**Software Requirement Specification**

For

**Music Streaming Service**

Prepared by

Anik Biswas 12021002016050 [biswasanik159@gmail.com](mailto:biswasanik159@gmail.com)

Instructor: Prof. Bipasha Mahato

Prof.

**Course: Database Management System**

**Contents**

1. **Introduction …………………………………………Page 3**

* **Document Purpose ……………………………Page 3**
* **Product Scope………………………………….Page 3**
* **Document Conventions………………………..Page 3**
* **References And Acknowledgments…………..Page 3**
* **Overview………………………………………..Page 3**
* **Audience………………………………………..Page 4**

1. **Specific Requirements………………………………Page 5**

* **External Interface Requirements…………….Page 5**
* **Functional Requirements……………………..Page 6**

1. **Data Flow Diagrams………………………………...Page 9**

* **Level-0-DFD……………………………………Page 9**
* **Level-1-DFD……………………………………Page 10**
* **Level-2-DFD……………………………………Page 11**

1. **Conclusion…………………………………………….Page 12**

**Introduction**

The Online Music System is a dynamic web application designed to provide users with an immersive and interactive music listening experience. In this digital age, music has become an integral part of people's lives, and our application aims to connect music enthusiasts with their favorite tunes seamlessly. This application offers user-friendly features such as user authentication for secure access, an extensive catalog of songs, the ability to create and manage personalized playlists, and convenient song playback controls.

**Product Purpose:**

The project's purpose is to develop an online music platform, offering users a secure, personalized, and immersive music experience. It will feature user authentication, extensive song catalogs, playlist management, and intuitive playback controls while ensuring cross-platform accessibility and compliance with music content copyright and licensing requirements.

**Product Scope:**

The product scope for the Online Music System comprises user authentication, a diverse song catalog, playlist management, and playback controls. Users can create and customize playlists, play, pause, skip, and view song details. The system emphasizes performance, security, and cross-platform compatibility while adhering to copyright and licensing standards for music content.

**Document Conventions:**

1. **User-Centric Design:** The product will prioritize user-friendly interfaces, intuitive navigation, and an engaging user experience to enhance user satisfaction and encourage active participation.
2. **Scalable Architecture:** The product's technical design will allow for easy scalability to accommodate an expanding user base and evolving feature requirements, ensuring long-term sustainability.

**Overview:**

The "Online Music System" is an ambitious web application project aimed at transforming the way users interact with music in a digital environment. This project seeks to create a dynamic platform that empowers music enthusiasts to access, manage, and enjoy their favorite songs with convenience and personalization. Key features include:

**Functional Requirements**:

1. **User Authentication:** The system must allow users to register accounts with a valid email address and password.
2. **Song Pages:** The application must display a list of available songs with details such as title, artist, album, and cover art.
3. **Song Playback:** The system must provide playback controls for playing, pausing, skipping to the next song, and going back to the previous song.

**Non-functional Requirements**:

1. The system must ensure performance optimization, with responsive and fast data retrieval.
2. The SQL database must be designed for efficient data retrieval and storage.
3. The application must be responsive and work seamlessly on various devices, screen sizes, and modern web browsers.

**Constraints**: 1

1. The system must adhere to copyright and licensing agreements for music content. It should only offer music that is legally acquired and distributed.
2. Dependencies on external services, such as third-party music data providers or email services, may introduce constraints related to their availability and reliability.

**Assumptions and Dependencies**:

1. Users will have access to the internet and modern web browsers to use the application.
2. Users will have a basic understanding of how to navigate and interact with web applications.
3. The application depends on third-party music services to source and retrieve song information.
4. The application relies on the availability and performance of the SQL database management system.

**Audience:**

The primary and secondary audiences for the Online Music System are as follows:

**Primary Audience:** This includes music enthusiasts and listeners who will use the application to access, play, and manage their music. They engage with the user interface, create playlists, and customize their profiles.

**Developers and IT Teams:** Those responsible for developing, maintaining, and deploying the application, including frontend and backend developers, quality assurance testers, and system administrators.

**Project Stakeholders:** This group comprises project managers, product owners, and decision-makers who oversee the project's progress and ensure it aligns with strategic objectives and timelines.

