

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

UNIVERSITE MOULOUD MAMMARI 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

Sofiane OUCHAREF

Mohamed OUIKENE

Thème

Conception et réalisation d'une application Web pour la gestion des archives (plans)

Cas : Entreprise d'ELECTRO-INDUSTRIE Azazga

Mémoire soutenu publiquement le 18/07/2016 devant le jury composé de :

Président : Mr KERBICHE

Encadreur : Mr RAMDANE

Co-Encadreur

Examineur : Mr OUAMRANE

Examinatrice : Mme BEL ATTAF

Remerciment

Louange à ALLAH de nous avoir donné le courage et la volonté pour la réalisation de notre projet.

Nous tenons à exprimer notre profonde gratitude à notre promoteur, monsieur RAMDANE Mohamed pour nous avoir encadré durant cette année, ainsi que pour ses conseils judicieux ; et on tient aussi à remercier ADDI Khelifa qui nous a beaucoup aidé et encouragé.

Que les membres du jury trouvent ici nos plus vifs remerciements pour avoir accepté d'honorer par leur jugement notre travail.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que ce modeste travail voit le jour.

Je dédie ce modeste travail à :

- + Ma mère et ma femme qui m'ont aidées et soutenues tout au long de ce projet et à mes deux filles Sarah et Asma...*
- + Mes amis qui m'ont encouragés.*
- + Mon binôme et toute sa famille.*

Ouikene Mohamed

Je dédie ce modeste travail à :

- + Mes chers parents qui m'ont beaucoup aidés et soutenues toute ma vie...*
- + Mon frère et mes sœurs et à toute ma famille.*
- + Mes amis qui m'ont encouragés.*
- + Mon binôme et sa famille.*

Oucharef Sofiane

Résumé

Notre mémoire qui est encadré par Mr RAMDANE Mohamed consiste à concevoir et développer une application Web pour la gestion des archives (plan) pour l'entreprise d'ELECTRO-INDUSTRIE ex : ENEL.

On a structuré le mémoire en quatre chapitres comme suite :

- Premier chapitre : « Généralités sur le Web » ; nous avons bien définie le Web et ces concepts et on a parlé des sites Web et de l'architecture Client /serveur.
- Deuxième chapitre : « Etude de l'existant » ; on a présenté l'organisme d'accueil 'notre espace de travail' et le besoin de l'entreprise à cet application qui gère ses milliers de plans.
- Troisième chapitre : « Analyse et conception » ; ou on a choisie l'UML comme langage de modélisation orienté objet.
- Quatrième chapitre : « Réalisation » ; nous avons réussi a faire une application convivial pour l'entreprise , en utilisant l'éditeur de texte Dreamweaver , la base de donnée SQL , et le serveur Apache , et les langages de programmation HTML,PHP et JavaScript. Et nous avons définie aussi quelques interfaces de notre application.

L'application faite répond convenablement au désir des employés de l'entreprise car ils ont passé du monde manuelle au monde informatique où ils gagnent beaucoup de temps et ne risquent pas de perde des informations ou de les oubliées.

Sommaire

Chapitre 1 : Généralités sur le Web

I. Le web.....	(03)
I.1. Historique du Web.....	(03)
I.2. Définition.....	(03)
I.3. Les concepts du Web.....	(04)
II. Les sites Web.....	(05)
II.1. Définition.....	(05)
II.2. Classification des sites Web.....	(05)
. Les sites Web statiques.....	(05)
. Les sites Web dynamiques.....	(06)
III. Architecture Client/serveur.....	(07)
III.1. Introduction de l'architecture Client/serveur.....	(07)
III.2. Les architectures Client/serveur.....	(07)
III.2.1. Avantages de l'architecture.....	(08)
III.2.2. Inconvénients de l'architecture.....	(08)
III.3. Fonctionnement d'un système Client/serveur.....	(09)
III.4. Classification des architectures Client/serveur.....	(10)
III.4.1. Le Client/serveur à deux tiers.....	(10)
III.4.2. Client/serveur à trois niveaux.....	(12)
. Caractéristiques de l'architecture à trois niveaux.....	(14)
III.4.3. Architecture Client/serveur à multi-tiers.....	(14)
. Répartition des données sur différent serveurs.....	(15)
. Répartition de la logique applicative.....	(15)
Conclusion.....	(18)

Chapitre 2 : Etude de l'Existant

I. Historique de l'organisme d'accueil.....	(20)
II. Les produits de l'ELECTRO-INDUSTRIE.....	(21)
III. Localisation de l'entreprise.....	(21)
IV. Description de la structure.....	(21)
V. Organigramme.....	(24)
VI. Situation informatique de l'entreprise.....	(25)
VI.1. Organisateur de la fonction informatique.....	(25)
VI.2. Inventaire du matériel informatique existant.....	(25)
VI.3. Ressources humains affectés.....	(26)
VI.4. Perspective.....	(27)
VI.5. Problématique (critiques).....	(27)
VI.6. Solutions.....	(27)
Conclusion.....	(27)

Chapitre 3 : Analyse et conception

Introduction.....	(29)
I. Les objectifs de notre travail.....	(29)
II.1. Analyse et conception.....	(30)
II.1.1. Méthode d'analyse et conception.....	(30)
II.1.1.1. Présentation d'UML.....	(30)
II.1.1.2. Quelques définitions de bases.....	(30)
II.1.2. Démarche.....	(30)
II.1.2.1. Identificateur des acteurs.....	(30)
II.1.2.2. Les cas d'utilisation global.....	(31)
II.1.2.3. Diagramme de cas d'utilisation global.....	(31)
II.1.2.4. Spécification des cas d'utilisation global.....	(33)

II.1.2.5. Spécification des scénarios.....	(34)
II.1.2.6. Description textuelle des cas d'utilisation.....	(38)
II.2. Conception.....	(46)
II.2.1. Elaboration des diagrammes de séquences	(46)
II.2.3. Conception de la base de donnée.....	(52)
II.2.3.1. Le diagramme de classe globale.....	(52)
II.2.3.2. Le modèle conceptuel physique de base de donnée.....	(53)
Conclusion.....	(54)

Chapitre 4 : Réalisation

Introduction.....	(56)
I.1. Outils de développement.....	(56)
I.2. Les serveurs	(57)
I. 2.1. Serveur Apache.....	(57)
I. 2.2. Serveur MySQL.....	(57)
I.3. Logiciel Dreamweaver.....	(58)
I.4. Les langages de programmation utilisés.....	(59)
1. HTML (HyperText Mark-up language)	(59)
2. Le langage PHP(Pre HyperTextProcessor).....	(59)
3. SQL (Structured Query Language).....	(60)
4. JavaScript.....	(60)
I.5. Le modèle physique de données.....	(60)
II. Les interfaces de notre application.....	(62)
Conclusion.....	(67)
Conclusion Générale.....	(68)
Références bibliographique.....	(69)

Table de figures

Figure -1- : Schéma du client/serveur	p9
Figure -2- : Architecture client-serveur à deux tiers.....	p10
Figure -3- : Schéma architecture trois niveaux.....	p12
Figure -4- : Architecture trois niveaux	p13
Figure -5- : Architecture multi-tiers.....	p16
Figure -6- : Chiffres d'affaire 2007-2011.....	p20
Figure -7- : Diagramme de cas d'utilisation global.....	p32
Figure -8- : Cas d'utilisation S'Authentifier.....	p38
Figure -9-: Cas d'utilisation Ajouter un plan moteur.....	p38
Figure -10-: Cas d'utilisation modifier informations plan moteur.....	p39
Figure -11-: Cas d'utilisation supprimer un plan moteur.....	p39
Figure -12-: Cas d'utilisation d'un plan remis moteur.....	p39
Figure -13-: Cas d'utilisation d'un plan restitué moteur.....	p40
Figure -14-: Cas d'utilisation d'un plan modifié moteur.....	p40
Figure -15-: Cas d'utilisation d'un plan non modifié moteur.....	p40
Figure -16-: Cas d'utilisation éditer l'avis de modifications plan moteur.....	p41
Figure -17-: Cas d'utilisation éditer la fiche plan moteur.....	p41
Figure -18- : Cas d'utilisation uploader un plan moteur.....	p41
Figure -19-: Cas d'utilisation Ajouter un plan transformateur.....	p42
Figure -20-: Cas d'utilisation modifier informations plan transformateur.....	p42
Figure -21-: Cas d'utilisation supprimer un plan transformateur.....	p42
Figure -22-: Cas d'utilisation d'un plan remis transformateur.....	p43
Figure -23-: Cas d'utilisation d'un plan restitué transformateur.....	p43
Figure -24-: Cas d'utilisation d'un plan modifié transformateur.....	p43
Figure -25-: Cas d'utilisation d'un plan non modifié transformateur.....	p44
Figure -26-: Cas d'utilisation éditer l'avis de modifications plan transformateur.....	p44
Figure -27-: Cas d'utilisation éditer la fiche plan transformateur.....	p44
Figure -28- : Cas d'utilisation uploader un plan transformateur.....	p45
Figure -29-: Cas d'utilisation Changer le mot de passe.....	p45

Figure -30-:Diagramme de séquence du cas d'utilisation S'authentifier.....	p48
Figure -31-:Diagramme de séquence du cas d'utilisation ajouter un plan.....	p49
Figure -32-:Diagramme de séquence du cas d'utilisation Rechercher les plans.....	p50
Figure -33-:Diagramme de séquence du cas d'utilisation changer mot de passe.....	p51
Figure -34- : Diagramme de classe globale (données).....	p52
Figure -35- : interface PHPMyAdmin.....	p56
Figure-36- : Interface du logiciel Dreamweaver.....	p58
Figure-37- : Interface d'authentification.....	p62
Figure -38-: Interface Ajouter plan.....	p63
Figure -39-: Interface Rechercher plan.....	p64
Figure -40-: Interface modifié plan.....	p65
Figure -41-: Interface visualisation plan.....	p66
Figure -42-: Interface Modifier mot du passe.....	p67

Introduction Générale

Il ne fait désormais plus aucun doute que l'informatique représente la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité en ce siècle passé. L'informatique vient nous apporter de multiples confort à notre mode de vie.

Aucun domaine n'est resté étranger à cette stratégie qui offre tant de services aussi bien pour l'entreprise ou l'administration que pour le personnel.

L'application web comme premier en la matière de l'information et de communication est vite devenu le domaine de commerce pour les entreprises. Dans ce cadre s'inscrit notre projet de fin Licence qui consiste à réaliser une application Web de gestion des archives pour l'entreprise ELECTRO-INDUSTRIES.

Pour se faire nous avons adopté le plan de travail suivant :

- Le premier chapitre intitulé « Généralités sur le Web », présente quelques notions de base concernant le Web et le modèle client/serveur.
- Le deuxième chapitre porte sur le domaine de notre application où on définit quelques notions concernant l'entreprise ELECTRO-INDUSTRIES.
- Le troisième chapitre qui s'intitule « *Analyse et conception* », est consacré à l'analyse et à la conception de l'application proprement dite.
- En fin le quatrième chapitre qui porte sur la réalisation et l'implémentation de l'application ainsi que son fonctionnement.

I. Le Web :

I.1. L'historique du Web :

Le World Wide Web (www) a été mis en place par Tim-Berners Lee qui est considéré comme le père fondateur du web. Au milieu des années 1990, Internet fait son apparition au grand public en version 1.0 via des pages statiques codés en HTML. Il s'agit de sites non interactifs principalement destinés { la recherche d'informations : encyclopédies, etc.

Au début des années 2000, le web a évolué et il est devenu dynamique. Il s'agit de sa version 1.5. Il est maintenant possible de consulter du contenu dynamique en ligne, via des bases de données : boutique en ligne, etc. (A noter que c'est aussi l'avènement des start-up qui surfent sur la vague du web dynamique, accessible à tous).

En 2004, Dale Dougherty utilise le terme « Web 2.0 » qui sera vite repris par Tim O'Reilly, spécialiste du World Wide Web ; l'apparition du web 2.0 prend une véritable dimension sociologique ,pour Joshua Porter, le web 2.0 « c'est le partage de l'information, fondé sur des bases de données ouvertes qui permettent a d'autres utilisateurs de les employer.

Depuis quelques mois, on entend de plus en plus parler du Web 3.0, c'est-à-dire un web encore plus humain, encore plus intelligent.

I.2. Définition :

Le world wide web (ou WWW ou W3, souvent appelé Web) signifie littéralement la "toile d'araignée recouvrant le monde" est le service d'information le plus récent d'internet, il est constitué de millions de pages webs, résident sur des serveurs web et accessible via leurs adresses internet respective ou URLs, par le biais d'un navigateur (explorateur) web.

Le web constitue une véritable bibliothèque virtuelle d'où l'on peut contenir un grand nombre de documents. son taux de croissance a été flagrant depuis les dernières années et s'est vite établie comme une ressource d'information incontournable.

I.3. Les concepts du web:

- **Page web** : Une page web est une ressource du world wide web conçue pour être consultée par des visiteurs à l'aide d'un navigateur web.
Techniquement, une page web est souvent constituée d'un document en HTML et d'images.
- **Navigateur web** : C'est un logiciel client qui permet à l'utilisateur de chercher et de consulter des documents sur internet, et d'exploiter les liens hypertextuels qu'ils comportent.
- **HTTP (Hyper Text Transfer Protocol)** : Le plus utilisé des protocoles de communication sur le world wide web. Il a un client web d'indiquer quelle page il veut obtenir, et au serveur web de lui répondre en lui donnant cette page.
- **URL (Uniform Resource Locator)** : est une chaîne de caractères respectant un schéma de construction bien précis. Il permet d'accéder à une ressource donnée, via internet. Dans le web cette ressource est un document ou un fragment.
- **FTP (File Transfert protocol)** : est un protocole de transfert de fichiers, qui est utilisé pour le transfert de fichiers sur internet.
- **Moteur de recherche** : Les moteurs de recherche sont des serveurs qui indexent un certain nombre de sites web et vous permettent généralement de rechercher les documents qui vous intéressent à l'aide d'un mot clé.
- **Hyper lien** : est une référence dans un système hypertexte permettant de passer automatiquement d'un document consulté à un document lié. Les hyperliens sont notamment utilisés dans le world wide web pour permettre le passage d'une page web à une autre d'un simple clic. Un hyperlien a une source (origine) et une destination (cible).
- **Hyper Text** : Un système hypertexte est un système contenant des nœuds liés entre eux par des hyperliens permettant de passer automatiquement d'un nœud à un autre. Un nœud est une unité minimale d'informations.
- **Site web** : aussi appelé site internet par abus de langage.

II. Les sites Web :

II.1. Définition :

Un site web est un ensemble de pages écrites dans un langage normalisé de manière à être compris par tous les navigateurs. L'organisme qui décrit et décide des évolutions de ce langage s'appelle le W3C, et préconise l'utilisation du HTML pour la création de pages web.

II.2. Classification des sites web :

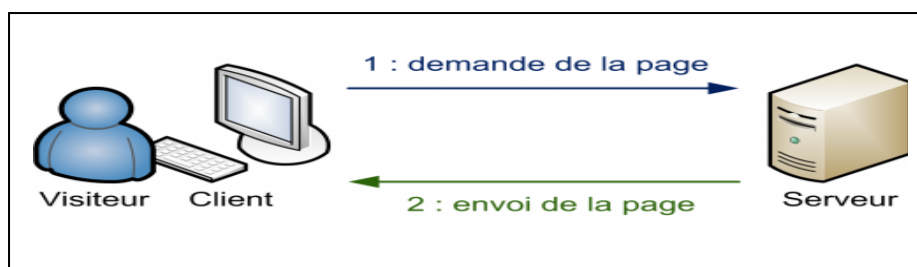
On peut classer les sites web selon le type des pages qu'ils offrent comme suite :

❖ Les sites web statiques :

On appelle site web statique la création d'un site internet composé de pages web statiques. Une page est dite statique lorsque son contenu est fixe, invariable quel que soit l'utilisateur qui la consulte, contrairement aux pages d'un site web dynamique dont le contenu s'affiche en fonction d'une demande de l'utilisateur.

Un site web statique est moins coûteux qu'un site web dynamique, sauf si le site est particulièrement volumineux ou que la priorité est mise sur un design pointu. Toutes les pages sont individualisées ce qui permet d'obtenir un bon référencement. L'historique du site est également conservé dans sa globalité et en l'état sur le net, ce qui facilite le travail de référencement sur la durée.

