

Software Requirements Specification

for

Spotify Playlist Synchronize

Prepared by:

Nilusche Liyanaarachchi

FH Aachen University of Applied Sciences, Germany
Software Engineering

Contents

1	Introduction	2
1.1	Disclaimer	2
1.2	Purpose	2
1.3	Intended Audience	2
1.4	Product Scope	2
1.5	Definitions, Acronyms, and Abbreviations	2
1.6	References	3
2	Overall Description	4
2.1	User Classes and Characteristics	4
2.2	Product Perspective	4
2.2.1	System Interface	4
2.2.2	User Interface	4
2.2.3	Operating Environment	4
2.3	Product functions	5
2.4	Constraints	5
3	System Features and Requirements	6
3.1	Functional Requirements	6
3.1.1	Create new Playlists	6
3.1.2	Add new songs to playlist	7
3.1.3	Delete Playlists	7
3.1.4	Delete Playlists	8
3.1.5	Synchronize Playlist onto Spotify	9
3.1.6	Enable Spotify Authorization	9
3.2	Non Functional Requirements	10
3.2.1	Performance Requirements	10
3.2.2	Software Quality Attributes	10
3.3	Extended Requirements	11
3.4	External Interface Requirements	11

3.4.1 Hardware Interfaces	11
Appendices	12
A Appendix: Firebase Document Structure	13
B Appendix: Sample Screenshots of Product as is	14

Chapter 1

Introduction

1.1 Disclaimer

This document is prepared or accomplished by Nilusche Liyanaarachchi (Student Assistant) in his own personal capacity.

This fictional product exists for educational grounds and is not intended for commercial usage.

1.2 Purpose

The purpose of this document is to present a detailed description of the requirements and features of the Spotify Playlist Synchronize Application

The software is used to learn about Oauth of Spotify and Firebase.

1.3 Intended Audience

This document does not have a particular audience in mind. Ideally this document should address fellow software engineers that have experience in reading/writing Software Requirements Specifications (SRS).

The Software is intended for users that would like to share playlist on a platform and synchronize them with Spotify.

1.4 Product Scope

The purpose of this clone is to provide the essential functionalities of an Exchange Platform

1.5 Definitions, Acronyms, and Abbreviations

Term/ Acronym / Abbreviation	Expansion / Description
Spotfiy Playlist Synchronize	SPS
GUI	Graphical User Interface

1.6 References

1. IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

Chapter 2

Overall Description

2.1 User Classes and Characteristics

Different Users of the Systems should be able to have different permissions to access all functionality of the app

- Unauthorized User
- Authorized User
- Spotify OAuth User

2.2 Product Perspective

2.2.1 System Interface

Since this product will only have a web-interface, appropriate Web-Browsers like Firefox, Chrome, Edge should be used in their latest stable version.

2.2.2 User Interface

The application GUI provides menus, buttons, containers, grids allowing for easy control by a mouse

2.2.3 Operating Environment

- Operating System: Windows
- Vue3 capable Browser

2.3 Product functions

- Create new Playlists
- Add songs to Playlist
- Delete Playlists
- Synchronize Playlists to Spotify
- Login/Logout

Product functions are refined in Section 3. **Functional Requirements**

2.4 Constraints

The application is developed only using the Vue-Firebase stack.

Chapter 3

System Features and Requirements

3.1 Functional Requirements

These specific functional requirements will be presented as User story and Use case to guide the developer during their implementation.

3.1.1 Create new Playlists

Title: Create new Playlists	Priority: high
As a general user I want to create a new playlist so that i can add songs to them	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Clicking the menu "Create new Playlist"

Procedure:

1. User fills in fields: "Playlist title" (required), "Playlist description" (required), Image field" (optional)
2. User clicks on Button "Create Playlist"
3. System generates new database record for playlist
4. System uploads Image to database if given
5. System redirects user to homepage

Postconditions:

1. A new playlist record in database

Exceptions:

1. Database error
2. Required fields error

3.1.2 Add new songs to playlist

Title: Add new songs to playlist	Priority: high
As a user I want to add songs to my playlist so that i can change my playlist	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized
2. User already created Playlist to add Songs onto

Trigger: Clicking the menu "Add a new Song" on Playlist Page

Procedure:

1. User searches for song keyword and clicks button "Search Track"
2. System shows number of tracks for keyword
3. User selects song or multiple songs
4. User clicks "Add"
5. System adds Song-IDs to list of songs in database
6. System updates Visuals

