

Assignment 1	Project Summary
Course	Full-stack Development with Node.js and React.js - 2016

Project author			
№	First name, last name	E-mail	Face-to-face/ online
1	Kristiyan Todorov	sw.ktod@gmail.com	face-to-face

Project name	Social Network App
--------------	--------------------

1. Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
1.1. Browse information	The <i>User</i> can browse the information views (Home, User Data, About).	Registered users
1.2. Register	<i>Anonymous User</i> can register in the system by providing a username and a password. By default the role is set to user.	<i>Anonymous User</i>
1.3. Login	<i>Registered User</i> can login by providing a valid combination of username and password. User will be prompted to complete his profile every time, unless a valid unique email is set;	<i>Anonymous User</i>
1.4. Change User Data	<i>Registered User</i> can view and edit own personal <i>User Data</i> (first name, last name, email, date of birth and password). <i>Administrator</i> can view and edit <i>User Data</i> of all <i>Users</i> and assign them <i>Roles: User or Administrator</i> .	<i>User, Administrator</i>

1.5. Manage Posts	<p>User can browse all Posts, add new <i>Post</i>, edit own Posts, delete own Posts, as well as removing relation to other posts(provided the user is a post receiver).</p> <p><i>Administrator</i> can edit and delete any Post.</p>	<i>User, Administrator</i>
1.6. Manage Comments	<p>User can browse all Comments, add new <i>Comment for a Post or another Comment</i> , edit own Comments, delete own Comments as well as comments for his own posts.</p> <p>Administrator can edit and delete any Post</p>	<i>User, Administrator</i>

2. Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
2.1. Home	Presents all the user posts with their comments.	/
2.2. User Data	Presents ability to view and edit personal <i>User Data</i> and User Related Posts as well	/user/{username}
2.3. Users	Presents ability to manage Users and their User Data (available for Administrators only).	/users

3. API Resources (Node.js Backend)		
View name	Brief Descriptions	URI

4.1 Validate authentication	GET returns credentials on valid authentication	<i>/api/auth</i>
4.2. Register	POST new <i>User Data</i> - <i>username, password, confirm password</i> . (Id is auto-filled and modified entity is returned as result from POST request).	<i>/api/auth/register</i>
4.3. Logout	GET clears stored credentials	<i>/api/auth/logout</i>
4.3. Login	POST <i>User Data</i> - <i>username , password</i> . <i>Credentials returned on success</i>	<i>/api/auth/login</i>
4.4. Change password	PUT User Data - password, new password, confirm password	<i>/api/auth/password</i>
4.5. Users	GET <i>User Data</i> for all users, PUT Authenticated Users can update own User Data. Admin users can modify every users` data	<i>/api/users</i>
4.6. User	GET <i>User Data (first name, last name, email, date of birth)</i> for <i>User</i> with specified <i>username</i> .	<i>/api/users/{username}</i>
4.7. User related posts	GET User related posts (own posts, received posts, commented posts) with specified username.	<i>/api/users/{username}/posts</i>
4.8. Posts	GET Posts and POST new <i>Post</i> (Id is auto-filled by and modified entity is returned as result from POST request). PUT,DELETE posts is available if user is post owner or admin	<i>/api/posts</i>
4.9. Comments	GET Comments and POST new <i>Comment</i> (Id is auto-filled by and modified entity is returned as result from POST request). PUT,DELETE Comments is available if user is post owner, comment owner or admin	<i>/api/comments</i>