z

|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2018** | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | aga | | Online |
|  |  | |  |
|  |  | |  |
|  |  | |  |

|  |  |
| --- | --- |
| Project name | Rare Species Posting and Reviewing |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| (**RSPR**) is a system where people can post, rate and comment different posts for rare species. The system will be developed using ***Spring 5 Application Development Framework (SpringMVC)*** in its backend part. The frontend part will be realized with **JSP/Bootstrap**. Database layer will be realized with **Hibernate,** **PostgreSQL**.  The main user roles are:  • *Anonymous User* – can only query and view post and register as regular user.  • User – can query, view, post,comment and edit(created by him) posts  • *Administrator* – can query,view,comment and edit any post and user. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | Any user can browse the information views (Home, Register/Login, About,Posts). | All users |
| * 1. **User login/register** | *Anonymous User* can login into the system using email and password  User can register using registration form. After registration user must verify his account using link sent to his/hers email.  The activation will happen with generated token binded to the actual user | *Anonymous User* |
| * 1. **Change User info** | *User* can change his personal infomration as his/hers real name, city, favourite species, etc. In addition they can hide their personal information so nobody except administrators can see | *User, Administrator* |
| * 1. **View posts** | Anyone can query posts for species based on species type(tag), time created, person who created it or by title. | *All users* |
| * 1. **Create/Edit/Delete post** | Every User and Aministrator can create posts as an html like format. (not direct html format, but one serialized to html)  Every User can edit/delete their posts.  Admin can edit/delete any post. | *User, Administrator* |
| * 1. **Review User profile** | Every User can view other users profiles and their details(if enabled by the user).  *Administrator* can only view User’s personal information, but cannot change it | *User, Administrator* |
| * 1. **Post/Edit/Delete/Rate Comment** | Users and administrators can post comments and like/dislike them. Users can edit and delete their comments. Administrators can edit and delete any comment. | *User, Administrator* |
| * 1. **View all users** | Administrators can view all users and query them by name and email. | *Administrator* |
| * 1. **Remove user** | Administrators can remove users. Mainly if they use inappropriate language in comments or posts | *Administrator* |
| * 1. **Add/Remove admin** | Administrators can add/remove other administrators | *Administrator* |

|  |  |  |
| --- | --- | --- |
| 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 how to start using it. Prominently offers ability to log . | / |
| * 1. **User/Admin Login** | Presents a view allowing anonymous users to login as User or Admin. | */login* |
| * 1. **User Details** | Presents the details for the logged in User or selected user.  Also allows allows editing the information for current user. | */user/details/{user}* |
| * 1. **User Register** | Presents a view allowing anonymous users to register using email and pasword. | */register* |
| * 1. **User Activation** | Presents a view for Users activation. | */activate* |
| * 1. **Posts** | Presents a view for viewing/filtering posts. | */posts* |
| * 1. **View Post/Comments** | Presents a view for detailed view of posts (and comments). | */post/view/{post}* |
| * 1. **Edit Post** | Presents a view for editing post. | */post/edit/{post}* |
| * 1. **Edit comment** | Presents a view for editing comment. | */comment/edit/{comment}* |
| * 1. **Manage users** | Presents a view for all users. | */admin/users* |
| * 1. **Manage admins** | Presents a view for managing admins. | */admin/admins* |
| * 1. **User Feed** | Presents a view for viewing all posts of a user | */user/post/{user}* |
| * 1. **About** | Presents information about the OTQR project and it’s owners. | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Java Spring Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET User *Data* for all users, and POST new User *Data*. Available for *Administrators* | */api/users* |
| * 1. **User** | GET, POST, DELETE *User Data* for *User* with specified id*.* | */api/users/{id}* |
| * 1. **Login** | POST *User Credentials* (PN and generated code) and receive a valid *Security Token* to use in subsequent API requests. | */api/login* |
| * 1. **Logout** | POST a logout request for ending the active sessionand invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Posts** | *GET alll posts* | */api/posts* |
| * 1. **Post** | GET,POST,DELETE all posts with filter. | */api/posts/{filter}* |
| * 1. **Comments** | *GET,POST,DELETE all comments by post id* | */api/comments/{filter}* |
| * 1. **Activate User** | Activate user by token | */api/activate/{token}* |
| * 1. **Remove User** | Remove user by id. | */remove/{id}* |
| * 1. **Add as admin** | Add user as admin by id | */admin/add/{id}* |
| * 1. **Remove as admin** | Remove user as admin | */admin/remove/{id}* |
| * 1. **Get User Details** | GET,POST user details by id | */user/details/{id}* |