République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida

Faculté des sciences

Département d'informatique

Mémoire Présenté par :

KHIMOUD Salem TOURI Anis

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique

Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Thème

Mise en œuvre et sécurisation d'une application de gestion de la Formation Professionnelle Spécialisée (FPS)

Mme. ARKAM	Meriem	Présidente Jury
Mme Hadj henni	M.	Jury
Mlle. BOUSTIA	Narhimene	Promotrice
Mme. BENSTITI	Imene	Encadrante

Promotion
2015 / 2016

RÉSUMÉ

Les entreprises sont souvent exposées aux attaques qui elles même pourraient conduire à des catastrophes diverses, et pour prévenir contre ça, une politique de sécurité est nécessaire pour protéger leur système informatique et en sachant que le domaine de la sécurité étant vaste, nous nous sommes tout particulièrement intéressés au contrôle d'accès aux données.

Dans ce mémoire, nous proposons une politique de contrôle d'accès basée sur le modèle OrBAC, ainsi que le chiffrement des données échangées, et cela en utilisant le chiffrement RSA. L'implémentation de notre modèle avec la plate-forme Java a permis d'obtenir des résultats probants.

Mot clés : Contrôle d'accès, Sécurité, OrBAC, Confidentialité, Application web, Intégrité

ABSTRACT

Companies are exposed to several attacks that leads them selves to various disasters. In order to avoid these laters; a security policy is necessary to protect thier computer's systems.

knowing that security feild is so large, we have particulary interested in controlling access Data. In our research, we propose an access control policy based on the model OrBAC, and encryption of exchanged Data, using RSA encryption. The occurenrce of our Model with the java platform allows significant results.

Key words : Access control, security OrBAC, Privacy, Web application, and integrity.

ملخص

تتعرض الكثير من الشركات يوميا إلى هجمات التي يمكن أن تؤدي إلى كوارث مختلفة منها تسريب تغيير أو حتى حذف البيانات ، لمنع هذه الظاهرة يجب انتهاج سياسة أمنية تعمل على حماية الأنظمة الخاصة بهم .مجال أمن تكنولوجيا المعلومات واسع و فيه عدة تخصصات، لذلك نحن مهتمون و بشكل خاص التحكم في الدخول إلى البيانات و منح للمستخدم الإذن بالدخول لتغيير حذف أو إضافة وهذا متعلق بالدور الذي يقوم به في الشركة . في هذه الذاكرة اقترحنا سياسة أمنية معتمدين على نموذج التحكم في الدخول استنادا للمنظمة وتشفير البيانات المتبادلة، وذلك باستخدام الشيفرة المبتكرة من طرف العلماء رونالد ريفست عدي شامير و ليونارد أدلمان تم تطبيق نموذجنا على منصة الجافا وكانت النتائج إيجابية.

الكلمات المفتاحية: البيانات، امن، التحكم في الدخول، التحكم في الدخول استنادا للمنظمة ،تشفير البيانات، الشيفرة.

REMERCIEMENTS

Au terme de ce modeste travail, nous remercions, en premier lieu, Dieu de nous avoir donné la force et le courage de le mener à terme.

Nos vifs remerciements vont à notre promotrice Dr N.BOUSTIA pour l'intérêt qu'elle a porté a notre recherche en acceptant d'examiner notre travail et de l'enrichir par ses propositions.

Nos vifs remerciements vont également à nos encadreurs Mme I.BENSTITI, et Mme F.HEDADDA qui nous ont suivis durant toute la période du travail et nous ont aidés avec ses précieux et valeureux conseils.

Nous remercions particulièrement le chef du département d'informatique Mme S. BENSTITI pour sa grande disponibilité et son encadrement durant notre cursus.

Nous tenons à remercier chaleureusement tous ce qui ont contribués de prêt ou de loin à la réalisation de ce mémoire de fin d'études qui fut difficile mais très bénéfique à tout point de vue.

DÉDICACE

Je dédie mon travail

À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance.

J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

À mon frère, ma sœur et a toute ma famille.

À tous mes ami(e)s et toute personne faisant partie de mon entourage.

anis

DÉDICACE

Je dédie mon travail à

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie, reçois à travers ce travail aussi modeste soit-il, l'expression de mes sentiments et de mon éternelle gratitude.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit ; Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

Mes frères et sœurs qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité.

À tous mes amis et collègues de la promotion.

Salem

LISTE DES ABRÉVIATIONS

Mot	Signification
JAVA EE	JAVA Entreprise Edition
DAC	Dynamic Access Control
MAC	Mandatory Access Control
TMAC	Team Access Control
RBAC	Role Based Access Control
OrBAC	Organization Based Access Control
RSA	Rivest Shamir Adleman
PGP	Pretty Good Privacy
DSA	Digital Signature Algorithm
MIT	Massachusetts institute of Technology
AES	Advanced Encryption Standard
HTTPS	Hypertext Transport Protocol Secure
SSL	Secure Sockets Layers
UML	Unified Modeling Language
SQL	Structured Query Language
XML	eXtensible Markup Language
MVC	Modèle Vue Contrôleur
SGBD	Système de Gestion de Base de Données
MCD	Modèle Conceptuel des données
ITSEC	Information Technology Security Evaluation Criteria
HTML	HyperText Mark-Up Language
CSS	Cascading Style Sheet
JSP	Java Server Pages
SAF	Subdivision Action Formation
RAF	Résponsable Action Formation
IDS	Intrusion Détection System

TABLE DES MATIÈRES

LISTE DES TABLEAUX

TABLE DES FIGURES

INTRODUCTION GÉNÉRALE

Les établissements d'aujourd'hui sont plus conscients de l'impact de la gestion efficiente des ressources internes sur l'amélioration de ses performances et par la suite de ses compétitivités. Or, la tâche de gérer s'avère de plus en plus difficile et complexe. En effet, la croissance des activités engendre un énorme flux de données, alors que la diversité des processus fonctionnels nécessite une gestion adéquate des compétences humaine. Pour surpasser ces difficultés, tout établissement est prêt à investir dans l'implantation de technologies logicielles afin d'améliorer ses services et d'accroître son agilité vis-à-vis des nouvelles technologies, tout en optimisant la communication au sein de ses équipes.

Pour l'Ecole Technique de Blida (ETB), la gestion de l'ensemble de ses activités se fait par le biais des applications Informatiques. L'école souhaite gérer ses activités via un système d'information plus sophistiqué, à l'aide d'une application web sécurisée.

C'est dans ce cadre que s'inscrit notre projet de fin d'études qui a pour objectif de mettre en place une solution pour la sécurisation de gestion pour les actions de formation et les stagiaires. Pour cela, l'étude de l'existant a permis de connaitre les modules à mettre en place dans l'application web à savoir la gestion des formations, la gestion des stagiaires, ainsi que le volet le plus important qui est de mettre en œuvre des mécanismes de sécurité adéquats pour sécuriser notre système.

Ce présent rapport se compose de cinq chapitres. Dans le premier chapitre, nous présentons l'organisation de l'école technique de Blida. Le deuxième décrit une étude du système existant au sein de l'école. Les mécanismes des politiques de sécurité sont détaillés dans le troisième chapitre. Le quatrième chapitre, quant à lui, englobe l'analyse et conception détaillée du système à développer. Le dernier chapitre est consacré à la réalisation de l'application sécurisée. Enfin, une conclusion pour dresser le bilan de ce travail.

CHAPITRE 1

L'ÉTUDE DE L'EXISTANT

1.1 Introduction :

Dans ce chapitre, nous présenterons le groupe SONELGAZ et son organisation pour bien connaitre notre champ d'étude ainsi qu'une étude de l'existant pour prendre connaissance en détail des objectifs poursuivis.

1.2 Présentation de l'organisme d'accueil :

1.2.1 Présentation :

Société Nationale de l'Electricité et du Gaz, abrégé SONELGAZ, est une société nationale publique sous tutelle du ministère de l'énergie et des mines, chargée de la production, du transport et de la distribution de l'électricité et du gaz en Algérie.

1.2.2 Historique :

Elle a été créée en 1969, en remplacement de l'entité précédente Électricité et gaz d'Algérie (EGA), et on lui a donné un monopole de la distribution et de la vente de gaz naturel dans le pays, de même pour la production, la distribution, l'importation, et l'exportation d'électricité.

En 2002, le décret présidentiel N° 02-195, la convertit en une Société par actions SPA entièrement détenue par l'État.

En 2010, on parle de Groupe Sonelgaz.

1.2.3 Présentation de l'école de Blida :

L'Ecole Technique de Blida (ETB), d'une superficie de 13 hectares est située au centre de la ville de Blida. Elle dispose d'un cadre d'apprentissage adapté à sa mission grâce à :

- Ses infrastructures et aires d'entrainement pédagogiques équipées d'installations électriques et gazières, conformes à celles de l'exploitation.
- La disponibilité de conditions de prises en charge complète pour les apprenants.
- Son accessibilité.

Créée en 1949 par EGA (électricité et gaz d'Algérie), ETB est aujourd'hui une Ecole de renom. Elle a conservé sa mission première : former dans les métiers de l'électricité et du gaz.

1.2.4 Son organisation :

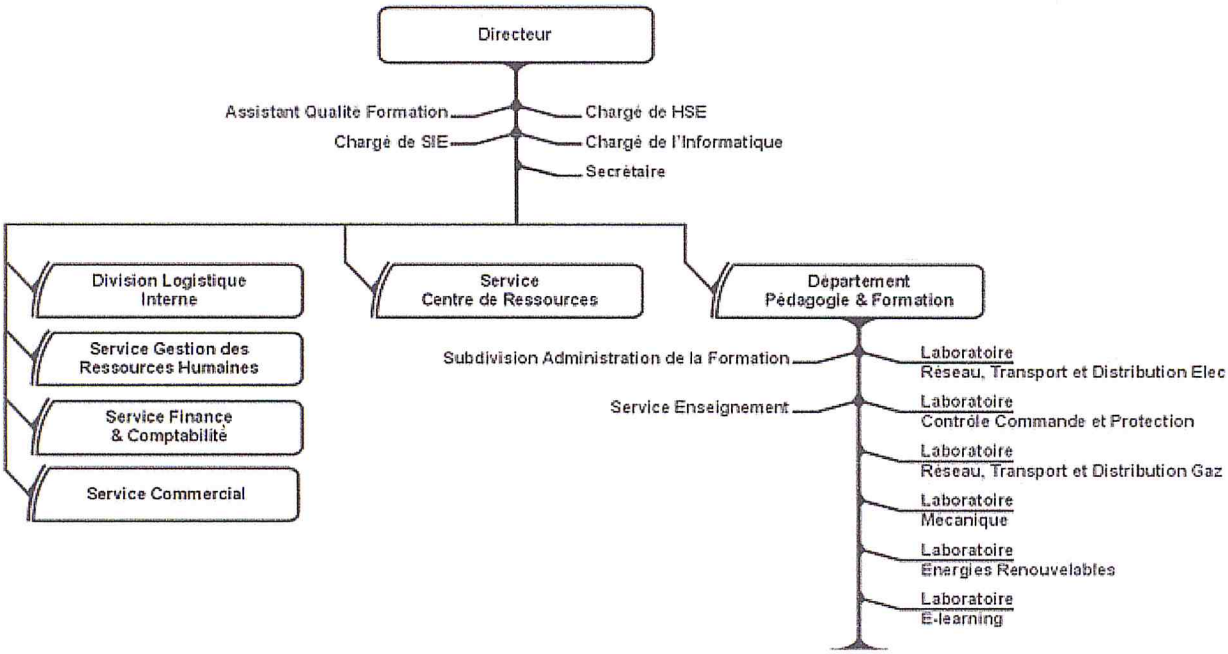


FIGURE 1.1 – Organisation Ecole Technique de Blida.

1.2.5 Le laboratoire e-learning :

Chargé de développer les plateformes de formation à distance dans des domaines cibles. A moyen terme ce laboratoire aura à être intégré dans les autres laboratoires.

Le rôle du laboratoire consiste à :

- Accompagner les laboratoires métier dans le choix des modules à diffuser en ligne.
- Encadrer l'auteur du module durant les étapes de scénarisation des supports pédagogiques de l'œuvre.

1.3 L'étude de l'existant :

1.3.1 L'existant au niveau de l'application :

La société utilise une application développée en langage Delphi, le problème est que celle-ci doit être déployée sur les PC des utilisateurs ce qui implique un grand travail lors de l'apparition d'une nouvelle version.

Voici quelques interfaces de l'application existante :

1. **Gestion des formations :**
   Dans cette partie, nous présentons les interfaces qui permettent de gérer les formations :

   1.1. Interface pour la création des formations : La figure ci-dessous représente l'interface pour créer les formations :

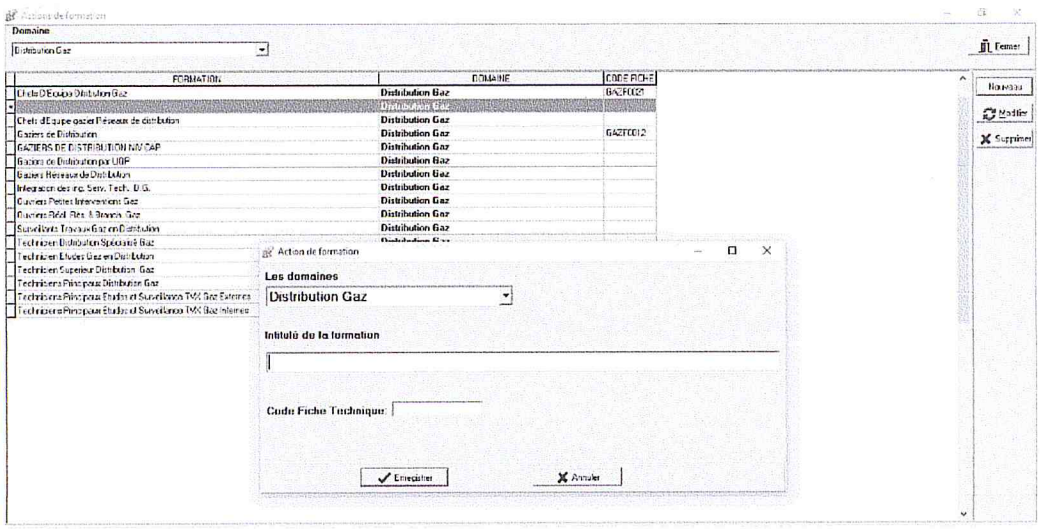


FIGURE 1.2 – Création des formations.

1.2 Consultation des formations : La figure ci-dessous représente l'interface pour consulter les formations

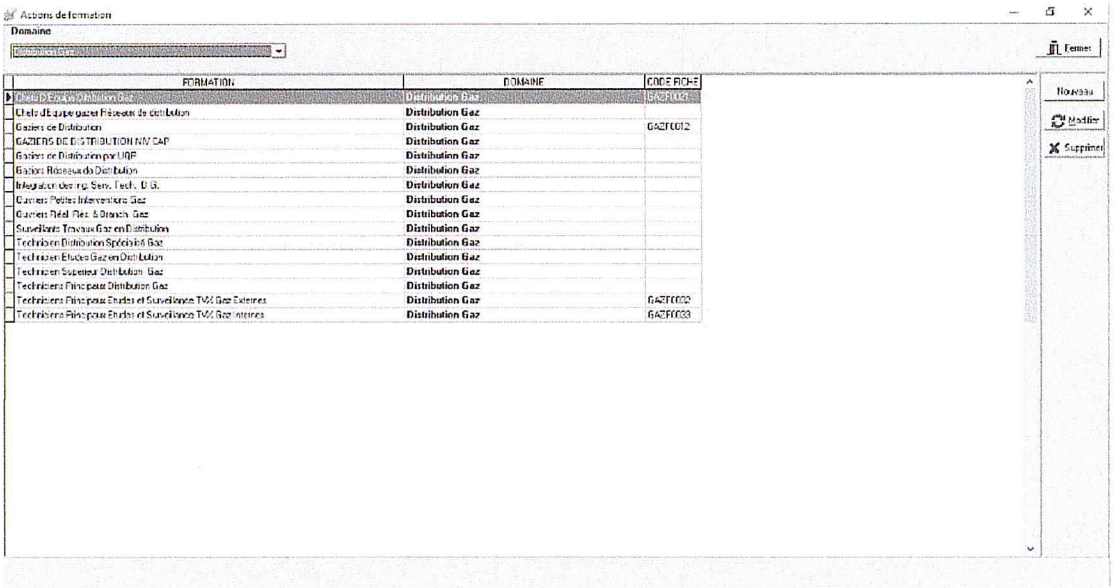


FIGURE 1.3 – Consultation des formations.

2. **Gestion des stagiaires :** Dans cette partie, nous présentons les interfaces qui gère les stagiaires provisoire et stagiaires admis :

2.1 Stagiaires prévisionnel : La figure ci-dessous représente l'interface pour inscrire les stagiaires avant qu'ils viennent faire leur formation à l'école :

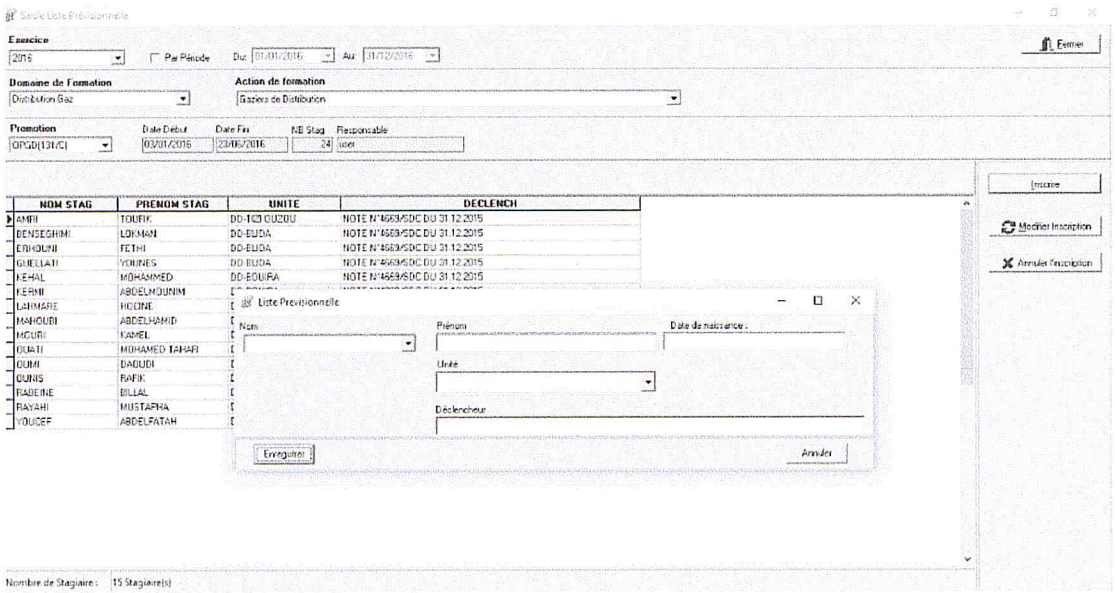


FIGURE 1.4 – Inscription des stagiaires prévisionnel.

2.2 Stagiaires admis : La figure suivante représente l'interface qui est utilisée pour inscrire les stagiaires qui sont admis et qui vont commencer leur formation :

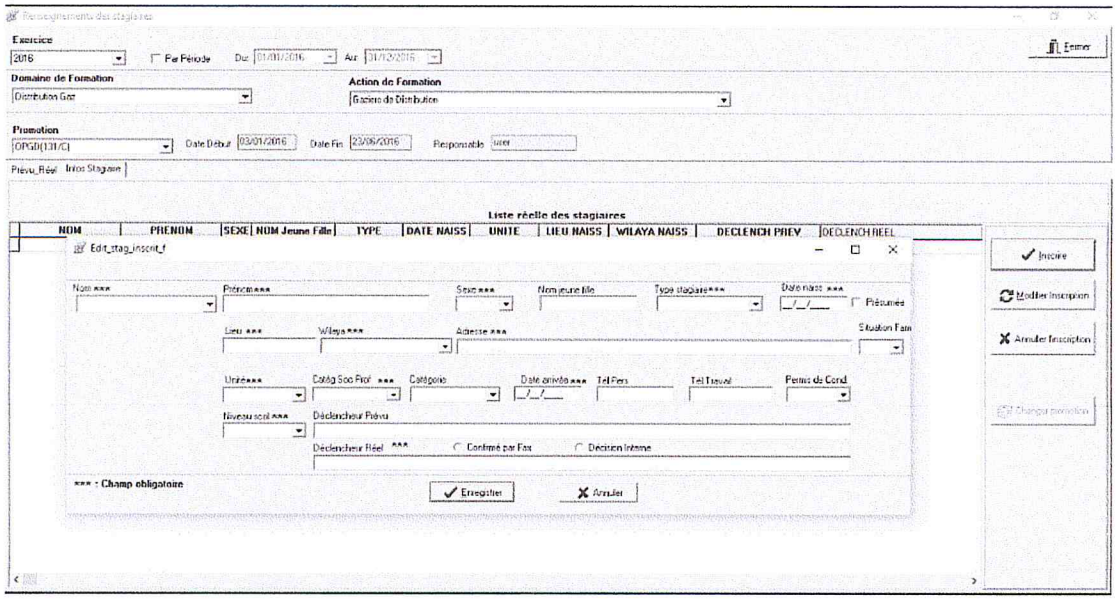


FIGURE 1.5 – Inscription des stagiaires réels.

1.3.2 L'existant au niveau de la base de donnée :

La société possède une base de données avec un SGBD oracle 9i sous License, cette base de données doit être modifiée pour qu'elle réponde à notre besoin dans la partie sécurité.

