REMERCIEMENT

TABLES DES MATIERES

[1. Introduction 4](#_Toc103108382)

[1.1. Contexte général du projet 4](#_Toc103108383)

[1.1.1. Ceci est un test 4](#_Toc103108384)

[1.2. Présentation générale du projet 4](#_Toc103108385)

[2. Analyse fonctionnelle 5](#_Toc103108386)

[2.1. Navigation entre écrans 5](#_Toc103108387)

[2.2. Diagramme Use Cases 6](#_Toc103108388)

[2.3. Descriptions textuelles des différents uses cases 7](#_Toc103108389)

[2.3.1. Use cases pour l’acteur Visiteur 7](#_Toc103108390)

[2.3.1.1. Créer un compte 7](#_Toc103108391)

[2.3.1.2. Effectuer une recherche 9](#_Toc103108392)

[2.3.1.3. Afficher les détails d’un trip 10](#_Toc103108393)

[2.3.2. Use cases pour l’acteur Utilisateur enregistré 11](#_Toc103108394)

[2.3.2.1. Se connecter 11](#_Toc103108395)

[2.3.2.2. Afficher ses trips 12](#_Toc103108396)

[2.3.2.3. Evaluer un trip 13](#_Toc103108397)

[2.3.2.4. Ajouter un trip aux favoris 14](#_Toc103108398)

[2.3.2.5. Créer un trip 15](#_Toc103108399)

[2.2.3. Use cases pour l’acteur Utilisateur trip participant 16](#_Toc103108400)

[2.2.3.1. Ajouter un point d’intérêt à un trip 16](#_Toc103108401)

[2.2.3.2. Modifier une étape 18](#_Toc103108402)

[2.2.4. Use cases pour l’acteur Utilisateur trip manager 19](#_Toc103108403)

[2.2.4.1. Modifier un trip 19](#_Toc103108404)

[2.2.4.2. Publier un trip 20](#_Toc103108405)

[2.2.4.3. Partager un trip 21](#_Toc103108406)

[2.3. Diagramme de navigation entre écrans 23](#_Toc103108407)

[2.4. Règles fonctionnelles 23](#_Toc103108408)

[3. Architecture générale 23](#_Toc103108409)

[3.2. DB + API + SPA + Mobile 23](#_Toc103108410)

[4. Base de données 24](#_Toc103108411)

[4.2. Schéma de base de données 24](#_Toc103108412)

[4.3. Tables : Contraintes et clefs 25](#_Toc103108413)

[4.3.1. Users 25](#_Toc103108414)

[4.3.2. Cities 25](#_Toc103108415)

[4.3.3. TripsUsers 25](#_Toc103108416)

[4.3.4. Trips 25](#_Toc103108417)

[4.3.5. Itinaries 25](#_Toc103108418)

[4.3.6. ItinariesPlaces 26](#_Toc103108419)

[4.3.7. Places 26](#_Toc103108420)

[4.3.8. ItinariesPlacesCategories 26](#_Toc103108421)

[4.3.9. Categories 26](#_Toc103108422)

[5. Web API 27](#_Toc103108423)

[5.3. CQRS 27](#_Toc103108424)

[5.4. Sécurité 27](#_Toc103108425)

[5.5. Business Controller 27](#_Toc103108426)

[6. SPA 27](#_Toc103108427)

[6.3. React 27](#_Toc103108428)

[6.4. Leaflet et OpenStreetMap 27](#_Toc103108429)

[7. Flutter 27](#_Toc103108430)

[8. Conclusion 27](#_Toc103108431)

# Introduction

## Contexte général du projet

## Présentation générale du projet

# Analyse métier

## Description de la solution envisagée

Création d’une suite d’applications permettant de créer et consulter des itinéraires pour la visite d’une ville ou l’organisation d’un city trip.

Après la création d’un compte, l’utilisateur aura la possibilité de créer un city trip. En fonction du nombre de jours, l’utilisateur pourra ensuite créer un itinéraire pour chaque jour. À l’aide d’une carte interactive, l’utilisateur choisit différents points d’intérêts pour un itinéraire. L’application permet ensuite de calculer la route optimale en fonction des étapes sélectionnées.

