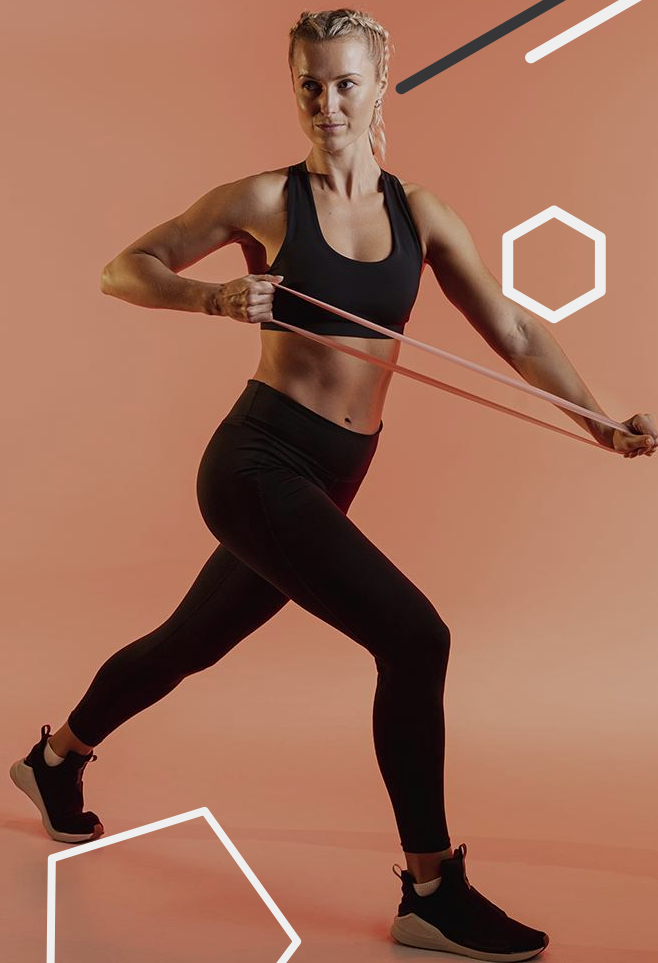




Crossfit Lov

Love in the box



SOMMAIRE



01

L'ARCHITECTURE

Vue d'ensemble sur
l'architecture du projet

02

LA DEMARCHE

Notre méthodologie

03

LES BRIQUES

Présentation de chaque
micro-service

04

LE FRONT

Présentation des interfaces

INTRODUCTION

Une application de rencontres conçue spécialement pour les adeptes du Crossfit. Son objectif est de mettre en relation des personnes pratiquant le crossfit dans la même "box" (appellation pour désigner le lieu d'exercice du crossfit).

Nous avons suivi le modèle des applications de rencontres classiques, tel que Tinder.

