

Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2021

Project author		
FN#	Name	Email
62379	Martin Ravanov	m.ravanov@gmail.com

Project name	Interband
--------------	-----------

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

Interband provides ability for bands to communicate and get to know each other by writing posts, posting and commenting on their albums and songs and creating and going to events. In addition, it allows users (bands) to register and later log in. 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 pagination on the suitable pages. The authorization and authentication will be done using **JSON Web Token**. The system uses **MongoDB - Atlas** and **Mongoose** as a database. The main user roles (actors in UML) are:

- *Anonymous User* – can view all bands, albums, events and posts, but cannot comment albums, like/unlike posts and go/won't go to events nor create, edit and delete any of them. *Anonymous users* can do all that only after logging in (or registering).
- *Registered User (Registered Users and Bands are interchangeable in the current scope)* – can use the main use cases stated below after logging in the system.

## 2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
<b>2.1. View home page (All Posts + Create post page)</b>	<i>Anonymous Users</i> and <i>Registered Users</i> can visit the Home page of the system and see and search all the posts, but only <i>Registered Users</i> can create, edit, delete or like a post.	<i>Anonymous User, Registered User</i>
<b>2.2. View all bands</b>	<i>Anonymous Users</i> and <i>Registered Users</i> can view a list of all the registered bands - the All Bands page.	<i>Anonymous User, Registered User</i>

<b>2.3. Register</b>	<i>Anonymous Users</i> can register in the system by providing a valid e-mail address, band name, genre, band photo URL and a valid password, confirming it.	<i>Anonymous User</i>
<b>2.4. Login</b>	<i>Anonymous Users</i> can log in the system by providing their e-mail address and password.	<i>Anonymous User</i>
<b>2.5. Logout</b>	<i>Registered Users</i> can log out of the system.	<i>Registered User</i>
<b>2.6. Change User Data/Edit profile</b>	<i>Registered Users</i> can view their own Profile page and also edit their personal <i>User Data</i> .	<i>Registered User</i>
<b>2.7. Create posts</b>	<i>Registered Users</i> can create new posts, later appearing in All Posts page.	<i>Registered User</i>
<b>2.8. Edit/Delete posts</b>	<i>Registered Users</i> can edit/delete only their own posts.	<i>Registered User</i>
<b>2.9. Like/unlike post</b>	<i>Registered Users</i> can like/unlike any post.	<i>Registered User</i>
<b>2.10. View post details</b>	<i>Registered Users</i> and <i>Anonymous Users</i> can see more details about a specific post.	<i>Registered User, Anonymous User</i>
<b>2.11. View all albums</b>	<i>Anonymous Users</i> and <i>Registered Users</i> can visit the All Albums page of the system and see all the albums, but only <i>Registered Users</i> can create, edit or delete an album.	<i>Anonymous User, Registered User</i>
<b>2.12. Create album with certain songs</b>	<i>Registered Users</i> can post their albums (plus all their songs), later appearing in their own Profile page and in All Albums page.	<i>Registered User</i>
<b>2.13. View album details</b>	<i>Registered Users</i> and <i>Anonymous Users</i> can see more details about a specific album.	<i>Registered User, Anonymous User</i>
<b>2.14. Comment album</b>	<i>Registered Users</i> can comment an album when visiting the Detail page of a specific album.	<i>Registered User</i>
<b>2.15. View All events</b>	<i>Anonymous Users</i> and <i>Registered Users</i> can visit the All Events page of the system and see all the events, but only <i>Registered Users</i> can create, edit, delete or go to an event.	<i>Anonymous User, Registered User</i>
<b>2.16. Create event</b>	<i>Registered Users</i> can create events, later appearing in All Events page.	<i>Registered User</i>

