalities, system capabilities, and performance criteria. This document will guide the design, development, and testing processes, facilitating a systematic approach to achieving project objectives. It also provides a framework for future maintenance and enhancements, ensuring the longevity and adaptability of the application in the dynamic landscape of digital music services.

1.2. Document Conventions

Headings	Arial/16 font size
Sub headings	Arial/12 font size
Body	Arial/11 font size

Numbered lists represent sequences or procedures.
Bulleted lists signify options or unordered sets.

1.3. Intended Audience and Reading Suggestions

This SRS document is intended for a diverse group of stakeholders, including but not limited to software developers, project managers, testers, marketing staff, document writers, and the end-users of the MusicStreaming application. The document is organized into 8 parts. 1. Introduction, 2. Overall Description, 3. External Interface Requirements, 4. System Features, 5. Other Nonfunctional Requirements, 6. Other Requirements and t appendix parts. All the parts are independent however reading the document sequentially helps the reader understand the MusicStreaming system better.

1.4. Project Scope

The scope of this project encompasses the development and enhancement of the MusicStreaming application, a premier music streaming service. The project aims to deliver a user-friendly platform that provides access to a vast library of music, podcasts, and videos from creators all over the world.

The key objectives include:

- Delivering high-quality audio streaming to users globally.
- Providing personalized content recommendations based on user preferences.
- Ensuring seamless integration across various devices and platforms.
- Maintaining a robust and scalable infrastructure to support a growing user base.
- Incorporating social features that allow for sharing and discovering content within user communities.

This SRS will detail the requirements necessary to achieve these objectives, including user interface design, system functionalities, performance requirements, and security measures. The document will also address the constraints, dependencies, and assumptions related to the project.

1.5. References

The following references provide the foundation and guidelines for the preparation of this Software Requirements Specification:

- IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998).
- IEEE Guide to Software Requirements Specifications (IEEE Std 1233, 1998 Edition).
- Spotify Developer Documentation and API Reference.
- Previous version documents and user feedback reports related to the Spotify application.
- Software Engineering course slides provided by lecturers.

As the project progresses, additional resources may be identified and documented in this section.

2. Overall Description

As we delve into the Overall Description, we present a bird's-eye view of the MusicStreaming application, its capabilities, and its interaction with the broader digital ecosystem. This section will pave the way for a granular look at the product's role in the competitive landscape of music streaming services

2.1. Product Perspective

The Music Streaming system is envisioned as a comprehensive digital platform designed to provide users with access to a vast library of musical content. It stands as an independent, full-featured service that allows for the streaming of music tracks, albums, and artist-curated playlists. The system is conceptualized to seamlessly integrate with users' digital ecosystems, offering compatibility across various devices and operating systems.

2.2. Product Functions

- *Music Library Access*: Users can stream music from a vast collection of tracks from various artists, genres, and languages. The service provides access to millions of songs, catering to diverse musical tastes.
- *Personalized Recommendations*: Leveraging data analytics, MusicStreaming offers personalized music recommendations based on individual listening habits, helping users discover new music and curated playlists.
- *Playlist Creation*: Users can create, share, and follow playlists, which can be customized and saved for offline listening.
- *Offline Listening*: Premium users have the option to download music and listen offline, providing flexibility and convenience.
- *High-Quality Audio*: MusicStreaming offers high-quality audio streaming options, ensuring an optimal listening experience for premium subscribers.
- *User Interface*: The application features a user-friendly interface with easy navigation, search functions, and access to different sections like 'Home', 'Search', and 'Your Library.

2.3. User Classes and Characteristics

- *General Users*: This class includes individuals who use MusicStreaming for personal entertainment. They are characterized by their diverse musical tastes and preferences, and they utilize MusicStreaming's free or premium services. General users are the largest group and include various demographics.

- *Premium Users*: These users pay a subscription fee for enhanced features such as ad-free listening, offline playback, and high-quality audio. They value uninterrupted and high-fidelity music experiences.

- *Artists and Content Creators*: This group uses MusicStreaming as a platform to publish and monetize their music and podcasts. They benefit from MusicStreaming's tools for artists, which provide insights into listener demographics and engagement.
- *Administrators:* Administrators oversee the system's operations, manage server infrastructure, security, and updates. Administrators ensure compliance with guidelines, handle reported content; address user inquiries, troubleshoot issues, and maintain a positive user experience.

2.4. Operating Environment

2.4.1. Hardware Platform

MusicStreaming Web is accessible through web browsers, eliminating the need for specific hardware requirements. It supports integration with various commercial hardware for music streaming, such as smart speakers and AV receivers, through MusicStreaming Connect.

2.4.2. Operating Systems and Versions

MusicStreaming is compatible with major desktop and mobile operating systems, ensuring broad accessibility. It supports:

+ Desktop: Windows 7 and above, macOS Yosemite 10.10 and above, and Linux distributions via the web browser.
+ Mobile: Available on Android and iOS devices with web browser support.

2.4.3. Co-existing Software

The web application is compatible with the latest versions of major web browsers such as Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge. It also integrates with various browser extensions and plugins that do not interfere with its functionality.

