|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2020 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | OCB | | face-to-face |

|  |  |
| --- | --- |
| Project name | Levski Sofia Fan Zone |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Most of the people in Bulgaria are interested in football especially in Levski Sofia. In this times where people should stay in home because of the virus, there is a need of fun. Here everyone of you can test their knowledge of the game and win great prizes. In addition to that it allows users to register, and administrators to manage them. 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 only view the information pages and see what he can win.  • User/Fan (extends *Registered User*) – can choose a Quiz to complete or register a ticket  • *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users*, as well as *Quizzes* and *Ranking*. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | The *User* can browse the information views (Home, Quizzes, Ranking, Prizes). | 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.  *Administrator* can register new by entering *User Data* and choosing a Role (*User* or *Administrator*). | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered User* can view and edit his personal *User Data*  *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *User* or *Administrator*. | *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* |
| **5.Add/Edit Quiz** | *Administrator* specifies/edits *Quiz* meta-data such as: quiz name, description, tags. *Administrator* creates/edits *Quizzes* using *Create/Edit Question UC,* and can delete chosen *Question. Administrator* saves the edited *Quiz.* | *Administrator* |
| 1. **Complete Quiz** | User/Fan browses quiz available and can sort and filter them using different test metadata fields.  User/Fan chooses a quiz to complete. | User/Fan |
| 1. **Browse Quiz Results** | *User/Fan* can browse his own completed *Quiz Results*. | *User/Fan* |
| 1. **Add ticket codes** | *User/Fan can add ticket codes and earn points for the ranking.*  *Administrator can edit ticket-codes* | *User/Fan, Administrator* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information of the store as well as instructions how to start using it. Prominently offers ability to register. | / |
| * 1. **Quizzes** | Presents quizzes to play and earn points for the ranking. | */quiz* |
| **3. Register ticket codes** | Presents a view that allows Registered Users to register codes from tickets to earn Points | */ticket-codes* |
| 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 Data** | Presents ability to view and edit personal *User Data.* | */profile* |
| 1. **Ranking** | Presents a view allowing the users to see the ranking | */ranking* |
| 1. **Prizes** | Presents ability to view the prizes available to win. | */prizes* |
| 1. **Users** | Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only, as described in UCs). | */users* |
| 1. **Manage quizzes** | Add and edit quizzes for the users (available only for Administrators) | */add-quiz* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* .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. | */api/logout* |
| 1. **Quiz** | GET, PUT, DELETE *Quiz Data* (including *Questions* and *Answers*) for *Quizzes* with specified *testId*. | */api/quiz/{testId}* |
| 1. **Ticket codes** | GET, PUT, DELETE TicketCodes (only for *Administrators*) | */api/ticket-codes* |