



Université  
de Limoges

## PROJET INFORMATIQUE – S4

### ■ GENEALOGIE FAMILIALE

REALISE PAR :

DIALLO Fatimatou

DIALLO Mamadou Talibe

BALDE Alpha Oumar

BALDE Mamadou Oury

Le, 17/05/2025

ENCADRE PAR :

M. Christophe CLAVIER,

Faculté des Sciences et Techniques de Limoges

# TABLE DES MATIÈRES

TABLE DES MATIÈRES.....	2
INTRODUCTION GÉNÉRALE .....	3
<b>CHAPITRE I : CONTEXTE GÉNÉRAL</b>	
1. INTRODUCTION.....	3
2. CAHIER DES CHARGES .....	4
3. CONCLUSION .....	5
<b>CHAPITRE II : ETUDE FONCTIONNELLE</b>	
1. INTRODUCTION.....	5
2. PREMIERE PARTIE (VERSION LIGNE DE COMMANDE) .....	5
3. DEUXIEME PARTIE (VERSION IHM) .....	5
4. CONCLUSION.....	6
<b>CHAPITRE III : ETUDE TECHNIQUE</b>	
1. INTRODUCTION .....	6
2. OUTIL UTILISÉ .....	6
3. CLASSES PRINCIPALES .....	7
4. CONTRAINTES DE VALIDITÉ .....	8
5. CONCLUSION .....	8
<b>CHAPITRE IV : REALISATION</b>	
1. INTRODUCTION.....	9
2. PROCESSUS D'UTILISATION.....	9
3. CONCLUSION.....	31
CONCLUSION GENERALE .....	34

# INTRODUCTION GENERALE

Dans le cadre du projet informatique du quatrième semestre à l'Université de Limoges, nous avons été amenés à développer une application logicielle dédiée à la gestion et à la visualisation de données généalogiques. Ce projet s'inscrit dans une démarche pédagogique, permettant d'appliquer les compétences acquises en programmation et en gestion de bases de données à une problématique concrète.

Le projet porte sur la conception d'un **logiciel de généalogie familiale**, permettant à un utilisateur de créer, visualiser et gérer l'arbre généalogique de sa famille. L'application vise à centraliser les informations personnelles des membres d'une famille (noms, prénoms, dates et lieux de naissance, liens familiaux, etc.), et à les représenter de manière structurée afin de préserver l'histoire familiale.

L'objectif principal est de proposer un outil simple, intuitif et accessible, ne nécessitant pas de connaissances techniques particulières. L'utilisateur peut, via une interface graphique conviviale, saisir des individus, établir des relations de mariage ou de filiation, visualiser l'organisation familiale, et enregistrer l'ensemble de ces informations dans une base de données locale.

Pour mener à bien ce projet, nous avons fait le choix d'utiliser :

- **Python**, pour son accessibilité, sa lisibilité et sa puissance dans le prototypage rapide ;
- **Tkinter**, bibliothèque graphique intégrée à Python, pour concevoir l'interface utilisateur ;
- **SQLite**, une base de données relationnelle légère et embarquée, idéale pour un usage local.

Ce rapport est structuré en plusieurs parties. Après avoir posé le **contexte général** du projet, nous aborderons l'**étude fonctionnelle**, qui précise les besoins utilisateurs et les fonctionnalités prévues. Nous enchaînerons avec l'**étude technique**, qui justifie les choix d'outils et présente l'architecture logicielle. La **partie de réalisation** décrira en détail le processus de développement, accompagné de captures illustrant l'évolution de l'application. Enfin, nous conclurons par une **synthèse globale** du travail accompli, en évoquant les limites actuelles et les perspectives d'amélioration futures.

Ce projet constitue ainsi une mise en œuvre concrète de nos acquis techniques, dans un cadre réaliste, tout en répondant à un besoin intemporel : celui de conserver et de transmettre la mémoire familiale.

## CHAPITRE I : CONTEXTE GENERAL

### 1) INTRODUCTION

Ce chapitre introduit le projet en présentant son cadre, ses objectifs et les besoins auxquels il répond. Il résume les grandes lignes du cahier des charges, les principales fonctionnalités attendues et les défis identifiés. Cette mise en contexte prépare à la compréhension des aspects techniques et fonctionnels développés dans les chapitres suivants.

## **2) CAHIER DES CHARGES**

### **1. Contexte du projet**

#### *1.1. Présentation générale*

Le projet consiste en la création d'un logiciel de généalogie familiale. Ce logiciel permet aux utilisateurs de créer un compte afin de gérer et visualiser l'arbre généalogique de leur famille. L'application permet aux utilisateurs d'ajouter des membres de la famille, de définir des relations de parenté, et de visualiser graphiquement l'arbre généalogique.

#### *1.2. Objectifs principaux*

- Permettre aux utilisateurs de créer, supprimer et gérer un arbre généalogique personnalisé.
- Offrir une représentation visuelle claire des liens familiaux.
- Garantir la sauvegarde et la confidentialité des données personnelles.
- Faciliter l'accès et l'organisation des informations généalogiques.

### **2. Problématiques**

Plusieurs défis doivent être résolus lors du développement de ce logiciel :

- **Ergonomie de l'interface graphique** : Le logiciel doit être simple à utiliser, même pour les utilisateurs non techniques. L'interface graphique doit être claire, intuitive et facilement navigable.
- **Gestion et organisation des données** : Le projet implique la gestion de nombreuses informations (noms, prénoms, relations familiales, etc.), il est essentiel d'organiser les données de manière structurée et efficace dans la base de données.
- **Performance du système** : Le logiciel doit être performant, même avec un grand nombre de membres ajoutés à l'arbre généalogique. Le traitement des informations et la génération du graphique doivent se faire rapidement.
- **Sécurité des données** : La confidentialité des données des utilisateurs est primordiale. Il faut s'assurer que seules les personnes autorisées puissent accéder et modifier les informations personnelles.

### **3. Fonctionnalités de l'application**

- **Création, gestion et suppression de comptes** : L'utilisateur peut créer, gérer ou supprimer son compte, ce qui lui permet de sécuriser et personnaliser sa généalogie familiale.
- **Ajout, recherche, modification des membres** : L'utilisateur peut ajouter, rechercher et modifier les informations des membres de sa famille, afin de refléter leur évolution au fil du temps (naissance, décès, etc.).
- **Définition des relations familiales** : L'application permet de créer des liens entre les membres de la famille : mariages, enfants, parents.
- **Visualisation de l'arbre généalogique** : Une interface graphique est proposée pour afficher de manière interactive et structurée les relations familiales.
- **Sauvegarde des données** : Toutes les données relatives aux membres de la famille sont stockées dans une base de données SQLite, garantissant la persistance des données d'une session à l'autre.

### 3) CONCLUSION

Ce chapitre a présenté une vue d'ensemble du projet, en mettant en évidence ses objectifs, les fonctionnalités attendues ainsi que les principales problématiques rencontrées. Il a également exposé les choix faits pour structurer le développement, depuis la mise en place de la base de données jusqu'à l'élaboration de l'interface graphique, posant ainsi les fondations nécessaires à la suite du rapport.

## CHAPITRE II : ETUDE FONCTIONNELLE

### 1. Introduction

Cette partie a pour objectif de détailler les fonctionnalités du logiciel de généalogie selon deux modes d'utilisation : une version en ligne de commande, et une version avec interface graphique. Ces deux versions répondent aux mêmes besoins fonctionnels, mais proposent une interaction différente avec l'utilisateur.

### 2. Partie 1 : Version ligne de commande

Cette version permet à l'utilisateur de gérer l'arbre généalogique via une interface textuelle.

► *Fonctionnalités proposées :*

- Création d'un compte utilisateur
- Ajout, recherche et modification de membres
- Définition des liens familiaux (parent, enfant, conjoint)
- Affichage de l'arbre généalogique sous forme textuelle
- Sauvegarde et chargement des données via une base de données SQLite

► *Cas d'utilisation typique :*

- L'utilisateur lance le programme → saisit son identifiant → accède à son arbre → ajoute un nouveau membre → enregistre les modifications.

### 3. Partie 2 : Version interface graphique

Cette version propose une interaction plus intuitive avec le logiciel grâce à une interface graphique développée avec Tkinter. Elle répond aux objectifs d'accessibilité et de visualisation mis en avant dans le contexte général du projet. L'utilisateur peut gérer son arbre généalogique de manière visuelle, sans avoir besoin de manipulations techniques complexes.

► *Fonctionnalités proposées :*

- Interface de connexion avec formulaire pour créer ou supprimer un compte
- Boutons de commande pour ajouter, modifier ou rechercher un membre
- Établissement des relations familiales (parent, enfant, conjoint)
- Affichage graphique de l'arbre généalogique sous forme d'arbres ou de listes
- Sauvegarde et chargement automatiques des données dans une base SQLite locale

► **Avantages :**

- Représentation claire des liens familiaux grâce à l'arbre visuel
- Utilisation simplifiée pour les personnes non techniques
- Navigation fluide entre les différentes fonctionnalités du logiciel

#### 4. Conclusion

L'étude fonctionnelle a permis d'identifier les principales fonctionnalités du logiciel, à la fois dans sa version console et graphique. Ces éléments sont essentiels pour guider le développement technique tout en garantissant une expérience utilisateur claire et cohérente.

## CHAPITRE III : ETUDE TECHNIQUE

### 1. INTRODUCTION

L'étude technique décrit les choix technologiques et l'architecture du projet. Elle présente les outils utilisés, ainsi que la structure du code, notamment les classes principales qui modélisent les éléments du système (Individu et Famille). Ces classes et leurs méthodes sont conçues pour permettre une gestion fluide des informations relatives à la généalogie familiale.

### 2. Outils utilisés

Les principaux outils utilisés dans ce projet sont les suivants :

- **Python** : Langage principal choisi pour sa simplicité, sa lisibilité et la richesse de ses bibliothèques.
- **Tkinter** : Bibliothèque graphique utilisée pour l'interface utilisateur. Elle permet de créer des fenêtres et des widgets interactifs pour la visualisation de l'arbre généalogique.
- **SQLite** : Base de données légère intégrée pour stocker les informations des individus et des familles, sans nécessiter de serveur externe.

Ces technologies sont bien adaptées à la création d'une application de généalogie locale et légère.

### 3. Classes principales

#### 3.1. Classe Individu

La classe **Individu** représente chaque membre d'une famille dans l'arbre généalogique. Elle contient des informations personnelles ainsi que des attributs pour gérer les relations avec les autres membres.

#### Attributs :

- **id\_individu** : Identifiant unique pour chaque individu (clé primaire dans la base de données).
- **nom** : Nom de l'individu.
- **prenom** : Prénom de l'individu.
- **sexe** : Sexe de l'individu.

- **date\_naissance** : Date de naissance de l'individu.
- **lieu\_naissance** : Lieu de naissance de l'individu.
- **occupation** : Profession de l'individu.
- **deces** : Statut de décès, avec par défaut "Non". Peut être mis à "Oui" si l'individu est décédé.
- **date\_deces** : Date du décès, si applicable.
- **lieu\_deces** : Lieu du décès, si applicable.
- **ids\_famille\_formes** : Liste d'identifiants de familles auxquelles cet individu appartient (peut contenir plusieurs références de famille).

## Méthodes principales :

- **saisir\_conjoint\_e()** : Permet de saisir les informations du conjoint ou de la conjointe.
- **saisir\_enfant()** : Permet de saisir les informations d'un enfant.
- **ajouter\_individu()** : Permet d'ajouter un nouvel individu à la base de données.
- **modifier\_individu()** : Permet de modifier les informations d'un individu existant.
- **afficher\_individu()** : Affiche les informations de l'individu (nom, prénom, date de naissance, etc.).
- **afficher\_ancestor()** : Permet d'afficher sous forme d'arbres ou listes les descendants d'un individu.
- **afficher\_descendant()** : Affiche sous forme d'arbres ou de listes les descendants d'un individu.

Cette classe joue un rôle central dans la gestion des membres d'une famille et permet de maintenir une organisation structurée des individus.

### 3.2. Classe Famille

La classe **Famille** représente l'union entre un homme et une femme, ainsi que leurs enfants. Elle permet de gérer les relations familiales et les informations liées aux mariages, divorces et enfants.

#### Attributs :

- **id\_famille** : Identifiant unique de la famille.
- **date\_mariage** : Date du mariage des deux membres de la famille (le couple).
- **lieu\_mariage** : Lieu où le mariage a eu lieu.
- **divorce** : Statut de divorce.
- **date\_divorce** : Date du divorce, si applicable.
- **id\_conjoint** : Identifiant du conjoint (la personne avec qui l'individu est marié).
- **id\_conjointe** : Identifiant de la conjointe (la personne avec qui l'individu est marié).
- **ids\_enfants** : Liste des identifiants des enfants (une famille peut avoir plusieurs enfants).

