

Cahier des spécifications

Table of Contents

Partie 1 : Synthèse.....	1
Le projet.....	1
Membres de l'équipe.....	2
Présentation.....	2
Personas et fonctionnalités de leur point de vue.....	2
Prévisions marketing.....	3
Partie 2 : Aspects techniques.....	4
Schéma d'architecture.....	4
Plateforme technologique.....	4
Plateforme opérationnelle.....	4
Partie 3 : Modélisation.....	6
Version v0.2.....	6
Version v0.3.....	6
Version v0.4.....	6
Version v0.5.....	7
Version v0.6.....	8
Version v0.7.....	8
Diagramme de classe.....	8
Description de l'API.....	8
Diagramme de séquence.....	9

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 déjà 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

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'authentification de propriétaire.
- S'il réussit ce test, il peut fixer un rendez-vous dans son agenda via son espace.
- Il peut supprimer son annonce via son espace personnel.
- Son solde de points de fidélité sera augmenté à chaque fois qu'un de ses objets qu'il a trouvé est restitué au propriétaire.

Administrateur Web :

- Signaler un utilisateur
- Bannir de l'application un utilisateur
- 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

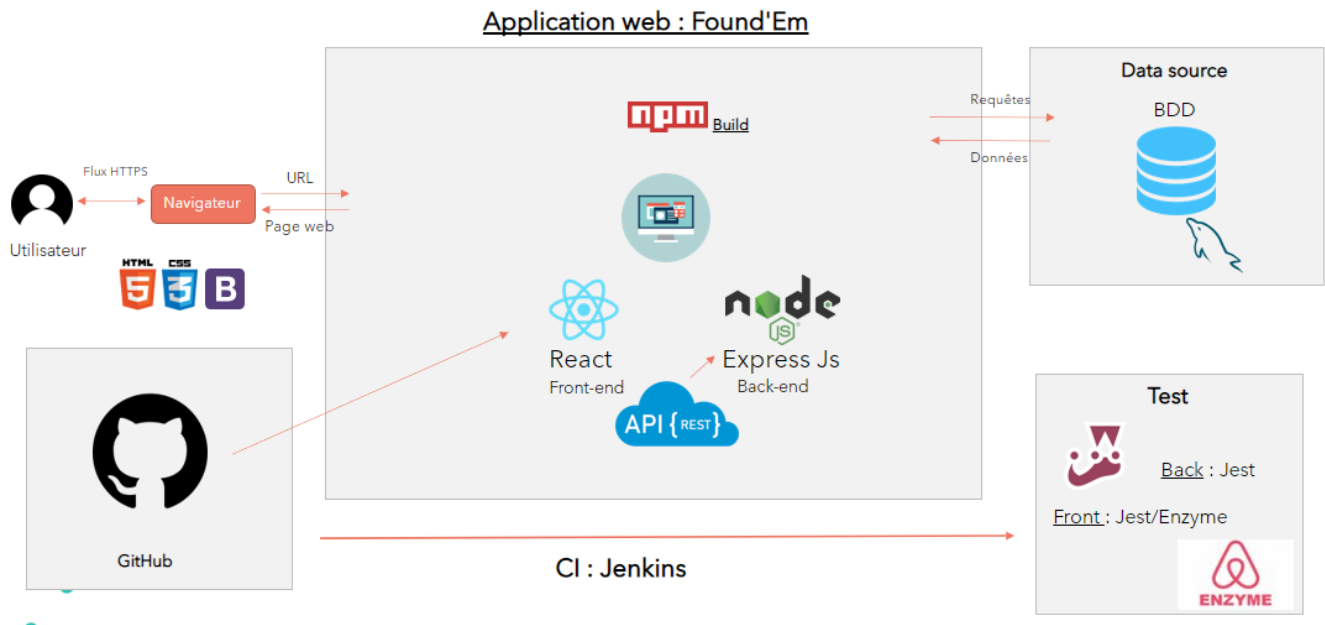
Prévisions marketing

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
- Jest/Enzyme pour le front-end

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

ExpressJS fera le lien métier et l'API.

