

Réf : Ing-GI-2016-03

# Rapport de Projet de Fin d'Études

Pour obtenir le

**Diplôme d'Ingénieur en Génie Informatique**

**Option : GLID**

Présenté et soutenu publiquement le 23 juin 2016

Par

**Zeineb HADDAD**

## Développement d'une application de gestion des tableaux de bord en Java EE

### Composition du jury

Madame Bisma FAYECH	Président
Madame Afef KACEM	Rapporteur
Monsieur Mohamed Selim BASSI	Encadrant Entreprise
Monsieur Ramzi FARHAT	Encadrant ENSIT
Monsieur Mohamed Ali KHENISSI	Encadrant ENSIT

Année universitaire : 2015-2016

# Dédicace

## *A mes très chers parents Mohamed & Najoua*

*Je* dédie ce modeste travail et je veux énoncer les vœux les plus ardents pour la conservation de leur santé et de leur vie pour l'éducation qu'ils m'ont prodiguée, pour les sacrifices qu'ils ont consentis à mon égard, pour le sens du devoir qu'ils m'ont enseignés depuis mon enfance, pour leur générosité et leur présence permanente.

## *A mes chers frères et sœurs*

Chers frères (Zied, Zouhair) et chères sœurs (Khaoula, Fatma) je veux vous exprimer ma profonde gratitude pour leurs affections, pour leurs compréhensions, pour leurs encouragements incessants, et la motivation de mon effort.

## *A mes chers amis*

*Je* dédie aussi ce travail à tous mes amis et spécialement (Ameni, Chourouk, Dhahbi, Hamza, Med Charfeddine, Safa, Shayma, Sofien, Souhir et Raouaa), en souvenir des bons moments que nous avons passés ensemble, pour leur soutien continu, leur précieuse et leur amour.

## *A mes professeurs*

*P*our leur aide et soutien pendant toute la période de ma formation.

Zeineb

## **Remerciements**

Le travail présenté dans ce mémoire a été effectué dans le cadre de la préparation du diplôme d'ingénieur à l'Ecole Nationale Supérieure d'Ingénieur de Tunis (ENSIT).

Au terme de ce projet, j'adresse mes vifs remerciements à mes encadreurs académiques Mr. Ramzi FARHAT et Mr. Mohamed Ali KHENISSI pour leurs encadrements, leurs soutiens et leurs disponibilités. Leurs conseils, leurs suggestions de lecture, leurs commentaires, leurs corrections et leurs qualités scientifiques ont été très précieux à bien ce travail.

Je profite de cette tribune pour remercier également Mr. Selim BESSI, ingénieur au sein de BFI, qui a cru en moi et qui n'a cessé de me faire profiter ses précieux conseils et remarques qui m'ont permis de réaliser ce travail dans les meilleures conditions.

Enfin, je tiens également à remercier et exprimer mon profond respect aux membres de jury d'avoir accepté de juger ce travail.

C'est ainsi et avec une immense fierté que j'adresse mes remerciements les plus distingués à tous mes professeurs.

# Sommaire

Introduction Générale.....	1
Chapitre 1 : Cadre Général.....	4
Introduction : .....	4
1.    Présentation de l'organisme d'accueil : .....	4
1.1.    Organisme d'accueil : .....	4
1.2.    Activités et Services : .....	4
1.3.    Solutions et Références : .....	4
2.    Contexte du projet : .....	4
2.1.    Etude de l'existant : .....	5
2.2.    Critique de l'existant : .....	5
2.3.    Synthèse : .....	6
3.    Mission confiée : .....	6
4.    Démarche de réalisation du projet : .....	7
4.1.    Méthodologie du projet : .....	7
4.2.    Cycle de vie du projet : .....	8
4.3.    Planification du projet : .....	9
Conclusion : .....	10
Chapitre 2 : Etat de l'Art .....	12
Introduction : .....	12
1.    Notion de l'informatique décisionnelle : .....	12
1.1.    Présentation du concept BI : .....	12
1.2.    Technologie ETL : .....	13
1.3.    Etude des solutions d'intégration et de gestion des données .....	13
1.3.1.    Présentation Talend Open Studio : .....	13
1.3.2.    Présentation Pentaho : .....	14

1.3.3. Critique des solutions présentées :.....	15
2. Etude de l'existant :.....	16
2.1. La plate-forme QlikView :.....	16
2.1.1. Présentation du logiciel QlikView :.....	16
2.1.2. Principe de QlikView : .....	16
2.1.3. Fonctionnalités offertes par QlikView : .....	17
2.2. La plate-forme SpagoBI :.....	18
2.2.1. Présentation de SpagoBI :.....	18
2.2.2. Architecture de SpagoBI : .....	18
2.2.3. Fonctionnalités offertes par SpagoBI : .....	20
2.3. Etude comparative : .....	22
Conclusion :.....	23
Chapitre 3 : Spécification et Analyse .....	25
Introduction :.....	25
1. Spécification des besoins : .....	25
1.1. Besoins fonctionnels :.....	25
1.1.1. Besoin fonctionnel 1 : Configurer une base des données :.....	25
1.1.2. Besoin fonctionnel 2 : Manipuler les données provenant des sources externes : .....	25
1.1.3. Besoin fonctionnel 3 : Gérer les comptes et les droits d'accès : .....	26
1.1.4. Besoin fonctionnel 4 : Gestion des statistiques et des tableaux de bord : .....	26
1.1.5. Besoin fonctionnel 5 : Visualiser un tableau de bord :.....	26
1.1.6. Besoin fonctionnel 6 : Générer un rapport : .....	26
1.2. Besoins non fonctionnels :.....	26
1.2.1. Contraintes ergonomiques :.....	26
1.2.2. Contraintes techniques :.....	26
1.2.3. Contraintes d'implémentation : .....	27

2.	Analyse des besoins :	27
2.1.	Identification des acteurs :	27
2.2.	Diagramme des cas d'utilisation général :	28
2.3.	Raffinement et Description textuelle des cas d'utilisation :	29
2.3.1.	Raffinement cas d'utilisation « Gérer un compte utilisateur » :	29
2.3.2.	Description textuelle de cas d'utilisation « Gérer un compte utilisateur » :	30
2.3.3.	Raffinement cas d'utilisation « Gérer une base de données » :	30
2.3.4.	Description textuelle de cas d'utilisation « Gérer une base de données » :	31
2.3.5.	Raffinement cas d'utilisation « Gérer une source des données » :	31
2.3.6.	Description textuelle de cas d'utilisation « Gérer une source de données » :	32
2.3.7.	Raffinement cas d'utilisation « Gérer un tableau de bord » :	33
2.3.8.	Description textuelle de cas d'utilisation « Gérer un tableaux de bord » :	33
2.3.9.	Raffinement cas d'utilisation « Visualiser un tableau de bord » :	34
2.3.10.	Description textuelle de cas d'utilisation « Visualiser un tableau de bord » :	34
2.3.11.	Description textuelle de cas d'utilisation « Modifier les paramètres des tableaux » :	35
2.3.12.	Description textuelle de cas d'utilisation « générer un rapport » :	35
3.	Diagramme de séquence du système :	35
	Conclusion :	36
	Chapitre 4 : Etude Conceptuelle.....	38
	Introduction :	38
1.	Conception architecturale :	38
1.1.	Architecture 3-Tiers :	38
1.2.	Patron de conception MVC :	39
2.	Conception statique :	40
3.	Conception dynamique :	42
3.1.	Diagramme d'activités : « Espace du chef de projet ».....	42

3.2. Diagramme d'activités : « Espace d'un agent ».....	43
3.3. Diagramme de séquences : « Configurer une source de données ».....	44
3.4. Diagramme d'activités : « Configurer une source des données ».....	45
3.5. Diagramme de séquence : « Gérer une source des données » .....	46
3.6. Diagramme d'activité : « Gérer une source de données » .....	47
3.7. Diagramme de séquences : « Visualiser une charte graphique».....	48
Conclusion :.....	49
Chapitre 5 : Réalisation .....	52
Introduction :.....	52
1. Environnement de développement :.....	52
1.1. Langage de développement : .....	52
1.2. Plateforme de développement :.....	52
1.3. Outil de développement : .....	52
1.4. Outil de conception :.....	53
1.5. Support matériel et logiciel de développement :.....	53
1.6. Bibliothèque utilisée : .....	53
2. Architecture physique : .....	53
2.1. Architecture opérationnelle :.....	53
2.2. Architecture technique :.....	54
3. Application Dashbord : .....	55
3.1. Aperçu sur l'interface d'authentification :.....	55
3.2. Aperçu sur l'interface d'accueil :.....	56
3.3. Gestion utilisateur :.....	58
3.4. Gestion Connexion BD :.....	59
3.5. Gestion sources BD : .....	61
3.6. Gestion chartes :.....	63
Conclusion :.....	64

Conclusion générale .....	65
Référence.....	66



# Table des figures

Figure 1.1. Cycle en spirale.....	9
Figure 1.2. Tâches du projet.....	9
Figure 1.3. Planification des tâches.....	10
Figure 2.1. Traitement ETL [5].....	13
Figure 2.2. Principe de QlikView.....	16
Figure 2.3. Boîte de dialogue Edit Script.....	17
Figure 2.4. Tableau de bord présenté par QlikView .....	18
Figure 2.5. Interface graphique de SpagoBI Server .....	19
Figure 2.6. Interface de SpagoBI Meta et Studio .....	20
Figure 2.7. Interface pour gérer les sources de données .....	21
Figure 2.8. Interface pour gérer les données .....	21
Figure 2.9. Interface pour gérer les tableaux de bord.....	22
Figure 3.1. Diagramme de cas d'utilisation global .....	29
Figure 3.2. Diagramme de cas d'utilisation gestion des comptes utilisateurs.....	30
Figure 3.3. Diagramme de cas d'utilisation gestion des bases de données .....	31
Figure 3.4. Diagramme de cas d'utilisation gestion des sources des données .....	32
Figure 3.5. Diagramme de cas d'utilisation gestion des tableaux de bord.....	33
Figure 3.6. Diagramme de cas d'utilisation visualisation des tableaux de bord .....	34
Figure 3.7. Diagramme de séquence système .....	36
Figure 4.1. Architecture 3-tiers [11].....	39
Figure 4.2. Le modèle MVC [12].....	40
Figure 4.3. Diagramme des classes .....	41
Figure 4.4. Diagramme d'activités « Espace d'un Chef de projet » .....	43
Figure 4.5. Diagramme d'activités « Espace d'un Agent ».....	44
Figure 4.6. Diagramme de séquence « Configurer une source des données ».....	45
Figure 4.7. Diagramme d'activités « Configurer une source des données ».....	46
Figure 4.8. Diagramme de séquence « Gérer une source de données ».....	47
Figure 4.9. Diagramme d'activités « Gérer une source des données » .....	48
Figure 4.10. Diagramme de séquence « Visualiser une charte graphique » .....	49
Figure 5.1. Architecture opérationnelle.....	54
Figure 5.2. Architecture technique .....	55

Figure 5.3. Interface d'accueil .....	56
Figure 5.4. Authentification .....	56
Figure 5.5. Menu d'accueil du chef de projet .....	57
Figure 5.6. Menu d'accueil de notre agent .....	57
Figure 5.7. Interface Liste Utilisateurs .....	58
Figure 5.8. Interface Ajout Utilisateur .....	58
Figure 5.9. Interface Modifier Utilisateur .....	59
Figure 5.10. Liste des bases de données configurées .....	59
Figure 5.11. Formulaire de configurer une nouvelle connexion BD.....	60
Figure 5.12. Formulaire de reconfigurer une connexion BD .....	60
Figure 5.13. Interface de connexion BD .....	61
Figure 5.14. Configurer requête SQL.....	62
Figure 5.15. Configurer paramètres charte.....	62
Figure 5.16. Interface de résultat requête .....	63
Figure 5.17. Interface agent de consultation charte.....	64

## Table des tableaux

Tableau 2.1. Etude comparative entre Talend Open Studio et Pentaho .....	15
Tableau 2.2. Etude comparative entre QlikView et spagoBI.....	22
Tableau 3.1. Scénario de gestion des comptes utilisateurs .....	30
Tableau 3.2. Scénario de gestion des bases de données.....	31
Tableau 3.3. Scénario de gestion des sources des données .....	32
Tableau 3.4. Scénario de gestion des tableaux de bord.....	33
Tableau 3.5. Scénario de visualisation des tableaux de bord .....	34
Tableau 3.6. Scénario de modification des paramètres d'un tableau de bord.....	35
Tableau 3.7. Scénario de génération des rapports .....	35

# Introduction Générale

La révolution technique et industrielle qu'a connue le monde au début du 20<sup>ème</sup> siècle a été le catalyseur d'une révolution dans le domaine du traitement automatique de l'information ou informatique. En effet, les outils et techniques n'ont cessé d'être inventés et adaptés au profit du progrès scientifique et économique en particulier.

En l'instance, l'émergence du secteur tertiaire, ou secteur de services, trouvant sa source dans l'évolution de la consommation et la diminution de l'emploi industriel, a fait que le domaine de l'informatique décisionnelle a fait son essor et est devenue indispensable pour l'aide à la décision.

En effet, de nos jours, les grandes entreprises prestataires de services, telles que dans le secteur bancaire, sont amenés à fournir à leurs clients des outils d'aide à la décision tout en étant performant sur les volets coût, et temps de réponse. Vu le nombre de données traitées qui ne cesse de s'accroître, les outils de traitement classiques telles que les simples bases de données relationnelles et les entrepôts de données ne suffisent plus. L'informatique décisionnelle, ou Business Intelligence - BI, est devenue par excellence l'informatique à l'usage des décideurs et des dirigeants d'entreprises. Elle permet de collecter, consolider, modéliser et restituer les données, matérielles ou immatérielles, d'une entreprise en vue d'offrir une aide à la décision.

A la notion de la BI s'associe le concept de tableaux de bords, comme interface entre utilisateur final et bases de données, permettant aux décideurs, à travers des indicateurs bien prédéfinis, d'avoir une vue d'ensemble sur l'activité traitée.

Les tableaux de bord ont l'avantage d'être paramétrables, personnalisables et évolutifs. Une bonne conception et réalisation d'un tableau de bord est la clé d'une bonne visibilité et par conséquent une décision efficace et rapide, voire en temps réel.

C'est dans ce cadre que s'inscrit ce projet de fin d'études, visant à concevoir et développer une application web de génération des tableaux de bord dynamique et paramétrable. En effet, le BFI, l'entreprise d'accueil pour ce projet, souhaite proposer à ses clients une telle application de tableau de bord. Toutefois, pour arriver à proposer un produit innovant il faut commencer par développer une première version qui répond aux critères fondamentaux d'être dynamique et paramétrable. Celle-ci sera le noyau sur lequel d'autres fonctionnalités seront greffé dans des versions ultérieures. Ainsi, il nous a été demandé d'analyser des solutions existantes afin de

définir les besoins fonctionnels et non fonctionnels auxquels il faut s'aligner par la suite. Après il faut développer ces fonctionnalités en utilisant la technologie Java Enterprise Edition.

Afin de présenter notre travail nous avons organisé le contenu de ce présent rapport en cinq chapitres.

Le premier chapitre « Cadre Général » est un chapitre introductif dans lequel nous effectuons une brève description de la société. Ensuite, nous exposons le contexte général du projet et la mission confiée ainsi que la démarche de réalisation adoptée.

Le second chapitre « Etat de l'Art » fait l'objet d'une brève description de quelques solutions de l'informatique décisionnelle (Business Intelligence - BI) existantes dans le marché ainsi qu'une description des différents environnements de développement choisis.

Le troisième chapitre intitulé « Spécification et Analyse des besoins » définit les besoins auxquels le présent projet se propose de répondre. Nous allons également définir les acteurs concernés. Par la suite, nous spécifions les besoins fonctionnels illustrés par des diagrammes de cas d'utilisation avec quelques scénarios suivis d'une spécification des besoins non fonctionnels de notre application développée.

Le quatrième chapitre « Etude Conceptuelle » contient une présentation de la conception de notre projet. Dans lequel nous allons présenter notre étude conceptuelle qui sera illustrée par des diagrammes UML statiques et dynamiques.

Le cinquième chapitre « Réalisation » présente l'architecture de notre application. Nous allons également présenter l'implémentation des différents modules développés dans le cadre de notre projet.

Nous clôturons, finalement, ce rapport par une conclusion générale dans laquelle nous évaluerons les résultats atteints et nous exposons les perspectives éventuelles du présent projet.

# CHAPITRE 1

## Cadre Général

---

Dans cette première partie, nous présenterons l'organisme d'accueil, le contexte du projet, la mission confiée et nous introduirons la démarche de réalisation du projet.

---

# Chapitre 1 : Cadre Général

## Introduction :

Ce chapitre a pour objectif de situer notre projet dans son contexte général, à savoir, l'organisme d'accueil, la problématique et le travail demandé. Nous commençons par une présentation de l'organisme d'accueil à savoir BFI. Puis nous passons à la description de la mission qui nous a été confiée. Nous clôturons ce chapitre par la description de la démarche de réalisation du projet qui a été choisie.

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

### 1.1. Organisme d'accueil :

BFI est spécialisée depuis 1994 dans l'édition et l'intégration de solutions destinées aux banques et institutions financières. Bien implantée sur le marché africain, BFI a réussi à se forger une réputation de très haut niveau, en équipant plus de 180 clients dans 15 pays différents notamment avec des institutions de renommée mondiale [1].

### 1.2. Activités et Services :

Forte d'une expérience de près de 20 ans d'installations réussies en Afrique et d'une politique de proximité auprès de ses clients, BFI offre un savoir-faire reconnu dans la mise en œuvre de projets courant : le suivre et le pilotage, l'intégration de systèmes, la méthodologie de test et de recette, la migration des données, la formation et le transfert de compétences, le conseil et l'accompagnement et enfin le support et la maintenance [1].

### 1.3. Solutions et Références :

BFI offre des solutions modulaires, innovantes et personnalisables, adressant des domaines aussi variés que les systèmes d'information bancaire, la gestion de la trésorerie et des salles de marchés, le marché des titres ainsi que les systèmes de paiements nets et bruts. De plus, elle a su créer une relation de confiance avec ses clients grâce à une stratégie de coopération étroite sur le long terme [1].

## 2. Contexte du projet :

L'objectif principal de notre travail consiste à intégrer une solution de suivi et d'aide à la décision. En effet, il s'agit de créer une application web permettant de générer des tableaux de

## Chapitre 1. Cadre Général

---

bord simples, faciles à manipuler qui répondent aux besoins de l'entreprise et à son mode de fonctionnement. C'est pour cette raison que nous avons consacré la partie suivante pour faire une étude de l'existant tout en avançant nos critiques et en proposer les éventuelles améliorations souhaitées dans le cadre de notre projet.

### 2.1. Etude de l'existant :

Depuis les années 70, les entreprises travaillent avec des bases de données de petites tailles, donc le patron est la seule personne qui gère la situation et prend la décision. En effet, la concurrence entre les entreprises commence à faire rage et on commence à comprendre que « celui détient l'information détient le marché ». Mais, vers la fin des années 70, la taille des données augmente et le patron trouve des difficultés pour gérer les données des années et des données de production. D'où l'apparition des départements informatiques. Ce département est chargé de l'envoi des requêtes directement sur les serveurs de production ce qui se révélait plutôt dangereux pour ces derniers [2].

Dans les années 80, l'arrivée des bases relationnelles et du mode client / serveur a permis d'isoler l'informatique de production des dispositifs décisionnels. Dans la foulée, des acteurs spécialisés se sont lancés dans la définition de couches d'analyse, afin de masquer la complexité des structures de données. D'où l'apparition des nouvelles technologies décisionnelles permettant de collecter les données provenant des plusieurs sources externes structurées et/ou non structurées, de les transformer et puis de les charger dans un entrepôt de données. Ces informations stockées sont ensuite exploitées afin de faciliter la prise de décision par les décideurs [2].

### 2.2. Critique de l'existant :

Dans les années 90, la demande en information ne pourra jamais être pleinement satisfaite puisque le département informatique est tout le temps sollicité. En effet, les décideurs ne sont pas satisfaits par les systèmes d'informatique. Ils ont constaté qu'à cause de l'augmentation rapide de données stockées et sans cesse de la demande en information, ces systèmes ne pourront pas répondre aux exigences des demandeurs d'information... [2]

**Donc comment peut-on répondre aux demandes des décideurs, en donnant un accès rapide et simple à l'information stratégique et un sens aux données ?**

L'apparition de cette question clé nous a mené de réfléchir que les systèmes d'information ne pourra pas satisfaits les besoins des décideurs. Donc il faut avoir d'autres solutions rien que des



## Chapitre 1. Cadre Général

---

simples systèmes afin d'assouvir la soif d'information des chercheurs en informatique et de leur permettre de fouiller eux même dans les données pour trouver ce qu'ils cherchent. En effet, pour bien répondre à la question, il faut penser « Comment aider les décideurs à prendre des décisions ? » et « Que doit-il y avoir pour aider les décideurs à décider ? ».

### 2.3. Synthèse :

Actuellement, l'entreprise suit différentes procédures afin de mesurer son évolution de façon dynamique et régulière. En effet, elle englobe les solutions informatiques en bout de chaîne, rapports et tableaux de bord de suivi à la fois analytiques et prospectifs dont le but de consolider les informations disponibles au sein de ses bases de données.

