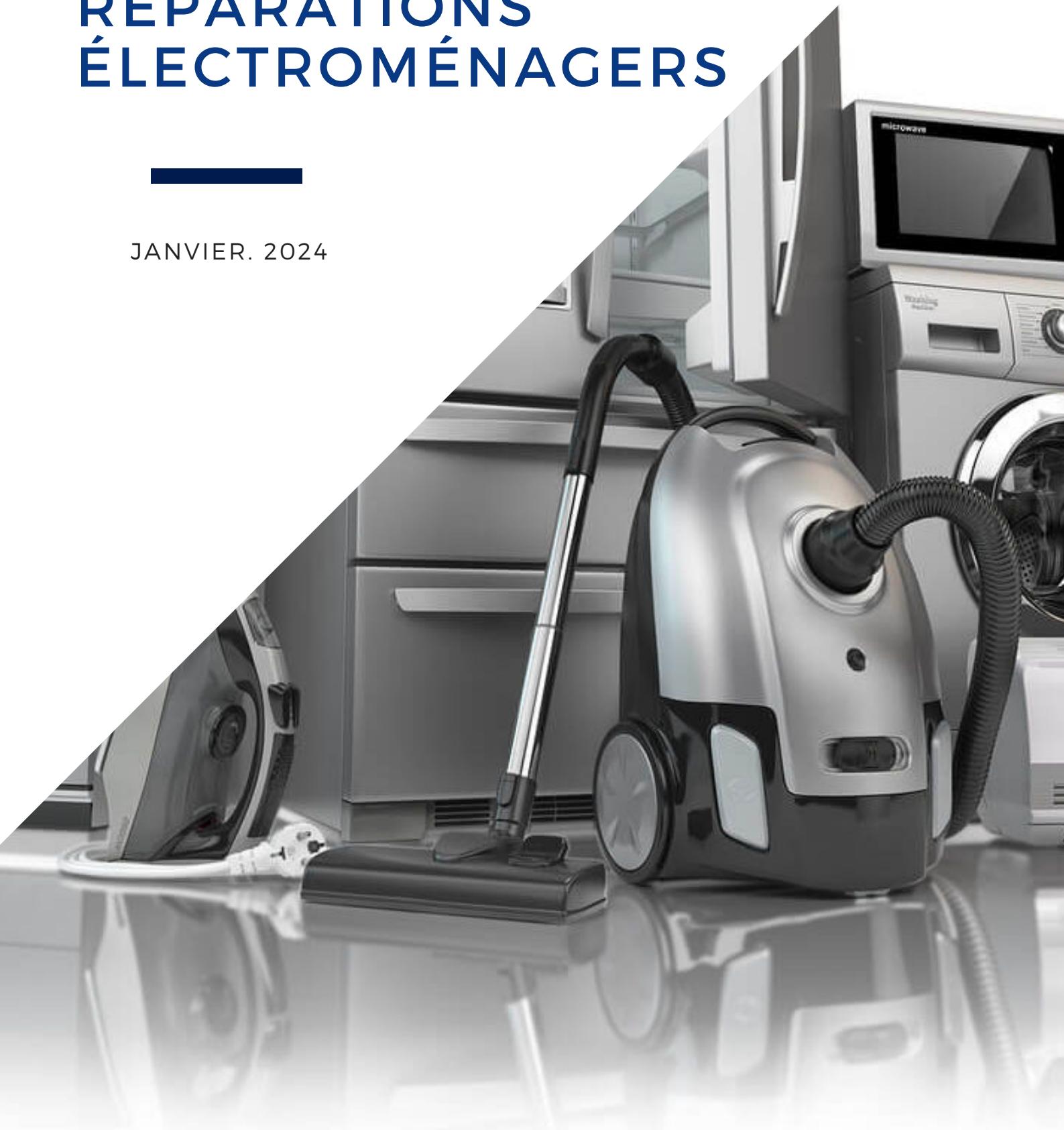


PRÉPARÉ PAR
MEZZOUR OMAR

PROFESSEUR
DR. ELMOUHTADI MERYAM

SQL - GESTION RÉPARATIONS ÉLECTROMÉNAGERS

JANVIER. 2024



Sommaire

I- Aperçu

II- Fonctionnalités

1- Rechercher Client

2- Insérer Ordre de Réparation

3- Calculer Montant Total pour le Client

4- Afficher Pièces Supérieures

5- Afficher Ordres Sans Pièces à Changer

III- Structure du code

IV- Dépendances

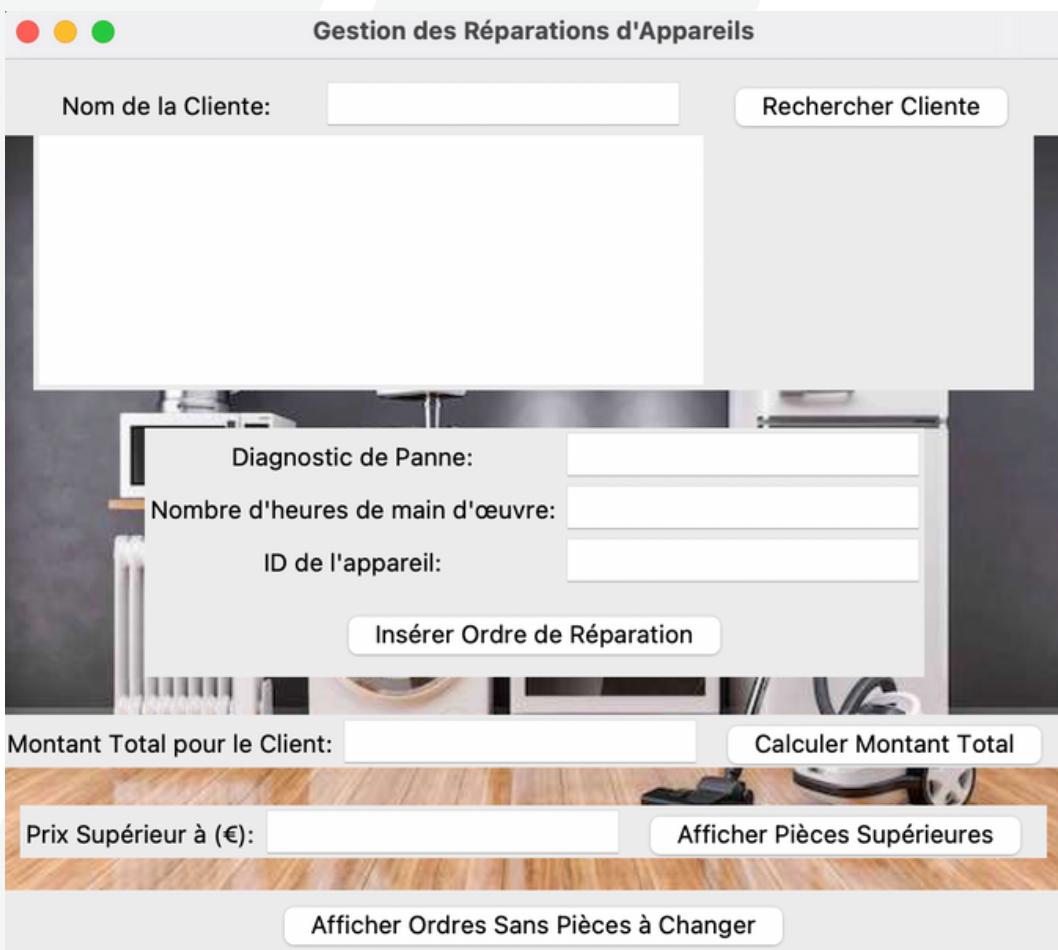
V- Compilation et exécution

VI- Idées d'améliorations futures

VII- Conclusion

Aperçu

Le projet de gestion des réparations d'appareils vise à fournir une solution simple et conviviale pour gérer les ordres de réparation, clients, pièces et catégories associées. L'application est développée en utilisant le langage Python et la bibliothèque Tkinter pour l'interface utilisateur, avec une base de données SQLite pour stocker les informations.



