

# Dossier Technique

## Projet BDCFOU V2.0.0

BARREH BOUNI Farhan

CDA7 Session 2021.



CENTRE NATIONAL D'ÉTUDES SPATIALES

### Tuteur

BAT Philippe

### Formateur

DANGU pascal

# Avertissement

La société Inetum demeure unique propriétaire des documents et données liés aux réalisations sur l'outil BDGFOU. Aucun transfert des composants de l'outil tels que le code, les données ou la documentation technique ne peut être réalisé sans accord préalable écrit. Il a été validé auprès du tuteur, BAT Philippe, qu'aucune donnée à caractère confidentiel n'a été transmise au titre de ce rapport technique.

# Remerciements

Avant tout, je tiens à remercier toutes les personnes qui ont contribué au succès de ce stage et qui ont aidé à ce que cette reconversion se réalise :

- L'entreprise INETUM et plus particulièrement le service **Digital Factory** de la **Division AS Sud-Ouest** pour leur accueil et leur convivialité et pour le stage qui m'a été proposé.
- BAT Philippe, le Delivery Manager du Digital Factory de la branche AS et mon tuteur pour la proposition de stage, son accueil chaleureux et sa disponibilité en tout temps.
- DI-MEGLIO Benoît responsable service, pour son immense aide à la compréhension du cœur du métier, sa disponibilité, son investissement, son soutien technique, ses encouragements et son professionnalisme.
- GAUZE Marie notre cheffe de projet, pour son professionnalisme, son accueil, son humour, sa gentillesse et son soutien.
- DERKSEN Alexandre, développeurs senior java pour son encadrement de tous les jours, son soutien, son investissement, Ses explications techniques, les conseils et les encouragements etc...
- Je remercie également toute l'équipe pédagogique de l'AFPA qui ont œuvré pour que ma reconversion se réalise dans des bonnes conditions. Notamment DANGU Pascal mon Formateur référent pour sa transmission de savoir, sa connaissance et ses conseils avisés, GALLAIS Patricia et COLLIN Bruno, Svetlana POPLAVSKAYA et Philippe VIGUIER pour leur professionnalisme, leur soutien et leurs encouragements.
- Je tiens particulièrement à remercier mon épouse pour son soutien indéfectible, sa patience pendant cette année d'éloignement et ses encouragements constants.

# Abstract

After 10 years of teaching experience and 5 years of part-time IT support which made me want to learn about IT development, I completely changed my field of work in favor of computer code. To achieve this goal, I started training as an IT Designer and Developer at the AFPA center in Balma and in order to complete my training, I did an internship of two and a half months in a real IT project within the service. Digital Factory from Inetum.

The Digital Factory is a team specialized in the realization of B2B and B2C web projects. They intervene throughout the value chain: industrial protocols as close as possible to the machines, complete support for projects. Today, the Digital Factory works with various clients in various fields of activity (Medical, Finance, Energy, Services, Aeronautics, Space etc.).

During my internship in the Digital factory, I was in charge of the development of corrective and update maintenances of the BDGFou application. This application allows users to view information about suppliers. I started my work with a modeling phase, using design diagrams, writing documentation with a detailed description of all expected features.

Then I started the development of the functionalities requested by the customer (Addition of new functions and fixes), performed a database anonymization procedure, then established the operating documentation and finally the tests and code quality.

During this internship, I had the chance to be trained in theory and practice on technologies such as GWT, Unix or the use of virtual machines. This period has been very rewarding for me professionally.

This report presents my activity during my internship and I hope you will enjoy reading it.

# Table des matières

<b>1. Introduction</b>	<b>6</b>
<b>2. Présentation de l'entreprise</b>	<b>7</b>
2.1. Historique et Présentation du groupe	7
2.2. Présentation de l'agence	7
2.3. Organisation des services	8
A. Secteurs d'Activités	8
B. DIGITAL FACTORY	8
<b>3. Présentation du Projet</b>	<b>9</b>
3.1. Architecture de l'Application BDCFou	10
3.2. L'Application Web BDCFou	11
A. Les différents profils	11
B. La fonction de recherche	13
C. L'IHM d'export des données	13
D. L'IHM d'édition des fournisseurs	13
3.3. Contexte et Objectifs du projet	14
3.4. Cahier de charges	15
3.5. La Gestion du projet	15
A. Programation du déroulé	15
B. Repartition des tâches	16
3.6. Productions attendues	16
3.7. Environnement et Contraintes techniques	17
A. Environnement Technique	17
B. Contraintes du Projet	20
<b>4. Analyse des Besoins</b>	<b>21</b>
4.1. Cas d'utilisation de BDCFou	21
4.2. Schéma de démarrage de l'application	22
4.3. La BDD BDCFou	22
<b>5. Conception</b>	<b>22</b>

5.1.	Les différents modules composant l'application	22
5.2.	La fonction principale de l'application « Recherche d'un fournisseur ».	25
<b>6.</b>	<b>Implémentation</b>	<b>25</b>
6.1.	Application des maintenances correctives et évolutives	25
A.	Demande de Modification 00011	26
B.	Demande de Modification 00021	26
C.	Demande de Modification 00023	27
D.	Demande de Modification 00027.1 et 00027.2	28
E.	Demande de Modification 00028	29
F.	Demande de Modification 00029	29
6.2.	Création d'une procédure d'anonymisation	29
A.	Procédure et environnement d'exécution	30
B.	Script SQL permettant l'anonymisation	32
C.	Script Bash exécutant les scripts SQL	34
<b>7.</b>	<b>Tests et validation</b>	<b>34</b>
A.	Cahier de recettes	34
B.	Vérification de la qualité avec SonarQube	35
<b>8.</b>	<b>Bilan du projet</b>	<b>35</b>
<b>9.</b>	<b>Bilan personnel et perspectives</b>	<b>35</b>
<b>10.</b>	<b>Annexes</b>	<b>36</b>

# 1. Introduction

Après avoir choisi la voie de la reconversion dans le métier de développeur, j'ai entamé une formation de concepteur développeur d'application dispensée par l'AFPA de Toulouse de Janvier 2021 à décembre de la même année.

Pour mettre en pratique les acquis durant la formation, il est nécessaire de réaliser une pratique d'autorisation d'exercice (PAE) de trois mois en entreprise. C'est dans ce cadre que j'ai effectué un stage, du 15 septembre au 10 décembre 2021, au sein de la division AS SUD-OUEST de la société Inetum, pour valider les compétences acquises et les mettre en application.

Mon choix s'est porté sur ce stage non seulement pour son caractère évolutif d'une application développée sous une technologie ancienne et qui tourne déjà en production, mais aussi son adéquation sur la mise en œuvre des technologies coïncidant avec la formation que j'ai reçue.

Tout au long de cette période d'application, l'objectif général de ce stage était l'élaboration de la version 2.0.0 de l'application BDCFou, demandée par le CNES. Pour ce faire j'ai dû mobiliser les connaissances acquises durant la formation afin de déterminer un certain nombre d'objectifs spécifiques pour atteindre ce dernier, à savoir :

- Découvrir le cœur de métier de l'application et intégrer l'environnement de développement.
- Monter en compétence sur le Framework sur lequel est développé BDCFou.
- Planifier les tâches à réaliser.
- Faire une analyse des besoins du projet.
- Réaliser les maintenances évolutives et correctives demandées par le CNES.
- Rédiger la documentation nécessaire.
- Déployer la version 2.0.0 sur l'environnement usine d'Inetum pour test.

## 2. Présentation de l'entreprise

### 2.1. Historique et Présentation du groupe

L'entreprise Inetum, autrefois connue sous le nom « Groupe Informatique Français » (Gfi) est une **entreprise de service numérique** (ESN) inscrite dans une dimension internationale. Créée en 1970, elle comptait déjà 12 500 collaborateurs en 2009.



A la suite du rachat de IECISA en Espagne en octobre 2020, Gfi a choisi de se rebaptiser « Inetum » comme « *incrementum* », signifiant « croissance » en latin. Ce coup de communication permet d'affirmer sa position, puisque cette société compte actuellement **32 agences dans plus de 26 pays**. Cette croissance s'explique par une stratégie proactive de fusions-acquisitions associée à une forte croissance organique, souvent supérieure à celle du marché. Aujourd'hui le groupe compte près de **27 000 collaborateurs**, soit une augmentation de 300% de ses équipes.

Leur chiffre d'affaires est estimé à **1 966 millions d'euros en 2020**. Un chiffre que le président et directeur général, Vincent ROUAIX, souhaiterait faire grossir aux alentours de 3 milliards d'euros dans les années à venir.

La volonté d'Inetum est explicitée dans son slogan « Positive digital flow », c'est-à-dire de soutenir la transformation digitale des entreprises. Sa particularité est d'accompagner sur la durée de façon évolutive et personnalisée, grâce à une maîtrise agile qui permet de s'adapter aux besoins de leurs clients.

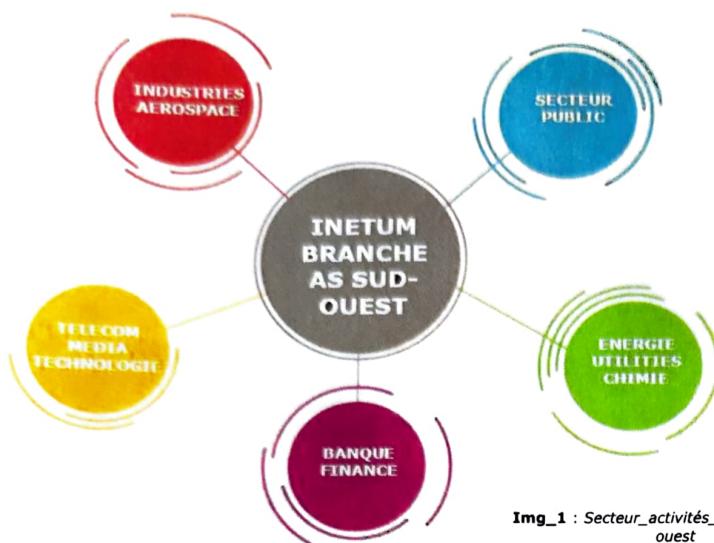
### 2.2. Présentation de l'agence

S'inscrivant dans cette dynamique de « Positive digital flow », la Branche Sud-Ouest constitue un pôle important sur le territoire. Avec ses 940 consultants et son chiffre d'affaire de 108 millions d'euros et ses 5 centres de compétences (ALM/PLM, SIG, MOBILITE, SOLUTIONS TELECOMES et PERFORMANCE ET TRANSFORMATION), cette dernière quadrille le sud-ouest avec quatre (4) implantations : Toulouse, Bordeaux, Pau et Orthez.

L'agence de Toulouse dans laquelle j'ai fait mon stage appartient à la Branche Sud-Ouest et est la plus importante.

## 2.3. Organisation des services

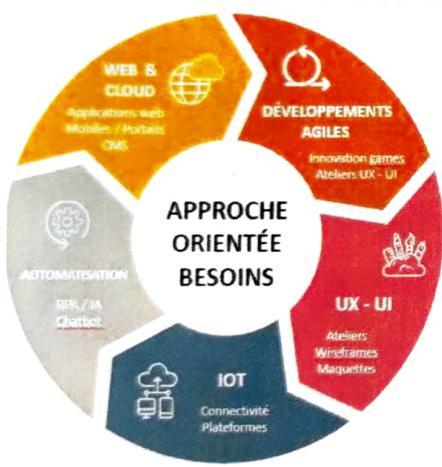
### A. Secteurs d'Activités



Inetum AS Sud-Ouest (AS pour Application Services) est organisée en plusieurs pôles représentant des secteurs d'activités diverses. Ces pôles entretiennent des partenariats avec différentes entreprises comme : Le **Ministère de la Justice français**, **Tigo, Orange, Rolls Royce en Europe**, **Auchan** et pleins d'autres : **Ville de Madrid, EDF**, Gouvernement du Mexique, etc.

Img\_1 : Secteur\_activités\_inetum-div\_sud-ouest

### B. DIGITAL FACTORY



Img\_2 : Approche\_Orientée\_Besoins

Dans un contexte de révolution numérique, la Digital Factory (DF) est un nouveau pôle destiné à relever les défis auxquels font face les entreprises. Elle revisite avec ses clients, les difficultés, les opportunités ou les besoins d'évolution qu'ils rencontrent dans leur métier. En utilisant les connaissances technologiques et les avancées qu'elle contribue à créer, la Digital Factory répond de manière originale à ces sujets. Elle offre :

- Un accompagnement complet au niveau projet (*Force de conseil, Conception, Réalisation, Intégration, TMA*).
- Projet à engagement multi-clients (*Médical, Finance, Energie, Services, Aéronautique, Spatial, ...*).
- Une expertise pérenne (*Réactivité*).

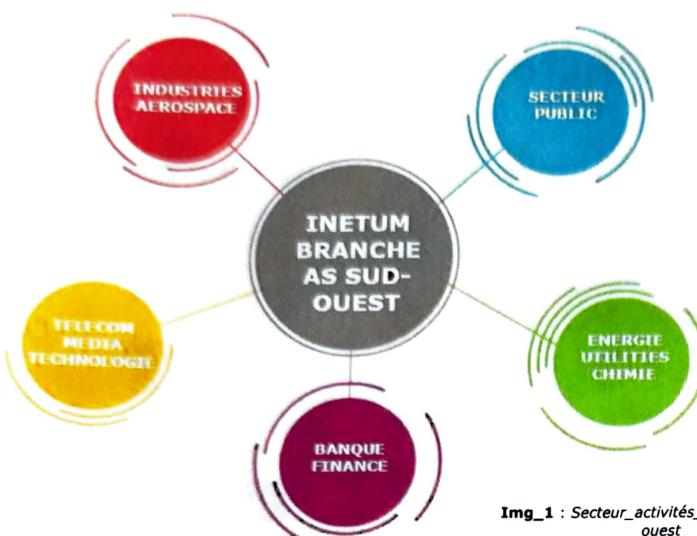
Forte de 20 collaborateurs expérimentés elle propose dans une approche innovante orientée besoins :



- Une offre de réalisation de **projet web** (Site web / Portails (CMS) et Applications métiers en entreprise basés sur les technos modernes).
- La mise en place d'un **Framework Agile** le mieux adapté au contexte des projets et des clients.
- L'intégration d'un processus **UX/UI** complet dans le cadre de ses projets.
- Une offre **IOT** pour intervenir sur toute la chaîne de valeur.
- Mise en place des **Chatbots** sur mesure Offre complète, Cloud native, Backoffice complet, License One Shot.

## 2.3. Organisation des services

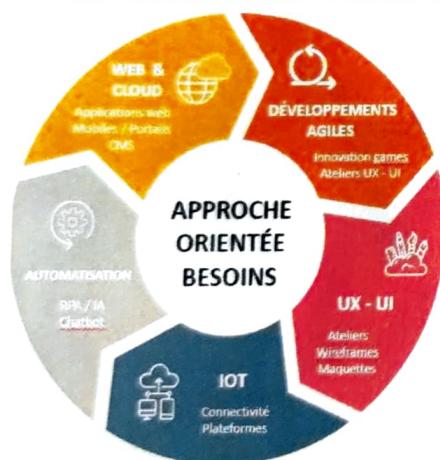
### A. Secteurs d'Activités



Inetum AS Sud-Ouest (AS pour Application Services) est organisée en plusieurs pôles représentant des secteurs d'activités diverses. Ces pôles entretiennent des partenariats avec différentes entreprises comme : Le **Ministère de la Justice français**, **Tigo**, **Orange**, **Rolls Royce en Europe**, **Auchan** et pleins d'autres : **Ville de Madrid**, **EDF**, Gouvernement du Mexique, etc.

Img\_1 : Secteur\_activités\_inetum-div\_sud-ouest

### B. DIGITAL FACTORY



Img\_2 : Approche\_Orientée\_Besoins

Dans un contexte de révolution numérique, la Digital Factory (DF) est un nouveau pôle destiné à relever les défis auxquels font face les entreprises. Elle revisite avec ses clients, les difficultés, les opportunités ou les besoins d'évolution qu'ils rencontrent dans leur métier. En utilisant les connaissances technologiques et les avancées qu'elle contribue à créer, la Digital Factory répond de manière originale à ces sujets. Elle offre :

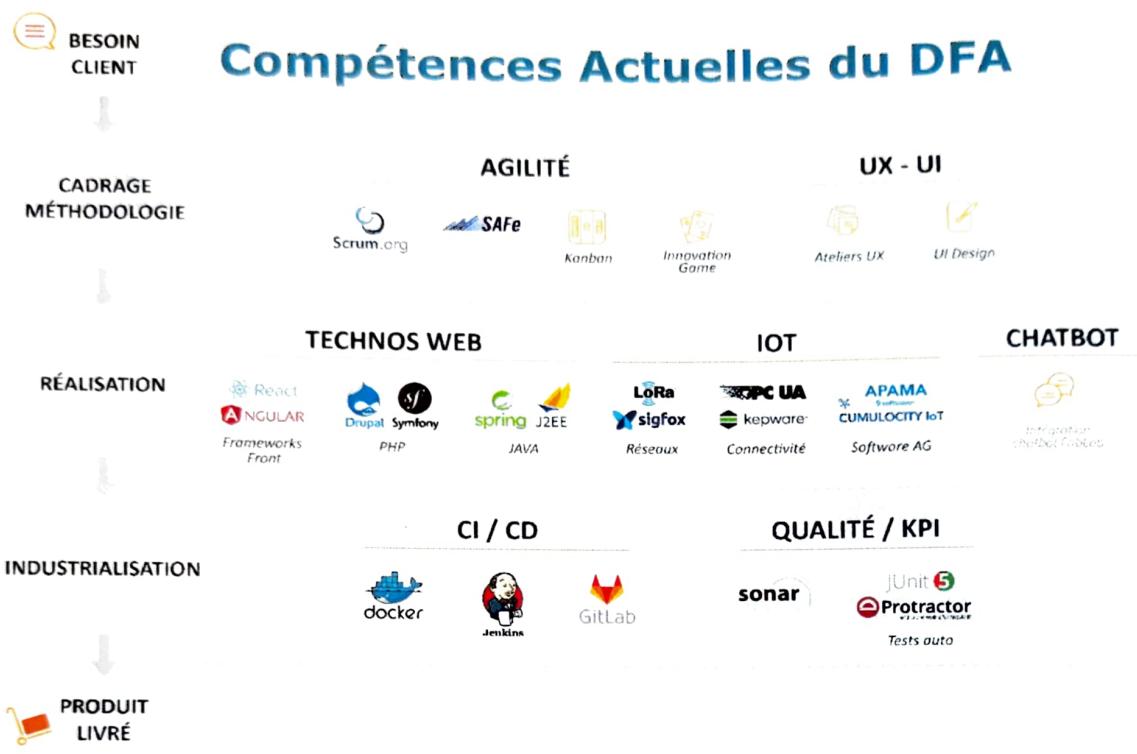
- Un accompagnement complet au niveau projet (*Force de conseil, Conception, Réalisation, Intégration, TMA*).
- Projet à engagement multi-clients (*Médical, Finance, Energie, Services, Aéronautique, Spatial, ...*).
- Une expertise pérenne (*Réactivité*).

Forte de 20 collaborateurs expérimentés elle propose dans **une approche innovante orientée besoins** :



- Une offre de réalisation de **projet web** (Site web / Portails (CMS) et Applications métiers en entreprise basés sur les technos modernes).
- La mise en place d'un **Framework Agile** le mieux adapté au contexte des projets et des clients.
- L'intégration d'un processus **UX/UI** complet dans le cadre de ses projets.
- Une offre **IOT** pour intervenir sur toute la chaîne de valeur.
- Mise en place des **Chatbots** sur mesure Offre complète, Cloud native, Backoffice complet, License One Shot.

