



RÉPUBLIQUE DU BÉNIN  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE  
RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88  
<http://www.ifri.-uac.net> Courriel : [contact@ifri.uac.bj](mailto:contact@ifri.uac.bj)



# MÉMOIRE

pour l'obtention du

**Diplôme de Licence en Informatique**

**Option : Génie Logiciel**

**Présenté par :**

Faouziya okpè BIAO A.

## **Application de gestion des archives pour le Ministère de l'Agriculture de l'Elevage et de la Pêche(MAEP)**

**Sous la supervision :**

M. Emery ASSOGBA (Ingénieur)

M. Abdou-Aziz SOBABE (Directeur de la DIP du MAEP)

Année Académique : 2016-2017

# Dédicaces

A

Mon père **BIAO AGBEDE** Salifou

Ma mère **MOUSSA** Adjaratou

Mes frères et sœurs

# Remerciements

Loin d'être une simple formalité académique, j'ai l'obligation intellectuelle de m'acquitter d'un devoir de cœur en exprimant mes reconnaissances à tous ceux qui de près ou de loin, de manière directe ou indirecte ont contribué à la réalisation de ce travail. Il a été réalisé grâce aux multiples conseils et collaborations de certaines personnes à qui je tiens à adresser mes sincères remerciements.

- Tout le personnel de l'administration de l'IFRI.
- Mon encadrant M. ASSOGBA Emery, Enseignant à l'Institut de Formation et de Recherche en Informatique. Malgré vos multitudes occupations, vous avez accepté diriger mon travail. Rassurez-vous de mon profond respect.
- A mon maître de stage M. Abdou-Aziz SOBABE, Directeur de l'Informatique et du Pré-Archivage (DIP) du Ministère de l'Agriculture de l'Elevage et de la Pêche.
- Messieurs les membres de jury pour l'honneur que vous me faites en acceptant de juger ce travail. Veuillez trouver ici l'expression émue de toute ma reconnaissance.
- Mes ami(e)s pour les bons moments passés ensemble et pour leurs soutiens au quotidien.
- Toute la famille BIAO.

# Table des figures

|     |  |    |
|-----|--|----|
| 3.1 | Diagramme de cas d'utilisation définissant les fonctionnalités de l'internaute . . . | 12 |
| 3.2 | Diagramme de cas d'utilisation définissant les fonctionnalités de l'archiviste . . . | 13 |
| 3.3 | Diagramme de cas d'utilisation définissant les fonctionnalités de l'administrateur   | 14 |
| 3.4 | Diagramme de cas d'utilisation définissant les fonctionnalités des trois acteurs .   | 15 |
| 3.5 | Diagramme de classes . . . . .   | 20 |
| 3.6 | Diagramme de séquence de la fonctionnalité Ouvrir document . . . . .                 | 22 |
| 3.7 | Diagramme de séquence de la fonctionnalité Imprimer document . . . . .               | 23 |
| 3.8 | Les différentes étapes du scrum. . . . .   | 27 |
| 4.1 | Interface d'accueil . . . . .  | 30 |
| 4.2 | Interface d'enregistrement des documents . . . . .                                   | 31 |
| A.1 | Diagramme montrant les activités à réaliser jusqu'à l'impression d'un document       | 38 |
| A.2 | Diagramme de composants . . . . .  | 39 |

# Glossaire

- **Archivage** : action de recueillir, de classer et de conserver les documents.
- **CSS** : Cascading Style Sheets : (feuille de style en cascade). C'est un langage de style qui définit la présentation des documents html. Il couvre les polices, les couleurs, les marges, les lignes, la hauteur, la largeur, les images d'arrière plan et bien d'autre chose.
- **Gestion** : Action ou manière de gérer, d'administrer, de diriger, d'organiser quelque chose.
- **Exception** : Ce qui est hors de la règle commune, qui paraît unique.
- **HTML** : HyperText Markup Language. C'est un format permettant de définir les éléments variés d'une page web qu'on observe à l'aide d'un navigateur.
- **IHM** : Interface Homme-Machine. C'est l'ensemble des moyens utilisés par l'homme pour communiquer avec une machine.
- **MVC** : Model Vue Contrôleurs. C'est un motif d'architecture logicielle destiné aux interfaces graphiques.
- **MySQL** : My Structured Query Language. C'est un système de gestion de bases de données permettant de stocker des données et pouvoir les utiliser.
- **Optimiser** : Fait de donner un rendement optimal en créant les conditions les plus favorables ou en tirant le meilleur parti possible.
- **PHP** : Hypertext Preprocessor. C'est un langage de script conçu pour le développement d'application web.
- **Scénario** : Déroulement programmé ou prévu d'une action.
- **UML** : Unified Modeling Language (langage de modélisation unifié). C'est un langage de modélisation graphique à base des schémas appelés diagrammes.

# Sigles et Abréviations

- **CSS** : Cascading Style Sheets.
- **DBA** : DataBase Administrator.
- **DIP** : Direction de l'Informatique et du Pré-archivage.
- **HP** : Hewlett-Packard.
- **HTML** : HyperText Markup Language.
- **IHM** : Interface Homme-Machine.
- **MVC** : Model Vue Contrôleurs.
- **MySql** : My Structured Query Language.
- **PHP** : Hypertext Preprocessor.
- **UML** : Unified Modeling Language.
- **Se** : Secrétariat.
- **SInf** : Service Informatique.
- **SPGS** : Service de Pré-archivage et de la Gestion des Savoirs.
- **SRU** : Service des Relations avec les Usagers.

# Résumé

La bonne gestion de l'archivage des documents nécessite une application assurant les fonctionnalités adéquates. L'archivage des documents nécessitant un suivi quotidien et permanent, une application assurant cette fonction serait d'une importance capitale. A cet effet, l'absence d'une application pour la gestion des documents a été constaté à la DIP du MAEP et conduisant ainsi à une recherche difficile et lente des documents archivés. Ainsi, dans ce rapport de stage sont présentées les différentes étapes de conception et de réalisation d'une application web de gestion des archives pour la DIP du MAEP. Cependant, cette application a pour but de permettre à la DIP de gérer l'archivage ainsi que la recherche simple, rapide et efficace des documents archivés. Elle doit implémenter les processus d'enregistrement (enregistrement des indices d'un document et celui du support numérique du document), de la recherche, et d'impression d'un document pour permettre à l'archiviste et ses collaborateurs d'éviter des pertes de temps lors de la recherche. L'application doit également permettre de gérer la numérisation d'un document afin d'avoir une copie du support physique qui peut être imprimée en cas de demande. Avec cette application, la DIP dispose désormais d'une solution pour la gestion de ses différents documents, qu'ils proviennent des services sous tutelle du MAEP ou des autres ministères. L'ancien procédé d'archivage de la DIP étant manuel, cette application permettra d'automatiser ce procédé et ainsi de permettre de gagner en temps et en efficacité lors de la recherche d'un document donné. Enfin, la réalisation de l'application a nécessité l'utilisation d'une méthode dans l'organisation du travail et celle utilisée ici est la méthode agile.

**Mots clés :** *Archivage, Informatiser, support physique, numérisation.*

# Abstract

A good management of document's archiving needs an application that handles appropriate functionalities. Documents' archiving needing a permanent and daily monitoring, the importance of an application handling this function should be crucial. To this end, the lack of an application for document's management has been noticed at DIP of MAEP, leading to a difficult and slow research of archived document. Then, in this training report are showed off the different steps of conception and achievement of a web application of archives' management for the DIP of MAEP. However, this application purpose is to allow the DIP to manage archiving and also simple, quick and effective research of archived documents. It must implement processes of registration (registration of a document indications and those of the document numeric version), searching and printing documents to allow the archivist and his colleagues to avoid waste of time while searching and get a specific idea of those documents location. The application must also allow to manage a document computing in order to get a copy of the physical support which can be printed at the right time if asked. With this application, the DIP gets from now a solution for managing its different documents, either they come from offices under MAEP or other Ministries. The old archiving process being manual, this application will allow to compute that process and then will allow to save time and efficiency while searching a given document. Finally, the achievement of this application has required using a method while organizing the work and the one used here was the agile development model.

**Key words:** *Archiving, compute, physical support, digitization.*



# Introduction

Les institutions publiques doivent faire face à de nombreuses problèmes en matière de gestion des documents : nouvelles réglementations, normes de l'Etat et le respect des délais impartis dans la gestion des dossiers des administrations. De la même manière, les documents produits, tels que les notes, les rapports, les courriers, les courriels etc... sont au cœur du bon fonctionnement de l'administration d'aujourd'hui. Cependant l'organisation, l'archivage ou plus précisément la gestion de ces documents administratifs représente un véritable problème dans les administrations. Pour faire face à ces problèmes de gestion documentaire, l'outil informatique met à la disposition du grand public et des institutions publiques des solutions d'archivage pour aboutir à une organisation rationnelle de l'information. Cette dernière recouvre tous les processus visant à gérer et organiser des documents de manière informatisée. Elle touche toutes les étapes de cycle de vie d'un document, de l'acquisition à l'archivage. Notre thème de mémoire «Développement d'une application pour la gestion des archives du MAEP » s'inscrit dans cette dynamique. Il s'agira de mettre en place une application qui permettra d'automatiser la gestion des archives au profit de la Direction de l'Informatique et pré-archivage (DIP) du Ministère de l'Agriculture, de l'Elevage et de la Pêche (MAEP).

## Problématique

Les documents administratifs et autres productions intellectuelles du MAEP sont déposés à la DIP et cette dernière a pour mission de garantir la sécurisation formelle, l'authentification et la sauvegarde desdits documents. Cependant, vu le besoin à un certains moments d'avoir à porter de mains des documents archivés il y a quelques années, la recherche manuelle devient très difficile et lente. Il en résulte alors une demande de gestion de ces archives. Ce faisant, ce thème met l'accent sur les contraintes que représente la recherche d'un document archivé ainsi que les procédés d'archivage d'un document. En effet, ce thème entend satisfaire la DIP en proposant un système informatisé de gestion de ces archives. Au vu des pertes de temps lors de la recherche des documents archivés, on peut bien rechercher les avantages liés à la mise en place d'une telle application. Dans ce contexte, il se pose alors le problème d'un procédé d'archivage plus simple, de la recherche plus optimisée et très efficace des documents.

---

## Objectif

Ce thème a pour objectif de mettre à la disposition de la DIP une application de gestion des archives qui servira à améliorer l'exécution des différentes actions à effectuer sur un document.

Plus précisément il s'agira de permettre à l'archiviste et ces collaborateurs de pouvoir :

- numériser un document.
- enregistrer un document (enregistrement des indices d'un document et l'enregistrement du fichier numérique du document) .
- effectuer une recherche plus optimisée des documents archivés.
- imprimer un document archivé.

Le présent mémoire fait le point de nos travaux et comporte cinq (5) chapitres. Le premier présente l'historique de la structure d'accueil. Le deuxième présente une revue de littérature sur la gestion des archives. Dans le troisième chapitre, nous présentons les choix techniques opérés en vue de la conception et de la réalisation de la solution proposée. Le quatrième chapitre présente les résultats de la solution proposée et enfin le dernier chapitre présente une discussion sur la solution proposée.

