

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES FES
DÉPARTEMENT D'INFORMATIQUE**



Projet de Fin d'Etudes

Licence Sciences et Techniques Génie Informatique

Portail de gestion de parc informatique



Lieu de stage : Siège de la banque populaire FES-TAZA

Réalisé par :

**Ghita BENSLIMANE
Zineb JRONDI**

Encadré par :

**Pr. F. MRABTI
Pr. A. EL AMRANI**

Soutenu le 06/06/2017 devant le jury composé de :

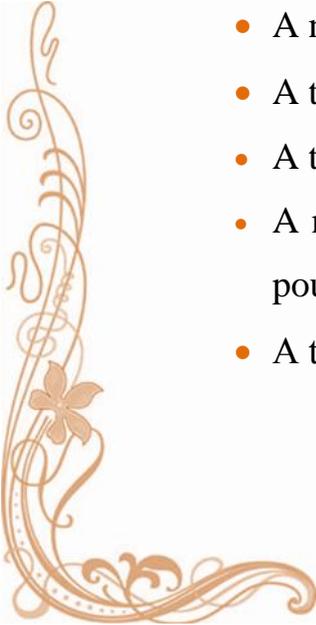
**Pr. A. ZARGHILI
Pr. I. CHAKER
Pr. F. MRABTI**

Année Universitaire 2016-2017

Dédicaces

On dédie ce travail avec respect

- A nos chers parents qui ont sacrifié tout ce qu'ils ont de cher pour nous et notre formation.
- A nos frères pour leurs encouragements.
- A tous membres du jury présent à notre soutenance.
- A tous nos amis pour leurs soutiens.
- A nos formateurs et professeurs durant notre formation, pour leurs efforts.
- A tous ceux qui auront l'occasion de lire ce rapport.



Remerciements

- A DIEU qui nous donne la force pour résister et affronter toutes les difficultés que nous obstruent dans notre vie.
- Nous tenons à exprimer nos vifs remerciements et notre profonde gratitude à Pr Fatiha MRABTI, notre encadrante à la faculté des sciences et techniques de Fès, pour ses conseils pertinents, et ses orientations judicieuses et pour ses suggestions qui ont facilité ce travail.
- Sans oublier dans nos remerciements notre encadrant Mr AMRANI Abdelali chef du Département gestion infrastructure et support informatique, pour son effort et son aide et aussi pour nous avoir intégré rapidement au sein de l'entreprise et nous avoir accordé toute sa confiance pour le temps qu'il nous a consacré toute au long de cette période.
- Ainsi que l'ensemble du personnel du Siege social de la Banque Populaire pour leur accueil sympathique et leur coopération professionnelle tout au long de ces deux mois et spécialement à Mr Badr Boujeddain.
- Nous remercions également, et nous exprimons nos profonds respects aux jurys qui ont accepté de juger notre modeste travail.
- A toute personne qui nous a encouragé et nous a aidé pour réaliser ce travail.

Table de matière :

<i>Table de matière :</i>	4
<i>Table de Figure :</i>	6
<i>Liste des abbreviations</i>	7
<i>Introduction</i>	8
<i>Chapitre 1 : Cadre du travail</i>	9
1. Présentation de l'entreprise d'accueil	10
1.1. Présentation du groupe banque populaire	10
1.1.1. Le Comité Directeur (CD).....	10
1.1.2. La Banque Centrale Populaire (BCP) :	10
1.1.3. La Banque Populaire régionale (BPR) :	10
1.2. Présentation de la BPR.....	11
1.2.1. Historique du siège.....	11
1.2.2. Organigramme de la BPR	11
1.2.3. Département gestion infrastructure et support informatique.....	12
2. Présentation du projet	12
2.1. Problématique.....	12
2.2. Solution	13
2.2.1. Cahier de charge:.....	13
2.3. Les spécifications techniques	14
<i>Chapitre 2 : Conception et modélisation de l'application</i>	15
1. Le langage de modélisation UML :	16
1.1. Diagramme des cas d'utilisation	16
1.2. Diagramme de séquence :	18
1.3. Diagramme de classe :.....	20
2. L'architecture de l'application	22
2.1. La couche de la base de données :.....	22
2.2. La couche métier :.....	22
2.3. La couche web :.....	23
2.4. La couche de présentation :.....	23
3. Modélisation de la couche de persistance et le Mapping Objet/Relationnel:	23
3.1. Modélisation de la couche web :.....	25
<i>Chapitre3 : Implementation et réalisation du projet</i>	27
1. Les technologies utilisées :	28
1.1. Java Enterprise Edition :	28
1.2. La notion de l'API :.....	28
1.2.1. L'APIServlet :.....	28
1.2.2. L'API JSP :.....	28
1.2.3. L'API JDBC :.....	29
1.2.4. JQuery :.....	29
1.2.5. L'API Java Mail :.....	29

Projet de fin d'études : Application web de Portail de gestion du parc

1.2.6. L'API JXL :.....	29
2. L'environnement du développement :.....	30
2.1. Eclipse NEON :.....	30
2.2. Apache Tomcat :	30
2.3. MySQL :.....	30
3. Réalisation de l'application :.....	30
3.1. Implémentation de la base de données :.....	30
3.2. La couche de persistance :.....	31
3.3. la couche web :.....	31
3.4. La couche présentation:.....	31
3.4.1. Page d'authentification :.....	31
3.4.2. Page d'accueil :	32
Conclusion.....	44
Webographie.....	45
Bibliographie	45

Table de Figure :

Figure 1 <i>Organigramme de la BPR</i>	11
Figure 2 <i>Diagramme des cas d'utilisation (Acteur employé)</i>	17
Figure 3 <i>Diagramme des cas d'utilisation (Acteur Administrateur)</i>	17
Figure 4 <i>Diagramme de séquence (authentification)</i>	18
Figure 5 <i>Diagramme de séquence (ajout direct d'équipement)</i>	19
Figure 6 <i>Diagramme de séquence (transfert d'équipement)</i>	19
Figure 7 <i>Diagramme de séquence (Modifier équipement)</i>	20
Figure 8 <i>Diagramme de classes qui modélise le schéma de la base de données</i>	21
Figure 9 <i>L'architecture de l'application</i>	22
Figure 10 <i>Le mapping objet relationnel.</i>	24
Figure 11 <i>Diagramme de classes de la couche de persistance</i>	25
Figure 12 <i>Diagramme de classes de la couche web</i>	26
Figure 13 <i>La page d'authentification</i>	32
Figure 14 <i>Oublier mot de passe</i>	32
Figure 15 <i>La page d'accueil Administrateur</i>	33
Figure 16 <i>La page d'accueil Utilisateur</i>	34
Figure 17 <i>gestion équipement</i>	34
Figure 18 <i>Ajout direct</i>	35
Figure 19 <i>Importer fichier Excel</i>	35
Figure 20 <i>Listage Equipement</i>	36
Figure 21 <i>Listage global</i>	36
Figure 22 <i>Modifier un équipement.</i>	37
Figure 23 <i>Listage particulier</i>	37
Figure 24 <i>Listage dans une Agence</i>	38
Figure 25 <i>Transférer équipement</i>	3839
Figure 26 <i>gestion de demande</i>	39
Figure 27 <i>historique</i>	40
Figure 28 <i>modifier mot de passe</i>	40
Figure 29 <i>Type équipement.</i>	41
Figure 30 <i>Ajouter Marque équipement.</i>	41
Figure 31 <i>Listage des agences</i>	42
Figure 32 <i>modifier fournisseur</i>	42
Figure 33 <i>Ajouter compte</i>	43

Liste des abbreviations

- **API**: Application Programming Interface
- **CSS**: Cascading Style Sheet
- **DAO**: Data Access Object
- **HTML**: Hypertext Markup language
- **Http** : Hypertext Transport Protocol
- **JDBC** : Java Database Connectivity
- **JEE** : Java Enterprise Edition
- **JSP** : Java Server Page
- **MVC 2** : Model View Control 2
- **SGBD** : Système de gestion de base de données
- **SQL**: Structured Query Language
- **UML**: Unified Modeling Language
- **XML**: Extensible Markup Language

Introduction

Considéré comme un élément fondamental d'une formation académique, la théorie demeure la base dont se fonde la compétence et le savoir, en outre la pratique reste un volet aussi important que la théorie et qui consiste à adapter la formation aux besoins du monde professionnel. C'est dans cette perspective qu'on devrait se placer comme stagiaires dans le milieu professionnel.

Nous avons choisi d'effectuer un stage d'initiation au sein de la BANQUE POPULAIRE au département gestion infrastructure et support informatique.

La Banque populaire possède un parc informatique géant, composé de centaine équipements informatiques et sa gestion pose un vrai problème.

Le parc informatique est un ensemble de matériels informatique qui se trouve au sein du siege, agence,cicursaille dans la region Fes-Taza

De ce fait, il nous a été confié un projet qui consiste à réaliser une application web concernant la gestion de ce parc des équipements informatiques qui répond à un besoin exprimé par les responsables de la BP, en utilisant des interfaces simple et facile à comprendre.

Cette application a été réalisée avec le langage JEE qui était imposé par notre encadrent de la BP par mesur de sécurité.et avec une base de donnée hébergées grâce au SGBD MySQL

Ce rapport est composé de trois chapitres, le premier est consacré à la présentation de l'entreprise d'accueil et la présentation du projet. Dans le deuxième nous décrivons la modélisation et la conception de notre application, et dans le troisième chapitre nous présentons les outils utilisés pour la réalisation de l'application ainsi des captures d'écran de l'application avec une description.

Chapitre 1 : Cadre du travail

- ✓ Présentation de l'entreprise d'accueil.
- ✓ Présentation du projet.

1. Présentation de l'entreprise d'accueil

1.1. Présentation du groupe banque populaire

1.1.1. Le Comité Directeur (CD)

Organe suprême de l'institution, le comité directeur est constitué de :

- Cinq Présidents des conseils des Banques Populaires Régionales.
- Cinq représentants du conseil de la Banque Centrale Populaire.

Ainsi le comité directeur exerce un contrôle administratif, technique et financier sur l'organisation et la gestion de la Banque Centrale Populaire et de chaque Banque Populaire Régionale. Il représente collectivement les organismes du Crédit Populaire pour faire valoir leurs droits et intérêts communs. Ainsi, le Comité Directeur a pour attributions principales de :

- Décider après accord des Banques Populaires Régionales concernées, le transfert partiel entre elles de leur actif et passif.
- Ratifier les décisions d'ouverture, de fermeture ou de transfert dans la même localité, tant au Maroc qu'à l'étranger, de filiales, de succursales, d'agences, de guichets...

1.1.2. La Banque Centrale Populaire (BCP) :

Elle est chargée d'exécuter les décisions du CD notamment à l'égard des Banques Populaires Régionales. Elle peut également effectuer directement toute opération pratiquée par les banques en vertu des dispositions de la loi bancaire. Toutefois, elle ne peut intervenir directement dans les circonscriptions territoriales où les Banques Populaires Régionales exercent leurs activités.

Par ailleurs, la BCP peut participer au capital d'une Banque Populaire Régionale sans limitation des parts, à titre provisoire et exceptionnel, lorsque la situation financière de la banque concernée le justifie. Elle peut toutefois prendre 5% des parts du capital d'une Banque Populaire Régionale ou d'un groupe de Banques Populaires à titre permanent.

1.1.3. La Banque Populaire régionale (BPR) :

9 banques de forme coopérative à capital variable à Directoire et à Conseil de Surveillance, dont le capital est détenu par plus de 430 000 clients sociétaires :

Oujda, Centre-Sud, Rabat - Kenitra, Tanger - Tétouan, Fès - Taza, Laâyoune, Marrakech - Béni Mellal, Meknès, Nador - Al Hoceima.

Projet de fin d'études : Application web de Portail de gestion du parc

Elle a pour mission toutes les opérations bancaires susceptibles de faciliter l'exercice normal de sa profession à savoir entre autre : l'escompte et le recouvrement de toutes valeurs, l'avance sur titre, sur marchandises et l'ouverture de crédit avec ou sans nantissement, recouvrement des dépôts de fonds de toute personne physiques ou morale etc.