La base de données sera en MySQL.

Plateforme opérationnelle

→ **Gestion de versions**

Git

→ **Le build**

npm

→ **La qualité de code**

sonarCloud

→ **CI**

Jenkins

Nous allons utiliser plusieurs API distantes telles que OpenLayers pour la cartographie (map) ou encore MapBox pour le Geocoding.

Concernant l'IA, nous pourrions comparer et mettre en relation des déclarations de pertes et déclarations d'objets perdus similaires. Nous pourrions retrouver la personne dans la base de données grâce à certains objets (carte bancaire, CNI etc...). Nous pourrions également identifier les objets perdus grâce aux photos.

Partie 3 : Modélisation

Version v0.2

Feature : Localiser l'utilisateur

[v0.2 localiser utilisateur] | *newWireframes/v0.2_localiser_utilisateur.PNG*

- L'utilisateur clique sur le bouton [**Centrer**]
- La page se rafraîchit
- La carte se centre sur la localisation de l'utilisateur

Version v0.3

Feature : Les objets les plus proches

[v0.3 objets plus proches] | *newWireframes/v0.3_objets_plus_proches.PNG*

- L'utilisateur accède à la page d'accueil
- L'utilisateur clique sur le bouton [**Centrer**]
- La carte se centre alors sur la position de l'utilisateur
- L'utilisateur peut consulter via la carte les objets les plus proches de sa localisation

Feature : Voir les points fusionnés

[v0.3 points fusionnées] | *newWireframes/v0.3_points_fusionnees.PNG*

- L'utilisateur dézoome la carte
- Les icônes fusionnent

Version v0.4

Feature : Consulter les informations d'un point sur la carte

[v0.4 consulter informations objet] | *newWireframes/v0.4_consulter_informations_objet.png*

- L'utilisateur accède à la page d'accueil de l'application
- L'utilisateur consulte la carte
- L'utilisateur clique sur un point
- Les informations concernant ce point s'affichent

Feature : Ajouter un objet trouvé

[v0.4 ajouter objet trouve 1] | *newWireframes/v0.4ajouter_objet_trouve_1.png*

[v0.4 ajouter objet trouve 2] | *newWireframes/v0.4_ajouter_objet_trouve_2.png*

- J'ai trouvé un objet perdu
- Je signale cet objet sur l'application web Found'Em
- Je suis sur la page d'accueil Home de Found'em
- Je clique sur le bouton [J'ai trouvé un objet]
- Je suis redirigé sur une nouvelle page "J'ai trouvé un objet"
- Je remplis le formulaire détaillé
- Je clique sur le bouton [Valider]
- Je suis redirigé(e) vers la page d'accueil

Version v0.5

Feature : Ajouter un objet perdu

[v0.5 ajout objet perdu1] | *newWireframes/v0.5_ajout_objet_perdu1.png*

[v0.5 ajout objet perdu2] | *newWireframes/v0.5_ajout_objet_perdu2.png*

- L'utilisateur accède à la page d'accueil du site
- L'utilisateur clique sur le bouton [J'ai perdu un objet]
- L'utilisateur est redirigé vers une nouvelle page
- L'utilisateur remplit le formulaire
- L'utilisateur valide les informations saisies en cliquant sur le bouton [Valider]
- L'utilisateur est redirigé vers la page d'accueil

Feature : Saisir une adresse sur le formulaire

[v0.5 saisir adresse formulaire] | *newWireframes/v0.5_saisir_adresse_formulaire.png*

- L'utilisateur accède au formulaire permettant de déclarer des objets trouvés
- Il saisit une adresse dans le champs Adresse
- Des résultats s'affichent au fur et à mesure de la saisie
- L'utilisateur sélectionne une adresse
- L'adresse s'affiche dans le champs de saisie "Adresse"

Feature : Chercher un objet perdu

[v0.5 recherche objet perdu1] | *newWireframes/v0.5_recherche_objet_perdu1.png*

[v0.5 recherche objet perdu2] | *newWireframes/v0.5_recherche_objet_perdu2.png*

- L'utilisateur accède à la page d'accueil du site

- L'utilisateur clique sur le bouton [J'ai perdu un objet]
- L'utilisateur est redirigé vers une nouvelle page
- L'utilisateur accède au formulaire permettant de chercher un objet perdu
- Il saisit un intitulé et coche des informations concernant l'objet
- L'utilisateur valide sa recherche avec le bouton [Rechercher un objet]
- Des résultats correspondant aux informations s'affichent

Version v0.6

Feature : Consulter les suggestions d'objets perdus

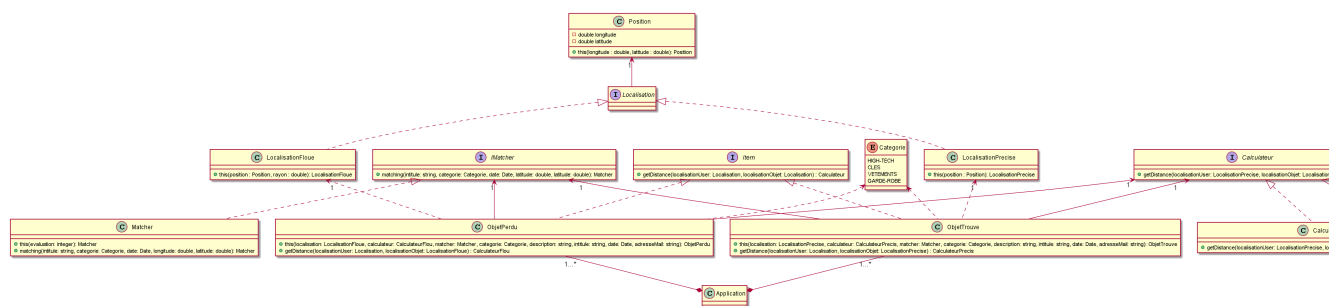
[v0.6 suggestions objets perdus1] | [newWireframes/v0.6_suggestions_objets_perdus1.png](#)

[v0.6 suggestions objets perdus2] | [newWireframes/v0.6_suggestions_objets_perdus2.png](#)

- L'utilisateur accède à la page d'accueil
- L'utilisateur consulte la liste des objets perdus proches de sa localisation

