**République Algérienne Démocratique et Populaire**
**Université Abou Bakr Belkaid– Tlemcen**
**Faculté des Sciences**
**Département d'Informatique**

**Mémoire de fin d'études**
**Pour l'obtention du diplôme de Master en Informatique**

***Option:*** *Système Information (SI)*

# Thème

# Système d'application d'une base de données XML « XML news »

**Réalisé par :**

- **Amina BENYAHIA**

*Présenté le 23 Juin 2015 devant le jury composé de :*

- **Mr BENAMAR A** *(Président)*
- **Mme. HALFAOUI A** *(Encadreur)*
- **Mr.** SMAHI M I *(Examinateur)*
- **Mme. BENMANSSOR F** *(Examinateur)*

Année universitaire : 2014-2015

# *REMERCIEMENTS*

Nous tenons à remercier :

Allah le tout puissant.

Mme HALFAOUI AMAL, notre encadreur, pour ses conseils, sa disponibilité et son encouragement qui nous ont permis de réaliser ce travail dans les meilleures conditions.

Les jurys pour leurs efforts et leur soin apporté à notre travail:

Mr. BENAMAR Abdekrim

Mr. SMAHI M Ismail

Mme. BENMMANSOUR F

Aux enseignants de notre université et du département d'informatique et notre collègue de promotion.

# *DEDICACE*

*Je dédie ce mémoire :*

- *A Ceux qui ont fait de moi une femme : mes très chers parents : ma mère, ma mère, ma mère et à mon père, que dieu les récompense et les garde, et surtout ma chère mère qui n'a jamais cessé de m'aider et qui m'a encouragé et soutenue toute au long de mon cursus.*
- *A mes chers enfants* ***Abdel Hadi*** *et* ***Abdellah el Karim.***
- *A la mémoire de Mon cher oncle* ***Benyahia Chaîb*** *qui a voulu assister à ma soutenance mais le destin était plus sévère.*
- *A mes frère* ***Sid ahmed, Abdesselam,Belahcene, Nesrine et Moussa.***
- *A toute la famille* ***Benyahia et Berradia.***
- *Spécial dédicace a mon cousin monsieur* ***Ghermoudi Mohammed el khamiss*** *et monsieur* ***Benamar Abdelkrim*** *pour son soutien pendant tous mon parcours.*
- *A tous mes professeurs de tout mon cursus.*
- *A tous mes voisins et mes amis*

*Benyahia Amina*

## Résumé

XML est devenu très utilisé autant par les SGBD que par les outils de bureautique et de documentation, par les logiciels de gestion aussi bien que par les applications techniques et scientifiques.

Dans ce PFE nous avons mis en place un site web de gestion d'une base de données XML afin de réaliser différentes opérations de gestion.

La modélisation du système XML NEWS a été faite par UML, en s'appuyant sur le processus UP. La base de données et l'application a été crées par le SGBD eXist.

## ملخص

أصبح LMX يستعمل كثير ا من قبل أنظمة إدارة قواعد البيانا ت أكثر من الأدوات المكتبية والتوثيق، و من إدارة البرامج أكثر من التطبيقات التقنية والعلمية.

في هذا العمل لقد قمنا بانشاء موقع الكتروني لتسيير قاعدة بيانات لملفات من نوع للقيام بعدة عمليات تسييرية .

تصميم النظام MLX swen كان باستعمال LMU بالتركيز على عملية UP، القاعدة البيانية صنعت باستعمال نظام إدارة قواعد البيانات eXist.

## Abstract

XML has become very much used by the database management system then office and documentation tools, also by management software much than technical and scientific software.

In this work we had establish an web site for the management of XML data base in order to perform different management operation.

The modeling of XML NEWS system was done by using UML, focusing on UP process.

The data base and the software was created by the eXist.

# Table de matière

## Liste des illustrations

## Glossaire

XML : Extensible MarkupLanguage

XPATH : Xmlpath langage

Xquery : xmlquery langage

SGBD :  système de gestion de base de donnée

SGBD-R : syteme de gestion de base de données relationnel

W3C :World Wide Web Consortium

HTML :Hypertext Markup Language

XSLT :eXtensibleStylesheet Language Transformations

DTD :Document Type Definition

RTF :Rich Text Format

PDF : Portable Document Format

SGML :Standard Generalized Markup Language

Xdiff: diff pour xml

XSL-FO :eXtensibleStylesheet Language - Formatting Objects

ODBC : Open Database Connectivity

JDBC :Java Database Connectivity

SQL :Structured Query Language

API :Application Programming Interface

UML :Unified Modeling Language

Up : unified process

CSS : Cascading Style Sheets

# Introduction générale

## Contexte

XML est l'acronyme de eXtendedMarkupLanguage, il est devenu l'un des langages les plus utilisé en informatique grâce à sa flexibilité, simplicité, extensibilité et lisibilité, en XML le contenu est lisible nous n'avons pas besoin de connaissance théorique pour comprendre le contenu du document.
Le domaine d'utilisation est très vaste, il peut y aller de données géographique jusqu'aux utilisations web.

Le domaine de stockage et d'interrogation des documents XML forme une filière de recherche qui connait un mouvement de développement explosif. A nos jours XML est devenu un standard de description de données. Aux niveaux des systèmes informatiques, il offre des possibilités importantes de développement, pour ça il est indispensable de stocker des données au format XML pour la persistance des données informatiques.

Ils existent plusieurs outils qui offrent la possibilité d'interroger des documents XML, d'autre sont en cours de normalisation.

C'est les systèmes de gestion de base de données qui sont chargés de la création, l'utilisation et la maintenance des bases de données relationnelles.

Généralement c'est dans des bases de données relationnelles que nous stockons les documents XML sou forme de tableau ou relation.

Il est primordial d'utiliser des bases de données XML native pour le stockage dans le cas de documents XML structurés, qui utilisent des langages de requêtes XML comme Xpath ou Xquery.

## Problématique

La publication des documents contenant des textes tel que les livres, les journaux est impossible si nous utilisons des tables pour le stockage des données telle que : un texte, un article ou bien un paragraphe d'un livre…..etc, la solution est l'utilisation d'un langage de description des données tel que le XML.
Les bases de données sont des outils de stockage, en générale la plupart des systèmes utilisent les bases de données avec des tables pour modifier, stocker et utiliser leurs informations.

Dans notre projet les bases de données avec les tables (relationnelle)ne conviennent pas avec nos besoins, car nous sommes entrains de traiter des textes (journaux, livres, revus … etc).

## Contribution

Nous avons proposé dans notre travail d'utiliser un de système de gestion de base de données XML native qui offre la possibilité de traiter les données structuré.

notre travail a pour objectifs la mise en place d'un site web de gestion d'une base de données XML afin de réaliser différentes opérations d'interrogations, d'affichage et de modification sur notre base de données ,nous avons choisi le SGBD eXist pour sa simplicité , sa capacité de stockage des données structurés et encore la facilité de recherche et de contrôle des données.

## Organisation du rapport

Notre mémoire est organisé comme suit : Le premier chapitre est consacré à présenter le langage XML, le principe générale de stockage de documents XML et aussi à détailler les différentes bases de données XML .le deuxième chapitre présente la modélisation de notre système et de la base de données, Et nous avons détaillé l'implémentation du système dans le troisième chapitre.

# Chapitre 1 : Langage XML et les bases de données XML

# 1. Introduction :

Dans ce chapitre on va présenter le langage XML en générale. Vue son importance on va se baser sur le stockage des documents XML, ensuite on va passer à la présentation des différents types de bases de données, en commençant avec les bases de données relationnelles qui sont étendus avec des outils de stockage et de traitement des documents XML.

Par la suite on va entamer la notion des bases de données natives, qui sont conçues spécifiquement pour le langage XML ; pour le stockage et l'accès à des arbres de données des documents structurés. On donnera une explication claire et simple de l'outil eXist qui est recommandé par le W3C (organisation de normalisation, qui est chargé de promouvoir la compatibilité des technologies du web).

# 2. Présentation de langage XML :

## 2.1Définition :

XML se base sur la description des données sue le concept les balises. Un fichier XML est un ensemble de balises (élément de base) qui sont organisé et arrangés dans une arborescence hiérarchique, le contenu des balises peut être du texte, d'autre élément ou une combinaison de texte et d'élément (balise) [1] [2].

Les éléments peuvent posséder de un ou plusieurs attributs servant à donner un sens au élément, ajouter des informations relative au contenu de l'élément. L'auto extensibilité est le plus grand point fort de ce métalangage, si nous faisons une comparaison avec HTML qui lui-même repose sur le principe des balises, propose un nombre finis de balises dans chaque fichier à un usage arrêté. XML est considéré comme une sorte de grammaire qui permet au usager de coder un document en définissant ses propre balises, le plus important et l'essentiel que la syntaxe soit respectée ce qui conduit a dire que le document est bien formé[3] .

## 2.2. Historique:

La création de langage XML à l'objectifs de répondre à un besoin très précis : l'échange de données**.**

Le développement de ce langage a débuté en 1996, et il est réellement « né » avec la spécification du W3C1 en 1998. Le large succès rencontré par ce langage, chez les informaticiens d'abord mais aussi très vite chez les professionnels de la documentation et les spécialistes de sciences humaines, a accompagné le développement de l'édition électronique,

à tel point qu'XML semble aujourd'hui s'imposer comme le format évident pour qui envisage de publier en ligne une édition de texte(s) ou un corpus de documents [4] [5].

## 2.3. Structure du document XML

Un document XML est constitué d'un ensemble d'éléments et d'attributs : nous considérons l'exemple d'un document XML simple, qui est illustré dans la figure suivante :

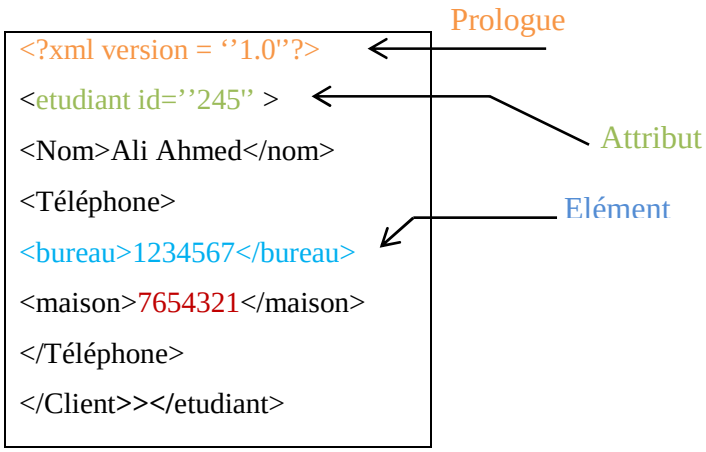



**Figure1.1 : Document XML**

## 2.4. Caractéristiques des documents XML :

- les objets XML nécessitent moins de modifications (que les enregistrements des tables).
- consulter XML signifie avoir soit le contenu complet, soit une partie du document présenter un résultat : forme textuelle ou autre (XSLT) [6].
- les documents XML peuvent être partitionnés de diverse manière en collections.
- un document XML peuvent être validé par l'utilisation de DTD XML XSchema [6].

## 2.5. Document Orienté données :

Les documents Orientés données utilisent le XML comme vecteur de données. Ils sont conçus pour être exploités par une machine et le fait que XML soit utilisé est généralement accessoire [7], L'exemple suivant présente un document XML orientés données :

<Vols>

<compagnie> Air Algérie </compagnie>

<départ> Constantine </départ>

<Vol>
<destination> Alger</destination>
<heuredepart>7.30</heure depart>
<heure arrive>8.30</heure arrive>
</Vol>
<Vol>
<destination> Alger</destination>
<heuredepart>14.00</heure depart>
<heure arrive>15.00</heure arrive>
</Vol>
</Vols>

## 2.6. Document Orienté document:

Documents conçus pour être utilisés par des humains. Ils sont ordinairement écrits manuellement en XML ou sous d'autres formats tels que RTF, PDF ou SGML, puis ils sont convertis en XML [7].