Le coût de développement d'un site statique est moins élevé que celui d'un site dynamique. Cependant, les mises à jour requièrent certaines compétences informatiques (langage de programmation et notions de webdesign). Si ces compétences ne sont pas présentes dans l'entreprise, le recours à un prestataire externe sera nécessaire. Des mises à jour fréquentes et un volume important du site web rendront ce travail de suivi plus lourd à gérer.



❖ Les site web dynamiques :

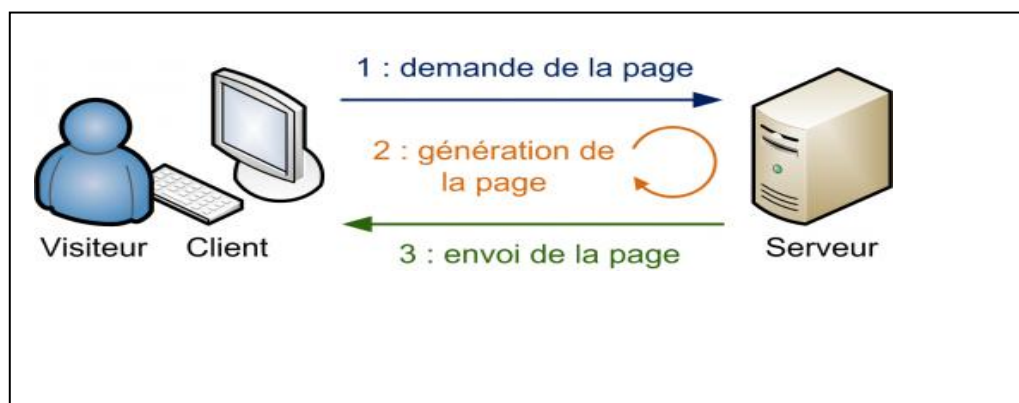
Un site web dynamique est un site dont le contenu peut être généré dynamiquement, c'est à dire que ce contenu s'affichera en fonction de l'utilisateur qui le consulte ou d'autres paramètres. Les sites web dynamiques sont créés dans des langages plus complexes que les sites web statiques (ou site vitrine) et incluent l'utilisation de bases de données, ce qui offre bien plus de possibilités de développement que pour un site web statique.

Les avantages d'un site web dynamique:

Les mises à jour ne nécessitent pas de connaissances informatiques spécifiques et peuvent être réalisées en interne : pas de risque de dépendance vis-à-vis d'un webmaster.

Les sites web dynamiques sont évolutifs : on peut leur greffer une multitude de modules. Ils permettent des fonctionnalités quasi illimitées en termes d'interactivité : newsletters, agendas, paiements en ligne, forums,...

Le coût de développement d'un site dynamique est plus élevé que celui d'un site statique. Cependant, les mises à jour étant réalisées au sein de l'entreprise, les économies prennent rapidement le pas sur l'investissement de base.



III. Architecture client/serveur :

III.1 Introduction de l'Architecture Client /serveur :

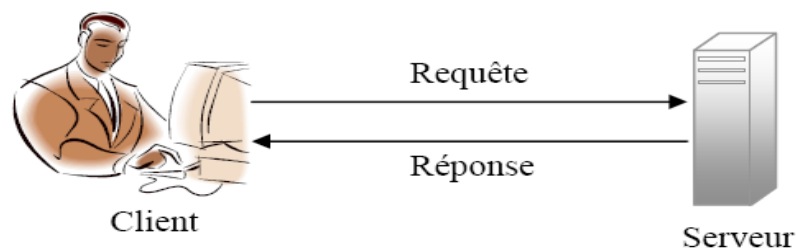
Dans un monde où la course à la productivité conduit les technologies à évoluer de plus en plus vite, le client/serveur s'est taillé une part de choix dès le début des années 90. Dans cette partie, Nous proposons une vue d'ensemble sur l'architecture client/serveur tout en définissant les notions de bases indispensables à la compréhension du modèle.

III.2 Les architectures Client /serveur : [GEOR 96]

Le client-serveur est avant tout, une technique de dialogue entre deux processus, l'un **Le Client** sous-traitant à l'autre **le serveur** des fonctions à réaliser.

Le modèle de communication Client/serveur est orienté vers la fourniture de services par un Processus serveur à un processus client. Un échange consiste donc en la transmission d'une Requête à un serveur, qui exécute l'opération demandée et envoi en retour la réponse. Nous Définissons ci-dessous plus précisément ces concepts de bases.

- **Client** : (*client*) Processus demandant l'exécution d'une opération à un autre processus Par 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** : (*server*) Processus accomplissant une opération sur demande d'un client, et Transmettre la réponse à ce client.
- **Réponse** : (*reply*) Message transmis par un serveur à un client suite à l'exécution d'une Opération contenant les paramètres de retour de l'opération.
- **Requête** : (*request*) Message transmis par un client à un serveur décrivant l'opération a Exécuté pour le compte client.



III.2.1 Avantages de l'architecture :

Le modèle client/serveur est particulièrement recommandé pour les réseaux nécessitant un grand niveau de fiabilité, ces principaux avantages sont :

- **Des ressources centralisées** : étant donné que le serveur est le centre du réseau il peut gérer des ressources communes à tous les utilisateurs comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.
- **Une meilleure sécurité** : car le nombre de points d'entrées permettant l'accès aux données est moins important.
- **Une administration au niveau de serveur** : l'administrateur de serveur s'occupe de la gestion de réseau.
- **Un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et modification majeur.

III.2.2 Inconvénients de l'architecture :

L'architecture client /serveur à tout de même quelques lacunes parmi lesquelles :

- **Un coût élevé** : le serveur est une machine très puissante et sophistiquée qui est fabriquée avec du matériel de haut technologie, car elle a beaucoup d'instructions à exécuter en temps réel (satisfaire tous les clients en temps voulu) et une énorme quantité de données.
- **Un maillon faible** : le serveur est seul maillon faible du serveur, étant donné que tout le réseau est architecturé autour de lui. Heureusement, le serveur à une grande tolérance aux pannes.

III.3 Fonctionnement d'un système client/serveur :

Un système client/serveur fonctionne celant le schéma suivant :

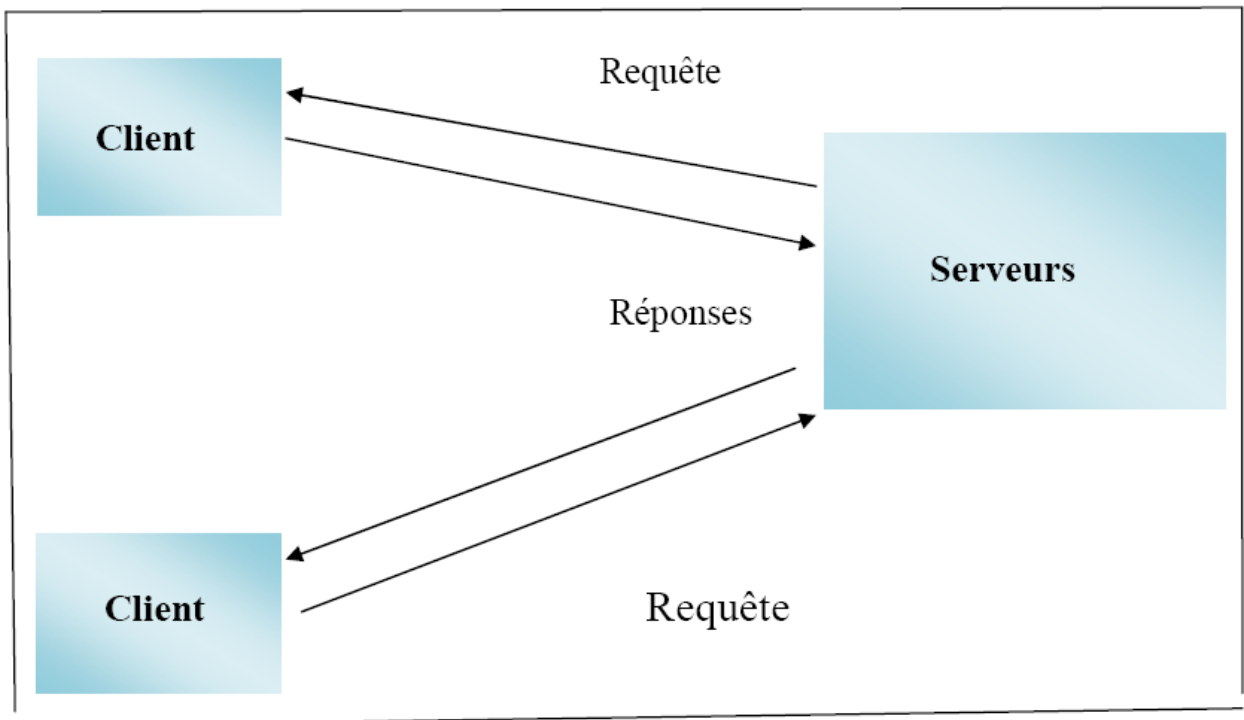


Figure-1-: schéma du client/serveur

Le client émit une requête vers le serveur grâce à son adresse IP et le port, qui désigne Un service particulier du serveur. Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine client et son port.

III.4 Classification des architectures client/serveur :

III.4.1 Le client serveur à deux-tiers: [ENSI 00]

En ce qui concerne les architectures à deux niveaux, on distingue deux approches :

La première consiste à séparer les données des traitements et de l'interface Utilisateur. Ainsi les données se trouvent côté serveur, tandis que la partie applicative est Intégralement installée côté client. Cette partie inclut naturellement tous les traitements De l'application, aussi bien les traitements fonctionnels que la présentation des Informations. Les données sont généralement stockées au niveau d'un SGBD ; pour Obtenir celles qui sont nécessaires au traitement en cours. IL faut déclencher des requêtes, Qui sont alors transmises au moteur de base de données via un "middleware" d'accès aux Données. Le dialogue entre le client et le serveur à travers le réseau se limite à des Requêtes sur les données.

Dans la deuxième approche, on sépare la présentation et interface utilisateur des Traitements, pour les isoler du côté client et côté serveur, on trouve comme pour la Première solution les données mais également les traitements possibles sur ces données, la Communication était bien entendue assurée par un "middleware". Une architecture à deux Niveaux permet de mettre facilement et rapidement en œuvre des projets de petites tailles.

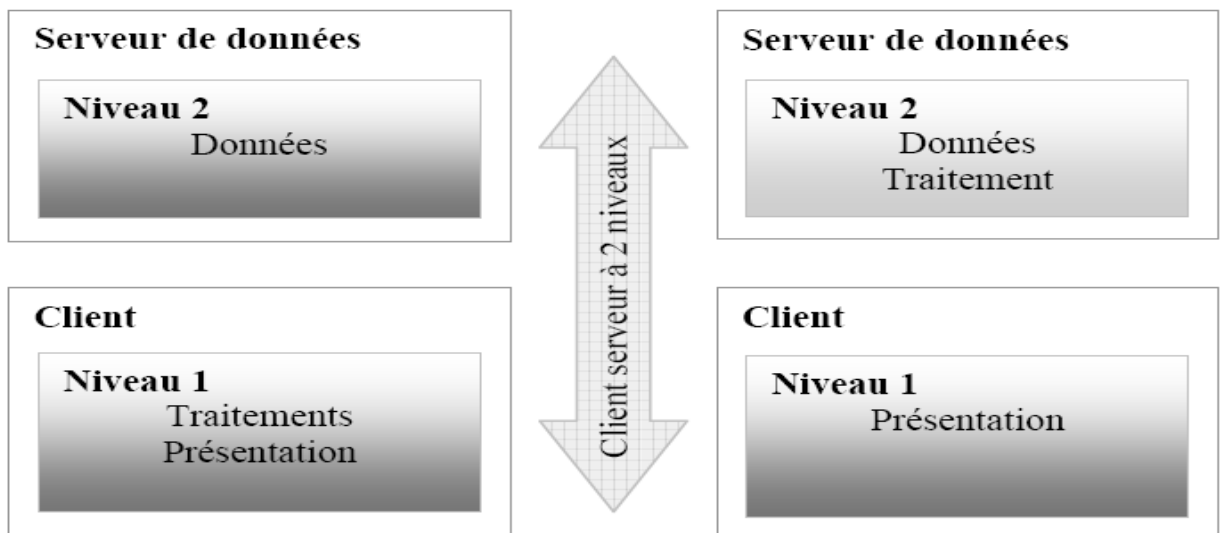


Figure-2-: Architecture client-serveur à deux tiers : deux approches possibles.

- ❖ Le "Middleware" se définit comme l'ensemble des logiciels mis en œuvre "en dessous des applications, au-dessus des O.S (Operating System). Et entre les plates-formes" pour

assurer les échanges entre les composants : Présentation, Traitement et Accès aux données dans les architectures Client/serveur. Autrement Dit, c'est une couche logicielle entre l'application du poste client et le serveur (Web, de base de données comme ODBC, de messagerie électronique, etc.)

Constituée d'une interface de programmation d'application (API) et d'un Protocole réseau (comme TCP/IP). API décrit le type d'interface entre L'application cliente et le serveur distant via un protocole de communication et de Formatage de données.

Les principales fonctions attendues du "Middleware" sont :

- La localisation des Services par un adressage symbolique à la base d'un annuaire réseau.
- Les échanges de messages entre Clients et Services, en mode synchrone (requête/réponse) ou asynchrone (via des files d'attente).
- Les mécanismes de formatage et conversion de données (requis du fait de l'hétérogénéité des plates-formes).
- Les fonctions de sécurité (authentification, contrôle d'accès, ...).

Le problème de ce type d'architecture se situe moins au niveau des performances qu'au niveau du déploiement des clients et de la maintenance. Les traitements fonctionnels sont par nature fortement liés à l'interface utilisateur, ce qui provoque des incidences sur la clarté des programmes, et sur la faculté d'isoler les traitements fonctionnels du reste : un isolement des diverses couches logicielles est quasiment impossible, ainsi des changements affectant la base de données ou le middleware de communication peuvent avoir des répercussions considérables sur l'ensemble de l'application.

Pour palier aux inconvénients de cette architecture, La communauté informatique a fondé la notion d'architecture à trois niveaux. Donc, l'application est divisée en trois niveaux logiques : la présentation, la logique applicative et les données.

Limites de l'architecture à deux-tiers :

Ce type d'architecture ne nécessite pas l'installation d'un gestionnaire de base de données sur chaque machine client mais elle présente les inconvénients suivants :

- Une architecture utilisée pour les petites applications

- Dans la première approche, les applications client à distance sont lourdes.
- Le non exploitation des machines clients dans la deuxième approche.

III.4.2 Client serveur à trois niveaux : [ENGR 00] et [ANDR 00]

Ce modèle est généralement conforme aux applications Web, Avec une application à trois niveaux, une application est divisée en trois niveaux logiques distincts, chacun d'entre eux étant pourvu d'ensemble d'interfaces bien définis. Le premier, appelé **niveau métier**, recouvre la logique métier (business logico) ou logique applicative. Le troisième niveau ou **niveau données**, contient les données nécessaires à l'application.

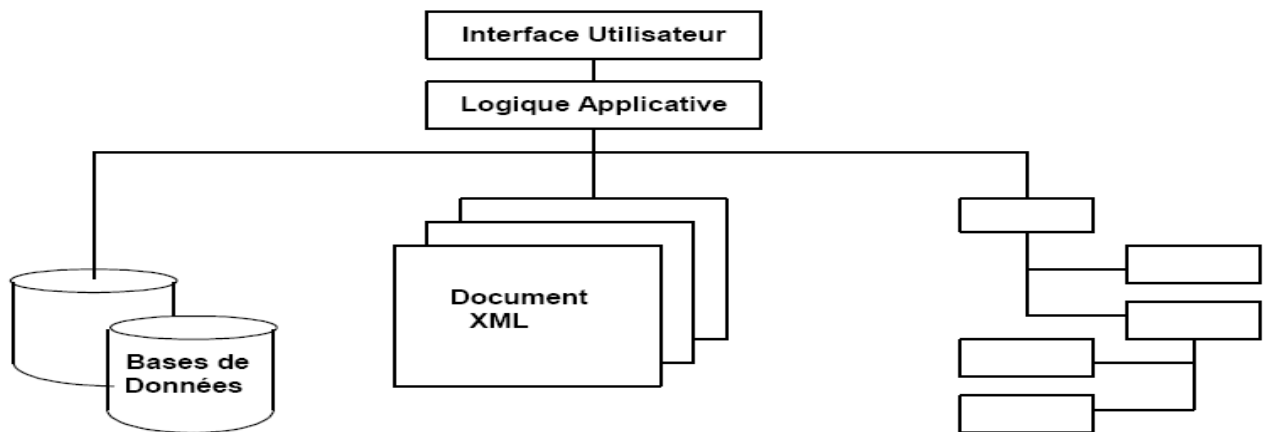


Figure-3- : schéma architecture trois niveaux.

