|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2022** | |
|  | | | |
| Project author | | | |
| № | Names of Participants | | FN |
| 1 | Trayan Iliev | |  |

|  |  |
| --- | --- |
| Project name | Online Knowledge Testing System |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The Internet age changed the way we learn and test our knowledge. The ***Online Knowledge Testing System (OKTS)*** provides ability for instructors to define tests, and for students to test their knowledge and abilities. In addition to that it allows users to register, and administrators to manage them. The system will be developed using ***Spring Application Development Framework***.It will implement web-based front-end client using ***Thymeleaf/JSP*** templates and ***jQuery*** or ***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*** 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 view the information pages and try few sample tests without saving test results.  • *Student* (extends *Registered User*) – can choose a *Test* to complete – *Test Results* are saved automatically on test completion.  • *Instructor* (extends *Registered User*) – can create *Tests,* see the *Test Results,* and monitor *Students’* progress on particular test in real time.  • *Administrator* (extends *Registered User*) – can manage (create, edit user data and delete) all *Registered Users*, as well as *Student Groups*, *Tests* and *Test Results*. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information and try sample tests** | The *User* can browse the information views (Home, Tests, About) in *OKTS*, and can choose to try sample tests using the *Complete Test UC* . | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, 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 User* can view and edit her personal *User Data* as well as enter more *Token Keys* to participate in more *Student Groups*.  *Administrator* can view and edit *User Data* of all *Users* and assign them *Roles*: *Student, Instructor*, or *Administrator*. | *Registered User, Administrator* |
| * 1. **Manage Users** | *Administrator* can browse and filter users based on different criteria: first and last name, 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 create a new user using *Register UC*. | *Administrator* |
| * 1. **Manage Student Groups** | *Administrator* or *Instructor* can browse and filter *Student Groups* based on different criteria: name of *Student Group,* creation date, etc.  *Administrator* or *Instructor* can create a new *Student Group,* and assign a *Token Key* for automatic *Student* enrollment*.*  *Administrator* can choose a *Student Group* to manage, and can add or remove *Students* to the *Student Group,* or delete it if necessary. | *Administrator, Instructor* |
| * 1. **Manage Tests** | *Instructor* can browse her/his *Tests*, add new *Test* using *Add/Edit Test UC*, and delete a *Test*, as well as view the *Student's Test Results* for her/his own *Tests* and for particular *Student Groups* assigned to the *Test*.  *Administrator* can browse test of all *Instructors*, edit and delete them. | *Instructor, Administrator* |
| * 1. **Schedule a Test** | *Instructor* can browse her/his *Tests*, add new *Test* using *Add/Edit Test UC*, and delete a *Test*, as well as view the *Student's Test Results* for her own *Tests*.  *Administrator* can browse test of all *Instructors*, edit and delete them. | *Instructor, Administrator* |
| * 1. **Add/Edit Test** | *Instructor* or *Administrator* specifies/edits *Test* meta-data such as: test name, subject, description, tags, and chooses/creates a *Student Groups* for the *Test* using *Manage Student Groups UC*. *Instructor* or *Administrator* creates/edits *Questions* using *Create/Edit Question UC,* and can delete chosen *Question. Instructor* or *Administrator* saves the edited *Test.* | *Instructor, Administrator* |
| * 1. **Add/Edit Question** | *Instructor* or *Administrator* specifies/edits *Test's Question Type, Question Content*, and *Answers* providing weights for each answer (default weight is 0). | *Instructor, Administrator* |
| * 1. **Complete Test** | *Student* or *Instructor* browses tests available and can sort and filter them using different test metadata fields. *Student* or *Instructor* chooses a test to complete. *OKTS* presents the *Test Questions* one by one to the *Student* or *Instructor*, who answers them. The *Answers* of each *Question* are saved in real time on the server*.* On *Test* completion the *Test Results* are saved by *OKTS.* | *Student, Instructor* |
| * 1. **Browse Test Results** | *Student* can browse her own completed *Test Results*.  *Instructor* can browse the *Test Results* for *Tests* created by her/him and for particular *Student Groups* assigned to the *Test*. | *Student, Instructor* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request). Available only for *Administrators*. | */api/users* |
| * 1. **User** | GET, PUT, DELETE *User Data* for *User* with specified *userId*, according to restrictions described in UCs. | */api/users/{userId}* |
| * 1. **Login** | POST *User Credentials* (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 *OKTS,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Student Groups** | GET *Student Groups*, and POST new *Student Group* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request), according to *User's Role* and identity security restrictions. | */api/groups* |
| * 1. **Student Group** | GET, PUT, DELETE *Student Group* (including assigned students) for *Student Group* with specified *groupId*. | */api/groups/{groupId}* |
| * 1. **Tests** | GET users (according to *User's Role* and identity) and POST new *Test* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request). | */api/tests* |
| * 1. **Test** | GET, PUT, DELETE *Test Data* (including *Questions* and *Answers*) for *Test* with specified *testId*. | */api/tests/{testId}* |
| * 1. **Test Results** | GET *Test Results* (according to *User's Role* and identity) ) for *Test* with specified *testId*, and POST new *Test Result* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request). | */api/tests/{testId}/results* |
| * 1. **Test Result** | GET, PUT, DELETE *Test Result* (according to *User's Role* and identity) for *Test* with specified *testId* and *Test Result* with specified *testResultId.* | */api/tests/{testId}/results/{testResultId}/* |