Rapport BadTracker

Jean-Michel REMEUR – Thomas LACAZE

M1 LSI APP Groupe A



Table des matières

[1- Outils 2](#_Toc73957931)

[2- Liste des fonctionnalités présentes 2](#_Toc73957932)

[3- Liste des fonctionnalités non présentes 3](#_Toc73957933)

[4- Les améliorations possibles 3](#_Toc73957934)

[5- Architecture générale 4](#_Toc73957935)

[a. Architecture de l’application mobile 4](#_Toc73957936)

[b. Architecture d’infrastructure 4](#_Toc73957937)

[6- Captures d’écrans 5](#_Toc73957938)

[a. Affichage match 5](#_Toc73957939)

[b. Création match 8](#_Toc73957940)

[c. Rotation écran 9](#_Toc73957941)

[d. Changement de langue 12](#_Toc73957942)

# Outils

* Retrofit
* Navigation Host
* SQLite

# Liste des fonctionnalités présentes

* Traduction de toutes les interfaces en anglais, français
  + Automatiquement en fonction de la langue du téléphone
* Adaptation des interfaces en mode portrait et paysage automatiquement
* Localisation des matchs en fonction d’un picker sur une carte
  + Gestion des permissions
  + Lecture des coordonnées pour afficher l’adresse
  + Localisation en temps réel (possibilité de forcer l’actualisation)
* Possibilité de prendre une photo lors de la création d’un match (stockage uniquement en local)
* Validateur sur la création du match empêchant la saisie d’information vide ou invalide
* Sauvegarde des entrées entre les différents fragments lors de la création d’un match. Ainsi nous pouvons revenir en arrière pour corriger une information et garder les informations déjà renseignées.
* Sauvegarde en local sur une base de données SQLite des 5 derniers matchs
* Sauvegarde à distance via une API Rest des matchs les plus anciens
* Affichage sur une carte des matchs présents dans la base de données locale et à distance
* Affichage des détails d’un match sur un fragment indépendant

# Liste des fonctionnalités non présentes

* RAS

# Les améliorations possibles

* Changement manuelle de la langue dans les paramètres de l’application
* L’envoi de l’image sur la base de données distante
* Utilisation de l’architecture MVVM (Modèle vue, vue modèle)
* UX plus facile d’utilisation et plus intuitive
* UI plus agréable, meilleur thème de couleurs

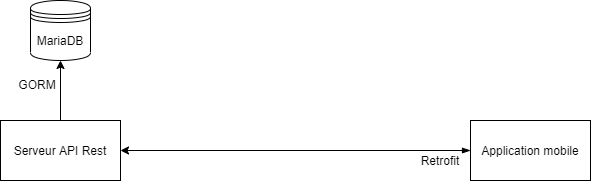
# Architecture générale

# Architecture de l’application mobile

# Architecture d’infrastructure

# Base de données distante

Nous avons utilisé une API Rest afin de faire la connexion entre une base de données externe et l’application mobile. Voici un schéma expliquant le fonctionnement



Le serveur API Rest est fait en Golang voici le [dépôt](https://github.com/LacazeThomas/badTracker). Nous avons utilisé une CI/CD qui permet de normalement lancer les tests unitaires et de lancer la création d’une image docker afin de pouvoir l’utiliser plus tard.

Nous avons un serveur permettant avec Traefik (reverse proxy écrit en Go) d’utiliser cette image docker avec un sous-domaine : badtracker.thomaslacaze.fr. Ainsi nous donnons ce sous-domaine à Retrofit afin qu’il puisse faire les appels souhaités à l’API.