1.2.Présentation de la BPR

1.2.1. Historique du siège

Le siège a été ouvert le 13/06/2008 par monsieur Mohammed Benchaaboune (le président du Groupe Populaire), comme étant la 700ème agence du groupe bancaire.

1.2.2. Organigramme de la BPR

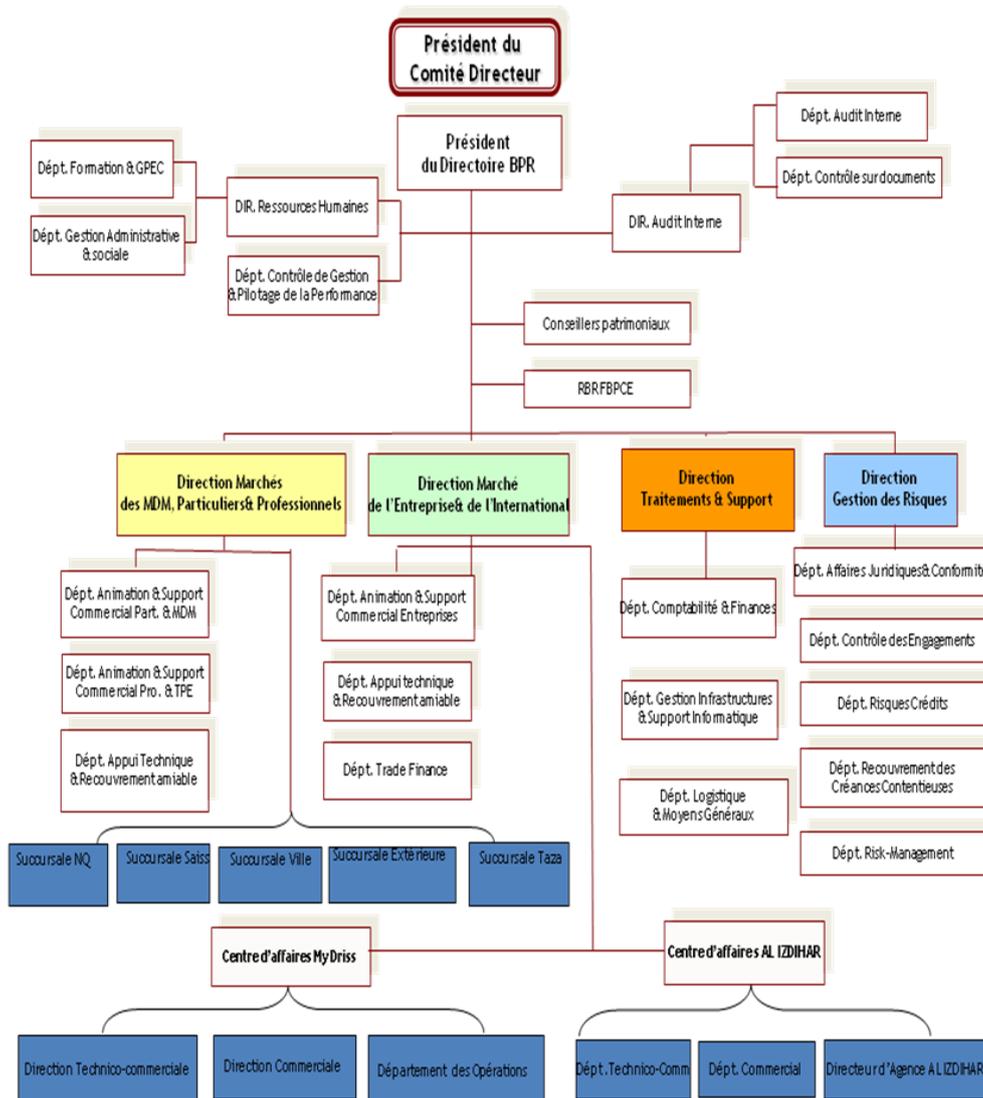


Figure 1 Organigramme de la BPR

Projet de fin d'études : Application web de Portail de gestion du parc

1.2.3. Département gestion infrastructure et support informatique

Ce département a pour but de faciliter la communication et centraliser les travaux des agences.

Le personnel de ce département assure les tâches suivantes :

- La gestion du parc informatique, de télécommunication et de téléphonie :
 - ✓ Gestion des habitations(Les droits d'accès aux applications)
 - ✓ Assistance aux utilisateurs que ce soit par accès à distance, par déplacement, par téléphone ou par messagerie.
- Télécommunication (réseau /téléphonie).
- Dotation des services en matériel informatique.
- Contrôle des présentations externes.

Le département a comme tâche principale le traitement des matériels informatiques, mais ça reste qu'une à remplir parmi d'autres tâches(Télécommunication, Contrôle des présentations externes..).

2. Présentation du projet

2.1.Problématique

Avant de plonger dans l'étude proprement dite de la solution, il est indispensable de prendre le recul et de faire un résumé des problèmes concrets existants que rencontrent nos différents acteurs.

La Banque populaire possède un parc informatique géant, composé de centaines d'équipements informatiques et sa gestion pose un vrai problème.

Actuellement, ce parc est géré avec l'outil Microsoft Excel qui n'est pas un système informatisé, il n'est pas optimal et il est un peu compliqué dans son déploiement de plus il est lent dans la recherche et le listage.

En effet, les informations de chaque équipement sont stockées dans une table Excel, cette tâche est faite par n'importe quel employé, avec l'absence d'une base de données, des tests de validité sur les informations saisies et la traçabilité de la tâche (la date, l'employé qui a fait l'ajout, la modification, le transfert et la suppression), chose qui est importante pour l'historique.

2.2.Solution

Afin d'y remédier, on a pensé à réaliser une application web pour résoudre les problèmes rencontrés dans la gestion de ce parc, en utilisant des interfaces simple et facile à comprendre.

Cette application a été réalisée avec le langage JEE et avec une base de donnée hébergées grâce au SGBD MySQL ayant comme objectif d'avoir une stratégie informatique dans la totalité du siège, d'automatiser le parc, de faciliter la tâche aux employés et de résoudre le problème de sécurité.

2.2.1. Cahier de charge:

Tout projet doit être codifié dans un cahier des charges, il est si important. Il s'agit d'un document décrivant la façon la plus précise possible, avec un vocabulaire simple, les besoins auxquels le maître d'oeuvre doit répondre. Le cahier de charge de notre projet est décrit comme suit :

Le parc informatique est un ensemble de matériels informatique qui se trouve au sein du siège, agence et, circuits. Les acteurs de l'application, ont la possibilité de gérer ce parc de la région Fes-Taza.

➤ Utilisateur:

L'utilisateur est un employé simple qui n'a pas de responsabilité s'occupe des tâches suivantes :

- Ajouter les équipements informatiques de deux manières, soit d'importer des fichiers (Excel...) à la base de donnée, soit de saisir le numéro de série et remplir le formulaire.
- Lister les équipements dans la base de données. Il y a 3 façons de listage des équipements: un listage de tous les équipements « listage globale », un listage d'un équipement en précisant les champs à voir « listage particulier » et un listage des équipements dans une Agence.
- Modifier chaque équipement.
- Déplacer les équipements à une autre agence facilement dans la base de données.
- Consulter l'historique en affichant toutes les actions (ajout, modification, transfert) avec la date de l'action et son acteur.

➤ Administrateur :

L'administrateur est le responsable du service gestion infrastructure et support informatique qui est Mr Amrni notre encadreur de la BP.

En plus des tâches de l'utilisateur, il s'occupe aussi des Gestions suivantes :

Projet de fin d'études : Application web de Portail de gestion du parc

- Ajouter, Lister, Modifier, Supprimer une agence
- Ajouter, Lister, Modifier, Supprimer un compte
- Ajouter, Lister, Modifier, Supprimer une marque
- Ajouter, Lister, Modifier, Supprimer un fournisseur
- Ajouter, Lister, Modifier, Supprimer un type d'équipement

L'application génère une page d'authentification comme première étapes, cette page est accessible à l'administrateur et aux différents utilisateurs pour assurer la sécurité de l'application.

Pour se connecter l'utilisateur doit donner son identifiant et son mot de passe. Si les informations saisies sont correctes ainsi que le compte est déjà créé et activé par l'administrateur Le système affiche la page d'accueil permettent l'accès aux fonctionnalités relatives au profil de l'utilisateur connecté : (administrateur ou utilisateur). Dans le cas contraire il y aura un échec de l'authentification.

2.3. Les spécifications techniques

L'application doit être développée en respectant les spécifications techniques ci-dessous :

- La modélisation du schéma de la base de données et de l'application doit être effectuée avec le langage de modélisation UML.
- L'architecture de l'application doit respecter le modèle MVC 2.
- Serveur de base de données : MySQL.
- Couche web : SERVLET, JSP et JQuery.

**Chapitre 2 : Conception et
modélisation de l'application**

Projet de fin d'études : Application web de Portail de gestion du parc

Dans ce chapitre, nous allons présenter le langage choisi pour la modélisation, les étapes de la modélisation, puis nous allons présenter l'architecture de l'application qui est une architecture en couches et la conception de chaque couche.

1. Le langage de modélisation UML :

Pour modéliser les fonctionnalités de notre application et représenter son architecture ainsi que les interactions entre ses différents composants, nous avons choisi le langage UML.

UML (en anglais Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel dans le cadre de la « conception orientée objet ».

Ce langage définit plusieurs diagrammes qui servent à visualiser un système sous différentes perspectives.

Parmi ces diagrammes on cite :

- diagramme de classes
- diagramme de cas d'utilisation
- diagramme de séquences
- Et bien d'autres.

Dans notre conception, on a réalisé des diagrammes des cas d'utilisation qui représentent l'ensemble des cas d'utilisations de l'application et d'acteurs et leurs relations, des diagrammes de séquence qui représentent les interactions entre les acteurs et le système selon un ordre chronologique, et des diagrammes de classes pour les différentes couches de l'application.

Pour la réalisation de nos diagrammes UML on a utilisé une version d'évaluation du logiciel Enterprise Architect qui est un logiciel de modélisation. Il permet de créer les différents diagrammes d'UML, il est simple d'utilisation, personnalisable et doté d'une interface intuitive.

1.1. Diagramme des cas d'utilisation

Les diagrammes des cas d'utilisation suivants représentent les différentes fonctionnalités offertes par notre application et les différents acteurs de cette dernière :

- Acteur : employé

Projet de fin d'études : Application web de Portail de gestion du parc

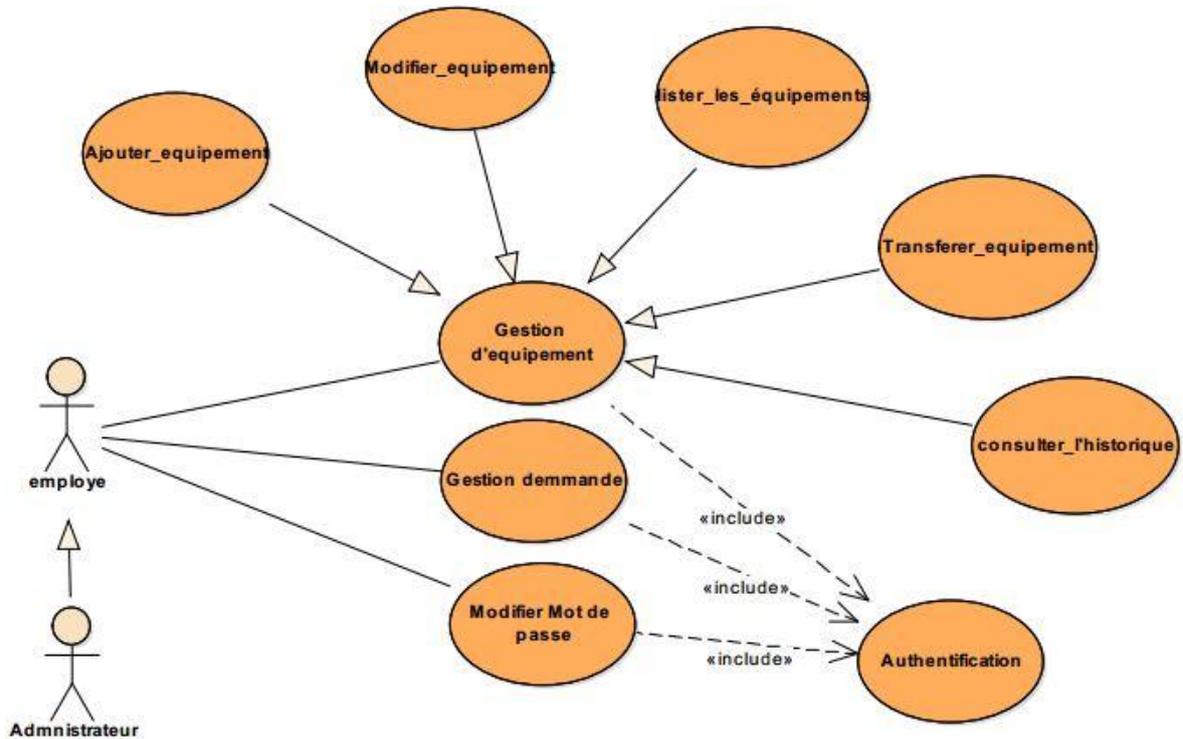


Figure 2 Diagramme des cas d'utilisation (Acteur employé)

