

UNIVERSITE MOHAMMED V AGDAL
ECOLE MOHAMMADIA D'INGENIEURS



Filière : Génie Informatique
Option : Systèmes d'Information
Ingénierie & Qualité du Logiciel

Mémoire de Projet de Fin d'Etudes

N° INF 2012/41

**ETUDE, CONCEPTION ET REFONTE D'UNE
PLATEFORME BI**

Réalisé par :

Mlle Mounia OUHAJOU

M. Jihad TAWFIQ

Encadré par :

Prof. N. EL FADDOULI

M.R.AGOUDAL

M.M.BENYAHYA

Année Universitaire : 2011- 2012

UNIVERSITE MOHAMMED V AGDAL
ECOLE MOHAMMADIA D'INGENIEURS



Filière : Génie Informatique
Option : Systèmes d'Information
Ingénierie & Qualité du Logiciel

Mémoire de Projet de Fin d'Etudes

N° INF 2012/41

**ETUDE, CONCEPTION ET REFONTE D'UNE
PLATEFORME BI**

Réalisé par :

Mlle Mounia OUHAJOU

M. Jihad TAWFIQ

Soutenu le 31 mai 2012 devant le jury :

M. S. BENNANI

Professeur à l'EMI

Président

Mme. A. RETBI

Professeur à l'EMI

Rapporteur

M. N. EL FADDOULI

Professeur à l'EMI

Encadrant interne

M. R. AGOUDAL

OriginalSoft System

Encadrant externe

Année Universitaire : 2011- 2012

Dédicace

A mes très chers parents,

Tous les mots du monde ne sauraient exprimer l'immense amour que je vous porte, ni la profonde gratitude que je vous témoigne pour votre affection, votre soutien et les valeurs que vous m'inculquez toujours. Sans vous, je ne serais pas l'homme que je suis aujourd'hui.

Au grand Dr. Abdessamad Tamouro, mon mentor, second père et ami,

C'est grâce à nos innombrables heures passées ensemble que mes navires ne perdent plus le cap. Merci pour tout l'amour que tu m'accordes, pour toutes les leçons que tu m'apprends. Que cet humble travail puisse te rendre fier de moi.

A mon frère Nizar, mon meilleur ami et compagnon de lutte.

Que ce travail s'ajoute à nos combats menés ensemble. Sache que je crois toujours en toi et que tes réalisations sont des médailles que j'accroche fièrement à ma poitrine.

A mon oncle, M. Jamal Benomar,

Ton parcours extraordinaire depuis les geôles de la répression jusqu'au sommet des Nations-Unis est ma véritable inspiration.

A la très chère Khalti Khadija, pour tous les sentiments d'amour que tu me portes.

A ma sœur Nadine.

A la mémoire de mes grand-mères, du grand-père que je n'ai jamais connu.

A ma binôme très spéciale Mounia,

A mes amis et camarades :

Ado, Anouar, Halim, Hassan, Java, Joe, Kristina, Lamya, Mahdi, Mehdi, Mouna, Mustapha, Omar, Ouajih, Oussama, Petra, Rubio, Salma, Solaymane, Soufiane, Souhail The Boss et tous les autres.

Jihad

Dédicace

*A mon adorable mère et mon agréable et cher père,
Nul mot ne pourra exprimer ma gratitude et ma reconnaissance envers vous deux,
mes très chers parents. Je ne cesserais de vous remercier pour tous les sacrifices
que vous avez faits à mon égard. Vous êtes les étoiles qui ornent mon univers. Je
vous aime...*

*A mon cher frère et ami Habib,
Cette diction est un rappel pour que tout au long de ma vie, je remercie toujours
le ciel de t'avoir comme frère*

*A ma chère sœur confidente Karima,
Je n'aurais espéré avoir meilleure sœur. En guise de mon amour, je te dédie le
présent travail.*

*A mon tante et seconde mère Rqia,
Tu m'as tant couvert d'attention. Merci pour ton soutien et appui.*

*A mes grands-parents maternels,
Merci pour toute l'attention que vous portez à mon parcours et mon travail.
A la mémoire de mes grands-parents paternels,
A mes oncles, tantes, cousins et cousines,*

*A mon binôme,
Merci pour ton appuie, ton soutien permanent et ta compréhension.*

*A mes chères amies,
Amina, Amal, Asma, Assia, Fatima, Ghizlane, Imane, Soukaina, Yasmine*

*A mes amis et camarades,
Abrar, Aicha, Aziz, Fadwa Hasna, Houda, Mariam, Maria, Meriem, Nabil,
Ouissam, Salah, Sara, Siham, Solaymane, Sophia, Smail, Redouane, Youssef,
Yassine, Wafaa, Wahiba ...*

Mounia

Remerciements

Au terme de ce projet, nous tenons à exprimer notre gratitude envers tous ceux qui ont assisté de près ou de loin à sa réalisation.

Nous remercions ardemment notre encadrant à l'Ecole Mohammadia d'Ingénieurs Monsieur le professeur N. EL FADDOULI pour la qualité inédite de son suivi et la pertinence de ses conseils

Nous tenons à remercier chaleureusement Messieurs R. AGOUDALE et M. BENYAHYA, chefs de projet à Original Soft System pour nous avoir impliqués dans ce projet, pour le savoir-faire qu'ils nous ont transféré.

Nous adressons nos vifs remerciements à :

Monsieur le professeur S. BENNANI, chef du département informatique, pour nous avoir honoré en présidant le jury.

Madame la professeure A. RETBI, professeure à l'Ecole Mohammadia d'Ingénieurs pour avoir accepté de juger et évaluer notre projet.

Un grand merci également à l'ensemble du corps professoral et administratif à l'Ecole Mohammadia d'Ingénieurs pour son support et sa disponibilité.

Résumé

Les logiciels de gestion intégrée et d'aide à la décision représentent un devis important pour l'amélioration des performances des entreprises. Cependant les coûts d'acquisition et de mise en place des solutions commercialisées par les acteurs majeurs du marché demeurent très élevés.

Conscients de ces faits, les responsables d'Original Soft System ont lancé le projet *Optimis*. Construit en J2EE/Swing et adapté aux PME marocaines, il s'agit d'un ERP qui assure la majorité des fonctionnalités exigées par une entreprise : gestion du personnel, de la documentation, des achats, des ventes, du stock...

Original Soft System désire compléter sa solution par une plateforme décisionnelle. Cette dernière doit fournir aux décideurs des entreprises utilisatrices une meilleure prise de décision. Parmi les traitements que la plateforme doit assurer : le chargement et transformation des données depuis différents bases de données, gérer les processus de transfert et garantir un reporting évolué.

Dans notre projet de fin d'études, nous avons mis en application les éléments déjà présents de la plateforme dans la construction d'un DataMart pour l'activité des ventes, en passant par la collecte des indicateurs, le choix du modèle et la phase ETL sous Talend Open Studio. Nous avons également conçu et généré des rapports d'état à l'aide d'iReport puis nous avons procédé au développement dans la plateforme *Optimis Report* des fonctionnalités d'accès et diffusion des rapports.

ملخص

تشكل برامج التسيير المتكاملة و نظم دعم القرارات دعامة أساسية للرفع من مردودية الشركات. غير أن تكاليف الحصول على البرامج المقترحة من طرف كبريات شركات المعلومات لا تزال جد مرتفعة.

في هذا السياق، أطلقت الشركة المغربية Original Soft System المشروع Optimis هذا الأخير هو عبارة عن برنامج لتدبير موارد الشركات طور بفضل تقنيات J2EE Swing مصمم خصيصا لصالح الشركات المغربية الصغيرة و المتوسطة مشتملا وظائف إدارة شؤون الموظفين، المحاسبة، المشتريات، المبيعات إلخ.

بغية تقديم منتج متكامل، تريد Original Soft System إغناء Optimis عبر تطوير منظومة لدعم إتخاذ القرارات. الوظائف المنوطة بهذه المنظومة تتراوح بين استخلاص، تحويل و شحن البيانات في مخازن بيانات و تصميم تقارير و إرسالها.

خلال مشروع تخرجنا، عمنا باستثمار المكونات الموجودة على مستوى المنظومة لبناء خزان معطيات خاص لتتبع أنشطة البيع و الأداء و ذلك عبر تحصيل المؤشرات اللازمة، اختيار النموذج الملائم لتمثيل البيانات ثم استخلاص، تحويل و شحن هذه البيانات باستعمال Talend Open Studio إضافة إلى ذلك قمنا بتصميم و خلق تقارير حول حجم المعاملات عبر برنامج iReport في الجزء الأخير من المشروع، قمنا بتطوير وظائف الوصول إلى التقارير و تعميمها داخل المنظومة.

Abstract

Enterprise Resource Planning and decision support softwares represent a significant tool for improving business performances. However, the costs of acquisition and implementation of solutions promoted by the market leaders remain very high.

Recognizing these facts, Original Soft System's executives launched the project "Optimis". The latter is an ERP built in J2EE/Swing customized to provide the majority of functionalities required by small-size Moroccan companies: Stock, sales, purchases, staff...

In order to market a complete solution, Original Soft Systems desires to associate a decisional platform to its ERP. The platform *Optimis Report* must perform many treatments in order to support decision makers from extraction, transformation and loading of data in new structures (datamarts and datawarehouses) to reporting.

In our graduation project, we have implemented the existing elements of the platform in the construction of a DataMart for sales activity, through the collection of indicators, the choice of model and the ETL phase via Talend Open Studio. We also designed and generated reports using iReport and then we proceeded to develop inside platform report access and distribution features.

Liste des acronymes

Acronyme	Signification
2TUP	Two-Track Unified Process
BI	Business Intelligence
CA	Chiffre d'affaires
CRM	Customer Relationship Management
DM	DataMart
DWH	DataWarehouse
ERP	Entreprise Ressources Planning
ETL	Extract-Transform-Load
OSS	OriginalSoft System
SI	Système d'information
SQL	Structured Query Language
TOS	Talend Open Studio
UML	United Modeling Language
XML	Extensible Markup Language

Liste des figures

N° figure	Nom figure	Page
Figure 1.1	Architecture fonctionnelle de la solution	17
Figure 1.2	Cycle de développement SCRUM	19
Figure 2.1	Exemple de schéma en étoile	22
Figure 2.2	Exemple de schéma en flocon	23
Figure 3.1	Schéma conceptuel de FAIT_VENTE	27
Figure 3.2	Schéma conceptuel de FAIT_PAIEMENTS	28
Figure 3.3	Schéma conceptuel global du DataMart Ventes	28
Figure 3.4	Paramétrage de la connexion à la BD source	30
Figure 3.5	Connexion avec BD source et sélection des tables des données à extraire	31
Figure 3.6	Connexion au DM et sélection des tables à alimenter	31
Figure 3.7	Job DIM_CLIENT	32
Figure 3.8	Job DIM_ARTICLE	32
Figure 3.9	Job DIM_FACTURE	32
Figure 3.10	Job FAIT_VENTE	33
Figure 3.11	Job FAIT_PAIEMENT	33
Figure 3.12	Scheduler de Talend Open Studio	34
Figure 4.1	Etapas de création de rapport	35
Figure 4.2	Fonctionnement de JasperReports	36
Figure 4.3	Interface d'accueil de iReport	37
Figure 4.4	Liste des drivers de connexion aux sources de données	37
Figure 4.5	Paramétrage de la connexion au DataMart	38
Figure 4.6	Liste des Templates proposées par iReport	39
Figure 4.7	Saisi de la requête SQL	39
Figure 4.8	Rapport des CA par client	41
Figure 4.9	Rapport des CA par produit pendant une durée donnée	42
Figure 4.10	Rapport du total des paiements reçus par rapport au total des factures émises pour un client.	43
Figure 4.11	Transfert des fichiers dans WinSCP	44
Figure 4.12	Chargement des rapports dans Optimis	44
Figure 4.13	Saisi de valeur de paramètre de rapport dans Optimis	45
Figure 5.1	Cycle de développement de la méthode 2TUP	47
Figure 5.2	Interface d'Optimis avec l'onglet « reporting » ouvert.	48
Figure 5.3	Interface de chargement des rapports dans Optimis	48
Figure 5.4	Interface de liste des catégories des rapports	49
Figure 5.5	Interface de liste des rapports affectés à une catégorie	49
Figure 5.6	Diagramme des cas d'utilisation du module « reporting »	51
Figure 5.7	Diagramme des classes	57
Figure 5.8	Diagramme de séquence d'autorisation de rapport à un utilisateur et exécution	58
Figure 5.9	Diagramme de séquence d'autorisation de rapport à une liste, définition d'horaires de diffusion et paramètres	59
Figure 5.10	Interface de création de listes de diffusion	60

Figure 5.11	Interface principale d'ajout de destinataires aux listes	60
Figure 5.12	Interface de choix de liste de diffusion pour ajout	61
Figure 5.13	Interface d'insertion de destinataires	62
Figure 5.14	Interface d'affectation d'un destinataire à une liste	62
Figure 5.15	Interface de saisi d'horaire et paramètre d'envoi d'un rapport.	62

Liste des tableaux

N° tableau	Nom tableau	Page
Tableau 1.1	Planning du projet	29
Tableau 5.1	Fiche de description de cas d'utilisation « Créer liste de diffusion »	52
Tableau 5.2	Fiche de description de cas « Ajouter destinataire à une liste »	53
Tableau 5.3	Fiche de description de cas « Définir horaires d'envoi à une liste »	54
Tableau 5.4	Fiche de description de cas « Autoriser rapport à une utilisateur »	55
Tableau 5.5	Fiche de description de cas « Consulter historique de paramétrage »	56

Table des matières

INTRODUCTION:.....	17
CHAPITRE I : CONTEXTE GENERALE DU PROJET :.....	18
1. Présentation de l'organisme d'accueil :.....	18
2. Présentation du projet :.....	18
2.1. Problématique :.....	18
2.2. Objectifs :.....	19
2.3. Contraintes :.....	20
2.4. Démarche :.....	20
2.5. Planning :.....	21
3. Conclusion :.....	22
1. Architecture décisionnelle :.....	23
1.1. Business Intelligence :.....	23
1.2. DataMart :.....	23
1.3. Modélisation DataMart/DataWarehouse :.....	24
2. Extraction des données :.....	25
3. Reporting :.....	26
4. Conclusion :.....	26
CHAPITRE III : CONCEPTION ET REALISATION DE DATAMART DE VENTES :.....	27
1. Choix des indicateurs :.....	27
2. Modélisation :.....	27
2.1. Démarche suivie pour la modélisation :.....	27
2.2. Dimensions recueillies :.....	28
2.3. Choix du modèle :.....	28
3. Schéma conceptuel des données:.....	28
4.1. FAIT_VENTE :.....	28
4.2. FAIT_PAIEMENT :.....	29
4. Implémentation ETL :.....	30
4.1. Composants Talend utilisés :.....	31
4.2. Connexion à la base de données :.....	31
4.3. Chargement des tables de dimensions :.....	33
4.4. Chargement des tables de faits :.....	35

5. Mise à jour des données du DataMart :.....	35
6. Conclusion :.....	36
CHAPITRE IV : PROCESSUS DE REPORTING :.....	37
1. Généralités :.....	37
2. Les rapports :.....	38
2.1. Chiffre d’affaire par client :	42
2.2. Chiffre d’affaire par produit pendant une durée donnée :.....	43
2.3. Total des paiements reçus par rapport au total des factures émises pour un client donné :.....	44
3. Exécution des rapports dans Optimis Report Viewer :	45
Conclusion :.....	47
CHAPITRE V : CONCPETION ET REALISATION D’OPTIMIS REPORT VIEWER ET SHARE :.....	48
1. Méthodologie :	48
2. Analyse fonctionnelle :.....	49
2.1. Analyse et critique de l’existant :.....	49
2.2. Etude de la problématique :	52
2.3. Description des fonctionnalités principales :	52
3. Analyse technique :	52
4. Conception :	53
4.1. Identification des profils utilisateurs :.....	53
4.2. Modélisation UML :	53
5. Réalisation :.....	61
5.1. Création de listes de diffusion :	61
5.2. Ajout de destinataires à une liste :	62
5.3. Saisi de détails de diffusion :	63
CONCLUSION ET PERSPECTIVES :.....	65
WEBOGRAPHIE:.....	66
ANNEXES :	67

Table des matières

INTRODUCTION:.....	17
CHAPITRE I : CONTEXTE GENERALE DU PROJET :.....	18
1. Présentation de l'organisme d'accueil :.....	18
2. Présentation du projet :.....	18
2.1. Problématique :.....	18
2.2. Objectifs :.....	19
2.3. Contraintes :.....	20
2.4. Démarche :.....	20
2.5. Planning :.....	21
3. Conclusion :.....	22
1. Architecture décisionnelle :.....	23
1.1. Business Intelligence :.....	23
1.2. DataMart :.....	23
1.3. Modélisation DataMart/DataWarehouse :.....	24
2. Extraction des données :.....	25
3. Reporting :.....	26
4. Conclusion :.....	26
CHAPITRE III : CONCEPTION ET REALISATION DE DATAMART DE VENTES :.....	27
1. Choix des indicateurs :.....	27
2. Modélisation :.....	27
2.1. Démarche suivie pour la modélisation :.....	27
2.2. Dimensions recueillies :.....	28
2.3. Choix du modèle :.....	28
3. Schéma conceptuel des données:.....	28
4.1. FAIT_VENTE :.....	28
4.2. FAIT_PAIEMENT :.....	29
4. Implémentation ETL :.....	30
4.1. Composants Talend utilisés :.....	31
4.2. Connexion à la base de données :.....	31
4.3. Chargement des tables de dimensions :.....	33
4.4. Chargement des tables de faits :.....	35

5. Mise à jour des données du DataMart :.....	35
6. Conclusion :.....	36
CHAPITRE IV : PROCESSUS DE REPORTING :.....	37
1. Généralités :.....	37
2. Les rapports :.....	38
2.1. Chiffre d'affaire par client :	42
2.2. Chiffre d'affaire par produit pendant une durée donnée :.....	43
2.3. Total des paiements reçus par rapport au total des factures émises pour un client donné :.....	44
3. Exécution des rapports dans Optimis Report Viewer :	45
Conclusion :.....	47
CHAPITRE V : CONCPETION ET REALISATION D'OPTIMIS REPORT VIEWER ET SHARE :.....	48
1. Méthodologie :	48
2. Analyse fonctionnelle :.....	49
2.1. Analyse et critique de l'existant :.....	49
2.2. Etude de la problématique :	52
2.3. Description des fonctionnalités principales :	52
3. Analyse technique :	52
4. Conception :	53
4.1. Identification des profils utilisateurs :.....	53
4.2. Modélisation UML :	53
5. Réalisation :.....	61
5.1. Création de listes de diffusion :	61
5.2. Ajout de destinataires à une liste :	62
5.3. Saisi de détails de diffusion :	63
CONCLUSION ET PERSPECTIVES :.....	65
WEBOGRAPHIE:.....	66
ANNEXES :	67

INTRODUCTION:

Le Maroc occupe une place pondérée sur la scène économique régionale que ce soit en tant que pays émergent à bon rythme de croissance ou en tant que nouveau hub attractif d'investissements. Comme le développement du tissu économique du pays s'appuie essentiellement sur les petites et moyennes entreprises, ces dernières se trouvent dans l'obligation d'améliorer leur rendement et de s'adapter à de hauts niveaux de performance que le décollage économique et la compétition imposent.

Le recours à l'informatisation a montré sa primordialité dans les stratégies d'encouragement et de mise à niveau de l'entrepreneuriat dans plusieurs pays. Pour cela, l'usage des systèmes d'information de gestion et d'aide à la prise de décision est de plus en plus exigé.

Comme les acteurs majeurs du marché des progiciels de gestion intégrés et d'outils d'informatique décisionnelle (SAP, SAGE, Microsoft, ...) imposent des coûts considérables d'acquisition et de mise en place de leurs solutions ainsi que ceux de la formation excédant les capacités des PME marocaines. Original Soft System, start-up marocaine avec solide expérience de ses collaborateurs, a réfléchi à présenter un produit rassemblant à la fois un ERP¹ et une suite BI adaptés à l'entreprise marocaine.

Le produit *Optimis* qu'Original Soft System développe se base principalement sur des outils Open Source qui ont prouvé leur efficacité comme : Talend Open Studio pour l'extraction, transformation & chargement des données, iReport pour le reporting ainsi que le Framework de développement J2EE Open Swing.

L'objectif de notre projet de fin d'études consiste à soumettre la chaîne d'outils décisionnels mis en place à des traitements séquentiels à commencer par la réalisation de DataMart de l'activité des ventes. Les résultats émanant de cette nouvelle structure de données sont exploités avec une plus large envergure dans les outils de reporting. Il est également question d'enrichir le produit en ajoutant des fonctionnalités d'accès aux rapports d'état desquels une entreprise peut avoir besoin.

Ce rapport se compose de cinq chapitres, le premier présente l'aspect organisationnel du projet. Quant au deuxième, il englobe les notions et les techniques impliquées. Le troisième chapitre expose la conception et la réalisation du DataMart des ventes. Nous avons consacré le quatrième chapitre à notre réalisation de rapports via iReport. Le dernier chapitre présente notre contribution à l'enrichissement de la plateforme *Optimis Report*.

¹ Annexe A : ERP

CHAPITRE I : CONTEXTE GENERALE DU PROJET :

1. Présentation de l'organisme d'accueil :

OriginalSoft System (OSS) est une société marocaine de services spécialisée dans les nouvelles technologies de l'information et de la communication. Créé en 2010, son siège social est basé à Rabat.

OSS est une entreprise à taille humaine qui a su développer à travers l'expertise de ses collaborateurs un véritable métier autour de sa fonction de société de services. La double expertise, technique et fonctionnelle, a permis à OSS de proposer des services allant du conseil sur le terrain (audit et conseil informatique, en production, maintenance, qualité, achat, stocks...), vers l'information des résultats et adaptation des logiciels.

A travers son expérience, OSS a conçu et réalisé le logiciel *Optimis*. Ce dernier est un logiciel intégré destiné à optimiser la gestion d'entreprises.

« Votre satisfaction, notre soucis, notre fierté » ; telle est la devise adoptée par les décideurs de l'entreprise. Ainsi, OSS s'est fixée comme objectif de répondre aux préoccupations de ses clients qui sont constitués des PME-PMI, leur apporter une expertise technique et fonctionnelle de pointe, leur proposer des solutions avec un meilleur rapport qualité/prix. La stratégie d'OriginalSoft System est d'établir avec ses clients une collaboration à long terme.

2. Présentation du projet :

2.1.Problématique :

L'offre d'OriginalSoft System est centrée autour du logiciel *Optimis* qui a été adapté, customisé et enrichi afin de répondre aux besoins des PME-PMI marocaines. C'est une solution basée sur *jAllInOne*² ; un ERP libre développé en communauté en s'appuyant sur la bibliothèque *Swing* de Java et suivant le paradigme *MVC*³.

Pour suivre les évolutions de l'informatique et afin de fournir aux entreprises utilisatrices d'*Optimis* les moyens qui leur permettent de collecter, consolider, modéliser et restituer leurs données dans le but d'offrir une aide à la décision et permettre aux responsables de stratégie d'avoir une vue d'ensemble sur les activités traitées, OSS a jugé important d'ajouter une couche décisionnelle à sa solution.

Afin de fournir un produit au coût réduit et vu aussi les hautes performances de certains outils libres de *Business Intelligence*, *Optimis Report* se combine entre l'Open Source et le propriétaire. La chaîne BI proposée par OSS utilise *Talend Open Studio*⁴ pour la conception des processus ETL, *JasperReports* pour la conception des rapports couplé à l'éditeur graphique *iReport*⁵ ainsi qu'un module de reporting dans l'ERP *Optimis* dont

² Annexe B : *jAllInOne*

³ Annexe C : *MVC*

⁴ Annexe D : *Talend Open Studio*

⁵ Annexe E : *iReport*

plusieurs fonctionnalités ne sont pas encore développées.

Dans notre projet de fin d'études nous sommes censés compléter et enrichir cette chaîne BI en développant les éléments manquants :

Optimis Report Share et *Optimis Report Viewer*.

2.2.Objectifs :

La finalité attendue de notre projet est d'avoir une plateforme BI générique qui peut supporter l'ensemble des processus décisionnels depuis l'extraction des données de la base de données de production jusqu'à la diffusion des rapports d'activité dans les boîtes mail des responsables correspondants.

Pour se faire, nous devons suivre d'abord l'enchaînement classique d'un projet BI en choisissant une activité donnée assurée par l'ERP Optimis (dans notre cas ventes & paiements), relever des indicateurs et construire le DataMart correspondant.

Vu que le reporting constitue un élément important dans le projet en ce concerne la mise en place d'outils de visualisation et diffusion de rapports, il est d'abord nécessaire d'enchaîner sur la démarche BI jusqu'à conception et génération de rapports via iReport. Ces rapports constitueront l'entrée principale d'*Optimis Report Viewer*, outil de visualisation des rapports et d'*Optimis Report Share*, outil de diffusion des rapports.

Nos objectifs se résument alors à concevoir et mettre en place un DataMart pour l'activité des ventes, concevoir et générer des rapports d'état, concevoir et développer les fonctionnalités d'accès et diffusion des rapports nécessaires au module de reporting.