**Specific Requirements**  
To provide more specific requirements for the Online Music System, I'll outline some detailed functional and non-functional requirements:

**External Interface Requirements:**

In this section of the Software Requirements Specification (SRS) document for the Online Music System Application, we will outline the external interfaces that the application will interact with. These interfaces include external systems, APIs, databases, and hardware components that are integral to the functionality of the application.

**1** **User Interface (UI):**

The user interface is the primary external interface through which users interact with the application.

**Requirements:**

1.The UI must be responsive, ensuring a consistent and user-friendly experience on various devices and screen sizes.

2. It should include intuitive navigation, user-friendly forms, and interactive controls for music playback and playlist management.

**API Endpoints:**

The system must expose a set of API endpoints to facilitate communication between the frontend and backend components.

1. API endpoints must adhere to RESTful principles for consistency and ease of use.
2. Endpoints should be documented, providing details on request parameters, response formats, and usage examples.

**Database Management System (DBMS):**

The application interfaces with an SQL database system to store and retrieve user data, song information, and playlist details. **MySQL**: An open-source relational database management system.

1. The database system must be capable of handling concurrent connections and large data volumes efficiently.
2. Data consistency and integrity should be maintained through appropriate database transactions and constraints.

**Third-Party Music Services:**

The application may interface with external music services or databases to retrieve song information.

1. Integration with third-party services must be secure and reliable, ensuring the availability and accuracy of song data.
2. Data synchronization mechanisms should be in place to keep the music catalog up-to-date.

**Email Service:**

The application may use external email services for sending verification and password reset emails to users.

1. Integration with the email service must support email delivery and verification processes securely.
2. It should provide mechanisms for tracking email delivery and handling bounced emails.

**Hardware Interfaces:**

The application does not have specific hardware interfaces as it primarily operates in a web-based environment. However, it may utilize standard hardware components such as servers, network infrastructure, and storage devices to host and maintain the application.

**Software Interfaces:**

The Online Music System Application may integrate with other software systems and libraries to enhance its functionality. These software interfaces may include:

**Operating Systems**: The application should be compatible with various operating systems, including Windows, Linux, and macOS.

**Web Browsers**: The GUI should function correctly on popular web browsers such as Chrome, Firefox, Safari, and Edge.

**Programming Languages and Frameworks**: The application may be built using programming languages such as JavaScript, Python, and frameworks like express.

**Functional Requirements:**

Certainly, here are some more detailed functional requirements for the Online Music System:

**User Authentication:**

1. Users should receive a session token upon successful login for secure access.
2. **User Login:** Registered users must be able to log in using their email and password.

Users should receive a session token upon successful login for secure access.

1. **Password Reset:**

Users must have the option to reset their password by providing a registered email address.

A password reset link or code should be sent to the user's email.

**Song Pages:**

1. **Song Catalog Display:** The application should present a paginated list of songs, with each page containing a specified number of songs.
2. **Song Sorting and Filtering:** Users should be able to sort songs by various criteria, such as title or artist.

**Playlist Management:**

1. **Create Playlists:** Users must be able to create new playlists by specifying a name and, optionally, uploading a cover image.
2. **Add and Remove Songs:** Users should be able to add songs to their playlists from the song catalog.

**Song Playback:**

1. **Playback Controls:** The system must provide user-friendly playback controls, including play, pause, next, and previous buttons for smooth song playback.
2. **Display Song Details:** When a song is playing, its details (title, artist, album) should be displayed for the user.

**Non-Functional Requirements:**  
Here are some non-functional requirements for the Online Music System:

**Performance:**

The system must respond to user interactions within one second to provide a seamless and engaging user experience.

The application must handle a minimum of 10,000 concurrent users without significant performance degradation.

It should be designed to scale horizontally to accommodate future user growth.

**Security:**

User passwords must be securely hashed and salted before storage in the database.

User authentication and data transfer must be protected using industry-standard encryption protocols.

The application should employ measures to protect user data from unauthorized access or data breaches.

Regular security assessments and penetration testing should be conducted to identify and address vulnerabilities.

**Database:**

The database should be optimized for read-heavy operations, allowing efficient retrieval of song and playlist data.

Data integrity constraints should be enforced at the database level.