- Acteur : Administrateur

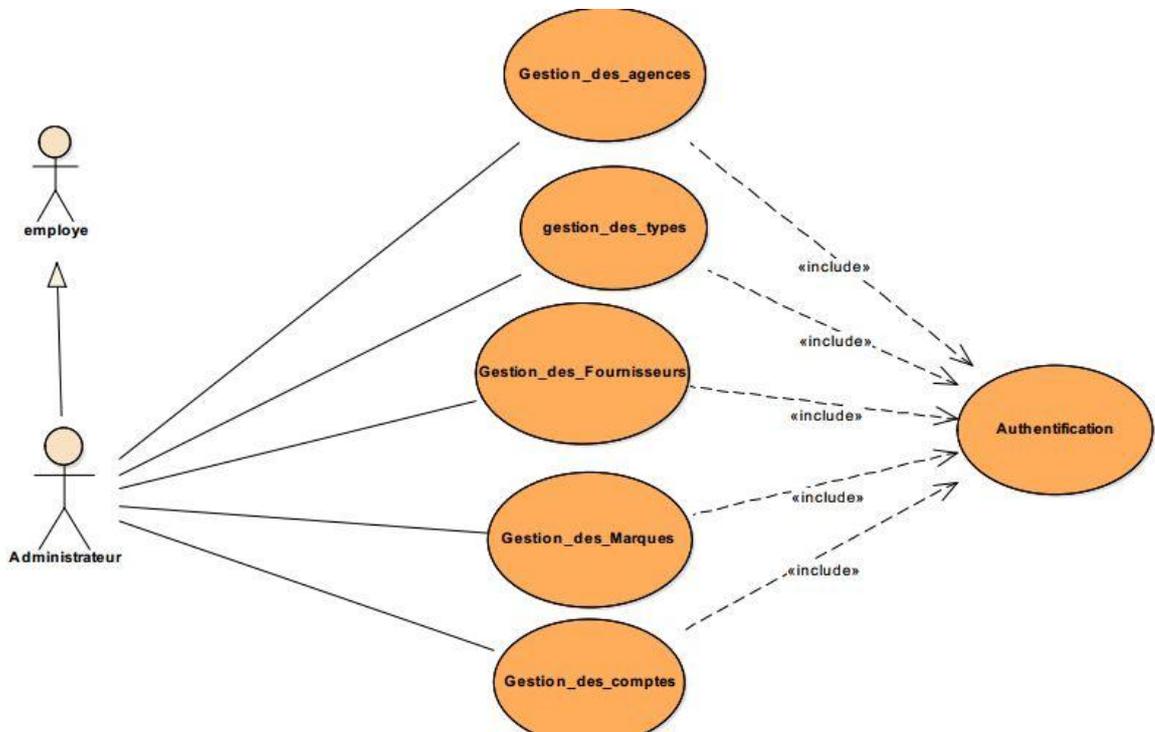


Figure 3 Diagramme des cas d'utilisation (Acteur Administrateur)

1.2. Diagramme de séquence :

Les diagrammes de séquence suivants représentent les interactions entre les acteurs et le système selon un ordre chronologique :

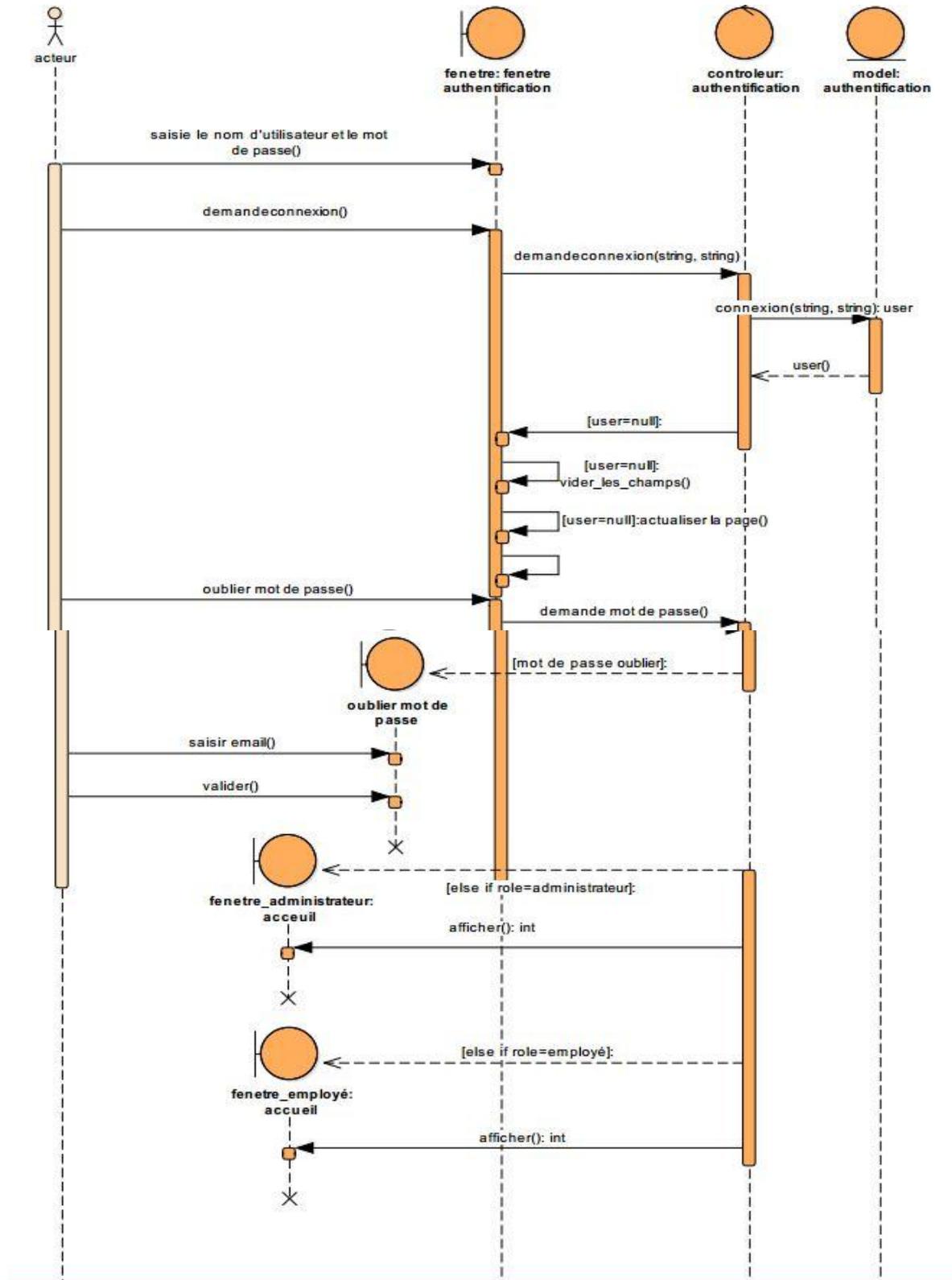


Figure 4 Diagramme de séquence (authentification)

Projet de fin d'études : Application web de Portail de gestion du parc

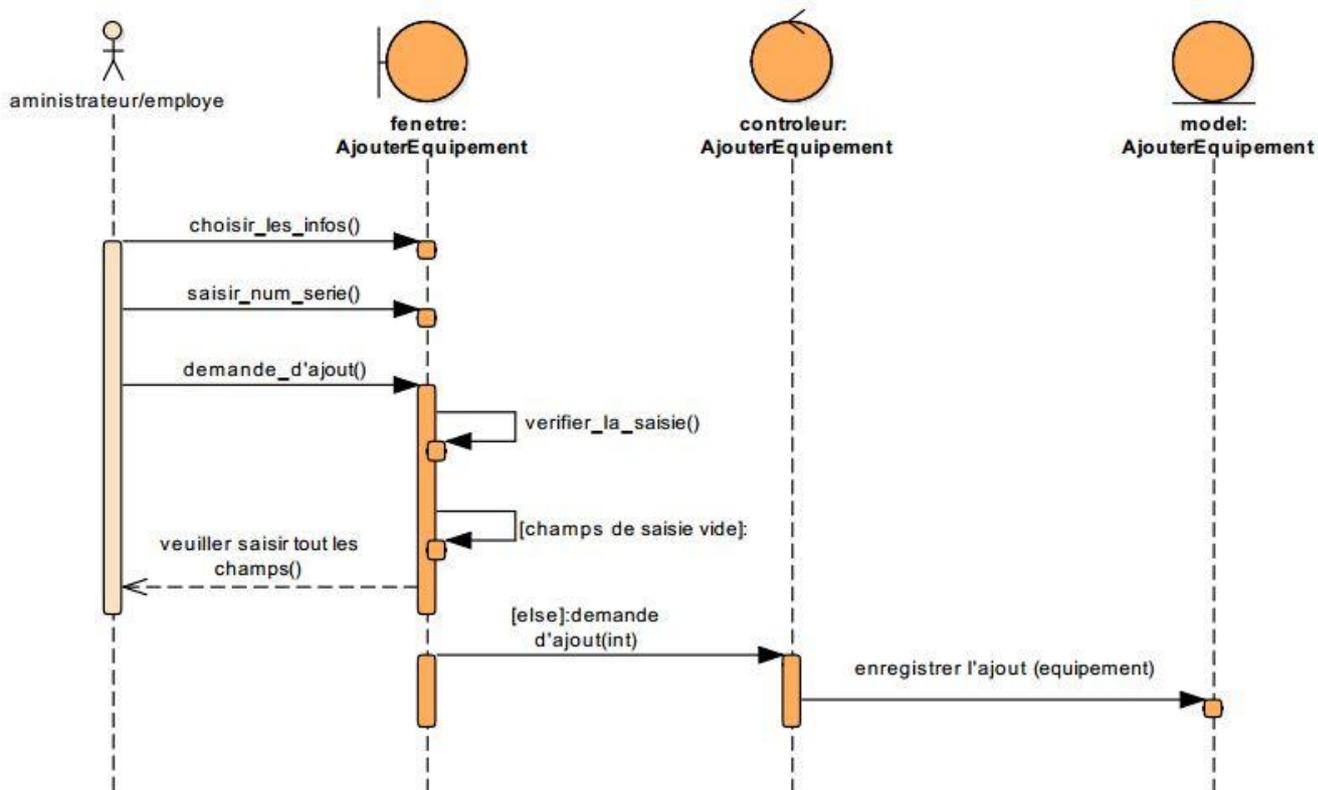


Figure 5 Diagramme de séquence (ajout direct d'équipement)

