



# Première Année du Cycle d'Ingénieur

## Intelligence Artificielle et Génie Informatique

**Compte Rendu de Projet POO :**

**Application de Gestion d'Hôpital**

**Réalisé par :**

Siham JDIRA  
Imane AIT-MOUH

**Encadré par :**

Pr. Mustapha HAIN

**Année Universitaire :**

2025/2026

## **INTRODUCTION GENERALE :**

La gestion d'un hôpital implique le suivi des patients, la planification des rendez-vous et la gestion du personnel médical. Dans ce contexte, il devient essentiel de disposer d'un outil numérique capable de centraliser et d'organiser efficacement ces informations. Le projet que nous présentons a donc pour objectif de proposer une solution informatique simple, intuitive et adaptée aux besoins d'un petit hôpital.

L'application développée permet de gérer les dossiers des patients et des médecins, de planifier et suivre les rendez-vous, ainsi que de faciliter la recherche et l'affichage des informations à travers une interface graphique. Ce système vise à automatiser des tâches souvent fastidieuses tout en améliorant la rapidité et la fiabilité de la gestion interne.

Ce travail s'inscrit dans le cadre du module *Python avancé* et a pour but de mettre en œuvre les compétences acquises dans la manipulation de bases de données et la conception d'interfaces graphiques avec Tkinter. Le choix de SQLite comme système de gestion de base de données assure à l'application légèreté et portabilité, tandis que Tkinter permet de créer une interface simple d'utilisation, accessible même à un personnel non technique.

En somme, cette application constitue une première étape vers la digitalisation d'un environnement médical. Elle offre un outil pratique, évolutif et maintenable, contribuant à une meilleure organisation du travail au sein d'un petit établissement de santé.

## I. Conception de la base de données :

La base de données se comporte de trois tables :

- **Table des patients :**

Champ	Type	Description
id_patient	INTEGER PK	Identifiant unique
CIN	TEXT UNIQUE	Numéro d'identification unique
nom	TEXT	Nom du patient
prenom	TEXT	Prénom du patient
date_naissance	TEXT	Date de naissance
telephone	TEXT	Numéro de téléphone

- **Table des médecins :**

Champ	Type	Description
id_medecin	INTEGER PK	Identifiant unique
nom	TEXT	Nom du médecin
prenom	TEXT	Prénom du médecin
specialite	TEXT	Spécialité médicale
telephone	TEXT	Numéro de téléphone

- **Table des rendez-vous :**

Champ	Type	Description
id_rdv	INTEGER PK	Identifiant unique
id_patient	INTEGER FK	Référence à patients
id_medecin	INTEGER FK	Référence à médecins
date_rdv	TEXT	Date du rendez-vous
heure_rdv	TEXT	Heure du rendez-vous
motif	TEXT	Motif de consultation

## II. Conception de l'interface graphique :

L'interface graphique de l'application a été développée à l'aide de la bibliothèque Tkinter, qui permet de créer des interfaces intuitives et interactives.

Elle est organisée en trois onglets principaux correspondant aux modules de gestion :

- Patients
- Médecins
- Rendez-vous

Chaque onglet contient :

- Des formulaires de saisie utilisant les widgets Entry, OptionMenu et Spinbox pour l'insertion ou la modification des données.
- Une Listbox affichant les informations existantes sous forme de liste.
- Des boutons fonctionnels pour réaliser les opérations CRUD : Ajouter, Modifier, Supprimer et Rechercher.
- Des fenêtres pop-up (toiles secondaires) pour la modification ou la recherche d'un enregistrement spécifique.

L'interface a été conçue pour être ergonomique et harmonieuse, avec un jeu de couleurs doux et une taille d'éléments adaptée pour une meilleure lisibilité. La navigation entre les différentes parties de l'application se fait via des boutons en haut de la fenêtre, permettant de passer rapidement d'un onglet à l'autre. Une fonction de recherche dynamique a également été intégrée, ouvrant une nouvelle fenêtre pour afficher les résultats correspondants.

### III. Les fonctions principales :

#### 1. Fonction « afficher(table) » :

Cette fonction affiche les enregistrements d'une table donnée dans une Listbox. Pour les rendez-vous, une jointure est utilisée pour afficher les noms complets des patients et des médecins, garantissant une meilleure lisibilité.

