|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2019** | |
|  | | | |
| Project author | | | |
| № | valentinvstoyanov | | Face-to-face/ online |
| 1 | Instructor | | face-to-face |

|  |  |
| --- | --- |
| Project name | Social platform for challenges |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Nowadays, there are a lot of people that want to challenge each other. ***Challenger*** provides people the opportunity to create or complete challenges in exchange for some prize. Each person will have to register in order to use the app and then can view some challenges that can try to complete. Users can follow each other and can track challenges.  The system will be developed using ***Spring 5 Application Development Framework.*** It will implement ***mobile application*** (using ***Flutter***) that will serve as a front-end. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse challenges** | The *User* can browse the already existing challenges and can order them by different properties like popularity and difficulty. | Anonymous User, Registered User |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, username, and choosing password. | *Anonymous User* |
| * 1. **User profile** | *Registered User* can view and edit his/her profile. | *Registered User* |
| * 1. **Complete challenge** | *Registered User* can complete challenge by providing some evidence(photo or video depending on what the challenge creator demands). | *Registered User* |
| * 1. **Track challenge** | *Registered User* can track challenges that he/she is interested in. | *Registered User* |
| * 1. **Follow user** | *Registered Users* can follow/unfollow each other | *Registered User* |
| * 1. **Add/Edit challenge** | *Registered Users can add new challenges.* | *Registered User* |
| **2.8. Login** | *Anonymous user can login in the system by providing username/e-mail address and password.* | *Anonymous User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| **3.1. Home** | Presents the introductory information for the purpose of the system as well as detailed instructions on how to start using it. Prominently offers ability to register. | / |
| **3.2. Feed** | Presents the most popular challenges and other people's activity. | */feed* |
| **3.3. Challenges** | Presents list of all the challenges allowing the user order them by multiple factors like popularity, etc. | */challenges* |
| **3.4. Users** | Presents list of users to follow. | */users* |
| **3.5. User Registration** | Presents a view allowing the *Anonymous Users* to register in Challenger. | */register* |
| **3.6. Login** | Presents a view allowing the users to login. | */login* |
| **3.7. User Profile** | Provides ability to view and edit personal profile. | */profile* |
| **3.8. Challenge Detail** | Presents a view that shows all the details of the challenge. | */challenge/id* |
| **3.9. Add/Edit challenge** | Presents a view that allows the user to create new challenge or edit existing ones. | */challenge* |
| **3.10. Complete Challenge** | Presents ability to complete given challenge by providing evidence and submitting it. | */challenge/id* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **User** | GET all users, and POST new user (Id is auto-filled and modified entity is returned as result from POST request). GET user by id. | */api/user* |
| * 1. **Login** | POST *User Credentials* (username/e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/login* |
| * 1. **Logout** | POST a logout request for ending the active session with Challenger*,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Challenges** | GET all challenges, and POST new challenge (Id is auto-filled and modified entity is returned as result from POST request). GET challenge by id, and POST evidence for completing it. | */api/challenge* |