Le MCD de la base de données existante contient des redondances donc il faut le corriger. La figure 2.5 représente le MCD existant :

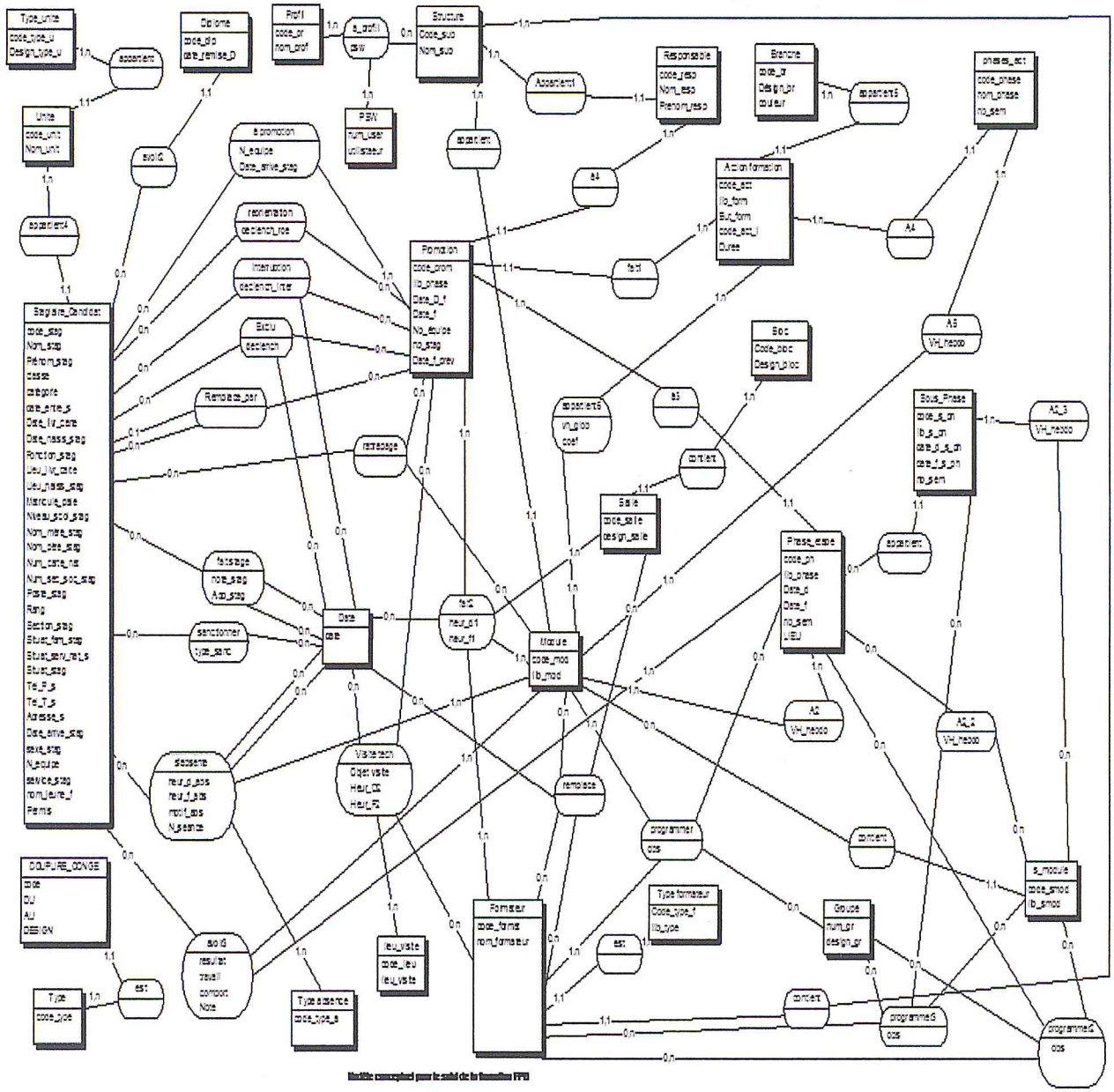


FIGURE 1.6 – MCD existant.

1.3.3 L'existant au niveau de la sécurité :

Dans cette partie, la société connait un grand manque dans la sécurisation du trafic de communication et de la base de donnée, elle utilise seulement un algorithme pour chiffrer les mots de passe des utilisateurs dans la base de donnée, ce qui est très insuffisant pour répondre a la sécurité de l'entreprise. La figure 2.6 représente l'algorithme utilisé :

```
If (:new.psw is not null) then  --and (:new.psw<>:old.psw)  then
 : new.psw:=omar.GET_HASH_VAL(:new.psw);
 end if;
```

Et la fonction «GET_HASH_VAL » est défini comme suite :

```
Create or replace  FUNCTION      "GET_HASH_VAL" (p_in VARCHAR2)
    RETURN VARCHAR2
  IS
    l_hash   VARCHAR2 (300);
  BEGIN
    l_hash :=RAWTOHEX(UTL_RAW.cast_to_raw (DBMS_OBFUSCATION_TOOLKIT.md5
(input_string=> p_in)));
    RETURN l_hash;
  END;
```

FIGURE 1.7 – L'algorithme de chiffrement de mot de passe.

1.4 Problématique :

Les paramètres d'authentification, d'habilitation et de contrôle d'accès soumis (par l'utilisateur, ou d'application à application) sont bien souvent incorrectement pris en compte, gérés ou contrôlés. Cette situation peut entraîner des risques d'usurpation d'identité et d'accès à des fonctionnalités ou données illégitimes, et donc d'atteinte à la confidentialité.

Le risque est exacerbé par le fait que le fonctionnement par défaut d'un serveur Web est d'autoriser l'accès au contenu publié. Si un contenu spécifique doit être fonctionnellement restreint à certains utilisateurs seulement, le développeur doit mettre explicitement en place des mécanismes d'authentification, d'habilitation et de contrôle d'accès pour protéger chaque contenu (pages Web notamment) devant être protégé.

1.5 Objectif :

L'objectif de ce projet est de munir une application web de mécanismes de sécurité dans le but d'assurer la confidentialité et l'intégrité des données échangées.

Pour cela, nous allons développer une application web pour la gestion des FPS pour l'école sonelgaz, fonctionnant de la manière suivante :

-Authentification et déconnexion.
-Gestion des actions de formation.
-Gestion des stagiaires.

Tous cela dans un environnement sécurisé. Pour se faire on doit passer par les étapes les plus importantes qui sont :

-Défenir une politique de sécurité en se basant sur le modèle Orbac(Organization Access Based Control)afin de définir les privilège de chacun.
-La sécurisation du trafic de communication « de navigateur a serveur » ou bien « de serveur a navigateur ».
-La sécurisation de la base de données qui se fait par l'authentification, la protection des échanges, la sécurisation des données.
-Assurer le chiffrement en utilisant la méthode RSA.

1.6 Conclusion :

Cette phase de recherche nous a permis de nous familiariser avec notre projet, bien cadrer et cibler nos objectifs et adopter une méthode de travail. Dans le chapitre suivant nous expliquerons les modèles de contrôle d'accée et les applications web .

CHAPITRE 2

SÉCURITÉ DES DONNÉES ET APPLICATION WEB

2.1 Introduction :

L'une des exigences majeures du partage des données entre plusieurs utilisateurs est la protection de ces données contre des atteintes à la confidentialité (divulgations d'information non autorisées), contre des atteintes à l'intégrité (modifications non autorisées) et contre des atteintes à la disponibilité (déni de service). Afin d'assurer cette protection, chaque accès aux données doit être contrôlé et bien évidemment tous les accès non autorisés doivent être impérativement bloqués. Cela est appelé le contrôle d'accès. Le développement d'un modèle de contrôle d'accès repose sur la définition de politiques de contrôle d'accès qui déterminent qui a le droit d'effectuer quelle action sur quelle donnée. Le modèle veille à ce que les données ne soient accessibles que par des utilisateurs ayant le droit d'y accéder.

2.2 Sécurité des données :

2.2.1 Définition :

ITSEC (Information Technology Security Evaluation Criteria) [3] définit la sécurité des données comme étant la combinaison de trois propriétés :

- la confidentialité des données,
- l'intégrité des données,
- la disponibilité du système.

On entend par :

- **Confidentialité :** "empêcher une divulgation non autorisée de l'information". En d'autres termes, cela consiste à protéger les informations sensibles contre les accès des utilisateurs non autorisés. Par exemple, dans le domaine médical "cacher le diagnostic des patients pour les secrétaires médicales".

- **Intégrité :** " empêcher une modification non autorisée", c'est-à-dire empêcher toute modification (suppression, ajout, mise à jour) d'une donnée par un utilisateur non légitime. Par exemple, dans le domaine médical, toute modification, malintentionnée ou pas, d'un diagnostic d'un patient peut mettre sa santé, voire sa vie, en danger.

- **Disponibilité :** "empêcher un déni non autorisé d'accès à l'information ou à des ressources". En d'autres termes, garantir de rendre une donnée accessible lorsqu'un utilisateur autorisé en a besoin.

2.3 Modèles de contrôle d'accès :

Avant de présenter ces différents modèles, nous allons définir, dans cette section, ce qu'est une politique de contrôle d'accès.

Les politiques de contrôle d'accès sont définies comme étant des directives (règles) de haut niveau [1] qui spécifient qui a la permission d'exercer quoi sur quelle donnée. A partir de cette définition, nous dégageons trois concepts fondamentaux d'une politique de contrôle d'accès qui sont :

- **Sujet :** entité active qui accède aux données du système. Le sujet peut être un utilisateur, une application, une adresse IP ...
- **Objet :** entité passive qui représente les données à protéger. L'objet peut être, par exemple, un fichier, une table relationnelle, une classe ...
- **Action :** représente l'action à traiter par le sujet sur l'objet. L'action peut être lire, écrire, exécuter...

2.3.1 Modèle de contrôle d'accès discrétionnaire (DAC)

DAC (Discretionary Access Control) a été proposé par Lampson [2] et popularisé par le système d'exploitation UNIX. Dans ce modèle ce sont les utilisateurs qui attribuent les permissions sur les ressources qu'ils possèdent. Ils définissent librement les droits d'accès pour eux, le groupe et les autres utilisateurs. Ce type de mécanisme est utilisé principalement dans les systèmes d'exploitation modernes. Les permissions sont représentées par une matrice, dans laquelle chaque ligne correspond à un utilisateur et chaque colonne à une ressource. Le contenu de chaque élément de cette matrice définit les droits d'accès (lecture, écriture et exécution) pour l'utilisateur sur la ressource. La mise en œuvre de ce modèle est coûteuse en mémoire lorsque le nombre des utilisateurs est important. Alors le regroupement des utilisateurs peut être envisagé [6].

	Objet 1	Objet 2	...	Objet n
Sujet 1	*Lire*			
Sujet 2		*Ecrire*		*Lire*
...				
Sujet n			*Exécution*	

FIGURE 2.1 – Matrice de permissions DAC[10].

2.3.1.1 Les limites du modèle de contrôle d'accès discrétionnaire :

Le modèle de contrôle d'accès discrétionnaire limite l'accès aux objets uniquement en se basant sur l'identité de l'utilisateur. Pour cela, le modèle DAC est appelé également, IBAC (Identity Based Access Control) [4]. Ce principe de base rend le contrôle d'accès vulnérable aux chevaux de Troie. Afin de comprendre comment le cheval de Troie peut amener à une fuite d'information vers des utilisateurs non autorisés, nous prenons un exemple pour illustrer le problème (Figure 3.2).

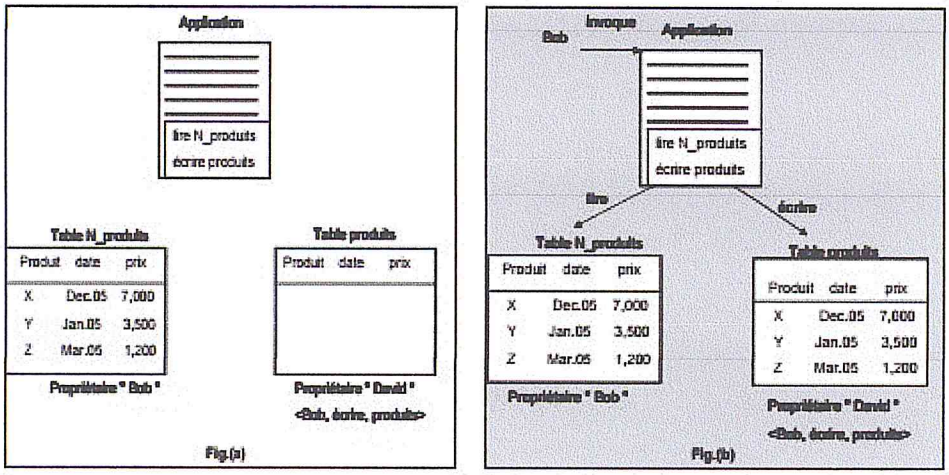


FIGURE 2.2 – Exemple chevaux de Trois[9].

Supposons dans une organisation, Bob, directeur, crée un fichier nouveaux_produit (Nproduits) contenant des informations très sensibles sur les nouveaux produits. Ces informations sensibles, d'après la politique de l'organisation, ne devraient être accessibles que par Bob. Supposons maintenant qu'un utilisateur malveillant David, un adjoint de Bob, veuille récupérer cette information sensible pour la vendre à une organisation concurrente. Pour cela, David crée un fichier Produits et donne l'autorisation à Bob d'écrire dans ce fichier. David, ensuite, introduit deux opérations cachées dans l'application utilisée par Bob. Ces opérations sont lire dans le fichier N-produits et écrire dans le fichier produits. Une fois que Bob exécute l'application, les opérations lire et écrire vont être permises. Puisque, l'utilisateur malveillant David est le propriétaire du fichier produit il pourra accéder à ce fichier et récupérer les informations désirées.

Nous constatons que malgré le fait que nous fassions confiance aux utilisateurs pour qu'ils obéissent aux politiques de l'organisation nous ne pouvons pas faire confiance aux processus qui s'exécutent pour leur compte d'où la nécessité de distinguer entre les utilisateurs et les processus qui s'exécutent pour leurs comptes (sujets).

Dans la section suivante, nous montrons comment les modèles MAC (plus précisément le modèle multi-niveaux) font la distinction entre sujets et objets pour résoudre les problèmes des chevaux de Troie et de fuite d'information.

2.3.2 Le contrôle d'accès mandataire (MAC)

Le concept MAC (Mandatory Access Control) a été introduit par Bell et LaPadula [7], il est utilisé principalement dans les environnements militaires à cause de son contrôle centralisé et il permet à l'administrateur du système de définir des privilèges pour protéger la confidentialité et l'intégrité des ressources dans le système. Le contrôle d'accès est dit obligatoire lorsque l'accès aux objets est basé sur le niveau de sensibilité de l'information (Figure 3.3) contenue dans les objets. L'autorisation d'accéder à un objet est accordée à un sujet si le niveau d'autorisation de ce sujet est en accord avec le niveau de sensibilité de l'information [8]. Le modèle MAC attribue un niveau de sécurité à chaque utilisateur et à chaque ressource. On accorde l'accès à un utilisateur seulement si son niveau de sécurité est supérieur ou égal au niveau de la ressource à laquelle il veut accéder. Les règles diffèrent selon qu'ils s'agissent de maintenir des propriétés de confidentialité ou d'intégrité. Les politiques obligatoires les plus fréquemment utilisées sont les politiques multi-niveaux. Ces politiques reposent sur des classes de sécurité affectées aux informations et des niveaux des habilitations affectées aux utilisateurs [8]. Un système supportant une politique de contrôle d'accès obligatoire peut être utilisé aisément dans une administration centralisée [8]. Par contre, il n'est pas recommandé pour les environnements distribués, les utilisateurs n'ont pas suffisamment de privilèges.

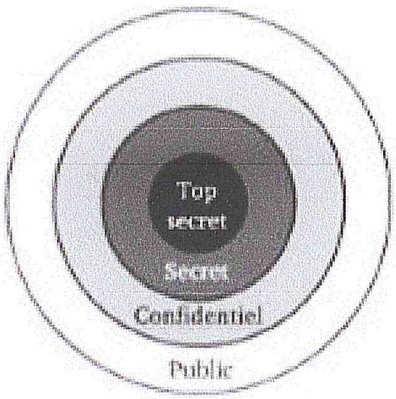


FIGURE 2.3 – Les niveaux de sensibilité dans le modèle MAC [10].

2.3.3 Modèle de contrôle d'accès à base de rôle (RBAC)

La motivation principale autour du contrôle d'accès à base de rôle RBAC (Role Based Access Control) est de faciliter l'administration de la politique de contrôle d'accès et de proposer un modèle de contrôle d'accès qui reflète la structure organisationnelle de l'entreprise [11]. Le cœur de RBAC est le rôle. Ce dernier représente d'une façon abstraite

une fonction particulière dans une organisation (par exemple, médecin, infirmière, statisticien...). Le rôle est une entité intermédiaire entre les permissions d'accès, appelées aussi privilège ou droit d'accès, et les utilisateurs. Il regroupe un ensemble de privilèges qui va être ensuite attribué aux utilisateurs en fonction de leurs positions organisationnelles. Par conséquent, contrairement au contrôle d'accès discrétionnaire, l'utilisateur ne reçoit pas directement ses permissions d'accès, mais les reçoit via des rôles. La Figure 3.4 montre l'attribution des opérations aux utilisateurs à travers les rôles.

FIGURE 2.4 – Attribution des permissions en RBAC [9].

Un rôle peut avoir plusieurs permissions et une permission peut être associée à plusieurs rôles. Un utilisateur peut jouer plusieurs rôles et un rôle peut être attribué à plusieurs utilisateurs. A titre d'exemple, dans le domaine médical, le médecin chirurgien est à la fois chirurgien et directeur de l'hôpital. Dans ce cas, il pourra accéder aux dossiers médicaux des patients en jouant le rôle chirurgien et aux dossiers administratifs de l'hôpital en jouant le rôle directeur de l'hôpital. Un utilisateur établit une session durant laquelle il active un sous-ensemble de ses rôles. Dans une session, plusieurs rôles peuvent être activés (à la discrétion de l'utilisateur) et chaque session est associée à un seul utilisateur [5].

2.3.4 Modèle de contrôle d'accès à base d'équipe (T-MAC) :

Le modèle T-MAC introduit la notion d'équipe. Dans T-MAC, des autorisations sont associées aux rôles ainsi qu'aux équipes. Les autorisations que possède un sujet résultent de la combinaison des autorisations associées aux rôles exercés par le sujet et des autorisations associées à l'équipe à laquelle est affecté le sujet. Plusieurs combinaisons (par exemple, l'union des autorisations) sont envisagées. En fait, le modèle T-MAC introduit deux relations binaires : rôle-autorisation et équipe-autorisation. Si l'on introduit la notion d'équipe, il est en fait nécessaire de considérer une relation ternaire équipe-rôle-autorisation pour spécifier que les autorisations dépendent non seulement du rôle mais aussi de l'équipe dans laquelle est exercé ce rôle. A l'aide d'une telle relation ternaire, on pourra ainsi facilement spécifier que les autorisations du rôle médecin changent suivant qu'il s'agit d'un médecin dans une équipe de garde ou d'un médecin dans une équipe d'urgence[12].

2.3.5 Modèle de contrôle d'accès Or-BAC :

Le contrôle d'accès basé sur l'organisation OrBac (organization-Based Access Control) a été présenté pour la première fois en 2003[8].Il reprend les principes de rôles des modèles du type RBAC, en offrant en plus, la possibilité de modifier la politique de sécurité de façon dynamique en fonction d'un contexte.

Dans OrBac, la possibilité d'exprimer des permissions, des obligations et des interdictions qui dépendent de contextes, est un élément qui va vers une plus grande expressivité. L'abstraction des entités traditionnelles du contrôle d'accès (sujet, action, objet) en méta entités (rôle, activité, vue) permet d'élaborer une politique de sécurité a deux niveaux, un niveau concret et un niveau abstrait.

L'introduction d'un niveau abstrait organisationnel permet aussi la structuration des entités comme on le voit sur la figue 3.5 nous obtenons une politique de sécurité a deux niveaux, le modèle OrBac permet ainsi d'établir une politique de sécurité abstraite (rôle, activité, vue) indépendante des choix d'implémentation (sujet, action, objet).

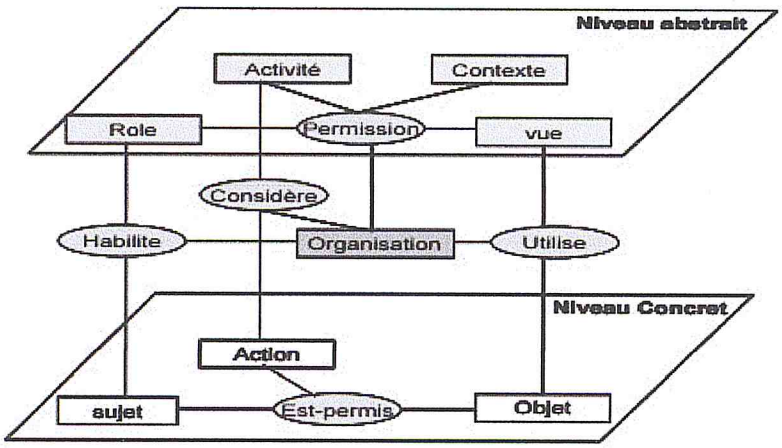


FIGURE 2.5 – Modèle OrBac [9].

2.3.5.1 La notion de contexte :

On voit apparaître sur ce schéma des interactions une entité qui n'apparait pas dans les autres modèles de contrôle d'accès : le contexte. Celui-ci est défini pour une organisation, un sujet, une action, des objets donnés. Les contextes permettent d'exprimer des permissions ou des interdictions dans certaines circonstances (urgence à l'hôpital, heures de travail dans une entreprise,...). Il est facile d'imaginer que dans un contexte d'urgence, on désirera qu'un infirmier puisse accéder au dossier d'un patient sans avoir besoin d'appeler l'administrateur afin que celui-ci lui donne les droits (peut-être trop tard). Cette possibilité de nuancer les autorisations n'est pas offerte par les autres modèles, alors que

dans de nombreuses organisations (hôpital, entreprise,...) il existe un réel besoin de ne donner des droits que dans des circonstances précises.

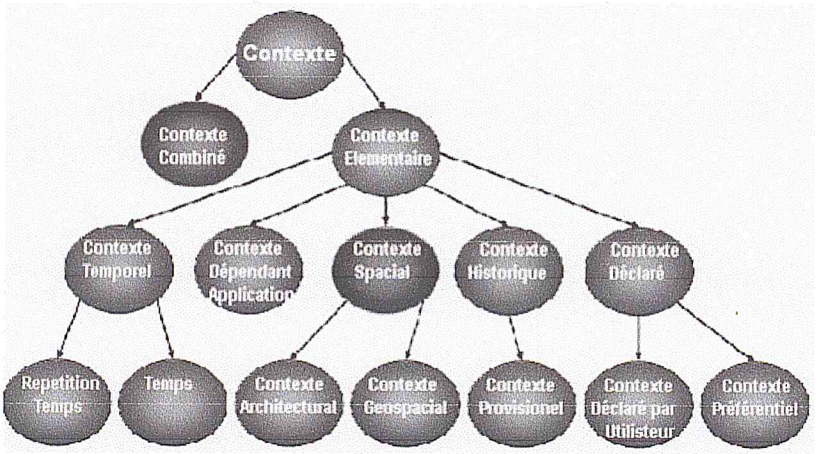


FIGURE 2.6 – Les différents contextes par type [12].

Pour le modèle Or-BAC, les contextes sont regroupés par type (comme sur le schéma ci-dessus) :

- **Contexte temporel :** ce sont des contextes régissant la durée de validité des privilèges ;

- **Contexte spatial :** il peut être lié à l'appartenance à un réseau, ou la position géographique, ou à toute autre situation spatiale ;

- **Contexte déclaré par l'utilisateur :** ce type de contexte est activé, par exemple, par le médecin en cas d'urgence, ou pour signaler que l'on effectue un audit. Dans ces cas exceptionnels, des permissions peuvent être données alors qu'elles seraient interdites dans un cas normal. L'utilisateur qui déclare le contexte est obligé en contrepartie de faire un compte-rendu des opérations effectuées et peut être des raisons qui l'ont motivé à déclarer ce contexte ;

- **Contexte prérequis :** leur utilisation permet de contraindre les sujets concernés par les permissions ou les interdictions dépendant de ces contextes et qui vient réduire ou étendre les droits d'accès hérités du rôle associé ;

-**Contexte provisionnel :** ce contexte permet de donner des privilèges en fonction de l'historique. Par exemple, le contexte "accès limités à 2 fois" regarde si le document, objet de l'action, a été accédé au moins 2 fois.

A noter que dans une modélisation Or-BAC, on définit toujours un contexte par défaut.

2.3.5.2 La notion de hiérarchie :

Afin de gérer plus facilement des sous-organisations, en automatisant la dérivation des permissions, Or-BAC permet de définir des hiérarchies sur les rôles, les activités, les vues et les contextes. On a ainsi l'héritage des permissions et des interdictions en descendant dans la hiérarchie des rôles, des activités, des vues et des contextes. Tout comme dans R-BAC, l'héritage permet de simplifier la tâche de l'administrateur en automatisant partiellement l'attribution des privilèges. Comme dans R-BAC, il existe deux façons de définir la hiérarchie de l'héritage :

La première vision pour définir la hiérarchie est la hiérarchie organisationnelle. Le directeur est hiérarchiquement supérieur à un ingénieur. Dans certains cas, il peut donc hériter de toutes les permissions de ce rôle (pour vérifier le travail de celui-ci). On dit alors que R1 est senior de R2 et R2 est junior rôle de R1, si un utilisateur jouant le rôle R1 est supérieur hiérarchique de R2 ;

La deuxième vision est la hiérarchie obtenue par la relation de spécification/généralisation est définie telle que R1 est un senior rôle de R2 si chaque fois qu'un utilisateur joue le rôle de R1, elle joue le rôle de R2. Par exemple sur la hiérarchie présentée sur le schéma un peu plus en dessous, le chirurgien est aussi un médecin. Donc à chaque fois qu'un utilisateur est associé au rôle de chirurgien, il joue aussi le rôle de médecin. Le rôle chirurgien est un senior rôle de du rôle médecin. Un rôle R1 senior de R2 hérite donc les permissions affectées à R2.

Dans Or-BAC, ces deux hiérarchies réapparaissent mais les droits qui leur sont associés sont quelque peu modifiés. En effet, avec le modèle Or-BAC, on peut définir des permissions mais aussi des interdictions[25].

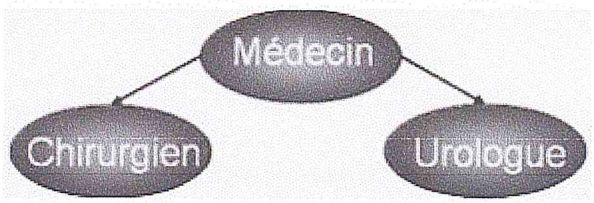


FIGURE 2.7 – Dérivation de privilèges par la hiérarchie dans Or-BAC.

2.3.5.3 La notion de délégation :

La délégation permet de donner à un utilisateur particulier un privilège, sans donner ce privilège à toutes les personnes ayant le même rôle que lui. La délégation, bien que

très utilisée, est très peu modélisée dans les politiques de sécurité car ce concept est très complexe. En effet, grâce à une délégation, une permission peut être donnée par le détenteur d'un droit à un tiers pour agir à sa place ou à la place d'un autre. On voit déjà ici apparaître qu'une délégation peut faire intervenir plusieurs parties :

- Le sujet qui possède le privilège ;
- Le sujet à qui on délègue le privilège ;
- Le sujet qui délègue le privilège (pour agir à sa place ou à la place d'un autre).

Il existe trois types de situation dans lesquelles la notion de délégation apparaît :

- La maintenance d'un rôle ;
- La décentralisation de l'autorité ;
- Le travail de collaboration.

La maintenance d'un rôle correspond au cas où un utilisateur doit déléguer une partie de ses permissions afin qu'on puisse remplir toutes ses obligations pendant son absence. La décentralisation de l'autorité est surtout utile dans le cas où on modifie une partie de l'organisation. En pratique, ce cas peut correspondre à l'ouverture d'un nouvel hôpital dans lequel on va transférer une partie des médecins exerçant dans les autres hôpitaux de la région. Le cas du travail en collaboration est évident, si on souhaite que notre partenaire puisse lire les documents que l'on possède sur un projet donné, il faut lui en donner l'autorisation.

Cependant, la délégation pose de nombreux problèmes. Entre autre, un utilisateur X ayant obtenu tous les droits d'un autre utilisateur Y peut ôter les droits à Y si X possède certains droits administratifs. Il peut aussi arriver que l'on oublie de révoquer une délégation faite à Z et qui n'a plus d'utilité d'être, ce qui peut laisser la possibilité à Z de se faire passer pour quelqu'un d'autre. C'est l'une des raisons pour lesquelles il est important de définir deux types de permission, celles qui sont délégables et celle qui ne peuvent l'être[25].

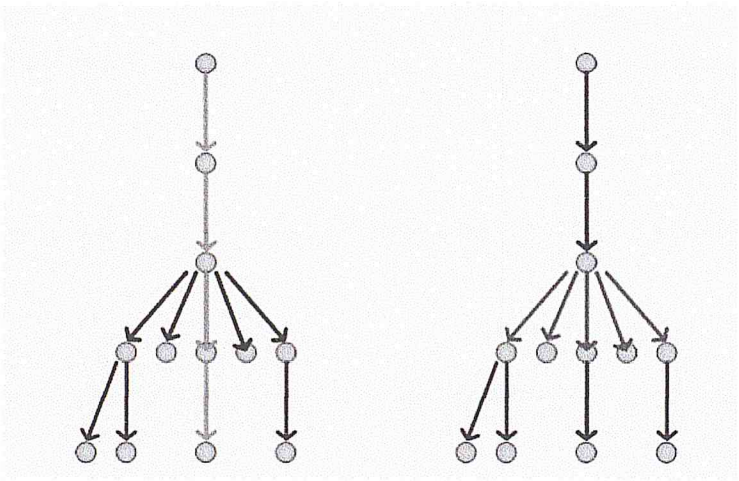


FIGURE 2.8 – La notion de délégation.

2.3.5.4 Les prédicats d'Or-BAC :

Afin de comprendre les règles définies dans Or-BAC, on récapitule dans ces tableaux les différents prédicats liés à Or-BAC.

- Les prédicats liés à l'affectation des entités abstraites aux organisations :

Nom de prédicat	Domaine	Description
Prédicat_Role	Org*Role	Si org est une organisation et r un rôle, alors Prédicat_Role signifie que le rôle r est défini dans l'organisation org
Prédicat_Activité	Org*Activité	Si org est une organisation et a une activité, alors Prédicat_Activité signifie que l'activité a est définie dans l'organisation org
Prédicat_Vue	Org*Vue	Si org est une organisation et v une vue, alors Prédicat_Vue signifie que la vue v est définie dans l'organisation org

TABLE 2.1 – Les prédicats liés à l'affectation des entités abstraites aux organisations.

- Les prédicats liés aux relations d'abstraction :

Nom de prédicat	Domaine	Description
Habilite	Org*Sujet*Role	Si org est une organisation, s un sujet et r un rôle, alors Habilite signifie que l'organisation org habilite le sujet s dans le rôle r
Considère	Org*Action*Activité	Si org est une organisation, A une action et a une activité, alors Considère signifie que l'organisation org considère l'action A comme faisant partie de l'activité a
Utilise	Org*Objet*Vue	Si org est une organisation, o un objet et v une vue, alors Utilise signifie que l'organisation org utilise l'objet o dans la vue v

TABLE 2.2 – Les prédicats liés aux relations d'abstraction.

-Les prédicats liés aux définitions des contextes :

Nom de prédicat	Domaine	Description
Définit	Org*Sujet*Action *Objet*Contexte	Si org est une organisation, s un sujet, A une action, o un objet et c un contexte, alors Définit signifie que dans l'organisation org, le contexte c est défini pour le sujet s, l'action A et l'objet o

TABLE 2.3 – Les prédicats liés aux définitions des contextes.

-Les prédicats liés aux permissions abstraites :

Nom de prédicat	Domaine	Description
Permission	Org*Role*Activité *Vue*Context	Si org est une organisation, r un rôle, a une activité, v une vue et c un contexte, alors Permission signifie que l'organisation org accorde la permission au rôle r de réaliser l'activité a sur la vue v dans le contexte c
Interdiction	Org*Role*Activité *Vue*Context	Si org est une organisation, r un rôle, a une activité, v une vue et c un contexte, alors Interdiction signifie que dans l'organisation org refuse la permission au rôle r de réaliser l'activité a sur la vue v dans le contexte c

TABLE 2.4 – Les prédicats liés aux permissions abstraites.

-Les prédicats liés aux permissions concrètes :

Nom de prédicat	Domaine	Description
Est_permi	Subjet*Action*Objet	Si s est un sujet, A une action et o un objet, alors Est_permi signifie que le sujet s a concrètement la permission de réaliser l'action A sur l'objet o
Est_interdit	Sujet*Action*Objet	Si s est un sujet, A une action et o un objet, alors Est_interdit signifie que le sujet s ne peut pas concrètement réaliser l'action A sur l'objet o

TABLE 2.5 – Les prédicats liés aux permissions concrètes.

2.3.6 Comparaison entre les modèles de contrôle d'accès :

Quel que soit le modèle implémenté au sein d'une organisation, l'objectif est de limiter la capacité d'action des entités utilisatrice sur les ressources au strict nécessaire pour réaliser leurs missions. Le contrôle d'accès discrétionnaire est généralement défini par opposition au contrôle d'accès obligatoire (MAC) [7] qui impose des règles incontournables garantissant l'atteinte des objectifs de sécurité visés. Dans ce type de contrôle d'accès les sujets ne peuvent pas intervenir dans l'attribution des droits d'accès.

Ce contrôle d'accès est plus rigide que le contrôle d'accès discrétionnaire mais plus sûr. MAC est donc un système supportant une politique de contrôle d'accès obligatoire, il peut être utilisé aisément dans une administration centralisée [8].

Par contre, il n'est pas recommandé pour les environnements distribués, car les utilisateurs n'ont pas suffisamment de privilèges pour gérer leurs propres besoins de confidentialité, notamment ceux concernant la vie privée.

Bien que les modèles DAC et MAC ont fait leurs preuves en matière de sécurité ils ont exprimé des limites à ce sujet et un manque de flexibilité, vu qu'ils ont une approche portée sur le sujet, c'est-à-dire qu'ils attribuent directement aux sujets, or si un sujet venait à disparaitre de l'organisation il faut refaire toute la politique pour un nouveau sujet.

Le modèle RBAC permet de diminuer la taille de liste des habilitations. Les contrôles d'accès sont réalisés sur les rôles attribués aux comptes. Les rôles applicatifs sont octroyés en fonction du profil métier. Cependant, le contrôle d'accès basé sur les rôles est insuffisant pour satisfaire tous nos besoins en matière de protection. L'un des problèmes majeurs de

ce modèle est le fait que tous les utilisateurs associent au même rôle possèdent forcément les mêmes privilèges. Ceci réduit la flexibilité des politiques de sécurité. En effet, l'expression des aspects contextuels liées aux autorisations d'accès n'est pas présente dans le modèle RBAC, on constate une confusion entre rôle et organisation, On peut seulement exprimer les permissions (pas d'interdiction), ce qui entraine une gestion complexe des exceptions.

Parmi les modèles de contrôle d'accès, le modèle Orbac a été choisi pour notre application.Pour cela nous allons définir l'application web ainsi que son fonctionnement avec le serveur d'application.

2.4 Application Web :

Le Web, C'est le service de consultation de documents sur Internet, le plus connu, le plus récent et aujourd'hui le plus utilisé. C'est le service d'Internet qui a contribué le plus à sa popularité.

Le principe du Web repose sur l'utilisation d'hyperliens pour naviguer entre des documents (appelés «pages Web») grâce à un logiciel appelé. Une page Web est ainsi un simple fichier texte écrit dans un langage de description (appelé HTML), permettant de décrire la mise en page du document et d'inclure des éléments graphiques ou bien des liens vers d'autres documents à l'aide de balises [15].

2.4.1 Définition :

Un site Web est un ensemble constitué de pages Web (elles-mêmes faites de fichiers HTML, CSS, JavaScript, etc.). Lorsqu'on développe puis publie un site Web, on met en réalité en ligne du contenu sur internet. On distingue deux types de sites :

2.4.1.1 Les sites statiques :

Ce sont des sites dont le contenu est « fixe », il n'est modifiable que par le propriétaire du site. Ils sont réalisés à l'aide des technologies HTML, CSS et JavaScript uniquement.

2.4.1.2 Les sites dynamiques :

Ce sont des sites dont le contenu est « dynamique », Le principe d'un page dynamique est d'être construit à la demande par le serveur (côté serveur), en fonction de critères spécifiques. La présentation et le contenu affichés peuvent ainsi être personnalisés de manière interactive, en fonction des produits, des internautes, des langues, etc.

Pour le fonctionnement de l'application web, un serveur d'application est nécessaire pour son exécution .

2.4.2 Serveur d'application :

Un serveur d'application est un environnement informatique qui fournit les briques nécessaires à l'exécution d'applications sur le Web.
Il doit répondre aux critères techniques suivants :

-S'interfacer avec un serveur http (HTML, XML, WML pour WAP) ;
-Fournir un moteur d'exécution des traitements (ex : Java Virtual Machine) ;
-S'ouvrir sur le système d'information de l'entreprise (XML, Web services, connecteurs SGBDR, ...)[16].

2.4.3 Fonctionnement d'un serveur d'application :

Pour comprendre le fonctionnement du serveur d'application, on va faire appel au schéma suivant :

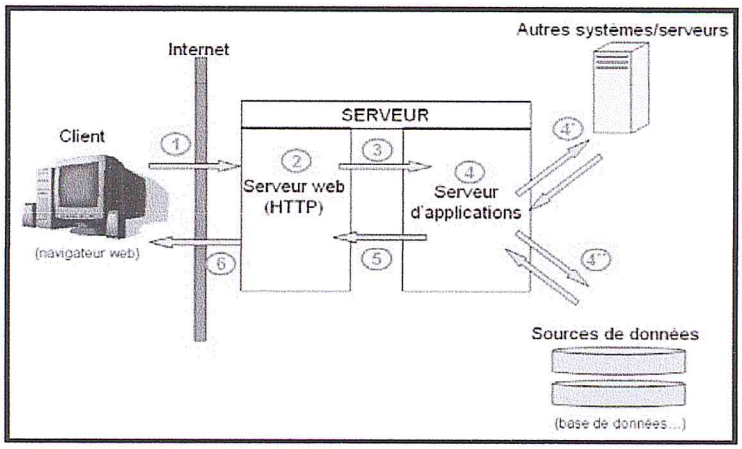


FIGURE 2.9 – Fonctionnement d'un Serveur d'Application.

Le scénario illustré dans le schéma ci-dessus est le suivant :

1. Le client émet une requête (i.e. appelle une URL) pour demander une ressource au serveur. Exemple : http ://leserveur.com/welcome.Il ne sait pas ici si la réponse qui va lui parvenir est statique (page HTML simple) ou dynamique (générée par une application Web).
2. Côté serveur, c'est le serveur Web (exemple : Apache) qui traite les requêtes HTTP entrantes. Il traite donc toutes les requêtes, qu'elles demandent une ressource sta-

tique ou dynamique. Seulement, un serveur HTTP ne sait répondre qu'aux requêtes visant des ressources statiques. Il ne peut que renvoyer des pages HTML, des images,... existantes.

3. Ainsi, si le serveur HTTP s'aperçoit que la requête reçue est destinée au serveur d'applications, il la lui transmet. Les deux serveurs sont reliés par un canal, nommé connecteur.
4. Le serveur d'applications (exemple : Tomcat) reçoit la requête à son tour. Il est, lui, en mesure de la traiter. Il exécute donc le morceau d'application (la Servlet) auquel est destinée la requête, en fonction de l'URL. Cette opération est effectuée à partir de la configuration du serveur. La Servlet est donc invoquée, et le serveur lui fournit notamment deux objets Java (Tomcat est un serveur d'applications Java) exploitables : un représente la requête, l'autre représente la réponse. La Servlet peut maintenant travailler, et générer la réponse à la demande. Cela peut passer par la consultation de sources de données, comme des bases de données (4" sur le schéma). Ou bien par l'interrogation d'autres serveurs ou systèmes (4' sur le schéma), l'environnement Java Web permettant de se connecter à de nombreux systèmes.
5. Une fois sa réponse générée, le serveur d'applications la renvoie, par le connecteur, au serveur Web. Celui-ci la récupère comme s'il était lui-même allé chercher une ressource statique. Il a simplement délégué la récupération de la réponse, et celle-ci a été générée, mais ce n'est plus le problème.
6. Le serveur HTTP peut donc retourner la réponse au client [17].

2.4.4 Serveur Web / Serveur d'application :

Dans le schéma précédent (figure 3.9), le serveur Web et le serveur d'applications sont séparés. Ces deux composants sont en effet nécessaires côté serveur, puisqu'ils se complètent : le serveur d'applications ne sait pas traiter une requête HTTP, le serveur Web ne sait pas exécuter d'applications.

Si ces deux composantes sont indispensables, elles ne sont pas nécessairement séparées. Il est possible qu'un serveur d'application inclue ainsi un serveur Web, et est donc capable de fonctionner en autonomie (StandAlone), pour traiter à la fois les requêtes HTTP simples (ressources statiques) et les applications Web. Le principe est de changer de connecteur (par rapport à notre schéma en haut de la page), pour en utiliser un comprenant les requêtes HTTP et non plus les requêtes triées venant du serveur Web.

2.4.5 Le serveur d'application Tomcat :

Apache Tomcat est un conteneur libre de Servlet Java EE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des Servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP (Web).

Le projet Tomcat a été lancé comme implémentation de référence des Servlets par James Duncan Davidson, architecte logiciel chez Sun. Il a contribué à rendre le projet libre et a joué un rôle majeur dans sa donation par Sun à la fondation Apache. Le projet Tomcat fait partie d'un collectif Open Source connu sous le nom de Jakarta.

Il est constitué de composants suivants :

• **Catalina** est le conteneur des Servlets, et implémente les spécifications de Sun pour les Servlets et les JSP ;
• **Coyote** est le connecteur HTTP : il écoute le trafic entrant, dirige les requêtes au moteur de Tomcat, processe la requête et renvoie la réponse au client ;
• **Jasper** est le moteur JSP. Il parse les fichiers JSP pour les compiler en tant que Servlets (gérable par Catalina). Il est capable de détecter les modifications des fichiers et de les recompiler à la volée.

Les avantages de ce serveur d'application sont :
-Tomcat est simple, beaucoup plus que les serveurs d'application Open Source «complets» ;
- Il est donc plus simple d'administrer une instance Tomcat qu'un serveur d'applications complet ;
- Il n'occupe que deux (2) ports sur la machine (8080 et 8009), alors que les autres en prennent une dizaine :

8080 : port propre de Tomcat
8009 : port de communication entre Apache et Tomcat [18].

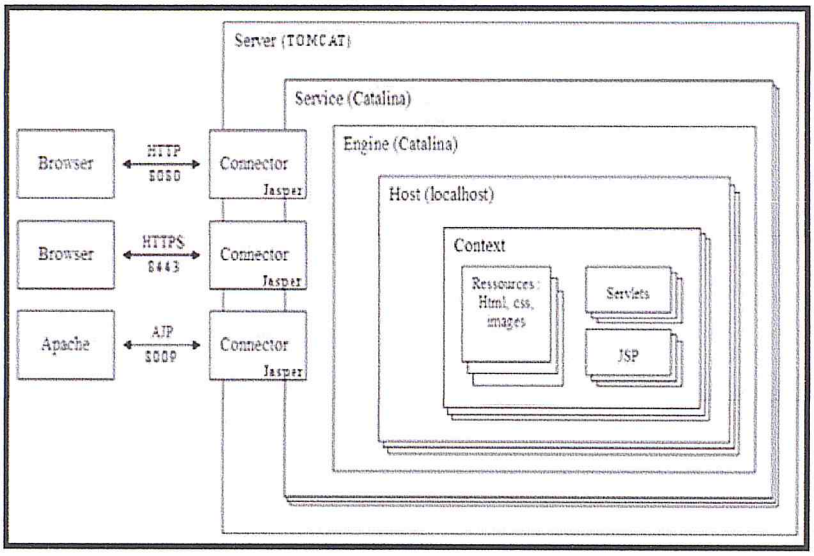


FIGURE 2.10 – Fonctionnement de Tomcat.

Dans cette partie, nous avons défini que ce que une application web ainsi que son fonctionnement, mais cela doit être sécurisé a l'aide de chiffrement des données que nous allons les définir dans la partie suivante.

2.5 Chiffrement des données :

Les algorithmes de hashage comme MD5, SHA1 et SHA256 sont destinés à être rapides et efficaces. Avec les équipements informatiques modernes, il est devenu facile d'attaquer par force brute la sortie de ces algorithmes pour retrouver la chaîne originale [20]. C'est la raison pour laquelle les experts conseillent l'utilisation d'algorithmes de chiffrement largement éprouvés.

On trouve principalement deux grandes familles de cryptage : le cryptage symétrique (ou dit à clé secrète) et le cryptage asymétrique (dit aussi à clé publique) [21].

2.5.1 Le cryptage symétrique :

On parle de cryptage symétrique lorsqu'un texte, document, etc. est crypté et décrypté avec la même clé, la clé secrète, ce procédé est à la fois simple et sûr. On trouvera principalement parmi les algorithmes de cryptage symétrique : AES, qui serait utilisé pour protéger des documents secrets aux États-Unis. Principal inconvénient : étant donné que l'on n'a qu'une clé, si vous la donnez à X pour qu'il puisse vous envoyer des messages cryptés avec celle-ci, il pourra aussi bien décrypter tous les documents que vous avez cryptés avec cette dernière. La clé est donc connue uniquement par le destinataire et

l'émetteur et il est plus sûr de faire une clé pour un échange entre X et Y, pour éviter qu'avec une clé on puisse tout décrypter [21].

2.5.2 Le cryptage asymétrique :

Contrairement au cryptage symétrique, ici avec l'asymétrique, on a 2 clés. Tout d'abord nous avons la clé publique. Celle-ci, tout le monde peut la posséder, il n'y a aucun risque, vous pouvez la transmettre à n'importe qui. Elle sert à crypter le message. Puis il y a la clé privée que seul le récepteur possède. Elle servira à décrypter le message crypté avec la clé publique [21].

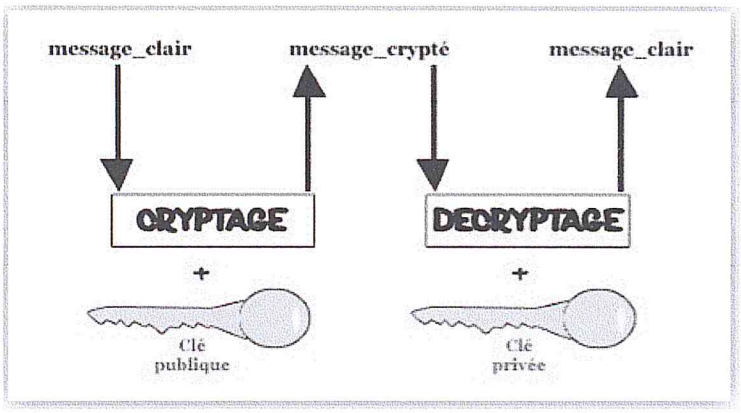


FIGURE 2.11 – Cryptage asymétrique [12].

Parmi les algorithmes de chiffrement asymétrique, on trouve le RSA (le plus connu), le PGP, le DSA...

2.5.3 Le cryptage RSA :

Le cryptage RSA, du nom de ses concepteurs, Ron Rivest, Adi Shamir et Leonard Adleman, est le premier algorithme de chiffrement asymétrique. Il a été découvert en 1977 au Massachusetts Institute of Technology.
Un chiffrement asymétrique est un cryptage où l'algorithme de chiffrement n'est pas le même que celui de déchiffrement, et où les clés utilisées sont différentes. L'intérêt est énorme : il n'y a plus besoin de transmettre la clé à son destinataire, il suffit de publier librement les clés de cryptage. N'importe qui peut alors crypter un message, mais seul son destinataire, qui possède la clé de décodage, pourra le lire.

