

# Remerciement

Il nous est particulièrement agréable avant de présenter ce travail, d'exprimer toute notre gratitude envers les personnes qui, de près ou de loin, nous ont apporté leur aide inestimable lors de la réalisation de ce projet. Nos remerciements s'adressent particulièrement à :

- Monsieur **Behjet BOUSSOFARA**, Directeur Général de Talan Tunisie Consulting, pour nous avoir prodigué l'honneur de travailler dans son équipe.
- Notre encadrante Mademoiselle **Sabrine CHERIF**, ingénieur d'études et développement chez Talan, pour son suivi, ses remarques constructives et la qualité d'encadrement dont elle nous a fait bénéficier.
- Notre encadrante académique Madame **Emna SOUISSI**, pour avoir accepté de superviser ce travail et pour sa disponibilité, son soutien inépuisable et aides précieuses tout au long de ce stage.
- Tous les enseignants qui ont participé à notre évolution scientifique durant les trois années écoulées de formation à l'**ENSIT**.
- Notre dernier mot s'adresse à tous les membres du jury pour l'honneur qu'ils nous font de participer à l'examen de notre travail.

## ***Table des matières***

Introduction générale.....	1
Chapitre 1 : Cadre général du projet.....	3
1. Présentation de l'organisme d'accueil.....	3
1.1. Présentation de l'entreprise .....	3
1.2. Secteurs d'activités.....	3
2. Contexte du projet .....	5
2.1. Problématique.....	5
2.2. Solution proposée .....	6
3. Méthodologie et formalisme adoptés .....	6
3.1. Méthodologie.....	6
3.2. Formalisme Utilisé .....	7
Chapitre 2 : Etude préalable .....	8
1. Système de facturation téléphonique BSCS .....	8
1.1. Présentation .....	8
1.2. Les Composantes de BSCS .....	9
2. Analyse des données .....	13
2.1. Présentation .....	13
2.2. Analyse multidimensionnelle .....	14
3. Reporting .....	16
3.1. Définition.....	16
3.2. Caractéristiques .....	16
Chapitre 3 : Analyse et spécification des besoins.....	17
1. Identification des acteurs.....	17
2. Identification des besoins .....	17
2.1. Besoins fonctionnels.....	17
2.2. Besoins non fonctionnels.....	18
3. Diagramme des cas d'utilisation .....	18
3.1 Diagramme des cas d'utilisation général.....	18
3.2. Cas d'utilisation se connecter à environnement .....	19
3.3. Cas d'utilisation extraire données .....	20
3.4. Cas d'utilisation analyser données .....	22
Chapitre 4 : Etude Conceptuelle.....	24
1. Conception globale.....	24

1.1 Architecture physique.....	24
1.2. Architecture Logicielle.....	25
2. Conception détaillée.....	26
2.1. Diagramme de modèles.....	26
2.2. Diagrammes de classes.....	28
2.3. Diagrammes de séquence.....	33
Chapitre 5 : Réalisation.....	37
1. Environnement de travail.....	37
1.1. Environnement matériel.....	37
1.2. Environnement logiciel.....	37
2. Choix Technologiques.....	38
2.1. JSF/Primefaces.....	38
2.2. Google Chart API.....	39
2.3. Spring.....	39
2.4. Dynamic Reports.....	39
2.5. EclipseLink/JPA.....	39
3. Scénarios d'exécution.....	40
3.1 Connexion à l'environnement.....	40
3.2. Extraction des données BSCS.....	41
3.3. Analyse des données BSCS.....	44
Conclusion Générale.....	46

## ***Table des figures***

Figure 1 Les métiers de Talan .....	4
Figure 2 Les secteurs de services de Talan .....	4
Figure 3 Processus de développement 2TUP [3].....	6
Figure 4 Structure du plan tarifaire .....	9
Figure 5 Structure du rating package .....	12
Figure 6 Opération slice [5] .....	15
Figure 7 Opération dice [5].....	15
Figure 8 Diagramme des cas d'utilisation général .....	19
Figure 9 Cas d'utilisation se connecter à l'environnement .....	19
Figure 10 Cas d'utilisation extraction des données.....	21
Figure 11 Cas d'utilisation analyser les données.....	22
Figure 12 Architecture 3 Tiers .....	25
Figure 13 Architecture logicielle.....	25
Figure 14 Diagramme de modèles .....	27
Figure 15 Diagramme de classe du package Entities .....	29
Figure 16 Diagramme de classe du Package DAO .....	30
Figure 17 Diagramme de classe du package DTO .....	31
Figure 18 Diagramme de classe du package Services .....	32
Figure 19 Diagramme de séquence de l'extraction des données .....	34
Figure 20 Diagramme de séquence de l'analyse des données.....	35
Figure 21 Interface de connexion à l'environnement d'extraction.....	40
Figure 22 Menu extraction des données BSCS.....	41
Figure 23 Aperçu d'application de filtre .....	41
Figure 24 Liaison entre notions BSCS .....	42
Figure 25 Confirmation des choix.....	42
Figure 26 Résultats de l'opération d'extraction .....	43
Figure 27 Sauvegarde des résultats d'extraction .....	43
Figure 28 Choix des dimensions d'analyse .....	44
Figure 29 Aperçu résultat d'analyse.....	44
Figure 30 Aperçu rapport analytique .....	45
Figure 31 Diagramme de classe des entités de services .....	49
Figure 32 Diagramme de classe des entités des free units .....	50
Figure 33 Diagramme de classe des entités des promotions.....	50
Figure 34 Diagramme de classe des entités des rating packages .....	51

# Introduction générale

Au cours de ces dernières années, le monde des télécommunications a vécu des mutations nombreuses et profondes : évolutions technologiques, ouverture à la concurrence, nouvelles stratégies économiques et des nouvelles habitudes des consommateurs.

Dans ce contexte marqué d'une forte rivalité, le rôle d'un opérateur de télécommunication ne se limite plus à l'établissement d'un réseau fiable et le garantit d'une communication sûre. Ces opérateurs se livrent à la concurrence en essayant d'offrir les meilleurs prix et de se mettre au point de nouveaux services et de nouvelles technologies qui répondent le mieux aux besoins des utilisateurs.

Cette compétition ne serait possible qu'avec des systèmes de facturation fiables qui offrent une large gamme de fonctionnalités et qui garantissent le bon déroulement des différents processus de suivi d'abonnés, de relations clientèles, de valorisation et de facturation.

Différents systèmes de facturation existent sur le marché, dont le BSCS : Business Support and Control System, qui est la tendance actuelle de nombreux opérateurs dans le monde et qui présente une large gamme de services métiers répondant à la plupart des besoins des opérateurs.

C'est dans ce cadre que s'inscrit notre projet de fin d'étude qui consiste à développer une solution permettant la génération d'un catalogue des offres, regroupant différentes composantes de ce système de facturation.

Le présent rapport est de ce fait la synthèse des étapes de mise en œuvre de cette solution. Il a pour but de situer le contexte du projet, de décrire son sujet, les méthodes et outils utilisés ainsi que les résultats obtenus. Il est organisé comme suit :

Le premier chapitre sera consacré à la présentation de l'organisme d'accueil et à la mise en contexte du projet et la présentation de la méthodologie et du formalisme adoptés tout au long de l'élaboration de ce projet.

Le second chapitre introduira certains concepts fondamentaux liés au sujet. Ensuite, la spécification et l'analyse des besoins seront présentées au troisième chapitre dans lequel nous étudierons en détail les besoins fonctionnels et non fonctionnels ainsi que la modélisation de ces besoins.

Puis, le quatrième chapitre présentera l'architecture générale de la solution qui mènera à la conception détaillée des différentes fonctionnalités offertes.

Le dernier chapitre décrira la réalisation de l'application et nous présenterons les résultats de notre travail.

Nous clôturons finalement ce rapport par une conclusion générale présentant le bilan de ce projet, ainsi que les perspectives de notre contribution.

# Chapitre 1 : Cadre général du projet

---

## Introduction

Dans ce chapitre, nous allons présenter le contexte général de ce projet. Tout d'abord, nous allons présenter l'organisme d'accueil « Talan Tunisie Consulting ». Ensuite, nous allons introduire le contexte du projet. Puis, nous exposerons la méthodologie et le formalisme adoptés lors de la réalisation du projet.

## 1. Présentation de l'organisme d'accueil

Afin de situer notre travail dans son environnement de réalisation, nous présentons dans cette section l'organisme d'accueil en exposant ses secteurs d'activité.

### 1.1. Présentation de l'entreprise

TALAN est un acteur alternatif aux grandes sociétés de conseil et de services fondé en 2002 par Mehdi Houas, Eric Benamou et Philippe Cassoulat. Il est spécialisé dans le domaine de la refonte des processus métiers et de l'intégration des nouvelles technologies, et concentre son expertise sectorielle sur les grands opérateurs de services (Finance, Télécom, Utilities et Services Publiques) [1].

Talan regroupe plus de 600 consultants en France, en Belgique, en Grande Bretagne, aux USA, à Hong-Kong et en Tunisie et propose un cadre novateur et motivant pour fidéliser ses collaborateurs.

Talan a devancé la tendance de délocalisation des projets de maîtrise d'œuvre ou de Tierce Maintenance Applicative, par l'intégration dès 2007 d'un centre de développement Near-shore en Tunisie, mobilisant à ce jour plus de 120 ingénieurs de développement de nouvelles technologies, travaillant pour les plus grands clients européens [1].

Les compétences de ce centre se partagent entre :

- Le développement Java JEE, Open source et .Net.
- Le développement d'applications Mobile.
- L'intégration de progiciels de Billing et CRM.

### 1.2. Secteurs d'activités

Comme illustré sur la figure 1, l'activité de Talan couvre les prestations suivantes :

- Conseil et Assistance à la Maîtrise d’Ouvrage.
- Systèmes d’Information et Intégration des nouvelles technologies.
- Support des Systèmes d’Information.
- Alignement des systèmes d’information aux changements d’organisation et d’accompagnement au changement.

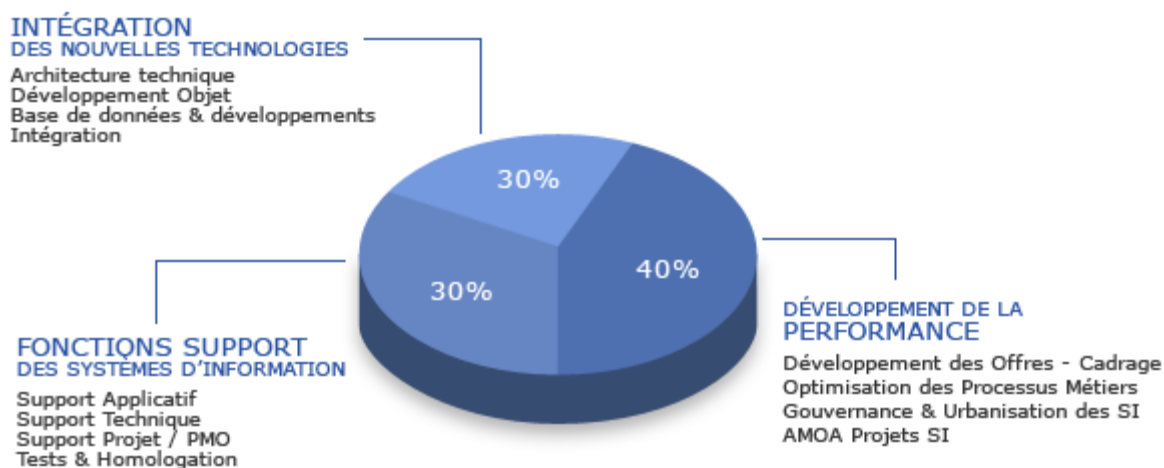


Figure 1 Les métiers de Talan

Ces activités touchent, essentiellement, les secteurs de services (voir figure 2) :

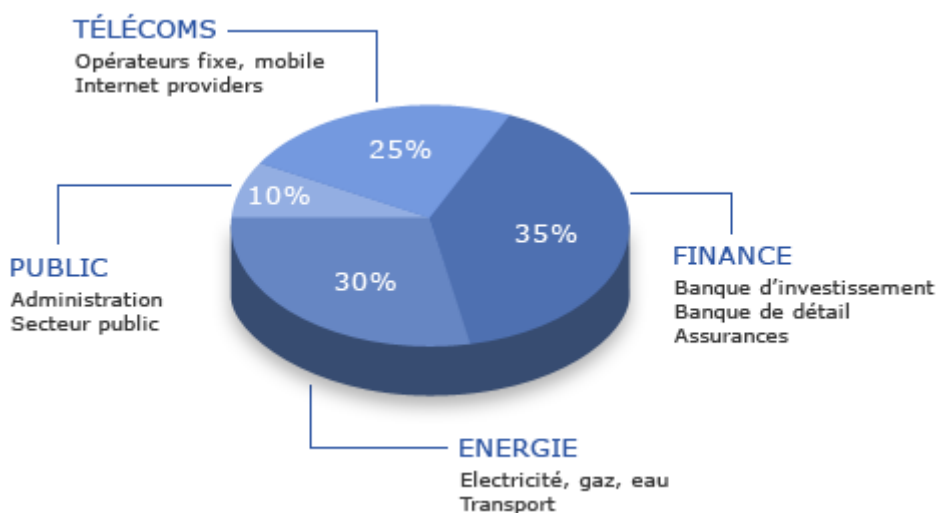


Figure 2 Les secteurs de services de Talan

- Le secteur financier à travers de multiples axes d’amélioration et d’optimisation des métiers de la banque d’investissement, de détail et des assurances.



- Le secteur des télécoms à travers des projets destinés aux opérateurs télécom et un ensemble de fournisseurs d'accès à internet. En Tunisie, ce secteur est géré par l'équipe Billing au sein duquel se déroule notre projet.
- Le secteur public à travers l'assistance à Maître d'Ouvrage et l'intégration de nouvelles technologies dans des projets tels que : e-administration, la réduction des coûts et la cohérence et l'ouverture des systèmes d'information.
- Le secteur d'énergie à travers la ligne de service Energie et Utilities de Talan qui regroupe plus de 80 consultants intervenants sur des projets majeurs de l'AMOA.

## **2. Contexte du projet**

### **2.1. Problématique**

Faisant référence à la section précédente, l'équipe Billing de Talan intervient dans le secteur de télécommunication pour répondre aux besoins des opérateurs télécom en Tunisie et à l'étranger. Ces opérateurs de services, en réflexion et recherche permanente d'excellence opérationnelle, représentent le secteur de prédilection de l'entreprise hôte. Celle-ci les aide à décliner leur démarche sur les processus, à personnaliser des fonctionnalités existantes et à en ajouter d'autres décrivant l'évolution du secteur. Ces opérateurs reposent sur des systèmes de facturation pour gérer les différents processus métier liés à leurs activités, parmi les différents systèmes existants sur le marché, nous trouvons le système de facturation et de services clients BSCS : Business Support and Control System, géré par Talan, et caractérisé par son architecture modulaire et hiérarchique qui facilite son intégration avec les systèmes des différents opérateurs.

Une des composantes principales de ce système est les plans tarifaires. Ils définissent les différents services offerts, les charges qu'y sont associées et les incitations et forfaits affiliés et qui sont essentiels à l'opération de facturation.

L'extraction des plans tarifaires et de leurs différents éléments permet de donner une représentation significative de l'état du système d'informations. Cette extraction est une démarche compliquée, qui se fait traditionnellement avec des scripts rigides moyennant des requêtes complexes, et qui nécessite une bonne connaissance de leurs structures qui s'étend sur plusieurs niveaux.

L'analyse de ces plans tarifaires suivant différentes dimensions permet d'exploiter et mettre en valeur les données existantes dans les historiques du système, pour faciliter la procédure d'analyse et par conséquent simplifier la prise de décision.

## 2.2. Solution proposée

Dans le cadre de la personnalisation des fonctionnalités autour du système de facturation BSCS et pour remédier aux différents problèmes cités dans ce qui a précédé, l'équipe de Billing opte à simplifier l'extraction des plans tarifaires et l'analyse de leurs évolution en développant une solution de génération de catalogue des offres.

Le présent sujet consiste à développer une application web qui traitera les différentes fonctionnalités liés à ces opérations.

## 3. Méthodologie et formalisme adoptés

### 3.1. Méthodologie

Une méthodologie de développement logiciel définit une séquence d'étapes, en partie ordonnées qui concourent à l'obtention d'un système logiciel.

Nous avons opté pour le processus de développement 2TUP qui propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches comme illustré sur la figure 3.

2TUP s'estime convenable pour couvrir les besoins de notre projet puisqu'il gère la complexité technologique en donnant part à la technologie dans son processus de développement, piloté par les cas d'utilisations et parce qu'il cible les projets de toute taille [2].