Après notre étude, nous avons constaté que les systèmes d'information de BFI ne satisfont pas les besoins de nos chercheurs d'information, d'où le besoin créer de nouveaux concepts, outils, et logiciels permettant de gérer des grands volumes de données, indépendants du système de production et pour un nombre restreint d'utilisateurs. De plus, il doit être simple, facile à utiliser, fiable et hétérogène. Ce dernier permet de générer de rapports principalement à base de menus pour faciliter le travail et qui vont contenir des rapports paramétrables que les utilisateurs pourront interroger à leur guise.

En se basant sur ces informations, les dirigeants devront prendre notamment les décisions les plus opportunes qui influenceront considérablement sur la stratégie de l'entreprise et donc sur son avenir. Ainsi, il est nécessaire de décider d'une manière fondée, basée sur des informations claires, fiables et pertinentes en prenant en considération la grande masse de données présentes.

### 3. Mission confiée :

Ce projet rentre dans le cadre du projet de fin d'étude qui vient de conclure notre formation d'ingénieur à l'Ecole Nationale Supérieure d'Ingénieur de Tunis (ENSIT). Mon travail consiste à concevoir et à développer une application web de génération des tableaux de bord. Cette dernière garantit à l'entreprise un bon suivi, à distance, de son activité ce qui facilite aux dirigeants la prise de décision. En effet, notre but est de créer des tableaux de bord simples, faciles à manipuler qui satisfont les objectifs de notre projet et qui répondent aux besoins de l'entreprise et à son mode de fonctionnement.

Cette application est destinée de générer des tableaux de bord :

- **Paramétrable** : permettant de visualiser des données multi vues par profil utilisateur,

- **Dynamique** : permettant d'exploiter n'importe quel schéma de données,
- **Personnalisable** : permettant la configuration de chaque élément du tableau de bord.

La mise en place de notre application nécessite le développement de trois axes à savoir :

- **Axe Gestion du projet** : Utilisation de la méthodologie spirale.
- **Axe de modélisation conceptuelle** : Utilisation du langage de modélisation UML pour développer les axes fonctionnels statique et dynamique.
- **Axe de développement** : Utilisation de langage Java et l'utilisation de la plateforme de développement Java EE.

Notre étude nous a conduits à implanter une solution qui englobe un grand nombre de services visant à aider nos utilisateurs à prendre leur décision et de prévoir l'évolution de leur entreprise. En effet, notre application développée doit nous permettre de visualiser des données, provenant des sources externes (base de données Oracle, MySQL, ...), de les manipuler et de faire des rapports avec différents niveaux de vues (diagramme ou graphiques) qui seront par la suite délivrés à nos utilisateurs applications (agent ou chef de projet).

Notre application permet le paramétrage de chaque élément du tableau de bord. Cette dernière a comme objectifs :

- Contrôler la marge de l'entreprise à court terme,
- Évaluer la progression de l'entreprise,
- Maintenir les performances de l'entreprise,
- Garantir à l'entreprise un bon suivi de son activité,
- Offrir aux dirigeants des données fiables et pertinentes pour la prise de décision...
- permettre le contrôle de gestion en mettant en évidence les performances réelles et potentielles et les dysfonctionnements.

## 4. Démarche de réalisation du projet :

### 4.1. Méthodologie du projet :

Le choix d'une démarche convenable pour entreprendre un projet est une étape cruciale qui conditionne sa réussite. En effet, il n'existe pas de démarche qui soit standard dans le sens où

## Chapitre 1. Cadre Général

---

elle garantit la bonne conduite de n'importe quel travail mais son adoption doit être en fonction des spécialités de chaque projet y compris les buts à atteindre.

En ce qui nous concerne, nous commençons notre travail par une planification du projet et par une analyse préliminaire afin de dégager les objectifs initiaux à atteindre. Initialement, nous allons faire une étude complète sur des solutions existantes similaires à notre futur produit et qui répond à nos besoins.

Nous analysons les différents risques qui peuvent se produire au cours du développement de cette application vis-à-vis son intégration avec les différents modules existant gestion de données externes et diverses, satisfaction des chercheurs d'information, assurance qualité des rapports délivrés à nos utilisateurs, etc... Ensuite nous développons notre application en se basant sur notre étude.

Puis, nous passons à la phase de test. Si le prototype est validé, nous analysons les résultats obtenus, afin de raffiner les besoins. Par la suite, nous mettons comme objectif, son intégration avec les autres modules existant ce qui permet de partager les résultats basés sur le premier prototype développé. L'étape qui suit est l'analyse du risque. Si aucun risque n'est détecté, nous passons au développement du prototype suivi de l'étape test. Nous raffinons les besoins et répétons la même procédure afin de rendre notre projet complet. Nous avons donc estimé que la méthode en cycle spiral nous convient le mieux.

### 4.2. Cycle de vie du projet :

Le cycle de vie en spirale est un modèle générique de cycle de vie évolutif qui a été proposé par Barry W. Boehm en 1984. Ce modèle, axé sur la maîtrise et la réduction des risques, est davantage un cadre de travail guidant la construction d'une démarche spéciale que de projet, plutôt qu'une démarche formalisée. Chaque boucle de spirale permet d'identifier les objectifs propres de la boucle, les moyens alternatifs pour atteindre les objectifs et les contraintes de chaque alternative [3].

On distingue quatre phases dans le déroulement du cycle en spirale :

- détermination des objectifs, des alternatives et des contraintes.
- analyse des risques, évaluation des alternatives
- développement et vérification de la solution retenue
- revue des résultats et vérification du cycle suivant.

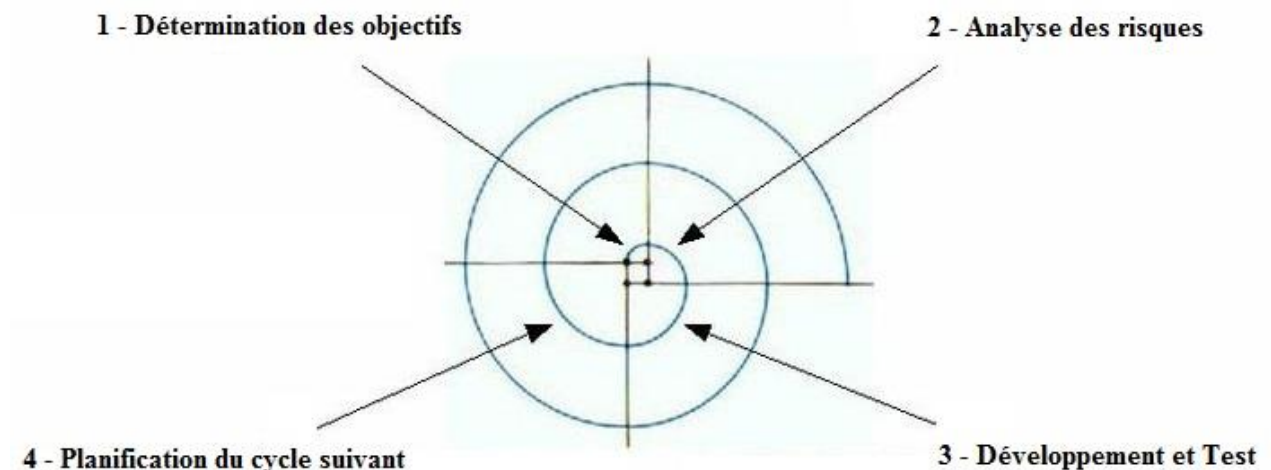


Figure 1.1. Cycle en spirale

### 4.3. Planification du projet :

Nous présentons dans la figure ci-dessous la planification des tâches du projet ainsi que leur date de début et de fin.

GANTT project		
Nom	Date de début	Date de fin
• Etude de l'existant de QlikView	01/02/16	05/02/16
• Spécification des besoins	08/02/16	12/02/16
• Etude de l'existant des solution d'intégration et de gestion des données	15/02/16	19/02/16
• Développement d'une interface de configurer une BD	22/02/16	26/02/16
• Développement d'une interface de configurer une requête	29/02/16	04/03/16
• Etude de l'existant de SpagoBI	07/03/16	18/03/16
• Recherche d'une bibliothèque graphique	21/03/16	25/03/16
• Architecture et Conception	28/03/16	01/04/16
• Développement de notre application	04/04/16	06/05/16
• Design de notre application	09/05/16	27/05/16
• Rédaction du rapport	01/02/16	03/06/16

Figure 1.2. Tâches du projet

La figure 1.3 présente la planification des tâches du projet en fonction du temps.

## Chapitre 1. Cadre Général

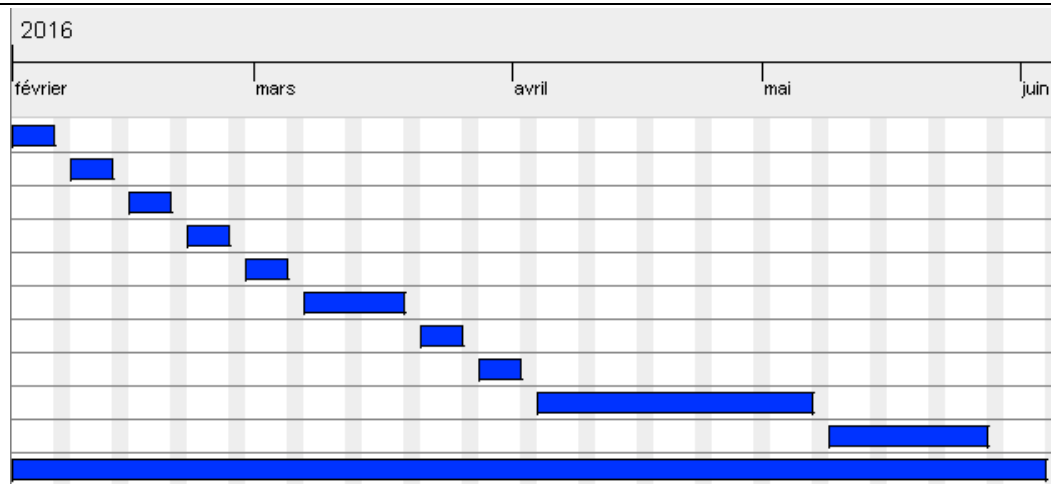


Figure 1.3. Planification des tâches

### Conclusion :

Le long de ce chapitre, nous avons abordé d'une façon générale le cadre de notre projet de fin d'étude. Tout d'abord, nous avons présenté l'organisme d'accueil, ensuite, nous avons abordé la notion de notre application développée. Nous y avons exposé l'étude du contexte général du notre projet afin de mettre en valeur notre travail. Le chapitre suivant sera consacré à la partie état de l'art.

## CHAPITRE 2

### ETAT DE L'ART

---

Dans ce chapitre, nous allons définir la notion de l'informatique décisionnelle pour la compréhension de notre projet. Nous présentons aussi une vue globale sur des solutions existantes. Nous aboutirons, enfin à la description de l'environnement de développement.

---

## Chapitre 2 : Etat de l'Art

### Introduction :

Avant de se lancer dans le développement d'une nouvelle solution permettant de générer des tableaux de bord simples, faciles à manipuler, il est primordial de bien étudier quelques solutions existantes dans le marché que nous allons critiquer par la suite.

Dans une première partie, nous allons présenter le concept de l'informatique décisionnelle en faisant une étude sur une technologie qui nous permet de gérer des données hétérogènes et de grandes tailles ainsi que de présenter des solutions d'intégration et de gestion des données. La seconde partie, nous allons élaborer une étude sur des solutions existantes qui satisfont les objectifs de notre projet et qui répondent aux besoins de l'entreprise.

### 1. Notion de l'informatique décisionnelle :

#### 1.1. Présentation du concept BI :

L'informatique décisionnelle (**en anglais** Business Intelligence ou BI) est une notion qui nous aide à exploiter des données de l'entreprise dans le but de faciliter la prise de décision par les décideurs, c'est-à-dire la compréhension du fonctionnement actuel et l'anticipation des actions pour un pilotage éclairé de l'entreprise. De ce concept est née la notion de modélisation dimensionnelle. Cette dernière est fondamentale pour répondre aux exigences de rapidité et de facilité d'analyse. Elle permet, en outre, de rendre les données d'un entrepôt cohérentes, lisibles, intelligibles et faciles d'accès [4].

En adaptant cette notion, nous allons faire une étude sur une technologie BI qui s'intéresse à collecter les données provenant des sources externes. En effet, nous allons présenter deux solutions logicielles décisionnelles disponibles en Open Source, utilisant la technologie informatique ETL (Extraction-Transformation-Chargement) et spécialisées dans l'intégration et la gestion des données. Ceci afin de faire une étude comparative dans le but de choisir la solution que nous allons utiliser pour l'extraction, le traitement et le chargement de données dans notre projet.

Nous allons donc commencer par présenter la technologie ETL avant d'entamer l'étude de deux solutions logicielles qui exploitent cette technologie : Talend et Pentaho.

### 1.2. Technologie ETL :

ETL est une technologie informatique qui nous permet de manipuler des données provenant des plusieurs sources externes. En effet, il assure la vérification et le nettoyage des données de prétraitement avant le chargement final dans une base de données décisionnelle [5].

Le traitement de la technologie ETL est illustré par la figure 2.1.

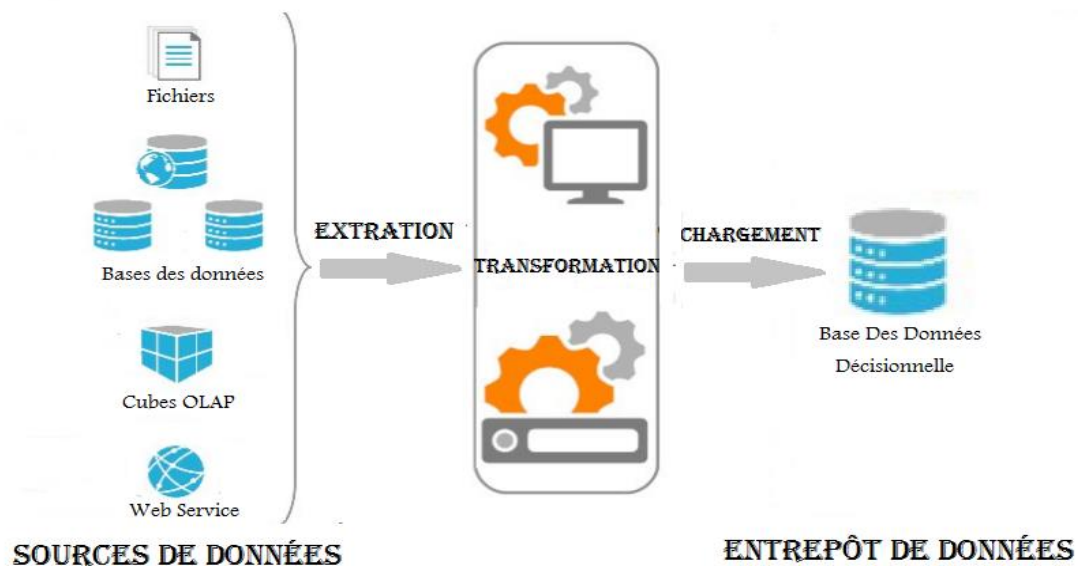


Figure 2.1. Traitement ETL [5]

Il permet au premier lieu d'extraire périodiquement des données de multiples sources externes, qui ne sont pas forcément implémentés au sein du même système de gestion de base de données (Base de données, Fichiers, web services, ...). Il les collecte puis les transforme pour alimenter les tables du modèle en étoile. Enfin, il charge ces données collectées dans n'importe quel format soit dans une base de données décisionnelle (MySQL, Oracle, PostgreSQL, ...) ou bien sous formes des fichiers (Excel, csv, xml, ...)

### 1.3. Etude des solutions d'intégration et de gestion des données

Après avoir décrit la technologie ETL, nous allons passer par la suite pour présenter les deux solutions d'intégration et de gestion des données.

#### 1.3.1. Présentation Talend Open Studio :

Talend est un éditeur de logiciel open source spécialisé dans l'intégration et la gestion des données. Il propose une interface permettant de concevoir graphiquement les connexions aux sources et les traitements sur les données [6].



## Chapitre 2. Etat de l'Art

---

Parmi les produits fournis par Talend, nous pouvons citer ces 5 plateformes :

- **Talend Open Studio For Big Data :**

C'est la première plate-forme d'intégration de données distribuées et non relationnelles. Elle est dédiée à gérer les données de grande taille de sorte que l'entreprise puisse transporter les données d'une base NOSQL vers la base MySQL dans les décisions en temps réel.

- **Talend Open Studio For Data Integration:**

Cette plate-forme nous permet de gérer des données stockées dans n'importe quel système de gestion de base de données afin de pouvoir les intégrer, nettoyer, masquer et de les transformer en décisions. Elle est plus efficace et plus rapide.

- **Talend Open Studio Data Quality :**

Cette plate-forme permet d'automatiser la qualité des données dans chaque intégration. Elle a le potentiel de réduire les coûts, augmenter les ventes et d'améliorer les performances en fournissant des données fiables disponibles dans tous les styles d'intégration.

- **Talend Open Studio For Enterprise Service Bus (ESB) :**

Cette plate-forme est dédiée pour l'intégration d'applications et de l'intégration de données permettant aux entreprises d'accroître sa productivité, réaliser des projets plus rapidement, et les coûts d'exploitation plus faibles.

- **Talend Open Studio For Master Data Management (MDM) :**

Cette plate-forme combine les données en temps réel et les partage à travers le Cloud et les applications mobiles. De plus, elle unifie toute quantité de données des clients à des produits aux fournisseurs.

Dans cette partie, nous allons nous intéresser seulement au produit Talend Open Studio For Data Integration puisqu'il utilise la technologie ETL. Ainsi, nous pouvons nous inspirer de ses fonctionnalités.

### 1.3.2. Présentation Pentaho :

Pentaho est une solution d'informatique décisionnelle open source entièrement développée en Java. Elle permet de collecter les données et les intégrer dans une base de données décisionnelle en utilisant la technologie ETL. Puis de les diffuser avec des serveurs Tomcat ou JBoss afin de créer des tableaux de bord et de faire une analyse multidimensionnelle [7].

## Chapitre 2. Etat de l'Art

Pentaho présente des produits de :

- **Big Data** afin de collecter les données qui figurent dans plusieurs sources de données,
- **Intégration des données** afin de générer les informations exploitables,
- **Analytique** afin d'analyser et visualiser des données collectées de différentes manières.

### 1.3.3. Critique des solutions présentées :

Après avoir présenté ces deux solutions d'information décisionnelle spécialisées dans l'intégration des données, nous allons les critiquer par la suite afin de prévoir qu'elle est la meilleure solution adoptée. En effet, nous présentons dans le tableau suivant une étude comparative entre Talend Open Studio et Pentaho.

Tableau 2.1. Etude comparative entre Talend Open Studio et Pentaho

Solutions décisionnelles	Talend Open Studio	Pentaho
Fichier Plats	Elle permet de manipuler n'importe quel type de fichiers et assure leur validité.	Pentaho garantit la validité seulement pour les fichiers XML.
Accès aux données relationnelles	Elle manipule tous les types de données simples ainsi que complexe.  La transformation des données sera basée sur des requêtes SQL.	Elle manipule que des données de types simples. Elle ne peut pas manipuler les données de types complexes.  Pentaho ne dispose pas d'un outil de création de requêtes SQL.
Développement avancé	Talend Open Studio génère des documents techniques et fonctionnels, comme elle peut les consulter à travers le web.	Pentaho n'admet pas ces fonctionnalités.

En se basant sur cette étude comparative entre ces deux solutions d'intégration et de gestion des données, nous avons constaté que Talend Open Studio est la meilleure solution qu'on peut admettre comme solution ETL. Parmi les avantages qui nous amènent à choisir cette solution comme une référence c'est la possibilité de gérer les sources de données de type fichiers plats (e.g. Excel, XML, etc.). En fait, notre solution sera essentiellement utilisée dans le secteur bancaire où les fichiers plats sont utilisés. Donc, afin de pouvoir convertir les données figurant

## Chapitre 2. Etat de l'Art

dans des fichiers plats vers des bases de données relationnelles faciles à manipuler par la suite, nous allons utiliser cette solution comme un outil qui nous aide à restructurer les données dans des tables.

## 2. Etude de l'existant :

Dans le domaine de l'informatique décisionnelle, il existe de nombreuses solutions qui permettent de générer des tableaux de bord. En effet, nous avons l'opportunité de pouvoir étudier les solutions les plus répandues et les plus utilisées dans le marché.

### 2.1. La plate-forme QlikView :

#### 2.1.1. Présentation du logiciel QlikView :

QlikView est une plate-forme d'aide à la décision. Il facilite la compréhension des tableaux de bord et le repérage des connexions, même avec des données complexes et volumineux. En effet, il permet d'interagir des données figurant dans des sources externes sans pouvoir connaître leur structure de stockage physique [8].

#### 2.1.2. Principe de QlikView :

Avant de commencer à présenter le principe de QlikView, nous allons le décrire par la figure présentée ci-dessous.

