

CS.CR4SB

Software Requirements Specification

Online Music Streaming Platform

Version: 1.0

Date: (10/27/2018)

Team members:

- **Alexandru Tatulescu**
- **Marius Popa**

Table of Contents

1. Introduction:

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. The Overall Description:

2.1 Product Perspective:

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communications Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8 Site Adaptation Requirements

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions and Dependencies

2.6 Apportioning of Requirements

3. Specific Requirements:

3.1 External interfaces

3.2 Functions

3.3 Performance Requirements

3.4 Logical Database Requirements

3.5 Design Constraints:

3.5.1 Standards Compliance

3.6 Software System Attributes:

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

3.7 Organizing the Specific Requirements:

3.7.1 System Mode

3.7.2 User Class

3.7.3 Objects

3.7.4 Feature

3.7.5 Stimulus

3.7.6 Response

3.7.7 Functional Hierarchy

3.8 Additional Comments

1. Introduction

The Software Requirements Specification is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested . Its primary purpose is to provide a clear and descriptive “statement of user requirements” that can be used as a reference in further development of the software system.

This document is broken into a number of sections used to logically separate the software requirements into easily referenced parts.

This Software Requirements Specification aims to describe the Functionality, External Interfaces, Attributes and Design Constraints imposed on Implementation of the software system described throughout the rest of the document.

1.1 Purpose

Defining and describing the functions and specifications of the Online music streaming platform is the primary goal of this Software Requirements Specification (SRS).

This Software Requirements Specification illustrates, in clear terms, the system’s primary uses and required functionality.

The intended audience of this document are the professor Dr. Mocanu Mihai as well as the other students attending this project course.

1.2 Scope

The software system being produced is called Online Music Streamig Service or OMSS. It is being produced for persons interested to bring their music everywhere via the Internet.

This system is designed to “provide automation support” for the process of listening music online and facilitate the world of on-demand music streaming.

This system is largely cross-platform and is available to anyone that has a working Internet Connection.

The system will be run on a central server with each user having a remote user interface through a web browser to interact with it.

The Online music streaming service will allow any user to create an account to become a customer. The customer, through the process of account creation, will have the option to become a member of the site. The system will allow customers to browse, search, select, and add songs to a playlist.

The OMSS also allows an admint to manage the inventory with full create, retrieve, update and delete (CRUD) functionality with regards to the songs in the system. The OMSS has full email capabilities; the automated email functionality will be used to send recomandations to members of the system.

The system also will not allow users to retrieve passwords or edit their user details

1.3 Definitions, Acronyms, and Abbreviations.

OMSS	Online Music Streaming Service
Song	A short poem or other set of words set to music or meant to be sung.
Button	A user interface element that allows a User to inform the system that he/she selected a particular item
Checkbox	A user interface element that allows a User to inform the system that he/she selected a particular item.
CRUD	Create, Retrieve, Update, Delete
Customer	A person that is a user of the system but has created an account
Inventory	Refers to the collection of music.

Session	The time which a User is actively using the system
SRS	Software Requirements Specification
User	The person who operates the software product
Playlist	A playlist is a list of video or audio files that can be played back on a media player either sequentially or in a shuffled order.
Manager	A single person that has the ability to create, retrieve, update and delete items.

1.4 References

- <https://www.digitaltrends.com/music/best-music-streaming-services/>
- <https://www.spotify.com/ro/>
- <https://en.wikipedia.org/wiki/Spotify>

1.5 Overview

This Software Requirements Specification document is divided in to multiple subsections.

The first section includes explanations of the Purpose, Scope and Organization of the document. The first section also handles the description of projects pecific words, acronyms and abbreviations that will be used in the document.

The second section of the document is separated into the following five different sections, each detailing specific details of system uses and their corresponding actions:

- Product Perspective
- Product Functions
- User Characteristics
- Constraints
- Assumptions and Dependencies
- Apportioning of Requirements.

The third section is an enumerated listing of all of the requirements described for this system.

The fourth section encompasses all of the Use-case, Sequence, State and Class diagrams that model the system.

2 Overall Description

This section includes details about what is and is not expected of the OMSS system in addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the OMSS system.

