
Software Requirements Specification

for

MMP

Version 1.0 approved

Prepared by D'Errico Vincenzo, Liguori Gianfranco

gruppo12

04/12/2020

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope.....	1
1.5 References	1
2. Overall Description.....	2
2.1 Product Perspective.....	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	3
2.4 Operating Environment.....	3
2.5 User Documentation	3
3. External Interface Requirements	3
3.1 User Interfaces.....	3
3.2 Software Interfaces	3
3.3 Communications Interfaces.....	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2	5
4.3 System Feature 3	6
4.4 System Feature 4	7

Revision History

Name	Date	Reason For Changes	Version
Liguori Gianfranco	02/12/20	First skeleton of MMP	0.9
D'Errico Vincenzo	05/12/20	Function update	1.0

1. Introduction

1.1 Purpose

MMP aim to bring an ecosystem where users can create playlists and listen to their favorite music in every place, also there is a reserved area for premium users where midi files can be uploaded and read after being reviewed and approved by the admin.

1.2 Document Conventions

NA

1.3 Intended Audience and Reading Suggestions

This SRS is intended to be read by both end users and developers. All the functions are described in the 4th paragraph, so an end user can read it for having an overall view of the project.

1.4 Product Scope

Streaming media is multimedia that is constantly received by and presented to an end-user while being delivered by a provider. The verb *to stream* refers to the process of delivering or obtaining media in this manner. Streaming is an alternative to file downloading, a process in which the end-user obtains the entire file for the content before listening to it (in this case). Through streaming, an end-user can use our media player to start playing digital audio content before the entire file has been transmitted.

1.5 References

At the following link is described the Java Server Page MVC model 2, which will be accomplished for the implementation of MMP

