SOFTWARE REQUIREMENTS SPECIFICATION

**Music Library Management System**

**Prepared by:-**

*BHUPATHI MN*

*BHUVANESH P*

*ERAI ANBU T*

# Introduction

## Purpose

## The purpose of a music library management system is to improve organization. The system can allow users to organize their music collections in a way that suits their needs, such as by genre, artist, or date added 1. This makes it easier for users to find the music they want to listen to and create playlists based on their preferences. Additionally, a music library management system can help users keep track of their music files and ensure that they are backed up in case of data loss.

## Document Conventions

* + - Entire document should be justified.
    - Convention for Main title

Font face: Times New Roman Font style: Bold

Font Size: 14

* + - Convention for Sub title

Font face: Times New Roman Font style: Bold

Font Size: 12

* + - Convention for body

Font face: Times New Roman Font Size: 12

## Scope of Development Project

## A music library management system is a software application that helps users organize and manage their music collections. The development of such a system involves several stages, including planning, design, implementation, testing, and maintenance 1. The scope of development for a music library management system can vary depending on the specific needs of the user. Some possible features that could be included are automatic metadata tagging, integration with music streaming services, and social sharing capabilities 2. The purpose of a music library management system is to improve organization. The system can allow users to organize their music collections in a way that suits their needs, such as by genre, artist, or date added 3. This makes it easier for users to find the music they want to listen to and create playlists based on their preferences. Additionally, a music library management system can help users keep track of their music files and ensure that they are backed up in case of data loss

## Definitions, Acronyms and Abbreviations

JAVA -> platform independence SQL-> Structured query Language ER-> Entity Relationship

UML -> Unified Modeling Language

IDE-> Integrated Development Environment SRS-> Software Requirement Specification

## References

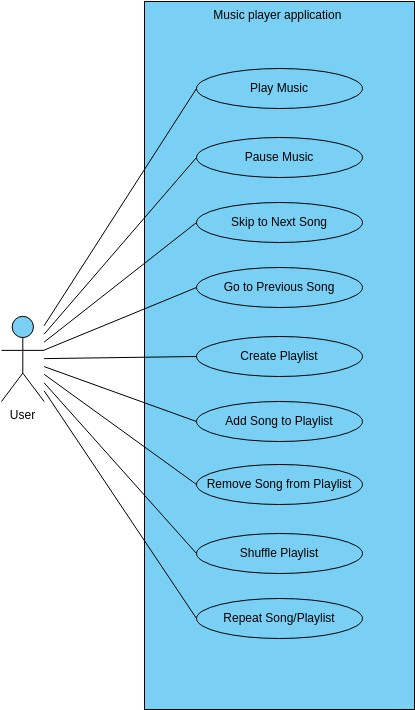
[**https://www.slideshare.net/nileshpadwal456/music-management-system**](https://www.slideshare.net/nileshpadwal456/music-management-system)

[**https://www.edureka.co/blog/library-management-system-project-in-java**](https://www.edureka.co/blog/library-management-system-project-in-java)

# Overall Descriptions

## Product Perspective

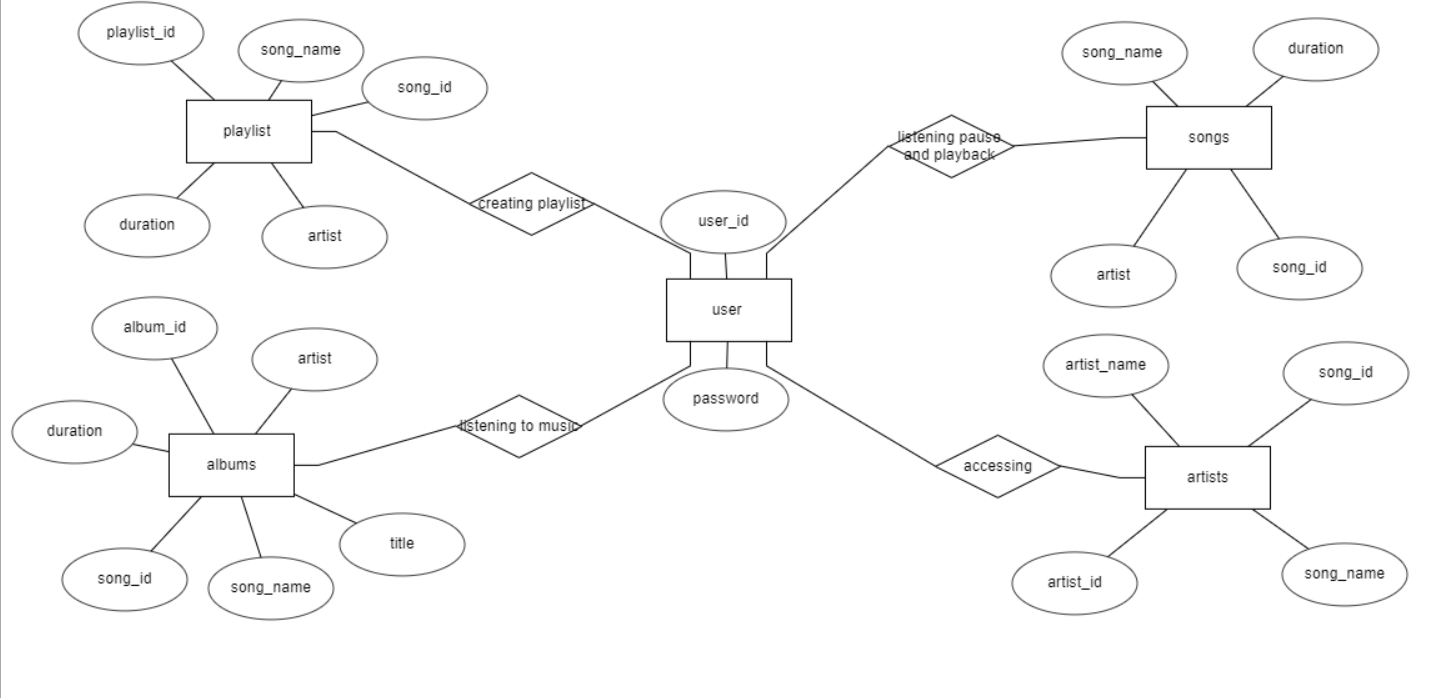
Use Case Diagram of Music Library Management System