- Le niveau intermédiaire (logique applicative) est composé du code appelé par l'utilisateur (via le niveau présentation) pour extraire les données utiles. Le niveau présentation reçoit alors les données et les formate en vue de leur affichage. Cette séparation de la logique applicative de l'interface accentue considérablement la souplesse de conception de l'application. Il devient alors possible de construire et de déployer de nombreuses interfaces utilisateur sans devoir modifier la logique applicative, à condition toutefois, que cette dernière soit dotée d'une interface parfaitement définie avec le niveau présentation.
- Le troisième niveau abrite les données nécessaires à l'application. Il peut s'agir de n'importe quelle source d'information : base de données d'entreprises Oracle ou Sybase, jeu de documents XML (c'est-à-dire des données ayant été enregistrées dans

des documents élaborés selon les spécifications XML), ou de service d'annuaire tel qu'un serveur LDAP (Lightweight Directory Access Protocol). Outre les bases de données relationnelles classiques, votre application peut ainsi avoir accès à nombreuses sources de données différentes relatives à l'entreprise.

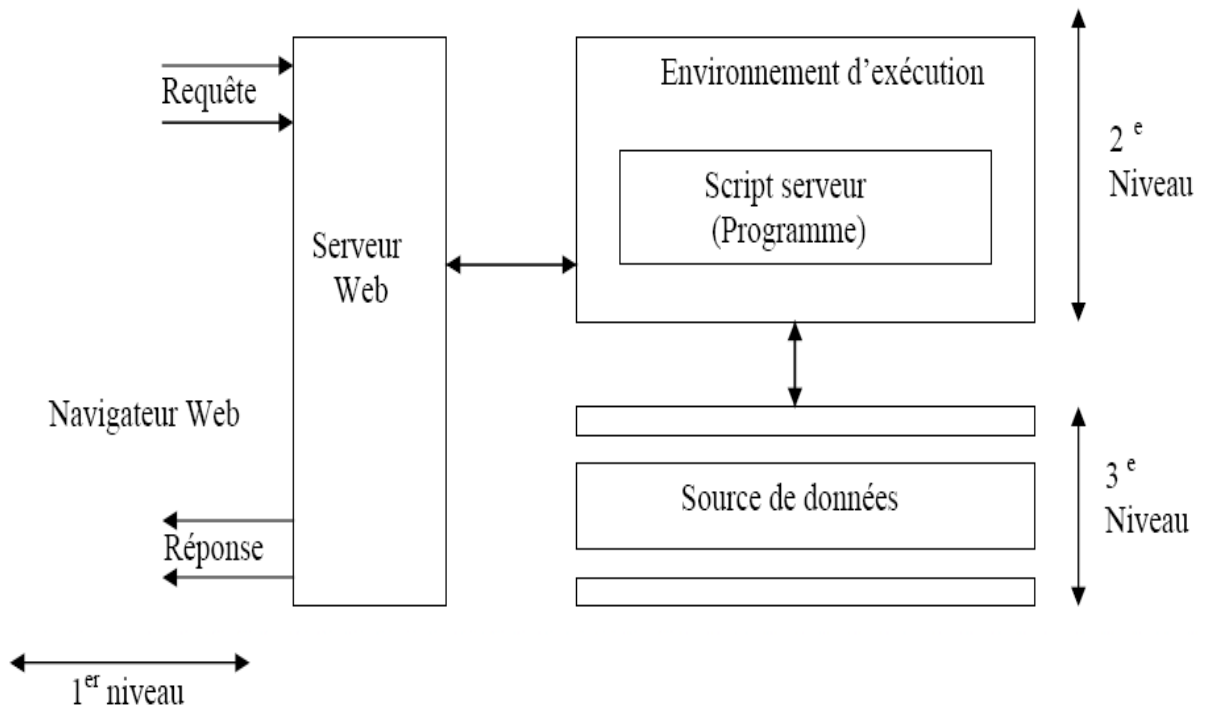


Figure-4- : Architecture trois niveaux.

Caractéristiques de l'architecture à trois tiers :

- De meilleures performances (les tâches sont partagées sur les différents serveurs).
- L'installation d'un gestionnaire de base de données n'est obligatoire que sur le serveur de base de données.

III.4.3 Architecture client /serveur multi-tiers :

Nous avons vu ce qu'était une architecture trois-tiers, une architecture multi-tiers va plus loin dans le découpage de l'application sur différents serveurs.

Elle est également appelée architecture distribuée du fait de la distribution des traitements et des données sur différents serveurs, lorsque le thème de développement d'application en entreprise est abordé, il est judicieux d'évoquer le concept d'architecture multi niveaux (n-tiers architecture).

Le découpage de base du système reste toujours le même : une partie gestion de données, une partie gestion de la logique applicative et bien entendu le client assurant la présentation. Toutefois les deux parties développées côté serveur vont pouvoir être déployées chacune sur plusieurs serveurs.

L'objectif général de ce type d'architecture est de permettre l'évolutivité du système sous plusieurs aspects : la quantité de données stockée, la disponibilité du serveur, le nombre d'utilisateurs, etc.

Un système conçu en architecture multi-tiers n'est pas forcément déployé sur plusieurs machines dès le départ. Toutefois son mode de programmation doit permettre de modifier le déploiement du système en cours d'exploitation par un administrateur. Le développeur doit donc rendre le système indépendant du serveur sur lequel il s'exécute. Il existe deux types de répartitions possibles dans une architecture distribuée. Il est possible de répartir les données et de répartir la logique applicative. Chacune de ces deux répartitions permet de résoudre des

Problèmes de natures différentes. Elles peuvent donc être mises en place soit séparément soit en Parallèle sur le même système.

- **Répartition des données sur différents serveurs :**

Il s'agit d'utiliser plusieurs sources de données dans une même application. Chaque Source possède sa propre structure de données. Chaque serveur de données peut être géré de façon très différente. Toutefois une interface de programmation commune à toutes les sources doit pouvoir exister. Il peut exister des relations entre les données des différents serveurs, il sera alors nécessaire de gérer des transactions distribuées entre des différents serveurs de données. Cette répartition de données correspond à la notion de base de données distribuée.

- **Répartition de la logique applicative :**

La répartition de la logique applicative permet de distribuer les traitements sur différentes machines. Cette distribution est facilitée par l'utilisation de la programmation orientée objet et plus particulièrement de ce qu'on appelle les composants. Un composant possède entre autre la caractéristique d'être accessible à travers le réseau. Un composant peut ainsi être instancié puis utilisé au travers du réseau.

Il est également possible de trouver un serveur permettant l'utilisation d'un composant, ce qui permet une forte évolutivité ainsi qu'une résistance aux pannes importantes (le service sera toujours disponible sur un serveur même si une machine tombe en panne). La réalisation de ce type de répartition nécessite l'utilisation de technologies spécifiques.

Les buts recherchés lors de la mise en place de ce type d'architecture sont la performance, l'évolutivité, la maintenabilité etc. La figure suivante présente un exemple d'architecture multi-tiers.

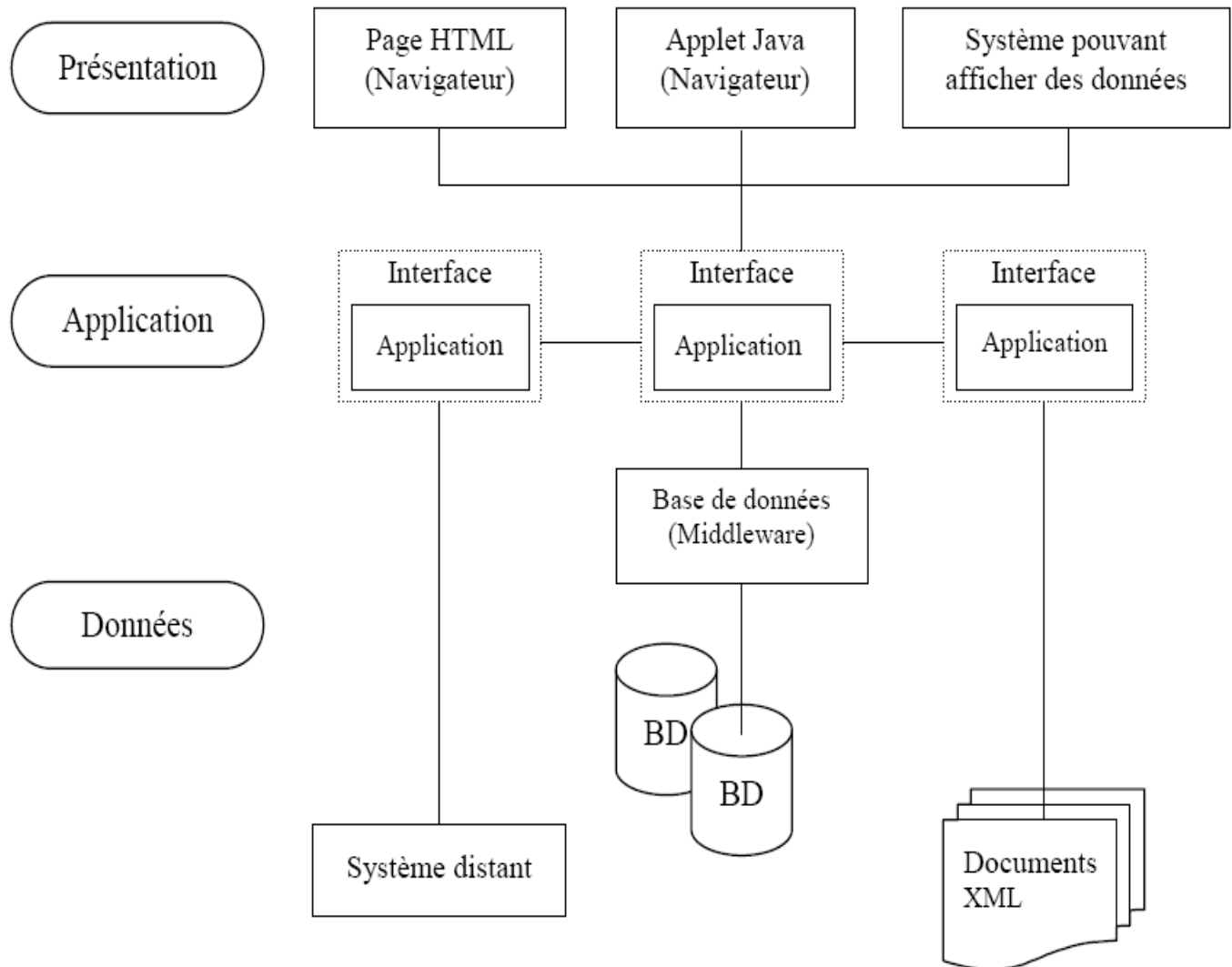


Figure-5- : Architecture multi-tiers.

Les avantages et les inconvénients des deux architectures (deux niveaux et trois niveaux) sont présentés dans le tableau suivant : [NICO 98]

Architecture	Avantages	Inconvénients
2 niveaux	<ul style="list-style-type: none"> - Mise en œuvre rapide - Programmation simple. - Nombreux outils disponibles 	<ul style="list-style-type: none"> - Peu extensible. - Sécurité faible. - Pas de séparation nette entre l'interface homme-machine (IHM) et les traitements. - Surcharge du client.
3 niveaux	<ul style="list-style-type: none"> - Les applications clients sont de petites tailles et ne sont pas concernées par l'installation, la configuration et la gestion du logiciel de connexion aux bases de données. - Indépendance complète entre la présentation, le traitement et la source de données. - Meilleure sécurité. - Extensibilité de l'architecture. - La répartition du travail d'une application entre plusieurs machines peut améliorer la performance par un meilleur équilibrage de la charge et permet aux systèmes redondants de prendre le relais en cas de défaillance d'un serveur. 	<ul style="list-style-type: none"> - Maîtrise de la programmation orientée objet. - Manque d'outils de réalisation.

Tableau : Avantages et les inconvénients.

Conclusion :

Le chapitre a été consacré aux architectures client/serveur. Ce qui a permis de voir les différents types d'architectures existantes.

Les différents concepts traités dans ce chapitre nous aideront à mieux comprendre notre mode d'opération et les notions fondamentales pour mener à bien notre application. Ainsi nous pouvons entamer l'analyse et la conception de notre système.

I. Historique de l'organisme d'accueil

L'entreprise Electro-Industries est créée sous forme actuelle en janvier 1999 avec la restructuration de l'Entreprise mère ENEL (l'Entreprise Nationale des Industries Electrotechniques). ELECTRO-INDUSTRIES est une entreprise publique économique « société par action » dont le capital social est détenu à 100% par la société des participations de l'état CABELLEQ. Elle est constituée d'un complexe intègre, réalisé avec des partenaires allemand(Siemens),qui produit et commercialise des moteurs, transformateurs depuis 1985, date de son démarrage et qui produit aussi des groupes électrogènes depuis l'an 2000.Le complexe est structuré en deux unités durant l'année 2007. ELECTO-INDUSTRIES est doté d'un capitale sociale de 4.753 Millions de Dinars.

Elle emploie actuellement 840 personnes répartie entre les Directions Générales et les deux entités de Production, elle a réalisé un chiffre d'affaire de 2.995 Millions de Dinars pour l'année 2011.

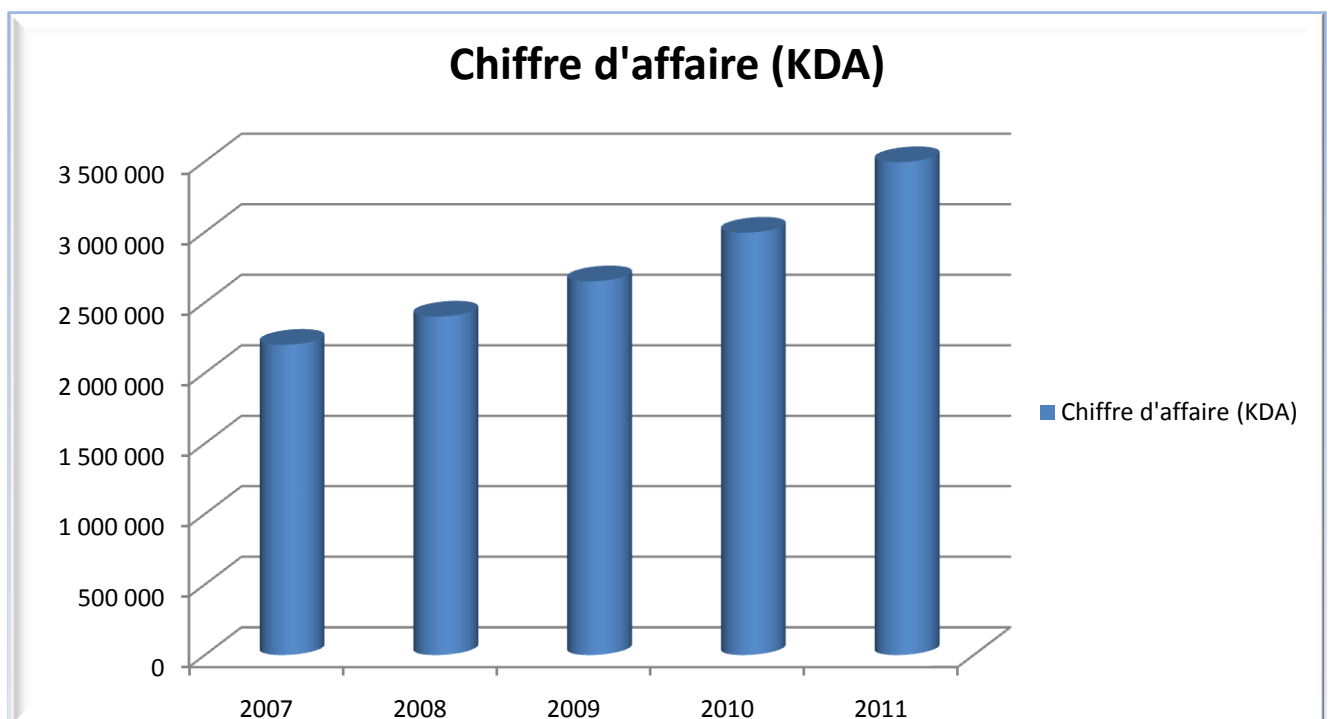


Figure-6- : Chiffres d'affaire 2007-2011.

II. Les produits de l'ELECTRO-INDUSTRIES

Les produits fabriqués par ELECTRO-INDUSTRIES sont conformes aux recommandations CEI et aux normes allemandes DIN/VDE. la production actuelle d'ELECTRO-INDUSTRIELS est écoulee sur le marché algérien et génère un chiffre d'affaire de 2.2 Milliards de Dinars (juin 2012).

La capacité de production de transformateurs de l'entreprise couvre les besoins de marché à 70% environ, les ventes de moteurs représentent 30% environ de ses capacités de production. Il est à signaler que l'ELECTRO-INDUSTRIES est le seul fabricant de ces produits en Algérie (sauf quelques entreprises privées installées récemment). L'ensemble emploie un effectif de 840 agents dont 16% de cadres, 33% de maîtrise et 51% d'exécution.

En matière de qualité, ELECTRO-INDUSTRIES dispose de ses propres laboratoires d'essai et de mesure, de ses produits ainsi pour le contrôle des principaux matériaux utilisés dans sa fabrication. L'entreprise a procédé à la mise en place de son système qualité en 2002 et certifiée par QMI Canada le 24/04/2004, ISO 9001 version 2000.

III. Localisation de l'entreprise

L'entreprise se situe sur la route nationale N°12 à AZAZGA à 35km à l'EST du chef lieu de la wilaya Tizi-Ouzou Et à 150km de la capitale Alger.

IV. Description de la structure

La structure d'une entreprise est un élément primordial, elle conditionne son bon fonctionnement, Son développement et sa réactivité à l'environnement économique. ELECTRO-INDUSTRIES est placée sous l'égide de la société de gestion des participations câbleries et matériels électriques.

Elle se compose d'une direction générale qui à sa disposition cinq sous-direction :

➤ ***Direction technique***

- ✓ Développe et améliore les produits.
- ✓ Contrôle la qualité des produits finis.

➤ ***Direction juridiques contentieux et recouvrement***

- ✓ Conseil juridique.
- ✓ Représentation de l'entreprise devant tout les assistantes juridiques.
- ✓ Prise en charge des dossiers litigieux.
- ✓ Recouvrement de créance par amiable ou par voie judiciaire.
- ✓ Modification des statuts de l'entreprise.

➤ ***DRHO (Direction des ressources humaines et de l'Organisation)***

- ✓ Gestion du personnel (recrutement, formation, retraite,...).
- ✓ Contrat de travail.
- ✓ Les assurances.
- ✓ Gestion des moyennes matérielles.
- ✓ Sécurité des biens des personnes et industriels.

➤ ***DFC (Directeur finances et comptabilité)***

- ✓ Supervise le service comptabilité et finance.

➤ ***DAP (Directeur approvisionnements)***

- ✓ Importation de la matière première.
- ✓ Contrôle la qualité de la matière.
- ✓ Livraison de la marchandise.

Et de deux unités de production qui sont :

➤ ***UTR (Directeur Unité Transformateurs)***

Cette unité assure la production des transformateurs de distribution, elle est constituée d'une direction et de trois départements:

- ✓ Département technique.
- ✓ Département production.
- ✓ Département commercial.

➤ ***UMP (Directeur Unité Moteurs Prestation)***

Cette unité permet de fabriquer des moteurs, des groupes électrogènes et assure l'entretien et la maintenance des équipements annexes.

Mise à part ses cinq sous directions et ses deux unités, ils ont aussi un :

➤ ***Assistant contrôle de gestion et planification***

- ✓ Relation avec le budget de l'entreprise.
- ✓ Planification (gestion du budget).

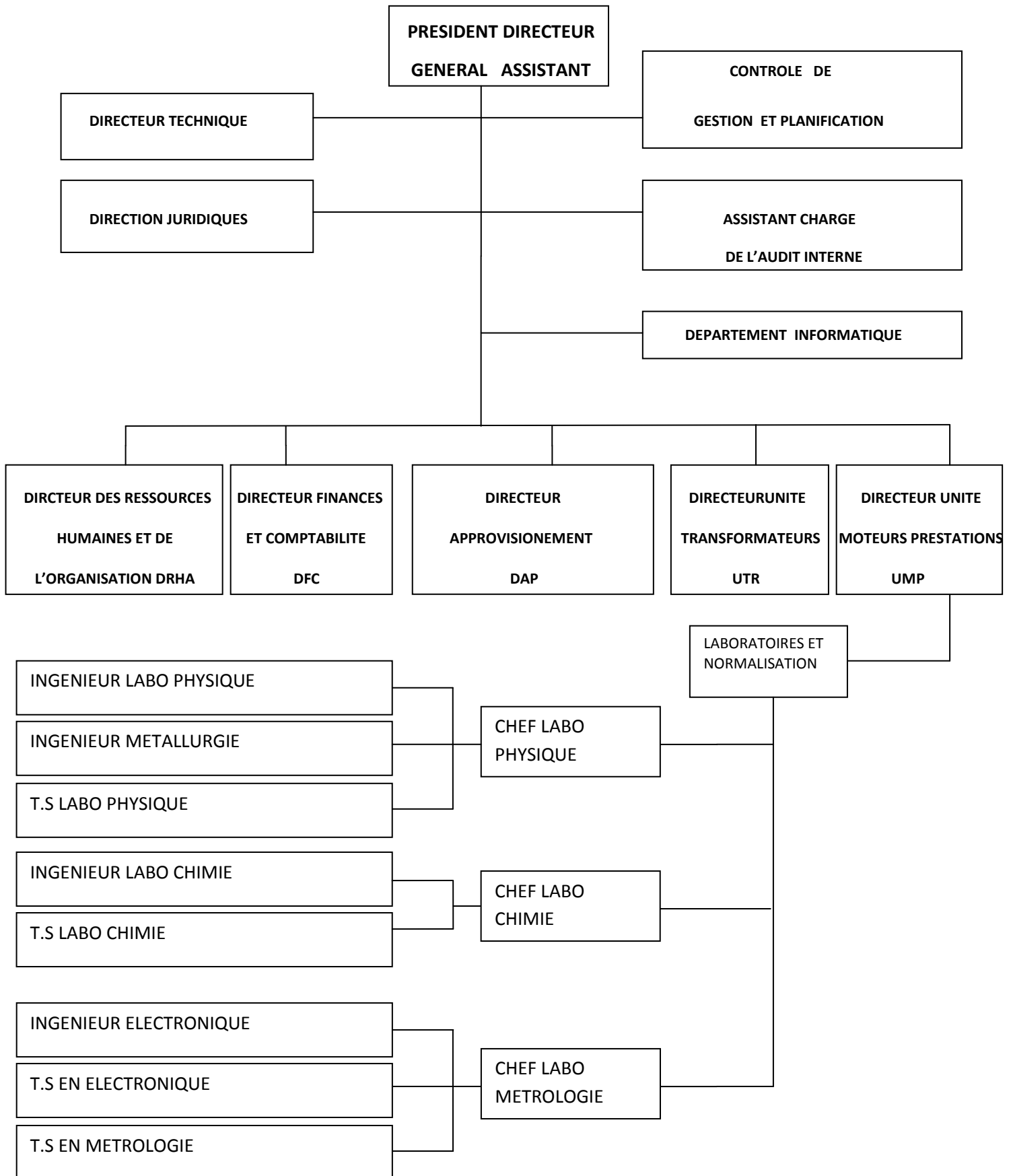
➤ ***Assistant charge de l'audit interne***

L'Audit Interne est une activité indépendante et objective qui vise par ses missions à atteindre deux objectifs principaux à savoir, l'objectif d'une mission donnée et l'objectif de l'audit Interne même en question afin de donner une vision claire à la direction d'une Entreprise et l'assurée quant à la maîtrise des opérations à un niveau acceptable pour l'atteinte de son objectif fixé.

➤ ***Département informatique***

- ✓ L'achat et les maintenances des équipements informatiques.
- ✓ Développement et maintenances des applications.
- ✓ Assure le bon fonctionnement du réseau.

V. Organigramme de l'EPE/ELECTRO-INDUSTRIELS/SPA-AZAZGA



VI. Situation informatique de l'entreprise

VI.1. Organisation de la fonction informatique

Le département informatique est constitué dans son état de quatre bureaux :

- ✓ Le bureau développement et étude.
- ✓ Le bureau gestion de production (GPAO).
- ✓ Le bureau exploitation.
- ✓ Le bureau administration réseau.

VI.2. Inventaire du matériel informatique existant

L'entreprise ELECTRO-INDUSTRIES est dotée de deux systèmes informatiques, l'un de type HP3000-68 (Mini-ordinateur) acquis en 1986, l'autre de type réseau installé récemment HP Pro Xeon ML350G6.

➤ *Inventaire physique du matériel informatique*

- HP-3000-68 :
 - Système HP-3000-68 avec 8 MO de RAM.
 - Un dérouleur de bandes magnétiques.
 - Une quarantaine de terminaux.
 - Une imprimante 900 lignes/mn.
 - 3 Unités de disque de 404 MO chacune (HP 6000 670Hz).
 - 1 imprimante (HP 2563A).
 - 28 modems Sematrans.
 - 6 multiplexeurs HP2334A.
 - 3 multiplexeurs HP2333A.

Compte tenu de la vétusté du matériel informatique existant(HP3000) et du fait qu'il n'a pas passé pas l'an 2000 (Bug de l'an 2000), son renouvellement s'impose.

Pour cela, la solution réseau a été choisie en remplacement du mini-ordinateur existant, aujourd'hui largement dépassé et obsolète.

➤ Réseau de micro-ordinateur

- 02 serveurs de moyenne capacité (Domaine).
- 120 micro-ordinateurs (station de travail dont 80 % sont actifs).
- 01 serveur doté de grandes capacités HP pro liant ML 350 G6 (Pas encore exploité).
- 04 onduleurs (Smart-UPS 7000).
- 01 modem (ZTE ZXDSL 831).
- Goulottes 4*2 ; 5*2.
- Câble 5° FTP RJ45.
- Prises murales (INFR).
- 04 armoires de brassages.
- 04 Switch.
- 04 hubs optiques.
- 04 amplificateurs.
- Des câbles optiques.
- 49 imprimantes dont :
 - ❖ 30 imprimantes matricielles EPSON DEX 9000 (grande capacité).
 - ❖ 18 imprimantes laser (marque HP).
 - ❖ 1 imprimante réseau (marque EPSON DFX 9000).

Les serveurs proposés ont une architecture dotée de deux micros processeurs avec un système de sauvegarde Données. La RAM étant de 256 Mo.

VI.3. Ressources humaines affectées

L'effectif total affecté est de 06 agents dont :

- ✓ Un responsable informatique.
- ✓ Deux informaticiens pour le bureau développement et études.
- ✓ Deux informaticiens pour le bureau exploitation.
- ✓ Un administrateur réseaux.

VI.4. Perspective

Le parc informatique de l'Electro-Industries est vieillissant et pour ce la l'entreprise a envisagé de renouveler Son matériel informatique (Windows 2008 serveur, micro-ordinateur Windows Vista).

VI.5. Problématique (critiques)

Même avec l'existence d'un grand effectif de travail et une bonne situation informatique, le problème de gestion des archives (plans) dans l'unité de transformateurs reste manuel, car quand on a besoin d'un plan pour le modifier ou le supprimer par exemple il faut qu'on le cherche entre 60 milles plans. Ajouter au problème de recherche d'un plan, si un plans est ajouter, modifier ou supprimer on mention tout ça a la main en le sauvegardant manuellement. Cela nous fait perdre beaucoup de temps et risque la perte des données.

VI.6. Solutions

Le principal objectif de notre application consiste à concevoir et développer une application web interactif permettant d'offrir une meilleur gestion des plans. Ainsi notre application devrait assurer les points suivants :

- Offrir une interface conviviale et simple d'utilisation.
- Faciliter l'ajout, la mise à jour et la consultation d'un plan.
- Simplifier la recherche des plans.
- La sauvegarde des informations (plans) dans la base de données.
- La sécurisation des données avec mot de passe.

Conclusion

L'électro-industrie est la principale entreprise dans le domaine de l'industrie électrotechnique, son activité s'adresse Au marché des biens d'équipements. Elle a réalisé un chiffre de 2.495 millions de dinars pour l'année 2008. Mais l'électro-industrie devrait en termes de perspective améliorer son parc informatique.

Introduction :

Dans les deux chapitres précédents nous avons défini le domaine de notre application et les technologies qui participent à son fonctionnement.

Nous présentons dans ce chapitre notre travail qui consiste à développer une application pour l'entreprise ELECTRO-INDUSTRIES «L'ENEL» située sur la route nationale N°12 à Azazga à 35km à l'EST du chef lieu de la wilaya de Tizi-Ouzou et à 150km de la capital Alger.

Nous commencerons par présenter les fonctionnalités de notre application ensuite nous consacrerons une large place pour la partie analyse et conception avec le formalisme UML.

I. Les objectifs de notre travail :

Notre travail consiste à développer une application qui permettra aux acteurs du domaine de l'entreprise de gérer tous types de transaction.

Le principal objectif de notre application est de faciliter la tâche aux administrateurs lors de la gestion des plans (moteurs et transformateurs) de leur entreprise et leurs permettre un gain de temps énorme en leurs délivrant des interfaces claires et faciles à utiliser. Pour ce faire notre système devra :

1. Permettre la consultation et la recherche des différents plans de l'entreprise.
2. Permettre à l'administrateur de faire la mise à jour de différents plans (moteurs et transformateur) de l'entreprise tel que : ajouter ,modifier ou supprimer les plans.

II.1 Analyse :

II.1.1 Méthode d'analyse et de conception :

Avant de développer toute application il est utile de faire référence à une méthode bien définie, alors Pour mener bien notre travail, il nous faut d'abord passer par des étapes qui rendent sa réalisation plus précise et organisée. Pour cela on a opté pour le langage UML.

II.1.1.1 Présentation d'UML :

Définition UML :

UML signifie Unified Modeling Language qui est un formalisme ou langage de modélisation orienté objet qui représente un moyen de représenter et spécifier les composants d'un système informatique. Il permet la spécification des décisions, qui doivent être prises lors du développement d'un système à forte composante logicielle, en matière d'analyse, d'implémentation et conception.

II.1.1.2 Quelques définitions de base :

➤ Acteur :

C'est un rôle que peut jouer une personne ou une chose dans un système. L'acteur peut interagir avec un cas d'utilisation, et participer à son scénario. Il est donc dit un objet actif. Les utilisateurs de notre système sont :

- **Tâche** : une tâche est l'ensemble des fonctions qu'un acteur bien spécifié peut effectuer.
- **Scénario** : C'est une succession particulière d'enchaînement, s'exécutant du début à la fin du cas d'utilisation.

II.1.2 Démarche :

II.1.2.1 Identificateur des Acteurs :

➤ Administrateur :

L'administrateur a pour rôle la gestion de l'application, et sa tâche est la gestion des différents plans (moteurs, transformateurs) en les modifiant, en les supprimant ou bien en ajoutant d'autres plans.

➤ Gestionnaire plan moteur :

Sa tâche est la gestion des différents plans (moteurs) en les modifiant, en les supprimant ou bien en ajoutant d'autres plans.

➤ Gestionnaire plan transformateur :

Sa tâche est la gestion des différents plans (transformateur) en les modifiant, en les supprimant ou bien en ajoutant d'autres plans.

II.1.2.2 Les cas d'utilisation global :

Les cas d'utilisation global de l'administrateur :

- S'authentifier.
- Gérer les différents plans (moteurs, transformateurs).
- Changer le mot de passe.

Les cas d'utilisation global de Gestionnaire plan moteur :

- S'authentifier.
- Gérer les différents plans (moteurs).

Les cas d'utilisation global de Gestionnaire plan transformateur:

- S'authentifier.
- Gérer les différents plans (transformateurs).