Fonctionnalités

1. Rechercher Client

- L'utilisateur peut rechercher un client en saisissant son nom.
- Les résultats de la recherche sont affichés dans une zone de texte.

RÉSULTAT

Gestion des Réparations d'Appareils

Nom de la Cliente: Rechercher Cliente

1 Client1 Adresse1 Ville1

Diagnostic de Panne:

Nombre d'heures de main d'œuvre:

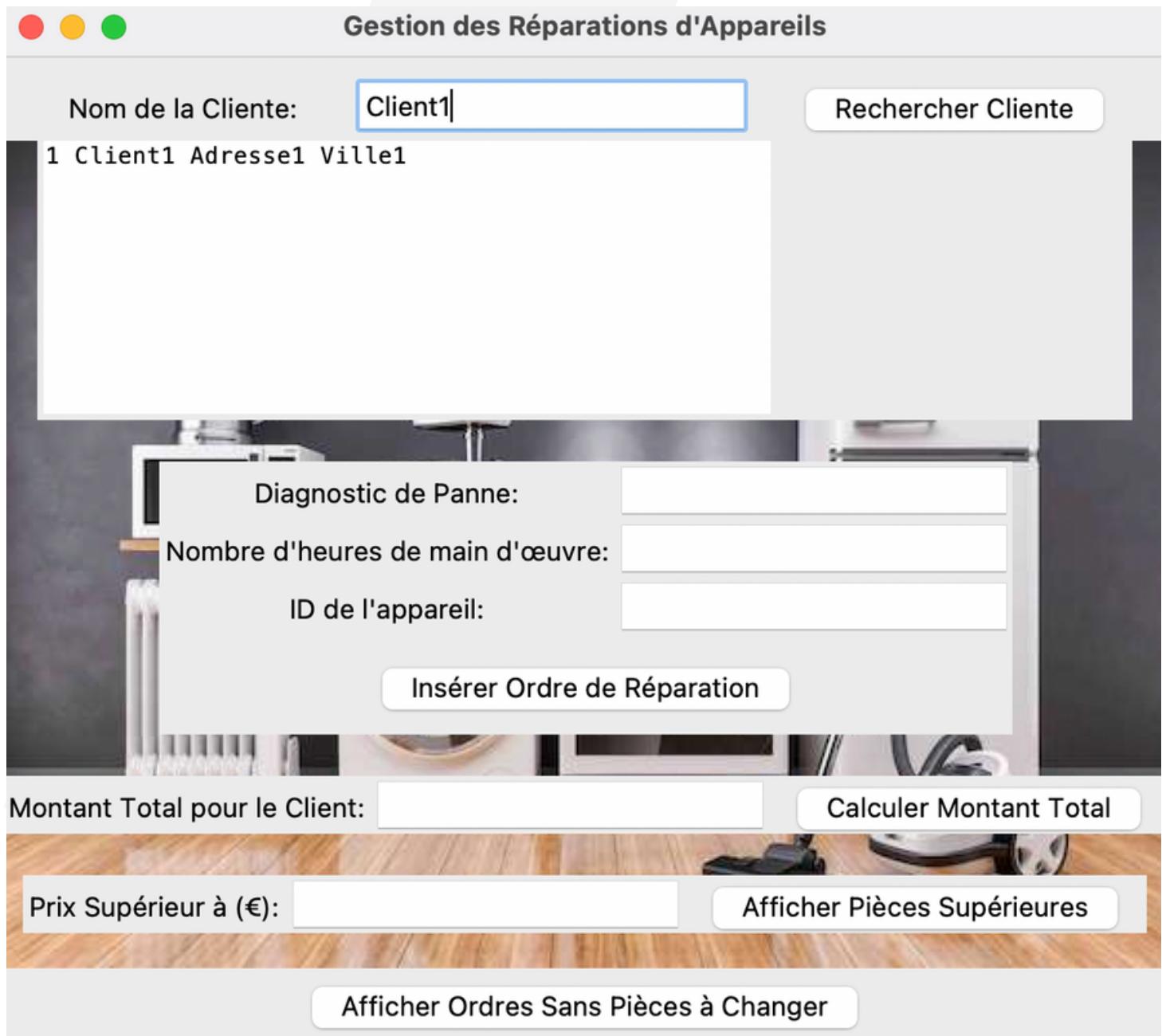
ID de l'appareil:

Insérer Ordre de Réparation

Montant Total pour le Client: Calculer Montant Total

Prix Supérieur à (€): Afficher Pièces Supérieures

Afficher Ordres Sans Pièces à Changer

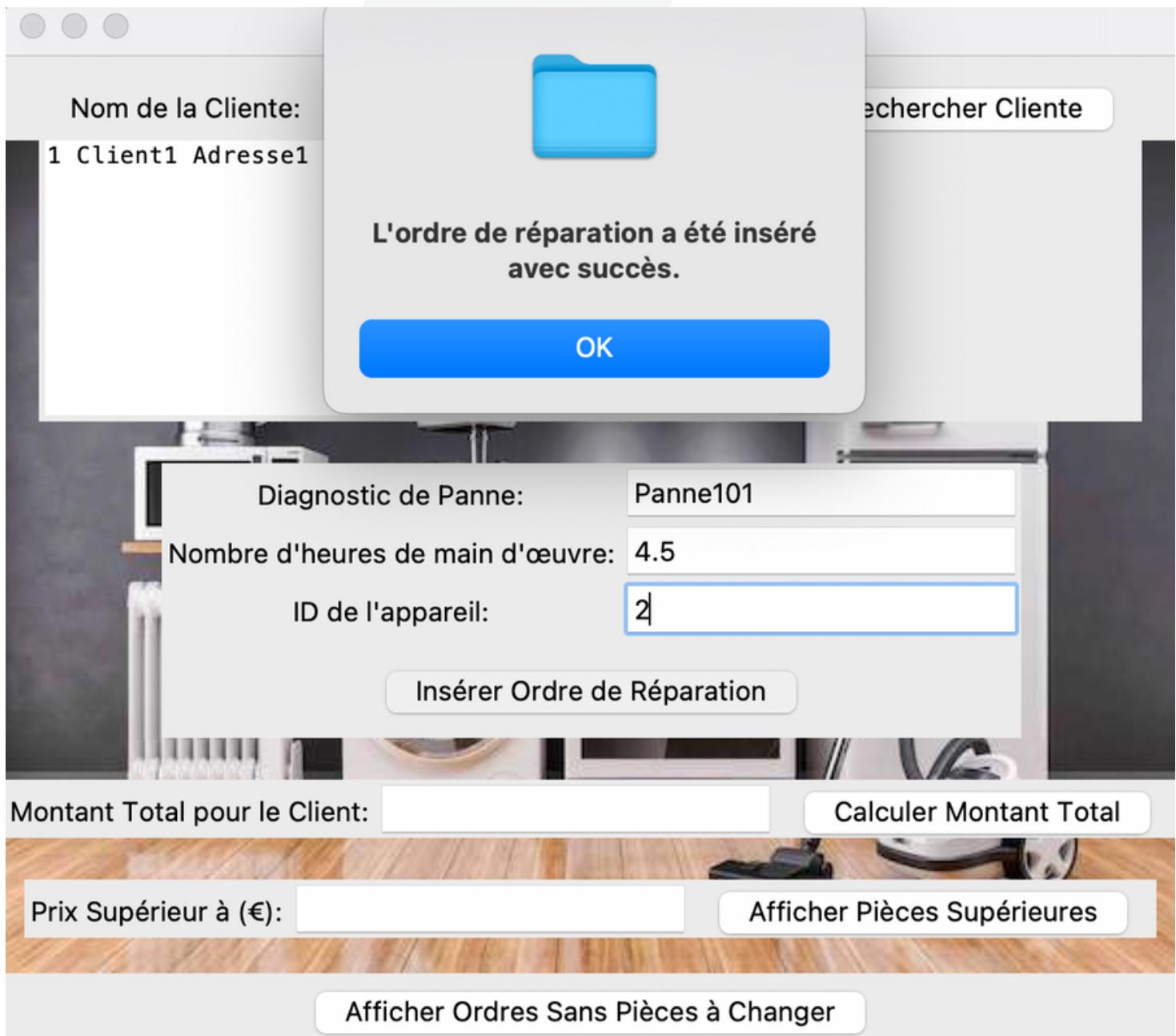


Fonctionnalités

2. Insérer Ordre de Réparation

- L'utilisateur peut insérer un nouvel ordre de réparation en fournissant le diagnostic de la panne, le nombre d'heures de main d'œuvre et l'ID de l'appareil.
- Les erreurs d'insertion sont gérées et des messages informatifs sont affichés.

