

***PROJET DE FIN D’ETUDES***

***Pour l’obtention du Diplôme National d’Ingénieur en Informatique***

***Intégration dans une équipe Projet IT de Gestion des obligations sécurisées***

Société d’accueil :

***CBS***

***Tuteur enseignant :***

***Année universitaire : 2014-2015***

1

***Covered Bonds System***

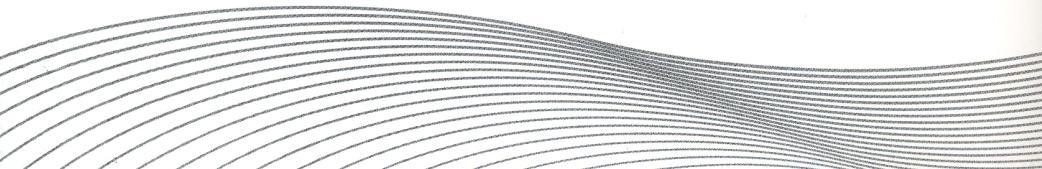
Entreprise d’accueil

******

***Réalisé par : Mme. Marwa BEN ABDA***

***Juillet 2015-Décembre 2015***

***Maîtres de stage :***

*** Mr. Damien DENIZART Mr. Wael ANIBA***

***Mr. Amen Allah BHAR***

***Année universitaire: 2015-2016***

***………. Mr. WIBA***

***Dédicaces***

*Je dédie ce travail :*

*A mon père, mon premier encadrant, depuis ma naissance ;*

*A ma très chère mère : qu’elle trouve ici l’hommage de ma gratitude qui, si grande qu’elle puisse être, ne sera à la hauteur de ses sacrifices et ses prières pour moi ;*

*A mon aimable mari : reçoit à travers ce travail tout mon amour, mon respect et ma profonde reconnaissance, merci d’être toujours à mes cotés, par ta présence et ta tendresse avouée, pour donner du goût et du sens à notre vie de famille.*

*A mes sœurs et frère : souraya, sarra, sofiene à qui je souhaite beaucoup de réussite et de bonheur ;*

*A mes beaux-parents : puisse Dieu, le tout puissant vous préserver du mal, vous combler de santé, de bonheur et vous procurer une longue vie ;*

*A ma belle sœur et mon beau frère :amani et jedidi à qui je souhaite aussi une vie pleine de santé et de prospérité ;*

*A tous mes amies et amis qui me sont chers, à tous ceux, que j’aime et qui m’aiment : qu’ils trouvent ici l’expression de mes sentiments les plus dévoués et mes vœux les plus sincères ;*

*Que dieu vous procure sagesse et bonheur.*

Marwa BEN ABDA

***Remerciements***

*Premièrement je tiens à remercier Dieu source de toute connaissance.*

*Au terme de ce travail, je voudrais adresser mes remerciements les plus sincères en premier lieu à Mme ZALEGH Fella et Mr. MSELLEK Youssef, managers de l’équipe en charge de l’application CBS, méritants tout le respect pour leurs encouragements, et de m’avoir accueillie dans la SOCIETE GENERALE et également pour leur aide qu’ils m’ont offerte tout au long du stage.*

*Un immense merci à Mr. DENIZART Damien, mon chef de projet, pour son suivi et son soutien tout au long de mon stage, aussi pour m’avoir accueillie au sein de son équipe et pour tout le temps qu’il m’a consacré.*

*Mes remerciements les plus cordiaux s’adressent à mon encadreur BHAR Amen Allah, ingénieur développeur à la SOCIETE GENERALE, pour sa disponibilité, son aide, ses conseils précieux, ses critiques constructives, ses explications et suggestions pertinentes ainsi que pour ses qualités humaines et morales que j’ai toujours appréciées.*

*Je remercie également mon superviseur Mr. ANIBA Wael pour ses encouragements et ses conseils qui m'ont été très bénéfiques, sans oublier sa participation au cheminement de ce rapport.*

*Je ne manquerais pas l’occasion de remercier tous les membres de l’équipe ITEC/FCC/OSD et plus particulièrement les membres de l’équipe projet CBS à Bangalore, Mr. GUPTA Deepak*

*, Mr .POSHAM Sathyanarayana, Mr. HANUMANTHAPPA*  *Santhoshkumara et Mr. KESSENTINI Habib qui travaille à Talan Tunis.*

*Signature du responsable à l'organisme d'accueil*

*SOCIETE GENERALE*

*Signature de l’encadrant pédagogique à ESPRIT Mr Wael ANIBA*

Sommaire

[Introduction générale 11](#_Toc435628730)

[Chapitre 1 : Contexte 12](#_Toc435628731)

[Introduction 13](#_Toc435628732)

[1. Présentation de l’organisme d’accueil 13](#_Toc435628733)

[1.1. La SOCIETE GENERALE 13](#_Toc435628734)

[1.2. GBIS 14](#_Toc435628735)

[1.3. Le service 14](#_Toc435628736)

[1.3.1. ITEC/FCC/OSD 14](#_Toc435628737)

[1.3.2. Interactions avec les autres équipes 16](#_Toc435628738)

[2. La méthodologie adaptée 17](#_Toc435628739)

[2.1. L’agilité chez ITEC 17](#_Toc435628740)

[2.2. Organisation de l’équipe CBS 19](#_Toc435628741)

[2.3. Les impacts dans CBS 22](#_Toc435628742)

[3. Présentation du projet 24](#_Toc435628743)

[Problématique 25](#_Toc435628744)

[4. La solution proposée 25](#_Toc435628745)

[5. Organisation de ma mission 26](#_Toc435628746)

[5.1. Intégration dans l’équipe 26](#_Toc435628747)

[5.2. Suivi de ma mission 26](#_Toc435628748)

[Conclusion 27](#_Toc435628749)

[Chapitre 2 : Analyse et Spécification des besoins 26](#_Toc435628750)

[Introduction 27](#_Toc435628751)

[1. Etude préliminaire 27](#_Toc435628752)

[1.1. Recueil des besoins fonctionnels 27](#_Toc435628753)

[1.2. Identification des acteurs 28](#_Toc435628754)

[1.3. Besoins non fonctionnels 30](#_Toc435628755)

[2. Branche fonctionnelle 31](#_Toc435628756)

[2.1. Identification des cas d’utilisation 31](#_Toc435628757)

[2.2. Description des cas d’utilisation 31](#_Toc435628758)

[2.2.1. Cas d’utilisation relatifs à l’administrateur 31](#_Toc435628759)

[2.2.2. Cas d’utilisation relatifs aux Utilisateurs 32](#_Toc435628760)

[2.2.3. Cas d’utilisation relatifs à la gestion des instruments 34](#_Toc435628761)

[2.3. Diagramme de séquence système 35](#_Toc435628762)

[2.3.1. Diagramme de séquence système de l’authentification 35](#_Toc435628763)

[2.3.2. Diagramme de séquence système de gestion d’un instrument (produit financier) 36](#_Toc435628764)

[3. Spécification technique 38](#_Toc435628765)

[3.1. Contraintes et choix techniques 38](#_Toc435628766)

[3.2. Modélisation et architecture de l’application CBD 39](#_Toc435628767)

[3.2.1. Architecture physique 39](#_Toc435628768)

[3.2.2. Architecture logique 40](#_Toc435628769)

[Conclusion 42](#_Toc435628770)

[Chapitre 3 : Conception 42](#_Toc435628771)

[Introduction 43](#_Toc435628772)

[1. Conception préliminaire 43](#_Toc435628773)

[1.1. Vue Statique 43](#_Toc435628774)

[1.1.1. Diagramme de packages 43](#_Toc435628775)

[1.1.2. Diagramme de Classes 46](#_Toc435628776)

[1.2. Vue Dynamique 48](#_Toc435628777)

[1.2.1. Diagramme de séquence objets 48](#_Toc435628778)

[1.2.1.1. Diagramme de séquence objet « authentification » 48](#_Toc435628779)

[1.2.1.2. Diagramme de séquence objet d’affichage et génération d’un rapport de type BO 49](#_Toc435628780)

[1.2.1.3. Diagramme de séquence objet de recherche d’un instrument 51](#_Toc435628781)

[1.3. Diagramme d'activité 53](#_Toc435628782)

[1.3.1. Diagramme d'activité modification d’un rapport 54](#_Toc435628783)

[1.3.2. Diagramme d'activité téléchargement d’un fichier Excel relatif aux prêts et échéanciers 55](#_Toc435628784)

[2. Conception détaillée 56](#_Toc435628785)

[2.1. Design Pattern 56](#_Toc435628786)

[Conclusion 53](#_Toc435628787)

[Chapitre 4 : Réalisation 53](#_Toc435628788)

[Introduction 54](#_Toc435628789)

[1. Environnement matériel 54](#_Toc435628790)

[2. Environnement logiciel 54](#_Toc435628791)

[2.1. Outil de conception 56](#_Toc435628792)

[3. Aperçu sur le travail réalisé 56](#_Toc435628793)

[3.1. Authentification 56](#_Toc435628794)

[3.2. Rôle administrateur 57](#_Toc435628795)

[3.3. Rôle utilisateur 60](#_Toc435628796)

[Conclusion 71](#_Toc435628797)

[Conclusion générale et perspectives 69](#_Toc435628798)

[Glossaire 70](#_Toc435628799)

[Bibliographie 73](#_Toc435628800)

FIGURE 1 : Organisation de ITEC/FCC 16

FIGURE 2 : Les étapes de l’agilité chez ITEC 18

FIGURE 3 : La méthodologie Scrum 20

FIGURE 4 : Aperçu du logiciel JIRA 23

FIGURE 5 : Schématisation du projet CBS 24

FIGURE 6 : Diagramme de cas d'utilisation général 30

FIGURE 7 : Diagramme de cas d'utilisation de gestion des utilisateurs 32

FIGURE 8 : Diagramme de cas d'utilisation de l’utilisateur 33

FIGURE 9 : Diagramme de cas d'utilisation gestion des instruments 34

FIGURE 10 : Diagramme de séquence système authentification 35

FIGURE 11 : Diagramme de séquence système de gestion d’un instrument 37

FIGURE 12 : Schéma technique de l'architecture physique de l'application 39

FIGURE 13 : Schéma technique de l'architecture logique de l'application 41

FIGURE 14 : Diagramme de package 44

FIGURE 15 : Diagramme de classes global 47

FIGURE 16 : Diagramme de séquence objet authentification 49

FIGURE 17 : Diagramme de séquence système Affichage et génération d’un rapport de type BO……………………..50

FIGURE 18 : Diagramme de séquence objet de recherche d’un instrument 52

FIGURE 19 : Diagramme d'activité modification d’un rapport 54

FIGURE 20 : Diagramme d'activité téléchargement d’un fichier Excel relatif aux prêts et échéanciers. 55

FIGURE 21 : Le pattern Modèle Vue Contrôleur 56

FIGURE 22 : Le principe d’inversion de contrôle 57

FIGURE 23 : Fenêtre authentification 57

FIGURE 24 : Fenêtre d’accueil de l’administrateur 58

FIGURE 25 : Fenêtre gestion des comptes 58

FIGURE 26 : Fenêtre ajout d’un compte 59

FIGURE 27 : Fenêtre modification d'un compte 60

FIGURE 28 : Fenêtre gestion des instruments 60

FIGURE 29 : Fenêtre validation d’un nouveau instrument 61

FIGURE 30 : Fenêtre filtre d’instruments 61

FIGURE 31 : Fenêtre vue maître des instruments filtrés 62

FIGURE 32 : Fenêtre vue détaillée d’un instrument 62

FIGURE 33 : Fenêtre rubrique actifs apportés en garantie 63

FIGURE 34 : Fenêtre affichage message d’erreur sur l’indicateur de données 64

FIGURE 35 : Fenêtre gestion des garanties 65

FIGURE 36 : Fenêtre exemple de saisie des données 65

FIGURE 37 : Fenêtre liste des actifs 66

FIGURE 38 : Fenêtre filtre des actifs 67

FIGURE 39 : Fenêtre gestion des rapports 67

FIGURE 40 : Fenêtre génération des rapports 67

FIGURE 41 : Fenêtre gestion des structures 67

FIGURE 42 : Fenêtre modification d’une structure 67

FIGURE 43 : Fenêtre gestion des taux indexés 67

FIGURE 44 : Fenêtre filtre d’un tiers 67

FIGURE 45 : Fenêtre liste des tiers filtrés 67

TABLEAU 1 : Identification des Acteurs 30

# Introduction générale

E

n 1769, après la guerre de Sept Ans il y avait un grand besoin de financer un

grand nombre de nouveaux bâtiments sur une période de temps court et avec un

coût inférieur en Prusse. Frédéric le Grand a présenté les «covered bonds» pour répondre à ce besoin. Ensuite, les obligations sécurisées sont apparues dans de nombreuses villes européennes comme elles ont montré un résultat financier très positif.

Ensuite ce phénomène a grandi dans le monde des affaires. Un marché pour la thèse des valeurs mobilières a été lancé en 2008 aux Etats-Unis. Puis il a été introduit en Nouvelle-Zélande et de l'Australie à partir de 2010.  
  
À partir de 2012 le volume des obligations sécurisées en circulation dans le monde entier était 2,813 milliards € tandis que les plus grands marchés sont l'Allemagne (525 000 000 000 €), l'Espagne (440 000 000 000 €), le Danemark (366 000 000 000 €) et la France (362 000 000 000 €).

Les obligations sécurisées sont un outil financier utilisé comme techniques de financement et de refinancement émises par les institutions financières. Elles sont garanties par les flux de trésorerie provenant des prêts hypothécaires sous-jacents ou des prêts du secteur public. Les obligations sécurisées sont semblables à des titres adossés à des actifs créés dans la titrisation, mais les obligations couvertes restent sur le bilan consolidé de l'émetteur.

# Chapitre 1 : Contexte

## Introduction

Dans le cadre du projet de fin d’études en vue de l’obtention du diplôme d’ingénieur en

informatique, j’ai eu l’opportunité d’effectuer mon stage au sein du département informatique de la Société Générale (ITEC).Celui-ci gère le maintien et l’évolution du système d’information de l’entité assurant les activités de financement et d’investissement du groupe : GBIS.

L’intérêt de ce premier chapitre est de présenter l’organisme d’accueil la SOCIETE GENERALE, ses activités et ses services, le groupe SGCIB, l’agilité chez ITEC .D’autre part, nous nous focalisons sur la présentation du projet CBS dans son contexte.

# Présentation de l’organisme d’accueil

### *La SOCIETE GENERALE*

Fondée en 1864, la Société Générale qui fête en 2014 ses 150 ans est l’une des plus anciennes et des plus grandes banques françaises mutualistes. Privatisée en 1987, elle a un chiffre d’affaire de 22,8 milliards d’euros en 2013.Avec un siège social à Paris, la Société Générale se développe à l’international et emploie actuellement plus de 148 000 personnes dans 76 pays en Europe, Asie, Afrique et Amérique.

La banque est organisée en trois piliers :

* Les réseaux de détail en France (BDD)
* Les réseaux de détail à l’étranger ou hors France métropolitaine (BHFM)
* La banque de financement et investissement (GBIS, autrefois SGCIB)

Les valeurs défendues par la Société Générale sont l’esprit d’équipe, le professionnalisme et l’innovation. Tout au long de mon stage j’ai pu observer que ces trois principes étaient effectivement mis en œuvre. En effet, le travail d’équipe a été déterminant et toujours très encouragé au cours de mon expérience dans cette entreprise, avec également des points et partages de connaissance pour mieux connaître le travail des autres ce qui aide à créer un esprit d’entreprise. Le professionnalisme est toujours au cœur du travail des équipes projets, et on y

est aidé par les procédures mises en place. Enfin des initiatives ne cessent de se mettre en place

pour améliorer la production en plus des gros projets innovants comme continous delivery.

### *GBIS*

GBIS est une entité de la Société Générale fondée par la fusion entre l’ancienne SGCIB et

les services de banque privée, gestion des actifs et service aux investisseurs. Elle est donc

dédiée aux entreprises, aux institutions financières et aux clients de banque privée.

L’organigramme se trouve en annexe, mais je vais brièvement présenter les départements

avec lesquels j’ai plus particulièrement travaillé. ITEC/FCC, la structure à laquelle j’ai été

intégrée, gère tout l’applicatif pour les activités de financement et de couverture clients. Ses

utilisateurs sont donc des collaborateurs chez GLFI(le front office), OPER (middle et back

office) et FIND pour la comptabilité.

### *Le service*

J’ai effectué ces six mois de stage intégrée à l’équipe ITEC/FCC/OSD. Pour pouvoir comprendre mon expérience, il faut donc tout d’abord savoir la place qu’occupe ce service dans GBIS ainsi que ses missions principales.

### *1.3.1. ITEC/FCC/OSD*

Le département ITEC (Information Technology Department) assure tous les services informatiques pour les métiers de GBIS. Ses deux missions sont :

* D’assurer une bonne production des applications et des infrastructures
* De faciliter les améliorations et les croissances d’activité

Au sein d’ITEC, l’entité FCC (Financing and client Coverage Technology) s’occupe plus

particulièrement des activités de financement et de couverture client. Elle gère donc les

projets d’évolution ainsi que la maintenance de toutes les applications couvrant ses secteurs.

Les sujets discutés actuellement chez FCC en plus de l’agilité sont la production en

Continuous Delivery (l’amélioration continue), Digitale transformation et le Knowledge

Management.

FCC est composé de trois services « projets » : OSD (Origination, structuring and

distribution), LTL (Loan Servicing, Trade Finance and Leasing) et RPP (Risk, Provisions and

Profitability). On y trouve également un service de production, PRS (Production & Services),

qui comporte une équipe de support applicatif et une équipe de support technique (AOP) pour

chaque service projet.

