Cahier des spécifications

Partie 1 : Synthèse

Le projet

Nous vous présentons notre projet qui est une application nommée Found'em.

Membres de l'équipe

Notre équipe est composée de 4 étudiants Master 1 MIAGE de l'université Paris Nanterre. Fatimata Soumaré, Rizlane Abalil, Raphael Meissonnier et Imane Kadi.

Présentation

Il nous est à tous arrivé de perdre un objet, se dire que ce serait génial qu'une personne le trouve et nous contacte pour nous le rendre, ou même de trouver un objet et de vouloir le rendre au propriétaire avec un moyen simple. Nous avons donc mis en place ce projet pour répondre à cette problématique. L'objectif est de créer une plateforme qui va permettre aux utilisateurs de gagner du temps et de l'argent dans leurs démarches à propos d'un objet perdu ou trouvé. Le fait que ce projet traite un problème dont la plupart de la population mondiale a déja été confronté motive notre équipe à délivrer une solution et ainsi simplifier la vie des gens. Bien que ce problème touche tout le monde, il existe peu de solutions efficaces, souvent elles ne sont pas optimales au niveau du temps comme les services physiques d'objets trouvés (SNCF, RATP, Police...). Il existe tout de même des applications telles que Troov, ou Ppbot.

Personas et fonctionnalités de leur point de vue

```
1) Utilisateur :
- Il peut créer son espace personnel et s'y connecter.
- Il peut aussi remplir un formulaire via son espace personnel pour déclarer un objet
qu'il a perdu ou qu'il a trouvé.
- Il peut faire une recherche via le moteur de recherche.
- Il peut trier et filtrer ses recherches par caractéristiques, catégories, date
d'ajout.
- Il peut consulter la liste correspondante à sa recherche.
- Il peut consulter la fiche d'un objet.
- Il peut contacter le trouveur de l'objet via la fiche de l'objet.
- Il peut recevoir des notifications lorsqu'il reçoit un message via sa messagerie.
- Il peut via cette messagerie discuter avec des potentiels propriétaires d'objets
trouvés.
- S'il pense avoir trouvé son objet, il peut remplir un formulaire de test
d'auhentification de propriétaire.
- S'il réussi ce test, il peut fixer un rendez-vous dans son agenda via son espace.
- Il peut supprimer son annnonce via son espace personnel.
- Son solde de points de fidelité sera augmenté à chaque fois qu'un de ses objets
qu'il a trouvé est restitué au propriétaire.
```

```
2) Administrateur Web :
- Signaler un utilisateur
- Bannir de l'application un utlisateur
- Supprimer une annonce
- Consulter KPI (nombres d'objets perdus, retrouvés, pourcentage objets restitués,
...)
- Attribuer des points de fidélité
- Contacter un utilisateur via la messagerie
- Ajouter des blocs de messages
- Consulter les avis des utilisateurs
```