II.1.2.3 Diagramme de cas d'utilisation global :

Les diagrammes de cas d'utilisation représentent un ensemble de cas d'utilisations, et d'acteurs et les différentes relations entre eux c'est la vue statique des cas d'utilisation d'un système.

- **La relation include :**

Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.

- **La relation extend :**

Une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B.

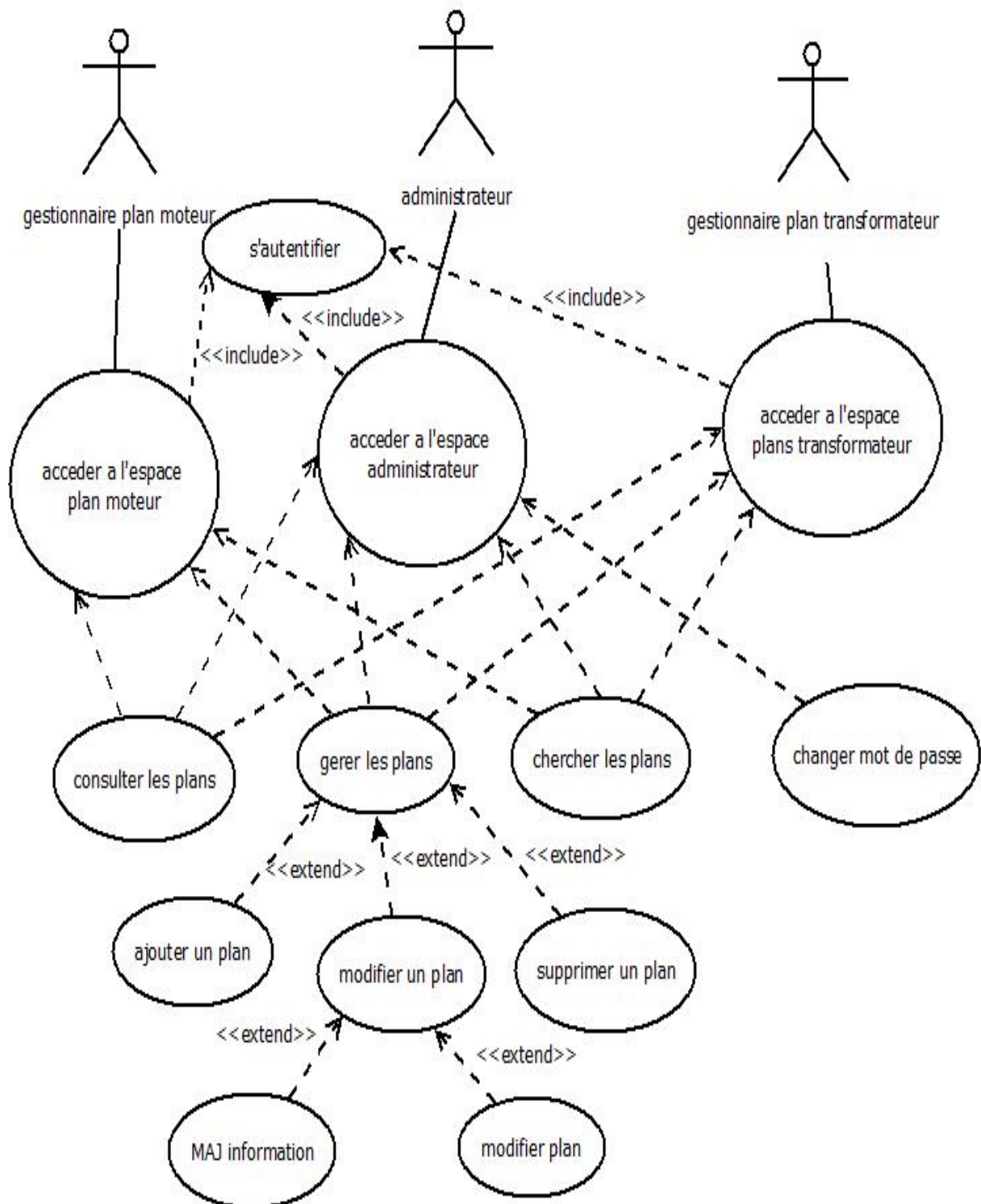


Figure III.1.2.1: Diagramme de cas d'utilisation global.
 Le diagramme ci dessus nous montre une vue globale de l'application.

II.1.2.4 Spécification des cas d'utilisation global :

Acteur	Cas d'utilisation global
Administrateur	C1 :S'authentifie. C2 : Gérer les plans moteurs. C3 : Gérer les plans transformateurs. C4 : Gérer la base de données. C5 : Changer le mot de passe.

Acteur	Cas d'utilisation global
Gestionnaire plan moteur	C1 :S'authentifie. C2 : Gérer les plans moteurs.

Acteur	Cas d'utilisation global
Gestionnaire plan transformateur	C1 :S'authentifie. C2 : Gérer les plans transformateurs.

II.1.2.5 Spécification des scénarios :

➤ **administrateur :**

Cas d'utilisation	Scenarios
C01 : Accéder a l'espace administrateur (s'authentifier) ;	S01: Saisir le login. S02 : Saisir le mot de passe et puis validé.
C02 : Gérer les plans moteurs.	S03 : Cliquer sur le menu <Plan > → nouveau plan/mais jour informations plan S04 : Ajouter un plan. S05 : modifier un plan. S06 : Supprimer un plan. S07 : Cliquer sur le menu<Plan remis > S08 : Remis plan. S09: restitute plan. S10 : Cliquer sur le menu<Modification > → modifier plan/plan modifié déjà S11 : modifier S12 : non modifier. S13 : éditer l'avis de modifications plan. S14 : Cliquer sur le menu<Visualisation > → Plan S15 : éditer la fiche plan. S16 : Cliquer sur le menu<PDF > → Uploadé S17 : uploader un plan.

<p>C03 : Gérer les plans transformateurs.</p>	<p>S18 : Cliquer sur le menu <Plan > → nouveau plan/mais jour informations plan S19 : Ajouter un plan. S20 : modifier un plan. S21 : Supprimer plan. S22 : Cliquer sur le menu<Plan remis > S23 : Remis plan. S24 : restituer plan. S25: Cliquer sur le menu<Modification > → modifier plan/plan modifié déjà S26 : modifier. S27 : non modifier. S28 : éditer l’avis de modifications plan. S29 : Cliquer sur le menu<Visualisation > → Plan S30 : éditer la fiche plan. S31 : Cliquer sur le menu<PDF > → Uploadé S32 : uploader un plan.</p>
<p>C04 : Gérer la base de données.</p>	<p>S33 : Cliquer sur le menu<Plan > → mais jour informations plan S34 : Visualiser la liste des plans. S35 : Modifier/supprimer un plan.</p>
<p>C05 : Changer le mot de passe.</p>	<p>S36 : Cliquer sur le menu< mot de passe> → modifié le mot de passe S37 : modifier S38 : Remplir le formulaire de changement de mot de passe; S39 : Valider le changement.</p>

➤ **Gestionnaire plan moteur :**

Cas d'utilisation	Scenarios
C01 : Accéder a l'espace gestionnaire plan moteur (s'authentifier) ;	S40: Saisir le login. S41 : Saisir le mot de passe et puis validé.
C02 : Gérer les plans moteurs.	S03 : Cliquer sur le menu <Plan > → nouveau plan/mais jour informations plan S04 : Ajouter un plan. S05 : modifier un plan. S06 : Supprimer plan. S07 : Cliquer sur le menu<Plan remis > S08: Remis plan. S09: restitute plan. S10 : Cliquer sur le menu<Modification > → modifier plan/plan modifié déjà S11 : modifier S12 : non modifier. S13 : éditer l'avis de modifications plan. S14 : Cliquer sur le menu<Visualisation > → Plan S15 : éditer la fiche plan. S16 : Cliquer sur le menu<PDF > →Uploadé S17 : uploader un plan.

Tableau: Spécification des scénarios Gestionnaire plan moteur.

➤ **Gestionnaire plan transformateur :**

Cas d'utilisation	Scenarios
C01 : Accéder a l'espace gestionnaire plan transformateur (s'authentifier) ;	S42: Saisir le login. S43 : Saisir le mot de passe et puis validé.
C02 : Gérer les plans transformateurs.	S18 : Cliquer sur le menu <Plan > → nouveau plan/mais jour informations plan S19 : Ajouter un plan. S20 : modifier un plan. S21 : Supprimer un plan. S22 : Cliquer sur le menu<Plan remis > S23: Remis plan. S24 : restituteur plan. S25 : Cliquer sur le menu<Modification > → modifier plan/plan modifié déjà S26 : modifier. S27 : non modifier. S28 : éditer l'avis de modifications plan. S29 : Cliquer sur le menu<Visualisation > → Plan S30 : éditer la fiche plan. S31 : Cliquer sur le menu<PDF > →Uploadé S32 : uploader un plan.

Tableau: Spécification des scénarios Gestionnaire plan transformateur.

II.1.2.6 Description textuelle des cas d'utilisation :

Après avoir identifiées les différents cas d'utilisations, nous allons maintenant dresser un descriptif textuel. Pour chaque cas utilisation on met en avant les acteurs qui l'active, les scénarios contenus et une description des actions à réaliser.

<p>Use case: S'authentifier.</p> <p>Scénarios: S01, S02.</p> <p>Rôle: Administrateur.</p> <p>Description :</p> <ol style="list-style-type: none">1. L'administrateur clique sur l'exécutable;2. Le système lui affiche le formulaire d'authentification ;3. L'administrateur saisit son login et son mot de passe puis valide ;

Figure -8-: Cas d'utilisation S'Authentifier.

<p>Use case: Ajouter un plan.</p> <p>Scénarios: S03, S04.</p> <p>Rôle : Administrateur, Gestionnaire plan moteur.</p> <p>Description :</p> <ol style="list-style-type: none">1. L'administrateur clique sur le menu <Plan > → nouveau plan;2. Le système lui affiche le formulaire à remplir ;3. L'administrateur saisit le formulaire et le valide ;4. Le système confirme l'insertion de plan dans la base de données ou renvoie un message
--

Figure -9-: Cas d'utilisation Ajouter un plan moteur.

Use case: modifier information plan .

Scénarios: S03, S05.

Rôle : Administrateur ,Gestionnaire plan moteur.

Description :

1. L'administrateur clique sur le menu <Plan > → mais jour plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien modifier;
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide ;

Figure -10-: Cas d'utilisation modifier informations plan moteur.

Use case: Supprimer un plan.

Scénarios: S03, S06.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur clique sur le menu <Plan > → mais jour plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien supprimer;
4. Le système confirme la suppression de client de la base de données.

Figure -11-: Cas d'utilisation supprimer un plan moteur.

Use case: Plan remis.

Scénarios: S07, S08.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur clique sur le menu<Plan remis >;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien remis ;
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide .

Figure -12-: Cas d'utilisation d'un plan remis moteur .

Use case: Plan restitué.

Scénarios: S07, S09.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur clique sur le menu<Plan remis >;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien restituteur ;
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide .

Figure -13-: Cas d'utilisation d'un plan restitué moteur.

Use case: Plan modifier.

Scénarios: S10, S11.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur Cliquer sur le menu<Modification > → modifier plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien modifié.
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide .

Figure -14-: Cas d'utilisation d'un plan modifié moteur.

Use case: Plan non modifier.

Scénarios: S10, S12.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur Cliquer sur le menu<Modification > → plan modifié déjà;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien non modifié.

Figure -15-: Cas d'utilisation d'un plan non modifié moteur.

Use case: éditer l'avis de modifications plan.

Scénarios: S10, S13.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur Cliquer sur le menu<Modification > → plan modifié déjà;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien numéro ;
4. Le système lui affiche l'avis de modifications plan.

Figure -16-: Cas d'utilisation éditer l'avis de modifications plan moteur.

Use case: éditer la fiche plan.

Scénarios: S14, S15.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur Cliquer sur le menu<Visualisation > → Plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien fiche plan ;
4. Le système lui affiche une fiche plane.

Figure -17-: Cas d'utilisation éditer la fiche plan moteur.

Use case: Uploader plan.

Scénarios: S16, S17.

Rôle : Administrateur, Gestionnaire plan moteur.

Description :

1. L'administrateur Cliquer sur le menu<PDF > → Uploadé;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien uploader un plan;
4. Le système lui affiche une fiche plan ;
5. Le système lui affiche le formulaire de téléchargement plan ;
6. L'administrateur clique sur le bottent uploader.

Figure -18- : Cas d'utilisation uploader un plan moteur.

Use case: Ajouter un plan.

Scénarios: S18, S19.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur clique sur le menu <Plan > → nouveau plan;
2. Le système lui affiche le formulaire à remplir ;
3. L'administrateur saisit le formulaire et le valide ;
4. Le système confirme l'insertion de plan dans la base de données ou renvoie un message

Figure -19-: Cas d'utilisation Ajouter un plan transformateur.

Use case: modifier information plan.

Scénarios: S18, S20.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur clique sur le menu <Plan > → mais jour plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien modifier;
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide ;

Figure -20-: Cas d'utilisation modifier informations plan transformateur.

Use case: Supprimer un plan.

Scénarios: S18, S21.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur clique sur le menu <Plan > → mais jour plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien supprimer;
4. Le système confirme la suppression de client de la base de données.

Figure -21-: Cas d'utilisation supprimer un plan transformateur.

Use case: Plan remis.

Scénarios: S22, S23.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur clique sur le menu<Plan remis >;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien remis ;
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide.

Figure -22-: Cas d'utilisation d'un plan remis transformateur.

Use case: Plan restitué.

Scénarios: S22, S24.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur clique sur le menu<Plan remis >;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien restituteur ;
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide.

Figure -23: Cas d'utilisation d'un plan restitué transformateur.

Use case: Plan modifier.

Scénarios: S25, S26.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur Cliquer sur le menu<Modification > → modifier plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien modifié.
4. Le système lui affiche le formulaire à remplir ;
5. L'administrateur saisit le formulaire et le valide.

Figure -24-: Cas d'utilisation d'un plan modifié transformateur.

Use case: Plan non modifier.

Scénarios: S25, S27.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur Cliquer sur le menu<Modification > → plan modifié déjà;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien non modifié.

Figure -25-: Cas d'utilisation d'un plan non modifié transformateur.

Use case: éditer l'avis de modifications plan.

Scénarios: S25, S28.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur Cliquer sur le menu<Modification > → plan modifié déjà;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien numéro ;
4. Le système lui affiche l'avis de modifications plan.

Figure -26-: Cas d'utilisation éditer l'avis de modifications plan transformateur.

Use case: éditer la fiche plan.

Scénarios: S29, S30.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur Cliquer sur le menu<Visualisation > → Plan ;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien fiche plan ;
4. Le système lui affiche une fiche plan.

Figure -27-: Cas d'utilisation éditer la fiche plan transformateur.

Use case: Uploader plan.

Scénarios: S31, S32.

Rôle : Administrateur, Gestionnaire plan transformateur.

Description :

1. L'administrateur Cliquer sur le menu<PDF >→Uploadé;
2. Le système lui affiche une liste des plans chacun avec son numéro ;
3. L'administrateur clique sur le lien uploader un plan;
4. Le système lui affiche une fiche plan ;
5. Le système lui affiche le formulaire de téléchargement plan ;
6. L'administrateur clique sur le bottent uploader.

Figure -28- : Cas d'utilisation uploader un plan transformateur.

Use case: Changer le mot de passe.

Scénarios: S36, S37, S38, S39.

Rôle : Administrateur.

Description :

1. L'administrateur clique Cliquer sur le menu< mot de passe> → modifié le mot de passe ;
2. Le système lui affiche un tableau contient l'ancien mot de passe et lien modifier ;
3. L'administrateur clique sur le lien modifier;
4. Le système lui affiche un formulaire qui contient un champ ancien mot de passe ;
5. L'administrateur remplit ce formulaire et le valide.

Figure -29-: Cas d'utilisation Changer le mot de passe.

II.2 Conception :

Dans cette phase une nouvelle vue du modèle fait son apparition. Cette vue exprime les modules et les exécutables physiques sans aller à la réalisation concrète du système. Elle est basée sur :

Les diagrammes de séquence : ils présentent la vue dynamique du système. L'objectif du diagramme de séquence est de représenter les interactions entre les objets en indiquant la chronologie des échanges. Cette représentation se réalise par cas d'utilisation.

