

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

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER PROFESSIONNEL

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Ingénierie des systèmes d'information**

Présenté par

**SARRA RAHMANI
TASSADIT AIT ISSAD**

Thème

Conception et réalisation d'une application web JEE pour la gestion des investissements Cas : ENIEM

Mémoire soutenu publiquement le 22/06/ 2016. devant le jury composé de :

Président : M MOHAND KARBICHE

Encadreur : M LOKMANE CHEBOUBA

Examineur : M TASSADIT BERKANE

Examineur : M DIB

Remerciements

Au terme de ce travail, on tient à exprimer toute notre gratitude, reconnaissance et nos remerciements à notre promoteur MrCheboubaLokmane pour tout le temps qu'il nous a consacré avec abnégation et sa directive précieuse, pour ses orientations et son suivi tout au long de notre travail.

Nous adressons nos remerciements les plus sincères aux membres du jury qui nous font l'honneur d'évaluer notre travail.

Nos remerciements vont aussi à notre encadreur au niveau de ENIEM production, Mr KABI, et tout le personnel qui nous ont réservé un accueil chaleureux au sein de l'entreprise.

Nous sommes également reconnaissantes à toutes personnes qui nous ont aidé de près ou de loin à la réalisation de ce travail.

Dedicaces

Je dédie ce travail à :

Mes très chers parents, pour leur amour, soutien et sacrifices, qui m'ont épaulé tout au long de ma vie, qu'ils trouvent ici ma profonde reconnaissance.

Ma très chère sœur Kahina

Mon très cher frère Karim

Mes Amis, et tous ceux qui ont contribués de près ou de loin à l'accomplissement de mon travail.

A celle avec qui j'ai partagé des moments inoubliables : TASSADIT à qui je souhaite le succès et la réussite.

R.Sarra

Dedicaces

Je dédie ce travail à :

Mes très chers parents, pour leur amour, soutien et sacrifices, qui m'ont épaulé tout au long de ma vie, qu'ils trouvent ici ma profonde reconnaissance.

Mes très cher frères ainsi, qu'à mes Amis, et tous ceux qui ont contribué de près ou de loin à l'accomplissement de mon travail.

A celle avec qui j'ai partagé des moments inoubliables : SARRA à qui je souhaite le succès et la réussite.

A.Tassadit

SOMMAIRE

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

CHAPITRE I : Présentation de l'architecture JEE

Introduction.....	2
I. Architecture client /serveur.....	2
I.1. Les notions de bases.....	2
I.2. Les types d'architecture client/serveur.....	4
I.3. Principe.....	5
I.4. Avantage	5
I.5. Inconvénients.....	5
II. Profil d'une architecture web idéale.....	6
III. Présentation de l'architecture en couches.....	7
IV. Architecture JEE et ses composants.....	8
IV.1. Présentation de JEE.....	8
IV.1.1. Qu'est-ce que java EE.....	8
IV.1.2. La différence entre java EE et java SE.....	9
IV.1.3. Historique de java EE.....	9
IV.1.4. Architecture java EE.....	11
IV.1.5. La notion de POJO.....	15
IV.1.6. Présentation du modèle de conception MVC.....	15
IV.1.6.1. Définition.....	15
IV.1.6.2. Principe.....	16
IV.1.6.3. Cinématique	17
V. Introduction au frameworkspring.....	18
V.1. Architecture du frameworkspring.....	18
Conclusion	22

CHAPITRE I : Présentation de l'organisme d'accueil

Introduction.....	23
I. Situation géographique	23
II. Historique	23
III. Organisation générale	25
IV. Organigramme général de l'ENIEM.....	25
V. Présentation du champ d'étude	29
V.1. Définition des mots techniques.....	29
V.1.1. Définition de l'immobilisation.....	29
V.1.2. Comptabilisation	30

V.1.3.L'amortissement.....	30
V.1.4.Formules de calculs	30
V.1.5. Cycle de vie d'une immobilisation	31
V.1.5.1. acquisition d'une immobilisation	31
V.1.5.2. achat externe.....	31
V.1.5.3. Les cessions inter unités.....	32
V.1.5.4. Transfert de l'immobilisation	32
V.2. Mission du champ d'étude.....	32
V.3. Situation informatique du champ d'étude.....	32
V.4. Etudes des postes de travail et des documents.....	33
V.4.1. Etudes des postes de travail.....	33
V.4.2. Etudes des documents.....	34
V.4.3. Liste des documents.....	34
V.5. Critiques et suggestions.....	35
V.5.1. Les anomalies	35
V.5.2. Les solutions.....	35
Conclusion	35

CHAPITRE III : Analyse et conception

Introduction	36
I.1. Analyse	36
I.1.1.L'étape d'analyse	36
I.1.2.Objectif du travail.....	36
I.1.3.Démarche de modélisation	37
I.1.3.1.Les acteurs	37
I.1.3.2.Spécification des tâches et des acteurs	37
I.1.3.3.Spécification des scénarios pour chaque tâche	40
I.1.4.Spécification des cas d'utilisation	57
I.1.4.1.Description des cas d'utilisation	57
I.1.4.2.Diagramme de cas d'utilisation.....	58
II.Conception	61
II.1.Les diagrammes de séquences	61
II.2.Diagramme de classes global.....	63
III.Etude conceptuelle statique	65
III.1.La base de données	65
III.1.1.Les concepts de base	65
III.1.2.Les règles de gestion du MCD.....	65
III.1.3.Représentation du modèle conceptuel de données	66
III.1.4.La liste des associations.....	67

IV.	Conception de la base de données	68
IV.1.	Représentation du modèle logique de données	68
IV.2.	Les tables.....	69
V.	Algorithme de calcul d’amortissement.....	75
	Conclusion.....	76
CHAPITRE IV : Réalisation		
	Introduction	77
I.	Environnement et outils de développement et d’implémentation.....	77
I.1.	Google Drive.....	77
I.2.	Wampserver	78
I.3.	Eclipse	79
I.4.	Apache Tomcat.....	79
I.5.	ApacheMaven.....	80
I.6.	Bootstrap.....	80
I.7.	Mysql.....	81
I.8.	L’interface graphique PhpMyAdmin.....	82
I.9.	Git.....	83
I.10.	Les langages de programmation	84
II.	Présentation de quelques interfaces de l’application.....	84
II.1.	Interface de « Connexion ».....	84
II.2.	Interface « Accueil ».....	85
II.3.	Interface « Ajouter un bien ».....	86
II.4.	Interface « Rechercher un bien ».....	86
II.5.	interface « Affichage du bien ».....	87
II.6.	interface « visualisation des biens d’un compte ».....	88
	Conclusion	89
	Conclusion générale.....	90

Les figures :

Chapitre I : Présentation de l'organisme d'accueil

Figure I.1 : tableau de l'historique de l'ENIEM

Figure I.2 : Organigramme générale de l'ENIEM.

Chapitre II : introduction à l'architecture JEE

Figure I.3 : notion de base Client//serveur

Figure I.4 : Architecture Client/Serveur à deux niveaux

Figure I.5 : Architecture Client/serveur à trois niveaux

Figure I.6 : les couches de base d'une applicatiob web

Figure I.7:Architecture JEE et ses composants

Figure I.8 : Architecture d'une application web avec le modèle MVC1

Figure I.9 : Architecture d'une application web avec MVC2

Figure I.10 : Architecture du frameworkSpring

Figure I.11 : Fonctionnement de spring MVC

Chapitre III : Analyse et conception

Figure III.12 : la démarche de modélisation de l'application

Figure III.13 : tableau des spécifications des tâches et identification des acteurs

Figure III.14 : tableau des scénarios et des tâches de l'utilisateur

Figure III.15 : tableau des scénarios des tâches de l'administrateur

Figure III.16 : Diagramme de cas d'utilisation « acteur,Utilisateur »

FigureIII.17 : Diagramme de cas d'utilisation « acteur,Administrateur »

FigureIII.18 : Diagramme de séquence du cas d'utilisation « s'authentifier »

Figure III.19 : Diagramme de séquence du cas d'utilisation « ajouter un bien »

Figure III.20 : Diagramme de séquence du cas d'utilisation « rechercher un bien »

Figure III.21 : Diagramme de classe générale

Figure III.22 : le modèle conceptuel de données de notre application

Chapitre IV : Réalisation

Figure IV.23 : Interface google Drive

Figure IV.24 : panneau de gestion du serveur

Figure IV.25 : Interface d'eclipse

Figure IV.26 : Interface d'accueil PhpMyAdmin

Figure IV.27 : Interface de connexion

Figure IV.28 : interface d'accueil

Figure IV.29 : interface ajouter un bien

Figure IV.30 : Interface rechercher un bien

Figure IV.31 : Interface de l'affichage du bien (partie1)

Figure IV.32 : Interface de l'affichage du bien (partie2)

Figure IV.33 : Interface de visualisation des biens d'un compte.

Les tableaux :

Tableau II.1 : Historique de l'ENIEM.

Tableau III.2 : Spécification des tâches et identification des acteurs.

Tableau III.3 : Scénario des tâches de l'administrateur.

Tableau III.4 : Liste des associations.

Tableau III.5 : Table bien.

Tableau III.6 : Table centre de frais.

Tableau III.7 : Table unité.

Tableau III.8 : Table utilisateur.

Tableau III.9 : Table historique cession.

Tableau III.10 : Table historique transfert.

Tableau III.11 : Table message.

Tableau III.12 : Table fournisseur.

Tableau III.13 : Table réévaluation.

Tableau III.14 : Table codification.

Introduction générale

Les systèmes d'information sont devenus indispensables au fonctionnement des entreprises car ils ont été utilisés comme un élément permettant d'améliorer leur productivité, ce qui donne à l'informatique une place importante au sein des entreprises.

La technologie de traitement de l'information offre une grande variété d'outils permettant d'améliorer les services rendus par les entreprises

ENIEM est une industrie algérienne qui se développe de jour en jour dédiée à la fabrication et montage des appareils électroménagers.

Après l'analyse de la méthode de travail de cet organisme et en particulier le service comptabilité au quel nous avons été orientés par les responsables de l'unité UPT (Unité Prestation Technique), nous avons relevé un ensemble de points que l'on pourrait améliorer. Ces derniers font l'objet de notre problématique qui peut être formulée comme suit : quelles sont les propositions envisageables pour faciliter la maintenance de l'application de gestion des immobilisations existante? Ajoutons à cela l'implémentation de la fonctionnalité réévaluation des biens, ainsi que la gestion des utilisateurs de cette application par un administrateur.

Notre travail consiste à répondre à cette problématique à travers les quatre chapitres qui le constituent.

Organisation du mémoire :

Dans le premier chapitre de ce mémoire, nous aborderons les points importants de la technologie J2EE qui repose sur le principe de l'orienté objet qui facilite la maintenance et l'extension de l'application.

Dans le deuxième chapitre, nous présenterons l'organisme d'accueil, quelques définitions techniques nécessaires pour la compréhension du besoin de l'entreprise.

Le troisième chapitre nous le consacrerons à l'analyse et la conception de notre application.

Quant au quatrième chapitre donnera une vue globale des solutions techniques qui vont nous permettre de réaliser notre application.

Et pour terminer, une conclusion synthétisera notre travail.

Chapitre I

Présentation de l'architecture JEE



Introduction :

Dans ce chapitre nous allons aborder les bases essentielles à la compréhension de notre application. Le standard JEE fait partie des technologies dites révolutionnaires de notre époque précisément en développement d'applications d'entreprises de grande envergure. Elle présente plusieurs solutions et elle est en perpétuelle évolution.

Nous n'allons pas aborder le standard J2EE avec l'ensemble de ses composants, car il est parfois lourd d'utiliser un composant conçu pour de grandes architectures alors que notre application est restreinte. C'est pour cela que nous avons choisi d'adopter une architecture JEE avec l'intégration des Framework.

L'ensemble de la communauté Open Source (principalement) s'est occupée (et s'occupe) de lancer sur le marché des frameworks servant à simplifier l'utilisation de telle ou telle technologie. Les Framework sont plus simples d'utilisation et plus performants dans certains cas.

Dans ce chapitre nous allons aborder quelques principes de Spring.

I. L'architecture Client/Serveur :

L'architecture client/serveur désigne un mode de communication entre plusieurs ordinateurs d'un réseau. Des machines clientes (faisant partie du réseau) contactent un serveur qui leur fournit des services qui sont des programmes fournissant des données tels que les fichiers, une connexion,...

I.1. Les notions de bases

- **Client** : processus demandant l'exécution d'une opération à un autre processus serveur par l'envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
- **Serveur** : processus accomplissant une opération sur demande d'un client et transmettant la réponse à ce dernier.
- **Réponse** : message transmis d'un serveur à un client suite à l'exécution d'une opération contenant les paramètres de l'opération.

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

- **Requête:** message transmis d'un client à un serveur décrivant l'opération à exécuter.

Les appels aux services de transport mis en jeu sont au nombre de quatre:

1. **SendRequest ()** : permet au client d'émettre le message décrivant la requête à une adresse correspondante à la porte d'écoute du serveur.
2. **ReceiveRequest()** : permet au serveur de recevoir la requête sur sa porte d'écoute.
3. **SendReply()** : permet au serveur d'envoyer la réponse sur la porte d'écoute du client.
4. **ReceiveReply ()** : permet au client de recevoir la réponse en prévenance du serveur.

En résumé, la figure suivante illustre ces notions:

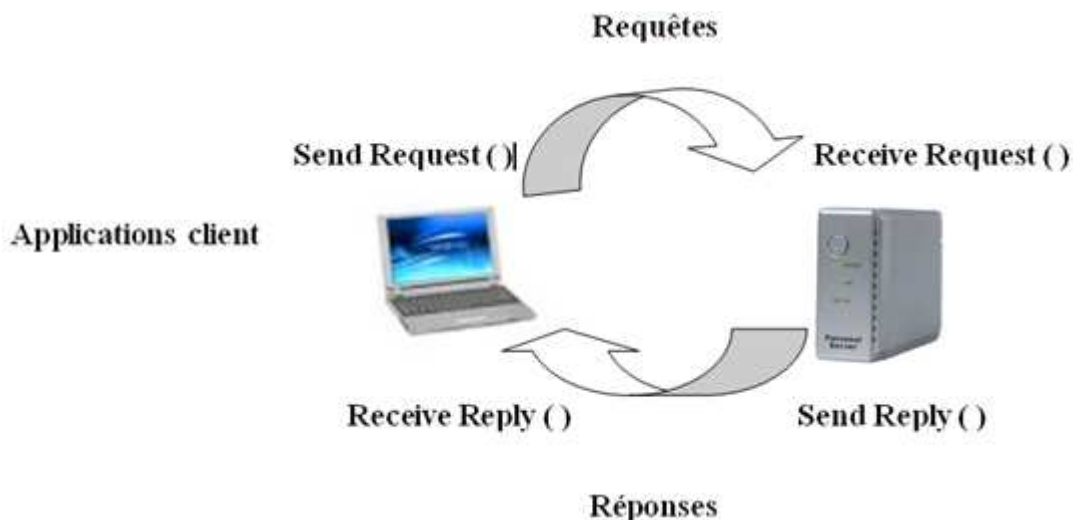


Figure I.1 : Notions de base client/serveur

- **Middleware** : ensemble des services logiciels jouant le rôle d'intermédiaires entre les clients et les serveurs en assurant le dialogue entre ces derniers.

I.2 Les types d'architecture Client/serveur :

On peut distinguer deux types d'architectures :

- **Architecture à deux niveaux (2-tiers):** Le client émet la requête et le serveur exécute directement sans passer par un service intermédiaire (serveur de bases de données).

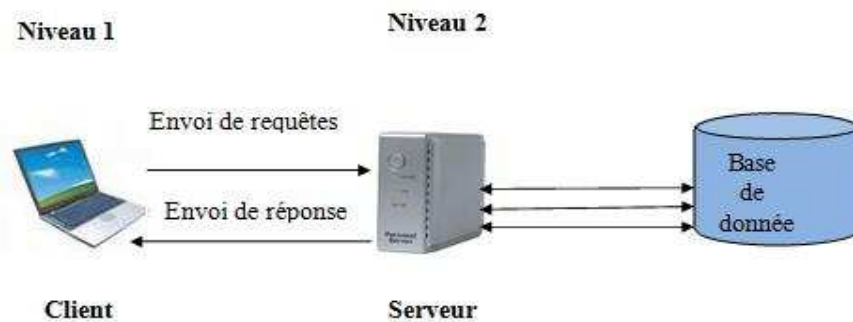


Figure I.2 : Architecture client-serveur à deux niveaux.

- **Architecture à trois niveaux (3-tiers) :** Le client lance une requête, le serveur d'application exécute en faisant appel à un autre serveur.

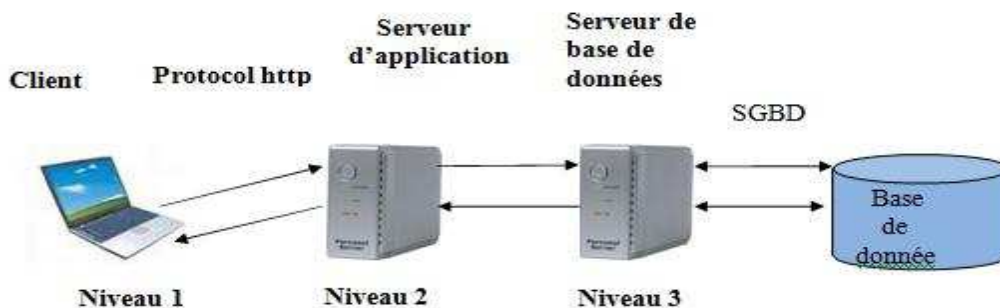


Figure I.3 : Architecture client-serveur à trois niveaux.

I.3 Principe

- Les ressources réseau sont centralisées.
- Un ou plusieurs serveurs sont dédiés au partage de ces ressources et en assurent la sécurité.
- Les postes clients ne sont en principe que des clients, ils ne partagent pas de ressources, ils utilisent celles qui sont offertes par les serveurs.

I.4 Avantages

- Les serveurs sont conçus pour le partage de ressources et ne servent pas de station de travail. Il suffit de les dimensionner en fonction de la taille du réseau et du nombre de clients susceptibles de s'y connecter.
- Ils proposent également des fonctions avancées à l'usage des utilisateurs comme par exemple les profils itinérants qui permettent à un utilisateur (sous certaines conditions) de retrouver son environnement de travail habituel, même s'il change de poste de travail.
- Les serveurs étant toujours en service (sauf en cas de panne...), les ressources sont toujours disponibles pour les utilisateurs

I.5 Inconvénients :

- La mise en place d'un tel réseau est beaucoup plus lourde qu'un cas simple de "poste à poste".
- Le coût est évidemment plus élevé puisqu'il faut la présence d'un ou de plusieurs serveurs.
- Si un serveur tombe en panne, ses ressources ne sont plus disponibles.

Ce type de réseau est évidemment le plus performant et le plus fiable. Ce n'est pas la solution la plus simple pour un réseau domestique, c'est cependant ce type d'architecture que l'on retrouve sur les réseaux d'entreprise, qui peut parfaitement supporter plusieurs centaines de clients, voire plusieurs milliers.

II. Profil d'une architecture web idéale :

En informatique, une application web (aussi appelée web app, de l'anglais) est une application manipulable grâce à un navigateur web. De la même manière que les sites web, une application web est généralement placée sur un serveur et se manipule en actionnant des requêtes à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.)

Une architecture applicative web moderne ne peut se contenter d'exister côté serveur. Il y a donc en réalité deux architectures symbiotiques à définir :

❖ **Côté serveur** : une application web doit réunir les qualités suivantes :

- une structure en couches minimalistes : le strict nécessaire, ni plus ni moins. Il est nécessaire d'organiser le code de son application, mais il est contre-productif de s'embarrasser d'une architecture plus complexe que le travail à effectuer ne l'exige.
- une approche orientée services : plutôt que systématiquement retourner l'intégralité de la page HTML à chaque réponse HTTP, une application web doit permettre de limiter les transferts de données au strict minimum : soumission partielle de formulaires, mise à jour de la seule partie de HTML concernée, voire seulement des données. Il ne s'agit donc plus d'appeler telle ou telle page mais d'invoquer tel ou tel service ultra spécialisé.
- des performances irréprochables : il n'est pas déraisonnable de demander à une application web de toujours répondre en quelques dizaines de millisecondes, et dans tous les cas de ne jamais dépasser une seconde de temps de réponse (à supposer bien sûr que l'utilisateur dispose d'une connexion décente). Les technologies actuelles permettent des traitements ultra rapides côté serveur et les débits de connexion sont énormes, il n'y a donc pas d'excuse pour des performances qui laissent à désirer. [1]

❖ Côté client :

- La fluidité de navigation : pas d'excès d'effets graphiques, pas de surcharges des fonctionnalités de base du navigateur
- la légèreté de l'architecture : N'oublions pas que par définition nous ne connaissons pas la puissance de la machine qui exécutera notre code.
- une empreinte minimaliste : On préférera ne pas polluer la machine de l'utilisateur et garder sagement nos données côté serveur. [2]

III. Présentation de l'architecture en couches :

L'architecture en couches consiste à diviser une application en différents modules, qui constituent autant de couches. L'objectif est de proposer une meilleure répartition des rôles (chaque module a un rôle clairement défini), la séparation des traitements, ainsi qu'une réduction des dépendances entre les services. Chaque module se doit d'être indépendant des autres pour permettre une meilleure maintenabilité.

