
CENTRE UNIVERSITAIRE - SALHI AHMED - DE NAAMA

INSTITUT DES SCIENCES & TECHNOLOGIES

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE



MÉMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme

MASTER EN INFORMATIQUE

Spécialité : Système d'information

Thème

**Conception et réalisation d'une application Web
sémantique pour la gestion des annonces de produits**

Présentés par :

- ☞ ZELLATI badreddine
- ☞ BOUCHIHA mohammed tawfiq

Encadré par :

- ☞ BOUOUGADA Benamar

Année Universitaire : 2017-2018



REMERCIEMENTS

Premièrement, nous remercions Allah le tout puissant de nous avoir donné la volonté et le courage de mener à bien ce travail. Nous remercions également nos familles pour les sacrifices qu'elles ont fait pour que nous terminions nos études.

Nous remercions également tous nos professeurs qui nous ont aidé durant toutes nos années d'études notamment notre encadreur M^r BOUOUGADA Benamor, qui nous a beaucoup aidé à apprécier ce travail et as mieux éclairé nos perspectives, et de nous avoir conseillé tout le long de notre projet,

Nos adressons nos chaleureux remerciements à tous les membres de jury pour l'effort qu'ils feront dans le but d'examiner ce modeste travail.

Nous tenons à remercier vivement toutes les personnes qui nous ont aidé dans notre projet

Et enfin à tous nos collègues de la promotion 2017-2018.

BOUCHIHA & ZELLATI



Dédicace

Je dédie ce modeste travail, aux deux êtres les plus chers à mon

Cœur auxquels je dois mon existence :

Mon père et ma mère; vous qui étiez toujours à mes côtés pour

Me soutenir et m'encourager à embattre sans jamais m'arrêter à

Mi-chemin; que dieu vous protège.

- A ma grande mère.*
- A mes chers frères.*
- A toute ma famille.*
- A tous mes amis.*
- A mon binôme bouchiha mohammed tawfiq .*

Zellati badreddine



Dédicace

Je dédie ce modeste travail :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chères sœurs pour leurs encouragements permanents, et leur soutien moral,

A mes chers frères pour leur appui et leur encouragement

A tous mes amis.

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

A mon binôme zellati badreddine

Bouchiha Mohammed TawfiQ

Table des matières

Table des matières.....	1
Table des figures.....	
Listings.....	

ORGANISATION DU MEMOIRE

Chapitre I : Introduction générale	1
1. Introduction.....	2
2. Problématique	2
3. Motivation.....	3
4. Objectif	3
Chapitre II : Application Web	5
1. Introduction.....	6
2. Application web	6
2.1. Définition d'une application web	6
2.2. Définition http.....	7
2.3. Cycle de vie d'une application web	7
3. Les types d'application web.....	8
3.1. Site web statique	8
3.2. Site web dynamique.....	8
4. Les Langages de Programmation Web :	9
4.1. Les langages Web statique	9
4.1.1. HTML.....	9
4.1.2. CSS.....	9
4.1.3. JAVASCRIPT	9
4.2. Les langages Web dynamique	10
4.2.1. PHP.....	10
4.2.2. JSP.....	10
4.2.3. ASP.....	10
5. Conclusion	10
Chapitre III : Java EE	11
1. Introduction.....	12
2. Historique.....	12

3	Java EE.....	12
4.	Développement des Plateformes Java EE :.....	13
5.	Structure d'une application Java EE.....	14
6.	Java	15
7.	Java Server Pages (JSP).....	16
7.1.	Avantages	16
7.2.	Inconvénients.....	16
7.3.	Cycle de vie d'une JSP	16
8.	SERVLET	18
8.1.	Servlet.....	18
8.2.	Cycle de vie d'une servlet	18
8.3.	Comparaison Servlet/JSP	18
9.	Conclusion	19
	Chapitre IV: Web Sémantique.....	20
1.	Introduction.....	21
2.	Web Sémantique	21
3.	Objectif du web sémantique.....	22
4.	Architecture du Web sémantique.....	22
5.	Langages du Web sémantique.....	24
5.1.	XML	24
5.2.	RDF	25
5.3.	RDFS	26
5.4.	OWL.....	27
5.4.1.	OWL-Lite	27
5.4.2.	OWL-DL	27
5.4.3.	OWL-Full	27
5.5.	SPARQL.....	28
6.	Ontologie.....	28
7.	Les composants d'une ontologie.....	28
7.1.	Les concepts	28
7.2.	Les relations.....	28
7.3.	Les propriétés	29
7.4.	Les fonctions.....	29

7.5.	Les axiomes	29
8.	Les types d'ontologies	29
8.1.	Les ontologies générales.....	29
8.2.	Les ontologies du domaine	30
8.3.	Les ontologies d'application.....	30
9.	Exemple de manipulation ontologie OWL :	31
9.1.	Création avec Protégé.....	32
9.1.1.	Créer la classe Personne	32
9.1.2.	Créer 2 ^{eme} classe module :	32
9.1.3.	Créer SubClassOf Enseignant	33
9.1.4.	Créer les propriétés d'objets	33
9.1.5.	Créer les propriétés de données.....	35
9.2.	Création d'ontologie avec java	37
9.2.1.	Création des classes.....	37
9.2.2.	Création des sub-classes.....	37
9.2.3.	Création des object properties	38
9.2.4.	Création datatype properties.....	38
9.3.	Les Requête SPARQL :	39
9.3.1.	Requête insertion.....	39
9.3.2.	Requête suppression.....	39
9.3.3.	Requête update	39
10.	Conclusion.....	40
	Chapitre V : Conception.....	41
1.	Introduction.....	42
2.	UML.....	42
1.1.	Définition.....	42
1.2.	Les type de diagramme UML.....	42
1.2.1.	Diagrammes structurels ou diagrammes statiques (UML Structure).....	42
1.2.2.	Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior).....	43
1.2.3.	Diagrammes d'interaction (Interaction diagram).....	43
3.	Conception du système	43
3.1.	Les acteurs de notre projet.....	43
3.1.1.	Le visiteur.....	43

3.1.2.	Le Client	43
3.1.3.	L'administrateur	43
3.2.	Diagramme de cas d'utilisation	44
3.3.	Diagramme de Séquence	45
3.4.	Diagrammes d'Activité.....	50
3.5.	Diagrammes de classe	52
4.	L'ontologie proposée	53
5.	Conclusion	54
Chapitre VI : Implémentation		55
1.	Introduction.....	56
2.	Les outils utilisés.....	56
2.1.	Apache Tomcat.....	56
2.2.	Protégé.....	56
2.3.	Jena.....	56
2.4.	NetBeans.....	57
2.5.	StarUML.....	57
3.	Guide d'utilisation	58
3.1.	Page principal	58
3.2.	Page Inscription	59
3.3.	Page d'accueil de client	59
3.4.	Page création annonce	60
3.5.	Page des annonces de client.....	60
3.6.	Page détaillée d'une annonce	61
3.7.	Page « modifier les annonces de client ».....	61
3.8.	Page « modifier les informations de client »	62
3.9.	Page « authentification administrateur »	62
3.10.	Gestion des annonces.....	63
3.11.	Page de gestion des clients	63
3.12.	Page de modification du profile de l'administrateur	64
4.	Conclusion	64
Conclusion générale		65
Bibliographie.....		67

Liste Des Figures

Figure 1 : schéma du processus d'accéder au système	2
Figure 2 : évolution du web.....	3
Figure 3 : application Web	6
Figure 4 : Schéma d'une requête http	7
Figure 5 : Cycle de vie d'une application web	8
Figure 6 : Développement de Java EE.	13
Figure 7: schéma Structure d'une application Java EE	14
Figure 8 : Cycle de vie d'une JSP	16
Figure 9 : Le web du futur.....	21
Figure 10 : Architecture du Web sémantique.....	23
Figure 11 : extrait d'un document XML.	25
Figure 12 : Modèle de triplet.....	25
Figure 13 : Modèle de triplet en RDF	26
Figure 14 : Exemple RDFs.....	27
Figure 15 : Exemple Ontologie	31
Figure 16: Créer la classe Personne	32
Figure 17 : Créer la classe module	32
Figure 18 : Créer SubClassOf Enseignant.....	33
Figure 19 : Créer la propriété d'objets « enseigné par ».....	34
Figure 20 : Domain de Object Properties « enseigné par ».....	34
Figure 21 : Range de Object Properties enseigné par	35
Figure 22 : créer Data Prophéties Age	36
Figure 23 : Domain de Data Properties age	36
Figure 24 : Range de Data Properties age	37

Figure 25 : Diagramme de cas d'utilisation	44
Figure 26 : Diagramme de Séquence entre visiteur, client.....	45
Figure 27 : Diagramme de Séquence D'inscription Client.	45
Figure 28 : Diagramme de Séquence Authentification Client.	46
Figure 29 : Diagramme de Séquence Création annonce	46
Figure 30 : Diagramme de Séquence gestion profile du client	47
Figure 31 : Diagramme de Séquence log out	47
Figure 32 : Diagramme de Séquence Authentification administrateur.	48
Figure 33 : Diagramme de Séquence consulter les annonces par l'administrateur.....	48
Figure 34 : Diagramme de Séquence gestion profil d'un administrateur	49
Figure 35 : Diagramme d'Activité Client et visiteur.....	50
Figure 36 : Diagramme d'Activité d'administrateur	51
Figure 37 : Diagramme de classe.	52
Figure 38 : l'ontologie de gestion des annonces	53
Figure 39 : L'ontographe de l'ontologie de gestion des annonces	53
Figure 40 : L'écran d'accueil	58
Figure 41 : Page Inscription	59
Figure 42 : Page d'accueil de client	59
Figure 43 : création d'une annonce	60
Figure 44: Page des annonces client	60
Figure 45 : Page détaillée d'une annonce.....	61
Figure 46 : Page de modification des annonces des clients	61
Figure 47 : Page de modification les informations de client.....	62
Figure 48 : La page d'authentification de l'administrateur.....	62
Figure 49 : Gestion Des Annonces.....	63
Figure 50 : Page gestion des clients par l'administrateur.....	63
Figure 51 : Page de modification profile de l'administrateur	64

Listings

Listing 1 : la méthode jspInit	17
Listing 2 : la méthode jspService	17
Listing 3 : La méthode jspDestroy	17
Listing 4 :Création Ontologie	37
Listing 5 : Create classes	37
Listing 6 : créeate sub classes.....	37
Listing 7 : créeate object properties	38
Listing 8 : créer des propriétés de type de données	38
Listing 9 : Ajouter les données	39
Listing 10 : Supprimer les données	39
Listing 11 : Modifier les données	39

Chapitre I :

Introduction générale

1. Introduction

Le Web sémantique n'est pas un Web à part. Il est une extension du Web actuel et se fonde sur ce dernier. Il utilise donc toute l'infrastructure technique du Web actuel (les URI Uniform Resource Identifiers, le langage XML) en ajoutant de nouveaux langages tels que le RDF (Resource Description Framework) et OWL (Web Ontology Language) qui permettent de donner un sens bien défini à une information afin que les ordinateurs et les humains puissent collaborer

2. Problématique

La recherche par mots-clés est actuellement en train de vivre ses derniers jours, ce type de recherche n'étant plus du tout adapté à la masse d'informations disponible de nos jours sur le web. La solution se trouverait dans le web sémantique.

Le web sémantique va permettre à toutes les applications web de communiquer entre elles, et de se partager des données, en transformant ces données en métadonnées compréhensibles par les machines. Aussi l'intelligence ne se trouverait plus au cœur des applications mais au cœur des données elles-mêmes.

La figure1 montre le processus d'accéder au système ; il est composé de deux parties à savoir web sémantique et web actuel. Dans ce mémoire on s'intéresse à la première partie « web sémantique».

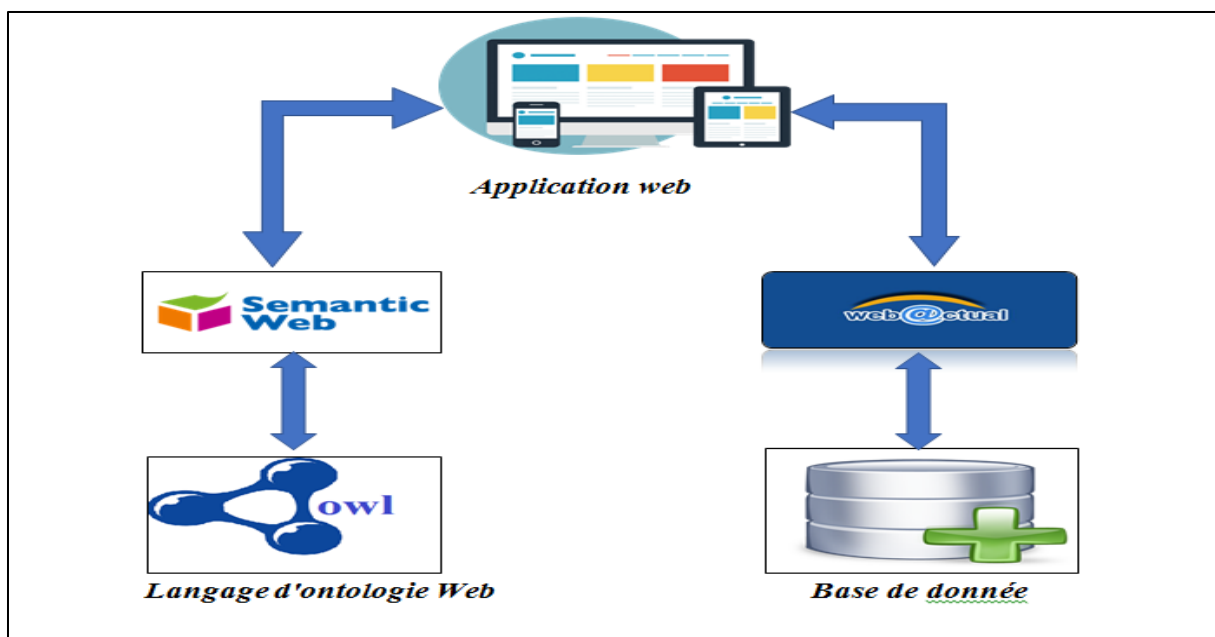


Figure 1 : schéma du processus d'accéder au système

3. Motivation

On a choisi le web sémantique parce qu' le web sémantique est la venir du web actuel selon les statistiques fournies par la figure 2 :

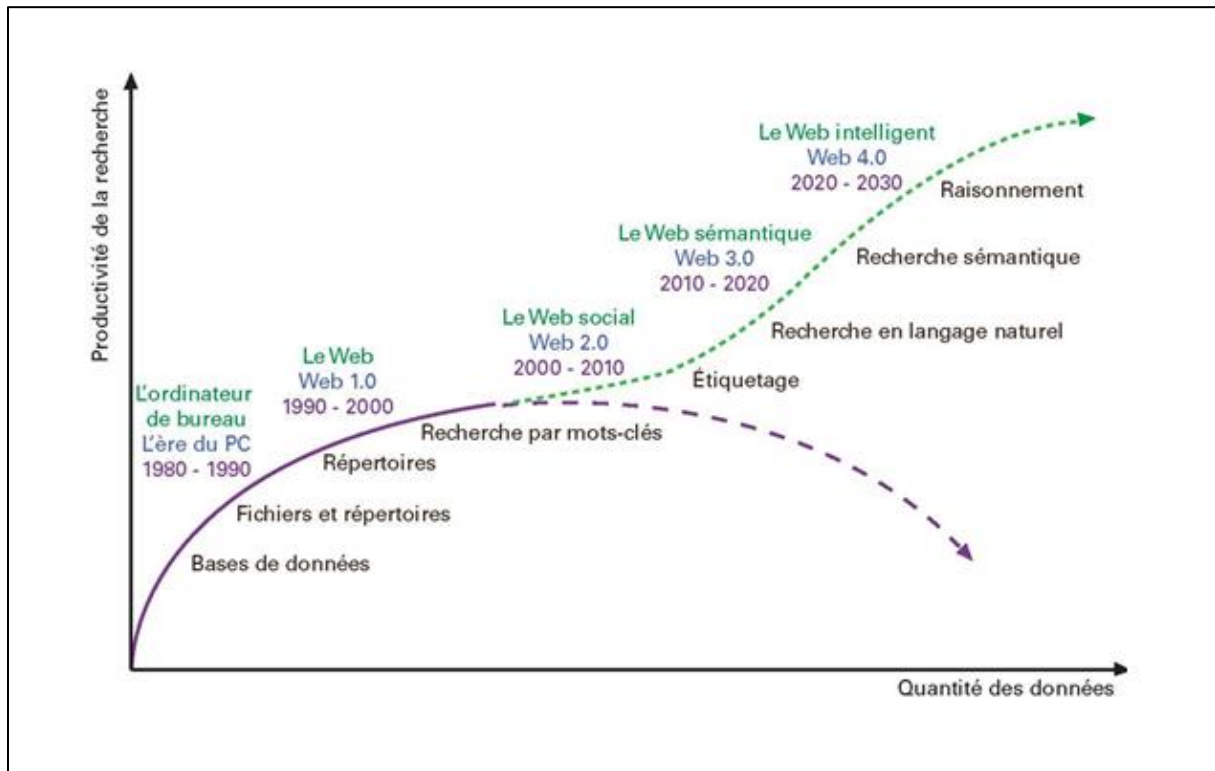


Figure 2 : évolution du web[1]

4. Objectif

Notre objectif est de développer une application Web basé sur le Web sémantique pour la gestion des annonces de produits. Afin d'achever cette tâche nous allons passer par une conception détaillée puis une implémentation. L'environnement utilisé pour l'implémentation est NetBeans IDE avec le package Jena pour traiter la partie sémantique c.à.d. tout qui est création et manipulation de l'ontologie

Organisation du mémoire

Ce mémoire est structuré en six chapitres :

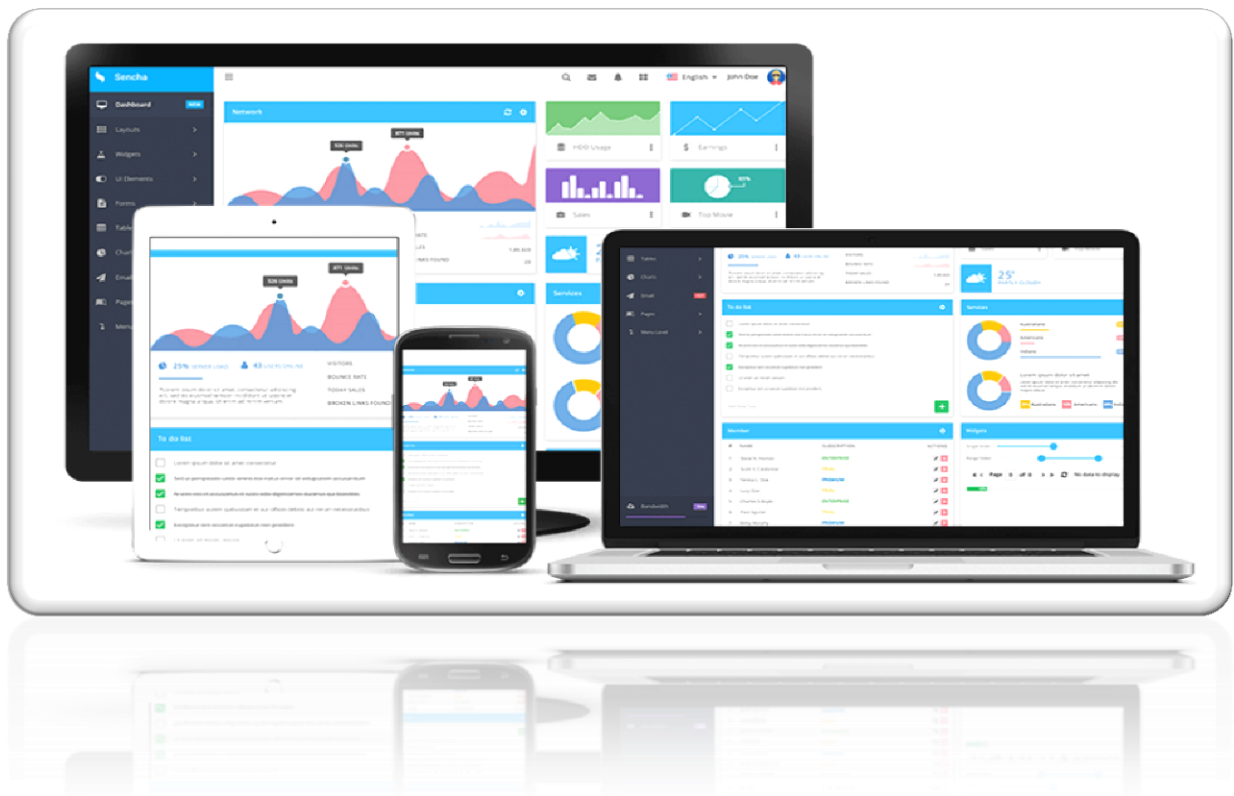
- Le premier chapitre parle sur Introduction générale.
- Le deuxième chapitre définit l'application web
- Le troisième chapitre s'intéresse aux Java EE

- Le chapitre quatrième présente une description du Web sémantique et sa pile de langage tel que RDF, RDFS, OWL
- Le cinquième chapitre concerne la conception de notre application.
- Le dernier chapitre permet de concevoir, réaliser et représenter graphiquement les activités des différentes interfaces du système.

Et finalement nous clôturons le mémoire par une conclusion générale qui résume notre travail.

Chapitre II :

Application Web



1. Introduction

Les applications Web sont très diverses et c'est précisément la variété des applications offerte aussi bien en matière d'information que de communication qui fait la force d'Internet.

Dans le cadre de ce chapitre, nous allons définir quelques généralités portant sur les concepts concernant une application web tels que les types des applications Web et les langages Web les plus utilisés.

2. Application web

2.1. Définition d'une application web

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur favori. La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs. Les technologies utilisés pour développer les applications web sont les mêmes que celles employées dans la création des sites internet.