<https://www.javatpoint.com/MVC-in-jsp>

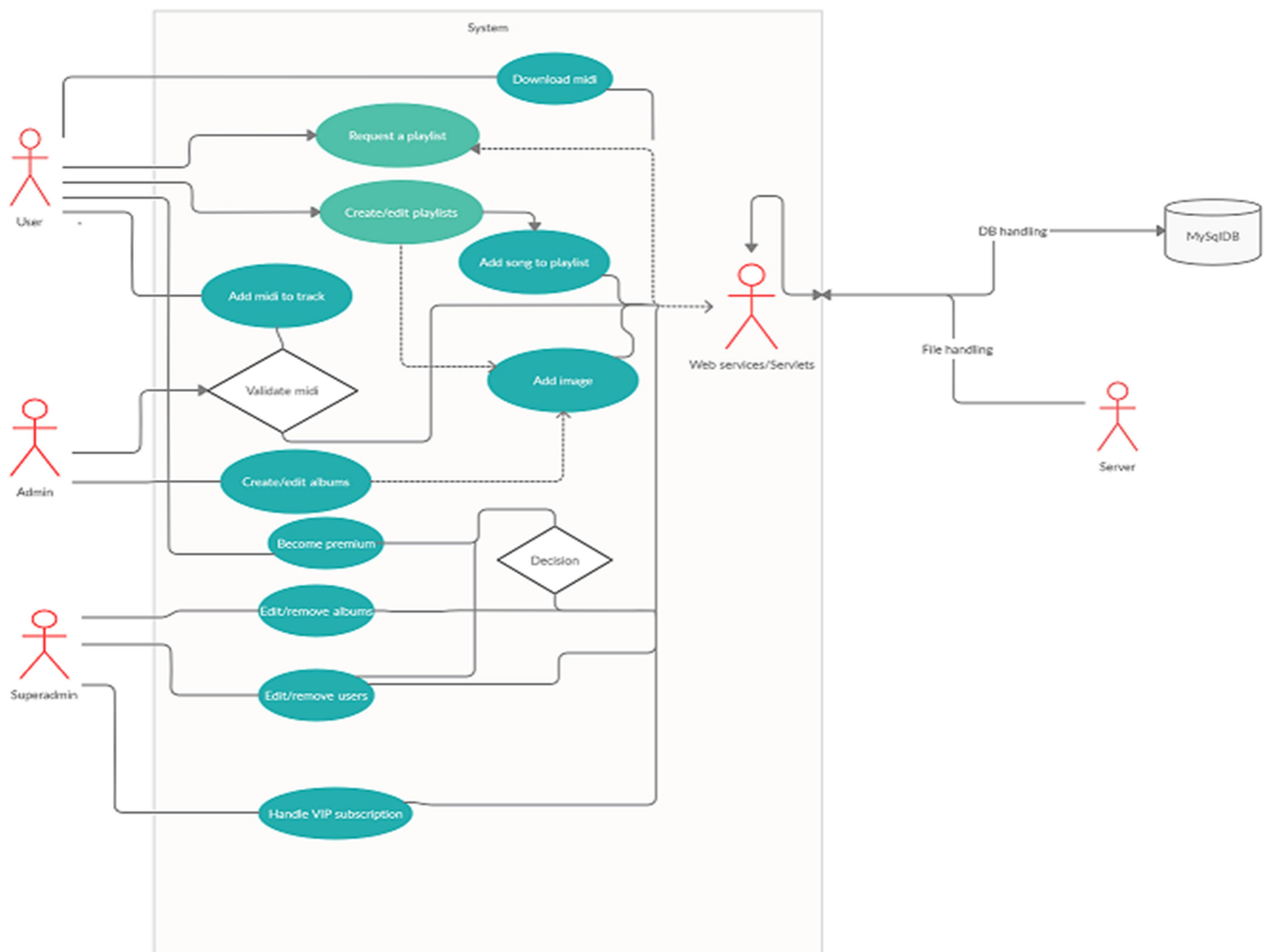
2. Overall Description

2.1 Product Perspective

MMP runs on web browsers and can be implemented on multiple devices using the provided API. The MMP system provides an end-user interface.

2.2 Product Functions

The product functions are described in use-cases.



2.3 User Classes and Characteristics

Users can stream music and suggest midi for already existing music. They have to buy a VIP subscription for having actually access to the midi files.

Admin can validate midi files suggested from the users and release albums and songs, they also can remove them and edit the data linked to them.

Superadmin is the owner of the project and can manipulate every uploaded data, he also handle the VIP subscriptions validating them if the user paid for them.

2.4 Operating Environment

The software will run on an Apache Tomcat server in combo with mysql for the DB management.

2.5 User documentation

MMP will be shipped with a short pdf manual.

3. External Interface Requirements

3.1 User Interfaces

The users have access to the system with a web browser or eventually an app. The web app will have a login/registration system and a simple interface for browsing into playlists or managing the system in every use case.

3.2 Software Interfaces

Servlets are used inside the system for handling DB uploads, a web service is instead provided to let developers release apps for having access to the stream from other devices (IoT).

3.3 Communication Interfaces

MMP can be executed in a web browser, the tomcat server will process the requests and will render the pages for the users.

4. System Features

The main goal of the project is to create a modular system for musing lovers, and so going through different devices is mandatory.

Logical components provided to users are:

- List of playlist
- Playlist of music
- Music player
- Search screen

An admin section is provided too, in that section the record label or the manager of an album rights can handle the upload and editing of the music available via the following interfaces:

- List of uploaded albums/add album and linked data (Cover)
- List of music in the album/add song and linked data (Metadata as title, author, composer)
- Validate midis uploaded from users

The server owner has the full management rights on every uploaded data, in that way he can also make new admins or remove them, remove album, songs and playlists.

4.1 System Feature 1

4.1.1 Playlist management.

User shall be able to create playlist, add his favourite songs to playlist and listen to it.

4.1.2 Stimulus/Response Sequence

The user will see a personal space, in which there are his playlists listed by name. Clicking on a playlist, will show the songs added in that playlist (nothing if is the first access).

Clicking on “Add Song”, a search bar in which to search songs by title will appear. User can add to playlist the searched song, with the apposite button.

Either, in the personal space, he can tap the “Create Playlist” button. Another page will appear, in which there are 2 inputs: the name of the playlist and an image. Clicking on “add”, the playlist name and its image are uploaded into database. Clicking on the title of a song, user can listen the selected song.

4.1.3 Functional Requirements

FR41-1 – Starting MMP

Input The user opens MMP and get logged in.

Action The system shall start the MMP application, read the database and detect the user type and privileges.

Output According to the user privileges a different home page is displayed.

If the user is a simple user his playlists are shown.

If the user is an admin a list of the album uploaded by that admin is shown

If the user is the owner a list of users is shown instead.

FR41-2 – Starting with an empty list of playlists

Input The user has not added any playlist yet.

Action The system shall initialize the database.
Output An empty list of playlists shall be displayed.

FR41-3 – Creating a new playlist

Input The user selects the “add playlist” button.
Action The system shall open the “add playlist” page, in which playlist name and its cover can be uploaded into DB.
Output The list of playlist visible to the user will have a new entry with the added object.

FR41-4 – Starting with an empty playlist.

Input The user selects a playlist for the first time.
Action The system loads the playlist from the DB.
Output A button for adding songs into the playlist is shown.

FR41-5 – Adding a new track to a playlist.

Input The user selects the symbol representing to add a new track.
Action The system shall open the search screen and prompt the user for input. When the search result is available and the user selects an item, the system adds the content to the chosen playlist.
Output A playlist with the new track-name displayed.

FR41-6 – Listen to a playlist.

Input The user chooses a playlist.
Action The system talks to the DB asking for the location of stored music files. The system shall start the player with the first track on the playlist.
Output The new track is being played. Information on the screen is updated.

FR41-7 – Delete a playlist.

