# Qatium Tech - Take Home Exercise

## Child care

Vanessa Ruiz Lago

Software developer

# Content

## About

The project has been developed in Angular, the web development framework I'm currently working with.

The project has been organized in different modules; each module is a "block" of functionality so:

- Landing page: Control of the landing page, in this case we've only one but here would be the code to load different ones based on client, for example.
- Portal: Layout control, first module loaded after the landing page. Here we define the way the app is going to look like (header, menus, footer…) in this case and to simplify only header and content.
- Child-care: the content. Here you could find the different components used to show each one of the functionalities requested in the exercise:
  - List of childcares
  - Add a care
  - Add a parent
  - Show balance
  - Show "transactions", this is minimum number of childcare requests to net the time debts.
- Shared: this is a special module I like to create in my projects to keep components or services that are used all over the application in this case:
  - A custom dropdown: can be reused to load it with different data, customize the placeholder, select a default option…
  - Loader
  - Feedback-button: a regular button that the developer can configure with different texts and is disabled and change the icon while the communication with the API hasn't finished (to avoid double clicks and double calls).

All the data has been mocked in the child-care.service.ts.

The project has tests, implemented with jest. I haven't covered 100% of the code because most of them are almost the same, so I've tried to cover, at least, one test per functionality:

- API calls
- Component creation
- Component behaviour depending on API returned data
- Button clicks
- Algorithms

## How to execute the app

1. Clone the repo https://github.com/vruizl/Qatium-Child-Care.git
2. To generate the dependency files execute npm install(could be necessary to add - -force)
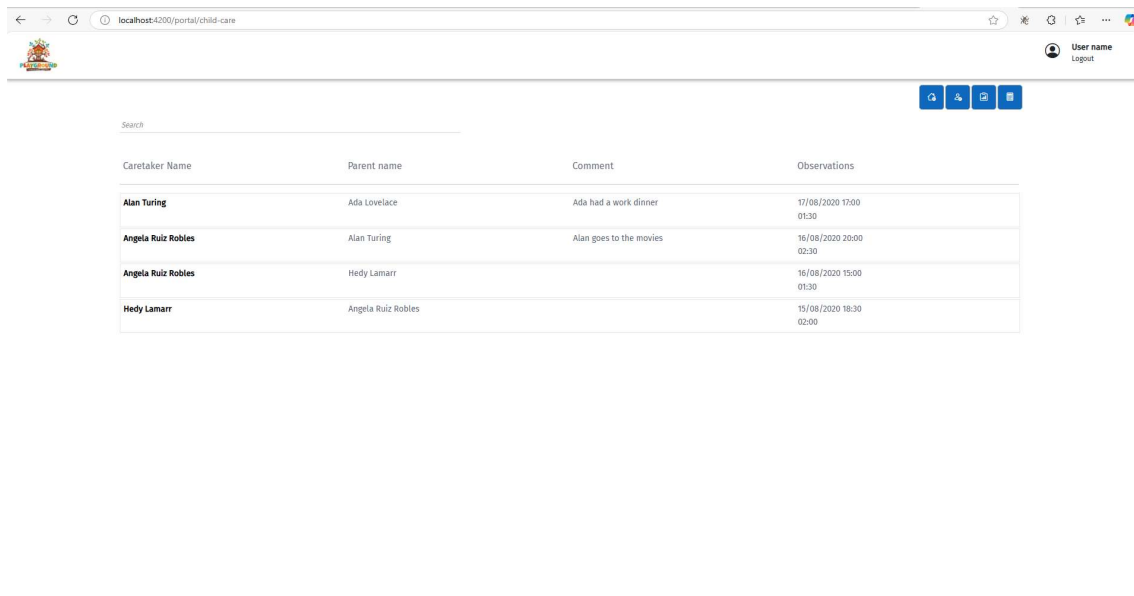3. Execute npm start

