

# Software Requirements Specification Document

**MMAD Music**

**1.0**

**September 17<sup>th</sup>, 2024**

**MMAD Productions**

**Mickael Agustin, Dan Rascop, Mikey Lupa,  
Aayaan Shaikh**

Submitted in partial fulfillment of the requirements of  
IT 326 – Principles of Software Engineering

# Table of Contents

## TABLE OF CONTENTS2

<b>1. INTRODUCTION</b>	<b>3</b>
1.1 PURPOSE	3
1.2 SCOPE	3
1.3 DEFINITIONS	3
1.4 OVERVIEW	3
<b>2. GENERAL DESCRIPTION</b>	<b>3</b>
2.1 PRODUCT PERSPECTIVE	3
2.2 USER CHARACTERISTICS	4
2.3 SYSTEM ENVIRONMENT	4
2.4 ASSUMPTIONS AND DEPENDENCIES	4
<b>3. SPECIFIC REQUIREMENTS</b>	<b>4</b>
3.1 FUNCTIONAL REQUIREMENTS	4
3.1.1 <i>Search Song</i>	4
3.1.2 <i>Search Album</i>	5
3.1.3 <i>Search Artist</i>	5
3.1.4 <i>Add Favorite Album</i>	5
3.1.5 <i>Remove Favorite Album</i>	5
3.1.6 <i>Add Favorite Song</i>	6
3.1.7 <i>Remove Favorite Song</i>	6
3.1.8 <i>Add Favorite Artist</i>	6
3.1.9 <i>Remove Favorite Artist</i>	6
3.1.10 <i>Delete Review</i>	7
3.1.11 <i>Send Friend Request</i>	7
3.1.12 <i>Remove Friend</i>	7
3.1.13 <i>Accept Friend Request</i>	7
3.1.14 <i>Decline Friend Request</i>	8
3.1.15 <i>Navigate to Friend's Profile</i>	8
3.1.16 <i>Navigate to Home Page</i>	8
3.1.17 <i>Create Account</i>	8
3.1.18 <i>Login</i>	9
3.1.19 <i>Change Password</i>	9
3.1.20 <i>Write Review</i>	9
3.1.21 <i>Navigate to Reviews Page</i>	10
3.1.22 <i>Navigate to Friends list</i>	10
3.1.23 <i>Search User</i>	10
3.2 NON-FUNCTIONAL REQUIREMENTS	11
3.2.1 <i>Performance</i>	11
3.2.2 <i>Reliability</i>	11
3.2.3 <i>Security</i>	11
3.2.4 <i>Portability</i>	11

# 1. Introduction

## 1.1 Purpose

The document's purpose is to provide a detailed description of the music app system “MMAD Music”. It will explain the scope of functionality that the system will have and what user interactions will be possible.

## 1.2 Scope

This software system will allow users to create accounts with usernames and passwords. It will store login information and allow users to alter their existing login by resetting their password. This system is intended to allow its users to search for, rate, review, and favorite songs, albums, and artists. Each user will be able to cultivate their own music libraries and display their favorite song, album, and artist on their profile. In addition, users can friend other users to view their profiles and create a shared playlist of mutually saved music. The software will not play the music locally, but rather just display the information regarding the record. The software will allow users to recommend and share music with their friends, expanding their musical horizons and fostering social connection around music.

## 1.3 Definitions

Home Page – Page that will display the user’s favorited music elements

Friends page - Page that will display the user’s friended users

Reviews page - Page that will display the user’s reviews

## 1.4 Overview

The next section of this document is primarily written with the developers in mind. It is intended to provide a technical description of the system's functionality and expectations.

# 2. General Description

*This section of the SRS should describe the general factors that affect 'the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.*

## 2.1 Product Perspective

This system is unique to other systems on the market because it allows users to rate and combine music with other users unlike other systems which do similar things but restrict their functions to other media. Spotify, the closest popular system comparatively like ours, does not have a social network or the ability to filter music based on multiple user's common taste. Letterbox'd is a similar app with much of the same functionality except that is solely used for film.

## 2.2 User Characteristics

The system's user demographics are those capable of using a terminal and want to rate and share their music with their friends. Anyone who wants to connect to others through the world of music.

## 2.3 System Environment

This system will be run in a terminal environment on a PC using Java Development Kit (JDK). It will have a MySQL database hosted on Amazon's Relational Database Service (RDS).

