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

|  |  |
| --- | --- |
| Project name | Emazon - Online bookstore |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Emazon will provides ability for users to search and preview for books and authors. Also they can add a comment or add in favorites or add to their cart if they are registered. The system will developed as a Single Page Application using Reasct.js for frontend and Node.js + express for backend. Using React Router there will be client side routing between pages. The backend will be implemented as a REST/JSON API using JOSN data serialization. MongoDB / NoSQL database/ will be used for storing data.  The main user roles (actors in UML) are:  • *Anonymous User* – can only view and search for books and authors.  • Registered User – can view, search for books and authors. Also he can add comments, add in his favorite list and add to his cart. Moreover he can edit his profile information.  • *Administrator* (extends *Registered User*) – He can manage users, books, authors, comments (add, edit, delete). |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse books or authors** | The *User* can browse information for books and authors. | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address and choosing password.  *Administrator* can register new by entering *user information* | *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 browse and filter users based on different criteria: first and last name, email.  *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 Books** | *Administrator can add, edit, delete book information* | *Administrator* |
| * 1. **Manage Authors** | *Administrator* can add, edit, delete author information.  Also he can assign book to author. | *Instructor, Administrator* |
| * 1. **Advanced Search** | *All Users can search books or authors with multiple criteria* | *Instructor, Administrator* |
| * 1. **Add to favorites** | *Users can add to their favorite’s list.*  *If Administrator delete a book it should be removed from all lists.* | *Instructor, Administrator* |
| * 1. **Add to cart** | *Users can add to cart.* | *Instructor, Administrator* |
| * 1. **Add/ Edit book** | *Administrator can add or edit book information* | *Administrator* |
| * 1. **Add/ Edit**   **Author** | *Administrator can add or edit author information* | *Administrator* |
| * 1. **Remove book/ author** | *Administrator can delete book or author* | *Administrator* |
| * 1. **Add comment and rating to book** | *Administrator or Registered User can add comment and rating to book* | *Administrator, Registered User* |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents list of top books with biggest rating. Also list of part of the authors. | / |
| * 1. **Administration console** | Presents administration console with lists of all books, authors and users. | */admin-console* |
| * 1. **Advanced Search** | Presents to the users ability to search with more than one filters | */advanced-search* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register. | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Profile** | Presents ability to view and edit personal *User Data*, to view their favorite books | */profile* |
| * 1. **Authors** | Presents ability to browse through all authors | */authors* |
| * 1. **Books** | Presents ability to browse through all books in the system | */books* |
| * 1. **Users** | Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only, as described in UCs). | */users* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users. Available only for *Administrators*. | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *userId* | */api/users/:userId* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) | */api/login* |
| * 1. **Logout** | POST a logout request for ending the active session | */api/logout* |
| * 1. **Books** | GET *all books* | */api/books* |
| * 1. **Book** | GET, PUT, DELETE *books* | */api/books/:bookId* |
| * 1. **Authors** | GET authors | */api/authors* |
| * 1. **Author** | GET, PUT, DELETE *auhtor* | */api/author/:authorId* |