# Historique

## 1.1 Présentation de la structure d'accueil

---

Le ministère de l'agriculture de l'élevage et de la pêche (MAEP) est situé dans le douzième arrondissement de la ville de Cotonou précisément dans le quartier cadjèhoun, il est érigé en face du trésor public sis à l'avenue Jean-Paul 2. Le décret n° 2016-422 du 20 juillet 2016 du MAEP précise ses attributions, son organisation et son fonctionnement. D'après ce décret, il a pour mission la conception, la coordination, la mise en œuvre et le suivi-évaluation de la politique de l'Etat en matière d'amélioration de la production, des revues des producteurs dans le secteur de l'agriculture, de l'élevage et de la pêche, conformément aux lois et règlement en vigueur au Bénin et à la vision de développement du gouvernement. Le MAEP comprend : le ministre, le cabinet du ministre, les personnes et services directement rattachés au ministre, l'inspection générale du ministère, le secrétariat général du ministère, les directions centrales, les directions techniques, les organismes sous tutelle, les organes consultatifs nationaux ou de gouvernance participative.

Pour notre stage dans ce ministère, nous avons été accueilli à la Direction de l'Informatique et du Pré-archivage qui fait partie des directions centrales du MAEP. Il comprend quatre (04) services à savoir :

- le Secrétariat ;
- le Service Informatique ;
- le Service de Pré-archivage et de la Gestion des Savoirs ;
- le Service des Relations avec les Usagers.

La DIP assure en relation avec toutes les structures du Ministère, la conception, la mise en œuvre, la coordination et le suivi-évaluation d'actions intégrées visant à :

- garantir la sécurisation formelle, l'authentification et la sauvegarde des documents administratifs et autres productions intellectuelles ;
- assurer la fluidité et l'accessibilité de l'information ;
- faciliter les relations entre les structures du Ministère et les usagers pour un service public plus efficace et plus efficient ;
- contribuer à la visibilité de l'image, des performances en termes de résultats et de crédibilité du Ministère.

## 1.2 Etude de l'existant

---

Pour assurer sa fonction du garant de la sécurisation formelle, de l'authentification et de la sauvegarde des documents administratifs et autres productions intellectuelles ainsi que celle de l'assureur de la fluidité et de l'accessibilité de l'information, la DIP avait eu à utiliser le logiciel WINISIS pour la gestion des archives du MAEP mais ce logiciel n'est plus utilisé et depuis lors la gestion des archives est manuelle. A cet effet lorsqu'un document est demandé, l'archiviste et ses collaborateurs se chargent de fouiller les boîtes d'archivages disposés sur des rayons dans la salle d'archives pour retrouver le document. Ainsi, quelques indices du document doivent être connus. Il s'agit de la date, l'objet et du type de document. Connaissant ces informations, la recherche se fait sur les rayons portant des boîtes d'archives de ce type de document.

# Revue de littérature

## 2.1 Etat de l'art

---

### 2.1.1 Présentation des solutions existantes

La gestion de l'information est un réel enjeu pour les entreprises. Cependant la gestion des documents est une solution qui permet de répondre à la problématique d'optimisation et de gestion de l'information. C'est un outil structurant qui contribue à faire circuler l'information de manière plus efficace et rend les connaissances moins éparses pour une meilleure gestion des compétences et une meilleure productivité. Ainsi, une multitude d'applications de gestions des documents existent et sont utilisées par le grand public. On peut citer entre autre : Ariane, Archimède, Maarch et la Gapi

**Ariane** couvre l'ensemble des besoins nécessaires à la mise en place d'un archivage de qualité dans les entreprises et les administrations. Ce progiciel permet de gérer tout le cycle de vie des dossiers, de leur création dans les bureaux jusqu'à leur archivage. Afin de mieux s'adapter aux besoins spécifiques des différents métiers des clients, Ariane a été développé en différentes versions à même d'apporter la solution de gestion la plus adaptée à l'environnement professionnel. Ainsi nous avons Ariane Patients et Ariane Santé pour les métiers de la Santé, Ariane Prêts et Ariane Finance pour les métiers de la Finance, Ariane Services pour le secteur public, les entreprises industrielles et de services [6] et [3].

**Archimède** est un logiciel de dépôt institutionnel, qui permet de stocker et de gérer les archives électroniques. C'est un système décentralisé et autogéré par les utilisateurs. Il permet de verser plusieurs documents à la fois, avec les formulaires pour entrer les métadonnées. Il possède un système de navigation et un puissant engin d'indexation (basée sur un vocabulaire contrôlé) et de recherche. Archimède est une open source [7].

**Maarch** propose des solutions sur mesure d'archivage électronique qui permettent : la gestion des archives électroniques et physiques, la sécurisation et standardisation des données archivées, la traçabilité de toutes les actions, le classement automatique des dossiers, la circulation des documents et la gestion des multiples fonds documentaires [8].

**Gapi** est aussi un logiciel de gestion d'archives [9]. Il permet :

- La gestion des entrées : versements / entrées extraordinaires.
- La gestion des éliminations.
- La saisie et l'indexation du répertoire numérique (article, dossier ou pièce cotés).
- La saisie et l'indexation du répertoire analytique (pièces non cotées).
- L'analyse des pièces.
- La gestion des communications.
- La publication : Diffusion sous forme de catalogues électroniques ou sur le Web.

### 2.1.2 Comparaison des solutions existantes

Bien qu'ayant en commun certaines fonctionnalités, ces applications possèdent des spécificités qui les caractérisent. Ainsi en tenant compte de ces spécificités citées dans l'Etude de l'art, on peut conclure que le progiciel Ariane service est plus complet vu les multitudes de fonctionnalités qu'il propose ainsi que les différentes versions sous lesquelles il existe. Chaque version est paramétrable et peut faire l'objet de développements complémentaires, à la demande des clients. Toutes les versions sont architecturées de façon à permettre aux utilisateurs de retrouver un environnement de travail connu et maîtrisable rapidement. Ariane est intégré c'est-à-dire que vous n'avez pas à ressaisir inutilement des informations déjà entrées dans le système. De plus Ariane est un site web. Ce n'est pas un logiciel installé sur un ordinateur. Il est évolutif (de nouveaux modules et services sont ajouté au fil du temps). Cependant à la différence d'Ariane, Archimède gère les archives électroniques, il s'installe sur un ordinateur mais il est un logiciel libre. Maarch s'installe également sur un ordinateur et c'est un logiciel libre aussi. Gapi gère les documents électroniques, il est payant.

## 2.2 Interêt du sujet

---

Notre application est conçue pour une structure donnée qui est le MAEP en tenant compte de ses réalités dans le domaine d'archivage. En effet, l'application ne gère pas l'intégralité des fonctions qu'assurent les applications de gestions d'archive citées précédemment. Elle assure uniquement : l'acquisition des documents, l'archivage, la recherche, le stockage, la numérisation et l'impression des documents puisqu'elle a été développée en tenant compte des besoins de la structure. Dans la suite nous décrirons la méthodologie utilisée ainsi que les matériels.

## Matériel et méthodes

### 3.1 Matériel

---

#### Matériel

La réalisation de cette application a nécessité l'utilisation des éléments suivants :

- **Une imprimante** : C'est un périphérique permettant de faire une sortie imprimée des données de l'ordinateur. L'application développée a une fonctionnalité Impression, l'imprimante sera alors connecté à l'ordinateur et permettra de récupérer le support papier.
- **Un scanner ou « numériseur à balayage »** : C'est un périphérique informatique qui permet de numériser des documents ou autres, comme les empreintes digitales par exemple. Il servira à numériser les documents dans cette application.
- **Un ordinateur** : C'est une machine électronique programmable qui fonctionne par la lecture séquentielle d'un ensemble d'instruction.

#### Logiciels

L'intégralité du travail a été effectuée sur un ordinateur portable personnel. Vue l'avantage des frameworks dans le développement, il serait adéquat d'opter pour l'un d'eux dans la réalisation du travail. Cependant le framework utilisé ici est Laravel.

Un framework est un ensemble de préconisations pour la conception et le développement d'applications. Laravel est un framework Open-Source écrit en langage PHP. Son architecture est clairement orientée MVC, il permet une facilité lors du développement, il est facile d'utilisation et permet au développeur de ne pas se perdre dans son code, il respecte les conventions et les bonnes pratiques en PHP. Ce framework organise également l'interface homme-machine d'une application. Le principe d'une telle structure est de diviser l'application en 3 parties distinctes :



- **Les Modèles** : cette partie récupère les informations dans la base de données et celles-ci sont traitées par le contrôleur. Elle sert donc à récupérer, aider et organiser les données d'un site.
- **Les Vues** : cette partie gère l'interface, c'est-à-dire ce qui s'affiche à l'utilisateur en fonction des données envoyées par le Contrôleur.
- **Les Contrôleurs** : cette partie gère les décisions, c'est-à-dire qu'elle communique avec le modèle pour prendre des informations, les étudier, prendre des décisions pour ensuite récupérer les informations à afficher et les rendre à l'utilisateur. Il sert donc d'intermédiaire entre le modèle, la Vue et l'utilisateur.

Ce framework a été choisi parce que l'un de ces principes repose sur sa simplicité et sa clarté d'écriture. Il permet au développeur de normaliser et d'ordonner son développement pour que ce dernier soit très facilement lisible par d'autres et surtout ne pas se perdre durant son projet. De plus il a été construit en se basant sur Symfony, un autre framework PHP reconnu mondialement pour sa robustesse. De ce fait, il embarque des briques logicielles testées et approuvées par une grande communauté permettant d'améliorer la rapidité des développements, la robustesse de l'application. Cependant, pour le développement de cette application le langage utilisé est le PHP. A elle s'ajoutent le CSS et du bootstrap.

**PHP** : il s'agit d'un langage facile d'apprentissage, accessible sur la plupart des systèmes d'exploitations et très populaire sur le web, ce qui permet un meilleur support et une meilleure maintenance. De plus, il s'agit d'un langage déjà éprouvé depuis plusieurs années et donc assez robuste pour répondre aux besoins de l'entreprise, qui veut s'appuyer sur des technologies matures et fiables pour fonctionner de manière optimale. Enfin, il est assez facile d'apprentissage, ce qui permettra à de futur développeurs de maintenir ou de faire évoluer rapidement l'application [4].

**Bootstrap** : c'est un framework front-end gratuit pour le développement web. Il contient plusieurs outils utiles à la réalisation de sites interactifs. Il permet entre autre de concevoir plus facilement un design responsive qui va permettre d'adapter l'affichage de l'application à tout type d'écran. Il fait partie des frameworks les plus populaires. Il est conçu pour être compatible avec tous les navigateurs majeurs tels que Google Chrome, Firefox, Safari ou encore Opéra. Le concept de Bootstrap est basé sur les grilles. Chaque élément de l'interface se situe à l'intersection d'une ligne et d'une colonne. Cette grille sert d'armature à la totalité de l'interface et est également très pratique en termes de design responsive [4].

**CSS** : Les feuilles de style en cascade, ou en anglais, Cascading Style Sheet, forment un langage informatique décrivant la manière dont les éléments d'une interface HTML doivent être

affichés. CSS est un langage utilisé par tous les navigateurs. C'est le langage standard pour la réalisation d'interfaces web riches. L'utilisation de CSS permet d'améliorer la maintenabilité du code, son évolution, et plus généralement le design de l'application, la rendant ainsi plus agréable à utiliser [4].

En outre, la modélisation réalisée pour l'application développée dans le cadre de ce thème a été réalisée avec le logiciel **ArgoUml**. C'est un logiciel libre de création de diagrammes UML. Programmé en Java, il supporte la génération de code et l'ingénierie inverse. La base de données utilisée est MySQL. Ce dernier est un système de gestion de bases de données relationnelles.

## 3.2 Méthodes

---

### 3.2.1 Analyse des besoins

Cette analyse a permis de chercher, de comprendre et de décrire de façon précise les besoins de la DIP en matière de gestion des archives. En effet, nous avons eu à réaliser des interviews avec l'archiviste de la DIP afin de pouvoir identifier les besoins. Ainsi, nous avons pu identifier ce qu'ils souhaitent faire avec le logiciel, les fonctionnalités voulues, et comment les fonctionnalités devraient-elles fonctionner. Cependant la DIP a besoin d'une application pour gérer ses archives. A cet effet, elle souhaite :

- pouvoir enregistrer des documents, à cet effet elle souhaite que la fonctionnalité «enregistrer document» permette d'enregistrer toutes les informations nécessaires concernant un document telles que : le nom de la structure où le document est produit, le type du document, la référence du document, la date de production ainsi que la date d'enregistrement du document. Toutes ces informations doivent être rangées dans une table dans l'application.
- numériser des documents : la fonctionnalité «numériser document» permettra de communiquer avec le scanner à travers l'application pour numériser le document.
- rechercher, ouvrir et imprimer un document donné : avec la fonctionnalité «rechercher document» elle doit pouvoir rechercher un document quelconque, l'ouvrir grâce à la fonctionnalité «ouvrir document» et arriver à l'imprimer grâce à la fonctionnalité «imprimer document».
- créer des boîtes d'archives pour conserver des documents par type, et créer des rayons pour stocker les boîtes d'archives ainsi donc les fonctionnalités «créer une boîte d'archive» et «créer un rayon» permettent de satisfaire ces besoins.
- gérer les utilisateurs (créer et supprimer des utilisateurs), ce qui fait appel aux fonctionnalités «créer un utilisateur» et «créer un archiviste».

### 3.2.2 Modélisation

A cette étape, les informations de l'analyse des besoins ont été exploitées pour aboutir à la réalisation des diagrammes de cas d'utilisations, des diagrammes de classes, et de séquences.

#### **-Diagramme de cas d'utilisation**

Les fonctionnalités étant déjà définies dans l'analyse des besoins, le diagramme de cas d'utilisation est réalisé pour chaque acteur. Ces diagrammes se présentent comme suit :

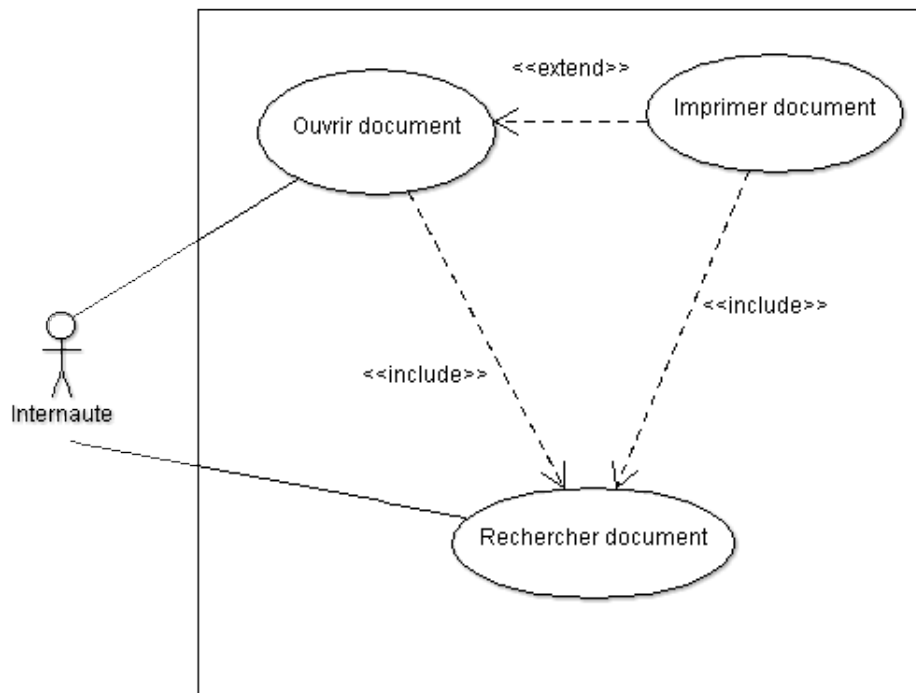


FIGURE 3.1 – Diagramme de cas d'utilisation définissant les fonctionnalités de l'internaute

**Internaute :**

**Description :** c'est un simple utilisateur ayant des fonctionnalités restreintes. Il a juste accès à la page d'accueil où il peut chercher un document donné, l'ouvrir et l'imprimer au besoin. Cependant, il n'a pas besoin de s'authentifier pour effectuer ces actions.

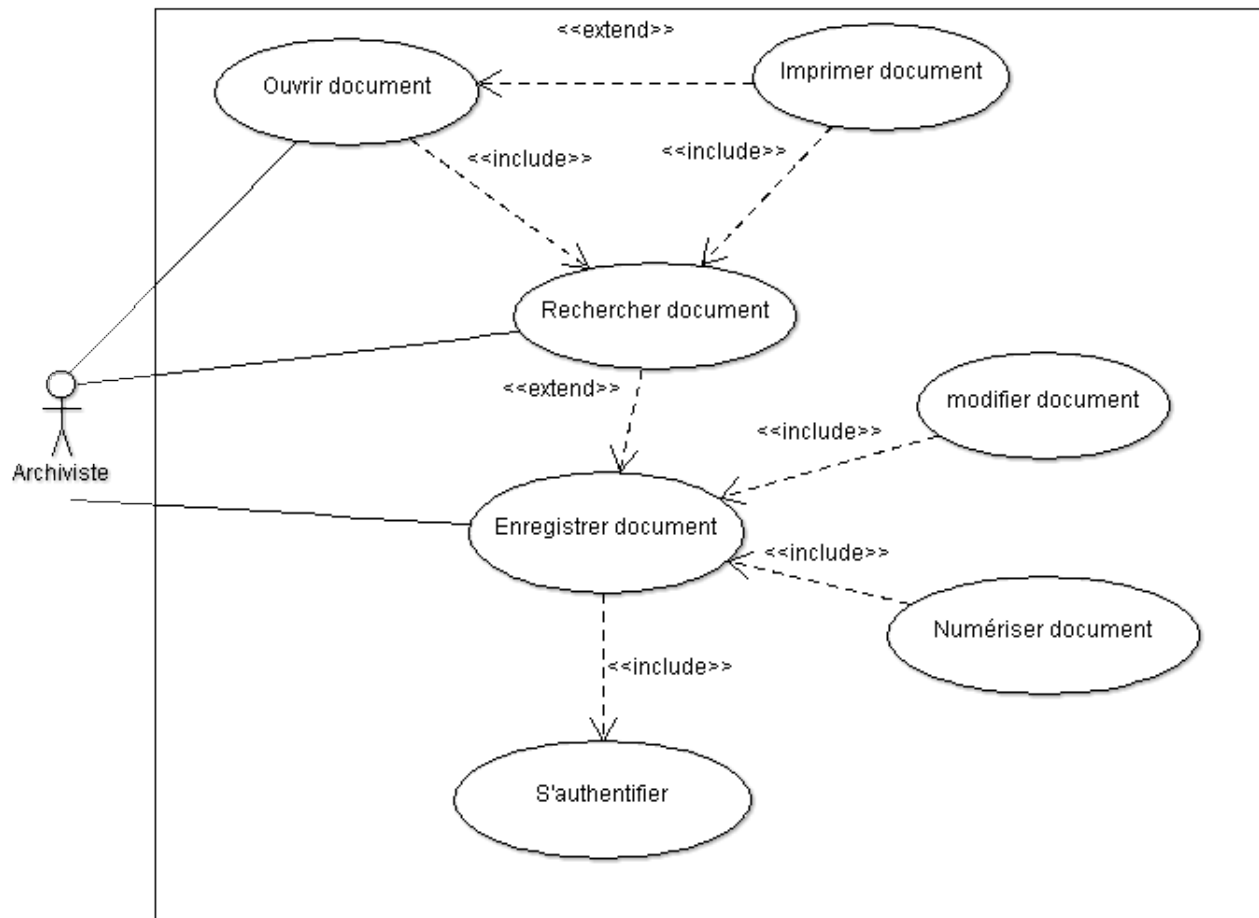


FIGURE 3.2 – Diagramme de cas d'utilisation définissant les fonctionnalités de l'archiviste

### Archiviste :

**Description :** c'est l'acteur principal de l'application. Il a la possibilité de rechercher un document, l'ouvrir et l'imprimer. Pour la fonctionnalité «rechercher », il lui est prévu en plus de la recherche rapide, une possibilité de recherche par intervalle de temps pour plus d'efficacité dans la recherche des documents. Ce qui lui permet de connaître les documents enregistrés pendant une période donnée. Il peut également modifier les indices d'un document en cas d'erreur. Il doit nécessairement s'authentifier avant d'effectuer ces différentes actions.

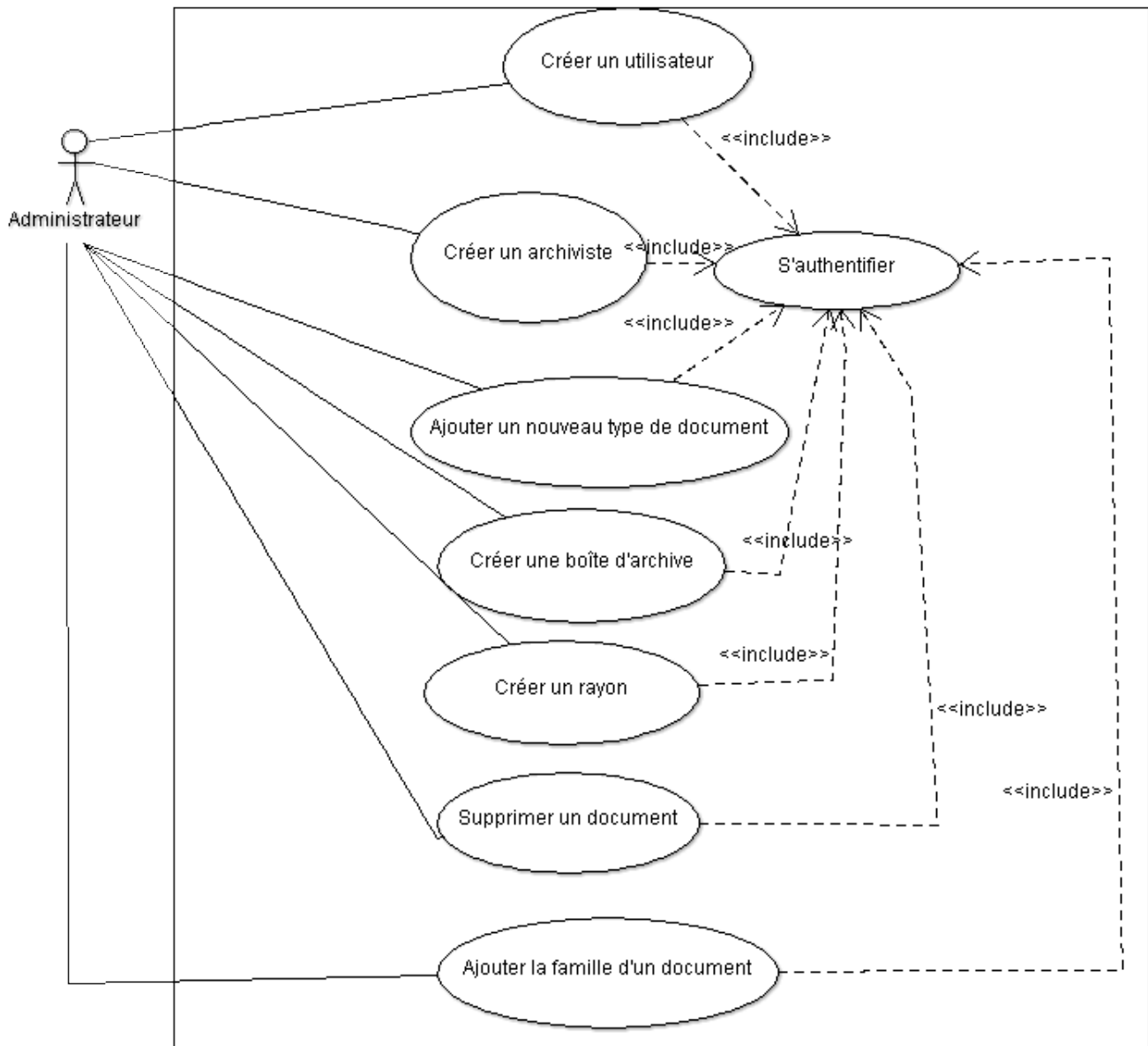


FIGURE 3.3 – Diagramme de cas d'utilisation définissant les fonctionnalités de l'administrateur

**Administrateur :**

**Description :** il administre le système. Pour effectuer ses différentes tâches, il faut également qu'il s'authentifie. En plus des fonctionnalités propres à lui que sont : créer un archiviste, créer un utilisateur, ajouter un nouveau type de document, supprimer un document, ajouter les familles de document, il a également la possibilité d'effectuer toutes les tâches qu'effectue l'archiviste à savoir : rechercher document, ouvrir document et modifier document.

Voici le diagramme de cas d'utilisation globale qui montre les fonctionnalités des différents acteurs

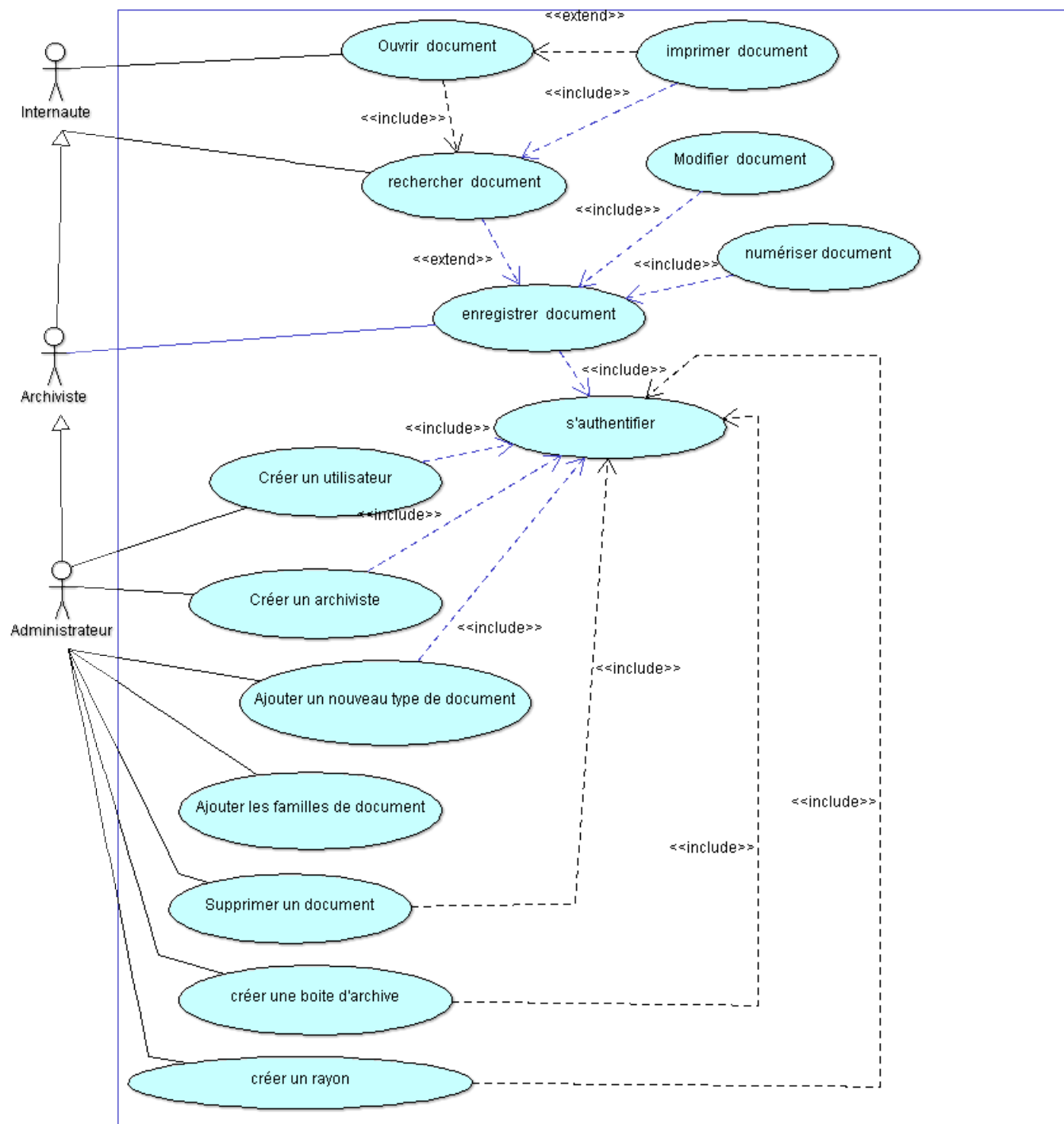


FIGURE 3.4 – Diagramme de cas d'utilisation définissant les fonctionnalités des trois acteurs

## Scénario de quelques cas d'utilisation

### Cas d'utilisation «Enregistrer document»

#### Identificateur

Nom du cas : «Enregistrer document».

But : décrire les étapes permettant à l'archiviste d'enregistrer un document via l'application.

Acteur principal : Archiviste.

Acteur secondaire : néant.

#### Séquencement

Le cas d'utilisation commence lorsque l'archiviste remplit le formulaire en y insérant des informations concernant le document à savoir : le nom de la structure, le type du document, sa référence, son objet, sa date d'enregistrement et sa date de production.

**Pré-condition** L'archiviste s'authentifie

#### Enchaînement nominal

1. L'application vérifie le mot de passe.
2. Remplissage du formulaire.
3. Vérification de la conformité des valeurs saisies dans les champs du formulaire.
4. Exécution de l'enregistrement après validation suite à un clic sur le bouton **Enregistrer**.

#### Enchaînement alternatif

**A1-** L'un des champs du formulaire n'est pas rempli.

L'enchaînement démarre après le point 3 de la séquence nominale :

4- l'application indique à travers un message d'alerte que l'un des champs est vide et il invite l'utilisateur à remplir tous les champs.

La séquence nominale reprend au point 2.

#### Enchaînement d'exception

**E1-** Le mot de passe n'est pas valide.

L'enchaînement démarre après le point 1 de la séquence nominale :

2- l'application indique à travers un message d'alerte que le mot de passe n'est pas valide et il invite l'utilisateur à corriger son mot de passe.(l'utilisateur a droit à trois tentatives).

**E2-** La valeur saisie dans un champ n'est pas correcte.

L'enchaînement démarre après le point 3 de la séquence nominale :

4- l'application indique à travers un message d'alerte que la valeur de l'un des champs n'est pas valide et il invite l'utilisateur à remplir à nouveau tous les champs.(l'utilisateur a droit à trois tentatives).

La séquence nominale reprend au point 2.

#### Post-condition



Les informations concernant le document sont enregistrées dans la base de données.

### **Rubrique optionnelle**

#### **-Contraintes non fonctionnelles**

Néant.

#### **-Contrainte liée à l'interface homme-machine**

Après l'exécution de l'enregistrement, envoyer un message succès d'enregistrement à l'utilisateur.

Cas d'utilisation «Imprimer document»

### **Identificateur**

Nom du cas : «Imprimer document».

But : décrire les étapes permettant à l'archiviste d'imprimer un document.

Acteur principal : Archiviste.

Acteur secondaire : néant.

### **Séquencement**

Ce cas d'utilisation commence après avoir cliqué sur le bouton ouvrir pour pouvoir ouvrir le document numérisé dans le navigateur.

### **Pré-condition**

L'imprimante est connectée à l'ordinateur.

Le document à imprimer est recherché et est trouvé. (Appelle du cas «rechercher document»)

### **Enchaînement nominal**

1. L'interface de configuration de l'impression s'affiche après un clic sur Impression dans la barre de menu de la fenêtre qui apparaît suite à un clic sur la fonction «Imprimer» du navigateur.
2. L'impression s'effectue.
3. L'imprimante délivre le document

### **Enchaînement alternatif**

Néant

### **Enchaînement d'exception**

Néant

### **Post-condition**

L'archiviste prend le document.

Cas d'utilisation «Rechercher document»

### **Identificateur**

Nom du cas : «Rechercher document».

But : décrire les étapes permettant à l'archiviste de rechercher un document.

Acteur principal : Archiviste.

Acteur secondaire : néant.

### **Séquencement**

Ce cas démarre à la deuxième pré-condition du cas «Imprimer document»

### **Enchaînement nominal**

1. L'utilisateur saisit des mots clés dans la zone de recherche.
2. Affiche la liste des documents comportant le mot clé.
3. L'utilisateur choisit le document qu'il cherche.

### **Enchaînement alternatif**

Néant

### **Enchaînement d'exception**

E1- Le document n'est pas trouvé

L'enchaînement reprend au point 2 de la séquence nominale :

2- Un message d'alerte s'affiche indiquant que le document n'a pas été retrouvé

### **Post-condition**

Le document est trouvé et ses indices sont affichés dans un tableau.

### **Rubrique optionnelle**

#### **-Contraintes non fonctionnelle**

Néant

#### **-Contrainte liée à l'interface homme-machine**

Ne pas afficher plus de 10 documents à la fois dans la liste.

### **-Diagramme de classes**

Les classes implémentées font partie de la logique de l'application. Celles-ci représentent les entités utilisées dans l'application, et plus précisément leurs attributs ainsi que les méthodes qu'ils utilisent pour traiter les données. Ces classes s'appuient essentiellement sur les entités réelles manipulées par les archivistes.