Version v0.7

Diagramme de classe



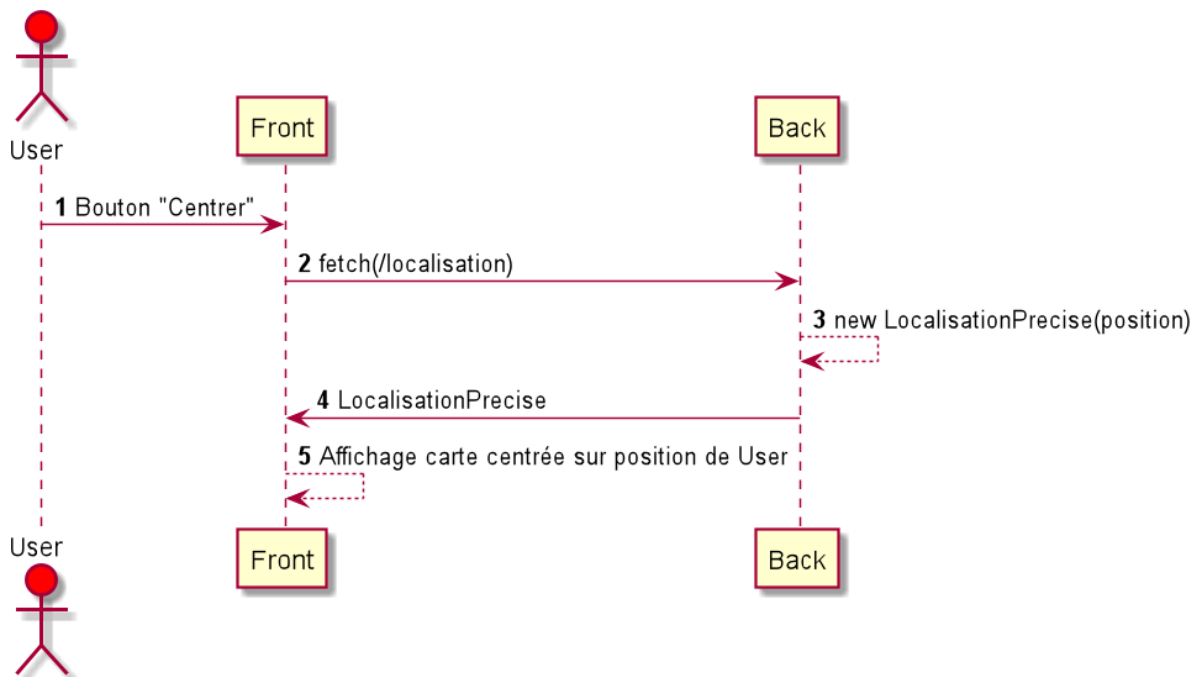
Description de l'API

Définition	Description
GET /objets/:longitude/:latitude/:rayon	La réponse retourne un tableau d'objets triés selon leur distance par rapport à l'utilisateur.
POST /localisation	La requête envoie la localisation de l'utilisateur
GET /objets/:longitude/:latitude	La réponse retourne un tableau d'objets perdus triés selon leur distance par rapport à l'utilisateur
POST /ajoutObjetTrouve	La requête envoie les informations d'un objet trouvé saisis par l'utilisateur

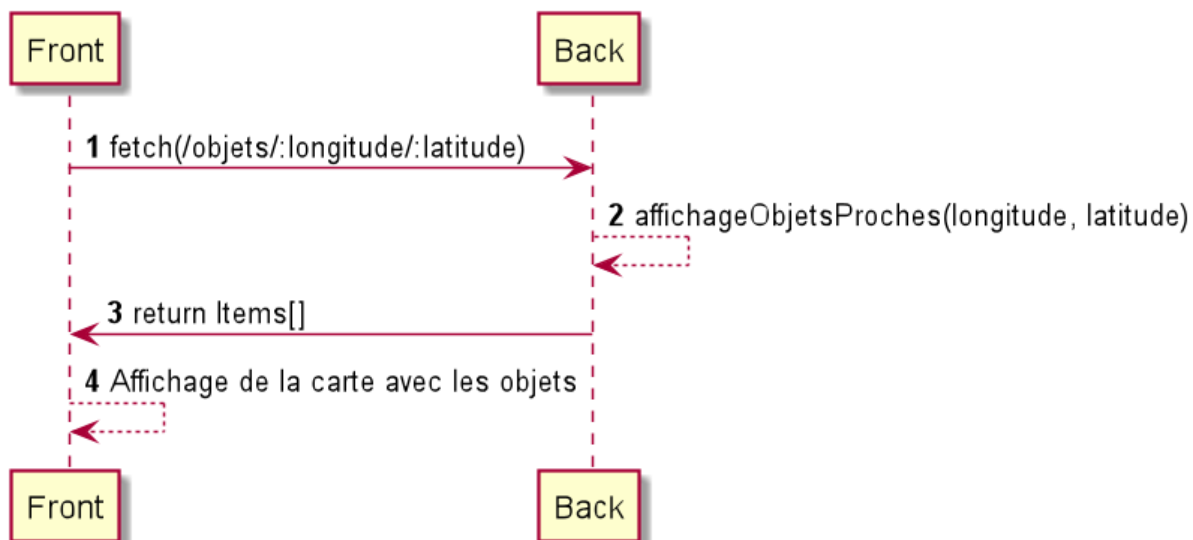
Définition	Description
POST /ajoutObjetPerdu	La requête envoie les informations d'un objet perdu saisies par l'utilisateur
GET /chercherObjetPerdu/:intitule/:categorie/:date/:longitude/:latitude	La réponse retourne une collection d'objets trouvés correspondant à des critères

