

Université Saad DAHLAB - Blida 1



Faculté des sciences

Département d'Informatique

Mémoire présenté par :

Mlles. SID ALI MEBAREK Imene et ZIDANE Meriem

Pour l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Traitement Automatique de la Langue

Sujet :

**Développement d'un système d'alimentation et de suivi et
de supervision dans une chaine de numérisation - Cas
d'étude : Cellules de numérisation manuelle de la
Dhakhira al-Arabiyya-"**

Devant le jury composé de :

Présidente : Mme OUKID Saliha

Université de Blida 1

Examinatrice : Mme NASERI Ahlem

Université de Blida 1

Promotrice : Mme MEZZI Melyara

Université de Blida 1

Encadreur : Mme KHALOUT Fatiha

Académie Algérienne de la
Langue Arabe

2018/2019

ملخص

يهدف هذا المشروع إلى:

إعداد نظام إدارة الوثائق الإلكترونية (EDM) ونظام الأرشفة تحت منصة التقييد EDM.

- تصميم وتطوير نظام لتخزين جميع المعلومات حول المستخدمين، الكتب المضبوطة والأخطاء المرتكبة. هذا النظام يستخدم كأداة لدعم القرار لضمان تخصيص أفضل للكتب.
 - تصميم وتطوير نظام للمساعدة في تصحيح الأخطاء، يدعم التطبيق المطبق إجراء مقارنة بين وثيقتين (قبل / بعد التصحيح) لنمذجة المعرفة مما يسمح للكشف عن الخطأ واقتراح نموذج مصحح من خلال تطبيق قاعدة معينة.
 - بناء قاعدة البيانات لتخزين المعلومات التي تم إنشاؤها لتحسين البحث عن المدقق الإملائي.
- ميزة حلنا هو تحسين الخدمة على مستويين:

- من جانب الوقت الذي تحسن بشكل ملحوظ مقارنة بأداء النهج القديم.
- من جانب جودة الخدمة التي تبرز بوضوح من خلال المقارنة مع أساليب العمل القديمة المستخدمة.

الكلمات المفتاحية:

اللغة العربية، المعالجة التلقائية للغة، TALA، GED، قاعدة المعارف، الأنطولوجيا، BDD، الأخطاء، التصحيح الإملائي.

Résumé

La présente étude a pour objectif de doter les cellules de saisie manuelle installées au niveau de l'Académie algérienne de la langue arabe pour la mise en œuvre du projet de la Dhakhira al-Arabiyya d'un système informatisé pour le suivi de la numérisation. La problématique essentielle de recherche concerne la conception et la réalisation d'un outil qui permet de comparer les documents à partir de deux états : avant et après la correction et cela afin de créer une base de traitement linguistique des erreurs contenues dans les textes numérisés. Cette base constituera par la suite une brique essentielle d'une base de connaissances linguistiques pour la langue arabe.

L'étude a abouti aux résultats suivants :

- Choix d'une solution GED répondant le plus possible aux besoins de la cellule, en terme de : Gestion du Workflow, multi usage des documents et réalisation des tâches-métier.
- Conception d'un outil de stockage des informations concernant les utilisateurs, les livres saisis et les erreurs faites. Ce système est utilisé comme un outil d'aide à la décision pour assurer une meilleure affectation des livres.
- Conception et développement d'un système qui, sur la d'une comparaison entre deux documents (avant / après correction) fait ressortir les formes erronées, celles corrigées et la catégorisation des erreurs de façon automatique. Ce savoir est enregistré au fur et à mesure dans une ontologie.

Mots clés :

Arabe, Traitement Automatique de la Langue, TALA, GED, Base de Connaissance, Ontologie, BDD, Erreurs, Correction Orthographique.

Remerciement

Nous remercions Allah de nous avoir donné le courage, la santé et la motivation pour finir ce projet de fin d'étude.

Au terme de ce travail nous tenons à exprimer notre profonde gratitude à notre chère professeur et promotrice Mme MEZZI pour son énorme soutien, ses précieux conseils, ses encouragements, la confiance qu'elle nous a toujours témoigné, sa disponibilité et son infinité de gentillesse tout long de ce travail.

Nos remerciements les plus sincères vont aussi à Mme KHALOUT Fatiha, notre encadreur au sein de l'Académie Algérienne de la Langue Arabe, pour ses précieuses directives qu'elle nous a prodiguées avec toute intérêt, ainsi que son appui considérable dans notre démarche. Sans oublier tout le personnel qu'on a contacté durant le stage au sein de l'AALA auprès lesquelles nous avons trouvé l'accueil chaleureux, l'aide et l'assistance dont on avait besoin.

Nous adressons nos vifs remerciements aux membres du jury d'avoir bien accepté de consacrer leurs temps afin d'évaluer notre travail.

Nous voudrions aussi remercier nos familles qui nous ont toujours apporté le courage et le soutien.

Enfin, nous tenons à témoigner toute notre gratitude à nos amis d'études et tous ceux, qui ont contribué, de près ou de loin à notre formation et à l'aboutissement de ce projet.

Table des matières

Introduction générale	1
-----------------------------	---

Chapitre I : Gestion Electronique des Documents.

Introduction	4
1. Les systèmes GED	4
1.1. Définition	4
1.2. Concepts de base.....	4
1.3. Principales fonctionnalités d'une GED.....	5
1.4. Avantages de la GED	8
1.5. GED open-source.....	9
1.6. GED open-source les plus utilisées	9
1.6.1. Alfresco	9
1.6.2. Maarch	10
1.6.3. Nuxeo	10
1.6.4. Knowledge Tree	11
2. Choix d'une GED	12
3. Présentation d'Alfresco	12
3.1. Définition	12
3.2. Historique	13
3.3. Les versions d'Alfresco	13
Conclusion.....	13

Chapitre II : Traitement Automatique de la Langue Arabe et la Correction Orthographique.

Introduction	14
1. Les niveaux de Traitement Automatique des Langues Naturelles	14
1.1. Analyse morphologique et lexicographique.....	14
1.2. Analyse syntaxique	15
1.3. Analyse sémantique	15
1.4. Analyse pragmatique	16
2. Les problèmes cernés dans le domaine du TALA	16
2.1. Voyellation	17
2.2. Agglutination des mots	17
3. Correction orthographique	18

3.1.	Notions relatives à la correction orthographique	18
3.1.1.	Approche basée sur le dictionnaire	18
3.1.2.	Approche Stochastique (analyse en n-gramme)	19
4.	Etat de l'art de la correction orthographique	19
4.1.	Détection	19
4.1.1.	Consultation du dictionnaire	19
4.1.2.	Analyse par n-gramme	20
4.2.	Correction.....	21
4.2.1.	Alpha-code	21
4.2.2.	Distance lexicographique : (Levenshtein et Damerau-Levenshtein).....	22
4.2.3.	N-gramme	22
5.	Détection et correction des fautes d'orthographe en arabe.....	24
	Conclusion.....	25

Chapitre III : Ingénierie de Connaissance.

	Introduction	26
1.	Définition	26
1.1.	Pour quelles raisons développer une ontologie ?.....	26
2.	Cycle de vie d'une ontologie.....	26
3.	Les types d'Ontologie	28
3.1.	Ontologie de représentation de connaissances	28
3.2.	Ontologie de haut niveau / supérieure	28
3.3.	Ontologie Générique (Generic Ontology).....	28
3.4.	Ontologie du domaine (Domain ontology)	28
3.5.	Ontologie de Tâches (Task ontology).....	29
3.6.	Ontologie d'application (Application ontology)	29
4.	Composantes d'une ontologie	29
5.	Méthode de recherche orientée ontologie	30
5.1.	Phase 1 : l'étude de faisabilité	30
5.2.	Phase 2 : Modélisation ontologique.....	30
5.3.	Phase 3 : Mise en œuvre informatique de l'ontologie	33
5.4.	Phase 4 : Méthode d'évaluation.....	34
6.	Les méthodes d'ingénierie ontologique	35
6.1.	Méthode de Bachimont	35

6.2. La méthode Methontology	36
7. Les outils de construction d'ontologie	38
7.1. Les outils dépendants de formalisme de représentation	38
7.2. Les outils indépendants de formalisme de représentation.....	39
8. Les notions relatives aux ontologies dans le domaine linguistique	40
8.1. Ontologies linguistiques.....	40
8.2. Ontologies métalinguistiques	40
Conclusion.....	42

Chapitre IV : Conception et Modélisation de la Solution.

Introduction	42
1. Présentation de la structure d'accueil	42
2. Objectifs	43
3. Spécification des besoins fonctionnels	43
3.1. Besoins des utilisateurs	44
3.2. Besoins des administrateurs	44
4. Spécification des besoins non fonctionnels.....	44
5. Conception globale	45
5.1. Processus métier	45
5.2. Schéma d'architecture.....	46
5.2.1. Tâche « Gestion Administrative »	46
5.2.2. Tâche « Gestion de saisie »	47
5.2.3. Tâche « Gestion de contrôle »	48
5.2.4. Tâche « Gestion de correction »	49
5.3. Architecture globale du Système	50
6. Conception de l'architecture	52
6.1. Identification des diagrammes.....	52
6.2. Conception des couches	52
6.2.1. Diagramme de Séquence.....	52
6.2.2. Diagramme d'activité.....	55
6.2.3. Diagramme de Classe	60
6.2.4. Règle de passage du Diagramme de Classe vers le modèle relationnel ...	63
6.2.5. Modèle relationnel	64
7. Architecture globale de l'Ontologie	65

Conclusion.....	70
Chapitre V : Implémentation de la solution.	
Introduction	71
1. Environnement de développement	71
1.1. Environnement matériel.....	71
1.2. Environnement logiciel	72
2. Langage de programmation.....	73
3. Description du travail réalisé.....	75
3.1. Logo	75
3.2. Maquettes du système GED	75
3.2.1. Espace Administrateur	76
3.2.2. Espace Utilisateurs.....	78
3.3. Scénario d’usage d’un système d’aide à la décision.....	81
3.3.1. L’ajout d’un opérateur de saisie	84
3.3.2. L’ajout d’un Livre.....	86
3.3.3. Faire une recherche	87
3.3.4. Les Statistique	92
3.3.5. Liste des livres.....	95
3.3.6. Liste des opérateurs de saisis.....	96
3.4. Scénario d’usage d’un système d’aide à la correction	97
3.5. Construction d’une base de connaissance « Ontologie ».....	102
Conclusion.....	106
Conclusion générale.....	107

Liste des figures

Figure 1.1: Système GED.....	5
Figure 1.2: Le logo d'Alfresco.....	10
Figure 1.3: Le logo de Maarch	10
Figure 1.4: Le logo de Nuxeo.....	11
Figure 1.5: Le logo de Knowledge Tree	11
Figure 3.6 : Cycle de vie d'une Ontologie.	27
Figure 3.7: Validation et Evaluation d'Ontologie.	35
Figure 3.8: Processus de la méthode Bachimont.	36
Figure 3.9: Méthode de METHONTOLOGY.	37
Figure 4.10 : Processus Métier	45
Figure 4.11: Tâche « Gestion Administrative ».	46
Figure 4.12: Tâche « Gestion de saisie »	47
Figure 4.13: Tâche « Gestion de contrôle »	48
Figure 4.14: Tâche « Gestion de correction »	49
Figure 4.15: Architecture globale du Système.	50
Figure 4.16: Diagramme de Séquence d'architecture globale.	54
Figure 4.17: Diagramme d'activité d'architecture globale.	55
Figure 4.18: Diagramme d'activité « Indexation ».....	56
Figure 4.19: Diagramme d'activité « Gestion de Saisie»	57
Figure 4.20: Diagramme d'activité « Gestion de contrôle »	58
Figure 4.21: Diagramme d'activité « Gestion de correction »	59
Figure 4.22: Diagramme de Classe.....	62
Figure 4.23 : Architecture globale d'Ontologie.	67
Figure 4.24 : Diagramme de classe d'Ontologie.....	68
Figure 4.25 : L'exploitation de l'ontologie.....	69
Figure 5.26 : Logo de l'AALA.....	75
Figure 5.27: Page d'authentification d'Alfresco.....	76
Figure 5.28 : Tableau de bord d'Administrateur.....	76
Figure 5.29 : Page d'ajout un utilisateur.....	77
Figure 5.30 : Création d'un Workflow.	77
Figure 5.31 : Contrôler les utilisateurs.....	78

Figure 5.32 : Tableau de bord d'utilisateur.....	79
Figure 5.33 : Vérification du travail demandé.....	79
Figure 5.34 : Page de saisi.....	80
Figure 5.35 : Page d'accueil 1 du système d'aide à la décision.	81
Figure 5.36 : Page d'accueil 2 du système d'aide à la décision.	81
Figure 5.37: Page d'accueil 3 du système d'aide à la décision.	82
Figure 5.38 : Page d'Authentification.	82
Figure 39 : Interface d'un administrateur via le système.	83
Figure 5.40 : Page des services.....	83
Figure 5.41 : Page d'Ajout un opérateur de saisie.	84
Figure 5.42 : Message de confirmation d'ajout d'un opérateur.....	85
Figure 5.43 : Message d'erreur d'ajout un opérateur.....	85
Figure 5.44 : Page d'Ajout un nouvel livre.....	86
Figure 5.45 : Message de confirmation d'ajout un livre.....	87
Figure 5.46 : Message d'erreur d'ajout un livre.	87
Figure 5.47 : Page de recherche.....	88
Figure 5.48 : Page de résultat de recherche d'un livre.	89
Figure 5.49 : Message de non disponibilité du livre.....	89
Figure 5.50 : Page de résultat 1 de recherche d'utilisateur.....	90
Figure 5.51 : Page de résultat 2 de recherche d'utilisateur.....	90
Figure 5.52 : Message de non disponibilité d'utilisateur.....	92
Figure 5.53 : Nombre des livres saisis par opérateur.....	92
Figure 5.54 : Nombres d'erreurs faites par livre / par opérateur.	93
Figure 5.55 : Durée saisie de livre.	93
Figure 5.56 : Pourcentage d'erreurs par livre / opérateur.....	94
Figure 5.57 : Pourcentage d'erreurs par domaine / opérateur.....	94
Figure 5.58 : L'intégralité des erreurs.	95
Figure 5.59 : Page de la liste des livres.....	95
Figure 5.60 : Profil d'utilisateurs.	96
Figure 5.61 : Interface globale du système d'aide à la correction.....	97
Figure 5.62 : L'interface du versionning.	98
Figure 5.63 : Evaluer le nombre d'erreurs.	98
Figure 5.64 : La création du rapport d'erreurs.	99
Figure 5.65 : L'ouverture du rapport.	100

Figure 5.66 : Le rapport d'erreurs.	101
Figure 5.67 : Récupération d'ID.	101
Figure 5.68: Stockage d'erreurs.	102
Figure 5.69 : Construction des classes.	102
Figure 5.70 : Construction des attributs.	103
Figure 5.71 : Construction des relations.	103
Figure 5.72 : Création d'un fichier OWL vide.....	105
Figure 5.73 : Code de remplissage du fichier OWL.	106
Figure 5.74 : Construction d'Ontologie.....	106

Liste des tableaux

Tableau 4.1 : Descreption de la tâche de la Gestion Administrative.	46
Tableau 4.2: Description de la tâche de Saisie.	47
Tableau 4.3 : Description de la tâche de Contrôle.	48
Tableau 4.4 : Description de la tâche de Correction.	50
Tableau 4.5 : Description Globale du Système.	51
Tableau 4.6 : Dictionnaire de Données.	61

Liste des abréviations

Abréviations	Significations
GED	Gestion Electronique des Documents
GEC	Gestion Electronique de Courrier
TAL	Traitement Automatique de la Langue
TALN	Traitement Automatique de la Langue Naturelle
TALA	Traitement Automatique de la Langue Arabe
BDD	Base De Données
SGBD	Système de Gestion de Base de Données
RDF	Resource Description Framework
OWL	Web Ontology Language

Introduction générale.

Introduction générale

De nos jours, nous constatons une croissance exponentielle de la masse de documents numériques et la quantité d'informations partagées de manière électronique. Dans ce contexte, la gestion de documents, leur classement fastidieux et pénible, la dépense d'énergie qui en découle, tout ceci demeure un problème persistant au quotidien. De même, la recherche d'information, devient une tâche indispensable à la performance et au développement des organisations quelles que soient leurs tailles, leurs missions, et leurs secteurs d'activité.

Dans notre cas, nous nous intéressons au domaine de la numérisation et cela à travers le projet de la Dhakhira al-Arabiyya. Celui-ci a pour objet de constituer une base de données textuelles automatisée afin de vulgariser, de façon étendue et profonde le patrimoine arabe ainsi que la production intellectuelle arabe contemporaine et le rendre accessible à tous.

Dans le cadre de ce projet, plusieurs aspects requièrent une méthodologie rigoureuse permettant d'identifier les différents types de documents, les classer selon des critères bien définis, les conserver et en garder une trace tout le long de leur cycle de vie. Pour faire face aux besoins d'archivage et de correction, une nécessité exige le déploiement d'un système de Gestion Electronique des Documents (GED) introduisant le travail collaboratif et le partage de fichiers ainsi que des outils permettant d'améliorer la qualité et l'efficacité du travail de numérisation ; à savoir des outils linguistiques, des outils d'aide à la décision et des bases de connaissances.

La mise en œuvre du projet de la Dhakhira al-Arabiyya se base, selon les porteurs du projet, sur le déploiement de cellules de numérisation manuelles. Ces dernières sont constituées pour numériser des textes rédigés en langue arabe. Ces textes sont par la suite corrigés et validés pour être introduits dans une base de données.

Notre travail consiste à améliorer le rendement de ces cellules, en les dotant d'un système informatisé d'alimentation et de suivi et de supervision de la chaîne de numérisation. La problématique essentielle de recherche concerne la conception et la réalisation d'un outil qui permet de comparer les documents à partir de deux états : avant et après la correction et cela afin de créer une base de traitement linguistique des erreurs contenues dans les textes numérisés. Cette base constituera par la suite une brique essentielle d'une base de connaissances linguistiques pour la langue arabe.

Pour ce faire, il est nécessaire de choisir une solution répondant le plus possible aux besoins de la cellule, en terme de :

- Collaboration pour la production de documents numériques de la Dhakhira (Gestion du Workflow, Gestion du flux d'échange des documents produits) ;
- Multi usage des documents (Gestion des rôles, permissions et droits d'accès) ;
- Réalisation des tâches-métier (Edition/création, enrichissement/indexation, modification/correction, validation/stockage, transformation/publication) ;
- Outil d'aide à la correction des erreurs basé sur la modélisation du savoir permettant la détection de l'erreur et la proposition d'une forme corrigée en appliquant une règle donnée. Dans un premier temps, cet outil doit permettre la génération de rapports statistiques et descriptifs des erreurs (taux d'erreurs, types d'erreurs, règles utilisées pour corriger les erreurs) selon les profils de celui qui a saisi le texte et celui qui l'a corrigé.

La démarche adoptée pour la réalisation d'un système informatisé pour faciliter le travail de la cellule de numérisation et améliorer son rendement est la suivante :

- Passer en revue les systèmes existants répondant aux fonctionnalités suivantes :
 - Gestion du flux d'échange des documents produits ;
 - Stockage des documents ;
 - Versionning des documents : en vue de comparaison pour déterminer les corrections effectuées ;
 - Génération de rapports : Taux et types d'erreurs et les règles aboutissant à la correction.
- Choix du système répondant le plus aux fonctionnalités demandées et permettant une adaptation aux besoins de la cellule ;
- Déterminer les modalités de récupération des informations nécessaires à la construction/amélioration d'une base de connaissance.

Dans ce cadre, le travail présenté dans ce mémoire, s'articule autour d'une introduction générale suivie de cinq chapitres.

L'introduction générale donne un aperçu sur l'organisme d'accueil et explique quelques notions liées au contexte général du projet. Le premier chapitre est consacré à l'étude des systèmes GED, pour solutionner la problématique liée au choix d'un système informatique pouvant répondre aux besoins de la cellule de numérisation de la Dhakhira al-Arabiyya. Le deuxième chapitre est consacré au traitement automatique de la langue arabe (TALA), comme étant le domaine dans lequel il est question de modélisation du savoir linguistique nécessaire à la conception de correction orthographique.

Le troisième chapitre est consacré à l'étude des ontologies et aux différents concepts qui y réfèrent, et cela compte tenu du fait que le projet a pour but final d'aller vers la construction d'une base de connaissance linguistique.

Le quatrième chapitre intitulé « Conception et modélisation de la solution », explique en détail le contexte de notre étude et sa problématique en se basant sur les besoins définis par les porteurs du projet. En effet, ce chapitre inclut une description du projet à travers une capture des besoins fonctionnels et une analyse de ces derniers. Ainsi que la spécification des besoins techniques et l'architecture adoptée. Il présente aussi la conception préliminaire et la conception détaillée des fonctionnalités attendues du système développé.

Le cinquième chapitre porte sur l'implémentation et l'intégration de la solution logicielle en présentant les différentes interfaces utilisateur. Il comporte aussi une description des outils de travail et les tests des cas d'utilisation.

Enfin, nous concluons ce mémoire en présentant des perspectives futures pour notre travail.

Chapitre I : Gestion Electronique des Documents.

Introduction

Ce chapitre présente, une étude sur les systèmes GED, et cela à fin de proposer une solution à la problématique liée au choix d'un système informatique pouvant répondre aux besoins de la cellule de numérisation de la Dhakhira al-Arabiyya. Une autre partie sera consacrée à la présentation du système de GED choisi et ses avantages par rapport à ses concurrents. Celle-ci sera suivie de la justification du choix de la solution adoptée pour mettre en place notre système.

1. Les systèmes GED

1.1.Définition

La gestion électronique des documents, désigne un procédé informatisé visant à organiser et à gérer des informations et des documents électroniques au sein d'une organisation [1]. Le terme GED désigne également les logiciels qui permettent la gestion des contenus documentaires.

Il y a quatre étapes majeures dans un processus de GED : l'acquisition, le classement, le stockage et la diffusion.

1.2.Concepts de base

Ce processus englobe l'acquisition du document papier et sa transformation en document électronique.

Le processus de GED peut être décomposé sous forme de 4 étapes [2]:

- Acquisition :

L'acquisition peut selon les cas inclure les documents multimédia tels que les images, les sons et les vidéos.

- Indexation :

Le document entre dans le système de gestion. Il doit être indexé selon ses métadonnées (type de fichier, date d'ajout, responsable de l'ajout) et son contenu pour faciliter les recherches ultérieures (recherche en texte intégral).

- Stockage :

La GED doit garantir l'intégrité, la sécurité et le sauvegarde des documents. Il faut donc s'intéresser au stockage initial des documents et à leurs sauvegardes.

- Diffusion et Archivage :

Certains documents doivent légalement être conservés mais n'ont plus de pertinence pour les opérations courantes. Ils deviennent alors des candidats à l'archivage.

La figure suivante reprend les quatre étapes décrites ci-dessus :

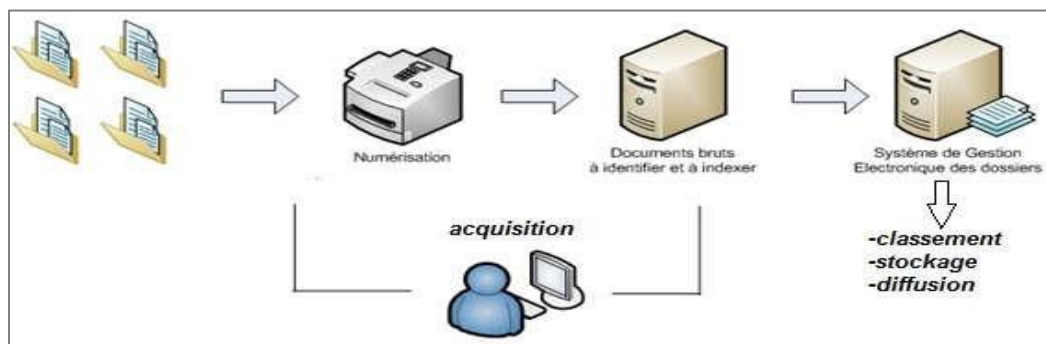


Figure 1.1: Système GED

1.3.Principales fonctionnalités d'une GED

Un dépôt centralisé de documents électroniques ne constitue pas une GED au sens strict du terme. La présence de la majorité des fonctionnalités présentées ici est nécessaire pour parler réellement de GED [3].

- Numérisation des documents :
 - La GED ne peut pas se limiter aux documents créés directement sous forme électronique, elle doit également prendre en compte les documents papiers en les numérisant.
 - La numérisation elle-même ne fournit qu'un fichier image peu exploitable. Un maximum d'informations doit donc en être extrait, par exemple grâce à un logiciel de reconnaissance des caractères (OCR).

- Métadonnées :

Chaque document est associé à des métadonnées qui peuvent inclure :

- Le titre ;
- L'auteur ;
- L'édition ;
- La source ;
- La date.

Ces métadonnées constituent un premier niveau d'indexation.

- Folksonomie :

- Est un système de classification collaborative décentralisée, spontanée, basé sur une indexation effectuée par des non-spécialistes.
- Lors de l'ajout, les utilisateurs saisissent des mots-clés qui décrivent mieux – ou qui leur permettront de mieux retrouver – le document qu'ils sont en train d'ajouter à la base documentaire.
- Exemple : Documents Comptables, Frais d'établissement, Frais d'avocat constitution.

- Taxonomie :

- La taxonomie permet de mieux encadrer l'indexation en contraignant le choix des mots-clés.
- Ceci en structurant les termes entre eux sous forme d'hierarchisation simple permettant d'établir un lien précis entre un terme et ses enfants dans l'arborescence.
- Chaque lien donne un sens additionnel et une signification au terme initial.
- Plus les termes enfants sont nombreux, plus on tend vers un vocabulaire organisé.

- Recherche en texte intégral ou full-text :
 - Lorsque l'indexation est automatique et couvre tout le contenu d'un document, on peut effectuer des recherches en texte intégral ou « full-text », similaire à ce que peut proposer un moteur de recherche sur le Web.
 - En revanche, il faut souvent employer les termes exacts utilisés dans les documents faux positifs.
 - La précision de la recherche full-text peut être améliorée lorsqu'elle est utilisée en conjonction avec d'autres critères.
- La restitution :
 - La restitution permet de rechercher et de remettre les documents numériques à l'organisme qui les a produits, puis de les détruire de manière définitive au sein de son système d'archivage.
 - Le mode d'indexation va impacter très considérablement les possibilités en matière de restitution.
 - D'où encore une fois l'importance de faire les bons choix dans la stratégie d'indexation.
- Sauvegarde :
 - La mise en place d'une GED impose la sauvegarde régulière et fiable du dépôt de document.
 - Par exemple, perdre une année entière de factures à cause du dysfonctionnement d'un disque dur est une perspective inacceptable.
 - A terme il faudrait aussi penser à faire une sauvegarde plus ou moins simultanée sur un site déporté.
- Sécurité et Confidentialité :
 - Un système de GED doit fournir une gestion forte des droits d'accès (lecture, modification, suppression, création).
 - Ces droits sont alors attribués à des personnes et/ou à des groupes de personnes prédéfinis.
 - En plus des droits d'accès, il serait intéressant pour la solution d'intégrer la notion de confidentialité.
- Workflow :

- L'utilisation correcte d'un système de GED suppose qu'un workflow (ensemble de procédures définissant les tâches individuelles à effectuer pour différentes opérations) est mis en place à tous les niveaux de l'entreprise.
- Une procédure de workflow permet d'organiser et d'automatiser le déroulement de campagnes entre les différents acteurs concernés (créateur, intégrateur, rédacteur, responsable marketing, responsable email...).

✓ **Principe [3]:**

- Un workflow, ou flux de travaux, décrit et formalise toutes les étapes d'un processus et les flux d'informations associés.
- Un workflow n'est pas nécessairement linéaire. Il peut contenir des embranchements et des boucles accompagnées de conditions.
- Par abus de langage, le terme de « workflow » peut aussi désigner le logiciel permettant d'exécuter ces flux.
- Exemple d'applications :
 - Production d'un rapport : écriture, corrections, validation des corrections, validation du document, signature, diffusion.
 - Gestion d'une non-conformité : déclaration de la non-conformité, validation, correction, validation de la correction.
 - Le cycle de vie d'un document : création, stockage, diffusion, archivage, destruction.

1.4. Avantages de la GED

Pourquoi utilise-t-on la GED ?

La GED est utilisée pour :

- Rendre l'information disponible pour tous les utilisateurs quel que soit leur lieu ;
- Limiter la circulation des documents papier et éviter les pertes des données ;
- Eviter la multiplication des documents pour faciliter leur mise à jour ;
- Rassembler l'ensemble des documents dans un unique dossier informatisé ;
- Sécuriser le stockage et l'accès aux documents et aux dossiers ;
- Favoriser la sauvegarde des documents sensibles ;
- Assurer l'archivage des informations papiers et la consultation avec des outils de Versionning.

En plus, la GED permet une diffusion beaucoup plus rapide et plus large de document (information), et donc l'entreprise va économiser en matière de coût d'espaces de stockages, d'impression et augmenter la productivité de ses employés [4].

1.5.GED open-source

Pourquoi Open-source ?

Les raisons du choix d'un produit open source sont multiples, dont les raisons économiques, l'ouverture et la standardisation, la maintenabilité, la sécurité et le support communautaire.

La philosophie de l'Open Source, face aux solutions propriétaires, consiste à partager les développements informatiques dans le but d'une libre circulation des idées et des outils.

Une des conséquences directes de cette impulsion pour la GED est l'extraordinaire vivacité des communautés de développeurs autour de solutions diffusées et améliorées sans cesse bénévolement. En effet, l'Open Source garantit, d'une part, une meilleure qualité de code par un partage et un contrôle plus ouvert des sources informatiques et garantit d'autre part, l'indépendance d'un client à son fournisseur car une solution de GED Open Source bien connue proposée par un prestataire peut être d'autant plus facilement reprise par un autre [5].

Les GED open source présentent des avantages et des inconvénients qui seront détaillées dans cette section [6].

1.6.GED open-source les plus utilisées

1.6.1. Alfresco

Alfresco est une plateforme collaborative, orientée historiquement sur un cœur GED, qui vise à augmenter la productivité d'une entreprise.

A cette fin, les documents sont dématérialisés et indexés au sein d'Alfresco pour permettre ainsi une recherche des documents indexés.

- Points forts :

Alfresco est un outil libre très employé pour gérer intelligemment les documents en entreprise. Il est écrit en Java. Il offre la possibilité de créer des Workflow adapté aux besoins.

- Points faibles :

Manque de suivi commercial entre la version communautaire et la version payante.



Figure 1.2: Le logo d'Alfresco

1.6.2. Maarch

Maarch est un système complet de GED et de Gestion Electronique de Courrier (GEC). Il traite le courrier Arrivée, Départ, Interne, ainsi que les documents de GED avec une grande richesse fonctionnelle.

Maarch Courrier est prêt à l'emploi, entièrement libre, et utilisable pour les besoins simples comme pour les plus complexes.

- Points forts :

Maarch utilise un système de gestion avancée à la fois sur les modules applicatifs et par rapport aux contenus.

- Points faibles :

Maarch dispose d'un système de workflow faisant appel à des paramétrages complexes.



Figure 1.3: Le logo de Maarch

1.6.3. Nuxeo

Nuxeo est une plateforme de gestion de contenu qui permet aux architectes et aux développeurs de faciliter, concevoir, déployer et exploiter les applications métiers. Sur site ou sur le Cloud, la technologie d'Enterprise Content Management (ECM) de Nuxeo offre des solutions intégrées de gestion de documents.

- Points forts

Basée sur Java et doté d'une architecture logicielle moderne et complètement modulaire et extensible, se démarquant ainsi des solutions d'ECM traditionnelles, Nuxeo permet de configurer de manière simple et rapide les applications de gestion de contenu les plus complexes.

- Points Faibles

L'utilisation semble hautement compliquée pour l'utilisateur simple. La configuration est compliquée vu que nous devons modifier des fichiers XML.



Figure 1.4: Le logo de Nuxeo

1.6.4. Knowledge Tree

Knowledge Tree est une solution de GED, disponible également en version gratuite et payante. Elle utilise les classiques PHP MySQL. Ainsi, elle propose les fonctionnalités qu'on doit attendre de ce type de logiciel.

- Points forts :

Knowledge Tree est facile à utiliser. Il est doté d'une interface agréable offrant des fonctionnalités suffisantes.

- Points faibles :

Knowledge Tree n'intègre ni Microsoft Office ni la signature électronique, dans sa version gratuite.



Figure 1.5: Le logo de Knowledge Tree

2. Choix d'une GED

Pour choisir une GED, la première chose à faire est de recenser les besoins. Ce cahier des charges permet d'identifier la solution appropriée dans notre cas. Les questions de réversibilité des informations, ou de sauvegarde des versions précédentes, si elles sont importantes pour nous, peuvent constituer des critères à l'appui de notre décision. Nous rappelons, que dans notre cas les aspects les plus importants sont liés à la proposition d'un système collaborative permettant d'effectuer les tâches allant de l'indexation à l'archivage.

Après cette étude des principaux systèmes GED open-source, nous avons choisi Alfresco pour sa mise en valeur du travail collaboratif, et ses fonctionnalités complètes de la GED [7].

- Alfresco offre la possibilité d'essayer la version Community avant d'acheter les licences Enterprise, Modèle de licence simple – coût particulièrement avantageux pour les déploiements concernant de nombreux utilisateurs, La licence couvre toutes les fonctionnalités, support téléphonique inclus.
- Alfresco est un produit performant, simple d'utilisation, hautement personnalisable, avec tous les avantages d'une solution open source.
- Alfresco a une architecture supportant la montée en charge. Le stockage est basé sur un système de fichiers permettant une meilleure montée en charge, une performance et une gestion aisée, Pas de limite de taille de fichiers.
- Il supporte différentes plateformes : Windows, Mac et Linux.

3. Présentation d'Alfresco

3.1. Définition

Alfresco est un système de gestion de contenu (ECM), open source développé en langage JAVA en 2005 et distribué sous licence libre par John Newton et John Powell, (Les co-fondateurs d'Alfresco software), en Grande-Bretagne.

En effet, il peut se comporter sur un ordinateur comme un disque virtuel, ce qui permet à l'utilisateur de partager des fichiers simplement en les déplaçant sur le disque dédié [8].

3.2. Historique

Alfresco est le produit d'une équipe composée du co-fondateur de Documentum et d'anciens d'Intervowen, deux entreprises leaders dans le domaine de la gestion des documents et qui ont comme but de trouver des solutions pour les documents qui sont non structurés, depuis 1990. En effet, Alfresco n'est pas né pour répliquer une solution open source vis-à-vis des solutions propriétaires comme Microsoft SharePoint, File Net d'IBM ou Documentum [8].

3.3. Les versions d'Alfresco

Alfresco se présente sous forme de trois versions :

- Alfresco Community :

Une version téléchargeable, modifiable et utilisable, distribuée en licence libre (LGPL)

- Alfresco Enterprise :

La version commerciale rajoute un support technique, ainsi que d'autres fonctionnalités complétant la solution GED (Alfresco offre 30 jours gratuits afin de tester cette version) [9].

- Alfresco Version Cloud :

Cette version permet de partager le contenu tout en centralisant les données sur les serveurs Cloud (Amazon Web Services aux Etats Unis). Elle offre une gestion complète et un niveau de sécurité supérieur.

Conclusion

Dans ce chapitre nous avons décrit le fonctionnement du système de GED et son apport aux besoins de la cellule de la numérisation de la Dhakhira al-Arabiyya. Nous avons également présenté Alfresco comme solution à adopter en faisant un survol de ses fonctionnalités les plus utiles.

Chapitre II : Traitement Automatique de la Langue Arabe et la Correction Orthographique.

Introduction

Le Traitement Automatique des Langues (TAL) est une discipline qui associe étroitement linguistes et informaticiens. Il repose sur la linguistique, les formalismes (représentation de l'information et des connaissances dans des formats interprétables par des machines) et l'informatique. Le but du TAL est réellement de “comprendre” le sens des phrases, les idées qui s'en dégagent et ce de manière à pouvoir “traiter” de la manière la plus optimale et la plus naturelle d'un point de vue humain ces phrases [10].

Le traitement automatique de la langue naturelle (TALN) touche plusieurs domaines, telles que, les applications de correction orthographique, les applications de communication homme/machine, les applications de traduction automatique, etc. [10]

La correction orthographique assistée par ordinateur fait l'objet de recherche depuis les années 50, et de nombreux logiciels d'aide à la rédaction existent. Ceux-ci se contentent en générale de signaler les erreurs détectées et de proposer les corrections possibles, la tâche de correction en elle-même est donc laissée à l'utilisateur lui-même [11].

1. Les niveaux de Traitement Automatique des Langues Naturelles

La mise en œuvre d'une véritable analyse linguistique nécessite la capacité, pour le système de pouvoir [12]:

- Reconnaître (niveau morpho-lexical) ;
- Structurer (niveau syntaxique) ;
- Comprendre (niveau sémantique) ;
- Contextualité (niveau pragmatique).

1.1. Analyse morphologique et lexicographique

La morphologie concerne l'étude de la structure des mots, alors que la lexicographie recense et classifie les formes d'une langue. Ces domaines s'intéressent plus particulièrement à la dimension paradigmatic de la langue (par opposition à sa dimension syntagmatic).

Nous pouvons présenter, ci-dessus un exemple d'analyse d'une chaîne de caractère en langue arabe :

يأكل محمد تفاحة

La segmentation se fera de la manière suivante : U1=يأكل , U2=محمد , U3=تفاحة .

Maintenant, on pourra associer toutes sortes d'informations aux U_i (i = 1, 2, 3, ...), comme par exemple :

U2 = محمد:

Informations morphosyntaxiques : nom propre, masculin, singulier.

Informations sémantiques : animé humain, prénom ...

1.2. Analyse syntaxique

La syntaxe vise à étudier des contraintes entre les catégories morphosyntaxiques devant être prises en compte pour la description des séquences de mots "acceptables" dans une langue donnée. Les contraintes peuvent être liées à la sélection (règles d'accord) ou peuvent être positionnelles. La description des contraintes caractéristiques d'une langue se fait par le biais d'une grammaire.

La syntaxe permet de résoudre (ou réduire) des ambiguïtés aux niveaux inférieurs :

- Au niveau phonétique :

إلى في قريش

إيلاف قريش

- Au niveau lexical :

يشرب كأسن

يشرب كأسا

1.3. Analyse sémantique

Au sens littéral, la sémantique vise à étudier du sens hors contexte. Le niveau sémantique est encore beaucoup plus complexe à d'écrire et à formaliser que les niveaux de traitements précédents, par conséquent les réalisations qui sont opérationnelles sont peu nombreuses, et elles concernent des applications très limitées où l'analyse sémantique se réduit à un domaine parfaitement circonscrit ; par contre, on est encore loin de savoir construire en grandeur réelle des analyseurs

sémantiques généraux qui couvriraient la totalité de la langue et seraient indépendants d'un domaine d'application particulier [13].

Le traitement sémantique prend comme unité d'analyse la phrase, et conduit à représenter sa partie significative. Ces phrases, dont l'analyseur sémantique doit d'écrire le sens, se composent d'un certain nombre de mots identifiés par l'analyse morphologique, et regroupés en structures par l'analyse syntaxique. Ces mots et ces structures constituent autant d'indices pour le calcul du sens : on pourrait dire, que le sens résulte du double donné du sens des mots et du sens des relations entre mots [13].

1.4. Analyse pragmatique

La pragmatique vise à l'étude du sens en contexte. L'analyse sémantique de la phrase isolée, traitée hors contexte, ne conduit à représenter que la partie de la signification des mots dans cette phrase. Elle n'épuise donc pas ce que l'on peut appeler la signification complète d'un texte, telle que l'humain l'appréhende lors d'un processus de compréhension. C'est la raison pour laquelle une analyse pragmatique est nécessaire pour trouver la signification "réelle" des phrases liées aux conditions situationnelles et contextuelles d'utilisation des mots [13].

Exemples de données :

- Au niveau morpho lexical : règles morphologiques (grammaire du mot) et construction de lexiques informatiques.
- Au niveau syntaxique : la construction de grammaires (formelles) de la langue.
- Aux niveaux sémantique et pragmatique : construction de modèles formels des connaissances sur le monde (propositions logiques, réseaux sémantiques, graphes conceptuels, ...).

2. Les problèmes cernés dans le domaine du TALA

La langue arabe rencontre deux principaux problèmes en traitement automatique : le premier, général, concerne l'agglutination des mots ; Le second, spécifique, se rapporte à l'absence de voyelles à l'écrit.

2.1. Voyellation

Comme nous l'avons évoqué, les textes arabes sont ordinairement dépourvus de diacritiques. Pour les lire, tout un processus mental est nécessaire : identifier le mot comme appartenant au lexique puis lui attribuer ses voyelles dans son contexte, ce qui nécessite la compréhension du texte.

Ce problème est similaire à celui de l'accentuation automatique des textes en français, mais dans des proportions beaucoup plus importantes : 28% des mots français en usage sont ambigus contre 95% en arabe (mesures effectuées sur un texte de 23.000 mots).

Pour le traitement automatique d'un texte, il est indispensable d'introduire les voyelles avant (synthèse à partir du texte) ou après (reconnaissance optique) le traitement. Cette opération est appelée voyellation automatique ou vocalisation automatique quand elle est effectuée par la machine.

Elle revient à choisir pour chaque mot du texte une voyellation possible parmi l'ensemble des voyellations acceptées dans la langue. Généralement, elle se déroule en deux étapes :

- Une analyse morphologique qui va assigner à chaque mot non-voyellé l'ensemble des mots voyelles correspondants. Cette analyse recourt à un lexique supposé exhaustif qui intègre toutes les formes canoniques et fléchies des mots.
- Une analyse syntaxique pour réduire l'ambiguïté au vu du contexte syntaxique. Au mieux, une seule voyellation est retenue.

2.2. Agglutination des mots

La plupart des mots arabes sont composés par agglutination d'éléments lexicaux de base (proclitique + base + enclitique). Par exemple, la détermination peut s'exprimer par agglutination de l'article / al / avant le mot الولد/alwaladu/ ' l'enfant ou par agglutination d'un pronom personnel après celui-ci ولده/waladuhu/ ' son enfant.

De même, les pronoms personnels peuvent se rattacher aux verbes ضربه/Darabahu/ ' il l'a frappé, les particules régissant le cas indirect aux noms كداره/kadArihi/ comme sa maison et les conjonctions de coordination aux verbes فذهب/favahaba/ ' et il est parti, etc.

Dans toute perspective de traitement automatique, le problème est donc de décomposer le mot en ses différentes parties. Cette décomposition nécessite des connaissances de niveau supérieur en cas d'ambiguïtés (si le mot accepte plusieurs segmentations) [14].

3. Correction orthographique

3.1. Notions relatives à la correction orthographique

Un correcteur orthographique est un logiciel sophistiqué permettant la détection (identification), puis la correction des erreurs orthographiques trouvées dans un texte. Sa mise en place requiert l'utilisation d'un lexique et d'un analyseur morphologique. En linguistique, le lexique d'une langue constitue l'ensemble de ses lemmes ou, d'une manière plus courante mais moins précise, « l'ensemble de ses mots ». Alors que l'analyseur morphologique permet de dériver la construction morphologique d'un mot ou de faire des hypothèses sur un mot inconnu. Deux approches sont souvent utilisées en correction orthographique : l'approche basée sur un dictionnaire et l'approche stochastique. Suivant les approches, différents outils sont utilisés [15] :

3.1.1. Approche basée sur le dictionnaire

Le correcteur teste chaque mot du texte à traiter dans le lexique du correcteur orthographique, ou dans le cas échéant, vérifie s'il peut être généré à partir des mots du lexique du correcteur. Cette approche considère comme erreur toute forme qui ne corresponde à aucune forme mémorisée dans le lexique du correcteur ou générée à partir des lemmes du lexique. De ce fait, un correcteur orthographique a besoin d'un dictionnaire qui lui sert de lexique, et d'un analyseur morphologique pour vérifier les formes dérivées. Après avoir détecté l'erreur, le correcteur orthographique doit passer à la correction qui consiste à une proposition de mots du lexique du correcteur orthographique les plus proches du mot erroné identifié.

3.1.2. Approche Stochastique (analyse en n-gramme)

Dans cette approche le correcteur exploite des données statistiques obtenues à partir de corpus d'apprentissage pour détecter et corriger des erreurs. Il faut aussi noter que la notion d'erreur diffère ici de celle définie dans l'approche basée sur le dictionnaire.

Ici, il s'agit d'un seuil et tout mot ayant une probabilité d'erreur qui dépasse ce seuil est considéré comme erroné. Grâce à cette faculté de détection et de correction, les correcteurs orthographiques sont utilisés dans diverses applications par exemple dans la traduction automatique, la recherche documentaire etc.

4. Etat de l'art de la correction orthographique

4.1. Détection

4.1.1. Consultation du dictionnaire

La méthode la plus utilisée en correction orthographique est celle basée sur la consultation du dictionnaire (lexique du correcteur). Ce dernier contient l'ensemble des mots de la langue ou du moins les formes de bases. La technique consiste à repérer les mots absents du lexique du correcteur et de les marquer comme erronés. Cependant, garder tous les mots de la langue dans le lexique du correcteur peut rendre la recherche fastidieuse. Dès lors seules les formes de bases sont stockées dans le lexique et d'autres outils tels que les analyseurs morphologiques sont utilisés pour générer les formes dérivées. Un analyseur morphologique permet de dériver la construction morphologique d'un mot connu ou de faire des hypothèses sur un mot inconnu. La morphologie est traditionnellement divisée en formation des mots (morphologie dérivationnelle ou compositionnelle) et en morphologie flexionnelle (Hacken et al. 2001), (Bouillon et al. 1998), (Nazarenko 2006). La morphologie flexionnelle consiste à décliner un substantif ou un adjectif à partir d'un lexème, tandis que la morphologie dérivationnelle décrit la composition d'un mot à partir de racine et d'affixes, ou encore la création d'un mot à partir d'autres mots.

Les outils les plus utilisés pour l'analyse morphologique en correction orthographique sont les lemmatiseurs, les conjugueurs, les déclineurs qui sont généralement réalisés avec des automates.

4.1.2. Analyse par n-gramme

L'analyse en n-grammes est une technique de détection d'erreur qui permet de relever les mots incorrects dans un texte. Il ne compare pas les mots aux mots du dictionnaire mais établit un processus de comparaison au niveau d'une matrice carrée de taille n , qui stocke les n-grammes acceptés qui sont les plus fréquents (E. M. ZAMORA*, J. J. Pollock, A. ZAMORA, 1981). Chaque chaîne de caractère du texte est fractionnée dans un ensemble de n-grammes adjacents. Le développement d'un système basé sur les n-grammes suppose avant tout de calculer la probabilité d'erreur de chaque n-gramme pris individuellement.

Cette probabilité est tout simplement le nombre de fois qu'il apparaît sur des mots erronés sur le nombre total de fois qu'il apparaît sur des mots du texte. Il se peut aussi qu'une séquence de n lettres (n-gramme) soit considérée comme valide dans un contexte et comme invalide dans un autre. Cette méthode requiert donc un corpus d'apprentissage. Seul les n-grammes construits à partir des mots incorrects représentent les n-grammes invalides. Un n-gramme, qui est à la fois présent sur un mot et sur sa forme erronée, est considéré comme ne contribuant pas à l'erreur. La probabilité d'erreur (P) d'un n-gramme est donnée par la formule suivante $p = \frac{E}{E+V}$

Où :

- E est le nombre de fois où il a été classé comme un n-gramme invalide ;
- V est le nombre de fois où il a été classé comme n-gramme valide.

4.2. Correction

L'étape qui suit la détection d'erreurs dans un processus de correction orthographique est la génération de mots susceptibles d'être la forme correcte du mot mal orthographié. Les techniques de correction étudiées ci-dessous se focalisent sur le mot mais pas sur le contexte dans lequel le mot doit être utilisé. La correction s'exécute en deux étapes :

- La génération de mots susceptibles d'être la bonne orthographe.
- La génération des rangs des mots proposés dans la première étape.

La deuxième étape permet de filtrer et d'ordonner les résultats de la première étape.

4.2.1. Alpha-code

Pour retrouver les formes les plus proches du mot inconnu, la plupart des vérificateurs orthographiques utilisent une représentation codée d'un mot. Cette technique par codage de chaînes est aussi connue sous le nom d'alpha-code. L'alpha-code d'un mot est la chaîne de caractères constituée de l'ensemble de ses lettres classées par ordre alphabétique (Gardier 1992). Elle détient différentes méthodes de calcul ; (Ndiaye et al. 2003) ont proposé une méthode qui consiste à ranger les consonnes suivies des voyelles par ordre alphabétique, respectivement. L'ensemble des mots relatifs à un alpha-code donné est appelé classe de cet alpha-code.

Le système de correction dispose de l'ensemble des alpha-codes correspondant au lexique et, pour chacun, de la classe de mots.

Le correcteur établit tout d'abord l'alphacode du mot erroné. Si cet alpha-code est répertorié, les mots de la classe correspondante sont retenus comme candidats. Si l'alpha-code est inconnu, le système y ajoute un, puis deux caractères, et vérifie si le nouvel alpha-code obtenu existe.

Dans ce cas, la classe est retenue. Il procède de même en supprimant puis en substituant un ou deux caractères. Les mots candidats obtenus font alors l'objet d'un calcul de proximité avec la chaîne incorrecte, ce qui autorise un classement de cette liste de mots correctifs. Un mot ne possède qu'un seul alpha-code mais plusieurs mots peuvent partager le même alpha-code. Ainsi, un vérificateur orthographique pourra rechercher les mots qui contiennent le même alpha-code que le mot inconnu.

4.2.2. Distance lexicographique : (Levenshtein et Damerau-Levenshtein)

La distance de Levenshtein, encore connue sous le nom de « distance d'édition » ou de « déformation dynamique temporelle », est une métrique entre deux chaînes de caractères. Elle est égale au nombre minimal de caractères qu'il faut supprimer, insérer ou remplacer pour passer d'une chaîne à l'autre (Levenshtein, 1966). Cette distance est d'autant plus grande que le nombre de différences entre les chaînes est grand. La distance de (Damerau-Levenshtein, 1964) est similaire à celle de Levenshtein. C'est-à-dire que c'est une métrique qui calcule le nombre minimal d'opérations pour transformer une chaîne en une autre. Cependant celle de Levenshtein calcule la distance d'édition en considérant seulement trois opérations à savoir la substitution, l'insertion et la suppression. Or, d'après notre typologie des erreurs, une faute d'orthographe peut aussi être causée par la transposition de deux caractères successifs. C'est dans cet optique que la distance de Damerau-Levenshtein amène un apport important car non seulement il inclut les opérations citées par Levenshtein (insertion, substitution, suppression) mais ajoute à cela la transposition.

4.2.3. N-gramme

Comme indiqué précédemment, la détection d'erreur par n-gramme permet de connaître la position de l'erreur dans le mot. Cette information facilite la correction de l'erreur. La méthode de correction proposée par (David Sundby) s'appuie sur cette position pour générer une liste de suggestion pour le mot erroné.

Son algorithme est basé sur les quatre opérations d'édérations (Damerau, 1964) à savoir l'insertion, la suppression, la substitution et la transposition pour rechercher des formes correctes pouvant correspondre au mot mal orthographié.

- Insertion :

C'est un algorithme qui corrige les mots erronés avec un caractère manquant. Le principe de l'algorithme consiste à insérer une lettre de l'alphabet à la position où l'erreur s'est produite et de vérifier si le mot obtenu est correct avant de l'ajouter dans la liste des mots candidats. Le procédé est réitéré pour toutes les lettres de l'alphabet.

- Suppression :

C'est un algorithme qui supprime des caractères sur un mot erroné. L'algorithme supprime un caractère du mot et vérifie si le mot formé est correct avant de l'ajouter dans la liste de suggestion. Le procédé est réitéré pour toutes les lettres du mot.

- Substitution :

L'algorithme de substitution remplace chaque lettre du mot, les unes après les autres par une lettre de l'alphabet et vérifie si le mot obtenu est correct avant de l'ajouter dans la liste de suggestion. On répète le même procédé pour toutes les lettres de l'alphabet.

- Transposition :

L'algorithme change la position d'un caractère du mot en le mettant dans toutes les autres positions et à chaque fois il vérifie si le mot formé est correct avant de l'ajouter dans la liste de suggestion. Le procédé est réitéré pour toutes les lettres du mot. Nous avons cité, dans les sections précédentes, quelques problèmes souvent rencontrés dans le domaine de traitement automatique de la langue arabe. Nous avons vu par exemple que la langue arabe possédait un système dérivationnel très riche et c'est dans cette caractéristique que réside la difficulté de traiter cette dernière.

5. Détection et correction des fautes d'orthographe en arabe

Un système de détection et de correction d'orthographe est généralement basé sur trois composants principaux [16]:

- Un dictionnaire (ou liste de mots de référence) est un élément essentiel du correcteur d'orthographe / correcteur, car c'est la référence à partir de laquelle il est possible de décider si un mot donné est correct ou mal orthographié ;
- Un modèle d'erreur ;
- Un modèle de langage.

Un bon système de correction orthographique nécessite une répartition équilibrée du travail entre les trois principaux composants : le dictionnaire, Erreur modèle et la langue modèle.

Pour clôturer ce chapitre, Nous présentons un travail qui traite la correction orthographique de la langue arabe, ce travail réalisé par les chercheurs (Mohammed Attia, Pavel Pecina, Younes Samih, Khaled Shaalan, Josef van Genabith).

Les chercheurs ont utilisé pour la détection des erreurs deux méthodes :

- Méthode directe : Utilisent un dictionnaire (faire des améliorations utilisés la liste ara complexe externe et corpus du gicaWord.
- Méthode de modélisation basée sur les caractères dans le cas d'un tel mot.

Et pour la correction, ils ont utilisés un modèle d'erreurs : génère une liste de correction utilise un transducteur d'état finis aux distance Levenshtein, un modèle de langage, ayas pelle et n_gramme.

Conclusion

L'étude des langues et des mécanismes nécessaires à la mise en œuvre de son traitement automatique par des machines est un domaine d'études foisonnant, et riche en applications potentielles ou émergentes.

Ce chapitre a porté sur le TALN où nous avons traité d'une manière systématique les différents niveaux et paliers d'analyses, ainsi que quelques problèmes rencontrés dans le domaine de TALA qui possède un système dérivationnel très riche et c'est dans cette caractéristique que réside la difficulté de traiter cette dernière.

De plus, nous avons également présenté un état de l'art de la correction orthographique. Précisément, nous avons évoqué les notions de bases relatives à la correction orthographique, les techniques de détection et de correction d'erreurs orthographiques.

Le chapitre suivant sera consacré à l'étude des Ontologies et aux différents concepts qui y réfèrent.

Chapitre III : Ingénierie de Connaissance.

Introduction

L'exploitation de connaissances est un sujet de recherche populaire dans divers communautés qui a pour objectif de ne plus faire manipuler en aveugle des informations à la machine mais de permettre un dialogue entre le système et les utilisateurs.

Dans ce chapitre, nous allons particulièrement nous intéresser à la notion d'« Ontologie » pour laquelle nous avons opté comme une solution pour construire une base de connaissance tout en aidant à la réutilisation des données pour détecter les erreurs et les corriger.

Ainsi, seront abordés dans ce chapitre les points suivants : la définition d'Ontologie, son intérêt pour la linguistique et la correction orthographique.

1. Définition

Une ontologie est une spécification explicite d'une conceptualisation. Le terme «conceptualisation» fait référence à un système de concepts. L'expression «spécification explicite» signifie que la conceptualisation est représentée dans un langage (langue naturelle ou langage formel). Le terme «ontologie» est un emprunt à la philosophie. Il désigne (Petit ROBERT, 1979) : la partie de la métaphysique qui s'applique à l'être en tant qu'être, indépendamment de ses déterminations [17].

1.1. Pour quelles raisons développer une ontologie ?

Les raisons pour lesquelles une ontologie peut s'avérer utile sont [18] :

- Partager la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels ;
- Permettre la réutilisation du savoir sur un domaine ;
- Expliciter ce qui est considéré comme implicite sur un domaine ;
- Distinguer le savoir sur un domaine du savoir opérationnel ;
- Analyser le savoir sur un domaine.

2. Cycle de vie d'une ontologie

Puisque les ontologies sont destinées à être utilisées comme des composants logiciels dans des systèmes répondant à des objectifs opérationnels différents, leur développement doit s'appuyer sur les mêmes principes que ceux appliqués en génie logiciel.

Ainsi, les ontologies doivent être considérées comme des objets techniques évolutifs et possédants un cycle de vie qui nécessite d'être précisé. Dans ce contexte, les activités liées aux ontologies sont, d'une part, des activités de gestion de projet (planification, contrôle, assurance qualité), et d'autre part, des activités de développement (spécification, conceptualisation, formalisation) ; s'y ajoutent des activités transversales de support telles que l'évaluation, la documentation, la gestion de la configuration.

La figure 3.6 représente une vision d'un cycle de vie d'une ontologie inspiré du cycle de vie du génie logiciel.

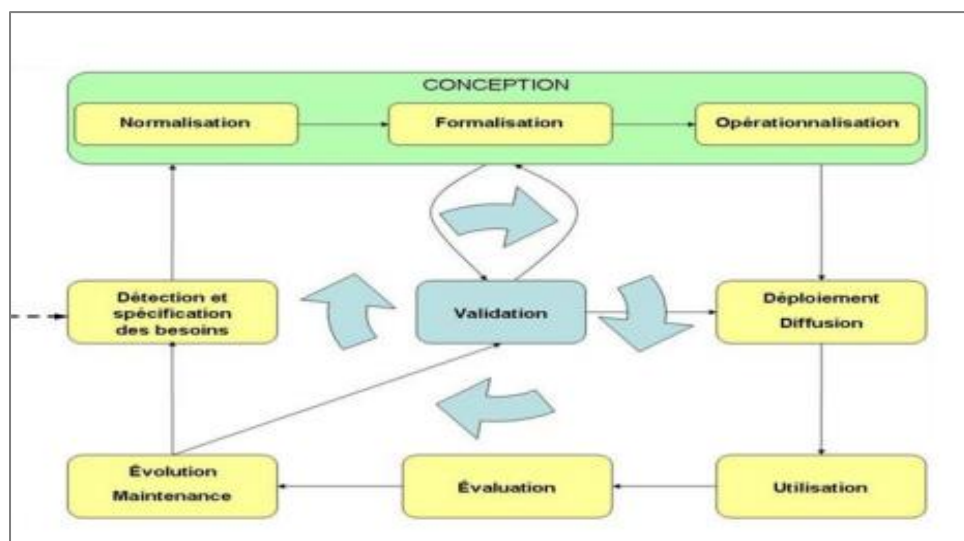


Figure 3.6 : Cycle de vie d'une Ontologie.

Ce cycle de vie comprend une étape initiale de détection et de spécification des besoins qui permet notamment de circonscrire précisément le domaine de connaissances. Une étape de conception qui se subdivise en quatre phases :

- Une étape de déploiement et de diffusion ;
- une étape d'utilisation ;
- une étape incontournable, d'évaluation ;
- et enfin, une étape consacrée à l'évolution et à la maintenance du modèle.

Après chaque utilisation significative, l'ontologie et les besoins doivent être réévalués et l'ontologie peut être étendue et, si nécessaire, en partie reconstruite. La validation du modèle de connaissances est au centre du processus et se fait de manière itérative [19].

3. Les types d'Ontologie

On distingue six types d'ontologie [20]:

3.1.Ontologie de représentation de connaissances

Ce type d'ontologie modélise les représentations primitives utilisées pour la formalisation des connaissances sous un paradigme donné. Par exemple, une ontologie sur le formalisme des Topic Maps comportera les concepts : Topic, Type de Topic, Association, Occurrence, Type occurrence,...

3.2.Ontologie de haut niveau / supérieure

Elle décrit des concepts très généraux comme l'espace, le temps, la matière, les objets, les événements, les actions, etc. Ces concepts ne dépendent pas d'un problème ou d'un domaine particulier, et doivent être, du moins en théorie, consensuels à de grandes communautés d'utilisateurs.

Ce type d'ontologies est fondé sur la théorie de la dépendance. Son sujet est l'étude des catégories des choses qui existent dans le monde. Comme les concepts de haute abstraction tels que les entités, les événements, les états, les processus, les actions, le temps, l'espace, les relations, les propriétés...

3.3.Ontologie Générique (Generic Ontology)

Elle est appelée également noyau ontologique, modélise des connaissances moins abstraites que celles véhiculées par l'ontologie de haut niveau mais assez générales néanmoins pour être réutilisées à travers différents domaines. Cette ontologie inclut un vocabulaire relatif aux choses, événements, temps, espace, causalité, comportement, fonction, etc.

3.4.Ontologie du domaine (Domain ontology)

Cette ontologie exprime des conceptualisations spécifiques à un domaine. Elle fournit les concepts et les relations permettant de couvrir les vocabulaires, activités et théories de ces domaines. Selon Mzoguchi.

L'ontologie du domaine caractérise la connaissance du domaine ou la tâche est réalisée. Par exemple, dans le contexte du e-Learning, le domaine peut être celui de formation.

C'est ce type qui nous sera utile pour définir les objectifs ciblés de notre projet.

3.5.Ontologie de Tâches (Task ontology)

L'ontologie de tâches fournit un vocabulaire systématisé des termes employés pour résoudre des problèmes liés aux tâches qui peuvent être ou non du même domaine. Elle fournit un ensemble de termes au moyen desquelles nous pouvons décrire généralement comment résoudre un type de problèmes. Elle inclut des noms, des verbes et des adjectifs génériques dans les descriptions de tâches.

3.6.Ontologie d'application (Application ontology)

C'est l'ontologie la plus spécifique, elle contient des concepts dépendants d'un domaine et d'une tâche particuliers, elle est spécifique et non réutilisable. Ces concepts correspondent souvent aux rôles joués par les entités du domaine lors de l'exécution d'une certaine activité. Il s'agit donc ici de mettre en relation les concepts liés à une tâche particulière de manière à en décrire l'exécution.

4. Composantes d'une ontologie

Les ontologies fournissent un vocabulaire commun d'un domaine et définissent la signification des termes et des relations entre elles. La connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de composants à savoir : concepts (ou classes), relations (ou propriétés), fonctions, axiomes (ou règles) et instances (ou individus) [17].

- Les concepts aussi appelés termes ou classe de l'ontologie, correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème) retenus en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie.
- Les relations traduisent les associations (pertinentes) existant entre les concepts présents dans le segment analysé de la réalité. Ces relations incluent les associations suivantes :
 - Sous classes de (généralisation-spécialisation) ;
 - Partie de (agrégation ou composition) ;
 - Associe à ;
 - Instance de, etc.

Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres.

- Les fonctions constituent des cas particuliers de relations, dans laquelle un élément de la relation, (le nième) est défini en fonction des N-1 éléments précédents ;
- Les axiomes constituent des assertions, acceptées comme vraies, à propos des abstractions du domaine traduites par l'ontologie.
- Les instances constituant la définition extensionnelle de l'ontologie. Ces objets véhiculent les connaissances (statiques, factuelles) à propos du domaine du problème.

5. Méthode de recherche orientée ontologie

Nous envisageons une méthodologie en quatre grandes phases :

5.1.Phase 1 : l'étude de faisabilité

Cette phase permet d'effectuer une analyse préliminaire. Il s'agit d'effectuer une étude d'opportunité et d'anticiper les difficultés. Par exemple, on identifiera le problème et les solutions potentielles

5.2.Phase 2 : Modélisation ontologique

La modélisation cognitive est orientée ontologie. Tout au long du processus. Cette phase comprend cinq étapes dont certaines peuvent avoir lieu en même temps, mais qui pour des raisons de clarté sont présentées dans l'ordre suivant :

- Étape 1 : Capture des besoins

Cette étape permet de définir un cadre de référence pour l'implémentation et l'évaluation de l'ontologie. Il s'agit de spécifier les besoins propres à la modélisation ontologique. À cette étape, on définit :

- a. Le domaine qui sera représenté par l'ontologie ;
- b. Le contexte d'utilisation de l'ontologie ;
- c. Le but de l'application exploitant cette ontologie ;
- d. Les caractéristiques de l'application exploitant l'ontologie ;
- e. Les directives de conception de l'ontologie ;
- f. Les sources d'information ;
- g. Les cas d'utilisation ;
- h. Les questions de compétences.

- Étape 2 : Analyse des sources d'information

À partir de cette étape, la démarche est itérative et prend fin lorsque l'on obtient un niveau de complétude ou de détail satisfaisant. L'analyse des sources consiste à :

- a. Analyser les questions de compétences pour trouver des termes (concepts et les relations) appropriés ;
- b. Compléter l'analyse des questions en explorant les scénarios d'utilisation ;
- c. Colliger les réponses aux questionnaires de compétences résultants ;
- d. Identifier les termes candidats à partir des questionnaires de compétence. Nous obtenons une liste de termes parmi lesquels se trouvent des concepts, des relations, et le cas échéant des propriétés de concepts. Ces termes seront retenus ou non, selon leur potentiel à représenter correctement le domaine.

- Étape 3 : Construction de l'ontologie initiale

- Étape 3.1 : Conceptualisation de l'ontologie

Cette étape consiste à mettre en relation les concepts « primitifs » pour former les taxonomies de base. Le domaine est articulé sans tenir compte des contraintes informatiques. À ce stade, la modélisation étant semi-informelle, une spécification en langage naturel complétée de graphiques est acceptable et des définitions minimales des concepts sont faites. On se concentre sur :

- a. La construction des taxonomies de base (concepts, relation taxinomique). La construction d'une taxonomie repose sur la mise en relation d'un concept général (ou abstrait) à un concept plus spécifique (ou concret) au moyen d'un lien de généralisation-spécialisation ou « Is-a » ou encore « subclass-of » (termes synonymes) ;
- b. L'ajout des rôles (propriétés, attributs) et les restrictions (contraintes de type et valeur).

- Étape 3.2 : Semi-formalisation de l'ontologie

Cette étape consiste à formaliser partiellement l'ontologie sans perdre d'expressivité. On se concentre surtout sur :

La formation de relation non taxinomique et la création de règles (ou axiomes) pour régir les taxonomies.

Une règle est composée d'un ensemble initial de concepts et relations (l'hypothèse), et d'un ensemble final de concepts et relations (la conclusion).

- Étape 4 : Raffinement de l'ontologie

- Étape 4.1 : Correction de l'ontologie conceptuelle

À cette étape, on fera un back-up de l'ontologie initiale, on corrigera le modèle conceptuel de l'ontologie en fonction des résultats de la détection d'erreurs, on documentera toutes les modifications, et on recompile le modèle formel.

Voici quelques modifications typiques qui pourront être apportées :

- a. Normalisation des noms (de classes et instances) ;
- b. Spécialisation de concepts (application du principe de distinction ontologique) ;
- c. Subdivision de taxonomies ;
- d. Inclusion de nouvelles propriétés et changements aux propriétés existantes ;
- e. Minimiser la distance sémantique entre les concepts enfants de mêmes concepts parents.

- Étape 4.2 : Validation (voir méthode d'évaluation)

La validation sert à s'assurer que l'ontologie modélise vraiment le monde réel (domaine) pour lequel l'ontologie a été créée. Le but est de prouver la conformité du modèle du monde (s'il existe et est connu) à partir du monde modélisé formellement.

- Étape 4.3 : Raffinement de l'ontologie

Le raffinement de l'ontologie consiste essentiellement à appliquer (En réponse à l'évaluation) les principes de cohérence, de complétude et de précision sans affecter l'intégrité de l'ontologie.

- Étape 5 : Formalisation de l'ontologie

À cette étape l'ontologie sera formalisée au moyen d'un langage interprétable par une machine. De même, les relations et les axiomes sont également traduits en langage machine.

5.3.Phase 3 : Mise en œuvre informatique de l'ontologie

- Étape 6 : Capture des besoins et analyse additionnels

Cette étape consistera à capturer les besoins fonctionnels et techniques propres au génie logiciel et à les analyser par rapport aux spécifications de l'ontologie. Notamment, il s'agit d'identifier et de documenter des scénarii pour exploitation de l'ontologie, en termes de services et fonctions rendus par l'agent qui l'exploitera. Les scénarios les plus viables seront retenus.

- Étape 7 : Opérationnalisation de l'ontologie

Cette étape consistera à encoder l'ontologie au moyen d'un langage de programmation afin de la rendre exécutable. ((Mizoguchi R. & al. 1995); (Breuker J. & de Velde W.V., 1994); (Seta K. & al., 1997); (Chandrasekaran B., Josephson J. R., & Benjamins R., 1998)). L'ontologie est dite opérationnelle puisque le prototype appelé CIAO, qui l'exploitera sera développé à partir des codes abstraits associés aux définitions de concepts de l'ontologie. Cependant, ce sont les modèles de l'ontologie qui sont en réalité opérationnalisés.

- Étape 8 : Prototypage

Cette étape consistera à implanter les services et fonctions rendus dans l'agent CIAO. Il s'agira aussi de concevoir l'interface de communication avec l'ontologie.

- Étape 9 : Migration de système auteur hôte

Cette étape consistera à effectuer les adaptations nécessaires pour que le système auteur hôte communique adéquatement avec le prototype CIAO.

- Étape 10 : Déploiement du prototype dans le système hôte

Cette étape consistera à implanter CIAO dans la plateforme hôte, à la suite de la migration de l'environnement.

5.4.Phase 4 : Méthode d'évaluation

La méthode de validation se divise en deux thèmes, la validation syntaxique et la validation sémantique.

La validation syntaxique a pour but d'assurer que tous les éléments du modèle semi-formel sont représentés dans l'ontologie selon les règles du langage ontologique utilisé, dans OWL-DL. La réalisation de cette validation se décompose en deux étapes, génération d'un modèle semi-formel à partir de l'ontologie de domaine et comparer les éléments de ce modèle.

La génération avec les éléments de modèle semi-formel d'origine. Si la concordance est exacte, alors il est véridique de conclure que l'ontologie du domaine représente syntaxiquement l'ensemble des éléments du modèle semi-formel du domaine.

La validation sémantique s'attarde sur le sens des représentations qui sont consignées dans l'ontologie. À titre d'exemple, une erreur occasionnellement commise par les concepteurs de modèles peu familiers au langage MOT est issue d'une confusion entre le lien S ("sorte de" [is-a]) et le lien C ("composé de" [part-of]). Par exemple, ils pourraient être tentés d'exprimer que planète se compose de vénus, terre, mars, etc., au lieu d'exprimer que vénus, terre... sont des sortes de planètes. Une inférence sur l'ontologie de domaine qui déterminerait que la propriété inverse de composeDe est faitPartieDe permettrait de conclure, que vénus, terre, mars... font partie de planète. Ce type de conclusion erronée devrait servir de signal à l'ingénieur et aux experts concernant une possible erreur de sémantique dans la représentation du domaine.

La première étape à réaliser pour une validation sémantique est la production de conclusions formelles en appliquant des inférences sur l'ontologie du domaine. Des scénarios de test de déductions peuvent être utilisés afin de formaliser d'avantage l'étape. L'étape de comparaison des conclusions entre les réponse qu'auraient données l'ingénieur et l'expert aux déduction et les conclusions automatiques permettent d'entreprendre une réflexion qui peut , d'une part, amener l'expert de contenu à revoir la façon de représenter ce qu'il exprime, ou d'autre part, mettre fin

au processus de construction de l'ontologie du domaine. Gómez-Pérez (2004) présente les critères selon lesquels l'ontologie devrait être validée:

la consistance identifie s'il y a des contradictions entre éléments ontologiques; la complétude assure que tous les éléments ontologiques sont soit explicitement déclarés ou soit insérables; la concision est un principe qui stipule que seuls les éléments à être définis doivent être définis; l'expansibilité est la capacité d'ajouter des nouvelles connaissances sans modifier les anciennes; la sensibilité est la capacité à réagir à des modifications [21].

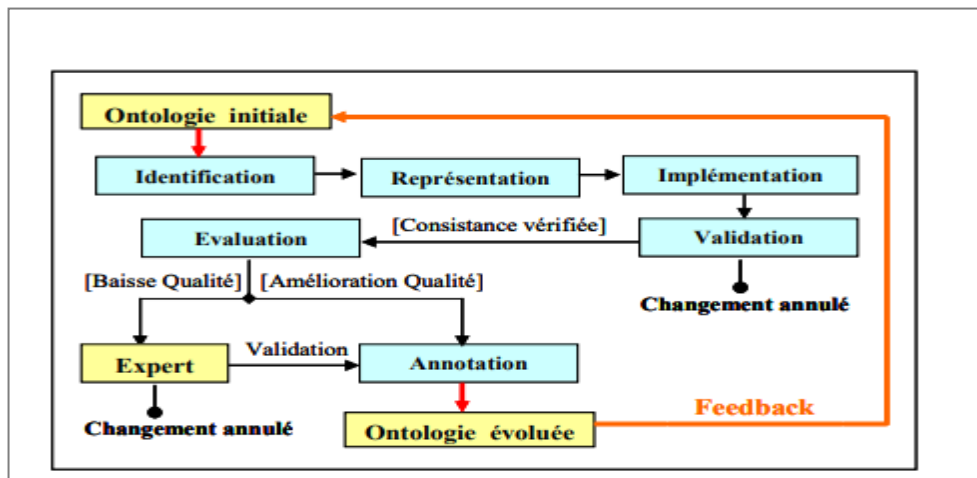


Figure 3.7: Validation et Evaluation d'Ontologie.

6. Les méthodes d'ingénierie ontologique

6.1.Méthode de Bachimont

Cette méthode propose de contraindre l'utilisateur à un engagement sémantique en introduisant une normalisation sémantique des termes manipulés dans l'ontologie. La méthode de normalisation suit trois étapes

- Normalisation sémantique :

L'utilisateur doit choisir les termes du domaine et les normaliser en explicitant leurs propriétés et en exprimant les identités et les différences dans leur voisinage proche. La place d'une notion dans l'ontologie doit être justifiée par rapport à la communauté et la différence avec le père et la fratrie.

- Formalisation des connaissances :

Cette étape consiste à désambiguïser les notions de l'ontologie référentielle obtenue par l'étape précédente et choisir leurs sens pour un domaine spécifique. Cela peut nécessiter la création de nouveaux concepts, l'ajout de propriétés et d'axiomes.

- Opérationnalisation des connaissances :

Le système utilise un langage opérationnel de représentation de connaissances qui possède les caractéristiques nécessaires pour répondre aux besoins exprimés lors de la spécification du système.

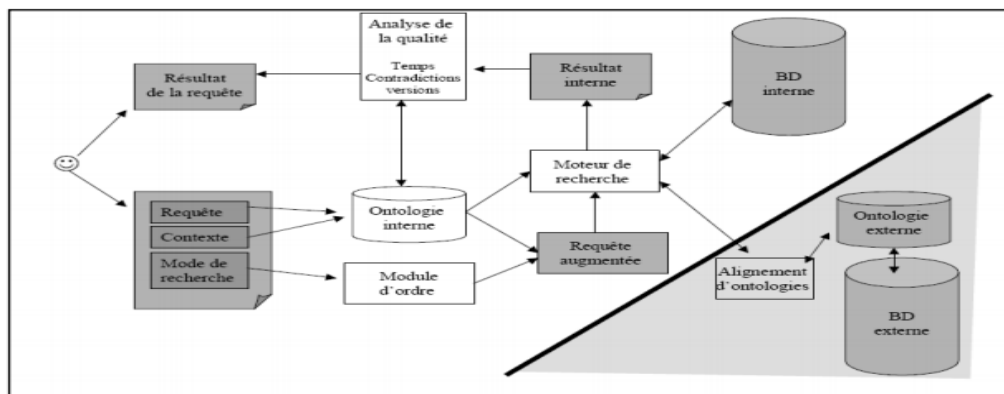


Figure 3.8: Processus de la méthode Bachimont.

6.2. La méthode Methontology

C'est une méthodologie mise au point par l'équipe du laboratoire de l'intelligence artificielle de l'Université polytechnique de Madrid.

Cette méthode inclut :

- L'identification du processus de développement de l'ontologie.
- Le cycle de vie basé sur l'évolution de prototypes.
- Les techniques de gestion de projet (planification, assurance qualité), de développement (spécification, conceptualisation, formalisation, implémentation, maintenance) et des activités de support (intégration, évaluation, documentation).

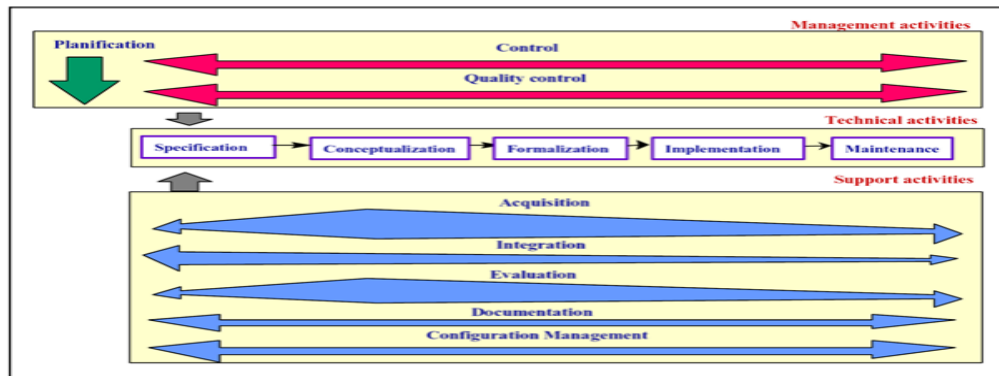


Figure 3.9: Méthode de METHONTOLOGY.

7. Les outils de construction d'ontologie

On distingue deux familles d'outils : les outils de construction d'ontologie dépendants de formalisme de représentation et les outils de construction d'ontologie indépendants de formalisme de représentation.

7.1. Les outils dépendants de formalisme de représentation

- Ontolingua:

Ontolingua est un serveur d'édition d'ontologies. Il utilise des classes, des relations, des fonctions, des instances et des axiomes pour décrire une ontologie. Une relation peut contenir des propriétés nécessaires (contraintes) ou nécessaires et suffisantes qui définissent la relation. En plus le serveur Ontolingua offre la possibilité d'intégrer les ontologies Ontolingua, ce qui permet la construction modulaire des ontologies.

- OntoSaurus :

OntoSaurus de l'Information Science Institute de l'Université de Southern California est composé de deux modules : un serveur utilisant LOOM comme langage de représentation des connaissances, et un serveur de navigation créant dynamiquement des pages HTML qui affichent la hiérarchie de l'Ontologie ; le serveur utilise des formulaires HTML pour permettre à l'utilisateur d'éditer l'ontologie.

- WebOnto :

WebOnto du Knowledge Media Institute de l'Open University, est une application Web pour naviguer et développer collaborativement les ontologies. Il supporte la navigation collaborative, la création et l'édition d'ontologies sur le Web. Les ontologies WebOnto sont implémentées dans le langage OCML. WebOnto distingue quatre types d'ontologies : ontologie de domaine, ontologie de tâche, ontologie de méthode, et ontologie d'application.

- OilEd:

OilEd (Oil Editor) est un éditeur d'ontologies utilisant le formalisme OIL. Il est essentiellement dédié à la construction de petites ontologies dont on peut ensuite tester la cohérence à l'aide de FACT, un moteur d'inférences bâti sur OIL.

7.2. Les outils indépendants de formalisme de représentation [22]

- Protégé 2000 :

Est une interface modulaire permettant l'édition, la visualisation, le contrôle d'ontologie, l'extraction d'ontologies à partir de sources textuelles, et la fusion semi-automatique d'ontologies. Le modèle de connaissances sous-jacent à protégé 2000 est issu du modèle des frames et contient des classes, des slots (propriétés) et des facets (valeurs des propriétés et contraintes), ainsi que des instances des classes et des propriétés. Il autorise la définition de méta-classes, dont les instances sont des classes, ce qui permet de créer son propre modèle de connaissances avant de bâtir une ontologie.

- ODE et WebOde :

L'outil ODE (Ontology Design Environment) permet de construire des ontologies au niveau connaissance, comme le préconise la méthodologie METHONTOLOGY. L'utilisateur construit son ontologie dans un modèle de type frame, en spécifiant les concepts du domaine, les termes associés, les attributs et leurs valeurs, les relations de subsumption.

- Onto Edit :

Onto Edit (Ontology Editor) est également un environnement de construction d'ontologies indépendant de tout formalisme. Il permet l'édition des hiérarchies de concepts et de relations et l'expression d'axiomes algébriques portant sur les relations, et de propriétés telles que la généralité d'un concept. Des outils graphiques dédiés à la visualisation d'ontologies sont inclus dans l'environnement. Onto Edit intègre un serveur destiné à l'édition d'une ontologie par plusieurs utilisateurs. Un contrôle de la cohérence de l'ontologie est assuré à travers la gestion des ordres d'édition.

8. Les notions relatives aux ontologies dans le domaine linguistique

Il existe deux grands types d'ontologies dans le domaine linguistique : les ontologies linguistiques et les ontologies métalinguistiques.

Il est nécessaire d'examiner ces ontologies, dans la mesure où elles proposent des descriptions formelles des notions de référence d'un domaine (ici la linguistique).

8.1.Ontologies linguistiques

Les ontologies linguistiques sont les plus courantes. Il s'agit d'ontologies servant à décrire le vocabulaire d'une langue. Elles sont plus particulièrement destinées à être utilisées dans une perspective de Traitement Automatique de la Langue (TAL).

Le WordNet de l'Université Princeton (Felbaum, 1997) en est sans doute la plus connue. Il s'agit d'une base de données lexicales de l'anglais (appelée ontologie à cause de la structure hiérarchique à laquelle elle répond), qui comprend différents types d'entités lexicales (mots-composés, collocations, locutions), mais dont l'unité lexicale constitue l'unité linguistique principale de description. Sa structure n'est cependant pas organisée autour des unités lexicales individuelles. En effet, la base de données est organisée principalement à partir de la relation d'hyper-/hyponymie connectant non les unités lexicales, mais des regroupements d'unités lexicales synonymes, appelés synsets. Chaque synset fonctionne comme unité de structuration de l'ontologie, et représente plus qu'un mot lui-même ; dans une perspective cognitive, le synset s'approche d'une unité psycholinguistique de raisonnement. En ce sens, WordNet peut bel et bien être considéré comme une ontologie, dans la mesure où des concepts (et pas seulement des « mots ») y sont explicitement représentés.

8.2.Ontologies métalinguistiques

Les ontologies du second type, que l'on pourrait qualifier de « métalinguistiques », ne décrivent pas exactement les langues elles-mêmes, mais en précisent les concepts descriptifs, concepts qui rendent compte de la structure de la langue à l'un ou l'autre de ses niveaux de fonctionnement (lexical, syntaxique, morphologique, phonologique). Certaines des ontologies métalinguistiques existantes se rattachent à des ontologies dites « supérieures ».

C'est le cas de l'ontologie GOLD (General Ontology for Linguistic Description), la première ontologie construite spécifiquement pour la description linguistique sur le Web sémantique. GOLD se rattache à une ontologie de type supérieur, SUMO (Farrar & Langendoen, 2003), qui fait état d'un certain nombre de catégories, en distinguant par exemple les concepts « physique » et « abstrait » de même que les concepts « objet » et « processus ». Elle fournit des axiomes qui permettent de régulariser le comportement et les propriétés de ces entités sous la forme de règles d'inférence [23].

Dans ce contexte, nous avons jugé utile de présenter un travail pertinent, qui comme nous s'intéresse à la correction des erreurs syntaxique dans les écrits arabe en utilisant une approche basée sur les Ontologies.

Ce travail est celui de « Chouaib MOUKRIM », « Tarik ALMALKI », « Abderrahim TRAGHA », « El Habib BENLAHMER ».

L'approches des auteurs est basée sur la génération de phrases pour le système de correction des erreurs syntaxiques, basé sur la génération automatique des phrases arabes. Tout d'abord, ils extrayant les mots de la phrase concernée. Ensuite à partir de ces mots et grâce à une description logique des règles de grammaire arabe par l'ontologie, ils génèrent toutes les phrases possibles. Ils comparent la phrase originale avec les phrases générées afin de détecter l'erreur. Puis nous passons à la phase de correction.

Dans le cas où le système n'a pas trouvé une phrase qui ressemble à la phrase originale, les phrases alternatives correctes sont automatiquement proposées résultat du ce travail. Ils ont réalisé la phase de segmentation, l'étape de catégorisation et les relations syntaxiques entre les mots arabes. Ce travail n'en est encore qu'à ses débuts. Ils comptent dans leurs prochains travaux compléter la phase de génération, ainsi que la phase de détection et de correction des erreurs [24].

Conclusion

L'approche de correction orthographique basée sur les Ontologies est une nouvelle approche de la détection et la correction automatique des erreurs syntaxiques dans les textes arabes.

Dans ce chapitre, nous avons présenté les différentes notions relatives aux Ontologies ainsi que l'ontologie comme une solution à adopter pour la construction d'une base de connaissances concernant les erreurs de type orthographique ainsi que la réutilisation de ses connaissances dans les autres fonctionnalités de la plateforme.

Chapitre IV : Conception et Modélisation de la Solution.

Introduction

Dans ce chapitre nous aborderons une description générale de notre système ensuite nous organiserons nos besoins et mettrons en évidence le côté conceptuel de notre système qui constitue une étape fondamentale avant le passage à son implémentation en permettant de détailler les différents diagrammes et scénarios à implémenter dans la phase suivante.

Cette étape va nous permettre d'élaborer une typologie de documents produits dans l'Académie et viser les différentes contraintes techniques et fonctionnelles.

A titre de rappel, l'objectif de notre travail est de concevoir un système qui récupère les documents et les informations à partir la GED pour pouvoir y apporter des corrections et les stocker dans une base de données textuelle ainsi qu'une Ontologie pour des besoins de corrections et une meilleures prise en charge des erreurs futures. Pour se faire, nous avons utilisé une démarche de conception basée sur le langage UML.

1. Présentation de la structure d'accueil

Le projet de la Dhakhira al-Arabiyya a pour objet de constituer une base de données textuelles automatisée afin de vulgariser, de façon étendue et profonde le patrimoine arabe ainsi que la production intellectuelle arabe contemporaine et le rendre accessible à tous.

Dans ce contexte, des cellules de numérisation manuelle sont constituées pour numériser des textes rédigés en langue arabe. Ces textes sont par la suite corrigés et validés pour être introduits dans la base de données de la Dhakhira al-Arabiyya.

Afin d'améliorer le rendement de ces cellules, il est nécessaire de les dotées d'un système informatisé d'alimentation et de suivi et de supervision de la chaine de numérisation.

A cet effet, une observation de près du mode de fonctionnement actuel de la cellule de numérisation de l'Académie ainsi que des entretiens avec les principaux intervenants dans le travail de cette cellule a été nécessaire. Cela nous a permis essentiellement de relever les difficultés auxquelles sont confrontées ces différents intervenants et de bien cerner leurs attentes dans le cadre de l'objectif de notre travail.

Nous signalons essentiellement que certaines tâches effectuées de façon quotidienne se verront facilitées et améliorées en termes de gain de temps et d'efforts grâce à l'utilisation du système GED préconisé dans notre solution ainsi que l'amélioration de la qualité du travail grâce au système d'aide à la décision ainsi que l'amélioration de la correction orthographique.

2. Objectifs

Nous devons préciser les attentes particulières des utilisateurs, concernant le système à mettre en place. Pour y parvenir, les objectifs ci-dessous ont été fixés :

- Etudier l'existant en matière d'organisation des documents, analyser les problèmes vécus, et attribuer des modèles de données propres à l'établissement.
- Proposition d'un système informatisé pour faciliter le travail de la cellule de numérisation et améliorer son rendement.
- Choix du système répondant le plus aux fonctionnalités demandées et permettant une adaptation aux besoins.
- Analyse et automatisation des processus métiers.
- Proposer une solution optimale qui facilite les différentes tâches de gestion des documents : (Gestion de flux, Stockage des documents, Génération des rapports).
- Utiliser un outil d'aide à la correction des erreurs basé sur la modélisation du savoir permettant la détection de l'erreur et la proposition d'une forme corrigée en appliquant une règle donnée.
- Construire une base de connaissance « Ontologie » pour stocker les informations générées pour améliorer la recherche du correcteur orthographique.

3. Spécification des besoins fonctionnels

Le projet consiste à mettre en place une solution GED. En premier lieu, elle doit permettre de modéliser les processus métiers des utilisateurs du système, permettant aux utilisateurs de saisir les documents.

En deuxième lieu, un correcteur orthographique qui va être utilisé pour la correction des documents saisis par les utilisateurs.

Ainsi, qu'un système qui va stocker toutes les informations concernant les utilisateurs, les livres saisis et les erreurs faites. Ce système peut être utilisé comme un outil d'aide à la décision.

Le système comporte un comparateur de documents (versions avant et après correction) permet de déterminer le nombre d'erreurs corrigés, d'élaborer un rapport général reprenant toutes les erreurs avec catégorisation (forme erronée, forme corrigée, type et catégorie). Le système permet d'effectuer des sauvegardes dans la BDD et l'Ontologie.

Enfin, les sauvegardes dans l'Ontologie permettent la création de nouvelles instances.

3.1.Besoins des utilisateurs

- S'authentifier avec un compte au serveur Alfresco. Si l'utilisateur ne possède pas déjà un compte Alfresco, il doit contacter l'administrateur pour lui créer un compte.
- Lister les règles de classement et leurs propriétés.
- Saisir ou corriger un ou plusieurs ouvrages.
- Manipuler les documents (modification, correction...).
- Consulter l'espace personnel via le navigateur web.

3.2.Besoins des administrateurs

- Gérer les règles de classement (Création, Modification, Suppression, Correction).
- Gérer les processus métiers et tâches (workflow).
- Gérer les modèles de données (Types, aspects, des métadonnées).
- Ajouter les listes des ouvrages et utilisateurs.
- Affecter des types de comptes aux utilisateurs.

4. Spécification des besoins non fonctionnels

En plus des besoins fonctionnels, plusieurs considérations et contraintes additionnelles doivent être prises en compte au cours de la réalisation de notre projet. Ces besoins se résument au niveau de :

- L'interface utilisateur simple et facile à utiliser qui montre proprement les fonctionnalités du système.
- La montée en charge en particulier dans le nombre d'utilisateurs.
- La fourniture d'un manuel d'utilisation décrivant les fonctionnalités de l'application.
- La portabilité de la solution avec les différentes plateformes (Linux, Windows, Mac).

5. Conception globale

A cette étape nous traitons la conception préliminaire qui consiste en l'intégration du modèle d'analyse fonctionnelle dans l'architecture technique.

Ensuite, nous allons passer à la conception détaillée qui décrit la façon de réaliser chaque objet du système.

5.1.Processus métier

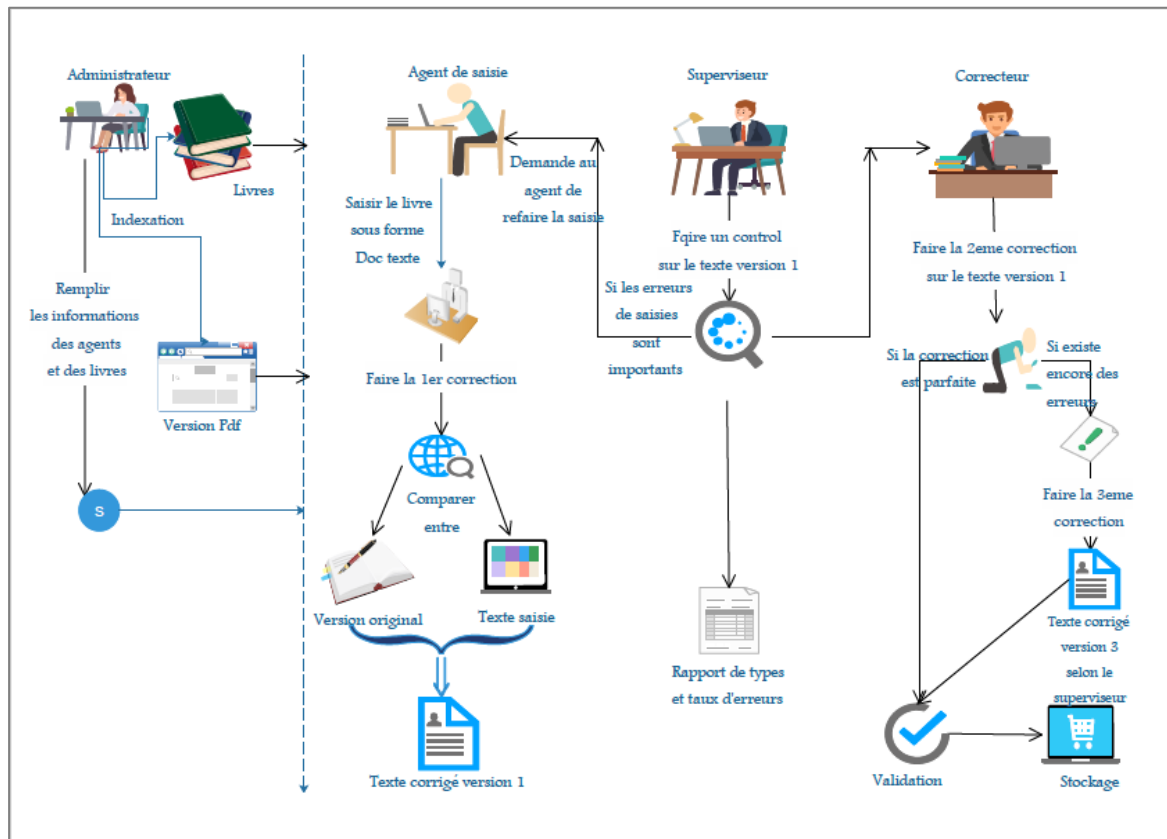


Figure 4.10 : Processus Métier

Ce diagramme décrit exhaustivement les exigences fonctionnelles du système. Il se compose de cinq tâches et englobe les fonctionnalités essentielles du système. Nous avons identifié les cinq tâches suivantes :

- Tâche 1 : Gestion Administrative.
- Tâche 2 : Gestion de saisie.
- Tâche 3 : Gestion de contrôle.
- Tâche 4 : Gestion de correction.
- Tâche 5 : Gestion des archives.

5.2. Schéma d'architecture

5.2.1. Tâche « Gestion Administrative »

La figure qui suit représente : Le processus de la tâche « Gestion Administrative ».

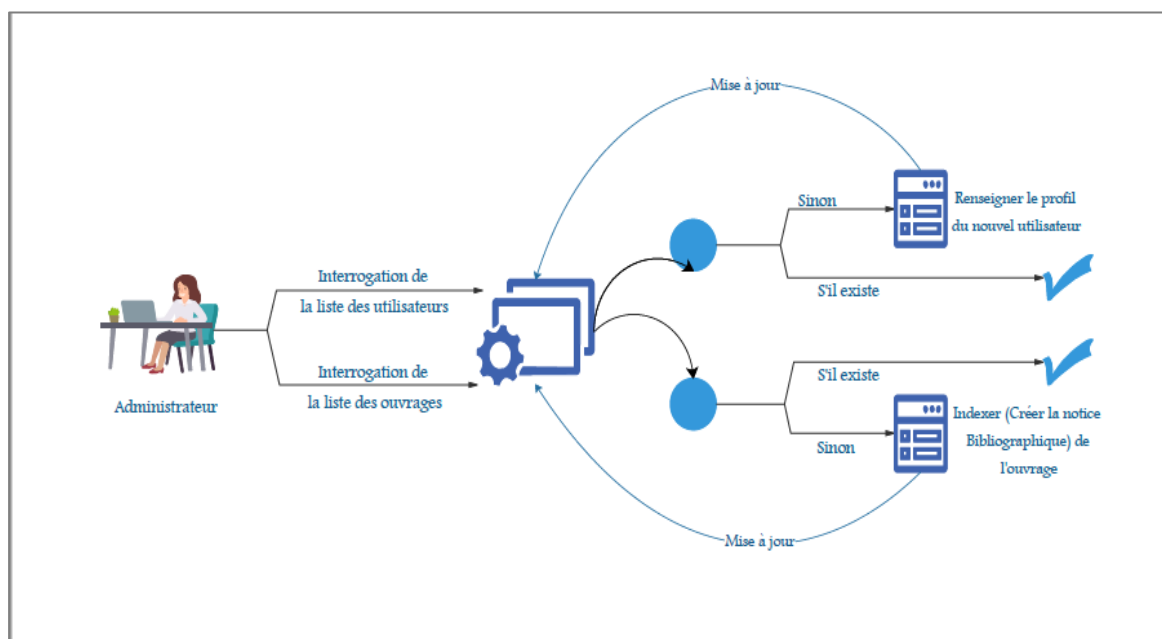


Figure 4.11: Tâche « Gestion Administrative ».

Cette illustration est détaillée dans le tableau suivant :

Tâche N° 1	Gestion Administrative
Résumé	Cette tâche permet de créer, modifier et supprimer des utilisateurs et des livres.
Acteur	Assistante Administrative.
Scénario nominal	L'administrateur consulte la page de gestion des utilisateurs et la gestion des livres sur l'interface d'Alfresco. Le système affiche la page, L'administrateur remplit les champs du formulaire, Le système vérifie les nouvelles données saisies et met à jour la base de données, enfin, un message de confirmation sera affiché.

Tableau 4.1 : Description de la tâche de la Gestion Administrative.

5.2.2. Tâche « Gestion de saisie »

La figure qui suit représente : Le processus de la tâche « Gestion de saisie ».

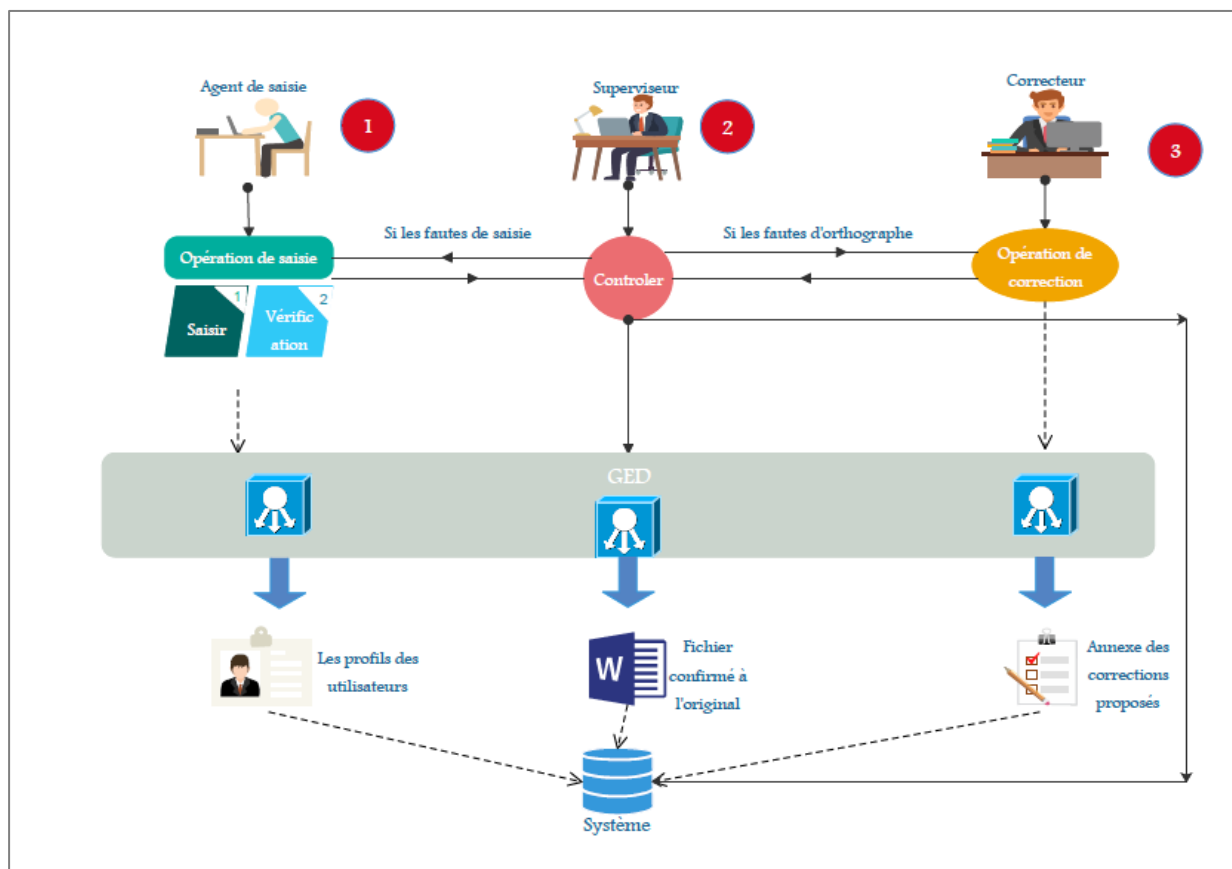


Figure 4.12: Tâche « Gestion de saisie »

Cette illustration est détaillée dans le tableau suivant :

Tâche N° 2	Gestion de Saisie
Résumé	Cette tâche permet de saisir et corriger les livres.
Acteur	Agent de saisie. Correcteur.
Scénario nominal	L'agent saisi les livres affectés dans un fichier Word et fait la correction primaire dans un autre fichier. Le système enregistre les deux versions. Le correcteur corrige les fichiers saisis par l'agent et fait une deuxième correction.

Tableau 4.2: Description de la tâche de Saisie.

5.2.3. Tâche « Gestion de contrôle »

La figure qui suit représente : Le processus de la tâche « Gestion de contrôle ».

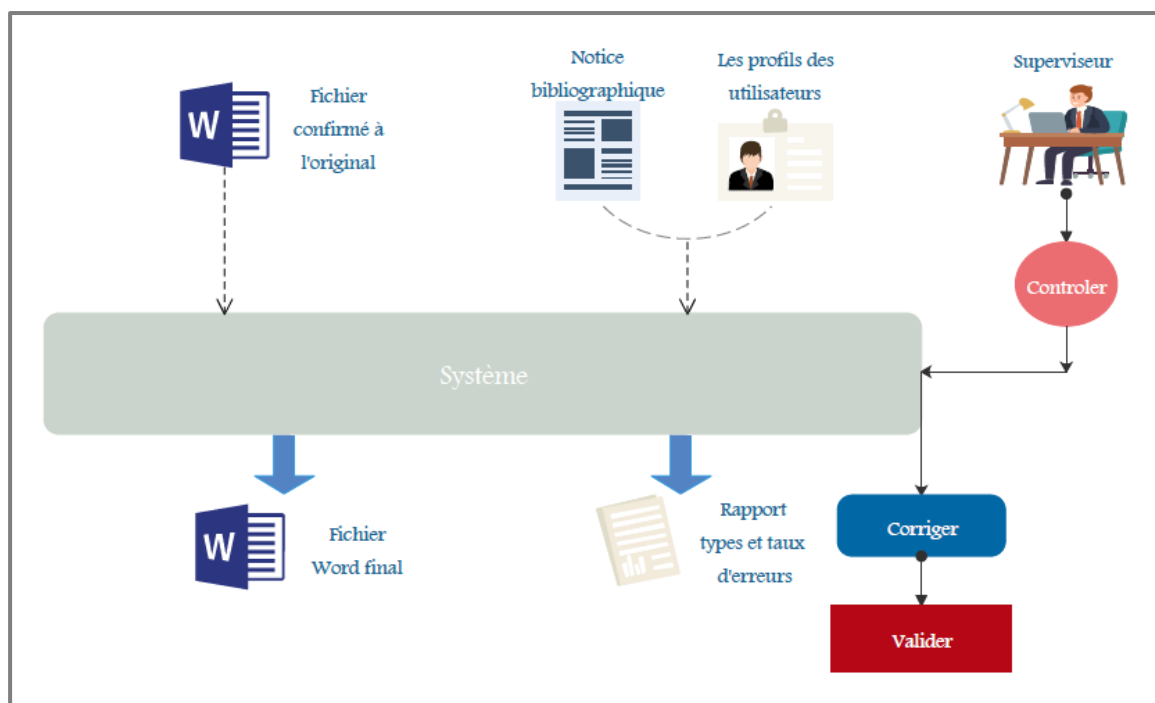


Figure 4.13: Tâche « Gestion de contrôle »

Cette illustration est détaillée dans le tableau suivant :

Tâche N° 3	Gestion de Contrôle
Résumé	Cette tâche permet de faire un contrôle sur les fichiers enregistrés dans la GED en utilisant le système pour aider le superviseur.
Acteur	Superviseur.
Scénario nominal	Le superviseur utilise le système pour faire un contrôle sur les fichiers saisis par l'Agent de saisie et le fichier corrigé par le correcteur. Après avoir fait la comparaison, le système nous gère un fichier Word final et un rapport de types et taux d'erreurs. Pour finir, le superviseur va valider le fichier final sinon il va faire une autre correction.

Tableau 4.3 : Description de la tâche de Contrôle.

5.2.4. Tâche « Gestion de correction »

La figure qui suit représente : Le processus de la tâche « Gestion de correction ».

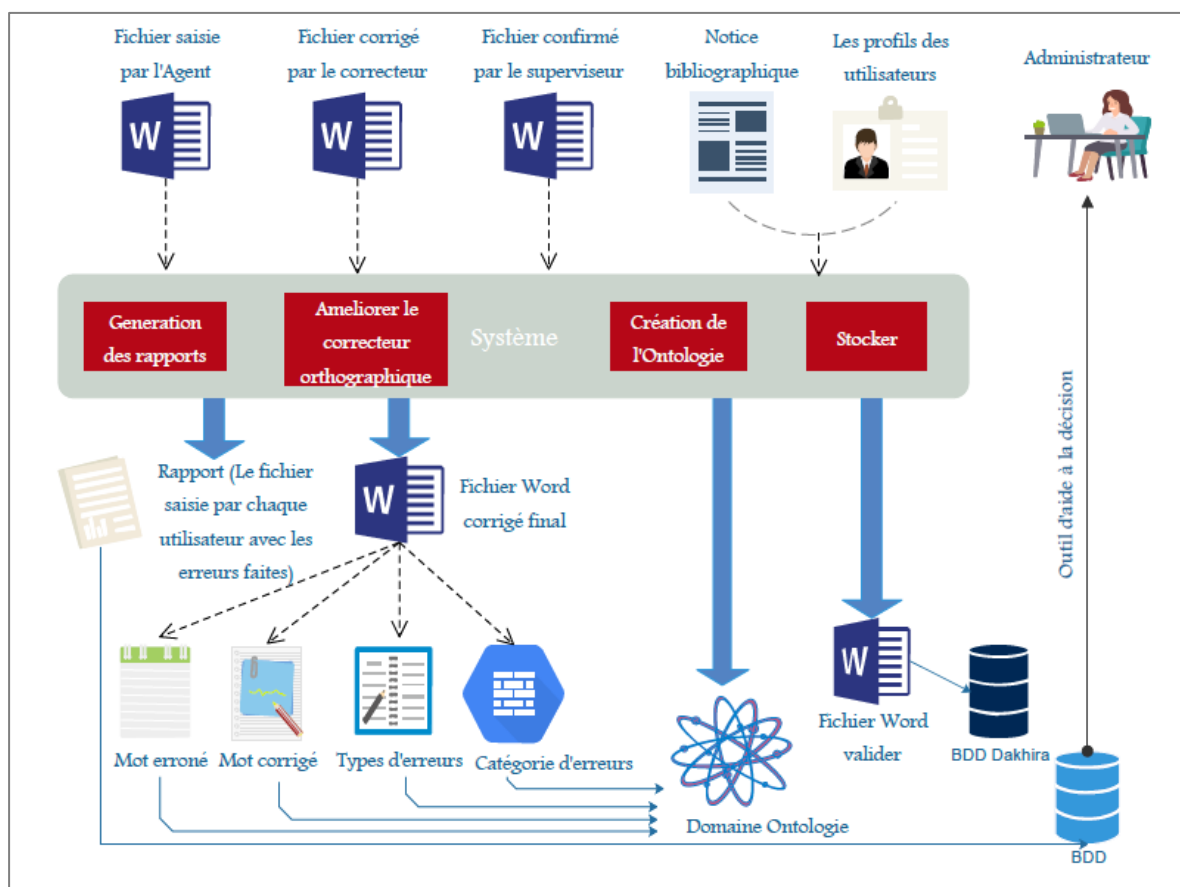


Figure 4.14: Tâche « Gestion de correction »

Cette illustration est détaillée dans le tableau suivant :

Tâche N° 4	Gestion de Correction
Résumé	Cette tâche permet de faire une correction sur les fichiers enregistrés dans la GED en utilisant le système pour aider le superviseur.
Acteur	Système.
Scénario nominal	Le système récupère le fichier saisi par l'Agent, le fichier corrigé par le correcteur et le fichier confirmé par le superviseur ainsi que la notice bibliographie et les profils des utilisateurs.

A partir des profils des utilisateurs et les notices bibliographiques ainsi que le fichier saisi par l'agent. Le système génère un rapport pour chaque utilisateur, le fichier saisi avec les types et taux des erreurs faites. Ce rapport sera stocké dans une Base de données comme un outil d'aide à la décision pour l'Administrateur.

Les utilisateurs utilisent le système pour faire la correction, ce qui aura pour effet d'améliorer le correcteur orthographique.

D'autre part, nous avons une Ontologie où nous allons stocker tous les mots erronés, les corrigés ainsi que les types et catégorie d'erreurs.

Finalement le superviseur stocke le fichier Word validé généré par le système dans la base de données d'El Dakhira Al Arabiyaa.

Tableau 4.4 : Description de la tâche de Correction.

5.3. Architecture globale du Système

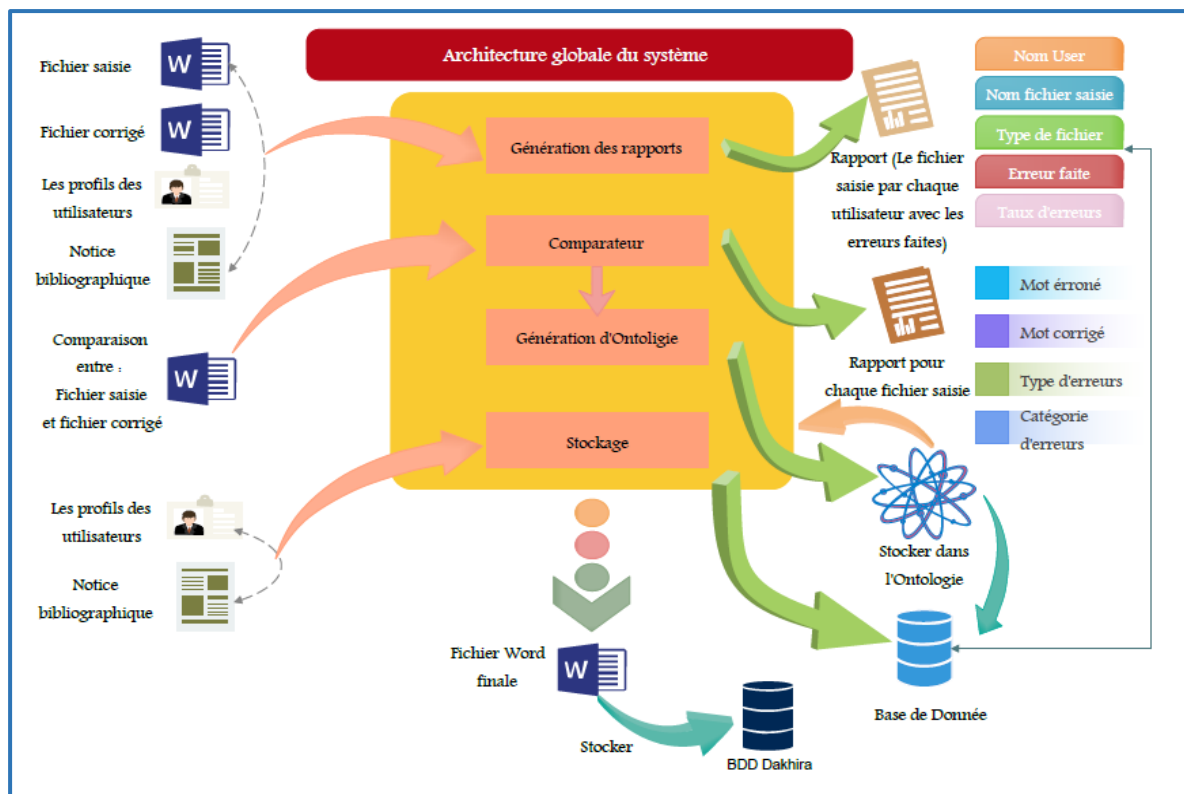


Figure 4.15: Architecture globale du Système.

Ce schéma présente l'architecture globale de notre système, Sa description sera représentée en détails dans le tableau suivant :

Les fonctionnalités du Système	Les besoins du Système (Entrées)	Ce que génère le Système (Sortie)	Objectifs
Génération des rapports	On a besoin du fichier saisi par l'Agent de saisie ainsi que le fichier corrigé, les profils des utilisateurs et la notice bibliographique pour réaliser : 	Un rapport pour chaque utilisateur contient : le Nom du livre saisi, le type, le taux d'erreurs faites.	Ce rapport sera utilisé par l'Administrateur comme un outil d'aide à la décision pour assurer une meilleure affectation des livres.
Correction orthographique	Après avoir fait une comparaison entre le fichier saisi et le fichier corrigé, on utilise le système pour créer : 	Un rapport pour chaque livre saisi qui contient : les Mots erronées, Mot corrigé, le Type et Taux d'erreurs et les catégories.	La réutilisation des informations sauvegardées dans l'Ontologie pour une prochaine correction.
Génération d'Ontologie	Le système utilise le rapport des erreurs généré dans l'étape précédente pour : 	Générer une Ontologie et stocker ces informations dans cette Ontologie.	Pour la création d'une base de connaissance linguistique pour améliorer le taux de détection d'erreurs.
Stockage	Cette fonctionnalité a besoin de fichier Word finale, des notices bibliographiques ainsi que les profils des utilisateurs pour : 	Connecter avec une Base de Données et les stocker.	Pour Archiver et Publier les Ouvrages.

Tableau 4.5 : Description Globale du Système.

6. Conception de l'architecture

6.1. Identification des diagrammes

Pour modéliser l'architecture du système, nous allons identifier trois diagrammes :

- Le diagramme de séquences qui fournit une représentation graphique de la technologie d'échange de messages entre des objets et des acteurs pour un cas d'utilisation dans un processus du système.
- Le diagramme d'activités décrit la succession des activités et leurs interactions dans notre système.
- Un diagramme de classes qui est une collection d'éléments de modélisations statiques, qui montre la structure d'un modèle. Les classes sont liées entre elles par des associations. Une association permet d'exprimer une connexion sémantique bidirectionnelle entre deux classes.

6.2. Conception des couches

Cette phase consiste à enrichir la description du procédé, de détails d'implémentation afin d'aboutir à une description très proche d'un programme. Nous allons modéliser toute l'architecture en diagramme de séquences, d'activités et de classe.

6.2.1. Diagramme de Séquence

Le diagramme suivant représente tous les interactions et les échanges de messages entre les acteurs et la GED.

- Scénario :
- Authentification
 - Pour que les utilisateurs puissent accéder à l'interface de la GED ils doivent s'inscrire.
- ✓ Administrateur
 - Après avoir s'inscrire le système affiche l'interface d'espace du travail.
 - L'Administrateur faire une indexation des ouvrages.
 - Le système fait une vérification sur les informations entrées et affiche la notice bibliographique.
 - L'Administrateur affecte les livres aux utilisateurs selon des conditions.

- ✓ L'Agent de saisie
 - L'Agent saisit les livres affectés dans l'espace de saisie de la GED.
 - Puis il va faire une première correction sur le fichier saisi.
 - Le système sauvegarde les deux versions du fichier.
 - Il affiche ces fichiers au Superviseur.
- Contrôle
 - ✓ Superviseur
 - Après avoir accédé à l'espace du travail, le superviseur va faire un contrôle sur les fichiers saisis par l'Agent de saisie.
- Correction
 - La correction des fichiers sera faite soit par le correcteur, soit par l'Agent de saisie.
- ✓ Superviseur
 - Le superviseur valide la correction et enregistre le fichier final.
 - Le système mis à jour la base de donnée et archive les ouvrages saisis.

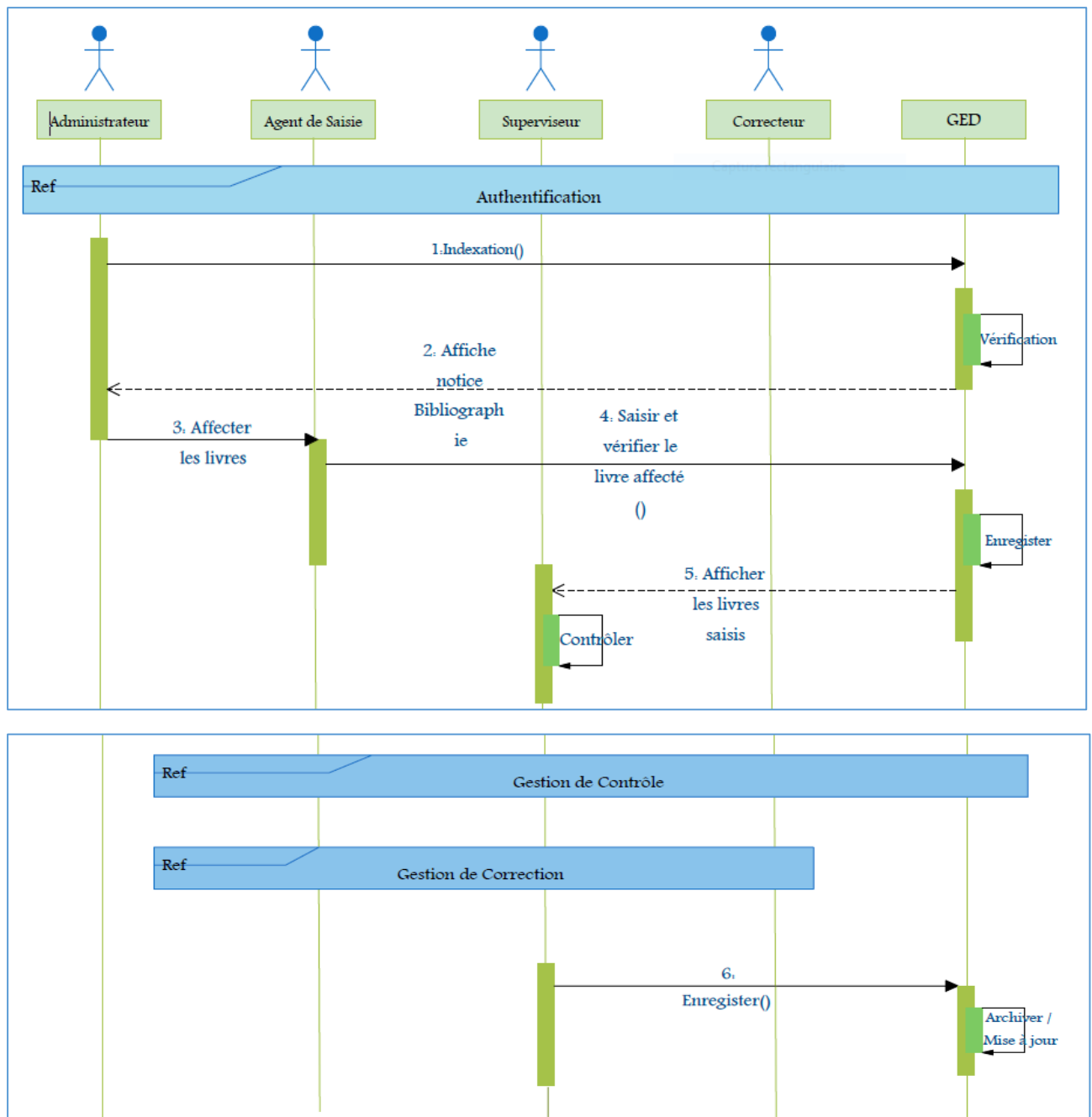


Figure 4.16: Diagramme de Séquence d'architecture globale.

6.2.2. Diagramme d'activité

Le diagramme suivant représente l'architecture globale de toutes les activités de la cellule de numération. Il décrit les fonctions des tâches et leurs interactions en assurant une activité idéale de la cellule (Fig 18, 19, 20, 21).

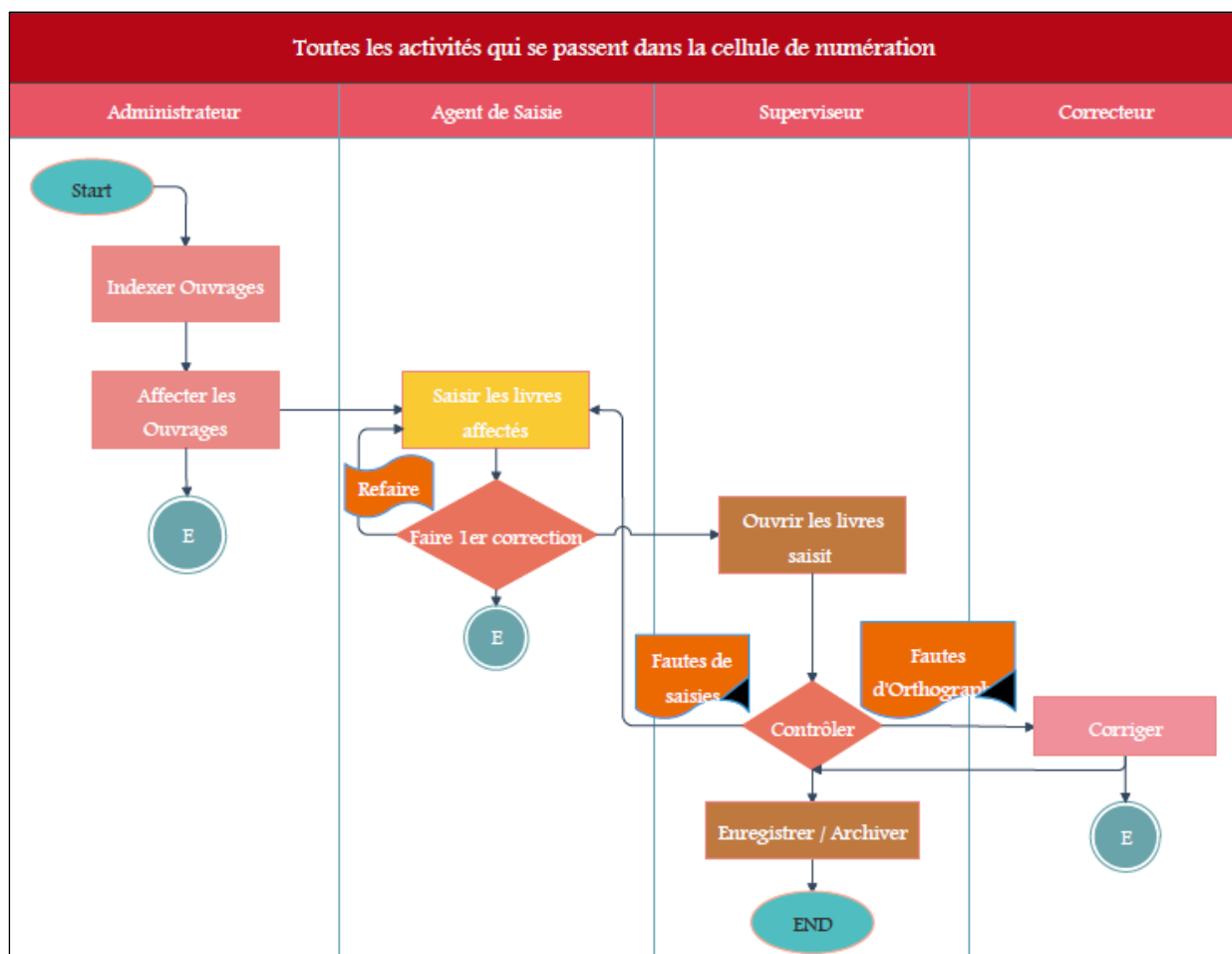


Figure 4.17: Diagramme d'activité d'architecture globale.

6.2.2.1. Diagramme d'activité « Indexation »

La figure qui suit représente : Le diagramme d'Activité de la tâche « Indexation ».

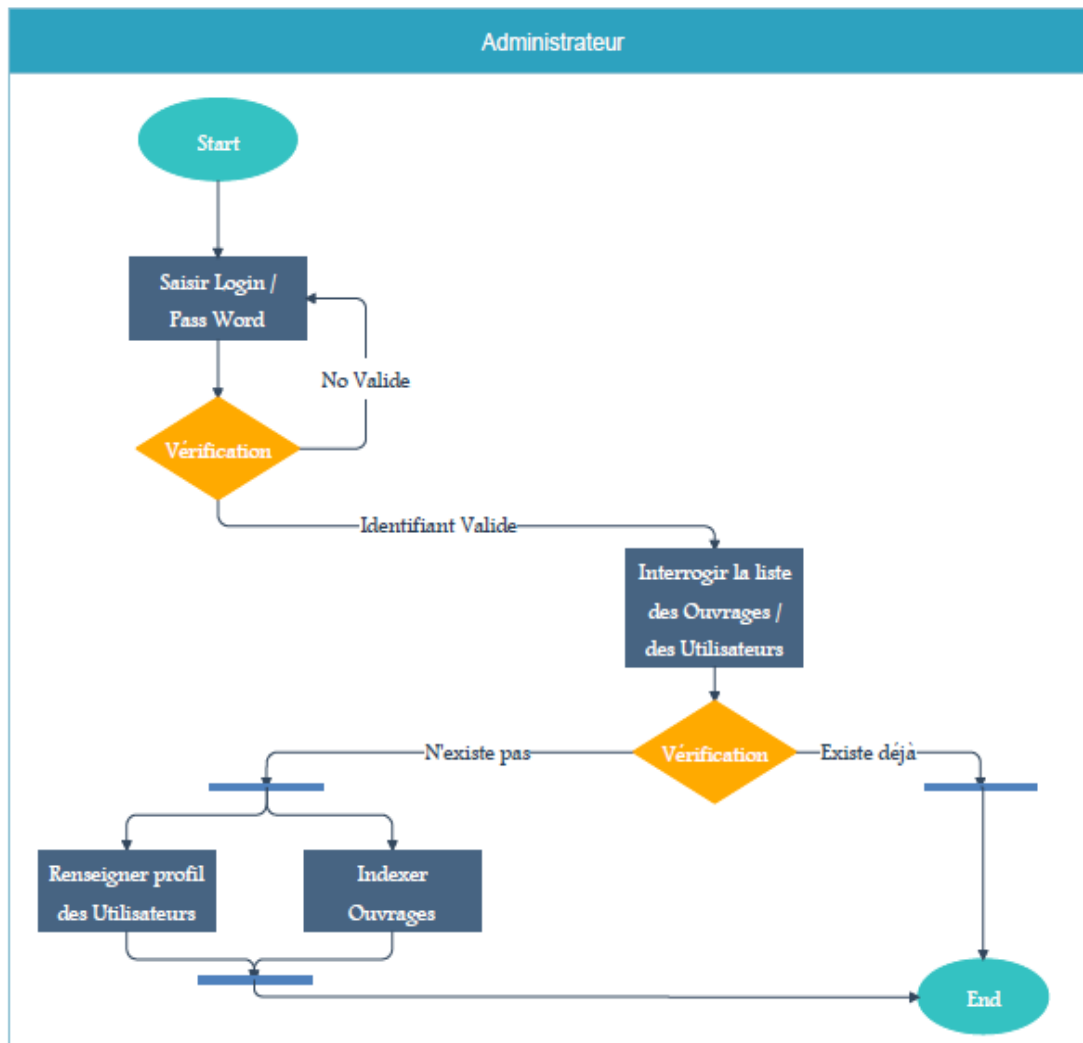


Figure 4.18: Diagramme d'activité « Indexation »

6.2.2.2. Diagramme d'activité « Gestion de Saisie»

La figure qui suit représente : Le diagramme d'Activité de la tâche « Gestion de saisie ».

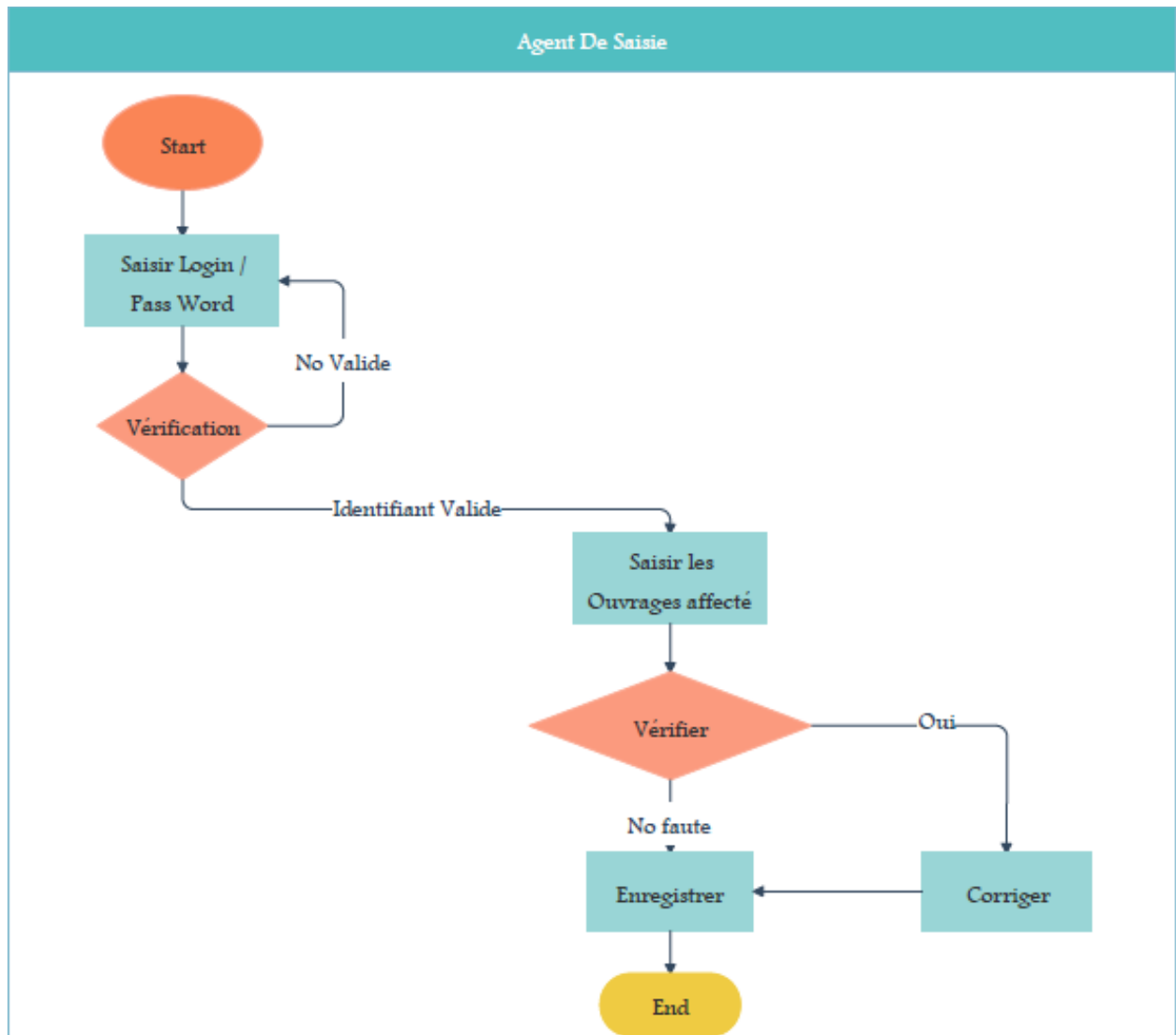


Figure 4.19: Diagramme d'activité « Gestion de Saisie»

6.2.2.3. Diagramme d'activité « Gestion de contrôle »

La figure qui suit représente : Le diagramme d'Activité de la tâche « Gestion de contrôle ».

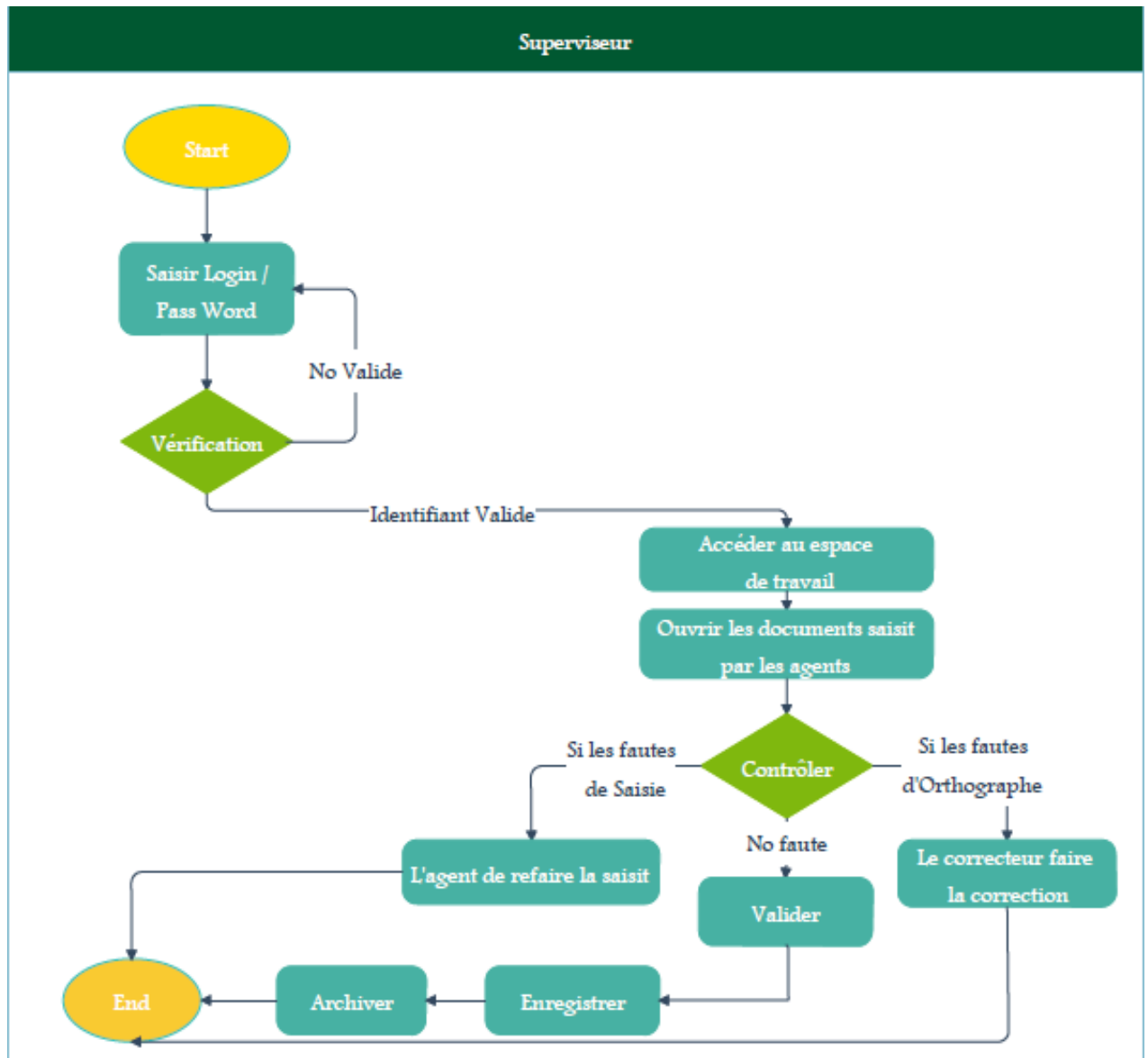


Figure 4.20: Diagramme d'activité « Gestion de contrôle »

6.2.2.4. Diagramme d'activité « Gestion de correction »

La figure qui suit représente : Le diagramme d'Activité de la tâche « Gestion de correction ».

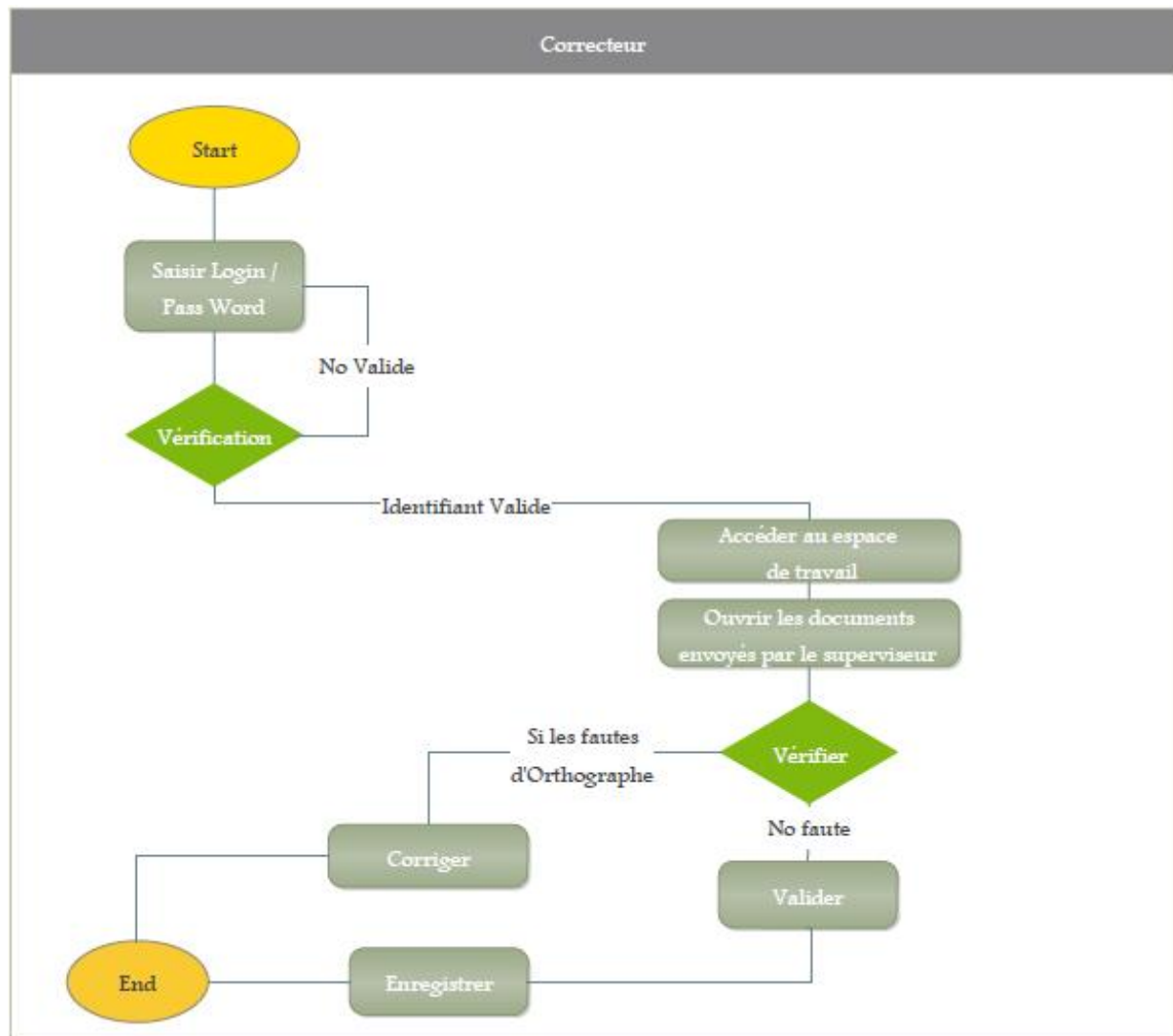


Figure 4.21: Diagramme d'activité « Gestion de correction »

6.2.3. Diagramme de Classe

6.2.3.1. Les règles de gestion

Le diagramme de classe du système étudié est basé sur les règles de gestion suivantes :

- Un administrateur peut ajouter un ou plusieurs utilisateurs ;
- Un administrateur peut ajouter un ou plusieurs livres ;
- Un utilisateur avoir une bibliographie linguistique ;
- Bibliographie linguistique peut-être de type orale ou écrit ;
- Un utilisateur peut saisir un ou plusieurs livres ;
- Un utilisateur fait un ou plusieurs erreurs ;
- Un livre contient plusieurs erreurs.

6.2.3.2. Dictionnaire de données

Nom de la classe	Codification	Désignation
Admin	Idadmin Username Password	Identifiant de l'Admin Nom d'utilisation de l'Admin Mot de passe de l'Admin
User	Iduser Nomuser Nom Prenom email Naiss Civillite Nationalite Niv Expr	Identifiant de l'Utilisateur Nom d'utilisation de l'Utilisateur Nom d'Utilisateur Prenom d'Utilisateur Email d'Utilisateur Date de naissance d'Utilisateur Sexe d'Utilisateur Nationalité d'Utilisateur Niveau d'étude d'Utilisateur Expérience d'Utilisateur
Livre	Idlivre Titre Domaine Typelivre Periode Auteur Pays Lieun Imprimerie Nbrepape Nbremots Datedbs	Identifiant du Livre Titre du Livre Domaine du Livre Type du Livre Période du Livre L'Auteur du Livre Pays de publication Lieu de publication Edition Nombre de page du Livre Nombre de mots du Livre Date début de saisi du Livre

	Datefns	Date fin de saisi du Livre
Erreur	Iderreur Moter Motcr Type Categorie	Identifiant d'Erreur Mot erroné Mot corrigé Type d'Erreur Catégorie d'Erreur
BibliographieL	Idbib	Identifiant du Bibliographie Linguistique
Orale	Idorale Arabe Francais Anglais Autre Preautre	Identifiant d'Orale Langue Arabe Langue Française Langue Anglaise Autre langue Précisant d'autre langue
Ecrit	Idecrit Arabee Francaiss Anglaiss Autres Preautres	Identifiant d'Ecrit Langue Arabe Langue Française Langue Anglaise Autre langue Précisant d'autre langue

Tableau 4.6 : Dictionnaire de Données.

6.2.3.3. Représentation du diagramme de classe

Le diagramme suivant montre les différentes classes métiers et la structure interne de notre système.

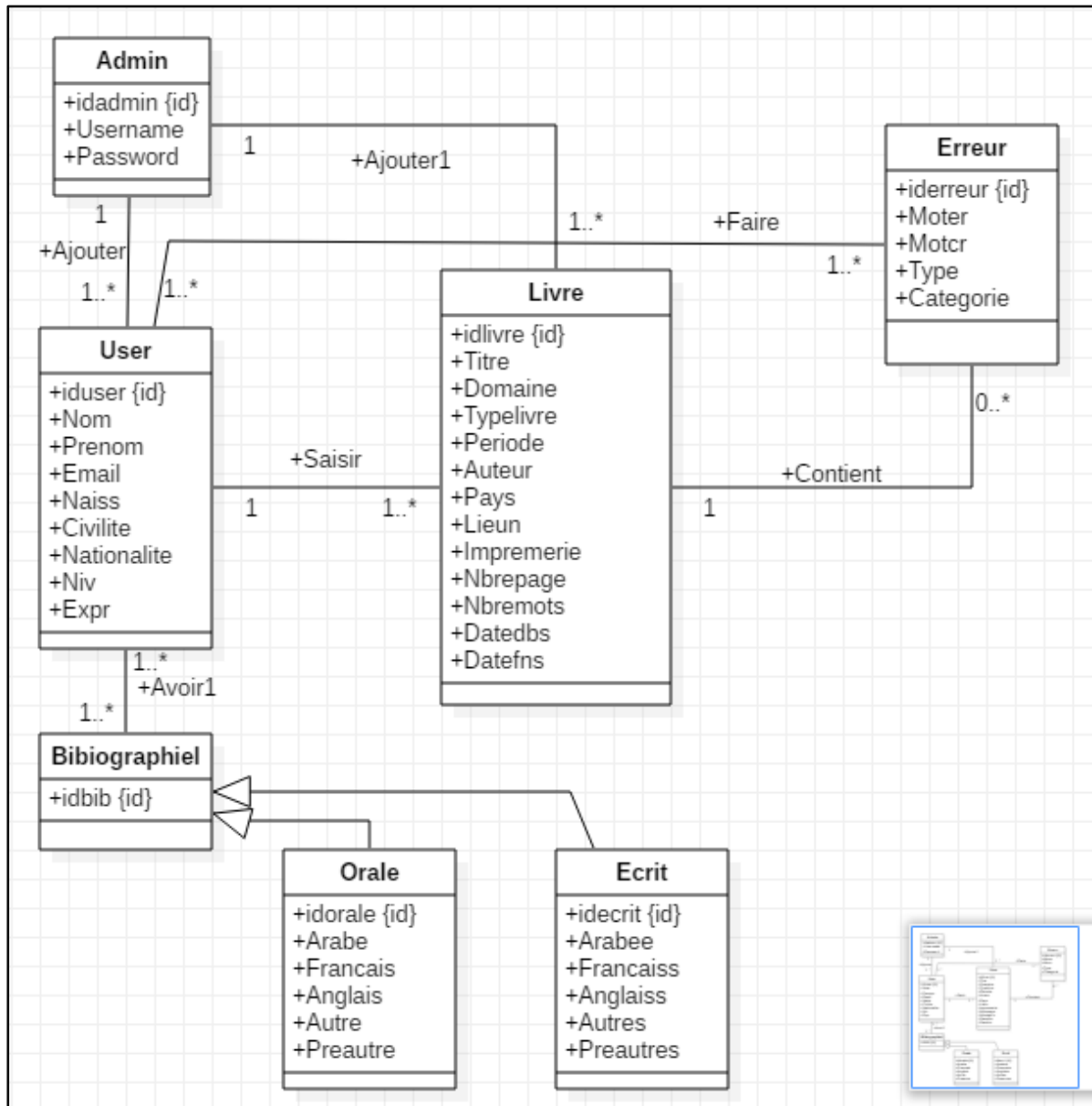


Figure 4.22: Diagramme de Classe.

6.2.4. Règle de passage du Diagramme de Classe vers le modèle relationnel

Du modèle conceptuel au modèle relationnel.

A partir de la description conceptuelle que nous avons effectuée, on peut réaliser le modèle relationnel, vu que le système d'information ne peut pas le manipulé directement, et ça en utilisons des règles de passages de l'UML vers le relationnel [25].

Quelques notions essentielles :

- Entité : toutes entités est transformé en table, les propriétés d'entités deviennent attribut de la table, l'identifiant de l'entité devient la clé primaire de la table.
- Association : relation entre deux ou plusieurs entités.
- Domaine : c'est l'ensemble des valeurs d'un attribut.
- Relation : c'est un sous ensemble du produit cartésien d'une liste de domaines. C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux Domaines et dont les lignes contiennent des tuples. On associe un nom à Chaque colonne.
- Attribut : Propriété d'une entité ou d'une association.
- Tuple : c'est la liste des valeurs d'une ligne d'une relation.
- Cardinalité : c'est le nombre minimal et maximal d'associations possibles entre deux entités.
- L'arité : est le nombre d'attributs d'une relation.
- Clé primaire : ensemble d'attributs dont les valeurs permettent de distinguer les n-uplets les uns des autres (notion d'identifiant).
- Clé étrangère : Attribut qui est clé primaire d'une autre entité.

Les règles de passage :

- Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.

- Transformation des associations : il existe trois types d'associations :
 - « Association 1..* » : se traduit par la création d'une clé étrangère dans la relation correspondants à l'entité coté (1).
Cette clé étrangère est la clé primaire de la relation correspondant à l'autre entité (n).
 - « Association *.* » : Se traduit par la création d'une relation dont La clé primaire est composé des clés étrangères référençant les relations correspondant aux entités liées par l'association.
 - « Association 1.. 1 » : on ajoute un attribut de type clé étrangère dans la relation ayant la multiplicité minimale égale à un, qui porte le nom de la clé primaire de l'autre relation connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux relations en une seule.

6.2.5. Modèle relationnel

En appliquant les règles précédentes de transformations d'un diagramme de classe vers un modèle relationnel, nous avons abouti au schéma relationnel suivant :

- Admin (Idadmin, Username, Password).
- User (Iduser, Nomuser, nom, Prenom, Email, Naiss, Civilite, Nationalite, Niv, Expr, *Idadmin).
- Livre (Idlivre, Titre, Domaine, Typelivre, Periode, Auteur, Pays, Lieun, Imprimerie, Nbrepape, Nbremots, Datedbs, Datefns, *Idadmin, *Iduser).
- Erreur (Iderreur, Moter, Motcr, Type, Categorie, * Idlivre).
- Faire (*Iduser, *Iderreur).
- BibliographieL (Idbib).
- Avoir1 (*Iduser, *Idbib).
- Orale (Idorale, Arabe, Francais, Anglais, Autre, Preautre, *Idbib).
- Ecrit (Idecrit, Arabee, Francaiss, Anglaiss, Autres, Preautres, *Idbib).
- Remarque :

Les attributs soulignés (__) représentent les clés primaires, les attributs commencés par (*) représentent les clés étrangères des différentes table.

7. Architecture globale de l'Ontologie

Dans le cadre de notre travail, nous avons comme objectif de créer une base de données des erreurs linguistiques contenues dans les documents produits (numérisés) par la cellule de l'Académie de la langue arabe. Cette base est utilisée pour la construction d'une base de connaissances linguistiques liée au domaine de détection et correction d'erreurs. La base de connaissances visées, contient les mots erronés, mots corrigés, type d'erreur, catégorie d'erreur et notice bibliographique du livre.

Pour se faire, nous proposons une solution basée sur les ontologies auxquelles nous avons consacré tout un chapitre pour aborder les notions relatives aux types et composantes des ontologies ainsi que la méthode de recherche orientée et les méthodes d'ingénierie et enfin les outils utilisés pour leur construction.

Dans cette section, nous décrivons notre démarche de la conception de l'ontologie. Pour rappel, la connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de composants à savoir : concepts (ou classes), relations (ou propriétés), fonctions, axiomes (ou règles) et instances (ou individus). Par ailleurs, la méthode de recherche orientée ontologie est axée sur quatre grandes phases qui sont :

- Phase 1 : l'étude de faisabilité : dans cette phase, nous identifierons le problème et les solutions potentielles.

Notre problème est comment générer une base de connaissances pour stocker les données récupérées par le système à partir de la base de donnée ?

Pour mettre en place la solution ontologique, il nous a fallu réfléchir à certains aspects. Notamment :

- Base de règles pour les corrections orthographiques (passage d'un mot erroné à un mot corrigé) ;
- Stockage des types, des catégories, et nombre d'erreurs pour l'amélioration des corrections ;
- La réutilisation des données relatives aux corrections effectuées pour mieux appréhender les erreurs futures.

- Phases 2 : Modélisation ontologique : Cette phase comprend cinq étapes :
 - Étape 1 : Capture des besoins : cette étape permet de définir un cadre de référence pour l'implémentation et l'évaluation de l'ontologie. Nous avons opté pour une ontologie de domaine car ce type nous sera utile pour définir les objectifs ciblés de notre projet. Nous utiliserons cette ontologie pour améliorer l'outil d'aide à la correction en enrichissant sa liste de mots erronés-mots corrigés.
 - Étape 2 : Analyse des sources d'information : la démarche est itérative et prend fin lorsque l'on obtient un niveau de complétude ou de détail satisfaisant.

L'analyse des sources consiste à identifier les classes et les relations qui répondent à notre besoin. D'abord, nous avons passé du temps à analyser les processus métiers au sein de la cellule de numérisation de l'Académie. Ce qui nous a permis de dégager les classes, instances, et règles qui pourraient servir pour régler les problèmes qu'ils rencontrent.

- Étape 3 : Construction de l'ontologie initiale :
 - Étape 3.1 : Conceptualisation de l'ontologie :
 Cette étape nous mettons en relation les concepts « primitifs » pour former les taxonomies de base. Le domaine est articulé sans tenir compte des contraintes informatiques. À ce stade, la modélisation étant Semi-informelle on se concentre sur :
 - Nous construisons des taxonomies de base (concepts, relation taxinomique). La construction d'une taxonomie repose sur la mise en relation d'un concept général (ou abstrait) à un concept plus spécifique au moyen d'un lien de généralisation-spécialisation ou « Is-a » ou encore « subclass-of »
 - Nous ajoutons des rôles (propriétés, attributs) et les restrictions (contraintes de type et valeur)

Nous avons deux classes qui se rapportent au concept « livre » :

« Les notices bibliographiques » et « les erreurs ».

- Livre contient les attributs suivantes : id_livre et nombre d'erreurs
 - La notice bibliographique contient les attributs suivants : id_livre, domaine, type de livre, lieu géographique, période).
 - Les erreurs qui vont contenir « Catégorie d'erreur » qui elles-mêmes possèdent un « type d'erreurs», « mot erroné », « mot corrigé ».
- Étape 3.2 : Semi-formalisation de l'ontologie :

Dans cette étape, nous allons formaliser partiellement l'ontologie sans perdre d'expressivité.

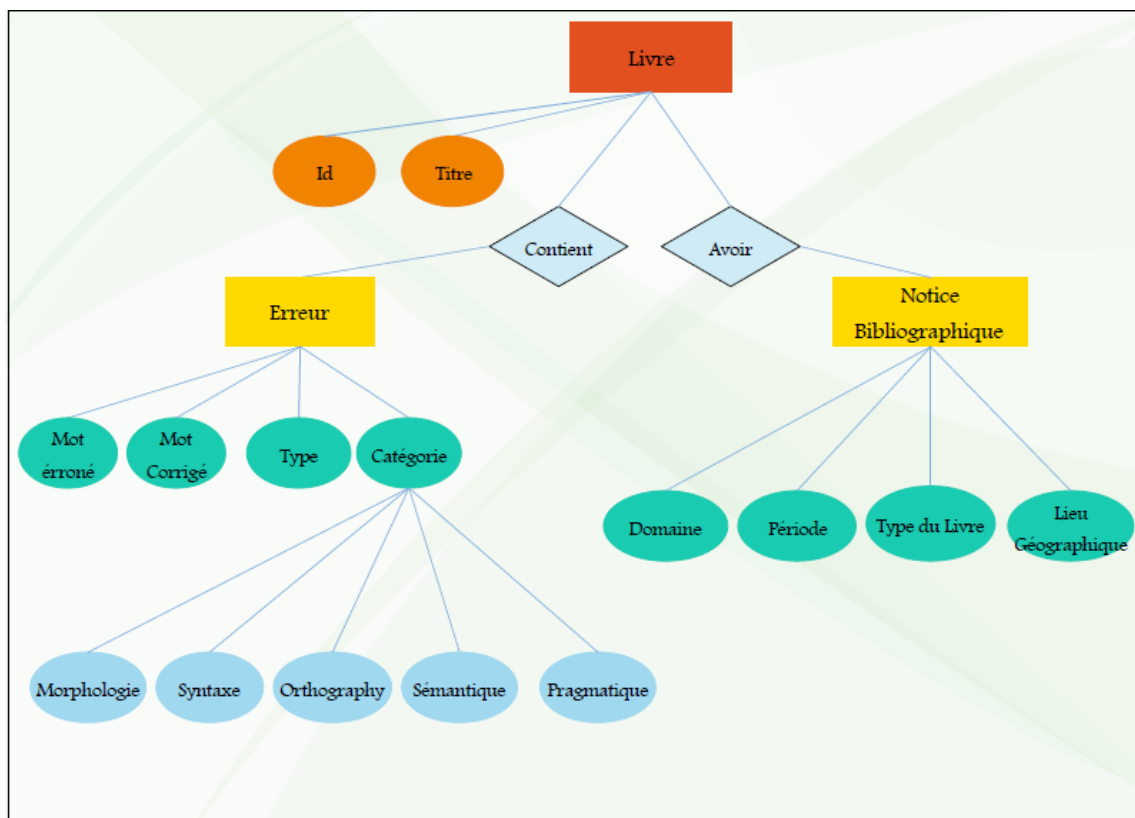


Figure 4.23 : Architecture globale d'Ontologie.

