

E. S. G. A. E
ÉOLE SUPÉRIEURE 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 ; esgae@esgae.org

Site Web : <https://www.esgae.org>

.....

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

THÈME

ÉTUDE ET MISE EN ŒUVRE D'UNE PLATEFORME WEB ET
MOBILE POUR LA CRÉATION COLLABORATIVE ET LE
PARTAGE D'ARBRES GÉNÉALOGIQUES À MAZALA-FIRM

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

Sous la direction de :
M. Chistopher BANDZOUZI
Ingénieur informaticien

Année académique : 2023 – 2024

Dédicace

Je dédie ce modeste travail à :

- Mes parents Rock Frédy ONIANGUE-DESO et Zita NGANBOMI, 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.
- Mes professeurs, pour leur expertise, leur patience et leur inspiration. Leurs conseils éclairés ont enrichi mon parcours académique et ont façonné ma pensée critique.
- 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.
- Mon adorable conjointe Merveille Chrisnat DOUMA LEBONGUI.
- 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 personnalisations près).

Samuel Exaucé NANDI

Remerciements

Nous souhaitons exprimer nos plus sincères remerciements à :

- Monsieur le professeur titulaire Roger Armand MAKANY, Directeur Général de l'ES-GAE, pour nous avoir accueillis dans cette enceinte universitaire ;
- Monsieur Christopher BANDZOUZI notre Directeur de Projet pour sa longue patience et sa compréhension sans égale ;
- Mesdames Vérita YAOUE-NGUESSIMOU et Marly POUABOUD-EUGE, étudiantes à l'ESGAE, pour leur aide précieuse ;
- Tout le personnel enseignant et administratif de l'ESGAE ;
- Tout le personnel de MAZALA-FIRM pour sa disponibilité, son aide et son accueil chaleureux ;
- Tous nos camarades de classe et nos collègues de l'école.
- Nos parents et nos familles pour leur soutien.

Table des figures

1.1	Organigramme de Mazala-Firm	4
1.1	Illustration du processus de développement en Y	16
2.1	Diagramme de contexte statique	19
2.2	Diagramme de cas d’utilisation	21
2.3	Diagramme de séquence du cas d’utilisation gérer un arbre généalogique	27
2.4	Diagramme de séquence du cas d’utilisation accoder des droits d’accès	28
2.5	Diagramme d’activité du cas d’utilisation gérer un arbre généalogique	29
2.6	Diagramme d’activité du cas d’utilisation accorder des droits d’accès	30
2.7	Diagramme de classe du système	32
2.8	Diagramme d’objet du système	33
2.9	Architecture du logiciel sous VSCodium	35
2.10	Illustration d’une architecture distribuée	36
2.11	Diagramme de déploiement logique	36
1.1	Diagramme de Gantt	40
2.1	Logo Neovim	43
2.2	Logo VSCodium	43
2.3	Logo Capacitor	43
2.4	Logo Next.js	44
2.5	Logo Git	44
2.6	Logo GitHub	44
2.7	Logo Figma	44
2.8	Logo L ^A T _E X	45
2.9	Logo PostgreSQL	45
2.10	Logo Prisma	45
2.11	Logo tRPC	45
2.12	Logo Docker	46
2.13	Logo Vercel	46
2.14	Logo AppFlow	46
2.15	Écran d’accueil sur l’application web	47
2.16	Écran de connexion sur l’application web	47
2.17	Écran de création de comptes sur l’application web	48
2.18	Écran de création d’un arbre généalogique sur l’application web	48
2.19	Écran d’ajout d’un membre à un généalogique sur l’application web	49
2.20	Écran de visualisation d’un arbre généalogique	49
2.21	Écran d’accueil sur l’application mobile	50
2.22	Écran de connexion sur l’application mobile	50
2.23	Écran de création d’un arbre généalogique sur l’application mobile	51
2.24	Écran d’ajout d’un membre à un généalogique sur l’application mobile	51

Liste des tableaux

1	Abréviation ou sigle et significations	vii
1.1	Récapitulatif des méthodes	13
2.1	Classification des cas d’utilisation	21
1.1	Organisation du projet	39
1.2	Intervenants	39
1.3	Planification des tâches	40
1.4	Estimation des charges	41

Sommaire

Introduction	1
Première Partie : Présentation générale de Mazala-Firm	3
Deuxième Partie : Analyse et conception	11
Troisième Partie : Évaluation et Réalisation	39
Conclusion	52
Table des matières	A
Bibliographie	C
Webographie	D

Abréviation et Sigles

Abréviation ou sigle	Signification
ESGAE	École Supérieure de Gestion et d’Administration des Entreprises
LPGL	Licence Professionnelle en Génie Logiciel
2TUP	Two Tracks Unified Process
UP	Unified Process ou Processus Unifié
UI	User Interface ou interface utilisateur en français
UX	User Experience ou expérience utilisateur en français
UML	Unified Modeling Language
API	Application Programming Interface
RAM	Random Access Memory ou Mémoire vive en français
Go	Giga octet
SSD	Solid State Drive ou Disque à état solide en français
HD	High Definition ou Haute Définition en français
sRGB	Standard Red Green Blue ou Rouge Vert Bleu en français
RAID	Redundant Array of Independent Disks ou Ensemble redondant de disques indépendants en français
NVMe	Non-Volatile Memory Express ou Express de mémoire non volatile en français
VPN	Virtual Private Network ou Réseau Privé Virtuel en français
XP	Extreme Programming
ORM	Object Relational Mapping
SGBD	Système de Gestion de Base de Données
SQL	Structured Query Language
CRUD	Create, Read, Update, Delete
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
SSR	Server Side Rendering
SSG	Static Site Generation
SEO	Search Engine Optimization
OTA	Over The Air
CI/CD	Continuous Integration/Continuous Deployment
tRPC	Typed RPC
RPC	Remote Procedure Call
TLS	Transport Layer Security

TABLE 1 – Abréviation ou sigle et significations

Introduction

Au cœur de l'ère numérique actuelle, l'humanité fait face à un défi majeur : exploiter la puissance des technologies modernes pour transformer la création et le partage des arbres généalogiques. À l'intersection de la généalogie et du développement de logiciels, ce défi nous invite à repenser la manière dont nous conservons et transmettons notre héritage familial. Intégrer les innovations numériques est essentiel pour explorer de nouvelles perspectives, facilitant ainsi la reconstitution et la diffusion des arbres généalogiques, rendant ce processus plus accessible et interactif pour tous.

Dans le cadre de notre projet pour l'obtention de la licence professionnelle en filière informatique de gestion, option génie logiciel, à l'École Supérieure de Gestion et d'Administration des Entreprises (ESGAE), nous nous sommes engagés à analyser et développer une plateforme visant à apporter une réponse à ce défi. C'est dans ce contexte que ce travail a été orienté sur le thème : « **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 répond 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 :

- Simplifier la construction participative d'arbres généalogiques : nous souhaitons fournir une plateforme conviviale où les membres de la famille peuvent coopérer efficacement pour bâtir et partager leurs arbres généalogiques.
- Faciliter le partage des informations généalogiques : nous désirons créer une interface intuitive sur les plateformes web et mobiles, permettant aux utilisateurs de partager facilement leurs arbres généalogiques avec leurs proches.
- Promouvoir la découverte de l'histoire et de l'héritage familial : notre plateforme est destinée à tous ceux qui aspirent à 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 démarche méthodologique rigoureuse en utilisant la méthode objet.

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

Première partie

Présentation Générale de Mazala-Firm

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. Créé en 2021 par Inefable KOUMBA, un passionné des nouvelles technologies, Mazala-Firm s'est rapidement établi 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'applications 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 :

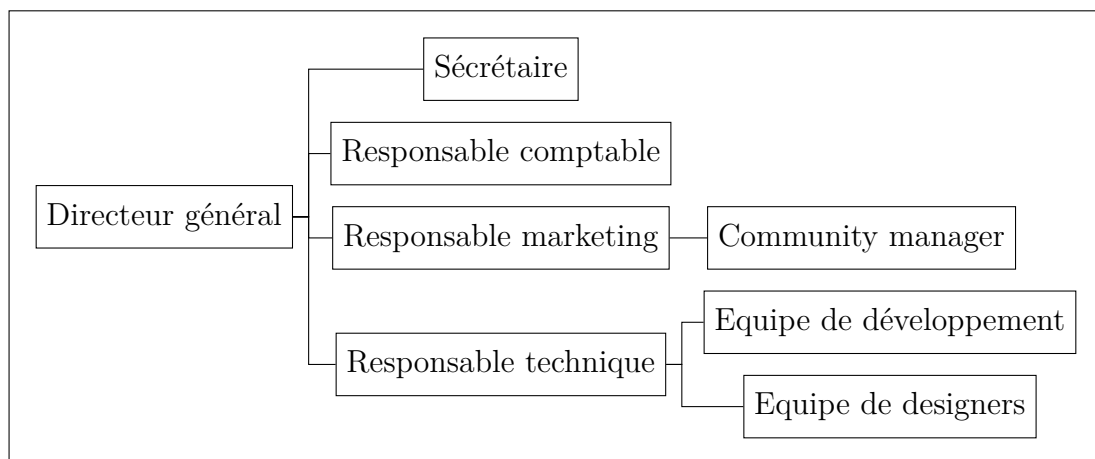


FIGURE 1.1 – Organigramme de Mazala-Firm

1.1.4 Attributions des structures

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

bilités :

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, mets 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, interagis avec les clients et crée du contenu pour engager la communauté.
7. **Équipe de développement** : conçoit, développe et maintiens 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, dotés 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 (Ultra HD) ou 3840x2160 (4K Ultra HD) 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 gigabits, et des points d'accès Wi-Fi couvrant l'ensemble du bâtiment pour assurer une connexion réseau fiable et sécurisée.
- Accessoires comme les claviers ergonomiques et souris de précision de marque HP, casques audio, 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 efficaces 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

Étude 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 Two Tracks Unified Process (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 Étude 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.

Méthodes Critères	Cartésiennes	Systémiques	Agiles	Objet	Formelles
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 [3], 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 sym-

boles 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.

