|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2019 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Instructor | | face-to-face |

|  |  |
| --- | --- |
| Project name | Online exchange app |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The ***Online exchange app*** provides ability for users to publish different things for sale. The can also browser other users’ advertisements and deal with them for sale. In order to publish own advertisements and send messages to the sellers of other items, the user needs to register.  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***. The backend will be implemented as a ***REST/JSON API*** using JSON data serialization. There will be also a real-time event streaming from the server to the web client using ***Socket.IO*** in order to receive and send messages from/to the other users in real time.  The main user roles (actors in UML) are:   * *Anonymous User* – can only browse items * Registered User – can publish new items for sale and also send/receive messages with other users, in order to make a deal for sale/buy. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse items** | The *User* can browse and search items of all categories | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, username, and a password. | *Anonymous User* |
| **2.3. Create new advertisements** | *Registered User* can *can publish new items for sale.* | *Registered User* |
| * 1. **Manage Items** | *Registered users can edit already published items for sale.* | *Registered User* |
| * 1. **Send message for specific ad** | *Registered users can send messages in order to deal with the seller for a specific ad.* | *Registered User* |
| * 1. **Receive messages** | *Registered users will receive messages that other users send to them and will be able to read them and answer.* | *Registered User* |
| * 1. **Mark items as sold** | *Once the item uploader makes a deal with somebody, he can mark it as sold and this will make the advertisement marked as “archived”.* | *Registered user* |
| * 1. **Delete item completely** | Once an item is marked as archived, the user can decide to delete it forever. | Registered user |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents search field with categories for items, as well as a list of the most recent advertisements. | / |
| * 1. **Register** | Presents a view allowing the *Anonymous Users* to register in the system | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **Search Advertisements** | Presents a view where the user can select different categories, search keyword and more filters, which will be added to the query string. | */search?query* |
| * 1. **Detailed view for a specific ad** | Presents ability to *view an ad in details, browse its images, read its description, as well as a message form to the owner.* | */ad/:id* |
| * 1. **View with form to create a new ad.** | Presents a view with a form, which the user can fill, and then submit in order to create a new ad. | */create* |
| * 1. **Edit an already created ad** | This is exactly the same form from the /create view, but this time prefilled with the ad data. The user is able to further edit this fields and save changes. | */create/:id* |
| 3.8. **View a list with owned ads.** | Shows a list with the owned ads for the current user. Each list item contains a short preview and buttons for view / edit / archive | /myaccounts/ads |
| 3.9. **View a list with archived ads** | Shows all archived ads for the current user. Provides a button for delete / restore. | /myaccounts/ads/archived |
| 3.10. **View a list with messages** | Shows all received messages from other users. | /myaccount/messages |
| 3.11 **Chat room** | Shows all messages for the chat with the given id. Also has a field to submit new reply. | /myaccount/messages/:id |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Register** | POST *User Credentials* (e-mail address and password) in order to create a new user. | */users/register* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */users/login* |
| * 1. **Create ad** | POST information about a new ad. | */ad/create* |
| * 1. **Prefill edit field for ad** | GET the information for ad with the given id, which fits the creation form. | */ad/create/:id* |
| * 1. **Edit ad** | PUT changes *for the add with given ID.* | */ad/create/:id* |
| * 1. **Get ad** | GET request to for an ad with the given ID | */ad/:id* |
| * 1. **Get ads with a search query** | GET a list of ads with the given query | */ads/?query* |
| * 1. **Get all the ads for the current user** | GET all the ads that the currently logged user owns. | /user/ads |