

---

# Software Requirements Specification

for

## SongSphere

Version 1.0

Prepared by

**Group Name:** SongSphere

Meet Daftary	60004210022
Param Shah	60004210046
Deep Gada	60004210049
Ansh Bhatt	60004210061

**Instructor:** *<place your instructor's name here>*

**Course:** *<place your course name here>*

**Lab Section:** *<place your lab section here>*

**Teaching Assistant:** *<place your TA's name here>*

**Date:** <place the date of submission here>

# Contents

<b>CONTENTS .....</b>	<b>ii</b>
<b>REVISIONS .....</b>	<b>iii</b>
<b>1. INTRODUCTION.....</b>	<b>4</b>
<b>1.1. DOCUMENT PURPOSE .....</b>	<b>4</b>
<b>1.2. PRODUCT SCOPE .....</b>	<b>4</b>
SONGSPHERE IS A FREEMIUM STREAMING PLATFORM OFFERING MILLIONS OF SONGS, PODCASTS, AND AUDIOBOOKS. WITH A FREE ACCOUNT, YOU CAN ACCESS A VAST LIBRARY WITH ADS AND LIMITED CONTROL. A PREMIUM SUBSCRIPTION UNLOCKS AD-FREE LISTENING, OFFLINE DOWNLOADS, HIGHER QUALITY AUDIO, AND EXCLUSIVE FEATURES. THE PRIMARY PURPOSE OF SONGSPHERE IS TO MAKE IT EASY FOR CONSUMERS TO ACCESS AND SHARE DIGITAL CONTENT WITH EACH OTHER. THIS ENCOURAGES PEOPLE TO SUPPORT THEIR FAVORITE ARTISTS, RESULTING IN GREATER ENGAGEMENT WITH THEIR MUSIC. IT ALSO AIMS TO REVOLUTIONIZE MUSIC AND AUDIO CONSUMPTION BY PROVIDING A LEGAL, ACCESSIBLE, AND PERSONALIZED PLATFORM FOR DISCOVERY AND ENJOYMENT. ....	4
<b>1.3. INTENDED AUDIENCE AND DOCUMENT OVERVIEW .....</b>	<b>4</b>
<b>1.4. DEFINITIONS, ACRONYMS AND ABBREVIATIONS .....</b>	<b>5</b>
<b>1.5. DOCUMENT CONVENTIONS .....</b>	<b>5</b>
<b>1.6. REFERENCES AND ACKNOWLEDGMENTS .....</b>	<b>5</b>
<b>2. OVERALL DESCRIPTION .....</b>	<b>6</b>
<b>2.1. PRODUCT PERSPECTIVE.....</b>	<b>6</b>
<b>2.2. PRODUCT FUNCTIONALITY .....</b>	<b>6</b>
<b>2.3. USERS AND CHARACTERISTICS .....</b>	<b>7</b>
<b>2.4. OPERATING ENVIRONMENT .....</b>	<b>7</b>
<b>2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS .....</b>	<b>8</b>
<b>2.6. USER DOCUMENTATION .....</b>	<b>9</b>
<b>2.7. ASSUMPTIONS AND DEPENDENCIES .....</b>	<b>9</b>
<b>3. SPECIFIC REQUIREMENTS .....</b>	<b>11</b>
<b>3.1. EXTERNAL INTERFACE REQUIREMENTS .....</b>	<b>11</b>
<b>3.2. FUNCTIONAL REQUIREMENTS .....</b>	<b>14</b>
<b>3.3. BEHAVIOUR REQUIREMENTS .....</b>	<b>15</b>
<b>4. OTHER NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>17</b>
<b>4.1. PERFORMANCE REQUIREMENTS .....</b>	<b>17</b>
<b>4.2. SAFETY AND SECURITY REQUIREMENTS .....</b>	<b>17</b>
<b>4.3. SOFTWARE QUALITY ATTRIBUTES .....</b>	<b>18</b>
<b>5. OTHER REQUIREMENTS .....</b>	<b>19</b>
<b>APPENDIX A – DATA DICTIONARY .....</b>	<b>19</b>
<b>APPENDIX B - GROUP LOG.....</b>	<b>19</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Meet Daftary Param Shah Ansh Bhatt Deep Gada	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00



Academic Year: 2021\_22

# 1. Introduction

## 1.1. Document Purpose

This document outlines the software requirements of SongSphere, version 1.0. The document highlights the requirements of SongSphere application. In simpler terms, it tells the developers what features and functions the software must have to make sure it works efficiently.

The scope of the document covers the entire SongSphere System, including its various features like music streaming, playlist creation, premium users, free users and provides details about the user interaction. It is a guide that ensures everyone involved in building and maintaining the software application understand the goals. This is a roadmap that helps the development team improve the application to meet the demands of the users.

## 1.2. Product Scope

SongSphere is a freemium streaming platform offering millions of songs, podcasts, and audiobooks. With a free account, you can access a vast library with ads and limited control. A premium subscription unlocks ad-free listening, offline downloads, higher quality audio, and exclusive features. The primary purpose of SongSphere is to make it easy for consumers to access and share digital content with each other. This encourages people to support their favorite artists, resulting in greater engagement with their music. It also aims to revolutionize music and audio consumption by providing a legal, accessible, and personalized platform for discovery and enjoyment.