La Digital Factory à ce jour, travaille avec des clients nationaux comme, le CNES, Pierre Fabre, TOTAL, etc. mais aussi avec des clients internationaux et couvre les compétences regroupées dans le tableau ci-dessous.



Img\_3 : Compétences\_Actuelles\_Digital-Fac

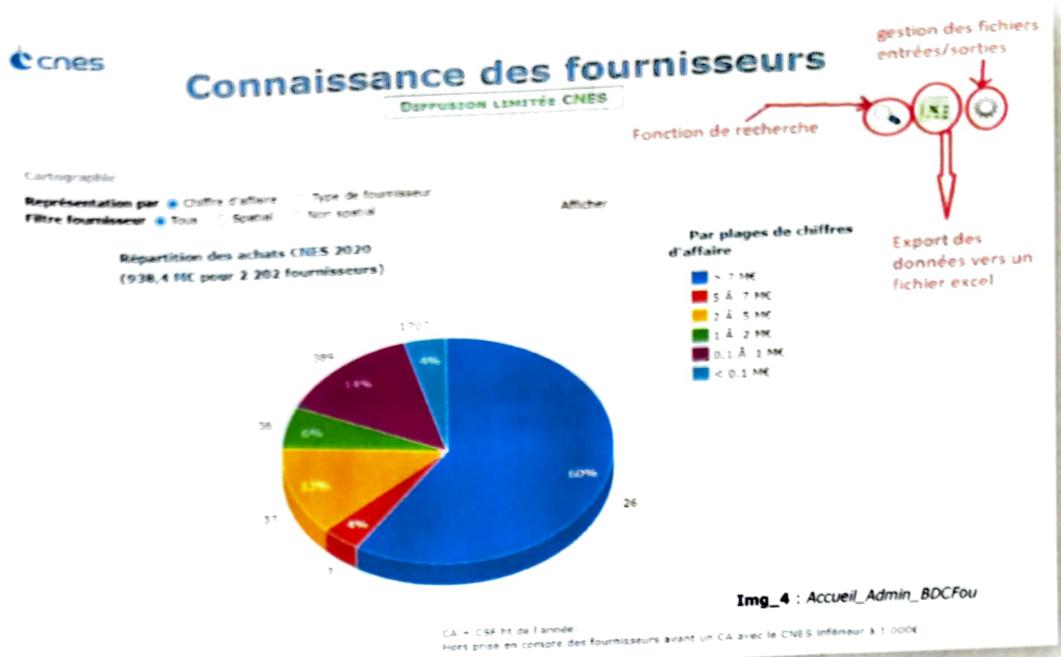
### 3. Présentation du Projet

BDCFou est une « Base de Données de Connaissance des Fournisseurs ». Elle permet aux utilisateurs de consulter les informations concernant les fournisseurs.

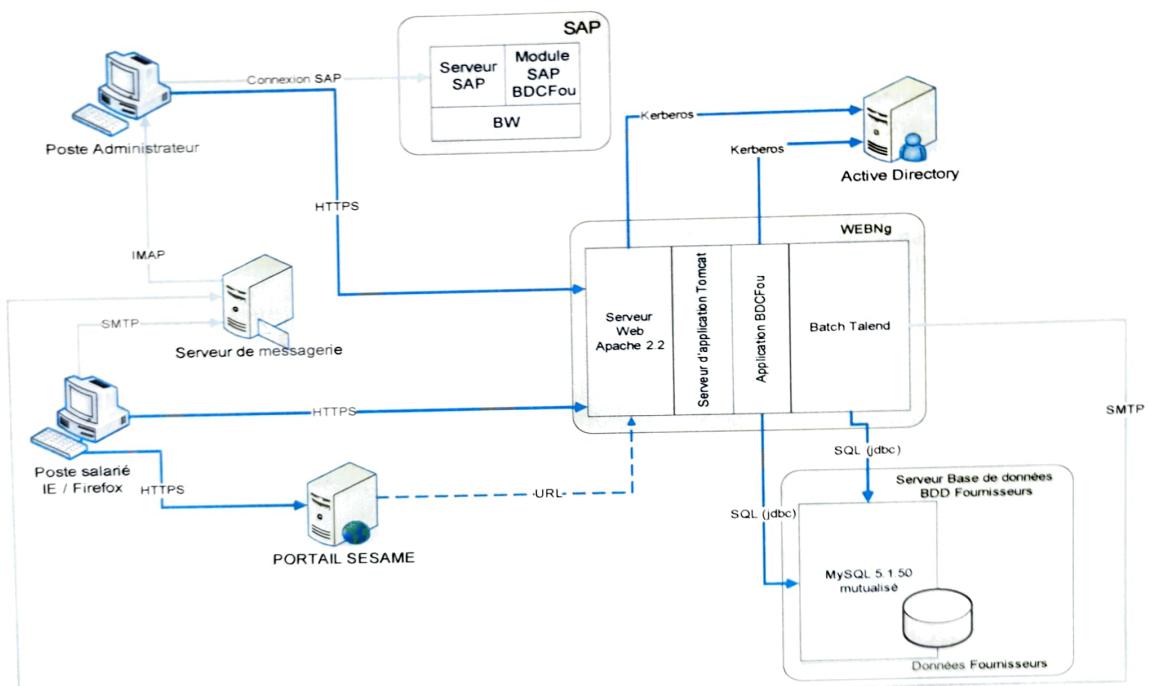
Les fonctionnalités principales mises en œuvre dans le projet BDCFou sont :

- L'alimentation d'une base de données Maria DB spécifique à BDCFou :
  - Via des données issues de SAP/BW (un système d'information).
  - Par des valorisations de données effectuées via l'application Web BDCFou.
- Le positionnement d'une application Web qui permet l'exploitation de la base de données des fournisseurs (Consultation simple, reporting ...).

Les éléments composant BDCFou sont donc une application Web qui proposera l'interface permettant de réaliser les différentes opérations, une base de données qui assurera la persistance et un traitement automatisé qui assurera l'initialisation ou la mise à jour depuis des informations du SI.



### 3.1. Architecture de l'Application BDCFou



BDCFou est hébergée dans une plateforme (**WEBNG**) sécurisée et mutualisée de CNES. Les différents modules applicatifs fonctionnent comme suit :

- Application BDCFou

Il s'agit de la base de données de connaissance des fournisseurs. Elle permet aux utilisateurs de consulter les informations concernant les fournisseurs. Certains utilisateurs, selon leur niveau d'habilitation, peuvent également mettre à jour les informations non issues du système d'information (SAP), soit directement dans l'application, soit indirectement par le biais d'un envoi de mail.

- Base de données BDCFou

Il s'agit du moyen de persistance des informations de la base de données de connaissance des fournisseurs.

- SAP BDCFou

Ce module est responsable de créer un ensemble de données permettant d'enrichir la base de données de connaissance des fournisseurs depuis des informations déjà présentes dans le système d'information (Données de référence, données comptables, ...).

- Batch Talend BDCFou

Le batch Talend récupère les fichiers d'imports déposés sur un filer par l'application BDCFou et stocke leur contenu en BD.

Mon intervention dans ce projet se situe au niveau de l'application Web BDCFou, la plateforme de déploiement WEBNG et de la base de données.

## 3.2. L'Application Web BDCFou

Dans cette partie, nous verrons l'application et ses plus importantes fonctionnalités.

### A. Les différents profils

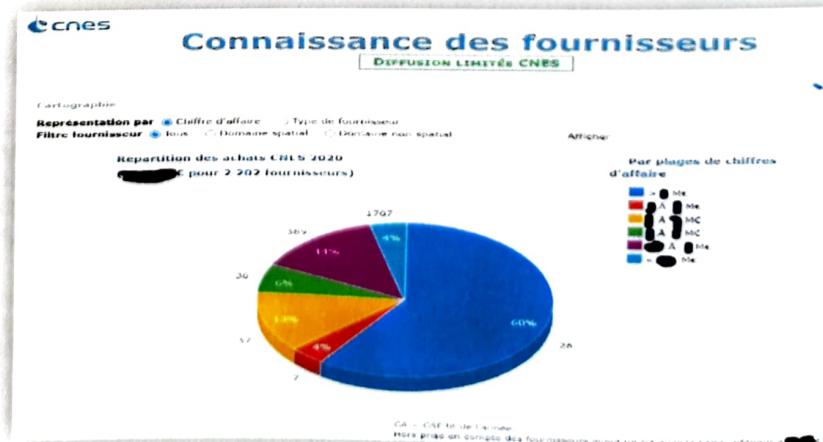
Sur BDCFou en mode développement on a trois profils peuvent intervenir en fonction de leurs autorisations. c'est un contournement GWT pour simuler le profil utilisateur. En production, celui-ci est porté par l'Active Directory et transmis directement à BDCFou via le jeton d'authentification. Ce panneau ne fait donc pas partie de l'application qui tourne en production.



#### Le profil administrateur :

Ce profil a tous les droits. Il peut :

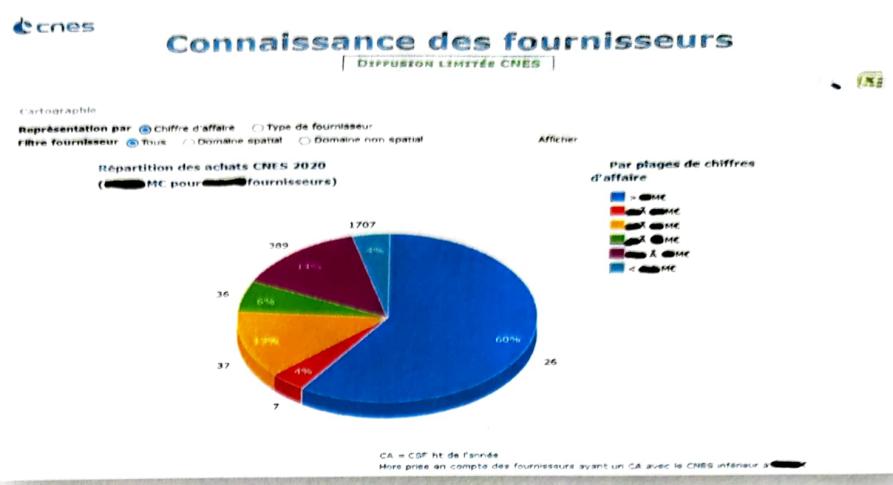
- ✓ Consulter la répartition des achats.
- ✓ Voir les détails d'un fournisseur.
- ✓ Exporter des données de fournisseurs.
- ✓ Importer des données.
- ✓ Modifier les critères de sélection.
- ✓ Lancer un calcul de statistique.
- ✓Modifier les données de fournisseurs.



**Le profil DAJ :**

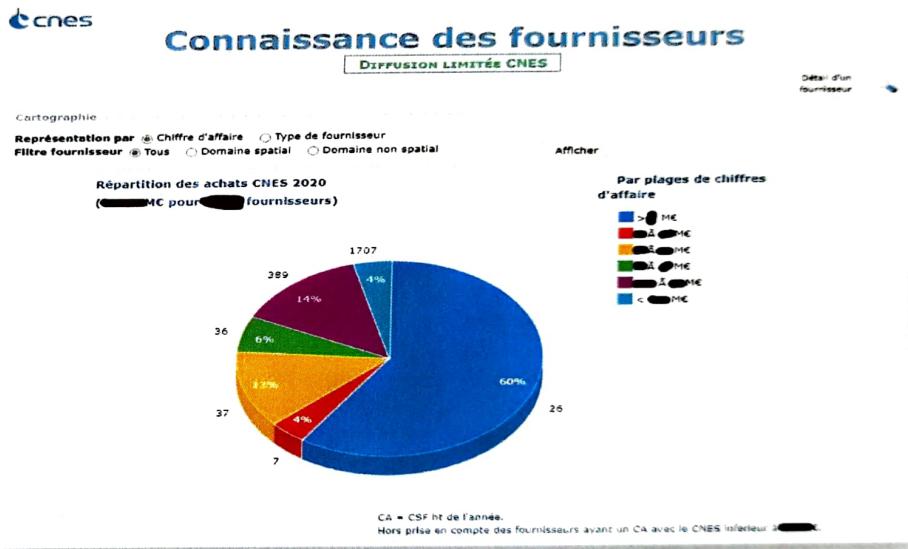
Ce profil quant à lui peut :

- ✓ Consulter la répartition des achats.
- ✓ Voir les détails d'un fournisseur.
- ✓ Exporter des données de fournisseurs.
- ✗ Importer des données.
- ✗ Modifier les critères de sélection.
- ✗ Lancer un calcul de statistique.
- ✓ Modifier les données de fournisseurs.

**Le profil Salarié :**

Ce dernier ne peut que :

- ✓ Consulter la répartition des achats.
- ✓ Voir les détails d'un fournisseur.
- ✗ Exporter des données de fournisseurs.
- ✗ Importer des données.
- ✗ Modifier les critères de sélection.
- ✗ Lancer un calcul de statistique.
- ✗ Modifier les données de fournisseurs.



## B. La fonction de recherche

Cette IHM est l'une des plus importantes de l'application, elle permet la recherche d'un fournisseur à travers différents filtres :

## C. L'IHM d'export des données

La fonctionnalité d'export sert à sélectionner les données des fournisseurs à exporter puis écrit ces dernières dans un fichier au format .xls.

La sélection des données à exporter offre une large gamme d'options permettant d'avoir des données pertinentes sur les fournisseurs.

## D. L'IHM d'édition des fournisseurs

Cette IHM permet de :

- De mettre à jour les données d'un fournisseur.
- De supprimer un fournisseur.
- D'exporter ces données aux formats .doc et .pdf

### 3.3. Contexte et Objectifs du projet

Pour mener à bien ses activités, le CNES doit avoir une bonne connaissance des industriels du secteur spatial et de leurs performances. Pour cela, il est important de capitaliser et de mutualiser dans une base de données certaines informations concernant ces fournisseurs.

Le CNES disposait à l'époque d'une base de connaissance fournisseurs dénommée Mopsos. Celle-ci contenait des éléments sur environ 350 fournisseurs, avec des mises à jour assez lourdes et difficiles. L'usage de la base était limité à la Direction des achats (DAJ) alors que d'autres acteurs du CNES pourraient bénéficier des informations disponibles et même contribuer à leur mise à jour.

Le projet BDCFou a vu le jour en 2013 et le développement de la version 1.0.0 a été mise en ligne courant la même année. Il a ainsi été décidé de faire évoluer la base Mopsos avec pour principaux objectifs de :

- Réunir dans une seule base les informations récurrentes (CA avec le CNES, appartenance à un groupe d'entreprises, etc.) pour l'ensemble des fournisseurs du CNES
- Limiter le nombre des fournisseurs objet d'un « suivi » particulier en ciblant mieux les besoins
- Ajouter les informations jugées utiles et facilement accessibles
- Ouvrir la base à tous les acteurs du CNES pouvant en bénéficier dans le cadre de leurs activités.

La base de données comprend deux types d'informations :

- Pour tous les fournisseurs du CNES (environ 2600 fournisseurs payés par an), la base de données enregistre :
  1. La catégorie d'appartenance de l'entreprise (PME/PME\*, ETI, Laboratoires/Universités, Grands Groupes, Autre)
  2. La classification des fournisseurs (Fournisseurs « clés » ou « non clés »),
  3. Le montant des dépenses annuelles comptabilisées par fournisseur.
- Ce qui permet d'établir des cartographies ou des statistiques annuelles ou pluri annuelles sur le suivi des chiffres d'affaires et des types de fournisseurs.
- Pour 200 à 250 fournisseurs, prioritairement les fournisseurs « clés » (ou fournisseurs « suivis »), la base permet un suivi détaillé comprenant les informations générales de la société (raison sociale, dirigeants etc.) ainsi que des informations économiques et techniques, ce qui permet d'établir des fiches de synthèse fournisseurs (notamment pour la Commission Interne des Marchés)

La base de données est classifiée « Diffusion limitée CNES », elle est ouverte à tous les salariés CNES (3 profils : Administrateur, DAJ, Salarié avec vue limitée aux fournisseurs « suivis »).

L'objectif du projet est de répondre aux nouveaux besoins du client sur des corrections applicatives et sur des demandes de nouvelles fonctionnalités. Ainsi les développements produiront une nouvelle version de l'application, **BDCFou version 2.0.0**, contenant les maintenances correctives et les maintenances évolutives.

### 3.4. Cahier de charges

Le périmètre du projet qui a été définit comportait 8 évolutions avant mon arrivée. Toutefois la demande du client a évolué au cours du stage, en ajoutant des nouvelles évolutions, voire en en supprimant certaines déjà en cours de développement. Voici la version finale du cahier des charges initial suivi des modifications du cahier des charges au cours du développement du projet.

NB : Les demandes client sont définies sous le sigle « **DM-suivi du numéro** » de la demande.

1. **DM-00011** : Ajouter un bouton radio "Tous", sélectionné par défaut, permettant de réinitialiser le filtre "Fournisseur suivi" dans la recherche.
2. **DM-00021** : Ajouter un choix "Toutes" dans la liste des régions du monde, sélectionné par défaut, permettant de réinitialiser ce filtre dans la recherche.
3. **DM-00023** : Sur la page d'accueil (avec le diagramme circulaire), permettre l'affichage d'une popup au survol des items (type de fournisseur, spatial et fournisseur suivi dans la page de recherche de fournisseur) contenant une définition de l'item.
4. **DM-00028** : Ajouter 4 colonnes supplémentaires (Chiffre d'affaires du titulaire, Effectifs du Titulaire, Indicateur clé, Indicateur innovant) dans la sélection des données à exporter.
5. **DM-00029** : Import des 3 nouvelles colonnes suivantes contenues dans le fichier Altares.
  - Colonne K : Nombre de salarié
  - Colonne L : Montant du chiffre d'Affaires (FL)
  - Colonne Q : FLAG Innovant (*DM annulée par le CNES*)

Nouvelles demandes du client :

6. **Nouvelle DM-00027.1** : Ajout d'un indicateur clé « *Fournisseur clé* » qui est un nouvel attribut à ajouter dans BDCFou avec un radio bouton oui / non (ajout dans l'IHM au même niveau que « Spatial » + BDD). Sa valeur sera entrée manuellement dans l'application.
7. **Nouvelle DM-00027.2** : Ajouter un nouvel attribut « *Start-up* » avec un radio bouton oui / non (comme pour le domaine spatial). Sa valeur sera entrée manuellement. Il sera nécessaire de pouvoir exporter cette valeur : à ajouter donc dans la pop-up de sélection des données à exporter.
8. **Nouvelle DM-00029** : Changer le nom « *Spatial* » en « *Domaine spatial* » dans l'IHM.
9. **Nouvelle DM-00030** : Dans le cadre du projet BDCFou, la sensibilité des données hébergées par l'application BDCFou nécessite la consultation de celle-ci depuis un poste sécurisé. Cette disposition entraîne des contraintes fortes dans le cadre du développement. Pour pallier ce problème, il est nécessaire de mettre en place une procédure d'anonymisation des données de l'application.

