



Institut Supérieur d'Électronique de Paris

Algorithmique et programmation

Cycle ingénieur (A1)

Application de gestion de portefeuille

Étudiants :

CHÉMERY Julien

MERBOUCHE Mouloud

TEBOUL Elisa

Groupe :

Émoulien

Encadrant :

ALLANIC Hervé

Code module : IR.1101

18 janvier 2024

Table des matières

| | | |
|-----|---------------------------------------|---|
| 1 | Présentation du sujet | 2 |
| 2 | Fonctionnalités de notre application | 2 |
| 3 | Diagramme UML | 3 |
| 4 | Description de la structure du projet | 4 |
| 4.1 | Structure interne | 4 |
| 4.2 | Structure de l'interface | 5 |
| 5 | Déroulement du projet | 6 |
| 5.1 | Répartition des tâches | 6 |
| 5.2 | Difficultés et solutions | 6 |
| 6 | Conclusion : expérience acquise | 7 |

Table des figures

| | | |
|---|---|---|
| 1 | Diagramme UML contenant les constructeurs et les variables. | 3 |
| 2 | Structure interne du projet. | 4 |
| 3 | Interface du dashboard et de l'historique | 5 |

1 Présentation du sujet

Le projet II.1102 de cette année consiste à développer une application graphique de gestion de portefeuille financier en Java. Cette application permettra de suivre les valeurs d'un ou plusieurs portefeuilles au fil du temps, qu'il s'agisse d'actions ou de cryptomonnaies. L'application doit répondre à des exigences et pour ce faire utilise deux API publiques : Coin Gecko (limité à 5 messages par minute) et Alpha Vantage (limité à 25 requêtes par jour).

2 Fonctionnalités de notre application

Les fonctionnalités sont les suivantes :

1. Compte

- Création : L'utilisateur peut créer un compte en renseignant son nom et un mot de passe complexe.
- Connexion : L'utilisateur pourra ensuite se connecter avec le compte qu'il a créé.

2. Portefeuille

- Création : L'utilisateur peut créer des portefeuilles avec des noms différents.
- Suppression : Il est possible de supprimer un portefeuille parmi ceux créés.
- Duplication : Il peut dupliquer un portefeuille existant, en conservant les informations relatives aux cryptomonnaies et actions, mais en remettant les quantités à 0.
- Rechargement : L'utilisateur peut recharger son compte avec de l'argent, qui sera ensuite disponible pour pouvoir acheter avec ses portefeuilles.
- Historique : L'utilisateur a accès à un historique de transaction global pour son compte qui lui permettra de voir toutes les actions et cryptos d'une vue globale.

3. Actions / Cryptomonnaies

- Achat : L'utilisateur peut acheter des actions ou des cryptomonnaies en renseignant le nombre d'unité à acheter et pourra voir le prix actuel de celle-ci.
- Vente : Il est possible de vendre des actions ou des cryptomonnaies, la plus-value ou la perte sera ajoutée ou déduit du solde de l'utilisateur.
- Recherche de cryptos : L'utilisateur peut rechercher parmi la base de données de l'API (Coin Gecko) la valeur d'une cryptomonnaie.
- Recherche d'actions : L'utilisateur peut rechercher, parmi la base de données de l'API Alpha Vantage, une entreprise (avec un mot clé, par exemple Nvidia au lieu de NVDA) pour avoir la valeur d'une action.

4. Affichage

- La visualisation de l'historique d'une cryptomonnaie (BTC et ETH), nous pouvons voir l'historique en fonction de différentes années, avec la valeur réelle affichée pour l'année en cours sur le graphique.
- L'affichage d'un camembert indiquera la répartition d'actions et de cryptomonnaies au sein d'un portefeuille en terme de quantité.

5. Sauvegarde

- L'application permet de sauvegarder automatiquement chaque portefeuille, chaque transaction et tout les éléments d'un compte.

3 Diagramme UML

Les diagrammes UML nous ont d'abord permis de pouvoir s'orienter de façon plus précise pour la partie réalisation du projet. Nous avons choisi pour cette partie d'afficher uniquement le diagramme UML contenant, les constructeurs et les variables. Ce choix est justifié par des raisons de simplicité pour le rapport. En effet, l'ajout des méthodes rend la taille du diagramme beaucoup trop grande. Toutefois, les autres diagrammes UML sont toujours disponibles à l'adresse suivante : UML.

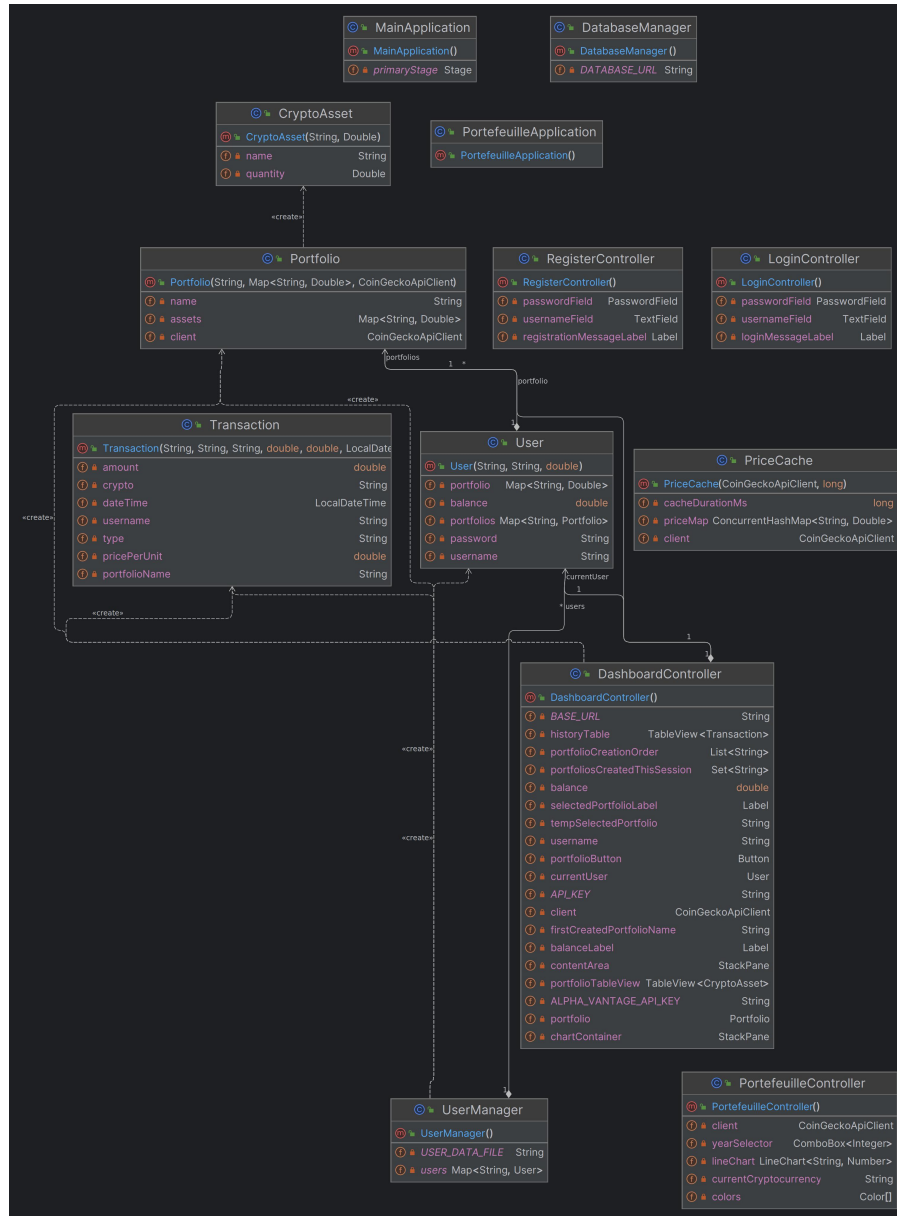


Figure 1 – Diagramme UML contenant les constructeurs et les variables.

4 Description de la structure du projet

4.1 Structure interne

Durant la phase de planification, la structure du projet fut l'une de nos priorités. Pour ce faire, nous avons choisi d'adopter une structure avec un tableau de bord centralisant les diverses fonctionnalités de notre projet. De plus, nous avons essayé de suivre le pattern M.V.C. (Model View Controller) pour certaines fonctionnalités, cela nous a permis de pouvoir tester les fonctionnalités sans avoir à se connecter à chaque fois. Chaque fichier de type Controller a un fichier FXML qui lui est associé. De plus, nous avons utilisé un fichier XML afin de pouvoir gérer les dépendances sous Maven. Notre projet est structuré de manière à faciliter l'accès et l'organisation des composants. Du point de vue de la structure des fichiers, les codes sont organisés au sein d'un dossier (package) `src` qui se divisent en deux sections principales.

1. La première avec les ressources c'est-à-dire les fichiers FXML et les images.
2. La seconde contient les codes des classes Java. Cette dernière section est elle même séparée en deux sections différentes : La première consacrée à l'API Coingecko. La seconde contenant le code des classes Java propre à notre projet.

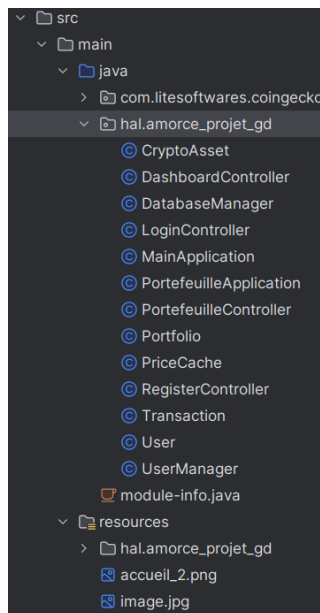


Figure 2 – Structure interne du projet.

4.2 Structure de l'interface

Concernant la structure de l'interface de notre projet, nous avons décidé de mettre toutes nos fonctionnalités sur une barre verticale (VBox) à gauche de notre tableau de bord. Grâce à cette structure nous avons une bonne visualisation générale de l'intégralité des fonctionnalités que nous avons réalisées. Ensuite, l'écran à droite de cette barre verticale sert d'affichage. Nous avons trouvé que de cette manière, il était plus simple et clair pour l'utilisateur de n'avoir qu'une seule et même interface.

Ensuite, pour les boutons, étant donné que nous avons beaucoup de fonctionnalités à implémenter, nous avons décidé d'en rassembler certains dans un menu déroulant. Par exemple, nous avons concentrer les boutons dupliquer, supprimer, ajouter et choix du portefeuille dans un même bouton appelé 'Options du portefeuille'. Nous avons aussi organisé l'UI sous forme de section (Portefeuille, Investir, Cryptos et Action) pour une meilleure lisibilité. Pour réaliser l'UI du programme nous avons utilisé un fichier styles.css qui régit les règles esthétiques des boutons, tableaux et menus déroulants. Cela nous a permis d'enrichir l'aspect visuel de l'application tout en réduisant la taille des fichiers fxml. Nous avons ainsi pu ajouter des fonctions de customisations précises comme des modifications esthétiques lorsque le curseur survole des boutons cliquables ou par exemple la couleur de la sélection des lignes des tableaux.

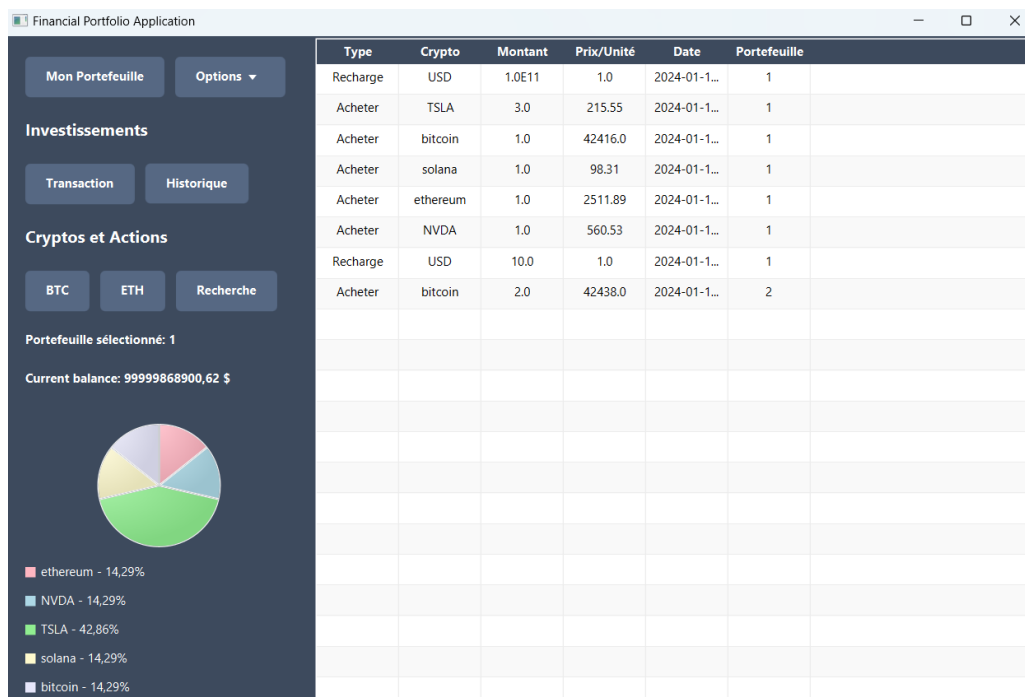


Figure 3 – Interface du dashboard et de l'historique

5 Déroulement du projet

Avant de commencer à coder, nous avons procédé en plusieurs étapes. Tout d'abord, nous avons regardé les différentes fonctionnalités du projet qui étaient à réaliser afin de comprendre au mieux ce qui était attendu. Ensuite, nous avons classé en terme de difficulté les différentes fonctionnalités du projet. Faire cette classification nous a permis non seulement de visualiser l'ensemble des attentes mais également de pouvoir commencer par des fonctionnalités qui étaient abordables afin de ne pas rester bloquer sur une fonctionnalité difficile et ne pas avancer. Dans cette partie, nous aborderons donc de la répartition des tâches que nous avons adopté durant ce projet, puis les difficultés rencontrées ainsi que les solutions à ces problèmes.