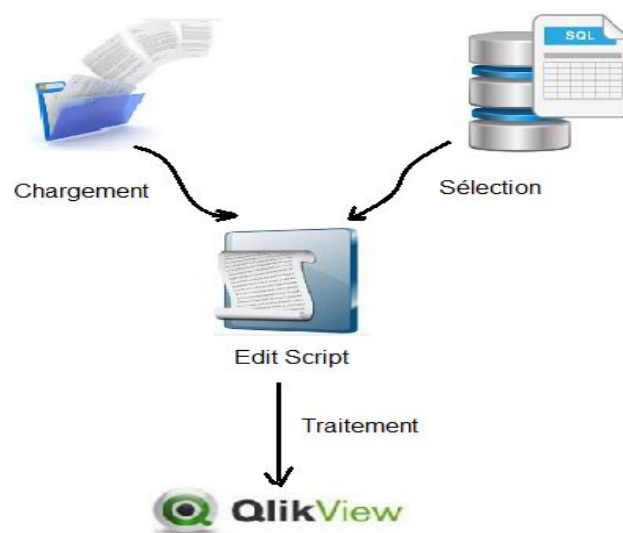


Figure 2.2. Principe de QlikView

QlikView nous permet de charger les données dans son environnement afin de créer une interface utilisateur destinée à un entrepôt de données. En effet, c'est un produit qui nous permet de récupérer et d'assimiler facilement les données provenant des sources externes (données

## Chapitre 2. Etat de l'Art

massives, fichiers Excel, fichiers, base de données, service web, ...), puis de faire des nouvelles associations entre ces données afin de créer des nouvelles structures de données. Enfin, nous permet de créer des tableaux et des graphiques dynamiques afin de réaliser des analyses statistiques qui nous aident à la suite de prendre nos décisions.

### 2.1.3. Fonctionnalités offertes par QlikView :

Parmi les fonctionnalités de QlikView, nous pouvons citer :

- **Première fonctionnalité : Consolider des données :**

Afin de créer un document QlikView, nous pouvons collecter les données figurant d'une ou plusieurs sources externes, telles qu'une base de données relationnelle ou des fichiers contenant des tables de données. En effet, nous avons la possibilité d'écrire un script, dans lequel nous spécifions notre base de données, les champs et les tables à récupérer.

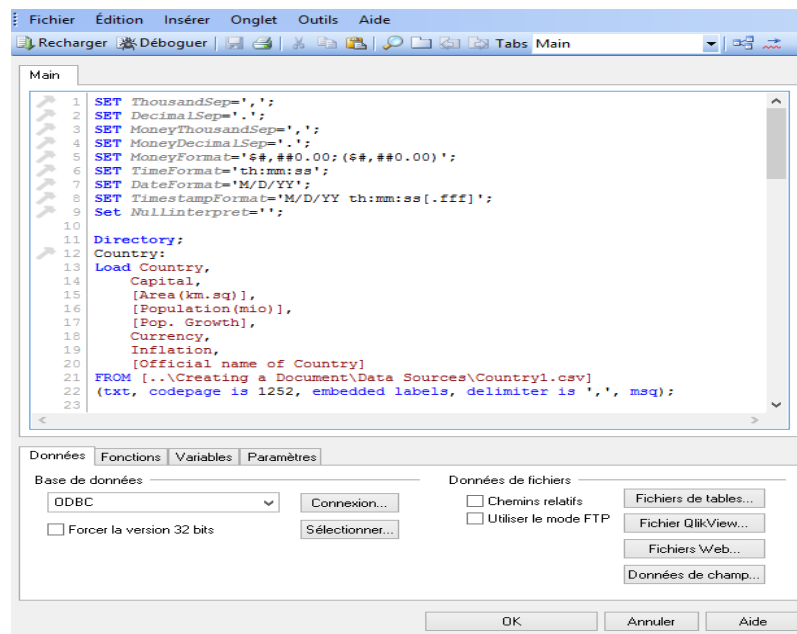


Figure 2.3. Boîte de dialogue Edit Script

- **Deuxième fonctionnalité : Rechercher des données :**

Le temps de réponse de ce logiciel est très rapide similaire à celui de Google. Il nous permet d'obtenir instantanément des résultats afin de visualiser de nouvelles connexions et relations entre vos données provenant de différentes applications.

- **Troisième fonctionnalité : Visualiser des données :**

## Chapitre 2. Etat de l'Art

Il nous permet d'analyser nos données chargées afin de les présenter dans des divers tableaux, graphiques... La figure 2.4 présente un exemple de tableau de bord généré par le logiciel QlikView qui nous permet de visualiser les données sous forme de chartes graphiques ainsi que des tables des données.

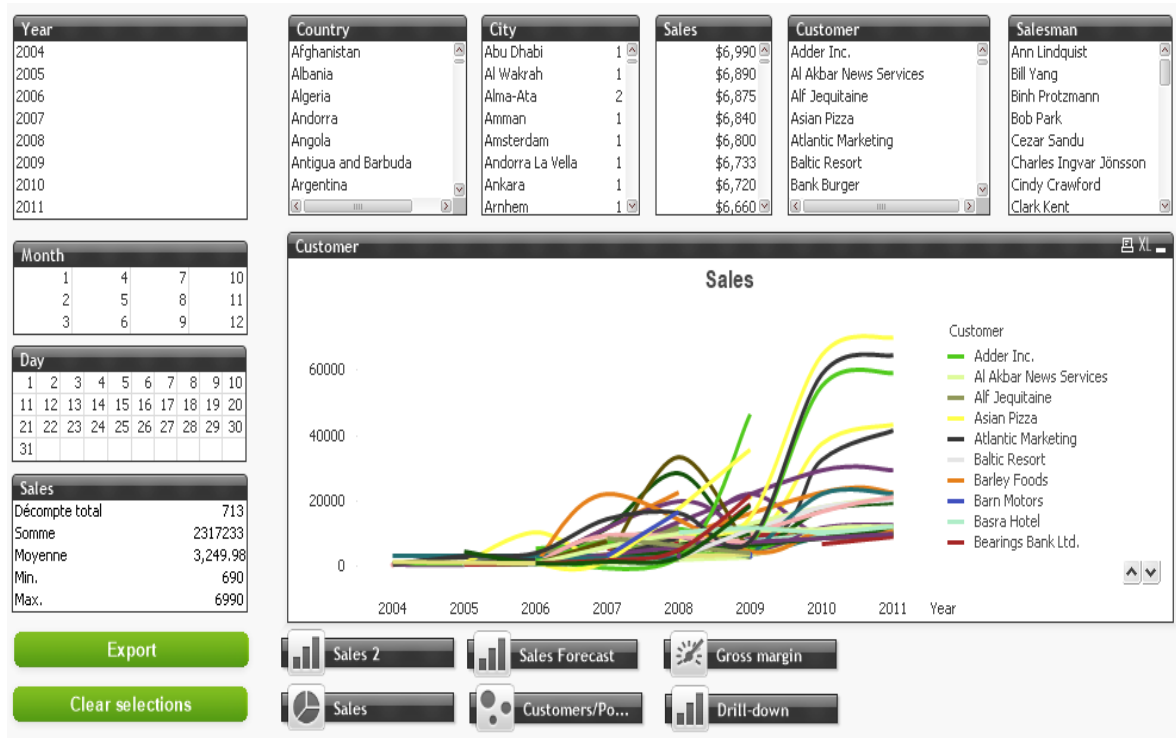


Figure 2.4. Tableau de bord présenté par QlikView

- **Quantième fonctionnalité : Piloter :**

Il permet de gérer les ressources, de manipuler des données et de faire l'analyse afin de prendre la bonne décision.

### 2.2. La plate-forme SpagoBI :

#### 2.2.1. Présentation de SpagoBI :

SpagoBI est la seule suite logicielle d'informatique décisionnelle entièrement open source, complète et flexible. Grâce à sa vaste offre d'outils analytiques et à son interface utilisateur, elle soutient toute activité stratégique, tant au niveau décisionnel qu'au niveau opérationnel [9].

#### 2.2.2. Architecture de SpagoBI :

Cette solution est constituée de quatre composantes.

- **SpagoBI Server** : Elle offre toutes les fonctionnalités centrales et analytiques de la plate-forme. Elle admet une interface graphique afin de nous permettre de configurer et paramétrer un tableau de bord. En effet, elle nous permet de connecter à une base de données afin d'extraire les données nécessaires en temps réel. De plus, elle offre l'opportunité de générer des tableaux de bord dynamiques et paramétrables.



Figure 2.5. Interface graphique de SpagoBI Server

- **SpagoBI Meta** : Cette plateforme permet nous utilisateurs de gérer métadonnées techniques et des métadonnées métier afin d'interpréter correctement leurs sources de données.
- **SpagoBI Studio** : C'est un environnement de développement intégré qui nous permet de gérer les modèles des différents moteurs d'analyse, comme rapports, diagrammes, tableaux de bord....

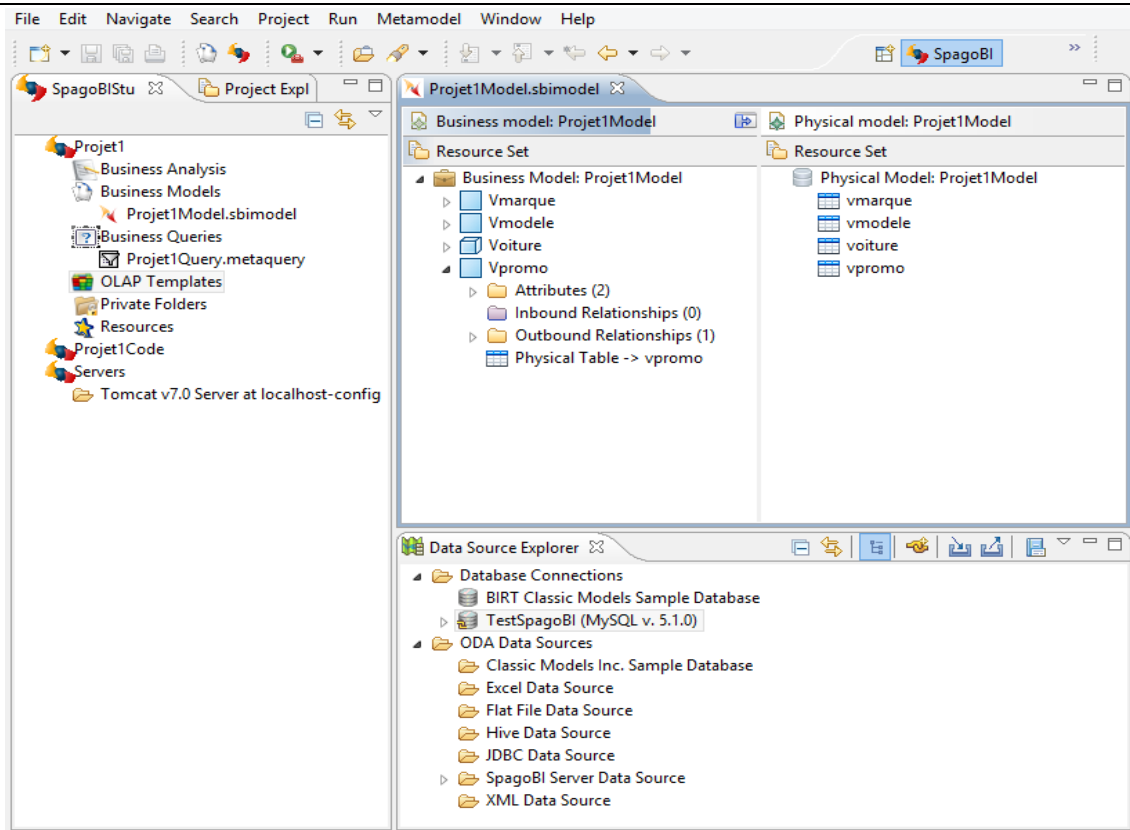


Figure 2.6. Interface de SpagoBI Meta et Studio

Ces deux plates-formes permettent d'établir une connexion avec une base de données externe afin de récupérer leurs tables, puis de créer des Business Models afin d'identifier les dimensions et jointures entre les tables. De plus, nous avons la possibilité d'identifier la requête SQL qui nous permet de collecter les données et de faire une analyse.

- **SpagoBI SDK** : Elle permet d'intégrer les documents créés par SpagoBI Studio afin de les publier et les tester sur SpagoBI Server.

### 2.2.3. Fonctionnalités offertes par SpagoBI :

La plate-forme SpagoBI se caractérise par quatre fonctionnalités :

- **Première fonctionnalité** : Gestion des sources de données afin de gérer les données provenant de multi-sources.

## Chapitre 2. Etat de l'Art

Label	Description
FoodmartHSQL	FoodmartHSQL
SpagoBIHSQL	SpagoBI repository
TestBD	MySQL_SpagoBI
TestSpagoBI	TestSpagoBI
ConnexionDB	ConnexionDB

Label:	<input type="text"/>
Description:	<input type="text"/>
Dialect:	<input type="text"/>
Multischema:	<input type="checkbox"/>
Read Only:	<input type="radio"/> Read Only <input checked="" type="radio"/> Read and write
Write Default:	<input type="checkbox"/>
Type:	<input checked="" type="radio"/> Jdbc <input type="radio"/> Jndi
URL:	<input type="text"/>
User:	<input type="text"/>
Password:	<input type="text"/>
Driver:	<input type="text"/>

Figure 2.7. Interface pour gérer les sources de données

Cette interface nous permet de se connecter à n'importe qu'elle base de données externe.

- **Deuxième fonctionnalité :** Gestion des données afin de récupérer les données nécessaires pour faire notre analyse. En effet, SpagoBI permet de charger les données provenant de bases de données relationnelles, des web services, des fichiers.... quelle que soit la nature du système gestion des données.

Label	Name	Type	Used By
DS_DEMO42_002	Sales and Costs ...	Query	1
DS_DEMO40_011	Sales and costs b...	Query	2
DS_DEMO42_001	Sales and costs b...	Query	1
DS_DEMO40_007	Sales by Media T...	Query	1
DS_DEMO42_011	Sales by education	Query	1
DS_DEMO42_009	Sales by gender a...	Query	1
DS_DEMO40_001	Sales by month	Query	1
DS_DEMO40_002	Sales by product ...	Query	3
DS_DEMO40_005	Sales by region	Query	1
DS_DEMO40_013	Sales by year	Query	1
DS_DEMO42_010	Sales by yearly in...	Query	1
DS_DEMO40_U02	Sales data set by ...	Query	0
DS_DEMO40_U03	Sales data set by ...	Query	0
DS_DEMO40_U14	Scatter data	Query	3
DEMO_TEST	TEST	Query	1
DS_DEMO42_012	Treemap - prova	Query	1
DEMO_Exemple1	Voiture list	Flat	0
DEMO_Marque	marque list	Flat	0
DEMO_Model	model list	Flat	0
DEMO_Promo	promo list	Flat	0

DataSet Type:	<input type="text" value="Query"/>
Data Source:	<input type="text" value="ConnexionDB"/>
Query:	<pre>select v.IdV, mq.LMarque, md.LModele, v.Coleur, v.NbrPlace, v.Taille, p.Val from voiture v, vpromo p, vmodele md, vmarque mq where v.IdModele = md.IdModele and v.IdMarque = mq.IdMarque and v.IdPromo = p.IdPromo group by v.IdV</pre>
<input type="button" value="Edit script"/>	

<input type="button" value="Add"/> <input type="button" value="Delete"/>	
Name	Type
<input type="text"/>	

Figure 2.8. Interface pour gérer les données



## Chapitre 2. Etat de l'Art

- **Troisième fonctionnalité :** Gestion des tableaux de bord afin d'aider les décideurs à prendre la bonne décision.

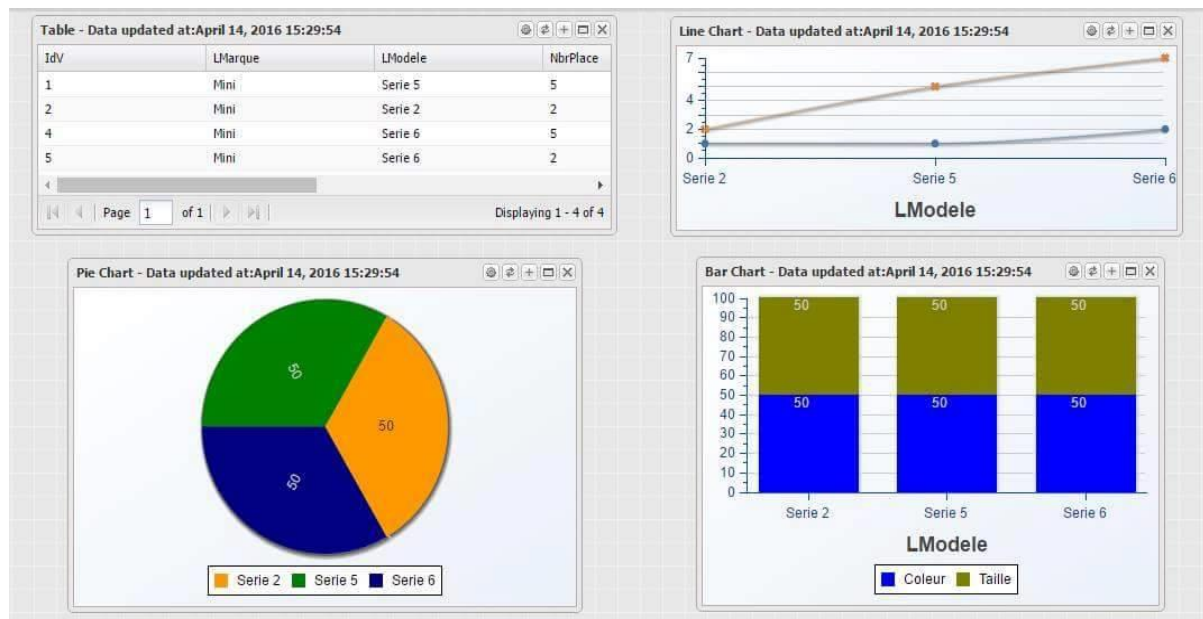


Figure 2.9. Interface pour gérer les tableaux de bord

SpagoBI nous permet de générer des tableaux de bord dynamiques paramétrables. En effet, elle nous permet de créer des tables de données et des graphiques multidimensionnels.

### 2.3. Etude comparative :

Nous présentons dans le tableau suivant une étude comparative entre QlikView et SpagoBI.

Tableau 2.2. Etude comparative entre QlikView et spagoBI

Solution BI	QlikView	SpagoBI
<b>Premier Critère</b>	QlikView possède un panneau de configuration plus détaillé relativement complexes comme elle rend des tâches banales complexes.	SpagoBI est facile à utiliser puisqu'elle décompose son environnement en 4 plates-formes dont chacune a son rôle.
<b>Deuxième Critère</b>	Les données visualisées que nous souhaitons analyser seront rechargées localement.	Notre plate-forme permet de se connecter à des sources de données externes sans les exporter.

## Chapitre 2. Etat de l'Art

<b>Troisième critère</b>	Elle n'offre pas l'accès à des sources de données en temps réel. En effet, toute modification de source de données nécessite du rechargement de nouveau.	Elle autorise l'accès à des sources de données en temps réel puisque elle se connecte directement à une base de données externe.
<b>Quatrième Critère</b>	Les données récupérées seront stockées dans son mémoire.	Les données ne seront pas stockées dans notre plateforme mais seulement analysées.

Qlikview est un produit plus puissant. Il gère facilement des données énormes importées d'un grand nombre de sources et offre la meilleure performance à ce niveau. Mais, bien qu'il puisse répondre à quelques besoins de BFI comme la génération des tableaux de bord personnalisables, nous ne pouvons pas l'admettre comme une meilleure solution à cause de ce qui précède.

En revanche, SpagoBI est une plate-forme d'aide à la décision. Et se basant notre étude, nous avons constaté qu'elle est la meilleure solution. Mais cette solution ne satisfait pas les besoins de BFI puisqu'elle est un produit open source et ne peut pas être convertie en un produit commercial. Ceci sera une violation de licence de ce produit. Donc, nous allons fournir une solution sur mesure et personnalisée pour BFI, ayant les fonctionnalités s'inspirant de SpagoBI.

En résumé, puisque BFI souhaite commercialiser ses produits, il s'est avéré nécessaire de développer une nouvelle solution personnalisée afin d'avoir le droit de commercialisation d'une part. D'autre part, l'appropriation du code source permettra d'intégrer dans des futures versions des fonctionnalités avancées, innovantes et sur mesure pour les clients potentiels.

### Conclusion :

Dans ce chapitre état de l'art, tout d'abord, nous avons fait une étude de l'existante. Ensuite, nous avons présenté et critiqué des solutions logicielles permettant d'intégrer et de gérer les données dans le cadre d'une solution BI. Finalement nous avons mis cette étude dans son contexte spécifique afin de montrer son intérêt pour la suite du projet.

## CHAPITRE 3

### SPECIFICATION ET ANALYSE

---

Dans le troisième chapitre, nous allons nous concentrer à la spécification des exigences et l'analyse des besoins de l'application.

---

## Chapitre 3 : Spécification et Analyse

### Introduction :

Pour déterminer précisément les objectifs à atteindre, il est impératif de spécifier les besoins fonctionnels et non fonctionnels et de faire les analyses nécessaires. En effet, la phase analyse et spécification formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire, et permet de délivrer l'étude fonctionnelle du système. Elle permet ainsi d'obtenir une idée sur ce que va réaliser le système d'un point de vue de métier.

Ce chapitre est consacré à l'analyse du système de gestion des tableaux de bord. Cela va nous permettre de détailler les besoins fonctionnels et non fonctionnels relatifs à ce dernier. Nous allons commencer par une étude des besoins fonctionnels et non fonctionnels. Ensuite, nous allons entamer l'analyse ces besoins.