SongSphere strives to be your go-to audio universe, offering a wealth of benefits, objectives, and goals. For users, it unlocks a massive library of music, podcasts, and audiobooks, personalized recommendations for discovery, and seamless listening across devices. SongSphere's objectives paint a broader picture. They aim to expand their user base globally, convert more users to premium subscriptions, and solidify their position as the leading platform for music and audio discovery. Ultimately, SongSphere's goals revolve around providing an unparalleled user experience. They continuously innovate with new features, build strong relationships with artists and creators, and strive for a positive social and environmental impact.

## 1.3. Intended Audience and Document Overview

The Software Requirements Specification (SRS) for the SongSphere is tailored to serve the large number of readers involved in the project. Developers would find detailed technical specifications, system architecture, and functionality requirements in sections such as "Functional Requirements" and "System Architecture." Project managers, on the other hand, would focus on high-level project scope, timelines, resource allocation, and risk management outlined in sections like "Project Scope" and "Timeline." Marketing staff would seek insights into the features, target audience, and competitive analysis provided in sections such as "Product Features" and "Competitive Analysis." Users would be interested in understanding the user interface design, user stories, and overall user experience presented in sections like "User Interface Design" and "User Experience."



### Academic Year: 2021\_22

We suggest beginning with the overview sections that provide a holistic understanding of the project's objectives and scope. Then we allow readers to dive into sections most pertinent to their roles, such as developers focusing on technical specifications and architecture, while project managers prioritize timelines and resource allocation. Functional requirements, system architecture, and user interface design follow, providing detailed insights into the system's functionalities, underlying architecture, and user interactions. Testing requirements ensure the quality and reliability of the application, with other relevant sections such as competitive analysis and marketing strategies rounding out the comprehensive SRS. By following this sequence, each reader can efficiently navigate through the SRS document, focusing on the sections most relevant to their role and interests in the SongSphere project.

## 1.4. Definitions, Acronyms and Abbreviations

API: Application Program Interface  
RAM: Random Access Memory  
GDPR: General Data Protection Regulation.  
TLS: Transport Layer Security  
UDP: User Datagram Protocol  
TCP: Transmission Control Protocol  
GUI: Graphical User Interface.  
HTTP: Hypertext Transfer Protocol.  
HTTPS: Hypertext Transfer Protocol Secure.  
iOS: iPhone Operating System.  
MacOS: Macintosh Operating System.  
SDK: Software Development Kit  
OAuth: Open Authorization

## 1.5. Document Conventions

In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text is be single spaced and maintain the 1" margins found in this template. For Section and Subsection titles please follow the template.

## 1.6. References and Acknowledgments

[1] J. Seo et al. Software Requirements Document for OneMusic <https://seniord.cs.iastate.edu/2020-May-02/files/inline-files/SRS%20Template.docx.pdf>

[2] D. Vicory et al. Formatting improvements, modifications to user interface, more possible use cases added [https://capstone.cs.ucsb.edu/team\\_docs\\_14/SRS/SRS\\_NP\\_Compete.pdf](https://capstone.cs.ucsb.edu/team_docs_14/SRS/SRS_NP_Compete.pdf)

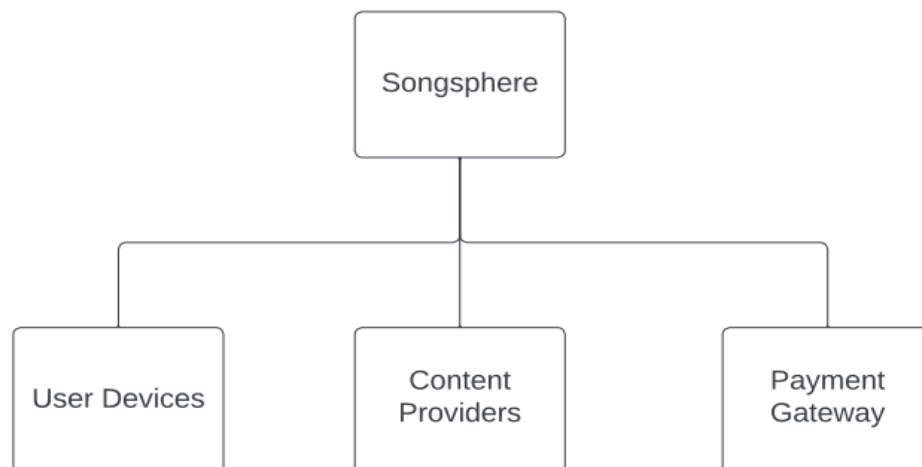


Academic Year: 2021\_22

## 2. Overall Description

### 2.1. Product Perspective

The product under the case study is SongSphere, a digital music, podcast, and video service designed to revolutionise the way users consume and interact with music and audio content. SongSphere is a new, self-contained product in the digital entertainment industry, aimed at providing users with a comprehensive platform for accessing millions of songs, podcasts, and other audiovisual content. It is not a replacement for existing systems but rather a fresh entry into the market, offering innovative features such as personalised playlists, offline listening, and social sharing capabilities. SongSphere serves as a bridge between creators and consumers, empowering artists to share their creations with a global audience while providing users with a seamless and enriching music discovery experience. how your product interacts with the environment and in what context it is being used.



