

**E. S. G. A. E**  
**ECOLE SUPERIEURE DE GESTION**  
**ET D'ADMINISTRATION DES ENTREPRISES**

**Brazzaville – Congo**

Agrément définitif par Arrêté n°10403/MESRSIT/CAB du 25 Août 2023  
Accrédité par le Conseil Africain et Malgache pour l'Enseignement Supérieur (CAMES)  
B.P : 2339 Tel. (+242) 06 691 96 79 / 05 739 26 89  
E-mail : [esgae@yahoo.fr](mailto:esgae@yahoo.fr) ; [esgae@esgae.org](mailto:esgae@esgae.org)  
Site Web : <https://www.esgae.org>

**PROJET INFORMATIQUE**  
**POUR L'OBTENTION DU DIPLOME DE LICENCE**  
**PROFESSIONNELLE**  
**PARCOURS : GENIE LOGICIEL**

**THEME**

**ETUDE ET MISE EN ŒUVRE D'UNE PLATEFORME WEB ET MOBILE POUR**  
**LA CREATION COLLABORATIVE ET LE PARTAGE D'ARBRES**  
**GENEALOGIQUES A MAZALA-FIRM**

**Présenté et soutenu par :**  
M. Samuel Exaucé NANDI  
M. Dieu-veille Frédy ONIANGUE-DESO

**Sous la direction de :**  
M. Chistopher BANDZOUZI

**Année académique : 2023 – 2024**

# Dédicace

Je dédie ce modeste travail à :

- À mes parents ONIANGUE-DESO Rock Frédy et NGANBOMI Zita, pour leur amour inconditionnel, leur soutien constant et leurs sacrifices sans fin. Vos encouragements m'ont donné la force de poursuivre mes études et de réaliser ce mémoire.
- À mes professeurs, pour leur expertise, leur patience et leur inspiration. Leurs conseils éclairés ont enrichi mon parcours académique et ont façonné ma pensée critique.
- À mes amis, pour leur soutien indéfectible et leurs encouragements tout au long de cette aventure. Leurs encouragements m'ont permis de surmonter les défis avec confiance.
- À mon adorable conjointe DOUMA LEBONGUI Merveille Chrisnat.
- Enfin, à tous ceux qui ont croisé mon chemin et m'ont apporté leur aide, je vous suis profondément reconnaissant. Ce mémoire est le fruit de nos efforts collectifs et je vous en suis éternellement reconnaissant.

**Dieu-veille Frédy ONIANGUE-DESO**

Je dédie ce travail à mon cousin Inefable KOUMBA, de qui je tiens ce projet (à quelques personnalisation près).

**Samuel Exaucé NANDI**

# Remerciements

Nous tenons à remercier :

- Le professeur Roger Armand MAKANI, Directeur général de l'École Supérieure de Gestion et d'Administration des Entreprises, pour nous avoir accueillis dans cette grande école ;
- Monsieur Christopher BANDZOUZI notre directeur de mémoire pour sa longue patience et sa compréhension sans égale ;
- Tous les enseignants et le personnel administratif de l'ESGAE ;
- Nos parents et nos familles pour leur soutien ;
- Le personnel de MAZALA-FIRM.

# Table des figures

1.1	Organigramme de Mazala-Firm . . . . .	4
1.1	Illustration du processus de développement en Y . . . . .	15
2.1	Diagramme de contexte statique . . . . .	18
2.2	Diagramme de cas d'utilisation . . . . .	19
2.3	Diagramme de séquence pour gestion d'un arbre généalogique . . . . .	22
2.4	Diagramme d'activité pour la gestion d'un arbre généalogique . . . . .	23
2.5	Diagramme de classe du système . . . . .	25
2.6	Diagramme d'objet du système . . . . .	25
2.7	Diagramme de déploiement . . . . .	27
2.1	Logo Neovim . . . . .	34
2.2	Logo VSCodium . . . . .	34
2.3	Logo React . . . . .	34
2.4	Logo Next.js . . . . .	34
2.5	Logo Expo . . . . .	35
2.6	Logo Git . . . . .	35
2.7	Logo GitHub . . . . .	35
2.8	Logo Figma . . . . .	35
2.9	Logo L <sup>A</sup> T <sub>E</sub> X . . . . .	36
2.10	Logo PostgreSQL . . . . .	36
2.11	Logo Prisma . . . . .	36
2.12	Logo tRPC . . . . .	36
2.13	Logo tRPC . . . . .	37
2.14	Logo Docker . . . . .	37
2.15	Logo Vercel . . . . .	37
2.16	Écran de création d'un arbre généalogique sur l'application web . . . . .	38

# Liste des tableaux

1.1	Récapitulatif des méthodes . . . . .	13
2.1	Classification des cas d'utilisation en fonction de la priorité, du risque et de l'itération . .	19
1.1	Organisation du projet . . . . .	29
1.2	Intervenants . . . . .	29
1.3	Planification des tâches . . . . .	30
1.4	Estimation des charges . . . . .	32

# Sommaire

<b>Introduction</b>	1
<b>Première Partie : Présentation générale du magasin DC</b>	3
<b>Deuxième Partie : Analyse et conception</b>	11
<b>Troisième Partie : Evaluation et Réalisation</b>	29
<b>Conclusion</b>	39
<b>Table des matières</b>	41
<b>Bibliographie</b>	42
<b>Webographie</b>	43

# Abréviation et Sigles



# Introduction

Aujourd'hui, les technologies informatiques influencent considérablement notre vie quotidienne. La convergence des plateformes web et mobiles a remodelé la façon dont nous nous connectons, partageons et préservons nos informations. En tant que discipline, la généalogie est ancrée dans la préservation et la transmission de l'héritage familial. Avec l'avènement des technologies modernes, il devient impératif d'explorer de nouvelles perspectives pour faciliter la création et le partage d'arbres généalogiques. Dans le cadre de notre projet pour l'obtention de la licence professionnelle en filière informatique de gestion, option génie logiciel, au sein de l'École Supérieure de Gestion et d'Administration des Entreprises (E.S.G.A.E) , nous nous sommes engagés dans l'étude et la mise en œuvre d'une plateforme visant à simplifier et à enrichir l'expérience de reconstitution et de partage d'arbres généalogiques.

Nous avons choisi comme thème de projet « **Étude et mise en œuvre d'une plateforme web et mobile pour la création collaborative et le partage d'arbres généalogiques à Mazala-Firm** ». Cette initiative vise à répondre à un besoin croissant : la préservation et le partage de l'héritage familial à l'ère numérique. Notre projet s'articule autour de trois objectifs principaux : 1. Faciliter la création collaborative d'arbres généalogiques : nous souhaitons fournir une plateforme conviviale où les membres de la famille peuvent collaborer efficacement pour construire et partager leurs arbres généalogiques. 2. Simplifier le partage des informations généalogiques : nous souhaitons créer une interface intuitive sur les plateformes web et mobiles, permettant aux utilisateurs de partager facilement leurs arbres généalogiques avec leurs proches. 3. Promouvoir la découverte de son histoire ou de son héritage familial. Notre plateforme est destinée à tous ceux qui souhaitent explorer et partager leur lignée familiale, que ce soit des novices ou des professionnels de la généalogie. Pour mener à bien ce projet, nous avons opté pour une approche méthodologique agile : la méthode agile Two Tracks Unified Process (2TUP). Cette méthode, axée sur la collaboration et l'itération, nous permettra de développer la plateforme de manière incrémentale, en nous adaptant aux besoins changeants des utilisateurs.

Tout au long du processus, nous mettrons en œuvre les meilleures pratiques de développement logiciel pour garantir la qualité, la sécurité et la performance de notre solution. Notre objectif est de fournir une expérience enrichissante à toute personne désireuse de découvrir l'histoire de sa famille et de la partager avec ses proches.

Ce document décrit le travail accompli et se compose de trois parties distinctes. La première partie présentera l'environnement de travail en décrivant la structure qui nous a accueillis, ainsi que le projet qui nous a été confié, son contexte et ses enjeux. Dans la deuxième partie, nous approfondirons l'analyse, la conception et la mise en œuvre du projet, puis présenterons les outils et technologies utilisés. Enfin, la troisième partie traitera de la gestion du projet, de son évaluation et des perspectives futures.

## Première partie

# Présentation Générale

# Chapitre 1

## Structure d'Accueil et le Sujet

### 1.1 Structure d'Accueil

#### 1.1.1 Historique

Mazala-Firm est une agence de développement spécialisée dans la création de solutions numériques novatrices. Créée en 2021 par Inefable KOUMBA, un passionné des nouvelles technologies, Mazala-Firm s'est rapidement établie comme un acteur de premier plan dans ce domaine. L'histoire de Mazala-Firm débute avec Inefable KOUMBA, un ingénieur informatique visionnaire qui avait pour but de fonder une agence proposant des services complets en développement web. Fort de son expertise et de sa passion pour l'innovation, il rassemble une équipe de talents aux compétences variées, allant de l'analyse de données au design UX/UI en passant par le développement logiciel. Dès ses débuts, Mazala-Firm se distingue par son approche holistique du développement web. L'agence comprend rapidement que la clé du succès réside dans la capacité à transformer les données en informations exploitables. Ainsi, l'analyse des données devient l'un des piliers fondamentaux de l'activité de l'agence. Grâce à une expertise avancée en analyse de données, Mazala-Firm aide ses clients à tirer parti de leurs données pour prendre des décisions éclairées et stratégiques. Parallèlement, l'agence met un point d'honneur à offrir des expériences utilisateur exceptionnelles à travers ses services de conception UI/UX. En combinant esthétique et fonctionnalité, les designers de Mazala-Firm façonnent des interfaces intuitives et attractives, garantissant ainsi des interactions mémorables avec les produits numériques de ses clients. Le développement logiciel sur mesure est une autre spécialité de Mazala-Firm. Que ce soit pour la création d'applications mobiles, de plateformes web complexes ou de logiciels d'entreprise, l'équipe de développeurs de l'agence excelle dans l'art de concevoir des solutions sur mesure répondant parfaitement aux besoins spécifiques de chaque client.

#### 1.1.2 Missions

Les missions de Mazala-Firm sont multiples et s'articulent autour de plusieurs axes. L'entreprise est engagée dans la transformation de données en informations exploitables, dans la création d'expériences utilisateur simples et intuitives, dans le développement de solutions logicielles personnalisées, dans la fourniture de conseils stratégiques et d'audits informatiques pertinents, dans la gestion fiable des systèmes et réseaux informatiques, ainsi que dans la proposition de solutions de stockage sur le nuage sécurisées et flexibles.

Voici les missions de Mazala-Firm :

1. **Analyse des données** : Notre mission est de transformer les données brutes en informations exploitables. Nous utilisons une expertise avancée en analyse de données pour aider nos clients à comprendre, interpréter et exploiter efficacement les données pour prendre des décisions stratégiques et éclairées.
2. **UI & UX Design** : nous nous engageons à créer des expériences utilisateur mémorables en combinant l'esthétique et la fonctionnalité. Notre équipe de concepteurs d'interface utilisateur/expérience utilisateur travaille en étroite collaboration avec nos clients pour concevoir des interfaces intuitives, attrayantes et ergonomiques, garantissant ainsi une satisfaction utilisateur optimale.
3. **Développement logiciel** : notre objectif est de créer des solutions sur mesure qui répondent parfaitement aux besoins spécifiques de chaque client. Que ce soit pour le développement d'ap-

plications mobiles, de plateformes Web ou de logiciels d'entreprise, notre équipe de développeurs s'engage à fournir des solutions innovantes et de haute qualité.

4. **Conseil et audit informatique** : nous proposons des conseils stratégiques et des audits informatiques afin d'aider nos clients à optimiser leurs systèmes et leurs investissements technologiques. Notre mission est d'accompagner nos clients dans leur transformation digitale en leur fournissant des recommandations avisées et des solutions adaptées à leurs besoins spécifiques.
5. **Administration des systèmes et des réseaux informatiques** : nous nous engageons à assurer une gestion efficace et optimale des systèmes et des réseaux informatiques de nos clients. Notre équipe d'administrateurs système expérimentés travaille en coulisses pour garantir la fiabilité, la sécurité et la performance des infrastructures informatiques de nos clients.
6. **Stockage des données sur le nuage** : Notre mission est de fournir des solutions de stockage sur le nuage sécurisées et flexibles. En collaboration avec les principaux fournisseurs de services infonuagiques, nous offrons à nos clients un accès simple et sécurisé à leurs données, où qu'ils se trouvent.

