**MSPR**

**Codes promotionnels - WESH**

**Sommaire**

Table des matières

[I. INTRODUCTION 2](#_Toc31640067)

[A. Contexte général 2](#_Toc31640068)

[B. Analyse du besoin 2](#_Toc31640069)

[C. Découpage du projet 2](#_Toc31640070)

[II. BENCHMARK TECHNOLOGIE MOBILE 2](#_Toc31640071)

[A. Langages natifs, React et Flutter 2](#_Toc31640072)

[III. SPÉCIFICATION FONCTIONNELLE 3](#_Toc31640073)

[A. Application mobile 3](#_Toc31640074)

[B. API 3](#_Toc31640075)

[IV. SPÉCIFICATION TECHNIQUE 3](#_Toc31640076)

[A. Application mobile 3](#_Toc31640077)

[1. Choix technique 3](#_Toc31640078)

[2. Implémentation 3](#_Toc31640079)

[B. API 3](#_Toc31640080)

[1. Choix technique 3](#_Toc31640081)

[2. Implémentation 4](#_Toc31640082)

[V. Procédure d’installation 4](#_Toc31640083)

[A. Démarrage serveur API 4](#_Toc31640084)

[B. Installation application mobile 5](#_Toc31640085)

[VI. Procédure de test 5](#_Toc31640086)

[VII. Lancement de l’application 6](#_Toc31640087)

# INTRODUCTION

## Contexte général

GoStyle est une boutique en ligne de vêtements streetwear. Le site en Prestashop gère le catalogue client et permet la réalisation des achats en ligne.

## Analyse du besoin

La société GoStyle souhaite faire développer une application mobile pour mettre en place un système de couponing.