In this diagram, SongSphere acts as the central hub connecting users with content providers and facilitating transactions through a payment gateway. Users interact with SongSphere through various devices, such as smartphones, tablets, and computers, to access music, podcasts, and other audiovisual content. Content providers supply their creations to SongSphere, which then distributes them to users based on their preferences and interactions with the platform. The payment gateway enables users to upgrade to premium plans or make other purchases within the app. This depiction highlights SongSphere's role in the digital entertainment ecosystem and its integration with external components to deliver a seamless user experience.

### 2.2. Product Functionality-

The major functions that the SongSphere system must perform or allow users to perform:

- Access a vast library of songs, podcasts, and other audiovisual content.



**Academic Year: 2021\_22**

- Create and share personalised playlists.
- Discover new music and podcasts based on user preferences.
- Enable offline listening by downloading content for playback without an internet connection.
- Facilitate seamless transition between devices through cross-device syncing.
- Provide curated playlists and recommendations tailored to user preferences.
- Support collaborative playlist creation, allowing multiple users to contribute and edit playlists.
- Offer detailed artist pages with information, discography, and related content.
- Enable users to search and explore the library with robust search functionality.
- Provide options for both free, ad-supported accounts and premium subscription plans with additional features like ad-free listening and higher audio quality.

## 2.3. Users and Characteristics

- **Casual listeners:-**  
Occasional users who enjoy listening to music. These users look forward to using an application with a simple and a user friendly interface that provides easy playlist creation and music discovery features.
- **Premium Subscribers:-**  
Users who opt for an ad free experience with some additional benefits like offline listening, high quality audio streaming along with some content that is exclusive for the premium users.
- **Creators and Artists:-**  
This category includes musicians, podcasters, and other content creators who use SongSphere as a streaming platform to share their work with the world. They have access to special features like analytics that help them understand their audience.
- **Education/Discovery Users:-**  
Users interested in discovering new music based on their preferences  
Or expanding their music knowledge. They look forward to the intelligent recommendation algorithm that helps them access a wide range of genres.

In terms of importance, satisfying the needs of the user is crucial as it helps in developing a broader user base. However Premium subscribers are essential as they contribute to the revenue of the service.