### 1.1.3 Organigramme Général

L'organigramme de Mazala-Firm est bien structuré et se présente comme suit :

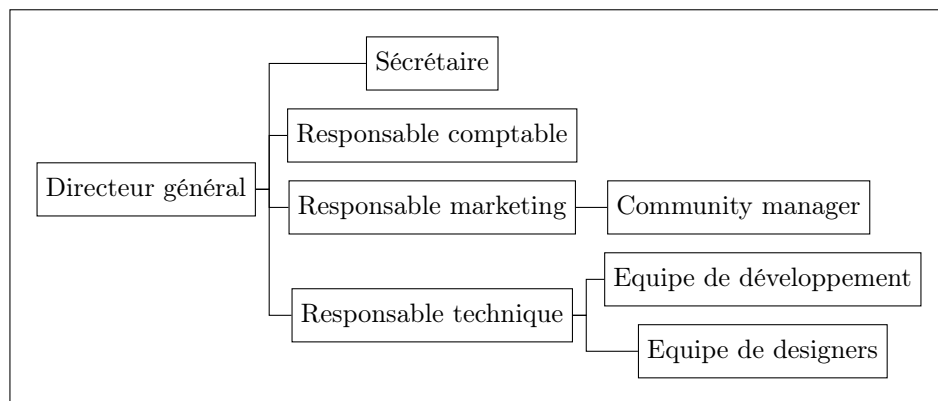


FIGURE 1.1 – Organigramme de Mazala-Firm

### 1.1.4 Attributions des structures

L'organigramme ci-dessus illustre le rôle crucial que jouent chaque poste et département au sein de Mazala-Firm. Voici une brève description de chaque structure et de ses principales responsabilités :

1. **Directeur général** : Chef d'entreprise, responsable de la gestion globale, des décisions stratégiques et de la supervision des départements et des employés.
2. **Secrétaire** : assiste les dirigeants et les cadres supérieurs dans leurs tâches administratives, notamment la gestion du courrier et des réunions.
3. **Responsable comptable** : gère les finances de l'entreprise, y compris la tenue des livres comptables et la préparation des rapports financiers.
4. **Responsable marketing** : planifie, met en œuvre et gère les stratégies marketing, y compris la publicité et les relations publiques.
5. **Responsable technique** : supervise les aspects techniques de l'entreprise, notamment l'infrastructure informatique et les processus de développement.
6. **Community manager** : gère et anime les communautés en ligne, interagit avec les clients et crée du contenu pour engager la communauté.
7. **Équipe de développement** : conçoit, développe et maintient les logiciels et les applications de l'entreprise et des produits.
8. **Équipe designer** : responsable de la conception visuelle des produits, incluant la conception graphique et l'interface utilisateur.

### 1.1.5 Situation informatique

Mazala-Firm dispose d'un département informatique dédié, supervisé par le Responsable technique. Ce département est structuré en deux équipes principales :

- **Équipe de développement** : conçoit et développe des solutions logicielles personnalisées en collaboration avec les clients.
- **Équipe de design** : crée des interfaces utilisateur attrayantes et intuitives pour les produits numériques de l'entreprise.

#### 1.1.5.1 Personnel informatique

L'équipe du service informatique comprend une équipe talentueuse composée de :

- un chef de Service technique
- des développeurs logiciels
- des designers UX/UI
- des ingénieurs et techniciens informatiques spécialisés dans
- le développement et la maintenance des solutions logicielles et des interfaces utilisateur.

Cette structure organisationnelle garantit que Mazala-Firm dispose des ressources et des compétences nécessaires pour répondre aux besoins technologiques de ses clients et fournir des solutions informatiques de haute qualité et innovantes.

#### 1.1.5.2 Matériels informatiques

L'efficacité de Mazala-Firm dépend étroitement de notre matériel informatique. En tant qu'entreprise spécialisée dans le développement web et le design, nous comprenons l'importance d'un équipement de qualité pour soutenir nos équipes dans la réalisation de projets innovants. Ci-dessous, nous présentons les principaux composants de notre infrastructure informatique, démontrant ainsi notre engagement envers des solutions performantes et une satisfaction client optimale.

Ainsi, Mazala-Firm dispose de l'équipement informatique suivant :

- Des ordinateurs portables performants dotés de processeurs Intel Core i7 ou i9, ou AMD Ryzen 7 ou 9, avec au moins 12 Go de RAM et des disques SSD de 512 Go à 1 To. Ils conviennent parfaitement aux développeurs et aux concepteurs, que ce soit au bureau ou sur le terrain.
- Stations de travail puissantes pour les tâches de développement et de conception les plus exigeantes, dotées de processeurs Intel Xeon ou AMD Threadripper avec 32 à 64 Go de RAM et des cartes graphiques dédiées comme les NVIDIA Quadro ou AMD Radeon Pro pour accélérer les rendus et la visualisation.
- Écrans de haute résolution de 27 à 32 pouces, avec une résolution minimale de 2560x1440 (WQHD) ou 3840x2160 (4K UHD) et une couverture de l'espace colorimétrique sRGB ou Adobe RGB à 99
- Tablettes graphiques professionnelles comme les Wacom Cintiq ou Huion Kamvas Pro, avec des tailles d'écran de 16 à 24 pouces et une résolution de 1920x1080 (Full HD) ou supérieure, permettant de dessiner et de retoucher des images avec précision.
- Serveurs de hautes performances équipés de processeurs Intel Xeon ou AMD EPYC, avec au moins 64 Go de RAM et des options de stockage RAID avec des disques SSD ou NVMe, garantissant une disponibilité et une sécurité maximales pour les services en ligne de Mazala-Firm.
- Équipements réseau tels que des routeurs compatibles Wi-Fi 6, des commutateurs Ethernet gigabit ou 10 gigabit, et des points d'accès Wi-Fi couvrant l'ensemble du bâtiment pour assurer une connexion réseau fiable et sécurisée.
- Accessoires comme Claviers ergonomiques comme les Microsoft Sculpt ou Logitech Ergo, souris de précision comme les Logitech MX Master, casques audio à réduction de bruit comme les Bose QC35 ou Sony WH-1000XM4, et autres accessoires informatiques pour permettre à chaque membre de l'équipe de travailler confortablement et efficacement.

#### 1.1.5.3 Logiciels informatiques

Mazala-Firm utilise une gamme variée de logiciels pour soutenir ses opérations. Ceux-ci incluent :

- des outils de développement tels que Visual Studio code, Vim, IntelliJ IDEA et PyCharm pour la programmation et la création de solutions logicielles sur mesure ;
- des logiciels de conception graphique tels que Figma, Adobe Creative Suite (Photoshop, Illustrator, InDesign) pour la création d'interfaces utilisateurs attrayantes et de graphiques de haute qualité ;

- des outils de productivité tels que Microsoft Office (Word, Excel, PowerPoint) pour la gestion des documents et la communication professionnelle ;
- des systèmes d'exploitation (Windows, Linux, MacOS et FreeBSD), qui prennent en charge les différentes plates-formes et les besoins spécifiques des développeurs, des concepteurs et des administrateurs système ;
- des outils de sécurité tels qu'Avast Antivirus Premium ainsi que des VPN comme ProtonVPN pour protéger les systèmes et les données contre les menaces en ligne.

## 1.2 Sujet

### 1.2.1 Contexte du Sujet

À l'heure actuelle, l'étude de la généalogie est en pleine mutation grâce aux avancées technologiques. Dans ce contexte évolutif, Mazala-Firm a identifié un besoin crucial : celui d'une plateforme dédiée à la création collaborative et au partage d'arbres généalogiques. Cette initiative répond à une demande croissante de préservation et de partage de l'héritage familial à l'ère numérique. Elle propose une solution innovante pour les utilisateurs qui souhaitent explorer et partager leur lignée familiale.

La généalogie occupe une place particulière dans la société, offrant aux individus une occasion unique de découvrir leurs racines et de se connecter à leur passé familial. En facilitant l'accès à ces connaissances et en encourageant la collaboration entre les membres de la famille, notre plateforme vise à renforcer le lien intergénérationnel et à préserver l'histoire familiale pour les générations futures.

Les méthodes traditionnelles de création et de partage d'arbres généalogiques sont souvent limitées par des contraintes physiques, comme la disponibilité de documents papier ou la difficulté de les partager avec des membres de la famille éloignés. En intégrant les dernières avancées technologiques, notre plateforme offre une solution moderne et conviviale. Elle permet aux utilisateurs de créer, de visualiser et de partager facilement leur arbre généalogique avec leur famille et leurs proches, peu importe où ils se trouvent dans le monde.

Les objectifs spécifiques de notre projet incluent la facilitation de la collaboration entre les membres de la famille, la convivialité de l'interface utilisateur et l'intégration de fonctionnalités de partage avancées. En mettant l'accent sur l'accessibilité et la simplicité d'utilisation, nous voulons démocratiser l'accès à la généalogie et encourager un engagement actif dans la préservation de l'histoire familiale.

À long terme, notre plateforme pourrait jouer un rôle crucial dans la préservation de l'histoire familiale pour les générations futures. Nous aspirons à développer des fonctionnalités supplémentaires, telles que des analyses généalogiques avancées et des outils de santé familiale, afin de créer une ressource complète pour les utilisateurs soucieux de comprendre leur passé et de planifier leur avenir en toute connaissance de cause.

### 1.2.2 Problématique du sujet

L'évolution technologique et la nécessité de préserver et de partager l'héritage familial à l'ère numérique soulèvent plusieurs questions. Elles incluent :

- Comment favoriser la collaboration pour la création collective d'arbres généalogiques parmi les membres de la famille, peu importe leur emplacement géographique ?
- Comment rendre plus simple le partage d'informations généalogiques tout en assurant la protection de la vie privée et de la sécurité des données ?
- Comment encourager la découverte de l'histoire familiale et inciter activement les gens à prendre soin de leur patrimoine familial ?
- Quelles sont les meilleures méthodes pour concevoir une plateforme web et mobile efficace pour la création collaborative et le partage d'arbres généalogiques ?
- Quels sont les technologies et les outils les plus appropriés pour développer une plateforme de généalogie conviviale et sécuritaire ?
- Quel choix architectural et quelle approche méthodologique adopter pour garantir la qualité, la performance et la sécurité de la plateforme ?

### 1.2.3 Intérêts du sujet

La création collaborative et le partage d'arbres généalogiques sur une plateforme web et mobile présentent un intérêt majeur à plusieurs niveaux.

D'abord, cette initiative répond à un besoin profond de notre société actuelle : celui de préserver et de transmettre l'histoire familiale aux générations futures. Elle permet aux individus d'explorer leurs racines et de découvrir leur lignée familiale, ce qui contribue à renforcer le lien intergénérationnel et à promouvoir un sentiment d'identité et d'appartenance au sein de la famille.

Ensuite, la dimension collaborative de cette plateforme offre aux utilisateurs une occasion unique de partager leurs connaissances et leurs découvertes avec leur famille élargie, favorisant ainsi les échanges interculturels et intergénérationnels. Cette collaboration peut également mener à la découverte de nouvelles branches familiales et à la réconciliation de membres de la famille séparés par le temps ou la

distance.

Sur le plan technologique, ce projet représente une avancée significative dans le domaine de la généalogie numérique. Il intègre les dernières innovations en matière de conception d'interfaces utilisateurs conviviales et d'analyse de données. En proposant une expérience utilisateur intuitive et personnalisée, il vise à démocratiser l'accès à la généalogie et à encourager un engagement actif dans la préservation de l'histoire familiale.

Enfin, sur le plan social, cette plateforme peut contribuer à sensibiliser les individus aux enjeux liés à la santé familiale. Elle permet par exemple d'identifier les prédispositions génétiques à certaines maladies et d'adopter des comportements préventifs, mais aussi la consanguinité. En favorisant la prise de conscience des liens familiaux et des risques pour la santé, cette plateforme joue un rôle crucial dans la promotion du bien-être familial et de la solidarité intergénérationnelle.

Dans l'ensemble, la création d'une plateforme web et mobile dédiée à la généalogie collaborative et au partage d'arbres généalogiques constitue une occasion unique de combiner les avancées technologiques aux besoins fondamentaux de préservation de l'histoire familiale et de renforcement du lien familial. En mettant l'accent sur l'accessibilité, la convivialité et la sécurité des données, ce projet vise à devenir une référence incontournable dans le domaine de la généalogie numérique. Il contribue ainsi à enrichir le patrimoine familial et culturel de notre société.

### 1.3 Concepts liés au sujet

Le sujet que nous abordons englobe une série de concepts clés qui sont au cœur de notre projet de création collaborative et de partage d'arbres généalogiques. Ces concepts incluent :

- **Généalogie** : il s'agit de l'étude des relations de parenté et de la succession des générations au sein d'une famille. La généalogie est le fondement même de notre projet, offrant aux individus la possibilité de découvrir et de préserver leur héritage familial.
- **Arbre généalogique** : un arbre généalogique est une représentation graphique des liens de parenté entre les membres d'une famille. Il permet de visualiser les relations familiales sur plusieurs générations et de retracer l'histoire familiale de manière structurée et organisée.
- **Collaboration en ligne** : Ce terme désigne le processus de travail d'équipe visant à atteindre un objectif commun par le biais de plateformes numériques. Dans notre cas, la collaboration en ligne revêt une importance particulière, car elle permet aux membres de la famille de contribuer à la construction et à l'enrichissement des arbres généalogiques, peu importe leur emplacement géographique.
- **Plateformes web et mobiles** : nous envisageons de développer une plateforme accessible à la fois sur le web et sur les appareils mobiles, offrant ainsi aux utilisateurs la flexibilité nécessaire pour accéder à leurs données familiales où qu'ils se trouvent. Cette approche garantit une expérience utilisateur optimale et facilite le partage d'informations entre les membres de la famille.
- **Sécurité des données** : La protection des informations personnelles des utilisateurs et des données sensibles relatives à l'histoire familiale est une priorité pour notre projet. Nous mettrons en place des mesures de sécurité robustes pour garantir la confidentialité et l'intégrité des données, tout en offrant aux utilisateurs un contrôle total sur leurs informations.
- **Partage de données** : nous accordons une grande importance au partage efficace et sécuritaire des données entre les membres de la famille. Notre plateforme facilitera ce processus en permettant aux utilisateurs d'échanger et de transmettre des informations de manière transparente, tout en respectant les préférences de confidentialité de chacun.
- **Analyse des données** : en plus de permettre la création et le partage d'arbres généalogiques, notre plateforme offrira des outils d'analyse avancés pour que les utilisateurs puissent explorer et comprendre leur histoire familiale de manière approfondie. Des fonctionnalités telles que des statistiques démographiques, des tendances généalogiques et des analyses de santé familiale seront intégrées pour fournir des informations précieuses aux utilisateurs et les aider à mieux comprendre leur lignée et leur héritage.
- **Accessibilité** : enfin, nous nous engageons à rendre notre plateforme accessible à tous, y compris aux personnes ayant des besoins spécifiques en matière d'accessibilité. Nous veillerons à ce que l'interface utilisateur soit intuitive et facile à utiliser pour que chacun puisse profiter pleinement de l'expérience généalogique, quel que soit son niveau de compétences technologiques.