2.4.4. Infrastructure Requirements

Standard web server and database configurations will suffice for the initial deployment:

Disk capacity	>=64GB
Memory (RAM)	>=2GB
Front-end	HTML, CSS , ReactJS
Back-end	FastAPI, PostgreSQL (latest versions as of March 2024)

2.5. Design and Implementation Constraints

The system runs under Windows 7 / 8 / 8.1 / 10 / 11, macOS 10.10 and above or any version of Linux.

The application is developed on ReactJS platform as frontend and FastAPI as backend.

2.6. User Documentation

- *User Manual:*
  - A comprehensive guide that explains how to use the app.
  - Includes step-by-step instructions for common tasks (e.g., searching for music, creating playlists).
  - Describes navigation, buttons, and icons.
  - Provides troubleshooting tips.
- *FAQs:*
  - Addresses common queries and issues.
  - Covers topics like account management, playback, and settings.
  - Helps users find quick solutions.
- *Tutorials and How-To Guides:*
  - Short videos or written guides demonstrating specific features.
  - Useful for visual learners.
- *Contextual Help:*
  - In-app tooltips or pop-ups that provide information relevant to the current screen.
  - Helps users learn on the go.
- *Release Notes:*
  - Summarizes changes in each app update.
  - Highlights new features, bug fixes, and improvements.
- *Contact Information:*
  - Provides details for customer support or feedback.
  - Includes email addresses, chat support, or community forums.

2.7. Assumptions and Dependencies

2.7.1. Assumptions

- *User Connectivity*: It is assumed that users have access to stable and high-speed internet connectivity to stream music without interruptions.
- *Device Capability*: Users' devices are assumed to be capable of running the MusicStreaming application, which includes having sufficient processing power, memory, and a compatible operating system.
- *Content Licensing*: It is assumed that MusicStreaming will continue to secure and maintain licensing agreements with music labels, artists, and publishers to offer a wide range of content.

2.7.2. Dependencies

- *Third-Party Services*: MusicStreaming's functionality depends on various third-party services, including payment gateways for subscription management, social media platforms for sharing content, and cloud services for hosting the application and data.
- *Hardware Compatibility*: MusicStreaming relies on the compatibility of hardware devices, such as smartphones and smart speakers, to ensure users can access the service across different platforms.
- *Regulatory Compliance*: The service is dependent on adhering to international laws and regulations related to digital services, data protection, and copyright.

3. External Interface Requirements

3.1. User Interfaces

a) **Logical Characteristics of Interface:**

- The web application shall have a responsive design capable of adapting to various screen sizes and resolutions.
- Required screen formats shall include a home page featuring recommended playlists, trending tracks, and user-specific recommendations.
- Page layouts should prioritize easy navigation with clear categories for browsing music genres, artists, albums, and playlists.
- The content of reports or menus should be customizable, allowing users to personalize their music browsing experience.
- Playback controls shall be easily accessible on the interface, including play, pause, skip, repeat, and volume adjustment functionalities.
- The interface should support the creation, management, and sharing of playlists, with options for editing and rearranging track order.

b) **Optimization for User Interaction:**

- The interface shall prioritize ease of use and intuitive navigation, ensuring that users can find and play music effortlessly.
- Do's and Don'ts guidelines shall be provided to optimize user experience, including clear labeling of buttons, intuitive icons, and consistent navigation patterns.
- Error messages shall be concise and informative, offering options for both long and short error messages based on user preferences.
- Users shall have the option to customize their interface preferences, such as choosing between light and dark themes or adjusting font sizes for better readability.
- The system shall provide tooltips or contextual help to guide users through unfamiliar features or functionalities.
- Verifiable requirements shall be established, such as specifying that a user with basic training can create a playlist and add tracks within a certain time frame.

3.2. Hardware Interfaces

- The web application shall support interfaces with various hardware components, including audio output devices such as speakers, headphones, and sound cards.
- It shall support multiple audio output channels for stereo and surround sound configurations.
- Supported devices shall include desktop and laptop computers, smartphones, tablets, and smart TVs with compatible web browsers.
- Compatibility with Bluetooth audio devices shall be provided, allowing users to stream music wirelessly to Bluetooth-enabled speakers and headphones.

3.3. Software Interfaces

- The Music Streaming system's web interface is crafted using modern web development technologies such as HTML, CSS, and JavaScript, ensuring compatibility across various operating systems like Unix, Linux, Mac, and Windows. The system utilizes PostgreSQL databases located in data centers worldwide to store and manage vast amounts of song data, user profiles, and artist information.

- The databases hold extensive details about music tracks, including album art, track names, genres, artists, release dates, and user reviews and ratings. This data is meticulously curated by the content management system to provide users with a rich and engaging experience.

- The system also integrates with a user account management system to oversee user subscriptions, preferences, and settings. This system is responsible for managing user data, including account creation dates, subscription types, and user-generated content like playlists and favorites.

- Music Streaming system communicates with content delivery networks (CDNs) to ensure efficient distribution of music files, enabling fast and reliable streaming services. The CDN system manages the complex task of delivering high-quality audio content to users around the globe, adapting to various network conditions and user demands.