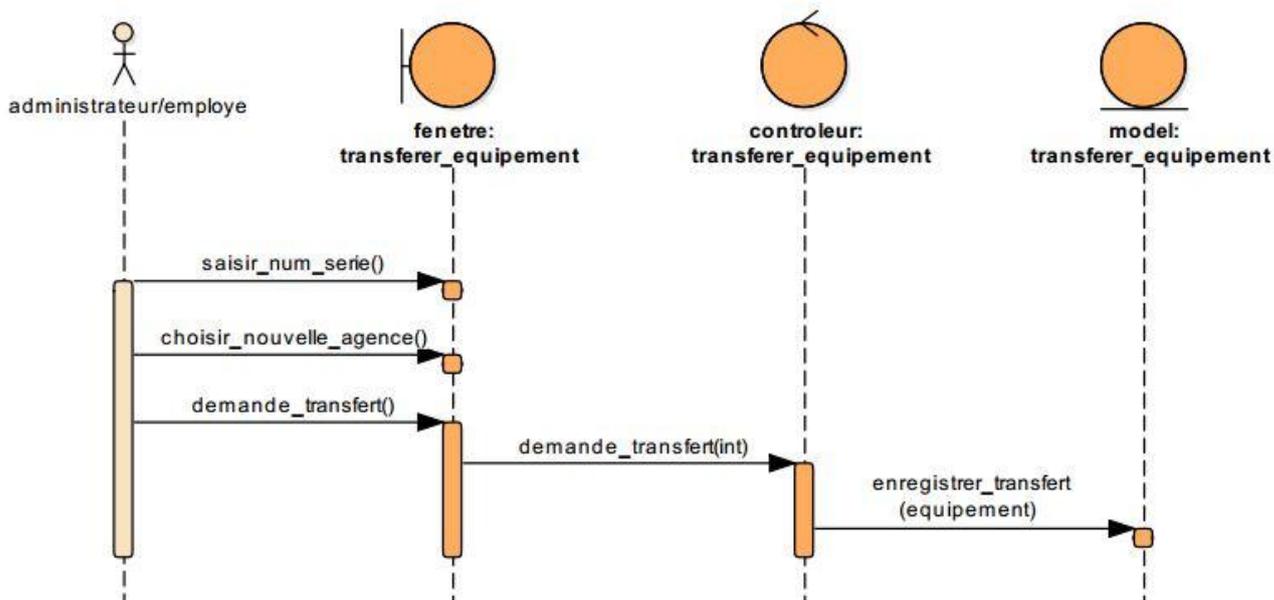


Figure 6 Diagramme de séquence (transfert d'équipement)

Projet de fin d'études : Application web de Portail de gestion du parc

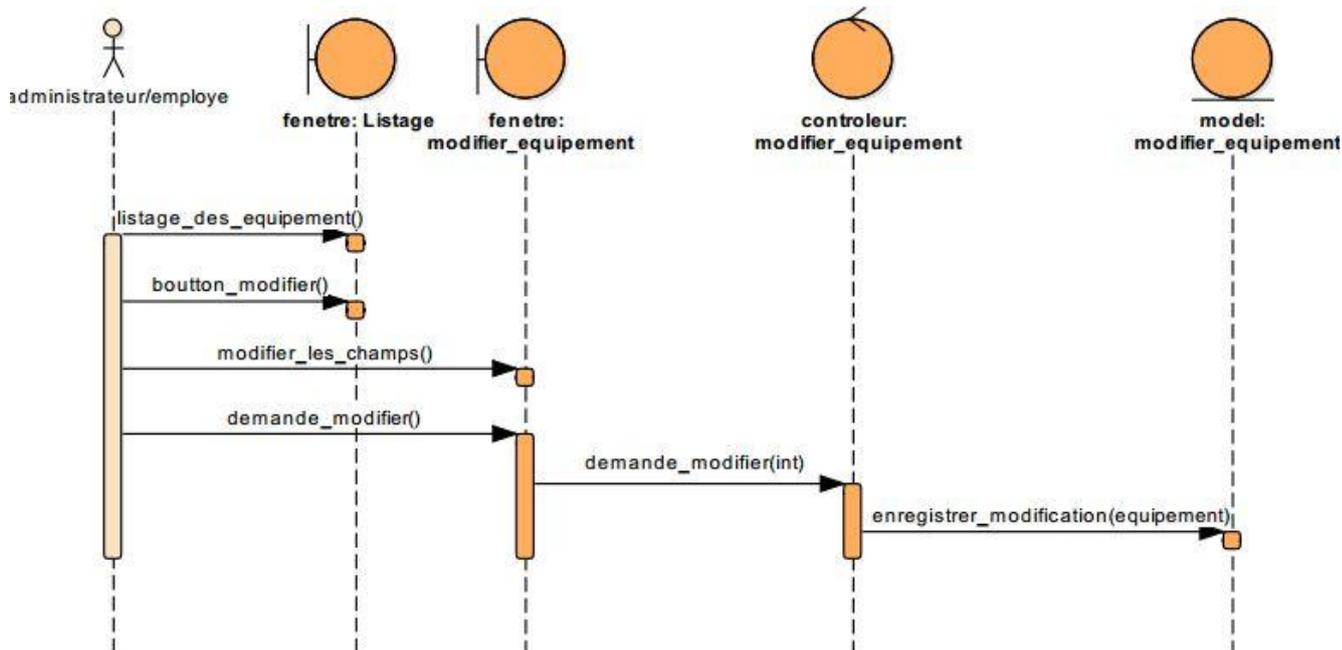


Figure 7 Diagramme de séquence (Modifier équipement)

1.3. Diagramme de classe :

Le diagramme de classes ci-dessous représente le schéma de la base de données de notre application.

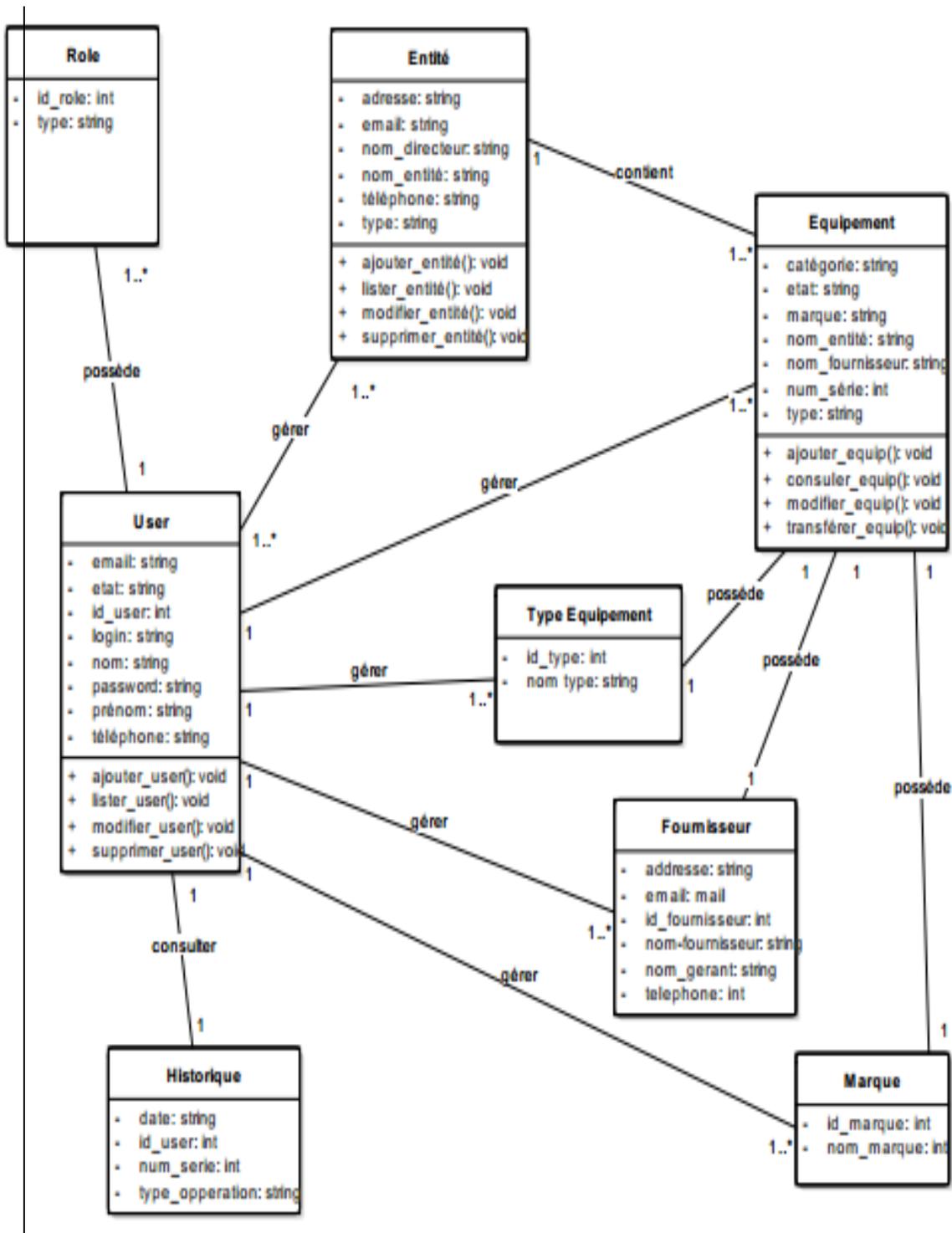


Figure 8 Diagramme de classes qui modélise le schéma de la base de données

2. L'architecture de l'application

Notre application est organisée en quatre couches (tiers, niveaux) comme suit :

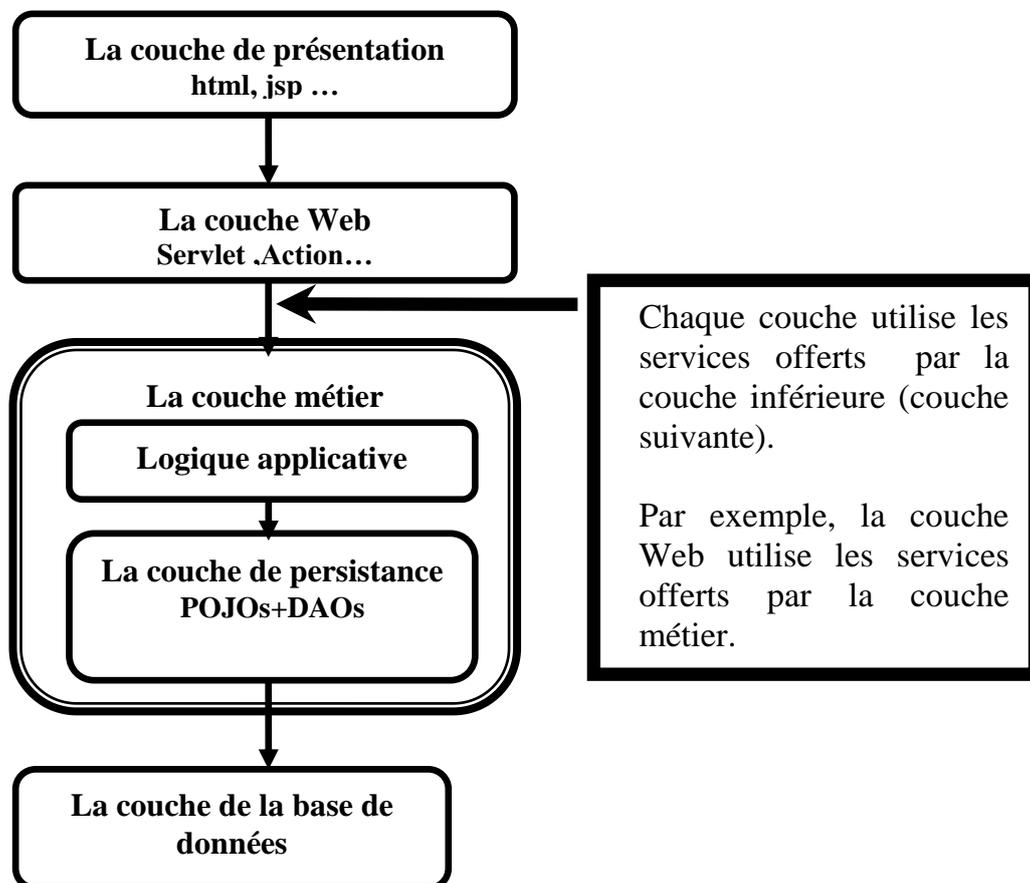


Figure 9 L'architecture de l'application

2.1. La couche de la base de données :

Cette couche représente la base de données relationnelle de l'application gérée par le SGBDR.