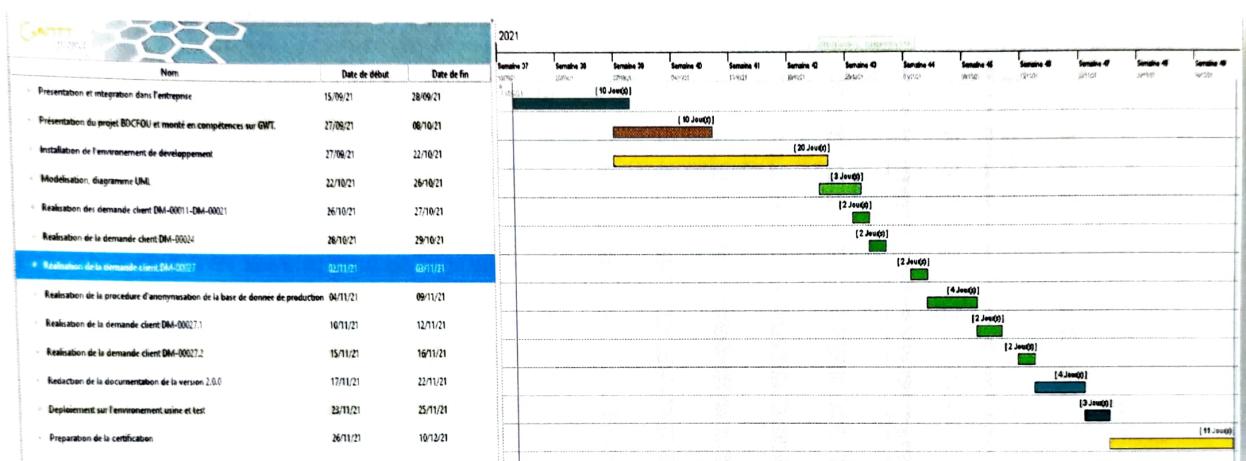
### 3.5. La Gestion du projet

Pour répondre à ces objectifs et atteindre le but fixé, j'ai planifié la gestion du projet BDCFou, et consigné l'organisation et la répartition des phases de développement et des tâches entreprises dans le diagramme de Gant ci-après.

#### A. Programmation du déroulé

Livraison	Date prévue
<b>Lancement</b>	T0
<b>Livraison des CR + journal des essais</b>	T0 + 05j
<b>Recette usine + BT et CRE de RU</b>	T0 + 10j
<b>Livraison des binaires</b>	T0 + 30j
<b>Livraison de la documentation</b>	T0 + 35j

## B. Repartition des tâches



### 3.6. Productions attendues

Les productions attendues étaient la livraison de la version V2.0.0 de l'application mais comme les demandes du client évoluaient, l'objectif de la fin du stage est fixé à la livraison d'une partie des évolutions développées avec la documentation, dû au fait d'une redéfinition du périmètre fonctionnel. Ces DMs seront traitées dans une livraison ultérieure.

Les livrables contenues dans le tableau ci-dessous seront attendus à la fin de mon stage.

Type	Sortie	Jalon de livraison	Critère de validation	Moyen de livraison
Dossier de conception détaillée	DEV-SP-BDCF0U-0223-Dossier de conception.docx	Livraison Dossier de conception	Recette usine validée	Galaxi
Cahiers de recette	RCT-BDCF0U-Cahier_Recette_BDCF0U.xlsxm	Livraison Cahier de Recette	Recette usine validée	Galaxi
Synthèse résultats des essais	Nom à définir	Livraison Cahier de Recette	Recette usine validée	Galaxi
Manuel d'installation	MII-BDCF0U-Manuel_Installation	Livraison Documentation	Recette usine validée	Galaxi
Manuel d'exploitation	MEX-BDCF0U-Manuel_d'exploitation	Livraison Documentation	Recette usine validée	Galaxi
Manuel utilisateur	MCO-MA-BDCF0U-0315-Manuel Utilisateur	Livraison Documentation	Recette usine validée	Galaxi
Binaires	Nom à définir	Livraison Binaires	Recette usine validée	File Sender

## 3.7. Environnement et Contraintes techniques

### A. Environnement Technique

#### 1. Eclipse

 L'IDE (environnement de développement intégré) Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent. Il est d'abord conçu pour le langage Java mais ses nombreux plugin (greffons) en font un environnement de développement pour de nombreux autres langages de programmation (C/C++, Python, PHP, Ruby, ...).

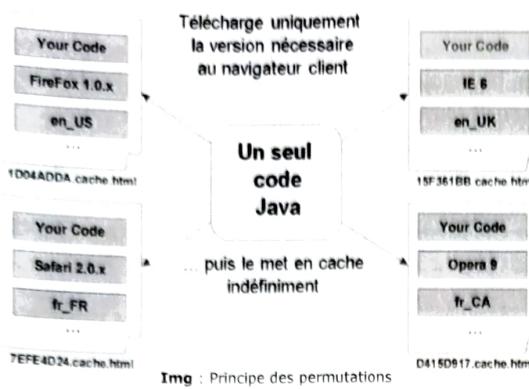
#### 2. Java

 Jakarta EE (anciennement Java 2 Platform, Enterprise Edition, ou J2EE, puis Java Platform, Enterprise Edition ou Java EE), est une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise. La plate-forme étend Java Platform, Standard Edition (Java SE) en fournissant une API de mapping objet-relationnel, des architectures distribuées et multi-tiers, et des services web. La plate-forme se fonde principalement sur des composants modulaires exécutés sur un serveur d'applications. Sur le projet BDCFou, la version du JDK utilisé est JDK/JRE 1.7

#### 3. GWT

 GWT (Google Web Toolkit) est un Framework open source de développement d'applications web mettant en œuvre AJAX développé par Google. A ce jour, une bonne partie des services de Google sont codés et développés avec GWT (Google AdWords, Google Wallet, etc..).

Le code de l'application est entièrement écrit en Java de façon classique, notamment la partie cliente qui devra s'exécuter dans un navigateur. Conçu à l'origine pour résoudre les problèmes de compatibilité entre navigateurs, ce code Java n'est pas compilé en bytecode mais en JavaScript **minifié** ce qui permet son exécution dans un navigateur.



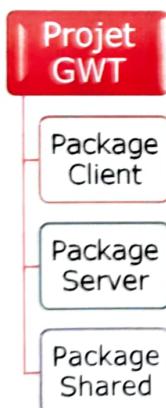
L'avantage du code JavaScript produit est qu'il est capable de s'exécuter sur les principaux navigateurs sans adaptation particulière du code source Java puisque le compilateur créé un fichier JavaScript optimisé pour chacun de ces navigateurs (voir Img : principe des permutations).

La partie graphique d'une application GWT est composée d'une petite partie en HTML, de CSS et surtout de classes Java dans lesquelles des composants sont utilisés avec des gestionnaires d'événements pour définir l'interface de l'application et les réponses aux actions des utilisateurs.

GWT utilise un patron de conception **MVP** (Model View Presenter) pour développement applicatif, ce dernier est dérivé du **MVC** (Model View Controller) classique sous lesquelles sont développées la plupart des applications JEE. Dans le pattern MVP, le modèle (partie des traitements ou services) est complètement séparé de la présentation ou de la vue (View). Dans ce pattern MVP, le présenteur joue le rôle de contrôleur au sens strict du terme afin d'isoler toute communication directe entre la vue et le modèle. Dans ce modèle MVP chaque partie a un rôle particulier.

- Le **Modèle** assure la logique métier. C'est lui qui fait les traitements. Il ne contient aucun composant graphique.
- La **Vue** est l'interface par laquelle l'utilisateur agit sur l'application. Aucun traitement ne devrait avoir lieu dans la vue. Elle sert uniquement d'affichage et contient donc des composants graphiques.
- Le **Présenteur** assure la communication entre la vue et le modèle. Il connaît toutes les méthodes de traitements et de mises à jour de la vue.

Un projet GWT est structuré en trois packages principaux :



4. **Le package client** qui contient les vues construites à base de Widgets ou composants fournis par GWT, donc pas de code métier dans ce package qui a pour rôle de fournir à l'utilisateur des moyens d'interaction avec l'application à travers des composants (formulaire, boutons etc..).
5. **Le package server** (ou serveur) qui contient la logique métier les échanges avec les autres applications et les services transverses. Donc toute modification du code dans le package server nécessite une recompilation du module. Les **services RPC** (Remote Procedure Call) seront implémentés dans cette couche. Un **service RPC** est un service qui fonctionne en mode asynchrone (utilise le mode de communication en différé) comparativement à un mode de communication synchrone où les échanges d'informations sont directs.
6. **Le package shared** (package partagé entre le client et le serveur) qui contient des modules ou classes qui sont à la fois nécessaires côté client et côté serveur. Il contient toutes les interfaces asynchrones afin d'intercepter tous les appels asynchrones sollicités par l'utilisateur. À chaque appel asynchrone correspond un service RPC. Mais on peut aussi mettre les interfaces asynchrones dans le module client, ce qui est le cas pour l'application BDCFou.

Le projet BDCFou est développé entièrement sous GWT **V2.5** et sa structure s'inscrit dans cette logique. Nous verrons plus en détails dans la partie [Conception](#) plus loin dans ce rapport.

#### 4. Maven

**Maven**™ Maven est un outil de gestion et de compréhension de projet, développé par la fondation Apache, qui est basé sur le concept d'un modèle d'objet de projet (POM) c'est-à-dire un ensemble de standards, un cycle de vie, et un système de gestion des dépendances. Il

embarque aussi la logique nécessaire à l'exécution d'actions pour des phases bien définies de ce cycle de vie, par le biais de plugins. Pour utiliser Maven, il est nécessaire de décrire l'application selon un modèle objet de projet clair. Maven peut alors lui appliquer la logique transverse d'un ensemble de plugins (partagés ou spécifiques).

Dans le cadre du projet BDCFou, l'utilisation de **Maven version 3.5.0** permet la gestion des librairies tierces, des Frameworks dont l'application a besoin mais aussi et surtout l'appel d'un module dans un autre comme une dépendance (ex : BDCFou-SI dépendance de BDCFou-App) Il permet aussi la génération de la structure de chaque module du projet. Chacun de ces modules a une structure différente en fonction des Frameworks mis en œuvre.

## 5. Hibernate



Hibernate est un Framework open source de type ORM (Object Relational Mapping) développé par Red Hat. Il permet de faciliter le développement de la couche de persistance d'une application et donc de représenter une base de données en objets Java et vice versa.

Dans notre projet la version utilisée est la 3.5.2. Hibernate facilite la persistance et la recherche de données dans une base de données en réalisant lui-même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant.

## 6. Tomcat



Tomcat est un projet issu de Jakarta, le groupe de projets open-source du groupe Apache dédié à Java. Apache produit un grand nombre d'autres outils opensource, et notamment le fameux serveur HTTP Apache qui fut le serveur le plus utilisé au monde jusqu'à qu'il se fasse dépasser par Nginx récemment. Comme tous les projets Jakarta, Tomcat est développé par une équipe de bénévoles, et est distribué sous la licence Apache, qui permet son utilisation dans n'importe quel cadre (personnel ou professionnel), y compris commercial.

Tomcat est un conteneur libre, intégralement écrit en Java qui implémente les spécifications des Servlets et JSP de la plate-forme JEE. Il inclut des outils pour la configuration et la gestion et comporte également un serveur HTTP. Dans le cadre du projet BDCFou, Tomcat est le serveur sur lequel l'application BDCFou-App est déployée et la version utilisée est la **7.0.109**.

## 7. Subversion



Pour la gestion de configuration on utilise Subversion (en abrégé svn) qui est un logiciel de gestion de versions, distribué sous licence Apache3. Il a été conçu pour remplacer CVS. Subversion fonctionne sur le mode client-serveur, Il s'appuie sur le concept du principe du dépôt centralisé et unique et sa tâche est de gérer l'évolution du contenu d'une arborescence. Dans le cadre de notre environnement le client que nous utilisons est sous forme de plug-in TortoiseSVN et Eclipse Subclipse.

## 8. MySQL



MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde3, autant par le grand public (applications web principalement) que par des professionnels. Il est en concurrence avec d'autre SGBDR tels que Oracle, PostgreSQL, MariaDB, Microsoft SQL Server etc.

J'ai travaillé en développement sur MySQL Workbench 8.0 CE (anciennement MySQL administrator) qui est un logiciel de gestion et d'administration de bases de données MySQL créé en 2004. Via une interface graphique intuitive, il permet, entre autres, de créer, modifier ou supprimer des tables, des comptes utilisateurs, et d'effectuer toutes les opérations inhérentes à la gestion d'une base de données. Pour ce faire, il doit être connecté à un serveur MySQL.

## 9. PuTTY

 PuTTY est un **émulateur de terminal** (console virtuelle ou terminale virtuel) c'est-à-dire qu'il émule le fonctionnement d'un terminal informatique. Il est doublé d'un client pour les protocoles **SSH**, **Telnet**, **rlogin**, et **TCP** brut. Il permet également d'établir des connexions directes par liaison série **RS-232**.

 Nous utilisons PuTTY pour émuler un terminal Unix afin de créer un script Bash SSH à destination des DBA (Data Base Administrators) qui permettra de faire appel à des scripts SQL pour anonymiser les données de la DB en prod lors d'un export d'un dump de cette dernière.

## 10. WEB NG

 Web NG est une plateforme d'hébergement du CNES. C'est un socle constitué d'une machine d'administration des sites qui seront déployés et une machine d'Hébergement qui lui n'est pas connecté à internet. Elle offre à la fois

## 11. WhiteStarUML

 WhiteStarUML est un logiciel de modélisation UML, qui a été « cédé comme open source » par son éditeur. Il offre la plupart des fonctionnalités attendues d'un outil UML comme une large sélection de diagrammes pris en charge, des modèles de conception prêts à l'emploi, la vérification de modèle, la génération de code et fournit toujours de multiples moyens d'extensibilité via des plugins, des scripts ou des profils, ce qui le rend même utilisable d'une manière qui n'était pas initialement prévisualisé par ses créateurs. Dans le cadre du projet BDCFou, nous l'utilisons pour réaliser la modélisation de la base de données et certains diagrammes de classes.

## 12. Microsoft Teams

Microsoft Teams est une application de communication collaborative, officiellement lancée par son propriétaire Microsoft en novembre 2016. Toute au long de ma période de stage, cette application nous servait de plateforme de communication pour les échanges (daily, partage de documents etc...).

## B. Contraintes du Projet

L'application est conçue sur une version GWT 2.3 plus ancienne que la version 2.5 juste avant l'abandon de cette technologie en 2016. Donc une vieille techno Obsolète et pas maintenu depuis son abandon et il existe très peu de documentation sur internet il a fallu qu'Inetum commande un bouquin qui traite le sujet pour que je monte en compétence.

L'essentiel des contraintes que j'ai rencontré sont techniques et étaient dû à un problème d'installation d'environnement de développement sachant que l'appli tournait déjà en production :

- L'application est développée sous Java 1.7.
- Impossibilité de trouver la version correspondante du plugin GWT d'Eclipse mise à part sa version 3.0.
- Gros problème pour trouver la version correspondante du serveur Tomcat pendant un temps. (Finalement après un temps version 7.0).
- Pareil pour la version maven car le projet BDCFou est un projet maven. (Finalement c'était la version 3.5).
- Impossibilité d'utiliser le « mode développeur » de GWT qui permettrait la compilation à la volée. Je développai donc en aveugle, compilai et déployai pour ensuite voir si les modifications que j'ai apportées sont bonnes ou pas sinon je reprenais le même processus.
- Des difficultés pour debugger le code.
- Absence d'un fonctionnel interlocuteur côté CNES qui maîtrise le périmètre de l'application.

- Documentation de conception, d'installation et de d'utilisation inexistante.

## 4. Analyse des Besoins

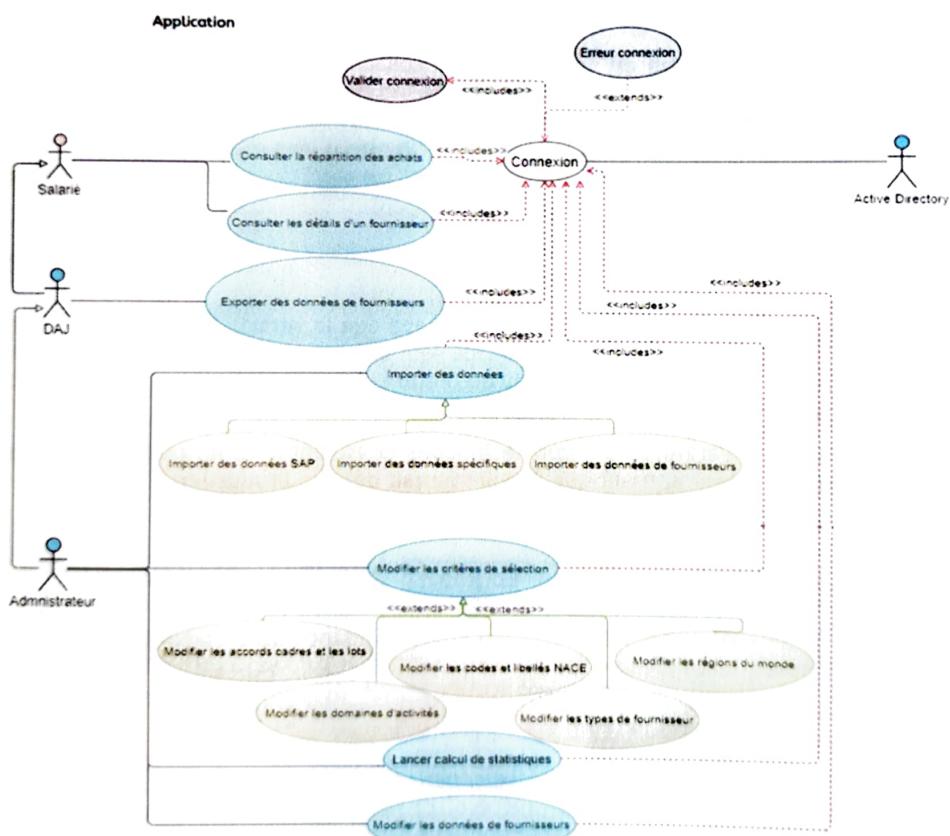
Après découverte de BDCFou et son cœur de métier et vu qu'il n'y avait pas de documents de conception, j'ai commencé par procéder en effectuant un reverse-engineering de l'application Web ainsi que de la base de données en m'a aidant du schéma architectural pour réaliser :

- La modélisation des cas d'utilisation de l'application.
- Le schéma représentant le démarrage de l'application.
- La modélisation de la base de données.

### 4.1. Cas d'utilisation de BDCFou

Le diagramme ci-dessous représente les cas d'utilisation de l'application pour :

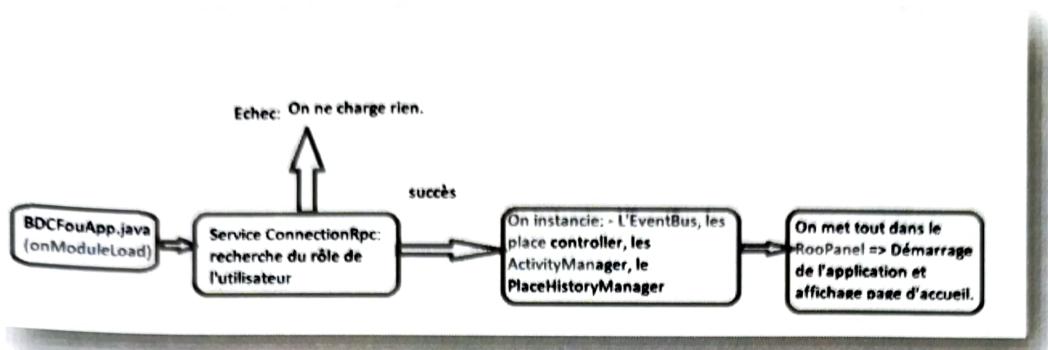
- Les différents profils utilisateurs (Administrateur, DAJ, Salarié avec vue limitée aux fournisseurs « suivis »).
- Les systèmes avec lesquels ils interagissent (Le système d'information SAP et le serveur active directory pour les identifications).



Ce travail m'a permis de comprendre et de voir toutes les tâches associées à un profil utilisateur et les interactions avec les systèmes.

## 4.2. Schéma de démarrage de l'application

Ce schéma représente les appels de classes à partir du point d'entrée `BDCFou.onModuleLoad()`.



