

|  |
| --- |
| Rapport projet JAVA |
| Hugues Bégeot  Camille Briand  Célia Bunouf  L3 Int – Groupe 4 |



Table des matières

[Introduction 3](#_Toc5872824)

[Les diagrammes conceptuels 3](#_Toc5872825)

[Justification des choix d’implémentation 3](#_Toc5872826)

[Explications de la gestion du format CSV 3](#_Toc5872827)

[Diagrammes Statistiques [JFreeChart] 3](#_Toc5872828)

[Conclusion 3](#_Toc5872829)

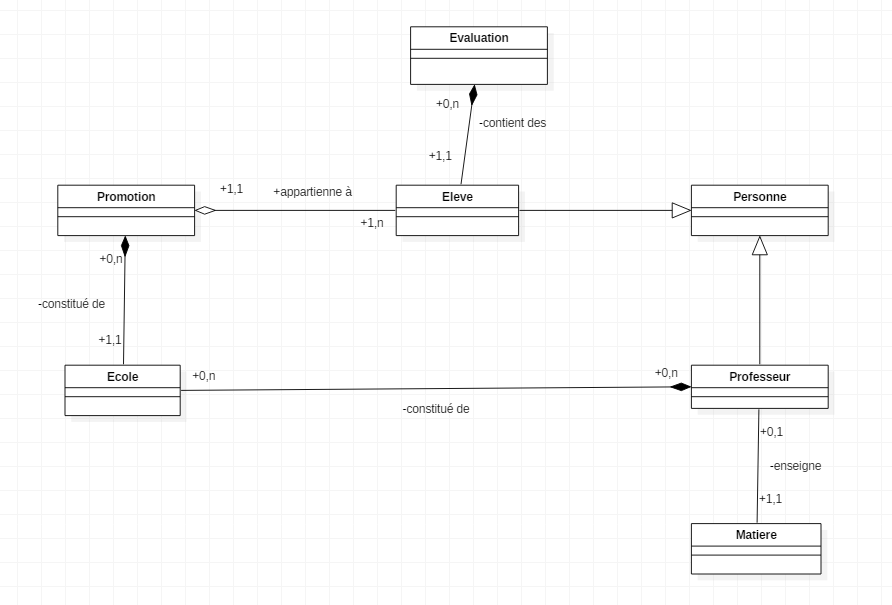
