



CSC 431

Playlist Finder and Music Social Media

Software Requirements Specification (SRS)

Team 17

Allegra Papera

Project Management, Product Design

Grant Yaniv

Full Stack Developer

Yasir Nemat

Full Stack Developer

Version History

Version	Date	Author(s)	Change Comments
1.0	2/23/2021	Papera, A., Yaniv, G., Nemat, Y.	
2.0	3/9/2021	Papera, A., Yaniv, G., Nemat, Y.	Added Constraints, and fixed TA recommendations from last draft

Table of Contents

1. System Requirements
 - 1.1 Functional Requirements
 - 1.1.1 Spotify Login
 - 1.1.2 Music Discovery Preferences
 - 1.1.3 Music Social Media User Profile Pages
 - 1.1.4 Social Media Post Feed
 - 1.1.5 Music Discovery Preferences
 - 1.2 Non-Functional Requirements
 - 1.2.1 Playlist Generation Speed
 - 1.2.2 Application Downtime
 - 1.2.3 Account Security
2. System Constraints
 - 2.1 Tool Constraints
 - 2.1.1 Slack
 - 2.1.2 Microsoft OneDrive
 - 2.1.3 Figma
 - 2.1.4 Visual Studio
 - 2.1.5 Spotify API
 - 2.2 Language Constraints
 - 2.2.1 HTML/CSS
 - 2.2.2 JavaScript
 - 2.2.3 Node.js
 - 2.3 Platform Constraints
 - 2.3.1 iOS
 - 2.3.2 Android
 - 2.3.3 Web Application
 - 2.4 Hardware Constraints
 - 2.4.1 Device Accessibility
 - 2.4.2 Team Productivity Tools
 - 2.5 Network Constraints
 - 2.5.1 Internet Requirements
 - 2.5.2 Website Server Requirement
 - 2.6 Deployment Constraints
 - 2.6.1 Semester Variability
 - 2.7 Transition & Support Constraints
 - 2.7.1 End of Semester
 - 2.8 Budget & Schedule Constraints
 - 2.8.1 Project Slideshow Deadline
 - 2.8.2 Project Package Deadline
 - 2.9 Miscellaneous Constraints
 - 2.9.1
 - a. Requirements Modeling
- 3.1.1
 - b. Evolutionary Requirements
- 4.1 Functional Requirements
 - 4.1.1 Web App Implementation
- 4.2 Non-Functional Requirements
 - 4.2.1 Web App Implementation

Table of Tables

<Generate table here>

Table of Figures

<Generate table here>

51.1. System Requirements

51.1 1.1 Functional Requirements

< List all functional requirements in the following example format >

1.1.1 1.1.1 Spotify Login

Title	Login to Spotify account
Description	Users will enter their Spotify login information (username and password) in order to give our application permission to access the user's Spotify account data.
Priority	2
Precondition(s)	User needs to have a Spotify account
Basic Flow	Two text boxes: one for username, one for password. Once the text boxes are filled out, user can press submit button.
Postconditions(s)	User is logged into their account on our app via Spotify, and now our app has access to their account data.
Use Case Diagram	<Link or number, if present>

1.1.2 1.1.2 Music Discovery Preferences

Title	Form to indicate music discovery preferences
Description	This will be a survey in which users can indicate their preferences for music that they want exposure to in order for our app to follow new playlists on their behalf. Users can enter genre, tempo, danceability, etc. These options will reflect Spotify's categorizations.
Priority	0
Precondition(s)	User must be logged into Spotify
Basic Flow	Users will interact with this like a survey. This will be an entry form with multiple choice buttons for each question. Most questions will allow for multiple answers.
Postconditions(s)	Our application will be able to generate a sample consisting of playlists that reflect the preferences of our users. From this sample, our application will be able to pull a smaller selection to follow on behalf of the user that week.
Use Case Diagram	<Link or number, if present>

1.1.3 1.1.3 Music Social Media User Profile Pages

Title	User profile pages
Description	Expanded user Spotify profiles that display the user's previous posts, favorite artists and songs, as well as who and what they currently enjoy listening to from the playlists that our application followed for them.
Priority	1

Precondition(s)	The user must login to this application with their Spotify login information
Basic Flow	Users will be able to interact with their own profile, containing all of their past posts, and can access other user's profile pages that contain their posts.
Postconditions(s)	Ability to access and view other user's profile pages and interact with all of their previous posts.
Use Case Diagram	<Link of number, if present>