2.2. La couche métier :

La couche métier implémente les algorithmes "métier" de l'application. Cette couche est indépendante de toute forme d'interface avec l'utilisateur. Ainsi elle doit être utilisable aussi bien avec une interface console, une interface web (navigateur) (client léger), une interface Swing (client riche).

C'est généralement la couche la plus stable de l'architecture. Elle ne change pas si on change l'interface utilisateur ou la façon d'accéder aux données nécessaires au fonctionnement

Projet de fin d'études : Application web de Portail de gestion du parc

de l'application .Ainsi, changer l'implémentation de cette couche n'a aucune incidence sur les couches supérieures.

Cette couche représente le Modèle **M** de l'architecture **MVC2** et contient deux parties :

- La couche de persistance :

Elle s'appelle aussi la couche d'accès aux données, elle permet d'effectuer la correspondance (mapping) Objet/Relationnel entre les objets java et les tables de la base de données. Ce mapping permet d'assurer la transformation d'objets java vers la base de données et vice et versa que cela soit pour des lectures ou des mises à jour (création, modification ou suppression).

- La logique applicative

la logique spécifique de l'application, qui doit être développée sans se préoccuper de savoir d'où viennent les données qu'on lui donne, ni où vont les résultats qu'elle produit.

2.3.La couche web :

Cette couche contient le contrôleur **C** de **MVC2**, qui reçoit les requêtes des clients (requêtes **HTTP**), les traite et les transmet à la couche métier (le modèle) chargée de traiter les données, ensuite il dirige les résultats vers la vue (couche de présentation) qui seront envoyés au client. Tous les composants web et métier, tel une **Servlet**, ou une **JSP** peut jouer ce rôle. Toutefois, les **Servlet** sont des composants dont la structure est la plus adaptée car la **Servlet** est conçue pour recevoir les requêtes des clients et leur retourner une réponse.

2.4.La couche de présentation :

Une fois la requête du client est traitée, la **Servlet** détermine ce qu'on appelle « laVue »qui doit être employée pour présenter à l'utilisateur des données fournies par les couches inférieures. Elle a aussi pour rôle de fournir des données provenant de l'utilisateur (données des formulaires).

Dans notre cas les pages **JSP** jouent le rôle des vues, parce qu'elles sont parfaitement adaptées à la présentation puisque ces dernières accueillent de façon naturelle le balisage **HTML**

3. Modélisation de la couche de persistance et le Mapping Objet/Relationnel:

La persistance est la notion qui traite de l'écriture de données sur un support informatique. Pour sa part, le mapping objet-relationnel **ORM** désigne l'interaction transparente entre le cœur d'une application, modélisé en conception orientée objet, et une base de données relationnelle. Le principe de l'**ORM** consiste à faire une correspondance

Projet de fin d'études : Application web de Portail de gestion du parc

entre les objets manipulés en java et les tables de la base de données d'une part, et entre les attributs de l'objet et les champs de la table correspondante d'autre part.

Pour l'exemple, nous allons voir en quoi consiste le mapping d'une table `équipement` avec un objet "Equipment" :

La table équipement

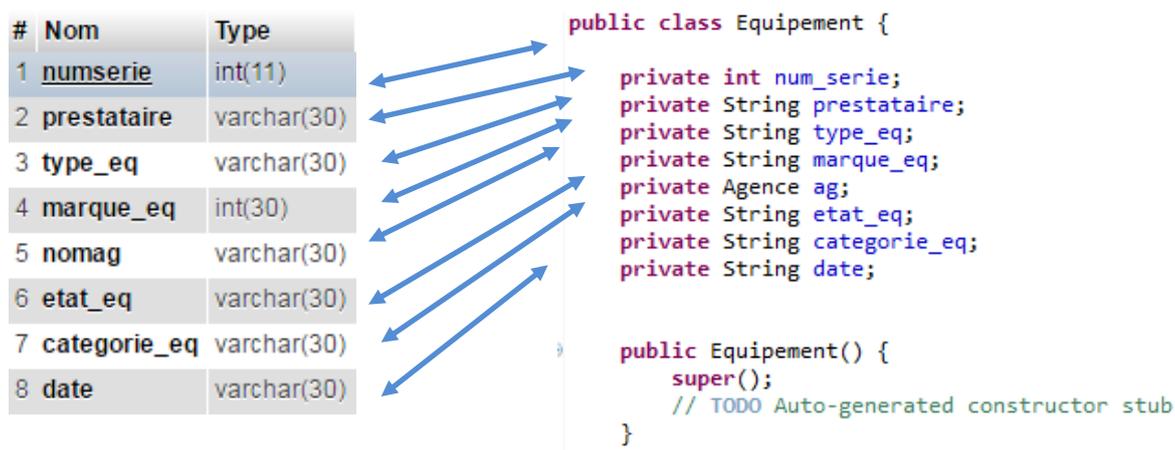


Figure 10 Le mapping objet relationnel.

Cette technique de conception nécessite un genre d'objet qui s'appelle POJO, et l'utilisation du design pattern DAO.

- Les objets POJO :

« POJO » c'est l'abréviation de « Plain Old Java Object », ce sont des objets java codés avec les accesseurs : Getters et Setters pour tous les attributs de la classe, les méthodes d'ajout et de suppression pour les objets ayant des listes d'objets.

- Le design pattern DAO :

Ce pattern assure la séparation entre le traitement de données et l'accès aux données. Il s'agit de faire en sorte qu'un type d'objet se charge de récupérer les données dans la base et qu'un autre type d'objet (souvent des POJO) soit utilisé pour manipuler ces données.

Suivant ce principe on propose la création d'une classe DAO par classe POJO. Chaque classe DAO contient les méthodes de liaison avec la base de données, parfois appelées **CRUD** (pour *Create*, *Retrieve*, *Update*, *Delete*), et pour cette raison, on propose que ces classes héritent toutes d'une superclasse générique DAO<T> (T correspond à la classe fille) dans laquelle sont déclarés toutes les méthodes communes entre les classes DAO.

Projet de fin d'études : Application web de Portail de gestion du parc

En voici le diagramme de classe correspondant:

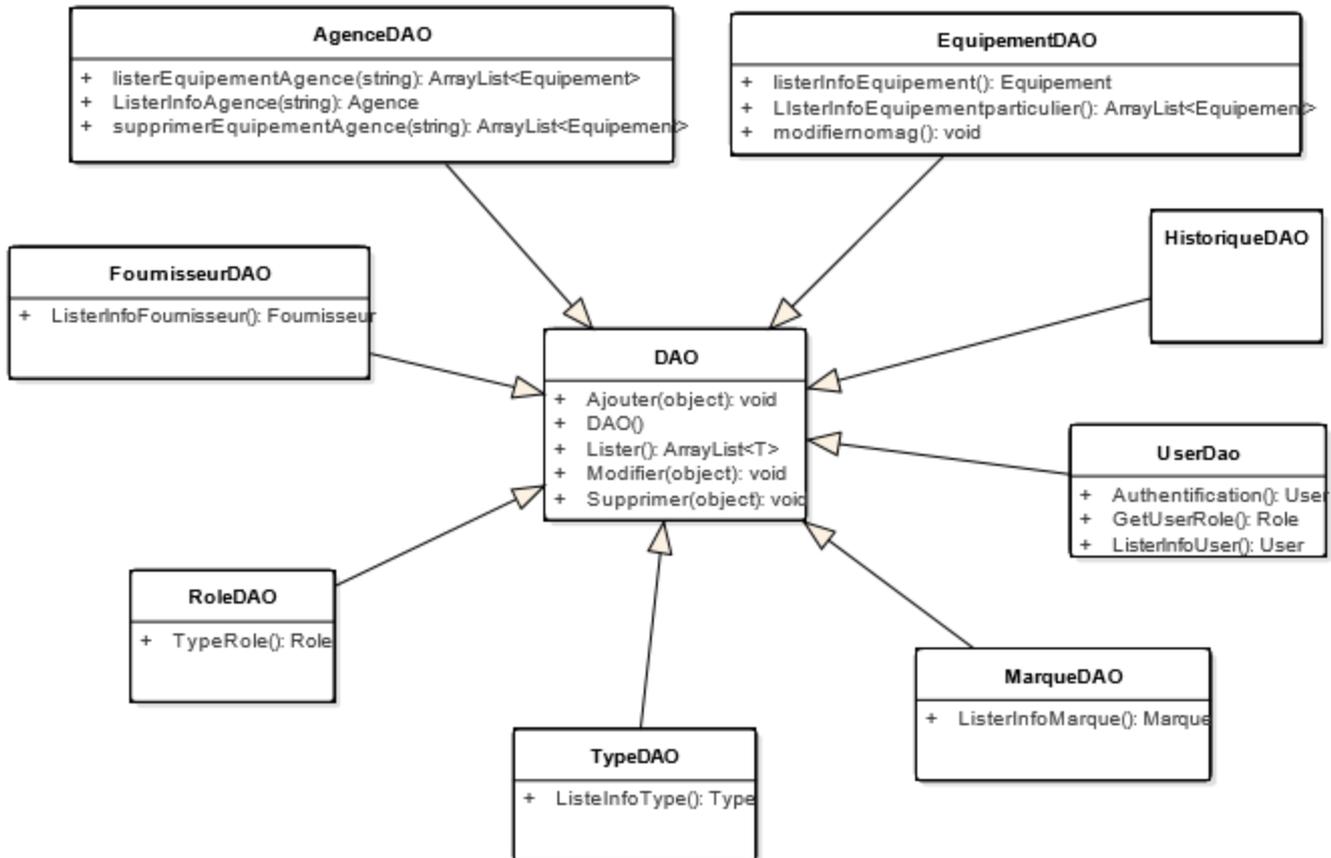


Figure 11 Diagramme de classes de la couche de persistance

3.1. Modélisation de la couche web :

Cette couche contient :

- Le contrôleur qui est le cœur de l'application, toutes les demandes du client transitent par lui. C'est une servlet nommée **ActionServlet**.
- Un ensemble des classes d'actions représentant des événements déclenchés par les utilisateurs de l'application. Ces classes héritent d'une classe abstraite nommée Action qui contient une méthode *exécute* () chargée de traiter la demande et il retourne le nom de la vue à envoyer à l'utilisateur. Le nom de chaque classe se compose du nom de l'action (AjouterEmploye par exemple) concaténé avec le mot « Action » (AjouterEmployeAction).