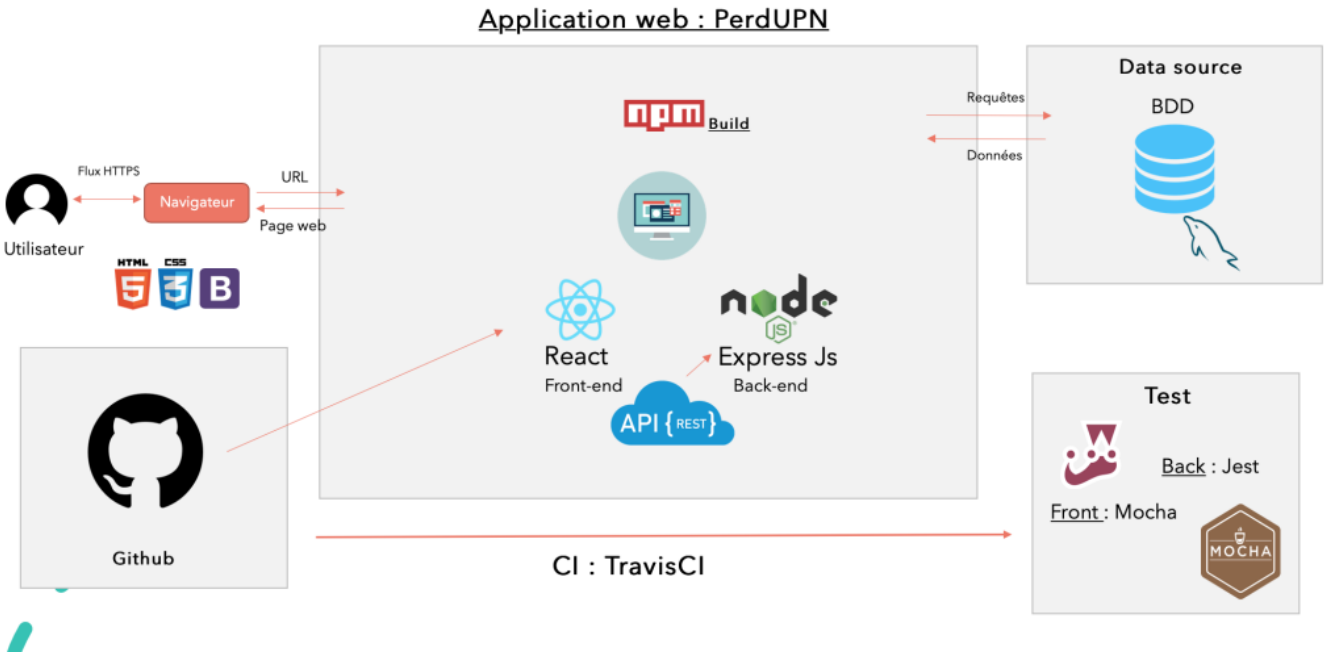
Prévisions marketing

La cible de notre produit est principalement les utilisateurs des réseaux. Pour cela, nous lancerons une campagne publicitaire sur les réseaux sociaux. Nous comptons aussi sur la bonne vieille méthode du bouche à oreille.

Partie 2 : Aspects techniques

Found’EM est une application Web.

Schéma d’architecture



Plateforme technologique

Found'Em est développée en :

- JavaScript avec React JS pour le front-end.
- Express JS pour le back-end.

Les tests se feront avec :

- Jest pour le back-end
- Mocha pour le front-end

L'ORM utilisé pour le lien entre la base de données et le code métier est Sequelise.

ExpressJS fera le lien métier et l'API.

La base de données sera en MySql.

Plateforme opérationnelle

```
Gestion de versions = Git
Le build = npm
Laualité de code = sonarQube
CI = Travis CI
```

Nous allons utiliser plusieurs API distantes telles que OpenLayers pour la cartographie (map).

