Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2019

Project author		
№	Pseudonym	Face-to-face/ online
1	Dan4inatora	face-to-face

Project name	BookLibrary

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

If you want to check a book or an author then the Book Library application is the one for you. It offers you an user friendly library that has all the information you will need about various books and authors. There will be a login system, which allows users to register and administrators to manage the library. There can be a contact us page, where regular users can communicate with the administrator. 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.

- *Regular User* (extends *Registered User*) – can search for books and get info like story description, publish date, author and info about him, page count.
- *Administrator* (extends *Registered User*) – can add/delete books and authors from the database, can change book's titles, authors, descriptions…etc

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
1. Browse information and try sample tests	The *User* can browse the information views (Home, Books, Authors) in *the Library*, and can choose to read more about a book or Author	All users
2. Register	*Regular User* can register in the system by providing a valid e-mail address, first and last name, and choosing password.	*RegisterUser, Administrator*

	Administrator can register new by entering *User Data*	
3. Change User Data	*Registered User* can view and edit her personal *User Data* *Administrators can edit user*	*Registered User, Administrator*
4. Manage Users	*Administrator* can choose a *User* to manage, and can manage the chosen User - edit (using Change User Data UC) or delete.	*Administrator*
5. Add/Edit Book	*Administrator can choose to edit every aspect of a book such as author, description, name, pages as well as add or delete books*	*Administrator*
6. Add/Edit Author	*Administrator can choose to edit every aspect of an author such as name, books written, pic as well as add or delete authors*	*Administrator*
7. Send Request	*In the contact page an user can send a request about a book or an author to the administrator*	*Registered User*

3. Main Views (SPA Frontend)

View name	**Brief Descriptions**	**URI**
1. Home	Presents the introductory information for the purpose of the system as well as detailed instructions how to start using it. Prominently offers ability to register.	/
2. User Registration	Presents a view allowing the *Anonymous Users* to register.	*/register*
3. Login	Presents a view allowing the users to login.	*/login*
4. User Data	Presents ability to view and edit personal *User Data*,.	*/personal*
5. Library	Shows a library of the most recently added books	*/library*
6. Books	Presents a library of books and a search engine to browse them	*/books*
7. Users	Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only, as described in).	*/users*
8. Add Book	Presents a functionality to manage (CRUD) add/delete/edit a book to the library (Administrator only)	*/addbook*

9. Authors	Presents a list of Authors and information about them and their related works	*/authors*
10. Add author	Presents a functionality to manage (CRUD) Authors and data about them	*/addauthor*
10. Wish-list	Shows a page of all the want to reads of an user	*/wishlist*
11. Book	Show all the info about a book	*/book/{id}*
12. Contact	A form to send information to the administrator by normal users	*/contact*

1. API Resources (Node.js Backend)

View name	**Brief Descriptions**	**URI**
1. Users	GET *User Data* for all users, and POST new *User Data*. Available only for *Administrators*.	*/api/users*
2. User	GET, PUT, DELETE *User Data* for *User* with specified *userId*.	*/api/users/{userId}*
3. Login	POST *User Credentials* (e-mail address and password)	*/api/login*
4. Logout	POST a logout request for ending the active session	*/api/logout*
5. Books	GET get a list of all books	*/api/books*
6. Book	GET, PUT, DELETE *book with specified bookId*	*/api/books/{bookId}*
7. Wishlist	GET,PUT, POST, DELETE from user's wishlist	*/api/wishlist*
8. Authors	GET a list of all authors	*/api/authors*
9. Author	GET, PUT, DELETE author from authors list by authorId	*/api/ authors/{authorId}*
10. Request to Admin	POST message by user for admin	*/api/contact*
11. Messages	GET all the messages sent from users(admin only)	*/api/messages*