

Liste des figures

<u>Figure 1</u> : Présence mondiale.....	11
<u>Figure 2</u> : Schéma d'organisation d'Atos Origin au Maroc.....	14
<u>Figure 3</u> : Liste des partenaires d'Atos Origin	15
<u>Figure 4</u> : Vue générale des modules du BSCS.....	18
<u>Figure 5</u> : Type et paquet de service.....	18
<u>Figure 6</u> : Eléments nécessaires pour la création d'un service.....	20
<u>Figure 7</u> : Processus de la transformation d'un script SQL en une offre commerciale.....	23
<u>Figure 8</u> : Déploiement de l'application web.....	24
<u>Figure 9</u> : Cycle de vie du projet web.....	29
<u>Figure 10</u> : Architecture Client/serveur.....	33
<u>Figure 11</u> : Architecture 3-tiers d'une application Web.....	34
<u>Figure 12</u> : Modèle MVC.....	36
<u>Figure 13</u> : Navigation.....	38
<u>Figure 14</u> : Diagramme de déploiement de l'application	39
<u>Figure 15</u> : Diagramme de cas d'utilisation de l'application WEB.....	41
<u>Figure 16</u> : Diagramme de séquence de l'authentification dans l'application web.....	42
<u>Figure 17</u> : Architecture J2EE.....	47
<u>Figure 18</u> : Gestion des pages JSP.....	48

Liste des abréviations

<i>Abréviation</i>	<i>Désignation</i>
XML	eXtensible Markup Language
SQL	Structured Query Language
MVC	Model View Controller
AO	Atos Origin
MT	Maroc Telecom
EJB	Entreprise Java Bean
DAO	Data Access Object
API	Application Programming Interface
TEH	Table Extraction Handler
CDR	Call Détail Record
PFIH	Pré File Input Handler
FIH	File Input Handler
PRIH	Pré Rate Input Handler
RIH	Rate Input Handler
RLH	Rate Load Handler
TIMM	Telecommunication Invoice for Multiple Markets
BCH	Bill Cycle Handler
JSP	Java Server Pages

Sommaire

Dédicace	2
Remerciements.....	3
Liste des figures.....	4
Liste des Abréviations.....	5
Sommaire.....	6
Introduction générale.....	7
<u>Partie : 1</u> Contexte général du projet.....	9
<u>Chapitre : 1</u> Présentation de l'organisme d'accueil.....	10
1. Présentation générale	10
2. Atos Origin en Afrique du Nord	11
<u>Chapitre : 2</u> Présentation du projet.....	16
<u>Partie: 2</u> Analyse et conception.....	25
<u>Chapitre : 3</u> Cahier de charges du projet.....	26
1. Avant-propos	26
2. Objectif	26
3. Configuration technique de l'application à développer	26
4. Périmètre.....	27
5. Utilisateurs de l'application.....	27
<u>Chapitre : 4</u> Cycle de vie	28
1. Phase préparatoire.....	29
2. Phase de la réalisation.....	30
3. Phase de la mise en œuvre.....	31
<u>Chapitre : 5</u> Conception générale.....	32
I. Architecture adoptée.....	33
II. Diagrammes UML adoptés.....	37
<u>Partie : 3</u> Mise en œuvre du projet.....	43
<u>Chapitre : 6</u> Technologies et outils de développement.....	44
1. Plateforme de travail J2EE.....	45
2. Les technologies de développement.....	46
3. Les langages de développement.....	50
4. Les outils de développement.....	52
<u>Chapitre : 7</u> Interfaces de réalisation.....	53
Conclusion et Perspectives.....	64
Bibliographie et Webiographie.....	65
Annexe	66

Introduction générale

De nos jours, les télécommunications revêtent une importance primordiale au sein de la société et de ce fait, elles doivent être considérées comme un secteur stratégique contribuant dans une large mesure, au développement économique et social.

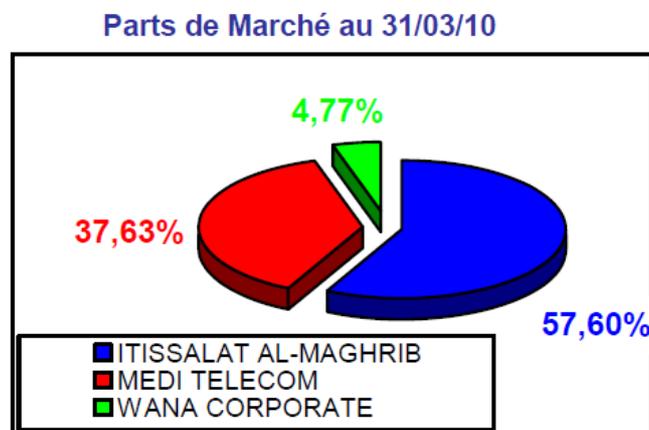
En effet, le Maroc est l'un des pays les plus avancés en Afrique en matière de télécommunications, avec un secteur majoritairement privatisé et une grande pénétration de la téléphonie mobile (55%) selon une étude faite par l'entreprise «Research and Markets», basée à Dublin, en Irlande du nord.

Il possède l'un des plus grands réseaux des télécommunications, de par son infrastructure et la qualité de ses services. Ce réseau a connu une forte croissance pendant la première moitié de la décennie, notamment avec la mise en service de nouvelles techniques et l'apparition de la concurrence entre trois opérateurs de télécommunications.

Et en parlant de concurrence, cette dernière constitue le meilleur levier pour pousser les opérateurs à améliorer le rapport qualité prix des produits et des services offerts aux clients dans le monde des télécommunications, et ceci peut être montré à travers les statistiques suivantes :

Marché de la téléphonie mobile :

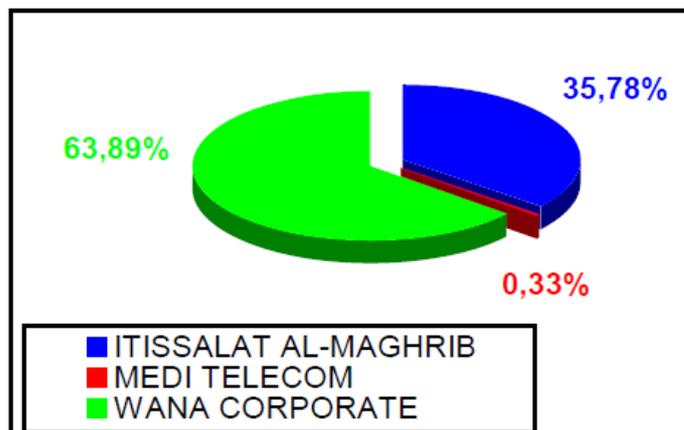
En terme de part de marché et à la date du 31/03/2010, l'opérateur historique IAM (Itissalat Al Maghrib) détient 57,60% du parc mobile suivi de Medi-Telecom avec 37,63% et Wana Corporate avec 4,77%.



Marché de la téléphonie fixe :

En termes de part de marché et à la date du 31/03/2010, l'opérateur Wana Corporate détient 63,89% du parc fixe suivi d'IAM avec 35,78% et Medi Telecom avec 0,33%.

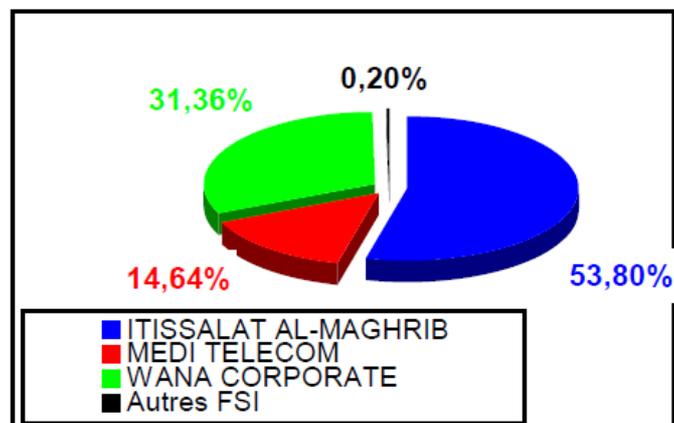
Parts de Marché au 31/03/10



Marché de l'internet :

En termes de part de marché et à la date du 31/03/2010, IAM détient 53,80% du parc Internet suivi de Wana Corporate avec 31,36% et de Medi Telecom avec 14,64%.

Parts de Marché au 31/03/10



De ce fait, cette concurrence évolutive entre les différents opérateurs les a poussé à augmenter le taux de la production des offres commerciales sur le marché, et donc pour gagner en temps, en coût et en qualité de travail. Maroc Telecom a ainsi proposé de créer un outil pour automatiser la majorité de ses offres, ce qui a constitué le sujet de mon projet qui porte sur l'automatisation des offres commerciales configurées dans BSCS (qui sera défini dans le chapitre 2) à l'aide d'une application WEB.

N.B : Vu que ces statistiques étaient effectuées en Mars 2010, on trouve toujours l'opérateur WANA au lieu d'INWI.

Partie 1 :

Contexte général du projet

L'objectif de cette partie est la présentation du contexte général du projet : elle propose de donner dans un premier temps une brève présentation de l'organisme d'accueil « Atos Origin », son client Maroc Telecom ainsi que sa structure et ses domaines d'activités, ensuite elle donne une brève présentation du projet.

Chapitre 1 : Présentation de l'organisme d'accueil

1. Présentation générale :

Atos Origin est l'un des principaux acteurs internationaux du secteur des services informatiques. Sa mission est de traduire la vision stratégique de ses clients en résultats par une meilleure utilisation de solutions de Conseil, Intégration de Systèmes et Infogérance.

Implanté dans 40 pays à travers le monde, Atos Origin réalise un chiffre d'affaires annuel de 5,8 milliards d'euros avec un effectif de plus de 51 000 personnes. Atos Origin est le partenaire informatique mondial des Jeux Olympiques et le Groupe compte parmi ses clients de grands comptes internationaux dans tous les secteurs d'activité. Il exerce ses activités sous les noms d'**Atos Origin, Atos Worldline et Atos Consulting.**

Une présence mondiale



Figure 2: Présence mondiale

2. Atos Origin en Afrique du Nord :

La société Atos Origin est présente dans plusieurs pays de l'Afrique du Nord tel que la Tunisie, l'Algérie et le Maroc.

En 2003, Atos Origin a été créé au Maroc, et ceci afin d'être plus proche des clients de l'Afrique du nord. Elle compte parmi ses effectifs plus de 420 personnes. Ses activités s'orientent principalement autour de la formation des ressources locales, du transfert des compétences dans le but de créer un savoir-faire contextuel, et par la suite, de la proposition de solution pérennes.

2.1 Activités :

Atos Origin se base sur quatre Activités :

- CONSEIL :

Le Groupe offre des services et solutions de «bout en bout », allant du développement de stratégies jusqu'au choix de solutions et de technologies appropriées en passant par la refonte des processus fonctionnels.

Ses clients sont ainsi en mesure d'améliorer leur productivité et de générer davantage de valeur grâce à une approche innovante des processus métiers, conjuguée à une intégration performante des technologies, ainsi que par des investissements stratégiques en ressources humaines. Atos Consulting s'assure que tous les aspects d'une organisation « ressources humaines, processus et technologie » sont entièrement alignés sur la stratégie de l'entreprise.

- INTÉGRATION DES SYSTÈMES :

L'association réussie de nouvelles solutions et des processus existants permet de transformer l'architecture de l'entreprise en un système opérationnel uniforme et transparent. La vaste expérience en matière d'intégration des personnes, des processus et des technologies permet à ATOS Origin de concevoir, de bâtir et d'exploiter des solutions pratiques et robustes. Ses spécialistes travaillent avec les clients au développement, au déploiement et à la maintenance de systèmes au service de leur stratégie métier globale. Atos Origin collabore

avec des partenaires stratégiques en vue de développer et mettre en œuvre des offres complètes et des solutions logicielles dans des environnements complexes utilisant les technologies les plus pointues.

- **INFOGÉRANCE :**

Atos Origin prend en charge la gestion des infrastructures informatiques clés de ses clients : centres de données, assistance micro-informatique, parcs de serveurs et réseaux de communication.

La société fournit, à travers son réseau mondial, des services accessibles en continue, dispose d'une expérience confirmée en matière de déploiement de solutions complexes multi sites. Pour assurer un service de qualité, Atos Origin s'appuie sur sa propre méthodologie de prestation de service continu (CSDM) tout au long du processus d'évaluation, de planification, de mise en œuvre et de transition.

- **SERVICES TRANSACTIONNELS À HAUTE VALEUR AJOUTÉE :**

Au travers de sa filiale Atos Worldline, Atos Origin est le leader européen dans les paiements et les transactions électroniques. Atos Worldline est spécialisé dans les services de paiements (émission, acquisition, solutions de paiement et traitement de cartes), les services pour les marchés financiers (clearing & settlement, services de courtage et gestion d'actifs), les e-Services (services Internet, vocaux et mobiles). Atos Worldline fournit des solutions et services innovants permettant d'améliorer la performance métier de ses clients. Atos Worldline offre également une forte capacité de traitement industriel permettant de traiter des milliards de transactions électroniques.