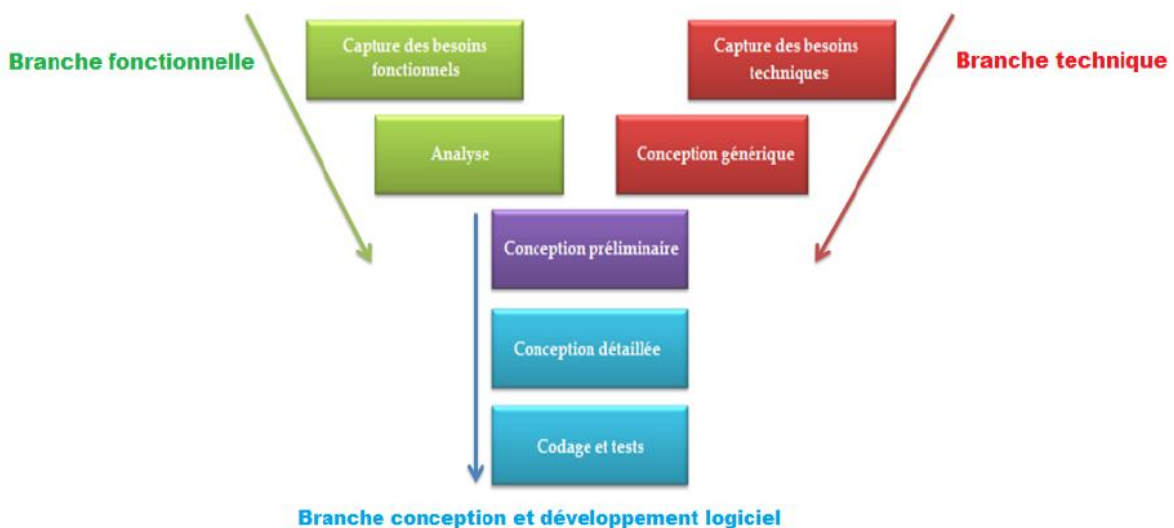


Figure 3 Processus de développement 2TUP [3]

### **3.2. Formalisme Utilisé**

Nous avons utilisé, pour la spécification et la conception de ce travail, le langage de modélisation UML (Unified Modeling Language). UML permet de décrire les besoins et documenter les systèmes ainsi que d'esquisser les architectures logicielles. Il s'articule, depuis sa dernière version, autour de quatorze diagrammes dédiés à la présentation d'un concept particulier du système étudié. Toutefois, pour éviter de surcharger le rapport et d'entrer dans certains détails techniques, nous ne présenterons que quelques diagrammes que nous avons jugés utiles pour comprendre le projet à savoir les diagrammes des cas d'utilisation, les diagrammes de modèles, les diagrammes de classes et les diagrammes de séquences :

- Les diagrammes de cas d'utilisation permettent l'énumération des fonctions de l'application de point de vue de l'utilisateur.
- Les diagrammes de classes et de modèles permettent une représentation statique du système étudié, en représentant les classes, les paquetages et les relations entre eux.
- Les diagrammes de séquences permettent une représentation temporelle et comportementale des objets et leurs interactions.

### **Conclusion**

Dans ce chapitre, nous avons défini le cadre du projet, à travers la présentation de l'organisme d'accueil, la problématique suscitant la réalisation de ce projet ainsi que la solution proposée. Nous avons clôturé en donnant la méthodologie et le formalisme adoptés lors de la réalisation de notre travail. Nous entamons par la suite l'étude préalable de notre système décrivant les différentes notions théorique impliquées.

# Chapitre 2 : Etude préalable

---

## Introduction

La réussite de toute étude dépend de la qualité des étapes initiales. De ce fait, l'étape de l'étude préalable constitue une base de départ pour notre projet. Ce chapitre est consacré à la présentation de différentes notions théoriques fondamentales à la compréhension de notre système ce qui va nous permettre de nous accommoder avec les différents standards et technologies utilisés au cours de son développement.

## 1. Système de facturation téléphonique BSCS

### 1.1. Présentation

Business Support and Control System BSCS est un système de gestion clientèle et de facturation téléphonique pour tous types d'opérateurs : téléphonie fixe, mobile, internet et télévision.

Construit à base de composants modulaires, BSCS est un moyen rentable pour gérer les opérations de facturation d'une entreprise. Il peut manager tout type de client ou partenaire et intègre les capacités pour manipuler des modèles d'affaires multi-filiales.

BSCS offre une grande souplesse dans la configuration des promotions, des réductions, des incitations et de nouveaux forfaits innovants pour toutes les technologies de réseau, ainsi que lors de la création de factures ou règlements pour tous les clients et les partenaires.

Ce système est populaire (150 installations dans 80 pays), grâce à sa haute configurabilité. Son architecture convergente ouverte, à base de composants permet d'augmenter l'efficacité opérationnelle et d'avoir également l'agilité nécessaire pour s'adapter rapidement aux changements de l'entreprise. Ce système intègre plusieurs fonctionnalités :

- Gestion de clientèle : support pour les services post payés en contrat avec une administration de la clientèle et des partenaires en utilisant une base de données commune et un riche ensemble de fonctionnalités de facturation.
- Facturation : il offre une facturation en temps réel, des taxes configurables, gestion des promotions, des algorithmes d'arrondi configurables et des interfaces ouvertes.

- Valorisation : intègre un moteur de tarification qui peut évaluer tout type d'utilisation, y compris, la voix, des données, du contenu et de toutes autres opérations événementielles.

## 1.2. Les Composantes de BSCS

Dans cette partie nous allons introduire plusieurs concepts fondamentaux liés à BSCS, traités par ce sujet, et qui sont essentielles à l'opération de facturation.

### 1.2.1 Contrats

Un contrat est un document juridiquement contraignant qui maintient des relations d'affaires entre un opérateur et ses clients. Les contrats sont utilisés pour mettre en place des règles de gestion qui régissent les produits qui sont offerts par l'opérateur, définissent les charges, les crédits, les impôts et les conditions de paiement pour les services souscrit.

### 1.2.2 Plan tarifaire

Un plan tarifaire est un ensemble de paquets (figure 4), définissant les différentes informations relatives à ses services, les charges qu'y sont associés, les prix unitaires et les unités gratuites, servant à savoir affecter une charge à chaque service offert aux clients. Ces paquets sont principalement :

- Les paquets de services : qui permettent de grouper des services spécifiques aux clients
- Les charges : représentent le prix que doit payer le client pour l'utilisation des services qu'ils lui sont offerts par l'opérateur
- Les unités gratuites : accordent l'utilisation gratuite des services pendant un intervalle de temps déterminé
- Les taxes : définissent les taxes associées aux charges

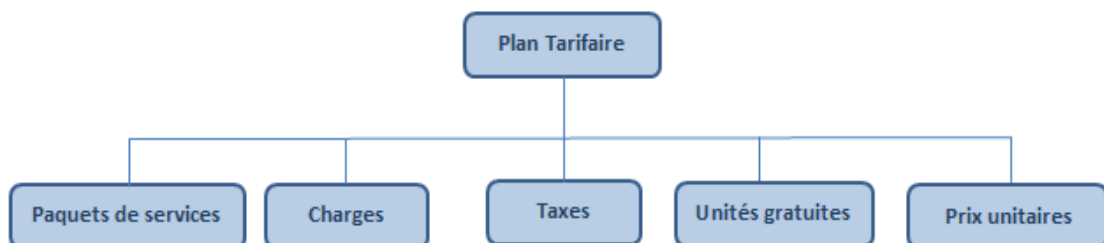


Figure 4 Structure du plan tarifaire

### 1.2.3 Paquets de services

Les paquets de services (service package) sont des unités organisationnelles qui regroupent un ou plusieurs services, pour permettre à l'opérateur de rassembler des services spécifiques aux clients, et qui sont nécessaires pour attribuer un service à un plan tarifaire. Un plan tarifaire peut comporter :

- Un ou plusieurs paquets de services qui peuvent avoir des services communs pour permettre différentes charges à un même service.
- Un seul paquet de service par défaut (default service package) qui sert à définir les services implicites qui sont automatiquement fournis aux clients lors de l'attribution du plan tarifaire.

### 1.2.4 Services

Les services sont les entités qui permettent aux clients de bénéficier des fonctionnalités offertes par l'opérateur et qui sont classés en deux types : les services réseaux et les services non réseaux.

**Services non réseaux :** Ce type de service ne nécessite pas une activation sur le réseau, ils sont utilisés pour définir des événements internes au système et permettent la facturation des services administratifs et orientés marketing.

**Services réseaux :** Ce type de service permet à un client de communiquer directement avec un autre client à travers le réseau. Il existe différents sous types de services réseaux :

- Les services d'utilisation (Usage service) : service de télécommunication standard (téléphonie, fax, SMS) qui inclut le transfert de voix ou de données.
- Les services d'événement (Event service) : des caractéristiques réseaux supplémentaires qui peuvent être activées par l'opérateur comme le renvoi ou l'interdiction d'appels.
- Services de contenu (Content service) : permet à l'utilisateur d'acheter un contenu
- Services de contrôle de coût (Cost control service) : utilisés pour surveiller l'utilisation des services contracturés

#### **Service de base**

Tous les services réseau se composent principalement de services de base. Il existe deux sous types de services de base :

- Teleservice : définit la partie fonctionnelle de la transmission des données
- Bearer service : définit les précisions techniques nécessaires pour établir la connexion au réseau
- Service supplémentaire : un service qui modifie ou ajoute à un service de base (renvoi d'appel, interdiction d'appel...)

### ***Paramètres de services***

Les paramètres de services sont utilisés pour ajouter des spécifications à un service ou dans d'autres cas ils sont nécessaires pour sa configuration : par exemple les services GPRS nécessitent un point d'accès APN.

### **1.2.5 Charges**

Les charges présentent le prix que doit payer un client pour l'utilisation des services qu'ils lui sont offerts par l'opérateur, et qui sont définis par service au sein d'un plan tarifaire. Nous distinguons différents types de charges :

- One Time charge : Une charge payée une seule fois lors de l'activation d'un service pour un contrat pour la première fois.
- Charges récurrentes : Charge appliquée périodiquement pour avoir un service disponible.
- Charges d'utilisation : Utilisées pour définir les charges d'utilisation des services de type réseau. Ce type de charge est basé sur le nombre d'unités consommées (durée d'un appel) ou d'évènement (envoi d'un SMS).
- Charges d'évènement : Les charges d'évènement sont utilisées pour appliquer des charges par évènement.

### **1.2.6 Rating Package**

Les ratings packages sont assignés individuellement à chaque service, et permettent de calculer les charges d'utilisation pour les services de type réseau en appliquant une formule sur les données en entrée résultant de l'utilisation d'un service (exemple la durée d'un appel) pour générer le montant associé. Les charges d'utilisation calculées dépendent de plusieurs paramètres dont la combinaison de zone et du temps, les conditions sous lesquels un appel a été transmis, la quantité utilisée ou encore les indicateurs d'utilisation.

### Combinaison zone et temps

La combinaison de la zone et du temps est défini à travers un paquet d'utilisation (usage package). Ce dernier permet de définir une combinaison spécifique à travers l'association d'un seul paquet de zone (zone package) avec un ou plusieurs paquets de temps (time package) comme indiqué sur la figure 5.

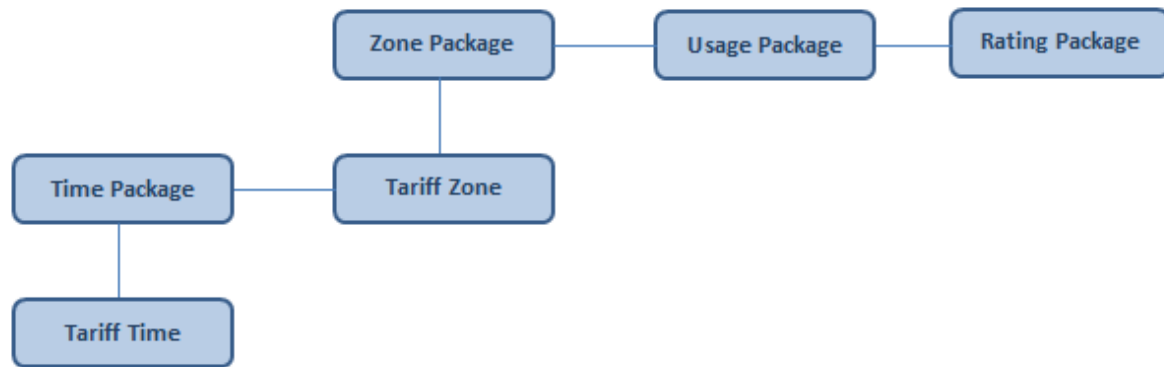


Figure 5 Structure du rating package

Les paquets de zones sont composés d'un ensemble de tarif zone, constitué d'une zone origine et un ensemble de zones destination et les paquets de temps qui eux-mêmes sont composés de plusieurs tarif time, qui définissent les types de jours (vacance, journée spéciale) et les intervalles de temps associés.

**Chargeable quantity :** C'est la quantité pour laquelle un service est chargé. Elle pourrait s'introduire en différentes unités de mesure (secondes, bytes...) selon le service associé. Un rating package peut être configuré avec différentes unités de mesure pour permettre son affectation à différents types de services.

**Indicateurs d'utilisation :** Ceux sont des attributs qui spécifient le type d'utilisation du service (appel sortant, rentrant...) et qui permettent de lier le rating package au service.

#### 1.2.7 Promotions

Les promotions sont principalement des critères d'évaluation associées à des actions appliquées lorsque ces critères sont remplis. Ils sont utilisés pour accorder des bonifications spéciales aux clients et qui permettent aux utilisateurs de bénéficier de promotions exceptionnelles et aux fournisseurs de services de les utiliser comme un outil de commercialisation.

#### Mécanismes de promotions

Le terme «mécanismes de promotions » regroupe les deux notions de mécanismes d'évaluation et d'application :



- Mécanismes d'application : spécifient les actions promotionnelles à appliquer. Ces actions varient entre des cadeaux immédiats, remises, crédit prépayé, points bonus et engagement minimum.
- Mécanismes d'évaluation : utilisés pour définir un ensemble de critères évalués pour qu'un mécanisme d'application soit exécuté. Les évaluations possibles sont : le revenu généré, le nombre de contrats actifs, le nombre de points bonus et les données personnelles des clients.

### **1.2.7 Unités gratuites**

Les Unités gratuites (Free Units) offrent une utilisation gratuite des services de type réseau pour un intervalle de temps spécifique et ainsi permettent de réduire les charges d'utilisation. Contrairement aux promotions les unités gratuites sont indépendantes du type et du volume des appels et elles sont accordées en différentes unités de mesure : minutes, bytes, message... Les unités gratuites non consommées durant la période spécifiée peuvent être reportées à une période postérieure.

#### **Types**

Les unités gratuites sont classées en plusieurs types :

- COFU : offertes à un seul contrat d'un utilisateur.
- POFU : partage entre une sélection de contrats d'un utilisateur ou client grand compte.
- POFUL : partage entre tous les contrats d'un utilisateur ou d'un client grand compte.

#### **Affectation aux contrats**

Un paquet d'unités gratuites peut se voir affecter à un contrat suivant deux méthodes

- Affectation implicite : Associer le paquet à un plan tarifaire et par conséquent il sera disponible à tous les contrats ayant ce plan tarifaire
- Affectation explicite : Affectation à travers les paramètres des services, ce qui permet une attribution du paquet à un contrat spécifique

## **2. Analyse des données**

### **2.1. Présentation**

#### **2.1.1 Définition**

L'analyse des données est un processus d'inspection, de nettoyage, de transformation, et de modélisation des données dans le but de découvrir des informations utiles. Elle englobe les solutions apportant une aide à la décision avec, en bout de chaîne, rapports et tableaux de bord analytiques afin consolider les informations disponibles au sein des bases de

données de l'entreprise. Elle a pour finalité principale de délivrer des informations pertinentes à chaque manager afin qu'il puisse prendre le plus efficacement possible les meilleures décisions selon son contexte d'action, ses prérogatives et ses objectifs tactiques et stratégiques.

### **2.2.2 Rôle dans l'entreprise**

L'analyse des données offre aux gestionnaires les informations nécessaires et utiles pour les aider dans le processus de prise de décision et de pilotage de l'entreprise. Elle permet d'avoir une vision claire sur la performance de l'entreprise et peut être utilisée pour des fins de prévisions ou de performances. L'analyse de données cache la complexité des traitements faits à l'intérieur de son processus pour fournir des informations de manière simple, fiable et transparente aux décideurs.

## **2.2. Analyse multidimensionnelle**

### **2.1.1 Définition**

L'analyse multidimensionnelle est un processus d'analyse de données qui permet d'extraire des informations stratégiques et opérationnelles sous des angles et axes multiples. Elle aide ainsi les décideurs à sélectionner des données selon divers dimensions dans le but de valider des décisions ou de vérifier des tendances. Cette procédure d'analyse de données regroupe les données dans deux catégories : les dimensions de données et les mesures.

### **Mesures**

Les mesures sont les graines fondamentales de l'activité de l'entreprise, et ils sont la cible ultime des différentes opérations analytiques. Ces entités sont les propriétés sur lesquelles peut être appliquées les différentes opérations de calcul : somme, moyenne, maximum...

