

Université Paris Dauphine – PSL

Cahier des charges du projet final C#

Outil de gestion et d'optimisation de portefeuilles
financiers

avec récupération automatique des données *Yahoo Finance*

Réalisé par :

Antoine Moniz et Adam Bousselmame

Formation : Master 272 – Économie et Ingénierie Financière

Année universitaire : 2025–2026

Encadrant : Laurent Davoust

Table des matières

1	Introduction au problème posé	2
1.1	Contexte	2
1.2	Objectifs du projet	2
1.3	Limites du projet	2
1.4	Sources théoriques	2
2	Expression fonctionnelle du besoin	3
3	Solution proposée	4
3.1	Données utilisées	4
3.2	Modèle mathématique	4
3.3	Architecture orientée objet	4
4	Planification du projet	5
4.1	Évolutions possibles	5

1 Introduction au problème posé

1.1 Contexte

Dans un contexte de volatilité accrue des marchés financiers, la gestion de portefeuille et l'optimisation du couple rendement/risque sont au cœur des stratégies d'investissement.

Les étudiants et analystes financiers ont souvent besoin d'un outil simple leur permettant :

- de gérer un portefeuille d'actifs ;
- d'analyser sa performance et d'optimiser la répartition des poids selon la théorie de Markowitz.

Ce projet vise à développer en C# (programmation orientée objet) un logiciel interactif capable :

- de récupérer automatiquement les données financières réelles (prix historiques et actuels) via *Yahoo Finance* ;
- de calculer les indicateurs de performance d'un portefeuille ;
- et de proposer une allocation optimale selon la théorie moyenne-variance.

1.2 Objectifs du projet

L'application doit permettre de :

- gérer plusieurs portefeuilles indépendants ;
- saisir des tickers d'actifs (ex. AAPL, MSFT, SAN.PA) et en extraire les prix historiques ;
- calculer automatiquement :
 - le rendement attendu,
 - la volatilité et la covariance,
 - le ratio de Sharpe ;
- optimiser la répartition des poids selon Markowitz et afficher la frontière efficiente.

1.3 Limites du projet

- Connexion uniquement à *Yahoo Finance*.
- Sauvegarde locale sans base de données.
- Nécessite une connexion Internet pour la récupération initiale des données.

1.4 Sources théoriques

- Markowitz, H. (1952). *Portfolio Selection. The Journal of Finance*.
- Documentation `YahooFinanceApi` (NuGet).
- Documentation `Math.NET Numerics`.

2 Expression fonctionnelle du besoin

Code	Description	Objectif	Critères de réussite	Contraintes / dépendances
F1 – Saisie des tickers	L'utilisateur entre une liste de symboles (ex. AAPL, MSFT, SAN.PA)	Définir le portefeuille à étudier	Les tickers sont valides et reconnus par Yahoo Finance	Vérification du format
F2 – Téléchargement des données	Récupération automatique des prix historiques via Yahoo Finance	Obtenir des données de marché réelles	Données cohérentes et complètes	Connexion Internet requise
F3 – Calcul des performances	Calcul du rendement moyen et de la volatilité de chaque actif	Évaluer les caractéristiques individuelles des actifs	Résultats exacts et sans valeurs manquantes	Données disponibles sur la période
F4 – Calcul du portefeuille global	Calcul du rendement, de la volatilité et du ratio de Sharpe du portefeuille	Mesurer la performance globale	Résultats cohérents avec les pondérations	Matrice de covariance valide
F5 – Optimisation Markowitz	Recherche des poids maximisant le ratio de Sharpe	Fournir une allocation optimale	Poids positifs, somme = 1	Utilisation de Math.NET
F6 – Visualisation graphique	Affichage de la frontière efficiente et du portefeuille optimal	Représenter la relation rendement/risque	Courbe lisible et sans erreur	Librairie OxyPlot / Chart
F7 – Gestion du portefeuille	Sauvegarde et recharge local (CSV/JSON)	Réutiliser les analyses	Fichiers créés correctement	Utilisation de CSVHelper
F8 – Interface utilisateur	Interface ergonomique en WPF/WinForms	Simplifier l'expérience utilisateur	Navigation fluide et intuitive	Composants .NET standards

TABLE 1 – Fonctionnalités principales du projet (F1 à F8)

3 Solution proposée

3.1 Données utilisées

Les données seront obtenues via la librairie `YahooFinanceApi` pour chaque ticker saisi :

- prix de clôture quotidiens (ajustés) sur une période d'un an ;
- dernier prix disponible pour la valorisation du portefeuille.

3.2 Modèle mathématique

Rendement espéré :

$$E(R_p) = \sum_{i=1}^n w_i E(R_i)$$

Variance du portefeuille :

$$\sigma_p^2 = w^T \Sigma w$$

Ratio de Sharpe :

$$S = \frac{E(R_p) - R_f}{\sigma_p}$$

Optimisation (Markowitz) :

$$\max S \quad \text{sous les contraintes : } \sum_i w_i = 1, \quad w_i \geq 0$$

3.3 Architecture orientée objet

L'application adopte une architecture modulaire orientée objet, structurée autour des classes principales suivantes :

- **Asset** : représente un actif financier (action, obligation, ETF). *Méthodes principales* : calcul du rendement et de la volatilité.
- **Portfolio** : regroupe les actifs et calcule le rendement global, la volatilité, le ratio de Sharpe et appelle l'optimiseur.
- **DataProvider** : récupère les données financières depuis Yahoo Finance et en déduit les rendements journaliers.
- **Optimizer** : effectue l'optimisation de portefeuille selon Markowitz (poids positifs, somme = 1).
- **UIManager** : gère l'affichage des résultats, des graphes et des statistiques dans l'interface.

4 Planification du projet

Tâche	Durée prévue	Responsable
Étude du sujet et conception	2 jours	Antoine et Adam
Intégration Yahoo Finance	3 jours	Adam
Calculs financiers et modèle Markowitz	3 jours	Antoine
Interface graphique et visualisation	4 jours	Adam
Tests et débogage	2 jours	Antoine et Adam
Documentation et présentation	2 jours	Antoine et Adam
Total estimé	16 jours	—

TABLE 2 – Planification et répartition des tâches

4.1 Évolutions possibles

Si le calendrier le permet, les fonctionnalités suivantes pourront être intégrées :

- ajout de stratégies d'investissement : **Value**, **Momentum** et **Carry** ;
- module d'analyse de performance avancée ;
- sauvegarde des portefeuilles dans une base de données locale.