# Captures d’écrans

## Affichage match

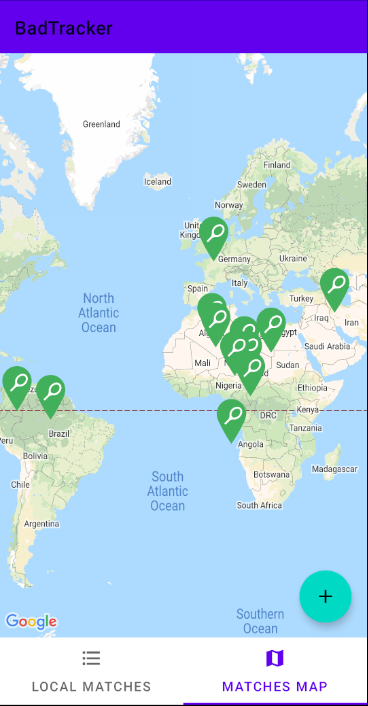


Figure 1 - Affichage des matchs à distance sur l'API

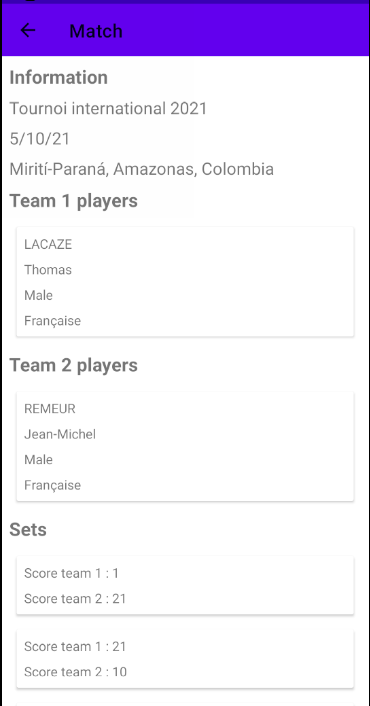
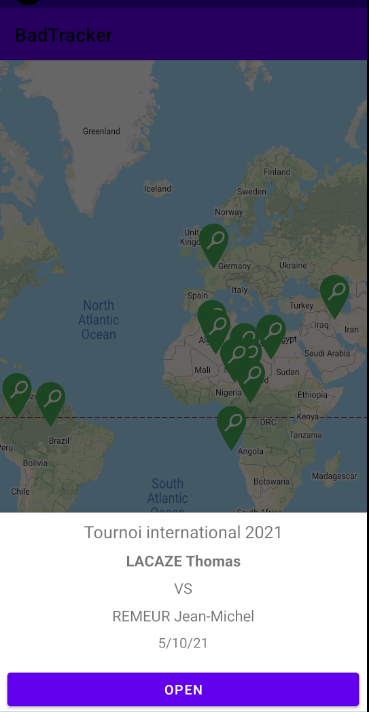
 

Figure 2 – Affichage des détails d’un match

Figure 3 - Affichage d'un fragment lors d'un clique sur un match

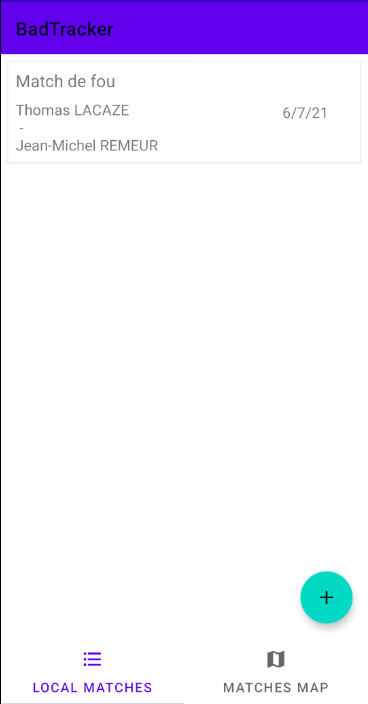


Figure 4 - Affichage des matchs en local

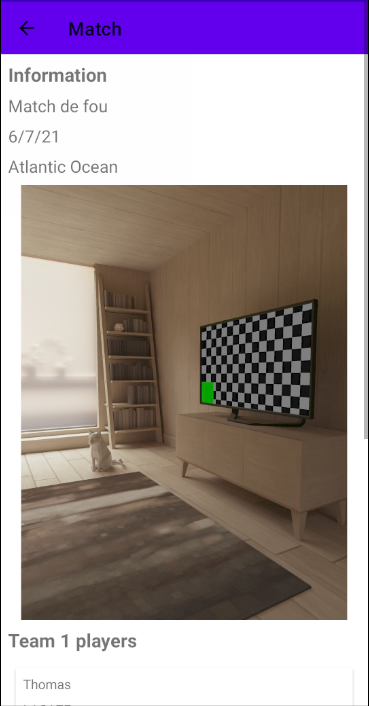
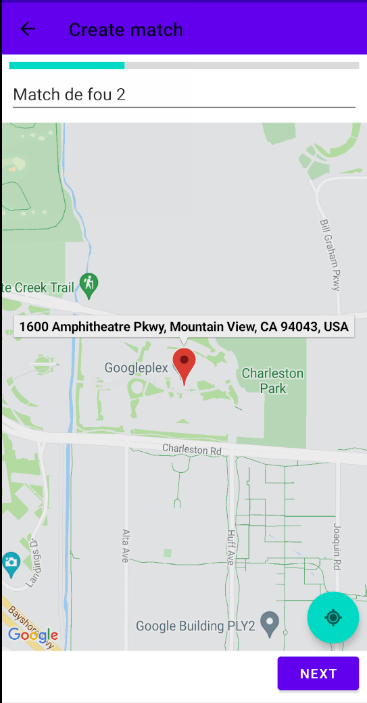
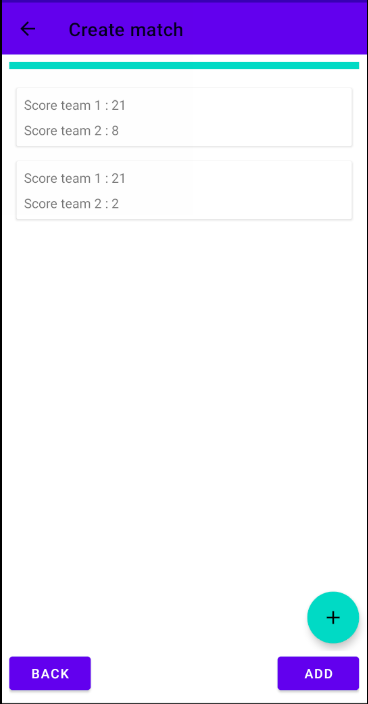


