|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2021 | |
|  | | | |
| Project author | | | |
| F№ | Name | | Email |
| [26520](javascript:__doPostBack('rptRoles$_ctl2$lblRoleName','')) | Veronika Valentinova | | vgvalentinova@gmail.com |

|  |  |
| --- | --- |
| Project name | Travel log |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Travel Log is a platform for those who document and capture the world around us, helping people everywhere can experience the cultural and natural wonders of our planet and can feel more connected.  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.  The main user roles (actors in UML) are:   * Anonymous User   + can view the stories   + search story by specific destination   + leave comments on a post by providing name and email * Traveler (Registered User)   + can bookmark stories   + can follow channels   + can add and manage stories   + can receive emails for new interesting stories - Receive one story a day, at a time of your choosing * Administrator (extends Registered User)   + can manage (create, edit user data and delete) all registered Usersand stories |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse stories** |  | All users |
| * 1. **Search story by destination** |  | All users |
| * 1. **Leave comments** |  | All users |
| * 1. **Bookmark stories** | Bookmarked stories are saved in Registered user’s profile | Registered User, Administrator |
| * 1. **Follow channels** |  | Registered User, Administrator |
| * 1. **Register** | Anonymous User can register in the system by providing a valid e-mail address, first and last name, username and choosing password.  Administrator can register new by entering User Data | 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 | Registered User, Administrator |
| * 1. **Manage Users** | 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 Stories** |  | Administrator, Registered User |
| * 1. **Subscribe for one story a day, at fixed time** |  | Registered User, Administrator |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** |  | */stories* |
| * 1. **User Registration** |  | */register* |
| * 1. **Login** |  | */login* |
| * 1. **User Data** |  | */profile* |
| * 1. **Users** |  | */users* |
| * 1. **View story** |  | */{userName}/stories/{story}* |
| * 1. **View channel** |  | */channels* |
| * 1. **View channel stories** |  | */{channel}* |

|  |  |  |
| --- | --- | --- |
| 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). Available only for *Administrators*. | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *userId*, according to restrictions decribed in UCs. | */api/users/{userId}* |
| * 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 with *OKTS,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Read story** | GET/PUT/POST/DELETE | */api/stories/{story}* |
| * 1. **View list of channels** | GET | */api/channels* |
| * 1. **View channel** | GET/PUT/POST/DELETE | */api/channel/{channel}* |
| * 1. **Follow/Unfollow channel** | POST/DELETE | */api/channel/{channel}/follow* |
| * 1. **Bookmark** | POST/DELETE | */api/stories/{story}/bookmark* |