Projet de fin d'études : Application web de Portail de gestion du parc

Le diagramme de classes correspond à cette couche :

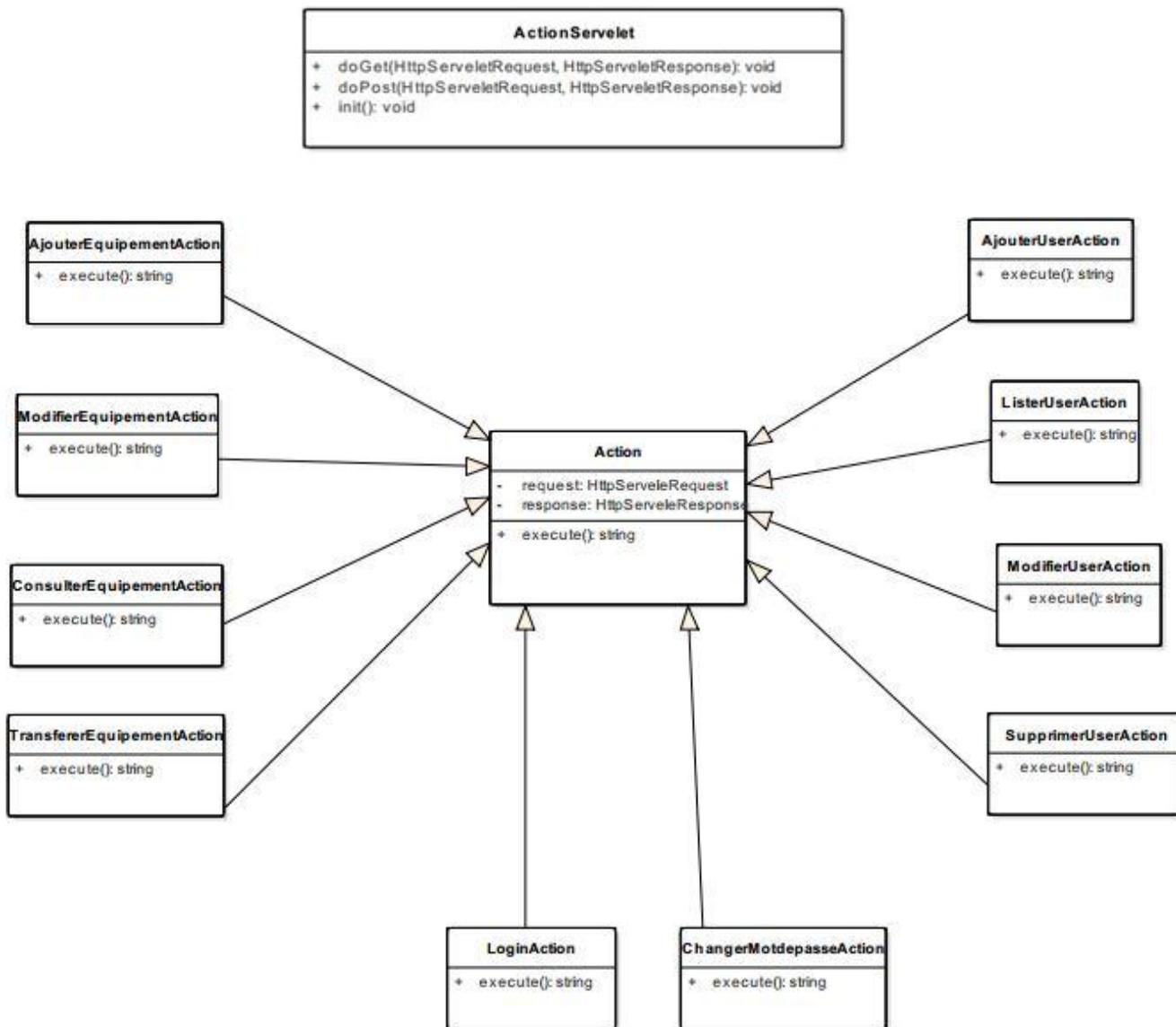


Figure 12 Diagramme de classes de la couche web

**Chapitre3 : Implementation et
réalisation du projet**

Projet de fin d'études : Application web de Portail de gestion du parc

Dans ce chapitre, nous allons décrire les étapes de l'implémentation de l'application, en précisant les choix technologiques et techniques qu' nous avons choisi. Ensuite nous allons présenter les différentes fonctionnalités offertes par l'application par quelques captures d'écrans des interfaces.

1. Les technologies utilisées :

1.1. Java Enterprise Edition :

Java Enterprise Edition, ou JEE, est une spécification pour la technique Java d'Oracle (X-Sun). Plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux. Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au Framework Java standard (JSE, Java Standard Edition) afin de faciliter la création d'applications réparties en général et les applications web en particulier.

1.2. La notion de l'API :

Une **interface de programmation** (Application Programming Interface ou *API*) est un ensemble de méthodes, classes ou interfaces mises à disposition des programmeurs informatiques par une bibliothèque logicielle, un système d'exploitation ou un service. Une interface en tant que telle est quelque chose d'abstrait ; les composants réalisant celle-ci étant des mises en œuvre (ou implémentation).

Dans notre projet on a utilisé les API JEE suivantes :

- Servlet.
- Java Server Page (JSP).
- JDBC
- JQuery
- JavaMail
- API JXL

1.2.1. L'APIServlet :

La servlet est une application Java fonctionnant du côté serveur au même titre que les langages de script côté serveur tels qu'ASP ou bien PHP. Les servlets permettent donc de gérer des requêtes HTTP et de fournir au client une réponse HTTP dynamique (donc de créer des pages web dynamiques).

Dans notre application, une servlet principale jouera le rôle de contrôleur, chargé de déclencher des actions spécifiques demandées par l'utilisateur.

1.2.2. L'API JSP :

Projet de fin d'études : Application web de Portail de gestion du parc

Les JSPs (ou les Java Server Pages) sont une extension standard Java définie au-dessus des extensions servlet. Le propos des JSP est de simplifier la création et la gestion des pages Web dynamiques.

Les JSP permettent de mélanger le code HTML d'une page Web et du code Java dans le même document, et d'assurer la séparation entre la représentation des données et le traitement. Le code Java est entouré de tags spéciaux qui indiquent au conteneur JSP qu'il doit utiliser le code pour générer une partie de la page. L'avantage que procurent les JSP est de maintenir un seul document qui est à la fois la page HTML et le code Java qui la gère.

1.2.3. L'API JDBC :

La technologie JDBC (Java DataBase Connectivity) est une API fournie avec Java (depuis sa version 1.1) permettant de se connecter à des bases de données, c'est-à-dire que JDBC constitue un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données (SGBD).

L'API JDBC a été développée de telle façon à permettre à un programme de se connecter à n'importe quelle base de données en utilisant la même syntaxe, c'est-à-dire que l'API JDBC est indépendante du SGBD.

De plus, JDBC bénéficie des avantages de Java, dont la portabilité du code, ce qui lui vaut en plus d'être indépendant de la base de données et d'être indépendant de la plate-forme sur laquelle elle s'exécute.

1.2.4. JQuery :

jQuery est une bibliothèque Javascript libre qui porte sur l'interaction entre JavaScript (comprenant AJAX) et HTML, et a pour but de simplifier des commandes communes de Javascript. Elle contient notamment les fonctionnalités suivantes :

- Parcours et modification du DOM.
- Événements.
- Effets et animations (fenêtres pop-up modales, transitions animés entre les pages).
- Manipulations des feuilles de styles en cascade.
- Ajax.

1.2.5. L'API Java Mail :

L'API Java Mail fournit un cadre indépendant de la plate-forme et indépendant du protocole pour créer des applications de messagerie. L'API Java Mail est disponible en option pour être utilisée avec la plate-forme Java SE et également incluse dans la plate-forme Java EE.

1.2.6. L'API JXL :

Projet de fin d'études : Application web de Portail de gestion du parc

L'API JXLest l'API la plus utilisée pour l'exécution des tests basés sur les données Selenium , qui permet aux utilisateurs de lire, écrire, créer et modifier des feuilles dans un classeur Excel (.xls) au moment de l'exécution. Il n'est pas pris en charge pour le format .xlsx

2. L'environnement du développement :

2.1.Eclipse NEON :

Eclipse NEON est la onzième version simultanée de la Fondation Eclipse, disponible le 22 juin 2016. Elle comprend le contenu de quatre-vingt-cinq projets Eclipse.

2.2.Apache Tomcat :

Tomcat est un conteneur de Servlet JEE, Tomcat est désormais un projet principal de la fondation Apache. C'est un conteneur de Servlet JEE qui implémente la spécification des Servlets et des JSP de Sun Microsystems. Tomcat est en fait chargé de compiler les pages JSP avec Jasper pour en faire des Servlets (une servlet étant une application Java qui permet de générer dynamiquement des données au sein d'un serveur http). Généralement, ces données sont présentées sous forme de page HTML coté client.

2.3.MySQL :

Pour implémenter notre base de données on a choisi le serveur de base de données MySQL pour les raisons suivantes :

- Sa licence libre.
- Sa haute performance.
- Sa fiabilité et sa simplicité d'utilisation.

3. Réalisation de l'application :

3.1.Implémentation de la base de données :

Pour modéliser notre base de données on a utilisé le modèle objet, mais pour stocker les données on a utilisé le modèle relationnel, donc il faut passer d'un modèle à l'autre, ce passage doit respecter certains règles de traduction :

- Chaque classe devient une table.
- Chaque attribut devient un champ.
- Et il y a d'autres règles consternent la traduction de différents types d'associations.

3.2. La couche de persistance :

Cette couche implémente le concept d'Objet/Relationnelle Mapping (*ORM*) qui permet d'automatiser la persistance des objets java dans les tables d'une base de données relationnelle, ainsi que son contraire, récupérer les données des tables directement dans des objets java.

Tout d'abord, il est nécessaire de construire des objets java (POJO) dont les membres seront porteurs des informations à stocker. Chaque membre doit posséder un getter et un setter. On obtient ces classes d'objets par le passage du diagramme de classes montré dans la Figure 7 au langage Java. La deuxième étape consiste à mapper nos entités dans les tables de la base de données. Ensuite on crée une classe DAO par classe POJO. (EquipementDAO, AgenceDAO,...).

Ces classes héritent tous d'une superclasse générique DAO<T> (T correspond à la classe fille) dans laquelle sont déclarées toutes les méthodes communes entre les classes DAO.

3.3. la couche web :

Nous avons développé une Servlet unique (architecture de type MVC-2) qui réceptionne toutes les requêtes et qui, suivant l'URI demandée, transmet les paramètres **request** et **response** à un objet de type Action qui s'occupe d'effectuer le traitement à l'aide de la méthode *execute* ()(interaction avec la couche métier pour récupérer les données nécessaires a la création des JSP ou pour récupérer les données venants de l'utilisateur pour les stocker dans la base de données) et de lui renvoyer le nom de la jsp à envoyer à l'utilisateur.

Les classes d'actions héritent de la classe abstraite **Action**. Le nom de chaque classe se compose du nom de l'action qui est extrait de l'URI (AjouterEquipementpar exemple) concaténé avec le mot « Action » (AjouterEquipementAction).

3.4. La couche présentation:

La couche présentation est composée de tout le coté html de l'application, c'est-à-dire les Jsp, les feuilles de style CSS, le JavaScript, ainsi que des contrôles nécessaires à charger le modèle de chaque page à afficher.

Dans ce qui suit on va donner une présentation visuelle de différentes fonctionnalités de l'application par quelques captures d'écrans avec des commentaires.

Notre application comporte 2 volets (un employé simple ou un administrateur).

3.4.1. Page d'authentification :

Projet de fin d'études : Application web de Portail de gestion du parc

Cette page offre la possibilité de s'authentifier par un login et un mot de passe. Ci-dessous, une capture de la page d'authentification :



Figure 13 La page d'authentification

Remarque:

Si un acteur oublie son mot de passe, il aura la possibilité de le récupérer en cliquant sur « Oublier mot de passe ». ce dernier reçoit un email qui affiche son mot de passe. Ci-dessous, une capture de la page oublier mot de passe:



Figure 14 Oublier mot de passe

3.4.2. Page d'accueil :

Projet de fin d'études : Application web de Portail de gestion du parc

Après l'authentification, on accède à la page d'accueil qui contient un menu composé de sous-menu, chaque sous-menu est composé d'un groupe de liens vers toutes les actions offertes. Ci-dessous, une capture de la page d'accueil de l'administrateur:



Figure 15 La page d'accueil Administrateur

La seule différence entre le compte de type administrateur et le compte de type utilisateur est le menu :

Menu de l'administrateur contient huit options :

- Gestion Equipement
- Gestion Agence
- Gestion Compte
- Gestion Fournisseur
- Historique
- Icône notification
- Modifier mot de passe
- Déconnexion

Menu de l'utilisateur contient seulement cinq options :

- Gestion Equipement
- Gestion Demande
- Historique
- Modifier mot de passe
- Déconnexion

Ci-dessous, une capture de la page d'accueil de l'utilisateur:

Projet de fin d'études : Application web de Portail de gestion du parc

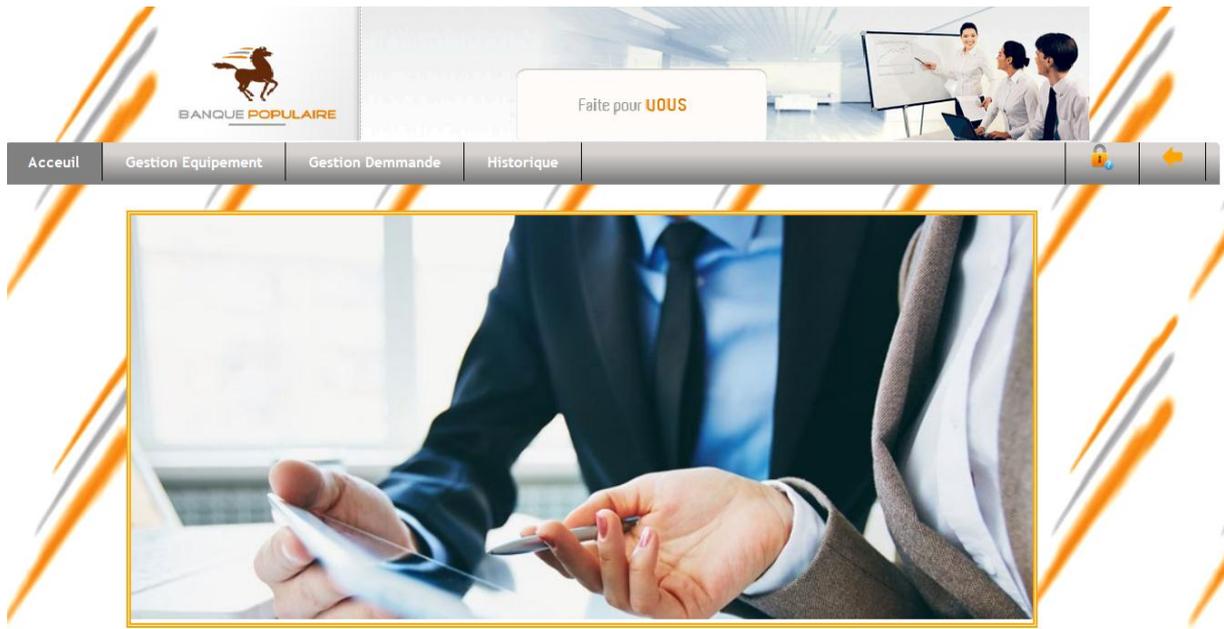


Figure 16 La page d'accueil Utilisateur

➤ Volet 1: Employé simple :

Ce volet permet aux employés simples (qui n'ont pas de responsabilité) après l'authentification d'effectuer les tâches suivantes :

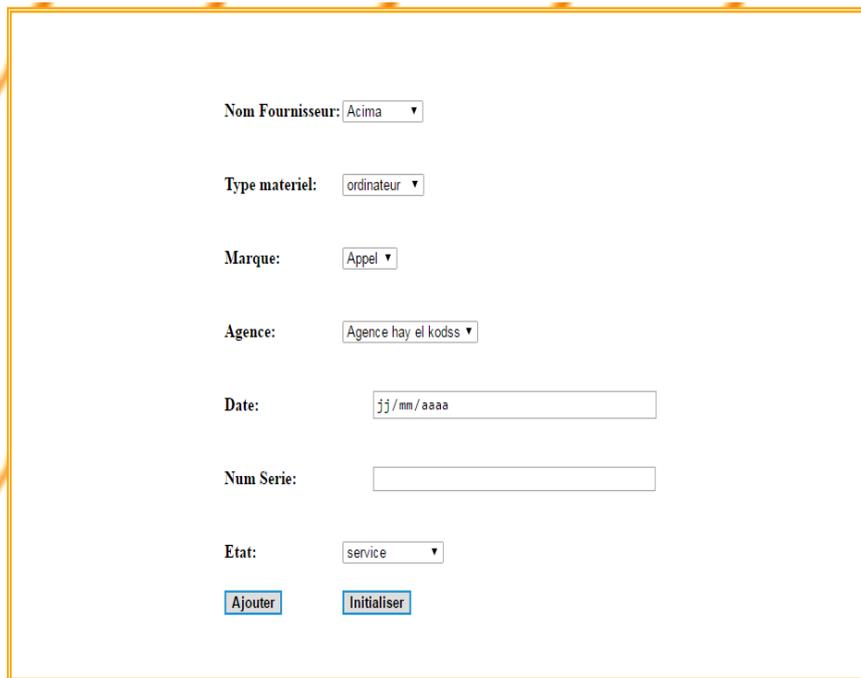
Ci-dessous, une capture d'écran de la gestion d'équipement :



Figure 17 gestion équipement

Lors d'ajout d'un équipement l'utilisateur doit remplir un formulaire d'ajout « Ajout Direct » ou d'importer un fichier Excel à la base de données. Voici des captures d'écran de l'ajout :

Ajout direct :



Nom Fournisseur: Acima ▼

Type materiel: ordinateur ▼

Marque: Appel ▼

Agence: Agence hay el kodss ▼

Date: jj/mm/aaaa

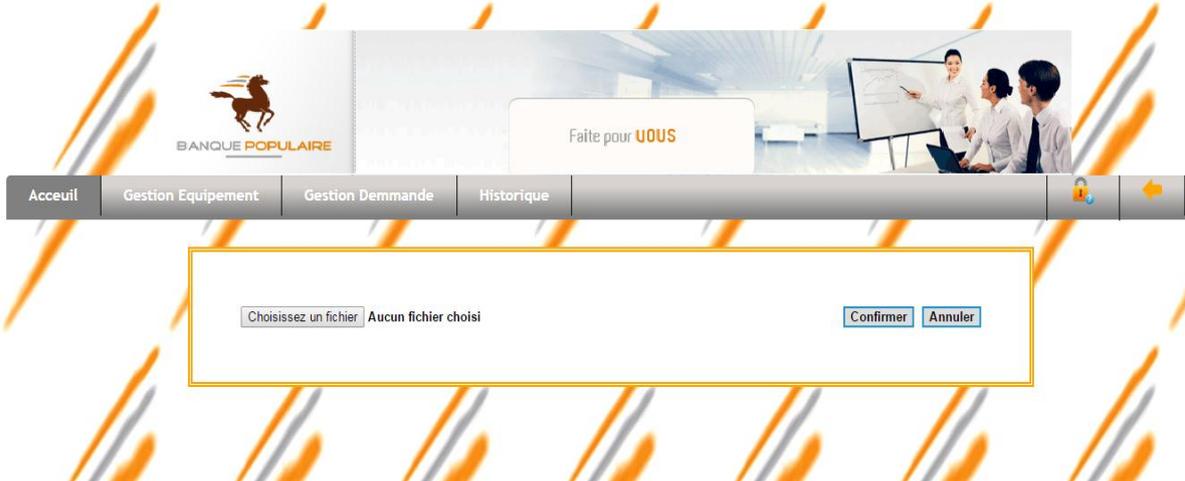
Num Serie:

Etat: service ▼

Ajouter Initialiser

Figure 18 Ajout direct

Importer fichier Excel:



BANQUE POPULAIRE

Faites pour UOUS

Acceuil Gestion Equipement Gestion Demande Historique

Choisissez un fichier | Aucun fichier choisi

Confirmer Annuler

Figure 19 Importer fichier Excel

Pour le listage des équipements, nous avons fait trois listage (listage global, listage particulier, listage dans une agence). Voici des captures d'écrans du Listage :

Listage d'équipement:

Projet de fin d'études : Application web de Portail de gestion du parc



Figure 20 Listage Equipement

Listage global:

The screenshot shows a table titled 'Listage globale du parc'. It has a search bar and a 'Show 10 entries' dropdown. The table contains 10 rows of equipment data.

Numero de serie	Prestataire	Type	Marque	Etat	Agence d'origine	Date d'ajout	Action
1	Acima	Pc	Appel	service	Agence hay el kodss	2017-05-26	
2	Microchoix	ordinateur	Appel	service	Agence hay el kodss	2017-05-26	
3	Marjane	PC	HP	service	AgenceNarjiss	2017-05-26	
4	Acima	ordinateur	Dell	service	Agence hay el kodss	2017-05-26	
5	Acima	ordinateur	Appel	service	Agence hay el kodss	30-05-2010	
6	Acima	ordinateur	sonny	reforme	Agencehay	2017-05-25	
8	Acima	ordinateur	Dell	service	Agence hay el kodss	2017-05-25	
17	Acima	routeur	samsung	cede	Agence hay el kodss	2017-05-18	
18	Acima	ordinateur	sonny	reforme	Agence narjiss	2017-05-18	

Figure 21 Listage global

Puisque notre application offre la possibilité de consulter les données, il permet aussi de les modifier. Si on clique sur cette icône  la ligne sélectionnée se transforme à un formulaire rempli modifiable :

Voici un capture d'ecran d'une formulaire à modifier :

Projet de fin d'études : Application web de Portail de gestion du parc

Num Serie:

Nom Fournisseur:

Type materiel:

Marque:

Agence:

Date:

Etat:

Figure 22 *Modifier un équipement.*

Listage particulier:

Le listage particulier permet de chercher un équipement avec les informations demander. Voici un capture d'écran d'un Listage particulier du parc :

Listage Particulier du parc

Show: entries Search:

Nom Fournisseur:	Type materiel:	Marque:	Etat:				<input type="button" value="Rechercher"/>
Numero de serie	Nom fournisseur	Type	Marque	Etat	Agence d'origine	Date d'ajout	
1	Marjane	Pc	HP	service	Agence hay el kodss	2017-05-26	
2	Marjane	PC	HP	service	siege atlas	2017-05-26	
3	Marjane	PC	HP	service	siege atlas	2017-05-26	
4	Marjane	PC	HP	service	sige atlas	2017-05-26	
6	Acima	ordinateur	Appell	reforme	Agence hay el kodss	2017-05-25	
8	Acima	ordinateur	Appell	service	Agence hay el kodss	2017-05-25	

localhost:8080/PFE/corps.jsp

Figure 23 *Listage particulier.*

Listage dans une Agence:

Projet de fin d'études : Application web de Portail de gestion du parc

Le listage dans une agence permet de consulter tout les équipement dans une agence demander. Voici un capture d'écran du listage dans une agence:

Numero de serie	Prestataire	Type	Marque	Etat	Agence d'origine	Date d'ajout
1	Marjane	Pc	HP	service	Agence hay el kodss	2017-05-26
2	Marjane	PC	HP	service	siege atlas	2017-05-26
3	Marjane	PC	HP	service	siege atlas	2017-05-26
4	Marjane	PC	HP	service	sige atlas	2017-05-26
6	Acima	ordinateur	Appell	reforme	Agence hay el kodss	2017-05-25
8	Acima	ordinateur	Appell	service	Agence hay el kodss	2017-05-25
17	Acima	routeur	samsung	cede	Agence narjiss	2017-05-18
18	Acima	ordinateur	sonny	reforme	Agence narjiss	2017-05-18
19	Acima	ordinateur	Appell	service	Agence hay el kodss	2017-05-25
44	Microchoix	ordinateur	sonny	service	Siege Atlase	2017-05-25

Figure 24 Listage dans une Agence.

