|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assignment 1 | | Project Summary | | |
| Course | | Full-stack Development with Node.js and React.js - 2016 | | |
|  | | | | |
| Project author | | | | |
| № | First name, last name | | E-mail | Face-to-face/ online |
| 1 | Emil Vladimirov | | eavladimirov@abv.bg | face-to-face |

|  |  |
| --- | --- |
| Project name | Car Ads System |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| **Car Ads System(CAS)** helps to people to sell their car with posting ads, which are viewable from people who want to buy a car. **Car Ads System** helps both to people who want to sell their cars and people who want to buy a car. The system willbe developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Node.js + hapi*** as backend technologies.  The main user roles (actors in UML) are:  • *Anonymous User* – can only view the posted car ads.  • *Seller* (extends *Registered User*) – can view the posted car ads, can posts ads, can edit and delete his own ads.  • *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users*, as well as *ads*. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information and try sample tests** | The *User* can browse the information views (Home, Ads, About) in CAS SPA. | 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. By default all new registered users have *Seller* role.  *Administrator* can register new by entering *User Data* and choosing a Role (*Seller* or *Administrator*). | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal *User Data*.  *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *Seller* or *Administrator*. | *Registered User, Administrator* |
| * 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 - edit (using Change User Data UC) or delete.  *Administrator* can create a new user using *Register UC*. | *Administrator* |
| * 1. **Manage Ads** | *Seller* can browse her Ad*s*, add new Ad using *Add/Edit Ad*, and delete an Ad.  *Administrator* can browse Ads of all *Sellers*, edit and delete them. | *Instructor, Administrator* |
| * 1. **Add/Edit Ads** | *Seller* or *Administrator* specifies/edits *Ads* meta-data such as: ad name, subject, description, and tags. | *Instructor, Administrator* |

|  |  |  |
| --- | --- | --- |
| 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. **Ads** | Presents ads available according to *User's Role* and identity. Offers abilities to create, read, update, delete (CRUD), as well as browse. | */ads* |
| * 1. **User Data** | Presents ability to view and edit personal *User Data*, as well as deregister.. | */personal* |
| * 1. **Users** | Presents ability to manage (CRUD) Users and their User Data (available for Administrators only, as described in UCs). | */users* |
| * 1. **About** | Presents information about the *CAS* 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* (Id is auto-filled by *CAS* 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 *userId*. | */api/users/{userId}* |
| * 1. **Ads** | GET users (according to *User's Role* and identity) and POST new *Test* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request). | */api/ads* |