## Méthodes principales :

- **ajouter\_premiere\_famille()** : Permet de créer la première famille, en ajoutant les informations du mariage, des conjoints et des enfants.
- **ajouter\_famille\_a\_un\_individu()** : Crée une famille par rapport à un individu recherché.
- **ajouter\_famille\_a\_deux\_individu()** : Crée une famille par rapport à deux individus.
- **ajouter\_parents()** : Ajoute des parents à un individu.
- **ajouter()** : Ajoute la famille dans la base de données.
- **modifier\_famille()** : Permet de modifier les informations d'une famille, telles que la date de mariage, les enfants, etc.

- **afficher\_famille()** : Affiche les informations relatives à la famille, y compris les conjoints, les enfants, la date de mariage, etc.

Cette classe permet de structurer les relations familiales et de gérer les différents types de liens entre les individus (mariages, divorces, enfants).

### *3.3. Architecture et interaction des classes*

Les classes **Individu** et **Famille** interagissent étroitement pour modéliser les liens familiaux. Un individu peut appartenir à plusieurs familles, et une famille est constituée de plusieurs individus (les conjoints et leurs enfants). Les classes sont reliées par des identifiants uniques, ce qui permet de lier chaque individu à sa famille respective et vice versa.

Le système utilise une base de données SQLite pour stocker ces informations de manière persistante, et l'interface graphique (Tkinter) permet à l'utilisateur de manipuler facilement ces données à travers des formulaires et des visualisations interactives.

### *4. Contraintes de validité*

Lors du développement du logiciel, plusieurs règles de cohérence ont été mises en place pour garantir la validité et la fiabilité des données enregistrées. Ces contraintes permettent d'éviter les erreurs logiques et assurent une modélisation réaliste des événements familiaux. Parmi les principales contraintes :

- **Validité des dates :**
  - La date de décès ne peut pas être antérieure à la date de naissance (si applicable).
  - La date de mariage ne peut pas être postérieure à la date de décès d'un des conjoints (si applicable).
  - Un enfant ne peut pas naître avant la date de naissance de ses parents.
  - Un individu ne peut pas se marier avant un certain âge.
- **Unicité des identifiants :**
  - Chaque individu et chaque famille possède un identifiant unique dans la base de données afin de garantir la traçabilité des relations.
- **Complétude des champs obligatoires :**
  - Certaines informations sont requises pour qu'un individu soit valide (nom, sexe).
  - Pour une famille, il faut au minimum deux conjoints identifiés pour pouvoir ajouter des enfants.

Ces règles de validation, bien qu'implémentées uniquement dans le code Python, jouent un rôle essentiel dans la fiabilité de l'application. Elles permettent de garantir la cohérence logique des données généalogiques, en encadrant les saisies utilisateur et en empêchant des situations aberrantes (comme une naissance après un décès ou un enfant plus âgé que ses parents).

### *5. Conclusion*

Cette étude technique a permis de définir les fondations du projet, tant sur le plan des outils que de l'architecture logicielle. Les classes **Individu** et **Famille**, associées aux contraintes de cohérence intégrées dans le code, assurent une gestion fiable des données généalogiques. L'ensemble constitue une base claire et évolutive pour le logiciel.

# CHAPITRE IV : REALISATION

## 1. INTRODUCTION

Cette partie présente la réalisation de l'application graphique du logiciel de généalogie. L'interface a été développée à l'aide de la bibliothèque Tkinter afin de proposer une expérience utilisateur simple, intuitive et accessible. Les différentes fonctionnalités sont accessibles via des formulaires, des boutons d'action et des fenêtres dédiées, permettant une gestion fluide des individus et des familles.

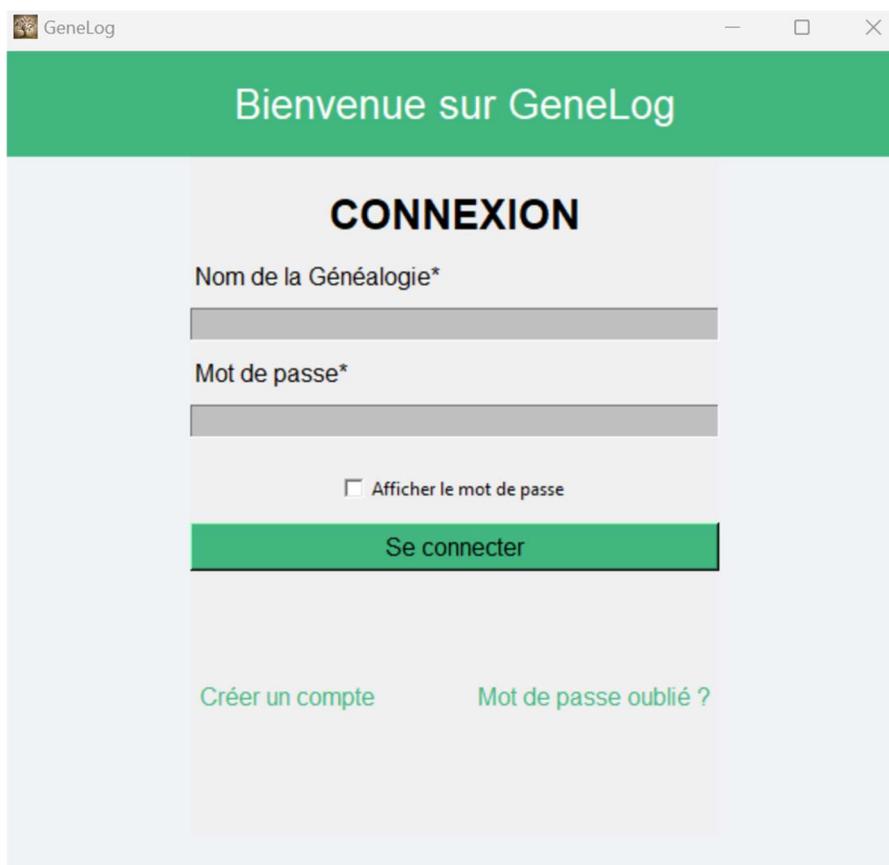
## 2. PROCESSUS D'UTILISATION

Cette section décrit les différentes étapes d'utilisation du logiciel. Chaque fonctionnalité est accessible via des boutons et des formulaires, facilitant la gestion des individus et des familles dans l'arbre généalogique.

### 2.1 Connexion, création du compte, Mot de passe oublier

#### a – Connexion :

À l'ouverture de l'application, l'utilisateur arrive directement sur la page de connexion. Elle propose trois actions : se connecter à un compte existant, créer un nouveau compte, ou modifier son mot de passe en cas d'oubli.



## b – Création de compte :

Lorsqu'un utilisateur clique sur "Créer un compte", il est redirigé vers une page d'inscription.

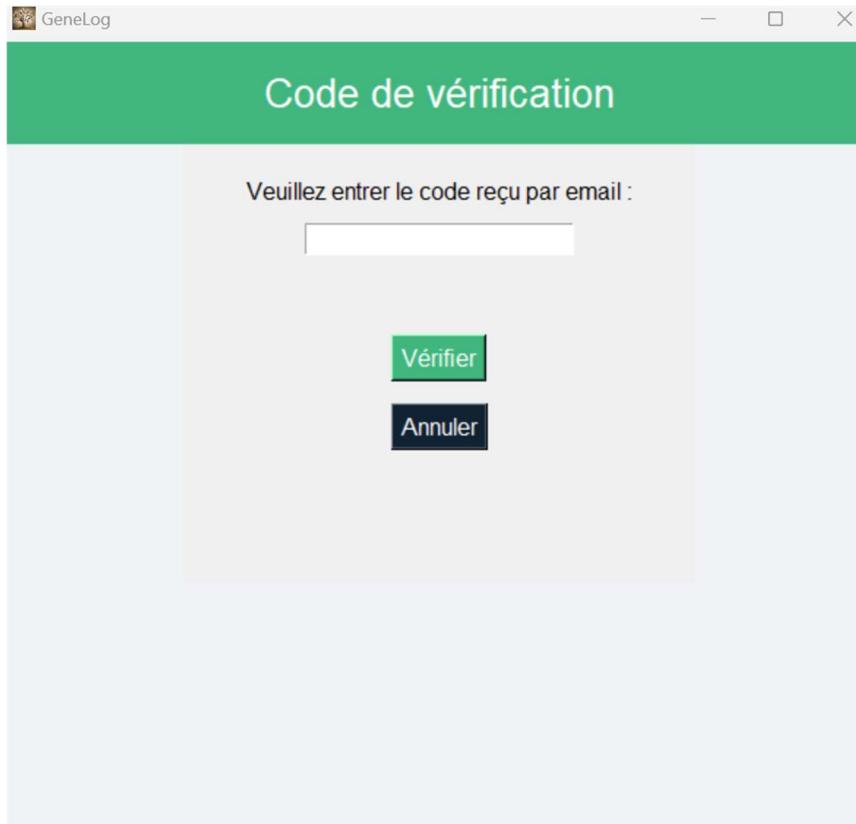
The screenshot shows a registration form titled "INSCRIPTION". It includes fields for "Nom de la Généalogie\*", "Mot de passe\*", "Confirmer le mot de passe\*", and "Email\*". There is also a checkbox for "Afficher le mot de passe". At the bottom are "Créer le compte" and "Retour" buttons.

## c – Mot de passe oublier :

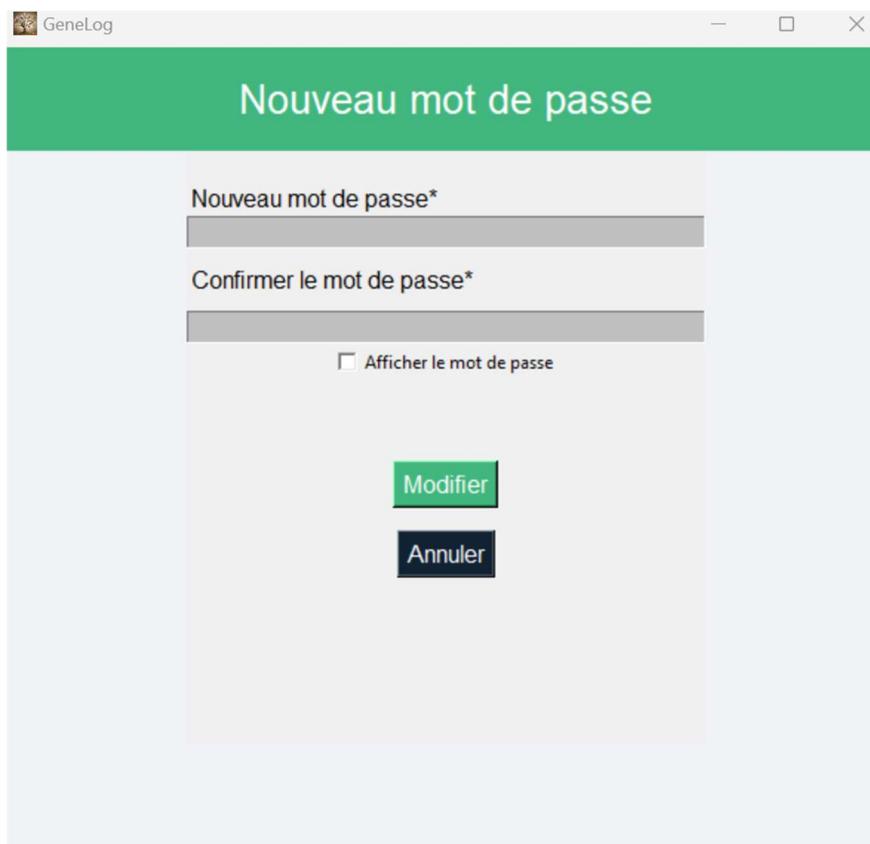
- *Saisie du nom de la généalogie et de l'adresse e-mail*

The screenshot shows a password reset form titled "Réinitialisation du mot de passe". It includes fields for "Nom de la Généalogie\*" and "Email\*". At the bottom are "Envoyer le code" and "Annuler" buttons.