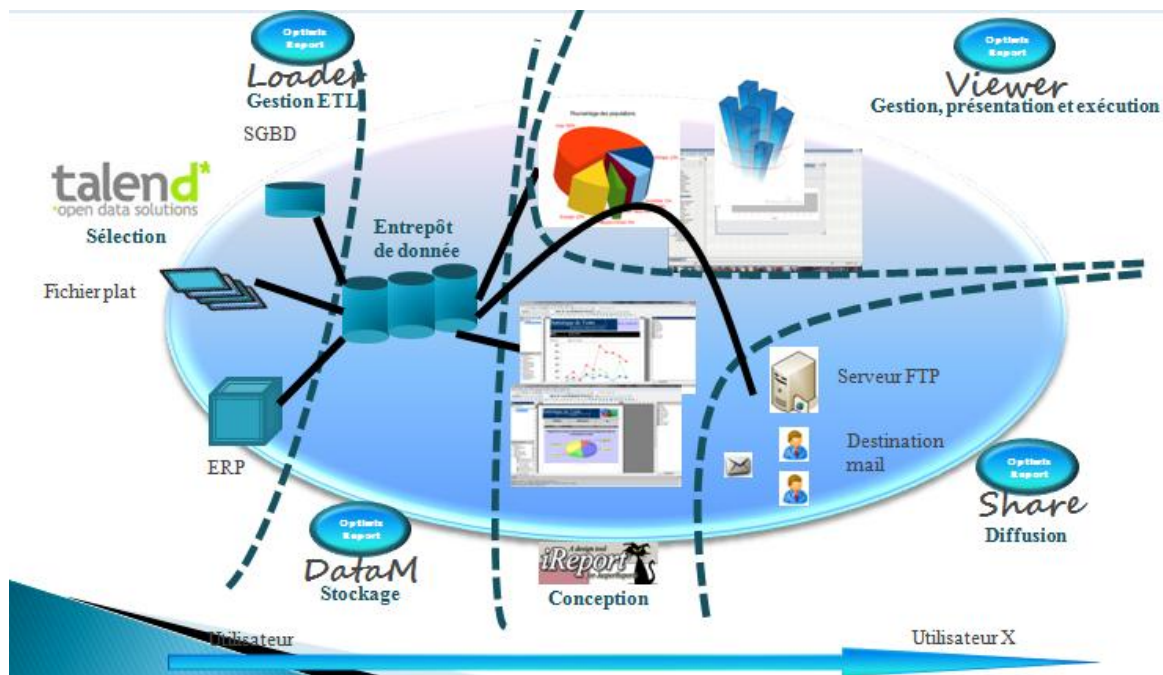


Figure 1.1 : Architecture fonctionnelle

2.3.Contraintes :

Nous dénombrons les différentes contraintes du projet :

- Taille importante :
Vu dans sa globalité, le projet représente un volume intéressant et par la suite une bonne conduite du projet est fortement exigée.
- Plusieurs composantes entrelacées :
Le projet impose une transition entre une partie menée proprement en démarche BI et dont les résultats doivent aider à conduire une partie de développement d'outils avec les méthodes qui correspondent. La transition doit être également bien gérée.
- Choix du cheminement à suivre :
Par conséquent des deux contraintes citées ci-dessus, il faut adopter une démarche tout au long du projet simple à implémenter et qui permet de faire des itérations sans causer des blocages au niveau organisationnel.
- Complexité de la base de données :
Il s'agit d'une base de données d'ERP construite pour soutenir un accès performant, les tables que nous devons créer et rajouter aux 166 tables existantes doivent alors respecter les contraintes d'intégrité de la base et conserver la persistance des données.

2.4.Démarche :

Après multiples réunions avec les responsables d'OSS, dans le but de clarifier leurs exigences en matière de réalisation de la plateforme BI requise, et suite aux recommandations de notre encadrant, nous avons opté pour l'une des méthodes Agiles pour la gestion de projet.

La méthode choisie pour mener le projet étant la méthode SCRUM, nous définissons d'abord la terminologie SCRUM que nous utiliserons :

Product Owner : représente le client et est responsable de la priorisation du backlog

Sprint Backlog : une liste des items de la plus haute priorité du backlog de produit.

Sprint : période qui dure, en pratique, entre 2 et 4 semaines focalisée sur l'effort de l'équipe pour atteindre les buts fixés.

En projetant la méthodologie définie ci-dessus sur notre projet, les Product Owners (PO) sont les responsables d'OSS représentés par MM. Agoudal et Benyahia.

Le projet peut être découpé selon les sprints suivants :

Sprint 1 : Etude préliminaire

Sprint 2 : Conception & création d'un datamart de ventes

Sprint 3 : Reporting

Sprint 4 : Développement d'Optimis Report Share et ajout de fonctionnalités au Viewer.

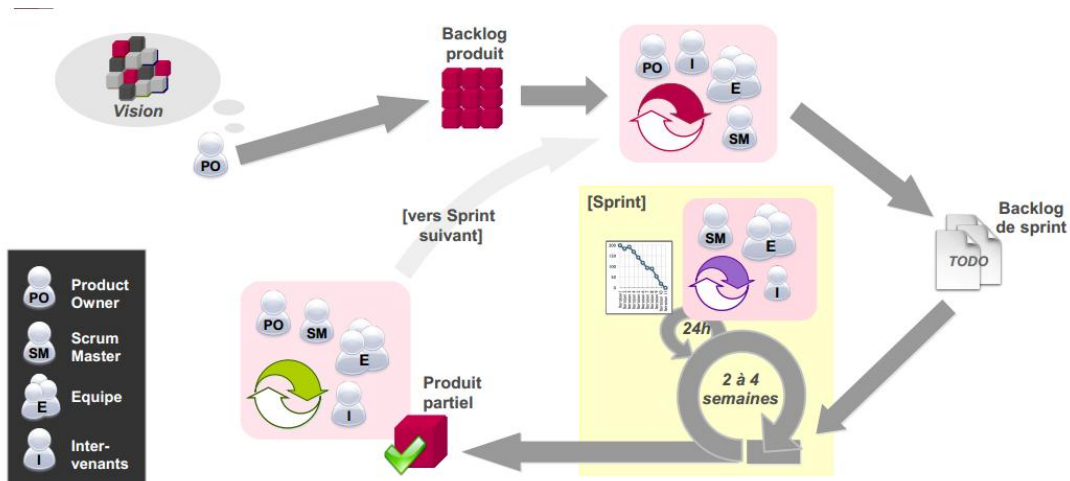


Figure 1.2 : Cycle de vie suivant SCRUM proposé par TIMWI Consulting

2.5.Planning :

N° du sprint	Description	Backlog du sprint	Livrables	Périodes
0	Préliminaire	<ul style="list-style-type: none"> Planification Etude de l'existant Etude des besoins 	<ul style="list-style-type: none"> Dossier d'étude préliminaire 	<ul style="list-style-type: none"> Du 06/02/2012 Au 17/02/2012
1	Conception & création du DataMart	<ul style="list-style-type: none"> Conception du DM <ul style="list-style-type: none"> ✓ Choix des indicateurs ✓ Modélisation ✓ Schéma conceptuel des données ETL <ul style="list-style-type: none"> ✓ Définition des sources de données ✓ Implémentation ETL 	<ul style="list-style-type: none"> Liste des indicateurs recueillis Schéma de conception du DM Dossier des jobs réalisés sous TOS 	<ul style="list-style-type: none"> Du 13/02/2012 Au 29/02/2012
2	Reporting	<ul style="list-style-type: none"> Conception & génération des rapports sous iReport Consultation sur Optimis Report Viewer 	<ul style="list-style-type: none"> Fichiers .jrxml et .xml des rapports Rapports format PDF Rapport de problèmes d'exécution 	<ul style="list-style-type: none"> Du 01/03/2012 Au 16/03/2012

3	Réalisation d'O.R. Share et ajout de fonctionnalités à O.R. Viewer	<ul style="list-style-type: none"> • Analyse de l'existant au niveau de Viewer • Conception des fonctionnalités à ajouter à Viewer • Conception des fonctionnalités à développer dans Share • Réalisation 	<ul style="list-style-type: none"> • Dossier de conception de la solution • Nouveau schéma physique des données • Maquettes des interfaces • Code source 	<ul style="list-style-type: none"> • Du 19/02/2012 • Au 25/05/2012
---	--	---	--	--

Tableau 1.1 : Planning du projet

Le tableau ci-dessus décrit le planning que nous avons réalisé et suivi afin de maîtriser notre travail. Il est illustré par le diagramme de Gant ⁶.

3. Conclusion :

Dans ce chapitre, nous avons présenté l'aspect organisationnel de notre projet ainsi qu'un descriptif de l'organisme d'accueil. Nous passons par la suite à une présentation des notions et des techniques employées au niveau du projet.

⁶ Annexe F : Diagramme de GANT

CHAPITRE II : NOTIONS ET TECHNIQUES :

Dans ce chapitre, nous allons présenter les notions d'informatique décisionnelle que nous utilisons ; à savoir l'architecture décisionnelle, le processus ETL et le reporting. Les outils engagés au cours du projet sont spécifiés.

1. Architecture décisionnelle :

1.1. Business Intelligence :

L'informatique décisionnelle (en anglais « Business intelligence) est une discipline de l'informatique qui cible l'exploitation des données de l'entreprise afin de faciliter la prise de décision par les responsables, c'est-à-dire la compréhension du fonctionnement actuel et l'anticipation des actions pour un pilotage éclairé de l'entreprise.

Les outils décisionnels sont basés sur l'exploitation d'un système d'information décisionnel alimenté grâce à l'extraction de données diverses à partir des données de production, d'informations concernant l'entreprise ou son entourage et de données économiques.

1.2. DataMart :

Le DataMart est un ensemble de données ciblées, organisées, regroupées et agrégées pour répondre à un besoin spécifique à un métier ou à un domaine donné. Il est donc destiné à être interrogé sur un panel de données restreint à son domaine fonctionnel, selon des paramètres qui auront été définis à l'avance lors de sa conception.

De façon plus technique, le DataMart peut être considéré de deux manières différentes, attribuées aux deux principaux théoriciens de l'informatique décisionnelle, Bill Inmon et Ralph Kimball :

- Définition d'Inmon : Le DataMart est issu d'un flux de données provenant du DataWarehouse. Contrairement à ce dernier qui présente le détail des données pour toute l'entreprise, il a vocation à présenter la donnée de manière spécialisée, agrégée et regroupée fonctionnellement.
- Définition de Kimball : Le DataMart est un sous-ensemble du DataWarehouse, constitué de tables au niveau détail et à des niveaux plus agrégés, permettant de restituer tout le spectre d'une activité métier. L'ensemble des DataMarts de l'entreprise constitue le DataWarehouse.

Considérant le contexte auquel elle se rapporte, on peut privilégier suivant les entreprises l'une ou l'autre de ces définitions.

Pour la modélisation des données et l'administration de la base de données, nous utiliserons MySQL Workbench ⁷.

⁷ Annexe J : MySQL Workbench 5.2

1.3.Modélisation DataMart/DataWarehouse :

La conception et la modélisation de DataMart (et plus généralement de DataWarehouse) se ramènent à définir deux concepts principaux : faits et dimensions. C'est la mise en place de faits et de dimensions qui permet de définir le schéma de modélisation que le DM/DWH doit suivre :

1.3.1. Les faits :

Une table de faits est une table qui contient les données observables (les faits) que l'on possède sur un sujet et que l'on veut étudier, selon divers axes d'analyse (les dimensions). Les « faits », dans un entrepôt de données, sont normalement numériques, puisque d'ordre quantitatif. Il peut s'agir du montant en argent des ventes, du nombre d'unités vendues d'un produit, etc.

1.3.2. Les dimensions :

Une dimension est une table qui contient les axes d'analyse (les dimensions) selon lesquels on veut étudier des données observables (les faits) qui, soumises à une analyse multidimensionnelle, donnent aux utilisateurs des renseignements nécessaires à la prise de décision.

On appelle donc « dimension » un axe d'analyse. Il peut s'agir des clients ou des produits d'une entreprise, d'une période de temps comme un exercice financier, des activités menées au sein d'une société, etc.

1.3.3. Schéma relationnel DM/DWH :

- **Schéma en étoile :**

Dans ce schéma, on associe à chaque dimension une clé primaire non composite qui va figurer aussi dans les tables de faits auxquelles est reliée la dimension.

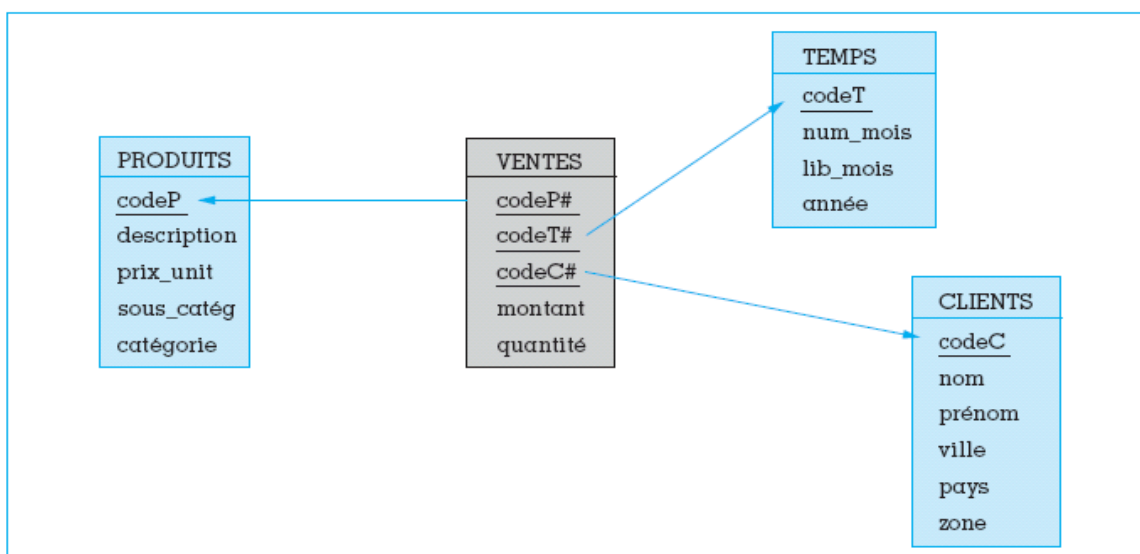


Figure 2.1 : Exemple de schéma en étoile

- **Schéma en flocon :**

Dérivé du schéma en étoile, ce schéma permet de factoriser les dimensions afin de réduire les redondances. Il permet également la gestion des relations multi-niveaux. Nous reprenons le même exemple des ventes cité dans le paragraphe ci-dessus.

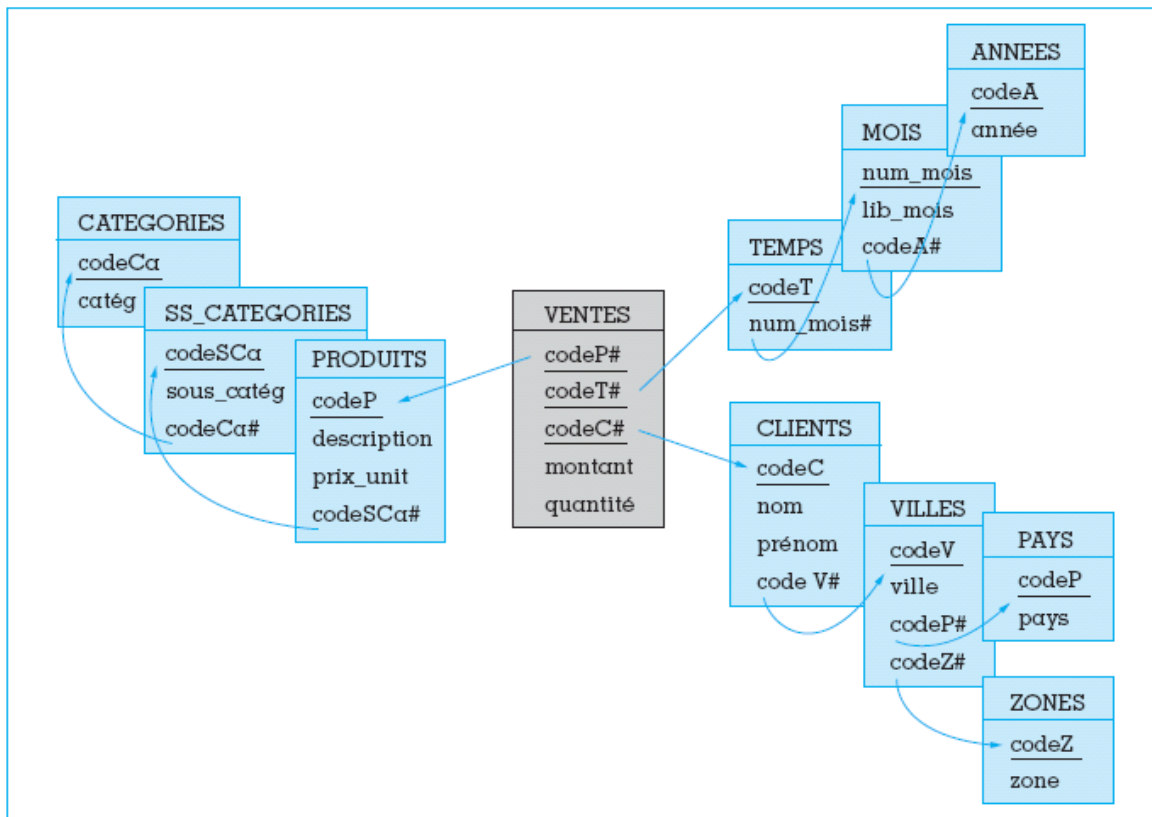


Figure 2.2: Exemple de schéma en flocon

2. Extraction des données :

Les processus ETL (*Extract-Transform-Load*) permettent d'extraire, transformer et charger les données depuis des sources diverses et hétérogènes (Bases de données, fichiers plats...) jusqu'au DWH ou DM. Préalablement, l'ETL doit effectuer dé-normalisation, nettoyage et remplacement dans le contexte de ces données puis les charger. Les données arrivant à la structure cible (DWH ou DM) sont alors homogènes.

Il s'agit de l'étape la plus importante dans un projet décisionnel car c'est à son niveau qu'on détecte les anomalies et on effectue les calculs complexes.

Plusieurs outils d'ETL sont présents sur le marché, variant entre le libre et le payant. Voici une liste des outils les plus demandés :

- Pentaho Data Integration (PDI)
- SQL Server Integration Services (SSIS)
- Talend Open Studio (TOS)

Dans notre projet, nous avons eu recours à Talend Open Studio pour

3. Reporting :

Le reporting occupe le dernier maillon dans une chaîne de projet décisionnel.

Il s'agit de techniques informatiques permettant l'extraction des données depuis un DM/DWH et leur présentation dans des rapports humainement lisibles et personnalisés selon le goût du lecteur.

Etant probablement l'application la plus utilisée encore aujourd'hui de l'informatique décisionnelle, le reporting permet aux gestionnaires :

- de sélectionner des données relatives à telle période, telle production, tel secteur de clientèle, etc.,
- de trier, regrouper ou répartir ces données selon les critères de leur choix,
- de réaliser divers calculs (totaux, moyennes, écarts, comparatif d'une période à l'autre, ...),
- de présenter les résultats d'une manière synthétique ou détaillée, le plus souvent graphique selon leurs besoins ou les attentes des dirigeants de l'entreprise

Il existe des outils d'analyse décisionnelle qui permettent de modéliser des représentations à base de requêtes afin de constituer des rapports (ou tableaux de bords).

Parmi ces outils, nous citons les plus connus :

- BIRT
- Pentaho Reporting
- iReport

iReport, dont le moteur de rapport est JasperReport, est le logiciel employé par l'entreprise pour la conception et la réalisation des différents rapports.

4. Conclusion :

Nous avons défini dans ce chapitre les notions et les technologies à employer dans notre projet. Les éléments non mentionnés seront inclus dans l'annexe. Nous exposons au chapitre suivant notre conception et réalisation du DataMart.

CHAPITRE III : CONCEPTION ET REALISATION DE DATAMART DE VENTES :

Dans ce chapitre, nous présentons les étapes d'une partie importante du volet décisionnel de notre projet. Il s'agit de la conception et la réalisation d'un DataMart pour l'activité des ventes. Cette partie est d'une telle importance car elle sert principalement à examiner la chaîne BI en cours de construction et obtenir, par la suite, grâce aux techniques de reporting des rapports d'activités qui vont servir d'entrée dans une partie ultérieure du projet.

1. Choix des indicateurs :

Après avoir déterminé l'activité sur laquelle l'étude décisionnelle va porter, il faut relever des indicateurs clés génériques permettant d'extraire des rapports sur l'état des ventes d'une entreprise.

Les responsables de la société d'accueil, et grâce à leur expérience dans le métier du conseil et dans le secteur des PME, ont rectifié et validé la liste d'indicateurs relatifs aux ventes et paiements que nous avons élaboré.

Les indicateurs choisis pour répondre aux besoins prédéfinis dans le secteur des ventes sont les suivants :

- Le chiffre d'affaire réalisé par l'entreprise pour un client donné.
- Le chiffre d'affaire obtenu pendant une durée donnée de l'activité.
- Le chiffre d'affaire réalisé par l'entreprise pour un article donné.
- Total des paiements reçus pour un client par rapport aux factures émises pour ce dernier.
- Total des paiements reçus pour une période donnée par rapport au total des factures émises.

2. Modélisation :

2.1. Démarche suivie pour la modélisation :

Avant d'entamer l'étape de modélisation des données, nous avons commencé par étudier le fonctionnement de l'ERP et notamment l'architecture de sa base de données.

Par la suite, nous avons procédé à la définition des données utiles ce qui nous a permis de définir les sources de données destinées à l'extraction pour cerner un ensemble de tables et leurs colonnes et inspecter les relations entre elles.

Nous avons examiné les indicateurs précédemment relevés pour en déduire quatre axes d'analyse : temps, client, et facture que nous allons illustrer par des dimensions correspondantes.

2.2. Dimensions recueillies :

- **Dimension date** : sert à localiser la date ou la période pour laquelle il faut effectuer un calcul. Elle est reliée aux tables de faits ventes et paiements.
- **Dimension client** : permet d'extraire des informations relatives à un client avec qui l'entreprise a effectué une opération de vente. Commune aux deux tables de faits.
- **Dimension article** : représente les articles que l'entreprise met en vente. Elle est reliée à la table de fait des ventes.
- **Dimension facture** : décrit les factures que l'entreprise émet aux clients lors d'une opération de vente. Elle est reliée uniquement à la table de faits des paiements.

2.3. Choix du modèle :

Nous avons dû faire le choix parmi les modèles suivants : en étoile, en flocon (et en constellation). Le modèle choisi est le modèle en étoile car il est le plus simple et le plus adéquat au cas de notre DataMart pour les raisons suivantes :

Performance : dans ce modèle, on n'a besoin que de peu de jointures dans une requête, ce qui fournit par la suite une facilité dans la navigation.

Volumétrie : les tables sources que nous avons définies dans l'ERP ne sont pas très volumineuses.

Hiérarchie : Il n'y a aucune relation d'hiérarchie entre les tables de dimensions précédemment présentées.

3. Schéma conceptuel des données:

Le choix des indicateurs et la déduction des dimensions d'analyse nous ont menés à élaborer deux tables de fait. Ces tables de fait permettent d'exprimer les différents indicateurs que nous avons recueillis auparavant.

Ces tables de faits seront notées : FAIT_VENTE et FAIT_PAIEMENT.

4.1. FAIT_VENTE :

Cette table va fournir les données nécessaires aux calculs relatifs aux indicateurs de chiffre d'affaire.

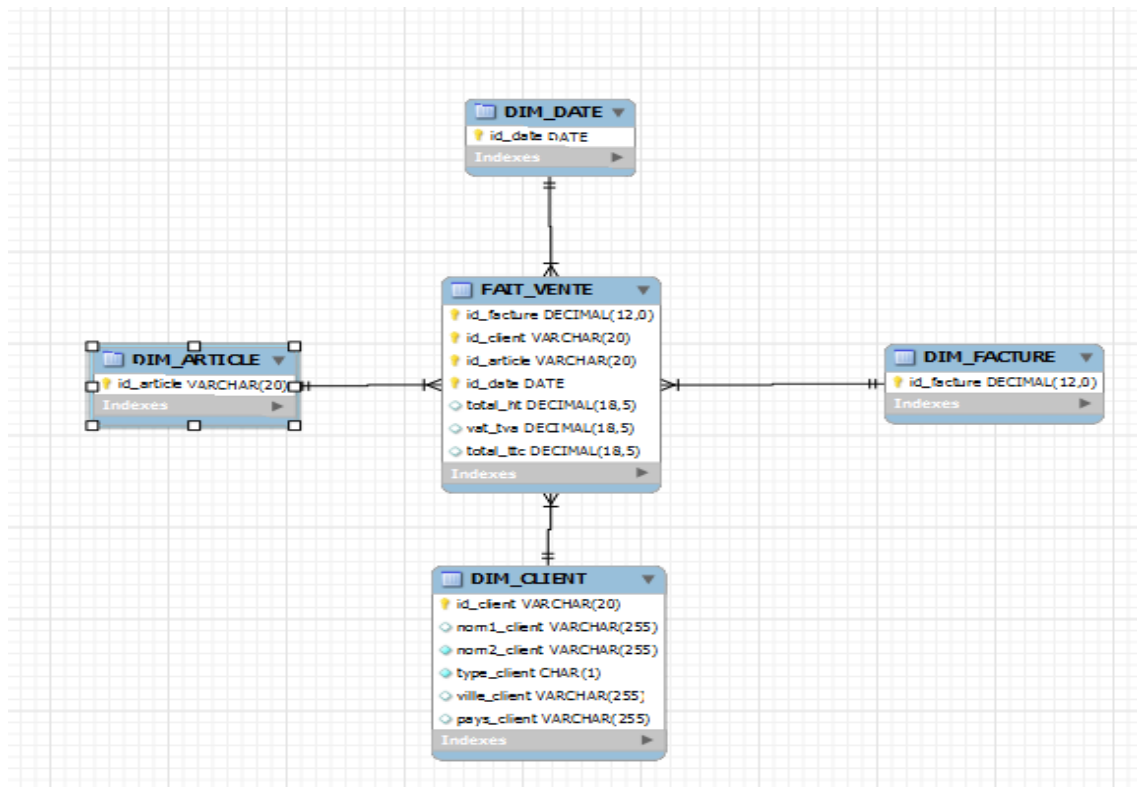


Figure 3.1 : Schéma conceptuel de la table FAIT_VENTE

4.2.FAIT_PAIEMENT :

Dans cette table, il y a les données relatives aux opérations de paiement. Reliée à DIM_DATE, DIM_CLIENT, ses données permettent de calculer les indicateurs relatifs aux soldes des clients.