## The app

If you don't want to execute the app from the code or have some problems, some screenshots:
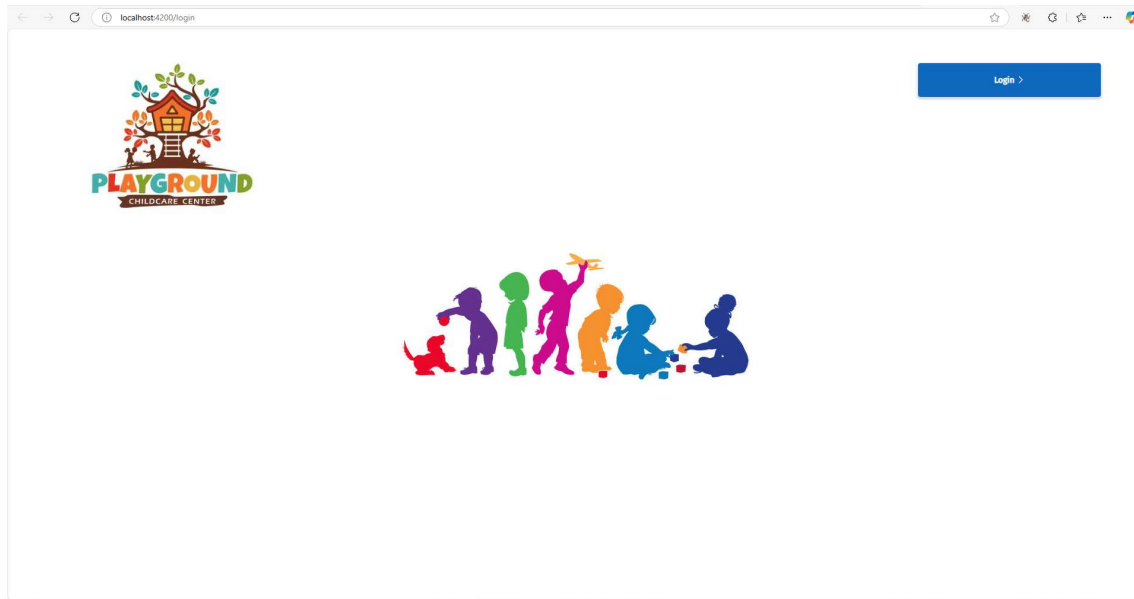


*Landing page*

If you click on the Login button, you will navigate to the portal and you will be able to see the list of already created child cares:
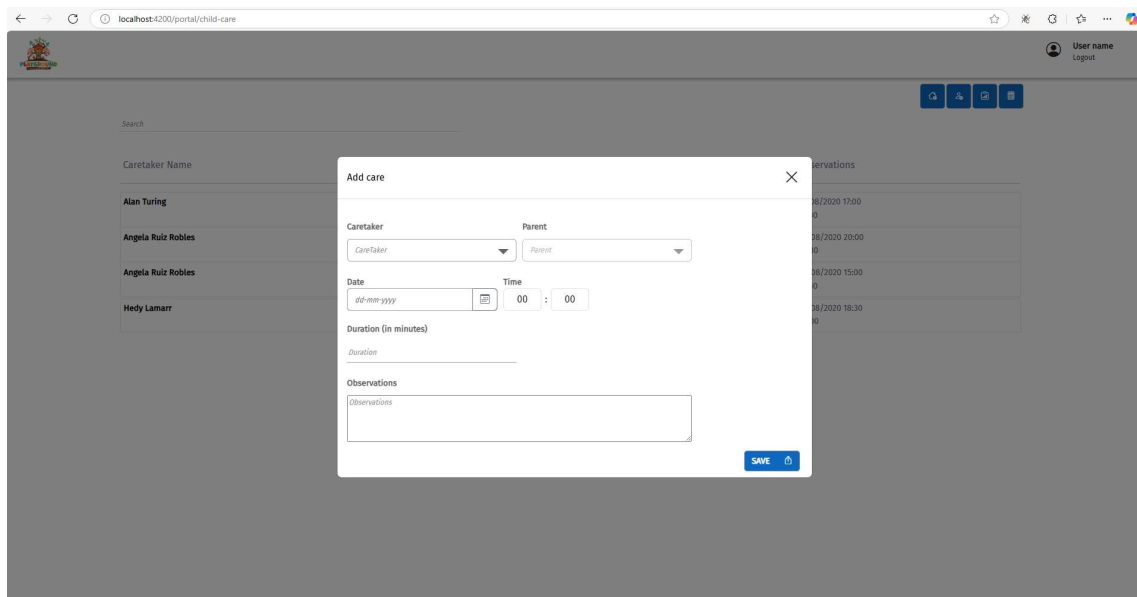
The rest of functionalities can be tested with the four buttons upper right

 Add childcare

 Add parent

 Show balance

 Net to 0

## Add chilcare

Allows the user to select a caretaker, then a parent (can't be the same as the caretaker) and then the rest of mandatory information.

The time-picker I've used is not the best in the world, but for the purpose of the development I think is enough.

## Add a parent



## Balance

# Transactions



**Time debts**

| Caretaker | Parent | Duration |
|---|---|---|
| Alan Turing | Angela Ruiz Robles | 01:00 |
| Ada Lovelace | Angela Ruiz Robles | 01:00 |
| Ada Lovelace | Hedy Lamarr | 00:30 |

## Comments

Some comments about the development:

- As I've said, the tests are not covering 100% of the code or all the use-cases, happy paths, errors… I've tried to cover a bit of everything.
- Validations, not all the validations are implemented, in a real case scenario all the validations will be there (empty values, error returned, undefined values, when enable/disable buttons…)
- Localize, all the texts are written in the code, in a real project I would have use a translation file to be able to cover multilanguage.