

Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2020

Project author		
No	Pseudonym	Face-to-face/ online
1	sis0k0	face-to-face

Project name	PupCare
--------------	---------

1. Short project description (Business needs and system features)
<p>PupCare provides 24/7 on-demand access to people seeking dog walking jobs as well as sitters and boarding services in your area. The system will be developed as a <i>Single Page Application (SPA)</i> 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. There might also be an Android/iOS app (React Native) accompanying the website. The main user roles (actors in UML) are:</p> <ul style="list-style-type: none"> • <i>Dog Carer</i> (extends <i>Registered User</i>) – can search for dog walking/sitting jobs in their area, apply for a job, submit post-walk reports. • <i>Dog Owner</i> (extends <i>Registered User</i>) – can register a job for dog sitting/walking, see profiles of dog walkers, has access to post-walk reports. • <i>Administrator</i> (extends <i>Registered User</i>) – can manage (create, edit user data and delete) all <i>Registered Users</i>, access walk reports, jobs.

2. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
2.1. Register	<i>Anonymous User</i> can register in the system by providing a valid e-mail address, first and last name,	<i>Anonymous User</i> ,

	<p>and choosing password. By default, all new registered users have <i>Dog Owner</i> role.</p> <p><i>Administrator</i> can register new by entering <i>User Data</i> and choosing a Role (<i>Dog Owner</i>, <i>Dog Carer</i>, or <i>Administrator</i>).</p>	<i>Administrator</i>
2.2. 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 User - 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>
2.3. Manage Jobs	<p><i>Dog Owner</i> can browse her/his <i>Jobs</i>, add new <i>Job</i> using <i>Add/Edit Job UC</i>, and delete a <i>Job</i>, as well as view the <i>Post-walk Reports</i> attached to their own <i>Jobs</i>.</p> <p><i>Administrator</i> can browse all jobs, edit and delete them.</p>	<i>Dog Owner,, Administrator</i>
2.4. Add a Job	<p><i>Dog Owner</i> can browse her/his <i>Jobs</i>, add new <i>Job</i> using <i>Add/Edit Job UC</i>, and delete a <i>Job</i>, as well as view the <i>Post-walk Reports</i> for their own <i>Jobs</i>.</p> <p><i>Administrator</i> can browse jobs of all <i>Dog Owners</i>, edit and delete them.</p>	<i>Dog Owner, Administrator</i>
2.5. Add/Edit Job	<i>Dog Owner</i> or <i>Administrator</i> specifies/edits <i>Job</i> meta-data such as: Job name, subject, description, tags. <i>Dog Owner</i> or <i>Administrator</i> saves the edited <i>Job</i> .	<i>Dog Owner, Administrator</i>
2.6. Apply for a Job	<i>Dog Carer</i> browses <i>Jobs</i> available and can sort and filter them using different Job metadata fields. <i>Dog Carer</i> chooses a <i>Job</i> to complete.	<i>Dog Carer,</i>
2.7. Select Dog Carer	<i>Dog Owner</i> can select one of the <i>Dog Carers</i> applying for a <i>Job</i> .	<i>Dog Owner</i>
2.8. Browse Job Results	<p><i>Dog Carer</i> can browse her own completed <i>Job Results</i>.</p> <p><i>Dog Owner</i> can browse the <i>Job Results</i> for <i>Jobs</i> created by her/him and for particular <i>Dog Carer Groups</i> assigned to the <i>Job</i>.</p>	<i>Dog Carer, Dog Owner</i>

2.10. Complete Job	<i>Dog Carer</i> completes a Job by submitting a post-walk report: <ul style="list-style-type: none"> - questionnaire - uploaded photo of the walked pup 	<i>Dog Carer</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. Jobs	Presents Jobs available according to <i>User's Role</i> and identity. Offers abilities to browse, choose, create, read, update, delete (CRUD) Jobs, as defined by <i>UCs</i> (for <i>Administrators</i> and <i>Dog Owners</i> only).	/jobs
3.3. Apply Job	Presents a Job if available to particular <i>Dog Carer</i> allowing them to apply for it.	/jobs/:id
3.4. Job Results	Presents <i>Job Results</i> available according to <i>User's Role</i> and identity as defined by <i>UCs</i> .	/job-results
3.5. User Registration	Presents a view allowing the <i>Anonymous Users</i> to register in <i>OKTS</i> .	/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> .	/profile
3.8. Dashboard	<p><i>Dog Carer</i> - Presents in real time the <i>Dog Carer's</i> progress on applied Jobs.</p> <p><i>Dog Owner</i> - Presents in real time the <i>Dog Owner's</i> progress on created jobs.</p>	/dashboard
3.9. Users	Presents ability to manage (CRUD) <i>Users</i> and their <i>User Data</i> (available for <i>Administrators</i> only, as described in <i>UCs</i>).	/users

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 OKTS 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 decribed 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/auth/login
4.4. Logout	POST a logout request for ending the active session with OKTS, and invalidating the issued <i>Security Token</i> .	/api/logout
4.5. Jobs	GET <i>Jobs</i> (according to <i>User's Role</i> and identity) and POST new <i>Job</i> (Id is auto-filled and modified entity is returned as result from POST request).	/api/job
4.6. Job	GET, PUT, DELETE <i>Job</i> with specified <i>jobId</i> .	/api/job/{jobId}
4.7 Pet's Jobs	GET all <i>Jobs</i> with the specified <i>petId</i>	/api/job/pet/{petId}
4.8 Owner's Jobs	GET all <i>Jobs</i> with the specified <i>ownerId</i>	/api/job/owner/{ownerId}
4.9 Carer's Jobs	GET all <i>Jobs</i> with the specified <i>carerId</i>	/api/job/carers/{carerId}
4.7. Job Results for Job	GET <i>Job Results</i> for <i>Job</i> with specified <i>jobId</i> .	/api/jobResult/job/{jobId}
4.8 Job Results for Pet	GET Job Results for Pet with specified <i>petId</i> .	/api/jobResult/pet/{petId}
4.9 Job Results for Carer	GET Job Results for Carer with specified <i>carerId</i> .	/api/jobResult/carers/{carerId}
4.10. Job Result	GET, PUT, DELETE <i>JobResult</i> with specified <i>jobResultId</i> .	/api/jobResult/{jobResultId}
4.11 Pets	GET <i>Pets</i> (according to <i>User's Role</i> and identity) and POST new <i>Pet</i> (Id is auto-filled and modified entity is returned as result from POST request).	/api/pets

4.12 Pet	GET, PUT, DELETE <i>Pet</i> with specified <i>petId</i> .	<i>/api/pet/{petId}</i>
-----------------	---	-------------------------