2.2 Organisation :

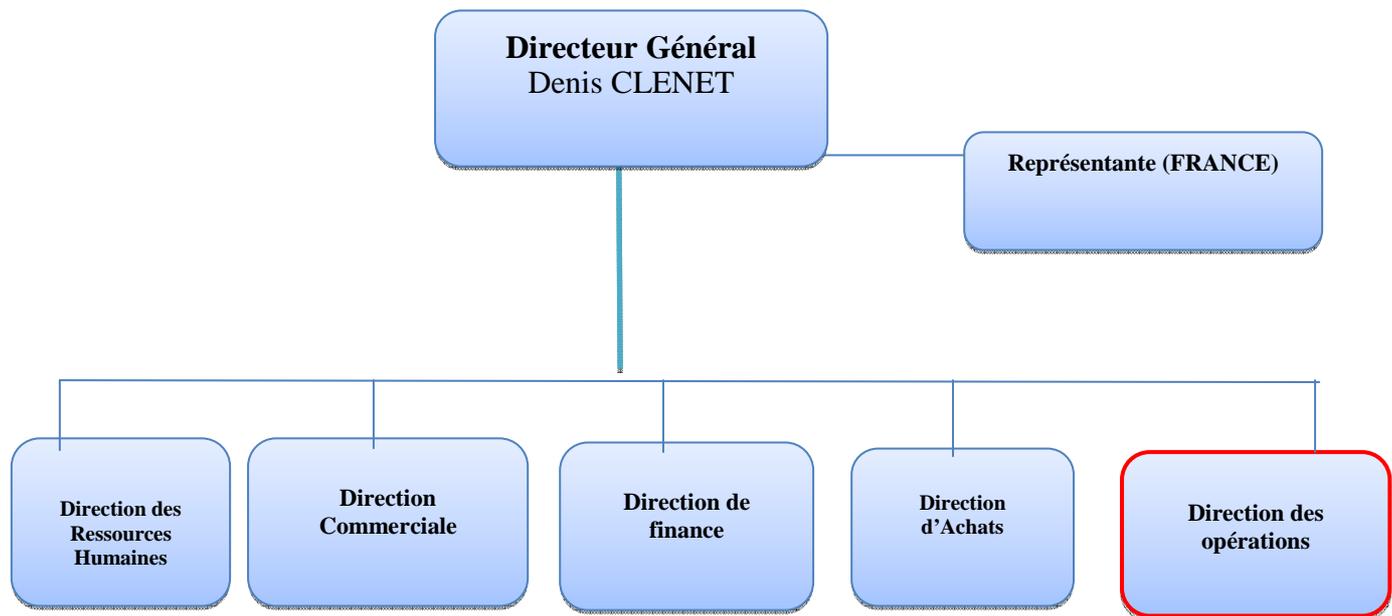


Figure2: Schéma d'organisation d'Atos Origin au Maroc

Les missions des différentes directions consistent en :

Direction des Ressources Humaines :

- ❖ Recrute des profils en adéquation avec sa culture interne.
- ❖ Favorise l'intégration des ingénieurs.
- ❖ Anticipe l'évolution des métiers, gestion de carrière et enrichit les compétences de ses collaborateurs,
- ❖ Accompagne les managers opérationnels : définit et met en œuvre les actions nécessaires pour le développement des opérations,
- ❖ Assure l'administration du personnel (paie, congés, couverture, astreintes...).

Direction Commerciale :

- ❖ La prospection du marché local et régional.

- ❖ Le suivi des comptes et détection des opportunités d'affaire,

Direction de Finance :

- ❖ S'assurer de l'équilibre financier de la société,
- ❖ Traduire les objectifs de la société en budget et suit leur réalisation.

Direction d'Achats :

- ❖ Effectuer tous les types d'achats pour les différentes entités en garantissant une meilleure prestation de produits et service en termes de prix et de qualité.

Direction des Opérations ou de la production ou j'ai effectué mon stage :

- ❖ Est chargée de la réalisation technique du produit demandé par le client soit en Système d'Intégration ou en Infogérance
- ❖ Est responsable de la déclinaison des plans de déploiement en matière d'industrialisation.
- ❖ Assurer le bon fonctionnement opérationnel des projets, l'industrialisation des outils et processus et le déploiement des meilleures pratiques.

2.3 Partenaires :



Figure 3: Liste des partenaires d'Atos Origin

Activités d'Atos Origin au Maroc :

❖ Elle se développe à grande vitesse. Dans son portefeuille, de grands comptes prestigieux privés comme publics : Maroc Telecom, Bank Al-Maghrib, ministère des Finances, Trésorerie Générale du Royaume, Bourse de Casablanca, etc.

Données sur Maroc Telecom :

Maroc Telecom est l'opérateur historique de télécommunications au Maroc, leader sur l'ensemble de ses segments d'activités (téléphonie fixe, mobile et Internet). Depuis décembre 2004, Maroc Telecom est coté en bourse à Casablanca et à Paris. Ses actionnaires de référence sont le groupe Vivendi Universel (51%) et le l'état marocain (34,1%). Avec plus de 8,576 millions de clients mobiles, 1,336 million de lignes fixes et 300 000 lignes ADSL Maroc Telecom a réalisé un chiffre d'affaire de plus de 22 milliards de dirhams durant les neuf premiers mois de l'année en cours de l'année 2010.

Chapitre 2 : Présentation du projet

Objectif du projet :

Il s'agit de concevoir et de réaliser une application web dont l'objectif principal est de générer des scripts SQL relatifs à plusieurs services liés aux domaines : fixe, mobile ou internet. Ces scripts seront par la suite configurés dans le progiciel BSCS qui va les traduire en des factures, gérant des offres commerciales de Maroc Telecom, que le client doit payer à cet opérateur.

Alors qu'est ce qu'un BSCS et sur quoi est-il basé ?

I. Présentation du BSCS (Business Support et Control System) :

BSCS est un progiciel qui sert à facturer les services offerts au client en liant la communication effectuée par ce dernier à la plateforme réseau afin de déterminer la facture qu'il doit payer. Le BSCS est basé sur l'architecture Client /Serveur :

- Serveur:
 - OS: UNIX,
 - Code: C/C++.
 - base de données Oracle,
- Client:
 - OS: Windows.
 - Code: C/PLSQL.

Fonctionnement du BSCS :

Le BSCS permet de transformer des fichiers CDR (Call Détail Record) qui contiennent le détail d'un appel (numéro de l'appelant et de l'appelé, l'heure, la date et la durée de l'appel) en des factures à payer par le client. Cette transformation passe par deux chaînes une de la

facturation et une autre de la valorisation. Le schéma suivant en donne l'illustration qui le montre :

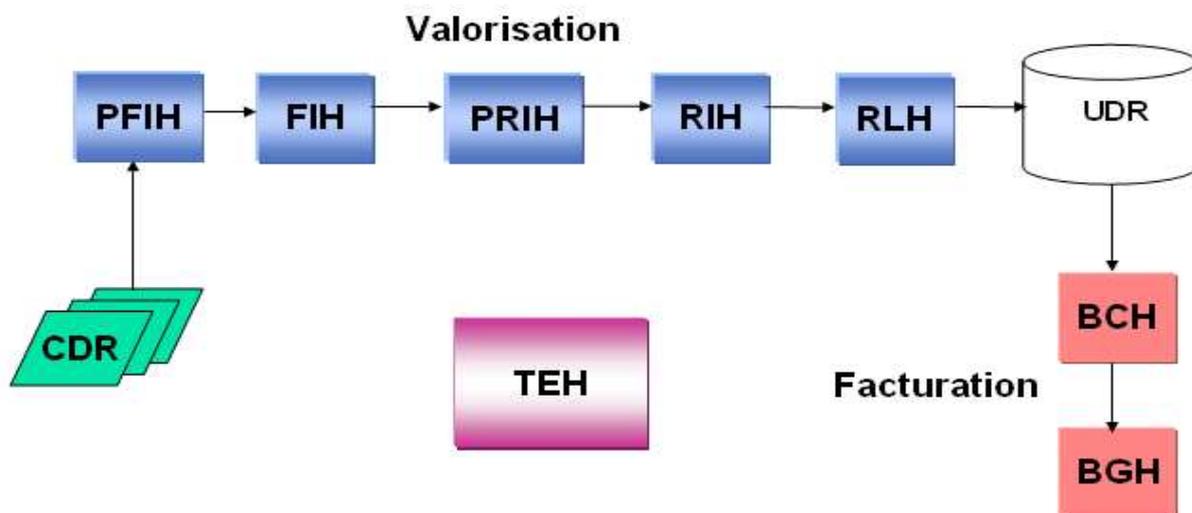


Figure4 : Vue générale des modules du BSCS

N.B : La figure 5 est détaillée dans l'annexe.

II. Services :

Comme un client de MAROC TELECOM, peut bénéficier d'un ou de plusieurs services dans les différents domaines : fixe, mobile, ou internet. Le progiciel doit facturer n'importe quel service quelque soit son paquet et son type. Comme illustré ci-dessous :

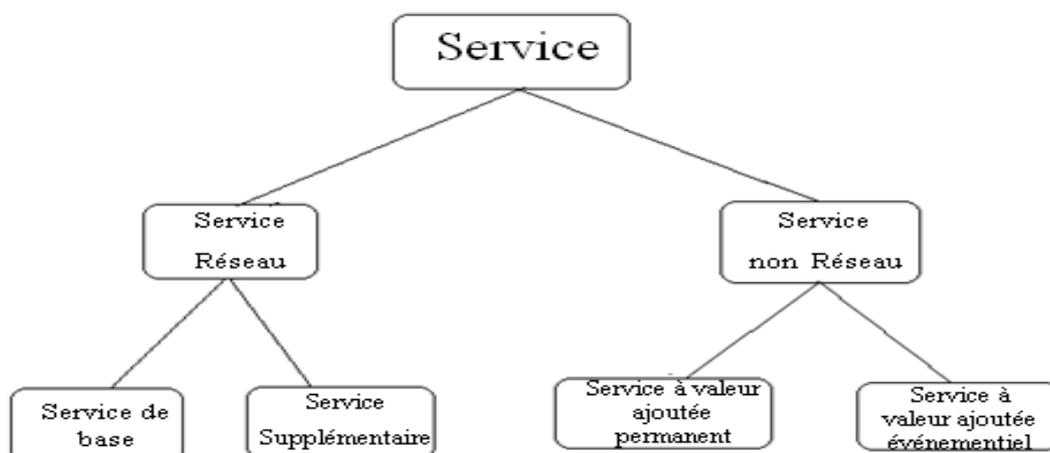


Figure 5 : Type et paquet de service

- Chaque type de service comporte des frais de mise en service, des frais périodiques et des frais d'usage.

Types de services	Exemples
➤ Service de base	➤ Téléphonie, sms, mms,....
➤ Service supplémentaire	➤ Présentation du numéro,...
➤ Service à valeur ajoutée permanent	➤ Facture détaillée, restriction d'appel,...
➤ Service à valeur ajoutée événementiel	➤ Retard de paiement → résiliation → pénalité d'engagement, Recharge d'Elmanzil,...

Tableau : 1 Présentation des types de services accompagnés par des exemples

Vu ces différents services, il y'aura par conséquent une multitude de scripts SQL à générer. Mon application web ne met le point que sur la génération des scripts les plus demandés par MAROC TELECOM. En effet, cette application traite les points suivants :

1. Ajout ou Modification d'un code comptable :

Un code comptable est utilisé par les écritures comptables de Maroc Telecom, afin de déterminer son revenu par type de service dans les domaines : fixe, mobile et internet. Selon les besoins de MT, on ajoute ou on modifie un code comptable. Le code peut être lié à un ou à plusieurs services.

2. Création d'un service :

La création d'un service se réalise selon la figure suivante :



Figure 6 : Eléments nécessaires pour la création d'un service

Interprétation :

Pour créer un service il est nécessaire de :

- Déterminer le type auquel il doit appartenir (fixe, mobile, internet).
- Déterminer le paquet auquel il doit appartenir (cela peut être un service de base, service permanent ou événementiel) et lui affecter un SPCODE (code lié au paquet du service).
- Lui attribuer un SNCODE (de type Number) car chaque service créé possède un SNCODE unique.
- Lier le type du service créé à son paquet.
- Lier, selon la demande du client, le service créé à un événement provoqué par ce dernier: par exemple la demande d'une recharge d'ELMANZIL à l'aide du numéro 114.
- Lier, au cas où le service créé possède des frais d'accès et d'abonnement, à un plan tarifaire qui s'occupe de facturer ses frais aux clients de MT (Maroc Telecom).

3. Modification du libellé d'un service, d'un paramètre ou d'une combinaison de paramètres dans un service :

- Chaque service créé possède un libellé qu'on peut modifier grâce à l'application web développée.
- Pour gérer les choix de certains éléments par le client dans un contrat, on a fait appel à la notion de « paramètre » tel que : le débit, la durée d'engagement etc. Ce paramètre tout comme un service a un libellé qu'on peut modifier à l'aide d'un script SQL généré.
- Chaque paramètre possède plusieurs valeurs par exemple le débit a comme valeurs 128kb/s, 256kb/s, 1méga, 2méga,... Il y a lieu de noter que la valeur d'un paramètre peut être liée à un ou à plusieurs valeurs d'un autre paramètre. Par exemple :

Paramètre1	Valeur	Paramètre2	Valeurs	Tarifification
Débit	128kb/s	Durée d'engagement	1an	100 DH
		Durée d'engagement	2ans	50 DH

4. Vérification de la compatibilité entre deux services :

Ce point permet de vérifier la compatibilité entre deux services tout en générant un fichier nommé ruletab et d'extension .stb. En effet à l'aide de ce fichier on peut dire que l'activation d'un service1 nécessite l'activation ou non d'un service 2.

- **L'utilité du fichier ruletab après sa génération :**

L'équipe technique de MT possède un module qui leur permet de créer de nouveaux contrats pour les nouveaux clients. Et pour éviter de saisir et donc de créer des contrats inutiles, le fichier ruletab permet de faire des contrôles sur la saisie des utilisateurs de ce module. Par exemple : un utilisateur de ce module ne peut pas créer un contrat à un client qui demande le service de la téléphonie sans avoir auparavant l'ADSL → Donc, grâce au fichier

ruletab, on peut s'assurer que le contrat d'avoir le service téléphonie existe déjà et après on peut créer le contrat pour avoir l'ADSL à un client. En termes de compatibilité, on peut dire que l'activation du service ADSL nécessite l'activation obligatoire du service de la téléphonie.

5. Création d'un plan tarifaire en liaison avec un rating package :

a) Un plan tarifaire :

Un plan tarifaire est un paquet de services dont chaque service a un coût précis. C'est pour cela qu'un plan tarifaire est utile pour facturer les accès et les abonnements. Chaque contrat d'un client de MT possède un plan tarifaire.

N.B :

Les frais d'accès sont les premiers frais payés pour avoir un service.

Les frais d'abonnement sont ceux payés pour garder un service même si on ne l'a pas utilisé. Par exemple : pour avoir la téléphonie chez nous (le poste fixe), on paie plusieurs frais tel que :

=>Les frais primaires pour avoir la ligne téléphonique, correspondant aux frais d'accès.

=> Les frais pour garder la ligne même si parfois, elle n'est pas utilisée pour des appels, correspondant aux frais d'abonnement qui sont fixes (à peu près 100DH)

=> Les frais pour utiliser cette ligne, qu'elle soit une ligne illimitée ou en charge à l'aide d'une carte ELMANZIL. Ils sont appelés frais d'usage.

b) Rating Package :

Le Rating Package est un fichier SQL qui permet de convertir les données d'un fichier CDR en montant que le client doit payer à Maroc Telecom. Le fichier CDR contient tous les détails d'un appel effectué par un individu tel que le numéro de l'appelant, le numéro de l'appelé, la date et la durée de l'appel. Ainsi un plan tarifaire sert à facturer les frais d'accès et d'abonnement, alors que le rating package sert à facturer les frais d'usage.