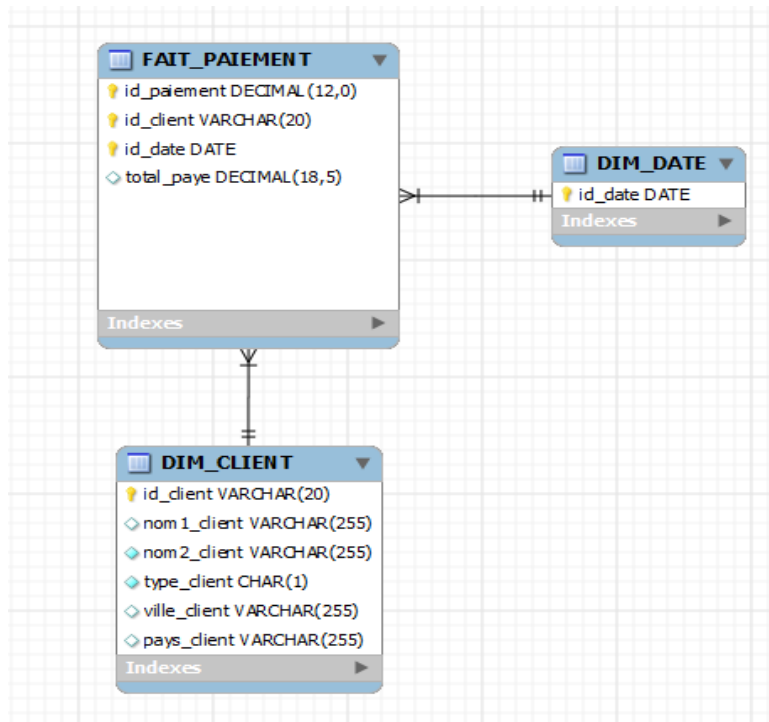


Figure 3.2 : Schéma conceptuel de la table FAIT_PAIEMENT

Ainsi, le schéma global de notre DataMart est comme suit.

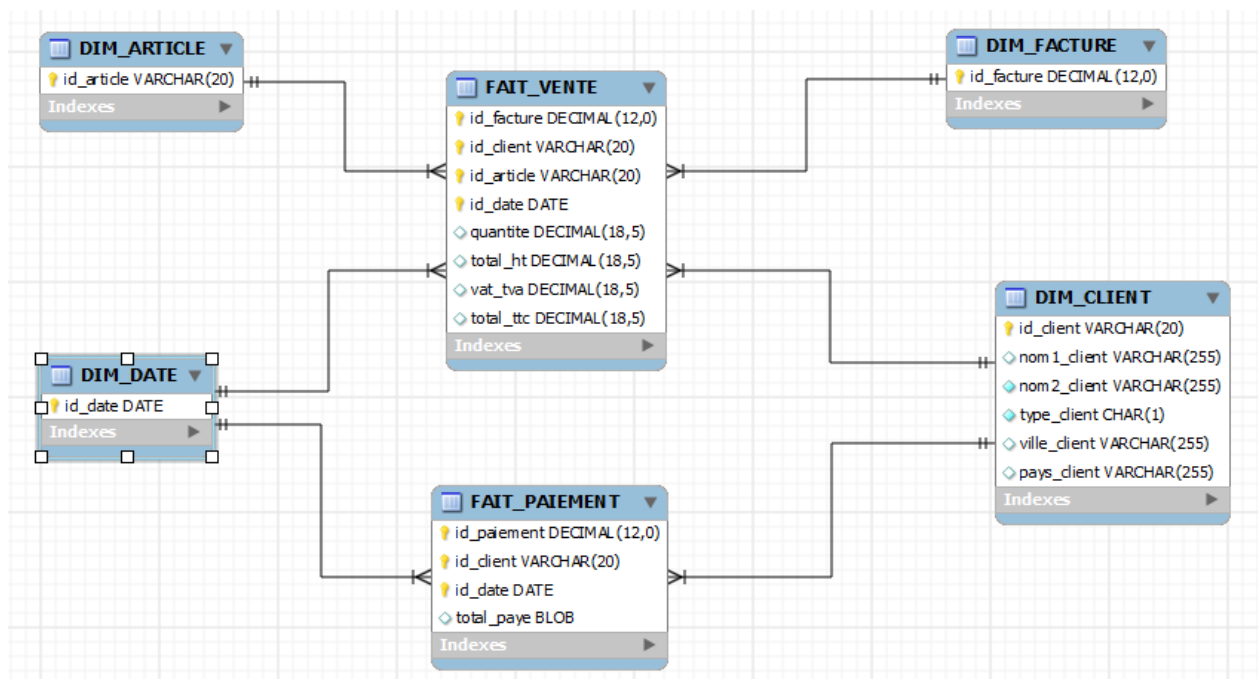


Figure 3.3: Schéma conceptuel global du DataMart des ventes

4. Implémentation ETL :


Dans cet axe, nous allons décrire les étapes de mise en œuvre du système décisionnel conçu pour cette phase du projet, pour cela nous allons présenter les composants utilisés dans

Talend Open Studio puis leur exploitation dans l'alimentation du DataMart puis l'étape de mise à jour de ce dernier une fois chargé en données.


4.1. Composants Talend utilisés :

L'implémentation ETL consiste à mettre en place ce qu'on appelle en nomenclature de Talend des « jobs » permettant d'extraire les données à partir de leurs sources, d'effectuer les transformations nécessaires et de les charger dans le DataMart cible. L'homogénéité des données est importante afin de respecter les contraintes d'intégrité de la structure cible.

Les composants que nous avons utilisés sont les suivants :


tMySQLInput : Ce composant (noté graphiquement ) permet d'indiquer les tables sources de la base de données de production (test2) qui vont subir une extraction des données.

Le tMysqlInput exécute une requête en base de données selon un ordre strict qui doit correspondre à celui défini dans le schéma. La liste des champs récupérée est ensuite transmise au composant suivant via une connexion de flux (Main row).

tMySQLOutput : Ce composant (représenté graphiquement par ) permet de déterminer les tables destination qui vont accueillir les données extraites et transformées depuis la source.

Le tMysqlOutput exécute l'action définie sur la table et/ou sur les données d'une table, en fonction du flux entrant provenant du composant précédent.

Des flèches reliant les tables sources ou cibles, permettent d'exprimer les flux de transformations ou de chargement des données. Le traitement des bases de données MySQL source et destination traitées dans cette partie du projet s'effectue via les deux composants tMySQLInput et tMySQLOutput cités ci-dessus.

tMap : L'élément tMap (dont la notation graphique est ) permet d'effectuer différentes jointures entre les divers composants des sources de données et même des transformations sur des flux. Il permet de choisir les données à extraire en se connectant à la base de données de production, les transformer, les rendre homogènes et les charger dans les tables de destination.

4.2. Connexion à la base de données :

Les tâches effectuées par Talend consistent principalement à extraire des données d'une base de données sources, qui est dans notre cas « Test2 » et les transformer avant de les charger dans la base de données destination « Mydbtest ».

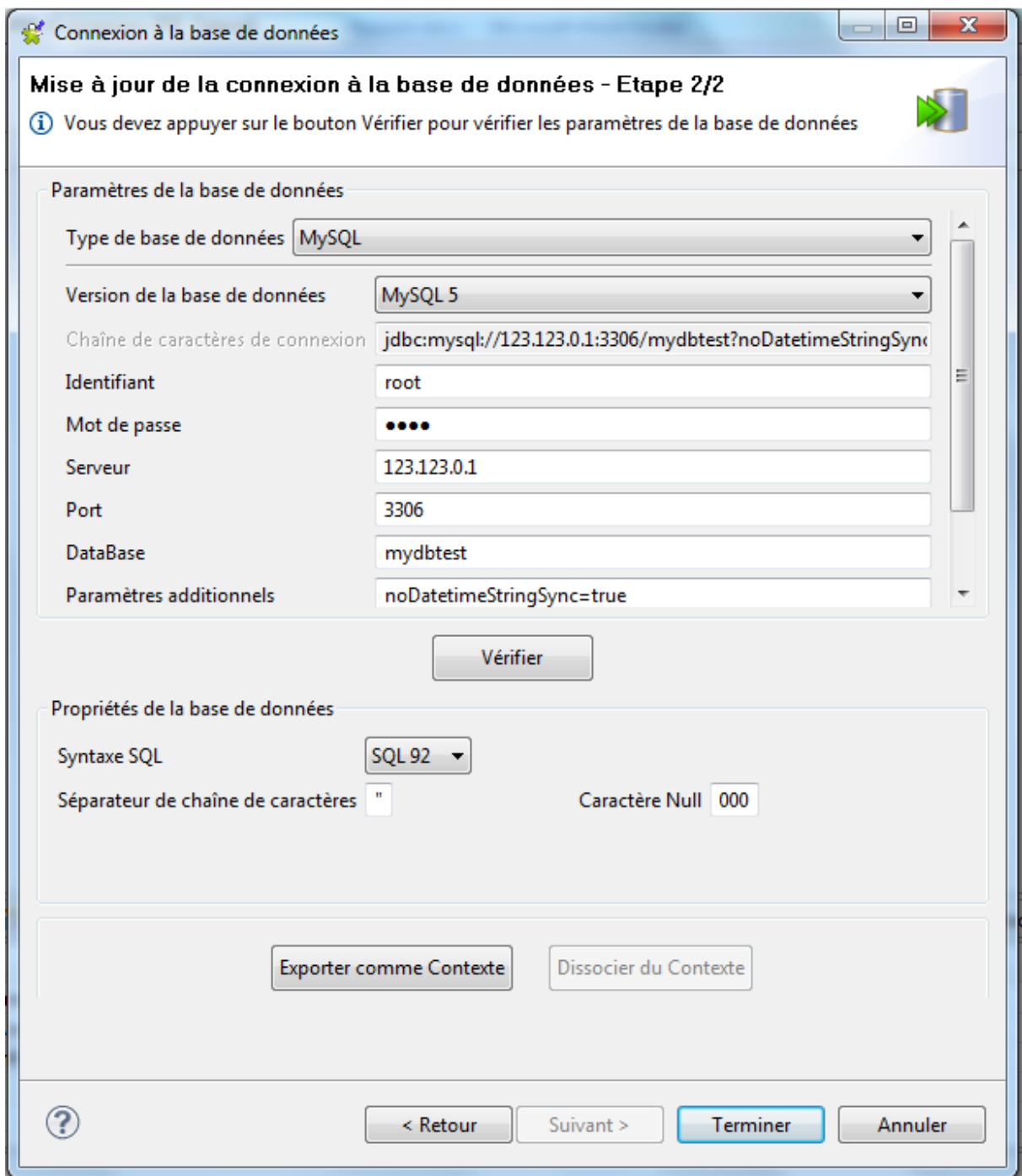


Figure 3.4 : Paramétrage de la connexion à la base de données source

Ainsi, établir la connexion avec les deux bases de données en question est indispensable afin de récupérer le schéma de la base de données Test2.

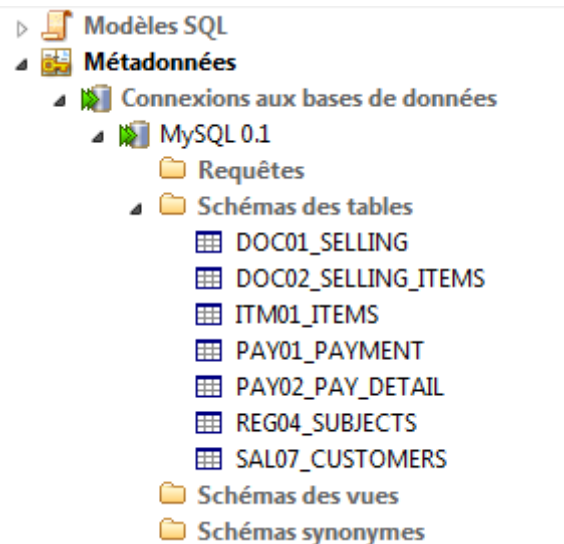


Figure 3.5 : Connexion avec la BD source et sélection des tables contenant les données à extraire

Une fois la connexion a été établie, nous pouvons importer les tables de la base source, sélectionner celles nécessaires pour réaliser la phase ETL et modifier à titre d'exemples leurs schémas et types de données s'il le faut.

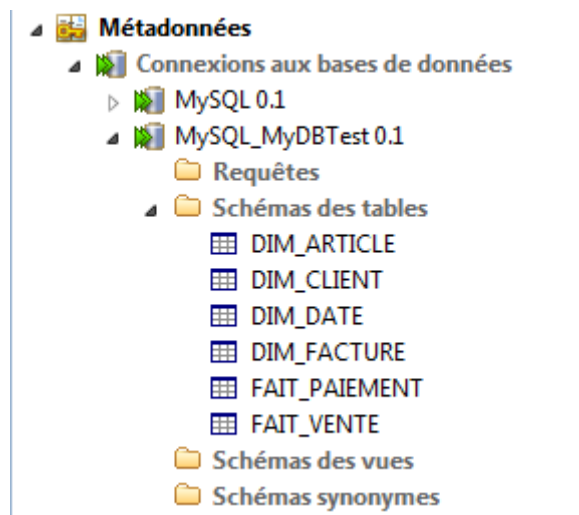


Figure 3.6 : Connexion avec le DataMart « MyDBtest » et sélection de toutes les tables de dimensions et de faits à alimenter.

4.3.Chargement des tables de dimensions :

Il y a quatre tables de dimensions à charger depuis les tables sélectionnées dans la base de données source en exploitant les composants Talend précédemment définis. Ces tables sont : DIM_CLIENT, DIM_ARTICLE, DIM_FACTURE et DIM_DATE.

4.3.1. DIM_CLIENT :

Cette table sera alimentée par les données contenues dans les deux tables SAL07_CUSTOMERS et REG04_SUBJECTS. La première, catégorisée SALES (ventes), contient les informations spécifiques uniquement aux clients à savoir les identifiants, types, modes de paiements etc. La deuxième, catégorisée REGISTER (registre) sert dans la base de données de l'ERP pour stocker les informations détaillées sur toute personne physique ou moral en interaction avec l'entreprise : (Nom, prénom/raison sociale, adresse, âge, etc.)

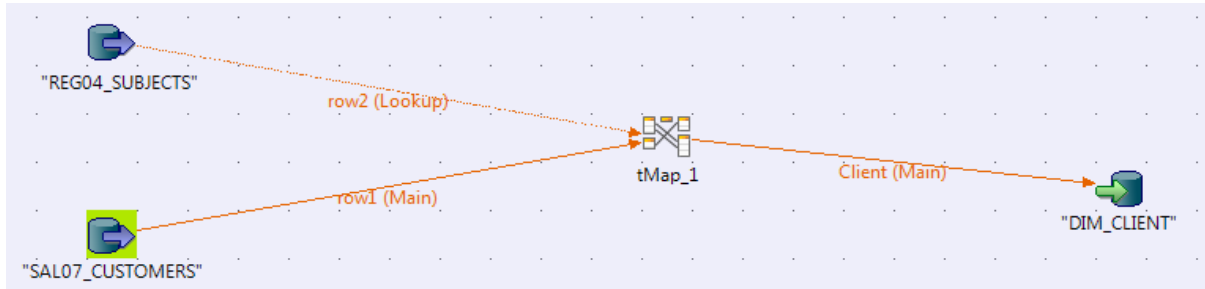


Figure 3.7 : Job DIM_CLIENT

4.3.2. DIM_ARTICLE :

Via un mapping grâce au composant tMap, la table DIM_ARTICLE va puiser juste les données nécessaires de la table ITM01_ITEMS réservée dans la base de données de l'ERP aux informations relatives aux articles mis en vente ou autre.

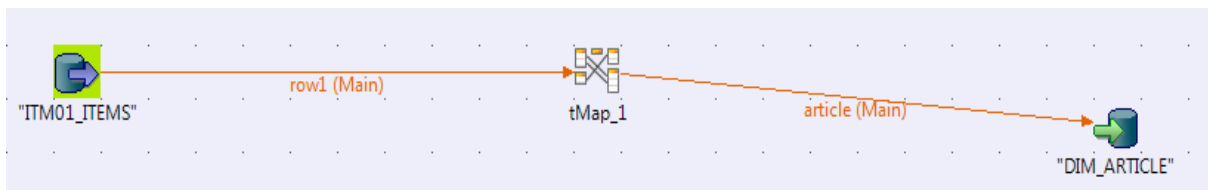


Figure 3.8 : Job DIM_ARTICLE:

4.3.3. DIM_FACTURE :

Une facture est une sorte de document qui indique le tiers auquel on a vendu un article avec des modalités de paiement données. La table DOC01_SELLING qui enregistre toutes les données relatives aux opérations de vente représente la source des données utiles pour former une dimension des factures.

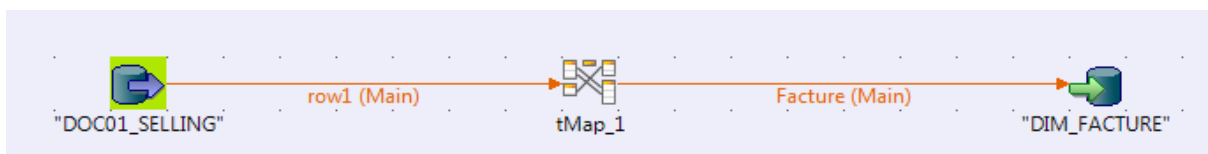


Figure 3.9 : Job DIM_FACTURE

4.3.4. DIM_DATE :

L'alimentation des dimensions relatives au temps a posé un nombre de questions dans les milieux de l'informatique décisionnelle. Les experts du domaine (nos remerciements aux membres du forum www.labeldecisionnel.com) nous ont recommandé de suivre une politique

de chargement à part pour la table DIM_DATE en utilisant un script SQL qui permet de charger un large intervalle de dates qui s'incrémentent régulièrement.

4.4.Chargement des tables de faits :

4.4.1. FAIT_VENTE :

Nous rappelons que la table de faits FAIT_VENTE est celle qui permet de calculer les indicateurs relatifs au chiffre d'affaire et ses projections sur les clients, les articles et le temps. C'est pour cela qu'elle sera alimentée via quatre tables de la base de données de l'ERP : DOC01_SELLING, DOC02_SELLING_ITEMS, SAL07_CUSTOMERS et REG04_SUBJECTS.

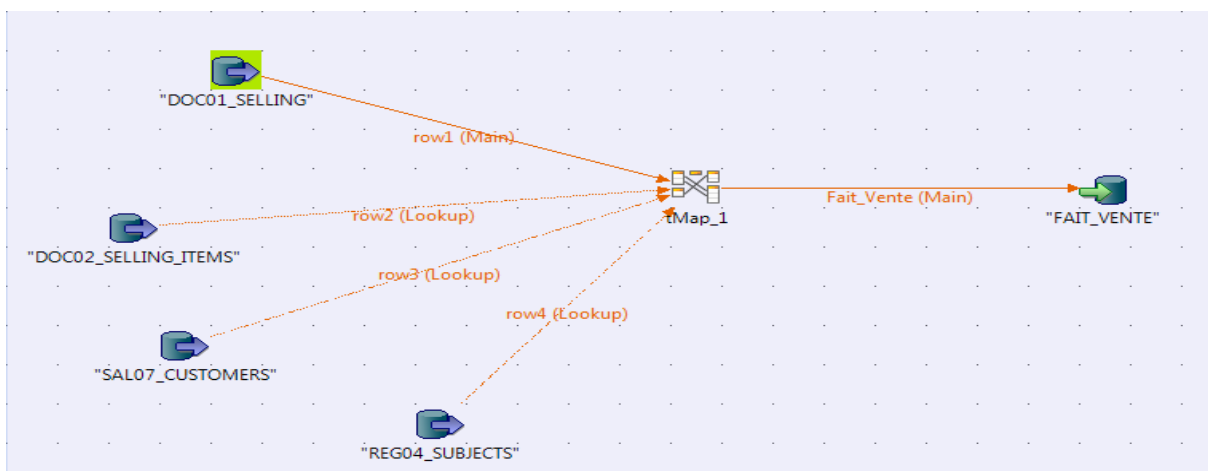


Figure 3.10 : Job FAIT_VENTE

4.4.2. FAIT_PAIEMENT :

Cette table doit contenir des données permettant de fournir des statistiques sur les paiements qu'un client a effectués et par la suite on peut calculer ses soldes, fiabilité...

Elle sera alimentée via jointure avec PAY01_PAYMENT et SAL07_CUSTOMERS.

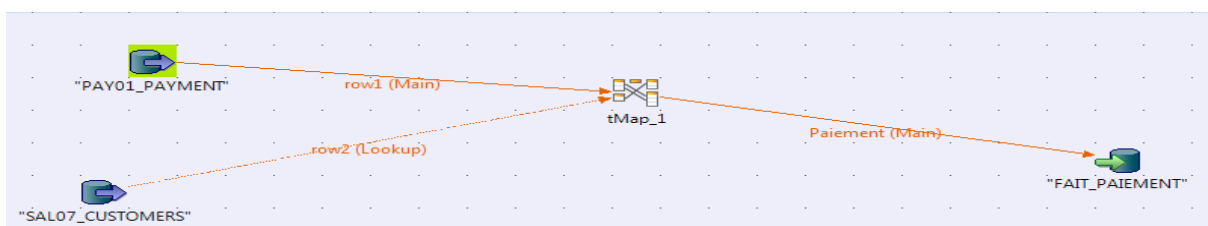


Figure 3.11: Job FAIT_PAIEMENT

5. Mise à jour des données du DataMart :

Le rafraîchissement périodique des données du DataMart est primordial. Plusieurs méthodes de rafraîchissement sont disponibles à savoir l'exécution manuelle des scripts de jobs ou l'utilisation des fonctionnalités garanties par l'ETL Talend Open Studio.

Il est possible de choisir l'action à appliquer sur les données entrantes aux tables de destination parmi « update or insert » si les mises-à-jours sont jugées plus volumineuses que

les insertions et « insert or update » en situation inverse. Dans notre cas de système décisionnel de ventes, nous prévoyons un nombre d'insertions beaucoup plus supérieur à celui des mise-à-jours.

Talend Open Studio contient un programmeur de tâches « scheduler » qui permet de configurer le script de chargement. Dans le cas du rafraîchissement des données de notre DataMart, nous choisissons d'ordonnancer une mise-à-jour chaque jour à minuit.

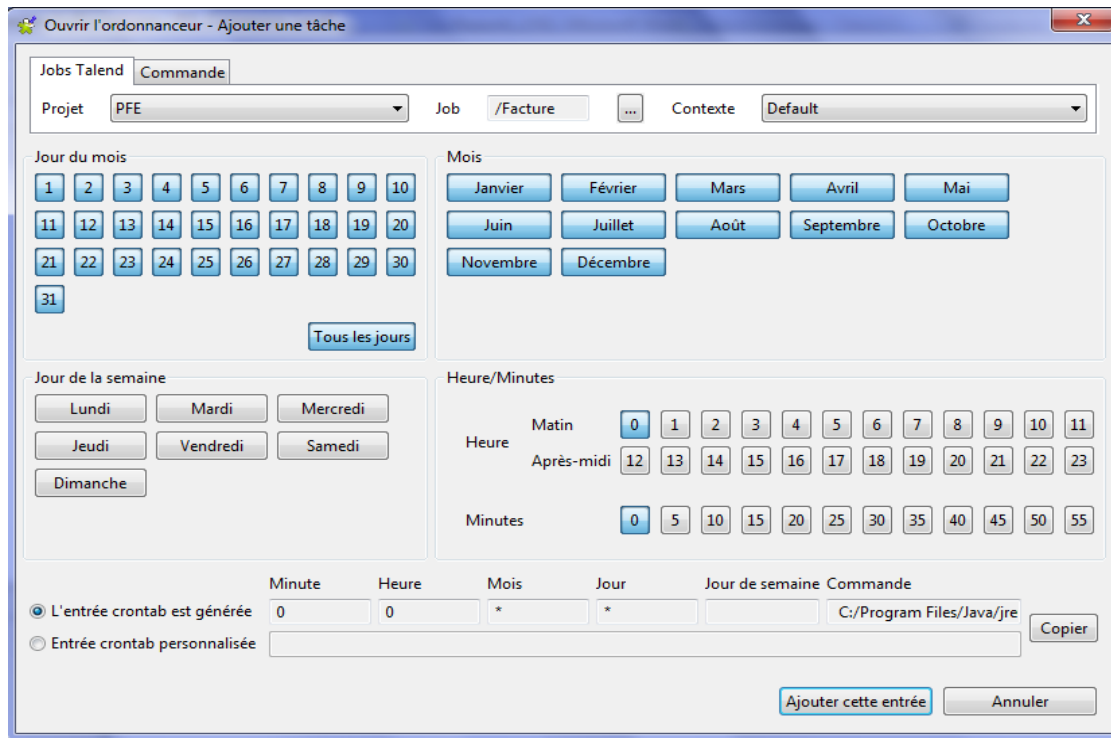


Figure 3.12: Scheduler de Talend Open Studio

6. Conclusion :

Nous avons présenté dans ce chapitre les étapes de réalisation du DataMart des ventes depuis la modélisation et la conception jusqu'à l'exécution des jobs générés par Talend Open Studio pour mise à jour. Maintenant que notre structure destination est opérationnelle, nous pouvons l'exploiter pour générer des rapports d'état à employer dans une partie ultérieure du projet.