Figure 5 - Affichage des matchs local avec image

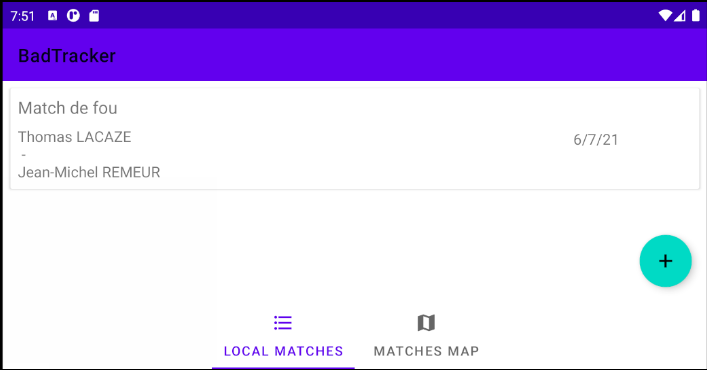
## Création match

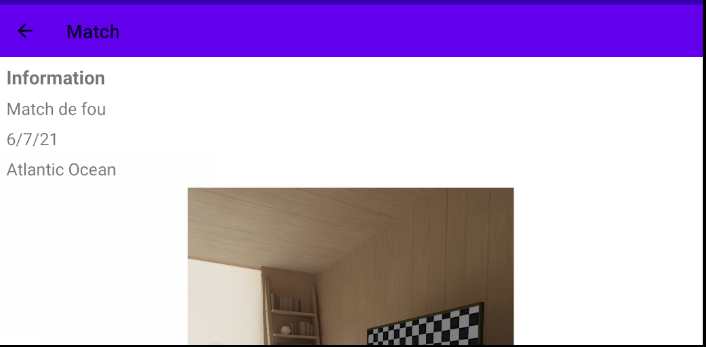


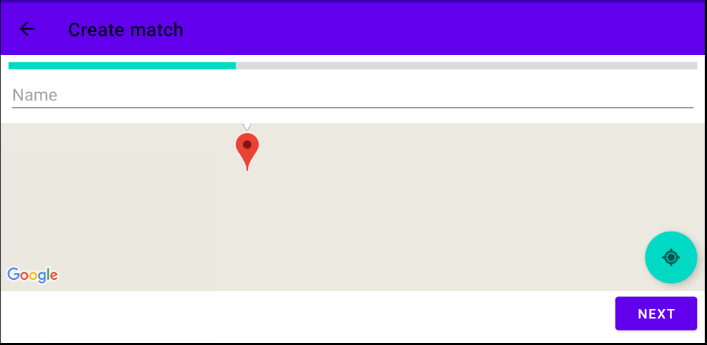


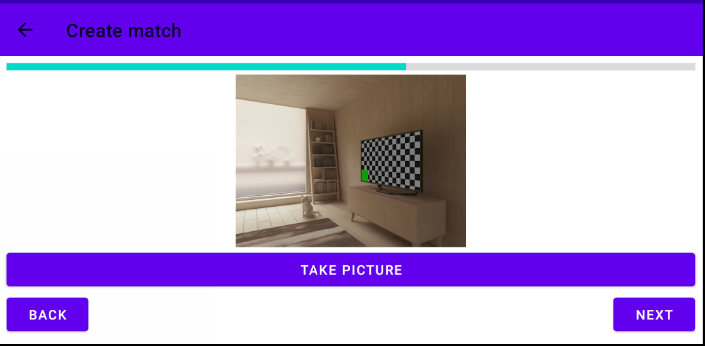


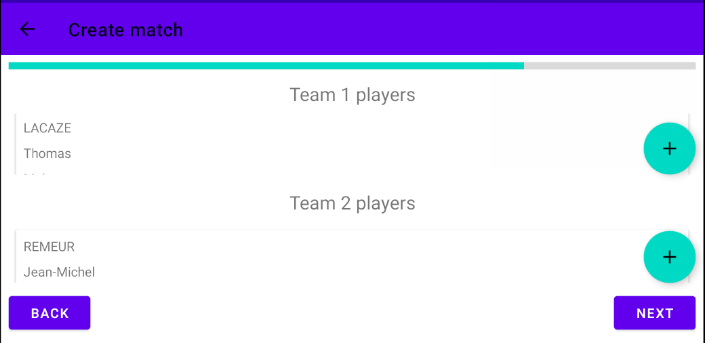
## Rotation écran













## Changement de langue