**Cross-Platform Compatibility:**

The user interface must be responsive and functional on various devices, including smartphones, tablets, and desktop computers.

The application must be tested and work seamlessly on popular web browsers, including Chrome, Firefox, Safari, and Edge.

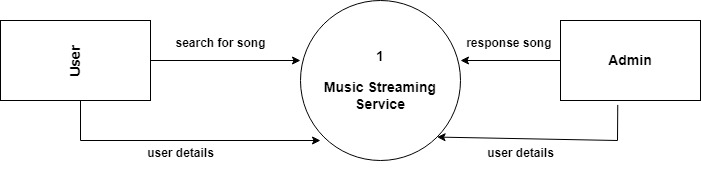
**Error Handling:**

Appropriate error messages and codes must be provided to assist users and system administrators in troubleshooting issues.

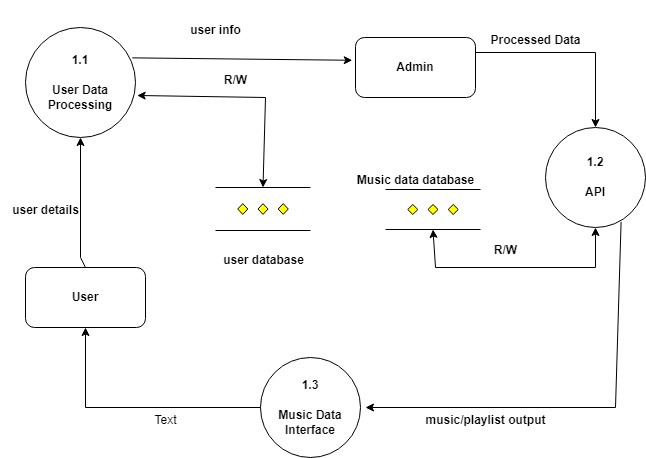
**Data Flow Diagrams**

DFD stands for "Data Flow Diagram." It is a visual representation used in software engineering and systems analysis to illustrate the flow of data within a system or process. DFDs are commonly used to model and understand how data moves through various components and processes in a system, helping in the analysis, design, and documentation of information systems.

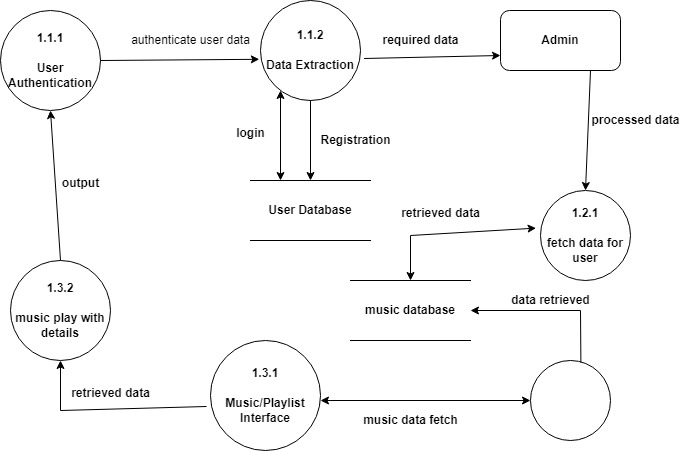
**Level-0-DFD:**

****

**Level-1-DFD:**

****

**Level-2-DFD:**

****

**Conclusion:**

In conclusion, the Online Music System project presents a comprehensive solution for enhancing the digital music listening experience. With a robust set of functional and non-functional requirements, the system is poised to offer users a secure, personalized, and responsive platform for exploring, managing, and enjoying their favorite songs. Key features, such as user authentication, an extensive song catalog, playlist management, and intuitive playback controls, are designed to meet the demands of music enthusiasts.

Furthermore, stringent non-functional requirements emphasize the importance of performance, security, compliance, and usability, ensuring that the system not only functions flawlessly but also respects legal and privacy considerations. The project's success depends on a user-centric approach and a scalable technical architecture, which collectively strive to deliver an engaging and immersive music experience.

As the Online Music System project moves forward, it holds the potential to transform how users engage with music in the digital realm, providing a seamless and customizable experience while adhering to the highest standards of security and performance.

Top of Form