Diagramme de séquence

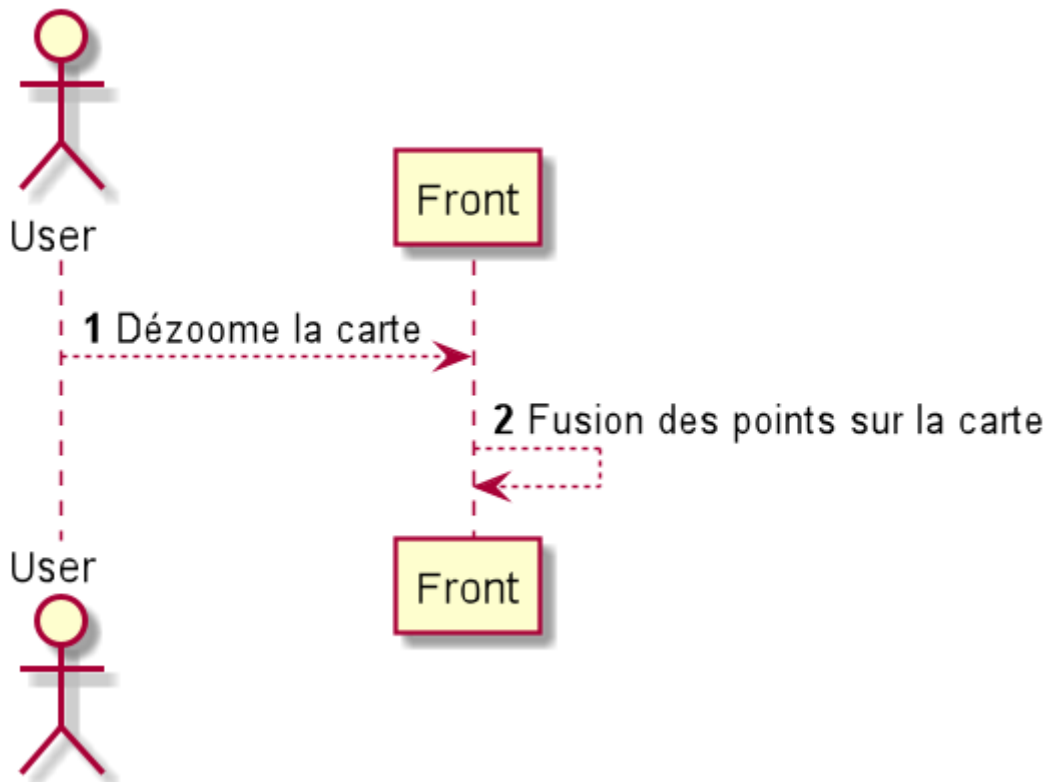
Feature "Localiser l'utilisateur"



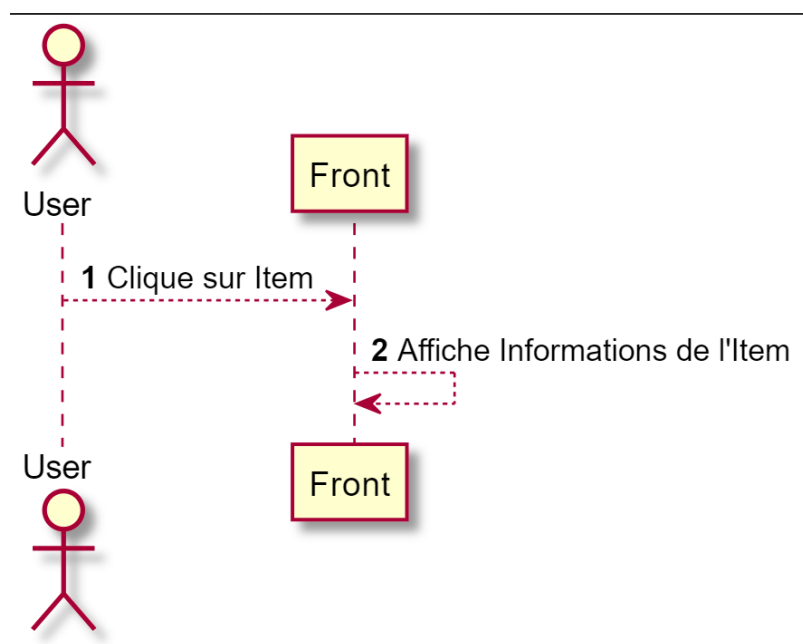
Feature "Afficher les objets les plus proches"



