National Technological Institute of Mexico

Software Requirements Specification (SRS)

The Ultimate Music Streaming and Exploration Experience

Yael Guillermo González Hernández
Manuel Enrique Toledo Ramirez
Christopher Alexander Román Castro

December 6, 2023

Document Information

Date	Revision	Author	Quality Dep. Verification
October 2, 2023		Yael González Hernández	
		Manuel Toledo Ramírez	
		Christopher Román Castro	

Contents

1 Introduction

This document is a Software Requirements Specification (SRS) for the Information System for Process Management and Inventory Control. This specification has been structured based on the guidelines provided by the IEEE Recommended Practice for Software Requirements Specifications ANSI/IEEE 830, 1998 standard.

1.1 Purpose

The purpose of this document is to outline the specific requirements and functionalities of the MusicStream App. It serves as a comprehensive reference for all stakeholders involved in the project, including developers, designers, testers, project managers, and clients. The document aims to provide a clear and detailed description of the software's features, interactions, and constraints, ensuring a common understanding of project goals and expectations.

1.2 Scope

The product to be developed is referred to as "Manía." Manía is a web-based music streaming and discovery application, which shares similarities with popular services like Spotify. It offers users the ability to listen to a vast catalog of music, create personalized playlists, discover new music, and engage in social interactions related to music. The scope of the "Manía" project aligns with and is consistent with any higher-level documents, such as a "System Description," that may exist. It is essential to maintain consistency between this Software Requirements Specification (SRS) and any overarching documents to ensure that the project's objectives and functionalities are accurately represented and coordinated across all project documentation.

1.3 Involved Staff

Name	González Hernández Yael Guillermo
Role	Scrum Owner
Contact information	l20211785@tectijuana.edu.mx
Aprobation	

Name	Román Castro Christopher Alexander
Role	Scrum Master
Contact information	l20211837@tectijuana.edu.mx
Aprobation	

Name	Toledo Ramirez Manuel Enrique
Role	Scrum Team
Contact information	manuel.toledo193@tectijuana.edu.mx
Aprobation	

1.4 Definitions, Acronyms, and Abbreviations

Name	Description
SRS	Software Requirements Specification - A comprehensive document outlining the specific requirements and functionalities of the MusicStream App.
MusicStream App	The web-based music streaming and discovery application to be developed, similar to services like Spotify, offering features such as listening to music, creating playlists, and discovering new music.
API	Application Programming Interface - A set of rules and protocols that allows different software applications to communicate with each other.

1.5 References

Number	Title	Rute	Date	Author
1	Learning Web Design	O'Reilly Media	2018	Jennifer Niederst Robbins
2	Eloquent JavaScript: A Modern Introduction to Programming	No Starch Press	2020	Marijn Haverbeke
3	Node.js Design Patterns	Packt Publishing	2016	Mario Casciaro
4	Pro JavaScript for Web Apps	Apress	2012	Adam Freeman
5	Web Audio API	O'Reilly Media	2013	Boris Smus
6	Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems	O'Reilly Media	2017	Martin Kleppmann
7	Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development	Apress	2011	Peter Lubbers, Brian Albers y Frank Salim
8	Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython	O'Reilly Media	2017	Wes McKinney
9	JavaScript: The Good Parts	O'Reilly Media	2008	Douglas Crockford

1.6 Abstract

This document consists of three sections. The first section provides an introduction and offers an overview of the system resource specification.

In the second section of the document, a general system description is provided to understand its main functions, associated data, and the factors, constraints, assumptions, and dependencies that affect development, without delving into excessive details.

Finally, the third section of the document is where the system requirements are defined in detail.

2 Overall Description

The "MusicStream App" is a web-based application designed to enable users to easily and conveniently listen to music online. With a focus on high-quality music and an engaging user experience, this application aims to offer a wide variety of features and functionalities to meet the needs of music enthusiasts worldwide.

The application will provide users with the ability to search and play their favorite songs, create customized playlists, discover new music through recommendations, and stay updated on the latest developments in the music world. Additionally, the "MusicStream App" will offer an intuitive and attractive user interface that allows users to enjoy a seamless listening experience from any Internet-accessible device.

2.1 Product Perspective

The "MusicStream App" positions itself in the market as a versatile and comprehensive music streaming solution. While there are other music streaming platforms in the market, the "MusicStream App" will differentiate itself through the following key perspectives:

Catalog Variety: The application will offer an extensive music catalog covering different genres, languages, and eras, providing users with a broad selection of songs to choose from.

Personalized Recommendations: It will use advanced recommendation algorithms to provide users with music recommendations based on their preferences and listening habits.

Attractive User Interface: The application will feature a modern and intuitive user interface, ensuring an engaging and user-friendly experience.

Multiplatform Accessibility: Users will be able to access the "MusicStream App" from various devices, including desktop computers, tablets, and mobile phones, allowing them to enjoy music anywhere and anytime.

2.2 Product Functions

Music Playback: Users can upload and play songs from their library. Continuous playback and the ability to pause, skip forward, and rewind.

Music Catalog: Search and browse by genre, artist, album, and song. Ability to add songs to customized playlists.

Playlists: Users can create and manage customized playlists. Drag-and-drop functionality for organizing songs in playlists.