Le Cloud Computing est donc ce phénomène en rapide extension qui vise à faire évoluer le modèle logiciel traditionnel vers internet et la téléphonie mobile s'intègre comme une extension d'utilisation à ce modèle. [2]

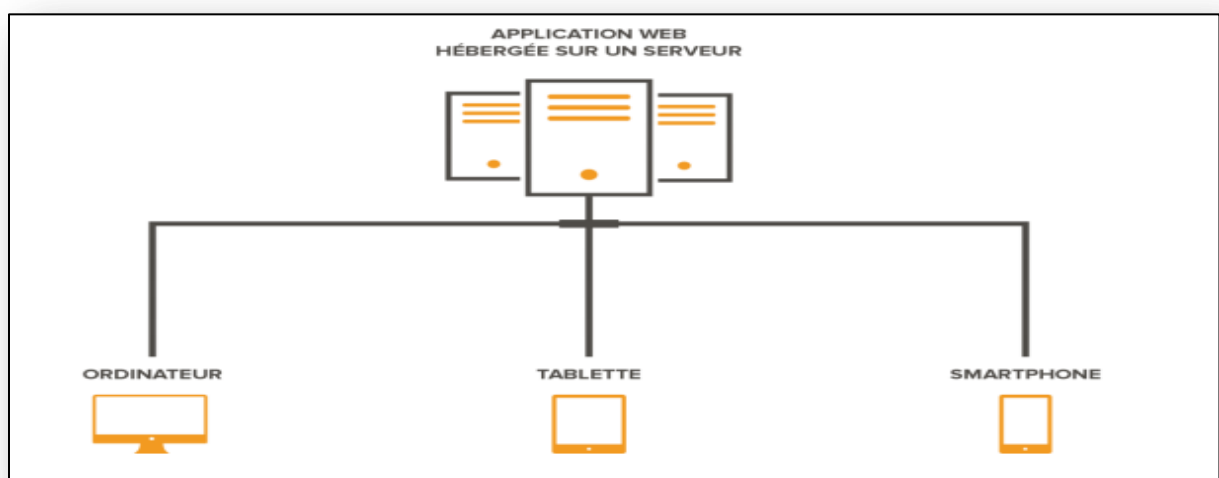


Figure 3 : application Web

2.2. Définition http

L'acronyme HTTP signifie HyperText Transfer Protocol (traduction : Protocole de transfert hypertexte). Ce protocole définit la communication entre un client (exemple: navigateur) et un serveur sur le World Wide Web (WWW). Ce protocole inventé par Tim-Berner Lee au début des années 1990, fonctionne sur le principe "requête-réponse". En prenant un exemple commun, de communication entre un navigateur web (le client) et un serveur web, la communication se déroule de la manière décrite sur le figure 4 :

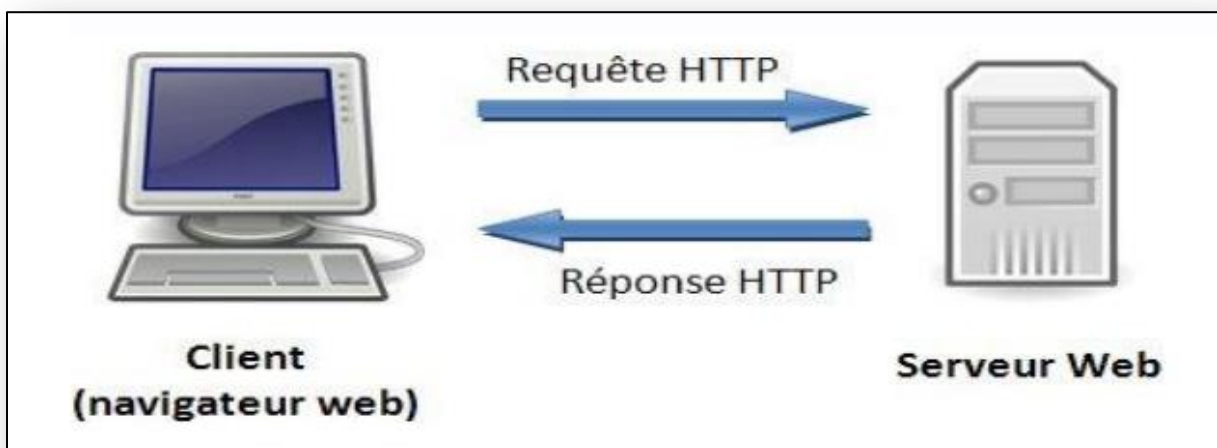


Figure 4 : Schéma d'une requête http

L'ordinateur de l'internaute utilise le navigateur pour envoyer une requête à un serveur web. Cette requête demande un document (exemple : page HTML, image, fichier CSS ...). Le serveur cherche les informations, puis il est peut-être amené à interpréter les résultats (exemple: PHP, Java ...), pour finalement envoyer la réponse. Cette réponse contient les entêtes du protocole HTTP et normalement le contenu demandé. [2]

2.3. Cycle de vie d'une application web

- **Analyse et définition des besoins** : les fonctionnalités du système et les contraintes sont établies en consultant les utilisateurs. Elles doivent être définies de façon compréhensive à la fois par les utilisateurs et par l'équipe de développement.
- **Conception de l'application** : c'est le processus qui consiste à représenter, et maquetter les diverses fonctions du système d'une manière qui permettra d'obtenir rapidement un ou plusieurs programmes réalisant ces fonctions.

- **Réalisation** : lors de cette étape, on réalise les programmes écrits dans un langage de programmation ou avec un CMS.
- **Intégration et tests du logiciel** : on intègre les unités de programmes et on réalise des tests globaux pour être sûr que les besoins logiciels ont été satisfaits, le système est alors livré.
- **Maintenance** : c'est le processus qui permet de garder opérationnel un logiciel ou de l'améliorer. [3]

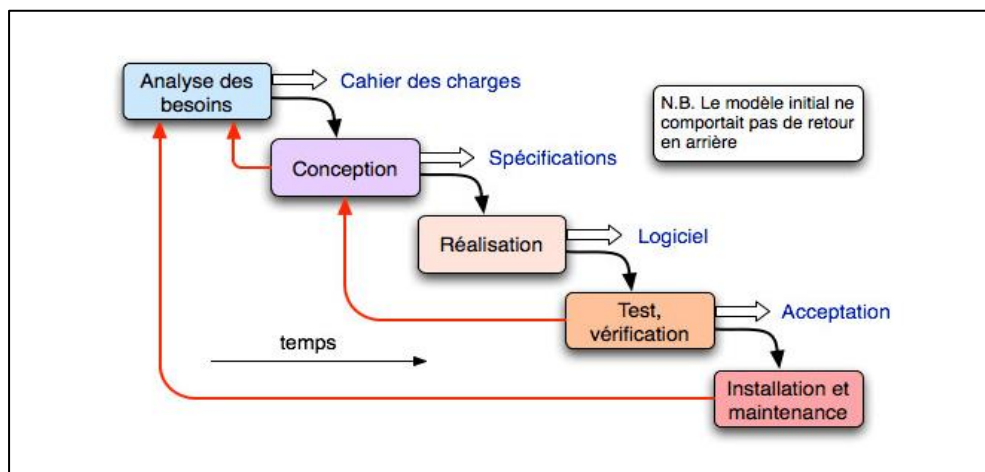


Figure 5 : Cycle de vie d'une application web

3. Les types d'application web

On considère qu'il existe deux types d'application web : les sites statiques et les sites dynamiques.

3.1. Site web statique

Site internet statique est un site composé de quelque page dont le contenu est écrit directement dans la page. Il faut choisir ce type de site si votre contenu n'a pas besoin d'être mis à jour régulièrement ou si les informations fournies par le site ne risquent pas de changer ou d'évoluer. Besoin : HTML pour le contenu et CSS pour la décoration.

3.2. Site web dynamique

Est un ensemble de pages dont le contenu est exclusivement stocké sur une base de données. Ce type de site web permet la modification du contenu à tout moment et en toute autonomie ce qui veut dire que vous aurez la possibilité de gérer votre site internet vous-même. Il peut avoir des galeries de photos, des rubriques d'actualités et de forum etc.... Les sites

dynamiques représentent un grand pourcentage de site présent sur internet car ils permettent la gestion facile e rapide, l'évolution et les modifications, l'ajout de photo, vidéo et tout type de média etc... Besoin : HTML pour le contenu et CSS pour la décoration + PHP et MySQL pour gérer les informations entre les utilisateurs et la base de donner au serveur.[2]

4. Les Langages de Programmation Web :

4.1. Les langages Web statique

4.1.1. HTML

Le HTML, acronyme d'*HyperText Markup Langague*, est le langage de base d'un site web. Il a été créé par Tim Berners-Lee au début des années 1990. Le HTML permet de créer des pages web au moyen de balises ouvrantes (`<html>`) et de balises fermantes (`</html>`). Ce balisage sert à définir la structure du texte entouré (titres, sous-titres, paragraphes, images, formulaires, liens). Les navigateurs (Chrome, Explorer, Mozilla, Safari) traduisent le langage HTML pour le rendre lisible pour les utilisateurs.[4]

4.1.2. CSS

La présentation visuelle des pages est définie par un langage complémentaire au HTML, soit le CSS (*Cascading Style Sheets*, ou *feuilles de style en cascade*). Ce langage, toujours sous forme de balises, permet d'associer des éléments visuels à la structure d'une page HTML : couleur, alignement, taille des polices, position des éléments.[4]

4.1.3. JAVASCRIPT

Le JavaScript est un langage de programmation de scripts orienté objet. Le JavaScript permet de programmer des scripts. Comme dit plus haut, un langage de programmation permet d'écrire du code source qui sera analysé par l'ordinateur. Les scripts sont majoritairement interprétés. Et quand on dit que le JavaScript est un langage de scripts, cela signifie qu'il s'agit d'un langage interprété ! Il est donc nécessaire de posséder un interpréteur pour faire fonctionner du code JavaScript, et un interpréteur, vous en utilisez un fréquemment : il est inclus dans votre navigateur Web ! Chaque navigateur possède un interpréteur JavaScript, qui diffère selon le navigateur. Si vous utilisez Internet Explorer, son interpréteur JavaScript

s'appelle JScript (l'interpréteur de la version 9 s'appelle Chakra), celui de Mozilla Firefox se nomme Spider Monkey et celui de Google Chrome est V8. [5]

4.2. Les langages Web dynamique

4.2.1. PHP

Le PHP (*HyperText Preprocessor*) est un langage de programmation qui s'intègre dans les pages HTML d'un site web. Il est à la base des sites web dynamiques, comme Facebook et Wikipédia, ou encore les sites e-commerce. La principale fonction de ce langage est d'automatiser des tâches répétitives, par exemple la gestion de nouvelles, d'articles, de galeries photos, de forums ou de moteurs de recherche à l'intérieur d'un site web.[4]

4.2.2. JSP

Le langage JSP est un langage de programmation particulier, doté d'un ensemble d'instructions qui permettent l'affichage de données textuelles ou numériques à l'intérieur d'une page web. Le langage JSP est en réalité un langage de scripts, ce qui signifiait qu'il est composé d'instructions Java directement placées dans le code d'une page HTML. Afin de différencier le code HTML du code Java, il est nécessaire d'entourer les instructions par des mots-clés spécifiques appelés étiquettes ou encore balises. Les instructions Java placées entre ces balises deviennent alors du code JSP.[6]

4.2.3. ASP

ASP (Active Server Pages) est un standard mis au point par Microsoft en 1996 permettant de développer des applications Web interactives, c'est-à-dire dont le contenu est dynamique. Ainsi une page web ASP (fichier repérable par l'extension .asp) aura un contenu pouvant être différent selon certains paramètres (des informations stockées dans une base de données, les préférences de l'utilisateur, ...) tandis qu'une page web « classique » (dont l'extension est .htm ou .html) affichera continuellement la même information.[7]

5. Conclusion

Nous avons cherché dans ce chapitre à présenter les concepts les plus connexes avec notre objectif notamment les applications web, les types d'application web et les langages de Programmation Web spécifiquement PHP, JSP, ASP. Dans le prochain chapitre intitulé «Java EE », on va présenter une explication détaillée pour Java EE.

Chapitre III :

Java EE



Java EE™

1. Introduction

Dans le cadre de ce chapitre, nous allons présenter comment développer une application Web en utilisant Java Enterprise Edition (JAVA EE). Afin de mieux comprendre leur fonctionnement nous identifions les différents composants utilisés lors de leur développement notamment le langage JSP et Servlet. et nous verrons aussi comment ses applications Web sont liées les uns aux autres et comment ils interagissent entre eux.

2. Historique

L'environnement Java pour l'entreprise a commencé à émerger assez rapidement après le début de Java au milieu des années 90. À son origine, Java était destiné aux environnements contraints (par exemple des petits équipements électroniques). Il a en fait percé dans l'environnement du Web, notamment dans les navigateurs pour le support d'interfaces graphiques riches (notion d'appliquette Java, en anglais *applet*). Les premières déclinaisons de Java dans l'environnement des serveurs sont apparues en 1997 avec les *servlets*, dont l'objectif est la construction programmatique de pages Web, puis avec les *Enterprise Java Beans* dont l'objectif est le support de code métier nécessitant un contexte d'exécution transactionnel.

Java est désormais bien installée dans l'écosystème des applications d'entreprise et est devenu le principal concurrent de l'environnement *.NET* de Microsoft dont.

Le langage lui-même a acquis de la pérennité puisqu'on compte désormais plus de 4 millions de développeurs Java dans le monde.[8]

3. Java EE

La plateforme java entreprise (Java EE) est un ensemble de spécifications coordonné et pratique qui permettent des solutions pour le développement, le déploiement, et de la gestion des applications multi-tiers centralisées sur un serveur. Construite sur la plateforme de java 2 éditions standard (Java SE), la plateforme java EE ajoute les possibilités nécessaires pour fournir une plateforme complète, stable, sécurisée, et rapide de Java au niveau entreprise. [9]

4. Développement des Plateformes Java EE :

La figure 6 présente le développement de la plateforme Java EE :

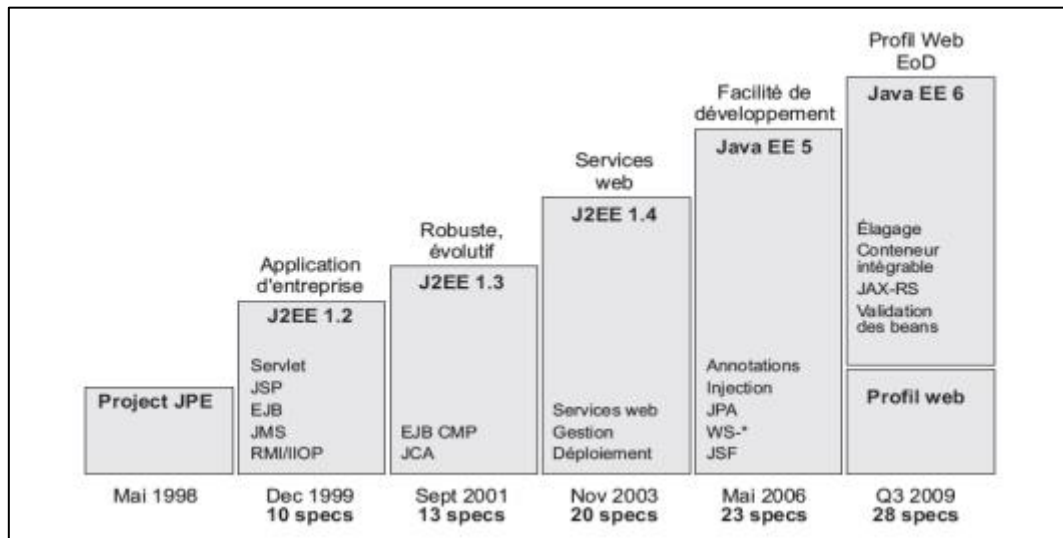


Figure 6 : Développement de Java EE.

La plate-forme Java EE est passée par plusieurs étapes nous citons :

La première version standard J2EE 1.2 a été développée par Sun en 1999. Les Entreprises JavaBeans (EJB) permettaient de gérer la couche métier d'accès aux données alors que les Servlets et Java Server Pages (JSP) permettaient de développer les applications serveur.

La version J2EE 1.3 (2001) n'apportait pas de nouveautés essentielles mais des améliorations de la plate-forme à la suite des demandes utilisateur, comme la configuration par fichier XML.

La version J2EE 1.4 (2003) Cette version apportait le support des services Web et des EJB version 2.1. Mais cette version conséquente était difficile à utiliser, tester et déployer.

Les développements par couches, de type Modèle Vue Contrôleur (MVC), ne facilitent pas la compréhension de l'ensemble, et les technologies Servlet et JSP ne sont pas adaptées pour un développement rapide. En réponse à cela, quelques Framework ou outils spécialisés on vu le jour, comme Struts et Spring pour la partie développement serveur, et Hibernate ou TopLink pour la persistance.

La version Java EE 5 (2006) apportait une réponse à cela en proposant un outil de développement côté serveur adapté au design pattern MVC, nommé Java Server Faces

Cette nouvelle version était une étape majeure pour la plate-forme Java EE qui s'orientait vers la simplicité, en proposant les EJB 3.

La version Java EE 6 (2009) se concentre sur la facilité d'utilisation de la plate-forme pour le développement, les tests et le déploiement. Cette version propose une simplification des EJB avec la version 3.1.

Java EE est devenu un standard utilisé par de nombreuses sociétés à travers le monde professionnel et celui de l'Open Source. Plusieurs outils peuvent être associés à Java EE comme les serveurs d'applications (Glass Fish, Web sphere, Web logic, JBoss...), des Framework de développement (Struts, Spring...), des outils de persistance. Ces mécanismes font en sorte qu'un programme puisse se terminer sans que ses données et son état d'exécution ne soient perdus. Ces informations de reprise peuvent être sauvegardées sur disque, éventuellement sur un serveur distant (un serveur de bases de données relationnelles, (Eclipse Link, Hibernate...)).

JEE est une plate-forme fortement orientée serveur, autrement dit, une application Java EE s'exécute sur un serveur d'applications. Un serveur d'applications est constitué d'un serveur HTTP et d'un conteneur web (pour exécuter les servlets et les JSP). [9]

5. Structure d'une application Java EE

Toute application web Java EE doit respecter une structure des dossiers standards, qui est définie dans les spécifications de la plate-forme. Vous en trouverez le schéma à la figure 7 :

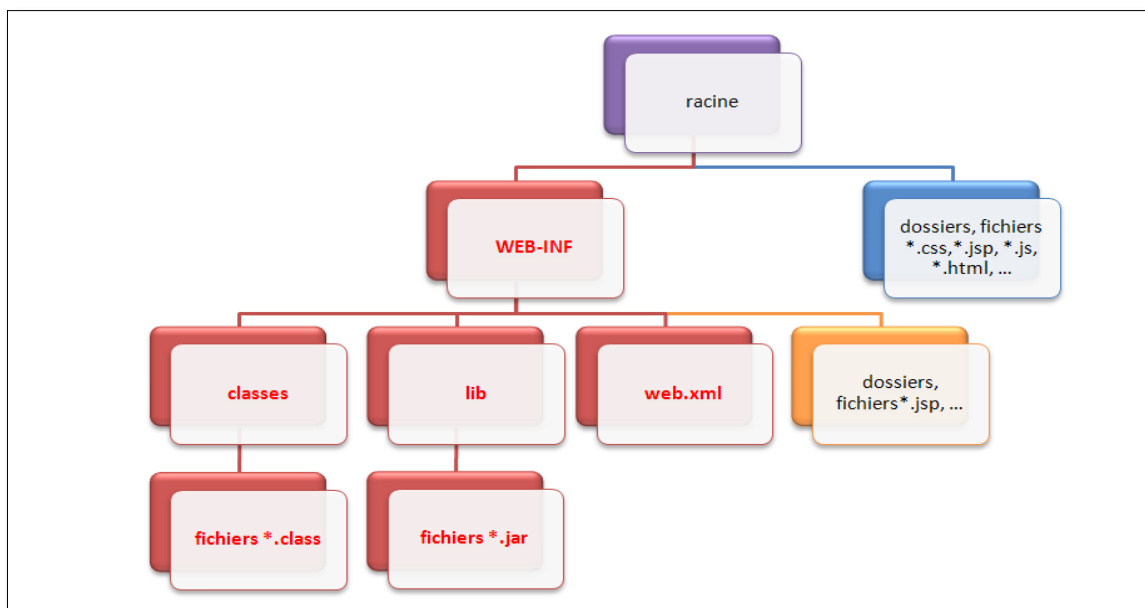


Figure 7: schéma Structure d'une application Java EE

Structure des fichiers d'une application web JSP/Servlet Quelques précisions :

- La racine de l'application, en violet sur le schéma, est le dossier qui porte le nom de votre projet et qui contient l'intégralité des dossiers et fichiers de l'application.
- Le dossier nommé **WEB-INF** est un dossier spécial. Il doit obligatoirement exister et être placé juste sous la racine de l'application. Il doit à son tour obligatoirement contenir :
 - le fichier de configuration de l'application (web.xml).
 - un dossier nommé **classes**, qui contient à son tour les classes compilées (fichiers .class).
 - un dossier nommé **lib**, qui contient à son tour les bibliothèques nécessaires au projet (archives .jar).

Bref, tous les dossiers et fichiers marqués en rouge sur le schéma doivent obligatoirement être nommés et placés comme indiqués sur le schéma.

- Les fichiers et dossiers persos placés directement sous la racine, en bleu sur le schéma, sont publics et donc accessibles directement par le client via leurs URL. (*)
- Les fichiers et dossiers persos placés sous le répertoire **WEB-INF**, en orange sur le schéma, sont privés et ne sont donc pas accessibles directement par le client. (*)

(*) Nous reviendrons en temps voulu sur le caractère privé du dossier **WEB-INF**, et sur la distinction avec les dossiers publics.[10]

6. Java

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts.[11]

7. Java Server Pages (JSP)

On a déjà défini JSP dans le deuxième chapitre application web dans la section « Les langages Web dynamique »

7.1. Avantages

- Les JSP reposent sur un le langage Java (un langage de programmation puissant, portable, sécurisé...).
- Si la page JSP est modifiée, nous n'avons pas besoin de recompiler et redéployer le projet.
- Une très bonne intégration du code Java et du HTML.

7.2. Inconvénients

- La technologie JSP demandera un serveur plus puissant avec plus de mémoire pour avoir des temps de réponse rapides. [6]

7.3. Cycle de vie d'une JSP

Un cycle de vie d'une JSP peut être défini comme l'ensemble du processus depuis sa création jusqu'à la destruction qui est similaire à un cycle de vie d'une servlet avec une étape supplémentaire qui est nécessaire pour compiler une JSP en servlet.

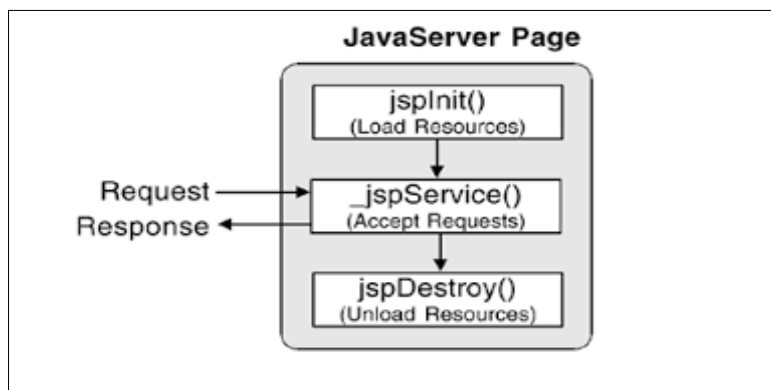


Figure 8 : Cycle de vie d'une JSP [12]

Une Page Jsp passe par les étapes suivantes :

- **Compilation** : Lorsqu'un navigateur demande une page JSP, le moteur JSP vérifie d'abord pour voir s'il a besoin de compiler la page. Si la page n'a jamais été compilée,

ou si la JSP a été modifié depuis la dernière compilation, le moteur JSP compile la page.