## 4.3. La BDD BDCFou

C'est dans cette logique que j'ai entamé la compréhension de la base de données MySQL de BDCFou à partir d'un dump de la BDD de la production. J'ai entrepris de même la réalisation du :

1. Dictionnaire de donnée (Voir section Annexe : [Figure BDD-01 : Partie Dictionnaire données BDCFou](#)).
2. Modèle conceptuel des données (MCD, voir section Annexe [Figure BDD-02 : MCD BDCFou](#)).
3. Modèle logique des données (MLD, voir section Annexe [Figure BDD-03 : MLD BDCFou](#)).

Cette tâche m'a permis de localiser aisément les tables sur lesquelles j'interviendrais pour réaliser les évolutions demandées par le client.

# 5. Conception

## 5.1. Les différents modules composant l'application

BDCFou est un projet MAVEN se composant de quatre modules :

- Project Explorer × Type Hierarchy
  - **BDCFou-App** : c'est l'application GWT : elle fait appel au \*.jar de BDCFou-SI.
  - **BDCFou-SI** : qui gère les interactions avec la base de données (Dao + services CRUD sur la base de données).
  - **BDCFou-BDD** : qui permet de gérer les scripts sql de la base de données et de générer un fichier Zip les contenant.
  - **BDCFou-WebNG** : Ce module a la même structure que BDCFou-BDD avec les différentes ressources.

- BDCFou-App

=> L'Entry-Point est la classe de démarrage d'une application GWT et contient obligatoirement la méthode **onModuleLoad()** chargée d'initialiser l'application. Classe BdcfouApp.java dans le package client.

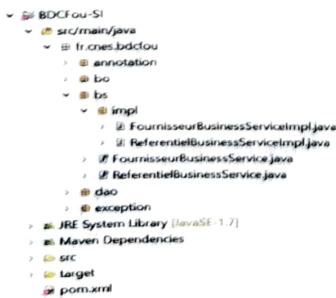
- (**Architecture GWT**) - **Sous-package app.client** (rappel: partie interface graphique de l'application) :



- **Activity** : Les activités gèrent ce que l'utilisateur fait. Une Activity (activité) représente ce que fait l'utilisateur dans une Place. C'est la classe chargée de traiter les demandes sollicitées par la vue, par exemple le clic sur un bouton. Ces demandes lui sont transmises grâce au présentateur. Chaque activité est liée à une Place c'est-à-dire une page. (Le Présenteur est une interface contenant des méthodes qui permettent à la vue de communiquer avec l'**Activity** (service traiteur)).
- **Factory** : La ClientFactory, intégrée à chaque Activity, sera chargée de fournir différentes implémentations de la vue. Elle permet de rendre les vues réutilisables.
- **Mapper** : ce package contient deux classes qui associent les Places (emplacement enregistrable) à leur Activity.
- **Place** : une Place représente une page de l'application. Elle correspond à l'endroit où on se trouve dans l'application. Donc chaque page possède sa propre Place. Une Place doit toujours être associée à une PlaceTokenizer qui permet de la sérialiser, c'est-à-dire transformer une Place en URL.
- **Resources** : contient les fichiers Java pour le CSS

- **Rpc (Remote Procedure Call)**: contient les interfaces des services RPC et leur interfaces Asynchrones correspondantes.
- **Ui** : contient les vues
- **Vo** : contient tous les mappers.
- **(Architecture GWT) Sous-package app.shared** (rappel : classes partagées entre serveur et client) :
  - **Dto** : toutes les DTO utiles
  - **Enums**: les classes enum
  - **Exception** : ServiceCallException.java : lancée quand il y a un problème avec les services RPC.
  - **Util** : StringUtil.java pour les manipulations des String.
  - **Verifier** : Toutes les classes pour faire des contrôles sur l'interface graphique. ( exemple, vérification des libellés graphiques, vérification des champs).
- **(Architecture GWT) Sous-package app.server** (rappel : classes exécutées côté serveur) :
  - **Export** : Contient les objets utilisés pour l'export + une classe d'utilitaire spécifique à l'export.
  - **Rpc** : Contient les classes d'implémentation des services RPC (des interfaces contenues dans le sous-package RPC du package client)
  - **Security** : Contient la classe pour la gestion et la récupération des données de l'utilisateur.
  - **Servlet** : Contient les servlets utilisées pour l'export de données.
  - **Utils** : Enum, constantes et classes d'outils.
- **Sous-package as (Application Service)** :
  - **Aspect** : contient la classe de type AspectJ pour la réalisation de logs.
  - **Export** : Classes pour la construction de l'export Excel provenant de la "page" Cartographie : les lignes, les colonnes, les Headers du fichier Excel par exemple.
  - **Impl** : contient les implémentations des services applicatifs pour les différents "aspects" de l'application (Cartographie, Fournisseur..etc).
  - **Utils**: constantes.
- **Sous-package bo (Business Object)** : différents objets métier + des classes d'énumérations.
- **Sous-package bs (Business Service)** : Interfaces + classes d'implémentation de services de gestion des statistiques et de gestion des paramètres applicatifs.
- **Sous-package dao** : dao pour les paramètres applicatifs et les statistiques.
- **BDCFou-SI**.

Ce module permet notamment les opérations et les liens vers la base de données.



- **Sous-package annotation** : BDCFouTransactional.java, interface qui définit une annotation à apposer sur TOUTES les classes d'implémentation des services pour que les transactions soient traitées de la même manière.

- **Sous-package bo (Business Object )** : Ensembles des différents objets métier.
- **Sous-package bs (Business Service)** : Service de gestion des fournisseurs + service de gestion des référentiels, qui vont faire appel aux implémentations des classes DAO pour l'accès à la base de données.
- **Sous-package dao** : Tous les dao ! La couche DAO est la couche qui implémente l'accès à la base de données + leur implémentation dans les sous-packages impl.rw (Read-Write) ou impl.ro (Read Only).
- **Sous-package exception** : BDCFouException : classe générale pour les exceptions.

- BDCFou-BDD.



Contient les différentes ressources pour la BDD pour chaque environnement (dev, prod, qualif, recette) dans src/main. Quand on fait un mvn install, crée un zip contenant les différents fichiers SQL de la ressource considérée. Ce zip généré se trouve dans le dossier target, créé par la compilation.

Dans le dossier assembly, le fichier sert à paramétriser la création du Zip.

- BDCFou-WebNG.

Ce module a la même structure que BDCFou-BDD avec les différentes ressources, la génération d'un zip + le dossier assembly.

## 5.2. La fonction principale de l'application « Recherche d'un fournisseur ».

Avant de démarrer le développement des maintenances, il m'a fallu d'abord comprendre le processus de récupération de la liste de fournisseurs avant de pouvoir apporter des modifications. Comment marche la recherche ? Combien de classes sont touchées ? etc...

Pour avoir des réponses, j'ai retracé à partir du code les appels des classes/méthodes les uns par rapport aux autres puis j'ai élaboré un diagramme de séquence qui retrace toutes les étapes. A partir de ce diagramme j'ai élaboré le diagramme de classe qui en résulte (Annexe partie conception Figure-01 Conception et Figure-02 Conception et Figure-0 page 52).

Cette étape a été pour moi déterminante dans la compréhension des échanges entre l'application et la base de données. C'est donc avec plus d'éclairage que j'ai entrepris la suite de mes travaux.

# 6. Implémentation

## 6.1. Application des maintenances correctives et évolutives

Après l'analyse des besoins et la conception j'ai entamé la réalisation des maintenances correctives et évolutives. Cette partie présentera les travaux réalisés.

Je vous présenterais une partie des évolutions détaillées et le reste des maintenances que j'ai effectué, je les mettrai dans la partie annexe les captures d'écrans des versions 1 et 2.

## A. Demande de Modification 00011

Cette demande concerne l'ajout d'un bouton radio "Tous", sélectionné par défaut, permettant de réinitialiser le filtre "Fournisseur suivi" dans la recherche.

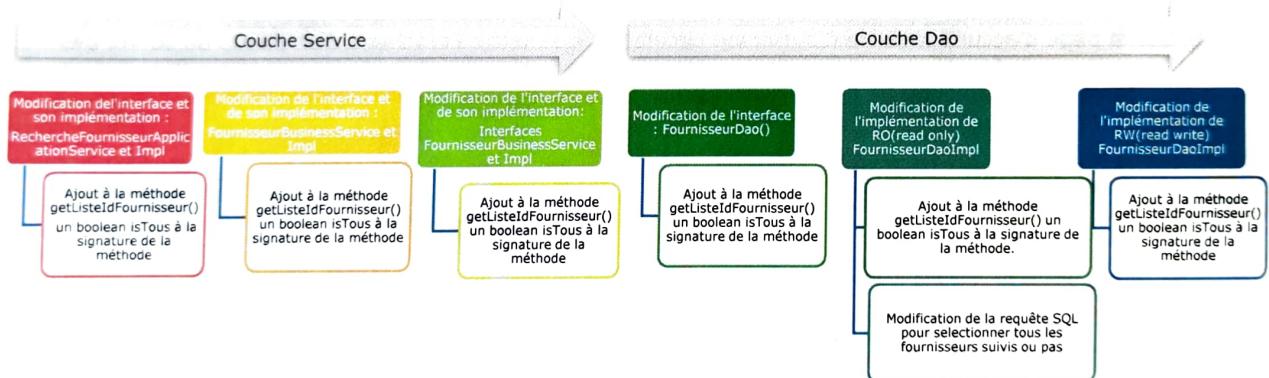
The screenshot shows the 'Connaissance des fournisseurs' application interface. At the top, there is a search bar with fields for 'Nom' (Name) and 'Domaine d'activités' (Activity domain). A yellow box highlights the 'Domaine d'activités' field. Below the search bar is a table titled 'Résultat de la recherche' (Search results) with columns for 'Nom', 'Région du monde', 'Pays', 'Dernière mise à jour', and 'Domaine d'activités'. One row in the table has a yellow background, matching the highlighted field above. At the bottom right of the search bar, there is a 'Rechercher' button.

Solution : Ajout d'un bouton radio "Tous" sélectionné par défaut et modification des fonctions de recherche pour permettre la suppression du filtre dans ce cas.

Pour ajouter cette fonctionnalité j'ai apporté des modifications à plusieurs tables appartenant à plusieurs couches et modules. Ces boutons radios sont générés dynamiquement et non pas à partir des classes XML.

Vous trouverez une partie du code ainsi que l'IHM portant la modification dans la partie 10.1. Image BDCFou V2.0.0 de l'Annexe ([Figure DM-11 01](#) et [Figure DM-11 02](#) à la page 37).

Le schéma ci-dessous représente les travaux de développement de la DM-0011 :



## B. Demande de Modification 00021

The screenshot shows the 'Connaissance des fournisseurs' application interface. At the top, there is a search bar with fields for 'Nom' (Name) and 'Domaine d'activités' (Activity domain). A yellow box highlights the 'Domaine d'activités' field. Below the search bar is a table titled 'Résultat de la recherche' (Search results) with columns for 'Nom', 'Région du monde', 'Pays', 'Dernière mise à jour', and 'Domaine d'activités'. One row in the table has a yellow background, matching the highlighted field above. At the bottom right of the search bar, there is a 'Rechercher' button.

Cette demande concerne le besoin de faire une sélection des fournisseurs de toutes les régions lors de la recherche. Il faut donc ajouter un choix "Toutes" dans la liste des régions du monde, sélectionnée par défaut, permettant de réinitialiser ce filtre dans la recherche. Pour ajouter cette fonctionnalité, j'ai ajouté une option "Toutes" dans la liste des régions du monde sélectionnée par défaut et modifié les fonctions de recherche pour permettre la suppression du filtre dans ce cas.

J'ai donc commencé par créer une classe pour ranger mes constantes dans le package des objets métiers, pour ne pas avoir des choses en dur dans le code. Je l'appellerai BDCFouConstantes.java. Ensuite dans la couche implémentation de service RPC du package server lors de la mise à jour du modèle (c'est-à-dire la DTO qui fournit à la vue les données), j'ai ajouté une région fictive « Toutes » qui supprimera le filtre de sélection des régions. Enfin j'ai ajouté à la requête de la DAO une condition qui fera les « **select** » sur toutes les régions.

Vous trouverez une partie du code ainsi que l'IHM portant la modification dans la partie [10.1 Image BDCFou V2.0.0](#) de l'Annexe ([Figure\\_DM-11\\_02](#) page 37, [Figure-01\\_DM-00021](#) et [Figure-02\\_DM-00021](#) page 39).

Le schéma ci-dessous représente les travaux de développement de la DM-0021 :



## C. Demande de Modification 00023

Sur la page d'accueil (avec le diagramme circulaire), permettre l'affichage d'un popup au survol des items (type de fournisseur, spatial et fournisseur suivi dans la page de recherche de fournisseur) contenant une définition de l'item (voir capture juste au-dessous).

au survol de Type de fournisseur, affiche l'info bulle

**PME :**  
La catégorie des petites et moyennes entreprises (PME, Incluant les TPE) est constituée des entreprises qui :

- emploient moins de 250 personnes ;
- ont un chiffre d'affaires annuel n'excédant pas 50 millions d'euros ou un total de bilan n'excédant pas 43 millions d'euros ;

et dont le capital n'est pas détenu à plus de 50 % par une Entreprise de Taille .... (ETI) ou une grande entreprise (GE).

**ETI :**  
La catégorie PME\* (Petite et moyenne entreprise « étendue », catégorie propre au CNES) est constituée des entreprises qui :

- emploient moins de 250 personnes ;
- ont un chiffre d'affaires annuel n'excédant pas 50 millions d'euros ou un total de bilan n'excédant pas 43 millions d'euros ;

et dont le capital est détenu à plus de 50 % par une ETI ou un GGE.

**NB : NB** : PME (droit français) = PME + PME dans BDCFou (droit EU)  
**ETI** (Entreprises de Taille Intermédiaire):  
La catégorie des ETI est constituée des entreprises qui n'appartiennent pas aux catégories des PME ou des PME\* et qui :

- emploient moins de 5 000 personnes ;
- ont un chiffre d'affaires annuel n'excédant pas 1,5 milliard € ou un total de bilan n'excédant pas 2 milliards €.

**GE (Grandes Entreprises) :** entreprises dont l'un des critères (effectif, CA, total bilan) est supérieur aux critères de l'ETI.

**LABORATOIRES/UNIVERSITÉS**

**AUTRE :**  
La catégorie AUTRE comprend :

- Les organismes ayant une forme juridique différente de celle des entreprises individuelles, des personnes morales de droit privé, ou des associations (exemple : établissements publics, collectivités locales, organismes privés spécialisés de type CPAM, GIE, ...).
- Les agences spatiales.
- Les organismes institutionnels (hors domaine Recherche comme CNRS ou INSU qui sont classés dans la catégorie Laboratoires-Universités).

Pour réaliser cette maintenance, il m'a fallu comprendre d'abord la logique GWT, puis celle utilisée pour la création des radios boutons dans cette application.

- L'utilisation d'une classe GWT, [ProvidesKey](#) (key) et de [Render](#) (affichage du rendu) qui permettent de manipuler des objets.
- Leurs

encapsulations dans un KeyRenderer puis d'en faire une classe générique ([GeneriqueKeyRendererDTO](#)) qui permettra la représentation graphique d'un élément d'une liste.

- Et enfin l'utilisation d'une DTO KeyRendererListeDTO qui permet d'afficher une liste d'éléments (type menu déroulant ou ensemble de radio boutons) et utilise ces éléments génériques pour faire le binding entre les éléments graphiques et les éléments métiers.

Vous trouverez une partie du code ainsi que l'IHM portant la modification dans la partie 10.1 Image BDCFou V2.0.0 de l'Annexe (**Figure-01 DM-23 01**, **Figure-02 DM-00023** et **Figure-03 DM-00023**) à la page 40.

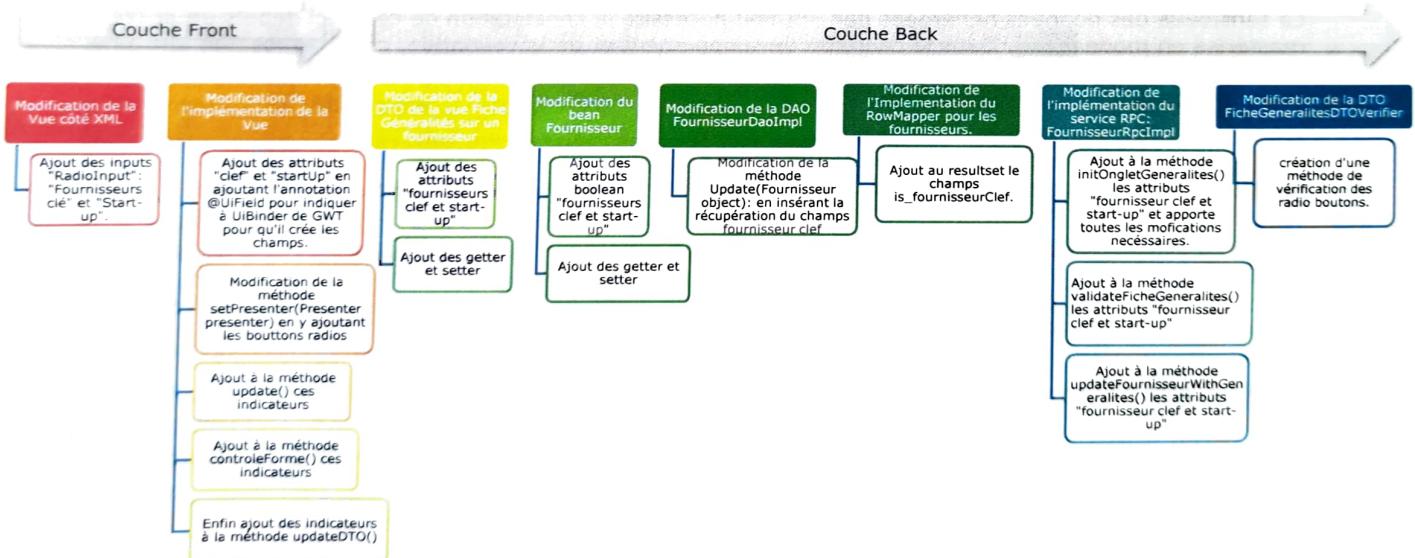
## D. Demande de Modification 00027.1 et 00027.2

Cette demande concerne l'ajout à l'IHM de deux indicateurs, **Fournisseur clé**, et **Start-up**. Ces indicateurs son nouveaux attributs à ajouter dans BDCFou avec des radio bouton oui / non au même niveau que « Spatial ». Sa valeur sera entrée manuellement dans l'application.

Cette demande l'intervention à tous les niveaux de l'application de l'IHM à la base de données.

Pour réaliser ces modifications, j'ai commencé par modifier l'IHM dans un 1<sup>er</sup> temps, puis développer la couche métier et enfin modifier la base de données afin d'intégrer les champs « fournisseur clé et start-up ».

Le schéma ci-dessous représente les travaux de développement réalisés :



Après avoir finis le développement de la vue et la partie back de cette nouvelle fonctionnalité de BDCFou, je suis passé à la troisième partie, la modification de la de la table fournisseur de la base de données car les champs « Fournisseur clef » et « Start-up » n'existaient pas.

Voici le script SQL de modification de la table ;

```

ALTER TABLE `fournisseur` ADD COLUMN `IS_CLEF` TINYINT NULL DEFAULT 0 AFTER `IS_SPATIAL`;
    > Ajout de la colonne Fournisseur clef.

ALTER TABLE `fournisseur` ADD COLUMN `IS_STARTUP` TINYINT NULL DEFAULT 0 AFTER `IS_CLEF`;
    > Ajout de la colonne Start-up.

```

Le développement de cette nouvelle fonctionnalité m'a permis de toucher à pratiquement l'ensemble des couches de l'application, mais aussi parfois de réécrire une ligne ou méthode développée par d'autres développeurs avant moi.