CHAPITRE IV : PROCESSUS DE REPORTING :

Pour faciliter la tâche aux responsables, un environnement de Reporting est primordial. La conception et la réalisation de « Optimis Viewer » a été faite pour répondre à cette fonction. Nous notons que les résultats qui apparaissent dans les rapports sont des données de test.

1. Généralités :

Maintenant que le DataMart que nous avons créé est alimenté avec toutes les informations nécessaires à l'élaboration des rapports et que le choix de l'outil de reporting est déjà fait, il suffit d'entamer le travail requis : la conception des rapports.

En fait, JasperReports, étant l'outil de reporting en question, couplé avec l'éditeur graphique iReport vont nous permettre de créer des rapports dont les résultats pourraient être affichés à l'écran, imprimés ou stockés dans des fichiers de différents formats.

JasperReports est entièrement développé en Java et peut être intégré dans une gamme très variée d'applications Java (y compris les applications J2EE). Son objectif principal est de fournir un moyen simple et flexible pour la génération de documents.

La création de rapports avec JasperReports se déroule généralement en 4 étapes :

- L'obtention d'un fichier modèle XML (à l'aide de l'éditeur graphique iReport)
- La construction du rapport à partir du modèle
- Le remplissage des différents champs du rapport avec les données en provenance de la base de données (comme elles peuvent provenir de d'autres sources).
- L'exportation du résultat dans plusieurs formats possibles (PDF, HTML, XHTML, EXCEL, document WORD, document CSV, PowerPoint ...).
- L'exportation du résultat dans plusieurs formats possibles (PDF, HTML, XHTML, EXCEL, document WORD, document CSV, PowerPoint ...).

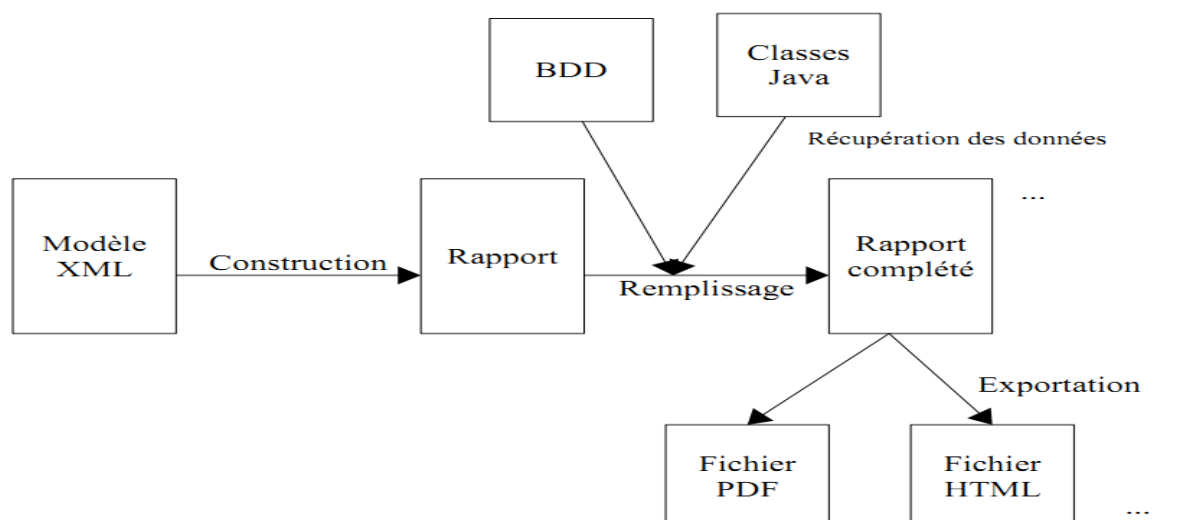


Figure 4 .1 : Etapes de création d'un rapport

Le fonctionnement de JasperReports est relativement simple. En effet, tous les concepts tournent autour du langage Java. Une fois le modèle XML (JasperDesign) compilé, il est chargé dans un objet Java (JasperReport) qui peut lui-même être sérialisé et stocké dans un fichier (avec l'extension .jasper). Cet objet sérialisé est alors utilisé lorsque l'application désire compléter le rapport avec des données. En fait, la définition du rapport nécessite la compilation de toutes les expressions Java déclarées dans le modèle XML. Le résultat obtenu après le processus de remplissage des champs est un nouvel objet Java (JasperPrint) qui représente le document final. Celui-ci peut être stocké sur disque pour un usage ultérieur (sous forme sérialisée et avec l'extension .jrprint), directement imprimé ou encore transformé dans un format lisible (PDF, HTML, ...).

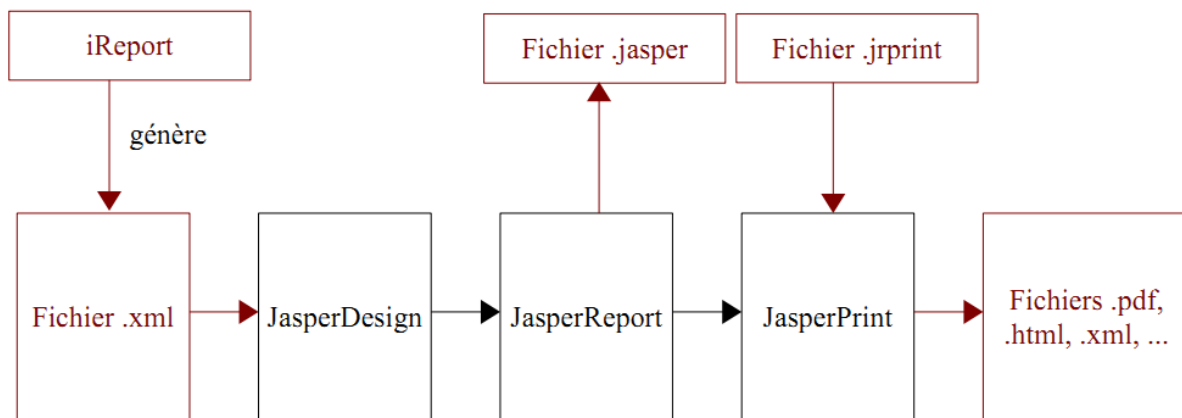


Figure 4.2 : Fonctionnement de JasperReports

2. Les rapports :

Les rapports répondants aux indicateurs relevés au préalable sont comme suit :

- La répartition du chiffre d'affaire réalisé par l'entreprise sur l'ensemble des clients enregistrés.
- La répartition du chiffre d'affaire réalisé pendant une durée donnée sur l'ensemble des articles vendus.
- Le total des paiements reçus par rapport au total des factures émises pour un client donné.

Bien sûr avant d'entamer la conception des rapports, il faut se connecter à la source de données, et ceci via un utilitaire très intuitif qu'on démarre depuis l'accueil de iReport en cliquant sur l'icône qu'affiche la capture ci-dessous.

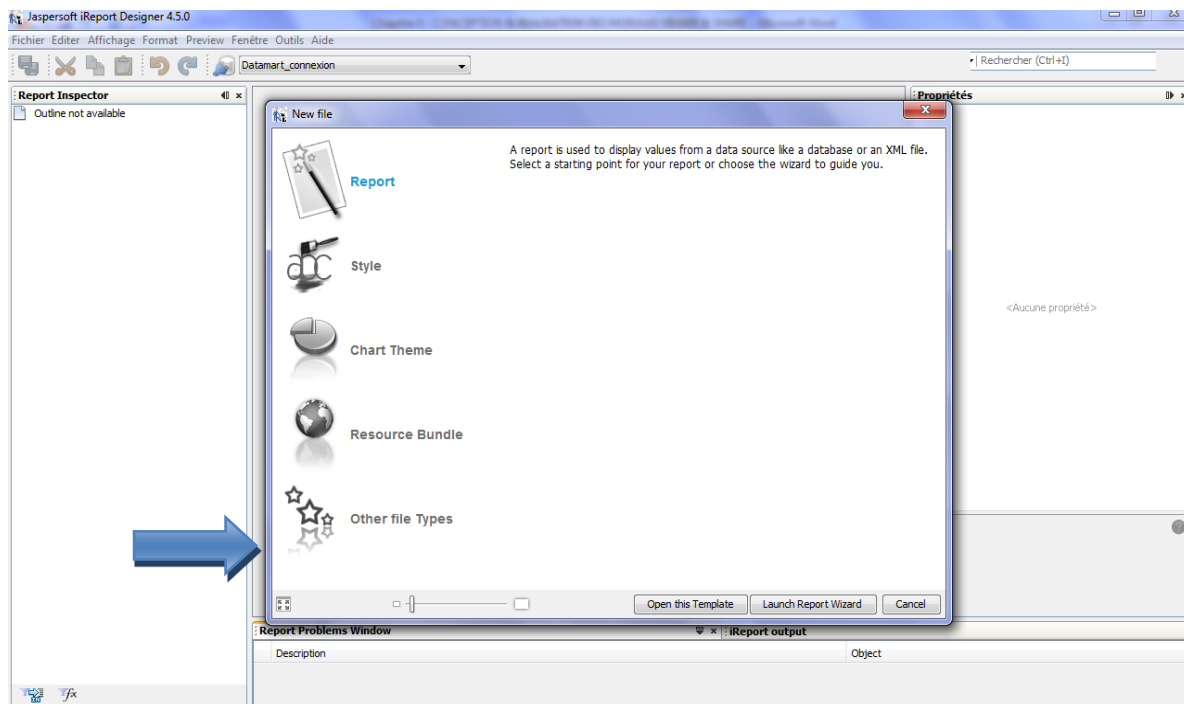


Figure 4.3 : Interface d'accueil de « iReport »

iReport offre le choix entre plusieurs DRIVERS de connexion, une liste qui couvre la majorité des sources de données courantes. Dans notre cas, nous devons choisir un driver JDBC.

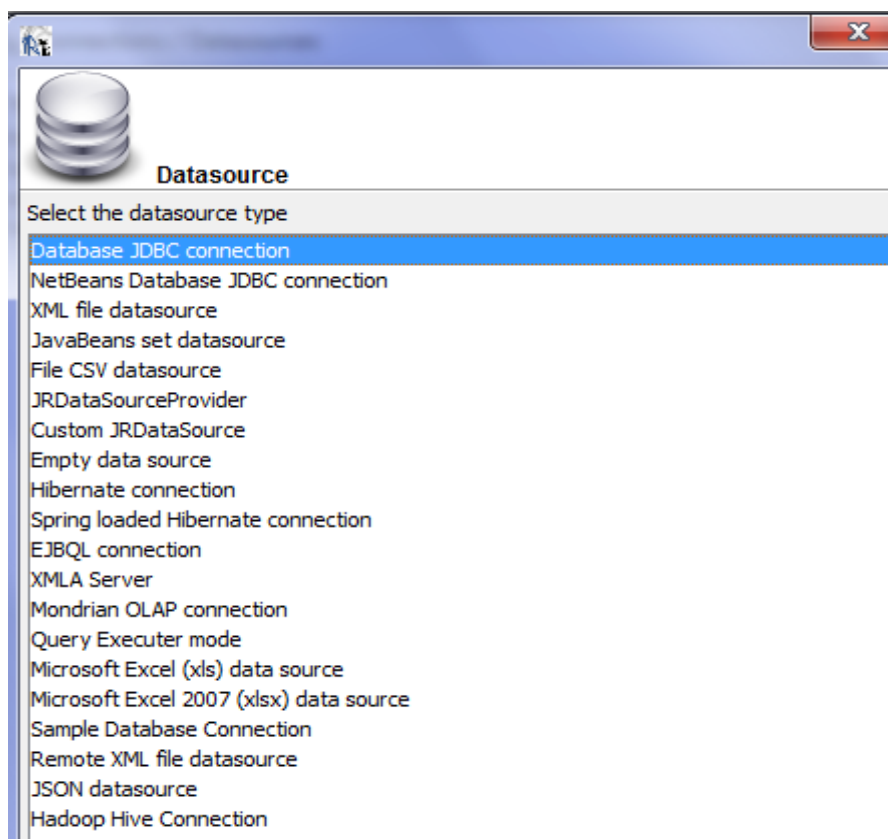


Figure 4.4 : Liste des drivers de connexions aux sources de données

Pour établir la connexion, il est impératif de saisir les informations nécessaires à la connexion. Chose qui se fait moyennant l'interface ci-dessous.

The screenshot shows a 'Database JDBC connection' dialog box. At the top left is a database icon. The title is 'Database JDBC connection'. Below the title is a 'Name' field containing 'Connexion_DataMart'. The 'JDBC Driver' is set to 'MySQL (com.mysql.jdbc.Driver)'. The 'JDBC URL' is 'jdbc:mysql://123.123.0.1/mydbtest'. Below this is a 'JDBC URL Wizard' section with 'Server Address' (123.123.0.1) and 'Database' (mydbtest) fields, and a 'Wizard' button. The 'Username' field contains 'root' and the 'Password' field is masked with dots. There is a 'Save password' checkbox. A warning message with a yellow triangle icon reads: 'ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.' At the bottom are 'Test', 'Save', and 'Cancel' buttons.

Figure 4.5 : Paramétrage de la connexion au DataMart

Avant de saisir la requête SQL et les paramètres correspondants, l'utilitaire de création propose de choisir un Template parmi plusieurs choix pré-faits, sinon concevoir un Template personnalisé (ce qui fut notre cas afin d'y ajouter les logos des entreprises concernées).

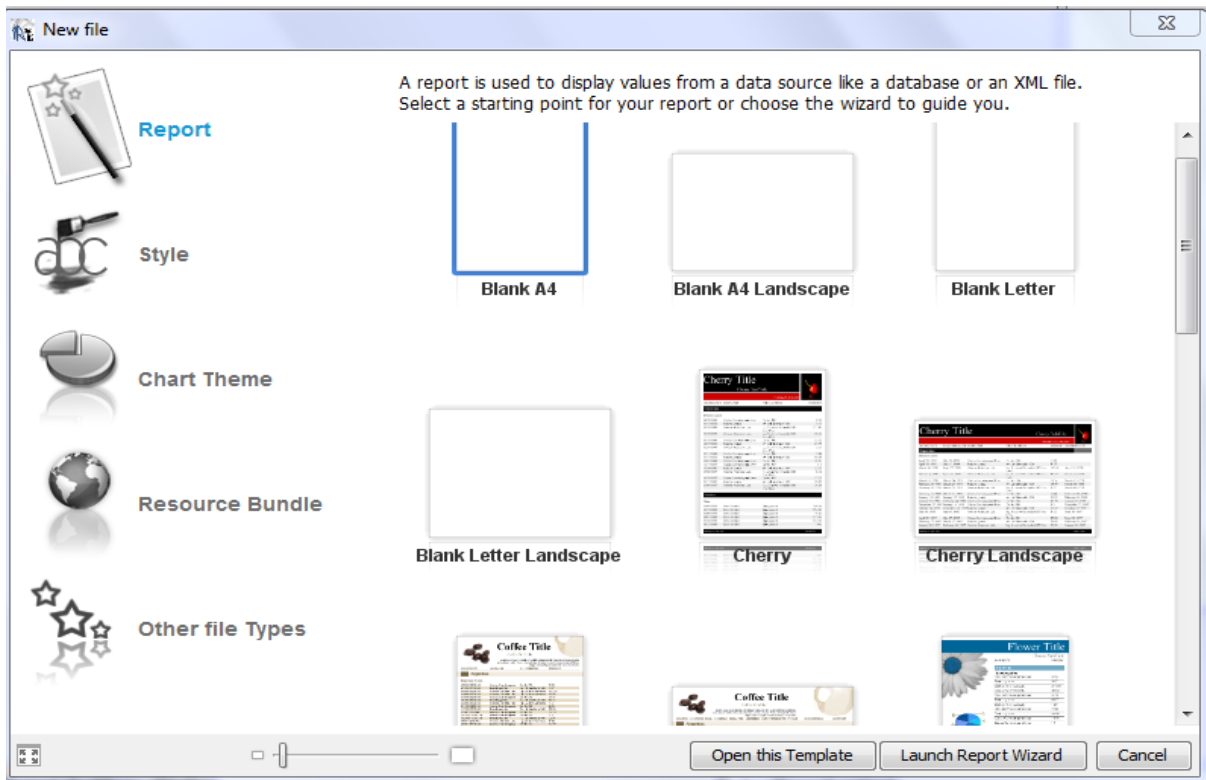


Figure 4.6 : Liste des Templates proposés par « iReport »

Le Template choisi, il est temps de saisir la requête SQL correspondante au rapport, une requête qui définira les données qui vont garnir le rapport une fois son gabarit conçu.

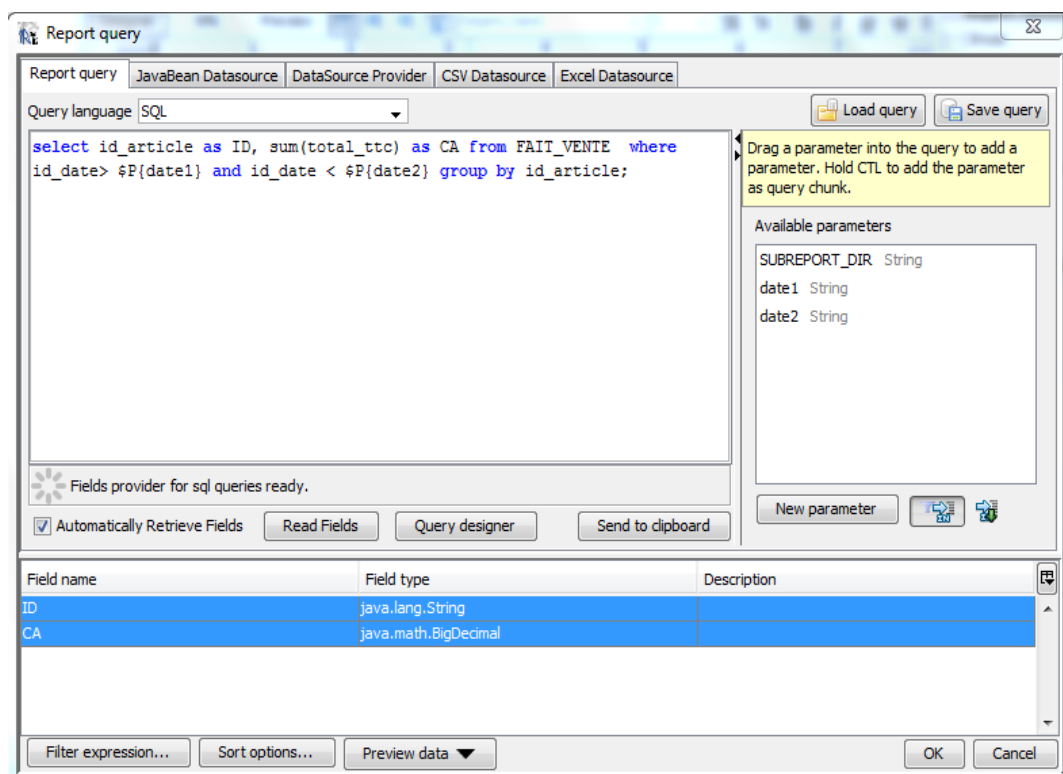


Figure 4.7 : Saisie de la requête SQL

Après avoir suivi les étapes, qui sont plus au moins similaires pour la conception de nos trois rapports nous passons par la suite à ce qui est spécifique à chaque rapport.

2.1.Chiffre d'affaire par client :

La requête SQL qui permet d'afficher le chiffre d'affaire réparti sur le total des clients

```
select c.id_client,c.nom1_client,c.nom2_client , sum(v.total_ttc) as  
chiffre_affaire_par_client  
from mydbtest.FAIT_VENTE v, mydbtest.DIM_CLIENT c  
where v.id_client=c.id_client  
  
group by v.id_client;
```

Dans ce rapport, nous désirons présenter à l'utilisateur final la répartition du chiffre d'affaire (dans notre DM, représenté par la sommation de *total_ttc* qui est la valeur tout taxe compris d'une opération de vente groupée par client). Ce champ figure dans la table de faits FAIT_VENTE.

Les informations spécifiques au client à savoir son identifiant et son nom (ou raison sociale en cas d'entreprise) sont extraits de la table de dimension DIM_CLIENT.

Dans ce rapport, nous avons préféré ne pas utiliser l'option de paramètres offerte par iReport parce qu'il est rapporté uniquement de la part de tous les clients enregistrés dans le chiffre d'affaire réalisé depuis le début des enregistrements.

OSS OriginalSoft System			
RAPPORT: CHIFFRE D'AFFAIRES PAR CLIENT			
Chiffre d'affaire total:		122098.00000	MAD
ID	Prénom (R.S.)	Nom (Desc.)	CA (MAD)
C10	Evistel	Client	115000.00000
C12	Mounia	OUHAJOU	1200.00000
C2	aa		48.00000
C6	EMI	Ecole	2400.00000
C8	Oracle	California based company specialized in IT services	3450.00000

lundi 12 mars 2012 Page 1 of 1

Figure 4.8 : Rapport des chiffres d'affaires par client

2.2.Chiffre d'affaire par produit pendant une durée donnée :

Ce rapport paramétrable, les paramètres à entrer sont les dates de début et de fin de la période dans laquelle il faut calculer le chiffre d'affaire total et sa répartition sur les produits (articles) vendus pendant cette période.

Dans ce rapport, il y a utilisation de sous rapports (*SubReports*) pour l'affichage du graphe en camembert et l'affichage à l'entête de la somme totale du chiffre d'affaire.

La requête SQL entrée est:

```
select id_article as ID, sum(total_ttc) as CA from FAIT_VENTE where
id_date > $P{date1} and id_date < $P{date2} group by id_article;
```

Voici un aperçu du rapport généré

(Rq : Nous utilisons la notation anglo-saxonne des dates en AAAA-MM-JJ vue que c'est la notation utilisée dans le DM).

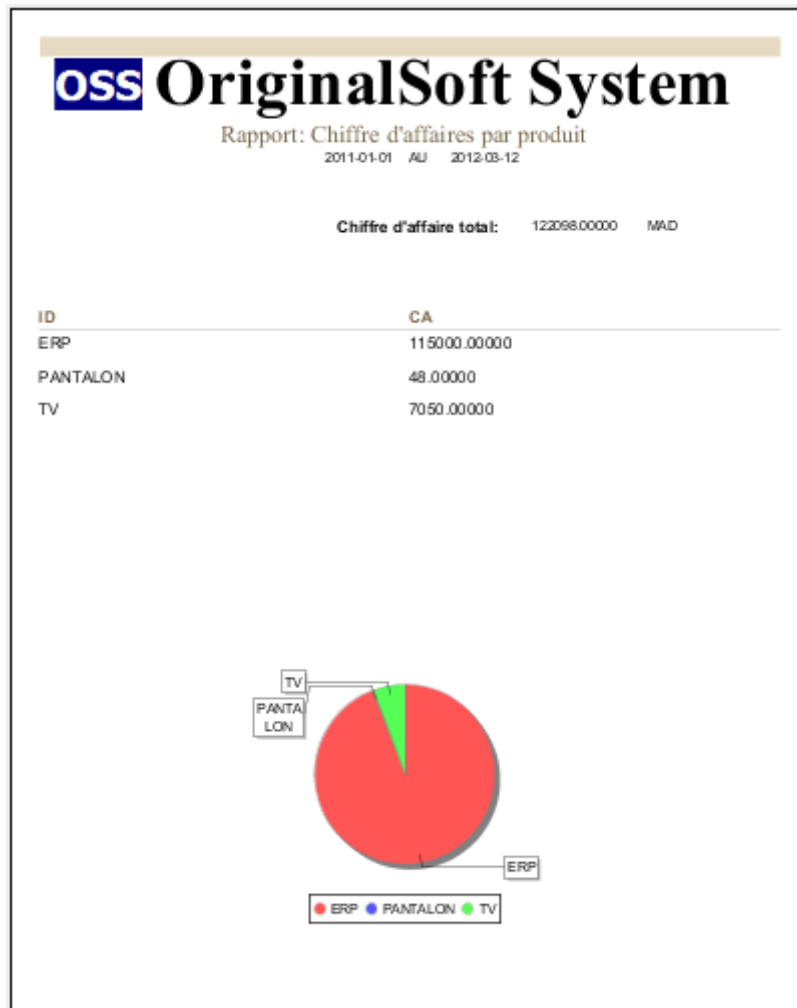


Figure 4.9 : Rapport des chiffres d'affaires par produit pendant une durée donnée

2.3.Total des paiements reçus par rapport au total des factures émises pour un client donné :

Ce rapport prend en paramètre le nom (raison sociale) du client et présente à son lecteur des calculs liés à ses paiements et facturations à savoir la balance qui est la différence des sommes facturées et des sommes payées et le ratio des mêmes variables. Ces calculs permettent à l'entreprise de dégager le profil du client traité, son aptitude à payer et par la suite sa fiabilité.