## 2.4 Assumptions and Dependencies

- The system assumes that Java Development Kit (JDK) and all the libraries included will be installed and functional on the machine running the software. Compatibility between versions will be checked during implementation.
- The system assumes the MySQL database hosted on Amazon's Relational Database Service (RDS) is online, accessible by the software application and capable of returning queries and storing data. It also assumes that the free service tier with Amazon will continue to meet our development needs and that the authentication credentials are accessible and valid.
- The system is designed to run on a Windows terminal environment and as such assumes access to terminal commands and all the permissions necessary to run a Java application on Windows.
- The system assumes that the Spotify API will be publicly accessible for the various search functions, that the rate limits are high enough for our software's functionality, and that the data is returned in an expected format.
- Lastly, the system assumes proper internet connectivity between the application and the online MySQL RDS server, that the server is going to be online when the application is run, and that the application will be able to both read and write to and from the database.

## 3. Specific Requirements

### 3.1 Functional Requirements

#### 3.1.1 Search Song

- a. Description – *The system will prompt user for the track title that the user would like to find and return a list of ten most relevant albums based on the entered string.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the 'Search for song' command.*
- d. Conditions
  - i. Pre – *Search bar must not be empty.*
  - ii. Post – *Returns a list of ten most relevant tracks.*

### 3.1.2 Search Album

- a. Description – *The system will prompt the user for the album title that the user would like to find and return a list of ten most relevant albums based on the entered string.*
- b. Actor(s) – *User*
- e. Trigger – *User enters the ‘Search for album’ command*
- c. Conditions
  - i. Pre – *Search bar must not be empty.*
  - ii. Post – *Returns a list of ten most relevant albums.*

### 3.1.3 Search Artist

- a. Description – *The system will prompt for artist/band name that the user would like to find and return a list of ten most relevant artist/bands based on the entered string.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Search for artist’ command.*
- d. Conditions
  - i. Pre – *The Profile has no album displayed.*
  - ii. Post – *Returns a list of ten most relevant artists/bands.*

### 3.1.4 Add Favorite Album

- a. Description – *The system will allow the user to add the selected album to their profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Add to favorite album’ command while an album is selected from the search results.*
- d. Conditions
  - i. Pre – *The Profile has no favorite album displayed.*
  - ii. Post – *The system will store the album information to be displayed on the user’s profile.*

### 3.1.5 Remove Favorite Album

- a. Description – *The system will allow the user to remove the displayed album from the user’s profile.*
- b. Actor(s) – *User*
- b. Trigger – *User enters the ‘Remove favorite album’ command.*
- d. Conditions
  - i. Pre – *The Profile has an album displayed.*
  - ii. Post – *The album information will be removed from the user’s profile.*

### 3.1.6 Add Favorite Song

- a. Description – *The system will allow the user to add the selected song to their profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Add to favorite song’ command while a song is selected from the search results.*
- d. Conditions
  - i. Pre – *The Profile has no song displayed.*
  - ii. Post – *The system will store the track information to be displayed on the user’s profile.*

### 3.1.7 Remove Favorite Song

- a. Description – *The system will remove the displayed track from the user’s profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Remove favorite artist’ command.*
- d. Conditions
  - iii. Pre – *The Profile has a song displayed.*
  - iv. Post – *The song information will be removed from the user’s profile.*

### 3.1.8 Add Favorite Artist

- a. Description – *The system will allow the user to add the selected artist to their profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Add to favorite artist’ command while a song is selected from the search results.*
- d. Conditions
  - i.Pre – *The Profile has no artist displayed.*
  - ii.Post – *The system will store the artist/band information to be displayed on the user’s profile.*

### 3.1.9 Remove Favorite Artist

- a. Description – *The system will remove the displayed artist from the user’s profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Remove favorite artist’ command.*
- d. Conditions
  - i. Pre – *The Profile has an artist displayed.*
  - ii. Post – *The artist’s information will be removed from the user’s profile*

### **3.1.10 Delete Review**

- a. Description – *The system will allow a review to be deleted from the user’s reviews page*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Delete review’ command on a selected review.*
- d. Condition
  - i. Pre – *The Profile has at least one review.*
  - ii. Post – *The review is deleted from the user’s reviews page.*

