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

|  |  |
| --- | --- |
| Project name | Carpooly |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| Идеята на проекта е да представи удобно и лесно приложение за хора, които си търсят пътници до избрана от тях дестниция, или обратното – хора да търсят превоз от зададена начална и крайна точка. Всеки потребител ще може да филтрира всички пътешествия в момента, както и да оценява потребители. Сървърната част на приложение използва SpringBoot с релационната база данни HSQLDB, докато клиентската част е уеб сайт написан на AngularJS. Приложението има защита, като за всеки потребител се генерира JWS Token, a паролата му се пази в хеширан вид. Приложението ще притежава следните потребители:   * Нерегистран – ще има възможност да си направи регистрация * Пътник – ще има възможност да си търси пътуване, но не и да прави * Шофьор – надграден пътник, но ще има право да прави пътуване. Това е с цел шофьорите да бъдат само оторизирани лица от системата * Модератор – човек, който се грижи за реда на потребителите – най-вече да удобрява пътници, които искат да станат шофьори * Админ – човек, който има всички права на системата |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Регистрация** | Нерегистрирания (Анонимния) потребител може да си създаде акаунт предоставайки следните данни: имейл, парола, име и фамилия. Той ще има роля - пътник | Всички потребители |
| * 1. **Обновяване на данните** | *Всеки един потребител ще може да си променя първото и последното име, както и паролата, но не и имейла.*  *Администратора може да променя на всеки потребител данните.* | *Всички регистрирани потребители* |
| * 1. **Взимане на текущите данни на потребителя** | *Всеки един потребител ще може да получи своите текущи данни.*  *Администратора и Модератора могат да виждат данните на всеки един потребител.* | *Всички регистрирани потребители* |
| * 1. **Триене на потребител** | *Всеки потребител може да изтрие своя акаунт в системата.*  *Администратора може да трие всеки един потребител.* | *Всички регистрирани потребители* |
| * 1. **Създаване на пътуване** | *Шофьорите могат да създават пътуване по дадените данни: начална и крайна точка, максимален брой пътници, цена на превоз* | *Шофьор, Модератор, Админ* |
| * 1. **Стартиране и спиране на пътуване** | *Шофьора на дадено пътуване може да маркира даденото пътуване като стартирано и спряно. Докато е в стартирано положение, пътници не могат да се присъединяват.* | *Шофьор, Модератор, Админ* |
| * 1. **Обновяване на пътуване** | *Шофьорите могат да обновяват своето текущо пътуване* | *Шофьор, Модератор, Админ* |
| * 1. **Присъединяване и излизане от пътуване** | *Пътниците могат да се присъединяват в нестартирани пътувания, както и да излезнат от него, ако то не е стартирано вече.* | *Всички регистрирани потребители* |
| * 1. **Триене на пътуване** | *Шофьорите могат да изтрият своето пътуване.* | *Шофьор, Модератор, Админ* |
| * 1. **Създаване на искан маршурут** | *Пътниците могат да направят искане на пътуване* | *Всички регистрирани потребители* |
| * 1. **Обновяване на искан маршрут** | *Пътниците могат да обновяват своето искане на пътуване* | *Всички регистрирани потребители* |
| * 1. **Триене на маршрут** | *Пътниците могат да трият своето искане на маршрут* | *Всички регистрирани потребители* |
| * 1. **Създаване и предлагане на пътуване по даден маршрут** | *Шофьорите могат да създават и предлагат пътуване по даден маршрут* | *Шофьор, Модератор, Админ* |
| * 1. **Плащане на пътуване** | *Пътниците могат да плащат своето пътуване по различен път – в брой, през онлайн банкиране или чрез ваучер* | *Всички регистрирани потребители* |
| * 1. **Купуване на ваучер** | *Потребителите могат да купуват ваучер, който да бъде използван за следващи пътувания. Това може да* | *Всички регистрирани потребители* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Представя основната информация на приложението | / |
| * 1. **Login** | Страница за логване | */login* |
| * 1. **Create Ride** | Създаване на пътуване | */create-ride* |
| * 1. **Join Ride** | Влизане във вече същестуващо пътуване | */join-ride* |
| * 1. **Search Ride** | Търсене на пътуване | */search-ride* |
| * 1. **User profile** | Страница за профила на даден потребител | */profile/{id}* |
| * 1. **Rank user** | Страница за оценяване на потребител | */profile/{id}/rank* |
| * 1. **Register** | Страница за регистриране | */register* |

|  |  |  |
| --- | --- | --- |
| 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. **Rides** | GET *Rides* for all users, and POST new *User Data* (Id is auto-filled by *OKTS* and modified entity is returned as result from POST request). | */api/rides* |
| * 1. **Ride** | GET, PUT, DELETE for *Ride* with specified *id*. | */api/rides/{id}* |
| * 1. **Start Ride** | POST to Start a Ride | */api/rides/start-ride/{id}* |
| * 1. **Stop Ride** | POST to Stop a Ride | */api/rides/stop-ride/{id}* |
| * 1. **JoinRide** | The current authenticated user to join a ride - POST | */api/rides/join-ride/{id}* |
| * 1. **Leave Ride** | The current authenticated user to leave a ride if he is in - POST | */api/rides/leave-ride/{id}* |
| * 1. **FindRide** | GET all Rides with matching coordinations as starting and final point. The coordinates are request parameters | */api/active-tests* |