|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2018** | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | GIS | | face-to-face |
| 2 | MVD | | face-to-face |
| 3 | SBT | | face-to-face |

|  |  |
| --- | --- |
| Project name | The Spring Social Network |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| In today’s day and age social networks are the most used type of website. But how many of them are running on Spring? **The Spring Social Network** or **SSN** will allow Spring enthusiasts all around the world to connect and share their experiences. The back-end of SSN will, of course, be developed with the **Spring 5 Framework** and will be a REST API that returns data using the JSON format**.** The front-end will be servlet based, using **Thymeleaf** and/or **JSP**. There will be real-time data streaming for notifications that relies on **Websockets** or **SSE** for the back-end and **JS** for the front-end. The main user roles are:  • *Unregistered User* – can only view the main page and register.  • *Registered user* – can create/edit/delete posts, comments and groups. Can add/remove friends, can view his/her feed. Can view other users’ profile and can edit or delete his/her profile.  • *Administrator*– can delete posts, groups, comments, can ban or delete users. Can administrate the application settings. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Register** | An Unregistered User can register by providing a first and last name, a valid e-mail address and choosing a password. | Unregistered User |
| * 1. **Manage posts** | *Registered user* can create/edit/delete his/her posts and view all posts of friends and from groups they have joined. They can also attach a picture with their post.  *Administrator* can delete posts at their discretion. | *Registered User, Administrator* |
| * 1. **Change User Profile** | *Registered User* can view and edit their profile. | *Registered User, Administrator* |
| * 1. **Manage Users** | *Registered User* can delete their profile and all associated data with it.  *Administrator* can choose to banor delete users and all associated data. | *Registered User, Administrator* |
| * 1. **Manage Groups** | *Registered User* can create groups and edit/delete their groups.  *Administrator* can delete groups. | *Registered User, Administrator* |
| * 1. **Manage Comments** | *Registered User* can create comments and edit/delete their comments. They can also attach a picture with their comment.  *Administrator* can delete comments. | *Registered User, Administrator* |
| * 1. **View user feed** | *Registered User* can view their feed which is all the posts and associated comments from them or their friends. | *Registered User* |
| * 1. **View group feed** | *Registered User* can view the feed of a public or a group they have joined. | *Registered User* |
| * 1. **Manage friends** | *Registered User* can add or remove another *Registered User* as his/her friend. | *Registered User* |
| * 1. **Administrate application settings** | *Administrator* can change the application settings | *Administrator* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Front-end) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Visualizes the current Registered User’s feed. The user can add a post or comment on a post from his/her feed or delete his/her own posts. | / |
| * 1. **Login** | Visualizes a login view in which Registered Users can login. | */login* |
| * 1. **Register** | Visualizes a register view in which Unregistered Users can register. | */register* |
| * 1. **Profile** | Visualizes a profile view of either the current Registered User, which he/she can edit or another Registered User’s profile | */profile/{profileId}* |
| * 1. **Group** | Visualizes a group view of either the feed of a group in which the Registered User participates or a view to join some group. If the Registered User is an owner of the group, he/she can also administrate it. | */group/{groupId}* |
| * 1. **Admin users** | Visualizes a view in which the admin can manage users. | */admin/users* |
| * 1. **Admin posts** | Visualizes a view in which the admin can manage posts. | */admin/posts* |
| * 1. **Admin groups** | Visualizes a view in which the admin can manage groups. | */admin/groups* |
| * 1. **Admin application** | Visualizes a view in which the admin can change settings of the application. | */admin/settings* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Spring Back-end) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Admin users** | GET*, PUT, DELETE Registered users for Administrators, POST user profile for Registered user* | */api/v1/users* |
| * 1. **Profile** | GET a Registered User’s profile, PUT, DELETE *Registered user’s own profile data.* | */api/v1/users/own/{userId}* |
| * 1. **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/v1/login* |
| * 1. **Logout** | POST a logout request for ending the active session and invalidating the issued *Security Token*. | */api/v1/logout* |
| * 1. **Admin groups** | GET*, DELETE groups for Administrators, GET groups for Registered User* | */api/v1/groups* |
| * 1. **Group** | GET, PUT, DELETE *Registered user’s owned group, POST group for Registered User* | */api/v1/groups/own/{groupId}* |
| * 1. **Admin posts, Home, Group** | GET, DELETE posts for Administrators, GET, POST posts for Registered User | */api/v1/posts* |
| * 1. **Admin posts, Home, Group** | PUT, DELETE Registered User’s own posts | */api/v1/posts/own/{postId}* |
| * 1. **Admin posts, Home, Group** | GET, DELETE comments for Administrators, GET, POST comments for Registered User | */api/v1/comments* |
| * 1. **Admin posts, Home, Group** | PUT, DELETE Registered User’s own comments | */api/v1/comments/own/{commentId}* |
| * 1. **Group** | GET a group’s feed. | */api/v1/groups/feed/{groupId}* |
| * 1. **Home** | GET a Registered User’s own feed*.* | */api/v1/users/feed/{userId}* |
| * 1. **All Registered User Views** | Live streaming for notifications | */api/v1/notifications* |
| * 1. **Admin application** | GET, POST, PUT application settings for admin | */api/v1/settings* |
| * 1. **Admin posts, Home, Group** | POST attachment file for a comment or post | */api/v1/file/upload* |
| * 1. **Admin posts, Home, Group** | GET attachment file for a comment or post | */api/v1/file/{fileId}* |