Les diagrammes de classes : représentent la vue statique des objets pages, leur intérêt majeur est de modéliser les entités d'un système. Autrement dit, ils expriment les relations existantes entre les pages client et serveur. Dans ce qui suit on représentera quelques diagrammes de séquence et de classe globale, correspondant aux cas d'utilisation déjà décrits.

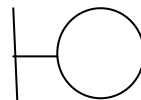
II.2.1. Elaboration des diagrammes de séquences :

Il illustre la dynamique d'enchaînement des traitements d'une application effectuée par le système, en mettant en évidence la chronologie d'envoi de messages entre les objets coopératifs. Ces traitements sont ordonnés dans le temps et traduisant ainsi, une série d'événements entre les différents objets du système, ce qui nous permet de déterminer de manière précise, le rôle de chacun d'eux vis-à-vis des autres.

Ce type de diagrammes a le principal intérêt d'illustrer les cas d'utilisation. Les objets utilisés, pour représenter les diagrammes de séquence dans la conception de notre application, sont les suivants :

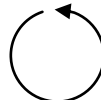
- **Objets d'interface** : Ils représentent l'interface entre l'acteur et le système. Ils sont représentés comme suit:

L'icône :



- **Objets contrôles** : ils représentent les traitements effectués par le système. Ces objets sont représentés comme suit:

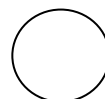
L'icône :



- **Objets Entité** : Ils représentent les tables des bases de données.

Nous donnons ci-dessous, un diagramme de séquence pour un cas d'utilisation, pour chaque acteur interagissant avec le système :

L'icône :



Dans les diagrammes de classes les associations entre les objets doivent respecter un certain nombre de règles qui sont les suivantes :

1. les acteurs n'interagissent qu'avec des objets d'interface.
2. les objets entité interagissent qu'avec les objets contrôle.
3. les objets contrôles interagissent avec tous les objets, y compris des instances d'objets contrôle.

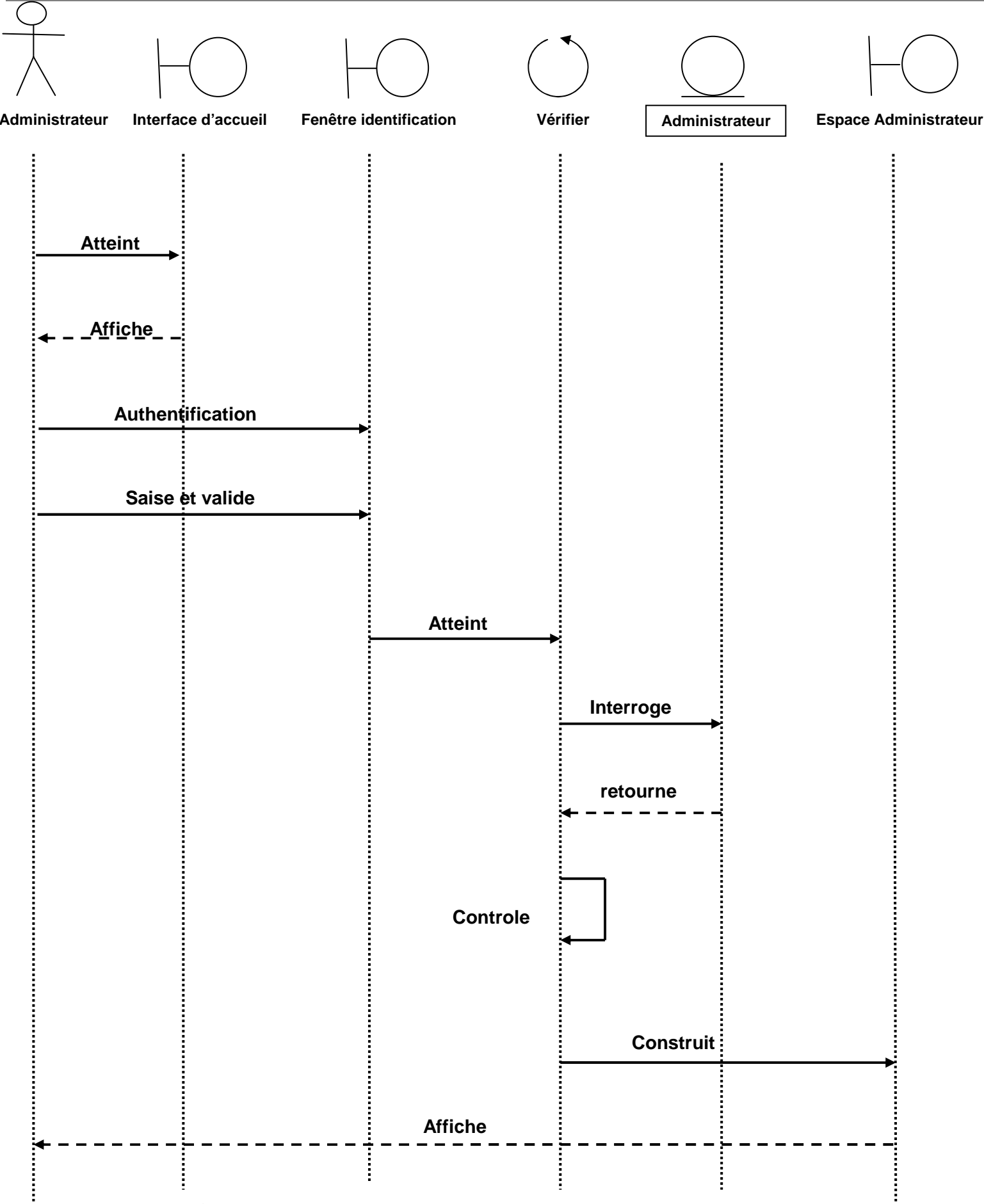


Figure III.2.1: Diagramme de séquence du cas d'utilisation S'authentifier.

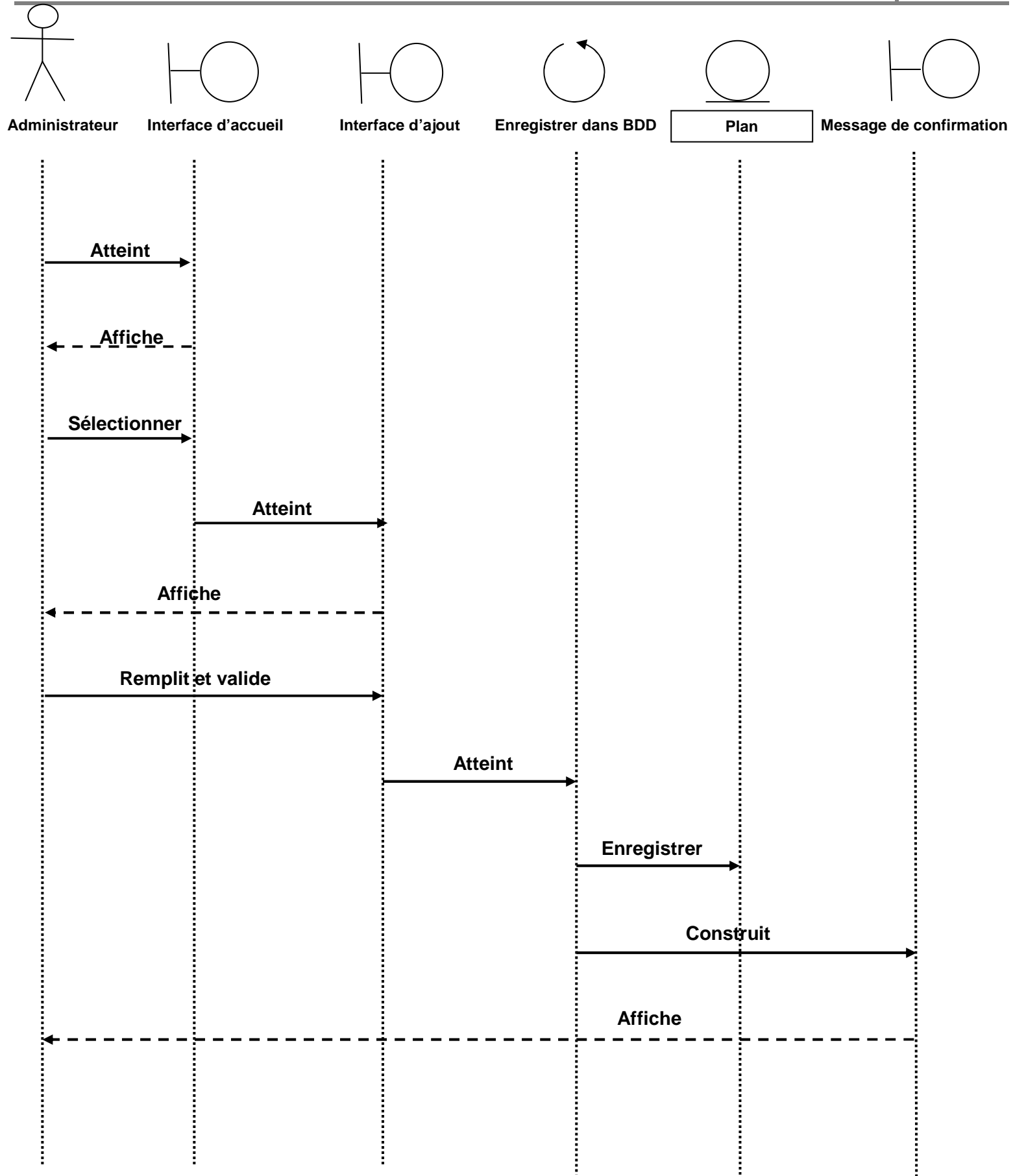


Figure III.2.2 Diagramme de séquence du cas d'utilisation ajouter un plan.

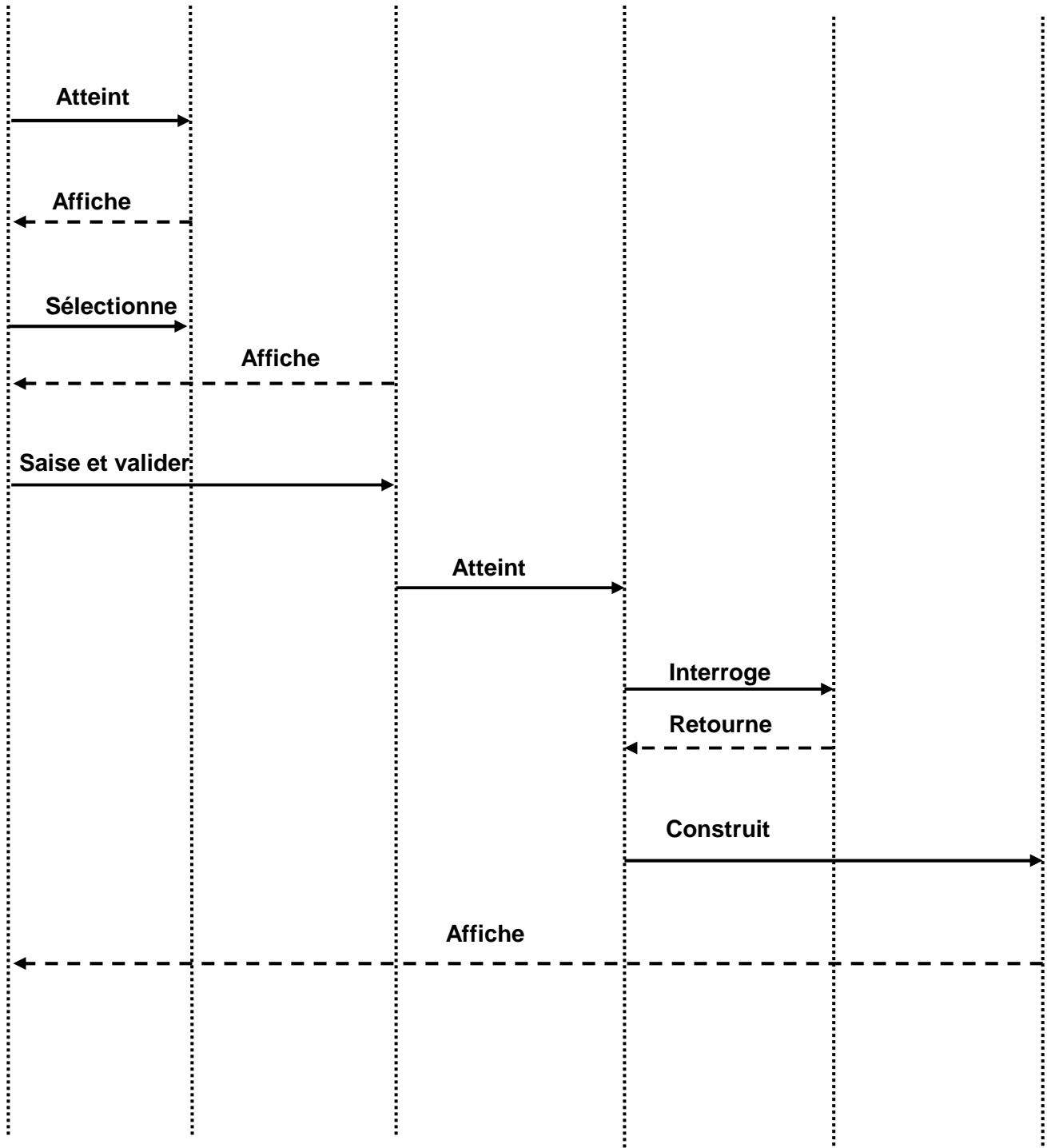
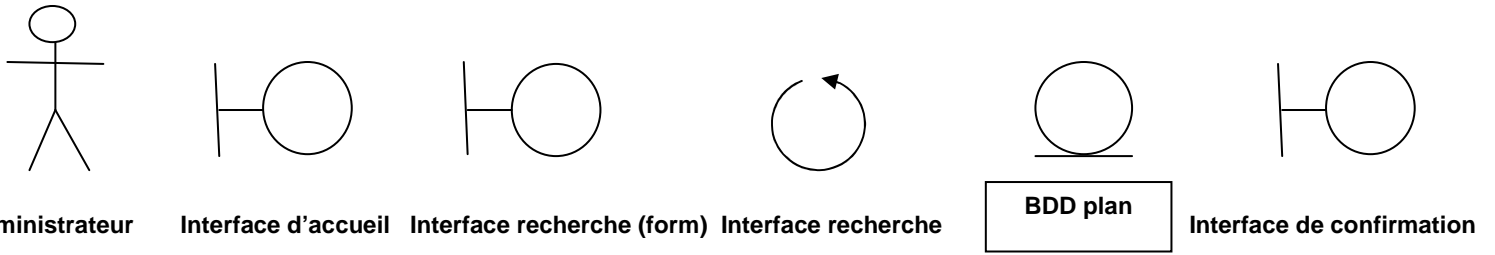


Figure III.2.3 : Diagramme de séquence du cas d'utilisation Consulter/Rechercher les plans.

