

GRAD 695 Research Methodology & Writing

Applied Project

Software Requirements Specification

Title: MusicOn, a music player emulator

Professor: Abrar Qureshi

Author: Prafulla Chandra Munugoti

Term: Spring 2023

Date: 02/05/2023

➤ Problem Statement

In the human history, Audio plays a crucial role in their everyday life. We are subjected to many forms of audio in every walk of life. One such audio that is very important in its own way is music. Music plays its role in a way to express one's feelings and emotions. It also causes a break to certain life hardships. It is an important means of stress relief option to many humans. Music acts like a therapy to keep us calm and makes one cheerful in the joyous moments. Music also plays its role in boosting the morale and self-confidence than being a source of entertainment.

Owing to many advantages of listening to music to humans in every walk of life during various situations they are in. MusicOn is one such application that would allow seamless, robust, freeware customizable music player emulator. It allows users to listen to the audio and immerse in their own world by reducing their day to day worries and hurdles. This emulator is to help them immerse by enabling to play music and digital audio files, creating a playlist by using a backend database to store, add, modify, and remove information on the existing playlists. The MusicOn emulator application graphical user interface opens with various functionalities like play a song, create, and display a playlist, pause, and resume a long song, and change the song, play the previous or next song using a user-friendly graphical user interface with each option available to pick as needed.

➤ System Requirements

The required system functions for the MusicOn a music player emulator product includes

- Add song to songs DB.
- Delete song from songs DB.
- Show songs from songs DB.
- Search songs by artist/Title/Year/Type from songs DB.
- Play a Song stored in songs DB.
- Create a playlist to playlists DB.
- Show all playlists in playlists DB.
- Add songs to existing playlist.
- Play songs in selected playlist in 1). Normal Mode 2). Shuffle Mode 3). Loop Mode

As a music player emulator, the interaction between the user and the service will be done through the Graphical User Interface.

➤ Functional Requirements

Requirement	ID	Name	Description
FR1 Song Functions	FR1.1	Addition	Functionality to add songs to the existing songs database
	FR1.2	Deletion	Functionality to delete songs from the songs database
	FR1.3	Display	Functionality to display songs in the existing songs database
	FR1.4	Search by Title	Functionality to search for the available songs in songs database
	FR1.5	Search by Artist	Functionality to search for available songs in songs database by artist name
	FR1.6	Search by Album	Functionality to search for available songs in songs database by album name
	FR1.7.1	Play Song Normal Mode	Functionality to play a song in normal mode displaying lyrics
	FR1.7.2	Play Song Shuffle Mode	Functionality to play a song in shuffle mode displaying lyrics
	FR1.7.3	Play Song Loop Mode	Functionality to play a song in loop mode displaying lyrics
	FR1.8	Sort Songs alphabetically	Functionality to sort the songs in alphabetical order
FR2 Playlist Functions	FR2.1	Creation	Functionality to create a new playlist to users
	FR2.2	Deletion	Functionality to delete an existing playlist from playlists database
	FR2.3	Display	Functionality to show all existing playlist from playlists database
	FR2.4	Song Addition to Playlist	Functionality to add songs to existing playlist from playlists database
	FR2.5	Song Deletion from Playlist	Functionality to delete songs to existing playlist from playlists database
	FR2.6	Search Playlist by Title	Functionality to search for the existing playlist by title from existing playlists database.
	FR2.7.1	Play Songs Normal Mode	Functionality to play all songs from selected playlist in normal mode displaying lyrics
	FR2.7.2	Play Songs Shuffle Mode	Functionality to play all songs from selected playlist in shuffle mode displaying lyrics
	FR2.7.3	Play Songs Loop Mode	Functionality to play all songs from selected playlist in loop mode displaying lyrics

