Software Requirements Specification

Advisor Project

Created by Antoine Ratat, Bastien Ratat

Version 1.0 - Issued September 21, 2020

This requirements specification is used to record the user requirements.

Contents

1 VERSIONS

1.1 VERSIONS

Ver.	Author(s)	Date	Description
I.0	Antoine RATAT	9/21/2020	Création du document

2 INTRODUCTION

2.1 PURPOSE

Advisor enables travellers to browse interest points for a particular destination. They can use this application to organise their trip by checking the place rating, popularity, previewing a picture and find the location.

2.2 PROJECT SCOPE

Advisor is a programming project, that allows its developers to practice client-side programming using API calls.

2.3 REFERENCES

React - https://reactjs.org/
React Router - https://reactrouter.com/web/guides/quick-start
React Bootstrap - https://react-bootstrap.github.io/
React Spinners - http://www.davidhu.io/react-spinners/
React Promise Tracker - https://lemoncode.github.io/react-promise-tracker/React
Places Autocomplete - https://www.npmjs.com/package/react-places-autocomplete

3 DESCRIPTION

3.1 PRODUCT PERSPECTIVE

The product is a web-based system implemented on the client side, it is using API calls to fetch data.

3.2 FEATURES

The Advisor system provides simple mechanism for users to acquire information.

The following are the main features that are include in the system:

- **Cross Platform Support**: Offer support for most of the known and commercial operating systems
- **Search**: Search is based on Google Places API and city suggestions
- **Weather**: Allows users to get the latest weather update for a specific destination and predictions on the next three days.
- **Interests**: Provide users with a list of interests point to visit for a specific destination, each interest, would include more detailed information such as rating, location, picture.
- **Interest Map**: The system allows users to check interests directly on a map for their destination.

3.3 USER OVERVIEW

It is considered that the user does have the basic knowledge of operating the internet and to have access to it. The administrator is expected to be familiar with HTML, CSS, JavaScript, React library and AJAX (Asynchronous JavaScript and XML).

3.4 OPERATING ENVIRONMENT

This is a web-based system and hence will require the operating environment for a client and server GUI.

Dependencies

- This software highly depends on type and version of browser being installed in the system i.e. browser version should be used which have HTML5 support.

3.5 CONTRAINTS : IMPLEMENTATION / DESIGN

This system is provisioned to be built in JavaScript using React library which is highly flexible.

Hardware Interface

Device should be enabled with Internet.

Software interface

The user’s browser should be HTML5 compatible for a satisfactory user experience.

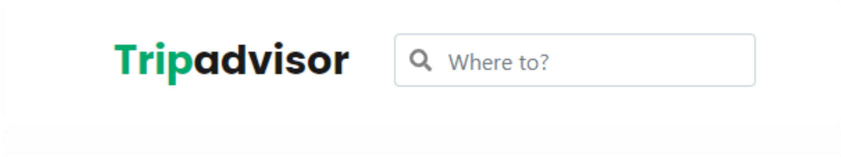
4 SYSTEM FEATURES

4.1 SYSTEM FEATURE 1

5 REQUIREMENTS OF EXTERNAL INTERFACE

5.1 USER INTERFACES

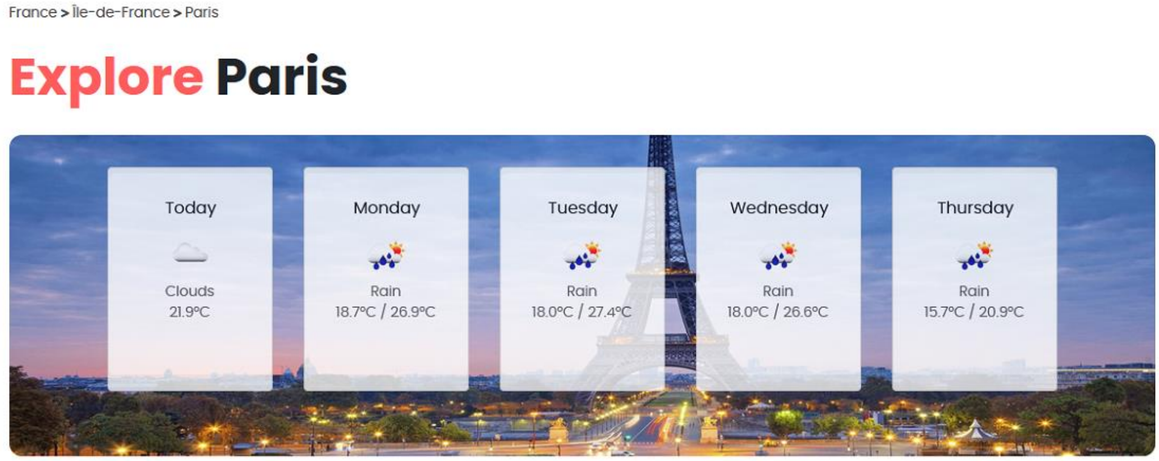
Navbar / Research Field:



Tripadvisor

Where to?