La requête SQL utilisée est la suivante :

```

select mydbtest.DIM_CLIENT.id_client ,mydbtest.DIM_CLIENT.nom1_client
as Raison_Sociale, mydbtest.DIM_CLIENT.nom2_client as
Description,mydbtest.DIM_CLIENT.ville_client as Ville,
mydbtest.DIM_CLIENT.pays_client as Pays, sum(total_paye) as total_paye,
sum(total_ttc) as total_facture, sum(total_ttc) - sum(total_paye) as
Balance, sum(total_paye)/sum(total_ttc) * 100 as Ratio from
mydbtest.FAIT_PAIEMENT,mydbtest.FAIT_VENTE,mydbtest.DIM_CLIENT where
mydbtest.FAIT_VENTE.id_client=mydbtest.DIM_CLIENT.id_client AND
mydbtest.FAIT_PAIEMENT.id_client=mydbtest.DIM_CLIENT.id_client AND
mydbtest.DIM_CLIENT.nom1_client=$P{nom_client} GROUP BY id_client;

```

En saisissant la valeur de paramètre Raison Sociale : EMI, nous obtenons le rapport suivant :

OSS OriginalSoft System

Rapport: Total des paiements reçus pour l'ensemble des
factures émises au client: **EMI**

12/03/2012

id_client	Raison_Sociale	Description	Ville	Pays	Total payé (MAD)	Total facturé (MAD)	Balance (MAD)	Ratio (%)
C6	EMI	Ecole	Rabat	MAROC	3400.00000	4800.00000	1400.00000	70.833333333

Figure 4.10 : Rapport du total des paiements reçus pour l'ensemble des factures émises pour un client donné

3. Exécution des rapports dans Optimis Report Viewer :

La compilation de rapports travaillés dans iReport, dans sa version 4.5.0 génère les fichiers portant les extensions : .jrxml et .jasper. Après exécution, le rapport se génère également dans le format sélectionné, dans ce cas, c'est un fichier PDF.

Pour tester la capacité de la plateforme à visualiser des rapports. Nous essayons d'exécuter les rapports créés dans *Optimis*.

Pour ce faire, nous transférons les fichiers des rapports de la machine où ils sont localisés au server via le client SFTP WinSCP afin de les charger dans la plateforme.

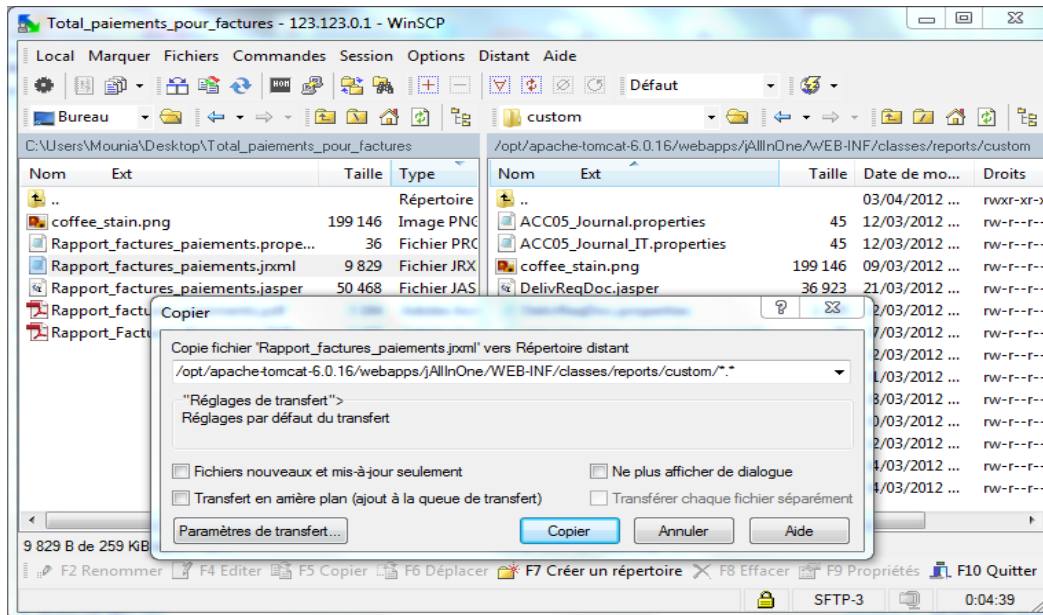


Figure 4.11 : Transfert des fichiers dans WinSCP

Nous chargeons le rapport dans *Optimis* et nous indiquons le paramètre `nom_client` du rapport relatif aux factures et paiements :

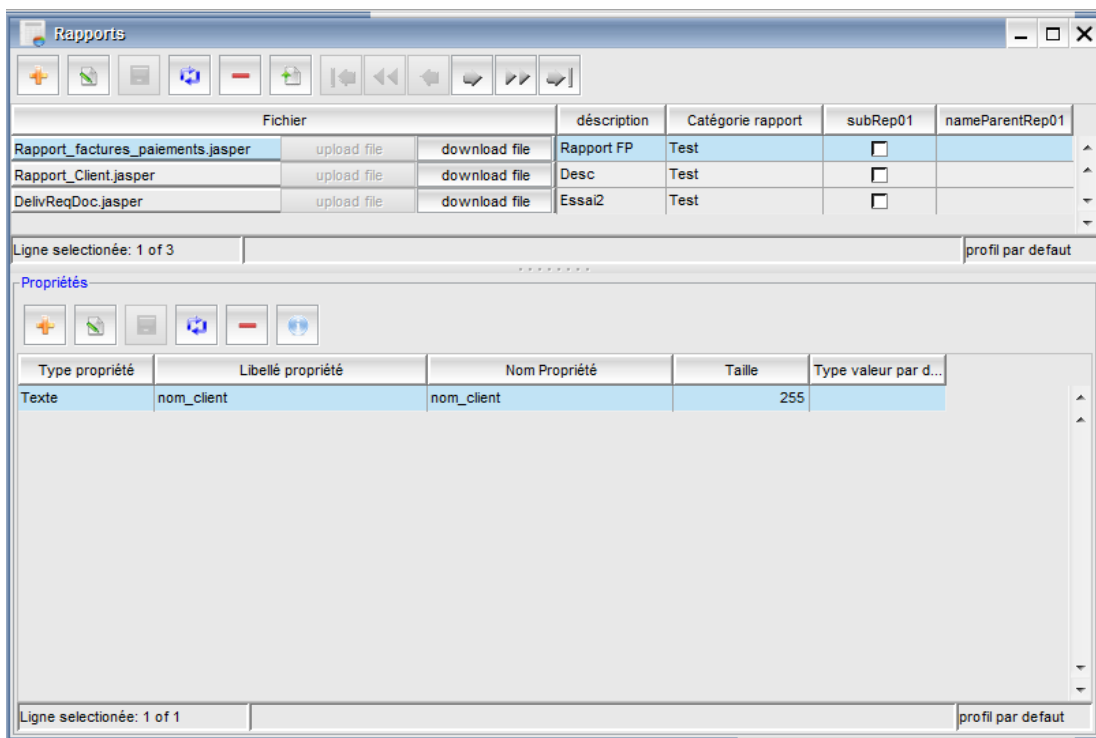


Figure 4.12 : Chargement de rapport dans *Optimis*

Nous saisissons la valeur : « EMI » du paramètre `nom_client` pour visualiser l'état de solvabilité du client EMI.

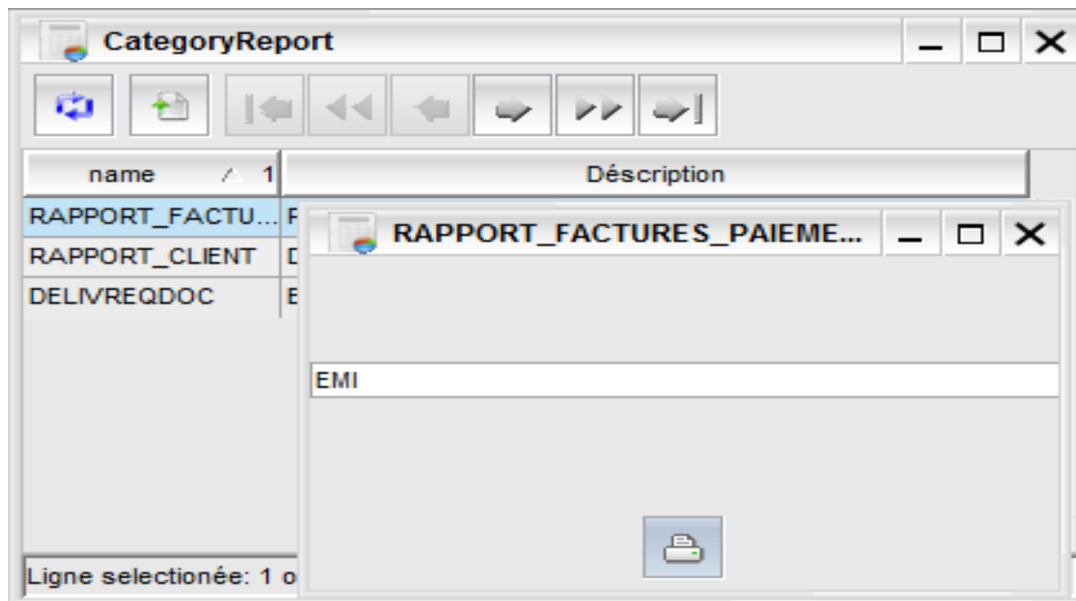


Figure 4.13 : Saisi de valeur de paramètre de rapport dans *Optimis*

A l'exécution, un message d'erreur est signalé. Après examen du code de la fonctionnalité de visualisation et demande de renseignement auprès du forum d'utilisateurs d'une version similaire de jAllInOne, nous constatons que les rapports développés dans les versions ultérieures à la 1.2.7 d'iReport ne sont pas compatibles avec l'outil de visualisation au niveau de la plateforme.

Nous avons remédié à ce problème en recompilant les fichiers de rapports dans iReport dans sa version 1.2.7. La recompilation génère des fichiers .jasper exécutables par Viewer.

Pour une solution perpétuelle, nous suggérons comme perspective d'upgrader le noyau d'exécution de Viewer pour s'aligner avec les dernières versions d'iReport.

Conclusion :

Ce chapitre a traité du processus de Reporting. Ce dernier consiste principalement à concevoir, à créer et à publier les rapports sous le format voulu. Ainsi, il faut mettre ces rapports sous la disposition des gens intéressés et ceci via « Optimis Viewer ».

CHAPITRE V : CONCPETION ET REALISATION D'OPTIMIS REPORT VIEWER ET SHARE :

Nous détaillons dans ce chapitre la dernière phase de notre projet. C'est la phase dans laquelle nous agissons directement sur la plateforme Optimis Report en concevant et développant le module Reporting.

1. Méthodologie :

La focalisation sur une méthode agile pour mener ce sprint découle primordialement du fait que nous avons adopté la méthode SCRUM, agile elle-même, pour gérer l'ensemble de notre projet de fin d'études. C'est pour cette raison, d'abord, que nous avons décidé de ne pas dévier de la logique agile adoptée dès le départ.

Avec la panoplie des méthodes agiles présentes, nous avons dû faire un choix judicieux pour adopter la méthode la plus adaptée à la conduite du dit sprint. Pour cela, nous nous sommes appuyés sur des documentations de plusieurs méthodes ainsi que sur la nature du projet à mener dans cette période pour cerner le choix entre les deux méthodes XP et 2TUP.

Notre expérience avec le cycle de développement en Y adopté en Two Track Unified Process a privilégié le choix de 2TUP face à la méthode XP.

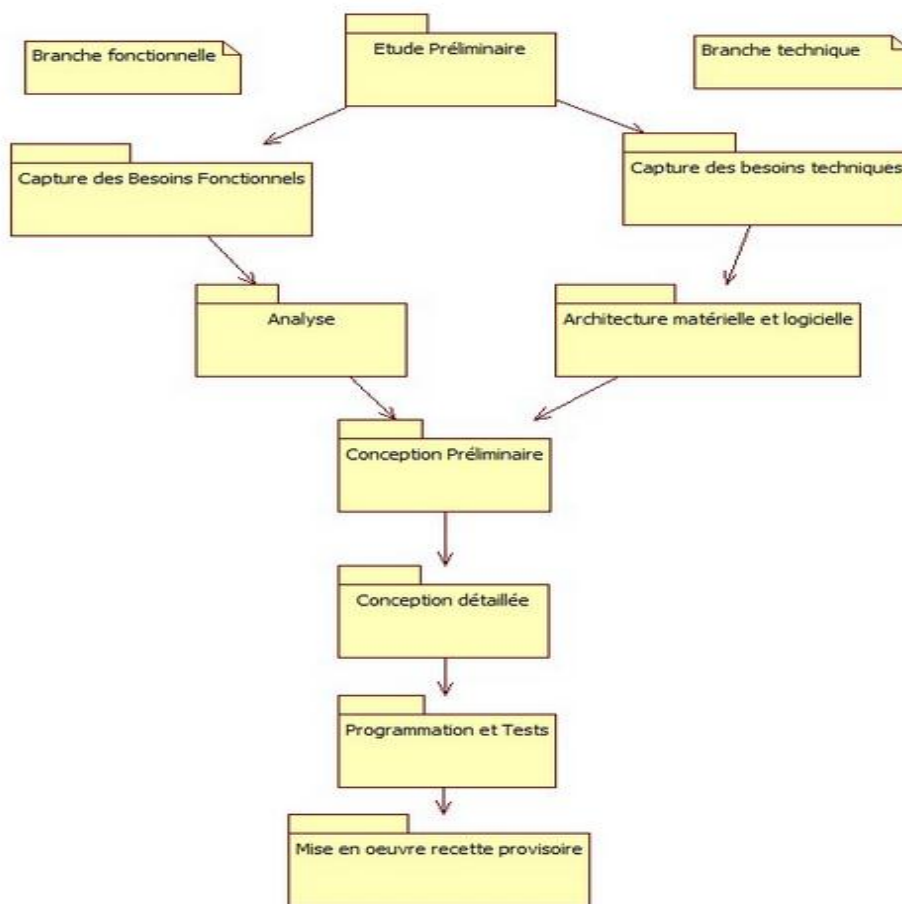


Figure 5.1 : Le cycle de développement selon la méthode 2TUP

2. Analyse fonctionnelle :

2.1. Analyse et critique de l'existant :

Avant notre intervention, le module Reporting de l'ERP Optimis ne permettait que la visualisation de rapports installés en disque dur.

La visualisation dans ce module est intégrée dans le Viewer. Le noyau de ce dernier est la bibliothèque libre JasperReports.

Les rapports conçus dans une version de iReport supérieure à 1.2.7 ne sont pas visualisables dans le Viewer du module Reporting.

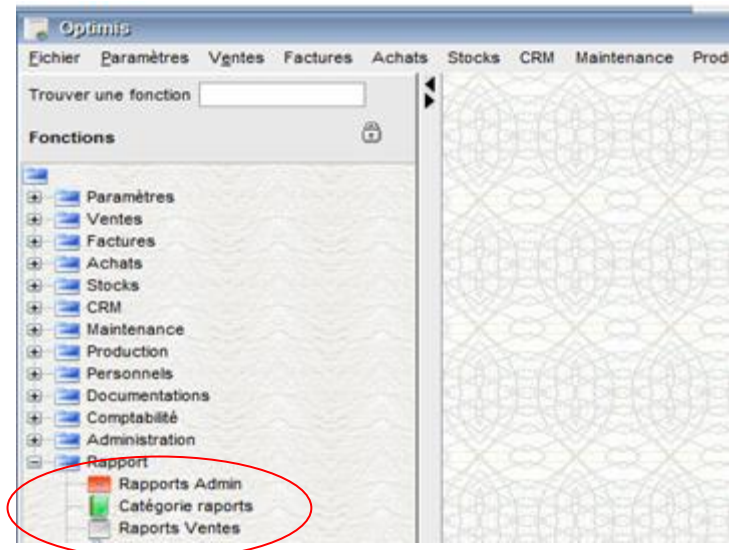


Figure 5.2 : Interface d'Optimis avec l'onglet du reporting ouvert

L'administrateur charge les rapports et définit les paramètres de chacun avec des valeurs par défaut :

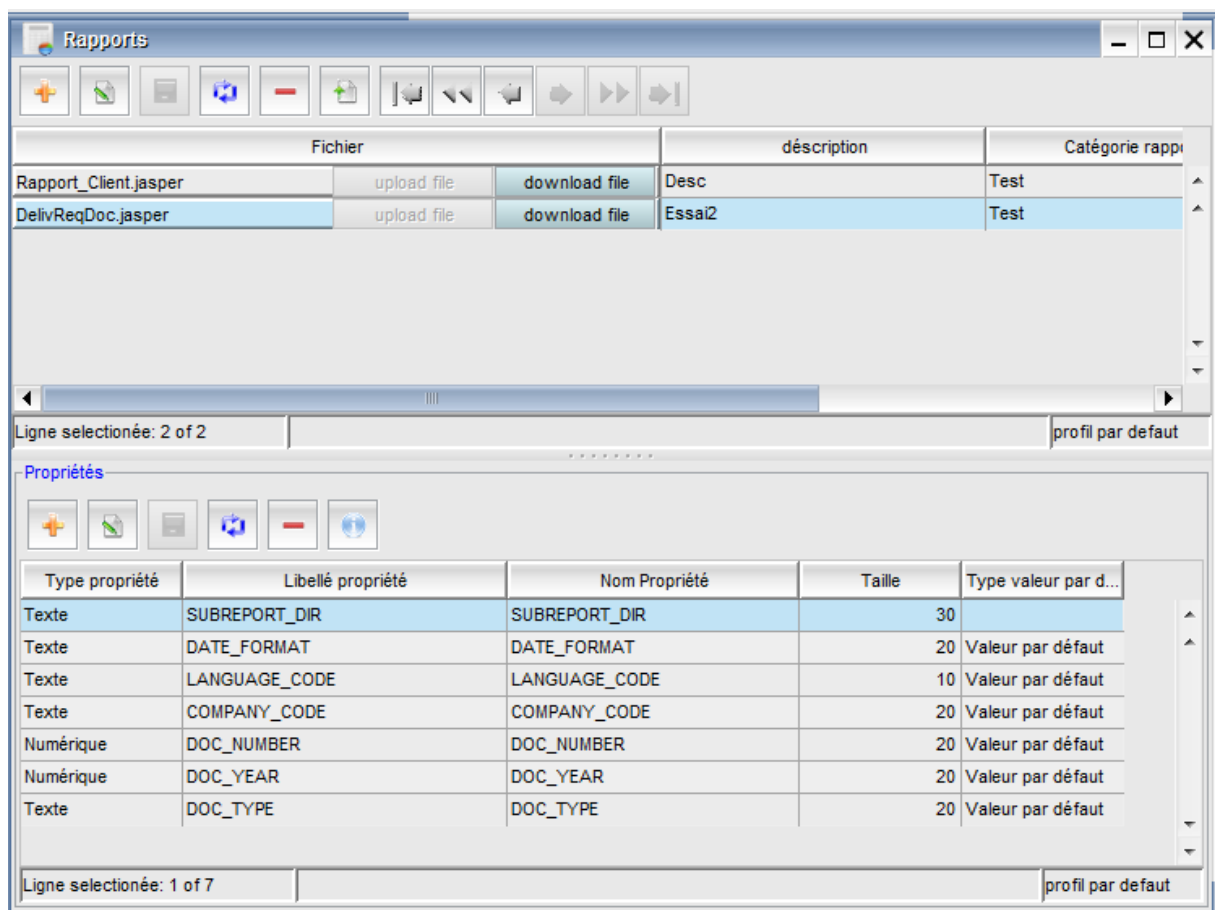


Figure 5.3: Interface de chargement des rapports dans Optimis

Le choix de la catégorie des rapports doit passer d'abord par la prédéfinition des catégories dans l'onglet : « Catégorie raports » :

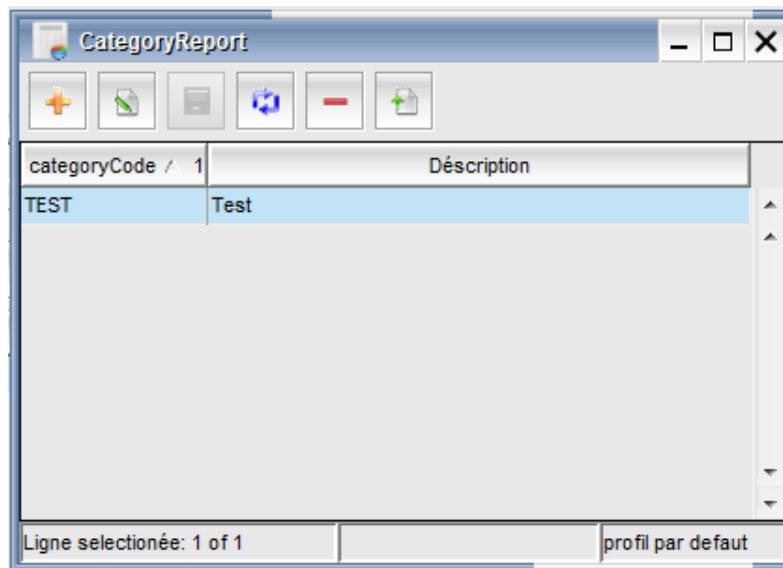


Figure 5.4 : Liste des catégories des rapports

A l'affectation des rapports à une catégorie, cette dernière s'ajoute à la barre des onglets :

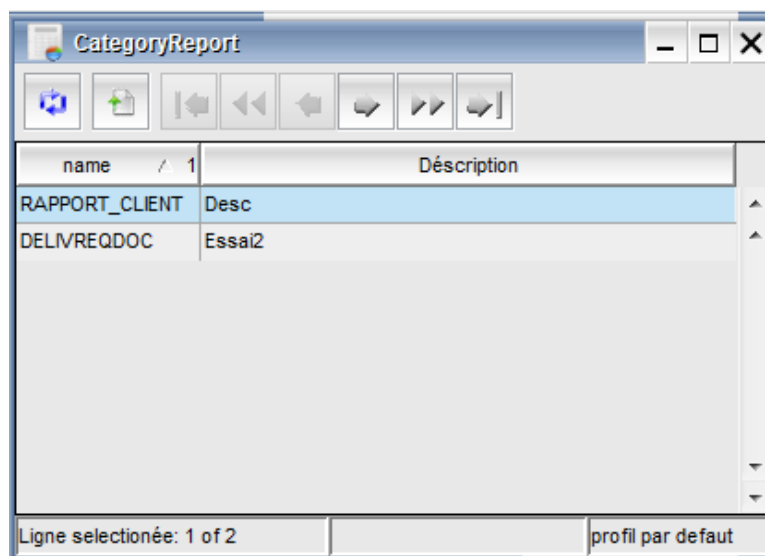


Figure 5.5 : Liste des rapports affectés à une catégorie

Nous sommes amenés à conclure que la plateforme ne permet que l'exécution des rapports. Le mécanisme de manipulation des rapports n'inclut pas la gestion des droits d'accès, ainsi, l'utilisateur exécute les rapports existant dans la machine ce qui prive l'utilisateur d'une manipulation simple et réduit la confidentialité des rapports.

2.2. Etude de la problématique :

La plateforme est destinée à l'usage des employés de PME désirant bénéficier de rapports d'activité de différents secteurs, pour cela, il faut étendre le périmètre fonctionnel de la plateforme jusqu'à permettre à chaque utilisateur de la plateforme d'accéder à des catégories de rapports données et les exécuter sur le Viewer.

Le but du reporting est de bénéficier les différents tiers d'un organisme de rapports d'état. Pour des raisons de confidentialité et d'hierarchie, la plateforme ne peut pas être accessible à tous les associés. Tenant compte de ces faits, nous avons proposé un autre mode d'accès aux rapports : la diffusion en mailing lists.

2.3. Description des fonctionnalités principales :

Après réunions avec les décideurs, nous sommes parvenus à relever les fonctionnalités principales à rajouter à la plateforme. Ces fonctionnalités sont les suivants :

- **Gestion des accès aux utilisateurs :** L'administrateur doit d'abord gérer la liste des comptes d'utilisateurs qu'il crée et auxquels il définit des rôles (profiles) précis. Pour des raisons de sécurité mais aussi de besoin métier de chaque utilisateur, l'administrateur doit gérer l'accès des utilisateurs à des catégories de rapports puis à des rapports eux-mêmes.
- **Exécution personnalisée des rapports :** L'utilisateur peut choisir et saisir les paramètres selon lesquels il va exécuter ses rapports et définir leurs formats (PDF, HTML...) pour qu'il puisse avoir exactement les informations désirées pour sa prise de décision.
- **Création des listes de diffusion :** L'administrateur doit créer des listes de diffusion et y ajouter des destinataires par saisi de leurs informations.
- **Gestion de la diffusion des rapports :** Via cette fonctionnalité, OSS vise à élargir les capacités de son produit jusqu'à l'automatisation de la diffusion des rapports sur leurs bénéficiaires. L'administrateur donne accès de certains rapports à des listes de diffusion et selon le besoin planifie les horaires d'envoi.

