|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2019/20** | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Hodini, vkatrankie | | face-to-face |

|  |  |
| --- | --- |
| Project name | Vehicles trading system |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Today the need from a car is something very common. We will implement system for vehicle trading in order to provide everyone with easier and faster way to find what they are looking for.  For the implementation **Spring 5 Application Dev Framework** will be used. Some front end framework like **ReactJS** may help us for the front-end implementation along with the asynchronous page update. The idea includes also an implementation of a real-time event streaming from the server to the web client using ***WebFlux*** and ***Server Sent Events (SSE)/WebSocket*** in order to allow all the people involved into the trading for concrete vehicle (all dealers and bidders) to see same updated data about how the trading is going, without presence of any inconsistent data. The main user roles are going to be:   * Anonymous - He can only scroll through offers, can`t bid, can`t comment, can`t write reviews * Dealer - He can submit a vehicle offer, set starting price, provide any additional information about the offer * Bidder - He can bid for a specific vehicle, write reviews, choose favorite offers * Administrator - Approve user account register   NOTE: The above mentioned roles can be expanded during the project implementation. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and choosing password, role (dealer or bidder). Then he have to be approved by the Administrator`s account. | *Anonymous User, Administrator* |
| **Login** | *User login into his account, by entering same email and password, which provided during registration.* | *Anonymous User* |
| **Change User Data** | *Registered User* can view and edit her personal *User Data* as well as request a role change, which has to be approved by the Administrator`s account.  *Administrator* can view and/or edit *User Data* of all *Users* and assign them *Roles*: *Dealer, Bidder* or *Administrator*. | *Registered User, Administrator* |
| **Manage Users** | *Administrator can manage every other account. Including:*   1. *Administrator* can manage every user`s account - edit any User Data or delete some user (cause of some objective reason). 2. Administrator can create new user 3. Administrator can ban/block temporarily any user account and unblock it whenever he needs to. | *Administrator* |
| **Manage balance** | *Bidders have to add money to their accounts in order to bid in further offers.*  *Bidders and Dealers also can withdraw any free money from their accounts.*  *(For now it is going to be fake money.)* | *Bidder, Dealer (only withdraw)* |
| **Manage Vehicle groups** | *Administrator can move a vehicle in other category(car, truck, motorcycle etc.) if it has been entered wrong.* | *Administrator* |
| **Add vehicle for sale** | Dealers and Administrator can add new vehicle/s for sale to the respective category by providing:   1. Car brand 2. Car brand -> model 3. Starting price 4. Usage status (new, used) 5. If used, then the mileage of the car in km 6. Fuel type 7. Transmission type 8. Choose some well known features which are included and which are not 9. Any further comments   After submitting the vehicle, bidding is started after the first bid and a three-days bidding period is started. | *Administrator, Dealer* |
| **Submit a bid** | *Bidder can submit an offer by going to the concrete vehicle, select bid and then enter price greater or equal to the starting one.*  **NOTE: Once submitted, bid can not be withdrawn!** | *Bidder* |
| **Remove an vehicle** | *Administrator can remove any vehicle offer.* | *Administrator* |
| **Remove a bid** | *Administrator can withdraw any bid.* | *Administrator* |
| **Search and filter** | *Every user can search through the vehicle offers,*  *filter them by (starting price, car brand, vehicle type, fuel type, transmission and usage status).* | *Any type of User* |
| **Monitor Bidding Progress** | *Dealer can monitor the progress of every of his offers.*  *Administrator can monitor them too.* | *Dealer, Administrator* |
| **Monitor Bid history** | *Bidder can view his bid history.* | *Bidder* |
| **Send message to user** | Current user can send a message to another user | Any type of user |
| **Add a forum post** | User can add a post in the forum | Any type of user |
| **Edit post in forum** | Edit post in forum | Any type of user |
| **Delete post** | Delete post from the forum | Any type of user |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| **Home** | Introduction to the system, possibility for login and registration. | / |
| **Search vehicle** | Presents vehicles filtered by the search filters selected by the user. | */search* |
| **Register** | View which presents a registration form. | */register* |
| **Login** | View which presents a login form. | */login* |
| **Balance manager** | Presents ability of the bidders to add and withdraw money from their accounts. |  |
| **User Data** | Presents the User`s ability to view and edit his personal data, password, email etc.. | */personal-data* |
| **Pending register requests** | Presents the ability of the Administrator to view and approve or disapprove any of the register requests, before account`s creation is finished. | */administrator/register-requests* |
| **Offer editor** | Presents ability to dealer to view and edit his offer if it is **NOT** in bidding process yet. | */dealer/offer* |
| **Offer viewer** | Presents the ability of the bidder to view concrete offer and make bids to it. | */bidder/offer* |
| **Users** | Presents the ability of Administrator to view and edit Users Data. | */users* |
| **Bids history** | Presents the ability of the bidders to monitor their bids history. | */bidder/history* |
| **Bids progress** | Presents all the dealer`s offers with their statuses and current winning bid as well as the time it has been submitted and time left to the bidding end. | */dealer/history* |
| **Add post to forum** | Add post to the forum | *forum/addPost* |
| **Edit/detele post** | Edit and delete posts view |  |
| **Forum posts** | Presents all the posts from the blog | *forum/posts* |
| **Chat by sender and addressee ids** | Loads messages from a conversation between two users by their ids | *messages?{userId}={value}&{adressee}={value}* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Java Spring Framework) | | |
| **View name** | **Brief Descriptions** | **URI** |
| **Users** | GET All Users data. | */api/users* |
| **Dealers** | GET All Dealers data. | */api/dealers* |
| **Bidders** | GET All Bidders data. | */api/bidders* |
| **User** | GET, PUT, DELETE *User Data* for *User* by specified *userId* | */api/user/{userId}* |
| **Login** | POST *User Credentials* (e-mail address and password) and receive a valid *Security Token* to use in subsequent API requests. UserID is auto generated by the system. | */api/login* |
| **Logout** | POST a logout request for ending the active session with the system*,* and invalidating the issued *Security Token*. | */api/logout* |
| **Offers by filter** | GET offers by filters selected | */api/search/{filterParameters}* |
| **Offer by id** | GET offers by offerId | */api/offer{offerId}* |
| **Dealer offers** | GET all offers of a dealer specified by userId | */api/dealerOffers{userId}* |
| **Bidder bids** | GET all bids of a bidder specified by userId | */api/biddderBids{userId}* |
| **Submit offer** | POST, PUT, DELETE to create, update and delete an offer by the Administrator or a dealer.  NOTE: In case of POST submission of a new offer offerId is auto generated by the system. | */api/offerAction* |
| **Submit bid** | POST new bid by the current user. | */api/bid* |
| **Offer bids** | GET all bids made for an offer, specified by offerId | */api/offerBids/{offerId}* |
| **Balance** | GET, PUT to add/withdraw money from the account. | */api/balance{amount}* |
| **Rate a dealer** | GET, POST, PUT for the current user to rate a dealer | ratings?userId=idValue&rate=rValue |
| **Messages between users** | GET, POST messages between users | messages?user1Id={value1}&userId2={value2} |
| **Forum get all posts** | GET, POST, PUT, DELETE to create, update or delete or get posts | forum/posts |