Audio Quality: Support for various audio qualities (e.g., standard and high quality). Audio equalizer with presets.

Sharing and Social: Users can share songs and playlists on social networks. Friend tracking feature and viewing their playlists.

History and Recommendations: Playback history log for personalized recommendations. Song and artist suggestions based on listening history.

User Profiles: Users can create personalized profiles with photos and profile details. Playback preferences and notification settings.

Comments and Ratings: Ability for users to leave comments and ratings on songs and albums. "Like" or "Dislike" feature for songs.

Background Playback: Continuous music playback even when the web page is in the background or minimized.

2.3 User Classes and Characteristics

User Type	Background	Activities
Listener	No specific technical background required	• Search for and listen to music • Create and manage personal playlists • Explore recommended music based on preferences • Interact with social features like following friends and sharing playlists

User Type	Background	Activities
Artist	Musical background and expertise in music creation	• Upload and manage their music tracks • Create artist profiles and bios • Monitor track performance and analytics • Engage with fans and listeners through comments and messages

User Type	Background	Activities
Curator	Music enthusiast with knowledge of different music genres	• Create and curate playlists for public or private use • Share curated playlists with the community • Discover and recommend new and trending music • Collaborate with other users to create themed playlists

2.4 ER Diagram

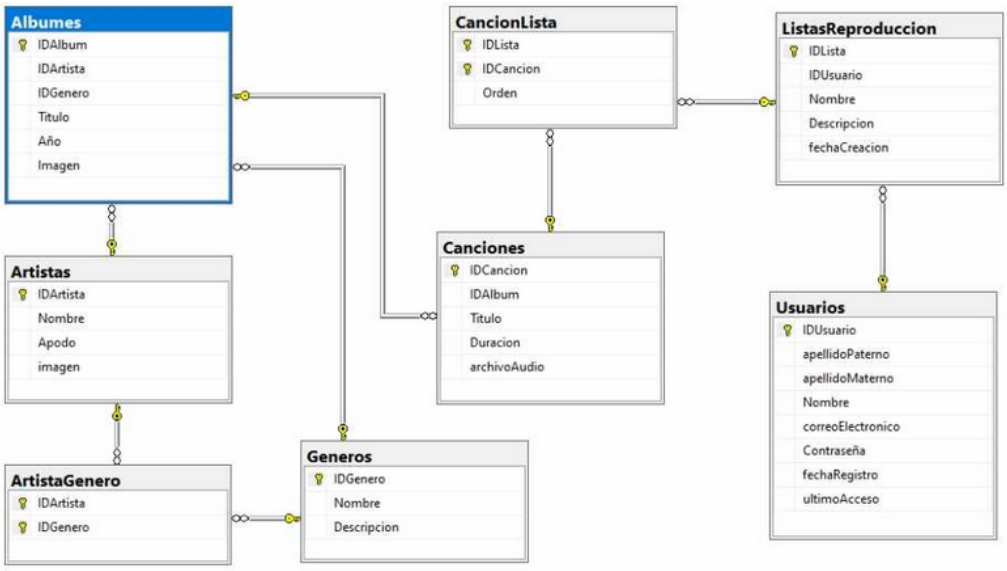

Figure 1: Data modeling Diagram

2.5 Constraints

1. Limited development timeline to meet project milestones.
2. Budget constraints for software development and maintenance.
3. Compliance with music licensing and copyright regulations.
4. Compatibility with a wide range of web browsers and devices.
5. Adherence to security and privacy standards for user data.
6. Scalability to accommodate potential growth in user base.
7. Integration with external APIs for music streaming and metadata.
8. Responsiveness to user feedback and feature requests.
9. Resource constraints, including server capacity and bandwidth.
10. Accessibility compliance to ensure inclusivity for users with disabilities.

2.6 Assumptions and Dependencies

1. Availability of a stable internet connection for users.
2. Access to a diverse and up-to-date music catalog for streaming.
3. Collaboration with music artists and record labels for content licensing.
4. Availability of development tools and frameworks for web application development.
5. User engagement and adoption of social features within the app.
6. Dependence on third-party services for analytics and recommendation algorithms.
7. Availability of user-generated content such as playlists and comments.
8. Availability of user profiles for personalized recommendations.

9. Continuous updates and improvements to keep the app competitive.
10. Integration with payment gateways for premium subscription services.

2.7 Projected System Evolution

Phase 1: Initial Development In this phase, the primary focus will be on creating the foundation of the application and its core features:

- Design and Development of the Web Platform: The "MusicStream App" website will be developed with an attractive design and an intuitive user interface. Basic functionality for music search, playback, and user registration will be implemented.
- Music Player Integration: A basic music player will be implemented, allowing users to search for and listen to songs.
- User Management: A user management system will be implemented, including registration, login, and basic user profiles.

Phase 2: Feature Expansion and Content Enrichment In this phase, additional features will be added, and the available content will be expanded:

- Expanded Music Catalog: The available music catalog will be expanded by incorporating new songs and albums from artists.
- Customized Playlists: Users will be allowed to create and manage customized playlists.
- Suggestions and Recommendations: A recommendation system will be implemented to suggest music based on the user's listening history.