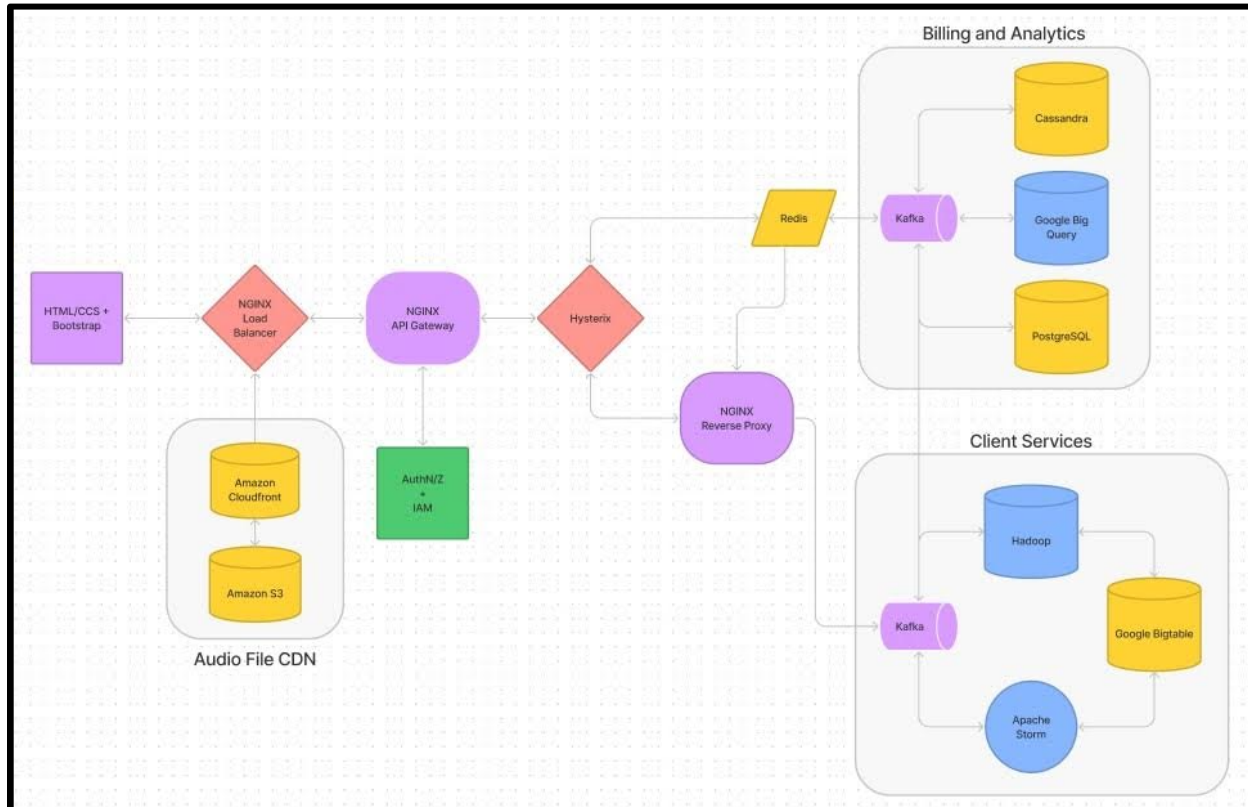
## 2.4. Operating Environment

For performance reasons a minimum RAM size of 1.4MB is recommended. TLS functionality must be integrated in the eSDK, approximately 412 kB will be allocated on the heap. The read-only memory footprint of the SongSphere Embedded SDK binary is approximately 378 kB if the integration provides audio decoders and TLS functionality, and 901 kB if the eSDK implements TLS and Vorbis decoding. Each target device needs to provide the SongSphere Embedded SDK binary with access to UDP/TCP sockets and hostname look-up. The Embedded SDK is buildable for several common chipset and OS combinations. You will need to provide us the toolchain to compile the binary to a particular chipset and operating system (a “build request”). The target device must run a SongSphere Connect ZeroConf stack. This can be done either by



Academic Year: 2021\_22

enabling the built-in ZeroConf stack in the Embedded SDK, or by implementing the ZeroConf API. The ZeroConf API specification is provided as part of the Embedded SDK package.



## 2.5. Design and Implementation Constraints

In developing a project like SongSphere, several constraints may limit the options available to developers:

- 1.Streaming Technology and APIs:** SongSphere relies heavily on streaming technology to deliver music to users. Developers must adhere to its API guidelines and streaming protocols, limiting their options for alternative streaming technologies or approaches.
- 2.Platform Compatibility:** SongSphere is available on various platforms such as iOS, Android, Windows, and macOS. Developers must ensure compatibility across these platforms, restricting their choices of programming languages, frameworks, and tools to those supported by each platform.
- 3.Security Considerations:** Given the sensitive nature of user data and payment information stored within Spotify accounts, developers face stringent security requirements. This includes encryption standards, secure authentication methods, and protection against cyber threats, constraining their options for implementing certain features or technologies.





**Academic Year: 2021\_22**

**4.Licensing and Copyright Regulations:** Developers must comply with licensing and copyright regulations when integrating third-party content (e.g., music tracks, albums) into the SongSphere's platform. This may limit the selection of available music content and restrict the use of certain technologies or algorithms for content recommendation and personalization.

**5.Scalability and Performance:** As SongSphere serves millions of users globally, developers must design the system to handle high traffic volumes and ensure optimal performance. This may impose constraints on database technologies, server infrastructure, and caching mechanisms to support seamless streaming and user interactions across different devices and network conditions.

By considering these constraints, developers can make informed decisions during the design and implementation phases of the SongSphere project, ensuring compliance with technical, legal, and user requirements.

## **2.6. User Documentation**

SongSphere is an extremely easy to use software which is easily available to download from play store or app store. There is a user manual available on our official website if any help is required. In-app support is also provided for any trouble you may face at any point. There are various tutorials available online from our team as well as contributors from around the world which make a simple and seamless experience possible for all. Our software team works hard 24\*7 to provide you with a smooth listening experience but sometimes bug may occur in the software which should be reported.

## **2.7. Assumptions and Dependencies**

Several assumptions may impact the requirements outlined in the SongSphere Software Requirements Specification (SRS):

### **Content Licensing Agreements:**

It is assumed that the necessary licensing agreements for music and podcast content are obtained legally and adhere to industry standards. Failure to secure proper licensing may result in limitations on the available content and could necessitate alterations to the platform's features and offerings.

### **Third-Party API Stability:**

The successful integration of third-party APIs for features like content recommendations and social media sharing is assumed to be reliable and stable. Any unforeseen changes or disruptions in these external services may require adjustments to maintain the intended functionality of SongSphere.

### **Internet Connectivity for Users:**

SongSphere relies on users having consistent internet connectivity for content streaming. It is assumed that users will have reliable internet access, and any deviations from this assumption may impact the user experience, potentially requiring offline features or alternative streaming solutions.



**Academic Year: 2021\_22**

**Device and Platform Updates:**

Users are expected to keep their devices and platforms (e.g., mobile apps, and web browsers) updated to the latest versions. Assumptions regarding user device capabilities and software updates may affect the compatibility and performance of SongSphere, requiring periodic assessments and adjustments.

**Compliance with Privacy Regulations:**

SongSphere assumes that user data handling and privacy practices align with existing and evolving regulations. Any changes in privacy laws or unexpected legal developments may necessitate modifications to the platform's data management and security measures.

These assumptions highlight potential areas of uncertainty that could influence the development and operational aspects of SongSphere. Regular monitoring and communication with relevant stakeholders are crucial to address any discrepancies and ensure the successful implementation of the platform.