Pour voir le code développé pour la réalisation, voir les captures d'écran dans la partie DM-00027 de l'annexe : [Figure01](#) à [Figure08 de la page 42](#).

## E. Demande de Modification 00028

Cette demande concerne l'ajoute 4 colonnes supplémentaires (Chiffre d'affaires du titulaire, Effectifs du Titulaire, Indicateur clé, Indicateur innovant) dans la sélection des données à exporter.

Pour voir le code développé pour la réalisation, voir les captures d'écran dans la partie DM-00028 de l'annexe : [Figure-01 DM-00028](#) et [Figure-02 DM-00028](#) page 46.

## F. Demande de Modification 00029

Cette maintenance corrective portait sur la modification de l'IHM. La solution était juste un changement de label. Voir le résultat de la modification dans la partie DM-00029 de l'annexe : [Figure-01 DM-00029](#) à la page 47.

### 6.2. Crédation d'une procédure d'anonymisation

La sensibilité des données hébergées par BDCFou nécessite la consultation de celles-ci depuis un poste masterisé en mode **██████**. Dans le cadre du développement et de la validation, cette disposition entraîne des contraintes fortes qui ne sont pas en adéquation avec l'organisation de l'équipe intégrée. Cela a pour impacts :

- De ne pas disposer de la volumétrie cible,
- De ne pas disposer d'un environnement représentatif.

Face à ce constat, et pour mettre en œuvre un processus pérenne par rapport à la maintenance de cette application, le CNES et INETUM préconise la mise en place d'une procédure d'anonymisation des données de BDCFou. Ce processus sera appliqué sur un dump de base de données avant sa diffusion aux équipes TMA ou MCO.

Un script sera mis en place. Il devra être exécuté sur une copie de la base de données de production.

Pour chaque champ identifié, il réalise une requête « Update » en BDD pour en modifier la valeur. Il met à jour le schéma de la base de données pour stocker la valeur et effectuer une vérification.

Le type de requête « Update » à prévoir est le suivant :

- Mise en place d'une donnée numérique entière « aléatoire ».

Cette requête « Update » s'exécutent au sein d'une transaction terminée par un Commit. Le script SQL d'anonymisation est exécuté localement sur la plateforme de travail depuis un script bash.

Les données concernées sont :

Table	Champs	Type	Description		
fournisseur_inf_financiere	CA_CNES_FT	Long integer	Chiffre titulaire.	d'affaires	fournisseur
fournisseur_inf_financiere	CA_CNES_FP	Long integer	Chiffre d'affaires fournisseur payé.		
fournisseur_inf_financiere	CA	Long integer	Chiffre d'affaires		
fournisseur_inf_financiere	CA_SPATIAL	Long integer	Chiffre d'affaires dans le domaine spatial		
fournisseur_inf_financiere	resultats	Long integer	Résultats		

Les champs ci-dessus devront être mis à jour par le script d'anonymisation selon la procédure ci-dessous :

- Récupération de la liste des enregistrements contenant le champ à anonymiser.
- Création d'une valeur aléatoire numérique entière pour toutes les données concernées et dont la valeur est non NULL :

Table	Champs	Plage de valeurs
fournisseur_inf_financiere	CA_CNES_FT	Entre 0 et 50 000 000 000
fournisseur_inf_financiere	CA_CNES_FP	Entre 0 et 50 000 000 000
fournisseur_inf_financiere	CA	Entre 0 et 50 000 000 000
fournisseur_inf_financiere	CA_SPATIAL	Entre 0 et 50 000 000 000
fournisseur_inf_financiere	resultats	Entre - 2 000 000 000 et 6 000 000 000

- Mise à jour de la table contenant la valeur à anonymiser en ajoutant une colonne de contrôle préfixée par « ANONYM\_<nomDeLaColonneAAnonymiser> ».
- Mise à jour de l'enregistrement avec la valeur aléatoire générée.
- Insertion de la valeur aléatoire dans la colonne de contrôle.

## A. Procédure et environnement d'exécution

Le script **BDCFou\_Anonymisation.sh** permet de réaliser cette procédure.

Le dump produit sera lors transféré à la TMA.

Pour exécuter la procédure d'anonymisation, suivre les étapes suivantes :

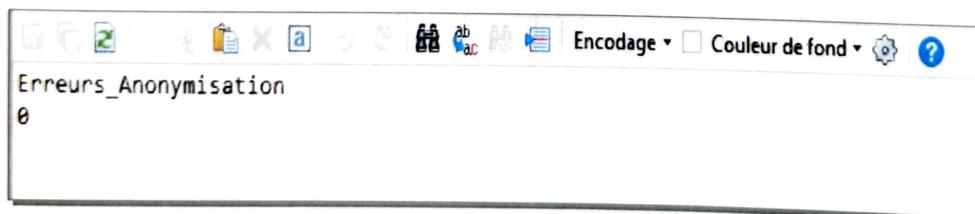
1. Dupliquer la base de données de production

Pour ne pas écraser les données réelles enregistrés dans la base, le script **BDCFou\_Anonymisation.sh** devra être exécuté sur une copie de la base de données de production!

2. Dézipper le fichier **BDCFou\_Anonymisation\_V2.0.0.zip**
3. Editer le fichier **BDCFou\_Anonymisation.sh** et renseigner les paramètres de connexion à la BDD **Hostname, Port, User, Database\_Name**.
4. Dans le shell vérifier les droits de lecture, écriture et exécution. A la racine du dossier de scripts, rentrer la commande `ls -al`  
Si pour le fichier **BDCFou\_Anonymisation.sh** les droits pour l'utilisateur Root ne sont pas ouverts (« rwx » n'apparaissent pas), exécuter la commande `chmod 0700 BDCFou_Anonymisation.sh` qui donnera les autorisations.

```
root@tu-srvinadm-q666:/home/BDCFouTeam/BDCFou-Anonymisation
[root@tu-srvinadm-q666 BDCFou-Anonymisation]# ls -al
total 28
drwxr-xr-x 2 root root 4096 Nov 25 14:56 .
drwxr-xr-x 6 root root 4096 Nov 24 14:03 ..
----- 1 root root 2403 Nov 25 14:39 BDCFou_Anonymisation.sh
----- 1 root root 439 Nov 15 10:08 Script_Drop.sql
----- 1 root root 2378 Nov 25 14:46 Script_Update_Anonymisation.sql
-rw-r--r-- 1 root root 24 Nov 26 11:17 StatutAnonymisation.txt
----- 1 root root 692 Nov 25 10:19 Verification_Erreurs.sql
[root@tu-srvinadm-q666 BDCFou-Anonymisation]# chmod 0700 BDCFou_Anonymisation.sh
[root@tu-srvinadm-q666 BDCFou-Anonymisation]# ls -al
total 28
drwxr-xr-x 2 root root 4096 Nov 25 14:56 .
drwxr-xr-x 6 root root 4096 Nov 24 14:03 ..
----- 1 root root 2403 Nov 25 14:39 BDCFou_Anonymisation.sh
----- 1 root root 439 Nov 15 10:08 Script_Drop.sql
----- 1 root root 2378 Nov 25 14:46 Script_Update_Anonymisation.sql
-rw-r--r-- 1 root root 24 Nov 26 11:17 StatutAnonymisation.txt
----- 1 root root 692 Nov 25 10:19 Verification_Erreurs.sql
[root@tu-srvinadm-q666 BDCFou-Anonymisation]#
```

5. En tant que Root, exécuter dans un Shell le fichier **BDCFou\_Anonymisation.sh** suivi du mot de passe de l'utilisateur de la base données. Exemple : `BDCFou_Anonymisation.sh password`
6. Suite à l'exécution du script, un fichier texte nommé **StatutAnonymisation.txt** décrivant le résultat de l'exécution est généré dans le même répertoire. Ce fichier indique le nombre d'erreurs relevées ainsi que les lignes éventuellement en erreur.  
Résultat du fichier texte généré dans le cas où la procédure s'est déroulée sans erreurs :



7. Créer un dump de la nouvelle base de données avec les nouvelles données anonymisées pour le transmettre à la TMA.
8. Supprimer la base de données anonymisée.

## B. Script SQL permettant l'anonymisation

Voici les scripts SQL qui seront exécutés par le script Bash :

### ➤ Script\_Drop.sql :

```
/*
DROP des colonnes anonymisées pour CA_CNES_FT, CA_CNES_FP, CA, CA_SPATIAL, resultats
*/
ALTER TABLE fournisseur_inf_financiere DROP COLUMN ANONYM_CA_CNES_FT;
ALTER TABLE fournisseur_inf_financiere DROP COLUMN ANONYM_CA_CNES_FP;
ALTER TABLE fournisseur_inf_financiere DROP COLUMN ANONYM_CA;
ALTER TABLE fournisseur_inf_financiere DROP COLUMN ANONYM_CA_SPATIAL;
ALTER TABLE fournisseur_inf_financiere DROP COLUMN ANONYM_RESULTATS;
```

### ➤ Script\_Update\_Anonymisation.sql

Solution 1 : Procédure stockée.

```
SET SQL_SAFE_UPDATES = 0;

-- Creation d'une procédure stockée pour anonymiser
drop procedure if exists anonymiser;
DELIMITER //
CREATE PROCEDURE anonymiser()
BEGIN
    Declaration des constantes
    DECLARE min_ca , max_ca, tranche_1, min_resultat, max_resultat BIGINT( ) ;
    SET min_ca = ;
    SET tranche_1 = ;
    SET max_ca= ;
    SET min_resultat= ;
    SET max_resultat= ;

    -- Creation des colonnes anonymes
    ALTER TABLE fournisseur_inf_financiere
    ADD COLUMN ANONYM_CA_CNES_FT BIGINT( ) NULL DEFAULT NULL AFTER CA_CNES_FT;
    ALTER TABLE fournisseur_inf_financiere
    ADD COLUMN ANONYM_CA_CNES_FP BIGINT( ) NULL DEFAULT NULL AFTER CA_CNES_FP;
    ALTER TABLE fournisseur_inf_financiere
    ADD COLUMN ANONYM_CA BIGINT( ) NULL DEFAULT NULL AFTER CA;
    ALTER TABLE fournisseur_inf_financiere
    ADD COLUMN ANONYM_CA_SPATIAL BIGINT( ) NULL DEFAULT NULL AFTER CA_SPATIAL;
    ALTER TABLE fournisseur_inf_financiere
    ADD COLUMN ANONYM_RESULTATS BIGINT( ) NULL DEFAULT NULL AFTER resultats;

    -- Génération des nombres random dans les colonnes anonymes
    update fournisseur_inf_financiere
    set ANONYM_CA_CNES_FT = floor(min_ca+RAND()*(tranche_1 - min_ca))
    WHERE CA_CNES_FT is not null;
    UPDATE fournisseur_inf_financiere
    SET CA_CNES_FT = ANONYM_CA_CNES_FT;

    update fournisseur_inf_financiere
    set ANONYM_CA_CNES_FT = floor(ANONYM_CA_CNES_FT+RAND() * (max_ca - ANONYM_CA_CNES_FT))
    WHERE CA_CNES_FT is not null;
    UPDATE fournisseur_inf_financiere
    SET CA_CNES_FT = ANONYM_CA_CNES_FT;

    update fournisseur_inf_financiere
    set ANONYM_CA_SPATIAL = floor(min_ca + RAND() * (tranche_1 - min_ca))
    WHERE CA_SPATIAL is not null;
    UPDATE fournisseur_inf_financiere
    SET CA_SPATIAL = ANONYM_CA_SPATIAL;

    update fournisseur_inf_financiere
    set ANONYM_CA = floor(min_ca+RAND() * (max_ca - min_ca))
    WHERE CA is not null;
    UPDATE fournisseur_inf_financiere
    SET CA = ANONYM_CA;

    update fournisseur_inf_financiere
    set ANONYM_RESULTATS = floor(min_resultat + RAND() * (max_resultat - min_resultat))
    WHERE resultats is not null;
    UPDATE fournisseur_inf_financiere
    SET resultats = ANONYM_RESULTATS;

END
DELIMITER //

-- Appel de la procédure stockée
CALL anonymiser();
```

Mais cette proposition est déclinée parce que la procédure est stockée justement et peut ramener des soucis de sécurité.

J'ai donc opté pour cette deuxième solution plus simple.

```
--> Creation des colonnes anonymisées pour CA_CNES_FT, CA_CNES_PP, CA, CA_SPATIAL, resultats
ALTER TABLE fournisseur_inf_financiere
ADD COLUMN ANONYM_CA_CNES_FT bigint NULL DEFAULT null AFTER CA_CNES_FT;
ALTER TABLE fournisseur_inf_financiere
ADD COLUMN ANONYM_CA_CNES_PP bigint NULL DEFAULT null AFTER CA_CNES_PP;
ALTER TABLE fournisseur_inf_financiere
ADD COLUMN ANONYM_CA bigint NULL DEFAULT null AFTER CA;
ALTER TABLE fournisseur_inf_financiere
ADD COLUMN ANONYM_CA_SPATIAL bigint NULL DEFAULT null AFTER CA_SPATIAL;
ALTER TABLE fournisseur_inf_financiere
ADD COLUMN ANONYM_RESULTATS bigint NULL DEFAULT null AFTER RESULTATS;

--> Remplacement de la liste des enregistrements contenant le champ à anonymiser
--> par une valeur random numérique entière pour toutes les données
--> concernées et dont la valeur est non nulle
--> Mise à jour de la table contenant la valeur à anonymiser en ajoutant une colonne
--> de contrôle préfixée par «ANONYM_»+nomDeLaColonneÀAnonymiser»
SET @MIN_CA= ;
SET @MAX_CA= ;
SET @MIN_RESULTAT= ;
SET @MAX_RESULTAT= ;
UPDATE fournisseur_inf_financiere SET ANONYM_CA_CNES_FT = RAND() *@MAX_CA WHERE ANONYM_CA_CNES_FT IS NOT NULL;
-- pour déterminer un max et un min au randon : (RAND() * (maximumValue-minimumValue)) + minimumValue;
UPDATE fournisseur_inf_financiere SET ANONYM_CA_CNES_PP = (RAND() * (@MAX_CA-CNES_FT-@MIN_CA)) + (@MIN_CA) WHERE ANONYM_CA_CNES_PP IS NOT NULL;
UPDATE fournisseur_inf_financiere SET ANONYM_CA = RAND() *@MAX_CA WHERE CA IS NOT NULL;
UPDATE fournisseur_inf_financiere SET ANONYM_CA_SPATIAL = RAND() *@MAX_CA WHERE CA_SPATIAL IS NOT NULL;
UPDATE fournisseur_inf_financiere SET ANONYM_RESULTATS = (RAND() * (@MAX_RESULTAT-@MIN_RESULTAT)) + (@MIN_RESULTAT) WHERE RESULTATS IS NOT NULL;

--> Insertion des données anonymisées dans la colonne d'origine */
UPDATE fournisseur_inf_financiere SET CA_CNES_FT = ANONYM_CA_CNES_FT;
UPDATE fournisseur_inf_financiere SET CA_CNES_PP = ANONYM_CA_CNES_PP;
UPDATE fournisseur_inf_financiere SET CA = ANONYM_CA;
UPDATE fournisseur_inf_financiere SET CA_SPATIAL = ANONYM_CA_SPATIAL;
UPDATE fournisseur_inf_financiere SET RESULTATS = ANONYM_RESULTATS;
```

#### ➤ Script\_VérificationErreurs.sql

```
--> Script de vérification de l'anonymisation.
SET SQL_SAFE_UPDATES = ;
-- Recherche de lignes d'erreurs.
SELECT count(*) as Erreurs_Anonymisation
FROM fournisseur_inf_financiere
WHERE CA <> ANONYM_CA
      OR CA_CNES_PP <> ANONYM_CA_CNES_PP
      OR CA_CNES_FT <> ANONYM_CA_CNES_FT
      OR CA_SPATIAL <> ANONYM_CA_SPATIAL
      OR RESULTATS <>ANONYM_RESULTATS;

--> Script de vérification de l'anonymisation.
SET SQL_SAFE_UPDATES = ;
-- Afficher les lignes d'erreurs si il y'en a.
SELECT *
FROM fournisseur_inf_financiere
WHERE CA != ANONYM_CA
      OR CA_CNES_PP != ANONYM_CA_CNES_PP
      OR CA_CNES_FT != ANONYM_CA_CNES_FT
      OR CA_SPATIAL != ANONYM_CA_SPATIAL
      OR RESULTATS != ANONYM_RESULTATS;
```

## C. Script Bash exécutant les scripts SQL

Ce script Bash Unix est destiné à l'administrateur de la base de données.

```
#!/bin/bash

echo "Entrez les informations de connexion à la BDD"
HOSTNAME=
PORT=
USER=
DATABASE_NAME=

#Connexion à la BDD et lancement du script d'anonymisation
mysql -h $HOSTNAME --port=$PORT -u $USER --password=$! $DATABASE_NAME < script_update_Anonymisation.sql

##Teste si réussite (0) ou échec (nul) de la dernière commande effectuée, lors l'exécution du script d'anonymisation
if [ "$?" -eq 0 ]
then
    echo «Anonymisation des tables effectuée avec succès»
else
    echo «Anonymisation des tables effectuée avec échec»
fi

#Connexion à la BDD et lancement du script de création du fichier de vérification des données anonymisées (si pas de données fournisseur présentes = anonymisation ok)
mysql -h $HOSTNAME --port=$PORT -u $USER --password=$! $DATABASE_NAME < Verification_Erreurs.sql> StatutAnonymisation.txt

##Génération du fichier .txt réussi
if [ "$?" -eq 0 ]
then
    echo «Fichier de vérification créé avec succès»
else
    echo «Echec de génération du fichier de vérification»
fi

##Si échec du traitement précédent, à savoir le lancement du script d'anonymisation alors
else
    # Connexion à la base bdcfou_team et lancement du script sql de drop des tables
    mysql -h $HOSTNAME --port=$PORT -u $USER --password=$! $DATABASE_NAME < Script_Drop.sql

    ##Suppression de la suppression des tables
    if [ "$?" -eq 0 ]
    then
        echo "Suppression des tables effectuée avec succès"
        #Connexion à la BDD et lancement du script d'anonymisation
        mysql -h $HOSTNAME --port=$PORT -u $USER --password=$! $DATABASE_NAME < Script_Update_Anonymisation.sql

        if [ "$?" -eq 0 ]
        then
            echo «Anonymisation des tables effectuée avec succès»
        else
            echo «Anonymisation des tables effectuée avec échec»
        fi

        #Connexion à la BDD et lancement du script de création du fichier de vérification des données anonymisées (si pas de données fournisseur présentes = anonymisation ok)
        mysql -h $HOSTNAME --port=$PORT -u $USER --password=$! $DATABASE_NAME < Verification_Erreurs.sql> StatutAnonymisation.txt

        ##Génération du fichier .txt réussie
        if [ "$?" -eq 0 ]
        then
            echo «Fichier de vérification créé avec succès»
        else
            echo «Echec de génération du fichier de vérification»
        fi
    fi
fi
fi
```

Cette phase m'a apporté beaucoup :

- Prise de conscience concernant la sécurité des données.
- Apprendre à faire un script Bash en langage Unix.
- Travailler avec des machines virtuelles pour déployer et tester la base de données.

## 7. Tests et validation

Une fois le développement fini, je vous parlerai dans cette partie de la dernière phase de réalisation des maintenances, les tests et la vérification de la qualité.

### A. Cahier de recettes

Pour s'assurer que les modifications effectuées soient maintenables, fiables et surtout qu'ils correspondent aux souhaits du client. Dans, j'ai apporté des modifications au cahier de recette déjà

existant pour valider le fonctionnement des fonctionnalités apportés. Vous trouverez le déroulé de la stratégie de recette qui décrit le contexte, la méthode de déroulement des tests, et les approbateurs dans la capture d'écran située dans la partie Annexe Figure-01 Test de la page 46.

