



المدرسة العليا للعلوم التطبيقية والتصرف
ÉCOLE SUPÉRIEURE PRIVÉE DES SCIENCES
APPLIQUÉES ET DE MANAGEMENT

RAPPORT DE STAGE DE PROJET DE FIN D'ETUDES INGENIEURS EN INFORMATIQUE

Option : Data Science

Intitulé du stage
Implémentation d'un module d'indexation et dictionnaire de
données pour la solution 'Data Flow Management'

Réalisé par
Hamza Ksentini

Entreprise d'accueil



Talys Consulting

Encadrant Entreprise
Hamdi Ben Nosra

Encadrant SESAME
Amal Bouaziz

Année Universitaire
2022-2023

Remerciements

Je tiens, avant de présenter mon travail, à exprimer ma grande reconnaissance envers les personnes qui m'ont de près ou de loin apporté leurs soutiens. Qu'ils trouvent ici collectivement et individuellement l'expression de toute ma gratitude et ma reconnaissance.

J'exprime mes sincères reconnaissances à mon encadrant professionnel Monsieur **Hamdi Ben Nosra**, pour m'avoir accueilli parmi son équipe, et pour la confiance qu'il m'a témoignée. Je le remercie également pour ces précieux conseils et ses orientations qui m'ont permis de mener à bien ce projet.

Ma profonde gratitude s'adresse également à mon encadrante académique Madame **AMAL BOUAZIZ**, qu'elle n'a pas cessé de m'apporter tout son soutien, son bénéfique encadrement qui m'a permis de mener à bien le présent travail.

J'exprime aussi ma gratitude à toute l'équipe de **Talys** pour leur chaleureux accueil et aide. Un grand merci s'adresse à mes enseignants de L'Université **SESAME** pour leur apport scientifique qui m'a beaucoup aidé dans l'élaboration de ce travail et à tous ceux qui m'ont aidé et encouragé de près ou de loin pour l'achèvement et la réussite de ce travail.

Que les membres de jury trouvent, ici, l'expression de mes remerciements pour l'honneur qu'ils me font en prenant le temps de lire et d'évaluer ce travail.

Table des matières

INTRODUCTION G E N E R A L E	1
Chapitre 1	2
Présentation générale	2
Introduction	2
1.1 Cadre général	2
1.1.1 Présentation de l'organisme d'accueil	2
1.1.2 Talys	2
1.1.3 Organnigramme de Talys	4
1.2 Cadre du Projet	5
1.2.1 Contexte du Projet et problématique	5
1.2.2 Analyse et étude de l'existant	6
1.2.3 Présentation de quelques interfaces existantes dans le projetDFM	7
1.2.4 Description et Critique de l'existant	10
1.3 Méthodologie de travail	12
1.4 Langages de modélisation	14
Conclusion	15
Chapitre 2	18
Introduction	17
2.1 Capture des besoins globaux	17
2.1.1 Spécifications des Besoins Fonctionnels	17
2.1.2 Présentation des différents modules	18
2.1.1 Spécifications des Besoins non fonctionnels	18
2.2 Pilotage du Projet avec Scrum	19
2.2.1 Equipe et rôle	19
2.2.2 Backlog produit	19
2.2.3 Planification des sprints	21
2.3 Analyse globale	21
2.3.1 Diagramme de cas d'utilisation globale	21
2.4 Architecture logique globale de l'application	25
2.4.1 Architecture logique BACKEND et FRONTEND	25
2.4.2 Architecture logique -FRONTEND-	26
2.4.3 Architecture logique -BACKEND-	26
2.5 Étude des outils	27
Conclusion	31
Chapitre 3	32
Introduction	33
3.1 Spécification et analyse des besoins	33
3.1.1 Backlog du sprint 1	33
3.1.2 Diagramme raffiné du cas d'utilisation : Sprint 1	34
1) Diagramme de cas d'utilisation : Créer un référentiel métier	34
2) Raffinement des cas d'utilisation :	35
3) Diagramme raffiné du cas d'utilisation : Création du référentiel métier	35
3.1.3 Description textuelle des cas d'utilisations Sprint 2	36
3.1.1 Diagrammes de séquence sprint 2 :	37
3.2 Réalisation	39
3.2.1 Réalisation du cas d'utilisation Créer un referentiel métier	39
1) Menu d'utilisateur des settings	40
3.2.1 Liaison des champs de structure de données avec un/des référentiels métiers	41
Conclusion	43
Chapitre 4	44
Introduction	45

