|  |  |  |
| --- | --- | --- |
|  | **Софийски университет „Св. Кл. Охридски”**  Факултет по математика и информатика  *Бакалавърска програма „Информационни системи”* |  |

**Предмет: Fullstack React**

***Летен семестър, 2018/2019 год.***

**Тема: „Система менажиране на стаи“**

**Курсов проект**

*Автор:*

*Боян Ракиловски, номер 71518*

Март, 2018

София

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

|  |  |
| --- | --- |
| Project name | FMI room reservation system |

|  |
| --- |
| * Short project description (Business needs and system features) |
| The application will help teachers to reserve a room very fast (by scanning a QR code and filling a simple form). When the room reservation is changed on spot and differs from the weekly schedule the students and guest can check that in the system. Also there will be statistics for the most popular rooms.  The system will be developed as a Single Page Application (SPA) using **React.js** and **Redux** for the front-end, and **Node.js** + **Express** as backend technologies. For data storage will be used **mongodb** with **mongoose** extension.  The main user roles (actors in UML) are:  • Anonymous User(Student or guest) – can search for reservations by a criteria (room number, tracher name, discipline)  • Teacher – can create Reservations and search for a room by a criteria.  • Administrator – can manage (create, edit user data and delete) all Registered Users, as well as Rooms. |
|  |

|  |  |  |
| --- | --- | --- |
| * Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * **Manage Users** | Administrator can browse and filter users based on different criteria: first and last name, email, discipline.  Administrator can Add/Edit/Delete and View details for user account. | Administrator |
| * **Manage Rooms** | Administrators can browse the Rooms, add new Room using Add/Edit, and delete a Room, as well as view the Room details. | Administrator |
| * **Manage Reservations** | Teachers can browse the Reservations, add new Reservation using Add/Edit, and Delete a Reservation, as well as view the Reservation details. | Teacher |
| * **View chart** | All users and guest can see the charts at the home page. | All Users and guests |
| * **Search for users** | All registered users can search for teacher accounts by search criteria. | All Users |
| * **Search for rooms** | Administrators can search by a criteria and get as a result a list of rooms. | Teacher, Administrator |
| * **Search for reservations** | All users and guests can search for reservations by a criteria and view the results. | All Users |

|  |  |  |
| --- | --- | --- |
| * Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * **Home** | Renders a dashboard container Component with Reservations component and Chart component inside. Prominently offers ability to register.  Charts will update using web sockets in order to display the most popular rooms in real time. | / |
| * **Reservations** | Presents a list of reservations. Offers abilities to create, read, update, delete (CRUD), as well as browse Reservations for each Reservation subject for every user or guest.  Renders search from in order to search for reservations by given criteria. | /reservations/ |
| * **Add reservation** | Presents a from with radio buttons for reservation duration and an input field for the discipline.  A request to this URL is made when a user is registered with a teacher account and scans a room QR code.  If the user isn’t loged in then a redirect to sign in page is made.  The teacher name is known, because only registered teachers can reserve a room. | /room/:id/reservation/add/ |
| * **User Data** | Presents ability to view and edit personal User Data. (The edit functionality is only for Administrator role). User data contains personal QR code. | /profile/ |
| * **Users** | Presents a list of users and gives the ability to manage (CRUD) Users and their User Data (available for Administrators only).  Renders search from in order to search for users by given criteria. | /users/ |
| * **Rooms** | Presents a list of rooms. Offers abilities to create, read, update, delete (CRUD), as well as browse Rooms for each Room subject depending on the user role.  Room details displays the room QR code as well as room number and indicators if the room has a proector and if the room has computers.  Renders search from in order to search for rooms by given criteria. | /rooms/ |
| * **qrToPdf** | Renders all QR codes for rooms and Teachers in a grid and gives the possibility to the user to export them in pdf format. | /qrToPdf/ |

|  |  |  |
| --- | --- | --- |
| * API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * **Users** | All registered users can get a list of teacher accounts by search criteria using GET request. | /api/users/?teacherName=x&email=y |
| * **User** | Administrator can use the following queries to manage user entities: POST, PUT, DELETE.  Users can sign in and log out from the system using POST and GET queries.  Teachers have to scan their QR code in order to access a page with possibility to enter a password and after that use their account. | /api/users/  /api/users/:userId/  /api/signin/  /api/signout/  /users/:id/ |
| * **Rooms** | Administrator and teacher can search for a room based on search criteria or for a room with specific number. | /api/rooms/?computerRoom=x&hasProector=y  &roomNumber=z  /rooms/:roomId |
| * **Room** | Administrator can manage rooms with the following queries: POST, PUT, DELETE. | /api/rooms/  /api/rooms/:roomId/ |
| * **Reservations** | Everyone can get a list of reservations based on a search criteria or for a given room (including scanning a QR code for the room) | /rooms/reservations/?roomNumber=x&discipline=y  &teacherName=z&startDate=xx&endDate=yy  /api/rooms/reservations/ |
| * **Reservation** | Teachers can manage reservations with the following queries: PUT, DELETE, POST. | /api/rooms/:roomId/reservations/:resevationId/  /api/rooms/:roomId/reservations/ |