L'exemple suivant présente un document XML orientés document :

<Document >
<Titre>
DBXML
</Titre>
<Soustitre>
</Soustitre>
<Introduction> une base de données et une collection d'informations
regroupées de telle manière que l'accès et la manipulation de ces données
soient souples et rapides
</Introduction><body> actuellement les différents types de base de données
</body>
<Conclusion></conclusion>
</Document>

## 2.7. Les fonctionnalités d'un SGBD supportant XML :

- la définition des bases de données et des "objets" contenus : documents, collections, procédures [8].

- la manipulation des données :
  - ✓ validation comparaison entre documents : Xdiff.
  - ✓ interrogation : langages de type XPath, Xquery.
  - ✓ traitements de type XSLT, transformation XSL-FO, etc.
  - ✓ modification : insertion, suppression, mise à jour.
  - ✓ recherche d'information.

- la confidentialité : les utilisateurs définissent des droits d'accès à des collections, mais pas à des éléments de documents [8].

- l'intégrité des données : pas plus que ID et IDREF à l'intérieur d'un même document [8].

## 2.8. Les objectifs du XML :

- **Objectif 1** : Le XML permet la séparation des données et de leur mise en forme.

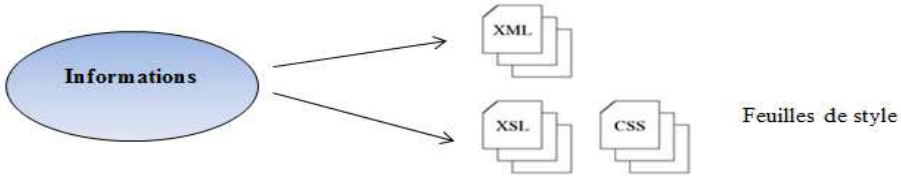



**Figure 1.2: La séparation des données XML** [9]

- **Objectif 3 :** Séparation des informations structurelles et des données.

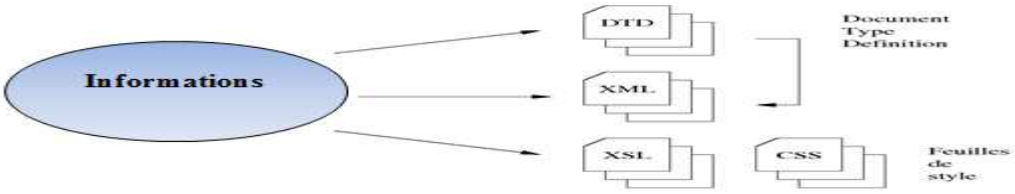



**Figure 1.3 : La séparation des informations structurelles et des données.**[9]

- **Objectif 4 :** Le XML permet de vérifier un document en utilisant le DTD :


**Figure 1.4 : La vérification des documents**[9]

- **Objectif 5 :** XML assurer la portabilité des données entre des environnements hétérogènes.


**Figure 1.5 : La Communication entre deux machines** [9]

- **Objectif 6 :** Production de documents à la demande.

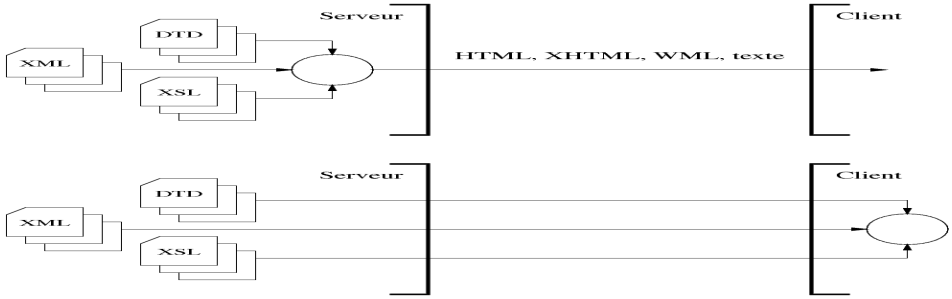
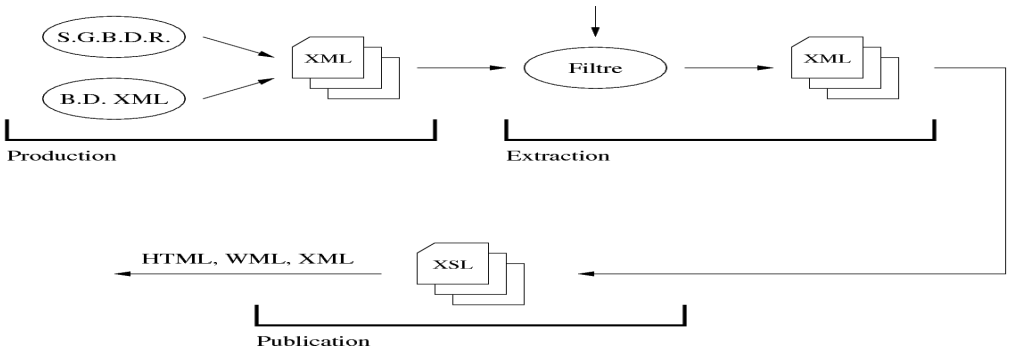



**Figure 1.6 : La Production des documents XML** [9]

- **Objectif 7 :** Intégration de plusieurs sources de données dans une chaine de traitement XML




**Figures 1.7 : L'intégration des données XML** [9]

## 3. Stockage des documents XML :

L'utilisation croissante du langage XML dans de nombreux domaines fait naître de nouveaux besoins en matière de stockage et de gestion de données.

Dans cette partie, je vais expliquer les deux grandes catégories de bases de données intégrant du XML : d'un côté l'extension des bases de données relationnelles "classiques" et de l'autre des bases de données dédiées à XML, dites XML natives [10].

Il existe deux  solutions pour mémoriser les données XML :

- les données sont transformées et stocker avec un SGBD relationnel.
- les données sont intégrées directement en XML.

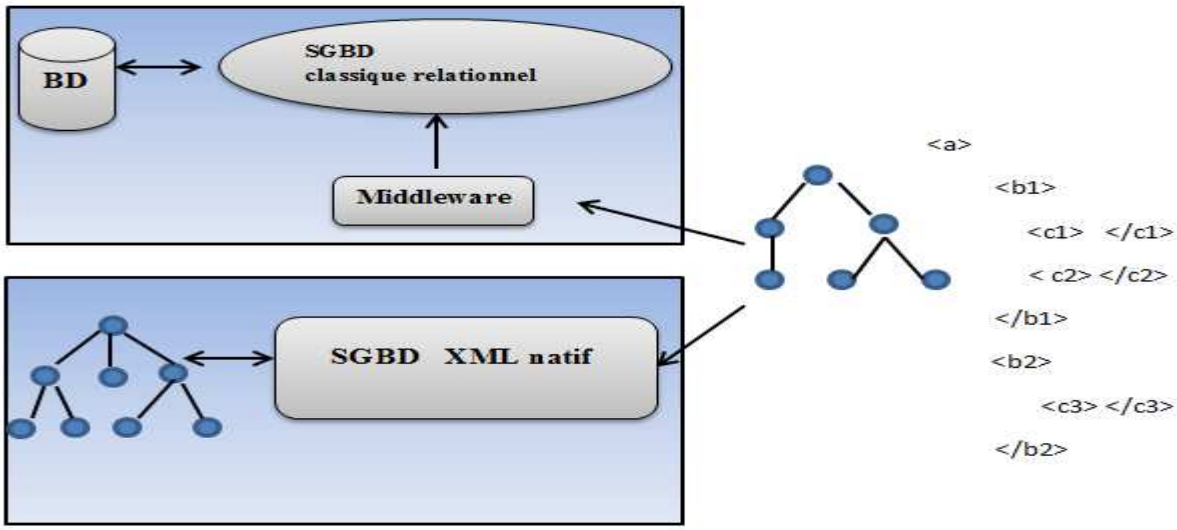



**Figure 1.8 : Les méthodes de mémorisation des données XML** [10]

Les critères de choix de la base de données dépend de l'usage des documents que nous voulons les stocker il faut donc déterminer le type du document XML à traiter, nous pouvons choisir les bases de données relationnelle ou les bases de données objet si les données contenues dans le document XML est importantes, sinon nous choisissons les bases de données en XML natif si la structure du document XML et aussi importante.

Aussi qu'il y a d'autre critères de choix tel que :

- Le temps d'exécution de requêtes.
- La complexité des mises-à-jour.
- La taille des données stockées Intégration avec applications existantes.

## 3.1. Les bases de données relationnelles et le XML:

Les bases de données relationnelles servent à organiser les éléments d'information (fichiers, textes, images, etc.) au sein d'une structure de tables liées entre-elles. Une galaxie de contenus structurés qu'ils rendent accessibles depuis une application cliente par le biais de langages de requêtes - exécutées via des interfaces (telles que ODBC et JDBC). Pour l'heure, le vocabulaire le plus utilisé pour l'interrogation de bases de données demeure le langage (SQL).

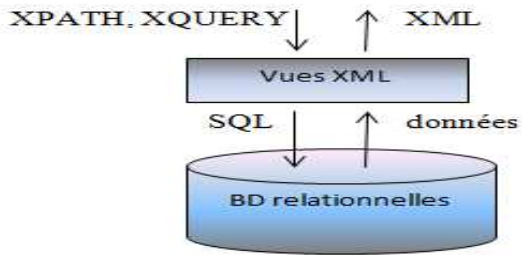



**Figure 1.9 : Le stockage de document XML dans une base de données relationnelle** [7]

### 3.1.1. Utilisation d'une base de données relationnelle "pure"

La mémorisation des flux XML dans une base de données relationnelle est habituellement fait par l'utilisation d'une interface (middleware) qui joue l'intermédiaire entre données ou requêtes XML et la base de données au modèle relationnel. Les données sont transformées en t-uples et les requêtes en SQL [10].

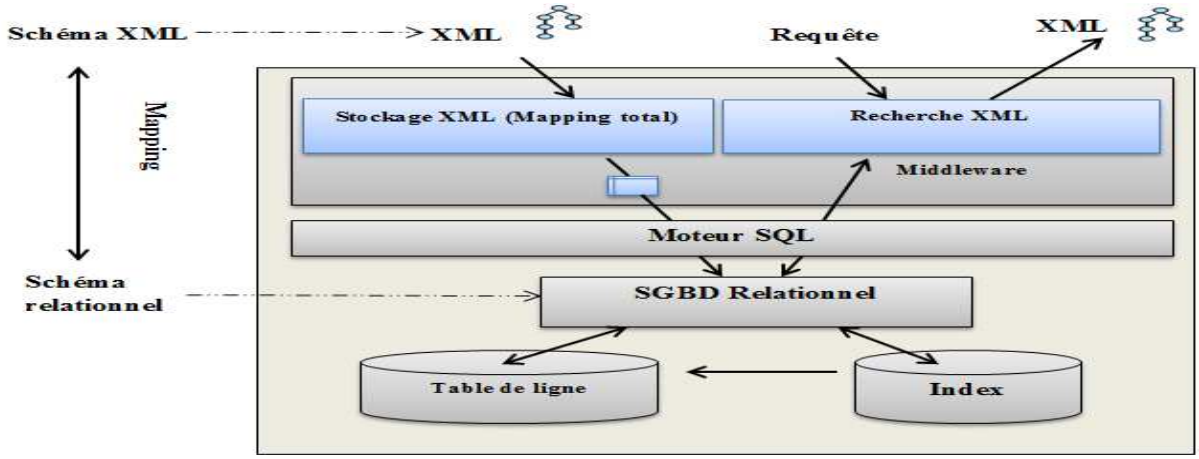



**Figure 1.10 : Base de données relationnelle** [10]

### 3.1.2. Utilisation d'une base de données relationnelle possédant un type XML.

Concrètement, le mapping complet de données XML n'est pas nécessaire, car la plupart des SGBD ont intégré des types XML, voire même des fonctions pour les manipuler (XPath ou autres). L'idée principale est donc de ne transformer qu'une partie des données, en laissant le reste en XML. On a ainsi un mapping partiel [10].

La plupart des SGBD-R proposent des outils pour manipuler XML de manière plus ou moins poussée. Généralement, ce type de base de données stocke et indexe les fichiers XML. Parmi ce type, on peut trouver : Interbase, MySQL, MaxDB, PostgreSQL (supporte aussi des fonctions XPath), Sybase ASE (supporte des requêtes au format XQL) [10].

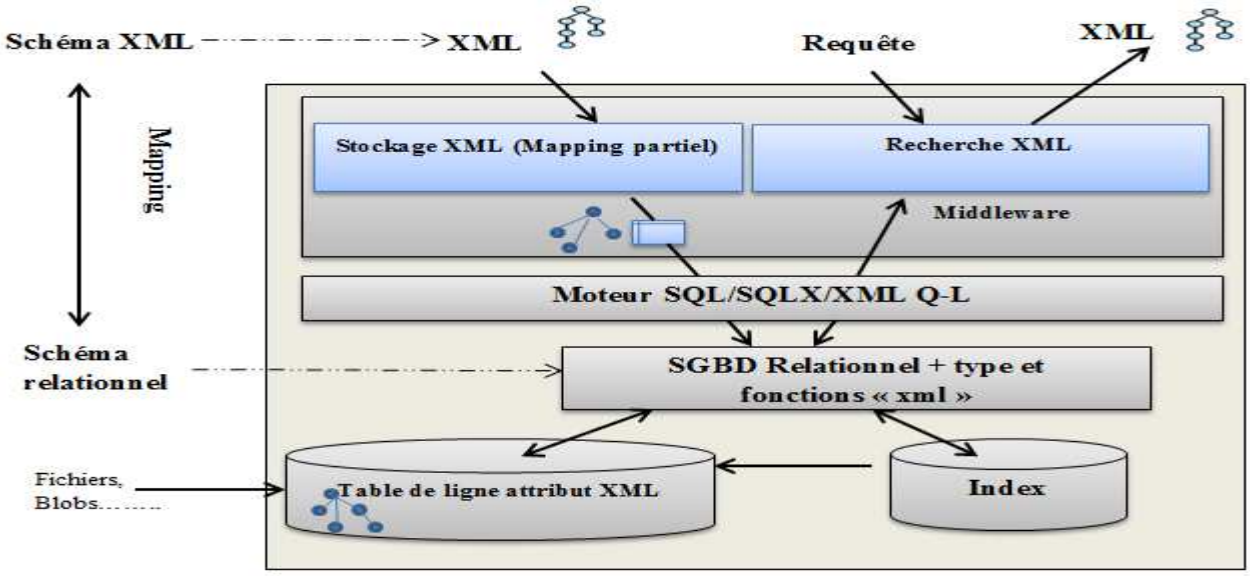



**Figure 1.11:Architecture de la Base de données relationnelle possédant un type XML**

[10]

### 3.1.3. Le mapping Relationnel/XML

Le mapping relationnel/XML permet l'importation de XML vers des tables et l'exportation du relationnel vers XML Il faut prendre en compte les contraintes du SGBD tel que le nombre de tables et d'attributs par table la taille des attributs [10] [11].

Le mapping relationnel/XML n'est pas toujours simple à mettre en œuvre. Cela dépend beaucoup des contraintes applicatives et des modèles de données de chaque côté [10] [11].

### 3.1.4. La mise en œuvre des transformations dans les deux sens.

L'opération de transformation dépend fortement des deux schémas. La traduction, parfois lourde, s'effectue en deux étapes : une série de transformations du modèle XML (en utilisant XSLT, XQuery) puis une transformation entre XML et le modèle relationnel. Pour cette dernière étape, des API spécialisées existent comme XSU d'Oracle. Du fait de la nature hiérarchique de XML. Le mapping vers une base relationnelle/objet est plus intéressante. Ceci dit, le modèle relationnel étant moins structuré, les transformations ne sont pas toujours bijectives, autrement dit, soit X un arbre XML, et T la transformation vers le modèle relationnel (peut-être elle-même décomposée en une suite de transformations élémentaires) et $T^{-1}$ l'opération inverse, alors, dans le cas général, $T^{-1}(T(X)) \neq X$, sauf à inclure beaucoup d'informations uniquement structurelles dans le modèle relationnel [10].

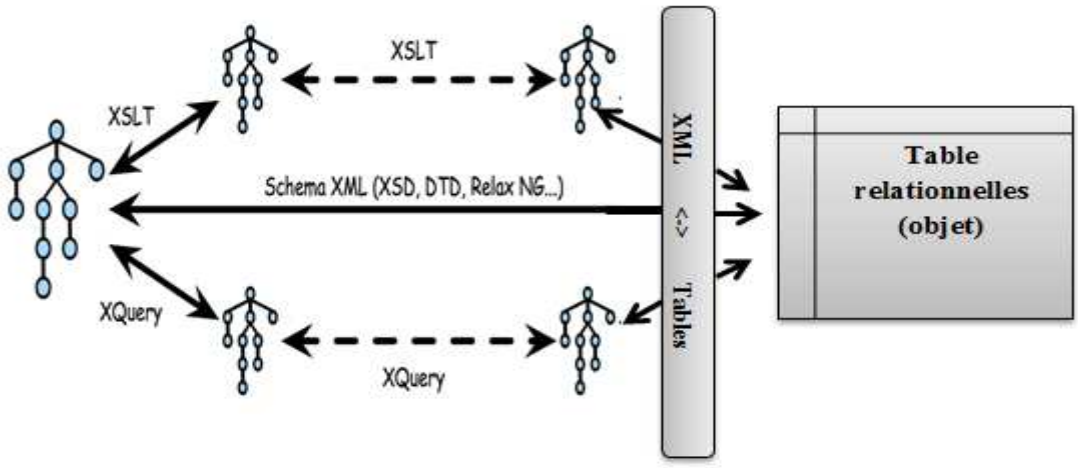



**Figure 1.12 : La transformation des données XML dans les deux sens** [10]

### 3.1.5. Avantages et inconvénients de bases de données relationnelles :

L'utilisation de bases de données relationnelles a pour avantages principaux l'utilisation de SGBD communs et des performances connue cependant il présente quelques inconvénients pour stocker des documents XML comme :

- Avec la mise en place du mapping, il y a une perte d'informations notable nécessitant de nombreux contrôles par le middleware.
- Ie SGBD relationnel ; Le langage de requête/modification basé sur SQL est un peu lourd, difficulté de manipuler des modèles différents comme du XML.

Des SGBD sont apparus traitent directement des données XML, sans transformation de modèle : les SGBD XML natifs, appelés aussi "SGBD-XML".

## 3.2. Les bases de données XML native

Les bases de données conçus pour le stockage des documents XML sont connu sous le nom de base de données native. Nous pouvons distinguer les base donnée natives des autre avec le faite que une vision logique et facile e compatibilité avec le modèle XML (organisation arborescence et hiérarchique des données), tout comme le fait que les bases de données relationnelles offre une vision logiques des informations adéquate avec le modèle relationnel(les données sont organisées sous forme de table).

La vision logique adéquate avec le modèle XML permet d'une manière facile l'apparition d'utilisation des standards reliés à XML tels que : XQuery, XPath, XSLT, pour la manipulation des informations stocker dans une base de données native.

L'unité fondamentale du stockage (logique) dans une base de données XML native est le document XML, tout comme une ligne d'une table constitue l'unité fondamentale du stockage (logique) dans une base relationnelle.

Les base de données XML native sont faite pour le besoin de plus de fonctionnalités et Pour le stockage des contenus plutôt orientés documents et des documents dont le format naturel est XML .

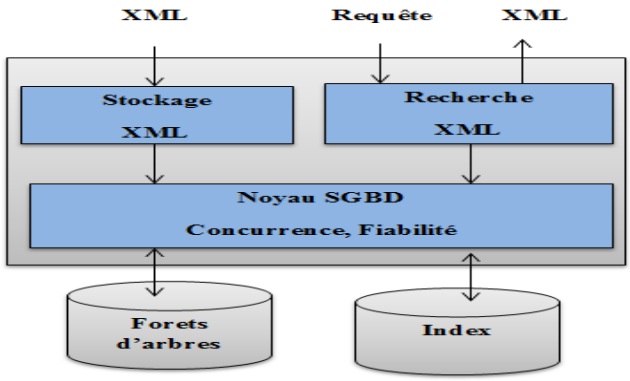



**Figure 1.13 : Architecture des bases de données XML native [7]**

### 3.2.1. Caractéristiques des BDD XML Natives

- **Présentation**

Il s'agit de la mise en forme des données stockées dans la base, les bases de données XML présentent tout leur intérêt : les résultats de requêtes sont directement des noeudsXML .

Le mode de fonctionnement en mode application web adopté par les bases eXist rend très aisé le branchement de ces bases sur des applications de publications

- **Les collections de document**:

Une collection dans les bases de données XML native est l'équivalant d'une table dans un système relationnel.

- **Les langages de requête**

Les langages de requête comme Xquery, Xpath…sont indispensables pour toutes les bases de données XML native.

- **Les mises à jour et effacements:**

Une grande diversité de stratégie pour réaliser les mises à jour dans en fonction des bases, chaque produit possède son langage, le plus utilisé c'est XUpdate du XML:DB Initiative.

Les bases de données XML permettent de faire cela à deux niveaux :

• au niveau du document, en supprimant l'ancien et insérant le nouveau ;

• au niveau de la granularité du modèle XML (mise à jour de noeuds, d'attributs...) à l'aide du langage XUpdate pour les bases qui les implémentent [12].

- **Verrouillage et accès concurrentiels**

La plupart des bases de données supportent les transactions (commit + rollback). Problème de verrouillage qui se fait généralement au document.

- **Echanges « Round-Tripping »**

Le Round-Tripping permet de récupérer le même document que nous l'avons stocké dans la base, et aussi de récupérer de l'ordre des éléments, et des commentaires.

Les bases de données XML natives assurent en théorie de meilleurs résultats qu'un stockage des données dans des tables, puisqu'elles sauvegardent des éléments spécifiques au modèle XML .D'autre part, la restitution de l'information sous forme de document depuis une base XML native est immédiate alors qu'elle nécessite une étape d'interrogation (jointures entre tables) et de mise en forme des résultats depuis une base de données relationnelle[12] .

- **Indexation**

La définition d'index sur les données XML est nécessaire pour satisfaire les besoins énoncés pour l'interrogation des métadonnées dans des délais raisonnables face à une montée en charge de la base :

Les requêtes sémantiques sur des mots-clefs nécessitent la définition d'index textuels sur certains champs des métadonnées. Les bases de données XML comme les SGBD Relationnels permettent de définir des index. Si cela peut se faire de manière standard pour les SGBDsavec des commandes SQL, la définition d'index pour des bases de données XML native reste propriétaire de la solution utilisée [12].

### 3.2.2. Avantages des bases de données XML native

L'avantage de l'utilisation des bases de données XML réside dans le fait qu'elles manipulent directement le format XML.

Les avantages des bases de données XML native peuvent être cernés dans les points suivants :

- **Tout est stocké dans le même endroit** : L'avantage le plus important d'une base de données XML native est qu'elle stocke tous les documents dans un seul endroit facile a contrôlé et a recherché.

- **Vues multiples des mêmes données** : le stockage des données dans une base de données XML native permet des vues multiples d'un même contenu.

- **Exécution** :

  Les requêtes sur une base de données XML natives sont simples et plus rapides que les requêtes sur les documents stockés dans un système de fichiers, ceci est pour plusieurs raisons. la base de données permet l'indexation pour accélérer la recherche. La deuxième raison est que les documents stockés dans la base de données sont pré-analysés.