Chaque script SQL au sein d'ATOS ORIGIN et de son client MAROC TELECOM passe ainsi par le processus suivant :

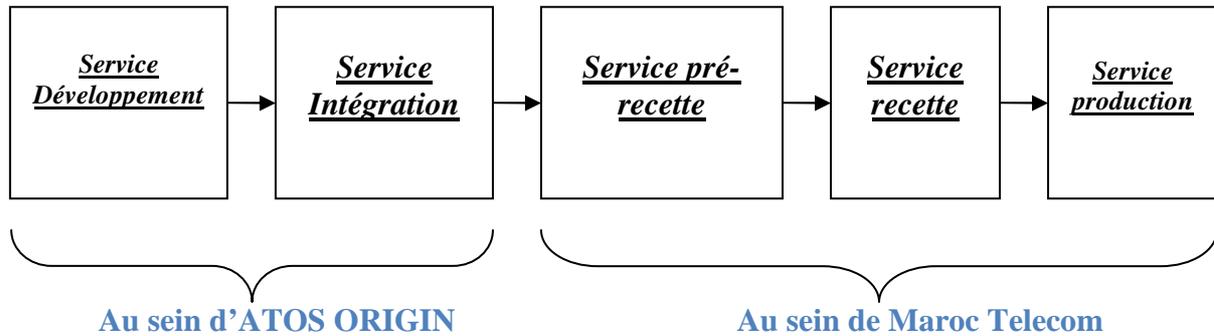


Figure7 : Processus de la transformation d'un script SQL en une offre commerciale

Interprétation :

Traitement du script SQL au niveau d'ATOS ORIGIN :

Le script SQL au sein d'Atos Origin passe par obligatoirement par les services suivants:

- ***Service de Développement*** : développe les scripts SQL selon le besoin demandé.
- ***Service Intégration*** : teste les scripts livrés par le service de développement.

Traitement du script SQL au niveau de MAROC TELECOM :

Le script SQL au sein de Maroc Telecom passe par les services suivants:

- ***Service pré-recette*** : c'est un service de Maroc Telecom, opéré par le DSI (Département Système Informatique) qui effectue un deuxième test des scripts livrés par le service d'Intégration d'Atos Origin.
- ***Service recette*** : est un service dans lequel service de Maroc adapte le script développé au concept de marketing de telle sorte que ce script va générer une offre commerciale différenciée et adaptée au besoin du marché.
- ***Service production*** : Il dirige l'offre commerciale vers le marché et la met à la disposition des consommateurs potentiels.

III. Schéma d'application :

L'application que j'ai eu à développer est une application web qui permet la gestion des offres commerciales de Maroc Telecom. Elle est déployée dans le serveur tomcat, et se répartie de la manière suivante :

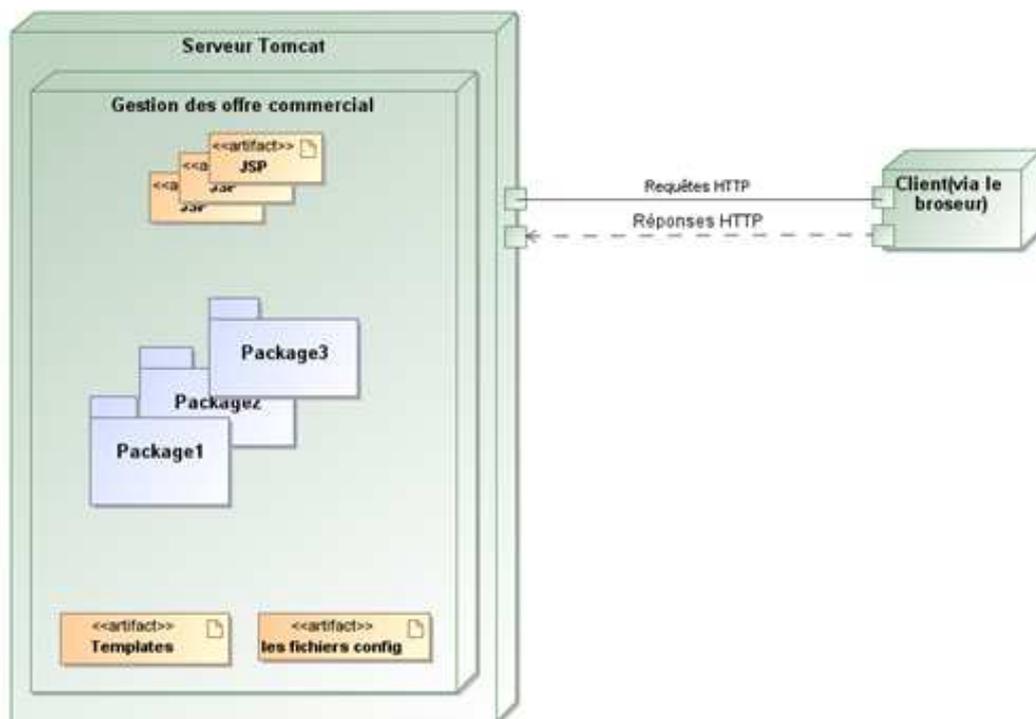


Figure8 : Déploiement de l'application web

- Lorsqu'un client se connecte via un browser (Internet explorer, Mozilla firfox,..), il envoie une requête via une communication HTTP au serveur tomcat. Celui ci contient l'application web qui est composée des éléments suivants :
 - Vues JSP : présentent l'Interface Homme Machine (IHM). Elles permettent à l'utilisateur de communiquer avec tout le système informatique.
 - Des packages : Dans cette application web, il existe plusieurs packages qu'on va voir en détails dans le chapitre de conception.

➤ **Fichiers :**

Cette application contient trois types de fichiers :

- Des templates SQL qui se transforment en des scripts SQL.
- Ruletab.stb dans lequel s'ajoutent automatiquement deux lignes de code pour vérifier la compatibilité ou non entre deux services.
- Fichier d'extension .csv : qui est un fichier de configuration qui permet de gérer l'accès aux pages JSP.

Ainsi selon la requête de l'utilisateur, tomcat répond en lui envoyant la page adéquate.

Il y a lieu de souligner que les objectifs de notre projet ne seraient atteints que si on répond au cahier des charges déterminé par le client. C'est précisément l'objet du chapitre suivant.

Partie 2 : Analyse et Conception

Cette partie traite les deux phases les plus importantes et les plus critiques de tout cycle de développement. Ce sont les phases d'analyse et de conception.

Le premier chapitre présente le cahier des charges demandé, le deuxième présente le cycle de vie adopté. Ainsi ces deux chapitres représentent la phase d'analyse dont le but est de délimiter le périmètre du projet en identifiant les fonctionnalités auxquelles le système doit répondre.

Les résultats de la phase d'analyse sont la base de la phase de conception traitée par le cinquième et sixième chapitre.

Chapitre III : Cahier des charges

1. Avant-propos :

Ce cahier des charges décrit les besoins fonctionnels d'ATOS ORIGIN en matière de la génération des scripts SQL configurées dans le progiciel BSCS pour une bonne gestion des offres et des promotions commerciales demandées par Maroc Telecom. Il ambitionne davantage pour gagner en temps en coût et en qualité de travail.

2. Objectif :

L'objectif de ce projet est la mise en place d'une application web permettant la gestion des Scripts SQL des offres commerciales à savoir :

- ❖ **Maintenabilité** : bien commenter le code en plus d'un document décrivant le programme et les différentes vues de l'application,
- ❖ **Accessibilité** : Prendre en compte les accès aux différents serveurs (qu'ils soient au siège Atos, sur un domaine client sur un site ou un autre, l'accès doit être possible),
- ❖ **Identification** : Seul l'administrateur doit avoir son propre identifiant et mot de passe pour des raisons de sécurité.
- ❖ **Droit administrateur** : Il possède une série de droits dont ceux d'écriture (en modifiant ou supprimant des scripts SQL liés aux différents services), d'interrompre l'accès aux différentes vues, de modifier l'identifiant et le mot de passe pour accéder à l'administration des scripts SQL et même d'ajouter dans le futur de nouveaux scripts liés à de nouveaux services.

3. Configuration technique de l'application à développer :

Mon application web utilise les techniques suivantes :

- ❖ Système d'exploitation : Windows, Unix.

- ❖ Les langages utilisés dans le développement de l'application : Java, Jsp, JavaScript et CSS (pour donner un joli style aux différentes vues de l'application web).
- ❖ Les données : fichier csv peut être exploité par Excel et des scripts SQL utilisés pour avoir des templates SQL.
- ❖ L'application est une application Client/serveur.

4. Périmètre :

L'application couvre le processus depuis le remplissage des champs des formulaires dans les différentes vues ainsi que la génération des scripts SQL ou d'un fichier ruletab selon le besoin de l'utilisateur.

5. Utilisateurs de l'application :

Les utilisateurs (professionnels du BSCS) peuvent se connecter et générer des scripts SQL selon leurs besoins sans s'identifier. Des normes d'utilisation précises et respectées par AO/MT sont édictées pour éviter les abus.

Le profil de chaque utilisateur devra lui permettre d'effectuer les tâches qui lui sont assignées, à savoir:

- ❖ L'administrateur devra gérer l'ensemble du contenu du site ainsi que de paramétrer l'application en gérant l'accès aux différentes vues et la génération des scripts SQL ou du fichier ruletab qui permet de vérifier la compatibilité entre deux services.

Les utilisateurs peuvent consulter, et générer les fichiers d'extensions .sql ou .stb.

Chapitre 4 : Cycle de vie du projet

Le cycle de vie d'un projet présente l'enchaînement dans le temps des étapes et des validations entre l'émergence du besoin et la livraison du produit commercial. De cela, l'adoption d'un cycle de vie prédéfinie pour le développement d'une application web peut s'avérer une tâche très importante pour sa réussite. Dans ce projet, j'ai suivi le cycle de vie circulaire qui est perçu le plus adéquat pour mieux répartir les tâches de ce projet.

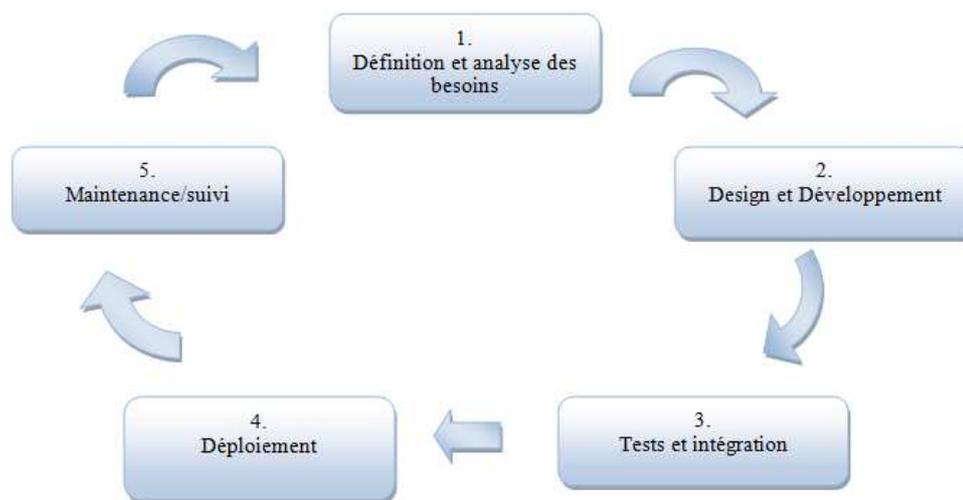


Figure9 : Cycle de vie du projet web

Ce projet peut être découpé de façon basique de la manière suivante : Phase préparatoire, Phase de la réalisation et Phase de la mise en œuvre. Chaque phase comporte une activité bien déterminée dans le cycle de vie.

I. Phase préparatoire :

Elle permet de prendre conscience du projet, puis d'étudier son l'objet pour s'assurer que sa mise en œuvre est pertinente et qu'il entre dans la stratégie de l'entreprise. Selon le cycle de vie adopté pour ce projet web, cette phase englobe deux activités : la définition et l'analyse des besoins. Ces activités sont basées principalement sur :

- **L'Etude d'opportunité** : présente une étude du contexte global dans lequel s'inscrit le client, et sur laquelle on se base pour extraire et définir les besoins.
- **L'Etude de faisabilité caractérisée par :**
 - **Analyse concurrentielle** : qui permet de déterminer la position du client vis-à-vis du marché et des autres concurrents. En effet, cette étude ne permet de valider les besoins donnés par l'étude d'opportunité que s'ils répondent aux critères permettant de mieux placer la société dans l'échelle de la concurrence et du professionnalisme.
 - **Analyse business, marketing** : qui permet en fonction des exigences du client, de cerner ses besoins et ainsi arriver à des moyens pour y répondre, par exemple : dans le contexte d'un projet web, grâce à cette analyse, on peut orienter le projet vers une interface graphique très attractive, interactive.

- **Etude technique :**

Cette étude adopte l'Analyse technique qui permet la traduction des moyens que nous avons en des termes techniques dans le but d'établir un cahier des charges.

II. Phase de la réalisation :

La phase de la réalisation est la phase opérationnelle de la création d'un site web. Elle commence par la réception du cahier des charges et se clôture par la réalisation de l'application demandée. Elle est basée sur deux activités : design et développement d'une part et tests et intégration d'autre part :

- **Design et développement :**

L'activité design et développement met en jeu plusieurs tâches telles que :

- La conception du cahier des charges tout en précisant l'architecture adoptée et la modélisation UML utilisée.
- La création des maquettes.
- La réalisation du site web.

- **Tests et intégration :**

Cette activité a la charge de tester le contenu du site web dans le serveur local avant de l'intégrer dans le serveur réel de la société pour faire d'autres tests de validation.

III. Phase de la mise en œuvre :

La phase de la mise en œuvre représente la phase de la mise en production du projet web. En effet, elle assure que le projet réalisé est conforme aux attentes des utilisateurs et fait en sorte que son installation et son utilisation se déroulent correctement, ainsi que sa maintenance au cours du temps. Cette phase est basée sur deux activités: le déploiement ainsi que la maintenance d'une part. et le suivi du projet crée d'autre part.

- **Déploiement :**

Souvent la phase de l'intégration n'est pas optimale, car en réalité, il y aura toujours quelques petites modifications à ajouter telles que : la rectification d'une page web, l'ajout d'une nouvelle page et le contrôle de saisie des champs des différents formulaires. On peut donc dire que le Déploiement est la résultante de l'Intégration en plus des modifications ajoutées. Cette phase une fois validée par la société est accompagnée par un manuel d'utilisation ou par une formation du personnel censé utiliser cette application web.

- **Maintenance et suivi :**

On appelle la maintenance de l'application, l'ensemble des activités de modifications de l'application après livraison .Il est important que l'application soit bien développée ou maintenable pour diminuer les coûts de la maintenance d'où la nécessité de cette activité qui

prend en charge la conception du programme de maintenance, le suivi des performances ainsi que l'évolution du projet web.

Ainsi, on se base sur le résultat de ce chapitre pour faire la conception générale et de ce projet web.