- **Initialisation** : Quand un conteneur charge une JSP, il invoque la méthode `jspInit ()` avant d'intervenir sur toutes les demandes.

```
1 public void jspInit(){
2
3 }
```

Listing 1 : la méthode `jspInit`

- **Exécution** : Chaque fois qu'un navigateur demande une JSP et la page a été chargée et initialisée, le moteur JSP appelle la méthode `_jspService ()` dans la JSP, la méthode `_jspService ()` est invoquée une fois par une demande et responsable de la génération de la réponse à cette demande et elle est également responsable de générer des réponses à toutes les autres méthodes HTTP.

```
1 void _jspService(HttpServletRequest request,
2                   HttpServletResponse response)
3 {
4     // code...
5 }
```

Listing 2 : la méthode `jspService`

- **Nettoyage** : La méthode `jspDestroy ()` est l'équivalente de la méthode `destroy()` pour les servlets. Charger `jspDestroy` lorsque vous devez effectuer un nettoyage, comme la libération des connexions de base de données ou la fermeture des fichiers ouverts.[12]

```
1 public void jspDestroy()
2 {
3     // code ...
4 }
```

Listing 3 : La méthode `jspDestroy`

8. SERVLET

8.1. Servlet

Une servlet est un programme qui s'exécute côté serveur en tant qu'extension du serveur. Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat. La liaison entre la servlet et le client peut être directe ou passer par un intermédiaire comme par exemple un serveur http.

Une servlet peut être invoquée plusieurs fois en même temps pour répondre à plusieurs requêtes simultanées.[13]

8.2. Cycle de vie d'une servlet

- init() : initialisation de la servlet chargement du code.

Souvent effectué lors de la première requête cliente (doGet, doPost) Allocation d'un pool de threads

- doGet () : Traitement des requêtes HTTP GET

- doPut () :Traitement des requêtes HTTP PUT

- doPost () :Traitement des requêtes HTTP POST

- destroy () : destruction de la servlet par le serveur.[14]

8.3. Comparaison Servlet/JSP

- ☞ JSP : compilé en servlet

- ☞ Servlet : possibilité de distinguer les requêtes HTTP (doPut, doGet, doPost, ...)

- ☞ JSP : bcp HTML, peu Java

- ☞ Servlet : bcp Java, peu HTML

- ☞ servlet : pur Java : facilement éditable dans IDE

- ☞ JSP : plutôt éditeur de pages HTML

- ☞ servlet compilation avant déploiement / JSP après JSP à redéployer si erreur.[15]

9. Conclusion

Dans ce chapitre on a présenté JAVA EE et le langage de Programmation web JSP, Nous avons donné des définitions sur java ee, et Jsp et servlet, et aussi parlé du développement des plateformes java EE et les structure d'une application java EE. Nous allons voir dans le chapitre suivant le Web sémantique.

Chapitre IV :



1. Introduction

Dans ce chapitre nous allons décrire le Web sémantique et sa pile de langage. Parmi ces langages on donne un aperçu sur RDF, RDFS, XML et un détail sur le langage OWL puisqu'il constitue le noyau de notre application Web. À travers ce langage on définit l'ontologie de notre application.

2. Web Sémantique

Le web sémantique est une extension du web qui facilite l'automatisation du traitement des connaissances disponibles.

C'est une extension du web classique (HTML, HTTP, etc. ne sont pas remis en cause). Les connaissances ne sont pas représentées dans une langue naturelle mais formalisées à l'aide de langages pouvant être interprétés par des machines.

Agrégation de connaissances venant de plusieurs sources, comparaisons publication de données sémantiques sous différentes formes génération de nouvelles connaissances (inférence) etc.

Pas uniquement utile pour le web-internet.

L'initiative web sémantique est soutenue par le W3C1.[16]

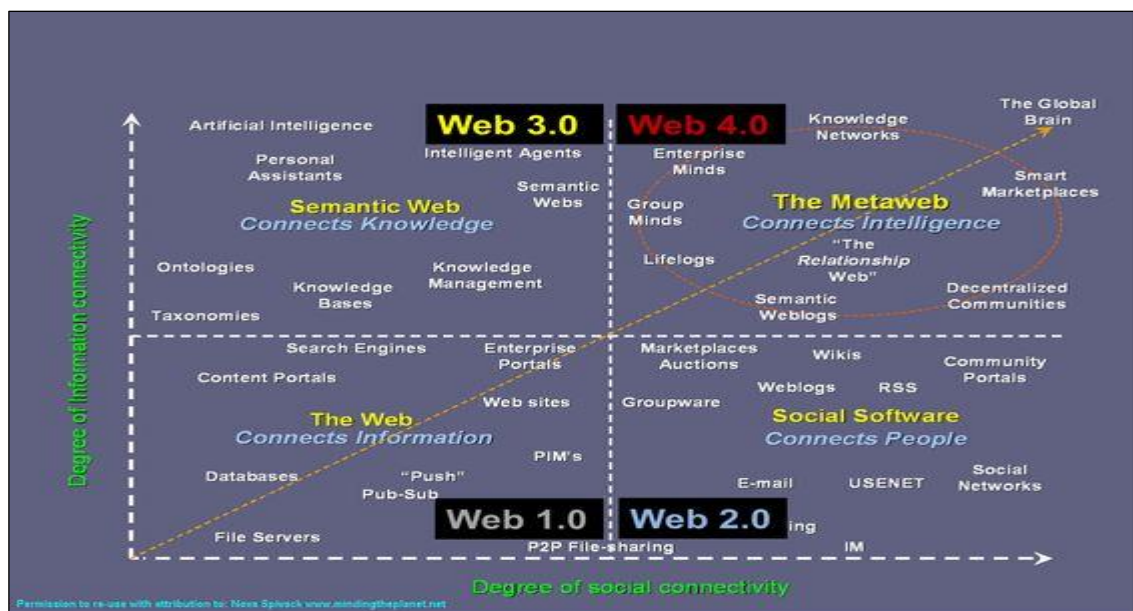


Figure 9 : Le web du futur[17]

3. Objectif du web sémantique

Un des principaux objectifs du Web sémantique est de permettre aux utilisateurs d'utiliser la totalité du potentiel du Web : ainsi, ils pourront trouver, partager et combiner des informations plus facilement. Aujourd'hui tout le monde est capable d'utiliser des forums, d'utiliser des réseaux sociaux, de chatter, de faire des recherches ou même d'acheter différents produits. Néanmoins, il serait mieux que la machine fasse tout ceci à la place de l'homme, car actuellement, les machines ont besoin de l'homme pour effectuer ces tâches. La raison principale est que les pages Web actuelles sont conçues pour être lisibles par des êtres humains et non par des machines. Le Web sémantique a donc comme principal objectif que ces mêmes machines puissent réaliser seules toutes les tâches fastidieuses comme la recherche ou l'association d'informations et d'agir sur le Web lui-même.[18]

4. Architecture du Web sémantique

La vision courante du Web sémantique proposée par Tim Berners-Lee-2001 peut être représentée dans une architecture en plusieurs couches différentes (Figure 10).

Les couches les plus bases assurent l'interopérabilité syntaxique : la notion d'URI fournit un adressage standard universel permettant d'identifier les ressources tandis que Unicode est un encodage textuel universel pour échanger des symboles. Rappelons que l'URL comme l'URI, est une chaîne courte de caractères qui est aussi utilisée pour identifier des ressources (physiques) par leur localisation.

XML fournit une syntaxe pour décrire la structure du document, créer et manipuler des instances des documents. Il utilise l'espace de nommage (namespace) afin d'identifier les noms des balises (tags) utilisées dans les documents XML. Le schéma XML permet de définir les vocabulaires pour des documents XML valides. Cependant, XML n'impose aucune contrainte sémantique à la signification de ces documents, l'interopérabilité syntaxique n'est pas suffisante pour qu'un logiciel puisse "comprendre" le contenu des données et les manipuler d'une manière significative.

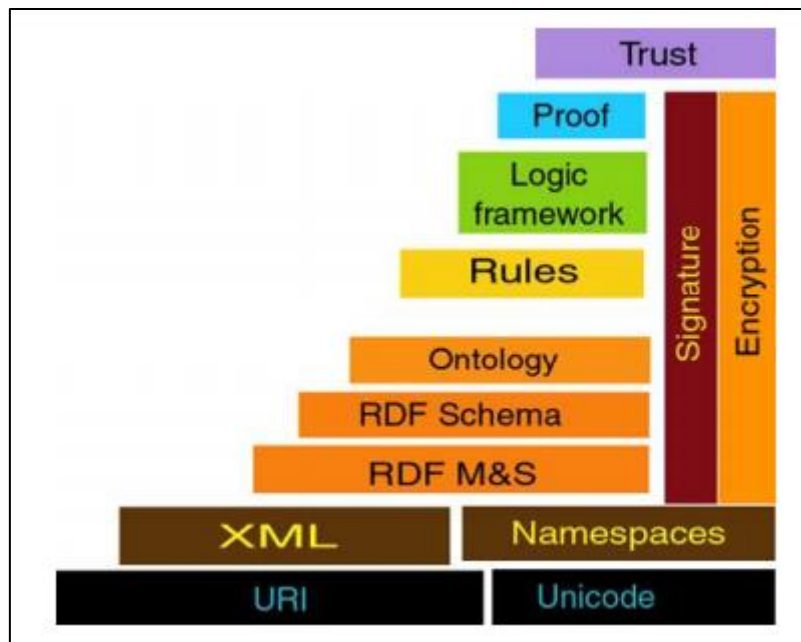


Figure 10 : Architecture du Web sémantique

Les couches RDF M&S (RDF Model and Syntax) et RDF Schéma sont considérées comme les premières fondations de l'interopérabilité sémantique. Elles permettent de décrire les taxonomies des concepts et des propriétés (avec leurs signatures). RDF fournit un moyen d'insérer de la sémantique dans un document, l'information est conservée principalement sous forme de déclarations RDF. Le schéma RDFS décrit les hiérarchies des concepts et des relations entre les concepts, les propriétés et les restrictions domaine/co-domaine pour les propriétés. Les trois langages XML, RDF et RDFS seront présentés plus en détail dans la partie suivante.

La couche suivante ontologie décrit des sources d'information hétérogènes, distribuées et semi-structurées en définissant le consensus du domaine commun et partagé par plusieurs personnes et communautés. Les ontologies aident la machine et l'humain à communiquer avec concision en utilisant l'échange de sémantique plutôt que de syntaxe seulement.

Les règles sont aussi un élément clé de la vision du Web sémantique, la couche Règles offre la possibilité et les moyens de l'intégration, de la dérivation, et de la transformation de données provenant de sources multiples, etc.

La couche Logique se trouve au-dessus de la couche ontologie. Certains considèrent ces deux couches comme étant au même niveau, comme des ontologies basées sur la logique et

permettant des axiomes logiques. En appliquant la déduction logique, on peut inférer de nouvelles connaissances à partir d'une information explicitement représentée.

Les couches Preuve (Proof) et Confiance (Trust) sont les couches restantes qui fournissent la capacité de vérification des déclarations effectuée dans le Web Sémantique. On s'oriente vers un environnement du Web sémantique fiable et sécurisé dans lequel nous pouvons effectuer des tâches complexes en sûreté. D'autre part, la provenance des connaissances, des données, des ontologies ou des déductions est authentifiée et assurée par des signatures numériques, dans le cas où la sécurité est importante ou le secret nécessaire, le chiffrement est utilisé. [19]

5. Langages du Web sémantique

Dans le contexte du Web sémantique, plusieurs langages ont été développés. La plupart de ces langages reposent sur XML ou utilisent XML comme syntaxe. Nous allons présenter brièvement certains langages principaux XML, XML Schéma, RDF(S), OWL, SPARQL. [19]

5.1. XML (eXtensible Markup Language)

XML est un langage de balisage extensible et est considéré comme une spécification pour les documents "lisibles par les machines". Il est naturellement utilisé pour encoder les langages du Web sémantique. Le balisage signifie que certaines suites des caractères du document peuvent contenir de l'information indiquant le rôle du contenu du document. XML se sert de balises (tags, par exemple ou) pour décrire un classement des données du document et sa structure logique. Cependant, le caractère extensible indique la différence importante avec d'autres langages précédents qui est aussi la caractéristique essentielle du XML. XML est un métalangage (une description de type de document, DTD, permet de décrire la grammaire des documents admissibles) : en effet XML fournit une structure pour représenter d'autres langages d'une manière normalisée.

Un document XML a toujours un élément racine. Des éléments XML peuvent contenir d'autres éléments qui sont représentés par les balises d'ouverture et de clôture. La figure 11 donne un extrait d'un document XML : [19]

```

1 <?xml version="1.0"?>
2 <etudiant>
3   <nom>Luong Phuc Hiep</nom>
4   <phone>5096</phone>
5   <bureau>
6     <numero>BS04</numero>
7     <batiment>Borel</batiment>
8   </bureau>
9 </etudiant>

```

Figure 11 : extrait d'un document XML.

5.2. RDF (Resource Description Framework)

Un document RDF est un ensemble de triplets de la forme <ressource, propriété, valeur> une ressource est une entité accessible via un URI¹¹ sur l'Internet (par exemple un document HTML ou XML, une image, une page Web ou même une partie de la page Web). Une propriété définit une relation binaire entre une ressource et une valeur (associer de l'information à une ressource). Une valeur est une ressource ou une valeur littérale (chaîne de caractères). Un énoncé (ou une déclaration) RDF spécifie la valeur d'une propriété d'une ressource. On peut le décrire comme propriété (ressource, valeur). Les éléments de ces triplets peuvent être des URI, des littéraux ou des variables. Cet ensemble de triplets peut être représenté de façon naturelle par un graphe (plus précisément un multi-graphe orienté étiqueté), où les éléments apparaissant comme ressource ou valeur sont les sommets, et chaque triplet est représenté par un arc dont l'origine est sa ressource et la destination sa valeur comme la montre la figure 12 : [19]

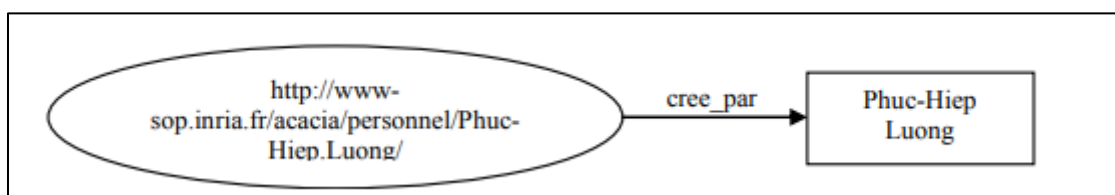


Figure 12 : Modèle de triplet

En utilisant ce modèle de triplet, le fait “la page Web à l'adresse <http://www-sop.inria.fr/acacia/personnel/Phuc-Hiep.Luong/> est créée par PhucHiep Luong” est présenté en RDF dans la figure 13 :

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:dc="http://purl.org/dc/elements/1.1/">
4   <rdf:Description rdf:about="http://www.sop.inria.fr/acacia/personnel/Phuc-Hiep.Luong">
5     <creer_par>
6       <rdf:Description>
7         <dc:nom>Luong Phuc Hiep</dc:nom>
8       </rdf:Description>
9     </creer_par>
10  </rdf:Description>
11 </rdf:RDF>
```

Figure 13 : Modèle de triplet en RDF

5.3. RDFS (Resource Description Framework Schema)

Le langage RDFS a été développé en se basant sur RDF. Le modèle des données RDF ne précise que le mode de description des données mais ne fournit pas la déclaration des propriétés spécifiques au domaine ni la manière de définir ses propriétés avec d'autres ressources. RDFS a pour but d'étendre RDF en décrivant plus précisément les ressources utilisées pour étiqueter les graphes. Pour cela, il fournit un mécanisme permettant de spécifier les classes dont les instances seront des ressources, comme les propriétés.

RDFS s'écrit toujours à l'aide de triplets RDF en utilisant deux propriétés fondamentales `subClassOf` et `type` pour représenter respectivement les relations de subsumption entre classes et les relations d'instanciation entre instances et classes. Les classes spécifiques au domaine sont déclarées comme des instances de la ressource `Class` et les propriétés spécifiques au domaine comme des instances de la ressource `Property`. Les propriétés `subClassOf` et `subPropertyOf` permettent de définir des hiérarchies de classes et de propriétés. D'autre part, RDFS ajoute à RDF la possibilité de définir les contraintes de domaine et co-domaine de valeurs avec l'aide des attributs `rdfs:domain` et `rdfs:range`. Les ressources instances sont décrites en utilisant le vocabulaire donné par les classes définies dans ce schéma. Pour résumer, XML peut être vu comme la couche de transport syntaxique, RDF comme un langage relationnel de base. RDFS offre des primitives de représentation de structures ou primitives ontologiques la figure 14 présenté un exemple RDF[19]

