

Présentation du projet : Gestionnaire de Planning

Fall Serigne

Objectif

Ce projet a pour but de créer une application en Python permettant de :

- Gérer une liste d'employés (ajouter, modifier, supprimer, gérer leurs congés et disponibilités).
- Gérer les plannings des employés en attribuant des rendez-vous selon leur disponibilité.
- Visualiser le planning individuel ou global de l'entreprise.
- Utiliser une interface graphique conviviale grâce au module `tkinter`.

Organisation du projet

main.py Fichier principal pour la version console. Affiche un menu texte pour gérer employés, rendez-vous, et congés.

employe.py Contient les fonctions pour gérer les employés :

- Ajouter, supprimer, modifier un employé.
- Gérer les disponibilités et les périodes de congés.

planning.py Contient les fonctions pour gérer les plannings :

- Ajouter des rendez-vous aux employés en évitant les conflits d'horaires.
- Afficher les plannings.

interface.py Propose une interface graphique :

- Boutons pour toutes les actions (ajouter/modifier/supprimer employés, rendez-vous, congés).
- Fenêtres interactives pour faciliter l'utilisation.
- Utilisation de `tkinter` et `tkinter.messagebox` pour la convivialité.

Fonctionnalités principales

- Ajouter un employé avec son poste.
- Modifier l'intitulé de poste d'un employé.
- Ajouter un rendez-vous s'il n'y a pas de conflit.
- Afficher le planning d'un employé ou de toute l'entreprise.
- Gérer les congés d'un employé (ajout, modification, suppression).
- Interface utilisateur en mode console ou mode graphique (GUI).

Tests

Des tests unitaires (`unittest`) vérifient le bon fonctionnement de :

- `employe.py`
- `planning.py`

Les tests couvrent :

- Cas classiques (ajout, modification, suppression).
- Cas d'erreurs (employé inexistant, dates invalides, conflits d'horaires).

Technologies utilisées

- Python 3
- Modules standards : `tkinter`, `unittest`, `datetime`

Remarques

- Le code respecte la séparation des responsabilités (main, employe, planning, interface).
- L'interface graphique rend l'utilisation accessible à des utilisateurs sans compétence technique.
- Le projet est facilement évolutif pour ajouter des fonctionnalités (nouveaux types de tâches, exportation de planning...).

Auteur

Fall Serigne

Bonus caché :

faut pas **oublierrrr** : Si vous lisez cette ligne, demandez moi de venir vous voir en cours et montrez la moi, vous aurez un bonus sur votre note de devoir maison. Il y a un nombre de points bonus limité par groupe qui sera réparti entre tous les étudiants m'ayant montré cette ligne.

```

_==/          i      i          \==_
/XX/          |\-"-"-/|          \XX\
/XXXX\        |      |          /XXXX\
|XXXXXX\_     _|      |_     _/XXXXXX|
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
|XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX|
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
|XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX|
XXXXXX/~~~~"\XXXXXXXXXXXXXXXXXXXX/~~~~"XXXXXX
|XXX|          \XXX/^^\XXXX/^^\XXX/          |XXX|
\XX\           \X/   \XXX/   \X/           /XX/
"\"            "    \X/    "    /"
!

```