This is a broad level diagram of the project showing a basic overview.This System will provide a search functionality to facilitate the search of resources. This search will be based on various categories such as artists or playlists. Further the playlist creator can add/update the songs. The users can create new playlists and other users can access it

## Product Function

Entity Relationship Diagram of Music Library Management System



The Music Online Library System helps to listen songs in online without downloading it and it allows users to create playlists of different vibes and interests. It groups songs based on the songs by different artists .It allows users to like songs and listen the liked songs in liked playlist and multiple users can access the playlist created by another users and can listen to the songs in it. Users can create multiple number of playlists with n number of songs in it and can listen tom it.

## User Classes and Characteristics

The system provides different types of services.

The features that are available to users are

* Users can listen any songs
* Users can create many number of playlists
* Users can add any number of songs to their prespective playlists
* User can listen to playlists created by another users also
* User can listen to songs in albums that are specified by the artists who made the songs
* But the user cannot enter any new songs into the app they can access only the existing songs
* In any new songs are to be added then it can be done by developer .

## Operating Environment

The product will be operating in windows environment. The Music Library Management System is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer, Google Chrome. The only requirement to use this online product would be the internet connection.

The hardware configuration include Hard Disk: 40 GB, Monitor: 15” Color monitor, Keyboard: 122 keys. The basic output device required is speaker and for increased performance you can add subwoofer and sound bars .

## Assumptions and Dependencies

The assumptions are:-

* + - The coding should be error free
    - The system should be user-friendly so that it is easy to use for the users
    - The information of all songs and playlists must be stored in a database that is accessible by the website
    - The system should have more storage capacity and provide fast access to the database
    - The system should provide search facility and start playing as soon as possible .
    - The Music Library System is running 24 hours a day
    - Users may access from any computer that has Internet browsing capabilities and an

Internet connection

* + - Users must have their correct usernames and passwords to login and listen songs .

The dependencies are:-

* + - The specific hardware and software due to which the product will be run
    - On the basis of listing requirements and specification the project will be developed and run
    - The end users (listener) should have proper understanding of the product
    - The system should have the general report stored
    - The information of all the songs must be stored in a database that is accessible by the users
    - Any update regarding the playlist and library is to be recorded to the database and the data entered should be correct.

## Requirement

Software Configuration:-

This software package is developed using java as front end which is supported by sun micro system. Microsoft SQL Server as the back end to store the database.

Operating System: Windows NT, windows 98, Windows XP Language: Java Runtime Environment, Net beans 7.0.1 (front end) Database: MS SQL Server (back end)

Hardware Configuration:- Processor: Pentium(R)Dual-core CPU Hard Disk: 40GB

RAM: 256 MB or more

## Data Requirement

## The input consist of entering the song name for searching or the name of the artist who sang the song or the name of the playlist . The output consists of the user receiving the song that matches the search results . and another input will be playlist name and the songs which are to be inserted into the playlist .

# External Interface Requirement

## GUI

The software provides good graphical interface for the user and the administrator can operate on the system, performing the required task such as create, update, viewing the details of the playlists.

* + - It allows user to view quick reports like existing playlists and albums .
    - It provides new songs and search facility based on different criteria.
    - All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined
    - The design should be simple and all the different interfaces should follow a standard

template

* + - The user interface should be able to interact with the user management module and a part of the interface must be dedicated to the login/logout module

Login Interface:-

In case the user is not yet registered, he can enter the details and register to create his account. Once his account is created he can ‘Login’ which asks the user to type his username and password. If the user entered either his username or password incorrectly then an error message appears.

Search:-

The user enter the name of the song or the artist of the song and search the song and can listen to it .

Categories View:-

Categories view shows the categories of playlists and albums existing in the app and the types of song present in the application.