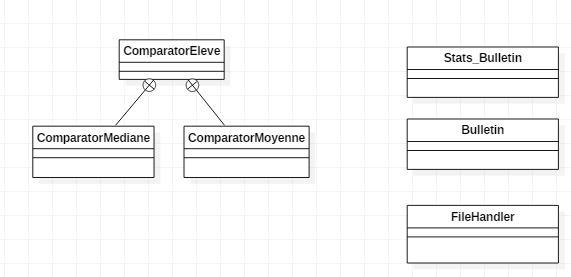
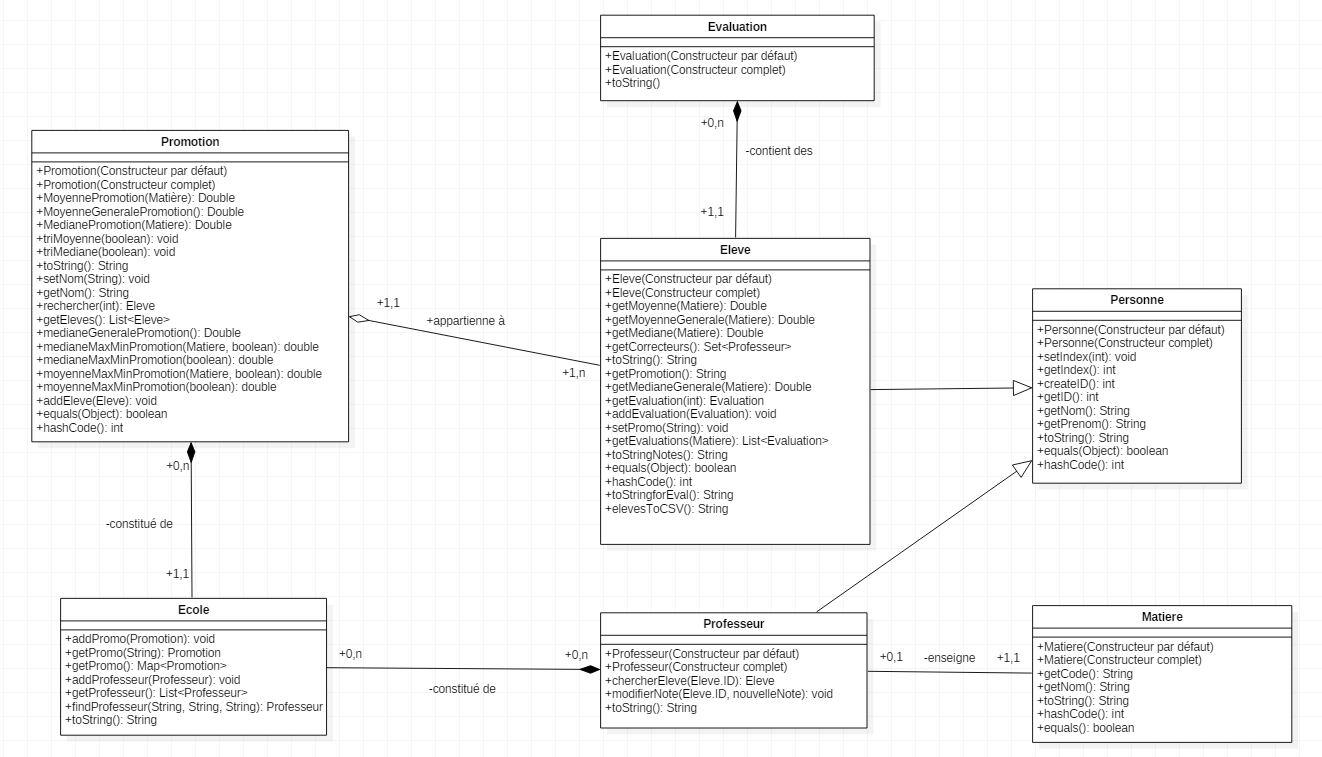
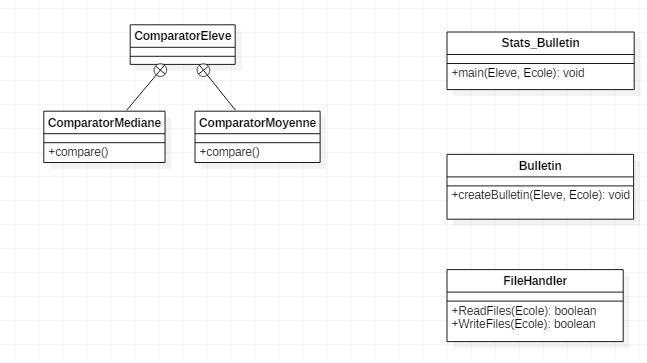
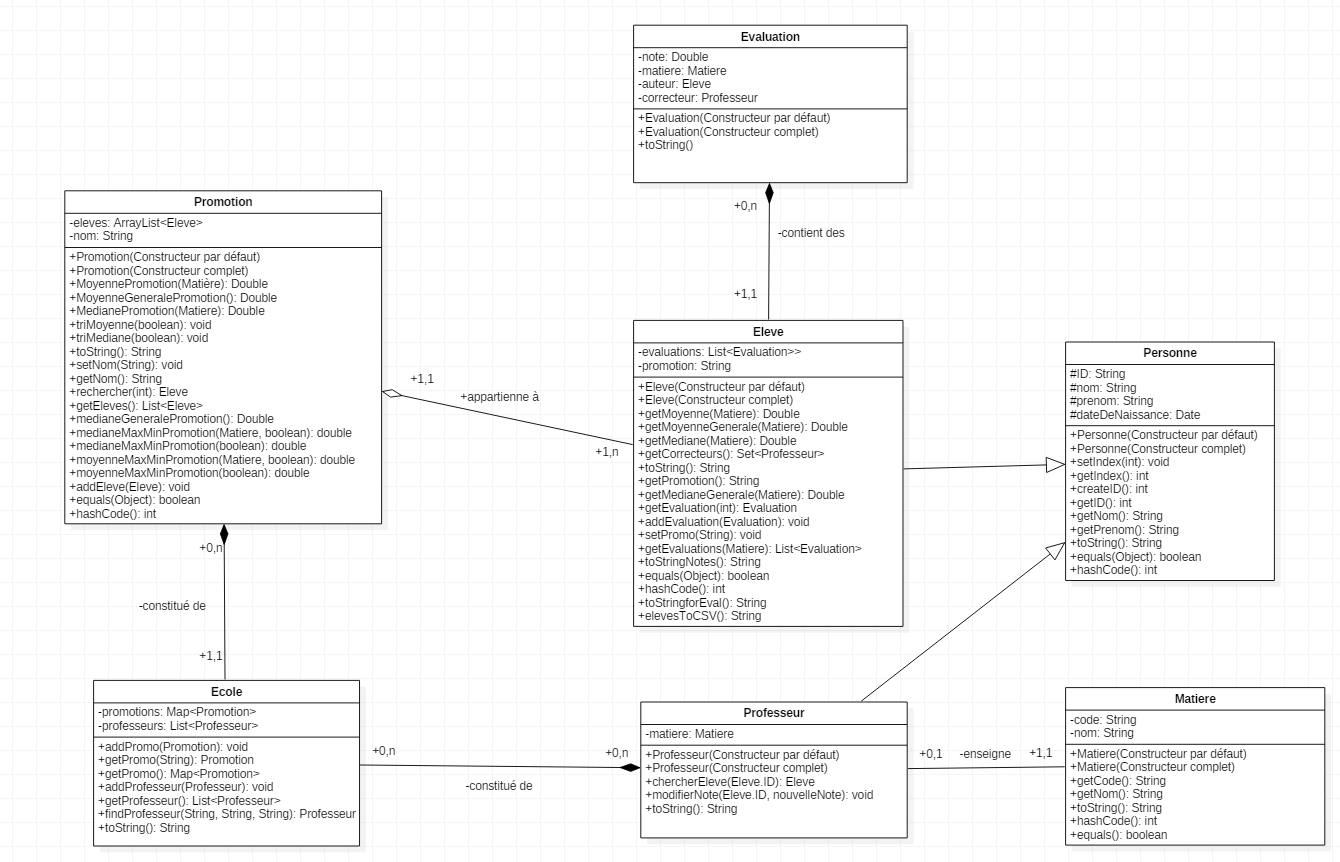
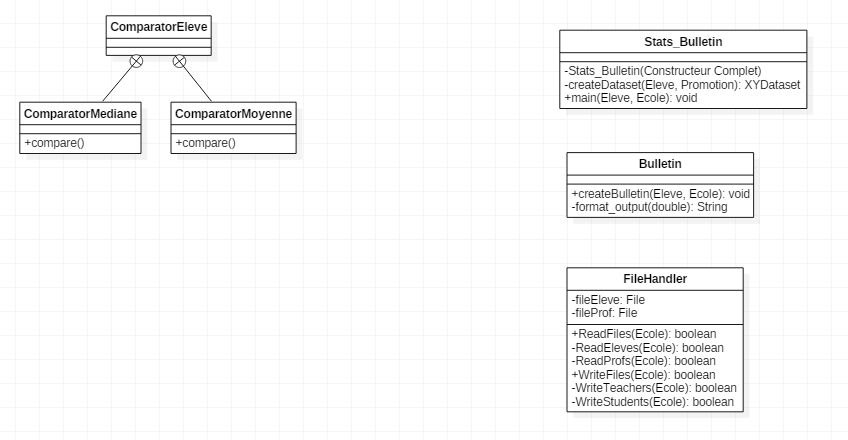
[Annexes 3](#_Toc5872830)