### **Dimension**

Une dimension est une collection d'informations de référence catégorisant un mesurable, et qui fournit une multitude de points d'entrée pour filtrer et grouper les différentes données.

### **2.2.2 Navigation dans les données**

Dans le monde du multidimensionnel, les données sont chargées dans des structures simulées à des cubes possédant plusieurs axes. Différentes opérations standard existent pour trancher cet ensemble d'informations en parties plus fines pour l'examiner à partir de différents points

de vue afin de pouvoir mieux comprendre leur structure. Parmi ces opérations nous trouvons les méthodes de slice et dice.

### L'opération slice

L'opération de slice (figure 6) permet de sélectionner des données en appliquant un critère de sélection à un seul axe. Elle permet à l'utilisateur de visualiser et de recueillir des informations spécifiques à une seule dimension [4].

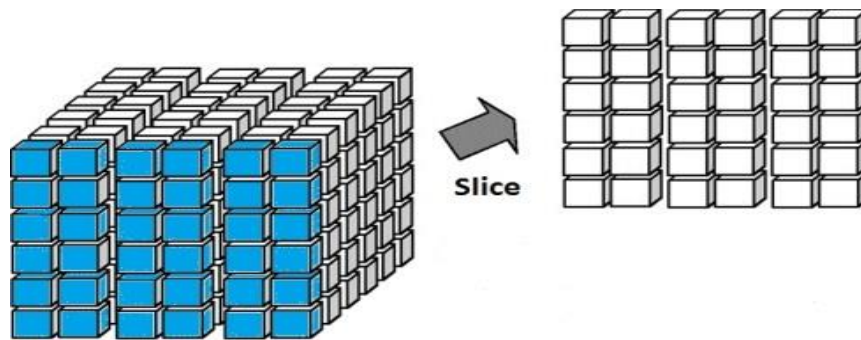


Figure 6 Opération slice [5]

### L'opération dice

L'opération de dice (figure 7) est similaire à celle de slice, mais elle fonctionne un peu différemment. Le dice est plus une fonction de zoom qui sélectionne un sous-ensemble sur toutes les dimensions, mais pour des valeurs spécifiques de la dimension. Cet outil est très utile pour permettre à l'utilisateur d'obtenir des informations plus détaillées sur ce qui se passe dans une plus petite échelle [4].

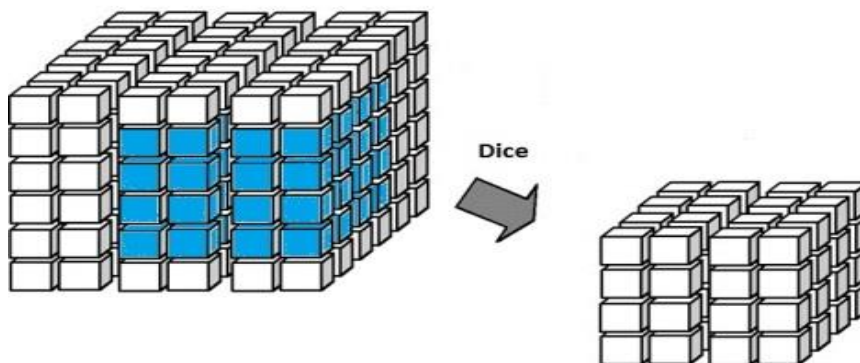


Figure 7 Opération dice [5]

### **3. Reporting**

#### **3.1. Définition**

L'opération de reporting est le processus d'organisation des données dans des résumés d'information afin de surveiller la performance des différents domaines d'une entreprise. Elle l'aide à examiner ses activités et l'alerter quand ces données tombent en dehors de plages attendues, en vue d'extraire des idées utiles et prendre les mesures appropriées.

#### **3.2. Caractéristiques**

L'opération de reporting doit suivre certaines contraintes afin de garantir la clarté des données incluses et la cohérence des informations qui peuvent être dégagées. Dans cette partie nous introduisons les différentes caractéristiques d'un rapport analytique :

- Compact et complet : le rapport est un document complet et compact qui fournit des informations mises à jour [8].
- Présentation systématique des faits : Le rapport est une présentation systématique des faits, des chiffres et des conclusions [8].
- Un document auto-explicatif : un document exhaustif et couvre tous les aspects de l'objet de l'étude. Il est auto-explicatif et complet en soi [8].
- Source d'information : Le rapport est un document précieux qui donne des informations et des conseils pour la gestion. Il facilite la planification et la prise de décision [8].
- Agit comme un record permanent : Un rapport sert comme enregistrement permanent relatif à un certain secteur fonctionnel. Il est utile pour référence et futures orientations [8].

### **Conclusion**

Nous avons étudié dans ce chapitre les différentes notions relatives à notre étude préalable. Nous avons tout d'abord introduit le système de facturation BSCS ainsi que ses différentes composantes. Nous avons également introduit le concept d'analyse des données, d'analyse multidimensionnelle et de reporting. Ceci fait, nous pouvons maintenant entamer le chapitre d'analyse et spécification des besoins qui nous permettra de cerner les périmètres fonctionnels et non fonctionnels de notre système.

# Chapitre 3 : Analyse et spécification des besoins

---

## Introduction

La spécification des besoins constitue la phase de départ de toute application à développer dans laquelle nous allons en premier lieu identifier les acteurs de notre système, ensuite nous allons spécifier les besoins fonctionnels et non fonctionnels de notre application, et en fin identifier les différents cas d'utilisations accompagnés des diagrammes illustratifs nécessaires.

### 1. Identification des acteurs

Les acteurs d'un système sont les personnes, les processus ou autres entités externes interagissant avec ce système.

Notre application va devoir communiquer avec les différents consultants du système de facturation téléphonique BSCS.

### 2. Identification des besoins

L'identification des besoins consiste à faire l'inventaire du système et de distinguer les différentes exigences liées à l'application.

Ces besoins sont classés en deux types, fonctionnels et non fonctionnels.

#### 2.1. Besoins fonctionnels

Les besoins fonctionnels définissent les caractéristiques du système et mettent en œuvre les processus que doit exécuter l'application.

La solution proposée doit répondre à un certain nombre d'exigences fonctionnelles :

- L'application doit permettre à l'utilisateur de se connecter à l'environnement d'extraction en spécifiant les paramètres d'accès à la base de données choisie.
- L'application doit permettre à l'utilisateur d'extraire les données BSCS correspondantes à la base et notamment de :
  - Choisir les données BSCS à extraire.
  - Appliquer des filtres sur ces données en indiquant un ou plusieurs paramètres qui permettent de simplifier le résultat de recherche.
  - Sauvegarder les résultats d'extraction sous le format standard Excel.

- L'application doit permettre à l'utilisateur d'analyser les données :
  - Choisir les axes d'analyse
  - Générer un rapport analytique sous format PDF.

## 2.2. Besoins non fonctionnels

Les besoins non fonctionnels définissent les caractéristiques et les contraintes techniques que doit satisfaire une application informatique.

Notre application doit contenter différentes exigences non fonctionnelles :

- La maintenabilité : La modification des fonctionnalités au sein de l'application doit se faire avec le minimum d'effort.
- L'extensibilité : la solution doit être évolutive afin de répondre aux éventuels changements.
- L'ergonomie : l'application doit présenter des interfaces simples, conviviales et facilement utilisables.

## 3. Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation représente toutes les actions susceptibles d'être réalisées par l'utilisateur. Un cas d'utilisation (CU) décrit ainsi une fonctionnalité de l'application.

### 3.1 Diagramme des cas d'utilisation général

La figure 8 illustre le diagramme général des cas d'utilisations avec les fonctionnalités principales de notre système. Un utilisateur connecté à l'environnement d'extraction a la possibilité de :

- Extraire les données
- Analyser les données

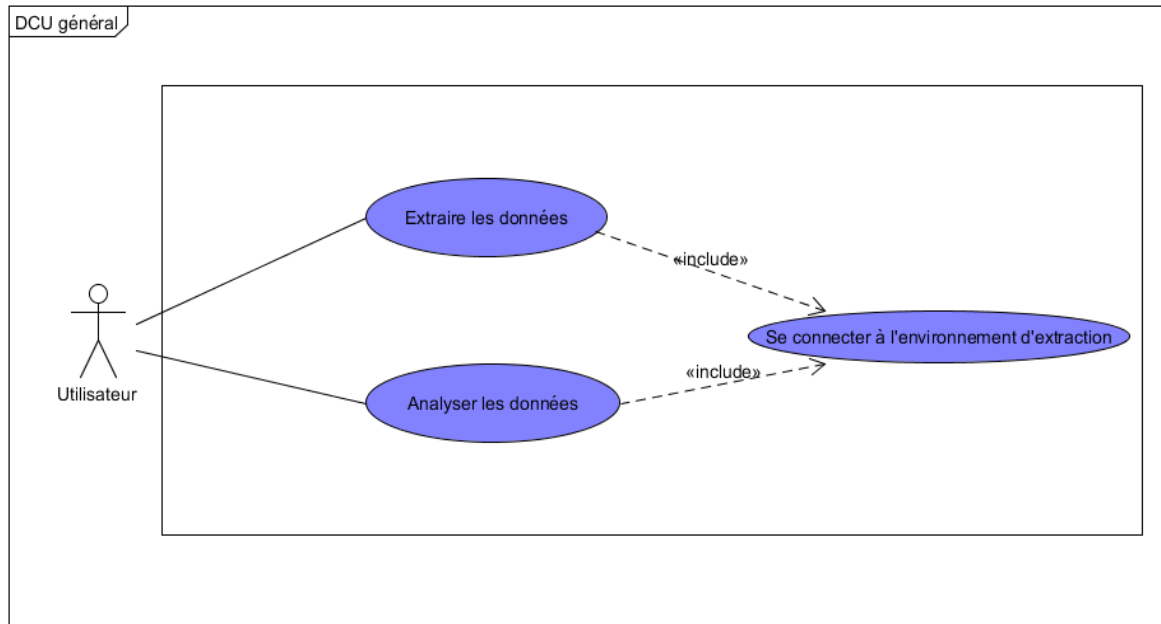


Figure 8 Diagramme des cas d'utilisation général

### 3.2. Cas d'utilisation se connecter à environnement

#### 3.2.1 Diagramme de cas d'utilisation choisir environnement

Avant de bénéficier des différentes fonctionnalités offertes par l'application, l'utilisateur doit, suivant la figure 9, se connecter à l'environnement d'extraction en spécifiant la machine hôte et la base de données.

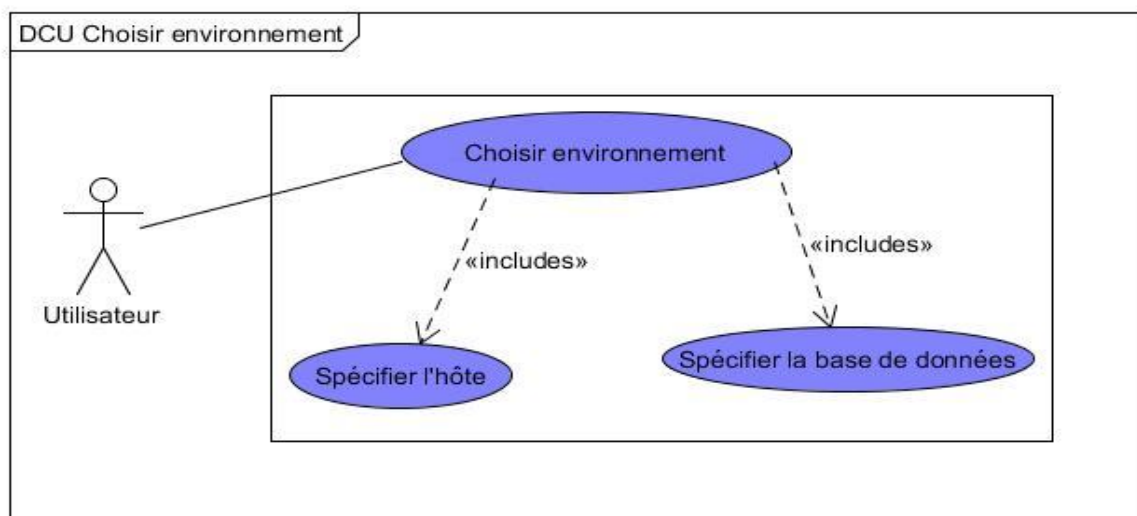


Figure 9 Cas d'utilisation se connecter à l'environnement

### 3.2.2 Description de cas d'utilisation choisir environnement

Nous détaillons le cas d'utilisation «choisir environnement » dans le tableau suivant :

Titre	Se connecter à l'environnement
Acteur	L'utilisateur de l'application
Pré conditions	L'utilisateur lance l'application
Scénario nominal	<ol style="list-style-type: none"><li>1. L'utilisateur saisit l'adresse IP et le numéro de port associé à l'hôte.</li><li>2. L'utilisateur saisit le nom de la base de données.</li><li>3. L'utilisateur saisit le nom d'utilisateur et le mot passe</li><li>4. L'utilisateur demande de se connecter</li></ol>
Exception	Les paramètres saisis sont incorrects
Post conditions	L'utilisateur accède à la page d'accueil de l'application

## 3.3. Cas d'utilisation extraire données

### 3.2.1 Diagramme de cas d'utilisation extraire données

Un utilisateur connecté à l'environnement d'extraction peut comme illustré sur la figure 10,

- Choisir le niveau de détails : l'utilisateur sélectionne les notions à inclure dans l'opération d'extraction.
- Appliquer des filtres : l'utilisateur applique des filtres sur les notions à extraire.
- Sauvegarder les résultats : l'utilisateur sauvegarde les résultats de l'extraction sous format Excel.



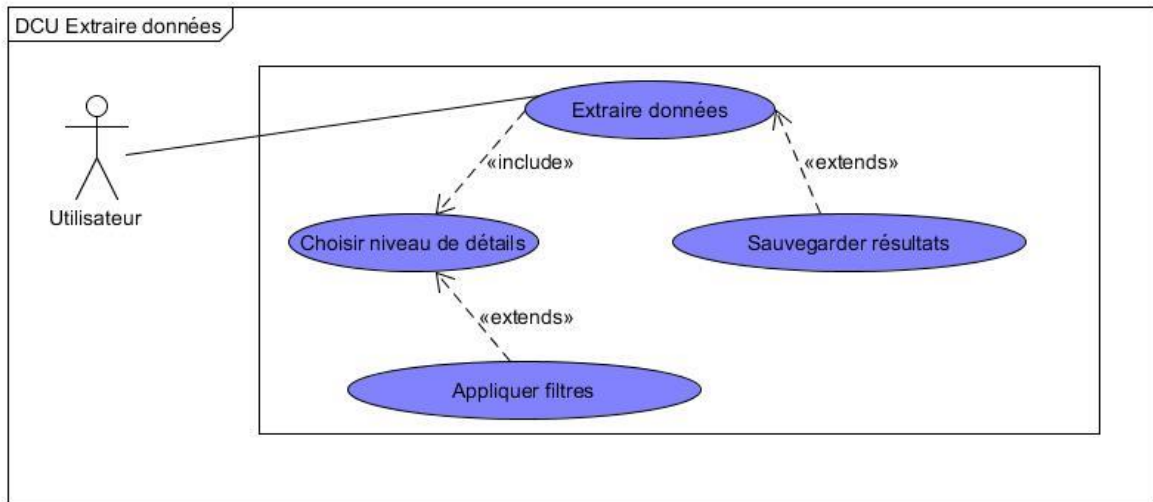


Figure 10 Cas d'utilisation extraction des données

### 3.3.2 Description de cas d'utilisation extraire données

Nous détaillons le cas d'utilisation « extraire données » dans le tableau suivant :

Titre	Extraire données
Acteur	L'utilisateur de l'application
Pré conditions	L'utilisateur accède au menu extraction données
Scénario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur choisit les informations à extraire.</li> <li>2. L'utilisateur choisit les filtres à appliquer pour chaque notion.</li> <li>3. L'utilisateur demande l'extraction des données</li> <li>4. L'utilisateur consulte le catalogue extrait</li> <li>5. L'utilisateur demande la sauvegarde des résultats d'extraction</li> </ol>
Post Conditions	Le catalogue est généré

### 3.4. Cas d'utilisation analyser données

#### 3.4.1 Diagramme de cas d'utilisation analyser données

Une fois l'utilisateur est connecté, il peut aussi analyser les données. Il devrait comme indiqué sur la figure 11 :

- Choisir axes d'analyse : l'utilisateur choisit la donnée et les dimensions selon lesquelles elle va être analysée, tout en ayant la possibilité de choisir une ou plusieurs valeurs distinctes pour chaque dimension.
- Générer rapport : l'utilisateur génère un rapport analytique sous format PDF.