- *Code de vérification envoyé par mail*



- *Modification sécurisée du mot de passe*



## 2.2 Page d'accueil

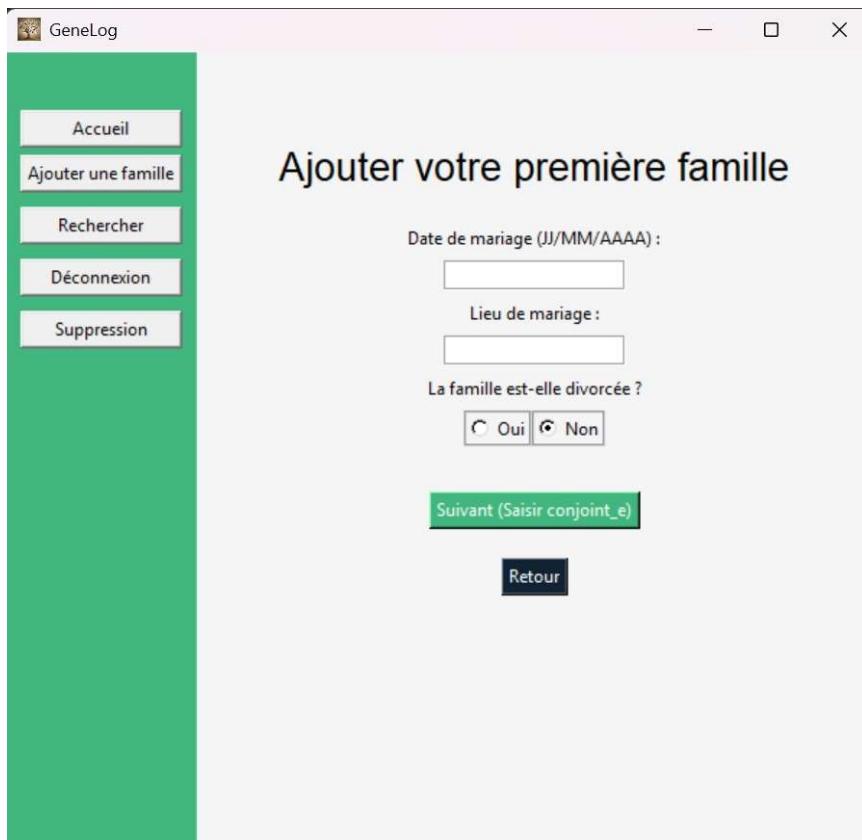
Une fois connecté, l'utilisateur accède à la page d'accueil, qui lui offre une vue d'ensemble de son espace personnel.



## 2.3 Ajout d'un individu

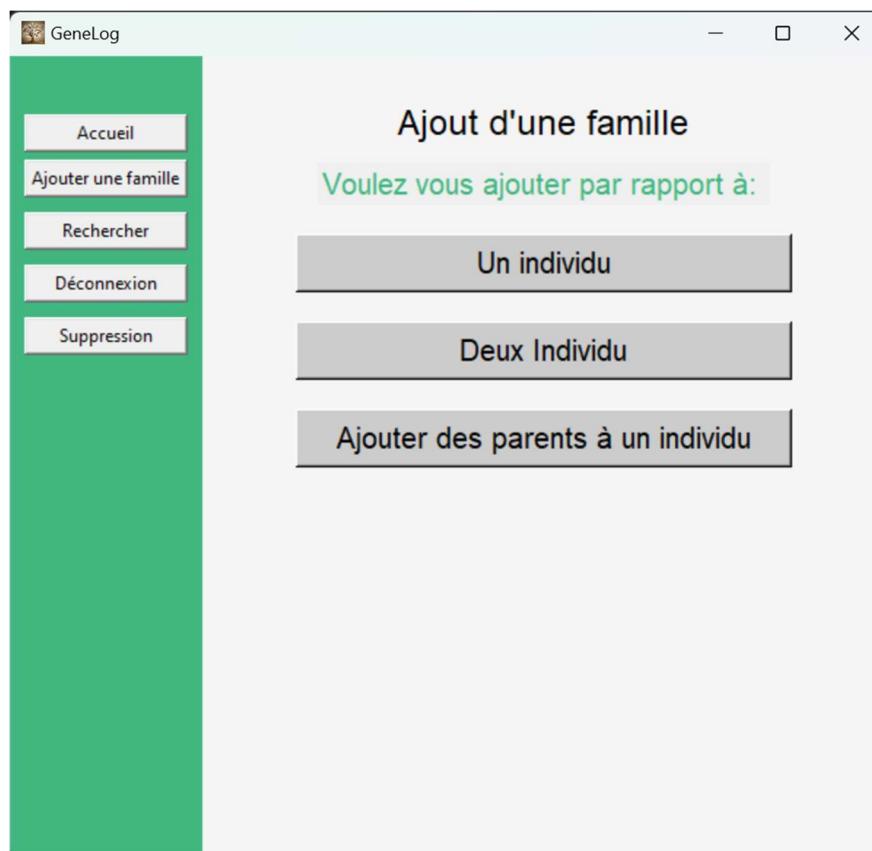
L'ajout d'un individu s'effectue via un formulaire de saisie comportant les informations de base : nom, prénom, sexe, date et lieu de naissance, profession, etc.

**a - Ajout de la première famille :** (après cette page tout est identique avec ajout par rapport à un individu)



**b - Ajout d'une famille par rapport à un individu :**

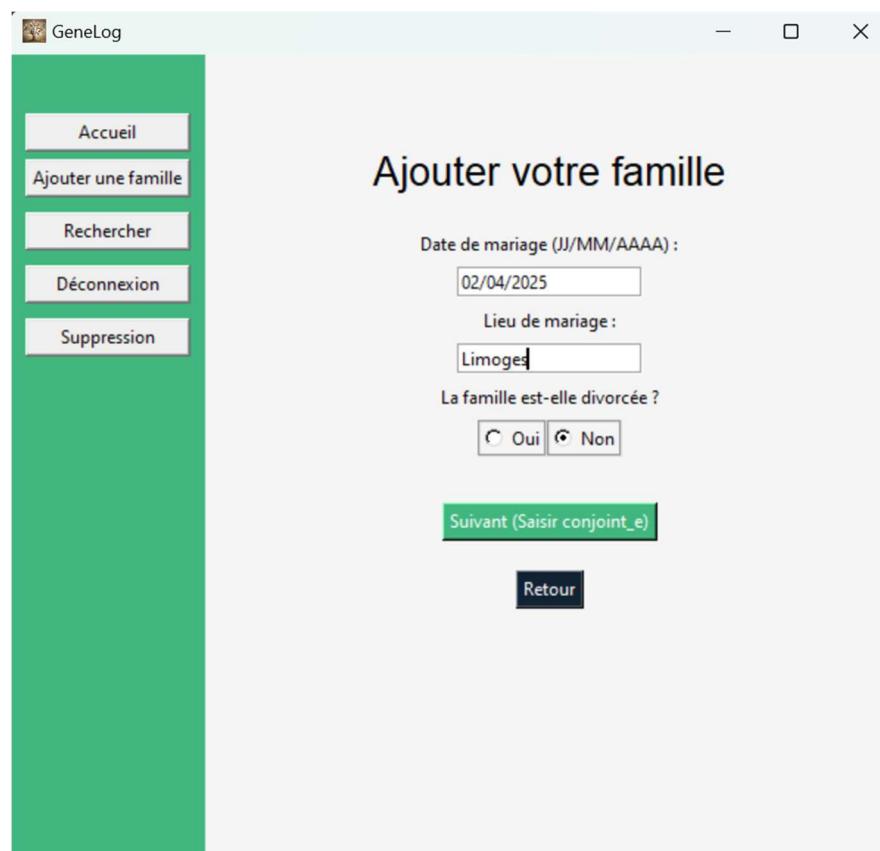
- 1) Choisir ajouter par rapport à un individu



2) Chercher l'individu auquel on veut ajouter une famille



3) Si l'âge supérieure à 15ans on passe à la prochaine étape (lui ajouter la famille)



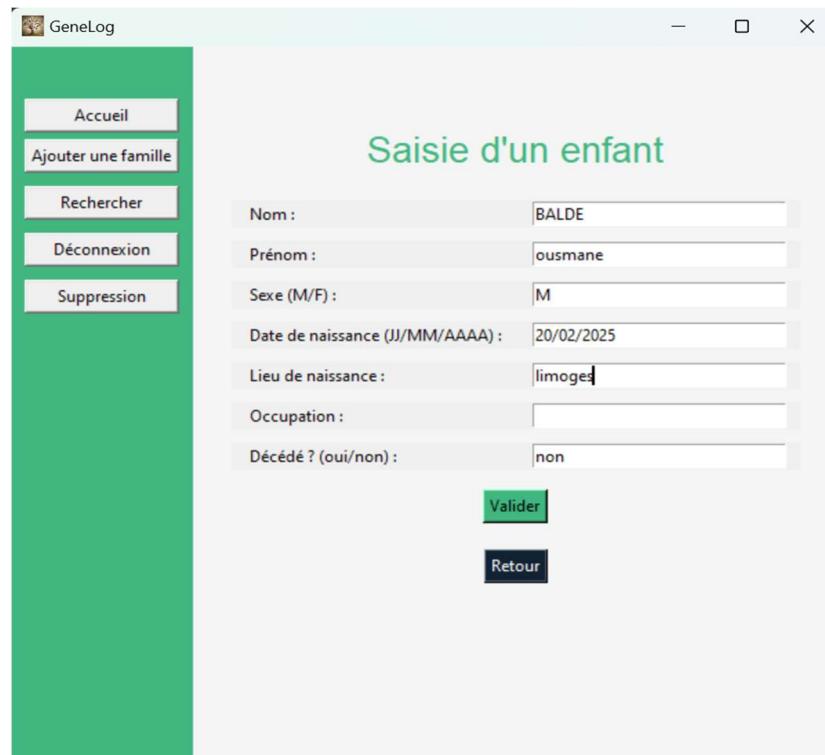
- 4) Si toutes les contraintes des dates sont respectées alors on passe à la saisie de la conjointe

The screenshot shows the GeneLog software interface. On the left, there is a vertical green sidebar with white buttons labeled: Accueil, Ajouter une famille, Rechercher, Déconnexion, and Suppression. The main window title is "Saisie de la conjointe". Inside, there are several input fields: Nom\* : Diallo, Prénom : Mariam, Date de naissance (JJ/MM/AAAA) : 20/02/2000, Lieu de naissance : Paris, Occupation : avocate, and Décédé ? (oui/non) : non. Below these fields are two buttons: "Valider" (in green) and "Retour" (in black).

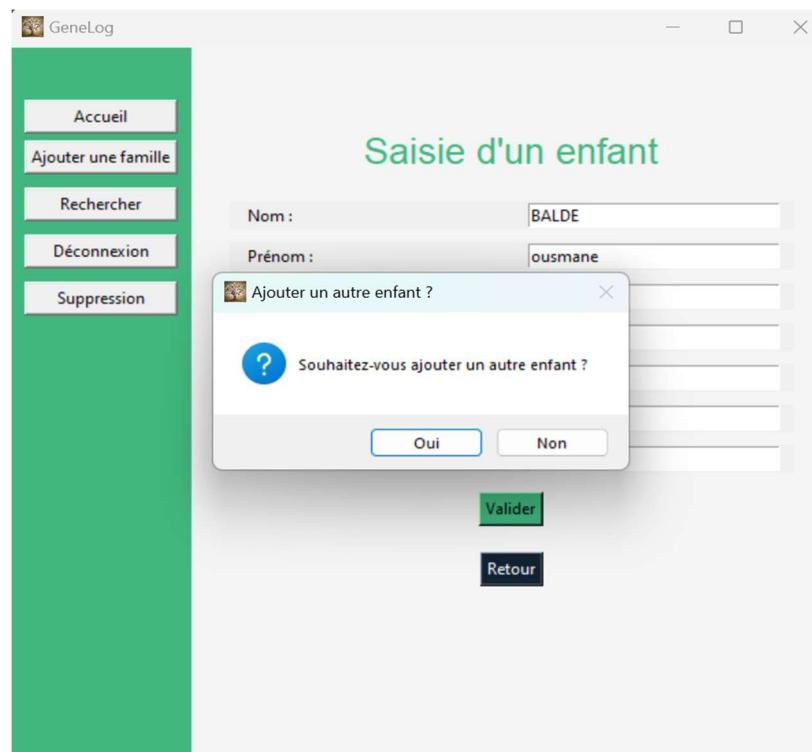
- 5) Une fois terminer un petit messageshow s'affiche en demandant à l'utilisateur s'il veut ajouter un enfant au couple.

The screenshot shows the GeneLog software interface with a modal dialog box overlaid. The dialog has a question mark icon and the text "Souhaitez-vous ajouter un enfant ?". It contains two buttons: "Oui" (Yes) and "Non" (No). In the background, the "Saisie de la conjointe" screen is visible, showing the same data as the previous screenshot. The "Valider" button is partially visible at the bottom of the main window.

