Assignment 1	Project Summary
Course	Fullstack Application Development with Node.js + Express.js + React.js - 2017

Project author			
№	First name, last name	E-mail	Face-to-face/ online
1	Anton Karamatov	antokara@abv.bg	face-to-face

Project name	OLKEKS

Short project description (Business needs and system features)
These days, people have less time to search for new recipes. After every delicious meal everyone deserves a top-level sweet dessert. OLKEKS is the place where they could find the best recipes that the world knows of so far. In addition to that it allows guests to search in our database, while users who register can submit new recipes and comments. The system will be developed as a *Single Page Application (SPA)* using *React.js* as front-end, and *Node.js + express.js* as backend technologies. The main user roles (actors in UML) are: • *Guest* – can only view the recipes; • *User* – can submit new recipes and comments;

Main Use Cases / Scenarios		
Use case name	Brief Descriptions	Actors Involved
Browse recipes	Everyone can browse the recipes available in our database.	All users
Register	*Guest* can register in the system by providing a username and password.	*Guest*
Manage Recipes	*User* can browse his/her *Recipes*, add new *Recipe* using *Add/Edit Recipe UC*, and delete a *Recipe*.	*User*

Add/Edit Recipe	*User* specifies/edits *Recipe* meta-data such as: recipe name and text and save the Recipe	*User*
Comment a Recipe	*User* can submit positive/negative feedback to the recipe creator via a comment under the recipe information.	*User*

Main Views (SPA Frontend)		
View name	Brief Descriptions	URI
RecipeList	Presents a landing page which allows everyone to search for a recipe.	/
User	Presents a page where Guests can create an account in our system.	/user
User	Presents a page where Guests can authenticate themselves in order to be able to comment and post new recipes.	/user
Recipe	Presents ability to manage (CRUD) Recipes and post comments.	/recipe/{recipeId}

API Resources (Node.js Backend)	
Brief Descriptions	URI
GET recipes and POST new *Recipe* (Id is auto-filled by *OLKEKS* and the created entity is returned as result from POST request).	/api/recipes
GET, PUT, DELETE *Recipe Data* for *Recipe* with specified *recipeId*.	/api/recipes/{recipeId}
POST a new comment for a *Recipe* with specified *recipeId*.	/api/recipes/comment/{recipeId}
POST to register a new user.	/api/users/register
POST to authenticate user and create user session.	/api/users/login
GET to destroy the current user session.	/api/users/logout