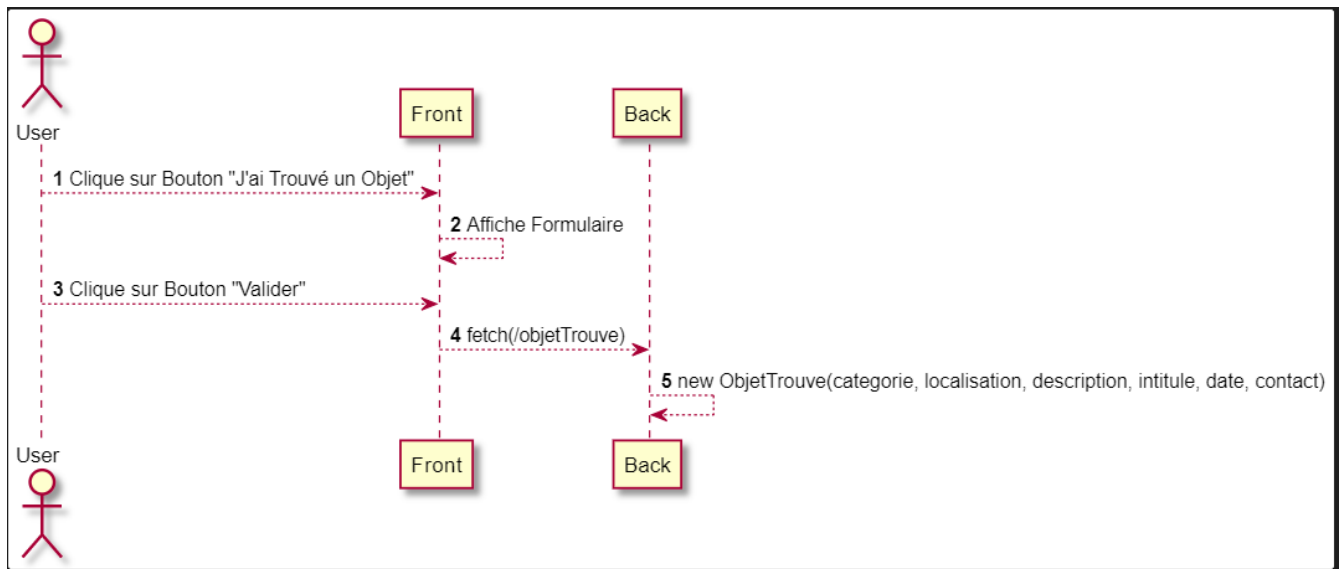
Feature "Voir les points fusionnés"