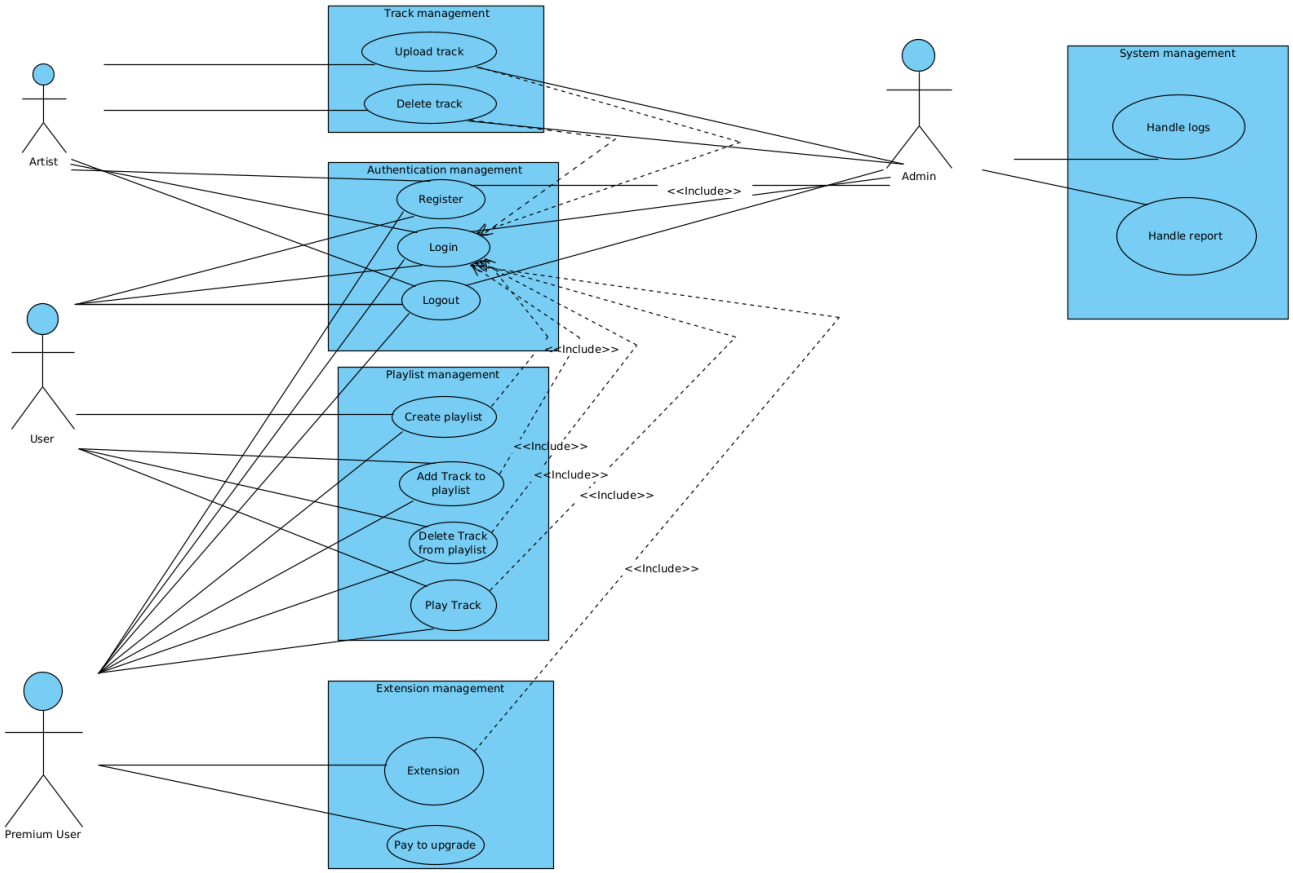
3.4. Communications Interfaces

- **Local Network Protocols:**
  - The web application shall support communication over local network protocols such as TCP/IP, UDP, and HTTP.
  - It shall utilize standard network sockets for establishing connections and transmitting data between the client and server components.
  - Support for IPv4 and IPv6 addressing schemes shall be provided to ensure compatibility with diverse network environments.
- **Data Exchange Formats:**
  - The application uses standardized data exchange format: JSON (JavaScript Object Notation) for transmitting structured data over the network.
  - Metadata for music tracks, playlists, and user preferences shall be encoded using JSON format for efficient communication between client and server.
- **API Endpoints:**
  - The web application exposes RESTful API endpoints for client-server communication, allowing clients to perform CRUD (Create, Read, Update, Delete) operations on music-related resources.
  - API endpoints shall be documented with clear specifications for request and response formats, authentication mechanisms, and error handling procedures.

4. System Features

4.1. Overall Actors

4.1.1. Overall Use Case Diagram



4.1.2. Actors Description

Actor	Description
Guest	Users in anonymous status, who do not have an account or have registered but have not logged into the system. Their access rights are limited, they are not allowed to customize the experience, and the three main use cases they perform will be signing up, and listening to music.
Normal User	Normal users primarily perform functions related to personalizing their music content and interacting with playlists.
Premium User	Premium users are users who have purchased a VIP package. Premium users can perform all the actions that a normal user can do. Additionally, premium users have access to advanced features.

Artist	Artists can perform functions related to their tracks and albums. Artist accounts require strict registration, requesting more information compared to a user's registration process.
Administrator	Administrators are individuals responsible for the content, information posted, and user reports within the system.

4.2. Guest

4.2.1. Play Track

Priority level	High
Description	Guest can select a specific track which they want to listen and serve immediately
Stimulus	Guest clicks on the track's title, album cover, or a designated play button.
Response	System starts playing the track. System displays the information of the track being played (name, artist, album, etc.). System displays playback controls (play, pause, rewind, fast forward).
Requirement	REQ-1: The server checks the availability and readiness of the selected track for playback REQ-2: If the track is available, the server streams the audio data to the user's device in real-time. REQ-3: In case the track is unavailable or there are any playback errors (e.g. network issues, corrupted audio data), the server notifies the user with an appropriate error message and it may also provide suggestions or alternative tracks for playback.

4.2.2. Search Tracks, Artists and Genres

Priority level	High
Description	Guest can search any tracks they like, artist they admire or their favorite genre to explore more nice tracks
Stimulus	Guest enters a search query, a specific genre or an artist’s name in the search bar.
Response	Guest sees a list of track results, including titles, artists, and album information. Guest can explore popular tracks or discover lesser-known gems within

	that genre. The system retrieves the artist's profile, including their top tracks, albums, and related artists.
Requirement	REQ-1: Upon receiving the search query, the server processes the input and determines the type of search requested, then retrieves relevant data from the database, including tracks, genres, and artists. REQ-2: The server presents the guest with search results based on the type of search performed. REQ-3: The presentation ensures clarity and ease of navigation for guests.

4.3. Normal User

4.3.1. Create User Account

Priority level	High
Description	Users can create their own accounts to manage personalized music experiences. Users can create playlists, save favorite tracks and albums, and potentially enjoy additional features like customized recommendations or offline listening (if applicable).
Stimulus	Navigates to the signup page and enters email address, password, and potentially other information (username, display name etc.). User clicks on the submit button.
Response	Validates user input (e.g., email format, password strength). Checks for existing email address. If valid, creates a new user account in the database and sends a confirmation email (optional). Displays a success message and redirects the user to the login page or dashboard.
Requirement	REQ-1: Validate information, add new user records to database. REQ-2: If the account creation is successful, redirects the user to the login view, indicating successful registration. REQ-3: If there are validation errors or database constraints (e.g., username or email already exists), display appropriate error messages to the user, prompting them to correct the information and resubmit the registration form.

4.3.2. Log In

Priority level	High
Description	User can access their own accounts to continue to listen to music and experience privacy extensions

Stimulus	Navigates to the login page and user provides login credentials, including email and password, in the designated login fields. User clicks on the login button.
Response	Verifies email address and password against stored credentials. If successful, creates a session for the user and redirects them to their personalized dashboard or music library. If unsuccessful, displays an error message with guidance (e.g., incorrect email/password, account disabled).
Requirement	REQ-1: Verifies if the provided email exists in the database and if the associated password matches the one provided by the user. REQ-2: If the authentication is successful, redirect the user to the application's main dashboard or designated landing page REQ-3: If the email or password is incorrect, display an error message indicating authentication failure and prompt the user to retry logging in with the correct credentials.

4.3.3. Log Out

Priority level	High
Description	User can back to guest view and log in other account.
Stimulus	User clicks the logout button and confirms logout.
Response	System displays a logout confirmation message (optional). System removes the user's login information from memory. System redirects the user to the home page or login page.
Requirement	REQ-1: Clears the user's session data and authentication tokens associated with the current session. REQ-2: Redirects the user to the application's login page or a designated landing page indicating successful logout.

4.3.4. Modify Account Details

Priority level	High
Description	User can edit their profile for authentication and update their information
Stimulus	User navigates to the account settings or profile management section within the application. User provides updated account details such as username, email, password, or any other customizable fields. User submits the updated account details by clicking on the "Save Changes" or similar button.
Response	System displays the edit account details form.

	System validates the new information. System updates the user's account information. System displays a confirmation message.
Requirement	REQ-1: Validates the updated account details to ensure they meet the required criteria (e.g., valid email format, password strength) REQ-2: Updates the user's account records in the database with the provided modifications. REQ-3: If the account details are successfully updated, display a confirmation message indicating the changes have been saved. REQ-4: If there are validation errors or any issues preventing the update, display appropriate error messages to the user, prompting them to correct the information and resubmit the form.

4.3.5. View User Information

Priority level	High
Description	This allows a user to retrieve and review their personal details within the system
Stimulus	User accesses the profile or account settings section within the application. Server: Retrieves the user's information from the database, including username, email and any other relevant details.
Response	System displays the account details page. The account details page displays the user's personal information, such as name, email, account creation date, etc.
Requirement	REQ-1: Provides options for the user to edit or modify their account details if desired REQ-2: Ensures that sensitive information is not displayed in plaintext and is appropriately masked or hidden. REQ-3: Displays an error message indicating that the user is not authorized to view the requested information.

