## 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:

```
using System;
     using System.Collections.Generic;
    namespace Web Test;
    public interface IDestination
         public List<Destination> GetAll();
        public bool DeleteById(int id);
         public bool Update(Destination destination);
         public bool Create(Destination destination);
    public class Destination
         public int Id { get; set; }
        public string Name { get; set; }
         public string Description { get; set; }
         public string CountryCode { get; set; }
         public DestinationType Type { get; set; }
         public DateTime LastModify { get; set; }
     public enum DestinationType
        Country,
        City
28
```

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.

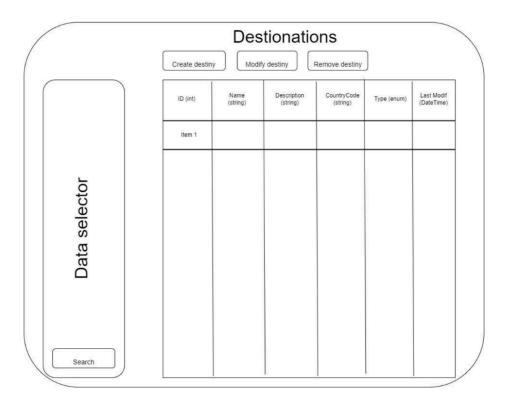
The required operations that must be implemented are the following:

- 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.



## Requirements

- Front-end:
  - SPA (Single Page Application)
  - o React/Angular/AngularJS, whichever you prefer
  - It must be prepared to handle a high amount of destinations (more than 200.000 records).
  - o Optimized UI and UX.
- Back-end call mocking:
  - o 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