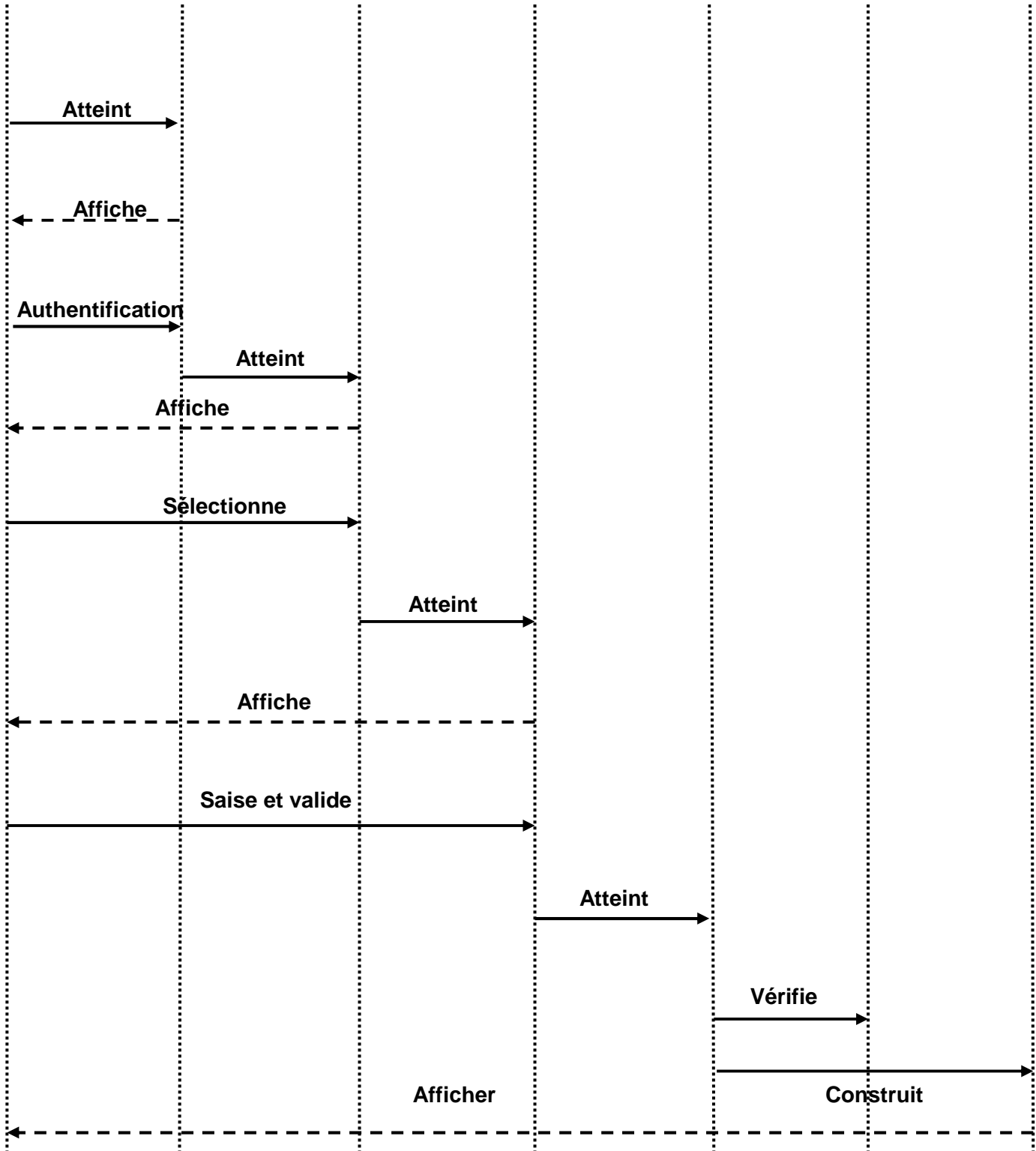
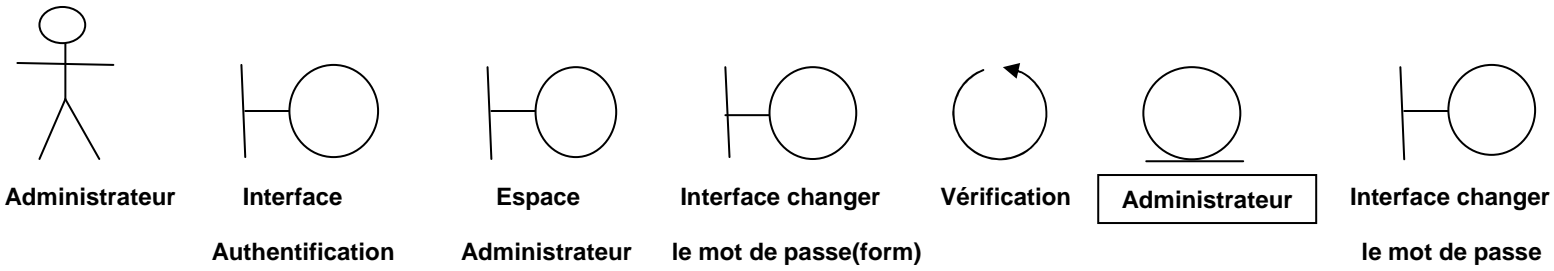


Figure III.2.1: Diagramme de séquence du cas d'utilisation changer le mot de passe.

II.2.3 Conception de la base de données :

Dans cette partie du chapitre on va décrire en premier lieu, la manière de concevoir la structure de la base de données a travers un modèle conceptuel : le diagramme de classe (classe entité). Ensuite, nous allons traduire ce dernier en modèle relationnel qui optimise l’implantation de la base de données.

II.2.3.1 Le diagramme de classe globale du schéma conceptuel :

Un diagramme de classe représente l’ensemble des classes d’objets en collaboration, ainsi que les relations qui lient ces objets. Il représente la description statique du système. Il intègre dans chaque classe une partie dédiée aux données, et une autre haut traitement.

Pour le cas de notre application, le diagramme de classe est donné ci-dessous.

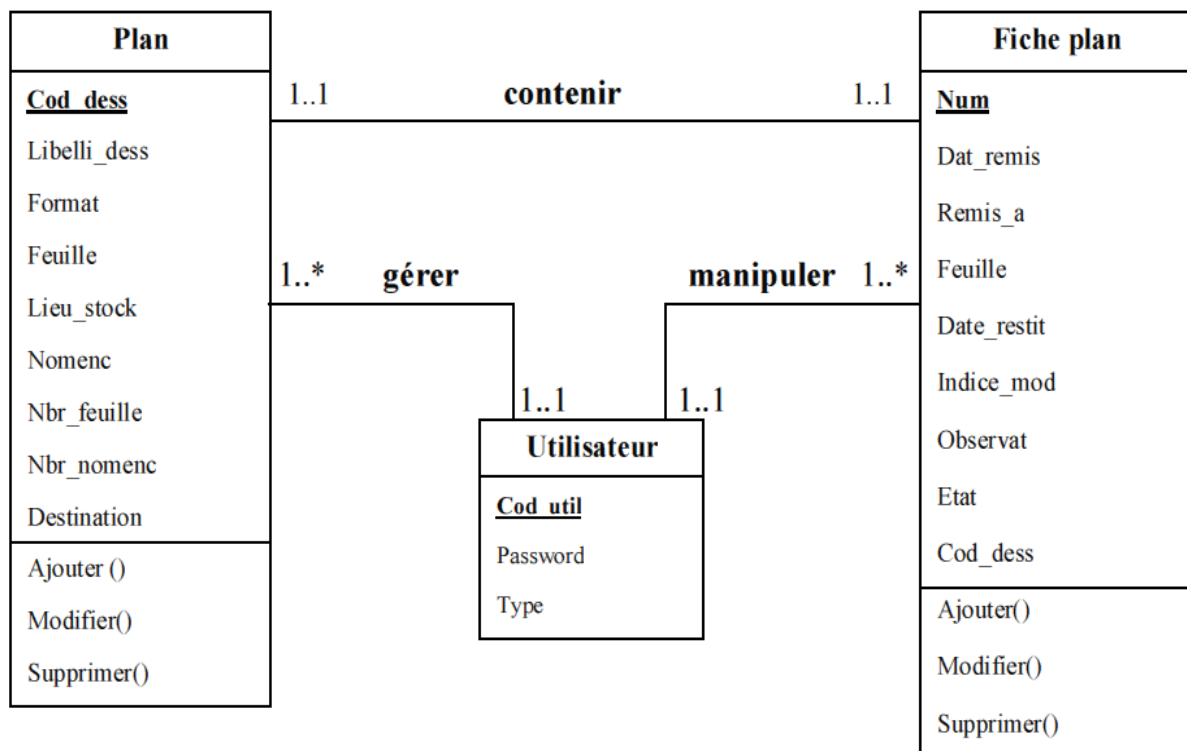


Figure-39- : diagramme de classe globale.

II.2.3.2 Le modèle conceptuel physique de base de donnée

Après avoir établi le diagramme de classe pour notre application, nous procédons à la représentation physique de nos classes d'objets persistants en s'appuyant sur le modèle relationnel de données.

Le Modèle relationnel :

Le modèle relationnel présente l'univers des données décrit dans le modèle entité/association en tenant compte du type de base de données choisi : relationnelle ou navigationnelle (réseau, hiérarchique), autrement dit, le modèle relationnel traduit le modèle entité/association en un formalisme compréhensible par la machine.

i) Passage au modèle relationnel [GRI. 99]

Pour obtenir un schéma relationnel à partir d'un diagramme de classe, il faut commencer par :

- Traduire les classes en relation.
- Traduire les associations en relations. celle qui ont un bout avec une multiplicité maximum inférieur égal à 1 (appelons A la classe qui est du cote de cette multiplicité) peuvent être traduites en ajoutant un attribut la clé primaire de la classe A dans l'autre classe.
- Traduire les autres types d'association, cette relation à une clé primaire composée des clés des relations représentant les classes entrant dans l'association. Les autres attributs de cette relation sont les attributs de la classe liée à cette association (si elle existe).
- La traduction de l'héritage se fait en partageant la même clé entre les deux classes. La clé primaire de la classe fille est déclarée comme une clé étrangère correspondant à la clé primaire de la classe mère...

ii) Schéma relationnel :

Plan (N°-plan, lieu stockage, nombre feuilles, feuille, format, désignation, nomenclature, nombre nomenclature).

Administrateur (Mot-passe, login).

Gestionnaire plan moteur (Mot-passe, login).

Gestionnaire plan transformateur (Mot-passe, login).

Remarque : Dans le schéma de relation on prit les conventions d'écriture suivant :

- les clés primaires sont notées en caractères soulignés.
- les clés étrangères sont suivis d'une étoile (*).

Conclusion :

Dans ce chapitre, nous nous sommes concentrés sur les aspects analytique et conceptuel de notre application ainsi que les Bases de données qui interagissent avec elle.

Pour la phase analyse, nous avons défini les différents cas d'utilisations puis, nous les avons traduit au travers la construction des diagrammes de séquence et de classe, dans la phase de conception.

Enfin, nous avons défini toutes les tables des bases de données utilisées.

Le chapitre suivant sera consacré à l'implémentation de notre application.

Introduction :

Après avoir abordé dans le chapitre précédent la conception de notre application, nous allons présenter dans ce chapitre la création et la mise en œuvre des différents programmes, les principales interfaces via lesquelles les utilisateurs (gestionnaire plan transformateur, gestionnaire plan moteur, administrateur) interagissent avec le système et sa base de données, qui servent à la constitution de notre application et assurent ses fonctionnalités.

I.1. Outils de développement :

. L'interface *PHP* MyAdmin :

L'outil PhpMyAdmin a été développé en PHP (ensemble de scripts PHP), il offre une interface graphique pour l'administration des bases de données MySQL via un navigateur Web.

Les fonctions principales de phpMyAdmin sont :

- Création de nouvelles bases de données.
- Création/suppression/modification des tables.
- L'édition, l'ajout et la suppression de champs.
- L'exécution de commandes SQL et de requêtes.

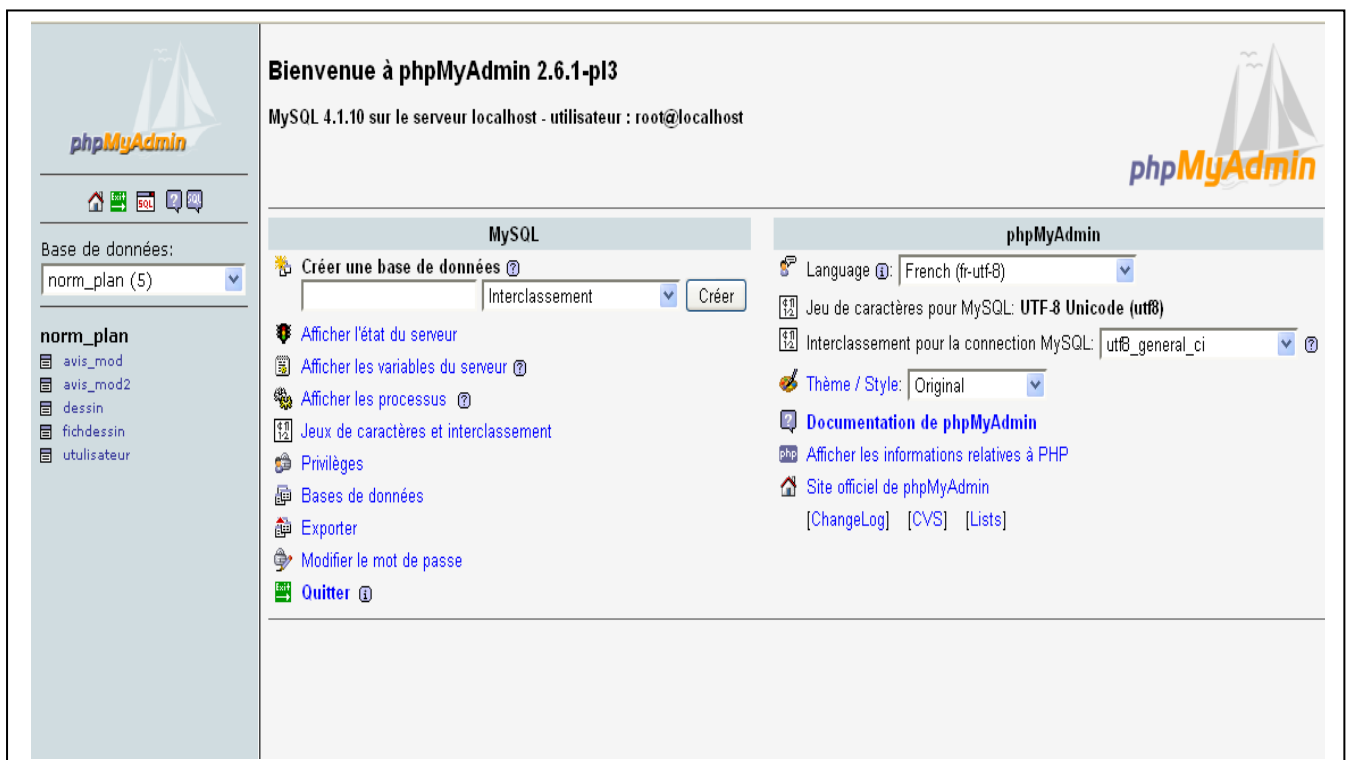


Figure -40- : interface PHPMyAdmin.

I.2. Les serveurs :

I. 2.1. Serveur Apache :

Le développement d'un site web nécessite un serveur web qui s'occupe du traitement des requêtes clients, le transfert des pages HTML au navigateur web et l'exécution des programmes sur la machine serveur, apache est l'un des serveurs web le plus réponsus sur internet, c'est une application fonctionnant a la base sur le système d'exploitation Unix et ensuite porté sur d'autre système dont Microsoft Windows, apache possède de nombreuse fonctionnalité dont la possibilité de définir une configuration spécifique a chaque fichier ou répertoire partager. nous avons utilisé la version 5.0.27 PHP 2.9.2.

I. 2.2. Serveur MySQL :

MySQL est un système de base de données relationnelles robuste, basé sur le langage de requête SQL (Structured Query Language) qui est un langage standard pour le traitement des bases de données. Le serveur MySQL peut fonctionner en mode client/serveur.

Il contrôle l'accès aux données pour assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données, pour y accéder rapidement et pour garantir que seuls les utilisateurs autorisés peuvent accéder aux données.

Le serveur MySQL offre des fonctions nombreuses et puissantes. Ses possibilités de connexions, sa rapidité et sa sécurité font de lui un serveur hautement adapté à Internet. Nous avons utilisé la version MySQL 5.0.22.

I.3. Logiciel Dreamweaver :

Macromédia Dreamweaver est un éditeur html professionnel destiné à la conception, au codage et au développement de sites, des pages et d'applications web (voir figure -41-).

Il constitue aujourd'hui une plate forme de développement d'applications Internet de haut niveau. Vu les multiples possibilités qu'il offre, nous l'avons utilisé pour créer nos interfaces.