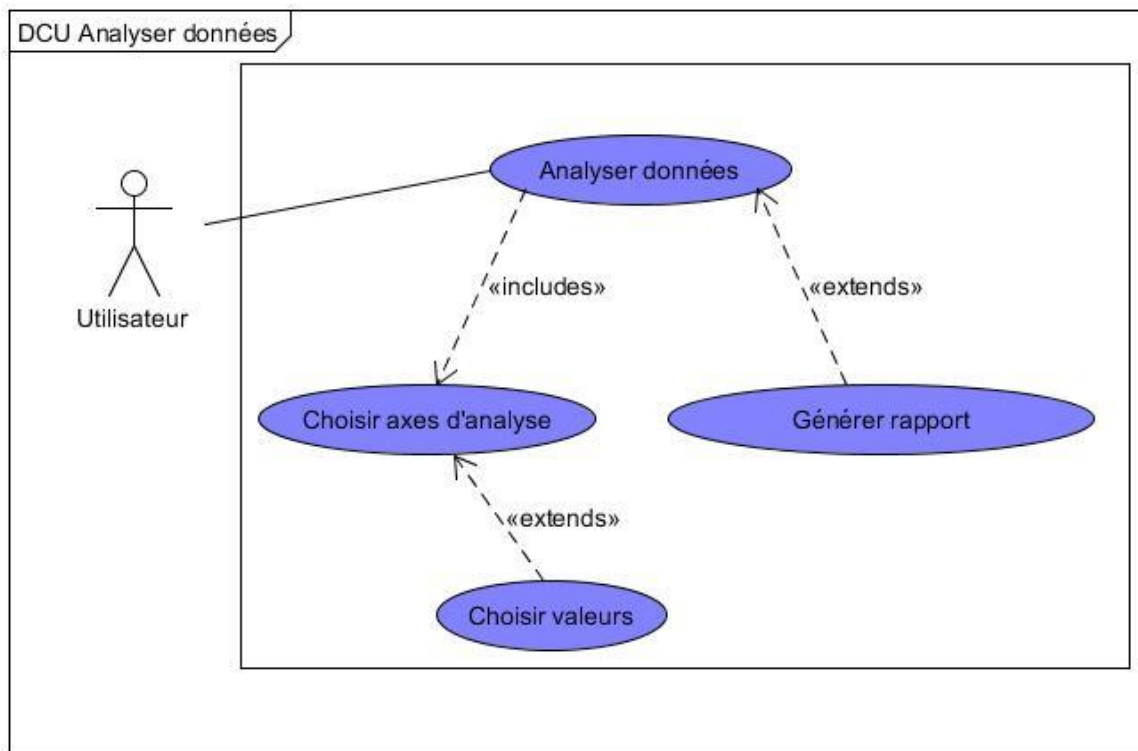


Figure 11 Cas d'utilisation analyser les données

#### 3.4.2 Description de cas d'utilisation analyser données

Le tableau suivant détaille le cas d'utilisation « analyser données »

Titre	Analyser données
Acteur	L'utilisateur de l'application
Pré conditions	L'utilisateur accède au menu extraction données
Scénario nominal	<ol style="list-style-type: none"><li>1. L'utilisateur choisit les données et les axes d'analyses.</li><li>2. L'utilisateur spécifie des valeurs spécifique pour chaque axe</li><li>3. L'utilisateur demande l'analyse des données</li><li>4. L'utilisateur consulte les résultats de l'analyse</li><li>5. L'utilisateur demande la génération d'un rapport</li></ol>
Post conditions	Un rapport d'analyse est généré

## **Conclusion**

Dans ce chapitre nous nous sommes intéressés à l'analyse des besoins fonctionnels et non fonctionnels de notre application. Nous avons formalisé par la suite cette analyse dans des diagrammes des cas d'utilisation et nous les avons détaillés dans des tableaux décrivant les scénarios d'interactions entre l'utilisateur et le système. Nous pouvons ainsi entamer la prochaine étape qui consiste à présenter l'étude conceptuelle de la solution.

# Chapitre 4 : Etude Conceptuelle

---

## Introduction

L'étude conceptuelle est une étape essentielle dans le cycle de développement de tout logiciel. En effet, il s'agit durant cette étape de fournir les éléments de réponse pour le comment faire technique d'une solution répondant aux besoins dégagés précédemment, aboutissant ainsi à une solution prête à être implémentée.

Nous nous intéresserons donc dans ce chapitre à la conception générale de l'application que nous détaillerons par la suite.

## 1. Conception globale

Dans cette partie nous nous intéresserons à l'architecture de notre application. Nous aborderons d'abord l'aspect physique à travers l'architecture technique, puis l'aspect logique à travers l'architecture logicielle.

### 1.1 Architecture physique

L'architecture physique est une structure de constituants et de liens physiques qui les connectent supportant l'application.

En vue de garantir un bon niveau d'évolutivité et d'extensibilité de l'application, on aurait besoin d'une architecture offrant une structure modulaire. Une solution serait d'adopter une architecture 3 tiers qui aboutirait à un système séparé en plusieurs niveaux. Le principe d'une telle architecture est relativement simple : il consiste à séparer la réalisation en trois parties : le stockage des données, la logique applicative et la couche présentation. Il est possible de déployer chaque partie sur un serveur indépendant, toutefois cela n'est pas obligatoire. La mise en place de ce type d'architecture permet dans tous les cas une plus grande évolutivité du système. Il est ainsi possible de commencer par déployer les deux serveurs sur la même machine, puis de déplacer le serveur applicatif sur une autre machine lorsque la charge devient excessive. Comme nous pouvons le constater sur la figure 12 les éléments permettant la réalisation classique d'un système en architecture trois tiers sont les suivants :

- Serveur base de données pour la gestion les données

- Serveur d'application pour la logique applicative qui correspond à la partie fonctionnelle de l'application, celle qui implémente et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation.
- Navigateur web pour la présentation qui correspond à la partie de l'application visible et interactive avec les utilisateurs.

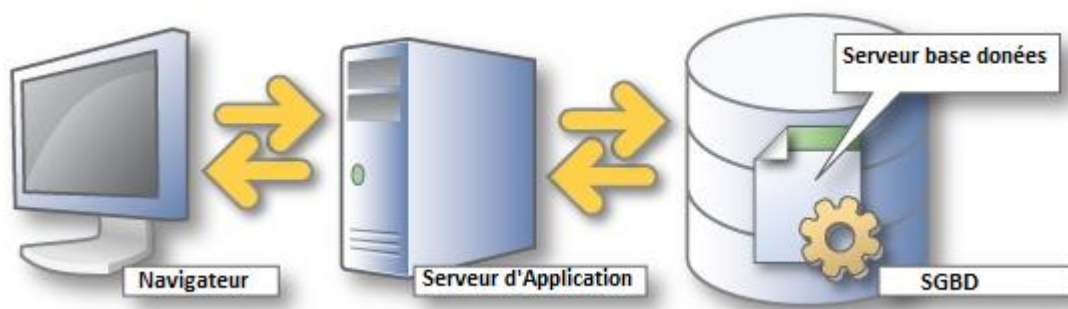


Figure 12 Architecture 3 Tiers

## 1.2. Architecture Logicielle

Par opposition à l'architecture physique de l'application, où il s'agissait de distinguer entre les différents niveaux physiques de l'application, l'architecture logicielle s'intéresse plutôt au découpage logique de l'application et la façon de regrouper les composants selon le type de fonctions et traitements qu'ils effectuent. Pour notre application nous avons opté pour une architecture en trois couches comme illustré sur la figure 13.

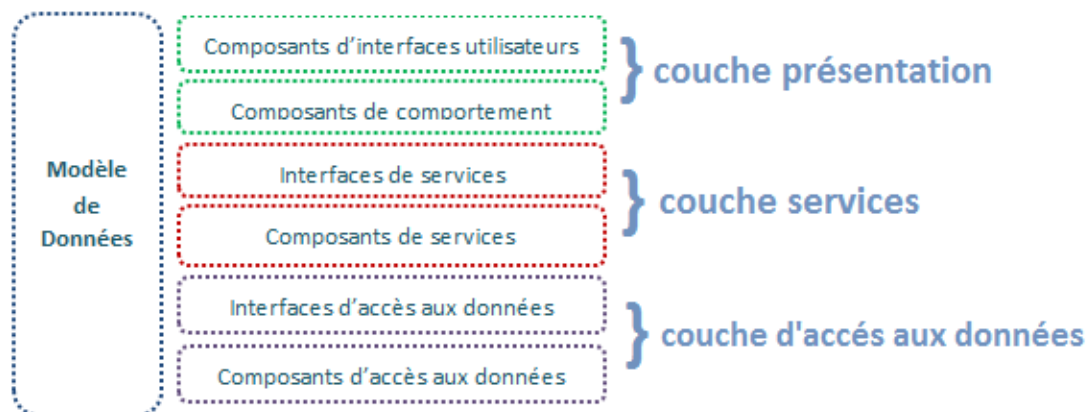


Figure 13 Architecture logicielle

Ces principales couches sont :

- La couche présentation, qui assure l’affichage des interfaces graphiques et gère les interactions avec les utilisateurs.
- La couche de services qui implémente la logique métier de l’application, et qui offre des interfaces pour gérer les différentes fonctionnalités.
- La couche d’accès aux données, qui fournit aux services une interface pour accéder aux données.
- La couche modèle de données, qui contrairement aux autres couches, est transverse à toute l’architecture, et chaque couche peut donc manipuler les entités qui la composent.

Pour mettre en œuvre cette architecture nous utilisons les mécanismes d’interfaces et implémentation : chaque couche service est ainsi composée d’interfaces qui publient un certain nombre de fonctionnalités et d’autre part les classes qui les implémentent.

Les couches supérieures travaillent ainsi uniquement sur les interfaces qui représentent des « contrats » entre les deux couches et ignorent donc la complexité de l’implémentation. Celles-ci peuvent donc changer sans impacter les couches qui utilisent le service.

## **2. Conception détaillée**

Nous détaillerons dans cette partie les différents modules composant notre système à travers tout d’abord un diagramme de modèles qui nous permettra de mieux comprendre sa structure générale. Nous détaillerons par la suite la composition de ces paquetages à travers des diagrammes de classes.

### **2.1. Diagramme de modèles**

Le diagramme de modèles, introduit dans la version 2.4 d’UML, est un diagramme de structure auxiliaire issu du traditionnel diagramme du package et qui permet donner une vue spécifique du système pour décrire ses aspects architecturaux, logiques ou comportementaux [6], permettant ainsi de surpasser les contraintes rigides du diagramme du package qui suppose un maximum de cohésion au sein d’un même package et un minimum de couplage entre deux packages différents. Vu le nombre important de classes de notre application, il s’avère utile de les regrouper en paquetages afin de mieux comprendre le rôle de chaque partie favorisant ainsi la maintenabilité du code. Nous avons regroupé ces classes selon la

couche logique à laquelle elles appartiennent. Nous obtenons alors le diagramme de modèles de la figure 14 composé des modules suivants :

- Contrôlers : regroupe les classes fournissant les données aux interfaces utilisateurs et spécifiant leur comportement.
- Services : regroupe les classes liées à l'extraction des données, d'analyse et de reporting.
- Utilities : regroupe les classes utilitaires.
- DTO : regroupe les classes liées au transfert de données entre le module d'accès aux données et les services.
- DAO : relatif à la couche d'accès aux données et qui regroupe les différentes classes permettant l'extraction des données persistées.
- Entities : représentent les objets persistés dans la base de données.

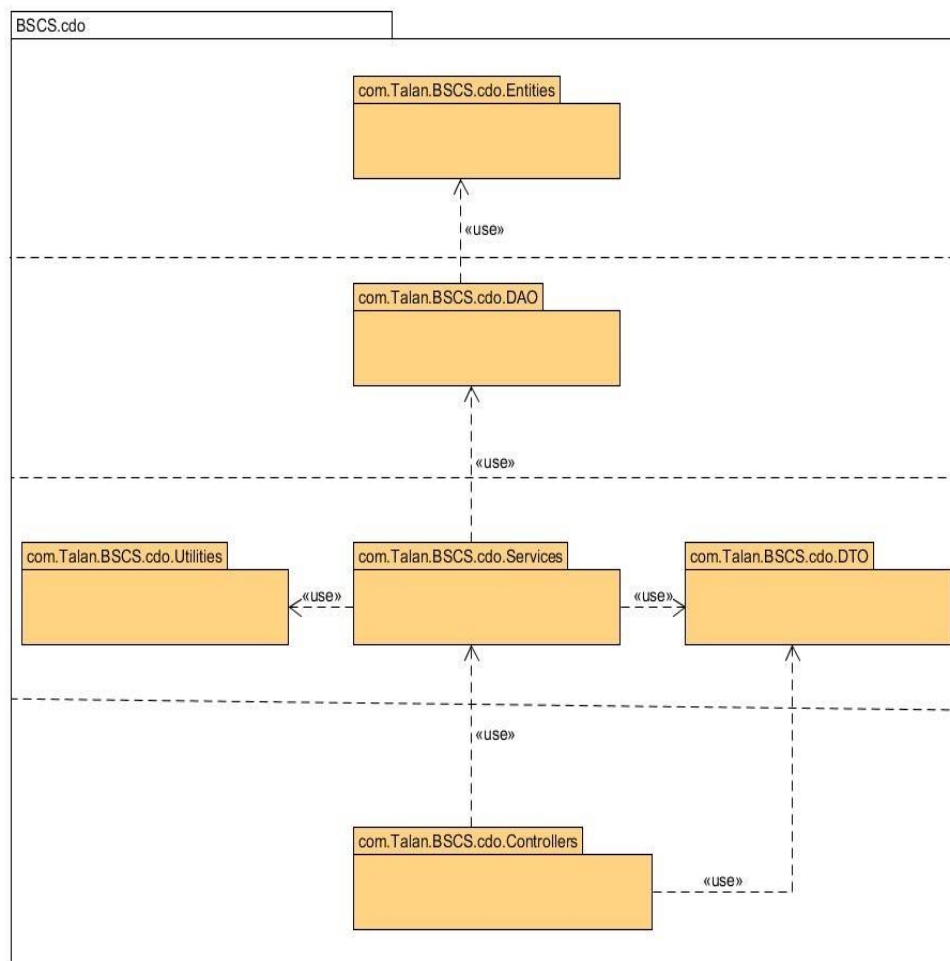


Figure 14 Diagramme de modèles

## **2.2. Diagrammes de classes**

Dans cette section, nous allons raffiner les packages pour chaque volet à un niveau d'abstraction plus bas. Chaque composant précédemment cité est maintenant détaillé par un diagramme de classes qui explicite les entités qui le forment et les relations entre elles.

### **2.2.1. Package Entities**

Le package « Entities » regroupe les objets qui sont persistés en des données par un moteur de persistance : chaque objet est la représentation d'un élément relationnel. Le système de facturation BSCS est un produit propriétaire qui intègre sa propre base de données. Nous avons eu recours à la documentation élaborée par son fournisseur et après une étude approfondie de sa base de données, nous avons dégagé les tables dont nous aurons besoin. Ceci nous a aidés à construire le diagramme de classes, représenté par la figure 15, pour présenter les classes de données ainsi que les différentes relations entre elles. Pour des raisons d'encombrement ce diagramme ne représentera qu'un aperçu des classes présentes dans ce package, il sera détaillé dans sa totalité dans l'annexe.

Ce diagramme comporte plusieurs classes dont :

- Les classes définissant les informations liées à chaque notion :
  - ContractAll : définit les données relatives aux contrats
  - Rateplan : regroupe les informations liées aux plans tarifaires
  - FreeUnits : regroupe les informations liées aux unités gratuites
  - ServicePackage & Service : introduisent les informations des paquets de services et services
  - Rating Package : définit les informations relatives au rating packages
  - ZonePackage & TimePackage : regroupe les informations liées respectivement aux paquets de zone et de temps
  - TariffZone & TariffTime : introduisent les informations des tarifs zone et temps
- Les classes de liaison permettant de modéliser les relations multiples :
  - Contract\_Services : classe de liaison qui permet de définir les services contracturés
  - FreeUnitsTariff : classe de liaison entre les plans tarifaires et les unités gratuites assignés implicitement.