Lors de la configuration de l’itinéraire, une liste de suggestion d’étapes populaires seront proposées aux utilisateurs. Cette liste sera composée en fonction des étapes précédemment sélectionnées par les autres utilisateurs de l’application.

Un utilisateur peut inviter un autre utilisateur à participer à la création du city trip. Le créateur du city trip peut ensuite décider de publier son trip publiquement.

Les city trips publiés seront accessibles à tous les utilisateurs. Les utilisateurs pourront ainsi accéder à une liste de trip qu’il sera possible de trier par ville, distance, nombre de jours, popularité. Les utilisateurs pourront ajouter des trips dans leur liste de favoris.

Un système de rating sera mis en place ou chaque utilisateur aura l’occasion de donner une note pour un itinéraire publier publiquement.

Les itinéraires créer sont prévus pour être fait exclusivement à pied. En effet, l’intention est de découvrir une ville en la parcourant à pied.

Le projet comportera 2 applications, la première pour créer et configurer des trips, la deuxième pour consulter les trips publiés. Les utilisateurs pourront ainsi en première lieu créer et organiser leur trip avec une application web, et par après, les consulter en temps réel avec l’application mobile afin de faciliter son voyage.

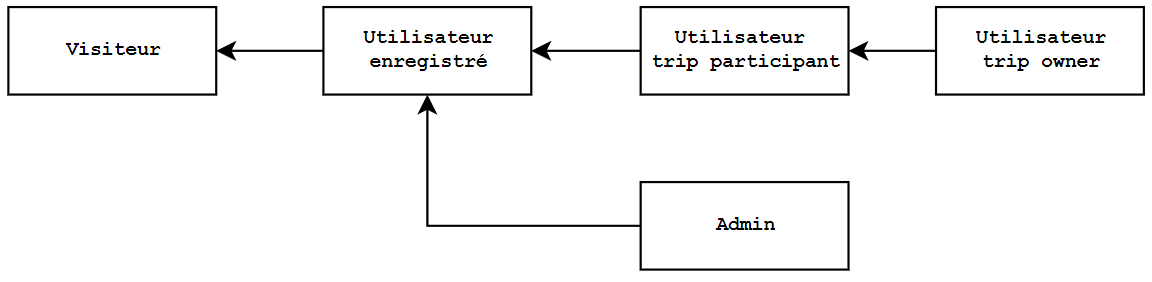
* **Application web** : Cette application permettra aux utilisateurs de créer et configurer des trips. Il sera possible d’ajouter des étapes, des générer des itinéraires, du publier et partager leurs trips avec d’autres utilisateurs. Il sera également possible de consulter les trips qui ont été publié publiquement.
* **Application mobile** : Cette application permettra aux utilisateurs de consulter leurs trips. Il sera possible de voir sa position sur la carte afin de pouvoir facilement suivre le parcours établit préalablement.

## Intervenants

Plusieurs types d’intervenants seront rencontrée dans l’application. Ces utilisateurs sont différenciés en fonction de leur relation avec un trip.

* *Visiteur* : Intervenant n’ayant pas de compte. Le visiteur peut accéder aux fonctionnalités ne nécessitant pas de connexion.
* *Utilisateur enregistré* : Un utilisateur enregistré est un utilisateur qui a préalablement créer un compte utilisateur et est identifié comme tel. Cet utilisateur à les même privilège qu’un visiteur. Il lui est donné la possibilité de créer son propre trip. Il peut également ajouter des trips dans ses favoris et donner une note à des trips publier.
* *Utilisateur trip participant* : Un utilisateur participant à les même privilèges qu’un utilisateur enregistré. Cet utilisateur est inscrit comme participant à un trip préalablement créer. Il à le droit d’ajouter, modifier, ou, supprimer des étapes.
* *Utilisateur trip owner* : Un utilisateur créateur à le même privilèges qu’un utilisateur participant. Il est le propriétaire du trip qu’il a créé. Il lui est donné la possibilité de publier son trip afin de le rendre visible pour les utilisateurs enregistrés. Il peut également inviter des utilisateurs à participer son trip.
* Administrateur : Il peut accéder à la section de gestion de étapes suggérées afin de contrôle les étapes pouvant être recommandées par des utilisateurs à d’autres utilisateurs. Il a également accès à la liste des utilisateurs.