Ces concepts joueront un rôle essentiel dans la conception et la mise en œuvre de notre plateforme. Ils guideront nos choix technologiques et nos stratégies de développement. En intégrant ces principes



fondamentaux, nous espérons créer une solution innovante et inclusive qui répond aux besoins variés des utilisateurs tout en préservant l'histoire familiale pour les générations futures.

## Deuxième partie

# Analyse et Conception

# Chapitre 1

## Etude préliminaire et Méthode

Dans ce chapitre, nous allons présenter l'étude préliminaire de notre projet de création collaborative et de partage d'arbres généalogiques, ainsi que la méthodologie que nous allons suivre pour sa conception et son développement. Nous commencerons par une analyse approfondie de l'existant. Nous décrirons la situation actuelle de Mazala-Firm dans le domaine de la généalogie en ligne, nous identifierons ses limites et nous proposerons des solutions pour les surmonter.

Ensuite, nous présenterons la méthode que nous allons utiliser pour concevoir notre plateforme. Nous mettrons l'accent sur le langage de modélisation Unified Modeling Language (UML), ainsi que sur les différents types de diagrammes UML que nous utiliserons pour représenter notre système de manière visuelle et structurée. Nous discuterons également des processus de développement que nous allons adopter, notamment le Unified Process ou Processus Unifié (UP) et le 2TUP. Nous verrons comment ils peuvent être intégrés à UML pour assurer une gestion efficace du projet.

Ce chapitre posera les bases nécessaires à la réalisation de notre projet en fournissant une compréhension approfondie de son contexte et en établissant les lignes directrices pour sa conception et son développement. En combinant une analyse approfondie de l'existant avec une méthodologie de conception rigoureuse, nous sommes bien placés pour créer une plateforme généalogique collaborative et innovante qui répondra aux besoins de nos utilisateurs tout en garantissant sa robustesse et sa pérennité.

### 1.1 Etude de l'existant

#### 1.1.1 Description des activités (la situation actuelle)

Actuellement, Mazala-Firm ne dispose pas d'une plateforme dédiée à la création collaborative et au partage d'arbres généalogiques. Les activités liées à la généalogie sont principalement effectuées de manière traditionnelle, impliquant la collecte de documents papier, la communication verbale et l'échange de fichiers numériques par courrier électronique ou d'autres moyens non structurés. Les membres de la famille rencontrent souvent des difficultés pour collaborer efficacement à la création et au partage d'arbres généalogiques en raison de l'absence d'un outil centralisé et convivial.

#### 1.1.2 Critique de l'existant (les limites)

Cette approche traditionnelle présente plusieurs limites. Elle rend difficile la collaboration entre les membres de la famille, en particulier ceux qui sont éloignés géographiquement, et ne permet pas un partage facile et sécurisé des informations généalogiques. De plus, la préservation à long terme de l'histoire familiale est compromise, car les documents papier peuvent se perdre ou se détériorer avec le temps, et les fichiers numériques peuvent être dispersés et mal organisés.

#### 1.1.3 Proposition de solutions

Pour surmonter ces défis, nous proposons les solutions suivantes :

- Utiliser des logiciels de généalogie hors ligne : cette solution implique l'installation de logiciels sur des ordinateurs personnels pour créer et gérer des arbres généalogiques. Cependant, cela peut présenter certaines limites en termes de collaboration et de partage, car les données sont souvent stockées localement et ne permettent pas une collaboration facile entre les membres de la famille.

- Utiliser des sites web de généalogie : cela consiste à utiliser des sites web spécialisés en généalogie pour créer et partager des arbres généalogiques. Cette méthode est pratique, car elle permet de partager ses arbres avec d'autres utilisateurs. Cependant, les fonctionnalités des sites sont limitées et ne permettent pas toujours une collaboration efficace.
- Développer d'une plateforme web et mobile dédiée à la création collaborative et au partage d'arbres généalogiques : Cette solution propose de créer une plateforme personnalisée qui répond spécifiquement aux besoins de Mazala-Firm. Elle permettra aux utilisateurs de collaborer à la création d'arbres généalogiques, de partager facilement des informations avec leur famille et leurs proches, et offrira des fonctionnalités avancées, notamment la gestion de la confidentialité des données et des outils d'analyse.

#### 1.1.4 Choix de la solution

Parmi les solutions présentées, nous choisissons de développer une plateforme web et mobile dédiée à la création collaborative et au partage d'arbres généalogiques. Elle présente la meilleure combinaison de convivialité, de fonctionnalités avancées et de contrôle sur la confidentialité des données. En offrant une plateforme centralisée et sécurisée, nous pouvons garantir une expérience utilisateur optimale tout en répondant aux besoins diversifiés des utilisateurs.

En résumé, notre choix reflète notre engagement à fournir une solution efficace et innovante pour relever les défis de Mazala-Firm.

## 1.2 Méthode

La méthode de développement joue un rôle majeur dans la réalisation efficace d'un projet informatique. En effet, une méthode d'analyse et de conception est un procédé qui a pour objectif de formaliser les étapes du développement d'un système en vue de s'assurer d'une compréhension fidèle des besoins des utilisateurs et d'une adéquation des solutions proposées en termes de couverture des besoins des utilisateurs. Elle fournit donc un cadre structuré pour planifier, concevoir et mettre en œuvre des solutions logicielles répondant aux besoins spécifiques des utilisateurs.

Il existe plusieurs types de méthodes d'analyse et de conception d'un système d'information. Parmi celles-ci, on compte :

- **Les méthodes cartésiennes ou fonctionnelles** : elles consistent à étudier le système par les fonctions qu'il doit assurer plutôt que par les données qu'il doit assurer. Le processus de conception est vu comme un développement linéaire. Le domaine étudié est décomposé en sous domaines, eux-mêmes décomposés en sous domaines jusqu'à un niveau élémentaire ;
- **Les méthodes objet** : (UML souvent associé à un UP), elles sont basées sur la notion d'objet. Elle permet de modéliser les objets du système, leurs attributs, leurs méthodes et leurs interactions. Elle est largement utilisée dans l'industrie du logiciel pour concevoir et documenter les systèmes logiciels ;
- **Les méthodes agiles** : elles sont basées sur des principes collaboratifs et itératifs. Elles visent à fournir des solutions logicielles de haute qualité en s'adaptant aux besoins changeants des utilisateurs. Parmi les méthodes agiles les plus populaires, on compte Scrum, Kanban et Extreme Programming (XP).
- **Les méthodes formelles** : elles s'appuient sur des techniques mathématiques pour spécifier, concevoir et vérifier les systèmes logiciels. Elles sont utilisées pour garantir la correction et la fiabilité des systèmes critiques.

<b>Méthodes</b> <b>Critères</b>	<b>Cartésiennes</b>	<b>Systémiques</b>	<b>Agiles</b>	<b>Objet</b>	<b>Formelles</b>
Itératif	✓	✓	✓	✓	✗
Prise en charge de projets d'envergure	✓	✓	✗	✓	✗
Incrémental	✗	✗	✓	✓	✗
Axé sur la documentation	✗	✓	✗	✓	✗
Gestion des risques	✗	✗	✓	✓	✗
Simplicité de mise en œuvre	✗	✓	✓	✗	✗
Langage de modélisation	✗	✓	✗	✓	✓
Dialogue entre les intervenants du projet	✓	✓	✓	✓	✗
Pilotage par les cas d'utilisation	✗	✗	✓	✓	✗
Axé sur le développement	✗	✗	✓	✗	✗

TABLE 1.1 – Récapitulatif des méthodes

Nous avons opté pour la méthode objet en raison de ses avantages. Rappelons qu'une méthode est constituée d'un couplage entre un langage de modélisation et un processus. Notre choix s'est arrêté sur le processus 2TUP et le langage de modélisation UML.

### 1.2.1 Le langage de modélisation

Un langage de modélisation est un ensemble de conventions et de symboles utilisés pour représenter les différents aspects d'un système complexe, comme un logiciel, un processus ou une organisation. Selon WEILKIENS [2], un langage de modélisation est défini comme "un langage formel utilisé pour exprimer les caractéristiques, les comportements et les interactions d'un système sous une forme abstraite et structurée". Cette abstraction permet aux développeurs et aux ingénieurs de communiquer efficacement et de collaborer sur la conception, la modélisation et l'analyse des systèmes complexes. Ainsi, l'importance d'un langage de modélisation réside dans sa capacité à fournir une représentation visuelle claire et concise des différents éléments d'un système, ainsi que de leurs interactions. Il existe deux types de langage de modélisation : les langages de modélisation graphique et les langages de modélisation textuelle.

Les langages de modélisation graphique utilisent des techniques de diagramme avec des symboles associés à des noms qui représentent les concepts, des lignes qui connectent les symboles et qui représentent les relations et d'autres annotations graphiques pour représenter les contraintes.

Les langages de modélisation textuelle utilisent typiquement des mots-clés standardisés accompagnés de paramètres pour rendre les expressions interprétables par les ordinateurs.

#### 1.2.1.1 Présentation d'UML

UML est un langage de modélisation largement utilisé dans l'industrie du logiciel pour la conception, la documentation et la communication des systèmes logiciels. Initialement créé par Grady Booch, Ivar Jacobson et James Rumbaugh, UML est devenu une norme de facto pour la modélisation des systèmes orientés objet.

UML propose un ensemble de conventions et de diagrammes qui permettent de représenter visuellement les différents aspects d'un système logiciel. Ces diagrammes peuvent être utilisés pour décrire la structure statique d'un système, ses comportements dynamiques, ses interactions avec d'autres systèmes, et bien plus encore. En utilisant UML, les développeurs et les ingénieurs peuvent collaborer efficacement, communiquer clairement et documenter de manière précise les spécifications d'un projet logiciel.

Selon FOWLER [1], UML peut être considéré comme le "langage de modélisation standard" pour exprimer la conception d'un système logiciel. En fournissant un cadre formel et structuré pour la modélisation des systèmes informatiques, UML permet aux parties prenantes du projet de mieux comprendre les besoins, les exigences et les contraintes du système. Cela facilite la prise de décision et la gestion des risques tout au long du cycle de vie du projet.

### 1.2.1.2 Les types de diagrammes d'UML

UML propose une variété de diagrammes qui permettent de représenter différents aspects d'un système logiciel. Ces diagrammes sont catégorisés en deux grandes classes : les diagrammes structurels et les diagrammes comportementaux.

Les diagrammes structurels sont utilisés pour décrire la structure statique d'un système logiciel, c'est-à-dire ses composants, leurs relations et leur organisation. Parmi les principaux diagrammes structurels d'UML, on compte :

- Le diagramme de classe : il représente les classes du système, ainsi que leurs attributs, méthodes et relations.
- Le diagramme d'objet : il montre des instances spécifiques de classes et les liens entre elles.
- Le diagramme de composants : il décrit les composants physiques du système et leurs dépendances.
- Le diagramme de déploiement : il représente la configuration matérielle sur laquelle le système est déployé.

Quant aux diagrammes comportementaux, ils se concentrent sur le comportement dynamique du système, c'est-à-dire sur la manière dont il réagit et interagit avec son environnement. Les principaux diagrammes comportementaux d'UML sont les suivants :

- Le diagramme de cas d'utilisation : il décrit les interactions entre le système et ses utilisateurs, en mettant l'accent sur les fonctionnalités offertes par le système.
- Le diagramme de séquence : il montre la séquence des messages échangés entre les objets du système au cours d'un scénario d'exécution.
- Le diagramme d'état : il représente les différents états possibles d'un objet et les transitions entre ces états.
- Le diagramme d'activité : qui décrit le flux de contrôle entre les différentes activités du système.

Il est important de souligner que : UML n'est pas une méthode ; l'utilisation des diagrammes est donc laissée à l'appréciation de chacun. Le diagramme de classes est généralement considéré comme l'élément central d'UML.

## 1.2.2 Processus de développement

Dans le domaine du génie logiciel, le processus de développement est un ensemble d'activités et d'étapes requises pour la création d'un système logiciel. Parmi les nombreuses approches disponibles, deux méthodes largement utilisées sont l'Unified Process (UP) et le Two-Tiered Unified Process (2TUP).