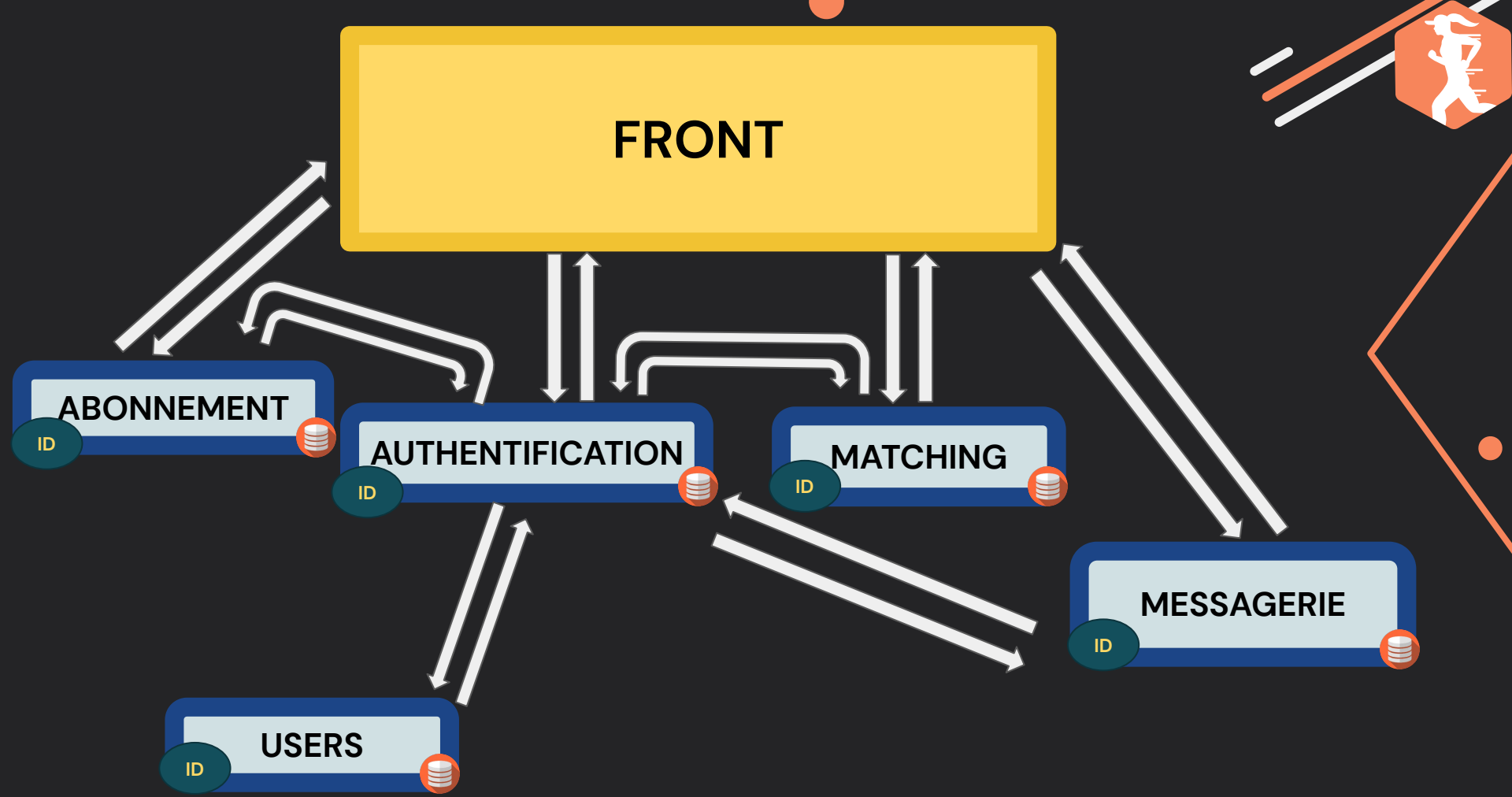




01

L'ARCHITECTURE

Vue d'ensemble sur l'architecture
du projet





02

LA DEMARCHE

Notre méthodologie de travail

PROBLEMATIQUE



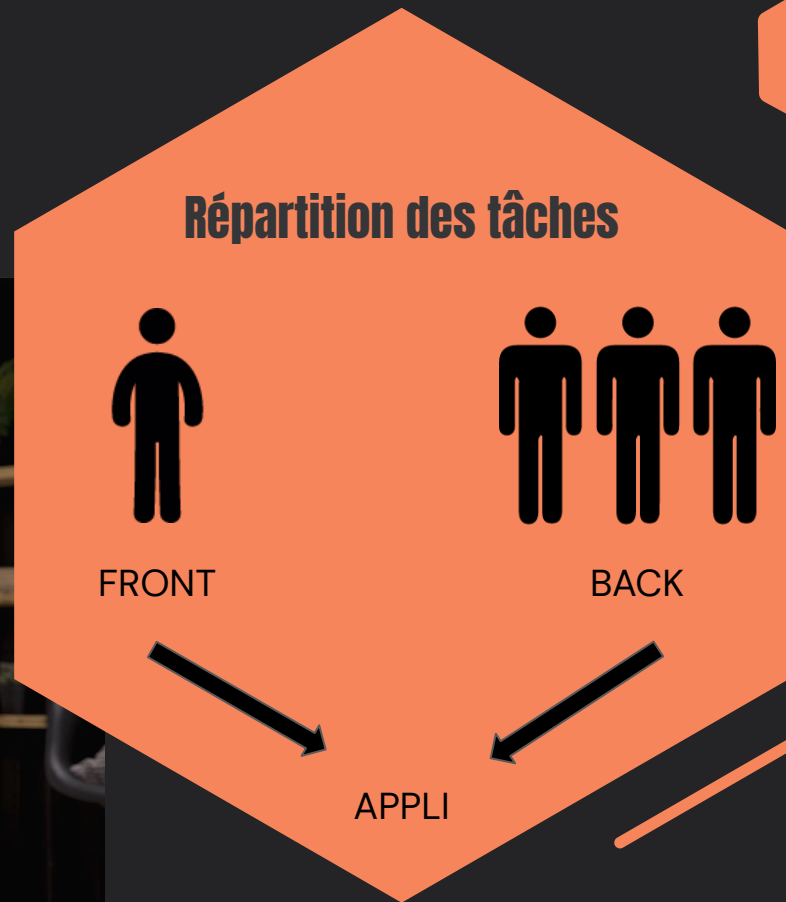
UN TEMPS IMPARTI

Afin d'exploiter au maximum le temps imparti, nous nous sommes fixés des paliers à atteindre par semaine:

- 2 semaines de rush par micro-service
- 2 semaine de rush pour l'interface correspondante (les deux semaines en parallèle)



SOLUTION





03

LES BRIQUES

Présentation de chaque
micro-service

LES SERVICES



AUTHENTIFICATION

Gère l'authentification de l'utilisateur, l'inscription et vérifie si il est toujours connecté



USERS

CRUD* des infos utilisateurs assez classique (cf. l'exemple de Netflix dans le cours)

MESSAGERIE

Permet de communiquer entre différents utilisateurs, une fois le matching passé



MATCHING

S'occupe du coeur de l'appli: le matching entre utilisateurs



ABONNEMENT

Une option d'abonnement qui permet à l'utilisateur de s'octroyer des droits supplémentaires

**CRUD: Create, Read, Update, Delete
Les 4 opérations de base de la persistance des données*

LA BRIQUE AUTHENTICATION



Le DETAIL

- Génération d'un **TOKEN** au login
- TOKEN renouvelé toutes les 5 minutes grâce à la route **REFRESH**
- A chaque appel du front envers une API, cette dernière check si le TOKEN est toujours valide