Les différents intervenants s’inscrivent dans une relation d’héritage.



## Fonctionnalités

### Exigences fonctionnelles

|  |  |
| --- | --- |
| **Fonctionnalités Visiteur** | |
| ***Code*** | ***Description*** |
| **EF-001** | Créer un compte |
| **EF-002** | Effectuer une recherche |
| **EF-003** | Afficher les détails d’un trip publié |

|  |  |
| --- | --- |
| **Fonctionnalités Utilisateur enregistré** | |
| ***Code*** | ***Description*** |
| **EF-004** | Se connecter |
| **EF-005** | Donner une note à un trip |
| **EF-006** | Ajouter un trip aux favoris |
| **EF-007** | Afficher ses trips |
| **EF-008** | Créer un trip |

|  |  |
| --- | --- |
| **Fonctionnalités Utilisateur participant** | |
| ***Code*** | ***Description*** |
| **EF-009** | S’inscrire à un trip |
| **EF-010** | Ajouter une étape |
| **EF-011** | Modifier une étape |
| **EF-012** | Supprimer une étape |
| **EF-013** | Générer un itinéraire |

|  |  |
| --- | --- |
| **Fonctionnalités Utilisateur participant** | |
| ***Code*** | ***Description*** |
| **EF-014** | Modifier un trip |
| **EF-015** | Supprimer un trip |
| **EF-016** | Partager un trip |
| **EF-017** | Publier un trip |

|  |  |
| --- | --- |
| **Fonctionnalités Administrateur** | |
| ***Code*** | ***Description*** |
| **EF-018** | Confirmer une étape comme suggestions |
| **EF-019** | Ajouter une catégorie |

### Exigences non fonctionnelles

# Analyse fonctionnelle

## Navigation entre écrans

Diagram

Description automatically generated

## Diagramme Use Cases

Diagram

Description automatically generated

## Descriptions textuelles des différents uses cases

### Use cases pour l’acteur Visiteur

#### Créer un compte

1. Créer un compte
   1. Le visiteur procède à la création d’un compte
      1. Visiteur, système (acteur secondaire)

*Pré-conditions :*

* Le visiteur n’a pas de compte

*Scénario nominal :*

1. Le visiteur accède à l’application web ou mobile
2. Le visiteur navigue vers la section « Login »
3. Le système présente un formulaire d’authentification au Visiteur
4. Le visiteur clique sur l’onglet « Créer un compte »
5. Le visiteur encode son adresse email, son mot de passe et clique « Envoyer »
6. Le système enregistre le visiteur en base de données
7. Le système redirige le visiteur vers l’écran d’accueil

*Enchainements alternatifs* *:*

1. Le visiteur choisis l’option de s’authentifier avec un compte Google préexistant

Démarre au point 3 du scénario nominal

1. Le visiteur clique sur l’option « Se logger avec Google »
2. Le système redirige le visiteur vers l’écran d’accueil

*Enchainement d’erreurs :*

1. Le visiteur encode des données erronées

Démarre au point 5 du scénario nominal

1. Le système présente au visiteur les erreurs de validation

Le scénario nominal reprend au point 5.

*Post-conditions :*

* Le visiteur est enregistré en base de données
* Le visiteur devient un utilisateur enregistré.

*Visuel/Ecran :*

#### Effectuer une recherche

1. Effectuer une recherche
   1. Le visiteur peut rechercher un trip en fonction de critères spécifiques.
      1. Visiteur, système

*Pré-conditions :* aucune

*Scénario nominal :*

1. Le visiteur accède à l’application web ou mobile
2. Le visiteur navigue vers la section « découvrir »
3. Le visiteur encode les critères de recherche et lance la recherche
4. Le système récupère les trips correspondant aux critères de recherche
5. Le système affiche les résultats de la recherche

*Enchainements alternatifs* *:*

1. Le visiteur choisis l’option « critères de recherche avancée » pour affiner sa recherche

Démarre au point 3 du scénario nominal

1. Le visiteur encode les critères de recherche avancé et lance la recherche

Le scénario nominal reprend au point 4.

