Un peuple-Un but-Une foi

UFR des Sciences Et Technologies

Département

Informatique

Filière

Informatique option Génie Logiciel

PROJET

MESURE QUALITE ET PERFORMANCE LOGICIELLE

Membres :

Mouhameth LO

Babacar GUEYE

Modou Bakh DIOP

Professeur:

Monsieur Mansour DIOUF,

Enseignant à l'université Iba Der THIAM de Thies

Année académique : 2022-2023

PLAN

I- Introduction :

Le projet de gestion de projet est une application innovante conçue pour planifier, suivre et gérer les différentes étapes d'un projet, qu'il s'agisse du développement d'un logiciel par exemple. Cette application propose des fonctionnalités essentielles pour garantir le succès de tout type de projet.

II- Présentation du Projet :

1- Description :

Le projet de gestion de projet est une application développée pour répondre aux besoins croissants de planification, de suivi et de gestion des projets dans divers domaines tels que le développement de logiciels. Cette application inclut des fonctionnalités avancées de gestion des notifications, qui est conçue pour améliorer la communication et assurer que les membres de l'équipe sont informés en temps réel des mises à jour importantes du projet.

2- Fonctionnalités :

L'application de gestion de projet comprend plusieurs fonctionnalités clés destinées à assurer une gestion efficace et une exécution réussie des projets :

- Création de Tâches : Les utilisateurs peuvent créer et organiser des tâches avec des dates limites pour structurer le projet.

- Affectation des Tâches : Les tâches peuvent être assignées aux membres de l'équipe de manière claire et précise.

- Suivi de l'Avancement : Un suivi en temps réel de la progression des tâches permet de s'assurer que le projet avance comme prévu.

- Gestion des Jalons et des Dépendances : Les jalons critiques et les dépendances entre les tâches peuvent être définis et suivis pour éviter les retards.

- Rapports de Performance : Génération de rapports détaillés sur l'état d'avancement et les performances du projet.

- Notifications : Système de notifications pour informer les utilisateurs des échéances imminentes et des changements.

- Gestion des Risques et des Changements : Outils pour identifier, évaluer et gérer les risques et les changements tout au long du cycle de vie du projet.

III- Présentation du Code Source :

1- Le Module Classe Principale :

Ce module Python fournit un ensemble de classes et de fonctions pour la gestion efficace de projets. Il comprend six classes principales : Membre, Tache, Equipe, Jalon, Risque, Changement.

Voici le lien GitHub de ce module :
https://github.com/mohamethlo/Projet_MQPL/blob/main/classes_principales/classes_principales.py

2- Le Module Notification :

Ce module Python propose un système complet de gestion des notifications, permettant d'envoyer et de recevoir des notifications de manière efficace et centralisée. Il est composé de quatre classes principales et d'une interface : EmailNotificationStrategy, SMSNotificationStrategy, PushNotificationStrategy, NotificationContext et l'interface NotificationStrategy
Voici le lien GitHub de ce module :
https://github.com/mohamethlo/Projet_MQPL/blob/main/notification/notifications.py

3- Le Module Projet :

Ce module Python fournit une classe unique nommée "Projet" pour encapsuler les informations essentielles d'un projet et centraliser sa gestion. Cette classe offre une structure flexible et extensible pour représenter et gérer les données liées à un projet.

Voici le lien GitHub de ce module :
https://github.com/mohamethlo/Projet_MQPL/blob/main/projet/projet.py

4- Le Module Test :

Ce module Python propose un ensemble d'outils et de fonctionnalités pour nous faciliter la création et l'exécution de tests unitaires dans notre projet. Il nous permet de tester efficacement les composants individuels de notre code, garantissant ainsi sa fiabilité et sa robustesse.

Voici le lien GitHub de ce module :
https://github.com/mohamethlo/Projet_MQPL/blob/main/test.py

5- Le Fichier Main :

Le fichier main.py est le point d'entrée de notre application de gestion de projet. C'est ici que tout commence. Lorsque vous exécutez ce fichier, il initialise les composants essentiels, charge les modules nécessaires et lance l'application.

Voici le lien de ce fichier : https://github.com/mohamethlo/Projet_MQPL/blob/main/main.py

Voici les résultats après exécution :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> python main.py
Notification envoyé à Bakh Diop par email: Bakh Diop a été ajouté a l'équipe
Notification envoyé à Babacar Gueye par email: Bakh Diop a été ajouté a l'équipe
Notification envoyé à Bakh Diop par email: Nouvelle tâche ajoutée : Analyse des besoins
Notification envoyé à Babacar Gueye par email: Nouvelle tâche ajoutée : Analyse des besoins
Notification envoyé à Bakh Diop par email: Nouvelle tâche ajoutée : Developpement
Notification envoyé à Babacar Gueye par email: Nouvelle tâche ajoutée : Developpement
Notification envoyé à Bakh Diop par email: Le budget du projet a été define à 50000 Unité Monétaire
Notification envoyé à Babacar Gueye par email: Le budget du projet a été define à 50000 Unité Monétaire
Notification envoyé à Bakh Diop par email: Nouveau risque ajouté: Retard de livraison
Notification envoyé à Babacar Gueye par email: Nouveau risque ajouté: Retard de livraison
Notification envoyé à Bakh Diop par email: Nouveau jalon ajouté: Phase 1 terminée
Notification envoyé à Babacar Gueye par email: Nouveau jalon ajouté: Phase 1 terminée
Notification envoyé à Bakh Diop par email: Changement enregistré: Changement de la portée du sujet (version 2)
Notification envoyé à Babacar Gueye par email: Changement enregistré: Changement de la portée du sujet (version 2)