  Les bases de données XML utilisent des techniques d'optimisation du temps de l'exécution, comme la réécriture des requêtes.

- **Documents très grands :** Les bases de données XML natives sont caractérisées par leurs capacités de traiter de grands documents. En utilisant des outils comme XQuery, XPath.

- **Aucun bit n'est perdu :** Les bases de données XML natives peuvent retrouver le document original. Cette fonctionnalité est critique dans certaines situations légales où il est nécessaire de reproduire le document original jusqu'au dernier byte. Cette fonctionnalité peut également être importante dans le développement des logiciels, en particulier dans l'optimisation de l'exécution.

### 3.2.3 Inconvénients des bases de données XML native

Le manque de standardisation est l'inconvénient major des bases de données XML native pour cela plusieurs solution impliquera l'utilisation de technologies propriétaires pour sa mise en œuvre.

### 3.3 Exemple de bases de données relationnelles et natives :

| **Les bases de données relationnelles** | |
|---|---|
| **DB2 (IBM)** | En vue d'accueillir des documents XML, DB2 a été complété d'un nouveau module : XML Extender. Concrètement, ce dispositif a pour but de traduire un document au format XML pour intégrer les données qu'il contient au sein de la structure relationnelle de la base. |
| **Oracle 9i (Oracle)** | Oracle 9i s'adosse à une fonction appelée XML Data base Support (XDS) en vue d'intégrer directement des contenus XML dans ses colonnes. Une technologie qui lui permet aussi de décrire sa structure relationnelle au format XML et, à l'inverse, des données XML sous forme relationnelle. |
| **SQL Server (Microsoft)** | Annoncée pour début 2003, la prochaine version de l'outil de Microsoft devrait supporter les données XML de manière native tout en assurant la définition de tables dans le même format. |
| **Sybase ASE(Sybase)** | La version actuelle de Sybase ASE stocke et indexe les fichiers XML au sein de sa structure. Ce qui lui permet ensuite de supporter les requêtes au format XQL lancées sur cette catégorie de contenu. |
| **Les bases de données XML natives** | |
| **Tamino (Software AG)** | Tamino XML Server couple une base de données XML à une base de données relationnelle (ODBC et JDBC). Une structure qui lui permet de constituer des documents XML bâtis à partir de sa propre structure relationnelle, mais également d'autres sources (systèmes de fichiers, etc.) |
| **Ipedo XML Database (IPedo)** | Ipedo XML Database a été conçue pour prendre en charge les documents XML au sein d'une structure dans le même format. Pour ce faire, elle propose des mécanismes d'indexation et de classement -qui s'appuient sur un système de catégorisation XML (couplant DTD et XML Schema). |
| **TextML (Ixiasoft)** | Tout comme Ipedo, la solution d'IXiasoft accueille des documents XML. L'outil s'appuie sur le "parser" d'IBM pour indexer l'ensemble des attributs XML en question sous forme de meta-données (auteur, titre, etc.). Un méthode dessinée pour faciliter les développements clients (syndication, recherche, etc.). |

**Tableau 1.14 : Bases de données XML du marché [13]**

## 3.4. Exemple de BDD XML Natives eXist

### 3.4.1. Définition

C'est un système open-source sous licence GNU LGPL qui existe depuis quelques années (Wolfgang Meier en 2000) qui est actif et souvent utilisé. Il est utilisable sur toutes les plates-formes courantes (Linux, Mac OS ou Windows). Il exploite de nombreux standard tels que XQuery, XSLT, XPath, XUpdate, etc. il supporte l'accès concurrent et optimise l'accès aux données par une indexation automatique des données [14].

### 3.4.2 Application Web

Les bases de données XML sont parmi les base NoSQL celles qui supportent la plus grande hétérogénéité de formats de stockage et d'interrogation des données.de plus elles s'intègrent parfaitement avec les standards web du W3C et offrent une pile complète de développement d'application web. En effet, eXist propose un serveur d'application « Jetty »qui permet d'utiliser la base de données XML par l'intermédiaire d'un ensemble de services Web, selon des protocoles différents.

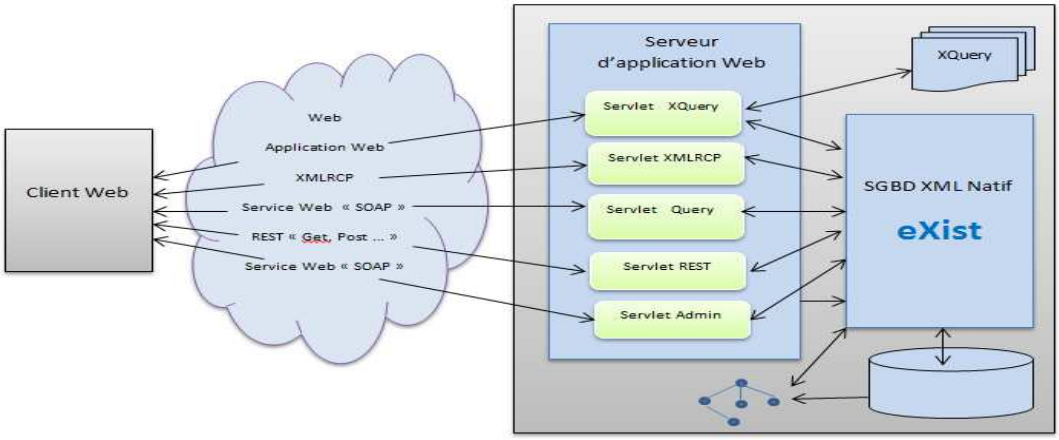



**Figure 1.15 : Architecture d'application Web [14]**

### 3.4.3. Application Web en XQuery

eXist donne la possibilité de créer des applications Web sur la base de XQuery. En effet, le serveur J2EE, couplé à une extension de XQuery permet de gérer aussi bien les requêtes HTML (GET, POST…) que les sessions [14]. Plusieurs modules permettent d'étendre la fonction standard de XQuery :

- Request Module (http://exist-db.org/xquery/request) &Response Module (http://exist-db.org/xquery/response) : pour gérer les requêtes HTTP (entrantes et sortantes) en proposant des fonctions pour gérer les cookies, les entêtes, etc[14].

- Session Module (http://exist-db.org/xquery/session) : pour traiter les sessions d'utilisateur [14].

- HTTTClient module (http://exist-db.org/xquery/httpclient) : pour effectuer des requêtes HTTP sur d'autres serveurs [14].

- XML:DB Module (http://exist-db.org/xquery/xmldb) : pour effectuer des modifications de la base (gère XUpdate et propose aussi des fonctions très proches du XQuery Update Facility) et gérer les droits d'accès [14].

### 3.4.4. REST

Il existe des API sont disponibles pour exploiter les données dans un SGBD-XML eXist. Parmi celles-ci, l'API REST qui permet de passer des requêtes à eXist. [14]

Plusieurs modes sont disponibles : GET, POST, PUT, etc.

- Pour le mode GET, il faut positionner quelques variables dans l'URL étendue. Ces variables sont nombreuses. On trouve, par exemple :

  - "_query" : la requête XQuery (ou XPath évidemment).
  - "_indent" : "yes" (par défaut) ou "no" pour la présentation.
  - "_encoding" : l'encodge attendu du résultat.

Pour le mode POST, les informations envoyées vers le serveur sont sous la forme :

soit d'un document XUpdate dans l'espace de noms "http://www.xmldb.org/xupdate" (pour des mises à Jours de la base), soit d'un document XML dans l'espace de nom "http://exist.sourceforge.net/NS/exist" [14].

### 3.4.5 SOAP

Soap est un protocole des services web pour accéder à exist, il propose une description de service en WSDL à l'url "http://localhost:8080/exist/services/Query?wsdl". [14]

Il existe des bibliothèques pour interroger eXist en SOAP, comme par exemple CelmaPheXistd'Oscar. Cette dernière propose une classe PHP "eXist" dont les principales méthodes et variables sont [14]:

- function__construct($user=”guest”, $password=”guest”,$wsdl=”http://localhost:8080/exist/services/Query?wsdl”)
- functionconnect () / functiondisconnect ()
- functionxquery ($query)…
- avec des entrées :
    - HITS : la longueur de la séquence retournée.
    - COLLECTIONS : la ou les collections concernées.
    - QUERY_TIME : le temps de traitement.
    - XML : tableau donnant le résultat de la requite.

### 3.4.6. Service Java

eXist peut être interrogée plus directement par une application en utilisant XML. En utilisant cette approche, eXist propose un client pour administrer les données[15].

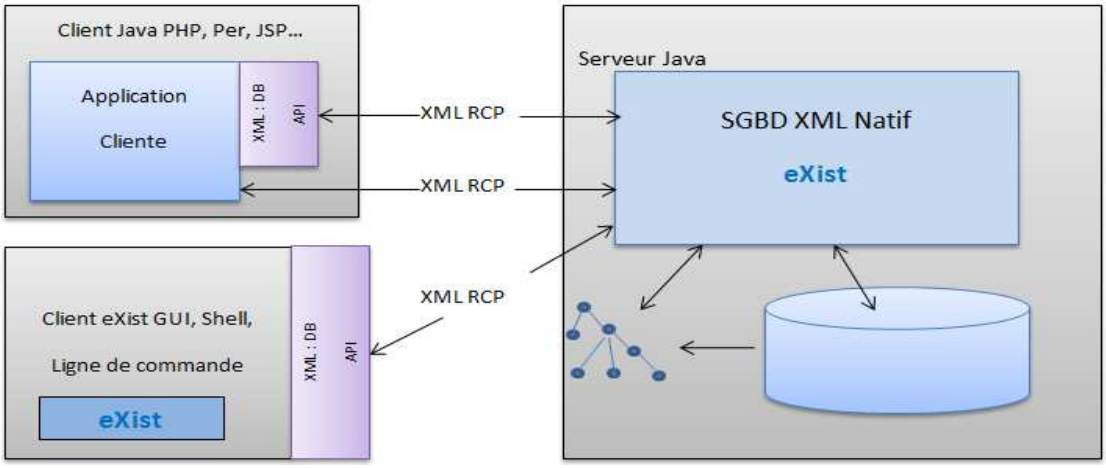



**Figure 1.16 : Architecture de service Java avec eXist [14]**

eXist propose une API Java dont voici quelques-unes des fonctions disponibles [14] :

- $id = executeQuery($xquery).
- getHits($id).
- retrieve($id,$no,$param).
- Xupdate($collection, $Xupdate)

# 4.Conclusion

Dans ce chapitre nous avons présenté le langage XML d'une manière générale

Ce langage est simple et universel permettant de décrire une grande variété de données, Il n'est lié à aucun langage de programmation.

Nous avons conclu que le XML n'est pas seulement utile pour décrire de nouveaux formats de document pour le Web, mais qu'il est également approprié pour décrire les données structurées. Les données structurées peuvent être par exemple des informations généralement contenues dans les tableurs, les fichiers de configuration du programme et les protocoles réseau.

Enfin, nous avons traité la problématique du stockage des données XML et les critères de choix entre les bases de données relationnelles et les bases de données XML native avec une présentation générale de chaque type. Le prochain chapitre sera consacré à la modélisation de notre système.

# Chapitre2 :

# Modélisation et conception du système XML News

# 1. Introduction :

Notre application est un site web dynamique qui offre à un utilisateur la possibilité de consulter, modifier ou chercher des articles de livre ou bien des journaux, enregistrés dans une base de données XML sous formes des documents XML.

Dans ce chapitre nous introduisons l'analyse et la conception de notre système en utilisant le langage UML, par le processus unifié UP.

En effet Ce chapitre est divisé en deux parties :

En premier, Nous allons justifier le choix de langage de modélisation de notre système avec une présentation générale de langage UML et le processus UP.