➤ Non-Functional Requirements

Requirement	Req.ID	Name	Description
NFR1 Performance	NFR1.1	Robustness	System is robust to deal and act upon common error scenarios like unavailable metadata, unsupported file types, data missing, compile issues.
	NFR1.2	Quickness	System is quick enough to play music and response time to be as minimal as possible to any of the user action without any time kill for loading, creating, playing buffering to have a pleasant experience.
	NFR1.3	Quick Recovery	In case of failures, it should be able to fail or recover quickly.
NFR2 Quality	NFR2.1	Memory Management	System should ensure that there is no memory leak for any of its functions.
	NFR2.2	Compatibility	System is compatible to any technology, platform and software that it comes in contact within terms of API's.
	NFR2.3	Error Handling	System should not cause or trigger any events that creates an unrecoverable state.
	NFR2.4	Exception handling	System should be capable of handling any exceptions that occur in the way of usage.
NFR3 Reliability	NFR3.1	Reliable	System needs to be reliable on its performance. example: play functionality should take same time for shuffle, or loop or do another operation.
NFR4 Consistency	NFR4.1	Consistent behavior	System should have a consistent behavior on its functionalities and should not behave differently to different users for same function.
NFR5 Security	NFR5.1	Safety	System needs to strongly secure user data and prevent them from theft or data leak and not lose confidential information
NFR6 Availability	NFR6.1	Available	System should be available for multiple users at the same time and should support the feature and not crash.
NFR7 Usability	NFR7.1	Usage	System is easy to operate and friendly to users. User interaction should be as error prone as possible.
NFR8 Scalability	NFR8.1	Scalable	System must be easy to scale depends on the traffic volume which helps in cost of resource allocation.
NFR9 Maintainability	NFR9.1	Maintenance	system need to be easy to maintain, not so complicated to add new features. System is breaking due to new code changes; it should be a quicky recovery back to using previous versions.

➤ Use cases

1.1 Library

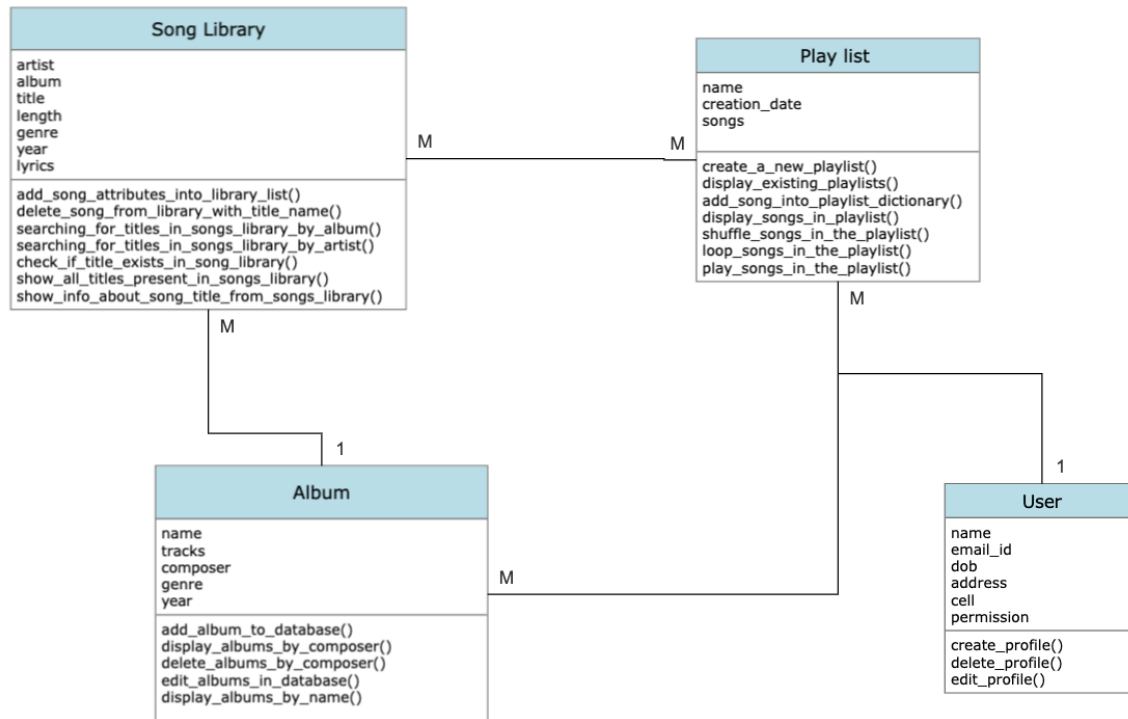
Use Case ID	001
Use Case Name	Library
Actors	Prafulla Chandra Munugoti (developer)
Description	This use case is for library level usage, the library is a set of all songs that are liked/added, a super set of all play lists. The player can add, delete, and search by song/artist from the library.
Preconditions	Python 3 installed Necessary Python packages installed
Postconditions	NA
Normal Flow	<ul style="list-style-type: none">• Start the tool• Login as a user• Add a song to the library• Display songs in the library• Add 10+ songs into the list• Display songs in the library• Search a song using song title• Play the searched song• Search songs by artist name• Play the searched song• Delete a song from the library• Delete all songs from the library
Alternative Flows	<ul style="list-style-type: none">• Search a song that does not exist• Search an artist that does not exist
Exception Flows	<ul style="list-style-type: none">• Delete a song that does not exist

