RAPPORT DE TEST

Objectifs : L'objectif de ce document est de montrer les tests effectués

Sommaire :

INTRODUCTION

Ce présent document fournis un rapport de test effectué sur le projet intitulé [HospitalManagemeTest] dans le cadre universitaire.

HISTORIQUE DES VERSION

Date de mise à jour : Mercredi le 10 Mai 2023

Auteur : ADJETEY Michel

Version : 1.0

Description : rapport de test

Client : M. JABER

Source : https://github.com/JaberAzWil/HospitalManageTEST

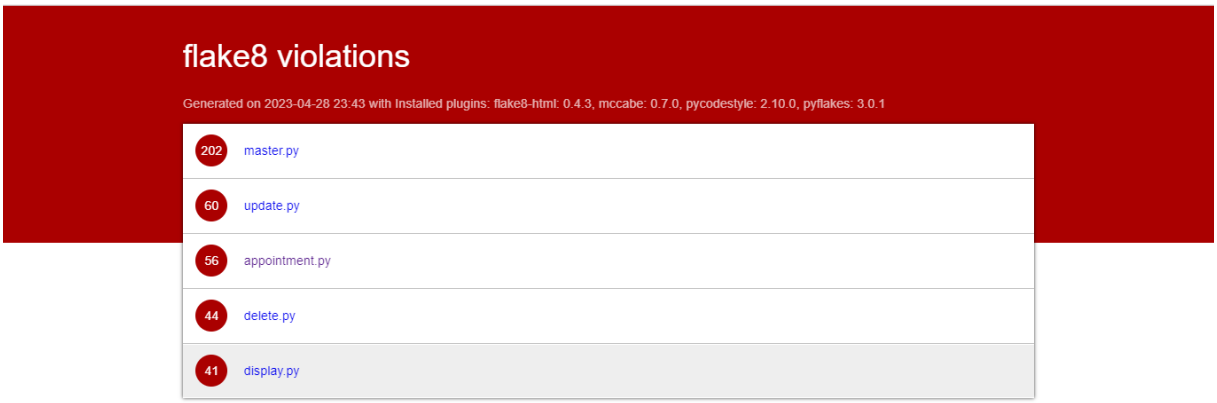
I. TEST SUR LA QUALITE DU CODE AVEC Flake8

1. Description du test

Le but du test est de vérifier que le logiciel respecte bien les règles pep8.

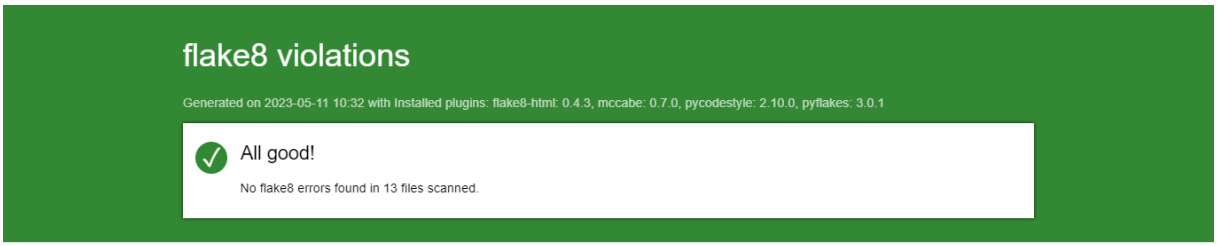
Le test consiste à analyser entièrement le code et dans le but de noter l'usage des bonnes pratiques en programmation Python. Pour ce cas de figure nous utiliserons le package Flake8 qui va se charger dans un premier temps d'analyser le code et par la suite relever le mauvaises pratiques qui seront consigner dans un rapport.

2. Résultat et analyse



Nous avons au total 403 violations et après limitation des caractères par ligne à 115 nous nous retrouvons maintenant avec 334 violations.

<u>Après correction</u> :



3. Bilan

Le code ne respecte pas vraiment les bonnes pratiques en programmation Python.

On déduit que le code est de mauvaise qualité.