- A la déconnexion: suppression du TOKEN
- Utilisation de **basicauth**: admin admin123
- Encodage en **JSON**
- **API REST**
- Technologies utilisées: **MySQL, MariaDB, Golang**





- **CHECK:** Si l'utilisateur est déconnecté (= TOKEN invalide), renvoie *"status unacceptable"*

- **SIGN UP:** gère 3 cas de figures

→ Si l'inscription s'est bien passée: renvoie *"200"*

→ Si l'utilisateur est déjà existant: renvoie *"406 not acceptable"*

→ Si le Front renvoie un mauvais basicauth: renvoie *"status unauthorized"*

- **REFRESH:** gère 3 cas de figures

→ Si l'utilisateur est toujours connecté: renvoie *"200"* + génère un nouveau TOKEN

→ Si le Front renvoie un mauvais basicauth: renvoie *"status unauthorized"*

→ Si l'utilisateur est déconnecté (= TOKEN invalide): renvoie *"status unacceptable"*

- **SIGN IN:** gère 3 cas de figures

→ Si l'utilisateur est connecté (avec TOKEN): renvoie *"200"*

→ Si l'utilisateur rentre un mauvais login: renvoie *"406 not acceptable"*

→ Si le Front renvoie un mauvais basicauth: renvoie *"status unauthorized"*



LA BRIQUE USERS



Le DETAIL

- Application du **CRUD** classique des infos utilisateurs
- Technologies utilisées: **MySQL, MariaDB, Golang**
- **API REST**
- Utilisation de **basicauth**: admin admin123
- Format des flux en **JSON**