2.5.3.1 Fonctionnement de l'algorithme RSA :

Un algorithme de chiffrement est un ensemble d'opérations, généralement mathématiques, à effectuer sur le message que l'on veut protéger, afin de le rendre en théorie incompréhensible par ceux auxquels il n'est pas destiné. Il vient toujours avec un algorithme de déchiffrement, qui permet au récipiendaire du message de retrouver le message d'origine [22].

Le chiffrement asymétrique est aussi appelé "à clé publique", même si en fait, on manipule une paire de clés : une publique et une privée [23].

Il est asymétrique car ne fonctionne que dans un seul sens. Par exemple :
- A veut recevoir une information provenant de B.
- A génère une clé publique et une clé privée.
- A envoie la clé publique à B.
- B chiffre son information avec la clé publique.
- B envoie le message chiffré à A.
- A déchiffre en utilisant la clé privée [23].

L'avantage de cette méthode de cryptage dépend des nombres premiers qui lui sont fournis en paramètres. Plus les nombres premiers sont grands, plus la sécurité de la clé privée est importante. A l'opposé, cette technique asymétrique est beaucoup plus lente que les techniques symétriques de cryptage, étant donné qu'elle utilise des calculs pouvant atteindre des valeurs astronomiques.

Il est utilisé pour de nombreuses choses mais sont emploi le plus connu et le plus courant concerne SSL (Secure Socket Layer), ce dernier est utilisé dans la sécurisation du trafic.

2.6 Sécurisation du trafic :

Avec l'évolution de l'Internet, de nouveaux mécanismes de sécurité du trafic sont devenus nécessaires, que ce soit en raison de l'émergence de nouveaux types d'attaques ou de l'identification de nouvelles failles de sécurité. Des solutions ont été proposées et déployées progressivement. Ces solutions comprennent entre autres : Les certificats SSL (Secure Socket Layer) pour sécuriser et authentifier les communications sur Internet et au sein des intranets d'entreprise, IPSec (Internet Protocol Security) pour sécuriser la couche réseau (également appelée couche IP), TLS (Transport Layer Security) pour sécuriser la communication entre deux applications Internet, telles qu'un serveur Web et un navigateur

Web, DNSSEC (Domain Name System Security Extensions) pour sécuriser le processus de résolution DNS, etc.

2.6.1 Le protocole SSL :

Développé par Netscape en 1995, le protocole SSL s'est rapidement imposé comme le mode de sécurisation privilégié des transmissions de données sur Internet.

Intégré aux principaux navigateurs et serveurs Web, SSL utilise des techniques de cryptage qui s'appuient sur un système de clé publique/privée initialement développé par RSA.

L'établissement d'une connexion SSL nécessite l'installation d'un certificat numérique sur le serveur Web. Ce certificat utilise alors les clés publiques et privées pour le cryptage, et identifie le serveur de manière unique et définitive. Les certificats numériques s'apparentent à une forme de carte d'identité électronique qui permet au client d'authentifier le serveur avant l'établissement d'un canal de communication crypté.

2.6.2 Champs d'application du SSL :

Le protocole SSL peut être utilisé de diverses façons et à des fins différentes :

Communications « **de navigateur à serveur** » : SSL sert à sécuriser les communications entre un serveur Web et un navigateur, notamment dans le cadre de transmissions d'informations sensibles (achats en ligne, dossiers médicaux ou transactions bancaires). La technologie SSL permet de confirmer à l'utilisateur l'identité du destinataire de ses informations personnelles, tout en assurant que seule cette entité autorisée y aura accès.

Communications « **de serveur à serveur** » : le protocole SSL peut également être utilisé pour sécuriser les communications entre deux serveurs, telles que les transactions entre deux entreprises. Dans ce scénario, les deux serveurs possèdent généralement un certificat qui leur permet de s'authentifier mutuellement et de sécuriser leurs communications bilatérales [19].

2.7 Conclusion :

A travers ce chapitre nous avons étudié quelque modèle de contrôle d'accès et quelques travaux qui ont été fait sur ces derniers ainsi que la construction d'une page Web et la sécurisation du trafic, grâce à des comparaisons faites sur les modèles, cela nous a permis de choisir le modèle ORBAC qui satisfait le plus nos besoins pour notre travail.

CHAPITRE 3

CONCEPTION

3.1 Introduction :

Ce chapitre a pour but d'analyser les fonctionnalités de l'application ainsi que sa sécurité qui est un aspect important dans notre projet, et de présenter les différents diagrammes et modèle de conception en utilisant le langage UML. Nous allons procéder comme suit : En premier lieu, nous présentons les approches de conception. Après , nous présentons la conception de l'application, et dans la dernière partie nous présentons la conception de la sécurité en utilisant le modèle Orbac.

3.2 conception générale :

Deux approches de conceptions existent : l'approche fonctionnelle qui voit le système comme un ensemble de fonctions à réaliser et l'approche objet qui voit le système comme un ensemble d'objets.

On choisira dans ce projet l'approche objet. La modélisation objet consiste à créer une représentation informatique des éléments du monde réel auxquels on s'intéresse.

3.2.1 Le Langage UML :

UML (Unified Modèleing Language), se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [14].

3.2.1.1 Les Vues UML

UML fournit un moyen astucieux permettant de présenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes.

A. Les vues statiques :

Elles permettent de représenter le système physiquement. On trouve alors les diagrammes suivants :

- Le Diagramme de classes : il représente la structure statique en termes de classes et de relations entre elles, il représente aussi un ensemble d'interface et de paquetages ainsi que leurs relations.
- Le Diagramme d'objets ou le diagramme d'instances : représente une instance possible

du diagramme de classes.
- Le Diagramme de composants : représente les morceaux d'applications packagés sous la forme de composants disposants d'interfaces. Il permet de décrire ces composants qui sont : le sous-système, le module, le programme et le sous-programme, le processus et la tâche.
- Le Diagramme de déploiement : complémentaire du diagramme de composants, il décrit la répartition physique des instances de composants, de processus et d'objets d'une application distribuée[26].

B. Les vues dynamiques :

Les cinq diagrammes comportementaux (ou dynamiques) représentent des vues dynamiques du système :

- Le Diagramme de cas d'utilisation : sont des vues qui décrivent les interactions entre les différents acteurs externes (utilisateurs du cas) et les fonctionnalités du système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur. Leur but est d'identifier les acteurs du domaine, leurs responsabilités respectives et de décrire leurs besoins.
- Le Diagramme de collaboration : il décrit l'interaction modélisée par les échanges de messages entre objets ou entre acteurs et objets.
- Le Diagramme de séquence : il diffère légèrement du diagramme de collaboration par l'ajout d'une dimension temporelle en précisant la chronologie des échanges de messages entre les objets.
- Le Diagramme d'états transitions : il décrit l'ensemble des états des objets du système et les transitions qui déclenchent le passage d'un état donné vers un autre état.
- Le Diagramme d'activités : est une variante des diagrammes d'états-transitions. Il décrit l'ensemble des activités effectuées par les acteurs du système en les décomposant en sous-activités et en spécifiant les contraintes relatives à l'enchaînement de ces dernières[26].

3.2.2 Avantages d'UML :

UML est un langage formel et normalisé qui facilite la compréhension de représentations abstraites complexes et le principal avantage d'UML est qu'il est devenu le standard en terme de modélisation objet, son caractère polyvalent et performant et sa souplesse ont fait un langage universel[26].

Ces avantages sont multiples :

· C'est un langage formel et normalisé.
· Gain de précision.
· Gage de stabilité.
· Il Encourage l'utilisation d'outils.
· UML est un support de communication performant.
· Il cadre l'analyse.
· Il facilite la compréhension de représentations abstraites complexes.

3.2.3 Conception et architecture :

Notre travail consiste à concevoir, réaliser et sécuriser une application de gestion de formation professionnelle spécialisée en basant sur le modèle MVC, qui est constitué de trois parties. La partie visuelle V (les interfaces Homme-Machine), le modèle M (le serveur de données) et la troisième partie représentée comme contrôleur de trafic C (le serveur d'application).

Cette architecture a plusieurs d'avantages, c'est la plus utilisée dans le monde du développement Web étant donnée qu'elle se caractérise par :
- L'allégement du poste de travail.
- La prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.).
- Une meilleure répartition de la charge entre les différentes entités clients et serveurs[27].

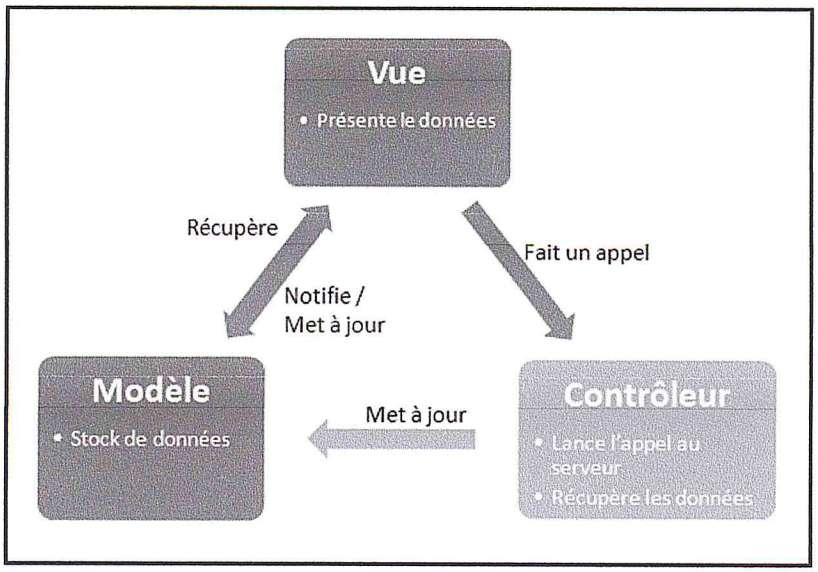


FIGURE 3.1 – L'architecture MVC [13].

3.3 Conception détaillée :

A-Conception de l'application :

Dans notre application, nous allons définir les étapes suivantes :

1- Définir les diagrammes des cas d'utilisations.
2- Définir les diagrammes d'activités.
3- Définir le diagramme de classes.

3.3.1 Diagrammes de cas d'utilisation :

On utilise les diagrammes des cas d'utilisation pour représenter et structurer au niveau conceptuel, les besoins des utilisateurs et les objectifs correspondants du système. Le but est d'identifier les acteurs du domaine et leurs interactions avec l'interface. Ce diagramme permet de déterminer le modèle objet sur lequel le système reposera.

3.3.1.1 Avantages :

· Formalisme simple : Les concepts proposés sont faciles à comprendre et à utiliser.
· Les modélisations résultats : Facile à comprendre, à lire et à interpréter.
· Un bon moyen de communication.

3.3.1.2 Identification des acteurs :

Un acteur représente un rôle joué par une entité externe qui interagit directement avec le système étudié. On distingue cinq intervenants qui interagissent avec l'interface : chef département, responsable de la subdivision de formation, responsable action de formation, Agent de saisie, Agent d'accueil.

3.3.1.3 Identification des activités :

Décrit l'ensemble des activités effectuées par les acteurs du système en les décomposant en sous activités et en spécifiant les contraintes relatives à l'enchaînement de ces activités.

On trouve treize activités dans notre projet. On va citer six d'entre elles qui sont essentiel :

Gestion des promotions.
Gestion des domaines.
Gestion des phases.
Gestion des exclusions.
Gestion des formations.

Gestion des Stagiaires.

3.3.1.4 Les diagrammes de cas d'utilisations :

1. Diagramme de cas d'utilisation général :

Le diagramme ci-dessous montre les activités ainsi que les acteurs du système :

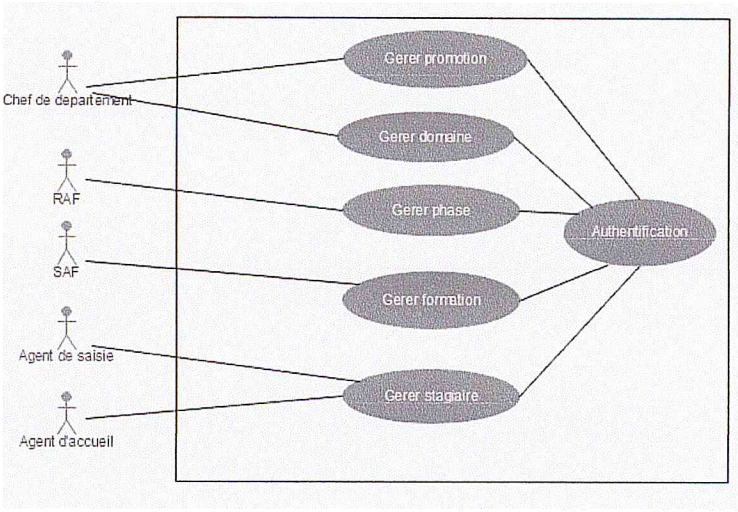


FIGURE 3.2 – Diagramme cas d'utilisation général.

2. Diagramme de cas d'utilisation «Gérer promotion » :

Le diagramme suivant montre la gestion des promotions d'une phase dans une période précise :

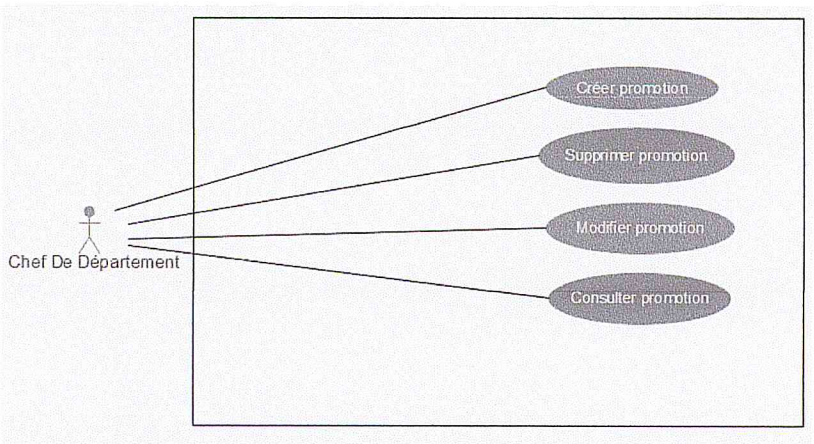


FIGURE 3.3 – Cas d'utilisation « Gérer promotion ».

3. Diagramme de cas d'utilisation «Gérer domaine » :

Le diagramme ci-dessous montre la gestion des domaines par l'acteur Chef de départent :

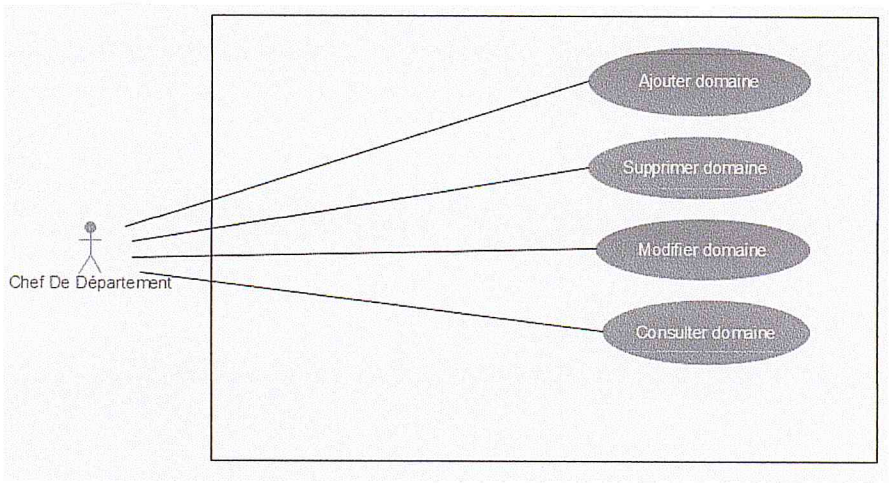


FIGURE 3.4 – Cas d'utilisation « Gérer domaine ».

4. Diagramme de cas d'utilisation «Gérer phase » :

Le diagramme suivant montre la gestion des phases d'une promotion :

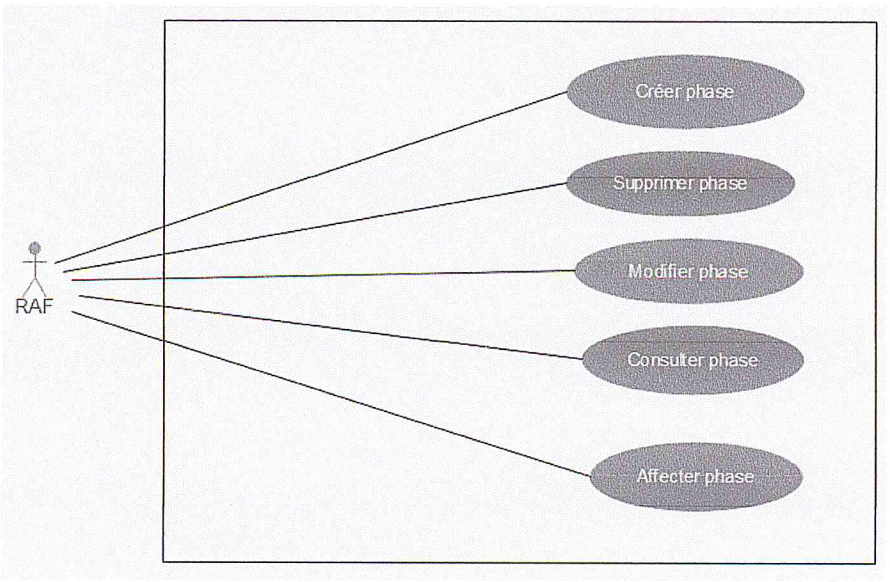


FIGURE 3.5 – Cas d'utilisation « Gérer phase ».

6. Diagramme de cas d'utilisation «Gérer formation » :

Le diagramme suivant montre la gestion des formations d'un domaine :

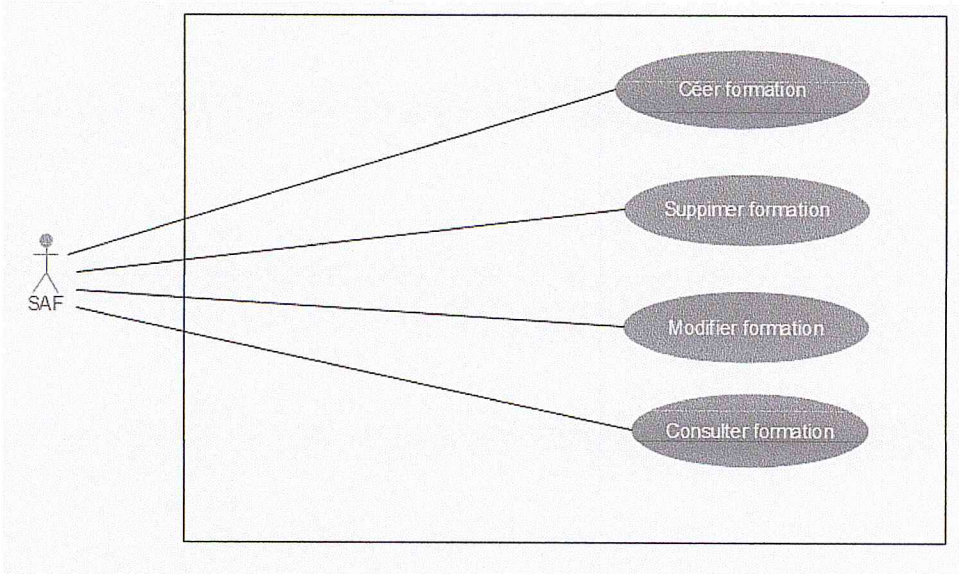


FIGURE 3.6 – Cas d'utilisation « Gérer formation ».

7. Diagramme de cas d'utilisation «Gérer stagiaires » :

Le diagramme suivant montre la gestion des stagiaires d'une promotion :

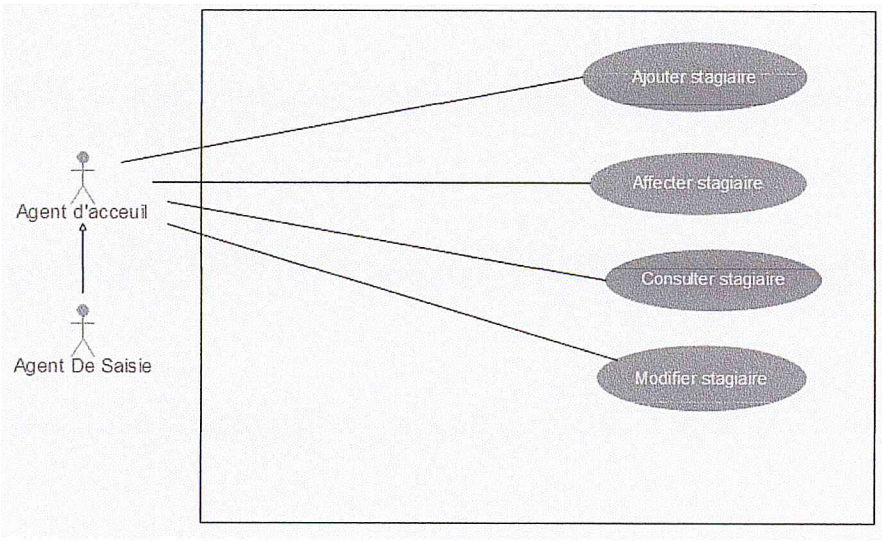


FIGURE 3.7 – Cas d'utilisation « Gérer stagiaire ».

3.3.2 Diagramme d'activité :

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

3.3.2.1 Les digrammes d'activités :

1. Diagramme d'activité d'authentification :

Le diagramme d'activité d'authentification nous permet de voir les comportements internes du système, lors du démarrage de l'application par l'utilisateur, le système lui affiche le formulaire d'authentification, après que le mot de passe soit saisit le système vérifie sa validité dans le modèle Orbac et affiche la page d'accueil sinon il affiche un message d'erreur.

