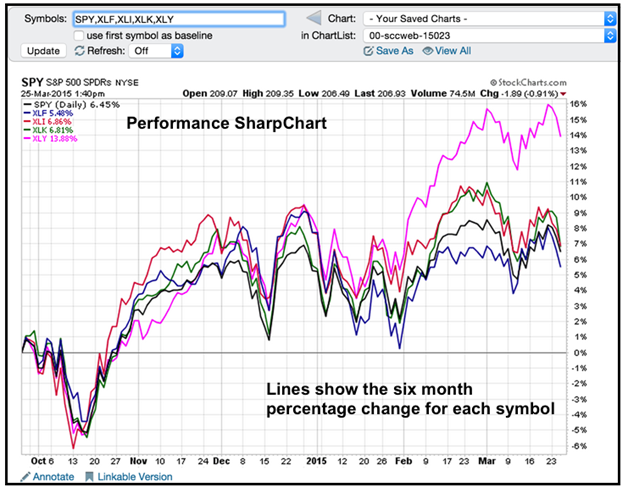
Plot That Line



Auteur : Dario Chasi

Classe : CID3A

Lieu : Vennes

Durée : 24 périodes

Chef de projet : M.Melly

Table des matières

[Spécifications 3](#_Toc175751514)

[Titre 3](#_Toc175751515)

[Description 3](#_Toc175751516)

[Matériel et logiciels à disposition 3](#_Toc175751517)

[Prérequis 3](#_Toc175751518)

[Cahier des charges 3](#_Toc175751519)

[Gestion de projet 3](#_Toc175751520)

[Qualité 3](#_Toc175751521)

[Illustration non contractuelle 4](#_Toc175751522)

[Fonctionnalités requises (du point de vue client) 4](#_Toc175751523)

[Contraintes techniques 4](#_Toc175751524)

[Livrables 4](#_Toc175751525)

[Planification initiale 6](#_Toc175751526)

[Analyse 7](#_Toc175751527)

[Document d’analyse et conception 7](#_Toc175751528)

[Réalisation 8](#_Toc175751529)

[Dossier de réalisation 8](#_Toc175751530)

[Tests 9](#_Toc175751531)

[Dossier des tests 9](#_Toc175751532)

[Conclusion 10](#_Toc175751533)

[Bilan des fonctionnalités demandées 10](#_Toc175751534)

[Bilan de planification 10](#_Toc175751535)

[Bilan personnel 10](#_Toc175751536)

[Divers 11](#_Toc175751537)

[Webographie 11](#_Toc175751538)

[Annexes 11](#_Toc175751539)

# Spécifications

## Titre

Plot that line

## Description

Concevoir un logiciel pour afficher des graphiques sur des données

## Matériel et logiciels à disposition

Un PC ETML

Accès à internet

## Prérequis

Modules de programmation de base ainsi que module 323

## Cahier des charges

## Gestion de projet

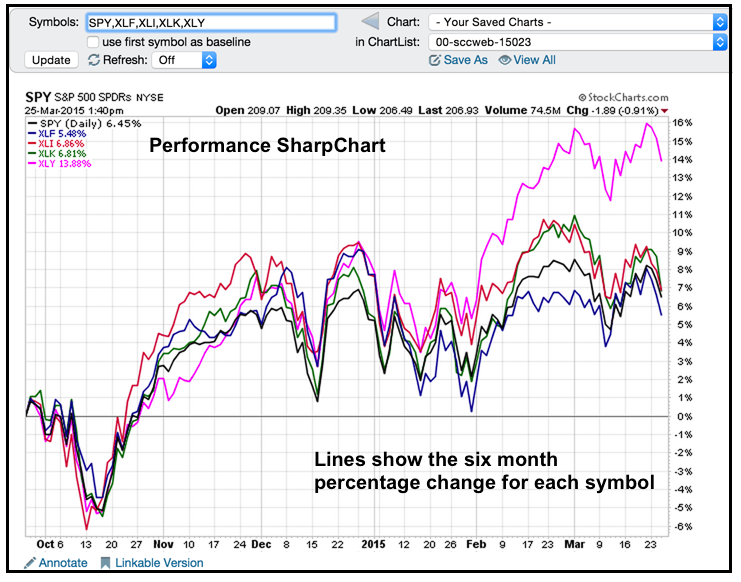
1. Marketplace
2. Planification
3. Journal de travail

## Qualité

1. Réaliser un programme informatique de qualité
   * Organisé (namespace, classes, commit log,…)
   * Compacté (pas de copié/collé,…)
   * Optimisé (utilisation de structures adaptées)
   * Testé (tests unitaires)
   * Commenté
   * Complet (code, script DB, maquettes PDF, éxécutable, …)
2. Prouver que vous êtes digne de confiance lorsqu’on vous confie un projet
   * Journal de travail à jour
   * Pro-activité
     + Poser des questions au client
     + Faire des démonstrations
     + Utiliser un système de versioning de code (GIT)

## Illustration non contractuelle

Ci-dessous, à titre indicatif, un exemple avec un graphique présentant plusieurs courbes pour un axe temporel commun…



## Fonctionnalités requises (du point de vue client)

* 1. Interface graphique pour piloter l’affichage des données
     1. Filtre sur les dates
     2. Minimum 3 courbes à choix
  2. Graphique
     1. Axe X correspond au temps
  3. Données

Choisir un sous ensemble de cryptomonnaies dans [la collection suivante](https://www.kaggle.com/datasets/kaushiksuresh147/top-10-cryptocurrencies-historical-dataset) ou toute autre source de données contenant des informations avec une information temporelle…

### Contraintes techniques

* Utiliser LINQ (pas de boucle for)
* Ajouter une surcouche en utilisant les extensions du langage
* Librairie graphique à choix (forms, maui, uno, wpf, fna, …)
* Librairie pour présenter les données à choix (<https://scottplot.net/> par exemple)

## Livrables

Il n’y a qu’un livrable : une release Github avec le code source et à laquelle sont attachés :

1. Rapport PDF contenant
   1. Introduction
   2. Planification
   3. Rapport de tests
   4. Journal de travail
   5. Chapitre explicatif de l’usage fait de l’IA dans ce projet

# Planification initiale

Une image contenant texte, nombre, mots croisés, capture d’écran

Description générée automatiquement

# Analyse

## Document d’analyse et conception

# Réalisation

## Dossier de réalisation

# Tests

## Dossier des tests

# Conclusion

## Bilan des fonctionnalités demandées

Toutes les fonctionnalités demandées sont présents, tel quel les axes X et Y, le code commenté, des test unitaires, le code présent sur github, des fonctionnalités comme

## Bilan de planification

## Bilan personnel

# Divers

## Webographie

[ScottPlot 5.0 Cookbook](https://scottplot.net/cookbook/5.0/)

# Annexes