3. Analyse technique :

Dans notre mise en œuvre sur la plateforme, nous devons tenir compte des contraintes techniques imposées par la partie prenante. Ceci pour respecter l'homogénéité de la solution et permettre par la suite une facilité d'intégration et maintenabilité.

Les tables à créer pour accueillir les données utiles pour ces fonctionnalités doivent se joindre à la base de données Test2 de l'ERP, une base de données MySQL. Le modèle relationnel des données⁸ englobant les nouvelles tables explicite les liens entre elles et quelques tables de la base de données Test2.

⁸ Annexe H : Modèle relationnel de données

Les développements doivent se faire également dans le cadre de l'architecture J2EE avec le Framework Open swing⁹.

4. Conception :

4.1. Identification des profils utilisateurs :

Deux profils vont agir sur la plateforme :

- Administrateur : c'est la personne chargée de la gestion du module et notamment la gestion des accès.
- Utilisateur : toute personne de l'organisme ayant un accès à des fonctionnalités de l'ERP. Plus spécifiquement les responsables.

4.2. Modélisation UML :

4.2.1. Diagramme des cas d'utilisation :

Dans ce diagramme, nous présentons tous les cas d'utilisation du module de reporting, certains cas représentent des fonctionnalités déjà existantes comme le chargement et l'exécution des rapports.

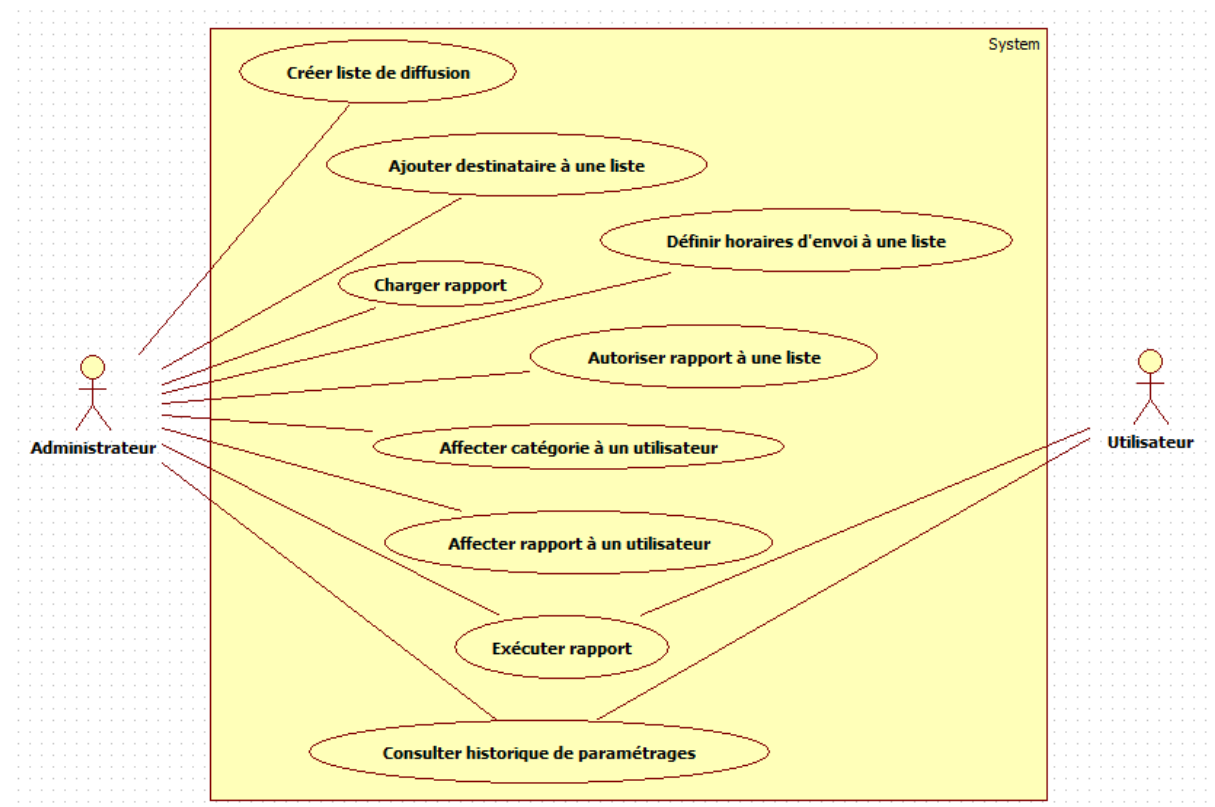


Figure 5.6 : Diagramme des cas d'utilisation du module de reporting

Seuls les cas d'utilisation que nous mettrons en place seront détaillés :

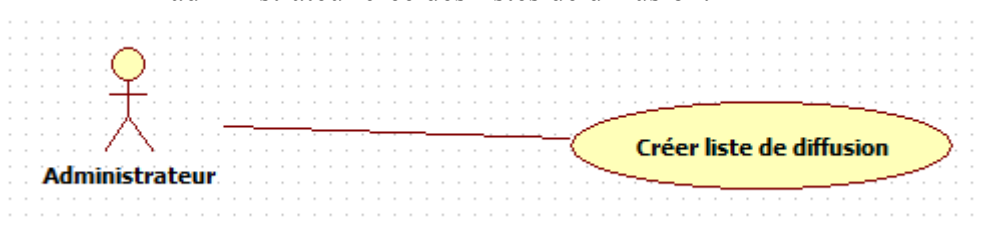
Fiche de description d'un cas d'utilisation	
Titre	Créer liste de diffusion
But	<ul style="list-style-type: none"> • Création des listes de diffusion qui vont contenir des destinataires de rapports.
Version	<ul style="list-style-type: none"> • Version 1.2
Flux principal	<ul style="list-style-type: none"> • L'administrateur crée des listes de diffusion.  <p>The diagram shows a stick figure actor on the left labeled 'Administrateur'. A red line connects it to a yellow oval use case on the right labeled 'Créer liste de diffusion'. The background of the diagram area is a light gray dotted grid.</p>
Acteurs	<ul style="list-style-type: none"> • Administrateur
Description des enchainements	
Scénario nominal	<ol style="list-style-type: none"> 1- L'administrateur accède à l'interface des listes de diffusion 2- L'administrateur saisit le nom de la liste et sa description 3- Le système valide l'ajout.
Pré condition	<ul style="list-style-type: none"> • L'administrateur est identifié et authentifié
Post condition	<ul style="list-style-type: none"> • Liste de diffusion créée prête à y ajouter destinataires.
Fréquence	<ul style="list-style-type: none"> • A la demande

Tableau 5.1 : Fiche de description du cas d'utilisation « Créer liste de diffusion »

Fiche de description d'un cas d'utilisation	
Titre	Ajouter destinataires à une liste
But	<ul style="list-style-type: none"> • Ajout de destinataires à liste de diffusion de rapports

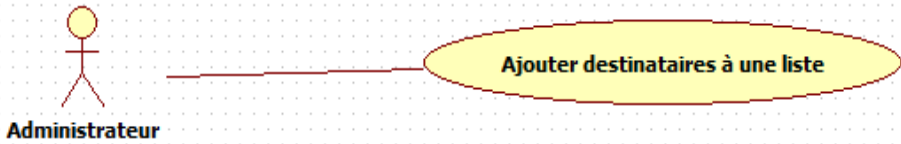
Version	<ul style="list-style-type: none"> Version 1.2
Flux principal	<ul style="list-style-type: none"> L'administrateur ajoute des destinataires à une liste précédemment créée. 
Acteurs	<ul style="list-style-type: none"> Administrateur
Description des enchainements	
Scénario nominal	<ol style="list-style-type: none"> L'administrateur ouvre la rubrique d'ajout de destinataires L'administrateur sélectionne une liste. L'administrateur saisit les informations sur un destinataire Le système valide l'ajout.
Pré condition	<ul style="list-style-type: none"> L'administrateur est identifié et authentifié Liste précédemment créée.
Post condition	<ul style="list-style-type: none"> Liste de diffusion contient des destinataires. Rapports peuvent être autorisés.
Fréquence	<ul style="list-style-type: none"> A la demande

Tableau 5.2 : Fiche de description du cas d'utilisation « **Ajouter destinataires à une liste** »

Fiche de description d'un cas d'utilisation	
Titre	Définir horaires d'envoi à une liste
But	<ul style="list-style-type: none"> Planifier les horaires dans lesquels les destinataires reçoivent les rapports.
Version	<ul style="list-style-type: none"> Version 1.2
Flux principal	<ul style="list-style-type: none"> L'administrateur définit la date de début et la fréquence d'envoi à une liste

	<pre> graph LR Admin[Administrateur] --- UC1(Définir horaire d'envoi) UC1 -.-> <<include>> UC2(Charger rapport) UC1 -.-> <<include>> UC3(Créer liste de diffusion) </pre>
Acteurs	<ul style="list-style-type: none"> Administrateur
Description des enchainements	
Scénario nominal	<ol style="list-style-type: none"> L'administrateur ouvre la rubrique d'autorisations de rapports aux listes L'administrateur sélectionne un rapport L'administrateur saisit le calendrier d'envoi. Le système valide l'ajout.
Pré condition	<ul style="list-style-type: none"> L'administrateur est identifié et authentifié. Liste précédemment créée. Rapport précédemment autorisé à une liste.
Post condition	-
Fréquence	<ul style="list-style-type: none"> A la demande

Tableau 5.3 : Fiche de description du cas d'utilisation « **Définir horaires d'envoi à une liste** »

Fiche de description d'un cas d'utilisation	
Titre	Autoriser un rapport à un utilisateur
But	<ul style="list-style-type: none"> Affecter à chaque utilisateur de la plateforme le rapport qui lui est autorisé.
Version	<ul style="list-style-type: none"> Version 1.2
Flux principal	<ul style="list-style-type: none"> L'administrateur définit les rapports accessibles à un utilisateur dans chaque catégorie.

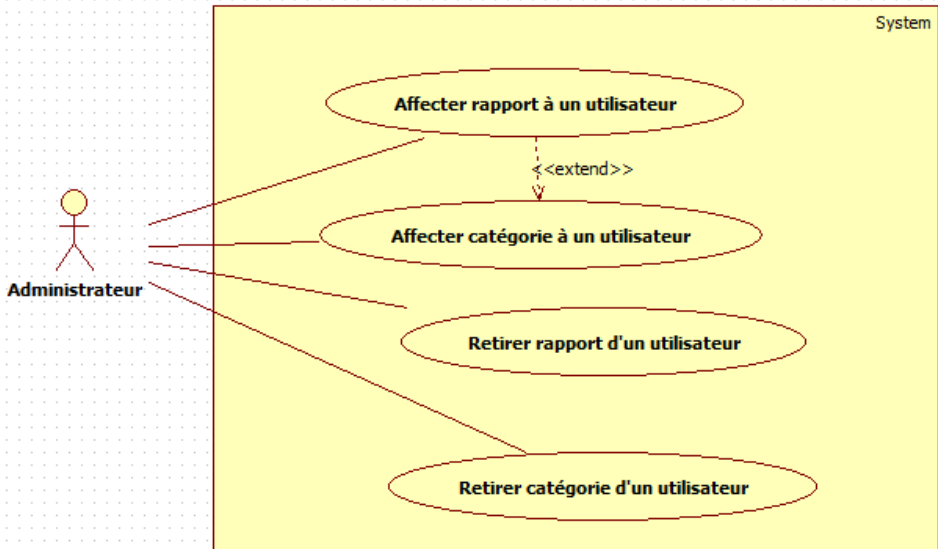
	
Acteurs	<ul style="list-style-type: none"> • Administrateur
Description des enchainements	
Scénario nominal	<ol style="list-style-type: none"> 1- L'administrateur ouvre la page des autorisations d'un rapport. 2- L'administrateur choisit l'onglet des utilisateurs 3- L'administrateur saisit un nouvel utilisateur 4- Le système valide l'ajout.
Pré condition	<ul style="list-style-type: none"> • L'administrateur est identifié et authentifié. • Rapport précédemment chargé
Post condition	<ul style="list-style-type: none"> - L'utilisateur peut exécuter le rapport.
Fréquence	<ul style="list-style-type: none"> • A la demande

Tableau 5.4 : Fiche de description du cas d'utilisation « **Autoriser un rapport à un utilisateur** »

Fiche de description d'un cas d'utilisation	
Titre	Consulter historique de paramètres
But	<ul style="list-style-type: none"> • Avoir une traçabilité sur les valeurs de paramètres avec lesquelles un utilisateur exécute un rapport
Version	<ul style="list-style-type: none"> • Version 1.2
Flux principal	<ul style="list-style-type: none"> • L'administrateur ou l'utilisateur veut consulter l'historique des paramètres.

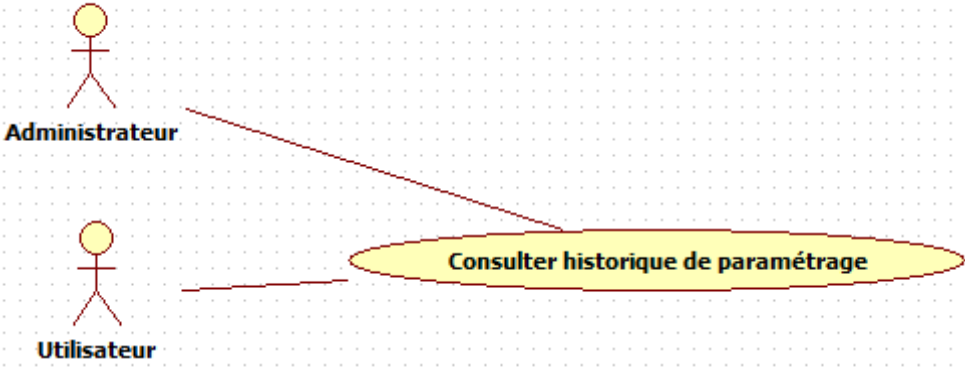
	
Acteurs	<ul style="list-style-type: none"> • Administrateur • Utilisateur
Description des enchainements	
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur ouvre la rubrique d'historiques. 2- Le système affiche les rapports autorisés et paramétrés 3- L'utilisateur sélectionne un rapport. 4- Le système affiche les valeurs de paramètres saisies par date.
Pré condition	<ul style="list-style-type: none"> • L'administrateur/l'utilisateur est identifié et authentifié. • Rapport précédemment chargé • Rapport précédemment autorisé à un utilisateur
Post condition	-
Fréquence	<ul style="list-style-type: none"> • A la demande

Tableau5.5 : Fiche de description du cas d'utilisation « **Consulter historique de paramétrages** »

4.2.2. Diagramme de classes:

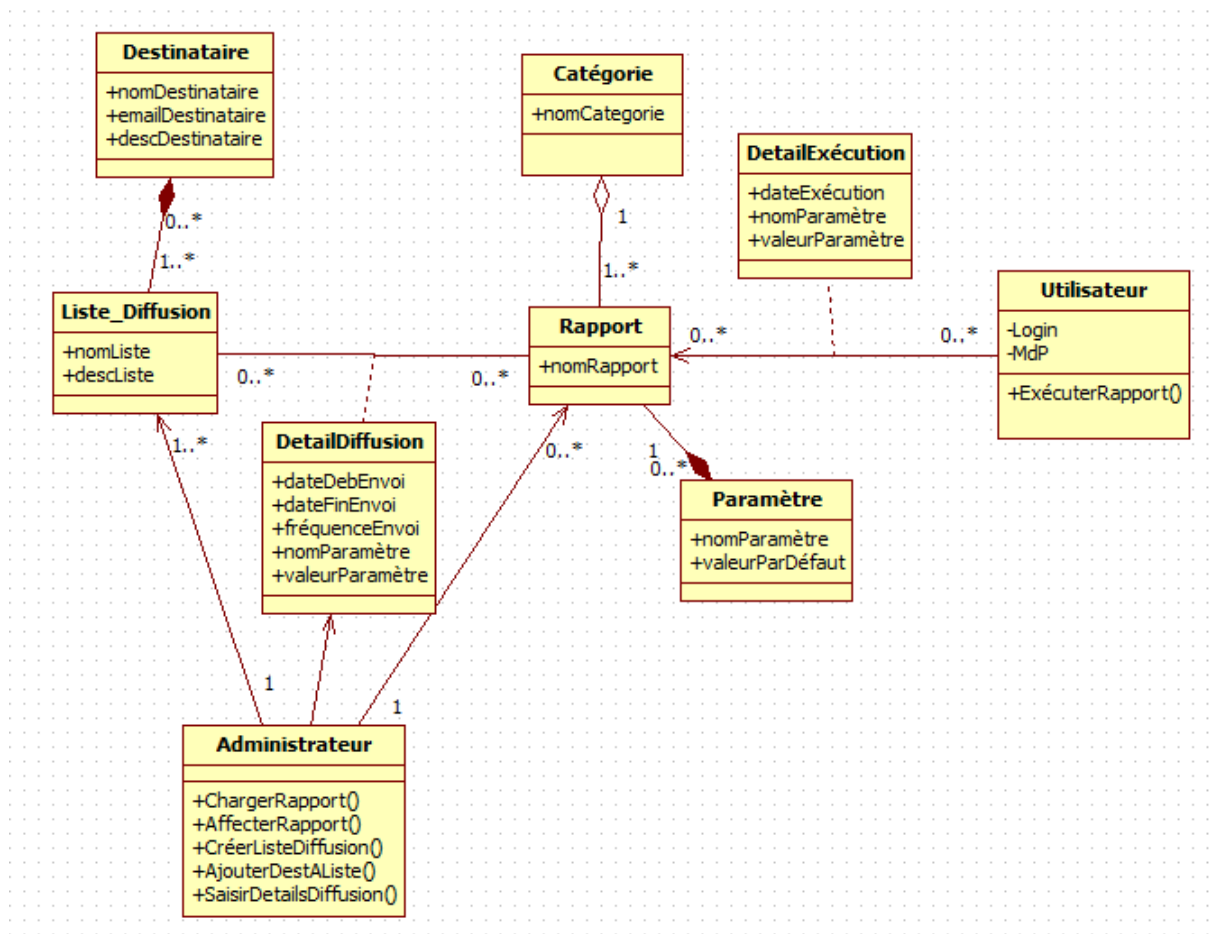


Figure 5.7 : Diagramme de classes.

4.2.3. Diagrammes de séquence :

Dans la notation UML, les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Nous détaillons dans cette partie la description des besoins par la description textuelle des cas d'utilisation.

Nous présentons ci-dessous deux diagrammes de séquence rassemblant une majorité des fonctionnalités

Diagramme de séquence : Autorisation de rapport à utilisateur et exécution.

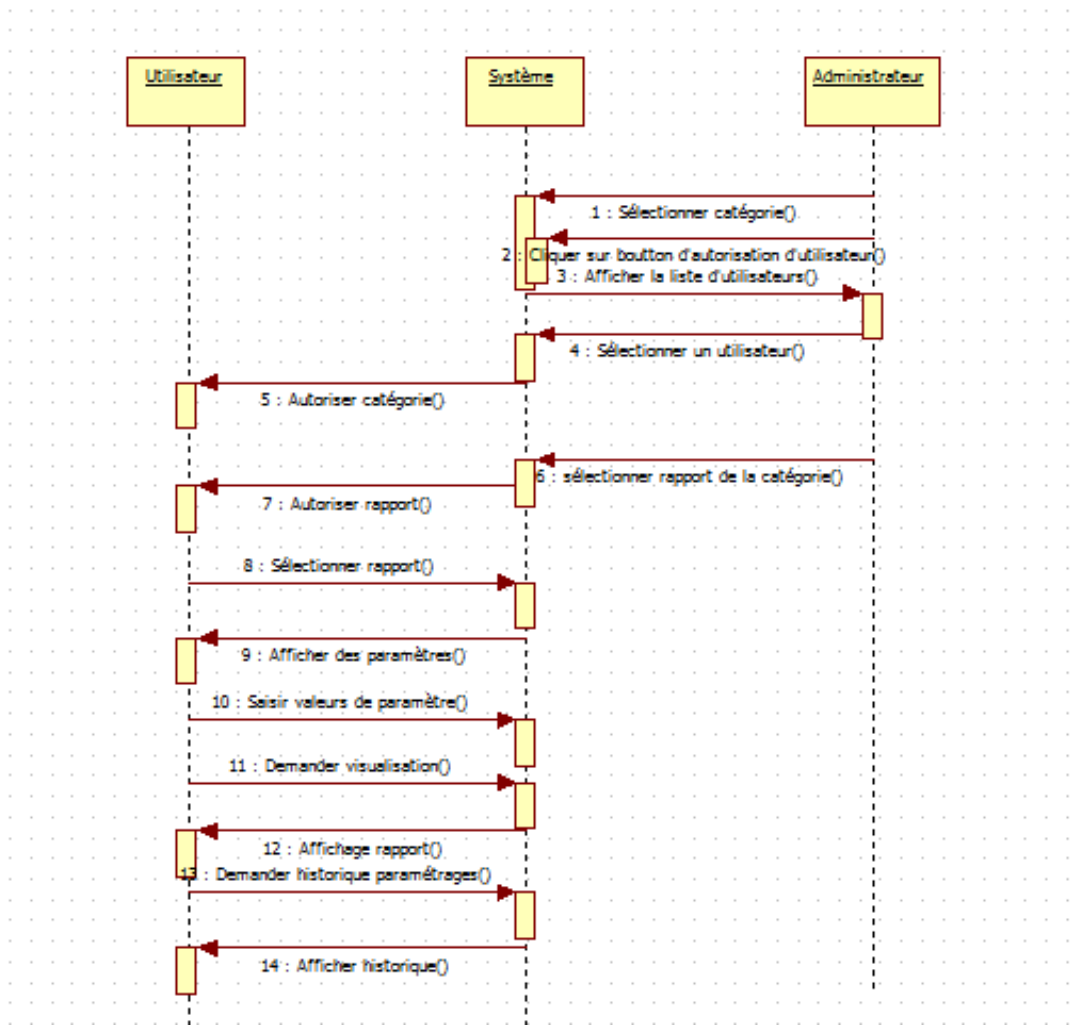


Figure 5.8 : Diagramme de séquence d'autorisation de rapport et exécution

Diagramme de séquence : Autorisation de rapport à une liste et définition d'horaires de diffusion et paramètres.

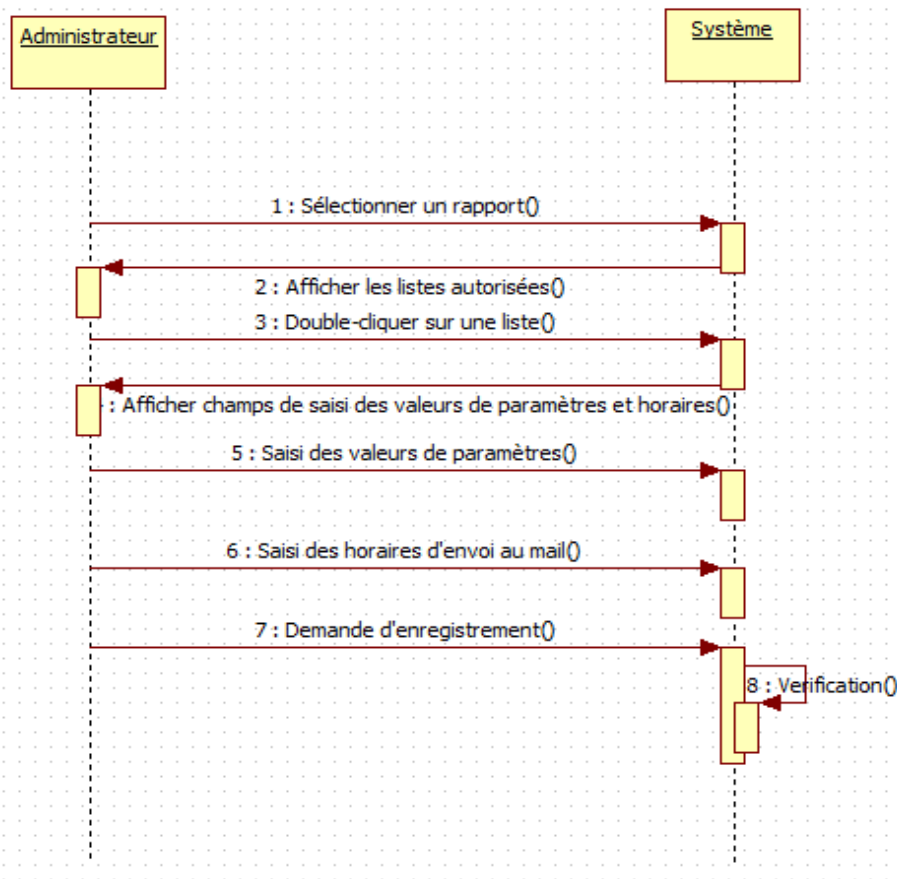


Figure 5.9 : Diagramme de séquence d'autorisation de rapport à une liste, définition d'horaires de diffusion et paramètres.