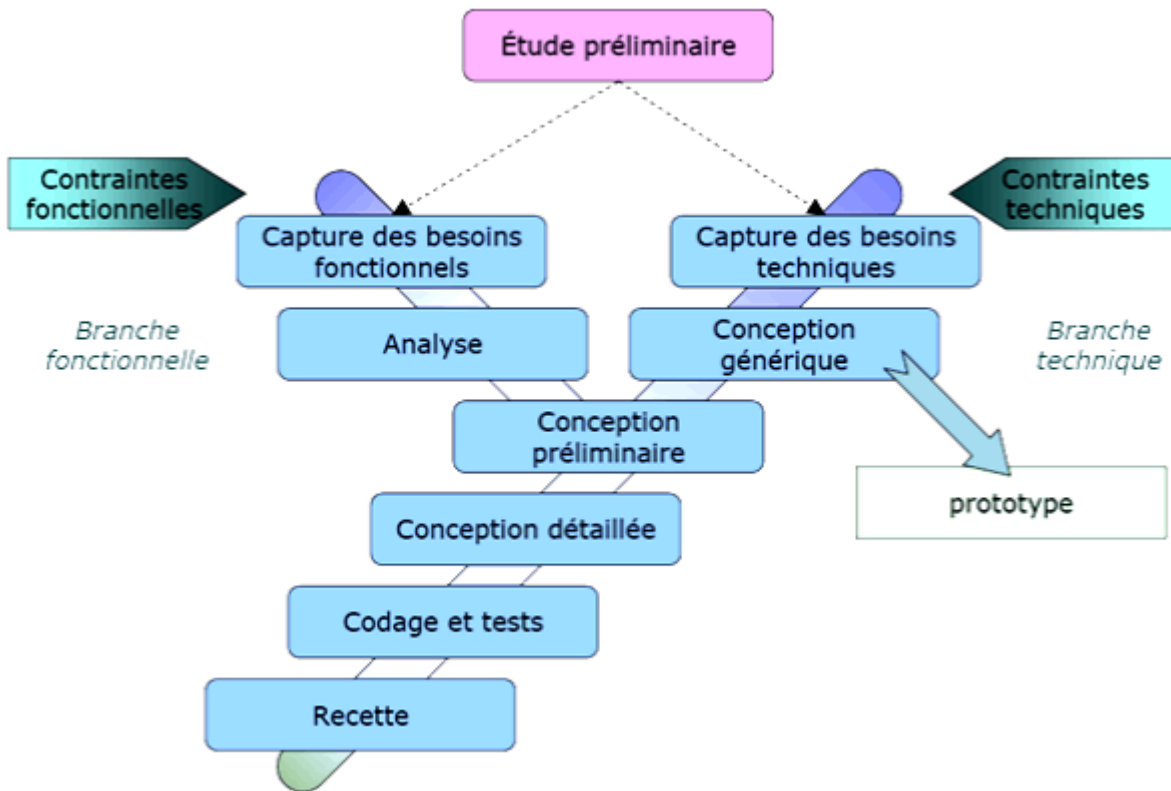


FIGURE 1.1 – Illustration du processus de développement en Y

1.2.2.3 2TUP et UML

L'utilisation de 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. 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 faire évoluer 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 créateurs d'arbre ;
- 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.

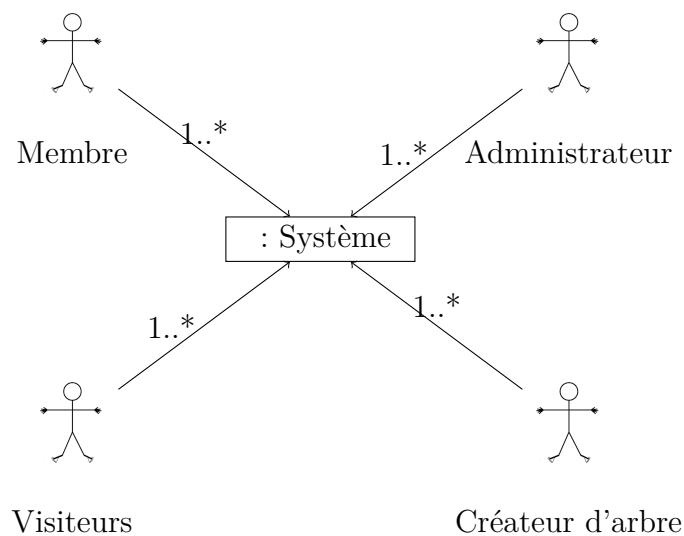


FIGURE 2.1 – Diagramme de contexte statique

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 :

- Créer un compte
- S'authentifier
- Gérer un arbre généalogique ;
- Gérer des membres ;
- Consulter un arbre généalogique ;
- Rechercher des membres de la famille ;
- Accorder des droits d'accès ;
- Partager un arbre généalogique ;
- 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. De même, le cas d’utilisation “S’authentifier” est inclus dans tous les cas d’utilisation des membres et de l’administrateur, car ils doivent être authentifiés avant de réaliser toute autre action.
- 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.

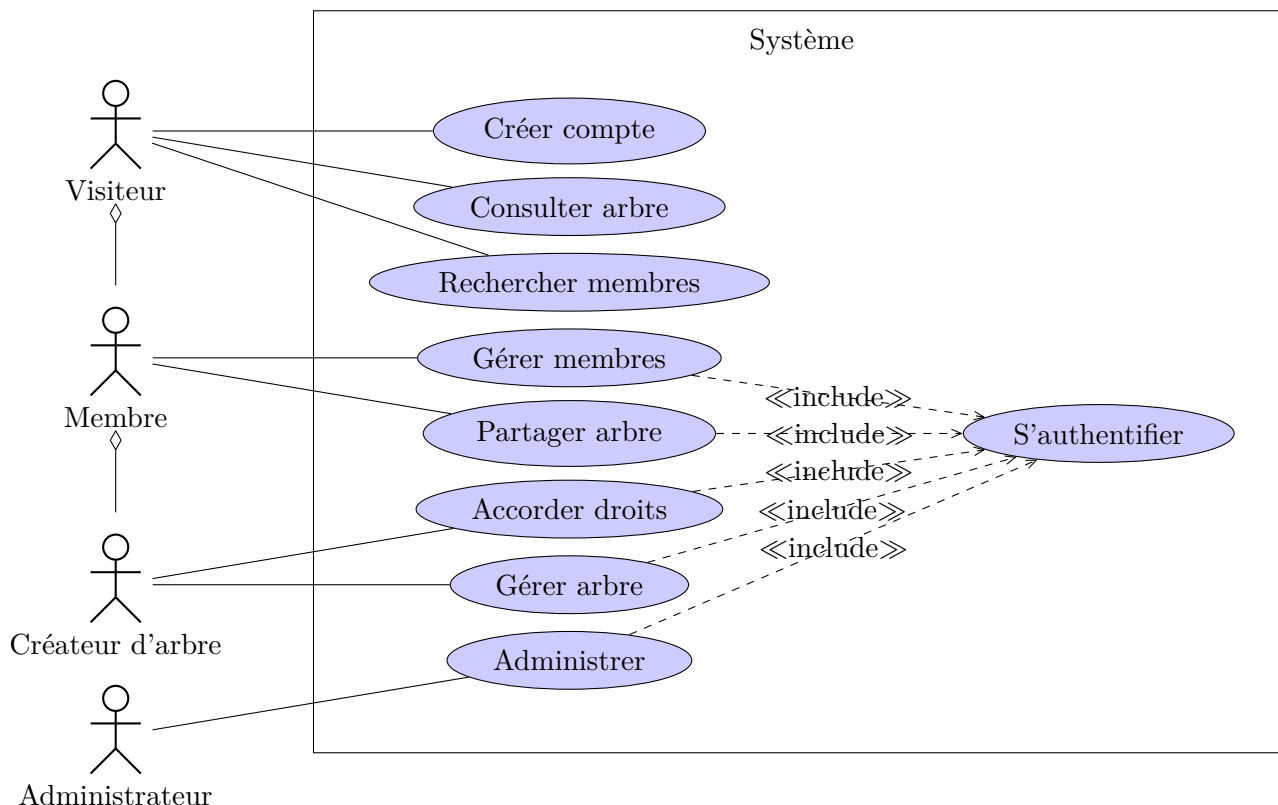


FIGURE 2.2 – Diagramme de cas d'utilisation

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
S'authentifier	Forte	Moyen	1
Gérer un compte	Moyenne	Moyen	1
Gérer un arbre généalogique	Forte	Moyen	1
Gérer des membres	Forte	Moyen	1
Consulter un arbre généalogique	Forte	Faible	1
Rechercher des membres de la famille	Moyenne	Moyen	2
Partager un arbre généalogique	Forte	Élevé	2
Accorder des droits d'accès	Moyenne	Élevé	3
Administrer le système	Moyenne	Élevé	3

TABLE 2.1 – Classification des cas d'utilisation

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 : Membre

Autres acteurs : Système

Date : 15/04/2024

Version : 2.0

Description : Ce cas d’utilisation permet à le membre de créer, de modifier, de visualiser et de supprimer des arbres généalogiques.

Préconditions : Le membre est authentifié et peut accéder à la fonctionnalité de gestion des arbres généalogiques.

Séquencement des événements

Le cas d’utilisation commence lorsque le membre se connecte à la plateforme et desire gérer un arbre généalogique.

Scénario nominal :

1. Le membre 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 ce membre.
3. Le membre 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 le membre décide de créer un nouvel arbre généalogique : (A1)
 - (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(public, privé) pour son nouvel arbre généalogique ;
 - (c) le membre valide ses informations ;
 - (d) le système crée alors un nouvel arbre généalogique et l’affiche dans la liste des arbres généalogiques du membre.
5. Si le membre décide de modifier un arbre généalogique existant : (A2)
 - (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) le membre effectue les modifications nécessaires (ajouts, suppressions, modifications de membres ou de relations) ou les informations de l’arbre généalogique ;

Pour ajouter un membre :

 - le membre 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.) ;
 - le membre 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 :

- le membre clique sur l'option “Modifier les informations” ;
 - le système affiche un formulaire de modification des informations de l'arbre généalogique ;
 - le membre 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 le membre souhaite consulter, un arbre généalogique existant : (A3)
- (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 le membre souhaite supprimer un arbre généalogique existant : (A4)
- (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 le membre accepte, le système efface l'arbre généalogique ainsi que toutes les données connexes de la liste dans la barre de navigation latérale.

Scénarios alternatifs :

1. **A1**

- **Étape 4a** : Le membre 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 ;
- Le scénario reprend à l'étape 4c.

2. **A2**

- **Étape 5a** : Le membre laisse des champs obligatoires vides.
 - le système affiche un message d'erreur demandant de compléter les champs obligatoires ;
- Le scénario reprend à l'étape 5c.

3. **A3**

- **Étape 6a** : Le membre 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.
 - le membre corrige les erreurs.
- Le scénario reprend à l'étape 6a.