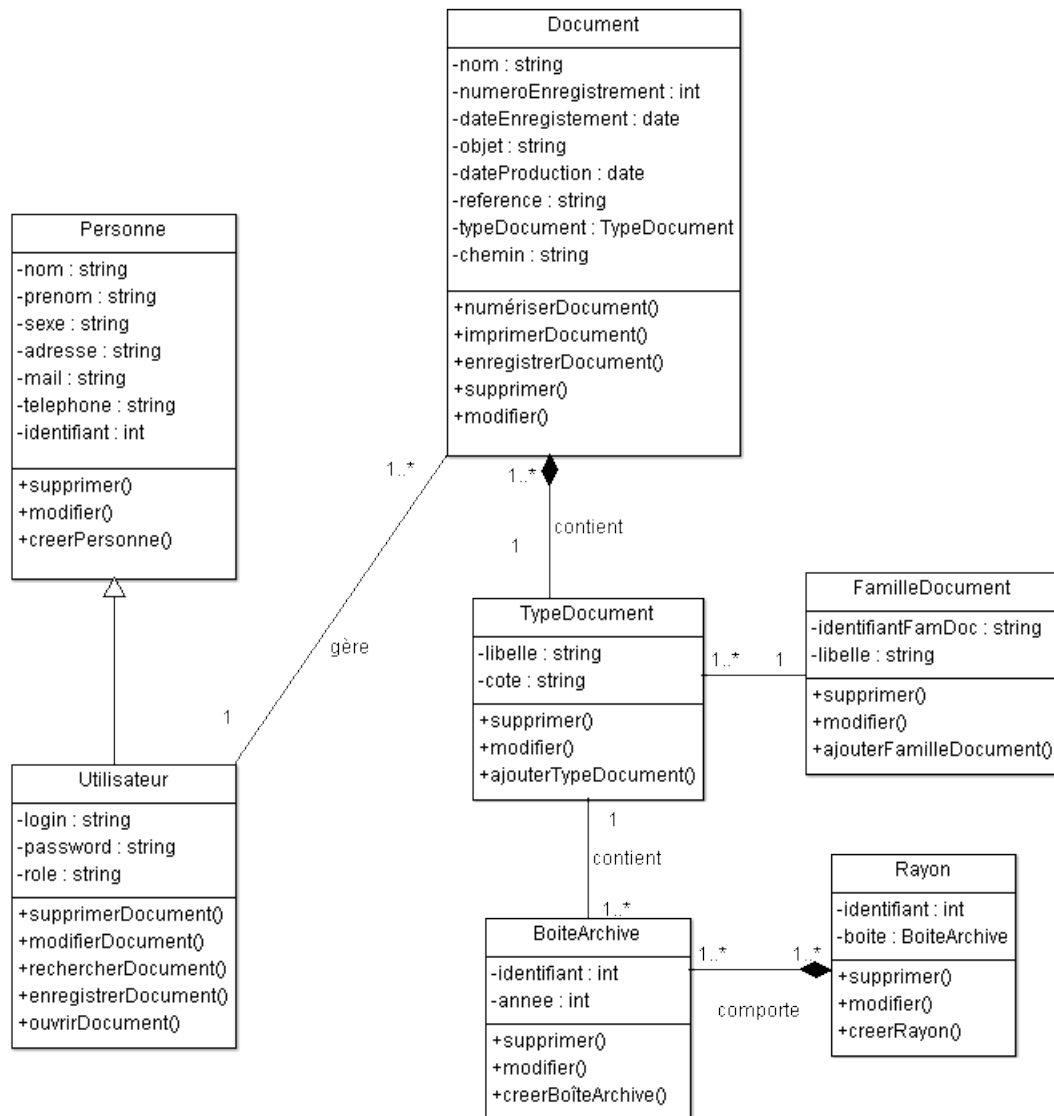


FIGURE 3.5 – Diagramme de classes

### Explication de l'origine des classes

Avec l'application, l'archiviste doit être en mesure d'enregistrer, de rechercher, de modifier, d'ouvrir et d'imprimer des documents. Un document a un nom, un numéro d'enregistrement, un type, un objet, une référence, une date d'enregistrement, une date de production et un chemin vers le lieu de son stockage, cela donne naissance à la classe Document avec pour attribut nom, numeroEnregistrement, typeDocument, objet, reference, dateEnregistrement, dateProduction et chemin. Cette classe a pour méthodes imprimerDocument(), enregistrerDocument(), supprimer(), modifier() et numériserDocument().

Puisque lors de l'enregistrement d'un document, l'archiviste attribue à ce dernier un type qui appartient à une famille donnée, cela donne les classes TypeDocument et FamilleDocument. Un type a une côte et un libellé, de la même manière une famille a une côte et un libellé ainsi, la classe typeDocument a pour attributs côte et libelle avec les méthodes supprimer(), modifier() et ajouterTypeDocument(). La classe FamilleDocument a pour attribut indentifiantFamDoc (cote) et le libelle avec les méthodes supprimer(), modifier() et ajouterFamilleDocument().

L'application doit permettre de créer des boîtes d'archives et des rayons pour stocker des documents. Une boîte d'archive a un identifiant, une année et comporte un type de document donné cela donne la classe BoiteArchive avec les attributs : identifiant, annee et les méthodes supprimer(), modifier(), creerBoiteArchive(). Un rayon a également un identifiant et il comporte des boîtes d'archive d'où la classe Rayon avec les attributs :indentifiant, boite et les méthodes supprimer(), modifier(), creerRayon().

L'application doit également permettre de gérer les utilisateurs et cette gestion est assurée par l'administrateur. Un utilisateur a un login, un mot de passe et un rôle cela donne la classe Utilisateur avec les attributs login, password, role avec les méthodes enregistrerDocument(), rechercherDocument(), ouvrirDocument(), supprimerDocument() et modifierDocument(). Cependant, l'utilisateur est une personne. Cette dernière a un numero, un nom, un prénom, un sexe, une adresse, un mail et un téléphone. Il doit être en mesure d'enregistrer, de modifier, de rechercher et d'ouvrir un document. D'où la classe Personne avec les attributs : nom, prenom, sexe, adresse, mail, telephone et les méthodes creerPersonne(),supprimer(), modifier().

### Explication des relations et des cardinalités

Un utilisateur est soit un utilisateur simple, un archiviste ou un administrateur. Le fait qu'une Personne soit un utilisateur donne un héritage entre la classe Personne et la classe Utilisateur. Cependant la classe Utilisateur hérite de la classe Personne, ce qui veut dire que la classe Utilisateur hérite des attributs et méthodes de la classe Personne en plus de ces propres attributs et méthodes. La relation d'agrégation ainsi que les cardinalités entre la classe Document et Type-

Document s'explique par le fait qu'un type de document donné contient plusieurs documents et qu'un document n'a qu'un et un seul type. De la même manière une famille de document donné est composée de plusieurs types de document et un type de document donné n'a qu'une et une seule famille. C'est ce qui justifie les cardinalités entre la classe TypeDocument et FamilleDocument.

En effet, dans notre application, on voudrait que les documents soient rangés par type dans des boîtes d'archives et que ces dernières soit contenues dans des rayons comme dans une salle d'archives. Ce qui justifie la relation entre les classes TypeDocument, BoîteArchive et Rayon. Une simple association entre la classe TypeDocument, BoîteArchive et une agrégation entre la classe BoîteArchive et la classe Rayon. Les cardinalités entre la classe TypeDocument et BoîteArchive s'expliquent par le fait qu'une boîte d'archive contient plusieurs documents d'un même type et un document d'un type donné est contenu dans une et une seule boîte d'archive. Enfin, un rayon comporte plusieurs boîtes d'archives et une boîte d'archive contenant des documents d'un type donné est contenue dans un ou plusieurs rayons, ce qui justifie la cardinalité entre la classe Rayon et la classe BoîteArchive.

### Diagrammes de séquences

Le diagramme ci-dessous montre les différents scénarios exécutés lors de l'ouverture d'un document.

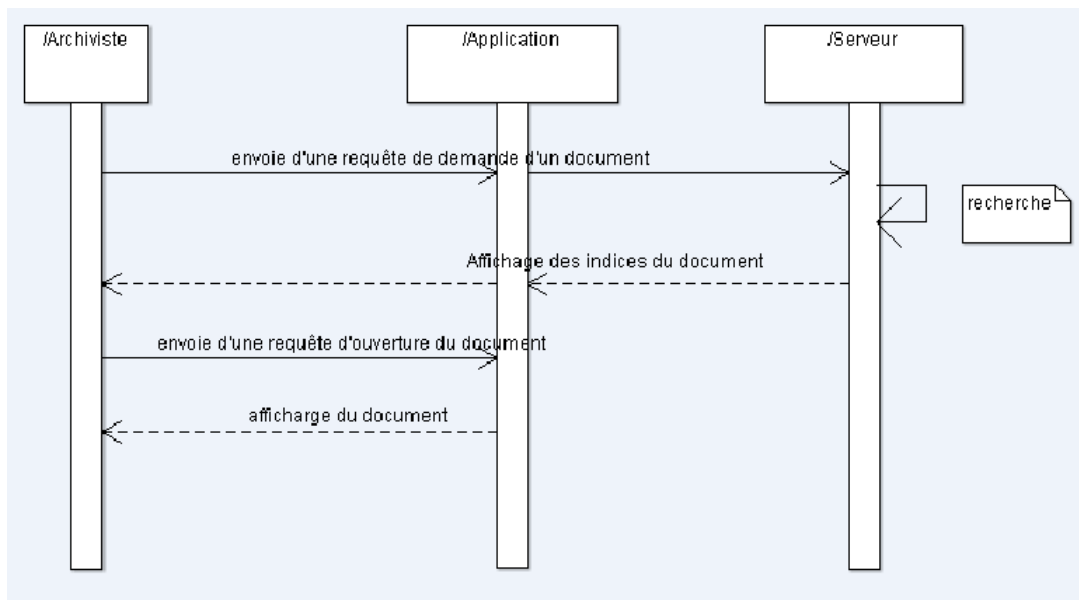


FIGURE 3.6 – Diagramme de séquence de la fonctionnalité Ouvrir document

Le diagramme ci-dessous montre les différents scénarios exécutés lors de l'impression d'un document.

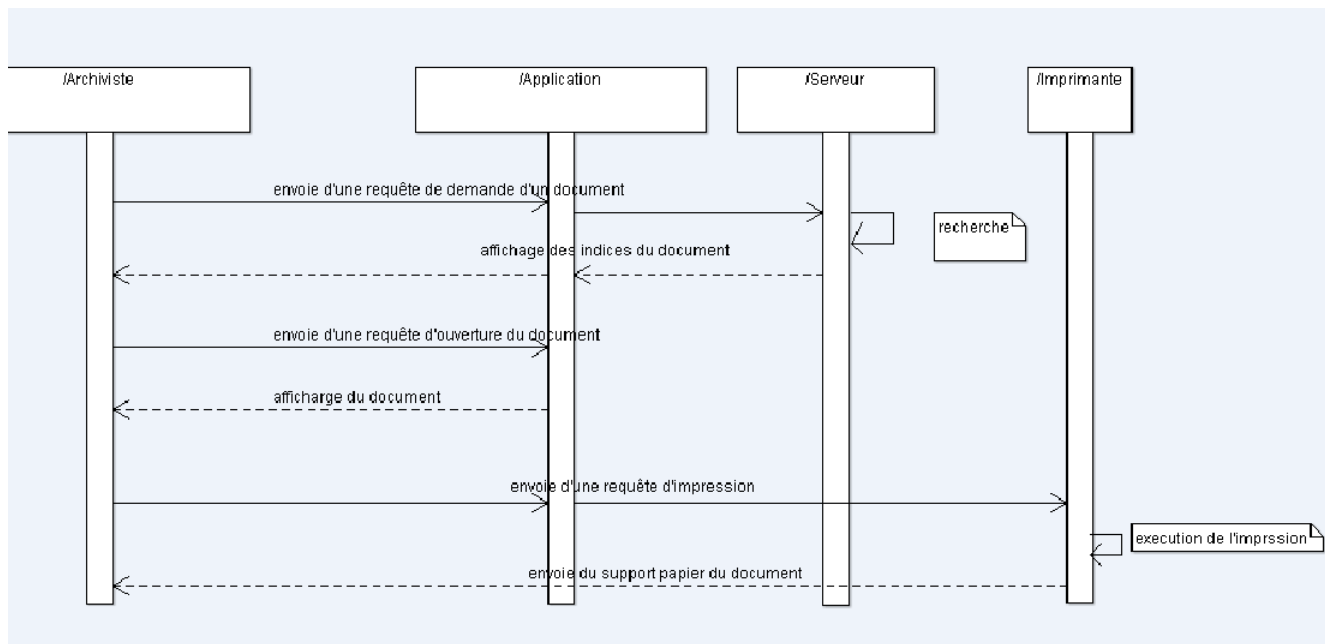


FIGURE 3.7 – Diagramme de séquence de la fonctionnalité Imprimer document

La réalisation de cette application s'est déroulée dans le respect des principes de la méthode agile [2]. Cette dernière utilise un principe de développement itératif qui consiste à découper le projet en plusieurs étapes qu'on appelle « itérations ». C'est donc une méthode itérative et incrémentale. Ces itérations sont en fait des mini-projets définis avec le client en détaillant les différentes fonctionnalités qui seront développées en fonction de leur priorité. Le chef de projet établit alors un macro planning correspondant aux tâches nécessaires pour le développement de ces fonctionnalités. Le but est d'assumer le fait que l'on ne peut pas tout connaître et anticiper quel que soit notre expérience. On découpe alors le projet en itération plutôt que de tout prévoir et planifier en sachant que des imprévus arriveront en cours de route. Elle offre :

**Une meilleure qualité de la communication :** l'utilisateur a la possibilité de clarifier ses exigences au fur et à mesure.

