
GROUP 08 - SESlay

**Museos
Vision Document**

Version 1.0

Museos	Version: 1.0
Vision Document	Date: 27/10/23
<document identifier>	

Revision History

Date	Version	Description	Author
27/10/23	1.0	Initial documentation	SESlay

Museos	Version: 1.0
Vision Document	Date: 27/10/23
<document identifier>	

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 References	4
2. Positioning	4
2.1 Problem Statement	4
2.2 Product Position Statement	4
3. Stakeholder and User Descriptions	5
3.1 Stakeholder Summary	5
3.2 User Summary	5
3.3 User Environment	5
3.4 Alternatives and Competition	6
4. Product Features	6
5. Non-Functional Requirements	7
5.1 Product requirements	7
5.2 Organization requirements	8
5.3 External requirements	8

Museos	Version: 1.0
Vision Document	Date: 27/10/23
<document identifier>	

Vision (Small Project)

1. Introduction

1.1 Purpose

The purpose of this document is to collect, analyze, and define high-level needs and features of the Museos web application. It focuses on the capabilities needed by the stakeholders and the target users, and why these needs exist. The details of how Museos fulfills these needs are detailed in the use-case and supplementary specifications.

1.2 References

- Theoretical slides and project's specifications provided by lecturers and TAs of the HCMUS' 2023 Introduction to Software Engineering course.

2. Positioning

2.1 Problem Statement

The problem of	finding a place to enjoy music without the interruption of advertisements during a song, and to upload music products with promising values in return
affects	anyone with an interest in music.
The impact of which are	music listeners who are annoyed by the massive amount of untimely advertisements that keep impeding their immersion, and artists who don't know a reliable platform to upload their music while receiving financial aid.
A successful solution would be	a web application lets music enjoyers have an unhampered musical experience, and provides artists with a place to share their creations and earn a portion of revenue in return.

2.2 Product Position Statement

For	anyone with an interest in music
Who	wants to discover new music, or share their creations to a community.
Museos	is a web application
That	provides music enthusiasts with a platform to share and listen to wonderful music.
Unlike	the current music-sharing platforms, i.e., Spotify, SoundCloud
Our product	does not only stop at contributing to sharing the joy of music but also promoting musical talents. Additionally, our platform can serve as a great kick-start for new upcoming artists who yearn to reach a wider audience, and even receive a small revenue through tipping in the process.

Museos	Version: 1.0
Vision Document	Date: 27/10/23
<document identifier>	

3. Stakeholder and User Descriptions

This section describes the users of Museos application. There are 3 types of users in the system, which are the User, the Artist (another aspect of the User), and the Administrator.

3.1 Stakeholder Summary

Name	Description	Responsibilities
Development team	Cooperate with stakeholders and users to turn the requirements and needs into real applications. Also help the administrators in maintaining and supervising the system.	Plan, design, develop, and maintain the application.
Supervisor	Work together with the development team to ensure the quality of the project output.	Provide step-by-step guidance and give recommendations for adjustments. Also provide documents for the development team.
Normal User	Listen to music and interact with artists on the system.	Help ensure that the music streaming and social functions of the application work correctly.
Artist User	Users whose primary uses are to produce, upload, and expose their music to the public.	Help ensure that the artist-based functions work correctly (e.g., song uploading, tip receiving, etc.).
Administrator	Super user with the role of supervising the system and other Users.	Help ensure the functionality of the administrative functions, and a healthy community.

3.2 User Summary

Name	Description	Responsibilities	Stakeholder
Normal User	One of the two keys User of the system.	Use the application to discover, listen to music, and support artists. They are the main interest of the Artist Users.	Self.
Artist User	The other key User of the system. An extent of a Normal User.	Use the application as a place to upload their musical products, and earn tips from the Normal Users.	Self.
Administrator User	Admin of the system, having full authority over other Users.	Supervise the platform and other Users' activities. Moderate the content coming from Users, and enact bans on Users, or deletions of songs or comments when necessary.	Self.

3.3 User Environment

- The application is used by people with smart devices that can access the internet and web platforms.
- The application's main audience is music enthusiasts, artists, or simply anyone with an interest in music.

Museos	Version: 1.0
Vision Document	Date: 27/10/23
<document identifier>	

- The application can be used singly by one user without the need to interact or cooperate with others. Otherwise, it can act as a platform for users to express their mutual passion for music.
- A song usually lasts about 3-5 minutes. A user with a listening need can spend any desirable amount of time on the platform. As for artists, they only need up to 2 minutes to upload their song.
- The system in use today runs on a web platform, it should be available as a mobile application in the future.