Chapitre 5 : Conception générale

La conception d'une application web permet de déterminer l'architecture et les méthodes UML qu'elle a adoptées.

Les architectures J2EE sont utilisées essentiellement pour l'élaboration d'applications présentant une architecture complexe. Ainsi, la mise en place de ce type d'application nécessite l'intervention de plusieurs acteurs et l'utilisation de plusieurs composants (diagramme de cas d'utilisation, diagramme de déploiement, diagramme de séquence,...).

Les applications J2EE sont typiquement utilisées dans le cadre des architectures distantes type client/serveur.

I. Architecture adoptée :

1. Architecture Client/serveur:

L'architecture client-serveur est la base d'une application Web. En effet, la consultation de pages sur un site web fonctionne selon une architecture client/serveur. Chaque client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.

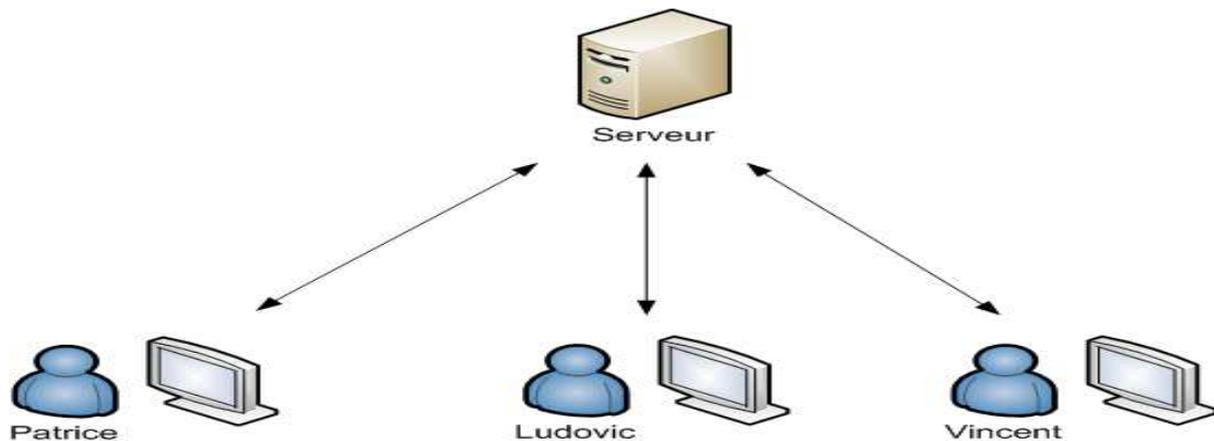


Figure 10 : Architecture Client/serveur

Une telle architecture est organisée sous forme de plusieurs couches interconnectées que l'on appelle «tiers». Elle met en place des interconnexions entre les différentes entités dans un traitement, on trouve ainsi, l'architecture 2-tiers et trois tiers.

Cette application web a adopté deux architectures dans ses deux parties : l'architecture 3-tiers pour gérer l'utilisation de l'application par un utilisateur différent de l'administrateur et l'architecture MVC (Model View Controller) pour contrôler la partie authentification gérée par l'administrateur.

En effet, ces choix n'étaient pas faits au hasard. On a choisi l'architecture 3-tiers dans la partie de l'application dédiée à un utilisateur différent de l'administrateur, car on a un nombre important de fichiers insérés dans la base (couche d'accès aux données) et donc pour ne pas saturer le serveur, on avait besoin de séparer la couche métier ou se fait les différents traitements de la couche accès aux données et ceci n'existe que dans l'architecture 3-tiers , et on a choisi l'architecture MVC dans la partie de l'application dédiée seulement à l'administrateur car on avait un seul fichier à gérer et on avait besoin de contrôler l'authentification pour accéder à l'administration des scripts (pour ne pas laisser n'importe qui y'accéder) et donc les couches qu'on avait n'étaient pas suffisantes et on avait besoin d'un contrôleur qui va effectuer cette tâche d'où l'utilisation du modèle MVC qui a fusionné la couche métier et la couche accès aux données et qui nous a fourni un contrôleur capable d'effectuer la partie authentification, ainsi on peut dire que loin d'être antagonistes, ces deux pratiques se combinent pour gérer les différentes applications Web.

1.1 Architecture 3-tiers :

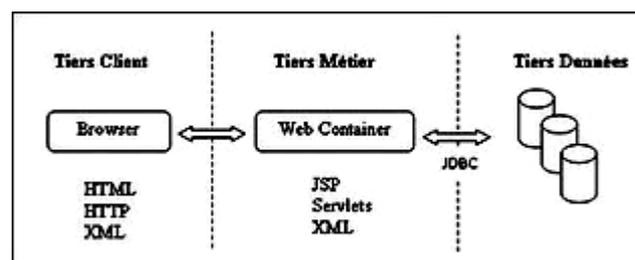


Figure 11 : Architecture 3-tiers d'une application Web

Comme le montre la figure ci-dessus, l'architecture 3-tiers d'une application Web est constituée de trois composantes dites 'tiers' :

Tiers client : gère les pages html via le protocole de communication http à l'aide du fichier de configuration XML.

Tiers Métier : est représenté par un web conteneur (web Container), ce dernier gère des pages JSP via des servlets en se basant sur le fichier paramétrable XML.

Tiers données : représente la base de données de l'application. Elle existe sous plusieurs types tels qu'une base de données relationnelle, ou des fichiers répartis d'extensions différentes.

Chaque « tier » cité est placé et représenté par une couche, ainsi on trouve trois couches principales :

Couche présentation :

La couche présentation correspond à la partie de l'application visible aux utilisateurs et qui leur confère l'interactivité. Elle transmet les requêtes de l'utilisateur à destination à la couche métier, et en retour, lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche métier

Couche métier :

La couche métier correspond à la partie fonctionnelle de l'application, qui est responsable est responsable d'effectuer tous les traitements par exemple dans mon application web , une telle couche permet de transformer un Template SQL en un script SQL,....

Couche d'accès aux données :

Elle sert à gérer l'accès aux données du système. Ces données peuvent être propres au système, ou administrées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme.

Cette application web a adopté deux architectures : l'architecture 3-tiers étudiées ci-dessus pour gérer l'utilisation de l'application par un utilisateur différent de l'administrateur et le modèle MVC (Model View Controller) qui sera traiter dans le reste de ce chapitre pour

contrôler la partie authentification gérée par l'administrateur, et donc loin d'être antagonistes, ces deux pratiques se combinent pour gérer des applications Web.

1.2 Architecture MVC :

Le MVC est un modèle d'architecture qui repose sur la volonté de séparer les données, les traitements et la présentation. En effet, elle lie les deux couches métiers, et accès aux données de l'architecture 3-tiers et ajoute un contrôleur qui répond aux événements de l'utilisateur et commande les actions sur le modèle. Ainsi l'application se retrouve segmentée en trois composants essentiels :

La vue (View) : correspond à l'IHM. Elle présente les données et interagit avec l'utilisateur. Dans le cadre des applications Web, il peut s'agir d'une interface HTML/JSP, mais n'importe quel composant graphique peut jouer ce rôle.

Le contrôleur (Controller): se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle puis de rediriger vers la vue adéquate. Il ne doit faire aucun traitement. Il ne fait que de l'interception et de la redirection.

Le modèle (Model) : représente les données et les règles métiers. C'est dans ce composant que s'effectuent les traitements liés au cœur du métier. Les données peuvent être liées à une base de données relationnelles ou contenant des fichiers répartis :

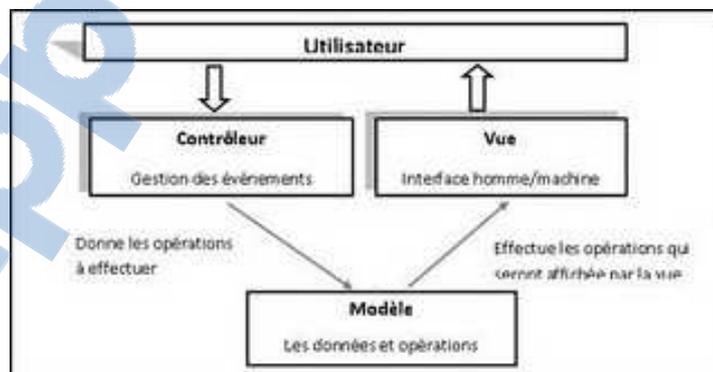


Figure 12 : Modèle MVC

II. Diagrammes UML adoptés :

En déterminant l'architecture adoptée par l'application web, il suffit de la modéliser pour pouvoir la réaliser.

1 - UML :

L'UML est un Schéma de réflexion fournissant la manière d'aborder les problèmes. La façon de mettre en œuvre ce schéma au cours du processus de conception se concrétise par une démarche proposant une succession progressive d'étapes.

L'UML permet de modéliser de manière claire et précise la structure et le comportement d'un système indépendamment de toute méthode ou de tout langage de programmation,

2- Moyen logiciel :

Pour utiliser une modélisation de données plus puissante, nous avons adopté le logiciel **MagicDraw**.

MagicDraw est un outil graphique de modélisation UML disposant de fonctions de travail collaboratif. Conçu pour les Consultants Métier, les analystes développeurs et les développeurs, cet outil de développement dynamique et polyvalent facilite l'analyse et la conception de systèmes orientés objets (OO) et de bases de données.

MagicDraw fonctionne sur un grand nombre de systèmes d'exploitation, tels que Windows 98/2000/XP/Vista, Solaris, Linux, MacOS, et tout autre système d'exploitation supportant Java 1.4, 5 ou 6. MagicDraw UML est actuellement disponible dans sa version 16.6.

3. Diagrammes UML utilisés :

On a utilisé MagicDraw pour réaliser la navigation présentée dans la figure suivante :

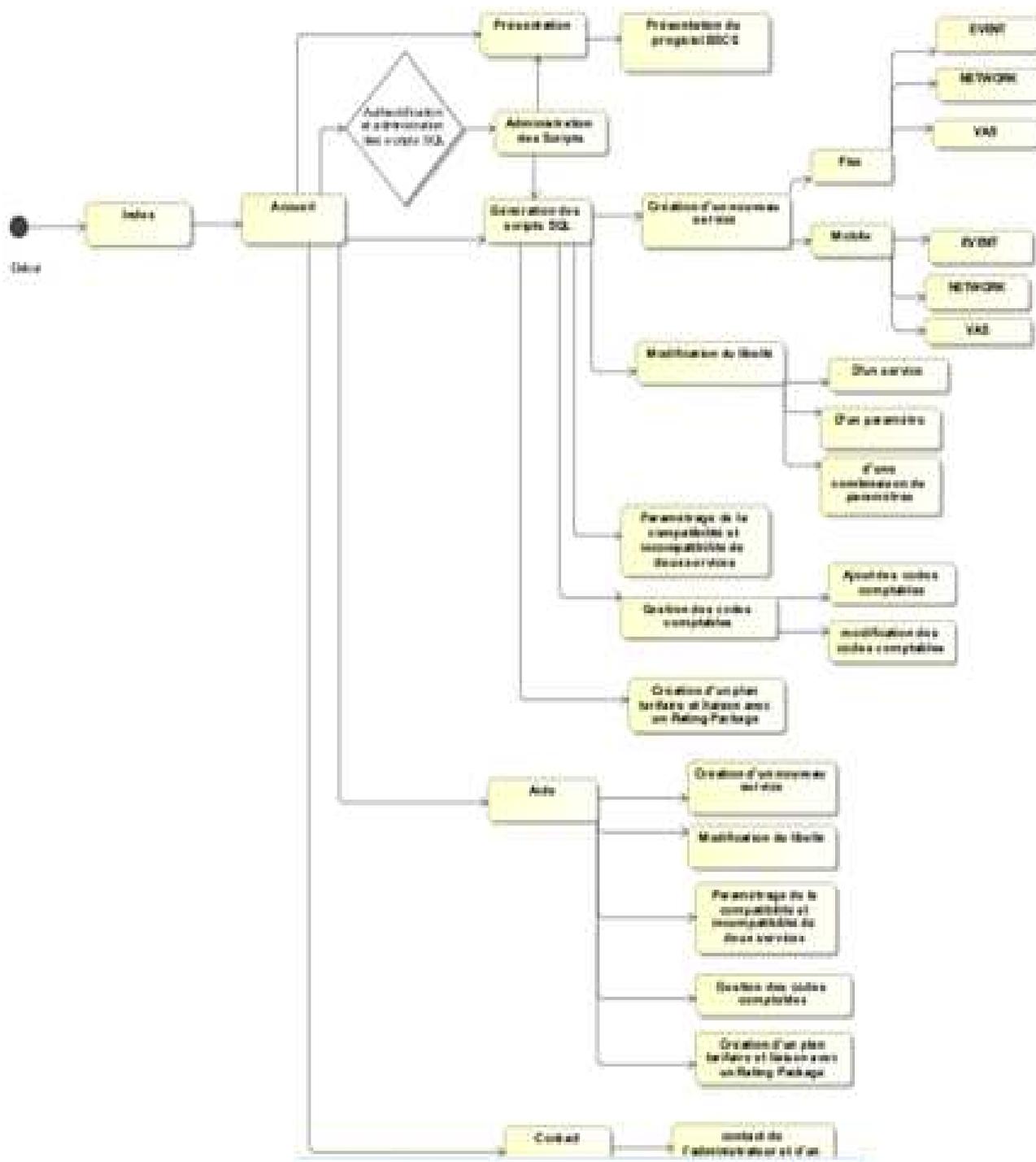


Figure13 : Navigation

La figure précédente présente le diagramme de navigation de toute l'application web .En effet on trouve toutes les pages JSP créés et l'interaction entre elles.

Il existe deux familles de diagrammes UML :

- **Famille de Description structurelle** : qui permet de décrire le côté statique du système.