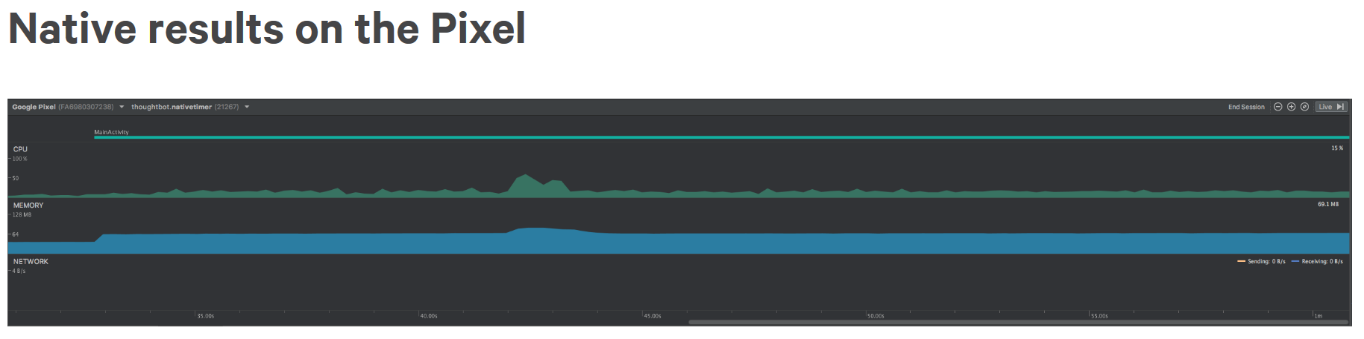
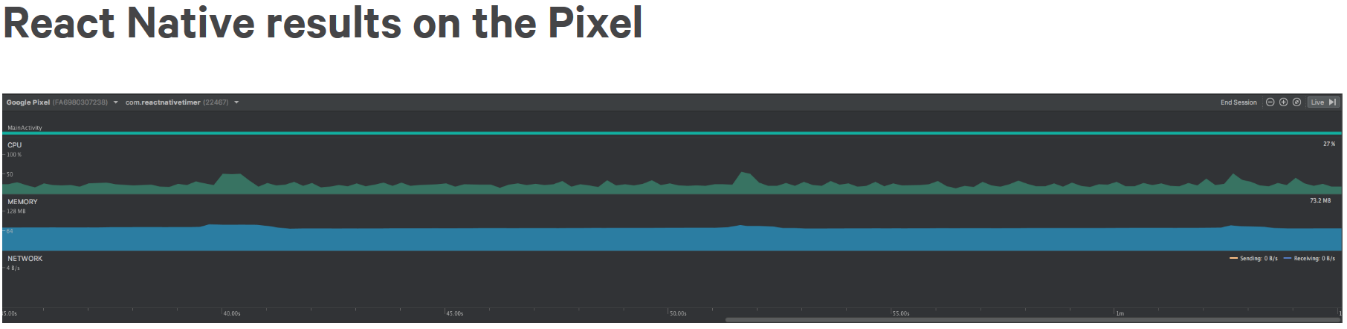
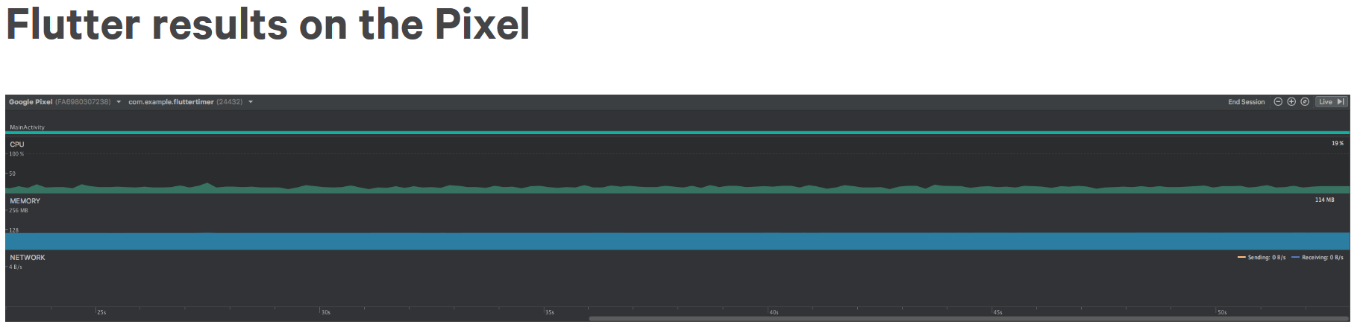
L’application aura besoin de scanner des QR code visible sur des panneaux publicitaire, de récupérer les données associées au QR code c’est-à-dire sa description, sa période de validité ainsi que la promotion que le code apporte.

## Découpage du projet

Nous sommes un groupe de 3, Sylvain Dupont est chargé du développement de l’API et tests, Quentin Métivier du développement mobile. Ludovic Luscap s’occupe de la documentation du projet, architecture, conception et aide au développement mobile.

# BENCHMARK TECHNOLOGIE MOBILE

## Langages natifs, React et Flutter

Nous avons effectué un benchmark des technologies mobile en nous basant sur l’étude [suivante](https://thoughtbot.com/blog/examining-performance-differences-between-native-flutter-and-react-native-mobile-development).

Ceci représente des tests de performances de technologies sur une application de timer réalisée sur chaque langage de programmation et testé sur des environnements identiques. On peut voir que la charge CPU est nettement plus faible sur les langages natifs que sur les Framework. Cependant on notera que sur Flutter elle est plus faible que sur react.

Nous avons choisi de prendre la technologie Flutter pour expérimenter ce nouveau Framework. Comme nous voulions utiliser une technologie pratique mais aussi efficace pour obtenir une application performante et agréable à utiliser.

# SPÉCIFICATION FONCTIONNELLE

## Application mobile

* Une fonctionnalité de scan de QR code
* Affichage des promotions en cours
* Affichage de l’historique des scans effectués
* Authentification des utilisateurs pour code promo privé

## API

- Stock les codes QR

- Historique de toutes les requêtes par utilisateur

- Fonctionnalité REST pour l’application

Application Mobile

Scan QR code

Afficher code promo en cours

Afficher historique des codes scannés

API rest

Connexion

<<Include>>

<<Include>>

<<Include>>

Utilisateur

# SPÉCIFICATION TECHNIQUE

## Application mobile

### Choix technique

Nous avons choisi le Framework de Google « Flutter » pour développer notre application mobile. Nous nous sommes basés sur l’étude du chapitre 2.1 pour faire notre choix.

