Grenoble INP-ENSIMAG

Conception : jeu de simulation boursière

CHRIF M'HAMED Mohamedou, EL IDRISSI Khaoula,
MOHAMED AHMED Mohamed Lemine

Contents

1 Introduction

Dans ce document, on explique la conception détaillée de la partie implémentée du cahier des charges. nous avons choisi d'implémenter une partie important du chaier de charges comme l'achat et la vente d'actions, la simulation du prix d'une action avec interface graphique simple.

2 choix de l'Architecture : MVC

Pour implémenter ce jeu de simulation, nous avons adopté l'architecture MVC (Modèle - Vue - Contrôleur), avec une partie Modèle responsable de toute la logique liée à la gestion du portefeuille, à l'utilisateur ainsi qu'à l'achat et à la vente des actions.

La partie Vue concerne l'interface utilisateur, telle que le tableau de bord, etc., détaillée dans la section 4 de ce document. Enfin, la dernière partie, le Contrôleur, assure la communication entre la partie Vue et le Modèle

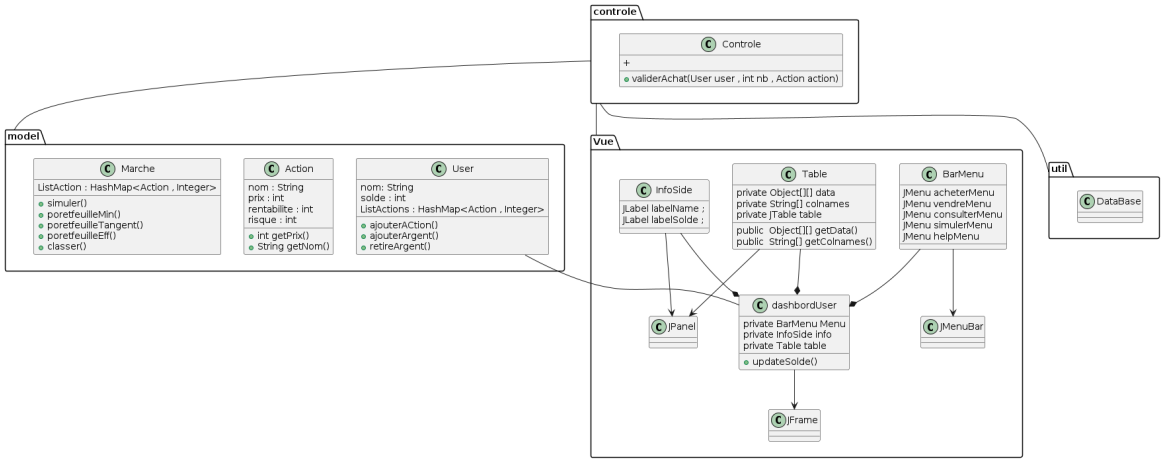


Figure 1: Conception MVC

3 Conception du Modèle

Dans la conception du modèle, nous avons trois classes principales dans un package appelée: **modele**

3.1 User

- cette classe liée à toutes les opérations et les propriétes liée à l'utilisateur comme son nom et son solde
- la classe **User** Présente une méthode essentielle **ajouterAction(Action choix , int nb)** qui est appelée lors de l'achat d'une action pour valider que l'utilisateur dispose d'un solde suffisant pour cet achat.
- Pour les actions de l'utilisateur, nous stockons celles-ci dans une **HashMap** afin de réduire la complexité de la recherche de $\Theta(n)$ à $\Theta(1)$.

