|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Reddit Clone Application & Service Development using Spring boot v5 and Angular 2022** | |
|  | | | |
| Project author | | | |
| № | Names of Participants | | FN |
| 1 | Genadi Nikolov | | 62519 |

|  |  |
| --- | --- |
| Project name | Discussion Clone Application |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Nowadays has changed the way we learn, discuss, communicate between each other. The ***Discussion Clone Application(DCA)*** provides ability for users to discuss topics, vote section where you can download and vote for a post, also comment section where you can see the number of comment for each post. In addition it allows the users to register or logged in their accounts in order to make posts or comments and administrators to control and manage their profiles. The system will be developed using ***Spring 5 Application Development Framework***. It will be provided with security system using **Spring Security with JWT(JSON Web Token)** *Authentication for encoding information that will be securely transmitted as a JSON object. Also system will work with* ***Spring Data JPA(Java Persistence API) based on MySQL*** *to examine and control data base application.* It will implement web-based front-end client using ***Angular*** JavaScript/TypeScript libraries for asynchronous page data updates. It will have different links which will be posted by the users and the routing between pages will be done server side using ***SpringMVC***. The backend will be implemented as a ***REST API*** using JSON data serialization. There will be also a real-time event streaming from the server to the web client using ***WebFlux*** and ***Server Sent Events (SSE)/WebSocket*** in order to allow the Instructor to monitor test completion progress of students in real time. The main user roles (actors in UML) are:  • *Anonymous User* – can only read posts and comments, make a new profile or logged in already created profile  • *Account users* – if users have signed in their accounts, they can post a topic, create a post, write a comment in other topics, vote for a comment or post, download a post and navigate your own profile;  • *Authentication manager*(extends *MailTrap email profile*) – can answer on account requests, see also how many requests are in the Mailtrap box, decide to accept/reject the request for a new profile;  • *Administrator* (extends *Registered User*) – can manage Account users data, also follow the process how people manage their own accounts. |
|  |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse different posts and scan the comment section** | The *User* can browse the information already written and posted from registered users. Also if they do not have an account in the DCA they can sign up for free in order to use more from the app. | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, username, and choosing password. By default, all new registered users have *Student* role. The student can provide a *Token Key* in order to identify the *Student Group* to be enrolled to.  *Administrator* can register new by entering *User Data* and choosing a Role (*Student, Instructor*, or *Administrator*). | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered Users can view and edit their personal data also they can scan what posts they have with information provided about comments and voting for each post.*  *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *Registered User and Anonymous User* | *Account Users, Administrator* |
| * 1. **Manage Users** | *Administrator* can browse and filter users based on different criteria: username, email, Role.  *Administrator* can choose a *User* to manage, and can manage the chosen User - edit (using Change User Data UC) or delete.  Administrator can view a User information: what they posted as a topic, what they write as a comment and what they provide as a post below the topic. | *Administrator* |
| * 1. **Manage registration requests** | Authentication manager will receive authentication requests from Anonymous Users profiles which should be accepted or rejected by the current User with Authentication manager permissions. | *Authentication manager* |
| * 1. **Manage topics** | Account users can share the tittle of the topic which they want to be discussed with other people from the organization. | Account users |
| * 1. **Create posts** | *Account users can create their own posts, but first they should select the topic they want* | Account users |
| * 1. **Write comment** | Account users can write comment in their own post or below another post. They will have permission for sharing what they think in the comment section. | Account users |
| * 1. **Log in/ Log out** | Account users can log in their account just with providing the correct username/email and password. Then they can continue upgrade the profile with more comments and posts. | Account users |
| * 1. **Verify Account** | *Administrator will receive emails from everyone who want to join in the reddit clone application. For verification we are going to user mailtrap.io where people will send their emails for verification. We will create token and link to the verification place. Then when they are successful logged in they will have a chance to enter in the system* | Administrators |
| * 1. **Security** | Check every time when you type new credentials, new information, new post. Check also whether you are successful logged in or not, check when you want to make a post whether you are account user or just scrolling down the page. | *Administrators* |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. API Resources (Backend) | | | |
| **View name** | **Brief Descriptions** | | **URI** |
| * 1. **verifyAccount** | GET *Users requests and check for authentication errors.* Available only for Authentication manager. | | */api/auth/accountVerification/*  *verifyAccount* |
| * 1. **signup** | POST User Credentials(username, email, password) , send valid email address to Authentication manager | | */api/auth/signup* |
| * 1. **login** | POST *User Credentials* (e-mail address and password) and receive a valid *Authentication token* to use in subsequent API requests. | | */api/login* |
| * 1. **logout** | POST a logout request for ending the current session with saving all posts, comments and votes made during the session | | */api/logout* |
| * 1. **createComment** | POST User with valid account post comment. Available only for users who are successfully passed the log in/sing up stages. | | */api/comments/createComment* |
| * 1. **getAllComments** | GET All comment for current post, searching by username. | | */api/comments/by-user/*  *getAllCommentsForUser* |
| * 1. **getAllPosts** | GET Users(according to whether they have created account) all posts. Available for Administrators and for Account Users | | */api/posts/getAllPosts* |
| * 1. **createPost** | POST USERS with valid account could create a post by tittle, subtitle and description. Available only for Account Users. | | */api/posts/createPosts* |
| * 1. **createSubPost** | POST USER with valid account should create sub post before create actual post. Available only for Users with valid accounts | | */api/subPosts/createSubPosts* |
| * 1. **getAllSubPosts** | GET Users all sub posts entered by username. Available only for Account Users and not Anonymous Users who are in the system. | | */api/subPosts/getAllSubPosts* |
| * 1. **voteController** | POST Users who are in the system can vote for present post, can like or dislike the post. Available only for Account Users with already input valid credentials to login stage. | | */api/votes/vote* |
| * 1. **refreshTokens** | POST -> update the tokens on the Users who are in the system. Helps for getting a long term access to an application on behalf of a particular user. Not bothering the user and balances security with usability and reinforces authentication and improves user experience. | | */api/auth/refresh/token* |
| * 1. **createPost** | POST -> mainly using for creating new post which are going to be published in the main page of Reddit Clone Application and other account users can see them. | | /api/posts/createPost |
| * 1. **getPostBySubreddit** | | GET -> request using for getting the post with selected subreddit post or comment with specified ID. | /api/posts/by-subreddit/{id} |
| * 1. **getPostByUsername** | | GET -> request that get selected post with sepcified Username | /api/posts/by-user/{name} |
| * 1. **getAllSubreddits** | | GET -> request that get all subreddit from the List. In order to get them you should specify what is the correct ID of the subreddits. IDs are in sequence. | api/subreddit/{id} |
| * 1. **createSubreddit** | | POST -> request with which you can create your subreddit when you are logged in the system. In order to create subreddit you should be in the subreddit data base | /api/subreddit |
| * 1. **getSubreddit** | | GET -> request with which you can get only one specified subpost with an ID | /api/subreddit/{id} |