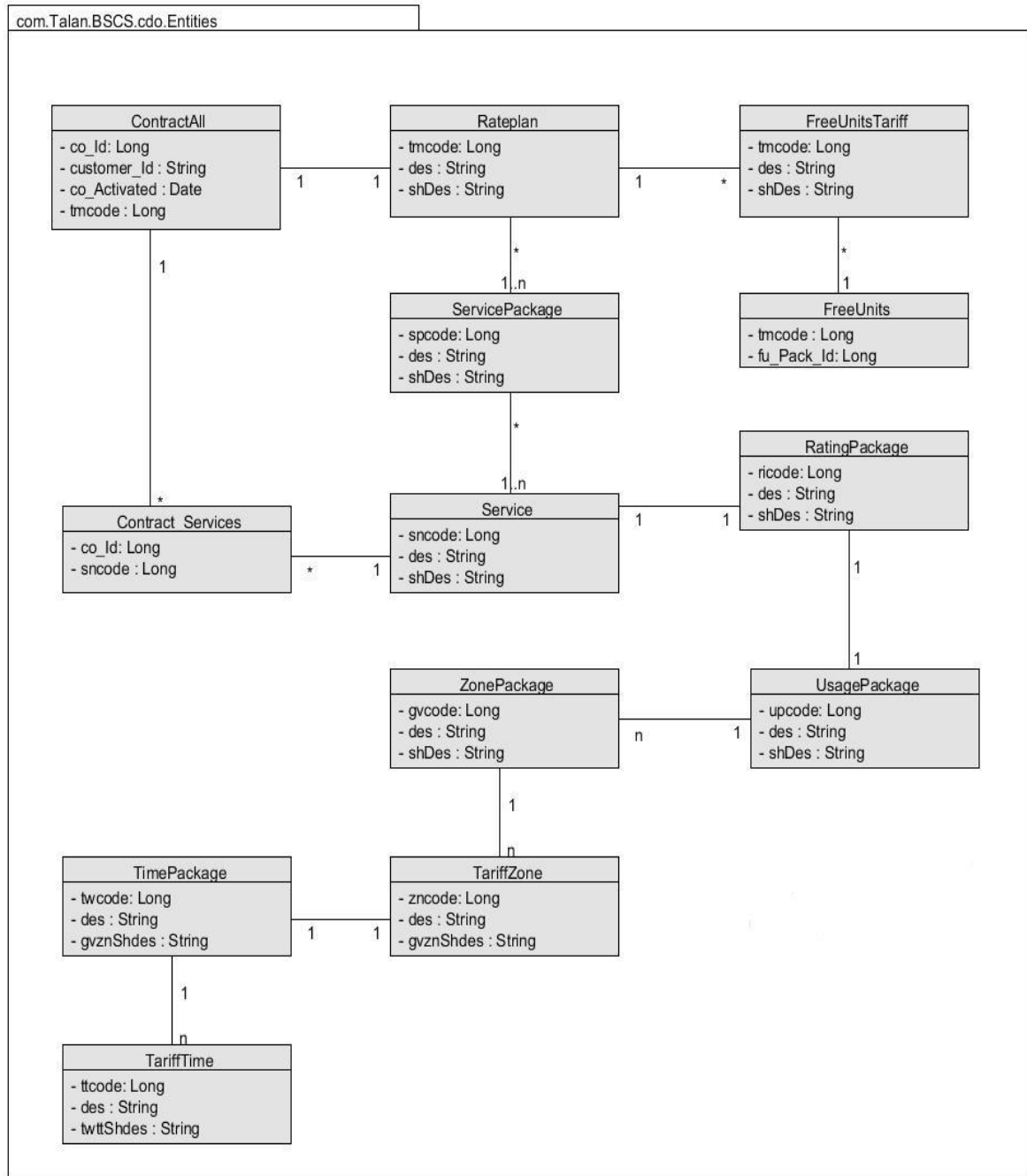


Figure 15 Diagramme de classe du package Entities

### 2.2.2. Package DAO

Ce package permet de faire la séparation entre la couche de services représentant la logique métier et la couche de persistance, et garantit ainsi un niveau d'abstraction pour la couche supérieure envers les mécanismes de persistance.

Parmi les éléments de ce package illustrés sur la figure 16 nous trouvons :

- Generic DAO : cette classe abstraite permet de définir les opérations standard d'accès aux données pour un type générique, ainsi les classes de DAO concrètes qui l'étendent n'auront qu'à spécifier l'objet persistant impliqué et par conséquent permet d'éviter des opérations répétitives au niveau de chaque DAO.
- DaoFactory : le patron de conception Factory permet de déléguer l'instanciation des objets à une seule classe. Cette méthode permet d'encapsuler la création des objets. Nous utilisons cette approche pour encapsuler l'instanciation des classes d'accès aux données en cette seule classe.

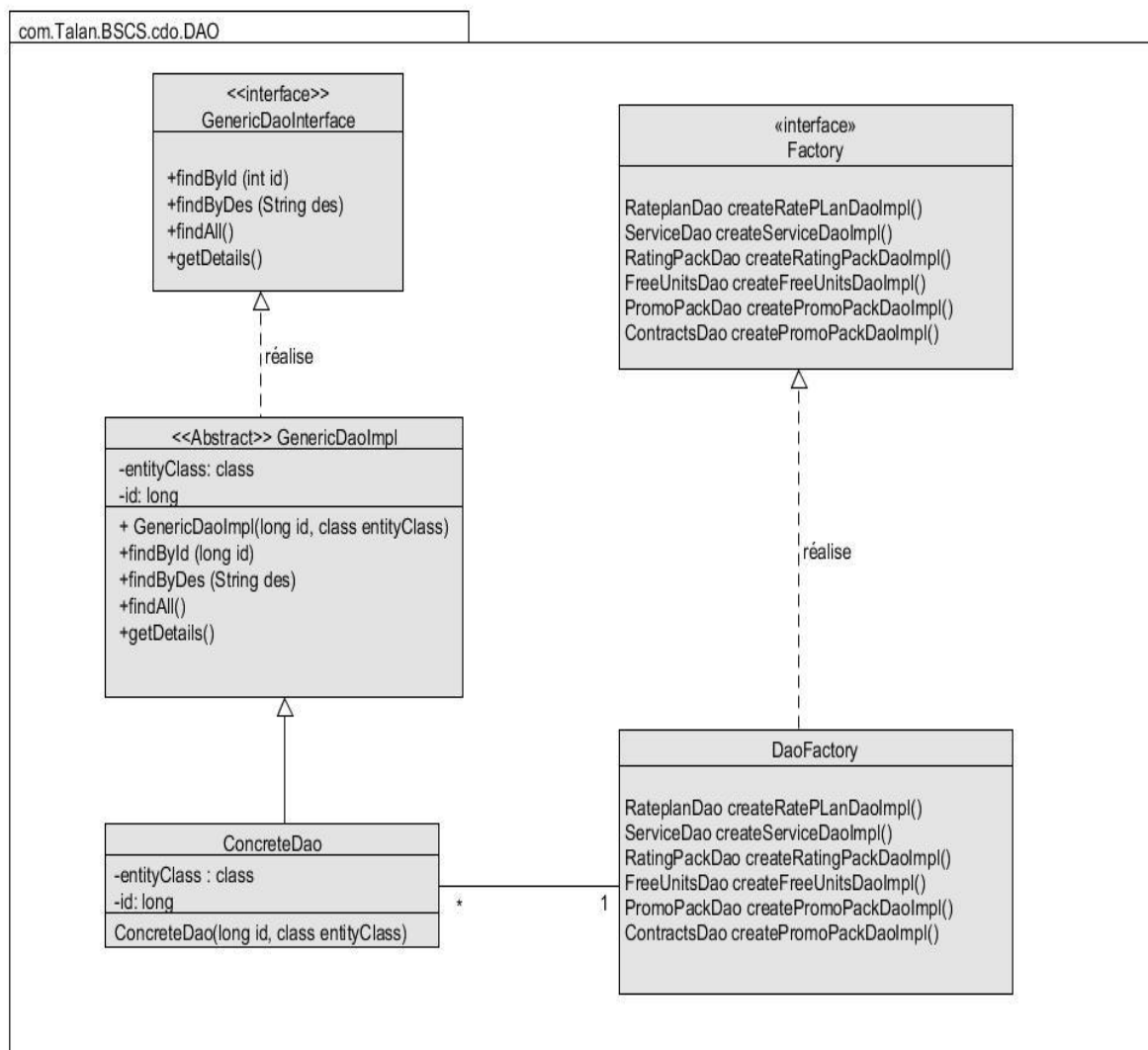


Figure 16 Diagramme de classe du Package DAO

### 2.2.3. Package DTO

Le Package « DTO », illustré sur la figure 17, définit les objets de transfert de données qui sont des entités distinctes exposant une nouvelle représentation du modèle de données dans le but est de définir la manière dont les données sont reçues et renvoyées par un service. Ces données sont épurées et pensées pour le transport des informations fournies par un ou plusieurs DAO en vue de les encapsuler en une structure, facilitant ainsi leur manipulation au niveau des services [7].

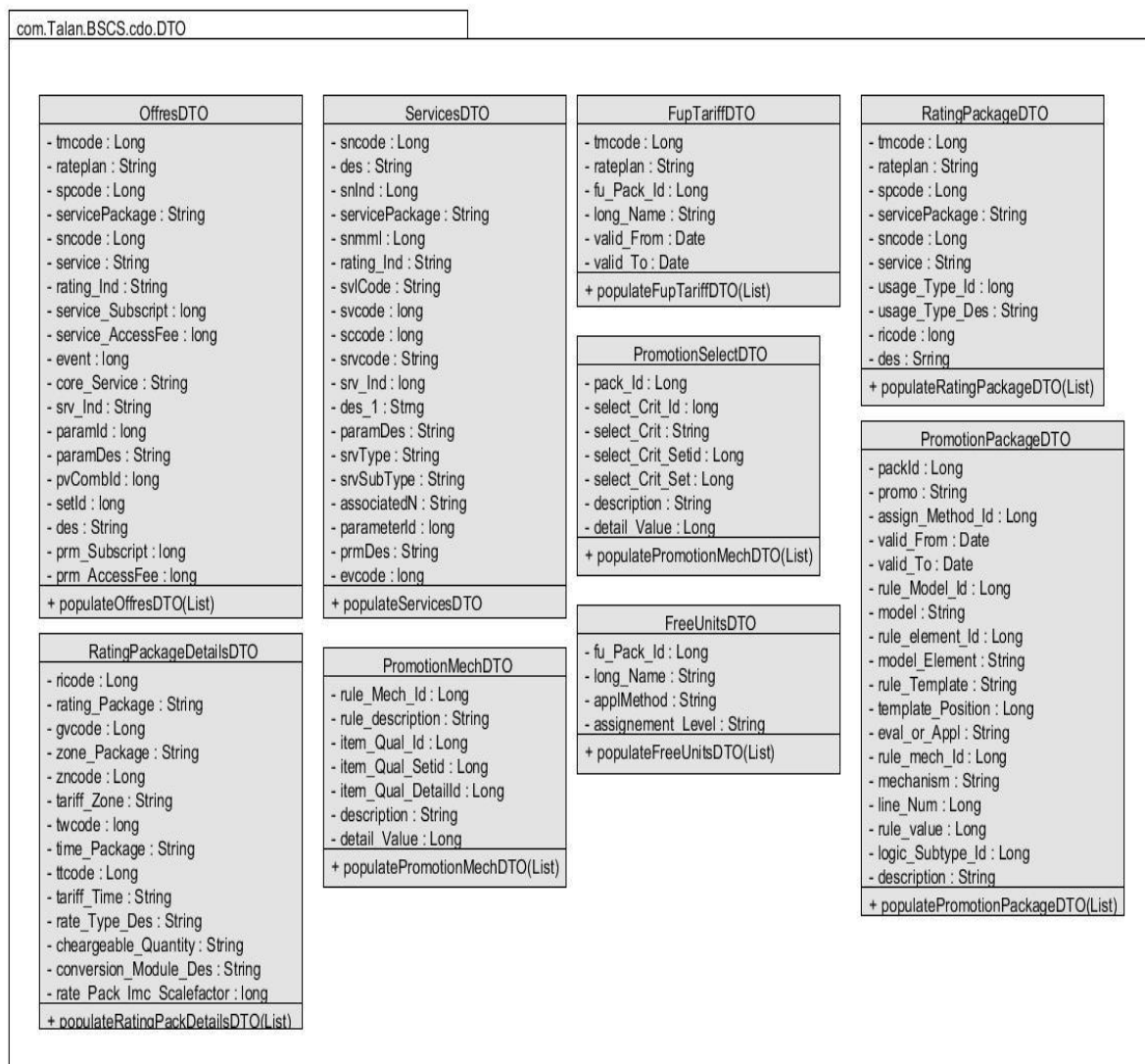


Figure 17 Diagramme de classe du package DTO

### 2.2.4. Package Services

Ceux sont les composants qui implémentent les différents scenarios des cas d'utilisations spécifiés auparavant. Nous trouvons parmi ces éléments illustrés sur la figure 18 :

- DataExtraction : c'est la classe qui gère les opérations d'extraction des différentes données.
- DataAnalysis : c'est la classe qui gère les opérations d'analyse.
- SaveExcel : la classe permettant de générer le catalogue des offres sous le format Excel.
- Report : la classe permettant de générer le rapport analytique.

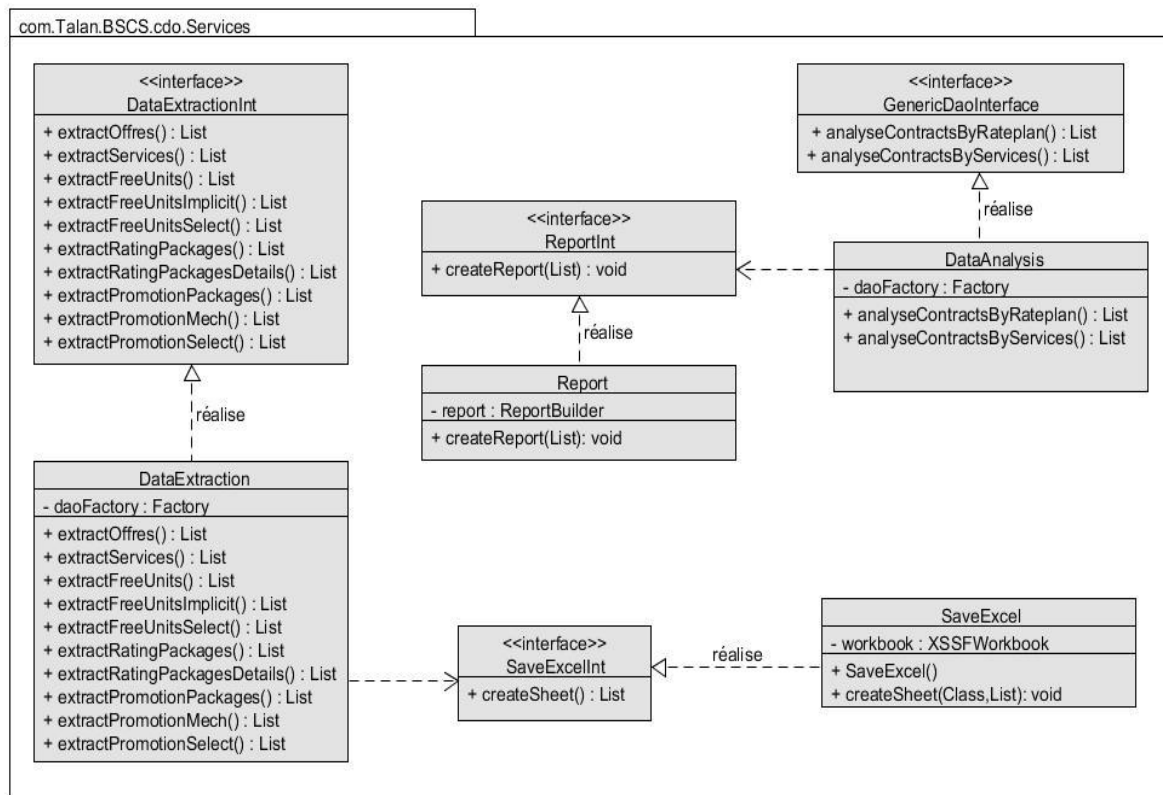


Figure 18 Diagramme de classe du package Services

### 2.2.5. Package Utilities

Les classes utilitaires regroupent des constantes ou des méthodes. Celles-ci sont définies en static, ce qui évite de devoir instancier la classe pour obtenir un résultat.

### **2.2.6. Package Controllers**

Ces composants de comportement seront associés aux différentes interfaces utilisateurs. Les méthodes de ces classes se chargeront donc de gérer la navigation entre les différentes pages, les événements émis par l'utilisateur ou encore la validation de ses entrées. C'est à travers ces classes que les pages accéderont aux données qu'ils auront à afficher.

## **2.3. Diagrammes de séquence**

Après avoir expliqué le comportement statique du système à travers le diagramme de modèles et les diagrammes de classes, nous allons maintenant exposer son comportement dynamique à travers des diagrammes de séquence relatifs aux scénarii de fonctionnement du système.

### **2.3.1 Diagramme de séquence relatif au cas d'utilisation « extraction des données »**

L'opération d'extraction des données, comme représenté sur la figure 19, nécessite le passage par plusieurs étapes, où l'acteur initie cette opération par l'invocation du service d'extraction à travers l'interface utilisateur. Cette demande sera gérée par le contrôleur qui invoque le service d'extraction *DataExtraction*. L'instance de cet objet récupère les données issues des instances des différents DAO par l'intermédiaire de l'objet *daoFactory*. Une fois ces données sont extraites, elles sont encapsulées sous forme d'objets DTO et passées au contrôleur pour mettre à jour l'interface utilisateur avec les résultats.

Finalement si l'utilisateur demande la sauvegarde des résultats d'extraction, l'instance *SaveExcel* est sollicitée, à travers l'instance *DataExtraction* par le flux de données résultants de l'opération d'extraction qui est ensuite convertit en un fichier Excel.