#########################################################################################################

Rapport d'activités du projet 'Projet MQPL':
Version: 0
Dates: 2024-05-30 à 2024-12-31
Budget: 0.0 Unité Monétaire
Equipes:
- Bakh Diop (Chef de projet)
- Babacar Gueye (Développeur)
Taches:
- Tâche 1 (2024-06-01 à 2024-06-15), Responsable: Bakh Diop, Statut: En cours
- Tâche 2 (2024-06-10 à 2024-06-30), Responsable: Babacar Gueye, Statut: En attente
Jalons:
- Jalon 1 (2024-06-15)
- Jalon 2 (2024-06-30)
Risques:
- Problème de ressources (Probabilite: 0.3, Impact: Élevé)
- Retard dans la livraison (Probabilite: 0.2, Impact: Moyen)
Chemin critique:
- Tâche 1 (2024-06-01 à 2024-06-15)
- Tâche 2 (2024-06-10 à 2024-06-30)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

IV- Test et Mesure Qualité du Code :

1- Exécution du fichier test :

Pour garantir la fiabilité et la qualité de notre code pour la gestion de projets, nous avons implémenté une série de tests unitaires en utilisant le framework unittest de Python.

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> python test.py -v
test_ajouter_jalon (__main__.TestProjet.test_ajouter_jalon) ... ok
test_ajouter_membre (__main__.TestProjet.test_ajouter_membre) ... ok
test_ajouter_risque (__main__.TestProjet.test_ajouter_risque) ... ok
test_ajouter_tache (__main__.TestProjet.test_ajouter_tache) ... ok
test_definir_budget (__main__.TestProjet.test_definir_budget) ... ok
test_enregistrer_changement (__main__.TestProjet.test_enregistrer_changement) ... ok
test_generer_rapport_performance (__main__.TestProjet.test_generer_rapport_performance) ... ok
test_notifier_email (__main__.TestProjet.test_notifier_email) ... Notification envoyé à Mohameth par email: Message de test
ok
test_set_notification_strategy (__main__.TestProjet.test_set_notification_strategy) ... ok

----------------------------------------------------------------------
Ran 9 tests in 0.002s

OK
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

2- La Commande flake8 :

Flake8 est un outil de vérification de la qualité du code Python. Il nous a permis de corriger les erreurs syntaxiques (les espacements, la longueur des lignes et l'indentation), les imports inutilisés, les références indéfinies.

a- Le Fichier Classe Principale :

- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> flake8 .\classes_principales\classes_principales.py
.\classes_principales\classes_principales.py:10:1: E302 expected 2 blank lines, found 1
.\classes_principales\classes_principales.py:12:80: E501 line too long (119 > 79 characters)
.\classes_principales\classes_principales.py:27:1: E302 expected 2 blank lines, found 1
.\classes_principales\classes_principales.py:32:1: E302 expected 2 blank lines, found 1
.\classes_principales\classes_principales.py:38:1: E302 expected 2 blank lines, found 1
.\classes_principales\classes_principales.py:44:1: E302 expected 2 blank lines, found 1
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

- Exécution après correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> flake8 .\classes_principales\classes_principales.py
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

b- Le Fichier Notification :

- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> flake8 .\notification\notifications.py
.\notification\notifications.py:1:1: F403 'from classes_principales.classes_principales import *' used; unable to detect undefined names
.\notification\notifications.py:3:1: E302 expected 2 blank lines, found 1
.\notification\notifications.py:4:51: F405 'Membre' may be undefined, or defined from star imports: classes_principales.classes_principales
.\notification\notifications.py:7:1: E302 expected 2 blank lines, found 1
.\notification\notifications.py:8:51: F405 'Membre' may be undefined, or defined from star imports: classes_principales.classes_principales
.\notification\notifications.py:11:1: E302 expected 2 blank lines, found 1
.\notification\notifications.py:12:51: F405 'Membre' may be undefined, or defined from star imports: classes_principales.classes_principales
.\notification\notifications.py:15:1: E302 expected 2 blank lines, found 1
.\notification\notifications.py:16:51: F405 'Membre' may be undefined, or defined from star imports: classes_principales.classes_principales
.\notification\notifications.py:19:1: E302 expected 2 blank lines, found 1
.\notification\notifications.py:23:53: F405 'List' may be undefined, or defined from star imports: classes_principales.classes_principales
.\notification\notifications.py:23:58: F405 'Membre' may be undefined, or defined from star imports: classes_principales.classes_principales
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

- Exécution après correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> flake8 .\notification\notifications.py
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

c- Le fichier Projet :

- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> flake8 .\projet\projet.py
.\projet\projet.py:2:1: F403 'from notification.notifications import *' used; unable to detect undefined names
.\projet\projet.py:4:1: E302 expected 2 blank lines, found 1
.\projet\projet.py:5:64: F405 'date' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:5:80: F405 'date' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:5:80: E501 line too long (85 > 79 characters)
.\projet\projet.py:11:23: F405 'Equipe' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:20:51: F405 'NotificationStrategy' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:21:37: F405 'NotificationContext' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:23:36: F405 'Tache' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:26:45: F405 'Membre' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:32:38: F405 'Risque' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:35:36: F405 'Jalon' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:40:22: F405 'Changement' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:44:80: E501 line too long (115 > 79 characters)
.\projet\projet.py:45:80: E501 line too long (216 > 79 characters)
.\projet\projet.py:46:80: E501 line too long (90 > 79 characters)
.\projet\projet.py:47:80: E501 line too long (148 > 79 characters)
.\projet\projet.py:48:80: E501 line too long (194 > 79 characters)
.\projet\projet.py:50:12: E127 continuation line over-indented for visual indent
.\projet\projet.py:64:53: F405 'List' may be undefined, or defined from star imports: notification.notifications
.\projet\projet.py:64:58: F405 'Membre' may be undefined, or defined from star imports: notification.notifications
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