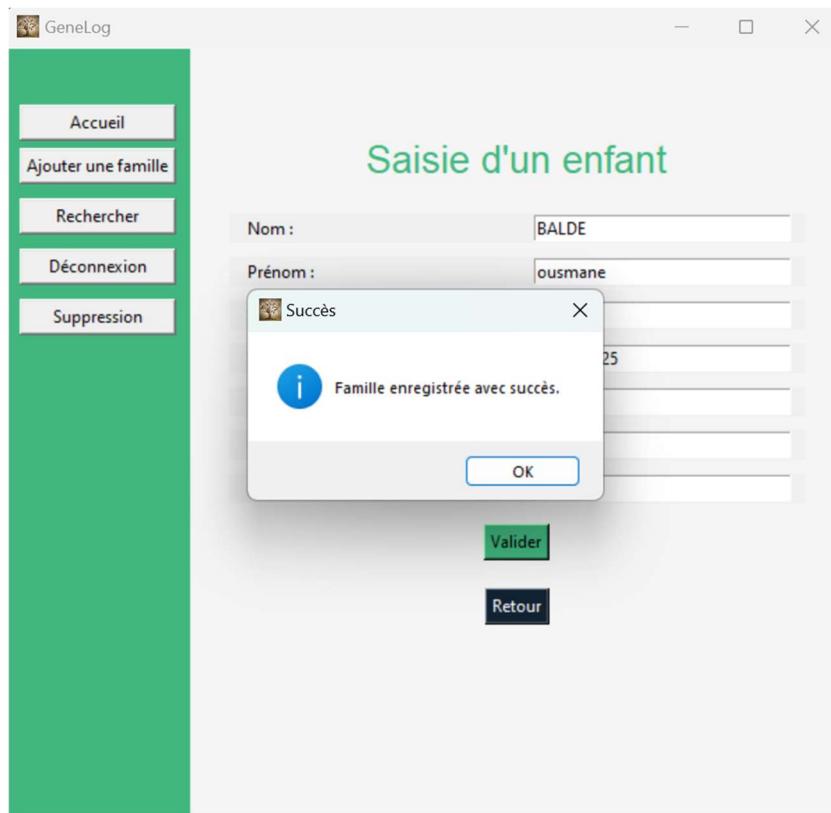
- 6) Si oui, alors on l'affiche une nouvelle fenêtre pour qu'il saisisse les informations de l'enfant qu'il veut ajouter



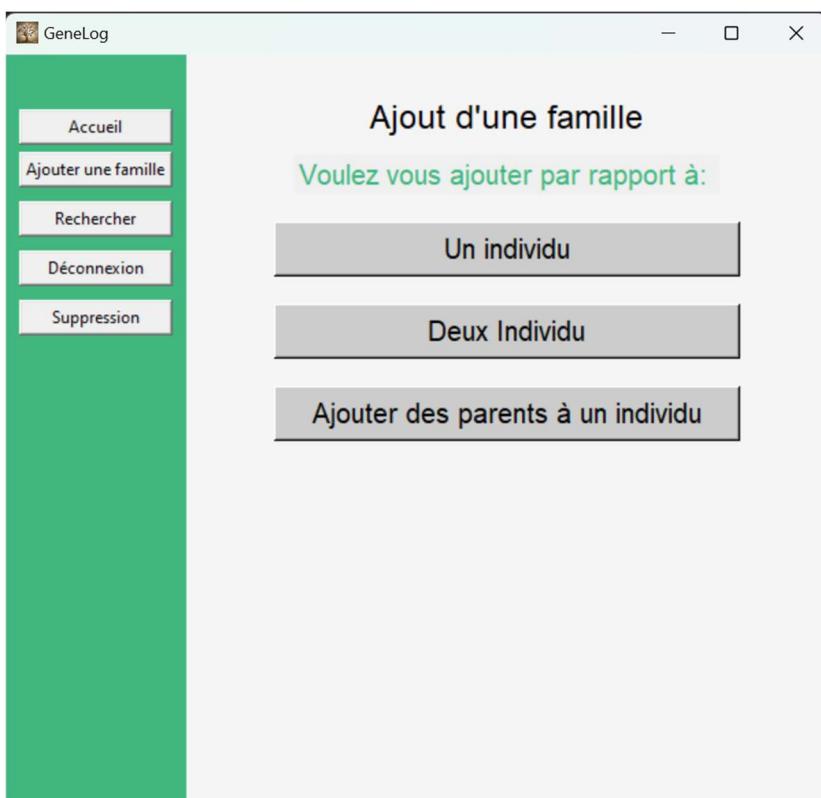
- 7) Une fois qu'il finit de saisir les informations de l'enfant et clique sur le bouton valider un messageshow apparaît pour lui demander s'il veut ajouter un autre enfant



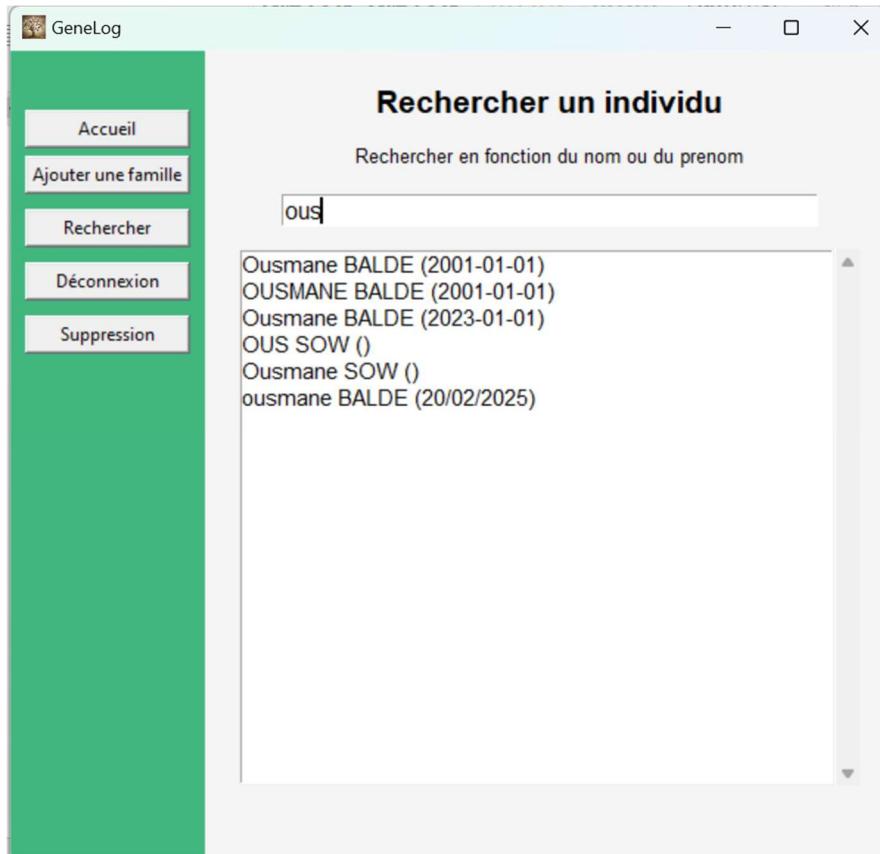
8) S'il clique sur non alors on lui propose d'enregistrer la famille dans la base.



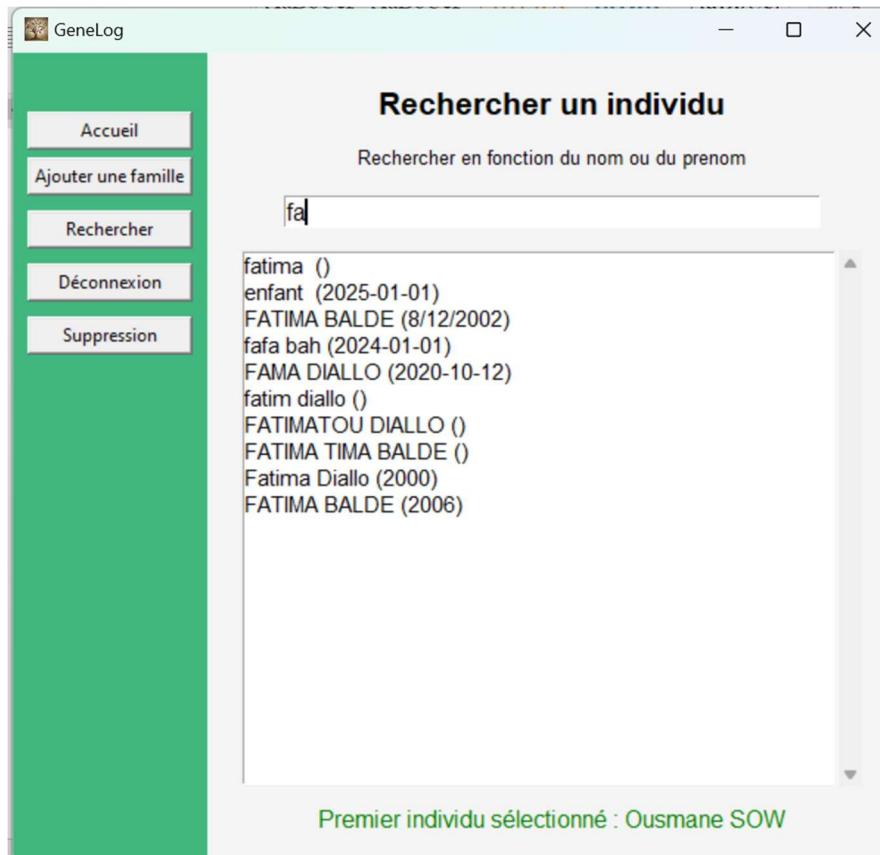
c - Ajout par rapport à 2 individu :



Rechercher les deux individus auxquels on veut ajouter une famille



Recherche de la 2eme personne



Une fois que l'âge des personnes sélectionnées atteint 15 ans, on passe à l'ajout de la famille. Les étapes qui suivent sont alors les mêmes que pour l'ajout d'une famille à partir d'un individu.

#### 2.4 Recherche / Modification d'une famille ou d'un individu :

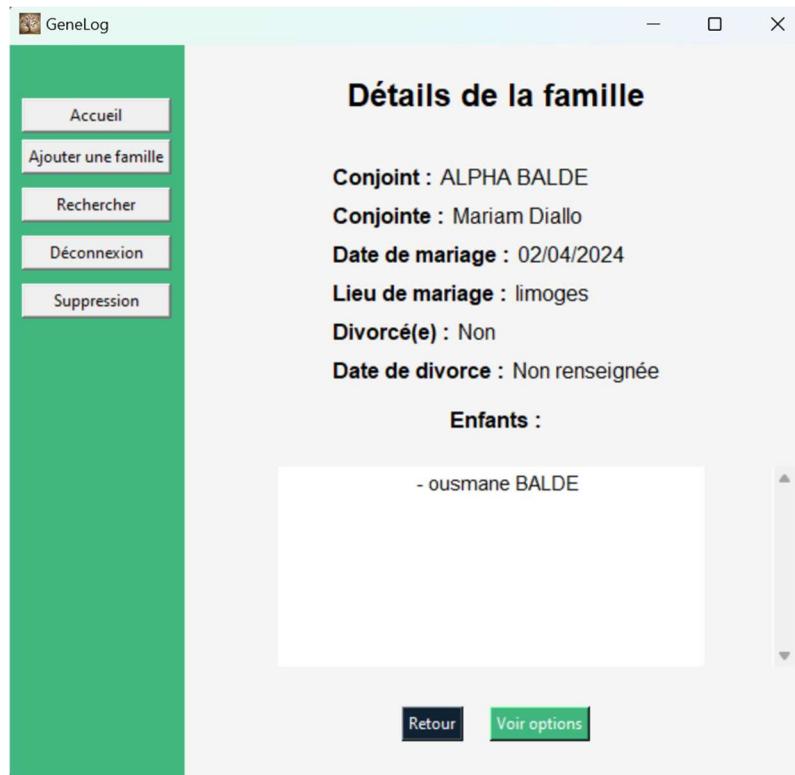
L'utilisateur peut rechercher un individu ou une famille, consulter et modifier leurs informations, ajouter des enfants, associer un individu à une famille, afficher ses ascendants, ses descendants, ainsi que les différentes familles auxquelles il appartient.