- Étape 4 : Raffinement de l'ontologie :

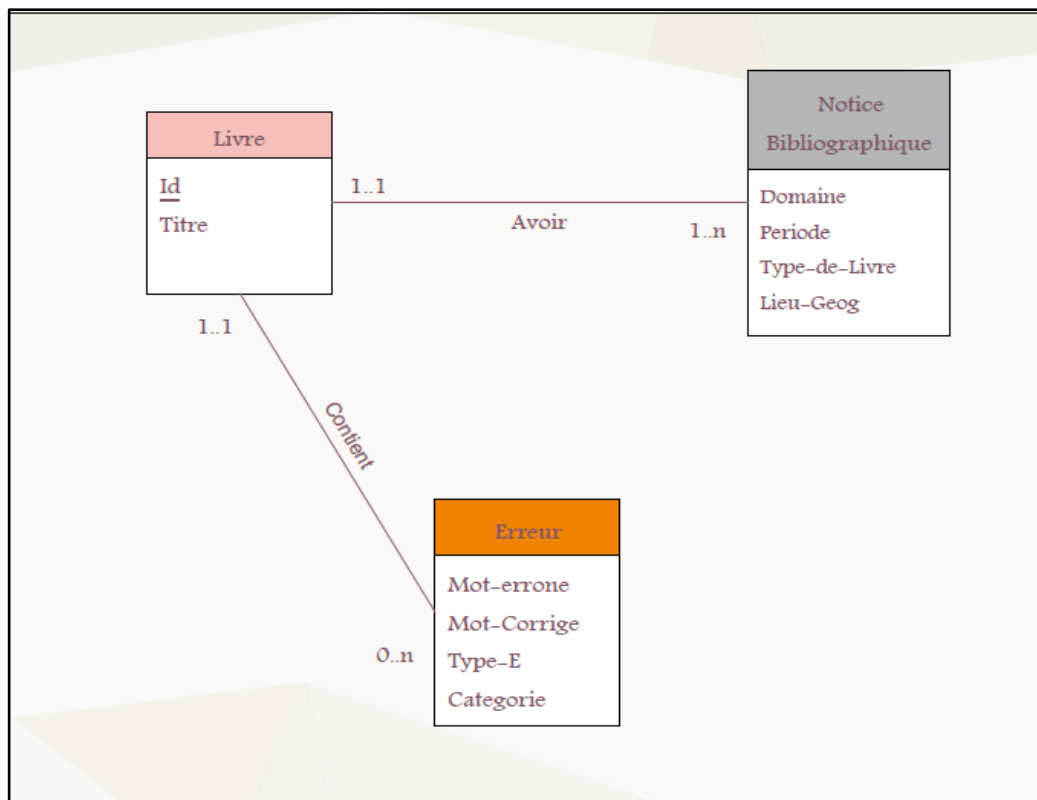


Figure 4.24 : Diagramme de classe d'Ontologie.

- Étape 5 : Formalisation de l'ontologie :

Pour la formalisation de notre Ontologie, nous commençons par la création des classes : livre, erreur, notice bibliographique, après les relations entre eux, Et enfin nous créons les attributs de chaque classe.

Nous allons présenter ces étapes dans le chapitre de l'implémentation en utilisant le logiciel Protégé.

- Phase 3 : Mise en œuvre informatique de l'ontologie :

Capture des besoins et analyse additionnels :

Il s'agit d'identifier et de documenter des scénarios pour l'exploitation de l'ontologie, en termes de services et fonctions rendus par l'agent qui l'exploitera. Nous résumé ça par schéma ci-dessous :

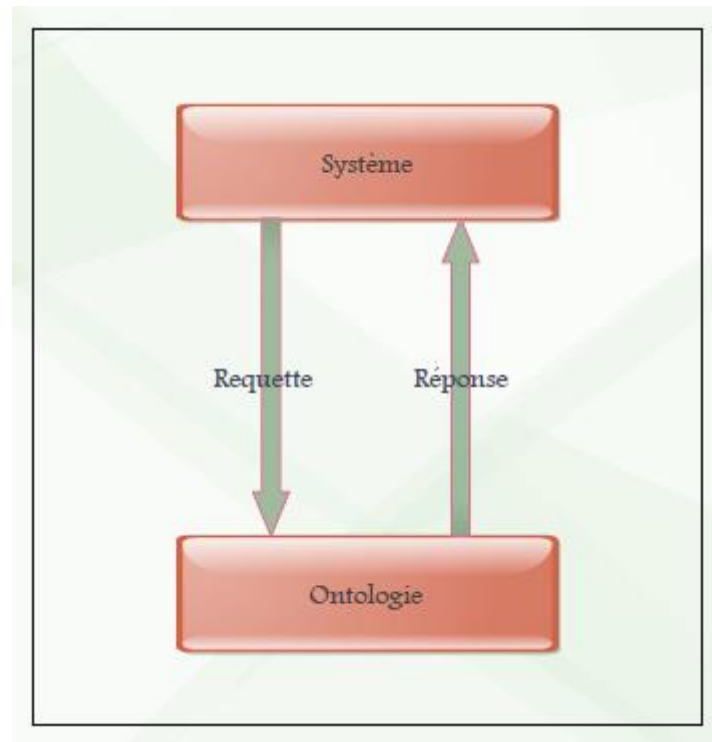


Figure 4.25 : L'exploitation de l'ontologie.

Enfin l'étape 4 de phase « Mise en œuvre informatique de l'ontologie » nous allons expliquer dans le chapitre 5 « implémentation » et même chose avec la phase 4 « Méthode d'évaluation ».

Conclusion

Dans ce chapitre, nous avons présenté la conception de la solution que nous proposons pour répondre au besoin exprimé par les porteurs du projet ; à savoir l'amélioration du travail de la cellule de numérisation. Notre solution se base sur la dotation de la cellule d'une GED et le développement d'un système qui permet de :

- Récupérer les données nécessaire de cette GED ;
- Utiliser un outil d'aide à la correction des erreurs permettant la détection de l'erreur et la proposition d'une forme corrigée en appliquant une règle donnée ;
- Construire un outil d'aide à la décision pour assurer une meilleur affectation des livres ;
- Construire une base de connaissance « Ontologie » pour stocker les informations générées.

Le prochain chapitre nous permettra de passer à l'implémentation de notre solution.

Chapitre V : Implémentation de la solution.

Introduction

L'implémentation est la phase la plus importante après celle de la conception. Le choix des outils de développement influence énormément le cout en temps de programmation, ainsi que sur la flexibilité du produit réalisé.

Cette phase consiste à transformer le modèle conceptuel établi précédemment en des composants logiciels formant notre système.

Dans ce chapitre, nous allons commencer par la description de l'environnement de développement et les langages de programmation. Puis les différents outils utilisés dans chaque étape dès le prétraitement jusqu'à l'apprentissage en finalisant par le décodage. Enfin, on présente une description des différentes interfaces pour expliquer le déroulement du système et commenter les résultats obtenus.

1. Environnement de développement

L'environnement de développement est constitué par deux parties nommées environnement matériel et environnement logiciel.

1.1. Environnement matériel

Pour la réalisation de ce projet on a disposé de :

- Un ordinateur portable de type ACER :
 - Système d'exploitation : Windows 10 Professionnel 64 bits.
 - Processeur : Intel® Core™ i3-6006U.
 - CPU : 2.00 GHz.
 - Mémoire : 4 Go.
- Un ordinateur portable de type LENOVO :
 - Système d'exploitation : Windows 10 famille 32 bits.
 - Processeur : Intel® Pentium®.
 - CPU : 2.00 GHz.
 - Mémoire : 2 Go.

1.2. Environnement logiciel

Les logiciels utilisés sont :

- XAMPP est une suite facilitant l'installation d'un serveur Web Apache. Cette « distribution » se chargera donc d'installer l'ensemble des outils dont vous pourriez avoir besoin lors de la création d'un site Web. Plus d'une dizaine d'utilitaires sont intégrés, comme MySQL, PHP, Perl ou encore phpMyAdmin [26].
- Modélisation conceptuelle
 - StarUml [27]:
Est un logiciel de modélisation UML, cédé comme open source par son éditeur, il gère la plupart des diagrammes spécifiés dans la norme UML 2.0, il est écrit en Delphi1.
 - EDRAW MAX [28]:
Est une solution logicielle permettant de réaliser des diagrammes, des graphiques et des dessins professionnels pour de nombreux besoins. Les développeurs, commerciaux et ingénieurs pourront l'utiliser pour concevoir des diagrammes de flux, des cartes mentales détaillées, des dessins électriques ou encore des plans pour les architectes. Vous pouvez aussi exporter vos schémas, ou encore les partager comme un simple fichier exécutable.
- Editeur texte
 - Notepad++ [29] :
Est un éditeur de code source qui prend en charge plusieurs langages. Ce programme, codé en C++ avec STL et win32 api, a pour vocation de fournir un éditeur de code source de taille réduite mais très performant. En réduisant l'utilisation de CPU, la consommation d'énergie des ordinateurs chute considérablement, en conséquence de quoi, la terre est plus verte.
- Environnement de développement
 - PyCharm [30]:
Est un environnement de développement intégré (EDI), il permette d'intégrer dans une même fenêtre tous les éléments utiles à la

programmation en python : un éditeur de texte pour écrire des scripts, une console pour exécuter des programmes, ainsi qu'un explorateur de fichiers pour parcourir le projet en cours.

- Editeur d'Ontologie

- Protégé [31]:

Est un système auteur pour la création d'ontologies. Il a été créé dans le domaine du Web sémantique et au niveau de la recherche en informatique. Protégé est développé en Java, il peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, etc.

2. Langage de programmation

Dans cette partie, nous allons présenter les différents langages de programmation utilisés pour la réalisation de notre projet :

- Python [32]:

Est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation fonctionnelle et orientée objet. Le langage Python est placé sous une licence libre et fonctionne sur la plupart des plates-formes informatique, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par MacOS, ou encore Android, Ios, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

- SQL [33]:

Est un langage informatique utilisé pour exploiter des BDD. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données.

Dans la pratique, ce langage est utilisé pour créer des tables, ajouter des enregistrements sous forme de lignes, interroger une BDD, la mettre à jour, ou encore gérer les droits d'utilisateurs de cette base. Il est bien supporté par la très grande majorité des systèmes de gestion de base de données (SGBD).

- PHP [34]:

C'est un langage compilé et exécuté du côté serveur et non du côté client. La syntaxe du langage provient de celles du langage C et du java, Ses principaux atouts sont :

- La simplicité d'écriture des scripts ;
- La possibilité d'inclure le script PHP au sein d'une page HTML ;
- La simplicité d'interfaçage avec des bases de données ;
- L'intégration au sein de plusieurs serveurs web.

- HTML [34]:

Est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie "HyperText Markup Language", ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.

- CSS [34]:

C'est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de styles, comprennent du code qui permet de gérer le design d'une page en HTML.

- JavaScript [34]:

Est un langage de programmation qui est incluse dans le code HTML. Il permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes.

- UML [35]:

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, etc.

- MySQL [36]:

Est un SGBD qui permet la gestion efficace d'une grande base de données ainsi qu'une bonne diversité de choix des types de données. MySQL assure aussi la sécurité l'intégrité des données de la base. Les principaux atouts de MySQL sont la rapidité, la robustesse et la facilité d'utilisation.

3. Description du travail réalisé

3.1. Logo

Etant donnée l'AALA possédant déjà un logo personnalisé Significatif, nous l'avons utilisé et le mettre en évidence sur la partie droite de chaque interface de notre systèmes.



Figure 5.26 : Logo de l'AALA.

3.2. Maquettes du système GED

Le maquettage reste une des étapes les plus importantes à définir afin d'avoir un rendu visuel de notre système, nous sommes passés par un maquettage sous forme des schémas (figures 10, 12 de chapitre de Conception) avant de pouvoir définir les suivantes (figures 27, 28, 29, 31).

Au lancement d'un navigateur WEB, la page d'accueil d'authentification est présentée.

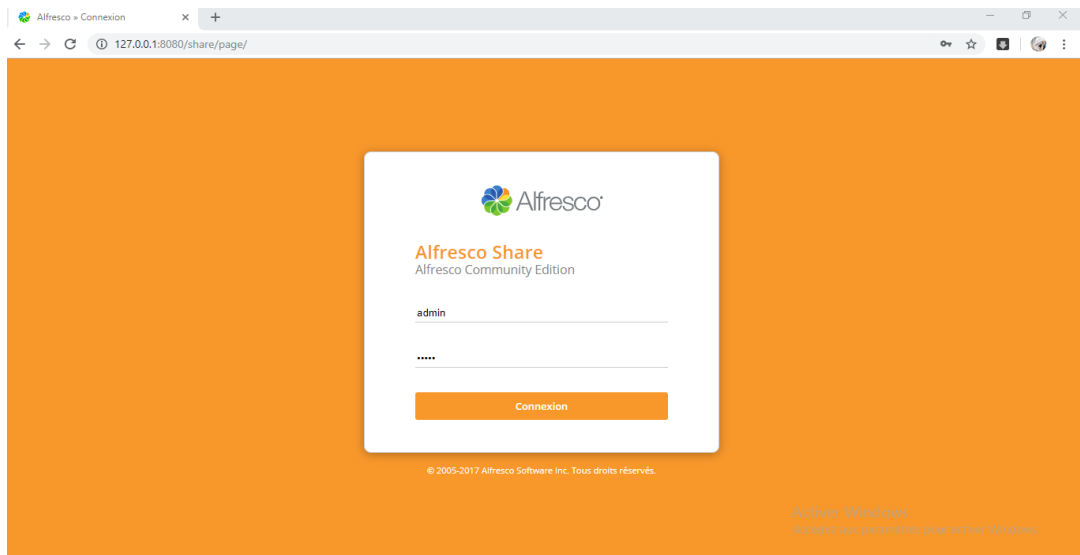


Figure 5.27: Page d'authentification d'Alfresco.

3.2.1. Espace Administrateur

Les maquettes suivantes représentent la page d'accueil d'espace Administrateur de notre système qui contient les principales fonctionnalités «Créer les utilisateurs», « Gérer les tâches », « Contrôler ».

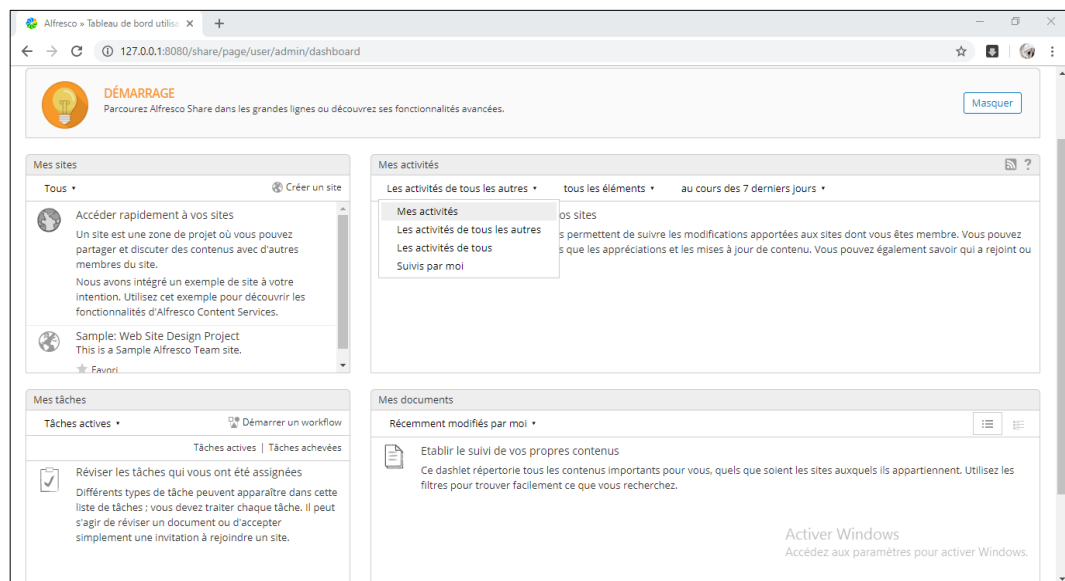


Figure 5.28 : Tableau de bord d'Administrateur.

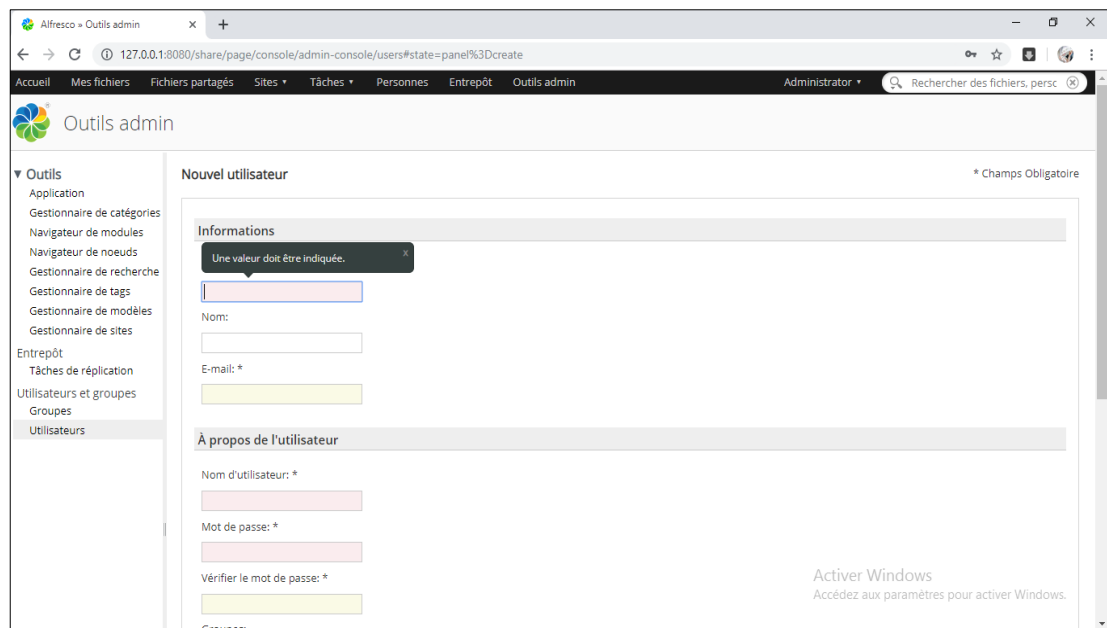


Figure 5.29 : Page d'ajout un utilisateur.

Cette figure présente, la création du Workflow

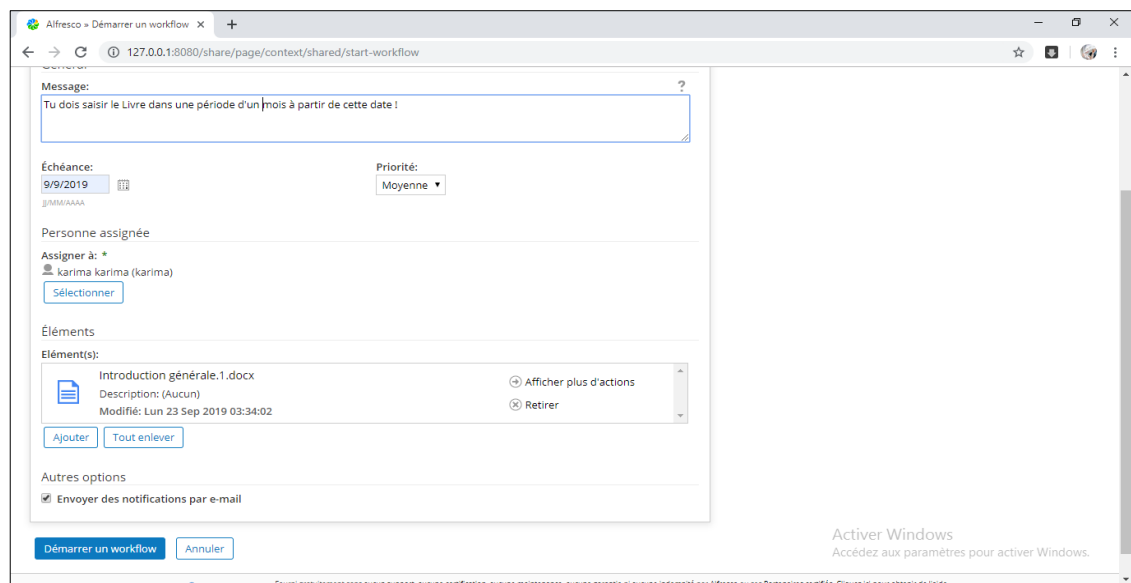


Figure 5.30 : Création d'un Workflow.

A partir de cette fenêtre, l'administrateur peuvent contrôlé tout les utilisateurs.

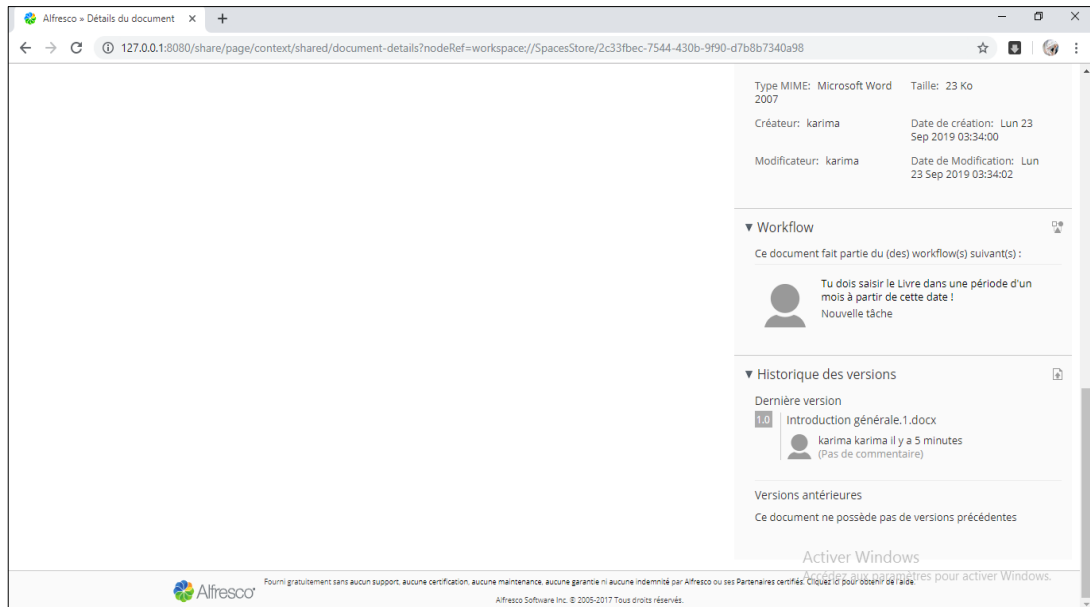


Figure 5.31 : Contrôler les utilisateurs.

3.2.2. Espace Utilisateurs

Les maquettes suivantes représentent la page d'accueil d'espace Utilisateurs du système qui contient les fonctionnalités suivantes : «Vérifier le travail demandé», « Saisir les livres affectés », «établir une 1^{er} correction ».

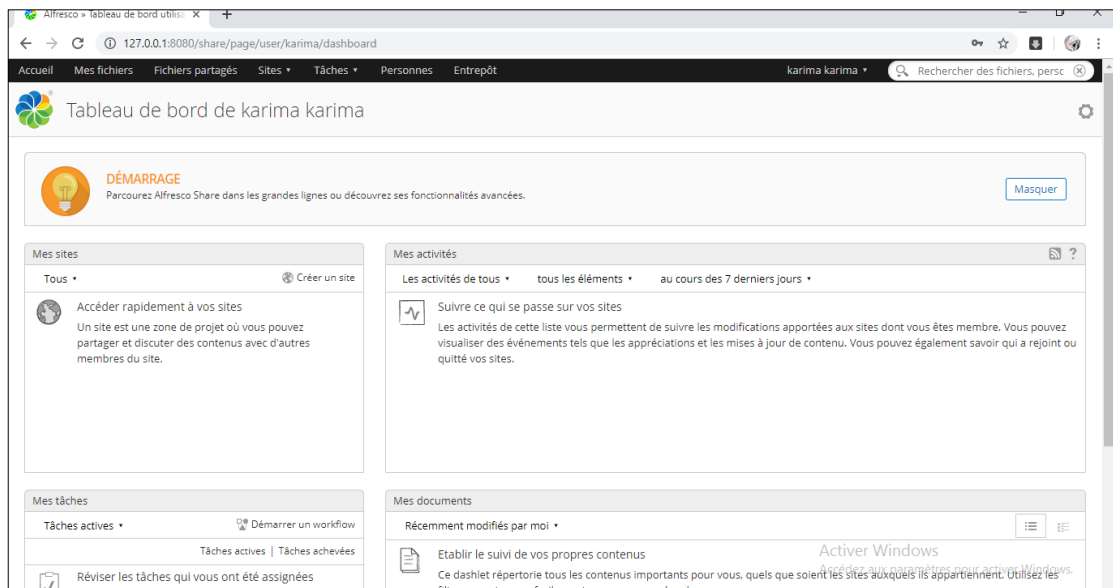


Figure 5.32 : Tableau de bord d'utilisateur.

La figure suivant représente la vérification d'utilisateur du travaux demandés par l'administrateur dans la case « Mes tâches »

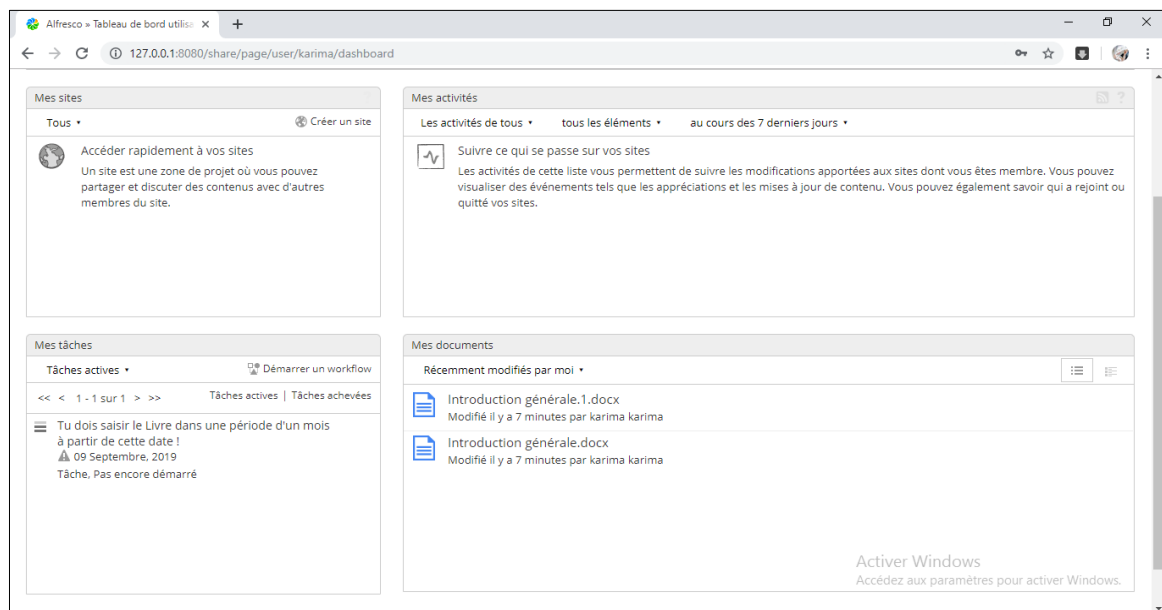


Figure 5.33 : Vérification du travail demandé.

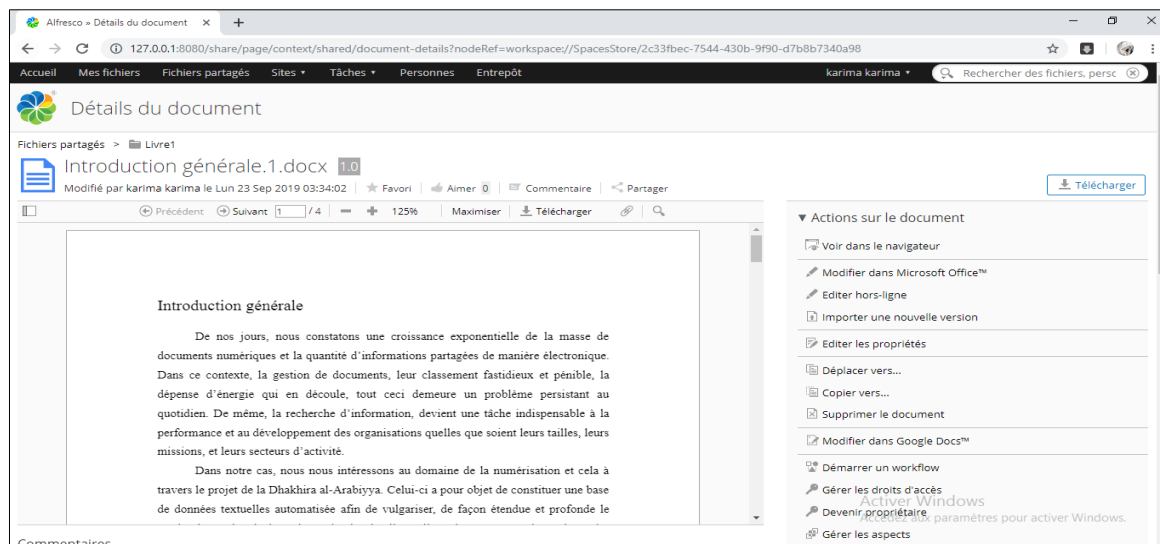


Figure 5.34 : Page de saisie.

Après avoir terminé la saisie et la 1^{ère} correction des livres l'administrateur va exporter les deux versions dans le bureau pour faire la correction orthographique (Figures 61, 62).

3.3.Scénario d’usage d’un système d’aide à la décision

Au lancement d'un navigateur WEB, la page d’accueil est présentée.



Figure 5.35 : Page d'accueil 1 du système d'aide à la décision.