**Figure 1 : Organisation de ITEC/FCC**

ITEC/FCC/OSD est donc le service qui gère le support fonctionnel pour les applications

de FCC/OSD. Il est divisé en deux périmètres : PCL pour le pré-closing (donc avant la

signature des deals avec les clients) et les applications permettant le suivi de la connaissance

des clients et ORI pour tout ce qui concerne la titrisation et la syndication.

### *1.3.2. Interactions avec les autres équipes*

Les «clients» d’ITEC/FCC sont en fait les utilisateurs des applications et donc des

collaborateurs de GBIS.

Les autres équipes avec lesquelles des points réguliers et du travail en commun est

mis en œuvre sont l’équipe projet Bangalore (travaillant sur CBS), le support applicatif

et les AOP (support technique) qu’ils soient à Paris ou à Bangalore(Inde).

# La méthodologie adaptée

### 2.1. L’agilité chez ITEC

ITEC a fait des méthodes Agiles l’un des points les plus importants de son

développement à partir de 2011. En effet une nouvelle approche a été définie en ne

définissant plus ses concurrents comme les systèmes d’information des autres banques mais

les géants du web comme Google, Amazon ou Facebook. Le but est d’assurer un service

extrêmement adapté aux besoins de l’utilisateur, de bonne qualité et rapide. Agile répond

parfaitement à cette attente.

Durant tout mon stage d’apprentissage j’ai pu assister à une formation sur l’agilité et

l’amélioration continue assurée par un coach agile. Pour l’accompagnement de ce

changement de méthode en effet, ITEC a constitué une équipe de spécialistes qui ne

travaillent que sur le sujet, il y a ainsi un coach pour chaque projet agile. Cette formation

expliquait les bénéfices, contraintes, méthodes et outils conseillés. Cela m’a été très utile

étant donné que j’ai pu avoir un aperçu clair de l’adaptation des principes de l’Agilité que

j’avais vus en cours au contexte particulier de la Société Générale.

Les centres Agiles chez ITEC a définit plusieurs étapes pour passer du cycle en V au

Continuous Delivery qui consiste à pouvoir livrer en continu sans action de

l’utilisateur et sans diminuer la qualité.

**Figure 2 : Les étapes de l’agilité chez ITEC**

1. Agile Foundation : première implantation des méthodes Agiles avec le travail par

itérations, implication des responsables métiers le plus tôt possible dans le processus pour

comprendre le besoin intelligemment, faire attention à apporter de la valeur métier.

Pour cela on a inventé le BDD (Behaviour Driven Developement) qui consiste à construire

les spécifications ensemble avec un développeur, un Business Analyste, un testeur et le

client. De cette façon on s’assure de bien comprendre le besoin et on éduque l’utilisateur à

être précis.

1. Craftsmanship : rendre le code plus robuste avec automatisation des tests.

La Société Générale utilise une méthode TDD (Test Driven Developpement), c’est-à-dire

que les tests unitaires sont écrits avant le développement. De plus certains outils permettent

d’automatiser tout ou partie des tests.

1. DevOps : gagner du temps, accélérer les livraisons, aller au rythme du client et ne plus

être un frein. C’est là qu’interviennent les équipes de support applicatif et technique.

Cela peut être atteint avec des « Non event releases », donc des livraisons de petites

fonctionnalités, plutôt que grosses releases de type « Big Bang ».

1. Continuous Delivery : être capable d’effectuer des releases en continu sur toute la

chaîne d’applications qui participent au business process.

### *2.2. Organisation de l’équipe CBS*

Dans le cadre du projet de CBS développement logiciel, le client élabore sa vision du

produit à réaliser et liste les fonctionnalités ou les exigences de ce dernier. Il soumet cette

liste à l’équipe de développement, communique directement avec elle (plutôt que par

papier) qui estime le coût de chaque élément de la liste. Le projet manager peut ainsi se

faire une idée approximative du budget global.

L’équipe sélectionne ensuite une portion des exigences à réaliser dans une portion de

temps courte appelée itération. Chaque itération inclut des travaux de conception, de

spécification fonctionnelle et technique quand c’est nécessaire, de développement et de

test. A la fin de chacune de ces itérations, le produit partiel mais utilisable et montré au

client. Ce dernier peut alors se rendre compte par lui-même très tôt du travail réalisé, de

l’alignement sur le besoin.



**Figure 3 : La méthodologie Scrum**

* Les éléments « Product Backlog » :

Objectif : liste de fonctionnalités attendues d’un produit.

Contenu : Les fonctionnalités listées sous forme de « User Stories ».

Rôles : Sous la responsabilité du Product Owner et Scrum Master qui se chargent de

classer les éléments par priorité et déterminent l’ordre de réalisation.

Utilisation : Elaboré avant le lancement des sprints et utilisé pour planifier une release.

PS : A tout moment le Backlog est visible dans l’outil Jira par tout le monde.

* Les éléments « Sprint planning » :
* Une sélection de fonctionnalités du Backlog à développer pendant le sprint.
* Le sprint planning a lieu à chaque début de sprint.
* Tout le monde est présent à cette réunion.
* Le scrum master sélectionne des fonctionnalités à développer pour le prochain sprint en

fonction de :

- La priorisation des fonctionnalités

-La capacité de production de l’équipe

* Les éléments « Sprint » et « Release » :
* Un sprint est une itération : période au bout de laquelle est livré un incrément de

produit.

* Chaque sprint est composé d’une liste de fonctionnalités à réaliser.
* Une release est une somme de sprint.
* A la fin d’une release on passe en production.
* Les cérémonies « Daily scrum meeting » :

Permet aux membres de l’équipe de faire un point de coordination sur les tâches en cours

et les difficultés rencontrées.

* A lieu tous les jours à la même heure
* Dure 15 minutes au maximum
* Se tient par conférence call avec l’équipe projet à Bangalore et à Tunis (Daily call meeting)
* A tour de rôle, chaque membre répond aux questions suivantes :

- Qu’ai-je fait hier ?

- Que dois-je faire aujourd’hui ?

- Quelles sont les difficultés rencontrées ?

* Mise à jour du « statut des items dans **Jira** »

- To do

- In progress

- Done

* Les cérémonies « Sprint retrospective » :

Permet de faire le bilan sur ce qui s’est passé pendant le sprint.

* Chacun s’exprime pour dire :

-Ce qui s’est bien passé

-Ce qui s’est mal passé

-Ce qu’il faudrait améliorer

* On vote ensuite pour ce qui doit être amélioré en priorité
* Permet de traiter les éventuelles tensions au sein de l’équipe

### *2.3. Les impacts dans CBS*

* CBS est une petite équipe qui suit la livraison en fonction de mêlée.
* Chaque sprint est de 2 semaines en longueur et permet les demandes de haute

priorité supplémentaires à ajouter au sprint sans enlever quoi que ce soit à partir du

backlog de sprint déjà traité.

* L’application CBS s’appuie essentiellement sur l’approche Kanban qui consiste

globalement à visualiser le Workflow (Le processus de traitement d’une tâche).

* La participation du client avec l'équipe à travers la discussion de priorisation

hebdomadaire sur une base cohérente.

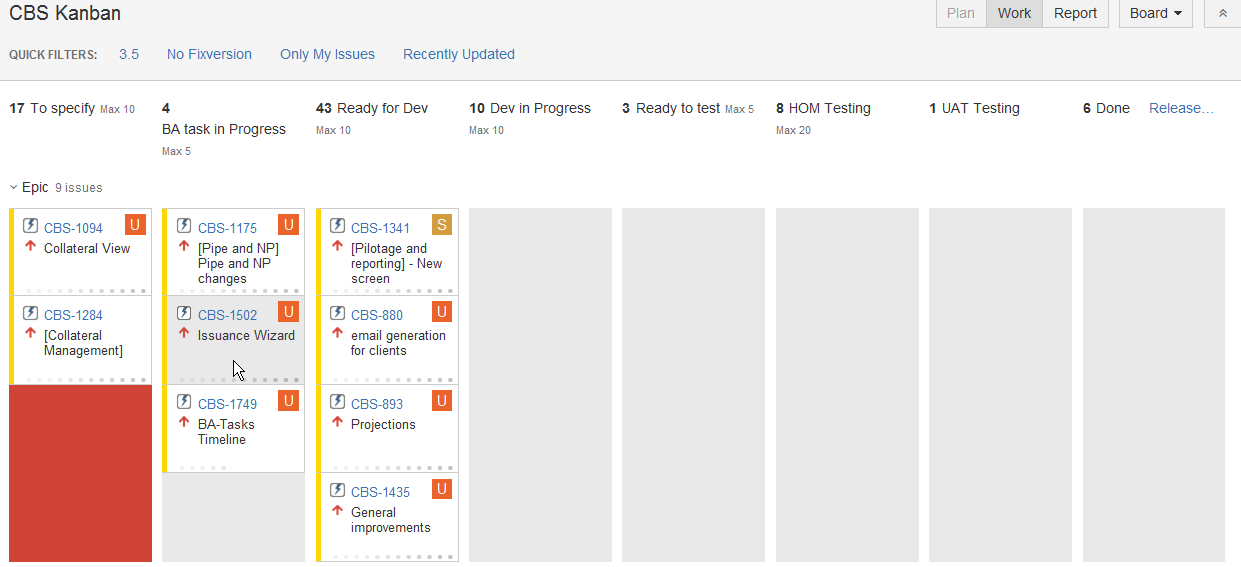
* Fixation des défauts hebdomadaires détectés afin d’améliorer la qualité.
* Personnalisation du tableau Kanban Jira qui est accessible par toute l'équipe et

par le PO (Product Owner).

* Réaliser des rétrospectives dans le but d'inspecter le processus et apporter des

améliorations au projet avec des nouvelles initiatives.

******



**Figure 4 : Aperçu du logiciel JIRA**

# Présentation du projet

Le but de notre application CBS est de créer un outil dédié à la surveillance et la déclaration des « Covered Bonds » de SG émis par des entités dédiées (SG SCF, SG SFH).

Structure Investisseurs

**Figure 5 : Schématisation du projet CBS**

SG transfert à la structure de la garde des prêts qui seront utilisés comme garantie. SFH ne prendra que les prêts immobiliers, SCF seulement les prêts publics.

Les structures couvrent les obligations pour le même montant (ou ci-dessous) que les prêts utilisés comme garantie. Ces obligations vont avoir un taux de coupon plus faible car ils sont moins risqués.

Lorsque les investisseurs achètent les obligations sécurisées, la structure émet un prêt

avec la même quantité de SG.

SG paye des intérêts à la structure.

Comme les intérêts de prêts sont basés sur les taux flottants et les coupons sont basés sur les taux fixes, un swap (échange) est nécessaire. Tout cela sera assuré par la structure de SGCIB.

Les coupons sont versés aux investisseurs.

## Problématique

Les Covered Bonds sont des obligations garanties par un portefeuille dédié d’actifs sous-

jacents, destinés à garantir aux porteurs obligataires une sécurité supérieure à celle d’un

titre obligataire classique. La crise a particulièrement mis en lumière l’intérêt et la

résilience des émissions de cette nature, qui jouent désormais un rôle très significatif dans

le paysage obligataire européen.

# La solution proposée

L'objectif de la gestion des Covered Bonds est d'émettre des obligations à faible taux

afin d'augmenter la liquidité pour le groupe de SG. Il doit veiller à ce que toutes les

obligations émises seront couvertes par des actifs utilisés comme collatéraux.

Afin d’améliorer le processus de la bonne conduite des « Covered Bonds », le

département ITEC/FCC/OSD était en mesure de créer l’application CBS répondant à tous

ces besoins en fusionnant deux équipes l’une à Bangalore et l’autre située à Paris.

# Organisation de ma mission

### *5.1. Intégration dans l’équipe*

Pour répondre à cette problématique, j’ai été totalement intégrée à l’équipe projet CBS localisée à Paris. J’ai donc eu la chance de pouvoir travailler avec chacune des personnes présentes et de participer à toutes les activités d’un développeur informatique. J’ai été traitée comme n’importe quel nouvel arrivant dans l’équipe et donc formée au fur et à mesure aux processus et application afin d’effectuer les mêmes tâches que les autres développeurs de l’équipe.

L’une des trois valeurs affichées par la Société Générale est l’esprit d’équipe, et j’ai trouvé tout au long de mon stage que c’était effectivement le cas. De fait, tout est mis en œuvre pour favoriser cet esprit, tant par les différents points permettant de suivre l’avancée des autres que par le travail en groupe ou les événements extérieurs à tous les niveaux. Pour moi, cette intégration complète dans une équipe soudée a été une réelle différence, en comparaison à mes précédents stages par exemple.

### *5.2. Suivi de ma mission*

Le stage consiste à plusieurs développements  ad ‘hoc (adéquat) dans un mode agile de type Kanban :

* Participer aux corrections des bugs et la mise en place des nouveaux besoins (Defects

& Backlog) :

* Développer des interfaces Front End dédiées aux utilisateurs de l’application.
* Apporter des enrichissements et des améliorations à des modules déjà existants.
* Résoudre des problèmes de régression liés à des défauts, défaillances et des erreurs issues

d’une mauvaise spécification ou d’une inappropriée implémentation du code.

* Participer à mettre en place/maintenir des tests unitaires (TDD : Test Driven Development).
* Analyser et tester la compatibilité de l’application avec les navigateurs Chrome,

Firefox et IE11. (L’application était seulement compatible avec IE8).

* Rendre l’application compatible avec les navigateurs indiqués au dessus

(sans créer des régressions pour le navigateur IE8).

## Conclusion

Ce premier chapitre a été élaboré afin de mettre en œuvre le contexte général du projet, il nous apparaît aussi possible de mettre au point le choix de la méthodologie CBS Kanban adaptée comme étant une démarche efficace et intuitive pour bien articuler la problématique du travail et tirer par la suite les objectifs à achever.

Passons au chapitre suivant qui sera voué à l’analyse et à la spécification des besoins.

# Chapitre 2 : Analyse et Spécification des besoins

## Introduction

#### Durant ce chapitre, nous allons procéder à présenter la phase de spécification et d’analyse des besoins. Dans le but de mieux comprendre l’objectif de l’application CBS nous présenterons d’une manière clarifiée l’étude préliminaire dans laquelle nous énonçons quelques besoins fonctionnels de notre application. A posteriori, nous examinerons la branche fonctionnelle suivie de la branche technique.

# Etude préliminaire

### *1.1. Recueil des besoins fonctionnels*

CBS est divisé en sept principaux modules fonctionnels :

* **Authentification**
* **Gestion des actifs et des passifs**
* Permet à l’utilisateur d’accéder à l’ensemble des prêts vivants à une date

donnée.

* **Gestion des collatéraux (garanties)**
* **Structure Sociale**
  + - Gestion de l’ensemble des instruments (prêts, échanges, obligations).
* **Pilotage & Rapports**
  + - Permet à l’utilisateur de générer tous les rapports dans l’application à un

seul point.

* **Gestion des Référentiels**
* Gérer les structures.
* Gérer le taux indexé.
* Gérer le référentiel tiers.
* **Administration des droits de l’utilisateur**

### *1.2. Identification des acteurs*

Dans cette partie, on peut distinguer les différents acteurs participant dans notre application

CBS :

* Utilisateurs :
* Les utilisateurs sont que des membres internes du groupe SG.
* Nombre par emplacement et par entité: 20 au maximum à PARIS.
* Participation à l'utilisation des applications: Middle Office, Front Office, Support.
* Pas d’utilisateurs internationaux ni d’utilisateurs externes (projet pas ouvert à

l'Internet).

* Administrateur :
* Gérer l’application dans son univers.
* L’équipe Support :
* Assurer les services de maintien et d’assistance de l’application.

Tableau 1: Identification des Acteurs

|  |  |
| --- | --- |
| **Acteur** | **Rôle** |
| **Utilisateurs internes**  **du groupe SG** | * Lecture seule: l'utilisateur n'a pas le droit   de créer, modifier ou supprimer les données de  l'entreprise.   * Lecture et écriture: l'utilisateur est   autorisé à créer, modifier ou supprimer les  données de l'application.   * Authentification. * Consulter les prêts vivants à une date   donnée.   * Importer les collatéraux (garanties). * Consulter l’indicateur de qualité de   données suite à l’import des collatéraux.   * Gestion de l’ensemble des instruments   (prêts, échanges, obligations).   * Lister, créer ou rechercher une structure. * Lister, créer ou rechercher un taux indexé. * Lister, créer ou rechercher un référentiel   tiers.   * Générer les rapports. |
| **Administrateur** | * Il a les mêmes autorisations que le rôle   lecture et écriture.   * Authentification. * Gérer les utilisateurs et leurs rôles (ajout,   modification et suppression des utilisateurs). |
| **L’équipe Support** | * Le support dans CBS prend en charge les   demandes du premier et du début du deuxième  niveau (Ajout d'utilisateur, Monitoring de  l'application, Gestion des incidents, prise en  compte des requêtes simples et suivi avec le  projet des requêtes complexes..).   * Ils ont les mêmes autorisations que l’administrateur et l’utilisateur, ils possèdent de plus l’accès à la rubrique support. |

### *1.3. Besoins non fonctionnels*

Tous les systèmes d’information à un certain point dans leur cycle de vie doivent considérer que les besoins non-fonctionnels apportés à leurs projets reflètent un travail très important et utile pour s’assurer d’une réalisation efficace et d’un bon fonctionnement.