### 1. Spécification des besoins :

#### 1.1. Besoins fonctionnels :

Il s'agit des fonctionnalités du système. En effet, les besoins fonctionnels ou les besoins métiers représentent les actions que le système doit exécuter. Ce sont des exigences spécifiant un comportement d'entrée / sortie du système [10].

Les principaux besoins fonctionnels de notre application sont :

##### 1.1.1. Besoin fonctionnel 1 : Configurer une base des données :

Il faut avoir la possibilité de configurer notre application afin de lui permet de se connecter et de communiquer avec des bases de données externes. Ceux-ci peuvent être gérés par des systèmes de gestion de bases de données variés.

##### 1.1.2. Besoin fonctionnel 2 : Manipuler les données provenant des sources externes :

Il faut avoir la possibilité de créer des requêtes SQL permettant de collecter les données provenant d'une base de données externe et les manipuler. Les manipulations consistent essentiellement en la transformation des données collectées afin de créer des nouvelles tables de données uniformes et bien structurées. Celles-ci seront utilisées par la suite pour pouvoir générer des tableaux de bord.

### **1.1.3. Besoin fonctionnel 3 : Gérer les comptes et les droits d'accès :**

Il faut avoir la possibilité d'effectuer des opérations de gestion telle que l'ajout, la modification, la suppression et la consultation des informations caractérisant chaque compte utilisateur. Egalement, il faut avoir la possibilité d'attribuer les droits d'accès à chaque utilisateur selon son rôle. De plus, il faut avoir la possibilité d'organiser les fonctionnalités de chaque utilisateur pour les opérations diverses selon son droit d'accès.

### **1.1.4. Besoin fonctionnel 4 : Gestion des statistiques et des tableaux de bord :**

Il faut avoir la possibilité de réaliser des tableaux de bord temps réel ayant les informations nécessaires de l'état actuel avec toutes les mises à jour nécessaires. Ces tableaux de bord doivent permettre de piloter et mesurer la performance de l'entreprise.

### **1.1.5. Besoin fonctionnel 5 : Visualiser un tableau de bord :**

Il faut permettre la consultation des graphiques ainsi que des tables statistiques. Cette consultation va permettre de faire une analyse des valeurs remarquables. Ceci favorise la prise de décision et la mise en œuvre des actions correctives.

### **1.1.6. Besoin fonctionnel 6 : Générer un rapport :**

Il faut permettre de générer des rapports récapitulant les statistiques de nos tableaux de bord ainsi que les tables des données présentés dans notre application afin de faciliter l'analyse des données par les décideurs.

## **1.2. Besoins non fonctionnels :**

Il s'agit des besoins qui caractérisent le système. En effet, les besoins non fonctionnels concernent les contraintes à prendre en considération pour mettre en place une solution adéquate. Ce sont des besoins en matière de performance, de portabilité décrivant les contraintes ergonomiques, de type de matériel ou de type de contraintes techniques. Comme elles peuvent concerner les contraintes d'implémentation [10].

Les principaux besoins non fonctionnels de notre application sont :

### **1.2.1. Contraintes ergonomiques :**

Notre future application doit être facile à manipuler, à exploiter et permet de satisfaire les besoins de notre utilisateur. Ceci ne doit pas nécessiter un apprentissage préalable ou une assistance particulière.

### **1.2.2. Contraintes techniques :**

- **La première contrainte : Sécurité d'accès :**

## Chapitre 3. Spécification et Analyse

---

L'application doit garantir la sécurité à travers la gestion des droits d'accès pour la base de données ainsi que l'exigence d'une authentification pour tout utilisateur.

- **La seconde contrainte : Portabilité :**

L'application ne doit pas dépendre d'un système d'exploitation ou d'un système de gestion de données particuliers. Ainsi, il sera facile de porter l'application d'un environnement à un autre sans avoir besoin de modifier le code source.

- **La troisième contrainte : Rapidité :**

Notre solution développée doit répondre aux besoins de nos utilisateurs avec un temps de réponse et de traitement raisonnable.

- **La quatrième contrainte : Maintenabilité :**

Notre solution doit être modulaire pour garantir la souplesse et l'évolutivité.

### 1.2.3. Contraintes d'implémentation :

L'application doit être développée avec les technologies Java EE. Outre le fait que l'usage de Java EE rejoint le besoin de portabilité, le BFI a une grande expertise et maîtrise du Java EE au sein de son staff. Ceci permet donc de continuer le développement sur ce projet dans le futur.

Egalement, il faut adopter un modèle de programmation qui respecte le patron de conception MVC. Ce découpage en couches permet une meilleure séparation entre les composantes de l'application. Ceci doit permettre de faciliter la maintenance de l'application.

## 2. Analyse des besoins :

### 2.1. Identification des acteurs :

Au niveau de cette section, nous présentons les différents acteurs susceptibles à interagir avec notre système, mais tout d'abord, nous devons définir le concept d'acteur.

Un acteur est une entité externe du système. Il peut être un utilisateur humain, un dispositif matériel ou bien un autre système qui interagit directement avec le système étudié. Ce dernier peut manipuler directement l'état du système, par l'envoi et la réception des messages.

Dans le cas de notre système nous avons deux acteurs :



**Chef de projet :** Responsable de la configuration et du paramétrage des tableaux de bord multi-sources. Son rôle consiste à gérer et manipuler les données de n'importe quelle type de base des données et même de configurer les éléments d'un tableau de bord.



**Agent :** Utilisateur de notre application. Son rôle consiste à consulter les tableaux de bord tels que les tables des données et les graphiques afin de faciliter la prise de décision pour l'entreprise.

En se basant sur ce qui précède, nous allons présenter le diagramme global des cas d'utilisation de notre système. Nous allons présenter par la suite les raffinements et les descriptions de quelques cas d'utilisations ainsi que le diagramme de séquence de notre système.

### 2.2. Diagramme des cas d'utilisation général :

Les cas d'utilisation offrent un moyen à la fois systématique et intuitif de saisir les besoins fonctionnels. Et à ce niveau, après avoir identifiant les acteurs qui participent au notre système, nous allons présenter le diagramme des cas d'utilisation pour bien présenter les fonctionnalités de notre application.

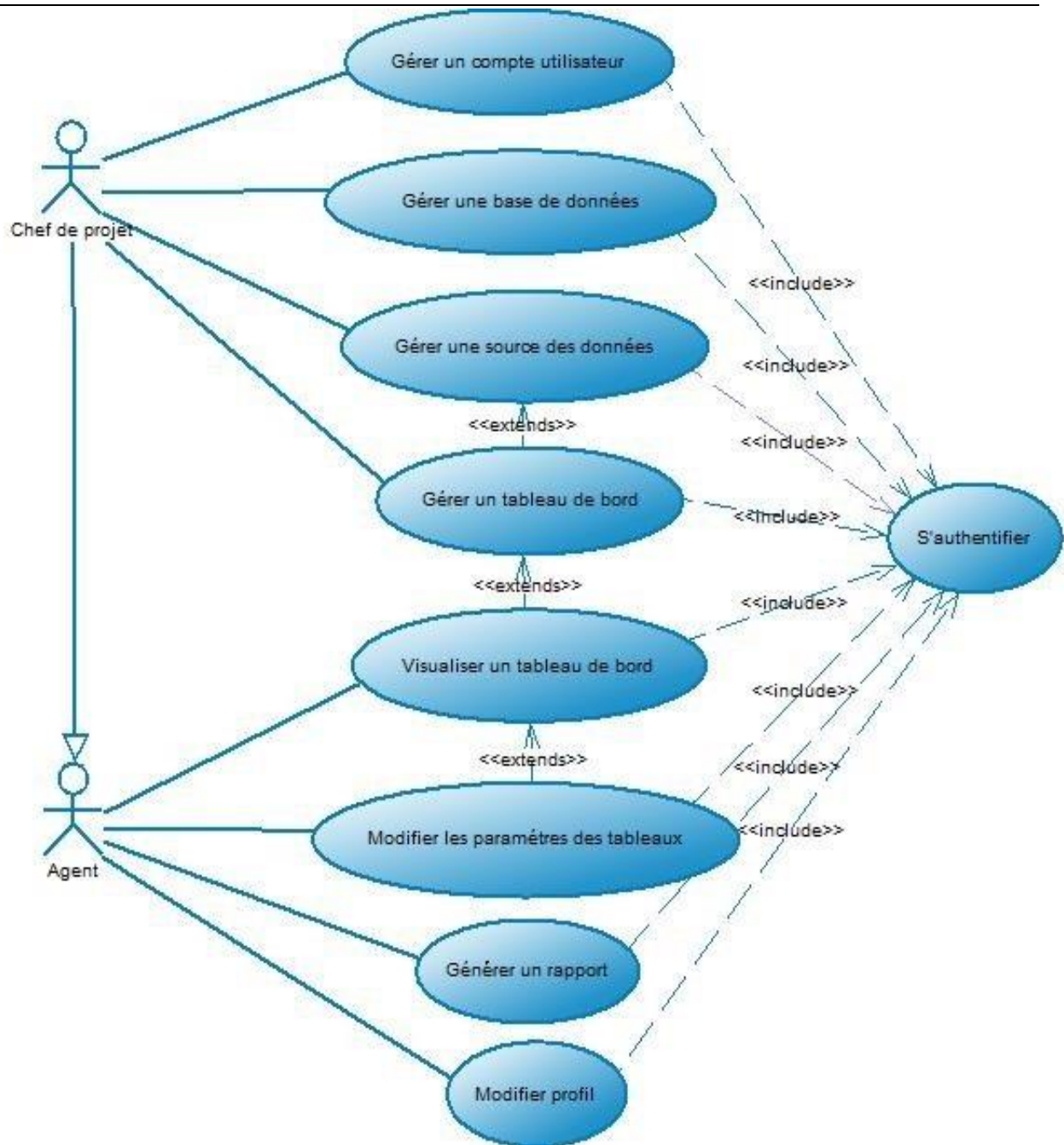


Figure 3.1. Diagramme de cas d'utilisation global

### 2.3. Raffinement et Description textuelle des cas d'utilisation :

#### 2.3.1. Raffinement cas d'utilisation « Gérer un compte utilisateur » :

La figure suivante donne un aperçu plus détaillé concernant le cas d'utilisation « Gérer un compte utilisateur ».



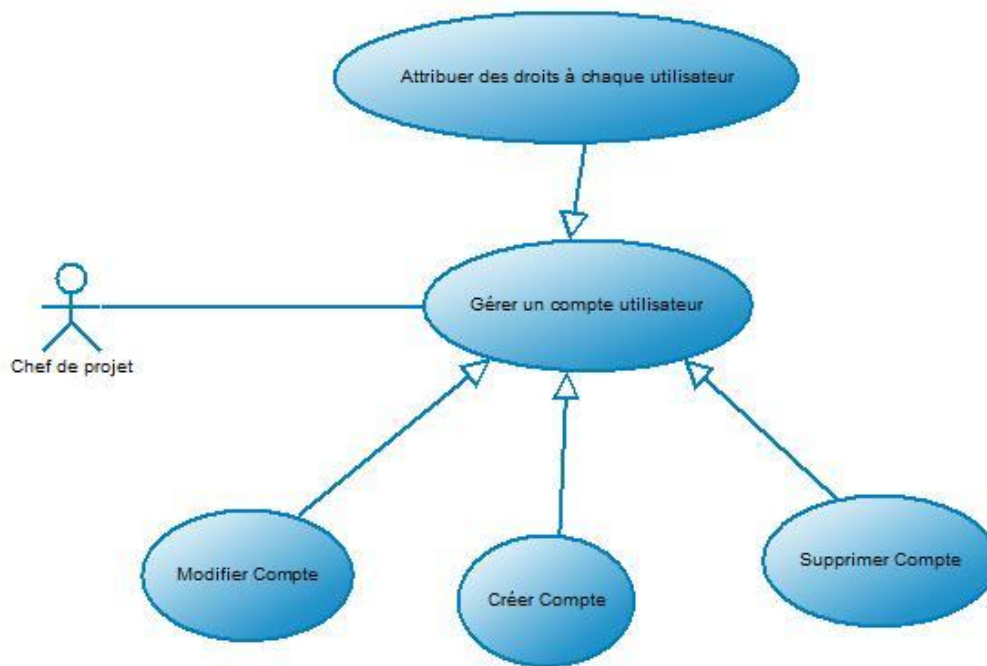


Figure 3.2. Diagramme de cas d'utilisation gestion des comptes utilisateurs

### 2.3.2. Description textuelle de cas d'utilisation « Gérer un compte utilisateur » :

Nous présentons dans le tableau suivant le scénario de gestion des comptes utilisateur.

Tableau 3.1. Scénario de gestion des comptes utilisateurs

Titre	Scénario
Objectifs	Le chef de projet a l'opportunité de consulter la liste des utilisateurs de notre application afin de gérer leurs comptes.
Acteurs	Chef de projet.
Précondition	Notre chef de projet doit s'authentifier pour accéder à sa session.
Scénario nominal	<ul style="list-style-type: none"><li>Le chef de projet consulte la liste des utilisateurs afin d'ajouter, de supprimer ou bien de modifier les coordonnées d'un compte d'utilisateur.</li><li>Le chef de projet donne les droits d'accès à chaque utilisateur pour qu'il puisse consulter les tableaux de bord.</li></ul>
Post condition	Chaque utilisateur possède un accès à sa session.

### 2.3.3. Raffinement cas d'utilisation « Gérer une base de données » :

La figure suivante donne un aperçu plus détaillé concernant le cas d'utilisation « Gérer une base de données ».

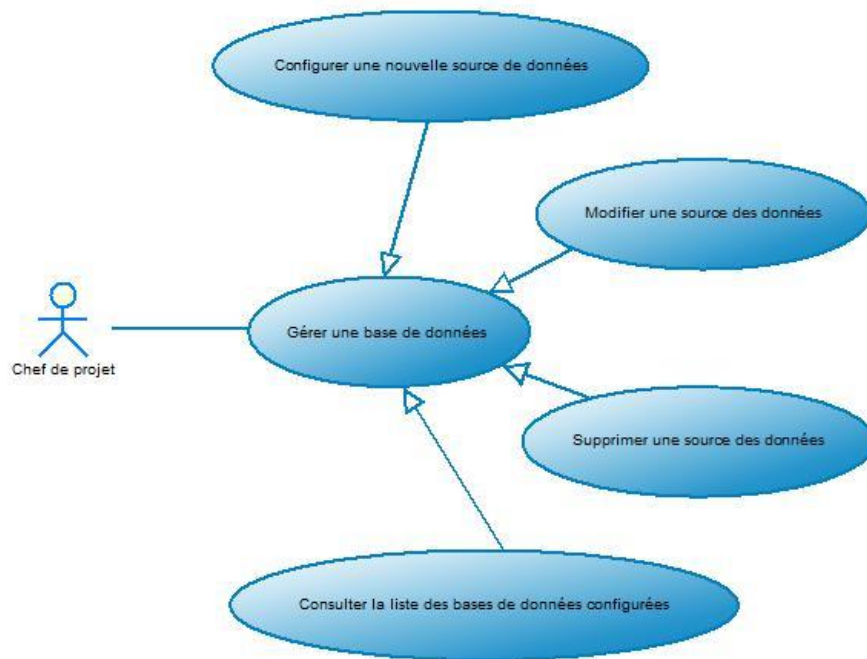


Figure 3.3. Diagramme de cas d'utilisation gestion des bases de données

### 2.3.4. Description textuelle de cas d'utilisation « Gérer une base de données » :

Nous présentons dans le tableau suivant le scénario de gestion des bases de données.

Tableau 3.2. Scénario de gestion des bases de données

Titre	Scénario
Objectifs	Le chef de projet gère une base de données.
Acteurs	Chef de projet.
Précondition	Notre chef de projet doit s'authentifier pour accéder à sa session.
Scénario nominal	<ul style="list-style-type: none"><li>• Le chef de projet configure une nouvelle connexion afin de se connecter à une base de données externe.</li><li>• Le chef de projet manipule les sources de données déjà configurées.</li><li>• Le chef de projet consulte la liste des bases de données configurées.</li></ul>
Scénario alternatif	Le chef de projet peut modifier les paramètres d'une base de données déjà configurée.
Post condition	Notre agent peut connecter à des bases des données externes afin de manipuler leurs données.

### 2.3.5. Raffinement cas d'utilisation « Gérer une source des données » :

La figure suivante donne un aperçu plus détaillé concernant le cas d'utilisation « Gérer une source des données ».

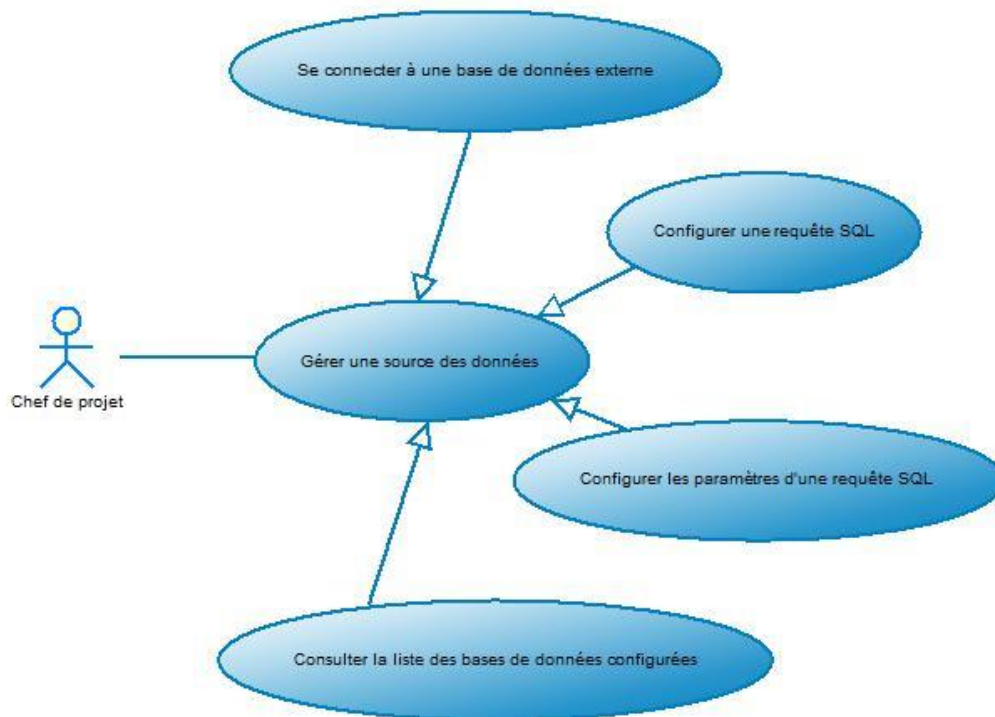


Figure 3.4. Diagramme de cas d'utilisation gestion des sources des données

### 2.3.6. Description textuelle de cas d'utilisation « Gérer une source de données » :

Nous présentons dans le tableau suivant le scénario de gestion des sources des données.

Tableau 3.3. Scénario de gestion des sources des données

Titre	Scénario
<b>Objectifs</b>	Le chef de projet gère une source des données.
<b>Acteurs</b>	Chef de projet.
<b>Précondition</b>	Notre chef de projet doit s'authentifier pour accéder à sa session.
<b>Scénario nominal</b>	<ul style="list-style-type: none"> <li>Le chef de projet consulte la liste des bases de données configurées afin de permettre de collecter les données stockées dans des bases externes.</li> <li>Le chef de projet saisi la requête SQL afin de récupérer les données d'une base pour créer des nouvelles tables statistiques.</li> <li>Le chef de projet filtre les données sélectionnées en configurant les paramètres d'une requête SQL créée.</li> </ul>
<b>Post condition</b>	Notre agent peut consulter les tableaux de bord gérés par notre chef de projet selon leurs droits d'accès.

### 2.3.7. Raffinement cas d'utilisation « Gérer un tableau de bord » :

La figure suivante donne un aperçu plus détaillé concernant le cas d'utilisation « Gérer un tableau de bord ».

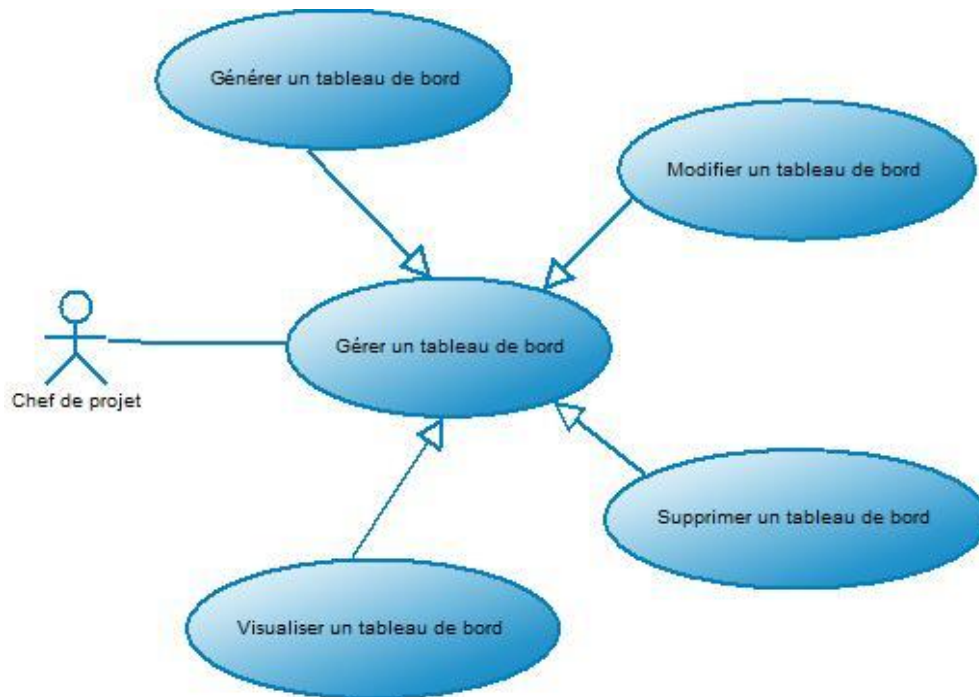


Figure 3.5. Diagramme de cas d'utilisation gestion des tableaux de bord

### 2.3.8. Description textuelle de cas d'utilisation « Gérer un tableaux de bord » :

Nous présentons dans le tableau suivant le scénario de gestion des tableaux de bord.

