

Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2020

Project author		
No	Pseudonym	Face-to-face/ online
1	singing-addict	face-to-face

Project name	MusicShare
--------------	------------

1. Short project description (Business needs and system features)

Imagine what it would be like if all music artists could gather together on a social network! **MusicShare** is a place where they can share own songs, add others' songs on their own playlists to listen to and develop their song ideas with the help of another user. And all accessible to whoever you are - no need to be a celebrity or to have your own career in the public eye. One can be a kid trying to learn the piano, a dreamer who wants to show others their music, or an advanced musician. It doesn't matter, one can just connect and start learning and sharing feedback with the others.

The system will be developed as a *Single Page Application (SPA)* using **React.js** as front-end, and **Node.js + express** as backend technologies. Each view will have a distinct URL, and the routing between pages will be done client side using **React Router**. The backend will be implemented as a **REST/JSON API** using JSON data serialization. There will be also a real-time event streaming from the server to the web client using **Socket.IO** and **Server Sent Events (SSE)/WebSocket**. The main user roles are:

- *Anonymous User* – can view others' pages, play their public songs and comment under their public song ideas
- *Registered User* – can upload own songs and song ideas, create playlists, listen to others' songs and comment under song ideas

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
2.1. Browse playlists	Users can browse albums and regular playlists in order to explore.	<i>All users</i>
2.2. Register	Users can register in the website by providing the following details: username, email, password, age, favourite genres, genres they're working in	<i>Anonymous User</i>
2.3. Login	Users can log in the website by providing their email and password	<i>Anonymous User</i>

2.4. Edit User Data	Users can view and edit their personal User Data.	<i>Registered User</i>
2.5. Add my own song	Users can upload the songs they've finished working on by providing the following information: title, genre, audio file, cover (an image). The song goes into the "Own songs" section.	<i>Registered User</i>
2.6. Add/Edit/Resolve my own song idea	Users can add their own song idea (a voice memo of a song that is only partially created) by providing the following information: title, genre, cover (optional), audio file, and the help they need with it (title, lyrics, melody, cover, rhythm or everything). The song idea goes into the "Song Ideas" section. Users can upload updates on their song ideas. When the idea is done, the User can click on a "resolve" button and the commenting section under the idea becomes no longer active.	<i>Registered User</i>
2.7. Create/Edit a regular playlist	Users can create their own collection of songs (playlist) belonging only to other artists by clicking on a drop-down menu (or just a button if no more options come to mind) on the selected song in their album, choose "add to playlist" and select playlist to add to. The newly-created playlist goes into the "Playlists" section. A playlist is created by providing a title. Users can edit it by providing a new title or uploading a cover (an image).	<i>Registered User</i>
2.8. Create/Edit an album	Users can create playlists with their own finished songs, which are called albums, by providing a title. They can add them to the album after going to the "Albums" section, clicking the (+) button and uploading it or by choosing a song from the section "Own songs" and choosing an album to add it to. Album title and cover can be edited as well.	<i>Registered user</i>
2.9. Add/Edit/Resolve your own comment under a song idea	Users can add a comment under a song idea in another user's page or reply to an existing comment under their own ideas. Users can edit their own comments once they've posted them by clicking an "Edit" button on them. Each comment opens a new discussion about an improvement about this song idea, so comments could also have a "button" which states that this improvement is added.	<i>All users</i>
2.10. Browse others' profiles	Users can see the public information other users provide, including their four song collection sections and a section called "All", which contains all songs uploaded, sorted by date of upload.	<i>All users</i>

2.11. Resolve/Read notifications	Users can receive a notification when someone has commented on their song ideas and go directly to it to see it. Users who have commented under someone else's idea will also be notified if the user has replied afterwards.	<i>Registered user</i>
---	---	------------------------