4.1	Spécification et analyse des besoins	45
4.1.1	Backlog du sprint 2 : enrichissement du dictionnaire de données	45
4.1.2	Les Diagrammes de cas d'utilisations : Sprint 2	46
1)	Diagramme de cas d'utilisation : Créer un modèle élastique « Mappings »	46
2)	Diagramme de cas d'utilisation : Chercher les données indexées	47
4.1.3	Description textuelle des cas d'utilisation : Sprint 2	47
1)	Description Textuelle de cas d'utilisation : Créer un modèle élastique « Mappings »	47
2)	Description Textuelle de cas d'utilisation : Chercher les données indexées	48
4.1.4	Les diagrammes de séquence de sprint 2 :	48
4.2	Réalisation	49
4.2.1	Réalisation du cas d'utilisation Créer modèle « Mappings »	50
1)	Interface des Flowcharts	50
2)	Mettre un mapping en Prod	50
3)	Visualisation sur Kibana :	51
4.2.2	Realisation de cas d'utilisation : rechercher les données indexées	52
1)	Champ de la recherche :	52
2)	L'interface du résultat de recherche :	52
	Conclusion	53
Chapitre 5		54
	Introduction	55
5.1	Spécification et analyse des besoins	55
5.1.1	Backlog du sprint 3 : Dashboarding des données et la préparation du module administratif	55
5.1.2	Les Diagrammes de cas d'utilisations : Sprint 2	56
1)	Diagramme de cas d'utilisation : Créer un Dashboard	56
5.1.3	Les Diagrammes de cas d'utilisations : Sprint 3	56
1)	Description Textuelle de cas d'utilisation : Créer un Dashboard	56
2)	Description Textuelle de cas d'utilisation : Supprimer un Dashboard	57
1)	Description Textuelle de cas d'utilisation : Consulter un Dashboard	57
		58
2)	Description Textuelle de cas d'utilisation : Modifier un Dashboard	58
5.1.4	Les diagrammes de séquence de sprint 3 :	58
1)	Diagramme de séquence de cas d'utilisation : Créer un Dashboard	58
		58
		58
2)	Diagramme de séquence de cas d'utilisation : Supprimer un Dashboard	59
		59
3)	Diagramme de séquence de cas d'utilisation : Consulter un Dashboard	59
		59
4)	Diagramme de séquence de cas d'utilisation : Modifier un Dashboard	59
		59
		60
5.2	Réalisation	60
5.2.1	Réalisation du Dashboard	60
1)	Interface de Dashboard	60
2)	Interface de Dashboard 2	61
3)	Interface de Creation d'une Visualisation	62
4)	Exemple de Création d'une Visualisation	63
	Conclusion	64
Conclusion générale et perspective :		65
	Conclusion générale	65
	Perspective	65
Webographie		66
		3
Tapez le titre du chapitre (niveau 1)		4
Tapez le titre du chapitre (niveau 2)		5
	Tapez le titre du chapitre (niveau 3)	6

Liste des Tableaux

TABLE 2.1 – Acteurs en interaction avec le système existant DFM	18
TABLE 2.2 – Acteurs en interaction avec les modules du projet	18
TABLE 2.3 – Backlog produit	20
TABLE 2.4 – Étude des outils	31
TABLE 3.1 – BACKLOG SPRINT 1	34
TABLE 3.2 – Description textuelle du cas d'utilisation "Création du référentiel métier"	36
TABLE 3.3 – Description textuelle du cas d'utilisation "Création du dictionnaire de données"	37
TABLE 4.1 – BACKLOG SPRINT 2	45
TABLE 4.2 – Description textuelle du cas d'utilisation " créer un modèle élastique « Mappings »"	47
TABLE 4.3 – Description textuelle du cas d'utilisation " Chercher les données indexées "	48
TABLE 5.1 – BACKLOG SPRINT 3	55
TABLE 5.2 – Description textuelle du cas d'utilisation " Créer un Dashboard "	56
TABLE 5.3 – Description textuelle du cas d'utilisation " Supprimer un Dashboard "	57
TABLE 5.4 – Description textuelle du cas d'utilisation " Consulter un Dashboard"	57
TABLE 5.5 – Description textuelle du cas d'utilisation " Modifier un Dashboard "	58