Ensuite nous allons passer à la modélisation et la conception de notre système. Tout d'abord nous allons détailler la spécification des besoins de notre système en définissant le diagramme de cas d'utilisations. Ensuite les diagrammes de cas d'utilisations vont être détaillés en plusieurs diagrammes de séquences.

Nous terminerons par la modélisation de la base de données dans la deuxième partie.

# 2. Description du système

Notre projet consiste à exploiter une base de données XML via une application web cette dernière est sous forme d'un site web qui interagit avec un SGBD XML Natif. Les documents XML portent sur le thème des livres, des articles des journaux ou bien de magazine. Notre application intitulée XML news

# 3. Le choix d'un outil de modélisation

Pour la réalisation le système XML news, notre choix s'est porté sur le Processus Unifié. Ce processus permettra de modéliser d'une manière claire et précise la structure et le comportement de notre système. Le langage de modélisation qu'on a utilisé est UML, qui est une partie intégrante de la démarche UP.

Nous avons choisi UML Grace à son efficacité et sa modularité par rapport aux autres langages de modélisation.

## 3.1. Définition d UML

UML est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML est un langage formel et normalisé :

- gain de précision.
- gage de stabilité.
- encourage l'utilisation d'outils.

UML est un support de communication performant :

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

## 3.2. Les diagrammes UML :

UML permet de définir et de visualiser un modèle, à l'aide de diagrammes.

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle, pas « le modèle ». Chaque type de diagramme UML possède une structure (les types des éléments de modélisation qui le composent sont prédéfinis).

Un type de diagramme UML véhicule une sémantique précise (un type de diagramme offre toujours la même vue d'un système).

Combinés, les différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système.
UML définit 13 diagrammes, Nous pouvons citer :

### 3.2.1. Diagrammes structurels ou diagrammes statiques (*UML Structure*)

- diagramme de classes (*Class diagram*)
- diagramme d'objets (*Object diagram*)
- diagramme de composants (*Component diagram*)
- diagramme de déploiement (*Deploymentdiagram*)
- diagramme de paquetages (*Package diagram*)
- diagramme de structures composites (*Composite structure diagram*)

### 3.2.2. Diagrammes comportementaux ou diagrammes dynamiques (*UML Behavior*)

- diagramme de cas d'utilisation (*Use case diagram*)
- diagramme d'activités (*Activitydiagram*)
- diagramme d'états-transitions (*State machine diagram*)

#### 3.2.3. Diagrammes d'interaction (*Interaction diagram*)

- diagramme de séquence (*Sequencediagram*)
- diagramme de communication (*Communication diagram*)
- diagramme global d'interaction (*Interaction overviewdiagram*)
- diagramme de temps (*Timing diagram*)

### 3.3. Le processus unifié PU

C'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 adaptée 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.

## 4. Modélisation du système

### 4.1. Spécification des besoins :

- L'utilisateur peut consulter :
  - la liste des journaux /livres.
  - la liste des journaliste/auteur ;
- L'utilisateur peut gérer les livres et les journaux  L'utilisateur peut aussi chercher un livre ou un journal.
- L'administrateur gère la sécurité du système, affecter des privilèges a l'utilisateur, ajouter des nouveaux attributs …

### 4.2. Diagramme de cas d'utilisation

Un cas d'utilisation représente une fonctionnalité du système. Ce diagramme nous permet d'identifier toutes les possibilités d'interaction entre le système et les acteurs, ce qui veut dire toutes les fonctionnalités que le système doit fournir.

La figure suivante représente le diagramme de cas d'utilisation pour l'acteur utilisateur et le système XML news :


**Figure 2.1 : Diagramme de cas d'utilisation coté utilisateur**

La figure suivante représente le diagramme de cas d'utilisation pour l'acteur administratif et le système XML news :

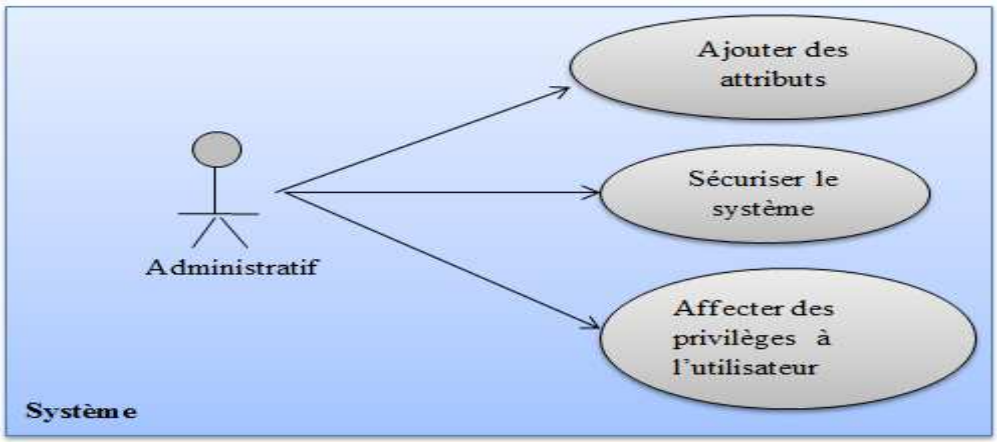



**Figure 2.2 : diagramme de cas d'utilisation coté administratif**

### 4.2.1. Description du scenario

- **Consulter** : L'utilisateur peut consulter la liste des journaux, livres, journaliste, auteur.

- **Ajouter** : l'utilisateur peut ajouter des nouveaux journaux, journalistes, articles, livres, auteurs, ou chapitres….

- **Supprimer** : il peut aussi supprimer un journal ou un livre.

- **Chercher** : l'utilisateur a la possibilité de chercher un journal, un livre, un article ou un chapitre.

- **Ajouter des attributs :** l'administratif peut ajouter des nouveaux attributs au document livre ou journal.il peut même ajouter des nouvelles sources d'information comme magazine

- **Sécuriser le système :** l'administratif peut créer un mot de passe pour sécuriser le système.

- **Affecter des privilèges a l'utilisateur :** l'administratif peut affecter des privilèges a l'utilisateur selon les besoins de système.

## 4.3. Diagramme de séquence

Dans notre projet, nous avons définis deux diagrammes de séquence pour les cas d'utilisations « consulté journal » et « ajouter journal».

- **Consulté la liste des journaux :**

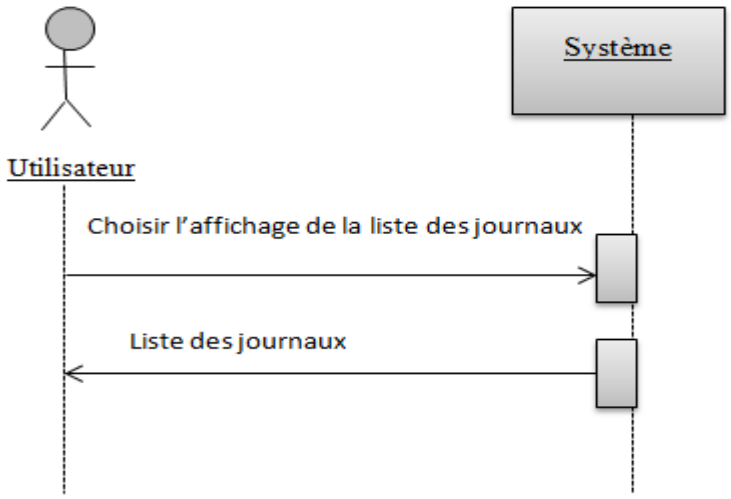



**Figure 2.3 : Diagramme de séquence pour le cas d'utilisation « consulter »**

➢ **Ajouter journal**

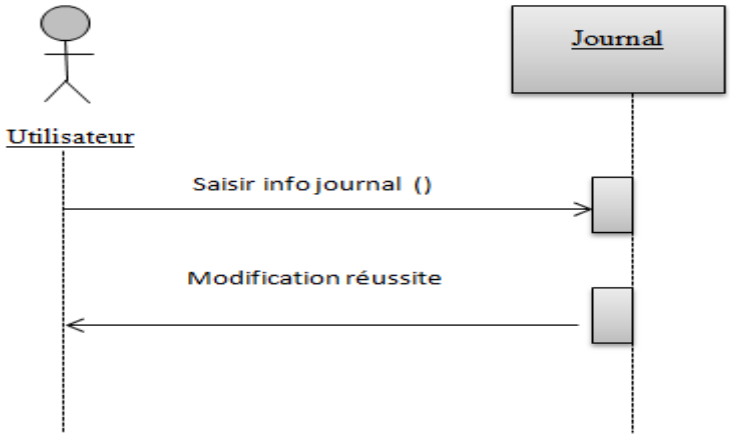



**Figure 2.4 : Diagramme de séquence pour le cas d'utilisation « ajouter »**

# 5. Modélisation de la base de données :

La norme XML en tant que telle doit être vue comme un outil permettant de définir un langage (on dit alors qu'il s'agit d'un métalangage), permettant de créer des documents structurés à l'aide de balises.

Nous avons choisi le XML pour la simple raison qui il est l'un des meilleurs langages de description des données. En plus le XML est un langage puissant et diversifié dans ses applications.

Notre base de données sera élaborée à partir des fichiers XML**,** La création de documents XML commence toujours en mode XML avec le code source XML. La norme XML décrit simplement comment construire un fichier texte permettant de stocker des informations en respectant une structure donnée. Après avoir ajouté les balises et le contenu initiaux.

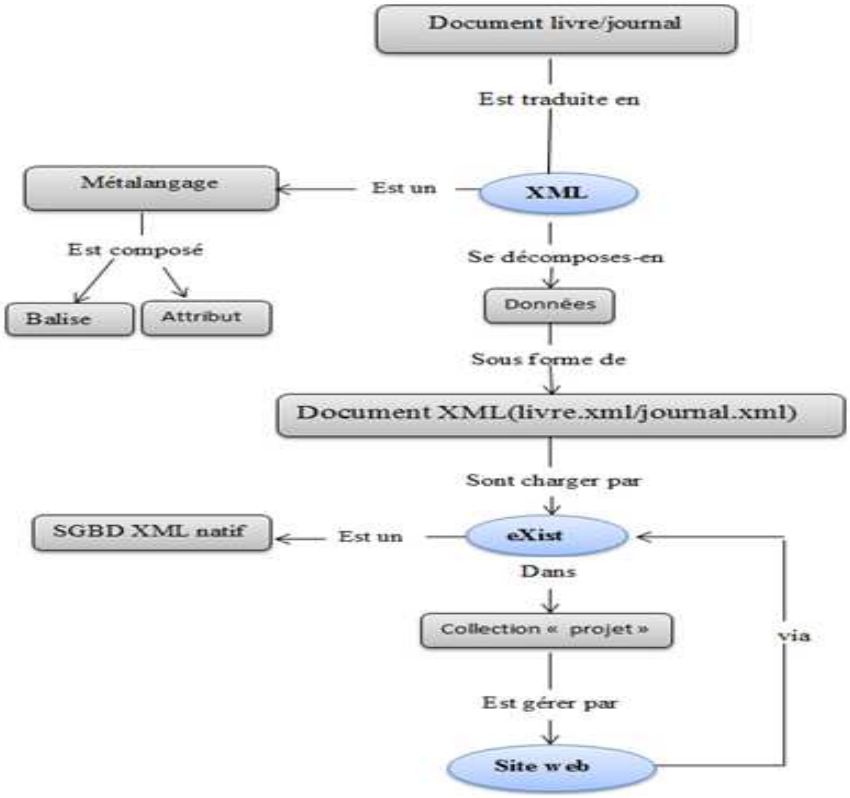



**Figure 2.5 : diagramme de construction de la base de données**

Notre base de données est constituée de deux documents XML « livre » et « journal » :

- **le fichier XML « livre » :**

<?xml version="1.0" encoding="UTF-8"?>

<!-- New XML document created with EditiX XML Editor (http://www.editix.com) at Sun Apr 19 11:47:39 CEST 2015 -->

<livre>

<domaine>informatique</domaine>

<type>scientifique</type>

<titre>Programmation JavaTM pour les enfants, les parents et les grands-parents</titre>

<editeur>Smart Data Processing

</editeur>

<auteur>

<nom>Yakov</nom>

<prenom>Fain</prenom>

</auteur>

<pages>

<page numero="12 to 20">
<chapitre numéro="1">
<titre> TON PREMIER PROGRAMME JAVA </titre>
<contenu> le même programme Java peut tourner (fonctionner)
</contenu></chapitre></page></livre>

- **Fichier XML « journal » :**

<journaux>
<journal>
<classe>quotidien</classe>
<type>politique</type>
<titre>el watan</titre>
<date>1/12/2014</date>
<pages>
<page numero="1">
<rubrique>evennements universitaires</rubrique>
<articles>
<article>
<titre>la grève des étudiants</titre>
<contenu>la grève de étudiants est illimitée, tant que leurs objectifs ne sont pas atteinds...</contenu>
<journaliste>
<nom>belmiloud</nom>
<prenom>farid</prenom>
</journaliste>
</article>
</articles>
</page></journal></journaux>

- **La collection « projet »**

Pour créer notre base de données nous avons choisi eXist comme un outil de stockage car eXist fournit un stockage sans schéma des documents XML dans des collections hiérarchiques.