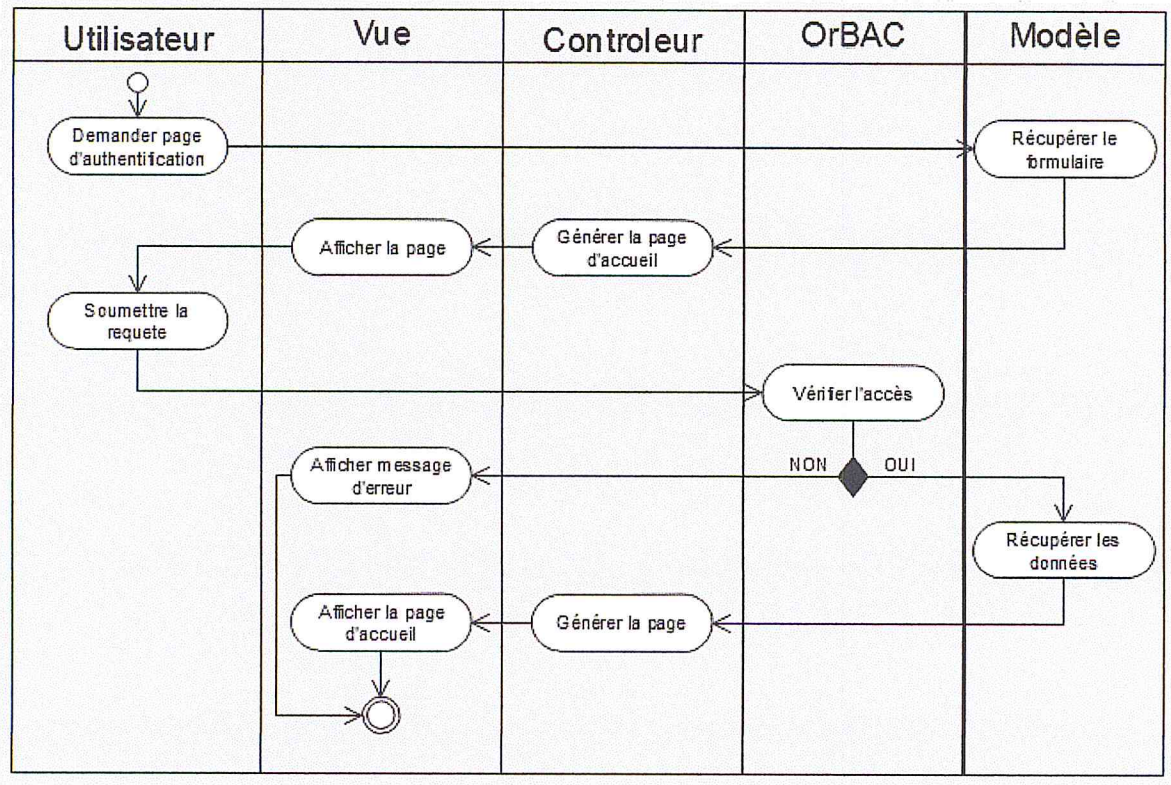


FIGURE 3.8 – Diagramme d'activité de l'authentification.

2. Diagrammes d'activité création de promotion :

Lors de la demande de formulaire de création de promotion, le système vérifie dans le modèle Orbac c.à.d. dans les permissions et les interdictions et affiche le formulaire sinon il affiche un message d'erreur.

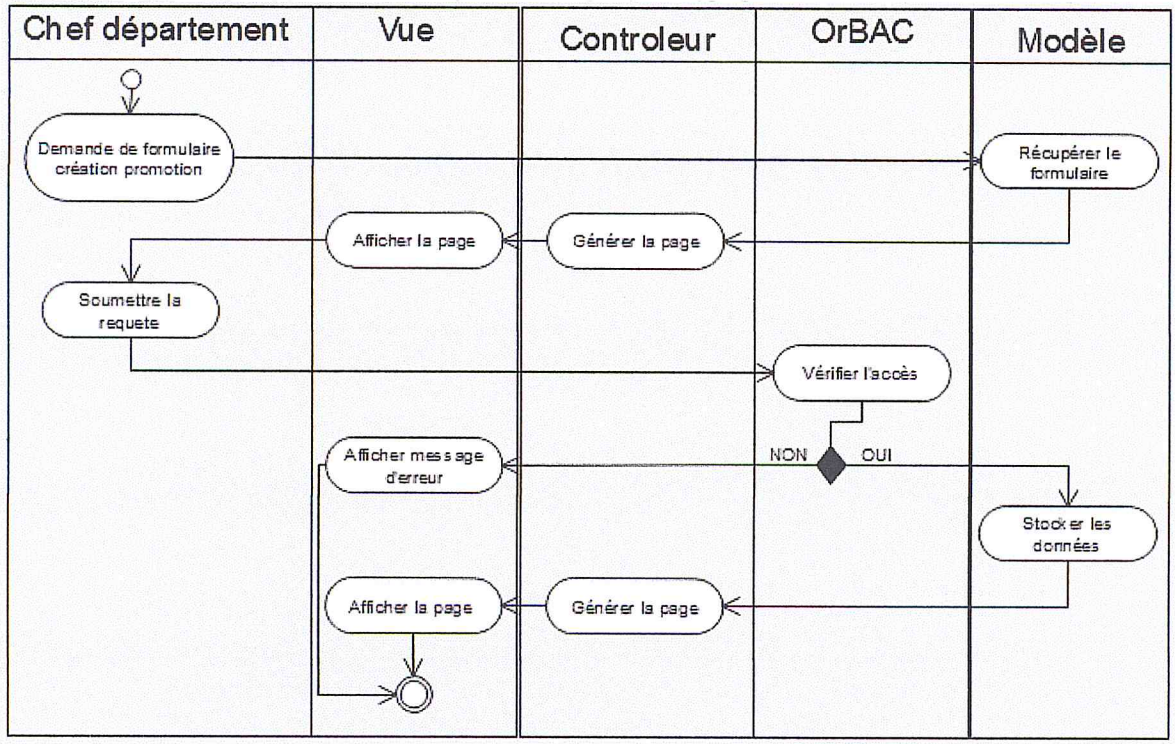


FIGURE 3.9 – Diagramme d'activité création de promotion.

3. Diagrammes d'activité annulation d'exclusion :

Lors de la demande de formulaire d'annulation d'exclusion, le système vérifie dans le modèle Orbac et affiche le formulaire sinon il affiche un message d'erreur.

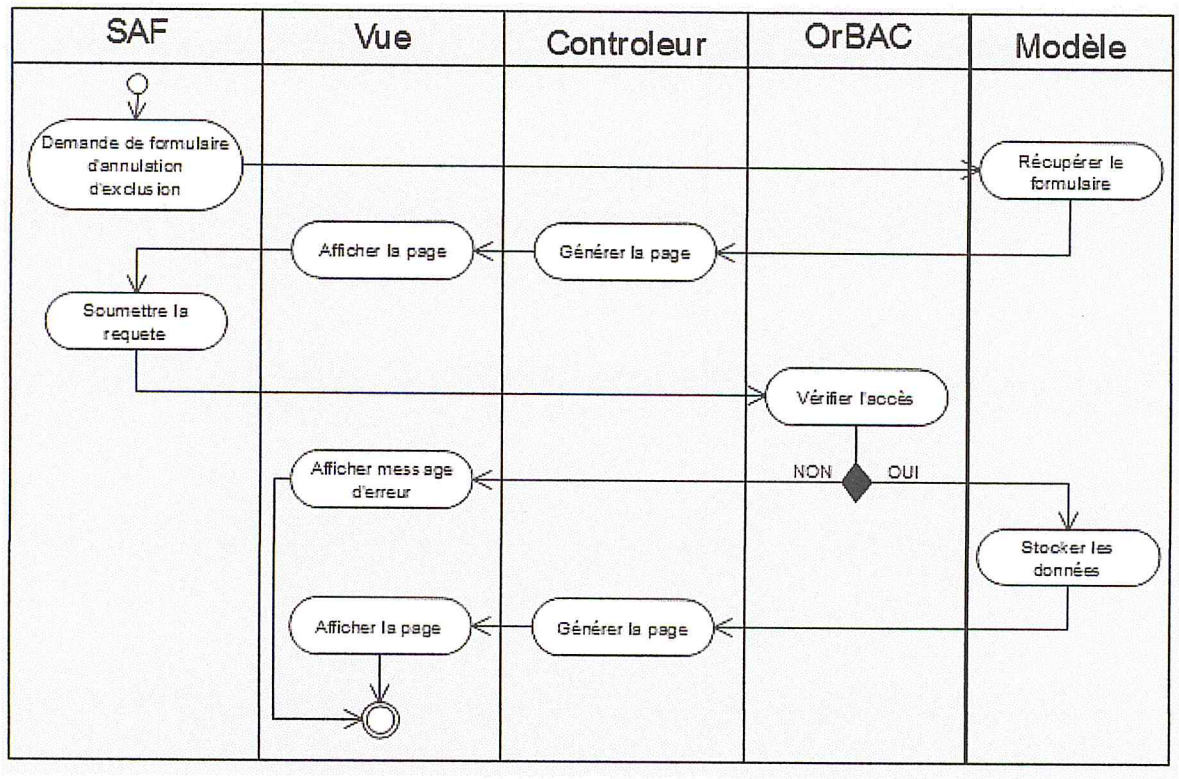


FIGURE 3.10 – Diagramme d'activité annulation d'exclusion.

4. Diagrammes d'activité affectation de la phase :

Lors de la demande de formulaire d'affectation de phase, le système vérifie dans le modèle Orbac et affiche le formulaire sinon il affiche un message d'erreur.

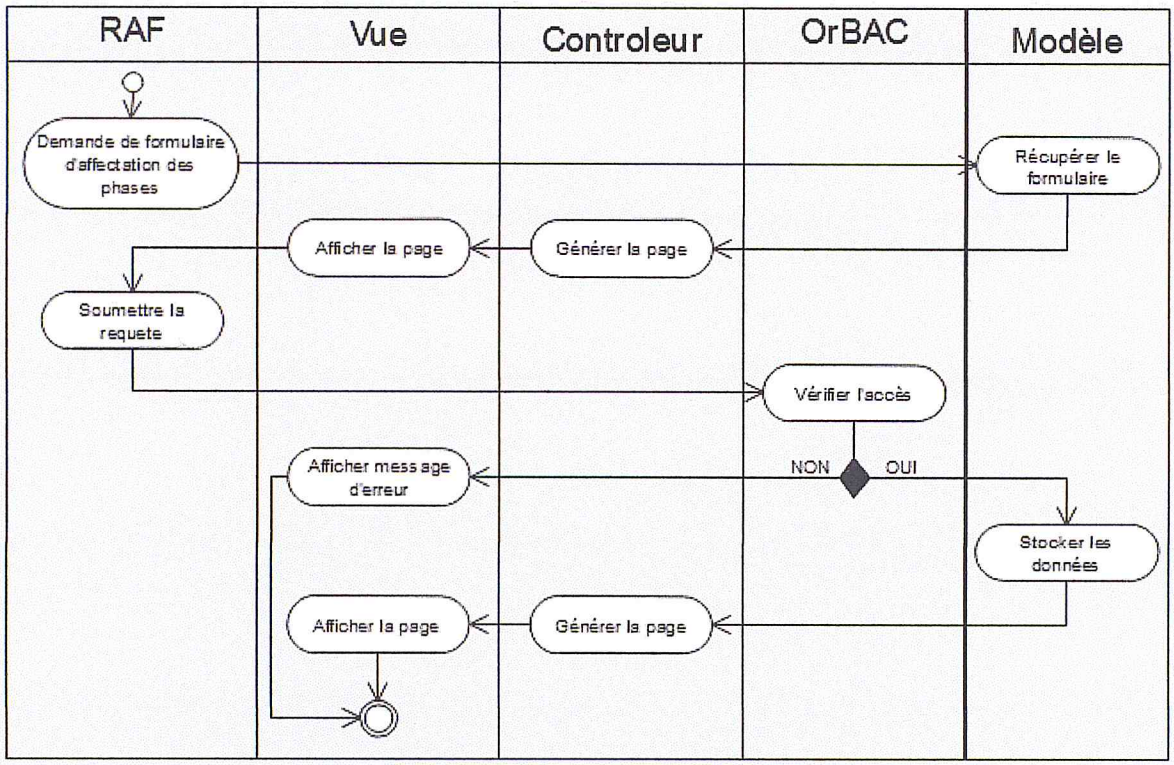


FIGURE 3.11 – Diagramme d'activité affectation de phase.

3.3.3 Diagramme de déploiement :

Le diagramme de déploiement spécifie un ensemble de constructions qui peut être utilisé pour définir l'architecteur d'exécution de systèmes qui représentent l'affectation d'artefacts logiciels a des nœuds. Les nœuds sont connectés via des chemins de communications pour créer des systèmes de réseau d'une complexité quelconque. Les nœuds sont général définis d'une manière imbriquée et représentent soit des périphériques matériels, soit des environnements d'exécutions de logiciels. Les artefacts représentent des éléments concrets du monde physique qui sont le résultat d'un processus de développement [24].

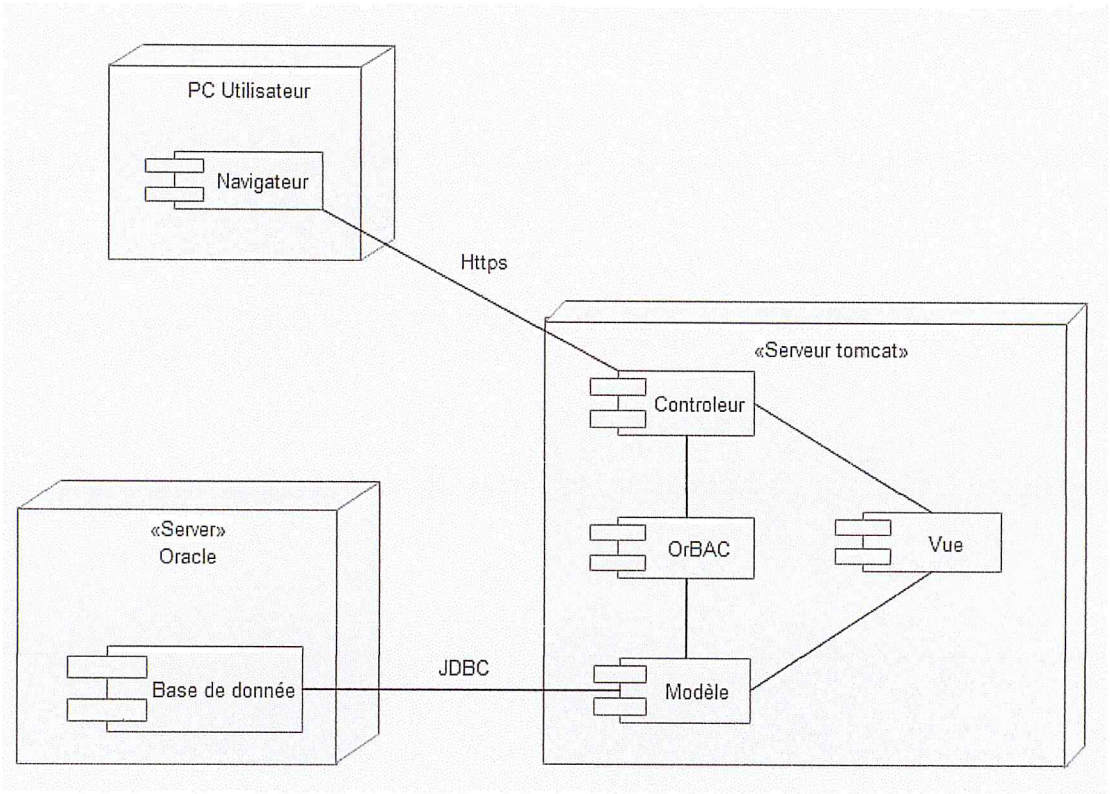


FIGURE 3.12 – Diagramme de déploiement.

3.3.4 Diagramme de classe :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Dans notre projet on a pris la base de donnée qui existe et on l'a modifier c.à.d. on a supprimé les redondances, et on a prié que les classes qui sont dans notre champs d'études.

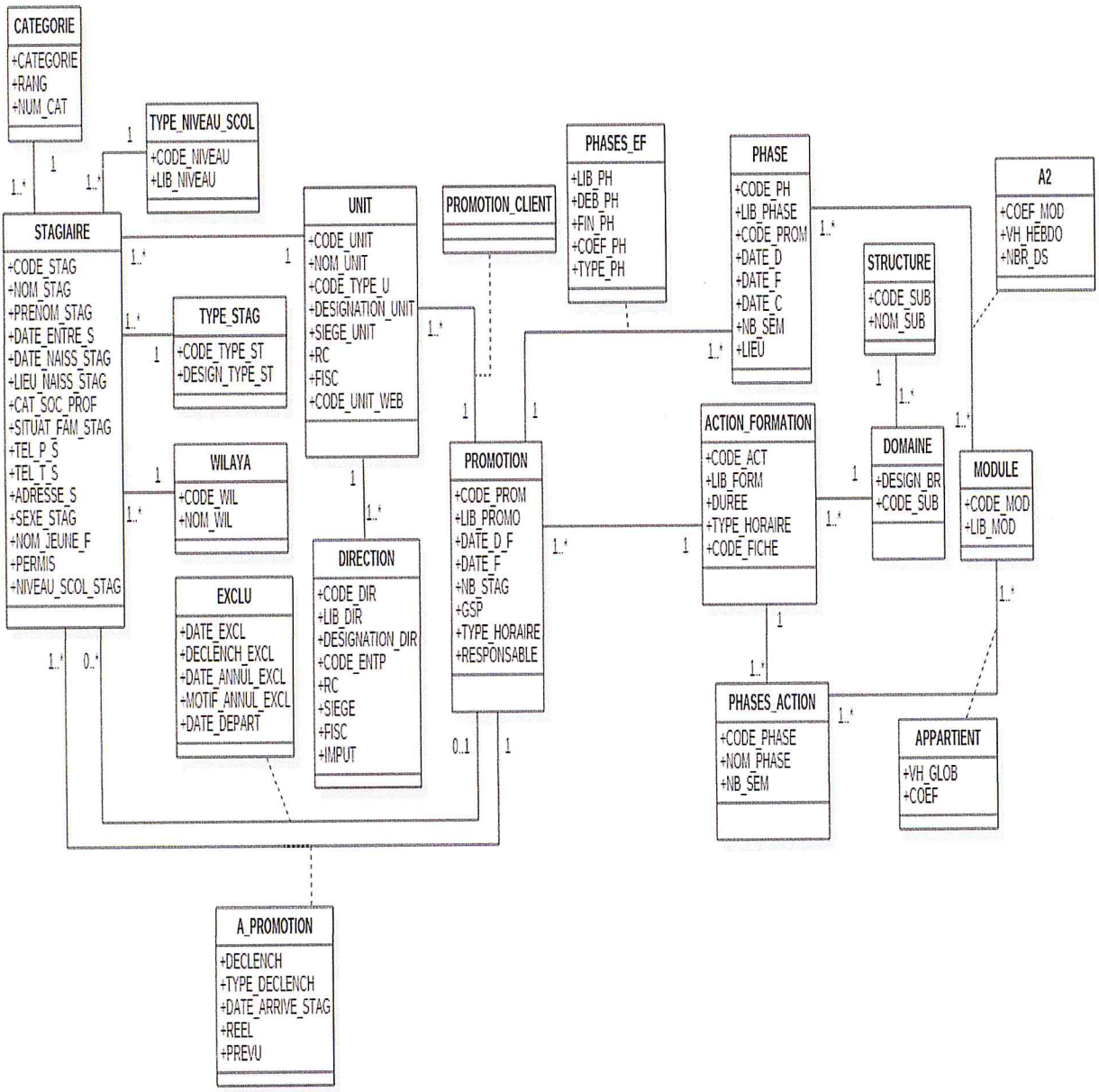


FIGURE 3.13 – Diagramme de classe.

B - Conception de modèle OrBAC :

L'organisation est l'entité centrale du modèle, donc la spécification des politiques de sécurité suit la structure de l'organisation. La possibilité d'exprimer des permissions, des obligations et des interdictions qui dépendent de contextes, est un élément qui va vers une plus grande expressivité. L'abstraction des entités traditionnelles du contrôle d'accès(sujet, action, objet)en méta entités(rôle, activité, vue)permet d'élaborer une politique de sécurité a deux niveaux, un niveau concret et un niveau abstrait. Tout cela dans l'organisation de l'école technique de Blida qui est défini comme org(ETB).

1- Les entités concrètes :

Les entités concrètes qui sont au niveau concret sont définies comme suit :
- Les sujets.
- Les actions.
- Les objets.

1.1- Les sujets : Un sujet est une entité active, c'est-à-dire un utilisateur, donc les sujets de notre modèle sont les suivants :

Sujet(Charif), Sujet(Labini), Sujet(Rakmi), Sujet(Rasib), Sujet(Akili), Sujet(Aboud)

1.2- Les actions : Les actions sont des opérations qu'un sujet peut faire sur un objet :

Action (lire), Action (écrire), Action (supprimer), Action (modifier)

1.3- Les objets : Représente principalement les entités non active comme les fichiers, les courriers électroniques, les formulaires imprimés, etc.

Dans notre modèle les objets sont les suivants :

Objet(AF), Objet(DOM), Objet(Exclu), Objet(Promo), Objet (StagiaireP), Objet(StagiaireR) ,Objet(Ph), Objet(Mod).

2- Les entités abstraites :

On définit trois entités abstraites :

2.1- Les rôles : Les entités Rôles sont abstraites à partir des sujets, et qui sont joués par des utilisateurs :

Rôle(Chef De Département), Rôle(Chef De Laboratoire), Rôle(RAF), Rôle(Responsable De la SAF),Rôle(Agent De Saisie),Rôle (Agent D'Accueil).

2.2- Les activités : Les entités activités sont l'abstraction des actions qui sont dans le niveau concret.

Dans notre modèle les activités sont :

Activité (gérer inscription prévisionnelle), Activité (gérer inscription réelle), Activité (gérer stagiaire), Activité (gérer exclusion), Activité (gérer formation), Activité (gérer promotion), Activité (gérer phase), Activité (gérer Module).

2.3- Les vues : Une vue correspond, comme dans la base de donnée relationnelles, a un ensemble d'objets qui satisfait une propriété commune.

Dans notre modèle les vues sont :

Vue (Promotion), Vue (Phase), Vue (Formation) Vue(Module), Vue (Exclusion), Vue (Stagiaire prévisionnel), Vue (Stagiaire réel), Vue (Domaine).

3- Relation entre niveau abstrait et concret :

3.1- Relation sujet rôle organisation :

La relation entre sujet, rôle et organisation s'appelle Habilite(org,s,r) signifie que org habilite le sujet s à jouer le rôle r en considérant org une organisation, s un sujet, et r est un rôle.

Dans notre modèle les relations habilite sont les suivants :
Habilite(H1) $\subset$ HabiliteS.Sujet(Charif) $\wedge$ HabiliteR.Role(Chef De Département) $\wedge$ HabiliteOr.Organisation (ETB).

Par exemple dans cette expression qui défini la relation Habilite(H1), relie le sujet Charif avec le rôle Chef De Département, et cela dans l'organisation ETB. Et les expressions qui suivent sont défini de la même manière.

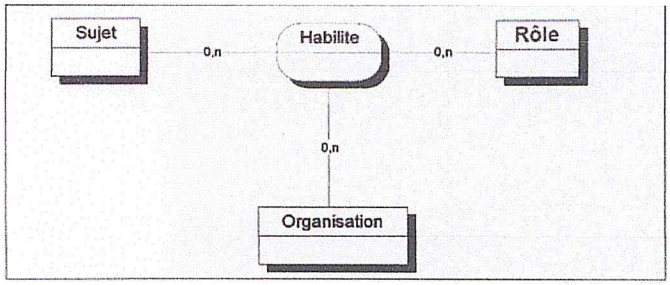


FIGURE 3.14 – La relation habilite [8].

Habilite(H2) ⊂ HabiliteS.Sujet(Labini) ∧ HabiliteR.Role(Chef De Laboratoire) ∧ habiliteOr.Organisation (ETB).
Habilite(H3) ⊂ HabiliteS.Sujet(Rakmi) ∧ HabiliteR.Role(RAF) ∧ habiliteOr.Organisation (ETB).
Habilite(H4) ⊂ HabiliteS.Sujet(Rasib) ∧ HabiliteR.Role(Responsable de la SAF) ∧ habiliteOr.Organisation (ETB).