1.1.4 1.1.4 Social Media Post Feed

Title	Social Media Post Feed
Description	A central social media post option where users can create posts for their followers on Spotify, and view posts created by people they follow.
Priority	3
Precondition(s)	Formation of expanded user profiles that allow for posts to be stored, and interaction with other users.
Basic Flow	Users will be able to open a textbox, in which they can fill with their thoughts, recommendations, attach playlists, artists, or songs, and share these custom-made posts with their friends and followers.
Postconditions(s)	These posts created will then be posted to the central feed for the followers of the user and added to the user's main profile page.
Use Case Diagram	<Link or number, if present>

1.1.5 1.1.5 Notifications Page

Title	Results/notifications page informing users of the playlists that were followed on their account
Description	This page will inform users of the playlists that the app followed for them. This will have a history of notifications that updates users with new playlists. They can choose to keep or delete such playlists (like Instagram notifications).
Priority	1
Precondition(s)	A precondition of this requirement is that users fill out a music discovery preference form.
Basic Flow	Once the playlists have been discovered and followed for users, they will all appear in a central page that notifies the user.
Postcondition(s)	The user will then be able to view each playlist and decide if they want to add them to their profile, or delete them from this result page.
Use Case Diagram	Form to indicate music discovery preferences

51.2 1.2 Non-Functional Requirements

< List all non-functional requirements in the following example format >

1.2.1 1.2.1 Playlist Generation Speed

Title	Playlist Generation Speed
Description	This non-functional requirement is the speed at which playlists should be generated for users once they fill out their preference forms. With advanced and efficient search algorithms, this process should only take at most 10 seconds to generate all possible playlists.
Priority	3
Applicable FR(s)	1.1.4 Music Discovery Preferences, since this is the speed at which this will occur for the user.

1.2.2 1.2.2 Application Downtime

Title	Application Downtime
Description	This is the downtime that the application will have each month for server and website related fixes, which will not exceed more than an hour per month.
Priority	4
Applicable FR(s)	N/A

1.2.3 1.2.3 Account Security

Title	User Account Security
Description	Each user account will be individualized and therefore, they need to be encrypted for the safety of our users. There will be inactivity timeouts so users are logged out of their accounts when they aren't using them.
Priority	1
Applicable FR(s)	1.1.1 Spotify Login

52.2. System Constraints

52.1 2.1 Tool Constraints

< List all tool constraints in the following example format >

2.1.1 2.1.1 Creation and Productivity Tools

Title	Slack
Description	Slack will be used to facilitate communication between team members, as well as between team members and advisors (TAs, Professor)
Priority	2

2.1.2 Collaboration Tools

Title	Microsoft OneDrive
Description	Microsoft OneDrive will be used to allow virtual collaboration on shared documents.
Priority	2

2.1.3 Design Tools

Title	Figma
Description	Used for building wireframes for the application.
Priority	1

2.1.4 Integrated Development Environment (IDE)

Title	Visual Studio (or other IDE)
Description	Used for writing code.
Priority	0

2.1.5 Spotify

Title	Spotify API
Description	Used in our code to facilitate interaction between our application and Spotify user data.
Priority	0

52.2 2.2 Language Constraints

< List all language constraints in the following example format >

2.2.1 2.2.1 Programming Languages and Frameworks

Title	HTML/CSS
-------	----------

Description	Used alongside JavaScript to create the music preference form, as well as the social media platform.
Priority	0

2.2.2

Title	JavaScript
Description	Used alongside HTML/CSS to create the music preference form. Will be used with Node.js and the Spotify API to collect information on users to discover preferences and post information onto the social media platform
Priority	0

2.2.3

Title	Node.js
Description	Required for the use of Spotify API, which will be used to gather information on user's music preferences for use in posting onto the social media platform
Priority	0

52.3 2.3 Platform Constraints

< List all platform constraints in the following example format >

2.3.1 2.3.1 IOS

Title	iOS
Description	In the application's first version, it will be functional on Apple iOS devices (iPhone, iPod).
Priority	0

52.4 2.3.2 Android

Title	Android
Description	In the application's first version, it will be functional on Android devices (smartphones and tablets).
Priority	0

2.4.1 2.3.3 Web

Title	Web Application
Description	In the application's second version, we will implement all of its functional and non-functional requirements as a web app, in addition to its pre-existing mobile apps on iOS and Android.
Priority	1