### **3.1.11 Send Friend Request**

- a. Description – *The system will allow the user to send a friend request to another user which can then be accepted and declined to allow the user to view the profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Send friend request’ command on a selected user.*
- d. Conditions
  - i. Pre – *The user is not already friends with the target user. The target user has an existing account.*
  - ii. Post – *The system will send a friend request message to the target user.*

### **3.1.12 Remove Friend**

- a. Description – *The system will allow a user to remove another user as a friend and disallow any future friend requests.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the Remove friend’ command on a selected friend.*
- d. Conditions
  - i. Pre – *The target user is a friend; the target user account exists.*
  - ii. Post – *The target user will be unable to send a friend request or view the user’s profile.*

### **3.1.13 Accept Friend Request**

- a. Description – *The system allows the user to “accept” a friend request on their profile. Accepting the friendship will allow the two users to view each other’s profiles. Rejecting a friend request will not allow the requesting user to view the user’s profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Accept friend request’ command on a selected friend request.*

- d. Conditions
  - i. Pre – *A user that is not already a friend sent a friend request.*
  - ii. Post – *The user becomes friends with the user who sent the friend request.*

#### **3.1.14 Decline Friend Request**

- a. Description – *The system allows the user to “decline” a friend request on their profile. Declining the friendship will not the two users to view each other’s profiles.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘Decline friend request’ command on a selected friend request.*
- d. Conditions
  - i. Pre – *A user that is not already a friend sent a friend request.*
  - ii. Post – *The friend request is deleted.*

#### **3.1.15 Navigate to Friend’s Profile**

- a. Description – *The system will display the user’s friend’s profile including their favorite song, favorite album, and favorite artist. The system will also allow the user to navigate the friend’s Reviews page from this profile.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the ‘View friend’s profile’ command on a selected friend.*
- d. Conditions
  - i. Pre – *The user has at least one friend.*
  - ii. Post – *The system will display the selected friend’s profile.*

#### **3.1.16 Navigate to Home Page**

- a. Description – *The system will allow the user to navigate to the user’s own home page.*
- b. Actor(s) – *User*
- c. Trigger – *User logs into the system or selects the ‘Navigate to home page’ command.*
- d. Conditions
  - i. Pre – *The user has a valid account and is logged in.*
  - ii. Post – *The system will display the user’s existing profile.*

#### **3.1.17 Create Account**

- a. Description – *The system will allow the user to enter a username and password to create a new account. The user will also be asked to answer a security question for account recovery.*
- b. Actor(s) – *User*



- c. Trigger – *User selects “Create Account” option in the starting menu.*
- d. Conditions
  - i. Pre – *Username must not already be in use by another account.*
  - ii. Post – *Stores new username, password, and security question in database.*

### **3.1.18 Login**

- a. Description – *The system will allow the user to log in using their username and password upon startup.*
- b. Actor(s) – *User*
- c. Trigger – *User enters “Login” command option in start menu.*
- d. Conditions
  - i. Pre – *User must have already created a valid account.*
  - ii. Post – *Logs user into account if credentials are correct. If credentials are incorrect, the user will be notified and allowed another chance to enter the correct credentials.*

### **3.1.19 Change Password**

- a. Description – *The system will allow the user to change their password.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the “Change password” command while signed in.*
- d. Conditions
  - i. Pre – *User must have already created a valid account.*
  - ii. Post – *User’s account password will be updated in the database.*

### **3.1.20 Write Review**

- a. Description – *The system will allow the user to write a new review of a song or album which includes a 10-point rating.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the “Create new review” command on the selected song or album.*
- d. Conditions
  - i. Post – *Adds the new review to the user’s Reviews page.*