4. **A4**

- **Étape 7a** : Le membre 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.

Scénario d'exception :

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

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

Cette description textuelle fournit une vue détaillée des étapes et des interactions impliquées dans le cas d'utilisation "Gérer un arbre généalogique".

Cas d'utilisation : Accorder des droits d'accès

Acteur : Créateur d'arbre

Autres acteurs : Système

Date : 15/04/2024

Version : 1.3

Description : Ce cas d'utilisation permet au créateur d'arbre d'accorder, de modifier, et de révoquer des droits d'accès aux membres pour les arbres généalogiques.

Préconditions : Le créateur d'arbre est authentifié et possède les droits nécessaires pour gérer les accès aux arbres généalogiques.

Séquencement des événements

Le cas d'utilisation commence lorsque le créateur d'arbre se connecte à la plateforme et désire gérer les droits d'accès à un arbre généalogique.

Scénario nominal :

1. Le créateur d'arbre se connecte à l'application.
2. Le système affiche une barre de navigation latérale comportant un bouton "Gérer les droits d'accès" ainsi qu'une liste d'arbres généalogiques pour lesquels il a des droits de gestion.
3. Le créateur d'arbre sélectionne l'arbre généalogique pour lequel il souhaite gérer les droits d'accès.
4. Le système affiche une interface permettant de voir les membres actuels et leurs niveaux d'accès.
5. Le créateur d'arbre peut ensuite sélectionner parmi les options suivantes :
 - Ajouter des droits d'accès pour un nouveau membre ;
 - Modifier les droits d'accès pour un membre existant ;
 - Révoquer les droits d'accès pour un membre existant.
6. Si le créateur d'arbre décide d'ajouter des droits d'accès pour un nouveau membre : (A1)
 - (a) il doit cliquer sur le bouton "Ajouter un membre" ;
 - (b) le système affiche une fenêtre où il devra entrer l'identifiant du membre et choisir le niveau d'accès (lecture seule, modification, gestion complète) ;
 - (c) le créateur d'arbre valide ses informations ;
 - (d) le système ajoute le membre avec les droits spécifiés et met à jour la liste des membres ayant accès à l'arbre généalogique.
7. Si le créateur d'arbre décide de modifier les droits d'accès d'un membre existant : (A2)
 - (a) il sélectionne le membre dans la liste ;
 - (b) le système affiche alors les droits actuels du membre sélectionné ;
 - (c) membre modifie les droits d'accès nécessaires ;
 - (d) le système enregistre les modifications apportées aux droits d'accès.
8. Si le créateur d'arbre souhaite révoquer les droits d'accès d'un membre existant : (A3)
 - (a) il sélectionne le membre dans la liste ;
 - (b) le système demande une confirmation avant de procéder à la révocation des droits d'accès ;

- (c) si le créateur d’arbre confirme, le système révoque les droits d’accès du membre sélectionné.

Scénario alternatif :

1. **A1**

- **Étape 6a :** Le créateur d’arbre tente d’ajouter un membre mais ne saisit pas un identifiant valide ou ne choisit pas un niveau d’accès.
 - le système affiche un message d’erreur demandant de saisir un identifiant valide ou de choisir un niveau d’accès ;Le scénario reprend à l’étape 6c.

2. **A2**

- **Étape 7a :** Le créateur d’arbre tente de modifier les droits d’accès mais laisse des champs obligatoires vides.
 - le système affiche un message d’erreur demandant de compléter les champs obligatoires ;Le scénario reprend à l’étape 7c.

3. **A3**

- **Étape 8a :** Le créateur d’arbre sélectionne un membre à révoquer mais change d’avis et annule l’opération.
 - Le système annule l’opération de révocation et revient à la liste des membres ayant accès à l’arbre généalogique.

Scénario d’exception :

1. Si, à l’étape 3, le créateur d’arbre ne sélectionne pas un arbre généalogique valide, le système affiche un message d’erreur et ne permet pas de gérer les droits d’accès.
2. Si le créateur d’arbre essaie de révoquer les droits d’accès d’un membre critique sans confirmation, le système annule l’opération.

Postconditions : Les modifications des droits d’accès sont enregistrées et mises en vigueur dans le système.

Cette description textuelle fournit une vue détaillée des étapes et des interactions impliquées dans le cas d’utilisation “Accorder des droits d’accès”.

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 membre.

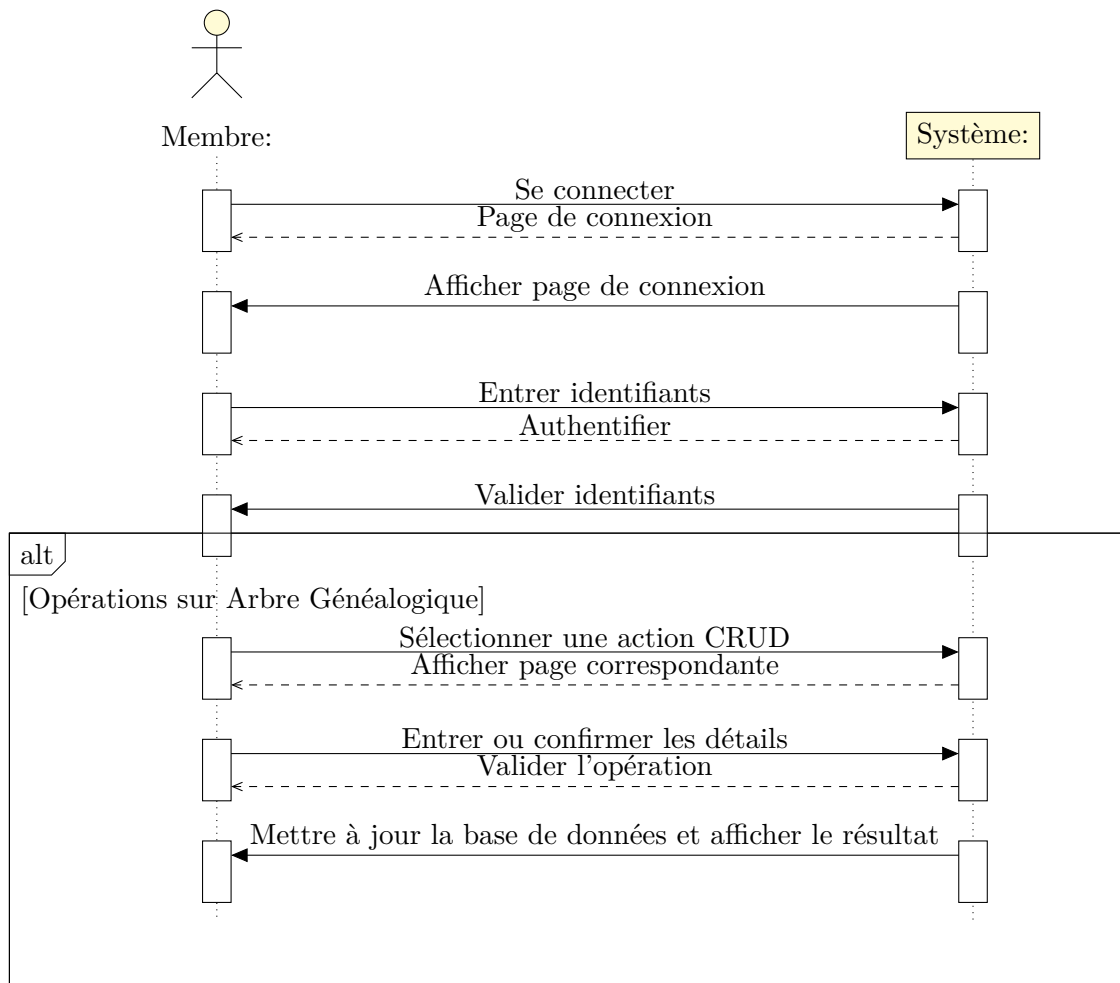


FIGURE 2.3 – Diagramme de séquence du cas d'utilisation gérer un arbre généalogique

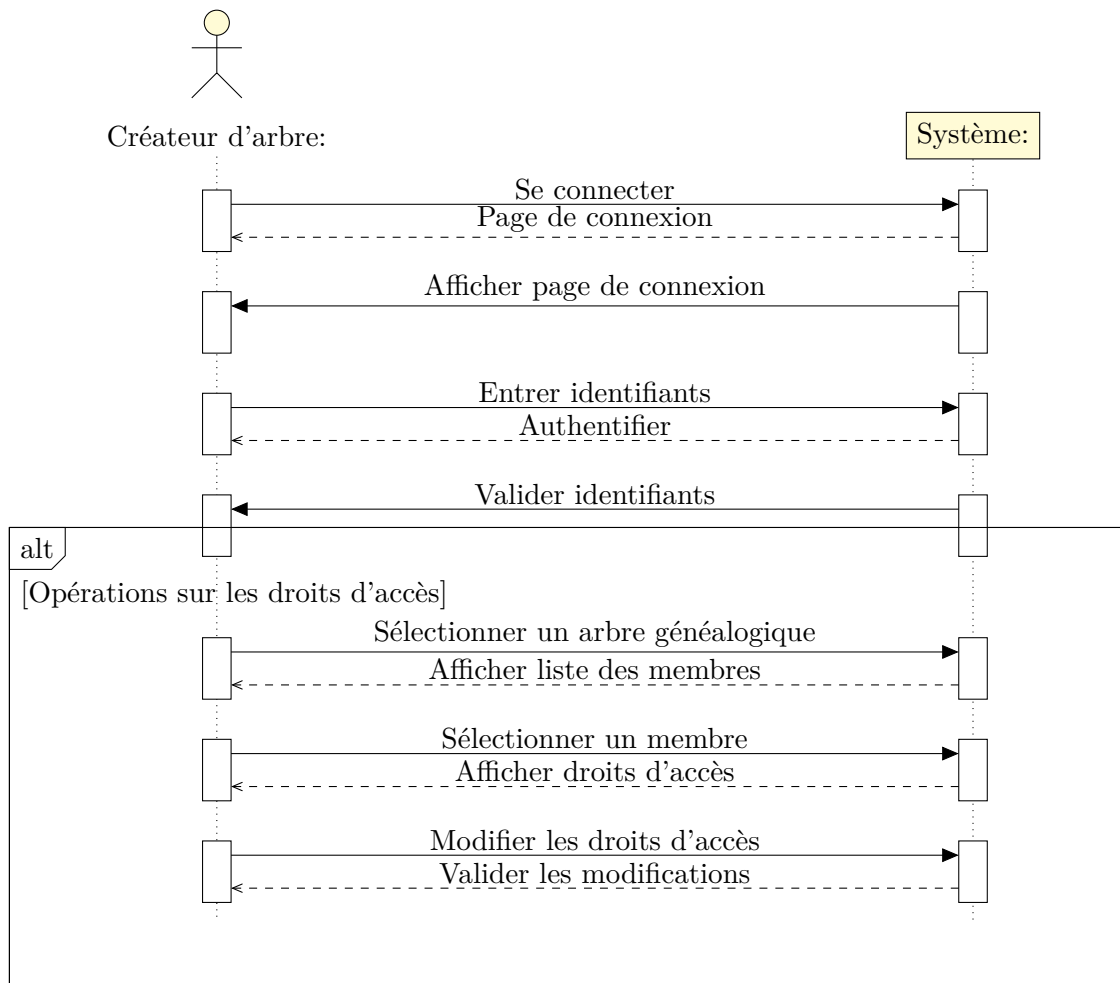


FIGURE 2.4 – Diagramme de séquence du cas d'utilisation accorder des droits d'accès

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.