Phase 3: Optimization and Enhancement In this phase, improvements will be made to the platform to optimize performance and the user experience:

- Speed Optimization: Work will be done to optimize page load speed and music playback for a smoother experience.
- User Interface Enhancements: Adjustments to the user interface will be made to make it more appealing and user-friendly.

3 External Interface Requirements

In this section, we will discuss the various interfaces that play a crucial role in our music player web application. The system interfaces can be categorized into the following types:

3.1 User Interfaces

User interfaces are the points of interaction between users and the software system. In the context of our music player web application, these interfaces include:

3.1.1 Graphical User Interface (GUI)

The GUI represents the visible part of the application where users directly interact with features such as buttons, menus, search bars, and playlists. Users use the GUI to search, play, and manage music.

3.1.2 Screens and Pages

Each screen or page within the application represents a unique user interface. For instance, the home page, search page, song playback page, and user profile pages are examples of specific user interfaces.

3.2 Hardware Interfaces

Hardware interfaces refer to the interaction between the software and physical devices. In a music player web application, this could encompass:

3.2.1 Audio Interface

The application must interact with the user's device speakers and headphones to deliver sound.

3.2.2 Network Interface

This interface is used to connect to the internet for music streaming from remote servers.

3.3 Software Interfaces

Software interfaces are communication points between different modules or components within the software. Some examples relevant to our music application could be:

3.3.1 APIs (Application Programming Interfaces)

These interfaces allow our application to communicate with external services, such as music databases or authentication services.

3.4 Server Hardware

The server hardware for a music streaming web application plays a pivotal role in ensuring smooth and uninterrupted service for users. To deliver a seamless music streaming experience, the server infrastructure should typically comprise:

3.4.1 Powerful CPUs

High-performance processors are essential for efficiently handling user requests, managing database operations, and processing audio streams in real-time. Multi-core processors are commonly used to distribute processing loads effectively.

3.4.2 Sufficient RAM

Adequate RAM is crucial for caching frequently accessed data, optimizing database performance, and ensuring quick response times. The specific amount of RAM required depends on the user base and expected concurrent connections.

3.5 Communication Interfaces

Communication interfaces are used for data transmission between systems or components. In the context of a music player application, these might include:

3.5.1 Network Interfaces

Used for transmitting music data from remote servers or for synchronizing information with other devices or users.

3.5.2 Component Communication

Interfaces used for different parts of the application to communicate with each other, such as the interface between the search and playback components.

4 System Features

4.1 Functional Requirements

Requirement Identification	RF01
Requirement Name	User Authentication
Characteristics	Users must authenticate to access any part of the system.
Requirement Description	The system can be accessed by any user depending on the module they are in and their level of accessibility.
Requirement Priority	High

Requirement Identification	RF02
Requirement Name	Song Search
Characteristics	Users can search for songs by title, artist, or genre.
Requirement Description	The system should provide a search feature that allows users to find songs based on various criteria, making it easy to discover music.
Requirement Priority	Medium

Requirement Identification	RF03
Requirement Name	Playlist Creation
Characteristics	Users can create and manage personalized playlists with songs of their choice.
Requirement Description	The system should allow users to create, edit, and organize playlists, providing a customized music listening experience.
Requirement Priority	High

Requirement Identification	RF04
Requirement Name	Music Recommendations
Characteristics	The system should suggest music based on the user's listening history and preferences.
Requirement Description	The system will use algorithms to recommend songs and artists to users, enhancing their music discovery experience.
Requirement Priority	High

Requirement Identification	RF05
Requirement Name	Continuous Playback
Characteristics	Users can enjoy uninterrupted music playback even when the app is in the background or minimized.
Requirement Description	The system should support background playback, allowing users to enjoy music without interruptions while using other apps.
Requirement Priority	Medium

Requirement No.	**Requirement Name**	**Type**	**Requirement Source**	**Priority**
RF-001	User Registration	Requirement	Stakeholder	High
RF-002	User Login	Requirement	Stakeholder	High
RF-003	Music Search	Requirement	Stakeholder	Medium
RF-004	Music Playback	Requirement	Stakeholder	High
RF-005	Playlist Management	Requirement	Stakeholder	Medium

4.2 Use Cases

User Registration: A new user can register on the application by providing their first name, last name, email, and password.