### 3.1.21 Navigate to Reviews Page

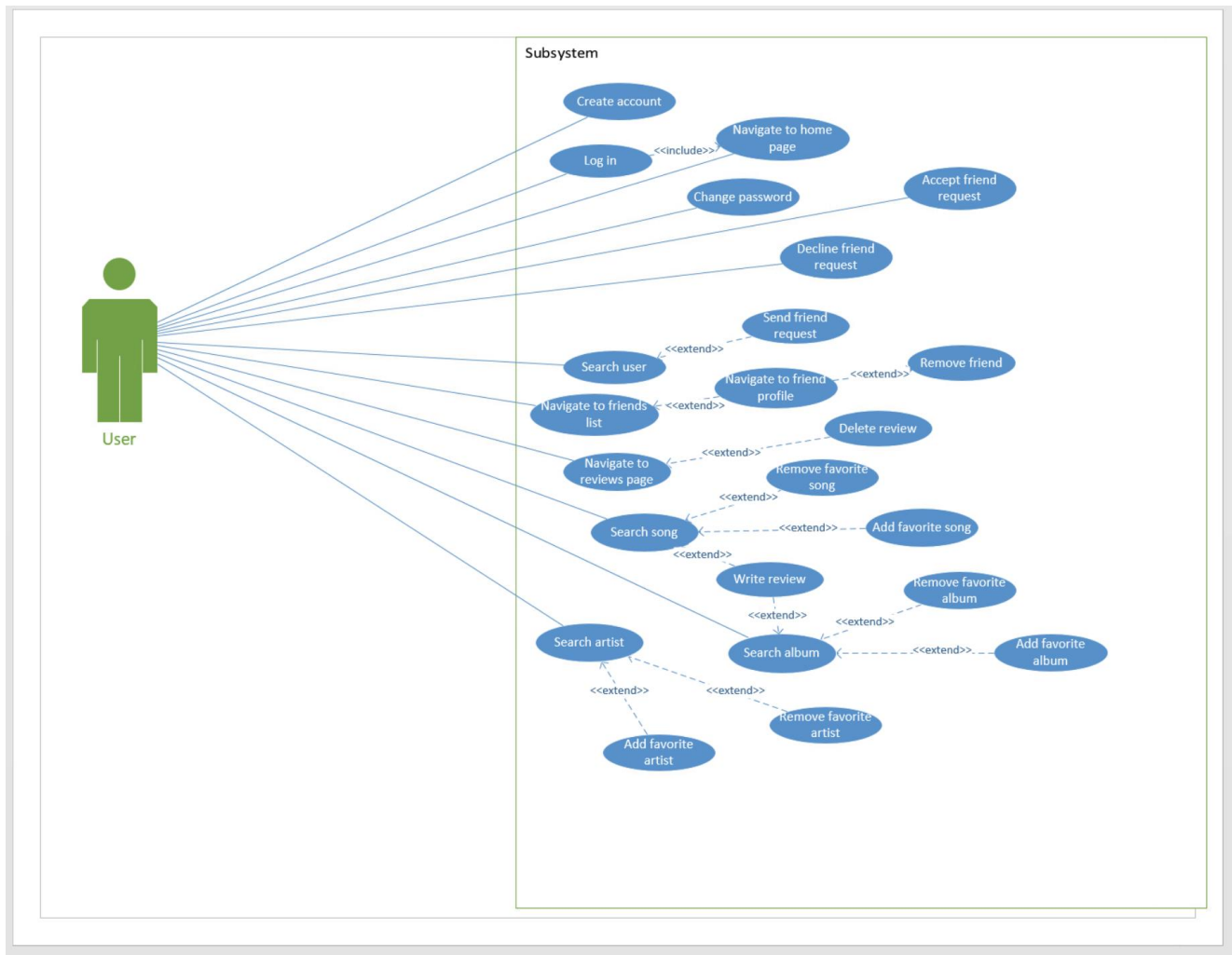
- a. Description – *The system will allow the user to navigate to the reviews page where they can view their reviews.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the 'Navigate to the reviews page' command.*
- d. Conditions
  - i. Pre – *User must be on their own profile.*
  - ii. Post – *User is navigated to their own reviews page.*

### 3.1.22 Navigate to Friends list

- a. Description – *The system will allow the user to navigate to the friends page. The friends page will list all the profiles of other users that are friends with the user.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the 'Navigate to the friends page' command.*
- d. Conditions
  - i. Pre – *User must be on their own profile.*
  - ii. Post – *User is navigated to their own friend's page.*

### 3.1.23 Search User

- a. Description – *The system will prompt the user for the username that the user would like to find and return a list of ten most relevant usernames based on the entered string.*
- b. Actor(s) – *User*
- c. Trigger – *User enters the 'Search user' command.*
- d. Conditions
  - i. Post – *Returns a list of ten most relevant usernames.*



## 3.2 Non-Functional Requirements

### 3.2.1 Performance

*This system should respond to every user command in under 10 seconds.*

### 3.2.2 Reliability

*This system should not lose user login information or music data.*

### 3.2.3 Security

*This system should not allow profile visibility for non-friends or allow any modification to any profile other than the user's.*

### 3.2.4 Portability

*This system should be functional on all devices capable of running JAVA code, with access to the database and a terminal.*