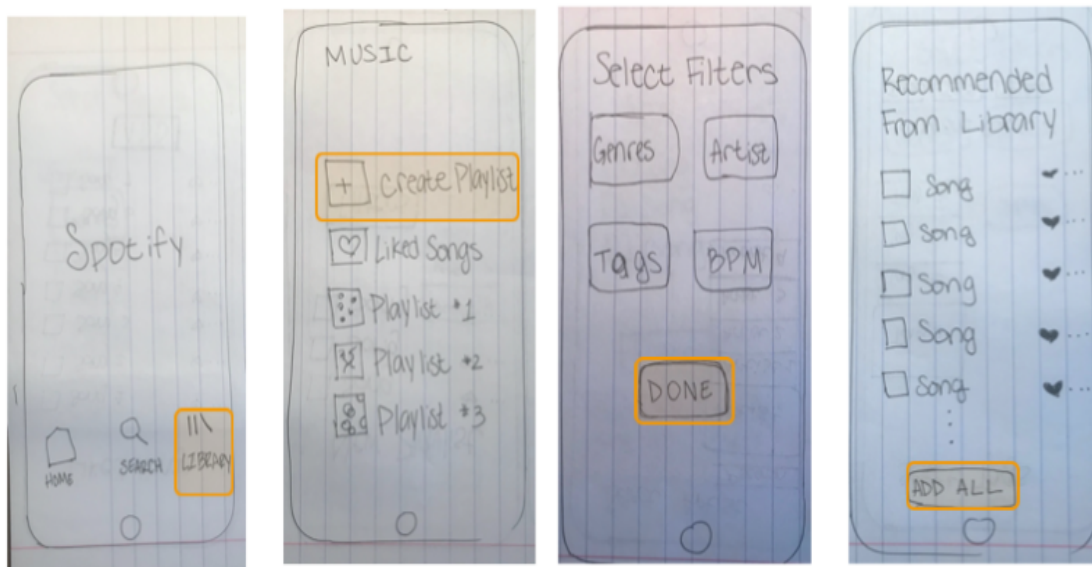
Academic Year: 2021\_22

## 3. Specific Requirements

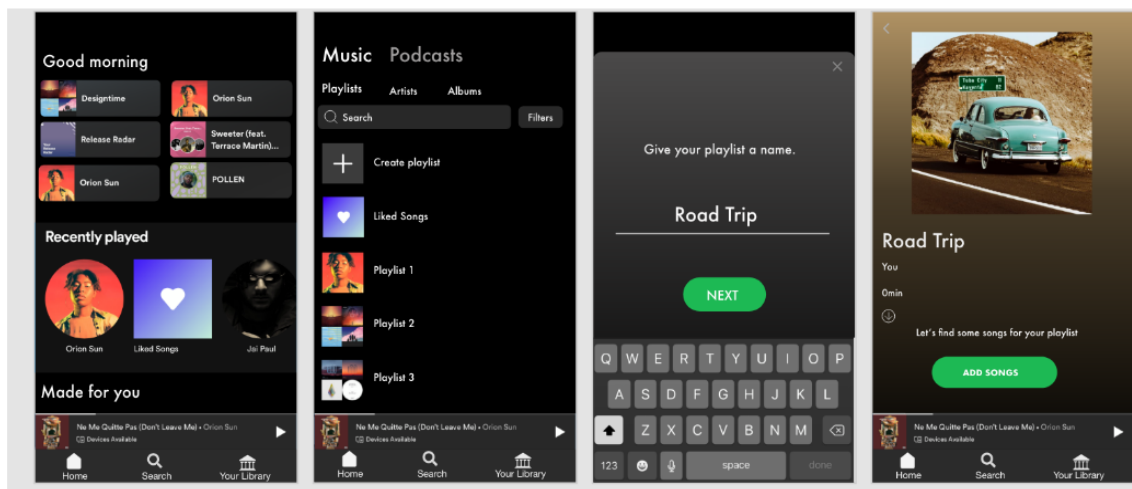
### 3.1. External Interface Requirements

#### 3.1.1. User Interfaces:

This is a low-fi prototype for the process of creating a playlist, incorporating the new filter feature.



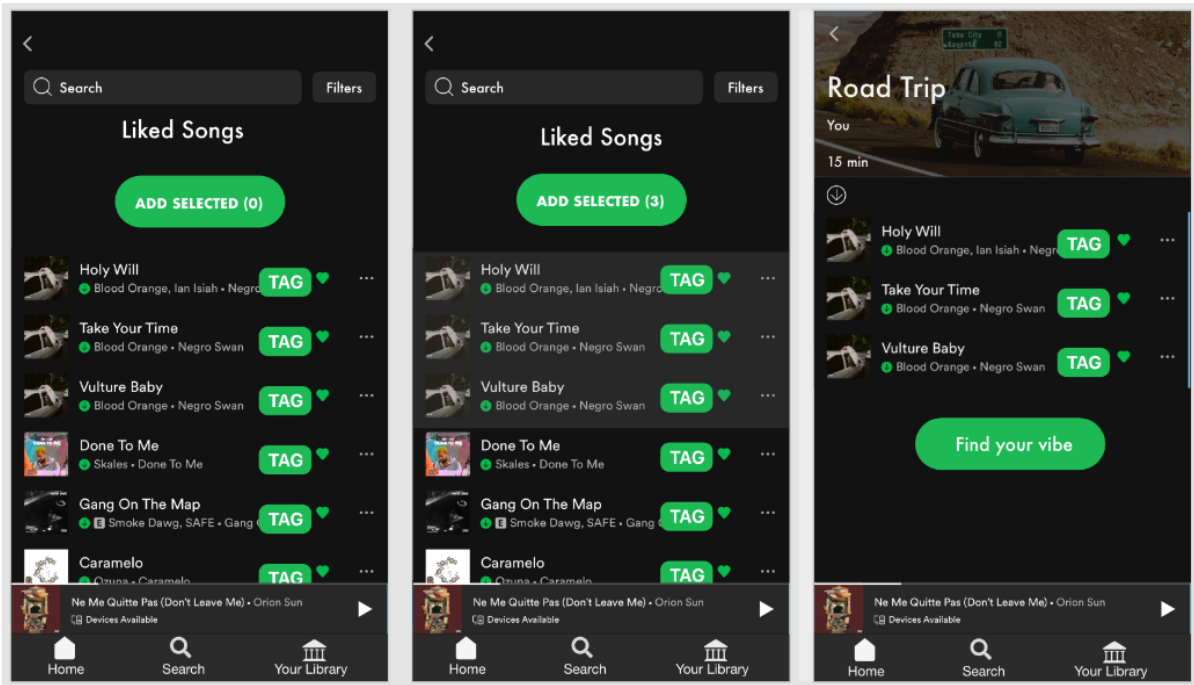
The process begins by going to the library and pressing “create playlist” and giving it a name. Let’s say we are going on a road trip!