3.2 Marche

- Cette classe est liée à toutes les opérations concernant le marché lui-même, telles que l'ajout d'actions particulières ou l'attribution des actions à un utilisateur lors de l'achat

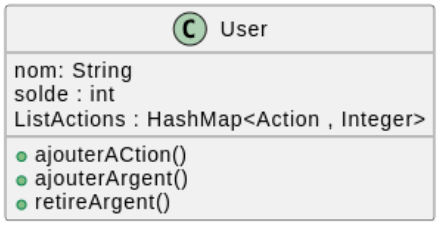


Figure 2: User

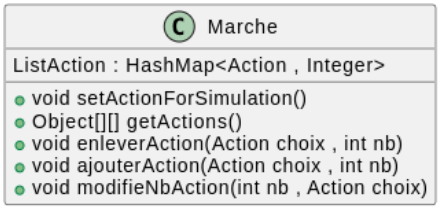


Figure 3: Marche

- Pour les actions du marché , nous générons des actions aléatoires pour les prix suivant une loi uniforme continue entre 100 et 400, et pour les rentabilités et les risques suivant une loi normale centrée réduite.

3.3 Action

- cette classe réprsente une action dans le marché finaiciére
- chaque action est carcatériser par : nom , prix , rentabilité , risque

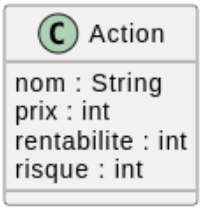


Figure 4: Action

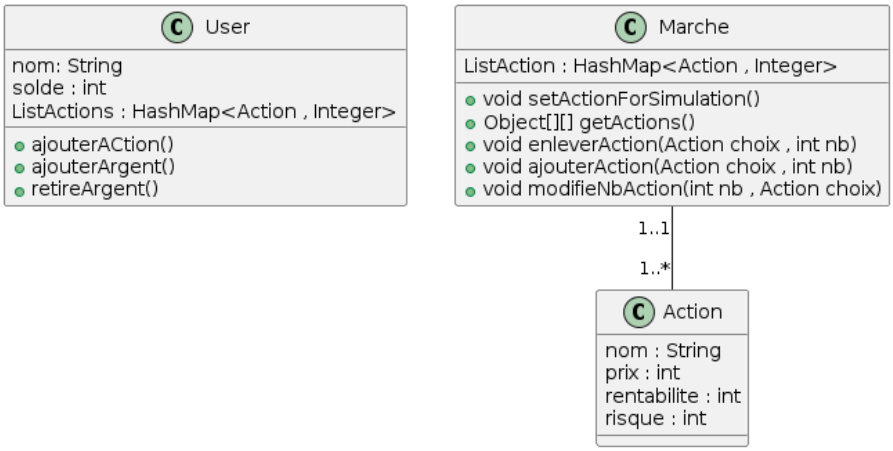
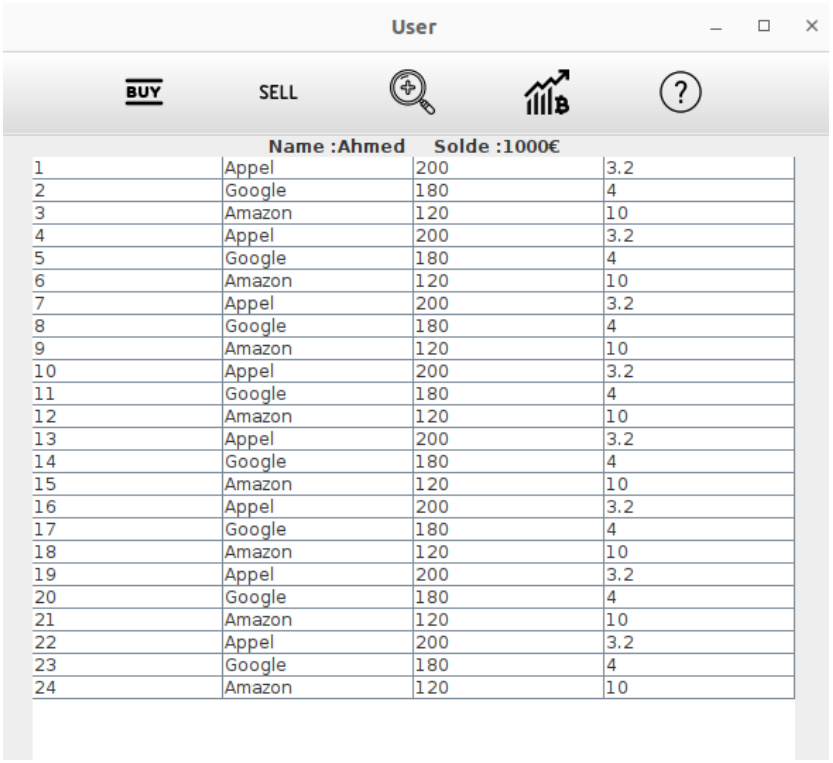