*Enchainement d’erreurs :*

1. Le visiteur encode des données erronées

Démarre au point 3 du scénario nominal.

1. Le système présente au visiteur les erreurs de validation

Le scénario nominal reprend au point 3.

*Post-conditions :*

* La liste des trips est affichée en fonctions de critères de recherche.

*Visuel/Ecran :*

#### Afficher les détails d’un trip

1. Afficher les détails d’un trip
   1. Le visiteur peut consulter en détails un trip
      1. Visiteur, Système

*Pré-conditions :*

* Effectuer une recherche

*Scénario nominal :*

1. Le visiteur clique sur le bouton « détails » sur la fiche du trip
2. Le système récupère les informations de ce trip
3. Le système redirige le visiteur vers l’écran détails d’un trip

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :* aucun

*Post-conditions :*

*Visuel/Ecran :*

### Use cases pour l’acteur Utilisateur enregistré

#### Se connecter

1. Se connecter
   1. Le visiteur peut se logger afin de s’identifier comme utilisateur enregistré
      1. Utilisateur enregistré, système

*Pré-conditions :*

* L’utilisateur à créer un compte
* L’utilisateur n’est pas encore connecté

*Scénario nominal :*

1. L’utilisateur accède à l’application web ou mobile
2. L’utilisateur navigue vers la section « Login »
3. Le système présente un formulaire d’authentification à l’utilisateur
4. L’utilisateur clique sur l’onglet « Se Logger »
5. Le visiteur encode son adresse email, son mot de passe et clique « Envoyer »
6. Le système vérifie les informations entrées par l’utilisateur
7. Le système redirige le visiteur vers l’écran d’accueil

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :*

1. Le visiteur encode des données erronées

Démarre au point 5 du scénario nominal

1. Le système présente au visiteur les erreurs de validation
2. Le visiteur encode à nouveau ses informations

Le scénario nominal reprend au point 6.

*Post-conditions :*

* L’utilisateur est connecté à l’application

*Visuel/Ecran :*

#### Afficher ses trips

1. Afficher ses trips
   1. L’utilisateur participant peut afficher ses différents trips
      1. Utilisateur enregistré, système

*Pré-conditions :*

* L’utilisateur a été ajouté comme participant pour un trip
* Ou, l’utilisateur a ajouté des trips dans ses favoris
* Ou, l’utilisateur a créé un trip

*Scénario nominal :*

1. L’utilisateur accède à l’application web ou mobile
2. L’utilisateur navigue vers la section « Mes trips »
3. Le système récupère les trips dans la base de données
4. Le système redirige l’utilisateur vers l’écran « Mes trips »

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :* aucun

*Post-conditions :*

* Les trips sont affichés

*Visuel/Ecran :*

#### Evaluer un trip

1. Evaluer un trip
   1. L’utilisateur peut noter un trip
      1. Utilisateur enregistré, système

*Pré-conditions :*

* Se connecter
* Afficher les détails d’un trip

*Scénario nominal :*

1. L’utilisateur clique sur le bouton « Note »
2. Le système présente un formulaire à l’utilisateur
3. L’utilisateur cliquer sur le nombre d’étoiles qu’il attribue à ce trip et confirme
4. Le système notifie l’utilisateur de la notation du trip

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :* aucun

*Post-conditions :*

* La note global du trip est calculé en fonction de la note attribué par l’utilisateur

*Visuel/Ecran :*

#### Ajouter un trip aux favoris

1. Ajouter un trip aux favoris
   1. L’utilisateur peut ajouter un trip dans ses favoris
      1. Utilisateur enregistré, système

*Pré-conditions :*

* Se connecter
* Afficher les détails d’un trip

*Scénario nominal :*

1. L’utilisateur clique sur le bouton « Favoris + »
2. Le système enregistre le trip dans les favoris de l’utilisateur
3. Le système notifie l’utilisateur de l’ajout du trip dans ses favoris

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :* aucun

*Post-conditions :*

* Le trip est ajouté aux favoris de l’utilisateur enregistré.

