

Application de paris sportifs

Version du 8 décembre 2020, 14:51

Objectifs

Le but de ce projet est de manipuler tous les concepts vus pendant l'Unité d'Enseignement à travers un jeu de paris sportifs.

1 Présentation générale du projet

L'objectif de ce projet est de réaliser une application Java EE pour un site de paris sportifs. Aucun argent réel ne sera utilisé bien entendu. Nous utiliserons pour ce projet une monnaie créée pour l'occasion : des Limcoins.

Cette application sera dotée, d'une interface Web réalisée en JSF permettant à l'Utilisateur de s'identifier en fonction de son type. Elle permettra également d'effectuer toutes les interactions entre l'utilisateurs et le reste de l'application. Elle devra également mettre en œuvre de la persistance à l'aide de JPA.

2 Les principales classes à écrire et les règles

Votre application devra comporter a minima 6 classes :

- La classe **Administrateur**
- La classe **Parieur**
- La classe **Bookmakeur**
- La classe **Pari**
- La classe **Match**
- La classe **Cote**

Vous devez respecter ces noms de classes !

L'**Administrateur** est chargé de la création, modification, suppression des **Parieurs** et des **Bookmakers**.

Le **Parieur** se caractérisera au minimum par son nom, et l'argent dont il dispose. Ce n'est pas limitatif, vous pouvez ajouter d'autres propriétés si vous le souhaitez. Par défaut, tout le monde démarre avec 1000 Limcoins en poche. Le **Parieur** peut visualiser tous les **Matches** proposés et poser des **Pari** sur tous les **Matches** qui l'intéressent dans la limite des fonds dont il dispose. Il peut également chercher un **Match** particulier qui l'intéresse.

Le **Bookmakeur** est chargé d'alimenter le jeu en match. Pour chaque match il peut créer une liste de **Cote** (score exact, victoire, défaite, nbre de buts ou points marqués etc.).

Un **Match** se caractérisera par un nom, un lieu, une liste de participants (joueurs ou équipes), une liste de cotes et une date de début et une durée. Pour des questions de traçabilité, les matchs gardent un historique de toutes les paris qui ont été faits. Ils pourront proposer une liste de **Cotes** et de **Paris** possibles. Les **Paris** pourront être modifiés par les **Parieurs** jusqu'au début du match.

Un **Pari** est attaché à un match et à une cote fixée au moment où il est passé. Il est remporté à l'expiration du match par tous les **Parieurs** ayant réussi leur pari. Si un **Pari** est modifié, c'est la cote disponible au moment de la modification qui s'applique.

Un classement des **Parieurs** et des **Bookmakers** sera également disponible dans l'application. Il s'effectue sur le montant en Limcoins dont ils disposent.

3 Fonctionnalités avancées

La section précédente présente les fonctionnalités minimum attendues, vous pouvez bien sûr aller beaucoup plus loin. Chaque amélioration sera prise en compte pour l'évaluation de votre projet. Voici quelques pistes possibles :

- Services REST : Un **Parieur** pourrait réaliser toutes les opérations de manipulation d'objets (CRUD) à travers des services Web REST. Ces services utiliseront le format JSON pour les échanges.
- Authentification et sécurité : Ajout d'un mode d'authentification pour renforcer la sécurité du site (voir cours) soit par des servlets de votre conception soit en utilisant les fonctionnalités de sécurité de JEE.
- Utilisation de twitter pour signaler à un **Parieur** qu'il vient de remporter un pari ou que de nouveaux matchs sont disponibles. La même chose est également envisageable pour les **Bookmakers**.
- Utilisation d'une API REST externe pour récupérer les listes et les résultats des matchs comme par exemple : <https://www.football-data.org/documentation/quickstart> ou <https://www.api-football.com/> (Attention les versions gratuites sont limitées dans le nombre d'appels que vous pouvez faire par heure ou par jour).

4 Mode d'évaluation

Le projet se fera en binôme.

Pour l'évaluation vous devrez :

- Fournir un document au format PDF (**et uniquement PDF**) d'**au plus 5 pages** présentant vos choix de conception ainsi que les principales fonctionnalités développées.

- Remettre un zip contenant toutes les sources de votre projet. Ces remises se feront **exclusivement** à travers la plateforme Communities de l'Université. Aucune remise par mail ne sera acceptée.
- Faire une démonstration de votre solution pendant la séance prévue à cet effet (voir ADE).

Concernant l'évaluation, la qualité de conception de votre application sera primordiale. La manière de répartir les différents composants et la façon dont ceux-ci vont communiquer sera déterminante dans la notation. La qualité et la clarté du code sera également prise en compte tout comme sa modularité et son extensibilité. La gestion des erreurs est également un plus.

L'ergonomie générale de l'application sera bien entendu importante pour l'évaluation mais elle ne sera pas valorisée au même niveau que les nouvelles fonctionnalités que vous pourrez apporter ou encore que l'architecture générale de votre application.