## B. Vérification de la qualité avec SonarQube

Enfin pour faciliter la maintenance et avoir un code de qualité, le respect des règles et normes du code, la documentation du code, la vérification de la duplication du code et la vérification de la vulnérabilités potentielles, etc... j'ai utilisé un serveur SonarQube pour scanner le code avant et après la maintenance pour vérifier si il y a eu des régressions. Le résultat était à la hauteur de mes attentes. Vous trouverez une capture d'écran de ces dernier dans la partie annexe test et validation Figure-01 Test et Figure-02 Test à la page 48.

# 8. Bilan du projet

Globalement les objectifs fixés dans leur ensemble ont été atteints, même si il reste encore une partie des maintenances évolutives qui ne sont pas développées ou développées partiellement faute de clarifications des besoins de la part du client. Les responsables attendent à ce jour un retour de la part de ce dernier.

Après la fin du développement, j'ai envoyé la version 2.0.0 de BDCFou pour qu'elle soit testée. Les tâches demandées ont été faite et l'application est livrée et tourne en préproduction.

Au terme de ce stage, j'ai réussi à prendre en main les différentes technologies du périmètre pour la réalisation du projet, même si au départ ça m'avait demandé plus d'investissement.

# 9. Bilan personnel et perspectives

Les 3 mois que j'ai passé en stage au sein d'Inetum se sont écoulés rapidement de mon point de vue. J'ai particulièrement apprécié la collaboration et l'esprit de groupe qui y régnait dans ce service. Je suis satisfait d'avoir mené à bien les tâches qui m'ont été confiées.

Cette première expérience m'a permis de mobiliser l'ensemble des compétence acquises durant ma formation, de les approfondir, de gagner en autonomie, d'élargir mes compétences en abordant des technologies telles que GWT ou langage Unix.

Cette expérience m'a aussi permis de découvrir le monde du développement dans une entreprise dédiée au numérique et interagir avec les acteurs du projet ou d'autres, Delivery Manager, Responsable Service, Cheffe de projet, équipes de développement etc... qui ont enrichi ma vision de ce milieu et ma posture.

Enfin, elle s'est soldée par un retour positif de mon stage et une proposition de CDI que j'ai accepté. C'est donc confiant que j'aborde la suite pour progresser jusqu'à l'autonomie complète.

# 10. Annexes

## 10.1. Images BDCFou V2.0.0

### A. DM-00011

```

package fr.cnrs.bdcfou.bo;
import java.util.EnumSet;

/**
 * Cette ENUM permet de faire le choix entre OUI, NON, TOUS <br>
 * BDCFou V2.0.0 DM Correctives: DM-00011-Ajout d'un bouton "Tous" à
 * Fournisseur suivis
 *
 * @author INETUM Group
 */

public enum ChoixMultiple implements IndexKeyValueEnum {
    OUI(1, 1, "Oui"), NON(0, 0, "Non"), TOUS(2, 2, "Tous");

    /**
     * Index de l'élément de l'énumération dans la liste. (Ordre)
     */
    private int index;

    /**
     * Clef de l'élément de l'énumération.
     */
    private int key;

    /**
     * Valeur de l'élément de l'énumération.
     */
    private String value;

    /**
     * Map des clefs / Enum.
     */
    private static final Map<Integer, ChoixMultiple> mapChoixMultiple = new HashMap<Integer, ChoixMultiple>();

    static {
        for (ChoixMultiple choixBinaire : EnumSet.allOf(ChoixMultiple.class)) {
            mapChoixMultiple.put(choixBinaire.getKey(), choixBinaire);
        }
    }

    /**
     * Constructeur
     *
     * @param index index du choix.
     * @param key   Clé du choix.
     * @param value Valeur du choix.
     */
    private ChoixMultiple(int index, int key, String value) {
        this.index = index;
        this.key = key;
        this.value = value;
    }

    @Override
    public int getIndex() {
        return index;
    }
}

```

Figure-03\_DM-00011: Création d'une enum avec 3 choix OUI NON

The screenshot shows a web-based application titled 'Connaissance des fournisseurs' with the subtitle 'DIFFUSION LIMITÉE CNES'. At the top, there are search filters for 'Nom', 'Domaine d'activités' (with a dropdown menu showing 'FR', 'ENF', and 'RDM'), 'Fournisseur suivi' (with radio buttons for 'Oui' and 'Non'), and a 'Toutes' dropdown menu. Below the filters is a table titled 'Résultat de la recherche' with columns: 'Nom courant', 'Raison sociale', 'Pays', 'Dernière mai', and 'Domaine d'activités'. The table currently displays one row of data.

Figure\_DM-11\_01: Ajout des fonctionnalités Tous et Toutes à l'IHM

```

//BDCCFou V2.0.0 DM-00011 Ajout d'un boolean "Tous" à la signature de getListeIdFournisseur().
//      Pour selectionner tous les fournisseurs suivis ou pas.
*/
/* {#inheritDoc}
 */
public List<String> getListeIdFournisseur(String nom,
    List<DomaineActivite> listeDomaineActivite,
    RegionMonde regionMonde, boolean isSuivi, boolean isTous) throws BDCCFouException {
    List<String> listeIdFournisseur = null;
    boolean isEmptyNom = StringUtils.isEmpty(nom);
    boolean isEmptyDomaineActivite = listeDomaineActivite == null
        || listeDomaineActivite.isEmpty();
    StringBuilder sql = new StringBuilder();
    sql.append("select distinct f.id_spécifique ");
    sql.append("from fournisseur f ");
    if (!isEmptyNom) {
        sql.append("left outer join historisation h ");
        sql.append("on f.id_spécifique = h.id_spécifique ");
        sql.append("and h.table_name = 'fournisseur' ");
        sql.append("and (h.table_column = 'nom_courant' ");
        sql.append("or h.table_column = 'raison_sociale') ");
    }
    if (!isEmptyDomaineActivite) {
        sql.append("inner join fournisseur_eco_domaine fed ");
        sql.append("on f.id_spécifique = fed.id_spécifique ");
    }
    // BDCCFou V2.0.0 DM-00011 Séparation du (where f.is_suivi = :isSuivi)
    sql.append("where 1=1 ");

    //BDCCFou V2.0.0 DM-00011 Ajout d'une condition vérifiant si le bouton est sélectionné.
    if (isTous) {
        sql.append("and f.is_suivi = :isSuivi ");
    }
    if (!isEmptyNom) {
        sql.append("and (f.nom_courant like :nom ");
        sql.append("or f.raison_sociale like :nom ");
        sql.append("or h.val like :nom) ");
    }

    //V2.0.0 DM-00021 création d'une option "toutes" dans la liste régions du monde
    //si le choix de région du monde n'est pas nul et n'est pas "Toutes" on ajoute cette ligne à la requête
    if (regionMonde != null && regionMonde.getId() != BDCCFouConstantes.ID_TOUTES_REGIONS_MONDE) {
        sql.append("and f.id_region_monde = :idRegion ");
    }
    if (!isEmptyDomaineActivite) {
        sql.append("and fed.id_eco_domaine in (:listeIdDomaineActivite) ");
    }
    sql.append("order by f.nom_courant");

    Map<String, Object> args = new HashMap<String, Object>();
    if (!isEmptyNom) {
        args.put("nom", "%" + nom + "%");
    }
    if (!isEmptyDomaineActivite) {
        List<Integer> listeIdDomaineActivite = new ArrayList<Integer>(
            listeDomaineActivite.size());
        for (DomaineActivite domaineActivite : listeDomaineActivite) {
            listeIdDomaineActivite.add(domaineActivite.getId());
        }
    }
}

```

**Figure-02 DM-00011 :** Ajout d'un bouton radio (couche DAO)

## B. DM-00021

```

package fr.cnes.bdcfou.bo;

/*
 * BDCFou V2.0.0 Classe répertoriant les constantes utilisées dans le code.
 * @author Farhan Barreh
 */

public class BDCFouConstantes {

    private BDCFouConstantes() {
    }

    public static final String TOUTES = "Toutes";
    public static final int ID_TOUTES_REGIONS_MONDE = -1;
    public static final String TXTCA = "Chiffre d'Affaire du Fournisseur";
    public static final String TXTFOURNISSEUR = "PME : \r\n"
        + "La catégorie des petites et moyennes entreprises (PME, incluant les TPE) est constituée des entreprises qui : \r\n"
        + "• emploient moins de 250 personnes ; \r\n"
        + "• ont un chiffre d'affaires annuel n'excédant pas 50 millions d'euros ou un total de bilan n'excédant pas 43 millions d'euros ; \r\n"
        + "et dont le capital n'est pas détenu à plus de 50 % par une Entreprise de Taille ... (ETI) ou une grande entreprise (GE). \r\n"
        + "\r\n"
        + "PME": \r\n"
        + "La catégorie PME" (Petite et moyenne entreprise « étendue », catégorie propre au CNES) est constituée des entreprises qui : \r\n"
        + "• emploient moins de 250 personnes ; \r\n"
        + "• ont un chiffre d'affaires annuel n'excédant pas 50 millions d'euros ou un total de bilan n'excédant pas 43 millions d'euros ; \r\n"
        + "et dont le capital est détenu à plus de 50 % par une ETI ou un GE. \r\n"
        + "NB : NB : PME (droit français) = PME + PME dans BDCFou (droit EU)\r\n"
        + "ETI (Entreprises de Taille Intermédiaire):\r\n"
        + "La catégorie des ETI est constituée des entreprises qui n'appartiennent pas aux catégories des PME ou des PME" et qui : \r\n"
        + "• emploient moins de 5 000 personnes ;\r\n"
        + "• ont un chiffre d'affaires annuel n'excédant pas 1,5 milliard € ou un total de bilan n'excédant pas 2 milliards €. \r\n"
        + "GE (Grandes Entreprises) : entreprises dont l'un des critères (effectif, CA, total bilan) est supérieur aux critères de l'ETI. \r\n"
        + "LABORATOIRES/UNIVERSITES\r\n"
        + "AUTRE :\r\n"
        + "Le catégorie AUTRE comprend :\r\n"
        + "• Les organismes ayant une forme juridique différente de celle des entreprises individuelles, des personnes morales de droit privé, ou des assoc
        + "• Les agences spatiales,\r\n"
        + "• Les organismes institutionnels (hors domaine Recherche comme CNRS ou INSU qui sont classés dans la catégorie Laboratoires-Universités).\r\n"
        + "";
}

}

```

**Figure-01 DM-00021 :** Création d'une classe pour les constantes.

```

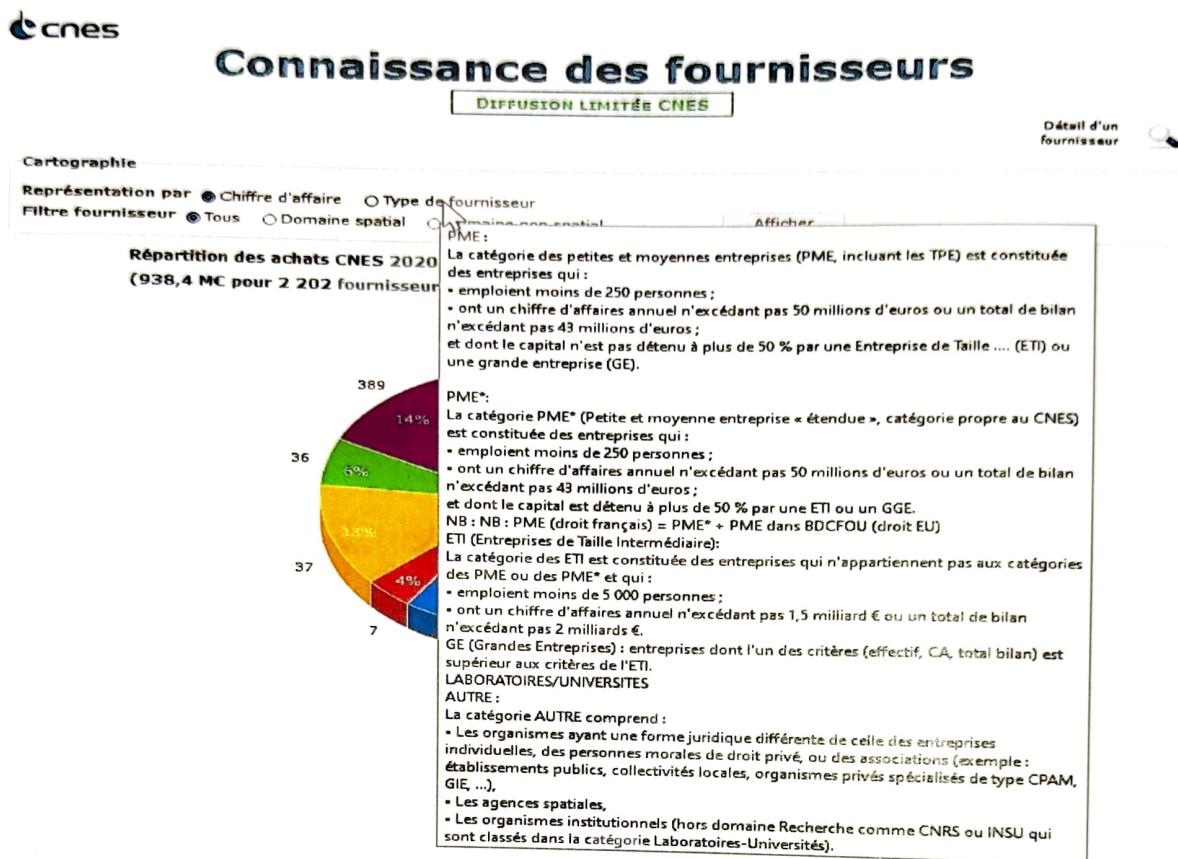
// REGIONS MONDE
    même méthode que pour les domaines d'activités
List<RegionMonde> listeRegionsMondeFromSI = new ArrayList<RegionMonde>();
// V2.0.0 DM-00021 création d'une option "toutes" dans la liste régions du monde
// Ajout d'une région fictive "Toutes" sélectionné par défaut
RegionMonde all = new RegionMonde();
all.setId(BDCFouConstantes.ID_TOUTES_REGIONS_MONDE);
all.setLibelle(BDCFouConstantes.TOUTES);
listeRegionsMondeFromSI.add(all);
// on rajoute la liste de régions les options récupérées dans la BDD
listeRegionsMondeFromSI.addAll(referentielBusinessService.getListeRegionMonde());

if (listeRegionsMondeFromSI != null && !listeRegionsMondeFromSI.isEmpty()) {
    KeyRenderListDTO<GenericKeyRenderDTO> regionsMondeKRLForClient = new KeyRenderListDTO<GenericKeyRenderDTO>();
    List<GenericKeyRenderDTO> regionsMonde = new ArrayList<GenericKeyRenderDTO>();
    for (RegionMonde regionMondeMetier : listeRegionsMondeFromSI) {
        GenericKeyRenderDTO regionMondeMetierDTO = new GenericKeyRenderDTO(
            String.valueOf(regionMondeMetier.getId()), regionMondeMetier.getLibelle());
        regionsMonde.add(regionMondeMetierDTO);
    }
    regionsMondeKRLForClient.setElements(regionsMonde);
    anonymise(regionsMondeKRLForClient);
    dtoRetour.setRegionMonde(regionsMondeKRLForClient);
}

```

**Figure-02 DM-00021 :** Ajout de la région « Toutes ».

## C. DM-00023

**Figure-01 DM-00023 :** ajout d'un Tool tip

```

/*
package fr.cnes.bdcfou.bo;

/*
 * BDCFou V2.0.0 Classe répertoriant les constantes utilisées dans le code.
 * @author Farhan BARREH
 */
public class BDCFouConstantes {
    public final static String TOUTES = "Toutes";
    public final static int ID_TOUTES_REGIONS_MONDE = -1;
    public static final String TXTCA = "Chiffre d'Affaire du Fournisseur";
    public static final String TXTFOURNISSEUR = "PME : \r\n"
        + "La catégorie des petites et moyennes entreprises (PME, incluant les TPE) est constituée des entreprises qui : \r\n"
        + "• emploient moins de 250 personnes ; \r\n"
        + "• ont un chiffre d'affaires annuel n'excédant pas 50 millions d'euros ou un total de bilan n'excédant pas 43 millions d'euros ; \r\n"
        + "et dont le capital n'est pas détenu à plus de 50 % par une Entreprise de Taille .... (ETI) ou une grande entreprise (GE). \r\n"
        + "PME* : \r\n"
        + "La catégorie PME* (Petite et moyenne entreprise « étendue », catégorie propre au CNES) est constituée des entreprises qui : \r\n"
        + "• emploient moins de 250 personnes ; \r\n"
        + "• ont un chiffre d'affaires annuel n'excédant pas 50 millions d'euros ou un total de bilan n'excédant pas 43 millions d'euros ; \r\n"
        + "et dont le capital est détenu à plus de 50 % par une ETI ou un GGE.\r\n"
        + "NB : NB : PME (droit français) = PME* + PME dans BDCFou (droit EU)\r\n"
        + "ETI (Entreprises de Taille Intermédiaire):\r\n"
        + "La catégorie des ETI est constituée des entreprises qui n'appartiennent pas aux catégories des PME ou des PME* et qui : \r\n"
        + "• emploient moins de 5 000 personnes ;\r\n"
        + "• ont un chiffre d'affaires annuel n'excédant pas 1,5 milliard € ou un total de bilan n'excédant pas 2 milliards €. \r\n"
        + "GE (Grandes Entreprises): entreprises dont l'un des critères (effectif, CA, total bilan) est supérieur aux critères de l'ETI. \r\n"
        + "AUTRE :\r\n"
        + "La catégorie AUTRE comprend :\r\n"
        + "• Les organismes ayant une forme juridique différente de celle des entreprises individuelles, des personnes morales de droit privé, ou des associations (exemple : établissements publics, collectivités locales, organismes privés spécialisés de type CPAM, GIE, ...),\r\n"
        + "• Les agences spatiales,\r\n"
        + "• Les organismes institutionnels (hors domaine Recherche comme CNRS ou INSU qui sont classés dans la catégorie Laboratoires-Universités).\r\n"
}

```

**Figure-06 DM-00023:** Text tooltip

```

import com.google.gwt.user.client.rpc.IsSerializable;
public class GenericKeyRenderDTO extends KeyRenderDTO<GenericKeyRenderDTO> {
    private String key;
    private String render;
    // BDCFou V2.0.0 DM-00023
    private String tooltip;
    // BDCFou V2.0.0 DM-00023
    public String getTooltip() {
        return tooltip;
    }
    public GenericKeyRenderDTO() {
    }
    public GenericKeyRenderDTO(String key, String render) {
        this.key = key;
        this.render = render;
    }
    public String getKey() {
        return key;
    }
    public void setKey(String key) {
        this.key = key;
    }
    public String getRender() {
        return render;
    }
}
/*
 * BDCFou V2.0.0 Ajout d'un Tooltip

```

**Figure-3 DM-00023:** Text tooltip

```

/*
 * @see fr.cnes.bdcfou.app.client.ui.generique.AbstractComposite#unsetErrorBehaviour()
 */
@Override
protected void unsetErrorBehaviour() {
}

/*
 * Crée le groupe de radio-boutons du champ.
 */
private void creerRadioBoutons() {
    if (radioBoutons == null) {
        radioBoutons = new ArrayList<RadioButton>();
    }
    radioBoutons.clear();
    if (data != null && data.getElements() != null) {
        RadioButton radioBtn;
        // Pour chaque élément de la KeyRenderListDTO d'entrée
        for (T dataItem : data.getElements()) {
            // Renseigner le radio-bouton avec les données de l'élément
            radioBtn = new RadioButton("RI-" + String.valueOf(groupName), dataItem.getRenderer().render(dataItem));
            radioBtn.setFormValue((String) dataItem.getKeyProvider().getKey(dataItem));
            // BDCFou V2.0.0 DM-00023 Ajout de tooltip
            if (dataItem instanceof GenericKeyRenderDTO) {
                radioBtn.setTitle(((GenericKeyRenderDTO) dataItem).getTooltip());
            }
            radioBoutons.add(radioBtn);
        }
        // Sélectionner la clé
        setSelectedKeyAsValue();
        groupName++;
    }
}