Postconditions:

1. A modified Playlist record in database

Exceptions:

1. Database error

3.1.3 Delete Playlists

Title: Delete Playlists	Priority: high
As a general user I want to delete Playlist so that i can start over	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized
2. User already created Playlist

Trigger: Clicking the menu "Create new Playlist"

Procedure:

1. User fills in fields: "Playlist title" (required), "Playlist description" (required), Image field" (optional)
2. User clicks on Button "Delete"
3. System deletes Database record
4. System deletes associated Picture if given
5. System redirects user to homepage

Postconditions:

1. Delete Playlist Record in Database

Exceptions:

1. Database error

3.1.4 Delete Playlists

Title: Delete Playlists	Priority: high
As a general user I want to delete Playlist that i can start over	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized
2. User already created Playlist

Trigger: Clicking the menu "Create new Playlist"

Procedure:

1. User fills in fields: "Playlist title" (required), "Playlist description" (required), Image field" (optional)
2. User clicks on Button "Delete"
3. System deletes Database record
4. System deletes associated Picture if given
5. System redirects user to homepage

Postconditions:

1. Delete Playlist Record in Database

Exceptions:

1. Database error

3.1.5 Synchronize Playlist onto Spotify

Title: Synchronize Playlist onto Spotify	Priority: high
As a general user I want to synchronize playlists i like to spotify so that i can start listening to them	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized
2. User has authorized Spotify

Trigger: Clicking the menu "Add Playlist to Spotify"

Procedure:

1. System sends request to Spotify API
2. Spotify API adds Playlist to User's Account

Postconditions:

1. New Playlist in User's Spotify account

Exceptions:

1. Request error

3.1.6 Enable Spotify Authorization

Title: Enable Spotify Authorization	Priority: high
As a general user I want to connect my spotify account to the platform so that i can start synchronizing playlists	

Actors: User with appropriate permissions

Preconditions:

1. User needs to be authorized

Trigger: Clicking the menu "Spotify Login"

Procedure:

1. System sends OAuth Request to Spotify API
2. User is prompted to allow Token generation
3. User clicks on "Agree"
4. Spotify API sends Auth-Token to Platform

Postconditions:

1. New Authorization Token

Exceptions:

1. Request error
2. Disagreement

3.2 Non Functional Requirements

3.2.1 Performance Requirements

1. The service needs to be highly available
2. Acceptable latency to generate the timeline is 200ms

3.2.2 Software Quality Attributes

- Availability: The Sorting Algorithms need to be immediately sorted after clicking the button
- Correctness: The Applet should ensure a algorithmics correctness
- Usability: The interface should be easy to learn without a tutorial and allow users to accomplish their goals without errors.

3.3 Extended Requirements

1. Preview feature on songs of the platform
2. Other forms of Login (Google, Apple)
3. Popularity Leader board
4. Voting Feature

3.4 External Interface Requirements

3.4.1 Hardware Interfaces

- A browser which supports Vue3

Appendices

Appendix A

Appendix: Firebase Document Structure

```
createdAt: 26 July 2022 at 22:49:08 UTC+2
description: ""
filePath: null
▶ songs: [{uri: "spotify:track:0mBP...}
title: "Trends"
url: null
userId: "VsOKOPz8oDRJCiaDZ8caceaoHd73"
userName: "root"
```

Figure A.1: Firebase Document Structure of "Playlists"

