

Projet professionnel réalisé en entreprise :

Développement d'un utilitaire de bureau pour le renommage automatique de fichiers

avec PYTHON et la librairie Tkinter

Sommaire

I. Contexte et problématique	3
II. Analyse du besoin	4
2.1 Règles de nommage	4
2.2 Fonctionnalités attendues	4
2.3 Diagramme de cas d'utilisation	4
III. Développement	5
3.1 Choix technique	5
3.2 Architecture de l'application	5
3.3 Interface graphique	7
3.4 Déploiement	7
IV. Développement	8
4.1 Fonctionnement de l'outil	8
4.2 Bénéfice mesuré pour l'entreprise	10
V. Conclusion et perspectives d'évolution	11

I. Contexte et problématique

IVC AUDIT est un cabinet d'expertise comptable basé à Cergy (Val-d'Oise), ses collaborateurs traitent quotidiennement un volume important de documents clients : factures, relevés bancaires, statuts juridiques, FEC (Fichier des Écritures Comptables), déclarations fiscales ...

Problématique identifiée :

Les documents reçus des clients portaient des noms de fichiers hétérogènes, illisibles et non normalisés. Les collaborateurs perdaient un temps considérable à les chercher, déchiffrer, et renommer manuellement, au détriment des tâches à valeur ajoutée.

J'ai identifié ce besoin et proposé une solution logicielle adaptée aux contraintes de l'environnement de travail : aucune installation Python requise pour les utilisateurs finaux, interface simple et accessible à tous.

II. Analyse du besoin

2.1 Règles de nommage

La convention de nommage adoptée par le cabinet suit le format suivant :