4.3.6. Play Track

Priority level	High
Description	User can select a specific track which they want to listen and serve immediately
Stimulus	User clicks on the track's title, album cover, or a designated play button.
Response	System starts playing the track. System displays the information of the track being played (name, artist, album, etc.).

	System displays playback controls (play, pause, rewind, fast forward).
Requirement	REQ-1: The server checks the availability and readiness of the selected track for playback REQ-2: If the track is available, the server streams the audio data to the user's device in real-time. REQ-3: In case the track is unavailable or there are any playback errors (e.g. network issues, corrupted audio data), the server notifies the user with an appropriate error message and it may also provide suggestions or alternative tracks for playback.

4.3.7. Download Track

Priority level	High
Description	User can download their favorite tracks to their own devices for offline listening
Stimulus	User clicks on “Download” button, then frontend sends a request which contains the ID of the track to be downloaded.
Response	System checks if the user has permission to download the track. If the user has permission, the system starts downloading the track. System displays a download progress bar. When the download is complete, the system notifies the user. System opens the folder containing the downloaded file.
Requirement	REQ-1: Access control to ensure only authorized users can download tracks. Bandwidth management to handle concurrent downloads efficiently. REQ-2: Upon successful authorization and bandwidth allocation, the server delivers the requested track to the user, ensuring a seamless download experience REQ-3: If error occurs, provide options for the user to try downloading the track again later or to explore other available tracks.

4.3.8. Create Playlist

Priority level	High
Description	User can have a list of favorite tracks to have private experience and listen sequentially
Stimulus	The user navigates to the playlist creation section within the application interface and initiates the process of creating a new playlist. The user provides a name for the playlist and may optionally add a description or select a cover image.
Response	The system prompts the user to provide a playlist name and an optional

	description for the new playlist. The system validates the playlist name and description. If valid, it creates an empty playlist with the specified name and description.
Requirement	REQ-1: Upon receiving the playlist creation request from the frontend, the server validates the provided playlist name to ensure it meets any required criteria (e.g., minimum length, uniqueness) and checks the user's authentication status to ensure they are logged in and authorized to create playlists REQ-2: If the provided playlist name is valid and the user is authenticated, the server creates a new playlist record in the database. It associates the playlist with the user's account and stores any additional metadata provided by the user, such as description or cover image. REQ-3: Upon successful creation of the playlist, the server confirms the creation process and notifies the user with a success message REQ-4: If there are any validation errors or issues preventing the creation of the playlist (e.g., duplicate name, invalid characters), the server notifies the user with an appropriate error message.

4.3.9. Add Track to Playlist

Priority level	High
Description	User can add more tracks into their own playlists which they create before
Stimulus	The user selects a specific track from the application's library or playlist and chooses the option to add it to an existing playlist. The user may be presented with a dropdown menu or a list of their existing playlists to choose from.
Response	Updates the playlist data in the database to include the newly added tracks. Reflects the changes in the user interface, showing the added songs within the playlist.
Requirement	REQ-1: If the user is authenticated and the selected track and playlist are valid, the server updates the database to associate the track with the selected playlist. REQ-2: It may also handle any additional metadata associated with the track in the context of the playlist, such as track order or user-specific preferences. REQ-3: Upon successful addition of the track to the playlist, the server confirms the action and notifies the user with a success message. It may provide feedback within the application interface, such as updating the playlist view to reflect the newly added track. REQ-4: If there are any issues preventing the addition of the track to the playlist (e.g., invalid track or playlist, authorization failure), the server notifies the user with an appropriate error message. It prompts the user to

	correct the issue and retry the operation.

4.3.10. Manage Playlist

Priority level	High
Description	User can edit such as add more tracks, edit playlist name, remove some songs,... in their own playlists which they create before
Stimulus	The user navigates to the playlist management section within the application interface, where they can perform various actions such as renaming playlists, changing playlist settings, reordering tracks, or deleting playlists.
Response	Updates the playlist data in the database based on user actions. Reflects the changes in the user interface, displaying the updated playlist details. Optionally allows deleting tracks from the playlist or deleting the entire playlist.
Requirement	REQ-1: Depending on the specific action requested by the user (e.g., renaming playlist, reordering tracks), the server updates the database accordingly. It ensures data integrity and consistency while applying the requested changes to the playlist. REQ-2: Upon successful completion of the requested playlist management action, the server confirms the operation and notifies the user with a success message. It may also provide feedback within the application interface, such as updating the playlist view to reflect the changes made REQ-3: If there are any issues preventing the completion of the requested playlist management action (e.g., authorization failure, database error), the server notifies the user with an appropriate error message. It prompts the user to correct the issue and retry the operation.

4.3.11. Search

Priority level	High
Description	User can search any tracks they like, artist they admire or their favorite genre to explore more nice tracks
Stimulus	User enters a search query, a specific genre or an artist’s name in the search bar.
Response	User sees a list of track results, including titles, artists, and album information. User can explore popular tracks or discover lesser-known gems within that genre.