2.1 Product Perspective

OMSS is an online streaming music service which supports a number of functions for the consumer.

The service must be available to anyone that has an Internet Connection and as such must work correctly in any web browser(desktop and mobile).

There are no hardware or software requirements beyond these including, but not limited to, memory or specific software packages that need to be utilised nor software packages that need not be utilised.

2.2 Product Functions

OMSS will provide a number of functions, each is listed below:

- Maintain records for many user.
- A customer has a username (unique across all users), password (no restrictions), email address (no restrictions).
- Anyone may sign up for a user account.
- Show a listing of available songs and playlists.
- Allow user to log in and out of the system.
- Choose what you want to listen to with **Browse** and [Search](#).
- Build collections of music.
- Create playlist with our online music streaming service.
- Users will be logged out if inactive for 30 minutes.

2.3 User Characteristics

The typical OMSS user is simply anyone that has access to the Internet and a web browser. It is assumed that the user is familiar enough with a computer to operate the browser, keyboard and mouse and is capable of browsing to, from and within simple websites.

2.4 Constraints

Security is a concern for this system as it is a web based application.

The database should not store passwords in plain text and there should be also a password recovery feature . As such, the system may not work correctly in cases when security is a concern. These cases include those listed above in addition to lack of an encrypted connection when sending credit card information and forcing users to use “strong” passwords.

A strong password is a password that meets a number of conditions that are set in place so that user's passwords cannot be easily guessed by an attacker. Generally, these rules include ensuring that the password contains a sufficient number of characters and contains not only lowercase letters but also capitals,numbers, and in some cases, symbols.

2.5 Assumptions and Dependencies

Client:

We have assumed that all of the computer systems are in proper working condition, have a proper functional Internet Connection and that the user is capable of operating these system's basic functions including but not limited to being able to power on the system, login and open either Internet Explorer, Chrome, Mozilla Firefox, and navigate the browser to the address of this OMSS website.

Provider:

We have assumed that the OMSS will be running on a properly working web server and database system with an Internet connection that allows this system to perform all communications with clients.

2.6 Appportioning of Requirements

As stated, security is a concern of this project. As such, it is essentially to focus on the encryption of personal user data, prevent unauthorized login attempts, or any other concern of this nature.

Additionally, the system is not responsible for the following:

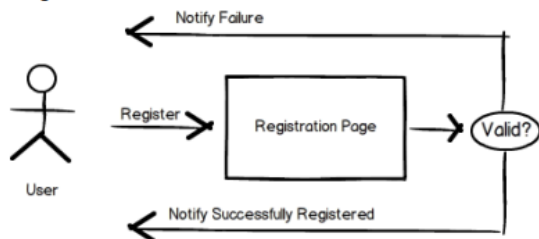
- Verifying that credit card information is valid
- Verifying the email address provided by a user

Additionally, the system may need to later be extended to provide additional functions.

3. Specific Requirements

Register:

Diagram



Description:

Primary Actor	User
Goal in Context	Purpose of this feature is to register user to the system.
Trigger	User wants to register to the system

Normal Flow of Events:

1. User opens the registration page

2. User specifies his information
3. System validate the specified information
4. User is registered to the system

Alternative Event Flow 1

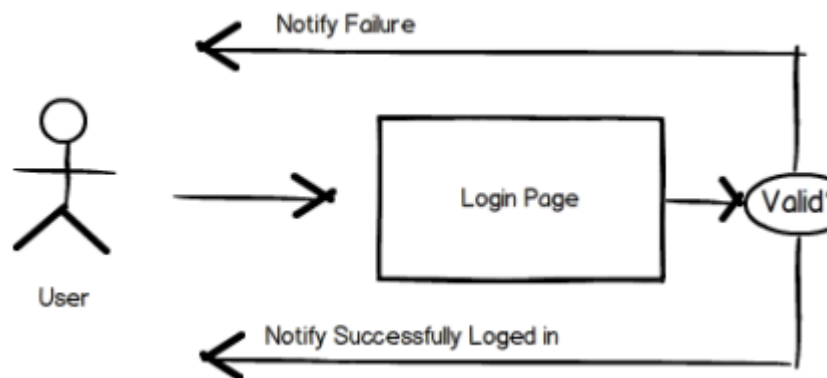
4. User can not registered to the system due to inappropriate information

