

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

Project author		
№	Pseudonym	Face-to-face/ online
1	Maria Cholakova	
2	Roksana Yaneva	
3	Petya Sokolova	

Project name	Fashion Days
--------------	--------------

1. Short project description (Business needs and system features)
<p>FashionDays is an online clothing store that has the following features: creating a user profile, shopping cart, online ordering, product search, chat room that you can use to communicate with other customers and help the customer. There are also administrators to manage all that. The system will be developed as a <i>Single Page Application (SPA)</i> using <b>React.js</b> as front-end, and <b>Node.js + Express.js</b> as backend technologies.</p> <p>The main user roles (actors in UML) are:</p> <ul style="list-style-type: none"> <li>• <i>Unregistered User</i> – Can view all products and information about them, but cannot make purchases</li> <li>• <i>Registered User</i> – Can view all products and information about them, as well as order products by adding them to his/her shopping cart</li> </ul>

- *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users*, as well as generate and update website content (products, content, promotions)

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
<b>2.1. Register</b>	<p><i>Unregistered User</i> can register in the system by providing a valid email address, first and last name, and choosing password.</p> <p><i>Administrator</i> can register new users by entering <i>User Data</i> and choosing a Role (<i>Normal user or Administrator</i>).</p>	<i>Unregistered User, Administrator</i>
<b>2.2. Change User Data</b>	<p><i>Registered User</i> can view and edit their personal <i>User Data</i>.</p> <p><i>Administrator</i> can view and edit <i>User Data</i> of all <i>Users</i>.</p>	<i>Registered User, Administrator</i>
<b>2.3. Manage Users</b>	<p><i>Administrator</i> can browse and filter users based on different criteria: first and last name, email.</p> <p><i>Administrator</i> can choose a <i>User</i> to manage, and can manage the chosen <i>User</i> - edit (using <i>Change User Data UC</i>) or delete.</p> <p><i>Administrator</i> can create a new user using <i>Register UC</i>.</p>	<i>Administrator</i>

<b>2.4. Manage Content</b>	<i>Administrator can add new products, modify existing products and announce promotions and discounts.</i>	<i>Administrator</i>
<b>2.5. Add to shopping cart</b>	<i>Registered users can add products to their shopping cart and then purchase them.</i>	<i>Registered User</i>
<b>2.6. Search for a product</b>	<i>All users can search for available products by entering their request in the search field.</i>	<i>Unregistered User, Registered User, Administrator</i>
<b>2.7. Filter products by some criteria</b>	<i>All users can filter the products by some criteria (type, price, size, colour, etc.)</i>	<i>Unregistered User, Registered User, Administrator</i>
<b>2.8. Use online chat</b>	<i>Registered users can use the online chat and communicate with other users.</i>	<i>Registered User, Administrator</i>
<b>2.9. Comment under a product</b>	<i>Registered users can post a comment in the product page.</i>	<i>Registered User</i>
<b>2.10. Send a question to the administrator</b>	<i>Registered user can send a request to the administrator with a question regarding products, shipping, etc.</i>	<i>Registered User</i>

3. Main Views (SPA Frontend)		
<b>View name</b>	<b>Brief Descriptions</b>	<b>URI</b>
<b>3.1. Home</b>	<i>Presents the introductory information for the purpose of the system as well as detailed instructions how to start using it.</i>	<i>/</i>

<b>3.2. User settings</b>	<i>Provides a form for editing user's settings - password, name, etc</i>	<i>/settings</i>
<b>3.3. User orders</b>	<i>Presents the orders the user has placed and their status</i>	<i>/orders</i>
<b>3.4. Product</b>	<i>Presents a specific product - picture, count available, size, description, buttons for adding to cart</i>	<i>/item={productId}</i>
<b>3.5. Shopping cart</b>	<i>Present the user's shopping cart - list of items in the shopping cart, buttons for removing and purchasing</i>	<i>/cart</i>
<b>3.6. Login/Register</b>	<i>Provides a login or register form</i>	<i>/loginRegister</i>
<b>3.7. Help</b>	<i>Presents FAQ section</i>	<i>/help</i>
<b>3.8. About</b>	<i>Presents information about the website</i>	<i>/about</i>
<b>3.9. Contacts</b>	<i>Presents contacts info</i>	<i>/contact</i>
<b>3.10. Return policy</b>	<i>Presents info about the return policy</i>	<i>/returnPolicy</i>
<b>3.11. Conditions</b>	<i>States the general conditions for using the website</i>	<i>/conditions</i>
<b>3.12. Users</b>	<i>Presents ability to manage (CRUD) Users and their User Data (available for Administrators only, as described in UCs).</i>	<i>/users</i>

4. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI
4.1. All customers	<i>GET all customers</i>	<i>/api/customers</i>
4.2. Customer	<i>GET, PUT, DELETE customer with the specified mobile</i>	<i>/api/customers/{mobile}</i>
4.3. All products	<i>GET all products</i>	<i>/api/products</i>
4.4. Product	<i>GET, PUT, DELETE product with the specified productName, and POST new product with parameters productName, price, count, size</i>	<i>/api/products/{productName}</i> <i>/api/products/{productId}</i>
4.5. Count	<i>GET and PUT count of the product specified by productName</i>	<i>/api/products/{productName}/count</i> <i>/api/count/{productName}</i> <i>/api/products/{productId}/count</i>
4.6.Users	<i>GET User Data for all users, and POST new User Data (Id is auto-filled and modified entity is returned as result from POST request). Available only for Administrators.</i>	<i>/api/users</i>
4.7.User	<i>GET, PUT, DELETE User Data for User with specified userId.</i>	<i>/api/users/{userId}</i>
4.8. All orders	<i>Presents the orders the user has placed and their status</i>	<i>/orders</i>
4.9.Order	<i>GET, PUT, DELETE Order with specified orderId.</i>	<i>/orders/{orderId}</i>
4.10.		
4.11.		

4.12.		
-------	--	--