II. TEST DE PERFORMANCE

1. Description du test

Le but du test est d'analyser les performances de notre application. Il est important de noter que la mesure des performances d'une application de desktop peut être plus difficile que pour une application web, car il y a souvent plus de facteurs à prendre en compte, tels que la charge du système, l'utilisation des ressources et les limitations matérielles. Il est donc important de tester votre application dans une variété de conditions pour obtenir des résultats précis et représentatifs.

2. Résultat et analyse
3. Bilan

III. TEST UNITAIRES ET TEST FONCTIONNELS

1. Description du test

Le but du test est de vérifier que le logiciel respecte bien ses spécifications.

Le test consiste à vérifier que le logiciel réalise les fonctions attendues et ce correctement. Pour cela nous allons écrire les tests unitaires et fonctionnels afin de tester les différentes fonctions et fonctionnalités et vérifier la concordance des résultats après exécution du test.

Il existe une petite nuance entre ces deux types de test :

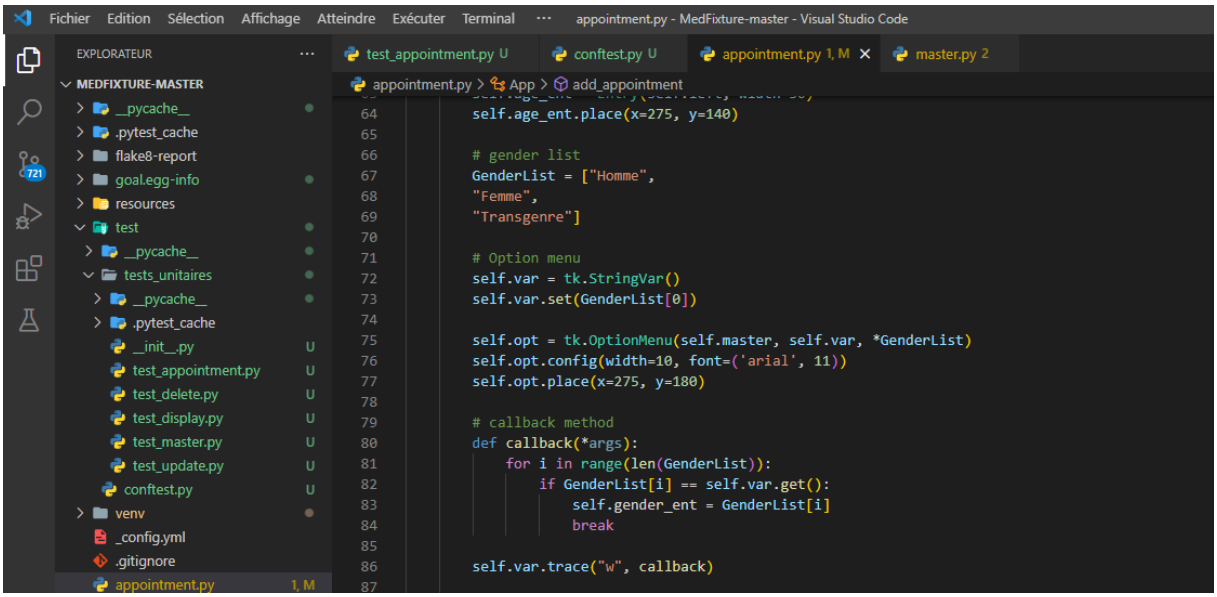
Les tests unitaires sont utilisés pour tester une unité de code (fonction, méthode, classe) de manière isolée, en vérifiant si elle retourne les résultats attendus pour des entrées spécifiques. Ils sont généralement écrits dans le but de s'assurer que chaque unité de code fonctionne correctement avant de l'intégrer à l'application. (Ex : Il teste une fonction spécifique de manière isolée, en fournissant des valeurs d'entrée spécifiques et en vérifiant si le résultat obtenu est celui attendu).