```

1 <rdf:RDF>
2 ...
3 <rdfs:Class rdf:ID="Personne">
4   <rdfs:subClassOf rdf:resource="#Entité"/>
5 </rdfs:Class>
6 <rdfs:Class rdf:ID="Etudiant">
7   <rdfs:subClassOf rdf:resource="#Personne"/>
8 </rdfs:Class>
9 ...
10 <rdf:Property rdf:ID="travailler_dans">
11   <rdfs:domain rdf:resource="#Personne"/>
12   <rdfs:range rdf:resource="#Institut"/>
13 </rdf:Property>
14 ...
15 </rdf:RDF>

```

Figure 14 : Exemple RDFs

5.4. OWL (Web Ontology Language)

OWL est un langage pour représenter des ontologies dans le Web sémantique. C'est une extension du vocabulaire de RDF(S). Le langage OWL offre aux machines de plus grandes capacités d'interprétation du contenu Web que celles permises par XML, RDF et RDF schéma (RDFS), grâce à un vocabulaire supplémentaire et une sémantique formelle. Inspiré des logiques de description (et successeur de DAML+OIL), OWL fournit un grand nombre de constructeurs permettant d'exprimer de façon très fine les classes de manière plus complexe correspondant aux connecteurs de la logique de description équivalente (intersection, union, restrictions diverses, etc.), les propriétés des classes définies (telles que la disjonction), la cardinalité (par exemple "exactement un"), plus des types des propriétés (propriétés d'objet ou d'annotation...), des caractéristiques des propriétés (par exemple la symétrie, la transitivité), et des classes énumérées.

Le langage OWL se compose de trois sous-langages offrant une expressivité croissante : OWL-Lite, OWL-DL et OWL-Full.[19]

5.4.1. OWL-Lite le plus simple, il est destiné à représenter des hiérarchies de concepts simples, sa simplicité lui permet d'avoir une complexité faible, et le calcul de réponse et en temps raisonnable.

5.4.2. OWL-DL plus complexe fondé sur la logique descriptive, Il est adapté pour faire des raisonnements.

5.4.3. OWL-Full plus complexe du OWL utilisé pour avoir un haut niveau de capacité de description, quitte à ne pas pouvoir garantir la complétude et la décidabilité des calculs liés à l'ontologie. [20]

5.5. SPARQL

SPARQL est un langage de requêtes RDF, proche de SQL. Une requête SPARQL simple est basée sur les modes de recherches. Le traitement des requêtes se compose de liaisons de variables pour générer des modèles de solutions (filtrage). SPARQL est toujours en cours d'amélioration. Il est défini par le groupe de travail Data Access (DAWG) du W3C, et devient, ainsi, les langages de requêtes les plus populaires pour les graphes sémantiques. C'est pour l'ensemble de ces raisons que nous avons choisi de construire notre travail sur le langage SPARQL. [21]

6. Ontologie

Le mot « ontologie » vient du grec : ontos pour être, Logie pour l'étude et logos pour univers. C'est un terme philosophique introduit au XIXème siècle qui caractérise l'étude des êtres dans son univers. En informatique, plusieurs définitions ont été données à l'ontologie parmi eux de « Une ontologie est une spécification explicite d'une conceptualisation » Ca veut dire une ontologie est une description formelle d'une vue abstraite et simplifiée du monde que l'on veut représenter. [20]

7. Les composants d'une ontologie

Pour décrire un domaine avec les ontologies on représente les connaissances de ce domaine par les cinq composants suivant :

7.1. Les concepts

Sont des notions (ou objets) permettant la description d'une tâche, d'une fonction, d'une action, d'une stratégie ou d'un processus de raisonnement, etc. Ils peuvent être abstraits ou concrets, élémentaires ou composés, réels ou fictifs. Habituellement, les concepts sont organisés en taxonomie. Une taxonomie est une hiérarchie de concepts (ou d'objets) reliés entre eux en fonction de critères sémantiques particulière.

7.2. Les relations

Sont les liens organisant les concepts de façon à représenter un type d'interaction entre les concepts d'un domaine. Elles sont formellement définies comme tout sous-ensemble d'un produit de n ensemble, c'est-à-dire $R: C_1 \times C_2 \times \dots \times C_n$. Des exemples de relations binaires sont : sous-concept-de, connecté-à, sorte-de, etc.

7.3. Les propriétés

Ou attributs, sont des restrictions des concepts ou des relations.

7.4. Les fonctions

Sont des cas particuliers de relations dans lesquelles le nième élément de la relation est unique pour les n-1 précédents. Formellement, les fonctions sont définies ainsi, $F: C_1 \times C_2 \times \dots \times C_{n-1}, C_n$. Comme exemple de fonction binaire, nous avons la fonction mère-de

7.5. Les axiomes

De l'ontologie permettent de définir la sémantique des termes (classes, relations), leurs propriétés et toutes contraintes quant à leur interprétation. Ils sont définis à l'aide de formules bien formées de la logique du premier ordre en utilisant les prédicats de l'ontologie. - Les instances sont utilisées pour représenter des éléments. [22]

8. Les types d'ontologies

Deux grandes classes d'ontologies sont à distinguer : la première est liée au type de la structure de la conceptualisation ; la deuxième au domaine à conceptualiser. Concernant la première classe, l'ontologie se présente de façon différente selon le degré de formalisation du langage utilisé pour définir la signification des termes. Une ontologie est dite hautement informelle dans le cas d'utilisation du langage naturel sans aucune restriction, semi-informelle lorsque le langage est de type langage naturel mais structuré avec un vocabulaire limité afin de restreindre les ambiguïtés. Si l'ontologie est représentée à l'aide d'un langage artificiel défini de façon formelle tel qu'Ontolingua, l'ontologie est alors semi formelle. Enfin, lorsque les termes possèdent une sémantique formelle dans un système tel que le calcul des prédicats du premier ordre comme c'est le cas de l'ontologie TOVE l'ontologie est dite rigoureusement formelle. Dans la deuxième classe, l'ontologie se définit selon le domaine étudié et le degré de généralité ou de précision des connaissances représentées, les types d'ontologie les plus utilisées sont :

8.1. Les ontologies générales

Les ontologies générales portent sur des concepts généraux qui se veulent indépendants d'un domaine ou d'un problème particulier, tels que les concepts de temps, d'espace, de notions mathématiques, « elles sont prévues pour être utilisées dans des situations diverses et variées et peuvent servir une large communauté d'utilisateurs, ainsi elles peuvent être utilisées dans l'organisation des parties substantielles des connaissances humaines, comme la

compréhension du langage naturel. ».Plusieurs ontologies ont été développées pour décrire des concepts généraux ou des domaines particuliers. Parmi les ontologies générales il y a i) Cyc développée avec le modèle logique, en utilisant le langage CycL. Cette ontologie a la possibilité de construire des applications pour l'extraction des connaissances, la recherche intelligente et la traduction, etc. ; et ii) KR Ontology qui utilise le modèle de treillis et le FCA (Formal Concept Analysis) pour représenter l'ontologie

8.2. Les ontologies du domaine

Ce type d'ontologies exprime des conceptualisations spécifiques à des ontologies de domaines particuliers tout en étant générique pour ce domaine. Ces conceptualisations mettent des contraintes sur la structure et les contenus des connaissances du domaine. De nombreuses ontologies de domaine ont été développées notamment dans le domaine de la modélisation de l'entreprise : Enterprise Ontology, Tove, et dans le domaine médical UMLS.

8.3. Les ontologies d'application

Ces ontologies sont les plus spécifiques. Les concepts correspondent souvent aux rôles joués par les entités du domaine tout en exécutant une certaine activité. Elles peuvent contenir des extensions spécifiques telles les méthodes et tâches. Elles contiennent toutes les définitions nécessaires pour décrire la connaissance requise pour une application particulière.[22]

9. Exemple de manipulation ontologie OWL :

La figure 15 montré Un exemple simple d'une ontologie :

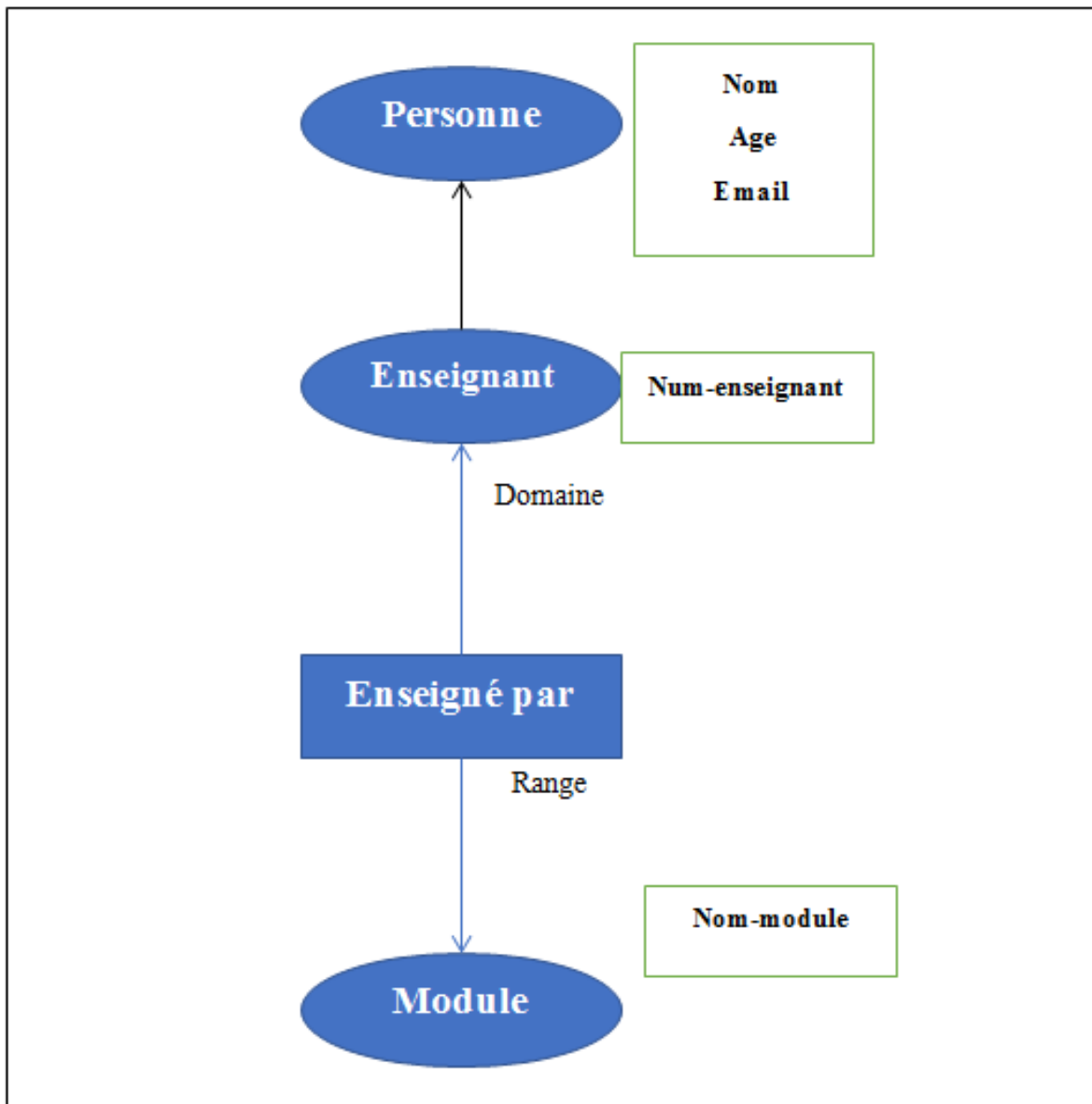


Figure 15 : Exemple Ontologie

9.1. Création avec Protégé

9.1.1. Créer la classe Personne

Pour la création des classes, réalisez les manipulations suivantes :

- ◆ Dans Protégé, activez l'onglet Classes et cliquez sur le bouton de création d'une classe.
- ◆ Dans la boîte de dialogue « Create à new OWL Class », saisissez Personne puis cliquez sur OK.

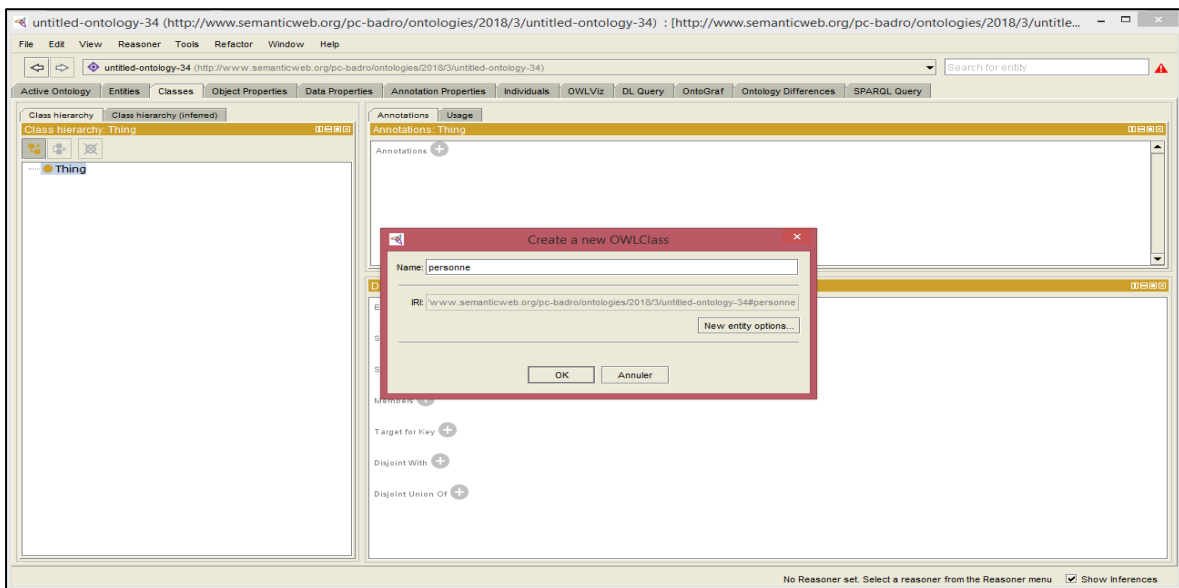


Figure 16: Créer la classe Personne

9.1.2. Créer 2^{ème} classe module :

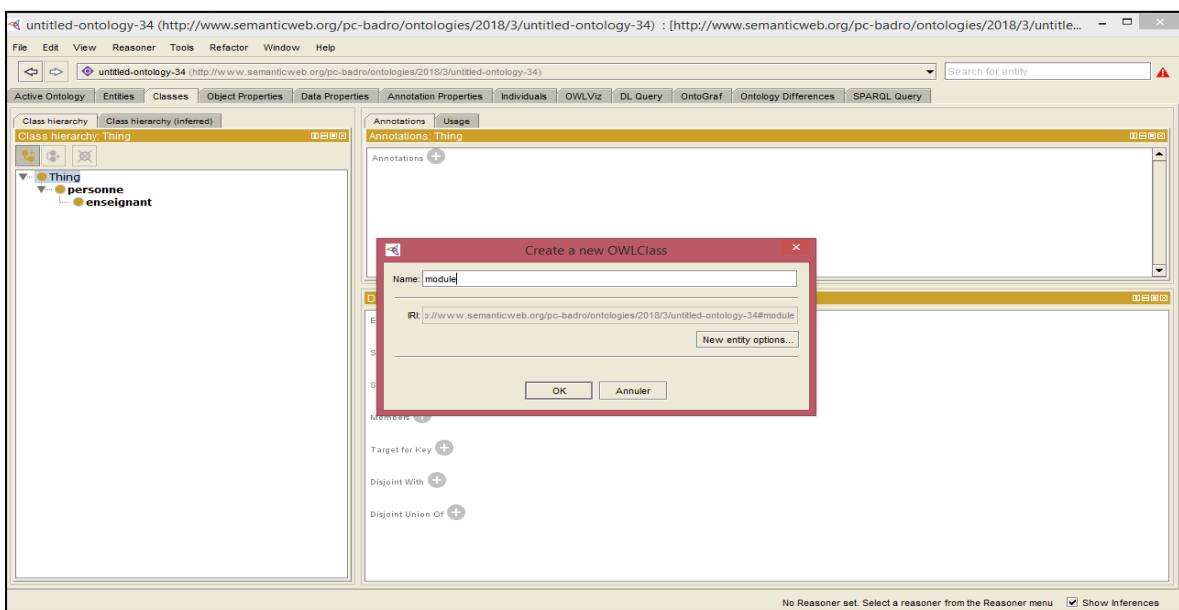


Figure 17 : Créer la classe module

9.1.3. Créer SubClassOf Enseignant

- ◆ Sélectionnez la casse **Personne** et cliquez sur le bouton de création d'une classe.
- ◆ Dans la boîte de dialogue « Create a new OWL Class », saisissez **Enseignant** puis cliquez sur **OK**.

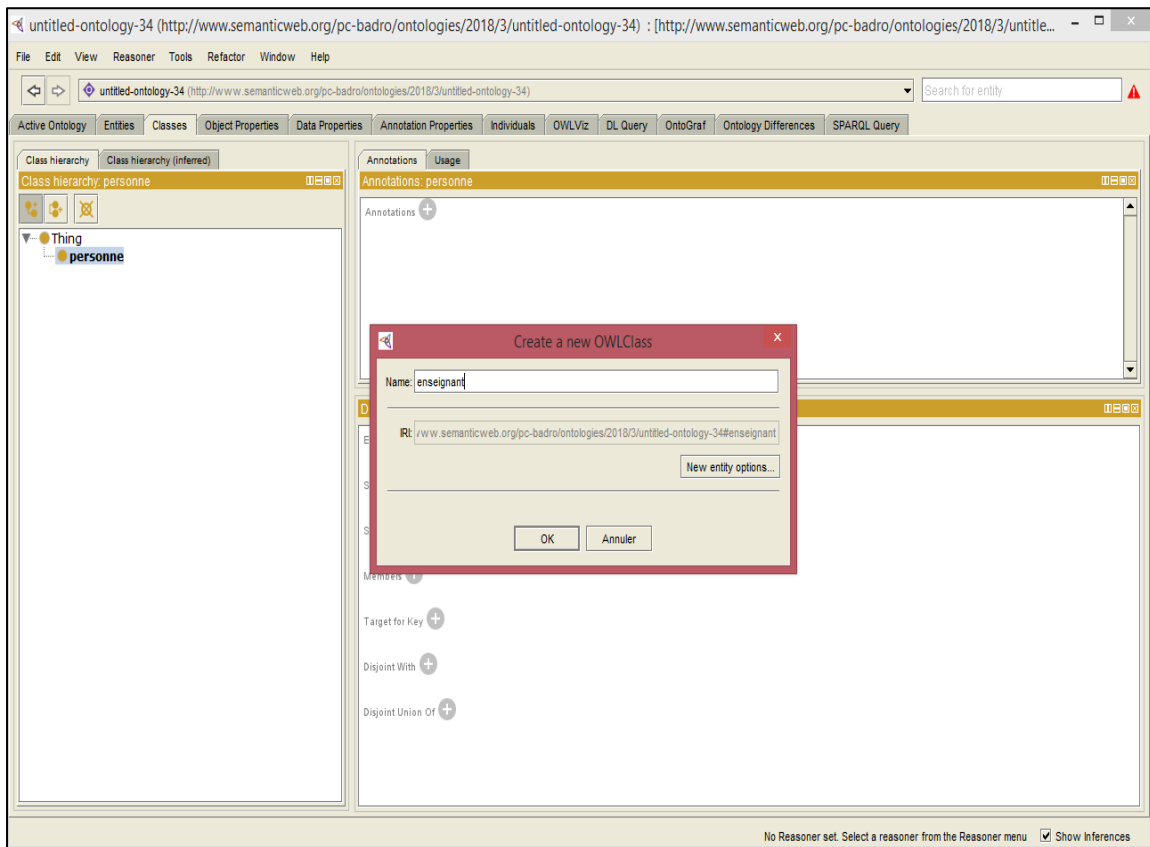


Figure 18 : Créer SubClassOf Enseignant

9.1.4. Créer les propriétés d'objets

Pour créer les propriétés d'objets, réalisez la manipulation suivante :

- ◆ De retour dans Protégé, cliquez sur l'onglet **Object Properties**.

Créer la propriété d'objets « **enseigné par** »

- ◆ Sélectionnez **topObjectProperty** et cliquez sur **Add sub property**.
- ◆ Dans le champ **Name**, entrez **enseigné par** puis cliquez sur **OK**.

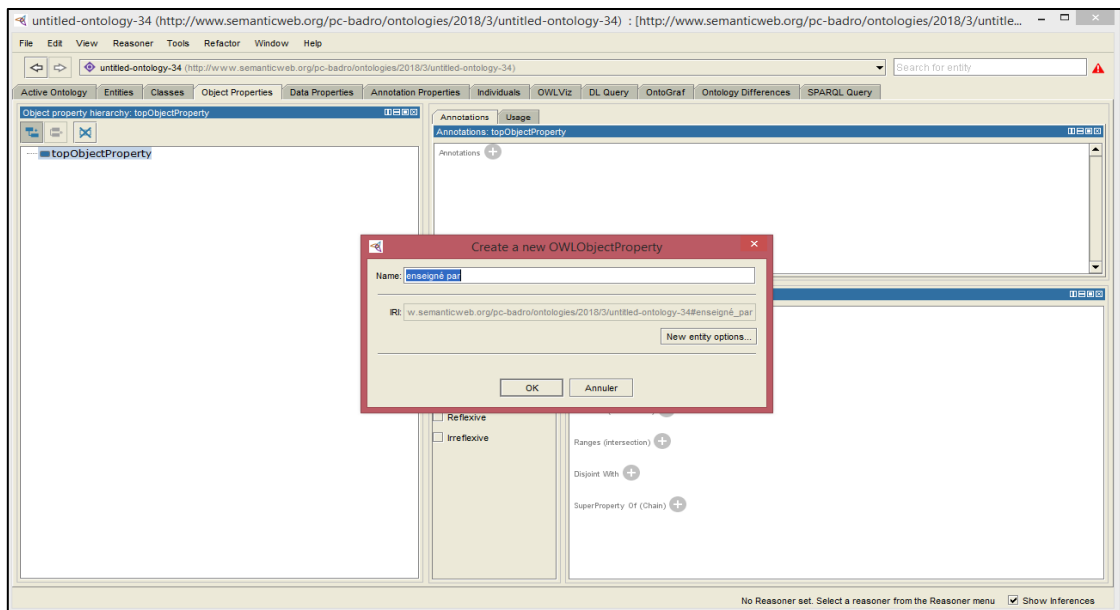


Figure 19 : Créer la propriété d'objets « enseigné par »

- ◆ À la description **Domain (intersection)**, cliquez sur le +, sélectionnez l'onglet **Class hierarchy**, puis sélectionnez la sub classe **enseignant** et cliquez sur **OK**

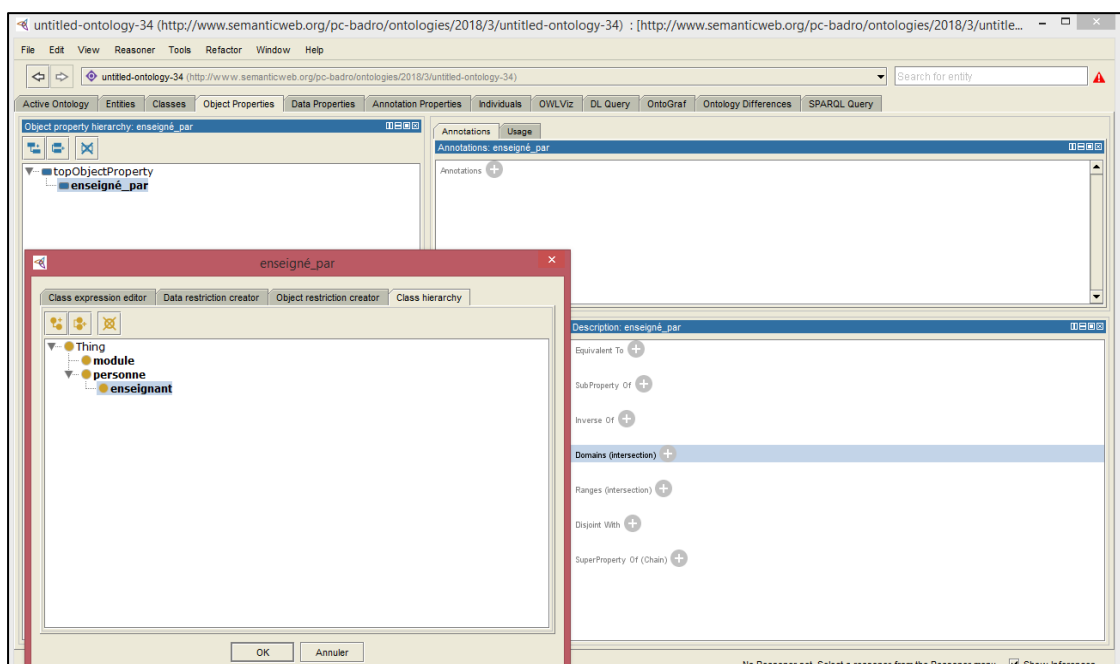


Figure 20 : Domain de Object Properties « enseigné par »

◆ À la description **Ranges (intersection)**, cliquez sur le +, sélectionnez

L'onglet **Class hierarchy**, puis sélectionnez la classe **module** et cliquez sur **OK**.

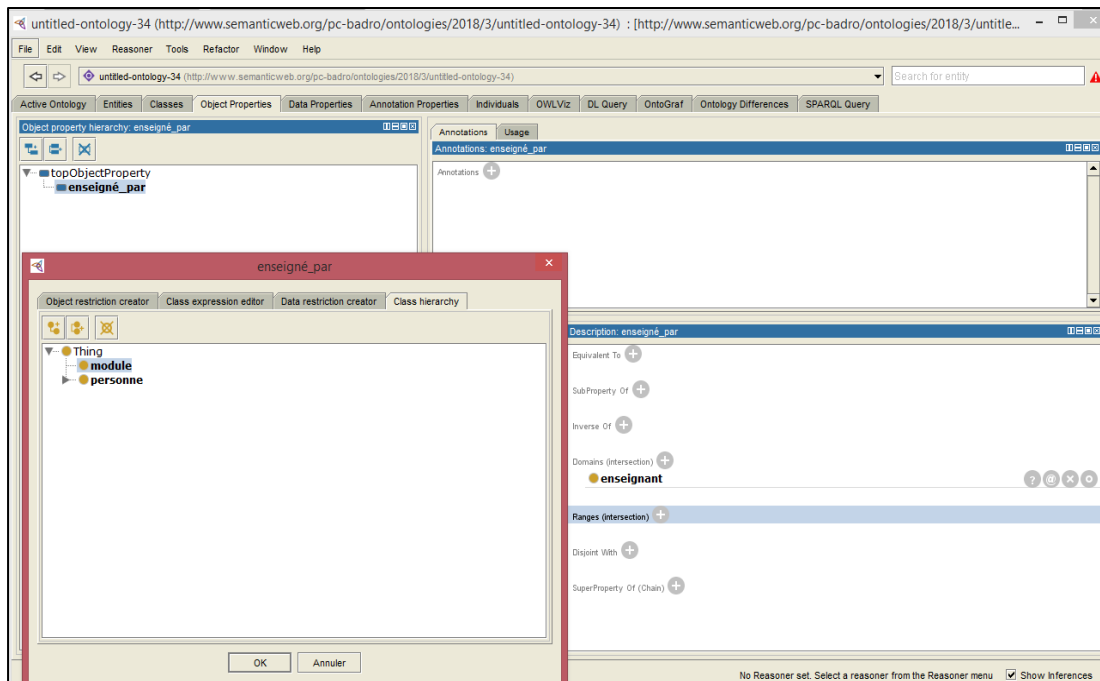


Figure 21 : Range de Object **Properties** enseigné par

9.1.5. Créer les propriétés de données

Pour créer les propriétés de données, réalisez la manipulation suivante :

◆ Sélectionnez l'onglet **Data Properties**.

Créer la propriété associée à un type de donnée **age** :

◆ Sélectionnez **topDataProperty**, puis cliquez sur le bouton **Add sub property**.

◆ Saisissez **age** dans le champ **Name**, puis cliquez sur **OK**.

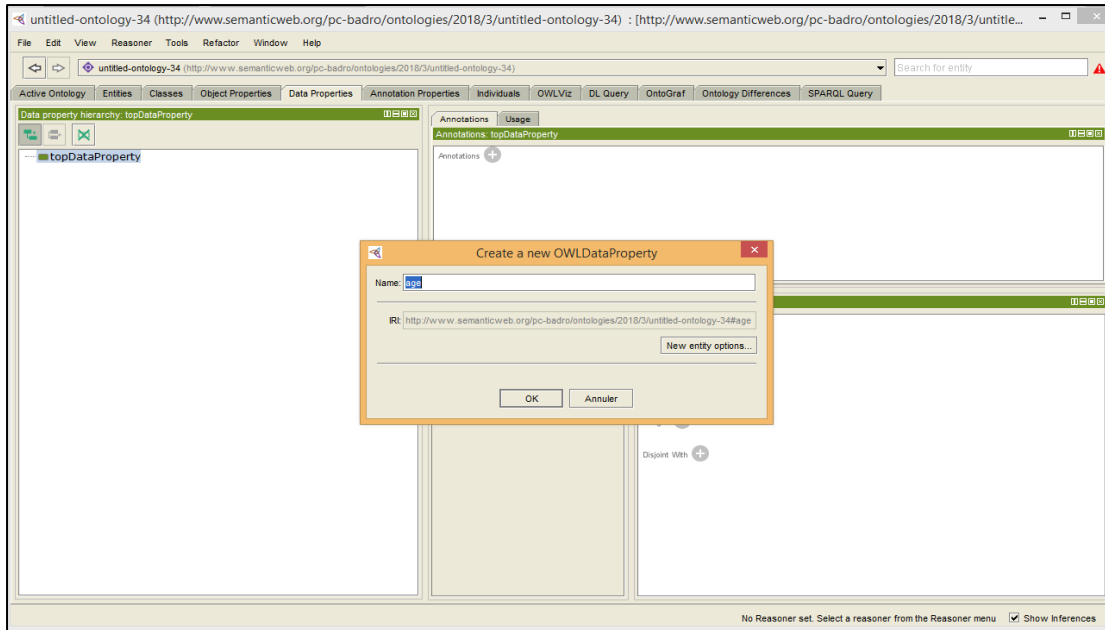


Figure 22 : créer Data Propriétés Age

◆ À la description **Domain (intersection)**, cliquez sur le +, sélectionnez

L'onglet **Class hierarchy**, puis sélectionnez la propriété de données et cliquez sur **OK**.

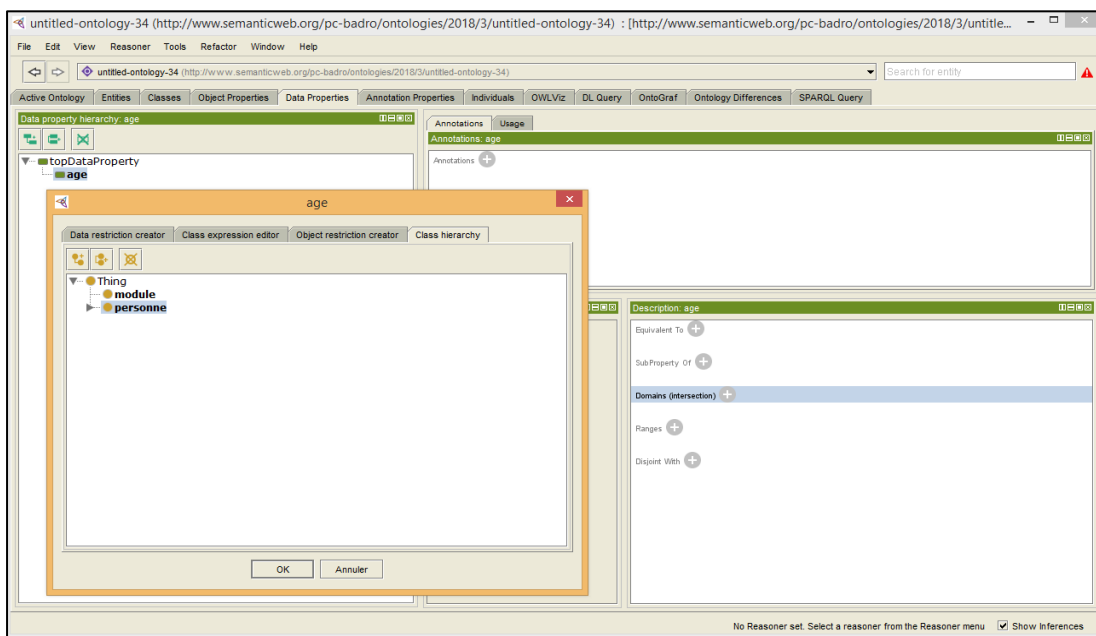


Figure 23 : Domain de Data Properties age

◆ Dans la section **Description**, cliquez sur le + de **Ranges**, dans l'onglet **Built in datatypes** sélectionnez **integer**, puis cliquez sur **OK**.

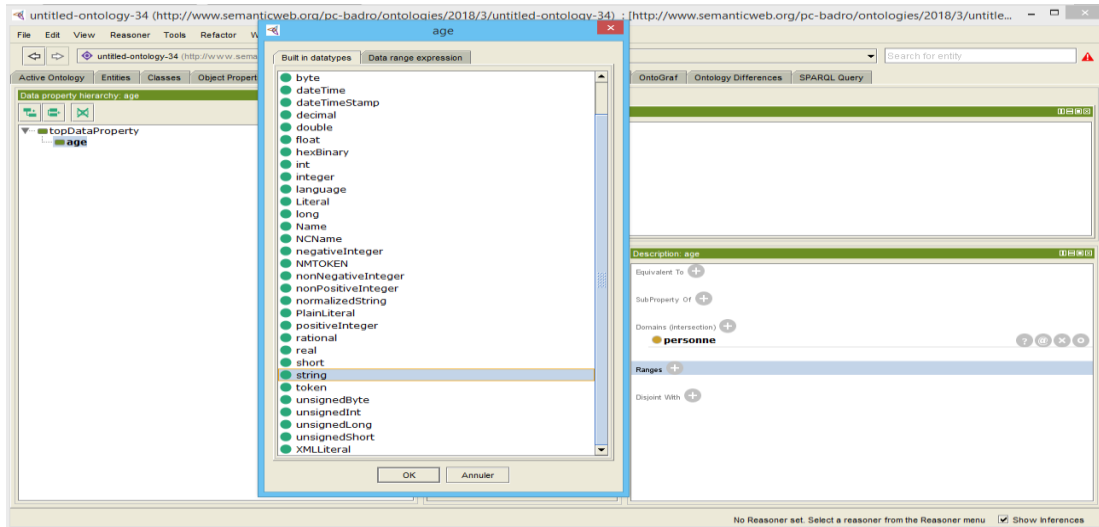


Figure 24 : Range de Data Properties age

9.2. Création d'ontologie avec java

Le listing 4 présent le code source de création ontologie avec java