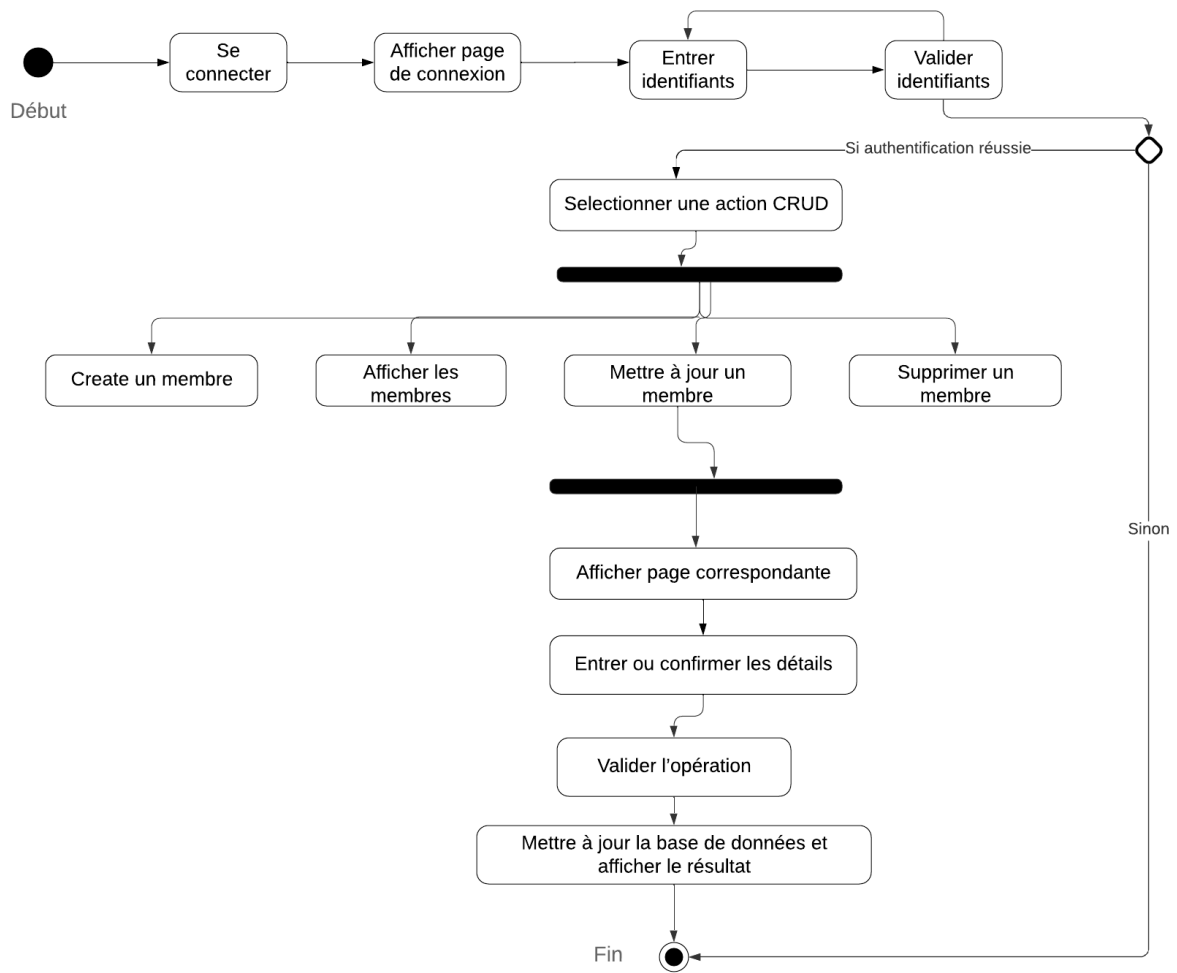


FIGURE 2.5 – Diagramme d'activité du cas d'utilisation gérer un arbre généalogique

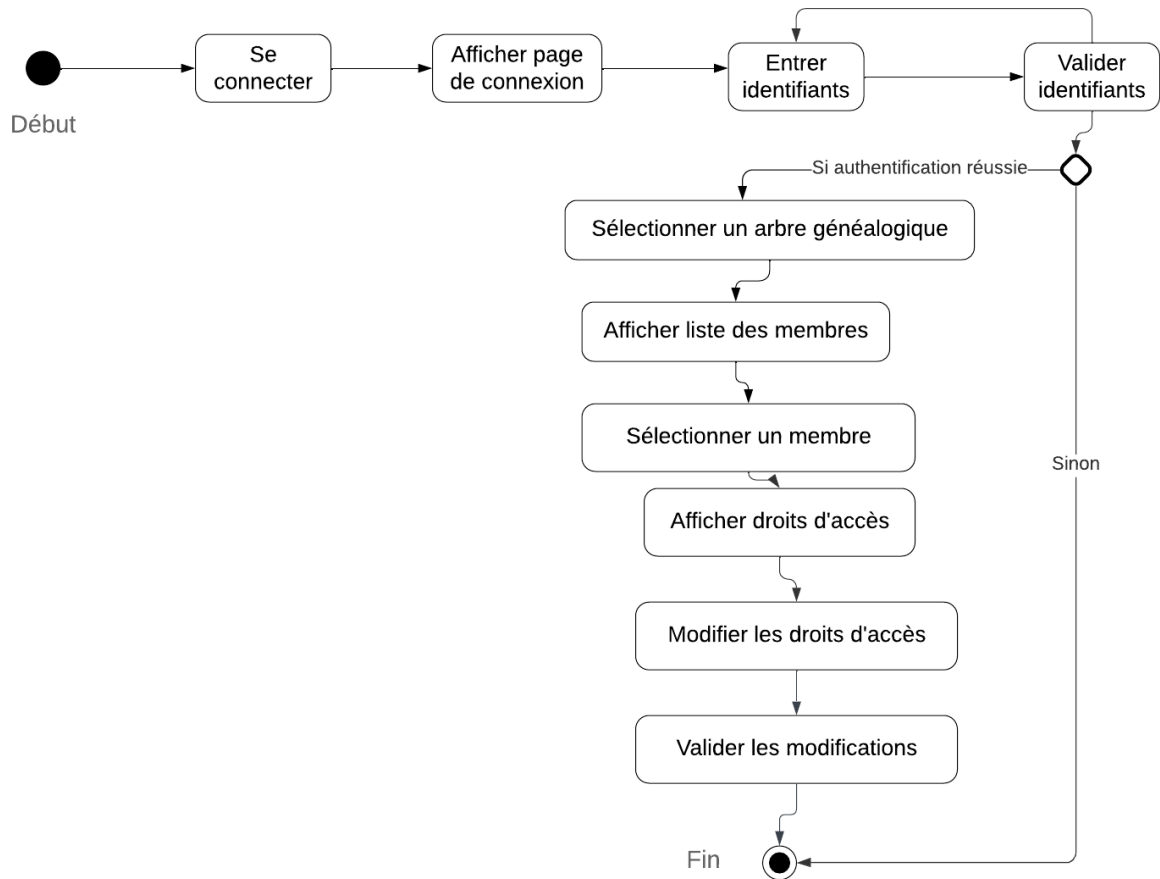


FIGURE 2.6 – Diagramme d'activité du cas d'utilisation accorder des droits d'accès

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 membre ne peut pas modifier un arbre généalogique qui ne lui appartient pas.
- **Règle de gestion 2** : un arbre généalogique privé ne peut être consulté que par ceux ayant des droits.
- **Règle de gestion 3** : un arbre généalogique public peut être consulté par tous les utilisateurs.
- **Règle de gestion 4** : un arbre généalogique ne peut être supprimé que par un membre ayant le droit d'administration.

- **Règle de gestion 5** : un membre ne peut pas être ajouté deux fois à un arbre généalogique.

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.

