

Assignment 1	Project Summary
Course	Multimedia Technologies with Angular and TypeScript - 2020

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

Project name	Online Knowledge Testing System
--------------	---------------------------------

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

Today's gaming community is as strong as ever and people need a platform where they can browse, review and buy games online. The **PCMRShop** provides just the place for that. The project will be developed as a *Single Page Application (SPA)* using **Angular** as front-end, and **Node.js + Express** as backend technologies. Data will be fetched from a database. The user types are:

- Anonymous User - can view the store, but not buy
- Logged-In User - can view the store, buy items, and add preferences
- Admin - all of the above + manage users (create, edit, delete)

### 2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
<b>2.1. Browse</b>	The user can browse the shop (Home, About) in the PCMRShop.	All users
<b>2.2. Register</b>	<i>Anonymous User</i> can register in the system by providing a valid e-mail address, first and last name, and choosing password.	<i>Anonymous User</i>
<b>2.3. Login</b>	<i>Anonymous User</i> can login in the system by providing a valid username and password combination. Upon successful login, he'll get Logged-In User privileges.	<i>Anonymous User</i>

<b>2.3. Change User Data</b>	<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 view their purchase history.</p>	<i>Registered User, Administrator</i>
<b>2.4. 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 Change User Data UC) or delete.</p> <p><i>Administrator</i> can create a new user with a personal form visible only to <i>Administrators</i>.</p>	<i>Administrator</i>
<b>2.5. Add products to cart</b>	<i>Logged In Users</i> can add games to their carts.	<i>Logged In User, Administrator</i>
<b>2.6. Checkout/Buy</b>	<i>Logged In Users</i> can purchase all items in their cart. For the purpose of the project, no actual purchasing methods will be implemented - upon 'purchasing', the items will be added to the <i>User's</i> purchase history.	<i>Logged In User, Administrator</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
<b>3.1. Home</b>	Presents the main page of the games shop, where people can browse games and filter them by genre, date of release, etc.	/
<b>3.2. Cart</b>	Presents the page with all added games to your cart.	/cart
<b>3.3. View Game</b>	Presents a place where all users can view a singular more detailed page about the selected game.	/game
<b>3.4. About</b>	Presents the about page for the site.	/about
<b>3.5. User Registration</b>	Presents a view allowing the <i>Anonymous Users</i> to register.	/register
<b>3.6. Login</b>	Presents a view allowing the users to login.	/login

<b>3.7. User Data</b>	Presents ability to view and edit personal <i>User Data</i>	<i>/user</i>
<b>3.8. Users</b>	Presents ability for <i>Administrators</i> to manage and select users	<i>/users</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, and POST new <i>User Data</i> . Available only for <i>Administrators</i> .	<i>/api/users</i>
<b>4.2. User</b>	GET, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>userId</i> .	<i>/api/users/{userId}</i>
<b>4.3. Login</b>	POST <i>User Credentials</i> (e-mail address and password).	<i>/api/login</i>
<b>4.4. Logout</b>	POST a logout request.	<i>/api/logout</i>
<b>4.5. Products</b>	GET <i>Products</i> , and POST new <i>Product</i>	<i>/api/products</i>
<b>4.6. Product</b>	GET, PUT, DELETE <i>Products</i>	<i>/api/products/{productId}</i>
<b>4.7. Purchase History</b>	GET <i>Purchase History</i>	<i>/api/users/{userId}/purchasehistory</i>