Tier signifie étage ou niveau, son nom provient de l'anglais. Il s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles (étages, niveaux, tiers ou strates). L'architecture d'une application se découpe idéalement en au moins trois tiers : [3]

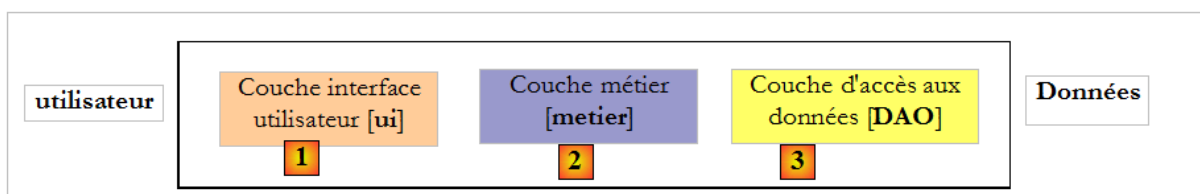


Figure I.4 : les couches de base d'une application web

1. La couche présentation :

La couche présentation est la couche qui dialogue avec l'utilisateur, via une interface. Elle a pour rôle de fournir des données provenant de l'utilisateur à la couche métier ou bien de présenter à l'utilisateur des données fournies par la couche métier.

2. La couche métier :

La couche métier est la couche qui applique les règles dites métier, c'est à dire la logique spécifique de l'application, sans se préoccuper de savoir d'où viennent les données qu'on lui donne, ni où vont les résultats qu'elle produit

3. La couche Accès aux données :

La couche Accès aux données, appelée ici DAO (Data Access Object) est la couche qui fournit à la couche métier des données et qui enregistre certains des résultats fournis par la couche métier.

Dans cette approche, les couches communiquent entre elles au travers d'un « modèle d'échange », et chacune d'entre elles propose un ensemble de services rendus. Les services d'une couche sont mis à disposition de la couche supérieure. On s'interdit par conséquent qu'une couche invoque les services d'une couche plus basse que la couche immédiatement inférieure ou plus haute que la couche immédiatement supérieure (chaque couche ne communique qu'avec ses voisins immédiats).

IV. Architecture JEE et ses composants :

IV.1 Présentation du Java EE

IV.1.1 Qu'est-ce que le Java EE ?

Le Java Entreprise Edition, comme son nom l'indique, a été créé par la société Sun Microsystems, désormais Oracle Corporation, porté par un consortium de sociétés internationales pour le développement d'applications d'envergure et supportant en standard la programmation, la persistance des données, les transactions, la sécurité ou encore le déploiement de projets sur serveur applicatif.

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

Oracle Corporation est une entreprise américaine créée à la fin des années 70. Sun Microsystems, éditeur de logiciels, était à l'origine du langage Java et des technologies Java EE. Le 20 avril 2009, Oracle rachète Sun Microsystems. Ainsi tout au long de l'ouvrage ces noms d'entreprises seront utilisés.

L'utilisation de J2EE pour développer et exécuter une application offre plusieurs avantages :

- une architecture d'applications basée sur les composants qui permettent un découpage de l'application et donc une séparation des rôles lors du développement
- la possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA ...
- la possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres [4]

IV.1.2 La différence entre Java EE et JavaSE

- Le terme « Java » fait bien référence à un langage, mais aussi à une plateforme connue sous le nom de « Java Standard Edition » qui est constituée de nombreuses bibliothèques ou API, mettant à la disposition des programmeurs un nombre conséquent de classes et de méthodes prêtes à l'emploi pour effectuer toutes sortes de tâches.
- La plate-forme Java EE est construite sur le langage Java puisqu'elle est simplement une extension de la plate-forme Java Standard. Elle y ajoute un grand nombre de bibliothèques qui lui permettent de remplir des fonctionnalités que la plate-forme standard ne remplit pas d'origine. [5]

IV.1.3 L'historique de Java EE :

La plate-forme JavaEE est passée par plusieurs étapes et plusieurs solutions sont possibles :

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

- La version J2EE 1.3 (2001) n'apportait pas de nouveautés essentielles mais des améliorations de la plate-forme à la suite des demandes utilisateur, comme la configuration par fichier XML.
- la version J2EE 1.4 (2003) cette version apportait le support des services Web et des EJB version 2.1 mais cette version conséquente était difficile à utiliser, tester et déployer. Les développements par couches, de type Modèles Vue Contrôleur(MVC), ne facilitent pas la compréhension de l'ensemble, et les technologies Servlet et JSP ne sont pas adaptées pour un développement rapide. En réponse à cela, quelques Framework ou outils spécialisés ont vu le jour, comme Struts et Spring pour la partie développement serveur, et Hibernate ou TopLink pour la persistance.
- La version Java EE 5 (2006) apporte une réponse à cela en proposant un outil de développement côté serveur adapté au design pattern MVC, nommé Java Server Faces. Cette nouvelle version est une étape majeure pour la plate-forme Java EE qui s'oriente vers la simplicité, en proposant les EJB3.
- la version JavaEE6(2009) se concentre sur la facilité d'utilisation de la plate-forme pour le développement, les tests et le déploiement. Cette version propose une simplification des EJB avec la version3.1.

Java EE est devenu un standard utilisé par de nombreuses sociétés à travers le monde professionnel et celui de l'Open Source. Plusieurs outils peuvent être associés à javaEE comme les serveurs d'applications (GlassFish, Websphere, Weblogic, JBoss...) des Framework de développement (Struts, Spring..), des outils de persistance. Ces mécanismes font en sorte qu'un programme puisse se terminer sans que ses données et son état d'exécution ne soient perdus. Ces informations de reprise peuvent être sauvegardées sur disque, éventuellement sur un serveur distant (un serveur de bases de données relationnelles, (EclipseLink, Hibernate...)). [6]

IV.1.4 Architecture J2EE :

J2EE est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

- un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'applications.
- un ensemble d'APIs qui peuvent être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

J2EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de J2EE.

➤ **Les API JEE :**

J2EE regroupe un ensemble d'API pour le développement d'applications d'entreprise. Ces API sont regroupées en trois grandes catégories mais nous ne citerons que celles que nous avons utilisées :

- ❖ les composants : dans cette catégorie nous avons utilisé :

✓ **les servlets :**

Une servlet est une classe Java qui permet de traiter une requête venant d'un client. Cette technologie doit s'occuper de traiter les données envoyées par l'utilisateur et choisir la Vue à retourner à celui-ci. On appelle cette partie : Contrôleur.

En général, la classe Java ne doit quasiment pas générer de code HTML. Les Servlets forment l'un des composants JEE les plus utilisés. Elles permettent de gérer des requêtes HTTP et de fournir au client une réponse HTTP et forment ainsi la base de la programmation Web JEE.

Les Servlets s'exécutent toujours dans un moteur de Servlet ou conteneur de Servlet permettant d'établir le lien entre la Servlet et le serveur Web. Dans notre travail, nous utiliserons Apache Tomcat.

✓ **Les JSPs :**

Les JSP (Java Server Page) sont les pages servant à générer l'ensemble du code HTML de l'interface utilisateur. On y intègre aussi bien du code HTML que des scriptlet Java (code java) ou encore des balises personnalisées (tag-lib).

Cette technologie est donc dédiée à la génération de HTML et non au traitement de la requête de l'utilisateur. On l'appelle généralement : Vue.

✓ **Java Persistence (JPA) :**

L'API de persistance Java JPA est une spécification de Sun. Fondée sur le concept POJO pour la persistance Java, elle est relativement récente.

JPA permet de mapper les objets POJO avec les tables de la base. Il devient dès lors possible d'utiliser JPA pour stocker les objets Java codés sans avoir à les sous-classer ou à implémenter une interface spécifique.

Un des grands atouts de l'API JPA est qu'elle est indépendante de tout fournisseur. Elle peut ainsi s'intégrer facilement à des serveurs d'applications JEE ou JSE (Tomcat). JPA est implémentée par deux produits de référence : TopLink, un produit commercial (Oracle) devenu libre, et Hibernate. L'architecture de JPA et son intégration dans l'architecture n-tiers sont illustrées à la figure suivante :

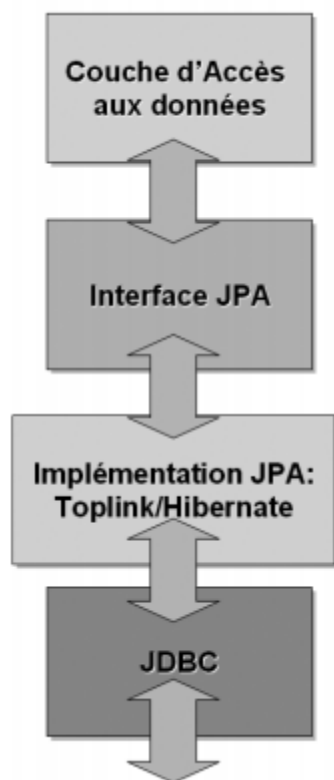


Figure I.5 : L'architecture de JPA et son intégration dans l'architecture n-tiers

La couche d'accès aux données dialoguant avec les interfaces JPA, les développements gagnent en souplesse, puisqu'il n'est plus nécessaire de changer de

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

modèle O/R ni de couche DAO (pour l'accès aux données) en fonction de l'outil de mapping utilisé. Quel que soit le produit qui implémente l'API, l'interface de la couche JPA reste inchangée.

Le mapping O/R (objet-relationnel) avec les tables de la base sont facilitées par les annotations. [8]

➤ Les services :

✓ Java DatabaseConnectivity (JDBC) :

(*Java DataBase Connectivity*) est une interface de programmation créée par Sun Microsystems — depuis racheté par Oracle Corporation — pour les programmes utilisant la plateforme Java. Elle permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC. Normalement, il s'agit d'une base de données relationnelle, et des pilotes JDBC sont disponibles pour tous les systèmes connus de bases de données relationnelles.

La figure suivante illustre parfaitement l'architecture JEE :

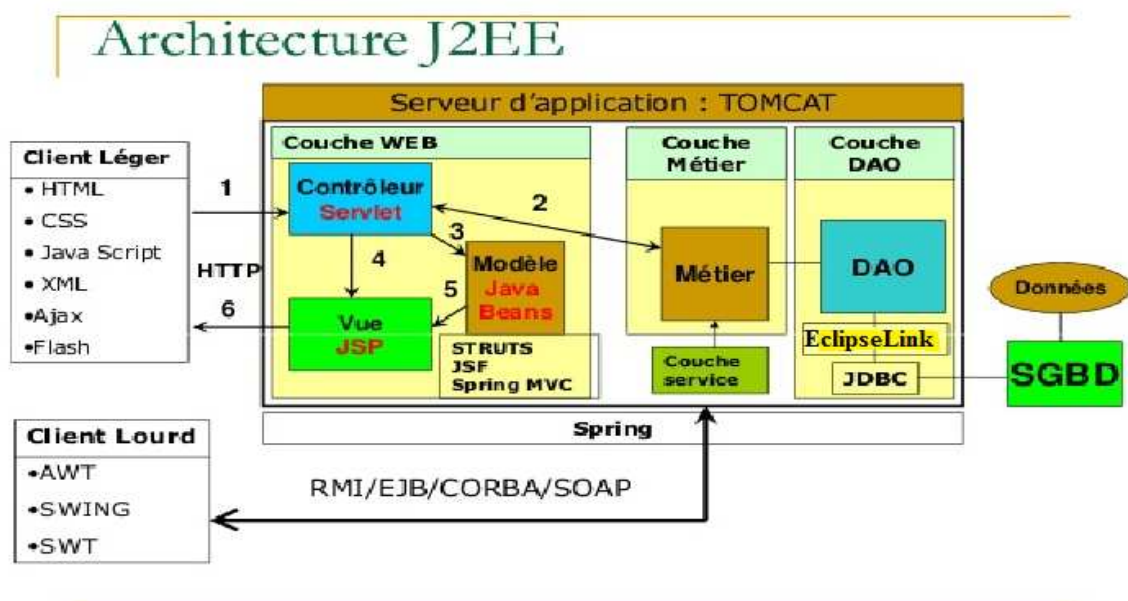


Figure I.6 : Architecture JEE et ses composants



Client lourd et client léger :

Un client **lourd** s'entend lorsque le client qui utilise l'application dispose de toute ou d'une partie de l'application sur son poste. Presque tous les traitements nécessaires à l'application sont réalisés côté client et transmis au serveur ensuite, le serveur ayant ainsi moins de choses à faire ...

- Avantages : allègement de la charge sur le serveur et réseau
- Inconvénient : gestion des versions/mise à jour chez le client, puissance de calcul nécessaire côté client.

Un client **léger** s'entend lorsque le client qui utilise l'application ne dispose de rien ou d'une infime partie de l'application sur son poste. Tous les traitements sont alors réalisés côté serveur, seules les commandes passent par le réseau.

- Avantages : poste client sans exigence de puissance, version unique gérée sur le serveur
- Inconvénient : charge forte sur le réseau et le serveur.

IV.1.5. La notion de POJO :

Spring Bean (POJO) : **POJO** est un acronyme qui signifie **plain old Java object** que l'on peut traduire en français par ***bon vieil objet Java***. Cet acronyme est principalement utilisé pour faire référence à la simplicité d'utilisation d'un objet Java. En théorie, un POJO est un objet Java lié à aucune autre restriction que celles forcées par la spécification du langage Java. En d'autres termes, il est impératif qu'un POJO :

1. n'étende pas des classes pré-spécifiées
2. n'implémente pas des interfaces pré-spécifiées
3. ne contienne pas des annotations pré-spécifiées

Toutefois, à cause de difficultés techniques et d'autres raisons, plusieurs programmes ou *frameworks* décrits comme conformes à POJO nécessitent encore l'utilisation des annotations pré-spécifiées pour les caractéristiques telles que la persistance pour un fonctionnement correct. L'idée, c'est que si l'objet (en fait la classe) était un POJO avant d'ajouter toute annotation, et revient à l'état de POJO si les annotations ont été éliminées, alors il peut encore être considéré comme un POJO. Ainsi, l'objet de base reste un POJO.

IV.1.6. Présentation du modèle de conception MVC :

Nous retrouvons ce modèle dans la couche présentation de l'architecture JEE. Le Model-View-Controller (MVC) est un modèle de conception logicielle très répandu et fort utile. Créé dans les années 80 par Xerox PARC pour Smalltalk-80, il est aujourd'hui fortement recommandé dans l'univers J2EE. Néanmoins il faut retenir que c'est un modèle de conception ; il est donc indépendant du langage de programmation.

IV.1.6.1 Définitions :

- En informatique, et plus particulièrement en développement logiciel, un **patron de conception** (plus souvent appelé *design pattern*) est un arrangement caractéristique de modules, reconnu comme bonne pratique en réponse à un problème de conception d'un logiciel. Il décrit une solution standard, utilisable dans la conception de différents logiciels.

IV.1.6.2 Principe :

Le modèle MVC est un design pattern (modèle de conception ou DP) de conception d'interface utilisateur permettant de découpler le modèle (logique métier et accès aux données) des vues (interfaces utilisateur [présentation des données et interface de saisie pour l'utilisateur]). Des modifications de l'un n'auront ainsi, idéalement, aucune conséquence sur l'autre ce qui facilitera grandement la maintenance tels que :

Modèle: gère les données et reprend la logique métier (le modèle lui-même peut être décomposé en plusieurs couches mais cette décomposition n'intervient pas au niveau de MVC). Le modèle ne prend en compte aucun élément de présentation!

Vue: elle affiche les données provenant exclusivement du modèle, pour l'utilisateur et/ou reçoit ses actions. Aucun traitement – autre que la gestion de présentation - n'y est réalisée.

Contrôleur: son rôle est de traiter les événements en provenance de l'interface utilisateur.

La figure suivante décrit une application trois tiers avec le modèle MVC de type 1 :

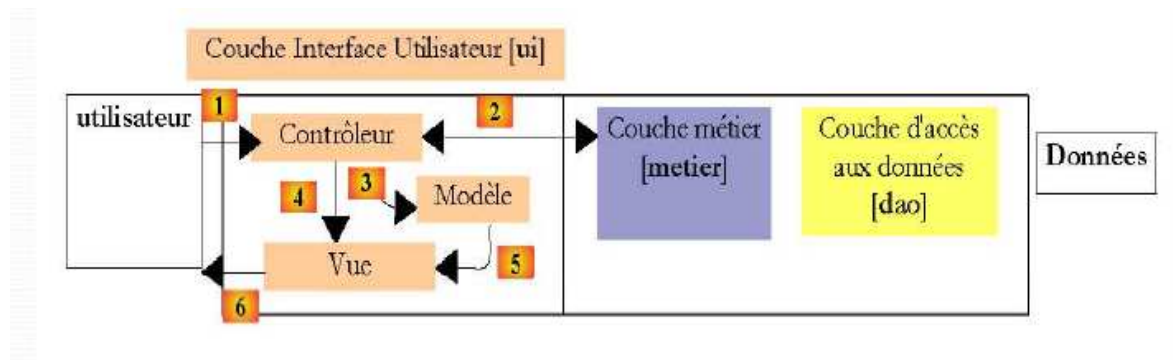


Figure I.7 : Architecture d'une application web avec le modèle MVC

IV.1.6.3 Cinématique

- L'utilisateur émet une requête
- Le contrôleur intercepte la requête de l'utilisateur
- Le contrôleur détermine quelle partie du modèle est concernée et quelle vue y est associée
- Le modèle traite les interactions avec les données, applique les règles métier et renvoie les données au contrôleur
- Le contrôleur sélectionne la vue et lui renseigne les données
- La vue présente les données à l'utilisateur

Dans ce modèle, chaque requête est traitée par un contrôleur sous la forme d'une servlet. Celle-ci traite la requête, fait appel aux éléments du model si nécessaire et redirige la requête vers une JSP qui se charge de créer la réponse à l'utilisateur.

Le principal défaut du modèle MVC1 est le nombre de servlets à développer pour une application.

Le MVC1 très pratique, peut se révéler lourd à mettre en place. Ceci à cause de la multitude de contrôleurs à implémenter. Afin de simplifier la réalisation d'un tel modèle, une nouvelle version a été introduite : le MVC2. C'est exactement le même modèle de Conception à la différence qu'il n'y a plus qu'un seul contrôleur qui se charge de rediriger la requête vers le bon traitement.

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

Pour simplifier les choses, le modèle MVC model 2 ou MVC2 de Sun propose de n'utiliser qu'une seule et unique servlet comme contrôleur. Cette servlet se charge d'assurer l'explication des traitements en fonction des requêtes http reçues.

Le modèle MVC 2 est donc une évolution du modèle 1 : une unique servlet fait office de contrôleur et gère toutes les requêtes à traiter en fonction d'un paramétrage généralement sous la forme d'un fichier au format XML.