Table des Figures

Figure 1.1 – Logo Talys.	2
Figure 1.2 – Partenaires de Talys.	3
Figure 1.3 – Organnigramme de Talys.....	4
Figure 1.4 – Interface d’authentification.....	7
Figure 1.5 – Interface de dashboarding existant	8
Figure 1.6 – Interface des projet	8
figure 1.7 – Interface de Système de données existant	9
Figure 1.8 – Interface de Menu administratif.....	10
Figure 1.9 – Les rôles Scrum.....	Erreur ! Signet non défini.
Figure 1.10 – Le cadre de travail SCRUM.....	14
Figure 2.1 – User Figure 2.2 – Admin Figure 2.3 – Système	17
Figure 2.4 – Equipe Scrum	19
Figure 2.5 – Structurations des sprints.....	21
Figure 2.6 – Diagramme de cas d’utilisation globale de projet.....	22
Figure 2.7 – Diagramme de classes d’analyse global de DFM.....	24
Figure 2.8 – Diagramme de classes d’analyse global de notre projet	25
Figure 2.9 – Architecture logique BACKEND et FRONTEND	25
Figure 2.10 – Architecture logicielle -FRONTEND	26
Figure 2.11 – Architecture logicielle -FRONTEND	26
Figure 3.1 – Diagramme de cas d’utilisation création d’un referentiel métier	34
Figure 3.2 – Cas d’utilisation "Création d’un référentiel métier "	35
Figure 3.3 – Cas d’utilisation "Création du dictionnaire des données "	35
Figure 3.4 – Diagramme de sequence de creation d’un nouveau référentiel métier.....	37
Figure 3.5 – Diagramme de séquence de consultation des référentiels métiers.	38
Figure 3.6 – Diagramme de séquence de modification des référentiels métiers.....	38
Figure 3.7 – Diagramme de séquence de la suppression des référentiels métiers.	39
Figure 3.8 – Diagramme de séquence de la création du dictionnaire de données.....	39
Figure 3.9 – Page de menu settings	40
Figure 3.10 – Interface de création du référentiel métier	40
Figure 3.11 – Liste des structures de données	41
Figure 3.12 – Liaison des champs de structure de données avec un/des référentiels métiers	42
Figure 3.13 – Visualisation des objets élastique	42
Figure 4.1 – Diagramme de cas d’utilisation créer un modèle elastique « Mappings »	46
Figure 4.2 – Diagramme de cas d’utilisation Chercher les données indexées.....	47
Figure 4.3 – Diagramme de cas d’utilisation création du modèle elastique « Mappings »	49
Figure 4.5 – Interface des flowcharts	50
Figure 4.6 – Interface de la liste de mappings.....	51
Figure 4.7 – Interface de la liste de mappings 2.....	51
Figure 4.8 – Interface des listes de « Mappings » créés dans Kibana.....	52
Figure 4.9 – Diagramme de cas d’utilisation Chercher les données indexées.....	52
Figure 4.10 – Interface de la Résultats de la recherche.....	53
Figure 5.1 – Diagramme de cas d’utilisation Créer un Dashboard.....	56
Figure 5.2 – Diagramme de séquence de cas d’utilisation Créer un Dashboard	58
Figure 5.3 – Diagramme de séquence de cas d’utilisation Supprimer un Dashboard	59
Figure 5.4 – Diagramme de séquence de cas d’utilisation Consulter un Dashboard	59
Figure 5.5 – Diagramme de séquence de cas d’utilisation Modifier un Dashboard	60
Figure 5.6 – Interface du Dashboard principale 1	61
Figure 5.7 – Interface du Dashboard principale 2	62
Figure 5.8 – Interface de la création d’une nouvelle visualisation.....	63
Figure 5.9 – Exemple de visualisation des données	64

INTRODUCTION GENERALE

Les données sont au cœur de la plupart des entreprises modernes. Elles sont utilisées pour prendre des décisions stratégiques, analyser les tendances du marché, évaluer les performances de l'entreprise, et bien plus encore. Cependant, avec l'augmentation constante du volume de données générées chaque jour, il est devenu difficile de les gérer efficacement. C'est là que le dictionnaire de données entre en jeu.

Le dictionnaire de données est un outil essentiel pour toute entreprise qui cherche à gérer efficacement ses données. Il s'agit d'un référentiel centralisé qui stocke des informations sur toutes les données utilisées dans l'entreprise, y compris les définitions, les sources, les formats, les relations et les règles d'utilisation. En utilisant un dictionnaire de données, les entreprises peuvent s'assurer que les données sont cohérentes, précises et faciles à comprendre pour toutes les parties prenantes.

