

Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2023

Project author		
	Name	FN
	Martin Marinov	OMI0800052

Project name	PCBuilding
--------------	------------

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

Many people want to have a good computer configuration. They have a choice to buy ready one or make custom one. In my opinion people choose more the second option, but some of them don't know what parts to choose. So this app will connect users with experts and experts can give advices and say that the configuration is good or bad. If it is good everyone can see it so that people can decide if it fits their needs.

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
2.1. Browse information and view configurations	The <i>User</i> can browse the information views and see all approved by experts configurations.	All users
2.2. Register	<i>Anonymous User</i> can register in the system by providing a valid e-mail address, first and last name, and choosing	<i>Anonymou</i>

	<p>password. By default, all new registered users have <i>User</i> role.</p> <p><i>Administrator</i> can register new by entering <i>User Data</i> and choosing a Role (<i>User</i>, <i>Expert</i>, or <i>Administrator</i>).</p>	<i>s User, Administrator</i>
2.3. Change User Data	<p><i>Registered User</i> can view and edit his personal <i>User Data</i>.</p> <p><i>Administrator</i> can view and edit <i>User Data</i> of all <i>Users</i> and assign them <i>Roles</i>: <i>User</i>, <i>Expert</i>, or <i>Administrator</i>.</p>	<i>Registered User, Administrator</i>
2.4. Manage Users	<p><i>Administrator</i> can browse and filter users based on different criteria: first and last name, email, Role.</p> <p><i>Administrator</i> can choose a <i>User</i> to manage, and can manage the chosen <i>User</i> - edit (using Change User Data UC) or delete.</p> <p><i>Administrator</i> can create a new user using <i>Register UC</i>.</p>	<i>Administrator</i>
2.5. Manage Configurations	Experts and Administrators can browse and approve all configurations	<i>Expert, Administrator</i>
2.6. Comment Configurations	<i>Experts and Admins</i> can put comments to the proposed configurations	<i>Expert, Administrator</i>
2.7. Add/Edit Configurations	<p><i>Users</i> can add new configuration and edit their configurations.</p> <p><i>Experts and Admins</i> can edit each configuration</p>	All users
2.8. Manage Parts	<i>Experts and Administrators</i> can add or remove certain part from the list of available parts.	<i>Expert, Administrator</i>
2.9. Manage Constraints	Experts and Administrators can add or remove certain constraint about parts compatibility	<i>Expert, Administrator</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
3.1. Home	Presents information for available	/

	configurations	
3.2. Parts	Presents ability to manage (CRUD) Users and their User Data (available for <i>Administrators</i> and Experts).	<i>/parts</i>
3.3. Test Results	Presents information about given configuration Gives ability to administrators and experts to comment.	<i>/configuration/{configurationId}</i>
3.4. Edit Configuration	Presents ability for user to edit his own configuration before aproval and to admins and experts to edit it whenever they want.	<i>/configuration/edit/{configurationId}</i>
3.5. User Registration	Presents a view allowing the <i>Anonymous Users</i> to register.	<i>/register</i>
3.6. Login	Presents a view allowing the users to login.	<i>/login</i>
3.7. User Data	Presents ability to view and edit personal <i>User Data</i>	<i>/account</i>
3.8. Users	Presents ability to manage (CRUD) Users and their User Data (available for <i>Administrators</i> only).	<i>/users</i>
3.9. Constraints	Presents ability to manage (CRUD) Constrants (available for <i>Administrators</i> and Experts).	<i>/constraints</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> 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>userId</i> , according to restrictions described in UCs.	<i>/api/users/{userId}</i>
4.3. Login	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/login</i>
4.4. Logout	POST a logout request for ending the active	<i>/api/logout</i>

	session, and invalidating the issued <i>Security Token</i> .	
4.5. Parts	GET <i>Parts</i> , POST new <i>Part</i> , <i>PATCH</i> edit <i>Part</i>	<i>/api/parts</i>
4.6. Register	POST new User Data	<i>/api/register</i>
4.7. Configurations	GET Configurations List and POST new <i>Configuration</i>	<i>/api/configuration</i>
4.8. Configuration data	GET, PUT, <i>PATCH</i> , DELETE <i>Configuration Data</i>	<i>/api/configuration/{configurationId}</i>
4.9. Approve Configuration	POST Available for administrators only	<i>/api/configuration/{configurationId}/approve</i>
4.10. Comments	POST Comment	<i>/api/comments/</i>
4.11. Edit Comments	<i>PATCH</i> , DELETE Comment	<i>/api/comments/{commentId}</i>
4.12. Constraints	GET, POST	<i>/api/constraints/</i>
4.13. Edit Constraints	<i>PATCH</i> , DELETE Constraint	<i>/api/constraints/{constraintId}</i>