```

**Figure-4 DM-00023:** Text tooltip

## D. DM-00027.1 ET 00027.2

```

<ns:input="urn:import:fr.cnes.bdcfou.app.client.ui.generique.input"
ns:historizable="urn:import:fr.cnes.bdcfou.app.client.ui.generique.historizable"
ns:tableau="urn:import:fr.cnes.bdcfou.app.client.ui.generique.table"
ns:upload="urn:import:fr.cnes.bdcfou.app.client.ui.generique.fileupload"
ns:res="urn:with:fr.cnes.bdcfou.app.client.resources.BdcfouResources"
ns:op="urn:import:com.sun.org.apache.xpath.internal.operations">
<ui:with type="fr.cnes.bdcfou.app.client.resources.BdcfouResources"
    field="res" />
<ui:import field="fr.cnes.bdcfou.app.shared.verifier.FormatConstraint.*" />
<ui:import field="com.google.gwt.user.client.ui.FormPanel.ENCODING_MULTIPART" />
<ui:import field="com.google.gwt.user.client.ui.FormPanel.METHOD_POST" />

<g:HTMLPanel>
    <g:FlowPanel styleName="{res.styleGlobal.row}">
        <historizable:HistorizableTextInput
            ui:field="raisonSociale" libelle="Raison sociale" maxLength="(TEXTE_60_REQUIRED.getMaxLength)"
            historizable="true" correctable="true" width="200px" marginRight="20" />
        <historizable:HistorizableSelect
            ui:field="typeFournisseur" libelle="Type de fournisseur"
            historizable="true" marginRight="20" />
        <input:RadioInput ui:field="spatial" intitule="Spatial" addStyleNames="{res.styleFiche.spatial}" />
    </g:FlowPanel>

```

**Figure-01 DM-00027.1:** Avant ajout des nouveaux indicateur fournisseur clef et start-up

```

<!DOCTYPE ui:UiBinder SYSTEM "http://dl.google.com/gwt/DTD/xhtml.ent">
<ui:UiBinder xmlns:ui="urn:ui:com.google.gwt.ui:Binder"
    xmlns:g="urn:import:com.google.gwt.user.client.ui" xmlns:generic="urn:import:fr.cnes.bdcfou.app.client.ui.generique"
    xmlns:input="urn:import:fr.cnes.bdcfou.app.client.ui.generique.input"
    xmlns:historizable="urn:import:fr.cnes.bdcfou.app.client.ui.generique.historizable"
    xmlns:tableau="urn:import:fr.cnes.bdcfou.app.client.ui.generique.table"
    xmlns:upload="urn:import:fr.cnes.bdcfou.app.client.ui.generique.fileupload"
    xmlns:res="urn:with:fr.cnes.bdcfou.app.client.resources.BdcfouResources"
    xmlns:op="urn:import:com.sun.org.apache.xpath.internal.operations">
<ui:with type="fr.cnes.bdcfou.app.client.resources.BdcfouResources"
    field="res" />
<ui:import field="fr.cnes.bdcfou.app.shared.verifier.FormatConstraint.*" />
<ui:import field="com.google.gwt.user.client.ui.FormPanel.ENCODING_MULTIPART" />
<ui:import field="com.google.gwt.user.client.ui.FormPanel.METHOD_POST" />

<g:HTMLPanel>
    <g:FlowPanel styleName="{res.styleGlobal.row}">
        <historizable:HistorizableTextInput
            ui:field="raisonSociale" libelle="Raison sociale" maxLength="(TEXTE_60_REQUIRED.getMaxLength)"
            historizable="true" correctable="true" width="200px" marginRight="20" />
        <historizable:HistorizableSelect
            ui:field="typeFournisseur" libelle="Type de fournisseur"
            historizable="true" marginRight="20" />
    </g:FlowPanel>
    <!-- BDCFou V2.0.0 DM-00027 Ajout d'un nouvel attribut "Fournisseur Clef" -->
    <g:FlowPanel styleName="{res.styleGlobal.row}">
        <input:RadioInput ui:field="spatial" intitule="Domaine spatial" addStyleNames="{res.styleFiche.spatial}" />
        <input:RadioInput ui:field="fournisseurClef" intitule="Fournisseur clef" addStyleNames="{res.styleFiche.spatial}" marginRight="20" />
        <input:RadioInput ui:field="startUp" intitule="Start-Up" addStyleNames="{res.styleFiche.spatial}" />
    </g:FlowPanel>

```

**Figure-01 DM-00027.1:** Après ajout des nouveaux indicateur fournisseur clef et start-up

BDCFou-App>src >main>java >fr >cnes>bdcfou >app>client>ui>fiche>generalites >FicheGeneralitesViewImpl

Page d'implémentation de la vue, à qui on passe où on appelle le champ «clef» créé dans la vue  
Ligne 85 ajoute

```

    /** Clef. */
    @UiField
    RadioInput<GenericKeyRenderDTO> clef;

    /** start up. */
    @UiField
    RadioInput<GenericKeyRenderDTO> startUp;

```

```

Ligne 463      /**
     * {@inheritDoc}
     */
    @Override
    public void setPresenter(Presenter presenter) {
        this.presenter = presenter;
    }

```

```

        // Etat commun aux 3 rôles
        siren.setEtat(EtatComposant.CONSULTATION);
        ...
        switch (this.presenter.getUserRole()) {
            case ADMIN:
                ...
                spatial.setEtat(EtatComposant.MODIFICATION);
                clef.setEtat(EtatComposant.MODIFICATION);
                startUp.setEtat(EtatComposant.MODIFICATION);
                ...
                break;
            case DAJ:
            case SALARIE:
                ...
                spatial.setEtat(EtatComposant.CONSULTATION);
                clef.setEtat(EtatComposant. CONSULTATION);
                startUp.setEtat(EtatComposant.CONSULTATION);
                domainesActivite.setEtat(EtatComposant.CONSULTATION);
                ...
                break;
            default:
                break;
    }

    Ligne 612
    @Override
    public void update() {
        // Reset du form pour remettre à zero les files chooser
        formPJsGeneralite.reset();
        ...
    }

    Ligne 512    clef.setListData(dto.getClef());
    startUp.setListData(dto.getStartUp());

    Ligne 604

    public Map<String, List<String>> controlForm() {
        FicheGeneralitesDTO dto = presenter.getDTO().getOngletGeneralites();
        ...
    }

    Ligne 626    result = FicheGeneralitesDTOVerifier.checkClef(dto.getClef()
        .getKeys(), dto.getClef().getSelectedKeys());
        clef.setError(!result.isEmpty());
        violations.putAll(result);

        result = FicheGeneralitesDTOVerifier.checkStartUp(dto.getStartUp()
        .getKeys(), dto.getStartUp().getSelectedKeys());
        startUp.setError(!result.isEmpty());
        violations.putAll(result);

    Ligne 664
    /**
     * {@inheritDoc}
     * @see fr.cnes.bdcfou.app.client.ui.fiche.IFicheView#updateDTO()
     */
    @Override
    public void updateDTO() {FicheGeneralitesDTO dto =
    presenter.getDTO().getOngletGeneralites();
    ...
    }

    ligne 673    clef.updateSelectedKey();
    startUp.updateSelectedKey();

```

**Figure-03 DM-00027 :** ajout des nouveaux indicateur fournisseur clef et start-up

```
BDCFou-
App>src>main>java>fr>cnes>bdcfou>app>shared>verifier>fiche>generalite>FicheGener
aliteDTOVerifier
Ligne 138 ajout des attributs
    public static Map<String, List<String>> checkClef(
        List<String> listeValue, List<String> listeSelected) {
        return checkRadio("Fournisseur clef", listeValue, listeSelected);
    }

    public static Map<String, List<String>> checkStartUp(
        List<String> listeValue, List<String> listeSelected) {
        return checkRadio("StartUp", listeValue, listeSelected);
    }
```

**Figure-04 DM-00027 :** ajout des nouveaux indicateur fournisseur clef et start-up

```
BDCFou-SI >src >main >java >fr >cnes >bdcfou >bo > Fournisseur.java
Ligne 34 ajout
    /**
     * Boolean indiquant si le fournisseur est spatial ou non. */
    private boolean clef;
    /**
     * Boolean indiquant si le fournisseur est une start up ou non. */
    private boolean startUp;
Ligne 156 ajout
    public boolean isClef() {
        return clef;
    }
    public void setClef(boolean clef) {
        this.clef = clef;
    }

    /**
     * Getter indiquant si le fournisseur est une start up ou non.
     *
     * @return Boolean indiquant si le fournisseur est une start up ou non
     */
    public boolean isStartUp() {
        return startUp;
    }

    /**
     * Setter indiquant si le fournisseur est une start up ou non.
     *
     * @param Start Up
     *          Boolean indiquant si le fournisseur est une start up ou non
     */
    public void setStartUp(boolean startUp) {
        this.startUp = startUp;
    }
```

**Figure-03 DM-00027 :** ajout des nouveaux indicateur fournisseur clef et start-up

BDCFou-

App>src>main>java>fr>cnes>bdcfou>app>shared>verifier>fiche>generalite>FicheGener  
aliteDTOVerifier

Ligne 138 ajout des attributs

```
public static Map<String, List<String>> checkClef(
    List<String> listeValue, List<String> listeSelected) {
    return checkRadio("Fournisseur clef", listeValue, listeSelected);
}

public static Map<String, List<String>> checkStartUp(
    List<String> listeValue, List<String> listeSelected) {
    return checkRadio("StartUp", listeValue, listeSelected);
}
```

**Figure-04 DM-00027 :** ajout des nouveaux indicateur fournisseur clef et start-up

BDCFou-SI >src >main >java >fr >cnes >bdcfou >bo > Fournisseur.java

Ligne 34 ajout

```
/** Booleen indiquant si le fournisseur est spatial ou non. */
private boolean clef;
/** Booleen indiquant si le fournisseur est une start up ou non. */
private boolean startUp;
```

Ligne 156 ajout

```
public boolean isClef() {
    return clef;
}
public void setClef(boolean clef) {
    this.clef = clef;

}

/**
 * Getter indiquant si le fournisseur est une start up ou non.
 *
 * @return Booleen indiquant si le fournisseur est une start up ou non
 */
public boolean isStartUp() {
    return startUp;
}

/**
 * Setter indiquant si le fournisseur est une start up ou non.
 *
 * @param Start Up
 *          Booleen indiquant si le fournisseur est une start up ou non
 */
public void setStartUp(boolean startUp) {
    this.startUp = startUp;
}
```

**Figure-05 DM-00027 :** Modification de la DAO

```
BDCFou-SI >src >main >java >fr >cnes >bdcfou >dao >impl  
>rw>FournisseurDaoImpl.java  
public void update(Fournisseur object) throws BDCFouException {  
    StringBuilder sql = new StringBuilder();  
    sql.append("update fournisseur f ");  
    sql.append("set f.id_type_fournisseur = :typeFournisseur, ");  
    ...  
    sql.append("f.is_clef = :clef, ");  
    sql.append("f.is_startUp= :startUp, ");  
    ...  
    args.put("clef", object.isClef() ? 1 : 0);  
    args.put("startUp", object.isStartUp() ? 1 : 0);  
  
Modification du StringBuilder la requête : update fournisseur f f.is_clef = :clef  
f.is_clef -> on update dans la table fournisseur la colonne is_clef  
:clef -> on passe à cette colonne la valeur de clef qui est définie dans args.put  
args.put("clef", object.isClef() ? 1 : 0); -> dans le hashmap « args » je rentre  
une valeur pour « clef » où on va chercher le booléen, si object.isClef() on met la  
valeur 1 sinon (les « : ») je donne la valeur 0.
```

**Figure-06 DM-00027 :** Modification du RowMapper

```
BDCFou-SI >src >main >java >fr >cnes >bdcfou  
>dao>rowmpaper>FournisseurRowMapper  
Ligne 22 ajout  
    public Fournisseur mapRow(ResultSet rs, int rownum) throws SQLException  
ligne.49        fournisseur.setSpatial(rs.getInt("IS_CLEF") == 1);  
                fournisseur.setStartUp(rs.getInt("IS_STARTUP") == 1);  
  
Explication : En au resulset un valeur  
==1 va tester si l'expression est égale à 1. Si c'est égal renverra true, sinon renverra false.
```



## Connaissance des fournisseurs

DIFFUSION LIMITÉE CNES

**Généralités** Société Economiques Autres infos et Liens Etablissements

Raison sociale [REDACTED] Suivi  Oui  Non Dernière mise à jour le 04/02/2021 Supprimer ce fournisseur

Domaine spatial  Oui  Non Fournisseur clé  Oui  Non Start-up  Oui  Non

Siren [REDACTED] Identifiant TVA [REDACTED] Pays France Code pays FR Région du monde FR

Domaines d'activité M&G canismes - Satellite-PF\Structure

Principaux clients [REDACTED]

Principaux projets / programmes [REDACTED]

Présentation générale Organigramme Non fourni Site Web

**Contrats importants avec le CNES**

No	Objet	Montant	Durée	Type contractualisation	Resp. technique	Resp. DAJ
X	[REDACTED]	[REDACTED] €	43 m			
X	[REDACTED]	[REDACTED] €	19 m			
X	[REDACTED]	[REDACTED] €	19 m			

**Accords Cadres liés à des politiques d'achat**

Accord cadre	Lots
[REDACTED]	[REDACTED]

Figure-07 DM-00027 : Résultat de la modification dans l'IHM

## E. DM-00028

**Connaissance des fournisseurs** DIFFUSION LIMITÉE CNES

Représentation par le filtre fournisseur à tous les chiffres

Reportez-vous à DM-00027 pour plus d'informations

**Sélection des données à exporter**

- ID spécifique
- Raison sociale
- Nom courant
- N° Siret
- Identifiant TVA
- Nom du pays
- Région du monde
- Type de fournisseur
- Indicateur client
- Indicateur Suivi
- CA fournisseur titulaire 2020
- CA fournisseur titulaire 2019
- CA fournisseur titulaire 2018
- CA fournisseur titulaire 2017
- CA fournisseur titulaire 2016
- CA fournisseur titulaire 2015
- CA fournisseur titulaire 2014
- CA fournisseur titulaire 2013
- CA fournisseur titulaire 2012
- CA fournisseur titulaire 2011
- CA fournisseur payé 2020
- CA fournisseur payé 2019
- CA fournisseur payé 2018
- CA fournisseur payé 2017
- CA fournisseur payé 2016
- CA fournisseur payé 2015
- CA fournisseur payé 2014
- CA fournisseur payé 2013
- CA fournisseur payé 2012
- CA fournisseur payé 2011

**Connaissance des fournisseurs** DIFFUSION LIMITÉE CNES

Représentation par le filtre fournisseur à tous les chiffres

Reportez-vous à DM-00027 pour plus d'informations

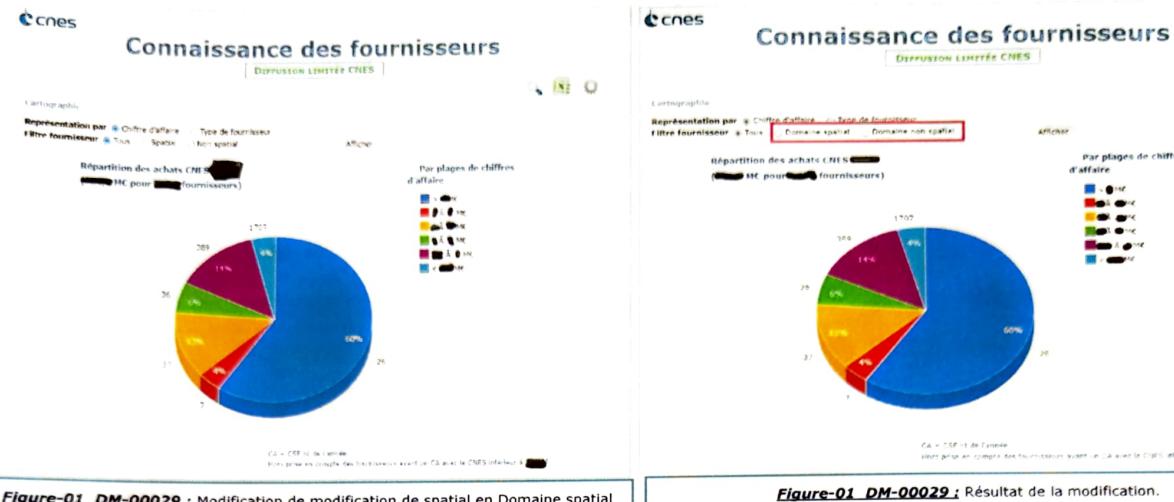
**Sélection des données à exporter**

- Identifiant TVA
- Nom du pays
- Région du monde
- Type de fournisseur
- Indicateur client
- Indicateur Suivi
- CA du titulaire
- Effectif du titulaire
- Fournisseur clé
- Indicateur client
- CA fournisseur titulaire 2020
- CA fournisseur titulaire 2019
- CA fournisseur titulaire 2018
- CA fournisseur titulaire 2017
- CA fournisseur titulaire 2016
- CA fournisseur titulaire 2015
- CA fournisseur titulaire 2014
- CA fournisseur titulaire 2013
- CA fournisseur titulaire 2012
- CA fournisseur titulaire 2011
- CA fournisseur payé 2020
- CA fournisseur payé 2019
- CA fournisseur payé 2018
- CA fournisseur payé 2017
- CA fournisseur payé 2016
- CA fournisseur payé 2015
- CA fournisseur payé 2014
- CA fournisseur payé 2013
- CA fournisseur payé 2012
- CA fournisseur payé 2011

Figure-01 DM-00028 : Ajout de 4 colonnes dans la sélection des données à exporter