3.2- Relation objet vue organisation :

La relation entre objet, vue et organisation s'appelle Utilise(org,o,v) signifie que org utilise l'objet o dans le vue v si org est une organisation, o est un objet et v est une vue.

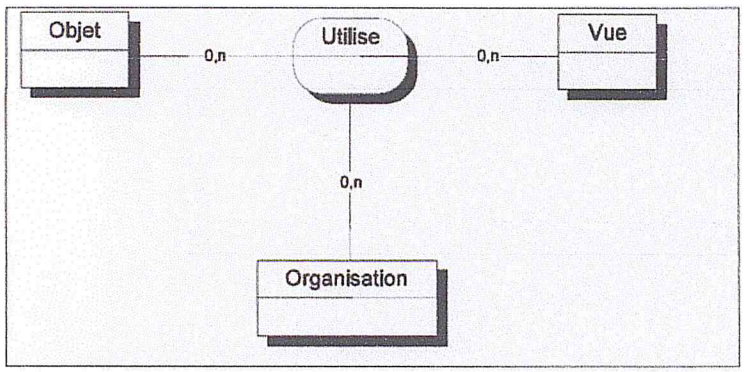


FIGURE 3.15 – Relation Utilise [1].

Dans notre modèle les relations objet vue organisation sont les suivants :

Utilise (U1) ⊂ UtiliseO.Objet(AF) ∧ UtiliseV.Vue(Formation) ∧ UtiliseOr.Organisation (ETB).

Cette expression défini la relation Utilise (U1), qui relie l'objet AF avec la vue For-

mation, et cela dans l'organisation ETB. Et les expressions qui suivent sont défini de la même manière.

Utilise (U2) ⊂ UtiliseO.Objet(Promo) ∧ UtiliseV.Vue(Promotion) ∧ UtiliseOr.Organisation (ETB).
Utilise (U3) ⊂ UtiliseO.Objet(Ph) ∧ UtiliseV.Vue(Phase) ∧ UtiliseOr.Organisation (ETB).
Utilise (U4) ⊂ UtiliseO.Objet(StagiaireP) ∧ UtiliseV.Vue(Stagiaire prévisionnel) ∧ UtiliseOr.Organisation (ETB).
Utilise (U5) ⊂ UtiliseO.Objet(StagiaireR) ∧ UtiliseV.Vue(Stagiaire Réel) ∧ UtiliseOr. Organisation (ETB).
Utilise (U6) ⊂ UtiliseO.Objet(Mod) ∧ UtiliseV.Vue(Module) ∧ UtiliseOr.Organisation (ETB).
Utilise (U7) ⊂ UtiliseO.Objet(Exclu) ∧ UtiliseV.Vue(Exclusion) ∧ UtiliseOr.Organisation (ETB).
Utilise (U8) ⊂ UtiliseO.Objet(Dom) ∧ UtiliseV.Vue(Domaine) ∧ UtiliseOr.Organisation (ETB).

3.3- Relation action activité organisation :

La relation entre action, activité et organisation s'appelle Considère (org, α ,a) signifie que l'organisation org considère l'action α comme faisant partie de l'activité a si org est organisation, α est une action et a est une activité.

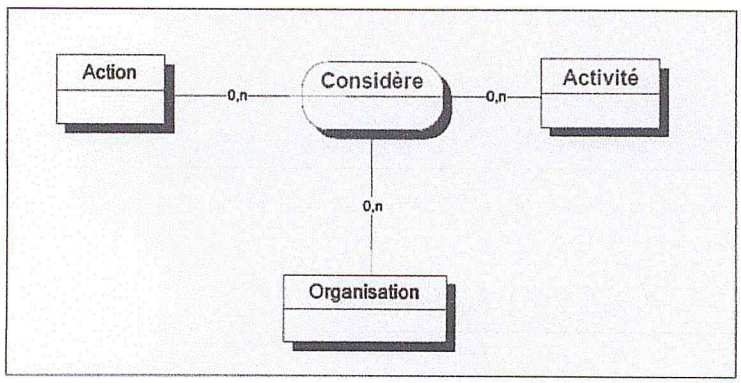


FIGURE 3.16 – La relation Considère [1].

Dans notre modèle y a beaucoup de relation considère.Dans ce qui suit,on cite juste quelques unes :
Considère(C1) ⊂ ConsidèreAc.Action(lire) ∧ ConsidèreAv.Activité(gérer inscription prévisionnelle) ∧ ConsidèreOr.Organisation(ETB).

Cette expression défini la relation Considère(C1),qui relie l'action Lire avec l'activité gérer inscription prévisionnelle , et cela dans l'organisation ETB. Et les expressions qui suivent sont défini de la même manière.

Considère(C2) ⊂ ConsidèreAc.Action(écrire) ∧ ConsidèreAv.Activité(gérer inscription prévisionnelle) ∧ ConsidèreOr.Organisation (ETB).

Considère(C3) ⊂ ConsidèreAc.Action(Modifier) ∧ ConsidèreAv.Activité(gérer inscription prévisionnelle) ∧ ConsidèreOr.Organisation (ETB).

Considère(C4) ⊂ ConsidèreAc.Action(Supprimer) ∧ ConsidèreAv.Activité(Inscription Prévisionnelle) ∧ ConsidèreOr.Organisation (ETB).

Considère(C5) ⊂ ConsidèreAc.Action(lire) ∧ ConsidèreAv.Activité(gérer module) ∧ ConsidèreOr.Organisation(ETB).

Considère(C6) ⊂ ConsidèreAc.Action(écrire) ∧ ConsidèreAv.Activité(gérer module) ∧ ConsidèreOr.Organisation (ETB).

Considère(C7) ⊂ ConsidèreAc.Action(Modifier) ∧ ConsidèreAv.Activité(gérer module) ∧ ConsidèreOr.Organisation (ETB).

Considère(C8) ⊂ ConsidèreAc.Action(Supprimer) ∧ ConsidèreAv.Activité(gérer module) ∧ ConsidèreOr.Organisation (ETB).

Considère(C9) ⊂ ConsidèreAc.Action(lire) ∧ ConsidèreAv.Activité(gérer formation) ∧ ConsidèreOr.Organisation (ETB).

Considère(C10) ⊂ ConsidèreAc.Action(ecrire) ∧ ConsidèreAv.Activité(gérer formation) ∧ ConsidèreOr.Organisation (ETB).

Considère(C11) ⊂ ConsidèreAc.Action(supprimer) ∧ ConsidèreAv.Activité(gérer formation) ∧ ConsidèreOr.Organisation (ETB).

4- Les contextes :

Les contextes sont utilisés pour spécifier les circonstances concrètes dans lesquelles les organisations accordent des permissions de réaliser des activités sur des vues. Ils peuvent être vus comme des relations entre les sujets, les objets et les actions définis dans une certaine organisation. La relations s'appelle Définit (org, s, α , o, c) signifie qu'au sein de l'organisation org, le contexte c est vraie entre le sujet s, l'objet o et l'action α si org est une organisation, s est un sujet, α est une action, o est un objet et c est un contexte.

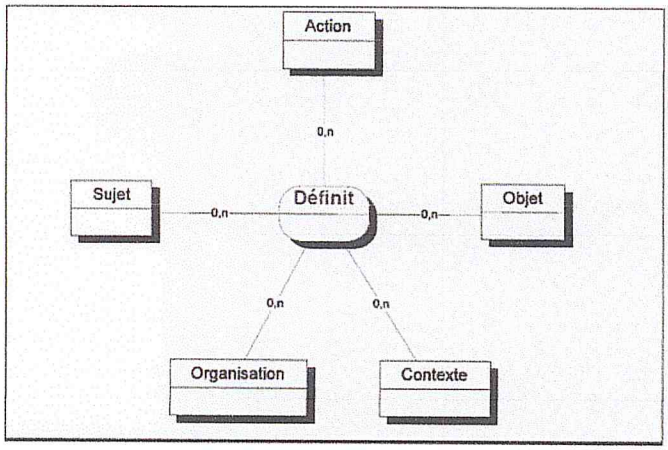


FIGURE 3.17 – La relation Définit [1].

Les contextes dans notre modèle sont les suivants :

- Les contextes défauts :

Les contextes défauts sont utilisés quand une permission utilise toute les actions dans un objet par un sujet.

Organisation (org) ∧ Sujet (s) ∧ Action(a) ∧ Objet(o) → Définit (org, s, a, o, défaut).
Organisation (ETB) ∧ Sujet (charif) ∧ Action (écrire) ∧ Objet(AF) → Définit (ETB, Charif, écrire, AF, défaut).

Cette expression défini le contexte défaut, relie le sujet Charif avec l'action ecrire ainsi que l'objet AF, et cela dans l'organisation ETB. Et les expressions qui suivent sont défini de la même manière.

Organisation (ETB) ∧ Sujet (Labini) ∧ Action (écrire) ∧ Objet(Promo) → Définit (ETB, Labini, écrire, Promo, défaut).
Organisation (ETB) ∧ Sujet (Labini) ∧ Action (lire) ∧ Objet(Promo) → Définit (ETB, Labini, lire, Promo, défaut).
Organisation (ETB) ∧ Sujet (Labini) ∧ Action (écrire) ∧ Objet(Ph) → Définit (ETB, Labini, écrire, Ph, défaut).
Organisation (ETB) ∧ Sujet (Labini) ∧ Action (lire) ∧ Objet(PhaseP) → Définit (ETB, Labini, lire, Ph, défaut).
Organisation (ETB) ∧ Sujet (Rakmi) ∧ Action (lire) ∧ Objet(Promo) → Définit (ETB, Rakmi, lire, Promo, défaut).
Organisation (ETB) ∧ Sujet (Rakmi) ∧ Action (écrire) ∧ Objet(Promo) → Définit (ETB, Rakmi, ecrire, Promo, défaut).

Organisation (ETB) ∧ Sujet (Rakmi) ∧ Action (lire) ∧ Objet(Ph) → Définit (ETB, Rakmi, lire, Ph, défaut).
Organisation (ETB) ∧ Sujet (Rakmi) ∧ Action (ecrire) ∧ Objet(Ph) → Définit (ETB, Rakmi, ecrire, Ph, défaut).

- Les contextes prérequis :

Les contextes prérequis sont définis pour les permissions qui ont une action particulière. Par exemple un sujet qui a un rôle qui permet de gérer les exclusions des stagiaires n'est pas forcement celui qui a la permission d'annuler les exclusions des stagiaires, c'est pour cela les contextes prérequis ont été définis.

Pour l'exemple de l'annulation des stagiaires, le contexte prérequis se définis comme suit :

Sujet(Rasib) ∧ Action(Supprimer) ∧ Objet(Exclu) → Définit (ETB, SAF, Supprimer, Exclu, Annuler Exclusion).

Ce contexte montre que le sujet Rasib a le droit de supprimer l'exclusion des stagiaires dans l'organisation(ETB).

Les contextes particuliers de notre modèle sont les suivants :

Organisation (ETB) ∧ Sujet(Rasib) ∧ Action (écrire) ∧ Objet(Stagiaire) → Définit (ETB, Rasib, écrire, Stagiaire, Ajouter Stagiaire).
Organisation (ETB) ∧ Sujet(Aboud) ∧ Action (écrire) ∧ Objet(Stagiaire_Prévisiennel) → Définit (ETB, Aboud, écrire, Stagiaire_Prévisiennel , Ajouter Prévisionnelle).
Organisation (ETB) ∧ Sujet(Rasib) ∧ Action (modifier) ∧ Objet(Exclu) → Définit (ETB, Rasib, écrire, Exclu, Modifier Exclusion).

5- Définition des relations Permission et Interdiction :

5.1- Relation Permission : la relation permission concrète est déduite à partir de relation permission abstraite.

5.1.1- Relation Permission abstraite : qui s'appelle Permission (org, r,a,v,c) signifie que dans l'organisation org une permission est accordée au rôle r de réaliser l'activité a sur la vue v dans un contexte c.

Les permissions abstraites de modèle qu'on a défini sont les suivants :

Permission (ETB, Chef Département, gérer formation, formation, défaut).

Dans cette expression, montre que le rôle Chef De Département a la permission de gérer les formations dans la vue formation en utilisant le contexte défaut, c.a.d toute les actions de l'activité gérer formation. Et les expressions qui suivent sont défini de la même manière.

Permission (ETB, Chef De Laboratoire, gérer promotion, promotion, défaut).
Permission (ETB, Chef De Laboratoire, gérer promotion, phase, défaut).
Permission (ETB, RAF, gérer promotion, promotion, défaut).
Permission (ETB, RAF, gérer phase, phase, défaut).
Permission (ETB, SAF, gérer formation, formation, défaut).
Permission (ETB, SAF, gérer Module, Module, défaut).
Permission (ETB, SAF, gérer exclusion, exclusion, annuler exclusion).
Permission (ETB, Agent Accueil, gérer inscription prévisionnelle, stagiaire prévisionnel, défaut)
Permission (ETB, Agent Accueil, gérer inscription réelle, stagiaire réel, défaut)
Permission (ETB, Agent Accueil, gérer inscription prévisionnelle, stagiaire prévisionnel, Ajouter Prévisionnel)

5.1.2- Relation Permission concrète : qui s'appelle Est_permi(s, a, o) signifie que le sujet s a la permission de réaliser l'action a sur l'objet o.

Pour ajouter les stagiaires prévisionnels, la relation permission concrète est déduite comme suite :

Permission (ETB, Agent Accueil, gérer inscription prévisionnelle, stagiaire prévisionnel, Ajouter Prévisionnel) $\wedge$ Habilite(H6) $\wedge$ Considère(C2) $\wedge$ Utilise (U5) $\wedge$ Définit (ETB, Aboud,écrire, Stagiaire Prévisionnel , Ajouter Prévisionnel) $\rightarrow$ Est_permi (AA, écrire, Stagiaire_Prévisiennel)

Les permissions concrètes de notre modèle sont définies de la même manière.

5.2 Relation Interdiction : la relation interdiction concrète est détruite à partir de relation interdiction abstraite.

5.2.1 Relation interdiction abstraite : qui s'appelle Interdiction (org, r,a,v,c) signifie que dans l'organisation org une interdiction est accordée au rôle r de réaliser l'activité a sur la vue v dans un contexte c.

Les interdictions abstraites de modèle qu'on a défini sont les suivants :

Interdiction (ETB, Chef De Laboratoire, gérer formation, formation, défaut).

Dans cette expression, montre que le rôle Chef DE Laboratoire n'a pas la permission de gérer les formations dans la vue formation en utilisant le contexte défaut, c.a.d toute les actions de l'activité gérer formation. Et les expressions qui suivent sont défini de la même manière.

Interdiction (ETB, Chef De Département, gérer phase, phase, défaut).
Interdiction (ETB, Chef De Départemen, gérer formation, formation, défaut).
Interdiction (ETB, Chef De Départemen, gérer module, module, défaut).
Interdiction (ETB, Chef De Département, gérer exclusion, exclusion, défaut).
Interdiction (ETB, Chef De Laboratoire, gérer module, module, défaut).
Interdiction (ETB, Chef De Laboratoire, gérer exclusion, exclusion, défaut).

5.2.2 Relation Interdiction concrète : qui s'appelle Est-interdit (s, a, o) signifie que le sujet s n'a pas le droit de réaliser l'action a sur l'objet o.

Pour bien illustrer ce point, on va expliquer la première interdiction.

Interdire le Chef De Laboratoire de gérer les formations plus précisément d'ecrire dans l'objet, la relation interdiction concrète est déduite comme suite :

Interdiction (ETB, Chef Laboratoire , gérer formation, formation, défaut) $\wedge$ Habilite(H2) $\wedge$ Considère(C11) $\wedge$ Utilise (U2) $\wedge$ Définit (ETB, Charif, écrire, AF, défaut) $\rightarrow$ Est-interdit (Charif, écrire, AF).

Les permissions concrètes de notre modèle sont définies de la même manière.

3.4 Conclusion :

Ce chapitre représente le cœur de notre projet, il nous a permis de concevoir et schématiser notre travail et représenter les différents éléments avec lesquels nous avons développé notre politique de contrôle d'accès. Il nous a permis aussi de concevoir le fonctionnement global de notre projet afin que tout le monde puisse le comprendre. Dans le chapitre suivant on présentera l'implémentation et les tests de notre application.

CHAPITRE 4

IMPLÉMENTATION

4.1 Introduction :

Dans ce chapitre on va présenter l'implémentation et les tests réalisés de notre application, on va mettre en pratique nos politiques de contrôle d'accès et présenter notre interface simplifiée pour que l'administrateur puisse facilement gérer les paramètres de confidentialité. Tout cela va être représenté grâce à des captures d'écran et des observations sur les résultats obtenus.

4.2 Environnement de travail :

Au niveau de cette partie, nous allons énumérer les différents outils matériels et logiciels que nous avons utilisés pour le développement et la réalisation de notre application Web[27].

4.2.1 Langage de programmation :

JSP : Java Server Page, est une technologie pour le développement de pages Web incluant du contenu dynamique. Contrairement à une page HTML qui ne contient que du contenu statique qui reste par définition toujours le même, JSP peut changer selon l'identité du visiteur, de son navigateur Internet, de l'heure, de la configuration du système, des actions du visiteur, etc.

4.2.2 Environnement de développement :

4.2.2.1 J2EE :

Java 2 Platform J2EE est une plate-forme Java, conçu pour du mainframe, typique de l'informatique des grandes entreprises. Sun Microsystems (en collaboration avec des partenaires de l'industrie tels qu'IBM) a conçu J2EE pour simplifier le développement d'applications en environnement client léger. J2EE simplifie le développement d'applications et permet au programmeur le développement normalisé de composants modulaires réutilisables[27].

4.2.2.2 Eclipse :

Eclipse est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation.

4.2.2.3 Oracle SQL Developer :

Oracle SQL Developer est un environnement de développement intégré (EDI) multi-plateforme, fourni gratuitement par Oracle Corporation et utilisant la technologie Java (Java Development Kit). C'est un outil graphique permettant d'interroger des bases de données Oracle à l'aide du langage SQL[28].

4.2.3 Outil de Conception :

4.2.3.1 Motorbac :

MotOrBAC est un outil permettant de spécifier des politiques OrBAC. MotOrBAC utilise l'interface de programmation (API) OrBAC, une implémentation du modèle OrBAC. L'outil MotOrBAC a été développé pour concevoir et à implémenter des politiques de sécurité utilisant le modèle OrBAC. Il permet de concevoir, charger et sauvegarder des politiques de sécurité et permet de simuler des politiques concrètes. La simulation de politique peut être utilisée pour tester une politique de sécurité. Etant donné que le modèle OrBAC permet l'expression de politiques mixtes contenant à la fois des permissions et des interdictions, MotOrBAC inclut un algorithme de détection de conflits et des stratégies de résolution de conflits pour aider les utilisateurs à trouver et résoudre les conflits[29].

1. Objectif de Motorbac :

L'objectif de MotOrBAC est de centraliser dans un modèle de sécurité unique l'expression de toutes les exigences de sécurité réseau, système ou applicatives. Pour cela, MotOrBAC est basé sur le modèle OrBAC.

2. Le prototype MotOrBAC :

Le prototype MotOrBAC assure les fonctionnalités suivantes :

- Saisie d'une politique de sécurité : l'administrateur peut introduire avec MotOrBAC les différentes entités spécifiques au SI dont il gère la sécurité (organisations et sous-organisations, rôles, activités, vues et contextes) et les règles de sécurité associées.

- Simulation de la politique : MotOrBAC permet de simuler la politique en saisissant les sujets, actions et objets de l'organisation et en dérivant automatiquement la politique au niveau concret à partir de la politique organisationnelle introduite par les administrateurs. Les sujets, actions, objets sont caractérisés par des attributs.

- Vérification de la cohérence de la politique de sécurité : MotOrBAC permet de détecter les conflits au niveau concret ou abstrait de la politique de sécurité spécifiée par l'administrateur.

- Gestion des droits d'administration : MotOrBAC permet de spécifier les droits donnant à un sujet affecté à un rôle d'administration la possibilité de gérer tout ou partie d'une politique de sécurité OrBAC[29].

4.2.3.2 StarUML :

StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL. StarUML gère la plupart des diagrammes spécifiés dans la norme UML2.0[27].

4.2.4 Serveur d'application :

Apache Tomcat : est un conteneur libre de Servlet Java EE. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des Servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP (web)[27].

4.2.5 Système de gestion de base de données :

Oracle 9i : est un puissant système de Gestion de Bases de Données Relationnelles proposant, en plus du moteur de la base, de nombreux outils à l'utilisateur, au développeur et à l'administrateur.

4.3 Realisation :

Apres avoir implémenté notre modèle Orbac sur le simulateur Motorbac, un fichier a été généré avec l'extension .pof qui contient toute les politiques de notre projet.

4.3.1 Relation Habilite :

Notre implémentation sur le simulateur Motorbac pour la création des relations Habilites qui relie les sujets et les rôles, en premier lieu faut sélectionner le rôle puis les sujets concerné.

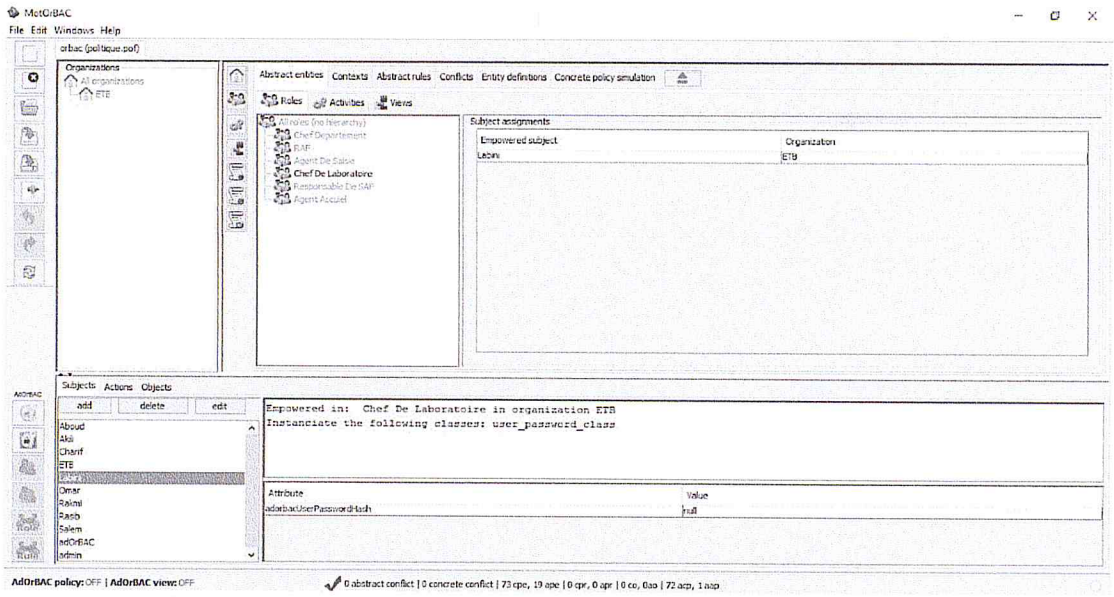


FIGURE 4.1 – Fenêtre de création de relation sujet rôle organisation.

Après avoir implémenté dans Motorbac, voici le code qui a été généré pour la relation Habilite pour le rôle Chef de département et le sujet Charif.

```
<object name="ra_ETB_Charif_Chef Departement">
    <instance_of name="role_assignment_class"/>
        <attribute name="assignment" value="Chef Departement"/>
        <attribute name="authority" value="ETB"/>
        <attribute name="assignee" value="Charif"/>
    <attribute name="grantor" value="null"/>
</object>
```

Pour les autres relations, elles sont définis de la même manière.

