SOFTWARE REQUIREMENTS SPECIFICATION

**For**

**Music**

**Library Management System**

**Prepared by:-**

Panda’s Crew

# Introduction

## Purpose

## The purpose of this project is to develop a comprehensive system that enables users to efficiently organize, catalog, and play their music collection. The system aims to enhance the overall music listening experience by providing robust features for music organization, playback, and playlist creation. This music organization and playback system aim to provide users with a comprehensive and enjoyable music listening experience. By focusing on access control, diverse feature sets, and responsive design, the system aims to cater to the needs of a wide range of users, making it a versatile and user-centric solution. Regular updates and user feedback integration will contribute to the ongoing improvement and refinement of the system.

## 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

## The scope of the development project encompasses the creation of a robust and user-centric music organization and playback system, offering a comprehensive solution for music enthusiasts to manage their collections, create playlists, and enjoy seamless playback experiences. The project will involve the design and implementation of a sophisticated user interface across multiple platforms, ensuring accessibility through web, mobile, and desktop interfaces. Key features include secure access control mechanisms, with varying user roles and authentication protocols to protect user data. The system will support music playback and streaming in various audio formats, providing a versatile solution for different music sources. Advanced cataloging and organization features, such as metadata tagging, search functionalities, and user-generated ratings and reviews, will enhance the overall user experience.

## 

## Playlist creation and management will be a focal point, with intuitive drag-and-drop capabilities, collaborative playlist sharing, and smart playlist generation based on user preferences. Additionally, the system will integrate seamlessly with external music streaming services, offering an extended music library and enabling users to discover new music effortlessly. Regular user feedback loops and updates will be integral to the project, ensuring continuous improvement and alignment with user needs. The development scope is not limited to online functionality; it will also include an offline mode, allowing users to download music for offline playback and ensuring a seamless experience regardless of internet connectivity. Overall, the project aims to deliver a feature-rich, intuitive, and versatile music management solution that caters to a diverse audience of music enthusiasts

## 1.4 References

* + - Books
* Designing Mobile Interfaces: Patterns for Interaction Design" by Steven Hoober

* "User Interface Design and Evaluation" by Debbie Stone, Caroline Jarrett, Mark Woodroffe, and Shailey Minocha
* Designing Interfaces by Jenifer Tidwell
*  Websites