### Implémentation

Pour le QR code nous utilisons une librairie externe en plus de flutter nommée « barcode\_scan ». Pour effectuer la demande d’accéder à l’appareil photo on utilise le « androidManifest » et « info.plist » qui font la demande d’accès.

Les méthodes de flutter permettent de récupérer les données venant d’une API pour afficher les promotions en cours.

## API

### Choix technique

Nous avons choisi le framework python Django avec son package django-REST-framework. La base de données est la base intégrée de django en SQLite.

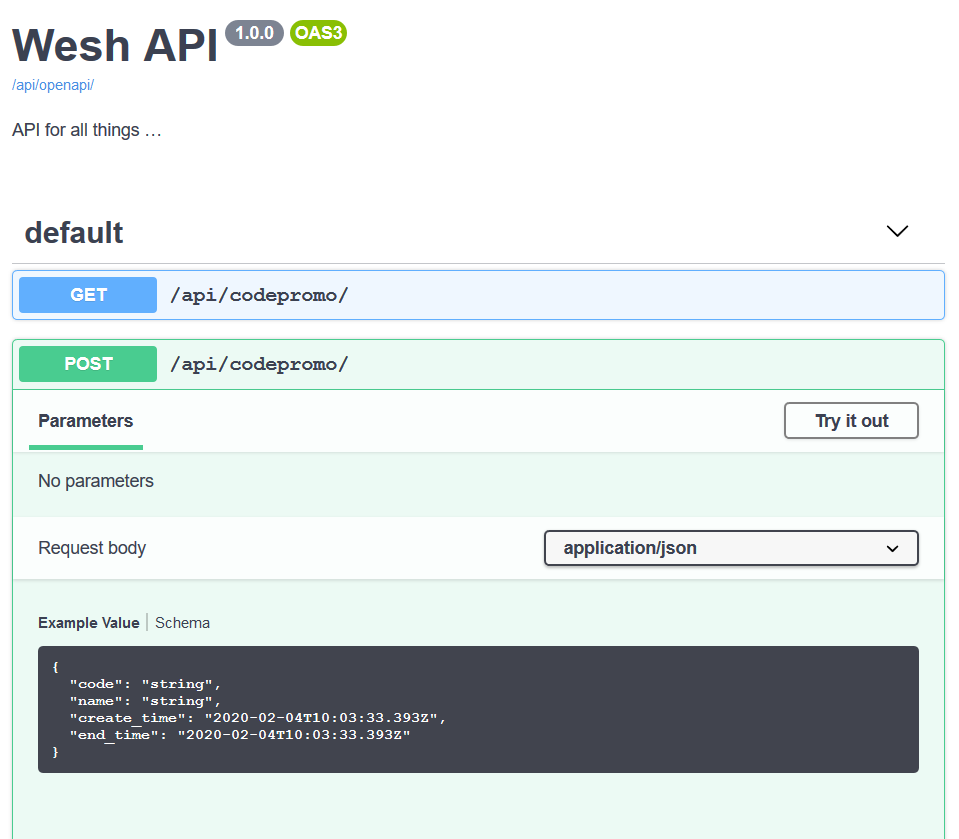
### Implémentation

On s’authentifie via token JSON via l’url <mainURL>/api/token-auth

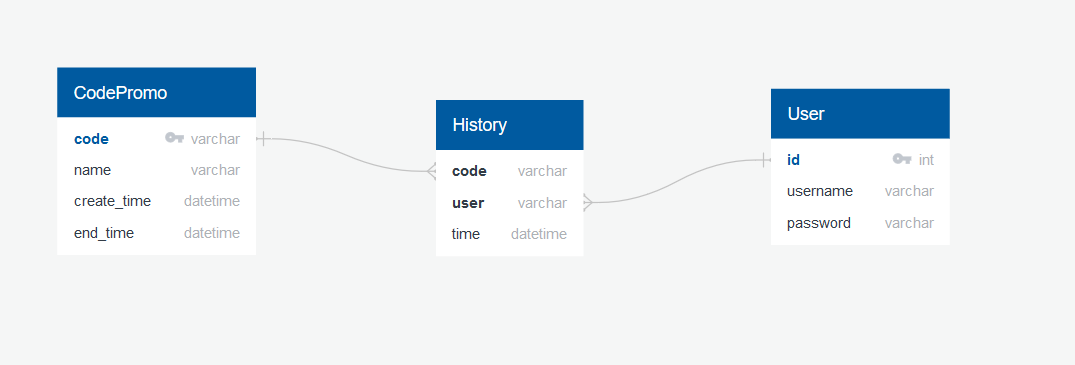
A chaque requête le token est passé dans le header.

Un swagger a été ajouté pour mettre en librairie les requêtes possibles de l’API.

[<mainURL>/api/swagger-ui/](http://127.0.0.1:8000/api/swagger-ui/)



L’API va stocker en base les données des codes promo et renvoyés à l’utilisateur sur l’application mobile.



De plus elle renverra les promotions en cours sur l’accueil de l’application.

Il garde en mémoire les derniers codes scannés par l’utilisateur.