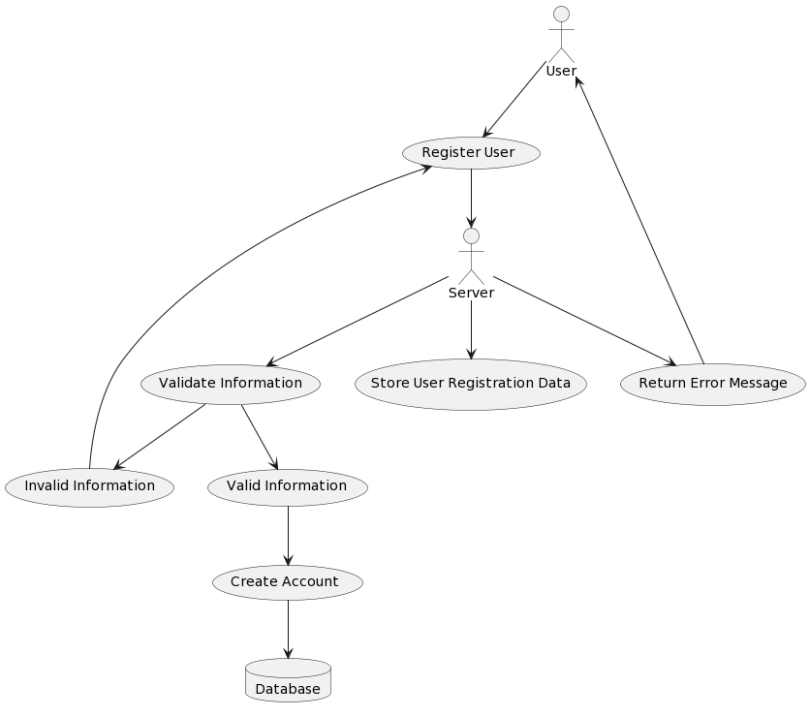

Figure 2: Structured Diagram

Login: A registered user can log in to the application using their email and password.

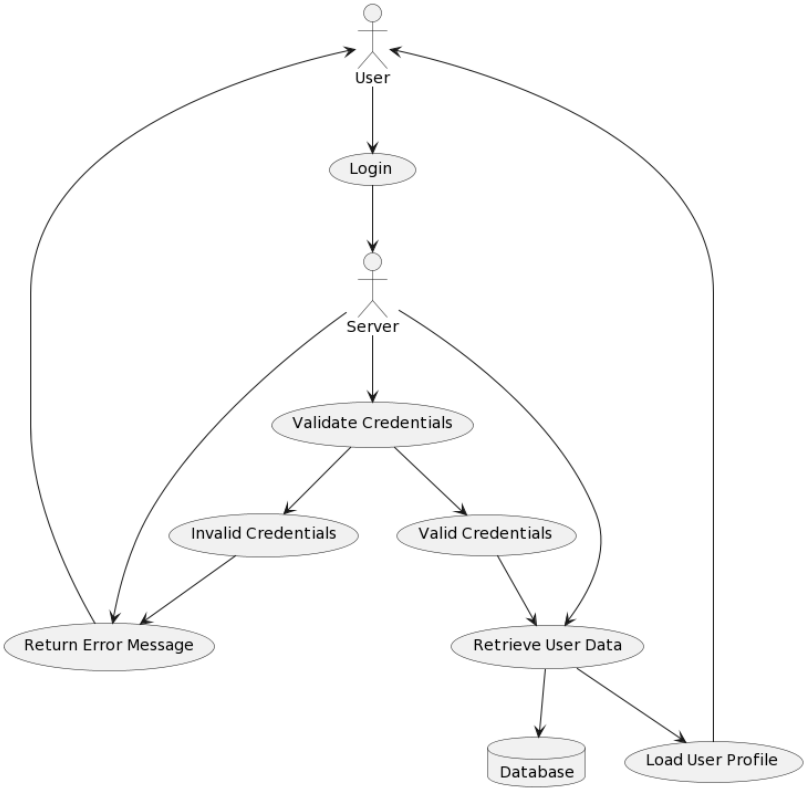

Figure 3: Structured Diagram

Explore Music: Users can browse the music available on the platform. They can search for albums by genre, artists, album titles, or songs.

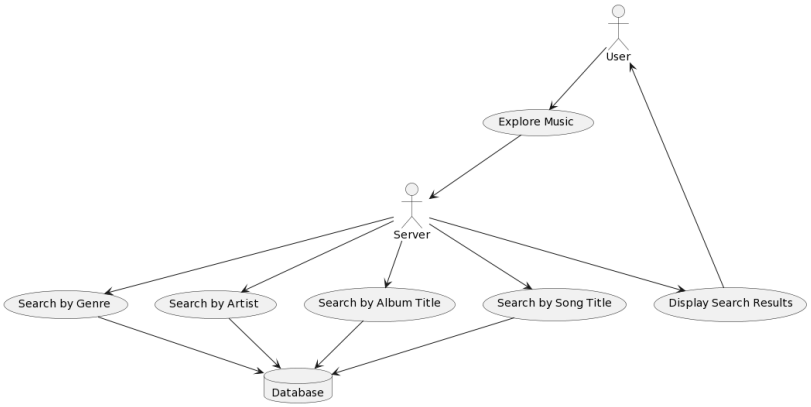

Figure 4: Structured Diagram

Create Playlists: Users can create customized playlists and add songs of their choice to them. They can give names and descriptions to their playlists.

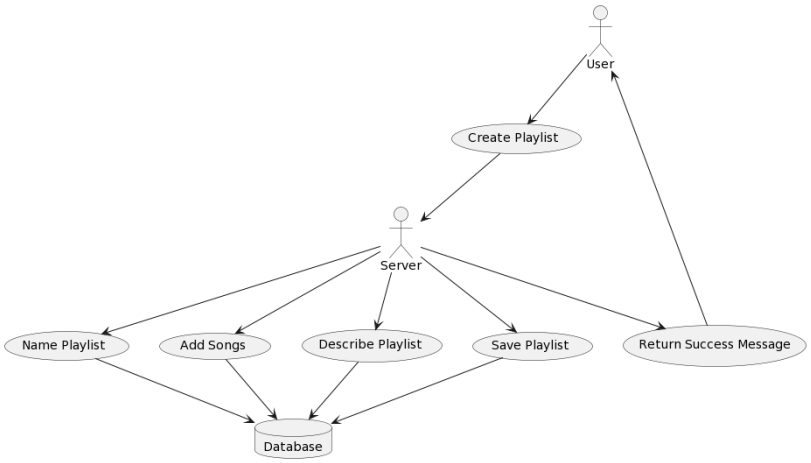

Figure 5: Structured Diagram

The UML class diagram presented below is a visual representation of the static structure of the 'Mania' music platform. This system provides users with the ability to explore, play, and manage their favorite music while also offering artists a platform to showcase their work.