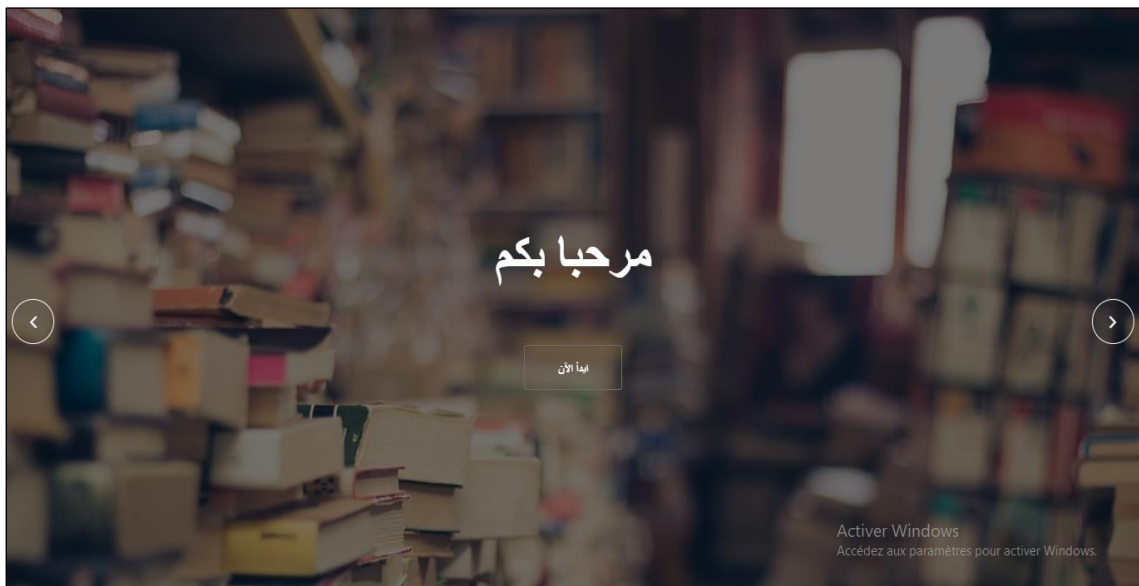


Figure 5.36 : Page d'accueil 2 du système d'aide à la décision.



Figure 5.37: Page d'accueil 3 du système d'aide à la décision.

Tout d'abord il faut cliquer sur "ابدأ الآن", pour surfer en toute sécurité. Alors qu'une page d'authentification est présentée.

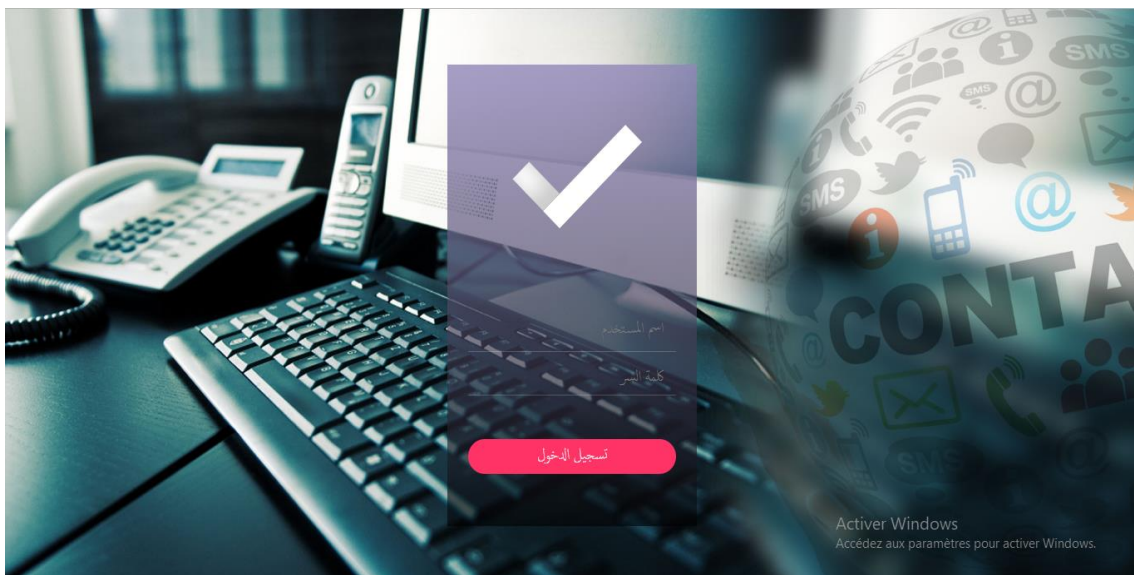


Figure 5.38 : Page d'Authentification.

Après avoir remplir les champs "إسم المستخدم" et "كلمة السر" obtenus préalablement, une fois l'authentification est effectuée, le navigateur affiche la première page.

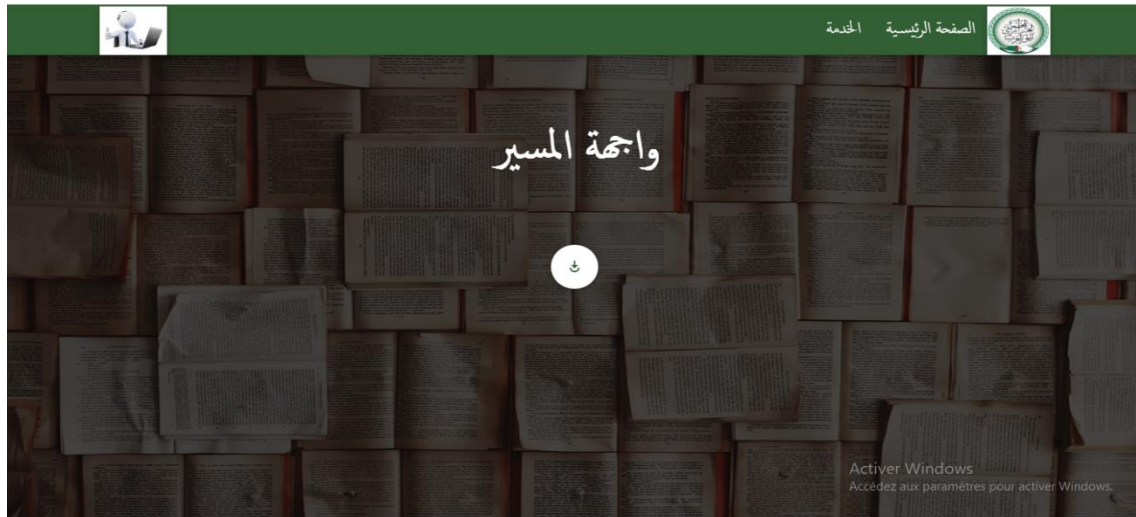


Figure 39 : Interface d'un administrateur via le système.

En cliquant sur le bouton au milieu de la page ou bien sur "الخدمة" dans la barre d'outils une deuxième page des services est présentée comme suit :

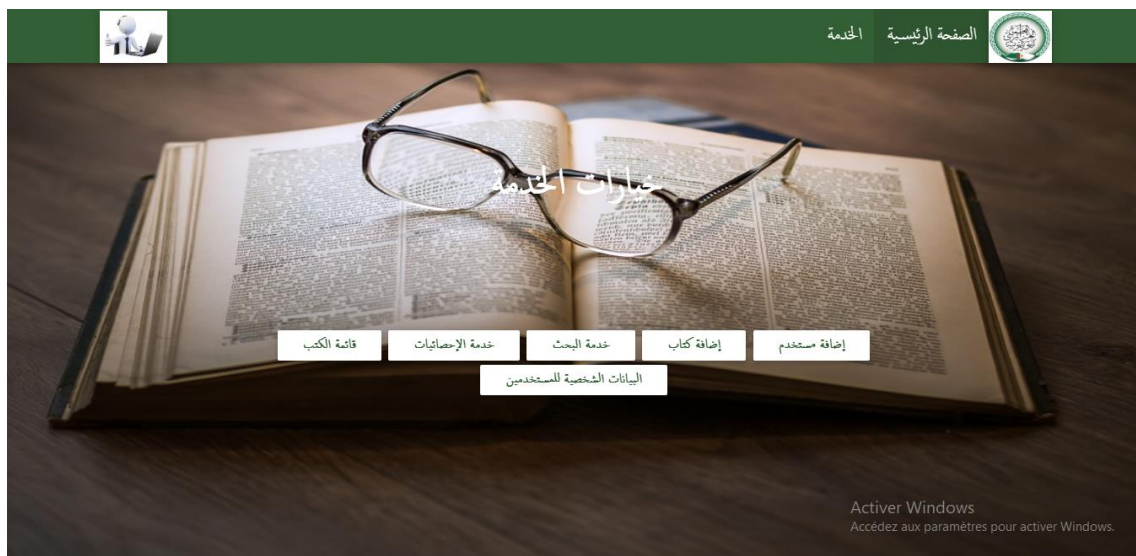


Figure 5.40 : Page des services.

3.3.1. L'ajout d'un opérateur de saisie

Cliquez sur le bouton "إضافة مستخدم" pour ajouter un nouvel utilisateur, la page d'ajout s'affiche comme suit :

إضافة مستخدم جديد

إسم المستخدم

الاسم

اللقب

البريد الإلكتروني

تاريخ الميلاد: jj/mm/aaaa

الجنس: ☐ ذكر ☐ أنثى

الجنسية: Algeria

المستوى الدراسي: ☐ ابتدائي ☐ متوسط ☐ ثانوي ☐ جامعي

الخبرة المهنية:

السيرة اللغوية: ☐ عربية ☐ فرنسية ☐ إنجليزية ☐ أخرى

المشاهدة:

الكتابة: ☐ عربية ☐ فرنسية ☐ إنجليزية ☐ أخرى

تسجيل

Figure 5.41 : Page d'ajout un opérateur de saisie.

Ensuite remplissez tous les champs présentés dans le formulaire puis cliquez sur le bouton "تسجيل", une fois que tous les champs sont remplis correctement, un message de confirmation sera affiché.



Figure 5.42 : Message de confirmation d'ajout d'un opérateur.

Sinon, si le nom d'utilisateur "إسم المستخدم" existe déjà, un message d'erreur est affiché.



Figure 5.43 : Message d'erreur d'ajout un opérateur.

3.3.2. L'ajout d'un Livre

Cliquez sur le bouton "إضافة كتاب" pour ajouter un nouvel Livre, la page d'ajout s'affiche comme suit :

إضافة كتاب جديد

عنوان الكتاب:

ميدان الكتاب:

نوع الكتاب: ☐ كتاب ☐ قصة ☐ قاموس ☐ مجلة ☐ أخرى

المدة الزمنية:

الكاتب:

بلد المنشأ:

مكان النشر:

دار النشر:

عدد الصفحات:

عدد الكلمات:

إسم المستخدم:

تاريخ بداية الحياة:

تاريخ إنهاء الحياة:

تسجيل

Figure 5.44 : Page d'Ajout un nouvel livre.

Ensuite remplissez tous les champs présentés dans le formulaire puis cliquez sur le bouton "تسجيل", une fois tous les champs sont remplis correctement, un message de confirmation sera affiché.



Figure 5.45 : Message de confirmation d'ajout un livre.

Sinon, si le livre existe déjà, un message d'erreur est affiché.



Figure 5.46 : Message d'erreur d'ajout un livre.

3.3.3. Faire une recherche

Cliquez sur le bouton "خدمة البحث" pour rechercher un utilisateur ou bien un livre, la page de recherche sera présentée comme suit :

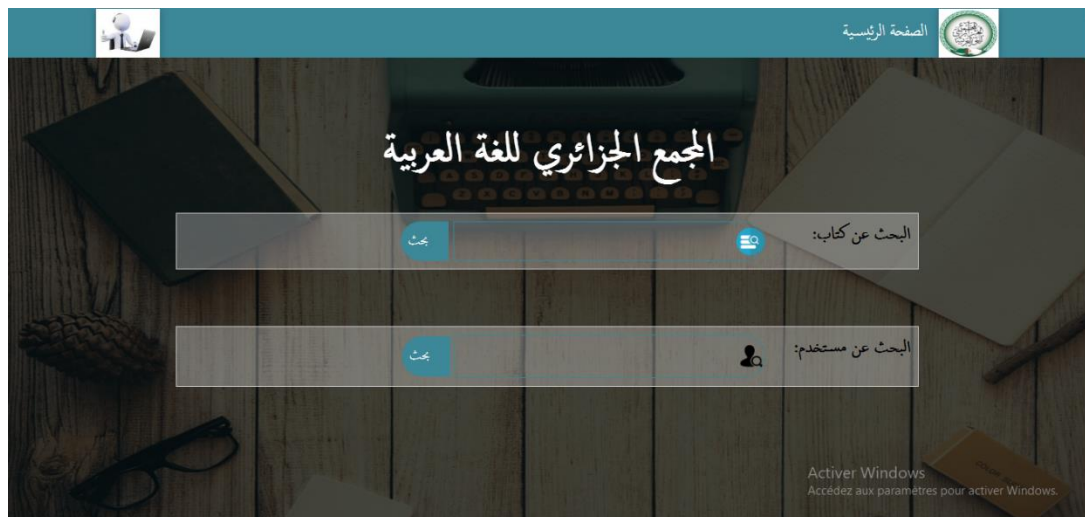


Figure 5.47 : Page de recherche.

- Si vous cherchez un livre, saisissez "عنوان الكتاب" ou "ميدان الكتاب" ou "الكاتب" dans la barre de recherche « البحث عن كتاب » le résultat sera affiché comme suit :

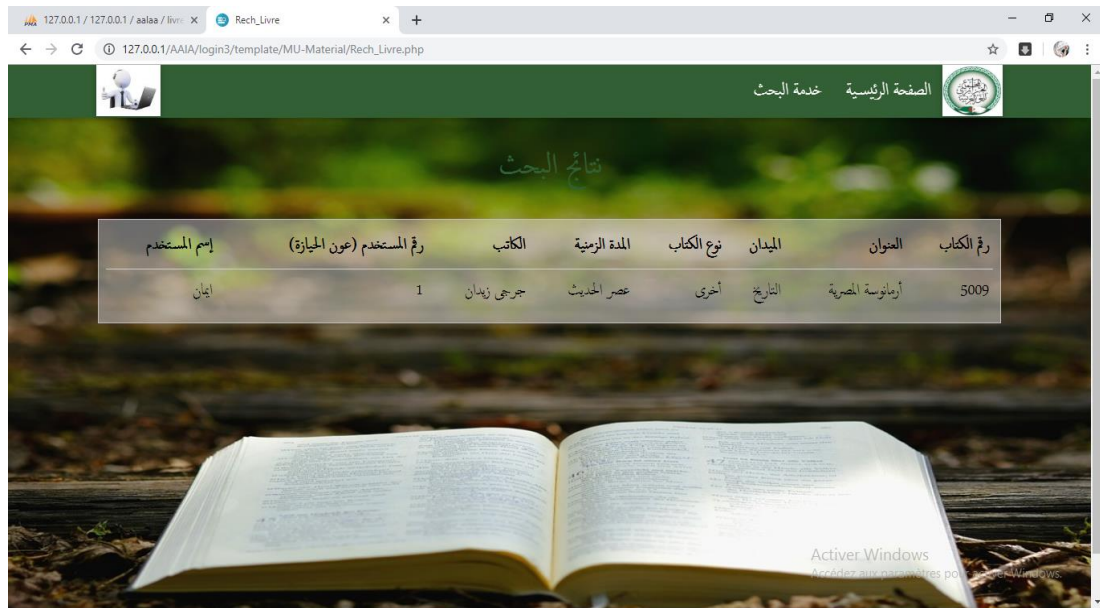


Figure 5.48 : Page de résultat de recherche d'un livre.

Si le livre n'existe pas, un message sera affiché :

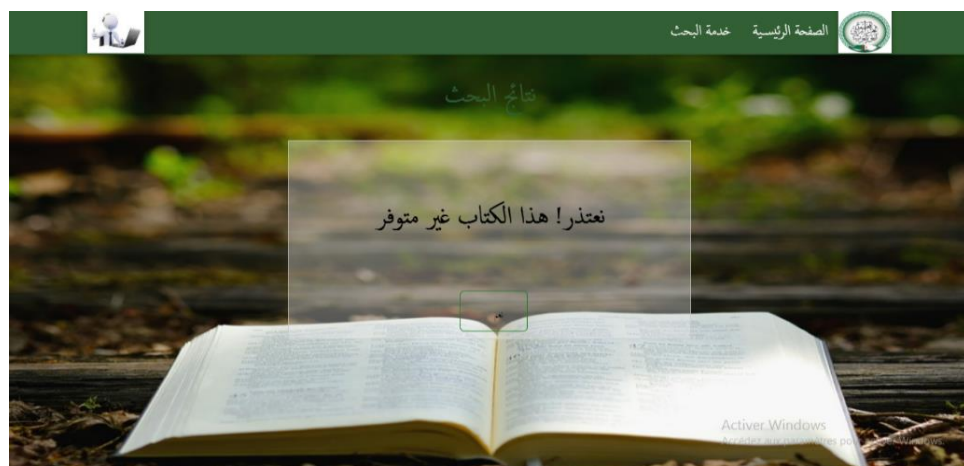


Figure 5.49 : Message de non disponibilité du livre.

- Si vous cherchez un utilisateur, saisissez "رقم المستخدم" ou "إسم المستخدم" dans la barre de recherche « البحث عن مستخدم » le résultat sera affiché en deux parties :

La 1^{ère} partie : les informations personnelles d'utilisateur :



Figure 5.50 : Page de résultat 1 de recherche d'utilisateur.

La 2eme partie : les informations du livre saisi par cet utilisateur, en cliquant sur le bouton qui s'affiche au milieu de la page, le résultat obtenu est le suivant :



Figure 5.51 : Page de résultat 2 de recherche d'utilisateur.

Si l'utilisateur n'existe pas, un message sera affiché :



Figure 5.52 : Message de non disponibilité d'utilisateur.

3.3.4. Les Statistique

Cliquez sur le bouton "خدمة الإحصائيات" pour avoir toutes les statistiques qui se définissent comme suit :

- Nombre des livres saisis par opérateur ;

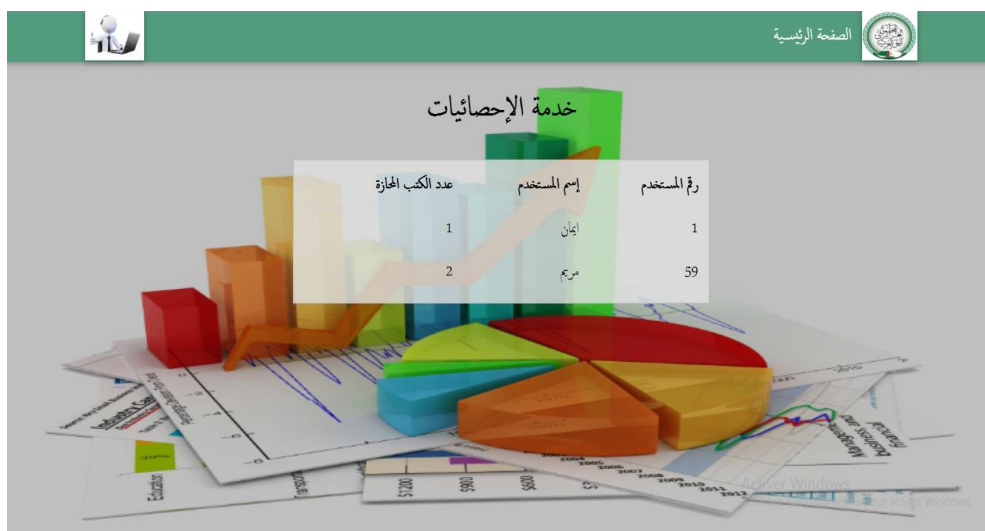


Figure 5.53 : Nombre des livres saisis par opérateur.

- Nombres d'erreurs faites par livre / par opérateur ;

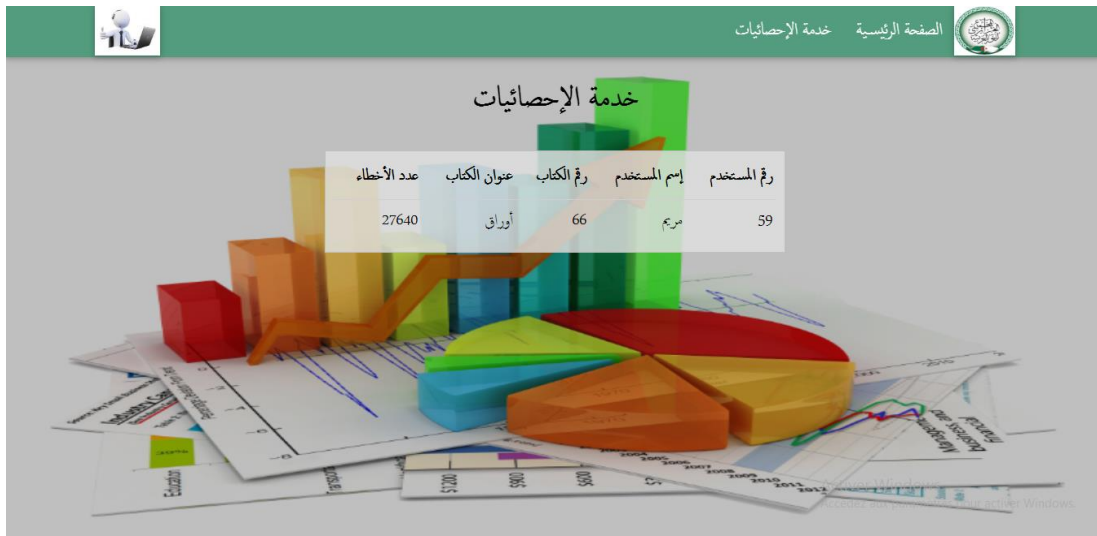


Figure 5.54 : Nombres d'erreurs faites par livre / par opérateur.

- Durée de saisie ;

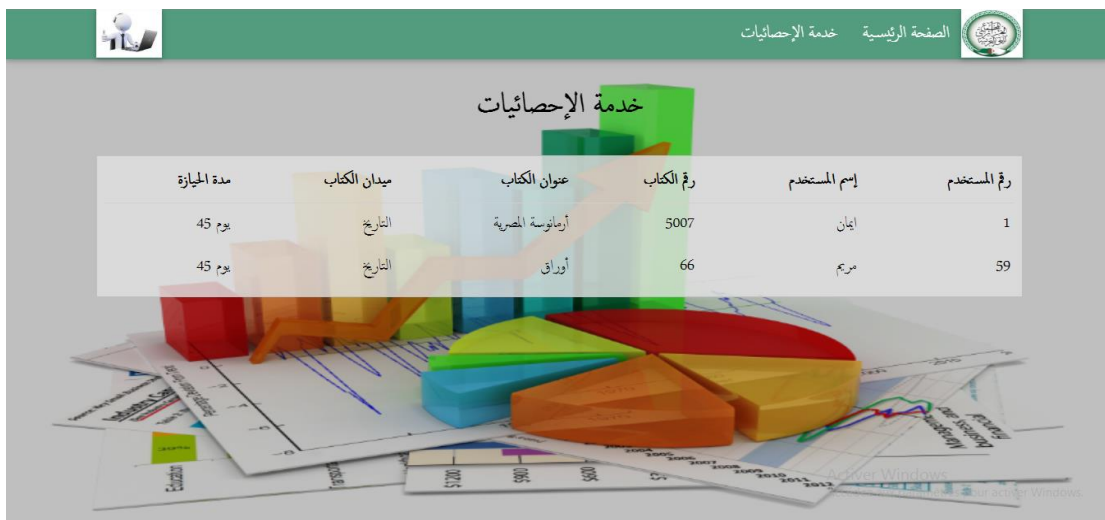


Figure 5.55 : Durée saisie de livre.

- Pourcentage d'erreurs par livre pour chaque opérateur ;

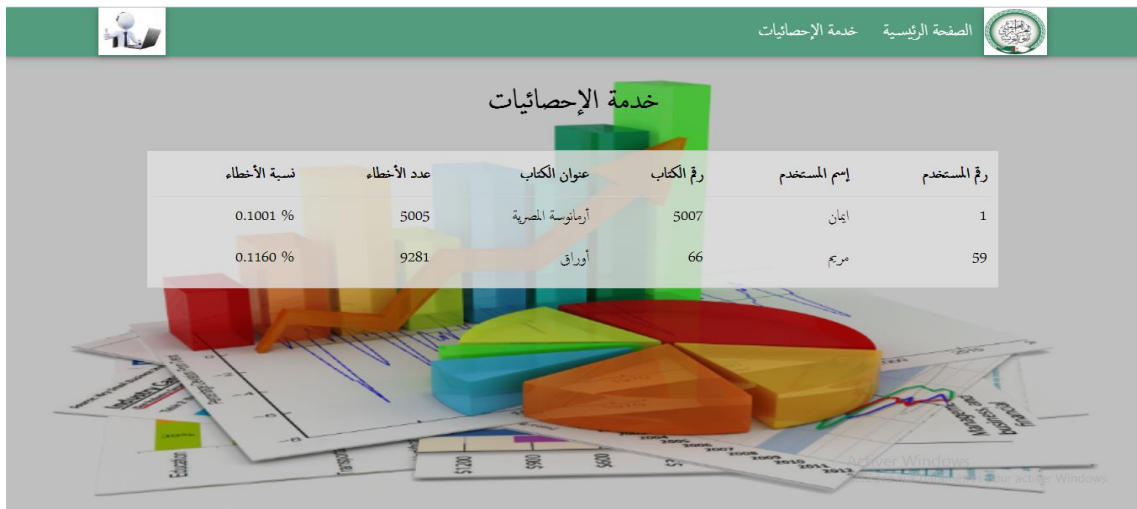


Figure 5.56 : Pourcentage d'erreurs par livre / opérateur.

- Pourcentage d'erreurs par domaine de livre pour chaque opérateur ;

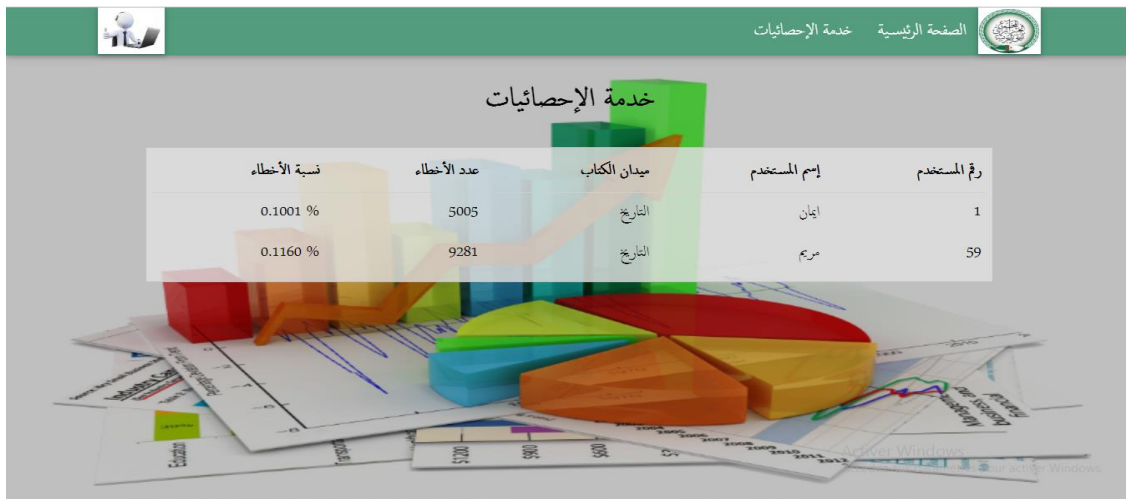


Figure 5.57 : Pourcentage d'erreurs par domaine / opérateur.

- Toutes les erreurs faites avec ses types et ses catégories pour chaque livre par opérateur.



Figure 5.58 : L'intégralité des erreurs.

3.3.5. Liste des livres

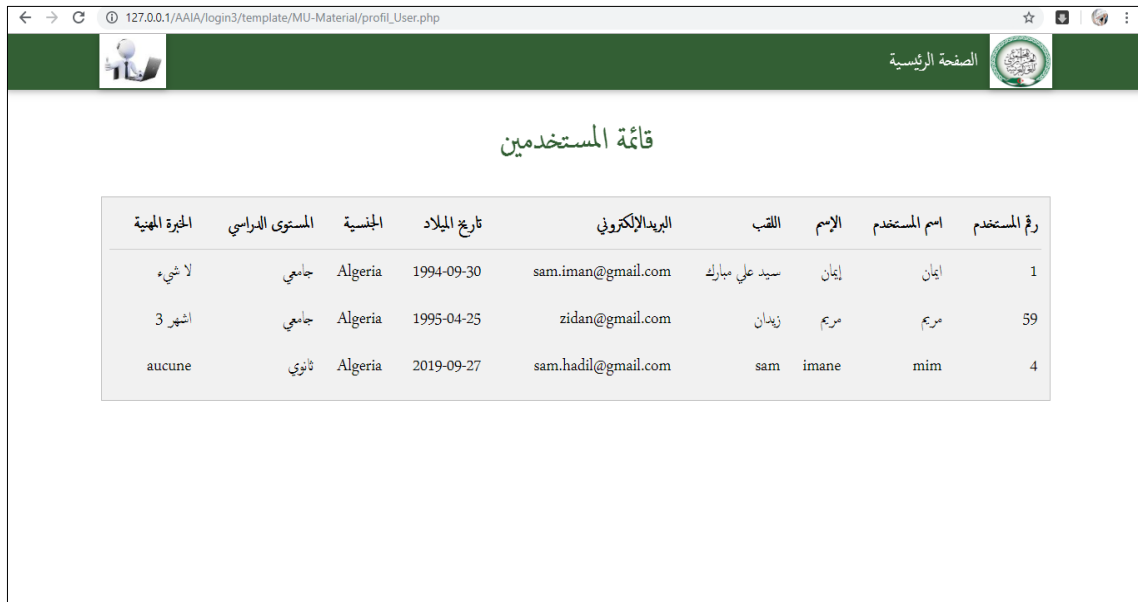
Cliquez sur le bouton "قائمة الكتب", une page contient toutes les livres stockés dans la BDD s'affiche comme suit :

رقم الكتاب	العنوان	الميدان	نوع الكتاب	المدة الزمنية	الكاتب	البلد المنشأ	مكان النشر	دار النشر	عدد الصفحات	عدد الكلمات	اسم عون الحيازة
5016	أوراق	التاريخ	قصة	العصر الحديث	عبد الله الغروي	المغرب	المغرب	المركز الثقافي العربي للنشر والتوزيع	500	80000	مريم
5015	صنف إفريقي	التاريخ	أخرى	عصر الحديث	محمد ديب	الجزائر	سوريا	مكتبة الأطلس	350	100000	مريم
5009	أروانوسة المصرية	التاريخ	رواية	عصر الحديث	جرجي زيدان	مصر	مصر ((القاهرة))	مؤسسة هنداوي للتعليم والثقافة	250	50000	ايمان

Figure 5.59 : Page de la liste des livres.

3.3.6. Liste des opérateurs de saisis

Cliquez sur le bouton "البيانات الشخصية للمستخدمين", une page contient toutes les informations des opérateurs de saisis stockés dans la BDD s'affiche comme suit :



رقم المستخدم	اسم المستخدم	الاسم	اللقب	البريد الإلكتروني	تاريخ الميلاد	الجنسية	المستوى الدراسي	المهنة المهنية	الاختصاص
1	ايمان	إيمان	سيد علي مبارك	sam.iman@gmail.com	1994-09-30	Algeria	جامعي	لا شيء	
59	مريم	مريم	زيدان	zidan@gmail.com	1995-04-25	Algeria	جامعي	اشهر 3	
4	mim	imane	sam	sam.hadil@gmail.com	2019-09-27	Algeria	ثانوي	aucune	

Figure 5.60 : Profil d'utilisateurs.

3.4.Scénario d’usage d’un système d’aide à la correction

On présente dans ce scénario les différentes étapes de déroulement du processus de versionning des documents.

On ouvrant le système l’interface globale du système se figure :

Figure 5.61 : Interface globale du système d'aide à la correction.

