



ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR
Membre de
HONORIS UNITED UNIVERSITIES

RAPPORT PROJET

FindIt - Plateforme de Recherche d'Objets Perdus

Ingénierie Informatique et Réseaux

Réalisé par :

LEMKHENTAR Abdelmoughith

EL AILY Lina

Encadré par :

Mr Iounousse

Remerciements

Nous tenons à exprimer notre gratitude à toutes les personnes ayant contribué à la réalisation de ce projet, notamment notre encadrant et toute personne ayant apporté son aide technique ou conceptuelle.

Résumé

FindIt est une application permettant aux utilisateurs de signaler des objets perdus ou trouvés et de faciliter leur récupération. Grâce à des technologies avancées telles que le traitement du langage naturel (NLP) et la reconnaissance d'images, elle optimise la correspondance entre objets déclarés perdus et retrouvés.

Table des matières

Remerciements.....	2
Résumé.....	3
Introduction.....	5
Chapitre 1 : Présentation du Projet.....	7
Chapitre 2 : Analyse et Conception UML.....	9
Chapitre 3 : Réalisation et Développement.....	10

Introduction

Dans un monde où les objets personnels sont de plus en plus nombreux et indispensables, la perte ou l'oubli de ces derniers constitue une situation fréquente et problématique. Chaque jour, des dizaines d'objets sont perdus dans les lieux publics ou privés sans possibilité de restitution efficace.

C'est dans ce contexte que notre projet **FindIt** a vu le jour. Il s'agit d'une application mobile collaborative dédiée à la recherche et à la déclaration d'objets perdus ou trouvés. Elle propose aux utilisateurs une plateforme simple, rapide et sécurisée pour publier des annonces, rechercher des objets ou contacter les potentiels propriétaires.

Ce projet scolaire, réalisé en binôme par **Lina El Aily** et **Abdelmoughith Lemkhentar**, a été encadré par notre professeur **Ionousse**. Il constitue une mise en pratique de nos acquis académiques et techniques. Il nous a permis d'aborder toutes les étapes de la gestion d'un projet informatique : analyse, conception UML, développement Flutter, intégration Firebase, tests et documentation.

Ce rapport retrace les différentes phases de notre projet et présente les choix techniques ainsi que les difficultés rencontrées au cours du développement.

Chapitre 1 : Présentation du Projet

1.1. Contexte

Dans le cadre de notre formation en Ingénierie Informatique et Réseaux, nous avons été amenés à réaliser un projet de fin de semestre ayant pour objectif la conception et le développement d'une application mobile innovante. Ce projet s'inscrit dans une démarche pédagogique visant à mettre en œuvre nos acquis en modélisation UML, programmation mobile, gestion de base de données et expérience utilisateur (UX/UI).

1.2. Objectif du Projet

Le projet consiste à développer une application mobile nommée FindIt, dont la mission est de faciliter la recherche et la déclaration d'objets perdus ou trouvés. Le système repose sur un principe communautaire où chaque utilisateur peut :

- Publier une annonce d'objet trouvé ou perdu
- Rechercher des annonces existantes à l'aide de filtres textuels ou d'images
- Contacter d'autres utilisateurs via une messagerie intégrée
- Recevoir des notifications en cas de correspondance

1.3. Problématique

Chaque jour, de nombreuses personnes perdent ou trouvent des objets dans des lieux publics sans pouvoir facilement retrouver ou restituer ces objets à leur propriétaire. Les réseaux sociaux peuvent être utilisés dans certains cas, mais ils ne sont pas adaptés à une recherche efficace et sécurisée.

FindIt vise à répondre à cette problématique en centralisant les annonces dans une interface conviviale, rapide et intuitive, avec des fonctionnalités avancées telles que la recherche intelligente et la messagerie sécurisée.

