

## **DÉDICACES**

### **À mes parents**

Je ne saurais exprimer l'immense amour que je vous porte, vous qui n'avez cessé de me soutenir moralement et spirituellement ; ni le sentiment de profonde gratitude que je ressens pour tous les efforts et les sacrifices que vous n'avez jamais cessés de consentir pour mon instruction.

### **À tous ceux et celles qui, de près ou de loin,**

Ont œuvré pour la réussite de ce projet. Merci pour votre soutien, votre présence effective et vos encouragements quotidiens.

## REMERCIEMENTS

Mes remerciements vont tout d'abord à **Dieu** Tout-Puissant, pour sa miséricorde sans cesse renouvelée dans nos vies. Je tiens à exprimer mes sincères remerciements à toutes les personnes qui m'ont permis de réaliser ce document et mener ce projet à savoir :

- ❖ **M. KASSALOUWA Kolemahah hermann**, Directeur Général de KALAMAR, pour son accueil dans son cabinet et pour avoir mis à notre disposition les outils nécessaires à la réalisation de ce projet ;
- ❖ **M. AGBETI Kodjo**, Représentant Résident de l'IAI-TOGO, pour le cadre d'études qu'il nous a offert et pour tous les efforts entrepris pour nous offrir de bonnes conditions d'études ;
- ❖ **M. AMEYIKPO Nicolas**, Directeur des Affaires Académiques et de la Scolarité de l'IAI-TOGO pour tous les efforts qu'il déploie dans le compte de l'institut ;
- ❖ **M. DAVON Essè**, Chef Division Etudes et Scolarité de l'IAI TOGO qui ménage de grands efforts pour le suivi et le bon déroulement des cours et du calendrier scolaire de l'année ;
- ❖ **M. TIDJANI**, mon superviseur, qui n'a ménagé aucun effort en matière de conseils, d'encouragements et d'encadrement à mon égard ;
- ❖ **Le Corps professoral et administratif de l'IAI-TOGO**, pour tous les enseignements transmis ;
- ❖ Tous nos collègues de promotion que nous avons eu à côtoyer pendant ces trois années de formation.

# SOMMAIRE

DEDICACES .....	i
REMERCIEMENTS.....	ii
SOMMAIRE.....	iii
RÉSUMÉ.....	v
ABSTRACT .....	vi
GLOSSAIRE.....	vii
LISTES DES FIGURES .....	viii
LISTES DES TABLEAUX .....	ix
LISTES DES PARTICIPANTS AU PROJET .....	x
INTRODUCTION.....	1
PARTIE I : CAHIER DES CHARGES.....	2
1.1. PRÉSENTATIONS .....	3
1.2. THÈME DU STAGE.....	6
1.3. ETUDE DE L'EXISTANT .....	8
1.5. PROPOSITIONS DE SOLUTIONS .....	8
1.6. PLANNING PRÉVISIONNEL DE RÉALISATION .....	11
PARTIE II : ANALYSE ET CONCEPTION .....	14
2.1. CHOIX DE LA MÉTHODE D'ANALYSE ET JUSTIFICATION.....	15
2.2. CHOIX DE L'OUTIL DE MODÉLISATION ET JUSTIFICATION .....	18
2.3. ETUDE DÉTAILLÉE DE LA SOLUTION.....	19
PARTIE III : REALISATION ET MISE EN ŒUVRE.....	33
3.1. MATÉRIELS ET LOGICIELS UTILISÉS .....	34
3.2. ARCHITECTURES MATÉRIELLES ET LOGICIELS DE L'APPLICATION.....	39
3.3. SÉCURITÉ DE L'APPLICATION .....	41
3.4. MISE EN PLACE DE LA BASE DE DONNÉES .....	42
PARTIE IV : GUIDE D'EXPLOITATION.....	45
4.1. CONFIGURATION LOGICIELLE ET MATÉRIELLE .....	46
4.2. DÉPLOIEMENT ET SUIVI.....	47
4.3. MAINTENANCE : ACTIONS À MENER EN CAS DE CERTAINES ERREURS .....	53
PARTIE V : GUIDE D'UTILISATION.....	54
5.1. DESCRIPTION TEXTUELLE DU DOCUMENT .....	55
5.2. PLAN DE NAVIGATION .....	56
5.3. PRÉSENTATION DES DIFFÉRENTES INTERFACES DE L'APPLICATION.....	57
5.4. PRÉSENTATION DES ÉTATS.....	59
CONCLUSION.....	60

<b>BIBLIOGRAPHIE INDICATIVE.....</b>	<b>I</b>
<b>WEBOGRAPHIE INDICATIVE.....</b>	<b>II</b>
<b>TABLE DES MATIERES .....</b>	<b>III</b>

## RÉSUMÉ

En vue de l'obtention du diplôme d'Ingénieur des Travaux Informatiques de l'IAI-TOGO, nous avons effectué un stage à KALAMAR où, il a été soumis à notre étude, le thème : « **Plateforme de création d'un arbre généalogique** ».

Ce thème consiste à concevoir une application web afin de permettre aux utilisateurs de la plateforme MIFA de pouvoir créer leur arbre généalogique et de partager celui-ci avec les membres de leur famille.

Pour la réalisation de ce projet, une étape de modélisation réalisée avec le langage UML couplé au processus 2TUP nous a permis de maîtriser la complexité du système. Le langage de programmation Python couplé au framework Django, le Système de Bases de Données SQLite nous ont permis de développer notre application.

## ABSTRACT

To obtain the diploma of Engineer of Computer Works of the IAI-TOGO, we did an internship in KALAMAR where it was submitted to our study: "Platform for creating a family tree". This theme consists of designing a web application to allow users of the MIFA platform to create their family tree and share it with their family members.

For the realization of this project, a modeling step conducted with the UML language coupled with the 2TUP process allowed us to master the complexity of the system. The Python programming language coupled with the Django framework; the SQLite Database System allowed us to develop our application.

## GLOSSAIRE

Tableau 1 : glossaire

ABRÉVIATIONS ET ACRONYMES	DÉFINITIONS
IAI	Institut Africain d'Informatique
GLSI	Génie logiciel et Système d'Information
UML	Unified Modeling Language
2TUP	Two Track Unified Process
WSGI	Web Server Gateway Interface
ASGI	Asynchronous Server Gateway Interface
IDE	Integrated development Environment
OS	Operating System
MVT	Model View Template
RAM	Random Access Memory
ASR	Administration Systèmes et Réseaux
MTWI	Multimédia Technologie Web et Infographie
ORM	Object-Relational Mapping

# LISTES DES FIGURES

Figure 1 : Plan de localisation de l'IAI – TOGO .....	4
Figure 2 : Organigramme de la société KALAMAR .....	5
Figure 3 : Plan de localisation de la société KALAMAR .....	6
Figure 4 : Diagramme prévisionnel de GANTT du projet .....	13
Figure 5 : Logo de UML .....	15
Figure 6 : Logo du processus 2TUP .....	17
Figure 7 : Logo du logiciel StarUml.....	19
Figure 8 : Diagramme des cas d'utilisation de l'administrateur.....	21
Figure 9 : Diagramme des cas d'utilisation de l'utilisateur.....	22
Figure 10 : Diagramme de classes de notre système.....	27
Figure 11 : Diagramme de séquence Authentification .....	28
Figure 12 : Diagramme de séquence S'inscrire.....	29
Figure 13 : Diagramme de séquence Créer un arbre .....	30
Figure 14 : Diagramme d'activité Authentification .....	30
Figure 15 : Diagramme d'activité S'inscrire .....	31
Figure 16 : Diagramme d'activité Créer un arbre.....	31
Figure 17 : Diagramme de déploiement du système .....	32
Figure 18 : Logo de python .....	34
Figure 19 : logo de django.....	35
Figure 20 : logo de visual studio code.....	36
Figure 21 : logo de bootstrap.....	36
Figure 22 : Logo de HTML, CSS et JavaScript .....	37
Figure 23 : logo de Git .....	38
Figure 24 : logo de GitLab .....	38
Figure 25 : logo de Google Chrome .....	39
Figure 26 : Architecture MVT.....	41
Figure 27 : table Arbre .....	43
Figure 28 : Table Membre.....	43
Figure 29 : Table Évènement .....	44
Figure 30 : objet application.....	48
Figure 31 : Clé secrète en production.....	48
Figure 32 : Debug à False.....	49
Figure 33 : installation de python, PostgreSQL, Nginx .....	49
Figure 34 : création de la base de données et d'un utilisateur.....	50
Figure 35 : fichier de configuration de service gunicorn.....	50
Figure 36 : test de gunicorn.....	50
Figure 37 : fichier de configuration du serveur nginx .....	51
Figure 38 : application déployée .....	51
Figure 39 : page d'accueil .....	57
Figure 40 : page liste des arbres .....	57
Figure 41 : page détail d'un arbre.....	58
Figure 42 : page présentation visuelle d'un arbre.....	58
Figure 43 : Exportation en PDF d'un arbre .....	59
Figure 44 : statistiques d'un arbre .....	59



## LISTES DES TABLEAUX

Tableau 1 : glossaire.....	vii
Tableau 2 : liste des participants au projet .....	x
Tableau 3 : coût matériel.....	9
Tableau 4 : coût d'hébergement .....	10
Tableau 5 : Coût de la connexion .....	10
Tableau 6 : coût de conception.....	10
Tableau 7 : coût total .....	11
Tableau 8 : Planning prévisionnel de réalisation du projet MIFA .....	12
Tableau 9 : cas d'utilisation du projet .....	20
Tableau 10 : Description textuelle du cas d'utilisation s'authentifier .....	23
Tableau 11 : description textuelle du cas d'utilisation s'inscrire .....	24
Tableau 12 : description textuelle du cas d'utilisation créer un arbre .....	25
Tableau 13 : configuration matérielle.....	46
Tableau 14 : configuration logicielle.....	46
Tableau 15 : fiche des évènements.....	52
Tableau 16 : sites consultés .....	II

**LISTES DES PARTICIPANTS AU PROJET***Tableau 2 : liste des participants au projet*

NOM ET PRENOMS	FONCTIONS	RÔLE
AGBOLOSSOU Gédéon-Benoit Koffi	Etudiant en 3ème année, option Génie Logiciel & Systèmes d'information à l'IAI-TOGO	Réalisateur
M. KASSALOUWA Hermann	Ingénieur informaticien & Directeur Général de KALAMAR	Maitre de stage
M. TIDJANI	Enseignant à l'IAI-TOGO	Superviseur

## INTRODUCTION

Un arbre généalogique nous permet de retracer l'histoire de notre famille à travers le temps. Il s'agit d'une représentation graphique qui remonte notre lignée aussi loin que possible. Les arbres généalogiques commencent souvent par la génération la plus récente, puis remontent en arrière.

La création d'un arbre généalogique fait appel à la collecte de beaucoup d'informations et nécessite un moyen sûr de conservation des informations car, elle a pour but de retracer non seulement l'histoire d'une famille à travers le temps mais aussi de la conserver.

Dans le but d'apporter une solution optimisée, de faciliter la collecte d'information et la création d'un arbre généalogique et surtout la conservation des données, le cabinet KALAMAR nous a confié le thème : « **PLATEFORME DE CRÉATION D'UN ARBRE GÉNÉALOGIQUE** ». Cette solution numérique vise à permettre aux utilisateurs de notre plateforme de créer leur arbre généalogique et de le partager avec les membres de leur famille.

Ainsi, ce document sera subdivisé en cinq (05) parties à savoir : le cahier des charges, l'analyse et la conception, la réalisation et mise en œuvre, le guide d'exploitation et le guide d'utilisation de l'application.

## PARTIE I : CAHIER DES CHARGES

Le cahier des charges est un document formel qui comporte les besoins exprimés par l'utilisateur ainsi que les différentes solutions à mettre en œuvre. C'est un document qui doit être respecté lors de la conception d'un projet. Il permet aux différentes parties prenantes de s'accorder sur les spécificités des objectifs et les résultats attendus à l'issue du projet. Les grandes lignes de notre cahier des charges porteront sur les présentations (de l'IAI-TOGO et du centre d'accueil), le thème du stage, l'étude et la critique de l'existant, les propositions et choix de solution ainsi que sur le planning prévisionnel de notre projet.

## 1.1. PRÉSENTATIONS

### 1.1.1. Brève présentation de l'IAI-TOGO

L'institut africain d'informatique représentation du Togo (IAI – TOGO) est une école supérieure inter-état d'enseignement informatique. Créé en 1971 à Fort Lamy actuel N'Djamena, au Tchad, son siège se trouve à Libreville au Gabon. Il compte onze (11) Etats membres à savoir le Bénin, le Burkina Faso, le Cameroun, la République Centrafricaine, la Côte d'Ivoire, la République du Congo, le Gabon, le Niger, le Sénégal, le Tchad et le Togo. La Représentation du Togo (IAI-TOGO) a ouvert ses portes le 24 octobre 2002. L'accord d'établissements entre la République Togolaise et l'Institut Africain d'Informatique a été signé le 12 mai 2006.

L'IAI – TOGO forme en trois (03) ans des ingénieurs des travaux informatiques dans les trois (03) filières à savoir : **Génie logiciel et systèmes d'information (GLSI)**, **Administrations Systèmes et Réseaux (ASR)** et le **Multimédia, Technologie Web et Infographie (M-TWI)**.

L'IAI – TOGO dans le but de permettre à ses étudiants d'avoir une meilleure intégration dans le milieu professionnel, les amènes à effectuer un stage d'une durée de trois (03) mois en entreprise dans le cadre d'un projet de fin de formation pour l'obtention du **Diplôme d'Ingénieur des Travaux Informatiques**.

L'IAI-TOGO est situé sur la rue de la Kozah à Nyékonakpoè, derrière le bâtiment de l'UTB circulaire, logé dans le bâtiment du Centre National d'Etudes et de Traitements Informatique (CENETI).

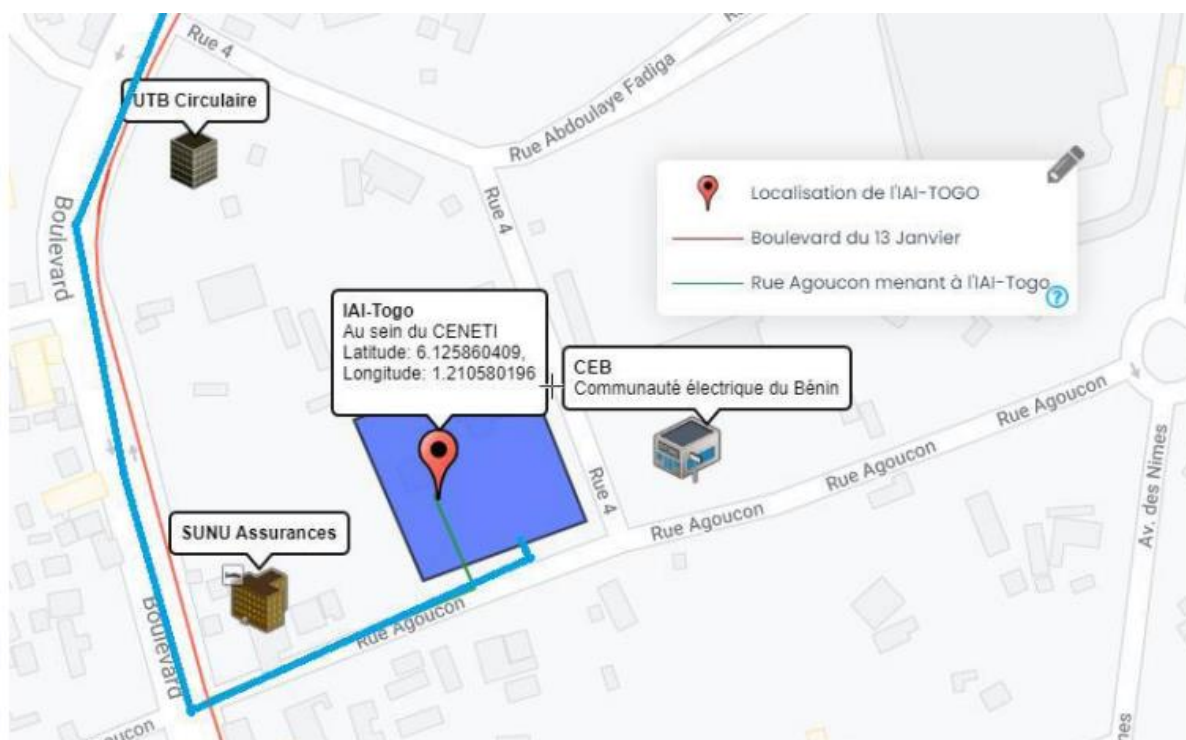


Figure 1 : Plan de localisation de l'IAI – TOGO

### 1.1.2. Présentation du cadre de stage

Notre stage de fin de formation a été effectué au sein du cabinet KALAMAR. Elle est un cabinet de prestation de services en informatique et en cartographie numérique.

#### ❖ Statut

KALAMAR se positionne comme la société de prestation de services basée sur le numérique. Elle propose des services et des logiciels d'aide à la gestion des entreprises, des services en conception des applications basées sur la cartographie. En plus de cela, elle intervient dans la collecte, le traitement et l'analyse de données cartographiques ; ainsi, fournit-elle des cartes web, des cartes sur papier et autres supports. La formation dans les domaines d'intervention de l'entreprise fait aussi partie des services qu'offre KALAMAR.

#### ❖ Mission

Kalamar s'est donné comme mission d'être la société qui peut conseiller, concevoir et distribuer des solutions simples et adaptées à chaque client ; accompagner toute structure vers une meilleure intégration des solutions technologiques dans leur environnement de travail et de développer une main d'œuvre locale qualifiée par la formation.

### ❖ Activités

Kalamar exerce dans les activités suivantes :

- Conseil
- Conception
- Prestation de services
- Formation

### ❖ Organigramme

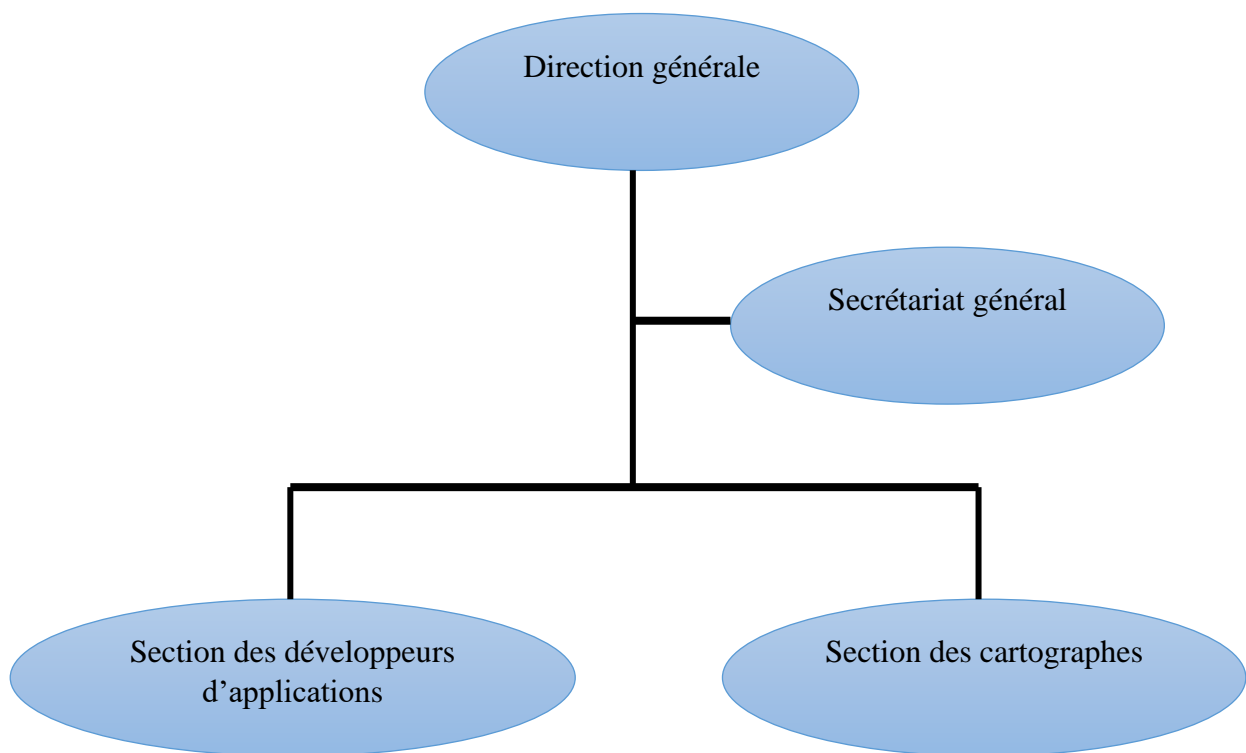


Figure 2 : Organigramme de la société KALAMAR

### ❖ Service d'accueil

Nous avons effectué notre stage dans le bureau dédié au stagiaire à la section des développeurs d'applications.

### ❖ Plan de localisation



Figure 3 : Plan de localisation de la société KALAMAR

## 1.2. THÈME DU STAGE

### 1.2.1. Présentation du sujet

Notre projet vise à mettre à la disposition de KALAMAR, un cabinet de développement informatique, une solution informatique performante et optimisée qui permettra à un individu de créer son arbre généalogique. Cette application permettra à KALAMAR de répondre aux besoins de la population pour leur permettre une conception plus rapide en matière de création d'arbre généalogique et un moyen de conserver numériquement leurs connaissances sur tous les membres de leur famille.

### 1.2.2. La problématique du sujet

Dans la création d'un arbre généalogique, le temps, la qualité de des informations à collecter et la manière de conserver cette dernière pour que cela soit utile aux générations à venir sont importants.



Après réflexion nous avons d'automatiser la création des arbres généalogiques ce qui était auparavant manuelle.

Dans ce fait, nos investigations ont relevé les problèmes suivants :

- Comment collecter les informations au prêt de tous les membres de sa famille ?
- Comment sauvegarder les informations collecter pour pouvoir les utiliser sur le long terme ?
- Comment avoir des statistiques par rapport aux données collectées ?

### **1.2.3. Intérêt du sujet**

#### **1.2.3.1. Objectifs**

Notre travail consistera à mettre en place une application permettant à un utilisateur de créer son arbre généalogique et le partager avec les autres membres de sa famille afin de le construire et conserver l'histoire de leur lignée. Cette application devra permettre de :

- ❖ Créer son arbre généalogique ;
- ❖ Sauvegarder des informations sur les personnes ;
- ❖ Imprimer son arbre généalogique ;
- ❖ Partager son arbre généalogique avec d'autres personnes ;
- ❖ Avoir des statistiques ;
- ❖ Conserver son arbre généalogique.

#### **1.2.3.2. Résultats attendus**

L'application à concevoir doit fournir les résultats suivants :

- ❖ L'arbre généalogique est créé ;
- ❖ Les informations des personnes sont sauvegardées ;
- ❖ L'impression de l'arbre généalogique est opérationnelle ;
- ❖ Le partage de son arbre généalogique est fait ;
- ❖ Les statistiques sont présents ;
- ❖ L'arbre généalogique est conservé.

### 1.3. ETUDE DE L'EXISTANT

Nous ne saurions débiter ce travail sans avoir une idée claire et précise sur l'existant tel qu'il soit. Notre étude de l'existant nous a permis de relever que le cabinet KALAMAR ne dispose pas encore de solution informatique performante et optimisée à mettre à la disposition de la population pour la création d'arbre généalogique. Les moyens utilisés jusqu'à présent sont :

- La création sur papier
- La création avec le logiciel office Word

### 1.4. CRITIQUE DE L'EXISTANT

Après notre étude de l'existant nous avons déceler les problèmes suivants :

- ❖ La création d'un arbre généalogique sur du papier est très soumise au risque de perte d'informations ;
- ❖ Le logiciel office n'est pas un moyen optimisé de conservation de données et risque la perte de l'information si le fichier se corrompt ;
- ❖ Très difficile d'exploiter les informations contenues sur ces deux supports à grande échelle ;
- ❖ Difficulté de garder sur un temps long et en bon état le papier ;
- ❖ Impossibilité d'avoir des statistiques en temps réel ;
- ❖ Impossibilité de faire travailler tous les membres d'une famille sur la création d'un arbre.

### 1.5. PROPOSITIONS DE SOLUTIONS

Afin de pallier ces insuffisances, le cabinet KALAMAR a décidé de se doter de sa propre solution informatique, à savoir une application web et ne souhaite pas promouvoir l'utilisation d'une autre application. Nous avons donc proposé de développer l'application pour le cabinet, ce qui constitue une opportunité d'apprentissage pour nous.

#### 1.5.1. Evaluation technique de la solution

Cette solution consiste à développer une application web permettant aux utilisateurs de la plateforme MIFA de créer de manière optimale leur arbre généalogique et de conserver de façon numérique leurs données.

❖ **Avantages :**

- Facile à maintenir et à faire évoluer puisque KALAMAR en serait le propriétaire ;
- Les suggestions et les remarques des utilisateurs seront plus facilement prises en compte et ce avec beaucoup plus de souplesses lors de l'implémentation des mises à jour ;
- Plus de visibilité sur le marché togolais du numérique ;
- L'objectif pédagogique rempli.

❖ **Inconvénients :**

- Solution non existante pour le moment ;
- Le développement de la solution et sa mise en œuvre peuvent prendre du temps compte tenu des nombreux besoins.

### 1.5.2. Evaluation financière de la solution

❖ **Coût matériel**

Tableau 3 : coût matériel

DÉSIGNATION	DESCRIPTION	COÛT UNITAIRE (F CFA)	QUANTITÉ	COÛT (F CFA)
Ordinateur Dell Precision 7560	Intel® Core™ i5- 11500H de 11 générations, vPro® (12 Mo de cache, 6 cœurs, 12 threads, 2,90 GHz à 4,60 GHz Turbo	1.282.300 <sup>1</sup>	1	1.282.300
Total				1.282.300

<sup>1</sup> Source : [Ordinateur portable Dell Precision 7560 AI & VR Ready Mobile Workstation | Dell États-Unis](#)

### ❖ Coût d'hébergement

Tableau 4 : coût d'hébergement

DÉSIGNATION	DESCRIPTION	COÛT UNITAIRE (F CFA)	QUANTITÉ	COÛT (F CFA)
Serveur d'hébergement DigitalOcean	Offre 4 processeurs <sup>2</sup> 8 Go de mémoire 160 Go de stockage 5 To de bande passante	30.336	12 Mois	364.032
Total				364.032

### ❖ Coût de la connexion

Tableau 5 : Coût de la connexion

DÉSIGNATION	DESCRIPTION	COÛT UNITAIRE (F CFA)	QUANTITÉ	COÛT (F CFA)
Abonnement internet	FAI (Fournisseur d'Accès Internet) Canalbox 50Mb/s	15.000/Mois	12 Mois	180.000
Total				180.000

### ❖ Coût de conception

Tableau 6 : coût de conception

DÉSIGNATION	COÛT UNITAIRE (F CFA)	COÛT (F CFA)
Développement d'application	5.000 * 480 heures (8h * 60 jours)	2.400.000
Formation	5.000 * 20 heures (4h * 5 jours )	100.000
Total		2.500.000

<sup>2</sup> Source : [Droplets | DigitalOcean's Scalable Virtual Machines](#)

## ❖ Coût total

Tableau 7 : coût total

DÉSIGNATION	COUT (F CFA)
Coût conception	2.500.000
Coût d'hébergement	364.032
Total	2.864.032

NB : Seuls les prix des matériaux non existants sont pris en compte pour le calcul du prix total.

Le cabinet dispose d'abonnement internet

### 1.6. PLANNING PRÉVISIONNEL DE RÉALISATION

Le planning prévisionnel représente le chronogramme que nous souhaitons suivre durant la phase de déroulement de notre projet pour finir à temps et exécuter toutes les fonctionnalités qui nous ont été demandées.

Tableau 8 : Planning prévisionnel de réalisation du projet MIFA

N°	Tâches	Date début	Date fin	Durée
1	Insertion dans l'entreprise	11 mai 2022	13 mai 2022	3 jours
2	Apprentissage des outils et langages utilisés dans la société	16 mai 2022	25 mai 2022	10 jours
3	Connaissance du thème	18 mai 2022	18 mai 2022	1 jour
4	Collecte des informations sur le thème	19 mai 2022	25 mai 2022	7 jours
5	Rédaction et validation du cahier de charges	26 mai 2022	31 mai 2022	6 jours
6	Analyse, conception et modélisation du système	1 juin 2022	10 juin 2022	10 jours
7	Création des apps	11 juin 2022	15 juin 2022	5 jours
8	Développement des Template	16 juin 2022	22 juin 2022	7 jours
9	Incorporation des Template dans Django	23 juin 2022	25 juin 2022	3 jours
10	Programmation de l'app arbre	26 juin 2022	30 juin 2022	5 jours
11	Programmation de l'app membre	1 juillet 2022	12 juillet 2022	12 jours
12	Programmation de l'app évènement	6 juillet 2022	12 juillet 2022	7 jours
13	Programmation de l'app utilisateur	13 juillet 2022	17 juillet 2022	5 jours
14	Test fonctionnels, corrections et améliorations	18 juillet 2022	22 juillet 2022	5 jours
15	Déploiement de la solution	22 juillet 2022	23 juillet 2022	2 jours
16	Rédaction du mémoire	17 mai 2022	31 juillet 2022	76 jours
17	Validation du mémoire et des corrections	1 août 2022	12 août 2022	12 jours
18	Dépôt du mémoire	17 août 2022	17 août 2022	1 jour

MIFA

KALAMAR  
AGBOLOSSOU GEDEON

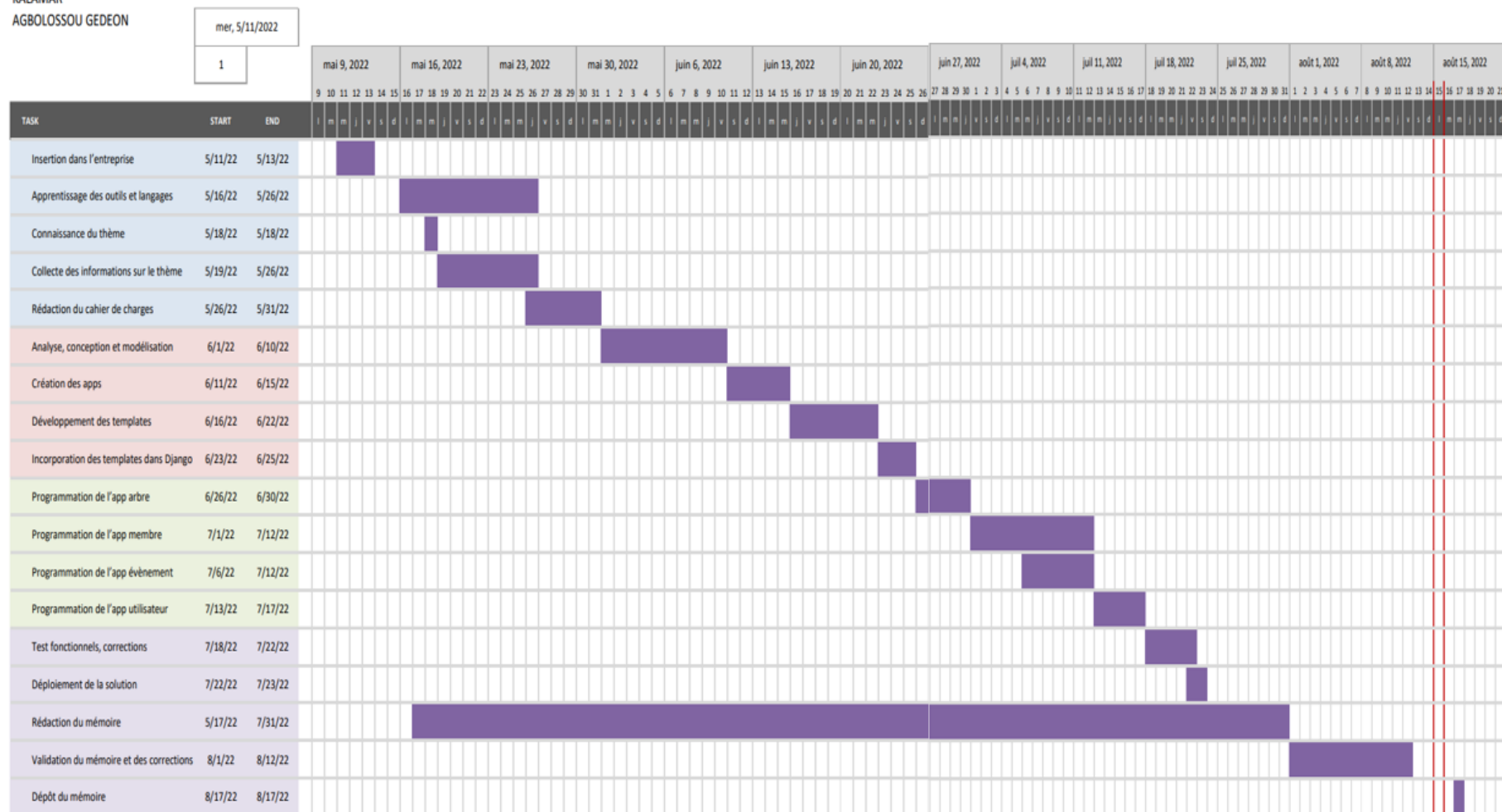


Figure 4 : Diagramme prévisionnel de GANTT du projet

## PARTIE II : ANALYSE ET CONCEPTION



La phase d'analyse et de conception constitue une phase très importante dans la conception d'un logiciel et elle est la seconde étape de notre projet. Cette phase nous permettra de décrire explicitement les différents processus à informatiser en utilisant une méthode d'étude et un outil de modélisation approprié. Dans cette partie, il sera question de présenter, en premier lieu les outils d'analyse et de modélisation choisis et en second lieu la méthode de modélisation pour la conception du système.

## 2.1. CHOIX DE LA MÉTHODE D'ANALYSE ET JUSTIFICATION

Une méthode d'analyse est un procédé qui a pour objectif de permettre la formalisation des étapes préliminaires du développement d'un système afin de rendre ce développement plus fidèle aux besoins du client.



Figure 5 : Logo de UML

Le langage de modélisation unifié, en anglais Unified Modeling Language (UML) est un langage de modélisation graphique à base de pictogrammes<sup>3</sup>, conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

Un seul type de modèle n'étant pas suffisant pour décrire correctement un système, UML dispose de plusieurs types de modèles appelés diagrammes (13 en tout), chaque diagramme représentant une vue distincte du système. Il n'est bien sûr pas nécessaire d'utiliser tous les

<sup>3</sup> Dessins figuratifs stylisés ayant fonction de signe

diagrammes pour modéliser un système, il faut choisir parmi tous les diagrammes disponibles ceux qui sont les plus pertinents et les mieux adaptés à la modélisation de notre système.

- ❖ Les diagrammes structurels ou statiques :
  - ✓ Le diagramme de classes ;
  - ✓ Le diagramme d'objets ;
  - ✓ Le diagramme de composants ;
  - ✓ Le diagramme de déploiement ;
  - ✓ Le diagramme de structures composites ;
  - ✓ Le diagramme de packages ;
  - ✓ Le diagramme de profils.
- ❖ Les diagrammes de comportement :
  - ✓ Le diagramme de cas d'utilisation ;
  - ✓ Le diagramme d'activité ;
  - ✓ Le diagramme d'état-transition.
- ❖ Les diagrammes d'interaction ou dynamiques :
  - ✓ Le diagramme de séquence ;
  - ✓ Le diagramme de communication ;
  - ✓ Le diagramme global d'interaction ;
  - ✓ Le diagramme de temps.

UML ne propose pas l'ordre qui doit régir les diagrammes. C'est ainsi que les méthodes viennent appuyer le langage pour parfaire l'approche objet. Pour ce faire, il faudrait coupler UML à un processus de développement dont le rôle est de produire des applications de qualité répondant aux besoins des utilisateurs, afin d'être utilisée comme méthode d'analyse et de conception.

Nous allons associer ici à notre langage de modélisation unifié, le processus 2TUP.

**Qu'est-ce que le processus 2TUP ?**

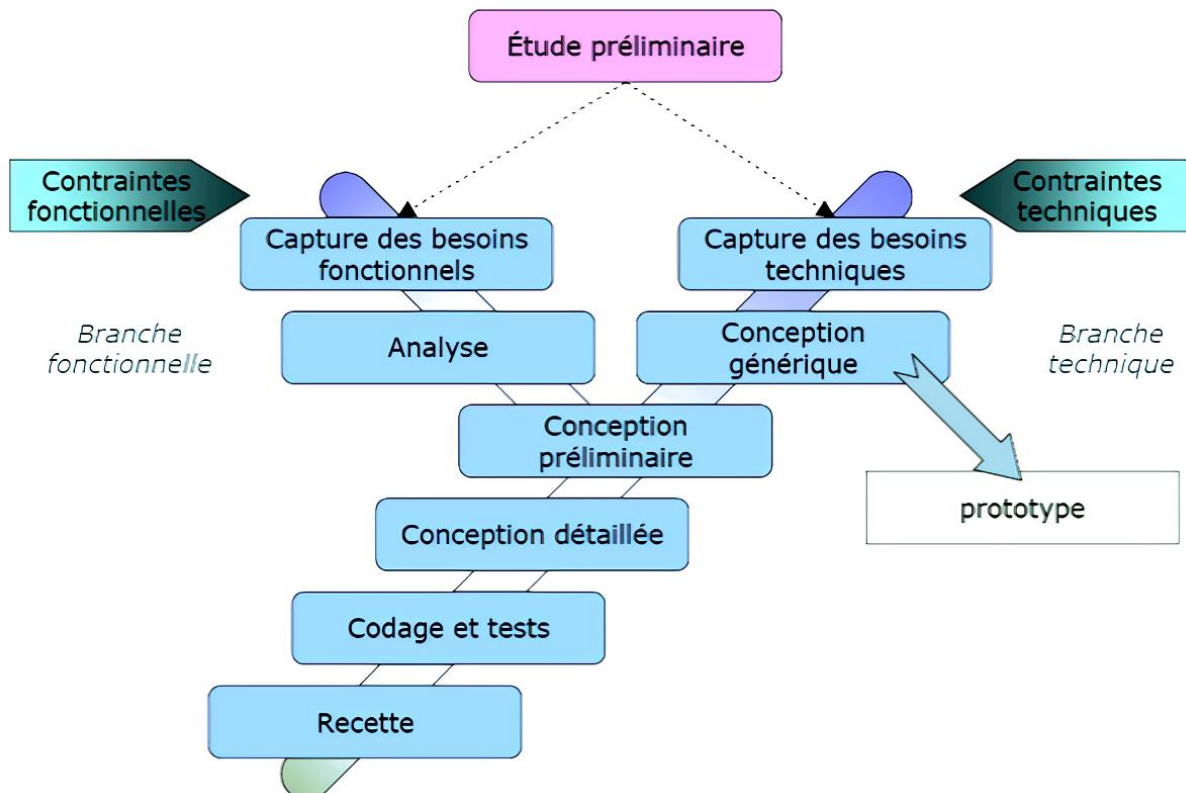


Figure 6 : Logo du processus 2TUP

Two Track Unified Process (2TUP), est un processus de Développement Logiciel qui implémente le Processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier d'abord les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système et ensuite à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire). Le processus s'articule ensuite autour de 3 phases essentielles : Une branche technique, une branche fonctionnelle, une phase de réalisation.

#### ❖ La branche technique(droite)

- ✓ La capture des besoins techniques qui recense toutes les contraintes sur les choix de dimensionnement et la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique) ;
- ✓ La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser

les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.

#### ❖ Branche fonctionnelle(gauche)

- ✓ Elle capture des besoins fonctionnels qui produisent le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs ;

Nous utiliserons les diagrammes de cas d'utilisation, d'activité et de séquence système d'UML pour décrire cette étape.

- ✓ L'analyse qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.

Nous utiliserons le diagramme de classe d'UML pour décrire cette étape.

#### ❖ Phase de réalisation(milieu)

Elle correspond à :

- ✓ Une conception préliminaire qui représente une étape délicate car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;
- ✓ La conception détaillée qui étudie comment réaliser chaque composant ;

Nous utiliserons le diagramme de déploiement d'UML pour décrire cette étape.

- ✓ L'étape de codage qui produit ses composants et teste au fur et à mesure les unités de code réalisées ;
- ✓ L'étape de recette, qui consiste à valider les fonctionnalités du système développé.

## 2.2.CHOIX DE L'OUTIL DE MODÉLISATION ET JUSTIFICATION

Ayant choisi le langage UML, nous avons une pléiade d'outils qui peuvent nous permettre de modéliser notre système. Nous pouvons citer notamment Astah, ArgoUML, BOUML, Modélio, PowerAMC, PowerDesigner, StarUML, Umbrello, UMLStudio et Entreprise Architect.

Pour mener à bien notre analyse et atteindre les objectifs du projet, notre choix s'est porté sur StarUML.

### 2.2.1. Présentation et justification du choix de StarUML

**StarUML** est un outil d'ingénierie logicielle pour la modélisation de systèmes utilisant le langage de modélisation unifié, ainsi que le langage de modélisation systèmes et les notations de modélisation classique. Il est publié par MKLabs et est disponible sur Windows, Linux et MacOS. Il est le successeur d'un logiciel de modélisation orienté objet appelé Plastic. Plastic 1.0 a été publié en 1997 pour soutenir la notation OMT<sup>4</sup>. La version 1.1 publiée en 1998 a abandonné l'OMT en faveur de l'UML. Le logiciel a été renommé StarUML en 2005 en vue de le publier en open source.



Figure 7 : Logo du logiciel StarUml

Nous avons choisi StarUML car :

- Compatible avec le métamodèle et les diagrammes standard UML 2.x : classe, objet, cas d'utilisation, composant, déploiement, structure composite, séquence, communication, graphique d'état, activité, synchronisation, débordement d'interaction, flux d'informations et diagramme de profil.
- Prise en charge de la modélisation avec les diagrammes SysML<sup>5</sup> : exigence, définition de bloc, bloc interne et diagramme paramétrique.
- Prise en charge de la création de diagrammes entité-relation (ERD), de diagrammes de flux de données (DFD) et de diagrammes d'organigramme.

## 2.3. ETUDE DÉTAILLÉE DE LA SOLUTION

Dans cette section, il sera question de représenter les différents diagrammes d'UML, conformément à la structure préconisée par 2TUP.

<sup>4</sup> Technique de Modélisation Objet

<sup>5</sup> Langage de modélisation systèmes

### 2.3.1. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation traduit tout ce que l'utilisateur exprime comme action sur le logiciel ou le système à modéliser. C'est une représentation faisant intervenir les acteurs et les cas d'utilisation(ovales). Il traduit les besoins des utilisateurs vis-à-vis du système développé.

#### ❖ Acteurs

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie.

- Les acteurs principaux :
  - ✓ Administrateur

C'est l'acteur qui effectue les configurations de base du système. Il a une vue d'ensemble sur le système et constitue un garde-fou pour le maintien des activités du système ;

- ✓ Utilisateur

C'est un utilisateur de la plateforme qui possède un compte.

- Les acteurs secondaires :
  - ✓ Serveur-SMTP : c'est le serveur mail qui intervient dans l'envoi des emails.

#### ❖ Les cas d'utilisation

Un cas d'utilisation exprime le comportement du système en termes d'action et réaction face à un besoin d'un utilisateur. Dans notre projet nous avons identifié les cas d'utilisations suivants :

Tableau 9 : cas d'utilisation du projet

Cas d'utilisation		Acteurs
<b>S'authentifier</b>		- Administrateur - Utilisateur
<b>Gérer les utilisateurs</b>	- Afficher la liste des utilisateurs - Supprimer un utilisateur - Désactiver un utilisateur - Créer un utilisateur	- Administrateur
<b>Gérer les articles</b>	- Ajouter un article - Modifier un article	- Administrateur

	<ul style="list-style-type: none"> <li>- Désactiver un article</li> <li>- Supprimer un article</li> <li>- Afficher la liste des articles</li> </ul>	
<b>Gérer son arbre généalogique</b>	<ul style="list-style-type: none"> <li>- créer un arbre généalogique</li> <li>- modifier un arbre</li> <li>- afficher la liste des arbres</li> <li>- supprimer un arbre</li> <li>- imprimer un arbre</li> <li>- voir un arbre</li> </ul>	<ul style="list-style-type: none"> <li>- Utilisateur</li> </ul>
<b>Gérer les membres</b>	<ul style="list-style-type: none"> <li>- Ajouter un membre</li> <li>- Modifier un membre</li> <li>- Supprimer un membre</li> <li>- Afficher la liste des membres</li> </ul>	<ul style="list-style-type: none"> <li>- Utilisateur</li> </ul>
<b>S'inscrire</b>		<ul style="list-style-type: none"> <li>- Utilisateur</li> </ul>
<b>Inviter une personne</b>		<ul style="list-style-type: none"> <li>- Utilisateur</li> </ul>
<b>Consulter les statistiques</b>		<ul style="list-style-type: none"> <li>- Utilisateur</li> </ul>

On peut donc déduire les diagrammes de cas d'utilisation par acteurs

#### ❖ Administrateur

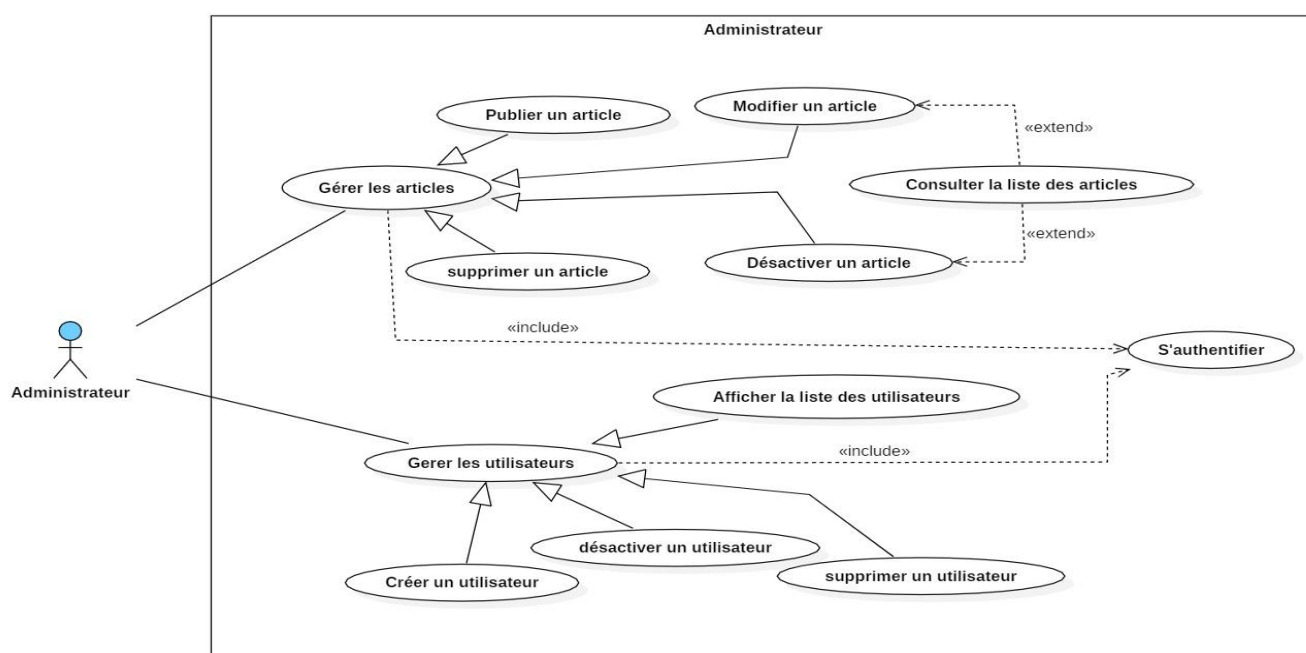


Figure 8 : Diagramme des cas d'utilisation de l'administrateur

❖ Utilisateur

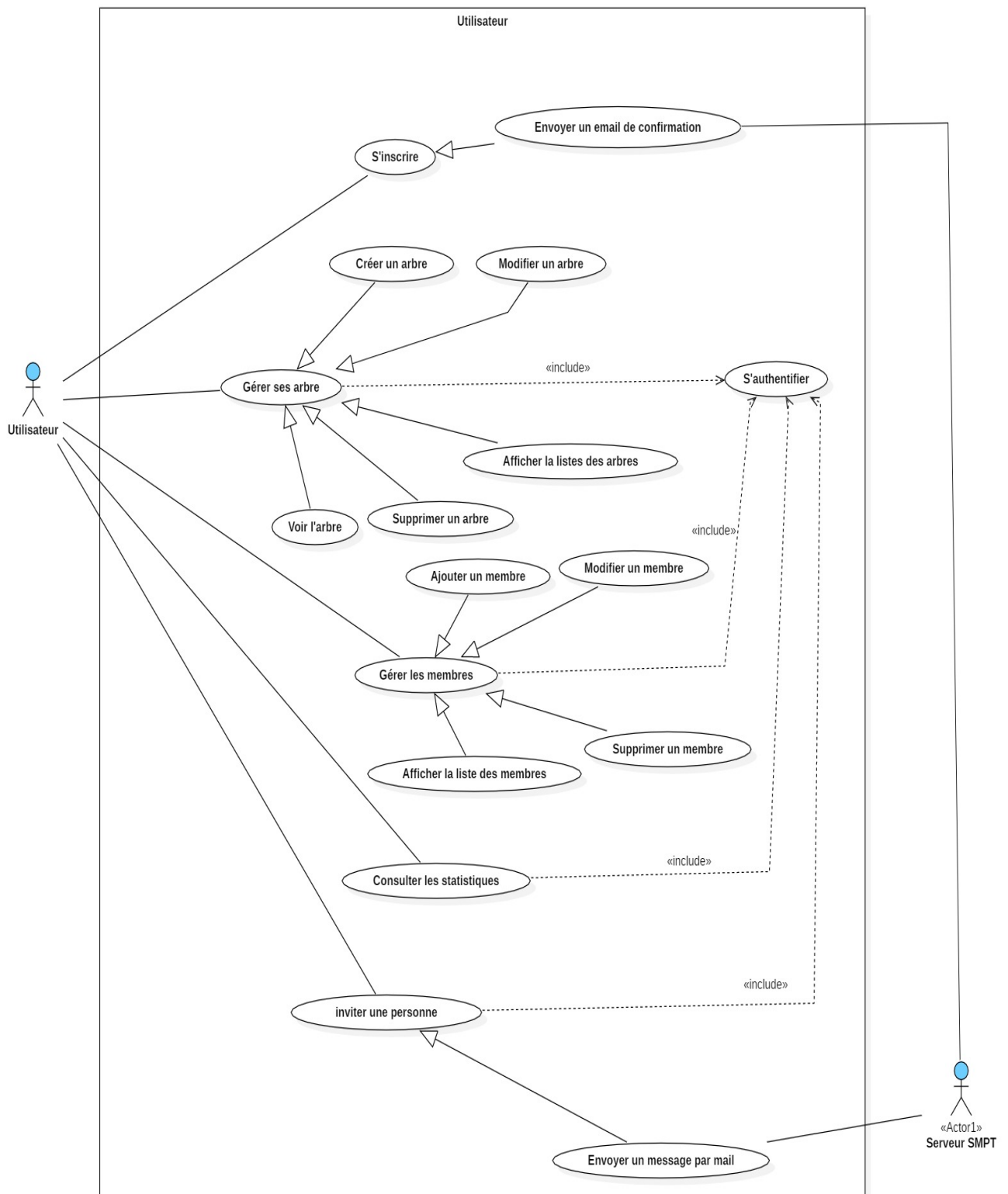


Figure 9 : Diagramme des cas d'utilisation de l'utilisateur



### 2.3.2. Description textuelle de certains cas d'utilisation

La description textuelle d'un cas d'utilisation permet de décrire au mieux le fonctionnement du système afin d'en faciliter la compréhension par tout le monde.

#### ❖ Cas d'utilisation S'authentifier

Tableau 10 : Description textuelle du cas d'utilisation s'authentifier

SOMMAIRE D'IDENTIFICATION
Titre : S'authentifier
Acteurs : Administrateur, Utilisateur
Résumé : Ce cas permet à un utilisateur d'être reconnu par le système afin d'accéder aux onglets qui le concernent
Responsable : AGBOLOSSOU Gédéon- Benoît Koffi
Version : 1.0
Date de création : 20/07/2022
DESCRIPTION
<b>Préconditions</b>
Le système est fonctionnel
La page d'authentification est accessible
<b>Scénario nominal</b>
<ol style="list-style-type: none"> <li>1- L'utilisateur saisit les informations nécessaires pour se connecter (E1, E2)</li> <li>2- L'utilisateur valide la saisie</li> <li>3- Le système vérifie si les informations saisies sont correctes (A1)</li> <li>4- Le système affiche la page correspondante au profil de l'utilisateur</li> </ol>
<b>Scénario Alternatif</b>
<p>A1 : les informations saisies sont incorrectes</p> <p>Ce scénario démarre au point 2 du scénario nominal lorsque l'utilisateur valide la saisie des informations pour se connecter</p> <p>L'utilisateur est informé que les informations saisies sont incorrectes</p> <p>Le scénario reprend au point 1 du scénario nominal</p>
<b>Scénario d'exception</b>
<p>E1 : l'utilisateur annule la saisie</p> <p>Le scénario nominal est interrompu</p>

E2 : l'utilisateur appuie sur le lien mot de passe oublié. L'utilisateur est redirigé vers une autre page l'invitant à mettre son email pour réinitialiser son mot de passe.

**Post condition**

L'utilisateur s'est authentifié

Le système est toujours fonctionnel

❖ **Cas d'utilisation S'inscrire**

Tableau 11 : description textuelle du cas d'utilisation s'inscrire

SOMMAIRE D'IDENTIFICATION
Titre : S'inscrire
Acteurs : Utilisateur
Résumé : Ce cas permet à un utilisateur de pouvoir s'inscrire afin d'avoir les droits pour accéder à l'application MIFA
Responsable : AGBOLOSSOU Gédéon- Benoît Koffi
Version : 1.0
Date de création : 11/07/2022
DESCRIPTION
<b>Préconditions</b>
Le système est fonctionnel
La page d'inscription est accessible
<b>Scénario nominal</b>
<ol style="list-style-type: none"> <li>1- L'utilisateur remplit les différents champs nécessaires pour s'inscrire (E1)</li> <li>2- L'utilisateur valide la saisie (A1)</li> <li>3- Le système procède à une vérification en tenant compte des champs obligatoires, des informations qui ne doivent pas être dupliquées dans la base de données comme le compte Gmail et l'identifiant et que les exigences du champ mot de passe sont respectées. (A2, A3)</li> <li>4- Le système envoie un lien de confirmation dans la boîte mail de l'utilisateur</li> <li>5- Le système informe l'utilisateur d'aller confirmer son email avant de se connecter</li> </ol>
<b>Scénario Alternatif</b>
A1 : les champs obligatoires du formulaire ne sont pas totalement renseignés
Ce scénario se déclenche lorsque l'utilisateur ne renseigne pas tous les champs obligatoires.

L'utilisateur est informé que ces champs ne doivent pas être vides

Le scénario reprend au point 1 du scénario nominal

A2 : le compte mail ou l'identifiant donné par l'utilisateur existe déjà dans la base donnée

Ce scénario se déclenche lorsque l'utilisateur donne un compte mail ou un identifiant qui existe déjà dans la base de données

L'utilisateur est informé que le compte mail ou l'identifiant existe déjà

L'utilisateur est invité à changer les données

Ce scénario reprend au point 1 du scénario nominal

A3 : le mot de passe donné par l'utilisateur ne respecte pas les exigences demandées

Ce scénario se déclenche lorsque l'utilisateur donne un mot de passe qui ne respecte les exigences en termes de mot de passe demandé par l'application

L'utilisateur est invité à prendre un autre mot de passe qui respecte les exigences

Ce scénario reprend au point 1 du scénario nominal

#### Scénario d'exception

E1 : l'utilisateur annule la saisie

Le scénario nominal est interrompu

#### Post condition

L'utilisateur s'est inscrit

### ❖ Cas d'utilisation Créer un arbre

Tableau 12 : description textuelle du cas d'utilisation créer un arbre

SOMMAIRE D'IDENTIFICATION
Titre : Créer un arbre
Acteurs : Utilisateur
Résumé : Ce cas permet à un utilisateur de pouvoir créer un arbre généalogique pour pouvoir y ajouter des membres
Responsable : AGBOLOSSOU Gédéon- Benoît Koffi
Version : 1.0
Date de création : 20/07/2022
DESCRIPTION

### Préconditions

Le système est fonctionnel  
L'utilisateur s'est authentifié

### Scénario nominal

- 1- L'utilisateur clique sur le menu arbre
- 2- L'utilisateur clique sur le bouton ajouter un arbre
- 3- Le système lui affiche le formulaire de création d'un arbre
- 4- L'utilisateur remplit les différents champs nécessaires du formulaire (E1)
- 5- L'utilisateur crée son arbre en cliquant sur le bouton créer
- 6- Le système procède à une vérification en tenant compte des champs obligatoires du formulaire. (A1)

### Scénario Alternatif

A1 : les champs du formulaire ne sont pas correctement renseignés  
Ce scénario se déclenche lorsque l'utilisateur ne renseigne pas correctement les champs.  
L'utilisateur est invité à reprendre  
Le scénario reprend au point 4 du scénario nominal

### Scénario d'exception

E1 : l'utilisateur annule la saisie  
Le scénario nominal est interrompu

### Post condition

L'arbre est créé

### 2.3.3. Diagramme de classes

Ce diagramme représente les différentes classes de notre système. Il détaille le contenu de chaque classe mais aussi les relations qui les lient entre elles.

Une classe est représentée par un rectangle séparé en trois (3) parties à savoir :

- La première partie contient le nom de la classe
- La seconde partie contient les attributs de la classe
- La troisième partie contient les méthodes appelées sur la classe

Le diagramme de classe de notre projet se schématise comme suit :

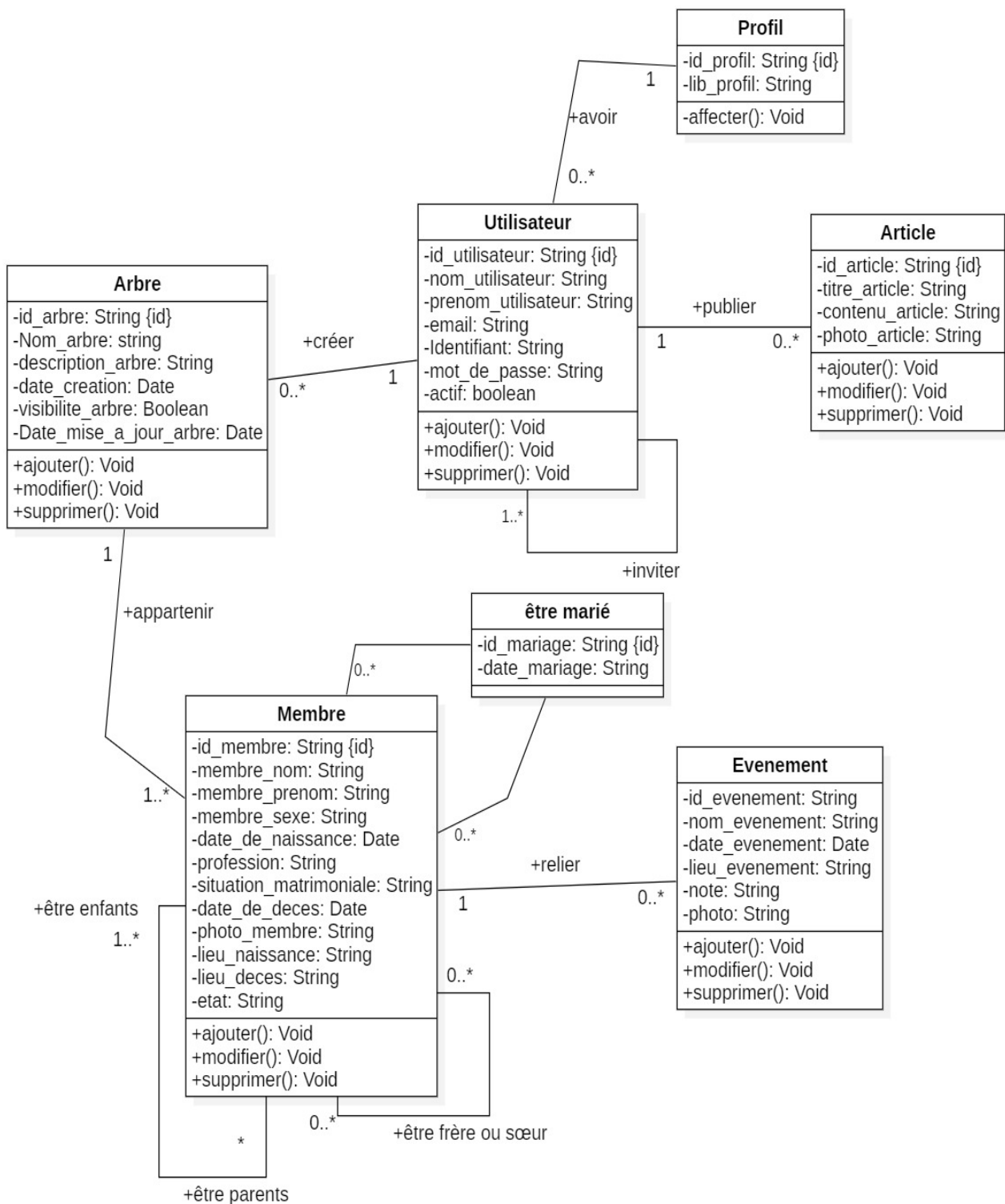


Figure 10 : Diagramme de classes de notre système

### 2.3.4. Diagramme de séquences

Le diagramme de séquence est une représentation graphique des interactions entre les acteurs d'un cas d'utilisation et le système selon un ordre chronologique. Il permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Voici quelques diagrammes de séquences de notre projet :

#### ❖ Authentification

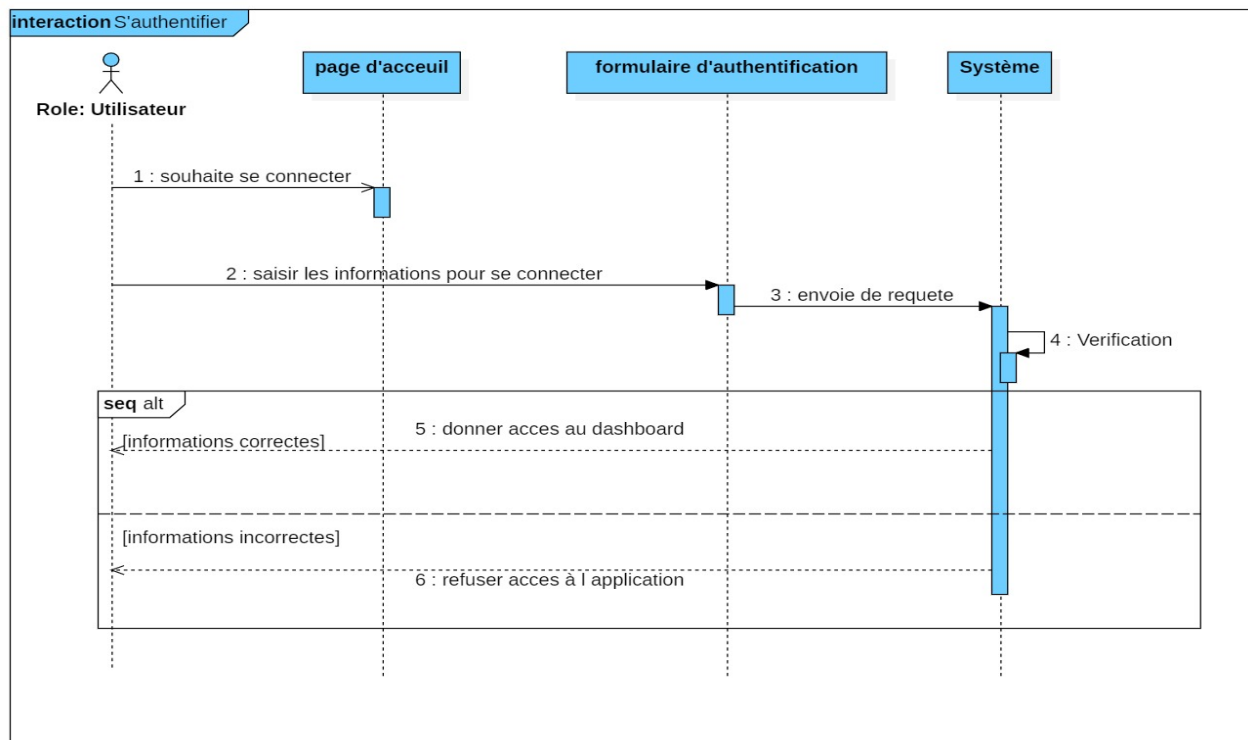


Figure 11 : Diagramme de séquence Authentification

## ❖ S'inscrire

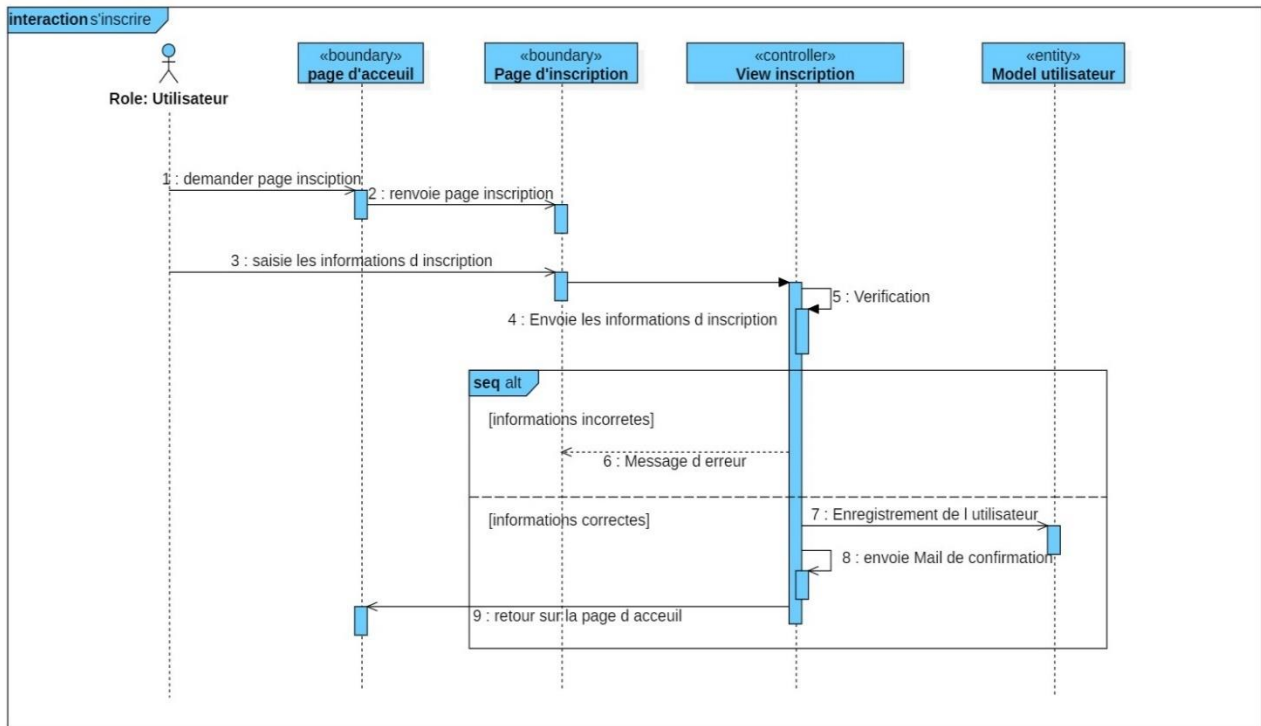


Figure 12 : Diagramme de séquence S'inscrire

## ❖ Créer un arbre

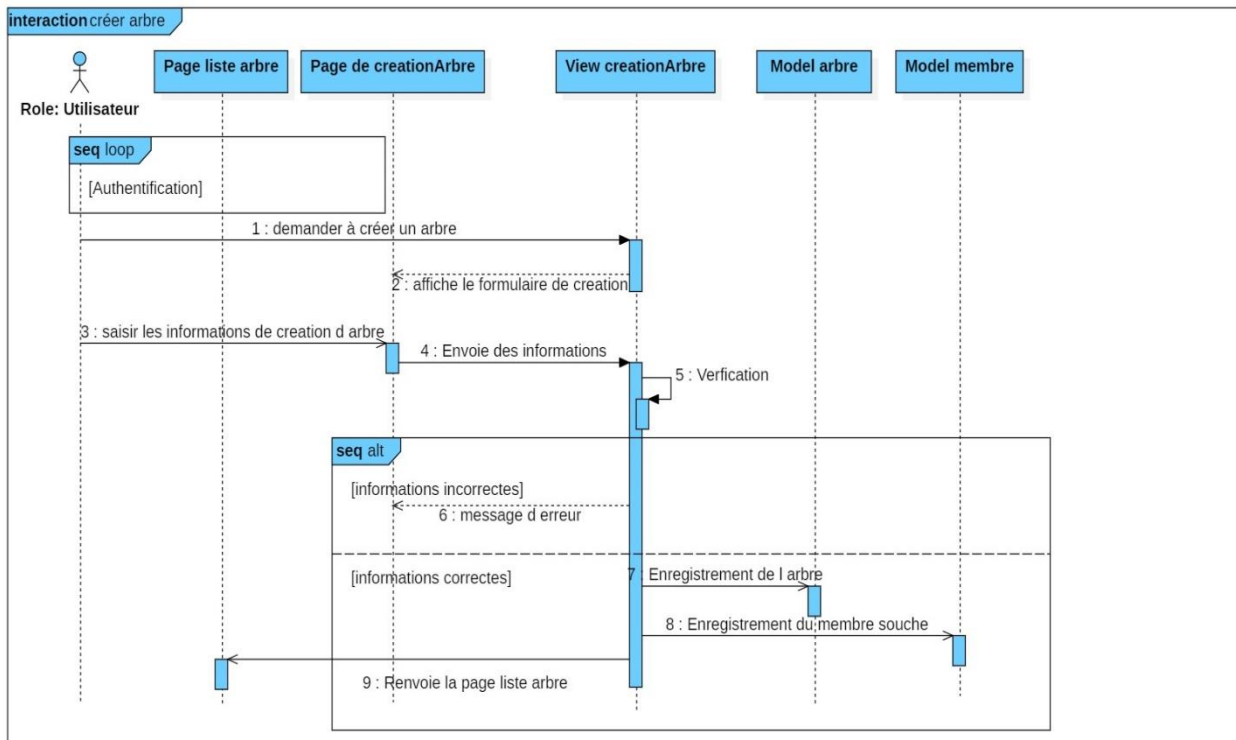


Figure 13 : Diagramme de séquence Créer un arbre

### 2.3.5. Diagramme d'activités

En UML, le diagramme d'activité est un diagramme qui permet de représenter le déclenchement d'événements en fonction des états du système et de modéliser les comportements. Il s'agit d'un organigramme qui décrit les enchaînements ou étapes dans le déroulement d'un cas d'utilisation ou d'un ensemble de cas d'utilisation.

Voici quelques diagrammes d'activités associés à notre projet :

#### ❖ Authentification

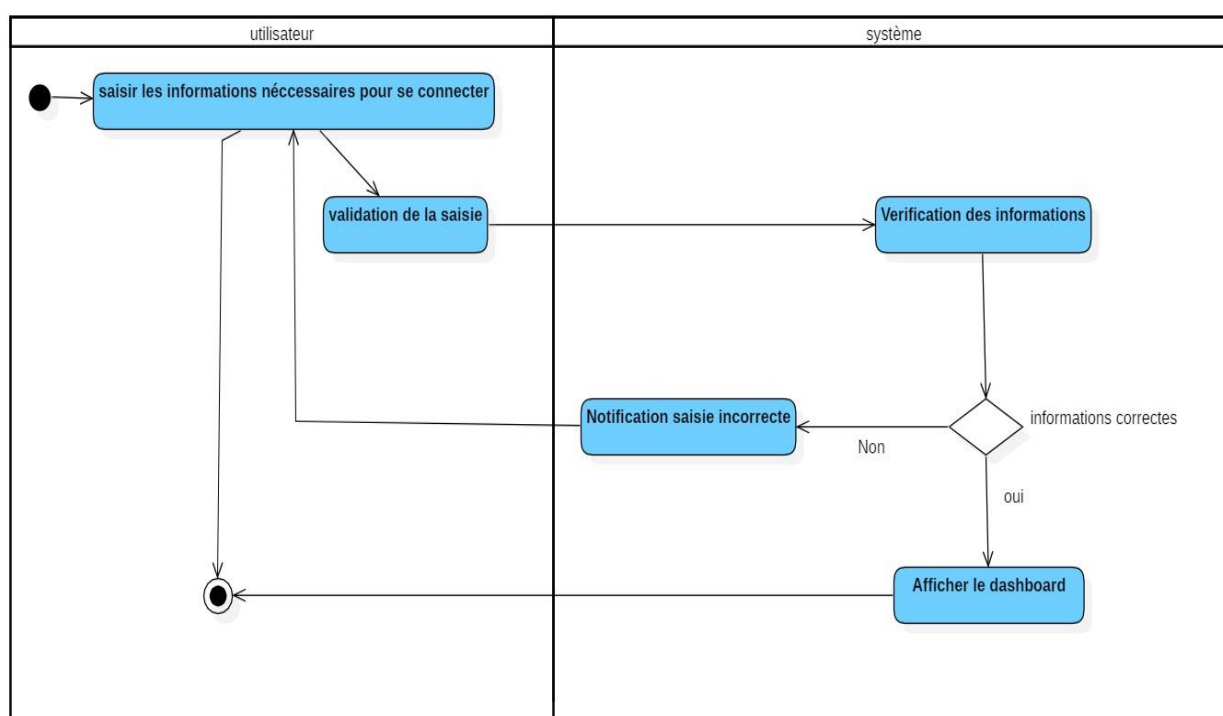


Figure 14 : Diagramme d'activité Authentification

#### ❖ S'inscrire



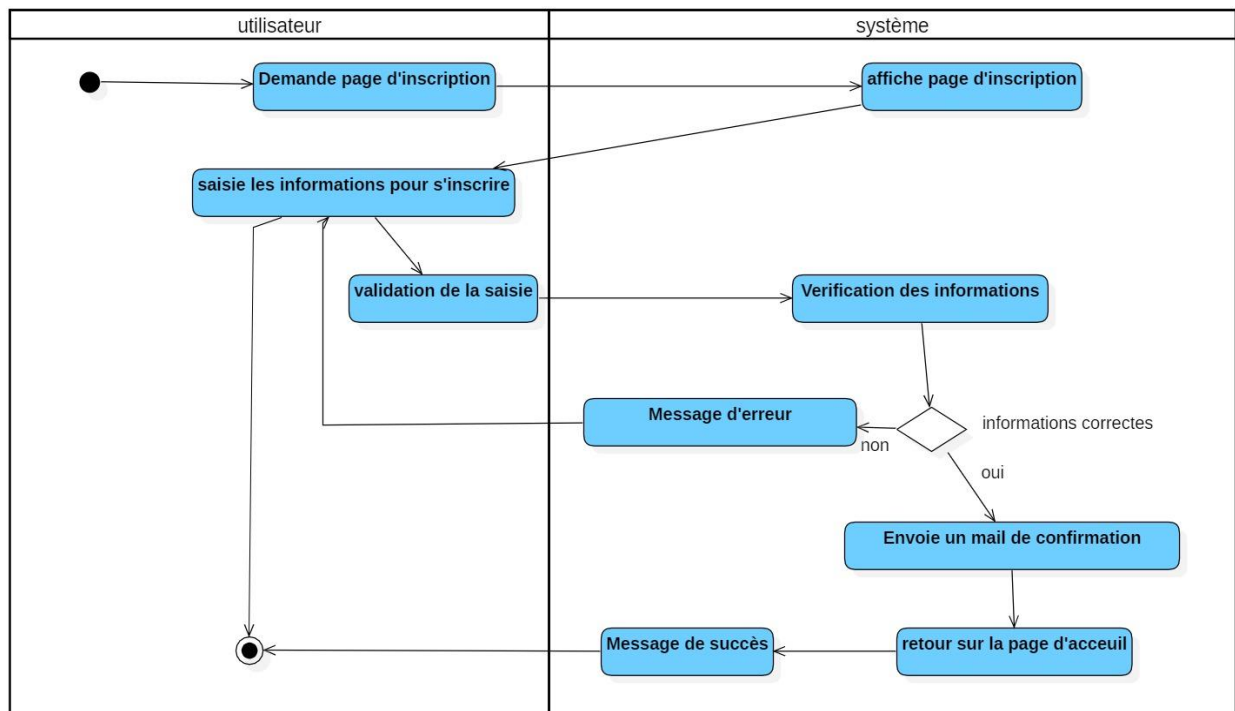


Figure 15 : Diagramme d'activité S'inscrire

### ❖ Créer un arbre

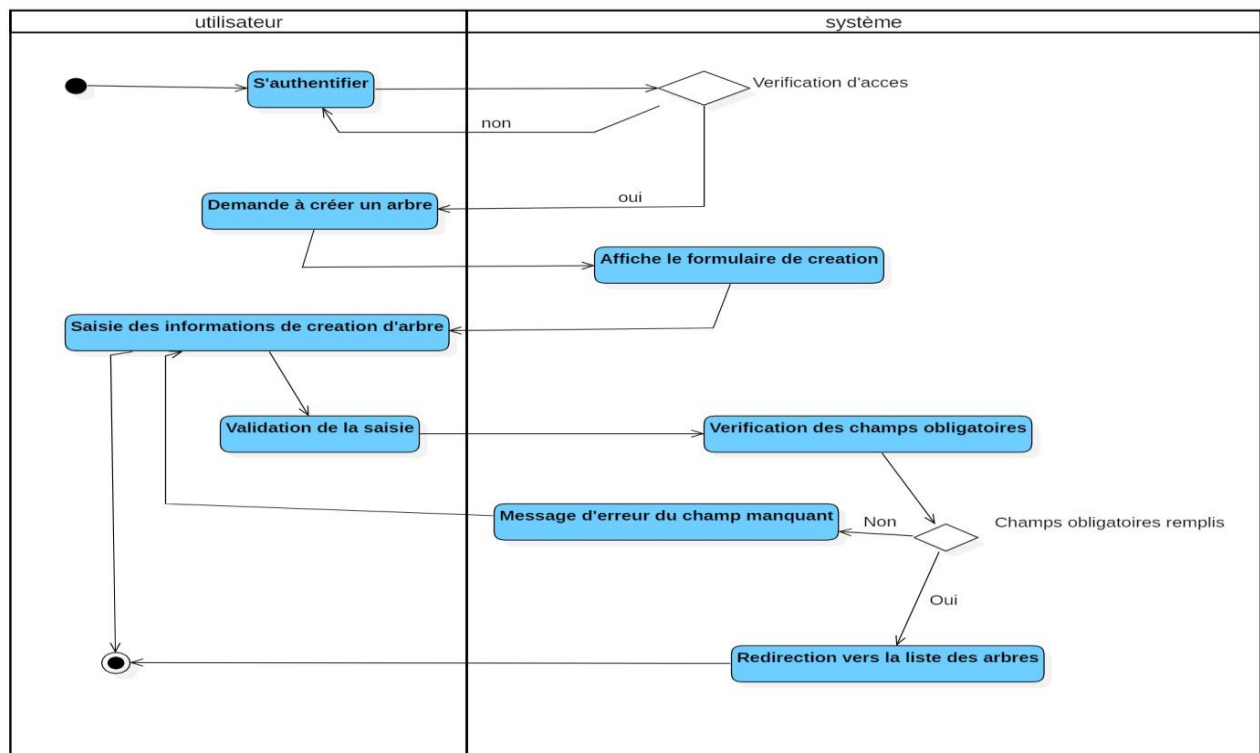


Figure 16 : Diagramme d'activité Créer un arbre

### 2.3.6. Diagramme de déploiement

Le diagramme de déploiement montre la disposition physique des différents matériels (les nœuds) qui entrent dans la composition d'un système et la répartition des instances de composant, processus et objet qui « vivent » sur ces matériels. Les diagrammes de déploiement sont donc très utiles pour modéliser l'architecture physique d'un système. Le diagramme de déploiement permet donc de modéliser l'architecture physique d'un système.

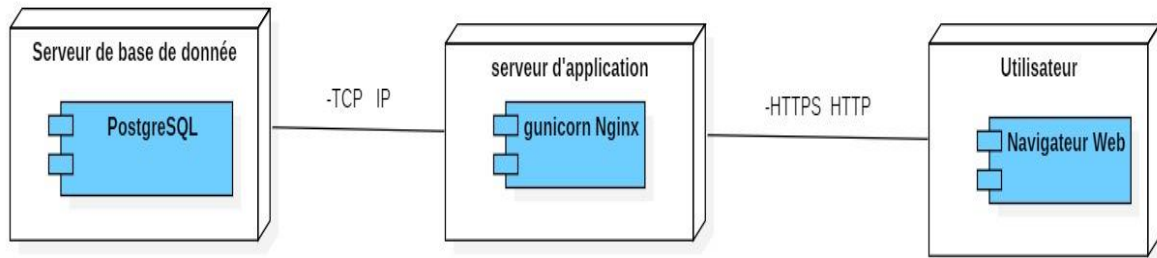


Figure 17 : Diagramme de déploiement du système

Le dossier de conception que nous venons de rédiger nous a permis d'identifier et de décrire les fonctionnalités du système à mettre en place. Ce dossier de conception va d'abord nous permettre de passer à la phase de réalisation et de mise en œuvre de notre projet qui consistera d'abord à faire le choix de nos outils de développement ; ensuite à définir ces outils de développement, à présenter une description de l'environnement technique et de l'architecture de l'application ; enfin à suivre la programmation proprement dite du logiciel.

PARTIE III : REALISATION ET MISE EN ŒUVRE

Après la présentation du cahier des charges et l'analyse et conception qui nous ont permis de déterminer les données à utiliser dans le domaine de notre étude et les traitements qui y sont effectués, nous présenterons dans cette partie la réalisation et la mise en œuvre de l'application.

### 3.1. MATÉRIELS ET LOGICIELS UTILISÉS

#### 3.1.1. Matériels utilisés

Pour mettre en œuvre notre projet nous avons utilisé comme matériel un ordinateur portable dont les caractéristiques sont :

ORDINATEUR	
Modèle	DELL
Processeur et fréquence	Intel(R) Core(TM) i7-6820HQ CPU @ 2.70GHz 2.70 GHz
Mémoire RAM	16 Go
Disque dur	256 Go SSD
Système d'exploitation	Windows 10 pro 64 bits, processeur x64

#### 3.1.2. Logiciels utilisés

Pour la réalisation de notre projet nous avons utilisé diverses technologies (Langage de programmation, outils et environnement de développement) dont nous parlons ci-dessous.

##### ❖ Python



Figure 18 : Logo de python

Python est un langage de programmation de haut niveau, interprété et à usage général. Guido van Rossum est un programmeur néerlandais connu pour être le créateur du langage de programmation Python. Sa philosophie de conception met l'accent sur la lisibilité du code avec l'utilisation d'une indentation significative. Python prend en charge plusieurs paradigmes de programmation à savoir la programmation structurée, l'orienté objet, fonctionnelle, procédurale

et réfléchissante. La plupart des implémentations Python incluent une boucle de lecture-évaluation-impression, ce qui leur permet de fonctionner comme un interpréteur de ligne de commande pour lequel les utilisateurs entrent des instructions séquentiellement et reçoivent les résultats immédiatement. Python est également livré avec un Environnement de Développement Intégré (IDE) appelé IDLE, qui est plus orienté vers les débutants. D'autres shells, notamment IDLE et IPython, ajoutent d'autres fonctionnalités telles que l'auto-complétion améliorées, la rétention de l'état de session et la coloration syntaxique. En plus des environnements de développement intégrés de bureau standard, il existe des IDE basés sur un navigateur Web, y compris SageMath, pour développer des programmes liés aux sciences et aux mathématiques. Python est supporté par les systèmes : Windows, Linux, Mac OS, Android.

### ❖ Django



Figure 19 : logo de django

Django est un framework Web Python de haut niveau qui encourage un développement rapide et une conception propre et pragmatique. Construit par des développeurs expérimentés de la Django Software Foundation<sup>6</sup>, il prend en charge le développement Web. C'est gratuit et open source. La dernière version de Django en utilisation est Django 4.1.

Bien que Python ait plusieurs autres frameworks web, nous avons choisi Django car :

- Le framework Django est composé d'un serveur léger et autonome pour le développement et les tests.
- Il offre un système d'authentification extensible
- Son interface d'administration est dynamique
- Un framework « Sites » qui permet à une installation de Django d'exécuter plusieurs sites Web, chacun avec son propre contenu et ses propres applications
- Un cadre pour la création des applications SIG<sup>7</sup> (Qui est le secteur d'activité principal du cabinet KALAMAR qui nous a servi de cadre de stage)

<sup>6</sup> Organisation à but non lucratif qui développe et maintient le framework Django

<sup>7</sup> Système d'Information Géographique

Les fichiers Django utilisent l'extension « .py »

### ❖ Visual Studio Code

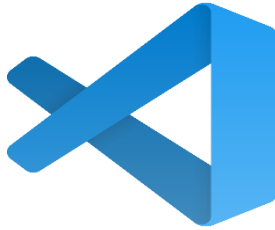


Figure 20 : logo de visual studio code

Visual Studio Code est un éditeur de code extensible développé par Microsoft qui prend en charge la majorité des langages de programmation communs et est développé pour les OS X, GNU/Linux et Windows. Il présente plusieurs avantages à savoir :

- Sélection et édition dans plusieurs sections de code en parallèle ;
- Sauvegarde automatique ;
- Multiplateforme ;
- Coloration syntaxique ;
- Visualisation de plusieurs fichiers au même moment ;
- Auto-complétions de code.

### ❖ Bootstrap



Figure 21 : logo de bootstrap

Bootstrap est un framework CSS gratuit et open source (sous licence MIT), collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. Il se base sur un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style aux différents composants d'une page web. Nous avons fait le choix de Bootstrap 4 car en plus de sa simplicité

d'utilisation, c'est l'un des projets Open Source les plus populaires sur la plateforme de gestion de développement GitHub.

#### ❖ HTML/CSS et Javascript



Figure 22 : Logo de HTML, CSS et JavaScript

HTML5 (HyperText Markup Language 5) est la dernière révision majeure d'HTML (format de données conçu pour représenter les pages web). C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). Les fichiers Html utilisent l'extension « .html »

Les CSS, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, tout type de page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus, dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les fichiers CSS utilisent l'extension « .css »

Le JavaScript est un langage de script incorporé dans un document HTML. Historiquement il s'agit même du premier langage de script pour le Web. Ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client, c'est-à-dire au niveau du navigateur et non du serveur web. Ainsi le langage JavaScript est fortement dépendant du navigateur appelant la page web dans laquelle le script est incorporé, mais en contrepartie il ne nécessite pas de compilateur, contrairement au langage Java, avec lequel il a longtemps été confondu. Les fichiers javascript utilisent l'extension « .js »

## ❖ Git



Figure 23 : logo de Git

Git est un logiciel décentralisé de gestion de versions. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes. Il nous a permis de mettre en place les versions des codes sources et documents de notre application. Pour notre développement, Git a été couplé avec Gitlab (décrit par la suite) pour permettre aux participants du projet de contrôler les versions et les codes.

## ❖ GitLab



Figure 24 : logo de GitLab

GitLab est la plate-forme DevOps<sup>8</sup> qui permet aux organisations de maximiser le retour global sur le développement de logiciels en fournissant des logiciels plus rapidement et efficacement, tout en renforçant la sécurité et la conformité. Avec GitLab, chaque équipe de votre organisation peut planifier, créer, sécuriser et déployer des logiciels en collaboration pour accélérer les résultats commerciaux avec une transparence, une cohérence et une traçabilité complètes. Il nous a permis de faciliter le développement et le suivi du projet par notre maître de stage.

<sup>8</sup> Ensemble de protique qui combine le développement (Dev) de logiciel et les opérations (Ops) informatiques



❖ **Google chrome**

Figure 25 : logo de Google Chrome

Google Chrome est un navigateur web propriétaire développé par Google. Avec son puissant inspecteur de code qui nous a permis de déboguer du JavaScript et aussi pour l'affichage de page web.

### **3.2. ARCHITECTURES MATÉRIELLES ET LOGICIELS DE L'APPLICATION**

#### **3.2.1. Architecture matérielle de l'application**

En informatique, l'architecture désigne la structure générale (inhérente à un système informatique), l'organisation des différents éléments du système (logiciels, matériels, humains, informations) et les relations entre ceux-ci. L'architecture matérielle d'un logiciel ou d'une application décrit l'ensemble des composants matériels supportant l'application. Ces composants peuvent être : des calculateurs, des postes de travail, des équipements de stockage, des équipements de sauvegarde, des équipements réseaux, etc. Pour mettre en œuvre notre application nous avons utilisé une architecture de type trois-tiers.

L'architecture trois tiers, aussi appelée architecture à trois niveaux ou architecture à trois couches, est l'application du modèle plus général qu'est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

- Couche de présentation qui correspond à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- Couche de traitement qui correspond à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- Couche d'accès aux données qui correspond aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Concrètement, le but d'une architecture 3-tiers est d'alléger la charge du client en déléguant la majeure partie du travail au serveur applicatif.

#### **3.2.2. Architecture logicielle de l'application**

Django est basé sur l'architecture MVT (Model-View-Template) traduit en français Modèle-Vue-Modèle. MVT peut être considéré comme une alternative au framework MVC qui se

compose d'un modèle, d'une vue et d'un contrôleur. Le contraste majeur entre MVT et MVC est que l'architecture Django prend en charge les responsabilités gérées par la partie contrôleur de l'architecture MVC. Django lui-même agit comme un contrôleur qui récupère une URL connectée à la partie vue de l'application et envoie éventuellement les réponses utilisateur respectives à l'application.

MVT comporte les trois parties suivantes :

**Modèle :** Dans Django, le modèle joue un rôle important en reliant l'ensemble de l'architecture à la base de données. Dans l'ensemble de ce processus, chaque classe de modèle représente une seule table dans la base de données. Les méthodes et les champs associés à ce processus sont déclarés dans un fichier distinct généralement défini comme `models.py`.

**Vue :** La logique globale de l'infrastructure de flux de données est définie dans cette partie de l'architecture. Le code entier est défini dans le fichier `views.py`. Une autre tâche du fichier d'affichage consiste à envoyer les réponses à l'utilisateur respectif chaque fois que l'application est utilisée.

Lorsqu'un fichier d'affichage est créé, la question suivante qui nous vient à l'esprit est de savoir comment lier ce fichier à notre application. Cette compréhension peut être établie en mappant les `views.py` dans un fichier `urls.py` distinct. `Urls.py` est responsable de garder une trace de toutes les pages distinctes et donc de cartographier chacune d'entre elles.

**Modèle :** Le framework Django gère et génère efficacement des pages Web HTML dynamiques visibles par l'utilisateur final. Django fonctionne principalement avec un backend donc, afin de fournir un frontend et de fournir une mise en page à notre application Web, nous utilisons des modèles. Les modèles sont la troisième et la partie visible de la structure MVT de Django par l'utilisateur. Notre modèle dans Django est écrit en HTML, CSS, Javascript et le langage de modèle Jinja2 qui est basé sur du texte et peut donc être utilisé pour générer des balises, des filtres et du code source. Le modèle est écrit dans un fichier « `.html` ». Il existe deux méthodes pour ajouter le modèle à notre application Web en fonction de nos besoins. Nous pouvons utiliser un seul répertoire de modèles qui sera réparti sur l'ensemble du projet, ou pour chaque application de notre projet, nous pouvons créer un répertoire de modèles différents.

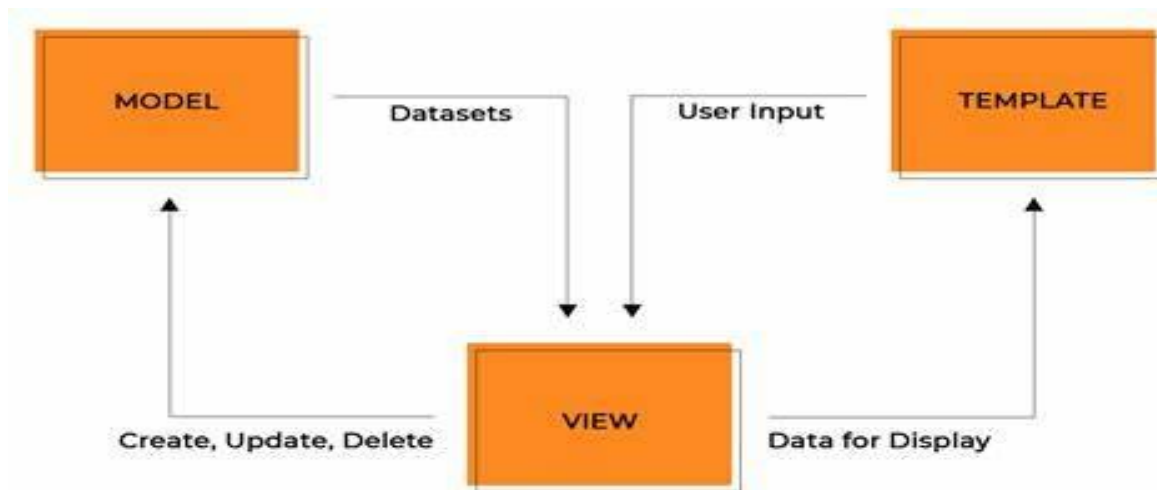


Figure 26 : Architecture MVT

### 3.3. SÉCURITÉ DE L'APPLICATION

La sécurité de l'application repose essentiellement sur les possibilités offertes d'une manière parfaitement ingénieuse par le Framework Django.

Le Framework Django fournit différents mécanismes pour sécuriser notre application. Certaines des fonctionnalités que nous avons utilisées sont listées ci-dessous :

#### ❖ Sécurité des mots de passes

Django fournit un système de stockage de mots de passe souple et emploie PBKDF2 par défaut. Il utilise l'algorithme PBKDF2 avec une fonction de hachage SHA256, un mécanisme d'étirement de mot de passe recommandé par l'Institut National des Normes et Technologies des États-Unis d'Amérique.

#### ❖ Les injections SQL

L'injection SQL est un type d'attaque où un utilisateur malveillant est capable d'exécuter du code SQL arbitraire sur une base de données. Il peut en résulter des suppressions d'enregistrements ou des divulgations de données.

Les jeux de requête de Django sont prémunis contre les injections SQL car leurs requêtes sont construites à l'aide de la paramétrisation des requêtes. Le code SQL d'une requête est défini séparément de ses paramètres. Comme ceux-ci peuvent provenir de l'utilisateur et donc non sécurisés, leur échappement est assuré par le pilote de base de données sous-jacent.

#### ❖ La validation de l'inscription

Après la création d'un compte utilisateur avant que le compte de l'utilisateur soit actif et que ce dernier puisse se connecter il doit aller cliquer sur un lien de confirmation qui est envoyé dans sa boîte Gmail.

#### ❖ Authentification

L'accès au site nécessite une authentification de la part de tous les utilisateurs.

### 3.4. MISE EN PLACE DE LA BASE DE DONNÉES

Django nous permet d'interagir avec ses modèles de base de données, c'est-à-dire d'ajouter, de supprimer, de modifier et d'interroger des objets, à l'aide d'une API d'abstraction de base de données appelée ORM (Object-Relational Mapping). Django possède un outil bien pratique pour gérer la structure d'une base de données : **migration**. C'est un outil en ligne de commande. Une migration est une classe qui va contenir la définition d'une table ou altération de la définition de cette table. Elle permet de créer et de mettre à jour un schéma de bases de données. Nous choisissons les migrations plutôt que la création grâce au script SQL car :

- Avec les migrations, nous avons une vue claire de l'altération du schéma de la base de données ;
- Elle permet de créer des tables avec leurs contraintes, les index ;
- Une migration écrite en Django peut créer des tables dans une base SQLite, MySQL, PostgreSQL, SQL Server ;
- Changer de SGBD lorsqu'on utilise les migrations ne cause donc plus de problèmes ;

Pour créer une migration sur Windows, il faut exécuter les commandes suivantes :

- « python manage.py makemigrations » pour créer une migration
- « python manage.py migrate » pour appliquer la migration

Voici quelques un des migrations :

- ❖ Migration de la table Arbre

```
class Migration(migrations.Migration):

    initial = True

    dependencies = [
        migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ]

    operations = [
        migrations.CreateModel(
            name='Arbre',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('nom_arbre', models.CharField(max_length=200, null=True)),
                ('description_arbre', models.TextField(null=True)),
                ('visibilite_arbre', models.BooleanField()),
                ('date_creation_arbre', models.DateTimeField(auto_now_add=True)),
                ('date_mise_a_jour_arbre', models.DateTimeField(auto_now=True)),
                ('id_user', models.ForeignKey(null=True, on_delete=django.db.models.deletion.CASCADE, to=settings.AUTH_USER_MODEL)),
            ],
        ),
    ]
```

Figure 27 : table Arbre

❖ Migration de la table Membre

```

class Migration(migrations.Migration):

    initial = True

    dependencies = [
        ('arbre', '0003_alter_arbre_id_user'),
    ]

    operations = [
        migrations.CreateModel(
            name='Membre',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('nom_membre', models.CharField(blank=True, max_length=200, null=True, verbose_name='Nom du membre')),
                ('membre_prenom', models.CharField(blank=True, max_length=200, null=True, verbose_name='prénom du membre')),
                ('membre_sexe', models.CharField(choices=[('female', 'Féminin'), ('male', 'Masculin')], default='female', max_length=10, verbose_name='sexe')),
                ('date_naissance_membre', models.DateField(blank=True, null=True, verbose_name='Date de naissance du membre')),
                ('lieu_naissance_membre', models.CharField(blank=True, max_length=200, null=True, verbose_name='Lieu de naissance du membre')),
                ('profession_membre', models.CharField(blank=True, max_length=200, null=True, verbose_name='profession du membre')),
                ('note_membre', models.TextField(blank=True, null=True, verbose_name='Note')),
                ('source_membre', models.CharField(blank=True, help_text="source d'information du membre", max_length=250, null=True, verbose_name='source dinfos')),
                ('situation_matrimoniale_membre', models.CharField(max_length=250, null=True)),
                ('nombre_union_membre', models.IntegerField(blank=True, null=True, verbose_name='nombre mariage du membre')),
                ('date_deces_membre', models.DateField(blank=True, null=True, verbose_name='Date de décès du membre')),
                ('lieu_deces_membre', models.CharField(blank=True, max_length=300, null=True, verbose_name='Lieu de décès du membre')),
                ('etat_membre', models.CharField(choices=[('vivant', 'Vivant(e)'), ('decede', 'Décédé(e)'], default='vivant', max_length=10, null=True)),
                ('photo', models.ImageField(blank=True, null=True, upload_to='photo_membre/', verbose_name='photo du membre')),
                ('createdDate', models.DateTimeField(auto_now=True)),
                ('updatedDate', models.DateTimeField(auto_now_add=True)),
                ('id_arbre', models.ForeignKey(null=True, on_delete=django.db.models.deletion.CASCADE, related_name='id_arbre', to='arbre.arbre')),
                ('membre_maris', models.ManyToManyField(blank=True, to='membre.membre')),
                ('membre_mere', models.ForeignKey(blank=True, null=True, on_delete=django.db.models.deletion.CASCADE, related_name='mere_du_membre', to='membre.membre')),
                ('membre_pere', models.ForeignKey(blank=True, null=True, on_delete=django.db.models.deletion.CASCADE, to='membre.membre')),
            ],
        ),
    ],
)

```

Figure 28 : Table Membre

## ❖ Migration de la table Évènement

```
class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='Evenement',
            fields=[
                ('id', models.BigAutoField(auto_created=True, primary_key=True, serialize=False, verbose_name='ID')),
                ('nom_evenement', models.CharField(max_length=200)),
                ('date_ajout_evenement', models.DateField()),
                ('lieu_evenement', models.CharField(max_length=300)),
                ('image', models.ImageField(blank=True, null=True, upload_to='event_medias/')),
                ('note', models.TextField()),
            ],
        ),
    ]
```

Figure 29 : Table Évènement

## PARTIE IV : GUIDE D'EXPLOITATION

La partie précédente, relative à la phase de réalisation et de mise en œuvre, nous a permis de mettre en œuvre le système. Une fois le système réalisé, il doit être déployé afin de permettre son utilisation. La présente partie vise alors à décrire la procédure de déploiement et de suivi de l'application, avec les configurations matérielles et logicielles nécessaires. Cette partie consistera en un guide pour montrer les démarches du déploiement et du suivi du système développé.

## 4.1. CONFIGURATION LOGICIELLE ET MATÉRIELLE

### 4.1.1. Configuration matérielle

Les caractéristiques matérielles dont nous devons disposer pour la mise en exploitation de notre application sont les suivantes :

Tableau 13 : configuration matérielle

Appareil	Configuration requise
Poste utilisateur	Processeur : Core i3, 1.6 GHz ou plus Disque Dur : 120 Go Mémoire RAM : 2 Go Port RJ45 ou carte réseau sans fil : Actif
Serveur d'application	4 Processeurs, 3.6 GHz Stockage : 160 Go Mémoire RAM : 16 Go Bande passante : 2To de bande passante

### 4.1.2. Configuration logicielle

Tableau 14 : configuration logicielle

Appareil	Configuration requise
Poste utilisateur	Navigateur : Mozilla, Chrome, Microsoft Edge OS : Tout type de système d'exploitation
Serveur d'application	Gunicorn et Nginx OS : Linux
Serveur de base de données	PostgreSQL 12 ou plus OS : Linux



## 4.2. DÉPLOIEMENT ET SUIVI

### 4.2.1. Déploiement

Dans cette partie nous parlerons du déploiement de notre application et quels sont les apports de Django pour nous faciliter cette tâche.

Django étant par nature un cadriciel Web a besoin d'un serveur Web pour fonctionner. Et comme la plupart des serveurs Web ne communiquent pas nativement avec Python, il y a besoin d'une interface pour permettre cette communication.

Django prend actuellement en charge deux interfaces : WSGI et ASGI.

- WSGI (Web Server Gateway Interface) est le standard Python principal pour la communication entre les serveurs Web et les applications, mais il ne gère que le code synchrone.
- ASGI (Asynchronous Server Gateway Interface) est le nouveau standard orienté sur le code asynchrone qui permet à un site Django de bénéficier des fonctionnalités asynchrones de Python et de Django au fur et à mesure de leur disponibilité.

Pour notre déploiement nous avons décidé de choisir la plate-forme principale de déploiement de Django qui est WSGI. La commande de gestion « **startproject** » de Django définit par défaut une configuration WSGI minimale pour nous, que nous pouvons ensuite adapter aux besoins de notre projet ; cette configuration est alors utilisable par tout serveur d'applications se conformant au standard WSGI.

Le concept-clé dans le déploiement avec WSGI est l'objet exécutable **application** que le serveur d'applications utilise pour communiquer avec notre code. Ceci est généralement fourni par un objet nommé **application** dans un module Python accessible par le serveur.

La commande « **startproject** » crée un fichier `<nom_de_projet>/wsgi.py` qui contient l'objet exécutable **application**.

```

core > wsgi.py > ...
1  """
2  WSGI config for core project.
3
4  It exposes the WSGI callable as a module-level variable named ``application``.
5
6  For more information on this file, see
7  https://docs.djangoproject.com/en/3.2/howto/deployment/wsgi/
8  """
9
10 import os
11
12 from django.core.wsgi import get_wsgi_application
13
14 os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'core.settings')
15
16 application = get_wsgi_application()
17
    
```

Figure 30 : objet application

### ❖ Déploiement de notre application avec Gunicorn et Nginx

Django est doté d'un serveur qui nous permet d'exécuter notre application en développement avec la commande « py manage.py runserver ». En production ce serveur est remplacé par le serveur WSGI Gunicorn et le serveur qui va s'occuper des fichiers statiques et multimédia de notre application à savoir Nginx.

Liste des contrôles à faire avant le déploiement :

### ❖ Gardé secret la clé utilisée en production

Au lieu d'inscrire statiquement la clé secrète dans notre module de réglages, nous avons envisagé de la charger à partir d'une variable d'environnement.

```

import os
SECRET_KEY = os.environ['SECRET_KEY']
    
```

Figure 31 : Clé secrète en production

### ❖ Désactiver le mode Debug en production

Le mode Debug est activé lors du développement pour nous renseigner sur la source des erreurs qui peuvent arriver, mais dans un environnement de production, c'est une très mauvaise idée, parce que cela révèle de nombreuses informations sur notre projet : extraits de code source, variables locales, réglages, bibliothèques utilisées, etc.

```
# SECURITY WARNING: don't run with debug turned on in production!
|
DEBUG = False
```

Figure 32 : Debug à False

### ❖ Unicorn et Nginx

Gunicorn est destiné à un environnement UNIX et est incompatible avec Windows. Pour pouvoir faire notre déploiement nous avons installé le terminal Ubuntu disponible sur Windows.

#### Première étape

```
sudo apt-get update
sudo apt-get install python3-pip python3-dev libpq-dev postgresql postgresql-contrib nginx
```

Cela installera , les fichiers de développement Python nécessaires pour construire Unicorn plus tard, le système de base de données PostgreSQL et les bibliothèques nécessaires pour interagir avec lui, et le serveur Web Nginx.

```
gedeon@DESKTOP-87QK3QK: ~
277 packages can be upgraded. Run 'apt list --upgradable' to see them.
gedeon@DESKTOP-87QK3QK:~$ sudo apt-get install python3-pip python3-dev libpq-dev postgresql postgresql-contrib nginx
[sudo] password for gedeon:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-10-base gcc-9 gcc-9-base
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6 libc6-dev libc6i-0
  libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1 libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-9-dev libgcc-s1 libgd3
  libgomp1 libisl22 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libllvm10 liblsan0 libmpc3 libnghttp3 libnghttp3-dev libnghttp3-minimal libnghttp3-stream
  libnghttp3-stream-dev libnghttp3-stream-minimal libnghttp3-stream-minimal-dev libnghttp3-stream-minimal-static libnghttp3-stream-minimal-static-dev
  libpython3.8-stdlib libquadmath0 libstdc++-9-dev libstdc++6 libtiff5 libtsan0 libubsan1 libwebp6 linux-libc-dev make manpages-dev nginx-common
  nginx-core postgresql-12 postgresql-client-12 postgresql-client-common postgresql-common python-pip-whl python3-wheel python3.8 python3.8-dev
  python3.8-minimal ssl-cert sysstat zlib1g zlib1g-dev
Suggested packages:
  binutils-doc cpp-doc gcc-9-locales debian-keyring g++-9-multilib gcc-9-doc gcc-multilib autoconf automake libtool flex bison gdb
  gcc-doc gcc-9-multilib libc-dev-bin libstdc++-9-dev libstdc++-9-doc make-doc fcgiwrap nginx-doc postgresql-doc libjson-perl
  python3.8-venv python3.8-doc binfmt-support openssl-blacklist isag
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-10-base gcc-9 gcc-9-base
  libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libc6i-0
  libcrypt-dev libctf-nobfd0 libctf0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl libgcc-9-dev libgcc-s1 libgd3
  libgomp1 libisl22 libitm1 libjbig0 libjpeg-turbo8 libjpeg8 libllvm10 liblsan0 libmpc3 libnghttp3 libnghttp3-dev libnghttp3-minimal libnghttp3-stream
  libnghttp3-stream-dev libnghttp3-stream-minimal libnghttp3-stream-minimal-dev libnghttp3-stream-minimal-static libnghttp3-stream-minimal-static-dev
  libpython3.8-stdlib libquadmath0 libstdc++-9-dev libstdc++6 libtiff5 libtsan0 libubsan1 libwebp6 linux-libc-dev make manpages-dev nginx-common
  nginx-core postgresql-12 postgresql-client-12 postgresql-client-common postgresql-common python-pip-whl python3-wheel python3.8 python3.8-dev
  python3.8-minimal ssl-cert sysstat zlib1g zlib1g-dev
The following packages will be upgraded:
  gcc-10-base libc6 libexpat1 libgcc-9-dev libpython3.8 libpython3.8-minimal libpython3.8-stdlib libstdc++6 python3.8 python3.8-minimal zlib1g
11 upgraded, 74 newly installed, 0 to remove and 266 not upgraded.
Need to get 93.9 MB of archives.
After this operation, 355 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 gcc-10-base amd64 10.3.0-1ubuntu1~20.04 [20.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libgcc-9-dev amd64 10.3.0-1ubuntu1~20.04 [41.8 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libstdc++6 amd64 10.3.0-1ubuntu1~20.04 [501 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libc6 amd64 2.31-0ubuntu9.9 [2722 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 python3.8 amd64 3.8.10-0ubuntu1~20.04.5 [387 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libpython3.8 amd64 3.8.10-0ubuntu1~20.04.5 [1625 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libpython3.8-stdlib amd64 3.8.10-0ubuntu1~20.04.5 [1675 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 python3.8-minimal amd64 3.8.10-0ubuntu1~20.04.5 [1905 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libpython3.8-minimal amd64 3.8.10-0ubuntu1~20.04.5 [717 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libexpat1 amd64 2.2.9-1ubuntu0.4 [74.4 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 zlib1g amd64 1:1.2.11.dfsg-2ubuntu1.3 [53.8 kB]
Get:12 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils-common amd64 2.34-6ubuntu1.3 [207 kB]
Get:13 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libbinutils amd64 2.34-6ubuntu1.3 [474 kB]
Get:14 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libctf-nobfd0 amd64 2.34-6ubuntu1.3 [47.4 kB]
Get:15 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libctf0 amd64 2.34-6ubuntu1.3 [46.6 kB]
Get:16 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils-x86-64-linux-gnu amd64 2.34-6ubuntu1.3 [1613 kB]
Get:17 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 binutils amd64 2.34-6ubuntu1.3 [3380 B]
```

Figure 33 : installation de python, PostgreSQL, Nginx

#### Deuxième étape

Nous allons créer la base donnée de notre projet, créer un utilisateur et lui donner les droits d'accès à la base de données.

```
gedeon@DESKTOP-87QK3QK:~$ sudo -u postgres psql
psql (12.11 (Ubuntu 12.11-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# CREATE DATABASE mifa;
CREATE DATABASE
postgres=# CREATE USER gedeon WITH PASSWORD 'gedeon';
CREATE ROLE
postgres=# ALTER ROLE gedeon SET client_encoding TO 'utf8';
ALTER ROLE
postgres=# ALTER ROLE gedeon SET default_transaction_isolation TO 'read committed';
ALTER ROLE
postgres=# ALTER ROLE gedeon SET timezone TO 'UTC';
ALTER ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE mifa TO gedeon;
GRANT
postgres=# \q
gedeon@DESKTOP-87QK3QK:~$
```

Figure 34 : création de la base de données et d'un utilisateur

### Troisième étape

Avec notre environnement virtuel actif, installons Gunicorn et l'adaptateur PostgreSQL avec l'instance locale de psycopg2.

```
pip install gunicorn psycopg2
```

```
root@DESKTOP-87QK3QK: ~/kalamar/famille
GNU nano 4.8 /etc/systemd/system/gunicorn.service
[Unit]
Description=gunicorn daemon
After=network.target

[Service]
User=root
Group=www-data
WorkingDirectory=/root/kalamar
ExecStart=/root/kalamar/venv/bin/gunicorn --access-logfile - --workers 3 --bind unix:/root/kalamar/famille/core.sock core.wsgi:application

[Install]
WantedBy=multi-user.target
```

Figure 35 : fichier de configuration de service gunicorn

Nous avons testé la capacité de gunicorn à servir notre projet, nous obtenons le résultat en dessous sur l'image. Les fichiers statistiques n'étant pas encore gérés par le serveur Nginx.