# Introduction

Dans le cours de JAVA, nous avions pour projet de gérer une école avec des professeurs et plusieurs promotions possédant des étudiants. L’objectif final du projet est de créer une interface graphique permettant aux professeurs d’évaluer les étudiants et aux étudiants de consulter leurs notes. Le bulletin d’un étudiant peut également être créer.

Le projet doit être réalisé en JAVA, avec une utilisation d’outils graphiques comme JFreeChart pour diverses statistiques ainsi que de fichiers CSV pour sauvegarder les données de l’école. L’interface à été réaliser à l’aide de JavaSwing.

# Les diagrammes conceptuels

* 1er diagramme : relation et cardinalité
* Classes fondamentales
* Classes fonctionnelles
* 2ème diagramme :
* Classe fondamentale
* Classe fonctionnelle
* 3ème diagramme :
* Classe fondamentale :
* Classe fonctionnelle :

# Justification des choix d’implémentation

# Analyse Fonctionnelle Générale :

Le projet doit pouvoir gérer des étudiants, des notes obtenues aux évaluations et des professeurs. C’est pour quoi nous avons basé le projet sur sept classes principales : une évaluation, une matière, une promotion, un élève et un professeur, tous deux héritant de la classe personne. Nous avons également fait le choix de créer une classe école. Cette classe école a pour but de regrouper tout ce qui constitue une école c.-à-d. un ensemble de professeur et un ensemble de promotion. Chaque promotion est constituée d’un ensemble d’élève.

Nous avons fait en sorte qu’un professeur n’enseigne qu’une seule matière ainsi quand ce professeur ajoute une note, l’évaluation est ajoutée dans sa matière.

# Analyse Fonctionnelle Détaillée :

# Explications de la gestion du format CSV

Les deux principales classes de notre implémentation sont les élèves ainsi que les professeurs. On a ainsi choisi de représenter les données sous deux fichiers CSV : un contenant les étudiants ainsi que leurs évaluations et un autre contenant les professeurs.

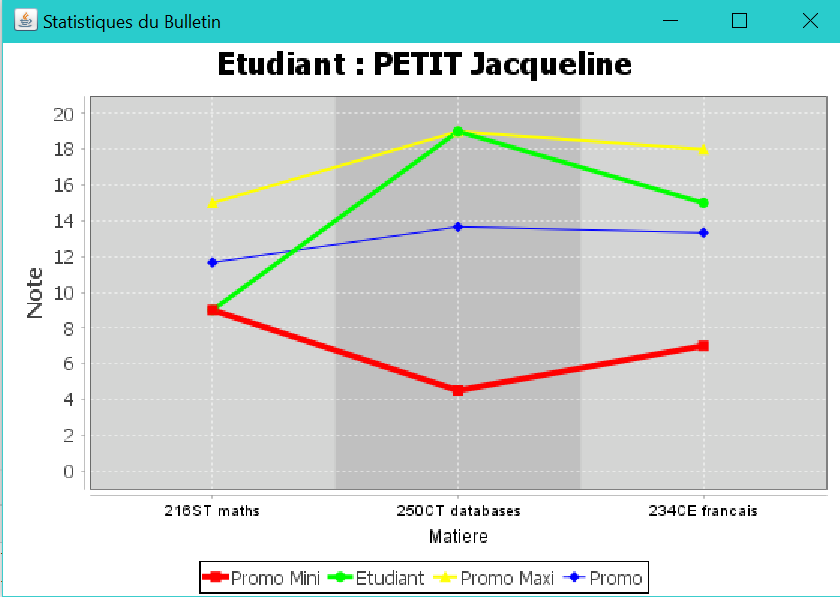
Le fichier CSV des professeurs contient ainsi le nom, prénom, date de naissance, ID, mot de passe ainsi que la matière enseignée par le professeur. Lors de la lecture du fichier les éléments Professeur sont créés et ajoutés à l’école.

Le fichier CSV des élèves contient le nom, prénom, date de naissance, ID, mot de passe, promotion, ainsi qu’autant de colonne nécessaire en fonction du nombre d’évaluations présentent pour l’étudiant. Dans cette base de données, une évaluation est représentée par la note et l’ID du professeur correcteur. Il a été choisi de l’ID du professeur correcteur car grâce à celui-ci nous pouvons retrouver la matière de l’évaluation étant donné que le professeur n’enseigne qu’une seule matière.

L’avantage des fichiers CSV ici par rapport aux bases de données telles que MySQL, est que le nombre de colonne est variable en fonction des élèves. Ce n’est pas une structure fixe.

# Diagrammes Statistiques [JFreeChart]

Sur les statistiques présentes sur le graph, nous avons décidé de représenter la moyenne de l’étudiant à côté de la moyenne générale de la promotion et des moyennes minimales et maximales de la promotion.