### 1.2.2.1 Présentation du UP

L'UP est un processus itératif et incrémental de développement logiciel. Il est basé sur le principe de gestion de projet itératif et itératif-incrémental, dans lequel le développement est divisé en plusieurs phases, chacune produisant des livrables spécifiques. Ces phases comprennent l'élaboration, la construction, la transition, et la production de divers artefacts tels que des spécifications, des modèles UML, et des versions de logiciel partielles ou complètes. L'UP se distingue par son adaptabilité, sa flexibilité et sa focalisation sur la qualité du produit final.

### 1.2.2.2 Présentation du 2TUP

Le 2TUP est une adaptation du UP conçue pour être plus flexible et adaptable aux projets de taille moyenne ou grande. Il se compose de deux branches : la branche fonctionnelle et la branche technique, ainsi qu'une phase de réalisation. La branche fonctionnelle permet de capturer les besoins, les spécifications et l'analyse. La branche technique permet la capture des besoins techniques, l'architecture logicielle et applicative ainsi que les frameworks techniques. Quant à la phase de réalisation, elle inclut la conception, le codage, les tests et la recette ou le déploiement.

Cette approche permet une personnalisation accrue du processus de développement en fonction des particularités et des contraintes du projet.

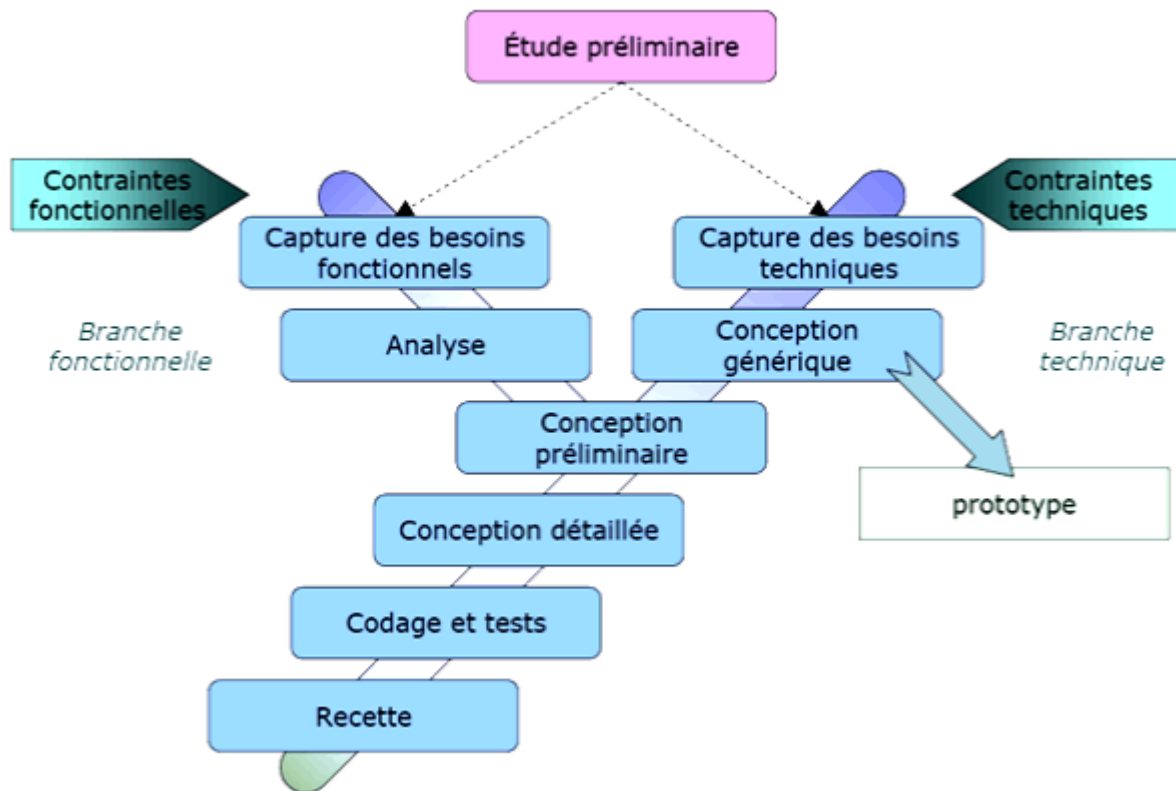


FIGURE 1.1 – Illustration du processus de développement en Y

### 1.2.2.3 2TUP et UML

L'utilisation de l'UML est étroitement liée au 2TUP. Les différents types de diagrammes UML (diagramme de cas d'utilisation, de classes, de séquence et d'état) sont largement utilisés tout au long du processus de développement pour modéliser et documenter les besoins, les architectures, les interactions et les comportements du système logiciel en cours de développement. L'UML fournit un langage visuel standard et universellement accepté pour la représentation des systèmes logiciels. Cela facilite la communication et la collaboration entre les membres de l'équipe de développement et les parties prenantes du projet.

## Chapitre 2

# Analyse et conception

Ce chapitre se concentre sur l'analyse approfondie des besoins de la plateforme web et mobile pour la création collaborative et le partage d'arbres généalogiques à Mazala-Firm. Il présente également la conception de cette plateforme. L'analyse approfondie des besoins est une étape importante dans le processus de développement, car elle permet de définir clairement les fonctionnalités et les exigences techniques nécessaires à la réalisation du projet. La conception, quant à elle, vise à traduire ces besoins en solutions techniques efficaces et robustes. Ce chapitre présentera donc en détail les besoins fonctionnels et techniques identifiés lors de l'analyse, donnant ainsi les bases solides pour la conception de la plateforme.

### 2.1 Analyse du besoin

L'analyse du besoin vise à comprendre en profondeur les attentes et les exigences des utilisateurs ainsi que les contraintes techniques et fonctionnelles qui guideront le développement de la plateforme. Cette analyse repose sur une étude approfondie des besoins fonctionnels et techniques.

#### 2.1.1 Besoins fonctionnels

Les besoins fonctionnels définissent les fonctionnalités et les services que la plateforme doit fournir à ses utilisateurs. Ils sont étroitement liés aux objectifs et aux cas d'usage du système. Ces besoins incluent :

- Création et gestion d'arbres généalogiques : les utilisateurs enregistrés peuvent créer, modifier et supprimer des arbres généalogiques et gérer les informations sur leurs proches ;
- Consultation des arbres existants : les utilisateurs doivent pouvoir consulter les arbres généalogiques disponibles, y compris les détails sur les membres de leur famille ;
- Recherche de membres de la famille : les utilisateurs doivent pouvoir trouver des personnes dans les arbres généalogiques existants sans avoir à créer de compte ;
- Collaboration entre utilisateurs sur le même arbre : les utilisateurs enregistrés doivent pouvoir collaborer à la création et à la gestion d'un arbre généalogique, en y ajoutant, modifiant ou supprimant des informations sur les membres de la famille ;
- Gestion des droits d'accès et de confidentialité : les utilisateurs doivent pouvoir déterminer qui aura accès aux arbres généalogiques qu'ils ont créés et gérés. Ils peuvent ainsi contrôler qui peut consulter, modifier ou contribuer aux informations contenues dans l'arbre ;
- Partage et diffusion des arbres : les utilisateurs doivent pouvoir partager leurs arbres généalogiques avec leur famille et leurs proches. Ceux-ci auront alors accès à des versions consultables et visualisables.

#### 2.1.2 Besoins techniques

Les besoins techniques décrivent les contraintes et les exigences liées à l'infrastructure et aux technologies utilisées pour développer la plateforme. Cela inclut :

- **Le choix des technologies de développement** : les langages de programmation, les frameworks et les outils qui seront utilisés pour créer la plateforme. Dans notre cas, nous utilisons :
  - Langage de programmation : TypeScript (web et mobile)
  - Frameworks : Next.js (web), Expo (mobile)



- **La gestion des données** : la manière dont les informations sur les arbres généalogiques et les membres de la famille seront stockées, gérées et sécurisées. Les données sont stockées dans une base de données PostgreSQL. Prisma est utilisé pour gérer les opérations de base de données, assurant l'intégrité et la sécurité des données grâce à des schémas stricts et des migrations contrôlées. Et Docker est utilisé pour créer un conteneur pour le Système de Gestion de Base de Données (SGBD) ;
- **L'interface utilisateur** : la conception de l'interface utilisateur pour garantir une expérience utilisateur intuitive et conviviale. Next.js est utilisé pour le rendu côté serveur et les pages réactives, assurant une performance élevée et une expérience utilisateur fluide ;
- **La compatibilité et la sécurité** : la compatibilité avec les navigateurs web et les appareils mobiles, la sécurité des données, la scalabilité pour gérer un grand nombre d'utilisateurs, ainsi que l'interopérabilité avec d'autres systèmes et services existants. Les pages et composants sont testés pour être compatibles avec les principaux navigateurs (Chrome, Firefox, Safari) et optimisés pour les appareils mobiles. Docker permet de facilement scaler l'application pour gérer une augmentation du nombre d'utilisateurs. Les mesures de sécurité incluent l'utilisation de HTTPS, la gestion des sessions sécurisées, et des pratiques de développement sécurisées pour éviter les vulnérabilités courantes ;
- **La gestion des erreurs** : la gestion des erreurs et des exceptions pour garantir la fiabilité et la robustesse de la plateforme. Les erreurs sont gérées de manière élégante et informative pour les utilisateurs, tout en étant enregistrées et surveillées pour permettre une résolution rapide des problèmes.

## 2.2 Conception du système

Dans cette section, nous aborderons les spécifications fonctionnelles du système. Nous décrirons les acteurs impliqués, les cas d'utilisation identifiés, ainsi que les relations et la classification de ces cas d'utilisation.

### 2.2.1 Spécifications fonctionnelles

Les spécifications fonctionnelles décrivent les fonctionnalités et les interactions du système avec ses utilisateurs. Elles fournissent une vue détaillée des besoins opérationnels du système.

#### 2.2.1.1 Identification des acteurs

Les acteurs sont les entités externes qui interagissent avec le système. Dans notre plateforme de création collaborative et de partage d'arbres généalogiques à Mazala-Firm, les acteurs identifiés comprennent :

- les membres ;
- les administrateurs ;
- et les visiteurs.

#### 2.2.1.2 Diagramme de contexte statique

Le diagramme de contexte statique représente graphiquement les acteurs et leurs relations avec le système. Il fournit une vue d'ensemble claire des interactions entre les différentes entités impliquées dans notre plateforme de création collaborative et de partage d'arbres généalogiques. Ce diagramme met en évidence les liens essentiels entre les utilisateurs inscrits, les administrateurs du système et les visiteurs non enregistrés, ce qui illustre la dynamique de notre écosystème numérique.

#### 2.2.1.3 Identification des cas d'utilisation

Les cas d'utilisation décrivent les interactions entre les acteurs et le système afin d'atteindre des objectifs spécifiques. Pour notre système, on retrouve :

- Gérer un arbre généalogique ;
- Gérer des membres ;
- Modifier les informations des membres ;
- Consulter un arbre généalogique ;

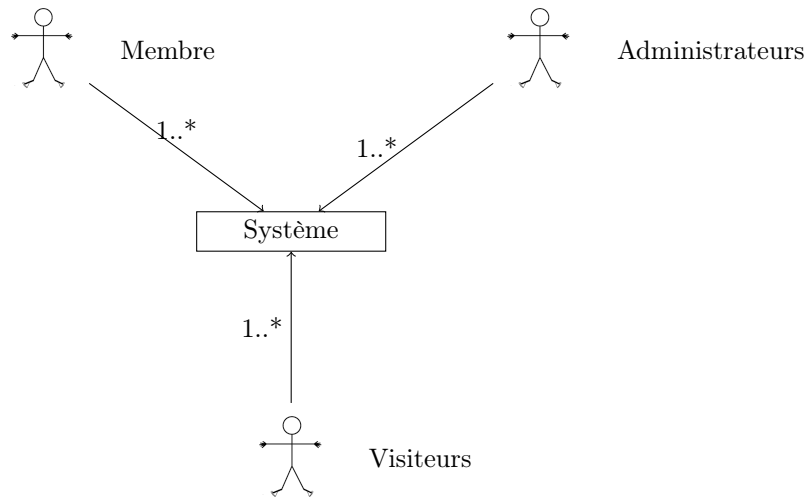


FIGURE 2.1 – Diagramme de contexte statique

- Rechercher des membres de la famille ;
- Accorder des droits d’accès et de confidentialité ;
- Partager un arbre généalogique ;
- Collaborer à la création d’un arbre généalogique ;
- Gérer un profile ;
- Administrer le système.

#### 2.2.1.4 Relation entre les cas d’utilisation (Use case)

Pour affiner la représentation des interactions entre les différents cas d’utilisation de notre système, nous utilisons les relations standardisées définies par UML. Ces relations permettent de mieux comprendre les liens entre les différentes fonctionnalités du système. Voici comment elles s’appliquent dans notre contexte :

- Inclusion (“include”) : cette relation est utilisée lorsqu’un cas d’utilisation de base incorpore explicitement un autre cas d’utilisation, de manière obligatoire. Dans notre système, nous pourrions par exemple inclure le cas d’utilisation “Modifier les informations des membres” dans le cas d’utilisation “Gérer des membres”, car la modification des informations des membres fait partie intégrante de la gestion des membres.
- Extension (“extend”) : cette relation est utilisée lorsqu’une utilisation de base incorpore implicitement un autre cas d’utilisation, de façon optionnelle. Par exemple, le cas d’utilisation “Partager un arbre généalogique” peut être étendu par le cas d’utilisation “Accorder des droits d’accès et de confidentialité”. Ainsi, lorsqu’un utilisateur souhaite partager un arbre généalogique, il a également la possibilité d’accorder des droits d’accès spécifiques à certains membres de sa famille.
- Généralisation/spécialisation : cette relation sert à décrire un lien du genre “est un” entre différents cas d’utilisation. Dans notre système, on peut trouver une généralisation entre les cas d’utilisation “Gérer un arbre généalogique” et “Gérer des membres”, puisque gérer un arbre généalogique inclut aussi la gestion des membres qui en font partie. Les fonctionnalités de gestion des membres peuvent donc être considérées comme une spécialisation du cas d’utilisation de gestion d’un arbre généalogique.