Tableau 3.4. Scénario de gestion des tableaux de bord

Titre	Scénario
Objectifs	Le chef de projet gère un tableau de bord.
Acteurs	Chef de projet.
Précondition	Notre chef de projet doit s'authentifier pour accéder à sa session.
Scénario nominal	<ul style="list-style-type: none"><li>• Le chef de projet utilise une requête SQL configurée afin de générer un nouveau tableau de bord.</li><li>• Le chef de projet peut manipuler les tableaux de bord.</li></ul>
Scénario alternatif	Le chef de projet peut modifier les paramètres d'un tableau de bord généré comme l'axe d'abscisse et de coordonnée d'un graphique...
Post condition	Notre agent peut consulter les tableaux de bord gérés par notre chef de projet selon leurs droits d'accès.

### 2.3.9. Raffinement cas d'utilisation « Visualiser un tableau de bord » :

La figure suivante donne un aperçu plus détaillé concernant le cas d'utilisation « Visualiser un tableau de bord ».

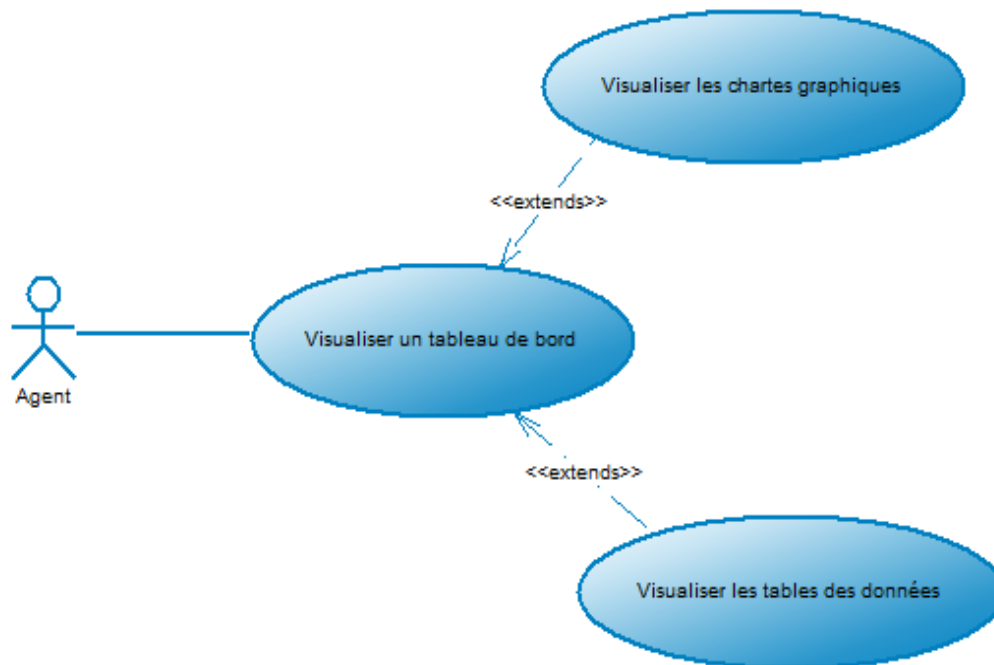


Figure 3.6. Diagramme de cas d'utilisation visualisation des tableaux de bord

### 2.3.10. Description textuelle de cas d'utilisation « Visualiser un tableau de bord » :

Nous présentons dans le tableau suivant le scénario de visualisation des tableaux de bord.

Tableau 3.5. Scénario de visualisation des tableaux de bord

Titre	Scénario
<b>Objectifs</b>	L'agent visualise les tableaux de bord.
<b>Acteurs</b>	Chef de projet et Agent.
<b>Précondition</b>	Notre agent doit s'authentifier pour accéder à sa session.
<b>Scénario nominal</b>	L'agent consulte les tableaux de bord afin de prendre les bonnes décisions. L'agent peut modifier les paramètres d'un tableaux de bord comme il peut modifier leur type.
<b>Scénario alternatif</b>	L'agent peut aussi consulter toutes les graphiques créées et même les tables statistiques créées.
<b>Post condition</b>	Notre application favorise la prise de décision et la mise en œuvre des actions correctives.

## Chapitre 3. Spécification et Analyse

### 2.3.11. Description textuelle de cas d'utilisation « Modifier les paramètres des tableaux » :

Nous présentons dans le tableau suivant le scénario de modification des paramètres d'un tableau de bord.

Tableau 3.6. Scénario de modification des paramètres d'un tableau de bord

Titre	Scénario
<b>Objectifs</b>	L'agent a l'opportunité de paramétrer un tableau de bord.
<b>Acteurs</b>	Agent.
<b>Précondition</b>	Notre agent possède le droit d'accès afin de visualiser les tableaux de bord et modifier les paramètres d'un tableau de bord visualisé.
<b>Scénario nominal</b>	L'agent gère les paramètres d'un tableaux de bord. En fait, il a le droit d'identifier l'axe d'abscisse et de coordonnées et même de déterminer les champs d'un tableau de bord consulté.
<b>Post condition</b>	L'agent

### 2.3.12. Description textuelle de cas d'utilisation « générer un rapport » :

Nous présentons dans le tableau suivant le scénario de génération des rapports.

Tableau 3.7. Scénario de génération des rapports

Titre	Scénario
<b>Objectifs</b>	Le chef de projet peut se connecter à une base de données externe.
<b>Acteurs</b>	Chef de projet et Agent.
<b>Précondition</b>	Notre acteur doit s'authentifier pour accéder à sa session.
<b>Scénario nominal</b>	Le chef de projet configure notre application pour la rendre accessible à n'importe quelle base de données externe.
<b>Post condition</b>	Notre chef de projet peut charger les données d'une base de données externe afin de générer des nouveaux tableaux de bord.

## 3. Diagramme de séquence du système :

Ce diagramme de séquence traite le système comme étant une boîte noire, il met l'accent sur un scénario particulier d'un cas d'utilisation. En fait, puisque tous les cas d'utilisations nécessitent une phase d'authentification alors nous allons présenter par la suite un diagramme de séquence qui traduit la phase d'authentification de notre application développée [10].

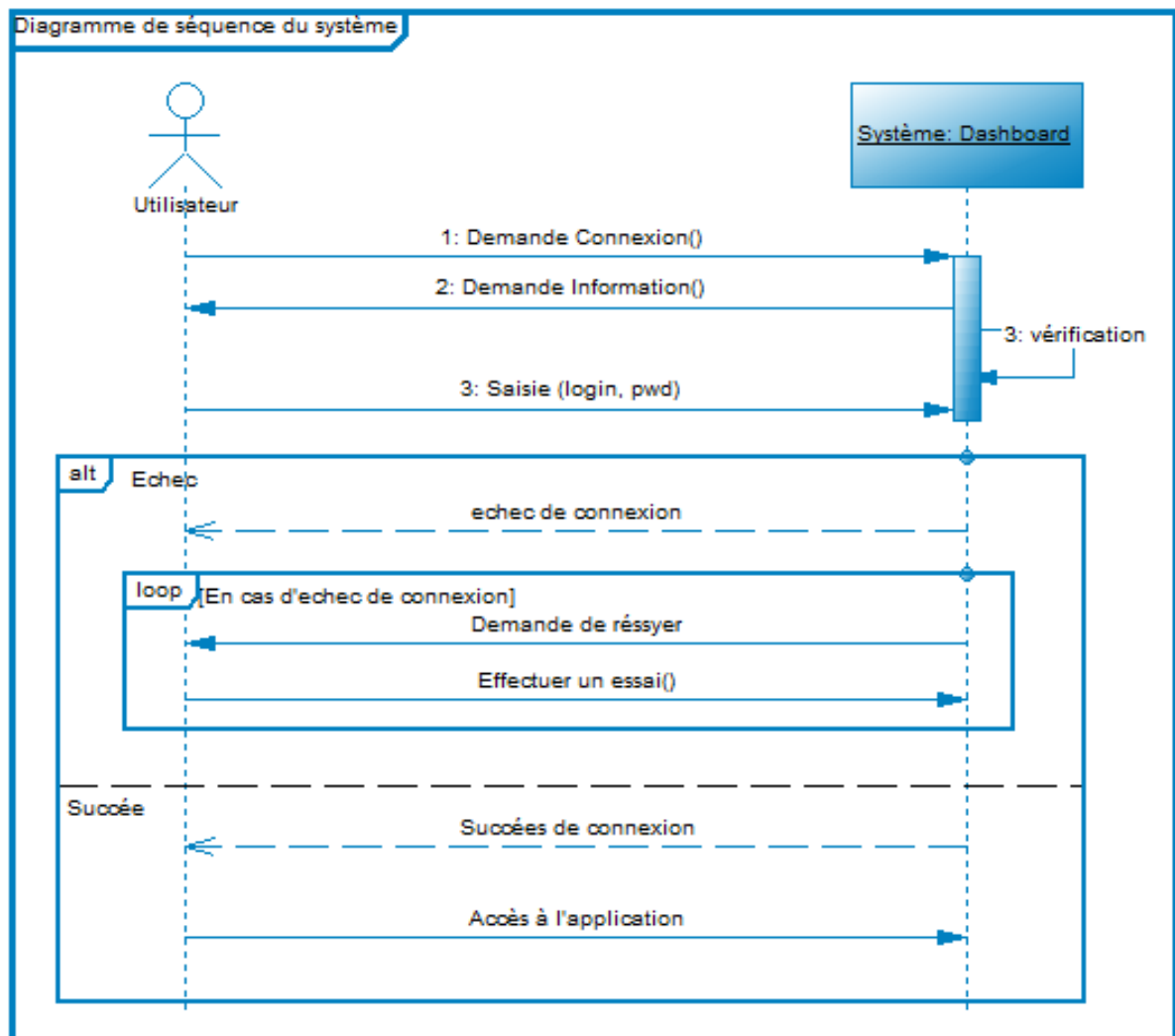


Figure 3.7. Diagramme de séquence système

### Conclusion :

Dans ce chapitre, nous avons passé en revue les différents besoins exprimés pour la réalisation du projet. Ce qui nous a permis de définir les besoins fonctionnels et non fonctionnels du système à développer. Ceci nous a permis d'avoir une vision plus claire du travail demandé et une compréhension plus profonde des tâches à réaliser. Nous allons aborder dans le prochain chapitre une étude détaillée de la conception de l'application.

## CHAPITRE 4

### ETUDE CONCEPTUELLE

---

Dans le quatrième chapitre, nous allons nous concentrer à la description de la conception architecturale, statique ainsi que dynamique de notre application.

---



# Chapitre 4 : Etude Conceptuelle

## Introduction :

La phase de la conception est la phase initiale de création et de mise en œuvre de notre projet. En fait, c'est l'étape la plus importante dans notre cycle de développement qui est située entre la phase d'analyse et spécification des besoins et la phase de réalisation.

Dans ce chapitre, nous allons présenter en détail la conception de notre projet à travers des diagrammes UML statiques tels que le diagramme des classes ainsi que des diagrammes UML dynamiques comme les diagrammes de séquences et d'activités.

## 1. Conception architecturale :

Avant de passer à détailler la conception statique et dynamique de notre système, nous allons présenter l'architecture de notre application. En effet, elle est basée sur le concept de couches d'abstractions, en outre, les composants qui permettent de découper notre application afin de faire une séparation des rôles lors du développement. Notre application va suivre l'architecture « 3-Tiers » combinée au patron de conception « Modèle Vue Contrôleur » - MVC.

### 1.1. Architecture 3-Tiers :

L'architecture matérielle qui sera mise en place pour notre application, étant une application web, est l'architecture « 3-Tiers » ou architecture à trois niveaux [11]. Elle divisée en trois niveaux ou couches :

- **La première couche c'est la couche DAO (Data Access Object) :**

C'est une couche de gestion des bases des données. Elle s'occupe de l'accès et du stockage des données, le plus souvent des données persistantes au sein des SGBD (Système de Gestion des Bases de Données).

- **La deuxième couche c'est la couche métier :**

C'est une couche qui encapsule les traitements d'où l'implémentation des algorithmes de l'application. Cette couche est indépendante de toute forme d'interface avec l'utilisateur. Ainsi elle doit être testée en dehors par ses utilisateurs via une interface web. Généralement, cette couche est la plus stable de l'architecture. Elle ne change pas si on change l'interface utilisateur ou la façon d'accéder aux données nécessaires au fonctionnement de l'application.

## Chapitre 4. Etude Conceptuelle

- **La troisième couche c'est la couche interface utilisateur :**

C'est la partie présentation de l'application qui nous permet de piloter l'application et d'en recevoir des informations.

La figure ci-dessous montre l'implémentation de cette architecture dans notre application.

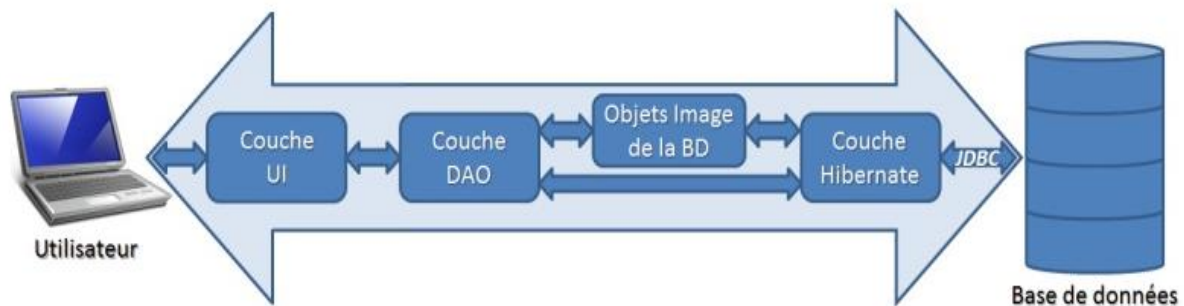


Figure 4.1. Architecture 3-tiers [11]

### 1.2. Patron de conception MVC :

Il s'agit d'une méthode de conception visant à organiser l'interface homme-machine d'une application. En effet, elle impose la séparation entre les données, la présentation et les traitements, ce qui donne trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur [12].

- **Le modèle :**

Le modèle représente le comportement de l'application : traitement des données, interactions avec la base de données, etc.... Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité. En effet, il offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur) et même des méthodes pour récupérer ces données.

Dans notre cas, il s'agit de l'ensemble des classes java tels que les classes du « package entities » contenant les données de notre base relationnelle récupérer par Hibernate et celles du « package DAO » et « packadeservice » contenant les méthodes qui manipulent les données récupérées.

- **La vue :**

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur. Ces différentes actions sont envoyées au contrôleur. Les vues générées par le MVC permettent à l'utilisateur d'interagir /communiquer avec l'application. Il s'agit de pagesjsp.

- **Le contrôleur :**

Le contrôleur prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle et les synchroniser. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement de données, le contrôleur demande la modification des données au modèle, ce dernier avertit la vue que les données ont changé pour qu'elle se mette à jour.

La figure 4.2 illustre le modèle MVC en décrivant les relations qui se trouve entre les trois couches (Modèle, Contrôleur et Vue).

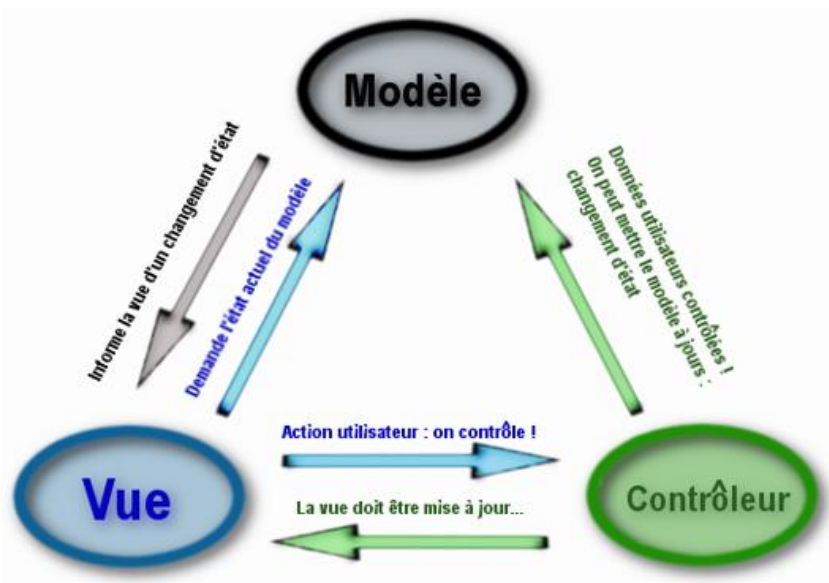


Figure 4.2. Le modèle MVC [12]

## 2. Conception statique :

Afin de pouvoir présenter la vue statique de notre système, nous allons utiliser le « diagramme des classes ». En effet, le « diagramme des classes » est généralement considéré comme le plus important dans un développement orienté objet. Il représente l'architecture conceptuelle du système : il décrit les classes que le système utilise, ainsi que leurs liens [10].

## Chapitre 4. Etude Conceptuelle

La figure ci-dessus met l'accent sur la structure statique du système à l'aide d'objets, attributs, opérations et relations.

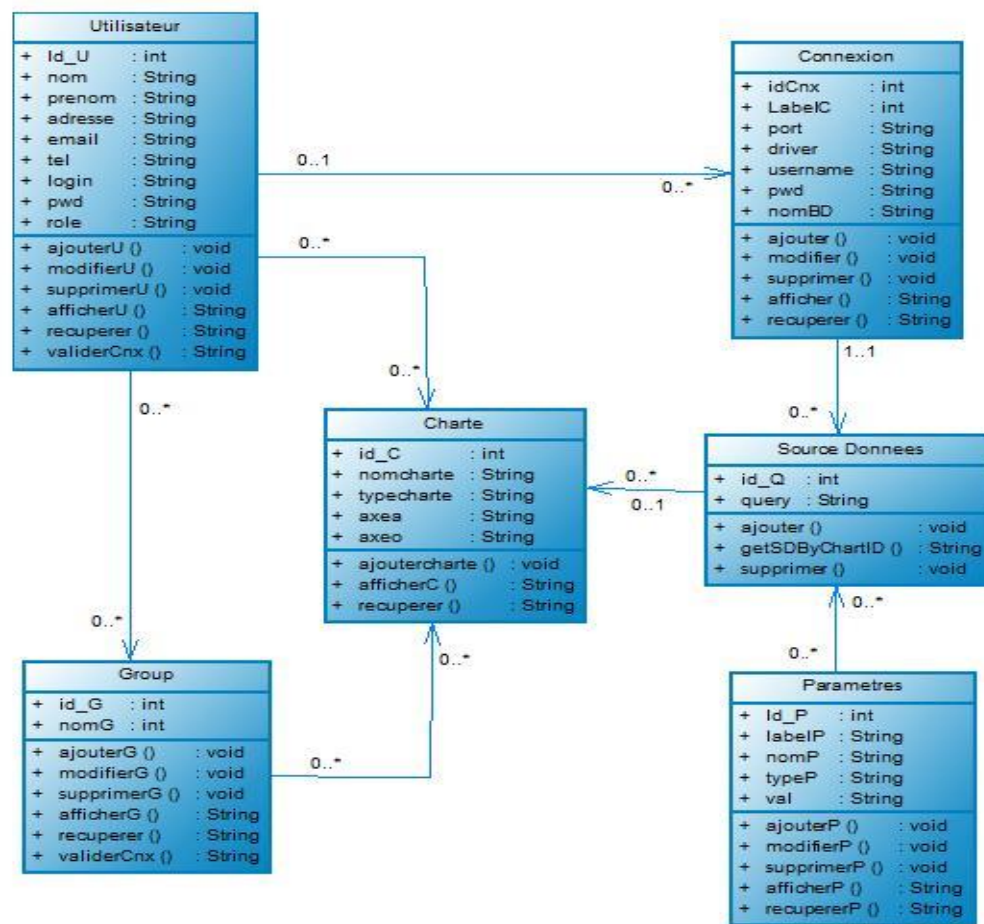


Figure 4.3. Diagramme des classes

Le diagramme de classes de la figure 4.3 illustre comment nous avons conçu notre application au niveau des classes qui constituent, les relations entre elles, les attributs et les méthodes qui caractérisent chaque classe. En effet, parmi les classes qui figurent dans ce diagramme, nous citons :

- Classe « Connexion » : c'est une classe permettant de manipuler la liste des bases de données externes déjà configuré par notre chef de projet ainsi que sauvegarder les paramètres d'une base de données externe.
- Classe « Source Données » : c'est une classe permettant de manipuler les requêtes SQL.
- Classe « Paramètres » : c'est une classe qui contient les paramètres d'une requête SQL configurée.