**Une meilleure visibilité :** le client a une meilleure visibilité sur l'avancement des travaux.

**Un meilleur contrôle de la qualité :** les tests sont effectués en continu

**Une meilleure détection des risques :** les risques sont détectés plus tôt

**Une motivation et confiance de l'équipe :** satisfaction d'atteindre un objectif fixé

**Un contrôle des coûts :** le projet peut être arrêté s'il n'y a plus de budget

Il existe plusieurs méthodes agiles, nous avons :

**ASD (Adaptive software Development)** créée par Jim Highsmith en 2000. Ces principales caractéristiques sont :

- Focaliser sur l'objectif
- Se baser sur des composants
- Itérer
- Découper le temps et fixer des deadlines
- Piloter le projet par les risques
- Accepter le changement

### **Crystal**

Cette méthode a été mise au point par Alistair Cockburn (signataire du Manifeste) en 1997. Elle consiste à sélectionner la méthode applicable en fonction du nombre de personnes à coordonner. Ses caractéristiques principales sont :

- Des livraisons fréquentes
- Des aménagements permanents
- Une bonne communication interpersonnelle



- Confiance, liberté d'expression et sécurité personnelle
- Focus sur l'objectif et disponibilité
- Un contact permanent avec les utilisateurs
- Un environnement de travail approprié pour l'automatisation des tests, la gestion de configuration et les intégrations fréquentes
- Une collaboration étroite entre toutes les parties prenantes, y compris en dehors de l'équipe
- Une réflexion constante sur ces propriétés

### Scrum

Le Scrum ou « mêlée », créée par Ken Schwaber et Jeff Sutherland (signataires du Manifeste) en 1993, est un terme emprunté au rugby qui désigne la solidarité et la force qui lie les membres de l'équipe au succès de l'itération [1]. Le cycle de vie de Scrum est rythmé par des itérations de deux ou quatre semaines qu'on appelle sprints. Les valeurs mises en avant par cette méthode sont les suivantes :

- Visibilité : Avoir une vision réelle sur le résultat
- Inspection : Vérifier l'écart par rapport à l'objectif initial
- Adaptation : S'adapter en fonction des écarts constatés afin de les ajuster. Scrum est favorable à des petits ajustements fréquents. En effet, le scrum définit seulement 3 rôles :

**Le Product Owner :** il définit les fonctionnalités du produit. Il décide des dates de déploiement et priorise également les fonctionnalités grâce à la fameuse liste "Product Backlog" (liste des tâches à exécuter) en accord avec le marché (demandes des clients, etc..). Ainsi, il est responsable du futur succès du produit et ajuste les fonctionnalités et leurs priorités à chaque sprint si besoin.

**Le Scrum Master :** il doit maîtriser Scrum et s'assurer que ce dernier est correctement appliqué. Il a donc un rôle de coach à la fois auprès du Product Owner et auprès de l'équipe de développement. Il doit donc faire preuve de pédagogie. Il est également chargé de s'assurer que l'équipe de développement est pleinement productive. Généralement le candidat tout trouvé au rôle de Scrum Master est le chef de projet. Celui ci devra cependant renoncer au style de management « commander et contrôler » pour adopter un mode de management participatif.

**L'Equipe de Développement :** est chargée de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables. Elle est pluridisciplinaire et peut donc encapsuler d'autres rôles tels que développeur, architecte logiciel, DBA, analyste fonctionnel, graphiste/ergonome, ingénieur système.

Cependant, des réunions de projets sont organisées entre les individus de l'équipe de développement. Ces réunions sont appelées des cérémonies, elles sont au nombre de 4 et se présentent comme suit :

#### **Le "Sprint planning"**

Une cérémonie par Sprint (2/3 heures par planning).

Durant les Sprint plannings, l'équipe sélectionne les tâches provenant de la "Product Backlog" qui sont éligibles au prochain sprint, c'est à dire terminables en un unique Sprint. A ce moment, une "Sprint Backlog" est créé. Les tâches sont identifiées et sont estimées afin de veiller à ce que l'intégralité des tâches soit terminée durant le Sprint.

#### **Le "Daily Scrum"**

Une cérémonie par jour (15 minutes max).

Souvent appelés "Standup meetings", les "Daily Scrum" durent environ 15 minutes et ont lieu chaque matin. Chaque membre de l'équipe doit être debout et est invité à répondre aux 3 questions suivantes :

Qu'est ce que j'ai fait hier ?

Que vais-je faire aujourd'hui ?

Ai-je rencontré des problèmes ?

Cette réunion quotidienne permet d'éviter d'autres réunions inutiles et évitables. Le "Daily meeting" n'est pas le moment opportun pour résoudre les problèmes mais plutôt de les énumérer pour informer tous les membres de l'équipe.

#### **La "Sprint review"**

Une cérémonie à la fin de chaque Sprint (2 heures par review).

Durant 2 heures à la fin de chaque sprint ont lieu les revues de Sprint : "Sprint reviews". Durant ces "reviews", l'équipe présente ce qu'elle a accompli durant le Sprint qui se termine. Les revues prennent généralement la forme de démonstration des nouvelles fonctionnalités et de l'architecture sous-jacente, les Powerpoints et slides sont interdits. Des gens externes aux projets (parties prenantes par exemple) sont éventuellement les bienvenus pour voir l'avancement des projets.

#### **La "Sprint retrospective"**

Une cérémonie à la fin de chaque Sprint (1 heure par rétrospective).

Enfin, la rétrospective du Sprint a lieu généralement juste après la revue du Sprint :

“Sprint review”. La rétrospective permet de prendre du recul sur ce qui a marché ou n’a pas marché durant le Sprint. Des actions sont déterminées par l’équipe afin d’améliorer les prochains cycles de développement

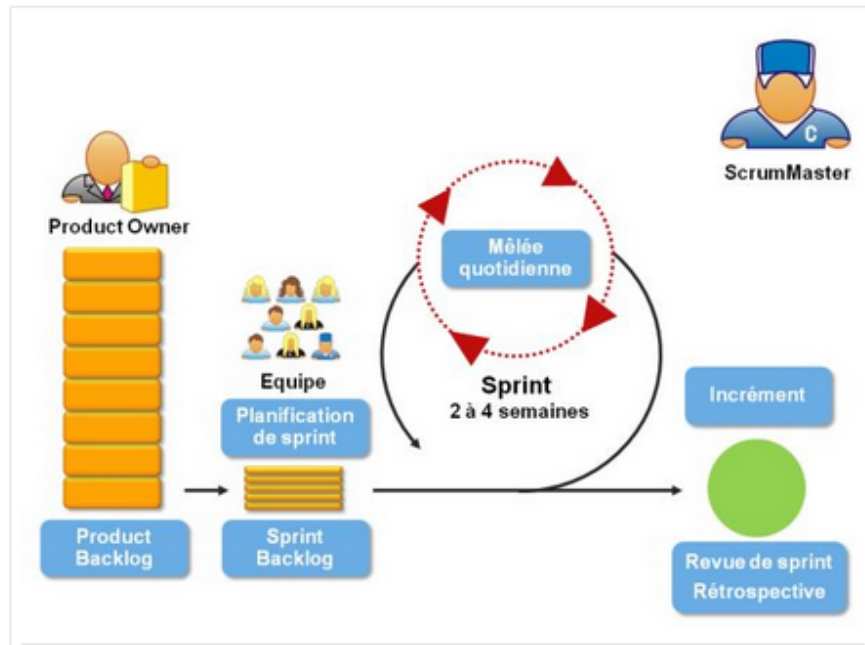


FIGURE 3.8 – Les différentes étapes du scrum.

Il existe également d'autres méthodes agiles :

- DSDM (Dynamic Software Development Method) créée en Grande-Bretagne en 1995.
- RAD (Rapid Application Development) créée par James Martin en 1991.
- XP (eXtreme Programming) créée en 1999.

En effet, parmi les méthodes citées plus haut, le scrum a été celle que nous avons choisi pour la conception de l'application. Cependant, puisque cette méthode est rythmée par des itérations de deux ou quatre semaines, nous avons choisi d'itérer après chaque deux semaines. Nous avons donc découpé la conception de l'application en trois grandes tâches à savoir :

- La réalisation des différentes interfaces de l'application.
- La conception de la base de données.
- L'implémentation des différentes fonctionnalités de l'application.

L'application comporte au total quatre grandes fonctionnalités à savoir : l'enregistrement, la recherche, la numérisation, l'impression d'un document et chacune d'elles a été implémentée en deux semaines ainsi que la réalisation des interfaces et la conception de la base de données par lesquelles nous avons commencé. En effet, en fonction des trois rôles définis par le scrum, le Product owner est le directeur de l'informatique et du pré-archivage du MAEP, nous sommes le scrum master et l'équipe de développement également, car dans le cadre de ce travail, l'application est à réaliser par le candidat uniquement. De façon résumée, la méthode agile valorise les individus et leurs interactions plus que les processus et les outils, des logiciels opérationnels plus qu'une documentation exhaustive, la collaboration avec les clients plus que la négociation contractuelle, l'adaptation au changement plus que le suivi d'un plan.

La méthode de modélisation choisie est UML. En effet, pour réaliser les phases de l'analyse et de la conception dans un projet informatique, nous utilisons des méthodes, des conventions et des notations. Le langage de modélisation unifié UML, fait partie des notations les plus utilisées aujourd'hui. La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel [10].

## Résultats

Au cours de notre stage, nous avons développé au profit du MAEP une application web que nous avons appelé ARCHIVE-Gest. Ledit développement permet l'utilisation des langages HTML et css du côté client et du côté serveur le PHP. Ici nous présentons quelques interfaces de l'application ainsi que les remarques issues des tests d'acceptabilité.