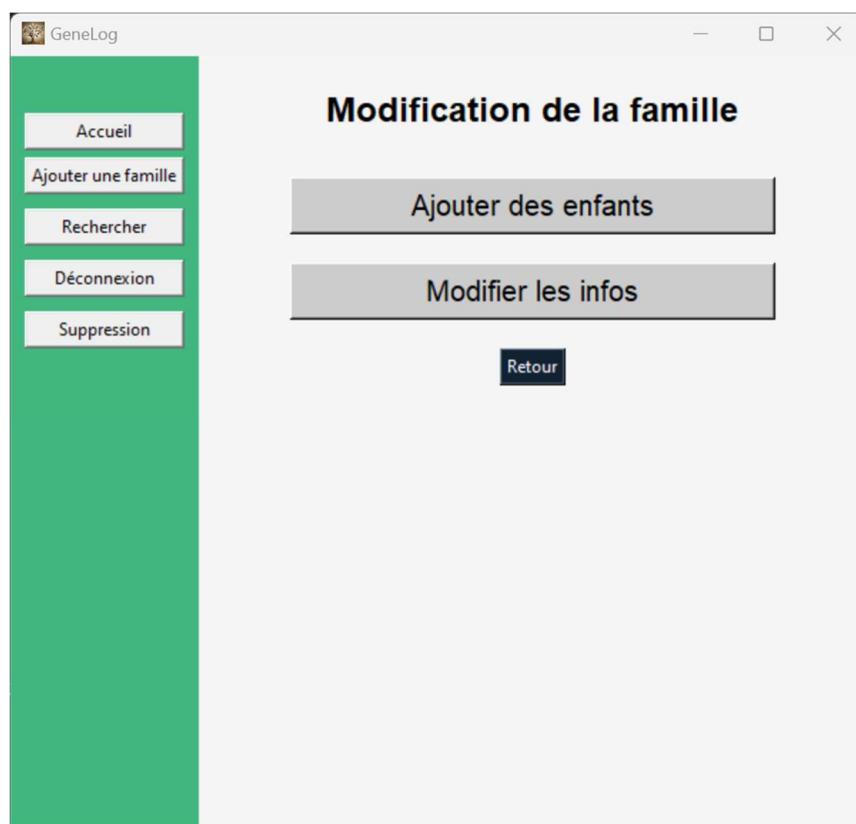
##### a - Rechercher la famille que l'on souhaite consulter ou modifier



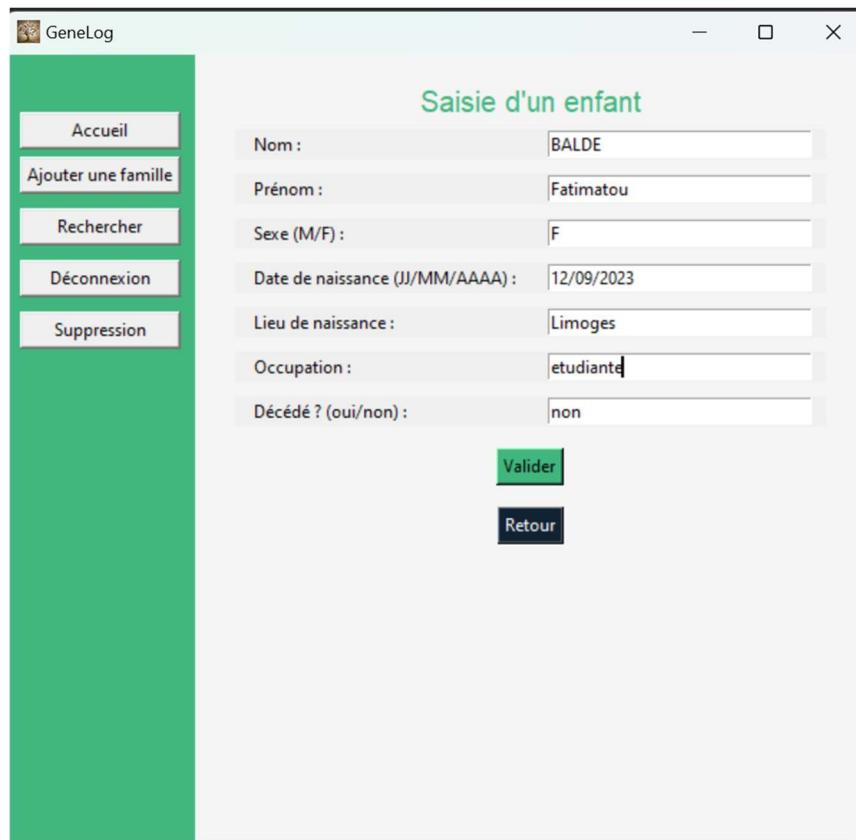
- 1) Affichage des informations de la famille(s'il choisit le bouton voir options alors il est redirigé vers les différentes options)



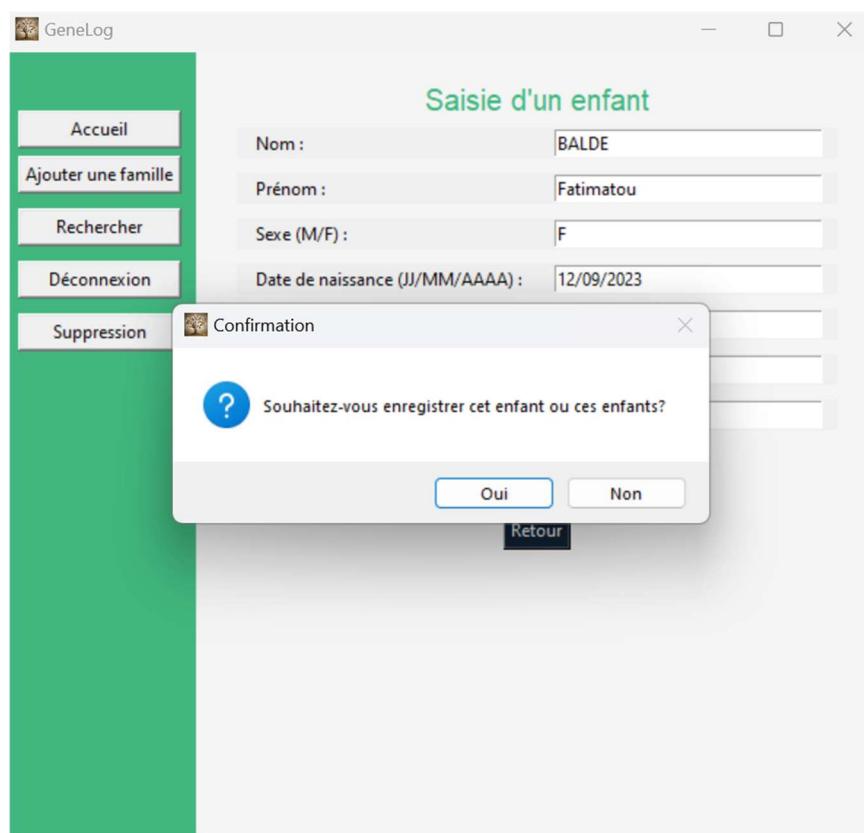
- 2) Choisir une option (ici on choisira la première pour un début)



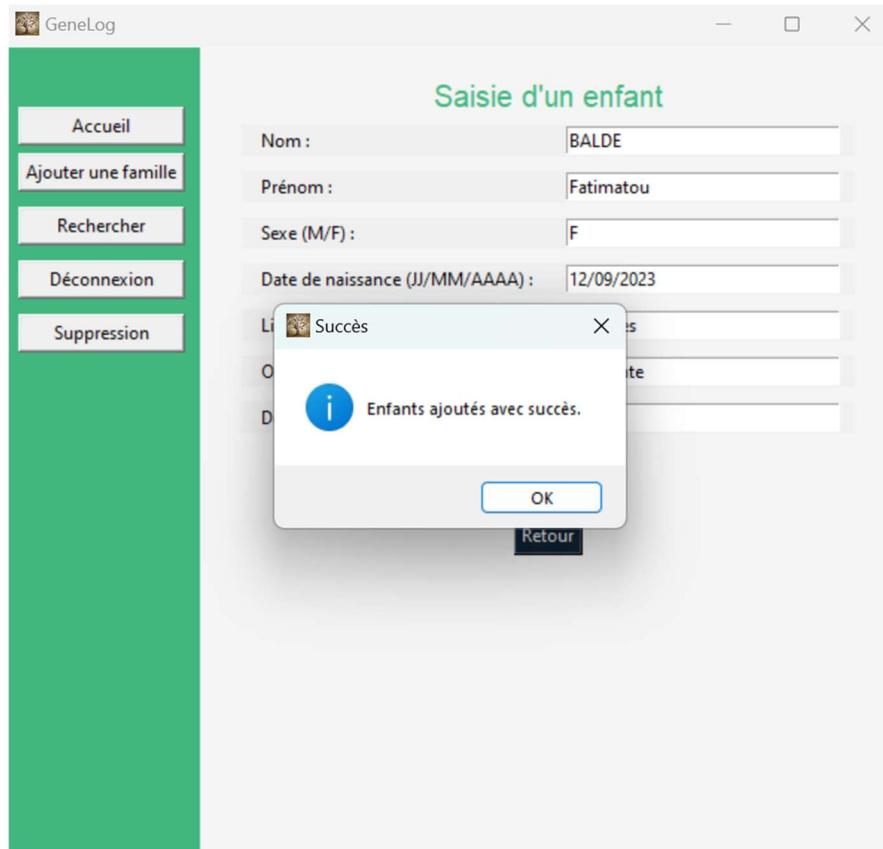
3) On remplit les informations de l'enfant



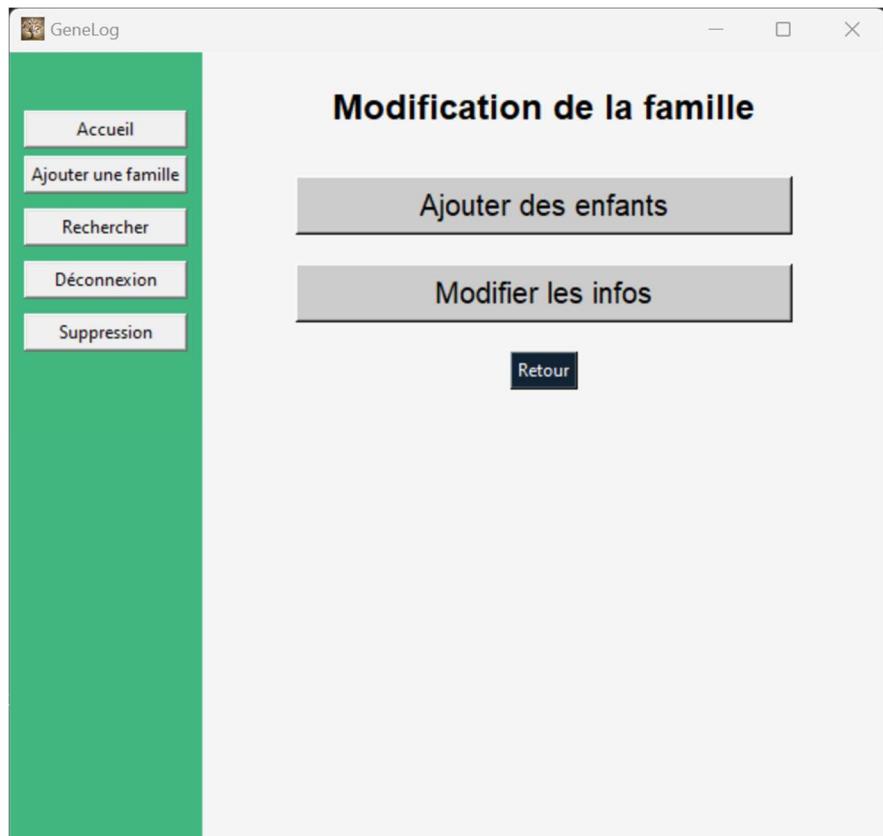
4) Messageshow pour demander à l'utilisateur s'il veut l'enregistrer



5) Si ‘oui’ enregistrement de l’enfant dans la famille



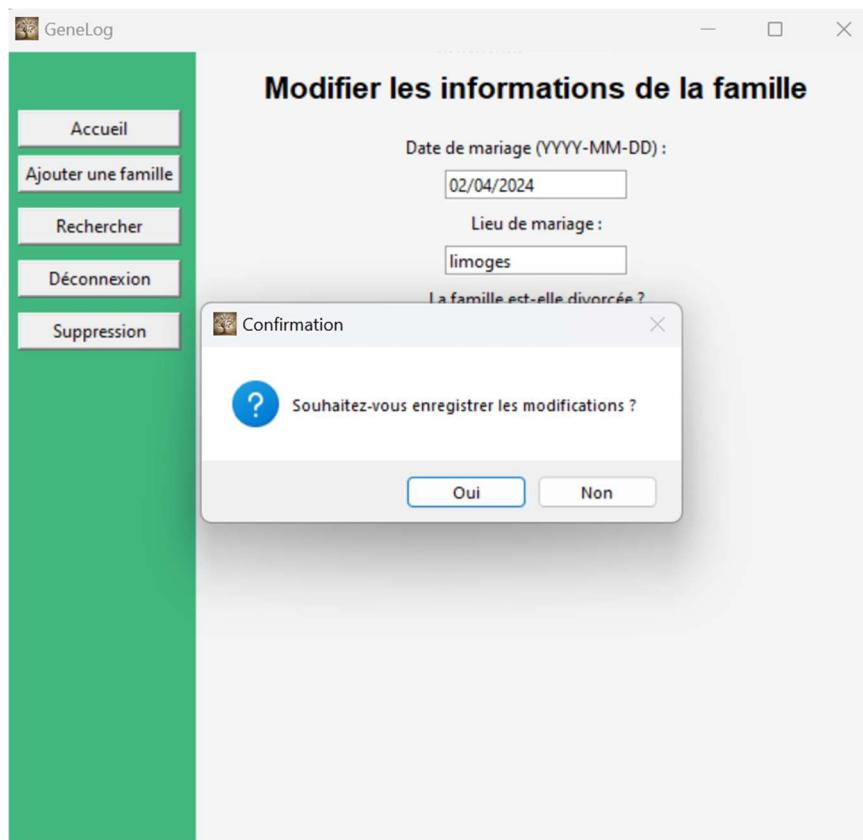
6) Si on choisit la seconde option (alors →)



