

Statement

HotelBediaX, a new client of FDSA and one of the most important fictional companies in the hotel sector, has the need to completely renew its tourist destination management portal.

The objective of this test is that you, a front-end developer specialized in UI and UX, create a web application for HotelBediaX that operates on its destinations REST API. The backend will be developed later, but its developers have already defined the API interface as following:

```
1  using System;
2  using System.Collections.Generic;
3
4  namespace Web.Test;
5
6  0 references
7  public interface IDestination
8  {
9      0 references
10     public List<Destination> GetAll();
11     0 references
12     public bool DeleteById(int id);
13     0 references
14     public bool Update(Destination destination);
15     0 references
16     public bool Create(Destination destination);
17 }
18
19 3 references
20 public class Destination
21 {
22     0 references
23     public int Id { get; set; }
24     0 references
25     public string Name { get; set; }
26     0 references
27     public string Description { get; set; }
28     0 references
29     public string CountryCode { get; set; }
30     0 references
31     public DestinationType Type { get; set; }
32     0 references
33     public DateTime LastModify { get; set; }
34 }
35
36 1 reference
37 public enum DestinationType
38 {
39     0 references
40     Country,
41     0 references
42     City
43 }
```

The final application will consist of several modules, the first of which will be the Destinations module, which is the one you are going to develop. This app will allow the navigation between the different modules, although at the moment there is only this one.

The required operations that must be implemented are the following:

- Create a new destination
- Read all created destinations and their data
- Update their data
- Delete them
- Filtering the data

Those are the minimum required operations, but you are free to add others if you feel that they are useful.

Users have created a first sketch to give you the idea that they have in mind, but it's important that you create an optimal UI that maximizes the user experience, based on your own previous experiences. You have creative freedom on the module's design, as long as you cover the minimum requirements mentioned above.

Users are mainly workers of the company with low-to-medium informatics knowledge.

Destinations

Create destiny Modify destiny Remove destiny

Data selector

Search

ID (int)	Name (string)	Description (string)	CountryCode (string)	Type (enum)	Last Modif (DateTime)
Item 1					

Requirements

- Front-end:
 - SPA (Single Page Application)
 - React/Angular/AngularJS, whichever you prefer
 - It must be prepared to handle a high amount of destinations (more than 200.000 records).
 - Optimized UI and UX.
- Back-end call mocking:
 - Fake responses for each implemented call (including path)
- Navigation
 - Menu listing the distinct modules
- CRUD
 - 4 basic operations (create, read, update, delete)
- Executable, standalone demo web app