### Les interfaces de l'application

#### 1-Liste des documents enregistrés

Cette liste de document se présente à la page d'accueil. Grâce à la zone de recherche proposée par la table, on peut rechercher un document et l'ouvrir à l'aide du bouton Ouvrir. (cf fig 4.1)

| Gestion des archives |   |                        |  |                              |                    |                         | Connexion              |
|----------------------|---|------------------------|--|------------------------------|--------------------|-------------------------|------------------------|
| Liste des documents  |   |                        |  |                              |                    |                         |                        |
| Show                 | 10  | ▼                      | entries  | Search: <input type="text"/> |                    |                         |                        |
| Structure            | Type  | Référence              | Objet  | Date d'enregistrement        | Date de production | Numero d'enregistrement | Action                 |
| DDR                  | Loi   | n°2/MCVDD /DC/SGM/DDR  | Loi portant nomination de .....  | 2017-06-03                   | 2017-05-03         | 13                      | <a href="#">Ouvrir</a> |
| DGM                  | Décret  | n°2/MCVDD /DC/SGM/DGM  | Decret portant nomination du directeur de l'informatique et du pré-archivage | 2017-06-04                   | 2017-05-03         | 14                      | <a href="#">Ouvrir</a> |
| DRFM                 | Loi   | n°2/MCVDD /DC/SGM/DRFM | Decret portant...  | 2017-05-14                   | 2017-05-03         | 3                       | <a href="#">Ouvrir</a> |
| DRFM                 | Communication introduite par le ministère/fiche | n°2/MCVDD /DC/SGM/DRFM | Communication  | 2017-06-03                   | 2017-05-09         | 12                      | <a href="#">Ouvrir</a> |
| DRFM                 | Ordre du jour                                   | n°2/MCVDD /DC/SGM/DRFM | Ordre du jour  | 2017-06-04                   | 2017-05-09         | 15                      | <a href="#">Ouvrir</a> |
| DRFM                 | Loi   | n°2/MCVDD /DC/SGM/DRFM | Loi portant nomination du directeur de l'informatique et du pré-archivage    | 2017-05-14                   | 2017-05-03         | 19                      | <a href="#">Ouvrir</a> |
| DRM                  | Décision-Loi                                    | n°2/MCVDD /DC/SGM/DRFM | Décision loi pour  | 2017-06-03                   | 2017-05-09         | 9                       | <a href="#">Ouvrir</a> |

FIGURE 4.1 – Interface d'accueil

## 2- Formulaire d'enregistrement d'un document

L'enregistrement d'un document passe par l'interface ci-dessous. Elle doit permettre d'enregistrer rapidement les indices d'un document dans la base, tout en permettant à l'archiviste de récupérer le document qui porte les indices enregistrés et de le sauvegarder dans l'application. Nous sommes ici devant une interface simple pour permettre à l'utilisateur de ne pas se perdre. Nous réutilisons également l'expérience utilisateur en mettant les éléments à remplir dans l'ordre auquel sont habitués les utilisateurs et en utilisant un système de formulaire classique pour permettre à l'utilisateur de rentrer les informations. A noter, la présence du bouton « Enregistrer » qui va permettre de sauvegarder les indices du document dans la base de données et aussi de sauvegarder le document en question dans l'application.

The screenshot shows the 'ARCHIVEGest' application interface. On the left is a dark sidebar with a 'Documents' menu and options for 'Nouveau' and 'Liste'. The main area is titled 'Enregistrement des documents'. It contains a form with the following fields: 'structure' (text input), 'Type' (dropdown menu with 'Loi' selected), 'Référence' (text input), 'Objet' (text input), 'Date d'enregistrement' (text input), and 'Date de production' (text input). Below the form is a blue button labeled 'Parcourir...' with the text 'Aucun fichier sélectionné.' next to it. At the bottom is a blue button labeled 'Enregistrer'. The top right corner of the interface has a 'Déconnexion' link.

FIGURE 4.2 – Interface d'enregistrement des documents

### Remarques issues des tests d'acceptabilité

Un test d'acceptabilité est un test visant à valider le fait que les caractéristiques d'un produit correspondent aux attentes des consommateurs. Il peut servir à mesurer les chances de réussite du lancement d'un produit et / ou à déterminer des voies d'amélioration du produit. Toutes les fonctionnalités actuelles de l'application qui sont terminées ont été testées. En effet le but de ces tests est de vérifier si les caractéristiques du produit correspondent aux attentes du consommateur, ainsi quelques remarques ont été apportées en vue d'améliorer l'application. Elles se présentent comme suit :

- Ajouter l'option de suppression et de modification des indices enregistrés pour un document.
- Etre en mesure de faire des recherches par année voire par mois.
- Prévoir une option pour permettre à l'administrateur de vérifier les enregistrements effectués et de pouvoir les corriger en cas d'erreur.
- L'application doit être en mesure d'être utilisée en mode hors connexion.
- Veiller au respect de l'ergonomie de l'application.

Actuellement, nous pouvons dire qu'une grande partie de nos objectifs est atteinte et aussi les différentes remarques issues des tests d'acceptabilité ont été prises en compte. Cependant, de façon spécifique, les fonctionnalités fonctionnelles sont les suivantes :

- enregistrer document.
- rechercher document.
- modifier document.
- ouvrir document.
- imprimer document.
- créer un utilisateur.
- ajouter un nouveau type de document.
- supprimer un document.
- ajouter la famille d'un document.



## Discussion

Une grande partie de notre stage a été consacrée aux recherches pour la réalisation de l'application. Nous avons eu également à assister aux travaux de maintenance matérielle, logicielle et de sertissage des câbles réseaux. Ce qui nous a permis d'améliorer nos compétences dans ces domaines. Cependant, puisque l'application que nous devons développer gèrera les archives, nous avons été un peu initiée à l'archivage au cours de ce stage pour acquérir les bases nous permettant de mettre l'application en place. En effet, la conception de l'application nous a permis d'améliorer les connaissances que nous avions et aussi d'en acquérir d'autres. Nous avons pu donc perfectionner ces connaissances dans l'utilisation du framework Laravel, en HTML et CSS. Ainsi, la réalisation de cette application a enrichi nos connaissances et a donc amélioré notre niveau en développement web.

En outre, parlant de ses atouts, l'application a été conçue en partant d'un procédé d'archivage simple qui permet une flexibilité dans la recherche des documents archivés.

En effet, l'avantage de cette application est qu'elle peut être adaptée. Même si les types de documents ainsi que les côtes attribuées aux documents ne sont pas identiques à ceux pris en compte dans l'application, l'administrateur peut créer les types de documents et les côtes dont-il a besoin dans la structure concernée. Ainsi, chaque structure qui adopterait cette application pourrait l'adapter aux types de documents qui sont produits dans ses différents services.

Le deuxième atout de ce logiciel est qu'il a été réalisé le plus simplement possible. Ainsi, toute personne peut l'utiliser aisément. A cet effet, des travaux futurs sont envisagés sur l'application dans l'intention de l'améliorer. Ainsi, puisque le support physique a toujours sa valeur et qu'on ne peut pas se fier entièrement à la gestion automatisée à cause de nombreuses difficultés qui pourraient survenir, il serait donc envisageable d'ajouter des fonctionnalités de gestion de la salle des archives. C'est-à-dire à travers l'application, être en mesure de connaître avec précision le rayon qui comporte un document donné, l'historique d'un rayon à savoir :

-La date de consultation du rayon, le motif de consultation, le nombre de boîte d'archive que

---

contient un rayon donné, et les détails sur l'archiviste qui l'a consulté. Pouvoir également à travers l'application, donner une statistique par types de document par année. Et pour améliorer ce qui est déjà fait, nous devons améliorer :

-la vérification de la validation, la robustesse de l'application, la montée en charge, l'harmonisation des labels et du menu, l'importation et l'exportation des informations de l'application, les zones de saisit des dates, l'organisation du code et enfin la gestion de l'option "mot de passe oublié" et "se rappeler du mot de passe".

# Conclusion

L'utilisation d'une application de gestion d'archives serait d'une importance non négociable pour la DIP du MAEP vu son rôle dans le suivi de ces archives. La DIP a pour mission d'assurer la sécurisation de ses archives. En effet notre passage dans ce service pour nos stages nous a permis de constater l'absence d'une application de gestion des archives, et cette absence conduit donc à une recherche difficile et lente des documents archivés. Ainsi donc pour remédier à ces problèmes, nous avons proposé de réaliser une application de gestion des archives pour le MAEP. Le point de départ de la réalisation de cette application est la collecte de l'information dans le but de faire l'étude de l'existant et de définir nos objectifs. Nous sommes passés au choix techniques qui nous a permis de sélectionner les outils qui ont servi pour la conception de notre application. Ensuite, nous nous sommes intéressées à l'analyse des besoins, à la conception qui nous a permis de distinguer les besoins de la DIP et les différents acteurs de notre application. Le dernier volet de notre document est la partie de présentation des résultats du travail où nous avons présenté quelques interfaces de notre application ainsi que les différentes fonctionnalités déjà fonctionnelles dans l'application. Cette dernière a été développée en tenant compte des besoins exprimés par la DIP. Elle assure plusieurs fonctions parmi lesquelles nous avons :

- l'enregistrement d'un document.
- la modification d'un document.
- la suppression d'un document.
- l'ouverture d'un document dans le navigateur.
- l'impression d'un document.

Le résultat de ce stage est donc une application de gestion des archives réalisée dans le respect de plusieurs règles de bonnes pratiques de programmation web, parmi lesquels on retrouve l'utilisation de framework ou encore l'utilisation d'un design simple et adapté au web.

# Bibliographie

- [1] David Chaplin, 2016 : *Scrum*,Eni,Open IT, 12 p,  
[http://www.eyrolles.com/Informatique/Theme/98/  
developpement-d-applications-modelisation-et-genie-logiciel-methodes-agile](http://www.eyrolles.com/Informatique/Theme/98/developpement-d-applications-modelisation-et-genie-logiciel-methodes-agile)  
consulté le 24/05/2017
- [2] Valérie Fernandez, Thomas Houy, Carine Khalil, 2013 : *Les méthodes agiles de développement informatique*,Presses des Mines - Transvalor,Vademecum, 86 p, [http://www.eyrolles.com/Informatique/Theme/98/  
developpement-d-applications-modelisation-et-genie-logiciel-methodes-agile](http://www.eyrolles.com/Informatique/Theme/98/developpement-d-applications-modelisation-et-genie-logiciel-methodes-agile)  
consulté le 24/05/2017
- [3] Fichier parlant du progiciel Ariane service v10 : [http://www.amig.fr/docs/Ariane\\_services\\_V10.pdf](http://www.amig.fr/docs/Ariane_services_V10.pdf) consulté le 20/04/2017
- [4] [http://lim.univ-reunion.fr/staff/fred/M2info/15-16/Stages/  
Rapports/NOEL\\_Loic\\_Rapport.pdf](http://lim.univ-reunion.fr/staff/fred/M2info/15-16/Stages/Rapports/NOEL_Loic_Rapport.pdf) consulté le 10/03/2017

# Webographie

- [5] Documentation et information sur laravel :  
<https://laravel.com> consulté le 10/03/2017
- [6] Site de la société AMIG <http://www.amig.fr/archives/fr/amig-accueil>  
consulté le 20/04/2017
- [7] Site du logiciel archimède <http://www.archimed.fr/solutions-pour-bibliotheque> consulté le 20/04/2017
- [8] Site du logiciel maarch <http://maarch.com> consulté le 26/05/2017
- [9] Site du logiciel Gapi <http://www.mobydoc.fr> consulté le 26/05/2017
- [10] Site pour en savoir plus sur la méthode UML <http://www.Développez.com> consulté le 26/05/2017