RÉSULTAT



Fonctionnalités

3. Calculer Montant Total pour le Client

- L'utilisateur peut calculer le montant total pour un client en saisissant le nom du client.
- Le montant total inclut le coût de la main-d'œuvre et une taxe de 20%.

RÉSULTAT

Gestion des Réparations d'Appareils

Nom de la Cliente: Client2

2 Client2 Adresse2 V

Diagnos

Nombre d'heures de main d'œuvre:

ID de l'appareil:

Insérer Ordre de Réparation

Montant Total pour le Client: Client2

Prix Supérieur à (€):

Afficher Ordres Sans Pièces à Changer

Fonctionnalités

4. Afficher Pièces Supérieures

- L'utilisateur peut calculer le montant total pour un client en saisissant le nom du client.
- Le montant total inclut le coût de la main-d'œuvre et une taxe de 20%.

RÉSULTAT

Gestion des Réparations d'Appareils

Nom de la Cliente: Client2

1 Pièce1 20.0
2 Pièce2 25.0

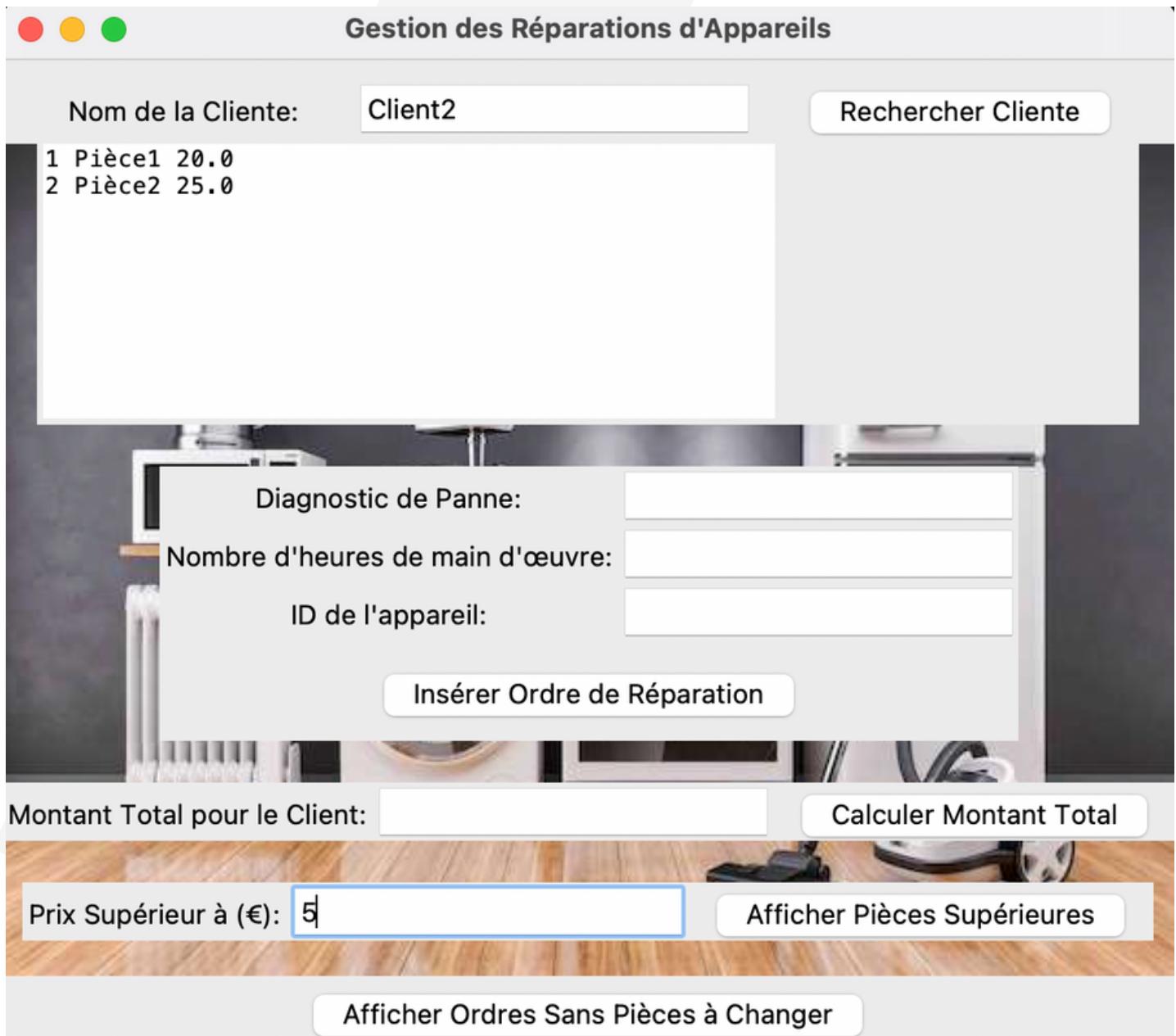
Diagnostic de Panne:

Nombre d'heures de main d'œuvre:

ID de l'appareil:

Montant Total pour le Client:

Prix Supérieur à (€):



Fonctionnalités

5. Afficher Ordres Sans Pièces à Changer

- L'utilisateur peut afficher les ordres de réparation qui ne nécessitent aucune pièce à changer.
- Les résultats sont affichés dans une zone de texte.

RÉSULTAT

Gestion des Réparations d'Appareils

Nom de la Cliente: Client2 Rechercher Cliente

3 Panne3 4.0 1
4 Panne100 6.0 2
5 Panne101 4.5 2

Diagnostic de Panne:

Nombre d'heures de main d'œuvre:

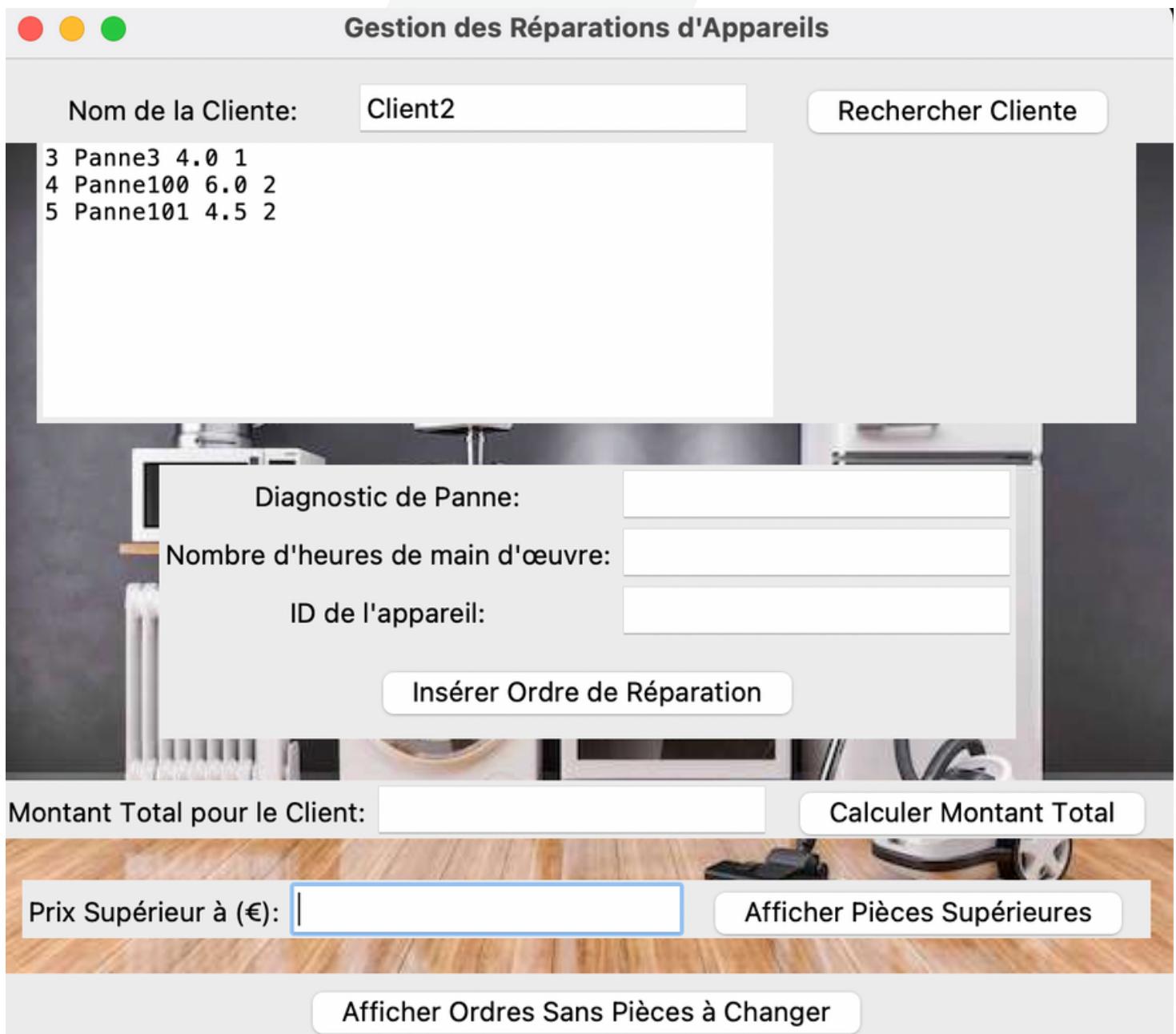
ID de l'appareil:

Insérer Ordre de Réparation

Montant Total pour le Client: Calculer Montant Total

Prix Supérieur à (€): Afficher Pièces Supérieures

Afficher Ordres Sans Pièces à Changer



Structure du code

Le code est divisé en deux fichiers principaux : “db.py“ et “main.py“.

1. db.py

- Contient des fonctions pour créer la base de données, les tables et ajouter des données de démonstration.
- Définit la structure de la base de données avec des tables telles que CLIENTE, CATEGORIE, APPAREIL, PIECE, ORDREREPARATION, et PIECESACHANGER.

Structure du code

2. main.py

- Implémente l'interface utilisateur en utilisant Tkinter.