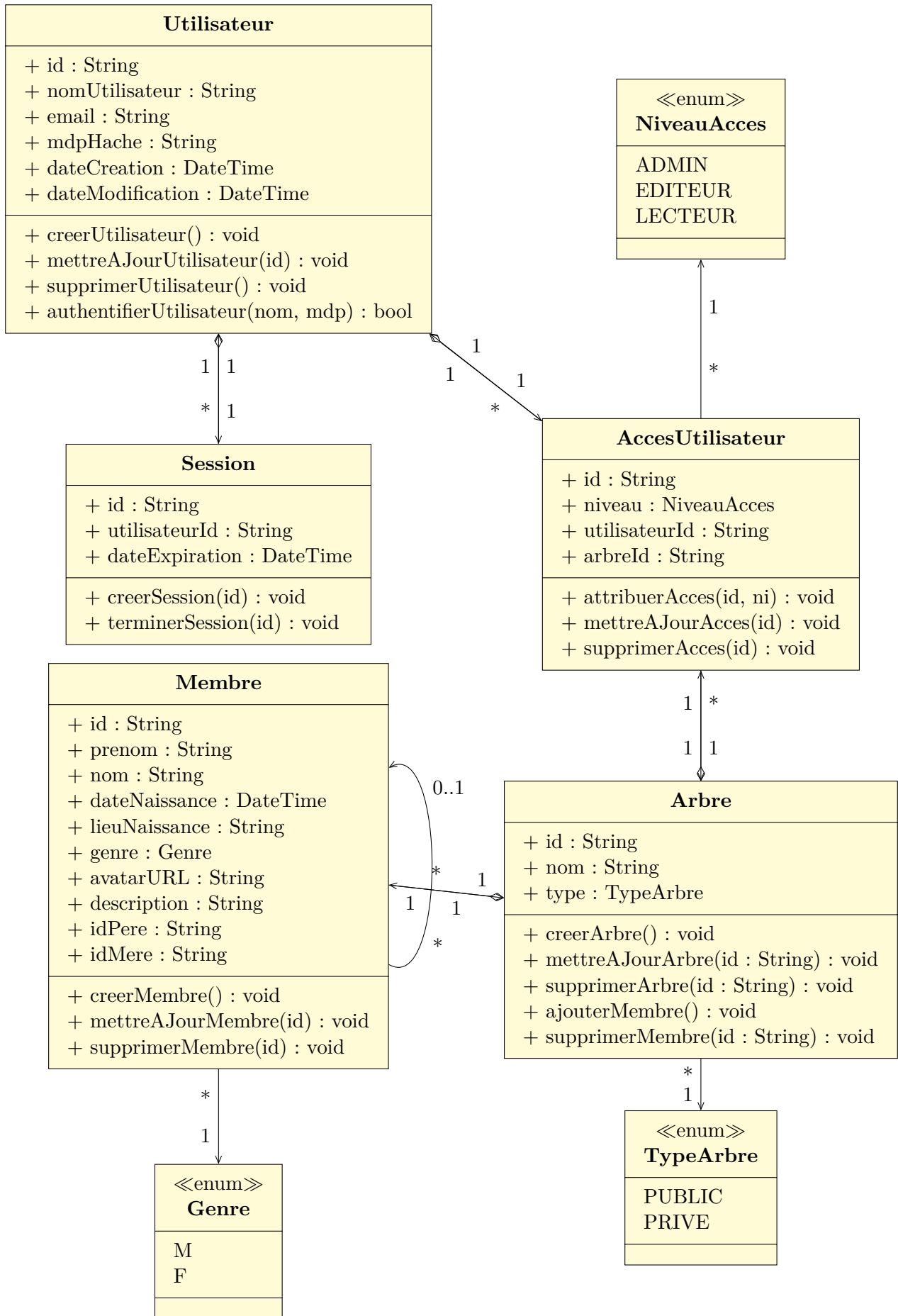


FIGURE 2.7 – Diagramme de classe du système

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 2.8 – 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 projets dans un seul référentiel (Monorepo). Cette architecture permet de centraliser le code des applications Android, iOS, et Web, ainsi que les composants partagés et le serveur backend.

Structure du Répertoire

Racine du Projet

- `.env` : Fichier de configuration des variables d'environnement.
- `package.json` : Dépendances et scripts du projet.
- `tsconfig.json` : Configuration de TypeScript.
- `next.config.js` : Configuration de Next.js.
- `capacitor.config.ts` : Configuration de Capacitor.

android

- Contient le code et la configuration de l'application Android utilisant Capacitor.

ios

- Contient le code et la configuration de l'application iOS utilisant Capacitor.

out

- Contient les exportations statiques de l'application Web Next.js.

src

- `actions` : Actions server Next.js.
- `app` : Code principal de l'application Next.js.
- `components` : Composants React réutilisables.
- `lib` : Bibliothèques et fonctions utilitaires.
- `server` : Code backend, incluant les routes API et la configuration de la base de données.
- `styles` : Fichiers de styles.
- `trpc` : Configuration et handlers de tRPC.
- `types` : Types et interfaces TypeScript.

Interaction entre les Parties

- **Android et iOS** : Utilisent Capacitor pour intégrer des fonctionnalités Web dans les applications mobiles. Ils partagent du code et des composants avec l'application Web.
- **Web (Next.js)** : L'application principale qui utilise React pour le frontend et tRPC pour les appels API.
- **Server** : Fournit les API et gère les interactions avec la base de données. Partagé entre les applications Web et mobiles pour assurer la cohérence des données.
- **Components et Lib** : Répertoires partagés contenant les composants réutilisables et les fonctions utilitaires, facilitant la maintenance et la réutilisation du code.

Cette structure Monorepo centralisée permet une gestion cohérente et efficace des différents aspects de l'application, en simplifiant le développement et la maintenance.

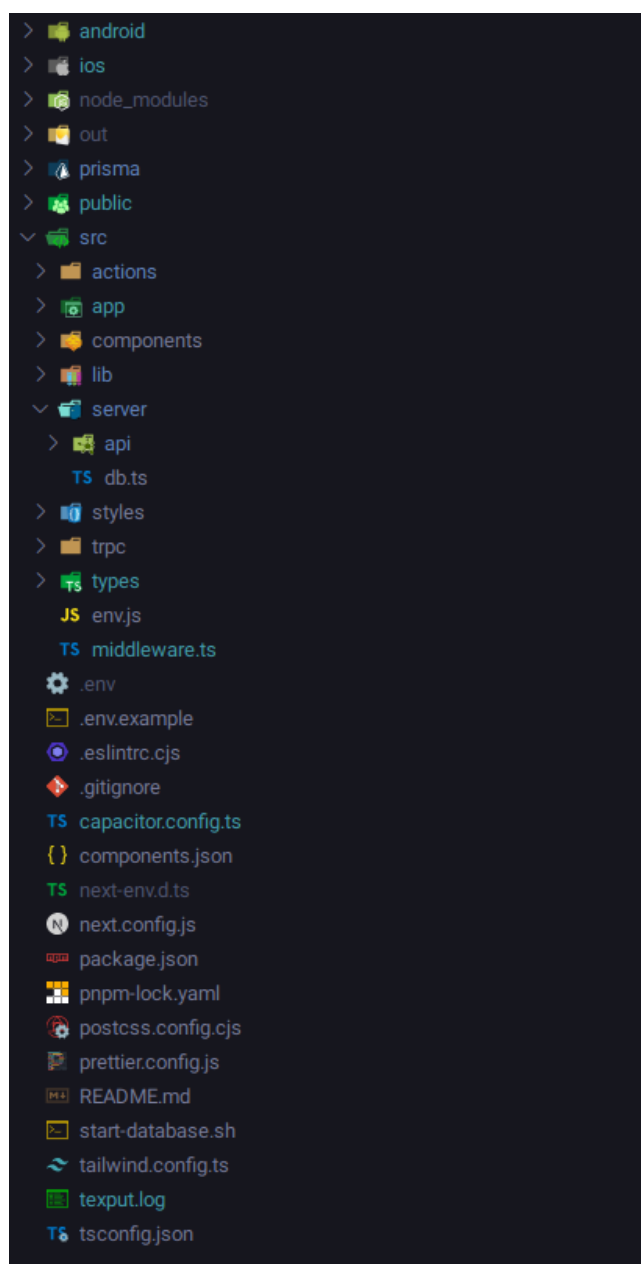


FIGURE 2.9 – Architecture du logiciel sous VSCodium

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).

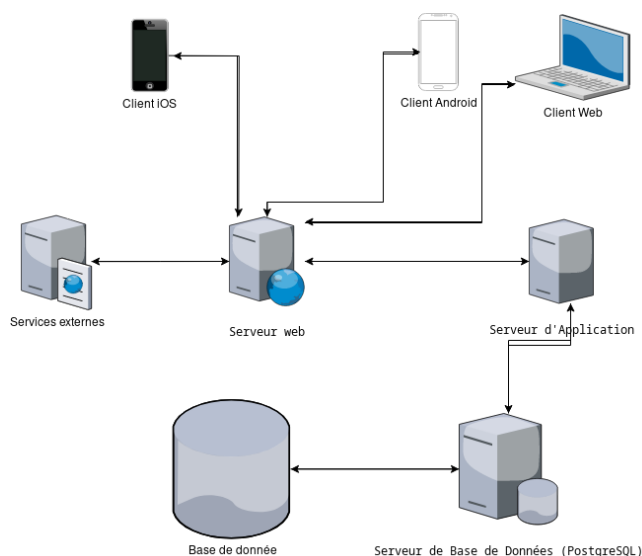


FIGURE 2.10 – Illustration d'une architecture distribuée

- **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 d'application** : utilise Node.js et Next.js pour le rendu côté serveur et la génération de pages statiques.
- **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 Capacitor pour l'accès mobile.
- **Services externes** : intègre éventuellement des services externes pour des fonctionnalités supplémentaires, telles que le stockage de fichiers et les services de messagerie.

Voici un diagramme de déploiement illustrant cette architecture :

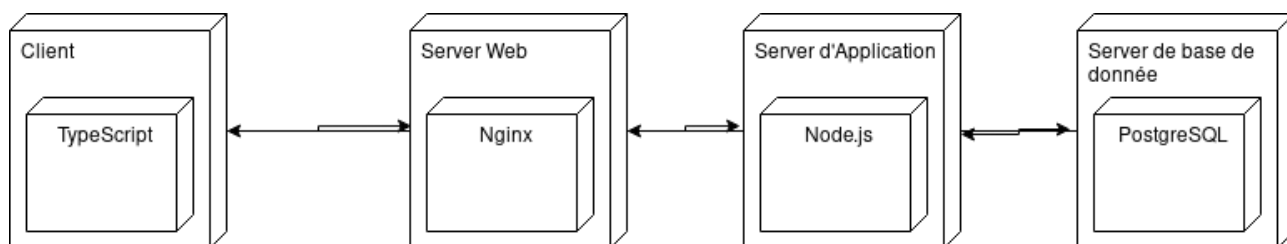


FIGURE 2.11 – Diagramme de déploiement logique

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

Évaluation et Réalisation

Chapitre 1

Évaluation 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.

Processus	Phases
P1 Arbre	Début du projet
	Documentation
	Analyse et conception
	Réalisation
P2 Droit	Documentation
	Analyse et conception
	Réalisation
Fin du projet	

TABLE 1.1 – Organisation du projet

1.2 Intervenant

Intervenants	Fonctions	Rôles
M. Christopher BANDZOUZI	Ingénieur Informaticien	Directeur de projet
M. Samuel Exaucé NANDI	Étudiant	Réalisateur
M. Dieu-Veille Frédy ONIANGUE-DESO	Étudiant	Réalisateur
Mme Rovélia MOUNTOU	Sécrétaire Mazala-Firm	Tutrice de stage

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
P1 Arbre	Début du projet	
	Analyse et conception	Étude de l'existant et Critiques Analyse et modélisation La conception
	Réalisation	Implémentation d'interfaces
	Documentation	Production du document et du PowerPoint Cahier de charge Documentation sur le projet
P2 Droit	Analyse et conception	Étude de l'existant et Critiques Analyse et modélisation La conception
	Réalisation	Implémentation d'interfaces Cahier de charge
	Documentation	Production du document et du PowerPoint Documentation sur le projet
		Fin du projet

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.