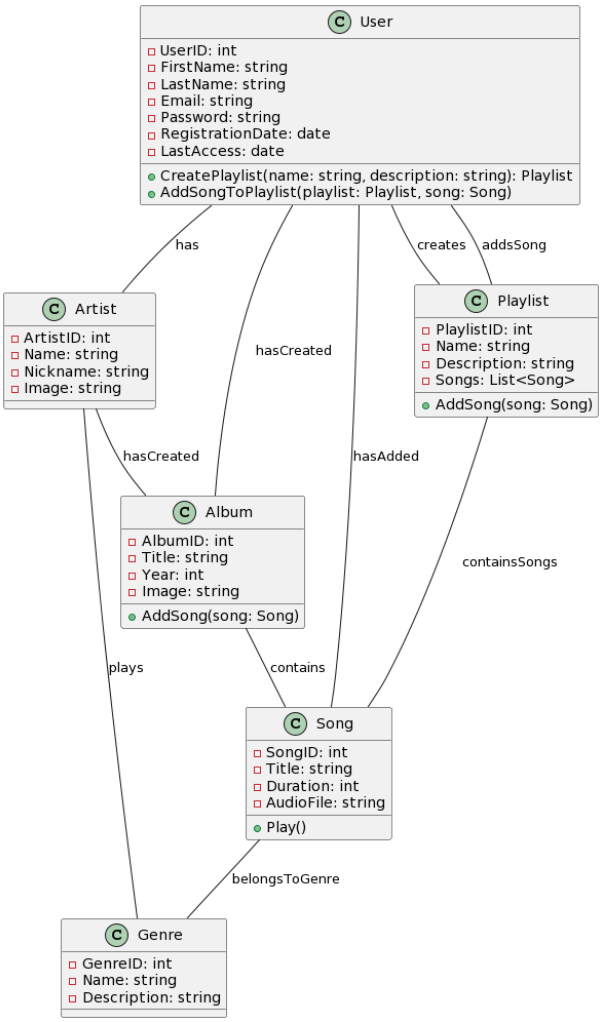

Figure 6: Modeling data Diagram

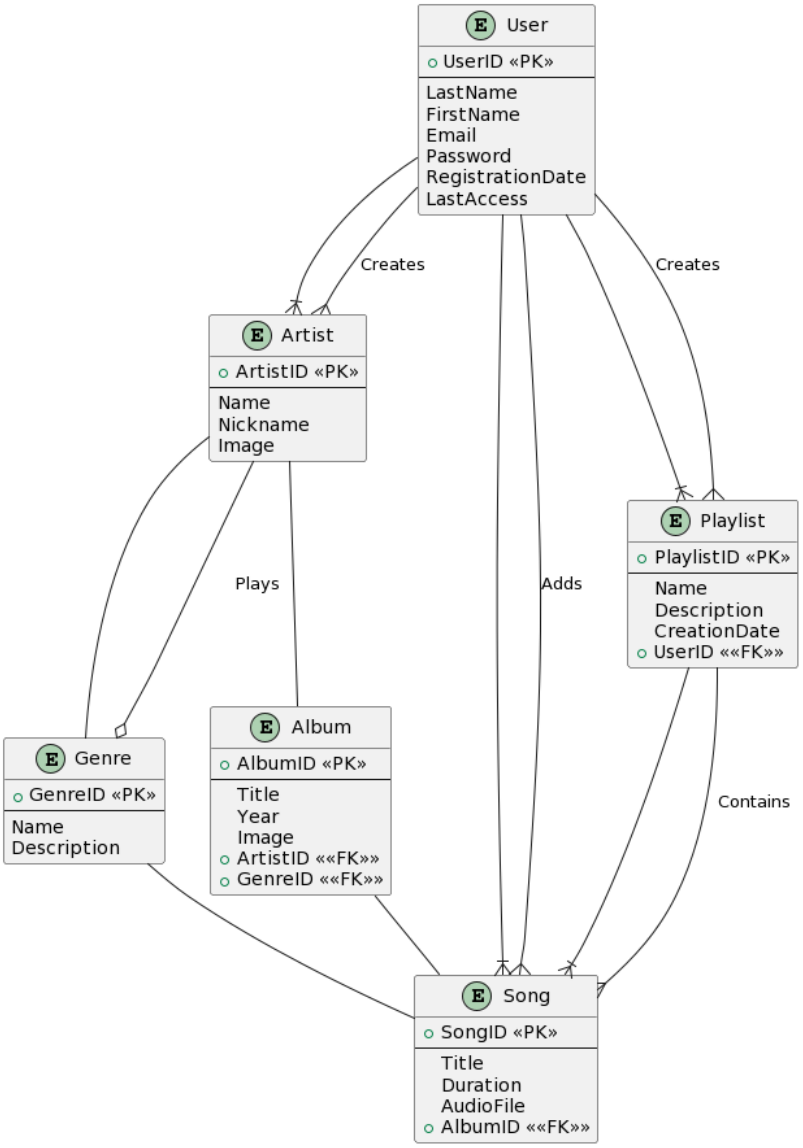

Figure 7: Modeling data Diagram

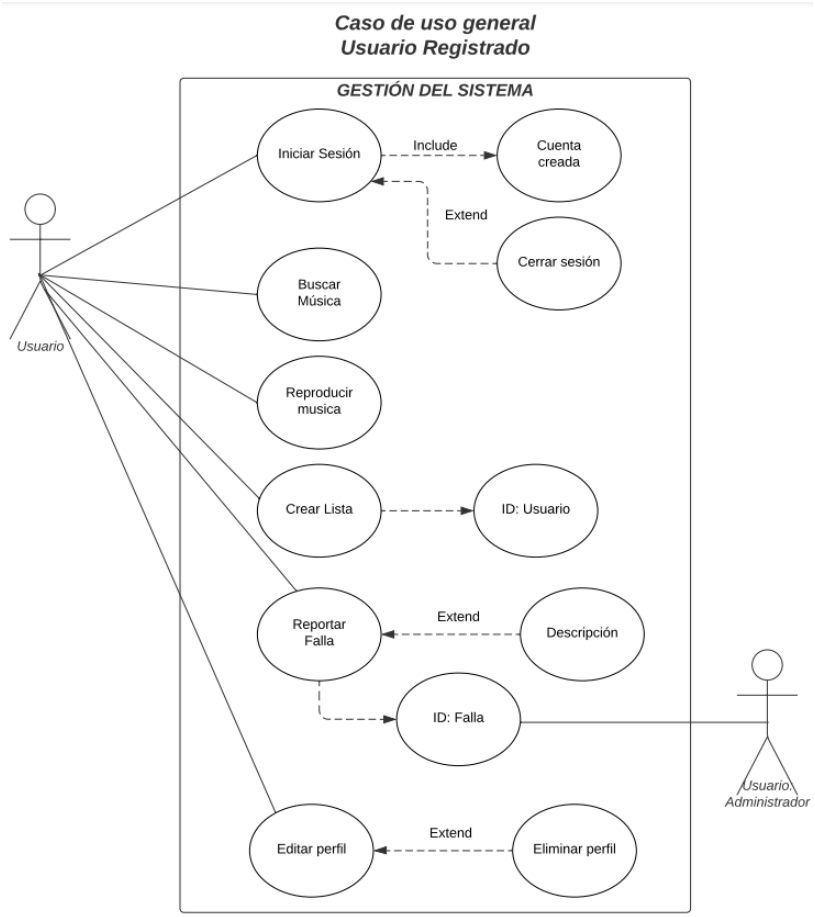

Figure 8: Behavior Diagram

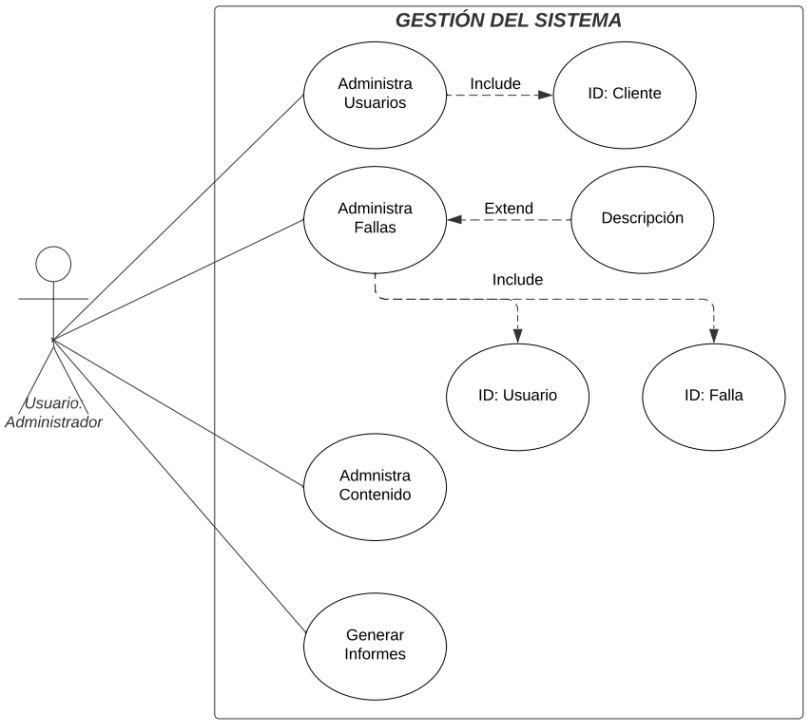

Figure 9: Use case Diagram

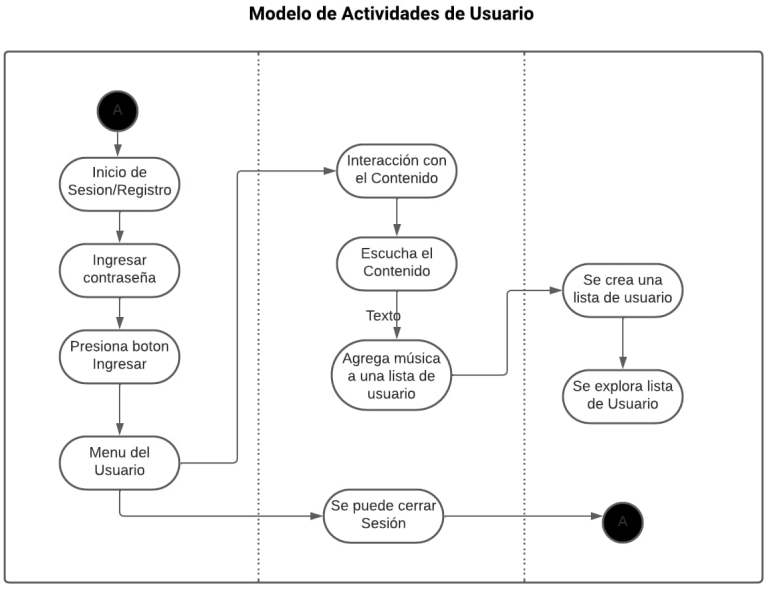

Figure 10: Use case Diagram

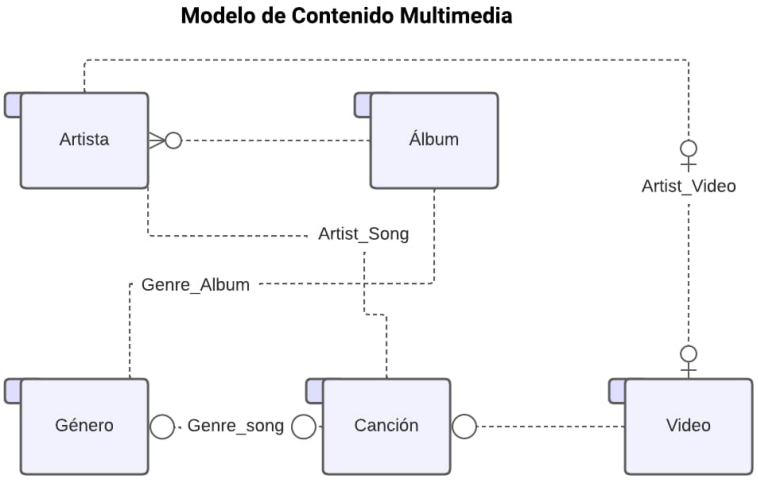

Figure 11: Activity Diagram

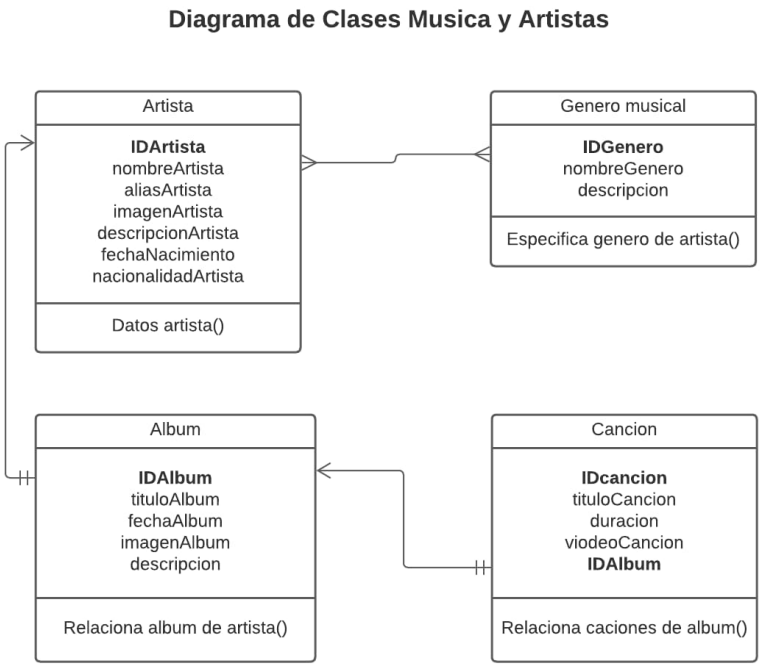

Figure 12: Content Diagram

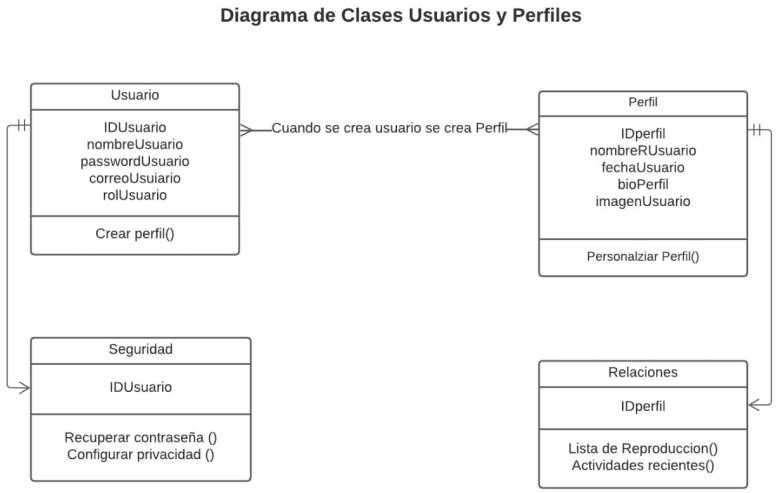