*Visuel/Ecran :*

#### Créer un trip

1. Créer un trip
   1. L’utilisateur enregistré peut créer un trip
      1. Utilisateur enregistré, système

*Pré-conditions :*

* Se connecter

*Scénario nominal :*

1. L’utilisateur accède à l’application web ou mobile
2. L’utilisateur navigue vers la section « créer »
3. Le système présente un formulaire de création de trip
4. L’utilisateur encode les informations du trip et clique sur envoyer
5. Le système enregistre le trip dans la base de données
6. Le système redirige l’utilisateur enregistré vers l’écran de configuration de trip

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :*

1. Le visiteur encode des données erronées

Démarre au point 4 du scénario nominal

1. Le système présente au visiteur les erreurs de validation
2. Le visiteur encode à nouveau ses informations

Le scénario nominal reprend au point 5.

*Post-conditions :*

* L’utilisateur enregistré devient utilisateur trip manager pour le trip

*Visuel/Ecran :*

### Use cases pour l’acteur Utilisateur trip participant

#### Ajouter un point d’intérêt à un trip

1. Ajouter un point d’intérêt à un trip
   1. L’utilisateur peut ajouter un point d’intérêt à un trip
      1. Utilisateur participant, système

*Pré-conditions :*

* Se connecter
* Afficher les détails d’un trip
* L’utilisateur est un utilisateur participant

*Scénario nominal :*

1. L’utilisateur clique sur un endroit dans sur la carte interactive
2. Le système enregistre l’étape dans la base de données
3. Le système notifie l’utilisateur de l’ajout de l’étape

*Enchainements alternatifs* *:*

1. L’utilisateur effectue une recherche d’un lieu spécifique dans la barre de recherche

Démarre au point 0 du scénario nominal.

1. L’utilisateur encode un le nom d’un lieu
2. Le système récupère les lieux qui corresponde au critère de recherche
3. L’utilisateur clique sur le lieu qu’il souhaite ajouter

Le scénario nominal reprend au point 2.

1. L’utilisateur ajoute une étape à partir d’une liste de suggestion

Démarre au point 0 du scénario nominal.

1. L’utilisateur clique sur le bouton « suggestion »
2. Le système présente à l’utilisateur un écran avec une liste de suggestion
3. L’utilisateur clique sur le bouton « + » pour ajouter une étape recommandée

Le scénario nominal reprend au point 2.

*Enchainement d’erreurs :*

1. L’utilisateur encode des données erronées

Démarre au point 3 du scénario nominal

1. Le système présente au visiteur les erreurs de validation
2. Le visiteur encode à nouveau ses informations

Le scénario nominal reprend au point 5.

*Post-conditions :*

* Une étape est ajoutée à un trip

*Visuel/Ecran :*

#### Modifier une étape

1. Modifier une étape
   1. L’utilisateur peut modifier les informations d’une étape
      1. Utilisateur participant, système

*Pré-conditions :*

* Se connecter
* Afficher les détails d’un trip
* L’utilisateur est un utilisateur participant

*Scénario nominal :*

1. L’utilisateur clique sur une étape dessinée sur la carte interactive
2. Le système enregistre présente un formulaire de modification d’étape
3. L’utilisateur encode les informations de l’étape et clique sur confirmer
4. Le système enregistre les informations de l’étape en base de données

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :*

1. L’utilisateur encode des données erronées

Démarre au point 3 du scénario nominal

1. Le système présente au visiteur les erreurs de validation
2. Le visiteur encode à nouveau ses informations

Le scénario nominal reprend au point 4.

*Post-conditions :*

*Visuel/Ecran :*

### Use cases pour l’acteur Utilisateur trip manager

#### Modifier un trip

1. Modifier un trip
   1. L’utilisateur peut modifier les informations d’un trip
      1. Utilisateurs trip manager, système

*Pré-conditions :*

* Se connecter
* Afficher les détails d’un trip
* L’utilisateur est un trip manager

*Scénario nominal :*

1. L’utilisateur clique sur le bouton d’édition d’un trip
2. Le système présente un formulaire de modification du trip
3. L’utilisateur encode les informations du trip
4. Le système enregistre les nouvelles informations dans la base de données
5. Le système notifie l’utilisateur de la modification du trip

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :*

1. Le visiteur encode des données erronées

Démarre au point 3 du scénario nominal

1. Le système présente au visiteur les erreurs de validation
2. Le visiteur encode à nouveau ses informations

Le scénario nominal reprend au point 4.

