**PROJET DE GESTION DES ÉTUDIANTS**

Cette application web réalisée en 2017, consistait en la gestion des étudiants au sein de l’Institut Supérieure du Sahel : cas du département d’Informatique et de Télécommunication.

À l’aide d’un certain nombre d’outils de mesure de la qualité du code source, nous allons à présent critiquer le code source de cette application. Il est à noté que cette application a été réalisé avec les langages de programmation PHP, HTML/CSS. Par conséquent, les outils que j’ai eu à utiliser tout au long de cette étude sont les suivants :

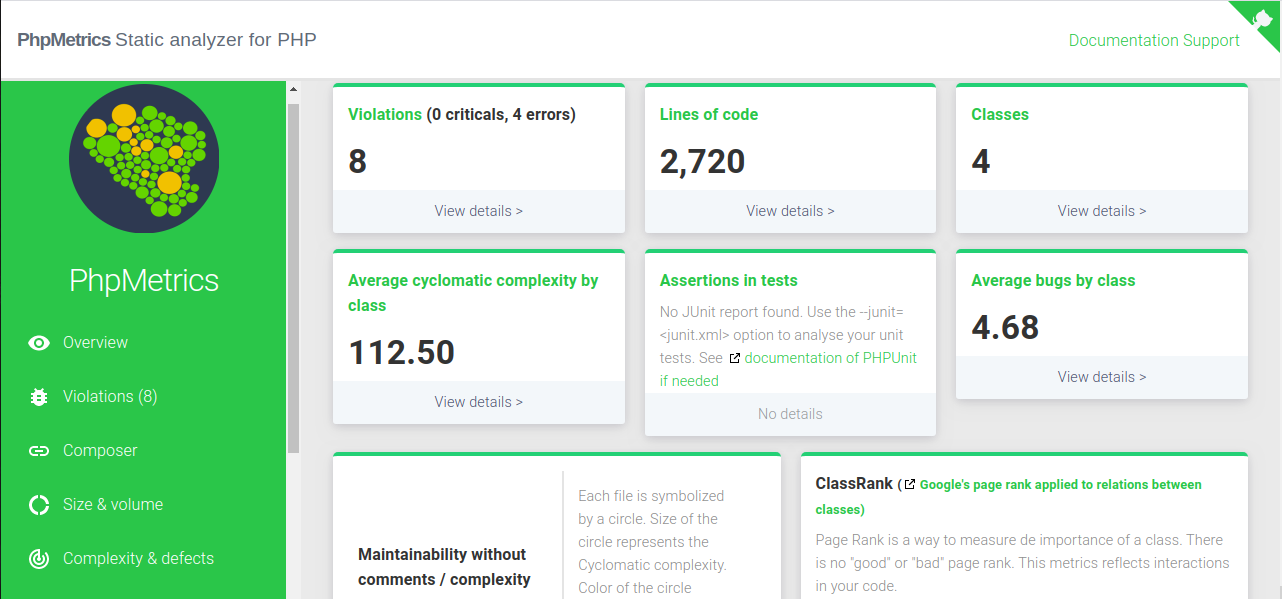
* **PHP Code Sniffer:** Vérification de la qualité de son code ;
* **PHPCpd:** Permet de détecter la duplication de code ;
* **PHP Mess Detector:** Permet d’évaluer la qualité d’un code en détectant entre autres les variables et méthode non utilisés ;
* **Sonar:** Permet d’évaluer et suivre la qualité du code ;
* **PHP Metrics:** Permet d’avoir des statistiques sur les mesures du code source.