Figure 13: Class Diagram

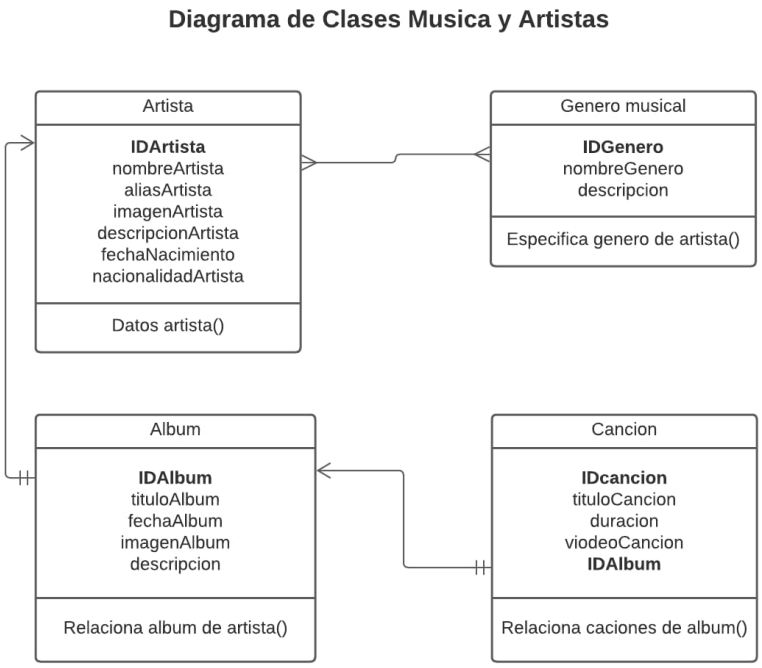

Figure 14: Class Diagram

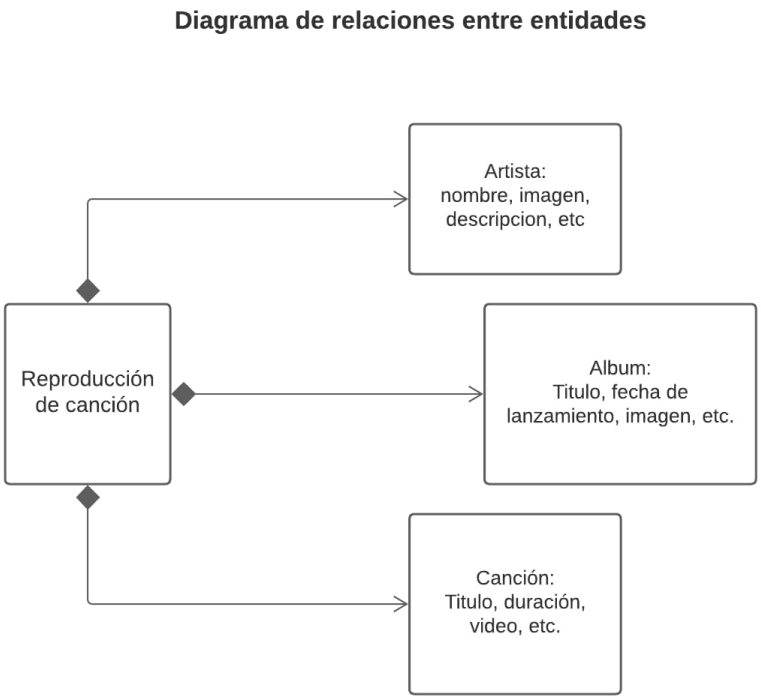

Figure 15: Entity relation Diagram

4.2.1 Flow for Creating Playlists

This subsection details the step-by-step process that users follow to create and manage playlists. It covers actions like adding songs, rearranging tracks, and sharing playlists with others.

4.2.2 Description and Priority

Here, we prioritize and provide a brief description of key use cases, emphasizing their importance and impact on the user experience.

4.2.3 Stimulus/Response Sequences

This subsection outlines the expected system responses to various user inputs or stimuli. It helps in understanding how the application behaves in different scenarios.

5 Other Nonfunctional Requirements

5.1 Usability

Focus on the user interface and overall user experience. They include aspects like intuitive navigation, responsive design, and accessibility features to ensure an enjoyable and inclusive experience for all users.

5.2 Performance

Define the system's speed and responsiveness, ensuring that music playback is smooth and seamless. This section may include metrics such as load times, buffering rates, and latency thresholds.

5.3 Scalability

Address the system's ability to handle growing user numbers and increasing data volumes. It discusses how the application will scale horizontally or vertically to accommodate higher demand.

5.4 Data Backup

How user data, playlists, and preferences will be securely backed up and restored in case of data loss or system failures. It encompasses backup frequency, redundancy measures, and disaster recovery plans.