```

1. OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM);
2. String uriBase = "http://www.annoncedz.com";
3. model.createOntology(uriBase);
4. model.setNsPrefix("",uriBase+"#");

```

Listing 4 :Création Ontologie

9.2.1. Création des classes

Le listing 5 montre le code source de création les classes personne et module et enseignant :

```

// create classes
1. OntClass personne = model.createClass(uriBase+"#Personne");
2. OntClass module = model.createClass(uriBase+"#Module");
3. OntClass enseignant = model.createClass(uriBase+"#Enseignant");

```

Listing 5 : Create classes

9.2.2. Création des sub-classes

Le listing 6 montre code source de création sub-classe enseignant

```

// set sub-classes
1. persomme.addSubClass(enseignant);

```

Listing 6 : crée sub classes

9.2.3. Création des object properties

Le listing 7 présent le code source de création Object properties Créer

```
// create object properties  
1. ObjectProperty enseigné_par = model.createObjectProperty(uriBase+"#Créer");  
2. enseigné_par.setDomain(enseignant);  
3. enseigné_par.setRange(module);
```

Listing 7 : crée object properties

9.2.4. Création datatype properties

Le listing 8 présent le code source de création datatype properties

```
1. DatatypeProperty Email = model.createDatatypeProperty(uriBase+"#Email");  
2. Email.setDomain(persomme);  
3. Email.setRange(XSD.xstring);  
4. DatatypeProperty Nom = model.createDatatypeProperty(uriBase+"#Nom");  
5. Nom.setDomain(persomme); Nom.setRange(XSD.xstring);  
6. DatatypeProperty Age = model.createDatatypeProperty(uriBase+"#Age");  
7. Age.setDomain(persomme);  
8. Age.setRange(XSD.xstring);  
9. DatatypeProperty Num_enseignant=model.createDatatypeProperty(uriBase+"#Num_enseignant");  
10. Num_enseignant.setDomain(enseignant);  
11. Num_enseignant.setRange(XSD.integer);  
12. DatatypeProperty Nom_module = model.createDatatypeProperty(uriBase+"#Nom-odule");  
13. Nom_module.setDomain(module);  
14. Nom_module.setRange(XSD.xstring);
```

Listing 8 : créer des propriétés de type de données

9.3. Les Requête SPARQL :

9.3.1. Requête insertion

Le listing 9 présent une requête insertion en sparql :

```
1. PREFIX dc:<http://www.semanticweb-exemple-owl.org#>
2. INSERT DATA {
3. dc:Personne01
4. dc:nom \"Omar\" ;
5. dc:age 20 ;"
6. dc:email \"Omar@gmail.com\".
7. } ;
```

Listing 9 : Ajouter les données

9.3.2. Requête suppression

Le listing 10 montre une requête de suppression en sparql :

```
1. PREFIX dc:<http://www.semanticweb-exemple-owl.org#>
2. DELETE DATA {
3. dc:Personne01
4. dc:nom \"Omar\" ;
5. dc:age 20 ;"
6. dc:email \"Omar@gmail.com\".
7. } ;
```

Listing 10 : Supprimer les données

9.3.3. Requête update

Le listing 11 présent une requête modifier en sparql :

