|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2022 | |
|  | | | |
| Project author | | | |
|  | Name | | FN |
|  | Ivan Tzvetomirov Ivanov | | 5MI3400182 |

|  |  |
| --- | --- |
| Project name | Cooking app (24kitchen fork) |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The project will be developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and Express.js as backend. 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. MongoDb for database and TypeScript for primary language. The main user roles (actors in UML) are:   * *Anonymous User* – can only view the recipes and comments. * *Registered User* – can *view, comment, rate and add recipes.* * Administrator – manage registered users and published content. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse recipes** | List recipes ordered by date, categories or chef.  User can search by categories, rating, date or selected chef. | All users |
| * 1. **Register user** | *Register user by email and username.*  *Each user can have followers.* | *Anonymous User,*  *Registered User* |
| * 1. **Login user** | *Login user by email or username.* | *Registered User* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal *User Data.* | *Registered User* |
| * 1. **Manage recipes** | *User can create, update and delete recipes and publish to selected category.*  *Еach recipe has a category, rating, list of necessary ingredients, the cooking of the recipe takes place step by step with the possibility of displaying photos or a link to a video from YouTube.* | *Registered User* |
| * 1. **Comment recipe** | User can comment and rate recipes. | *Registered User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Home page with latest recipes ordered by categories. | / |
| * 1. **Recipes** | All list recipes sorted by categories. | */recipes* |
| * 1. **Recipe** | Selected recipe by id. Can registered users comments. | */recipes/:id* |
| * 1. **User Sign up** | Presents a view allowing the *Anonymous Users* to sign up in website. | */signup* |
| * 1. **User Sign in** | Presents a view allowing the users to sign in in website. | */signin* |
| * 1. ***User Dashboard*** | *The user manages his profile, edits a list of recipes.* | */home* |
| * 1. **Admin *Dashboard*** | Manage users, recipes | */admin* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for current u*ser* with specified *id*. | */api/users/:id* |
| * 1. **Sign in** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/login* |
| * 1. **Sign out** | POST a logout request for ending the active session with *OKTS,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Recipes** | List of all recipes | */api/recipes* |
| * 1. **Recipe** | Get or create recipe | */api/recipes/:id* |