|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2020** | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | samoleti | | online |
| 2 | tsu | | online |

|  |  |
| --- | --- |
| Project name | Online events |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Intro…. The ***Online Events Platform (OEP)*** enables artists to organize online events and to share their talents with a virtual audience. In addition an artist can request a verification badge which guarantees the authentication of the user. The platform will be developed using ***Spring 5 Application Development Framework***.It will implement web-based front-end client using ***React.js/ Vue.js/ Angular*** JavaScript/TypeScript libraries for asynchronous page data updates. Each page will have a distinct URL, and the routing between pages will be done server side using ***SpringMVC/ WebFlux***. The backend will be implemented as a ***REST/JSON 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*** so that the host of an event can stream events in real time. The main user roles (actors in UML) are:  • *Anonymous* *User* – can view the information pages, attend only free events  • *User* – can view the information pages, attend and organize events. Some users can request a verification badge. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information and attend free events** | The *User* can browse the information views (Home, Events, About) in *OEP* and attend free events. | *All users* |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and choosing username and password. Google login? | *Anonymous User* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal *User Data* as well as request a Verification Badge. | *User* |
| * 1. **Manage events** | *User* can select date(s) and name of event, starting hour and predicted duration. The event also has a genre and description. The *User* can upload pictures and video. He/she can also change event data and delete an event. | *User* |
| * 1. **Digital wallet** | Each *User* has a digital wallet where he/she can add money for paid concerts. Also a *User* who created an event will receive money in his/her digital wallet from each visitor of his/her event. | *User* |
| * 1. **Host event** | The host of the event can choose to start and end livestream. | *User* |
| * 1. **Attend free event** | *The User* can choose a free event to attend. Then the event will be added only to the dashboard of a *User*. | *Anonymous User, User* |
| * 1. **Attend paid event** | *The User* can choose a paid event to attend and proceed with the payment. Then the event will be added to his/her dashboard. | *User* |
| * 1. **Event review** | The *User* can write a review which consists of a brief description of his/her impression of the event. Then the review is added to list of reviews in the profile of the *User* who created the event. | *User* |
| * 1. **Rating** | The User can add a rating varying from 1 to 5 stars. The given rating is accumulated in the overall rating of the *User* who created the event. | *User* |
| * 1. **Verification badge** | The *User* can request a verification badge. If the account represents a person, the *User* should submit a copy of his/her official government-issued photo identification (example: passport, driver's license, national identification card) to validate his/her request. If the account represents an organization, the *User* should give a copy of a phone or utility bill, a certificate of formation, articles of incorporation, or tax exemption documents. The *User* should include a few sentences explaining why the account should receive the verified badge, and relevant URLs that help illustrate the account's notability. For example, you can link to it from an official website, Facebook or Instagram profile. When the request is reviewed the *User* will receive a notification. | *User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information for the purpose of the platform. Search bar, filters such as dates, genres.. below top events to be listed, button show all events without filters  Prominently offers ability to register. | / |
| * 1. **Events** | Shows all events. Offers abilities to browse and choose events to attend. Also filters will be available. | */events* |
| * 1. **Event** | Presents all information about a future event. | */events/{event-id}* |
| * 1. **Attend event ?** | The *User* attends the event as a viewer. He/she can also provide feedback of his/her experience – rate and write review. | */{host-username}/{title}/{event-id}* |
| * 1. **Host event ?** | The *User* | */your-events/{event-id}* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register in *OEP* as well as to opt for a verification badge. | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | Provides ability to view and edit personal *User Data* as well as to request a verification badge. | */personal* |
| * 1. **Browse another User’s events** | The *User* can see all future events created by another *User*. | */{host-username}/all-events* |
| * 1. **User’s events** | Presents all events created by the *User*. | */your-events* |
| * 1. **Dashboard** | Presents future events which the User has previously chosen to attend. | */dashboard* |
| * 1. **About** | Presents information about the *OEP* project and his owner. | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* for *User* | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *username*. | */api/users/{username}* |
| * 1. **Login** | POST *User Credentials* (username 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 *OEP,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Events** | GET all events and POST new *Event* | */api/events* |
| * 1. **Event** | GET, PUT, DELETE *Event Data* | */api/event/{event-id}* |
| * 1. **Streaming Event** | ??? | */api/{username}/{event-id} ???* |
| * 1. **Reviews** | GET all reviews and POST new review | */api/{username}/reviews* |