*Post-conditions :*

*Visuel/Ecran :*

#### Publier un trip

1. Publier un trip
   1. L’utilisateur peut donner son accord pour partager son trip avec les autres utilisateurs de l’application
      1. Utilisateur trip manager, système

*Pré-conditions :*

* Se connecter
* Afficher les détails d’un trip
* L’utilisateur est un trip manager

*Scénario nominal :*

1. L’utilisateur clique sur le bouton d’édition d’un trip
2. Le système présente un formulaire de modification du trip
3. L’utilisateur cliquer sur l’onglet « Publier »
4. L’utilisateur confirme son intention de publier son trip publiquement.
5. Le système enregistre l’information dans la base de données
6. Le système notifie l’utilisateur de la publication de son trip.

*Enchainements alternatifs* *:* aucun

*Enchainement d’erreurs :* aucun

*Post-conditions :*

* Le trip est publié

*Visuel/Ecran :*

#### Partager un trip

1. Partager un trip
   1. L’utilisateur peut partager un trip avec un autre utilisateur afin de collaborer dans l’organisation du trip.
      1. Utilisateur trip manager, système.

*Pré-conditions :*

* Se connecter
* Afficher les détails d’un trip
* L’utilisateur est un trip manager

*Scénario nominal :*

1. L’utilisateur clique sur le bouton d’édition d’un trip
2. Le système présente un formulaire de modification du trip
3. L’utilisateur cliquer sur l’onglet « Partager »
4. L’utilisateur encode un mot de passe pour partager son trip et confirme
5. Le système enregistre l’information dans la base de données.
6. Le système notifie l’utilisateur du partage de son trip et lui donne un lien URL à transmettre aux potentiels utilisateur participant

*Enchainements alternatifs* *: aucun*

*Enchainement d’erreurs :*

1. L’utilisateur encode des données erronées

Démarre au point 4 du scénario nominal

1. Le système présente au visiteur les erreurs de validation
2. Le visiteur encode à nouveau ses informations

Le scénario nominal reprend au point 5.

*Post-conditions :*

* Le trip peut être partagé

*Visuel/Ecran :*

## Diagramme de navigation entre écrans

## Règles fonctionnelles

# Architecture générale

## DB + API + SPA + Mobile

# Base de données

## Schéma de base de données

Diagram

Description automatically generated

## Tables : Contraintes et clefs

### Users

1. user\_id
   1. /
      1. /
         1. user\_email

La table User représente un utilisateur. L’utilisateur possède un nom, un prénom, une adresse email.

### Cities

1. city\_id
   1. /
      1. /
         1. latitude, longitude, name

La table Cities représente une ville. Une ville possède un nom, un latitude et une longitude

### TripsUsers

1. user\_trip\_id
   1. user\_id, trip\_id
      1. /
         1. /

La table TripsUser représente la représente la relation entre un trip et un utilisateur. Cette table permet de déterminer si l’utilisateur est le manager d’un trip, ou, participant d’un trip.

Un utilisateur peut être liée à 0 ou plusieurs trips

### Trips

1. trip\_id
   1. city\_id
      1. /
         1. Username var char

La table Trips représente un trip. Il possède un nombre de jours et permet de déterminé si le trip est publié ou partagé.

### Itinaries

1. itinary\_id
   1. /
      1. /
         1. Username var char

Paragraphe d’explication de la table user

### ItinariesPlaces

1. userId
   1. /
      1. /
         1. Username var char

Paragraphe d’explication de la table user

### Places

1. userId
   1. /
      1. /
         1. Username var char

Paragraphe d’explication de la table user

### ItinariesPlacesCategories

1. userId
   1. /
      1. /
         1. Username var char

Paragraphe d’explication de la table user

### Categories

1. userId
   1. /
      1. /
         1. Username var char

Paragraphe d’explication de la table user

# Web API

## CQRS

## Sécurité

## Business Controller

# SPA

## React

## Leaflet et OpenStreetMap

# Flutter

# Conclusion