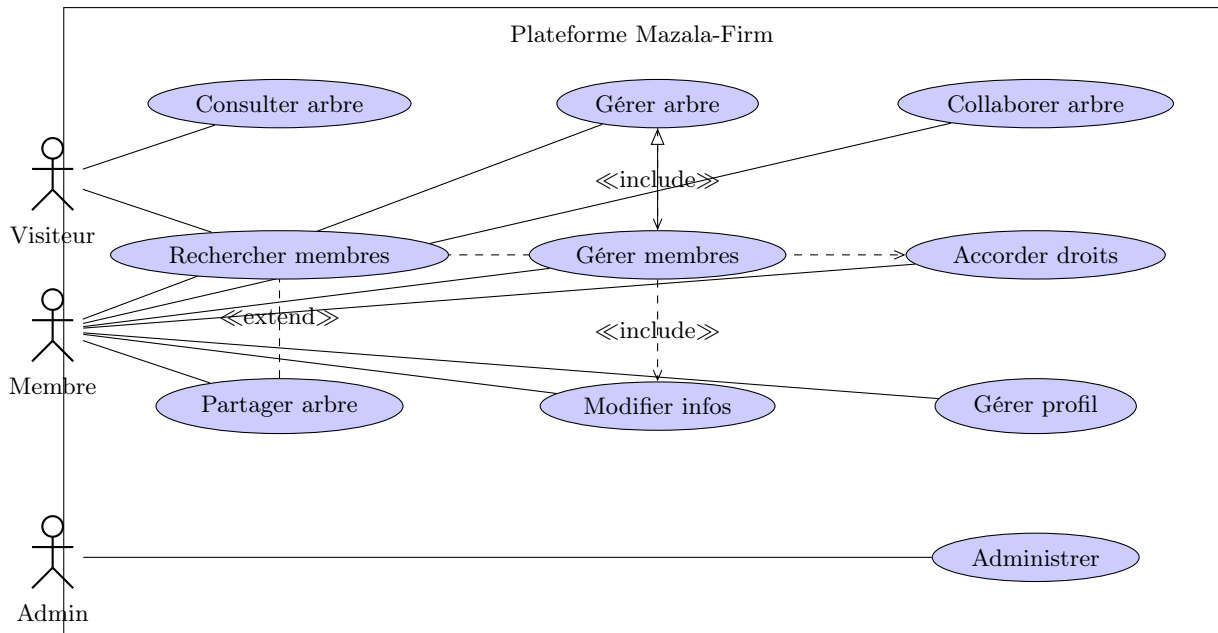


FIGURE 2.2 – Diagramme de cas d'utilisation

#### 2.2.1.5 Classification des cas d'utilisation (priorités, risques et itérations)

Dans cette section, nous classons les cas d'utilisation identifiés selon leur priorité, leur niveau de risque et leur inclusion dans les itérations du développement. Cette classification servira à guider la planification et l'exécution du projet en mettant en évidence les fonctionnalités les plus importantes, les risques potentiels à surveiller et les étapes itératives du développement.

Cas d'utilisation	Priorité	Risque	Itération
Gérer un arbre généalogique	Forte	Moyen	1
Gérer des membres	Forte	Moyen	1
Modifier les informations des membres	Moyenne	Faible	1
Consulter un arbre généalogique	Forte	Faible	1
Rechercher des membres de la famille	Moyenne	Moyen	2
Accorder des droits d'accès et de confidentialité	Moyenne	Élevé	2
Partager un arbre généalogique	Forte	Élevé	2
Collaborer à la création d'un arbre généalogique	Forte	Élevé	3
Gérer un profil	Moyenne	Moyen	3
Administrer le système	Moyenne	Élevé	3

TABLE 2.1 – Classification des cas d'utilisation en fonction de la priorité, du risque et de l'itération

Dans le tableau ci-dessus, chaque cas d'utilisation est évalué selon trois critères principaux :

- Priorité : indique l'importance relative du cas d'utilisation pour l'achèvement du projet. Les priorités peuvent être classées comme forte, moyennes ou basses.
- Risque : évalue le niveau de risque associé à la mise en œuvre du cas d'utilisation. Les risques peuvent être classés comme faibles, moyens ou élevés.
- Itération : indique dans quelle itération du développement le cas d'utilisation sera implémenté. Les itérations peuvent être numérotées séquentiellement.

Cette classification constituera un outil précieux pour la planification et la gestion du projet. Elle permettra une allocation efficace des ressources et une identification proactive des risques potentiels.

## 2.2.2 Spécification détaillée

Dans cette section, nous fournissons une spécification détaillée des cas d'utilisation identifiés. Nous décrivons les interactions entre les acteurs et le système, ainsi que les flux d'événements associés.

### 2.2.2.1 Description textuelle d'un cas d'utilisation

**Cas d'utilisation :** Gérer un arbre généalogique

**Acteur :** Utilisateur enregistré

**Autres acteurs :** Système

**Description :** Ce cas d'utilisation permet à l'utilisateur de créer, de modifier, de visualiser et de supprimer des arbres généalogiques.

**Préconditions :** L'utilisateur est authentifié et peut accéder à la fonctionnalité de gestion des arbres généalogiques.

**Postconditions :** Les modifications apportées à l'arbre généalogique sont enregistrées dans le système.

**Séquencement des événements**

Le cas d'utilisation commence lorsque l'utilisateur se connecte à la plateforme et desire gérer un arbre généalogique.

**Scénario nominal :**

1. L'utilisateur se connecte à l'application.
2. Le système affiche alors une barre de navigation latérale comportant un bouton "Créer un arbre" ainsi qu'une liste d'arbres généalogiques déjà existants chez cet utilisateur.
3. L'utilisateur peut ensuite sélectionner parmi les options suivantes :
  - Créer un nouvel arbre généalogique ;
  - Modifier un arbre généalogique existant ;
  - Consulter un arbre généalogique existant ;
  - Supprimer un arbre généalogique existant ;
  - Visualiser un arbre généalogique existant.
4. Si l'utilisateur décide de créer un nouvel arbre généalogique :
  - (a) il doit cliquer sur le bouton "Créer un nouvel arbre" ;
  - (b) le système affiche une fenêtre où il devra entrer un nom et un type(publique, privé) pour son nouvel arbre généalogique ;
  - (c) l'utilisateur valide ses informations ;
  - (d) le système crée alors un nouvel arbre généalogique et l'affiche dans la liste des arbres généalogiques de l'utilisateur.
5. Si l'utilisateur décide de modifier un arbre généalogique existant :
  - (a) il sélectionne l'arbre généalogique dans la barre de navigation latérale. ;
  - (b) le système affiche alors les détails de l'arbre généalogique sélectionné ;
  - (c) utilisateur effectue les modifications nécessaires (ajouts, suppressions, modifications de membres ou de relations) ou les informations de l'arbre généalogique ;
    - Pour ajouter un membre :
      - l'utilisateur clique sur l'option "Ajouter un membre" ;
      - le système présente un formulaire qui demande des renseignements au sujet du nouveau membre (son nom, sa date de naissance, ses relations, etc.) ;
      - l'utilisateur remplit les champs requis et valide son entrée ;
      - le système ajoute le nouveau membre à l'arbre généalogique ;
    - Pour modifier les informations de l'arbre généalogique :
      - l'utilisateur clique sur l'option "Modifier les informations" ;
      - le système affiche un formulaire de modification des informations de l'arbre généalogique ;
      - l'utilisateur effectue les modifications nécessaires et valide ses changements ;
      - le système enregistre les modifications apportées à l'arbre généalogique.
  - (d) le système sauvegarde toutes les modifications apportées à l'arbre généalogique.
6. Si l'utilisateur souhaite consulter, un arbre généalogique existant :
  - (a) il sélectionne l'arbre généalogique dans la barre de navigation latérale ;

- (b) le système affiche alors l'arbre généalogique sélectionné avec les informations sur ses membres et leurs liens.
- 7. Si l'utilisateur souhaite supprimer un arbre généalogique existant :
  - (a) il sélectionne l'arbre généalogique dans la barre de navigation latérale ;
  - (b) le système demande une confirmation avant de procéder à la suppression définitive de l'arbre généalogique ;
  - (c) si l'utilisateur accepte, le système efface l'arbre généalogique ainsi que toutes les données connexes de la liste dans la barre de navigation latérale.

#### Scénario alternatif :

#### Scénarios alternatifs

1. **Scénario alternatif pour la création d'un arbre généalogique :**
  - **Étape 4a :** L'utilisateur tente de créer un arbre généalogique mais ne saisit pas un nom valide ou ne choisit pas un type.
    - le système affiche un message d'erreur demandant de saisir un nom valide ou de choisir un type ;
    - l'utilisateur saisit un nom valide ou choisit un type et confirme ;
    - Le système crée le nouvel arbre généalogique et l'ajoute à la liste.
2. **Scénario alternatif pour la modification d'un arbre généalogique :**
  - **Étape 5a :** L'utilisateur tente d'ajouter un membre mais laisse des champs obligatoires vides.
    - le système affiche un message d'erreur demandant de compléter les champs obligatoires ;
    - l'utilisateur complète les informations manquantes et confirme ;
    - le système ajoute le nouveau membre à l'arbre généalogique.
3. **Scénario alternatif pour la suppression d'un arbre généalogique :**
  - **Étape 7a :** L'utilisateur sélectionne un arbre généalogique à supprimer mais change d'avis et annule l'opération.
    - Le système annule l'opération de suppression et revient à la liste des arbres généalogiques.
4. **Scénario alternatif pour la visualisation d'un arbre généalogique :**
  - **Étape 6a :** L'utilisateur tente de visualiser un arbre généalogique mais celui-ci contient des erreurs de données (ex. : relations incorrectes).
    - le système affiche un message d'erreur et propose des options de correction.
    - l'utilisateur corrige les erreurs et le système affiche l'arbre généalogique mis à jour.

#### Scénario d'exception :

1. Si, à l'étape 3, l'utilisateur ferme la barre de navigation latérale ,le scénario principal est interrompu.
2. Si l'utilisateur essaie de supprimer un arbre généalogique sans avoir obtenu de confirmation, le système annule

Cette description textuelle fournit une vue détaillée des étapes et des interactions impliquées dans le cas d'utilisation "Gérer un arbre généalogique". Cela permet une compréhension claire des fonctionnalités offertes par ce cas d'utilisation.

#### 2.2.2.2 Diagramme de séquence

Le diagramme de séquence est un outil visuel permettant de représenter les interactions entre les acteurs et le système dans un scénario donné. Il montre la séquence des messages échangés entre les objets du système au cours d'un scénario d'exécution. Dans notre cas, nous allons illustrer le scénario de création et de suppression d'un arbre généalogique par un utilisateur.

### Gestion d'un arbre généalogique

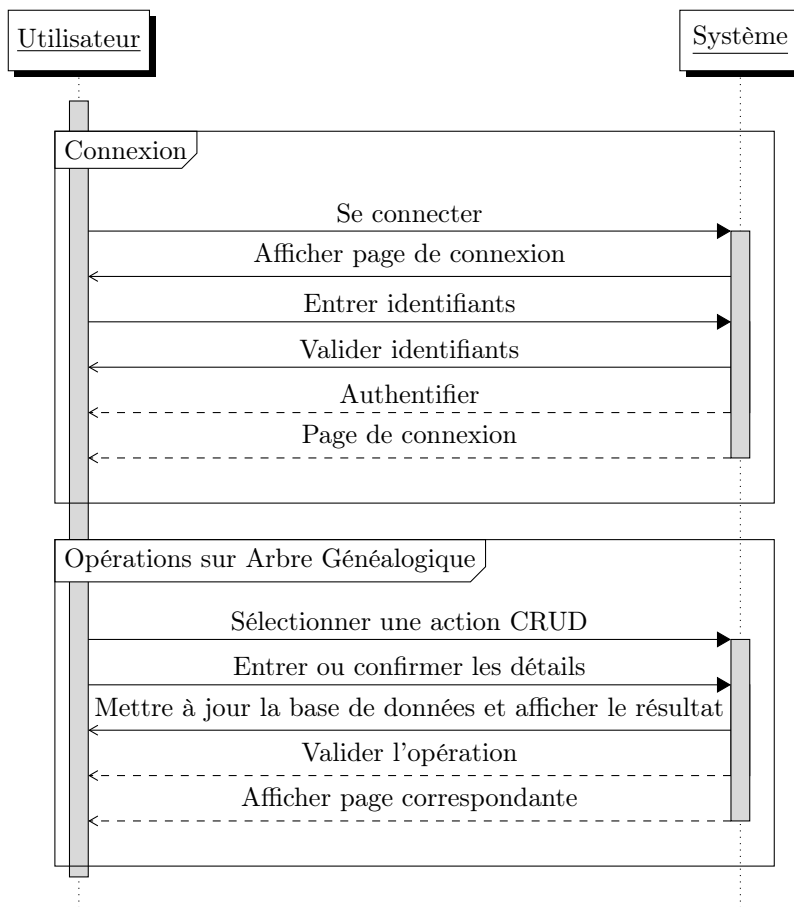


FIGURE 2.3 – Diagramme de séquence pour gestion d'un arbre généalogique

#### 2.2.2.3 Diagramme d'activité

Le diagramme d'activité est un outil visuel qui permet de représenter le flux de contrôle entre les différentes activités d'un système. Il montre comment les activités sont organisées et enchaînées pour atteindre un objectif spécifique. Dans notre cas, nous allons illustrer le processus d'ajout d'un membre à un arbre de la famille dans un arbre généalogique.