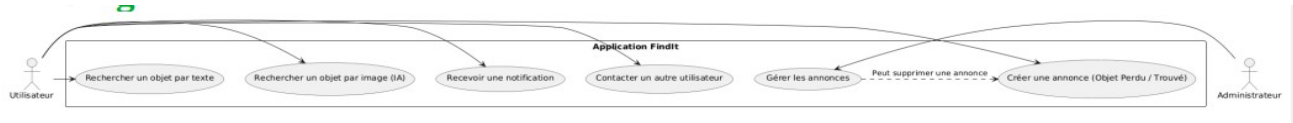
1.4. Outils et technologies utilisés

Pour la réalisation du projet, nous avons utilisé les outils et technologies suivants :

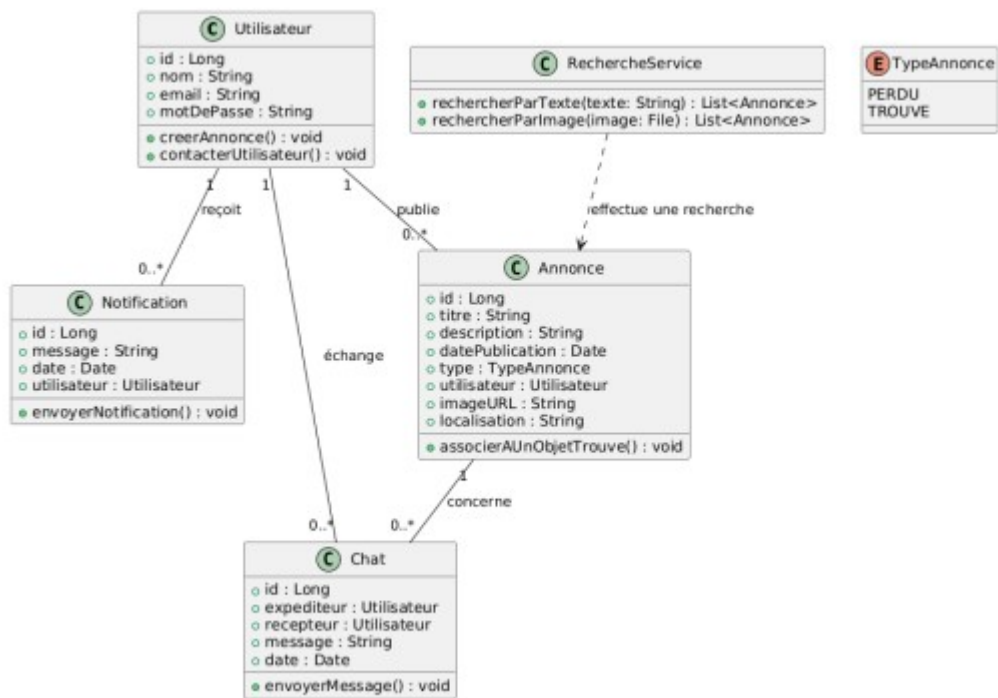
- Flutter : Développement mobile multiplateforme (Android/iOS)
- Firebase : Authentication, Firestore, Storage, Messaging
- Figma : Conception des maquettes de l'application
- UML (StarUML) : Modélisation des cas d'utilisation, des classes et des séquences

Chapitre 2 : Analyse et Conception UML

2.2. Diagramme de cas d'utilisation



2.2. Diagramme de classe



Chapitre 3 : Réalisation et Développement

3.1. Structure générale de l'application

L'application FindIt a été développée en Flutter. L'architecture est divisée en modules :

- Authentication
- Home
- Ajout d'annonce
- Recherche
- Messagerie
- Profil
- Notifications

3.2. Fonctionnalités principales

Recherche d'images par texte libre : l'utilisateur saisit une description naturelle (ex : "sac à dos noir avec fermeture cassée") et le système renvoie les images les plus pertinentes en comparant la description à celles en base (vectorisées par SBERT).

Recherche d'images par image : l'utilisateur fournit une image, et le modèle ResNet50 extrait un vecteur qui est comparé aux autres images de la base pour trouver les plus similaires visuellement.