3. Main Views (SPA Frontend)

View name	Brief Descriptions	URI
3.1. Home	Contains a brief introduction to the purpose of the website and provides the opportunity to browse randomly selected albums and playlists presented in different genre sections. Anonymous users can register and Registered users can log in. A user logged in can click on a "notifications" button in order to see them.	/
3.2. User profile	Presents a user's profile which consists of a picture, username, genres working in and buttons leading to the different song sections ("Albums", "Playlists", "Own songs", "Song ideas"). Currently displays section "All".	/ {username}
3.3. All albums	Presents all albums of the user {username} and provides the opportunity to click on each in order to explore its songs.	/ {username} /albums
3.4. All playlists	Presents all playlists of the user {username} and provides the opportunity to click on each in order to explore its songs. The owner can delete/edit/create a new playlist.	/ {username} /playlists
3.5. Own Songs	Presents a list of all finished songs that the user has uploaded and the opportunity to play them. The owner can delete/edit/add songs.	/ {username} /own-songs
3.6. All song Ideas	Presents a list of song ideas (cover and title) which the user can click on to explore. The owner can as well delete/add ones.	/ {username} /song-ideas
3.7. Album	Presents a list of all songs in the album with {albumId} and provides the opportunity to listen to them by clicking "play" on them. When a song is clicked "play" on while another is being played, the one being played stops playing and the one clicked on starts playing. A song can be deleted/edited/added by the owner.	/ {username} /albums/ {albumId}

3.8. Playlist	Provides a list of songs in the current playlist. Songs can be played. A playlist can also be deleted/edited/added by the owner.	<i>/ {username} /playlists /{playlistId}</i>
3.9. Song Idea	Presents the basic information about a song idea as well as the comments left under it.	<i>/ {username} /song-ideas /{songIdeaId}</i>
3.10. Notifications	Presents all notifications of {username} in a list as well as the opportunity to click on them and make them read (or go to the song idea page in which to find them (/{someUsername}/song-ideas/{songIdeaId}))	<i>/ {username} /notifications</i>
3.11. Register	Provides a form used to fill in the details needed to register.	<i>/register</i>
3.12. Login	Provides a form in which the user can fill in the details needed to log in.	<i>/log-in</i>

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
4.1. Users	GET <i>User Data</i> for all users, and POST new <i>User Data</i> (Id is auto-filled by OKTS and modified entity is returned as result from POST request). Available only for <i>Administrators</i> .	<i>/api/users</i>
4.2. User	GET, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>username (they are unique)</i> . GET all own songs and song ideas for this username.	<i>/api/{username}</i>
4.3. Register	POST <i>User Credentials</i> (e-mail address and password).	<i>/api/login</i>
4.4. Logout	POST a logout request for ending the active session with <i>MusicShare</i> .	<i>/api/logout</i>
4.5. Playlists by genre	GET <i>Playlists (albums included) and divide them</i> by genre (first 10 in each section, sorted by date of upload)	<i>/api/</i>
4.6. All Albums	GET all albums belonging to {username}, POST <i>an Album</i> .	<i>/ {username}/albums</i>
4.7. All Playlists	GET all playlists for a specified {username}, POST <i>a Playlist</i> .	<i>/ {username}/playlists</i>

4.8. Own Songs	GET all own songs for a specified {username}, POST a song.	<i>/api/{username}/own-songs</i>
4.9. All song ideas	GET all song ideas for a user with specified {username}. POST a song.	<i>/api/{username}/song-ideas</i>
4.10. Album	GET, PUT, DELETE an album (and all its songs if needed). DELETE a song (only from the album). GET all songs sorted by time of addition.	<i>/api/{username}/albums/{albumId}</i>
4.11. Playlist	GET, PUT, DELETE a playlist (and all its songs if needed). DELETE a song (only from the playlist). GET all songs sorted by time of addition.	<i>/api/{username}/playlists/{playlistId}</i>
4.12. Song Idea	GET, PUT, DELETE a song idea. GET all comments sorted in order of posting. POST, PUT, DELETE a comment.	<i>/api/{username}/song-ideas/{songIdeaId}</i>
4.13. Notifications	GET all notifications sorted by time of reception.	<i>/api/{username}/notifications</i>