

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	Online Knowledge Jobing System
--------------	--------------------------------

1. Short project description (Business needs and system features)
<p><b>PupCare</b> 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 <b>React.js</b> as front-end, and <b>Node.js + express</b> as backend technologies. Each view will have a distinct URL, and the routing between pages will be done client side using <b>React Router</b>. The backend will be implemented as a <b>REST/JSON API</b> using JSON data serialization. There might also be an Android/iOS app (<b>React Native</b>) accompanying the website. The main user roles (actors in UML) are:</p> <ul style="list-style-type: none"> <li>• <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.</li> <li>• <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.</li> <li>• <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.</li> </ul>

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>
<b>2.2. Change User Data</b>	<p><i>Registered User</i> can view and edit her 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>Dog Owner</i>, <i>Dog Carer</i>, or <i>Administrator</i>.</p>	<i>Registered User</i> , <i>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, 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>
<b>2.4. Manage Jobs</b>	<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</i> , <i>Administrator</i>
<b>2.5. Add a Job</b>	<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</i> , <i>Administrator</i>
<b>2.6. Add/Edit Job</b>	<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</i> , <i>Administrator</i>
<b>2.7. Apply for a Job</b>	<i>Dog Carer</i> browses <i>Jobs</i> available and can sort and filter them using different <i>Job</i> metadata fields. <i>Dog Carer</i> chooses a <i>Job</i> to complete.	<i>Dog Carer</i> ,
<b>2.8. Select Dog Carer</b>	<i>Dog Owner</i> can select one of the <i>Dog Carers</i> applying for a <i>Job</i> .	<i>Dog Owner</i>

<b>2.9. Browse Job Results</b>	<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>
<b>2.10. Complete Job</b>	<p><i>Dog Carer</i> completes a <i>Job</i> by submitting a post-walk report:</p> <ul style="list-style-type: none"> <li>- questionnaire</li> <li>- uploaded photo of the walked pup</li> </ul>	<i>Dog Carer</i>

3. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
<b>3.1. Home</b>	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.	/
<b>3.2. Jobs</b>	Presents <i>Jobs</i> available according to <i>User's Role</i> and identity. Offers abilities to browse, choose, create, read, update, delete (CRUD) <i>Jobs</i> , as defined by <i>UCs</i> (for <i>Administrators</i> and <i>Dog Owners</i> only).	/jobs
<b>3.3. Apply Job</b>	Presents a <i>Job</i> if available to particular <i>Dog Carer</i> allowing them to apply for it.	/jobs/:id
<b>3.4. Job Results</b>	Presents <i>Job Results</i> available according to <i>User's Role</i> and identity as defined by <i>UCs</i> .	/job-results
<b>3.5. User Registration</b>	Presents a view allowing the <i>Anonymous Users</i> to register in <i>OKTS</i> .	/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> .	/profile
<b>3.8. Dashboard</b>	<p><i>Dog Carer</i> - Presents in real time the <i>Dog Carer's</i> progress on applied <i>Jobs</i>.</p> <p><i>Dog Owner</i> - Presents in real time the <i>Dog Owner's</i> progress on created <i>jobs</i>.</p>	/dashboard
<b>3.9. Users</b>	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
<b>4.1. Users</b>	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> .	<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> , according to restrictions decribed in UCs.	<i>/api/users/{userId}</i>
<b>4.3. Login</b>	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>
<b>4.4. Logout</b>	POST a logout request for ending the active session with OKTS, and invalidating the issued <i>Security Token</i> .	<i>/api/logout</i>
<b>4.5. Jobs</b>	GET users (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).	<i>/api/jobs</i>
<b>4.6. Job</b>	GET, PUT, DELETE <i>Job Data</i> (including <i>Questions</i> and <i>Answers</i> ) for <i>Job</i> with specified <i>jobId</i> .	<i>/api/jobs/{jobId}</i>
<b>4.7. Job Report</b>	GET <i>Job Result</i> (according to <i>User's Role</i> and identity) ) for <i>Job</i> with specified <i>jobId</i> , and POST new <i>Job Result</i> (Id is auto-filled and modified entity is returned as result from POST request).	<i>/api/jobs/{jobId}/report</i>
<b>4.8. Job Result</b>	GET, PUT, DELETE <i>Job Result</i> (according to <i>User's Role</i> and identity) for <i>Job</i> with specified <i>jobId</i> and <i>Job Result</i> with specified <i>jobReportId</i> .	<i>/api/jobs/{jobId}/results/{jobReportId}</i>