- Classe « Charte » : c'est une classe qui contient les paramètres d'une charte graphique.
- Classe « Utilisateur » : c'est une classe contient les coordonnées de chaque utilisateur de notre application tels que le profil ainsi que leur droit accès.
- Classe « Group » : c'est une classe qui permet de spécifier l'ensemble de groupe utilisateurs qui peuvent consulter les mêmes données.

### 3. Conception dynamique :

Tout au long de cette section, nous allons présenter la vue dynamique de notre projet. En effet, nous allons présenter quelques diagrammes UML tels que les « diagrammes de séquences » ainsi que les « diagrammes d'activités » des différents cas d'utilisation afin de schématiser les interactions de nos acteurs avec notre système.

En fait, un diagramme de séquence représente la succession chronologique des opérations et des interactions entre les objets du système. Chaque message transitant sur un lien est symbolisé par une flèche porteuse d'une expression. En revanche, un diagramme d'activité représente le déclenchement d'événements en fonction des états du système et modélise des comportements pouvant être parallèles. Donc il décrit le flux de travail [10].

#### 3.1. Diagramme d'activités : « Espace du chef de projet »

La figure 4.4 présente le digramme des différentes activités qui modélise l'ensemble de fonctionnalités de l'espace d'un Chef de projet. En effet, notre espace est divisé en Cinq sections. La première est réservée à la page d'accueil. La seconde pour la gestion des utilisateurs tels que l'ajout des nouveaux utilisateurs et la consultation de la liste des utilisateurs afin de modifier leur profil. De plus, une section de gestion connexion BD permettant d'afficher la liste des bases de données configurées afin de les manipuler ou bien de configurer des nouvelles connexions à des bases de données externes. Et une section de gestion sources de données qui dépend de ce qui précède afin de pouvoir connecter à une base de données déjà configurée, de créer une requête et de configurer leurs paramètres. La dernière section est réservée à la gestion charte permettant de configurer une charte et même de consulter la liste des chartes configurées afin de les manipuler par la suite.

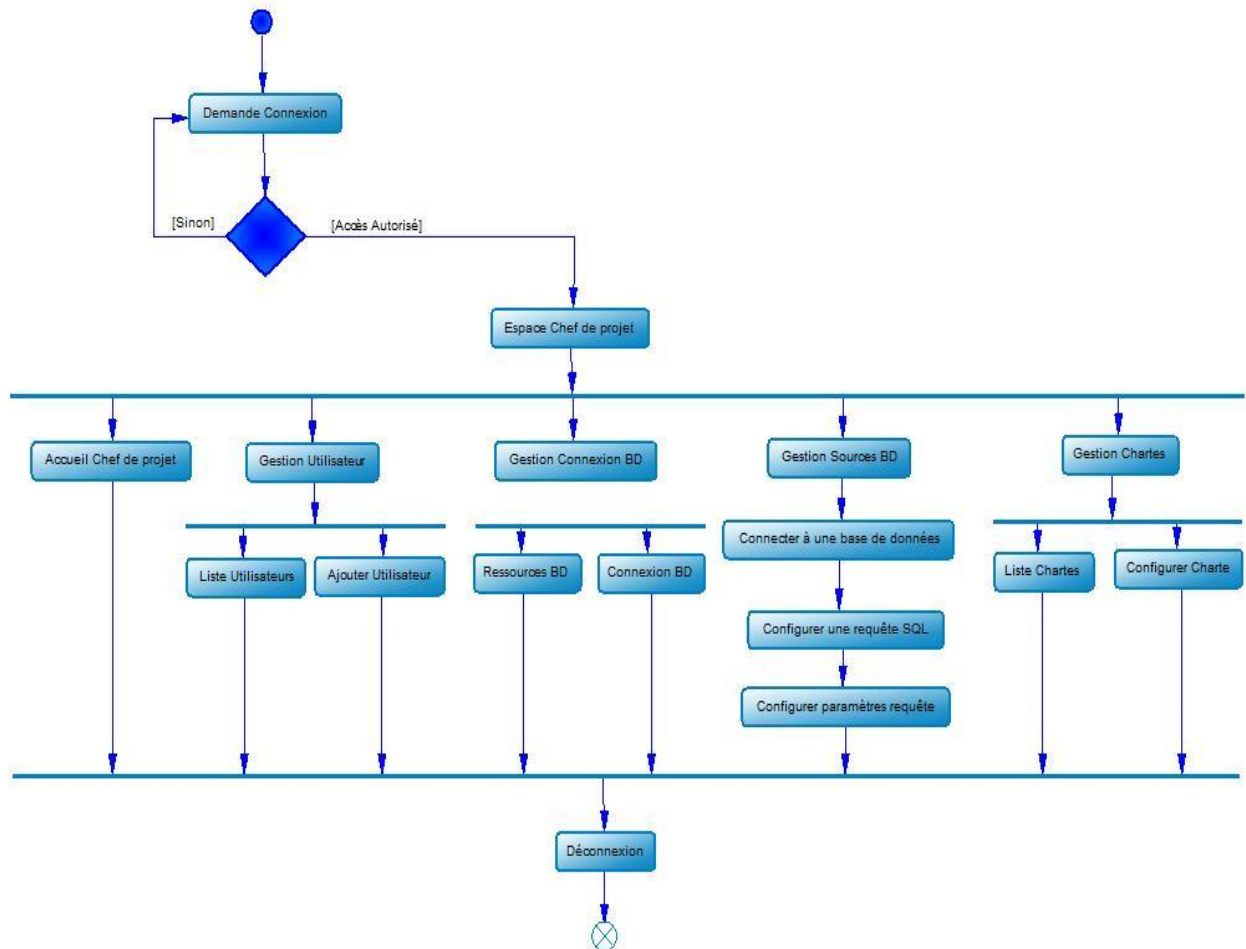


Figure 4.4. Diagramme d'activités « Espace d'un Chef de projet »

### 3.2. Diagramme d'activités : « Espace d'un agent »

La figure 4.5 présente le diagramme des différentes activités qui modélise l'ensemble de fonctionnalités de l'espace d'un agent. En effet, notre espace est divisé seulement en deux sections. La première est réservée à la page d'accueil et la seconde pour consulter la liste des chartes configurées par notre chef de projet.



Figure 4.5. Diagramme d'activités « Espace d'un Agent »

### 3.3. Diagramme de séquences : « Configurer une source de données »

La figure ci-dessous illustre le diagramme de séquences qui décrit le scénario de configuration d'une base de données. En effet, notre chef de projet est la seule personne qui a le droit de configurer notre application afin de la rendre accessible de se connecter à une base de données externe.

En outre, les paramètres des droits d'accès d'un serveur de données tels que le « Driver », le « Username » et le « Password » doivent être saisis en premier afin de pouvoir consulter la liste des bases de données. Ensuite, si l'accès n'est pas autorisé, alors notre système affiche un message pour informer notre administrateur qu'il n'a pas le droit pour accéder à ce serveur. Sinon, il va nous afficher une liste de toutes les bases des données qui figurent dans ce serveur. En fin, après avoir choisi notre base de données externe, notre système va enregistrer ces données dans sa base de données interne en vue de les réutiliser par la suite.

## Chapitre 4. Etude Conceptuelle

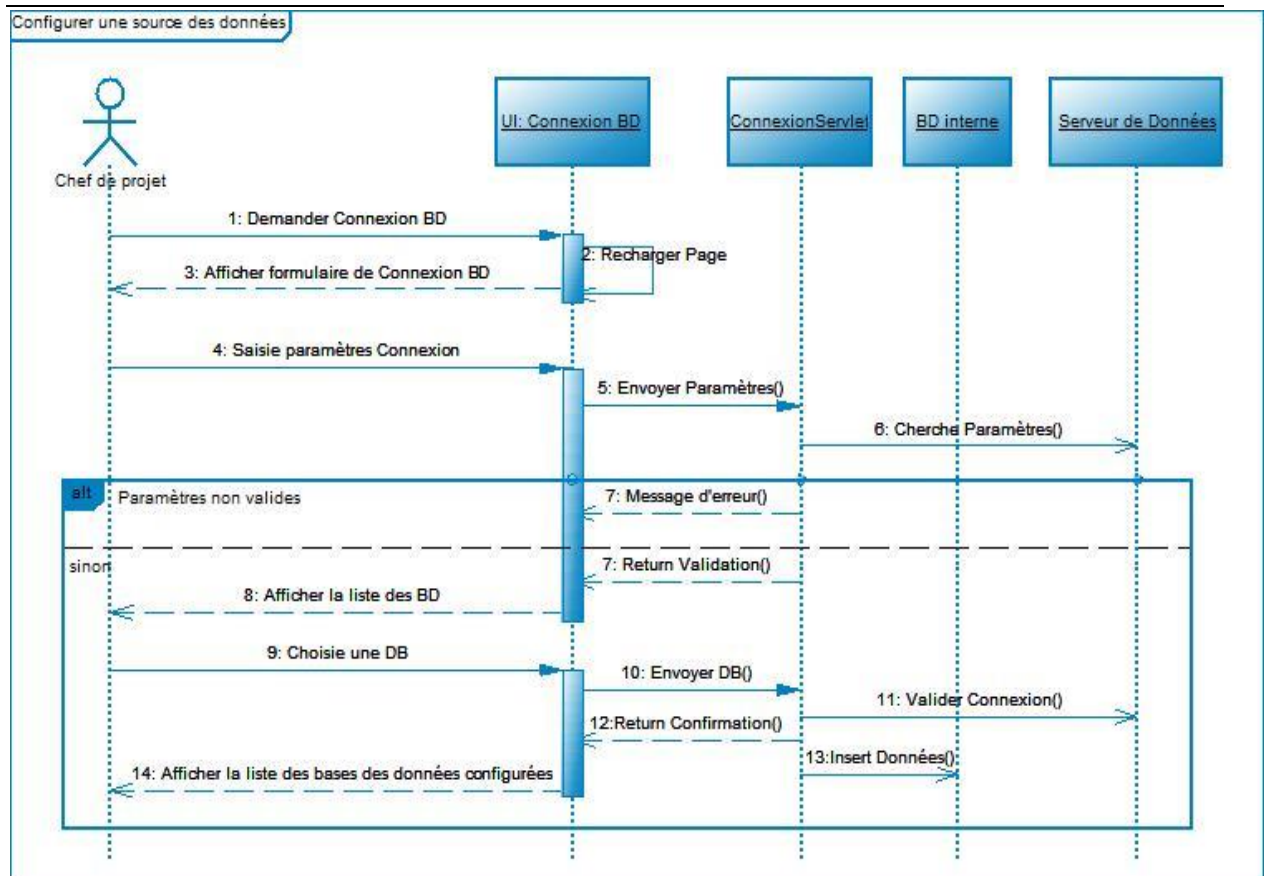


Figure 4.6. Diagramme de séquence « Configurer une source des données »

### 3.4. Diagramme d'activités : « Configurer une source des données »

La figure 4.7 présente le diagramme des différentes activités qui modélise le comportement qui nous offre l'opportunité de configurer une source des données afin d'avoir la possibilité de se connecter à une base de données externe.



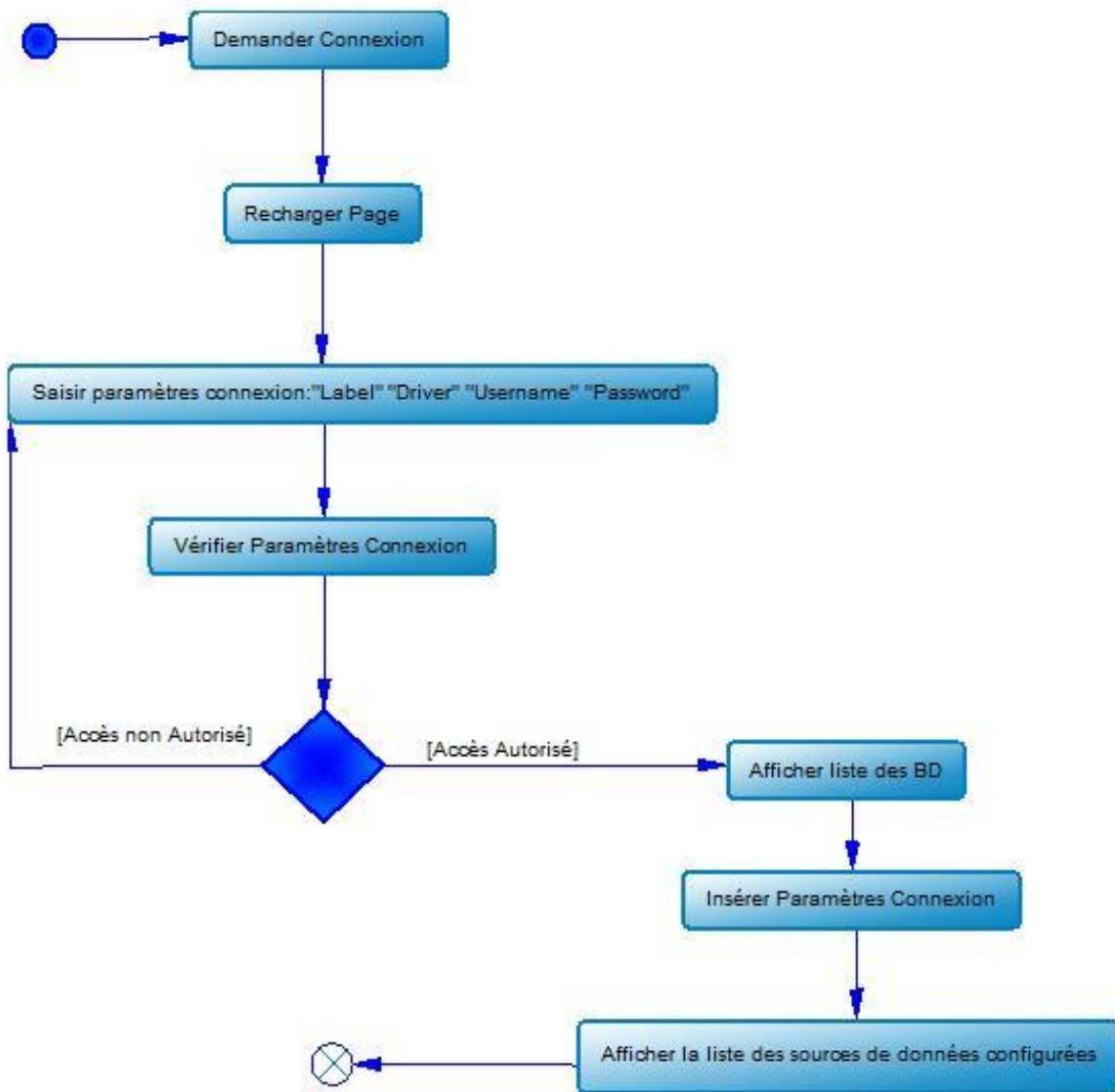


Figure 4.7. Diagramme d'activités « Configurer une source des données »

### 3.5. Diagramme de séquence : « Gérer une source des données »

La figure ci-dessous illustre le diagramme de séquence qui décrit le scénario de gestion d'une source de données. En effet, afin de pouvoir gérer une source de données, notre chef de projet doit se passer par le scénario de configuration d'une base de données.

En outre, il a le droit de consulter la liste des bases de données déjà configurées et de choisir une base de données afin de pouvoir récupérer leur données. Puis, il doit écrire une requête SQL dont les paramètres seront configurés par la suite.

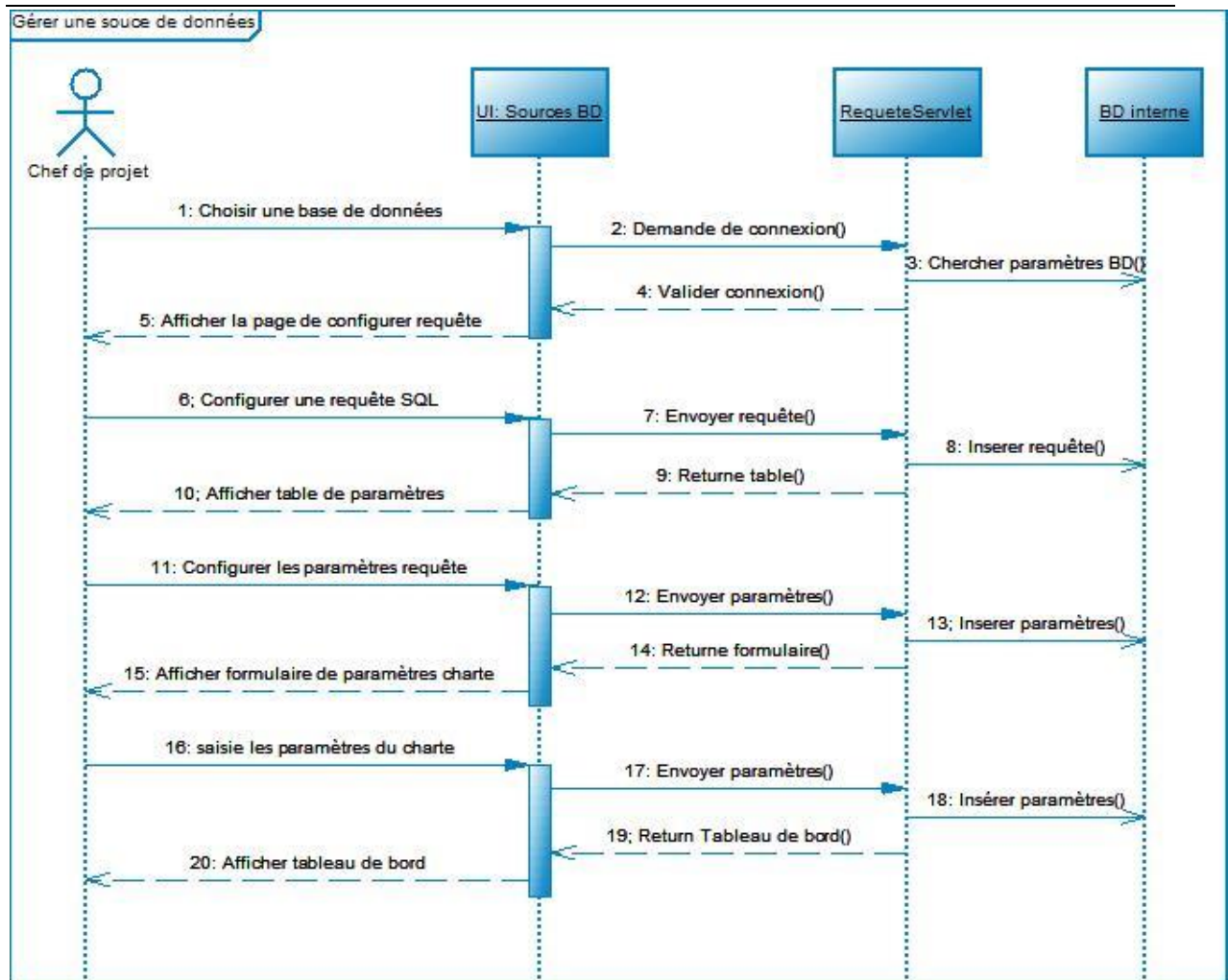


Figure 4.8. Diagramme de séquence « Gérer une source de données »

### 3.6. Diagramme d'activité : « Gérer une source de données »

La figure 4.9 présente le diagramme des différentes activités qui modélise le comportement qui nous offre l'opportunité de gérer une source des données afin d'avoir la possibilité de se configurer des nouveaux tableaux de bord.