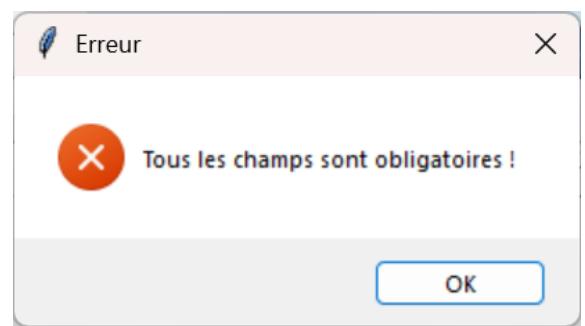
ID	Nom	Prénom	CIN	Date Naissance	Téléphone
1	NADIR	Hamza	BA123456	9-7-2001	0654231677
2	JDIRA	Siham	BH42356	20-9-2004	0754231566
3	AIT-MOUH	Imane	CA324567	16-8-2005	0654231567
4	ALAoui	Amina	BH143256	24-11-1995	0642315677
5	BAGHIR	Reda	JA324516	14-10-1998	0614326678

#### 2. Fonction « ajouter(table) » :

La fonction ajouter(table) permet d'insérer de nouveaux patients, médecins ou rendez-vous dans la base tout en contrôlant la validité des données. Des messages d'erreur clairs via showerror() préviennent l'utilisateur en cas de l'un de ces problèmes :

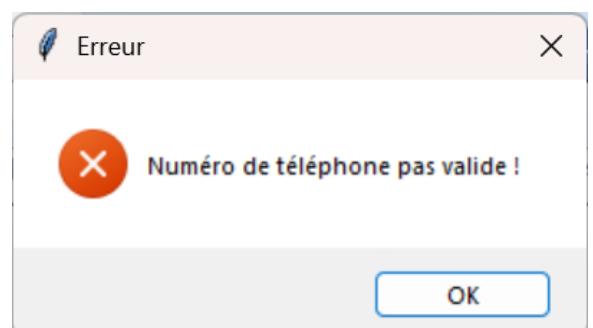
##### a) Champs obligatoires :

Avant l'ajout, la fonction vérifie que tous les champs sont remplis. Si un champ est vide, elle utilise la fonction **showerror()** de tkinter.messagebox pour afficher un message d'erreur et empêche l'enregistrement.



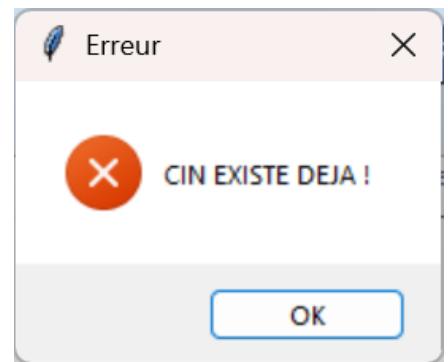
##### 2. Vérification du numéro de téléphone :

Doit contenir **10 chiffres** et commencer par **05, 06 ou 07** pour assurer la validité du format téléphonique marocain. En cas de format incorrect, **showerror()** affiche un message d'erreur pour informer l'utilisateur.



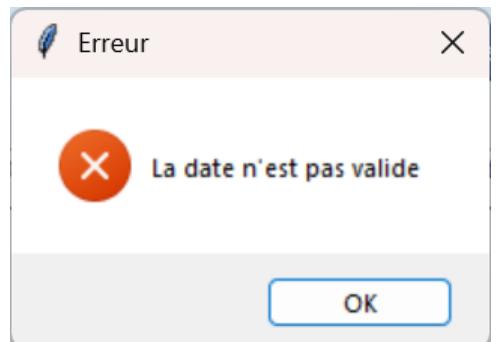
### 3. Contrôle du CIN (doublons) :

Pour éviter les doublons, la fonction utilise **db.execute()** (SQLite) pour interroger la base de données et **fetchone()** pour vérifier si le CIN existe déjà. Si c'est le cas, **showerror()** affiche un message et l'ajout est bloqué.



### 4. Validation des dates :

Pour contrôler les dates, la fonction utilise **datetime.now()** (après avoir importé la bibliothèque **datetime**) pour comparer les dates de naissance ou de rendez-vous. Si une date est invalide (Date de naissance future ou date de rendez-vous passé), **showerror()** affiche un message et l'insertion est empêchée.

