

# COMPTE RENDU DE PROJET:

## Création d'une plateforme de gestion de joueurs et résultats

### MEMBRES DU GROUPE:

- BOUKOU Grace
- KADRI MOUNKAILA Askia Mohamed

### ENVIRONNEMENT DE DÉVELOPPEMENT:

**Java version:** java version "1.8.0\_181"

Java(TM) SE Runtime Environment (build 1.8.0\_181-b13)

Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

**IDE:** IntelliJ IDEA Ultimate Edition

**Grails version:** 3.3.8

### BASE DE DONNEES UTILISEE: SGBD H2

### DESCRIPTION GENERALE:

Nous avons conçu une plateforme de gestion de joueurs et de résultat sous forme de site permettant d'administrer ou de consulter son contenu selon son rôle. Selon les spécifications du projet nous nous somme plus penchés sur la partie administration de la plateforme et gestion des joueurs et des résultats.

L'administrateur peut créer des

### SATISFACTION DES CONTRAINTES ET DES BESOINS:

Nous avons deux niveaux d'accès via des rôles attribués à nos utilisateur:

- **ROLE\_ADMIN:** qui a un accès total au back office et peut opérer toutes les fonctions de CRUD. Il peut créer des utilisateurs avec des rôles d'administrateur ou des rôle de simple joueur. Il peut les modifier (changer leurs noms, photo de profile, décider si leurs comptes sont actifs ou non...) ou les supprimer. Il peut créer des matchs entre des joueurs et les administrer également de la même manière que les User en gérer les résultats associés à ces matchs. Il peut enfin créer des mettre entre des User. Il peut totalement administrer la base générée
- **ROLE\_USER:** qui sur la plateforme peut se connecter, se déconnecter à sa guise tant qu'il a un compte actif et consulter son profil. Dans l'ensemble il a un accès très restreint à la plateforme mais pourra sur l'API soumettre et récupérer des informations sans possibilité de modification.

Notre modèle se compose de trois domaines:

- **User:** qui peuvent s'inscrire avec un **username** (qui ne peut être null), un **password** (qui ne peut être null et crypté à la saisie) et fournir un **avatar** servant de photo de profile s'ils ne désire, autrement une photo par défaut leur est attribuée.

- Match: impliquant deux joueurs (max) permettent de garder une trace des résultats des confrontations des joueurs en gardant l'id du **winner**, celui du **looser**, et le score de chacun (**winnerScore** et **looserScore**)
- Message:

Pour chacune de ces entités, la date de création est sauvegardée.

Le scaffolding de base a été modifié pour donner un minimum de cohérence à la plateforme et un design moins verbeux.

Les images physiques sont affichées sur les pages et peuvent être chargées à partir des formulaires de modification et de création. Elle peuvent être chargée en faisant un simple drag'n'drop.

CRON: Un job à été mis en place pour lancer un trigger qui fait une purge de tous les messages lus à 4h du matin (pour qu'il puisse être testé, nous l'avons configuré pour qu'il fasse la purge chaque 30s)