La figure suivante décrit une application trois tiers avec le modèle MVC de type 2 :

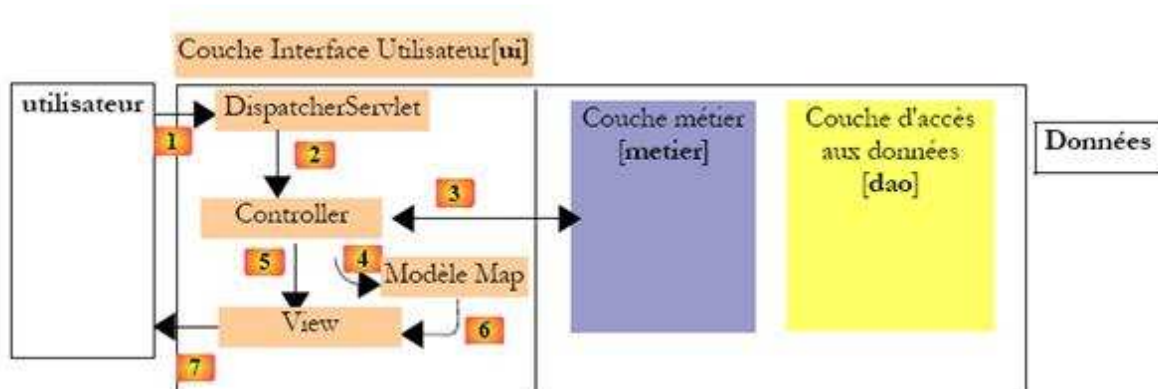


Figure I.8 : Architecture d'une application web avec le modèle MVC 2

V. Introduction au Framework spring :

Spring est le Framework le plus populaire dans le développement d'applications d'entreprises ; il est utilisé par des millions de développeurs dans le monde et cela pour de nombreuses raisons telles que la possibilité de réutilisation de code et ses grandes performances. Spring est un Framework open source Java Plateforme et il a été initialement écrit par Rod Johnson et a été publié la première fois sous la licence Apache 2.0 en juin 2003.

V.1. Architecture du Framework Spring :

Spring est conçu comme une sorte de boîte à outils, au contraire d'autres frameworks. Il vous laisse libre de n'utiliser que l'un ou l'autre de ses modules.

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

Spring est d'ailleurs disponible sous deux formes, celle d'un jar (librairie Java) unique et celle de plusieurs fichiers jar permettant de ne rajouter au projet que la partie que

l'on souhaite utiliser (i.e. SpringCore, SpringRemoting, etc...) :

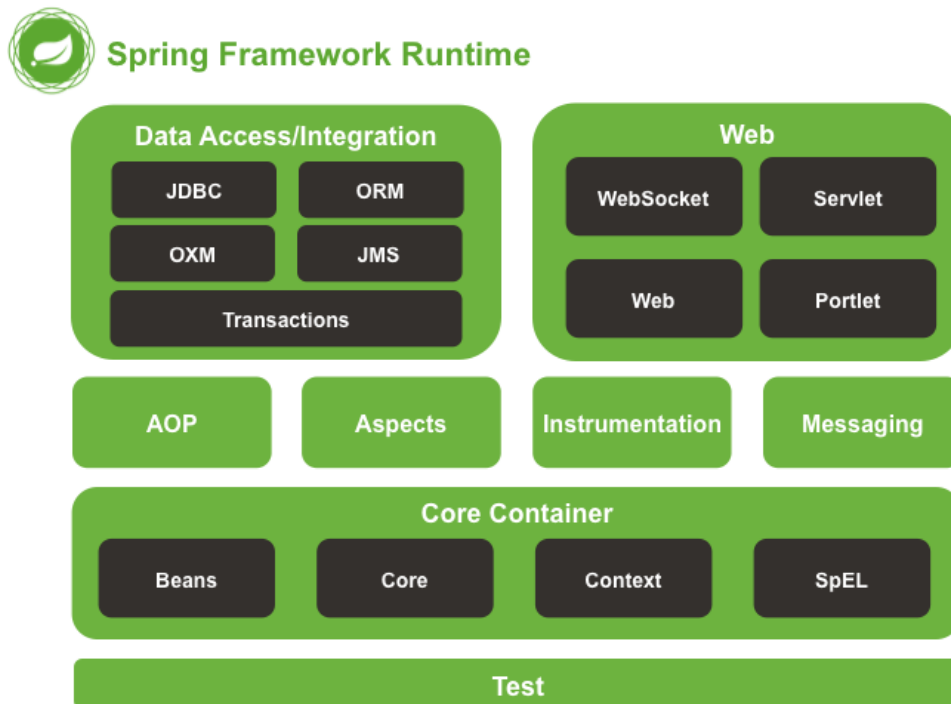


Figure I.9 : Architecture du Framework Spring

Le Framework est organisé en modules, reposant tous sur le module SpringCore:

1. **SpringCore** : implémente notamment le concept d'inversion de contrôle (injection de dépendance). Il est également responsable de la gestion et de la configuration du conteneur.

➤ **Principe de l'inversion de contrôle :**

L'IoC est un principe abstrait qui définit un motif de conception dans lequel le flux de contrôle d'un système est inversé par rapport à un développement procédural. L'inversion de contrôle est assurée de deux façons différentes :

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

- ❖ **La recherche de dépendance** : consiste pour un objet à interroger le conteneur, afin de trouver ses dépendances avec les autres objets.
- ❖ **L'injection de dépendances** : cette injection peut être effectuée de trois manières possibles :
 - L'injection de dépendance via le constructeur.
 - L'injection de dépendance via les modificateurs (setters).
 - L'injection de dépendance via une interface.

L'injection de dépendances est un motif de conception qui propose un mécanisme pour fournir à un composant les dépendances dont il a besoin. C'est une forme particulière d'inversion de contrôle.

L'injection de dépendances permet à une application de déléguer la gestion du cycle de vie de ses dépendances et leurs injections à une autre entité. L'application ne crée pas directement les instances des objets dont elle a besoin : les dépendances d'un objet ne sont pas gérées par l'objet lui-même mais sont gérées et injectées par une entité externe à l'objet.

Dans le cas classique, l'objet invoque le constructeur de ses dépendances pour obtenir les instances requises en utilisant l'opérateur new. Cela induit un couplage fort entre l'objet et sa dépendance. Pour réduire ce couplage, il est possible par exemple de définir une interface et d'utiliser une fabrique pour créer une instance mais cela nécessite beaucoup de code.

Avec le motif de conception IoC, la gestion des objets est confiée à un objet dédié. Celui-ci se charge de créer les instances requises et de les fournir par injection. Cette injection peut concrètement se faire de plusieurs manières :

- passer la ou les instances en paramètre du constructeur
- fournir l'instance en invoquant le setter d'une propriété
- fournir la ou les instances en paramètre de la fabrique invoquée pour créer l'instance

CHAPITRE I : PRESENTATION DE L'ARCHITECTURE JEE

Les classes et les interfaces de base du conteneur Spring sont contenues dans les packages `org.springframework.beans` et `org.springframework.context`.

Une configuration permet de définir les objets qui sont gérés par le conteneur, généralement sous la forme d'un fichier XML : les informations fournies permettent au conteneur d'instancier et d'initialiser l'objet et ses dépendances. [9]

2. **SpringContext**: Ce module étend `SpringCore`. Il fournit une sorte de base de données d'objets, permet de charger des ressources (tels que des fichiers de configuration) ou encore la propagation d'évènements et la création de contexte comme par exemple le support de Spring dans un conteneur de Servlet.
3. **Spring AOP** : Permet d'intégrer de la programmation orientée aspect.
4. **Spring DAO** : Ce module permet d'abstraire les accès à la base de données, d'éliminer le code redondant. Il fournit en outre une gestion des transactions.
5. **Spring ORM** : Cette partie permet d'intégrer des frameworks de mapping Object/Relationnel tel que Hibernate, JDO ou iBatis avec Spring. La quantité de code économisé par ce package peut être très impressionnante (ouverture, fermeture de session, gestion des erreurs)
6. **SpringMVC** : Implémentation du modèle MVC, que nous avons présenté précédemment. La figure suivante illustre le fonctionnement de Spring MVC. [10]

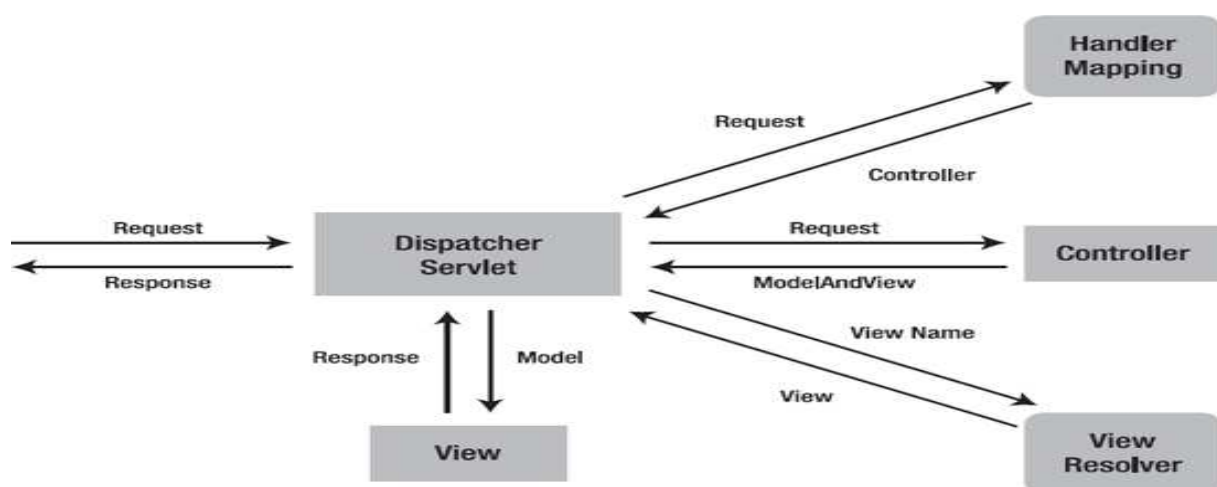


Figure I.10 : Fonctionnement de spring MVC

Les étapes sont :

1. Le DispatcherServlet reçoit une requête.
2. Le DispatcherServlet consulte son Handler Mapping pour connaître le contrôleur, En effet avec Spring MVC, vous créez vos contrôleurs.
3. Le DispatcherServlet dispatche la requête au contrôleur identifié.
4. Le contrôleur retourne au DispatcherServlet un objet de type ModelAndView possédant comme paramètre au minimum le nom logique de la vue à renvoyer.
5. Le DispatcherServlet consulte son ViewResolver lui permettant de trouver la vue.
6. Le DispatcherServlet récupère la vue associée à la requête puis la renvoie comme réponse.

Spring est considéré comme un conteneur dit « léger ». La raison de ce nommage est expliquée par Erik Gollot dans l'introduction du document *Introduction au frameworkSpring*

« SPRING est effectivement un conteneur dit “ léger ”, c’est-à-dire une infrastructure similaire à un serveur d'applications J2EE. Il prend donc en charge la création d’objets et la mise en relation d’objets par l’intermédiaire d’un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ces objets. Le gros avantage par rapport aux serveurs d’application est qu’avec SPRING, les classes n’ont pas besoin d’implémenter une quelconque interface pour être prises en charge par le framework. C’est en ce sens que SPRING est qualifié de conteneur “ léger ”. »

Spring est un Framework modulaire, il permet aux développeurs d'utiliser les modules dont ils ont besoin et de laisser le reste. Il peut s'intégrer dans toutes les couches de l'application. Il est considéré comme une grande boîte à outils dont on se sert au besoin.

Conclusion :

Dans ce chapitre nous avons abordé les notions de bases qui nous ont permis de comprendre le fonctionnement du modèle client/serveur, l'architecture générale des applications web, et enfin l'architecture J2EE et l'un de ses framework (Spring) les plus utilisés dans le monde.

Chapitre II

Présentation de l'organisme d'accueil



Introduction

Afin de vous permettre de vous familiariser avec notre domaine d'application, nous allons présenter dans ce chapitre l'entreprise ENIEM.

I. Situation géographique

Le siège social de l'ENIEM se situe au chef-lieu de la wilaya de Tizi-Ouzou. Les unités de production sont implantées dans la zone industrielle de Oued-Aissi, distantes de 7km du chef-lieu de wilaya, sur l'axe reliant la ville de Tizi-Ouzou et la commune de Tizi-Rached.

II. Objectif social et champs d'étude de l'entreprise ENIEM :

ENIEM est leader de l'Electroménager en Algérie, elle possède des capacités de production et une longue expérience dans le domaine dans la fabrication et le développement dans les différentes branches de l'électroménager, notamment :

- Les appareils ménagers domestiques.
- Les appareils de collectivités.
- Les lampes d'éclairage.
- Les produits sanitaires. [11]

II. Historique

ENIEM est entrée en production à partir de janvier 1977 dont l'activité principale est la fabrication et la commercialisation d'appareils électroménagers.

L'entreprise est certifiée ISO 9001/2008 QUALITÉ et ISO 14001/2004 ENVIRONNEMENT. [12]

Année	Faits marquants
De 1977 à 1981	-Démarrage de la fabrication des réfrigérateurspetit modèle en collaboration avec le partenaire allemand RIAG-BOCH/1977. - Démarrage de l'ancienne usine de cuisinière de réchauds à plat en

Chapitre II : présentation de l'organisme d'accueil

	<p>collaboration avec le partenaire allemand RIAG-SPELL-FRENCKE en 1977.</p> <ul style="list-style-type: none">- Montage du premier climatiseur de type FRENCK en 1978.- Entrée en production des chauffes eau\bain en collaboration avec le partenaire allemand BOCH-CHAFFOTEAUX, le partenaire français MAURY et le partenaire Espagnol en 1979.
De 1982 à 1986.	<ul style="list-style-type: none">- Introduction de montage de petits appareils de ménagers MAC, SC en 1982.- Entrée en exploitation de nouvelle usine de réfrigérateur et congélateur grands modèles en collaboration avec le partenaire japonais MITSUI-TOSHIBA en 1986.- Elargissement de la gamme de climatisation du SPLIT système en 1986.
De 1987 à 1989.	<ul style="list-style-type: none">- Abandon de la production de PAM, cédée aux sections privées en 1987.- Montage des armoires frigorifiques (CAF) en 1988.- Arrêt de production de réchaud à plat à deux feux en 1989.- Passage de l'ENIEM à l'autonomie conformément à la loi 88-01 portant sur l'orientation des EPE en octobre 1989.
De 1990 à 1991	<ul style="list-style-type: none">-Arrêt de la production des anciennes cuisinières et vente des équipements en 1990.- Entrée en production de l'usine de cuisinière avec le partenaire italien INTERCOOP-TECHNOGAZ en 1991.
De 1992 à 1993.	<ul style="list-style-type: none">-Mise en place de l'usine des congélateurs horizontaux (BAHUT) avec le partenaire libanais LEMATIC en 1992.- Reprise des montages des petits appareils ménagers de type MAC, SC, THV et mixeurs avec le partenaire Espagnol TAUBEN-TAURIS en 1993.

Chapitre II : présentation de l'organisme d'accueil

De 1994 à 1997.	<ul style="list-style-type: none">- Réalisation des réchauds à plat quatre feux en 1994.- Abandon de la production CAF en 1994.- Entrée en production des réfrigérateurs 5201 en 1995.- Montage de petits appareils ménagers BLENDER et ventilateurs en collaboration avec le partenaire italien ITALISMAMP en 1995.- Entrée en production des PAM (SC et MAC) avec le partenaire italien ITALISMAMP en 1996.-Reprise de la production des comptoirs frigorifiques CAF.-Abandon de la production des réchauds à plat à quatre feux en 1997.
De 1998 à 2000	<ul style="list-style-type: none">- Compte tenu de l'importance de son activité, la direction de l'ENIEM a procédé la restriction du CAM en sept unités à savoir ; unité siège ; unité commercial ; trois unités de production (froid, cuisson et climatisation) ; deux unités de prestation (technique et service).- Certification ISO-9001 de l'ENIEM par l'organisme français AFAQ-ASSERT.- Suppression de l'unité prestation de service en 2002.

Tableau II.1 : Historique de l'ENIEM

III. Organisation générale :

A partir de janvier 1998, l'Entreprise s'est réorganisée en centre d'activités stratégiques qui s'articulent autour de la restructuration du complexe d'appareils ménagers créant plusieurs unités de production et de soutien.

IV. Organigramme général de l'ENIEM :

La figure ci-dessous montre l'organisation générale de l'ENIEM qui est représentée par un organigramme [13]

Chapitre II : présentation de l'organisme d'accueil

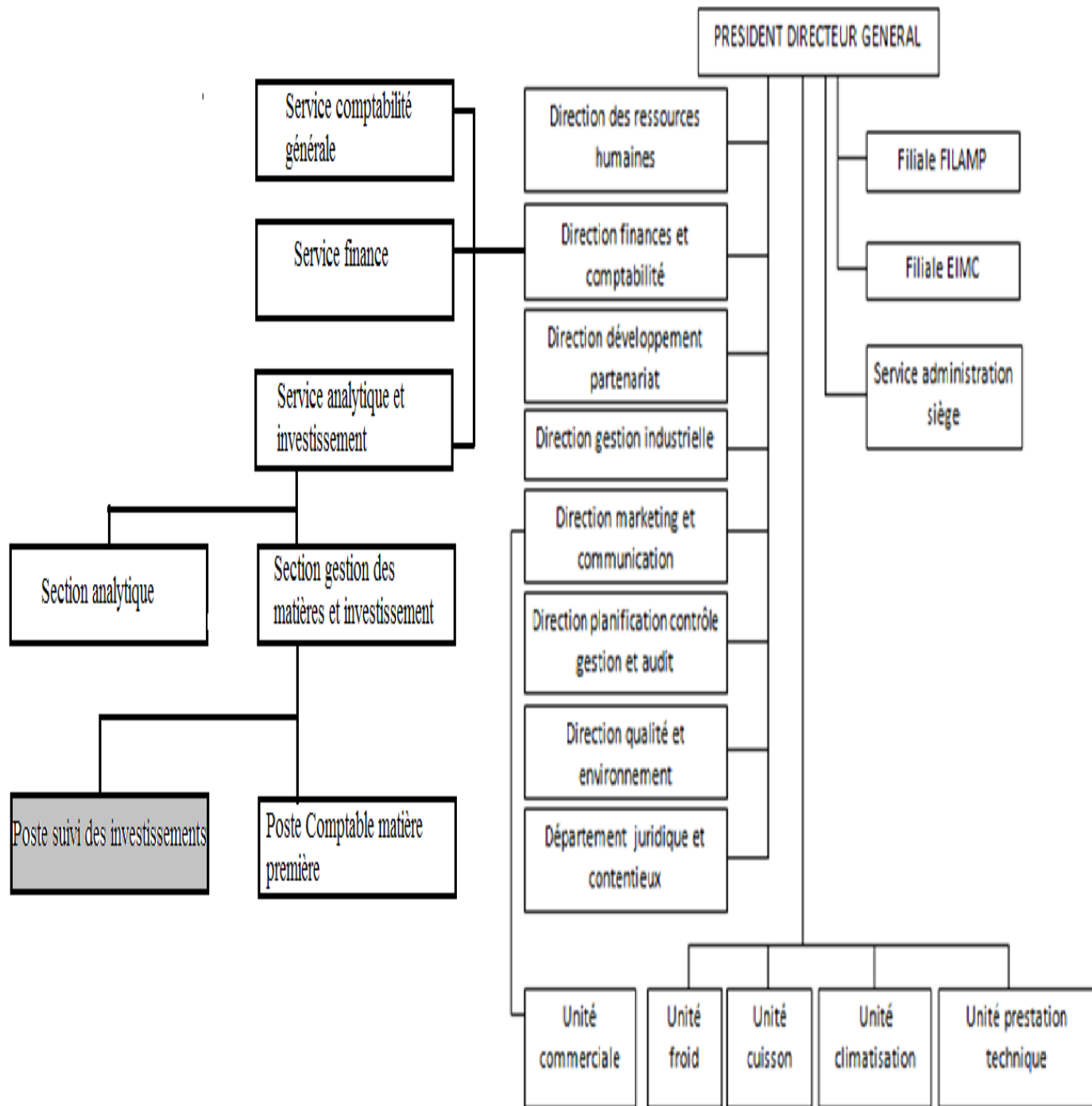


Figure II.11 : Organigramme Général de l'ENIEM

Direction générale :

Elle est responsable de la stratégie et veille au développement de l'entreprise. Elle exerce son autorité hiérarchique sur les autres directions.

Chapitre II : présentation de l'organisme d'accueil

Direction des ressources humaines :

Est chargée de mettre en œuvre la gestion des emplois, des contrats, des carrières, des dossiers de pension, de la paie et de la formation de tous les personnels, elle pilote aussi les opérations de gestion collective (recrutement, avancement...).

Direction marketing et communication :

Elle permet une analyse des ventes, donne des objectifs commerciaux et stratégiques, participe à la définition du plan marketing.

Direction développement et partenariat :

Permet de coordonner toutes les actions visant à développer la collecte de fonds au profit d'actions de recherche et formation de l'entreprise.