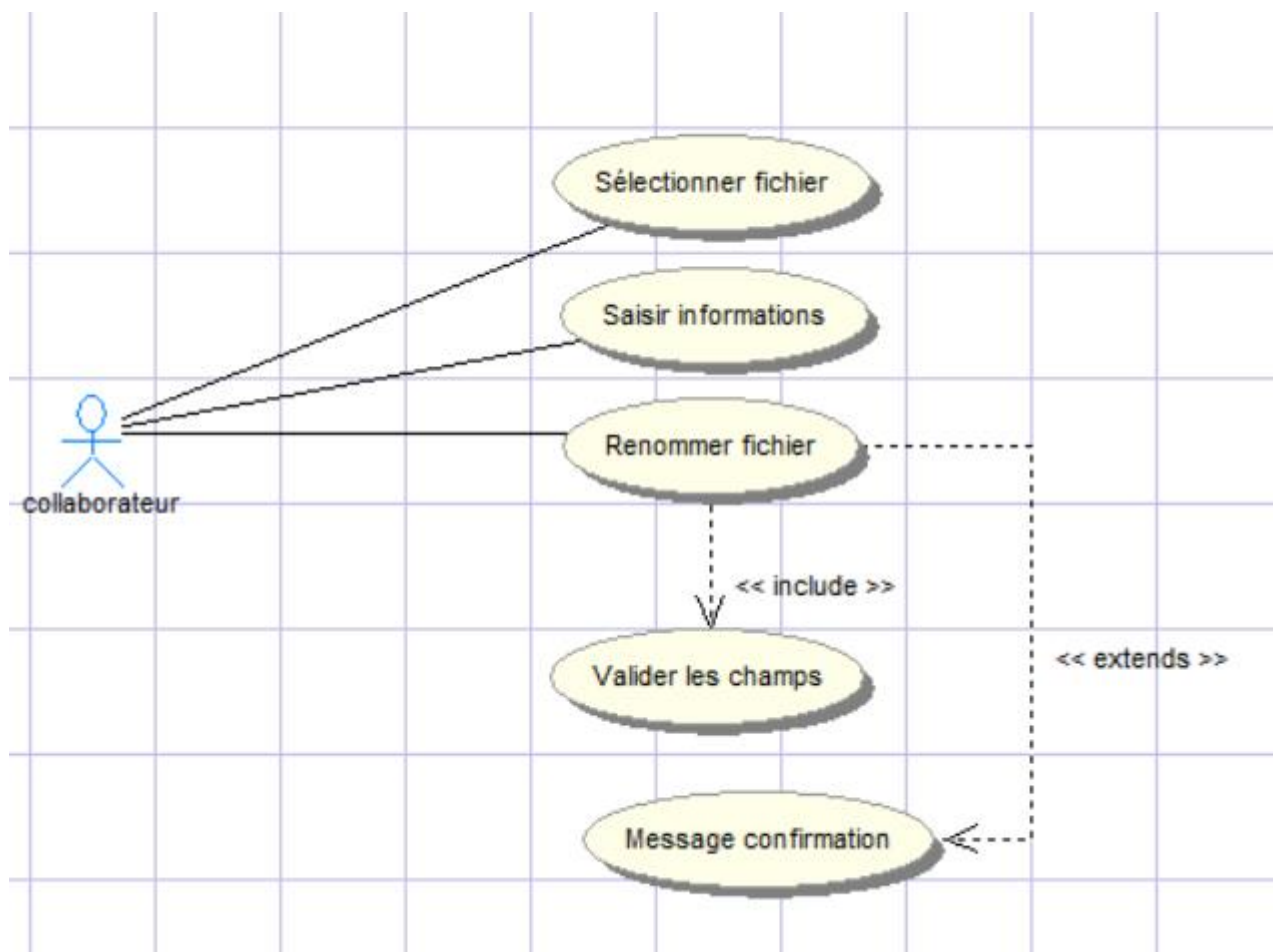
AAAA-MM_NOM-CLIENT_TYPE-DOCUMENT.ext

Exemple : 2025-03_IVC_AUDIT_FACTURE.pdf

2.2 Fonctionnalités attendues

- Sélectionner un fichier via un explorateur de fichiers graphique
- Saisir le nom du client (espaces automatiquement remplacés par des underscores)
- Renseigner l'année et le mois (pré-remplis avec la date courante)
- Choisir le type de document parmi une liste prédéfinie (FACTURE, STATUTS, RELEVÉ_BANCAIRE, FEC, IMPOTS)
- Renommer le fichier en un clic avec application immédiate de la convention
- Fonctionner sans installation de Python sur les postes utilisateurs
- Être accessible depuis SharePoint (Microsoft 365) à l'ensemble des collaborateurs

2.3 Diagramme de cas d'utilisation (DCU avec Windesign)



III. Développement

3.1 Choix techniques

Technologie	Usage	Justification du choix
Python 3	Langage principal	Appris en cours de formation, adapté aux scripts utilitaires
Tkinter	Interface graphique (GUI)	Bibliothèque native Python, aucune dépendance externe
os / os.path	Manipulation de fichiers	Bibliothèque standard Python pour la gestion des chemins
PyInstaller	Compilation en .exe	Permet le déploiement sans environnement Python installé
SharePoint (M365)	Distribution de l'exécutable	Infrastructure déjà en place au cabinet, accessible à tous

3.2 Architecture de l'application

L'application est structurée en deux parties distinctes :

- La fonction de renommage **renommer_fichiers()** : traite la logique métier, construit le nouveau nom de fichier selon la convention, effectue la vérification d'existence et renomme le fichier via **os.rename()**.

```
def renommer_fichiers():

    chemin_complet = entry_fichier.get()
    client = entry_client.get().replace(" ", "_")
    annee = entry_annee.get()
    type_doc = var_type.get()

    if not chemin_complet or not client or not annee:
        messagebox.showwarning("Attention", "Veuillez remplir tous les champs")
        return

    try:
        # on sépare le dossier, le nom et l'extension
        dossier = os.path.dirname(chemin_complet)
        nom_origine = os.path.basename(chemin_complet)
        extension = os.path.splitext(nom_origine)[1]

        # construit le nouveau nom et met 01 par défaut
        nouveau_nom = f"{annee}_{client.upper()}_{type_doc.upper()}_01{extension}"
        nouveau_chemin = os.path.join(dossier, nouveau_nom)

        # renommage
        if os.path.exists(chemin_complet):
            os.rename(chemin_complet, nouveau_chemin)
            messagebox.showinfo("Succès", f"Fichier renommé en : {nouveau_nom}")
            # maj champ de saisie avec le nouveau nom
            entry_fichier.delete(0, tk.END)
            entry_fichier.insert(0, nouveau_chemin)
        else:
            messagebox.showerror("Erreur", "Le fichier est introuvable.")

    except Exception as e:
        messagebox.showerror("Erreur", f"Une erreur est survenue : {e}")
```