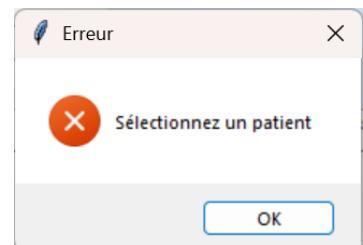


### 3. Fonction « modifier(table) » :

La fonction **modifier(table)** met à jour les informations des patients, médecins ou rendez-vous. Après sélection et ouverture d'une popup, elle applique **les mêmes vérifications que la fonction ajouter** : champs obligatoires remplis, format des téléphones correct, contrôle du CIN (doublons) et dates valides. Les erreurs sont signalées avec **showerror()** pour garantir des données cohérentes.

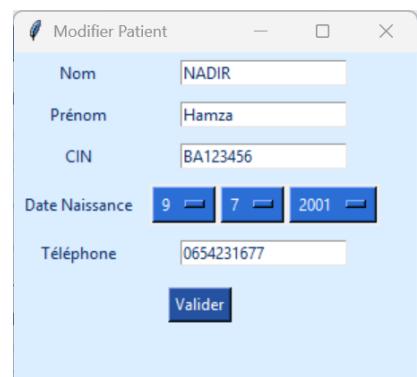
#### a) Vérification de la sélection

La fonction vérifie d'abord si un élément est sélectionné dans la liste correspondante (Listbox) avec la fonction **curselection()**. Si aucune sélection n'est faite, **showerror()** affiche un message d'erreur pour prévenir l'utilisateur.



#### b) Ouverture et validation dans la popup

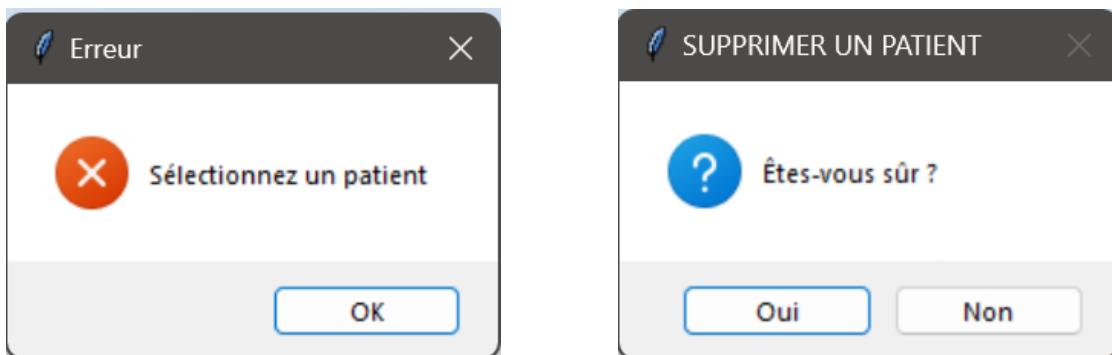
La fonction crée une fenêtre **Toplevel()** avec des widgets (**Entry**, **OptionMenu**, **Spinbox**) pour modifier les informations. Après validation, les changements sont enregistrés avec **db.execute()**, confirmés par **db.commit()**, la liste est actualisée via **afficher()**, et la popup est fermée avec **destroy()**.



#### *4.Fonction « supprimer(table) » :*

Cette fonction supprime un enregistrement après une confirmation de l'utilisateur.

Elle gère les cas où aucune ligne n'est sélectionnée afin d'éviter les erreurs d'exécution.



#### *5. Fonction « rechercher(table) » :*

La fonction rechercher(table) permet d'effectuer une recherche selon différents critères :

- Pour les patients : nom, prénom, CIN.

A search interface for patients. It features three input fields labeled "Nom" (Name), "Prénom" (First name), and "CIN" (Social Insurance Number) with corresponding text input boxes. To the right of these fields is a blue "Rechercher" (Search) button.

- Pour les médecins : nom, spécialité.

A search interface for doctors. It features three input fields labeled "Nom du medecin" (Doctor's name), "Prénom du medecin" (First name of doctor), and "Spécialité du medecin" (Specialty of doctor), each with a corresponding text input box. To the right of the last field is a blue "Rechercher" (Search) button. Below the "Spécialité" field is a button labeled "Sélectionner une Spécialité" (Select a specialty).

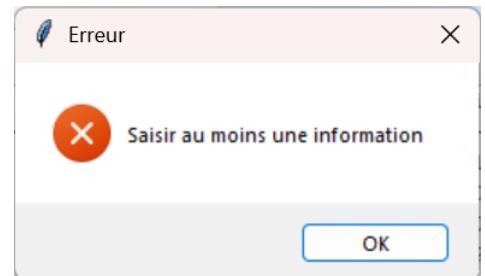
- Pour les rendez-vous : date, patient, médecin.