## Annexe 1

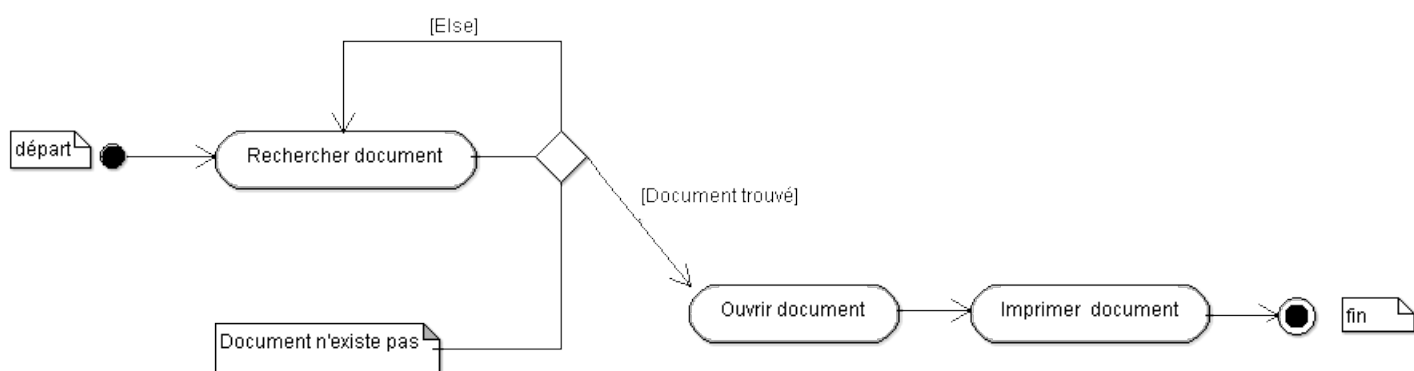


FIGURE A.1 – Diagramme montrant les activités à réaliser jusqu’à l’impression d’un document

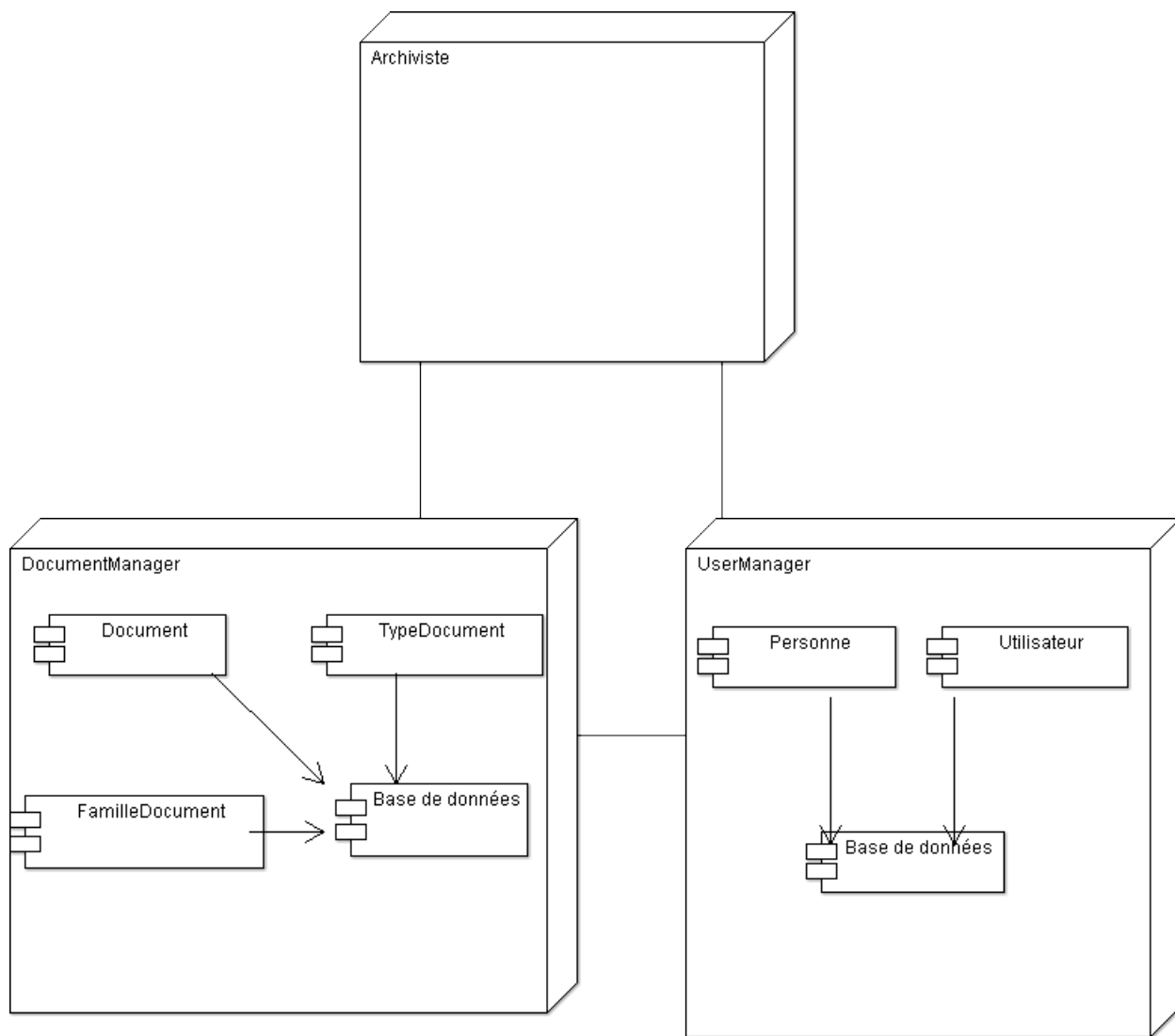
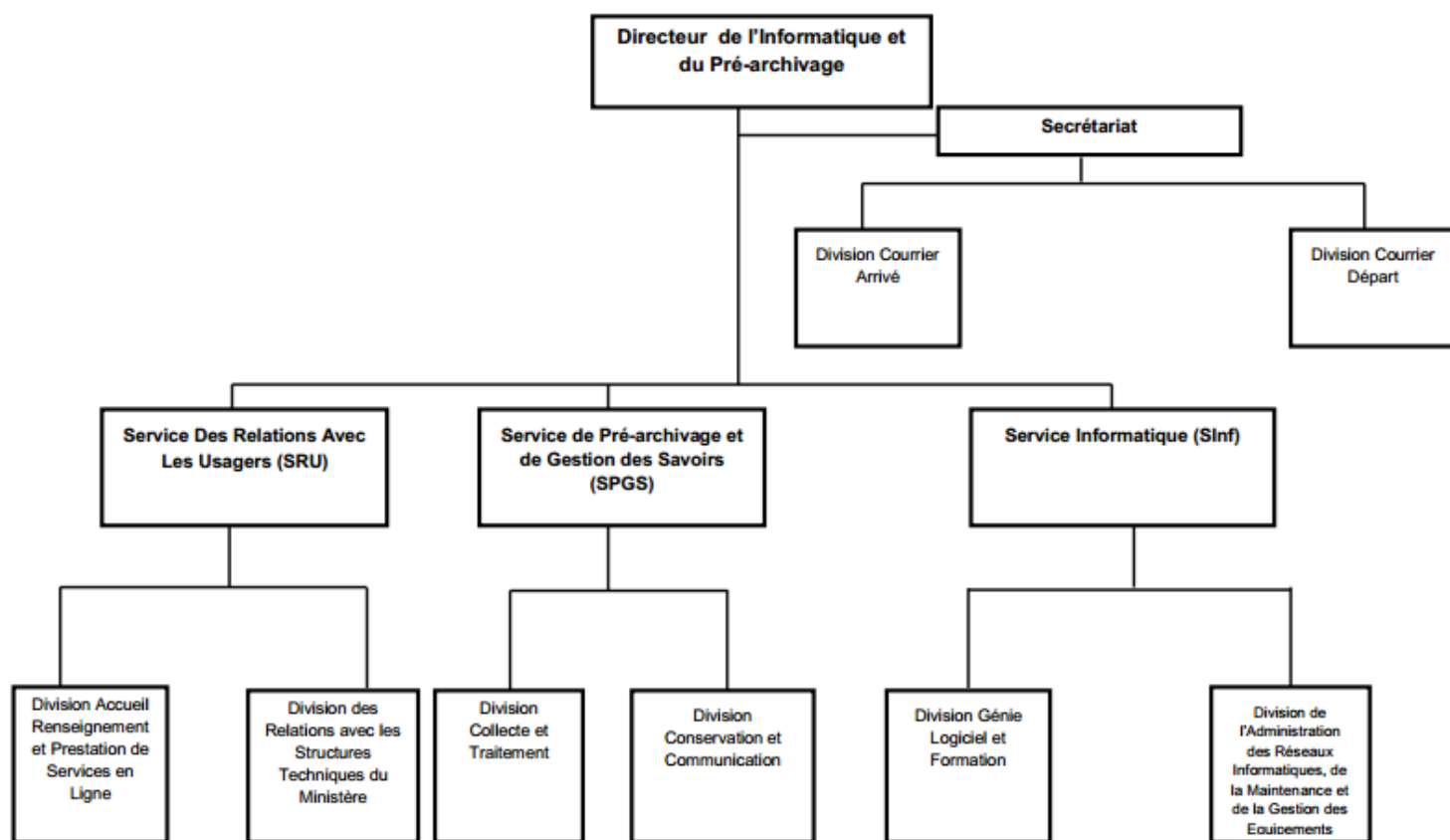


FIGURE A.2 – Diagramme de composants

**ORGANIGRAMME DE LA DIRECTION DE L'INFORMATIQUE ET DU PRÉ-ARCHIVAGE**





# Table des matières

|   |           |
|---|-----------|
| Dédicaces   | i         |
| Remerciements   | ii        |
| Glossaire   | iv        |
| Sigles  | v         |
| Résumé  | vi        |
| Abstract  | vii       |
| Introduction  | 1         |
| <b>1 Historique</b>                                   | <b>3</b>  |
| 1.1 Présentation de la structure d'accueil . . . . .  | 3         |
| 1.2 Etude de l'existant . . . . .                     | 4         |
| <b>2 Revue de littérature</b>                         | <b>5</b>  |
| 2.1 Etat de l'art . . . . .                           | 5         |
| 2.1.1 Présentation des solutions existantes . . . . . | 5         |
| 2.1.2 Comparaison des solutions existantes . . . . .  | 6         |
| 2.2 Interêt du sujet . . . . .                        | 7         |
| <b>3 Matériel et méthodes</b>                         | <b>8</b>  |
| 3.1 Matériel . . . . .                                | 8         |
| 3.2 Méthodes . . . . .                                | 11        |
| 3.2.1 Analyse des besoins . . . . .                   | 11        |
| 3.2.2 Modélisation . . . . .                          | 11        |
| <b>4 Résultats</b>                                    | <b>29</b> |
| <b>5 Discussion</b>                                   | <b>33</b> |
| <b>Conclusion</b>                                     | <b>35</b> |

---

|                        |           |
|------------------------|-----------|
| <b>Bibliographique</b> | <b>36</b> |
| <b>webographie</b>     | <b>37</b> |
| <b>A Annexe 1</b>      | <b>38</b> |

---