HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and Communication Technology



**FEASIBILITY REPORT**

**for Jellimix Music Website**

**Team name: Jellimix**

**Team ID: 08**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team member** | **Student ID** | **Team member** | **Student ID** |
| Nguyen Thi Hong Anh | 20176679 | Nguyen Thi Minh Chau | 20184238 |
| Tran Thai Duong | 20176732 | Dao Cao Duy | 20176738 |
| Nguyen Viet Hoang | 20176762 | Nguyen Tri Hung | 20176773 |
| Tran Phi Hung | 20176774 | Pham Vu Minh | 20176818 |
| Nguyen Manh Phuc | 20176845 | Le Thanh Son | 20176860 |

*Hanoi, October 2021*

**Table of Contents**

[1 Executive Summary 2](#_Toc86133494)

[2 Preliminary Requirements Analysis 2](#_Toc86133495)

[2.1 Application Overview 2](#_Toc86133496)

[2.2 Functional Requirement 3](#_Toc86133502)

[2.3 Nonfunctional Requirement 4](#_Toc86133503)

[3 Process to be followed 5](#_Toc86133507)

[4 Suggested Deliverable 6](#_Toc86133515)

[5 Business Consideration 7](#_Toc86133516)

[6 Risk Analysis 7](#_Toc86133517)

[6.1 Time risks 7](#_Toc86133518)

[6.2 Resource risks 8](#_Toc86133519)

[6.3 Functionality risks 8](#_Toc86133520)

[6.4 Risk management/minimization 8](#_Toc86133521)

[7 Appendix 9](#_Toc86133522)

# Executive Summary

The following proposed system is a music streaming service, built upon an ideal to help ease users’ experience while exploring and engaging in the wonderous music industry. Anyone who loves music and has access to the internet can be a potential user of the system. Though there have been many similar music streaming systems, Jellimix can sure stand out with its elegant, alluring yet friendly user interface, state-of-the-art algorithms to help users personalize their space according to their preference, and of course, quality music with little to no payment fee.

# Preliminary Requirements Analysis

## Application Overview

### Objective

The project aims to provide a wide variety of songs to cover all existing genres and preferences, allow users to create, update, delete playlists of choice, as well as follow multiple artists on the platform. The project can also help users discover new songs that may fit their personalities by analyzing their personal playlists.

There will be no administrative roles, every procedure is automated so that users can be serviced without delay. The goal is to provide the user with utmost care and comfort while using the product.

### Business Objective

This project makes profits through advertisement, thus can help reduce the cost for users. By lowering the cost of use but still maintaining high-quality service, a competitive playfield is created. The project can still attract newcomers through its appealing price range and keep hold of loyal users using exquisite service.

### Current Business process and Rules

Music has long been a haven for everyone. It itself contains so many nuances of emotions that it is always possible for one to find a song suitable for his/her feeling. With such resonance, music plays a crucial part in each culture, and life in general. Therefore, people seek platforms to divulge their favorite songs. However, it is sometimes difficult for people to find media sources easily and conveniently.

Currently, in order to hear a piece of music, users have to surf through different websites to find their favorite track. Usually, this stage consumes a considerable amount of time since for the same song, there is a variety of tracks with different qualities and mixes. After having found the desired song, users choose it to play on the music player embedded in the application or website. Not all music platforms support creating a community where people with the same tastes of music can share their interests. This makes it inconvenience for users to connect and discuss.

Jellimix is expected to make a name for itself in the market and compete heads on with other equivalent products. Its goal is to solve the stated inconveniences that music lovers have stumbled upon and help users access various songs with ease as much as possible.

### User roles and responsibilities

* Play song
* Create, update, delete playlists
* Add/Remove song from playlist
* Follow other users
* View information

### Interaction with other systems

The system will incorporate an existing technology, namely Jellyfin, to build a hosting server for all songs (and music videos, if expansion is needed). The system will depend on this server to store and fetch songs for users.

## Functional Requirement

* Play song (with controls)
* View song's information
* Download songs
* Search songs, artists, playlists, albums
* Create, update, delete playlists
* Recommend song
* Add song to queue
* Register
* Login/logout
* Follow another user
* View other users’ playlists
* View personal activities

## Nonfunctional Requirement

### Security

All information about users must be kept confidential and used for the project’s purpose only. Any related transactions also have to be secured.

### User interface

The interface of the website must be user-friendly and intuitive. It should support switching between dark and light mode or choosing theme in a set of themes. The texts should be designed for ease of reading. The response speed should be fast so as not to interrupt user’s experience.

### Response time & Durability

The response time for each request must not exceed 1 second in normal time. In peak time, this should not be more than 1.5 seconds.

The system is expected to operate 24/7, with the number of users accessing per hour is 100, and the number of requests can be served at the same time is 50.

#### Operating system

The system will be run on a Linux server and serve users on different operating systems (e.g. Windows, Linux, macOS) and browsers (e.g., Chrome, Firefox, Opera, etc.)

#### Optional features

Jellimix may support multiple languages, in particular, English and Vietnamese, since the aim of the project is to provide a free and convenient music platform for users in Vietnam.