Users Control Panel:-

This will allow the user to create add or update or delete songs in the playlist which he/she created

Or allows to create new playlist.

# System Features

The users of the system should be provided the surety that their account is secure. This is possible by providing:-

* Allow users to create new accounts with basic profile information.
* Implement secure login mechanisms for user authentication
* Enable users to update and manage their profile information.
* Provide a user-friendly interface to browse and explore the music catalog.
* Allow users to search for music tracks or albums based on title, artist, genre, etc.
* Implement filtering options to refine search results
* Display detailed information about each music track or album, including artist, genre, duration, and release date.

# Other Non-functional Requirements

## Performance Requirement

## The system should respond to user interactions (e.g., searching, browsing, playlist management) within 2 seconds for 90% of requests .Music tracks should start playing within 3 seconds after the user initiates playback The system should handle a minimum of 100 concurrent user searches or browsing sessions without degradation in response time. The system should utilize no more than 80% of available CPU and memory resources during peak usage. Minimize network latency to ensure that content (e.g., music previews) is delivered within 5 seconds. Optimize bandwidth usage for efficient data transfer, especially during music streaming, to ensure smooth playback without buffering. The system should generate error responses within 1 second, providing clear and informative error messages to users.

## Safety Requirement

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost. Proper UPS/inverter facility should be there in case of power supply failure.

## Security Requirement

## User passwords must be securely hashed and stored to prevent unauthorized access.

## Enforce a password policy, including requirements for length, complexity, and regular updates.

## Implement MFA to add an additional layer of security for user authentication.

## Implement account lockout mechanisms after a certain number of failed login attempts to prevent brute force attacks.

## Implement account lockout mechanisms after a certain number of failed login attempts to prevent brute force attacks.

## Implement secure communication using protocols like HTTPS to encrypt data transmitted between clients and servers.

## Implement session timeouts to automatically log users out after a period of inactivity.

## Requirement attributes

* + - There may be multiple admins creating the project, all of them will have the right to create changes to the system. But the members or other users cannot do changes
    - The project should be open source
    - The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
    - The user be able to easily download and install the system

## Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data.This includes the rules and regulations that the System users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor member should cross the rules and regulations.

## User Requirement

## Users should be able to create individual accounts with a unique username and a secure password.

## The system must include mechanisms for secure authentication, ensuring that only authorized users can access their accounts.

## Users should be able to search for music tracks and albums based on criteria such as title, artist, genre, and release date

## The system should provide browsing features to explore music catalog categories, new releases, and popular tracks.

## users should have the ability to create, modify, and delete personalized playlists.

## Adding and removing music tracks or albums from playlists should be intuitive and user-friendly.

## Users should have the ability to submit reviews and ratings for music tracks and albums.

## Users should be able to update their profile information, including email address, password, and personal details.

## The system must support account recovery mechanisms for users who forget their passwords.

# Other Requirements

## Data and Category Requirement

## User Data: User ID (Unique identifier),Username, Password ,First name, Last name ,Email address, Physical address, Phone number. Product Data: Track ID ,Title, Artist, Genre, Duration, Release date, Price, Stock quantity, Album ID Album Data: Album ID, Title, Artist ,Genre ,Release date ,Price, Stock quantity

,

## Appendix

A: Admin, Abbreviation, Acronym, Assumptions; B: Books, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; L: Library, Librarian; M: Member; N: Non-functional Requirement; O: Operating environment; P: Performance,Perspective,Purpose; R: Requirement, Requirement attributes; S: Safety, Scope, Security, System features; U: User, User class and characteristics, User requirement;

## Glossary

The following are the list of conventions and acronyms used in this document and the project as well:

* + - Administrator: A login id representing a user with user administration privileges to the software
    - User: A general login id assigned to most users
    - Client: Intended users for the software
    - SQL: Structured Query Language; used to retrieve information from a database
    - SQL Server: A server used to store data in an organized format
    - Layer: Represents a section of the project
    - User Interface Layer: The section of the assignment referring to what the user interacts with directly
    - Application Logic Layer: The section of the assignment referring to the Web Server. This is where all computations are completed
    - Data Storage Layer: The section of the assignment referring to where all data is recorded
    - Use Case: A broad level diagram of the project showing a basic overview
    - Class diagram: It is a type of static structure diagram that describes the structure of a system by showing the system’s cases, their attributes, and the relationships between the classes
    - Interface: Something used to communicate across different mediums
    - Unique Key: Used to differentiate entries in a database

## Class Diagram

A class is an abstract, user-defined description of a type of data. It identifies the attributes of the data and the operations that can be performed on instances (i.e. objects) of the data. A class of data has a name, a set of attributes that describes its characteristics, and a set of operations that can be performed on the objects of that class. The classes’ structure and their relationships to each other frozen in time represent the static model. In this project there are certain main classes

which are related to other classes required for their working. There are different kinds of relationships between the classes as shown in the diagram like normal association, aggregation, and generalization. The relationships are depicted using a role name and multiplicities. Here ‘Librarian’, ‘Member’ and ‘Books’ are the most important classes which are related to other classes.