### Academic Year: 2021\_22

After pressing “add songs” users would go directly to the liked songs page and would be prompted to choose as many songs as they would like to start the playlist as a basis for the “vibe.” Once a few songs are selected, users go to “Find your vibe” which is where they can specify preferences for suggestions.



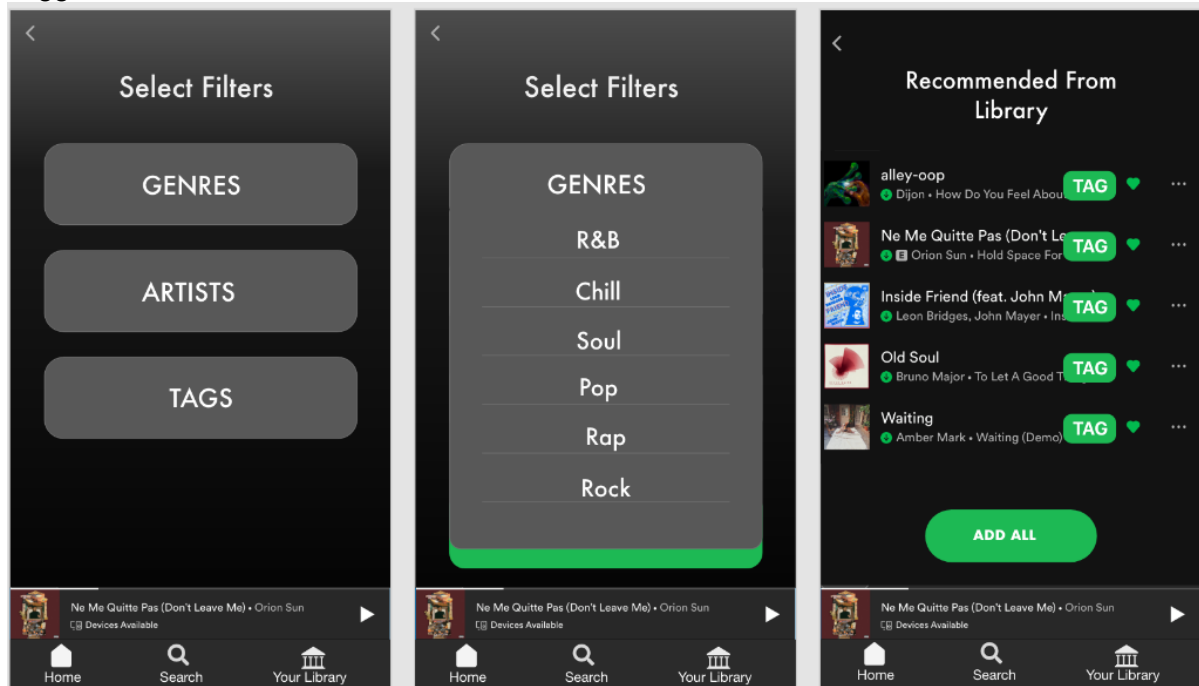
The tag can be defined as a label or set of labels attached to each song. Songs can have several tags that describe their “vibe” or mood. Tags are reported by users themselves, and the most prominent tag is displayed on these screens. However, if the user wants to find songs with the same tag, they can click each tag to be taken to the “tag dashboard” where songs with the same tag are listed. — This is yet another way that users can add songs of a certain “vibe” and find new music!

Then, users go through the filter feature of drop down menus to select their specific criteria for suggestions. Here, the user can be as **specific** or as **vague** as they would like, to accommodate very specific preferences and those who are more open to a variety of



Academic Year: 2021\_22

suggestions.



### 3.1.2. Hardware Interfaces

The recommendation system can work on any internet connected device. These devices should have some limit requirements to make the application run effectively. We expect that the processor speed and internet speed are high.

### 3.1.3. Software Interfaces

First of all the system will work on any platform. Internet connection is a must to reach the system. Moreover, most of the application will be coded by Java. Java APIs of database management tools such as Neoclipse, which is a standalone workbench application to interact with Neo4j. Moreover, some query languages like Cypher will be used. Some tools and software are listed below.