Optimisation (pré-traitement au démarrage) : Les embeddings textuels et les vecteurs d'images sont pré-calculés et chargés en mémoire au démarrage de l'application, assurant des recherches ultra-rapides à l'exécution.

3.3. Interface utilisateur (UI/UX) :

Interface conçue sur Figma puis implémentée avec :

- Design minimaliste et intuitif
- Navigation fluide et responsive
- Widgets Flutter adaptés à chaque écran

3.4. Android :

- SDK :

compileSdk: 35

minSdk: 29

targetSdk: 35

- Versioning :

versionCode: 1

versionName: "1.0"

- Outils et Bibliothèques :

Navigation : navigation-fragment, navigation-ui pour la gestion de la navigation entre les écrans.

ViewBinding : activation de la gestion des vues simplifiée.

Lifecycle : utilisation de lifecycle-livedata et lifecycle-viewmodel pour la gestion des cycles de vie et des données asynchrones.

Retrofit : pour les requêtes réseau avec converter-gson pour JSON et logginginterceptor pour le suivi des requêtes HTTP.

Picasso : pour charger et afficher des images de manière simple et performante.

OkHttp : gestion avancée des requêtes HTTP.

Gson : sérialisation et désérialisation JSON.

SwipeRefreshLayout : pour implémenter des interfaces avec rafraîchissement par glissement.

GIF : prise en charge des GIFs avec android-gif-drawable.

3.5. Technologies Back-end utilisées :

Framework principal :

- Spring Boot 3.4.3 – pour le développement rapide d'applications Java basées sur le modèle Spring.

Sous-technologies intégrées :

- Spring Web : pour exposer des services RESTful (API).
- Spring Data JPA : pour la gestion de la persistance et les opérations CRUD via Hibernate.
- Spring Security : pour la gestion de l'authentification et des autorisations.
- JWT (JSON Web Token) avec jjwt et spring-security-jwt : pour l'implémentation de l'authentification sécurisée via tokens.

- Spring Boot Starter Validation : pour valider automatiquement les données reçues via les contrôleurs.
- Spring Boot Starter Actuator : pour la surveillance et la gestion de l'application (métriques, endpoints système, etc.).
- Spring GraphQL avec graphql-java-tools : pour la mise en place d'API GraphQL.
- Firebase Admin SDK : pour la gestion du stockage d'images et l'interaction avec Firebase.
- Lombok : pour générer automatiquement le code boilerplate (constructeurs, getters/setters...).
- Thumbnailator : pour la génération de miniatures d'images.

3.6. Technologies IA & FastAPI utilisées :

Framework principal : FastAPI pour développer une API performante, asynchrone et rapide en Python.

Bibliothèques d'intelligence artificielle : PyTorch / torchvision : pour l'utilisation du modèle ResNet50 pré-entraîné dans l'analyse visuelle des images.

SentenceTransformers (SBERT) : pour générer des vecteurs sémantiques à partir de textes libres saisis par l'utilisateur (modèle : paraphrase-MiniLM-L6-v2).

scipy.spatial.distance (cdist) : pour le calcul de la similarité cosine entre les vecteurs (textes et images).

NumPy : pour le traitement et la manipulation des vecteurs numériques.

Pillow (PIL) : pour le chargement et le traitement d'images à partir de leurs URLs.

3.7. Base de données

MySQL – utilisée pour stocker les annonces (titres, descriptions) et les URLs des images, accessibles via le connecteur mysql-connector-python.

Tests : • Spring Boot Starter Test et spring-security-test : pour les tests unitaires et les tests d'intégration avec support de sécurité.

3.8. Tests

Spring Boot Starter Test et spring-security-test : pour les tests unitaires et les tests d'intégration avec support de sécurité.

3.9. Authentification et sécurité

- Firebase Authentication (email et mot de passe)
- Vérification d'email et protection des données

3.10. Stockage et services cloud :

- Firebase Storage : utilisé pour stocker et gérer les images et autres fichiers multimédias de manière sécurisée dans le cloud. Firebase offre une solution simple et rapide pour stocker des fichiers volumineux, avec une intégration fluide dans les applications Android et web.