Cette famille est représentée par plusieurs types de diagrammes parmi lesquels, il existe le diagramme de déploiement qui a pour but d'indiquer la disposition et l'organisation des différents équipements qui composent le système. L'adoption de ce diagramme à mon application web donne la figure suivante :

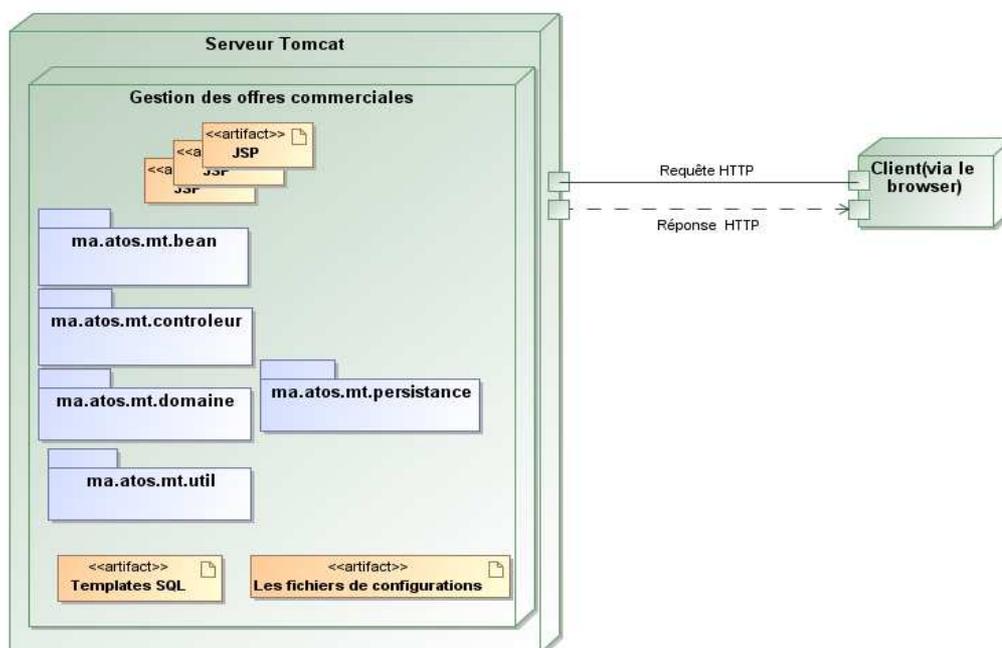


Figure 14: Diagramme de déploiement de l'application WEB

Les composants qu'on possède sont un composant client et un autre serveur. Dans le composant serveur, se loge mon application web. Ainsi, voila son principe de fonctionnement :

Lorsqu'un client se connecte via un browser (Internet explorer, Mozilla firefox,...), il envoie une requête via une communication HTTP au serveur tomcat. Celui ci contient l'application web qui est composée des éléments suivants :

- **Vues JSP** : présentent l'Interface Homme Machine (IHM). Elles permettent à l'utilisateur de communiquer avec tout le système informatique.

- **Des packages tel que :**

ma.atos.mt.bean : contient les informations de la session d'authentification ainsi que de la déconnexion de la session.

ma.atos.mt.contrôle : contrôle l'authentification faite par l'administrateur et les différentes vues JSP de telle sorte qu'il gère l'accès d'un simple utilisateur.

ma.atos.mt.domaine : contient le métier de toute l'application tel que la gestion des servlets qui génèrent les fichiers d'extension .sql et .stb.

ma.atos.mt.util : contient toutes les constantes utilisées dans les JSP (Java Server Pages), en plus d'une classe java pour crypter le mot de passe utilisé dans la session de l'authentification et ce pour des raisons de sécurité.

ma.atos.mt.persistance : contient la classe java qui nous permet d'utiliser un fichier .csv .Ce fichier est exploité pour gérer l'accès aux pages JSP permettant la gestion des scripts SQL.

- **Fichiers :**

Cette application contient trois types de fichiers :

- Des templates SQL qui se transforment en des scripts SQL grâce au remplissage, par un utilisateur professionnel, des différents formulaires existants dans les pages JS`P

- Ruletab.stb dans lequel s'ajoute automatiquement deux lignes de code pour vérifier la compatibilité ou non entre deux services.

- Fichier d'extension .csv : qui est un fichier de configuration qui permet de gérer l'accès aux pages JSP. En effet, soit le status des pages contenant les scripts à générer est actif et dans ce cas l'utilisateur génère le script selon son

besoin, soit le status est inactif est dans ce cas, un message d'erreur s'affiche en signalant que l'administrateur a désactivé ce script et qui est suivi d'une redirection vers la page d'accueil. Elle contient, par ailleurs, le login et le mot de passe qu'en cas de rectification, il est modifié au niveau de ce fichier.

Ainsi selon la requête de l'utilisateur envoyé via le protocole http, tomcat répond en lui envoyant la page adéquate.

- **Famille de Description comportementale** : qui permet de modéliser l'aspect dynamique du système.

Cette famille est représentée par plusieurs types de diagrammes parmi lesquels, il existe le diagramme de cas d'utilisation qui a pour objet de décrire les situations auxquelles le système doit répondre. L'adoption de ce diagramme à mon application web donne la figure suivante :

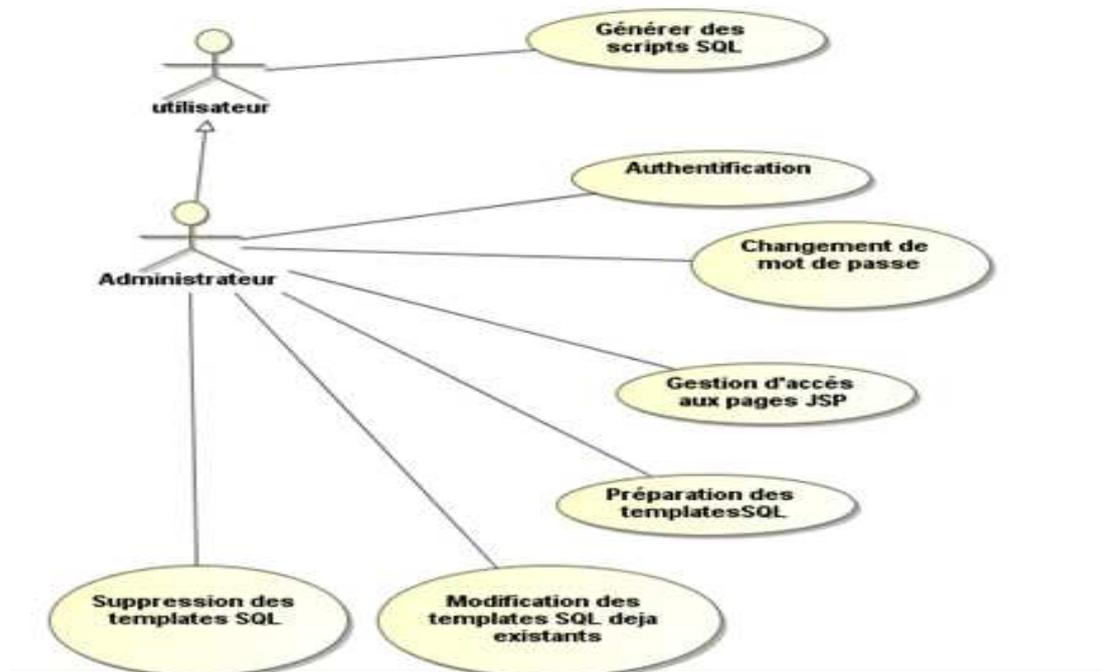


Figure 15: Diagramme de cas d'utilisation de l'application WEB

Chaque diagramme de cas d'utilisation possède les éléments suivants :

- Des acteurs : ce sont des personnes ou systèmes extérieurs à l'application, et qui interagissent avec elle. Dans notre exemple, les acteurs sont l'administrateur et l'utilisateur.
- Le système : présente l'application et les cas d'utilisation.
- Les cas d'utilisation : représentent les fonctionnalités du système à construire. Ils sont en relation avec des acteurs et d'autres cas d'utilisation.
- Relation : qui permet de lier les acteurs aux cas d'utilisation.

Ainsi, l'utilisateur peut gérer des scripts SQL selon son besoin et l'administrateur peut effectuer plusieurs tâches telle que l'authentification, le changement de mot de passe, gestion d'accès aux pages JSP, préparation des templates SQL, modification des templates SQL déjà existants et suppression des templates SQL.

En plus du diagramme de cas d'utilisation, il existe un autre diagramme dans la famille de description comportementale tel que le diagramme de séquence qui a pour but de montrer les relations entre objets d'un point de vue temporel. Il complète un cas d'utilisation. L'adoption de ce diagramme à mon application web à la partie de l'authentification donne la figure suivante :

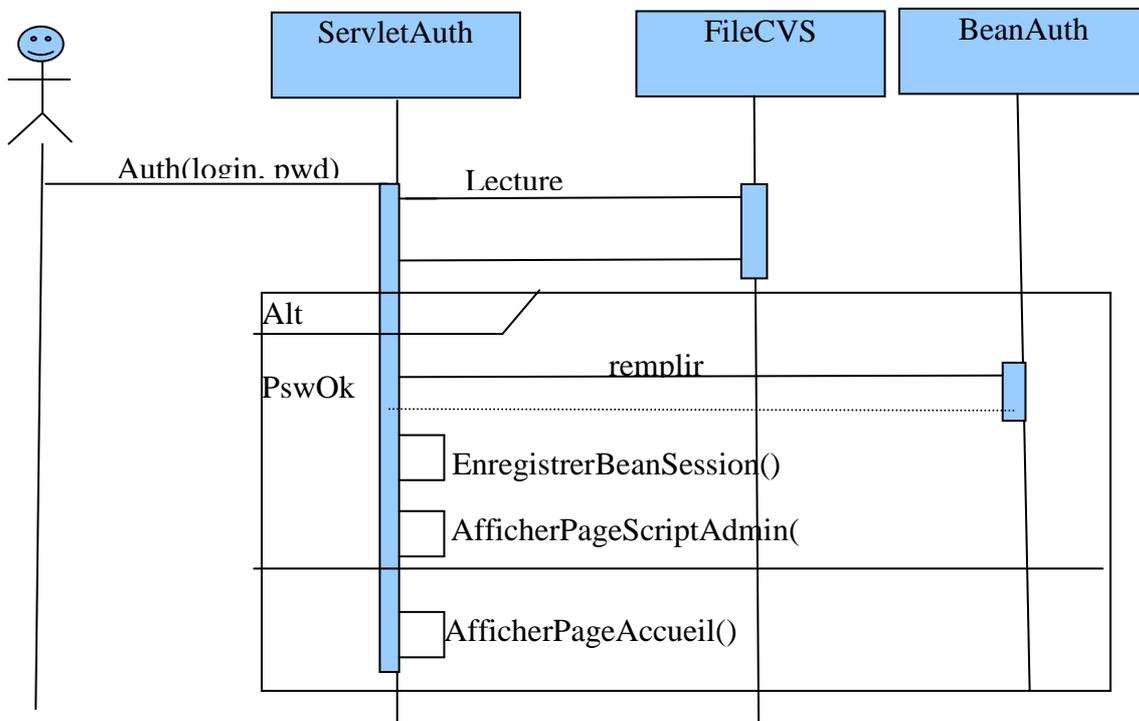


Figure 16 : Diagramme de séquence de l'authentification dans l'application web

Lorsque l'utilisateur s'authentifie, la servletAuth gère sa demande en comparant le login et le mot de passe (pwd) saisis par se dernier au login et mot de passe pwd correctes existant dans le fichier FILECSV. à ce niveau la servlet effectue le test suivant :

Si le pwd saisi par l'utilisateur est incorrect, la servletAuth dirige l'utilisateur à la page d'accueil.

Sinon, si le pwd est correct, la servletAuth remplit le BeanAuth. Ce dernier lui envoie un Bean ,sous forme de données, qui est enregistré dans la session et après l'utilisateur est dirigé vers la page de l'administration des scripts SQL.

Ainsi, pour rendre cette conception réelle et la réaliser, on aura besoin de plusieurs technologies et outils de développement et c'est l'objectif de la partie 3.

Partie3 : Mise en œuvre du projet

Cette partie donne une brève présentation des technologies et des outils utilisés pour la réalisation du projet web ainsi qu'une présentation de la solution finale.

Le premier chapitre présente les différentes technologies de développement, le deuxième traite les différents outils adoptés lors de la réalisation du projet, ensuite, le troisième chapitre met l'accent sur la solution finale en donnant les interfaces réalisées.

Chapitre 7 : Technologies et outils de développement

Le présent chapitre fournit une description globale de la plateforme de travail et des principales techniques de développement qui ont été utilisées dans ce projet.

I. Technologies de développement :

1. Plateforme du travail :

Une plateforme est une base générique qui fournit un ensemble de fonctionnalités utiles pour une majorité d'applications. Elle se construit sur la base d'un ensemble de besoins génériques partagés entre plusieurs applications. Il peut exister plusieurs types de plateformes, de la plus générique à la plus spécifique optimisée pour un type de métier précis.

L'avantage principal que présente une plateforme est que l'équipe de développement n'a pas à s'acquitter de développer certaines tâches (connexion à la base de données par exemple, gestion d'objets ...). Ce sont des tâches que l'on retrouve très souvent dans un grand nombre de projet et qui n'ont pas d'intérêt à être recodées à chaque fois, ce qui nous évite une perte de temps et d'argent).

De plus, il vaut mieux travailler sur une plateforme qui présente une forte stabilité pour éviter des débogages inutiles.

Un autre avantage réside dans la facilité de prise en main des API de cette plateforme. En effet, celle-ci cache très souvent la complexité d'accès à telle ou telle ressource et permet donc un gain de temps énorme pour le développeur qui a donc plus de temps pour se préoccuper du fonctionnement réel de son application (pas de tâche ardue ou générique à développer).

La plateforme utilisée dans ce projet est la plate-forme J2EE (Java 2 Enterprise Edition) C'est une norme proposée par la société Sun, visant à définir un standard de développement d'applications d'entreprises, basées sur des API (Application Programming Interface).

Les API de J2EE sont des outils logiciels constitués d'un ensemble de modules dotés de fonctions différentes et peuvent se répartir en deux grandes catégories : les services et les composants.

- Les services, sont classés par catégories :
 - Les services d'infrastructures : il en existe un grand nombre, définis ci-dessous :
 - JDBC (Java DataBase Connectivity) est une API d'accès aux bases de données relationnelles.
 - JNDI (Java Naming and Directory Interface) est une API d'accès aux services de nommage et aux annuaires d'entreprises tels que DNS.
 - JCA (J2EE Connector Architecture) est une API de connexion au système d'information de l'entreprise
 - Les services de communication :
 - JAAS (Java Authentication and Authorization Service) est une API de gestion de l'authentification et des droits d'accès.
 - JavaMail est une API permettant l'envoi de courrier électronique.
- Les composants :

On distingue habituellement deux familles de composants :

- **Les composants web** : Servlets et JSP (Java Server Pages). Il s'agit de la partie chargée de l'interface avec l'utilisateur (on parle de logique de présentation).
- **Les composants métier** : EJB (Enterprise Java Beans). Il s'agit de composants spécifiques chargés des traitements des données propres à un secteur d'activité (on parle de logique métier ou de logique applicative) et de l'interfaçage avec les bases de données.