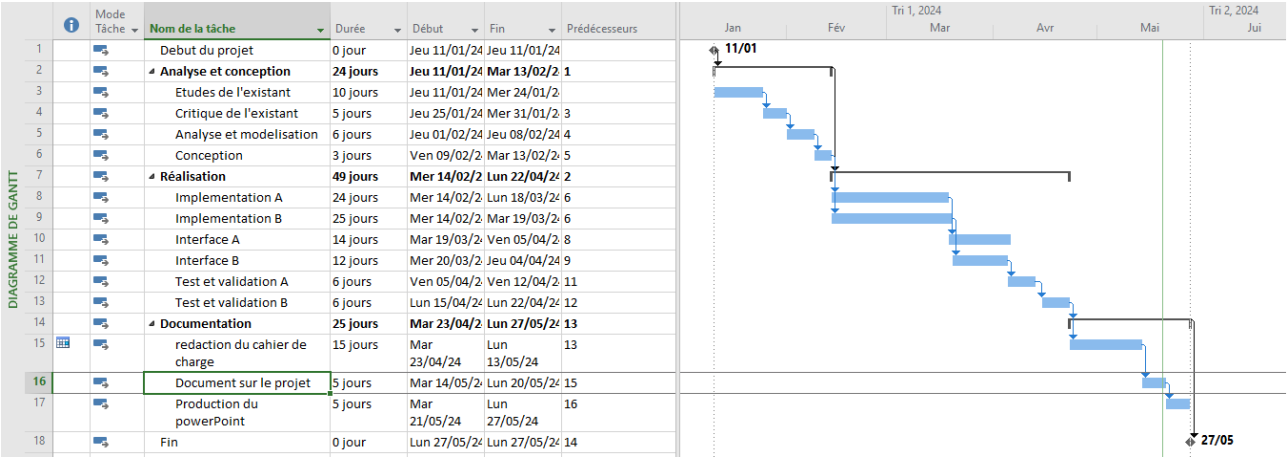


FIGURE 1.1 – Diagramme de Gantt

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	Quantité	Prix
L ^A T _E X	1	0 FCFA
TypeScript	1	0 FCFA
Next.js	1	0 FCFA
Capacitor	1	0 FCFA
PostgreSQL	1	0 FCFA
Docker	1	0 FCFA
Git	1	0 FCFA
GitHub	2	0 FCFA
Neovim	1	0 FCFA
VSCodium	1	0 FCFA
Figma	1	0 FCFA
Microsoft Project	6	217 506CFA
PC Portable	2	800 000 FCFA
Connexion internet	6	150 000 FCFA
Réalisateurs	2	4 000 000 FCFA
Total		5 167 506 FCFA

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.
- **Capacitor** Capacitor crée un environnement de développement pour les applications en transformant les applications web en applications mobiles natives avec JavaScript, HTML et CSS.

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 un système de session lié à la base de donnée pour l'authentification. Les autorisations, pour sécuriser les accès à l'application et aux API sont fait par rapport à ces sessions.

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 plugins, 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.

— Capacitor

Capacitor est un framework qui permet de créer des applications mobiles multiplateformes avec des technologies web comme React, Vue et bien d'autres. Il offre des fonctionnalités telles que l'accès aux API natives, la gestion des plugins, et la compatibilité avec les stores d'applications.



FIGURE 2.3 – Logo Capacitor

— 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

— **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 2.5 – 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.

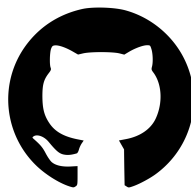


FIGURE 2.6 – Logo GitHub

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.

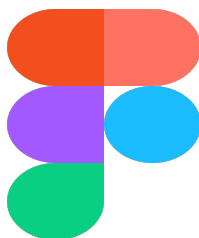


FIGURE 2.7 – Logo Figma

— **L^AT_EX**

L^AT_EX est un système de composition de documents utilisé pour rédiger les documentations techniques et les rapports. Il offre une mise en page professionnelle et une gestion

avancée des références. Il permet également de créer des diagrammes UML et de toute sorte.



FIGURE 2.8 – Logo L^AT_EX

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.

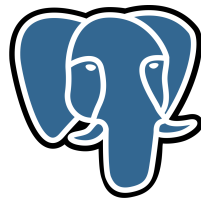


FIGURE 2.9 – Logo PostgreSQL

— 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.

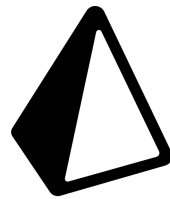


FIGURE 2.10 – Logo Prisma

— 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 2.11 – Logo tRPC

Outils de Sécurité

— 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 2.12 – 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 2.13 – Logo Vercel

— AppFlow

AppFlow est utilisé pour le déploiement et la gestion des applications mobiles développées avec Capacitor, React Native et bien d'autre, en facilitant le processus de publication sur les stores d'applications.



FIGURE 2.14 – Logo AppFlow

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.