5. Réalisation :

Dans cette partie, nous allons présenter quelques interfaces graphiques des fonctionnalités que nous avons développées en se basant sur la conception proposée dans les parties précédentes.

Nous tenons à noter que les nouvelles fonctionnalités doivent se conformer à l'architecture technique de la solution et respecter l'exigence d'interfaces graphiques de la solution.

Toutes les nouvelles fonctionnalités sont groupées sous l'onglet "Rapports". Le nom choisi pour désigner le module Reporting par l'entreprise.

5.1.Création de listes de diffusion :

L'administrateur peut ajouter des listes de diffusion selon le besoin tout en saisissant le nom de la liste et un descriptif pour spécifier la raison de diffusion aux destinataires de cette liste.

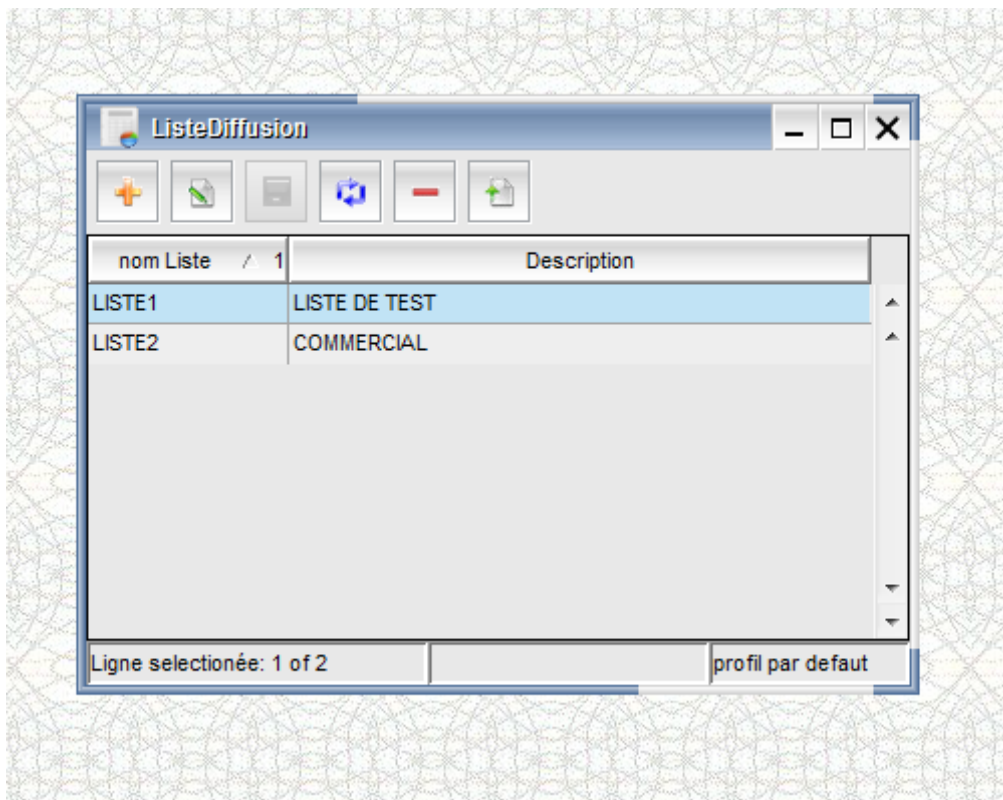


Figure 5.10 : Interface de création de listes de diffusion

5.2.Ajout de destinataires à une liste :

L'administrateur sélectionne d'abord une liste puis y insère des destinataires en saisissant les informations suivantes: nom & prénom, description (raison d'envoi au destinataire), e-mail.

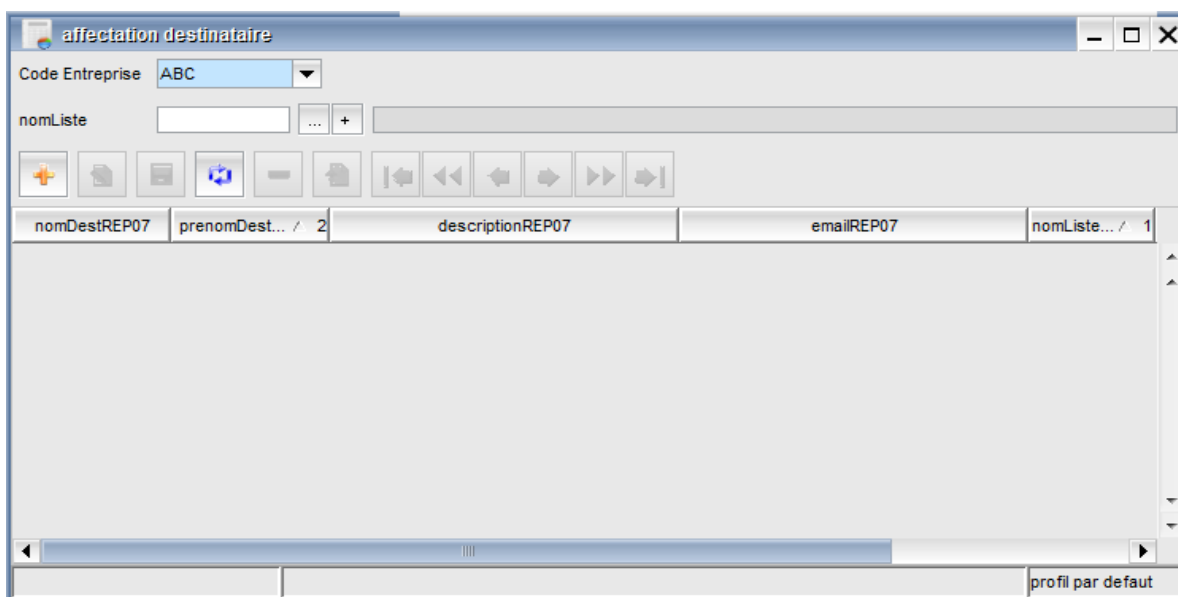


Figure 5.11: Interface principale d'ajout de destinataires aux listes

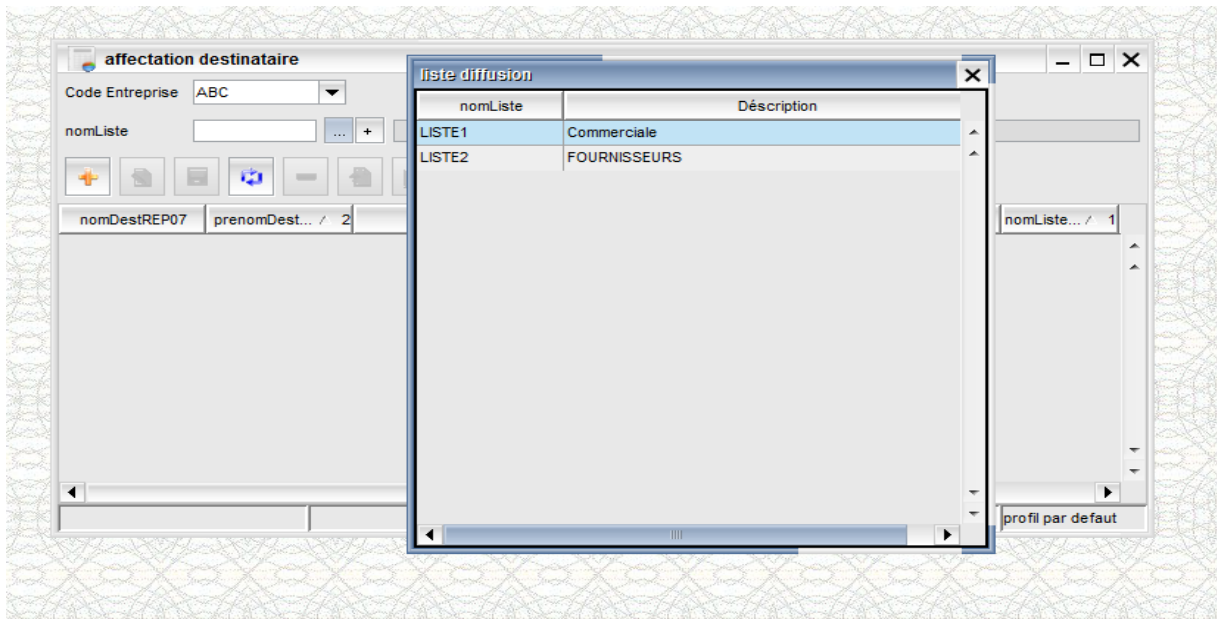


Figure 5.12 : Interface de choix de liste de diffusion pour ajout.

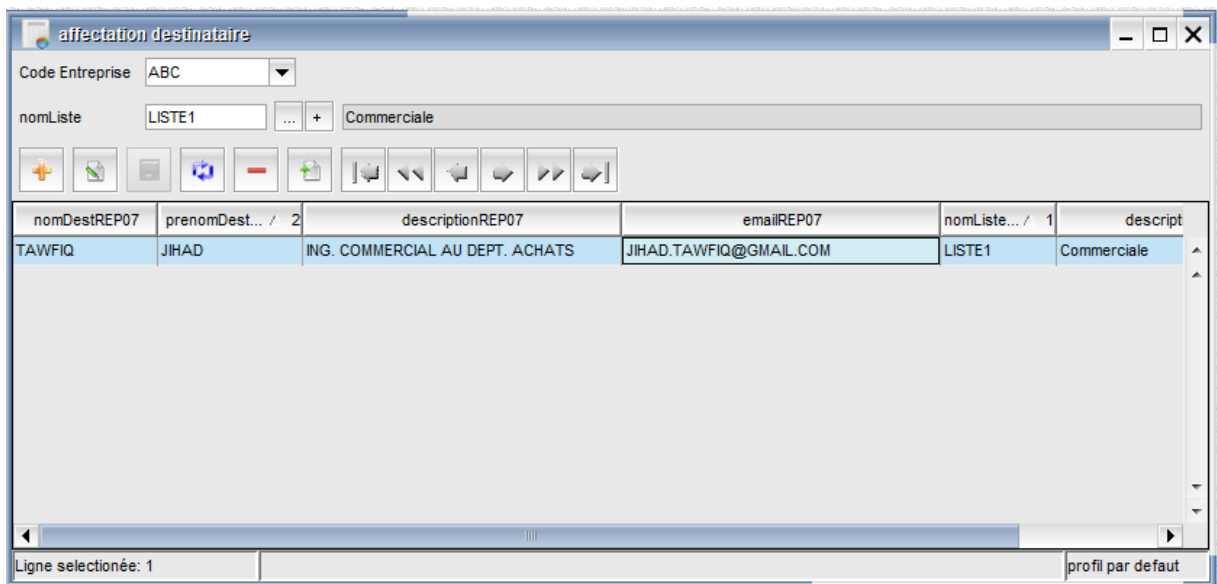


Figure 5.13 : Interface d'affectation d'un destinataire à une liste.

5.3.Saisi de détails de diffusion :

L'administrateur saisit les paramètres d'exécution du rapport et définit le calendrier de son envoi à la liste.

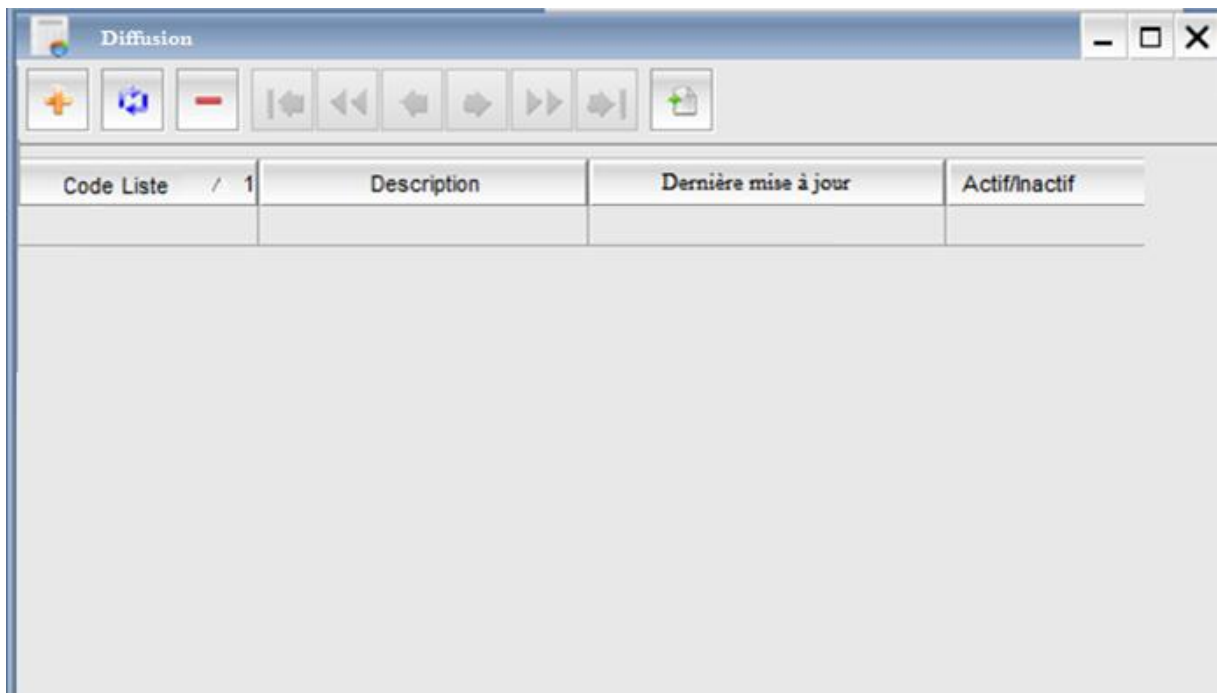


Figure 5.14 : Interface d'affectation d'un destinataire à une liste.

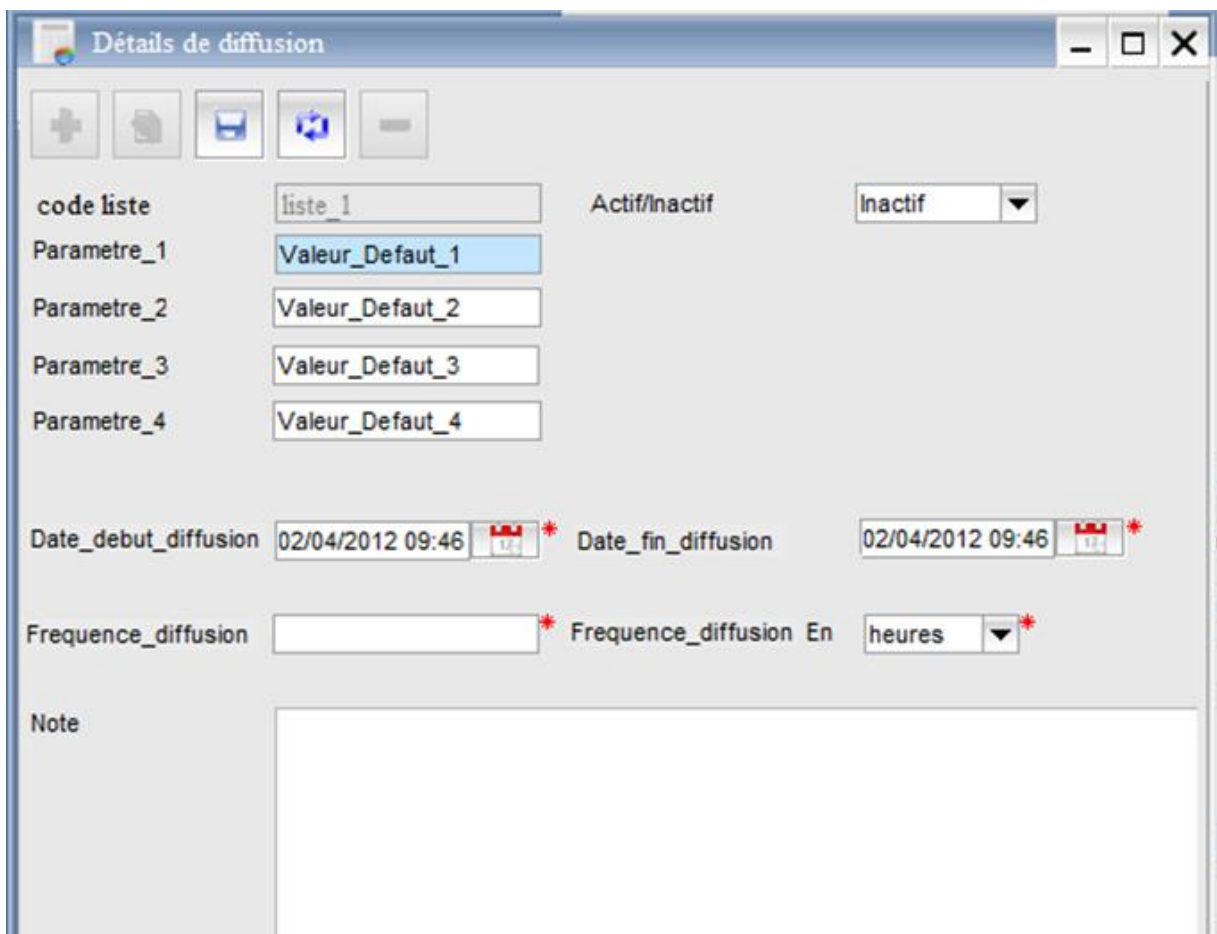


Figure 5.15 : Interface de saisi d'horaire et paramètre d'envoi d'un rapport.

CONCLUSION ET PERSPECTIVES :

Au terme de ce rapport, nous rappelons qu'il s'agit d'un projet de fin d'études que nous avons effectué au sein d'Original Soft System pendant une durée de quatre mois. Notre mission consistait à étudier, concevoir et mettre en place la plateforme BI *Optimis*. Notre projet se découpe en deux sections :

La première suit l'enchaînement de projet décisionnel. Elle est composée des parties suivantes : analyse de l'activité des ventes pour relève des indicateurs clés de performance ; le choix des axes d'analyse et du modèle ; la construction du schéma du DataMart ; l'alimentation de ce dernier à partir de la base de données de l'ERP Optimis via l'ETL Talend Open Studio. Ensuite, la création de rapports d'état à partir des données du DataMart à l'aide de l'outil de conception de rapports iReport.

Dans la deuxième partie du projet, nous étions amenés à enrichir la plateforme par développement du module de reporting. Pour ce faire, nous avons décelé les besoins en matière de consultation et diffusion des rapports. Nous avons fait la conception des fonctionnalités à ajouter conformément aux spécifications UML. Dans la mise en œuvre, le framework Open Swing en suivant le pattern MVC.

Le projet nous a permis de cumuler des connaissances diverses dans les champs de l'informatique décisionnelle, du développement de modules en ERP. Nous avons eu également l'occasion de pratiquer les méthodes agiles pour la gestion du projet et d'acquérir l'expérience du travail collaboratif.

Dans le but de garantir une continuité souple au projet, il est important de proposer certaines perspectives. Nous suggérons les points suivants :

- Upgrade du noyau d'exécution des rapports pour s'aligner avec les dernières versions de JasperReports.
- Utilisation de l'ordonnanceur d'exécution des jobs ETL offert par Talend Open Studio pour son efficacité.
- Ajout des mécanismes d'envoi des rapports à un serveur FTP.
- Ajout de la gestion de versioning des rapports (CVS).

WEBOGRAPHIE:

[1] Dossier sur l'informatique décisionnelle. Consulté en mi février 2012

<http://www.commentcamarche.net/contents/entreprise/business-intelligence.php3>

[2] Article de Wikipédia sur les datamarts. Consulté en mi février 2012

<http://fr.wikipedia.org/wiki/Datamart/>

[3] Article sur l'adoption de la méthode SCRUM. Consulté circa 07/02/2012.

<http://www.cprime.com/community/articles/whentousescrum.html>

[4] Cours sur la modélisation de datamarts. Visité le 25/02/2012

<http://blog.developpez.com/jmalkovich/p8718/modelisation>

[5] Forum dédié au BI Labeldecisionnel. Visité circa 01/03/2012

<http://labeldecisionnel.com>

[6] Tutoriel de JasperReports/iReport. Visité circa 14/03/2012

<http://jpbq.developpez.com/bi/tutoriels/jasperreports/initiation/>

[7] Article sur la méthodologie 2TUP. Consulté circa 05/04/2012

http://www.e-bancel.com/Processus_2TUP.php

[8] Portail de JAllInOne. Visité mi avril 2012.

<http://jallinone.source.net>

http://jaspersoft.com/JasperSoft_Products.htm

BIBLIOGRAPHIE:

[9] Talend Open Studio, Composants, Guide de référence. Edition 2010

[10] iReport 4.2.7 Ultimate User Guide, édition 2010

ANNEXES :

ERP :

ERP est l'acronyme signifiant *Enterprise Resource Planning* (en français : Progiciel de Gestion Intégré (PGI) mais ERP reste le terme le plus courant).

C'est un progiciel qui intègre les principales composantes fonctionnelles de l'entreprise : gestion de production, gestion commerciale, logistique, ressources humaines, comptabilité, contrôle de gestion...

Le principe fondateur des ERP est de construire des applications informatiques pour répondre aux besoins fonctionnels cités ci-dessus tout en ayant recours à la modularité. Ces modules doivent être indépendants entre eux mais partagent une base de données unique en commun.

La décision de mise en place d'ERP est avantageuse pour les considérations suivantes :

- Intégrité et unicité du SI : un ERP permet une logique et une ergonomie unique à travers sa BD, elle aussi unique au sens logique, c'est-à-dire, il peut exister plusieurs bases de données « physiques » mais respectant la même structure. En autres mots : un ERP permet d'éviter les redondances d'informations entre différents systèmes d'informations de l'entreprise.
- Récupération immédiate des données : l'utilisateur peut récupérer des données ou les enregistrer de manière immédiate, les mises à jour s'effectuent en temps réel et se répercutent sur l'ensemble des modules concernés.
- Multilingue et multidevise : les ERP étaient en premier lieu destinés à l'exploitation dans les grandes entreprises et surtout les multinationales, d'où cette adaptation au marché mondial.
- Pas d'interfaçage entre les modules : les traitements sont synchronisés et les processus de gestion sont optimisés. Egalement, les services informatiques de l'entreprise ne s'occupent que de la maintenance évolutive (à savoir, améliorer les règles de gestion etc.) laissant la maintenance corrective au soin de l'éditeur.
- Etc.

Les ERP sont orientés principalement aux grands organismes (surtout les multinationales) du fait des coûts considérables exigés par les éditeurs majeurs : SAP, Oracle, SAGE, Microsoft ...

Cependant, le marché des ERP commence à s'ouvrir peu à peu aux petites et moyennes structures avec l'édition de versions dédiées aux PME mais aussi avec des ERP Open Source qui coûtent beaucoup moins cher en l'absence de coût d'acquisition de licence.

jAllInOne :

jAllInOne est une application orientée services réalisée en java de type ERP/CRM avec un front-end Swing.

L'application jAllInOne a été développée en quelques mois grâce aux outils Open Swing, c'est une plateforme de développement qui permet de créer des RIAs (*Rich Internet Applications*) avec des interfaces graphiques complexes en contenu mais qui ne nécessitent pas un temps important pour le développement.

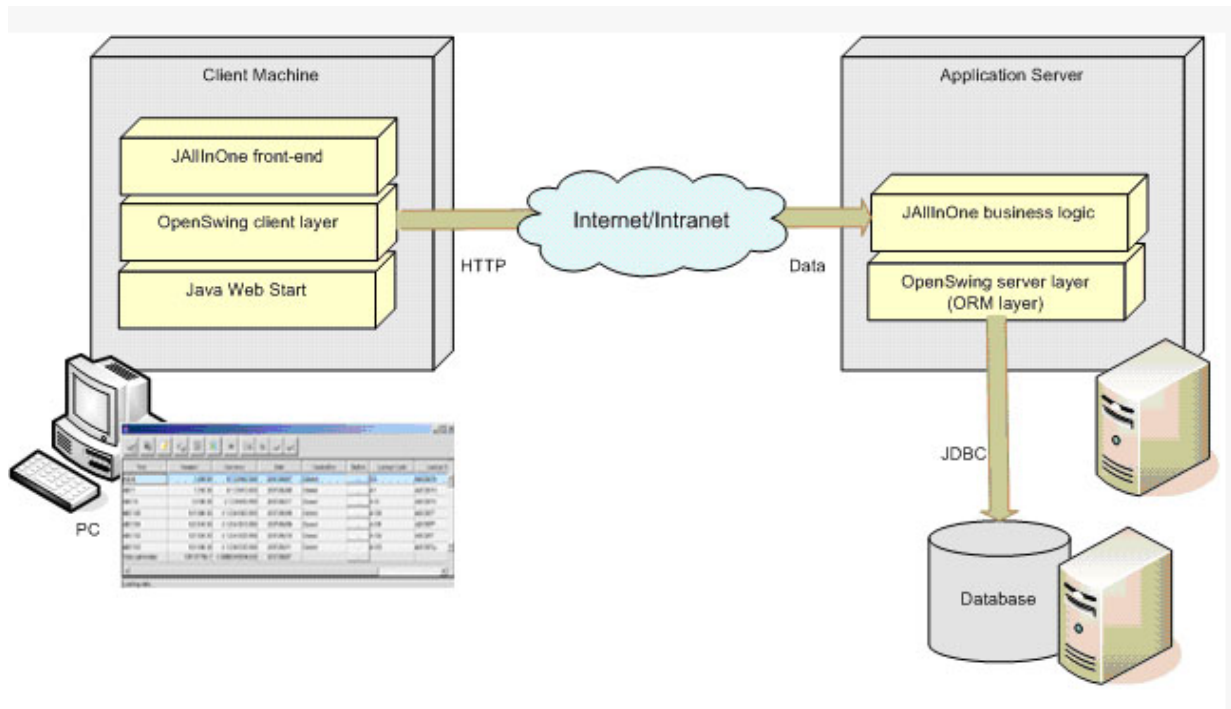
Grâce à la quantité limitée de bits transférés sur internet - vu que seuls les blocs de données circulent sans contenu de présentation -, jAllInOne se caractérise par un bon temps de réponse sur internet.