###### Sécurité :

###### Besoin d’établissement de la connexion selon les règles d’autorisation.

###### Besoin de mot de passe et d’identifiant répondant à une politique d’utilisation :

###### longueur de mot de passe, expiration….

###### Performance :

* Respecter un temps de réponse : le chargement de l’application, ouverture d’écran et des

délais de rafraîchissement…

* Performance en temps de traitements (fonctions, calculs, importations/exportation de

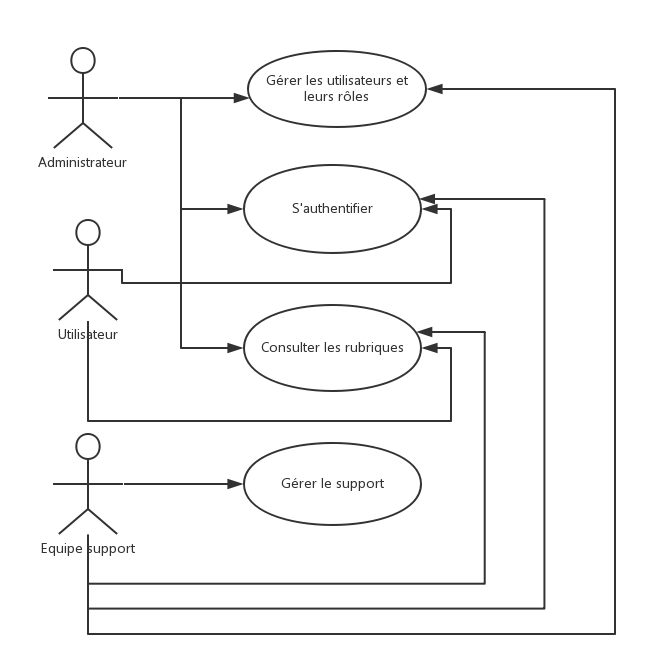
données…

# Branche fonctionnelle

Dans cette partie, nous allons projeter un point de vue fonctionnel sur l’ensemble de l’application. Pour cela, nous nous appuierons sur la schématisation des diagrammes de cas d’utilisation afin de présenter l’unité discrète d’interaction entre l’utilisateur et le système.

### *2.1. Identification des cas d’utilisation*

Voila ci-dessous le diagramme des cas d’utilisation général de notre application CBS :



**Figure 6 : Diagramme de cas d'utilisation général**

Le diagramme de cas d'utilisation général illustré ci-dessus définit d’une manière assez globale d’une part nos acteurs déjà identifiés auparavant et d’autre part les principales fonctionnalités associées à chacun d’entre eux.

Par la suite, nous allons distinguer pour chaque acteur ses fonctionnalités clefs adéquates.

### *2.2. Description des cas d’utilisation*

### *2.2.1. Cas d’utilisation relatifs à l’administrateur*

L’administrateur possède un double accès à l'application : il gère les utilisateurs, il possède aussi les droits d’écriture et de lecture des données de l'application et ceci après authentification.

Cette figure représente le diagramme de cas d’utilisation de l’administrateur :

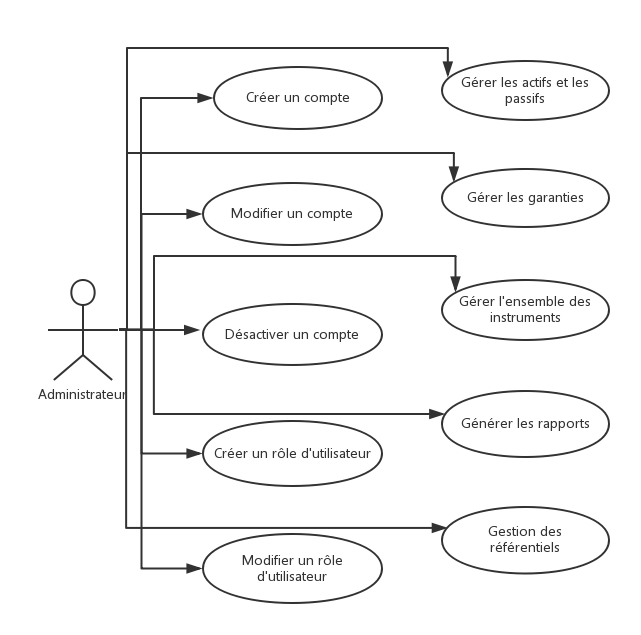


Figure 7 : Diagramme de cas d'utilisation de gestion des utilisateurs

### *2.2.2. Cas d’utilisation relatifs aux Utilisateurs*

Tout utilisateur interne du groupe SG possède soit les droits de lecture seulement soit les droits de lecture et écriture, par conséquence, cet acteur peut après toute authentification Consulter les prêts, importer des garanties que suite à ça il peut consulter l’indicateur de qualité relatif, gérer tout type d’instruments (prêt, échange ou bien des obligations) , gérer toute structure, gérer les taux indexés ,contribuer aussi à la gestion d’un référentiel tiers et générer des rapports.

Cette figure représente le diagramme de cas d’utilisation de l’utilisateur :

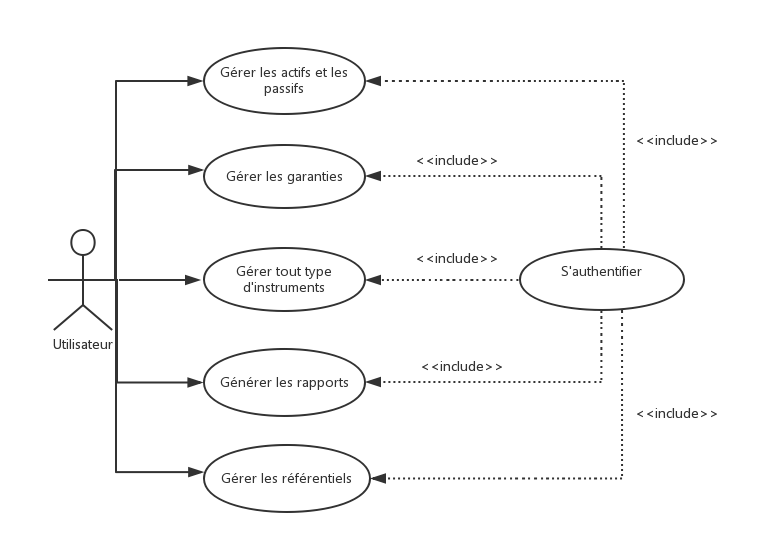


Figure 8 : Diagramme de cas d'utilisation de l’utilisateur

### *2.2.3. Cas d’utilisation relatifs à la gestion des instruments*

Une fois que l’utilisateur se connecte à l’application, il peut ainsi consulter l’interface vue sociale de la rubrique structure afin de créer un montage avec l’un des instruments (prêt, échange, obligation), éditer un instrument, valider un type simulé des instruments ou bien même il peut rechercher un type de ces derniers selon la date de maturité saisie.

Cette figure représente le cas d'utilisation de la gestion des instruments:

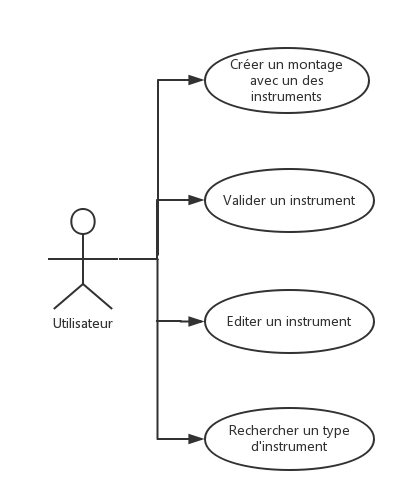


Figure 9 : Diagramme de cas d'utilisation gestion des instruments

### *Diagramme de séquence système*

### *2.3.1. Diagramme de séquence système de l’authentification*

L’authentification est la première étape à faire pour accéder aux fonctionnalités de l’application. La connexion se fait par saisie du login/mot de passe Sesame.

Dans la figure ci-dessous nous illustrons son séquencement :

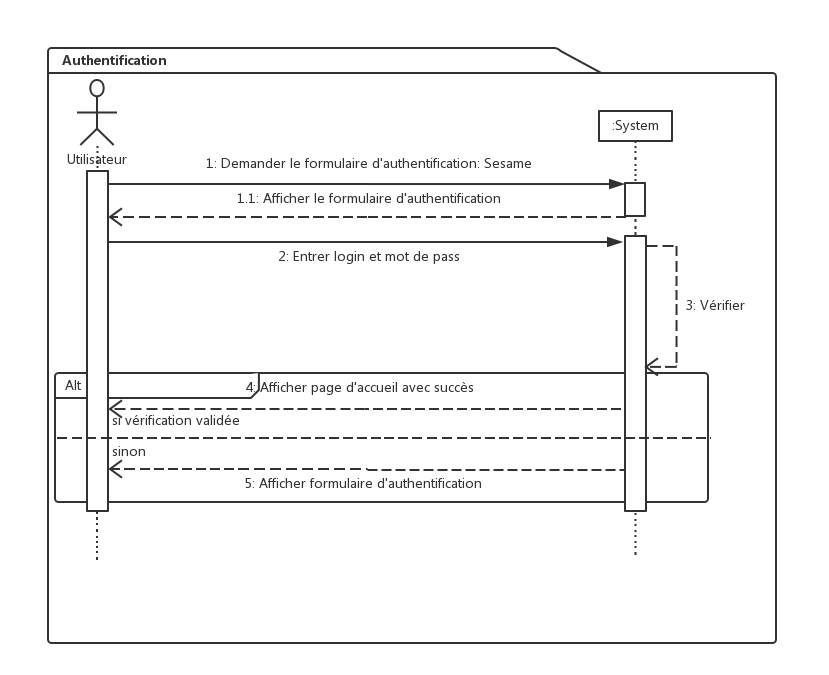


Figure 10 : Diagramme de séquence système authentification

La simulation de l’authentification sera traitée comme l’enchaînement suivant :

1. L’utilisateur accède à l'interface d'authentification.
2. Il saisie son login et son mot de passe.
3. Le système vérifie les données saisies par l'utilisateur.
4. Si les données saisies sont valides alors il aura accès aux différentes sections de l’application.
5. Si le login/mot de passe est incorrecte il sera rejeté.

### *2.3.2. Diagramme de séquence système de gestion d’un instrument (produit financier)*

CBS dispose d’un ensemble de vues dans la rubrique «vue sociale» sous la barre de menu «Structure» permettant de valider un nouveau instrument, retrouver rapidement chaque instrument ou groupe d’instruments mais aussi éditer un instrument.

Dans la figure ci-dessous nous illustrons son séquencement:

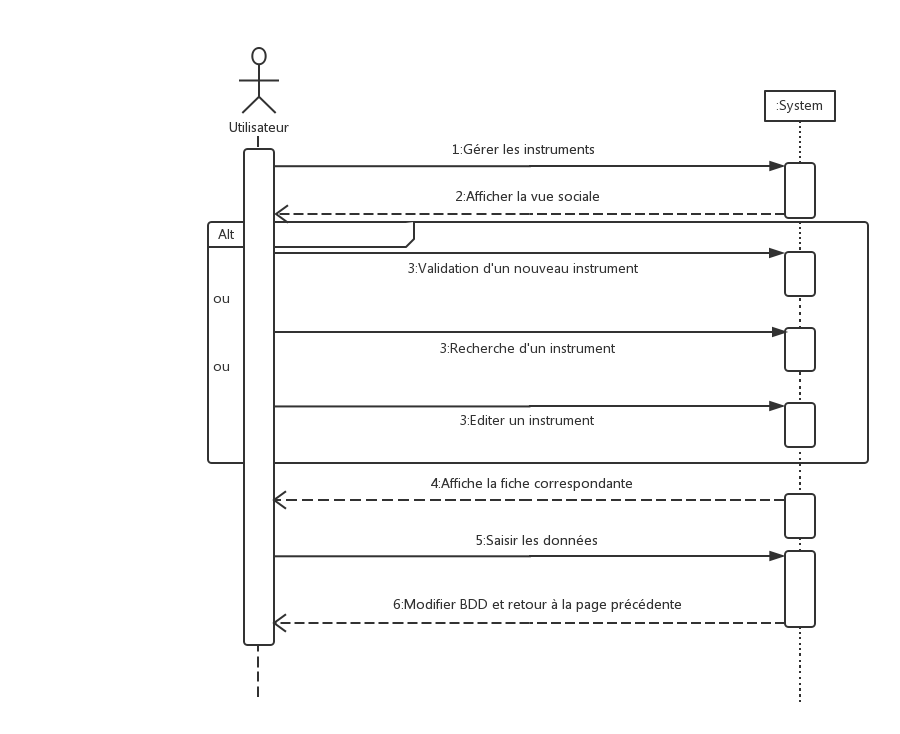
****

Figure 11 : Diagramme de séquence système de gestion d’un instrument

L’application CBS met à la disposition de ses utilisateurs la possibilité de gérer tout type de produit financier. Comme indiqué ci-dessus, le principe consiste à interroger notre système afin d’avoir la vue correspondante à chaque action soulevée par l’utilisateur, qui suite à ça aura la possibilité de traiter son intervention en ayant comme retour les modifications effectués et redirection vers la page précédente.

# 3. Spécification technique

L’architecture technique de CBS est une fusion de technologies bien établies et émergentes, cette phase est évaluée comme étant une branche fondamentale dans le cycle de vie d’un projet, elle illustre une illusion générale sur l’architecture globale de notre application.

### *3.1. Contraintes et choix techniques*

* Respecter les normes d’ITEC/FCC

CBS est développée et maintenue par ITEC/FCC, ceci conduit à choisir une architecture basée sur la technologie J2EE.

* Accès facile et sécurisé

Une contrainte importante est de veiller à ce que CBS peut être facilement accessible par les

utilisateurs de SG à partir de différents endroits, et de manière sécurisée. Cette contrainte nous conduit à choisir l'architecture client léger disponible à travers HTTPS.

* Confidentialité et accès des données

Les autorisations seront gérées par Spring Security 3. Cela permettra d'avoir une solution de

maintenance de coûts flexibles et faibles pour la gestion de l'autorisation.

* Authentification

Sesame sera utilisé pour la couche d'authentification.

* Description de niveau de service

Il n'y a aucune nécessité de soutenir des volumes élevés comme le nombre maximum

d'utilisateurs ne dépassera pas 20 utilisateurs, dont 10 à 15 pourraient être connectés

simultanément et en même temps. L'application doit prendre en charge au moins 3 clics

par seconde, et répondre à toute clic dans deux secondes ou moins.

### *3.2. Modélisation et architecture de l’application CBD*

### *3.2.1. Architecture physique*

L’architecture physique de l’application décrit l’ensemble des composants matériels qui supportent l’application dans son environnement.

La figure ci-dessous représente le schéma physique de l’architecture de l’application.

**Prod Linux VM**

**ʷä Prod**

**Tomcat 7**

**Homol Linux VM**

**ʷä Homol**

**Tomcat 7**

**ʷä UAT**

**Tomcat 7**

**#è Prod**

**Business DB**

**#è Homol**

**Business DB**

**#è UAT**

**Business DB**

**#è Dev**

**Business DB**



**Developer**

**Figure 12 : Schéma technique de l'architecture physique de l'application**

Notre architecture physique décrit brièvement les composants logiciels (schéma de base de

Données spécifique) déployés sur les composants matériels (machines virtuelles).Comme c’est

illustré dans la figure ci-dessus l’infrastructure de production est totalement indépendante des

autres environnements.

### *3.2.2. Architecture logique*

###### L’architecture logique d’une application permet de décrire d’une manière symbolique et

###### schématique les différentes couches communicantes entre elles à travers des requêtes

###### (HTTPS, SQL..).

La figure ci-dessous représente le schéma logique de l’architecture de l’application.

**Data Access**

Spring managed beans

Business layer

Application tier

Presentation layer

**Data Model**

(POJO)

HTTPS

**Client**

(Web browser/ IE8)

**Control**

(Spring MVC controllers)

**Excel views**

(Aspose)

**HTML views**

(JSP, jQuery)

**User Authorization**

(Spring beans)

**Sesame**

(Only Authentication)



Persistence layer

**S**

**P**

**R**

**I**

**N**

**G**

**F**

**R**

**A**

**M**

**E**

**W**

**O**

**R**

**K**

**Business Logic**

(Spring beans)

HTTPS

JDBC

Data tier

External systems

**1 Oracle Database**

CBS Business Schema

**Figure 13 : Schéma technique de l'architecture logique de l'application**

Les niveaux d'application suivent une architecture en couches superposées:

* Couche de présentation
* Couche métier encapsulant la logique métier et le contrôle
* Couche de persistance encapsulant la logique d’accès aux données

Nous privilégions une approche légère pour assembler ces couches, basée sur POJO et en

utilisant le Framework Spring.

Les Frameworks qui sont utilisés dans chaque couche :

* JPA et Hibernate 3 pour la couche de persistance.
* Spring, Spring MVC, JSP, Aspose pour les deux couches: présentation et métier.
* ***Organisation des couches***
* Pour réduire le couplage entre les couches, l'interaction entre les objets sera gérée par les

caractéristiques d'injection de dépendance de Spring.

* Un client de type browser qui prend en compte du HTML, java Script/AJAX.
* Un serveur web (Apache Tomcat7) pour le traitement des requêtes HTTP en

provenance d’un navigateur.

* Chaque mise en œuvre d'une fonctionnalité de la couche métier sera représentée par

une interface façade.

## Conclusion

Dans ce chapitre, on a essayé de cerner les objectifs du système cible. Ces objectifs doivent tenir comptes des besoins de l’utilisateur. Toute cette étude a permis de bien définir les cas d’utilisation et de décrire toutes les fonctionnalités de l’application, d’où la spécification du système cible. Dans la prochaine étape, on va procéder à la description détaillée et la conception de l’application.

# Chapitre 3 : Conception

## Introduction

Dans ce chapitre, nous allons présenter la phase conceptuelle de notre projet. Cette phase est constituée de deux étapes : la conception préliminaire qui représente le croisement entre branche fonctionnel et branche technique, et la conception détaillée qui va se focaliser sur les patrons de conception d’une part et quelques vues statiques et dynamique du système.