Direction planification et statistique organisationnelle

Veille à améliorer les performances des structures de l'entreprise, le contrôle des achats, approvisionnement, délais de paiement etc.

Direction d'exploitation

Elle a comme mission la mise en œuvre de stratégies afin d'atteindre des objectifs de qualité de service.

Direction finance et comptabilité

Elle a pour objectif d'élaborer les prévisions budgétaires et de tenir une bonne comptabilité.

Direction développement industriel

Elle est chargée de l'élaboration et de la mise en œuvre d'axes d'optimisation en vue d'augmenter la productivité ou d'améliorer les conditions de travail.

Unité cuisson :

Elle a pour mission de produire et de développer les produits de cuisson à gaz (cuisinière, four, ou autre produit de technologie similaire)

Chapitre II : présentation de l'organisme d'accueil

Activités :

- Transformation de la tôle.
- Traitement et revêtement de surface (émaillage, chromage...).
- Montage final.

Unité froid :

Spécialisée dans la production des équipements de froid domestique tels que réfrigérateurs, congélateur.

Activités :

- Transformation de la tôle.
- Traitement et revêtement de surface.
- Injection plastique et polystyrène.
- Fabrication de pièces métalliques (condenseurs, évaporateurs).
- Thermoformage.
- Assemblage.

Unité climatisation

Spécialisée dans la production des équipements de climatisation et de chauffage tels que chauffe bain, radiateur à gaz de butane, comptoirs et armoires frigorifique, équipement de climatisation pour les particuliers et industriels.

Activités :

- Transformation de la tôle.
- Traitement et revêtement de surface (peinture, plastification).
- Assemblage.

Unité prestation technique :

Chargée de fournir des prestations techniques et des services nécessaires aux unités de production.

Activités :

- La distribution et la commercialisation des produits électroménagers.

Chapitre II : présentation de l'organisme d'accueil

- Traitement de la vente de produits dans le cadre des marchés suite aux appels d'offres.
- Le stockage des produits finis.
- L'assurance du service après-vente pour les produits à travers son réseau d'agents agréés à travers le territoire national.

Service suivie des investissements : il se décompose en deux sections,

- Section analytique
- Section gestion des matières et investissements qui a son tour possède deux postes, le premier étant « poste comptable matière première, et le second qui représente notre champ d'étude le poste « suivi des investissements »

V. Présentation du champ d'étude :

Notre champ d'étude est le service suivie des investissements (immobilisation), il est rattaché à la direction finance et comptabilité, après avoir intégré l'équipe de prestation technique qui a pour principale mission de fournir les outils nécessaires aux différents services de l'ENIEM, nous avons été affecté à la direction comptabilité et finance afin de fournir un outil qui apporte une solution meilleure que celle déjà existante d'où la nécessité de définir certaines règles de la comptabilité des immobilisations ainsi que quelques mots techniques.

V.1. Définition des mots techniques :

V.1.1. Définition de l'immobilisation :

Les immobilisations sont des biens dont l'utilisation s'étale sur plusieurs années contrairement aux matières premières qui se consomment durant un cycle de production.

Les immobilisations sont classées selon leur origine (immobilisation acquises ou immobilisations produites) ou leur classification au bilan (immobilisation corporelles, incorporelles, financières).

Chapitre II : présentation de l'organisme d'accueil

- Immobilisation corporelle : équipements, machines, locaux utilisés dans les processus de production physique de l'entreprise.
- Immobilisation incorporelle : fonds de commerce, logiciels, brevets, procédés, frais de recherche et développement, ensemble des moyens non matériels, savoir-faire et compétence qui permettent l'activité de l'entreprise.
- Immobilisation financières : ce sont des titres financiers destinés à rester durablement dans le patrimoine de l'entreprise ou à permettre le contrôle d'autres entreprises, (titres immobilisées, titres de participation). [14]

Nous nous intéressons dans notre étude aux immobilisations corporelles.

V.1.2. Comptabilisation :

Trois phases peuvent être distinguées :

- La comptabilisation de l'acquisition de l'immobilisation au moment où elle a lieu.
- La répartition du coût d'acquisition sur la période d'utilisation.
- La réévaluation du bilan.

V.1.3. L'amortissement :

Il existe plusieurs types d'amortissement, mais notre champ d'étude n'est concerné que par l'amortissement linéaire qui désigne l'une des deux méthodes d'amortissement d'un bien qui permet à une entreprise de constater, de manière comptable, la dépréciation ou la perte de valeur d'un bien au fil des années et de son utilisation.[15]

V.1.4 Formules de calcul :

➤ La valeur d'origine pour les immobilisations :

Amortissement = VO * taux amortissement = VO / durée de vie

➤ La valeur d'origine diminuée de la valeur résiduelle :

Amortissement = (VO – valeur résiduelle)/ durée de vie

Amortissement = (VO – valeur résiduelle)* taux amortissement

Chapitre II : présentation de l'organisme d'accueil

La valeur résiduelle est la valeur attendue de la vente probable d'une immobilisation à l'issue de la période d'amortissement (durée de vie de l'immobilisation)

La valeur résiduelle nette ; c'est-à-dire diminuée des frais de vente, est soustraite de la valeur d'acquisition pour déterminer la valeur amortissable.

➤ **L'annuité :**

Annuité = VO * taux d'amortissement

➤ **Valeur Nette Amortie :**

VNA = VO - Amortissement cumulé.

V.1.5. Cycle de vie d'une immobilisation :

V.1.5.1. Acquisition de l'immobilisation :

C'est l'entrée de l'investissement, elle est due à :

- achat externe.
- cession inter unité.
- le transfert.
- réalisation ou production de l'immobilisation.

V.1.5.2. Achat externe :

Tout achat externe d'immobilisation est réalisé suivant les procédures citées. Après codification de l'immobilisation acquise, celle-ci est saisie sur l'application informatique.

Les informations saisies sont :

- Code de l'immobilisation.
- Désignation de l'immobilisation.
- Date d'acquisition.
- Fournisseur de l'immobilisation.
- Affectation (centre de frais).
- Valeur d'origine.
- Compte comptable.
- Marque, type, série.

-Date de mise en service.

V.1.5.3. Les cessions inter unités :

Les informations à fournir sont identiques à celles de l'achat externe, avec l'ajout des amortissements cumulés et déterminer le tableau d'amortissement sur la base de la Valeur Nette Comptable transférée et la durée d'utilité restante

V.1.5.4. Transfert de l'immobilisation :

Transférer une immobilisation d'un centre de frais vers un autre centre de frais, dans ce cas il y a modification de l'affectation de l'immobilisation.

V.1.5.5. La sortie de l'immobilisation :

On distingue quatre cas :

-La vente.

-Destruction.

-cession.

-transfert.

-perte

V.2 Mission du champ d'étude :

Le poste suivi des investissements a pour missions :

- Ré-estimation des biens en cas de réévaluation
- Suivi des investissements et leurs amortissements, par centre de frais et par compte comptable.

V.3. Situation informatique du champ d'étude :

- **Aspect matériel :** la section de gestion de matière et investissement est équipé du matériel informatique suivant :
 - **Désignation du matériel : MICRO ORDINATEUR**
 - **Caractéristiques :**
 - Processeur : processeur pentium® Dual-Core CPU E5200 @ 2.5GHZ
 - La RAM : 1G
 - Nombre : 01
- **Aspect logiciel :** Les logiciel installés sur le matériel cité est :

- Microsoft Windows seven.

V.4. Etude des postes de travail et des documents :

V.4.1. Etude des postes de travail :

Un poste de travail est caractérisé par sa désignation, son lieu d'implantation, les responsabilités et les tâches effectuées par la personne occupant ce poste.

L'étude des postes est une étape importante afin de pouvoir comprendre les relations entre les postes, repérer tous les documents utilisés par ce poste, et enfin spécifier les tâches et les responsabilités de ce poste.

Dans notre cas il n'y a qu'un seul poste de travail à étudier qui est le poste de **comptable analytique et investissement**, il est rattaché au service finance et comptabilité, tel qu'il est composé d'un seul agent.

❖ Fiche d'étude du poste de travail :

Désignation : comptable analytique et investissement

Service de rattachement : finance et comptabilité

Nombre d'agents effectifs : 01

Tâches et responsabilité :

1. Responsabilités :

- Assurer le bon fonctionnement du suivi des investissements
- Effectuer les opérations comptables

2. Tâches

- Identifications de l'investissement sur terrain
- Attribuer un numéro d'inventaire
- Ouvrir une fiche d'investissement
- Justifier l'investissement sur le registre
- Procéder à l'inventaire des investissements
- Comparer les résultats d'inventaire à la situation comptable et dégager les écarts
- Calcul d'amortissement (établissement du tableau d'amortissement)
- Procéder à la réévaluation des investissements

V.4.2. Etude des documents :

Un document est un élément d'information dont l'utilisation est régulière, il joue un rôle essentiel dans la plupart des sociétés, tant pour le fonctionnement de leur administration que dans l'élaboration de leurs savoir.

Il existe deux types de document :

- **Documents internes** : Ce sont tous les documents internes à l'entreprise et qui circulent seulement dans le champ d'étude.
- **Documents externes** : Ce sont tous les documents qui sont destinés ou qui proviennent de l'extérieur du champ d'étude.

V.4.3. Liste des documents :

➤ Documents parvenant au poste :

1. Facture.
2. Fiche de mise à disposition
3. Avis de cession
4. Note débit/ crédit
5. Fiche d'investissement
6. Décision d'écart
7. PV de réforme

➤ Document émis par le poste

1. Fiche état d'écart
2. Fiche état d'inventaire
3. Fiche état de réévaluation
4. Fiche d'investissement

➤ Documents établit par le poste

1. Fiche d'investissement
2. Fiche d'état d'amortissement
3. Fiche état de réévaluation

4. Fiche état d'inventaire

V.5. Critiques et suggestions :

L'étude du système existant nous a permis de déceler certaines insuffisances d'où la nécessité de suggérer et de proposer des améliorations de façon à déterminer des solutions futures.

V.5.1. Les anomalies

- Beaucoup de difficulté pour maintenir l'application existante.
- Absence d'un administrateur pour gérer les utilisateurs de l'application existante.

V.5.2. les solutions :

- Mettre en place l'architecture J2EE, pour pouvoir structurer les code et l'organiser afin de faciliter la maintenance.
- Rendre l'application accessible à travers une page web après l'avoir déployer sur le serveur local.
- Désigner un administrateur pour gérer les utilisateurs.
- Mettre en place, une interface simple et conviviale

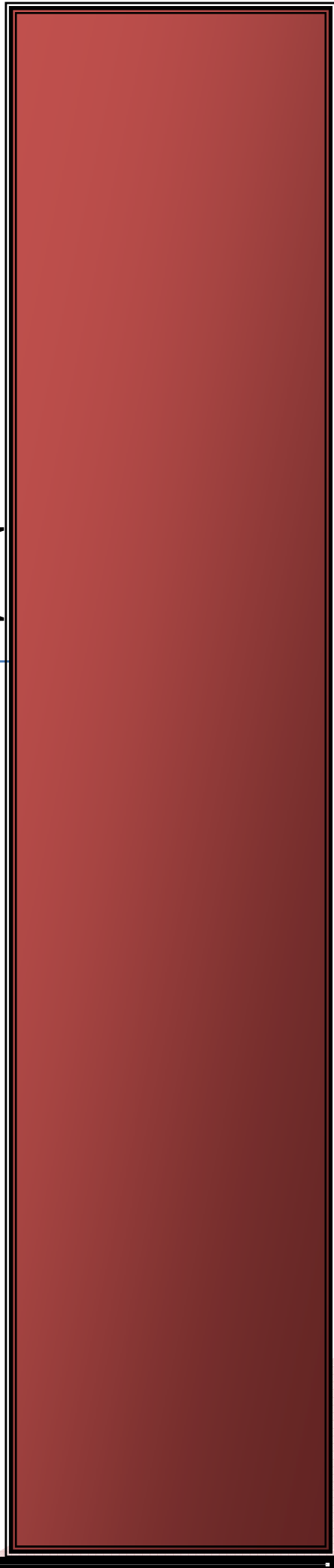
Conclusion :

Dans ce chapitre nous avons étudié notre organisme d'accueil en générale et le poste suivi des investissements en particulier.

Nous avons étudié les postes de travail, les documents, les fichiers et les registres ainsi que la codification existante, ce qui nous a permis de cerner les problèmes et les insuffisances et de proposer des solutions.

Chapitre III

Analyse et conception



Introduction :

Le développement d'une application ou d'un système informatique requière l'usage d'une méthodologie afin d'assurer une organisation consciencieuse et de pouvoir cerner les tâches à accomplir.

Dans tout projet informatique la conception est importante et doit être traitée avec précision et en détail, précédée d'une analyse profonde et réfléchie, car elle est le reflet du système avant même sa concrétisation. L'approche orientée objet s'est avérée un modèle d'analyse et de conception très puissant et se trouve de plus en plus utilisée. Dans ce chapitre, notre objectif est de modéliser une application de gestion des immobilisations de l'entreprise ENIEM, précisément pour le service finance et comptabilité, à l'aide du langage UML (UnifiedModelingLanguage) qui permet de bien représenter les aspects de notre projet par la série des diagrammes qu'il offre. Ensuite la conception de la base de données utile à notre application est présentée à la fin de ce chapitre.

I.1Analyse :

I.1.1. L'étape d'analyse :

L'étape d'analyse est primordiale pour la réalisation et le développement d'une application, ou d'un système informatique, elle permet de faire une étude approfondie du domaine que couvre notre application et cela dans le but de cerner, de comprendre et de définir les besoins.

I.1.2. Objectif du travail :

Le principal objectif de notre application est de répondre aux besoins fonctionnels de l'entreprise qui est de faciliter la procédure de gestion des immobilisations en fournissant un outil qui permet de gérer les biens de l'entreprise à partir de leurs acquisition (entrée) jusqu'à leurs sortie, appliquer les règles de calculs dédiées à la comptabilité des immobilisations, en plus des exigence fonctionnelles nous allons répondre à quelque besoins non fonctionnels, comme la portabilité, l'extension, et la maintenabilité.

I.1.3. Démarche de modélisation :

Pour la réalisation de notre projet nous avons opté pour une démarche guidée par le besoin de l'utilisateur du système, car c'est le but principal du développement, tel que chaque étape sera affinée et validée en fonction des besoins des utilisateurs. La démarche de modélisation choisie pour concevoir notre application peut être représentée graphiquement comme suite :

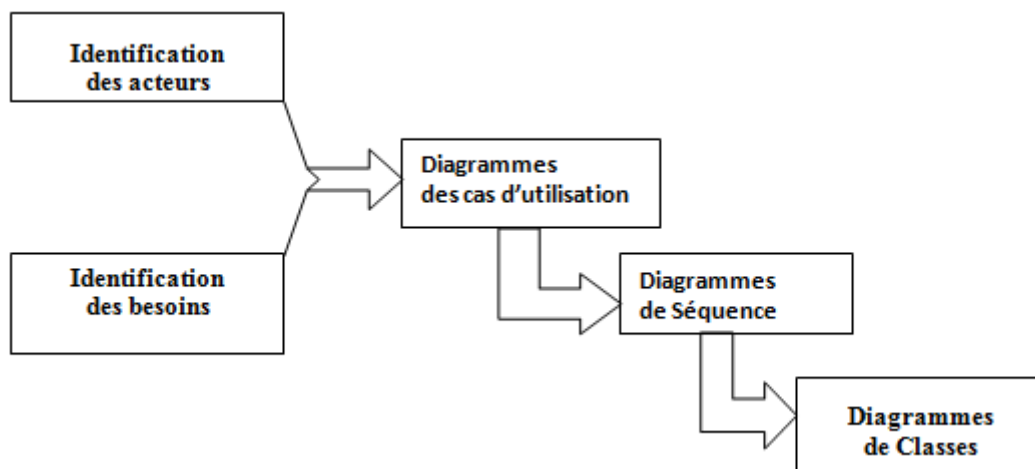


Figure III.11 : La démarche de modélisation de l'application

I.1.3.1. Les acteurs :

Durant la période de notre stage au sein du service comptabilité de l'ENIEM, nous avons procédé à l'identification des principaux acteurs qui seront les utilisateurs de notre application, et nous les avons classés en deux types, d'un côté nous avons l'administrateur et de l'autre les simples utilisateurs.

- **Administrateur** : l'administrateur de notre application a pour principal rôle la gestion des utilisateurs mais peut effectuer les mêmes tâches que l'utilisateur.
- **Utilisateur** : l'utilisateur peut effectuer les différentes tâches précisées dans les prochains titres de ce chapitre.

I.1.3.2. Spécification des tâches des acteurs : Le tableau suivant représente les tâches pour chaque acteur:

CHAPITRE III : analyse et conception

ACTEUR : « UTILISATEUR, ADMINISTRATEUR »

Acteur	Tâches
Utilisateur	<ul style="list-style-type: none">➤ S'authentifier➤ Gestion des biens<ul style="list-style-type: none">• Ajouter un bien• Rechercher un bien par code<ul style="list-style-type: none">○ Modifier un bien○ Supprimer un bien○ Effectuer le transfert d'un bien○ Effectuer la cession d'un bien○ Effectuer la sortie du bien○ Effectuer la réévaluation d'un bien• Visualiser les biens d'un compte• Visualiser les biens d'un centre de frais➤ Gestion des comptes<ul style="list-style-type: none">• Ajouter un compte• Rechercher un compte<ul style="list-style-type: none">○ Modifier le compte○ Supprimer le compte➤ Gestion des unités<ul style="list-style-type: none">• Ajouter une unité• Rechercher une unité<ul style="list-style-type: none">○ Modifier l'unité○ Supprimer l'unité➤ Gestion des fournisseurs<ul style="list-style-type: none">• Ajouter un fournisseur• Rechercher un fournisseur

CHAPITRE III : analyse et conception

	<ul style="list-style-type: none">○ Modifier○ Supprimer➤ Se déconnecter
Administrateur	<ul style="list-style-type: none">➤ S'authentifier➤ Gestion des biens<ul style="list-style-type: none">• Ajouter un bien• Rechercher un bien par code<ul style="list-style-type: none">○ Modifier un bien○ Supprimer un bien○ Effectuer le transfert d'un bien○ Effectuer la cession d'un bien○ Effectuer la sortie du bien○ Effectuer la réévaluation• Visualiser les biens d'un compte• Visualiser les biens d'un centre de frais➤ Gestion d'un compte<ul style="list-style-type: none">• Ajouter un compte• Rechercher un compte<ul style="list-style-type: none">○ Modifier le compte○ Supprimer le compte➤ Gestion des unités<ul style="list-style-type: none">• Ajouter une unité• Rechercher une unité<ul style="list-style-type: none">○ Modifier l'unité○ Supprimer l'unité➤ Gestion des fournisseurs<ul style="list-style-type: none">• Ajouter un fournisseur• Rechercher un fournisseur

CHAPITRE III : analyse et conception

	<ul style="list-style-type: none">○ Modifier○ Supprimer➤ Gestion des utilisateurs• Ajouter un utilisateur• Rechercher un utilisateur<ul style="list-style-type: none">○ Modifier un utilisateur○ Supprimer un utilisateur➤ Se déconnecter
--	---

Tableau 3 : Spécification des tâches et identification des acteurs

I.1.3.3. Spécification des scénarios pour chaque tâche :

Les scénarios représentent les étapes successives pour la réalisation d'une tâche donnée. Le tableau suivant représente les différents scénarios à chaque acteur :

Tâches	Scénario
T01 :S'authentifier	S01 : Le système affiche la page d'authentification S02 :L'utilisateur saisit son Nom d'utilisateur son mot de passe puis clique sur le bouton « connexion » S03 : le système renvoie la page d'accueil de l'utilisateur si les informations saisies sont valides, sinon il renvoie un message d'erreur.
T02 :Gestion des biens	S01 :L'utilisateur clique sur le lien « gestion des bien » S02 : Le système affiche la liste déroulante des différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).
T03: Ajouter un bien	S01 : L'utilisateur clique sur le lien « Ajouter un bien » S02 : Le système renvoie le formulaire d'ajout du bien