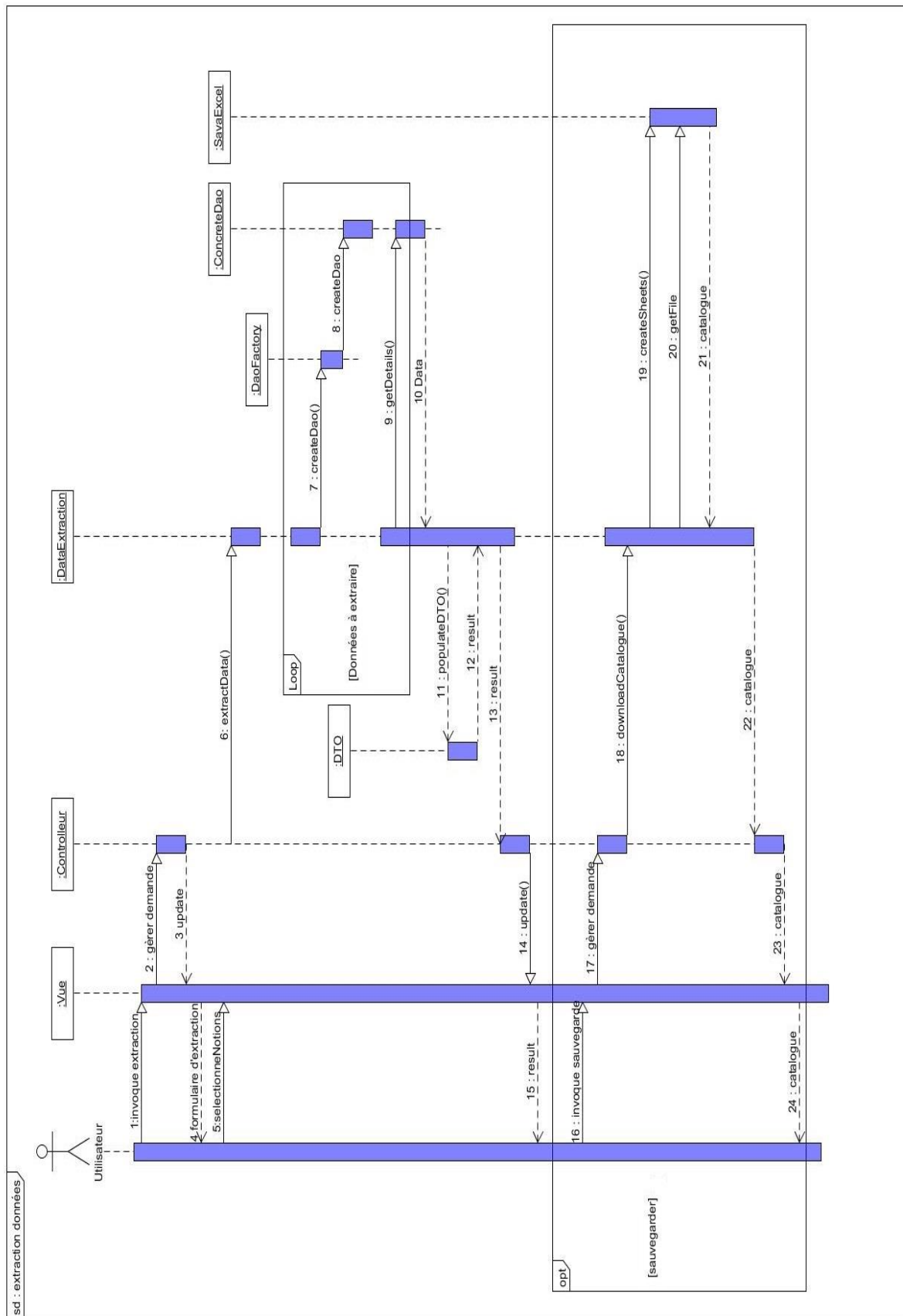


Figure 19 Diagramme de séquence de l'extraction des données

### 2.3.2 Diagramme de séquence relatif au cas d'utilisation « analyse des données »

L'opération d'analyse des données, illustrées sur la figure 20, nécessite elle-même le passage par plusieurs étapes, où l'acteur initie cette opération par l'invocation du service d'analyse via l'interface utilisateur. Cette demande sera gérée par le contrôleur qui invoque le service d'extraction DataAnalysis. L'instance de cet objet récupère les données issues des instances des différents DAO par l'intermédiaire de l'objet daoFactory.

Une fois ces données sont extraites, elles sont encapsulées sous forme d'objets DTO et passées au contrôleur pour mettre à jour l'interface utilisateur avec les résultats.

Finalement si l'utilisateur demande la sauvegarde des résultats d'analyse, les données sont passées à l'instance de l'objet Report, pour créer un rapport analytique, qui est passé en résultat à l'utilisateur.

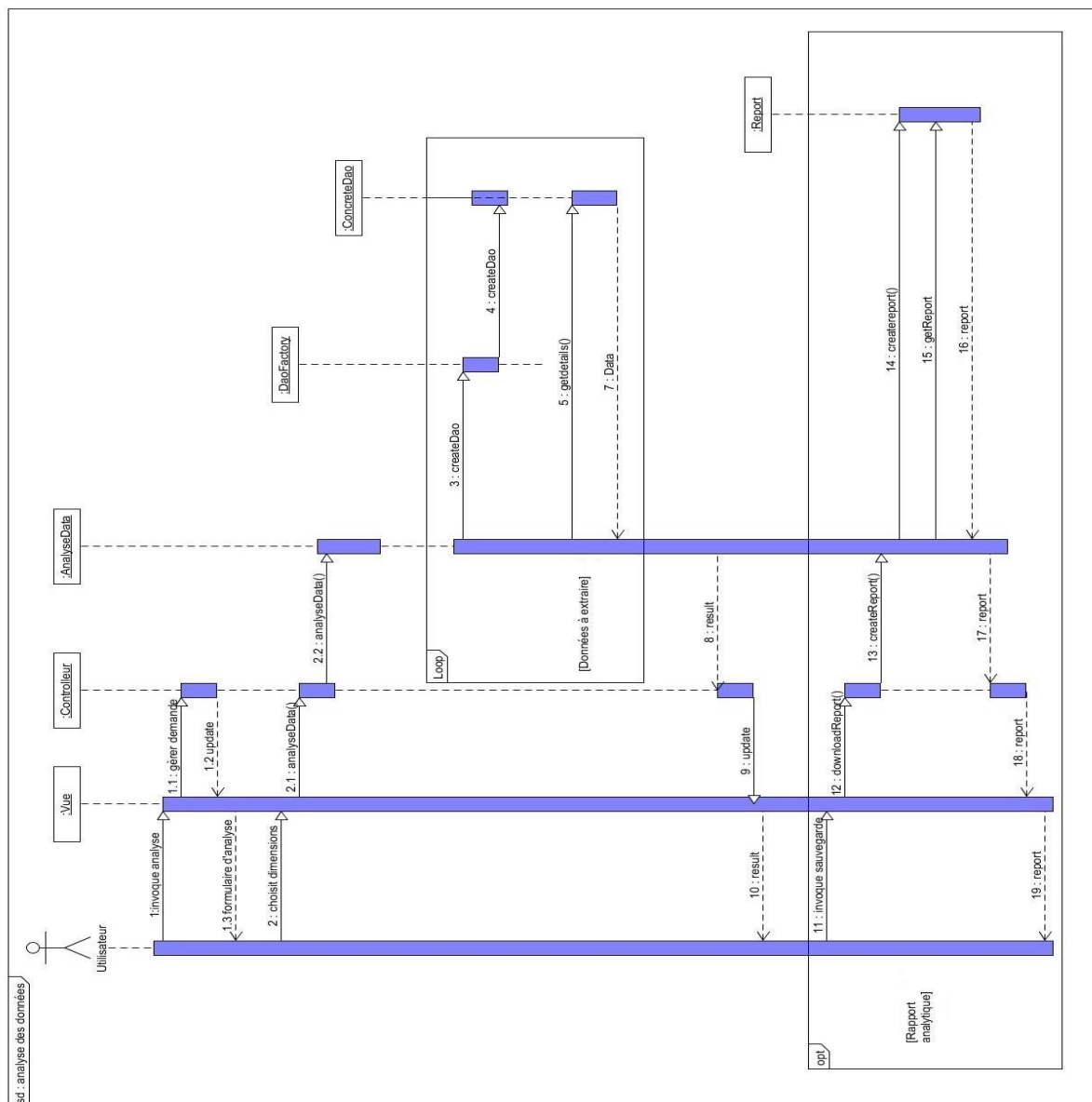


Figure 20 Diagramme de séquence de l'analyse des données

## **Conclusion**

Nous avons traité dans ce chapitre l'étude conceptuelle de notre système. Nous avons présenté, dans un premier lieu son architecture globale. Nous avons décrit par la suite son architecture détaillée en commençant par la présentation de ses principaux modules à l'aide du diagramme de modèles, que nous avons détaillé par la suite à travers des diagrammes de classes. Nous nous sommes également intéressés à l'aspect dynamique du système en élaborant les diagrammes de séquences des principales fonctionnalités. Nous entamons à présent la partie relative à la réalisation de notre application.



# Chapitre 5 : Réalisation

---

## Introduction

Nous avons suivi un enchaînement logique qui nous a permis de développer notre application. La validation de notre approche ne peut être confirmée que par la présentation des résultats obtenus et la mise en valeur des gains réalisés. C'est pour cela que nous consacrons ce chapitre à la concrétisation des idées antérieurement développées. Le présent chapitre décrira la réalisation de notre application. Il commencera tout d'abord, par la présentation de l'environnement matériel et logiciel, des choix technologiques ,suivi par la description des IHM de l'application réalisée, tout en détaillant les fonctionnalités de notre système.

## 1. Environnement de travail

Nous présentons dans cette section l'environnement matériel ainsi que celui logiciel utilisés pour le développement de notre application.

### 1.1. Environnement matériel

Pour mener à bien ce projet il a été réalisé par un ordinateur portable équipé d'un microprocesseur d'une vitesse d'exécution de 2,3 GHz, une mémoire vive (RAM) de 6 Go et une capacité de stockage de 600 Go. Ces ressources matérielles assurent le bon fonctionnement des logiciels utilisés pour l'implémentation.

### 1.2. Environnement logiciel

Cette partie est consacrée à la présentation des langages et des outils composant notre environnement de développement.

#### 1.2.1 Langage de développement

Le langage Java est un langage de programmation informatique orienté objet. La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications.

#### 1.2.2 Environnement de développement

Pour être conforme à l'architecture pré-spécifiée, notre solution est implémentée sous la forme d'une application web conforme à la plate-forme Java EE développée sous la plateforme JDK 1.8. Pour ce faire nous avons eu recours aux outils suivants : Eclipse Luna

comme environnement de développement intégré. Nous avons utilisé apache Tomcat 8.0.15 comme conteneur de servlet web, Maven comme outil de build et Oracle 10g comme serveur de base de données relationnelle.

### **Eclipse Luna**

Eclipse est un environnement de développement libre, extensible et polyvalent. Son objectif est de produire et fournir divers outils gravitant autour de la réalisation de logiciel, englobant les activités de codage logiciel proprement dites (avec notamment un environnement de développement intégré), mais aussi de modélisation, de conception, de test, de reporting, etc. Son environnement de développement notamment vise à la généricité pour lui permettre de supporter n'importe quel langage de programmation.

### **Apache Maven**

Maven est un outil de build, de gestion des dépendances et du cycle de vie pour des projets Java. Fruit des meilleures pratiques concernant le build de projets de la fondation Apache, Maven est un outil très puissant, qui saura alléger efficacement le temps de maintenance du build de notre projet, tout en nous permettant une industrialisation et une répétabilité complète de toutes les étapes. Maven utilise un fichier de configuration XML appelé POM (Project Object Model) afin de décrire un projet logiciel, ses dépendances avec des modules externes et l'ordre à suivre pour sa production. Il est livré avec un grand nombre de tâches prédéfinies, comme la compilation de code Java ou encore sa modularisation. Dans le cadre de notre projet, nous avons utilisé Maven 2 dans le but de gérer nos dépendances.

## **2. Choix Technologiques**

Nous avons été amenés à choisir certaines parmi les technologies Java EE pour couvrir certains aspects de notre application, et ce à tous les niveaux de notre architecture. Nous présentons dans la suite celles qui ont été le plus utilisées.

### **2.1. JSF/Primefaces**

JSF est un framework MVC reprenant le modèle des interfaces utilisateur locales. Il permet de créer des pages web dynamiques en utilisant AJAX sans avoir à programmer en JavaScript. Les modèles utilisés dans JSF sont représentés par des Beans Java, tandis que les vues sont sous la forme de composants Java sur le serveur (pages jsp ou xhtml) qui sont transformés en des pages html chez le client. De plus, l'intégration de Primefaces une bibliothèque de composants riches pour JSF, nous fournira une panoplie de composants supplémentaires prêt

à l'emploi et qui prennent en charge l'Ajex ce qui nous épargne ainsi l'écriture manuelle du code JavaScript.

## **2.2. Google Chart API**

L'API Google Chart est un outil qui permet de créer facilement des graphiques à partir d'une liste de données et de paramètres et de les intégrer dans une page Web. A l'origine il était un outil interne pour soutenir l'incorporation rapide de graphiques dans les applications de Google, mais peu après Google a pensé que ce serait un outil utile pour ouvrir aux développeurs web.

## **2.3. Spring**

Spring est un conteneur léger, plutôt simple d'utilisation, extensible, et qui permet un réel découplage entre les composants grâce à son approche basée sur les notions d'inversion de contrôle et de programmation par aspect. Sa facette que nous avons le plus exploité est l'injection de dépendances. En effet grâce à cette inversion de contrôle, les composants sont nettement plus indépendants de leur environnement d'exécution : c'est ce conteneur léger qui gère le cycle de vie des composants de l'application, et qui injecte les dépendances de manière appropriée. Ce qui révèle très utile pour une architecture multicouches comme celle de notre application.

## **2.4. Dynamic Reports**

Dynamic Reports est un outil qui permet de créer des designs dynamiques de rapport. Cet outil performant permet de rapidement créer des rapports et de produire des documents qui peuvent être affichés, imprimés ou exportés dans de nombreux formats populaires tels que PDF, Word etc. Le rapport est créé à travers un code Java pur et simple où on peut mettre en œuvre une logique qui décide comment le rapport va ressembler, contrairement aux rapports statiques où la conception définie ne peut être modifiée lors de l'exécution.

## **2.5. EclipseLink/JPA**

EclipseLink est une implémentation de JPA qui n'est autre qu'une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java. EclipseLink offre la possibilité de créer une couche d'abstraction d'accès aux bases de données. Son utilisation rend l'application indépendante du SGBD auquel elle fait appel. Un simple changement des paramètres de configuration permet de passer par exemple d'une base de données à une autre. C'est la

garantie d'avoir des applications plus homogènes et facilement migrable.

### 3. Scénarios d'exécution

Après avoir présenté les principaux outils et technologies utilisés dans la réalisation de notre application, nous présentons, à travers un enchaînement de captures d'écran, des scénarios d'exécution donnant un aperçu général sur le fonctionnement de notre solution.

#### 3.1 Connexion à l'environnement

La première interface correspondant au démarrage de l'application est l'interface de connexion à l'environnement d'extraction (figure 21).

L'utilisateur doit fournir les paramètres d'accès à la base :

- Adresse du serveur de données
- Numéro de port
- La base de données
- Le nom d'utilisateur et le mot de passe

Une fois les paramètres saisis correspondent à une base de données BSCS l'utilisateur pourrait accéder à l'application.