# 1. Conception préliminaire

La fusion entre la branche fonctionnelle et la branche technique s’effectue dans ce stade, c’est l’étape la plus délicate, qui va intégrer la branche fonctionnelle qui constitue l’âme de projet par la branche technique qui constitue le squelette de ce dernier de manière à tracer la cartographie des composants du système à développer.

### *1.1. Vue Statique*

Dans cette partie, nous nous attardons sur la vue statique de la conception. Le point de vue statique met l’accent sur la structure statique du système à l’aide d’objets, attributs, les opérations et les relations. Ce modèle inclut le diagramme de package et le diagramme de classe.

### *1.1.1. Diagramme de packages*

Un diagramme de packages permet d’étudier la représentation graphique des relations existants entre les paquetages composant un système et de représenter la dépendance fonctionnelle en terme de données entre eux.

La figure ci-dessous représente le diagramme de package de notre application :

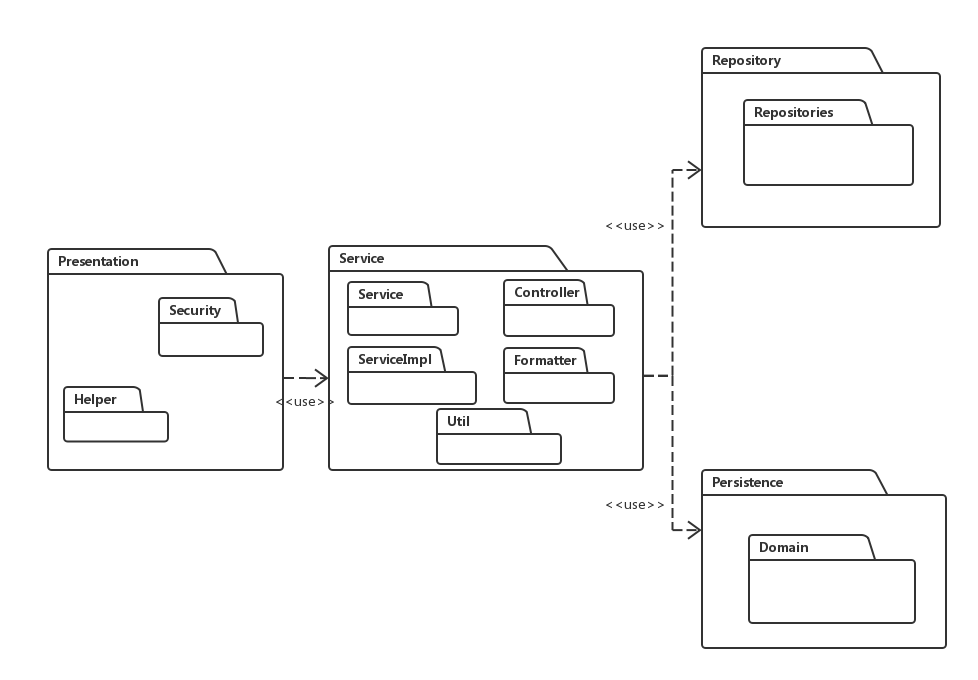


Figure 14 : Diagramme de package

Notre application est répartie en quatre paquetages :

* Paquet « Persistence » :
* Présentant les classes principales ou POJO qui sont en

Interaction avec la base de données. Ces classes sont gérées par l’API JPA (Java Persistence

API).

* Hibernate 3 est préoccupé par la persistance des données, en permettant d’établir la

communication entre les classes définies dans le sous-paquet « Domain » avec les

SGBD (système de gestion de base de données) qui est ORACLE dans notre application.

* Paquet « Repository » :
* Faciliter l'écriture des couches d'accès aux données.
* Paquet « Service » :
* Des classes de services traitant les méthodes nécessaires d’accès à la base de

données.

* Paquet « Présentation » :

* Le paquet Présentation communique directement avec le paquet service afin de d’importer les

éléments nécessaires pour créer les pages JSP.

### *1.1.2. Diagramme de Classes*

Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations. A ce titre, chaque application qui va mettre en œuvre un système sera une instance des différentes classes qui le compose.

La figure ci-dessous représente le diagramme de classe global de l’application :

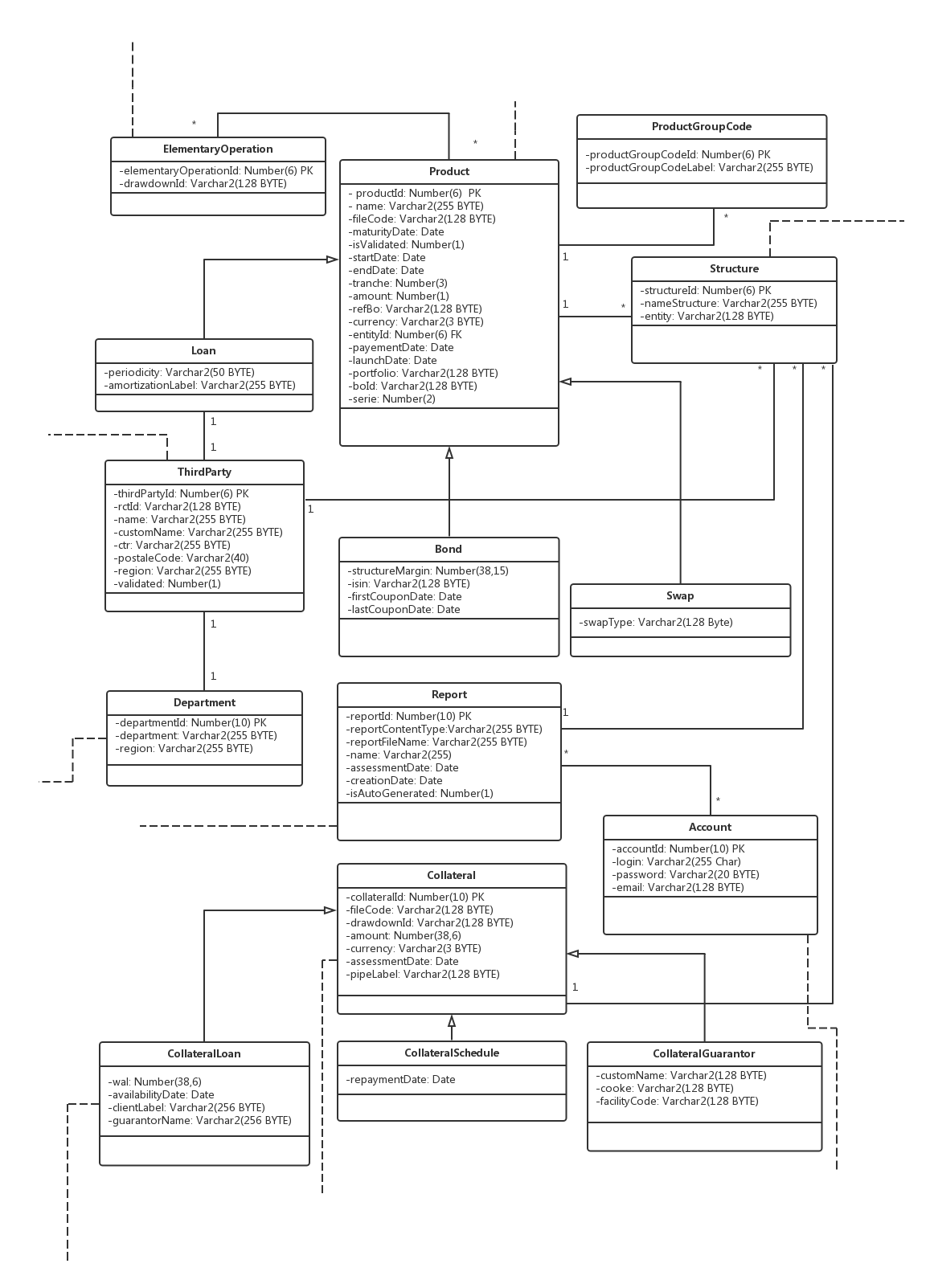


Figure 15 : Diagramme de classes global

### *1.2. Vue Dynamique*

Cette vision inclut des diagrammes de séquences, des diagrammes de collaborations, des diagrammes d’activités et des diagrammes d’états. Dans cette section, nous allons décrire les différentes opérations qui conduisent à la construction du modèle d’évolution du système et une séquence de ses opérations. Nous avons vu dans le chapitre précédent que le système contient trois modules importants ce qui nous pousse à choisir quelques-uns parmi eux pour pouvoir terminer la conception et la réalisation dans les délais.

### *1.2.1. Diagramme de séquence objets*

Le diagramme de séquence objets montre des interactions entre objets selon un point de vue temporel. Le contexte des objets n’est pas représenté de manière explicite comme dans les diagrammes de collaboration. La représentation se concentre sur l’expression des interactions. Ce type de diagramme représente une interaction entre objets en insistant sur la chronologie des envois de messages.

### *1.2.1.1. Diagramme de séquence objet « authentification »*

L’authentification est la phase qui permet à l’utilisateur d’apporter la preuve de son identité et par la suite assurer la sécurité et la confidentialité de ses données. Tout utilisateur interne à la SG doit saisir correctement son login et son mot de passe afin d’avoir le droit d’accès pour consulter l’espace de l’application, dans le cas échéant il y aura une redirection vers la page d’authentification Sesame.

**System**

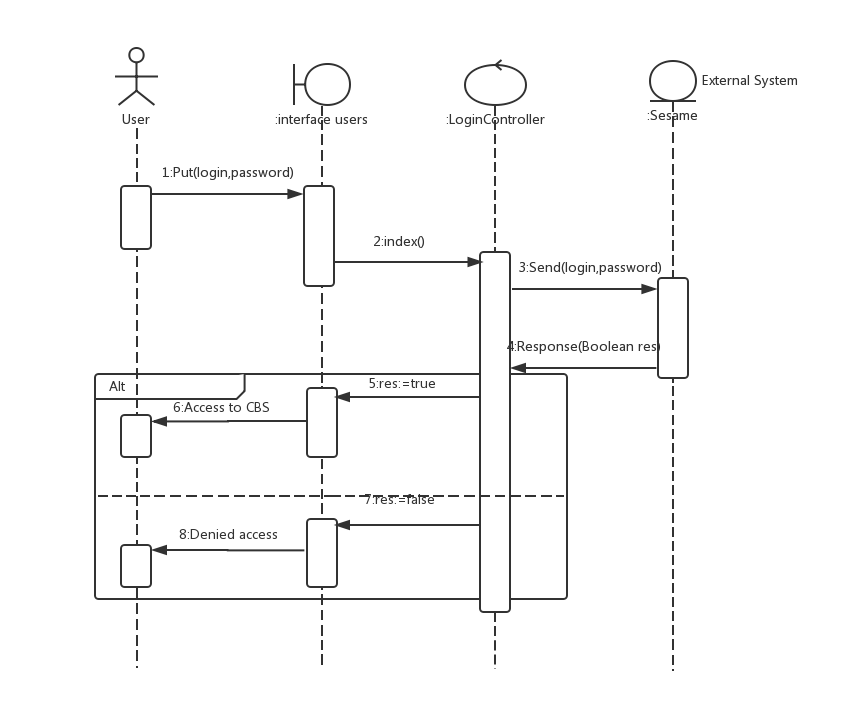


Figure 16 : Diagramme de séquence objet authentification

### *1.2.1.2. Diagramme de séquence objet d’affichage et génération d’un rapport de type BO*

Il existe trois types de rapports dans l’application que nous pouvons les générer : rapport Moody’s, rapport S&P et rapport BO, dans ce diagramme de séquence système nous allons travailler sur le type de rapport BO.

Dans la figure ci-dessous nous illustrons son séquencement :

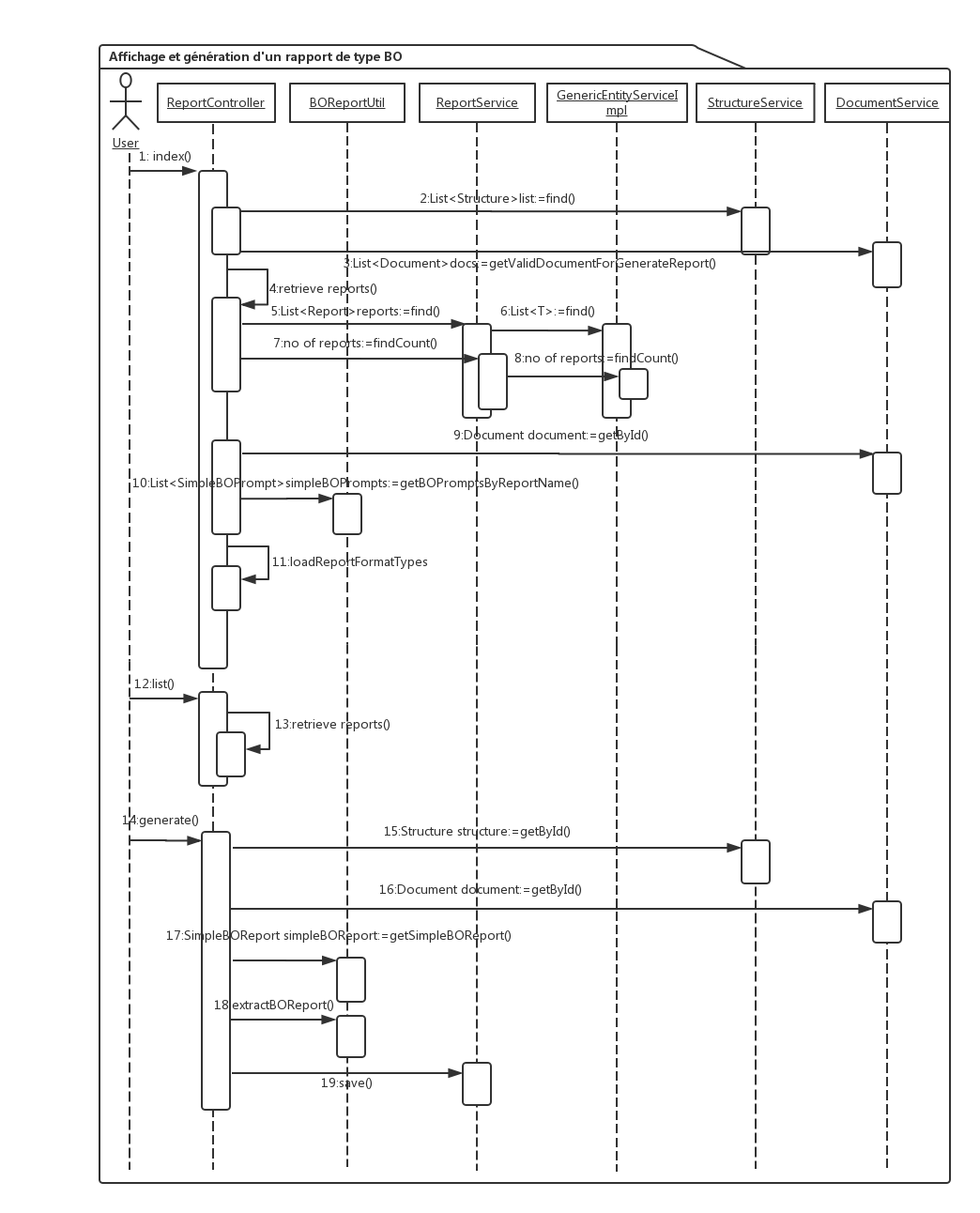


Figure 17 : Diagramme de séquence système Affichage et génération d’un rapport de

type BO

Une fois que l’utilisateur aura l’accès à la vue de «Pilotage et Rapports», il devrait suivre les étapes ci-dessous pour générer le rapport :

1. Saisir des données (Structure, date d’inventaire, type de rapport et format du

rapport) afin de lister les noms des rapports existants pour ce type d’entrées.

1. Pour pouvoir générer un rapport, il suffit de cliquer sur le bouton «voir» de manière

à y accéder à la page «Générer» via laquelle nous y trouvons le bouton «Générer» permettant la génération du rapport approprié.

### *1.2.1.3. Diagramme de séquence objet de recherche d’un instrument*

CBS dispose d’un ensemble de filtres permettant de retrouver rapidement chaque instrument

ou groupe d’instruments.

Dans la figure ci-dessous nous illustrons son séquencement :

###### D:\mbenabda062215\Stage\Docs_Rapport\seq syst diag 2.png

Figure 18 : Diagramme de séquence objet de recherche d’un instrument

Une fois que l’utilisateur aura l’accès à la vue de «Vue Sociale», il devrait saisir des données (Structure, type de produit, année de maturité..) suivant lesquelles un filtre sera généré permettant de retrouver rapidement un produit financier qui aura une vue donnant les caractéristiques détaillées. Dans la figure ci-dessus, comme c’est illustré pour filtrer un tel instrument, Social View Controller fait appel à Elementary Operation Service qui se charge d’envoyer un message de recherche au service spécifique d’un tel produit (prêt, échange…), ce dernier renvoie la liste d’instruments à Elementary Operation Service.

### *1.3. Diagramme d'activité*

Dans la phase de conception, les diagrammes d’activités permettent de mettre l’accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d’une méthode ou le déroulement d’un cas d’utilisation, plus précisément, ils viennent illustrer et consolider la description textuelle des cas d’utilisation. De plus, leur représentation sous forme d’organigrammes les rend facilement intelligibles et beaucoup plus accessibles.