L'architecture J2EE permet ainsi de séparer le projet en des couches telle que :

- La couche présentation, correspondant à l'interface homme-machine (IHM)
- La couche métier contenant l'essentiel des traitements de données en se basant dans la mesure du possible sur des API existantes,
- La couche de données correspondant aux informations de l'entreprise stockées dans des fichiers, dans des bases de données relationnelles ou XML, dans des annuaires d'entreprise ou encore dans des systèmes d'information complexes

Ainsi, on trouve ci-dessous, la figure qui représente cette architecture :

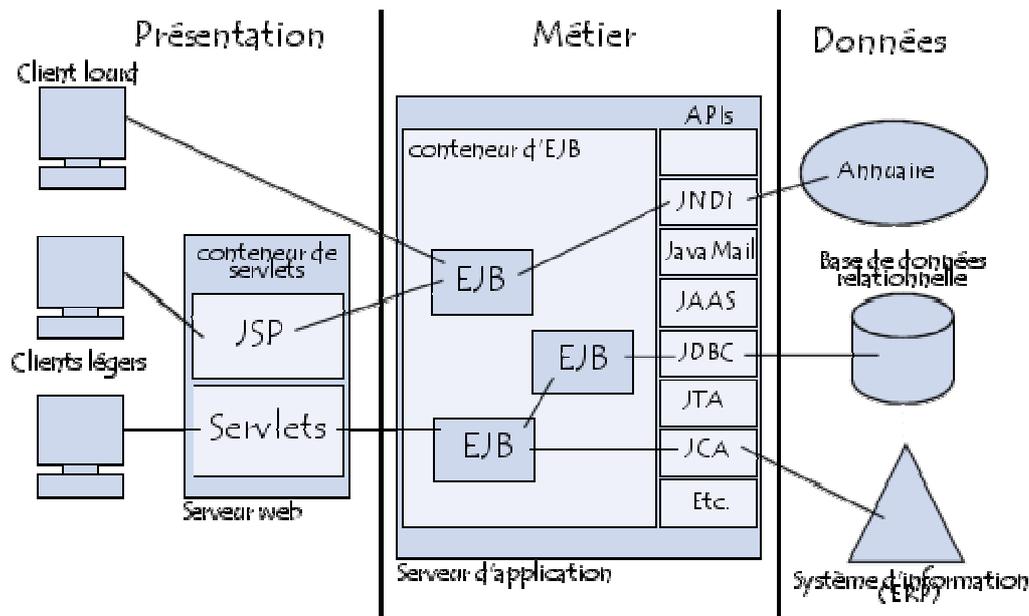


Figure 17 : Architecture J2EE

2. Les Technologies du développement :

Les technologies du développement utilisées sont : les JSP (Java Server Pages), les servlets, et le conteneur des composants web qui gère les deux technologies citées auparavant.

- **JSP (Java Server Pages) :**

La technologie Java Server Page (JSP) permet de mettre des fragments de code java dans une page HTML statique. Elle est Comparable aux langages côtés serveur de type PHP (*Hypertext Preprocessor*), ASP (*Active Server Page*).

Les spécifications JSP définissent un jeu de balises standards implémentées par toutes les implémentations de serveurs web Java. Pour différencier le code html du code java, on utilise plusieurs tags tel que :

- **Tag de directive :**
 - `<%@ ... %>`
- **Tag de commentaire :**
 - `<%-- blabla --%>`
- **Tag de déclaration :**
 - `<%! ... %>`
- **Tag de Scriptlet :**
 - `<% ...%>`
- **Tag d'expression :**
 - `<%= ... %>`

Attention ne pas se tromper dans la nomination des tags. Ils ont tous une signification différente

Éléments de scripts

Les tags de directives contiennent soient :

- `<@ include...>` : pour indiquer au compilateur d'inclure un autre fichier.
- `<@taglib... >` : pour indiquer une bibliothèque de balises à utiliser.
- `<@ page ...>` : pour définir les attributs spécifiques à une page.

Les tags de commentaire servent pour ajouter des commentaires.

Les tags de déclaration servent pour déclarer les différents attributs.

Les tags de scriptlet ou on met le code java tel que les boucles for, les if et les else.

Les tags d'expression servent à évaluer une expression et à renvoyer sa valeur, en effet il correspond à une scriptlet comme `<% out.println(...); %>`.

Le fonctionnel des JSP :

Les pages JSP sont converties en Servlet par le moteur de Servlets lors du premier appel à la JSP, comme le montre le schéma explicatif du fonctionnement de la technologie JSP :

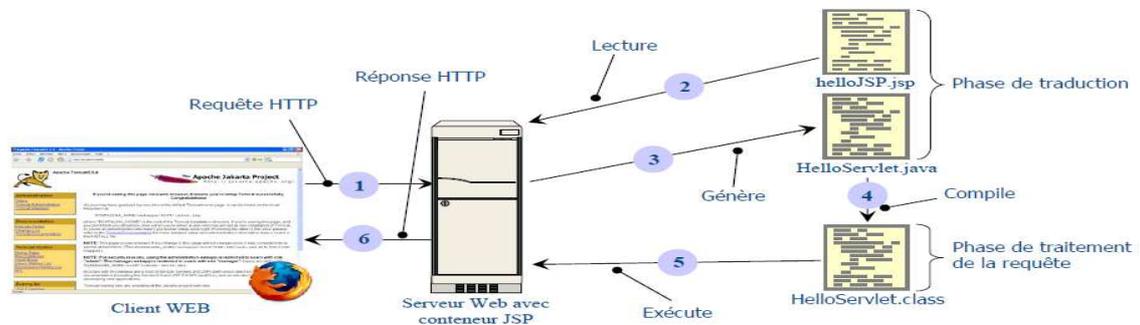


Figure18 : Gestion des pages JSP

Lorsque la page JSP est chargée par un navigateur web, le code java est exécuté sur le serveur. Celui-ci crée une servlet correspondante, qui est ensuite compilée et exécutée en tâche de fond.

La servlet retourne une page HTML ou un rapport en XML qui peut alors être transmis au client ou subir d'autres traitements.

Les spécifications JSP définissent un jeu de balises standards implémentées par toutes les implémentations de serveurs web Java.

- **Servlet** : La servlet est un composant web conçu sous forme d'une classe java qui existe au sein d'une application web et dont la mise en œuvre est générée par un conteneur web du côté du serveur web.

Une servlet interagit avec un client web par l'intermédiaire d'un protocole http, via un mécanisme de requêtes/réponses.

Les servlets sont définies dans les packages suivants:

- javax.servlet, contient les classes génériques (indépendantes du protocole) des servlets. La classe HttpServlet utilise la classe ServletException de ce package pour indiquer un problème de servlet.
- javax.servlet. Http, contient la classe de servlet conçue pour le protocole HTTP (classe HttpServlet).
- En général les servlets utilisent aussi le package java.io pour les entrées/sorties du système. La classe HttpServlet utilise la classe IOException de ce package pour signaler les erreurs d'entrée-sortie.

- **Conteneur web :**

L'environnement d'exécution d'une application web est appelé le conteneur web (web container).

Le conteneur web gère le cycle de vie des composants web et distribue les requêtes aux composants (servlet, JSP).

Il fournit aussi des interfaces standards pour accéder au contexte, notamment aux informations de session et aux informations sur la requête courante.

Le conteneur web fournit une interface consistante aux composants, indépendamment du serveur web. C'est ce qui permet de porter et déployer un composant sur différents serveurs web Java sans modification du code.

- **Cycle de vie d'une servlet java au sein d'un conteneur web :**

Le cycle de vie d'une servlet java au sein d'un conteneur web correspond aux tâches suivantes:

1. La classe java représentant la servlet est déployée au sein du conteneur web.
2. Le conteneur web crée une instance de la classe de servlet et la charge en mémoire. Cette opération s'effectue, selon la configuration de l'application web, soit lors du démarrage du conteneur web, soit lors de la première invocation de la servlet par l'envoi d'une requête client.
3. Le conteneur web passe ensuite à la phase d'initialisation de la servlet en invoquant la méthode `init` (servlet config), L'implémentation de cette méthode permet par l'intermédiaire de l'objet servlet config de récupérer les paramètres d'initialisation de la servlet, définis dans le descripteur de déploiement (web.xml) dans l'application web.
4. La servlet passe ensuite dans un état dit « en service » ou plus précisément en attente de réception des requêtes clients. La réception de requêtes est gérée par le conteneur web qui associe un thread à chaque requête client. Le conteneur web gère ensuite une file d'attente de threads de la servlet dans un contexte multitâche .Chaque thread

5. invoque la méthode service (...request, response) de la servlet, qui s'occupe ensuite d'exécuter la méthode http correspondante (doPost, doGet)
6. L'instance de la servlet reste chargée en mémoire tant que le conteneur web s'exécute. Lors de l'arrêt du conteneur web, la méthode destroy () est invoquée sur l'instance de la servlet indiquant qu'elle n'est plus en service.
7. Et pour finir, comme une servlet est un objet java, son instance est supprimée de la mémoire.

II. Les langages de développement :

Au cours de ce projet, nous avons développé avec plusieurs langages, parmi lesquels on cite :

- **Java :**

Le langage Java est un langage de programmation informatique orienté objet, présenté officiellement le 23 mai 1995 au SunWorld.

Il ne faut pas le confondre avec JavaScript (petit langage de scripts utilisé sur les sites web). Une de ses plus grandes forces est son excellente portabilité car une fois le programme est créé, il fonctionnera automatiquement sous Windows, Mac, Linux, etc.

- On peut faire de nombreuses sortes de programmes avec Java telles que :
 - des applications, sous forme de fenêtre ou de console ;
 - des applets, qui sont des programmes Java incorporés à des pages web ;
 - des applications pour appareils mobiles, avec J2ME ;
 - et bien d'autres tel que J2EE
- **JavaScript :** JavaScript est un langage de programmation de scripts qui incorporé aux balises HTML. Il permet d'améliorer la présentation et l'interactivité des pages Web. Il présente un langage de script coté client. Il permet d'étendre les capacités du langage. Ces scripts vont être gérés et exécutés par le browser lui-même sans devoir faire appel aux ressources du serveur

- **Dans une page Web :**

Du code JavaScript peut être intégré directement au sein des pages Web, pour y être exécuté sur le poste client. C'est alors le navigateur Web qui prend en charge l'exécution de ces programmes appelés scripts. Généralement, JavaScript sert à contrôler les données saisies dans des formulaires HTML, ou à interagir avec le document HTML via l'interface Document Object Model, fournie par le navigateur (on parle alors parfois de HTML dynamique ou DHTML).

- **XML : Extensible Markup Language**

Extensible Markup Language (XML) (français : « langage extensible de balisage ») est un langage informatique de balisage générique. Il sert essentiellement à stocker et transférer des données de type texte structurées en champs. Ce langage est qualifié d'extensible car il permet à l'utilisateur de définir les balises des éléments.

- **CSS (Cascading Style Sheets (feuilles de styles en cascade):**

Les CSS servent à mettre en forme des documents web de type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle. En effet, On écrit le code CSS dans un fichier à part (à l'extension .css au lieu de .html). C'est un fichier dans lequel on écrit l'apparence ce que notre site doit avoir : la couleur et la police du texte, la taille des titres, la position des menus, la couleur ou l'image de fond etc.

Ceci permet de:

- Ne pas répéter dans chaque page le même code de mise en forme.
- Utiliser des styles génériques, avec des noms explicites (par exemple un style encadré pour du texte ou des images).

- Pouvoir changer l'apparence d'un site web complet en ne modifiant qu'un seul fichier.
- Faciliter la lecture du code de la page.

Ainsi, les technologies et les langages utilisés sont exploités et configurées par la suite dans des outils de développement tel qu'ECLIPSE et TOMCAT.

III. outils de développement :

1. Eclipse :

Eclipse est un environnement de développement intégré ,libre ,extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre le langage de programmation JAVA .

La spécificité d'Eclipse IDE vient du fait de que son architecture est totalement développée autour de la notion de plugin. En effet, par exemple : pour pouvoir lire un fichier XML, il faut ajouter le plugin XML au plugin d'Eclipse, pour pouvoir piloter Tomcat à partir de Eclipse, on doit ajouter le plugin de Tomcat.

2. Apache Tomcat :

Apache Tomcat est un conteneur libre de servlet Java 2 EE. Tomcat est un projet principal de la fondation Apache. Tomcat est souvent utilisé en association avec un autre serveur web, en général Apache. Celui ci s'occupe de toutes les pages web traditionnelles (pages HTML), et Tomcat prend en charge uniquement des pages d'une application web Java (pages JSP). Il est paramétrable par des fichiers XML.

Tomcat a été écrit en langage Java, et peut donc s'exécuter via la JVM (machine virtuelle java) sur n'importe quel système d'exploitation. Il existe plusieurs versions de tomcat, la version utilisée dans ce projet est : tomcat 6.0 qui est incluse dans la famille tomcat 6.x. Cette famille est caractérisée par :

--Implémentation des spécifications servlet 2.5et jsp 2.0. --Support Java 5 ;

Chapitre 8 : Interfaces de réalisation

L'objectif principal que vise chaque développeur d'une application web est d'arriver à une solution qui permet de résoudre la problématique proposée par la société d'accueil. Pour notre cas, la solution atteinte est présentée sous forme d'interfaces de réalisation suivantes.

- **Page d'accueil :**

Cette page présente la page d'accueil de l'application web qui sert à générer des scripts SQL. Ces derniers, une fois configurés dans le progiciel BSCS, permettent de créer les offres commerciales de Maroc Telecom.

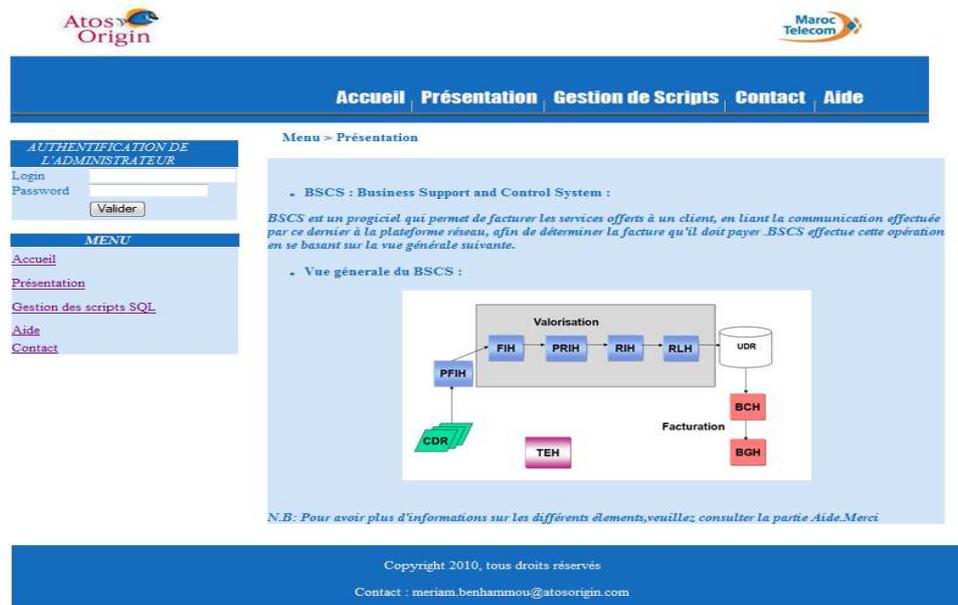
Le menu est divisé en deux parties :

- Partie de l'authentification de l'administrateur qui contient un login et un password à saisir.
- Partie contenant les différents onglets de l'application tel que : accueil, présentation, gestion des scripts SQL, aide et contact.