4.3.2 Relation Considère :

Dans cette partie, on va montrer l'interface pour implémenter la relation Considère, en premier lieu, il faut sélectionner l'action puis l'activité.

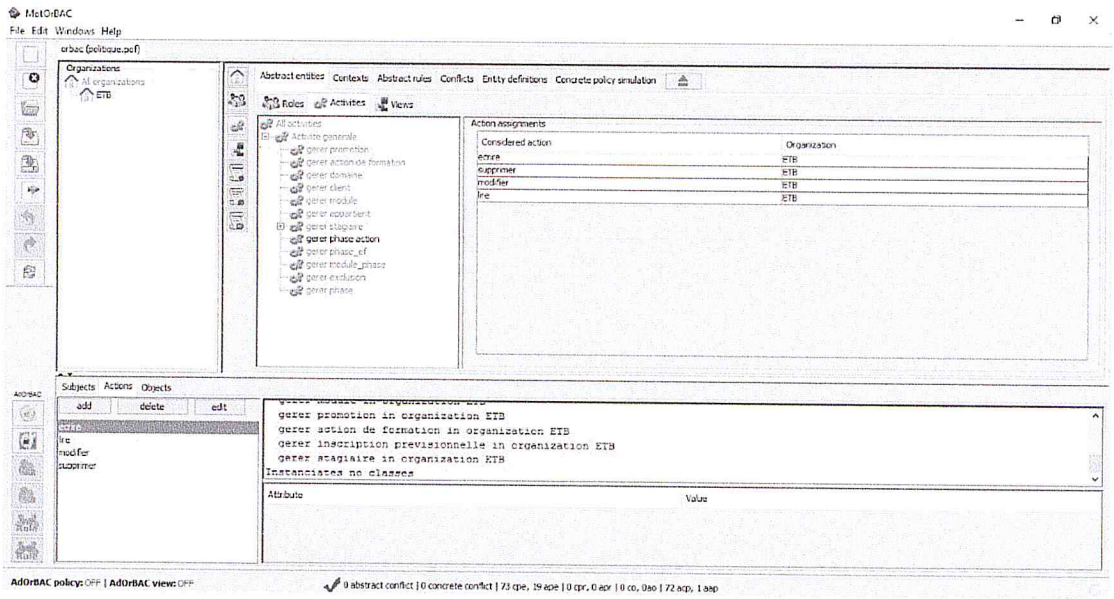


FIGURE 4.2 – Fenêtre pour la création de relation Considère.

Le code généré pour implémenter la relation Considère qui relie l'action écrire avec l'ajout d'exclusion est le suivant.

```
<object name="aa_ETB_ecrire_gerer exclusion ">
    <instance_of name="activity_assignment_class"/>
        <attribute name="assignment" value="gerer exclusion "/>
        <attribute name="authority" value="ETB"/>
    <attribute name="assignee" value="ecrire"/>
</object>
```

Pour les autres relations considère sont définis de la même manière.

4.3.3 Relation Utilise :

L'interface d'implémentation de la relation Utilise qui relie les objets et les vues, dans cette fenêtre c'est la relation qui relie l'objet AF avec la vue Action_formation, et pour les autres relation Utilise sont défini de la même manière :

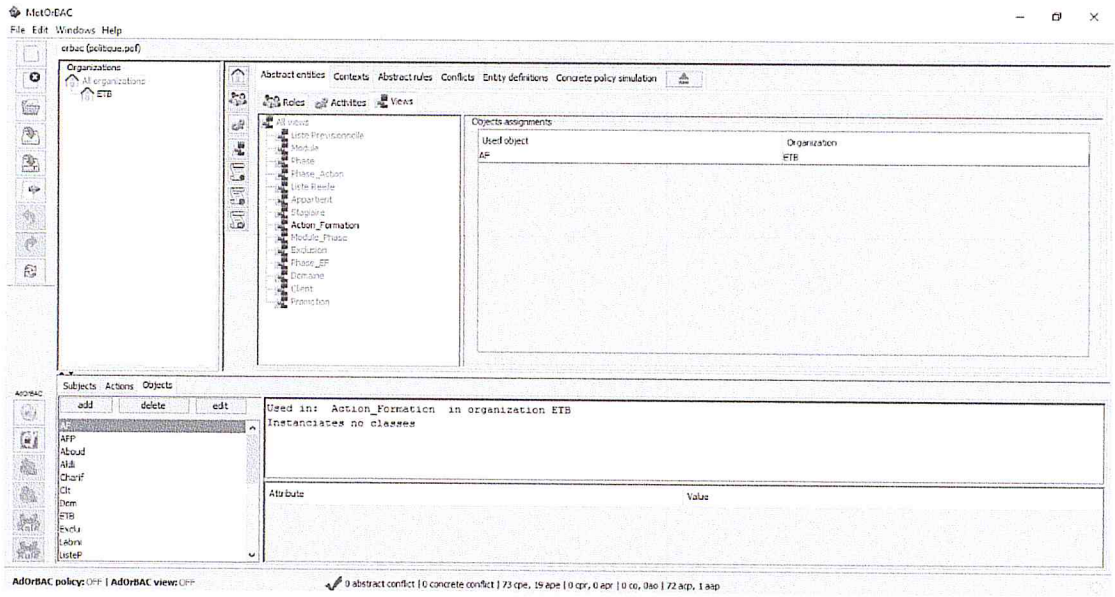


FIGURE 4.3 – Fenêtre pour la création de relation Utilise.

Le code généré pour implémenter la relation Utilise qui relie l'objet AF avec la vue formation est le suivant :

```
<object_assignment  object="AF" organization="ETB" view="Action_Formation "/>
```

4.3.4 Contexte :

Apres avoir implémenté les trois relations (Habilite, Considère, Utilise), nous allons montrer l'interface pour l'implémentation des cotextes :

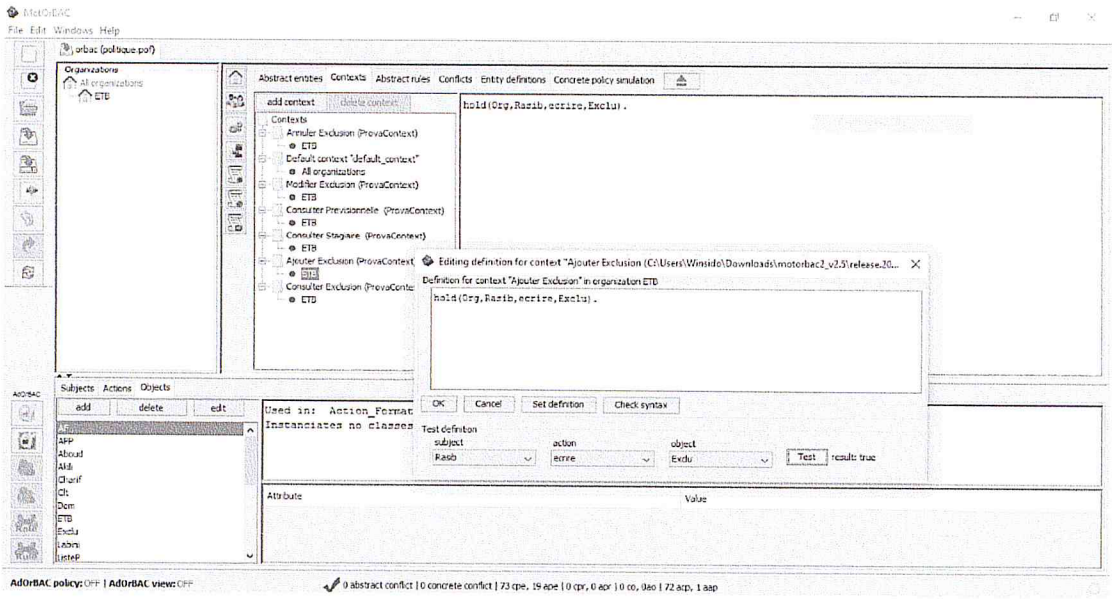


FIGURE 4.4 – Fenêtre pour la création des contextes.

Le code généré pour implémenter le contexte pour l'ajout des exclusions est le suivant :

```
<context name="Ajouter Exclusion" type="ProvaContext">
        <definition organization="ETB">hold(Org,AS,ecrire,Exclu).</definition>
</context>
```

4.3.5 Les permissions :

La fenêtre pour implémenter les permissions abstraite dans le simulateur Motorbac est la suivante :

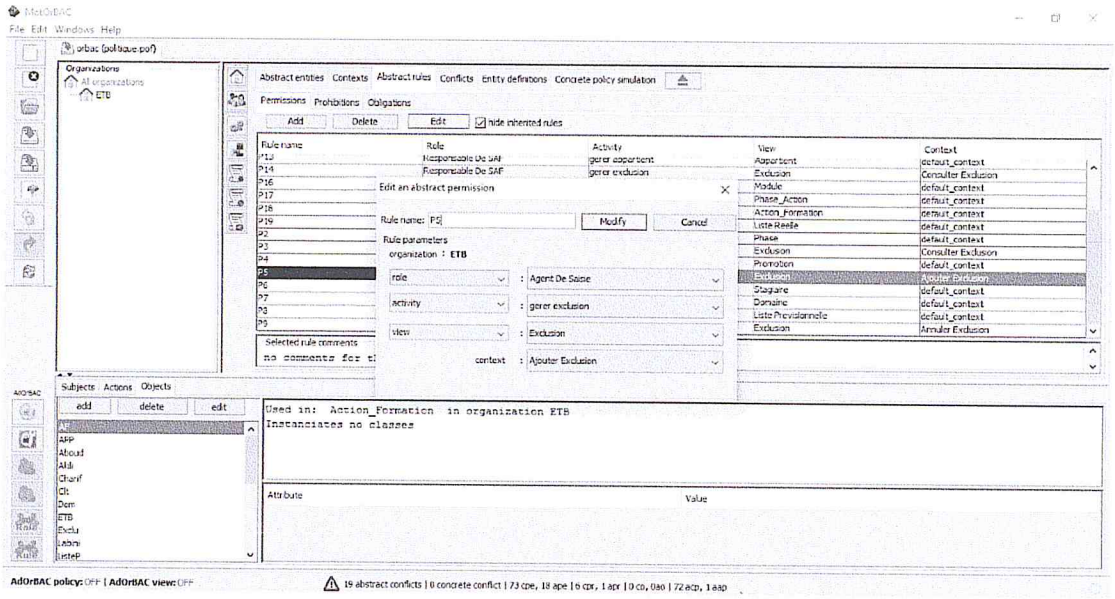


FIGURE 4.5 – Fenêtre pour la définition des permissions.

Le code généré pour implémenter la permission P5 qui permet d'ajouter une exclusion par le rôle Agent de saisie est le suivant :

```
<object name="P5">
        <instance_of name="license_class"/>
            <attribute name="authority" value="ETB"/>
                <attribute name="grantee" value="Agent De Saisie"/>
                    <attribute name="context" value="Ajouter Exclusion"/>
                    <attribute name="parentLicense" value="null"/>
                <attribute name="grantor" value="null"/>
            <attribute name="privilege" value="gerer exclusion "/>
        <attribute name="target" value="Exclusion"/>
</object>
```

4.3.6 Les interdictions :

La fenêtre pour implémenter les interdictions abstraite dans le simulateur Motorbac est la suivante :

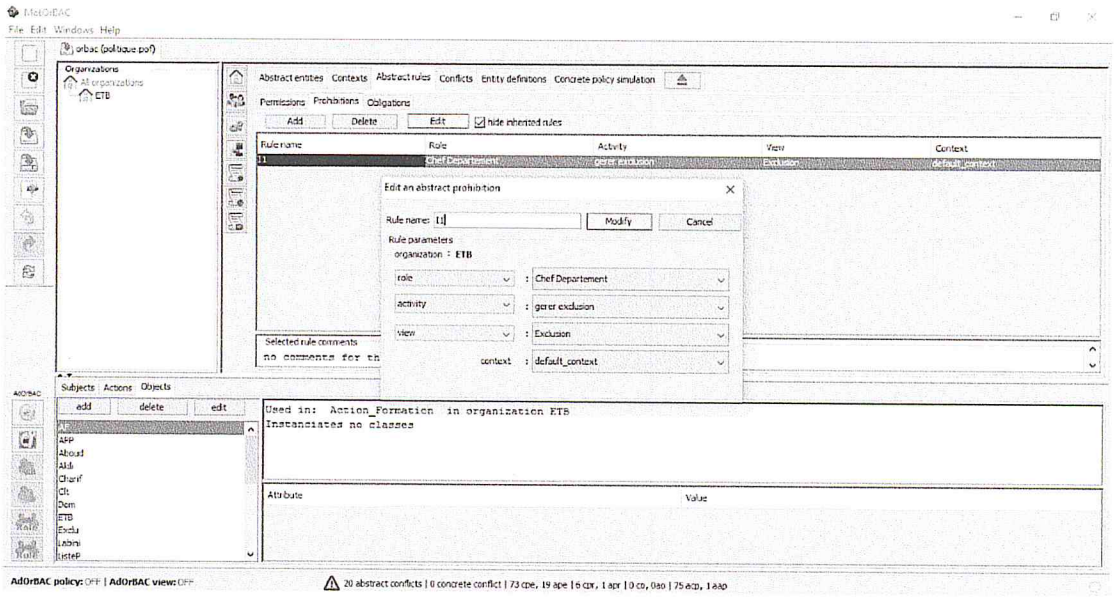


FIGURE 4.6 – Fenêtre pour la définition des interdictions.

Le code généré pour implémenter l'interdiction abstraite qui interdit l'exclusion d'un stagiaire par le rôle Agent de saisie est le suivant :

```
<object name="I1">
        <instance_of name="inhibition_class"/>
            <attribute name="authority" value="ETB"/>
                <attribute name="grantee" value="Chef Departement"/>
                <attribute name="context" value="default_context"/>
            <attribute name="privilege" value="gerer exclusion "/>
        <attribute name="target" value="Exclusion"/>
</object>
```

4.4 Injection de la politique de sécurité dans l'application :

Apres l'implémentation de la politique de sécurité sur le simulateur Motorbac et obtention du fichier « Politique.pof », on doit injecter ce denier dans notre application, où il sera une couche entre la couche contrôleur et la couche modèle.

Chaque opération faite par un sujet, doit passer par cette couche crée qui sert à vérifier les droits de chaque sujet sur les objets dans les permissions et les interdictions.

La figure ci-dessous explique l'emplacement de la couche intégré :

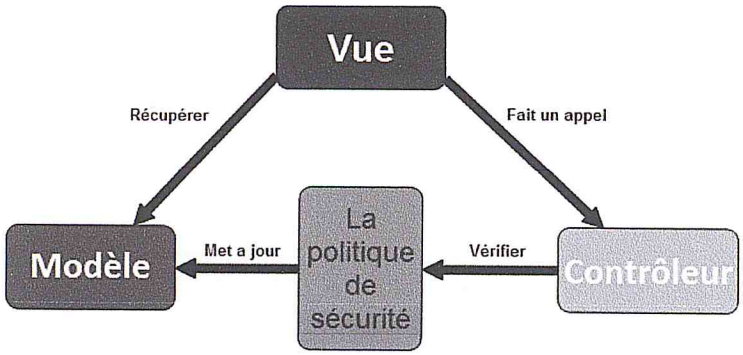


FIGURE 4.7 – L'emplacement de la couché ajouté.

4.5 Le protocole HTTPS :

Nous avons configuré l'obligation de passer par du HTTPS plutôt que par du HTTP. Les échanges HTTPS sont chiffrés de sorte qu'ils ne soient plus lisibles si on ne dispose pas des bons certificats de déchiffrement.

4.5.1 Génération du Keystore :

Un Keystore est un fichier qui va comprendre la clé privée du serveur ainsi que le certificat auto-signé. Pour le générer, on utilise la commande suivante :

Keytool.exe -genkey -alias signpaps -keyalg RSA -keystore ETB-keystore

Nous allons alors répondre à plusieurs questions afin de remplir notre Keystore :

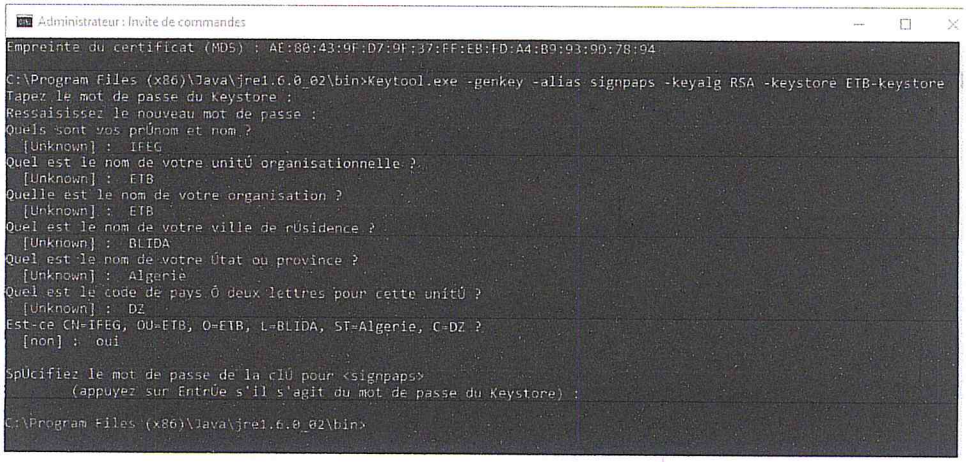


FIGURE 4.8 – Création du Keystore.

4.5.2 Création du connecteur SSL :

Après la génération du Keystore, il faut indiquer à Tomcat quel connecteur (port) utiliser pour communiquer via SSL. Par défaut, il s'agit du port 8443 mais il n'est pas activé. Donc nous allons dans notre fichier « conf/server.xml » pour modifier la configuration de notre Tomcat.

La configuration du connecteur SSL est déjà présente dans le fichier « conf/server.xml », Nous avons décommenté son paragraphe et on a rajouté le champ « keystoreFile » pour spécifier l'endroit où nous stockons notre keystore, ainsi que le champ « keystorePass » pour specifier le mot de passe du notre keystore. Le port utilisé par défaut est « 8443 ».

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
           maxThreads="150" scheme="https" secure="true"
          clientAuth="false" sslProtocol="TLS"
          keystoreFile="path/ETB-keystore"
          keystorePass="******"   />
```

FIGURE 4.9 – Création du connecteur SSL.

4.5.3 Forçage d'utilisation de SSL :

On a forcé la connexion en HTTPS pour ne plus avoir la possibilité de nous connecter en HTTP simple. Il nous fallait pour cela de modifier le fichier « web.xml » de notre application pour y rajouter ces lignes à la fin du fichier (juste avant </webapp>) :

```
<security-constraint>
      <web-resource-collection>
      <web-resource-name>HTTPSonly
      </web-resource-name>
      <url-pattern>/*</url-pattern>
      </web-resource-collection>
      <user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
      </user-data-constraint>
 </security-constraint>
```

FIGURE 4.10 – Forçage du protocole HTTPS.

L'ajout de ces lignes permet au serveur Tomcat de nous rediriger lorsque l'on arrive sur le port 8080. En réalité, dans la configuration du connecteur HTTP (port 8080), il y a une option "redirectPort" qui indique où rediriger les requêtes arrivant sur le port 8080 :