### *1.3.1. Diagramme d'activité modification d’un rapport*

Dans la figure suivante nous présentons les différentes activités du processus de modification d’un rapport :

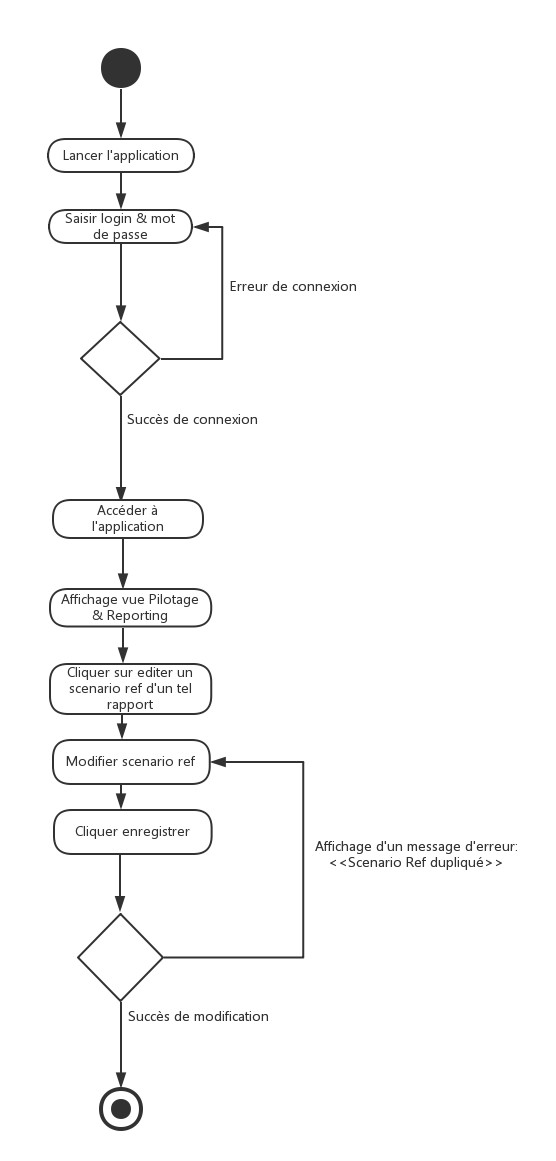


Figure 19 : Diagramme d'activité modification d’un rapport

Dans ce diagramme nous avons illustré les différentes étapes effectuées par l’utilisateur pour modifier la référence d’un scénario correspondant à un tel rapport avec la précision des tests et des vérifications faites tout au long du scenario.

### *1.3.2. Diagramme d'activité téléchargement d’un fichier Excel relatif aux prêts et échéanciers*

Dans la figure suivante nous présentons les différentes activités du processus de téléchargement d’un fichier Excel relatif aux prêts et échéanciers :

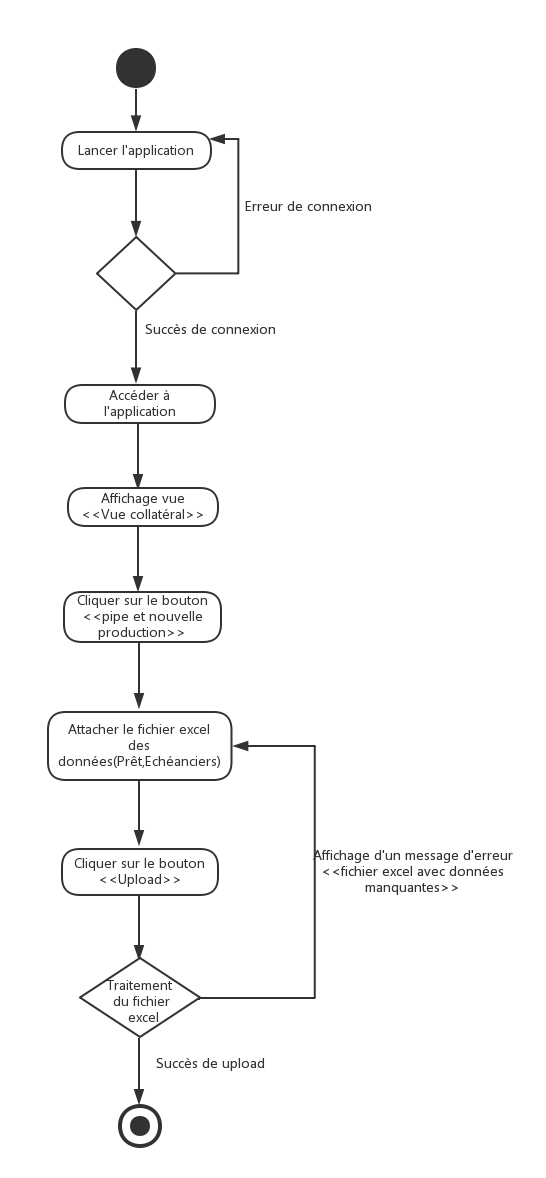


Figure 20 : Diagramme d'activité téléchargement d’un fichier Excel relatif aux prêts et échéanciers

Dans le diagramme ci-dessus nous avons représenté graphiquement le comportement du téléchargement d’un fichier Excel contenant des données relatives aux prêts et échéanciers en annonçant les conditions et les tests si précis qui affirment un bon enchainement de l’action déclenchée par l’utilisateur.

# 2. Conception détaillée

Le travail de conception consiste à déterminer les solutions techniques qui permettent de satisfaire le cahier des charges et donc répondre aux attentes de l’usager. C’est ici que l’ingénieur se base sur son expérience, ainsi que sur les patrons de conception, modèles de solution déjà éprouvés. Le but recherché est d’obtenir des logiciels de grande ampleur qui soient fiables, de qualité, et correspondent aux attentes de l’utilisateur.

### *2.1. Design Pattern*

Un design pattern est une solution à un problème récurrent dans la conception d'applications orientées objet. Un patron de conception décrit alors la solution éprouvée pour résoudre ce problème d’architecture de logiciel.

***Représentation d’un patron de conception***

* Nom : qui permet de l’identifier clairement.
* Problématique : description du problème auquel il répond.
* Solution : description de la solution souvent accompagnée d’un schéma UML.
* Conséquences : les avantages de cette solution.

***Les avantages***

L’utilisation des Design Patterns offre de nombreux avantages. Tout d’abord cela permet

de répondre à un problème de conception grâce à une solution éprouvée et validée par des

experts. Ainsi on gagne en rapidité et en qualité de conception ce qui diminue également les

coûts.  
  
 De plus, les Design Patterns sont réutilisables et permettent de mettre en avant les bonnes

pratiques de conception.

Nous pouvons énumérer ainsi quelques types de designs patterns :

* Singleton
* Spring ( IOC : Inversion of control , DI : Dependency injection )
* Façade
* MVC ( Model-view-controller )
* DAO ( Data Access Object )

Dans la suite on va s’intéresser à l’étude du patron d’architecture MVC :

**Le patron d’architecture MVC** :

Le modèle vue contrôleur est souvent décrit comme simple design pattern(motif de conception) mais c’est plus un architectural pattern(motif d’architecture) qui donne le ton à la forme générale d’une solution logicielle plutôt qu’à une partie restreinte.

Comme l’architecture 3-tiers il possède trois parties qui sont :

***Le Modèle****:* Le modèle définit les données de l’application et les méthodes d’accès. Tous les traitements sont effectués dans cette couche.

***La Vue :*** La vue prend les informations en provenance du modèle qui lui sont passées par le contrôleur et les présente à l’utilisateur.

***Le Contrôleur :*** Le contrôleur répond aux événements de l’utilisateur et commande les actions sur le modèle. Cela peut entrainer une mise à jour de la vue.

La figure ci-dessous représente le patron d’architecture MVC :

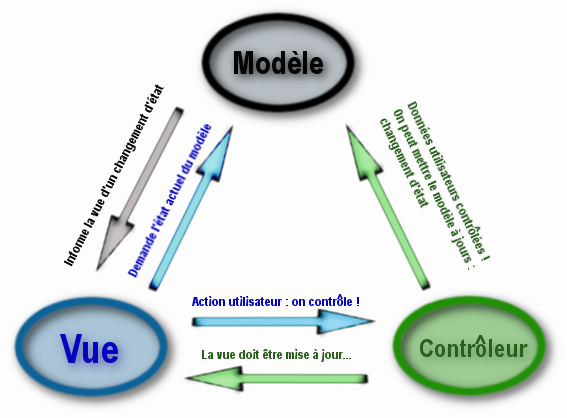
******

Figure 21 : Le pattern Modèle Vue Contrôleur

Passons maintenant à étudier le design pattern IOC ( Inversion de contrôle) :

**Le patron de conception Inversion de contrôle (IOC)** :

L’inversion de contrôle est un motif de conception logicielle commun à tous les frameworks, devenu populaire avec l’adoption des conteneurs dits légers. Cette notion fait partie des principes de la programmation orientée aspect.

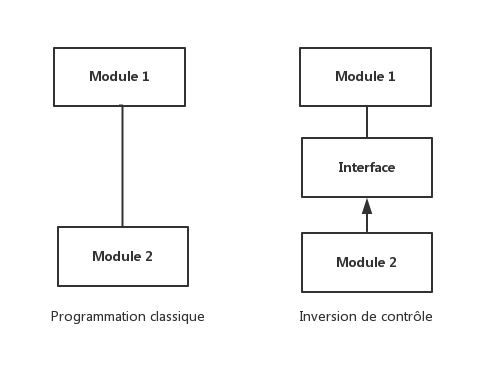


Figure 22 : Le principe d’inversion de contrôle

L’idée principale est que les modules de hauts et bas niveaux doivent dépendre d’interfaces, et non les modules dépendre les uns des autres.les modules dérivent des interfaces. L’inversion de contrôle les rend donc indépendants entre eux.

## Conclusion

Tout au long de ce chapitre, nous avons étudié la conception de notre application, à travers les diagrammes de séquence objets et le diagramme de classes qui s’avèrent les plus importants pour la réalisation du projet. Le résultant de l’implémentation de ces diagrammes fera l’objet du prochain chapitre.

Ainsi, tout au long du chapitre suivant, nous présenterons le travail réalisé ainsi que les environnements de développement utilisés.

# Chapitre 4 : Réalisation

## Introduction

Après avoir donné l’architecture de notre application ainsi que la conception des différents modules, nous abordons à présent la partie réalisation de la solution adoptée. Nous justifions d’abord nos choix portés sur les techniques de développement utilisées, puis nous décrivons l’environnement du travail aussi bien matériel que logiciel, et enfin nous exposerons les interfaces les plus importantes de notre projet.

# 1. Environnement matériel

Les outils matériels utilisés dans notre application sont :

|  |  |  |  |
| --- | --- | --- | --- |
| Equipement | Processeur | Mémoire | Système d’exploitation |
| Poste fixe HP | Processeur Intel® CORE™ i3 | Mémoire RAM de 4.00 GB | Windows 7 Entreprise (64bit) |

# 2. Environnement logiciel

Afin de préparer l’environnement logiciel de travail nous avons disposé de :

IDE « Integrated Development Environment »: Eclipse Luna Release 4.4.0

**Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n’importe quel langage de programmation. IL est principalement écrit en java. Ce langage est utilisé pour écrire** des extensions, grâce à des bibliothèques bien spécifiques.

***SVN «Subversion» : Logiciel de gestion de versions***

SVN est un logiciel conçu pour le contrôle de versions. Il s’agit d’un logiciel qui enregistre tous les états d’une arborescence au fil du temps. C’est de là que vient le terme de « version » : le logiciel surveille les différentes versions d’un répertoire.

Le développement d’un logiciel est composé de multiples modifications de fichiers au fil du temps ; SVN permet d’enregistrer tous ces changements faits au code source, ou pouvoir revenir au code tel qu’il était il y a 5 mois.

Framework Hibernate 4.2.1

Hibernate est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d’une application.Hibernate permet donc de représenter une base de données en objets java et vice versa.

Hibernate facilite la persistence et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données.

Spring ORM

Propose un support pour des outils de type ORM(JPA,Hibernate…).

Framework Spring Web-Servlet

Framework pour le développement d'applications qui met en œuvre le motif de conception MVC. Ceci permet entre autres de choisir la technologie utilisée pour la vue (JSP).

Framework Spring Test

Framework qui propose un support pour les tests automatisés avec un support de JUnit.

### *2.1. Outil de conception*

Nous avons utilisés, pour la spécification et la conception de ce travail, la méthodologie UML (Unified Modeling Language) qui est un langage de modélisation qui permet de représenter les différentes vues du système : les vues statiques et les vues dynamiques. Les vues statiques représentent le système physiquement tandis que les vues dynamiques montrent le fonctionnement du système. Chaque vue est constituée de plusieurs diagrammes. Ce choix est influencé par les avantages que cette méthode présente comparée aux autres méthodes de conception.

# 3. Aperçu sur le travail réalisé

En vue de présenter le travail accompli, nous allons dans ce qui suit exposer différentes captures d’écran montrant la réalisation de notre travail.

### *3.1. Authentification*

Tout utilisateur souhaitant accéder à l'application doit se connecter en saisissant son login / mot de passe Sesame.

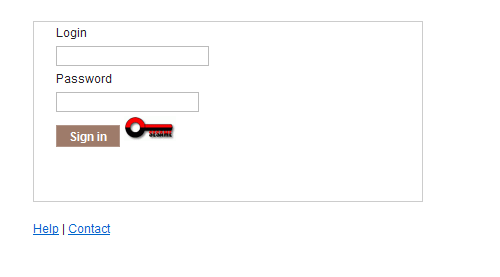


Figure 23 : Fenêtre authentification

### *3.2. Rôle administrateur*

Après authentification avec un compte administrateur, une redirection vers la page d'accueil de l'administrateur s'effectue.

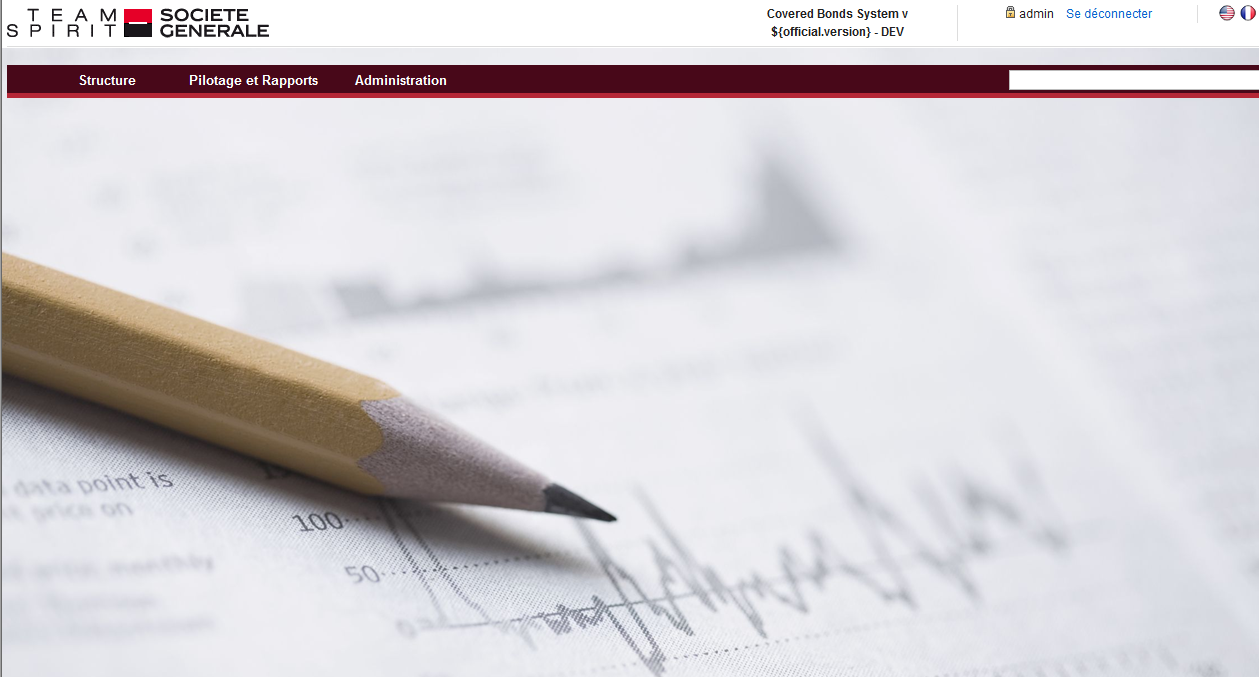


Figure 24 : Fenêtre d’accueil de l’administrateur

Le rôle de l'administrateur au niveau de l'application consiste à la gestion des comptes.

