|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2019** | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Bibi | | face-to-face |

|  |  |
| --- | --- |
| Project name | Office events |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The ***Office events application*** provides ability for companies to create events, visible for the whole company or only for team, to remind their employees for important activities and to organize different rallies and trainings. In addition to that it allows users to register, and administrators and managers to manage them.  The system will be developed using ***Spring 5 Application Development Framework and Spring Boot***.It will implement web-based front-end client using ***Angular CLI*** and **TypeScript** libraries for asynchronous page data updates. The application will have pure backend (**Spring**) and frontend (**Angular** **CLI**)  The backend will be implemented as a ***REST/JSON API*** using JSON data serialization. The main user roles are:  • *Anonymous User* – can only view the homepage with short description about the web applications  • *Dev* (extends *Registered User*) – can create events visible for the whole company and for his team and also only for him, delete and edit his own events, see all company events and these for his team  • Manager (extends Registered User) – can do what a dev can , delete and edit all events in his team, add and remove team member from his team  • *Admin*(extends *Registered User*) – can do what a manager can, change user role, create new teams, add managers to teams and remove them, create, delete, edit new rooms for events |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | The *User* can browse the information for the applications. | 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 ***DEV Use***r role.  ***Admin*** can register new by entering *User Data* and choosing a Role (***Dev****,* ***Manager***, or ***Admin***). | *Anonymous User, Admin* |
| * 1. **Change User Data and Role** | *Registered User* can view and edit her personal *User Data.*  ***Admin*** can view and edit *User Data* of all *Users* and assign them or changes *Roles*: ***DEV****,* ***MANAGER***, or ***ADMIN***. | *Registered User, Admin* |
| * 1. **Manage Users** | ***Admin*** can browse and filter users based on different criteria: first and last name, email, Role.  ***Admin*** can create a new user using *Register UC*. | *Admin* |
| * 1. **Manage teams and team members** | ***Admin*** *can browse and create/delete a new team*. After that they can assign to it a **Manager**. *One team can have only one manager. One manager can manage more than one team.*  ***Manager*** *can browse and add/remove* ***Dev*** *to the teams. One* ***Dev*** *can be in only one team.* | *Admin, Manager* |
| * 1. **Manage Rooms** | ***Admin*** *can create, edit and remove new rooms for events.* | *Admin* |
| * 1. **Team search** | ***Admin*** *and* ***Manager*** *can browse and search teams and see all team’s member.* | *Manager, Admin* |
| * 1. **Add own events** | *All Registered users can create events visible only for them. These events will not be assigned to any room. The event has title, description, data and room.* | All users |
| * 1. **Add team events** | ***Dev*** *and* ***Manager*** *can create a team event only if they are assigned to any team.* | *Dev, Manager* |
| * 1. **Add company events** | ***Dev****,* ***Admin*** *and* ***Manager*** *can crate events visible for the whole company.* | *Dev, Admin, Manager* |
| * 1. **Edit/Delete events** | *All registered users can edit/delete their own events.* ***Dev*** *can edit/delete his own team events.* ***Manager*** *can edit/delete its team events.* ***Admin*** *can delete/edit all events.* | *Depend of role* |
| * 1. **View events** | ***Dev*** *and* ***Manager*** *can see all events in their team.* ***Admin*** *can see all events in the app. The company events are visible for all. Own events are visible only for creator.* | *Depend of role* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information for the purpose of the system as well as detailed instructions on how to start using it. Prominently offers ability to register. | / |
| * 1. **Company Events** | See all company events. (edit depend of role right) | */company-events* |
| * 1. **Team Events** | See all team events. (edit depend of role right | */team-events* |
| * 1. **My Events** | Can create edit and delete own events. | */my-events* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register in *the applications* | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | Provides ability to view and edit personal *User Data*. | */personal* |
| * 1. **Admin Panel** | See all users in the system and change their role. (It can be merged with **/users** ) Delete all events in the whole applications. | */admin-panel* |
| * 1. **Dashboard** | Show the events that will happened the current week. Filter. Amount | */dashboard* |
| * 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 project and his owner. | */about* |
| * 1. **Manager Panel** | Can edit and delete its team events. | */manager-panel* |

|  |  |  |
| --- | --- | --- |
| 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 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 described in UCs. | */api/users/{userId}* |
| * 1. **User Role** | GET,PUT Change user role with specified *userId* to Dev, Manager or Admin. | */api/users/role{userId}* |
| * 1. **Login/Register** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/login*  */api/register* |
| * 1. **Logout** | POST a logout request for ending the active session with *OKTS,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Teams** | GET POST Get all teams or create e new one | */api/teams* |
| * 1. **Team** | GET PUT DELETE team with specified teamId | */api/teams/{teamId}* |
| * 1. **Team Dev** | GET PUT DELETE a dev to a team with specified teamId | /api/teams/{teamId}/dev |
| * 1. **Team Manager** | GET PUT DELETE a manager to a team with specified teamId | /api/teams/{teamId}/manager |
| * 1. **Team Events (Pagination)** | GET PUT POST DELETE the events of team with specified teamId | /api/ events /{teamsId}/?room=”Some”  /api/ events /{teamsId}?page=2/?limit=2 |
| * 1. **Company Events**   **(Pagination)** | GET PUT POST DELETE the events of the whole company | /api/ events /company?room=”Some”  /api/ events /company?page=2/?limit=2 |
| * 1. **User Events**   **(Pagination)** | GET PUT POST DELETE the events of some user | */api/* events */{userId}*?room=”Some”  /api/ events */{userId}*?page=2/?limit=2 |
| * 1. **Rooms** | GET PUT POST DELETE room | */api/rooms*  *{body}* |