Les ROUTES

- **DELETE** users: se charge de supprimer des utilisateurs
- **PUT** users: se charge de l'update des utilisateurs
- **POST** users: se charge de créer de nouveaux utilisateurs
- **GET** users: se charge de récupérer des données d'utilisateurs

LA BRIQUE MESSAGERIE

Le DETAIL

- Se charge de la partie interaction entre les matchs
- Technologies utilisées: **MySQL, MariaDB, Python, Flask, POSTMAN**
- **API REST**
- Utilisation de **basicauth**: admin admin123
- Format des flux en **JSON**



Les ROUTES

- **PUT** messages: se charge de l'update des messages
- **POST** messages: se charge de créer de nouveaux messages



LA BRIQUE MATCHING

Le DETAIL

- Se charge de bien mettre en relation deux personnes ayant matchées
- Technologies utilisées: **MySQL, MariaDB, Python, Flask, POSTMAN**
- **API REST**
- Utilisation de **basicauth**: admin admin123
- Format des flux en **JSON**



Les ROUTES

- **PUT** matches: se charge de l'update des matchs
- **POST** matchs: se charge de la création de nouveaux matchs



LA BRIQUE ABONNEMENT



Le DETAIL

- Se charge de placer le statut "d'abonné" à un utilisateur
- Technologies utilisées: **MySQL, MariaDB, Python, Flask, POSTMAN**
- **API REST**
- Utilisation de **basicauth**: admin admin123
- Format des flux en **JSON**



Les ROUTES

- **PUT** abonnement: se charge de l'update des utilisateurs avec ou sans abonnement
- **POST** abonnement: se charge d'ajouter de nouveaux utilisateurs abonnés



04

LE FRONT

Présentation des interfaces



Bienvenue sur CrossfitLove

INSCRIPTION

CONNEXION

PAGE DE BIENVENUE

Au moment de la connexion sur l'application, il y a possibilité de **créer un nouvel utilisateur**, ou bien d'**utiliser un compte existant**

Appel à **AUTHENTIFICATION** ...

... mais aussi à **USERS**

A white rounded rectangular form on a dark background. At the top, the word 'Connexion' is centered. Below it are two input fields: 'Email' and 'Mot de passe'. At the bottom is a red button with the text 'SE CONNECTER' in white.

Connexion

Email

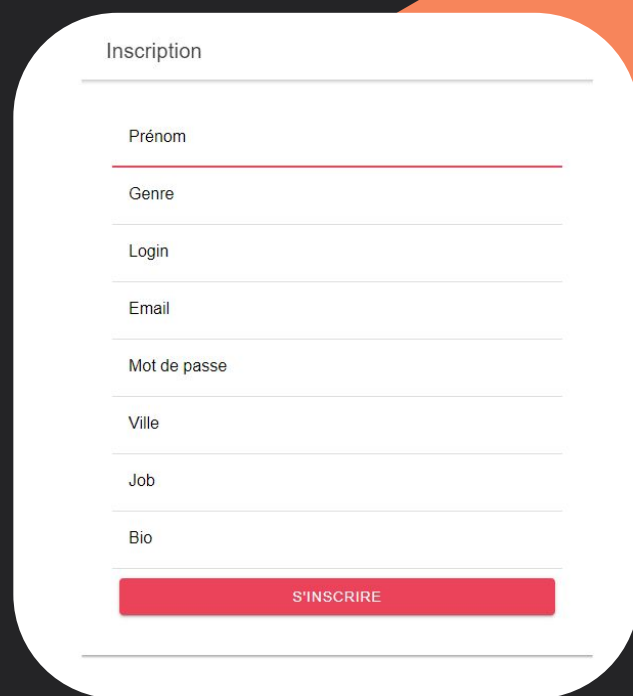
Mot de passe

SE CONNECTER

la **CONNEXION**

L'utilisateur renseigne ses ID, appel à **USERS**
& **AUTHENTIFICATION**

VS.