Input The user chooses a playlist to be removed.
Action The system talks to the DB in an attempt to remove the chosen playlist. The system shall give a positive or negative output depending to the successful/unsuccessful DB operation.
Output The response is shown to the user.

4.2 System Feature 2

4.2.1 VIP area

A reserved area for VIP users where midis and musical scores are shown.

4.2.2 Stimulus/Response Sequence

Users can register as VIP, they will have a subscribe button with a form with the data to be fulfilled. Once his subscription is accepted he can see the midis and musical scores of the songs (if any) and request to add them uploading his versions. Admins will have a section where the midis and the musical scores uploaded can be validated or refused.
The owner validates the VIP subscriptions or discard them.

4.2.3 Functional Requirements

FR42-1 – VIP subscription(user side)

Input The user compiles the subscription form.

Action The system shall save the data in a temp recipe available to the owner.

Output The subscribe button is replaced by a message: "Your subscription has already been submitted, waiting for an answer from the admin".

FR42-2 – VIP subscription(admin side)

Input The admin checks and approves or declines the received subscription.

Action The system updates the user privileges in the DB.

Output The admin sees an updated list with the other pending subscriptions.

FR42-3 – Midi subscription(user side)

Input The VIP user sends a new midi file as a suggestion.

Action The system shall save the data in a temp recipe available to the admin of the song.

Output An error page is shown if the file can't be uploaded, the playlist space is shown instead if the operation is successful.

FR42-4 – Midi subscription(admin side)

Input The admin checks and approves or declines the received file.

Action The system updates the DB.

Output The admin sees an updated list with the other pending subscriptions.

FR42-5 – Midi download

Input A VIP user chooses a midi file to download it.

Action The system asks the DB for the location of the stored file.

Output The chosen midi is provided to the user for the download.

FR42-6 – Musical scores subscription(user side)

Input The VIP user sends a new musical score file as a suggestion.

Action The system shall save the data in a temp recipe available to the admin of the song.

Output An error page is shown if the file can't be uploaded, the playlist space is shown instead if the operation is successful.

FR42-7 – Musical scores subscription(admin side)

Input The admin checks and approves or declines the received file.

Action The system updates the DB.

Output The admin sees an updated list with the other pending subscriptions.

FR42-8 – Musical scores download

Input A VIP user chooses a musical score file to download it.

Action The system asks the DB for the location of the stored file.

Output The chosen musical score is provided to the user for the download.

4.3 System Feature 3

4.3.1 Admin area

A control panel for admins where they can handle the stored songs and midis.

4.3.2 Stimulus/Response Sequence

Admins can see the album already uploaded by themselves. They can add new albums, add cover to them and add songs into every album. Every song is linked to an existing album and the admin should provide the data for every song he's uploading.

4.3.3 Functional Requirements

FR43-1 – Starting with an empty list of albums

Input The admin has not added any album yet.

Action The system shall initialize the database.

Output An empty list of albums shall be displayed.

FR43-2 – Starting with an empty album.

Input The admin selects an album for the first time.

Action The system loads the album from the DB.

Output A button for adding songs into the album is shown.

FR43-3 – Creating a new album

Input The admin selects the “add album” button.

Action The system shall open the “add album” page, in which an album name and its cover can be uploaded into DB.

Output The album is added in the list and the updated list is shown.

FR43-4 – Adding a new track to an album.

Input The admin selects the symbol representing to add a new track.

Action The system shall open the “add song” page, in which a song and linked metadata can be uploaded into DB.

Output An album with the new track-name displayed.

4.4 System Feature 4

4.4.1 Owner/superadmin area

A control panel for the owner of the system where he can handle all the stored files, he has full access to the DB(editing/removing users, albums, songs...).

4.4.2 Stimulus/Response Sequence

The owner chooses the data to be edited or removed, he can promote new admins and VIP users, see the privileges of every user and remove users. He can do the same operations with albums. If an admin is removed by the owner all the songs and albums from that admin will be removed.

4.4.3 Functional Requirements

FR44-1 – Showing users list

Input The owner login.

Action The system shall open a page and ask the DB for a list of all users and their data .

Output The list of users is shown with the managing buttons for every entry of the list.

FR44-2 – Showing album list

Input The owner chooses an admin to edit albums uploaded by that admin.

Action The system shall open a page and ask the DB for a list of albums matching that admin .

Output The list of albums is shown with the management buttons for every entry of the list.

FR44-3 – Editing/removing users

Input The owner chooses an user to edit or remove the relative data and once he has defined how edit the chosen user he press a confirm button.

Action The system shall update the DB and give a positive/negative answer.

Output The list of users is updated with the given modifications.

FR44-4 – Editing/removing album

Input The owner chooses an album to edit or remove the relative data and once he has defined how edit the chosen album he press a confirm button.

Action The system shall update the DB and give a positive/negative answer.

Output The list of albums is updated with the given modifications.