<https://uxdesign.cc/>

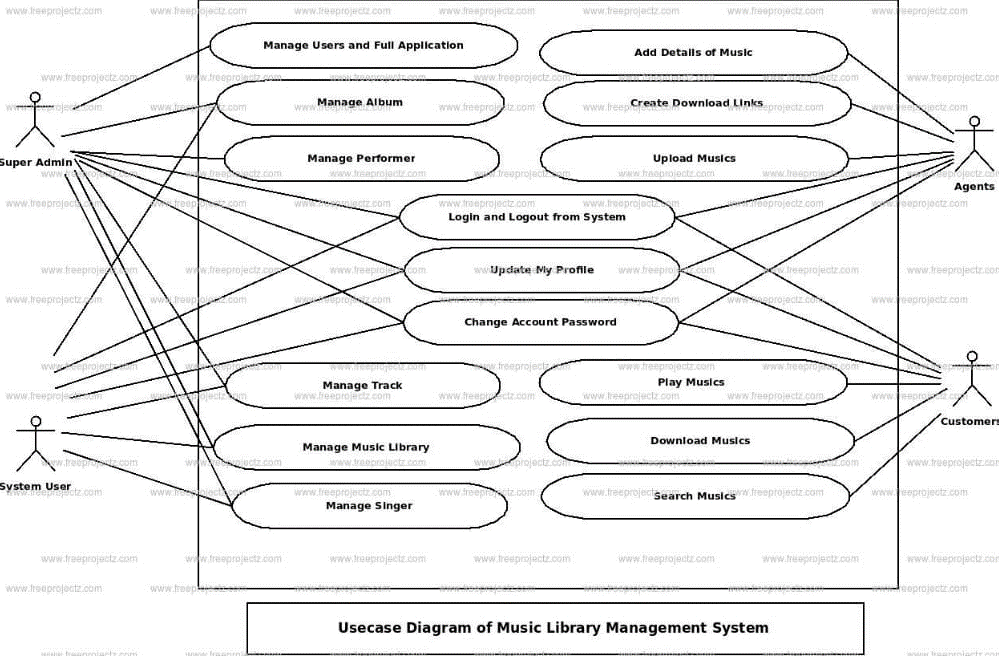
<https://www.smashingmagazine.com/>

# Overall Descriptions

# 2.1 Product Perspective

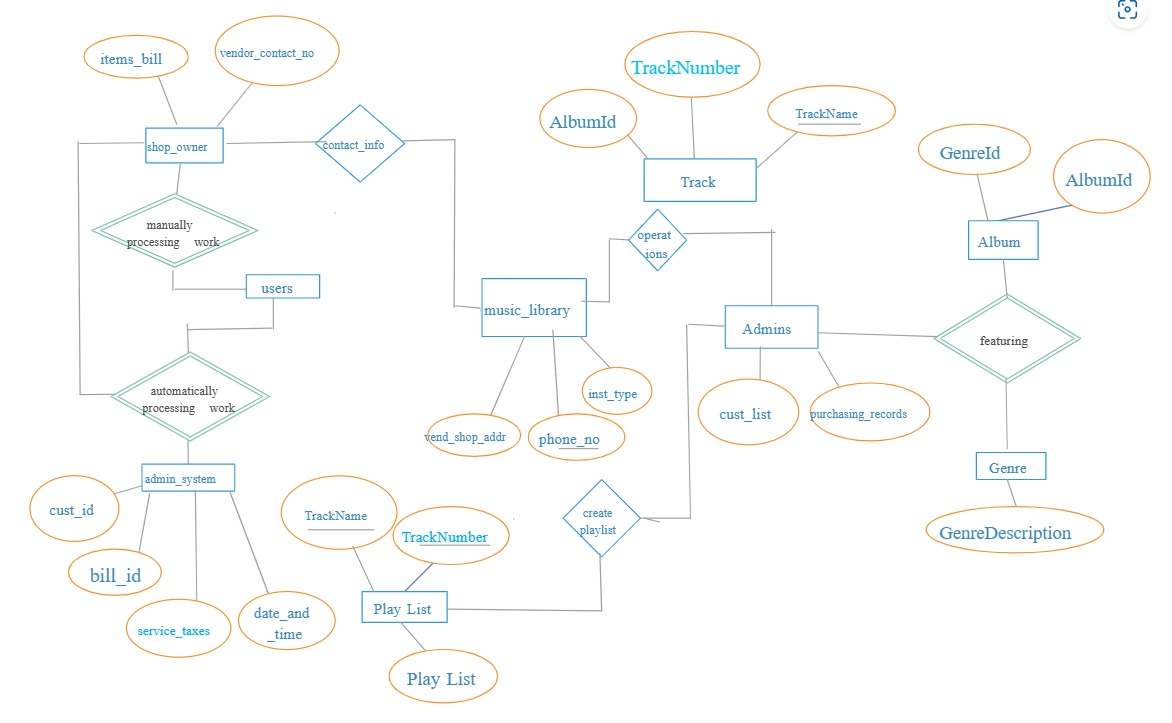
Use Case Diagram of Music Libraury Management System

1



## Product Function

Entity Relationship Diagram of Music Library Management System



The Music Library Management System offers real-time online access to comprehensive information about the available music collections and user data. The primary objective of this project is to streamline and automate manual tasks associated with managing music resources. The software efficiently handles functions such as Music Checkouts, Returns, Fine Calculation and Management, and Report Generation tailored to meet specific end-user needs.This project aims to enhance the efficiency of music library operations by providing a centralized and automated platform for managing music assets and member activities

The Music Management System is a web-based application that allows users to organize and manage their music library. The system will support the upload, storage, and retrieval of music tracks, as well as the creation and management of playlists.

Features of music management system:

* Secure login mechanisms with different user roles (e.g., administrator, regular user) to control access.
* Ability to add and organize music tracks with metadata (artist, album, genre, release year).
* Advanced search options for quick access to specific tracks or albums.
* Filters based on criteria such as artist, genre, and release date.
* User-friendly interface for creating, editing, and managing playlists.
* Smart playlist generation based on user preferences or criteria.
* Smooth music playback with options for repeat and shuffle.
* Integration with music streaming services or APIs for an extended music library.
* Tracking of listening history.
* Automated recommendations based on user preferences.

## Operating Environment

The product will be operating in windows environment. The Library Management System is a website and shall operate in all famous browsers, for a model we are taking Microsoft Internet Explorer,Google Chrome,and Mozilla Firefox. Also it will be compatible with the IE 6.0. Most of the features will be compatible with the Mozilla Firefox & Opera 7.0 or higher version. The only requirement to use this online product would be the internet connection.This system should ideally operate seamlessly across different devices, including web browsers for desktops, mobile applications for smartphones and tablets (both iOS and Android), and potentially dedicated desktop applications for a comprehensive user experience. Cross-browser compatibility is essential to guarantee accessibility across popular web browsers. The system should integrate with diverse music platforms and streaming services to offer an extensive music library. Furthermore, it should be compatible with major operating systems such as Windows, macOS, and Linux, catering to a broad user base.

## 2.3Assumptions and Dependencies

The assumptions are:-

* The system assumes that users have regular access to the internet for streaming music and accessing online features.
* It is assumed that users will access the system through devices that support modern web browsers, ensuring compatibility and optimal performance.
* Assumption that the system will integrate with external music databases or services to enrich its catalog and provide accurate metadata for tracks.
* Assumes that users are familiar with basic music terms and digital interfaces for effective navigation and use of the system.
* Assumes that the external music databases provide accurate and up-to-date information to maintain the system's catalog integrity.