CHAPITRE III : analyse et conception

	<p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout du bien si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T04 : rechercher un bien par code	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien par code »</p> <p>S02 : le système affiche un formulaire de recherche de bien.</p> <p>S03 : l'utilisateur saisit le code du bien puis clique sur « rechercher ».</p> <p>S02 : Le système renvoie la fiche du bien qui contient toutes les informations concernant le bien s'il existe dans la base de données sinon il renvoie un message « le bien n'existe pas dans la base de données »</p> <p>S03 : L'utilisateur peut cliquer sur le bouton « imprimer » pour imprimer la fiche du bien.</p>
T05 : Visualiser les biens d'un compte	<p>S01 : L'utilisateur clique sur le lien « Visualiser les biens par compte »</p> <p>S02 : le système affiche un formulaire de recherche du compte par code ou désignation</p> <p>S03 : l'utilisateur saisit le code ou la désignation du compte puis clique sur « valider ».</p> <p>S03 : Le système renvoie la liste des biens que le compte contient et affiche toutes les informations relatives à chaque bien.</p>
T06: Visualiser les	<p>S01 : L'utilisateur clique sur le lien « Visualiser les biens d'un</p>

CHAPITRE III : analyse et conception

biens d'un centre de frais	<p>centre de frais»</p> <p>S02 : le système affiche un formulaire de recherche du centre de frais par code ou par désignation.</p> <p>S03 : Le système renvoie la liste des biens que le centre de frais contient et affiche toutes les informations relatives à chaque bien.</p>
T07 : Modifier un bien	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p> <p>S02 Le système renvoie la fiche du bien.</p> <p>S03 : L'utilisateur clique sur le bouton « modifier » .</p> <p>S04 : Le système renvoie le formulaire du bien afin de le modifier.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S06 : le système effectue les modifications.</p>
T08: Effectuer Transférer un bien	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p> <p>S02 : Le système renvoie la fiche du bien, pour visualiser les informations le concernant.</p> <p>S03 : L'utilisateur clique sur le bouton « Transfert »</p> <p>S04 : Le système renvoie le formulaire du transfert.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « Valider».</p>
T09 : Effectuer la cession du bien	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p>

CHAPITRE III : analyse et conception

	<p>S02 : Le système renvoie la fiche du bien, pour visualiser les informations le concernant.</p> <p>S03 : L'utilisateur clique sur le bouton « cession»</p> <p>S04 : Le système renvoie le formulaire de la cession.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « Valider».</p>
T10 : Effectuer la sortie d'un bien	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p> <p>S02 : Le système renvoie la fiche du bien, pour visualiser les informations le concernant.</p> <p>S03 : L'utilisateur clique sur le bouton « Sortie»</p> <p>S04 : Le système renvoie le formulaire de Sortie.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « Valider».</p>
T11: Supprimer un bien	<p>S01 : L'utilisateur clique sur le lien « Supprimer un bien ».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.</p>
T12 :Gestion des centres de frais	<p>S01 :L'utilisateur clique sur le bien « gestion des centre de frais »</p> <p>S02 : Le système affiche la liste déroulante les différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).</p>

CHAPITRE III : analyse et conception

T13: Ajouter un centre de frais	<p>S01 : L'utilisateur clique sur le lien « Ajouter un centre de frais »</p> <p>S02 : Le système renvoie le formulaire d'ajout d'un centre de frais</p> <p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout du centre de frais si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T14 : rechercher un centre de frais par code	<p>S01 : L'utilisateur clique sur le lien « rechercher un centre de frais par code »</p> <p>S02 : le système affiche un formulaire de recherche du centre de frais.</p> <p>S03 : l'utilisateur saisit le code du centre de frais voulu puis clique sur « rechercher ».</p> <p>S04 : Le système renvoie les informations concernant le centre de frais si celui-ci existe dans la base de données sinon il renvoie un message « le centre de frais n'existe pas dans la base de données »</p>
T15 : Modifier un centre de frais	<p>S01 : L'utilisateur clique sur le bouton « modifier ».</p> <p>S02 : Le système renvoie le formulaire du centre de frais afin de le modifier.</p> <p>S03: l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04 : le système effectue les modifications.</p>

CHAPITRE III : analyse et conception

T16: Supprimer un centre de frais	<p>S01 : L'utilisateur clique sur le lien « Supprimer».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.</p>
T17 : Gestion des unités	<p>S01 :L'utilisateur clique sur le lien « gestion des unités »</p> <p>S02 : Le système affiche la liste déroulante des différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).</p>
T18: Ajouter une unité	<p>S01 : L'utilisateur clique sur le lien « Ajouter une unité»</p> <p>S02 : Le système renvoie le formulaire d'ajout d'une unité</p> <p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout de l'unité si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T19 : rechercher une unité par code	<p>S01 : L'utilisateur clique sur le lien « rechercher une unité par code»</p> <p>S02 : le système affiche un formulaire de recherche.</p> <p>S03 : l'utilisateur saisit le code de l'unité et clique sur « rechercher »</p> <p>S04 : Le système renvoie les informations concernant l'unité si celle-ci existe dans la base de données sinon il renvoie un message « l'unité n'existe pas dans la base de données »</p>
T20 : Modifier une unité	<p>S01 : L'utilisateur clique sur le bouton « modifier».</p>

CHAPITRE III : analyse et conception

	<p>S02 : Le système renvoie le formulaire de l'unité afin de modifier.</p> <p>S03 : l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04: le système effectue les modifications.</p>
T21: Supprimer une unité	<p>S01 : L'utilisateur clique sur le lien « Supprimer».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.</p>
T22 : Gestion des comptes	<p>S01 :L'utilisateur clique sur le lien « gestion des comptes »</p> <p>S02 : Le système affiche la liste déroulante les différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).</p>
T23: Ajouter un compte	<p>S01 : L'utilisateur clique sur le lien « Ajouter un compte»</p> <p>S02 : Le système renvoie le formulaire d'ajout de compte.</p> <p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout du bien si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T24 : rechercher un compte par code	<p>S01 : L'utilisateur clique sur le lien « rechercher un compte par code»</p> <p>S02 : le système affiche un formulaire recherche.</p> <p>S03 : l'utilisateur saisit le code du compte puis clique sur le</p>

CHAPITRE III : analyse et conception

	<p>bouton « rechercher »</p> <p>S04 : Le système renvoie les informations concernant le compte si celui-ci existe dans la base de données sinon il renvoie un message « le compten'existe pas dans la base de données ».</p>
T25 : Modifier un compte	<p>S01 : L'utilisateur clique sur le bouton « modifier»</p> <p>S02 : Le système renvoie le formulaire du compte afin de le modifier.</p> <p>S03 : l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04 : le système effectue les modifications.</p>
T26: Supprimer un compte	<p>S01 : L'utilisateur clique sur le lien « Supprimer».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.</p>

Tableau 3 : Scénarios des tâches de l'utilisateur

❖ **Acteur : « Administrateur »**

Tâches	Scénario
T01 :S'authentifier	<p>S01 : Le système affiche la page d'authentification</p> <p>S02 :L'utilisateur saisit son Nom d'utilisateur son mot de passe</p>

CHAPITRE III : analyse et conception

	<p>puis clique sur le bouton « connexion »</p> <p>S03 : le système renvoie la page d'accueil de l'utilisateur si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T02 : Gestion des biens	<p>S01 : L'utilisateur clique sur le lien « gestion des bien »</p> <p>S02 : Le système affiche la liste déroulante des différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).</p>
T03: Ajouter un bien	<p>S01 : L'utilisateur clique sur le lien « Ajouter un bien »</p> <p>S02 : Le système renvoie le formulaire d'ajout du bien</p> <p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout du bien si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T04 : rechercher un bien par code	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien par code »</p> <p>S02 : le système affiche un formulaire de recherche de bien.</p> <p>S03 : l'utilisateur saisit le code du bien puis clique sur « rechercher ».</p> <p>S02 : Le système renvoie la fiche du bien qui contient toutes les informations concernant le bien s'il existe dans la base de données sinon il renvoie un message « le bien n'existe pas dans la base de données »</p> <p>S03 : L'utilisateur peut cliquer sur le bouton « imprimer » pour imprimer la fiche du bien.</p>

CHAPITRE III : analyse et conception

T05 : Visualiser les biens d'un compte	<p>S01 : L'utilisateur clique sur le lien « Visualiser les biens par compte»</p> <p>S02 : le système affiche un formulaire de recherche du compte par code ou désignation</p> <p>S03 : l'utilisateur saisit le code ou la désignation du compte puis clique sur « valider ».</p> <p>S03 : Le système renvoie la liste des biens que le compte contient et affiche toutes les informations relatives à chaque bien.</p>
T06: Visualiser les biens d'un centre de frais	<p>S01 : L'utilisateur clique sur le lien « Visualiser les biens d'un centre de frais»</p> <p>S02 : le système affiche un formulaire de recherche du centre de frais par code ou par désignation.</p> <p>S03 : Le système renvoie la liste des biens que le centre de frais contient et affiche toutes les informations relatives à chaque bien.</p>
T07 : Modifier un bien	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p> <p>S02 Le système renvoie la fiche du bien.</p> <p>S03 : L'utilisateur clique sur le bouton « modifier» .</p> <p>S04 : Le système renvoie le formulaire du bien afin de le modifier.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S06 : le système effectue les modifications.</p>

CHAPITRE III : analyse et conception

T08 : Transférer un bien	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p> <p>S02 : Le système renvoie la fiche du bien, pour visualiser les informations le concernant.</p> <p>S03 : L'utilisateur clique sur le bouton « Transfert »</p> <p>S04 : Le système renvoie le formulaire du transfert.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « Valider».</p>
T09 : Effectuer la cession du bien	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p> <p>S02 : Le système renvoie la fiche du bien, pour visualiser les informations le concernant.</p> <p>S03 : L'utilisateur clique sur le bouton « cession»</p> <p>S04 : Le système renvoie le formulaire de la cession.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « Valider».</p>
T10 : Effectuer la sortie	<p>S01 : L'utilisateur clique sur le lien « rechercher un bien »</p> <p>S02 : Le système renvoie la fiche du bien, pour visualiser les informations le concernant.</p> <p>S03 : L'utilisateur clique sur le bouton « Sortie»</p> <p>S04 : Le système renvoie le formulaire de Sortie.</p> <p>S05 : l'utilisateur remplit le formulaire, puis clique sur le bouton « Valider».</p>

CHAPITRE III : analyse et conception

T11 : Supprimer un bien	<p>S01 : L'utilisateur clique sur le lien « Supprimer un bien ».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.</p>
T12 : Gestion des centres de frais	<p>S01 :L'utilisateur clique sur le bien « gestion des centre de frais »</p> <p>S02 : Le système affiche la liste déroulante les différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).</p>
T13: Ajouter un centre de frais	<p>S01 : L'utilisateur clique sur le lien « Ajouter un centre de frais »</p> <p>S02 : Le système renvoie le formulaire d'ajout d'un centre de frais</p> <p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout du centre de frais si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T14 : rechercher un centre de frais par code	<p>S01 : L'utilisateur clique sur le lien « rechercher un centre de frais par code»</p> <p>S02 : le système affiche un formulaire de recherche du centre de frais.</p> <p>S03 : l'utilisateur saisit le code du centre de frais voulu puis clique sur « rechercher ».</p>

CHAPITRE III : analyse et conception

	<p>S04 : Le système renvoie les informations concernant le centre de frais si celui-ci existe dans la base de données sinon il renvoie un message « le centre de frais n'existe pas dans la base de données »</p>
T15 : Modifier un centre de frais	<p>S01 : L'utilisateur clique sur le bouton « modifier ».</p> <p>S02 : Le système renvoie le formulaire du centre de frais afin de le modifier.</p> <p>S03: l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04 : le système effectue les modifications.</p>
T16 : Supprimer un centre de frais	<p>S01 : L'utilisateur clique sur le lien « Supprimer ».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.</p>
T17 : Gestion des unités	<p>S01 : L'utilisateur clique sur le lien « gestion des unités »</p> <p>S02 : Le système affiche la liste déroulante des différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).</p>
T18 : Ajouter une unité	<p>S01 : L'utilisateur clique sur le lien « Ajouter une unité »</p> <p>S02 : Le système renvoie le formulaire d'ajout d'une unité</p> <p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout de l'unité si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>

CHAPITRE III : analyse et conception

T19 : rechercher une unité par code	<p>S01 : L'utilisateur clique sur le lien « rechercher une unité par code »</p> <p>S02 : le système affiche un formulaire de recherche.</p> <p>S03 : l'utilisateur saisit le code de l'unité et clique sur « rechercher »</p> <p>S04 : Le système renvoie les informations concernant l'unité si celle-ci existe dans la base de données sinon il renvoie un message « l'unité n'existe pas dans la base de données »</p>
T20 : Modifier une unité	<p>S01 : L'utilisateur clique sur le bouton « modifier ».</p> <p>S02 : Le système renvoie le formulaire de l'unité afin de modifier.</p> <p>S03 : l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04: le système effectue les modifications.</p>
T21 : Supprimer une unité	<p>S01 : L'utilisateur clique sur le lien « Supprimer ».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.</p>
T22 : Gestion des comptes	<p>S01 : L'utilisateur clique sur le lien « gestion des comptes »</p> <p>S02 : Le système affiche la liste déroulante des différentes tâches que l'utilisateur peut effectuer (rechercher, ajouter ...).</p>
T23 : Ajouter un compte	<p>S01 : L'utilisateur clique sur le lien « Ajouter un compte »</p>

CHAPITRE III : analyse et conception

	<p>S02 : Le système renvoie le formulaire d'ajout de compte.</p> <p>S03 : L'utilisateur remplit le formulaire, puis clique sur le bouton « Valider »</p> <p>S04 : Le système effectue l'ajout du bien si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T24 : rechercher un compte par code	<p>S01 : L'utilisateur clique sur le lien « rechercher un compte par code »</p> <p>S02 : le système affiche un formulaire recherche.</p> <p>S03 : l'utilisateur saisit le code du compte puis clique sur le bouton « rechercher »</p> <p>S04 : Le système renvoie les informations concernant le compte si celui-ci existe dans la base de données sinon il renvoie un message « le compte n'existe pas dans la base de données ».</p>
T25 : Modifier un compte	<p>S01 : L'utilisateur clique sur le bouton « modifier »</p> <p>S02 : Le système renvoie le formulaire du compte afin de le modifier.</p> <p>S03 : l'utilisateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04 : le système effectue les modifications.</p>
T26 : Supprimer un compte	<p>S01 : L'utilisateur clique sur le lien « Supprimer ».</p> <p>S02 : Le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la</p>

CHAPITRE III : analyse et conception

	base de données, ou bien l'utilisateur clique sur le bouton « Non » pour ne pas continuer la procédure de suppression.
T27 : Gestion des utilisateurs	<p>S01 : l'administrateur clique sur le lien « gestion des utilisateurs ».</p> <p>S02 : le système affiche la liste déroulante des différentes tâches que l'administrateur peut effectuer (rechercher, ajout...).</p>
T28 : Ajouter un utilisateur	<p>S01 : l'administrateur clique sur le lien « ajouter un utilisateur ».</p> <p>S02 : le système renvoie un formulaire de l'ajout d'un utilisateur.</p> <p>S03 : d'administrateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04 : le système effectue l'ajout de l'utilisateur si les informations saisies sont valides, sinon il renvoie un message d'erreur.</p>
T29 : Rechercher un utilisateur par code	<p>S01 : l'administrateur clique sur le lien « rechercher un utilisateur ».</p> <p>S02 : le système renvoie un formulaire de recherche.</p> <p>S03 : l'administrateur saisit le code de l'utilisateur puis clique sur le bouton « rechercher ».</p> <p>S04 : le système renvoie les informations concernant l'utilisateur si celui-ci existe dans la base de données, sinon il renvoie un message « l'utilisateur n'existe pas dans la base de données ».</p>

CHAPITRE III : analyse et conception

T30 : modifier un utilisateur	<p>S01 : l'administrateur clique sur le bouton « modifier ».</p> <p>S02 : le système renvoie le formulaire de l'utilisateur afin de le modifier.</p> <p>S03 : l'administrateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04 : le système enregistre les modifications.</p>
T31 : supprimer un utilisateur	<p>S01 : l'administrateur clique sur le lien « supprimer ».</p> <p>S02 : le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'administrateur clique sur le bouton « NON » pour annuler la suppression.</p>
T32 : gestion des fournisseurs	<p>S01 : l'administrateur clique sur le lien « gestion des fournisseurs ».</p> <p>S02 : le système affiche la liste déroulante des différentes tâches que l'administrateur peut effectuer (rechercher, ajout ...)</p>
T33 : ajouter un fournisseur	<p>S01 : l'administrateur clique sur le lien « ajouter un fournisseur ».</p> <p>S02 : le système renvoie le formulaire d'ajout de fournisseur.</p> <p>S03 : l'administrateur remplit le formulaire, puis clique sur le bouton « valider ».</p> <p>S04 : le système effectue l'ajout du fournisseur si les informations sont valides, sinon il renvoie un message d'erreur.</p>
T34 : rechercher un fournisseur par	<p>S01 : l'administrateur clique sur le lien « rechercher un</p>

CHAPITRE III : analyse et conception

code	<p>fournisseur par code ».</p> <p>S02 : le système renvoie un formulaire de recherche.</p> <p>S03 : l'administrateur saisit le code du fournisseur puis clique sur « rechercher ».</p> <p>S04 : le système renvoie les informations concernant le fournisseur si celui-ci existe dans la base de données, sinon il renvoie un message « le fournisseur n'existe pas dans la base de données ».</p>
T35 : modifier un fournisseur	<p>S01 : l'administrateur clique sur le bouton « modifier »</p> <p>S02 : le système renvoie le formulaire du fournisseur pour le modifier.</p> <p>S03 : l'administrateur remplit le formulaire, puis clique sur « valider ».</p> <p>S04 : le système enregistre les modifications.</p>
T36 : supprimer un fournisseur	<p>S01 : l'administrateur clique sur le lien « supprimer ».</p> <p>S02 : le système renvoie un message de confirmation.</p> <p>S03 : l'utilisateur clique sur le bouton « OUI » pour confirmer la suppression, puis le système effectue la suppression de la base de données, ou bien l'administrateur clique sur le bouton « NON » pour annuler la suppression.</p>

Tableau 4 : Scénario des tâches de l'administrateur

I.1.4. Spécification des cas d'utilisation :

Nous utilisons les cas d'utilisation pour modéliser les différentes interactions entre les acteurs et l'application. Le but est de modéliser les actions que les acteurs du domaine souhaitent effectuer sur l'application, en résumé il permet de décrire ce que le système devra faire sans spécifier comment le faire. Nous allons définir les cas d'utilisation de notre application et leurs acteurs, ainsi que les diagrammes associés à chacun de ces derniers.

I.1.4.1. Description des cas d'utilisation :

Nous procéderons à la description des cas d'utilisation dans notre système :

- **Cas d'utilisation « s'authentifier, utilisateur »**

Cas d'utilisation : S'authentifier

Rôle : Utilisateur

Description

- 1- L'utilisateur saisit l'URL de l'application ;
- 2- Le système affiche la page d'accueil spécifiée ;
- 3- L'utilisateur saisit son« **EMAIL**» et « **Mot de passe** »puis clique sur le lien « **connexion** »
- 4- Le système affiche la page d'accueil.

Cas d'utilisation « Ajouter un bien»

Cas d'utilisation : ajouter un bien

Rôle : Administrateur

Description :

- 1- Le système affiche la page d'accueil.
- 2- L'administrateur clique sur le lien « Ajouter un bien »

- 3- Le système affiche le formulaire d'ajout d'un bien.
- 4- L'administrateur remplit le formulaire du bien, puis clique sur le bouton « valider ».
- 5- Le système enregistre le bien dans la base de données et renvoie un message de confirmation s'il n'y a pas eu d'anomalie, sinon il renvoie un message d'erreur.

- **Cas d'utilisation « rechercher un bien »**

Cas d'utilisation : rechercher bien

Rôle : utilisateur

Description

- 1- Le système affiche la page d'accueil.
- 2- L'utilisateur clique sur le lien « rechercher bien ».
- 3- Le système lui affiche un formulaire de recherche.
- 4- L'utilisateur saisit le code du bien voulu puis clique sur « rechercher ».
- 5- Le système affiche le résultat de la recherche.