	The system retrieves the artist's profile, including their top tracks, albums, and related artists.
Requirement	REQ-1: Upon receiving the search query, the server processes the input and determines the type of search requested, then retrieves relevant data from the database, including tracks, genres, and artists. REQ-2: The server presents the user with search results based on the type of search performed. REQ-3: The presentation ensures clarity and ease of navigation for users.

4.3.12. Advanced Filters

Priority level	Medium
Description	User can explore more feature by some clicks into advance filters
Stimulus	The user accesses the advanced filter option within the application interface, which allows them to refine search results or narrow down content based on specific criteria. This could include filters for genre, release date, duration, popularity, and more.
Response	The system refines the search results based on the selected filters. Users can find recently released tracks, popular hits, or family-friendly content.
Requirement	REQ-1: Upon receiving the user's request to apply advanced filters, the server processes the selected filter criteria and retrieves relevant data from the database. It applies the specified filters to the dataset, ensuring that only content matching the user's criteria is included in the results. REQ-2: The server presents the user with filtered search results based on the selected criteria. It ensures that the filtered content meets the user's preferences and provides options for further refinement or exploration REQ-3: If there are no search results matching the user's selected filter criteria, the server notifies the user with a message indicating that there are no matches found. It may suggest adjusting the filter settings or provide alternative suggestions for finding relevant content.

4.4. Premium User

This actor has all use case of "Normal User" actor above and some addition extensions blow

4.4.1. Upgrade Account

Priority level	High
Description	Allows a user to elevate their account status by opting for a premium or

	advanced subscription level. Upgrading provides additional features, benefits, and an enhanced music experience.
Stimulus	User access setting section and click to Upgrade To Premium User fills in blank more information and choose one of upgrade tickets User selects payment method and pay
Response	The system provides higher permissions to the user's account User accounts display Premium titles and unlock advanced extensions
Requirement	REQ-1: The system must display available subscription plans clearly, including their features and pricing and user must be able to compare different plans easily. REQ-2: The system must securely process payment information provided by the user. Payment methods must be supported. REQ-3: If the payment fails, the system notifies the user and provides troubleshooting steps. REQ-4: The user can cancel the upgrade process at any point before final confirmation.

4.5. Artist

4.5.1. Create Artist Account

Priority level	High
Description	Artists can create their own accounts to create albums and upload more tracks to serve listeners. Artist can create their own styles to attract more listeners.
Stimulus	Navigates to the signup page and enters email address, password, and potentially other information (artist name, display name etc.). Artist clicks on the submit button.
Response	Validates artist input (e.g., email format, password strength). Checks for existing email address. If valid, creates a new artist account in the database and sends a confirmation email (optional). Displays a success message and redirects the artist to the login page or dashboard.
Requirement	REQ-1: Validate information, add new artist records to database. REQ-2: If the account creation is successful, redirects the artist to the login view, indicating successful registration. REQ-3: If there are validation errors or database constraints (e.g., username or email already exists), display appropriate error messages to the artist, prompting them to correct the information and resubmit the registration form.

4.5.2. Log In

Priority level	High
Description	Artist can access their own accounts to continue to listen to music and experience privacy extensions
Stimulus	Navigates to the login page and artist provides login credentials, including email and password, in the designated login fields. Artist clicks on the login button.
Response	Verifies email address and password against stored credentials. If successful, creates a session for the artist and redirects them to their personalized dashboard or music library. If unsuccessful, displays an error message with guidance (e.g., incorrect email/password, account disabled).
Requirement	REQ-1: Verifies if the provided email exists in the database and if the associated password matches the one provided by the artist. REQ-2: If the authentication is successful, redirect the artist to the application's main dashboard or designated landing page REQ-3: If the email or password is incorrect, display an error message indicating authentication failure and prompt the artist to retry logging in with the correct credentials.

4.5.3. Log Out

Priority level	High
Description	Artist can back to guest view and log in other account.
Stimulus	Artist clicks the logout button and confirms logout.
Response	System displays a logout confirmation message (optional). System removes the artist's login information from memory. System redirects the artist to the home page or login page.
Requirement	REQ-1: Clears the artist's session data and authentication tokens associated with the current session. REQ-2: Redirects the artist to the application's login page or a designated landing page indicating successful logout.

4.5.4. Upload Track

Priority level	High
Description	Upload track with a catchy title and choose a genre to help listeners find it.

Stimulus	Track file (MP3, MP4, WAV) and information about the uploaded track, such as title, artist name, genre, duration, etc.
Response	The system validates the file format, checks for duplicates, and ensures that all required information is provided. The track is uploaded to Spotify's database. The artist receives a confirmation message indicating successful upload.
Requirement	REQ-1: validate file format, store uploaded tracks REQ-2: If successful, display the message "Track successfully uploaded." REQ-3: If the uploaded file format is unsupported, prompt the user with a message explaining the supported formats and request re-upload. REQ-4: If the upload process is interrupted due to network issues or server downtime: Display a message informing the user about the interruption and advise them to retry the upload later.