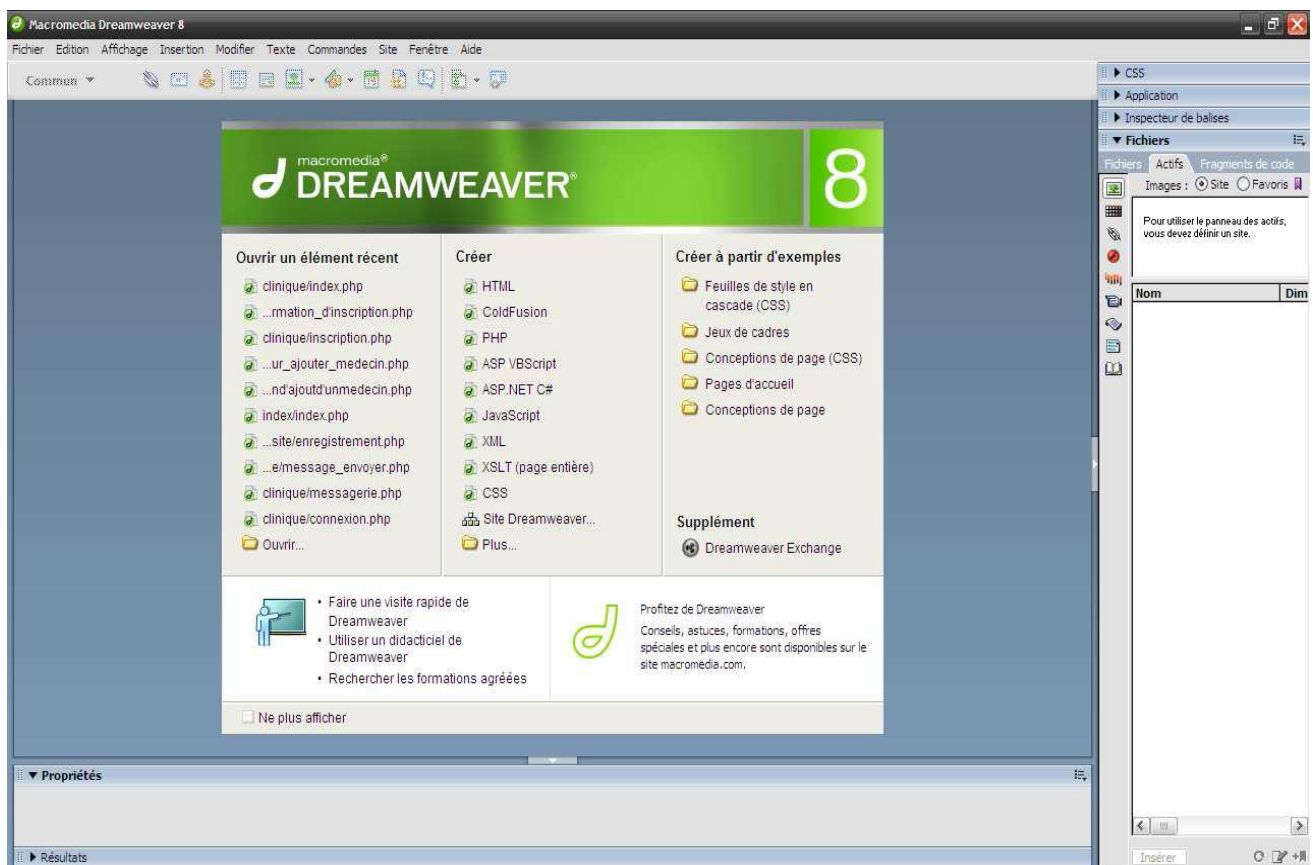


Figure -41- : Interface du logiciel Dreamweaver.

I.4. Les langages de programmation utilisés :

1. HTML (HyperText Mark-up language) :

C'est un langage de **marquage** (de **structuration** ou de **balisage**) dont le rôle est de créer des pages web utilisant un serveur web et aussi formaliser l'écriture d'un document avec des balises. Les balises permettent d'indiquer la façon dont doit être présenté un document et les liens qu'il doit établir avec d'autre document.

Le langage **HTML** permet notamment la lecture des documents sur internet a partir de différentes machines, grâce au protocole **http**, permettant d'accéder via les réseaux a des documents repérés par une adresse unique appelée **URL**.

2. Le langage PHP(Pre HyperTextProcessor):

Le langage **PHP** a été mis au point en **1994** par **Rasmus Lerdorf**. Ce langage de script lui permettait de conserver la trace des utilisateurs venant consulter son CV sur son site, grâce à l'accès à une base de données par l'intermédiaire de requêtes **SQL**. Ainsi étant donnée que nombreux internautes lui demandèrent ce programme qu'il baptisa Personal Sommaire Page Tools, puis Personal Home Page v1.0 (traduisez page personnelle version 1.0).

Le langage **PHP** est l'un des langages le plus répondu et utilisé pour la programmation de sites web. C'est l'un des langages utilisés pour l'extraction et le traitement des données d'une base de données **MySQL**. Une fois les données traitées, elles peuvent être renvoyées sur navigateur d'un utilisateur ou bien réinsérées dans la base **MySQL**.

PHP est un langage interprété (un langage de script) exécuté du côté serveur (comme les scripts **CGI**, **ASP**, ...) et non du côté client (un script écrit en JavaScript ou une applet Java s'exécute sur votre ordinateur...) La syntaxe de langage provient de celles du langage **C**, du **Perl** et de **Java**.

3. SQL (Structured Query Language):

C'est un langage de manipulation de base de données mise au point dans les années 70 par IBM, il permet notamment :

- La manipulation des bases de données (création et suppression).
- La manipulation des tables.
- La gestion des droits d'accès aux tables : contrôle des données et validation des Modifications.

4. JavaScript :

Le JavaScript est le premier langage de script pour le web développé par Netscape en 1995, incorporer dans un document HTML, ce langage est un langage de programmation qui permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes du côté client c'est-à-dire au niveau de navigateur et non au côté serveur web, JavaScript permet d'effectuer des contrôles de saisie pour valider les formulaires, ouvrir ou fermer des fenêtres ou encore de gérer les éléments graphiques.

I.5. Le modèle physique de données :

Le modèle physique de données est la traduction du modèle logique de données dans un langage de description de données spécifique au système de gestion des bases de données utilisé. Le modèle physique des données est l'implantation des données (tables) issues du modèle logique des données en machine afin d'aboutir à la description des fichiers de base de données.

• **Table plan :**

Nom du champ	Type de donnée	Taille	Description	Observation
Cod_dess	varchar	12	Identifiant plan	Clé primaire
Libelli_dess	varchar	150	désignation plan	
format	char	2	format du plan	
feuille	varchar	10	feuille du plan	
lieustock	varchar	10	lieu du plan	
Nomenc	varchar	10	nomenclateur plan	
nbrfeuille	int	1	nombre feuille plan	
Type	varchar	10	type du plan	
destination	varchar	100	destination plan	

• **Table utilisateur :**

Nom de champ	Type de donnée	Taille	Description	Observation
Cod_util	int	3	Code utilisateur	Clé primaire
Type	varchar	20	type utilisateur	
passwd	varchar	10	Mot du passe utilisateur	

• **Table fiche plan :**

Nom du champ	Type de donnée	Taille	Description	Observation
num	int	6	numéro fiche plan	Clé primaire
etat	int	1	etat plan	
dat_remis	dat	30	date remis plan	
remis_a	varchar	100	remis plan	
feuille	varchar	10	nombre feuille plan	
dat_restit	dat	30	date restitution plan	
indic_mod	char	1	indic modification plan	
observat	varchar	100	observation plan	
cod_dess	varchar	12	identifiant plan	Clé étrangère

II. Les interfaces de notre application :

Coté administrateur :

- **Interface d'authentification :**

Dans cette interface l'administrateur saisit son login et son mot de passe pour accéder à son espace. De cette manière on contrôle l'accès à cette Interface.



Figure -42-: Interface d'authentification.

- Interface « Ajouter plan » :

Dans cette interface l'administrateur peut ajouter les plans (moteurs ou transformateurs) de l'entreprise.

The screenshot displays a web application interface for adding a new plan. At the top, there is a navigation menu with the following items: 'accueil', 'Plan', 'Remis Plan', 'Modification', 'Visualisation', 'Mot_du_passe', and 'PDF'. The main content area features a central form titled 'Ajouter Nouveau Plan'. The form is organized into two columns of input fields. The left column includes: 'N° de plan : *' (with a red asterisk indicating a required field), 'Lien de stockage :', 'Nombre de Feuilles :', 'Section Destinataires :', and 'Format :'. The right column includes: 'Désignation : *' (with a red asterisk), 'Feuille :', 'Nomenclateurs :', and 'Nombre de nomenclateurs :'. Below the input fields, there is a legend '* Champs Obligatoire' and two buttons: 'Ajouter Plan' and 'Effacer'. The background of the application shows a banner for 'EPE-ELECTRO-INDUSTRIES SPA' and 'Moteurs' and 'que-Trans' and 's Électroge'. At the bottom of the page, there is a footer with the text 'unité administrateur' and a 'QUITER' button with a red dot.

Figure -43-: Interface Ajouter plan.

- **Interface ‘Chercher un plan’ :**

Dans cette interface l’administrateur peut faire une recherche sur les plans (moteurs ou transformateurs) de l’entreprise.

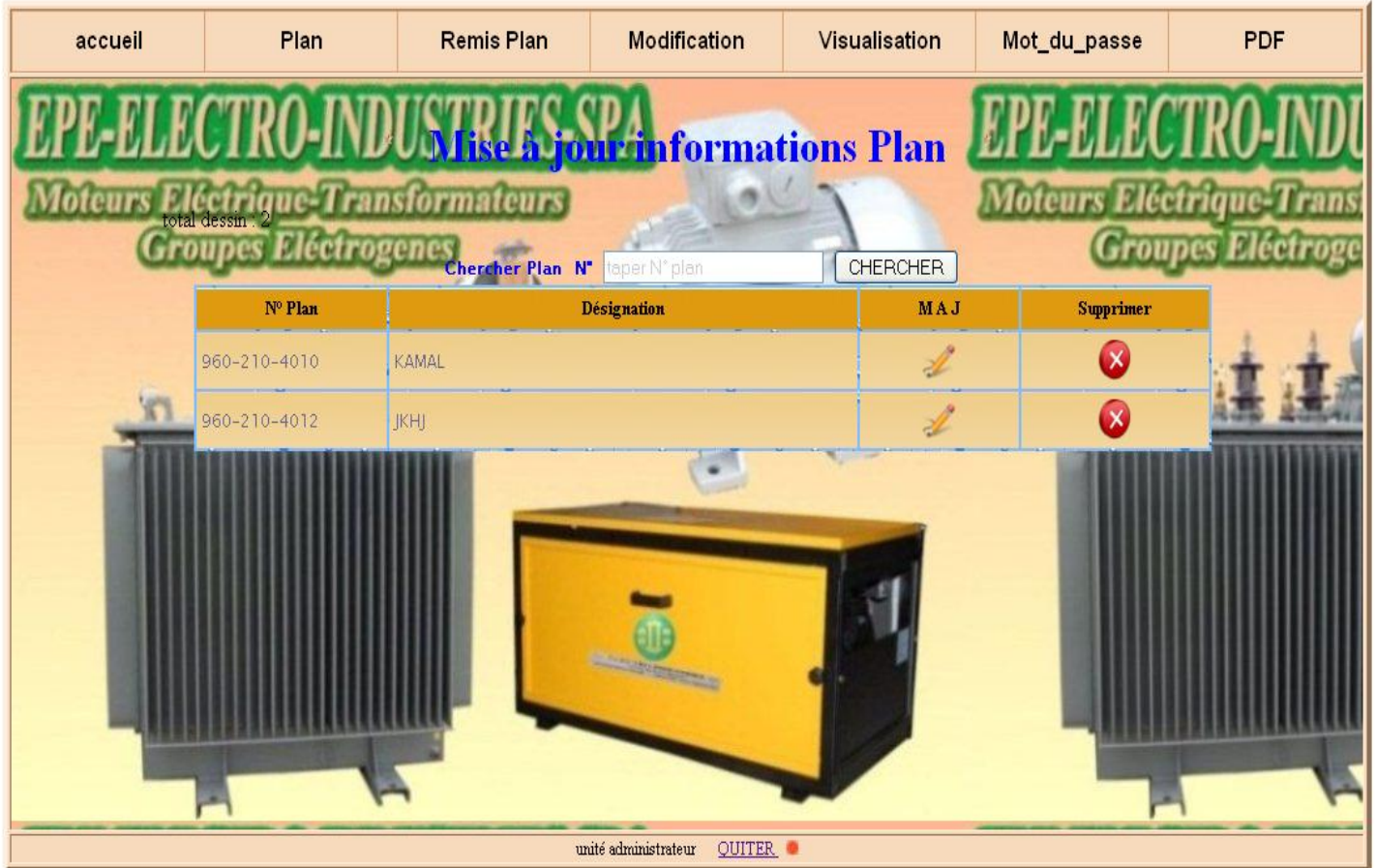


Figure -44-: Interface Rechercher plan.

- Interface ‘modifier un plan’ :

Dans cette interface l’administrateur peut faire une modification sur les plans (moteurs ou transformateurs) de l’entreprise.

The screenshot displays a web application interface for modifying plan information. At the top, there is a navigation menu with the following items: accueil, Plan, Remis Plan, Modification, Visualisation, Mot_du_passe, and PDF. The main content area features a banner for 'EPE-ELECTRO-INDUSTRIES SPA' with the text 'Moteurs' and 'que-Trans' visible. A central modal window titled 'Modification Informations Plan' contains the following form fields:

N° de Plan :	960-210-4010	Feuille :	4
Designation :	KAMAL	Nbr_feuille:	4
Lieu de stockage :	ak1	Nbr_nomenclatures:	3
Format :	A2	Sections_Destinataires:	tizi
Nomenclatures:	freha		

A 'modifier' button is located at the bottom center of the modal window. At the bottom of the page, there is a footer with the text 'unité administrateur' and a 'QUITER' button with a red dot icon.

Figure -45-: Interface modifié plan.

- **Interface ‘visualisation plan’ :**

Dans cette interface l’administrateur peut visualise et éditer les fiches plans (moteur ou transformateur) de l’entreprise.

The screenshot shows a web interface for 'EPE-ELECTRO-INDUSTRIES SPA' with the title 'Schémas de PLAN'. At the top, there is a navigation menu with links: accueil, Plan, Remis Plan, Modification, Visualisation, Mot_du_passe, and PDF. Below the menu is a search bar with the text 'Chercher Plan N°' and a text input field containing ' taper N° dessin', followed by a 'CHERCHER' button. A table displays the following data:

N° Plan	Désignation	Fiche plan	Schéma plan / Format
960-210-4010	KAMAL	fiche plan	NON Upload
960-210-4012	JKHJ	fiche plan	NON Upload

Below the table, there are images of electrical equipment: two large grey transformers on the left and right, and a yellow and black control cabinet in the center. At the bottom of the interface, there is a footer with the text 'unité administrateur' and a 'QUITER' button with a red dot.

Figure -46-: Interface visualisation plan.

- **Interface « Modifier mot du passe » :**

Ici l'administrateur peut modifier son mot du passe ou bien le mot du passe de (Gestionnaire plans moteur ou Gestionnaire plan transformateur) dans la base de données.

total de mot du passe : 3

pws	unité	modifier
123	LMP	
1234	LTR	
12345	Admin	

unité administrateur [QUITER](#)

Figure -47-: Interface Modifier mot du passe.

Conclusion :

Dans ce chapitre nous avons, au premier lieu, présenté les différents outils et langages que nous avons utilisé pour implémenter notre application. Par la suite, nous avons présenté quelques interfaces de notre application.

Livre et cours :

[01] The PostgreSQL Global Development Group.

[02] Chantal Morley & Jean Hugues & Bernard Leblanc, UML pour l'analyse d'un système.

[GEOR 96] : Le client Serveur par George et Olivier GARDARIN.

Edition Eyrolles [1996].

[NICO 98] : JAVA Client-serveur par Cédric NICOLAS, Frédéric NAJMAN Christophe

AVARE Edition Eyrolles [1998].

[ENSI 00] : Applications cobol sur le Web, avec une étude de cas détaillée par Roger ENGEL,

Patrice SILVERIO (Eyrolles) [2000].

[ENGR 00] : application cobol sur le Web par roger angel, patrice silverio edition eyrolles

[2000]

[GRI. 99] : Richard GRIN, "Introduction aux bases de données, modèle relationnel".

Université de Nice, janvier 1999

Site Internet :

[03] <http://www.siteduzero.com>.

[04] <http://www.commentcamarche.com>.

[05] <http://www.developpez.com>.

[06] <http://www.lexique-immobilier.com>.

[07] <http://fr.wikipedia.org>.