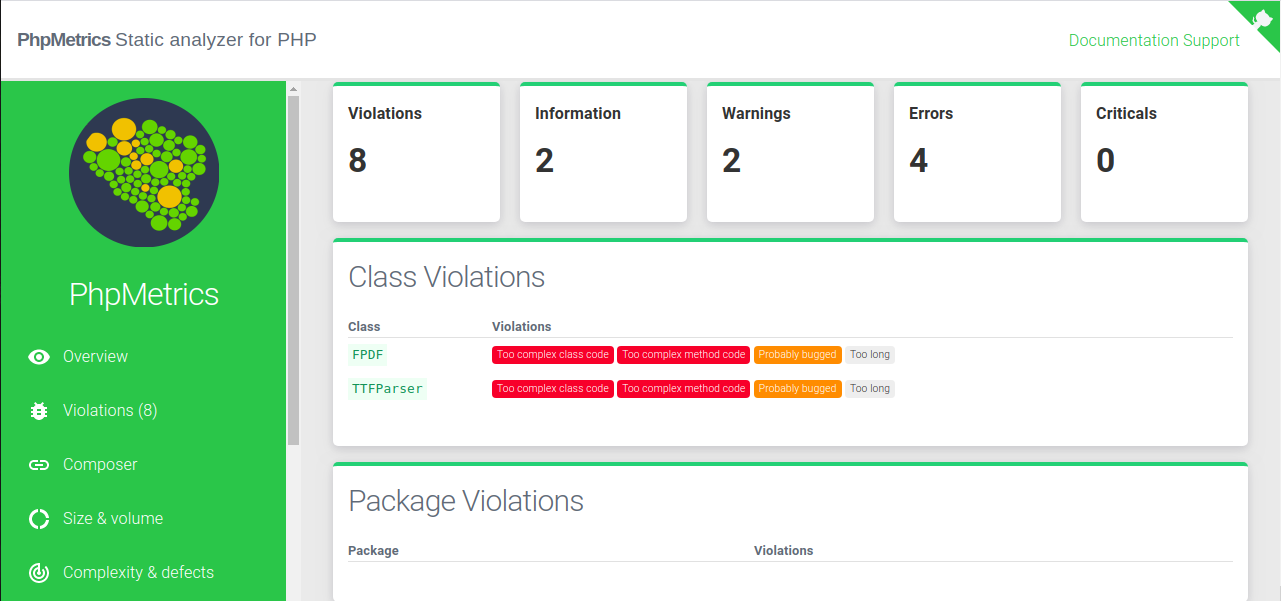
# RÉSULTAT DE L’OUTIL D’ANALYSE PHP METRICS

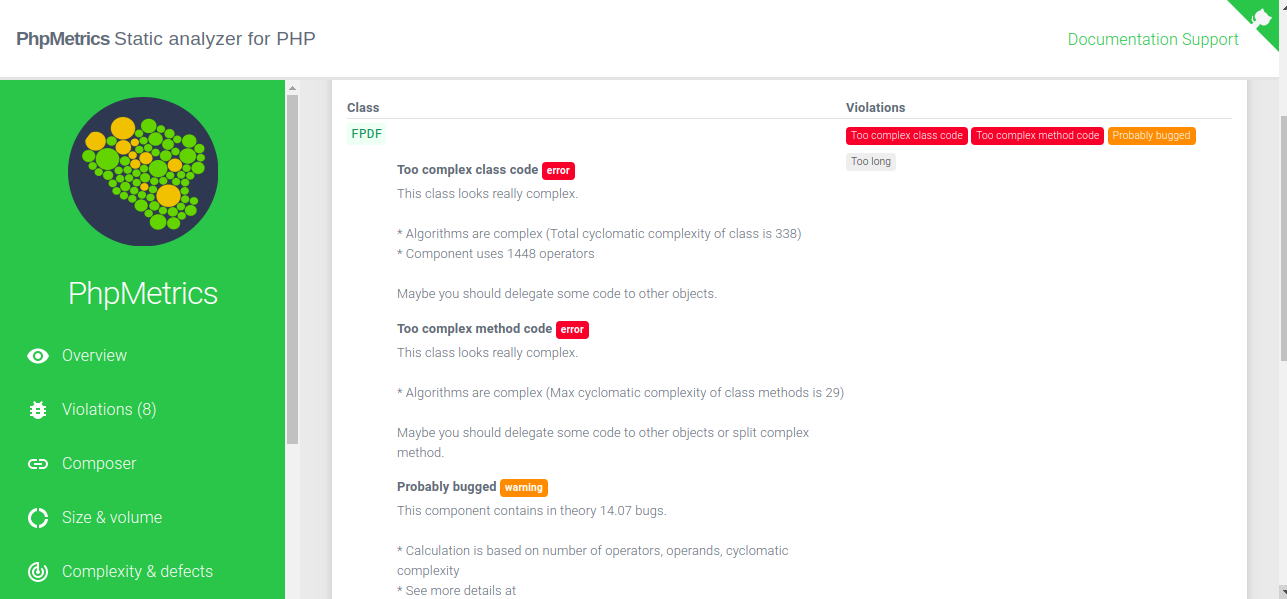
Comme nous l’avons dit plus haut, l’outil d’analyse PHP METRICS permet d’avoir des statistiques sur les mesures du code source. Il nous renvoie entre autres :