- Exécution après correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> flake8 .\projet\projet.py
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

d- **Le Fichier Test :**

- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> flake8 .\test.py
.\test.py:2:1: F403 'from projet.projet import *' used; unable to detect undefined names
.\test.py:8:23: F405 'Projet' may be undefined, or defined from star imports: projet.projet
.\test.py:9:51: F405 'date' may be undefined, or defined from star imports: projet.projet
.\test.py:9:69: F405 'date' may be undefined, or defined from star imports: projet.projet
.\test.py:9:80: E501 line too long (86 > 79 characters)
.\test.py:11:24: F405 'Membre' may be undefined, or defined from star imports: projet.projet
.\test.py:12:24: F405 'Membre' may be undefined, or defined from star imports: projet.projet
.\test.py:13:23: F405 'Tache' may be undefined, or defined from star imports: projet.projet
.\test.py:16:13: F405 'date' may be undefined, or defined from star imports: projet.projet
.\test.py:17:13: F405 'date' may be undefined, or defined from star imports: projet.projet
.\test.py:21:23: F405 'Tache' may be undefined, or defined from star imports: projet.projet
.\test.py:24:13: F405 'date' may be undefined, or defined from star imports: projet.projet
.\test.py:25:13: F405 'date' may be undefined, or defined from star imports: projet.projet
.\test.py:29:23: F405 'Jalon' may be undefined, or defined from star imports: projet.projet
.\test.py:29:40: F405 'date' may be undefined, or defined from star imports: projet.projet
.\test.py:30:24: F405 'Risque' may be undefined, or defined from star imports: projet.projet
.\test.py:33:26: F405 'EmailNotificationStrategy' may be undefined, or defined from star imports: projet.projet
.\test.py:35:80: E501 line too long (83 > 79 characters)
.\test.py:65:80: E501 line too long (80 > 79 characters)
.\test.py:69:26: F405 'EmailNotificationStrategy' may be undefined, or defined from star imports: projet.projet
.\test.py:72:80: E501 line too long (85 > 79 characters)
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

3- La commande pylint :

La commande pylint est un outil d'analyse statique pour le code Python qui nous a permis de respecter les conventions de style de code, et d'améliorer la qualité générale du code, de détecter diverses erreurs comme :

- Docstrings Manquantes : Il nous signale l'absence des commentaires dans les modules, classes, fonctions ou méthodes.

- Conformité des Docstrings : IL nous a signalé qu'il y'a les docstrings qui ne respectent pas les conventions appropriées.

- Trop d'Arguments : Il nous a signalé qu'il y a des méthodes avec beaucoup d'arguments.

- Trop de Méthodes Publiques : Il nous a signalé qu'il y a des classes avec moins de deux (2) méthodes publiques ou avec beaucoup de méthodes publiques.

a- Le Fichier Classe Principale :

- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> pylint .\classes_principales\classes_principales.py
************* Module classes_principales.classes_principales
classes_principales\classes_principales.py:1:0: C0114: Missing module docstring (missing-module-docstring)
classes_principales\classes_principales.py:22:4: C0116: Missing function or method docstring (missing-function-docstring)
classes_principales\classes_principales.py:25:4: C0116: Missing function or method docstring (missing-function-docstring)
classes_principales\classes_principales.py:29:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:30:33: W0621: Redefining name 'date' from outer scope (line 1) (redefined-outer-name)
classes_principales\classes_principales.py:29:0: R0903: Too few public methods (0/2) (too-few-public-methods)
classes_principales\classes_principales.py:35:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:35:0: R0903: Too few public methods (0/2) (too-few-public-methods)
classes_principales\classes_principales.py:42:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:43:55: W0621: Redefining name 'date' from outer scope (line 1) (redefined-outer-name)
classes_principales\classes_principales.py:42:0: R0903: Too few public methods (0/2) (too-few-public-methods)
classes_principales\classes_principales.py:49:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:53:4: C0116: Missing function or method docstring (missing-function-docstring)
classes_principales\classes_principales.py:56:4: C0116: Missing function or method docstring (missing-function-docstring)

-------------------------------------------------------------------
Your code has been rated at 5.50/10 (previous run: 5.50/10, +0.00)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

- Exécution apres correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> pylint .\classes_principales\classes_principales.py


-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

b- Le Fichier Notification :