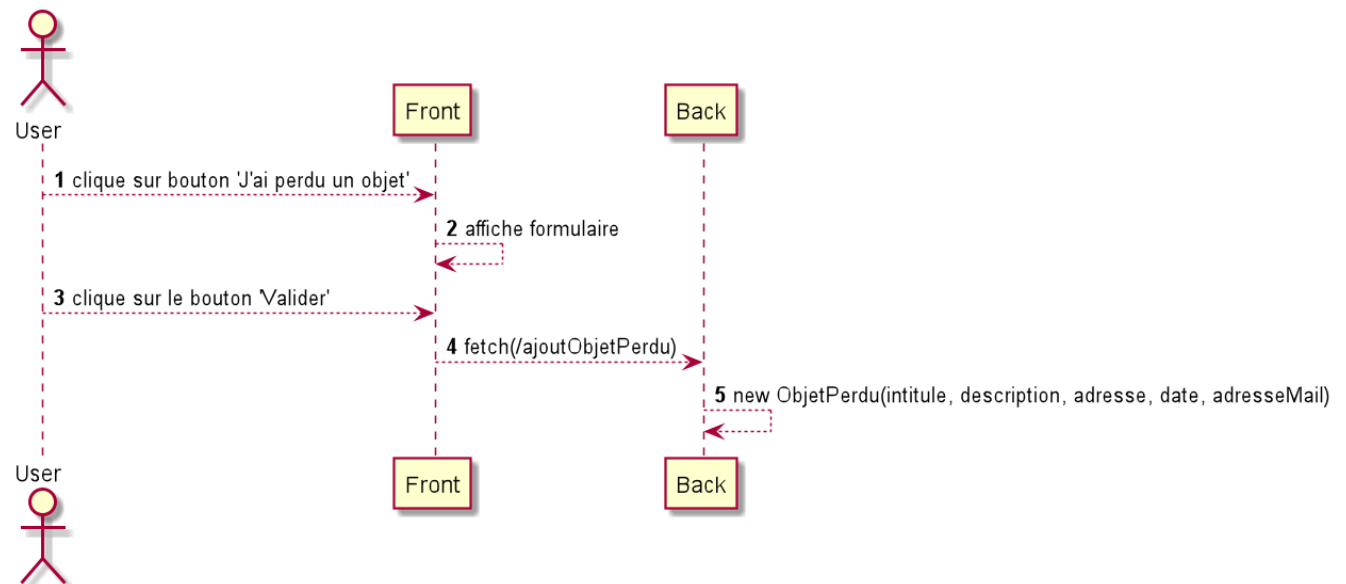
Feature "Voir Informations d'un Item"



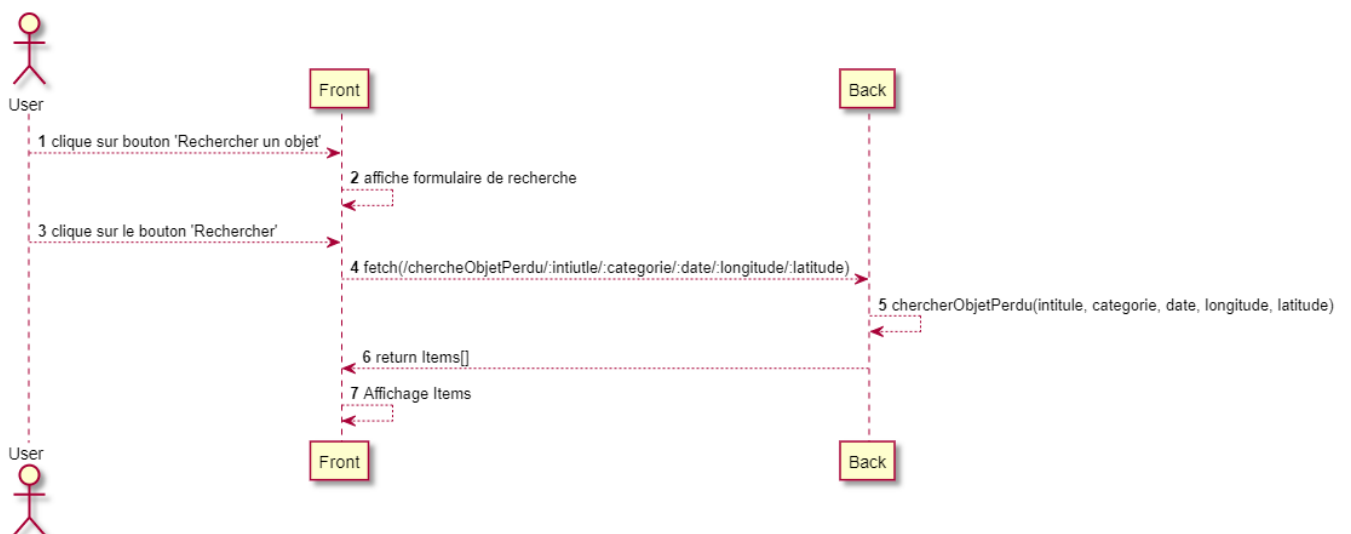
Feature "Declarer un Objet Trouvé"



