The Garage

This exercise consists of two tasks which belong together. So before starting read the whole document.

Task 1 – Domain logic

The city of Vence needs help with the implementation of a new parking garage and asks you for your support.

The garage should support different types of vehicles:

- Cars
- Motorbikes

Every vehicle has a unique identifier (the license plate), and can exist only once – thus being either within the garage or outside of it.

The planned garage should support multiple parking levels – the city of Vence is currently undecided how high they will be able to build for stability reasons. As a result, your implementation should allow for arbitrary numbers of parking levels – at least 1 level, but keep it flexible.

The same goes for the number of parking spaces per level – the area where the garage will be built is not yet decided upon. So again, keep this flexible and configurable.

Your task is to develop a simulation program for the garage. Vehicles should be able to enter and exit the garage – the garage should then assign a free space or reject the vehicle if there are no more free parking lots.

Task 2 - UI

To check the occupancy, the staff of the garage needs a nice looking UI. Luckily there is already a mockup for the UI available (see 2nd page). Please build a JavaScript single page application based on the mockup. As domain logic use the code from task one.

Functionality:

- List all vehicles in the garage. The list should support paging

- The list can be filtered by level and/or by type. Multiple filters can be selected, also from different categories. Filters within one category are OR-connected; filters in different categories are AND-connected.
- Option: allow the list to be filtered by typing the license number into the search field

A UI for creating new vehicles entering and leaving the garage is not mandatory but would be nice.

Some tips

Try to use a popular MVC Front-End framework but you are free to decide which one: AngularJS, Backbone, Ember or any other framework that we may not have heard. It's up to you. In general, you are welcome to incorporate the front-end technologies that you are comfortable with and think fit to this project. However, we'd prefer that you didn't use CoffeeScript. We'd also prefer that you don't use a CSS framework as we'd like to see how well you write mobile-first CSS. And finally, don't bother to write a back-end. it's enough to write a service in JavaScript that simulates the backend and serves the data by direct method calls.

Any questions? Just ask