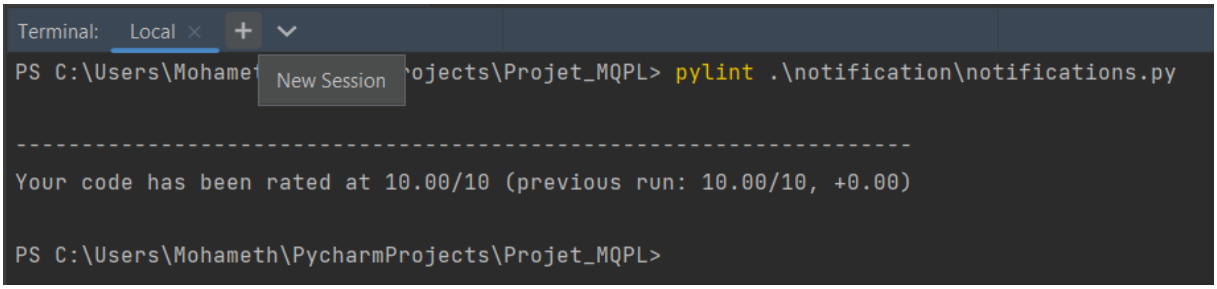
- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> pylint .\classes_principales\classes_principales.py
************* Module classes_principales.classes_principales
classes_principales\classes_principales.py:1:0: C0114: Missing module docstring (missing-module-docstring)
classes_principales\classes_principales.py:22:4: C0116: Missing function or method docstring (missing-function-docstring)
classes_principales\classes_principales.py:25:4: C0116: Missing function or method docstring (missing-function-docstring)
classes_principales\classes_principales.py:29:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:30:33: W0621: Redefining name 'date' from outer scope (line 1) (redefined-outer-name)
classes_principales\classes_principales.py:29:0: R0903: Too few public methods (0/2) (too-few-public-methods)
classes_principales\classes_principales.py:35:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:35:0: R0903: Too few public methods (0/2) (too-few-public-methods)
classes_principales\classes_principales.py:42:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:43:55: W0621: Redefining name 'date' from outer scope (line 1) (redefined-outer-name)
classes_principales\classes_principales.py:42:0: R0903: Too few public methods (0/2) (too-few-public-methods)
classes_principales\classes_principales.py:49:0: C0115: Missing class docstring (missing-class-docstring)
classes_principales\classes_principales.py:53:4: C0116: Missing function or method docstring (missing-function-docstring)
classes_principales\classes_principales.py:56:4: C0116: Missing function or method docstring (missing-function-docstring)

-------------------------------------------------------------------
Your code has been rated at 5.50/10 (previous run: 5.50/10, +0.00)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

- Exécution après correction :



c- Le Fichier Projet :

- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> pylint .\projet\projet.py
************* Module projet.projet
projet\projet.py:1:0: C0114: Missing module docstring (missing-module-docstring)
projet\projet.py:8:0: C0115: Missing class docstring (missing-class-docstring)
projet\projet.py:8:0: R0902: Too many instance attributes (13/7) (too-many-instance-attributes)
projet\projet.py:26:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:29:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:32:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:35:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:38:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:41:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:44:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:50:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:92:4: C0116: Missing function or method docstring (missing-function-docstring)
projet\projet.py:95:4: C0116: Missing function or method docstring (missing-function-docstring)

-------------------------------------------------------------------
Your code has been rated at 7.23/10 (previous run: 7.23/10, +0.00)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

- Exécution après correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> pylint .\projet\projet.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

d- Le fichier Test :

- Exécution avant correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> pylint .\test.py
************* Module test
test.py:1:0: C0114: Missing module docstring (missing-module-docstring)
test.py:6:0: C0115: Missing class docstring (missing-class-docstring)
test.py:36:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:43:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:47:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:52:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:56:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:60:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:64:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:76:4: C0116: Missing function or method docstring (missing-function-docstring)
test.py:85:4: C0116: Missing function or method docstring (missing-function-docstring)

-------------------------------------------------------------------
Your code has been rated at 8.07/10 (previous run: 8.07/10, +0.00)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

- Exécution après correction :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> pylint .\test.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

4- La commande mypy :

Mypy est un outil de vérification de type statique pour Python. Il a aidé à détecter les erreurs de type dans notre code. Voici les corrections que mypy nous a permis de faire :

- Ajout d'Annotations de Type pour les Listes

- Annotation de Type pour l'attribut notification_context de la classe projet

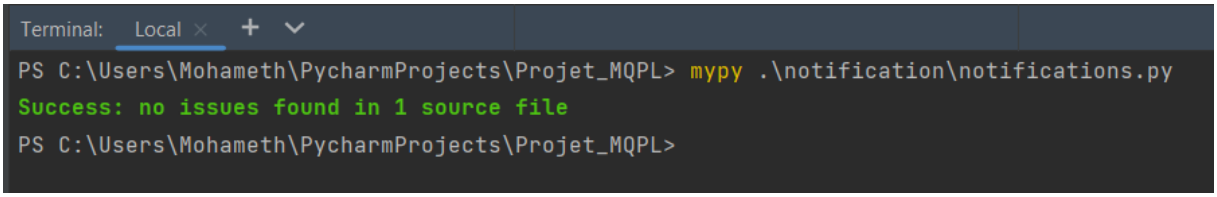
a- Le Fichier Classe Principale :