Dans ce projet de fin d'études, nous allons examiner de plus près l'importance du dictionnaire de données pour les entreprises. Nous allons explorer les avantages qu'il peut offrir en termes de gestion des données, de collaboration interdépartementale, de prise de décision éclairée et d'efficacité opérationnelle. Nous allons également étudier les meilleures pratiques pour la mise en place d'un dictionnaire de données efficace, ainsi que les outils et technologies disponibles pour le soutenir.

Dans le cadre de ce projet, nous allons mettre en œuvre une solution basée sur Elastic Search pour l'indexation et la création du dictionnaire de données. Elastic Search est un moteur de recherche et d'analyse distribué qui offre des fonctionnalités avancées pour la recherche et la récupération de données. Nous utiliserons également Kibana, une plateforme de visualisation de données, pour intégrer un tableau de bord interactif et convivial.

En fin de compte, ce projet vise à démontrer l'importance du dictionnaire de données pour les entreprises modernes et à fournir des conseils pratiques pour aider les entreprises à mettre en place un dictionnaire de données efficace et durable, en utilisant les technologies d'indexation et de visualisation appropriées.

Ce rapport comporte cinq (5) chapitres organisés comme suit :

Dans le premier chapitre, nous présenterons le cadre général de notre application, l'étude et la critique de l'existant, la solution proposée, la méthodologie adoptée ainsi que le langage de modélisation. Le deuxième chapitre, s'intéressera à l'analyse de besoins fonctionnels et non fonctionnels, la planification des sprints et le cas d'utilisation globale de notre solution ainsi que l'architecture globale et l'environnement de travail. Le troisième chapitre comportera le premier sprint de notre solution en se basant sur la méthodologie Scrum. Nous consacrons le quatrième chapitre à la conception et la réalisation de la deuxième Sprint. Dans le chapitre cinq, nous abordons le sujet du troisième sprint de notre solution, qui se concentre sur le développement d'un tableau de bord (dashboarding). Finalement nous clôturons notre rapport par une conclusion générale qui présentera une récapitulation du travail réalisé et les éventuelles perspectives.

Chapitre 1

Présentation générale

Introduction

Ce premier chapitre expose le cadre général et le contexte de travail du projet, ainsi que les objectifs à atteindre. Nous présentons dans un premier temps Talys, l'organisme d'accueil, sa position, ses activités et services, puis nous expliciterons le projet et son contexte.

1.1 Cadre général

Le sujet introduit par le présent rapport est intitulé "Implémentation d'un module d'indexation et dictionnaire de données pour la solution 'Data Flow Management'. Il a été élaboré dans le cadre de projet de fin d'études en vue d'obtenir le diplôme National d'ingénieur en Informatique réalisé au sein de l'entreprise Talys.

1.1.1 Présentation de l'organisme d'accueil

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

1.1.2 Talys

TALYS est un groupe d'entreprises technologique spécialisé en Organisation, Systèmes d'Information et Transformation Digitale.

Depuis 2006, TALYS a eu la volonté de créer de la valeur ajoutée chez les clients en les accompagnant dans la concrétisation et réussite de leurs projets innovants.



Figure 1.1 – Logo Talys.

Secteurs d'activité : Talys cible une grande marge des marchés :

- Banques
- Assurance
- Microfinance
- Leasing
- Industrie et Services

Les Services offerts :

Solutions :

- Solutions MEGA on HOPEX
- Solutions Digitales Microfinance

Expertise :

- Data Intelligence
- Outsourcing et développement applicatif

Partenaires : Afin de disposer d'un savoir-faire et de compétences d'ordre international et surtout d'actualité en termes d'outils de développement et d'exploitation, Talys collabore avec des partenaires internationaux de renom dont notamment Mega, Fern Software et AMEF Consulting.

- **Mega :** TALYS est représentant officiel de la solution MEGA en Tunisie, MEGA est un éditeur de solutions logicielles d'Architecture d'Entreprise, Gestion du Portefeuille Applicatif Gouvernance et Risques et Conformité.
- **Fern Software :** TALYS est représentant exclusif de la solution ABACUS en Tunisie et Afrique francophone. Cette solution est éditée par FernSoftware : un leader reconnu dans la fourniture de services bancaires pour la microfinance, les coopératives de crédit ainsi que les banques rurales et institutions de développement.
- **AMEF Consulting :** AMEF Consulting est un cabinet de conseil en stratégie et management, spécialisé dans le secteur de la finance, ayant une dimension régionale couvrant l'Afrique.