Weather Component:



France > Île-de-France > Paris

Explore Paris

Today
Clouds
21.9°C

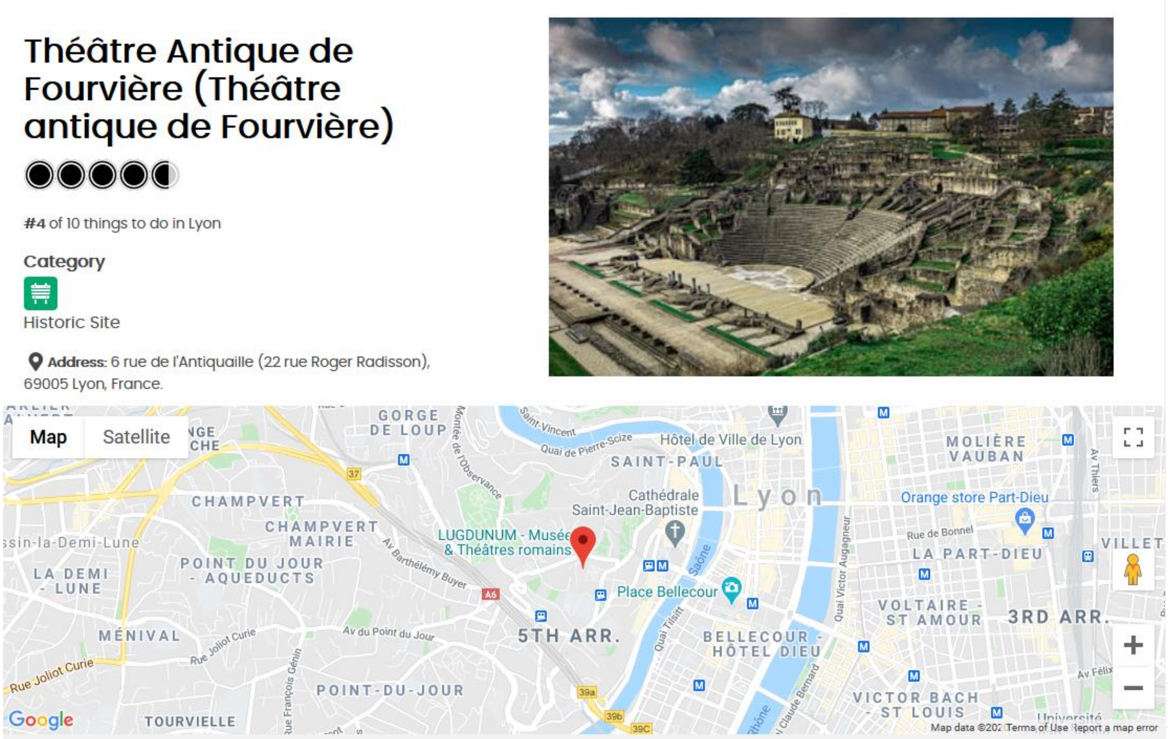
Monday
Rain
18.7°C / 26.9°C

Tuesday
Rain
18.0°C / 27.4°C

Wednesday
Rain
18.0°C / 26.6°C

Thursday
Rain
15.7°C / 20.9°C

Interest:



Théâtre Antique de Fourvière (Théâtre antique de Fourvière)