Academic Year: 2021\_22

Name	Version number	Source
NEO4j	2.0.0-M06	<a href="http://www.neo4j.org/">http://www.neo4j.org/</a>
Netbeans	7.4	<a href="https://netbeans.org/">https://netbeans.org/</a>
Windows	7/8	Microsoft Company
Cypher	-	<a href="http://www.neo4j.org/">http://www.neo4j.org/</a>
Neoclipse	-	<a href="http://www.neo4j.org/">http://www.neo4j.org/</a>
Java	SE 7	Oracle Corporation

### 3.1.4. Communications Interfaces

**3.1.4.1 OAuth for Authentication:** SongSphere uses OAuth (Open Authorization) for user authentication. This involves a secure and standardized authorization protocol that allows the app to access a user's account without exposing the user's credentials.

**3.1.4.2 Internet Communication:** The app relies on internet communication to connect with its servers for tasks such as streaming music, syncing playlists, and updating content. This involves the use of standard internet protocols such as HTTP/HTTPS for data transfer. The SongSphere app uses streaming protocols to deliver music content efficiently over the internet. These protocols enable the app to stream audio data in real-time, providing a seamless listening experience.

## 3.2. Functional Requirements

In this section, we will explain the major functions of the Recommendation System.

### 3.2.1 Client

The client-side of the system will be an application with a user interface that is integrated into a music listening website or application. This application gathers the information from users, investigates some actions of the users, and provides the connection with the server. This application is the client-side interface of the Music Recommender, so it does not include the functionalities of the host music environment such as playing music etc.

- **Requesting recommendations-** The client-side application must allow user to request recommendations manually, and

interact with the server to receive recommendations.

- **Evaluating songs-** It must be able to evaluate songs and send appropriate information to the server.





**Academic Year: 2021\_22**

- **Investigating user-** The system will follow the interaction of users with music items., and send these obtain back to the server.
- **Display recommendations-** The application must display the recommendations that are obtained from the server to the user in a proper way by providing a GUI.

### **3.2.2 Server**

The server-side system will hold the entire data in a graph database, and must include all functionality to perform operations on this database, receive requests from the clients, evaluate, create and send recommendations etc.

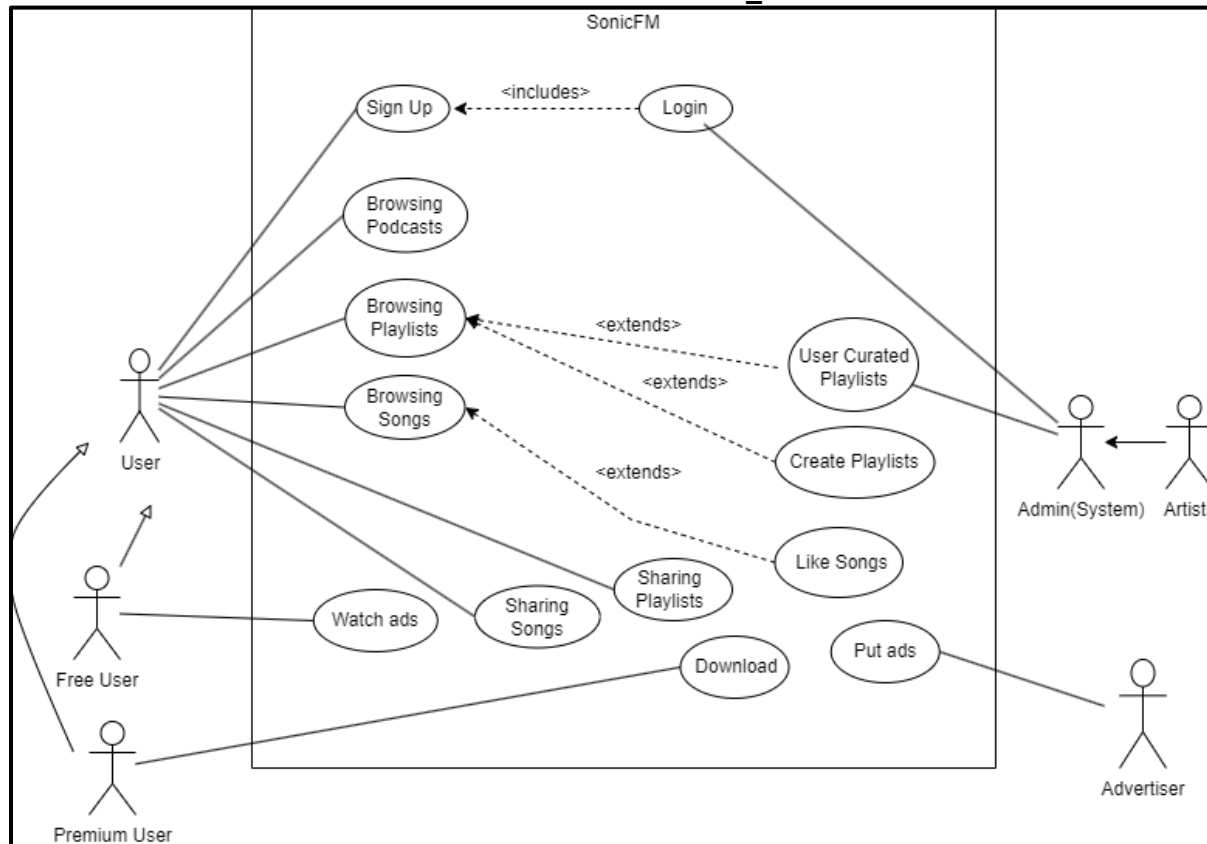
- **Handle recommendation requests-** The server application shall obtain and handle requests for recommendations.
- **Store evaluations-** The server application shall receive and store music evaluations
- **Data storing-** The server application shall be able to store the newly retrieved data to the database.
- **Recommend using content based filtering-** The server application shall be capable of producing recommendations by interpreting the content and evaluations by actual user.
- **Recommend using collaborative filtering-** The server application shall be capable of producing recommendations by interpreting the evaluations given by actual user and other similar users.
- **Recommend using contextual collaborative filtering-** The server application shall be capable of producing recommendations by interpreting contextual information given by the users and evaluations given by the actual user and other similar users.