Tout d’abord, cliquez sur "إضافة كتاب قبل التصحيح" afin d’ajouter la 1^{ère} version du livre (avant la correction), cliquez ensuite sur "إضافة كتاب بعد التصحيح" pour ajouter la 2eme version du livre (après la correction). Ces livres sont t’exporter depuis la GED.



Figure 5.62 : L'interface du versionning.

Pour déterminer le nombre total des erreurs faites dans ce livre il faut cliquer sur "عدد الأخطاء" :



Figure 5.63 : Evaluer le nombre d'erreurs.

Pour ajouter le rapport d'erreurs de ce livre, vous créez tout d'abord un fichier texte (.txt) à l'emplacement du livre.

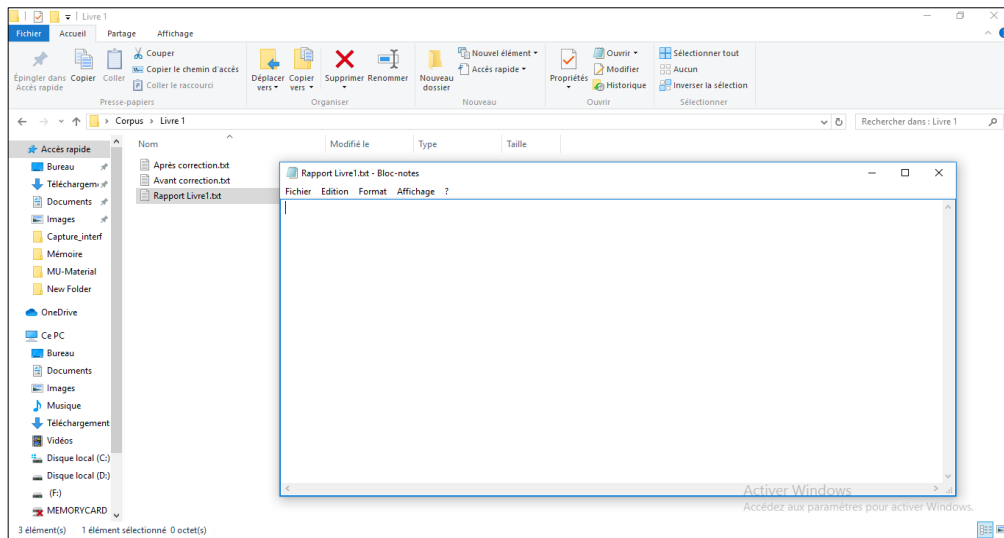


Figure 5.64 : La création du rapport d'erreurs.

Par la suite, en cliquant sur "إضافة التقرير للكتاب", une fois que le fichier est sélectionné il sera ouvert :

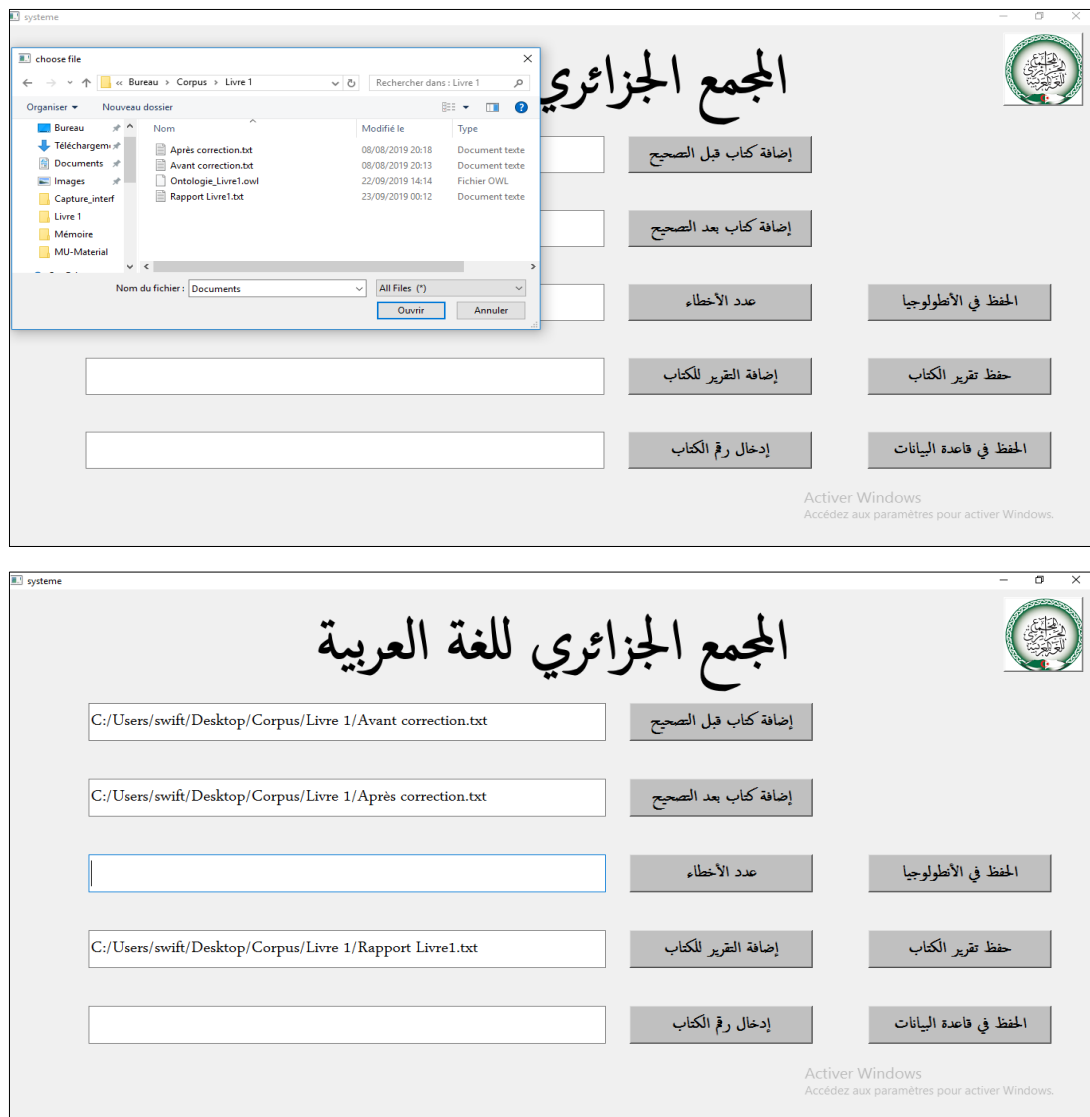


Figure 5.65 : L'ouverture du rapport.

Enfin, cliquez sur "حفظ تقرير الكتاب" pour stocker le rapport :

```

Rapport livr1.txt - Bloc-notes
Fichier Edition Format Affichage Aide
{"moter": "انه", "motcr": "انه", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "تشدد", "motcr": "تشدد", "type": "autre", "categorie": "autre"},
{"moter": "تشدد", "motcr": "تشدد", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "كل", "motcr": "كل", "type": "autre", "categorie": "autre"},
{"moter": "كل", "motcr": "كل", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "يكنر", "motcr": "يكنر", "type": "autre", "categorie": "autre"},
{"moter": "يكنر", "motcr": "يكنر", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "بالري", "motcr": "بالري", "type": "autre", "categorie": "autre"},
{"moter": "بالري", "motcr": "بالري", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "وان", "motcr": "وان", "type": "الهجرة ناقصة", "categorie": "الاملاء رسم الهجرة"},
{"moter": "وان", "motcr": "وان", "type": "autre", "categorie": "autre"},
{"moter": "وان", "motcr": "وان", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "انه", "motcr": "انه", "type": "الهجرة ناقصة", "categorie": "الاملاء رسم الهجرة"},
{"moter": "انه", "motcr": "انه", "type": "autre", "categorie": "autre"},
{"moter": "انه", "motcr": "انه", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "السن", "motcr": "السن", "type": "autre", "categorie": "autre"},
{"moter": "السن", "motcr": "السن", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "اذ", "motcr": "اذ", "type": "الهجرة ناقصة", "categorie": "الاملاء رسم الهجرة"},
{"moter": "اذ", "motcr": "اذ", "type": "autre", "categorie": "autre"},
{"moter": "فان", "motcr": "فان", "type": "الهجرة ناقصة", "categorie": "الاملاء رسم الهجرة"},
{"moter": "فان", "motcr": "فان", "type": "autre", "categorie": "autre"},
{"moter": "فان", "motcr": "فان", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "يسددها", "motcr": "يسددها", "type": "autre", "categorie": "autre"},
{"moter": "يسددها", "motcr": "يسددها", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "كل", "motcr": "كل", "type": "autre", "categorie": "autre"},
{"moter": "كل", "motcr": "كل", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "كانها", "motcr": "كانها", "type": "autre", "categorie": "autre"},
{"moter": "كانها", "motcr": "كانها", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "صينية", "motcr": "صينية", "type": "autre", "categorie": "autre"},
{"moter": "صينية", "motcr": "صينية", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "يسديتها", "motcr": "يسديتها", "type": "autre", "categorie": "autre"},
{"moter": "يسديتها", "motcr": "يسديتها", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "وتامل", "motcr": "وتامل", "type": "autre", "categorie": "autre"},
{"moter": "وتامل", "motcr": "وتامل", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"},
{"moter": "اي", "motcr": "اي", "type": "autre", "categorie": "autre"},
{"moter": "اي", "motcr": "اي", "type": "علامة الشدة ناقصة", "categorie": "الاملاء علامة التشكيل"}

```

Figure 5.66 : Le rapport d'erreurs.

Cliquez sur "إدخال رقم الكتاب" afin de remplir l'Id_livre qui correspond à le même id stocker dans la BDD du premier système :

systeme

المجمع الجزائري للغة العربية



C.:/Users/swift/Desktop/Corpus/Livre 1/Avant correction.txt

إضافة كتاب قبل التصحيح

C.:/Users/swift/Desktop/Corpus/Livre 1/Après correction.txt

إضافة كتاب بعد التصحيح

4452

عدد الأخطاء

الحفظ في الأطلوجيا

C.:/Users/swift/Desktop/Corpus/Livre 1/Rapport Livre1.txt

إضافة التقرير للكتاب

حفظ تقرير الكتاب

5000

إدخال رقم الكتاب

الحفظ في قاعدة البيانات

Activer Windows

Accédez aux paramètres pour activer Windows.

Figure 5.67 : Récupération d'ID.

A la fin, le bouton "الحفظ في قاعدة البيانات" est désigné pour stocker toutes les erreurs faites ainsi que leurs types et catégories sur la table « Erreur » de la BDD du système d'aide à la décision, ces derniers vont être utilisés pour générer un rapport (Figure 58).

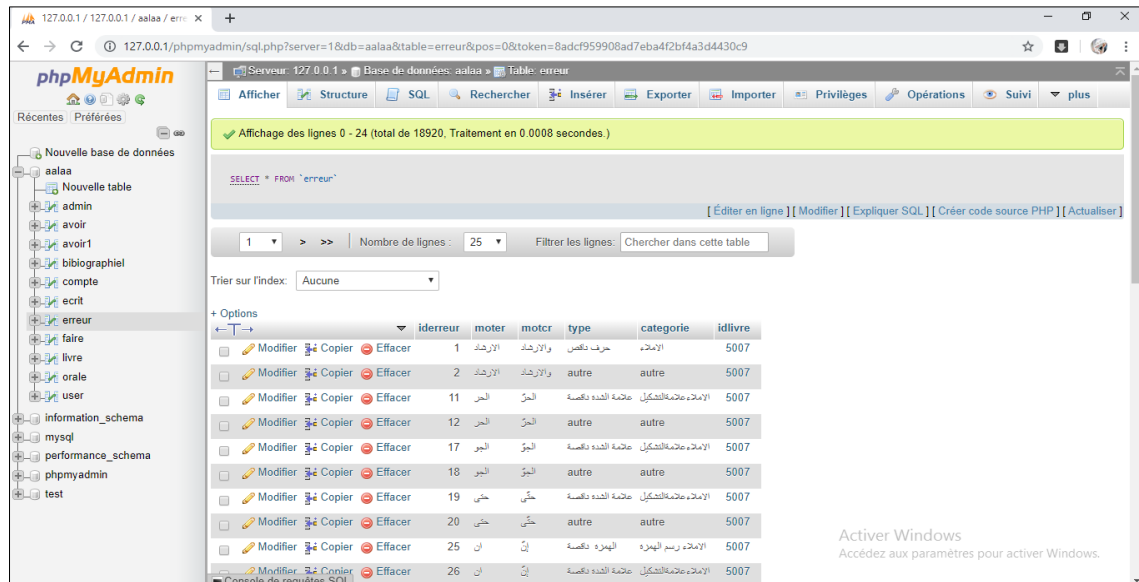


Figure 5.68: Stockage d'erreurs.

3.5.Construction d'une base de connaissance « Ontologie »

Afin de construire une Ontologie, on a utilisé un éditeur d'Ontologie "Protégé" pour la création du suivant :

- Les classes

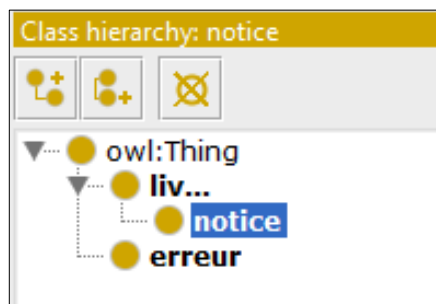


Figure 5.69 : Construction des classes.

- Les attributs

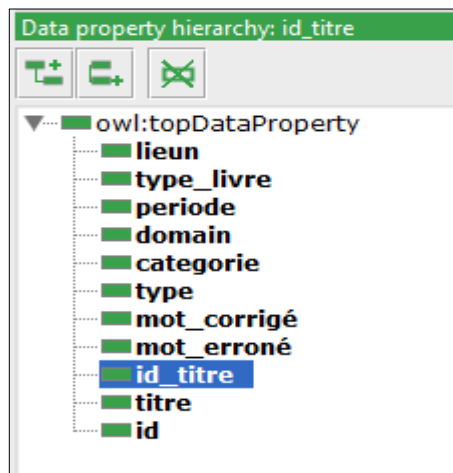


Figure 5.70 : Construction des attributs.

- Les relations

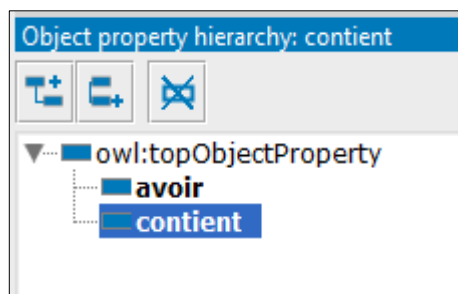


Figure 5.71 : Construction des relations.

Après avoir créé l'Ontologie, un fichier (.OWL) contenant les classes, les attributs et les relations va être créé automatiquement.


```

69 <SubClassOf>
70 <Class IRI="#erreur"/>
71 <DataMaxCardinality cardinality="50000">
72 <DataProperty IRI="#categorie"/>
73 </DataMaxCardinality>
74 </SubClassOf>
75 <SubClassOf>
76 <Class IRI="#erreur"/>
77 <DataMaxCardinality cardinality="50000">
78 <DataProperty IRI="#id"/>
79 <DataProperty abbreviatedIRI="owl:rational"/>
80 </DataMaxCardinality>
81 </SubClassOf>
82 <SubClassOf>
83 <Class IRI="#erreur"/>
84 <DataMaxCardinality cardinality="50000">
85 <DataProperty IRI="#mot_corrigé"/>
86 <DataProperty abbreviatedIRI="owl:rational"/>
87 </DataMaxCardinality>
88 </SubClassOf>
89 <SubClassOf>
90 <Class IRI="#erreur"/>
91 <DataMaxCardinality cardinality="50000">
92 <DataProperty IRI="#mot_erreur"/>
93 <DataProperty abbreviatedIRI="owl:rational"/>
94 </DataMaxCardinality>
95 </SubClassOf>
96 <SubClassOf>
97 <Class IRI="#erreur"/>
98 <DataMaxCardinality cardinality="50000">
99 <DataProperty IRI="#type"/>
100 <DataProperty abbreviatedIRI="owl:rational"/>
101 </DataMaxCardinality>
102 </SubClassOf>

```

length : 8125 lines : 252 Ln : 153 Col : 28 Sel : 0 | 0 UNIX UTF-8 INS

```

34 <!-- Les Attributs -->
35 <Declaration>
36 <DataProperty IRI="#categorie"/>
37 </Declaration>
38 <Declaration>
39 <DataProperty IRI="#domain"/>
40 </Declaration>
41 <Declaration>
42 <DataProperty IRI="#id"/>
43 </Declaration>
44 <Declaration>
45 <DataProperty IRI="#id_titre"/>
46 </Declaration>
47 <Declaration>
48 <DataProperty IRI="#lieux"/>
49 </Declaration>
50 <Declaration>
51 <DataProperty IRI="#mot_corrigé"/>
52 </Declaration>
53 <Declaration>
54 <DataProperty IRI="#mot_erreur"/>
55 </Declaration>
56 <Declaration>
57 <DataProperty IRI="#periode"/>
58 </Declaration>
59 <Declaration>
60 <DataProperty IRI="#titre"/>
61 </Declaration>
62 <Declaration>
63 <DataProperty IRI="#type"/>
64 </Declaration>
65 <Declaration>
66 <DataProperty IRI="#type_livre"/>
67 </Declaration>

```

Figure 5.72 : Création d'un fichier OWL vide.

Une fois que le bouton "الحفظ في الأنطولوجيا" trouvé à l'interface du système d'aide à la correction soit appuyé, le fichier (.OWL) sera rempli.

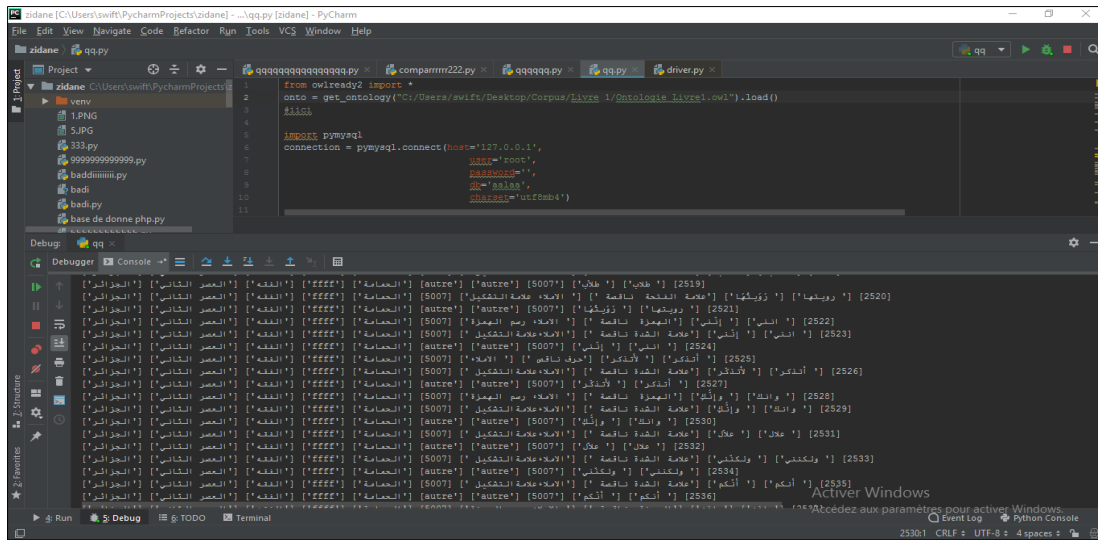


Figure 5.73 : Code de remplissage du fichier OWL.

Finalement l'Ontologie sera établie :

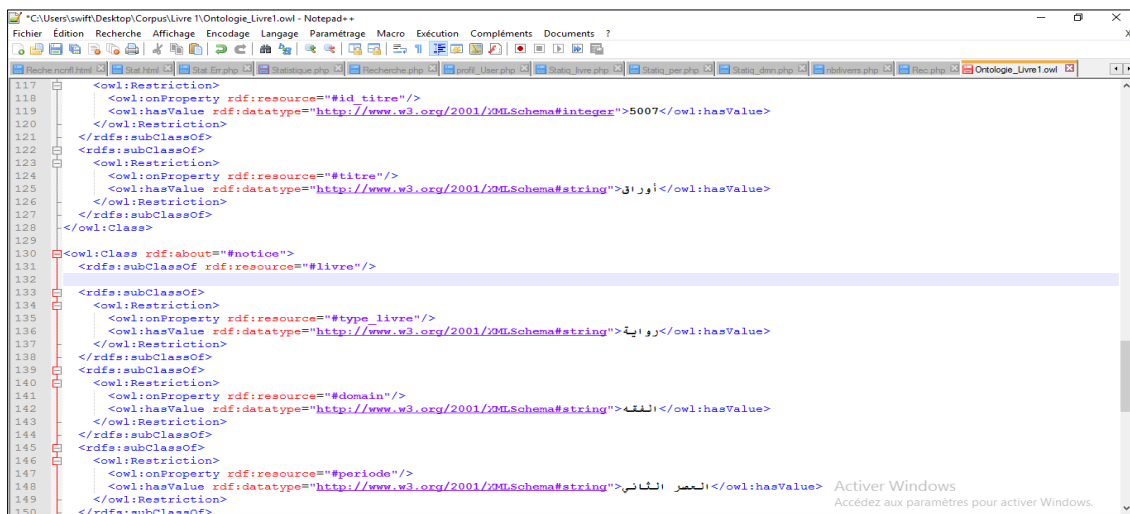


Figure 5.74 : Construction d'Ontologie.

Conclusion

A ce chapitre, notre projet d'études atteint sa fin. Tout au long de ce chapitre, nous avons abordé notre environnement de travail. Par la suite, nous avons expliqué notre architecture du système afin de présenter finalement les différentes principales parties d'implémentation de notre système réalisée.

Conclusion générale

L'objectif de notre projet de fin d'étude avait pour but de concevoir et de réaliser un outil de comparaison des documents selon deux états (avant et après la correction) afin de créer une base de connaissance linguistique pour la langue arabe. Notre mémoire est composé de deux phases : une phase théorique qui a englobé trois chapitres et une phase pratique incluse dans les deux derniers chapitres.

Dans le premier chapitre nous avons fait une description détaillée sur les systèmes de gestion électronique de documents liée à un choix précis pouvant répondre aux besoins de la cellule de numérisation manuelle de la Dhakhira al-arabiyya, et cela en examinant la possibilité d'effectuer les fonctionnalités suivantes par le système choisit :

- Gestion du flux d'échange des documents produits.
- Stockage des documents.

Le deuxième chapitre a été consacré au traitement automatique de la langue arabe ainsi que les problèmes cernés dans le domaine du TALA pour arriver à la correction orthographique des erreurs.

Ensuite, nous avons expliqué en détail, dans le troisième chapitre, les différentes notions relatives aux Ontologies. Cette dernière (Ontologie) a été adoptée comme une solution pour la construction d'une base de connaissance contenant les erreurs de types orthographique afin de réutiliser ses connaissances dans les autres fonctionnalités de la plate-forme.

Quant à la conception, nous avons présenté la solution répondante au besoin exprimé par les porteurs du projet, cela vise à améliorer le travail de la cellule de numérisation. Notre solution se base sur la dotation d'une GED et le développement d'un système qui permet de :

- Utiliser un outil d'aide à la correction qui permet de détecter l'erreur et proposer une forme corrigé en appliquant une règle donnée ;
- Construire un outil d'aide à la décision pour assurer une meilleure affectation des livres ;

- Construire une base de connaissance « Ontologie » pour stocker les informations générées.

Pour conclure le dernier chapitre, notre projet d'étude atteint sa fin. Tout au long de ce chapitre, nous avons implémenté notre système qui permet de faire le versionning afin de déterminer un rapport d'erreurs pour chaque livre et l'enregistrer automatiquement dans la base de connaissance (Ontologie) ainsi que la BDD, ce rapport sera utilisé comme un outil d'aide à la décision.

A la fin de la réalisation de notre projet on a pu atteindre les points suivants :

- Organiser le travail collaboratif de la cellule de numérisation à travers la plateforme GED ;
- Construire un outil de comparaison des livres pour déterminer le rapport d'erreurs ;
- Construire un outil d'aide à la décision pour assurer une meilleure affectation des livres ;
- Construire une base de connaissance « Ontologie » pour stocker les informations générées.

En conclusion, ce travail nous a été très formateur, puisqu'il a permis de découvrir une nouvelle technologie innovante, et nous a permis également de se confronter à plusieurs contraintes à la fois : contraintes de temps, contraintes d'expérience et de technologie, En outre, ce projet nous a permis d'approfondir nos connaissances dans les bonnes pratiques informatique.

Perspectives

Notre réalisation est encore d'actualité. En effet plusieurs perspectives s'offrent à ce projet :

Pour rendre l'application plus expansive, le système doit être connecté avec la GED pour la récupération directe des données. Cela a été prévu après avoir une formation sur Alfresco.

Finalement, vu l'accomplissement de projet, nous souhaitons très fortement qu'il soit le fruit du progrès, de l'évolution et qu'il reste à la hauteur des exigences de l'académie.

Bibliographie

- [«<http://www.amoks.com/rep-lexique/ido-325/ged.html>».
1
]
- [«http://igm.univ-mlv.fr/~dr/XPOSE2013/Approche_de_la_gestion_documentaire/concept_ged.html».
2
]
- [«<https://fr.slideshare.net/babakourouma/conduire-un-projet-de-ged-concepts-de-base-points-de-repre-pour-la-mise-en-uvre>».
3
]
- [«<https://www.qualishare.fr/qualishare-logiciel-qse-metier-collaboratif-%20sharepoint/gestion-electronique-des-documents/>».
4
]
- [«<https://www.pulsar-informatique.com/services/ged-open-source/pourquoi-une-ged>».
5
]
- [«<https://www.pulsar-informatique.com/services/ged-open-source/technologies-et-solutions-de-ged-open-source>».
6
]
- [«http://www.starxpert.fr/PDF/Alfresco_vs_SharePoint.pdf».
7
]
- [«<https://fr.wikipedia.org/wiki/Alfresco>».
8
]
- [«<https://community.alfresco.com/docs/DOC-6139-enterprise-edition>».
9
]
- [«A., Blache P. Grammaires et analyseurs syntaxiques, dans J.-M. Pierrel Édition
1 IngÈnierie des Langues, Paris, HermÈs, 2000 .».
0
]
- [«https://www.academia.edu/7434276/Impl%C3%A9mentation_dun_correcteur_orthographique_automatique_pour_lextraction_dinformation?fbclid=IwAR0YX_TRIAd3n2IfIP3TnjR8C7Qc8FAOWh5PHpmDcMMXiAvLbT6lQht6040».
1

1

]

[«Jean Veronis " Informatique et linguistique", Centre informatique pour les lettres et sciences humaines, Université de Provence, 2001 Space Odyssey».

2

]

[«Lionel Delafosse, Traitement automatique des langues, 1999.».

1

3

]

[«النحو العصرى الناشر مركز الأهرام تاريخ الطبعة 1995».

1

4

]

[«BOUILLON, PIERRETTE, VANDOOREN, FRANCOISE ET LEHMANN, SABINE (Eds.)(1998).

1 Traitement automatique des langues naturelles. Universités francophones. Champs

5 linguistiques. Recueils. Bruxelles: Duculot».

]

[«Mohammed Attia,Pavel Pecina ,Younes Samih,Khaled Shaalan,Josef van

1 Genabith,<https://www.cambridge.org/core/product/630859B3EC3365926DBAFF92436A>

6 89A6/core-reader le 1/06/2019».

]

[«Chapitre I -Les ontologies , [[http://dspace.univ-](http://dspace.univ-tlemcen.dz/bitstream/112/1062/5/ChapitreI.pdf)

1 tlemcen.dz/bitstream/112/1062/5/ChapitreI.pdf, le 13/.3/2019]».

7

]

[«[https://docplayer.fr/27517-Developpement-d-une-ontologie-101-guide-pour-la-](https://docplayer.fr/27517-Developpement-d-une-ontologie-101-guide-pour-la-creation-de-votre-premiere-ontologie.html)

1 creation-de-votre-premiere-ontologie.html, le 13/.3/2019».

8

]

[«[https://www.researchgate.net/figure/Exemple-de-constructeurs-en-logique-de-](https://www.researchgate.net/figure/Exemple-de-constructeurs-en-logique-de-description-et-leur-traduction-en-logique-du_fig10_30519234)

1 description-et-leur-traduction-en-logique-du_fig10_30519234».

9

]

[«https://www.academia.edu/4388491/Cours_Types_Ontologies».

2

0

]

[«<https://pdfs.semanticscholar.org/9fdf/ec07c83a8dba8ab5eb8129142ecbc8fcdf1a.pdf>
2 le 18/06/2019».

1

]

[«[https://www.researchgate.net/publication/327253257_Proposition_d'une_methode_d'
2 'ingenierie_ontologique_pour_les_EIAH_application_aux_systemes_auteurs,2014](https://www.researchgate.net/publication/327253257_Proposition_d'une_methode_d'ingenierie_ontologique_pour_les_EIAH_application_aux_systemes_auteurs,2014)».

2

]

[«M .Alain Polguère, Université de Lorraine & ATILF CNRS, [https://www.shs-
2 conferences.org/articles/shsconf/pdf/2014/05/shsconf_cmlf14_01383.pdf,2014](https://www.shs-conferences.org/articles/shsconf/pdf/2014/05/shsconf_cmlf14_01383.pdf,2014)».

3

]

[«M. Chouaib MOUKRIM, Tarik ALMALKI, Abderrahim TRAGHA, El Habib BENLAHMER,
2 Faculté des sciences Ben M'sik, Université Hassan II, [https://docplayer.fr/64334586-De-
4 la-generation-des-phrases-par-l-ontologie-a-la-correction-des-erreurs-
\] syntaxiques.html](https://docplayer.fr/64334586-De-la-generation-des-phrases-par-l-ontologie-a-la-correction-des-erreurs-syntaxiques.html)».

[<https://stph.scenari-community.org/idl-bd/idl-bd4.pdf>, 14/08/2019.

2

5

]

[h. 10/07/2019.

2

6

]

[<https://fr.wikipedia.org/wiki/StarUML>, 08/07/2019.

2

7

]

[<https://www.clubic.com/telecharger-fiche54328-edraw-max.html>, 05/09/2019.

2

8

]

[<https://notepad-plus-plus.org/fr/>, 30/08/2019.

2

9

]

[http://doplab.unil.ch/wp-content/uploads/2018/09/Pycharm_Tutorial-v1.pdf,

3 30/08/2019.

0
]

[(. e. ([https://fr.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_\(logiciel\)](https://fr.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_(logiciel))), 30/08/2019.

3
1
]

[[https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)), 08/07/2019.

3
2
]

[h.-t.-d.-w.-s.-s.-q.-l.-d.-t.-e.-a. 01/09/2019.

3
3
]

[<http://glossaire.infowebmaster.fr/php/>, 09/08/2019.

3
4
]

[h.-d.-l.-l.-a.-u.-u.-c.-e.-q. 02/09/2019.

3
5
]

[h.-a.-v.-b.-d.-d.-a.-m. 02/09/2019.

3
6
]

[
3
7
]

[«<https://independent.academia.edu/IJSTE>».

3
8
]