I.1.4.2. Diagrammes de Cas d'utilisation

Partant de l'identification des acteurs ainsi que la description des besoins et l'identification des activités de chaque acteur, nous pouvons construire les diagrammes de cas d'utilisation suivants :



Figure III.12 : Diagramme de cas d'utilisation « Acteur : Utilisateur »

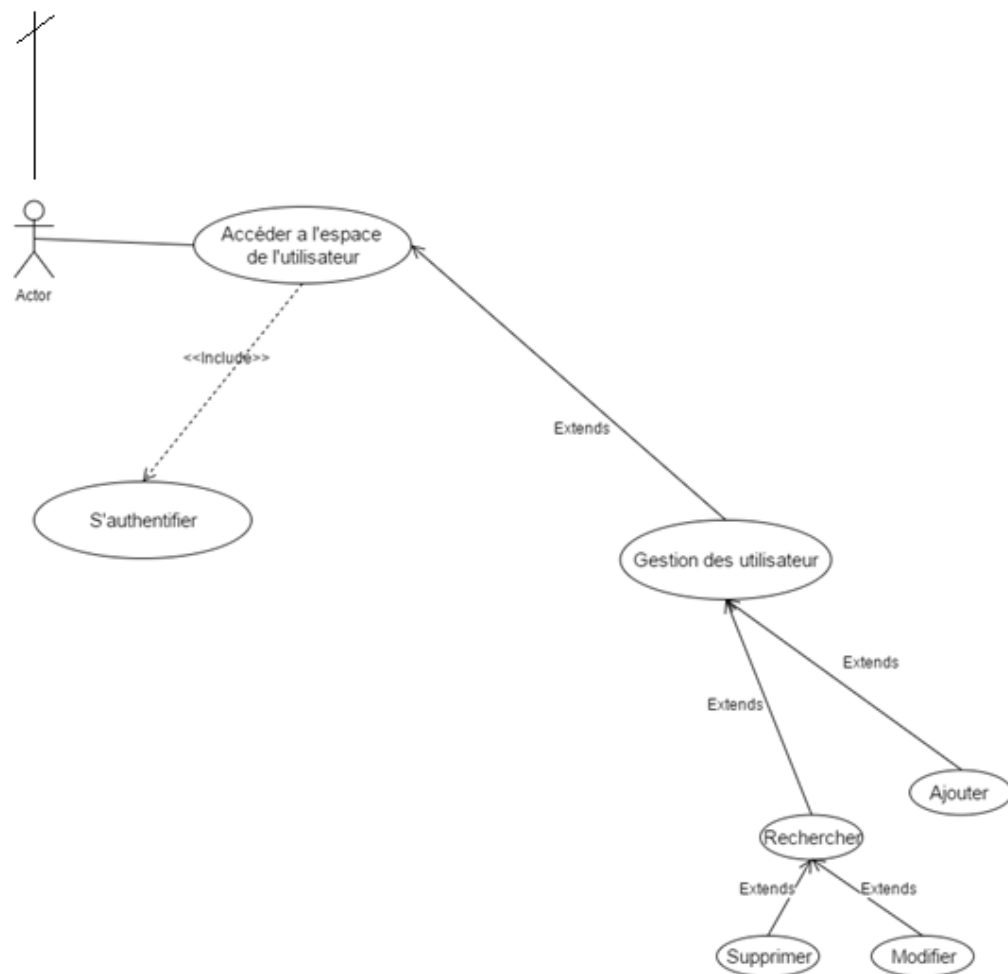


Figure III.13 : Diagramme de cas d'utilisation « Acteur : Administrateur »

II. Conception

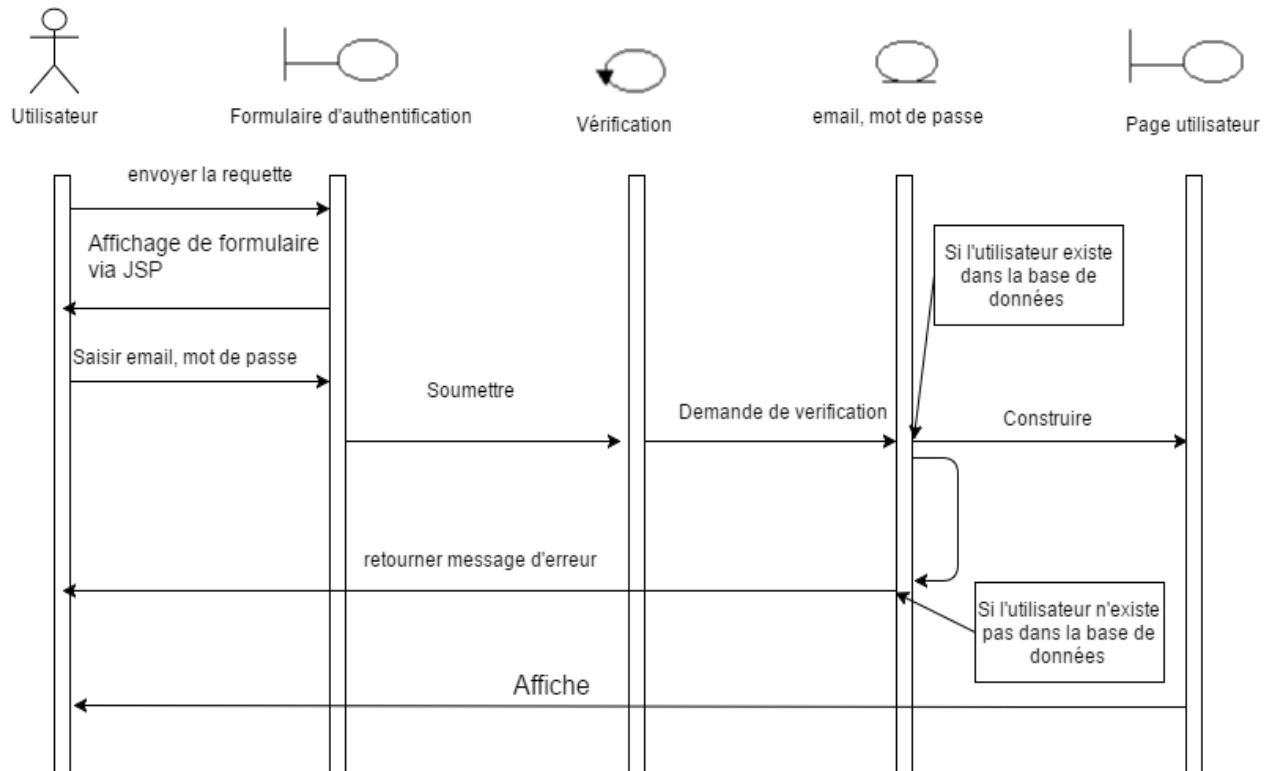
II.1. Les diagrammes de séquences

Les diagrammes de séquences permettent de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

CHAPITRE III : analyse et conception

Ce type de diagramme insiste sur l'aspect temporel, ils sont formés avec des classes traduisant la dynamique du système et qui seront utilisés dans l'activité de conception.

- **Cas d'utilisation «s'authentifier» :**



FigureIII.14 :Diagramme des séquences du cas d'utilisation«s'authentifier».

- **Cas d'utilisation «Ajouter un bien» :**

CHAPITRE III : analyse et conception

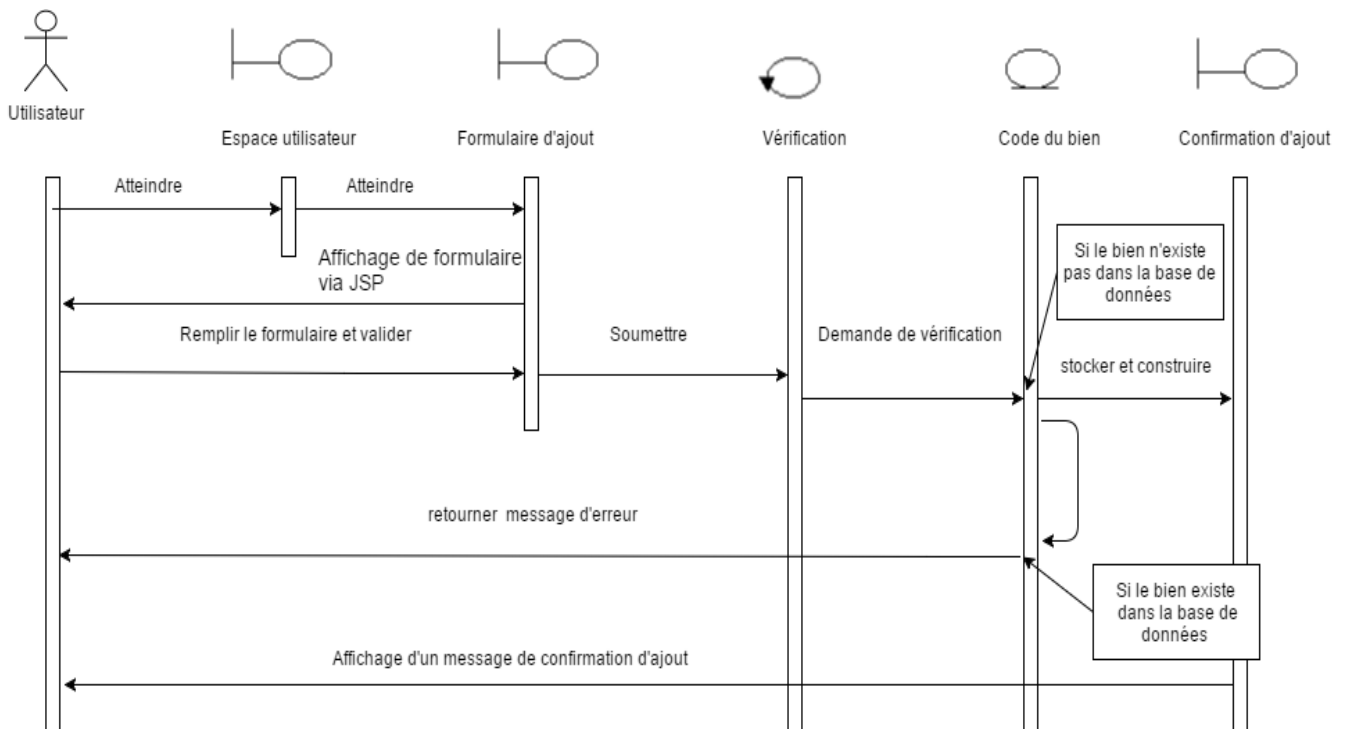


Figure III.15 :Diagramme des séquences du cas d'utilisation«Ajouter un bien»

- Cas d'utilisation «Rechercher un bien» :

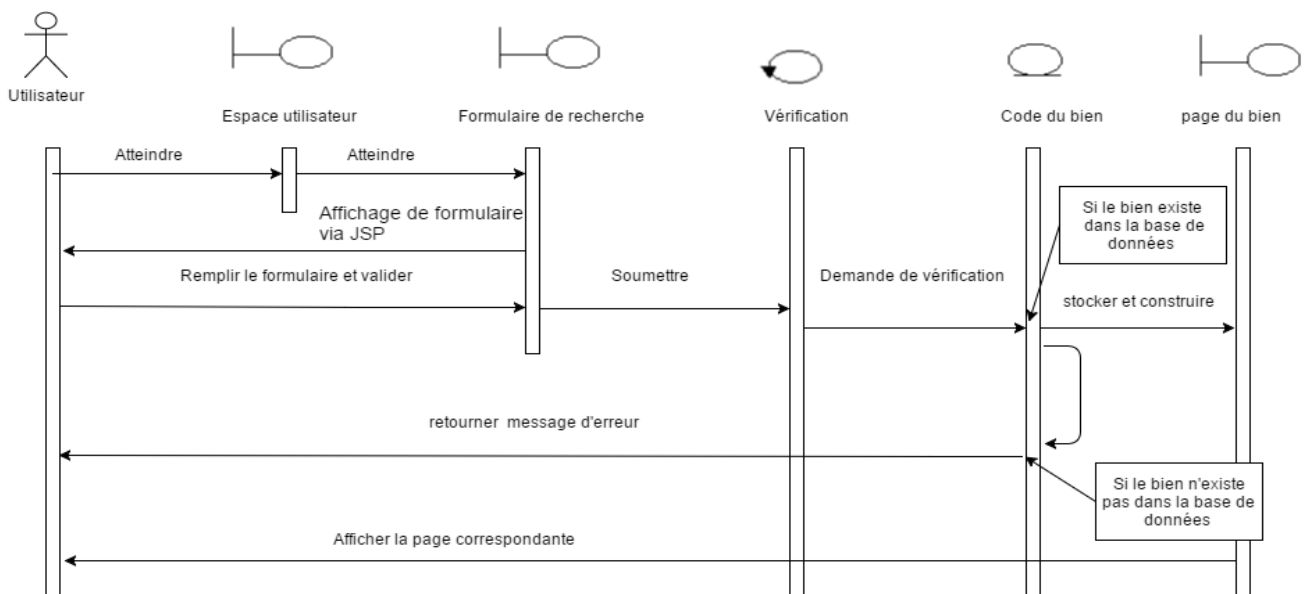


Figure III.16:Diagramme des séquences du cas d'utilisation«Rechercher un bien».

II.2. Diagramme de classe globale :

Les diagrammes de classes sont les plus courants dans la modélisation des systèmes orientés objets qui montrent la structure interne du système. Permet de définir un

CHAPITRE III : analyse et conception

ensemble de classes, d'interface ainsi que de leurs relations. Ces diagrammes sont utilisés pour modéliser la vue de conception statique. la figure suivante montre le diagramme de classes globale de notre application :

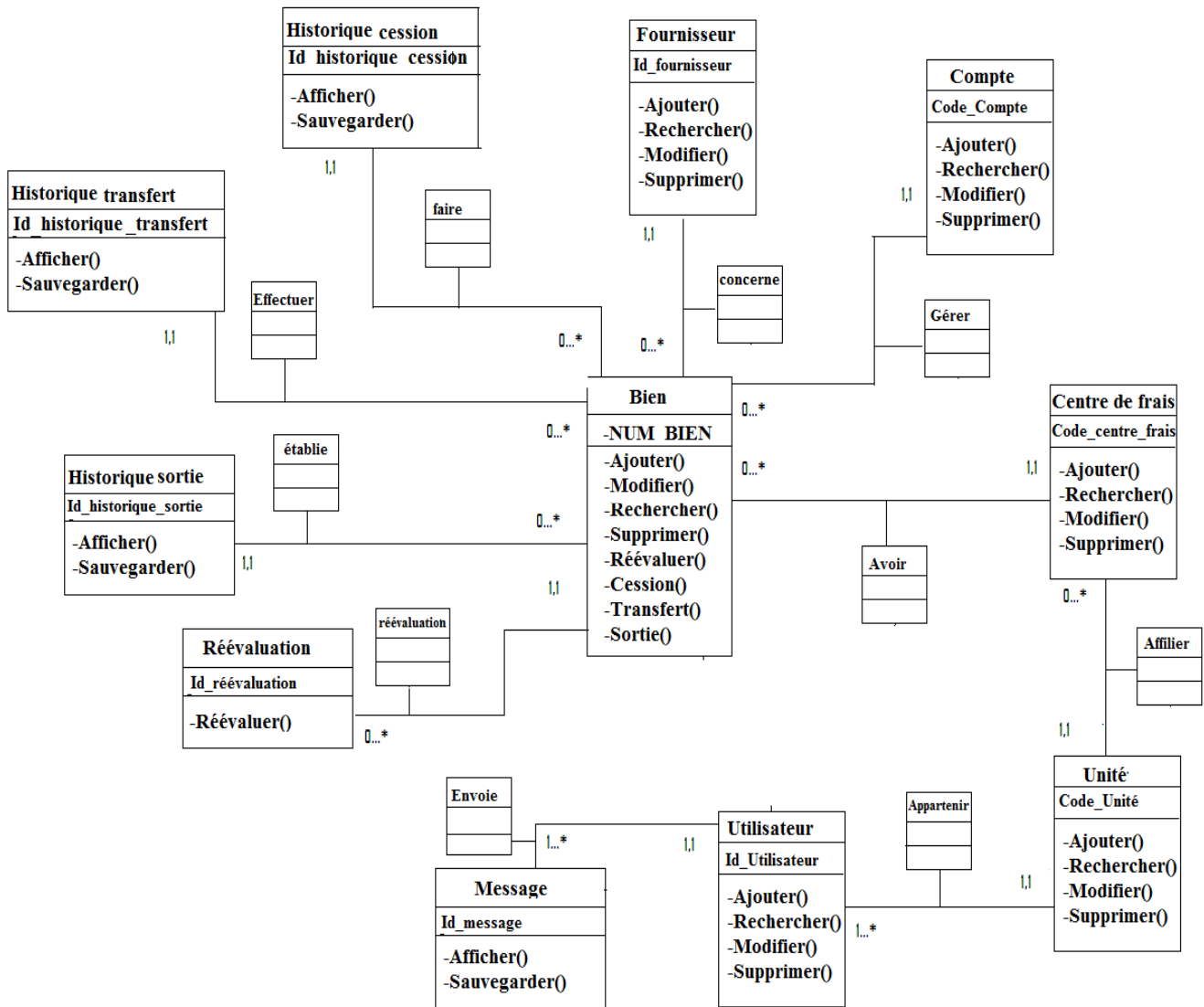


Figure III.17 : diagramme de classe globale.

III. Etude conceptuelle statique :

III.1. La Base de données :

Le modèle conceptuel des données (MCD) a pour but d'écrire de façon formelle les données qui seront utilisées par le système d'information. Il s'agit donc

d'une représentation des données, facilement compréhensible, permettant de décrire le système d'information à l'aide d'entités.

III.1.1. Les concepts de base :

- **Entité (Objet) :** une entité modélise un ensemble d'objets matériel qui présente un intérêt de gestion dans le cadre des systèmes d'informations.
- **Association (Relation):** c'est un lien entre deux ou plusieurs entités. Elle n'a pas d'existence propre.
- **Propriété :** Une donnée élémentaire que l'on perçoit sur une entité ou une association. Elle est généralement descriptive.
- **Identifiant :** Il permet de connaître de façon sûre et unique l'ensemble des propriétés qui participent à l'entité.
- **Cardinalités :** Elles expriment le nombre de fois ou l'occurrence d'une entité participe aux occurrences de la relation.

III.1.2. Les règles de gestion du MCD :

- Un bien peut appartenir à un et un seul centre de frais
- Un bien peut appartenir à une et une seule unité
- Un bien peut appartenir à un et un seul compte
- Un bien peut faire une à plusieurs cessions.
- Un bien peut faire un à plusieurs transferts.
- Un utilisateur peut faire partie d'une seule unité.
- Un utilisateur peut envoyer un message.
- Un utilisateur peut recevoir un message.
- Un bien peut avoir un et un seul fournisseur
- Un fournisseur concerne un ou plusieurs biens
- Une réévaluation peut s'effectuer pour un ou plusieurs biens
- L'historique des cessions peut être sauvegardé
- L'historique des transferts peut être sauvegardé

- L'historique des sorties peut être sauvegardé
- La mise en service d'un bien acquis se fait dès le lendemain de son acquisition.