Apres la création de la collection « projet »**,** nous avons chargé les deux fichiers XML « journal » et « livre » :

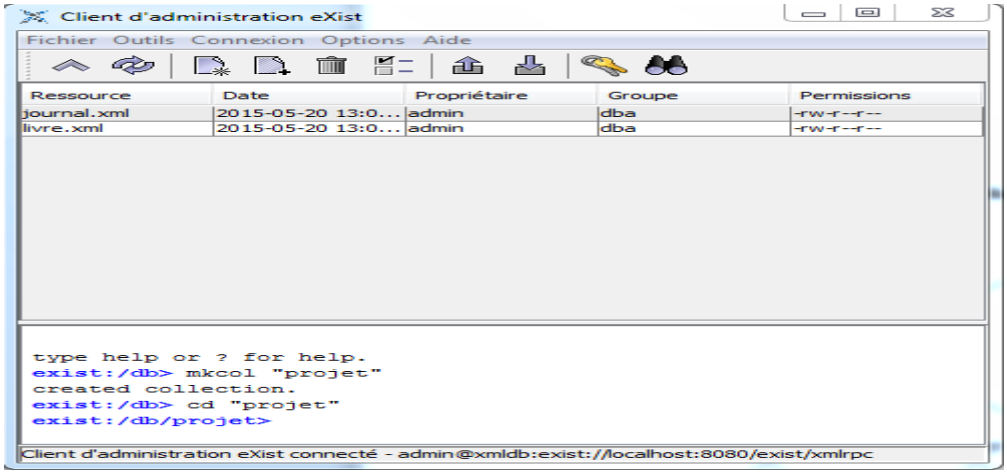



**Figure 2.6 : Ecran de la création de la collection « projet »**

- **Structure arborescence de la collection « projet» :**

La structure de données représentée dans un fichier XML est hiérarchisée dans une arborescence (un élément racine unique; chaque élément peut contenir d'autres éléments qui peuvent à leur tour contenir du texte ou d'autres éléments).

Nous pouvons peut donc représenter les données des fichiers XML sous forme d'arbre. Si on utilise cette représentation pour les documents livre et journal on obtient l'arbre suivant :

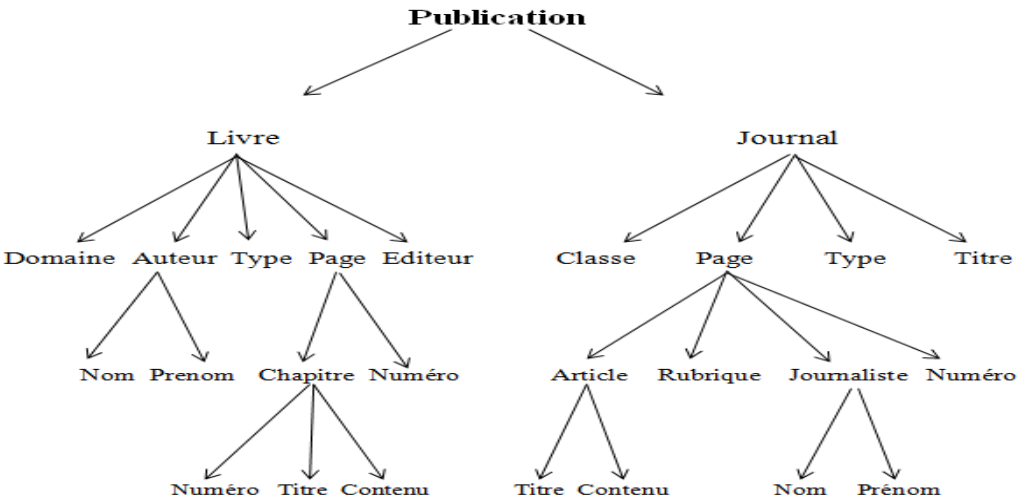



**Figure 2.7 : Arbre XML**

# 6. Conclusion :

On adoptant une conception orienté objet basé sur la méthode up Nous avons pu modéliser notre système XML news. Ensuite nous avons réalisé la modélisation de notre base de données.

Le chapitre suivant, quant à lui, sera consacré à la phase de développement de notre application et que se réalisera en détaillant les différentes interfaces qui le composent

# Chapitre III :

# Implémentation du système XML News

# 1. Introduction

Après la modélisation, vient l'implémentation qui est l'étape de concrétisation technique du projet. C'est la phase de développement pur, celle où il faut produire le code nécessaire aux besoins du site.

Le chapitre suivant est dédié à la présentation de notre application qui porte sur les bases de données XML déjà expliquées dans le chapitre 1.

En fait notre application est un site web dynamique qui offre à un utilisateur la possibilité de consulter des articles de livre ou bien des journaux, enregistrés dans une base de données XML sous formes des documents XML, voir même l'édition de ses derniers avec la possibilité de modification, insertion et suppression en utilisant XQuery.

L'outil eXist (SGBD XML natif) sert à stocker les documents XML : livre ou journal dans une collection avec lequel nous avons créé notre base de données.

Cet outil nous facilite la gestion de notre base de données, par la formulation des requêtes XPath et XQuery, grâce à l'intégration des deux noyaux XPath 2.0 et XQuery 1.0.

En plus il fournit également des extensions à XPath, comme des fonctions adaptés a la recherche textuelle une option très utile dans notre travail. Et comme cité précédemment dans le chapitre1, eXist offre la possibilité d'écrire des modules XQuery, il supporte également quelques composants des technologies telles que XPointer et XPath.

Le "webapp" du serveur permet de traiter les requêtes concernant des fichiers XQuery positionnés dans le répertoire "webapp/xquery".

Le site web a été pensé de manière à ce qu'il soit le plus dynamique possible, il permet en effet la création, la suppression et la modification des informations (article, journal, journaliste…), et ces modifications sont visibles au niveau du site.

Nous avons créé un site web en utilisant le langage le plus célèbre qui est le HTML en utilisant les feuilles des styles CSS et les scripts créent avec java script.

Nous présentons dans ce qui suit l'implémentation de notre système en donnant un exemple de requête pour illustrer le dynamisme de notre site.

# 2. Choix de langage de programmation

Dans cette partie nous allons détailler les différents langages utilisés pour la réalisation du projet. le couple HTML/CSS sont utilisé pour le développement du site web, nous détaillerons dans les sous-parties les raisons pour lesquelles nous avons fait ce choix et pas d'autre.

## 2.1 HTML/CSS

Pour la création de site nous avons choisi un langage informatique de balisage « HTML », pour la simple raison que notre site à la présentation des données et offrir une interface interactif simple, et aisément manipuler par des utilisateurs simple comme affichage (texte, image, hyper texte, vidéo ....), du côté des requête, elles sont effectué par le SGBD eXist qui a comme rôle d'exécuter les requêtes.

Le CSS (« Cascading Style Sheets » : feuilles de style en cascade) est un langage informatique complétant le HTML. Alors que le HTML structure la page Web, le CSS va la mettre en forme en y apportant du style.

HTML/CSS sont complémentaires : HTML est un langage de balise qui sert à définir la structure sémantique de la page d'accueil de notre site : le menu, ses blocs de texte, ses images... tandis que CSS permet de définir la manière dont s'affichent tous ces éléments à l'écran : la couleur du texte, les marges, les bordures…

# 3. Choix de SGBD :

Pour que nos document soient  modifiables et consultables nous avons également cherché un SGBD qui peut répondre aux exigences de notre application, le choix été porté sur un SGBD natif, nous expliquerons également plus tard la raison qui nous à conduit à faire le choix de cet outil.

Il existe de nombreux SGBD-XML propriétaires ou open-sources disponible sur le marché, nous pouvons citer: eXist Sedna et BaseX.

Nous avons choisi eXist parce qu'il est utilisé dans le cadre d'applications web de publication et d'interrogation de données linguistiques. En plus c'est un logiciel open source complet et son mise en œuvre d'est très facile et si simple.

# 4. Langage de requête :

Pour accéder, interroger et modifier les données de notre base de données nous avons choisi des langage de requête tel que le XQuery, XPath, XUpdate

## 4.1. XQuery

XQuery est un langage de requête informatique permettant d'extraire des informations d'un document XML, ou d'une collection de documents XML, il permet aussi d'effectuer des calculs complexes à partir des informations extraites et de reconstruire de nouveaux documents ou fragments XML.

XQuery joue par rapport aux données XML un rôle similaire à celui du langage SQL vis-à-vis des données relationnelles

## 4.2. XPath

XPath est un langage permet de désigner un ou plusieurs nœuds dans un document XML, à l'aide d'expressions de chemin.

## 4.5. XUpdate

XUpdate est un langage de requête XML léger pour modifier des données XML.

# 5. Architecture de l'application

L'objectif premier d'un système d'information, quel qu'il soit, est de permettre à plusieurs utilisateurs d'accéder aux mêmes informations, pour cela, il faut donc regrouper les informations, en terme technique veut dire la centralisation des données au sein d'une base de données. Ces données doivent pouvoir être utilisées par des programmes ou par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de base. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.