3.11. Outils de versionnement :

- Git : utilisé pour la gestion du code source et la collaboration via des plateformes comme GitHub ou GitLab.
- GitHub : pour la gestion des versions et la collaboration avec les équipes.

3.12. Outils de gestion de projet :

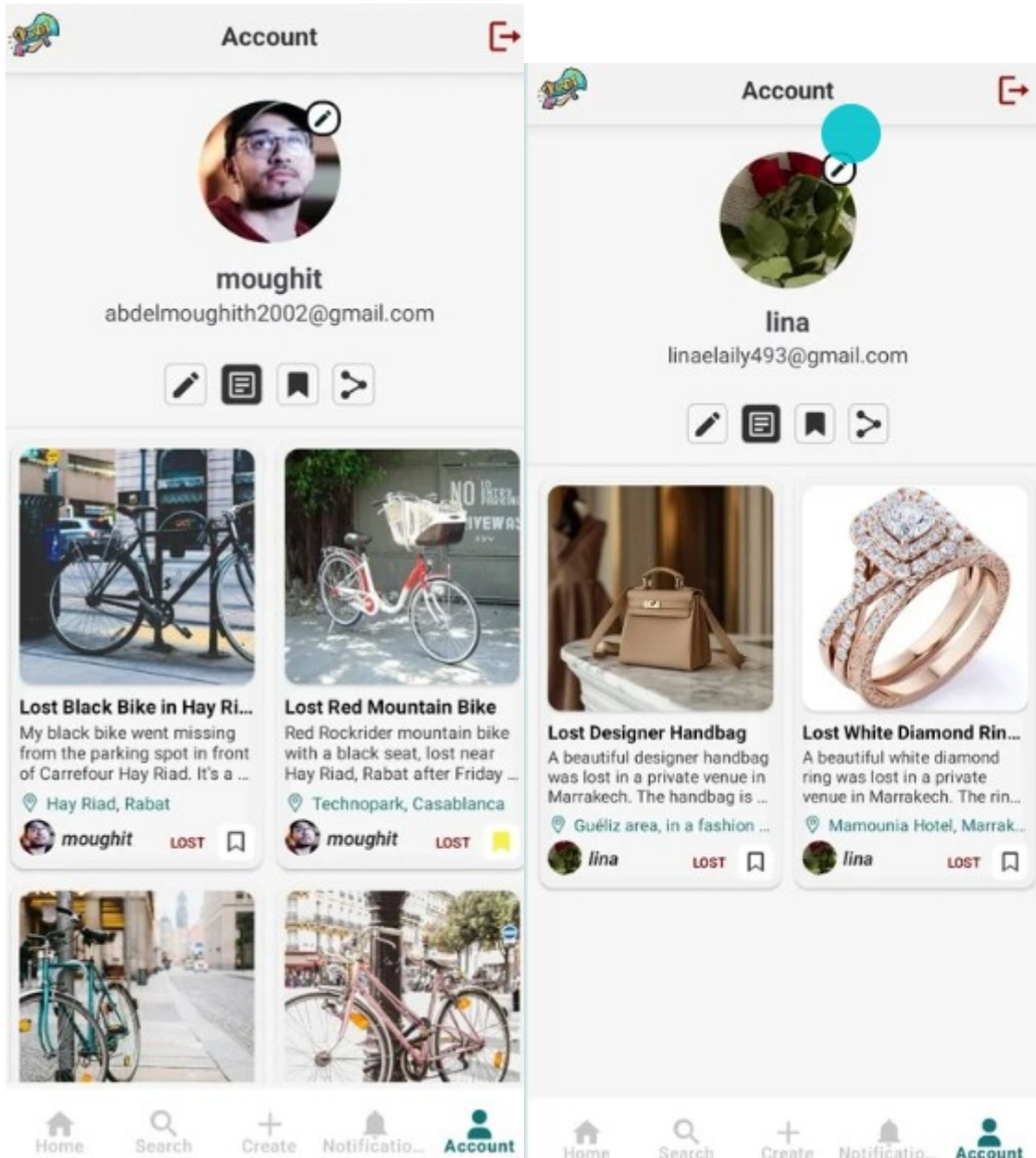
- Maven : utilisé pour la gestion des dépendances et la compilation du projet Spring Boot.
- Gradle : utilisé pour la gestion du projet Android, avec les dépendances et la configuration des tâches de build.