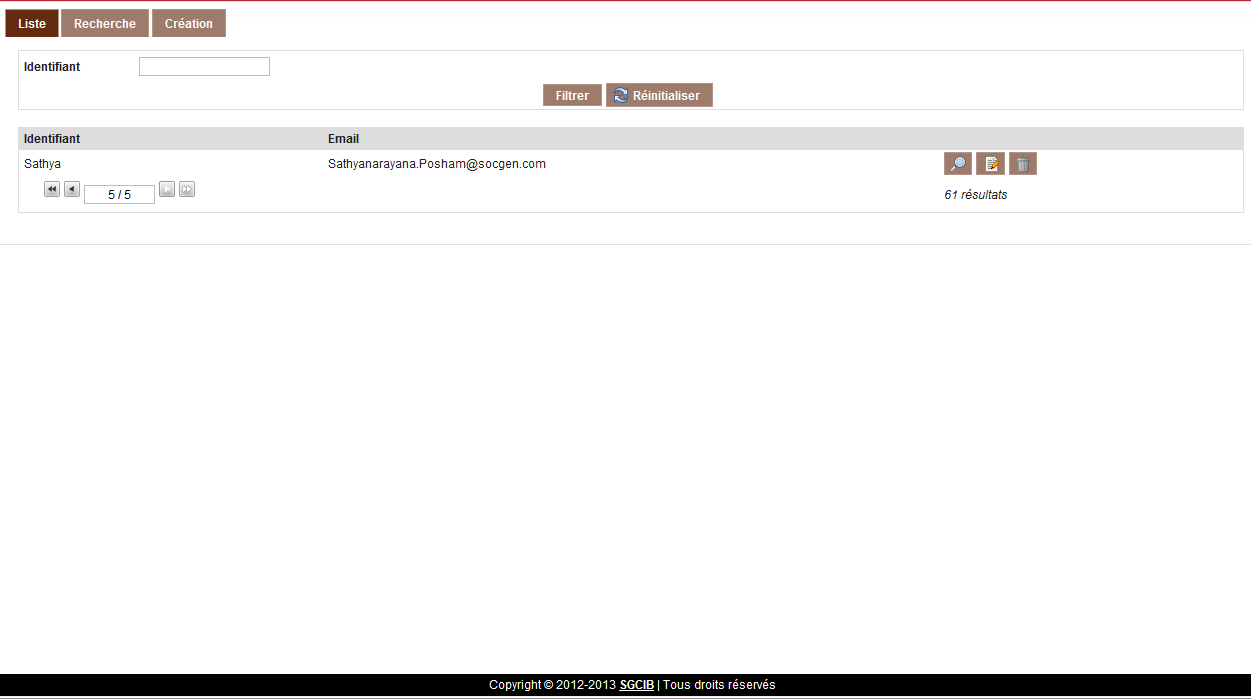


Figure 25 : Fenêtre gestion des comptes

Cette fenêtre permet de visualiser la liste des utilisateurs de l'application. Elle permet aussi de sélectionner un compte afin de modifier son rôle, de créer un nouveau compte, supprimer un compte ou rechercher un nom de compte qui existe déjà.

La figure suivante illustre la fenêtre de création d'un compte.

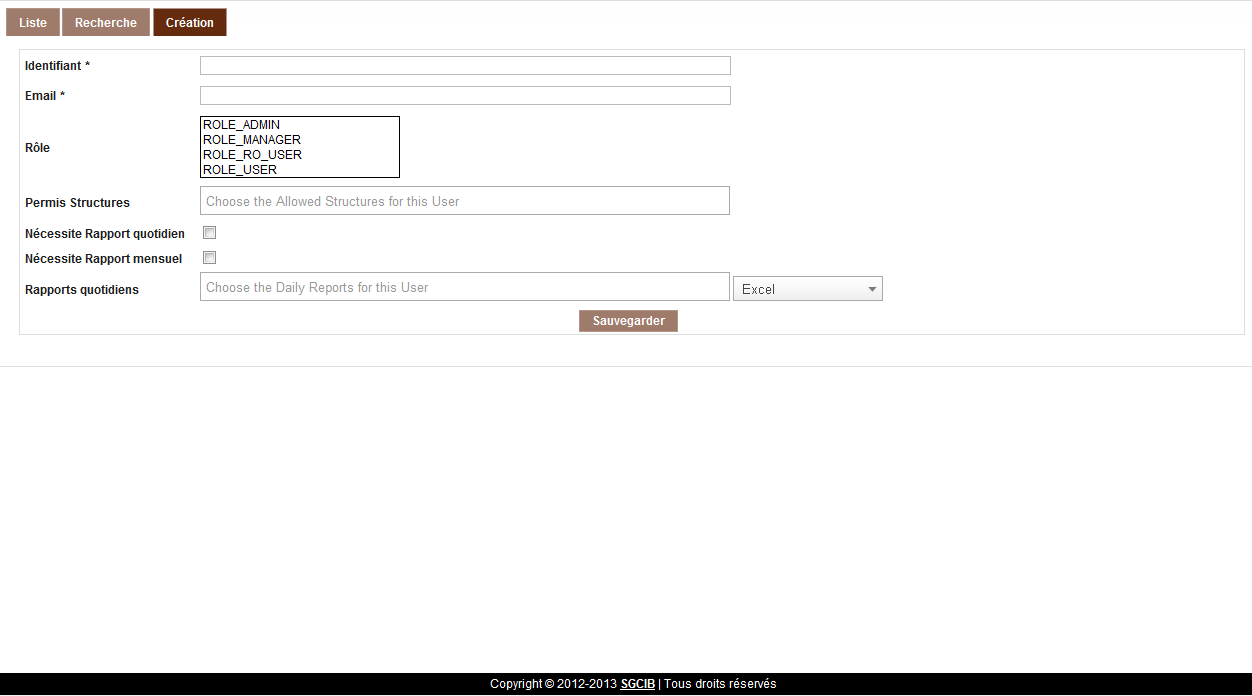


Figure 26 : Fenêtre ajout d’un compte

L'administrateur peut aussi modifier les données d’un compte. La figure suivante présente la fenêtre de modification des données d’un compte bien déterminé.

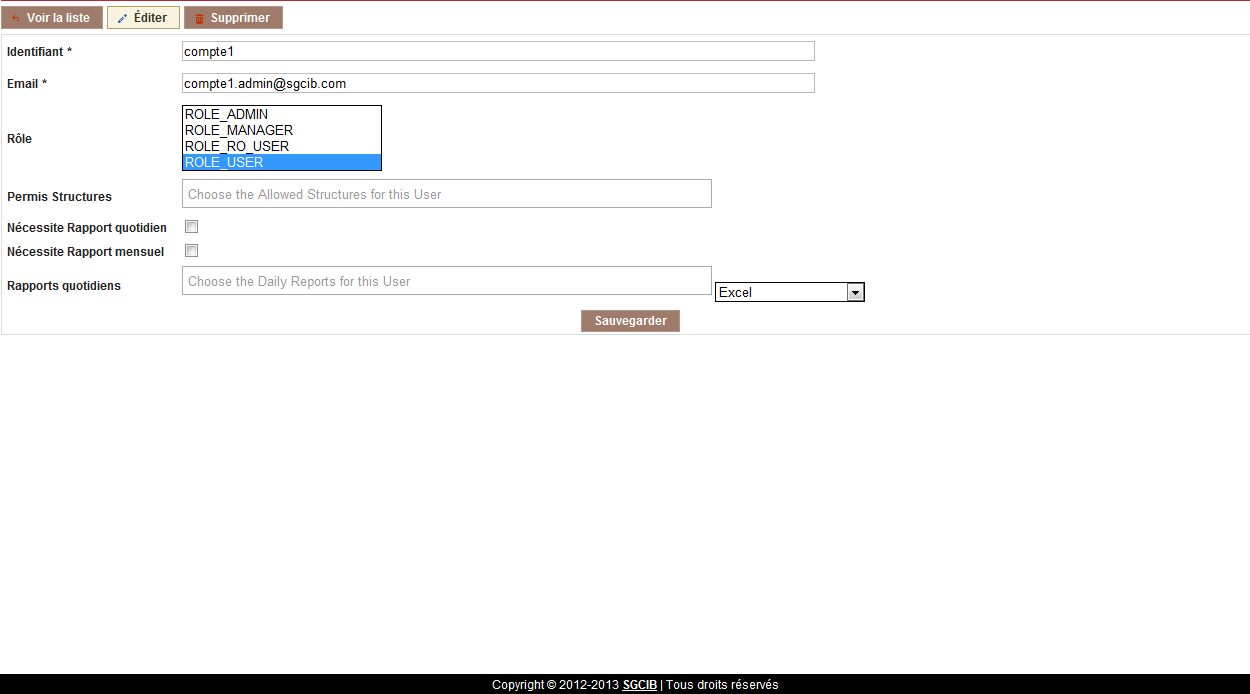


Figure 27 : Fenêtre modification d'un compte

### *3.3. Rôle utilisateur*

Après authentification avec un compte utilisateur, ce dernier aura comme privilège :

* La gestion des instruments dans la vue «Vue sociale».
* La gestion des actifs et des passifs dans la vue «Actifs apportés en garantie».
* La gestion des garanties dans la vue «Vue collatéral».
* La génération des rapports dans la vue «Pilotage et Rapports».
* La gestion des référentiels sous la rubrique «Référentiels».

Tout d'abord, comme toute personne interne à la SG, un utilisateur de l’application CBS aura accès à la rubrique de gestion des instruments (prêts, échanges, obligations).

La figure suivante présente la vue sociale/social view permettant d’accéder à l’ensemble des produits financiers.

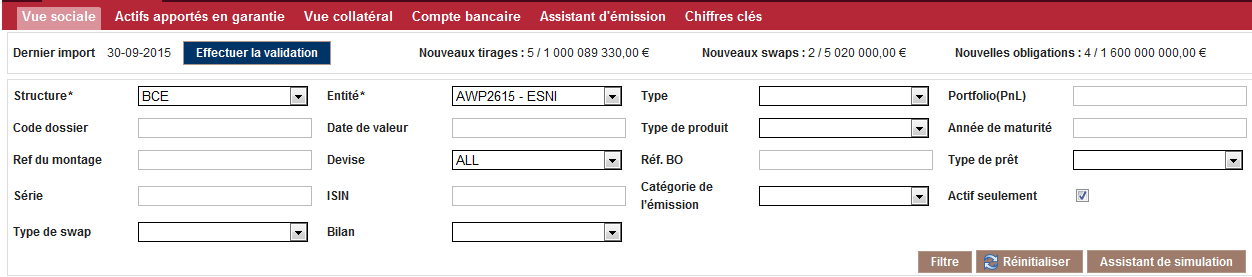


Figure 28 : Fenêtre gestion des instruments

Une série de filtres permet de retrouver rapidement chaque instrument ou groupe d’instruments.

Les instruments nouvellement importés doivent faire l’objet d’une validation.

* **Validation des nouveaux instruments**

Pour effectuer la validation des nouveaux instruments, l’utilisateur clique sur le bouton « Effectuer la validation ». D:\mbenabda062215\Stage\Docs_Rapport\socialView2.png

L’écran affiche alors une fiche par instrument.

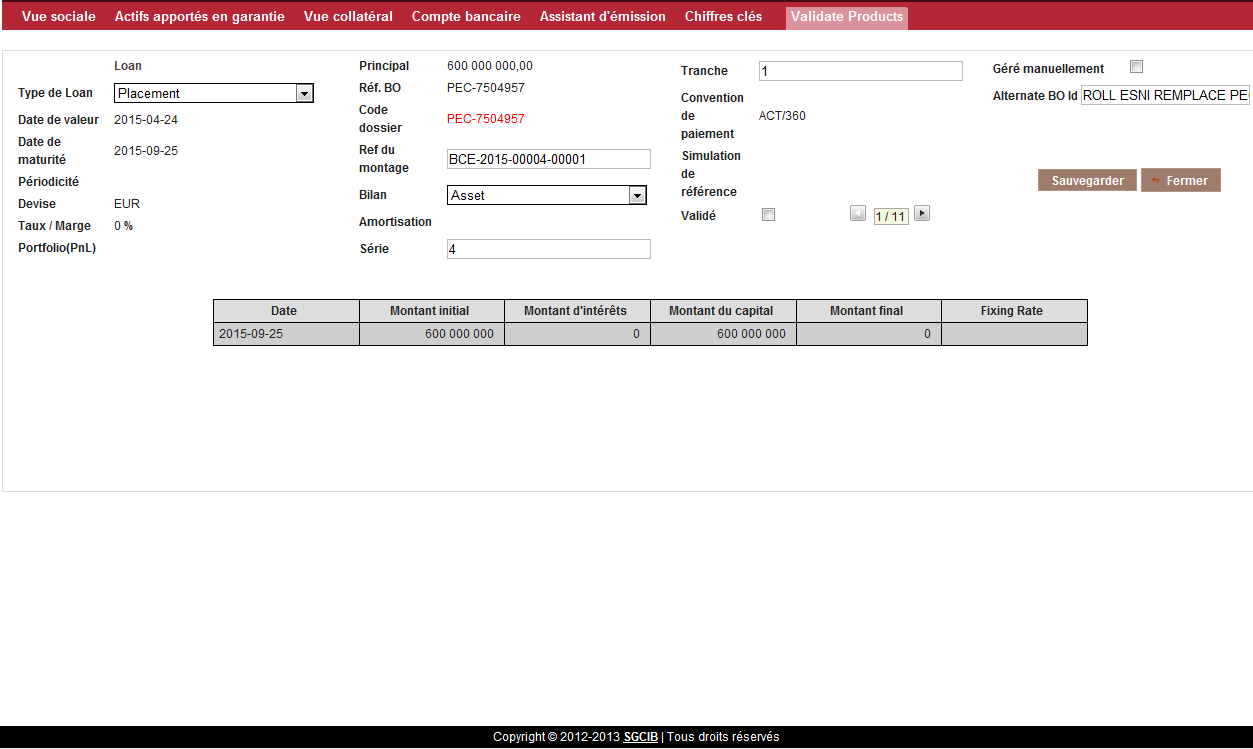


Figure 29 : Fenêtre validation d’un nouveau instrument

Pour valider l’instrument, l’utilisateur coche la case « Validé ».

* **Filtres et vue maître/détaillée**

CBS dispose d’un ensemble de filtres permettant de retrouver rapidement chaque instrument ou groupe d’instruments.

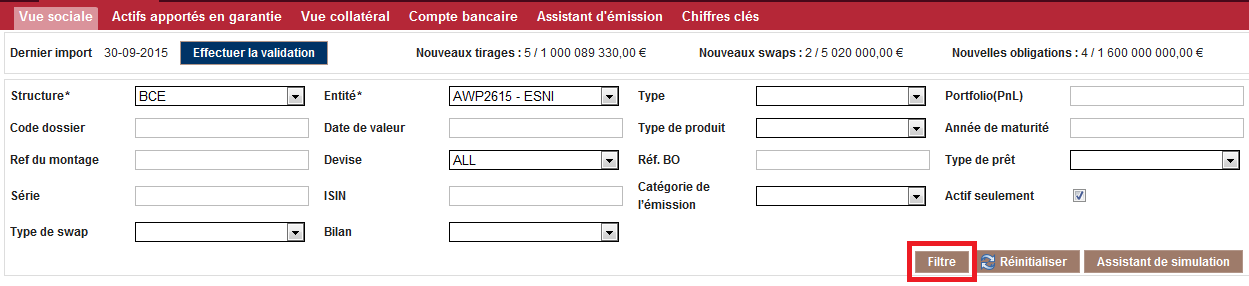
****

Figure 30 : Fenêtre filtre d’instruments

Pour filtrer l’ensemble d’instruments, l’utilisateur clique sur le bouton « Filtre ». D:\mbenabda062215\Stage\Docs_Rapport\BtnFiltre.png

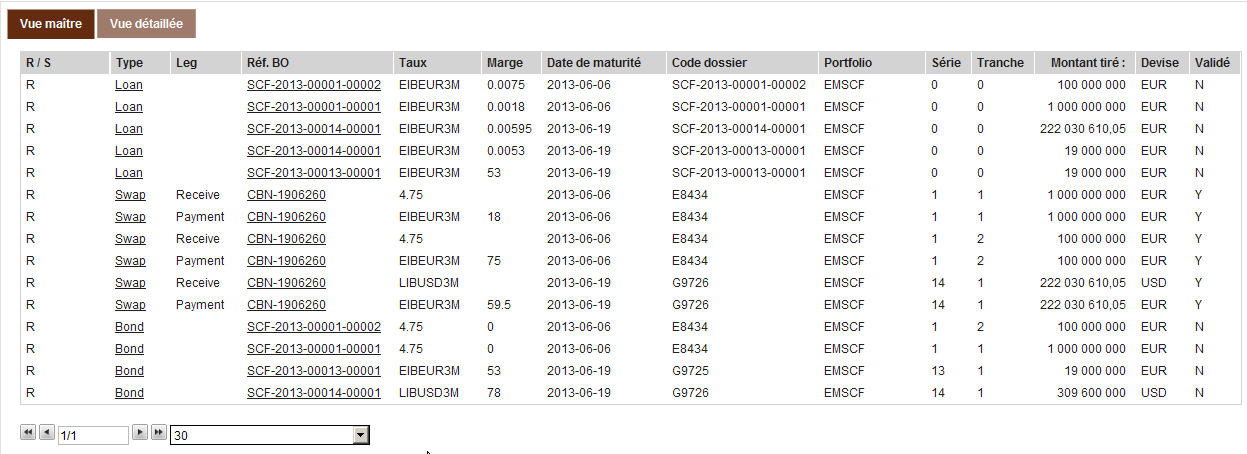


Figure 31 : Fenêtre vue maître des instruments filtrés

Chaque instrument dispose d’une vue donnant les caractéristiques détaillés.



Figure 32 : Fenêtre vue détaillée d’un instrument

Un mode édition permet de mettre à jour les caractéristiques d’un tel instrument en cliquant sur le bouton «Editer». D:\mbenabda062215\Stage\Docs_Rapport\editer.png

Tout utilisateur interne à la SG peut aussi consulter la rubrique «Actifs apportés en garantie».

