|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2020 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Mon | | face-to-face |

|  |  |
| --- | --- |
| Project name | CarPooling |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| A lot of people travel between cities every day. With the CarPooling app you can find someone that is travelling in the same time to the same place that you need to go. You just have to register. You can be a passenger or/and a driver, create a trip or join a trip. If you do something wrong, administrators will take care of you.  The Application will be developed as a *Single Page Application (SPA)* using ***React.js*** as front-end, and ***Node.js + express*** as backend technologies. Each view will have a distinct URL, and the routing between pages will be done client side using ***React Router***. The backend will be implemented as a ***REST/JSON API*** using JSON data serialization.  The main user roles (actors in UML) are:  • *Anonymous User* – can only view the available trips.  • *Registered User – create trip, join trip, manage his trips, add his car details*  • *Administrator* (extends *Registered User*) – can manage (delete) all *Registered Users and their trips*. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse trips** | The *User* can browse the available trips and see their details. | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, username, phone number and choosing password.  *Administrator*s are only the owners of the app. | *Anonymous User, Administrator* |
| * 1. **Change User Data** | *Registered User* can view and edit his personal *User Data*.  *Administrator* can edit his User Data, but can see all users data. | *Registered User, Administrator* |
| * 1. **Manage Users** | *Administrator* can browse all users.  *Administrator* can choose a *User* and delete him. | *Administrator* |
| * 1. **Manage Trips** | *Administrator* can browse the trips and delete them.  *Administrator can create his own trips and edit them.*  *Registered User* can browse the trips and edit only his trips. | *Administrator, Registered User* |
| * 1. **Manage Cars** | *Administrator* can edit and delete his cars and delete User’s cars.  Registered User can edit his cars and delete them. | *Administrator, Registered User* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Trips** | List all available Trips. | / |
| * 1. **Registration** | A view with fields for registering a user. | */sign-up* |
| * 1. **Login** | A view with fields allowing user to log in. | */login* |
| * 1. **Create Trip** | A view where you can fill the available fields (some fields are required) and create your own trip. | */trip/create* |
| * 1. **Create Cars** | A view where you can fill the available fields (some fields are required) and create your own car. | */cars/create* |
| * 1. **Edit User** | A view where you can see your user details and edit some of them or all of them. | */profile/:username/edit* |
| * 1. **User Profile** | A view of a user profile. Displaying user’s names, email, phone and avatar. | */profile/:username* |
| * 1. **Trip** | A view of a Trip details :driver, car, departure time, origin, destination, available places, status, allows smoking, pets and luggage. | */trips/:tripid* |
| * 1. **Edit Trip** | A view of a Trip details (only if you are the driver) where you can change some of them or all of them. It depends on the Trip status. | */trips/:tripid/edit* |
| * 1. **Manage Trip** | A view where you can change the status of your trip , approve passengers, leave and read comments. | */trips/:tripid/manage* |
| * 1. **Cars** | A view with list of your cars. | */user/cars* |
| * 1. **Edit car** | A view of car details: model, seats, has air conditioner, year made, images. | */cars/edit/:carid* |
| * 1. **View or Delete Trip** | A view with list of all trips, available only for Administrators, where they can view or delete the trip. | */admin/trips* |
| * 1. **View or Delete User** | A view with list of all users, available only for Administrators, where they can view or delete the trip. | */admin/users* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET all users, PUT – update user | */api/users* |
| * 1. **User** | GET user by username | */api/users/{username}* |
| * 1. **Login** | POST *User Credentials* and receive a valid *Security Token* to use in subsequent API requests. | */api/users/authenticate* |
| * 1. **Logout** | POST a logout request for ending the active session with *OKTS,* and invalidating the issued *Security Token*. | */api/users/logout* |
| * 1. **User** | GET *the current logged user*. | */api/users/me* |
| * 1. **Register** | PUT- create new user | */api/users/register* |
| * 1. **Avatar** | GET user profile picture  POST – upload/change a profile picture | */api/users/{userId}/avatar* |
| * 1. **Cars** | GET list of the current user’s cars | */api/users/{userId}/cars* |
| * 1. **Cars** | GET page of the current user’s cars | */api/users/cars* |
| * 1. **Trips** | GET all trips  POST- create a trip  PUT- edit trip | */api/trips* |
| * 1. **Trip** | GET trip by ID  PATCH- change trip status | */api/trips/{id}* |
| * 1. **Comment** | POST- add comment to a trip | */api/trips/{id}/comments* |
| * 1. **Passengers** | POST- apply for a trip | */api/trips/{id}/passengers* |
| * 1. **Passenger Status** | PATCH- change passenger status | */api/trips/{tripID}/passengers/{passengerID}* |
| * 1. **Passengers** | GET- get trip’s passengers | *api/trips/{tripID}/passengers* |
| * 1. **Comments** | GET trip’s comments | *api/trips/{tripID}comments* |
| * 1. **Locations** | GET available locations | *api/trips/locations* |
| * 1. **Passenger** | GET- shows are you a passenger in this trip | *api/trips/{tripID}/amIPassenger* |
| * 1. **Car** | GET-list of all cars  POST-create a car  PUT-edit a car | *api/cars* |
| * 1. **Car Images** | GET-get car images  POST-add car image | *api/cars/{carID}/images* |
| * 1. **Car Image** | GET-get one car image | *api/cars/{carID}/images/{imageID}* |
| * 1. **Car** | GET- get car by ID  DELETE- delete car by ID | *api/cars/{carID}* |
| * 1. **Users** | DELETE-delete user’s profile | *api/admin/users/{userID}* |
| * 1. **Trips** | DELETE-delete trip by ID | *api/admin/trips/{tripID}* |