1.2 Playlist

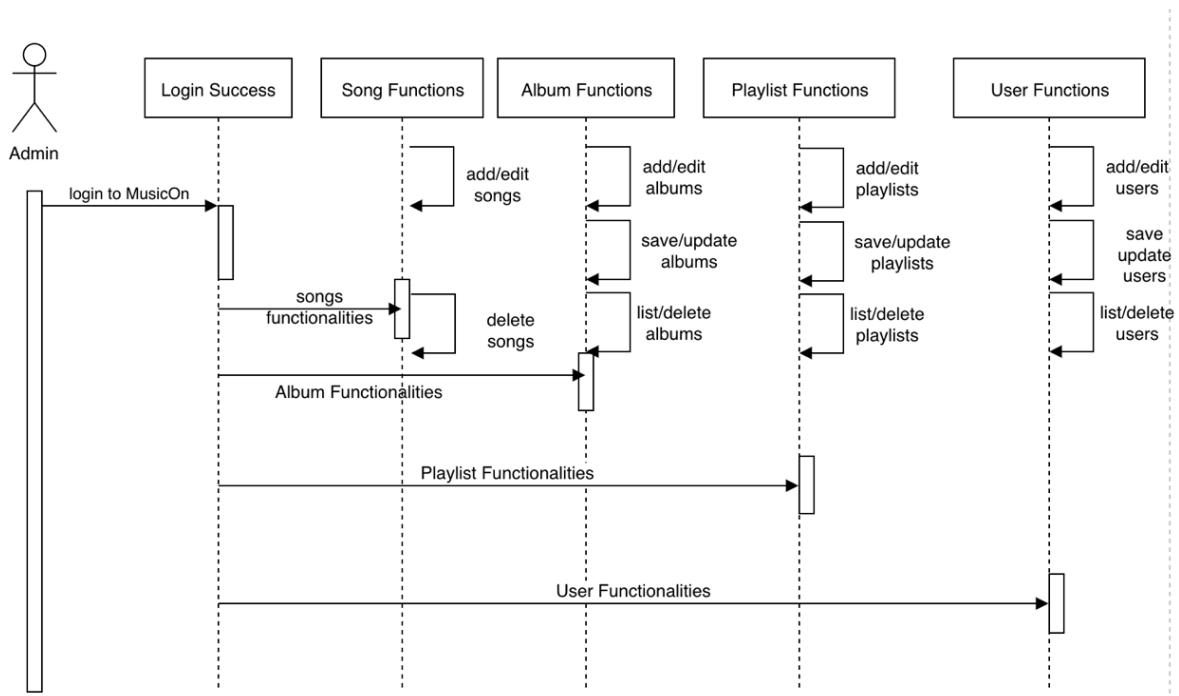
Use Case ID	002
Use Case Name	Playlist

Actors	Prafulla Chandra Munugoti (developer)
Description	This use case will be mainly for the play list that can be created, deleted, and store the songs and could be played in 3 different modes.
Preconditions	Python 3 installed Necessary Python packages installed
Postconditions	NA
Normal Flow	<ul style="list-style-type: none"> • Start the tool • Login as user • Create a play list • Add one song to the play list • Play the song from the list • Add 10+ music into the list • Play the list in normal mode • Play the list in shuffle mode • Play the list in loop mode • Delete a song from the play list • Delete the play list
Alternative Flows	<ul style="list-style-type: none"> • Play the song from outside the list
Exception Flows	<ul style="list-style-type: none"> • Play an empty list • Add one song to a list that does not create • Delete a song that does not exist

➤ Class Diagram



➤ Sequence Diagram:



➤ Use Case Diagram:

