

Université Assane Seck de Ziguinchor

UFR Sciences et Technologies

Département Informatique



## Mémoire de fin d'études

Pour l'obtention du diplôme de Master

Mention : Informatique

Spécialité : Génie Logiciel

Dématérialisation des procédures et documents administratifs de  
l'état-civil de Ziguinchor

Présenté par :

**M. Amadou DIEDHIOU**

Soutenance le 18/02/2022

Membres du jury

-Pr. Youssou DIENG (**Président du jury**)

- M. Bassirou DIENE (**Rapporteur**)

- Pr. Youssou FAYE (**Examineur**)

- Dr. Ibrahima DIOP (**Encadreur**)

Sous la direction de :

- Dr. Ibrahima DIOP

Sous la supervision de :

- Pr. Youssou DIENG

Année universitaire : 2019 - 2020

## RESUME

L'hôtel de ville de Ziguinchor utilise depuis très longtemps des fichiers Word, Excel, Dactylographie pour la gestion de l'enregistrement des actes d'état-civil. Ces fichiers ne répondent pas entièrement aux attentes des utilisateurs, ainsi un logiciel de gestion des actes d'état-civil a été conçu par AIMF, installé et exploité au centre principal de Dakar, dans 14 centres secondaires, à Kaolack, Thiès, Saint-Louis et en 2005 à Ziguinchor.

Mais, ce logiciel présente des limites notamment dans la manipulation que dans l'archivage. En 2007, la société ATI qui a conçu le logiciel n'existe plus et des problèmes de mises à jour et de complétude des fonctionnalités ont commencés à se poser au niveau de l'application.

Il y a des difficultés pour adapter le logiciel par rapport à la déclaration de l'acte III de la décentralisation et l'acceptation du registre électronique par le code de la famille.

Mais aussi par rapport à la complétude des fonctionnalités et des documents imprimables. Et enfin, la partie du logiciel qui était censé gérer la gestion des statistiques n'est pas fonctionnelle, de même que la gestion de l'archivage.

Dans ce mémoire, nous proposons ainsi une nouvelle application dans le but de remplacer l'ancienne. Pour ce faire, des études ont été menées ainsi que des séries d'entretiens avec les utilisateurs (Officier état-civil, Agents d'état-civil, administrateur). Nous avons aussi opté pour la méthode (ou processus) unifiée **2TUP** pour le processus de développement de notre application. Une spécification des besoins nous a permis de formaliser les données avec UML. En ce qui concerne la structuration et le stockage des données, nous avons utilisé le Système de Gestion de Bases de Données Relationnelles (SGBDR) MySQL. L'implémentation en Spring Boot (JAVA) nous a permis de mettre en place une application souple, sécurisée, facile à utiliser et permettant de gérer parfaitement la gestion des actes d'état-civil dans la commune de Ziguinchor.

# Remerciements

*Avant tout je rends grâce à ALLAH, le tout Miséricordieux, le très Miséricordieux, pour nous avoir donné la force, le courage et l'endurance d'accomplir ce travail.*

*Au terme de ce travail, je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mon encadrant Dr. Ibrahima DIOP pour ses conseils judicieux, précieux et ses directives pertinentes pour sa disponibilité et pour l'intérêt qu'ils portent à mon sujet.*

*Mon grand respect et mes sincères remerciements aux honorables membres du jury Pr. Youssou FAYE, M. Bassirou DIENE et Pr. Youssou DIENG d'avoir bien voulu participer à l'évaluation de ce travail.*

*J'adresse mes reconnaissances au Maire de la commune de Ziguinchor Dr. Abdoulaye BALDE de m'avoir permis de m'inscrire à l'Université Assane SECK de Ziguinchor pour l'obtention du diplôme en Master Génie Logiciel.*

*Je remercie tous les agents municipaux de la commune de Ziguinchor particulièrement le service d'état-civil, en l'occurrence, les Officiers d'état-civil du centre principal et du centre secondaire je veux nommer M. Cheikh Ahmet Tidiane COLY et Samba BA qui m'ont accordé leur temps pour la collecte des informations.*

*A tous les enseignants du département d'informatique, nous vous disons merci pour toutes les compétences que vous nous avez permis d'acquérir durant cette formation.*

*Je tiens à dire combien le soutien quotidien de ma famille a été important durant les deux années de formation, je leur dis aussi merci.*

*En fin nous tenons à remercier toute personne qui de près ou de loin, a contribué à l'accomplissement de ce travail et à notre réussite dans notre quête du savoir.*

# Dédicaces

*Je dédie ce travail :*

*A ma Mère, Fatou SANE, en vous je vois la maman parfaite, toujours prête à se sacrifier pour le bonheur de ses enfants. Merci pour tout !*

*A mon défunt Père, Fodé DIEDHIOU, en vous je vois un père dévoué à sa famille. Ta présence en toute circonstance m'a plusieurs fois rappelé le sens de la responsabilité. Merci pour tout !*

*À ma tante Fily DIALLO coépouse de mon père qui m'a toujours soutenu pendant toutes les années d'études.*

*A madame DIEDHIOU née Fatou GOUDIABY, nous te disons merci pour ta compréhension et ton aide. Car nous sommes rarement à la maison pendant la journée et le soir l'ordinateur occupe tout notre temps jusque tard dans la nuit.*

*A mes frères et sœurs qui m'ont toujours soutenu. Que Dieu vous Bénisse.*

*A Alpha SANE, qui est le premier à m'accueillir en classe, un camarade de classe qui a rendu l'âme l'année passée. Que Dieu l'accueille dans son paradis !*

*A tous mes amis.*

# SOMMAIRE

## Table des matières

LISTE DES FIGURES .....	6
LISTE DES TABLEAUX.....	7
LISTE DES ABREVIATIONS .....	8
INTRODUCTION GÉNÉRALE .....	9
CHAPITRE I : CONTEXTE JUSTIFICATIF DU SUJET .....	11
<b>I.1. Présentation de la mairie de Ziguinchor</b> .....	11
<b>I.2.1. Les acteurs</b> .....	14
<b>I.2.2. Description des processus actuel de la gestion de l'E.C</b> .....	14
<b>I.2.3. Problèmes liés à la gestion des A.E.C à la C.Z</b> .....	15
<b>I.3. Problématique du sujet</b> .....	17
<b>I.3.1 Solution proposée</b> .....	18
<b>I.3.2. Les objectifs spécifiques du sujet</b> .....	20
<b>I.4 Cadre méthodologique : le processus unifié 2TUP</b> .....	21
<b>I.4.1 Processus Unifié</b> .....	21
<b>I.4.2 Le processus 2TUP</b> .....	22
CHAPITRE II: SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS .....	24
<b>II.1 Spécification des besoins fonctionnels</b> .....	24
<b>II.1.1 Identification des acteurs du système</b> .....	24
<b>II.1.2 Identification des fonctionnalités du système</b> .....	26
<b>II.1.3 Diagrammes de cas d'utilisation</b> .....	27
<b>II.2 Analyse des besoins fonctionnels du système</b> .....	31
<b>II.2.1 Analyse de l'authentification</b> .....	31
<b>II.2.2 Analyse d'une déclaration de naissance</b> .....	34
<b>II.2.3 Analyse d'une déclaration de mariage</b> .....	37
<b>II.2.4 Analyse de la déclaration de décès</b> .....	40
CHAPITRE III : CONCEPTION DU SYSTEME.....	44
<b>III.1 Conception générale</b> .....	44
<b>III.1.1 Architecture de l'application</b> .....	44
<b>III.1.2 Diagramme de composants</b> .....	45
<b>III.1.3 Diagramme de packages</b> .....	46
<b>III.1.4 Diagramme de déploiement</b> .....	48
<b>III.2 Conception détaillée</b> .....	48
<b>III.2.1 Diagramme de classes</b> .....	49
<b>III.2.2 Dictionnaire de données</b> .....	52
CHAPITRE IV: IMPLEMENTATION ET PRESENTATION DE L'APPLICATION.....	53
<b>IV.1 Outils et technologies utilisés</b> .....	53

<b>IV.1.1 Outil de conception : PowerAMC</b> .....	53
<b>IV.1.2 Environnement de développement : Eclipse</b> .....	53
<b>IV.1.3 Langage de programmation : JAVA</b> .....	53
<b>IV.1.4 Framework Spring: SPRING BOOT</b> .....	54
<b>IV.1.5 Moteur de Template : THYMELEAF</b> .....	54
<b>IV.1.6 Tomcat</b> .....	55
<b>IV.1.7 Langage de requête : SQL</b> .....	55
<b>IV.1.8 Frameworks: Bootstrap et jQuery</b> .....	55
<b>IV.1.9 Le SGBDR : MySQL</b> .....	55
<b>IV.2 Implémentation</b> .....	56
<b>IV.2.1 Implémentation de la base de données</b> .....	56
<b>IV.2.1.1 Modèle physique de données (MPD)</b> .....	56
<b>IV.2.1.2 Etablissement de connexion à la base de données</b> .....	58
<b>IV.2.2 Implémentation de l'application</b> .....	58
<b>IV.3 Sécurisation de l'application</b> .....	64
<b>IV.4 Présentation de quelques interfaces graphiques de l'Application</b> .....	65
<b>IV.4.1 Authentification</b> .....	65
<b>IV.4.2 Page d'accueil</b> .....	66
<b>IV.4.3 Déclaration de naissance</b> .....	66
<b>IV.4.4 Déclaration de mariage</b> .....	67
<b>IV.4.5 Déclaration de décès</b> .....	68
<b>IV.4.3 Gestion d'un acte de naissance</b> .....	68
<b>IV.4.4 Gestion d'un acte de mariage</b> .....	69
<b>IV.4.5 Gestion d'un acte de décès</b> .....	70
<b>IV.4.4 Rechercher un acte de naissance</b> .....	70
<b>IV.3.5 Gestion statistiques des actes de naissances</b> .....	75
<b>IV.3.6 Gestion administrative</b> .....	75
<b>CONCLUSION GENERALE ET PERSPECTIVES</b> .....	77
<b>WEBOGRAPHIE</b> .....	78
<b>ANNEXE</b> .....	79

## LISTE DES FIGURES

Figure 1: Organigramme de la mairie de Ziguinchor .....	13
Figure 2 : Présentation du processus 2TUP .....	23
Figure 3 : Diagramme de cas d'utilisation d'une déclaration de naissance .....	28
Figure 4: Diagramme de cas d'utilisation d'une déclaration de mariage.....	29
Figure 5: Diagramme de cas d'utilisation d'une déclaration de décès .....	30
Figure 6: Diagramme de cas d'utilisation pour l'administrateur.....	31
Figure 7: Diagramme d'activité du cas d'utilisation "s'authentifier" .....	33
Figure 8: Diagramme de séquence du cas d'utilisation "s'authentifier" .....	34
Figure 9: Diagramme d'activité du cas "déclaration de naissance" .....	36
Figure 10: Diagramme de séquence du cas d'utilisation "déclaration de naissance" .....	37
Figure 11: Diagramme d'activité du cas "déclaration de mariage" .....	39
Figure 12: Diagramme de séquence du cas d'utilisation "déclaration mariage" .....	40
Figure 13: Diagramme d'activité du cas "déclaration de décès" .....	42
Figure 14: Diagramme de séquence du cas d'utilisation "déclaration de décès" .....	43
Figure 15: Architecture de l'application .....	45
Figure 16: Diagramme de composants.....	46
Figure 17: Diagramme de packages .....	47
Figure 18: Diagramme de déploiement.....	48
Figure 19: Diagramme de classes participantes aux fonctionnalités de la gestion l'authentification et des utilisateurs .....	49
Figure 20: Diagramme de classes participantes aux fonctionnalités de la gestion des actes de naissances d'état-civil.....	50
Figure 21: Diagramme de classes participantes aux fonctionnalités de la gestion des actes de mariages d'état-civil .....	51
Figure 22: Diagramme de classes participantes aux fonctionnalités de la gestion des actes de Décès d'état-civil.....	52
Figure 23: Schéma relationnel du MPD.....	57
Figure 24: Exemple de fichier application.properties .....	58
Figure 25: l'arborescence générale de notre application.....	59
Figure 26 : Class « Naissance »: Modèle.....	60
Figure 27: Implémentation d'une interface .....	61
Figure 28: Classe «NaissanceController » : Contrôleur .....	62
Figure 29: Fichier « Naissance » : Vue .....	63
Figure 30: La classe « NaissanceMetier » : métier .....	64
Figure 31:Interface d'authentification .....	66
Figure 32:Interface page d'accueil .....	66
Figure 33: Interface page déclaration de naissance.....	67
Figure 34: Interface page déclaration de mariage .....	67
Figure 35: Interface page déclaration de décès .....	68
Figure 36: Interface page gestion de naissance .....	69
Figure 37: Interface page gestion de mariage.....	69
Figure 38: Interface page gestion de décès.....	70
Figure 39: Interface page recherche acte naissance .....	71

Figure 40: Interface page affichage d'un acte de naissance trouvé.....	71
Figure 41: Exemple d'un volet de naissance choisi .....	72
Figure 42: Exemple d'un extrait de naissance choisi.....	73
Figure 43: Exemple d'un certificat de naissance choisi.....	74
Figure 44: Interface d'un tableau statistique de naissance.....	75
Figure 45: Interface administration.....	76
Figure 46: Diagramme de classes aux fonctionnalités de la gestion de l'état-civil .....	80

## LISTE DES TABLEAUX

Tableau 1 :Tâches des acteurs.....	15
Tableau 2:Identification des acteurs .....	25
Tableau 3: Identification des fonctionnalités .....	26
Tableau 4: Description de cas d'utilisation « s'authentifier » .....	32
Tableau 5: Description de cas d'utilisation « déclaration de naissance » .....	35
Tableau 6: Description de cas d'utilisation « déclaration de mariage » .....	38
Tableau 7: Description de cas d'utilisation « déclaration décès » .....	41



## LISTE DES ABREVIATIONS

<b>CZ :</b>	Commune de Ziguinchor
<b>UML :</b>	Unified Modeling Language
<b>SGBD :</b>	Système de Gestion de Bases de Données
<b>SGBDR :</b>	Système de Gestion de Bases de Données Relationnelles
<b>E.C :</b>	Etat-Civil
<b>O.E.C :</b>	Officier d'Etat-Civil
<b>A.D :</b>	Agent de déclaration
<b>A.T :</b>	Agent de tirage
<b>A.E.C :</b>	Agent d'Etat-Civil
<b>SQL :</b>	Structured Query Language
<b>AIMF :</b>	Association Internationale des Maires Francophones
<b>PGM :</b>	Plateforme de Gestion Municipale

## INTRODUCTION GÉNÉRALE

L'E.C (Etat-Civil) est la situation de la personne dans la famille et la société, résultat d'une procédure écrite d'identification administrative. L'E.C désigne ainsi également le service public chargé de dresser les actes constatant ces faits, notamment les actes de naissance, de mariage et de décès [1].

À la mairie de Ziguinchor, le processus de gestion des actes d'E.C se fait grâce à un logiciel conçu par l'AIMF (Association Internationale des Maires Francophones), installé et exploité au centre principal de Ziguinchor depuis 2005. Mais, ce logiciel présente des failles notamment dans la manipulation, dans l'archivage et même dans la statistique.

L'utilisation du logiciel n'est pas très pratique et cause beaucoup de désagréments telles que :

- ❖ Les bogues du système en cas d'ajout d'accents « ' » par exemple lors de la saisie;
- ❖ Difficulté de rechercher des actes de mariages et de décès ;
- ❖ L'absence de prise en compte des statistiques de naissances, mariages et décès ;
- ❖ L'absence d'une page de validation de l'O.E.C (Officier d'Etat-Civil) des actes saisis par ses agents ;
- ❖ La non prise en compte des actes « né(e) en » et « né(e) vers », donc l'O.E.C est obligé de les saisir à mains levée ;
- ❖ La suppression des actes d'E.C est gérée par l'administrateur et non par l'O.E.C ;
- ❖ L'absence de sécurité sur les comptes des utilisateurs.
- ❖ Le problème de mise à jour ;
- ❖ L'application ne marche qu'avec un server Windows server et Microsoft SQL server or ces systèmes sont payants.
- ❖ etc.

Pour pallier aux manquements du logiciel existant, la mairie de Ziguinchor a cherché à acheter de nouvelles applications telles que la EASY GEC développé par EASYTECS GAM, PGM développé par DECISION CONSULTING, etc. Mais, souvent ces sociétés vendent déjà leurs produits à des prix exorbitants, en plus, elles recommandent de renouveler la licence chaque année pour pouvoir continuer à utiliser ceux-ci. Finalement, elle a décidé, à travers son service informatique, en collaboration avec l'O.E.C, a jugé nécessaire de mettre en place une application optimale et fiable pour la gestion automatique des actes d'E.C pour le compte de cette dernière.

Cette application web permettra, d'abord, à la mairie à travers son service d'E.C d'être plus efficace, permettre à l'O.E.C d'avoir une marge de manœuvre complet sur le système mis en place, mais aussi une meilleure satisfaction de la population lors des déclaration, délivrance et publication des actes d'E.C.

Ainsi, afin de mettre en place une telle solution, le service informatique de la mairie de Ziguinchor dont nous sommes les responsables, s'est fixé comme objectif de concevoir et de développer cette solution pour la **dématérialisation des procédures et documents administratifs de l'état-civil de Ziguinchor.**

Une étude du système existant a été faite en organisant des entretiens avec les O.E.C du centre principal et du centre secondaire de Ziguinchor. Ces derniers nous ont fait part du mode de fonctionnement de l'E.C, du système existant ainsi que les problèmes rencontrés avec celui-ci. Face à cela, le service d'E.C dans son ensemble a exprimé ses besoins en précisant ses attentes sur la nouvelle application. Elle doit répondre à toutes les qualités d'un logiciel.

De cette manière, le système développé et qui va être déployé courant cette année va permettre de:

- Gérer les informations de la collectivité locale;
- Gérer les utilisateurs ;
- Gérer les déclarations de naissances ;
- Gérer les déclarations de mariages ;
- Gérer les déclarations de décès ;
- Gérer la validation des actes enregistrés par l'O.E.C et ses Agents
- Gérer la modification et de la suppression des actes enregistrés par l'O.E.C
- Gérer l'interconnexion entre les centres (principal et secondaires)
- Imprimer des documents (Volet, Extrait, Bulletin, Copie littérale et certificat).
- Gérer l'accessibilité via l'internet.
- ...

Le travail que nous avons réalisé est décrit dans la suite de ce document de mémoire de master 2, qui est organisé en cinq principaux chapitres, que sont:

- **Chapitre I: Contexte justificatif du sujet et processus de développement de l'application** expose les problèmes dans la gestion des actes d'E.C et permet de faire connaissance à la méthode unifiée **2TUP** adapté dans notre cas.
- **Chapitre II: Spécification et l'analyse des besoins fonctionnels** identifie les

acteurs et leurs fonctionnalités, analyse les besoins qui doivent être satisfaits par ce système.

- **Chapitre III: Conception du système** aborde la conception générale et la conception détaillée du système.
- **Chapitre IV: Implémentation et réalisation** présente l'application ainsi que les outils utilisés pour la réaliser.

Pour bien finaliser ce travail, ce mémoire dispose d'une conclusion et des perspectives.

## CHAPITRE I : CONTEXTE JUSTIFICATIF DU SUJET

### I.1. Présentation de la mairie de Ziguinchor

Créée en 1956, la Mairie de Ziguinchor est une Collectivité territoriale décentralisée par l'Etat du Sénégal.

Le conseil municipal est composé de soixante-seize (76) Conseillers dont 07 adjoints au Maire et des commissions spécialisées. Elle a pour mission d'assurer à l'ensemble des populations, sans discrimination, les meilleures conditions de vie.

Au-delà des élus, la mairie a une administration constituée d'agents municipaux. Cette administration est divisée en deux : le cabinet du Maire et le Secrétariat Général.

Le cabinet du Maire a pour mission de coordonner les activités du maire ; de suivre le courrier du bureau municipal, d'appuyer le secrétariat général dans la gestion administrative de la commune en mettant en œuvre des stratégies économiques et sociales dans le but de satisfaire les objectifs du bureau municipal (le maire et ses adjoints).

Le Secrétariat Général a une mission juridique et réglementaire, de coordination des pôles ressources humaines, financier, des moyens généraux et autres pôles définis par le code des collectivités territoriales.

Le secrétariat Général est composé du bureau du courrier, de quatre cellules et de 5 directions (figure 1).

## **I.2. La gestion de l'état-civil dans la C.Z**

L'E.C est un ensemble de règles à la fois juridiques pour enregistrer les naissances, les mariages et les décès sur les registres de manière continue et obligatoire, de conserver, de délivrer copies des informations relatives aux événements liés à l'état-civil.

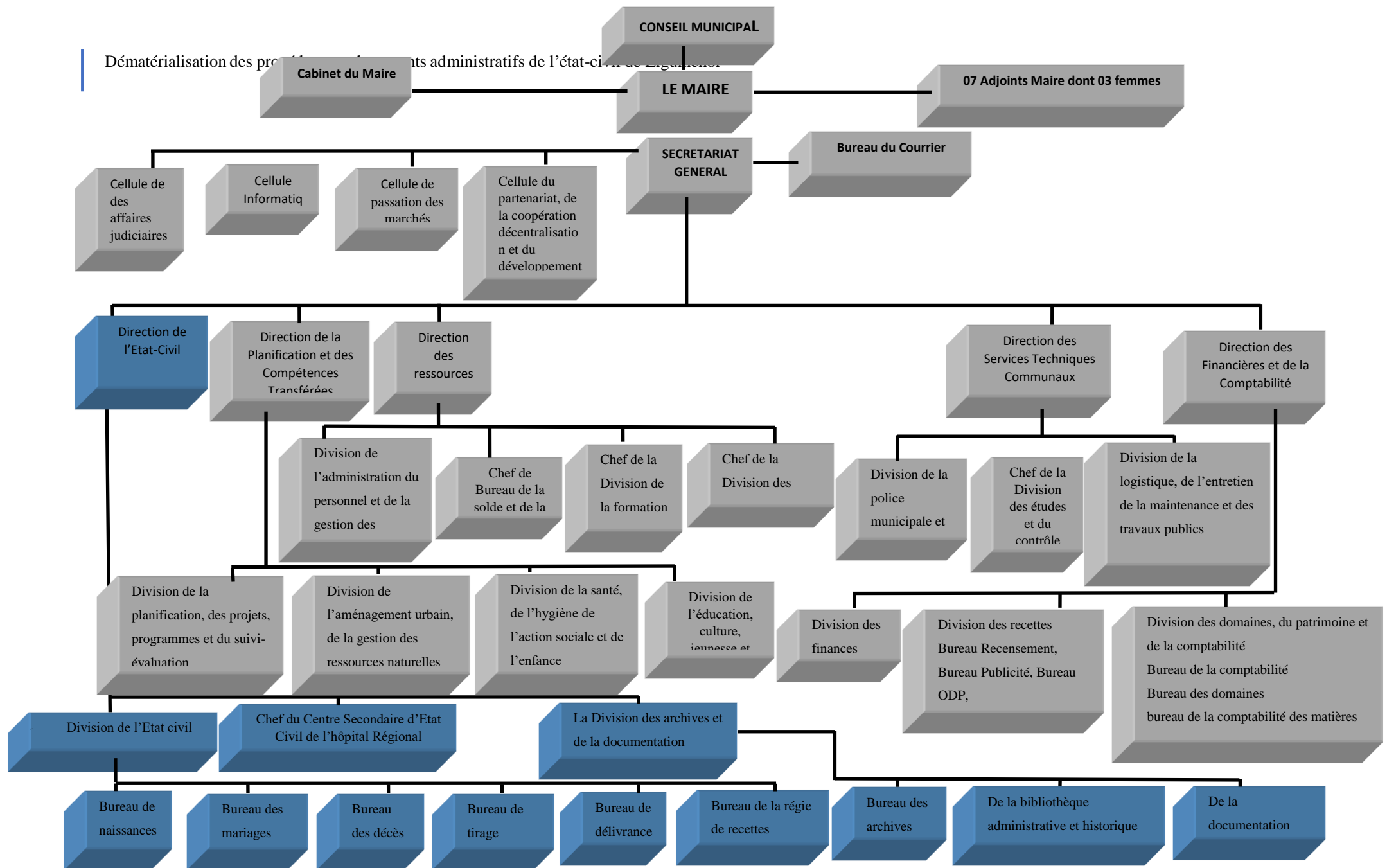


Figure 1: Organigramme de la mairie de Ziguinchor

### **I.2.1. Les acteurs**

Le système actuel constitue trois (3) principaux acteurs, qui sont à décrire dans le processus de gestion de l'E.C.

#### **➤ L'officier d'Etat Civil**

Le fonctionnement du système est assuré par l'O.E.C. Il gère la gestion des actes, en corrigeant les erreurs des actes saisis, délivre aux ayant droits des extraits ou des copies partielles ou intégrales des déclarations qu'ils établissent et conservent.

#### **➤ L'agent de saisi(e) ou l'agent de tirage**

L'agent de saisi(e) fait la saisi et l'impression des actes d'état-civil sous le l'autorisation de l'O.E.C.

#### **➤ L'administrateur**

L'administrateur du système, est un acteur principal. C'est à lui d'installer, de configurer l'application. Créer les comptes d'utilisateurs, ajouter les années de registres, les codes des centres principaux et secondaires, les structures sanitaires et les officiers d'état-civil.

### **I.2.2. Description des processus actuel de la gestion de l'E.C**

Dans cette partie de notre travail, nous allons décrire le processus actuel de la gestion des actes d'état-civil dans la commune de Ziguinchor. Pour ce faire nous allons utiliser un tableau avec une colonne pour les étapes (elles sont numérotées), une deuxième pour les tâches dans chaque étape et une troisième pour les acteurs qui exécutent les tâches (tableau 1).

Tableau 1 :Tâches des acteurs

ETAPES	TACHES	ACTEURS
Etape 1	Installation et configuration du serveur	Administrateur
Etape 2	Installation et configuration de l'application <b>Etatcivil</b> dans les machines clients rattachées au serveur (Windows server et SQL server)	Administrateur
Etape3	Créer les comptes utilisateurs	Administrateur
Etape 4	Saisi des années de registres(par requêtes SQL)	Administrateur
Etape 5	Saisi des codes de centres d'état-civil	Administrateur
Etape 6	Ajouter les structures sanitaires	Administrateur
Etape 7	Ajouter les officiers d'état-civil	Administrateur
Etape 8	Saisi de déclarations (naissances, mariages et décès)	Agent de saisi et Officier d'état-civil
Etape 9	Modifier les déclarations (naissances, mariages et décès)	Officier d'état-civil
Etape 10	Supprimer les déclarations (naissances, mariages et décès)	Administrateur
Etape 11	Imprimer les déclarations (naissances, mariages et décès)	Agent de tirage et Officier d'état-civil

Cette procédure établie dans le tableau ci-dessus représente le processus de gestion de l'E.C de façon théorique, mais dans le cas pratique, de nombreuses difficultés sont rencontrées. Ainsi, dans la partie suivante, nous allons décrire quelques problèmes liés à la gestion des actes d'E.C à la mairie de Ziguinchor.

### I.2.3. Problèmes liés à la gestion des A.E.C à la C.Z

A la C.Z (commune de Ziguinchor) on note l'utilisation d'un logiciel nommé **Etatcivil**, incomplet dans lequel on enregistre toutes les déclarations de naissance, mariage et décès. Au fil du temps, son utilisation dans la gestion d'E.C, a permis de déceler des problèmes. Nous pouvons citer :



❖ **Les déclarations né (e) en et né (e) vers:**

Ce sont des actes dont on ne connaît pas la date de naissance exacte de l'individu et donc il faut soit fixer une année ou étaler vers une année, le logiciel **Etatcivil** ne prend pas en compte ces derniers alors, c'est à l'O.E.C d'utiliser un fichier Word où est déjà établi des exemples qu'il faut juste modifier en cas de besoin. Les conséquences majeures de l'utilisation de ces fichiers sont l'absence d'une certaine automatisation de cette tâche dans le processus de délivrance des actes d'E.C.

❖ **Le niveau de sécurité faible :**

Avec l'utilisation du logiciel **Etatcivil**, les droits accès sont basés par l'utilisation d'un nom utilisateur et d'un mot de passe non crypté. Ces derniers sont octroyés par l'administrateur à chaque utilisateur, ainsi il peut accéder à tous les comptes et procéder à la modification des A.E.C.

❖ **L'absence de page de validation des A.E.C :**

Il faut aussi noter l'absence d'une page de validation par l'officier d'E.C des actes saisis par ses agents. Qui aurait dû permettre à l'O.E.C de vérifier ces derniers en les acceptant ou les rejetant.

❖ **L'absence de traçabilité :**

Il est impossible de tracer l'historique des activités de mise à jour (saisie, modification, suppression, impression) des données effectuées par les utilisateurs. Cette fonctionnalité est nécessaire pour éviter les fraudes et la concurrence.

❖ **La difficulté de rechercher des actes de mariages et de décès par nom et/ou prénoms:**

Avec l'utilisation du logiciel **Etatcivil**, il est difficile de rechercher par prénom et/ou nom d'un acte de mariage ou de décès, car la recherche se fait ligne après ligne sur une liste de plus de mille actes enregistrés. Ce qui cause des désagréments tels que des lenteurs dans le travail.

❖ **Absence de sécurité des documents imprimés :**

Avec l'utilisation du logiciel **Etatcivil**, les papiers imprimés ne sont pas protégés pour garantir leur intégrité à valeur de justificatifs.

❖ **Absence de statistiques :**

Avec l'utilisation du logiciel **Etatcivil**, la partie statistique n'est pas fonctionnelle, d'où la difficulté de prendre en compte des statistiques de naissances, mariages et décès.

### I.3. Problématique du sujet

La gestion de l'E.C se faisait avec des papiers imprimés qu'ont rempli avec la main levée, ensuite des machines dactylographes métallique puis électriques. Avec l'acquisition d'ordinateurs, d'imprimantes et de photocopieuses, l'administration de la mairie d'alors a jugé opportun d'aller vers l'automatisation des tâches afin de réduire l'utilisation du papier. L'application **Etatcivil** est utilisée dans la plupart des communes du Sénégal, c'est en 2005 qu'elle est installée dans la commune de Ziguinchor.

Mais, en 2007 la société ATI qui a conçu le logiciel n'existe plus et des problèmes de mises à jour et de complétude des fonctionnalités ont commencés à se poser au niveau du logiciel. Les difficultés rencontrées sont de plusieurs ordres :

- ✓ **Par rapport à la déclaration acte III de la décentralisation et l'acceptation du registre électronique par le code de la famille** : Le logiciel **Etatcivil** a été créé avant l'acte III de la décentralisation donc incompatible avec celui-ci. En outre, le registre électronique n'est toujours pas pris en compte dans le code de la famille.
- ✓ **Par rapport à la complétude des fonctionnalités et des documents imprimables** : Des bogues du système sont souvent signalés à l'administrateur et il faut aussi noter le manque de garantir l'intégrité des papiers imprimés, à valeur de justificatifs.
- ✓ **Par rapport à la maintenance du logiciel** : Il n'y a plus de maintenance depuis 2007, l'année à laquelle la société ATI qui a conçu le logiciel a cessé d'exister.
- ✓ **Par rapport à la gestion des statistiques** : la partie statistique n'a pas été développée, ce qui fait que l'O.E.C ne peut pas, à la fin de chaque trimestre, adresser au Service des statistiques un état des naissances, des mariages, divorces, décès et des enfants sans vie inscrits au cours du trimestre.
- ✓ **Par rapport à la gestion des archivages** : Elle est inexistante et pourtant utile car l'archivage du premier volet du registre est impératif du fait qu'il arrive souvent que des registres se détériorent à cause du mauvais entretien de ceux-ci.
- ✓ etc.

Nous allons donc proposer une solution aux problèmes rencontrés dans la gestion des actes d'E.C, puis nous décrivons les objectifs spécifiques du sujet de notre mémoire.

### I.3.1 Solution proposée

Face aux problèmes et difficultés rencontrés dans le logiciel **Etatcivil** à la CZ, nous proposons une application web adaptée aux besoins spécifiques de la gestion des actes d'E.C. Avec cette dernière, nous pensons pouvoir disposer d'un outil adéquat permettant de pallier aux manquements du logiciel **Etatcivil** utilisé présentement. Elle contribuera non seulement à remédier aux problèmes ci-dessus mais aussi à renforcer le système de gestion des A.E.C à travers de nouvelles fonctionnalités. Celles-ci comprennent entre autres :

- ✚ Gestion des utilisateurs : l'accès de tout utilisateur (ou agent) à l'application est conditionné par une identification en renseignant son nom utilisateur et son mot de passe. Ces derniers ne sont pas les seules informations de l'utilisateur qui sont renseignées lors de la création d'un utilisateur. Rappelons qu'un mot de passe est octroyé par défaut à chaque utilisateur, qui doit le modifier lors de sa première connexion. Ce nouveau mot de passe sera alors la propriété privée de l'utilisateur et modifiable autant de fois qu'il est nécessaire si le besoin se fait sentir. L'administrateur n'a pas besoin de supprimer un ancien utilisateur, il lui suffit seulement de l'archiver en transformant son statut actif en inactif.
- ✚ Gestion des années de registres : les années de registre à ajouter seront celles qui datent depuis la mise en place de services d'état-civil dans notre pays. Chaque année de registre peut être activé ou désactivé par l'administrateur. Aussi, l'année de registre peut être choisie par défaut par tout utilisateur en fonction des droits autorisés dans l'application. Toutes ces données sont saisies, modifiées et supprimées par l'administrateur
- ✚ Gestion des formations sanitaires: Tous les noms des structures de santé de la commune seront ajoutés par l'administrateur. Ces données font partie des informations liées à la commune.

- ✚ Gestion des centres d'état-civil: Les codes de centre des communes sont uniques pour chaque centre d'état-civil principal comme secondaire et seront ajoutés par l'administrateur. Ces données font partie des informations liées à la commune.
  
- ✚ Gestion l'impression des actes d'état-civil: l'impression des documents comme les volets, extraits de naissance ou mariage, les bulletins et les copies littérales par Agent d'état-civil autorisé par l'O.E.C. Il suffit seulement de rechercher l'acte et de choisir le document désiré puis de cliquer sur « Imprimer » pour avoir la version papier de celui-ci.
  
- ✚ Gestion l'interconnexion du centre principal aux centres secondaires d'état-civil: mettre en place un service reliant les deux centres principaux et centre secondaire entre eux, par des liens d'accès ADSL, SDSL ou FIBRE, dans un réseau privé sécurisé (VPN). Les techniciens informatiques de la mairie pourront travailler à distance sur les serveurs informatiques installés sur l'hôtel de ville, ensuite, les agents pourront échanger des données de centre à centre en toute sécurité.
  
- ✚ Optimisation de la recherche d'actes : La recherche d'actes est essentielle. Premièrement on recherche l'acte par le numéro et l'année de registre, deuxièmement on recherche par prénoms et/ou nom mais cette dernière n'est pas suffisante parfois, il faut aussi trouver la date de naissance, les noms des parents, pour identifier l'acte.
  
- ✚ Paiement en ligne des frais de timbres : Grace à une plateforme sécurisée de paiement en ligne pour faciliter les transactions entre la perception municipale et les clients qui souhaitent avoir un acte d'état-civil à la mairie de Ziguinchor, à travers l'application Web.
  
- ✚ Signature électroniques des Officiers : Permettre un mécanisme de signature numérique rapide des Officiers d'état-civil dans les 24 heures en garantissant la propriété d'intégrité du document signé.

### **I.3.2. Les objectifs spécifiques du sujet**

L'objectif général est de proposer une application web responsive qui permettra d'optimiser le processus de gestion des faits d'état-civil dans la commune de Ziguinchor.

De façon spécifique, nous nous sommes fixés comme objectifs de :

- ❖ Permettre une utilisation synchrone de notre application ;
- ❖ Avoir un système d'information sécurisé et évolutif
- ❖ Archiver tout utilisateur (Officier d'état-civil, Agent de saisi, Administrateur.) inactif au lieu de le supprimer ;
- ❖ Permettre d'avoir un archivage numérique des actes d'état-civil
- ❖ Autoriser les années de registres par activation ou désactivation
- ❖ Créer une interface homme-machine (IHM) ergonomique et facile d'utilisation ;
- ❖ Adapter l'application aux terminaux les plus récents sur n'importe quels réseaux informatiques connectés à l'application ;
- ❖ Permettre l'enregistrement des actes de naissance
- ❖ Permettre l'enregistrement des actes de mariage célébré ou constaté
- ❖ Permettre l'enregistrement des actes de décès
- ❖ Permettre la visualisation d'une déclaration de naissance ;
- ❖ Permettre la visualisation des actes de mariage enregistrés ;
- ❖ Permettre la visualisation d'une déclaration de décès ;
- ❖ Faciliter l'insertion, la modification et la suppression des données ;
- ❖ Optimiser la recherche des actes enregistrés dans la base de données;
- ❖ Permettre la validité des actes saisies avant de les enregistrer ;
- ❖ Imprimer des volets, des extraits, des bulletins et des copies littérales pour les actes de naissance;
- ❖ Imprimer des volets, des extraits, des bulletins et des copies littérales pour les actes de décès;
- ❖ Imprimer des volets, des extraits, des certificats et des copies littérales pour les actes de mariages ;
- ❖ Permettre de faire la statistique des actes de naissances ;
- ❖ Permettre de faire la statistique des actes de décès ;
- ❖ Permettre de faire la statistique des actes de mariage ;
- ❖ Filtrer les accès suivant les profils des utilisateurs.

- ❖ Permettre le paiement en ligne des frais de timbres.
- ❖ Permettre la signature électronique des Officiers.

Pour atteindre ces objectifs, nous avons besoin de travailler en collaboration avec l'O.E.C du centre principal et du centre secondaire, c'est la raison pour laquelle nous avons opté pour une méthodologie de développement qui implique ces derniers. Pour ce faire, nous optons pour la méthode (ou processus) unifiée **2TUP**. **2TUP** (*2 Track Unified Process*) est un processus de développement logiciel qui implémente le Processus Unifié.

## I.4 Cadre méthodologique : le processus unifié 2TUP

### I.4.1 Processus Unifié

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques.

C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise. [3]

Un processus unifié se distingue par ses caractéristiques suivantes [4] :

- **Itératif** : le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et permet de développer une solution effective de façon incrémentale par des itérations multiples.
- **Piloté par les risques** : les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité.
- **Centré sur l'architecture** : le choix de l'architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants du système est basée sur ce choix.
- **Conduit par les cas d'utilisation** : le processus est orienté par les besoins utilisateurs représentés par des cas d'utilisation.

Les activités de modélisation reposent sur UML. Ce langage de modélisation couvre les aspects structurels et dynamiques de l'architecture et de la conception des logiciels. Il facilite une modélisation par composants en utilisant une approche orientée objet. [3]

Dans la communauté objet, il existe plusieurs processus unifiés en vogue comme eXtreme Programming (XP) et Rational Unified Process (RUP). Dans notre étude, on a choisi de

travailler avec le processus 2TUP ; parce qu'il couvre des projets de toute taille et il a pu faire une large place dans le domaine de la technologie et les risques des projets.

### I.4.2 Le processus 2TUP

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 les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le [cahier des charges](#) et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens).

Le processus s'articule ensuite autour de trois phases essentielles :

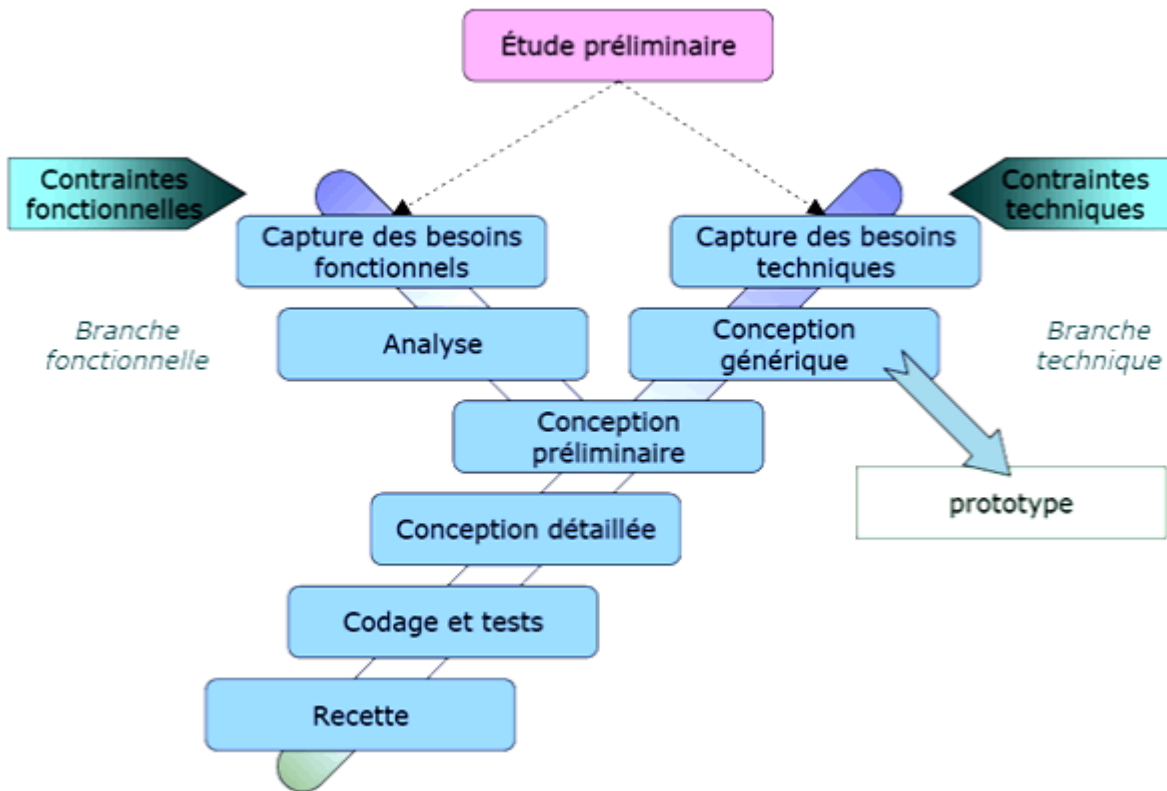
- une branche technique ;
- une branche fonctionnelle ;
- une phase de réalisation.

**La branche fonctionnelle** capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.

**La branche technique** capitalise un savoir-faire technique et/ou des contraintes techniques. Les techniques développées pour le système sont indépendantes des fonctions à réaliser.

**La phase de réalisation** consiste à réunir les deux branches, permettant de mener une conception applicative et enfin la livraison d'une solution adaptée aux besoins [\[4\]](#).

La **figure 2** suivante détaille les étapes de développement des trois branches du processus 2TUP.



Source : [https://www.memoireonline.com/05/13/7195/m\\_Mise-en-place-dune-application-webmapping-de-geolocalisation-des-points-dintert-de-la-vill6.html](https://www.memoireonline.com/05/13/7195/m_Mise-en-place-dune-application-webmapping-de-geolocalisation-des-points-dintert-de-la-vill6.html)

Figure 2 : Présentation du processus 2TUP

En résumé, ce chapitre nous a permis de définir le périmètre de notre sujet. Il a permis de faire une étude détaillée de l'existant, de fixer les objectifs de notre projet et de choisir la méthode à utiliser pour sa réalisation.

Dans le chapitre suivant, nous allons présenter la capture et l'analyse des besoins fonctionnels du système qui correspond aux deux premières étapes de la branche fonctionnelle du processus 2TUP.



## **CHAPITRE II: SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS**

Suite à la définition du processus de développement de notre application, nous entamons, dans ce chapitre, la spécification (identification des acteurs et fonctionnalités) et l'analyse (description des fonctionnalités) des besoins fonctionnels.

Nous y parlerons un peu de la modélisation objet en faisant appel à UML [5] afin de traduire, en diagrammes fonctionnels, les spécifications des besoins fonctionnels.

### **II.1 Spécification des besoins fonctionnels**

Dans cette section, nous allons commencer par identifier les acteurs, ensuite les fonctionnalités et enfin terminer par le diagramme de cas d'utilisation.

#### **II.1.1 Identification des acteurs du système**

Un acteur est une entité qui agit sur le système, comme une personne humaine ou un robot. Une même personne (ou robot) peut jouer le rôle de plusieurs acteurs dans un système, c'est pourquoi les acteurs doivent surtout être décrits par leur rôle. Ce rôle décrit les besoins et les capacités de l'acteur. L'activité du système a pour but de satisfaire les besoins de l'acteur.

Pour connaître les acteurs de notre système, nous nous sommes basés sur le cahier des charges, obtenu à partir d'une étude préliminaire auprès du Directeur et Officier de l'E.C en collaboration avec l'Officier du Centre Secondaire de l'E.C. Dans le tableau, ci-dessous, sont répertoriés tous les acteurs ainsi que leurs actions dans la gestion de l'E.C à la commune de Ziguinchor.

Tableau 2: Identification des acteurs

<b>ACTEURS</b>	<b>ACTIONS</b>
<b>L'Officier</b>	Définir les étapes de :  Recevoir les volets de déclaration de naissance et de décès ;  Recevoir les déclarations de mariage et procéder à leur célébration ;  Établir et signer les déclarations d'état civil ;  Délivrer les extraits et copies des déclarations [1] ;
<b>Agent de déclaration ou Agent de saisie ou Agent d'état-civil</b>	Définir les étapes de :  Recevoir et enregistrer les déclarations de naissance et de décès ;  Transmettre les volets et les tableaux de récapitulation par voie administrative au centre d'état civil du ressort où la déclaration doit être établi ;  Ils sont habilités à recevoir les déclarations dans les délais prescrits par la loi.  Les agents de l'état civil constituent un personnel auxiliaire affecté à la section d'état civil dans les Mairies, les centres secondaires d'état civil, et dans les ambassades et consulats. ...[1].
<b>Administrateur</b>	Définir les étapes de :  Ajouter les éléments nécessaire pour le bon fonctionnement de l'application tels que ajouter les années de registres, les Officiers, les formations sanitaires, les code de centres et la gestion des utilisateurs...

Un acteur est toujours lié à une ou des action(s) du système. Ces actions sont appelées des **fonctionnalités**. Nous allons, dans la section suivante, identifier l'ensemble des fonctionnalités de notre système.

## II.1.2 Identification des fonctionnalités du système

Les besoins fonctionnels représentent les actions que doit exécuter un système: les fonctionnalités du système. Ce dernier ne sera opérationnel que lorsqu'il satisfait les besoins.

Dans le tableau, ci-dessous, sont répertoriées toutes les fonctionnalités identifiées, accompagnées de leurs acteurs.

Tableau 3: Identification des fonctionnalités

<b><u>Fonctionnalités</u></b>	<b><u>Déclarations</u></b>
<b>1. S'authentifier</b>	Officier d'Etat Civil, Agent d'état-civil, Administrateur
<b>2. Saisir une déclaration</b>	Officier d'Etat Civil ,Agent d'état-civil
<b>3. Enregistrer une déclaration</b>	Officier d'Etat Civil, Agent d'état-civil
<b>4. Visualiser une déclaration</b>	Officier d'Etat Civil, Agent d'état-civil
<b>5. Imprimer un déclaration d'état-civil</b>	Officier d'Etat Civil, Agent d'état-civil
<b>6. Gérer les Déclarations EC (ajouter, modifier et supprimer )</b>	Officier d'Etat Civil
<b>7. Gérer les Statistiques</b>	Officier d'Etat Civil, Administrateur
<b>8. Gérer les utilisateurs (ajouter, modifier, supprimer et droit)</b>	Administrateur
<b>9.Gérer les Années de Registre, Les Officiers, Les Centres et les Formations sanitaires (ajouter, modifier, supprimer et droit)</b>	Administrateur

Les liaisons entre acteurs et fonctionnalités sont représentées, en UML, par des diagrammes de cas d'utilisation. Nous allons, dans la section qui suit, élaborer les diagrammes de cas d'utilisation de notre système.

### **II.1.3 Diagrammes de cas d'utilisation**

Un diagramme de cas d'utilisation est un diagramme UML qui fournit une représentation graphique des exigences du système, il aide à identifier la façon dont les utilisateurs interagissent avec ce dernier.

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation.

Pour ce travail, nous présenterons les diagrammes de cas d'utilisation par groupe de fonctionnalités : une déclaration de naissance, une déclaration de mariage, une déclaration de décès, les affichages et impressions.

#### **Le diagramme de cas d'utilisation d'une déclaration de naissance**

Pour gérer une déclaration de naissance, l'O.E ou A.D doit avoir au préalable reçu les documents complets du déclarant et connaître le type de déclaration à saisir. Ensuite passer à la saisie des données en ouvrant le formulaire, s'il n'y a pas d'erreur, l'acte est enregistré.

Il est important de noter que l'exécution de tout cas d'utilisation nécessite, au préalable, une authentification.

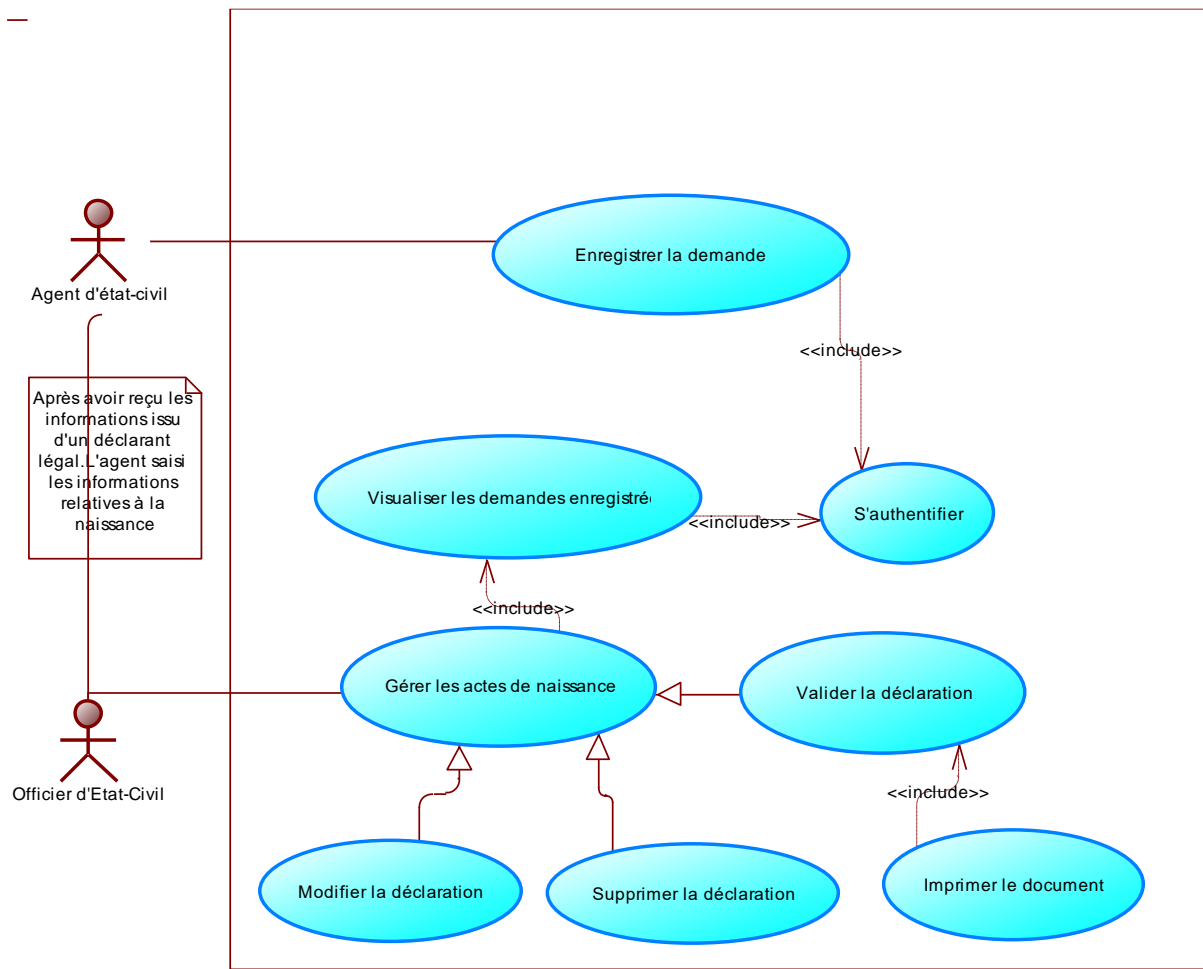


Figure 3 : Diagramme de cas d'utilisation d'une déclaration de naissance

### Le diagramme de cas d'utilisation d'une déclaration de mariage

La déclaration de mariage doit être faite devant l'O.E que le mariage soit célébré ou constaté.

Après avoir reçu au préalable les documents complets, rédiger le projet de mariage ensuite on prononce l'union. L'acte de mariage sera saisi en ouvrant le formulaire, s'il n'y a pas d'erreur, l'acte est enregistré.

Il est important de noter que l'exécution de tout cas d'utilisation nécessite, au préalable, une authentification.

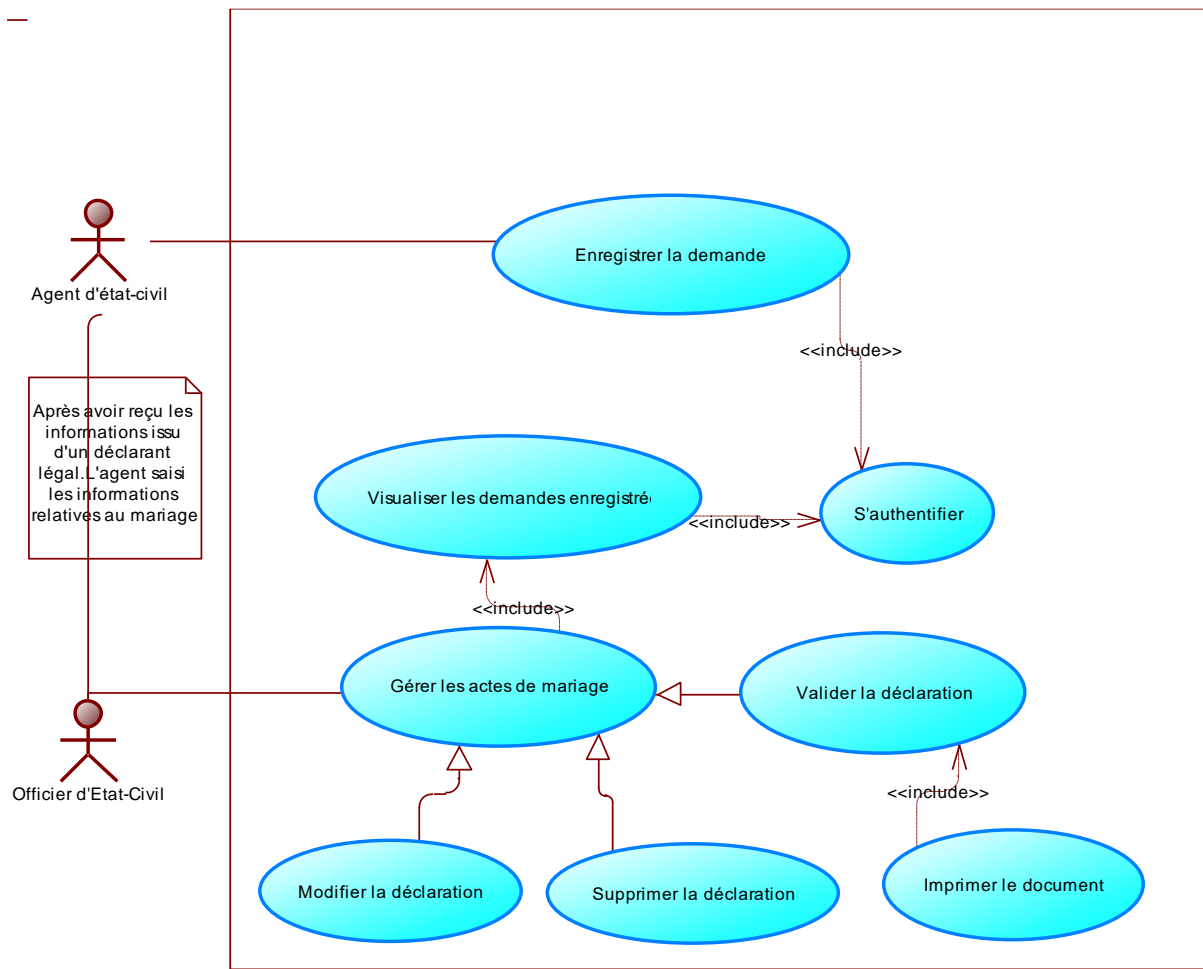


Figure 4: Diagramme de cas d'utilisation d'une déclaration de mariage

### Le diagramme de cas d'utilisation d'une déclaration de décès

Pour gérer une déclaration de décès, l'O.E ou A.D doit avoir au préalable reçu les documents complets du déclarant et connaître le type de déclaration à saisir. Ensuite passer à la saisie des données en ouvrant le formulaire, s'il n'y a pas d'erreur, l'acte est enregistré.

Il est important de noter que l'exécution de tout cas d'utilisation nécessite, au préalable, une authentification.

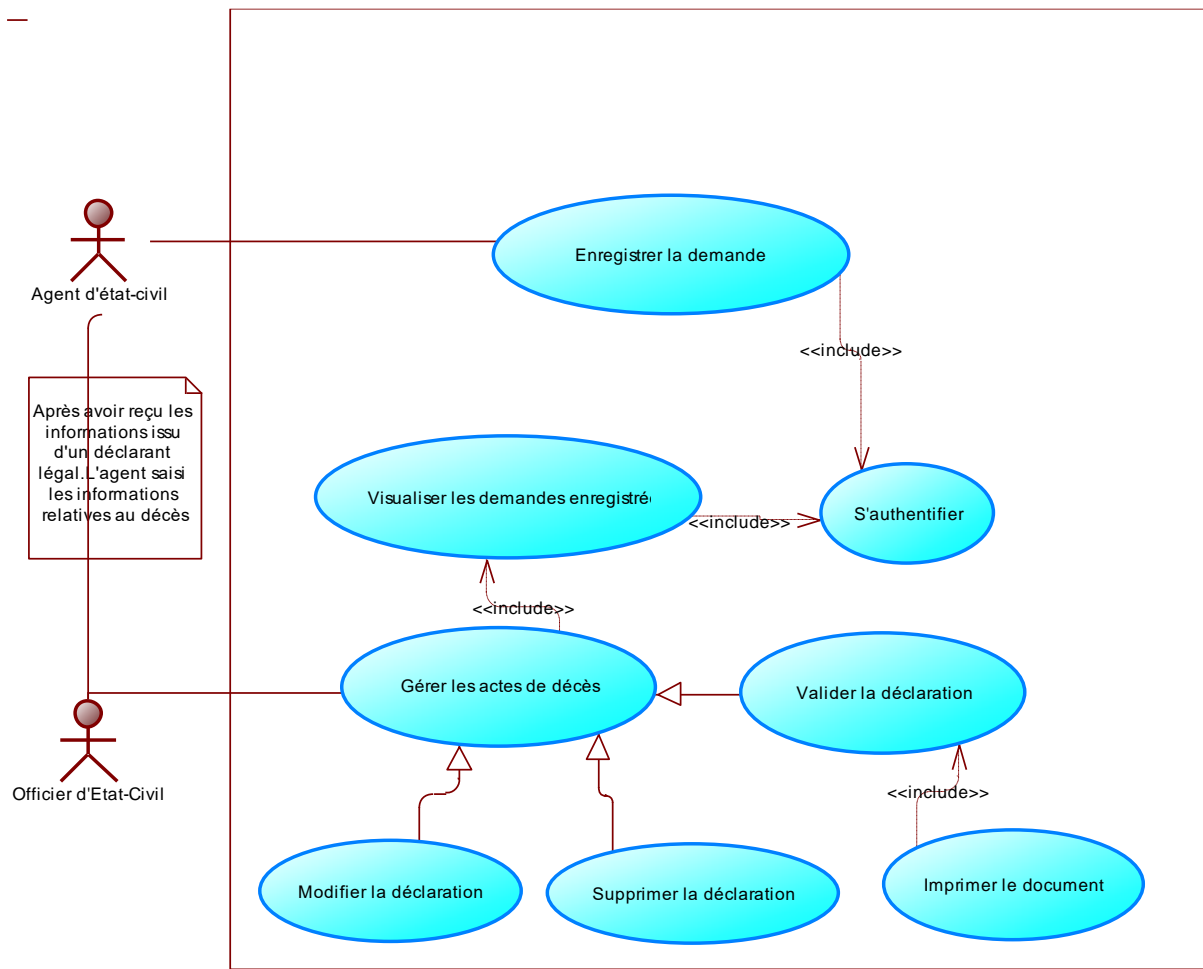


Figure 5: Diagramme de cas d'utilisation d'une déclaration de décès

### Le diagramme de cas d'utilisation pour l'administration du système

Le chef de service ou de la cellule informatique assure la gestion (ajout, modification et suppression) des utilisateurs, des années de registres, des centres d'état-civil, des formations sanitaires, des officiers d'état-civil et l'archivages. L'administrateur, en plus des tâches du chef de service informatique, assure la gestion de la base de données, sa sauvegarde, sa restauration et sa maintenance.

Une authentification est nécessaire avant toute opération sur le système.

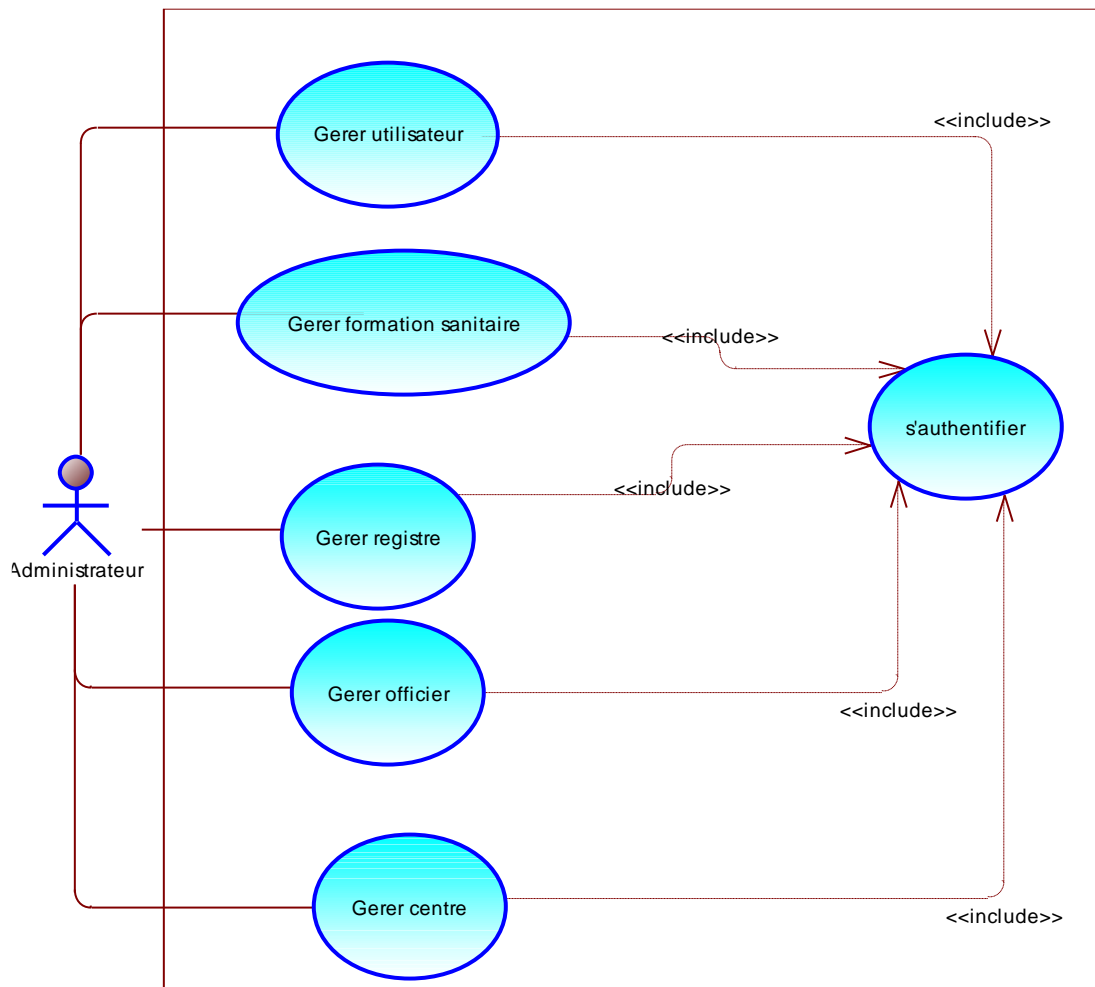


Figure 6: Diagramme de cas d'utilisation pour l'administrateur

Les besoins spécifiés nécessitent souvent d'être examinés en leurs différentes parties. C'est pour cela, dans la section qui va suivre, nous analyserons quelques besoins fonctionnels de notre système.

## II.2 Analyse des besoins fonctionnels du système

L'analyse des besoins fonctionnels est une démarche qui consiste à décrire les fonctionnalités du système. Dans cette partie, nous analysons les besoins fonctionnels qui semblent les plus complexes. Pour ce faire, nous commencerons par analyser le besoin d'authentification, ensuite, analyser celui de la gestion (officier d'Etat Civil, agent de saisie et administrateur) des étapes et enfin analyser le besoin du déclarant.

### II.2.1 Analyse de l'authentification

Dans cette section, nous décrivons d'abord le cas d'utilisation « s'authentifier », ensuite



élaborer ses diagrammes d'activité et de séquence.

## Description du cas d'utilisation « s'authentifier »

Le tableau ci-dessous permet de décrire le cas d'utilisation « s'authentifier ».

Tableau 4: Description de cas d'utilisation « s'authentifier »

<b>Description de cas d'utilisation « s'authentifier »</b>	
<b>Titre</b>	S'authentifier
<b>Résumé</b>	Permet de vérifier l'accès au système
<b>Acteurs</b>	Directeur et Officier d'état-civil, Chef du centre principal, Chef et Officier d'état-civil du centre secondaire, Administrateur, Agent de saisie
<b>Pré condition</b>	Avoir un compte d'utilisateur
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>▪ L'utilisateur saisit son pseudo et son mot de passe ;</li> <li>▪ Le système vérifie les informations saisies ;</li> <li>▪ Le système récupère le profil de l'utilisateur.</li> </ul>
<b>Post condition</b>	Affichage de la page d'accueil correspondant au profil de l'utilisateur
<b>Exception</b>	Saisie du pseudo ou du mot de passe incorrect

## Diagramme d'activités du cas d'utilisation « s'authentifier »

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Une activité est l'exécution d'une partie du cas d'utilisation. Elle est représentée par un rectangle aux bords arrondis.

Le cas d'utilisation « s'authentifier » regroupe plusieurs activités. Nous commençons par saisir un pseudo et un mot de passe. Après l'exécution de cette activité, le système passe à une vérification. Si les données saisies sont incorrectes, un message d'erreur s'affiche et l'activité « Saisir un pseudo et un mot de passe » doit être reprise. Par ailleurs, si les données saisies sont correctes, le système donne accès à la page d'accueil.

L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

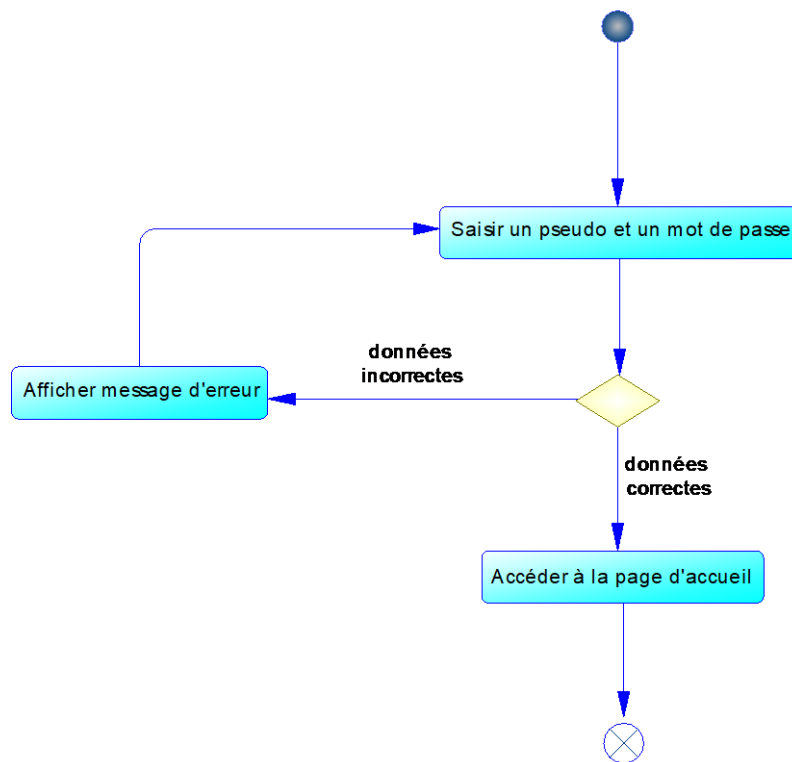


Figure 7: Diagramme d'activité du cas d'utilisation "s'authentifier"

### Diagramme de séquence du cas d'utilisation « s'authentifier »

Le diagramme de séquence est une représentation graphique d'interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Il permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme de cas d'utilisation.

Dans notre cas, nous représentons le scénario nominal. L'utilisateur saisit son pseudo et son mot de passe. Le système procède à une vérification. Après cela, le système affiche la page d'accueil.

La figure ci-dessous représente le diagramme de séquence illustrant le scénario nominal de l'authentification.

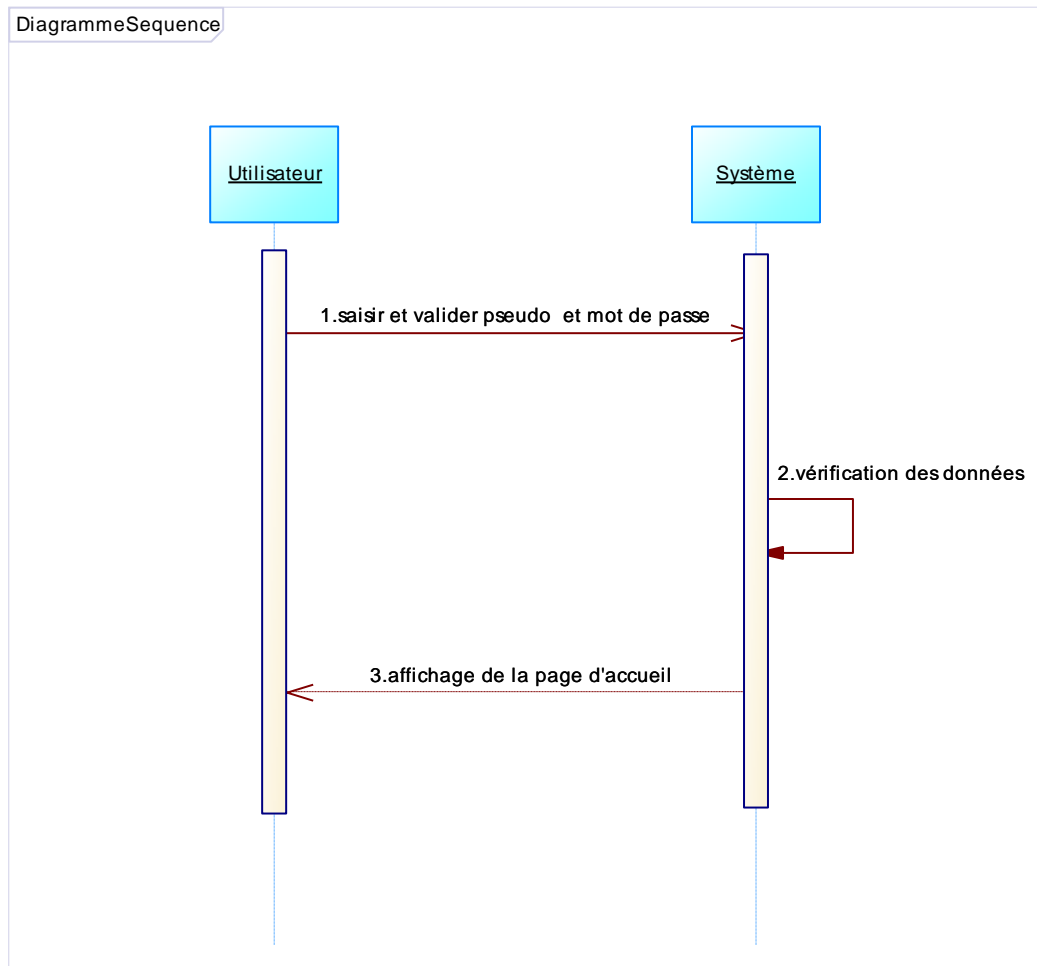


Figure 8: Diagramme de séquence du cas d'utilisation "s'authentifier"

Suite à l'analyse de l'authentification, nous allons, dans la section qui suit, décrire le besoin fonctionnel de l'ouverture d'étape.

## II.2.2 Analyse d'une déclaration de naissance

Dans cette partie, nous décrivons textuellement d'abord le cas d'utilisation « déclaration de naissance », ensuite son diagramme d'activité et enfin le diagramme de séquence.

### Description du cas d'utilisation « déclaration de naissance »

Le tableau ci-dessous permet de décrire le cas d'utilisation « déclaration de naissance ».

Tableau 5: Description de cas d'utilisation « déclaration de naissance »

<b>Description de cas d'utilisation « déclaration naissance »</b>	
<b>Titre</b>	Déclaration naissance
<b>Résumé</b>	Permet de faire une déclaration de naissance
<b>Acteurs</b>	Officier d'état-civil et Agent de saisie
<b>Pré condition</b>	Authentification, ouverture du formulaire demande pour la déclaration de naissance
<b>Scénario nominal</b>	Le système affiche le formulaire ; L'utilisateur saisit le type de déclaration, complète les autres champs et valide; Le système enregistre ; Message de confirmation de l'opération effectuée.
<b>Post condition</b>	L'étape est fermer ou faire une nouvelle déclaration
<b>Exception</b>	La déclaration de naissance est déjà saisie ou vous n'êtes pas autorisé à saisir une

### **Diagramme d'activité du cas d'utilisation « déclaration naissance »**

Le cas d'utilisation « déclaration de naissance » regroupe plusieurs activités en fonction du type de déclaration. Nous commençons par choisir dans la liste déroulante « le type de déclaration de naissance », ensuite le système passe à l'affichage des champs nécessaires. Une fois, cela est fait, on passe à la saisie des données. Après l'exécution du bouton enregistrer, le système passe à une vérification. Si une erreur est détectée, un message d'erreur s'affiche et l'agent de saisie doit l'identifier et la corriger. Par ailleurs, si la déclaration est enregistrée, le système redirection du formulaire vers une liste d'attente et l'O.E est chargé de valider ou de rejeter l'acte déclaré. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

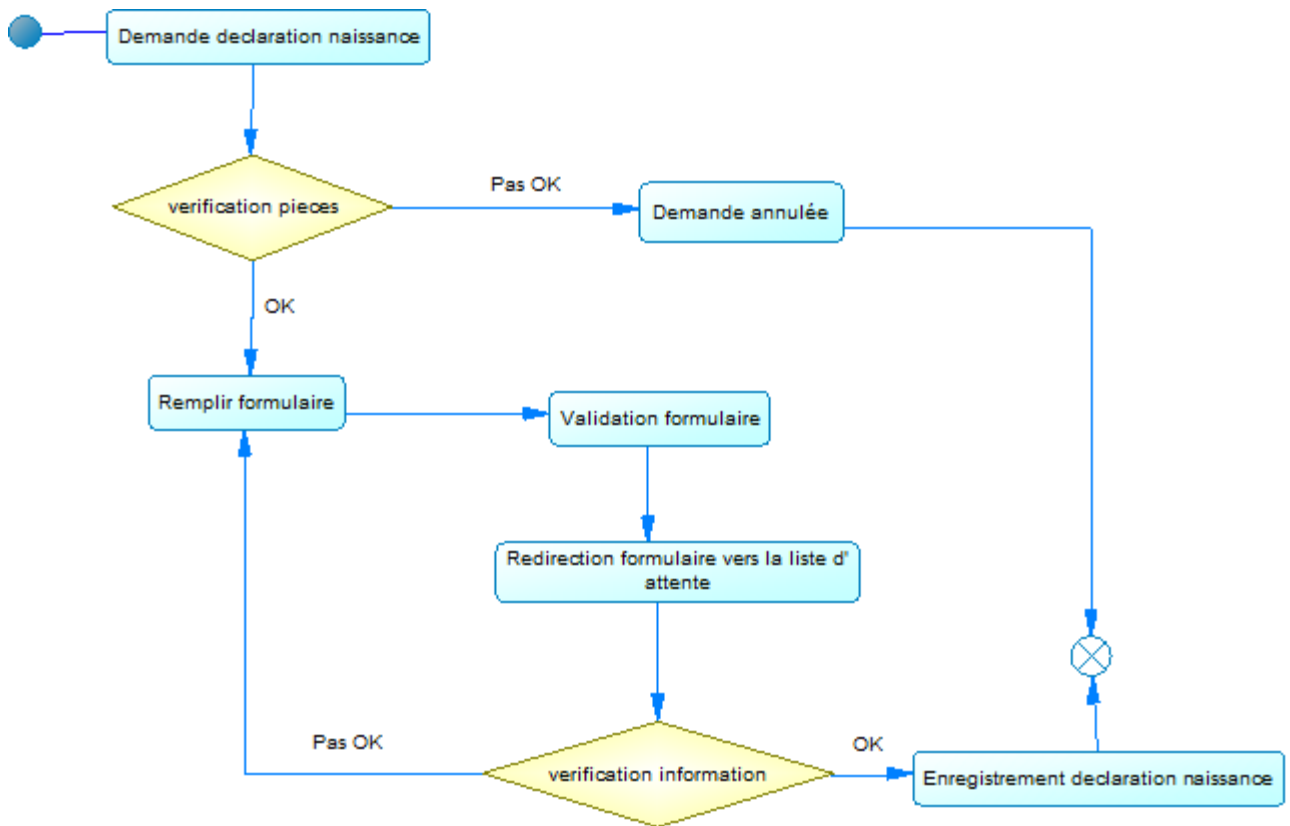


Figure 9: Diagramme d'activité du cas "déclaration de naissance"

### Diagramme de séquence du cas d'utilisation « déclaration de naissance »

Nous représentons, dans cette partie, le scénario nominal de déclaration de naissance. Après validation des documents reçus du déclarant. L'A.S passe à la saisie et la validation des informations du déclaré(e). Le système procède à une vérification des données. Si une erreur est détectée, un message d'erreur s'affiche et l'agent de saisie doit l'identifier et la corriger. Sinon la déclaration est enregistrée, le système redirection du formulaire vers une liste d'attente. Après cela, si tout se passe bien, L'O.E valide l'enregistrement de l'opération sinon il rejette l'acte déclaré. La figure ci- dessous représente le diagramme de séquence illustrant le scénario nominal du cas d'utilisation « déclaration de naissance ».

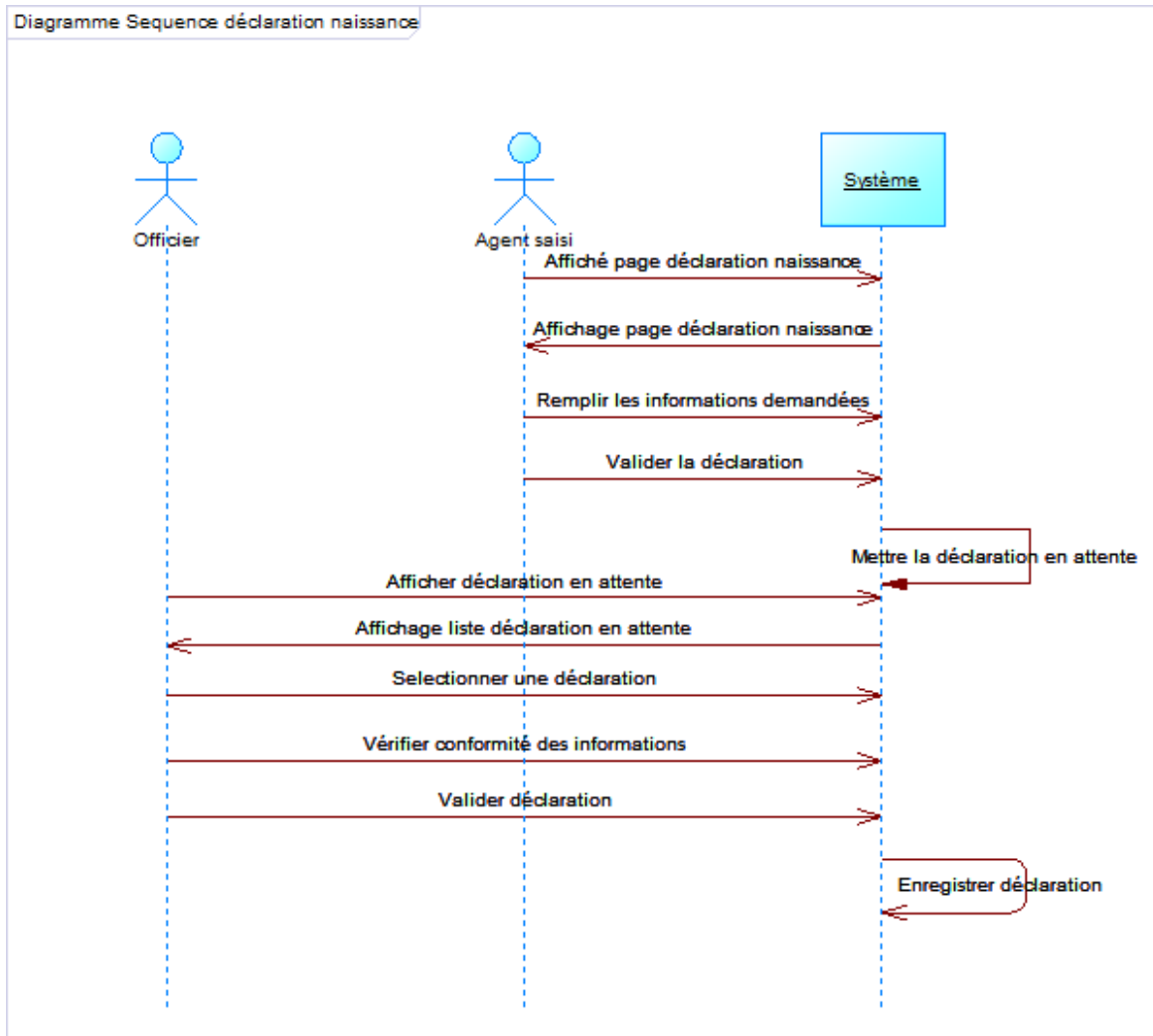


Figure 10: Diagramme de séquence du cas d'utilisation "déclaration de naissance"

Après l'analyse d'une déclaration de naissance, nous allons, dans la section suivante, décrire le besoin fonctionnel d'une déclaration de mariage.

### II.2.3 Analyse d'une déclaration de mariage

Dans cette partie, nous décrivons d'abord le cas d'utilisation « déclaration de mariage », ensuite le diagramme d'activité d'une déclaration de mariage et enfin le diagramme de séquence.

## Description du cas d'utilisation « déclaration de mariage »

Tableau 6: Description de cas d'utilisation « déclaration de mariage »

<b>Description de cas d'utilisation « déclaration de mariage »</b>	
<b>Titre</b>	Déclaration de mariage
<b>Résumé</b>	Permet de faire la déclaration de mariage
<b>Acteurs</b>	L'O.E et l'A.S
<b>Pré condition</b>	Authentification
<b>Scénario nominal</b>	L'O.E ou A.S déroule le menu Déclaration et clique sur Déclaration de mariage Le système affiche le formulaire  L'O.E ou A.S choisit le type de déclaration de mariage et complète les autres champs et valide;  Le système enregistre ; Message de confirmation de l'opération effectuée.
<b>Post condition</b>	L'étape est fermer ou faire une nouvelle déclaration
<b>Exception</b>	La déclaration de mariage est déjà saisie ou vous n'êtes pas autorisé à saisir une déclaration de mariage

## Diagramme d'activité du cas d'utilisation « déclaration de mariage »

Le cas d'utilisation « déclaration de mariage » regroupe plusieurs activités. Nous commençons par ouvrir le formulaire et choisir le « type de déclaration de mariage », ensuite compléter les champs suivant. Une fois, cela est fait, on passe l'activité « cliquer sur le bouton enregistrer ». Après l'exécution de ces activités, le système passe à une vérification. Si aucune erreur n'est signalée, l'acte est enregistré, sinon un message d'erreur s'affiche. Par ailleurs, si l'acte est enregistré, le système l'envoie dans la liste d'attente pour est valider par l'O.E. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

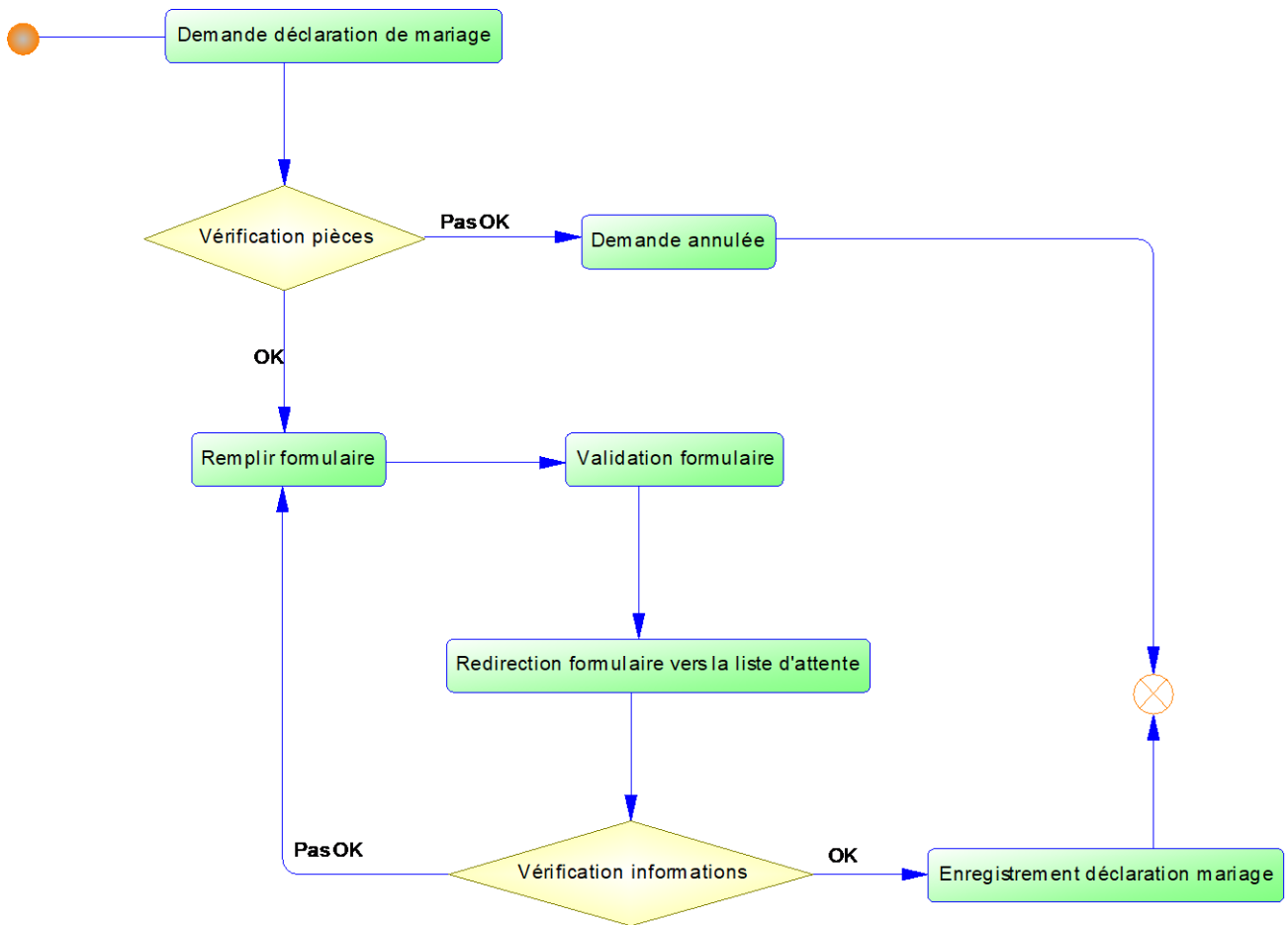


Figure 11: Diagramme d'activité du cas "déclaration de mariage"

### Diagramme de séquence du cas d'utilisation « déclaration de mariage »

Nous représentons, dans cette partie, le scénario nominal du cas d'utilisation « déclaration de mariage ». L'A.S demande l'interface déclaration de mariage. Le système procède à une vérification de droit d'accès. Après cela, si tout se passe bien, le système affiche le formulaire de saisi de l'acte de mariage. L'utilisateur choisit le type de mariage ensuite remplit les champs de saisi nécessaire à la validation de l'acte de mariage. La figure ci-dessous représente le diagramme de séquence illustrant le scénario nominal du cas d'utilisation « déclaration de mariage ».



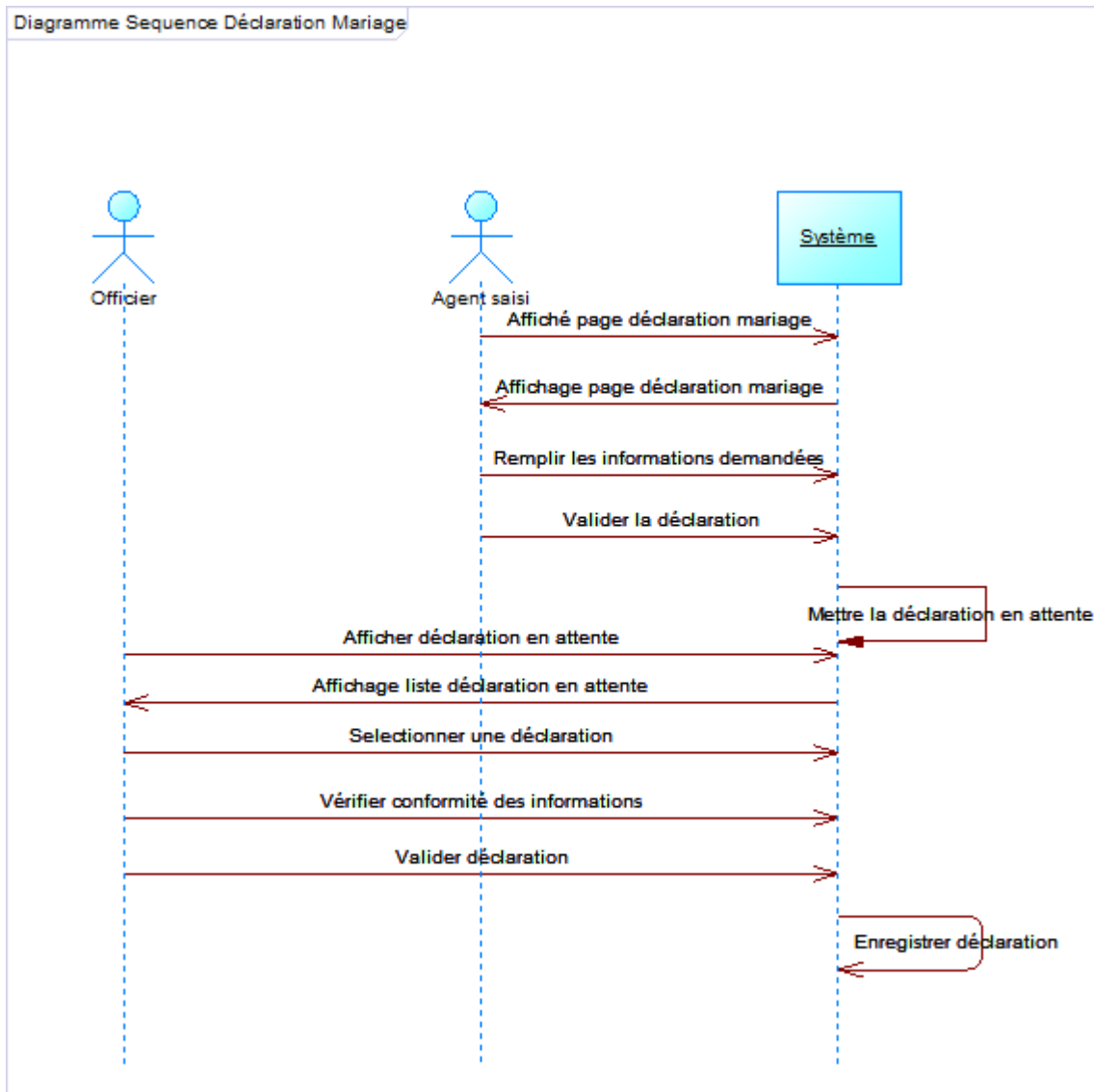


Figure 12: Diagramme de séquence du cas d'utilisation "déclaration mariage"

Ainsi après l'analyse d'une déclaration de mariage, nous allons, dans la section suivante, décrire le besoin fonctionnel d'une déclaration de décès.

## II.2.4 Analyse de la déclaration de décès

Dans cette partie, nous décrivons textuellement d'abord le cas d'utilisation « déclaration décès », ensuite leurs diagrammes d'activité et enfin les diagrammes de séquence.

### Description du cas d'utilisation « déclaration décès »

Le tableau ci-dessous permet de décrire le cas d'utilisation « déclaration décès ».

Tableau 7: Description de cas d'utilisation « déclaration décès »

<b>Description de cas d'utilisation « déclaration décès »</b>	
<b>Titre</b>	Déclaration décès
<b>Résumé</b>	Permet de faire une déclaration de décès
<b>Acteurs</b>	Officier d'état-civil et Agent de saisie
<b>Pré condition</b>	Authentification, ouverture du formulaire demande de déclaration de décès
<b>Scénario nominal</b>	Le système affiche le formulaire ; L'utilisateur saisit le type de déclaration, complète les autres champs et valide; Le système enregistre ; Message de confirmation de l'opération effectuée.
<b>Post condition</b>	L'étape est fermer ou faire une nouvelle déclaration
<b>Exception</b>	La déclaration de décès est déjà saisie ou vous n'êtes pas autorisé à saisir une déclaration de décès

### **Diagramme d'activité du cas d'utilisation « déclaration décès »**

Le cas d'utilisation « déclaration de décès » regroupe plusieurs activités en fonction du type de déclaration. Nous commençons par choisir dans la liste déroulante « le type de déclaration de décès », ensuite le système passe à l'affichage des champs nécessaires. Une fois, cela est fait, on passe à la saisie des données. Après l'exécution du bouton enregistrer, le système passe à une vérification. Si une erreur est détectée, un message d'erreur s'affiche et l'agent de saisie doit l'identifier et la corriger. Par ailleurs, si la déclaration est enregistrée, le système redirection du formulaire vers une liste d'attente et l'O.E est chargé de valider ou de rejeter l'acte déclaré. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

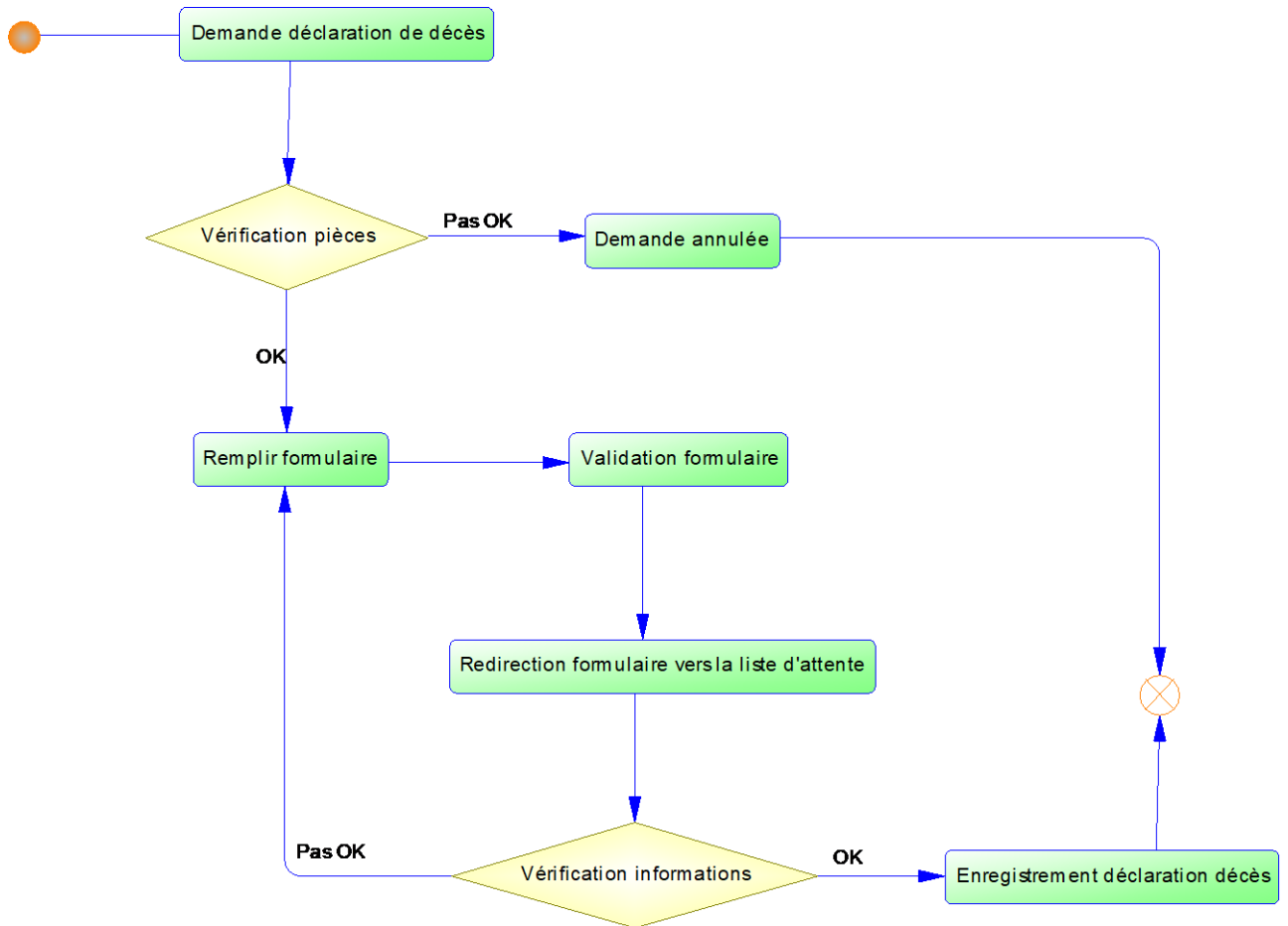


Figure 13: Diagramme d'activité du cas "déclaration de décès"

### Diagramme de séquence du cas d'utilisation « déclaration de décès »

Nous représentons, dans cette partie, le scénario nominal de déclaration de décès. Après validation des documents reçus du déclarant. L'A.S passe à la saisie et la validation des informations du déclaré(e). Le système procède à une vérification des données. Si une erreur est détectée, un message d'erreur s'affiche et l'agent de saisie doit l'identifier et la corriger. Sinon la déclaration est enregistrée, le système redirection du formulaire vers une liste d'attente. Après cela, si tout se passe bien, L'O.E valide l'enregistrement de l'opération sinon il rejette l'acte déclaré. La figure ci- dessous représente le diagramme de séquence illustrant le scénario nominal du cas d'utilisation « déclaration de décès »

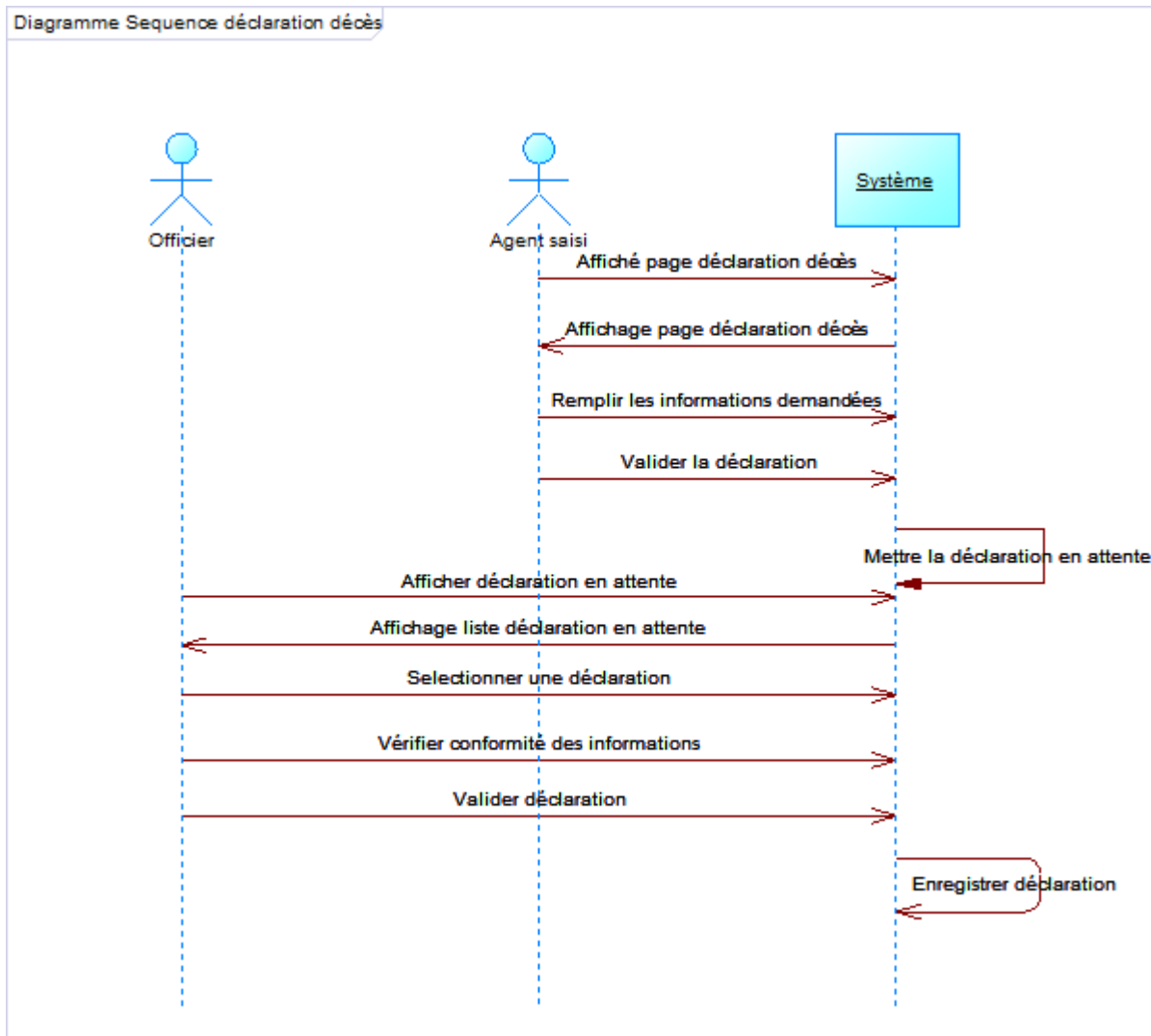


Figure 14: Diagramme de séquence du cas d'utilisation "déclaration de décès"

Dans ce chapitre, nous avons présenté la spécification et l'analyse des besoins fonctionnels accompagnés de la modélisation de notre système d'informations de par les diagrammes de cas d'utilisation, d'activités et de séquence dans le but d'aborder le chapitre suivant sur la conception de notre système.

## CHAPITRE III : CONCEPTION DU SYSTEME

La conception d'un système est une phase préliminaire et essentielle pour la concrétisation ou la réalisation d'une application. Il s'agit de mettre en œuvre un ensemble d'activités qui, à partir d'une demande d'informatisation d'un processus (demande qui peut aller de la simple question orale jusqu'au cahier des charges complet), permettent la conception, l'écriture et la mise au point d'une application (et donc de programmes informatiques) jusqu'à sa livraison ou déploiement au client ou demandeur.

Dans ce chapitre, nous parlerons d'abord de la conception générale de notre système en s'appuyant sur l'architecture, et les diagrammes de packages et de déploiement. Ensuite, nous abordons la conception détaillée en s'appuyant sur le diagramme de classes et l'établissement du dictionnaire de données.

### III.1 Conception générale

Dans cette section, nous aborderons l'architecture ainsi que les diagrammes de composants, de packages et de déploiement de notre système.

#### III.1.1 Architecture de l'application

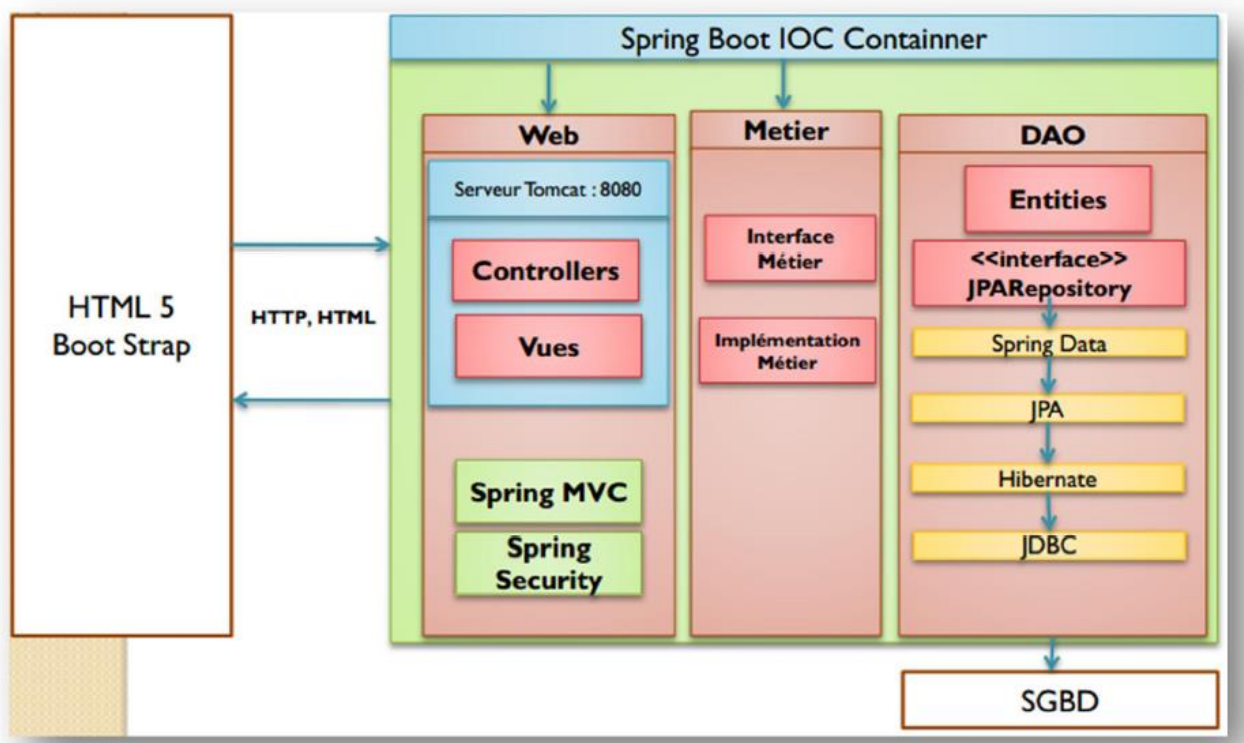
L'architecture d'une application décrit de façon symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions. Dans le cas de notre système, nous avons choisi une architecture multicouches Spring MVC avec Thymeleaf composée de trois couches incluant : la couche Web, la couche Metier et la couche DAO [7].

**La couche Web** : elle est la couche en contact avec l'utilisateur de l'application Web en utilisant le moteur de Template Thymeleaf. Celle-ci interagit avec l'application Web au travers de pages Web visualisées par un navigateur. C'est dans cette couche que se situe Spring MVC et uniquement dans cette couche. Nous y avons utilisé aussi du Bootstrap, JQuery, JavaScript etc...

**La couche métier :** elle implémente les règles de gestion de l'application, tels que le calcul du nombre de naissances ou du nombre d'impressions par jour. Cette couche utilise des données provenant de l'utilisateur via la couche [Web] et du SGBD via la couche [DAO] ;

**La couche DAO :** c'est la seconde couche de l'architecture de notre système. La couche DAO est basée sur Spring Data, JPA, Hibernate et JDBC. Elle est responsable de l'accès au système d'information pour lire ou modifier des données. Comme notre application stocke ses données dans une base de données MySQL, les classes DAO sont donc les classes qui contiennent le code qui permet d'échanger des informations avec la base de données.

La figure ci-dessous représente l'architecture de notre application (figure 15).



Source : <https://www.youtube.com/watch?v=ugKzvcpkmY4>

Figure 15: Architecture de l'application

### III.1.2 Diagramme de composants

Un composant représente une entité logicielle d'un système. Autrement dit, un composant est une entité autonome représentée par un classeur structuré, stéréotypé « component », comportant une ou plusieurs interfaces requises ou offertes. Le diagramme de composants

décrit l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases des données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les composants.

La figure ci-dessous représente le diagramme de composants de notre système.

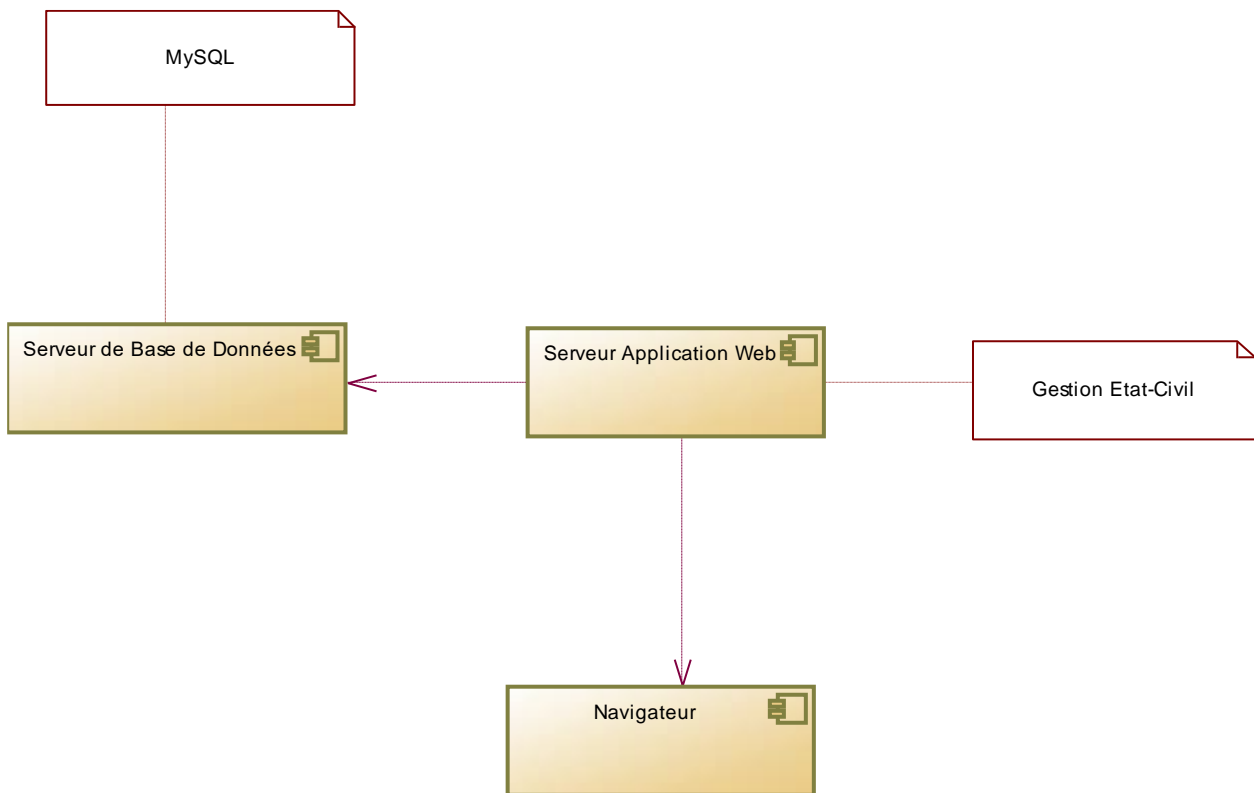


Figure 16: Diagramme de composants

### III.1.3 Diagramme de packages

Un package en UML (ou paquetage en français) sert à regrouper des éléments en un ensemble cohérent, et à fournir un espace de noms pour ces éléments. Un package peut contenir la plupart des éléments UML : classes, objets, cas d'utilisation, etc. Il peut également contenir d'autres packages, selon une organisation hiérarchique. L'intérêt des packages est de permettre de structurer les diagrammes et de donner une vision globale plus claire [13]. Ainsi, le diagramme de packages permet d'identifier les liens de généralisation et de dépendance entre les packages. La représentation ci-dessous est le diagramme de packages de

notre système.

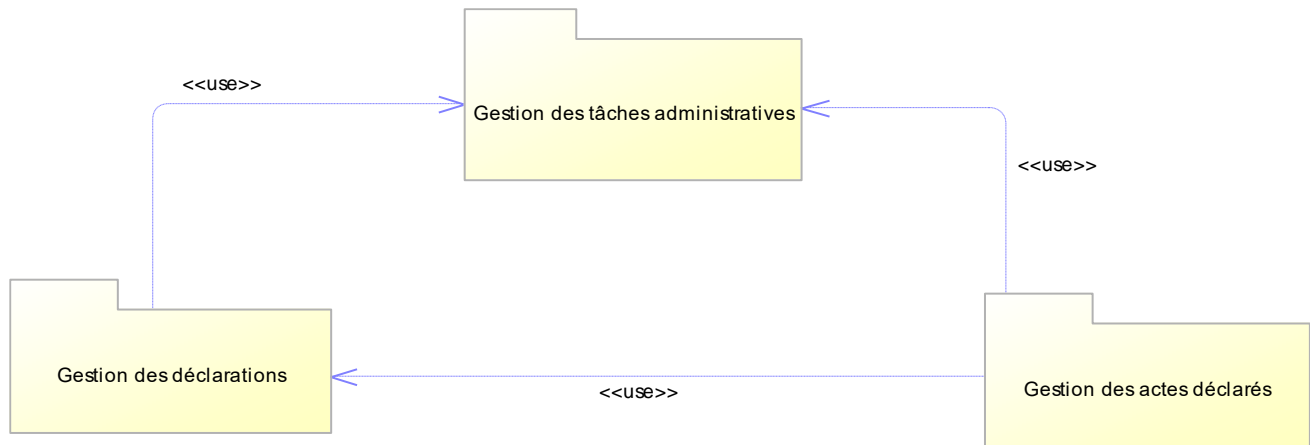


Figure 17: Diagramme de packages

Notre diagramme de packages (Figure 17) est composé de trois packages à savoir :

- ❖ **Gestion des déclarations** : dans ce package, nous assurons la gestion de l'ajout, afficher la liste des actes de naissances ou de mariages ou de décès et imprimer des actes de naissances, mariages et décès. Ce package utilise les classes « classe Naissance, classe Mariage, classe Décès, classe Jugement, classe Déclaré, classe Epoux, classe Epouse, classe Père, classe Mère, classe Témoin, classe Mention ».
- ❖ **Gestion des actes déclarés** : dans ce package, nous retrouvons les fonctionnalités comme : afficher la liste des actes de naissances ou de mariages ou de décès, valider des actes de naissances, mariages et décès, modification des actes de naissances, mariages et décès, suppression des actes de naissances, mariages et décès.
- ❖ **Gestion des tâches administratives** : dans ce package, nous assurons la gestion des utilisateurs : afficher liste, ajouter, modifier et supprimer des utilisateurs, de l'année des registres, des officiers état-civil, des formations sanitaires, des codes centres d'état-civil, mais aussi des tâches administratives telles que sauvegarder, restaurer, sécuriser la base de donnée et les serveurs.



### III.1.4 Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication serveurs et routeurs peuvent être précisées par stéréotype [14]. La représentation ci-dessous illustre le diagramme de déploiement de notre système.

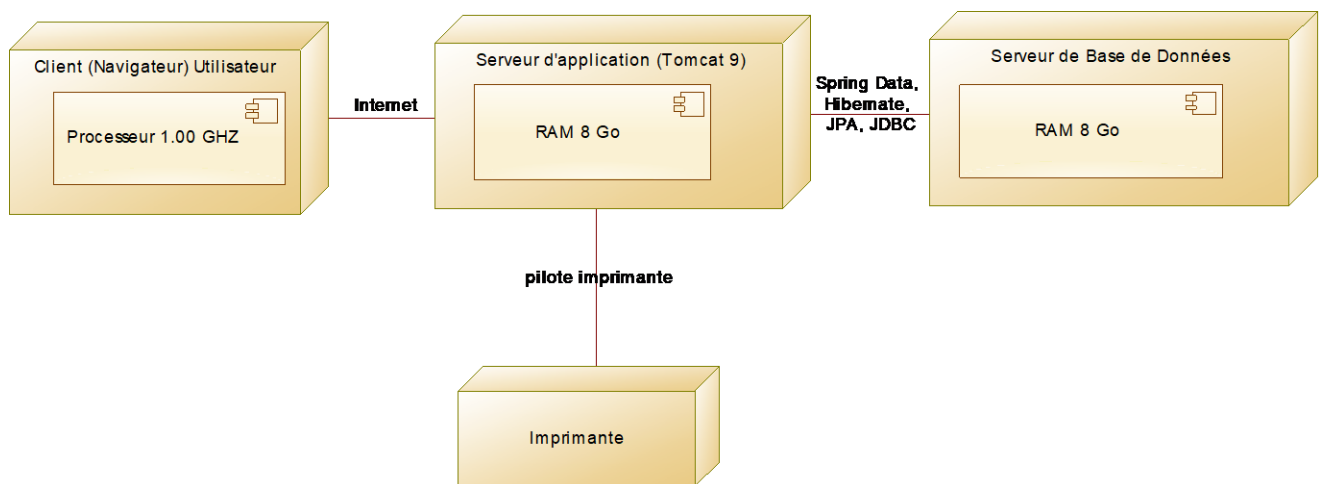


Figure 18: Diagramme de déploiement

Comme le montre la figure ci-dessus, notre application sera déployée sur du matériel avec des caractéristiques différentes de celles indiquées et qui respectent le minimum. Les serveurs sont contenus dans des machines de 8 Go de RAM.

Dans la section qui suit, nous aborderons la conception détaillée qui entrera dans les détails de la conception générale.

## III.2 Conception détaillée

La conception détaillée est une phase ultime de la conception de notre système d'information. Elle consiste à construire les classes qui interviennent dans le codage de l'application. Pour cela, nous abordons le diagramme de classes ainsi que le dictionnaire de données de notre

système.

### III.2.1 Diagramme de classes

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations ou méthodes) et de relations avec d'autres objets (les associations et les agrégations). Une classe de conception est composée par des attributs (données de la classe) et des méthodes (comportement de la classe). Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

Dans cette section, nous allons décrire le diagramme de classes de notre système. Pour cela, nous représenterons uniquement les classes participantes aux principales fonctionnalités à savoir la gestion de l'authentification et des utilisateurs, la gestion des déclarations et la gestion des actes d'état-civil. Le diagramme de classes complet est mis en *Annexe 1*.

#### III.2.1.1 Diagramme de classes participantes aux fonctionnalités de la gestion de l'authentification et des utilisateurs

Dans le diagramme de classe, ci-dessous, Utilisateur, Officier, Agent-de-saisi, Administrateur sont les classes participantes. Il y a trois utilisateurs que sont: l'Officier d'Etat-Civil, l'Agent de saisi et l'Administrateur. Ce diagramme permet de gérer les utilisateurs, leur authentification ainsi que leur espace de travail.

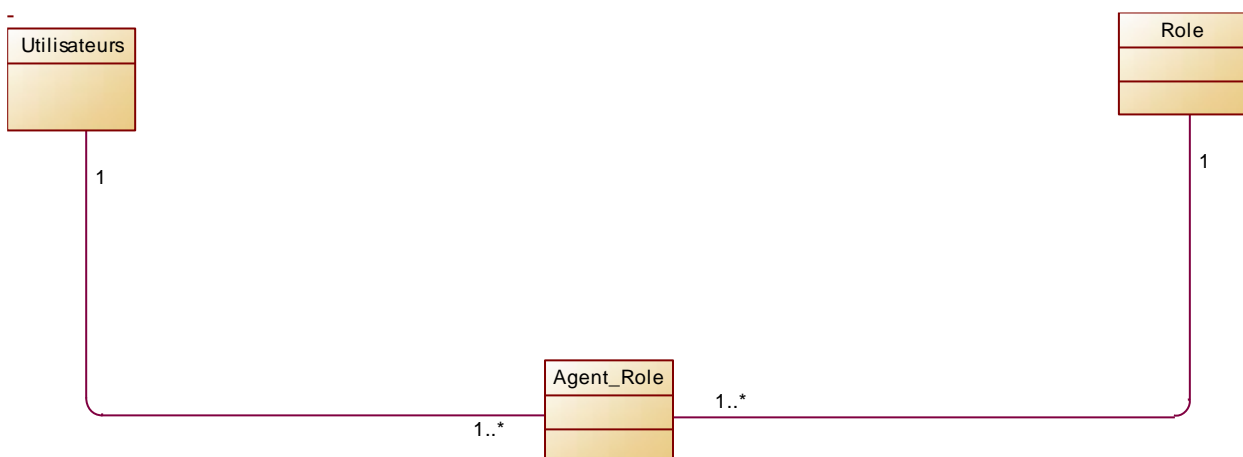


Figure 19: Diagramme de classes participantes aux fonctionnalités de la gestion l'authentification et des utilisateurs

### III.2.1.2 Diagramme de classes participantes aux fonctionnalités de la gestion des actes de naissance d'état-civil

Les classes participantes dans ce diagramme sont : Naissance, Utilisateurs, Registre, Formations sanitaires, Centre, Témoins et Mentions. Lors d'une déclaration de naissance toutes ces classes sont utilisées, sauf s'il s'agit d'un jugement que la formation sanitaire a omis. Pour ce faire, l'administrateur doit au préalable ajouter tous les années de registre, de formations sanitaires, du code de centre et des utilisateurs. (Figure 21)

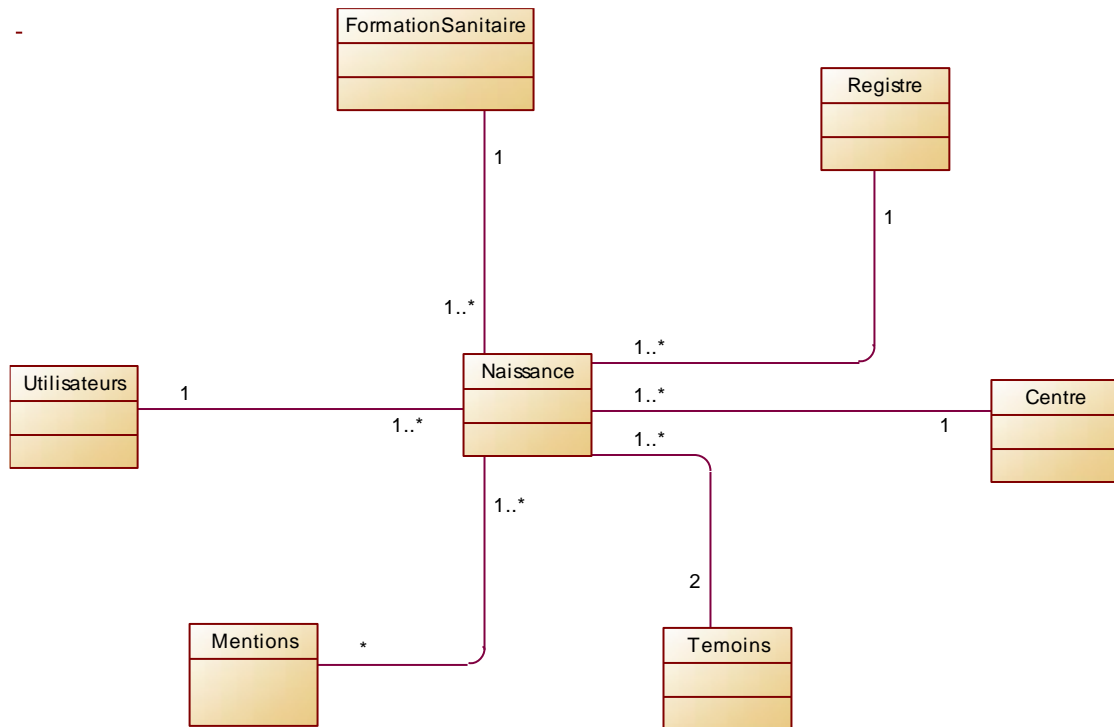


Figure 20: Diagramme de classes participantes aux fonctionnalités de la gestion des actes de naissances d'état-civil

### III.2.1.3 Diagramme de classes participantes aux fonctionnalités de la gestion des actes de mariage d'état-civil

Les classes participantes dans ce diagramme sont : Mariage, Utilisateurs, Registre, Epoux, Epouse, Témoins et Mentions. Lors d'une déclaration de mariage toutes ces classes sont utilisées, même s'il s'agit d'un jugement. Pour ce faire, l'administrateur doit au préalable ajouter toutes les années de registre, de formation sanitaires, du code de centre et des utilisateurs. (Figure 22)

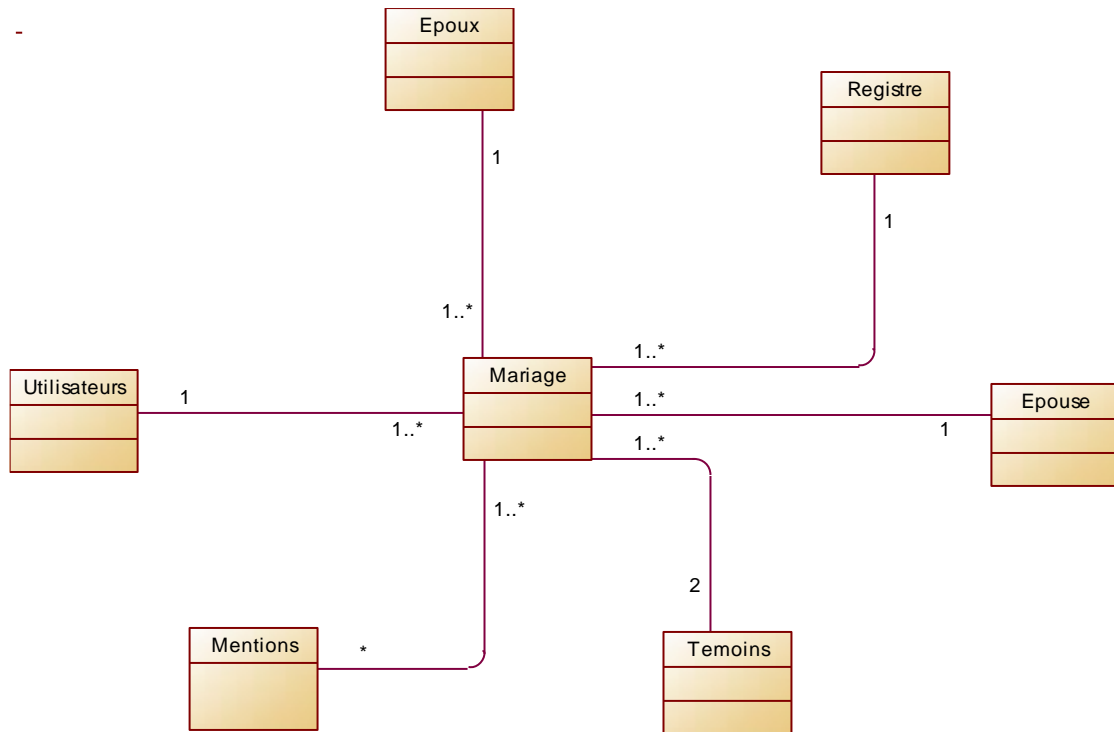


Figure 21: Diagramme de classes participantes aux fonctionnalités de la gestion des actes de mariages d'état-civil

### III.2.1.4 Diagramme de classes participantes aux fonctionnalités de la gestion des actes de décès d'état-civil

Les classes participantes dans ce diagramme sont : Décès, Utilisateurs, Registre, formations sanitaires, Centre, Témoins et Mentions. Lors d'une déclaration de décès toutes ces classes sont utilisées, sauf s'il s'agit d'un jugement que la formation sanitaire ne sera pas prise en compte. Pour ce faire, l'administrateur doit au préalable ajouter toutes les années de registre, de formations sanitaires, du code de centre et des utilisateurs.

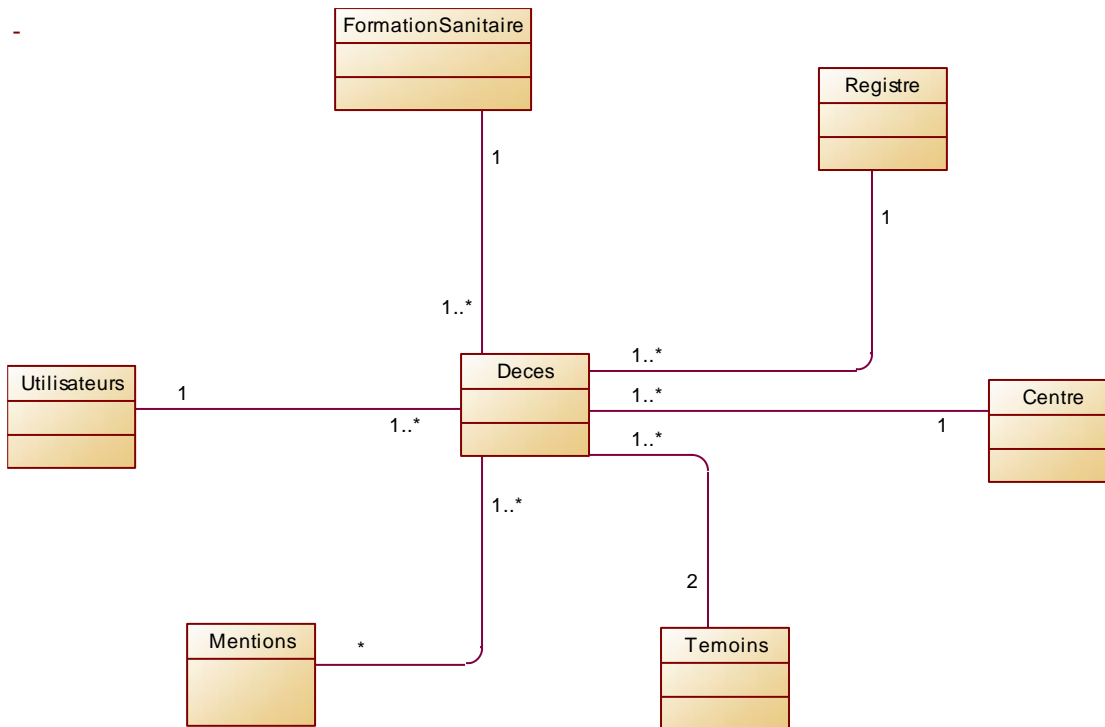


Figure 22: Diagramme de classes participant aux fonctionnalités de la gestion des actes de Décès d'état-civil

### III.2.2 Dictionnaire de données

Notre dictionnaire des données est une collection de métadonnées ou de données de référence nécessaire à la conception d'une base de données relationnelles. Il revêt une importance particulière, car il est le vocabulaire commun de la conception, mais aussi décrit des données importantes. Il est représenté, dans notre cas, par un tableau à quatre colonnes contenant le nom des tables, les champs (les données), le libellé ainsi que le type de données. Il faut noter que les données de la table Naissance sont identiques à celles de la table Décès. Raison pour laquelle on n'a représenté la table Décès dans notre dictionnaire de données

Le dictionnaire de données de notre système est représenté en *Annexe 2*.

Dans ce chapitre, nous avons évoqué la conception générale dans laquelle nous avons parlé de l'architecture du système, les diagrammes de package et de déploiement. Nous avons terminé par la conception détaillée où nous avons présenté les diagrammes de classes participant aux fonctionnalités, les diagrammes de classes ainsi que le dictionnaire de données. Tout cela va dans le sens de préparer l'implémentation et la réalisation de l'application qui feront l'objet du chapitre suivant.

## CHAPITRE IV: IMPLEMENTATION ET PRESENTATION DE L'APPLICATION

Ce chapitre est consacré à l'implémentation (ou codage) et à la présentation de l'application en tant que telle. Pour ce faire, nous allons d'abord présenter les outils et technologies (soient des outils de conception, de codage, de SGBDR et de langage de programmation) utilisés pour le codage. Ensuite, nous présenterons les principales interfaces graphiques de l'application.

### IV.1 Outils et technologies utilisés

Le développement d'une application telle que la nôtre (système de gestion des actes Etat Civil) nécessite l'utilisation de certains outils et technologies, que ce soit lors de la conception, du codage, de la gestion de base de données ou de l'implémentation.

#### IV.1.1 Outil de conception : PowerAMC

Un outil de conception est un logiciel qui permet de concevoir des diagrammes, comme les diagrammes de cas d'utilisation, de classes, de séquences, etc. **PowerAMC** est un des logiciels de conception parmi tant d'autres (ArgoUML, Papyrus, UML Designer, Software Ideas Modeler, etc.). C'est un outil d'UML et de génie logiciel assisté par ordinateur. Le modelleur prend en charge tous les 14 types de diagramme spécifiés dans UML 2.4. Il supporte également entre autres les diagrammes et standards, comme les organigrammes, les diagrammes de flux de données, etc [8].

#### IV.1.2 Environnement de développement : Eclipse

Eclipse est un environnement de développement intégré (IDE) écrit en Java, extensible par des greffons, multi-langages et multi-plates-formes. Conçu à l'origine pour le langage Java, il est devenu un IDE pour beaucoup de langages de programmation (C, C++, Python, PHP, ...).

Nous avons téléchargé et installé par marketplace (espace de téléchargement d'Eclipse) Spring Tool Suite 4 pour pouvoir travailler avec le Framework Spring Boot [13].

#### IV.1.3 Langage de programmation : JAVA

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. **Java** est un langage de

programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems est un langage de programmation parmi tant d'autres (PHP, C, C++, etc.).

Autrement dit, c'est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web [19].

#### **IV.1.4 Framework Spring: SPRING BOOT**

Spring est un Framework open-source (gratuit) destiné aux applications java (J2EE) pour les applications 3-tiers dont il facilite le développement et les tests.

Ce Framework est très apprécié par la communauté Java pour sa souplesse d'utilisation et sa modularité. En effet, il n'est pas nécessaire d'implémenter la totalité de Spring pour pouvoir l'utiliser, et c'est ce qui en fait son principal point fort, vous n'utilisez que ce dont vous avez besoin.

Spring et Spring MVC sont de bons outils de développement, mais nécessitent beaucoup de configurations. Les applications développées avec ces Framework possèdent beaucoup de fichiers XML qui indiquent les configurations des servlets, des vues, des contenus statiques, etc.

Spring Boot rend beaucoup plus facile le travail de configuration en proposant deux fonctionnalités principales :

- L'auto-configuration : une fonctionnalité très importante, elle permet de configurer automatiquement notre application à partir des jars de notre classpath. C'est-à-dire Spring Boot parcourt la liste des dépendances importées et produit les configurations nécessaires pour que tout fonctionne normalement.
- Les starters : ils apportent à notre projet un ensemble de dépendances communément utilisées pour un type de projet donné. Ils viennent compléter l'auto-configuration et font gagner énormément de temps. Un autre avantage et pas le moindre est la gestion des versions. Il suffit simplement d'ajouter une dépendance au starter de notre choix dans le fichier pom.xml, cette dernière va alors ajouter, à son tour, les éléments dont elle dépend, avec les bonnes versions.

Exemples :

- Spring-boot-starter-mail : pour les applications et services d'envoi de mails.
- Spring-boot-starter-thymeleaf : si nous souhaitons créer une application qui offre une interface utilisateur en utilisant le moteur de template thymeleaf [6].

#### **IV.1.5 Moteur de Template : THYMELEAF**

**Thymeleaf** est un moteur de Template, sous licence Apache 2.0, écrit en Java pouvant générer du XML/XHTML/HTML5. Thymeleaf peut être utilisé dans un environnement web (utilisant l'API

Servlet) ou non web. Son but principal est d'être utilisé dans un environnement web pour la génération de vue pour les applications web basées sur le modèle MVC [21].

#### IV.1.6 Tomcat

Tomcat est un serveur HTTP à part entière. De plus, il gère les servlets et les JSP (par un compilateur Jasper compilant les pages JSP pour en faire des servlets). Il a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant [15].

#### IV.1.7 Langage de requête : SQL

Le langage de requête **SQL** (Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles [12].

#### IV.1.8 Frameworks: Bootstrap et jQuery

En programmation informatique, un framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel [18].

L'utilité d'un framework est d'éviter de passer du temps à développement ce qui a déjà été fait par d'autres, souvent plus compétents, et qui a en plus utilisé et validé par de nombreux utilisateurs. **Bootstrap** et **jQuery** sont des frameworks parmi tant d'autres (Symfony, Zend Framework, Laravel, etc.).

**Bootstrap** est une collection d'outils utile à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option [19].

**Query** est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web [11]

#### IV.1.9 Le SGBDR : MySQL

MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous double licence libre et propriétaire. MySQL fait partie des logiciels de gestion de bases de données les plus utilisés au monde, autant par le grand public que par des professionnels. Parmi ses concurrents nous pouvons citer : Oracle Database, PostgreSQL, Microsoft SQL Server, MariaDB, ... Dans le cadre de



notre projet nous avons utilisé MySQL qui est contenu dans **XAMPP**, ainsi l'installation de ce dernier nous permet d'utiliser MySQL [9].

## **IV.2 Implémentation**

La phase de l'implémentation consiste au codage (ou à la programmation) du système. Pour ce faire, nous allons d'abord présenter l'implémentation de la base de données, ensuite celle de l'application.

### **IV.2.1 Implémentation de la base de données**

L'implémentation de la base de données consiste d'abord à mettre en place le modèle logique de données pour ensuite passer à l'implémentation de ce dernier.

#### **IV.2.1.1 Modèle physique de données (MPD)**

Le modèle physique de données (MPD) consiste à implémenter une base de données dans un SGBDR, c'est-à-dire la traduire dans un langage de définition de données. Le langage utilisé pour ce type d'opération est le SQL.

La migration des clés primaires dépend des cardinalités des tables. Les clés primaires migrent vers les tables où les cardinalités sont minimales. Dans ce modèle, les clés primaires sont soulignées tandis que les clés étrangères sont précédées par un dièse « # ».

La figure ci-dessous représente notre MPD

**Acte**(idActe, numActe, #registre, dateDeclaration, #officier, #jugement, #centre, scanActe, scanCINPereEpoux, scanCINMereEpouse, scanAccoucheDeces, scanTemoins1, scanTemoins2, scanTemoins3, scanTemoins4, validate, dateModification, #user)

**Naissance**(#id\_acte, date\_declaration, date\_modification, num\_acte, scan\_accouche\_deces, scan\_acte, scancinmere\_epouse, scancinpere\_epoux, scan\_temoins1, scan\_temoins2, scan\_temoins3, scan\_temoins4, validate, #centre\_id, #jugement\_id, #officier\_id, #registre\_id, #user\_id, declarationn, #declarant\_id, #declaree\_id, #formationsanitaire\_id, #mere\_id, #pere\_id)

**Mariage**(#id\_acte, date\_declaration, date\_modification, num\_acte, scan\_accouche\_deces, scan\_acte, scancinmere\_epouse, scancinpere\_epoux, scan\_temoins1, scan\_temoins2, scan\_temoins3, scan\_temoins4, validate, #centre\_id, #jugement\_id, #officier\_id, #registre\_id, #user\_id, coutume, declarationm, heure\_mariage, lieu\_mariage, montant\_dot, option\_mariag, regime\_mariag, scan\_coutumiere, #epouse\_id, #epoux\_id, #mere\_id, #pere\_id, date\_mariage)

**Deces**(#id\_acte, date\_declaration, date\_modification, num\_acte, scan\_accouche\_deces, scan\_acte, scancinmere\_epouse, scancinpere\_epoux, scan\_temoins1, scan\_temoins2, scan\_temoins3, scan\_temoins4, validate, #centre\_id, #jugement\_id, #officier\_id, #registre\_id, #user\_id, declaration\_deces, #declarant\_id, #defunt\_id, #formationsanitaire\_id, #mere\_id, #pere\_id)

**Registre**(id\_registre, active, annee)

**Centre**(id\_centre, code\_centre, collectivite\_locale, departement, libelle, region)

**FormationSanitaire**(id\_form\_san, code\_formation, libelle, type\_formation)

**Declaree**(id\_declaree, date\_naissance\_declare, heure\_naissance\_declare, lieu\_naissance\_declare, nom\_declare, prenom\_declare, sexe)

**Defunt**(id\_defunt, date\_deces, heure\_deces, lieu\_deces, nom\_defunt, prenom\_defunt, sexe\_defunt)

**Epoux**(id\_epoux, date\_naissance\_epoux, domicile\_epoux, lieu\_naissance\_epoux, nationalite\_epoux, nom\_epoux, num\_piece\_epoux, prenom\_epoux, profession\_epoux, typepiece\_epoux)

**Epouse**(id\_epouse, date\_naissance\_epouse, domicile\_epouse, lieu\_naissance\_epouse, nationalite\_epouse, nom\_epouse, num\_piece\_epouse, prenom\_epouse, profession\_epouse, typepiece\_epouse)

Figure 23: Schéma relationnel du MPD

### IV.2.1.2 Etablissement de connexion à la base de données

La connexion à la base de données nécessite le démarrage du serveur XAMPP. A ce niveau nous pourrons nous connecter à la base de données nommée « db\_Etatcivilzig ». Ce nom sera mentionné lors de la configuration du fichier application.properties (Figure 24) pour établir la connexion.

```

1 spring.datasource.url=jdbc:mysql://localhost:3306/db_eczig?useUnicode=true&use:
2 spring.datasource.username=root
3 spring.datasource.password=
4
5 # Show or not log for each sql query
6 spring.jpa.show-sql = true
7
8 # Hibernate ddl auto (create, create-drop, update)
9 spring.jpa.hibernate.ddl-auto =update
10 # The SQL dialect makes Hibernate generate better SQL for the chosen database
11 spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
12 spring.jpa.properties.hibernate.id.new_generator_mappings=false
13 #security.basic.enabled=false
14 #spring.servlet.multipart.max-file-size:10000KB
15 #spring.servlet.multipart.max-request-size:10000KB
16 dir.images:../images

```

Figure 24: Exemple de fichier application.properties

Dans la section qui suit, nous parlons d'implémentation de l'application.

### IV.2.2 Implémentation de l'application

L'implémentation est une phase du projet qui consiste à coder l'application en tant que telle.

#### IV.2.2.1 Création et arborescence de l'application

Lors de la création d'une application web Java EE avec Spring Boot dans Eclipse, nous avons certains choix à faire par exemples.

- Le type: Maven ou Gradle qui sont des outils pour gérer les bibliothèques (Maven).
- Le packaging de l'application: JAR ou WAR, fichier autonome et exécutable pour le premier ou fichier pour le déploiement sur un conteneur de servlet externe (JAR pour l'instant).
- La version de Java: Java Version (Java 8).
- Le langage (langage en anglais): Java, Kotlin, Groovy (Java).
- La version de Spring Boot: Spring Boot Version (2.1.6. RELEASE).
- Les dependances nécessaires pour application: Spring Web, Spring Data JPA,

MySQL, Thymeleaf, Spring Security, DevTools, etc. Il n'est pas nécessaire d'ajouter d'un seul coup toutes les dépendances, il est possible de le faire après dans le fichier pom.xml.

La figure (Figure 25) nous montre de l'arborescence générale de notre application.

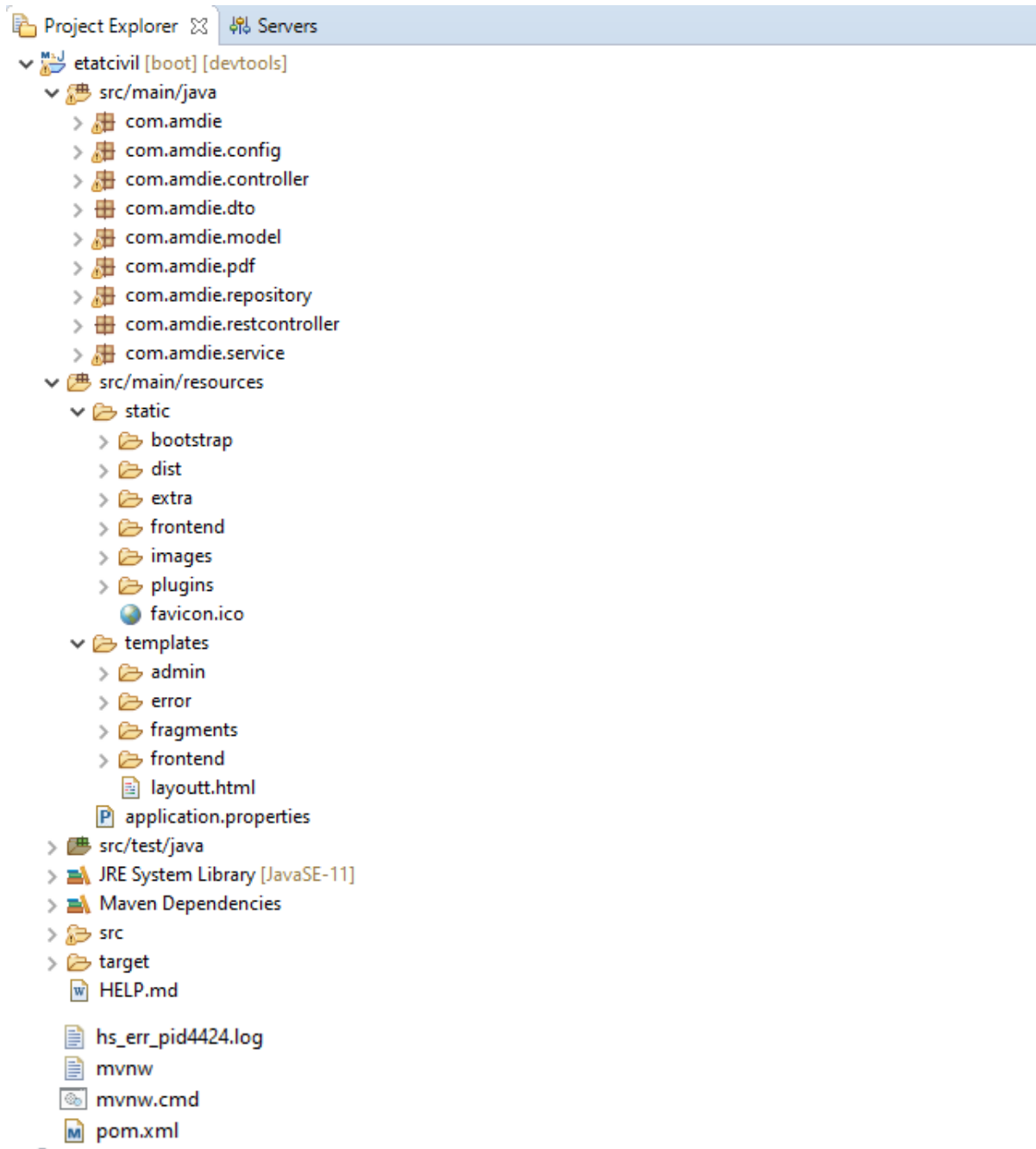


Figure 25: l'arborescence générale de notre application

### IV.2.2.2 La classe « Naissance » : Modèle

Avec JPA (Java Persistence API) les programmeurs n'ont plus besoin d'être des spécialistes en base de données ou d'un SGBD quelconque. JPA leur permet d'implémenter les classes (entités) qui seront transformées en tables dans une base de données. JPA permet aux programmeurs d'utiliser des notions avancées en base de données tout en restant programmeurs.

@Entity, signifie que cette classe est une table. JPA créera une table de même nom que la classe dans notre base de données.

@Table (name= "nom de la table"), permet de donner un nom différent de la classe à la table.

@Id, signifie que l'attribut qui est en dessous est la clé primaire de la table.

@GeneratedValue, signifie que la clé primaire est générée automatiquement par le SGBD.

@OneToMany, traduit la relation : un élément d'une entité est associé à un ou plusieurs éléments d'une autre entité.

La figure ci-dessous représente l'entité Naissance permettant d'afficher l'ensemble des données pour un acte de naissance.

```

1 package com.amdie.model;
2
3 import java.util.Date;
4
5
6
7
8 @Entity
9 @DiscriminatorValue("NAI")
10 public class Naissance extends Acte {
11
12     /**
13      *
14      */
15     private static final long serialVersionUID = 1L;
16
17     private String declarationN;
18     @ManyToOne(cascade = CascadeType.ALL)
19     @JoinColumn(name="DECLAREE_ID")
20     private Declaree declaree;
21     @ManyToOne(cascade = CascadeType.ALL)
22     @JoinColumn(name="DECLARANT_ID")
23     private Declarant declarant;
24     @ManyToOne(cascade = CascadeType.ALL)
25     @JoinColumn(name="PERE_ID")
26     private Pere pere;
27     @ManyToOne(cascade = CascadeType.ALL)
28     @JoinColumn(name="MERE_ID")
29     private Mere mere;
30     @ManyToOne
31     @JoinColumn(name="FORMATIONSANITAIRE_ID")
32     private FormationSanitaire formationSanitaire;
33
34     @OneToMany(mappedBy="naissance")
35     private Collection<Temoin> temoins;
36
37     @OneToMany(mappedBy="naissance")
38     private Collection<Mention> mentions;
39
40     public Naissance() {
41         super();
42         // TODO Auto-generated constructor stub
43     }
44
45
46
47
48
49
50
51
52
53

```

Figure 26 : Class « Naissance » : Modèle

### IV2.2.3 L'interface « NaissanceRepository » : Interface

Toutes les interfaces de notre application héritent de l'interface JpaRepository. Ceci permet l'utilisation des opérations CRUD (Create, Read, Update et Delete) auto-générées c'est-à-dire les opérations (insertion, lecture, modification et suppression), vers la base de données sans avoir à écrire la moindre requête, ni même la moindre implémentation DAO (en anglais Data Access Object ou Objet d'Accès aux Données en français).

Pour certaines opérations spécifiques ou particulières, nous avons utilisé des requêtes HQL (Hibernate Query Language) qui est un langage de requête du Framework Hibernate. Il est presque similaire au SQL.

La figure (Figure 28) ci-dessous, nous montre un exemple d'implémentation d'une interface.

```

14 public interface NaissanceRepository extends JpaRepository<Naissance, Long>{
15
16     @Query("Select n from Naissance n where n.numActe =:x and n.registre.annee=:y ")
17     Naissance existeDejaNaissance(@Param("x") String numActe, @Param("y") String annee);
18
19     /* @Query("Select a from Naissance a where a.decleree.prenomDeclare like :x OR a.decleree.nomDeclare l:
20     public Page<Naissance> rechercherNaissance( @Param("x") String mc, Pageable pageable);
21     */
22     @Query("Select a from Naissance a where a.numActe like:x AND a.registre.annee like:y AND a.centre.co
23     public Naissance rechercherNaissanceNum( @Param("x") String numActe, @Param("y") String annee,@Param(
24     |
25
26     @Query("Select a from Naissance a where a.numActe like :x OR CONCAT(a.decleree.prenomDeclare,' ',a.de
27     public Page<Naissance> rechercherNaissance( @Param("x") String mc, @Param("y") String annee,@Param("z
28
29
30 // @Query("Select a from Naissance a where a.idActe=:ab OR a.numActe=:b OR a.dateDeclaration=:c OR a.de
31 // Page<Naissance> impressionNaissance(@Param("ab") Long id,@Param("b")String numActe,@Param("c") Date dat
32
33     @Query("Select a from Naissance a where a.idActe=:aaa OR a.numActe=:b AND a.registre.annee=:tt AND a.c
34     + " OR a.decleree.dateNaissanceDeclare=:j OR a.decleree.heureNaissanceDeclare=:k OR a.decleree
35     + " OR a.pere.professionPere=:s OR a.mere.prenomMere=:t OR a.mere.nomMere=:u OR a.mere.dateNa
36     + " OR a.declarant.dateNaissanceDeclarant=:cc OR a.declarant.lieuNaissanceDeclarant=:dd OR a.d
37     + " OR a.scanAccoucheDeces=:oo OR a.scanCINPereEpoux=:pp OR a.scanCINMereEpouse=:qq OR a.scanT
38     public Naissance listeRechercherNaissance(@Param("aaa") Long idActe,@Param("b") String numActe,@Param(
39     @Param("i") String nomDeclare, @Param("j") Date dateNaissance,@Param("k") String heureNaissanc
40     @Param("r") String domicilePere,@Param("s") String professionPere,@Param("t")String prenomMere
41     @Param("aa") String prenomDeclarant,@Param("bb") String nomDeclarant, @Param("cc")Date dateNai:
42     @Param("oo") String acteAccoucheNaissance,@Param("pp") String cinPereEpoux, @Param("qq") Str:
43     @Param("rr") String cinTemoins1, @Param("ss") String cinTemoins2);
44
45     @Query("Select count(a.idActe) from Naissance a where a.declarationN=:x AND a.decleree.sexe=:y AND l
46     public int nbreDeclNormalHomme( @Param("x") String typeDeclaration, @Param("y") String sexe, @Param("
47
48     @Query("Select count(a.idActe) from Naissance a where a.declarationN=:x AND a.decleree.sexe=:y AND
49     public int nbreDeclNormalFemme( @Param("x") String typeDeclaration, @Param("y") String sexe, @Param("
50
51     @Query("Select count(a.idActe) from Naissance a where a.declarationN=:x AND a.decleree.sexe=:y AND l
52     public int nbreDeclTartivHomme( @Param("x") String typeDeclaration, @Param("y") String sexe, @Param("
53
54     @Query("Select count(a.idActe) from Naissance a where a.declarationN=:x AND a.decleree.sexe=:y AND l

```

Figure 27: Implémentation d'une interface

### IV.2.2.4 La classe « NaissanceController » : Contrôleur

Toute application web développée avec Spring Boot possède un contrôleur frontal par défaut « DispatcherServlet » qui intercepte toutes les demandes entrantes et les transfère à tous les autres

Dématérialisation des procédures et documents administratifs de l'état-civil de Ziguinchor composants (contrôleurs) de l'application. Les autres contrôleurs sont codés par le développeur et sont des fichiers java avec l'annotation « @Controller » placée avant le nom de la classe.

D'autres annotations sont utilisées dans notre contrôleur, comme :

@Autowired, qui signifie que la dépendance d'objet est implicitement injectée.

@Secured, permet de déterminer les utilisateurs qui peuvent appeler la méthode en-dessous

@RequestMapping, elle est utilisée pour mapper des requêtes Web sur des classes et des méthodes de gestionnaires spécifiques.

```
121 //l'affichage du formulaire de naissance
122 @GetMapping("/add")
123 public String addNaissance(Naissance naissance, Model model){
124     List<Registre> registreList = registreRepository.findByActive();
125     List<Centre> centreList = centreRepository.findAll();
126     List<FormationSanitaire> formSanList = formSanRepository.findAll();
127     List<Officier> officierList = officierRepository.findAll();
128
129     model.addAttribute("Registre", registreList);
130     model.addAttribute("DeclarationNaissance", getDeclarationNaissance());
131     model.addAttribute("Centre", centreList);
132     model.addAttribute("FormationSanitaire", formSanList);
133     model.addAttribute("Officier", officierList);
134
135     model.addAttribute("declaree", new Declaree());
136     model.addAttribute("jugement", new Jugement());
137
138     model.addAttribute("TypePiece", getTypePiece());
139     model.addAttribute("pere", new Pere());
140     model.addAttribute("mere", new Mere());
141     model.addAttribute("declarant", new Declarant());
142
143     Temoin temoin1 = new Temoin();
144     TemoinFormat temoin2 = new TemoinFormat();
145     model.addAttribute("temoin1", temoin1);
146     model.addAttribute("temoin2", temoin2);
147
148     model.addAttribute("naissance", naissance);
149
150     return add_edit_template;
151 }
```

Figure 28: Classe «NaissanceController » : Contrôleur

#### IV.2.2.5 Le fichier « Naissance » : Vue

Les vues de notre application sont dans le dossier *src/main/resources/templates*. Dans nos vues, nous avons utilisé du HTML avec l'espace de nom du moteur de Template Thymeleaf, du Bootstrap et un peu de JavaScript dans certaines.

Ci-dessous (Figure 30), nous vous présentons un exemple de vue « formNaissance.html » notre formulaire d'ajout d'une déclaration de naissance.

```

101@ <div id="bloc_jug" style="display:none;" >
102@ <div class="form-group col-lg-6">
103 <label for="num_jug" class="control-label col-lg-4">N° Jugement</label>
104
105@ <div class="col-lg-8" >
106 <input type="text" name="jugement.numJugement" th:value="{jugement.numJugement}" class="form-control" placeholder="Numéro jugement" />
107 <span th:errors="{jugement.numJugement}" class="alerte alert-danger"></span>
108 <span th:text="{message}" class="alerte alert-danger"></span>
109
110 </div>
111 </div>
112@ <div class="form-group col-lg-6">
113 <label for="date_jug" class="control-label col-lg-4">Date Jugement</label>
114
115@ <div class="col-lg-8">
116 <input type="date" name="jugement.dateJugement" th:value="{jugement.dateJugement}" class="form-control" placeholder="Date jugement" />
117 <span th:errors="{jugement.dateJugement}" class="alerte alert-danger"></span>
118 </div>
119 </div>
120@ <div class="form-group col-lg-6">
121 <label for="lieu_jug" class="control-label col-lg-4">Lieu Jugement</label>
122
123@ <div class="col-lg-8" >
124 <input type="text" id="lieujugement" name="jugement.lieuJugement" th:value="{jugement.lieuJugement}" class="form-control" onkeyup="this.val
125 <span th:errors="{jugement.lieuJugement}" class="alerte alert-danger"></span>
126 </div>
127 </div>
128 </div>
129@ <div class="form-group col-lg-6">
130 <label for="prenom" class="control-label col-lg-4">Prénom(s)</label>
131
132@ <div class="col-lg-8">
133 <input type="text" name="declaree.prenomDeclare" th:value="{declaree.prenomDeclare}" class="form-control" onkeyup="this.value=this.value.toUpperCase
134 <span th:errors="{declaree.prenomDeclare}" class="alerte alert-danger"></span>
135 </div>
136 </div>
137@ <div class="form-group col-lg-6">
138 <label for="nom" class="control-label col-lg-4">Nom</label>
139
140@ <div class="col-lg-8">

```

Figure 29: Fichier « Naissance » : Vue

#### IV.2.2.6 Le fichier «NaissanceMetier» : métier

La figure ci-dessous représente la classe NaissanceService permettant d'implémenter les règles de gestion de l'application. Cette couche utilise des données provenant de l'utilisateur via la couche [Web] et du SGBD via la couche [DAO] ;



```

28 @Service
29 @Transactional
30 public class NaissanceService {
31
32     @Autowired
33     private NaissanceRepository naissanceRepository;
34
35     @Autowired
36     private TemoinRepository temoinRepository;
37
38     @Autowired
39     private MentionRepository mentionRepository;
40
41     public List<Naissance> listAll() {
42         return naissanceRepository.findAll();
43     }
44
45     public void save(Naissance naissance, User user) {
46
47         naissance.setUser(user);
48         naissanceRepository.save(naissance);
49     }
50
51     @Transactional
52     public Naissance update(Naissance naissance, User user) {
53         Long naiss= naissance.getIdActe();
54         if(naiss==null) {
55             return null;
56         }
57         naissance.setUser(user);
58
59         return naissanceRepository.save(naissance);
60
61     }
62
63     private Naissance findOne(Long id) {
64         // TODO Auto-generated method stub
65         Naissance naissance = naissanceRepository.getOne(id);
66         return naissance;
67     }

```

Figure 30: La classe « NaissanceMetier » : métier

De façon générale, les logiciels et applications sont exposés à des risques d'attaques (cybercriminalité). Il faudra donc, avant, pendant et après le développement, mettre en œuvre des techniques de sécurité afin de protéger les applications des éventuelles attaques. Ainsi, dans la section qui suit, nous parlerons de la sécurisation de notre application.

### IV.3 Sécurisation de l'application

**Spring Security** est un Framework de sécurité léger qui fournit une authentification et un support d'autorisation afin de sécuriser les applications Spring. Il est livré avec des implémentations d'algorithmes de sécurité populaires.

Pour utiliser Spring Security, il faut ajouter la dépendance suivante au pom de notre application.

1. <dependency>
2. <groupId>org.springframework.boot</groupId>
3. <artifactId>spring-boot-starter-security</artifactId>
4. </dependency>

- Au démarrage de notre application nous voyons Spring Boot nous fournit donc un mot de passe par défaut grace à ajout Spring Security au classpath de notre application.
- **Spring Security** fournit une variété d'options pour effectuer l'authentification.
- L'implémentation standard et la plus courante est le **DaoAuthenticationProvider** qui récupère les détails de l'utilisateur à partir d'un simple DAO utilisateur en lecture seule
- Le **UserDetailsService**. Ce service de détails de l'utilisateur a uniquement accès au nom d'utilisateur afin de récupérer l'entité utilisateur complète et dans un grand nombre de scénarios, cela suffit.
- Mais pour personnaliser la configuration nous allons créer une classe qui hérite de **WebSecurityConfigurerAdapter**. Cette classe doit avoir les annotations **@EnableWebSecurity** et **@Configuration**. Les classes de configuration sont scannées au démarrage de l'application.

## IV.4 Présentation de quelques interfaces graphiques de l'Application

Les interfaces graphiques concernent une partie importante dans la réalisation d'une application (que ce soit web, mobile ou desktop) car permettant à l'utilisateur d'y naviguer (utiliser) facilement.

Dans cette partie, afin de bien présenter les principales interfaces graphiques, nous allons faire un petit scénario. Ce dernier consistera à simuler, de façon pratique et générale, la gestion des actes d'Etat-civil dans la commune de Ziguinchor.

### IV.4.1 Authentification

Le processus de la gestion des actes d'Etat-civil dans la commune de Ziguinchor commence par l'ouverture d'une session : Agent, Officier et Administrateur.

Mais avant cela, ce dernier doit s'identifier via l'interface suivante en saisissant son **speudo** et son **mot de passe**.



## Authentification

Authentification default

Login Votre nom utilisateur

Mot de Passe Votre mot de passe

Connexion

Figure 31:Interface d'authentification

### IV.4.2 Page d'accueil

La page d'accueil est celle vers laquelle vous êtes redirigé lorsque vous êtes identifier via l'interface login.



Déclaration des Actes <

Gestion des Actes <

Impression <

Statistiques <

Deconnexion

République du Sénégal

Un Peuple - Un But - Une Foi

ETAT-CIVIL DU SENEGAL

Figure 32:Interface page d'accueil

### IV.4.3 Déclaration de naissance

Une fois son authentification réussie, l'agent de saisie ou Officier d'Etat Civil pourra accéder à l'interface (figure 33) dédiée à la déclaration de naissance.

Commune de Ziguinchor: Gestion Etat Civil Bienvenue: basma

**Déclaration de Naissance**

**N° Registre**  **Registre**

**Type Déclaration**  **Centre**

**Prénom(s)**  **Nom**

**Date de Naissance**  **Heure de Naissance**

**Lieu de Naissance**  **Date Déclaration**

**Formation Sanitaire**  **Sexe**  M  F

**Père** **Mère** **Déclarant(e)** **L'officier** **Témoins** **Pièces fournies** **Mentions**

**Père**

**Type pièce**  **N° Identification**

Figure 33: Interface page déclaration de naissance

#### IV.4.4 Déclaration de mariage

Une fois son authentification réussie, l'agent de saisie ou Officier d'Etat Civil pourra accéder à l'interface (figure 34) dédiée à la déclaration de mariage.

Commune de Ziguinchor: Gestion Etat Civil Bienvenue: basma

**Déclaration de Mariage**

**N° Registre**  **Registre**

**Type Déclaration**  **Centre**

**Date de Mariage**  **Heure de Mariage**

**Lieu de Mariage**  **Date Déclaration**

**Epoux** **Epouse** **Le Régime** **L'officier** **Témoins** **Pièces** **Mentions**

**Epoux**

**Type pièce**  **N° Identification**

**Prénom(s)**  **Nom**

**Date de Naissance**  **Lieu de Naissance**

Figure 34: Interface page déclaration de mariage

### IV.4.5 Déclaration de décès

Une fois son authentification réussie, l'agent de saisi ou Officier d'Etat Civil pourra accéder à l'interface (figure 35) dédiée à la déclaration de décès.

Figure 35: Interface page déclaration de décès

### IV.4.3 Gestion d'un acte de naissance

Après l'ouverture de la déclaration de naissance, l'Officier d'Etat Civil aura la possibilité, via l'interface ci-dessous, de modifier l'acte ou de le supprimer et de visualiser le volet de l'acte scanné (figure 36).

Commune de Ziguinchor: Gestion Etat Civil Bienvenue:basma

Gestion d'acte de Naissance

Entrez prénom,nom ou numéro et année

Registre	Numero	Prénom	Nom	Date de naissance	Lieu de naissance	Le volet de l'acte	Actions
2020	1	Salimata	DIALLO	2021-11-29	ZIGUINCHOR	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>
2020	2	AMADOU	DIEDHIOU	2021-11-29	ZIGUINCHOR	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>
2021	2	SOULEYMANE	DIEDHIOU	2021-11-29	ZIGUINCHOR	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>
2021	3	Pauline	MDoum	2021-11-29	ZIGUINCHOR	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>
2021	5	Alassane	DIOP	2021-12-08	ZIGUINCHOR	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>

Figure 36: Interface page gestion de naissance

#### IV.4.4 Gestion d'un acte de mariage

Après l'ouverture de la déclaration de mariage, l'Officier d'Etat Civil aura la possibilité, via l'interface ci-dessous, de modifier l'acte ou de le supprimer et de visualiser le volet de l'acte scanné.

Commune de Ziguinchor: Gestion Etat Civil Bienvenue:basma

Gestion d'acte de Mariage

Entrez prénom,nom ou numéro et année

Registre	Numero	Date mariage	Prénom époux	Nom époux	Prénom épouse	Nom épouse	Le volet de l'acte	Actions
2021	1	2021-12-01	Sambajhhh	BAGyygg	Fatoumataggg	Syllagyygy	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>
2021	2	2021-12-01	Lamine	CAMARA	Fatoumata	Dabo	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>
2021	3	2021-11-30	Tony	CAMARA	Fatoumata	Dabo	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>
2021	4	2021-12-06	Tony	CAMARA	Fatoumata	Dabo	<input type="button" value="Voire volet"/>	<input type="button" value="Editer"/> <input type="button" value="Supprimer"/>

Figure 37: Interface page gestion de mariage

#### IV.4.5 Gestion d'un acte de décès

Après l'ouverture de la déclaration de décès, l'Officier d'Etat Civil aura la possibilité, via l'interface ci-dessous, de modifier l'acte ou de le supprimer et de visualiser le volet de l'acte scanné.

Registre	Numero	Prénom	Nom	Date de naissance	Lieu de naissance	Le volet de l'acte	Actions
2021	1	AMADOU	BADJI	2021-03-12	ZIGUINCHOR	Voire volet	Editer Supprimer
2020	1	JEAN	GOMIS	2021-12-14	ZIGUINCHOR	Voire volet	Editer Supprimer
2020	2	Salif	Djiba	2021-12-07	ZIGUINCHOR	Voire volet	Editer Supprimer
2021	2	Fatou	Mendy	2021-11-29	ZIGUINCHOR	Voire volet	Editer Supprimer
2022	3	ABDOU	DIEME	2022-03-12	ZIGUINCHOR	Voire volet	Editer Supprimer

Figure 38: Interface page gestion de décès

#### IV.4.4 Rechercher un acte de naissance

Une fois l'acte trouvé, les informations du déclaré s'affiche et alors on pourra choisir le format à imprimer, volet, extrait de naissance, bulletin ou copie littérale (figure 39 puis figure 40).

# Dématérialisation des procédures et documents administratifs de l'état-civil de Ziguinchor

Commune de Ziguinchor: Gestion Etat Civil Bienvenue: basma

**Recherche Acte de Naissance**

Numéro  Registre  Centre

Entrez Année et ou centre et ou prénom et ou nom de la personne déclarée

Registre	Numero	Prénom	Nom	Date de naissance	Lieu de naissance	Prénom Père	Prénom et Nom Mère
2020	1	Salimata	DIALLO	2021-11-29	ZIGUINCHOR	Lamine	Aissatou BALDE
2020	2	AMADOU	DIEDHIOU	2021-11-29	ZIGUINCHOR	FODÉ	FATOU SANÉ
2021	2	SOULEYMANE	DIEDHIOU	2021-11-29	ZIGUINCHOR	FODÉ	FATOU BADJI
2021	3	Pauline	MDoum	2021-11-29	ZIGUINCHOR	Lamine	Marie Cisse
2021	5	Alassane	DIOP	2021-12-08	ZIGUINCHOR		

0

localhost:8080/acte\_naissance/recherchNaissance

Figure 39: Interface page recherche acte naissance

Commune de Ziguinchor: Gestion Etat Civil Bienvenue: basma

**Impression d'acte de Naissance**

N° Acte	Année	Date déclaration	Type déclaration	Lieu	N° jugement	Date jugement	Le volet de l'acte
2021	3	2021-12-02 00:00:00.0	RECONNAISSANCE POSTERIEURE	ZIGUINCHOR			

**Enfant**

**Prénom(s):** Pauline      **Nom:** MDoum  
**Date de Naissance:** 2021-11-29      **Lieu de Naissance:** ZIGUINCHOR  
**Heure de Naissance:** 20:11      **Sexe:**  
**Formation sanitaire:** HOPITAL REGINAL DE ZIGUINCHOR

Figure 40: Interface page affichage d'un acte de naissance trouvé



### ✓ Le volet de l'acte de naissance

Après avoir choisi le type de copie, l'Officier ou l'Agent d'état-civil pourra le visualiser via un navigateur et l'imprimer ensuite (figure 41).

sur 1      - | +      100 %      ▾

REGION DE ZIGUINCHOR	REPUBLIQUE DU SENEGAL	VOLLET 1
COMMUNE DE ZIGUINCHOR	2021	<input type="checkbox"/> R <input type="checkbox"/> DT
CENTRE PRINCIPAL DE ZIGUINCHOR		<input type="checkbox"/> A
		<input type="checkbox"/> C/CR
ACTE DE NAISSANCE		
[ ][ ][ ][ ]		
Numéro 3		
RENSEIGNEMENTS SUR L'ENFANT		
Prénoms	Pauline Nom MDoum	
Sexe	null	<input type="checkbox"/> SEXE
Date de naissance	vingt-neuf novembre deux mille vingt et un	<input type="checkbox"/> JJ <input type="checkbox"/> MM <input type="checkbox"/> ANNEE
Heure	20:11	
Lieu de naissance	ZIGUINCHOR	
Formation sanitaire	HOPITAL REGINAL DE ZIGUINCHOR	<input type="checkbox"/> FS
RENSEIGNEMENTS SUR LE PERE		
Prénoms	Lamine Nom MDoum	
Date de naissance	2021-08-30	<input type="checkbox"/> JJ <input type="checkbox"/> MM <input type="checkbox"/> ANNEE
Lieu de naissance	Ziguinchor	
Profession	Ingenieur télécom	<input type="checkbox"/> PP
Domicile	Alwar1	<input type="checkbox"/> DP
RENSEIGNEMENTS SUR LA MERE		
Prénoms	Marie Nom MDoum	
Date de naissance	2021-12-07	<input type="checkbox"/> JJ <input type="checkbox"/> MM <input type="checkbox"/> ANNEE
Lieu de naissance	Bignona	
Profession	Commerçante	<input type="checkbox"/> PM
Domicile	Alwar	<input type="checkbox"/> DM
SUR LA DECLARATION DE		

Figure 41: Exemple d'un volet de naissance choisi

✓ **Extrait de naissance**

Après avoir choisi le type de copie, l'Officier ou l'Agent d'état-civil pourra le visualiser via un navigateur et imprimer l'extrait de naissance ensuite (figure 42).

- + 70 % ▾

REGION DE ZIGUINCHOR COMMUNE DE ZIGUINCHOR  CENTRE PRINCIPAL DE ZIGUINCHOR  Registre N° 3 / 2021	REPUBLIQUE DU SENEGAL Un Peuple - Un But - Une Foi  <hr style="width: 20%; margin: auto;"/> ETAT-CIVIL
<b>EXTRAIT DU REGISTRE DES ACTES DE NAISSANCE</b>  Pour l'année deux mille vingt et un  N° dans le Registre trois <small>(En lettres)</small>	An 2021  3 <small>(N° dans le Registre en chiffres)</small>
Le vingt-neuf novembre deux mille vingt et un à vingt heure(s) onze minute(s) est né(e) à ZIGUINCHOR  Un enfant de sexe null Pauline <span style="float: right;">MDoum</span> <small>Prénoms <span style="float: right;">Nom de famille</span></small> De Lamine <small>Prénoms du Père</small> Marie <span style="float: right;">Cisse</span> <small>Prénoms de la mère <span style="float: right;">Nom de famille de la mère</span></small>	
Jugement d'autorisation d'inscription ex supplétif	Délivré par le juge Le _____ Sous le numéro _____ Inscrit le _____ Sur le registre des actes de naissance
Extrait délivré par CENTRE PRINCIPAL DE ZIGUINCHOR	POUR EXTRAIT CERTIFIE CONFORME A ZIGUINCHOR Le 21/12/2021 L'officier d'état-civil soussigné PRENOM ET NOM :
(1)(2)(3) Notes et mentions marginales au verso	
RESERVE    □ □    □ □ □    □ □    □ □    □ □ □ □ □ □    □ □ □    □ □ □ □	
DEP            CL    CEC    DN                            S            PN            JDP	

Figure 42: Exemple d'un extrait de naissance choisi

## ✓ Certificat de mariage

Après avoir choisi le type de copie, l'Officier ou l'Agent d'état-civil pourra le visualiser via un navigateur et imprimer le certificat de mariage (figure 43).

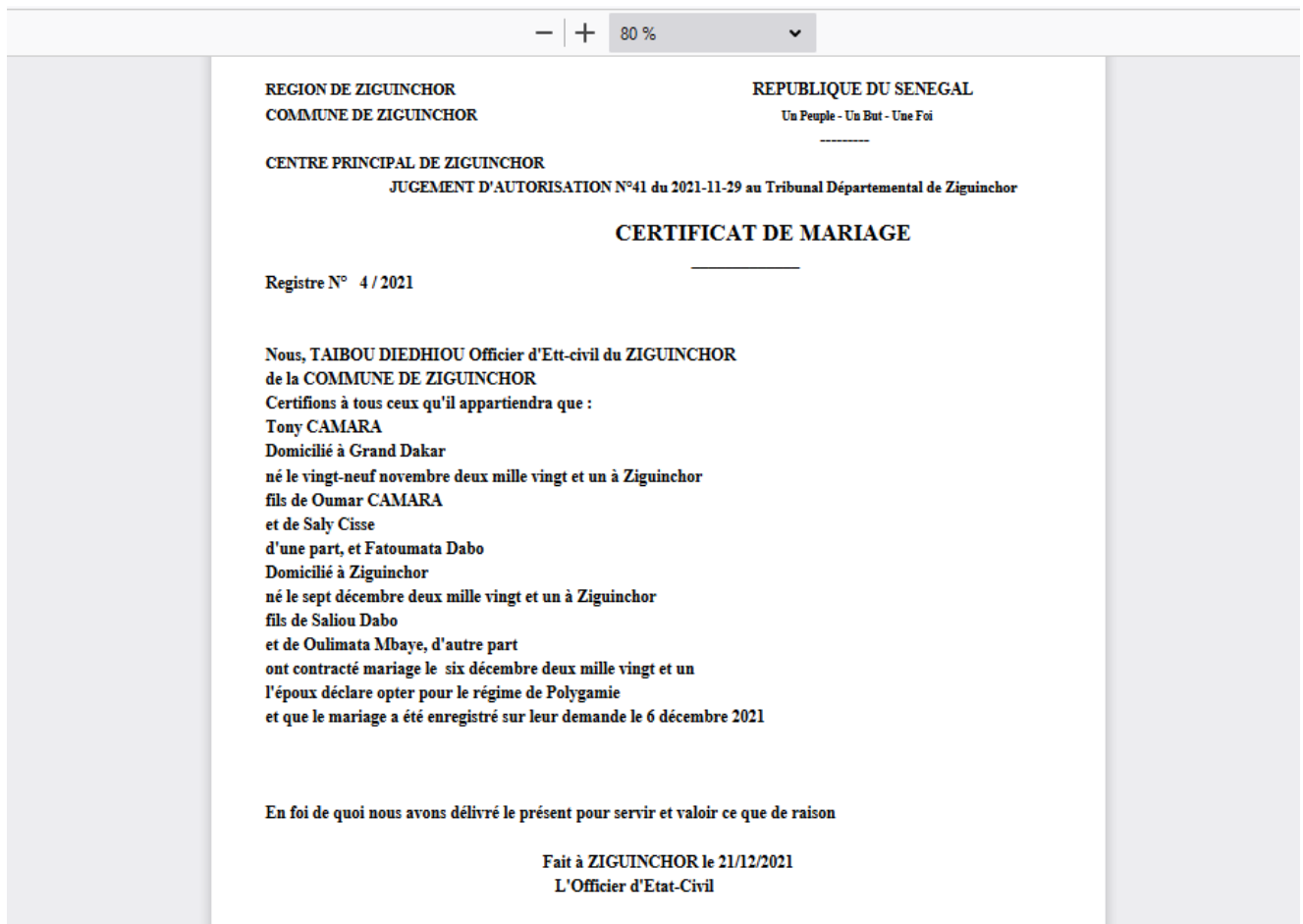


Figure 43: Exemple d'un certificat de naissance choisi

### IV.3.5 Gestion statistiques des actes de naissances

La figure suivante permet de faire la statistique des actes de naissance. Ainsi, connaitre le nombre de déclaration faite dans le mois ou durant une année.

Mois	Décl. nor(H)	Décl. nor(F)	Décl. tar(H)	Décl. tar(F)	Transcr.(H)	Transcr.(F)	Jug.(H)	Jug.(F)	TOTAUX
Janvier	0	0	0	0	0	0	0	0	0
Février	0	0	0	0	0	0	0	0	0
Mars	0	0	0	0	0	0	0	0	0
Avril	0	0	0	0	0	0	0	0	0
Mai	0	0	0	0	0	0	0	0	0
Juin	0	0	0	0	0	0	0	0	0
Juillet	0	0	0	0	0	0	0	0	0
Août	0	0	0	0	0	0	0	0	0
Septembre	0	0	0	0	0	0	0	0	0
Octobre	0	0	0	0	0	0	0	0	0

Figure 44: Interface d'un tableau statistique de naissance

### IV.3.6 Gestion administrative

C'est la partie Back end de notre application qui nous permet non seulement de faire la gestion des années de registres, des codes centres, des formations sanitaires, des officiers d'état-civil et des utilisateurs du système. Mais aussi, il faut noter le tableau de bord qui nous permet de connaitre le nombre actes de naissance, mariages et décès saisis dans la base de données et nombre d'utilisateurs connectés (figure 45).

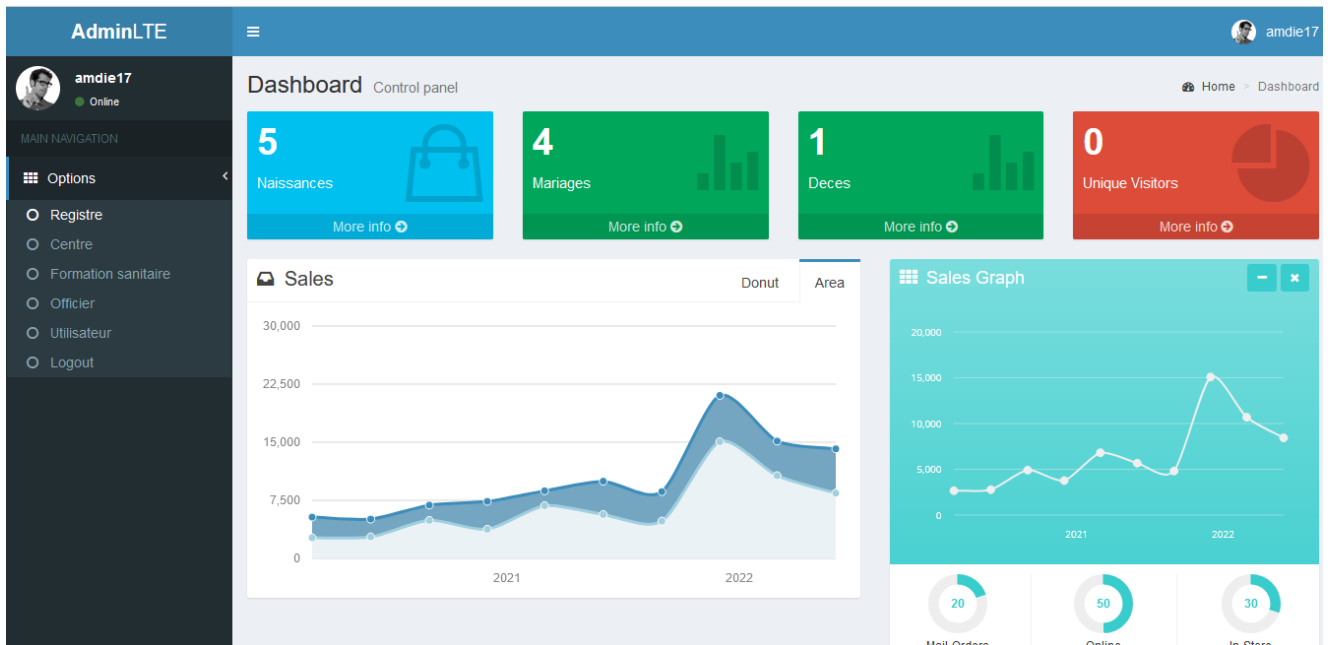


Figure 45: Interface administration

Dans ce dernier chapitre, nous avons enfin montré le résultat tant attendu, l'application en tant que telle. En effet, les technologies et outils utilisés pour implémenter l'application ont été montrés ainsi que quelques interfaces graphiques de l'application.

## CONCLUSION GENERALE ET PERSPECTIVES

Ce travail avait comme objectif de concevoir et d'implémenter un logiciel pour la gestion des actes d'E.C à la mairie de Ziguinchor.

Pour cela, l'objectif est loin d'être atteint, malgré un travail important qui a été fait, des modules restent à être implémentés. Quant à la partie développée, elle répond parfaitement aux attentes décrites dans le cahier des charges. En effet, ce logiciel permet, entre autres, à chaque O.E.C, de pouvoir faire son travail sans trop de contraintes, de faire ou de faire faire les déclarations de naissances, de mariages et de décès, de valider les actes en règles, de délibérer, de transcrire et de publier des actes d'Etat-Civil à la population de la commune.

Afin de bien présenter les objectifs et les résultats de notre travail, ce mémoire a été scindé en cinq principaux chapitres. Le premier chapitre a fait l'objet de la présentation du contexte justificatif du sujet ainsi que les solutions aux problèmes rencontrés lors de la conception de la nouvelle application. Le deuxième chapitre a parlé du processus de développement dans lequel nous avons présenté et adapté la méthodologie 2TUP à notre projet. Le troisième chapitre s'est intéressé à la spécification et l'analyse des besoins fonctionnels du système. Le quatrième chapitre a abordé la conception de l'application. Enfin, le cinquième chapitre a présenté le résultat, à savoir les interfaces de l'application créée.

Cependant, l'application que nous avons présentée dans ce document présente quelques limites, à savoir : La partie validation de l'Officier des actes saisies est en cours de traitement. Il y a aussi le menu « Divers » qui doit traiter les Certificat de non inscription(mariage), Certificat de non inscription(naissance), Certificat de célibat, Certificat de vie individuel, Certificat de d'inhumation etc. est en cours de traitement.

Dans les perspectives, nous envisageons d'améliorer l'application de par les fonctionnalités suivantes :

✓ **La création de compte personnel en ligne:** Il s'agit de permettre les gens d'accéder à l'application mise en ligne via l'internet et d'en créer leur compte personnel. Ainsi, ils pourront faire leur demande pour l'obtention de leurs documents d'acte d'état-civil.

- ✓ **La gestion du paiement en ligne:** Après la création de compte personnel, faire la demande des actes d'état-civil, si la demande est autorisée. Il sera demandé de passer au paiement des frais de services et des timbres inclus.
  
- ✓ **La signature électronique de l'officier d'état-civil:** les documents qui seront imprimés après validation du paiement, auront la signature de l'un des officiers qui était en ligne ou de permettre à l'application de produire de façon automatisée une signature électronique de l'un des Officier d'état-civil.
  
- ✓ **L'interconnexion entre le centre principal et le centre secondaire ou L'interconnexion entre les communes:** Ce sera la première interconnexion entre les centres. Ainsi les deux centres pourront travail de la façon suivante : l'un des centres fait des demandes en ligne à l'autre centre et attend la validation d'autorisation d'accéder aux actes demandés par l'Officier d'état-civil. Ensuite, une seconde demande pour la signature des documents par les ou l'Officier qui est en ligne et fini par imprimé les actes d'état-civil.

## WEBOGRAPHIE

- [1] « CODE\_FAMILLE.pdf ». Consulté le: déc. 30, 2021. [En ligne]. Disponible sur: [https://armedeterre.sec.gouv.sn/sites/default/files/CODE\\_FAMILLE.pdf](https://armedeterre.sec.gouv.sn/sites/default/files/CODE_FAMILLE.pdf).
- [2] « Sénégal Services ». <https://senegalservices.sn/> (consulté le déc. 28, 2021).
- [3] <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/>
- [4] [http://fmi.univ-tiaret.dz/images/3lmd/2020-2021/Cour\\_introduction\\_aux\\_mthodes\\_de\\_dveppement.pdf](http://fmi.univ-tiaret.dz/images/3lmd/2020-2021/Cour_introduction_aux_mthodes_de_dveppement.pdf)
- [5] « Qu'est-ce que le langage UML (langage de modélisation unifié)? », *Lucidchart*. <https://www.lucidchart.com/pages/fr/langage-uml> (consulté le déc. 30, 2021).
- [6] « Spring Boot ». <https://spring.io/projects/spring-boot> (consulté le déc. 30, 2021).
- [7] « Introduction à Thymeleaf — Spring Framework / Spring Boot ». <https://gayerie.dev/docs/spring/spring/thymeleaf.html> (consulté le déc. 30, 2021).
- [8] Futura, « Télécharger gratuitement SAP PowerDesigner (PowerAMC) sur Futura », *Futura*. <https://www.futura-sciences.com/tech/telecharger/sap-powerdesigner-poweramc-305> (consulté le déc. 30, 2021).
- [9] « MySQL ». <https://www.mysql.com/fr/> (consulté le déc. 30, 2021).
- [10] « Java | Oracle ». <https://www.java.com/fr/> (consulté le déc. 30, 2021).
- [11] J. F. - js.foundation, « jQuery ». <https://jquery.com/> (consulté le déc. 30, 2021).
- [12] « Cours et Tutoriels sur le Langage SQL ». <https://sql.sh/> (consulté le déc. 30, 2021).

- [13] E. Foundation, « Eclipse Downloads | The Eclipse Foundation ». <https://www.eclipse.org/downloads/> (consulté le déc. 30, 2021).
- [14] « Bootstrap · The most popular HTML, CSS, and JS library in the world. » <https://getbootstrap.com/> (consulté le déc. 30, 2021).
- [15] « Apache Tomcat® - Welcome! » <https://tomcat.apache.org/> (consulté le déc. 30, 2021).
- [16] « CODE\_FAMILLE.pdf ». Consulté le: déc. 30, 2021. [En ligne]. Disponible sur: [https://armedeterre.sec.gouv.sn/sites/default/files/CODE\\_FAMILLE.pdf](https://armedeterre.sec.gouv.sn/sites/default/files/CODE_FAMILLE.pdf)
- [17] « guide\_de\_1\_usager\_v21\_9\_2014.pdf ». Consulté le: déc. 30, 2021. [En ligne]. Disponible sur: [https://www.platforme-ane.sn/IMG/pdf/guide\\_de\\_1\\_usager\\_v21\\_9\\_2014.pdf](https://www.platforme-ane.sn/IMG/pdf/guide_de_1_usager_v21_9_2014.pdf)
- [18] « RAPPORT ETAT CIVIL DEC 2015-BECPD.pdf ». Consulté le: déc. 30, 2021. [En ligne]. Disponible sur: <https://www.ansd.sn/ressources/publications/RAPPORT%20ETAT%20CIVIL%20DEC%202015-BECPD.pdf>
- [19] : Steinmann, Alain et al. “Qu’est-ce que Java ?”. CCM Benchmark Group. Consulté le 27 octobre 2021 sur <https://journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203555-java-definition/>.
- [20] : Steinmann, Alain et al. “Qu’est-ce que Java ?”. CCM Benchmark Group. Consulté le 17 octobre 2021 sur <https://journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/>.
- [21] : Fernandez, Daniel et al. “Thymeleaf”. Consulté le 28 octobre 2021 sur <https://www.thymeleaf.org/index.html>.

## ANNEXE

### Annexe 1 : Diagramme de classes



## Dématérialisation des procédures et documents administratifs de l'état-civil de Ziguinchor

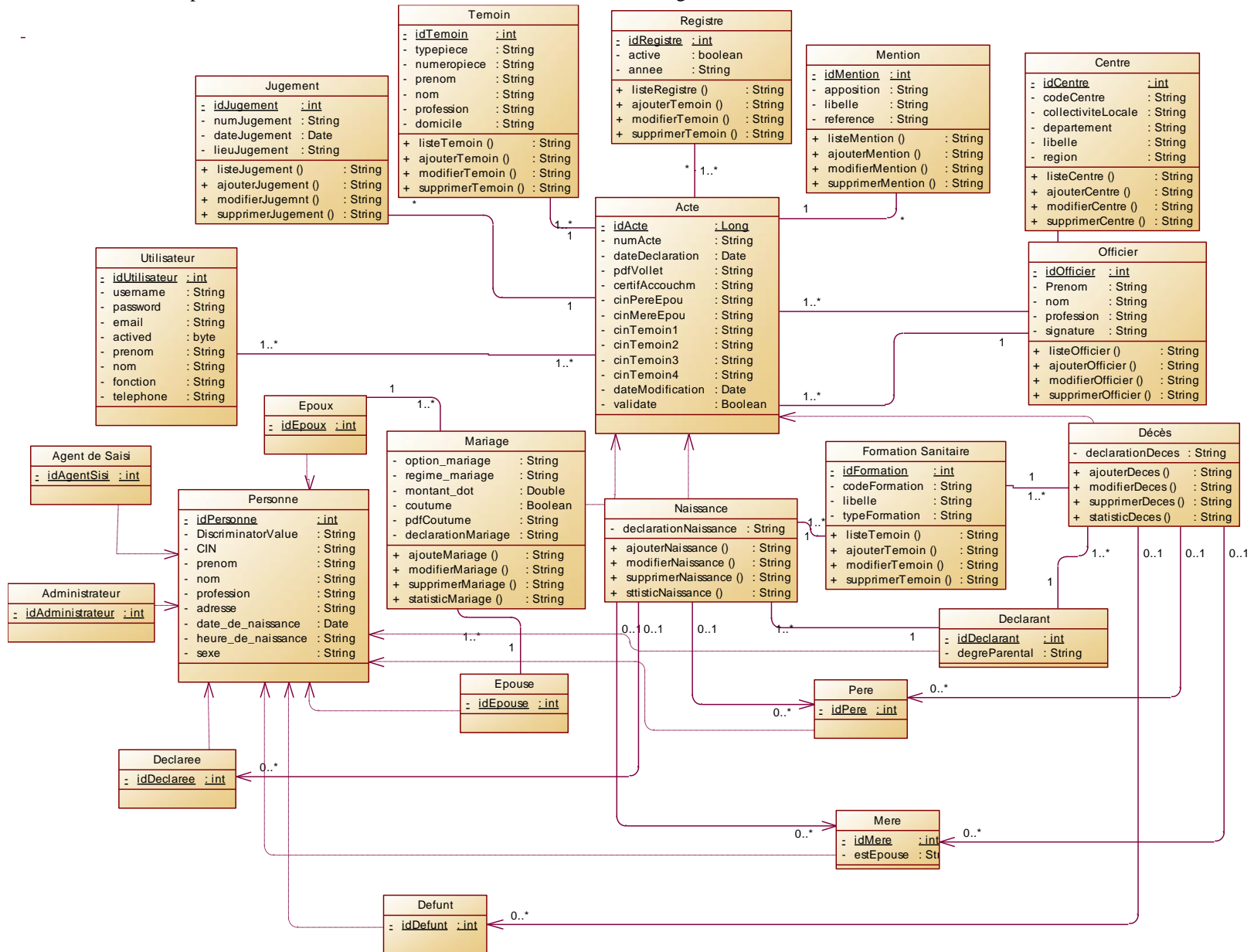


Figure 46: Diagramme de classes aux fonctionnalités de la gestion de l'état-civil

**Annexe 2:** Dictionnaire de données

<b>Tables</b>	<b>Champs</b>	<b>Libellé</b>	<b>Types</b>
<b>Utilisateurs</b>	<b>login</b>	<b>Login (identifiant) de l'utilisateur (Agent)</b>	<b>String</b>
	<b>motDePasse</b>	<b>Mot de passe de l'utilisateur (Agent)</b>	<b>String</b>
	<b>prenom</b>	<b>Le prénom de l'agent municipal</b>	<b>String</b>
	<b>nom</b>	<b>Le nom de l'agent municipal</b>	<b>String</b>
	<b>fonction</b>	<b>La fonction de l'agent municipal</b>	<b>String</b>
	<b>statut</b>	<b>Activé ou désactivé</b>	<b>Boolean</b>
	<b>email</b>	<b>L'email de l'agent municipal</b>	<b>String</b>
	<b>telephone</b>	<b>Le téléphone de l'agent municipal</b>	<b>String</b>
<b>Rôle</b>	<b>name</b>	<b>Le rôle de l'agent municipal</b>	<b>String</b>
<b>Naissance</b>	<b>idActe</b>	<b>Le numéro d'identifiant d'un acte de naissance</b>	<b>Long</b>
	<b>numActe</b>	<b>Le numéro de l'acte de naissance</b>	<b>String</b>
	<b>année</b>	<b>L'année de registre d'un acte de naissance</b>	<b>String</b>
	<b>declarationN</b>	<b>Déclaration d'un acte de naissance</b>	<b>Sting</b>
	<b>numJugement</b>	<b>Le numéro du jugement d'un acte de naissance</b>	<b>String</b>
	<b>dateJugement</b>	<b>La date du jugement d'un acte de naissance</b>	<b>Date</b>
	<b>lieuJugement</b>	<b>Le lieu du jugement d'un acte de naissance</b>	<b>String</b>

<b>prenomOff</b>	Le prénom d'un officier d'état-civil	String
<b>nomOff</b>	Le prénom d'un officier d'état-civil	String
<b>codeCentre</b>	Le numéro du code de centre	String
<b>nomFormaSanit</b>	Le nom de la formation sanitaire	String
<b>prenomDeclaree</b>	Le prénom de l'enfant ou personne déclarée	String
<b>nomDeclaree</b>	Le nom de l'enfant ou personne déclarée	String
<b>dateNaissanceDeclaree</b>	La date de naissance de l'enfant ou personne déclarée	Date
<b>lieuNaissanceDeclare</b>	Le lieu de naissance de l'enfant ou personne déclarée	String
<b>sexeDeclaree</b>	Le sexe de naissance de l'enfant ou personne déclarée	String
<b>dateDeclaration</b>	La date de déclaration	Date
<b>numPiecePere</b>	Le numéro de pièce du père	int
<b>prenomPere</b>	Le prénom du père	String
<b>nomPere</b>	Le nom du père	String
<b>dateNaissancePere</b>	La date de naissance du père	Date
<b>lieuNaissancePere</b>	Le lieu de naissance du père	String
<b>professionPere</b>	La profession du père	String
<b>domicilePere</b>	Le domicile du père	String
<b>numPieceMere</b>	Le numéro de pièce de la mère	int
<b>prenomMere</b>	Le prénom de la mère	String
<b>nomMere</b>	Le nom de la mère	String

	<b>dateNaissanceMere</b>	<b>La date de naissance de la mère</b>	<b>Date</b>
	<b>lieuNaissanceMere</b>	<b>Le lieu de naissance de la mère</b>	<b>String</b>
	<b>professionMere</b>	<b>La profession de la mère</b>	<b>String</b>
	<b>domicileMere</b>	<b>Le domicile de la mère</b>	<b>String</b>
	<b>numPieceTemoin</b>	<b>Le numéro de pièce du témoin</b>	<b>int</b>
	<b>prenomTemoin</b>	<b>Le prénom du témoin</b>	<b>String</b>
	<b>nomTemoin</b>	<b>Le nom du témoin</b>	<b>String</b>
	<b>dateNaissanceTemoin</b>	<b>La date de naissance du témoin</b>	<b>Date</b>
	<b>lieuNaissanceTemoin</b>	<b>Le lieu de naissance du témoin</b>	<b>String</b>
	<b>professionTemoin</b>	<b>La profession du témoin</b>	<b>String</b>
	<b>domicileTemoin</b>	<b>Le domicile du témoin</b>	<b>String</b>
	<b>apposition</b>	<b>La personne qui a mis la mention marginale</b>	<b>String</b>
	<b>libelle</b>	<b>La mention marginale</b>	<b>String</b>
	<b>reference</b>	<b>La référence (code de la famille)</b>	<b>String</b>
	<b>validate</b>	<b>La validation de l'acte</b>	<b>Boolean</b>
	<b>scanActe</b>	<b>Le fichier de l'acte de naissance</b>	<b>String</b>
	<b>scanAccouchement</b>	<b>Le certificat d'accouchement</b>	<b>String</b>
	<b>dateModification</b>	<b>Les dates de modification de l'acte</b>	<b>Date</b>
<b>Mariage</b>	<b>idActe</b>	<b>Le numéro d'identifiant d'un acte de mariage</b>	<b>Long</b>
	<b>numActe</b>	<b>Le numéro de l'acte de mariage</b>	<b>String</b>
	<b>année</b>	<b>L'année de registre d'un acte de mariage</b>	<b>String</b>

<b>declarationM</b>	<b>Déclaration d'un acte de mariage</b>	<b>String</b>
<b>numJugement</b>	<b>Le numéro du jugement d'un acte de mariage</b>	<b>String</b>
<b>dateJugement</b>	<b>La date du jugement d'un acte de mariage</b>	<b>Date</b>
<b>lieuJugement</b>	<b>Le lieu du jugement d'un acte de mariage</b>	<b>String</b>
<b>prenomOff</b>	<b>Le prénom d'un officier d'état-civil</b>	<b>String</b>
<b>nomOff</b>	<b>Le prénom d'un officier d'état-civil</b>	<b>String</b>
<b>dateMariage</b>	<b>La date du mariage</b>	<b>Date</b>
<b>lieuMariage</b>	<b>Le lieu du mariage</b>	<b>String</b>
<b>heureMariage</b>	<b>L'heure du mariage</b>	<b>String</b>
<b>numPieceEpoux</b>	<b>Le numéro de pièce de l'époux</b>	<b>int</b>
<b>prenomEpoux</b>	<b>Le prénom de l'époux</b>	<b>String</b>
<b>nomEpoux</b>	<b>Le nom de l'époux</b>	<b>String</b>
<b>dateNaissanceEpoux</b>	<b>La date de naissance de l'époux</b>	<b>Date</b>
<b>lieuNaissanceEpoux</b>	<b>Le lieu de naissance de l'époux</b>	<b>String</b>
<b>sexeEpoux</b>	<b>Le sexe de l'époux</b>	<b>String</b>
<b>nationaliteEpoux</b>	<b>La nationalité de l'époux</b>	<b>String</b>
<b>numPiecePereEpoux</b>	<b>Le numéro de pièce du père</b>	<b>int</b>
<b>prenomPereEpoux</b>	<b>Le prénom du père</b>	<b>String</b>
<b>nomPereEpoux</b>	<b>Le nom du père</b>	<b>String</b>
<b>dateNaissancePereEpoux</b>	<b>La date de naissance du père</b>	<b>Date</b>
<b>lieuNaissancePereEpoux</b>	<b>Le lieu de naissance du père</b>	<b>String</b>
<b>professionPereEpoux</b>	<b>La profession du père</b>	<b>String</b>

<b>domicilePereEpoux</b>	<b>Le domicile du père</b>	<b>String</b>
<b>numPieceMereEpoux</b>	<b>Le numéro de pièce de la mère</b>	<b>int</b>
<b>prenomMereEpoux</b>	<b>Le prénom de la mère</b>	<b>String</b>
<b>nomMereEpoux</b>	<b>Le nom de la mère</b>	<b>String</b>
<b>dateNaissanceMereEpoux</b>	<b>La date de naissance de la</b>	<b>Date</b>
<b>lieuNaissanceMereEpoux</b>	<b>Le lieu de naissance de la mère</b>	<b>String</b>
<b>professionMereEpoux</b>	<b>La profession de la mère</b>	<b>String</b>
<b>domicileMereEpoux</b>	<b>Le domicile de la mère</b>	<b>String</b>
<b>numPieceEpouse</b>	<b>Le numéro de pièce de l'époux</b>	<b>int</b>
<b>prenomEpouse</b>	<b>Le prénom de l'époux</b>	<b>String</b>
<b>nomEpouse</b>	<b>Le nom de l'époux</b>	<b>String</b>
<b>dateNaissanceEpouse</b>	<b>La date de naissance de</b>	<b>Date</b>
<b>lieuNaissanceEpouse</b>	<b>Le lieu de naissance de</b>	<b>String</b>
<b>sexeEpouse</b>	<b>Le sexe de l'époux</b>	<b>String</b>
<b>nationaliteEpouse</b>	<b>La nationalité de l'époux</b>	<b>String</b>
<b>numPiecePereEpouse</b>	<b>Le numéro de pièce du père</b>	<b>int</b>
<b>prenomPereEpouse</b>	<b>Le prénom du père</b>	<b>String</b>
<b>nomPereEpouse</b>	<b>Le nom du père</b>	<b>String</b>
<b>dateNaissancePereEpouse</b>	<b>La date de naissance du père</b>	<b>Date</b>
<b>lieuNaissancePereEpouse</b>	<b>Le lieu de naissance du père</b>	<b>String</b>
<b>professionPereEpouse</b>	<b>La profession du père</b>	<b>String</b>
<b>domicilePereEpouse</b>	<b>Le domicile du père</b>	<b>String</b>
<b>numPieceMereEpouse</b>	<b>Le numéro de pièce de la mère</b>	<b>int</b>
<b>prenomMereEpouse</b>	<b>Le prénom de la mère</b>	<b>String</b>
<b>nomMereEpouse</b>	<b>Le nom de la mère</b>	<b>String</b>
<b>dateNaissanceMereEpouse</b>	<b>La date de naissance de la</b>	<b>Date</b>
<b>e</b>	<b>mère</b>	

<b>lieuNaissanceMereEpouse</b>	<b>Le lieu de naissance de la mère</b>	<b>String</b>
<b>professionMereEpouse</b>	<b>La profession de la mère de</b>	<b>String</b>
<b>domicileMereEpouse</b>	<b>Le domicile de la mère de l'épouse</b>	<b>String</b>
<b>Option_mariage</b>	<b>Choisir si vous optez : monogamie ou polygamie</b>	<b>String</b>
<b>Regime_mariage</b>	<b>Choisir si vous optez : séparation de biens ou communauté de biens</b>	<b>String</b>
<b>Montant_dot</b>	<b>Le montant de la dot</b>	<b>Double</b>
<b>numPieceTemoin</b>	<b>Le numéro de pièce du témoin</b>	<b>int</b>
<b>prenomTemoin</b>	<b>Le prénom du témoin</b>	<b>String</b>
<b>nomTemoin</b>	<b>Le nom du témoin</b>	<b>String</b>
<b>dateNaissanceTemoin</b>	<b>La date de naissance du témoin</b>	<b>Date</b>
<b>lieuNaissanceTemoin</b>	<b>Le lieu de naissance du témoin</b>	<b>String</b>
<b>professionTemoin</b>	<b>La profession du témoin</b>	<b>String</b>
<b>domicileTemoin</b>	<b>Le domicile du témoin</b>	<b>String</b>
<b>apposition</b>	<b>La personne qui a mis la mention marginale</b>	<b>String</b>
<b>libelle</b>	<b>La mention marginale</b>	<b>String</b>
<b>reference</b>	<b>La référence (code de la famille)</b>	<b>String</b>
<b>validate</b>	<b>La validation de l'acte</b>	<b>Boolean</b>
<b>scanActe</b>	<b>Le fichier de l'acte de mariage</b>	<b>String</b>
<b>scanCINEpoux</b>	<b>La C.I.N de l'époux</b>	<b>String</b>
<b>scanCINEpouse</b>	<b>La C.I.N de l'épouse</b>	<b>String</b>

<b>dateModification</b>	<b>Les dates de modification de l'acte de mariage</b>	<b>Date</b>
-------------------------	---	-------------