## **3.3. Behaviour Requirements**

### **3.3.1. Use Case View**



Academic Year: 2021\_22







Academic Year: 2021\_22

## 4. Other Non-functional Requirements

### 4.1. Performance Requirements

- 1) The app will run in the background without consuming too many resources.
- 2) The app should ensure that music playback actions, such as play, pause, skip, and volume control, are responsive, with a recommended time until action takes effect, about 500 ms for standalone devices and 1000 ms for multi-device systems.
- 3) The app should support high-quality audio streaming at 320 kbps, and ensure that the audio system does not generate any unwanted noise or glitches.
- 4) The app should track user listening habits for better prediction of playlists.
- 5) The song library and playlists should be able to handle more than 100 songs to be stored

### 4.2. Safety and Security Requirements

#### 4.2.1 Safety Requirements:

- **Volume Limitation for Hearing Protection:** The product must include a feature that allows users to set a maximum volume limit to prevent potential hearing damage. This limit should be customizable by users based on their preference, but it should have a default setting in compliance with recognized safety standards for prolonged audio exposure.
- **Driver Mode for Safe Driving:** SongSphere must have a "Driver Mode" that automatically simplifies the user interface, limits interaction, and promotes hands-free operation when the platform detects that the user is in a moving vehicle. This is to ensure distraction-free driving and reduce the risk of accidents caused by interacting with the app while driving.
- **Emergency Notification Override:** The product should be capable of integrating with emergency alert systems to automatically lower or pause the playback in the event of critical alerts, such as severe weather warnings, public safety announcements, or disaster notifications. This ensures that users receive important information without distraction.

#### 4.2.2 Security Requirements:

- **Data Encryption:** All user data, including personal information, payment details, and listening history, must be encrypted both in transit and at rest to prevent unauthorized access.
- **Multi-Factor Authentication (MFA):** SongSphere must implement a multi-factor authentication mechanism to enhance user account security, requiring users to verify their identity through at least two different methods, such as a password and a one-time code.
- **Privacy Controls:** Provide users with granular privacy controls, allowing them to manage who can access their listening history, playlists, and other personal information. Transparent privacy policies should be in place, adhering to relevant data protection regulations.



Academic Year: 2021\_22

### 4.3. Software Quality Attributes

**4.3.1 Reliability:** Reliability is a critical quality attribute for SongSphere to ensure a consistent and dependable user experience.

- **Availability:** SongSphere shall aim for 99.99% uptime, allowing users to access the platform without significant disruptions. This will be achieved through redundant server architecture, automated failover mechanisms, and proactive monitoring to detect and address issues promptly.
- **Error Handling:** The system shall provide informative and user-friendly error messages, guiding users in the case of unexpected errors. Error logs will be maintained for analysis and continuous improvement of system reliability.

**4.3.2 Usability:** Usability is paramount for customer satisfaction and engagement with the SongSphere platform.

- **Intuitive User Interface:** The user interface shall be designed following established usability principles, ensuring simplicity, consistency, and intuitiveness. Usability testing will be conducted during development to validate the design choices.
- **Personalization Features:** SongSphere shall incorporate machine learning algorithms to analyze user preferences and behavior, providing personalized recommendations and playlists, enhancing user engagement and satisfaction.

**4.3.3 Maintainability:** Maintainability is crucial for facilitating future updates, bug fixes, and overall system evolution.

- **Modularity and Code Comments:** The codebase shall be organized into modular components, each with well-defined functionalities. Additionally, comprehensive comments within the code will be maintained to enhance code readability and ease of understanding for future developers.
- **Version Control:** SongSphere shall use a version control system (e.g., Git) to manage and track changes systematically. This ensures the ability to revert to previous versions, collaborate effectively, and maintain a stable and evolving codebase.

**4.3.4 Security:** Security is a fundamental attribute, ensuring the protection of user data and system integrity.

- **Regular Security Audits:** The platform shall undergo regular security audits, including penetration testing and code reviews, to identify and address potential vulnerabilities. This will be conducted at least twice a year to stay ahead of emerging security threats.
- **Encrypted Communication:** All communication between the SongSphere app and servers shall be encrypted using industry-standard protocols (e.g., TLS). This ensures the confidentiality and integrity of user data during transit.



Academic Year: 2021\_22

## 5. Other Requirements

<This section is **Optional**. Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

## Appendix A – Data Dictionary

<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>

## Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist the Teaching Assistant to determine the effort put forth to produce this document>