Il n'y a de correction à faire, car ce fichier ne contient pas d'erreurs de type .

```
Terminal:  Local  +
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> mypy .\classes_principales\classes_principales.py
Success: no issues found in 1 source file
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

b- Le Fichier Notification :

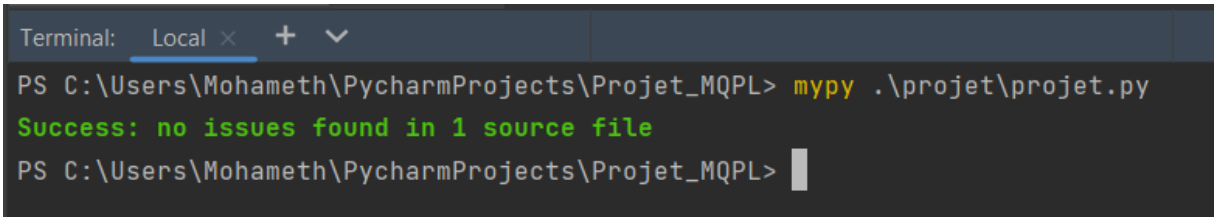
Ce fichier aussi ne contient pas d'erreur de type.



c- Le Fichier Projet :

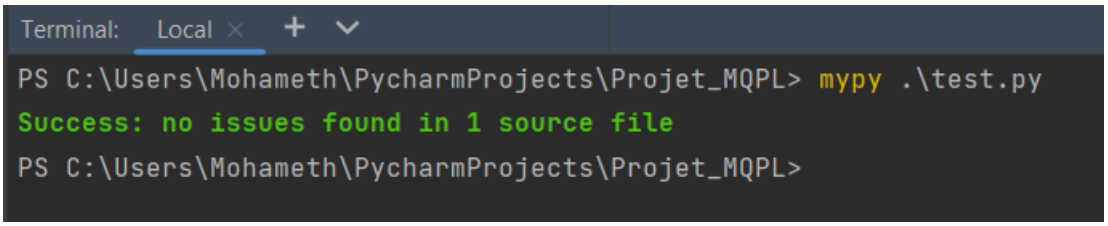
- Exécution de la commande avant correction :

- Exécution de la commande après correction :



d- Le Fichier Test :

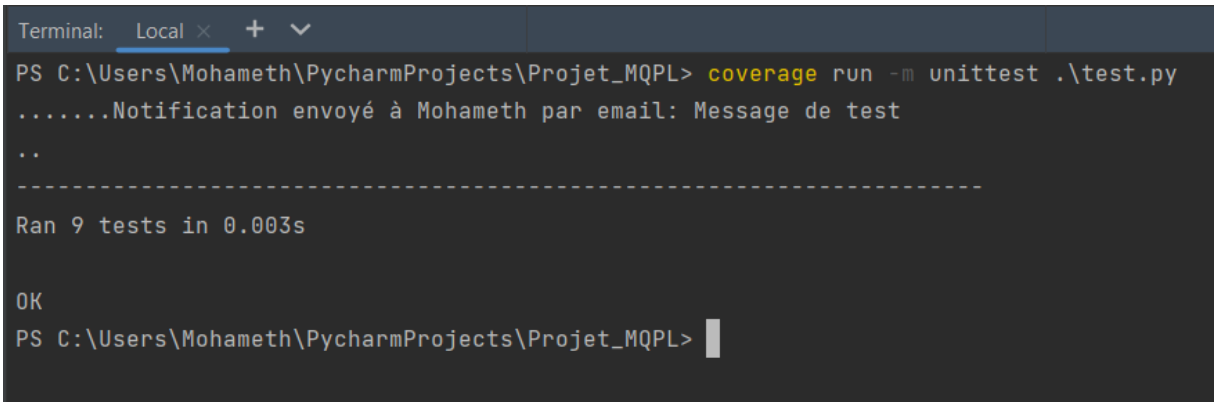
Ce fichier ne contient pas d'erreur aussi :



5- La commande coverage :

Dans le cadre de l'évaluation de la couverture de code et de l'efficacité des tests automatisés de notre projet, nous avons utilisé l'outil coverage. Cet outil nous permet de vérifier quelle proportion de notre code source est exécutée par les tests, nous aidant ainsi à identifier les éventuelles parties du code non testées

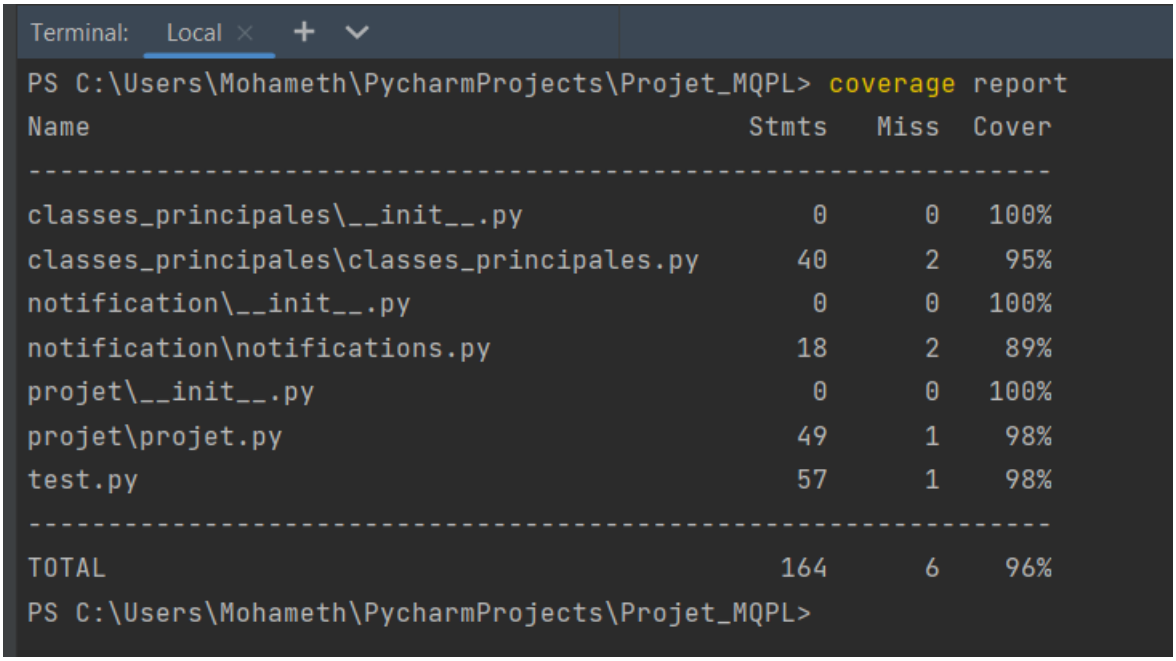
- Resultat des testes :



Les résultats des tests montrent que toutes les fonctionnalités testées fonctionnent correctement et que la couverture de code est satisfaisante. L'utilisation de l'outil coverage a permis de valider l'exécution des tests et de mesurer la couverture de code, assurant ainsi une meilleure qualité et fiabilité du logiciel développé. Ces informations sont cruciales pour garantir que le code est bien testé et que toutes les fonctionnalités critiques sont couvertes par les tests automatisés.

- Génération du rapport de couverture :

Après avoir exécuté nos tests unitaires, nous avons utilisé la commande coverage report pour générer un rapport détaillé de la couverture de code. Ce rapport nous permet d'identifier les parties du code qui sont testées et celles qui ne le sont pas, facilitant ainsi l'amélioration de notre couverture de tests.

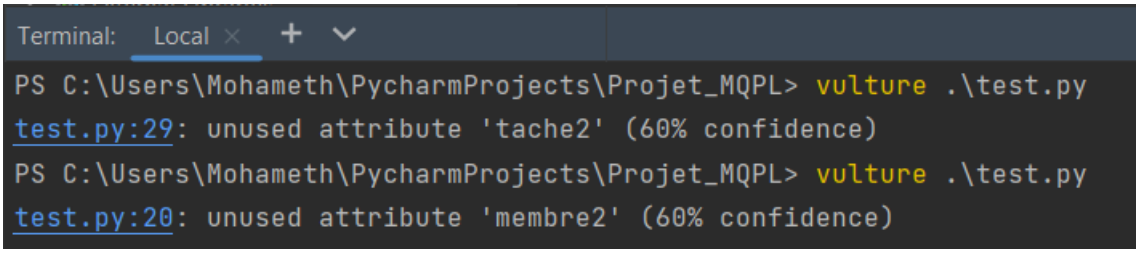


Le rapport de couverture indique une couverture globale très élevée de 96%, ce qui est très satisfaisant. Cependant, il reste quelques lignes de code non couvertes par les tests, indiquant des opportunités pour améliorer encore la qualité et la robustesse de notre projet en ajoutant des tests pour ces parties spécifiques. L'utilisation de la commande coverage report s'est avérée être un outil précieux pour identifier ces zones et garantir une couverture de code optimale.