Transférer un équipement :

Pour transférer un équipement, on choisit son numéro de série existant dans la base de données, ensuite on choisit le nouveau emplacement. Voici un capture d'écran du transfère :

Accueil | Gestion Equipement | Gestion Demande | Historique

Ajouter
Listage
Transférer

Num Serie: 1

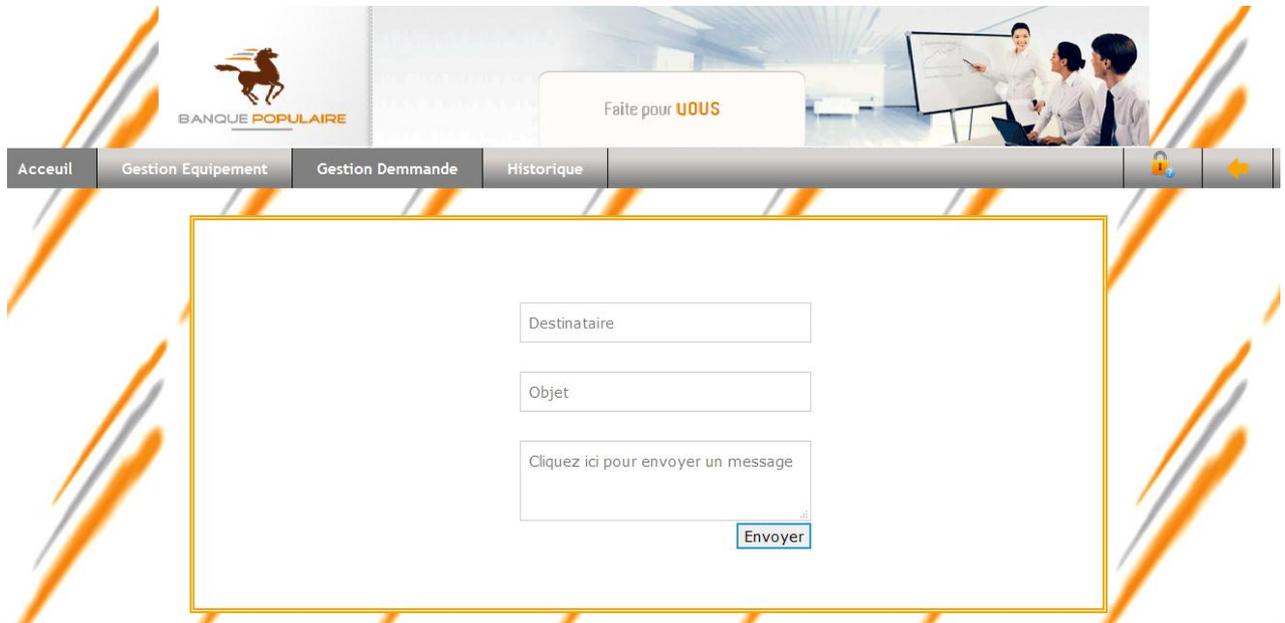
Agence: Agence hay el kodss

Transférer Initialiser

Figure 25 Transférer équipement

Projet de fin d'études : Application web de Portail de gestion du parc

Remarque : s'il s'agit d'un nouveau fournisseur, nouvelle marque, nouvelle Agence ou nouveaux type d'équipement, le système permet à l'administrateur seulement d'ajouter ses informations à la base de données. L'employé envoie un email à l'administrateur en lui demandant l'ajout de ces informations. Voici un capture d'écran pour la gestion de demande :



The screenshot shows a web application interface for 'BANQUE POPULAIRE'. At the top, there is a navigation menu with the following items: 'Accueil', 'Gestion Equipement', 'Gestion Demande', and 'Historique'. The 'Gestion Demande' item is currently selected. Below the navigation menu, there is a main content area with a yellow border. Inside this area, there are three input fields: 'Destinataire', 'Objet', and a larger text area with the placeholder text 'Cliquez ici pour envoyer un message'. To the right of the text area is a blue button labeled 'Envoyer'. The background of the application features a banner with the text 'Faites pour VOUS' and an image of three business professionals in a meeting.

Figure 26 gestion de demande.

L'employé peut consulter toutes les tâches faites dans l'application « Historique » en précisant la date de la tâche et son acteur. Voici un capture d'écran pour l'historique :

Projet de fin d'études : Application web de Portail de gestion du parc

Historique

Show entries Search:

Num_serie	Marque_equipement	Type_equipement	Nom_user	Prenom_user	Date	type_operation
1	Appel	Pc	benslimane	ghita	2017-05-26	Ajouter(admin)
1	Appel	Pc	benslimane	ghita	26-05-17	Transferer(admin)
1	Appel	Pc	benslimane	ghita	30-05-17	Modifier(admin)
2	Appel	ordinateur	benslimane	ghita	30-05-17	Ajouter(admin)
2	Appel	ordinateur	benslimane	ghita	30-05-17	Transferer(admin)
2	Appel	ordinateur	benslimane	ghita	30-05-17	Modifier(admin)
3	HP	PC	jrondi	zineb	30-05-17	Transferer(user)
4	Dell	ordinateur	jrondi	zineb	30-05-17	Modifier(user)
5	Appel	ordinateur	benslimane	ghita	30-05-17	Ajouter(admin)
5	Appel	ordinateur	benslimane	ghita	30-05-17	Transferer(admin)

Showing 1 to 10 of 27 entries

Previous **1** 2 3 Next

Figure 27 historique

L'utilisateur a la possibilité de modifier son mot de passe en cliquant sur cette icône



.Voici un capture d'écran pour la modification du mot de passe:

Faites pour UOUS

Acceuil | Gestion Equipement | Gestion Demande | Historique |

zineb+

Ancien mot de passe

Nouveau mot de passe

Confirmer le mot de passe

Envoyer Initialiser

Figure 28 modifier mot de passe

Projet de fin d'études : Application web de Portail de gestion du parc

Pour se deconnecter, on clique sur cette iconne  le système affiche la page d'authentification.

➤ Volet 2 : Administrateur:

Ce volet permet à l'administrateur après l'authentification d'effectuer les mêmes opérations de l'employé simple, et en plus il peut réaliser les opérations suivantes :

- Type équipement (ajouter, lister).
- Marque Equipement (ajouter, lister).
- Gestion Agence (ajouter, lister).
- Gestion Compte (ajouter, lister).
- Gestion fournisseur (ajouter, lister).

Voici une capture d'écran du Type équipement :



Figure 29 Type équipement.

Ajouter Marque Equipement :



Figure 30 Ajouter Marque équipement.

Projet de fin d'études : Application web de Portail de gestion du parc

Listage des Agences:



Figure 31 Listage des agences

Si on clique sur cette icône l'agence sera supprimé.

Puisque notre application offre la possibilité de consulter les données, elle permet aussi de les modifier ou les supprimer. Si on clique sur cette icône la ligne sélectionnée se transforme à un formulaire rempli modifiable :

Voici un capture d'écran de la modification d'un fournisseur:

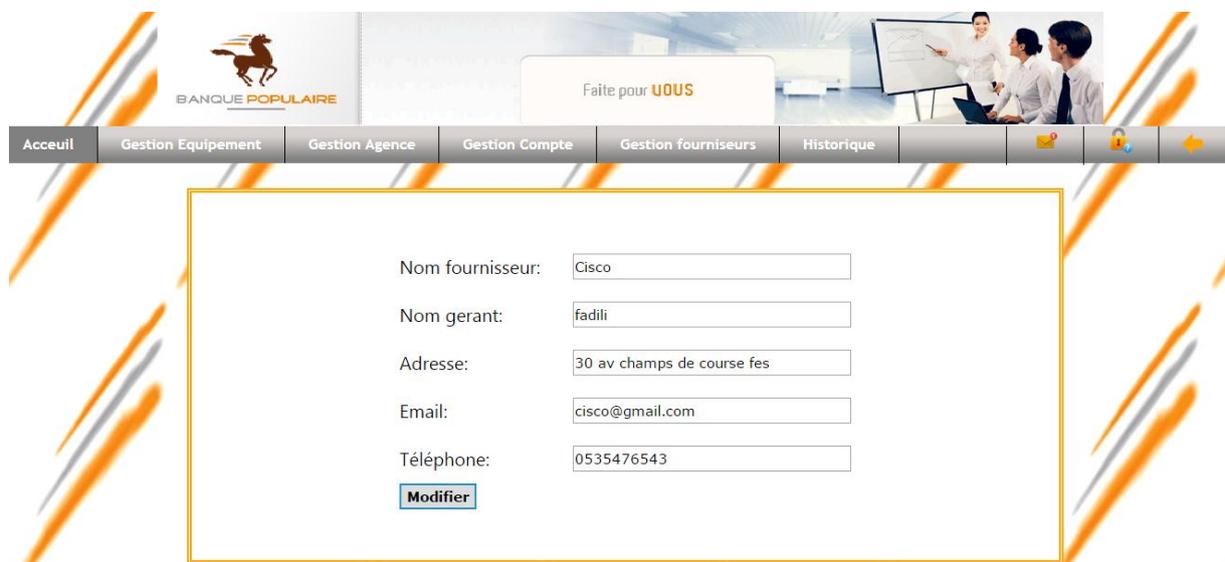
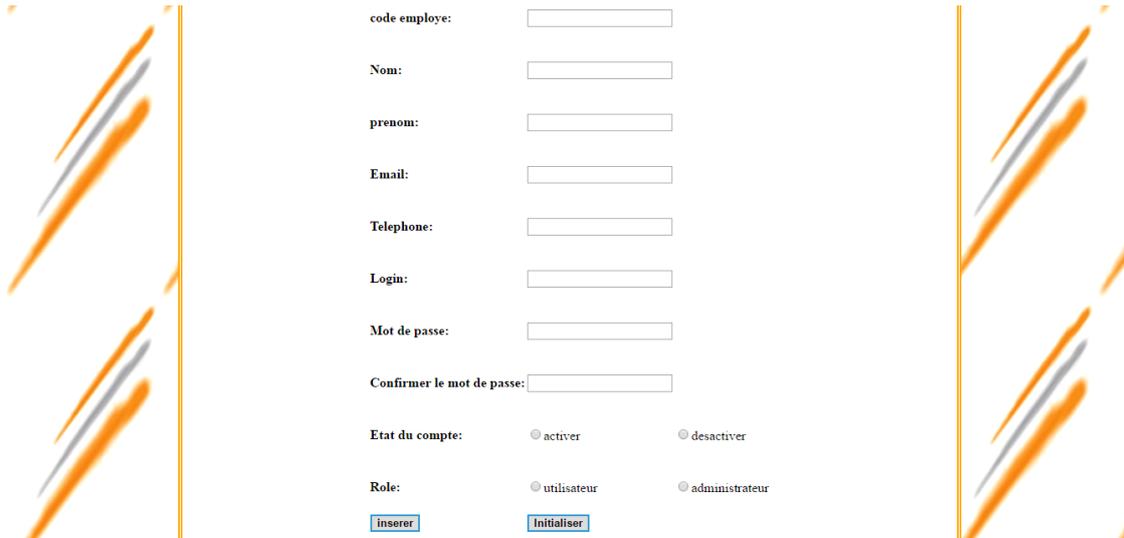


Figure 32 modifier fournisseur

L'ors d'ajout d'un compte, le système affiche un formulaire à remplir. Voici un capture d'écran de l'ajout d'un compte :

Projet de fin d'études : Application web de Portail de gestion du parc



code employe:

Nom:

prenom:

Email:

Telephone:

Login:

Mot de passe:

Confirmer le mot de passe:

Etat du compte: activer desactiver

Role: utilisateur administrateur

Figure 33 Ajouter compte

L'administrateur clique sur cette icône  Pour consulter sa boîte mail et voir les demandes.

Pour se deconnecter, on clique sur cette icône  le système affiche la page d'authentification.

Conclusion

Notre application, réalisé au profit de la BP pour la gestion de son parc informatique, qui est un ensemble de matériels informatiques. Il est répartie sur différents emplacements : siege, agence et cicursaille dans les régions Fes-Taza

Au cours de ce projet, nous avons créer une application web au sein du siège, pour pouvoir faciliter la tâche aux personnels du Département gestion infrastructure et support informatique en ce qui concerne la gestion de parc.

Au cours de ce projet ,on a pu apprendre mieux les bases de l'architecture JEE ,et de decouvrir les APIs JDBC ,servlet JSP. En outre ,on a eu l'occasion de decouvrir le langage de modelisation UML et le design pattern MVC2 .

Webographie

- <http://www.siteduzero.com/tutoriel-3-112219-apprenez-a-creer-des-applications-web-dynamiques-avec-jee.html/>.
- [http:// www. java.developpez.com/faq/jdbc/](http://www.java.developpez.com/faq/jdbc/).
- [http:// www.jquery.com/](http://www.jquery.com/).
- <http://www.xul.fr/xml-ajax.html>.
- [http://www.wikipedia.org /](http://www.wikipedia.org/).
- <http://www.gbp.ma/>.
- <http://www.w3school.com/>.
- [http://www. open classroom.com/](http://www.openclassroom.com/).

Bibliographie

- Les cahiers du programmeur J2EE (Jérôme Mplière)
- Les cahiers du programmeur Java EE5 (Antonio Goncalves)
- Beginning Java EE 6 Platform with GlassFish 3(Novice to Professional)