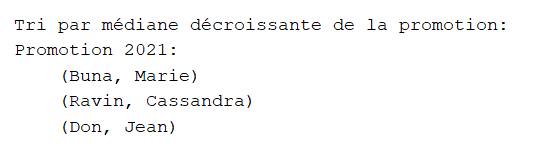
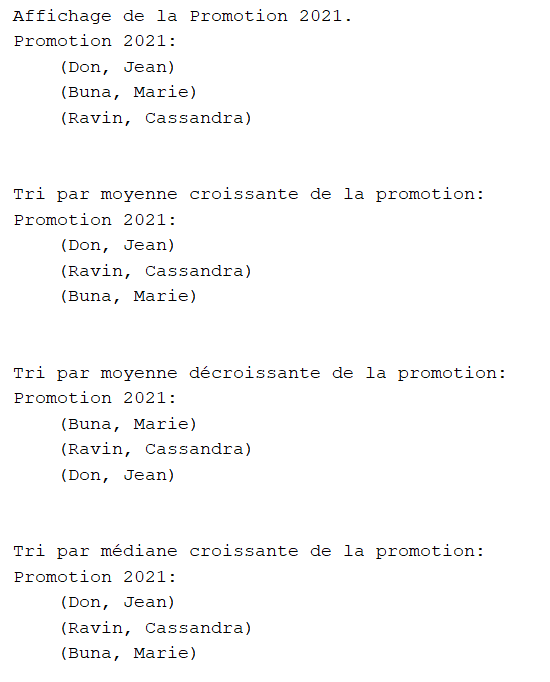
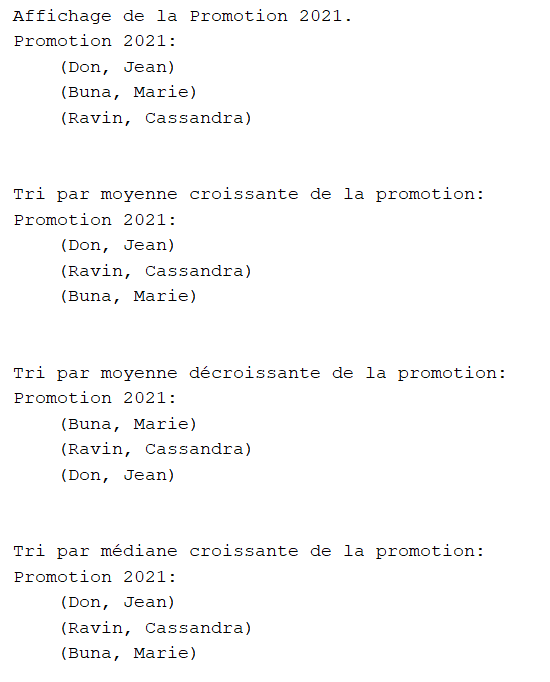
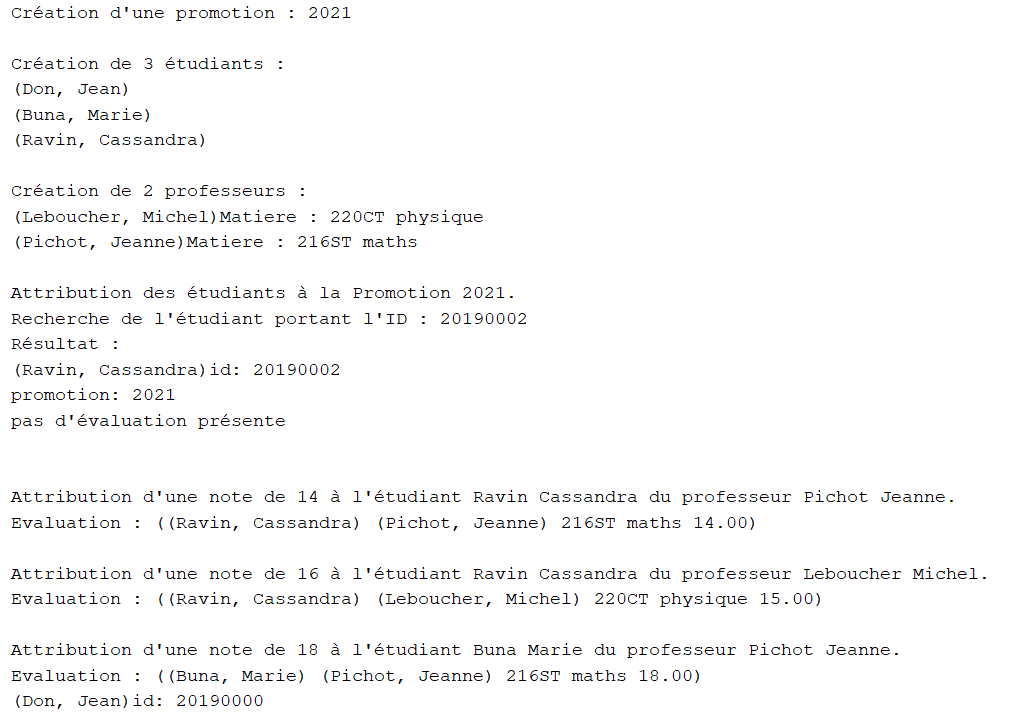
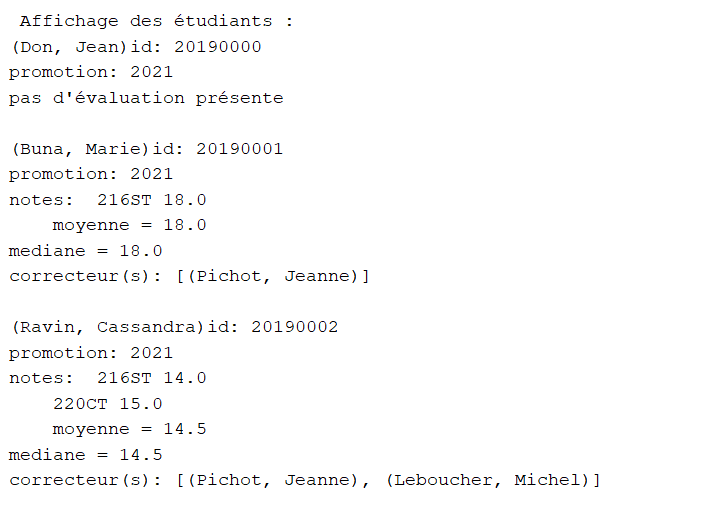
# Conclusion