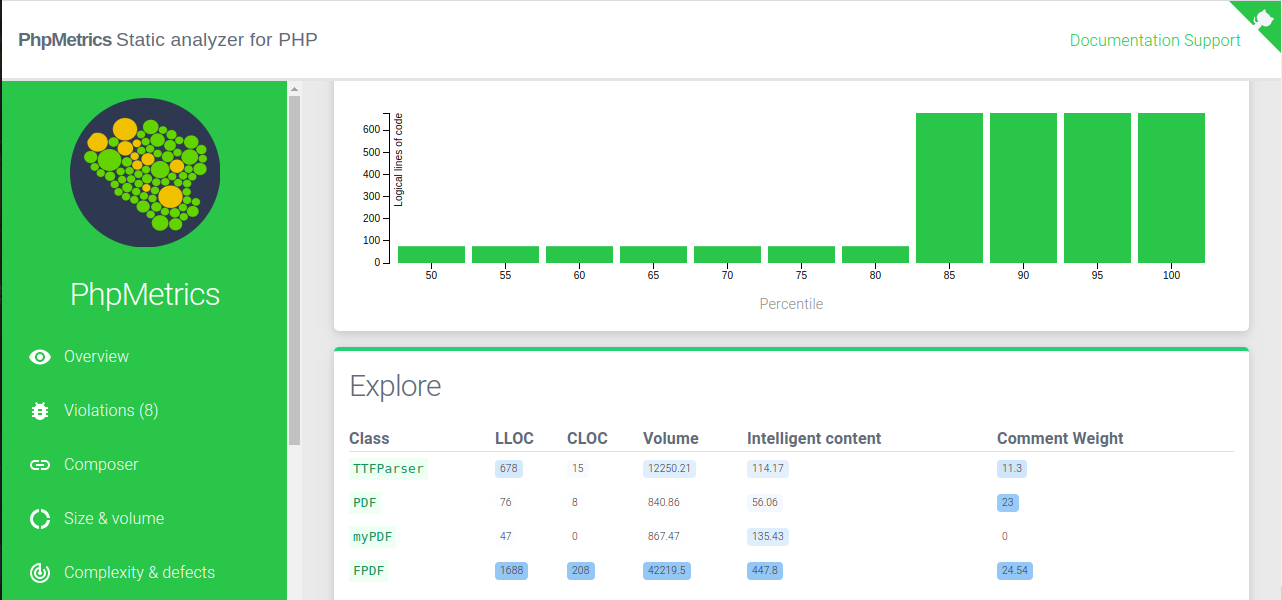
* Le nombre de violations ;
* Le nombre de lignes de code ;
* Le nombre de classe utilisée ;
* Le taux de complexité présente au sein de chaque classe ;
* L’utilisation des tests d’assertions ;
* Le taux de bug présent au sein de chaque classe.

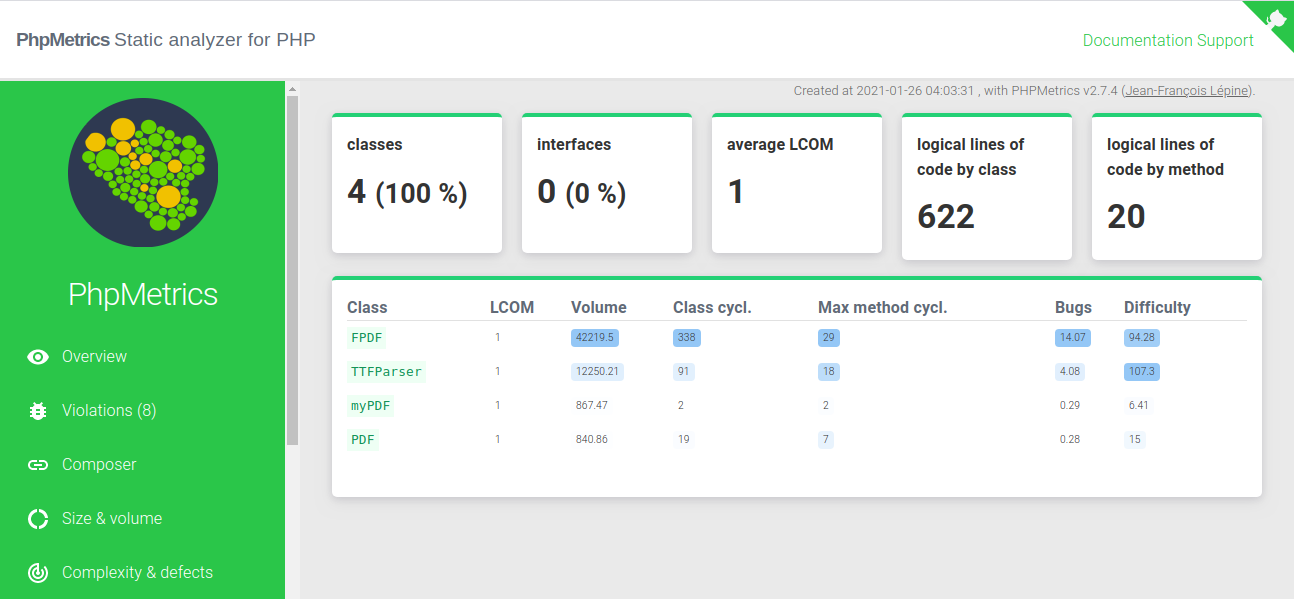
Il est à noté qu’a chaque résultat on y ajoutera des information plus détaillé.

