|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | Fullstack Application Development with Node.js + Express.js + React.js - 2019 | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Stan | | face-to-face |

|  |  |
| --- | --- |
| Project name | Uber-like app |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The Internet age changed the way we live. Uber provides the ability for people to post rides request (from -> to) and the amount of money they are willing to pay, and then the registered drivers can decide whether or not to take the request as it is or make an offer for it. This is basically an innovative alternative to the taxis. The system will be developed as SPA (Single Page Application) using ***React.js*** as front-end, and ***Node.js + express*** as backend technologies.  The main user roles (actors in UML) are:  • *Anonymous User* – can only view ride requests and listed drivers.  • *Traveler User* – can post ride requests as well as leave rating/comments about the drivers.  • *Driver* (extends *Traveler)*– a Traveler that can accept and post ride requests. They will have two separate ratings (as Traveler and as a Driver).  • Moderators – (whitelisted user) – can delete requests/ban users. Their main role is to keep the platform clean. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse through ride requests and drivers.** | All users can browse through the current ride requests and registered drivers. | 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. By default all new registered users are Travelers. They have the option to become Drivers by giving further information about their car and driving license.  To become a moderator one must have their nickname whitelisted before registering in order to be moderator when registered. | *Anonymous User* |
| * 1. **Change User Data** | *Travelers and Drivers* can view and edit their personal *data*.  *Moderators* can view and verify *data* of all users requested a driver “promotion” and assign them as drivers. | *Traveler, Driver, Moderator* |
| * 1. **Manage Users** | *Moderators* can browse and filter users based on different criteria: nickname, rating (Drivers), reports.  *Moderators* can edit (rename) or delete users. | *Moderator* |
| * 1. **Report/rate comments/drivers.** | Travelers can leave rating and comments about drivers after every ride.  Drivers can leave rating and comments about travelers after every ride.  Travelers can report drivers for a particular reason. They can report comments that they think should be removed as well as report other users` nicknames.  Drivers can report travelers for a particular reason + report comments and nicknames. | *Traveler, Driver* |
| * 1. **Add ride requests** | Travelers can add new ride requests alongside with the price they are willing to pay.  Drivers can still use the platform as travelers. | *Traveler, Driver* |
| * 1. **Accept offers** | *Drivers can accept ride requests as they are or edit the price and make an offer to the travelers.*  *Travelers can accept or deny an offer, made by a driver, for their request. If the offer is denied the request remains.* | *Drivers, Traveler* |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (SPA Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | The home page that gives an overview of the application. Sections for current requests, drivers leaderboard and favourite travelers/drivers. | / |
| * 1. **Create requests** | The page where requests are created. Provides the ability to choose destination, price and give some information. | */createRequest* |
| * 1. **Drivers** | Complete list of verified Drivers, filtered by rating/rides. | */drivers* |
| * 1. **Account** | Account panel with settings and a preview option. | */account* |
| * 1. **About** | Presents information about the *platform* and its creator. | */about* |
| * 1. **Inspect someone`s account** | Presents a person`s account with their rating/badges/comments. | */previewAccount* |
| * 1. **Take requests** | Page available only for drivers. There the requests will be filtered for the drivers depending on the region they are in. | */takeRequest* |
| * 1. **Reports table** | A table of reported users or user`s comments available for moderators only. | */reports* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Node.js Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Accounts** | PUT, GET, POST account information. (username, email, password, account type, rating, comments, favourites, requests, reports). | */api/accounts* |
| * 1. **Requests** | GET, POST, DELETE requests for rides. Requests will have status, negotiable price, information for the traveler, amount of people, region, comments, timestamp. (Google maps api fetch for route ?). | */api/requests* |
| * 1. **Drivers** | PUT, GET, POST information about car, driver license, and real time location(region). (extends account) | */api/drivers* |
| * 1. **Reports** | GET information for reports made by users*.* Information  Available only for moderators. | */api/reports* |

-Some statistical analysis preferably with graphics.

-Add statistics for driver to check their performance.

-Add statistics for moderators to monitor the drivers and if needed remove drivers.