- **Page de présentation :**

Cette page présente le progiciel BSCS dans lequel les scripts SQL seront configurés une fois développés, ainsi qu'elle montre les différents modules qui interviennent dans ce processus



BSCS : Business Support and Control System :

BSCS est un progiciel qui permet de facturer les services offerts à un client, en liant la communication effectuée par ce dernier à la plateforme réseau, afin de déterminer la facture qu'il doit payer. BSCS effectue cette opération en se basant sur la vue générale suivante.

Vue générale du BSCS :

N.B: Pour avoir plus d'informations sur les différents éléments, veuillez consulter la partie Aide. Merci

Copyright 2010, tous droits réservés
Contact : meriam.benhammou@atosorigin.com

- **Page gestion des scripts SQL:**

Cette page permet de gérer les différents scripts SQL qu'on possède tel que la gestion des codes comptables (ajout et modification), la création d'un service en fixe ou en mobile, la modification du libellé d'un service, d'un paramètre ou des deux, la vérification de la compatibilité de deux services et la création d'un plan tarifaire en liaison avec un rating package.



The screenshot shows the Atos Origin web application interface. At the top, there is a navigation bar with links: Accueil, Présentation, Gestion de Scripts, Contact, Aide. Below this, on the left, is an 'AUTHENTIFICATION DE L'ADMINISTRATEUR' section with fields for Login and Password, and a 'Valider' button. Below that is a 'MENU' section with links: Accueil, Présentation, Gestion des scripts SQL, Aide, Contact. The main content area is titled 'Gestion des scripts SQL' and contains several sections:

- Gestion des codes compatibles:**
 - [Ajout d'un code compatible](#)
 - [Modification d'un code compatible](#)
- Création d'un nouveau service :**
 - Fixe :**
 - [EVENT](#)
 - VAS (En cours de réalisation)
 - NETWORK (En cours de réalisation)
 - Mobils :**
 - [EVENT](#)
 - VAS (En cours de réalisation)
 - NETWORK (En cours de réalisation)
- Modification du libellé:**
 - [d'un service](#)
 - [d'un paramètre](#)
 - [d'une combinaison de paramètres](#)
- [Paramétrage de la compatibilité et l'incompatibilité de deux services.](#)
- [Création d'un plan tarifaire en fonction avec Rating Package.](#)

At the bottom of the page, there is a footer with the text: Copyright 2010, tous droits réservés. Contact : meriam.benhammou@atosorigin.com

Parmi les points traités dans cette page, on trouve :

- **[Page modification du libellé d'un service](#) :**

Cette page contient un formulaire qui est validé une fois rempli par un utilisateur professionnel en BSCS. Il est emmené vers une autre page qui contient le script SQL prêt à être généré à l'aide du bouton « Générer le script SQL » et qui permet donc à l'utilisateur de garder une trace de ce script. Ainsi on trouve ci-dessous les pages qui montrent le déroulement de ce processus :

[Accueil](#) | [Présentation](#) | [Gestion de Scripts](#) | [Contact](#) | [Aide](#)

AUTHENTIFICATION DE L'ADMINISTRATEUR

Login:
Password:

Menu > Gestion des scripts SQL > Modification du libellé d'un service

NOM_USER	meriam
SCRIPT_NAME	IAM02072010
REMARK	Modification du libelle
DESCRIPTION	DI
SERVICESHDES	MMS
NEW_SERVICEDES	SMS
<input type="button" value="Initialiser"/> <input type="button" value="Valider"/>	

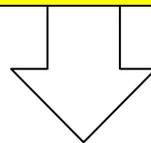
MENU

[Accueil](#)
[Présentation](#)
[Gestion des scripts SQL](#)
[Aide](#)
[Contact](#)

Copyright 2010, tous droits réservés

Contact : meriam.benhamanous@atosorigin.com

Quand l'utilisateur professionnel en BSCS remplit le formulaire et le valide on obtient la page suivante :



[Accueil](#) | [Présentation](#) | [Gestion de Scripts](#) | [Contact](#) | [Aide](#)

AUTHENTIFICATION DE L'ADMINISTRATEUR

Login:
Password:

Menu > Gestion des scripts SQL > Modification du libellé d'un service

NOM_USER	meriam
SCRIPT_NAME	IAM02072010
REMARK	Modification du libelle
DESCRIPTION	DI
SERVICESHDES	MMS
NEW_SERVICEDES	SMS
<input type="button" value="Initialiser"/> <input type="button" value="Valider"/>	

Message de la page Web

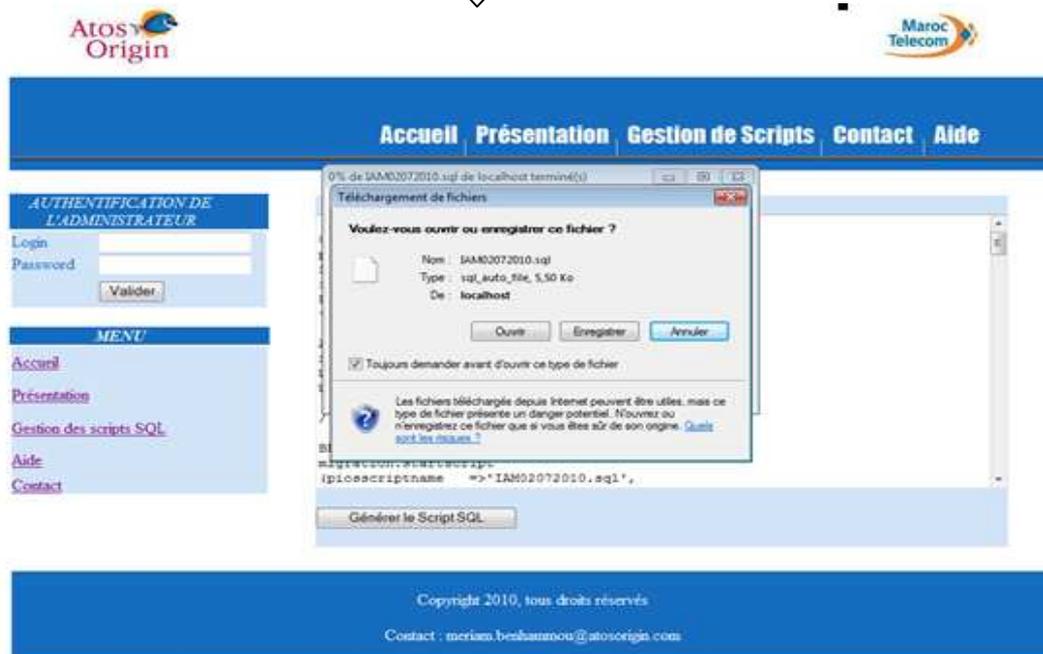
Veillez confirmer ce que vous avez saisi ou appuyer sur annuler!

NOM_USER : meriam
SCRIPT_NAME : IAM02072010
REMARK : Modification du libelle
DESCRIPTION : DI
SERVICESHDES : MMS
NEW_SERVICEDES : SMS

Copyright 2010, tous droits réservés

Contact : meriam.benhamanous@atosorigin.com

Une fois l'utilisateur clique sur ok, il est dirigé vers la page où se trouve le script SQL et il peut le générer



Si l'utilisateur clique sur « annuler » il va rester dans la même page d'où il aura la page suivante:

Accueil | Présentation | Gestion de Scripts | Contact | Aide

AUTHENTIFICATION DE L'ADMINISTRATEUR

Login:

Password:

MENU

[Accueil](#)

[Présentation](#)

[Gestion des scripts SQL](#)

[Aide](#)

[Contact](#)

```

/*
Created By : meriam
project : Maroc Telecom
Script Name : IAM02072010.sql
remarks : Modification du libelle
Description : DI
*/

ALTER SESSION SET nls_date_format='DD-MON-YYYY';
SET SERVEROUTPUT ON SIZE 999999
WHENEVER SQLERROR EXIT 1 ROLLBACK;
WHENEVER OSERROR EXIT 2 ROLLBACK;

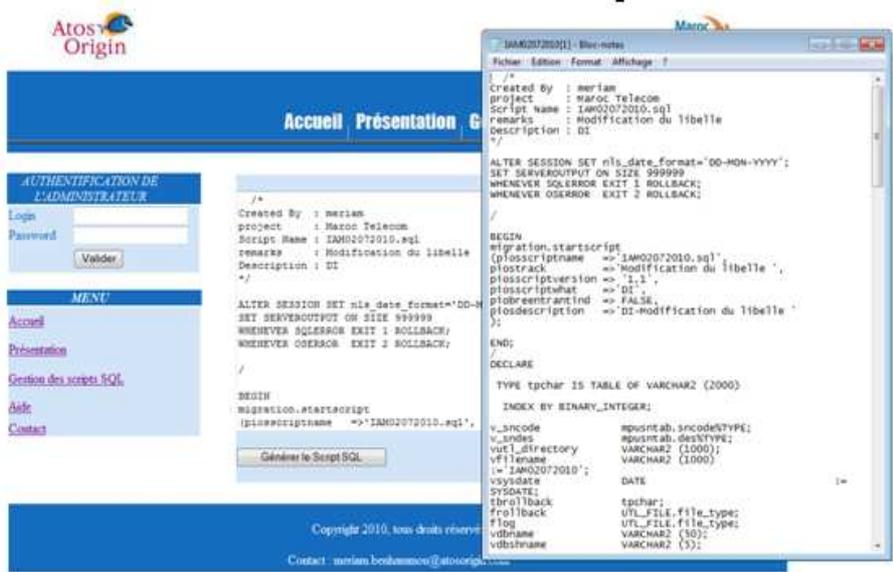
/

BEGIN
migration.startscript
(piosscriptname =>'IAM02072010.sql',

```

Copyright 2010, tous droits réservés
 Contact : meriam.benhammou@atosorigin.com

Si l'utilisateur clique sur « ouvrir » ou « enregistrer », il aura la page suivante :



The screenshot shows the application interface with the 'Générer le Script SQL' button clicked. A Notepad window titled 'IAM02072010 - Bloc-notes' displays the following SQL script:

```

/*
Created By : meriam
project : Maroc Telecom
Script Name : IAM02072010.sql
remarks : Modification du libelle
Description : DI
*/

ALTER SESSION SET nls_date_format='DD-MON-YYYY';
SET SERVEROUTPUT ON SIZE 999999
WHENEVER SQLERROR EXIT 1 ROLLBACK;
WHENEVER OSERROR EXIT 2 ROLLBACK;

/

BEGIN
migration.startscript
(piosscriptname =>'IAM02072010.sql',
piosscriptversion =>'1.1',
piosscriptwhat =>'DI',
piosentrained =>FALSE,
piosdescription =>'DI-Modification du libelle'
);

END;

DECLARE
TYPE tpchar IS TABLE OF VARCHAR2 (2000);
INDEX BY BINARY_INTEGER;

v_sncode          iposintab.sncode%TYPE;
v_sndes          iposintab.des%TYPE;
vutl_directory   VARCHAR2 (1000);
vfilename        VARCHAR2 (1000);
v:=IAM02072010';
vsysdate         DATE;
vsysdate         SYSDATE;
vrollback        tpchar;
vflag            UTL_FILE.file_type;
vdbname          VARCHAR2 (50);
vdbshname        VARCHAR2 (5);

```

- **Page de vérification de la compatibilité ou non entre deux services :**

Cet exemple est différent du premier du fait que les champs de son formulaire sont différents, en effet :



The screenshot shows the Atos Origin web application interface. At the top left is the Atos Origin logo, and at the top right is the Maroc Telecom logo. A blue navigation bar contains the links: Accueil, Présentation, Gestion de Scripts, Contact, Aide. Below this, there is a section for 'AUTHENTIFICATION DE L'ADMINISTRATEUR' with fields for 'Login' and 'Password' and a 'Valider' button. To the right, a breadcrumb trail reads 'Menu > Gestion des scripts SQL > Paramétrage de la compatibilité et de l'incompatibilité de deux services'. The main form area contains:

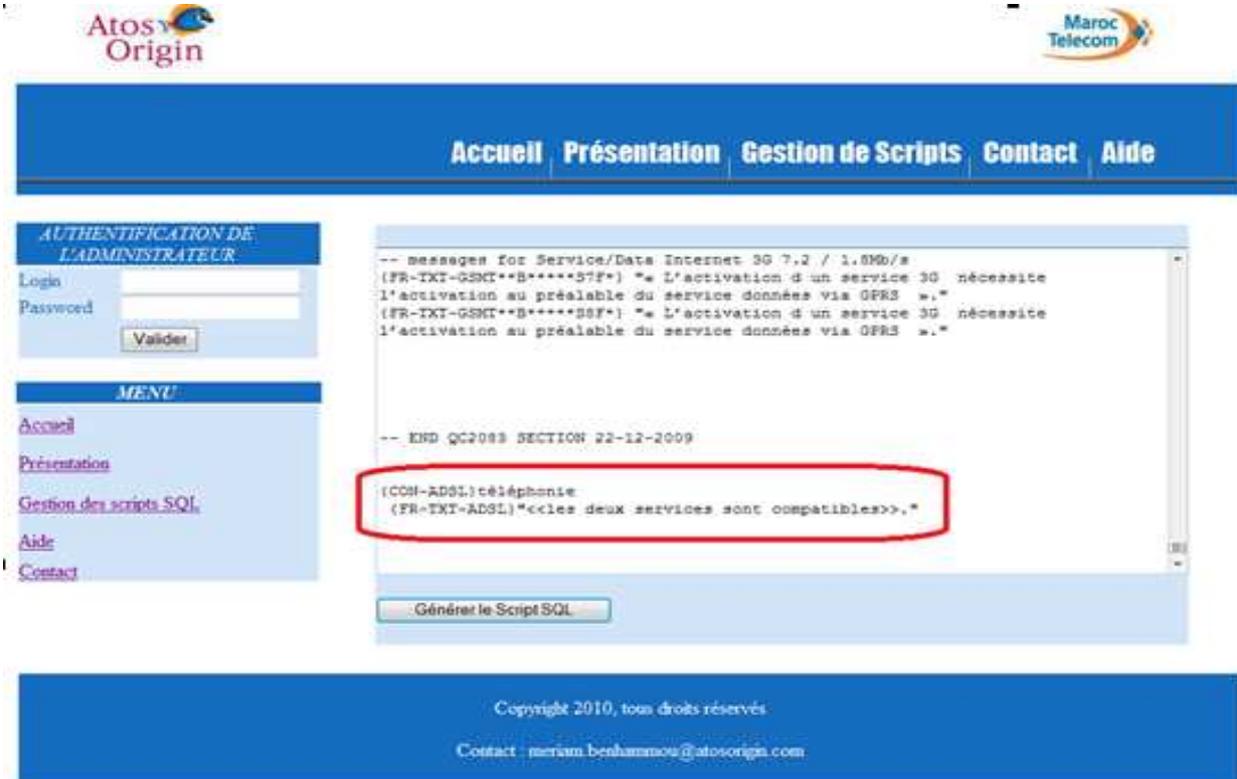
- Service1: ADSL
- Service2: Téléphone
- Compatible: Oui Non
- Description du message: les deux services sont c
- A button labeled 'Ouverture et insertion dans Ruletab'

 At the bottom of the page, there is a footer with 'Copyright 2010, tous droits réservés.' and 'Contact : meriam.benhamou@atosorigin.com'.

