|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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  2 | Mihail Statelov  Ivanka Zaycheva | | [mstatelov@abv.bg](mailto:mstatelov@abv.bg)  vanqz1@outlook.com | face-to-face  face-to-face |

|  |  |
| --- | --- |
| Project name | Online recipe system |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Online recipe system is a system where users could find recipes, publish their own. Also they could rate and comment each others’ recipes. Users who are not registered will be able just to view recipes and the registered one will be able to publish, edit, rate and comment recipes. The system will be developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Node.js + express*** as backend technologies.  The main user roles (actors in UML) are:  • *Anonymous User* – can only view the recipes;  • *Registered* User– can add new recipes, edit his own one, rate and comment; |
|  |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse recipes** | Users will be able to view all published recipes. | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and choosing password. | *Anonymous User* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal *User Data*. | *Registered User* |
| * 1. **Add/Edit recipe** | *Registered users would be able to add new recipes and edit them. The form for adding recipe will have description, picture and description.* | *Registered User* |
| * 1. **Comment/Rate recipe** | *Registered user will be able to comment and rate published recipes.* | *Registered User* |
| * 1. **Search** | *All users will be able to search recipes by title* | *Registered User, Anonymous* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | A home page will have a toolbar where user could search for recipe by title, log in/edit profile. In the main part of the page on left there will be a list with facets (choose recipe with concrete category). On the middle part will be a list with recipes and pagination. | / |
| * 1. **Register** | A page where user will fill his/her to register; | */register* |
| * 1. **Edit profile** | A page where user will be able to see/edit his/her profile. | */profile/{userId}* |
| * 1. **Add and edit recipe** | Registered users will be able to add new recipe or edit existing one | */recipe/{recipeId}* |
| * 1. **Recipe from concrete category** | A page where users will see only a recipe from concrete category | */category/{categoryName}* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Recipes** | GET all recipes | */api/recipes* |
| * 1. **Recipe** | GET, PUT,DELETE a single recipe | */api/recipe/{recipeId}* |
| * 1. **Recipe** | POST add a new recipe | */api/recipe/add* |
| * 1. **Recipes by concrete category** | GET all recipes by concrete category | */api/recipe/category/{category}* |
| * 1. **Recipes by title** | GET all recipes which title contain a concrete phrase | */api/recipe/search/{searchString}* |
| * 1. **Rate recipe** | POST rate a concrete recipe | */api/recipe/{recipeId}/rate/{rate}* |
| * 1. **Comment recipe** | POST comment a concrete recipe | */api/recipe/{recipeId}/comment/{coment}* |
| * 1. **User** | POST add a new user | */api/user/add* |
| * 1. **User** | GET, PUT,DELETE get or edit user profile | */api/user/{userId}* |
| * 1. **Test Result** | GET, PUT, DELETE *Test Result* (according to *User's Role* and identity) for *Test* with specified *testId* and *Test Result* with specified *testResultId.* | */api/tests/{testId}/results/{testResultId}/* |