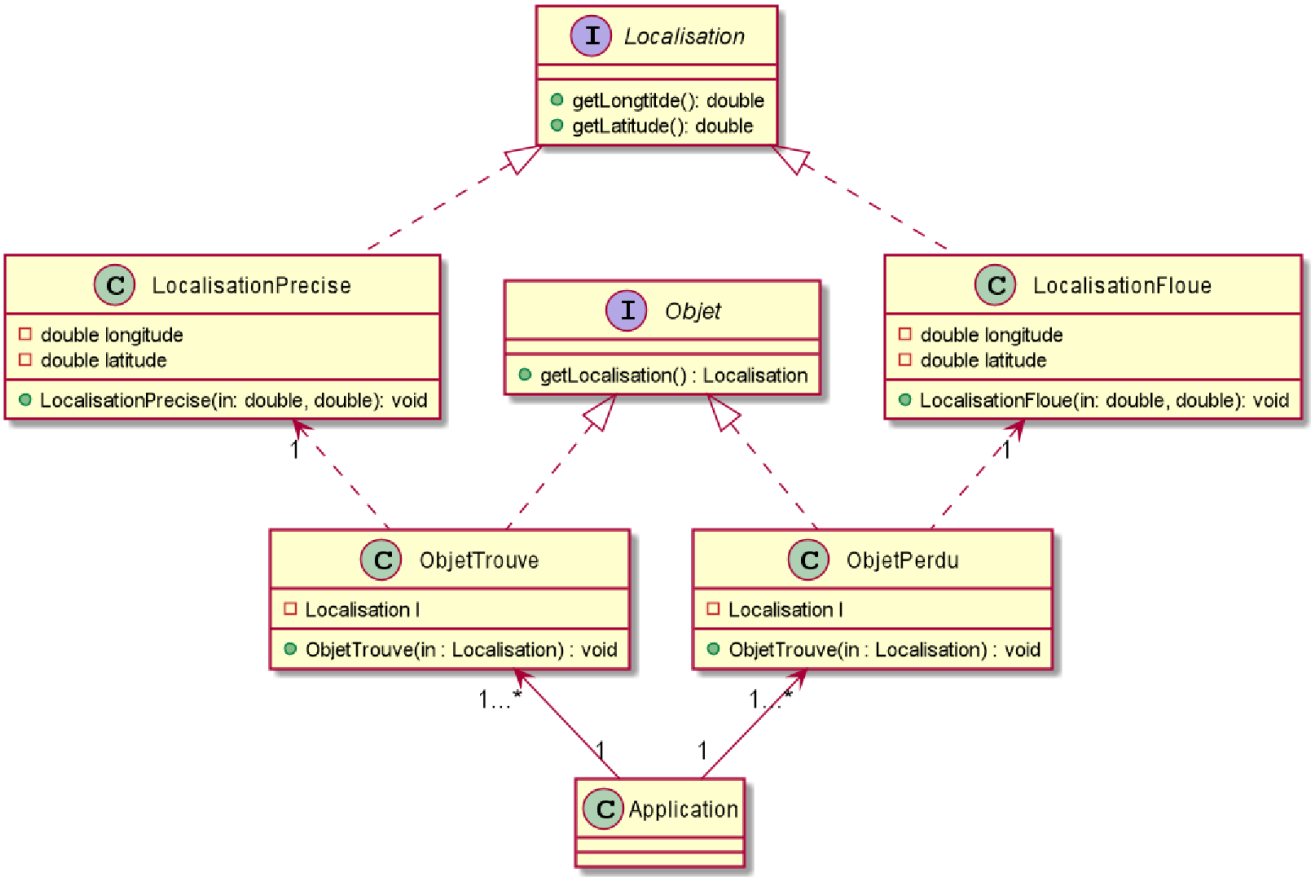
Concernant l'IA, nous pourrons comparer et mettre en relation des déclarations de pertes et

déclarations d'objets perdus similaires. Nous pourrons retrouver la personne dans la base de données grâce à certains objets (carte bancaire, CNI etc...). Nous pourrons également identifier les objets perdus grâce aux photos.

Partie 3 : Modélisation

Maquette du front

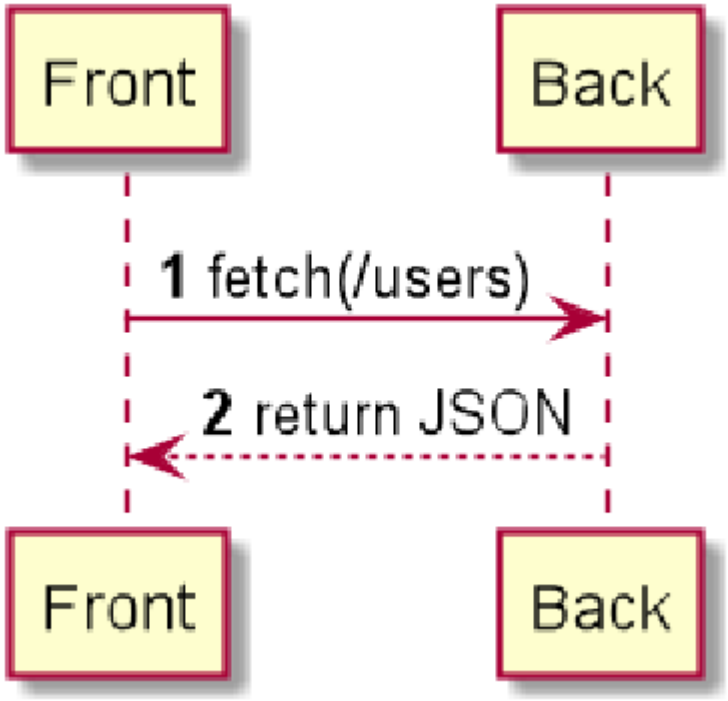
Diagramme de classe



Description de l'API

Définition	Description
GET /objets	La réponse retourne un tableau d'objets.

Diagramme de séquence


Front
Back
1 fetch(/users)
2 return JSON
Front
Back