|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Multimedia web technologies with Angular and Typescript - 2019 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Tod | | face-to-face |

|  |  |
| --- | --- |
| Project name | Project task management system |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Assigning and tracking project task progress is made much simpler and accessible thanks to the Project Task Management System (PTMS). It provides ability for the project manager(admin) to assign people to projects, schedule meetings and checking the progress status of each workers given task. In addition workers can in turn see the projects they are delegated to and fill in the progress of the tasks they are working on. The system will be developed as a *Single Page Application (SPA)* using ***Angular*** as front-end, and ***Java***(or ***Node.js + express*** if Java is not accepted) as backend technologies.  The main user roles (actors in UML) are:  • *Administrator(project manager)* – can view all the users, projects, meeting and tasks. He can create any number of projects, meetings and tasks and can assign any number of people to a project.  • *Ordinary user(worker) – can only have access to projects he is assigned to, together with the meetings and tasks. He can change completion status of a task given to him.* |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | Browse information about Users, Projects, Meeting and Tasks by administrator, browse information about delegated task by user | All users |
| * 1. **Register** | *Administrator* can register new by entering *User Data* and choosing a Role (*User* or *Administrator*). | *Administrator* |
| * 1. **Change User Data** | *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *User* or *Administrator*. | *Administrator* |
| * 1. **Manage Users** | *Administrator* can browse and filter users based on different criteria: first and last name, email, Role.  *Administrator* can choose a *User* to manage, and can manage the chosen User - edit or delete. | *Administrator* |
| * 1. **Manage Projects** | *Administrator* can browse and filter projects based on different criteria: title or key  *Administrator* can choose a *Project* to manage, and can manage the chosen Projects – edit, delete and assign people to the corresponding project. | *Administrator* |
| 1. **Manage Meetings** | *Administrator* can browse and filter Meetings based on single criteria: key  *Administrator* can choose a *meetings* to manage, and can manage the chosen Meetings - edit or delete. | *Administrator* |
| 1. **Manage Tasks** | *Administrator* can browse and filter tasks based on different criteria: key and status  *Administrator* can choose a *Tasks* to manage, and can manage the chosen Task - edit or delete.  Users assigned to certain project can update the task completion information | *Administrator, Users* |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information for the purpose of the system as well as detailed instructions how to start using it. Prominently offers ability to register. | / |
| * 1. **Users** | Presents user information and offers abilities to create, update, delete (CRUD), available according to *User's Role* and identity. | */users* |
| * 1. **Projects** | Presents project information and offers abilities to create, update, delete (CRUD) and assign project to a worker, available according to *User's Role* and identity. | */projects* |
| * 1. **Meetings** | Presents meeting information and offers abilities to create, update, delete (CRUD), available according to *User's Role* and identity. | */meetings* |
| * 1. **Tasks** | Presents task information and offers abilities to create, update, delete (CRUD), available according to *User's Role* and identity. | */tasks* |

|  |  |  |
| --- | --- | --- |
| 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*. | */usersResource* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *userId*. | */usersResource/{userId}* |
| * 1. **Projects** | GET projects (according to *User's Role* and identity) and POST new *Project, which is available only for Administrators*. | */projectsResource* |
| * 1. **Project** | GET, PUT, DELETE *Project Data* for *Project* with specified *testId*. | */projectsResource/{projectId}* |
| * 1. **Meetings** | GET *meetings* (according to *User's Role* and identity) ) and POST new *Meetings, which is available only for Administrators*. | */meetingsResource* |
| * 1. **Meeting** | GET, PUT, DELETE *Meeting Data* (according to *User's Role* and identity) and POST new Meeting, which is available only for Administrators. | */meetingsResource/{meetingId}* |
| * 1. **Tasks** | GET *tasks* (according to *User's Role* and identity) ) and POST new *Tasks, which is available only for Administrators*. | */tasksResource* |
| * 1. **Task** | GET, PUT, DELETE *Test Data* (according to *User's Role* and identity) and POST new Tasks, which is available only for Administrators. And update project task completion, available to every user working on that task + administrator. | */tasksResource/{taskId}* |