3.13. Difficultés rencontrées

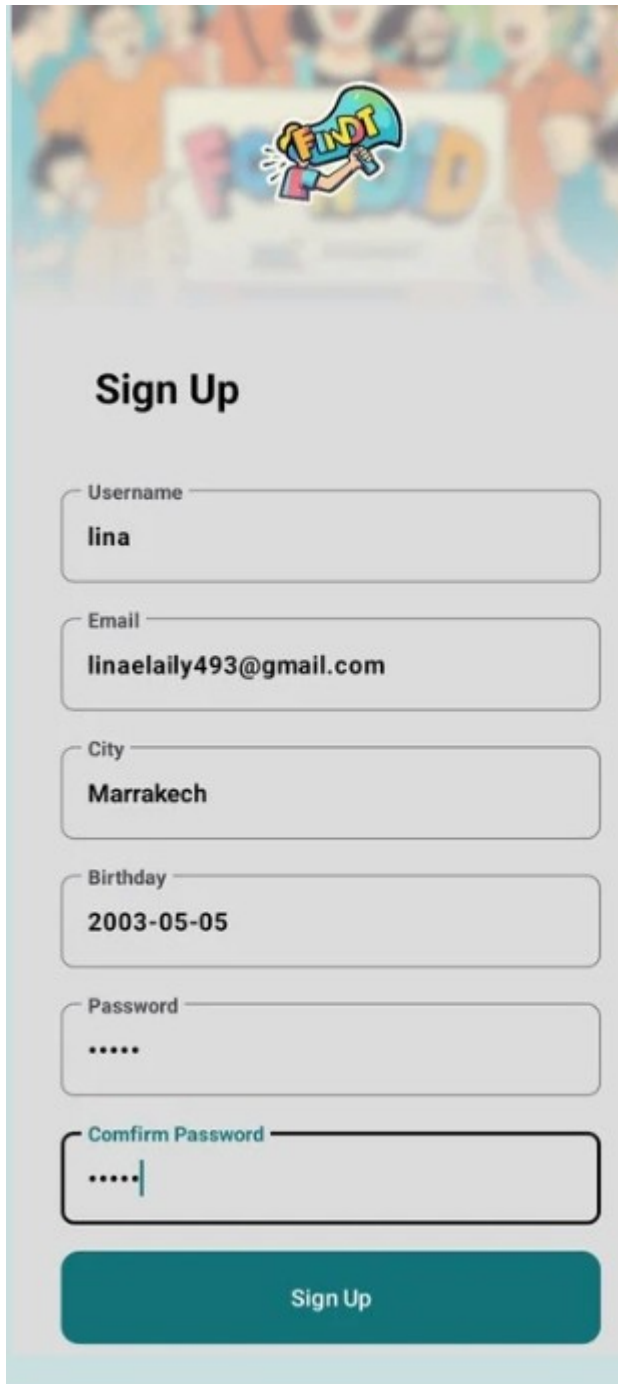
- Logique asynchrone en Flutter
- Recherches complexes avec Firestore
- Implémentation de la messagerie temps réel
- Optimisation des images et chargement rapide

4. Aperçu de l'Application

Compte :



Sign up :



The form is titled "Sign Up" and is set against a light gray background. At the top, there is a banner with a cartoon illustration of people and a speech bubble that says "FIND FOOD". Below the banner, the form contains several input fields with labels to their left: "Username" (filled with "lina"), "Email" (filled with "linaelaily493@gmail.com"), "City" (filled with "Marrakech"), "Birthday" (filled with "2003-05-05"), "Password" (filled with five dots), and "Confirm Password" (filled with five dots and a cursor). At the bottom of the form is a teal "Sign Up" button.