<b>2.17. Edit/Delete event</b>	<i>Registered Users</i> can edit/delete only their own events.	<i>Registered User</i>
<b>2.18. Go to event/Won't go to event</b>	<i>Registered Users</i> can select the 'Go to event/Won't go to event' option on any event.	<i>Registered User</i>
<b>2.19. View event details</b>	<i>Registered Users and Anonymous Users</i> can see more details about a specific event.	<i>Registered User, Anonymous User</i>
<b>2.20. View own profile page</b>	<i>Registered Users</i> can view their own Profile page – all the info they have given as well as their albums.	<i>Registered User</i>
<b>2.21. View the profile of a specific band</b>	<i>Registered Users and Anonymous Users</i> can view all the info a specific band has provided as well as their albums on their Profile page.	<i>Registered User, Anonymous Users</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
<b>3.1. Home page (All Posts + Create post page)</b>	Presents a list of all the posts and their likes made by the bands as well as a view allowing bands to create, edit, delete or like/dislike a post and a view for post searching.	<i>/posts</i>
<b>3.2. Post details page</b>	Presents a view of additional information regarding a specific post	<i>/posts/{postId}</i>
<b>3.3. All Bands page</b>	Presents a list of all the registered bands and some short information - their band photo, genre, and name.	<i>/users</i>
<b>3.4. Band's Profile page</b>	Presents all the info a certain band has provided as well as their albums and songs. If this is the currently logged in band's page, the user can view and edit one's personal <i>User Data</i> .	<i>/users/{userId}</i>
<b>3.5. All Events + Create event page</b>	Presents a list of all the events and their goings selected by the bands as well as a view allowing bands to create, edit, delete or select the go/won't go option on an event.	<i>/events</i>
<b>3.6. Event details page</b>	Presents a view of additional information regarding a specific event	<i>/events/{eventId}</i>
<b>3.7. All Albums + Create album page</b>	Presents a list of all the albums made by the bands as well as a view allowing bands to create, edit or delete an album.	<i>/albums</i>

<b>3.8. Album details + Comment album page</b>	Presents a view of additional information regarding a specific album and allows users to leave a comment on this album.	<i>/albums/{albumId}</i>
<b>3.9. Login page</b>	Presents a view allowing the bands to login.	<i>/signIn</i>
<b>3.10. Register page</b>	Presents a view allowing the bands to register.	<i>/signUp</i>

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
<b>4.1. Users</b>	GET <i>User Data</i> for all users.	<i>/api/users</i>
<b>4.2. User</b>	GET, PUT <i>User Data</i> for <i>User</i> with specified <i>userId</i> , according to restrictions described in UCs.	<i>/api/users/{userId}</i>
<b>4.3. Login</b>	POST <i>User Credentials</i> (e-mail address and password) and receive a valid <i>Security Token</i> to use in subsequent API requests.	<i>/api/users/signIn</i>
<b>4.4. Register</b>	POST new <i>User Credentials</i> (name, e-mail address, genre, image URL and password + confirmPassword) and receive a valid <i>Security Token</i> to use in subsequent API requests.	<i>/api/users/signUp</i>
<b>4.5. Posts</b>	GET <i>Post Data</i> for all posts, and POST new <i>Post Data</i> .	<i>/api/posts</i>
<b>4.6. Search Posts</b>	GET <i>Post Data</i> for all posts based on the search data.	<i>/api/posts/search</i>
<b>4.7. Post</b>	GET, PUT, DELETE <i>Post Data</i> for <i>Post</i> with specified <i>postId</i> , according to restrictions described in UCs.	<i>/api/posts/{postId}</i>
<b>4.8. Like Post</b>	PUT <i>Post Data</i> for <i>Post</i> with specified <i>postId</i> , according to restrictions described in Ucs when this <i>Post</i> has been liked/unliked.	<i>/api/posts/{postId}/likePost</i>
<b>4.9. Events</b>	GET <i>Event Data</i> for all events, and POST new <i>Event Data</i> .	<i>/api/events</i>
<b>4.10. Event</b>	GET, PUT, DELETE <i>Event Data</i> for <i>Event</i> with specified <i>eventId</i> , according to restrictions described in UCs.	<i>/api/events/{eventId}</i>
<b>4.11. Go to Event</b>	PUT <i>Event Data</i> for <i>Event</i> with specified <i>eventId</i> , according to restrictions described in Ucs when this <i>Event</i> has gotten the go to/ won't go to option.	<i>/api/events/{eventId}/goTo</i>
<b>4.12. Albums</b>	GET <i>Album Data</i> for all albums, and POST new <i>Album Data</i> .	<i>/api/albums</i>
<b>4.13. Album</b>	GET, PUT, DELETE <i>Album Data</i> for <i>Album</i> with specified <i>albumId</i> , according to restrictions described in UCs.	<i>/api/albums/{albumId}</i>

<b>4.14. Comment album</b>	PUT <i>Album Data</i> for <i>Album</i> with specified <i>albumId</i> , according to restrictions described in Ucs when this <i>Album</i> has been commented.	<i>/api/albums/{albumId}/comment</i>
----------------------------	--	--------------------------------------