Plan de projet

*2013-12-03*

(Une fois complété, le document devrait faire 2 à 3 pages.)

Composition de l'équipe

Listez les membres de l'équipe.

* Rodier Madiande
* Juntie Xia
* Lionel Seleucus

Description du jeu réalisé

Quel est le principe du jeu que vous allez réaliser ? Décrivez dans les grandes lignes la règle du jeu, la manière dont les points seront décomptés, etc.

A chaque nouvelle connexion l'utilisateur est invité soit a s'authentifier ou a créer un nouveau compte.

Le compte est crédité de 1000 jetons à la création, quand ce dernier atteindra 0, il ne pourra miser qu'un jeton par parie et un seul parie maximum à la fois.

Le principe c'est de parier sur des matchs, et en fonctions du résultat l'utilisateur gagner la somme calculé par la formule suivante si le résultat prédit est juste.

somme = cote \* mise. (choix d'un seul match)

somme = (cote1 + coteN) \* mise (choix de plusieurs matchs)

~~L'utilisateur débloque des succès quand il arrive à telle ou telle somme de jetons, ça lui donne des titres.~~

La cote d'une équipe est calculé à partir de son pourcentage de victoire, pour 50 % de victoire la cote est de 1, pour chaque pourcentage en moins on lui ajoute 0,02 de cote et inversement, 0,02 de cote en moins pour chaque pourcentage de cote en plus.(51 % => 0,98 | 49 % => 1,02 | attention ! Une équipe qui à 100 % de victoire n'a pas 0 de cote, mais 0,02 qui est la cote minimum)

A la fin de chaque mois, les 3 meilleurs utilisateurs affichant le taux de réussites NbreParisTentés/NbreParisRéussi, se verront offrir des cadeaux (Jetons ou bonus de multiplicateur sur le prochain parie).

Architecture d'ensemble

Décrivez l'architecture d'ensemble de l'application que vous allez réaliser (client, serveur, base de donnée, back-end, etc.). Quels sont les principaux composants et où sont implantées les principales fonctions?

\*Client Web :

* Fournit l’interface utilisateur permettant de créer un compte, s’authentifier, sélectionner des événements afin d’enregistrer des paris, consulter les résultats des événements, consulter les statistiques des autres joueurs et enfin consulter son historique.
* Possibilité via l’interface de pouvoir mettre à jour ses informations.
* Demande un nouveau de mot de passe, si on ne s’en souvient plus.

\*Serveur Web :

* Le serveur web implémentera les requêtes permettant de :

1. Enregistrer un nouvel utilisateur
2. Valider ou non un utilisateur lors de l’authentification
3. Possibilité de demander un nouveau mot de passe
4. Possibilité d’éditer ses informations utilisateurs
5. Afficher la liste des événements sur les lesquelles un pari peut être enregistré
6. Afficher les statistiques pour un événement donné
7. Afficher des informations le classement des équipes, leurs joueurs, leurs profiles, les bléssés
8. Enregistrer un pari (Simple ou Cumulé)
9. Consulter les paris en cours
10. Consulter les résultats des événements en fonction de la date
11. Consulter son historique de paris
12. Se Désinscrire (Est ce vraiment utile?)

\*Base de données :

* Stocker les utilisateurs (identifiants + *mots passes*)
* Stocker les événements
* Stocker les résultats
* Stocker l’historique des paris de enregistrés par chaque utilisateur
* Stocker l’historique des jetons de réinitialisation de mots de passes demandé

\*Back-end :

* Mise en place d’un index de recherche pour améliorer l’efficacité des recherches
* Traitement périodique afin de récupérer la liste des événements et les résultats
* Traitement afin de statuer sur les paris + envoi d’un mail de notification à l’utilisateur

Interface utilisateur

Décrivez les principales vues de l'interface utilisateur (informations présentées, fonctionnalités, navigation, etc.)

L'application est censé affiché:

Les résultats des derniers matchs.

Les paris des matchs qui ferment prochainement.

Un lien vers la liste de tous les paris d'un tel sport.

Un lien vers les résultats des paris de l'utilisateur. (Ainsi que les paris en cours, + possibilité de recevoir la notification du paris par email au moment de la prise de paris)

Source de données

Identifiez la source externe de données que vous allez utiliser (API). Décrivez les différents appels que vous allez faire à cette API et leur périodicité.

On va utiliser l'API Sportsdatallc.

On va volontaire limiter l’utilisateur à pouvoir parier sur des événements sur un jour donné.

~~Etant donnée que nous sommes limitées sur le nombres d’interrogations par rapport à l’API~~

~~On va périodiquement mettre à jour les résultats des évènements + statuer sur les paris en cours + envoyer les notifications aux utilisateurs.~~

Ca va donner :

* Récupération du calendrier des rencontres une fois par jour. (Single Game Day Schedule)
* Idem pour les résultats (Game Boxscore), 3h après chaque évenements.
* Idem pour le classement (Standings)
* Idem pour les informations équipes (Team Profile)
* Idem pour les informations joueurs (Players Profile)
* Récupération des informations sur les blessures 3h, 2h, 1h, 30min avant l’événement (Injuries)
* Récupération d’un avant-match de l’événement 3h avant.
* Récupération des données statistiques sur la saison une fois par jour

NB : Il se peut qu’il y eu +/- d’informations de proposés selon tests réels effectutés par rapport à l’API

Choix technologiques

Listez les principales technologies que vous allez utiliser pour la réalisation de chacun des composants. Identifiez le système que vous allez utiliser pour déployer le ligne votre application.

* WebServer : Google App Engine
* Persistence des données : MySQL
* Client : JEE + toutes les technos autour de HTML5
* Back-end : Java

Organisation du travail

Définissez le rôle de chaque membre de l'équipe dans la réalisation du projet. Spécifiez les outils que vous allez utiliser pour partager le code et déployer l'application. Définissez les principaux jalons du projet.

On va utiliser Github pour héberger le code pendant le développement.

Principaux jalons :

* Traitement lié à l’API (Parsing + Mise à jour des données + Design BDD)
* Design BDD
* Design Architecture de l’application WEB
* Développement + Tests Unitaires (Sous formes de livrables après chaque *features* développées)
* Recettage Globales
* Livraison