Appendix B

Appendix: Sample Screenshots of Product as is

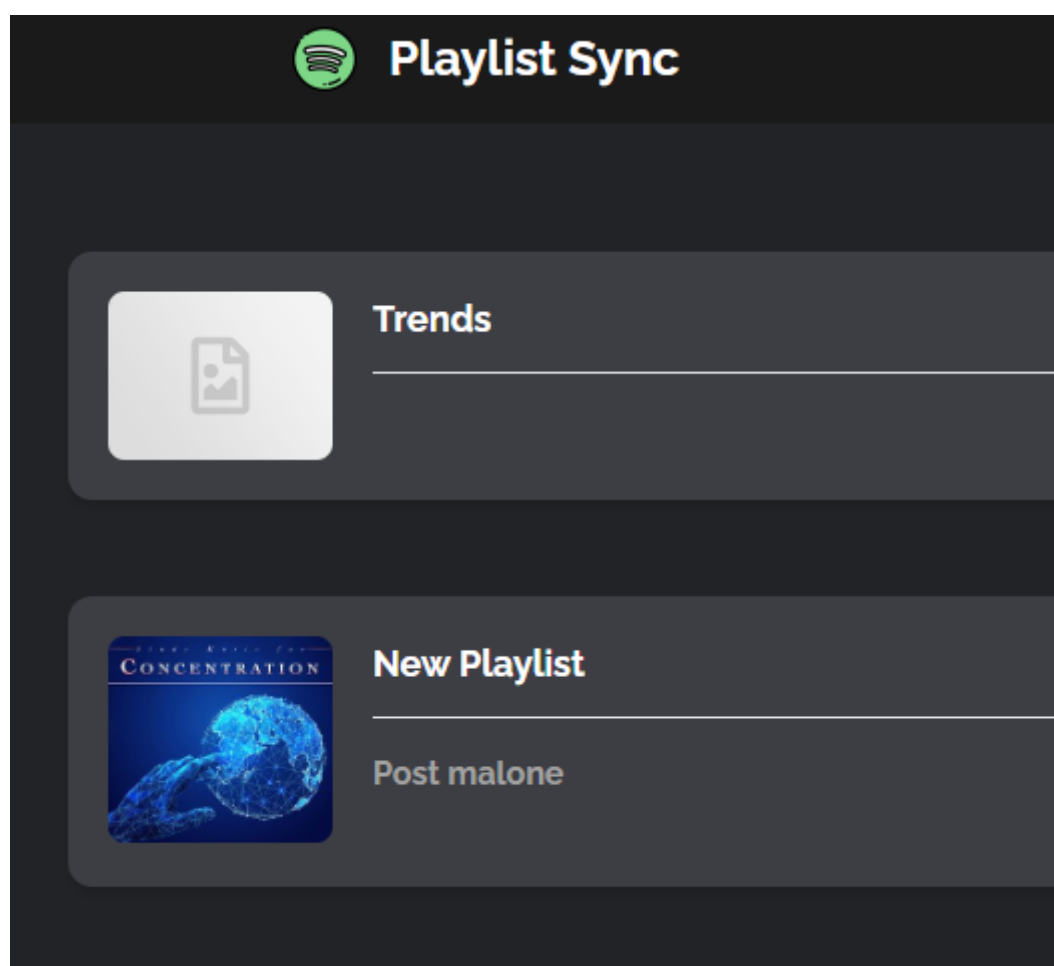
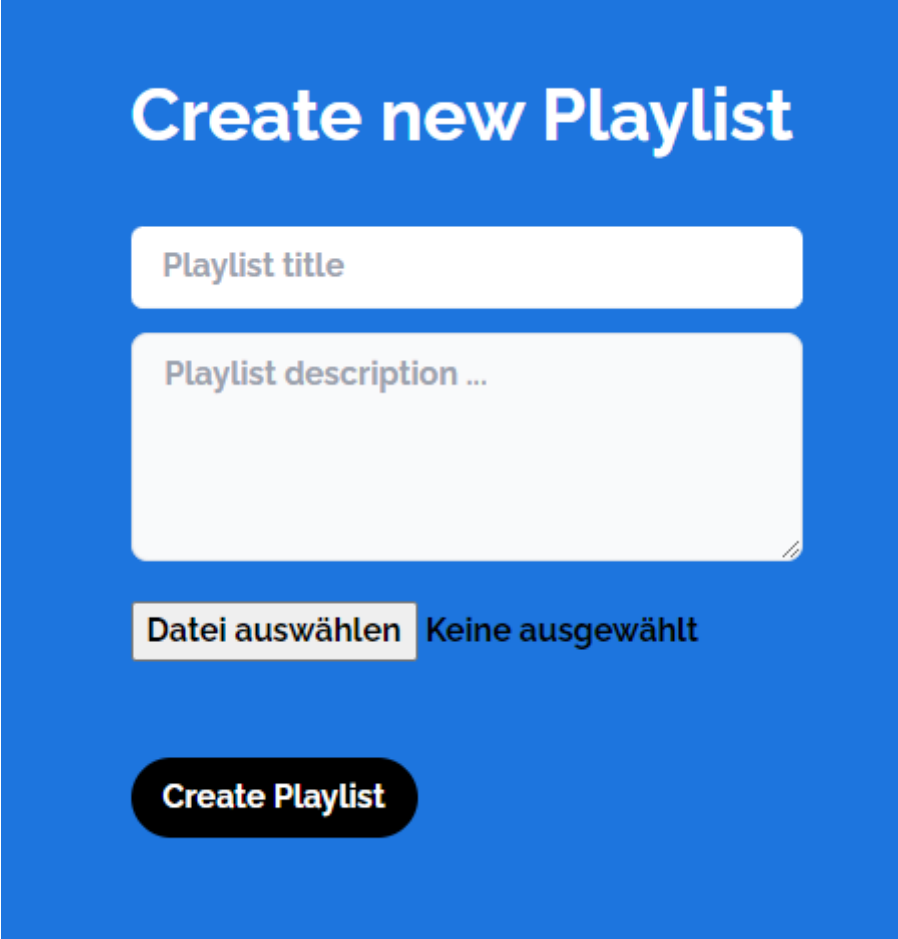


