# Front-End (React/Angular2) Practical Exam

## 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 and the overall look and feel of the external result is of the highest importance, the elegancy 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.

## Development Environment (choose between react or ng-2)

You can use the IDE of your liking to develop this part of the solution.

### React

It is expected that you base your design on either a Flux architecture, preferably, Redux, or on MobX.

Use babel and webpack to support es2015 style jsx syntax and module loading.

### Angular 2

You can use either webpack or gulp based build environments.

Use typescript, modern style (es2015 etc).

Choose an architecture based on RxJS and/or Redux and/or observables.

## Sample Application Description

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

The domain model contains two main entities: a Category and a Location.

A Category has a single property: Name.

A Location has the following properties: Name, Address, Coordinates, and Category.

All data is saved to the locale storage of the browser (an HTML5 feature) for simplicity.

The application should use the react-router module.

## Use Cases

The user can manage (view, add, remove and edit) the list of Categories.

The user can manage (view, add, remove and edit) the list of Locations.

The user must fill all properties when saving an item.

The user must choose a category from a list of existing categories when defining a Location.

Each screen has a top toolbar with title and action buttons. The user executes an operation on a list item by clicking the appropriate button in the top toolbar.

The application screen has a bottom bar with two iconic buttons: Categories and Locations. The user moves between Categories and Location management by clicking on their respective icons on the bottom button bar.

The user can view all Locations sorted by alphabetical order, grouped or ungrouped by category.

The user can view only the locations assigned a specific category he chooses.

When clicking a location on the list, the user can choose to see the properties of the item or view it on an actual map (using google maps or similar service).

When the user clicks on a location, the device will vibrate (via native bridge support).

Bonus: allow selecting the coordinates from the map service and not entering by hand.

Bonus: allow relating multiple categories to a single item, define and enhance the use cases and ui related to this.

Good Luck