4.5.5. Modify Track Information

Priority level	High
Description	Artists can edit their own track information.
Stimulus	The artist navigates to the track management section within the application and selects a specific track to modify. The artist provides updated information such as track title, album, genre, or any other relevant details.
Response	System displays the edit track information form. System validates the new information. System updates the track information. System displays a confirmation message.
Requirement	REQ-1: The server validates the updated track information to ensure it meets the required criteria and adheres to any constraints (e.g., character limits, data formats). It then updates the corresponding record in the database with the provided modifications. REQ-2: If the track information is successfully updated, the server confirms the changes and notifies the user with a success message. REQ-3: If there are any validation errors or issues preventing the update, the server notifies the user with appropriate error messages, indicating the nature of the problem and prompting them to correct the information accordingly.

4.5.6. Create Album

Priority level	High
Description	Artists can create one or some albums.

Stimulus	The artist navigates to the album creation section within the application interface and initiates the process of creating a new album. The artist provides a name for the album and may optionally add a description or select a cover image.
Response	The system prompts the artist to provide a album name and an optional description for the new album. The system validates the album name and description. If valid, it creates an empty album with the specified name and description.
Requirement	REQ-1: Upon receiving the album creation request from the frontend, the server validates the provided album name to ensure it meets any required criteria (e.g., minimum length, uniqueness) and checks the artist's authentication status to ensure they are logged in and authorized to create albums REQ-2: If the provided album name is valid and the artist is authenticated, the server creates a new album record in the database. It associates the album with the artist's account and stores any additional metadata provided by the artist, such as description or cover image. REQ-3: Upon successful creation of the album, the server confirms the creation process and notifies the artist with a success message REQ-4: If there are any validation errors or issues preventing the creation of the album (e.g., duplicate name, invalid characters), the server notifies the artist with an appropriate error message.

4.6. Administrator

4.6.1. User Management

Priority level	High
Description	Administrator is the process of adding, editing, deleting accounts, assigning roles, granting/revoking permissions, monitoring, and reporting user activity.
Stimulus	A new employee needs a system account (user creation request). An existing user's role or permissions need to be changed (access control modification). Suspicious activity is detected from a user account (security concern).
Response	The administrator uses user management tools to create new accounts, assign roles, adjust permissions, or potentially suspend/delete accounts.
Requirement	REQ-1: The administrator should verify user identity and access requests before granting access. REQ-2: User roles and permissions should be assigned based on the principle of least privilege (granting only the minimum access needed for tasks).

	REQ-3: User activity logs should be monitored to identify potential security breaches or misuse of the system.

4.6.2. Content Management

Priority level	High
Description	A Content Management Administrator keeps the content flowing by managing users, content, and the system itself.
Stimulus	A content creator submits a new article for publication. An existing blog post needs edits based on user feedback. The website experiences a surge in traffic, requiring performance optimization.
Response	The CMA uses the CMS tools to review content, publish articles, manage user permissions, or adjust server configurations.
Requirement	REQ-1: CMAs should ensure content adheres to established editorial guidelines and quality standards. REQ-2: User access and permissions within the CMS should be managed for optimal security and content ownership control. REQ-3: CMAs should monitor system performance and implement changes (e.g., caching, image optimization) to ensure smooth content delivery.

4.6.3. System Configuration

Priority level	High
Description	System Configuration (Administrator) refers to the process of managing a system's settings to optimize performance, security, and user experience. This typically involves an administrator making adjustments through built-in tools or configuration files.
Stimulus	An administrator identifies a need to improve system performance (e.g., slow boot times), enhance security (e.g., new software vulnerabilities), or customize user experience (e.g., disable unnecessary startup programs).
Response	The administrator accesses the system configuration tools (e.g., BIOS settings, Control Panel applets) and makes adjustments to relevant settings.
Requirement	REQ-1: The administrator should understand the potential impact of each configuration change before implementing it. REQ-2: Configuration changes should be documented for future reference and troubleshooting. REQ-3: The administrator may need to test the system after making

	changes to ensure they achieve the desired outcome without causing unintended consequences.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

The application shall adhere to the following performance requirements to ensure a responsive and reliable user experience:

- *Response Time*: Display the initial content within 2 seconds of user interaction under standard operating conditions. Subsequent content, such as songs and playlists, shall load within 3 seconds under the same conditions.

- *System Availability:* The service shall be available 24/7, with a monthly uptime of at least 99.99%, excluding scheduled maintenance windows.

- *Scalability:* Support up to 1000 concurrent users without degradation of performance.

- *Latency*: Have a network latency of less than 150 milliseconds for 95% of all transactions.

- *Capacity*: The database shall handle at least 10 thousand songs, with the ability to scale as the library grows.

- *Resource Utilization*: The application shall optimize resource usage, with the backend services not exceeding 70% CPU utilization and 80% memory utilization under peak load conditions.

5.2. Safety Requirements

- *Content Moderation:* Implement robust content moderation policies to prevent the distribution of harmful or inappropriate content. This includes automated filtering, user reporting mechanisms, and a dedicated review team.