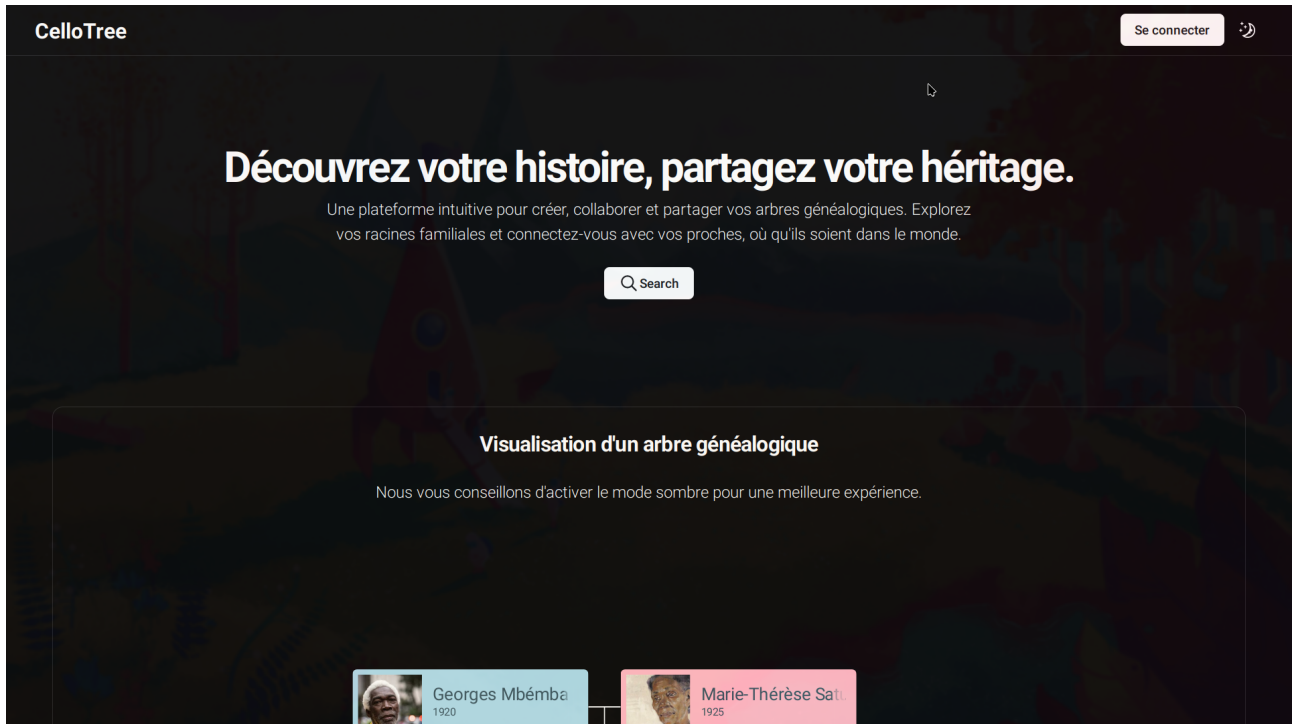


FIGURE 2.15 – Écran d’accueil sur l’application web

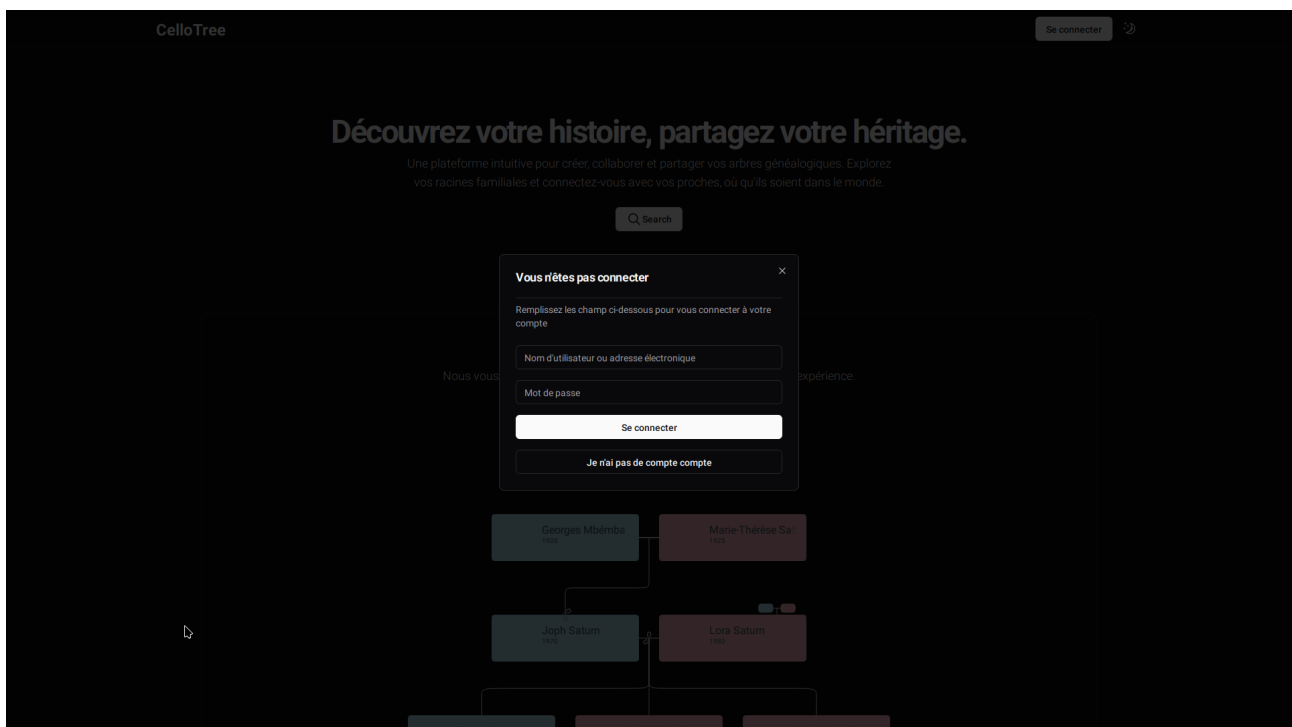


FIGURE 2.16 – Écran de connexion sur l’application web

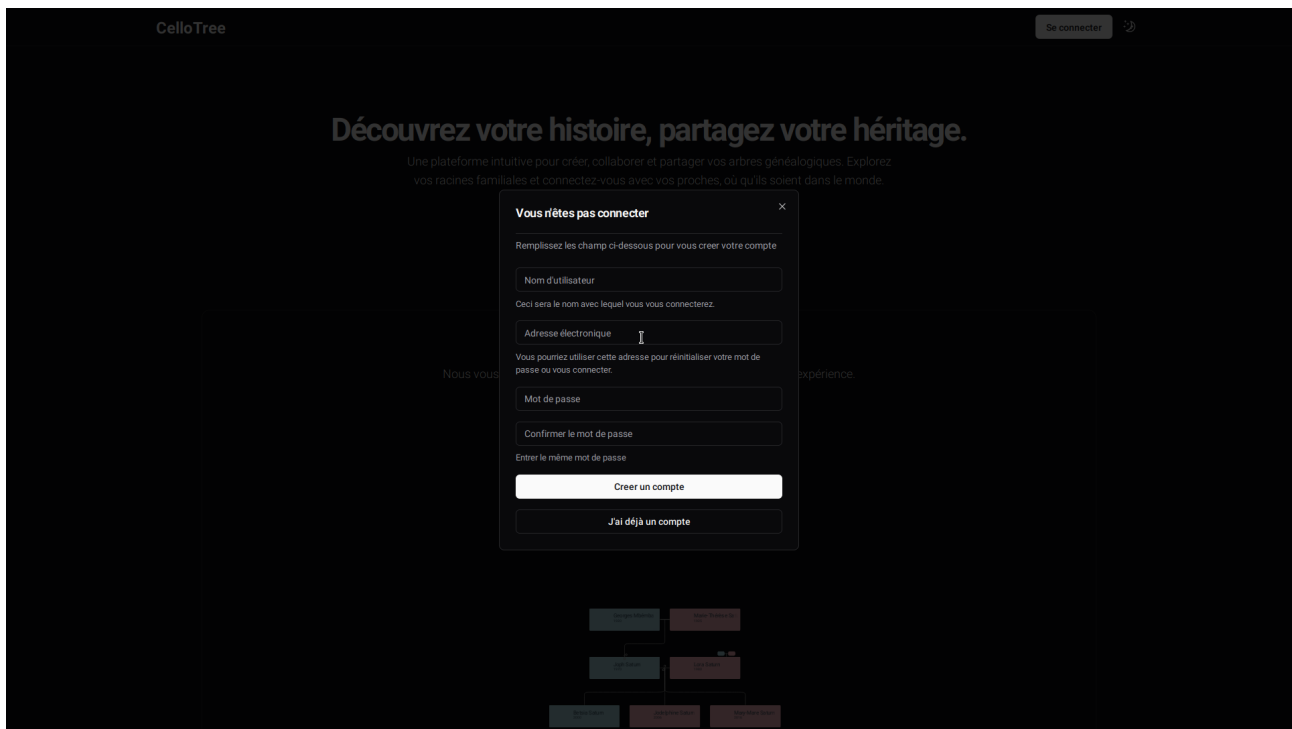


FIGURE 2.17 – Écran de création de comptes sur l'application web

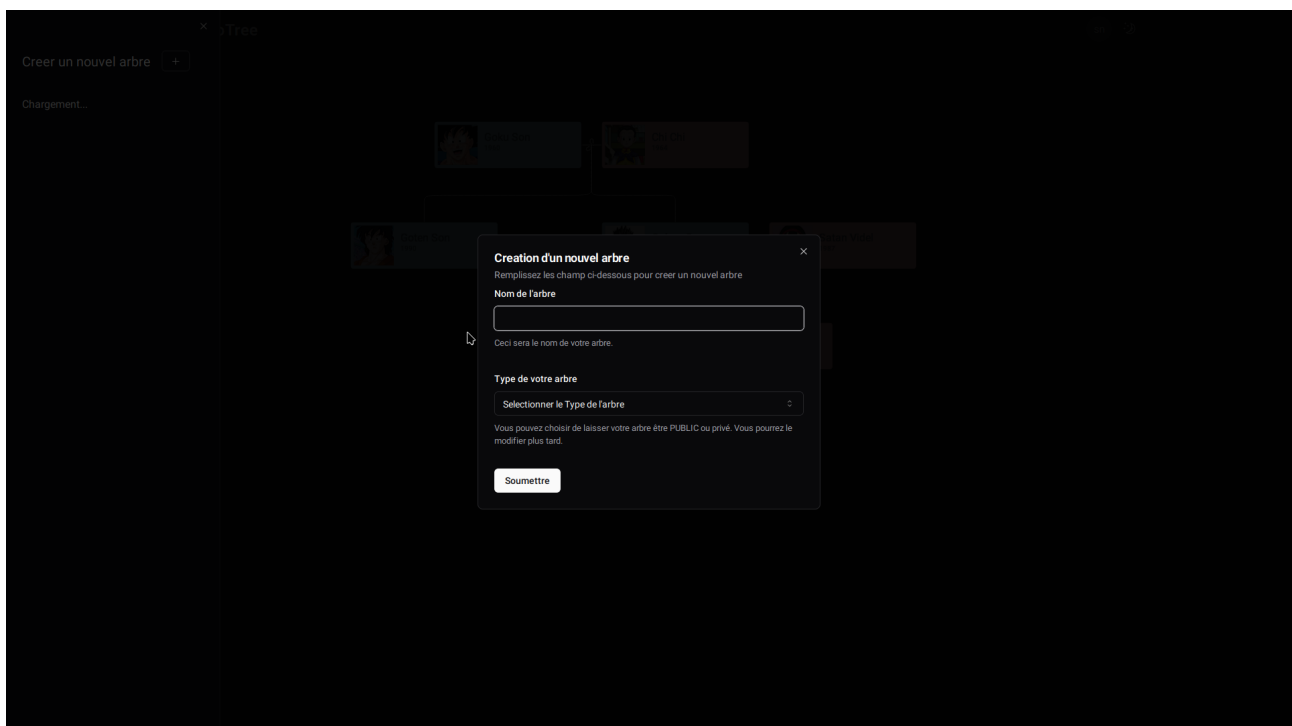


FIGURE 2.18 – Écran de création d'un arbre généalogique sur l'application web

The screenshot shows a web application interface for CelloTree. A modal form titled "Ajout d'un nouveau membre à l'arbre" is centered on the screen. The form contains several input fields: "Nom" (Name), "Prénoms" (First names), "Date de naissance" (Date of birth) with a date picker, and "Lieu de naissance" (Place of birth). Below these are sections for "Sex" (with a dropdown menu) and "Avatar" (with a "Browse..." button). A "Description" field is also present. A "Soumettre" (Submit) button is at the bottom right of the form. The background shows a blurred view of a family tree and a sidebar with filters.

