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

|  |  |
| --- | --- |
| Project name | Auto Shop |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| In the modern world, most people use digitalized technologies in all industries. Our project represents a modern way of selling cars, namely the use of the Internet space for placing ads. In Bulgaria, many adhere to an outdated approach, when to select a car you need to visit specialized areas where cars are located. Our system is designed to improve the way we sell cars. The system will be developed using ***Spring 5 Application Development Framework***.It will implement web-based front-end client using ***Thymeleaf/JSP*** templates and ***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***. The backend will be implemented as a ***REST*** using JSON data serialization. The main user accounts (actors in UML) are:  • *Anonymous User* – can only view ads, posts, comments, reviews and accounts of other users.  • *User, registered as a regular seller* – can publish ads, write comments to the ads of other users, write his own posts on various topics, make reviews.  • *User, registered as a* car site – has identical functionality with a regular user, but also has a legal entity registration, work schedule, location.  • *User, registered as a service provider* – can post offers about their services, such as a car service or car wash.  • *Administrator* – can manage (create, edit data and delete) all another entities. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse ads, posts, comments, reviews and accounts** | The *User* can browse the information views. | 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. When registering, the user must select the type of account that suits him, after which he will be required to enter information relating specifically to this type of account.  *Administrator* can register new user by entering specific data and choosing a type of account. | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered User* can view and edit her personal data.  *Administrator* can view and edit data of all users and assign them type of account: *User*, *Car Site* or *Administrator*. | *Registered User, Administrator* |
| * 1. **Manage Users** | *Administrator* can browse and filter users based on different criteria: first and last name, email, type of account.  *Administrator* can choose a *User* to manage, and can manage the chosen User - edit or delete.  *Administrator* can create a new user using *Register*. | *Administrator* |
| * 1. **Manage sale announcement** | *User* can browse her/his *sale announcement*, add new *sale announcement* using *Add/Edit sale announcement*, and delete a *announcement*.  *Administrator* can browse *announcements* of all *Users*, edit and delete them. | *Registered User, Administrator* |
| * 1. **Add/Edit announcement** | *User* or *Administrator* specifies/edits announcement data such as: name, description. | *Registered User, Administrator* |
| * 1. **Brows reviews** | *User* can browse reviews and can Create new one or edit old his/her reviews. | *Registered User, Administrator* |
| * 1. **Add/edit reviews** | *User* can create new reviews or edit old his/her reviews.  Administrator can create/edit/delete any review. | *Registered User, Administrator* |
| * 1. **Brows comments** | *User* can see comments under a announcement. | *Registered User, Administrator* |
| * 1. **Add/edit reviews** | *User* can create new comments or edit old his/her comments.  Administrator can create/edit/delete any comments. | *Registered User, Administrator* |
| * 1. **Brows posts** | *User* can see posts in blog section. | *Registered User, Administrator* |
| * 1. **Add/edit posts** | *User* can create new posts or edit old his/her posts.  Administrator can create/edit/delete any posts. | *Registered User, Administrator* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents a general page that hosts several ads, multiple table of contents for posts. | / |
| * 1. **Cabinet** | Represents a view with data about a user, his ads and posts. | */cabinet* |
| * 1. **Posts** | Represents a view with posts by the current user. | */posts* |
| * 1. **Create post** | Represents a view with a template for creating a post. | */create-post* |
| * 1. **Post** | Represents a view with a specific post. | */post* |
| * 1. **Announcement** | Represent a view with a specific announcement | */announcement* |
| * 1. **Create announcements** | Represents a view with a template for creating an announcement. | */create-announcements* |
| * 1. **User Registration** | Presents a view allowing the *Anonymous Users* to register in the system. | */register* |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **Edit Data** | Provides ability to view and edit any type of data, like user information, ad information and others. | */edit* |
| * 1. **Users** | Presents ability to manage (CRUD) Users and their User Data (available for *Administrators* only, as described in UCs). | */user* |
| * 1. **Comment** | Presents a view with a comments, that can be written by users. | */comment* |
| * 1. **Reviews** | Present a view with reviews from other users. | */reviews* |

|  |  |  |
| --- | --- | --- |
| 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** | 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*,* and invalidating the issued *Security Token*. | */api/logout* |
| * 1. **Reviews** | GET *Reviews*, and POST new *Review* (Id is auto-filled and modified entity is returned as result from POST request), according to account type and identity security restrictions. | */api/reviews* |
| * 1. **Comments** | GET *Comments*, and POST new *Comment* (Id is auto-filled and modified entity is returned as result from POST request), according to account type and identity security restrictions. | */api/comments* |
| * 1. **Comment** | GET, PUT, DELETE *Comment* with specified *Id*. | */api/comments/{id}* |
| * 1. **Posts** | GET *Posts*, and POST new *Post* (Id is auto-filled and modified entity is returned as result from POST request), according to account type and identity security restrictions. | */api/posts* |
| * 1. **Post** | GET, PUT, DELETE *Post* with specified *Id*. | */api/posts/{id}* |
| * 1. **Announcements** | GET *Announcements*, and POST new *Announcement* (Id is auto-filled and modified entity is returned as result from POST request), according to account type and identity security restrictions. | */api/announcements* |
| * 1. **Post** | GET, PUT, DELETE *Announcement* with specified *Id*. | */api/announcements/{id}* |