Welldone - Front-End Practical Exam - Part 1

Estimated time: **4 hours**

Overview

The exam is designed to test your abilities to cope with new technologies, check your learning capabilities and see how you integrate with external frameworks.  There is no algorithmic complexity or computation challenge. It is completely trivial in this respect.

While the correctness of the implementation, the lack of bugs, overall look and feel of the external result and user experience is of the highest importance, the elegance of the code, usage of the correct language idioms and patterns and the coding conventions and practices applied will also play a major factor when evaluating the solution. It should be elegant both externally and internally.

What you need to build

The sample application is called myLocations. It allows the user to maintain a list of categorized name locations.

The domain model has two main entities: *Category* and *Location*.

*Category* has a single property: name.

*Location* has the following properties: name, address, coordinates, and category.  
All properties of the entities are required.

The first part in this exercise, which you will implement now, is a basic CRUD application to manage the category entity.

Use Cases

* The user can view a list of all existing categories.
* When the user chooses a category from the list, it is highlighted, and he can click any of the actions in the toolbar to manage the category - edit, view details, delete.
* There is a permanent toolbar at the top, for all screens. It contains a title and actions. The title and actions update according to context. For example, when the user is in the “category list” or “new category” screens there is no specific category in the context so it doesn’t make sense to make the “edit” / “delete” actions available. The title in this case can be “Categories” and the only available action is “new category”. If there is a category in context the available actions would be “edit”, “delete”, “view”.
* The user can create a new category (this action is also in the toolbar).

Tech Stack

* Use ES2015 standard
* Static typing (TS / flow) isn’t mandatory
* All data is saved to the local storage of the browser (an HTML5 feature) for simplicity.
* You don’t have to implement UI components from scratch. Just use your favorite UI toolkit.
* You can choose between React and Angular.
  + If you choose React, you may use either Redux, or MobX.
  + If you choose Angular, you may use either MGRX or Standard RxJS services

Remember the important things:

* Code quality
* Solution correctness and bug free
* UI & UX

Generally, we expect to see a production-ready system.

Give us your best!