Le lancement de l'application cliente se fait via Java Web Start utility (inclus dans la distribution JRE). Juste la première fois où jAllInOne est invoqué, l'application cliente doit être téléchargée depuis le serveur : après le premier téléchargement, l'application cliente s'installe dans la machine cliente. Java Web Start synchronise automatiquement l'application cliente lorsque des mises à jour sont disponibles au niveau du serveur.

Le côté serveur peut être lancé en utilisant Java Servlet qui ne nécessite pas un EJB Container, pour cela, jAllinOne côté serveur peut s'exécuter dans n'importe quel web container (comme Tomcat, JBoss, Jetty...) en utilisant n'importe quelle version de java supérieure à 1.4.

jAllInOne couvre la majorité des fonctionnalités dont une entreprise de petite ou moyenne taille peut avoir besoin à savoir : ventes, achats, gestion de production, gestion de maintenance, gestion de personnel, CRM et gestion des activités etc.

Il faut noter aussi que jAllInOne est une application multiutilisateur, multi compagnie, et multilingue (français, anglais, italien, croate, allemand et russe).



Architecture globale de jAllInOne

MVC :

1. Architecture Modèle/Vue/Contrôleur

L'architecture *Modèle/Vue/Contrôleur* (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le *modèle*, la *vue* et le *contrôleur* ayant chacun un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Dans l'architecture MVC, les rôles des trois entités sont les suivants.

- modèle : données (accès et mise à jour)
- vue : interface utilisateur (entrées et sorties)
- contrôleur : gestion des événements et synchronisation

i. Rôle du modèle

Le modèle contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient.

Le modèle offre des méthodes pour mettre à jour ces données (insertion suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ses données. Dans le cas de données importantes, le modèle peut autoriser plusieurs vues partielles des données. Si par exemple le programme manipule une base de données pour les emplois du temps, le modèle peut avoir des méthodes pour avoir, tous les cours d'une salle, tous les cours d'une personnes ou tous les cours d'une groupe de Td.

ii. Rôle de la vue

La vue fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir tous les actions de l'utilisateur (clic de souris, sélection d'une entrées, boutons, ...). Ses différents événements sont envoyés au contrôleur.

La vue peut aussi donner plusieurs vues, partielles ou non, des mêmes données. Par exemple, l'application de conversion de bases a un entier comme unique donnée. Ce même entier est affiché de multiples façons (en texte dans différentes bases, bit par bit avec des boutons à cocher, avec des curseurs). La vue peut aussi offrir la possibilité à l'utilisateur de changer de vue.

iii. Rôle du contrôleur

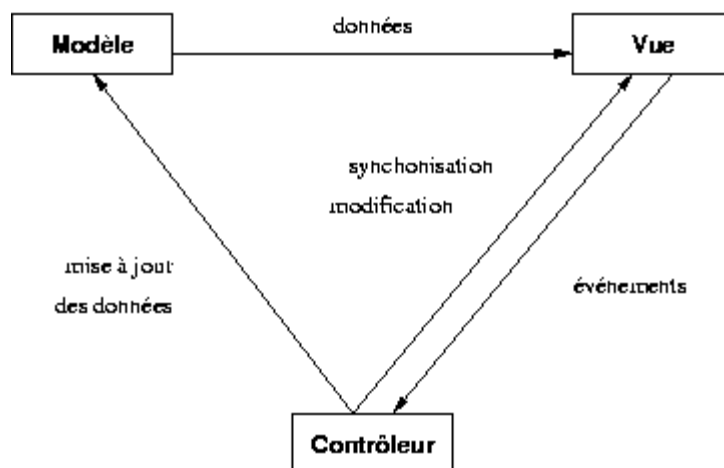
Le contrôleur est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour. Certains événements de l'utilisateur ne concerne pas les données mais la vue. Dans ce cas, le contrôleur demande à la vue de se modifier.

Dans le cas d'une base de données des emplois du temps. Une action de l'utilisateur peut être l'entrée (saisie) d'un nouveau cours. Le contrôleur ajoute ce cours au modèle et demande sa prise en compte par la vue. Une action de l'utilisateur peut aussi être de sélectionner une nouvelle personne pour visualiser tous ses cours. Ceci ne modifie pas la base des cours mais nécessite simplement que la vue s'adapte et offre à l'utilisateur une vision des cours de cette personne.

Le contrôleur est souvent scindé en plusieurs parties dont chacune reçoit les événements d'une partie des composants. En effet si un même objet reçoit les événements de tous les composants, il lui faut déterminer quelle est l'origine de chaque événement. Ce tri des événements peut s'avérer fastidieuse et peut conduire à un code pas très élégant (un énorme `switch`). C'est pour éviter ce problème que le contrôleur est réparti en plusieurs objets.

iv. Interactions

Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure suivante.



Interactions entre le modèle, la vue et le contrôleur

2. Utilisation native en Swing

La plupart des composants Swing (sauf les conteneurs) utilisent une classe spécifique pour contenir leurs données.

- Les boutons (`JButton`) utilisent un objet de classe `ButtonModel` pour les données.
- Les curseurs (`JSlider`) utilisent un objet de classe `BoundedRangeModel` pour les données.
- Les listes (`JList`) utilisent un objet de classe `ListModel` pour les données et un objet de classe `ListSelectionModel` pour gérer les sélections.
- Les arbres (`JTree`) utilisent un objet de classe `TreeModel` pour les données et un objet de classe `TreeSelectionModel` pour gérer les sélections.
- Les tables (`JTable`) utilisent des objets de classe `TableModel` et `TableColumnModel` pour les données et un objet de classe `ListSelectionModel` pour gérer les sélections.
- Les composants de texte (`JTextComponent` et les classes dérivées `JEditorPane`, `JTextArea` et `JTextField`) utilisent un objet de classe `Document` pour gérer le texte.

Talend Open Studio :



Talend Open Studio est développé par Talend, une société française (www.talend.com). La première version de « Talend Open Studio » a vu le jour au 2ème semestre 2006, et la version actuelle est la 5.0.2

Talend Open Studio est un ETL (*Extract – Transform – Load*) du type « générateur de code ». Lors de chaque opération d'intégration de donnée, un code spécifique est généré, soit en Java ou en Perl selon le choix de l'utilisateur. Les données traitées et les traitements effectués sont donc intimement liés.

Les traitements sur TOS se réalisent via une interface graphique, le « Job Designer » (basée sur Eclipse RCP) qui permet la création des processus de manipulation de données.

De nombreux types d'étapes sont disponibles pour se connecter aux principaux SGBD (Oracle, DB2, MS SQL Server, PostgreSQL, MySQL,...) ainsi que pour traiter tous les types de fichiers plats (CSV, Excel, XML), aussi bien en lecture qu'en écriture. Talend facilite la construction des requêtes dans les bases de données en détectant le schéma et les relations entre tables. Un référentiel permet de stocker les métadonnées afin de pouvoir les exploiter dans différents jobs.

Le Job Designer intègre une « Component Library »: une palette graphique de composants et connecteurs.

Les processus d'intégration sont construits simplement en déposant des composants et connecteurs sur le diagramme, en dessinant leurs connexions et relations, et en modifiant leurs propriétés.

La plupart de ces propriétés peut être issue des métadonnées déjà définies.

La Component Library inclut plus de 80 composants et connecteurs, fournissant des fonctions basiques telles que des associations, transformations, agrégations et recherches; des fonctions spécialisées comme le filtrage de données, le multiplexage de données...

Cette librairie supporte tous les principaux SGBDR, formats de fichiers, annuaires LDAP...

La Component Library peut facilement être complétée en utilisant des langages standards tels que Perl, Java ou SQL.

La conception très visuelle des "jobs" permet de présenter des statistiques d'exécution en temps réel ou encore de tracer les données transitant ligne à ligne dans les composants de la chaîne de traitement.

Quand un job d'intégration est lancé via le Job Designer (en mode graphique), il est possible d'afficher les statistiques de traitement en temps réel, montrant le nombre de lignes traitées et rejetées, ainsi que la vitesse d'exécution (lignes par secondes). On peut ainsi repérer immédiatement les goulots d'étranglement.

Il est aussi possible d'activer un mode de traçage, qui affiche pour chaque ligne le comportement adopté et montre le résultat des transformations. Les fonctionnalités de débogage traditionnelles sont évidemment disponibles.

L'enrichissement des traitements par ajout de code spécifique :

La totalité du code généré par Talend Open Studio, quel que soit le langage cible, est toujours visible et accessible depuis l'environnement de conception.

On peut bien sûr implémenter des spécificités « métiers » propres aux données traitées, ceci en ajoutant de nouvelles « routines ».

iReport/JasperReport :



JasperReports est un moteur de rapport développé par la société JasperSoft et distribué sous une licence open source. iReport est l'éditeur de rapport de JasperSoft. Ces outils, fortement demandés dans les métiers de reporting, existent au marché depuis 2001.

Dans notre projet, nous utilisons JasperReports et iReport dans leur version 4.5.0, sortie en décembre 2011.

Les rapports générés sont des fichiers XML et peuvent également être créés et modifiés manuellement.

1. Générateur de rapport :

Le moteur JasperReports permet la génération de rapports au format PDF, HTML, XML, CSV, RTF, XLS et TXT. Il utilise JFreeChart pour générer les graphiques et peut être intégré dans toute application développée avec le langage Java.

Il supporte, en plus des bases de données classiques, les serveurs d'analyse multidimensionnelle ce qui permet d'exploiter les possibilités du serveur Mondrian directement dans un rapport JasperReports.

2. Conception des rapports :

La conception des états se fait soit par description XML soit par outil graphique (iReport). C'est à cette dernière option qu'on a souvent recours grâce à son ergonomie.

Les rapports sont décomposés en bandes dans lesquelles les éléments graphiques sont déposés. Chaque bande a un comportement spécifique et apparaît une ou plusieurs fois.

Un rapport exécute une itération sur un jeu de données principal. Certaines bandes sont affichées avant ou après l'ensemble des données de l'état, d'autres le sont une fois pour chaque élément du jeu de données.

Les différentes bandes disponibles sont :

- titre du rapport, affiché au début de la première page.
- entête de la page, affichée au début de chaque page.
- en-tête des colonnes, affichées avant les données.
- détails, répétés pour chaque élément des données.

- fin des colonnes, affichée après l'ensemble des données.
- pied de page, affichée en bas de chaque page.
- dernière page, affichée dans la dernière page.
- page de résumé, conclut le rapport.

Pour créer des rapports plus riches, il est possible d'utiliser des jeux de données secondaires dans certains éléments, comme les graphiques et les tableaux, ou d'insérer des états secondaires.

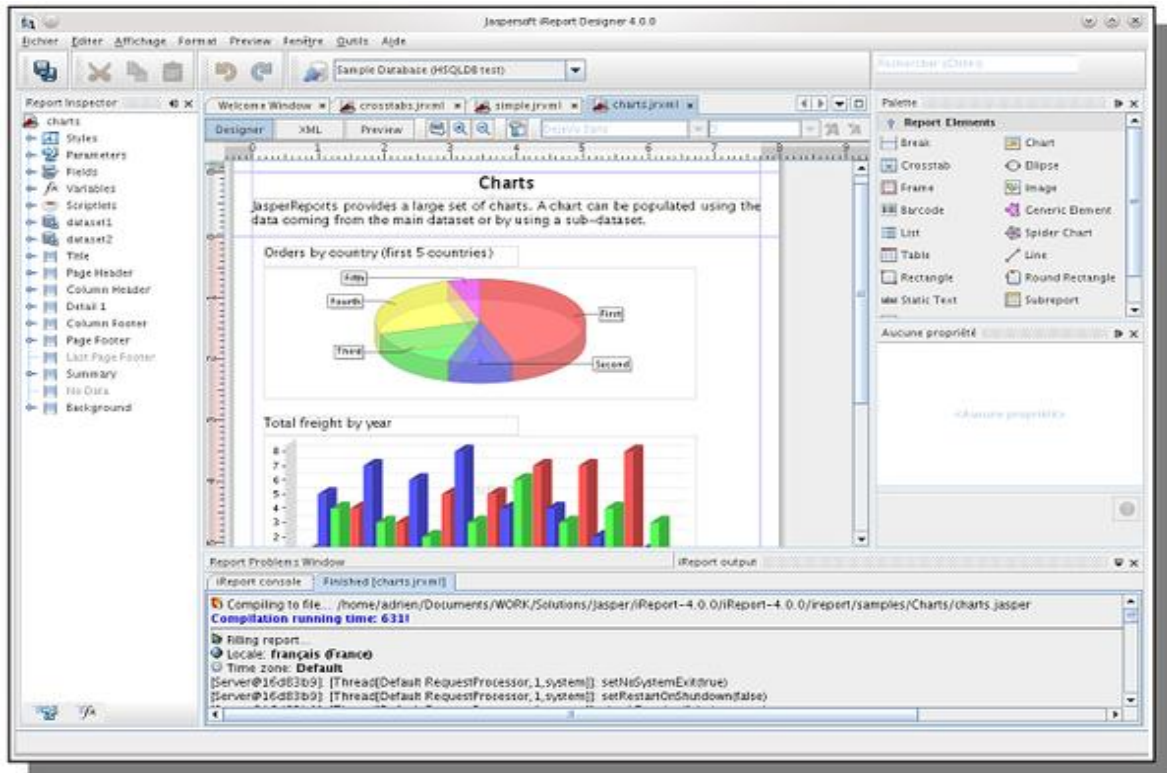
3. Interface graphique : iReport :

Plusieurs éditeurs graphiques étaient associés à JasperReports comme Jasper Assistant et JasperPal, mais depuis que JasperSoft a émis iReport, ce dernier a dominé tous les autres outils.

iReport est donc l'outil de conception d'états officiel de JasperReports et se présente sous la forme d'une application Java dédiée.

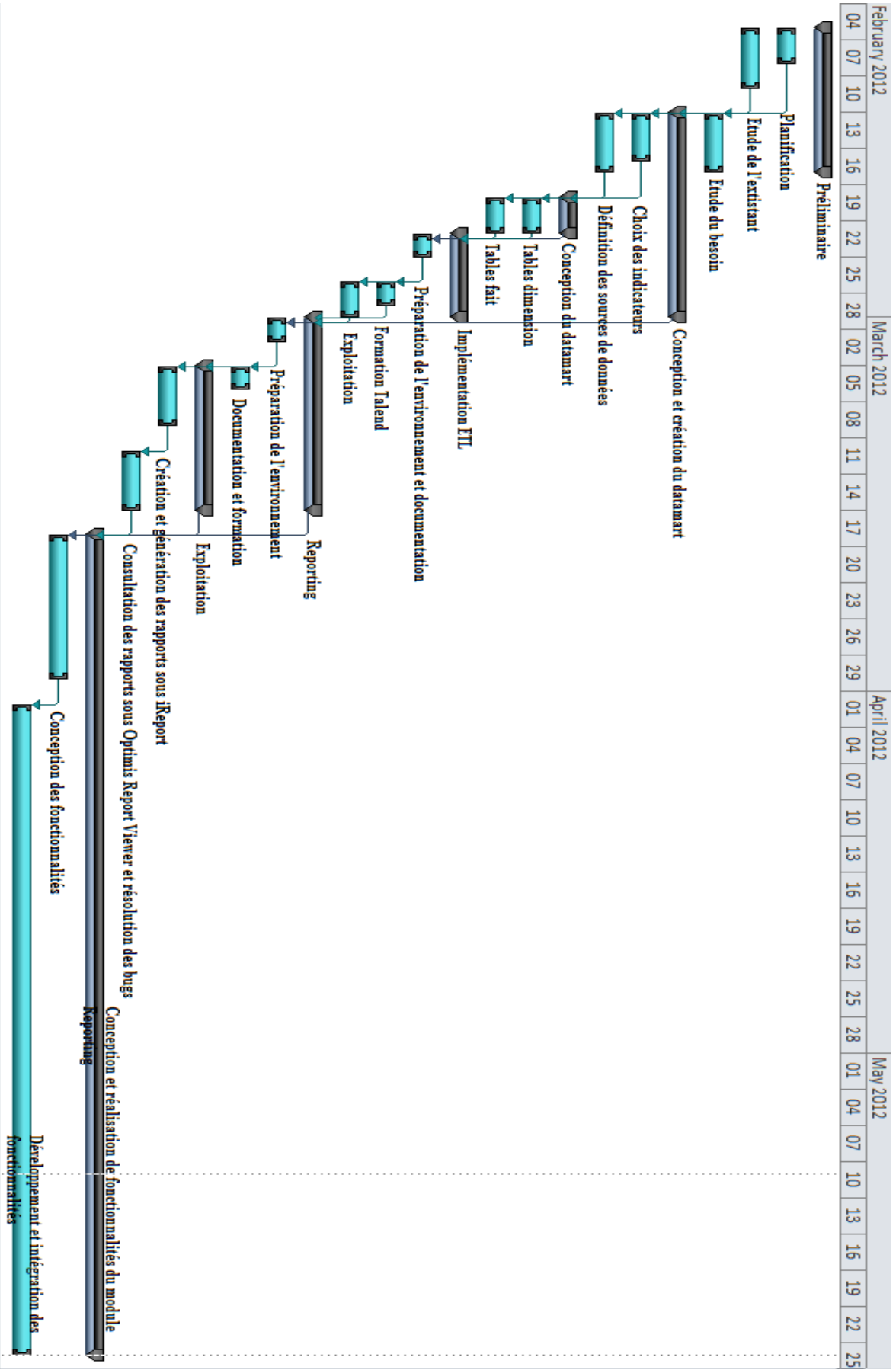
Il supporte la quasi-totalité des fonctionnalités de JasperReports (tableaux, tableaux croisés, graphiques) et dispose également d'une extension dédiée à l'administration de la plateforme décisionnelle de JasperSoft.

L'interface et le mode de fonctionnement des rapports JasperReports destinent principalement iReport à des spécialistes.



iReport 4.0
Interface d'iReport

Diagramme de GANT :



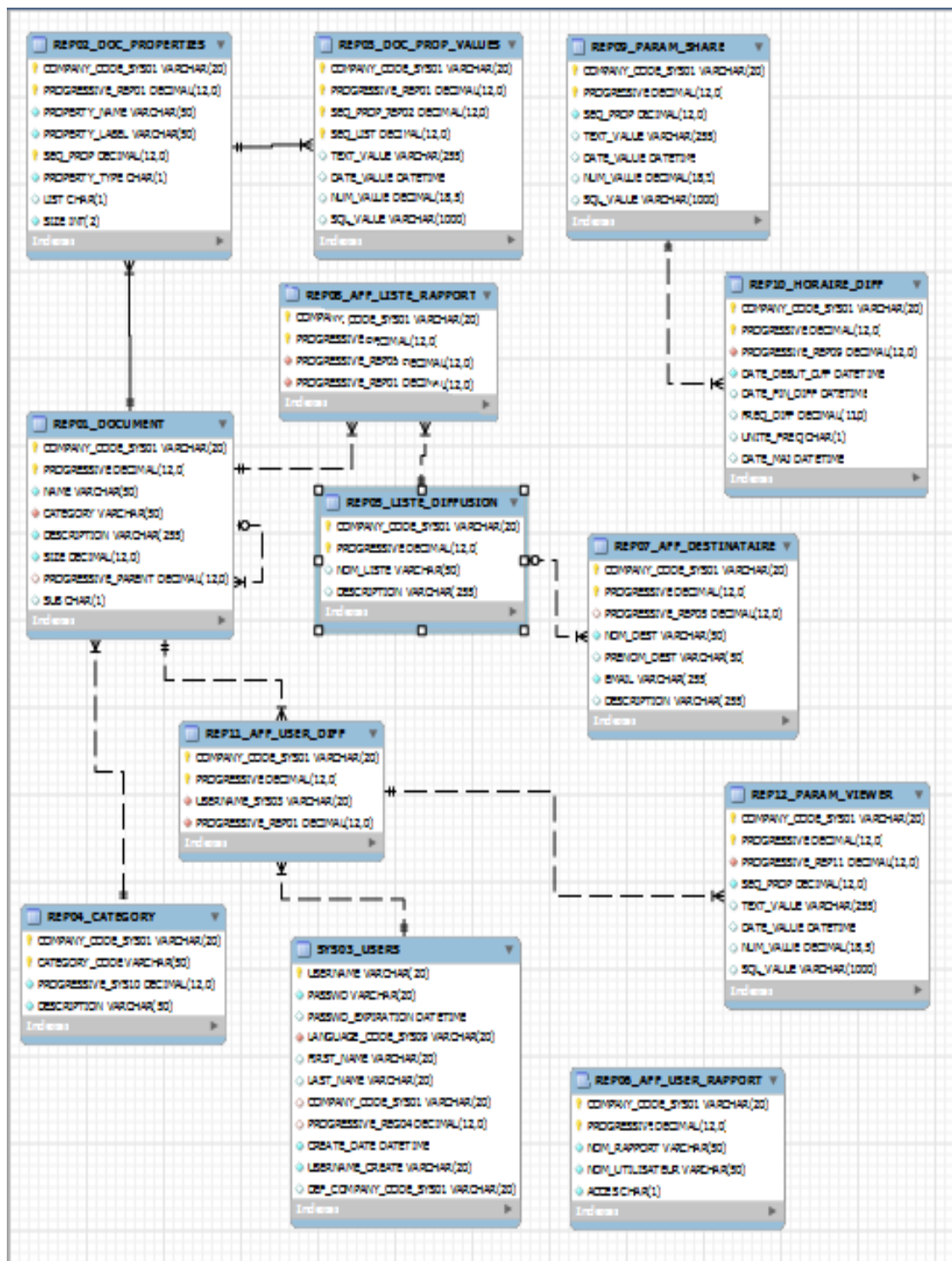
MySQL Workbench :



MySQL Workbench est un outil visuel unifié pour architectes, développeurs et administrateurs de base de données. MySQL Workbench fournit la modélisation des données, le développement SQL et des outils d'administration complets pour la configuration des serveurs, l'administration des utilisateurs et davantage. MySQL Workbench est disponible sous Windows, Linux et Mac OS.

MySQL Workbench permet à un administrateur de base de données, un développeur ou un architecte de données de concevoir, de modéliser, de générer et de gérer visuellement des bases de données. Il comprend tous les éléments nécessaires à un modélisateur de données pour créer des modèles entité-relation complexes et procéder à une pro-ingénierie ou à une rétro-ingénierie. Il offre également des fonctionnalités clés qui permettent d'accomplir les tâches délicates de gestion des modifications et de documentation, qui exigent habituellement beaucoup de temps et d'efforts.

Modèle relationnel de données :



Les tables dont la nomenclature se précède par « REP » sont réservées au reporting .

Les 4 premières tables REP01_DOCUMENT, REP02_DOC_PROPERTIES, REP03_DOC_PROP_VALUE, REP04_CATEGORY existaient précédemment dans la base de données. Elles servent au stockage des informations relatives uniquement aux rapports, leurs paramètres et valeurs par défaut et catégories.

Le tableau suivant présente les tables que nous avons ajoutées et l'utilité de chacune.

Table	Utilité
REP05_LISTE_DIFFUSION	Informations relatives aux listes de diffusion
REP06_AFF_USER_RAPPORT	Accès des utilisateurs aux rapports
REP07_AFF_DESTINATAIRE	Informations relatives aux destinataires ajoutés aux listes.
REP08_AFF_LISTE_RAPPORT	Enregistrement des autorisations des rapports à des listes de diffusion.
REP09_PARAM_SHARE	Enregistrement des valeurs de paramètres saisies par l'administration avant diffusion d'un rapport
REP10_HORAIRE_DIFF	Enregistrement des dates de début et fin et fréquence d'envoi de rapports aux destinataires d'une liste
REP12_PARAM_VIEWER	Historique des saisies de paramètres faites par les utilisateurs pour exécution des rapports.

Open Swing :

Open Swing est le Framework open source de composants graphiques basés sur Swing et une plateforme pour développement d'applications client/serveur à deux ou trois tiers ; Il permet de développer des RIAs (*Rich Internet Applications*) basées sur le protocole http et un front-end Swing. La création des interfaces graphiques utilise le designer d'interfaces de l'IDE java adopté : Eclipse, NetBeans, JDeveloper ou JBuilder et même des IDE non-java (JB, Delphi, Visual Studio .NET). Plusieurs technologies et mécanismes sont utilisés dans Open Swing comme POJO, MVC, *data binding*...

Il peut être facilement intégré dans plusieurs couches serveur comme Spring, Hibernate, iBatis, JPA.