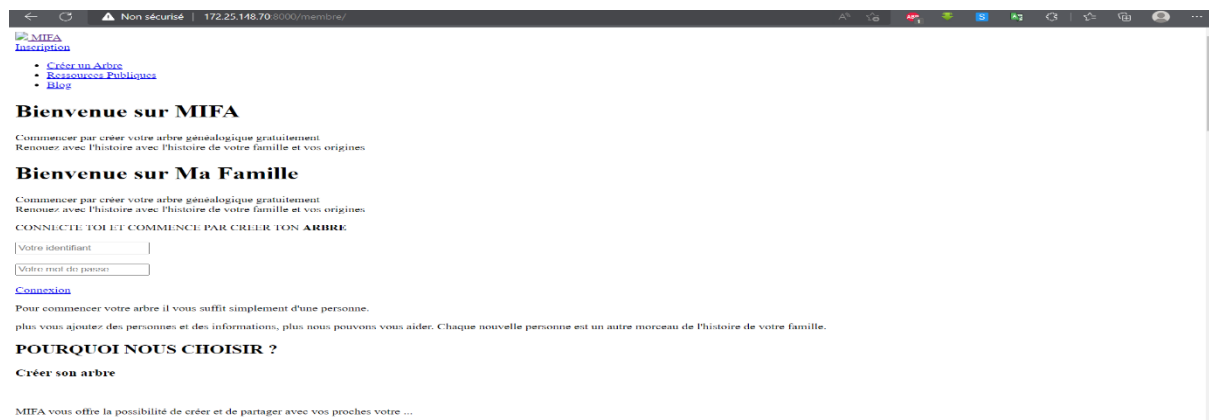


Figure 36 : test de gunicorn

Nous allons collecter tout le contenu statique dans l'emplacement du répertoire que nous avons configuré en tapant :

Python manage.py collectstatic

```
root@DESKTOP-87QK3QK: ~/kalamar/famille
GNU nano 4.8 /etc/nginx/sites-available/core
server {
    listen 80;
    server_name 192.168.1.80 DESKTOP-87QK3QK.me ;

    location /static/ {
        root /root/kalamar/famille;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/root/kalamar/famille/core.sock;
    }
}
```

Figure 37 : fichier de configuration du serveur nginx

#### Quatrième étape

Nous allons redémarrer nos serveurs pour appliquer les modifications.

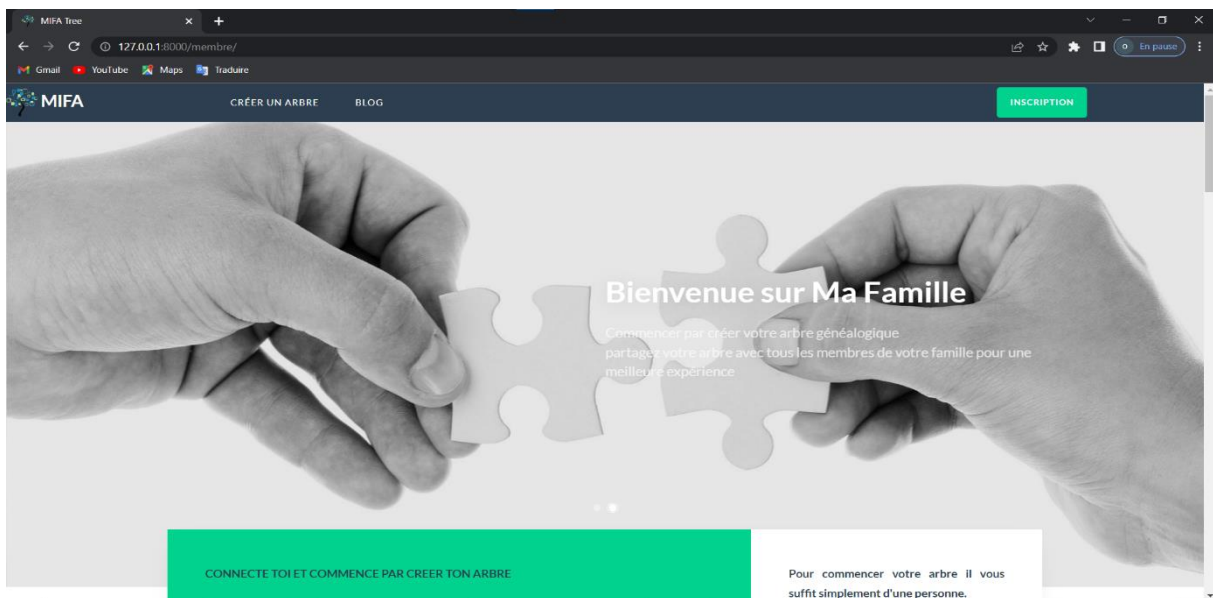


Figure 38 : application déployée

#### ❖ Fiche des événements

Tableau 15 : fiche des évènements

Date et heure	Libellé d'erreur	Description	Solution apportée
22/07/2022 09h20	ModuleNotFoundError: No module named 'fcntl'	Gunicorn ne trouve pas le module nommé 'fcntl' pour permettre le déploiement.	Le module recherché se trouve seulement sur les systèmes UNIX. Pour notre déploiement nous avons installé un terminal Ubuntu pour pouvoir utiliser Gunicorn.
22/07/2022 11h05	Error : 404 de Nginx	Le serveur Nginx est inaccessible	Correction du port d'accès au serveur comme déclaré dans le fichier de configuration nginx.

#### 4.2.2. Suivi

Le suivi d'une application après son déploiement permet d'éviter au cours de son utilisation des problèmes qui pourraient entraîner une indisponibilité de l'application. Les principales actions à mener pour le suivi d'une application sont le suivi des bugs et la gestion de versions.

##### 4.2.2.1. Suivi des bugs

Kalamar dispose à l'interne des techniciens intervenant en cas de bugs ou de dysfonctionnements des applications.

##### 4.2.2.2. Versionning du projet

Un gestionnaire de versions est un système qui enregistre l'évolution d'un fichier ou d'un ensemble de fichiers d'un projet au cours du temps de manière qu'on puisse rappeler une version antérieure d'un fichier ou du projet tout entier. Utiliser un gestionnaire de versions nous permet de suivre les différents changements qui ont été effectués et surtout, nous garantir qu'on puisse revenir à un état « stable » des fichiers à tout moment. Pour notre projet, nous avons utilisé le logiciel Git avec le serveur de gestion de développement logiciel GitLab (tous présentés dans la partie Réalisation et mise en œuvre). La documentation qui suivra présentera le guide d'utilisation de notre application et les actions à mener en cas d'erreur.

#### 4.2.2.3. Sauvegarde de données

Dans le domaine informatique, le processus de sauvegarde et de restauration désigne une combinaison de technologies et de pratiques qui permettent de faire des copies des données importantes dans des périodes prédéterminées. Ce processus implique généralement la copie des données de sauvegarde de leur emplacement d'origine vers un nouvel emplacement, de sorte que les fichiers sauvegardés puissent être utilisés à la place de ceux qui ont été perdus. Les sauvegardes peuvent empêcher la perte permanente de données et éviter des coûts de récupération élevés en cas de catastrophe, d'erreur humaine ou de cyber-attaque. Notre application nécessite une sauvegarde importante des données car elle se propose de conserver les données de ses utilisateurs afin de leur permettre une utilisation sur le long terme.

Django nous donne la possibilité de faire des sauvegardes entières ou partiel de notre base de données en utilisant les commandes suivantes :

- `./manage.py dumpdata > db.json` (tout le contenu de la base de données)
- `./manage.py dumpdata arbre > admin.json` (tout contenu de l'application arbre)

#### 4.3. MAINTENANCE : ACTIONS À MENER EN CAS DE CERTAINES ERREURS

Les erreurs les plus courantes sont entre autres les exceptions de type :

- ❖ **403** : L'utilisateur n'est pas autorisé à accéder à la page à laquelle il tente d'accéder. Il peut contacter l'administrateur pour plus d'explication ;
- ❖ **404** : L'utilisateur tente d'accéder à une URL non disponible dans le contexte de l'application ;
- ❖ **500** : Erreur interne du serveur. Pour ce cas d'erreur, l'utilisateur doit actualiser la page ; et si cela persiste, il devra informer l'administrateur.

PARTIE V : GUIDE D'UTILISATION

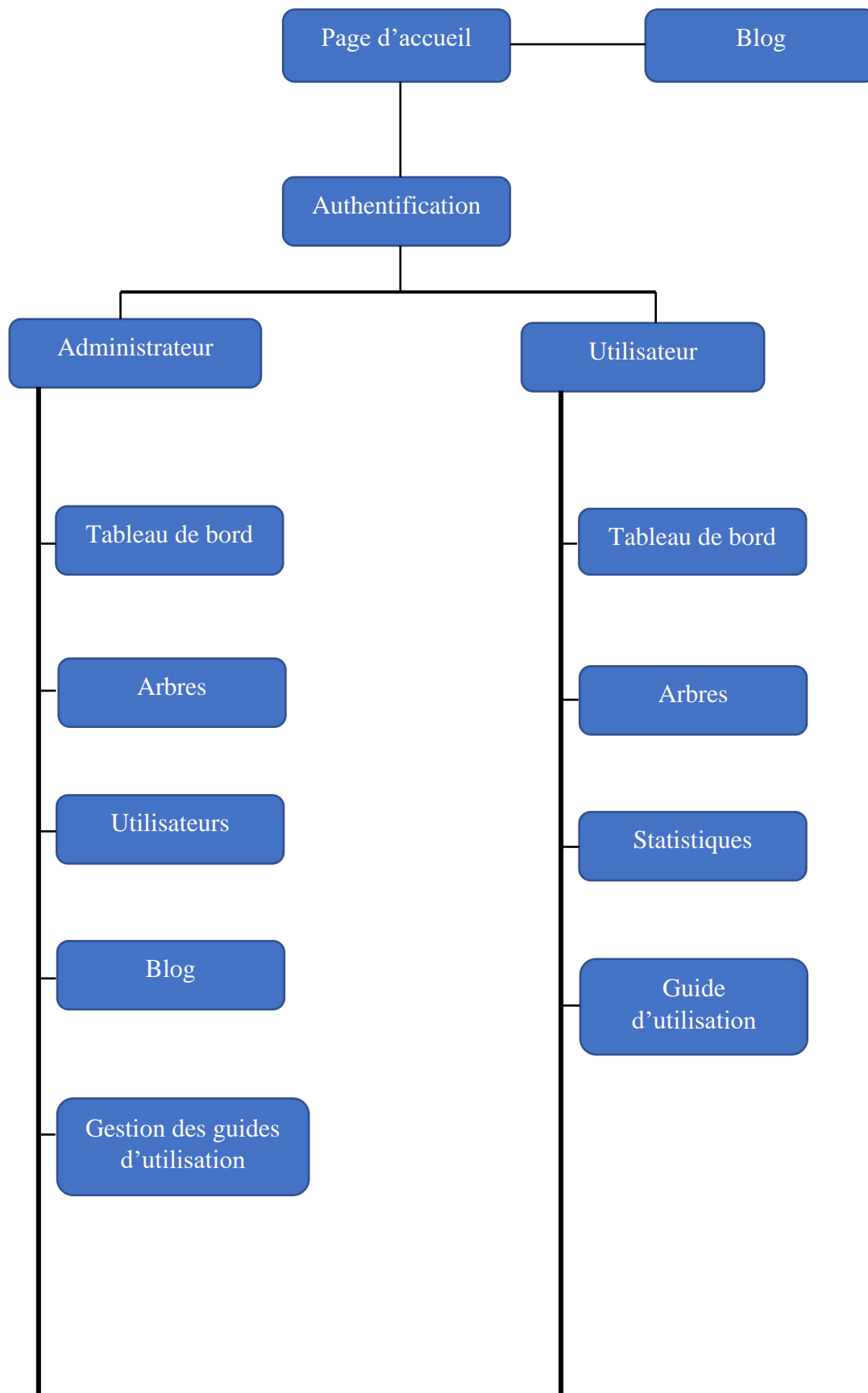


Le guide d'utilisation ou mode d'emploi est un document expliquant le fonctionnement d'un logiciel ou d'un objet en général. Concrètement, il permet aux utilisateurs de prendre en main le logiciel, de se familiariser à son fonctionnement et son interface. Il sera également très utile pour les formateurs dans le cadre de la formation des utilisateurs.

### **5.1. DESCRIPTION TEXTUELLE DU DOCUMENT**

Notre application est une application web destinée à la création d'arbre généalogique. Elle doit être capable de proposer une interface pour la création d'arbre, permettre d'ajouter les membres de sa famille sur l'arbre, permettre de partager l'arbre avec des membres de sa famille et de pouvoir imprimer le rendu de son arbre.

## 5.2. PLAN DE NAVIGATION



## 5.3. PRÉSENTATION DES DIFFÉRENTES INTERFACES DE L'APPLICATION

### 5.3.1. Page d'accueil

Cette page est notre page d'accueil, elle est la première page qui s'affiche. Elle communique des informations sur l'importance et l'utilité de créer un arbre généalogique et permet aussi aux utilisateurs inscrits de se connecter.

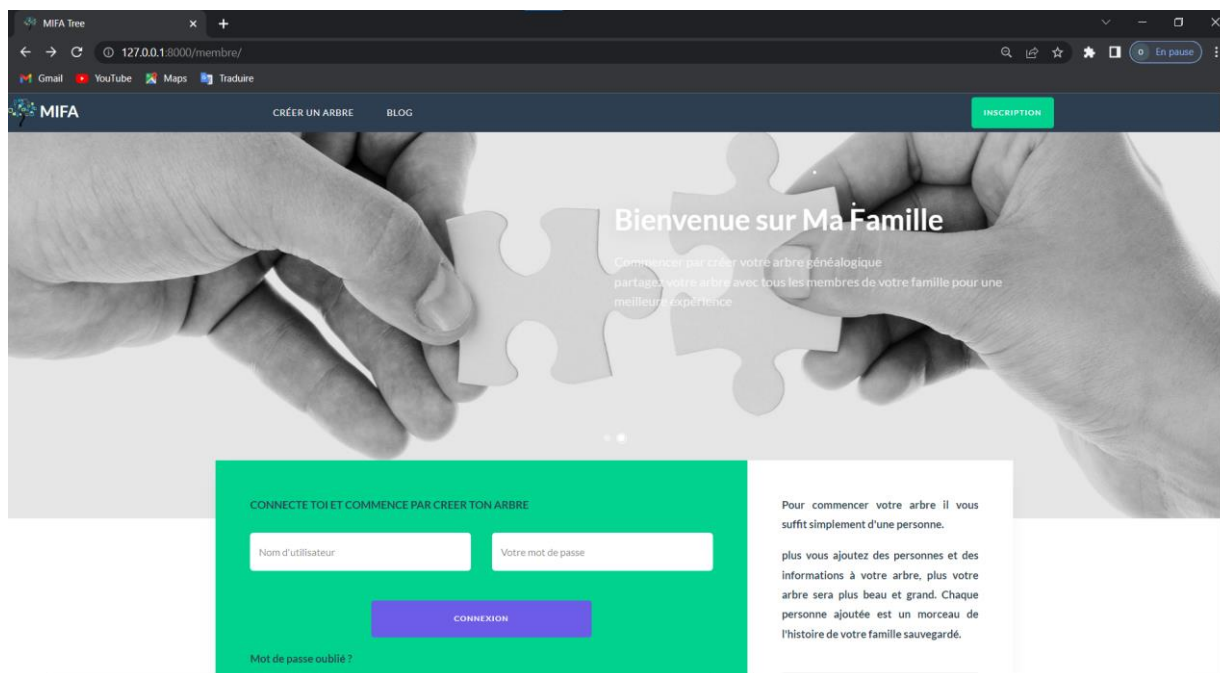


Figure 39 : page d'accueil

### 5.3.2. Page liste des arbres

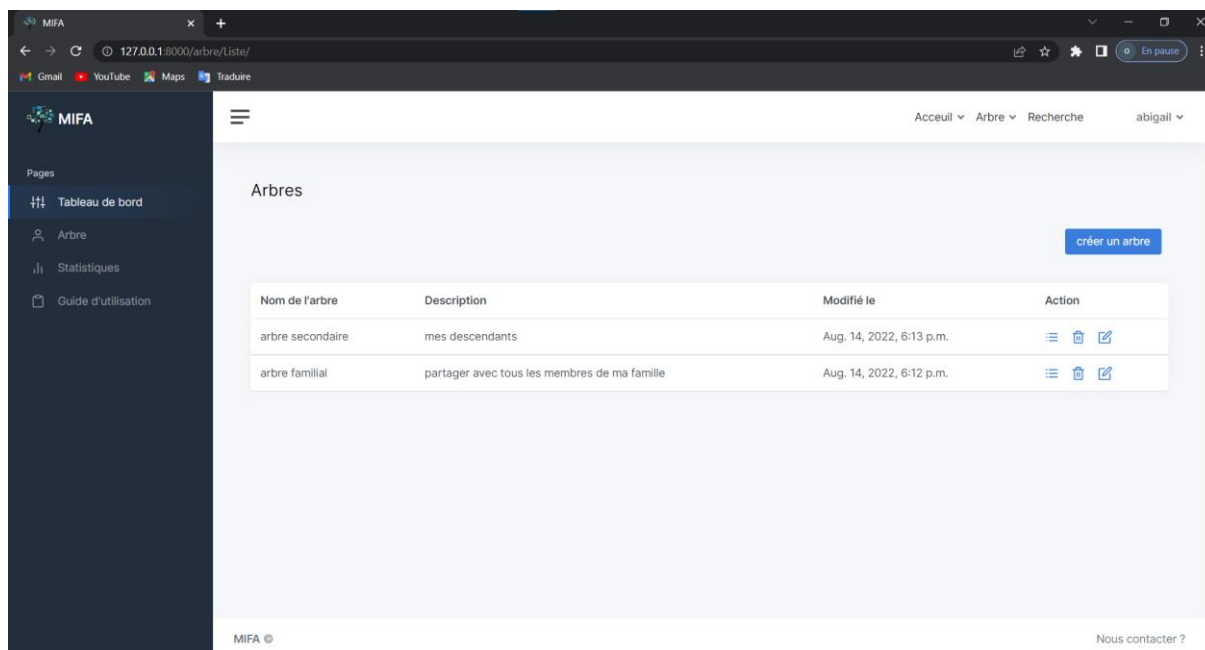


Figure 40 : page liste des arbres

### 5.3.3. Page détail d'un arbre

Cette page nous permet de voir la liste des membres de notre arbre et d'ajouter d'autres membres.