●●●●◐

#4 of 10 things to do in Lyon

Category

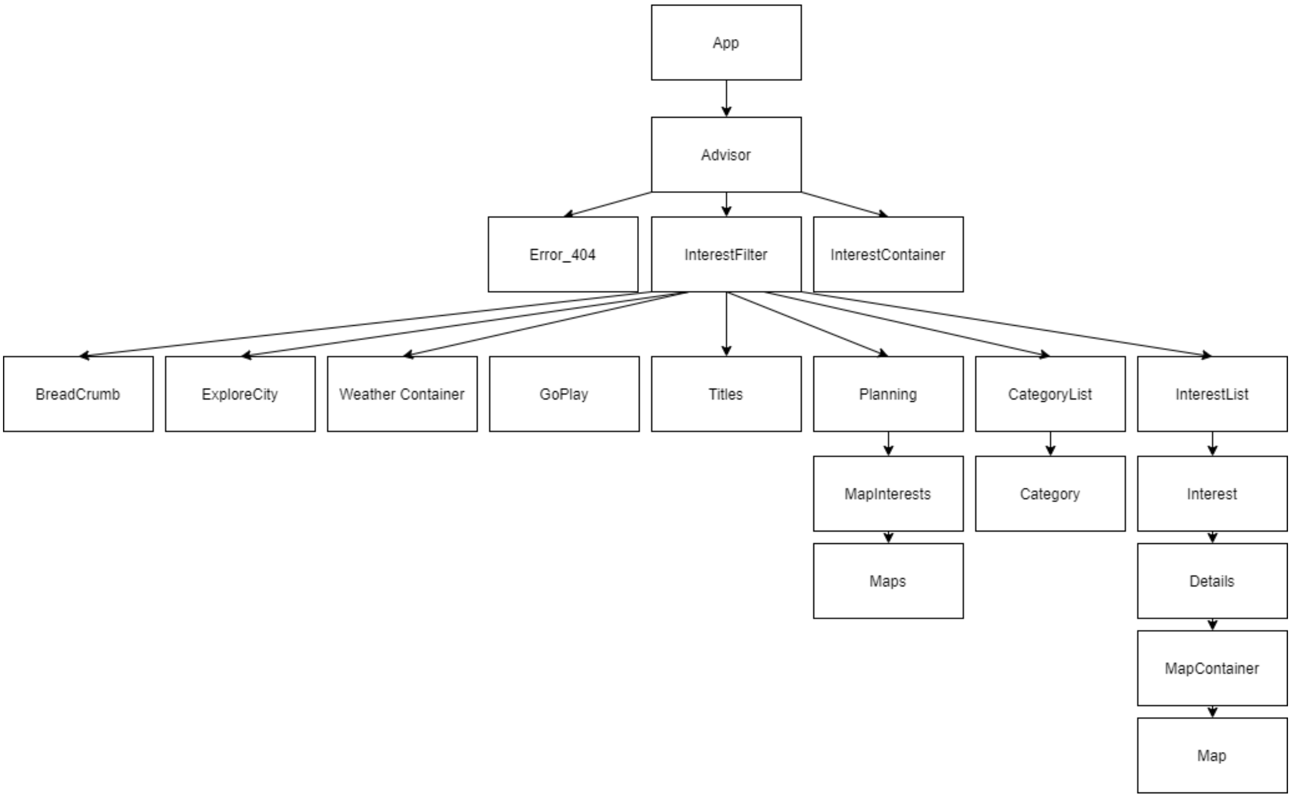
Historic Site

Address: 6 rue de l'Antiquaille (22 rue Roger Radisson), 69005 Lyon, France.

5.2 SOFTWARE INTERFACES

Software is designed in small fragmented atomic components. Each component has a specific functionality and assembled together create our application.

This is easier to maintain, replace and re-use. The component organisation of the Advisor app is available in the scheme below:



5.3 COMMUNICATION INTERFACES

Communication are assured to external interfaces. The system is connected to several APIs using REST (representational state transfer), The payload is defined in the request itself and is formatted in JSON.

- Fetch interests list from Foursquare API.
- Fetch place from interest name using Google Place API.
- Fetch image from photo reference using Google Place API.
- Fetch map using Google Map API.
- Fetch Weather using Openweathermap API.

6 ADDITIONAL NONFUNCTIONAL REQUIREMENTS

6.1 PERFORMANCE

The system must be interactive and the delays involved must be less. So, in every action-response of the system, there are no immediate delays. In case of opening new windows, of popping error messages and saving the settings or sessions there is delay much below 2 seconds.

In case of calling external APIs, the delay is based editing on the distance of the secondary system and the configuration between them so there is high probability that there will be or not a successful connection in less than 5 seconds for sake of good communication.

6.2 AVAILABILITY

If the internet service gets disrupted while fetching information from the external APIs, the information can be fetched again later.
The system is highly dependant on external APIs, if the communication between the client and the API is disrupted, the information could be requested again.

6.3 SECURITY

This application is not storing any user information, neither have access to modify external APIs.

6.4 USABILITY

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.