```
1. PREFIX dc: < http://www.semanticweb-exemple-owl.org#> ".
2. DELETE { ?prs dc:nom \"Omar\" ;
3. dc:age 20 ;"
4. dc:email \"Omar@gmail.com\".
5. }
6. INSERT { ?prs dc:nom \"ali\" ;
7. dc:age 23 ;"
8. dc:email \" ali@gmail.com\".
9. }
10. WHERE { ?prs dc:nom \"Omar\" ;
11. dc:age 20 ;"
12. dc:email \"Omar@gmail.com\".
13. } ;
```

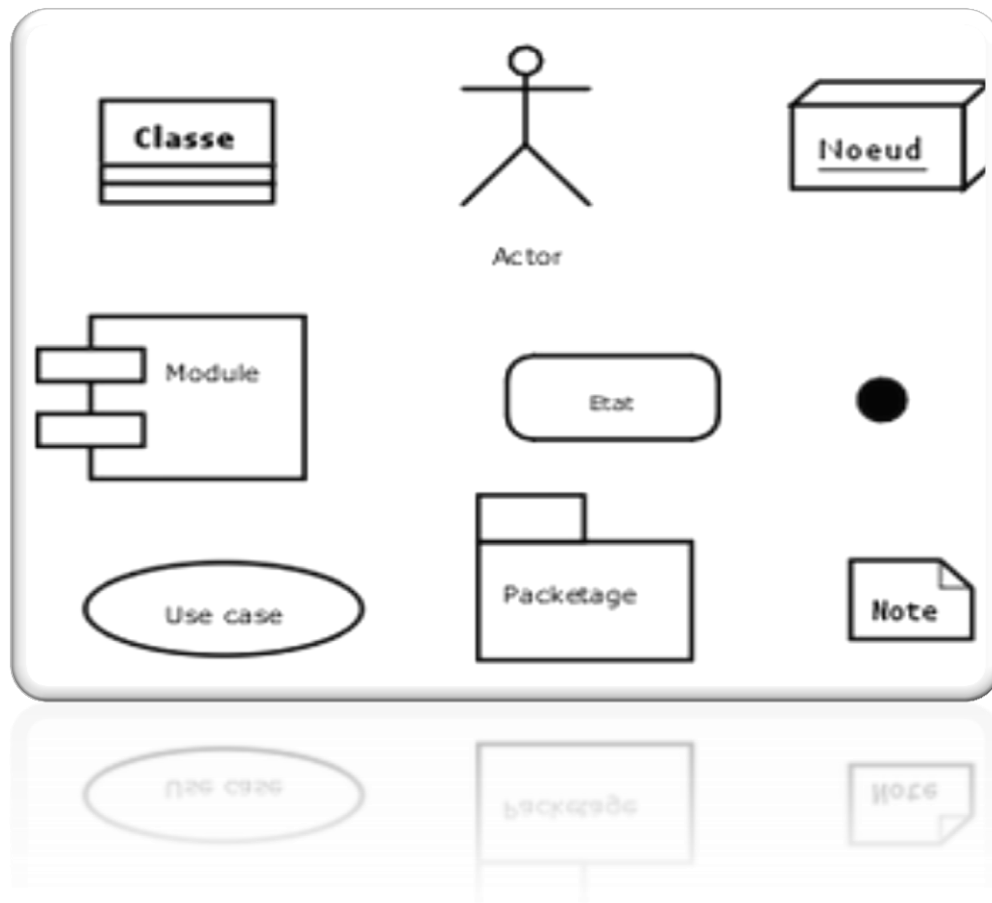
Listing 11 : Modifier les données

10. Conclusion

Dans ce chapitre, nous avons vu un aperçu sur le Web sémantique et ses langages tels que RDF, RDFS, SPRQL, OWL. En particulier, nous avons donné un détail sur le langage des ontologies, à savoir OWL, et nous avons vu comment construire et manipuler une ontologie avec ce langage en utilisant des outils tels que Protège, API Jena de java. Maintenant nous allons passer à l'étape de conception et l'implémentation de notre application.

Chapitre V :

Conception



1. Introduction

La conception est une étape préliminaire et primordiale qui doit précéder l'étape de développement de toute application informatique. Pour décrire la conception de l'application, on commencera par les diagrammes de cas d'utilisation, les diagrammes de séquence, diagramme de classes et les diagrammes d'activité. Par la suite on passera à l'ontologie.

2. UML

1.1. Définition

UML (sigle désignant l'unified modeling language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes, il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet» UML est couramment utilisé dans les projets logiciels. UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet, UML offre un standard de modélisation, pour représenter l'architecture logicielle.[2]

1.2. Les type de diagramme UML

UML 2.0 comporte ainsi treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information.

Ils se répartissent en deux grands groupes : [23]

1.2.1. Diagrammes structurels ou diagrammes statiques (UML Structure)

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

1.2.2. Diagrammes comportementaux ou diagrammes dynamiques (UML Behavior)

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

1.2.3. Diagrammes d'interaction (Interaction diagram)

- diagramme de séquence (Sequence diagram)
- diagramme de communication (Communication diagram)
- diagramme global d'interaction (Interaction overview diagram)
- diagramme de temps (Timing diagram)

Nous avons utilisés dans notre projet dans l'étape de conception Ces diagrammes : diagrammes de classes, cas d'utilisation, activité et de séquence.

3. Conception du système

3.1. Les acteurs de notre projet

3.1.1. Le visiteur

C'est un individu qui est en train de fouiller sur le net, cherchant un produit pour l'acheter ou pour avoir une idée sur les modèles et les prix. Jusqu' à ce stade c'est un utilisateur inconnu donc il n'est pas encore un client. Il doit être inscrit pour devenir un client.

3.1.2. Le Client

Cette acteur est un visiteur ayant déjà créé un compte sur notre application, il peut se connecter dans le site pour créer des annonces. L'administrateur : Pour les applications ou pour le site web on l'appelle généralement « le webmaster ». C'est celui qui assure le dynamisme du site et du contrôle des mises à jour des produits. En plus il gère les annonces et les clients.

3.1.3. L'administrateur

Pour les applications ou pour le site web on l'appelle généralement « le webmaster ». C'est celui qui assure le dynamisme du site et du contrôle des mises à jour des produits. En plus il gère les annonces et les clients.

3.2. Diagramme de cas d'utilisation

❖ Diagramme de cas d'utilisation

La figure 25 présente le diagramme de cas d'utilisation.

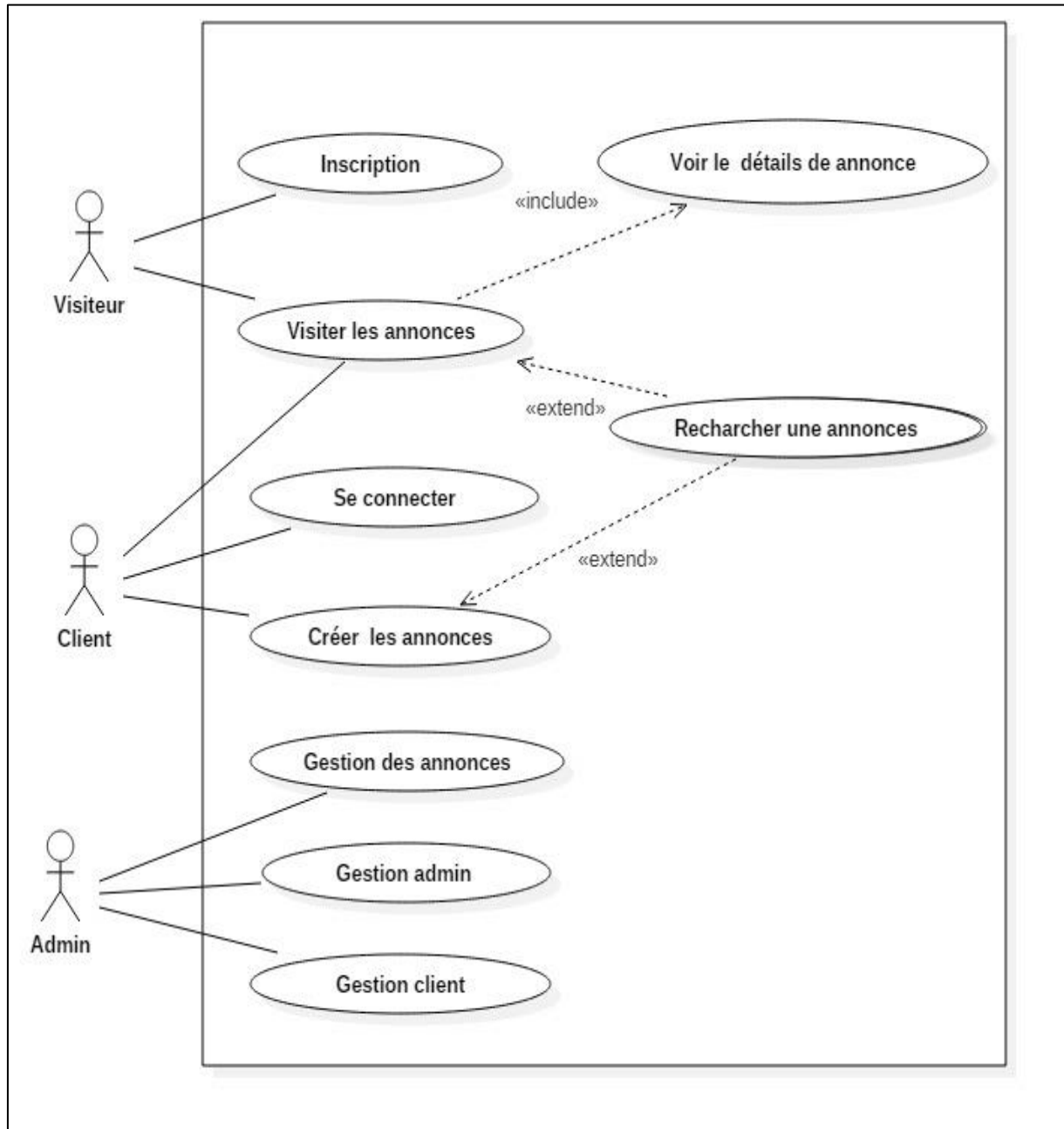


Figure 25 : Diagramme de cas d'utilisation

3.3. Diagramme de Séquence

❖ Diagramme de Séquence de l'Acteur Client et visiteur

La figure 26 présente le diagramme de Séquence d'un visiteur, Dans ce diagramme le visiteur et le client visitent l'application, en plus ils peuvent faire des recherches pour des annonces

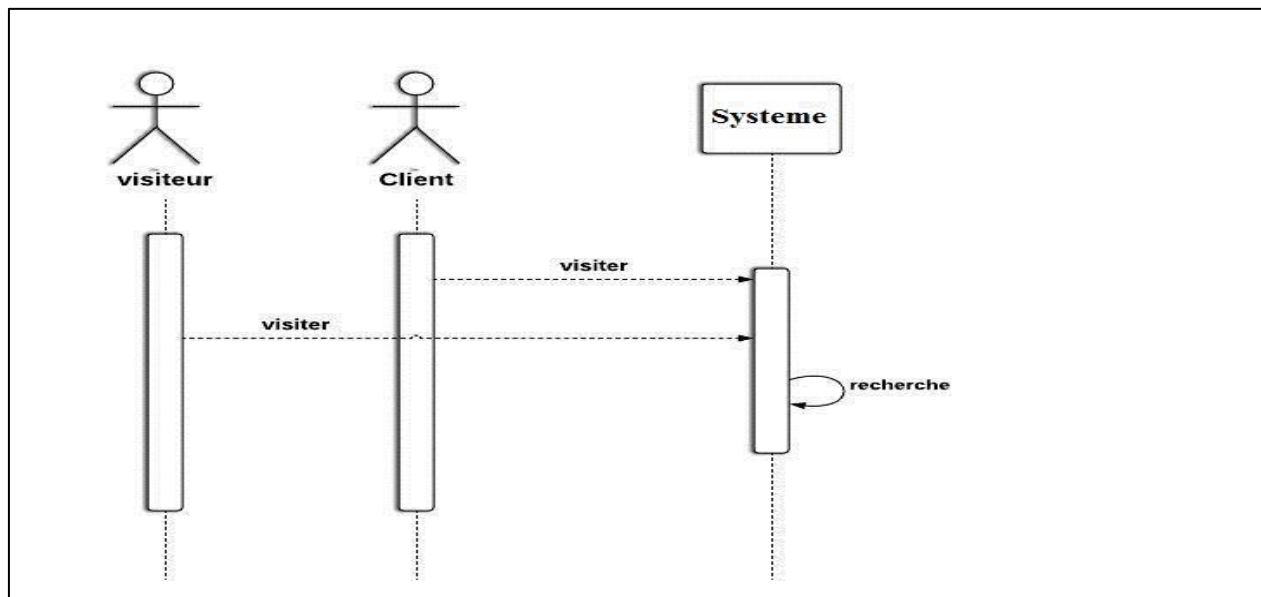


Figure 26 : Diagramme de Séquence entre visiteur, client

❖ Diagramme de séquences d'une inscription :

La figure 27 montre le diagramme de Séquence d'inscription client.

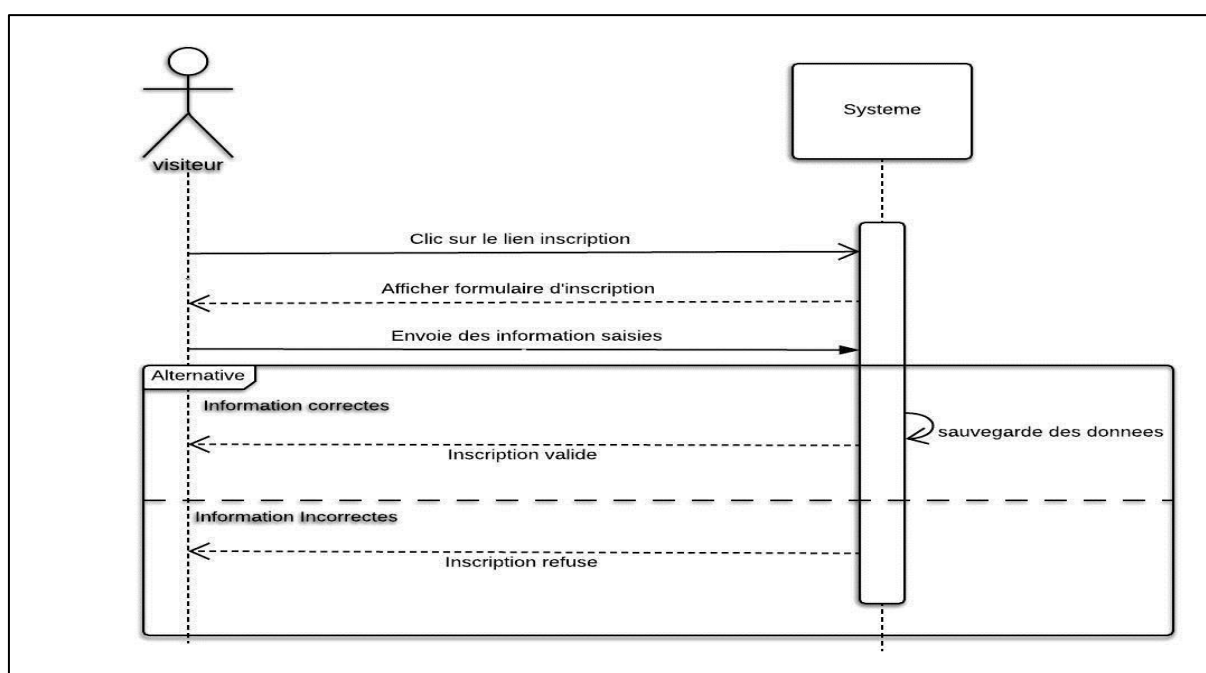


Figure 27 : Diagramme de Séquence D'inscription Client.

La figure 28 présente le diagramme de Séquence authentification client.

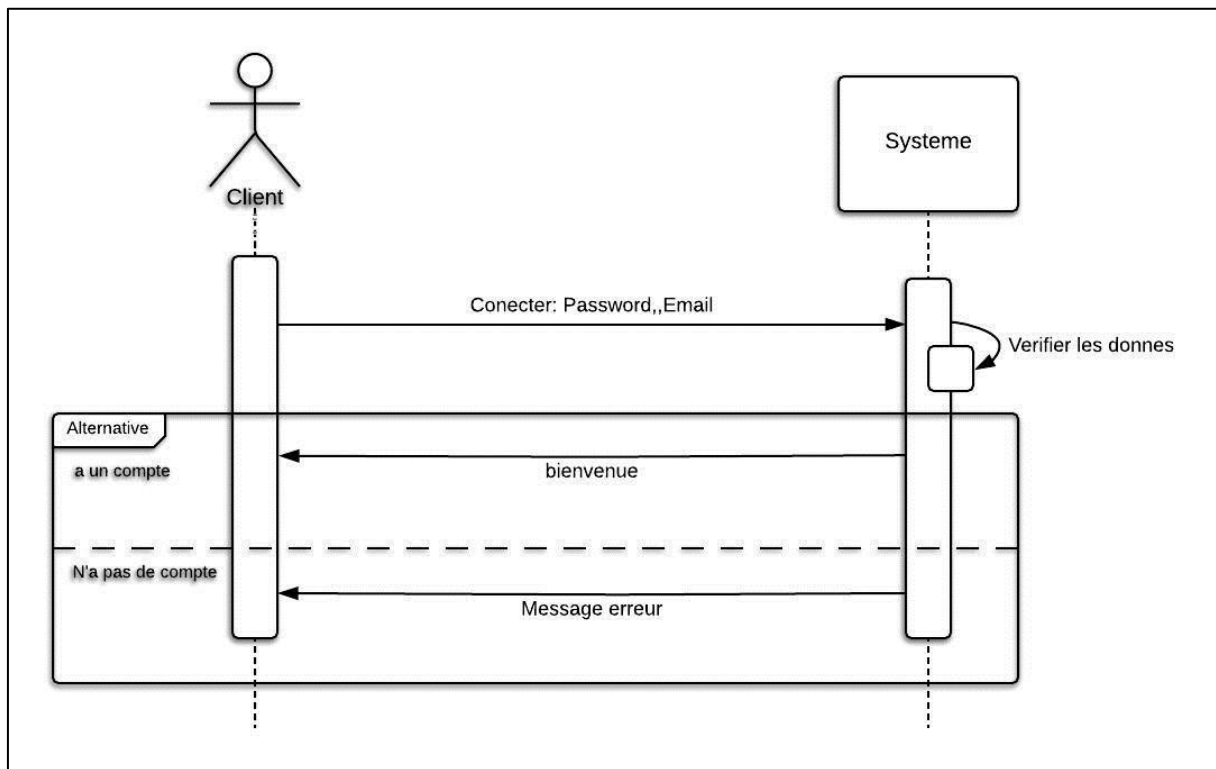


Figure 28 : Diagramme de Séquence Authentification Client.

La figure 29 montre le diagramme de Séquence Création annonce

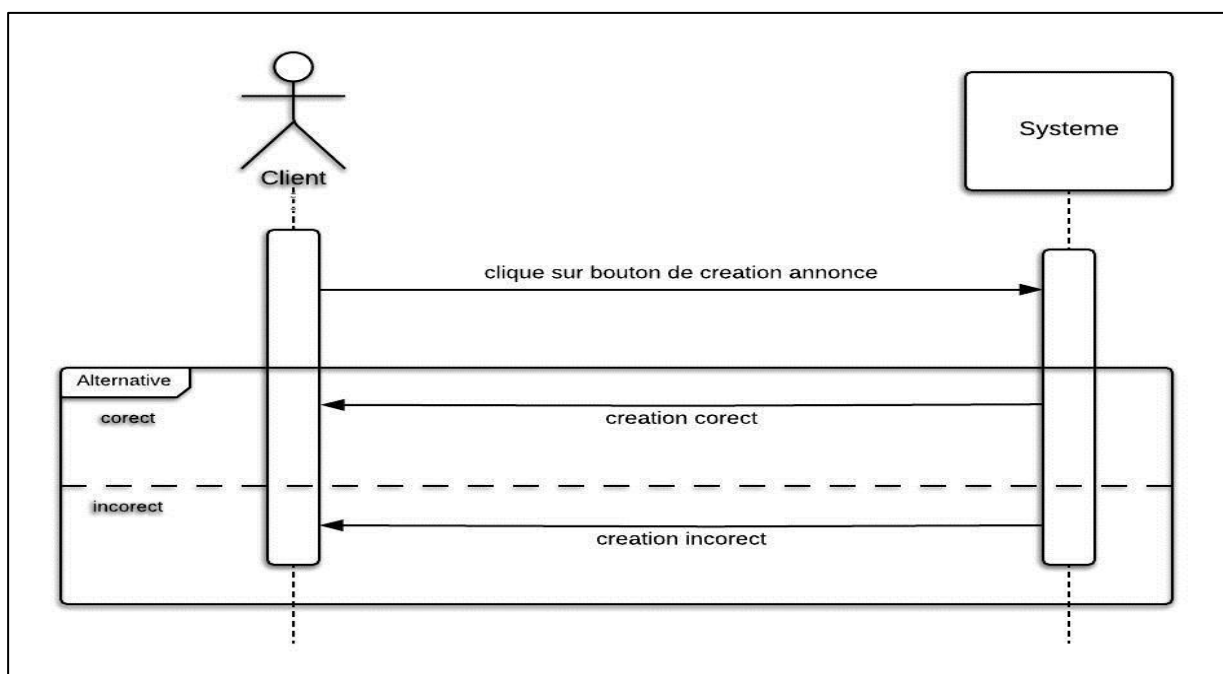


Figure 29 : Diagramme de Séquence Création annonce

La figure 30 présente le diagramme de Séquence profile

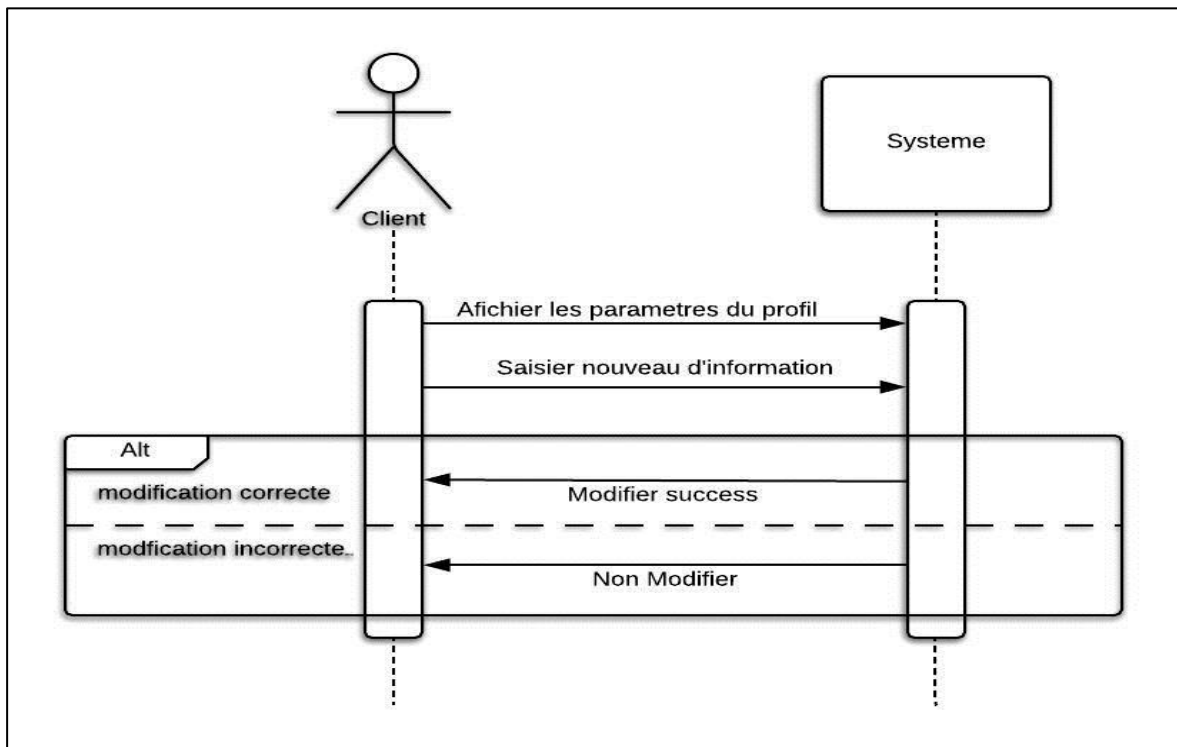


Figure 30 : Diagramme de Séquence gestion profile du client

La figure 31 présente le diagramme de Séquence log out

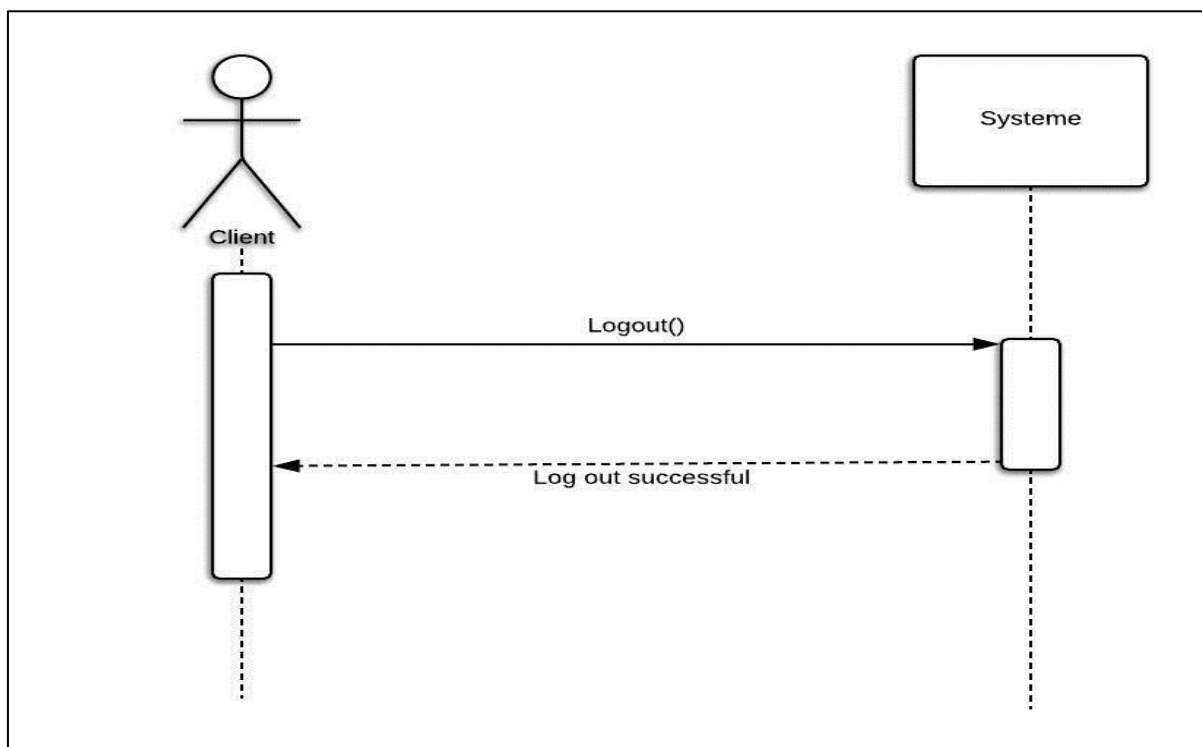


Figure 31 : Diagramme de Séquence log out

❖ Diagramme de Séquence de l'Acteur administrateur :

La figure 32 présente le diagramme de Séquence Authentication administrateur.

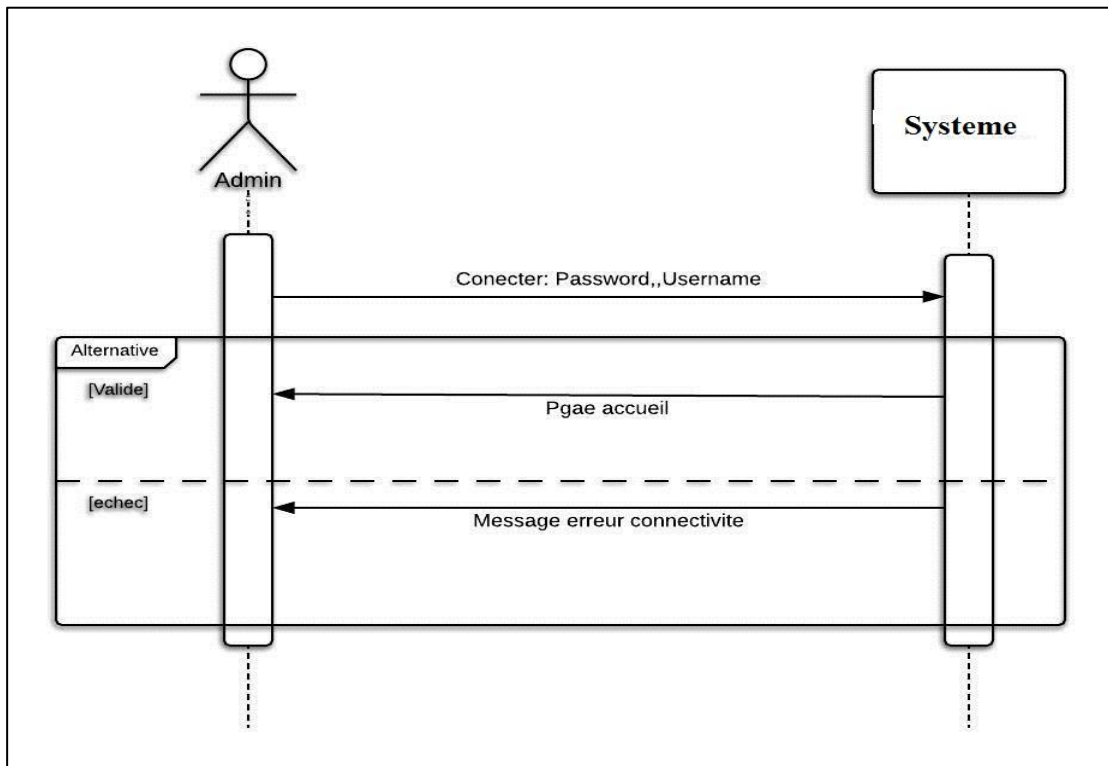


Figure 32 : Diagramme de Séquence Authentication administrateur.

La figure 33 présente le diagramme de Séquence consultation les annonces par 'administrateur

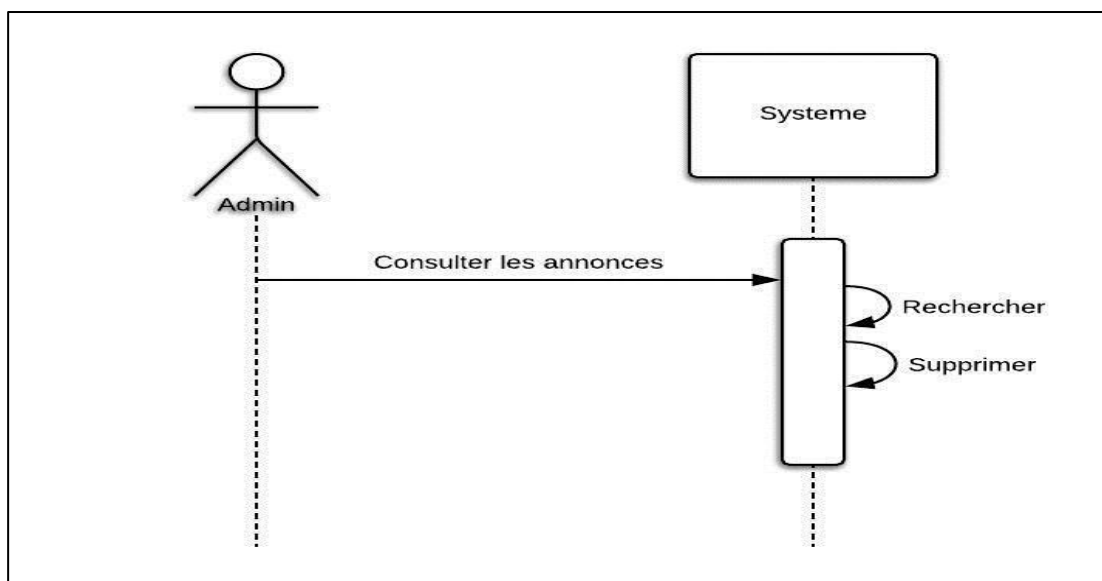


Figure 33 : Diagramme de Séquence consulter les annonces par l'administrateur

La figure 34 présente le diagramme de Séquence gestion profil d'un administrateur

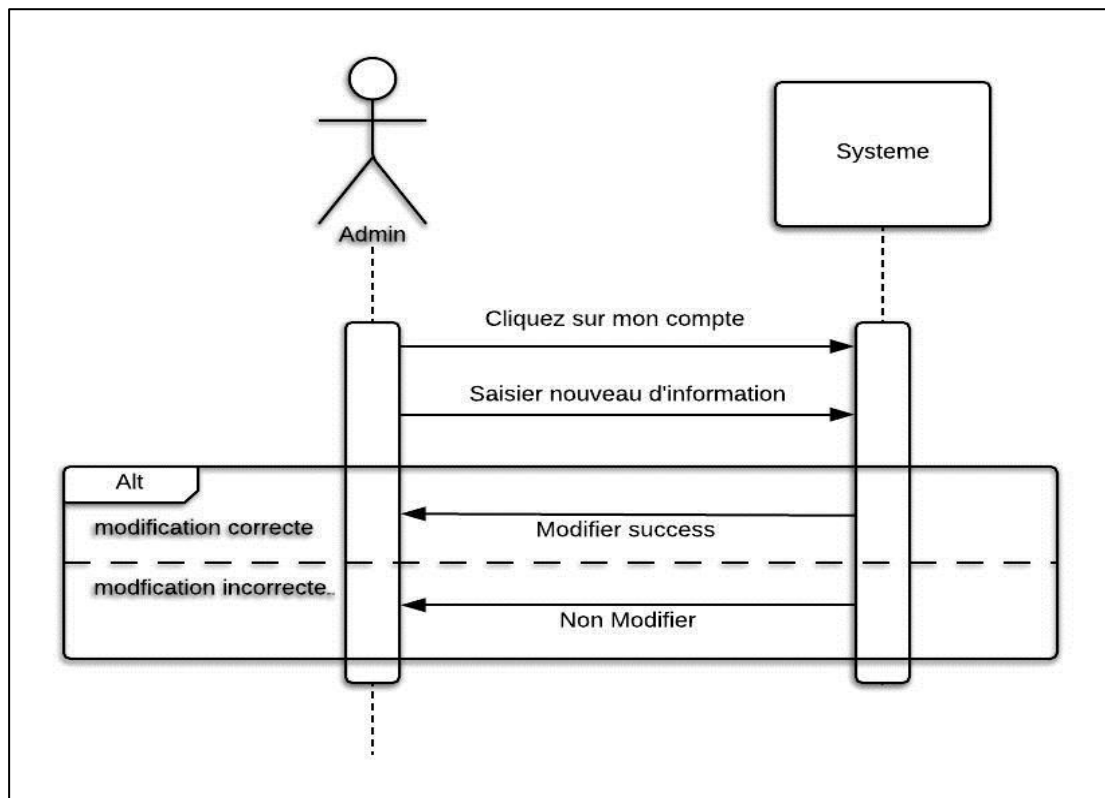


Figure 34 : Diagramme de Séquence gestion profil d'un administrateur

3.4. Diagrammes d'Activité

❖ Diagramme d'Activité Client et visiteur

La figure 35 présente le Diagramme d'Activité Client et visiteur

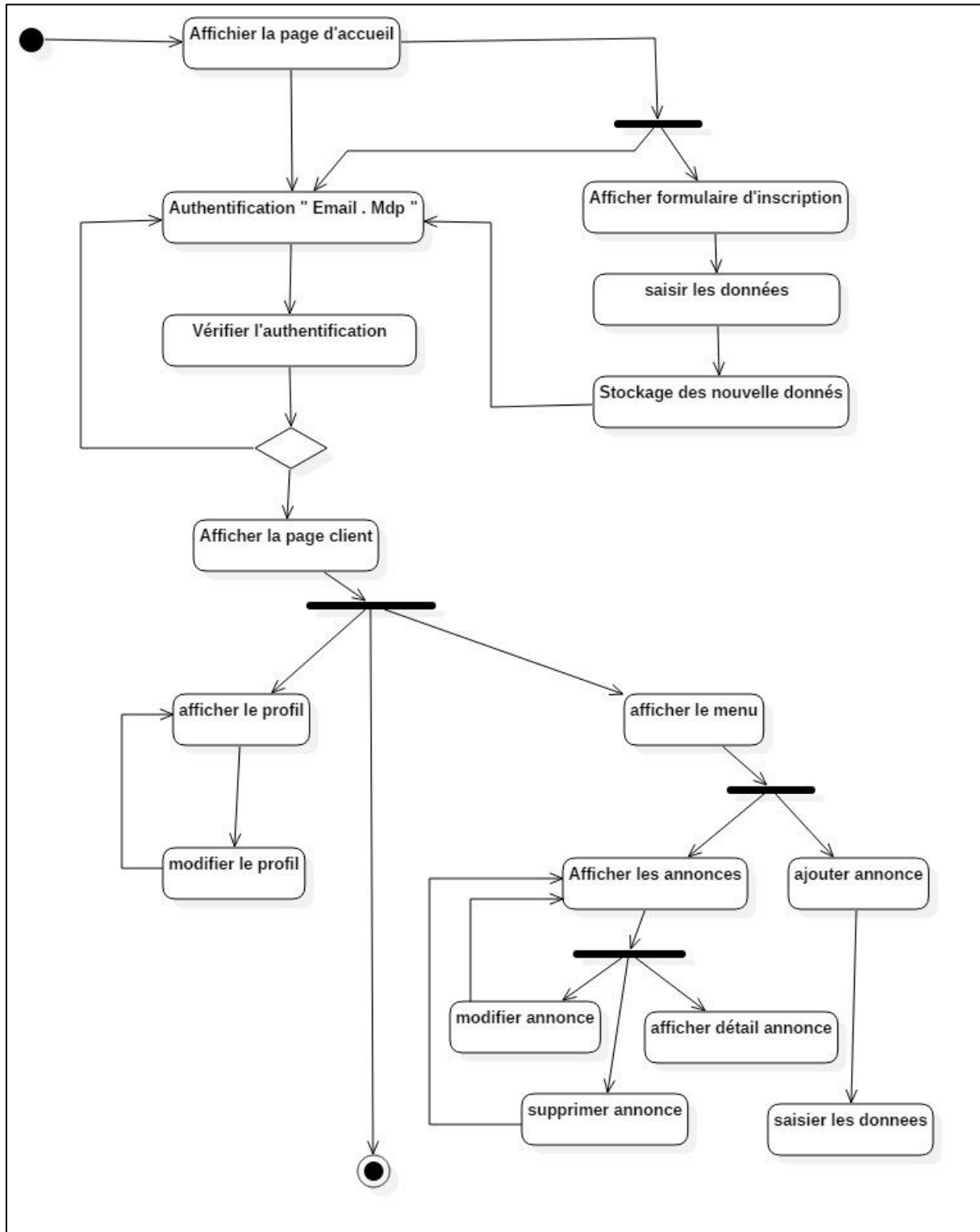


Figure 35 : Diagramme d'Activité Client et visiteur

❖ Diagramme d'Activité l'administrateur

La figure 36 montre le Diagramme l'administrateur

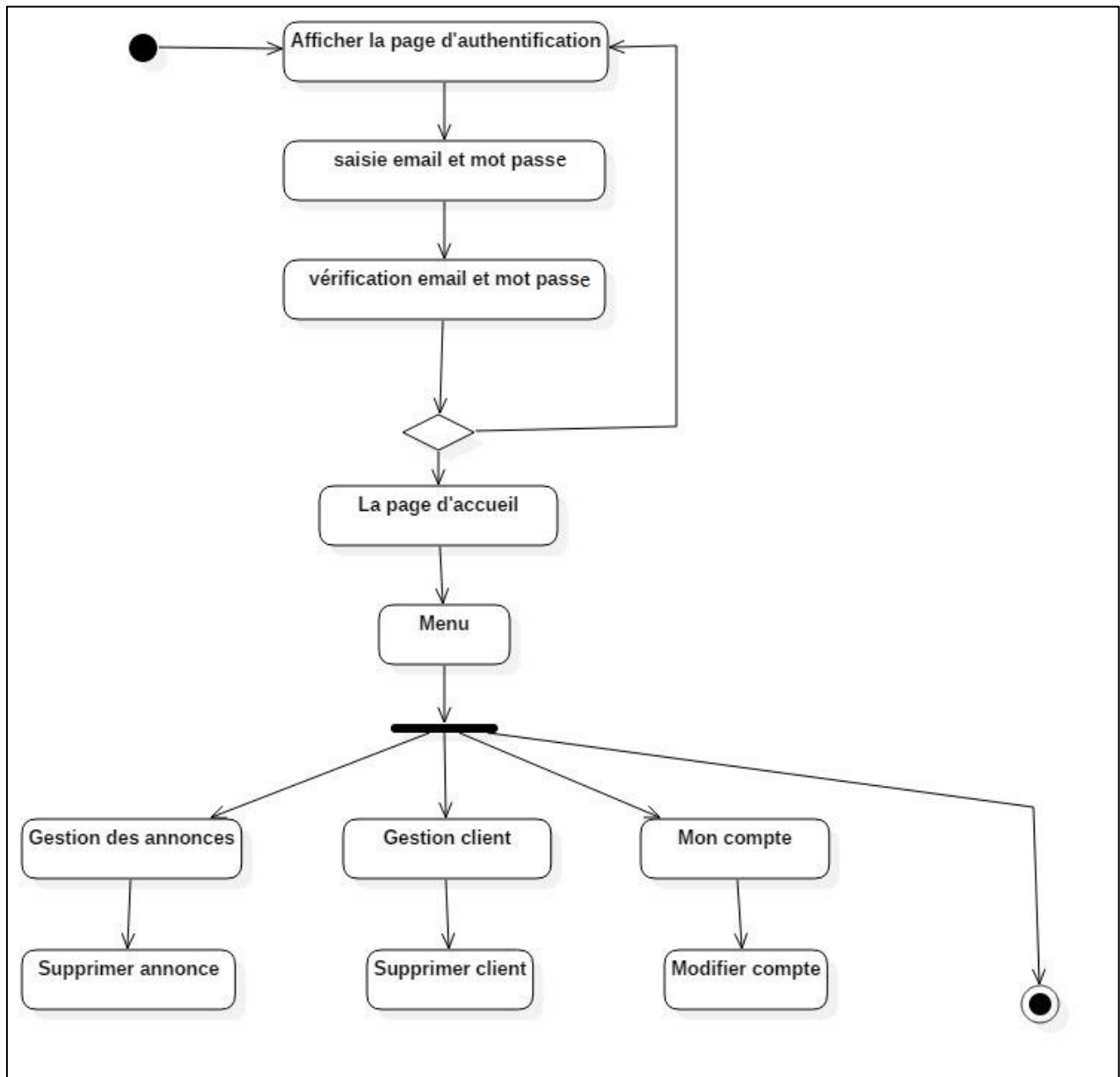


Figure 36 : Diagramme d'Activité d'administrateur

3.5. Diagrammes de classes

Le diagramme de classes identifie les différentes classes de notre application Web et les associations entre elles. Ce diagramme contient six classes. Il se donne par la figure 37 :

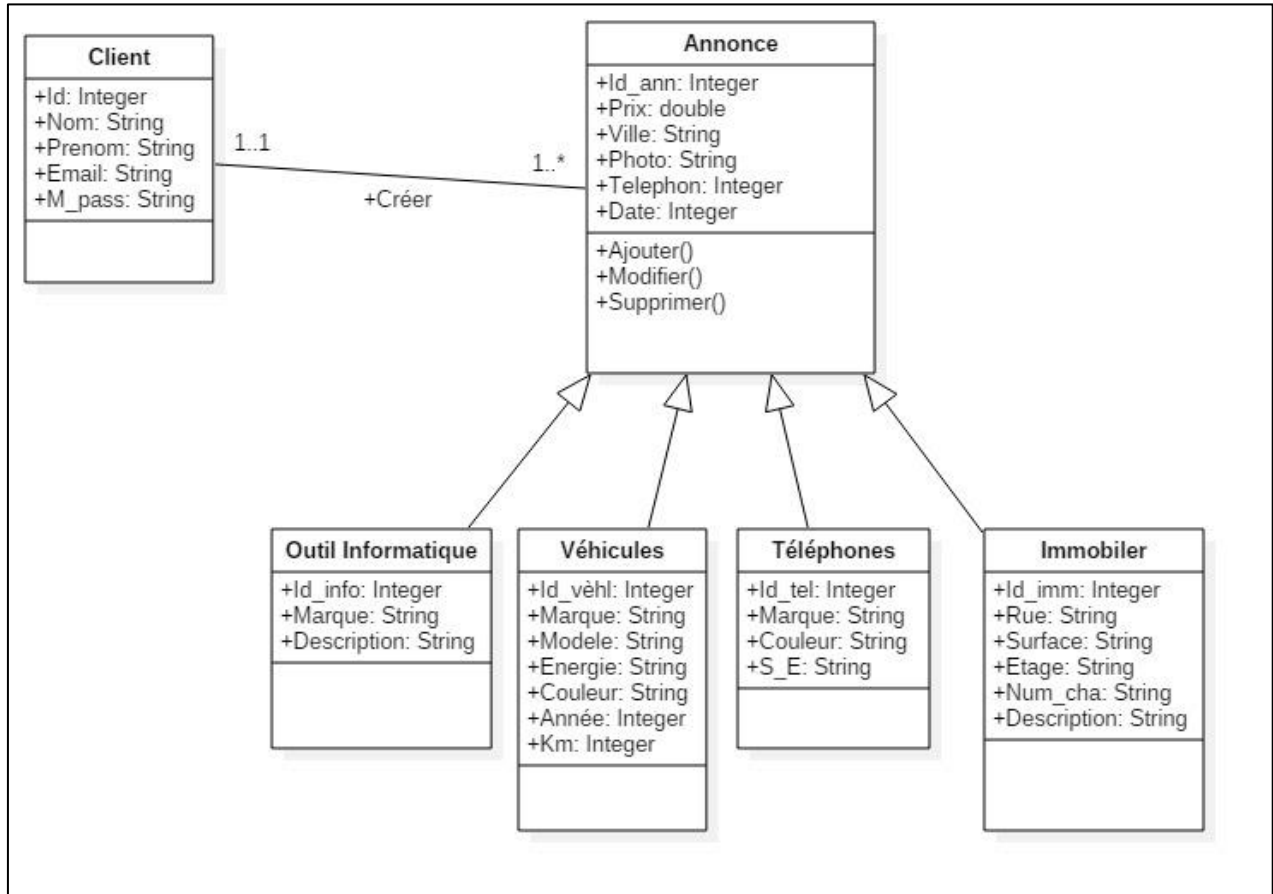


Figure 37 : Diagramme de classe.

4. L'ontologie proposée

L'ontologie modélise un ensemble de connaissances dans un domaine donné. la figure 38 présente l'ontologie de notre système. Elle est compose de deux classes principales client et annonce, le client peut créer une ou plusieurs annonces ce qui est illustré par la relation créer par. Une annonce peut être une annonce informatique ou Immobilier ou véhicule ou bien téléphone.