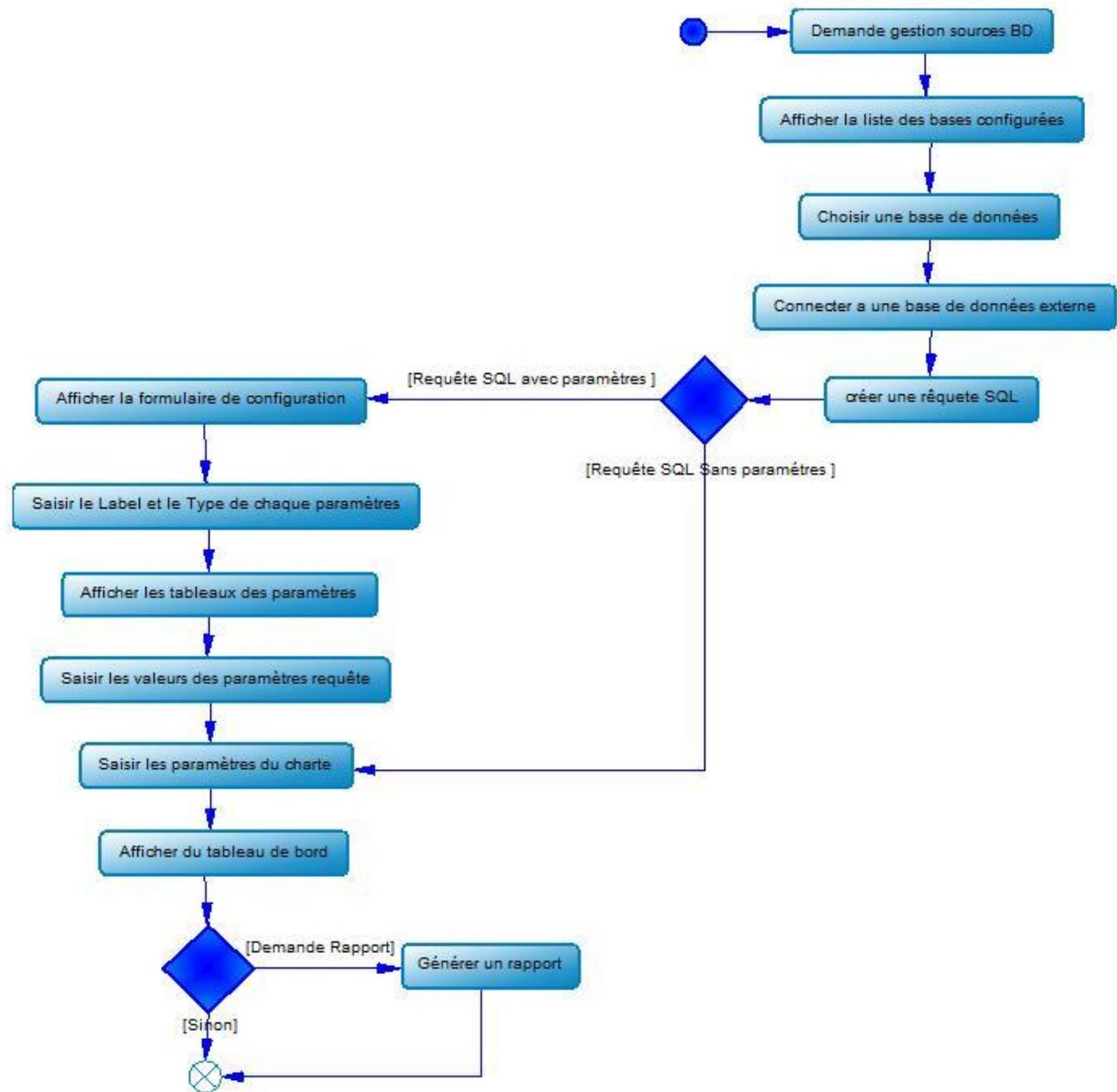


Figure 4.9. Diagramme d'activités « Gérer une source des données »

### 3.7. Diagramme de séquences : « Visualiser une charte graphique»

La figure ci-dessous illustre le diagramme de séquence qui décrit le scénario de visualisation d'une charte graphique. En effet, ce scénario nécessite la configuration des sources de données ainsi que des requêtes SQL afin de permettre notre agent de poursuivre dynamiquement l'évolution des indicateurs de l'entreprise.

En outre, notre agent a l'opportunité de choisir un tableau de bord déjà configuré par notre agent de projet. Lors de l'affichage de notre tableau de bord, notre agent a l'opportunité de

## Chapitre 4. Etude Conceptuelle

reconfigurer les l'axe d'abscisse et l'axe d'ordonnées de notre charte graphique. De plus, il peut modifier les paramètres de notre charte graphique tels que type et titre de charte.

D'autre part, après l'affichage du graphique configuré par notre administrateur et paramétré par notre agent, celui-ci peut avoir un rapport qui schématise l'évolution en temps réel des indicateurs de notre entreprise.

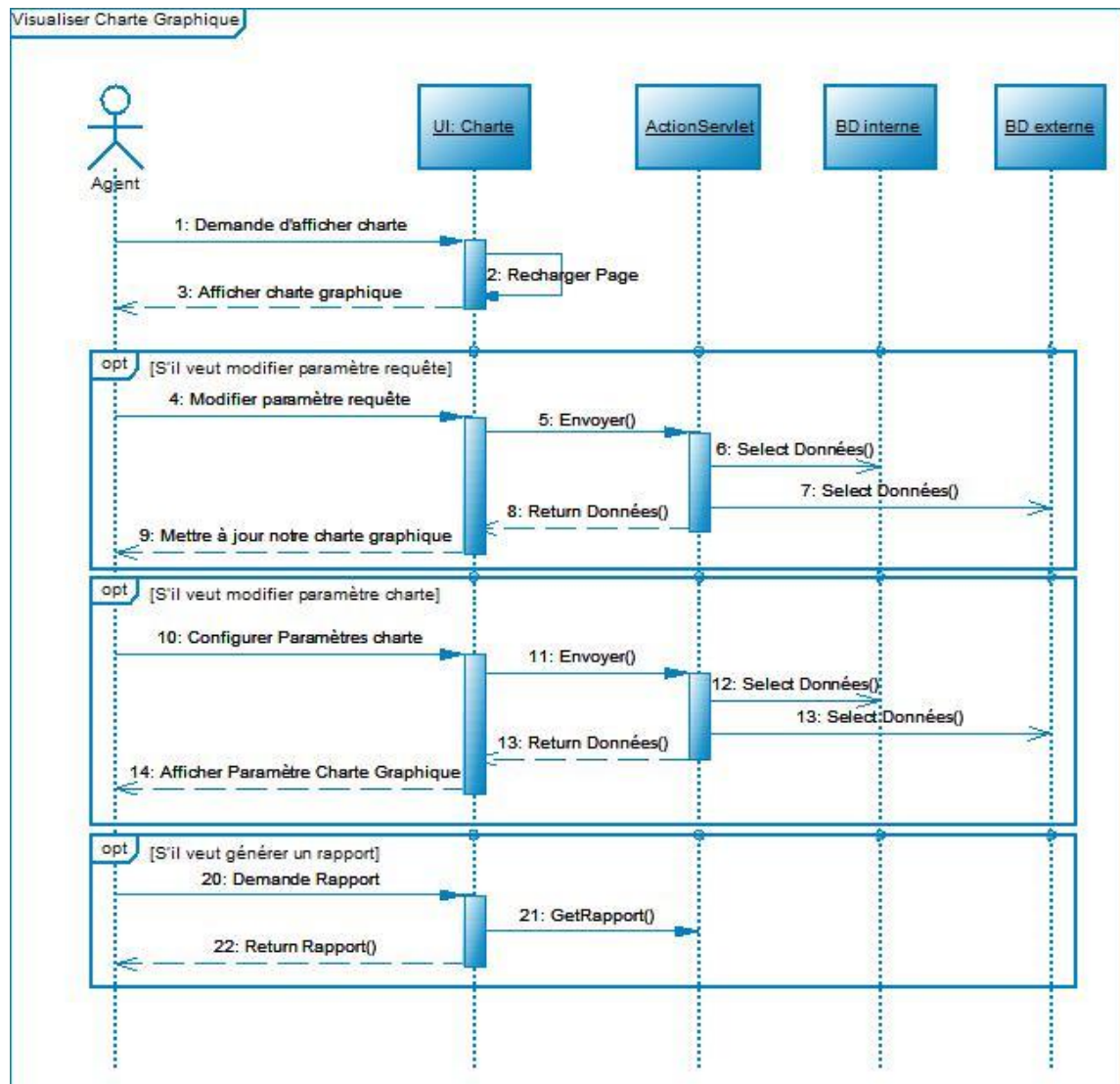


Figure 4.10. Diagramme de séquence « Visualiser une charte graphique »

## Conclusion :

Dans ce chapitre, nous avons traité la partie conceptuelle, primordiale pour la réalisation du projet. Nous avons étudié les vues statiques et dynamiques de notre projet. En fait, nous avons illustré des exemples de diagrammes UML relatifs à des opérations principales à savoir

## **Chapitre 4. Etude Conceptuelle**

---

« visualiser une table de donnée », « configurer un requête SQL », et « visualiser un graphique ».

La partie conceptuelle est essentielle pour faciliter l'étape de la « mise en œuvre » qui sera couverte par le chapitre suivant.

# CHAPITRE 5

## REALISATION

---

Le dernier chapitre modélise la partie réalisation de notre projet tout au long la période de stage.

---

## Chapitre 5 : Réalisation

### Introduction :

Dans le chapitre précédent, nous avons présenté la phase conception de notre application. En fait, nous avons détaillé cette phase en établissant les diagrammes UML tels que le diagramme de classe ainsi que les diagrammes de séquences et d'activités.

Dans ce chapitre, nous traitons les différentes étapes d'implémentation de l'application. En effet, nous commençons par introduire l'environnement de travail ainsi que l'architecture physique de notre application développée qui donne par la suite un aperçu sur le travail accompli au cours de la période de développement.

### 1. Environnement de développement :

#### 1.1. Langage de développement :

**Java** : est un langage de programmation qui a été créé par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable [13].

#### 1.2. Plateforme de développement :

**Hibernate** : permet de lier/mapper un objet défini en Java avec une table dans une base de données, via un fichier déclaratif de mapping. Le système peut s'occuper de la création des tables en fonction des fichiers de configuration et mettre aussi à jour les tables si nécessaire lors d'un changement dans l'un des fichiers de configuration [14].

**Spring** : est un Framework libre pour construire et définir l'infrastructure d'une application Java2, dont il facilite le développement et les tests [15].

#### 1.3. Outil de développement :

**Eclipse IDE for Java Developers** : est la version packagée pour faire principalement du développement en langage Java EE. Cependant, d'autres packages sont également disponibles afin de démarrer rapidement dans les domaines ou les langages tels que Java EE, C/C++, Eclipse RCP, Modeling, Reporting, PHP, etc.... [16]

### 1.4. Outil de conception :

**PowerDesign V12** : est un logiciel de conception créé par la société SDP, qui permet de modéliser les traitements informatiques et leurs bases de données associées. Il permet de modéliser tous les types de modèles informatiques. Et à ce niveau, nous allons travailler seulement avec les modèles orientés objets (MOO) dans le but de schématiser le modèle conceptuel de notre projet [17].

### 1.5. Support matériel et logiciel de développement :

Nous avons utilisé pour le développement un ordinateur portable ASUS avec un processeur i7 épaulé par 8GO de RAM. Le système d'exploitation présent sur la machine est Windows8. Pour le système de gestion de base de données utilisé dans le cadre de ce projet nous avons utilisé MySQL5.

### 1.6. Bibliothèque utilisée :

Afin de pouvoir créer des chartes, nous avons utilisé la bibliothèque Highcharts. C'est une bibliothèque de cartographie écrit en JavaScript pur, offrant un moyen facile d'ajouter des graphiques interactifs à votre site web ou une application web [18].

## 2. Architecture physique :

### 2.1. Architecture opérationnelle :

EN fait, notre application développée est une solution de l'informatique décisionnelle. Elle est destinée pour gérer les données de sources externes. En outre, elle nous permet comme le montre la figure suivante de se connecter à des sources externes afin de récupérer leurs données. De plus, elle interagit avec une base de données interne afin de pouvoir manipuler la configuration ainsi que les paramètres des tableaux de bord. Elle possède une interface web permettant de gérer les fonctionnalités de notre application.



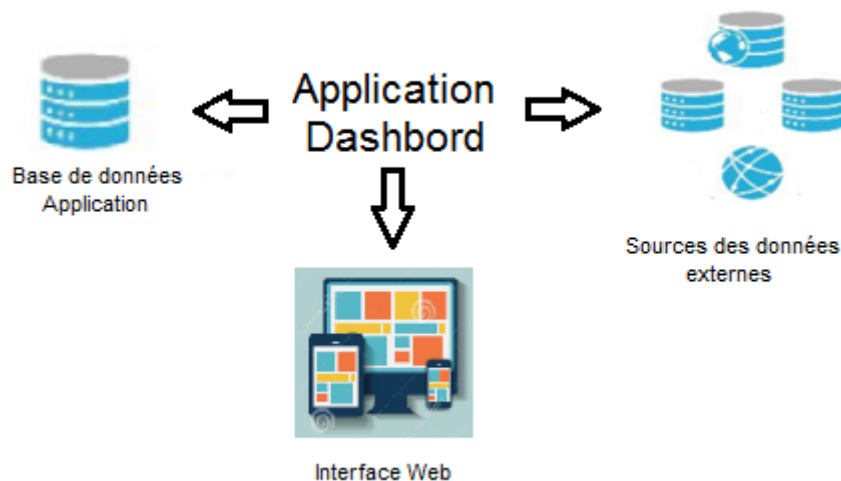


Figure 5.1. Architecture opérationnelle

### 2.2. Architecture technique :

Notre application contient cinq packages. En effet, la première package « `com.tn.bfi.hibernate` » présente un fichier de configuration de notre base de données ainsi qu'une classe `HibernateUtil` permettant de gérer les sessions Hibernate et de récupérer les données de notre base lors du lancement du programme. La deuxième package « `com.tn.bfi.entities` » contient les entités de notre base de données.

En outre, une couche dao contient deux packages l'une « `com.tn.bfi.dao` » contenant des interfaces qui liste les classes java prédéfinies dans le « `com.tn.bfi.daoImpl` ». Cette couche est chargée de l'accès aux données et de leur manipulation.

De plus, une couche métier contient deux autres packages la première « `com.tn.bfi.service` » et l'autre « `com.tn.bfi.serviceImpl` ». Cette couche fait le lien entre la couche d'accès aux données (couche dao) et les couches supérieures (contrôleur).

En fin, une dernière package « `com.tn.bfi.servlet` » contenant des servlets. En effet, ils sont chargés de présenter la couche web de notre application. D'où la redirection des pages jsp, les affichages et tous les traitements d'une page web.

La figure suivante présente le schéma de l'architecture technique de notre application qu'on vient juste de présenter ces différentes composantes :

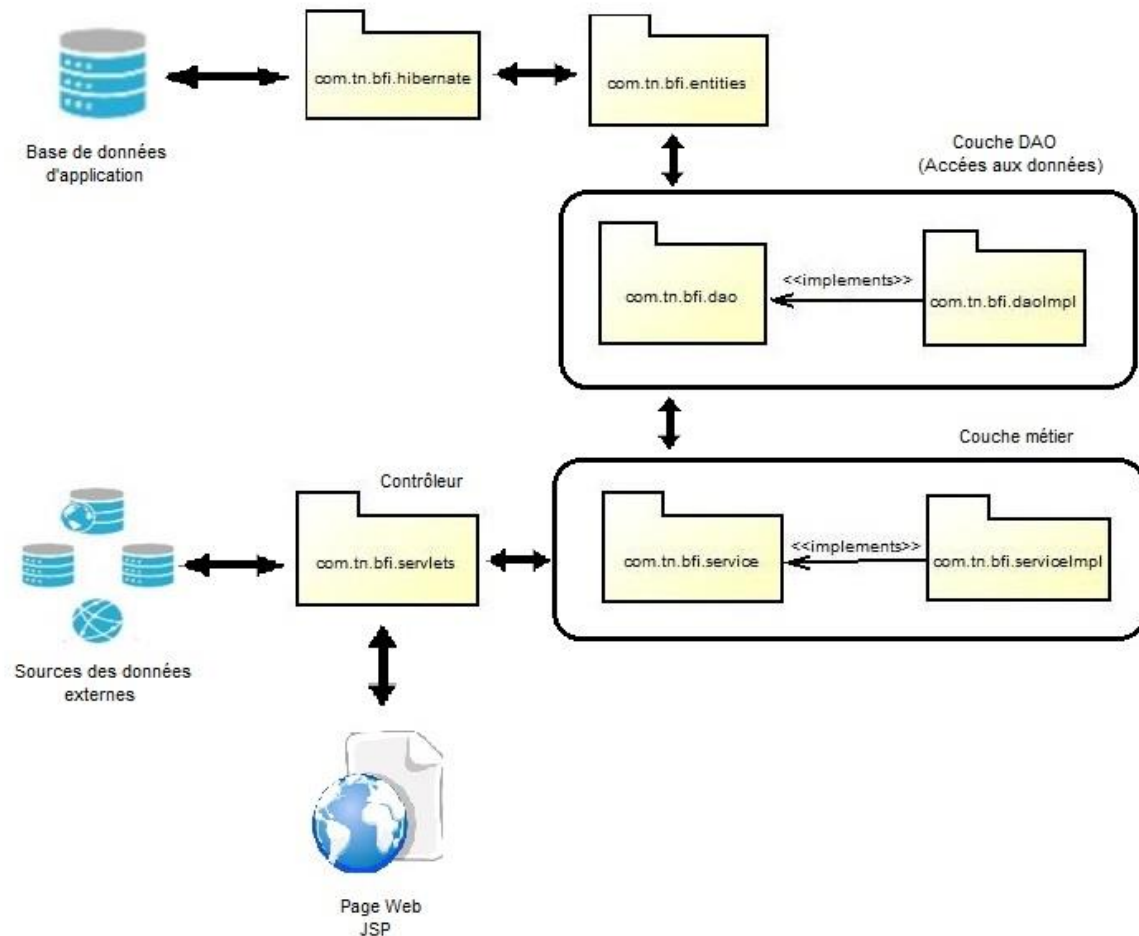


Figure 5.2. Architecture technique

### 3. Application Dashbord :

Dans cette partie, nous allons présenter notre travail réalisé. Pour ce faire, nous allons présenter les interfaces développées au cours de notre phase de développement afin de bien présenter les différentes fonctionnalités de notre application.

#### 3.1. Aperçu sur l'interface d'authentification :

L'interface principale qui apparaît après le lancement de l'application présente à l'utilisateur le choix de s'authentifier. En effet, notre utilisateur sera demandé à entrer son « Login » et son « Password » afin de pouvoir accéder à leur espace ou en cliquant sur « Mot de passe oublié ? » pour récupérer son mot de passe (Figure 5.3).



Figure 5.3. Interface d'accueil

En cas de saisir un login ou mot de passe incorrect, notre système va informer notre utilisateur qu'il faut vérifier votre login ou bien votre mot de passe (Figure 5.4).

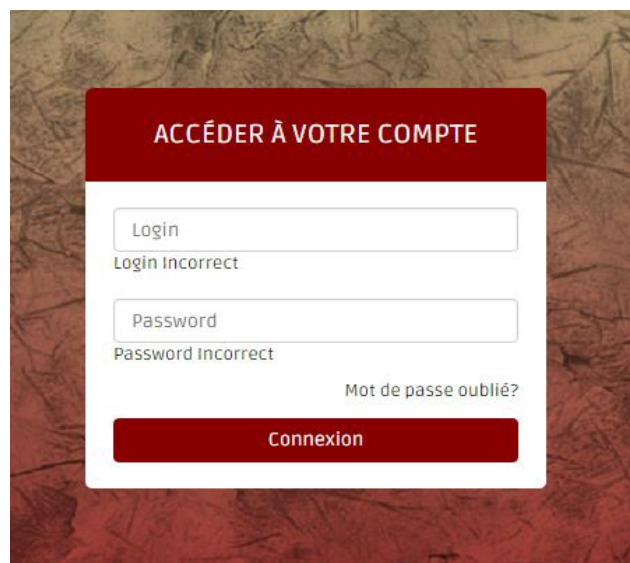


Figure 5.4. Authentification

### 3.2. Aperçu sur l'interface d'accueil :

Après l'authentification, notre utilisateur sera redirigé vers son espace selon leur droit d'accès. En effet, s'il a le droit de gérer notre application donc il sera redirigé vers l'interface du chef de projet (Figure 5.5) sinon vers l'interface d'agent (Figure 5.6) où il peut consulter la liste des tableaux de bord configurée par notre chef de projet.

## Chapitre 5. Réalisation

En outre, s'il s'agit d'un chef de projet. Donc il a l'opportunité de gérer les compte utilisateur, de gérer les connexions et les sources BD ainsi que les chartes configurées.

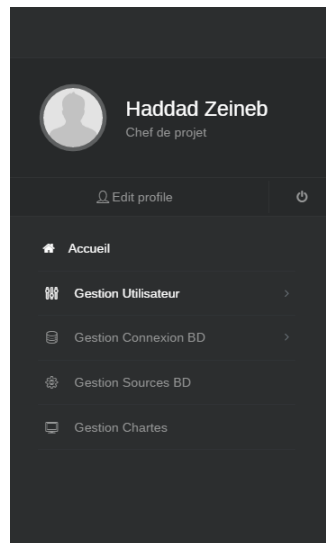


Figure 5.5. Menu d'accueil du chef de projet

S'il s'agit d'un agent, notre utilisateur n'a pas le droit que de consulter et de paramétrer les chartes déjà configurées par notre chef de projet afin de pouvoir mesurer leur valeur de performance.

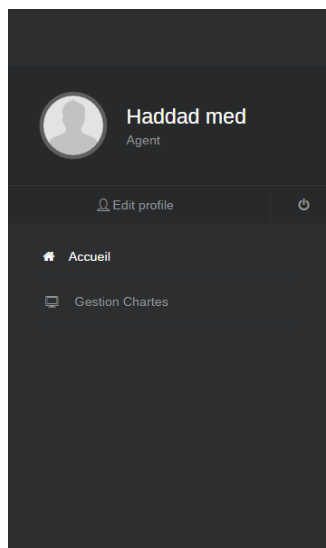


Figure 5.6. Menu d'accueil de notre agent

### 3.3. Gestion utilisateur :

Dans cette section, nous présentons les interfaces de gestion des comptes utilisateurs. En effet, notre chef de projet peut consulter la liste des utilisateurs (Figure 5.7) afin de pouvoir modifier leur profil ainsi que les supprimer.

MEMBRE ET ROLE	ADRESSE	COURRIEL	TEL	ACTION
Haddad med agent	tunis	mohamed.haddad@gmail.com	97434072	<a href="#">Modifier Utilisateur</a> <a href="#">Supprimer Utilisateur</a>
Bessi Selim chefDeProjet	tunis	Selim.Bessi@gmail.com	55000000	<a href="#">Modifier Utilisateur</a> <a href="#">Supprimer Utilisateur</a>
Haddad Zeineb chefDeProjet	Tunis	zeineb.haddad@gmail.com	53542812	<a href="#">Modifier Utilisateur</a> <a href="#">Supprimer Utilisateur</a>

Figure 5.7. Interface Liste Utilisateurs

Notre chef de projet peut aussi ajouter un nouvel utilisateur en spécifiant leur rôle (Figure 5.8).