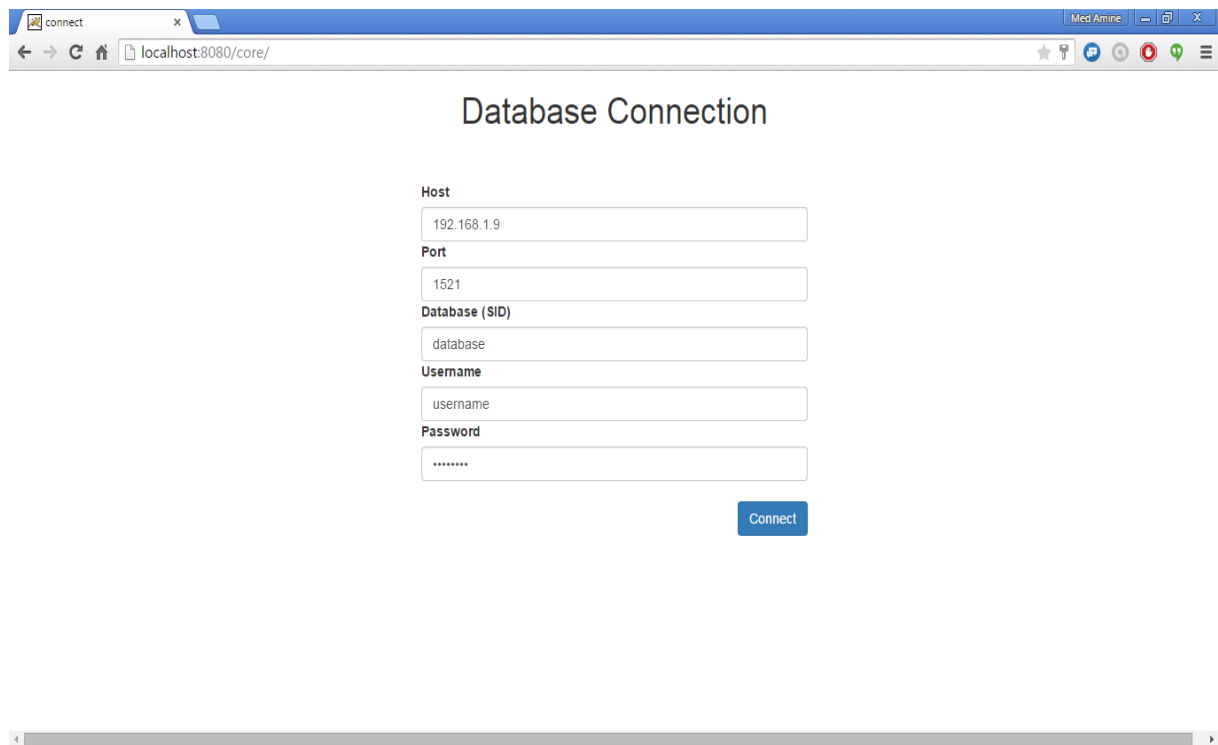
The image shows a web browser window with the address bar displaying 'localhost:8080/core/'. The page title is 'Database Connection'. The form contains the following fields: 'Host' with the value '192.168.1.9', 'Port' with the value '1521', 'Database (SID)' with the value 'database', 'Username' with the value 'username', and 'Password' with masked characters '\*\*\*\*\*'. A blue 'Connect' button is located at the bottom right of the form.

Figure 21 Interface de connexion à l'environnement d'extraction

### 3.2. Extraction des données BSCS

Une fois l'utilisateur est connecté à un environnement d'extraction, il peut procéder avec l'opération d'extraction des données via le menu « Extraction des données » illustré sur la figure 22. Où il consulte les options de filtrage associé à chaque notion BSCS présente et pourrait choisir des valeurs spécifiques pour un ou plusieurs paramètres (figure 23) pour simplifier l'opération de recherche.

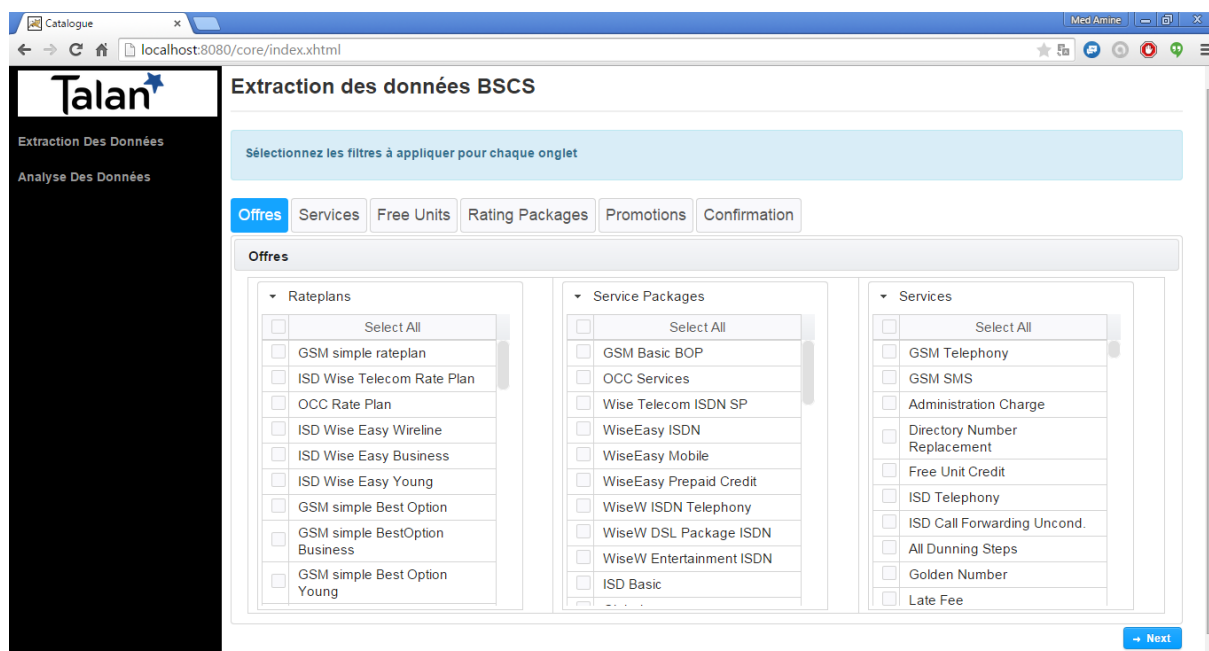


Figure 22 Menu extraction des données BSCS

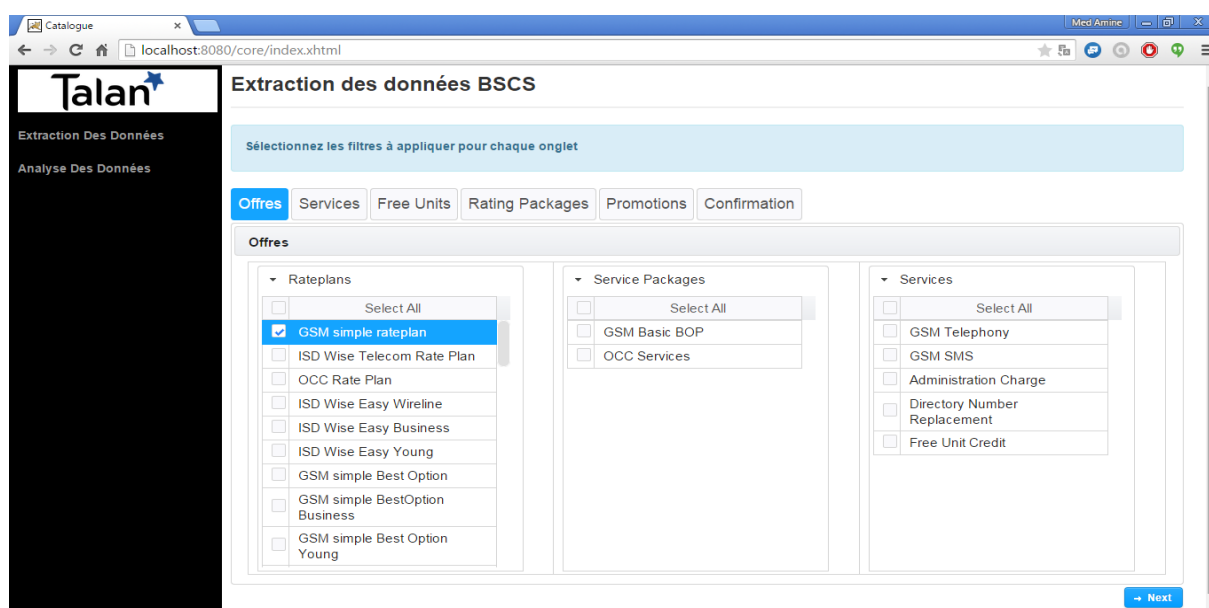


Figure 23 Aperçu d'application de filtre

Pour chaque notion ayant une liaison directe avec une les offres, l'utilisateur pourrait aussi filtrer par une jointure entre les deux notions (figure 24). Ainsi tous les champs seront mis à jour automatiquement pour correspondre offres déjà sélectionnées.

The screenshot shows a web application interface for 'Extraction des données BSCS'. The 'Rating Packages' tab is active, and the 'Lier Aux Plans Tarifaires' checkbox is checked and circled in red. The 'Rateplan' section has four items selected: 'GSM simple rateplan', 'ISD Wise Telecom Rate', 'OCC Rate Plan', and 'ISD Wise Easy Wireline'. The 'Service Packages' section has 'GSM Basic BOP' selected. The 'Services' section has 'GSM Telephony' and 'GSM SMS' selected. The 'Usage Type' section has 'Outbound' and 'Inbound' selected. The 'Rating Packages Details' section shows 'Rating Packages', 'Zone Packages', 'Tariff Zone', 'Time Packages', 'Tariff Time', 'Rate Type', and 'Chargeable Quantity'.

Figure 24 Liaison entre notions BSCS

A la fin de cette procédure une récapitulation des sélections explicites de l'utilisateur lui sera affiché (figure 25) et il devrait confirmer ses choix pour procéder à l'affichage des résultats.

The screenshot shows the 'Confirmation' tab of the 'Extraction des données BSCS' interface. The 'Confirmation' section displays the selected options: 'Offres: GSM simple rateplan', 'Services: GSM Telephony', 'Free Units: LA: 3 hours free', and 'Rating Packages: Rateplan: GSM simple rateplan, Service package: GSM Basic BOP, Service: GSM Telephony'. The 'Submit' button is highlighted.

Figure 25 Confirmation des choix

Les résultats sont affichés sous forme de tableaux (figure 26) qui permettent d'appliquer des filtres combinés dynamiques sur toutes les colonnes.

TMCODE	Rateplan	SPCODE	Rateplan	SNCODE	Service	Usage Type Id	Usage Type Des	RICODE	Rating Package
1	GSM simple rateplan	3	GSM Basic BOP	1	GSM Telephony	2	Outbound	59	GSM Tel Peak/Off Peak
1	GSM simple rateplan	3	GSM Basic BOP	2	GSM SMS	2	Outbound	2	Rating Pkg SMS
8	GSM simple Best Option	3	GSM Basic BOP	1	GSM Telephony	2	Outbound	1	Rating Pkg Tel
8	GSM simple Best Option	3	GSM Basic BOP	2	GSM SMS	2	Outbound	2	Rating Pkg SMS
9	GSM simple BestOption Business	3	GSM Basic BOP	1	GSM Telephony	2	Outbound	11	Rating Pkg Tel Business

Figure 26 Résultats de l'opération d'extraction

L'utilisateur demande de sauvegarder les résultats en appuyant sur le bouton « Export Catalogue » est un fichier sous format Excel sera généré (figure 27).

TMCODE	Rateplan	SPCODE	Rateplan	SNCODE	Service	Usage Type Id	Usage Type Des	RICODE	Rating Package
1	GSM simple rateplan	3	GSM Basic BOP	1	GSM Telephony	2	Outbound	59	GSM Tel Peak/Off Peak
1	GSM simple rateplan	3	GSM Basic BOP	2	GSM SMS	2	Outbound	2	Rating Pkg SMS
8	GSM simple Best Option	3	GSM Basic BOP	1	GSM Telephony	2	Outbound	1	Rating Pkg Tel
8	GSM simple Best Option	3	GSM Basic BOP	2	GSM SMS	2	Outbound	2	Rating Pkg SMS
9	GSM simple BestOption	3	GSM Basic BOP	1	GSM Telephony	2	Outbound	11	Rating Pkg Tel

Figure 27 Sauvegarde des résultats d'extraction

### 3.3. Analyse des données BSCS

Un utilisateur connecté pourrait aussi bénéficier de la fonctionnalité d'analyse via le menu « Analyse des Données » et pourrait par suite choisir les dimensions de son analyse (figure 28).

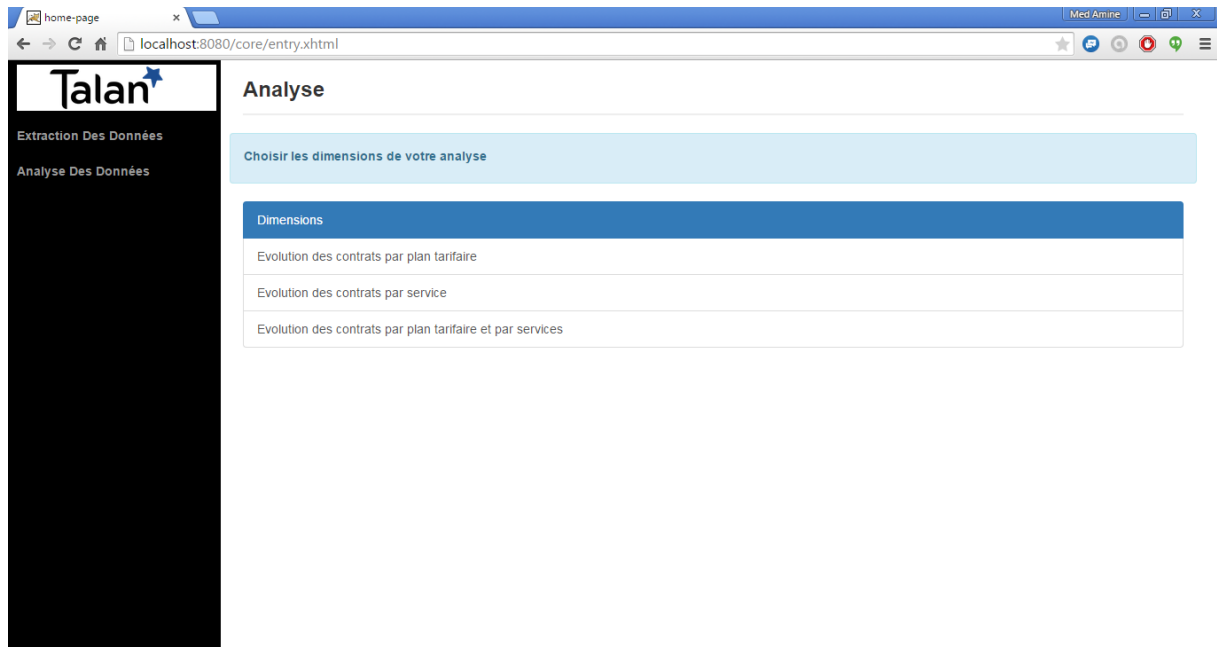


Figure 28 Choix des dimensions d'analyse

L'utilisateur pourrait ensuite fixer des valeurs spécifiques pour chaque axe d'analyse et procéder à l'affichage des résultats (figure 29).

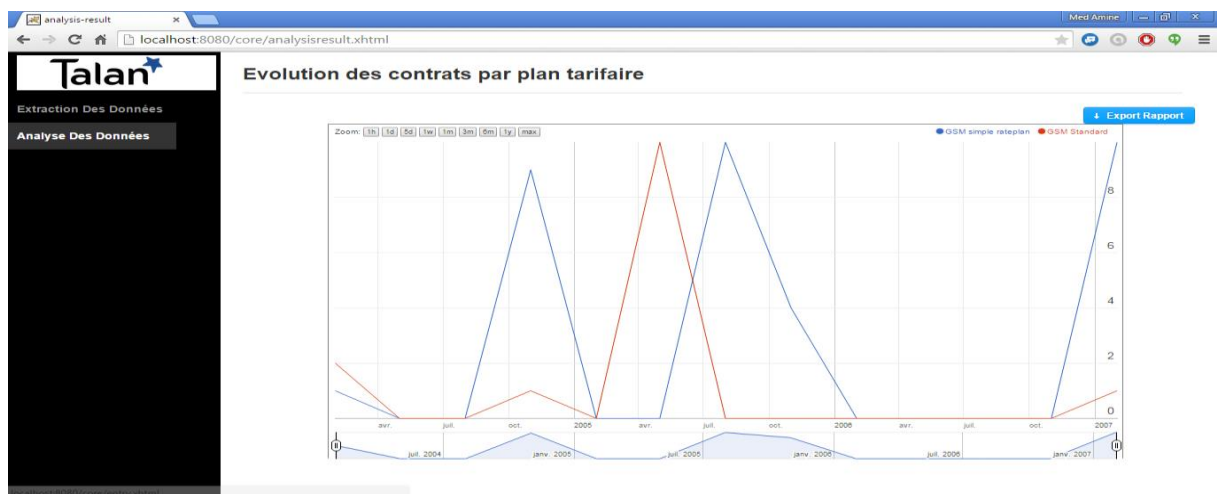


Figure 29 Aperçu résultat d'analyse



Enfin si l'utilisateur exige une sauvegarde de cette opération il pourrait générer un rapport analytique sous format PDF en cliquant sur le bouton « Export rapport ».

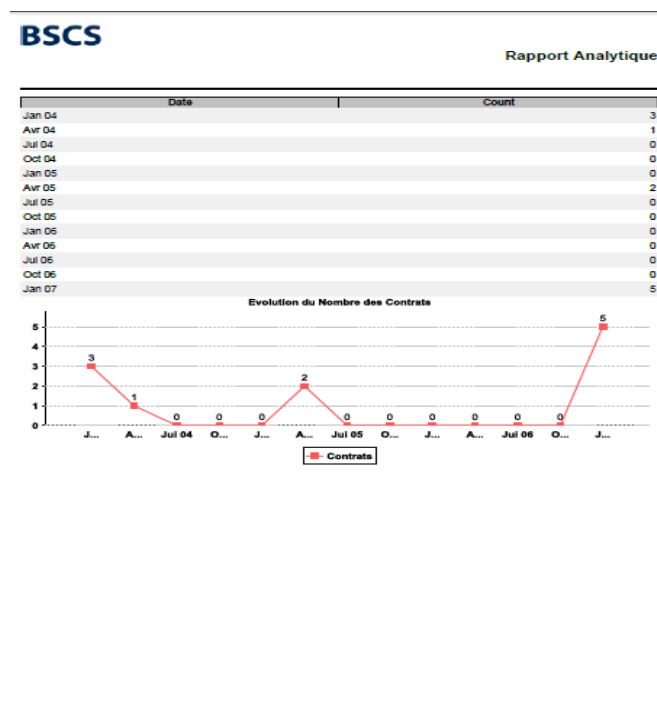


Figure 30 Aperçu rapport analytique

## Conclusion

Au cours de ce chapitre, nous avons décrit les plateformes matérielles et logicielles sur lesquelles nous avons construit notre application. Nous avons ensuite présenté des scénarios d'exécution étayé de captures écran des interfaces de notre solution.