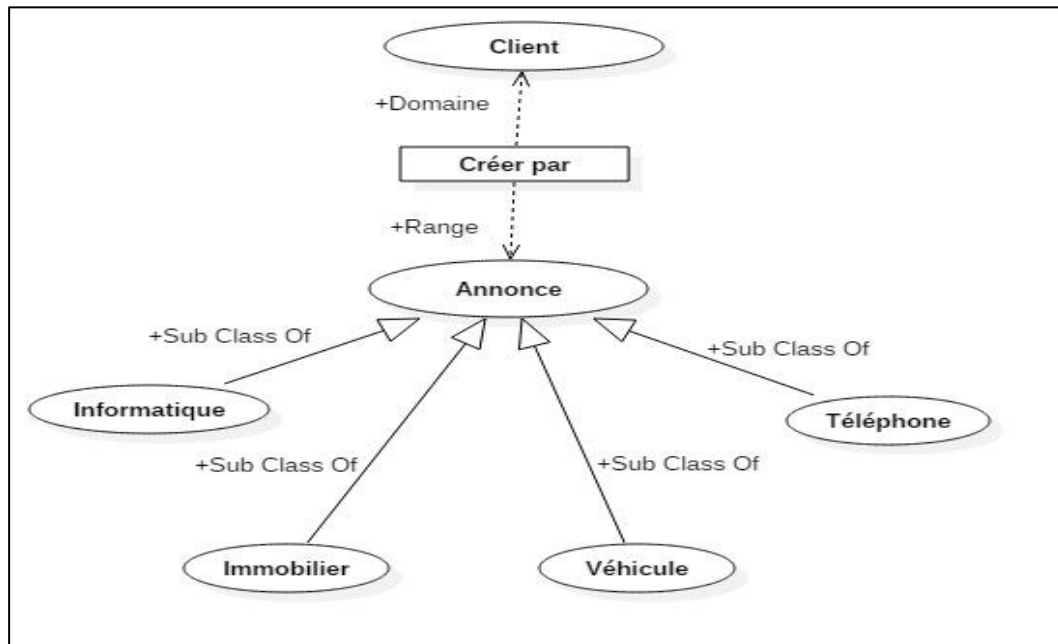


Figure 38 : l'ontologie de gestion des annonces

Nous avons utilisé éditeur protégé pour la manipulation et la représentation d'ontologies

La figure 15 montre un Onto Graf de l'ontologie de gestion des annonces

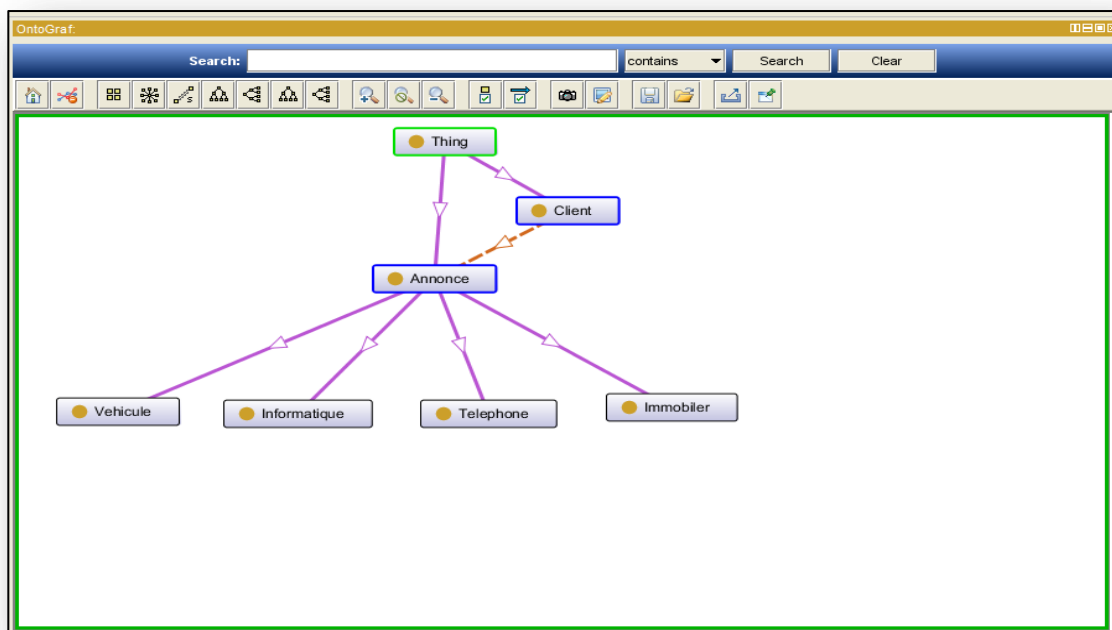


Figure 39 : L'ontographe de l'ontologie de gestion des annonces

5. Conclusion

Nous avons présenté dans ce chapitre la conception de notre application web en utilisant le langage de modélisation UML. Cette conception est une étape nécessaire et très importante pour pouvoir passer, à l'étape de réalisation de notre application. Le prochain chapitre détaillera les étapes de l'implémentation de notre application.

Chapitre VI:

Implémentation

1. Introduction

L'implémentation est la phase la plus importante après celle de la conception. Dans ce chapitre nous présentons deux parties importantes : la première sert à présenter les outils utilisés dans le développement de ce projet, et la deuxième pour l'implémentation, on donne dans cette partie une présentation de l'interface de l'application sous forme d'un guide.

2. Les outils utilisés

2.1. Apache Tomcat

Apache Tomcat est une implémentation open source d'un conteneur web qui permet donc d'exécuter des applications web reposant sur les technologies servlets et JSP.

Tomcat est diffusé en open source sous une licence Apache. C'est aussi l'implémentation de référence des spécifications servlets jusqu'à la version 2.4 et JSP jusqu'à la version 2.0 implémentées dans les différentes versions de Tomcat.[13]

2.2. Protégé

Protégé est un logiciel gratuit (JAVA), plate-forme open-source qui fournit une suite d'outils pour construire des bases de connaissances et des ontologies.

Protégé inclut de nombreux plugins pour la manipulation et la représentation d'ontologies dans différents formats.

L'éditeur Protégé-OWL permet aux utilisateurs de construire des ontologies pour le Web sémantique en OWL.[24]

2.3. Jena

Qui est un Framework Java pour construire les applications du web sémantique. Il fournit un environnement de programmation pour RDF, RDFS and OWL, SPARQL et comprend une règle basée sur le moteur d'inférence. Cette API qui de l'environnement java est utilisée pour télécharger l'ontologie et remplir les individus.

2.4. NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en Open Source par Sun. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, PHP, HTML ... Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris, Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Développement Kit (JDK) est requis pour les développements en Java.[25]

2.5. StarUML

Le logiciel StarUML est un logiciel open source cédé par son ancien éditeur sous licence GNU GPL, dédié aux plateformes Windows, il est développé à Delphi. Ses principaux avantages sont son implicité d'installation et de prise en main, et la possibilité de générer le squelette des classes en langages Java, C++, C#, Action Script3.0.... De plus, le logiciel a été conçu en prévoyant l'ajout de plugins supplémentaires afin de pouvoir être adapté simplement aux besoins évolutifs dès ses utilisateurs. Enfin StarUML gère l'exportation des données au format XMI, le standard pour l'échange d'informations de métadonnées UML basé sur XML.[26]

3.4. Page création annonce

La figure 43 présente la création d'une annonce téléphone :



Figure 43 : création d'une annonce

3.5. Page des annonces de client

La figure 44 représente les annonces publiées par client. Cette page vous permet d'effectuer une opération de suppression ou modification d'une annonce:

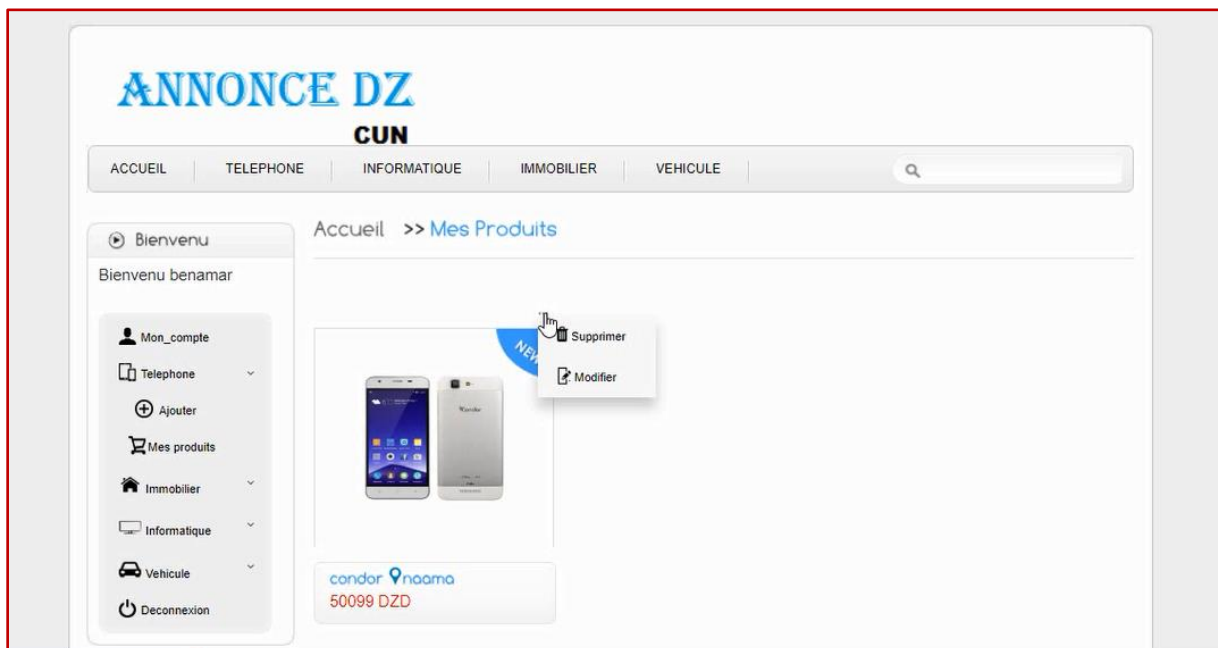


Figure 44: Page des annonces client

3.6. Page détaillée d'une annonce

La figure 45 montre la page de détail d'une annonce

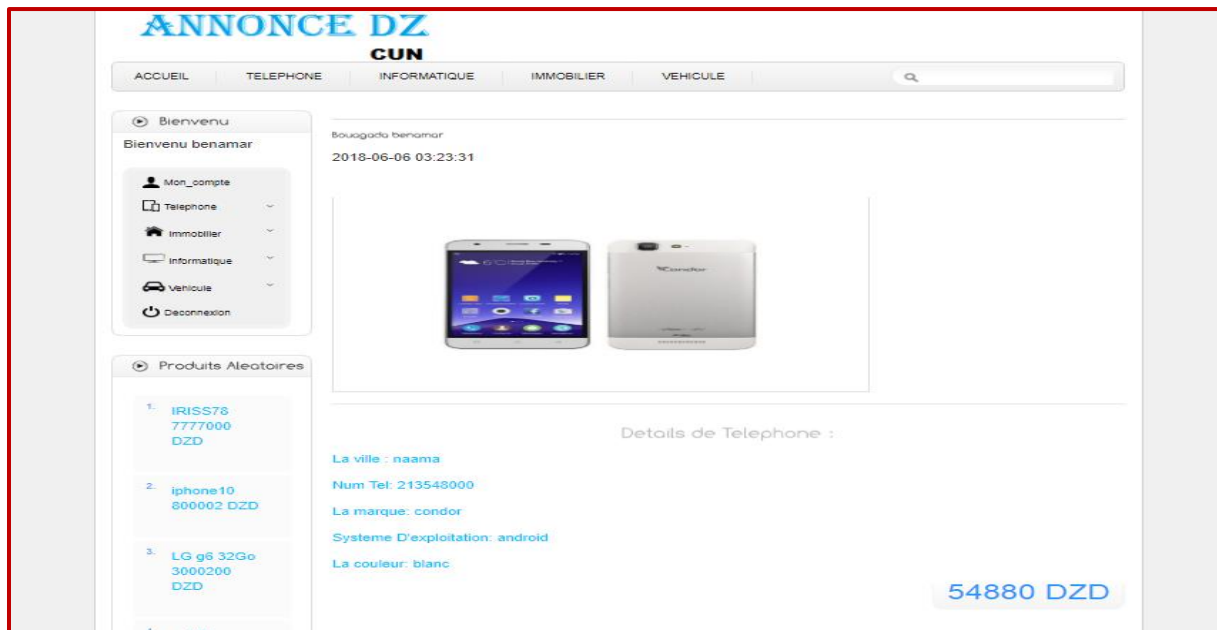


Figure 45 : Page détaillée d'une annonce

3.7. Page « modifier les annonces de client »

Cette interface graphique représente la modification des annonces de ce client

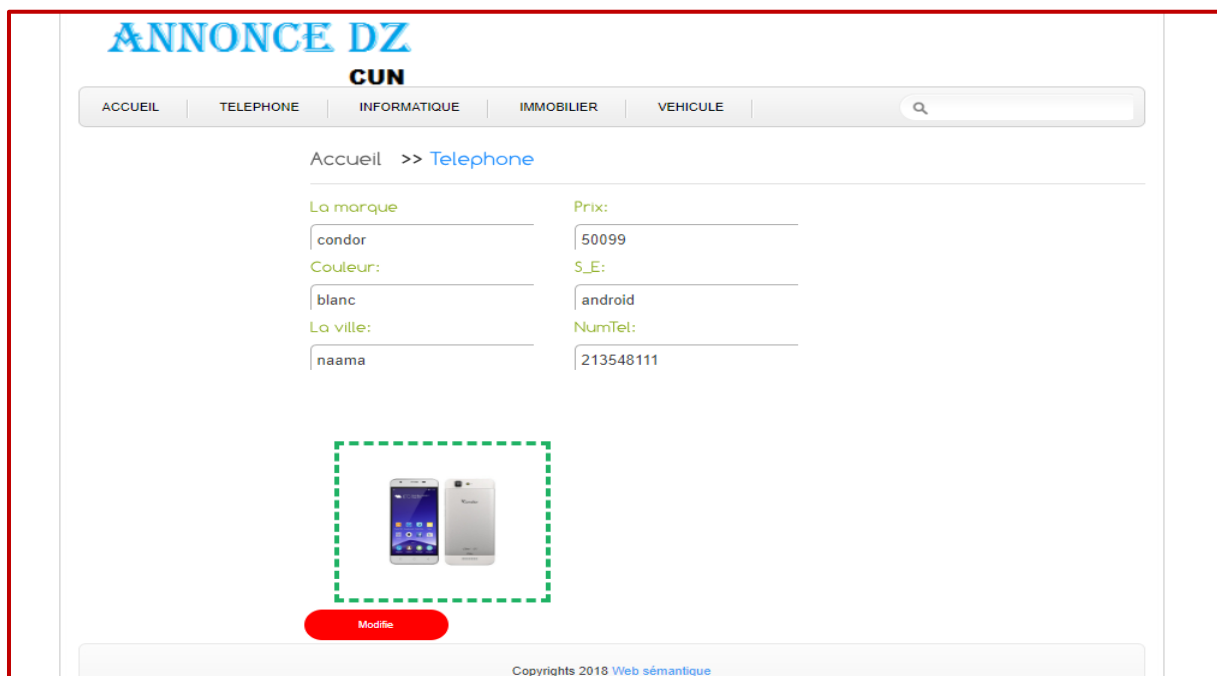


Figure 46 : Page de modification des annonces des clients

3.8. Page « modifier les informations de client »

La figure 47 représenté la modification des informations de client :

The screenshot shows the 'ANNONCE DZ CUN' website interface. At the top, there is a navigation menu with links for ACCUEIL, TELEPHONE, INFORMATIQUE, IMMOBILIER, and VEHICULE, along with a search bar. The main content area is titled 'Accueil >> Mon Compte'. On the left, a sidebar displays 'Bienvenu benamar' and a list of menu items: Mon_compte, Telephone, Immobilier, Informatique, Vehicule, and Deconnexion. The main form area contains the following fields: 'Le nom' (Bouagada), 'Le Prenom' (benamar), 'Email' (benamar@gmail.com), and 'Mot de passe' (masked with dots). A 'Modifier' button is located below the password field. At the bottom left, there is a 'Produits Aleatoires' section showing a product 'oppo' for '9000 DZD'.

Figure 47 : Page de modification les informations de client

3.9. Page « authentification administrateur »

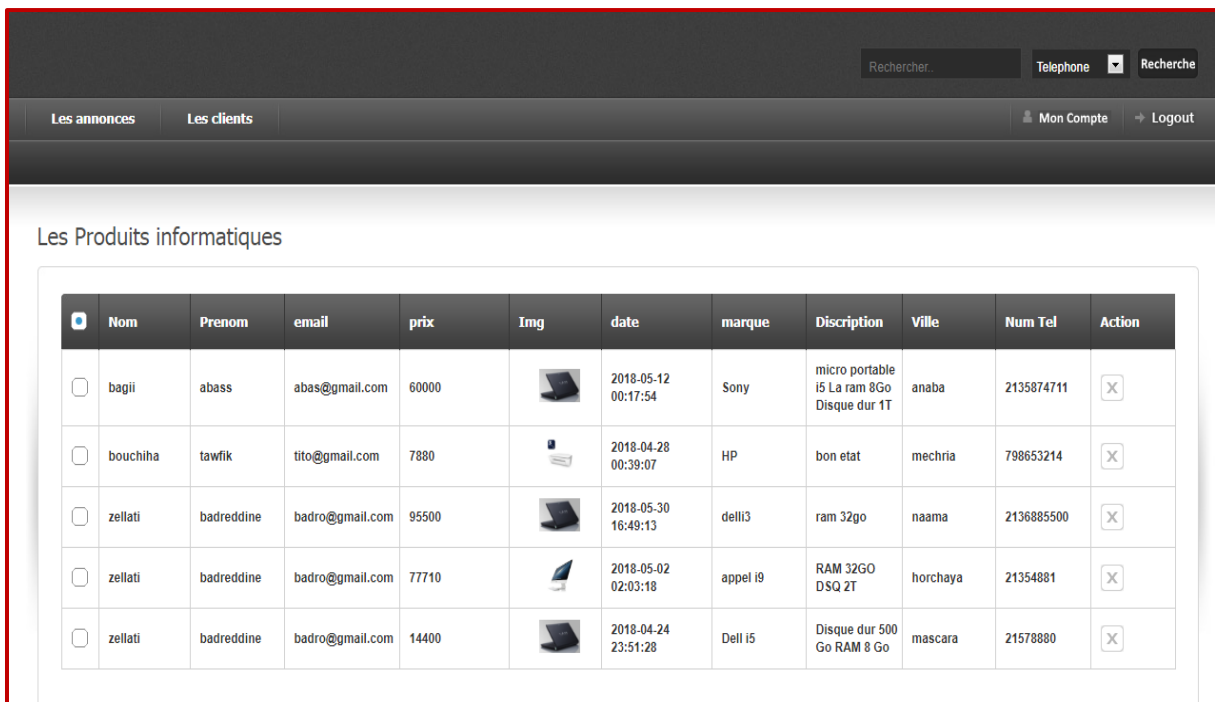
La figure 48 montre l'authentification de l'administrateur:

The screenshot shows the 'Administration' page for 'ANNONCE DZ CUN'. The page has a dark background with the website logo at the top. The main content is a login form with the following elements: 'Nom d'utilisateur' (text input), 'Mot de passe' (password input with dots), a checkbox for 'Se rappeler de moi', and a red 'Se connecter' button.

Figure 48 : La page d'authentification de l'administrateur

3.10. Gestion des annonces

La figure 49 présente la gestion des annonces (informatique) par administrateur :








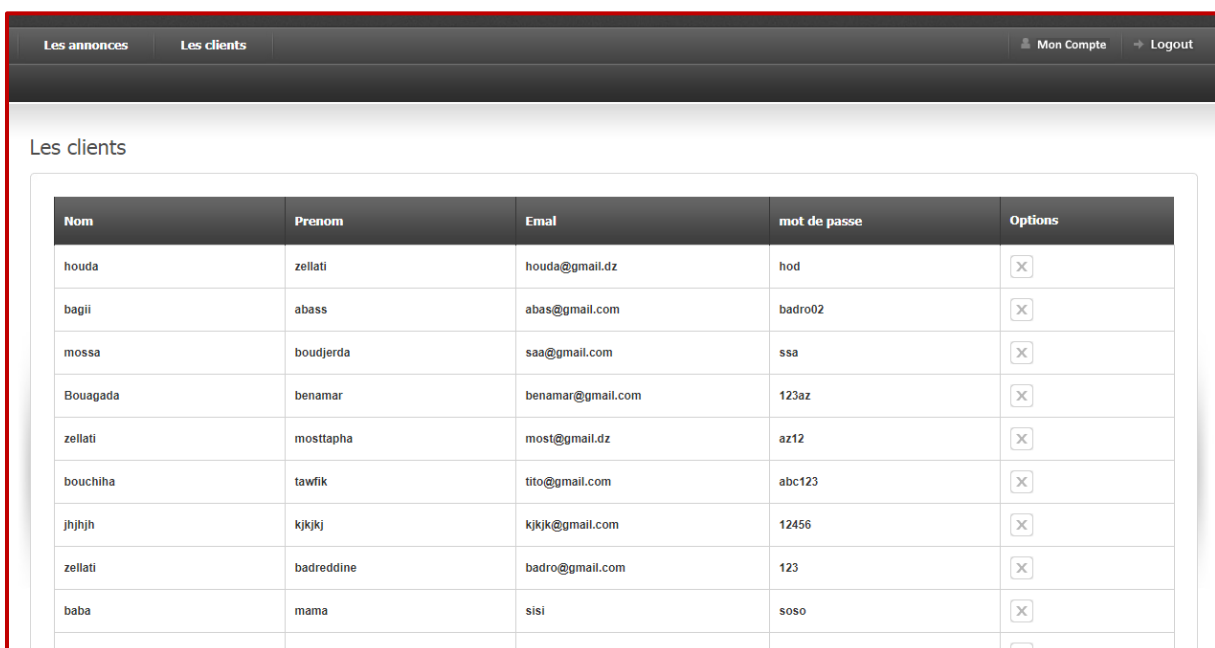
<input type="checkbox"/>	Nom	Prenom	email	prix	Img	date	marque	Discription	Ville	Num Tel	Action
<input type="checkbox"/>	bagii	abass	abas@gmail.com	60000		2018-05-12 00:17:54	Sony	micro portable i5 La ram 8Go Disque dur 1T	anaba	2135874711	<input type="checkbox"/>
<input type="checkbox"/>	bouchiha	tawfik	tito@gmail.com	7880		2018-04-28 00:39:07	HP	bon etat	mechria	798653214	<input type="checkbox"/>
<input type="checkbox"/>	zellati	badreddine	badro@gmail.com	95500		2018-05-30 16:49:13	dell i3	ram 32go	naama	2136885500	<input type="checkbox"/>
<input type="checkbox"/>	zellati	badreddine	badro@gmail.com	77710		2018-05-02 02:03:18	appel i9	RAM 32GO DSQ 2T	horchaya	21354881	<input type="checkbox"/>