Quand l'utilisateur clique sur « ouverture et insertion dans Ruletab », deux lignes de code s'ajoutent :

```
{CON-service1} service2
{FR-TXT-service1} " << Description du message >>."
```

Il aura la page suivante :



- **Gestion de l'authentification de l'administrateur :**

Quand l'utilisateur s'authentifie :

- Soit son password est incorrecte et dans ce cas, on aura la page suivante :



- Soit son password est correct, et dans ce cas la page suivante s'affiche:



Script	Status
Ajout d'un code comptable	<input type="radio"/> A <input type="radio"/> I
Modification d'un code comptable	<input checked="" type="radio"/> A <input type="radio"/> I
Création d'un service fixe EVENT	<input checked="" type="radio"/> A <input type="radio"/> I
Création d'un service fixe VAS	<input type="radio"/> A <input type="radio"/> I
Création d'un service fixe NETWORK	<input type="radio"/> A <input type="radio"/> I
Création d'un service mobile EVENT	<input checked="" type="radio"/> A <input type="radio"/> I
Création d'un service mobile VAS	<input type="radio"/> A <input type="radio"/> I
Création d'un service mobile NETWORK	<input type="radio"/> A <input type="radio"/> I
Modification du libellé d'un service	<input checked="" type="radio"/> A <input type="radio"/> I
Modification du libellé d'un paramètre	<input checked="" type="radio"/> A <input type="radio"/> I
Modification du libellé d'une combinaison de paramètres	<input checked="" type="radio"/> A <input type="radio"/> I
Paramétrage de la comptabilité et de l'incomptabilité de deux services	<input type="radio"/> A <input type="radio"/> I
Création d'un plan tarifaire en liaison avec un rating package	<input checked="" type="radio"/> A <input type="radio"/> I

On remarque que sur cette page chaque script, peut avoir deux status soit actif ou inactif. Lorsque le statut d'un script est actif par l'administrateur, l'utilisateur peut y accéder et générer le script SQL selon son besoin, Par contre, quand l'administrateur a rendu le statut du script inactif on aura la page suivante :



- Page aide :

Cette page contient des éléments d'aide pour les différents points traités dans la page gestion des scripts SQL :



<ul style="list-style-type: none"> 1: Code comptable 2: Création d'un service 3: Modification du libellé d'un service 4: Modification du libellé d'un paramètre 5: Modification du libellé d'une combinaison de paramètres 6: Paramétrage de la compatibilité de deux services 7: Création d'un plan tarifaire en liaison avec un rating package 8: Vue générale BSCS <p>Retour à la page d'accueil</p>	<table border="0"> <tr> <td>NOM_USER</td> <td>Nom de l'utilisateur</td> </tr> <tr> <td>FILENAME</td> <td>Nom du script à saisir sans extension .sql</td> </tr> <tr> <td>REMARK</td> <td>Commentaire sur le script</td> </tr> <tr> <td>DESCRIPTION</td> <td>Objectif du script : FC, DI, anomalie</td> </tr> <tr> <td>GL_CODE</td> <td>Code comptable</td> </tr> <tr> <td>GLCODE_DESC</td> <td>Description du code comptable</td> </tr> <tr> <td>GLA_TYPE</td> <td>Type du code comptable : 1=asset, 2=Li, 4=Inc, 8=Exp</td> </tr> <tr> <td>GLENTYDATE</td> <td>Date effet du code comptable</td> </tr> </table>	NOM_USER	Nom de l'utilisateur	FILENAME	Nom du script à saisir sans extension .sql	REMARK	Commentaire sur le script	DESCRIPTION	Objectif du script : FC, DI, anomalie	GL_CODE	Code comptable	GLCODE_DESC	Description du code comptable	GLA_TYPE	Type du code comptable : 1=asset, 2=Li, 4=Inc, 8=Exp	GLENTYDATE	Date effet du code comptable
NOM_USER	Nom de l'utilisateur																
FILENAME	Nom du script à saisir sans extension .sql																
REMARK	Commentaire sur le script																
DESCRIPTION	Objectif du script : FC, DI, anomalie																
GL_CODE	Code comptable																
GLCODE_DESC	Description du code comptable																
GLA_TYPE	Type du code comptable : 1=asset, 2=Li, 4=Inc, 8=Exp																
GLENTYDATE	Date effet du code comptable																

- **Page contact :**

Cette page comporte le contact du personnel de la page d'accueil et le contact de l'administrateur du site web.



Dans ce qui précède nous avons, présenté les maquettes de l'application web développée, mais puisqu'une application web est une application évolutive, elle peut être mise-à-jour selon le besoin du client d'où la nécessité des perspectives avancées dans la partie suivante.

Conclusion et perspectives

Déjà annoncé au début, le travail réalisé lors de ce stage porte sur une application web dédiée au professionnel du progiciel BSCS et dont le but est d'augmenter le taux de la production des offres commerciales de Maroc Telecom tout en gagnant en coût, en temps et en qualité de travail.

Une application web étant une application souvent en évolution, elle peut donc subir à chaque instant des modifications. En effet, concernant cette application, on peut ajouter des contrôles de saisie sur les différents champs des formulaires utilisés, ajouter de nouvelles pages JSP qui permettent de gérer d'autres points que ceux cités dans ce projet selon le besoin de Maroc Telecom.

L'application web développée dans ce projet a bien répondu au cahier des charges de la société selon les estimations des ingénieurs d'ATOS ORIGIN qui ont approuvé cette application. Vue l'importance de cette application, il y'a eu des promesses qu'elle sera mise sur le plan réel à partir du mois ramadan ou règne une variété des offres commerciales dans le domaine des télécommunications.

Bibliographie et webiographie

Documents livrés par la société relative au progiciel BSCS :

- Modules BSCS.
- Chaîne de valorisation
- Chaîne de facturation.

JSP :

<http://tecfa.unige.ch/guides/tie/html/java-jsp/java-jsp-2.html> :

Servlet :

<http://www.dil.univ-mrs.fr/~massat/ens/jee/servlet.html>

Tomcat

http://fr.wikipedia.org/wiki/Apache_Tomcat

Eclipse

[http://fr.wikipedia.org/wiki/Environnement de d%C3%A9veloppement int%C3%A9gr%C3%A9](http://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9)

<http://www.ecliptotale.com/tomcatPlugin.html#A3>

UML :

- ❖ UML EN ACTION DE L'ANALYSE DES BESOINS A LA CONCEPTION EN JAVA :

<http://www.editions-eyrolles.com/Chapitres/9782212112139/chap2.pdf> chapitre 2 : processus et architectures.

MVC

<http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>

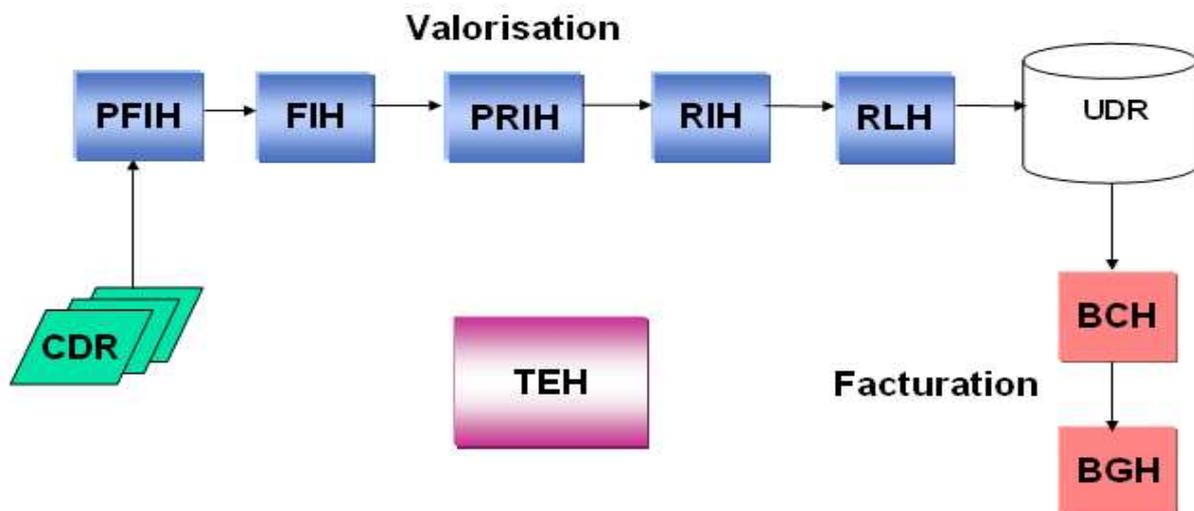
<http://fr.wikipedia.org/wiki/Mod%C3%A8le-Vue-Contr%C3%B4leur>

En réseau local :

- Pour accéder à l'application, tapez l'url suivant :
<http://10.0.62.226:8080/GestionDesOffresCommerciales/>
- Si vous voulez plus d'informations, tapez l'url du manuel d'utilisation de l'application suivant : [http://10.0.62.146/mediawiki/index.php/Gestion des offres commerciales](http://10.0.62.146/mediawiki/index.php/Gestion_des_offres_commerciales)

Annexe : Vue générale du progiciel BSCS

La vue générale des modules du BSCS est basée sur la figure suivante :



Cette vue est composée de plusieurs modules gérés par deux chaînes tel que :

- **CHAINE DE VALORISATION :**

Les fonctionnalités essentielles de la valorisation sont :

- Gérer les données Client/Contrat/Ressource/Service
- Recevoir du Réseau Technique les CDR (Call Detail Record) des Clients
- Associer les CDR au Client/Contrat/Ressource/Service approprié
- Calculer le Coût du CDR en fonction des données:
Client/Contrat/Ressource/Service
- Gérer la Persistance des CDR traités.

Ainsi, la valorisation des communications est réalisée par des modules Batch qui tournent sur le serveur UNIX. Ces batchs échangent des informations à travers un bus

de communication: Data. Ces modules sont dans l'ordre de traitement : TEH, PFIH, FIH, PRIH, RIH, RLH

○ **TEH (Table Extraction Handler) :**

Extrait les données 'Client/Contrat/Ressource/Service' de la base de données relationnelle dans des fichiers.

○ **PFIH (Pré File Input Handler) :**

PFIH effectue plusieurs tâches tel que:

- PFIH reçoit les tickets(CDR) collectés par CADOT;
- PFIH transforme les tickets CADOT en CSV ASCII;
- PFIH ajoute aux tickets des informations nécessaires pour la valorisation;
- PFIH enregistre les tickets traités dans THUFITAB.

N.B : un ticket d'appel (ou CDR) est un ticket qui arrive du réseau (du Switch, ou commutateur) et qui est traité par plusieurs batches, chacun d'eux effectuant une opération visant à modifier le format du ticket arrivant, ou enrichir son contenu. Le résultat final est ensuite stocké dans une base de données, pour la facturation.

○ **FIH (File Input Handler)**

FIH sert à transformer les fichiers de CDR en un format spécifique à BSCS. Une fois lancé, FIH recherche la présence des fichiers signalés à traiter par 'PFIH' dans la table THUFITAB. FIH effectue un pré-traitement de ces fichiers dont le but est d'éliminer les fichiers qui auraient déjà été traités. Cette opération est basée sur l'heure de début et de fin de communication correspondant respectivement au premier et au dernier ticket de CDR du fichier. Si le fichier n'a jamais été traité, il va être transformé par FIH au format « UDR » spécifique à BSCS.

○ **PRIH (Pré Rate Input Handler):**

PRIH analyse chaque communication figurant dans l'UDR pour déterminer à quel type de scénario il correspond (quel marché, partie impactées,...). L'UDR est enrichi d'information (information client...).

Chaque appel traité par PRIH doit correspondre à un appel effectué dans le cadre d'un scénario d'appel identifié, par un client clairement identifié.

Il existe plusieurs raisons pour lesquelles un appel pourrait être rejeté. Ces appels sont placés dans un répertoire de recyclage, en attendant que la cause du rejet soit corrigée. A l'issue de la correction, il devient possible de retraiter ces appels.

- **RIH (Rate Input Handler):**

Après l'identification des parties impactées (appelant/appelé) d'un appel par PRIH, RIH identifie les parties chargeables (qui va payer la communication?).

RIH calcule le montant de l'appel après identification du:

Plan tarifaire/services.

Rating package.

- **RLH (Rate Load Handler) :**

RLH (Rate Load Handler) est un processus permanent qui permet de charger les appels (qui existent encore sous forme de fichiers sur le serveur Unix) dans la base de données UDR (base de données qui contient tous les appels devant être facturés).

- **CHAINE DE FACTURATION :**

Les principaux modules de la chaîne de facturation sont:

BCH (Bill Cycle Handler): rassemble toutes les informations Client, Contrat, coûts d'accès, et génère un ensemble de 'Données-Facture' dans les messages TIMM (Telecommunication Invoice for Multiple Markets) pour chaque Client Responsable de Paiement.

BGH (Bill Generation Handler): Lit les messages TIMM créés par BCH et les convertit en un format spécifique à l'opérateur pour traitement externe (ex. HTML, DOC1, ...)

Ainsi, on aura comme résultat la facture créée que le client doit payer et qui contient 3 parties :

- un récapitulatif global (montant à payer, montant de la facture)
- un détail contrat par contrat.
- une facture détaillée (le cas échéant).