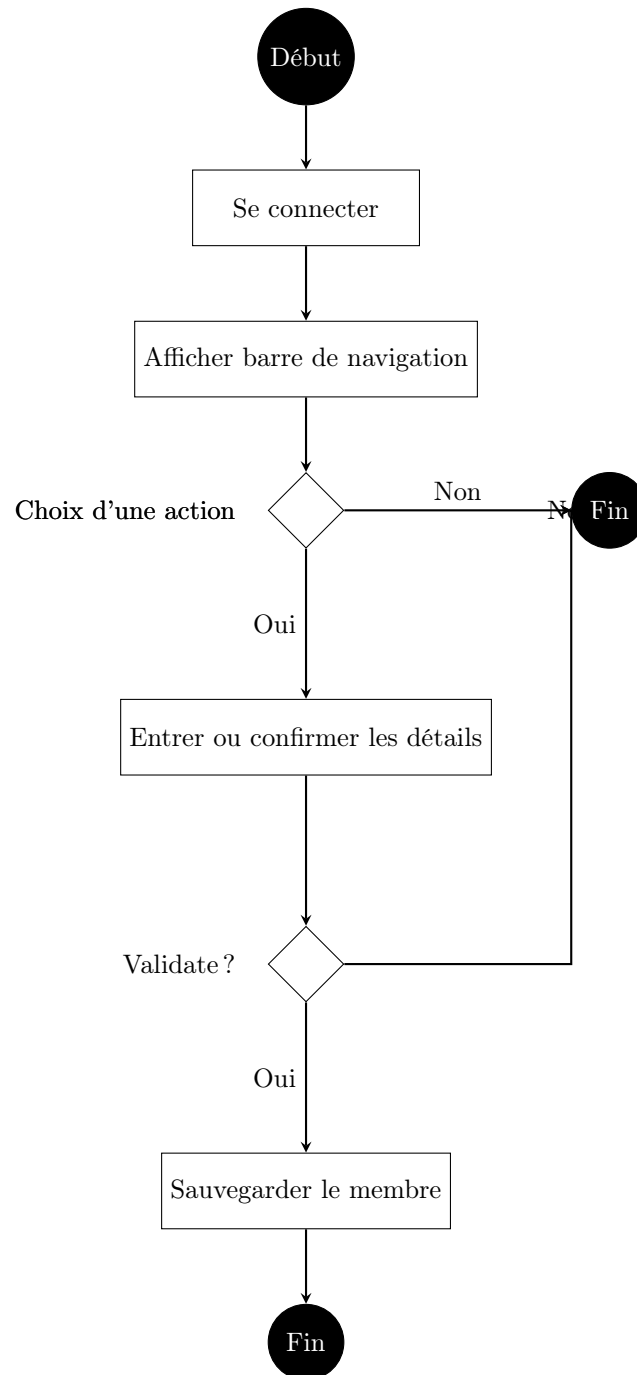


FIGURE 2.4 – Diagramme d'activité pour la gestion d'un arbre généalogique

### 2.2.3 Réalisations des cas d'utilisation

Dans cette section, nous abordons la mise en œuvre concrète des cas d'utilisation identifiés précédemment. À ce stade, nous traduisons les interactions entre les acteurs et le système en fonctionnalités opérationnelles. Nous détaillons également les règles de gestion spécifiques qui guident le comportement du système lors de l'exécution de chaque cas d'utilisation. Enfin, nous présentons le modèle du domaine sous forme d'un diagramme de classe du système, offrant ainsi une vue structurée des entités et de leurs relations au sein du système.

#### 2.2.3.1 Quelques règles de gestion

Nous allons établir ici un ensemble de règles de gestion qui régissent le comportement des fonctionnalités mises en œuvre. Ces règles définissent les contraintes et les conditions d'exécution pour garantir la cohérence et la fiabilité des opérations réalisées par le système.

- **Règle de gestion 1** : Un utilisateur ne peut pas modifier un arbre généalogique qui ne lui appartient pas.
- **Règle de gestion 2** : Un arbre généalogique ne peut pas être partagé avec des utilisateurs non enregistrés.
- **Règle de gestion 3** : Un arbre généalogique privé ne peut être consulté que par son propriétaire.
- **Règle de gestion 4** : Un arbre généalogique publique peut être consulté par tous les utilisateurs enregistrés.
- **Règle de gestion 5** : Un arbre généalogique ne peut pas être supprimé s'il contient des membres.
- **Règle de gestion 6** : Un arbre généalogique ne peut pas être partagé avec des utilisateurs qui n'ont pas de compte.

#### 2.2.3.2 Modèle du domaine (Diagramme de classe du système)

Le diagramme de classe du système est une représentation visuelle des entités principales du domaine et de leurs relations. Il offre une vue d'ensemble de la structure du système, en mettant en évidence les différentes classes d'objets, leurs attributs et leurs associations. Cette représentation facilite la compréhension des concepts clés du domaine et guide la conception de la solution logicielle.



### **2.2.3.3 Diagramme d'objet**

Un diagramme d'objet représente graphiquement les objets et leurs relations à un moment donné. À l'inverse du diagramme de classe, qui décrit la structure statique des classes et de leurs relations dans un système, le diagramme d'objet illustre des instances spécifiques des classes (objets) et les liens entre eux dans un contexte particulier. Il est utilisé pour visualiser l'état d'un système à un instant précis, facilitant la compréhension des interactions dynamiques et des configurations temporaires des objets.

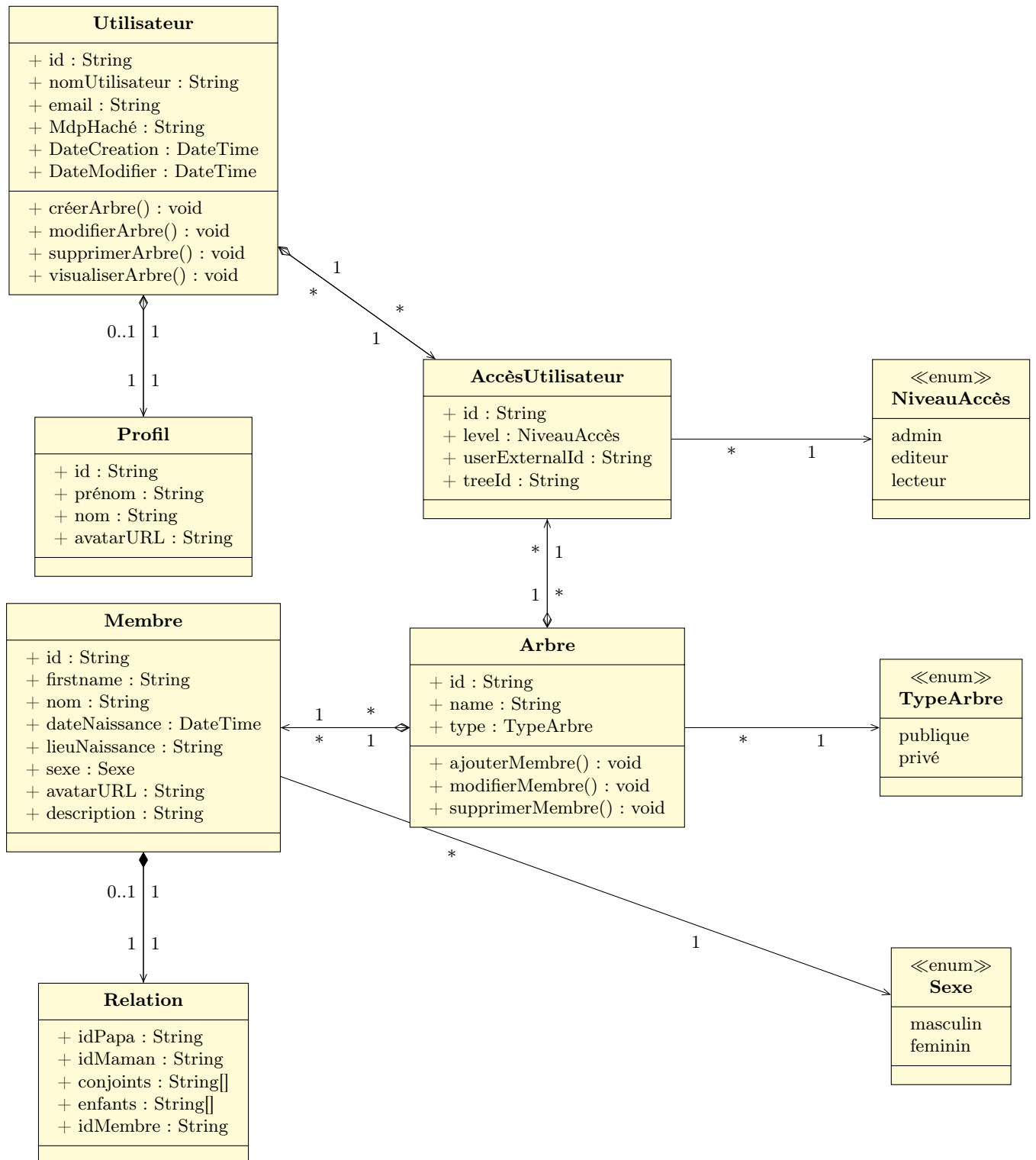


FIGURE 2.5 – Diagramme de classe du système

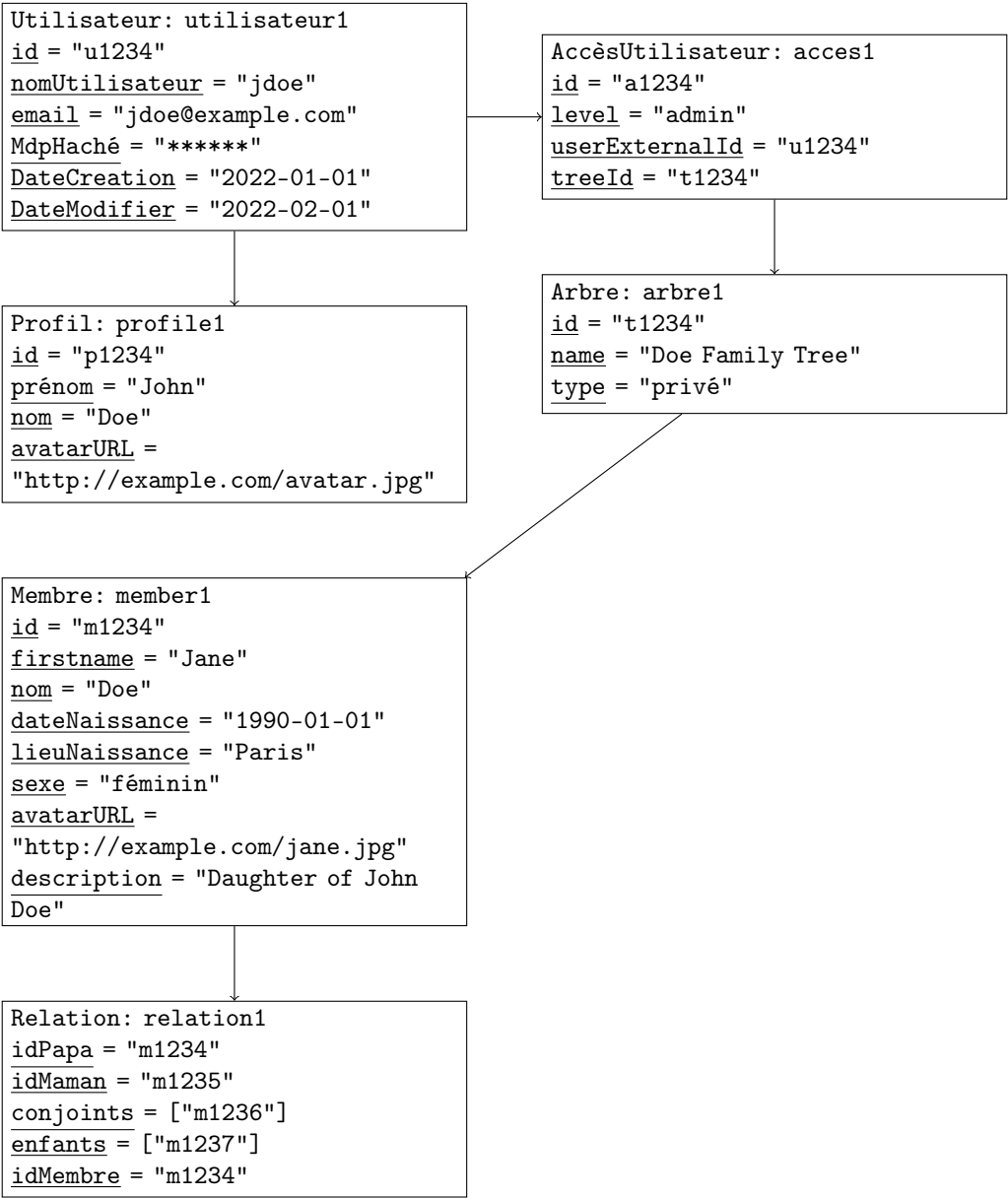


FIGURE 2.6 – Diagramme d'objet du système

## 2.2.4 Conception architecturale

La conception architecturale est une étape importante du développement logiciel qui vise à définir la structure globale d'un système et de ses composants principaux. Elle comprend l'organisation des modules, la communication entre eux et les contraintes techniques à respecter. Cette phase permet de créer une base solide pour le développement, afin d'assurer que les exigences fonctionnelles et non fonctionnelles seront satisfaites. La conception architecturale se décline en plusieurs sous-sections, notamment l'architecture logicielle et l'architecture globale de la solution.