III.1.3. Représentation du Modèle Conceptuel des Données :

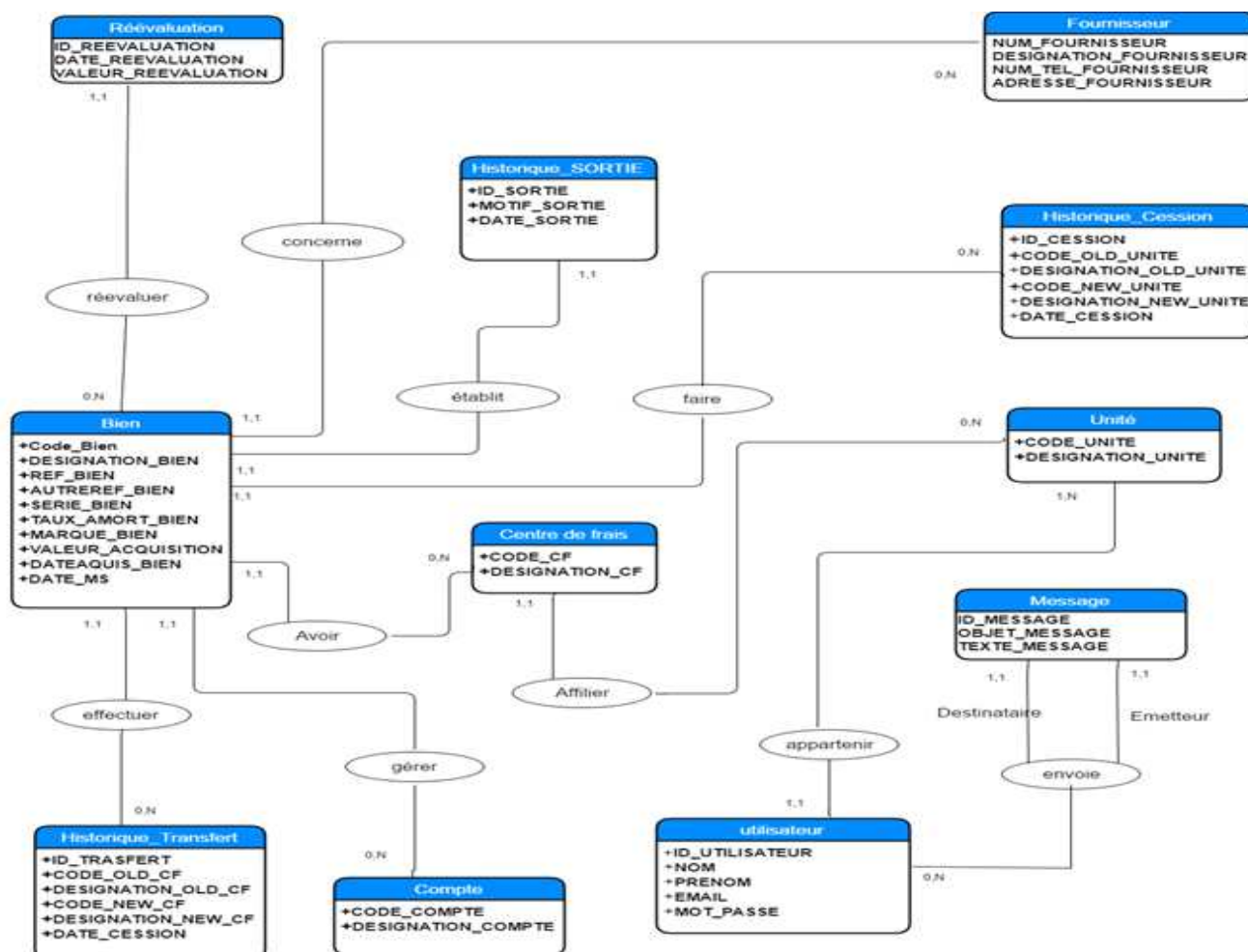


Figure III.18 : le modèle conceptuel de données de notre application

III.1.4. La liste des associations :

Association	Entité/Association	Cardinalités
Faire	Bien	1-1
	Historique_cession	0-N
rattacher	Bien	1-1
	Centre de frais	0-N

CHAPITRE III : analyse et conception

Gérer	Bien	1-1
	Compte	0-N
Effectuer	Bien	1-1
	Historique_transfert	0-N
Affilier	Centre de frais	1-1
	Unité	1-N
Appartenir	Unité	1-1
	Utilisateur	1-N
Réévaluer	Bien	1-1
	réévaluation	0-N
Envoie	Message	0-N
	utilisateur (émetteur)	1-1
	utilisateur (destinataire)	1-1
Etablit	Bien	1-1
	Historique_sortie	0-N

Tableau 4 :liste des associations

IV. Conception de la base de données :

Le modèle logique des données relationnel se situe entre le modèle conceptuel des données MCD et le modèle physique de données (MPD), il représente l'univers des données décrit dans le MCD en tenant compte du type de base de données choisi relationnel ; réseau ou bien hiérarchique. autrement dit le MLD traduit le MCD en un formalisme machinale c'est-à-dire compréhensible par la machine.

IV.1. Représentation du modèle logique des données

Bien (code bien, designation_bien, ref_bien, autreref_bien, serie_bien, taux_amort_bien, marque_bien, valeur_aquisition, dateaquis_bien, date_ms,

CHAPITRE III : analyse et conception

FK_code_compte*, FK_code_cf*, FK_num_fournisseur*, FK_Id_sortie*,
FK_Id_cession*, FK_Id_transfert*)

Centre de frais(code_cf, designation_cf, FK_code_unité*)

Compte(code_compte, designation_compte)

Unité (code unité, designation_unité)

Utilisateur (Id_user, nom, prenom, e-mail, mot_passe, fk_Id_unité*)

Historique_cession (Id cession, code_old_unité, designation_old_unité,
code_new_unité, designation_new_unité, date_cession)

Historique_transfert(Id transfert, code_old_cf, designation_old_cf, code_new_cf,
designation_new_cf, date_transfert)

Historique_sortie(Id sortie, motif_sortie, date_sortie, code_bien*)

Message(Id message, objet_message, texte_message,
FK_Id_user_emmeteur*, FK_Id_message_recepteur*)

Fournisseur(Num fournisseur, Designation_fournisseur, num_tel_fournisseur, adresse
_fournisseur)

Réévaluation(Id reevaluation, date_reevaluation, valeur_réévaluation,
Fk_code_bien)

IV.2. Les tables :

Bien

Nom de champ	Type	NULL	Index
Code_bien	Varchar(20)	NON	Clé primaire
Designation_bien	Varchar(20)	NON	
Ref_bien	Varchar(20)	NON	

CHAPITRE III : analyse et conception

AutreRef_bien	Varchar(20)	NON	
Serie_bien	Varchar(20)	NON	
Taux_amort_bien	Int(20)	NON	
Marque_bien	Varchar(20)	NON	
Valeur_acquisition	Varchar(20)	NON	
DateAcqui_bien	Date	NON	
Date_ms	Date	NON	
Fk_code_compte	Varchar(20)	NON	clé étrangère
Fk_code_cf	Varchar(20)	NON	clé étrangère
Fk_num_fournisseur	Varchar(20)	NON	clé étrangère
Fk_Id_sortie	Varchar(20)	NON	clé étrangère
Fk_Id_cession	Varchar(20)	NON	clé étrangère
Fk_Id_transfert	Varchar(20)	NON	clé étrangère

Tableau 5 : table bien

Centre de frais

Nom de champ	Type	NULL	Index
Code_cf	Varchar(20)	NON	Clé primaire
Designation_cf	Varchar(20)	NON	
Fk_code_unité	Varchar(20)	NON	clé étrangère

Tableau 6 : table centre de frais

Unité

Nom de champ	Type	NULL	Index
Code_unité	Varchar(20)	NON	Clé primaire
Designation_unité	Varchar(20)	NON	

Tableau 7 : table Unité

Utilisateur

Nom de champ	Type	NULL	Index
--------------	------	------	-------

CHAPITRE III : analyse et conception

Id_user	Int(20)	NON	Clé primaire
Nom	Varchar(20)	NON	
Prenom	Varchar(20)	NON	
E-mail	Varchar(20)	NON	
Mot_passe	Varchar(20)	NON	
Fk_id_unité	Varchar(20)	NON	Clé étrangère

Tableau 8 : table utilisateur

Historique cession :

Nom de champ	Type	NULL	Index
Id_cession	Int(20)	NON	Clé primaire
Code_old_unité	Varchar(20)	NON	
Désignation_old_unité	Varchar(20)	NON	
Code_new_unité	Varchar(20)	NON	
Date_cession	Varchar(20)	NON	

Tableau 9 : table Historique cession

Historique_transfert

Nom de champ	Type	NULL	Index
Id_transfert	Int(20)	NON	Clé primaire
Code_old_cf	Varchar(20)	NON	
Désignation_old_cf	Varchar(20)	NON	
Code_new_cf	Varchar(20)	NON	
Désignation_new_cf	Varchar(20)	NON	
Date_transfert	Varchar(20)	NON	

Tableau 10: table Historique transfert

Historique_sortie

Nom de champ	Type	NULL	Index
Id_sortie	Int(20)	NON	Clé primaire

CHAPITRE III : analyse et conception

Motif_sortie	Varchar(20)	NON	
Date_sortie	Varchar(20)	NON	
FK_Code_bien	Varchar(20)	NON	Clé étrangère

Tableau 10: table Historique sortie

Message

Nom de champ	Type	NULL	Index
Id_message	Int(20)	NON	Clé primaire
Objet_Message	Varchar(20)	NON	
Texte_message	Varchar(20)	NON	
Fk_id_user_emmeteur	Varchar(20)	NON	Clé étrangère
Fk_id_message_recepteur	Varchar(20)	NON	Clé étrangère

Tableau 11 : Tablemessage

Fournisseur

Nom de champ	Type	NULL	Index
Num_fournisseur	Int(20)	NON	Clé primaire
Designation_fournisseur	Varchar(20)	NON	
Num_Tel_Fournisseur	Varchar(20)	NON	
Adresse_fournisseur	Varchar(20)	NON	

Tableau 12 : Tablefournisseur

Réévaluation

Nom de champ	Type	NULL	Index
Id_reevaluation	Int(20)	NON	Clé primaire
Date_réévaluation	Varchar(20)	NON	
Valeur_réévaluation	Varchar(20)	NON	
Fk_code_bien	Varchar(20)	NON	Clé étrangère

Tableau 13 : Table réévaluation

CHAPITRE III : analyse et conception

Codification

Code	Désignation	Type	Taille
Code_bien	code du bien	AN	20
Désignation_bien	désignation du bien	N	20
Ref_bien	référence du bien	AN	20
AutreRef_bien	autre référence pour le bien	AN	20
Serie_bien	serie du bien	N	20
Taux_amort_bien	taux d'amortissement du bien	N	20
Marque_bien	marque du bien	AN	20
Valeur_acquisition	valeur d'acquisition du bien	N	20
DateAcquis_bien	date d'acquisition du bien	D	10
Date_ms	date de mise en service	D	10
Code_cf	code du centre de frais	AN	20
Désignation_cf	Désignation centre de frais	A	20
Code_compte	code du compte	AN	20
Désignation_compte	désignation du compte	A	20
Code_unité	code de l'unité	AN	20
Désignation_unité	Désignation de l'unité	A	20
Id_user	identifiant utilisateur	N	20
Nom	Nom de l'utilisateur	A	20
Prénom	Prénom de l'utilisateur	A	20
E-mail	e-mail de l'utilisateur	AN	20
Mot_passe	mot de passe	AN	20

CHAPITRE III : analyse et conception

	utilisateur		
Id_cession	identifiant cession	N	20
Code_old_unité	code de l'ancienne unité	AN	20
Désignation_old_unité	désignation de l'ancienne unité	A	20
Code_new_unité	Code de la nouvelle unité	AN	20
Désignation_new_unité	désignation de la nouvelle unité	A	20
Date_cession	date de cession	D	10
Id_transfert	Identifiant transfert	N	20
Code_old_cf	code de l'ancien centre de frais	AN	20
Désignation_old_cf	désignation de l'ancien centre de frais	A	20
Code_new_cf	code du nouveau centre de frais	AN	20
Désignation_new_cf	désignation du nouveau centre de frais	A	20
Date_transfert	date de transfert du bien	D	20
Id_message	Identifiant message	N	20
Objet_message	Objet du message	AN	30
Texte_message	Texte du message	AN	150
Num_fournisseur	Numéro du fournisseur	N	20
Designation_fournisseur	Designation du fournisseur	AN	20
Num_tel_fournisseur	Numéro de téléphone du fournisseur	N	20

CHAPITRE III : analyse et conception

Adresse_fournisseur	Adresse du fournisseur	AN	50
Id_reevaluation	Identifiant de la réévaluation	N	20
Date_reevaluation	Date de la réévaluation	D	10
Valeur_reevaluation	Valeur de la réévaluation	N	20
Id_sortie	Idantifiant de la sortie	N	20
Motif_sortie	Le motif de la sortie	A	20
Date_sortie	La date de la sortie	D	10

Tableau 14 : tableau de la Codification

V. Algorithme de calcul :

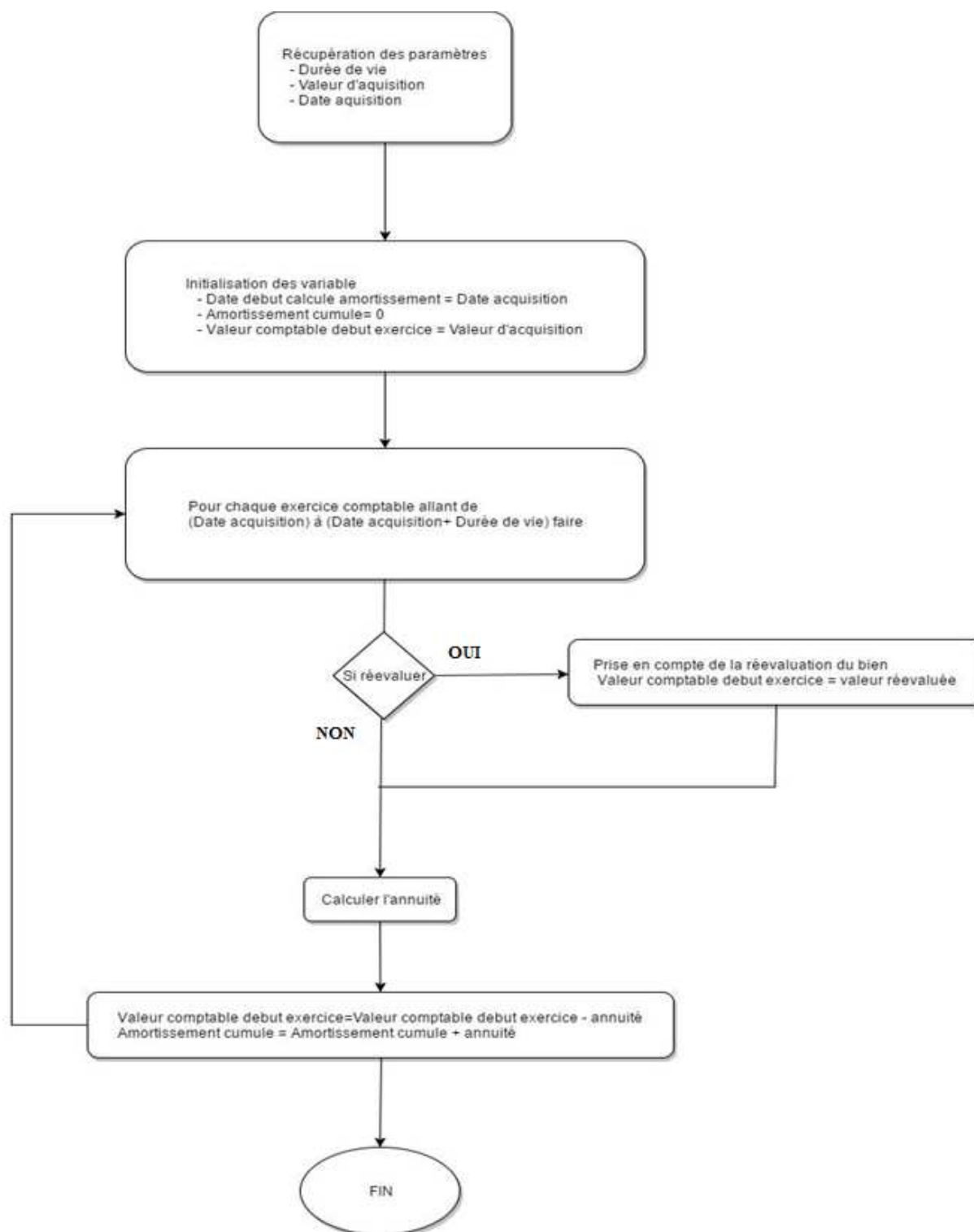


Figure III.19 : algorithme de calcul.

Conclusion :

Dans ce chapitre, nous avons proposé une démarche de modélisation pour développer notre application, en basant sur la méthode **UML**, en commençant par la spécification des besoins et les divers cas d'utilisation, puis la conception des diagrammes de collaboration et de séquence en phase d'analyse, et en phase de conception nous avons élaboré les diagrammes de classe, et enfin, nous avons défini les tables de notre base de données. Reste à définir les outils et les langages de programmation qui vont nous aider à mettre en œuvre notre application, ce qui sera l'objet du chapitre suivant, qui est la réalisation.

Chapitre VI

Réalisation



Introduction :

Dans ce chapitre nous allons présenter l'environnement de développement de l'application ainsi que les différents outils utilisés pour sa conception et réalisation. Ce dernier chapitre sera clos par la présentation de quelques interfaces de l'application qui exposeront différentes fonctionnalités.

I.Environnement et outils de développement et d'implémentation :

I.1.Google Drive :

Google Drive est un service de stockage et de partage de fichiers *dans le cloud* lancé par la société Google. Google Drive, qui regroupe Google Docs, Sheets et Slides, est une suite bureautique permettant de modifier des documents, des feuilles de calcul, des présentations, des dessins, des formulaires, etc. Les utilisateurs peuvent rechercher les fichiers partagés publiquement sur Google Drive par le biais de moteurs de recherche Web.

Lancé le 24 avril 2012, Google Drive compte 240 millions d'utilisateurs actifs en octobre 2014.

Google Drive remplace Google Docs une fois actif. Les documents existant déjà sur Google Docs sont automatiquement transférés sur Google Drive. Il sert à synchroniser, partager et modifier les données entre plusieurs ordinateurs et/ou utilisateurs.

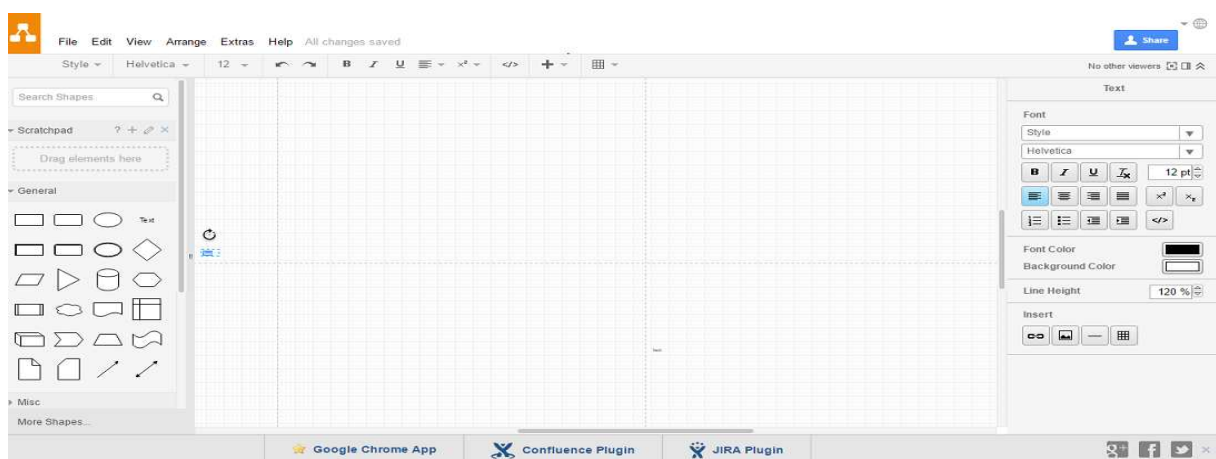


Figure IV.18 : Interface Google Drive



I.2. Wampserver :

WampServer (anciennement **WAMP5**) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un *trayicon* (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.[16]



Figure IV.19 : Panneau de gestion du serveur



I.3. Eclipse :

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

L'environnement de développement open source Eclipse évolue pour répondre aux besoins des développeurs, et dans notre travail nous avons opté pour la version **eclipse 4.5** ou **eclipse mars**. [17]

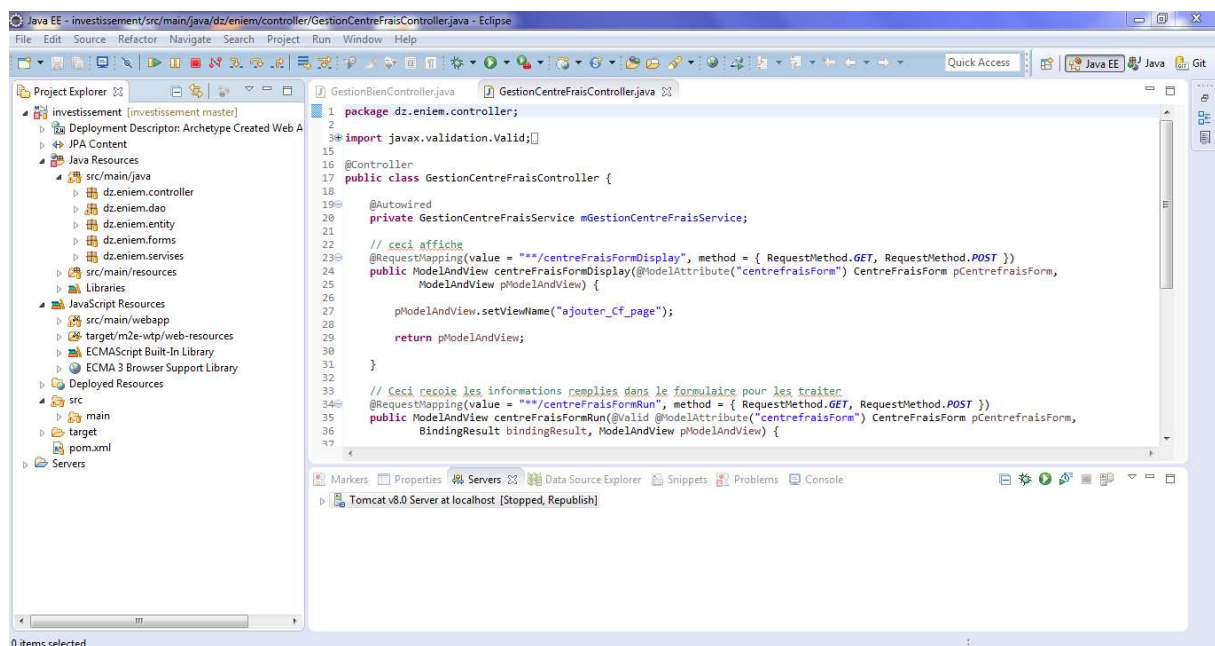


Figure IV.20 : Interface Eclipse



I.4. ApacheTomcat :

Apache Tomcat est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'*Apache Software Foundation*. Il implémente les spécifications des servlets et des JSP du Java CommunityProcess, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP. Tomcat est un serveur HTTP à part entière. De plus, il gère les servlets et les JSP (par un compilateur Jasper compilant les pages JSP pour en faire des servlets).