A search interface for appointments. It features three input fields labeled "ID Patient" (Patient ID), "ID Médecin" (Doctor ID), and "Date" (Date), each with a corresponding text input box. To the right of the last field is a blue "Rechercher" (Search) button.

Les résultats sont affichés dans une nouvelle fenêtre sous forme de liste :

ID	Nom	Prénom	Spécialité	Téléphone
3	KADIR	Laila	Gynécologue/Obstétricien	0716524356
5	NASIM	Jamal	Gynécologue/Obstétricien	0514235677

Si tous les champs de recherche sont vides, la fonction déclenche **showerror()** pour avertir l'utilisateur qu'il doit saisir un mot-clé avant de lancer la recherche.



## IV. L'interface graphique :

- Onglet Patients : formulaire de saisie et liste des patients.

ID	Nom	Prénom	CIN	Date Naissance	Téléphone
1	NADIR	Hamza	BA123456	9-7-2001	0654231677
2	JDIRA	Siham	BH42356	20-9-2004	0754231566
3	AIT-MOUH	Imane	CA324567	16-8-2005	0654231567
4	ALAOUI	Amina	BH143256	24-11-1995	0642315677
5	BAGHIR	Reda	JA324516	14-10-1998	0614326678

- Onglet Médecins : gestion des informations médicales.

Gestion Hôpital

Patients		Médecins		Rendez-vous
Nom du medecin				
Prénom du medecin				<input type="button" value="Ajouter le medecin"/>
Spécialité du medecin	<input type="button" value="Sélectionner une Spécialité"/>			
Téléphone du medecin				
----- Recherche -----				
Nom du medecin				
Prénom du medecin				<input type="button" value="Rechercher"/>
Spécialité du medecin	<input type="button" value="Sélectionner une Spécialité"/>			
ID	Nom	Prénom	Spécialité	Téléphone
1	NAJJAR	Hanane	Dentiste	0534251677
2	ALLAM	Amine	Médecin généraliste	0614256678
3	KADIR	Laila	Gynécologue/Obstétricien	0716524356
4	SAMIR	Najoua	Psychiatre	0615423167
5	NASIM	Jamal	Cardiologue	0514235677

- Onglet Rendez-vous : affichage complet avec noms liés.

Gestion Hôpital

Patients		Médecins		Rendez-vous	
ID Patient					
ID Médecin					
Date	<input type="button" value="1"/>	<input type="button" value="1"/>	<input type="button" value="2025"/>	<input type="button" value="Ajouter le rendez-vous"/>	
Heure	08 : 00				
Motif					
----- Recherche -----					
ID Patient					
ID Médecin				<input type="button" value="Rechercher"/>	
ID	Patient	Médecin	Date	Heure	Motif
1	JDIRA Siham	NAJJAR Hanane	24-12-2025	10:30	Consultation
2	NADIR Hamza	KADIR Laila	15-11-2025	12:00	Contrôle
3	ALAOUI Amina	ALLAM Amine	21-1-2026	13:45	Bilan médical
4	AIT-MOUH Imane	SAMIR Najoua	27-11-2025	9:30	Consultation
5	BAGHIR Reda	NASIM Jamal	3-12-2025	11:00	Contrôle

## V. Conclusion :

Ce projet a permis de mettre en pratique les compétences en programmation Python, en particulier dans les domaines suivants :

- Conception d'interfaces graphiques avec Tkinter.
- Manipulation de bases de données SQLite pour la gestion des informations.
- Validation et traitement des données utilisateurs.
- Organisation du code autour de fonctions réutilisables pour les opérations CRUD.

L'application obtenue est fonctionnelle, ergonomique et évolutive, offrant une solution complète pour la gestion d'un hôpital. Elle constitue une expérience formatrice, alliant la rigueur de la programmation à la créativité de la conception d'interface.

Nous remercions notre professeur **Mustapha HAIN** de nous avoir confié ce projet, qui nous a aidés à renforcer nos connaissances et à les appliquer de manière concrète.

## VI. Références :

- Lien GitHub : <https://github.com/Imane123DEV/Projet-POO-Imane-Ait-mouh-et-Siham-Jdira/tree/main>
- Vidéo de présentation de l'interface graphique :  
<https://drive.google.com/file/d/1Tb53-RvXZUuhe4kd59-agIkWiEveEmLm/view?usp=sharing>