FIGURE 2.19 – Écran d'ajout d'un membre à un généalogique sur l'application web

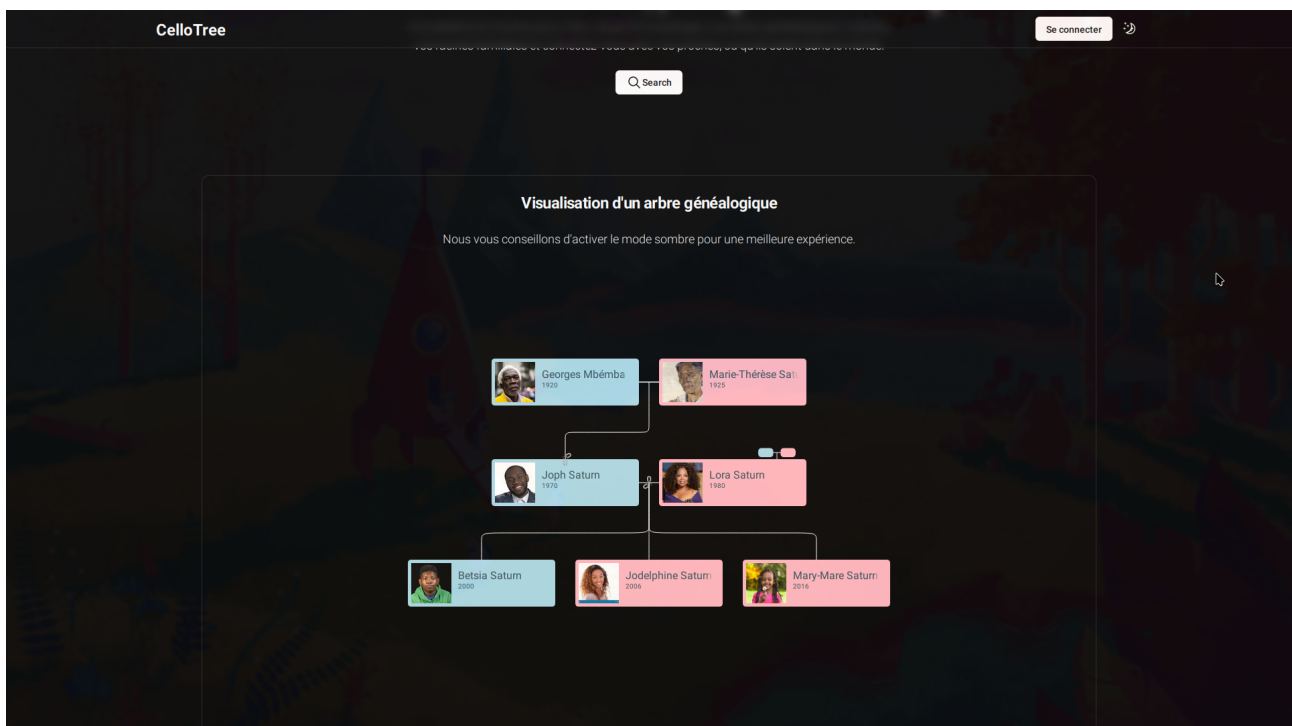


FIGURE 2.20 – Écran de visualisation d'un arbre généalogique

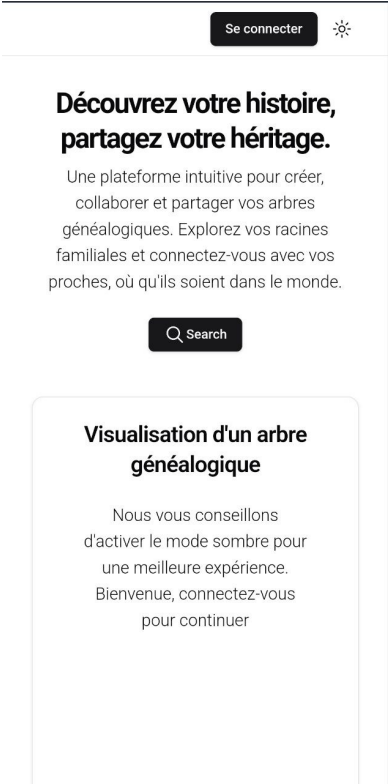


FIGURE 2.21 – Écran d’accueil sur l’application mobile

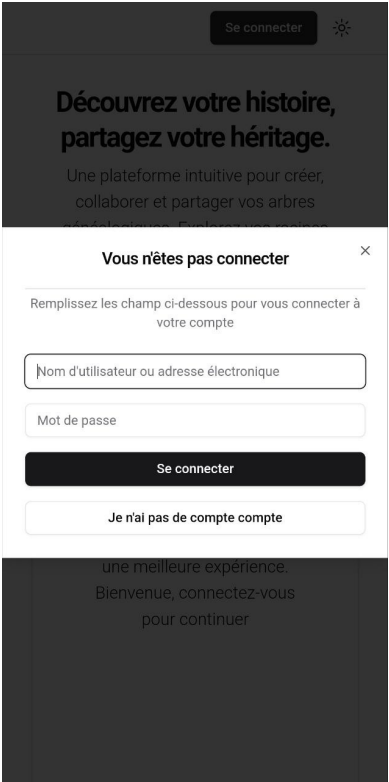


FIGURE 2.22 – Écran de connexion sur l’application mobile

The screenshot shows a mobile application interface for creating a new family tree. At the top, the status bar displays the time 00:05, signal strength, and battery level at 80%. The app title is "Creation d'un nouvel arbre" with a close button (X). Below the title, a subtitle reads "Remplissez les champ ci-dessous pour creer un nouvel arbre". The form includes a text input field for "Nom de l'arbre" with a placeholder "Ceci sera le nom de votre arbre." Below this is a dropdown menu for "Type de votre arbre" with the text "Selectionner le Type de l'arbre". A note states: "Vous pouvez choisir de laisser votre arbre être PUBLIC ou privé. Vous pourrez le modifier plus tard." At the bottom of the form is a "Soumettre" button. A virtual keyboard is visible at the bottom of the screen.

FIGURE 2.23 – Écran de création d'un arbre généalogique sur l'application mobile

The screenshot shows a mobile application interface for adding a new member to a family tree. At the top, the status bar displays the time 00:39, signal strength, and battery level at 80%. The app title is "Ajout d'un nouveau membre à l'arbre" with a close button (X). Below the title, a subtitle reads "Remplissez les champ ci-dessous pour ajouter un nouveau membre." The form includes a text input field for "Nom", a text input field for "Prénoms", a date picker for "Date de naissance" with the text "Choisissez une date", a text input field for "Lieu de naissance", a dropdown menu for "Sex" with the text "Selectionner le sexe du membre", a note stating "Nous ne concevons pas qu'il est d'autres sex ni genre c'est de la bêtises et nous ne l'encourageons pas.", a file picker for "Avatar" with the text "Choose file No file chosen", a text input field for "Description" with the placeholder "Dites nous un peu plus sur ce membre".

FIGURE 2.24 – Écran d'ajout d'un membre à un généalogique sur l'application mobile

Conclusion

En somme, notre projet de création d'une plateforme web et mobile dédiée à la généalogie collaborative répond à un besoin croissant de préservation et de partage de l'héritage familial à l'ère numérique. Il s'inscrit dans une dynamique de démocratisation de l'accès à la généalogie, permettant à chacun d'explorer ses racines et de se connecter à son histoire familiale, indépendamment de ses connaissances préalables ou de sa localisation géographique.

La méthode objet nous a permis une approche itérative et flexible, garantissant que la plateforme s'adapte constamment aux besoins des utilisateurs. Les résultats obtenus jusqu'à présent sont prometteurs et démontrent l'efficacité de notre approche méthodologique ainsi que la pertinence des technologies choisies pour le développement de cette solution.

Toutefois, notre travail ne constitue qu'une première étape d'un processus plus large. Outre les fonctionnalités mises en place, plusieurs questions restent sans réponse, ouvrant ainsi la porte à de futurs développements. Par exemple : comment pouvons-nous intégrer des analyses généalogiques avancées pour offrir des connaissances plus profondes aux utilisateurs ? Quelles nouvelles fonctionnalités pourraient enrichir davantage l'expérience utilisateur, tout en garantissant la sécurité et la confidentialité des données ? Les possibilités sont nombreuses.

En fin de compte, nous espérons que notre plateforme deviendra une référence incontournable dans le domaine de la généalogie numérique. Elle contribuera ainsi à la préservation et à la valorisation de l'histoire familiale pour les générations futures. Ce projet, bien qu'ambitieux, n'est qu'une étape dans la mission plus large de connecter les individus à leurs racines et de renforcer le lien familial à travers les outils numériques. Ce travail reste ouvert à toute amélioration envisageable.

Table des matières

Dédicace	i
Remerciements	iii
Table des figures	iv
Liste des tableaux	v
Sommaire	vi
Abréviation et Sigles	vii
Introduction	1
 I	 2
Présentation Générale de Mazala-Firm	2
1	3
Structure d’Accueil et le Sujet	3
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	6
1.2 Sujet	7
1.2.1 Contexte du sujet	7
1.2.2 Problématique du sujet	7
1.2.3 Intérêts du sujet	8
1.3 Concepts liés au sujet	8
 II	 10
Analyse et Conception	10
1	11
Étude préliminaire et Méthode	11
1.1 Étude 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	12
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	14
1.2.1.2	Les types de diagrammes d'UML	14
1.2.2	Processus de développement	15
1.2.2.1	Présentation du UP	15
1.2.2.2	Présentation du 2TUP	15
1.2.2.3	2TUP et UML	16
2	Analyse et conception	17
2.1	Analyse du besoin	17
2.1.1	Besoins fonctionnels	17
2.1.2	Besoins techniques	18
2.2	Conception du système	18
2.2.1	Spécifications fonctionnelles	18
2.2.1.1	Identification des acteurs	18
2.2.1.2	Diagramme de contexte statique	19
2.2.1.3	Identification des cas d'utilisation	19
2.2.1.4	Relation entre les cas d'utilisation (Use case)	19
2.2.1.5	Classification des cas d'utilisation (priorités, risques et itérations)	21
2.2.2	Spécification détaillée	23
2.2.2.1	Description textuelle d'un cas d'utilisation	23
2.2.2.2	Diagramme de séquence	26
2.2.2.3	Diagramme d'activité	28
2.2.3	Réalisations des cas d'utilisation	30
2.2.3.1	Quelques règles de gestion	30
2.2.3.2	Modèle du domaine (Diagramme de classe du système)	31
2.2.3.3	Diagramme d'objet	33
2.2.4	Conception architecturale	34
2.2.4.1	Architecture logicielle	34
2.2.4.2	Architecture globale de la solution (Diagramme de déploiement)	36
III	Évaluation et Réalisation	38
1	Évaluation du projet	39
1.1	Organisation du projet	39
1.2	Intervenant	39
1.3	Planification des tâches	39
1.4	Diagramme de Gantt	40
1.5	Estimation des charges	40
2	Les outils et techniques utilisés	42
2.1	Présentation des techniques	42
2.2	Présentation des outils	43
2.3	Les captures d'écrans	46
	Conclusion	52
	Webographie	D

Bibliographie

- [1] Martin FOWLER. *UML Distillé : Un guide succinct du langage de modélisation objet standard*. Addison-Wesley, 2003.
- [2] Franck ROQUES Pascal et Vallée. “UML 2 en action”. In : *De l’analyse des besoins à la conception J2EE* (2004).
- [3] Tim WEILKIENS. *Ingénierie des systèmes avec SysML/UML : Modélisation, Analyse, Conception*. Morgan Kaufmann, 2008.

Webographie

— **Architecture distribuée**

Lien : http://fr.wikipedia.org/w/index.php?title=Architecture_distribu%C3%A9e&oldid=212787328

Description : Article Wikipédia (encyclopédie libre).

Consulté le : 12-Avril-2024 à 15h00

— **TypeScript Documentation**

Lien : <https://www.typescriptlang.org/docs/>

Description : Documentation officielle pour TypeScript, un sur-ensemble typé de JavaScript qui se compile en JavaScript pur.

Consulté le : 16-février-2024 à 09h30

— **Next.js Documentation**

Lien : <https://nextjs.org/docs>

Description : Documentation officielle pour Next.js.

Consulté le : 16-février-2024 à 15h45

— **Capacitor**

Lien : <https://capacitorjs.com/docs>

Description : Documentation officielle pour Capacitor.

Consulté le : 16-février-2024 à 17h00

— **Prisma Getting Started**

Lien : <https://www.prisma.io/docs/getting-started>

Description : Guide de démarrage pour Prisma.

Consulté le : 18-février-2024 à 16h15

— **LaTeX Project Documentation**

Lien : <https://www.latex-project.org/help/documentation/>

Description : Documentation officielle du projet LaTeX.

Consulté le : 16-Janvier-2024 à 18h30

— **Lucia Auth Documentation**

Lien : <https://lucia-auth.com/>

Description : Documentation pour Lucia.

Consulté le : 03-Mai-2024 à 14h45

— **shadcn/ui Documentation**

Lien : <https://ui.shadcn.com/docs>

Description : Documentation pour shadcn/ui.

Consulté le : 16-février-2024 à 19h00

— **Docker Guides**

Lien : <https://docs.docker.com/guides/>

Description : Guides et documentation officielle pour Docker.

Consulté le : 27-février-2024 à 19h22