# Introduction

The aim of this document is to gather and analyze and give an in-depth insight of the complete **Music Information Retrieval for audio fingerprinting** by defining the problem statement in detail. Nevertheless, it also concentrates to develop a software for users or creators that provides them with the facility to protect their work against plagiarism using **statistical signal processing.**

## Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

## Scope

This software system will be a Signal Processing System for a local user and creator. This system will be designed to maximize the user’s productivity by providing tools to assist in fast search for songs, which would otherwise have to be performed manually. By maximizing the user’s work efficiency the system will meet the user’s needs while remaining easy to understand and use.

More specifically, this system is designed to allow a user to manage songs, create playlist, and search songs on sample input. The system also contains a relational database containing a list of Songs, User, Playlist and Hash Table.

## Definitions, acronyms and abbreviations

|  |  |
| --- | --- |
| MIR | Music Information Retrieval |
| FAQ | Frequently Asked Questions |
| ERD | Entity Relationship Diagram |
| JDBC | Java Database Connectivity |
| FFT | Fast Fourier Transform |
| DFT | Discrete Fourier Transform |
| DSP | Digital Signal Processing |
| PCM | Pulse Coded Modulation |

## References

[1] Jacob George and Ashok Jhunjhunwala. “Scalable and Robust Audio Fingerprinting Method Tolerable to Time-stretching”. In: 2015 IEEE International Conference on Digital Signal Processing (DSP).

[2] M. Covell. “Content fingerprinting using wavelets”. English. In: IET Conference Proceedings (Jan. 2006), 198–207(9). URL: http : / / digital - library . theiet . org / content/conferences/10.1049/cp 20061964.

[3] Vijay Chandrasekhar, Matt Sharifi, and David A Ross. “Survey and Evaluation of Audio Fingerprinting Schemes for Mobile Query-by-Example Applications.” In: ISMIR. Vol. 20. 2011, pp. 801–806.

[4] Avery Wang et al. “An Industrial Strength Audio Search Algorithm.” In: ISMIR. 2003, pp. 7–13.

[5] Houssemeddine Khemiri, Gerard Chollet, and Dijana ´ Petrovska-Delacretaz. “Automatic detection of known ´ advertisements in radio broadcast with data-driven ALISP transcriptions”. In: Multimedia Tools and Applications 62.1 (2013), pp. 35–49.

[6] Mani Malekesmaeili and Rabab K Ward. “A novel local audio fingerprinting algorithm”. In: Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on. IEEE. 2012, pp. 136–140.

[7] Mark Dolson. “The phase vocoder: A tutorial”. In: Computer Music Journal (1986), pp. 14–27.

[8] Jean Laroche and Mark Dolson. “Improved phase vocoder time-scale modification of audio”. In: Speech and Audio Processing, IEEE Transactions on 7.3 (1999), pp. 323–332.

[9] Heinrich A Van Nieuwenhuizen, Willie C Venter, and MJ Grobler. “The Study and Implementation of Shazams Audio Fingerprinting Algorithm for Advertisement Identification”. In: Proceedings of SATNAC 2011.

[10] Bilei Zhu et al. “A novel audio fingerprinting method robust to time scale modification and pitch shifting”. In: Proceedings of the international conference on Multimedia. ACM. 2010, pp. 987–990.

[11] Rolf Bardeli and Frank Kurth. “Robust Identification of Time-Scaled Audio, 2004”. In: Proceedings of the AES 25th International Conference on Metadata for Audio, London, UK.

[12] Sebastien Fenet, Ga ´ el Richard, Yves Grenier, et al. “A ¨ Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting.” In: ISMIR. 2011, pp. 121–126.

[13] Daniel S Hirschberg. “Algorithms for the longest common subsequence problem”. In: Journal of the ACM (JACM) 24.4 (1977), pp. 664–675.

[14] D. Ellis (2009), ”Robust Landmark-Based Audio Fingerprinting”, web resource, available: http://labrosa.ee.columbia.edu/matlab/fingerprint/.

## Overview

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product.  General description of the project is discussed in section 2 of this document.  Section 2 gives the functional requirements, data requirements and constraints and assumptions made while designing the Music Player.  It also gives the user viewpoint of product.  Section 3 gives the specific requirements of the product.

# Overall Description

## Product Perspective

## As mentioned earlier, Signal Processing System forms part of a Music Player. SPS is responsible for receiving the audio samples submitted by the user and process it to produce the corresponding hash values. SPS communicates its result to the main system and depending on the user’s request either performs store or search operation as per the given input.

## Product Functions

The main purpose of this system is to provide the following facilities:-

* Create an account
* Login
* Create a playlist.
* Add Songs to a playlist
* Add song to the database.
* Delete the playlist.
* Search Samples.

It will aim to optimize resource utilization *within* user-imposed constraints: thus, user satisfaction is the primary concern, as opposed to maximizing CPU utilization.

## User characteristics

Customers can utilize the product to provide services in providing the protection against plagiarism so as to provide the integrity of their original product.

Developers can utilize the libraries so created in this project and utilize them to build innovative products such as Emotion based filtering etc. to provide better user experience

Users can create their customized playlist, upload songs and can search the database.

## Constraints

The current constraints on the project are related to centralized data center for large collection of songs to improve the performance of the system. For testing purposes, a noise free sample is required.

## Assumptions and dependencies

The input sample must be noise free and it must be in .WAV format. The number of sample points in the given audio must be in the power of 2. The frame size is assumed to be 1024. PCM encoding is assumed to be used while recording the sample.

# Specific requirements

## Functionality

### Create an account

#### Inputs

User Information – relevant user data such as name and authentication id that needs to be used to identify the user.

Password – it is used to provide access to the information associated to a particular user.

#### Processing

It stores the user information and password into the database.

#### Outputs

Boolean – Whether the user is accepted or not.

### Login

#### Inputs

User Id and Password – to ensure that only the appropriate user is able to see the playlist associated to that user.

#### Processing

Check whether the user is available or not.

#### Outputs

Boolean – Whether the user is available or not if available it display the information associated to that user otherwise it ask for login information.

### Create a playlist

#### Inputs

Playlist name – name used to identify a playlist.

#### Processing

Provide a unique id to the playlist and store it into the database.

#### Outputs

Boolean – true if the playlist is created otherwise false.

### Search Samples

#### Inputs

Sample to be searched in wav format.

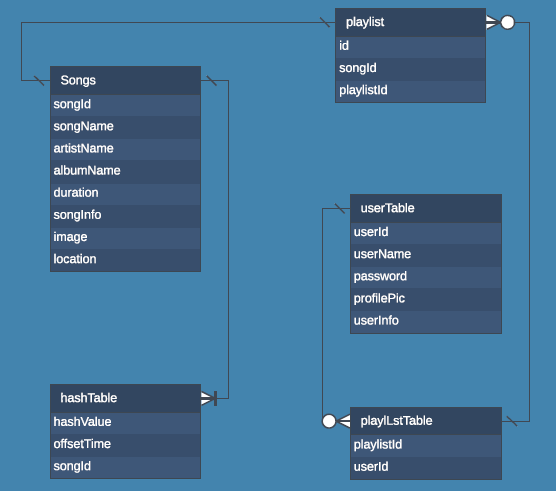
#### Processing

We apply FFT on the given sample and convert the data into frequency domain and create hash values. Then it search the hash in the given database.

#### Outputs

Boolean – true if the song is available otherwise false.

Song info – if the Boolean is true the display the song details.



**Appendixes**

**Index**