### 2.2.4.1 Architecture logicielle

L'architecture logicielle décrit la structure organisationnelle d'un système logiciel, y compris ses modules, leurs responsabilités et les interactions entre eux. Elle vise à garantir la cohérence, la performance, la maintenabilité et la scalabilité du système. L'architecture logicielle est un guide pour les développeurs qui permet d'aligner les décisions techniques avec les objectifs métier et les contraintes du projet.

Pour notre projet, nous avons opté pour une architecture basée sur plusieurs patterns et concepts pour structurer nos applications.

- **Architecture pour l'application web**

Notre architecture se distingue par sa capacité à offrir un rendu côté serveur (SSR - Server-Side Rendering) ainsi que des pages statiques générées à la compilation (SSG - Static Site Generation). Cela améliore à la fois les performances et l'optimisation pour les moteurs de recherche (SEO).

En outre, cette architecture permet de séparer les pages, les composants et les API, offrant ainsi une organisation plus claire et modulaire.

- **Architecture pour l'application mobile**

Notre architecture offre une expérience de développement simplifiée. Elle permet de développer des applications mobiles rapidement et efficacement grâce à des outils intégrés pour la gestion de la configuration, l'accès aux API natives, et les mises à jour en direct (OTA — Over-The-Air).

Enfin, elle permet de séparer les vues, les composants et les services, offrant ainsi une organisation claire et modulaire des éléments de l'application.

Cette approche optimise le développement cross-plateforme en utilisant une seule base de code pour les applications iOS et Android, tout en garantissant une expérience utilisateur native de haute qualité.

### 2.2.4.2 Architecture globale de la solution (Diagramme de déploiement)

L'architecture globale de la solution est souvent illustrée par un diagramme de déploiement qui montre la configuration physique des composants logiciels sur le matériel. Ce diagramme détaille comment les éléments logiciels sont distribués à travers différents nœuds de réseau, tels que serveurs, bases de données, et terminaux utilisateurs. Il aide à comprendre les aspects de performance, de sécurité et de scalabilité de la solution, en montrant comment les composants interagissent dans un environnement réel.

Dans notre cas, la plateforme pour la création collaborative et le partage d'arbres généalogiques sont basés sur une architecture distribuée qui comprend plusieurs composants clés.

Une architecture distribuée désigne un système d'information ou un réseau pour lequel l'ensemble des ressources disponibles ne se trouvent pas au même endroit ou sur la même machine (WIKIPÉDIA [3]).

- **Serveur Web** : Héberge l'application Next.js pour le rendu côté serveur (SSR) et le serveur API. Utilise Node.js pour exécuter le code JavaScript côté serveur.
  - **Serveur de bases de données** : utilise PostgreSQL comme SGBD pour stocker les données relatives aux utilisateurs, aux arbres généalogiques et aux collaborations.
  - **Terminal utilisateur** : comprend les navigateurs Web pour accéder à l'application web Next.js et les applications mobiles Expo/React Native pour l'accès mobile.
  - **Services externes** : intègre éventuellement des services externes pour des fonctionnalités supplémentaires, telles que l'authentification OAuth, le stockage de fichiers et les services de messagerie.
- Voici un diagramme de déploiement illustrant cette architecture :

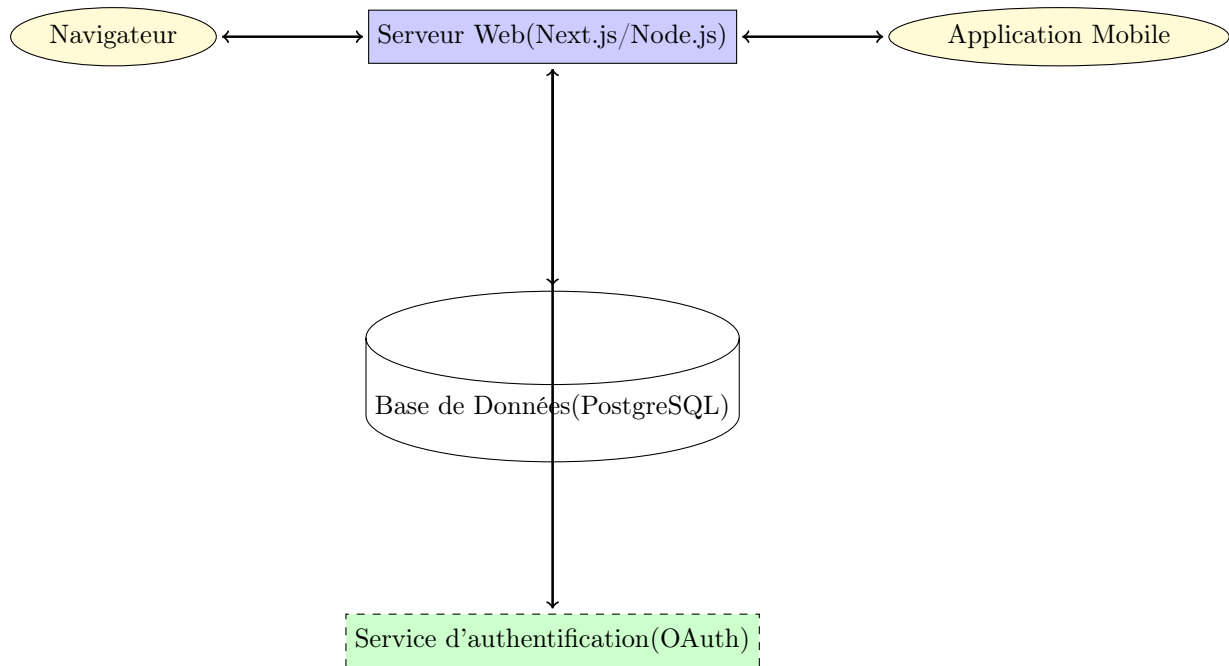


FIGURE 2.7 – Diagramme de déploiement

#### Avantages de cette architecture

- **Scalabilité** : les composants peuvent être mis à l'échelle indépendamment en fonction de la demande.
- **Sécurité** : la séparation des préoccupations permet d'améliorer la sécurité, notamment en isolant la base de données du reste de l'application.
- **Performance** : l'utilisation du rendu côté serveur et de la génération de pages statiques améliore la vitesse de chargement des pages et l'expérience utilisateur.
- **Maintenabilité** : l'architecture distribuée permet d'intégrer de nouveaux services et de mettre à jour les composants sans affecter l'ensemble du système.

## Troisième partie

# Evaluation et Réalisation

# Chapitre 1

## Evaluation du projet

Ce chapitre traite de l'organisation et de la gestion du projet, en analysant les éléments clés nécessaires à une évaluation efficace.

### 1.1 Organisation du projet

L'organisation du projet implique la définition des rôles, des responsabilités et des processus de travail. Une structure organisationnelle claire facilite la coordination et la communication entre les membres de l'équipe, assurant ainsi une progression harmonieuse du projet.

Dans le cadre de ce projet, l'organisation est structurée autour de deux principaux processus : le processus de développement et le processus de validation.

Processus	Phases
Développement	Analyse des besoins : Définir les fonctionnalités nécessaires à la création collaborative et au partage d'arbres généalogiques. Conception : Concevoir l'architecture logicielle de la solution, ainsi que l'interface utilisateur pour le web et le mobile. Développement : Implémenter les fonctionnalités selon les spécifications définies lors de l'analyse et de la conception. Tests : Réaliser des tests unitaires et d'intégration pour garantir le bon fonctionnement de l'application.
Validation	Vérification : Vérifier que les fonctionnalités implémentées répondent aux besoins et aux spécifications du projet. Validation : Valider l'application avec les utilisateurs finaux pour recueillir leurs retours et effectuer les ajustements nécessaires.

TABLE 1.1 – Organisation du projet

### 1.2 Intervenant

Intervenants	Fonctions	Rôles
M. Christopher BANDZOUZI	Ingénieur Informaticien	Directeur de projet

TABLE 1.2 – Intervenants

### 1.3 Planification des tâches

La planification des tâches consiste à décomposer le projet en activités distinctes et à établir un calendrier pour leur réalisation. Elle permet de suivre les progrès, de gérer les ressources efficacement et de respecter les délais.

Processus	Phases	Tâches
Développement	Analyse des besoins	<ul style="list-style-type: none"> <li>- Étude des fonctionnalités nécessaires à la création collaborative et au partage d'arbres généalogiques.</li> <li>- Identification des exigences de performance, de sécurité et de convivialité de la plateforme.</li> </ul>
	Conception	<ul style="list-style-type: none"> <li>- Conception de l'architecture logicielle de la solution, en mettant l'accent sur la scalabilité et la maintenabilité.</li> <li>- Conception de l'interface utilisateur pour le web et le mobile, en tenant compte des principes de conception UX/UI.</li> </ul>
	Développement	<ul style="list-style-type: none"> <li>- Implémentation des fonctionnalités de création collaborative d'arbres généalogiques sur la plateforme web.</li> <li>- Développement des fonctionnalités de partage et de visualisation des arbres généalogiques sur les applications mobiles.</li> </ul>
	Tests	<ul style="list-style-type: none"> <li>- Réalisation de tests unitaires et d'intégration pour garantir le bon fonctionnement des fonctionnalités développées.</li> <li>- Tests de performance pour évaluer la réactivité et la scalabilité de la plateforme.</li> </ul>
Déploiement	Mise en production	<ul style="list-style-type: none"> <li>- Configuration des serveurs et déploiement de l'application web Next.js sur un environnement de production sécurisé.</li> <li>- Publication des applications mobiles sur les stores (App Store et Google Play) après une phase de tests approfondis.</li> </ul>
	Formation	<ul style="list-style-type: none"> <li>- Formation des utilisateurs finaux à l'utilisation de la plateforme, en mettant l'accent sur les fonctionnalités clés et les bonnes pratiques.</li> <li>- Documentation complète de la plateforme pour une référence ultérieure.</li> </ul>
Suivi	Maintenance	<ul style="list-style-type: none"> <li>- Surveillance continue de la plateforme pour détecter et corriger les éventuels problèmes de performance ou de sécurité.</li> <li>- Mise à jour régulière de la plateforme avec de nouvelles fonctionnalités et correctifs de bugs.</li> </ul>

TABLE 1.3 – Planification des tâches



## 1.4 Diagramme de Gantt

Le diagramme de Gantt est un outil visuel de gestion de projet qui affiche les tâches à accomplir sur une ligne de temps. Il permet de suivre les progrès, d'identifier les dépendances entre les tâches et de prévoir les éventuels retards.

## 1.5 Estimation des charges

L'estimation des charges implique de calculer le temps et les ressources nécessaires pour accomplir chacune des tâches du projet. Des estimations précises sont essentielles pour la planification budgétaire et la gestion des ressources humaines.

Outil et besoin	Prix
L <sup>A</sup> T <sub>E</sub> X	0 FCFA
TypeScript	0 FCFA
Next.js	0 FCFA
Expo	0 FCFA
PostgreSQL	0 FCFA
Git	0 FCFA
GitHub	0 FCFA
Neovim	0 FCFA
VSCodium	0 FCFA
Figma	0 FCFA
PC Portable	800.000 FCFA
Connexion internet	25 000/mois FCFA
<b>Total</b>	<b>825 000 FCFA</b>

TABLE 1.4 – Estimation des charges

## Chapitre 2

# Les outils et techniques utilisés

Ce chapitre présente les outils et les techniques utilisées pour développer et gérer le projet.

### 2.1 Présentation des techniques

Les techniques utilisées dans le cadre de ce projet sont les suivantes :

#### Développement Web et Mobile

Pour le développement de la plateforme, nous avons utilisé des techniques modernes adaptées au web et aux applications mobiles.

- **Next.js** Next.js est un framework React qui permet de créer des applications web côté serveur (SSR) et des applications statiques (SSG). Il offre des fonctionnalités avancées telles que le rendu côté serveur, le pré-rendu statique, et une gestion automatique des routes, ce qui améliore la performance et le SEO des applications web.
- **Expo** Expo est une plateforme et un ensemble d'outils pour le développement d'applications mobiles avec React Native. Elle simplifie le processus de développement, de déploiement et de mise à jour des applications mobiles, tout en offrant des fonctionnalités telles que le rechargement en temps réel, l'accès aux API natives et une gestion simplifiée des dépendances.

