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

|  |  |
| --- | --- |
| Project name | High Tech Online Store |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Today we use modern technologies more than ever. The ***High Tech Online Store (HTOS)*** provides ability for users to search and buy the most high technologies in the market worldwide. The system will be developed using ***Spring 5 Application Development Framework***.It will implement web-based front-end client using ***React.js/Vue.js/Angular*** 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 ***Server Sent Events (SSE)/WebSocket*** in order to allow the Administrator to send notifications for recently added high tech products and available discounts in real time. The main user roles (actors in UML) are:  • *Anonymous User* – can only view the products and their information but cannot make purchases  • *Registered User* – can view all products and information about them, as well as order products by adding them to their shopping cart  • *Administrator* – can manage (create, edit user data and delete) all *Registered Users, send notifications for discounts* |
|  |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | The *User* can browse the information views (Home, Product categories, Notifications, About) | 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. | *Anonymous User* |
| * 1. **Login** | *Anonymous User can login in the system by providing the same e-mail address and password, which was provided during registration.* | *Registered User, Administrator* |
| * 1. **Change User Data** | *Registered Users can view and edit their personal User Data.* | Registered User |
| * 1. **Manage Users** | Administrator can browse and filter users based on different criteria: first and last name, email. | *Administrator* |
| * 1. **Add to shopping cart** | Registered Users can add products to their shopping cart and then purchase them. | *Registered User* |
| * 1. **Search for a product** | *All users can search for available products by entering their request in the search field* | *All users* |
| * 1. **Search by category** | All users can search for available products on their category | *All users* |
| * 1. **View photos and information of a product** | *All users can view the photos and the information of a product* | *All users* |
| * 1. **Send notifications about discounts, etc.** | *Administrator can send notifications for discounts to all registered users* | *Administrator* |
| * 1. **Checkout** | *Registered users can choose the checkout details of his purchase (way of payment, way of shipment, etc)* | *Registered User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents header with logo, cart icon, login/register section, search section notification banner, new and upcoming releases section, products on sale section and footer with helpful links and other information | / |
| * 1. **Register** | Presents a view allowing the Anonymous User to register | */register* |
| * 1. **Login** | Presents a view allowing the Users to login | */login* |
| * 1. **Products** | Presents the products that are stored in the datebase and the ability to add, change, delete products as defined by the Ucs | */products* |
| * 1. **Orders** | Presents the orders the customer has placed and their status | */orders* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User Data** | Provides ability to view and edit personal *User Data* | */personal* |
| * 1. **Checkout** | Presents the ability to choose checkout details for your order | */checkout* |
| * 1. **About** | Presents information about the website | */about* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for all users, and POST new *User Data* (Id is auto-filled 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/Register** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. | */api/auth/login*  */api/auth/register* |
| * 1. **Logout** | POST a logout request for ending the active session and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Products** | GET products and POST new products (id is auto-filled and modified entity is returned as a result from POST request. | */api/products* |
| * 1. **Product** | GET, PUT, DELETE Product Data for product with specified productid | */api/products/{productId}* |
| * 1. **Comments on products** | GET comment for a product with specified productId and POST new comments (id is auto-filled and modified entity is returned as a result from POST request | */api/products/{productId}/comments* |
| * 1. **Comment** | GET, PUT, DELETE Comment Data for product with specified productID | */api/prducts/{productstId}/comments/{commentId}* |
| * 1. **Orders** | GET orders and POST new orders (id is auto-filled and modified entity is returned as a result from POST request) | /api/orders |
| * 1. **Order** | GET, PUT, DELETE Order Data for a Customer with specified orderID | */api/orders/{orderID}* |
| * 1. **Order Statuses** | GET Orders with specified status | */api/orders/status/{status}* |
| * 1. **About** | GET information on contacts, return policy, conditions, etc | */api/about* |