La figure suivante présente la vue Actifs apportés en garantie :

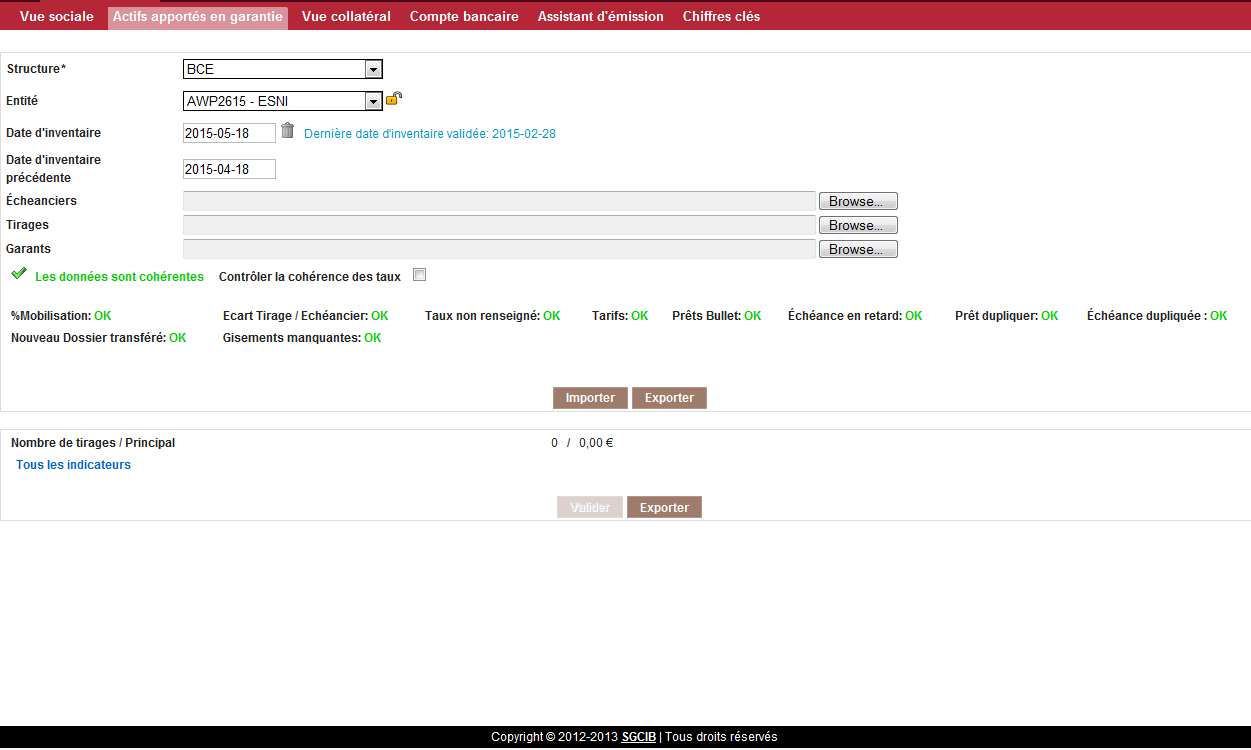


Figure 33 : Fenêtre rubrique actifs apportés en garantie

* **Indicateur de qualité des données**

A l’issue de l’import, l’application calcule un indicateur de qualité Global en comparant

les Echéanciers aux tirages. En cas d’écarts, un message en rouge (ci-dessous) est affiché à l’écran et permet d’accéder à la liste des prêts en écarts.

Les contrôles effectués pour aboutir à cet indicateur de qualité portent sur les points

suivants :

* Cohérence tirages/ échéanciers
* Ecart sur le taux de mobilisation
* Taux variables incohérents

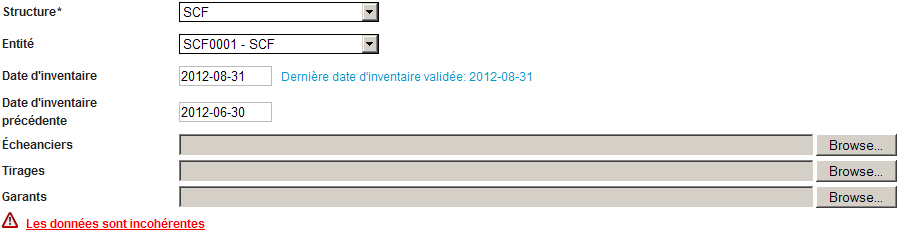


Figure 34 : Fenêtre affichage message d’erreur sur l’indicateur de données

* **Actions de data cleaning**

En cliquant sur le message d’erreur, un écran permet de réaliser les actions suivantes :

* + - Ajout d'une échéance
    - Suppression d'une échéance
    - Correction du taux de mobilisation
    - Correction du montant de tirage
    - Correction du type de taux

Ayant le rôle d’un utilisateur appartenant à la SG, cette personne a la possibilité d’y accéder à la gestion des garanties sous la rubrique «Vue collatéral».

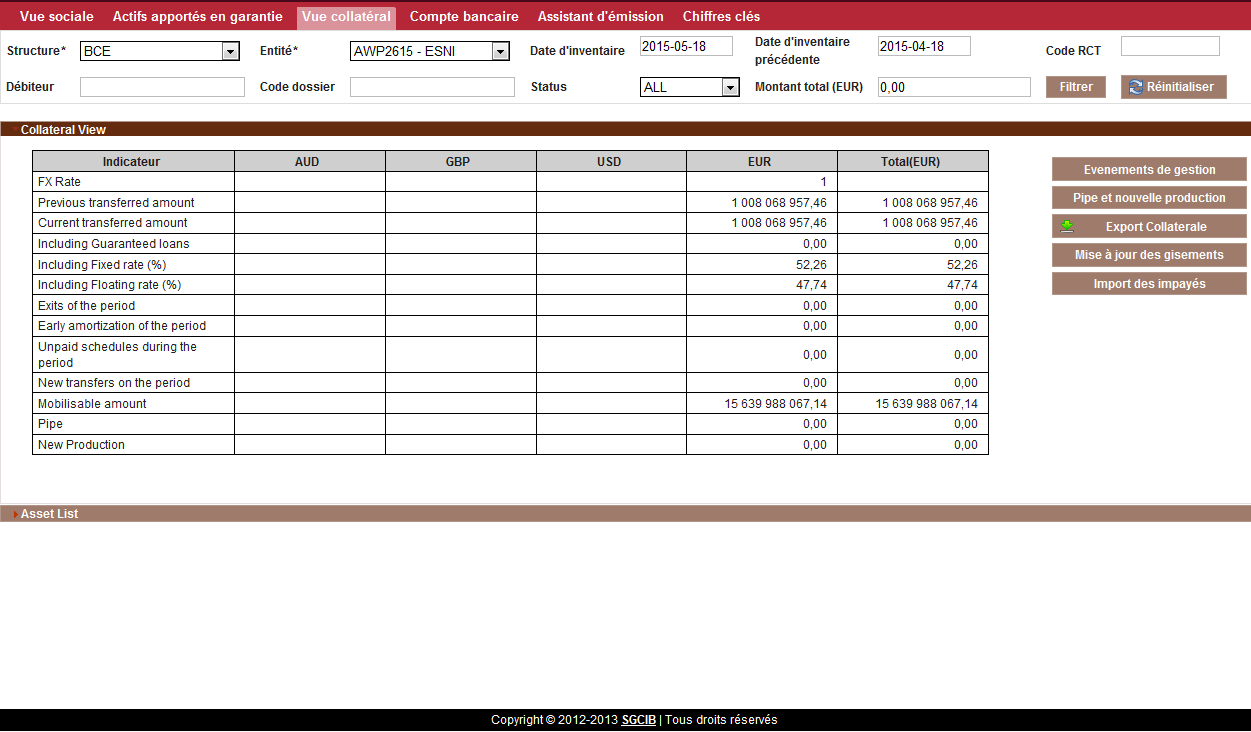


Figure 35 : Fenêtre gestion des garanties

L’écran Vue collatéral permet d’accéder à l’ensemble des prêts vivants à une date donnée.

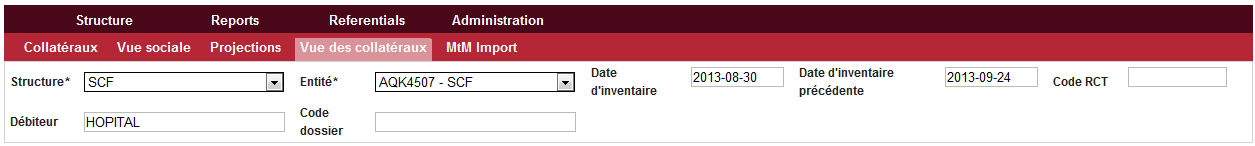


Figure 36 : Fenêtre exemple de saisie des données

Exemple : les filtres suivants permettent de filtrer sur les prêts présents au 30-08-2013 et portant sur des débiteurs comportant la mention « HOPITAL ».

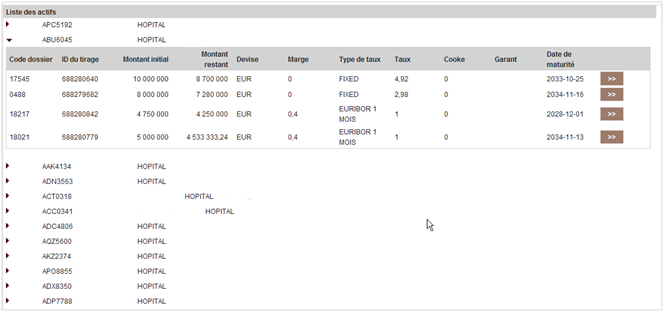
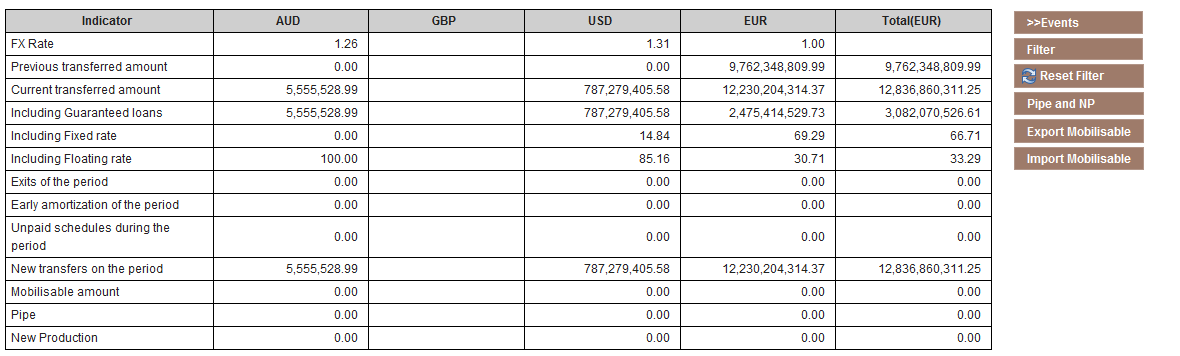


Figure 37 : Fenêtre liste des actifs



**Figure 38 : Fenêtre filtre des actifs**

Le tableau ci-dessus est obtenu en sélectionnant le bouton “Filtre” .Ce dernier permet de cibler la recherche de prêts vivants selon un ensemble de critères saisis par l’utilisateur (Structure, entité, date…). Les montants sont affichés dans leur devise d’origine et convertis en EURO selon le taux de change correspondant à la date sélectionnée.

Pour pouvoir générer un rapport, l’utilisateur peut accéder dans notre application CBS à la rubrique «Pilotage et Rapports».

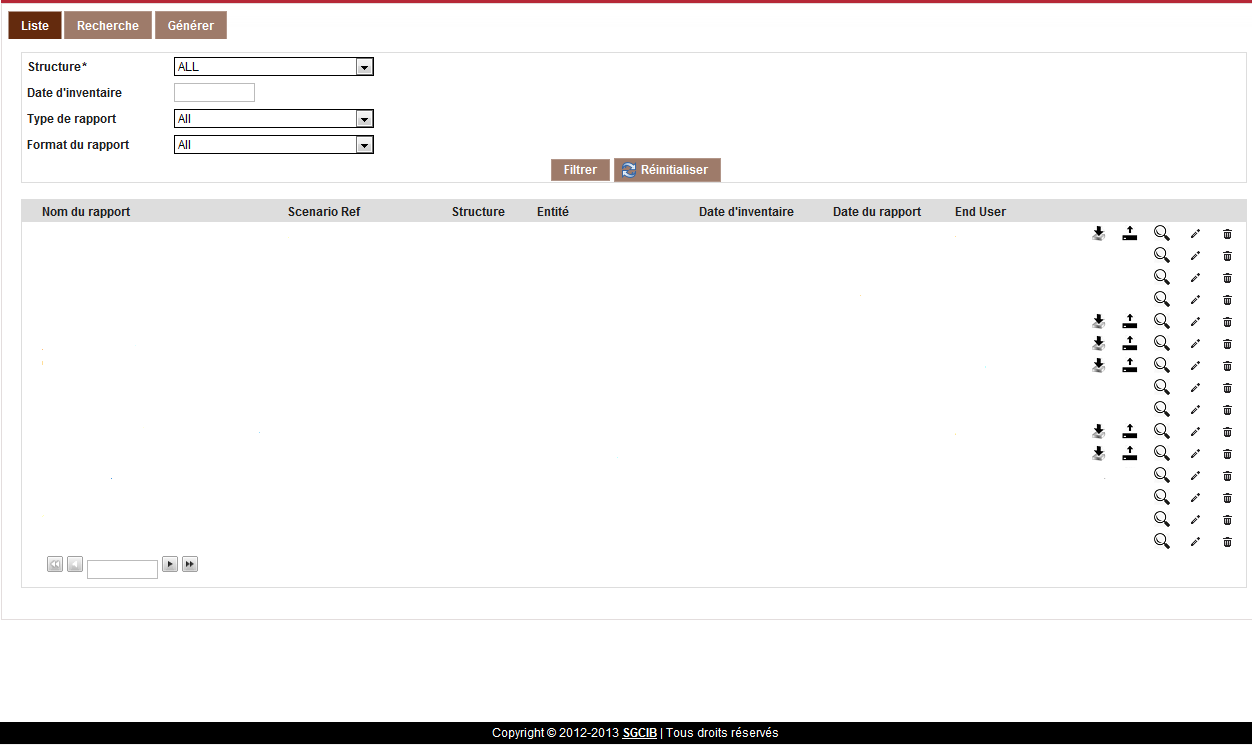


Figure 39 : Fenêtre gestion des rapports

Cette fenêtre permet de visualiser la liste des rapports à générer. Elle permet aussi de générer un rapport, d’éditer le champ« scenario Ref» d’un tel rapport, supprimer un rapport ou rechercher un nom de rapport qui existe déjà.

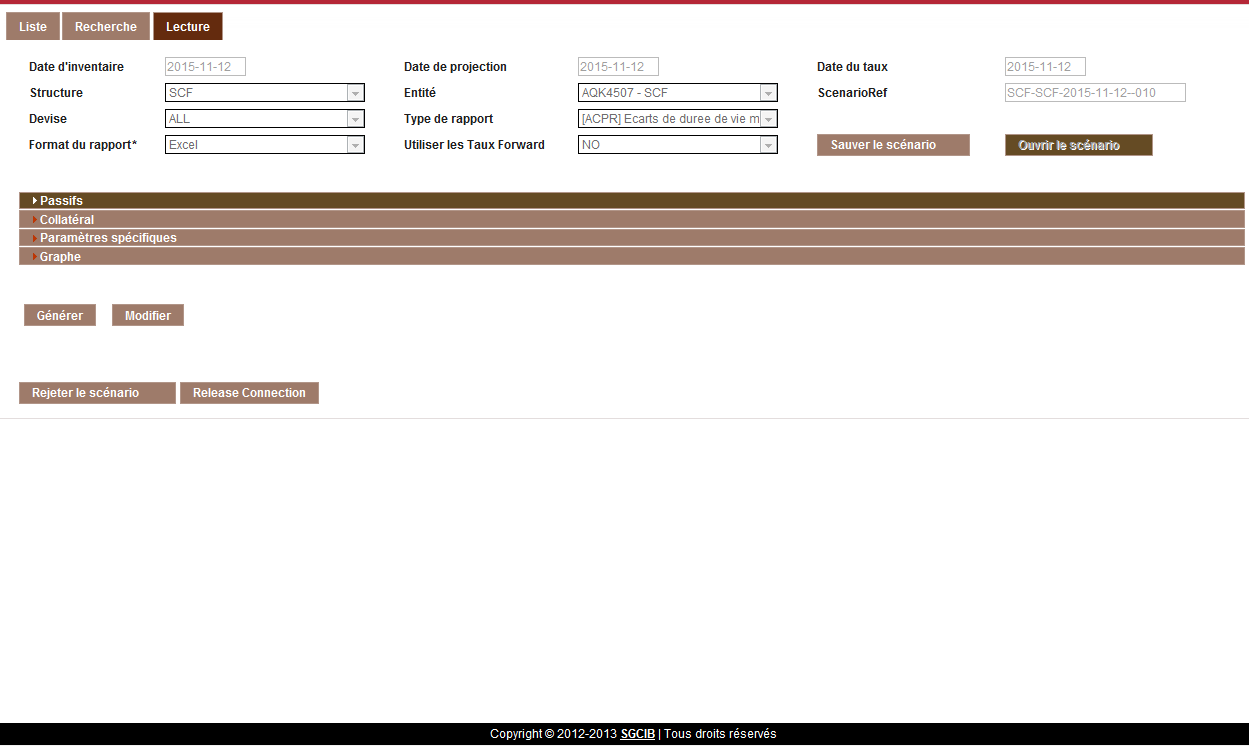


Figure 40 : Fenêtre génération des rapports

La figure ci-dessus est obtenue en sélectionnant le bouton “View” D:\mbenabda062215\Stage\Docs_Rapport\rapport.png correspondant à un

tel rapport.

Le dernier rôle que peut y avoir l’utilisateur dans notre application c’est la gestion des

référentiels (structures, taux indexé, référentiel-tiers..).

* **Structures**

L’utilisateur peut lister, créer, supprimer ou rechercher une structure.