#### Base de Données

Pour stocker les données de l'application, nous avons utilisé une base de données relationnelle PostgreSQL, qui offre des fonctionnalités avancées telles que la conformité ACID, la gestion des transactions, et la prise en charge de données complexes.

#### Sécurité

La sécurité est un aspect crucial du développement de la plateforme. Nous avons mis en place plusieurs techniques de sécurité pour protéger les données des utilisateurs.

- **Authentification et Autorisation** Nous avons utilisé des techniques modernes d'authentification et d'autorisation, telles que JWT (JSON Web Tokens) pour sécuriser les accès à l'application et aux API.
- **Chiffrement des Données** Toutes les communications entre les clients et le serveur sont chiffrées en utilisant TLS (Transport Layer Security), garantissant que les données transmises restent confidentielles et intactes.

#### Gestion de Projet

Pour la gestion du projet, nous avons utilisé la méthode objet, qui permet d'organiser les tâches en fonction de leur importance et de leur priorité.

### 2.2 Présentation des outils

Les outils utilisés dans le cadre de ce projet sont les suivants :

#### Outils de Développement

##### — Neovim

Neovim est un éditeur de texte puissant et extensible, adapté au développement de logiciels. Il offre des fonctionnalités avancées telles que la coloration syntaxique, la complétion automatique, et la gestion des plug-ins, ce qui améliore la productivité des développeurs.



FIGURE 2.1 – Logo Neovim

— **VSCodium**

VSCodium est un éditeur de code open-source basé sur Visual Studio Code,



FIGURE 2.2 – Logo VSCodium

**Frameworks et Bibliothèques**

— **React**

React est la bibliothèque JavaScript principale utilisée pour construire les interfaces utilisateur de la plateforme web.

— **React Native**

React Native est le framework JavaScript utilisé pour développer les applications mobiles de la plateforme.

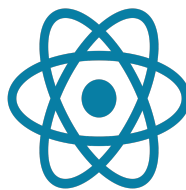


FIGURE 2.3 – Logo React

— **Next.js**

Next.js est le framework React utilisé pour le développement côté serveur et la génération de sites statiques de la partie web de la plateforme.



FIGURE 2.4 – Logo Next.js

— **Expo**

Expo est la plateforme utilisée pour le développement et le déploiement des applications mobiles, en fournissant un ensemble d'outils et de services pour React Native.



FIGURE 2.5 – Logo Expo

— **shadcn/ui**

shadcn/ui est une bibliothèque de composants React réutilisables, qui permet de créer des interfaces utilisateur cohérentes et esthétiques.

**Outils de Contrôle de Version**

— **Git**

Git est le système de contrôle de version distribué utilisé pour suivre les modifications du code source et faciliter la collaboration entre les développeurs.



FIGURE 2.6 – Logo Git

— **GitHub**

GitHub est la plateforme utilisée pour héberger le code source et faciliter la collaboration via des fonctionnalités comme les pull requests et les revues de code.

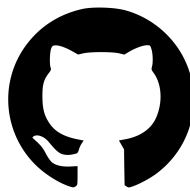


FIGURE 2.7 – Logo GitHub

**Outils de Conception**

— **Figma**

Figma est l'outil de conception utilisé pour créer les maquettes et les prototypes de l'interface utilisateur de la plateforme, en facilitant la collaboration et le partage des designs.

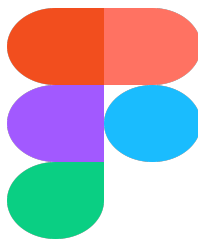


FIGURE 2.8 – Logo Figma

— **L<sup>A</sup>T<sub>E</sub>X**

L<sup>A</sup>T<sub>E</sub>X est un système de composition de documents utilisé pour rédiger la documentation technique et les rapports du projet. Il offre une mise en page professionnelle et une gestion avancée des références. Il permet également de créer des diagrammes UML.



FIGURE 2.9 – Logo L<sup>A</sup>T<sub>E</sub>X

## Outils de Base de Données

### — PostgreSQL

PostgreSQL est le système de gestion de base de données relationnelle utilisé pour stocker les données de l'application, en offrant des fonctionnalités avancées de gestion des données et de sécurité.

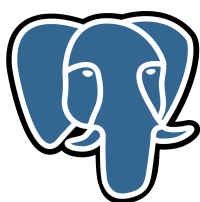


FIGURE 2.10 – Logo PostgreSQL

### — Prisma

Prisma est un Object Relational Mapping (ORM) utilisé pour simplifier l'accès à la base de données PostgreSQL et faciliter les opérations de lecture et d'écriture des données.

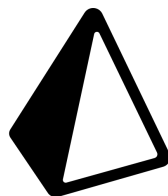


FIGURE 2.11 – Logo Prisma

### — tRPC

tRPC est une bibliothèque de communication entre le client et le serveur qui facilite l'appel des API et la gestion des données de manière sécurisée et efficace.



FIGURE 2.12 – Logo tRPC

## Outils de Sécurité

### — JWT (JSON Web Tokens)

JWT est utilisé pour l'authentification et l'autorisation sécurisées des utilisateurs, en générant des jetons d'accès et de rafraîchissement pour contrôler l'accès aux ressources de l'application.



FIGURE 2.13 – Logo tRPC

— **TLS (Transport Layer Security)**

TLS est utilisé pour chiffrer les communications entre les clients et le serveur, en garantissant la confidentialité et l'intégrité des

**Outils de conteneurisation**

— **Docker**

Docker est utilisé pour créer des conteneurs légers et portables qui encapsulent les applications et leurs dépendances, facilitant le déploiement et la gestion des applications dans différents environnements.



FIGURE 2.14 – Logo Docker

**Outils de Déploiement**

— **Vercel**

Vercel est utilisé pour le déploiement et l'hébergement de la partie web de la plateforme développée avec Next.js.



FIGURE 2.15 – Logo Vercel

— **Expo Go**

Expo Go est utilisé pour tester et déployer les applications mobiles développées avec Expo et React Native.

## 2.3 Les captures d'écrans

Les captures d'écran illustrent visuellement les étapes importantes du développement et l'interface utilisateur du produit final. Elles servent à documenter le travail accompli et à fournir des exemples concrets de l'application en action.

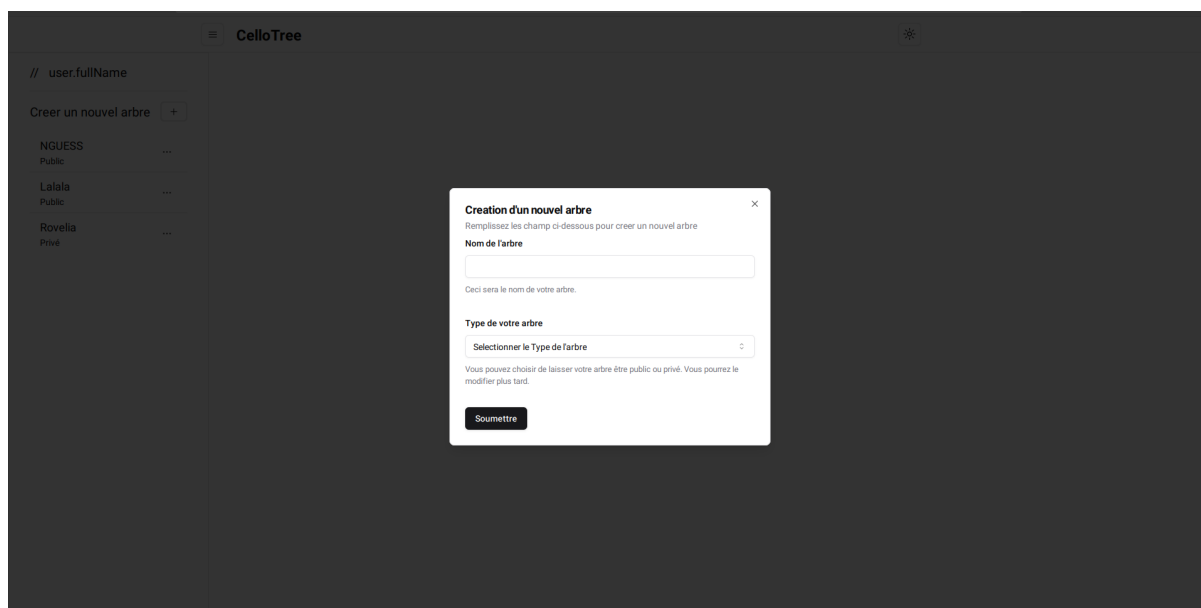


FIGURE 2.16 – Écran de création d'un arbre généalogique sur l'application web



# Conclusion

# Table des matières

Dédicace	i
Remerciements	iii
Table des figures	iv
Liste des tableaux	v
Sommaire	vi
Abréviation et Sigles	vii
Introduction	1
<b>I Présentation Générale</b>	<b>2</b>
<b>1 Structure d'Accueil et le Sujet</b>	<b>3</b>
1.1 Structure d'Accueil . . . . .	3
1.1.1 Historique . . . . .	3
1.1.2 Missions . . . . .	3
1.1.3 Organigramme Général . . . . .	4
1.1.4 Attributions des structures . . . . .	4
1.1.5 Situation informatique . . . . .	5
1.1.5.1 Personnel informatique . . . . .	5
1.1.5.2 Matériels informatiques . . . . .	5
1.1.5.3 Logiciels informatiques . . . . .	5
1.2 Sujet . . . . .	7
1.2.1 Contexte du Sujet . . . . .	7
1.2.2 Problématique du sujet . . . . .	7
1.2.3 Intérêts du sujet . . . . .	7
1.3 Concepts liés au sujet . . . . .	8
<b>II Analyse et Conception</b>	<b>10</b>
<b>1 Etude préliminaire et Méthode</b>	<b>11</b>
1.1 Etude de l'existant . . . . .	11
1.1.1 Description des activités (la situation actuelle) . . . . .	11
1.1.2 Critique de l'existant (les limites) . . . . .	11
1.1.3 Proposition de solutions . . . . .	11
1.1.4 Choix de la solution . . . . .	12
1.2 Méthode . . . . .	12
1.2.1 Le langage de modélisation . . . . .	13
1.2.1.1 Présentation d'UML . . . . .	13
1.2.1.2 Les types de diagrammes d'UML . . . . .	14
1.2.2 Processus de développement . . . . .	14
1.2.2.1 Présentation du UP . . . . .	14

1.2.2.2	Présentation du 2TUP . . . . .	14
1.2.2.3	2TUP et UML . . . . .	15
<b>2</b>	<b>Analyse et conception</b>	<b>16</b>
2.1	Analyse du besoin . . . . .	16
2.1.1	Besoins fonctionnels . . . . .	16
2.1.2	Besoins techniques . . . . .	16
2.2	Conception du système . . . . .	17
2.2.1	Spécifications fonctionnelles . . . . .	17
2.2.1.1	Identification des acteurs . . . . .	17
2.2.1.2	Diagramme de contexte statique . . . . .	17
2.2.1.3	Identification des cas d'utilisation . . . . .	17
2.2.1.4	Relation entre les cas d'utilisation (Use case) . . . . .	18
2.2.1.5	Classification des cas d'utilisation (priorités, risques et itérations) . . . . .	19
2.2.2	Spécification détaillée . . . . .	20
2.2.2.1	Description textuelle d'un cas d'utilisation . . . . .	20
2.2.2.2	Diagramme de séquence . . . . .	21
2.2.2.3	Diagramme d'activité . . . . .	22
2.2.3	Réalisations des cas d'utilisation . . . . .	24
2.2.3.1	Quelques règles de gestion . . . . .	24
2.2.3.2	Modele du domaine (Diagramme de classe du système) . . . . .	24
2.2.3.3	Diagramme d'objet . . . . .	25
2.2.4	Conception architecturale . . . . .	26
2.2.4.1	Architecture logicielle . . . . .	26
2.2.4.2	Architecture globale de la solution (Diagramme de déploiement) . . . . .	26
<b>III</b>	<b>Evaluation et Réalisation</b>	<b>28</b>
<b>1</b>	<b>Evaluation du projet</b>	<b>29</b>
1.1	Organisation du projet . . . . .	29
1.2	Intervenant . . . . .	29
1.3	Planification des tâches . . . . .	29
1.4	Diagramme de Gantt . . . . .	31
1.5	Estimation des charges . . . . .	32
<b>2</b>	<b>Les outils et techniques utilisés</b>	<b>33</b>
2.1	Présentation des techniques . . . . .	33
2.2	Présentation des outils . . . . .	33
2.3	Les captures d'écrans . . . . .	37
	<b>Conclusion</b>	<b>39</b>
	<b>Webographie</b>	<b>43</b>

# Bibliographie

- [1] Martin FOWLER. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 2003.
- [2] Tim WEILKIENS. *Systems Engineering with SysML/UML: Modeling, Analysis, Design*. Morgan Kaufmann, 2008.
- [3] WIKIPÉDIA. *Architecture distribuée* — *Wikipédia, l'encyclopédie libre*. [En ligne; Page disponible le 25-février-2024]. 2024. URL : [http://fr.wikipedia.org/w/index.php?title=Architecture\\_distribu%C3%A9e&oldid=212787328](http://fr.wikipedia.org/w/index.php?title=Architecture_distribu%C3%A9e&oldid=212787328).

# Webographie