- *Data Protection:* User data shall be protected in accordance with the highest industry standards, including encryption of sensitive data, secure data storage solutions, and regular security audits.

- *User Privacy:* Adhere to strict privacy policies, ensuring that user data is not shared without explicit consent and that users have control over their personal information.

- *Safety Features:* offer safety features such as parental controls, content filters, and privacy settings, allowing users to customize their experience and manage risks.

- *Safety Education:* Provide resources and guidance to educate users about safe online practices and the responsible use of the platform.

- *Incident Response:* Have a clear and accessible process for users to report safety concerns and incidents, with a commitment to timely and appropriate action.

5.3. Security Requirements

- *User Authentication and Authorization*
  - Use strong password policies and securely hash passwords.
  - Define access control rules based on user roles.
- *Data Protection*
  - Encrypt sensitive data during transmission (HTTPS) and at rest.
  - Use industry-standard encryption algorithms.
  - Validate and sanitize user inputs to prevent attacks.
- *Secure APIs*
  - Implement rate limiting and authenticate API requests.
  - Validate input parameters.
- *Infrastructure Security*
  - Regularly update server software.
  - Harden server configurations.
  - Set up intrusion detection systems and incident response procedures.
- *Third-Party Dependencies*
  - Use reputable libraries.
  - Monitor vulnerabilities.

5.4. Software Quality Attributes

- *Design Qualities:*
  - Modifiability: This refers to how easily the software can be modified or updated without causing unintended side effects. A well-designed software system should be flexible and allow for changes to be made efficiently.
  - Maintainability: This is the ease with which the software can be maintained over time. It includes aspects such as code readability, documentation, and adherence to coding standards.
  - Testability: Testability measures how easily the software can be tested to ensure its correctness and reliability. A well-designed software system should facilitate comprehensive testing at various levels, from unit tests to system tests.

- *Runtime Qualities:*

- Performance: Performance relates to the speed and responsiveness of the software while executing tasks. It includes factors such as latency, throughput, and resource utilization.
- Scalability: Scalability refers to the ability of the software to handle increasing workload or user demand by adding resources such as servers or processing power.
- Reliability: Reliability measures the ability of the software to perform consistently under varying conditions without failures or errors. It includes aspects such as fault tolerance and error recovery mechanisms.

- *System Qualities:*
  - Security: Security involves protecting the software from unauthorized access, data breaches, and other malicious activities. It encompasses authentication, authorization, encryption, and other security measures.
  - Availability: Availability refers to the accessibility of the software to users when needed. It involves minimizing downtime and ensuring continuous operation through redundancy and fault tolerance mechanisms.
  - Interoperability: Interoperability measures the ability of the software to interact and integrate seamlessly with other systems or components. It includes support for standard protocols and data formats.

- *User Qualities:*
  - Usability: Usability refers to how easy and intuitive the software is to use for its intended users. It involves aspects such as user interface design, navigation, and user feedback.
  - Accessibility: Accessibility ensures that the software can be used by people with disabilities, including those with visual, auditory, motor, or cognitive impairments. It involves providing alternative means of interaction and adhering to accessibility standards.
  - Acceptability: Acceptability measures the extent to which the software meets the expectations and requirements of its users. It involves gathering user feedback and incorporating user preferences into the design and development process.

5.5. Business Rules

- *User Roles and Permissions:*
  - Users can search for and play music, create and manage playlists, follow artists and other users and personalize their music recommendations
  - Artists and Content Creators can upload and manage their music content, view analytics related to their tracks and have additional features (e.g., artist profiles, promotional tools).
  - Administrators need to manage user accounts and permissions, handle copyright infringement claims and monitor system performance and security
- *Content Licensing and Copyright:*
  - Ensure that only licensed music content is available for streaming.

    - Enforce takedown procedures for unauthorized content.
- *User Behavior and Community Guidelines:*
    - Enforce community guidelines to prevent abusive behavior (e.g., hate speech, harassment).
    - Implement reporting mechanisms for inappropriate content.
    - Address user complaints and disputes.
- *Quality of Service (QoS):*
    - Prioritize premium users during peak times to ensure smooth streaming.
    - Limit free users' access to maintain service quality.
    - Implement adaptive streaming based on network conditions.
- *Privacy and Data Protection:*
    - Protect user data (e.g., personal information, listening history).
    - Obtain explicit consent for data collection and personalized recommendations.

6. Other Requirements

The system will incorporate stringent security protocols, including regular audits, to safeguard user and system data. Content management will be streamlined through an advanced backend, supporting copyright adherence and royalty distribution. Scalability and performance are key, with the system designed to handle growth and maintain service quality. Lastly, legal compliance and ethical considerations will be at the forefront, ensuring content legality and data transparency.

Appendix A: Glossary

Term	Definition
API	Application Programming Interface. A communication protocol enabling interaction between servers.
Authentication	The process of proving that you are registered and have the relevant permissions to access specific data.
FAQs	Frequently Asked Questions. A list of common questions and answers that address typical user concerns and issues related to the topic.
Content Delivery Networks	A network of servers distributed globally that work together to provide fast delivery of internet content, including music streaming.

Appendix B: Analysis Models