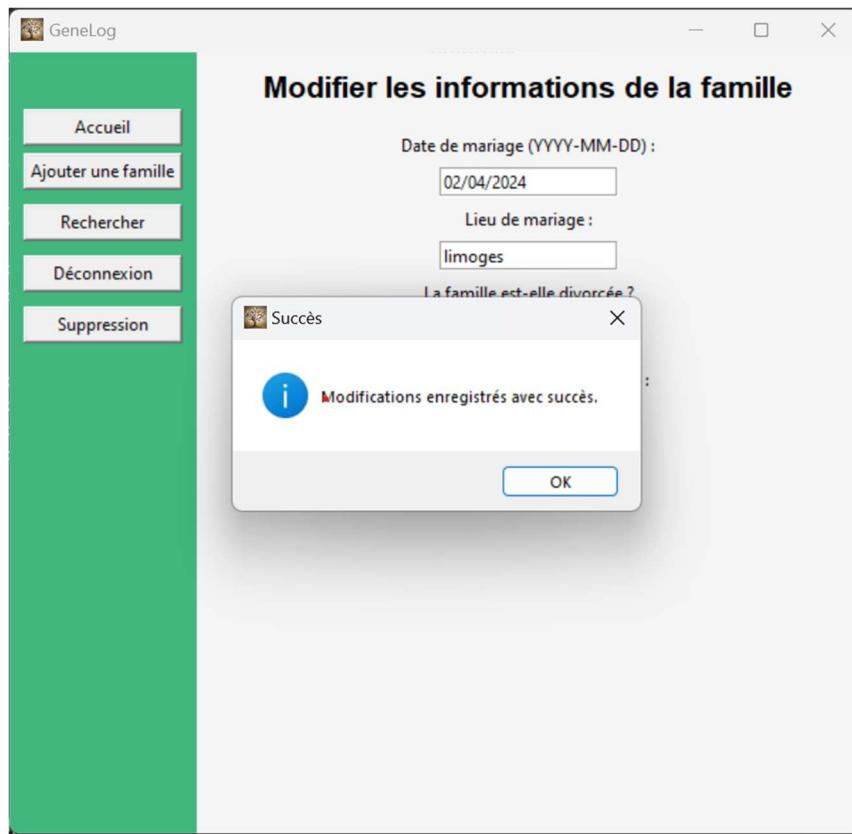
7) Je vais changer divorce en « oui » et mettre la date de divorce.



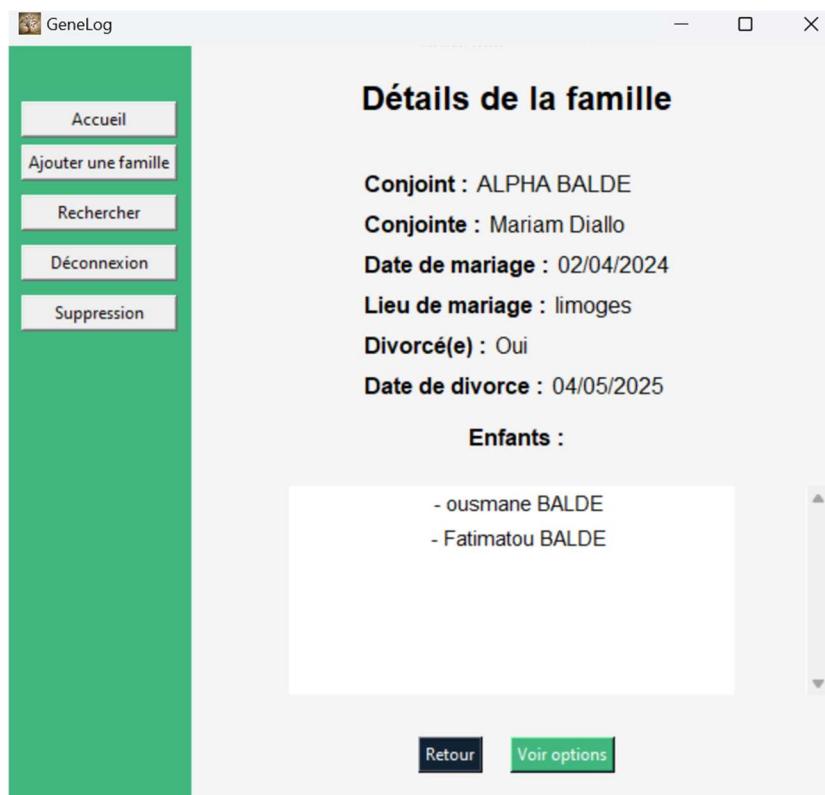
8) Affichage du messageshow



9) Si oui alors on enregistre les modifications dans la base

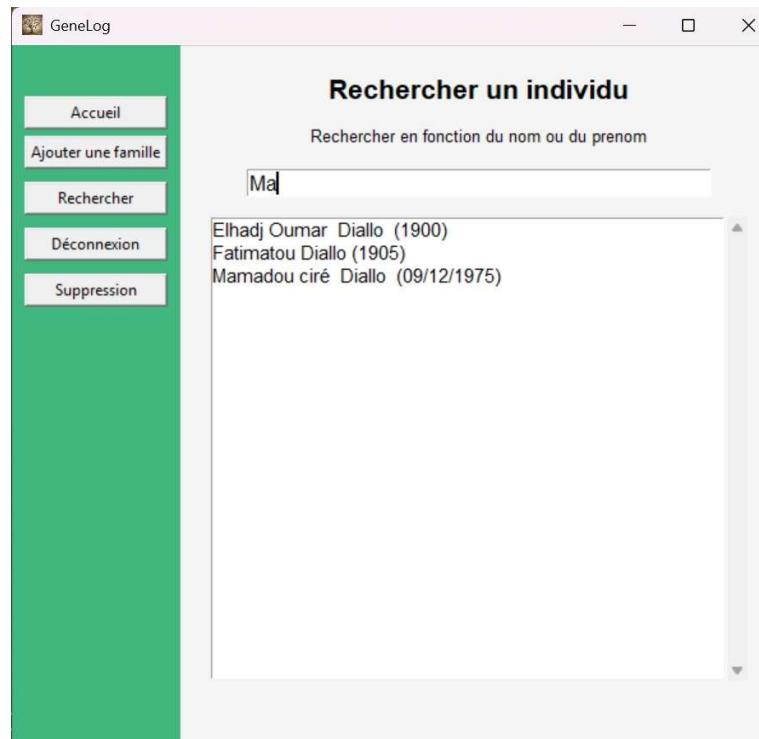


10) Si on affiche à nouveau les informations de la famille on verra qu'il y a eu une mise à jour. (Ajout d'un enfant et modification de l'état de divorce)

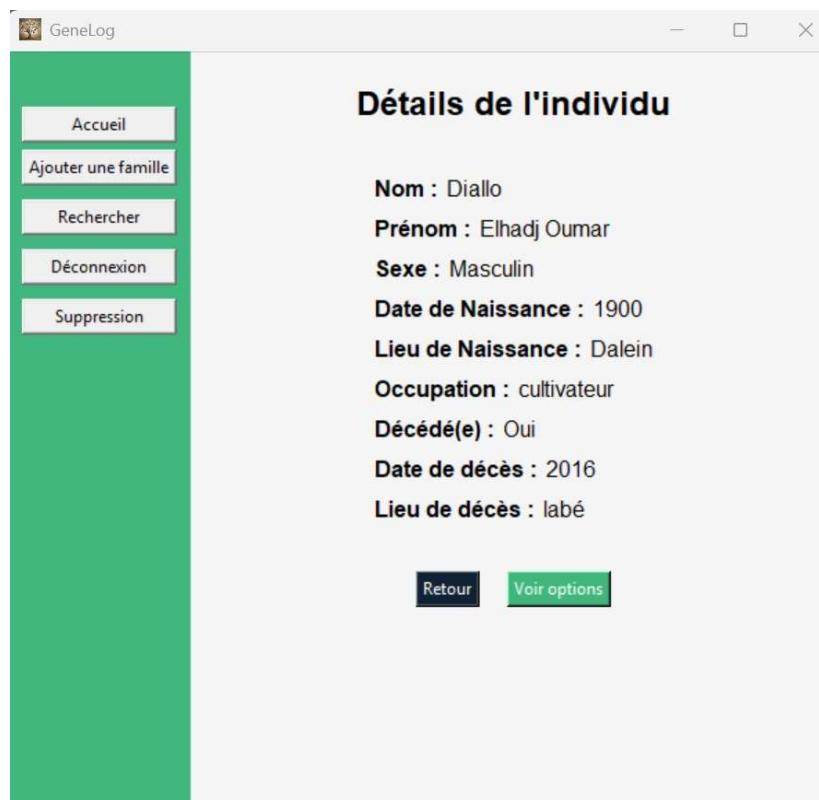


**b - Rechercher l'individu que l'on souhaite consulter, modifier ou auquel l'on veut ajouter une famille**

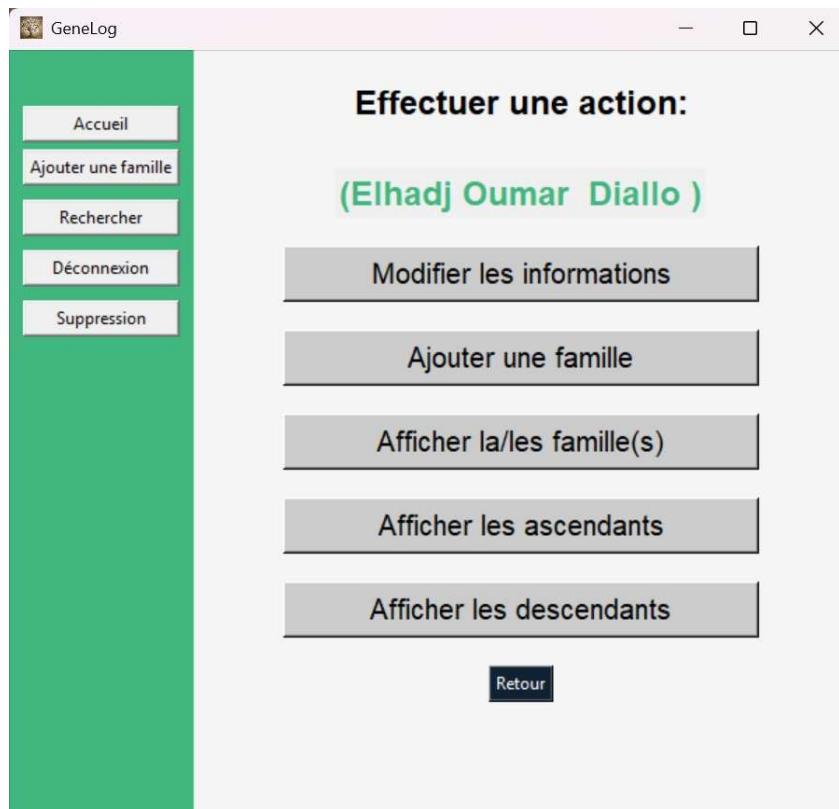
1) Recherche de l'individu



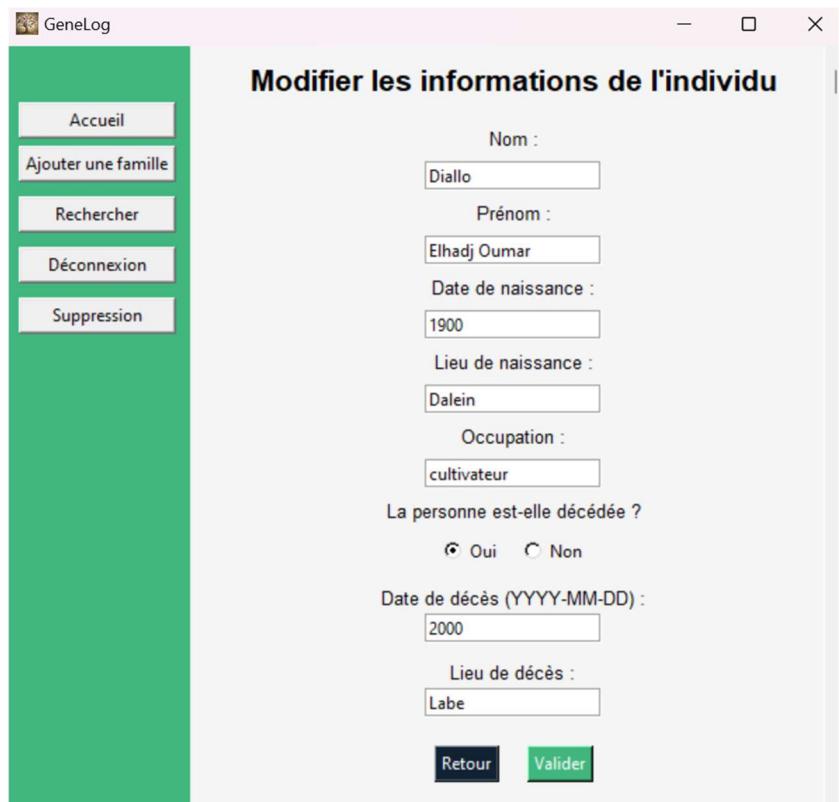
2) Affichage automatique de ces informations une fois qu'on lui trouve (sélectionné)