En conclusion, nous avons réussi à créer un programme capable de gérer le fonctionnement des notes d’une école ainsi que leur étudiants et professeurs. Ce projet nous a permis d’utiliser un nouvel outil, JFreeChart, pour la création de différents graphs et également d’approfondir nos connaissances dans l’utilisation du JavaSwing.

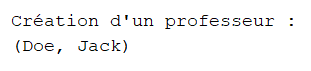
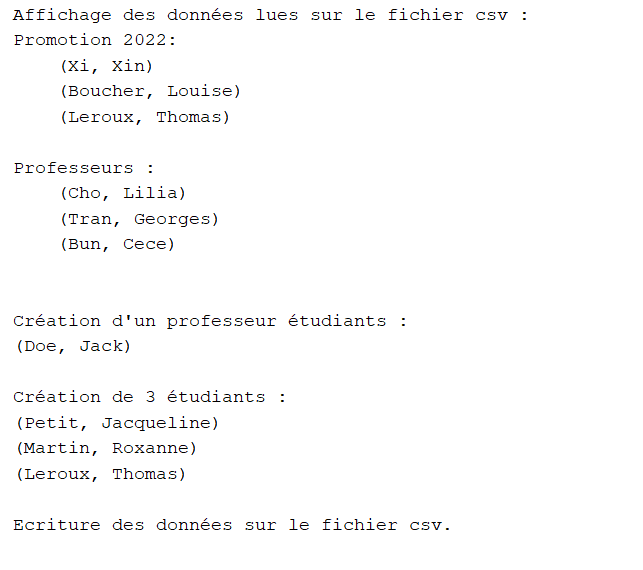
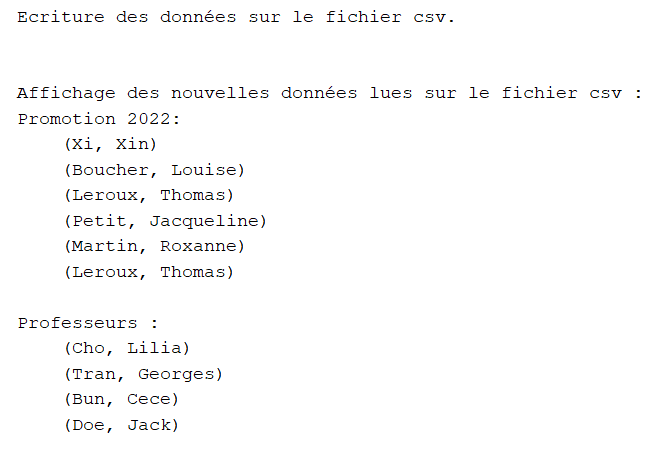
# Annexes