<input type="checkbox"/>	zellati	badreddine	badro@gmail.com	14400		2018-04-24 23:51:28	Dell i5	Disque dur 500 Go RAM 8 Go	mascara	21578880	<input type="checkbox"/>

Figure 49 : Gestion Des Annonces

3.11. Page de gestion des clients

La figure 50 présente la gestion des clients par administrateur :

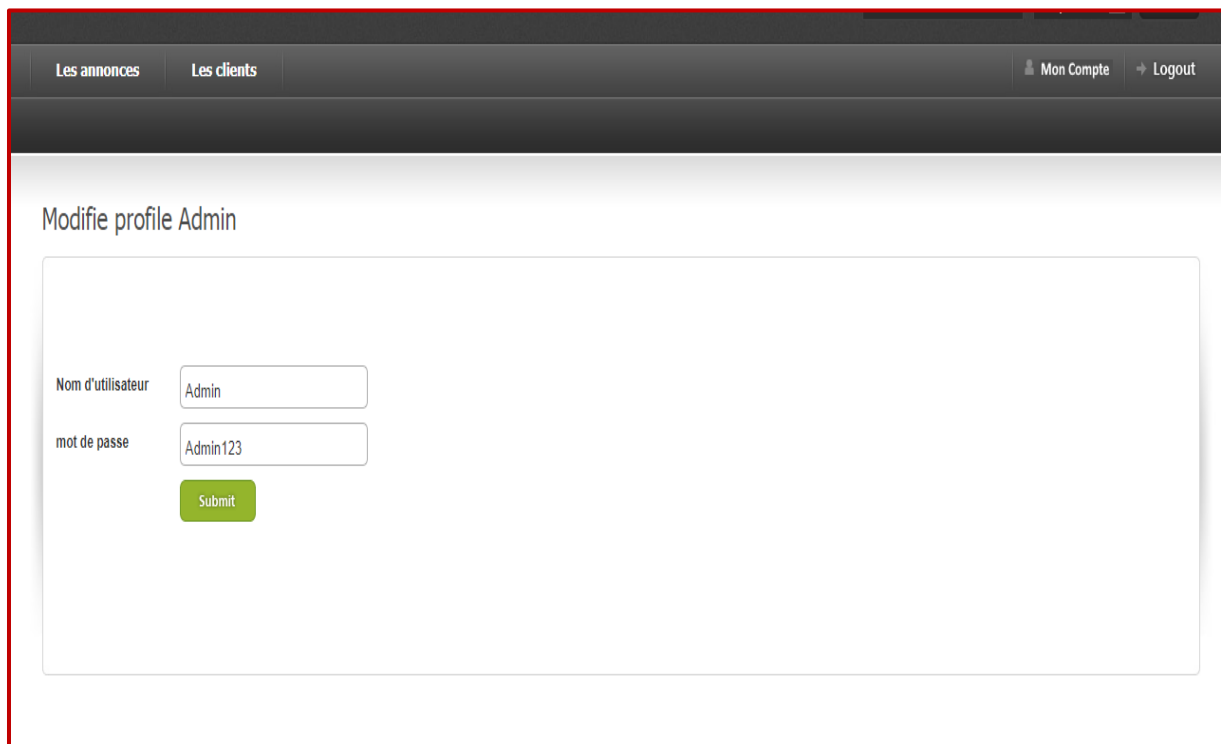


Nom	Prenom	Email	mot de passe	Options
houda	zellati	houda@gmail.dz	hod	<input type="checkbox"/>
bagii	abass	abas@gmail.com	badro02	<input type="checkbox"/>
mossa	boudjerda	saa@gmail.com	ssa	<input type="checkbox"/>
Bouagada	benamar	benamar@gmail.com	123az	<input type="checkbox"/>
zellati	mostapha	most@gmail.dz	az12	<input type="checkbox"/>
bouchiha	tawfik	tito@gmail.com	abc123	<input type="checkbox"/>
jthjhj	kjkjkj	kjkjk@gmail.com	12456	<input type="checkbox"/>
zellati	badreddine	badro@gmail.com	123	<input type="checkbox"/>
baba	mama	sis	soso	<input type="checkbox"/>
bon	bon	bon@naama.com	2018	<input type="checkbox"/>

Figure 50 : Page gestion des clients par l'administrateur

3.12. Page de modification du profil de l'administrateur

La figure 51 montre la modification de profil d'un administrateur :



The screenshot shows a web application interface for modifying an administrator's profile. At the top, there is a dark navigation bar with links for 'Les annonces', 'Les clients', 'Mon Compte', and 'Logout'. Below this, the page title is 'Modifie profile Admin'. The main content area contains a form with two input fields: 'Nom d'utilisateur' (containing 'Admin') and 'mot de passe' (containing 'Admin123'). A green 'Submit' button is positioned below the password field.

Figure 51 : Page de modification profile de l'administrateur

4. Conclusion

Dans ce dernier chapitre, nous avons présenté la partie réalisation de notre projet avec les outils utilisés et nous avons décrit les interfaces les plus importantes de notre application.



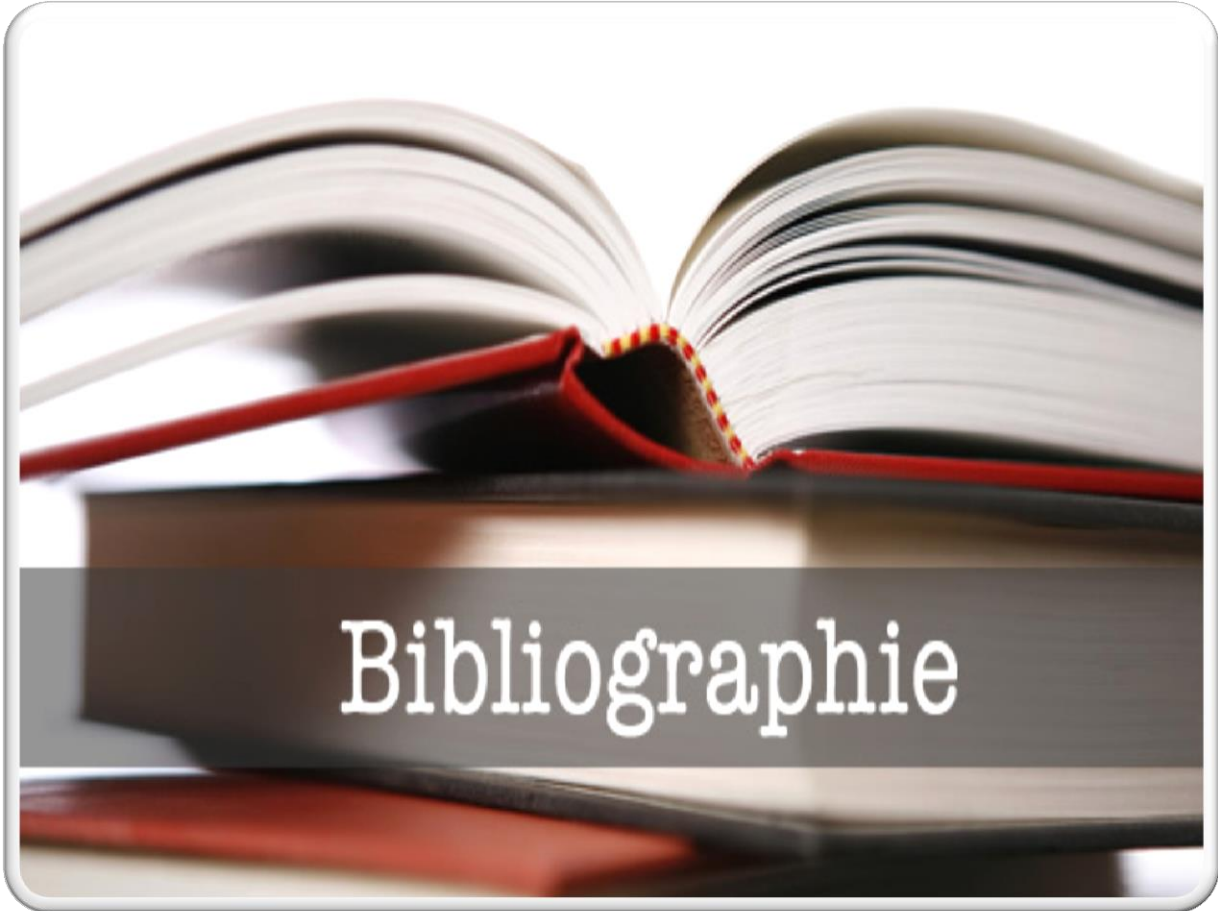
Générale

Conclusion générale

Au cours de ce mémoire, nous avons choisi comme sujet Conception et réalisation d'une application Web sémantique pour la gestion des annonces de produits. Au cours de ce mémoire, nous avons présenté les différentes étapes de la conception et la réalisation de notre application.

La réalisation de ce projet nous a permis d'apprendre à organiser, à travailler en groupe, d'améliorer nos connaissances et nos compétences dans le développement des applications Web sous Java EE.

Nous avons découvert le web sémantique ce qui nous a pris plus de temps que nous ne l'escomptions pour sa maîtrise et nous avons appris à mieux manipuler les langages de web sémantique tels que RDF, RDFS, OWL et SPARQL En effet, ce travail étant un essai, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer davantage cette initiative ."Comme perspective nous allons essayer de développer une application Androïde qui travaille avec notre application Web sémantique.



Bibliographie



Bibliographie

Références

- [1] FEYLER, F. Le Web sémantique. Une approche nouvelle de l'accès à l'information. Dernier accès le: 01/06/2018; disponible sur: <http://www.cndp.fr/ecolenumerique/tous-les-numeros/numero-5-septembre-2010/focus-recherche-documentaire/article/article/le-web-semantique-une-approche-nouvelle-de-lacces-a-linformation-pertinente.html>.
- [2] Bouchiha-zellati, ConceptionRéalisation d'une application Web e-Commerce pour la gestion des annonces de produits.. 2016 -2015.
- [3] LEUBOU, R.N. Cycle de vie d'une application web. Dernier accès le: 01/06/2018; disponible sur: <https://fr.linkedin.com/pulse/cycle-de-vie-dun-application-web-richard-ngu-leubou>. "16 /06/ 2016"
- [4] Termes et langages du web : HTML, CSS, PHP, MySQL et FTP. Dernier accès le: 02/06/2018; disponible sur: <https://lacliniquewp.com/comprendre-langages-du-web/>. "24 /07/2016"
- [5] Dynamisez vos sites web avec JavaScript ! 9/5/2017 Dernier accès le: 29/05/2018; disponible sur: <https://openclassrooms.com/courses/dynamisez-vos-sites-web-avec-javascript>.
- [6] JSP, c'est quoi. Dernier accès le: 05/06/2018; disponible sur: <http://www.opentuto.com/jsp-cest-quoi/>.
- [7] ASP - Active Server Pages - Introduction. Dernier accès le: 05/06/2018; disponible sur: <https://www.commentcamarche.com/contents/31-asp-active-server-pages-introduction>.
- [8] Intergiciel et Construction d'Applications Réparties. Dernier accès le: 05/06/2018; disponible sur: <http://www.lifl.fr/icar/Chapters/J2EE/j2ee-body.html>. "2006"
- [9] Edition La technologie java Enterprise. Dernier accès le: 05/06/2018; disponible sur: <https://www.scribd.com/document/223239591/La-technologie-java-Enterprise-Edition-docx>.
- [10] Créez votre application web avec Java EE. Dernier accès le: 05/06/2018; disponible sur: <http://sdz.tdct.org/sdz/creez-votre-application-web-avec-java-ee.html#Qu039est-cequeJavaEE>.
- [11] JAVA. Dernier accès le: 05/06/2018; disponible sur: https://www.java.com/fr/download/faq/whatis_java.xml.
- [12] Cycle de vie d'une JSP. Dernier accès le: 05/06/2018; disponible sur: <http://www.opentuto.com/cycle-de-vie-dune-jsp/>.

- [13] Jean-Michel. Développons en Java. 1999-2016 Dernier accès le: 05/06/2018; disponible sur: <https://www.jmdoudoux.fr/java/dej>.
- [14] NGOC, T.T.D., Java avancé. Université de Versailles-Saint-Quentin.
- [15] Thomas, G., Servlets et JSP. Université Pierre et Marie Curie
- [16] Genest, D., Web et représentation de données Web Sémantique. 2013-2014. Université Uni d'Angers.
- [17] Christiane. Du web 1.0 au web 4.0. Dernier accès le: 02/06/2018; disponible sur: <https://c-marketing.eu/du-web-1-0-au-web-4-0/>. "01/02/2012"
- [18] Plu, J. Introduction au Web sémantique. Dernier accès le: 02/06/2018; disponible sur: <https://jplu.developpez.com/tutoriels/web-semantique/introduction/>"21/04/2011"
- [19] LUONG, P.H., GESTION DE L'ÉVOLUTION D'UN WEB SÉMANTIQUE D'ENTREPRISE 14/11/2007 l'Ecole des Mines de Paris
- [20] EDDARIOUACHE, B., Les ontologies : concept de base et application. 2013-2014, Université Sidi Mohamed Ben Abdellah.
- [21] Gueffaz, M., ScaleSem : Model Checking et Web Sémantique. 11/11/2012, l'Université de Bourgogne.
- [22] BAHLOUL, D., Une approche hybride de gestion des connaissances basée sur les ontologies. 2006, L'Institut National des Sciences Appliquées de Lyon.
- [23] AUDIBERT, L., UML 2 De l'apprentissage à la pratique. 2006.
- [24] Giudicelli, V., Ontologies et l'éditeur Protégé 5 mai 2010.
- [25] LACHEVRE, P., Site d'aide à la compréhension de l'anglais (SACA).
- [26] StarUML. Dernier accès le: 02/06/2018; disponible sur: <https://air.imag.fr/index.php/StarUML>.