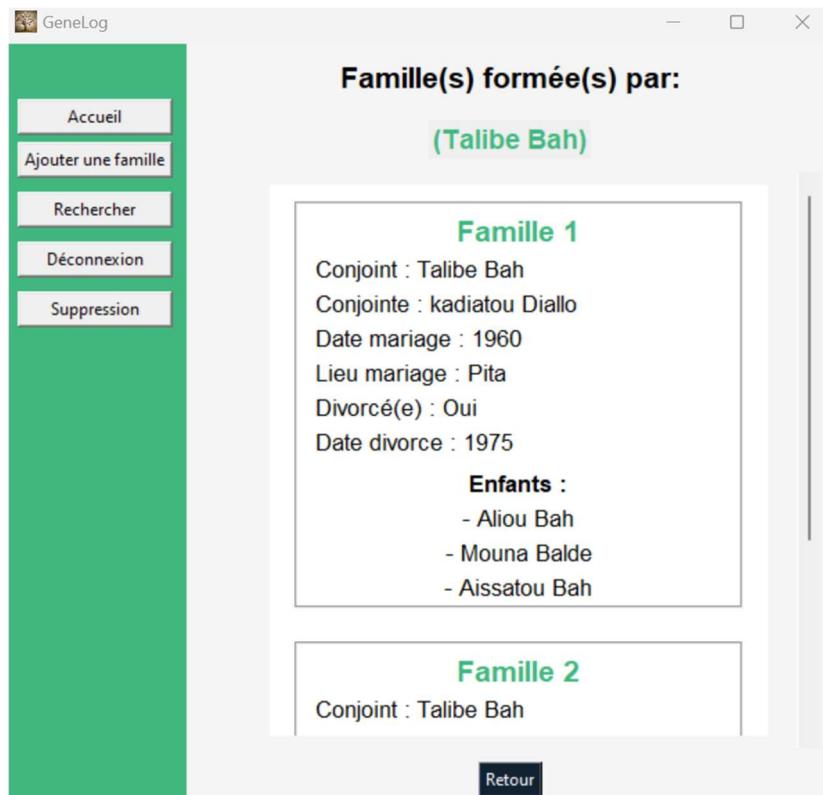
3) Si on clique sur « voir options »



4) Si on choisit l'option Modifier les informations :

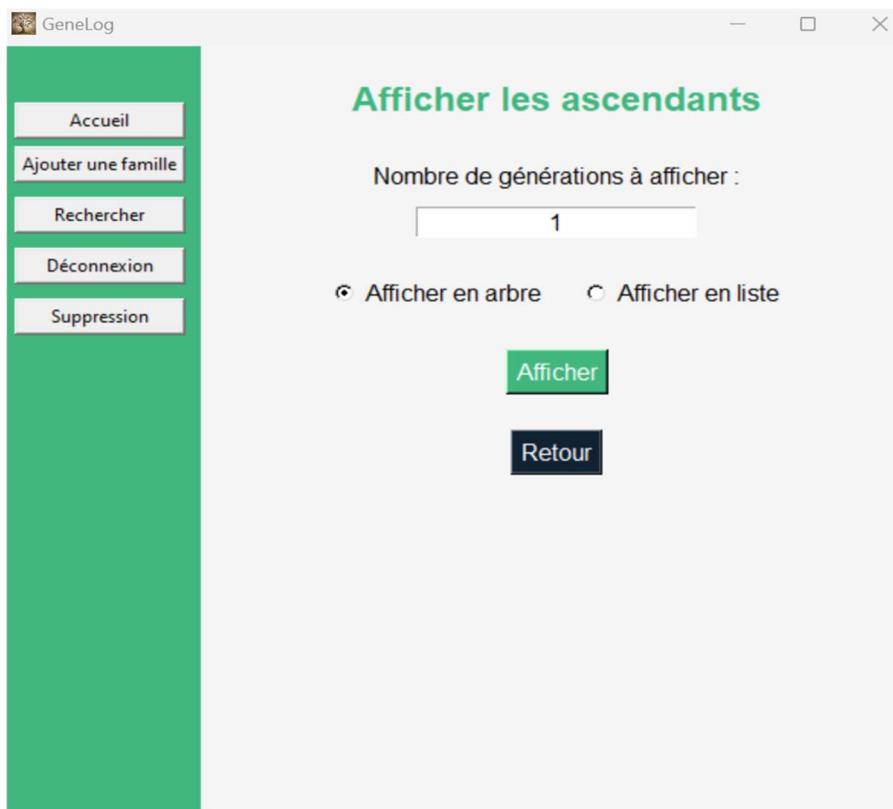


- 5) Si on choisit l'option Afficher la(les) famille(s) :

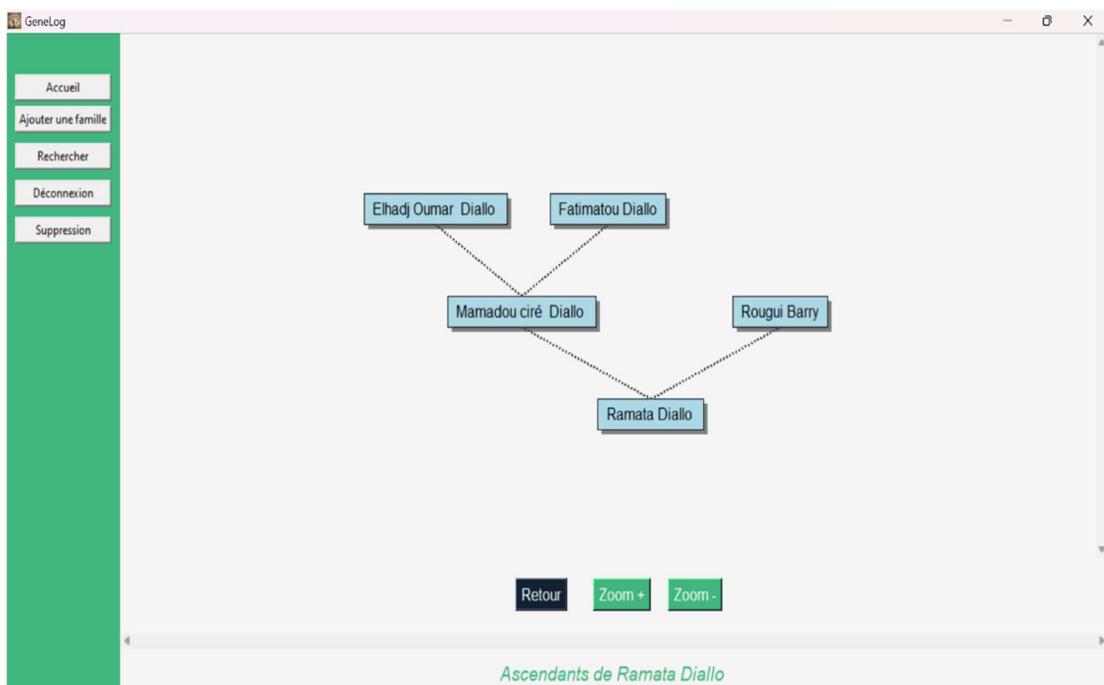


- 7) Si on choisit l'option Afficher les ascendants :

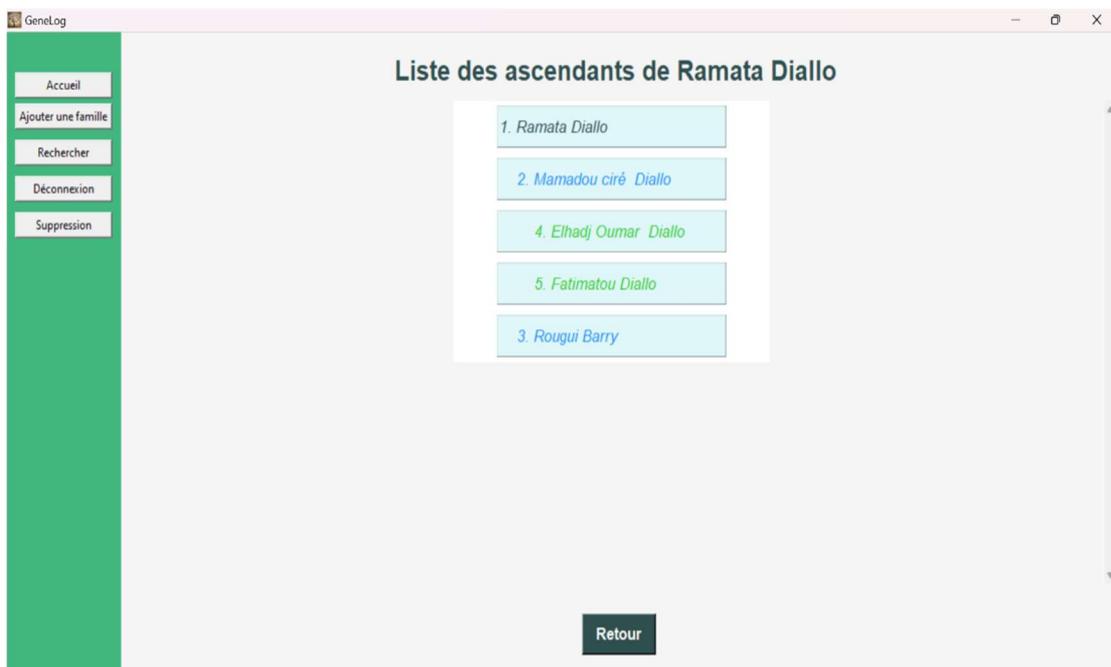
(Ici on nous demande de choisir si on veut l'afficher en arbre ou en liste, avec le nombre de générations à afficher)



a. Affichage en Arbre :

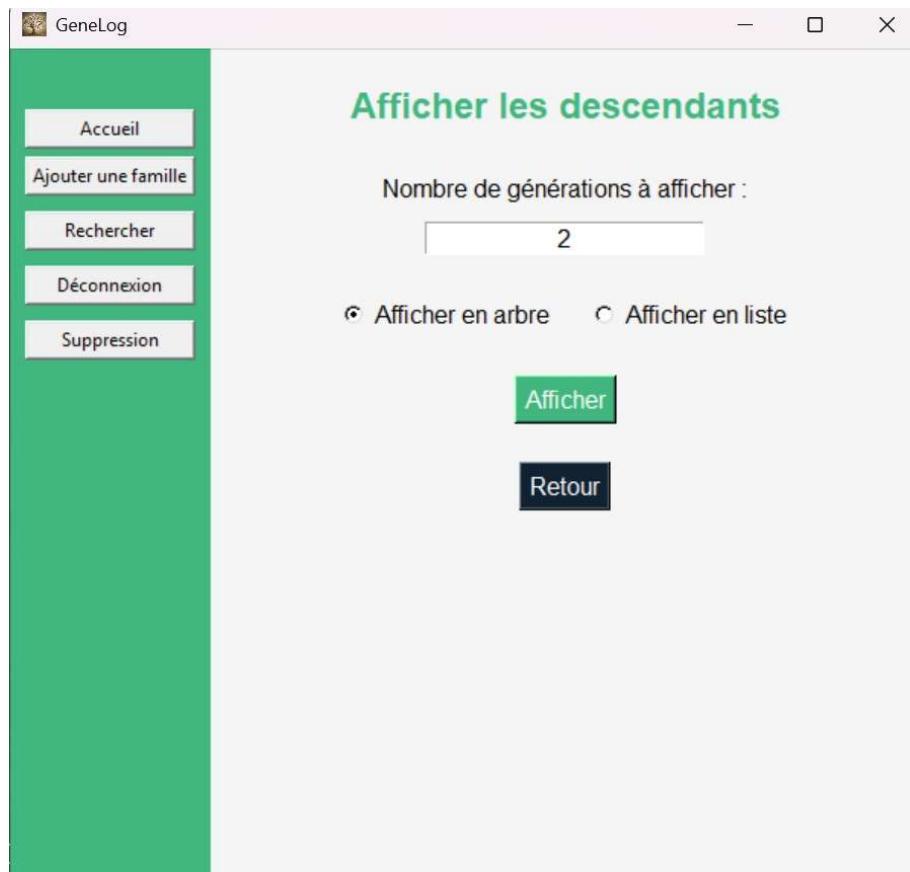


b. Affichage en Liste :