Figure-02 DM-00028 : Résultat de l'ajout de 4 colonnes dans la sélection

## F. DM-00029



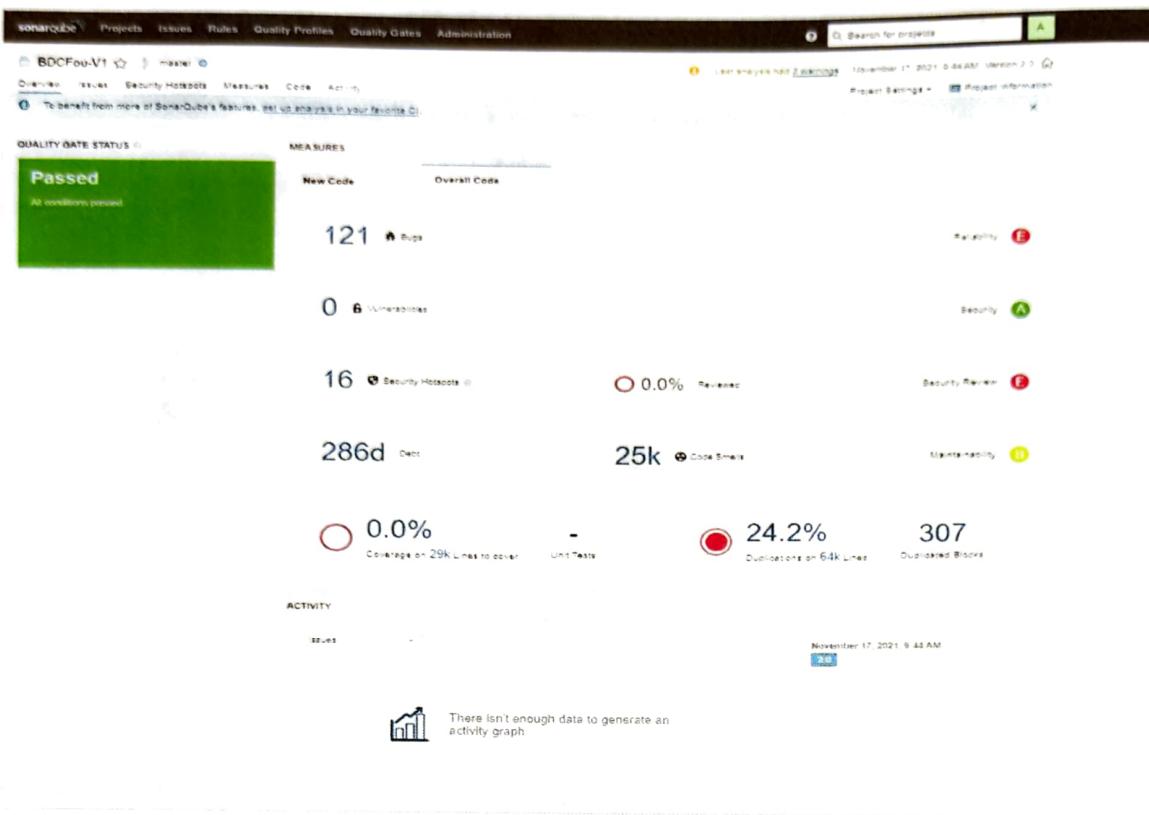
### 10.2. Test et validation

#### A. Cahier de recette

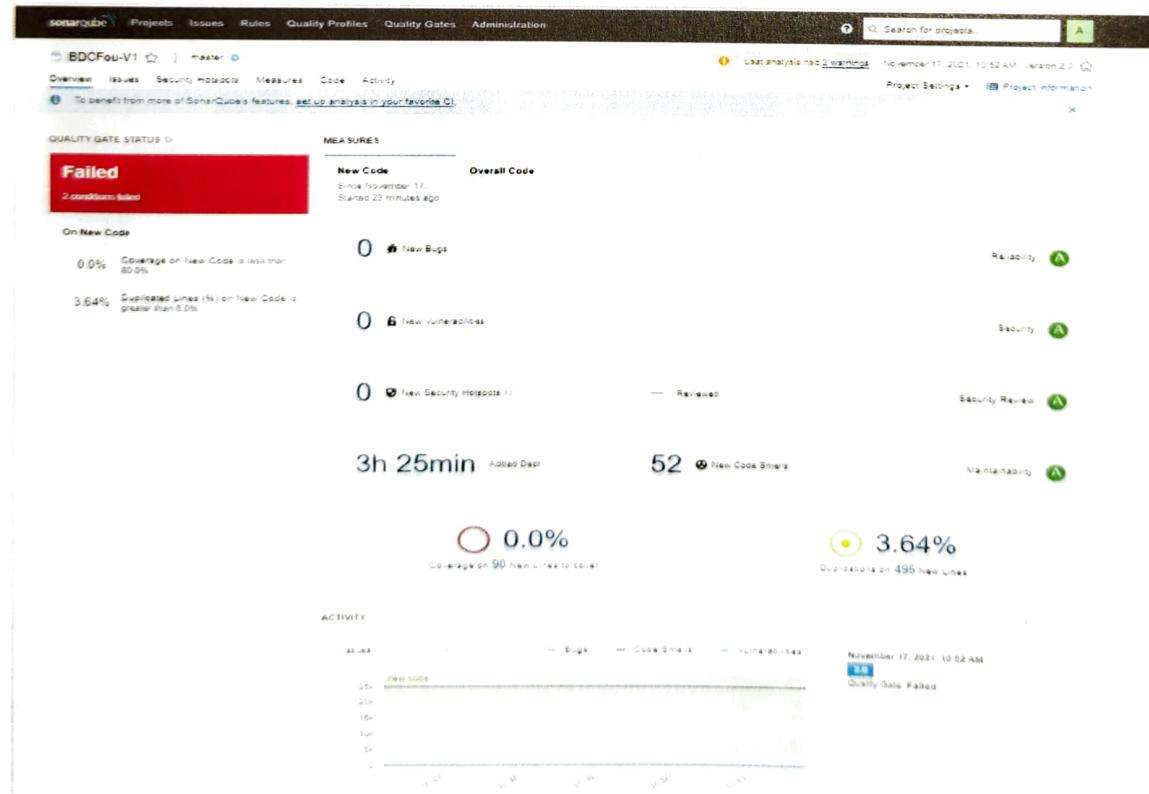
Numéro	Nom	Description	Référencement	Date exéc.	Commentaire
0001	Step 1	Se connecter sur la page d'accueil du site. Cliquier sur la loupe en haut à droite pour afficher la liste des fournisseurs.	La page d'accueil est affichée sans demande d'utilisateur ou de mot de passe à l'utilisateur.		
0002	Step 2	Cliquier sur le bouton Rechercher.	La page "Connaissance des fournisseurs" est affichée. La recherche est possible via les éléments suivants : - Nom - Région du monde - Domaine d'activités - Fournisseur suivi	Passed	• L_DR_BDCFou_MediatelTypeFournisseur_001-16 • L_DR_BDCFou_MediatelTypeFournisseur_002-16 • L_DR_BDCFou_MediatelTypeFournisseur_003-16 • L_DR_BDCFou_MediatelTypeFournisseur_004-16
0003	Step 3	Sélectionner "FR" pour la région. Cliquier sur le bouton Rechercher.	Le tableau contient une liste de fournisseurs de la région FR.	Passed	
0004	Step 4	Sélectionner "Oui" pour les fournisseurs suivis. Cliquier sur le bouton Rechercher.	Le tableau contient une liste de fournisseurs suivis de la région FR.	Passed	
0005	Step 5	Sélectionner "Tous" pour les fournisseurs suivis et de la région Toutes. Cliquier sur le bouton Rechercher.	Le tableau contient une liste de fournisseurs suivis et non suivis de la région FR.	Passed	• L_DR_BDCFou_MediatelTypeFournisseur_001-16
0006	Step 6	Sélectionner "Non" pour les fournisseurs suivis et de la région Toutes. Cliquier sur le bouton Rechercher.	Le tableau contient une liste de fournisseurs non suivis de la région Toutes.	Passed	• L_DR_BDCFou_MediatelTypeFournisseur_002-16
0007	Step 7	Sélectionner le premier fournisseur dans la liste des résultats de recherche.	La liste du fournisseur est affichée.		
... Accueil	Synthèse	Manipulation des données du bandeau. A l'aide des flèches de navigation du BDCFou_Fiche_Fournisseur_001	La liste du fournisseur suivant s'affiche à l'écran. L'ordre des fournisseurs est celui du BDCFou_Fiche_Fournisseur_002	BDCFou_Fiche_Fournisseur_003	BDCFou_Fiche_Fournisseur_004

**Figure-01 Test :** Capture d'écran cahier de recette.

## B. Qualité avec SonarQube



**Figure-02 Test:** Qualité du code version 2.0.0



**Figure-02 Test:** Qualité du code version 2.0.0

## 10.3. Images Base de données

### A. Dictionnaire des données

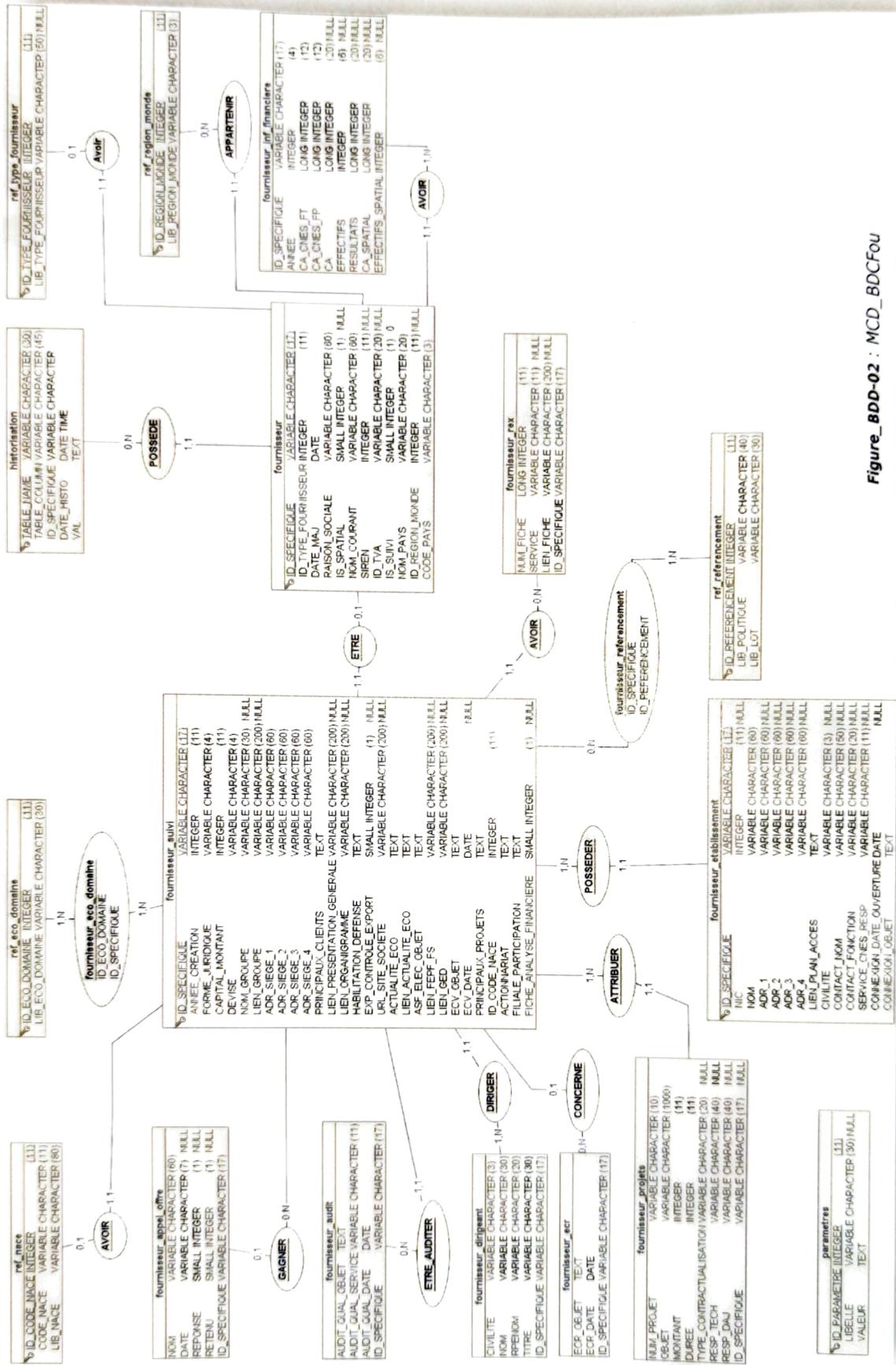
**inetum.** Positive Digital Now  
BARREH BOUNI Farhan  
Concepteur Développeur d'Application  
Session 2021

**Dictionnaire de données BDCFou**

NUM	NOM	DESCRIPTION	TYPE	IDENTIFIANT	MODE D'OBSTENTION	RÈGLE DE CALCUL	DOMAINE DE VALEURS	COMMENTAIRE
<b>FOURNISSEUR</b>								
1	ID_SPECIFIQUE	Clé primaire de la table fournisseur	ALPHANUM	X	Mémorisé			
	ID_TYPE_FOURNISSEUR	Type de fournisseur	NUM		Mémorisé			
	DATE_MAJ	Date de mise à jour	DATE		Mémorisé			
	RAISON_SOCIALE	Raison sociale	ALPHANUM		Mémorisé			
	IS_SPATIAL	Si c'est un fournisseur spatial ou pas	NUM		Mémorisé			
	SIREN	Numéro de SIREN du fournisseur	NUM		Mémorisé			
	ID_TVA	TVA	ALPHANUM		Mémorisé			
	IS_SUIVI	Si c'est un fournisseur suivi ou pas	NUM		Mémorisé			
	NOM_PAYS	Pays du fournisseur	ALPHANUM		Mémorisé			
	ID_REGION_MONDE	Région du monde du fournisseur	NUM		Mémorisé			
	CODE_PAYS	Code Pays	ALPHANUM		Mémorisé			
<b>fournisseur_appel_offre</b>								
2	NOM	Nom de l'Appel d'offre	ALPHANUM		Mémorisé			
	DATE	Date d'appel d'offre	ALPHANUM		Mémorisé			
	REPONSE	Reponse	NUM		Mémorisé			
	RETENU	Fournisseur retenu	NUM		Mémorisé			
	ID_SPECIFIQUE	Identifiant du fournisseur retenu	ALPHANUM		Mémorisé			
<b>fournisseur_audit</b>								
3	AUDIT_QUAL_OBJET		ALPHA		Mémorisé			
	AUDIT_QUAL_SERVICE		ALPHANUM		Mémorisé			
	AUDIT_QUAL_DATE		DATE		Mémorisé			
	ID_SPECIFIQUE	Fournisseur audité	ALPHANUM		Mémorisé			
<b>fournisseur_dirigeant</b>								
4	CIVILITE	Civilité	ALPHANUM		Mémorisé			
	NOM	Nom du Dirigeant	ALPHANUM		Mémorisé			
	FRENOM	Frename du Dirigeant	ALPHANUM		Mémorisé			
	TITRE	Titre du Dirigeant	ALPHANUM		Mémorisé			

**Figure\_BDD-01 : Partie\_Dictionnaire\_données\_BDCFou**

• Modèle conceptuel des données



Figure\_BDD-02 : MCD\_BDCFou

# Dossier technique de l'application BDCFou | BARREH BOUNI Farhan

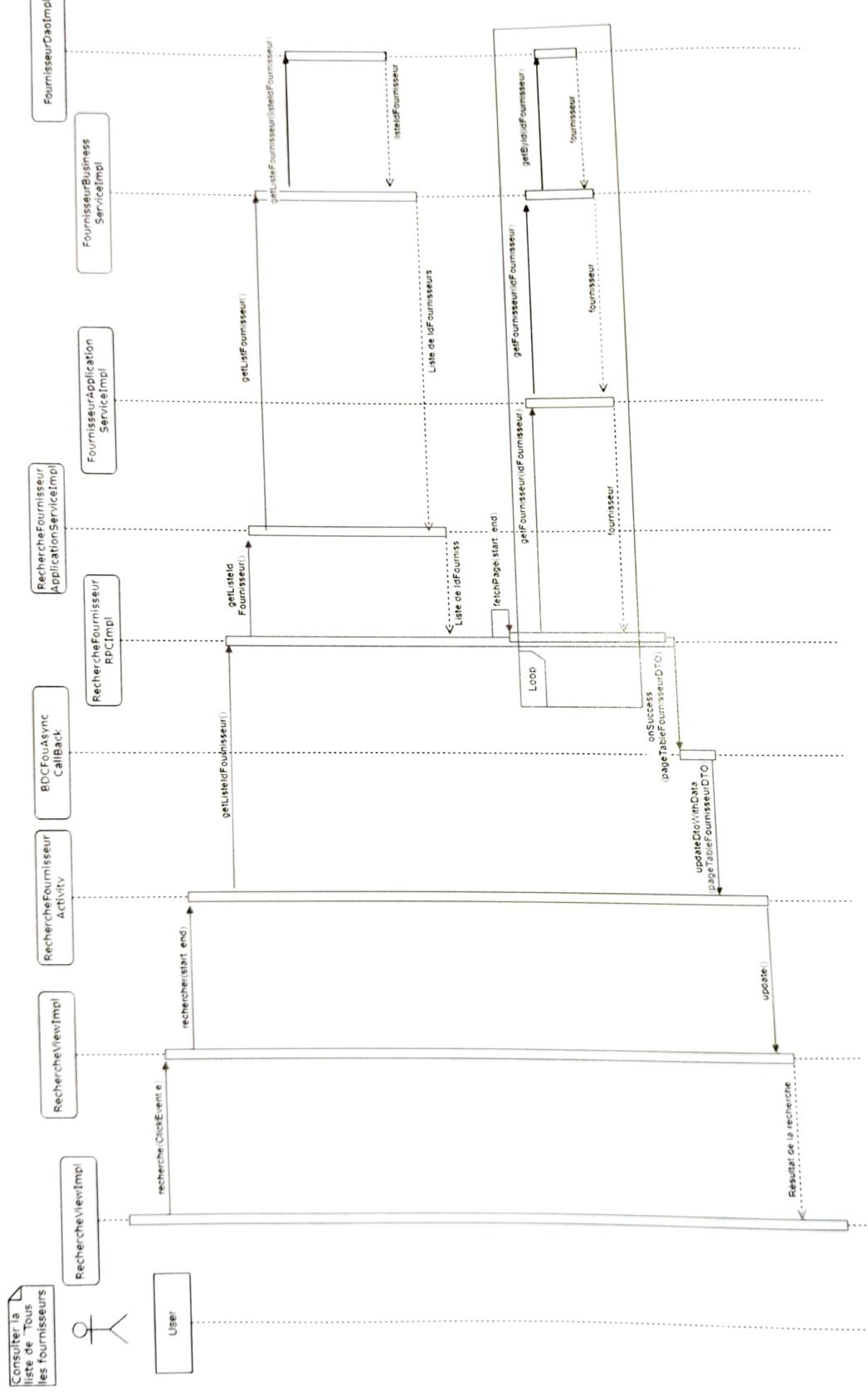
## • Modèle logique des données



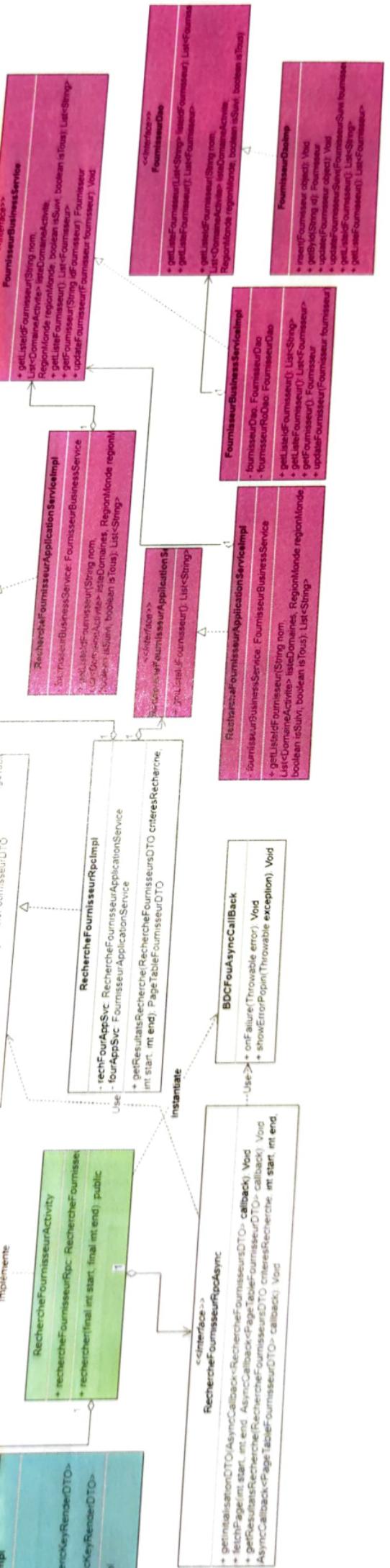
**Figure\_BDD-D3 : MLD\_BDCFou**

## 10.4. Conception

**Figure-01 Conception :** Diagramme de séquences de la fonction recherche



**Figure-01 Conception :** Diagramme de classes de la fonction recherche



### Diagramme de Packages BDCFOU