Tomcat a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant.

I.5. ApacheMaven :

ApacheMaven est un outil pour la gestion et l'automatisation de production des projets logiciels Java en général et Java EE en particulier. L'objectif recherché est produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.

il permet notamment de :

- créer une arborescence standard du code et de ses ressources
- télécharger, mettre à jour et configurer les bibliothèques nécessaires au projet
- compiler le projet, effectuer des tests unitaires sur le code et packager le résultat

Eclipse permet de générer des packages mais ces derniers ne respectent pas la structure de Maven et ne sont pas non plus installés dans le référentiel, cela implique que sans l'utilisation d'un outil externe il faut installer manuellement tous les packages générés.

Un plugin Maven pour Eclipse est disponible permettant à Eclipse d'utiliser Maven en arrière-plan et donc d'utiliser Maven et Eclipse conjointement.

Ceci permet d'avoir des projets Maven générés et stockés dans le référentiel, il permet notamment de dépendre de projets qui sont dans le référentiel et donc on n'a pas besoin de les importer sous Eclipse comme c'est le cas dans les outils existants, le classpath est généré par Maven.

I.6. Bootstrap :

Bootstrap est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plateforme de gestion de développement GitHub.

La modularité de Bootstrap consiste en une série de feuilles de styles LESS qui implémentent différents composants du *toolkit*. Une feuille de style principale (*bootstrap.less*) englobe les feuilles de style des composants. Les développeurs peuvent alors sélectionner les composants dont ils ont besoin en modifiant cette feuille principale. L'utilisation de LESS permet la manipulation de variables, de fonctions, d'opérateurs,...

Depuis la version 2, Bootstrap dispose d'une option supplémentaire de « personnalisation ».

Le système de grille et le style adaptatif sont standardisés sur une grille de 940 pixels de large, que les développeurs peuvent adapter. Ces définitions sont déclinées en quatre variations qui peuvent être utilisées dans différents formats et supports : téléphones (portrait et paysage), tablette et PC (haute et basse résolution). Ces déclinaisons adaptent automatiquement l'affichage de la page.



I.7. My SQL

MySQL est un véritable serveur de base de données SQL multiutilisateur. Le langage SQL (StructuredQueryLanguage) étant le langage le plus populaire de base de données dans le monde. Il en prend la syntaxe mais n'en conserve pas toute la puissance puisque de nombreuses fonctionnalités de SQL n'apparaissent pas dans MYSQL (sélection imbriquées, clés étrangère...). SQL est un langage standardisé qui rend facile le stockage, la mise à jour et l'accès à l'information. Il est suffisamment rapide et flexible pour gérer des historiques et des images.

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête.

I.8. L'interface graphique PhpMyAdmin

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, qu'ils soient gratuits ou payants, le proposent ce qui permet à l'utilisateur de ne pas avoir à l'installer.

Intégralement développé en PHP, phpMyadmin permet une administration aisée des bases de données MySQL. C'est un outil libre et gratuit développé par la communauté des programmeurs libres. Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances dans le domaine des bases de données, de nombreuses requêtes comme la création de tables de données, les insertions, les mises à jour, les suppressions, les modifications de structure de la base de données. Ce système est très pratique pour sauvegarder une base de données sous forme de fichier. Sql et ainsi transférer facilement ces données. De plus celui-ci accepte la formulation de requêtes SQL directement en langage SQL, cela permet de tester ses requêtes par exemple lors de la création d'un site et ainsi de gagner un temps précieux.

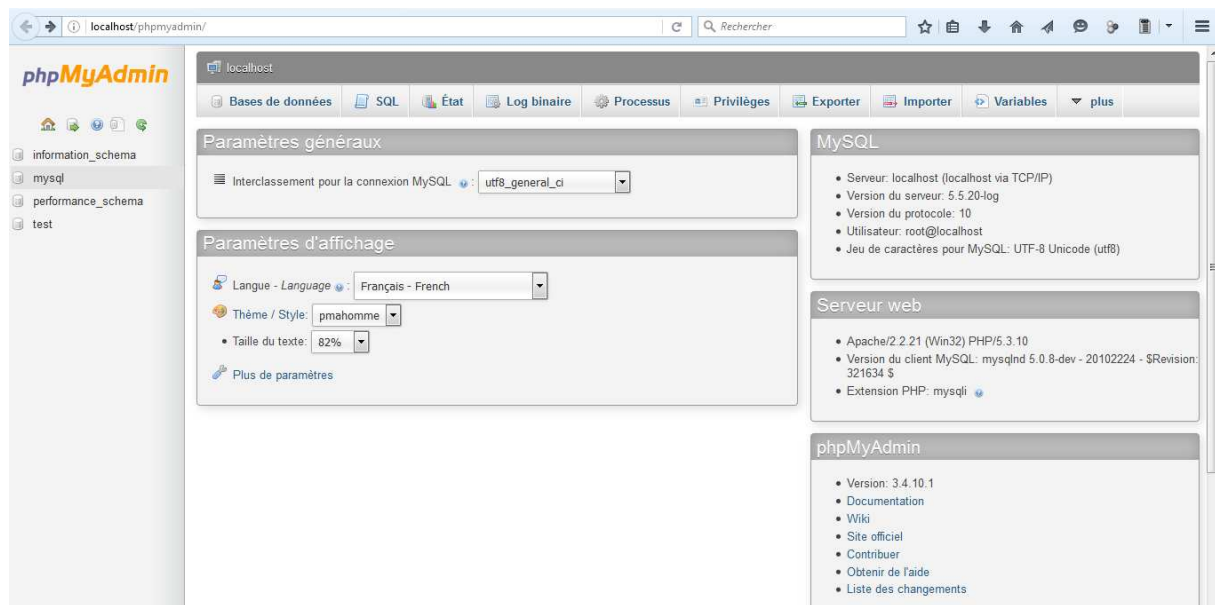


Figure IV.21 : Interface d'accueil phpMyAdmin

I.9. Git :

Git est un logiciel de gestion de versions décentralisé parmi les plus populaire. Il a été conçu pour gérer les projets de très grandes envergures. Il permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet notamment de retrouver les différentes versions d'un lot de fichiers connexes

C'est un logiciel libre créé par Linus Torvalds qui a souhaité remplacer l'outil propriétaire qui était utilisé pour le développement du noyau Linux.

Un logiciel de gestion de versions agit sur une arborescence de fichiers afin de conserver toutes les versions des fichiers, ainsi que les différences entre les fichiers.

Ce système permet par exemple de mutualiser un développement. Un groupe de développeurs autour d'un même développement se servira de l'outil pour stocker toute évolution du code source.

Le système gère les mises à jour des sources pour chaque développeur, conserve une trace de chaque changement. Ceux-ci sont, en bonne utilisation, chaque fois

accompagnés d'un commentaire.

Le système travaille par fusion de copies locale et distante, et non par écrasement de la version distante par la version locale. Ainsi, deux développeurs travaillant de concert sur une même source, les changements du premier à soumettre son travail ne seront pas perdus lorsque le second, qui a donc travaillé sur une version non encore modifiée par le premier, renvoie ses modifications.

Généralement, chaque nouvelle version d'un fichier est appelée *révision* et son numéro de version est incrémentée de 1 par rapport à la précédente.

I.10. Les langages de programmation :

Pour concrétiser les différentes parties de l'application, nous avons utilisés plusieurs langages :

- Le langage HTML.
- Le langage de requêtes MySQL.
- Le langage de script CSS.
- Le langage de script JavaScript.

II. Présentation de quelques interfaces de l'application :

II.1. Interface de connexion :

L'interface de connexion est celle qui permet à l'utilisateur de s'authentifier afin d'accéder à la page d'accueil de l'application.

CHAPITRE IV : réalisation

ENIEM - Gestion des investissements

Nom

Mot de passe

Connexion

Figure IV.22 : interface de connexion

II.2. Interface d'accueil :

L'interface d'accueil est celle qui permet à l'utilisateur d'accomplir les différentes tâches prévus par l'application.

ENIEM - Gestion des investissements

Gestion des biens <

Gestion des centres de frais <

Gestion des comptes <

Gestion des unités <

Gestion des utilisateurs <

Gestion des fournisseurs <

Bienvenue sur la page d'accueil

Infos Utilisateur

Nom: RAHMANI

Prénom: SARRA

Unité: FROID

Admin: Oui

Se déconnecter

Figure IV.23 : interface d'accueil

II.3. Interface ajouter un bien :

L'interface d'ajout d'un bien est celle qui permet à l'utilisateur de saisir les informations qui concernent un bien afin de le sauvegarder dans la base de données.

The screenshot shows a web browser window with the URL `localhost:8080/investissement/bienFormDisplay`. The application title is "ENIEM - Gestion des investissements". On the left, a sidebar menu lists various management options: "Ajouter un bien", "Rechercher/Modifier un bien", "Visualiser les biens d'un compte", "Visualiser les biens d'un centre de frais", "Gestion des centres de frais", "Gestion des comptes", "Gestion des unités", "Gestion des utilisateurs", and "Gestion des fournisseurs". The main content area is titled "Ajouter un bien" and contains a form with the following fields: "Code de bien", "Désignation du bien", "références du bien", "Autres références", "Série du bien", "Taux d'amortissement du bien", "La durée de vie du bien", "Marque du bien", "valeur d'acquisition", "Date d'acquisition" (with a date picker), "Date de mise en service" (with a date picker), "Liste des centres de frais" (a dropdown menu), and "Liste des comptes" (a dropdown menu). A "Validez" button is at the bottom of the form. On the right, a box labeled "Infos Utilisateur" displays the current user's details: Nom: RAHMANI, Prénom: SARRA, Utilis: FROID, Admin: Oui, and a "Se déconnecter" link.

Figure IV.24 : Interface ajouter un bien

II.4. Interface recherche un bien :

Cette interface permet à l'utilisateur de rechercher un bien grâce au code du bien.

CHAPITRE IV : réalisation

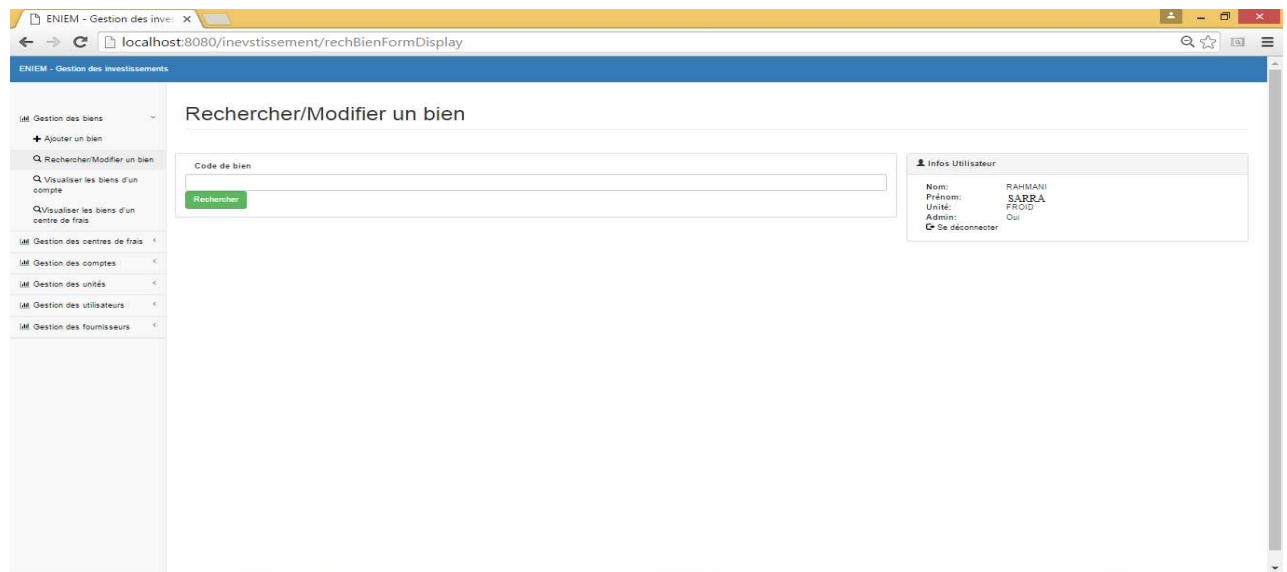


Figure IV.25 : Interface rechercher un bien

II.5. Interface de l’affichage du bien :

Sur cette interface l'utilisateur peut voir toutes les informations concernant le bien rechercher.

II.5.1 Interface l’affichage du bien (Partie 1) :

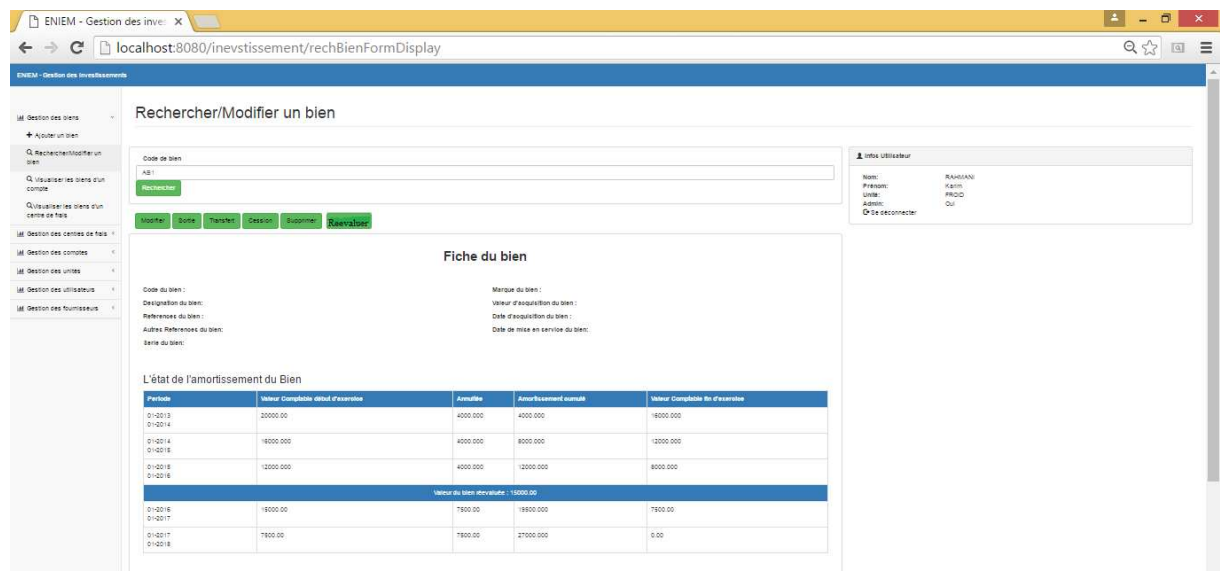


Figure IV.26 : Interface de l’affichage du bien (partie1)

II.5.2. Interface l'affichage du bien (partie2) :

Transferts du Bien				
Code centre de frais d'origine	Désignation centre de frais d'origine	Code centre de frais destination	Désignation centre de frais destination	Date de transfert
2	good	1	sestui	22-09-2016
1	sestui	2	good	22-09-2016
2	good	1	sestui	22-09-2016
1	sestui	2	good	22-09-2016
2	good	1	sestui	21-09-2016

Cessions du Bien				
Code unité d'origine	Désignation unité d'origine	Code unité destination	Désignation unité destination	Date de cession
2	CUISSON	1	FROID	22-09-2016
1	FROID	2	CUISSON	22-09-2016
2	CUISSON	1	FROID	22-09-2016
1	FROID	2	CUISSON	22-09-2016
1	FROID	2	CUISSON	21-09-2016

Figure IV.26 : Interface l'affichage du bien (partie2)

II.6. Interface de visualisation des biens d'un compte :

L'interface de visualisation des biens permet à l'utilisateur de visualiser tous les biens d'un compte.

ENIEM - Gestion des investissements

Visualiser les biens d'un compte

Code du compte

aezsrttdtgdg

Rechercher

Infos Utilisateur

Nom: RAHMANI

Prénom: Karim

Unité: FROID

Admin: Oui

Se déconnecter

Le compte ne possède aucun bien

- Gestion des biens
- + Ajouter un bien
- Rechercher/Modifier un bien
- Visualiser les biens d'un compte
- Visualiser les biens d'un centre de frais
- Gestion des centres de frais
- Gestion des comptes
- Gestion des unités
- Gestion des utilisateurs
- Gestion des fournisseurs

Figure IV.27 : interface de visualisation des biens d'un compte

Conclusion

Ce dernier chapitre a été consacré à la présentation de l'étape implémentation, dans laquelle nous avons présenté les outils logiciels qui nous ont permis de faire cette application à savoir l'environnement de développement, les langages de programmation, puis nous avons décrit notre application en présentant quelques interfaces avec leurs explications.

Conclusion générale

Conclusion générale :

Le stage effectué, nous a introduit dans le monde de l'entreprise et à travers ce projet de fin d'étude, nous avons pu mettre en pratique nos connaissances théoriques concernant les systèmes d'informations, nous avons également appris ce qu'est la comptabilité des immobilisations dans une entreprise.

La réalisation de l'application nous a permis de nous familiariser avec l'environnement de développement orienté web à savoir Eclipse et MySql pour les bases de données, elle nous a permis aussi d'appréhender la technologie JEE, et enfin maîtriser les outils et les méthodes de conception UML.

Nous espérons que notre travail répond aux besoins exprimés par le comptable analytique et investissement.

Bibliographie :

Titre : Introduction au système d'information

Auteur : Dalila Taouri

Edition : Page bleu

Titre : Développement JEE 5 avec Eclipse Europa

Auteur : Karim Djaafar

Edition : Eyrolles

Webographie :

[1]<https://blog.axopen.com/2015/10/architecture-application-web-2015/>

[2]<https://blog.axopen.com/2015/10/architecture-application-web-2015/>

[3]<https://www.mistra.fr/tutoriel-jee-introduction/tutoriel-jee-architecture-web.html>

[4]<https://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.html>

[5]<https://openclassrooms.com/courses/creez-votre-application-web-avec-java-ee>

[6]<https://www.clicours.com/telecharger-cours-introduction-a-la-plateforme-j2ee/>

[7]<https://www.clicours.com/telecharger-cours-introduction-a-la-plateforme-j2ee/>

[8]<https://www.eyrolles.com/chapitres/9782212120615/Chap10Djaafar.pdf>

[9]https://www.jmdoudoux.fr/java/dej/chap-spring_core.html

[10]https://www.jmdoudoux.fr/java/dej/chap-spring_core.html

[11]www.eniem.com.dz

[12]www.eniem.com.dz

[13]www.eniem.com.dz

[14]http://ethicpedia.org/IMG/pdf/Processus5immobilisationsV1_1.pdf

[15]<http://www.journaldunet.com/business/pratique/dictionnaire-comptable-et-fiscal/14442/amortissement-lineaire-definition-calcul-traduction.html>

[16]<http://eric-bellot.fr/files/tutorielwampserver.pdf>

[17]<http://eclipse.developpez.com/actu/72543/Eclipse-Luna-la-nouvelle-version-de-l-IDE-apporte-le-support-natif-de-java-8/>