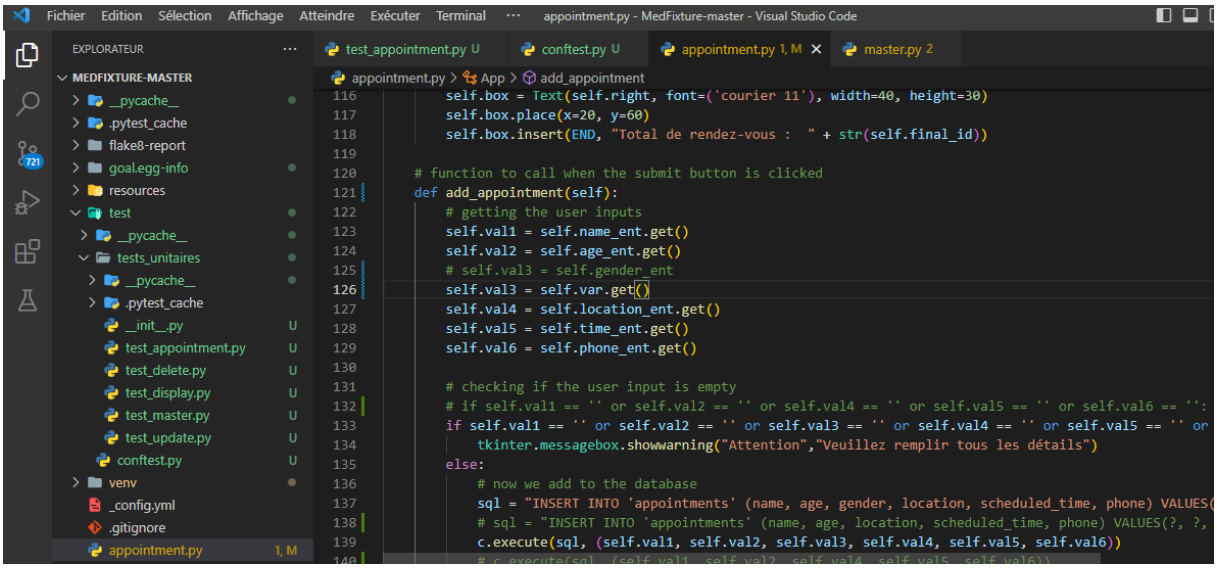
Les tests fonctionnels, en revanche, sont utilisés pour tester l'application dans son ensemble, en vérifiant si toutes les fonctionnalités de l'application fonctionnent correctement ensemble. (Ex : Le test fonctionnel vérifie si l'application fonctionne

correctement lorsqu'elle est utilisée par un utilisateur, en vérifiant toutes les possibilités possibles).

2. Résultat et analyse

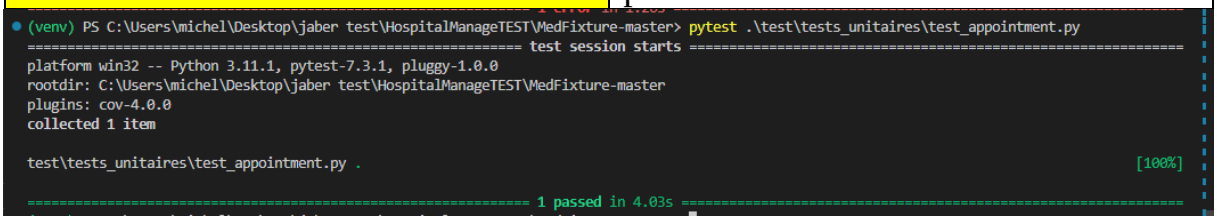
Déjà je tiens à souligner le fait que le code est truffé d'erreur ce qui entraine des potentiels bugs





Test sur le fichier appointment.py	
Add_appointment()	Permet de prendre un nouveau rendez-vous
Résultat attendu	Résultat observé
"Réussie", "Rendez-vous pour " + PATIENT + " a été créer"	ERREUR : gender_ent is not defined

Analyse	La variable gender_ent n'est pas définie dans le programme
Remarque	Les résultats attendus ne concordent pas avec les résultats observés.



Test sur le fichier delete.py	
test_search_db ()	Permet de récupérer les infos d'un utilisateur
Résultat attendu	Résultat observé
Aucun	Ok
Analyse	OK
Remarque	Le test de cette fonction est réussi.
test_delete_db()	Permet de supprimer un rendez-vous dans la base de données
Résultat attendu	Résultat observé
Analyse	OK
Remarque	Le test de cette fonction est réussi.