A white rounded rectangular form on a dark background. At the top, the word 'Inscription' is centered. Below it are seven input fields: 'Prénom', 'Genre', 'Login', 'Email', 'Mot de passe', 'Ville', 'Job', and 'Bio'. At the bottom is a red button with the text 'S'INSCRIRE' in white.

Inscription

Prénom

Genre

Login

Email

Mot de passe

Ville

Job

Bio

S'INSCRIRE

L' **INSCRIPTION**

Le nouvel utilisateur renseigne ses
données personnelles, appel à **USERS**

LE
MATCHING



ET L'ABONNEMENT
(histoire de pouvoir subscribe à tout moment, pour plus de match et plus de fun)

LA
MESSAGERIE
(Accessible depuis la page des swipes)

Matches



Alexis

Listen, I've had a pretty messed up day...



Megan

Listen, I've had a pretty messed up day...



Scarlett

Listen, I've had a pretty messed up day...



LA MESSAGERIE

(Le Listing de tous les matchs avec possibilité d'accès
à la conversation privée)

← MATCHS

Alexis Ren
En ligne



Listen, I've had a pretty messed up day ... 21:16

It's over baby 21:17

LA MESSAGERIE

(Partie 2)

Votre message

ENVOYER

LE DETAIL EN BDD



MESSAGES

User
id
d34929
d66669
d12345
d35748

Room
id
1
2
3
4

Messages				
id	user id	room id	message	dates
1001	d34929	1	Go MH ce soir ?	01/01/2020 14h03
1002	d6669	1	Chaud	01/01/2020 14h40
1003	d12345	2	Kévin tu fais quoi ?	01/01/2020 14h44
1004	d12345	2	Je suis au BBQ	01/01/2020 14h46
1005	d35748	2	Ahhh...	01/01/2020 14h48
1006	d34929	1	Parfait!	01/01/2020 14h52

ABONNEMENTS

id	y/n
d34929	1
d66669	0
d12345	1
d35748	1

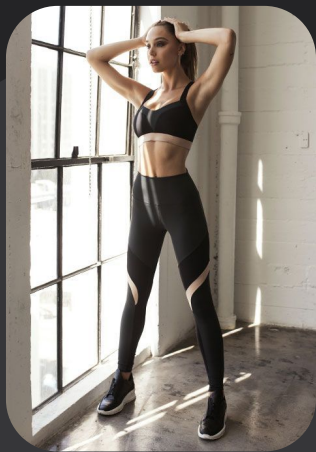
SWIPES

SWIPE_ID	usr_id	swipe_with
1	Hugo	Megan Fox
2	Hugo	Alexis Ren
3	Alexis Ren	Hugo
4	MGK	Megan Fox
5	Megann Fox	MGK

MATCHS

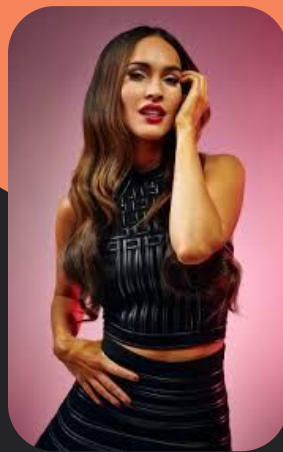
match_id	usr_id_1	usr_id_2
1	Alexis Ren	Hugo
2	Megan Fox	MGK

LES RETOURS



ALEXIS REN

“Grâce à **CrossfitLov**, j’ai pu rencontrer Hugo, l’amour de ma vie. Les créateurs de CrossfitLov ont donné un nouveau sens à ma vie.”



MEGAN FOX

“Après ma rupture avec MGK, j’ai pu me consoler grâce à cette application et aux génies qui l’ont produit. Ils méritent vraiment une bonne note.”



MERCI !

D'être arrivé jusque là ...

Maintenant, il ne reste plus qu'à écouter Megan Fox !
;)

(Et merci pour cette unité, c'était vraiment cool)