The dependencies are:-

* Dependency on third-party APIs for music streaming services and external databases for real-time music updates and recommendations.
* Dependencies on specific programming languages and frameworks for system development, which may include technologies for front-end, back-end, and database management.
* Dependency on robust security protocols and encryption mechanisms to ensure the protection of user data and privacy.
* Dependency on the compatibility of the system with various operating systems, including Windows, macOS, and Linux, to reach a diverse user base.
* Dependency on ensuring that the system functions correctly across different web browsers (e.g., Chrome, Firefox, Safari) to guarantee a consistent user experience.
* Dependency on a stable and reliable network infrastructure to support music streaming and data synchronization between the system and external services.

## 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 data requirements for a Music Management System encompass a comprehensive set of information necessary for effective cataloging, organization, and user interaction. These include detailed metadata for each music track, such as artist names, album titles, genres, release years, and track durations. Additionally, the system requires user data, including account information, preferences, and listening history, to offer personalized recommendations and track usage patterns. To enrich the music catalog, dependencies on external databases or APIs should be considered, requiring data on music titles, artists, and related information. The system also needs to store data on user-generated content, such as playlists and reviews, facilitating collaborative and interactive features.

# 3.External Interface Requirement

## 3.1 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 book.

* + - It provides stock verification and search facility based on different criteria.
    - The user interface must be customizable by the administrator
    - 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:**

For users who haven't registered yet, they can input their details to create an account. Once registered, they can log in by entering their username and password. In case of incorrect credentials, an error message will prompt the user to re-enter the correct information.

**Search:**

Both members and administrators can search for specific music items by entering keywords such as genre, artist, or title. This feature facilitates easy discovery of the desired music within the system.

**Genres View:**

The Genres View section displays available music genres, allowing administrators to add, edit, or remove genres from the list. This feature provides a comprehensive overview of the music collection's categorization.

**Administrator's Control Panel:**

The Administrator's Control Panel empowers administrators to manage users, add, edit, or remove music resources, and oversee lending options. This central hub facilitates efficient control and organization of the music library.

# System Features

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

* User authentication and validation of members using their unique member ID
* Proper monitoring by the administrator which includes updating account status, showing a popup if the member attempts to issue number of books that exceed the limit provided by the library policy, assigning fine to members who skip the date of return
* Proper accountability which includes not allowing a member to see other member’s account. Only administrator will see and manage all member accounts

# Other Non-functional Requirements

## Performance Requirement

The proposed system that we are going to develop will be used as the Chief performance system within the different campuses of the university which interacts with the university staff and students. Therefore, it is expected that the database would perform functionally all the requirements that are specified by the university.

* + - The performance of the system should be fast and accurate
    - Library Management System shall handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus it should have inbuilt error testing to identify invalid username/password
    - The system should be able to handle large amount of data. Thus it should accommodate high number of books and users without any fault

## 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

* + - System will use secured database
    - Normal users can just read information but they cannot edit or modify anything except their personal and some other information.
    - System will have different types of users and every user has access constraints
    - Proper user authentication should be provided
    - No one should be able to hack users’ password
    - There should be separate accounts for admin and members such that no member can access the database and only admin has the rights to update the database.

## 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

User requirements for a Music Management System involve ensuring a seamless and user-friendly experience for both members and administrators. Members should be able to easily register, navigate a straightforward and intuitive interface, search and explore the music catalog, and effortlessly create and manage playlists. Secure account management, along with the ability to update preferences, is crucial. Administrators, on the other hand, need tools for efficient catalog management, including backup and recovery options, data migration, and replication functionalities. Auto-recovery mechanisms, organized file structures, and regular server maintenance with updates are essential for system reliability.

The admin provides certain facilities to the users in the form of:-

* + - Backup and Recovery
    - Forgot Password
    - Data migration i.e. whenever user registers for the first time then the data is stored in the server
    - Data replication i.e. if the data is lost in one branch, it is still stored with the server
    - Auto Recovery i.e. frequently auto saving the information
    - Maintaining files i.e. File Organization
    - The server must be maintained regularly and it has to be updated from time to time

# 5.Other Requirements

## Data and Category Requirement

## 

## Administrators possess comprehensive rights, enabling them to modify, delete, and append data, while all other users, excluding librarians, have limited access for information retrieval from the database. Similarly, diverse categories of music should be meticulously coded, ensuring the relevant data for each category is accurately displayed. The system must adhere to a specific format to effectively organize and present information based on music categories. This categorization is crucial for streamlined access and retrieval of music data aligned with user preferences and requirements.

## Appendix

A: Admin, Abbreviation, Acronym, Assumptions; M:Music, Business rules; C: Class, Client, Conventions; D: Data requirement, Dependencies; G: GUI; K: Key; P: Playlist, 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.