A ce stade, nous atteignons la fin de l'étude du projet.

## Conclusion Générale

Au terme de ce rapport, nous dressons le bilan complet de ce travail nous avons essayé de présenter tout ce qui s'avère indispensable pour décrire clairement toutes les phases du projet, le présent document a détaillé les étapes par lesquelles nous sommes passés pour mettre en place une solution de génération d'un catalogue des offres, qui permet de regrouper différents concepts liés au système de facturation téléphonique BSCS.

Pour aboutir à cette solution, nous avons tout d'abord commencé par la présentation du contexte du projet suivie par différentes notions théoriques liés à notre système, en suite nous avons analysé les besoins et exigences relevés. Une fois ces besoins spécifiés et les cas d'utilisation détaillés, nous nous sommes intéressés à l'étude conceptuelle de notre projet en commençant par l'architecture adoptée, pour aboutir par la suite à une conception détaillée, qui met l'accent sur l'aspect statique et dynamique du système. Finalement, nous avons abordé l'étape de réalisation en exposant l'environnement du travail, ainsi que certains choix technologiques adoptés lors de la réalisation de notre système pour finir avec une description de certains scénarios d'exécution.

Ce travail nous a été très instructif de point de vue des connaissances acquises. Il nous a procuré une opportunité pour confirmer une fois de plus nos connaissances dans le développement Java/J2EE et d'étudier et d'utiliser une panoplie de technologies et d'outils.

Hormis le côté technique, ce projet a été une opportunité pour appréhender le travail dans une hiérarchie professionnelle au sein d'une grande société et les difficultés inhérentes comme la répartition du temps ainsi que les bonnes pratiques nécessaires à la réalisation d'un produit de qualité.

Pour conclure, nous estimons avoir satisfait les objectifs initialement fixés. Néanmoins, nous tenons à présent à souligner quelques extensions intéressantes de notre projet. Grâce à son caractère extensible notre réalisation peut être enrichie par l'inclusion d'autres données BSCS que le présent sujet pas traités, aussi pour la partie analytique qui pourrait être étendue soit par l'intégration d'un moteur OLAP qui serait utile pour la prise en charge des requêtes analytiques complexes ou par une structure de données associée à un moteur de règle définissant la liaison entre les différents axes d'analyse qui permettrait d'automatiser l'opération tout en gardant son aspect algorithmique.

## Webographie

[1] [http : //www.talan.fr](http://www.talan.fr), [Consulté le 13 mai 2015].

[2] [www.freewebs.com](http://www.freewebs.com) <http://www.freewebs.com/fresma/Formation/UML/ProcessuUnifie> [Consulté le 13 mai 2015].

[3] [www.uml-sysml.org](http://www.uml-sysml.org) <http://www.uml-sysml.org/images-du-site/2TUP.gif/view> [Consulté le 7 juin 2015].

[4] [whatis.techtarget.com](http://whatis.techtarget.com) <http://whatis.techtarget.com/definition/slice-and-dice> [Consulté le 16 mai 2015].

[5] [searchdatamanagement.techtarget.com](http://searchdatamanagement.techtarget.com) <http://searchdatamanagement.techtarget.com/feature/Best-practices-for-data-warehouse-access-and-reports>

[6] [www.uml-diagrams.org](http://www.uml-diagrams.org) <http://www.uml-diagrams.org/uml-24-diagrams.html> [Consulté le 19 mai 2015].

[7] [www.uml-diagrams.org](http://www.uml-diagrams.org) <http://www.developpez.net/forums/d1430872/java/developpement-web-java/frameworks/utilite-design-pattern-dto/> [Consulté le 22 mai 2015].

[8] [kalyan-city.blogspot.com](http://kalyan-city.blogspot.com) <http://kalyan-city.blogspot.com/2010/11/what-is-report-meaning-features-or.html> [Consulté le 2 juin 2015].

## Acronymes

**BSCS** : Business Support and Control System

**2TUP** : Two Track Unified Process

**UML** : Unified Modeling Language

**COFU** : Contract Only Free Units

**POFU** : Pooled Free Units

**POFUL** : Pooled Free Units for Large Accounts

**Java EE** : Java Enterprise Edition.

**API** : Application Programming Interface

**JSF** : Java Server Faces

**JPA** : Java Persistence API

**AJAX** : Asynchronous JavaScript and XML

**XHTML** : Extensible HyperText Markup Language.



## Diagramme de classe complet du package Entities

### 1. Services

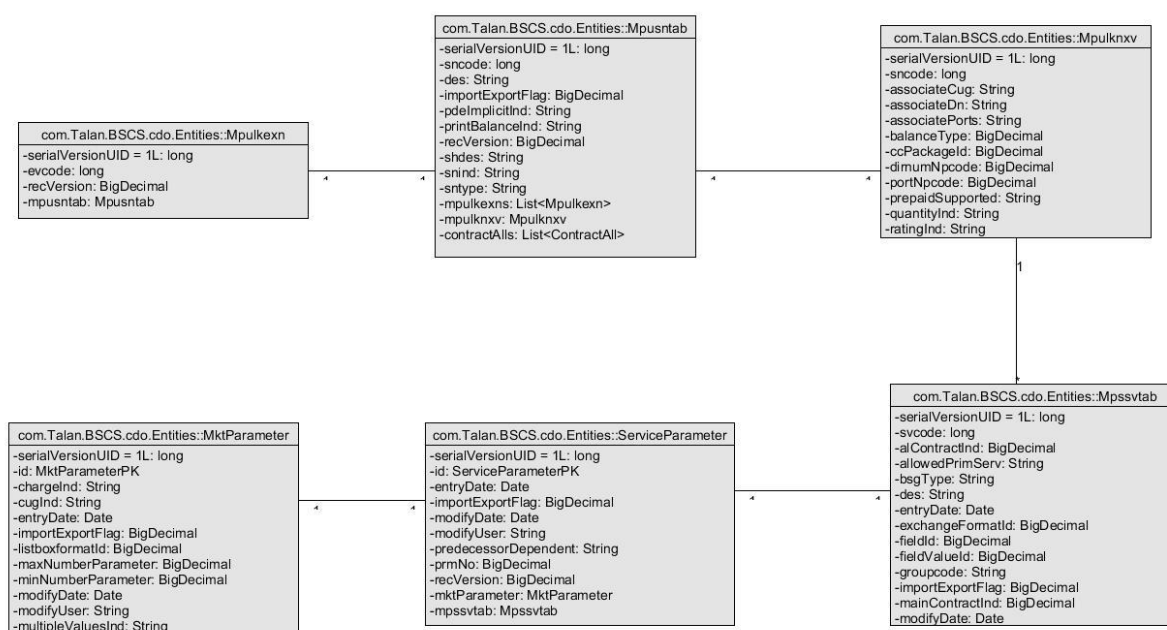


Figure 31 Diagramme de classe des entités de services

Classe Mpusntab : Définit les noms de services à utiliser par les plans tarifaires.

Classe Mpusntab : Définit le SVL code l'identifiant du service.

Classe Mpssvtb : Définit les services de base.

Classe Mpulkexn : Définit les codes des évènements des services.

Classe ServiceParameter : Lie le service à un marché.

Classe MktParameter : définit les paramètres du marché.

## 2. Free units

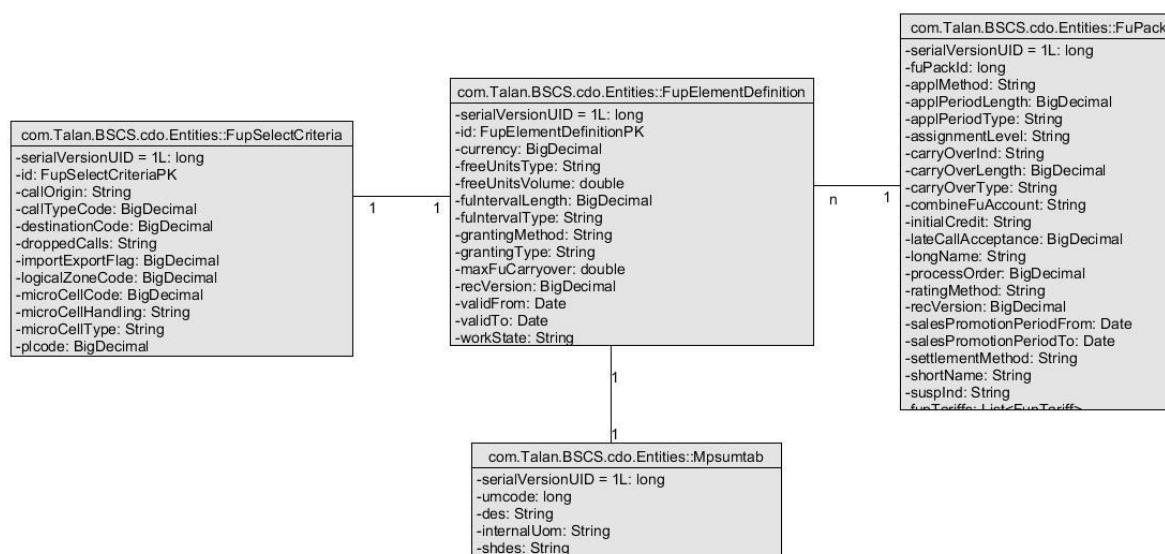


Figure 32 Diagramme de classe des entités des free units

Classe FuPack : Définit les paquets des unités gratuites

Classe FupElementDefinitin : Définit les éléments d'un paquet d'unités gratuites.

Classe FupSeleccriteria : Définit les critères de sélection d'un paquet d'unités gratuites.

Classe Mpsumtab : Définit les unités de mesures.

## 3. Promotions

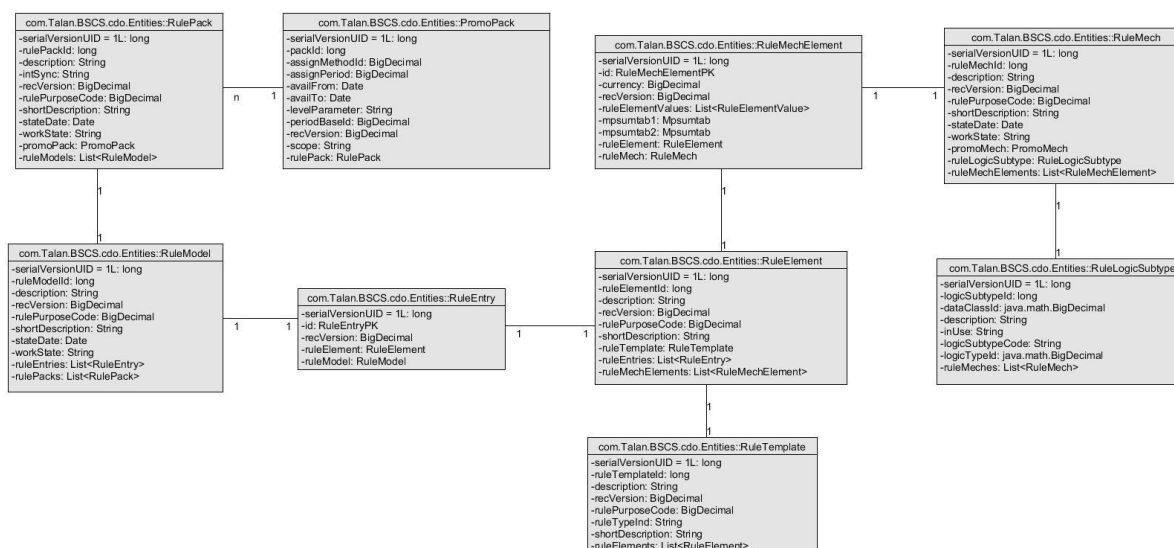


Figure 33 Diagramme de classe des entités des promotions

Classe PromoPack : Définit les paquets de promotions.

Classe RulePack : Définit toutes les règles incluses dans le système.

Classe RuleModel : Définit les modèles de règle.

Classe RuleElement : Définit les éléments de règle.

Classe RuleTemplate : Définit les modèles de règle.

Classe RuleMechElement: Contient tous les éléments de la règle avec la position définie pour chaque type de mécanisme.

Classe RuleLogicSubtype : Définit les sous types de chaque règle.

## 4. Rating Packages

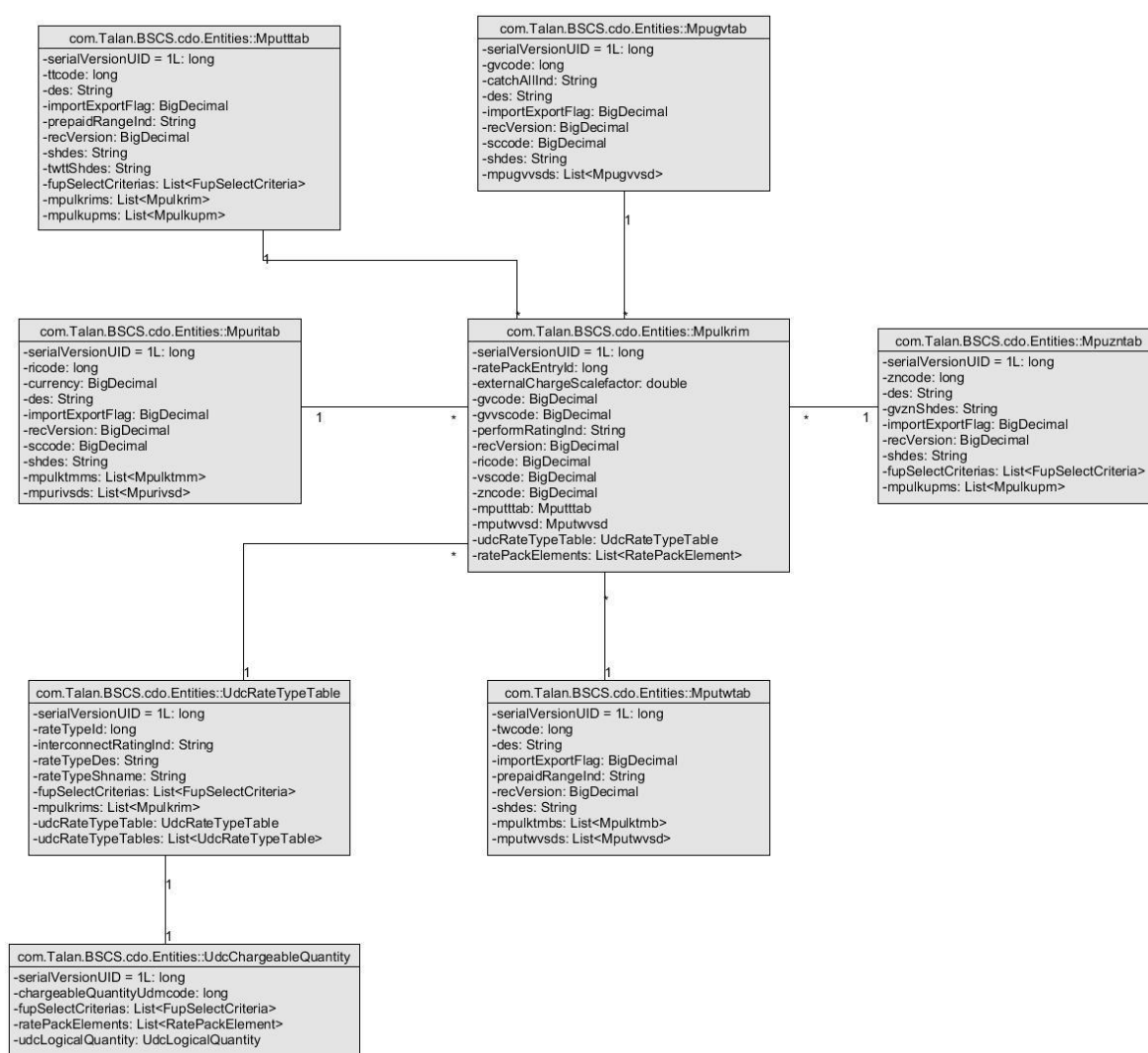


Figure 34 Diagramme de classe des entités des rating packages

Classe Mpulkrim : Définit les rating packages en production.

Classe UdcRateTypeTable : Définit les rate types.

Classe UdcChargeableQuantity : Définit le type des chargeable quantities.