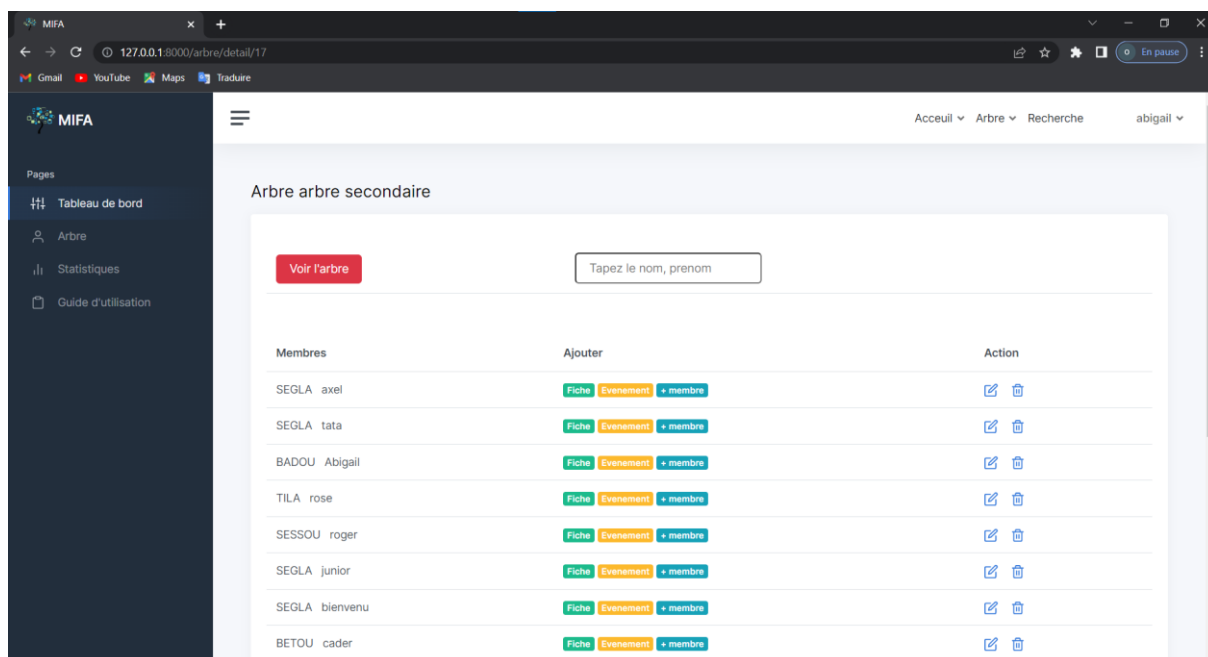


Figure 41 : page détail d'un arbre

### 5.3.4. Page présentation visuelle d'un arbre

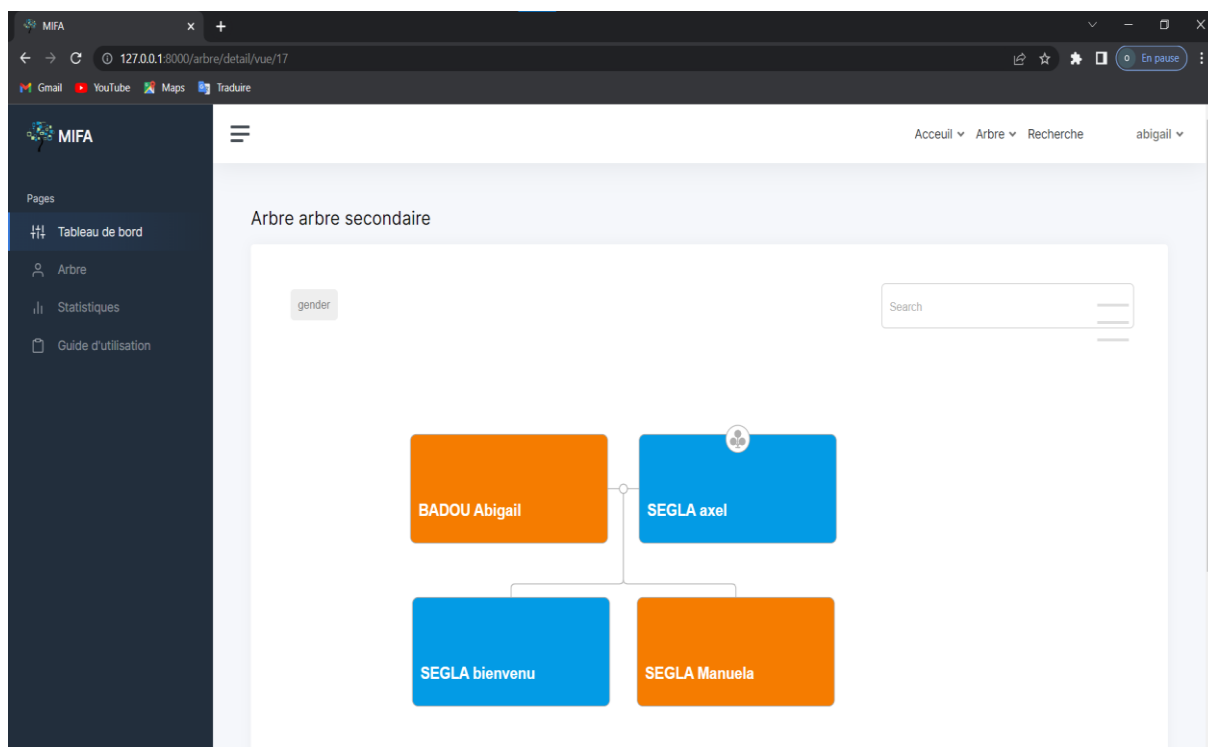


Figure 42 : page présentation visuelle d'un arbre

## 5.4. PRÉSENTATION DES ÉTATS

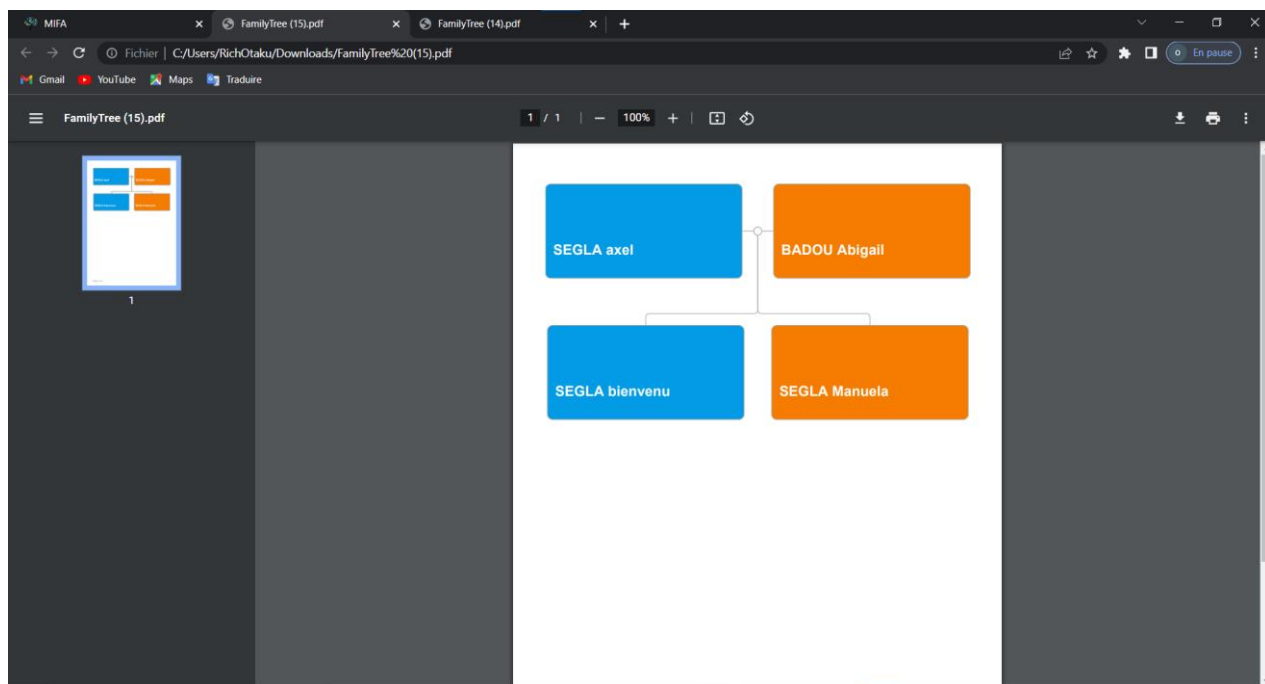


Figure 43 : Exportation en PDF d'un arbre

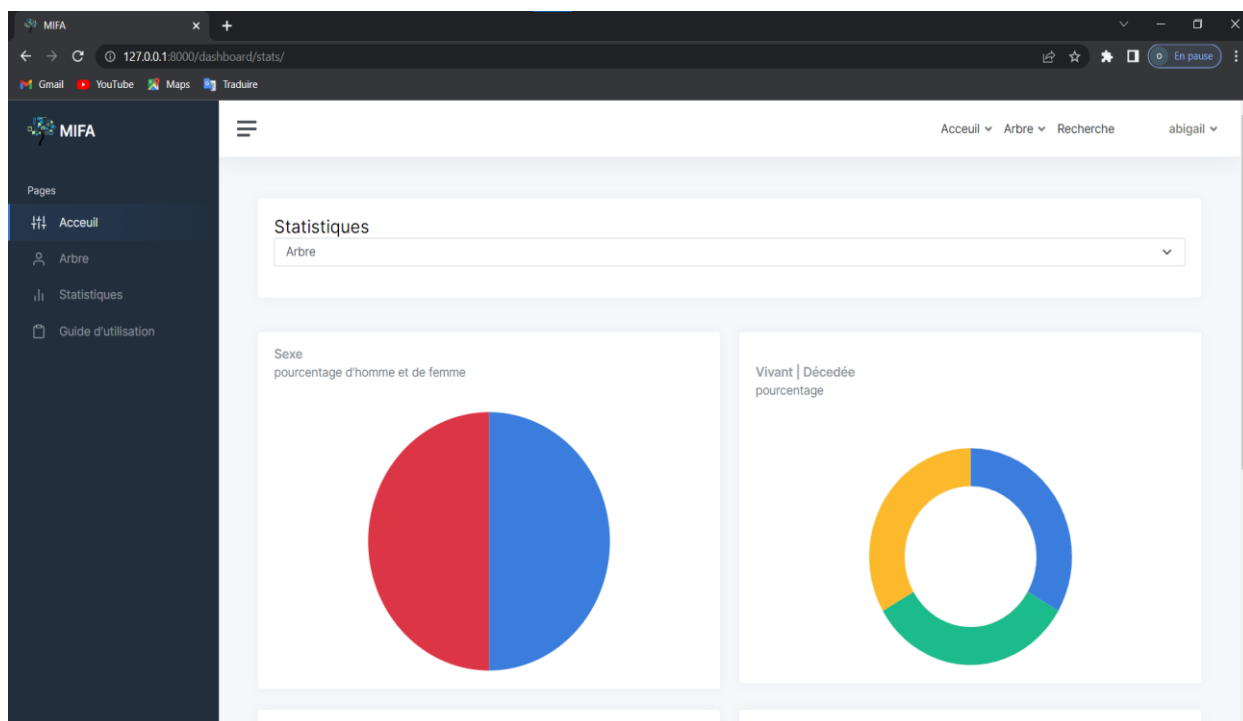


Figure 44 : statistiques d'un arbre

## CONCLUSION

L'objectif de ce projet est de mettre en place une plate-forme web pour permettre aux utilisateurs de créer leurs arbres généalogiques. Au bout d'une analyse guidée par le processus 2TUP et le langage UML, nous sommes parvenus à mettre sur pied le module satisfaisant les différents points de notre cahier de charges. Notre module a été développé avec le Framework Django basé sur le langage Python.

Ce projet a été pour nous, une occasion de sortir du cadre théorique et d'appliquer les connaissances acquises lors de notre formation à IAI-TOGO dans un environnement réel de travail. Ce qui nous a également permis de nous initier dans le domaine professionnel et d'apprendre plusieurs attitudes et habitudes sociales. Il nous a également permis d'apprendre de nouvelles technologies du développement web et ainsi consolider nos connaissances dans le domaine des web service.

En perspective nous continuons de réfléchir pour l'amélioration de notre application et l'ajout de nouvelles fonctionnalités dans un futur proche.

## BIBLIOGRAPHIE INDICATIVE

### ❖ **Ouvrages**

- Documentation de Django
- UML2 par la pratique 5è Edition : Pascal Roques ;

### ❖ **Notes de cours**

- Cours d'UML de M. AMEVOR (2020 – 2021) ;
- Cours de Gestion de projets Informatique de M. AMEVOR (2021 – 2022) ;
- Cours d'Introduction au Génie Logiciel de M. SANI (2021-2022) ;
- Plateforme et outils de développement : M. AKPEKE (2021 - 2022).

### ❖ **Anciens mémoires consultés**

- NONON SAA Diyane David Dissoclima (2020-2021), Conception et implémentation du module de commande et de suivi des pièces détachées : cas de la plateforme mekano ;
- AGBODJI Sandrine (2020 – 2021), Mise en place d'un logiciel pour l'évaluation du patrimoine des structures audiovisuelles publiques au TOGO ;
- WOAGOU YENDOUBOAME Aimé (2020 – 2021), E-data : plateforme de collecte, de gestion et de contrôle de données en temps réel pour la sécurité nationale.

## WEBOGRAPHIE INDICATIVE

Tableau 16 : sites consultés

Site consultés	Durée de consultation
<a href="https://docs.djangoproject.com/en/4.1/">https://docs.djangoproject.com/en/4.1/</a>	Tout au long du projet
<a href="https://stackoverflow.com">https://stackoverflow.com</a>	Tout au long du projet
<a href="https://www.youtube.com">https://www.youtube.com</a>	Tout au long du projet
<a href="https://www.web-dev-qa-db-fra.com/fr/javascript/comment-creez-vous-un-arbre-genealogique-dans-d3.js/">https://www.web-dev-qa-db-fra.com/fr/javascript/comment-creez-vous-un-arbre-genealogique-dans-d3.js/</a>	25/07/2022 au 31/07/2022
<a href="https://www.filae.com/">https://www.filae.com/</a>	Lors de la phase de collecte d'informations sur le thème
<a href="https://en.geneanet.org/">https://en.geneanet.org/</a>	Lors de la phase de collecte d'informations sur le thème
<a href="https://balkan.app/FamilyTreeJS/">https://balkan.app/FamilyTreeJS/</a>	Tout au long du projet
<a href="https://www.delftstack.com/fr/">https://www.delftstack.com/fr/</a>	02/08/2022 8h45 à 13h20



## TABLE DES MATIERES

<b>DEDICACES</b> .....	i
<b>REMERCIEMENTS</b> .....	ii
<b>SOMMAIRE</b> .....	iii
<b>RÉSUMÉ</b> .....	v
<b>ABSTRACT</b> .....	vi
<b>GLOSSAIRE</b> .....	vii
<b>LISTES DES FIGURES</b> .....	viii
<b>LISTES DES TABLEAUX</b> .....	ix
<b>LISTES DES PARTICIPANTS AU PROJET</b> .....	x
<b>INTRODUCTION</b> .....	1
<b>PARTIE I : CAHIER DES CHARGES</b> .....	2
<b>1.1. PRÉSENTATIONS</b> .....	3
<b>1.1.1. Brève présentation de l'IAI-TOGO</b> .....	3
<b>1.1.2. Présentation du cadre de stage</b> .....	4
<b>1.2. THÈME DU STAGE</b> .....	6
<b>1.2.1. Présentation du sujet</b> .....	6
<b>1.2.2. La problématique du sujet</b> .....	6
<b>1.2.3. Intérêt du sujet</b> .....	7
<b>1.2.3.1. Objectifs</b> .....	7
<b>1.2.3.2. Résultats attendus</b> .....	7
<b>1.3. ETUDE DE L'EXISTANT</b> .....	8
<b>1.5. PROPOSITIONS DE SOLUTIONS</b> .....	8
<b>1.5.1. Evaluation technique de la solution</b> .....	8
<b>1.5.2. Evaluation financière de la solution</b> .....	9
<b>1.6. PLANNING PRÉVISIONNEL DE RÉALISATION</b> .....	11
<b>PARTIE II : ANALYSE ET CONCEPTION</b> .....	14
<b>2.1. CHOIX DE LA MÉTHODE D'ANALYSE ET JUSTIFICATION</b> .....	15
<b>2.2. CHOIX DE L'OUTIL DE MODÉLISATION ET JUSTIFICATION</b> .....	18
<b>2.2.1. Présentation et justification du choix de StarUML</b> .....	19
<b>2.3. ETUDE DÉTAILLÉE DE LA SOLUTION</b> .....	19
<b>2.3.1. Diagramme de cas d'utilisation</b> .....	20
<b>2.3.2. Description textuelle de certains cas d'utilisation</b> .....	23
<b>2.3.3. Diagramme de classes</b> .....	26

2.3.4.	Diagramme de séquences .....	28
2.3.5.	Diagramme d'activités .....	30
2.3.6.	Diagramme de déploiement .....	32
PARTIE III : REALISATION ET MISE EN ŒUVRE .....		33
3.1.	MATÉRIELS ET LOGICIELS UTILISÉS .....	34
3.1.1.	Matériels utilisés .....	34
3.1.2.	Logiciels utilisés .....	34
3.2.	ARCHITECTURES MATÉRIELLES ET LOGICIELS DE L'APPLICATION .....	39
3.2.1.	Architecture matérielle de l'application .....	39
3.2.2.	Architecture logicielle de l'application .....	39
3.3.	SÉCURITÉ DE L'APPLICATION .....	41
3.4.	MISE EN PLACE DE LA BASE DE DONNÉES .....	42
PARTIE IV : GUIDE D'EXPLOITATION .....		45
4.1.	CONFIGURATION LOGICIELLE ET MATÉRIELLE .....	46
4.1.1.	Configuration matérielle .....	46
4.1.2.	Configuration logicielle .....	46
4.2.	DÉPLOIEMENT ET SUIVI .....	47
4.2.1.	Déploiement .....	47
4.2.2.	Suivi .....	52
4.2.2.1.	Suivi des bugs .....	52
4.2.2.2.	Versionning du projet .....	52
4.2.2.3.	Sauvegarde de données .....	53
4.3.	MAINTENANCE : ACTIONS À MENER EN CAS DE CERTAINES ERREURS .....	53
PARTIE V : GUIDE D'UTILISATION .....		54
5.1.	DESCRIPTION TEXTUELLE DU DOCUMENT .....	55
5.2.	PLAN DE NAVIGATION .....	56
5.3.	PRÉSENTATION DES DIFFÉRENTES INTERFACES DE L'APPLICATION .....	57
5.3.1.	Page d'accueil .....	57
5.3.2.	Page liste des arbres .....	57
5.3.3.	Page détail d'un arbre .....	58
5.3.4.	Page présentation visuelle d'un arbre .....	58
5.4.	PRÉSENTATION DES ÉTATS .....	59
CONCLUSION .....		60
BIBLIOGRAPHIE INDICATIVE .....		I
WEBOGRAPHIE INDICATIVE .....		II
TABLE DES MATIERES .....		III