

# Pitch

Vous êtes chargé de concevoir une application qui permettra aux utilisateurs de choisir des activités en fonction des conditions météorologiques.

En effet, vous avez constaté, en réalisant une étude que bon nombre d'utilisateurs souhaiteraient pour organiser leur temps libre — et celui de leurs enfants, typiquement — en fonction de la météo annoncée. Si les conditions sont suffisamment favorables, ils souhaiteraient privilégier des activités de plein air, alors que si elles sont maussades, il préféreraient se réfugier à l'abri de bâtiments confortables.

Le projet que vous imaginez est réalisé à l'échelle d'une ville. Vous vous êtes mis en relation avec la municipalité qui souhaite promouvoir la richesse de son potentiel culturel, sportif et associatif. L'équipe municipale pourrait donc faire en sorte de mettre à votre disposition un catalogue des activités accessibles sur son territoire.

## Objectifs

L'objectif principal de l'application est de permettre aux utilisateurs de trouver des activités qui leur conviennent en fonction de certaines contraintes, notamment temporelles et météorologiques.

**Objectif 1** : Définir un modèle de données qui corresponde à la finalité de l'application ; ce modèle peut être implémenté de différentes manières, on s'attachera principalement ici à le décrire sous forme de schéma UML ou équivalent.

**Objectif 2** : Fournir un tableau de bord avec les prévisions météorologiques, pour une ville donnée, pour des dates déterminées

**Objectif 3** : Fournir une liste des activités possibles en fonction de contraintes de dates et de météo

**Objectif 4** : Etablir de profils d'utilisateur et concevoir un système simple de recommandation algorithmique

**Objectif 5** : Mettre en place un système de vote par préférence, selon la méthode Condorcet, pour déterminer quelle est l'activité préférée des utilisateurs

## Fonctionnalités

## Introduction

Cette application est un service. On ne s'occupe donc ici que de la partie « serveur ». On admettra qu'il existe par ailleurs une interface utilisateur, ou des clients divers, qui permettent d'accéder à ce service.

Dans ce qui suit, on considérera que l'accent est mis sur ce qui est produit (ou reçu) par le serveur. Ce qui importe est donc la réponse telle qu'elle est envoyée par ce dernier, typique sous forme d'un objet JSON, et non ce qui est, en fin de compte, perçu par l'utilisateur.

## Cas d'utilisation

- **UC1** : En tant qu'utilisateur, je vous connaître les prévisions météorologiques pour une date déterminée, en choisissant cette date
- **UC2** : En tant qu'utilisateur, je voudrais connaître les activités possibles à une date donnée, en choisissant cette date
- **UC3** : En tant qu'utilisateur, je voudrais savoir quelle activité seraient possibles, à une date déterminée, en fonction des prévisions météo
- **UC4** : En tant qu'utilisateur, j'aimerais pouvoir donner mon avis sur les activités que j'ai préférées, en les triant dans une liste qui m'est fournie
- **UC5** : En tant qu'administrateur, j'aimerais pouvoir enrichir une liste d'activités proposées dans ma commune
- **UC6** : En tant qu'administrateur,

## Contraintes

- **Contrainte 1** : Le code métier devra être écrit sous forme de classes selon les principes SOLID
- **Contrainte 2** : Vous devrez utiliser les services en ligne disponibles pour récupérer les données relatives à la météo et à la qualité de l'air
- **Contrainte 3** : Votre code devra être intégralement testé avec `pytest` et, éventuellement, avec une bibliothèque de tests d'intégration.
- **Contrainte 4** : Votre code devra être documenté de manière à être reconnu par un générateur de documentation comme Sphinx

## Ressources

### Météo

- OpenWeatherMap
- WeatherAPI

### Qualité de l'air

- OpenAQ
- AirVisual

## **Événements**

- TicketMaster