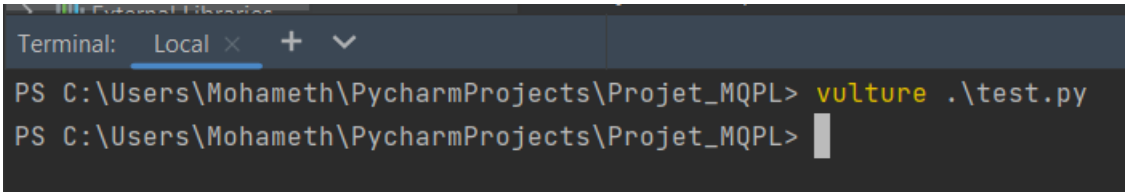
6- La commande vulture :

Vulture est un outil statique de qualité de code pour les projets Python. Il nous a permis d'analyser notre code source afin d'identifier les éléments inutilisés, tels que les fonctions, les classes, les variables, et les importations. L'utilisation de vulture nous a permis de nettoyer notre code en supprimant ces éléments inutilisés, ce qui améliore la lisibilité et la maintenance notre code.

- Exécution de la commande avant correction :



- Exécution de la commande après correction :

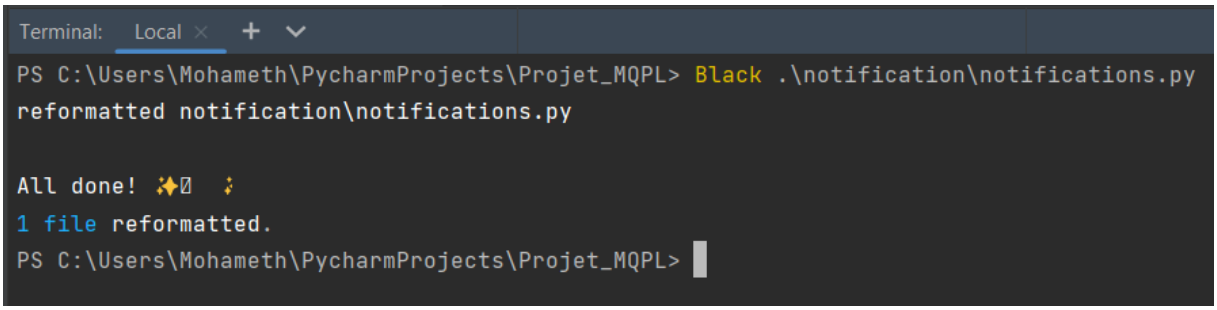


7- Lacommande Black :

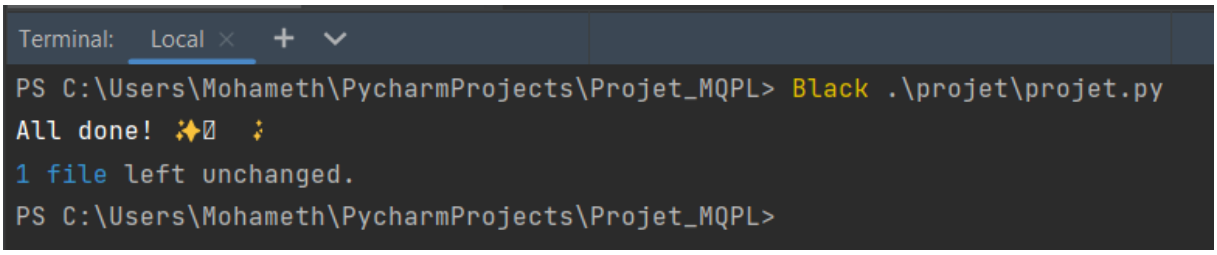
Black est un formateur de code automatique qui applique un style de codage cohérent à tout le code Python. L'utilisation de Black dans notre projet nous a permis d'améliorer la qualité et la cohérence du code, facilitant ainsi la collaboration et la maintenance du projet.

a- Le Fichier Classe Principale :

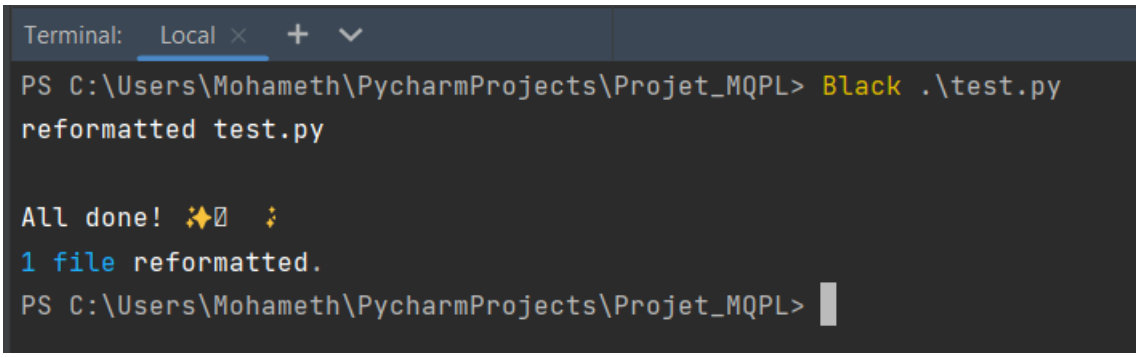
b- Le Fichier Notification :



c- Le Fichier Projet :



d- Le Fichier Test :



Ces résultats sont importants car ils confirment que notre code est déjà conforme aux normes de formatage de Black. Cela signifie que notre code est bien structuré et respecte les meilleures pratiques de style de codage, ce qui améliore sa lisibilité et sa maintenabilité.

L'utilisation de Black nous a permis de vérifier et de valider que notre code est correctement formaté, contribuant ainsi à la qualité globale et à la standardisation de notre projet.

8- La commande radon :

Nous avons utilisé l'outil Radon pour analyser la complexité cyclomatique pour notre projet. La complexité cyclomatique est une métrique qui permet de mesurer le nombre de chemins indépendants à travers le code, ce qui pourra nous aider à identifier les parties du code qui pourraient être trop complexes et nécessiter une refactorisation. L'objectif de cette analyse est d'assurer que notre code reste simple, lisible et facile à maintenir.

a- Le Fichier Classe Principale :