Alternative Event Flow :

1. User can not registered to the system due to inappropriate information

Login:

Diagram



Description

Primary Actor	User
Goal in Context	Purpose of this feature is to login to the system with user credentials in order to use system.
Trigger	User wants to login to the system

Normal Flow of Events

1. User opens the login page
2. User tries to login to the system with his credentials
3. System validate the specified information
4. User is logged into the system

Alternative Event Flow

4. User cannot logged into the system due to incorrect credentials

Logout:

Use Case:	Logout
-----------	--------

Actors: Type:	User, System Primary and essential
Description:	The customer or manager will have the option to logout and if that user is inactive for a given amount of time then that user should be logged out by the system automatically.
Includes:	None
Use-Cases:	User must have completed the Log In use case

Browse Playlists:

Use Case: Actors: Type:	Browse playlists User Primary and Essential
Description:	All the songs listed in playlists. List should be sorted by title, artist, etc.

Create Playlist:

Use Case: Actors: Type:	Create playlist User Primary and Essential
Description:	Allows users to create different playlists.

Delete a Playlist:

Use Case: Actors: Type:	Delete a playlist User Primary and Essential
Description:	Allows users to delete different playlists.

Add songs to Playlist:

Use Case: Actors: Type:	Add songs to Playlist User Primary and Essential
Description:	Allows users to add songs to a previous created playlist

Delete a songs from Playlist:

Use Case: Actors: Type:	Delete a song from Playlist User Primary and Essential
Description:	Allows users to delete song to a previous created playlist

Add songs to Favorites:

Use Case: Actors: Type:	Add songs to Favorites User Primary and Essential
Description:	Allows users to add different songs to a favorites songs.

Delete a songs from Favorites:

Use Case: Actors: Type:	Delete a songs from Favorites User Primary and Essential
Description:	Allows users to delete different songs to a favorites songs.

Change playlists name:

Use Case: Actors: Type:	Change playlists name User Primary and Essential
Description:	Allows users to change playlist name already created.

Change theme application:

Use Case: Actors: Type:	Change theme application User Primary and Essential
Description:	Allows users to change theme application

Revert to default theme application:

Use Case: Actors: Type:	Revert to default theme application User Primary and Essential
Description:	Allows users to change theme application

3.3. Non-functional Requirements

In this section, last group of the requirements which is non-functional requirements will be explained in detail. Non-functional requirements include performance requirements, security requirements and portability requirements.

3.3.1 Performance Requirements

Since this software is going to web – based, it does require a powerful server machine with high band internet access. Server machine should have a powerful CPU and high speed internet access so that it can handle multiple users at the same time. Another performance requirement is the storage space. Higher storage space means more user and bigger workspace per user so higher the storage, better the performance. Performance requirement by the user side is, web application should be developed as a lightweight web app so that it can work on almost any platform even with slower internet connections. Expected number of simultaneous user should be at least 100. System should be able to deal with 100 users at the same time. Also database of the system should handle at least a thousand of users at any periods.

3.3.2 Security Requirements

Since this software will be hosted on cloud server, all the user data will be kept on the cloud server. Product should be able to protect privacy of user data. Workspace of the user should only be accessed through user own credentials and any other user should not be able to access to the user private data. Since execution will also be done in the machine in the cloud, user should be restricted in terms of user rights. User should only access to his own workspace and should not access to any other workspace with the programs they run on the cloud. Also rights of the user should be restricted so that user can not harm to system by the programs they run or by the commands they run on terminal. Since all the data will be transferred on the web, system should also use an encryption and decryption mechanism only intended user can decode the data and work on the data.

3.3.3 Portability Requirements

Main purpose of developing web-based IDE is to improve the portability of software development process. To improve portability, software should run on variety of platforms and variety of connection speeds. As explained in the performance requirements section, software should be lightweight so that it can run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase. Such as using JavaScript and HTML5 instead of Apache Flex. 30 Portability also means running on most number of different platform without an additional effort. To achieve this, web application should

be developed by using the common technologies and tools which are provided by all common web browsers and operating system such as HTML5, JQuery etc.