


## 1 Présentation du problème

L'objectif de ce projet est de réaliser un site web de Bourse d'Information. Une bourse d'information est un système boursier permettant de gérer des marchés sur une information qui sera vérifiée à coup sûr dans le futur. Le marché d'une information est un système permettant aux utilisateurs d'acheter ou de vendre des bons sur cette information jusqu'à la date d'échéance. Cela fonctionne comme un système de paris, à la différence près que l'on choisit à tout moment de modifier ses actifs avec les critères (quantité, montant, achat ou vente) que l'on souhaite. Comme dans n'importe quelle bourse, il y a deux rôles dans ce système : celui qui lance des marchés, le MarketMaker, et celui qui joue en bourse. Votre objectif est de réaliser un tel système, permettant à chacun de gérer ses actifs.

Quand un MarketMaker lance un marché (par ex "PSG battra l'OM d'ici le 1/07/2014"), un autre marché symétrique est automatiquement créé avec l'option inverse. Initialement, chaque nouvel inscrit se voit créditer de 10.000 euros. Chaque joueur peut alors émettre des ordres d'achat ou de vente dans certaines quantités selon son portefeuille. Un prix d'achat est considéré comme un prix maximum, tandis qu'un prix de vente est considéré comme un prix minimum. Dans ce système, chaque bon est toujours coté entre 0 et 100. Les ordres sont triés sur leur prix et, à égalité de prix, sur la date de saisie. On appelle cela un carnet d'ordres. Quand un ordre d'achat est supérieur ou égal à un ordre de vente, l'ordre est exécuté au prix du premier arrivé pour la quantité possible. Cet accord de vente fixe le cours du moment (utilisé pour des courbes ou des évaluations de portefeuille). Un ordre peut évidemment "matcher" plusieurs autres ordres ou même qu'une partie d'un ordre. Evidemment, on ne peut acheter que s'il y a un vendeur, et réciproquement. Chaque ordre d'achat (resp de vente) sur une information possède automatiquement sa réciproque en ordre de vente (d'achat) au complémentaire à 100 sur l'information inverse. Par ex si je passe un ordre d'achat de 10 bons à 40 euros, réciproquement il y a un ordre de vente de 10 bons à 60 euros sur l'information inverse. A tout moment on peut passer des ordres d'achat et de vente jusqu'à la date de fin du marché. A la date d'échéance, le MarketMaker rentre l'option réalisée, ce qui déclenche les paiements : chaque bon vérifié est racheté par la banque à son propriétaire à une valeur de 100. Celui qui avait 40 bons achetés à 60 euros obtient donc 40.000 euros sur son compte. Les bons perdants ne rapportent rien pour leurs propriétaires.

### Jack Lang entrera au Gouvernement avant l'Hiver 2008



Vendeurs		
---	5 bons	à 49D
---	61 bons	à 47D
---	563 bons	à 44D
pierremr	9 bons	à 43D

Acheteurs		
---	17 bons	à 37D
---	395 bons	à 36D
fabyo	500 bons	à 35D
---	33 bons	à 31D

**ASTUCE** : Si vous ne croyez pas à ce pronostic, [investissez dans le pronostic inverse.](#)

nombre de  
bons

prix  
unitaire

Acheter

**Mon Compte**

**Cash disponible**  
2 000

**Mes Bons**  
0

**Valeur**  
0

**Ma Performance**  
0%

**Mon Gain**  
0

[Mes Transactions](#)

[Tous les ordres](#) | [Pronostic Inverse](#) | [A l'aide !](#)

issu de <http://science.newsfutures.com>

Techniquement, à l'instant  $t$ , chaque inscrit possède un portefeuille et des bons sur différents marchés. La base de données doit bien évidemment contenir et gérer le portefeuille de chacun. Chaque marché est défini par une date de fin et l'affirmation supportée (avec son inverse). Tout est entièrement symétrique : les marchés comme les ordres, ce qui garantit l'équilibre financier du système.

## 2 Différents niveaux

Il y a grosso-modo 8 pages WEB à faire, décrites ci-dessous. Comptez une 1h par page. Faites les dans l'ordre. Vous êtes en binôme : répartissez vous les tâches !

**Modélisation** . Commencez par une petite modélisation de la base et des pages WEB que vous voulez réaliser. Commencez par choisir vos tables, établir le MCD et créer la base. Il vous faut décider si les symétriques sont explicites ou implicites dans la base. Faites des maquettes HTML de vos pages (il y en a très peu), une fois certain de vos choix, codez les en dynamique.

**Achat-Ventes** . C'est la (les) page principale. Vous commencez par celle-ci en ayant mis les infos en dur dans la base. Elle est de manière évidente en 5 ou 6 parties (l'information, les 2 carnets d'ordre, le passage d'une opération, l'état de mon compte et les liens complémentaires).

**Charte graphique** . Afin de rendre le projet plus professionnel on utilisera avantageusement des CSS pour définir un style de site. Toute mise en forme doit être dans la CSS ... rien dans la page web. Par exemple, le positionnement des blocs précédents se fera avantageusement avec des `div`

**Enregistrement d'un marché** . On doit notamment saisir l'information, sa négation et la date de fin de ce marché. une fois la date dépassée, il ne doit plus être possible de passer un ordre. Néanmoins, seul le market-maker peut clore et distribuer les agins d'un marché manuellement.

**Authentification** . Chacun doit pouvoir s'authentifier. Idéalement il doit donc y avoir des rôles (joueur, marketmaker, admin) pour chaque individu (donc au minimum (login,mdp,rôle)). Dans un premier temps cela peut être simplement en dur dans la base. Les pages ne deviennent maintenant accessibles que si on a le rôle correct : L'enregistrement d'un marché pour la banque, l'Achat-Ventes pour le joueur. Un Realm doit être utilisé pour cela.

**Les meilleurs joueurs** . Les très bons joueurs, qui ont une balance de gains positive (10 par ex) doivent pouvoir devenir Marketmaker.

**Saisie d'un nouvel arrivant** . Permet la saisir des informations pour un nouvel arrivant.

**Mes transactions** . Affiche mes transactions en cours avec éventuellement possibilité de les annuler.

**Documentation** .

**Cette partie est obligatoire** : Une documentation PDF sera rendue avec le projet. Elle détaillera notamment les choix d'utilisation effectués, les choix techniques, les difficultés rencontrées et une rubrique de perspectives d'améliorations possibles. Elle contiendra évidemment le script SQL de création des tables, le MCD et la hiérarchie des pages HTML.

**A rendre pour le 3 février (5 semaines)**