3.4 Alternatives and Competition

- There are many alternatives to Museos, as music streaming platform, for instance:
 - Spotify: Despite the strength of massive music databases and user community, they still lack any method for interactions between users, and any scheme to support artists.
 - SoundCloud: Nearly identical to Museos. However, no donating function is in place, and their act of moderation is progressively worsening.
- Our golden opportunity is to buy out Spotify's music content. Then through that, attempt to out-perform all of SoundCloud's functions to become the dominating music streaming service.

4. Product Features

No.	Feature	Description	Priority
1	Sign up	Allow the user to create an account on the platform.	High
2	Sign in	Require the user to sign in before any function is made available.	High
3	Home page	The home page covers currently trending songs and artists. This page will be shown to the user before any further interaction.	Medium
4	Search for song/artist	A method for users to search for a specific song or artist.	High
5	Listen to music	A method for users to listen to music.	High
6	Uploading music	A method for artists to upload the music to the platform.	High
7	View the song lyrics	A way for users to view the content of the song they are listening to.	Low
8	Add a song to user playlist	Help users to compile their favorite songs.	Medium
9	Create artist special playlist	A special playlist only available for fans who have donated.	Low
10	Donate star to artists	A method of giving donations from fans to their favorite artists.	High
11	Comment section for songs	A comment section for each song for fans to share their thoughts.	Medium

Museos	Version: 1.0
Vision Document	Date: 27/10/23
<document identifier>	

12	Convert/Exchange currency	Converting from real money to star for users and vice versa for artists. As a proof of concept, we will only implement a simple money-to-star for users (i.e., no actual banking services involved).	Medium
13	Give heart	A method for users to rate hearts for their favorite songs or artists.	Medium
14	Ban user	A method for the Administrator to ban users who violate the community standards.	Medium
15	Delete Song	Artists can delete their songs.	High
16	Delete Song (Admin)	Administrators can delete songs that go against community standards.	Medium

5. Non-Functional Requirements

5.1 Product requirements

- Usability requirements:
 - Availability: the website should be accessible and compatible with devices.
 - Recognition: allow songs and artists to be approached widely.
 - Consistency: maintain the consistency of both design and functionality of the web with detailed guidelines for down cases or having problems, come along with support assistants as well.
 - Accessibility: all people can access the website.
 - Audio quality: ensure the best quality of sound for the songs.
 - Navigation: the function on one page is limited and designed obviously so that users can easily follow the path to songs and artists.
- Efficiency requirements:
 - Performance requirements: music should be loaded immediately or within 1 to 2 seconds delay when users start playing, skip tracks, etc. The website is ensured to be run smoothly for the best user's experiences.
 - Space requirements: the library contains up to 30000 songs with each one having a quality approximately between 96 kbps and 160 kbps, and the highest can be up to 256 kbps or 320 kbps.
- Reliability requirements:
 - Data backup: songs are backed up weekly so that the data is ensured from loss by system errors.
 - Copyright requirements: when publishing songs, this means authors agree with web's copyright that the songs are their product themselves.
 - Payment via donating feature: users' donation is secured and there will be refusal whenever the system has any problems so that users' donation is safe.
- Portability requirements:
 - Responsiveness: website UI/UX is responsive on various devices including mobiles or tablets beside laptop.
 - Platforms and browsers support: the website is fully functional throughout various platforms and browsers such as Windows, macOS, Android, Chrome, Safari and Edge.

Museos	Version: 1.0
Vision Document	Date: 27/10/23
<document identifier>	

5.2 Organization requirements

- Implementation requirements:
 - Implementation team: there are 7 roles within a 5-member team including: project manager, business analyst, front-end developer, back-end developer, UI/UX designer, database developer and tester go with corresponding responsibilities.
 - Implementation plan: the website will be implemented and tested in 3 months before being put into practice.
- Standard requirements:
 - Legal requirements: provide comprehensive documentation such as data privacy policies, terms of use, terms and conditions and requirements regarding copyright, plagiarism, anti-spam laws, etc.

5.3 External requirements

- Interoperability requirements:
 - Information Interoperability: ensure data is saved with appropriate format such as M4A, MP3 or WAV.
 - API compatibility: if a program written against one version of that API will continue to work the same way, without modification, against future versions of the API.
- Ethical requirements:
 - User protection: data of users including their playlist and personal information is protected when they agree with terms and conditions of the website for the purpose of preventing harassment, plagiarism, etc.
- Legislative requirements
 - Payment Regulations: ensure secure payment when using the feature “donate” for favorite artists.