Figure 5: Conception Modele

4 Conception de l'interface utilisateur

Nous avons tenté de concevoir une interface utilisateur conviviale en utilisant le package **java.swing**, en intégrant des dépendances externes telles que **JFreeChart** pour simuler des actions .

Dans cette section, nous expliquons brièvement la hiérarchie de classes que nous avons utilisée pour implémenter l'interface utilisateur



1	Appel	200	3.2
2	Google	180	4
3	Amazon	120	10
4	Appel	200	3.2
5	Google	180	4
6	Amazon	120	10
7	Appel	200	3.2
8	Google	180	4
9	Amazon	120	10
10	Appel	200	3.2
11	Google	180	4
12	Amazon	120	10
13	Appel	200	3.2
14	Google	180	4
15	Amazon	120	10
16	Appel	200	3.2
17	Google	180	4
18	Amazon	120	10
19	Appel	200	3.2
20	Google	180	4
21	Amazon	120	10
22	Appel	200	3.2
23	Google	180	4
24	Amazon	120	10

- **class dashbordUser** : Cette classe est la classe principale de l'interface graphique. Elle contient 3 attributs : un attribut représentant la table des actions du client, le BarMenu qui contient les fonctionnalités de l'application, et un attribut 'info' qui contient des informations personnelles telles que le nom et le solde."
- **class BarMenu** Cette classe contient toutes les fonctionnalités de l'application, telles que l'achat/la vente d'actions ou la simulation du marché financier.
- **InfoSide** Cette classe contient les informations personnelles du client, telles que son nom et son solde.
- **Table** La classe Table est responsable de la partie vue de la table des actions de l'utilisateur.

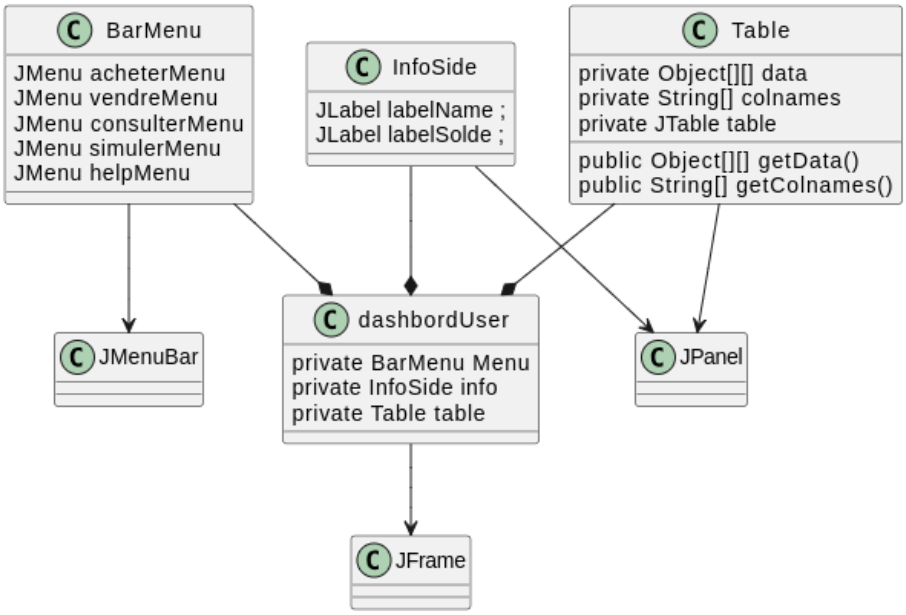


Figure 6: Conception de l'interface utilisateur