Test sur le fichier display.py	
test_search_db ()	Permet de récupérer les infos d'un utilisateur
Résultat attendu	Résultat observé
Aucun	Ok
Analyse	OK
Remarque	Le test de cette fonction est réussi.

Plus on avançait dans les tests cela devenais de plus en plus complexe de réaliser des tests corrects sans aucune indication (absence de documentation dans documentation).

3. Bilan

TEST UNITAIRES : à l'exception du fichier appointment.py, les tests unitaires effectués sont tous validés ce qui atteste que les résultats attendus correspondent bien aux résultats observés.

TEST FONCTIONNELS : Liés au manque de documentations sur le code, aucun test fonctionnel a été effectué mais au vu de la qualité du code, le manque d'exception au niveau des différentes fonctions ainsi que l'absence de vérification des informations saisies dans les formulaires, nous pouvons déduire que les tests fonctionnels sont en totalité un échec car ils seront tous validés s'il avaient été implémenté.

En sommes les tests unitaires sont validés à environs 70% et les tests fonctionnels sont un échec à plus de 50%.

IV. AUTRES

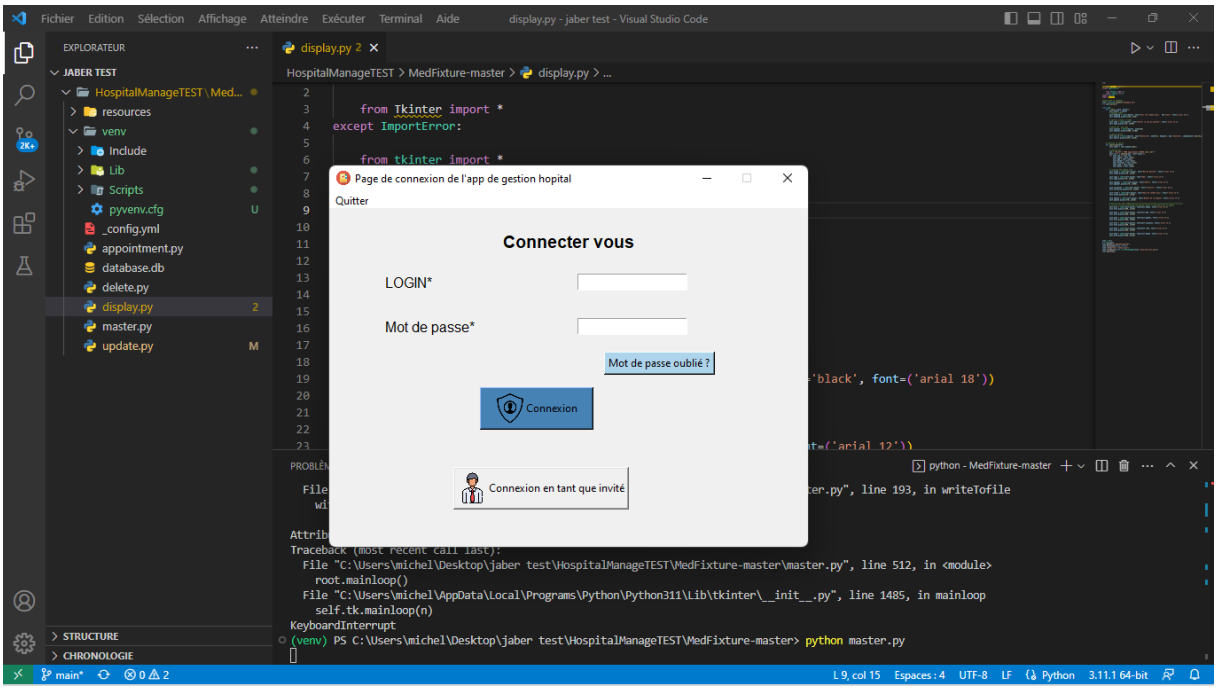
Le premier constat est que le code contient des commentaires plus ou moins explicite.