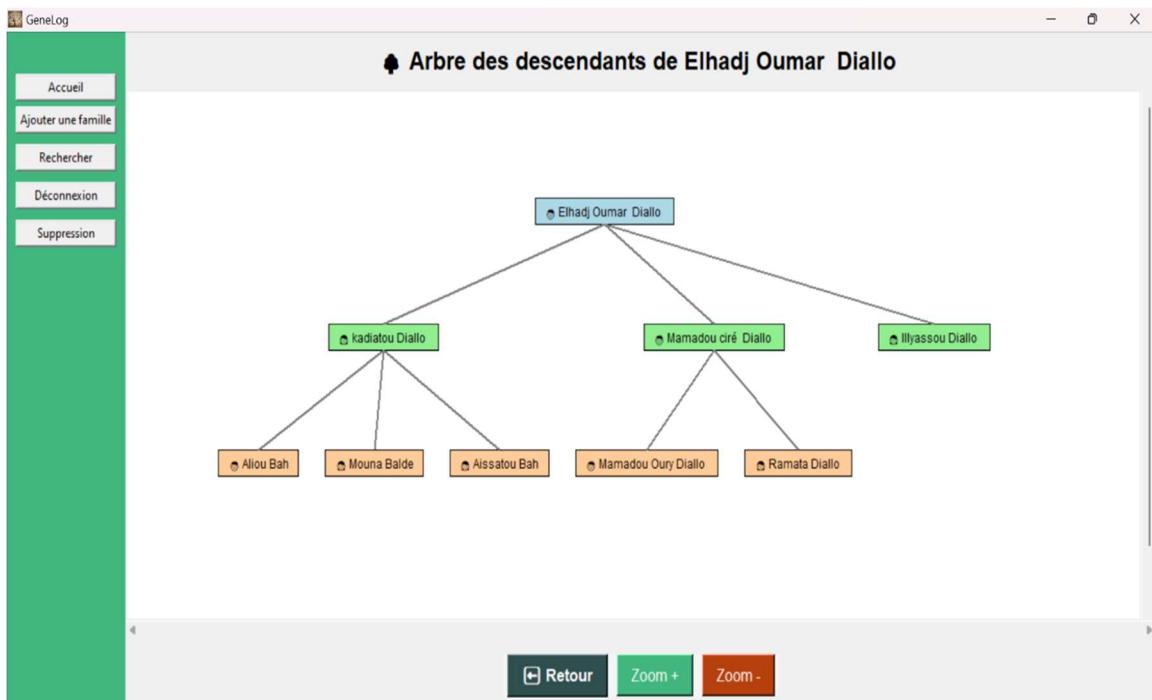


8) Si on choisit l'option Afficher les descendants :

(Ici on nous demande de choisir si on veut l'afficher en arbre ou en liste, avec le nombre de générations à afficher)



a. Affichage en arbre



b. Affichage en liste :



2.5 Déconnexion :

L'utilisateur peut se déconnecter à tout moment en cliquant sur le bouton *Déconnexion*, ce qui le redirige vers la page de connexion.



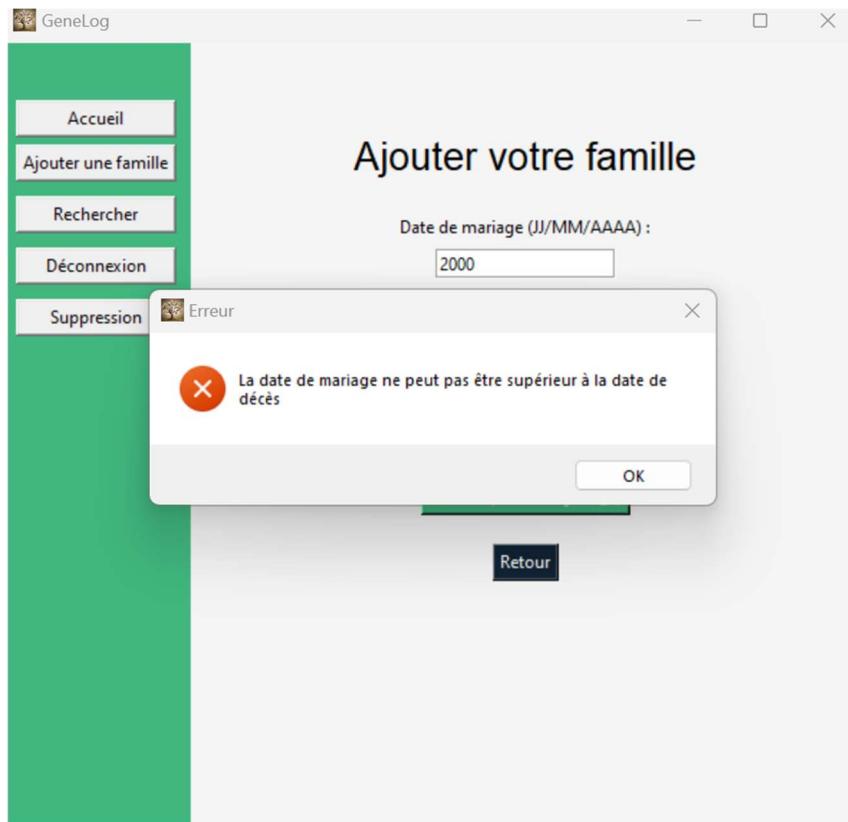
## 2.6 Suppression de la généalogie :

La suppression d'une généalogie nécessite la saisie du mot de passe, suivie d'une confirmation finale par pop-up, après confirmation, l'utilisateur est redirigé vers la page de connexion.

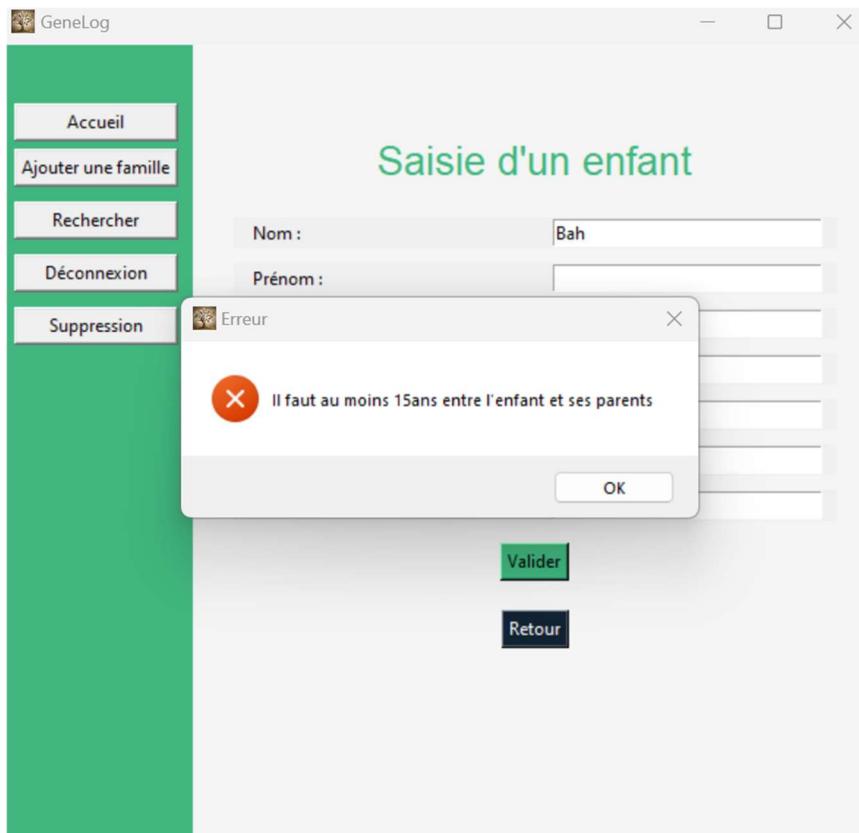


## 2.6 Contraintes:

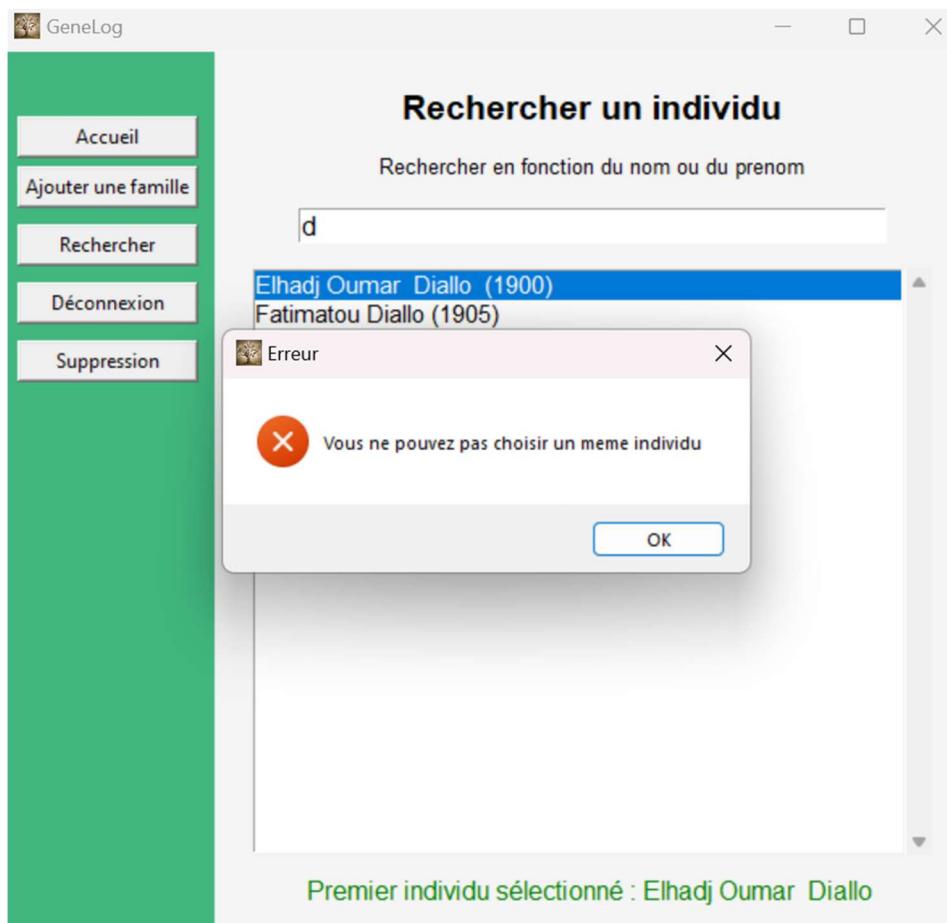
Un exemple de contrainte : Date de mariage et date de décès



Un autre exemple : Date de naissance de l'enfant et celle de ces parents



Un autre exemple : Il n'est pas possible de choisir le même individu lors de l'ajout d'une famille à deux individus



### 3. CONCLUSION

La réalisation de l'interface graphique a permis de concrétiser les fonctionnalités essentielles du logiciel de généalogie tout en assurant une utilisation simple et agréable. Grâce à l'utilisation de Tkinter, l'utilisateur peut créer, consulter et gérer les membres de sa famille de manière intuitive, sans compétences techniques particulières.

## CONCLUSION GENERALE

Ce projet de développement d'un logiciel de gestion de généalogie a permis de concevoir une application complète, fonctionnelle et accessible, permettant à tout utilisateur de créer, organiser et visualiser sa propre histoire familiale. De la phase d'analyse à celle de la réalisation, chaque étape a été guidée par une volonté de simplicité d'usage et de clarté dans la modélisation des données familiales.

L'étude fonctionnelle a permis de définir avec précision les besoins essentiels : la gestion des individus, la création de relations familiales (parents, enfants, conjoints), la visualisation de l'arbre généalogique ainsi que la sauvegarde des données. Cette base a servi à structurer l'architecture du projet autour de deux classes centrales, **Individu** et **Famille**, interconnectées et appuyées par une base de données **SQLite** assurant la persistance des informations.

Sur le plan technique, le choix de **Python** s'est révélé pertinent grâce à sa lisibilité, sa flexibilité et la richesse de ses bibliothèques. L'interface graphique développée avec **Tkinter** propose une expérience conviviale, avec des boutons intuitifs et des formulaires pour les principales opérations telles que l'ajout ou la modification des membres. Une attention particulière a été portée aux contraintes logiques liées à la cohérence des données, notamment la validité des dates ou la gestion des relations familiales.

Le projet a été centré sur la réalisation d'une interface graphique simple et intuitive, afin de permettre une utilisation fluide même par des personnes non initiées à l'informatique. Grâce à cette approche, l'application répond à un besoin réel : conserver et organiser les informations liées à la généalogie de manière structurée et visuelle.

Ce projet constitue ainsi une base solide pour une application de généalogie locale, avec un fort potentiel d'évolution. Parmi les pistes d'amélioration envisagées figurent l'**exportation de l'arbre au format PDF**, l'**ajout de photos** pour les individus, ou encore l'**enrichissement des données** par des notes biographiques ou des documents familiaux. Ce travail nous a permis de mettre en œuvre nos compétences en programmation orientée objet, en conception d'interfaces utilisateur, et en gestion de données, tout en répondant à une problématique concrète.

## ANNEXE

### 1. Méthodes pour lancer le programme

Méthode : Via le terminal Python

1. Ouvrir un terminal ou une invite de commande.
2. Se placer dans le dossier du projet : genealogie\_familiale
3. Exécuter le fichier principal : python genelog.py

### 2. Informations de connexion à la généalogie

Pour se connecter à la généalogie qu'on a utilisé pour les tests

- Nom de la généalogie : Diallo
- Mot de passe : Fatima@123