**Figure 1 :** Analyse globale de l’application avec PHP METRICS

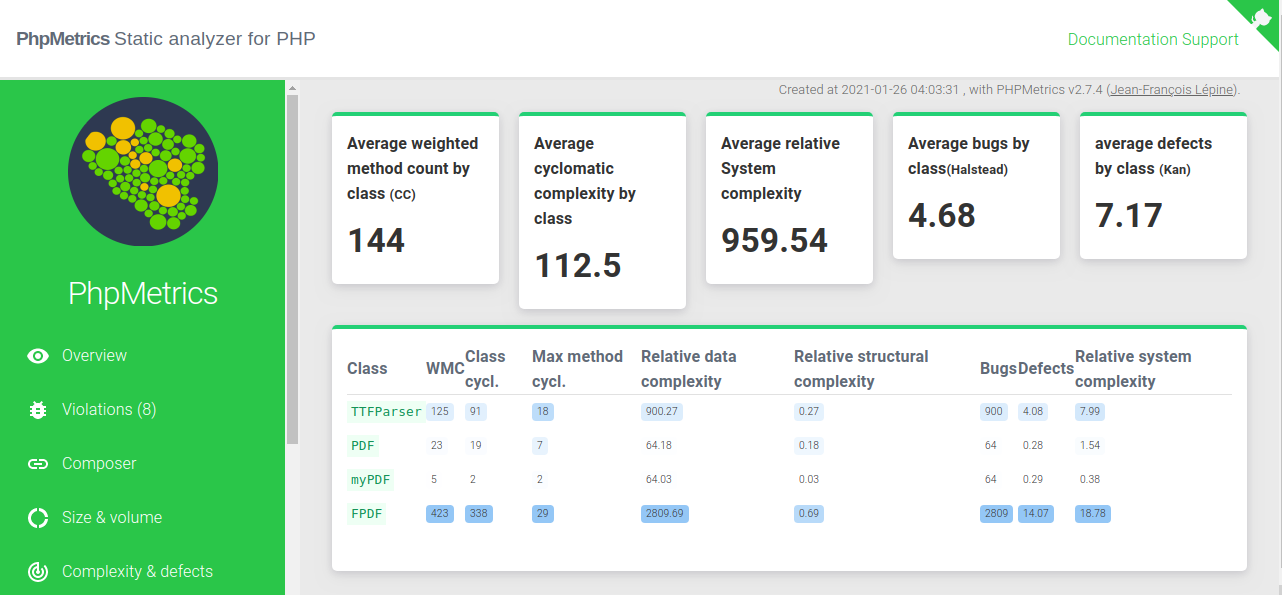
**Figure 2 :** Détails sur le nombre de violation