```
Terminal:  Local
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> radon cc .\classes_principales\classes_principales.py
.\classes_principales\classes_principales.py
    C 9:0 Membre - A
    C 22:0 Tache - A
    C 50:0 Jalon - A
    C 59:0 Risque - A
    C 69:0 Changement - A
    C 79:0 Equipe - A
    M 13:4 Membre.__init__ - A
    M 27:4 Tache.__init__ - A
    M 41:4 Tache.ajouter_dependance - A
    M 45:4 Tache.mettre_a_jour_statut - A
    M 54:4 Jalon.__init__ - A
    M 63:4 Risque.__init__ - A
    M 73:4 Changement.__init__ - A
    M 82:4 Equipe.__init__ - A
    M 86:4 Equipe.ajouter_membre - A
    M 90:4 Equipe.obtenir_membres - A
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

b- Le Fichier Notification :

```
Terminal:  Local
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> radon cc .\notification\notifications.py
.\notification\notifications.py
    C 40:0 NotificationContext - A
    C 6:0 NotificationStrategy - A
    C 15:0 EmailNotificationStrategy - A
    C 23:0 SMSNotificationStrategy - A
    C 32:0 PushNotificationStrategy - A
    M 48:4 NotificationContext.notifier - A
    M 11:4 NotificationStrategy.envoyer - A
    M 19:4 EmailNotificationStrategy.envoyer - A
    M 27:4 SMSNotificationStrategy.envoyer - A
    M 36:4 PushNotificationStrategy.envoyer - A
    M 44:4 NotificationContext.__init__ - A
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

c- Le Fichier Projet :