Figure B.1: Homepage View



A screenshot of a 'Create new Playlist' form on a blue background. The form consists of a title input field, a larger description text area, a file selection button labeled 'Datei auswählen', and a status indicator 'Keine ausgewählt'. At the bottom is a large black button with white text that says 'Create Playlist'.

Create new Playlist

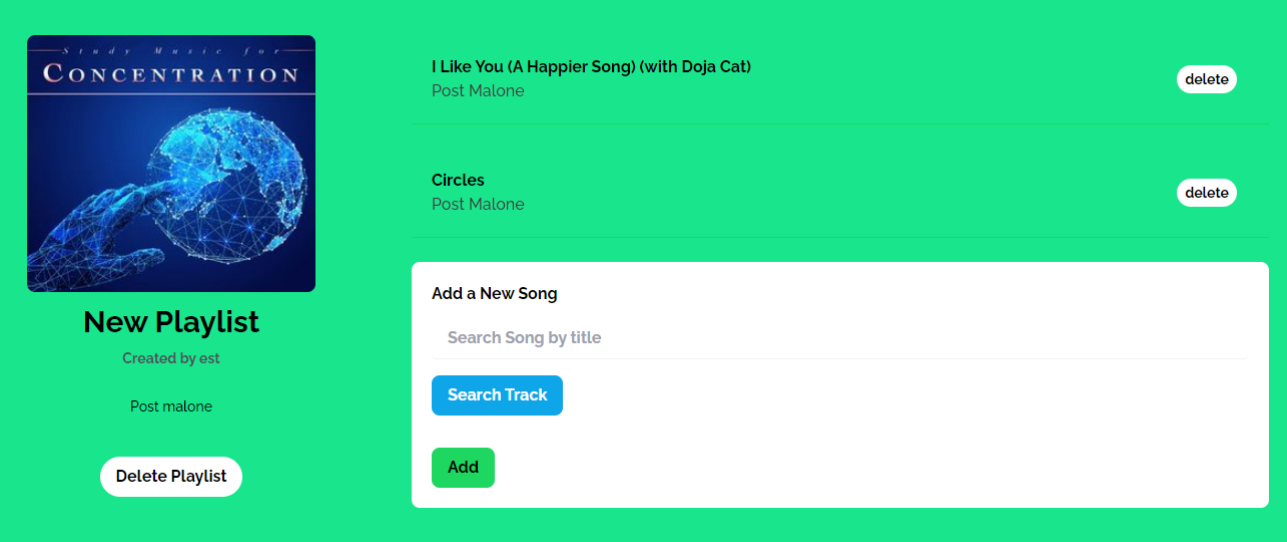
Playlist title

Playlist description ...

Datei auswählen Keine ausgewählt

Create Playlist

Figure B.2: Create Form View



A screenshot of a 'New Playlist' interface on a green background. On the left is a playlist cover image titled 'CONCENTRATION' with a hand holding a globe. Below the image, it says 'New Playlist', 'Created by est', 'Post malone', and a 'Delete Playlist' button. On the right, there is a list of songs: 'I Like You (A Happier Song) (with Doja Cat)' by Post Malone and 'Circles' by Post Malone, each with a 'delete' button. Below the list is a section titled 'Add a New Song' with a search bar labeled 'Search Song by title', a blue 'Search Track' button, and a green 'Add' button.