- L'interface graphique (root Tkinter) : expose les champs de saisie, la liste déroulante des types de documents, le bouton d'exploration de fichiers et le bouton d'exécution.

```
# interface graphique
root = tk.Tk()
root.title("IVC AUDIT - Renommage de fichier")
root.geometry("400x450")

tk.Label(root, text="Sélectionner le fichier :", font=("Arial",10,"bold")).pack(pady=(20,5))
entry_fichier = tk.Entry(root, width=50)
entry_fichier.pack(padx=10)

# on vide avant d'insérer
def bouton_parcourir():
    f = filedialog.askopenfilename()
    if f:
        entry_fichier.delete(0, tk.END)
        entry_fichier.insert(0, f)

btn_parcourir = tk.Button(
    root,
    text="Parcourir",
    bg="#4A6572",
    fg="white",
    command=bouton_parcourir,
    activebackground="#25353F", # Couleur encore plus sombre au clic
    activeforeground="white",
    relief="flat"
)
btn_parcourir.pack(pady=5)
btn_parcourir.bind("<Enter>", lambda e: btn_parcourir.config(bg="#344955"))
btn_parcourir.bind("<Leave>", lambda e: btn_parcourir.config(bg="#4A6572"))

tk.Label(root, text="Indiquer le client :", font=("Arial",10,"bold")).pack(pady=(20,5))
entry_client = tk.Entry(root, width=40)
entry_client.pack()
```

```
tk.Label(root, text="Année & mois :", font=("Arial",10,"bold")).pack(pady=(20,5))
entry_annee = tk.Entry(root, width=10)
# affichage année
entry_annee.insert(0, time.strftime("%Y-%m"))
entry_annee.pack()

tk.Label(root, text="Type de document :", font=("Arial",10,"bold")).pack(pady=(20,5))
var_type = tk.StringVar(value="FACTURE")
options = ["FACTURE", "STATUTS", "RELEVÉ BANCAIRE", "FEC", "IMPOTS"]
menu_choix = tk.OptionMenu(root, var_type, *options)
menu_choix.config(
    bg="#4A6572",
    fg="white",
    activebackground="#344955",
    activeforeground="white",
    relief="flat"
)
menu_choix["menu"].config(
    bg="#F4F6F8",
    fg="#212A37",
    activebackground="#4A6572",
    activeforeground="white",
    font=("Helvetica", 10)
)
menu_choix.pack()

tk.Button(root, text="Renommer le fichier →", bg="green", fg="white", command=renommer_fichiers, font=("Arial",10,"bold")).pack(pady=(70,20))

root.mainloop()
```

3.3 Interface graphique

L'interface graphique Tkinter a été conçue pour être intuitive et utilisable sans formation préalable par des collaborateurs non-techniciens :

Composant UI	Rôle et comportement
Champ « Fichier »	Chemin complet du fichier à renommer
Bouton « Parcourir »	Ouvre l'explorateur Windows via <code>filedialog.askopenfilename()</code>
Champ « Client »	Nom du client, les espaces sont automatiquement remplacés par des underscores
Champ « Année & mois »	Pré-rempli avec la date courante (format AAAA-MM) via <code>time.strftime()</code>
Liste « Type de document »	Menu déroulant : FACTURE, STATUTS, RELEVÉ_BANCAIRE, FEC, IMPOTS
Bouton « renommer »	Déclenche le renommage et affiche une boîte de dialogue de confirmation ou d'erreur

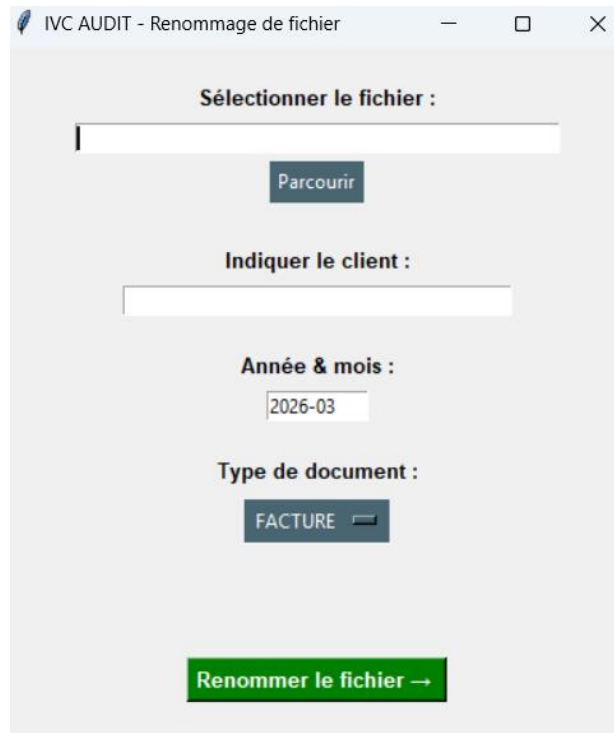
3.4 Déploiement

- Compilation en exécutable Windows autonome (.exe) via PyInstaller : aucune installation Python nécessaire sur les postes utilisateurs.
- Dépôt de l'exécutable sur un répertoire partagé SharePoint (Microsoft 365), accessible à l'ensemble des collaborateurs du cabinet.
- Mise à jour facilitée : toute nouvelle version compilée remplace le fichier sur SharePoint sans intervention sur les postes.