52.5 2.4 Hardware Constraints

< List all hardware constraints in the following example format >

2.5.1 2.4.1 Device Accessibility

Title	Internet Connected Device
Description	Hardware wise, any device that wants to access this web application will be required to have internet connection and access to the web, since this will be a person-to-person social media type application.
Priority	1

2.5.2 2.4.2 Team Productivity Tools

Title	Laptop or Desktop
Description	To implement this application, any team member will be required to have access to a laptop or desktop
Priority	0

52.6 2.5 Network Constraints

< List all network constraints in the following example format >

2.6.1 2.5.1 Internet Requirements

Title	Basic Internet Connection
Description	To access this web application, a basic internet connection of at least 10 Mbps download/upload will be required.
Priority	2

2.6.2 2.5.2 Website Server Requirement

Title	Network and Server Bandwidth
Description	The servers that this application is running on will limit the number of users that are able to access the application. We will have to host the application on servers large enough to handle our daily active users.
Priority	0

52.7 2.6 Deployment Constraints

< List all deployment constraints in the following example format >

2.7.1 2.6.1 Semester Variability

Title	Semester Length and Difficulty
Description	Depending on how long the semester is, and how difficult it is for everyone in the team, that will dictate how and when this application gets deployed.
Priority	3

52.8 2.7 Transition & Support Constraints

< List all transition & support constraints in the following example format >

2.8.1 2.7.1 End of Semester

Title	End of Life
Description	Since the lifespan of this project is only the Spring 2021 semester, support for this application will stop after May 2021.
Priority	5

52.9 2.8 Budget & Schedule Constraints

< List all budget & schedule constraints in the following example format >

2.9.1 2.8.1 Project Slideshow Deadline

Title	April 20th Project Slideshow Deadline
Description	The team must complete a slideshow presenting their project by April 20 th .
Priority	0

2.9.2 2.8.2 Project Package Deadline

Title	May 1 Project Package Deadline
Description	The team must push the entirety of the project (outlines, wireframes, presentations, etc) to Github by May 1.
Priority	0

52.102.9 Miscellaneous Constraints

< List all miscellaneous constraints in the following example format >

2.10.12.9.1

Title	
Description	
Priority	<Priority from 0 (highest) – 5 (lowest)>

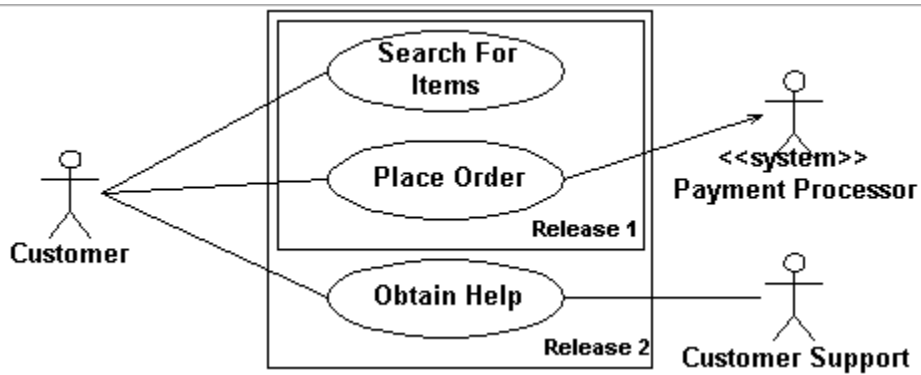
2.10.22.9.2

Title	
Description	
Priority	<Priority from 0 (highest) – 5 (lowest)>

53.3. Requirements Modeling

< List all Use-case diagrams for the functional requirements in the following format >

3.1.1 Requirement Title



54.4. Evolutionary Requirements

54.1 4.1 Functional Requirements

< List all functional requirements in the following example format >

4.1.1 4.1.1 Web App Implementation

Title	Web app implementation
Description	Implementation of all of version 1's functional requirements, now accessible through browser as a web application.
Priority	2
Precondition(s)	Launch of version 1 (mobile apps for iOS and Android devices).
Postconditions(s)	We acquire userbase through App Store and Google Play ratings. Then we will make the platform accessible through web browser.
Use Case Diagram	<Link or number, if present>

54.2 4.2 Non-Functional Requirements

< List all non-functional requirements in the following example format >

4.2.1 4.2.1 Requirement Title

Title	
Description	
Priority	
Applicable FR(s)	