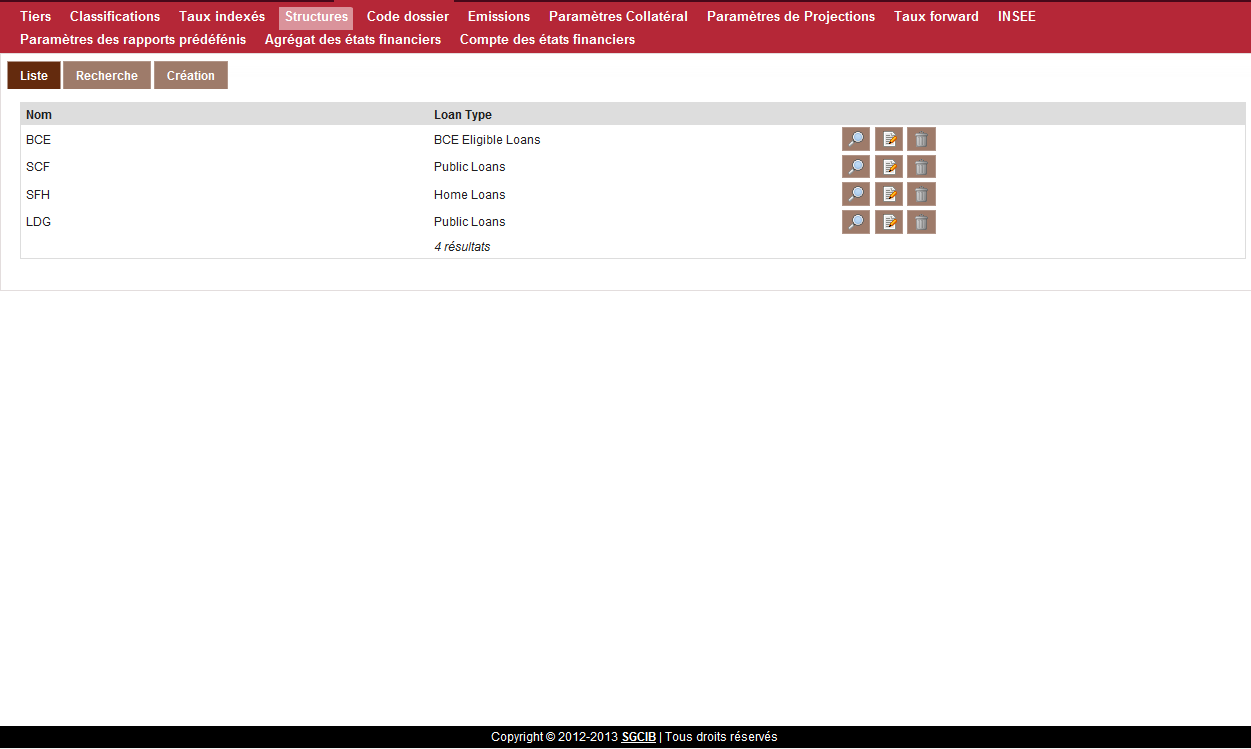
****

Figure 41 : Fenêtre gestion des structures

L’édition de la structure permet de rattacher une ou plusieurs entités à la structure.

La barre d’outils suivante  permet de :

*  consulter la structure
*  Editer la structure
* Supprimer la structure

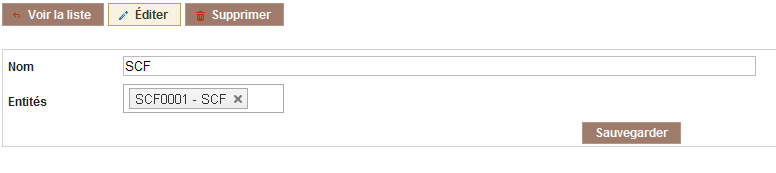


Figure 42 : Fenêtre modification d’une structure

En cliquant dans « Entités », l’utilisateur peut ajouter une entité.

* **Taux indexé**

L’utilisateur peut lister, créer, supprimer ou rechercher une structure.

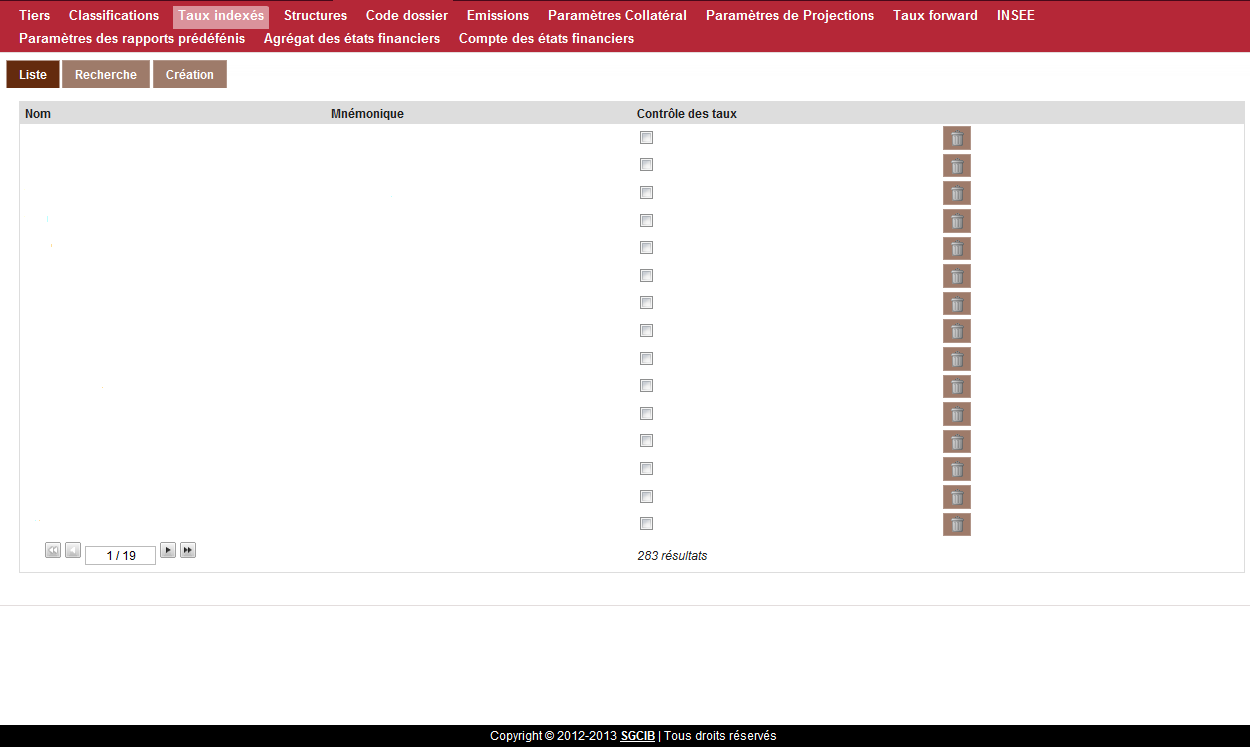
****

Figure 43 : Fenêtre gestion des taux indexés

* **Tiers Référentiels**

L’utilisateur peut lister, créer, supprimer ou rechercher un tiers.

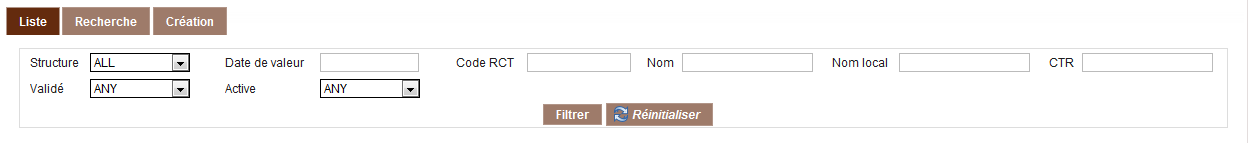


Figure 44 : Fenêtre filtre d’un tiers

Pour filtrer un tiers, l’utilisateur dispose pour cela de 8 filtres :

* Structure
* Date de valeur
* Code RCT
* Nom
* Nom local
* CTR
* Validation
* Activation

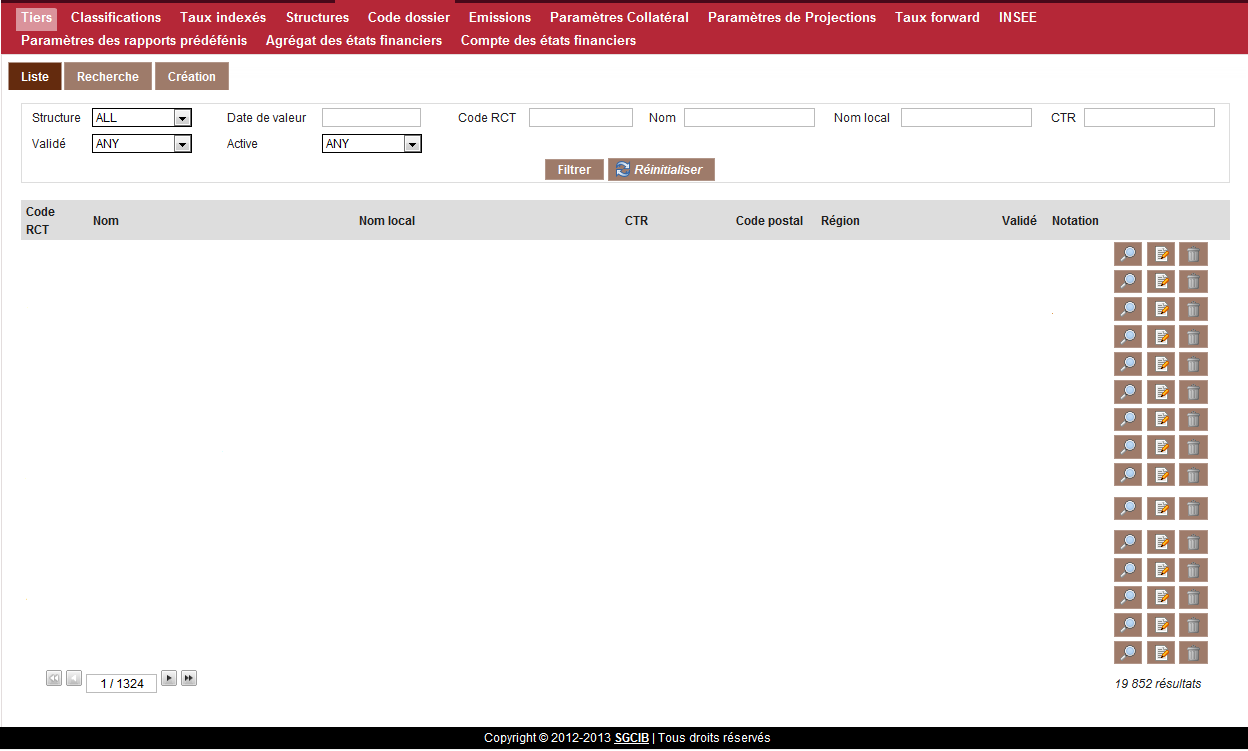
****

Figure 45 : Fenêtre liste des tiers filtrés

## Conclusion

Dans ce chapitre, nous avons détaillé la partie réalisation du projet. Nous avons aussi précisé notre choix technologique et notre environnement de travail. Ensuite, nous avons présenté les principales fonctionnalités qu'offre notre application. A présent, nous passerons dans la partie suivante à la conclusion générale et aux perspectives du projet.

# Conclusion générale et perspectives

Le But de notre projet est la conception et développement d'une application de gestion des obligations sécurisées.

Ce projet est une solution propriétaire pour la Société Générale pour la gestion de ses projets dans le cadre d'une méthode agile KANBAN.

Dans ce rapport nous avons essayé de présenter tout ce qui s'avère indispensable pour décrire clairement toutes les étapes du projet. Ainsi nous avons commencé par une étude du contexte général du sujet, ce qui nous a permis de dégager la problématique ayant engendré le besoin au lancement de ce sujet. Puis nous avons spécifié les besoins fonctionnels et non fonctionnels auxquels doit répondre notre application. Ensuite, nous avons présenté une conception de la solution à travers la représentation des vues statiques et dynamiques. Une fois la conception est terminée, nous avons fait le choix des technologies à utiliser. Enfin, nous avons exposé le travail effectué à travers des captures d'écran.

Pour conclure, nous estimons avoir satisfait la plupart des besoins initialement fixés.

# Glossaire

**Applications et termes standards**

SGCIB  Société Générale Corporate and Investment Banking

SG Société Générale

ITEC Information Technology Department

FCC Financing and Client Coverage Technology

OSD Origination, Structurating and Distribution

CBS Covered Bonds System

AOP Architecte Opérationnel de Production: support technique

SCF  Sociétés de Crédit Foncer

SFH Sociétés de Financement de l’Habitat

Sesame service d’authentification

BOR Rapports configurés pour l’application CBS dans le serveur Business Object

UAT User Acceptance Tests

HOM La phase d’homologation d’un projet : étape primordial consiste à effectuer un certain nombre de tests pour vérifier le niveau de qualité de l’application livrée, sa conformité aux spécifications fonctionnelles détaillées ainsi qu’aux exigences techniques.

PROD La phase recette et mise en Production : étape basique consiste à valider le bon fonctionnement de l’application.

**Acronymes et définitions techniques**

J2EE Java 2 Enterprise Edition

Hibernate JPA Implementation

ORACLE c’est un SGBD (Système de Gestion de Bases de Données)

Spring c’est un framework libre pour construire et définir l’infrastructure d’une application java.

AOP Aspect Oriented Programming

IOC Inversion Of Control

DI Dependency Injection

JDBC Java Datababase Connectivity

JPA Java Persistence API

POJO Plain old java object

JSP Java Server Pages

JavaScript langage orienté objet conçu pour la programmation de scripts.

Jquery est une [bibliothèque](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_logicielle) [JavaScript](https://fr.wikipedia.org/wiki/JavaScript) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) créée pour faciliter l'écriture de [scripts côté client](https://fr.wikipedia.org/wiki/Client_Side_Scripting) dans le code [HTML](https://fr.wikipedia.org/wiki/HTML) des [pages web](https://fr.wikipedia.org/wiki/Page_web).

AJAX Asynchronous JavaScript and XML: permet de construire des [applications Web](https://fr.wikipedia.org/wiki/Application_Web) et des [sites web dynamiques](https://fr.wikipedia.org/wiki/Page_Web_dynamique) [interactifs](https://fr.wikipedia.org/wiki/Interactions_homme-machine) coté client.

ASPOSE c’est un fournisseur de composants pour créer, modifier, l'exportation et convertir des documents Office comme Word, Excel, PDF…

JUnit c’est un framework de test unitaire pour le langage de programmation Java.

HTTPS HyperText Transfer Protocol Secure

# Bibliographie

[1] Intranet « My Société Générale »

[2] Présentation de l’application CBS, document interne, 7 janvier 2014

[3] Agile Maturity Model, document interne

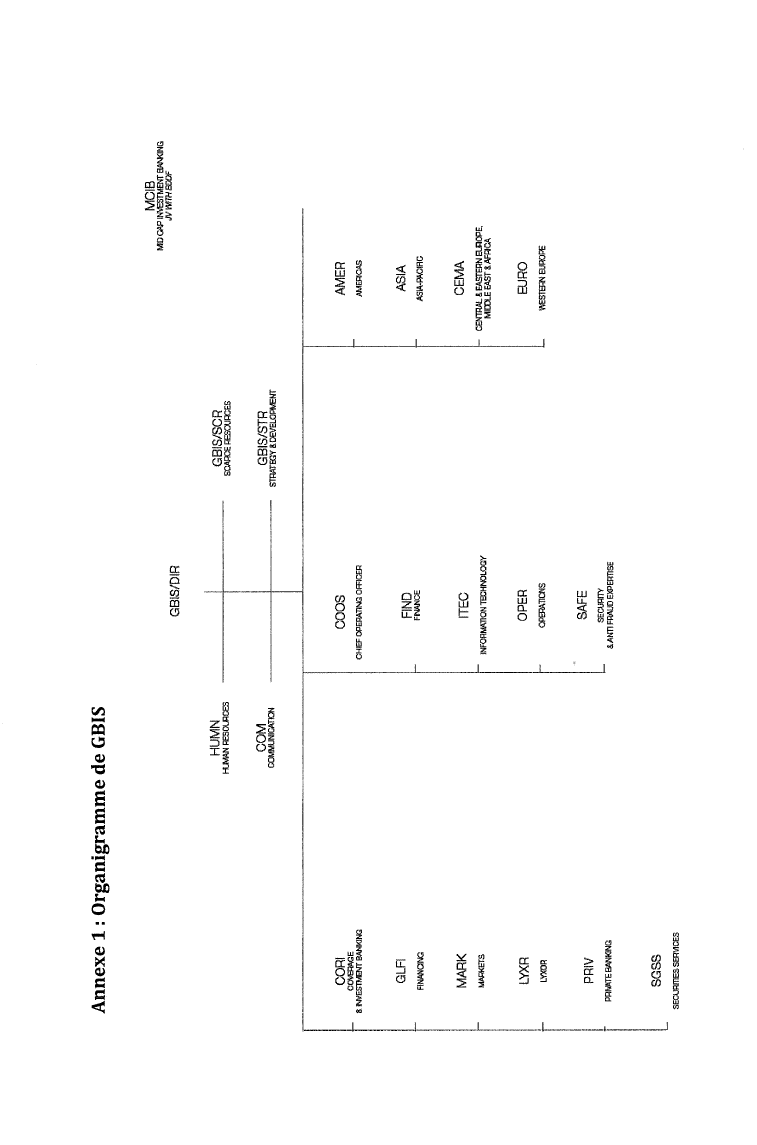
[http://spring.io](http://spring.io/)

http://spring.developpez.com/cours/

<http://www.lifl.fr/~dumoulin/enseign/2012-2013/ipint/8.spring/Cours%20Spring%20MVC.pdf>

<http://igm.univ-mlv.fr/~dr/XPOSE2010/guicespring/di_spring.html>

<http://www.cienum.fr/sites-internet-mobiles/projets/methodologie-de-projets/kanban>



71



*Adresse : Z.I Chotrana II - B.P. 160 - 2083 Pole Technologique - El Ghazela - Ariana*

*Tél : (+216) 70 68 56 85 Fax : (+216) 70 68 54 54*

*Email* [*: contact@esprit.tn*](mailto::%20contact@esprit.tn)