|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2020 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Instructor | | face-to-face |

|  |  |
| --- | --- |
| Project name | Flexible diet |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Flexible diet practitioners have a clearer idea of what macronutrients are contained in food, and know exactly how much they can afford without interfering with their diet, and are free from depression, regret and feeling of shame. Once you've learned how to keep track of your macronutrients, you'll be able to make your daily menu the way you like, fitting it into your individual lifestyle without having to exclude your favorite foods. Of course, this diet does not exclude the intake of healthy products, on the contrary, they are the basis. A flexible diet gives you tremendous freedom and extraordinary flexibility. Yes, the keyword is "flexibility" because we do not live in an optimal world where everything revolves around our diet and that we want to gain or lose weight. In such situation, you would risk missing another dinner with your parents or friends. The system will be developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Node.js + express*** as backend technologies. Each view will have a distinct URL, and the routing between pages will be done client side using ***React Router***. Set up UI project from scratch using: **Node**, **Webpack** – used to bundle or compile JavaScript into a single minified file that works in the browser. Webpack also includes a development web server, so I can serve my app locally. Modern JS and JSX with **Babel** used to transpile code, **ESLint**, **npm Scripts**, **Jest** and **Enzyme** for testing. The backend will be implemented as a ***REST/JSON API*** using JSON data serialization.The main user roles (actors in UML) are:  • *Anonymous User* – can only view the information pages and view few sample recipes  • *Customer* (extends *Registered User*) – can search for recipes and create recipes with calculated macronutrients for themselves only. As a customer you can edit your own user data.  • *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users.* Also as administrator you can search for recipes and create recipes with calculated macronutrients for all Registered Users.  • *Owner*  (*Developer*) – Maintain the system, make regular updates on UI experience and add more functionality to the REST API. Also as owner you are responsible for all private data of your users. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information and view sample recipes** | The *User* can browse the information views (Home, Recipes, About) in *Flexible diet app*, and can choose to view sample 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.  *Administrator* can register new by entering *User Data* and choosing a Role (*Customer* *or* *Administrator*). | *Anonymous User, Administrator,Owner* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal *User Data* as well as enter more recipes for themselves.  *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *Customer* or *Administrator*. | *Registered User, Administrator, Owner* |
| * 1. **Manage Users** | *Administrator* can browse and filter users based on different criteria: first and last name, email, Role.  *Administrator* can choose a *User* to manage, and can manage the chosen User – Create, read, update and delete (CRUD) operations | *Administrator,Owner* |
| * 1. **Manage All Recipes** | *Administrator* or Owner can add new recipes for all user to use. | *Administrator, Owner* |
| * 1. **Propose new recipes** | All registered user can propose new recipe. That request will be reviewed by Administrator and then approved for all customers or declined | *Registered User, Administrator, Owner* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | 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. | / |
| * 1. **Recipes** | Presents different types of meals (desserts, soups, pasta etc). And correspondingly 3.2.1 Presents desserts recipes  3.2.2 Presents soups recipes  3.2.3 Presents pasta recipes  Etc. | */recipes /recipes/desserts/recipes/soups /recipes/pasta* |
| * 1. **Calculate calorie intake** | Presents data about how many calories you should eat | */calculate-intake* |
| * 1. **Macronutrient calories** | Presents *list of* macronutrients of products contained in the recipes | */nutritions-cals* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register. | */join* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | Presents ability to view and edit personal *User Data* | */profile* |
| * 1. **Users** | Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only). | */users* |
| * 1. **User Details** | Presents ability to view (only allowed information - GDPR) and edit (Delete) User. This is managed by administrator. | */user/{userGuid}* |
| * 1. **About** | Presents information about the *Flexible diet* project and his owner. | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* (Guid is auto-filled and modified entity is returned as result from POST request). Available only for *Administrators*. | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *user Guid*. | */api/users/{userGuid }* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/login* |
| * 1. **Logout** | POST a logout request for ending the active session*,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Recipes** | GET a list of groups of different types of meals | */api/recipes* |
| * 1. **Desserts** | GET a list of desserts recipes | */api/recipes/{mealType}* |
| * 1. **Soups** | GET a list of soups recipes | */api/recipes/{mealType}* |
| * 1. **Pasta** | GET a list of pasta recipes | */api/recipes/{mealType}* |
| * 1. **Specific recipe** | GET, PUT, POST, DELETE recipe with specific recipe Guid | */api/recipes/{mealType}/{recipeGuid}* |
| * 1. **Propose new recipes** | POST recipe by user for administrator to approve. | */api/recipes/proposeNew* |
| * 1. **New local recipe** | POST new recipe from user for themselves to use | */api/recipes/new* |
| * 1. **Calculate calorie intake** | Get data about how many calories you should eat | */api/calculateIntake* |
| * 1. **Macronutrient calories** | Get *list of* macronutrients of products contained in the recipes | */api/nutritionsCals* |