```
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> radon cc .\projet\projet.py
.\projet\projet.py
    M 69:4 Projet.generer_rapport_performance - B
    C 18:0 Projet - A
    M 123:4 Projet.notifier - A
    M 22:4 Projet.__init__ - A
    M 38:4 Projet.set_notification_strategy - A
    M 42:4 Projet.ajouter_tache - A
    M 46:4 Projet.ajouter_membre_equipe - A
    M 50:4 Projet.definir_budget - A
    M 54:4 Projet.ajouter_risque - A
    M 58:4 Projet.ajouter_jalon - A
    M 62:4 Projet.enregistrer_changement - A
    M 119:4 Projet.calculer_chemin_critique - A
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

d- Le Fichier Test :

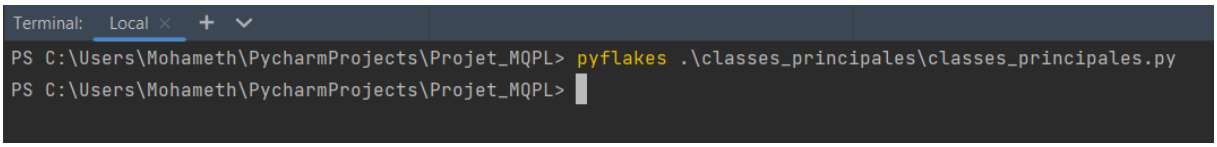
```
Terminal:  Local
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL> radon cc .\test.py
.\test.py
    C 8:0 TestProjet - A
    M 11:4 TestProjet.setUp - A
    M 31:4 TestProjet.test_set_notification_strategy - A
    M 37:4 TestProjet.test_ajouter_membre - A
    M 42:4 TestProjet.test_definir_budget - A
    M 48:4 TestProjet.test_ajouter_tache - A
    M 53:4 TestProjet.test_ajouter_jalon - A
    M 58:4 TestProjet.test_ajouter_risque - A
    M 63:4 TestProjet.test_enregistrer_changement - A
    M 74:4 TestProjet.test_notifier_email - A
    M 81:4 TestProjet.test_generer_rapport_performance - A
PS C:\Users\Mohameth\PycharmProjects\Projet_MQPL>
```

Après avoir exécuté la commande radon cc, les résultats nous indiquent que toutes les classes et méthodes évaluées ont obtenu une note de complexité cyclomatique de niveau A. Cela signifie que notre code est très simple et bien structuré. Toutes les classes et méthodes présentent une faible complexité. Ces résultats montrent que le code est facile à comprendre et à maintenir, ce qui est essentiel pour garantir la qualité et la longévité du projet.

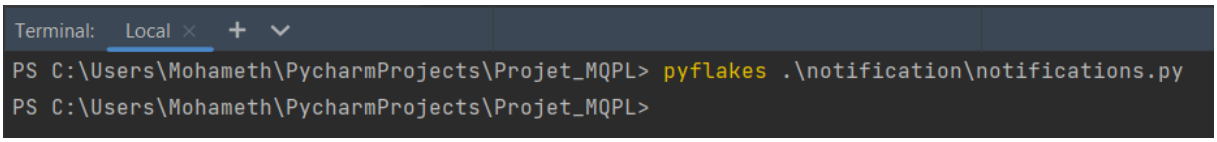
9- La commande pyflakes :

Pyflakes est un outil d'analyse statique pour Python qui se concentre sur la détection des erreurs de programmation courantes, telles que les variables non définies, les importations inutilisées et les redéfinitions de variables. Son utilisation régulière permet de maintenir un code propre et sans erreur, ce qui facilite sa lecture, sa compréhension et sa maintenance.

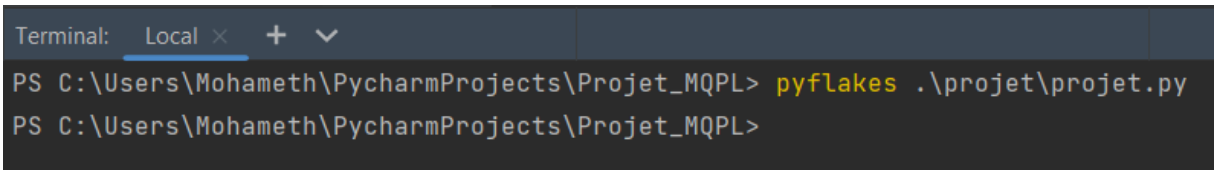
a- Le Fichier Classe Principale :



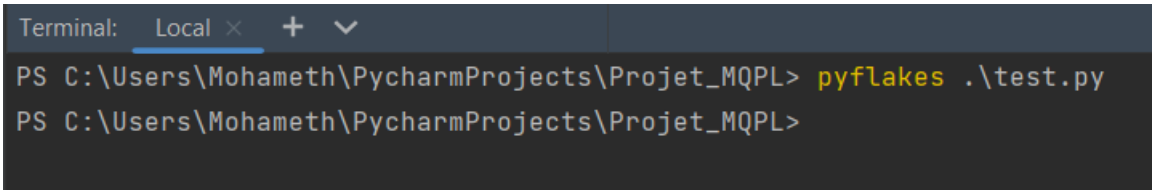
b- Le Fichier Notification :



c- Le Fichier Projet :



d- Le Fichier Test :



V- Conclusion :

En somme, ce projet de gestion de projet est conçu pour améliorer la manière dont les projets sont planifiés, suivis et gérés, en offrant une gamme complète de fonctionnalités adaptées aux besoins des gestionnaires de projets modernes. Cette application vise à renforcer l'efficacité, la communication et la gestion proactive des ressources et des risques, contribuant ainsi à la réussite des projets entrepris par ses utilisateurs.