**Rare Species**

**Posting and Reviewing**

**By Andrey Andreev**

**(№ 62173)**

**Sofia University**

**Web Application & Service Development**

**with Spring Framework v5**

**2018**

**1. Short Description**

Rare Species Posting and Reviewing (**RSPR**) is a system where people can post, rate and comment different posts for rare species.

**2. Use cases**

* Browse information: Anonymous User can browse the information views (Home, Register/Login, About).
* User login/register: *Anonymous User* can login into the system using username and password. User can register using registration form. After registration user must verify his account using created from registration token.
  + 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.
  + View posts: Any reigstered user can query posts for species based on species type(tag), time created, person who created it or by title.
  + 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.
  + 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.
  + Post/Edit/Delete/Rate Comment: Users and administrators can post comments. Users can edit and delete their comments. Administrators can edit and delete any comment.
  + View all users: Administrators can view all users and query them by name and username.
  + Remove user: Administrators can remove users. Mainly if they use inappropriate language in comments or posts
  + Add/Remove admin: Administrators can add/remove other administrators.

**3. Used Technologies**

1. Frontend

* Thymeleaf
* Bootstrap 4
* JavaScript

2. Backend

* SpringMVC
* MySQL

**4. MVC Architecture Description**

The project’s architecture is MVC based. Controllers create models

using services which communicate with data repositories and manage buisness logic. After that they pass those models to view templates which render them to html pages.

Frontend Views:

1. Home

endpoint: /

description: Presents the introductory information.

2. User Login

endpoint: /login

description: Presents a view allowing Anonymous Users to login as User or Admin.

3. User

endpoint: /user

description: Presents a view for current user’s information and an editing form

4. User Details

endpoint: /user/details/{user}

description: Presents the details for the logged in User or selected user.

5. User Register

endpoint: /register

description: Presents a view allowing anonymous users to register using email and pasword.

6. User Activation

endpoint: /activate

description: Presents a view for Users activation.

7. Posts

endpoint: /posts

description: Presents a view for viewing/filtering posts.

8. View Post/Comments

endpoint: /post/view/{post}

description: Presents a view for detailed view of posts (and comments).

9. Create post

endpoint: /post/create/

description: Presents a view for creating a post

10. Edit post

endpoint: /post/edit/{post}

description: Presents a view for editing post.

11. Manage users

endpoint: /users

description: Presents a view for all users.

12. About

endpoint: /about

description: Presents information about the RSPR project and it’s owners.

**5. REST API Architecture Description**

The REST API is used to serve interface for managing server’s resources. It exposes endpoints which are managed by API Controller and server’s services underneath.

REST API Endpoints:

1. Users

endpoint: */api/user/users*

*description: GET User Data for all users, and POST new User Data. Available for Administrators*

2. User

endpoint: */api/user/{user}*

*description: GET, DELETE User Data for User with specified id.*

3. User Details

endpoint: */api//user/details/{user}*

*description: GET,POST user details by id*

4. Activate User

endpoint: */api/activate/{token}*

*description: Activate user by token*

5. Post

endpoint: */api/user/activate/{token}*

*description: Activate user by token*

6. Post Comments

endpoint: */api/post/{post}/comments*

*description: GET post comments*

*7. Comments*

endpoint: */api/comment/{comment}*

*description: PUT,POST,DELETE comments*

*8. Change User Role*

endpoint: */api/user/role/{user}/{role}*

*description: Change user’s role*

***6. Installation***

*1.* ***application.properties*** *is edited with current system’s information:*

* *Server’s port*
* *MySQL endpoint,user and password*

*2.* ***mvn package*** *command is run to compile project to jar*

*3. Project jar is started to start the system*

**7. Usage**

User registers, then activates his account and then logs in into the system. Users can then view, create and manage posts. They can view other users information if it’s shown and can edit their own.

**8. Conclusion**

**(RSPR)** is a new post/based project targeting Animal lovers community. The project’s perspective is to gather a large community of people who can share information about different rare species and etc.