. Le client-serveur se situe dans ce besoin de centralisation

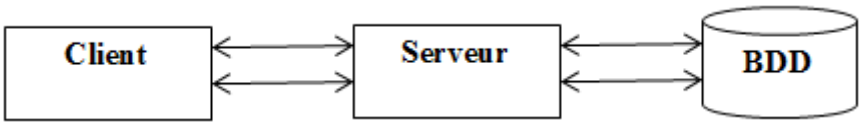



**Figure 3.1 : Architecture client serveur**

Architecture de notre application est de type client-serveur et se découpe idéalement en trois tiers :

- Partie cliente : c'est la partie qui permet le dialogue avec l'utilisateur (interface utilisateur) elle est composé d'une application web
- Partie serveur : c'est la partie qui encapsule les traitements
- Partie SGBD : c'est la partie qui stocke et gère les données

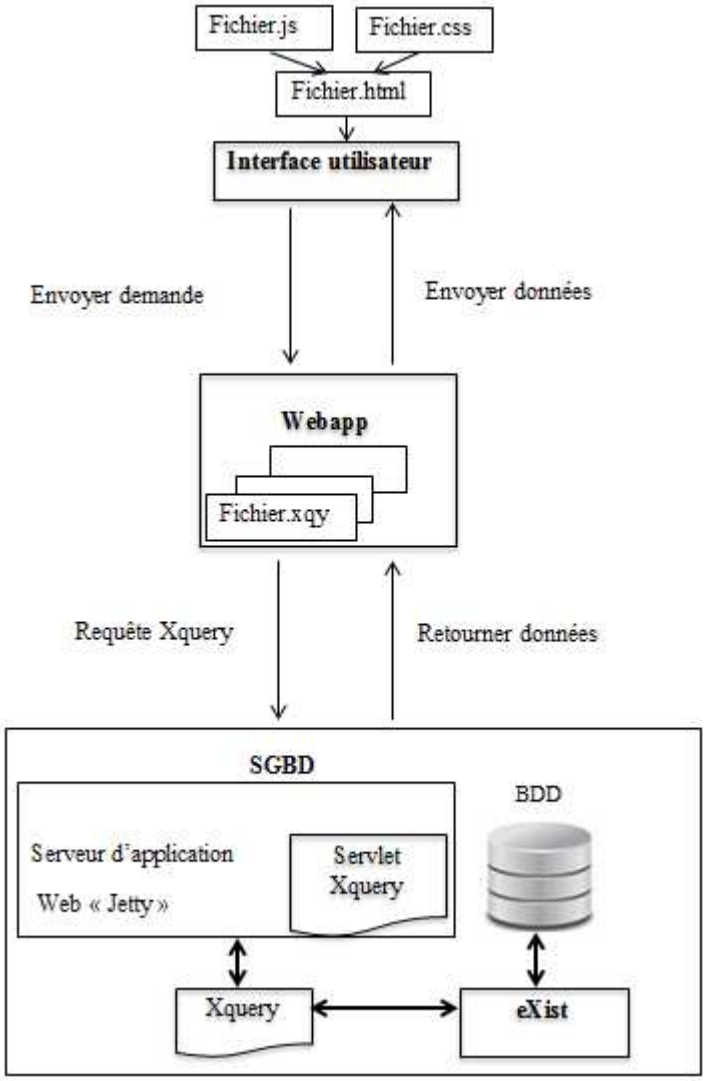



**Figure 3.2 : Architecture de l'application XML News**

- **Coté client (interface utilisateur)**

Un site web est un ensemble de pages web et de ressources liées, accessibles depuis une adresse web et peuvent être consultées en suivant des hyperliens à l'intérieur du site

La conception de l'interface Web comportait deux phases. La première était de construire un affichage web statique qui permis la navigation facile du texte de document de la BDD. La seconde est la phase permettant aux utilisateurs d'effectuer des recherches complexes interrogation de documents XML livre ou journal….. Pour ce prototype, la conception d'interface implique un certain nombre de composants séparés. Les composants statiques de la conception du site comprennent la conception Web en général, HTML mise en page et pour le style nous utilisons les feuilles de style CSS. Les composants dynamiques impliquent l'utilisation de JavaScript pour l'interactivité côté client. Ces composants sont mis ensemble pour former des documents exploitables sur le Web.

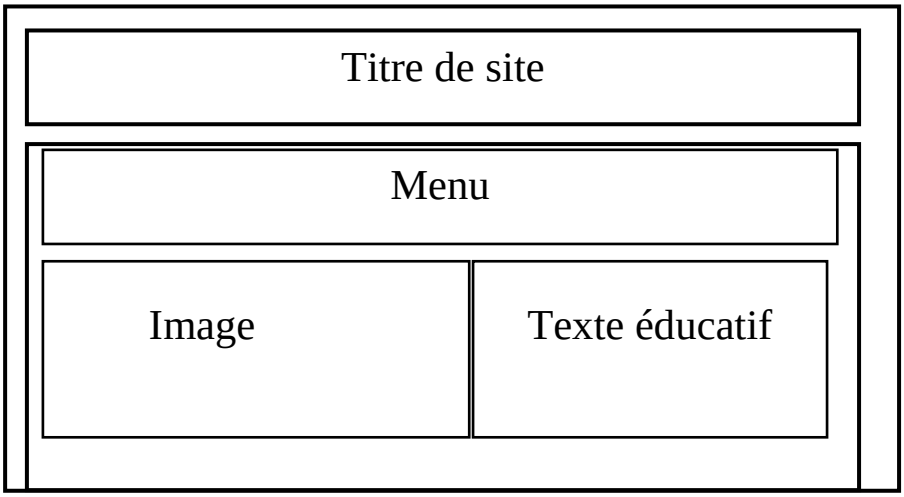



**Figure 3.3 : Maquette de site**

- **SGBD eXist**

Une phase ultérieure de ce projet d'interface est de rendre les documents livre et journal consultable en ligne .pour cela nous avons choisi eXist comme un SGBD XML natif

Le moteur de recherche de eXist fait usage d'une syntaxe étendue de la langue de requête XPathqui est un langage (non XML) pour localiser une portion d'un document XML. XPath a rapidement été adopté par les développeurs comme langage d'interrogation simple d'emploi**,** l'interrogation renforcée de eXist comprend des expressions XPath de base à la recherche à travers la structure nodale du notre base de données XML alors une expression XPath est un chemin de localisation, constitué des pas delocalisation qui sont séparés par le caractère « / ».

eXist nous permet de manipuler notre base de données, comme expliqué auparavant dans le chapitre1, cet outils de gestion propose un serveur d'application ("Jetty") qui permet d'utiliser la base de données XML par l'intermédiaire d'un ensemble de services Web, et il donne aussi la possibilité de créer des applications Web sur la base de XQuery pour récupérer des informations à partir de la base de données et les formater pour laprésentation.

Plusieurs modules permettent d'étendre la fonction standard de XQuery par exemple : XML:DB Module (http://exist-db.org/xquery/xmldb) qui nous permet d'effectuer des modifications sur notre base de données tel que la suppression ou l'ajout.

Les données enregistrées en base devront pouvoir être consultables via un navigateur Web. C'est la puissance d'une Web App.

- **Webapp**

Une application web (aussi appelée web app, de l'anglais) est une application manipulable grâce à un navigateur web « chrome » il permet de traiter les requête XQUERY sur les fichiers .xqy que nous avons créé dans le répertoire webapp comme par exemple display.xqy ou query.xqy...

En fait, l'application va envoyer une demande via Webapp (accessible via une URL: http://localhost:8080/exist/BD/query.xqy

Par exemple "Je veux récupérer toutes les auteurs stockées dans ma base". Le webapp va recevoir la demande, et la traduire en requête compréhensible par la BDD. Il va ensuite envoyer cette requête à la BDD XML, qui va retourner la liste des personnes. Le webapp va récupérer et adapter ces données au format souhaité par le système XML news, puis renvoyer la liste des personnes à l'application, qui va les afficher.

# 6. Description de l'application

- **Ecran da la page d'accueil**

Un site web est habituellement architecturé autour d'une page centrale, appelée «page d'accueil» et proposant des liens vers un ensemble d'autres pages hébergées sur le même serveur.

Lorsque l'utilisateur accède à notre site à partir d'un moteur de recherche ou depuis une référence, c'est sur la page d'accueil qu'il aboutit. Et c'est depuis la page principale qui peut

parcourir les autres pages de différentes opérations de gestion tel que l'affichage, la mise à jour l'interrogation de la base de données. C'est dans ce segment que nous avons appuyé plus lors de la conception et la réalisation de la page d'accueil qui est la plus importante parce que c'est elle qui fait face a l'internaute et joue un rôle très important dans l'impression de ce dernier. Cette page d'entrée prend le nom de page d'index (en référence à l'adresse index.htm), ou de page d'accueil, vers laquelle on pourra revenir à partir de toutes les pages du site. Les internautes trouvent les éléments constitutifs de celle-ci :

- le nom ou le titre de notre site qui en reprendra l'objet.
- un menu qui donne une vue d'ensemble du contenu et amorcera les outils de navigation.
- un élément visuel (image, icône symbolique) pour l'aspect attractif.
- un court texte attractif qui reprend le descriptif fourni aux moteurs de recherche, un texte choisis soigneusement pour donner de valeur a notre site et aussi pour le référencement de ce dernier
- la date et l'heure courante.


**Figure 3.4 : Écran de la page d'accueil**

le menu livre et journal contient des sous menu :

- affichage de la base de données
- affichage les titre des livres/journaux

- interrogation de la base de données
- mise ajour de la base de données
- recherche d'un chapitre/article
- mise a jour de chapitre/article
- suppression de livre/journal
- liste des auteurs/journalists

- **Affichage de la base de données:**

L'utilisateur peut consulter la base de données, En fait, l'application va envoyer une demande via Webapp (accessible via une URL: http://localhost:8080/exist/BD/query.xqy
par exemple "Je veux récupérer la liste entière de tous les journaux stockés dans ma base". Le webapp va recevoir la demande, et la traduire en requête compréhensible par la BDD. Par la suite il va l'envoyé à la BDD XML, qui va répondre avec la liste des journaux qui contient le titre le type et la classe. Le webapp va récupérer et adapter ces données au format souhaité par l'application site de gestion BDD XML, puis renvoyer la liste des journaux à l'application, qui va les afficher comme indiquer dans la figure suivante.

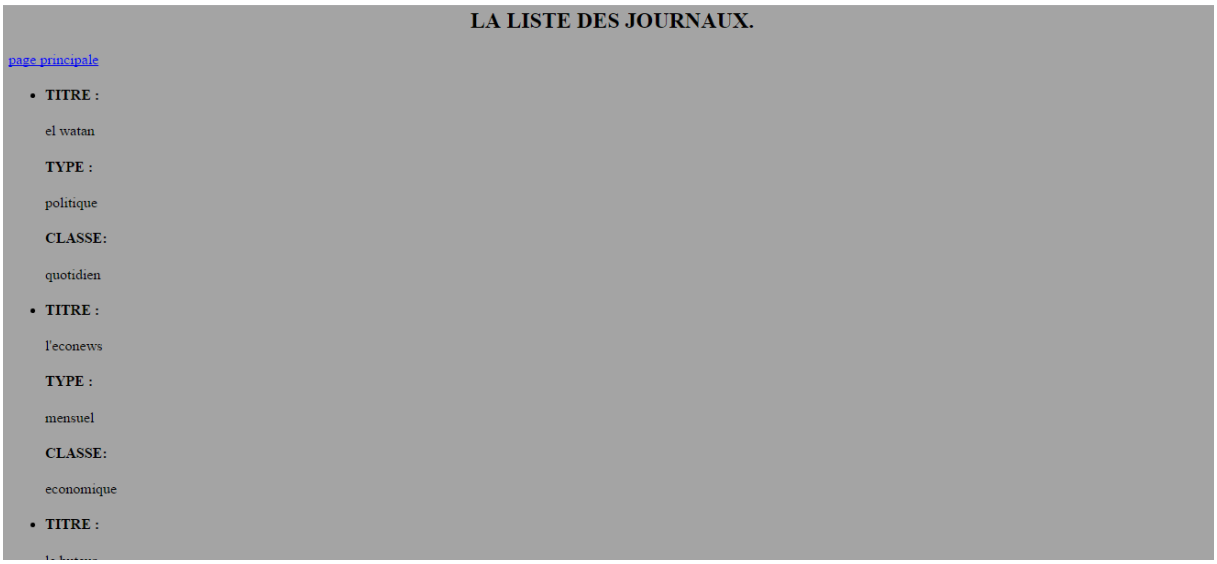



**Figure 3.5 : la liste des journaux**

- **Ecran de l'interrogation de la base de données :**

La page suivante offre à l'internaute la possibilité de faire une recherche dans la base de données, avec certains critères ou nous pouvons le considérer comme un filtre de

recherche qui se résume en recherche souhaité et le mot de recherche, et c'est avec le bouton recherche que se déclenche la recherche.