Study Music for
CONCENTRATION

New Playlist

Created by est

Post malone

Delete Playlist

I Like You (A Happier Song) (with Doja Cat)
Post Malone delete

Circles
Post Malone delete

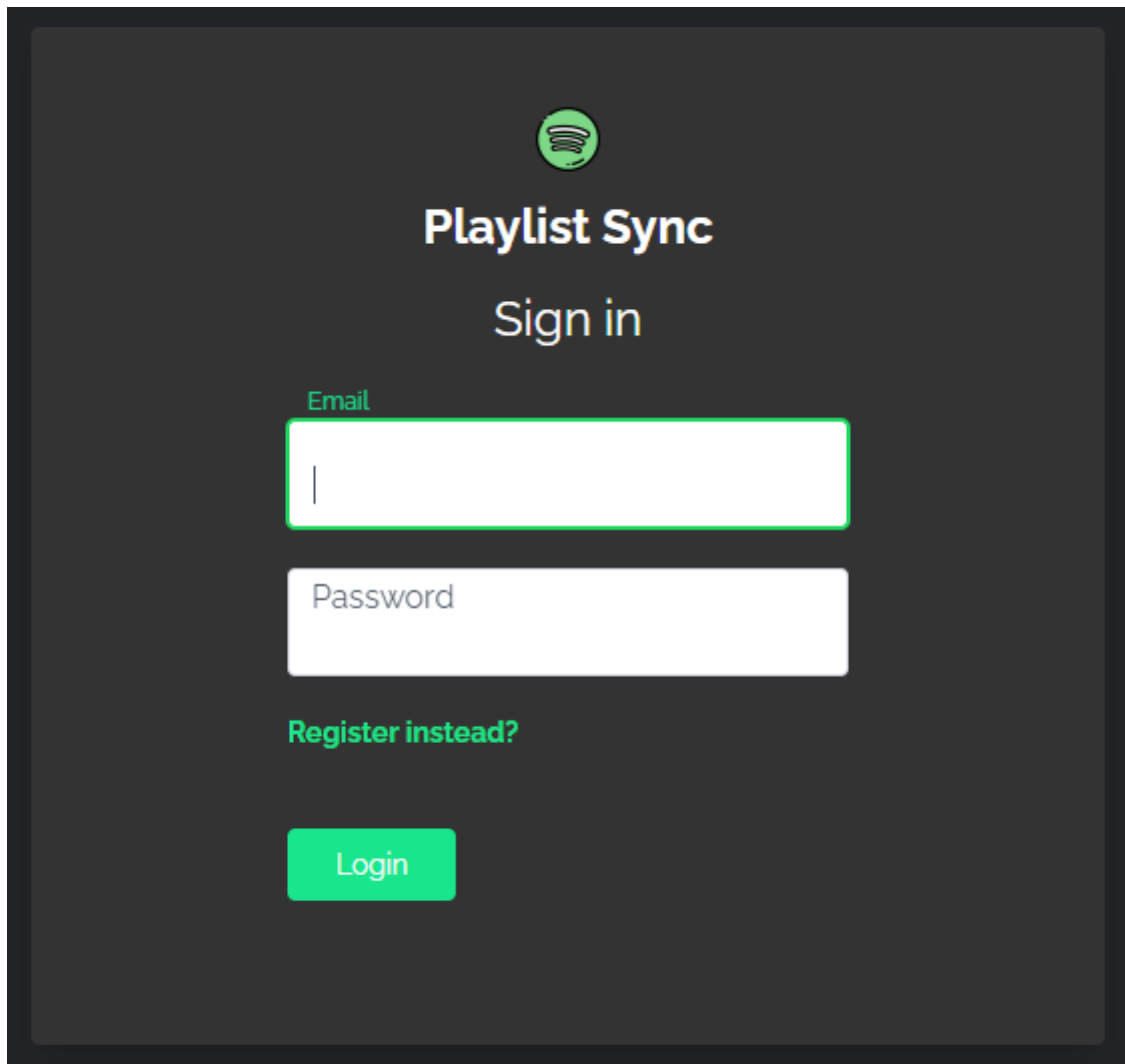
Add a New Song

Search Song by title

Search Track

Add

Figure B.3: Playlist View



The image shows a login screen for 'Playlist Sync'. At the top center is a green circular logo with a white stylized 'S' and a play button icon. Below the logo, the text 'Playlist Sync' is displayed in a bold, white, sans-serif font. Underneath that, 'Sign in' is written in a smaller, white, sans-serif font. There are two white input fields: the first is labeled 'Email' in green text above it and contains a single vertical line; the second is labeled 'Password' in gray text above it. Below the password field, the text 'Register instead?' is shown in green. At the bottom center is a green rectangular button with the word 'Login' in white.

Figure B.4: Login View