

Rapport Technique

Projet Arbre Généalogique Pro++

Bougdal, Ghilas Dridi, Iskander Guillarme, Arno Sadiqi, Mohamed-elghali
Aissaoui, Ahmed

24 mai 2025

Introduction

Le projet **Arbre Généalogique Pro++** a été réalisé dans le cadre du module d'ingénierie logicielle en première année du cycle ingénieur à CY Tech. L'objectif était de concevoir une application Java complète permettant de modéliser, gérer et visualiser des arbres généalogiques avec une interface utilisateur graphique intuitive, ainsi qu'une persistance des données via une base SQLite.

Organisation de l'équipe

Le travail a été réparti entre les membres de l'équipe de manière collaborative. L'intégration s'est faite progressivement, avec une forte communication via un groupe WhatsApp. Chaque membre du groupe a travaillé sur un environnement local, sans partager systématiquement ses avancements sur la branche commune. Cela a permis de garder une fluidité dans les modifications. Une fois les modifications essayées avec succès, puis vérifiées, les changements sont effectués sur la branche commune. Certains membres...

Objectifs du projet

- Permettre à un utilisateur de créer un arbre généalogique
- Ajouter, consulter, modifier ou supprimer des personnes
- Gérer les liens familiaux (parenté, fratrie, etc.)
- Fournir une interface graphique fonctionnelle via JavaFX
- Stocker les données dans une base SQLite embarquée
- Implémenter des filtres, une gestion des droits (admin/utilisateur), un historique de consultation

Architecture du projet

Le projet est construit sur une architecture en couches selon le modèle MVC :

- **Model** : contient les entités principales (Personne, Utilisateur, Lien, etc.)
- **DAO** : couche d'accès à la base de données SQLite
- **Controller** : coordonne les interactions entre l'utilisateur et le modèle
- **View** : interface graphique JavaFX
- **Service** : services transversaux (emails, gestion de liens)
- **Utils** : outils de validation

Fonctionnalités implémentées

- Authentification utilisateur et admin
- Ajout/suppression de membres à un arbre
- Interface graphique complète avec JavaFX
- Base de données SQLite avec persistance
- Filtres de consultation, historique, vues en lecture seule
- Gestion des demandes de liens et rôles (admin, utilisateur)

Technologies utilisées

- Java 17
- JavaFX
- SQLite via JDBC
- Maven pour la gestion de projet
- Git & GitHub pour la collaboration
- JUnit 5 pour les tests unitaires

Diagrammes UML

- **Diagramme de classes** : représentation des entités du modèle (**Personne**, **Utilisateur**, **Lien**, etc.), leurs attributs, méthodes et relations.
- **Diagramme de cas d'utilisation** : interactions principales entre l'utilisateur et l'application (authentification, création d'arbre, consultation, modification...).

Problèmes rencontrés et solutions

- **Intégration tardive** d'un membre → montée en compétence rapide via pair programming
- **Erreurs de mapping DAO** → refactorisation de la classe **Database** et utilisation de requêtes préparées
- **Dépendances JavaFX** non reconnues → configuration manuelle du module-path et des VM arguments

Limites fonctionnelles

- Absence de moteur de recherche avancé (requêtes complexes non supportées)
- Visualisation graphique de l'arbre perfectible (non interactive)
- Pas d'export PDF ou d'impression de l'arbre

Planning de réalisation

- Semaine 1-2 : prise en main, découpage, création des entités
- Semaine 3-4 : base de données, DAO, authentification
- Semaine 5-6 : interface JavaFX, fonctionnalités avancées
- Semaine 7 : tests, finalisation, correction de bugs

Lien GitHub

Le dépôt GitHub du projet est disponible ici :
<https://github.com/GhaSad/CYFamTree/tree/ghali>

Conclusion

Ce projet nous a permis d'aborder toutes les couches d'une application logicielle moderne : base de données, logique métier, interface graphique et travail en équipe. Malgré quelques difficultés, nous avons atteint une version stable, fonctionnelle, bien structurée et extensible de l'application.

Annexe : Diagrammes UML complets

Les diagrammes UML complets du projet sont présentés ci-dessous, tels qu'ils figurent dans le document produit avec l'outil de modélisation.

