

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

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

Project name	Car Service Search
--------------	--------------------

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

The Internet age changed the way we learn and test our knowledge. The **Car Service Search (CSS)** provides ability for service owners to register their companies and for users to search and rate them. In addition to that it allows users to register, and administrators to manage them. The system will be developed as a *Single Page Application (SPA)* using **Angular** 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. The main user roles (actors in UML) are:

- *Anonymous User* – can only view the information pages and try few sample tests without saving test results.
- *User (extends Registered User)* – can write comments, start new themes and rate the services.
- *Car-service (extends Registered User)* – can offer its services.
- *Administrator (extends Registered User)* – can manage (create, edit user data and delete) all *Registered Users*, as well as *Car infos and comments*.

2. Main Use Cases / Scenarios

Use case name	Brief Descriptions	Actors Involved
2.1. Browse information	The <i>User</i> can browse the information views (Home, Car-infos, Car-services, Blogs) in CSS.	All users

2.2. Register	<p><i>Anonymous User</i> can register in the system by providing a valid e-mail address, first and last name, and choosing password.</p> <p><i>Administrator</i> can register new by entering <i>User Data</i> and choosing a Role (<i>User</i>, <i>Car-service</i>, or <i>Administrator</i>).</p>	<i>Anonymous 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>Registered Services</i> can edit their information.</p> <p><i>Administrator</i> can view and edit <i>User Data</i> of all <i>Users</i>.</p>	<i>Registered User, Car-service, Administrator</i>
2.4. Manage Users	<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 using <i>Register UC</i>.</p>	<i>Administrator</i>
2.5. Manage Car Infos	<p><i>Administrator</i> can browse and filter <i>Cars</i> based on different criteria: Car brand, model, year, etc.</p> <p><i>Administrator</i> can add or remove <i>Cars</i>.</p>	<i>Administrator</i>
2.6. Manage blogs	<i>Administrator</i> can browse comments of all <i>Users</i> and <i>Car-services</i> , edit and delete them.	<i>Administrator</i>
2.7. Add/Edit comments	<i>Users</i> and <i>Service owners</i> can write comments for different models cars and also for the car-services.	<i>User, Car-service</i>
2.8. Rate Car-services	<i>Users</i> can rate Car-services with stars from 1 to 5.	<i>User</i>
2.9. Propose adding a new cars	<i>Users</i> can send suggestions to the admin to add car models that are not currently available in the list.	<i>User</i>
2.10. Signal for already inactive service	<i>Users</i> can send signal to the admin if they find registered Car-service that is no longer working.	<i>User</i>

3. Main Views (SPA Frontend)

View name	Brief Descriptions	URI
3.1. 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.	/
3.2. Cars	Displays a list of all available car models and also an easy search filter.	/cars
3.3. Comments for specific car	Presents comments from users and car-service owners.	/{{car model}}/comments
3.4. Services for a specific car model	Presents list of car-services who noted that they support the car.	/{{car model}}/appropriate-services
3.5. User Registration	Presents a view allowing the <i>Anonymous Users</i> and <i>Service owners</i> to register in CSS	/register
3.6. Login	Presents a view allowing the users to login.	/login
3.7. User Data	Presents ability to view and edit personal <i>User Data</i> , as well as deregister from CSS.	/personal
3.8. Services list	Presents ability to browse and filter the available registered car-services.	/services
3.9. Users	Presents ability to manage (CRUD) Users and their User Data (available for <i>Administrators</i> only, as described in UCs).	/users
3.10. About	Presents information about the CSS project and his owner.	/about

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> (Id is auto-filled by CSS and modified entity is returned as result from POST request). Available only for <i>Administrators</i> .	/api/users
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.	/api/users/{userId}
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.	/api/login
4.4. Logout	POST a logout request for ending the active session with CSS, and invalidating the issued	/api/logout

	<i>Security Token.</i>	
4.5. Cars list	GET <i>Cars list.</i>	<i>/api/cars</i>
4.6. Comments	GET, PUT, DELETE <i>comments</i> for <i>specific car.</i>	<i>/api/{carid}/comments</i>
4.7. Car-services	GET services available for specific car.	<i>/api/{carid}/services</i>
4.8. More	There will be more, but now im not exactly sure what.	<i>/</i>