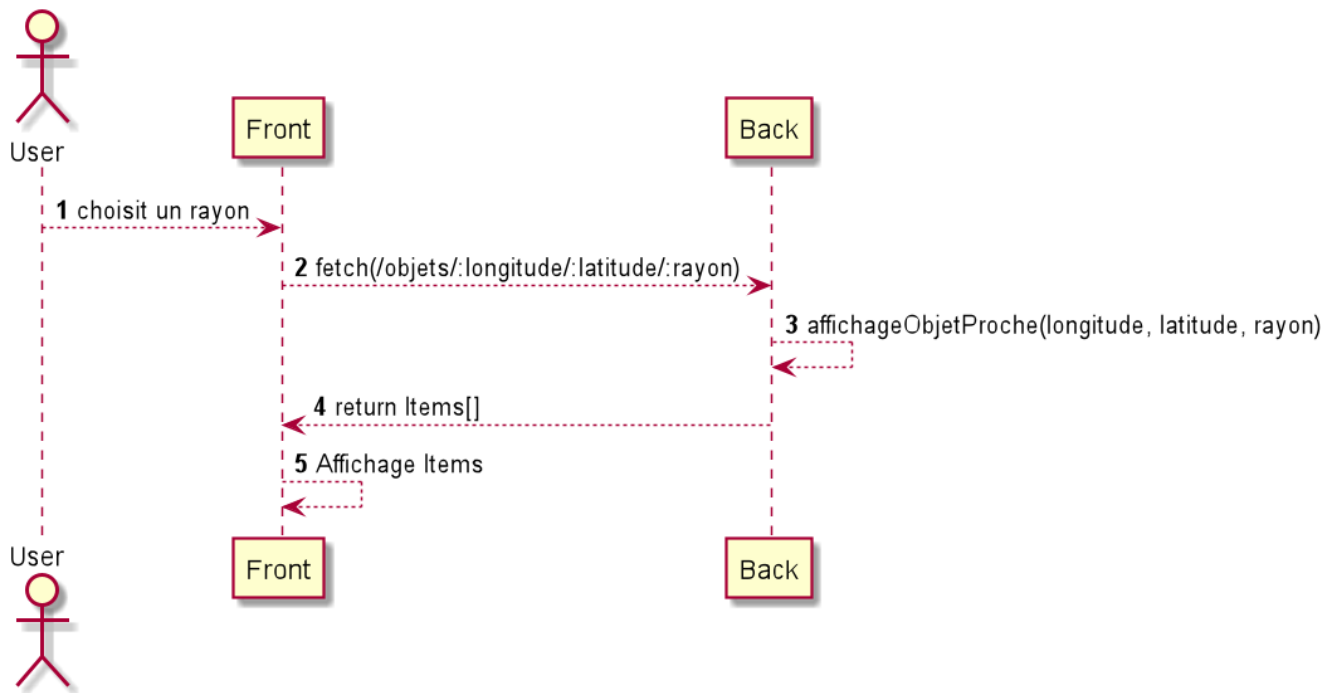
Feature "Declarer un Objet Perdu"



Feature "Rechercher un Objet Perdu"



Feature "Choisir le rayon des objets"



Feature "Consulter les suggestions d'objets perdus"