# Process to be followed

## Phase 1 (29/09/2021 - 05/10/2021)

Project orientation and technology research

## Phase 2 (06/10/2021 - 19/10/2021)

Requirement analysis

* Use case diagram, use case analysis: Design the main functionalities of the system
* Analyze functional requirements

## Phase 3 (20/10/2021 - 09/11/2021)

Architecture analysis (3)

* Class diagram, interaction diagram: Design components of the system and how the components interact with each other

## Phase 4 (10/11/2021 - 15/11/2021)

Interface design: Design initial graphical user interface

## Phase 5 (10/11/2021 - 23/12/2021)

Data modeling: Design database using ER models,

* Jellyfin
* DBMS: SQL or NoSQL

## Phase 6 (10/11/2021 - 07/12/2021):

Programming and Testing

* Programming
* Unit test
* Integration test

## Phase 7 (08/12/2021 - 14/12/2021):

Project deployment

# Suggested Deliverable

## Management deliverables

### Requirement analysis

Official documents of the system, describing both functional and non-functional requirements from the users, with details on actors, use cases and external system interaction. These documents shall be updated regularly and follow suit with the current design and implementation of the system.

### Design documents

Includes Diagrams, SRS and Interface design documents. These documents give customer a more coherent view on what the system will look like and how it will be implemented. These documents are based on the Requirement analysis of the system, therefore must also be updated accordingly.

### Reports

Reports include progress documents (in the form of a 20-slide presentation of 10-page PDF) to let customer keep track of the current progress of the project team. These documents give specific description on the goals that need to be achieved, what have been done by whom at what specific time frame.

## Technical deliverables

### Source code

The development team produces source code for modules that need implementation according to schedule to the customer. These modules must be tested before submission and delivered with a detailed description of the test cases passed or failed.

# Business Consideration

The project uses free music resources from other available online music platforms. There will be some pieces of media that will have to be paid before use. The team try not to use these resources to respect the copyright rules. The team will also not use any third-party applications to download, store and distribute these licensed tracks, with the same purpose of copyrights.

The project will use open-source softwares and technologies, as well as given hardware resources only to implement the functionalities of the system.

Since the team does not plan to trademark any names concerning the software system, copyright is not an imminent issue. However, at the time of implementation, all source codes are accessible to team members and the instructors only to prevent plagiarism.

All the information of users used in this systeF,{656c3e79-1b5b-4d90-8c7c-d4ce0ed7e3aa}{181},3.125,3.125m is guaranteed to be kept confidential for project purposes only.

# Risk Analysis

As for any project, this undertaking is not risk-free. These are the major risk categories that have been identified:

## Time risks

As the course requirements, this project must be completed within one academic semester, without any extensions. This introduces the risk that the system may not have all of its functionalities fully covered or covered with errors.

Although the team has 10 members, each of which has different timetable and schedules. Furthermore, because of the COVID-19 pandemic, there is little chance that the team members can gather in-person to discuss about problems. Therefore, the possibility of having little time for meetings is obvious. This drives to the second risk that some parts of the development part will take longer than expected due to miscommunication.

## Resource risks

Resource risk includes both technological resources, and human resources.

For the tools, due to cost and external constraints, only free-features of the desired technologies are used to make up this system. Most of the software is open-source, and hardware for implementation is also limited.

For the human resources, most of the team has knowledge about backend implementation, but not on the frontend. Moreover, there is few people who understand and have good practices frontend and backend integration before the release.

## Functionality risks

Functionality risks are related to the functionalities of the system. Issues that may arise in this category include:

* GUI is not well-linked to the backend
* Minor bugs due to uncovered cases of user behaviors.
* Because of limited server storage, the system is not as robust as desired, and may not be able to reply in a timely fashion.

## Risk management/minimization

Having understood above pitfalls, the team prepares to take actions to minimize and eliminates the risks

* **For the technology risk**: The team tries to have profound knowledge to take advantages of the open-source software, to minimize the data memory, data transfer cost to make the system faster. Tests are conducted frequently to recognize bugs as soon as possible, prevent the project from being corrupted when there are too many unsettled errors at the end.
* **For the human and time risk**: The team is divided into smaller groups based on different aspects, each group consists of one or two member who has more knowledge in the selected field and will in charge of guiding the group. Each smaller group will decide on their meeting schedule, and the whole team will have an online meeting every Thursday evening to report and discuss about the progress. Every team member is obligated to participate into the progress and follow the schedule strictly. If there is any problem that must be negotiated between team members or groups, he or she must raise it up immediately, and there must be a meeting to notify to the whole team and confirm the changes.
* **For the functionality risk**: The team puts effort into implement a detailed documentation for building the system at the beginning of the project. Discussions are frequently performed to make sure everyone understands and is following the desired architecture and implementation. Any requirement changes must be carefully analyzed and accepted by the whole team.

# Appendix

*S*

*Streaming:* Relating to or being the transfer of data (such as audio or video material) in a continuous stream especially for immediate processing or playback.