Nom

Prenom

Adresse

Courriel

Téléphone

Role

Login

Password

Figure 5.8. Interface Ajout Utilisateur

## Chapitre 5. Réalisation

De même, il peut manipuler les coordonnées d'un compte utilisateur déjà inscrit (Figure 5.9).

The screenshot shows a web application interface for 'Haddad Zeineb, Chef de projet'. The main content area is titled 'Connexion BD' with the subtitle 'Modifier les paramètres d'une connexion configurée.' The breadcrumb trail is 'Accueil / Gestion Connexion BD / Connexion BD'. The form contains the following fields and controls:

- 'Base de données de statistique': A text input field.
- 'Driver': A dropdown menu currently showing 'MySql'.
- 'Paramètres Connexion': A section containing a 'root' text input field.
- 'Password Connexion': A text input field.
- 'Connect': A button.
- 'liste des bases de données': A dropdown menu currently showing 'dashbord'.
- 'Modifier': A button.

Figure 5.9. Interface Modifier Utilisateur

### 3.4. Gestion Connexion BD :

Dans cette section, nous présentons la gestion des connexions BD. En effet, notre chef de projet a l'opportunité de configurer des nouvelles connexions. Alors, notre application sera accessible à se connecter à des bases de données externes.

En outre, nous présentons une interface qui nous permet de consulter la liste des bases de données configurées afin de les manipuler par la suite (Figure 5.10).

The screenshot shows the 'Ressources BD' interface with the subtitle 'Afficher les sources des données configurées.' The breadcrumb trail is 'Accueil / Gestion Connexion BD / Ressources BD'. The table below lists the configured database resources:

TYPE DE DRIVER	NOM BD	ACTION
com.mysql.jdbc.Driver	abc	<a href="#">Modifier Connexion</a> <a href="#">Supprimer Connexion</a>
com.mysql.jdbc.Driver	mysql	<a href="#">Modifier Connexion</a> <a href="#">Supprimer Connexion</a>
com.mysql.jdbc.Driver	dashbord	<a href="#">Modifier Connexion</a> <a href="#">Supprimer Connexion</a>
com.mysql.jdbc.Driver	essai	<a href="#">Modifier Connexion</a> <a href="#">Supprimer Connexion</a>

© 2016 Dashboard - Développer par Zeineb Haddad

Figure 5.10. Liste des bases de données configurées

## Chapitre 5. Réalisation

Nous présentons par la suite une interface permettant de configurer des nouvelles connexions BD (Figure 5.11).

The screenshot shows a web application interface for configuring a new database connection. On the left is a dark sidebar with the user's name 'Haddad Zeineb' and role 'Chef de projet', along with navigation links: 'Accueil', 'Gestion Utilisateur', 'Gestion Connexion BD' (selected), 'Ressources BD', 'Connexion BD', 'Gestion Sources BD', and 'Gestion Chartes'. The main content area is titled 'Connexion BD' with the subtitle 'Configurer une source des données.' and a breadcrumb trail 'Accueil / Gestion Connexion BD / Connexion BD'. The form contains the following fields: 'Base de données' (text input), 'Driver' (dropdown menu with 'MySql' selected), 'Paramètres Connexion' (text input with 'root'), 'Password Connexion' (text input), and 'liste des bases de données' (dropdown menu). There are 'Connexion' and 'Valider' buttons. The footer indicates '© 2016 Dashboard - Développer par Zeineb Haddad'.

Figure 5.11. Formulaire de configurer une nouvelle connexion BD

De même, nous présentons une interface qui nous permet de modifier les coordonnées d'une connexion BD déjà configurée (Figure 5.12).

The screenshot shows the same web application interface but for modifying an existing database connection. The sidebar and main header are identical. The main content area is titled 'Connexion BD' with the subtitle 'Modifier les paramètres d'une connexion configurée.' and the same breadcrumb trail. The form contains the following fields: 'Base de données de statistique' (text input), 'Driver' (dropdown menu with 'MySql' selected), 'Paramètres Connexion' (text input with 'root'), 'Password Connexion' (text input), and 'liste des bases de données' (dropdown menu with 'dashboard' selected). There are 'Connect' and 'Modifier' buttons.

Figure 5.12. Formulaire de reconfigurer une connexion BD

### 3.5. Gestion sources BD :

Dans cette section, nous présentons la gestion des sources BD. En effet, notre chef de projet a l'opportunité de se connecter à une base de données externe afin de pouvoir créer des requêtes SQL dynamique.

Nous présentons par la suite une interface (Figure 5.13) qui permet d'afficher la liste des bases de données configurées afin de pouvoir se connecter et de récupérer les données nécessaires pour créer des chartes sur mesure.

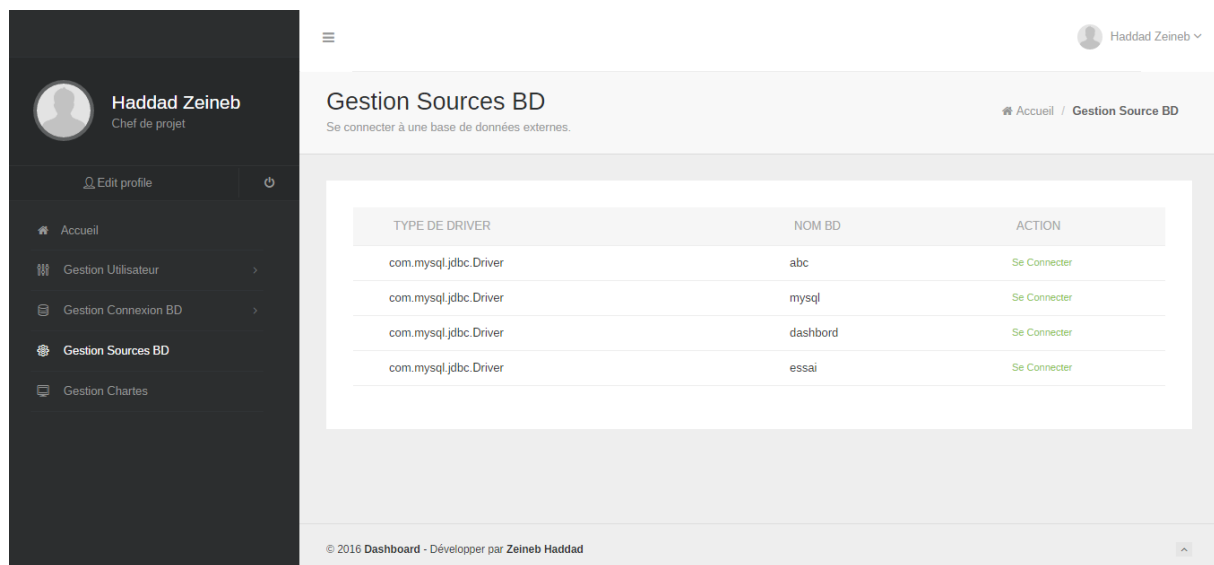


Figure 5.13. Interface de connexion BD

Après la connexion à une base de données externe, notre chef de projet doit créer une requête SQL. En effet, nous présentons par la suite une interface de configuration requête (Figure 5.14) permettant au premier lieu de créer une requête SQL puis en cliquant sur le bouton « valider la requête » un tableau de paramètres apparait dans la même interface afin de pouvoir configurer les paramètres du requête saisie.



**Requête SQL**  
Configurer votre requête SQL.

Accueil / Gestion Sources BD / Requête SQL

**Source des données**

Les paramètres de notre base de données externe:

Driver	com.mysql.jdbc.Driver
Username	root
Password	
Nom BD	essai

**Configurer une requête SQL**

Saisie votre requête SQL:

```
select mois, humidite from temperature where ville=? and annee=?
```

Valider la requête

**Configurer paramètres requête SQL**

Nom parametre	Type	Label
Parametre 1(ville)	<input type="radio"/> Date <input checked="" type="radio"/> String <input type="radio"/> Entier <input type="radio"/> Double	Nom ville
Parametre 2(annee)	<input type="radio"/> Date <input checked="" type="radio"/> String <input type="radio"/> Entier <input type="radio"/> Double	Année

Valider les paramètres

Figure 5.14. Configurer requête SQL

Puis, en cliquant sur le bouton « valider les paramètres », notre chef de projet va être dirigé vers une nouvelle interface (Figure 5.15) afin de pouvoir configurer les paramètres de la charte graphique.

**Paramètres Requête**  
Configurer les paramètres de votre requête.

Accueil / Gestion Sources BD / Paramètres Requête

**Tableau des paramètres**

Nom libelle	Nom parametre	Valeur
Nom Ville	ville	Tunis
Année	annee	2016

mois

humidite

**Paramètres Charte graphique**

Nom chart: Temperature 2016

Nom ligne: Par Mois

Type chart: ☐ Basic Column ☐ Basic Line ☒ Scatter Plot ☐ 3D

Valider

© 2016 Dashboard - Développer par Zeineb Haddad

Figure 5.15. Configurer paramètres charte

## Chapitre 5. Réalisation

En cliquant sur le bouton valider, notre agent va être redirigé vers une autre interface (Figure 5.16) permettant d'afficher le résultat de la requête. Cette interface permet de modifier les paramètres d'affichage d'une charte graphique tels que titre, type, axe abscisse et cordonnées....

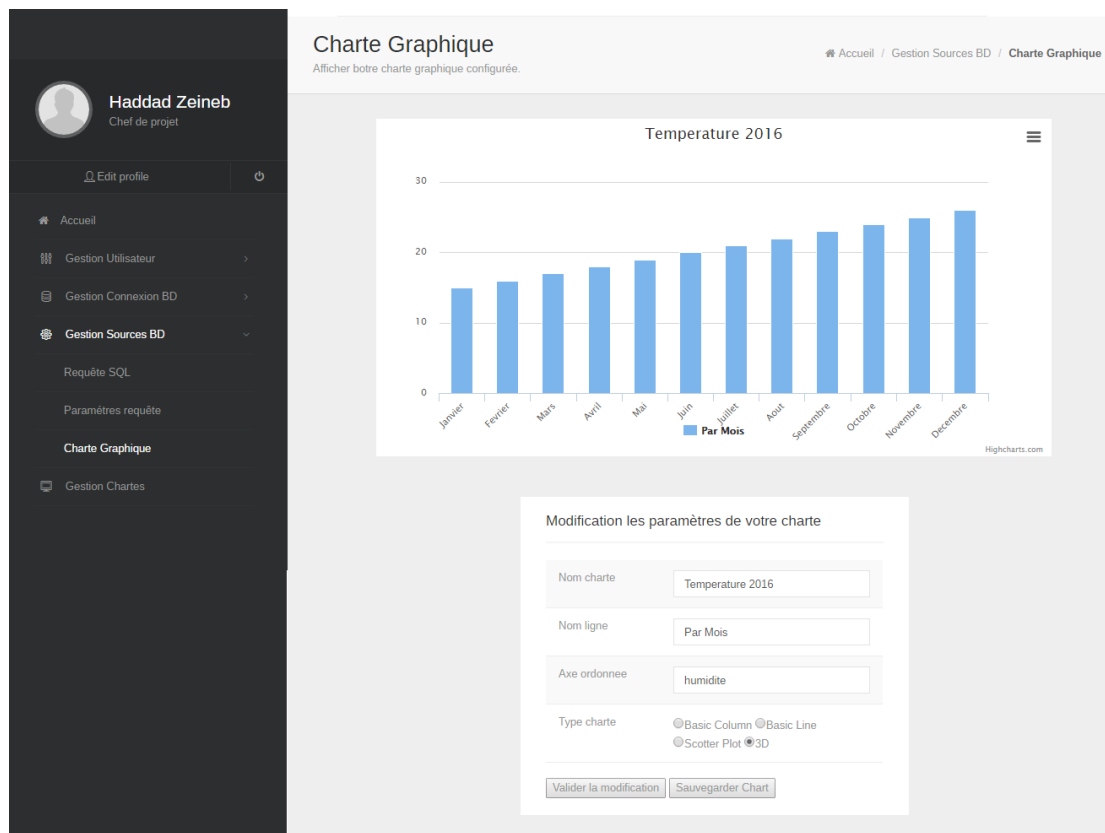


Figure 5.16. Interface de résultat requête

### 3.6. Gestion chartes :

Dans cette section, nous présentons la gestion des chartes. En effet, après avoir configuré les chartes, notre chef de projet peut consulter la liste des chartes disponible. De plus, il peut les manipuler en modifiant leurs paramètres de collection des données et même de supprimer une charte enregistrée dans notre base de données. EN revanche, notre agent n'a pas le droit de supprimer une charte mais il peut modifier leurs paramètres.

La figure 5.17 présente une interface agent permettant de consulter une charte configurée par notre chef de projet.

## Chapitre 5. Réalisation

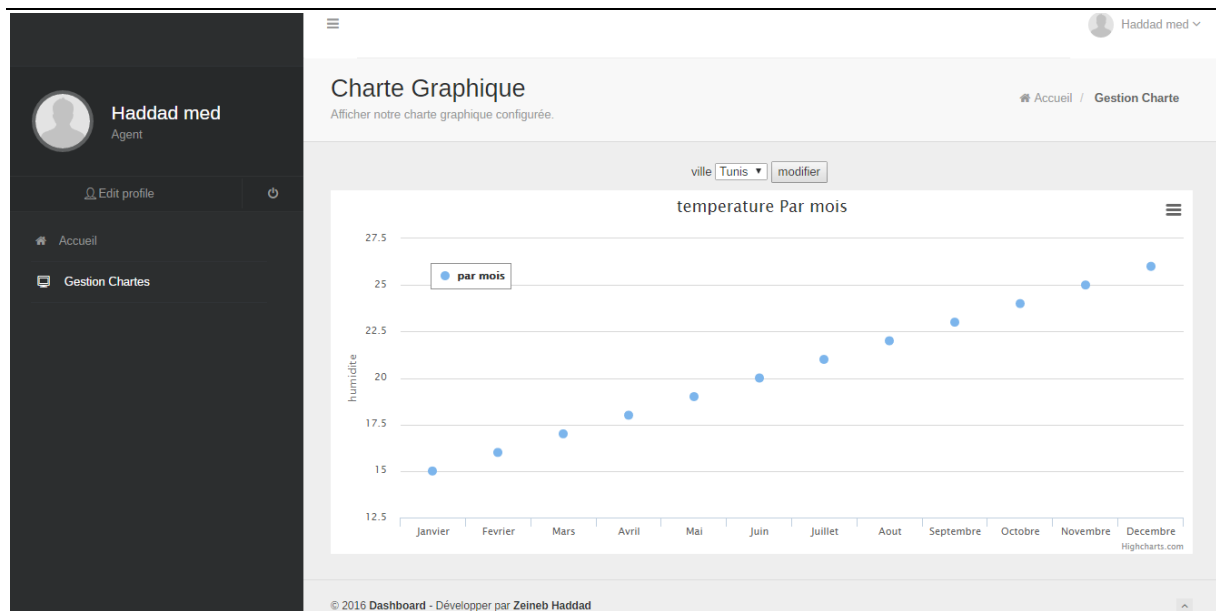


Figure 5.17. Interface agent de consultation charte

## Conclusion :

Ce chapitre nous a permis de présenter la mise en œuvre notre conception sous forme d'une application Java Entreprise Edition. Nous avons commencé par présenter l'environnement de développement. Par la suite, nous avons présenté quelques scénarios ainsi que les interfaces développées dans notre application afin de donner une idée claire sur le travail réalisé.

# Conclusion générale

Ce travail a été effectué dans le cadre de projet de fin d'études réalisé pendant quatre mois au sein de l'entreprise BFI. Dans ce projet, nous avons conçu et développé une application de génération de tableaux de bord. Le présent manuscrit détaille toutes les étapes par lesquelles nous sommes passées pour arriver au résultat attendu.

Nous avons débuté par comprendre le contexte général du projet et les exigences du futur système. Pour ce faire, nous avons passé par la partie recherche afin de pouvoir étudier quelques solutions existantes dans le marché. Nous avons préparé, par la suite, la démarche de réalisation du projet en respectant les priorités par rapport aux besoins exprimés par BFI et les objectifs qu'elle souhaite atteindre.

Afin de mettre en œuvre notre application, nous avons utilisé les diagrammes UML dans la partie conception afin de décrire la vue statique et dynamique du notre système. Ainsi que nous avons utilisé la plateforme Java EE pour développer les modules de notre projet.

Malgré les contraintes rencontrées, notamment la limite de temps et d'accès aux bases de données clients, nous avons réussi à réaliser une application web permettant de se connecter à des sources des données externes et pouvant collecter les données nécessaires afin de générer des tableaux de bord sur mesure.

Par ailleurs, ce stage nous a donné la chance de s'implique d'avantage dans le domaine de l'informatique décisionnelle ainsi que de manipuler des techniques innovantes. En outre, ce projet nous a permis d'approfondir nos connaissances acquises au sein de l'Ecole Nationale Supérieure d'Ingénieurs de Tunis comme il nous a fourni l'occasion de s'intègre dans la vie professionnelle. Cette expérience nous a permis de mettre en pratique et dans un même projet nos connaissances dans le domaine de la conception et de développement.

Dès le départ de ce projet l'idée était de construire le noyau d'une solution logicielle qui sera ensuite améliorée pour devenir une solution commercialisable. Ainsi, cette première version peut être améliorée. Parmi les perspectives d'amélioration de ce projet nous avons l'aspect sécurisé informatique et la protection des accès aux données est très importante dans certains secteurs d'activité. Egalement, même si les bases de données représentent la source de données la plus utilisée actuellement pour la gestion de données, il faut pouvoir gérer le cas des fichiers plats en XML, JSON ou autres formats.

## Référence

- [1] <http://www.bfigroupe.com/> consulté le 01/02/2016
- [2] <http://www.journaldunet.com/solutions/intranet-extranet/business-intelligence/> consulté le 20/02/2016
- [3] [http://capirossi.org/info/prj\\_mgt/cycle\\_6.htm](http://capirossi.org/info/prj_mgt/cycle_6.htm) consulté le 16/02/2016
- [4] <http://www.lavastorm.com/blog/2016/03/21/the-evolution-of-business-intelligence/> consulté le 16/02/2016
- [4] <https://www.mapr.com/blog/evolution-business-intelligence-and-self-service-analytics-whiteboard-walkthrough> consulté le 16/02/2016
- [5] O.Boussaid « Les entrepôt de données – le processus ETL » Eric Univ Lyon, 2014. consulté le 17/02/2016
- [6] <http://www.talendforge.org/> consulté le 15/02/2016
- [7] <http://www.pentaho.com/> consulté le 15/02/2016
- [8] <http://www.qlik.com/> consulté le 01/02/2016
- [9] <http://www.spagobi.org/> consulté le 07/03/2016
- [10] Pascal Roques, « Livre UML 2 par la pratique » EYROLLES, Septembre 2006 consulté le 28/06/2016
- [11] <https://aresu.dsi.cnrs.fr/spip.php?article95> consulté le 28/03/2016
- [12] <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/mieux-structurer-son-code-le-pattern-mvc> consulté le 28/03/2016
- [13] [https://www.java.com/fr/download/faq/whatis\\_java.xml](https://www.java.com/fr/download/faq/whatis_java.xml) consulté le 28/06/2016
- [14] <http://hibernate.org/> consulté le 28/06/2016
- [15] <https://spring.io/> consulté le 28/06/2016
- [16] <http://www.eclipse.org/> consulté le 28/06/2016

[17] <http://www.intercambiosvirtuales.org/software/sybase-powerdesigner-v125-la> consulté le 28/06/2016

[18] <http://www.highcharts.com/> consulté le 21/03/2016

## Résumé

Ce travail s'inscrit dans le cadre de projet de fin d'études réalisé au sein de l'entreprise BFI. L'objectif principal de ce travail consiste à développer une application web basée sur la plateforme Java EE pour la gestion des tableaux de bord en temps réel.

Cette application permettra aux chefs de projets de récupérer des données stockées dans des sources de données externes afin de générer des tableaux de bord dynamiques et paramétrables.

**Mots clés :** Java EE, Informatique décisionnelle, Tableau de bord, Dynamique, Paramétrable, Temps réel.

## Abstract

This work has been performed in the framework of a graduation project done at BFI. The main objective of this work consists in developing a web application using the Java EE platform, in order to manage dash-boards in real time.

This application will let the project managers to recover data stored on external databases in order to generate dash boards that are dynamic and configurable.

**Keywords:** Java EE, Business Intelligence, Dash Board, Dynamic, Configurable, Real time.

## ملخص

يندرج هذا العمل في إطار مشروع التخرج الذي أنجز في مؤسسة بي إف أي. ويتمثل الهدف الرئيسي لهذا العمل في تطوير تطبيق ويب يعتمد على برنامج Java EE بهدف التصرف في "لوحات القيادة" في الوقت الفوري.

وسيمكن هذا التطبيق رؤساء المشاريع من تجميع البيانات المخزنة في قواعد بيانات خارجية، وإنتاج "لوحات القيادة" حيوية ومتغيرة .

الكلمات المفتاحية : ، Java EE الاستخبارات التجارية، لوحة قيادة، حيوي، متغير، الوقت الفوري .