5.1 Répartition des tâches

Tout d'abord, concernant la répartition des tâches, nous avons procédé d'une manière bien précise. Comme énoncé précédemment, nous avons classifié par intensité de travail chaque aspect du projet. Une fois cette classification faite, nous avons pour méthode de commencer par la première fonctionnalité à implémenter. Nous essayons tous de travailler sur cette fonctionnalité. Lorsqu'une personne arrive à faire marcher son code, elle le met sur Github et le reste du groupe partait donc du code de la personne ayant réussi afin de pouvoir commencer la deuxième. Nous avons adopté ce processus du début à la fin du projet. Nous avons trouvé que cette méthode était la plus efficace afin d'avoir un maximum de productivité. En effet, lorsque chacun avance sur une partie du projet, il est possible que les deux codes ne soient pas compatibles ce qui provoquerait donc une énorme perte de temps afin de rendre les deux codes compatibles.

Enfin, en ce qui concerne la soutenance et le rapport, nous nous sommes répartis les parties du rapport et nous avons présenté chaque partie que nous avons rédigé afin d'être le plus efficace possible.

5.2 Difficultés et solutions

Ensuite, au cours du développement de ce projet nous avons rencontré plusieurs difficultés. Cette partie abordera donc les différentes difficultés rencontrées et la solution que nous avons trouvée pour pallier ces problèmes.

Tout d'abord, trouver les classes qui allaient constituer notre projet était compliqué. En effet, ce projet était le premier durant lequel nous devions choisir nous même le nom des classes et l'utilité de chaque classe. La réalisation du diagramme nous a beaucoup aidé à visualiser les différentes classes qui allaient constituer notre projet. En outre, nous avons trouvé que interagir avec l'API (Coin gecko) était compliqué. Cependant, en faisant des recherches sur internet ainsi que grâce au cours du professeur nous expliquant brièvement comment interagir et importer coin gecko, nous avons finalement pu réussir. Nous avons été limité par l'API qui n'autorise que 5 requêtes toutes les 5 minutes donc certaines idées on dû être abandonnées. Enfin, une difficulté qui nous a posé problème au cours du projet était l'ajout de fonctionnalités. Au fur et à mesure de l'ajout de fonctionnalités, nous avons trouvé que parfois ajouter une fonctionnalité provoquait des bugs dans une autre fonctionnalité. Il était donc compliqué de rajouter des fonctionnalités au fur et à mesure de l'avancement du projet. La sauvegarde via une base de données SQL, fut également une tâche difficile, nous avons d'abord essayé plusieurs autres méthodes de sauvegarde qui n'aboutissaient pas. Puis, après diverses recherches, nous en avons

conclus que la base de données SQL était la meilleure option. Enfin, nous avons été confronté à une autre difficulté trouvée à la fin de notre projet. En effet, nous nous sommes rendus comptes que l'API Coingecko pouvait nous fournir uniquement des valeurs de cryptomonnaies, toutes les valeurs liées à des actions d'entreprises étaient absurdes. Pour résoudre ce problème, nous avons du procéder à l'ajout d'une nouvelle API : Alpha Vantage qui nous permet d'obtenir les valeurs des actions d'entreprises. Cependant travailler avec cette API a été un nouveau problème supplémentaire puisque nous sommes limités à 25 requêtes par jour. Nous avons donc organisé des sessions produit où l'un d'entre nous utilisait toutes ses requêtes pour tester les bugs de l'application et le reste de la journée on limitait au maximum les requêtes pour faire un test final en fin de journée.

Ainsi, ce projet nous a permis de nous confronter à des problèmes et donc de savoir comme réagir face à ces derniers lors de projets futurs.

6 Conclusion : expérience acquise

Ainsi, ce projet nous a permis d'acquérir de nouvelles compétences ou d'en approfondir certaines. Premièrement, ce projet nous a permis d'améliorer notre niveau en java. En effet, en proposant un projet d'une difficulté élevé, cela nous a permis de nous dépasser et d'apprendre d'une manière ludique à coder en java. Deuxièmement, ce projet nous a permis de mieux savoir s'organiser en équipe que ce soit en terme de communication ou encore de collaboration, ce projet nous aura été utile dans nos projets futurs. Par ailleurs, la résolution de problème a constitué une grosse partie de notre projet. En effet, le fait de réfléchir en équipe aux différents problèmes que nous avons rencontré a été enrichissant. Ce projet a donc été bénéfique sur le plan mental comme sur le plan social et nous a permis d'améliorer nos compétences.