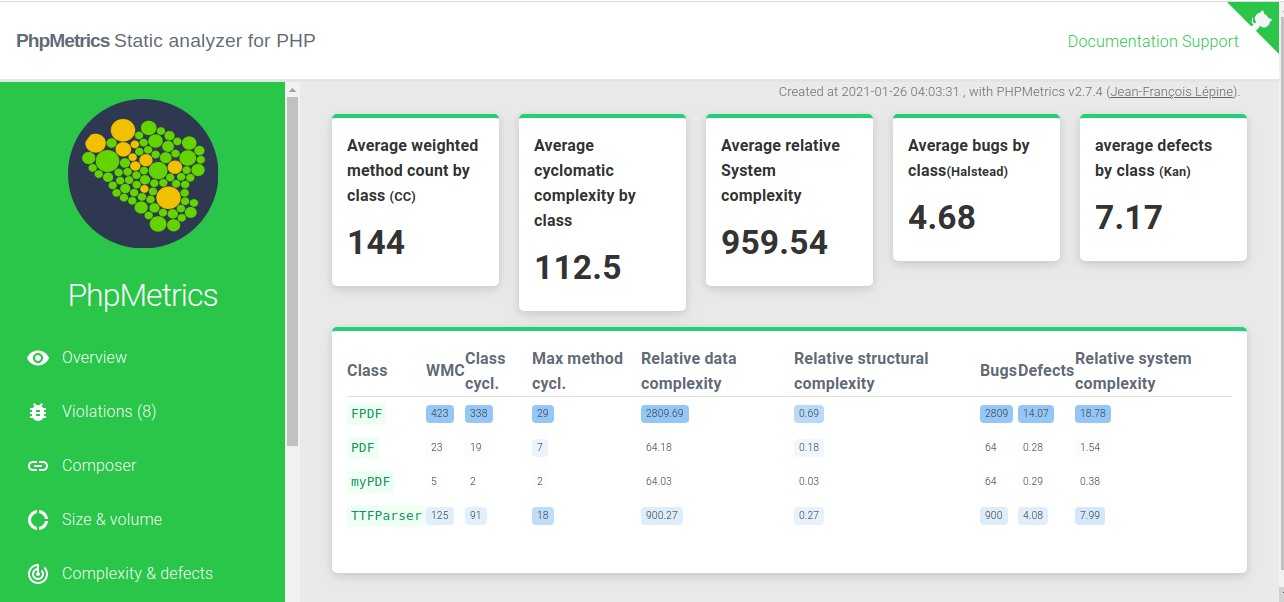
**Figure 3 :** Présentation des violations présentent dans la classe FPDF

**Figure 4 :** Détail sur les lignes de code ; Percentile distribution of logical lines of code by class

****

**Figure 5 :** Détail sur les classes

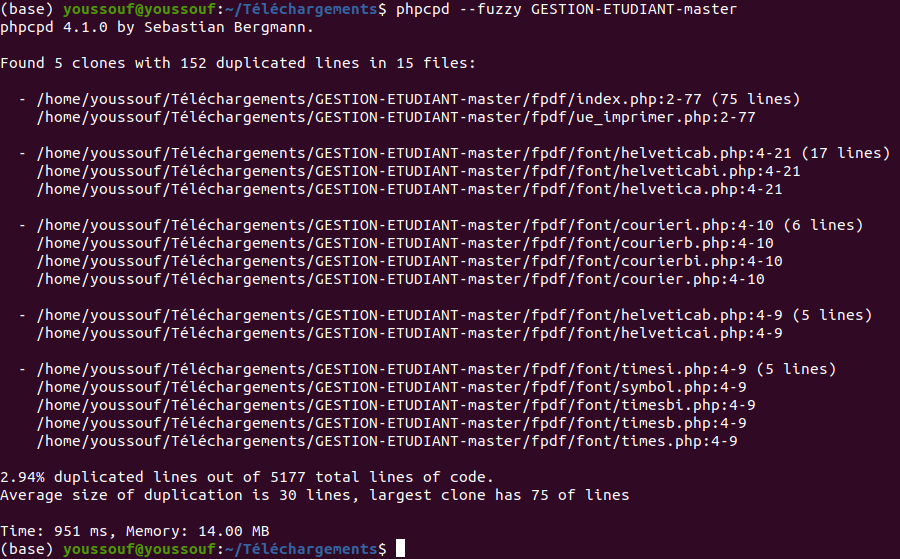
**Figure 6 :** Détail sur le [Average cyclomatic complexity by class](http://complexity.html/)

**Figure 7 :** Détail sur le taux de bugs présente dans chaque classe

Il est à noté que cette outil renvoie par analyse le volume de ligne de code présente dans chaque classe, le poids des commentaires ; en bref un certain nombre d’informations.

# RÉSULTAT DE L’OUTIL D’ANALYSE PHPCpd

Le principal avantage de cette outil est qu’il nous renvoie les informations sur le nombre de ligne de code qui a été dupliqué, mais également sur le nombre de ligne de code présent dans le projet. Pour ce projet, voici un aperçu du résultat qui a été renvoyé.



**Figure 8 :** Résultat de l’analyse avec PHPCpd

Nous constatons que avec cette outil, notre projet contient 5177 lignes de code, avec un taux de 2.94 % de lignes de code qui a été dupliquées. Cet outil nous renvoie les fichiers au sein desquels ces lignes de code ont été dupliquées, la taille moyenne de la duplication qui est de 30 lignes, mais également la taille du plus grand clone qui est de 75 lignes.