```
<Connector connectionTimeout="20000"
port="8080" protocol="HTTP/1.1"
redirectPort="8443"/>
```

FIGURE 4.11 – Redirection du port 8080 vers le port 8443.

Il faut noter que dans la configuration précédente on a forcé le SSL iniquement pour notre application Web, on peut aussi forcer le SSL pour toutes les applications web du serveur Tomcat. Il faudra alors modifier le fichier "web.xml" du Tomcat dans le dossier «Tomcat7.0/conf / » et non pas celui du notre application.

4.6 Chiffrement de mot de passe :

Dès que y a un nouveau employé dans l'école, l'administrateur doit l'ajouter dans la politique de sécurité comme un nouveau sujet, et il doit lui affecter un rôle et un mot de passe, mais ce dernier il ne doit pas être visible dans le fichier qui contient la politique de sécurité.

Pour cela la modification de mot de passe est obligatoire dès la première connexion à l'application, donc cette étape est importante pour chiffrer le mot de passe en utilisant le chiffrement RSA.

La fenêtre qui permet de modifier le mot de passe est la suivante :

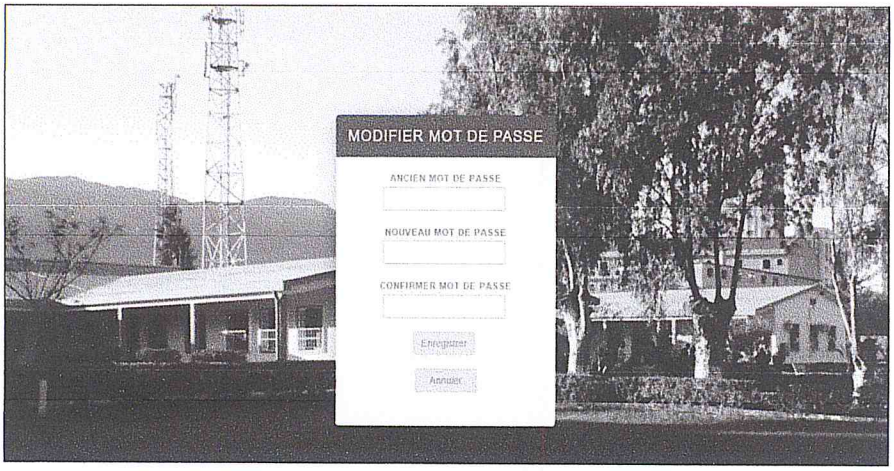


FIGURE 4.12 – Page modification de mot de passe .

Après la vérification de mot de passe (c.à.d. le nombre de caractère doit être entre 6 et 12) et la validation, en ce moment-là le cryptage se fait en utilisant l'algorithme de chiffrement RSA en basant sur la clé publique.

Apres avoir chiffré le mot de passe, ce dernier va être transférer vers le serveur, sauf que après le cryptage, le mot de passe est stocké de la manière que personne ne puisse l'utilisé pour accéder à l'application.

De même, pour qu'un utilisateur puisse se connecté, le système doit transférer le mot de passe crypté du serveur vers le navigateur et le décrypter en utilisant l'algorithme de déchiffrement RSA.

4.7 Présentation de l'application :

Dans cette partie nous allons présenter quelques interfaces de notre application web sous forme de prises d'écran, avec explication de certains éléments essentiels.

4.7.1 Page d'authentification :

À partir de la page d'authentification de l'application, l'utilisateur peut s'authentifier et aussi accéder à la page où il peut réinitialiser son mot de passe s'il oublie ce dernier.

FIGURE 4.13 – La page d'authentification de l'application.

Un utilisateur inscrit et qui a bien validé son inscription peut s'authentifier grâce à l'interface d'authentification.

4.7.2 Page d'accueil :

Lorsqu'un utilisateur se connecte à son profil, il est automatiquement redirigé vers la page d'accueil. Avant toutes actions nous nous assurons que nous avons bien un utilisateur connecté et que son profil lui permet ou non une telle action.

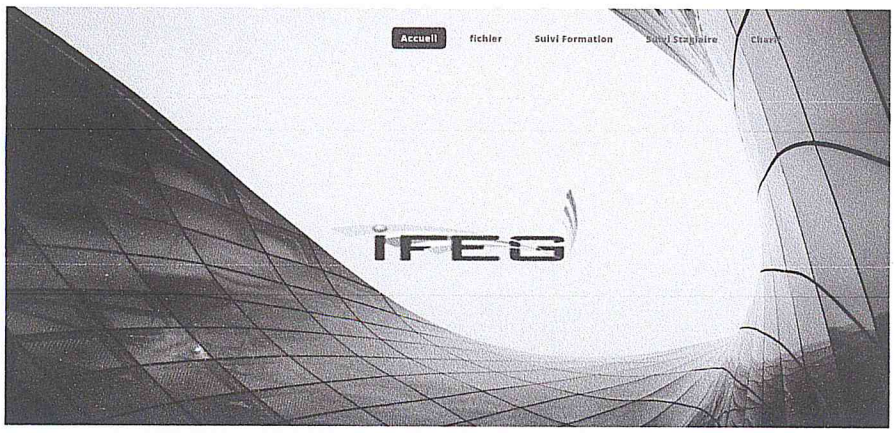


FIGURE 4.14 – La page d'accueil de l'application.

4.7.3 Page Exclusion stagiaire :

Pour exclure un stagiaire faut que le sujet possède le rôle agent de saisie ou responsable SAF, comme dans l'interface ci-dessous le sujet Akili peut exclure un stagiaire car il possède le role agent de saisie, donc quand il clique sur le bouton Ajouter y a le formulaire qui apparait pour remplir le détail de l'exclusion après avoir sélectionné le stagiaire dans le tableau à gauche.

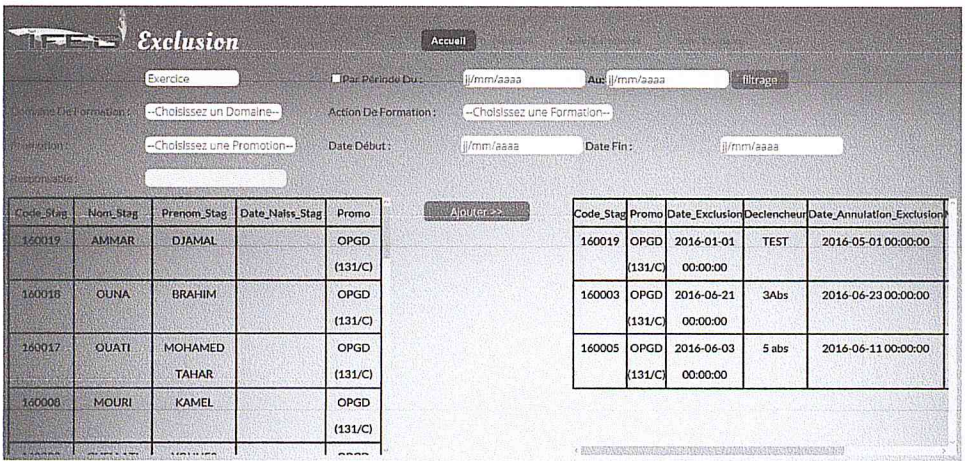


FIGURE 4.15 – Page Exclusion.

Pour annuler une exclusion d'un stagiaire, seulement les sujets qui ont le rôle Responsable SAF qui peuvent la faire, si non une fenêtre qui apparait pour informer le sujet connecté qu'il n'a pas le droit d'annuler l'exclusion.

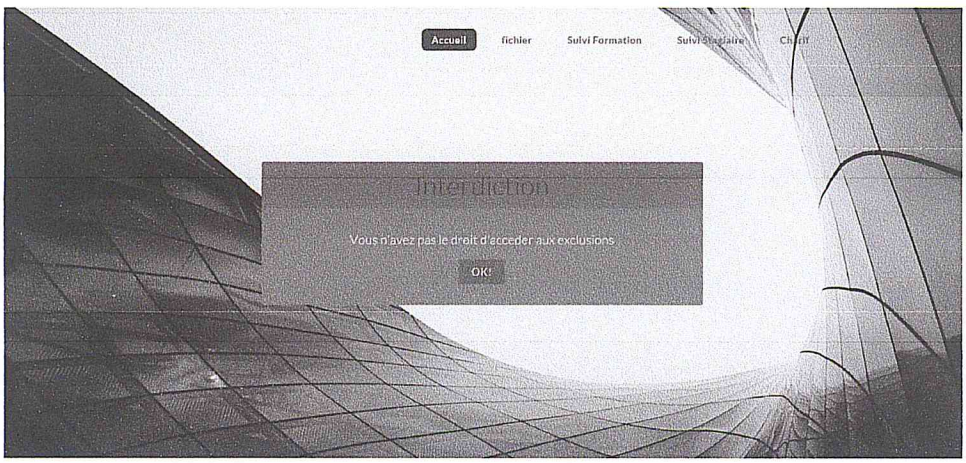


FIGURE 4.16 – Page Annulation exclusion interdit.

4.7.4 Page Ajout promotion :

La création d'une promotion se fait par le sujet qui a le rôle chef de département. Dans notre exemple c'est le sujet Charif.

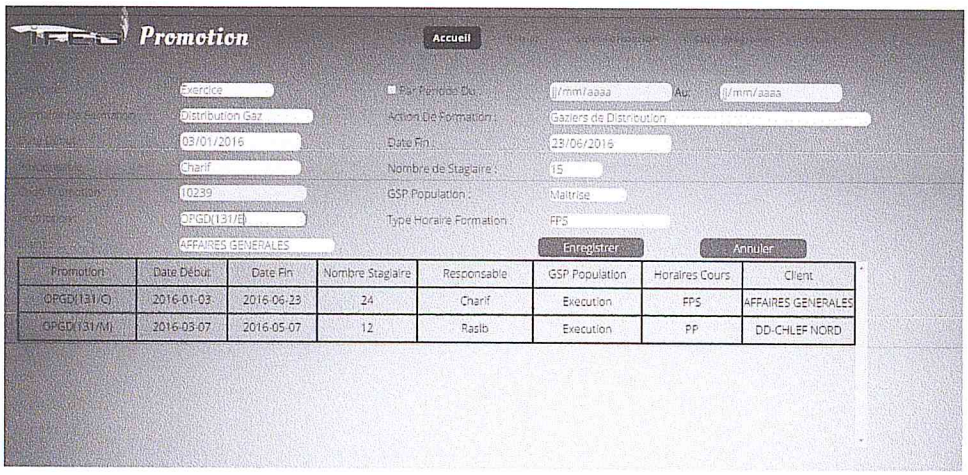


FIGURE 4.17 – Page création des promotions.

Dans la figure suivante nous allons montrer l'interdiction d'accéder a l'interface qui gère les promotions par le sujet Rasib qui a le rôle Responsable de la SAF.

FIGURE 4.18 – Page interdiction d'accéder aux promotions.

4.7.5 Page Ajout stagiaire :

L'ajout d'un stagiaire se fait par le sujet qui a le rôle Agent de saisie. Dans notre exemple c'est le sujet Aboud.

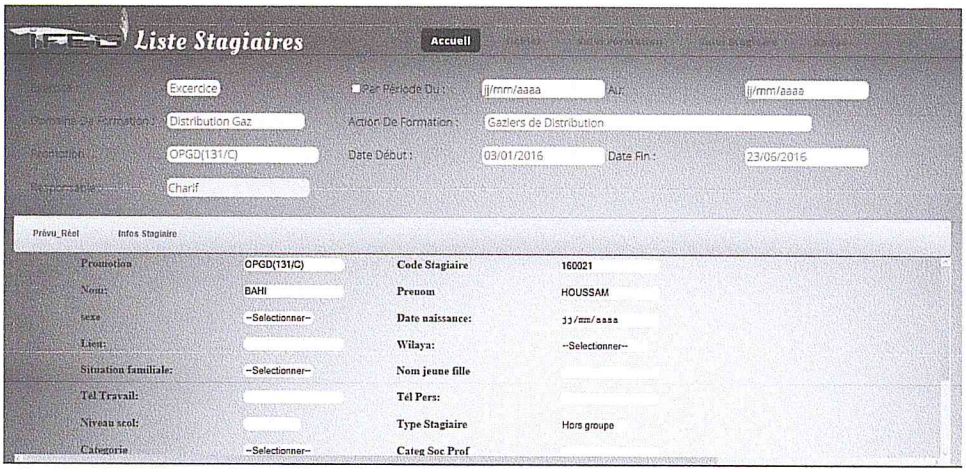


FIGURE 4.19 – Page ajout d'un stagiaire.

Dans la figure suivante nous allons montrer l'interdiction d'ajouter un stagiaire par le sujet Akili qui a le rôle Agent d'accueil.

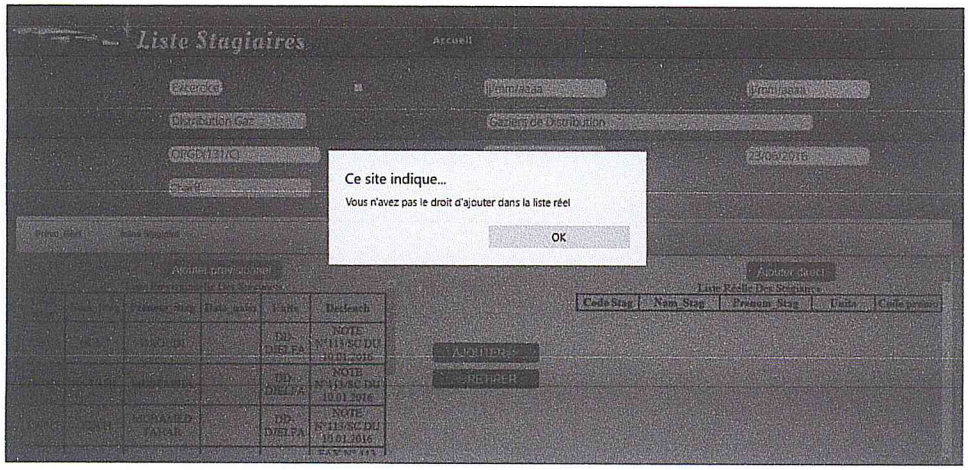


FIGURE 4.20 – Page interdiction d'ajouter un stagiaire.

4.8 Conclusion :

Dans ce chapitre, nous avons montré l'implémentation de notre politique de sécurité et l'integration de cette derniere dans le modele MVC. En premier lieu, nous avons implémenté les mécanismes de sécurité implémentés dans notre application, en suite nous avons exposé les principales interfaces réalisées dans notre application web pour clarifier les étapes d'utilisation.

CONCLUSION GÉNÉRALE ET PERSPECTIVES :

Notre projet de fin d'étude a été réalisé au profit d'ETB. Dans ce cadre, nous avons mis en place une application web sécurisé pour la gestion des formations et des stagiaires.

Pour ce faire, nous avons commencé par l'étude de l'existent pour la description des besoins de futurs système.

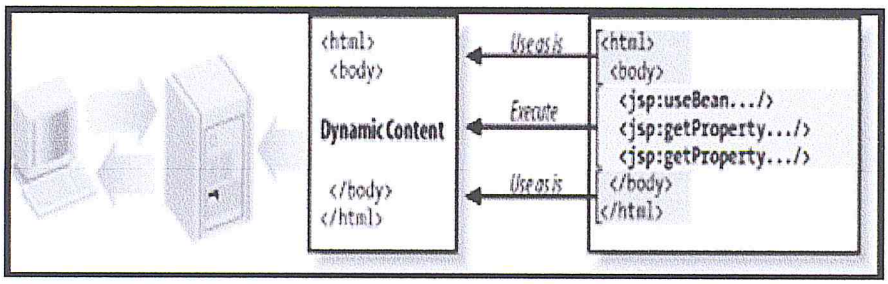
Par la suite, nous avons cité les différents modèles de sécurité existants, ce qui nous a permis d'établir une comparaison et de choisir le modèle le plus approprié. Dans la phase suivante, nous avons choisi la méthode de modélisation UML, ce dernier a été utilisé pour modéliser notre système par les différents diagrammes. Finalement nous avons implémenté une application web, dotée d'un mécanisme de sécurité, en utilisant le modèle OrBAC dans la partie conception et en respectant les besoins qui nous avons jugés les plus importants. Beaucoup de chercheurs s'intéressent à la sécurité des données dans les grandes entreprises, on espère que notre modeste travail les aidera à trouver des réponses ou à leur éviter des erreurs.

Enfin, le sens que pourrait prendre l'amélioration de cette application, se placerai dans l'ajout d'un IDS (Intrusion Détection System), afin de détecter les intrusion provenant de l'extérieur.

ANNEXES

Annexe 1. Java Server Page (JSP) :

JSP est une technologie pour le développement de pages Web incluant du contenu dynamique. Contrairement à une page HTML qui ne contient que du contenu statique qui reste par définition toujours le même, JSP peut changer selon l'identité du visiteur, de son navigateur Internet, de l'heure, de la configuration du système, des actions du visiteur, etc.
Une page JSP contient des balises standard, comme du HTML (ou du WML, XML...), comme toute page web normale. Pourtant, une page JSP contient aussi des éléments JSP spécifiques (scriptlets), permettant au serveur l'insertion dynamique de contenu (contenu de BDD, préférences du visiteur...)

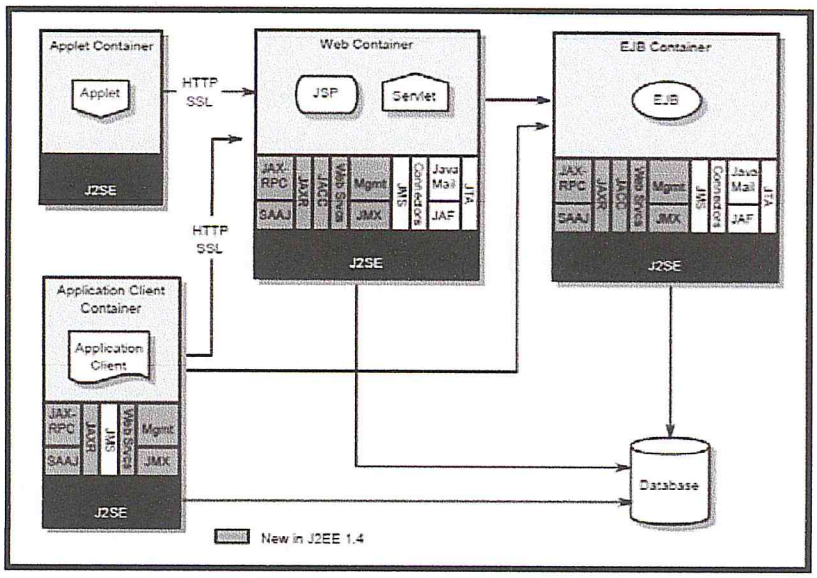


Lorsqu'un utilisateur accède à une page JSP, le serveur exécute les éléments JSP, fusionne les résultats avec les parties statiques de la page, et envoie le tout au navigateur.

Annexe 2. Java 2 Platform, Enterprise Edition (J2EE) :

J2EE est une plate-forme Java, conçu pour du mainframe, typique de l'informatique des grandes entreprises. Sun Microsystems (en collaboration avec des partenaires de l'in-

dustrie tels que IBM) a conçu J2EE pour simplifier le développement d'applications en environnement client léger. J2EE simplifie le développement d'applications et permet au programmeur le développement normalisé de composants modulaires réutilisables.
Inclus le JDK, la technologie «Write Once Run Anywhere » (portabilité), communique avec CORBA (Common Object Request Broker Architecture), le JDBC (Java Database Connectivity), permet EJB, XML, Servlet, JSP.



Annexe 3 . Algorithme de chiffrement et déchiffrement RSA :

```
class Rsa
{
  private BigInteger n, d, e;

  public Rsa(int bitlen)
  {
    SecureRandom r = new SecureRandom();
    BigInteger p = new BigInteger(bitlen / 2, 100, r);
    BigInteger q = new BigInteger(bitlen / 2, 100, r);
    n = p.multiply(q);
    BigInteger m = (p.subtract(BigInteger.ONE))
                   .multiply(q.subtract(BigInteger.ONE));
    e = new BigInteger("3");
    while(m.gcd(e).intValue() > 1) e = e.add(new BigInteger("2"));
    d = e.modInverse(m);
  }
  Public BigInteger encrypt(BigInteger message)
  {
    return message.modPow(e, n);
  }
  public BigInteger decrypt(BigInteger message)
  {
    return message.modPow(d, n);
  }
}
```

Annexe 4 . Guide d'installation de l'application :

Ce guide d'installation comporte une liste d'instructions à suivre pour la mise en place de l'application.

4.1. Serveurs necessaires :

a. Serveur d'application : Tomcat (Version 7).
b. Serveur de base de données : Oracle 9i.

4.2. Installation de la base de données :

a. Créer une nouvelle base de données, avec le nom « FPS ». b. Importer le fichier « FPS.sql » qui se trouve sur le CD de l'application pour la création des tables nécessaires.

4.3. Configuration du serveur d'application Tomcat :

a. Configuration du HTTPS :

1. Créer un certificat pour votre serveur, ou bien utiliser le fichier « ETB_keystore » qui se trouve sur le CD de l'application.

2. Configurer le fichier « conf/server.xml » du votre serveur Tomcat comme il est indiqué dans la partie «Création du connecteur SSL» de ce mémoire.

3. Importer le fichier « Politique.pof » qui contient la politique de sécurité qui se trouve sur le CD de l'application.

BIBLIOGRAPHIE

[1] P Samarati and C Vimercati, S. Access control : Policies, models, and mechanisms. *Foundations of Security Analysis and Design FOSAD,*, 2000.

[2] Lampson B. Protection. in 5th princeton symposium on information sciences and systems. 1971.

[3] ITSEC. « information technology security evaluation criteria ». [en ligne] http ://www.ssi.gouv.fr/site-documents/ITSEC/ITSEC-fr.pdf. Consulté le 29 mars 2016.

[4] F Cuppens and P Pucheral. "encyclopedie informatique". Edition Viubert.

[5] MEDJDOUB Saïda. Modèle de contrôle d'accès pour xml : "application à la protection des données personnelles". *Thése de doctorat en informatique, Université de Versailles Saint-Quentin-en-Yvelines*, 8 décembre 2005. application à la gestion de données biomédicales dans le cadre d'architectures de grilles de calcul/données,.

[6] Seitz L. Conception et mise en œuvre de mécanismes sécurises d'échange de données confidentielles. *thèse de doctorat, L'Institut National des Sciences Appliquées de Lyon*, Juillet 2005. application à la gestion de données biomédicales dans le cadre d'architectures de grilles de calcul/données,.

[7] Bell D, E and LaPadula L, J. Secure computer systems : Unied exposition and multics interpretation. Technical Report ESD -TR - 73 -306.

[8] Abou El Kalam A, El Baida R, Balbiani P, Benferhat S, Cuppens F, Deswart Y, Miège A, Saurel C, and Trouessin G. Organisation based access control. *IEEE 4th International Workshop on Policies for Distributed Systems and Networks (Policy 2003)*, 4 - 6 Juin 2003. Lake Come, Italy.

[9] Saïda MEDJDOUB. Modèle de contrôle d'accès pour xml : "application à la protection des données personnelles". *Thèse de doctorat d'université*, page 120p, 2005. Lieu de soutenance : Université de Versailles Saint-Quentin-en-Yvelines.

[10] BELLAL T. « expression d'une politique de sécurité dans un réseau social ». juin 2010. pages 12, France.

[11] D Ferraiolo, R Sandhu, and S Gavrila. "proposed nist standard for role-based access control". *ACM Transactions on Information and System Security*, Vol. 4(No.3), 2001.

[12] Gustave KOUALOROH. Audit et définition de la politique de sécurité du réseau informatique de la first bank. *Master professionnel en réseaux & applications multimédia, Mémoire de master*, page 113p, 208. Université de Yaoundé I.

[13] Voisin Guillaume. L'architecture mvc dans le développement d'un site internet. [en ligne]. http ://www.guillaumevoisin.fr/internet/larchitecture-mvc-dans-le-developpement-dun-site-internet. Consulté le 23 Avril 2016.

[14] Juliard F. Uml unified method language. *Journal Université de Bretagne Sud UFR SSI-IUP Vannes*, 2001-2002.

[15] SALMAN Ola. Création de pages web pour les branches de la faculté de génie. *Université Libanaise-Faculté de génie Branche 3*, 2013. Mémoire de projet de fin d'études.

[16] Virginie SALAS. Synthèse de lecture nfe107 urbanisation des systèmes d'information. *Serveurs d'application BEA, WebSpère, Inprise et DNA*, Janvier 2009.

[17] TRANCHANT Michaël. Java webserver tomcat, jboss, jrun, jonas. *Veille technologique dans le cadre de l'UE NFE107 Architecture et Urbanisation de Systèmes d'Informations*, Décembre 2008.

[18] TRANCHANT Michaël. Java webserver tomcat, jboss, jrun, jonas. *NFE 107*, Décembre 2008.

[19] Thawte. Mieux comprendre les certificats ssl. 27 Janvier 2011. AL100685.

[20] The PHP Group. «hashage de mots de passe sûr,». [En ligne]. http ://php.net/manual/fr/faq.passwords.php. Consulté le 12 Avril 2016.

[21] LO1c and Django20. «l'algorithme rsa,». [En ligne]. http ://openclassrooms.com/courses/l-algorithme-rsa. Consulté le 13 Avril 2016.

[22] Vayel and Carnufex Dominus. «le cryptosysteme rsa entre theorie et pratique,». [En ligne]. http ://zestedesavoir.com/tutoriels/663/la-cryptographieasymetrique-avec-rsa/915/le-cryptosysteme-rsa-entre-theorie-et-pratique/3814/un-peudhistoire-et-beaucoup-de-mathematiques, 29 Mars 2015. Consulté le 20 Avril 2016.

[23] Affringue Guillaume. «chiffrement et hash en php contre l'attaque man in the middle,». [En ligne]. http ://guillaumeaffringue.developpez.com/securite/chiffrement/ ?page=5, 25 Janvier 2007. Consulté le 21 Avril 2016.

[24] Oussama BEN ZEKRI. Conception et développement d'une application gmao biomédicale hospitalière. *Institut Supérieur des Technologies Médicales de Tunis : Université de Tunis El Manar*, page 72p, 2013.

[25] Gustave KOUALOROH. Audit et définition de la politique de sécurité du réseau informatique de la first bank. *Master professionnel en réseaux & applications multimédia : Université de Yaoundé I*, page 79p, 2008.

[26] Guermazi Emna. Application web pour la gestion de la bibliothèque. *Ingénieur en informatique, techniques web et multimédia : Sfax (ISIMS)*, page 61p, 2010.

[27] Hamza BOUHEDIR, KASSA, and Mouhammed BAGHDOUCHE. Conception et developpement d'une application web securisee pour l'evaluation de la paps. *Mémoire de fin d'études :CERIST*, page 107p, 2015.

[28] Oracle. Oracle technology network. [en ligne]. http ://www.oracle.com/technetwork/developer-tools/sql-developer/documentation/index.html. consulté le 20 mai 2016.

[29] Frederic CUPPENS, Nora CUPPENS-BOULAHIA, and Céline COMA. Motorbac : un outil d'administration et de simulation de politiques de sécurité. *Loctudy : Institut Mines-Télécom-Télécom Bretagne-UEB*, page 12p, Octobre 2008.