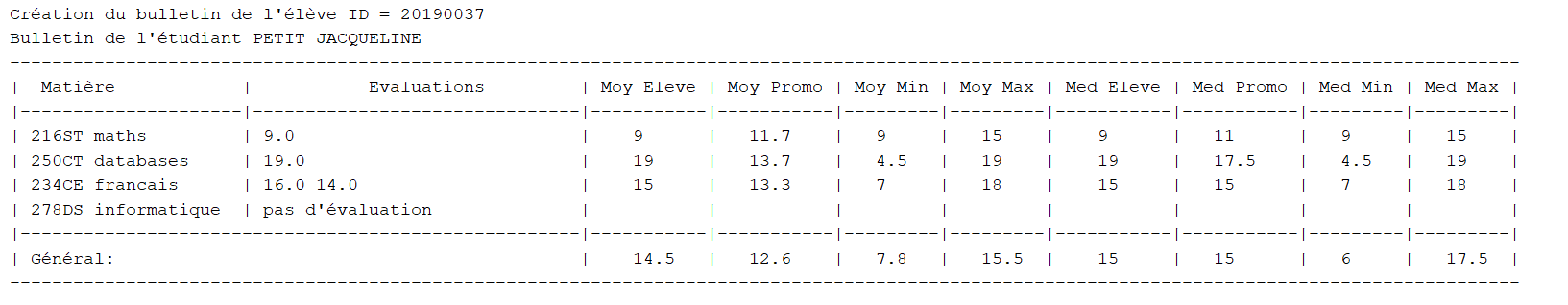
Source JFreeChart :

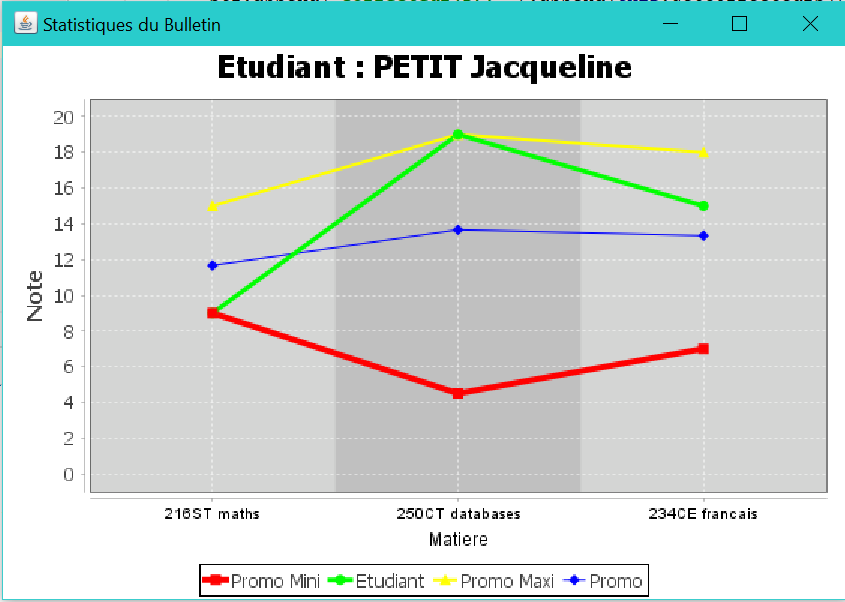
Capture d’écran Version 1 :



Capture d’écran Version 2 :



Capture d’écran Version 3 :



Capture d’écran Version 4 :