- Comprend des fonctions pour la connexion à la base de données, l'affichage des résultats, l'insertion d'ordres de réparation, la recherche de clients, le calcul du montant total et l'affichage de pièces supérieures et d'ordres sans pièces à changer.
- Les widgets sont organisés dans des cadres pour une meilleure organisation.

Dépendances

```
1 import tkinter as tk  
2 from tkinter import messagebox  
3 from PIL import Image, ImageTk  
4 import sqlite3  
5
```

Le projet utilise les bibliothèques suivantes :

- **tkinter**: Pour la création de l'interface utilisateur.
- **PIL**: Pour la manipulation des images.
- **sqlite3**: Pour la gestion de la base de données SQLite.

Exécution

Pour exécuter le projet, il faut s'assurer d'avoir Python installé et exécutez “**main.py**”. Les dépendances doivent être installées au préalable, ce qui peut être fait en utilisant un gestionnaire de packages tel que pip.

Idées d'améliorations futures

- Interface Graphique Améliorée.
- Authentification et Autorisation.
- Gestion des Stocks de Pièces.
- Historique des Réparations

Conclusion

Le projet offre une solution de gestion des réparations d'appareils simple mais efficace. Son interface utilisateur conviviale et ses fonctionnalités de base en font un outil utile pour les petites entreprises de réparation. Les améliorations futures pourraient étendre ses capacités et améliorer son utilité globale.