Sign Up

Username
lina

Email
linaelaily493@gmail.com

City
Marrakech

Birthday
2003-05-05

Password
.....

Confirm Password
.....|

Sign Up

Login :



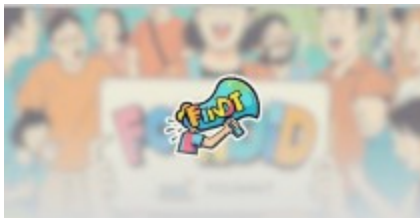
Email

abdelmoughith2002@gmail.com

Password

LOGIN

Créer une annonce ;



CREATE

Title

City

Location

Status

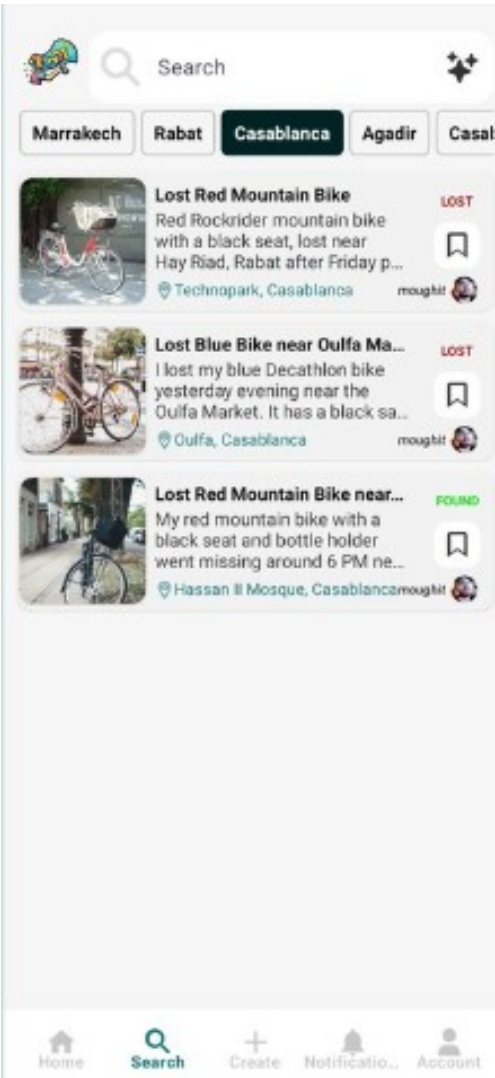
Description

Add Images

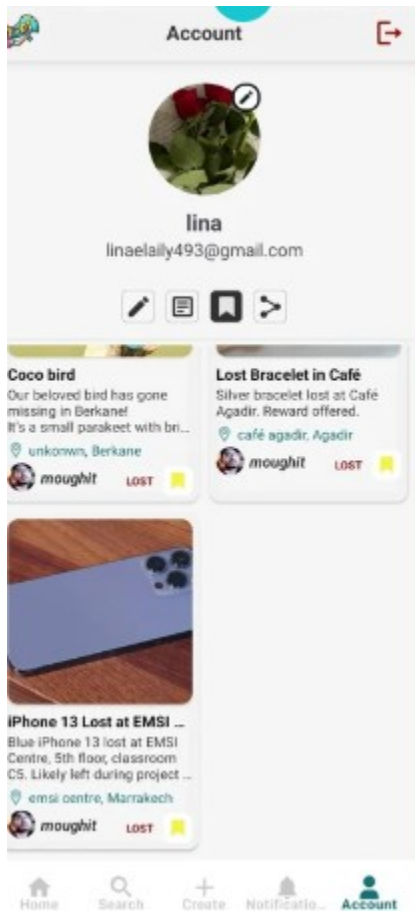
+

CREATE

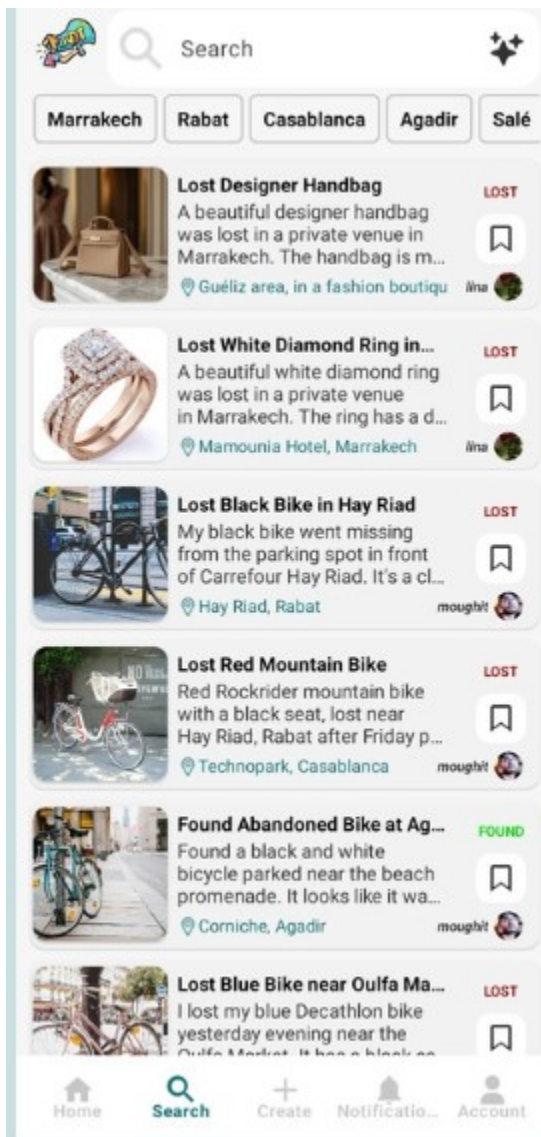
Recherche(par ville) :



Annonces enregistrées :



Page d'annonces et de recherche :



5. Recherche

Modèles de deep learning utilisés

1. ResNet-50

Recherche: ResNet-50 torchvision model

+ Variante utilisee: torchvision.models.resnet50

Paper d'origine : Deep Residual Learning for Image Recognition, Kaiming He et al. (2015)

Recherche Google: ResNet-50 paper site:arxiv.org

2. Sentence-BERT (SBERT) - paraphrase-MiniLM-L6-v2

. Recherche: paraphrase-MiniLM-L6-v2 Sentence-BERT model

· I Paper d'origine : Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, Reimers & Gurevych (2019)

Recherche Google: sentence-transformers paraphrase-MiniLM-L6-VZ site:arxiv.org

6. Conclusion

Le projet **FindIt** s'inscrit dans une volonté d'apporter une réponse concrète à un problème de société : la perte et la récupération d'objets dans les lieux publics. Grâce à une combinaison de technologies modernes telles que le **traitement du langage naturel**, la **reconnaissance d'images**, et des fonctionnalités interactives comme la **carte géolocalisée** ou les **notifications en temps réel**, FindIt offre une expérience utilisateur fluide, intuitive et efficace.

Au cours de ce projet, nous avons pu mobiliser des compétences variées en **développement web**, **intelligence artificielle**, **sécurité informatique** et **expérience utilisateur**, tout en respectant un cahier des charges précis. Le résultat est une plateforme fonctionnelle, évolutive et adaptée aux besoins d'un large public.

Ce projet ouvre également la voie à de nombreuses perspectives d'évolution, tant sur le plan technique que fonctionnel, avec l'ambition de faire de FindIt un outil de référence dans le domaine de la gestion des objets perdus.