

|              |   |
|--------------|---|
| Assignment 1 | Project Summary   |
| Course       | Fullstack Application Development with Node.js + Express.js + React.js - 2020 |

| Project author |           |                         |
|----------------|-----------|-------------------------|
| №              | Pseudonym | Face-to-face/<br>online |
| 1              | alien1976 | face-to-face            |

|              |                    |
|--------------|--------------------|
| Project name | Shopping assistant |
|--------------|--------------------|

| 1. Short project description (Business needs and system features)   |
|---|
| <p>Sometimes going to the shop becomes a nightmare when you have a list of products to buy but you need to walk through the shop a couple of times to find them. <b>Shopping assistant</b> will provide you the opportunity to find all the products from your list easily with no effort and give you the shortest path to them so your time won't be wasted anymore walking around. The system will be developed as a <i>Single Page Application (SPA)</i> using <b>React-Redux</b> as front-end, and <b>Node.js + express</b> as backend technologies. For database will be used Mongo DB to store all the relevant data that will be fetched from the API (for ex. Users, shops, etc.)</p> <p>The main user roles (actors in UML) are:</p> <ul style="list-style-type: none"> <li>• <i>Anonymous User</i> – can only view the information pages and choose from products in the selected shop so the path will be drawn.</li> <li>• <i>User</i> – can choose products, create different paths, save paths from previous shoppings, save the favorite products, can offer a new product that is not in the database</li> <li>• <i>Administrator</i> - Can do the same things as the user but can approve User offers for new products, can add new products to the shop</li> </ul> |

| 2. Main Use Cases / Scenarios                                |  |                                       |            |
|--|--|---------------------------------------|------------|
| Use case name  | Brief Descriptions   | Actors Involved                       | Estimation |
| <b>2.1. Browse information, select products, make a path</b> | <i>Users</i> can browse information views. They can search and select the products they want to buy from the shop they want. A map will be generated where the shortest route in the shop will be generated for them   | All users                             | 48h        |
| <b>2.2. Register</b>   | <i>Anonymous User</i> can register in the system by providing a valid e-mail address, first and last name, and chosen password.<br><br><i>The administrator</i> can approve the newly registered users   | <i>Anonymous User, Administrator</i>  | 8h         |
| <b>2.3. Change User Data</b>                                 | <i>Registered User</i> can view and edit her personal <i>User Data</i> , add favorite products, save routes from shops<br><br><i>The administrator</i> can view and edit <i>User Data</i> of all <i>Users</i>  | <i>Registered User, Administrator</i> | 8h         |
| <b>2.4. Manage Users</b>                                     | <i>The administrator</i> can browse and filter users based on different criteria: first and last name, email, Role.<br><br><i>The administrator</i> can choose a <i>User</i> to manage, and can manage the chosen User - edit (using Change User Data UC) or delete. | <i>Administrator</i>                  | 8h         |

|  |  |             |     |
|--|--|-------------|-----|
| <b>2.5. Add/Edit map</b>                   | <p><i>Users</i> can edit some of the saved maps and routes by choosing new products to the route.</p> <p><i>User</i> can create a route in a map of a shop with checkpoints where he/she could add a product to the checkpoint</p> | <i>User</i> | 16h |
| <b>2.6. Add/Edit Favorites</b>             | <i>User</i> can add new favorites products or edit the old one by reordering them or deleting them   | <i>User</i> | 4h  |
| <b>2.7. Start Route</b>                    | <p><i>Users</i> can start a custom made route or favorite route. Starting the route will make the map to me showed in full screen.</p> <p>(investigate how hard will be to add the exact location of the user to the map!)</p>     | <i>User</i> | 8h  |
| <b>2.8. Price of the selected products</b> | While the <i>User</i> is searching and selecting some products the price of all of them will be calculated automatically.  | <i>User</i> | 2h  |

Total hours estimation: 102

| 3. Main Views (SPA Frontend) |  |           |
|------------------------------|--|-----------|
| View name                    | Brief Descriptions   | URI       |
| <b>3.1. Home</b>             | Presents the introductory information for the purpose of the system as well as detailed instructions on how to start using it. Prominently offers the ability to register. | /         |
| <b>3.2. User Data</b>        | Presents the ability to view and edit personal <i>User Data</i> , as well as deregister from <i>OKTS</i> .   | /personal |

|                   |   |               |
|-------------------|---|---------------|
| <b>3.3. Users</b> | Presents the ability to manage (CRUD) Users and their User Data (available for Administrators only, as described in UCs). | <i>/users</i> |
| <b>3.4. Map</b>   | The shop map with the generated route of the selected products.   | <i>/map</i>   |

| 4. API Resources (Node.js Backend) |   |                                  |
|------------------------------------|---|----------------------------------|
| View name                          | Brief Descriptions  | URI                              |
| <b>4.1. Users</b>                  | GET <i>User Data</i> for all users, and POST new <i>User Data</i> (Id is auto-filled by OKTS and the modified entity is returned as a result from POST request). Available only for <i>Administrators</i> . | <i>/api/users</i>                |
| <b>4.2. User</b>                   | GET, PUT, DELETE <i>User Data</i> for <i>User</i> with specified <i>userId</i> .  | <i>/api/users/{userId}</i>       |
| <b>4.3. Maps</b>                   | GET users and POST new <i>Map</i> (Id is auto-filled by OKTS and the modified entity is returned as a result from POST request).  | <i>/api/maps</i>                 |
| <b>4.4. Map</b>                    | GET, PUT, DELETE <i>Map Data</i> for <i>Map</i> with specified <i>mapId</i> .   | <i>/api/maps/{mapId}</i>         |
| <b>4.5. Routes</b>                 | GET <i>Map</i> with specified <i>MapId</i> , and POST new <i>Route</i> (Id is auto-filled by OKTS and the modified entity is returned as result from POST request).   | <i>/api/maps/{mapId}/routes</i>  |
| <b>4.6 Products</b>                | GET users and POST new <i>Product</i> (Id is auto-filled by OKTS and the modified entity is returned as a result from POST request).  | <i>/api/products</i>             |
| <b>4.7 Product</b>                 | GET, PUT, DELETE <i>Product Data</i> for <i>Product</i> with specified <i>productId</i> .   | <i>/api/products/{productId}</i> |