IV. Résultats et bénéfices

4.1 Fonctionnement de l'outil

Une petite fenêtre s'affiche à l'exécution de l'utilitaire, l'utilisateur est invité à sélectionner son fichier via l'explorateur avec le bouton « parcourir » et sélectionner le type de document. L'année et le mois sont automatiquement calculés et renseignés :



The screenshot shows a window titled "IVC AUDIT - Renommage de fichier". It contains the following elements:

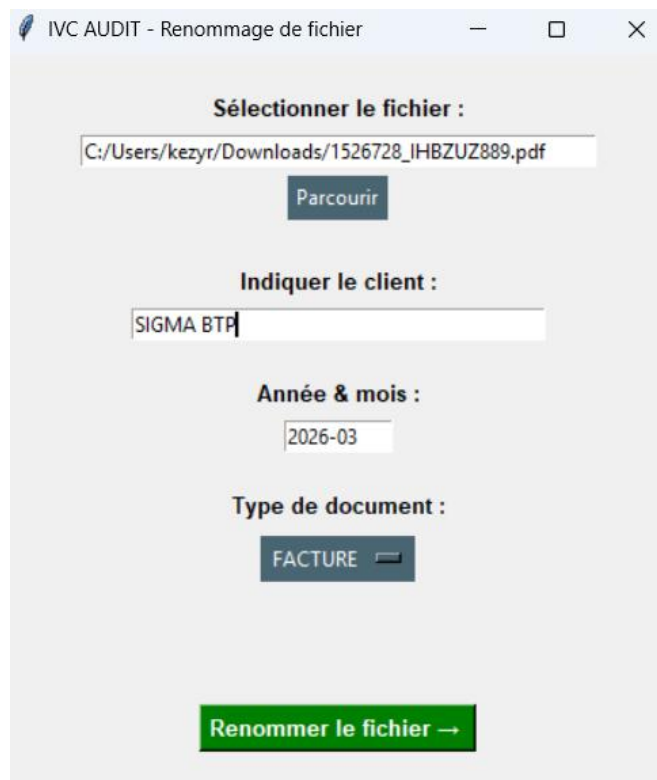
- Sélectionner le fichier :** A text input field is empty.
- Parcourir** : A blue button located below the file selection field.
- Indiquer le client :** A text input field is empty.
- Année & mois :** A text input field containing the value "2026-03".
- Type de document :** A blue button labeled "FACTURE" with a right-pointing arrow.
- Renommer le fichier →** A large green button at the bottom of the window.

Ensuite, l'utilisateur parcourt ses dossiers via l'explorateur et choisi le fichier qui va être sujet au traitement :



This screenshot shows the same window as the previous one, but with the file path "C:/Users/kezyr/Downloads/1526728_IHBZUZ889.pdf" entered into the "Sélectionner le fichier :" input field. The other elements remain the same.

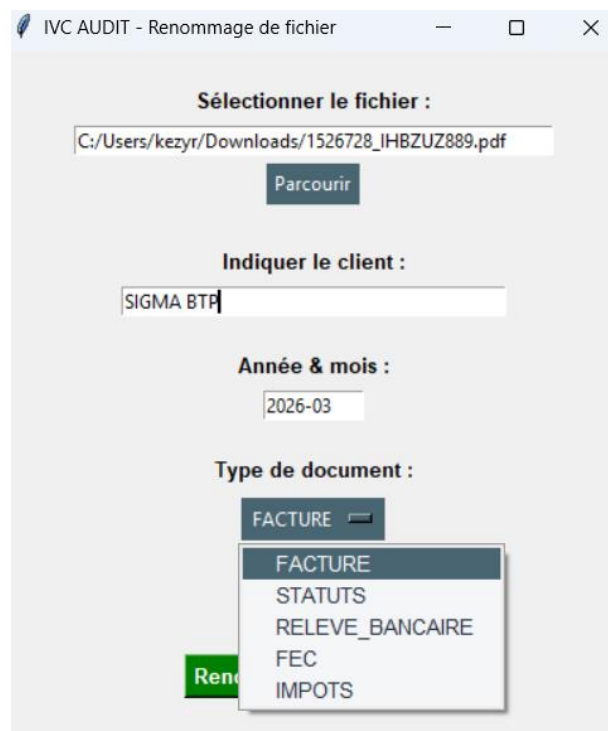
L'utilisateur peut maintenant indiquer le nom/dénomination de son client, qui sera inclus désormais dans le nom du fichier, afin de pouvoir identifier à qui appartient le document :