**Figure 3.6 : Ecran de l'interrogation de la base de données**

**Code source de la recherche:**

```
xquery version "1.0";
(: InitializeDatabase :)
//déclaration des espaces de nom
declare namespace request="http://exist-db.org/xquery/request";
declare namespace xmldb="http://exist-db.org/xquery/xmldb";
//déclaration de la fonction d'affichage
declarefunctionlocal:print_journal($journal as element()){
<div><center><tr><td>Titre : </td><td>{$journal/titre/text()}
</td></tr><tr><td>Type : </td><td>{$journal/type/text()}</td></tr></center>
</div>
};
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
// le titre de la page d'affichage
<title> Résultat de la recherche</title>
</head>
// la page va afficher une erreur si le fichier n'existe pas
{
```

```
if (not ( doc( "/db/projet/journal.xml" ))) then
<body>
<p>
  Erreur : Le fichier journal.xml n'existe pas.
</p>
<a href="javascript:history.go(-1)"> Back </a>
</body>
else
<body>
<center><h2>
//Affichage de toutes les informations relatives au critère de recherche entré par l'utilisateur
Resultats de la requeteXQuery.
</h2></center>
<p>
  {
let $name := request:get-parameter("name",""),
$field :=request:get-parameter("choix","tous")
return
if ($field = "tous") then
for  $journal in doc("/db/projet/journal.xml")//journal
where ($journal/titre=$name or $journal/type=$name
or $journal/classe=$name)
return
local:print_journal($journal)
else if ($field = "titre") then
for  $journal in doc("/db/projet/journal.xml")//journal
where ($journal/titre=$name)
return
local:print_journal($journal)
else if ($field = "type") then
for  $journal in doc("/db/projet/journal.xml")//journal
where ($journal/type=$name)
return
local:print_journal($journal)
```

```
else if ($field = "classe") then
for  $journal in doc("/db/projet/journal.xml")//journal
where ($journal/classe=$name)
return
local:print_journal($journal)
else ()
  }
</p>
<a href="javascript:history.go(-1)"> Back to Query page</a><br/>
<a href="javascript:history.go(-2)"> Back to Main page</a>
</body>
}
</html>
```

- **Ecran de la  recherche d'un article :**

  L'utilisateur peut chercher un article après la saisie de titre de l'article comme il a le choix de chercher dans tous ou titre et lorsqu' il clique sur le bouton chercher tous les informations de l'article seront afficher.

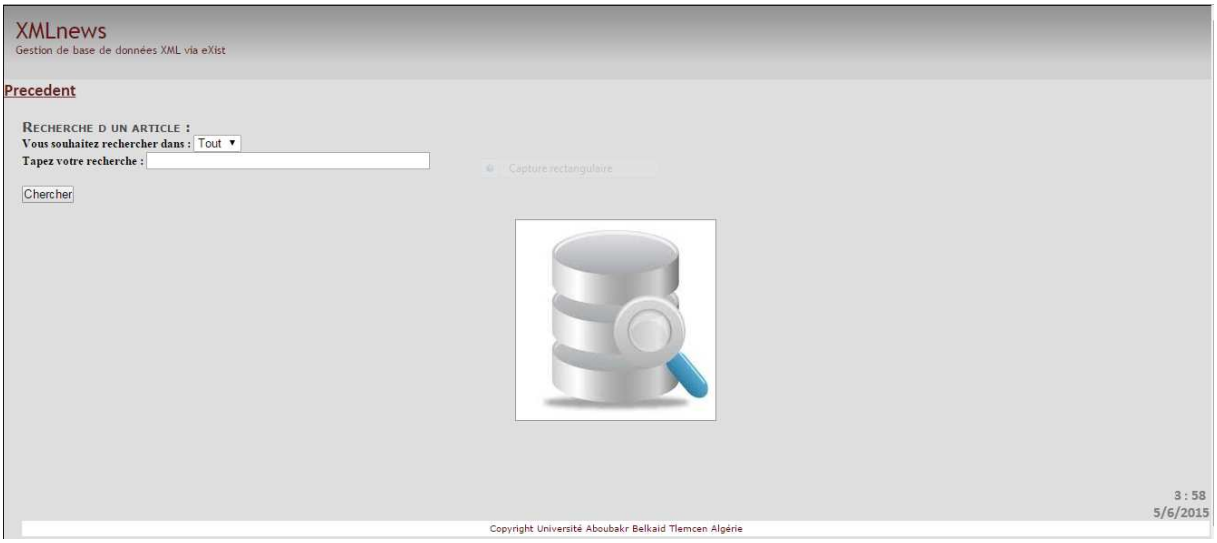



**Figure 3.7 : Ecran de la recherche d'un article**

- **Ecran de la mise a jour de la base de données « ajout d un nouveau journal»**

La page suivante permet la mise à jour d'un objet dans la base de données, nous entrons les informations nécessaires pour la mise à jour comme titre, type et spécialité après la valisation

avec le bouton valider les information de l'objet concernées vont se modifiées avec les nouvelle données entrées via le formulaire indiqué dans la figure suivante :

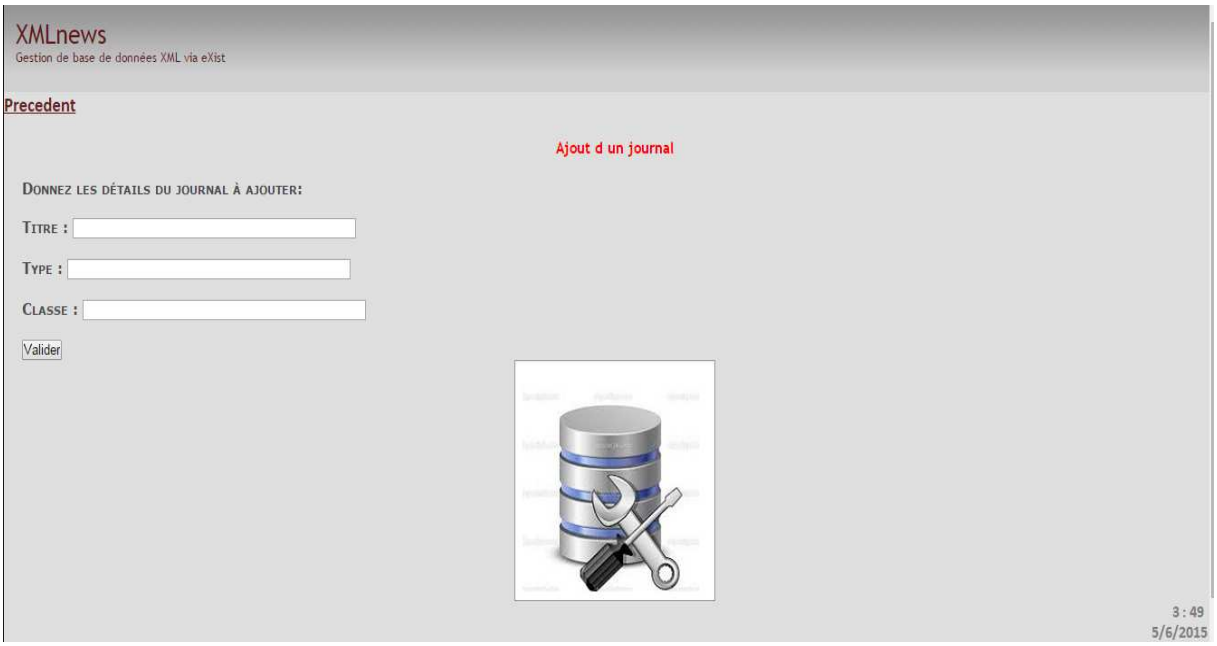



**Figure 3.8 : Ecran de la  mise ajour de la base de données**

- **Ecran de la mise à jour des articles « ajout »**

Là c'est l'opération d'ajout qui aura lieu avec l'entrée des informations nécessaires titre et contenu ensuite la validation avec le bouton ajouter, le nouveau article va être stocké dans la base de données et il peut être consultable avec le menu adéquat à la consultation.


**Figure 3.9 : Ecran de la mise à jour des articles « ajout »**

- **Ecran de mise à jour de journal « suppression »**

Ici c'est pour la suppression d'un article, pour une raison ou d'autre l'internaute peut personnaliser et supprimer un article cela se fait en entrant les informations nécessaires pour l'identification de l'article concerné par la suppression et l'appuie sur le bouton valider


**Figure 3.10 : Ecran de la mise à jour de journal « suppression »**

- **Ecran d'ajout d'un journaliste**

Notre site web offre même la possibilité d'ajouter d'un journaliste en faisant entré les informations de ce dernier nom et prénom et appuyant sur le bouton ajouter et par la suite les informations de nouveau journaliste seront stocké et mémoriser dans notre base de donnée


**Figure3.11: Ecran d'ajout d'un journaliste**

# 7. Conclusion

Dans ce chapitre nous avons, au premier lieu, présenté les différents outils et langages que nous avons utilisé pour la mise en place de notre application, nous avons cité le langage HTML pour la réalisation des pages web, le CSS pour la mise en page, le langage java script pour la partie dynamique, et en fin XQUERY via eXist pour l'élaboration des requêtes d'interrogation de la base de donnée.

Par la suite nous avons présenté brièvement le processus de réalisation de notre application, l'architecture de cette dernière, Aussi nous avons parlé de quelques requêtes qui servent à invoquer la base de données et qui sont pas visible depuis l'interface de site web puisque cette dernière est offerte à des gens jugés comme des utilisateurs sans information pré requises sur les langages informatiques.

Nous avons pu réaliser une application web simple et facile à manipuler, et en même temps offre la possibilité d'interagir avec une base de données XML, via des requête qui se tournent en arrière-plan.

# Conclusion générale

Ce projet de fin d'étude consiste à réaliser un système pour exploiter et gérer une base de données XML.

Au cours de ce mémoire, nous avons présenté dans le chapitre1 le langage XML, le stockage des documents XML dans les différents types de base de données avec une présentation simple de l'un des SGBD XML natif eXist.

Afin de satisfaire les besoins des utilisateurs nous avons réalisé la modélisation dans le deuxième chapitre en utilisant le formalisme UML et la mise en œuvre des bases de données avec le gestionnaire de bases de données eXist.

Dans le deuxième chapitre nous avons présenté les différentes étapes de la modélisation et la de notre système et de la base de données.

Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation. Nous avons appris à mieux manipuler les langages HTML, XQUERY et Java Script.

En effet, ce travail étant une œuvre humaine, n'est pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer d'avantage cette étude. Etant donné que tout travail informatique a été toujours l'œuvre d'une équipe.

Et nous terminons par une citation d'Albert Einstein « La vie, c'est comme une bicyclette, il faut avancer pour ne pas perdre l'équilibre.» Nous envisageons d'améliorer notre système en intégrant d'autres fonctionnalités comme, rajouter une rubrique rédaction, gérer des commentaires laissée par les utilisateurs sur la lecture de certains articles, utiliser le standard XqueryFacility pour la mise à jours des documents XML.

# Références bibliographiques

[1] X.tannier « recherche d'information dans les documents XML » 2006

[2] http://www.liafa.jussieu.fr/~carton/Enseignement/XML/Cours/Annexes/introduction.html 06/03/2015 18h45

[3] Sarah O'Keefe « Structured authoring and XML» 2009

[4] Marjorie Burghart « Editer des sources historiques en ligne grâce à XML »2013

[5 ] Martin Sévigny, AJLSM « introduction a XML » 2002

[6] Mihaela JUGANARU-MATHIEU « XML et les bases de données » 2012

[7] R. Bourett "XML et les bases de données", http://www.rpbourret.com 2003

[8] Mihaela JUGANARU-MATHIEU « XML et les bases de données » 2012

[9] http://jean-luc.massat.perso.luminy.univ-amu.fr/ens/xml/1-intro.html10/03/2015 12H30

[10] http://miage.univ-nantes.fr/miage/D2X1/chapitre_bdxml/section_principes.html 12/03/2015 14h30

[11] B. Amann - Cours No 3 - Stockage de données XML 2006

[12] G. Defrain - « Les bases de données XML natives » 2008

[13] Panorama « les bases de données XML » 2002

[14] http://miage.univ-nantes.fr/miage/D2X1/chapitre_bdxml/section_sgbd_xml.ht 13/04/2015 13h15

[15] A.Belaid « La base de données xml eXist » 2015