

# **Améliorez votre application avec des systèmes distribués**

# **Documentation fonctionnelle et technique**

## **Sommaire**

**1. Présentation du projet**

**1.1 Objectifs du projet**

**1.2 Hors du champ d’application**

**1.3 Mesures du projet**

**2. Spécifications fonctionnelles**

**3. Spécifications techniques**

**3.1 Schémas de conception technique**

**3.2 Glossaire**

**3.3 Solutions techniques**

**3.4 Autres solutions non retenues**

1. **Présentation du projet**

Améliorer l'application TourGuide, qui connaît une croissance rapide avec plus 30 000 utilisateurs connectés simultanément avec une estimation de 100 000 utilisateurs sous peu. Le public cible comprend les utilisateurs de l'application TourGuide et l'équipe de développement. Les avantages commerciaux incluent une meilleure expérience utilisateur, une gestion efficace de la croissance et une maintenance simplifiée de l'application.

### Objectifs du projet

Les objectifs comprennent la résolution des problèmes de lenteur lors de l'appel à gpsUtil et RewardsCentral, ainsi que l'optimisation des tests unitaires et la mise en place d'un pipeline d'intégration continue.

### Hors du champ d’application

Optimiser l’accès à la liste d’attractions, charger les attractions une fois par jour par exemple au lieu d’appeler le service GPSutil pour chaque utilisation pour ne plus dépendre d’un service externe.

### Mesures du projet

Les métriques du projet seront alignées sur le guide des étapes clés, nous permettant ainsi de suivre l'estimation sur l'avancement global tout en avançant sur les différentes étapes et points de vigilance avec le guide prévue à cet effet.

## **Spécifications fonctionnelles**

Voici une liste non-exhaustive des différentes fonctionnalités de l’application :

Récupérer la localisation de l’utilisateur.

Récupérer les attractions touristiques à proximité de l’utilisateur.

Récupérer des réductions sur les séjours à l’hôtel / billets de spectacles.

Calculer les récompenses de l’utilisateur en fonction des attractions touristique visité.

Calculer les récompenses possibles des cinq attractions à proximité.

## **3. Spécifications techniques**

### 

### 3.1 Schémas de conception technique

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

**Services externes :**

**GpsUtil :**

Renvoie la localisation de l’utilisateur.

Renvoie la liste des attractions partenaire au service.

**TripPricer :**

Fournit en fonction des points de l’utilisateur des réductions aux différents séjour/attractions

**RewardsCentral :**

Détermine les points de récompense en fonction de l’attraction.

**Services & Controller internes :**

**TourGuideController :**

Gestion des requêtes HTTP ce soit pour les localisations, les attractions, les récompense ou encore les réductions.

**TourGuideService :**

Service fournissant des méthodes varier permettant d’appeler la localisation de l’utilisateur, récupérer des attractions à proximités ou encore de récupérer les réductions des séjours.

**RewardsService :**

Fournit des méthodes de calculs pour affecter les récompenses aux utilisateurs.

### 3.2 Glossaire

**Localisation :** Emplacement définit par une latitude et longitude.

### 3.3 Solutions techniques

***Tests unitaires qui échouent :***

Utilisation de la classe « CopyOnWriteArrayList » pour éviter les exceptions dû à la concurrence.

***5 attractions les plus proches :***

Création de « data transfert object » pour récupérer les données nécessaires.

***Améliorez les performances :***

Mise en place de CompletableFuture pour implémenter le multithreading sur les méthodes nécessitant plus de ressources.

***Pipeline d’intégration continue :***

Mise en place d’un pipeline d’intégration via GitHub Action.

### 3.4 Autres solutions non retenues

***Tests unitaires qui échouent :***

Mise en place de bloc ou méthodes synchronized.

La problématique étant un fort impact sur les performances limitant les accès du multithreading.

***Améliorez les performances :***

Mise en place de CompletableFuture avec un Executors : newSingleThreadExecutor.

La problématique étant une consommation de ressource trop importante sur le pipeline d’intégration en continue (GitHub Action).

### 

### 