The screenshot shows a window titled "IVC AUDIT - Renommage de fichier". It contains the following fields and buttons:

- Sélectionner le fichier :** A text box containing the file path "C:/Users/kezyr/Downloads/1526728_IHBZUZ889.pdf" and a "Parcourir" button below it.
- Indiquer le client :** A text box containing "SIGMA BTP".
- Année & mois :** A text box containing "2026-03".
- Type de document :** A button labeled "FACTURE" with a dropdown arrow.
- Renommer le fichier →** A large green button at the bottom.

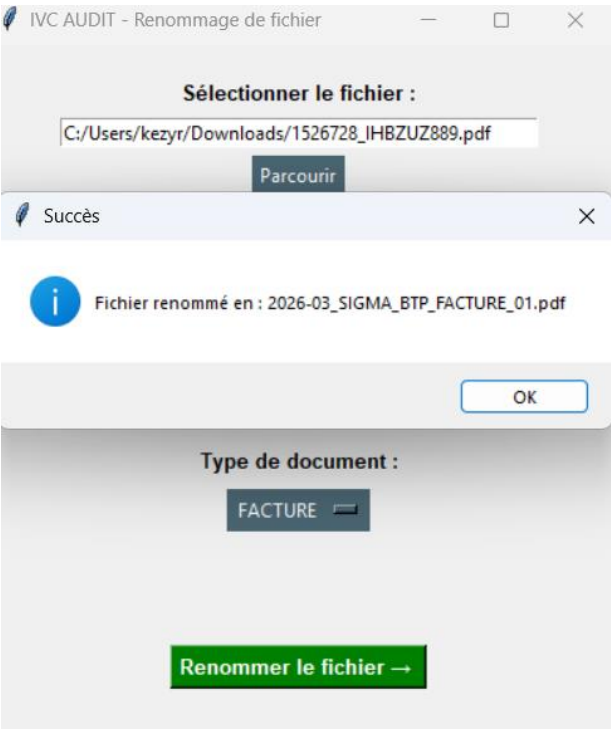
L'utilisateur va ensuite indiquer la nature du document, celle-ci sera aussi intégrée au nom du fichier pour améliorer sa lisibilité et accélérer les processus :



This screenshot is identical to the previous one, but with the "Type de document" dropdown menu open, showing the following options:

- FACTURE (selected)
- STATUTS
- RELEVÉ_BANCAIRE
- FEC
- IMPOTS

Enfin, l'utilisateur n'a plus qu'à confirmer la dénomination du fichier et une fenêtre contextuelle s'affiche afin de confirmer cette validation :



Renommage effectif :

Avant : 1526728_IHBZUZ889.pdf
Après : 2026-03_SIGMA_BTP_FACTURE_01.pdf

4.2 Bénéfices mesurés pour le cabinet

Avant la solution	Après la solution
Renommage manuel, source d'erreurs et de perte de temps	Renommage automatisé en 3 clics, uniforme et sans erreur
Noms hétérogènes selon les collaborateurs	Convention stricte appliquée systématiquement
Difficultés de recherche et de tri dans les dossiers clients	Arborescence lisible
Dépendance aux habitudes individuelles de chaque collaborateur	Outil partagé, accessible via SharePoint, indépendant du poste

V. Conclusion et perspectives

Ce projet, bien qu'en apparence simple, illustre une démarche professionnelle complète : identification d'un besoin réel, analyse, développement, tests et déploiement en environnement de production.

Perspectives d'évolution

- Ajout d'un historique des derniers renommages pour permettre une annulation rapide.
- Extension à la gestion de lots (renommage de plusieurs fichiers en une seule opération).
- Intégration d'une vérification de la conformité du nom existant pour signaler les fichiers déjà bien nommés.
- Création d'une version web légère accessible directement depuis le navigateur (pas de téléchargement d'exécutable requis).

En résumé, cet utilitaire témoigne d'une capacité à analyser un dysfonctionnement organisationnel, à proposer une solution technique adaptée et à la déployer de façon autonome dans un environnement professionnel réels.