1. Statement

Goal

Create an application for searching and booking airline tickets.

Architecture

The application will be composed of a layer (frontend) made in React.

The React layer must exchange information with different services created with Java. Several microservices must be designed to serve to cover the different functionality required.

Minimum required microservices

The price requests according to some indicated parameters.
The flight reservation
Information about access to the application.

User-facing functionality

You must present a list of origins (minimum 5 origins). Once the origin has been selected, the API must be queried to obtain the possible destinations. After selecting the destination, a form will appear where you can select if it is a one-way or round trip.

The search must return different possibilities throughout the same day for the selected date, the 3 days before and the 3 days after the selected date. If any of the days prior to the searched date is less than the current day, you must add a day to the following days.

The returned information must include both the company, the flight number, the time and day of departure, the transit time, if there is a layover (and how many), if it allows luggage in the cabin and the price of the journey.

You should be able to filter the results by the following parameters:

Air Line
scales
Whether or not to allow luggage
By schedule

Once a flight has been selected, the passengers on the flight will be asked, initially showing the fields to know the information of a single passenger. The necessary fields will be:

- Name and surname
- Nationality
- Identification (NIF or passport)
- Age (list with possible values < 2 years, between 2 and 9 years, > 9 years). The default value will be > 9 years

If you will carry bags (if selected you must show a warning that it will have an additional cost) Every time the number of passengers or the age is changed, it must call the API to obtain the price of the flight (it must return different prices depending on what was selected). If age < 2 years is selected, the price will not increase).

When everything is filled in and the process continues, it will be sent to a new page where it will be simulated that the flight is being booked.

It will be necessary to simulate a correct sale and another one that fails. To make it easier it will depend on a key value indicated on the name of the first passenger.

You have to create Unit Tests for the API calls and tests in Selenium for the interface made in React.

Extra

Add a page with information about the application (fictitious data).

Add a, fake, payment gateway screen.

Add an administration page (with user and password validation).

On the administration page the user can do the following:

- View completed (paid) flights
- View unfinished flights (not paid)
- View information about the number of views on the home page
- View information about the views of all pages, including the home page and the information page.
- View the number of times each origin has been selected
- View the number of times each destiny has been selected
- View statistical data about the number of passenger
- Airport management