# Procédure d’installation

## Démarrage serveur API

Prérequis : installer la version 3.7 de python.

Récupérer le projet de l’API sur github avec la commande suivante :

* git clone <https://github.com/Syldup/Wesh_API.git>

Une fois cette commande lancée, exécuter la commande pip :

* pip3 install -r requi.txt

Démarrer le serveur de l’API

* python3.7 manage.py runserver

## Installation application mobile

Prérequis : avoir un smartphone android ou iOS

Récupérer le projet d’application mobile sur github avec la commande suivante :

* git clone <https://github.com/LuscapLudovic/wesh.git>

Récupérer le fichier .apk dans le dossier wesh/prod/app-release.apk

Installer l’application

Lancer l’application comme login/password : user1 / Aqwzsx123

# Procédure de test

Nous avons utilisé les fonctionnalités de tests fourni par le framework [flutter](https://flutter.dev/docs/testing).



# Lancement de l’application

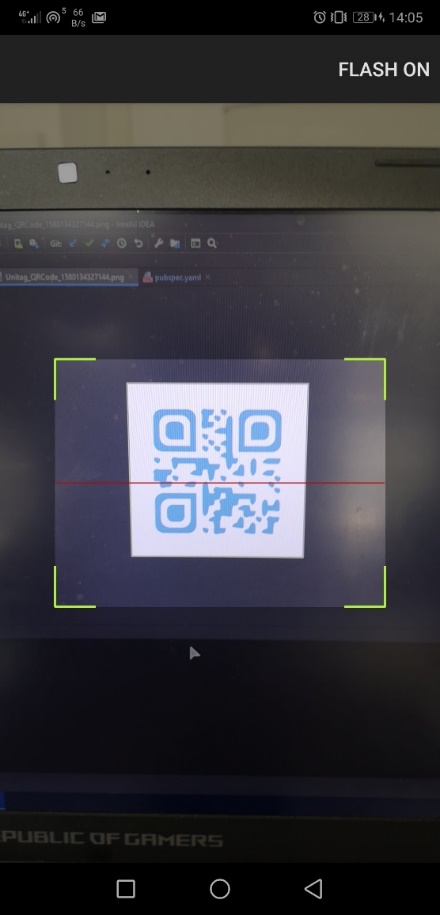


←

La page d’accueil de l’application montre les promotions en cours.

→

Accéder à l’historique des QR codes



Appuyer sur l’icône  pour effectuer un scan de QR code.