Certains codes se répètent plusieurs fois comme la fonction search_db() qui est redéfinie dans 2 fichiers différents.

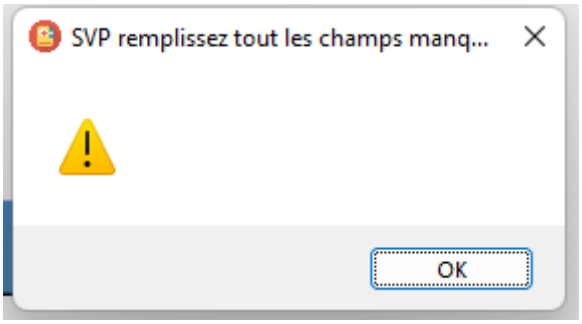
Pas d'architecture MVC

Sur les interfaces les boutons ne fonctionnent pas correctement.

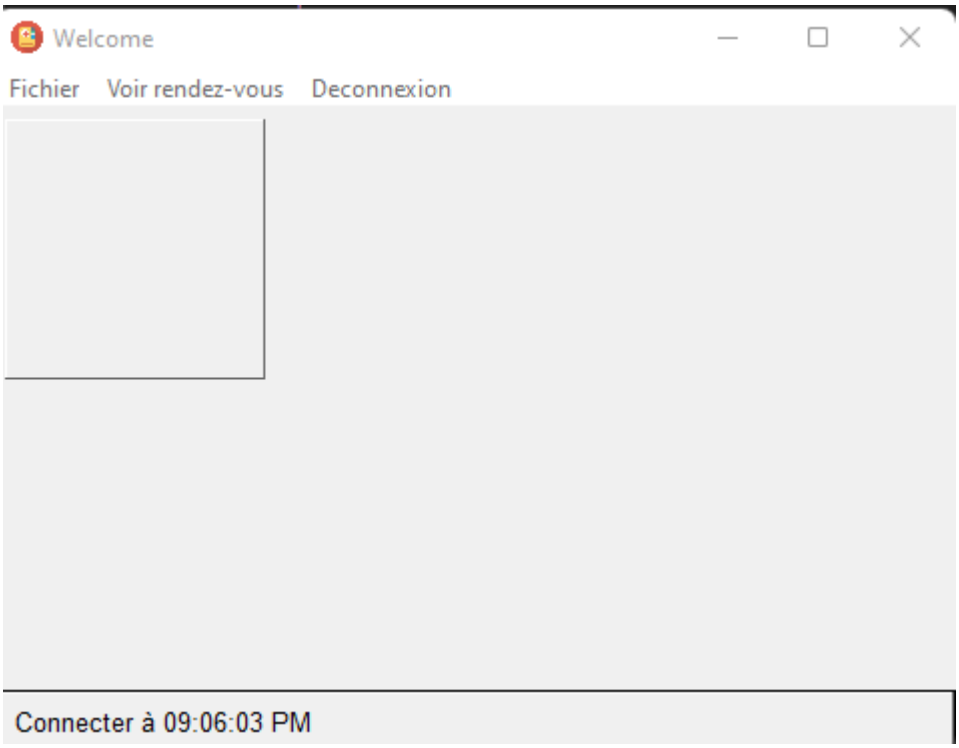
La totalité des pages ne s'ouvrent pas en prenant en compte tout l'espace disponible (mode plein écran)



- Le bouton de connexion ne fonctionne pas
- Le popup qui apparait en cas de mauvaise information de connexion de l'utilisateur n'est pas explicite



- Dashboard inexistant une fois la connexion établie



- Pas vraiment d'utilité de se déconnecter en mode invité
- Faille très grave car la personne connecté en mode invité peut consulter certains des patients

CONCLUSION GENERALE

L'application fonctionne partiellement, certaines commandes ne répondent pas correctement (Création d'un nouveau rdv) nous avons rencontré un problème lors des tests de performances et le principal problème c'est le manque de documentation qui nous a fait perdre du temps à force de tâtonner dans l'optique de réussir à exécuter le code.