Figure 1.2 – Partenaires de Talys.

1.1.3 Organnigramme de Talys

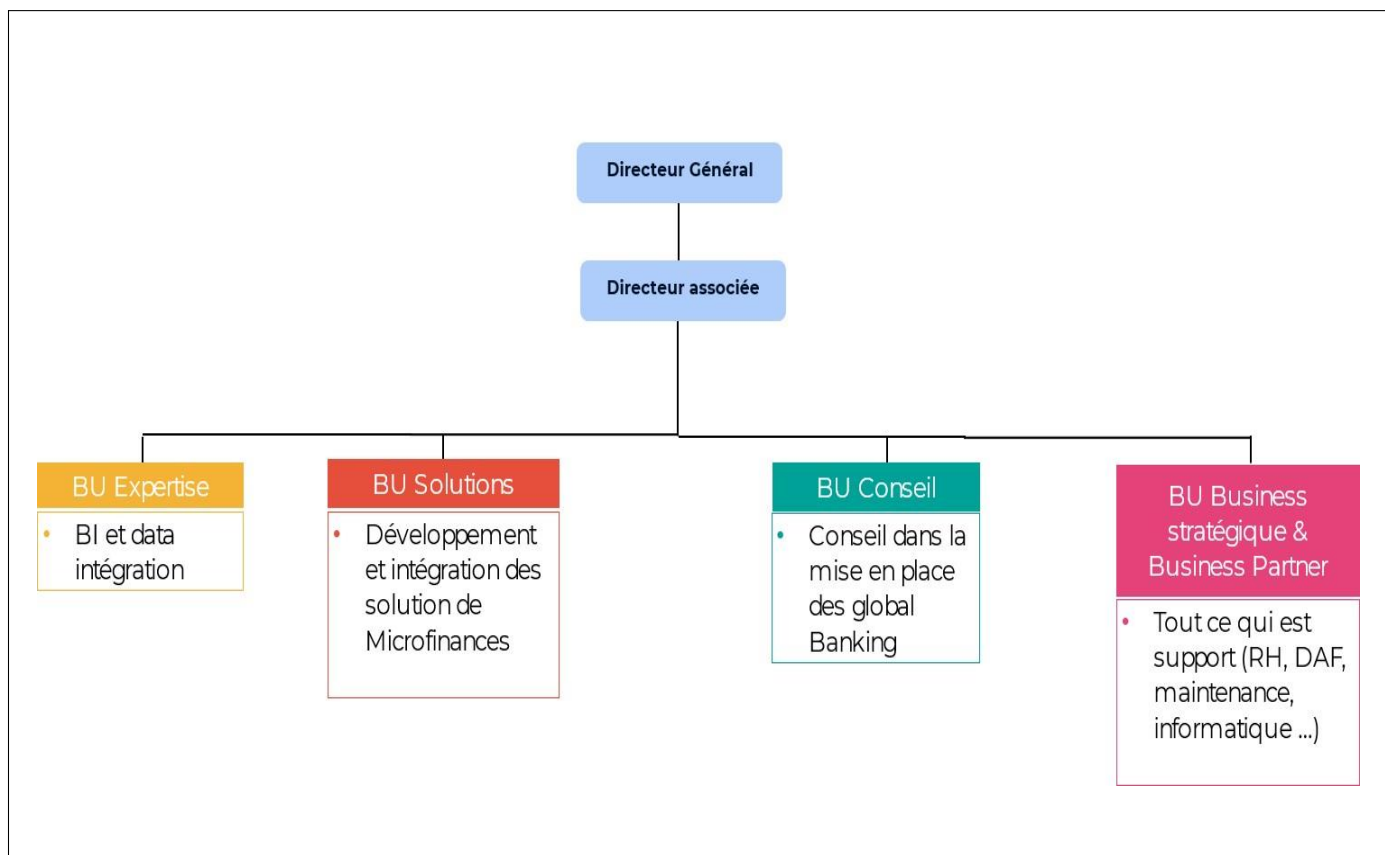


Figure 1.3 – Organnigramme de Talys

1.2 Cadre du Projet

1.2.1 Contexte du Projet et problématique

Le présent projet intitulé « **Implémentation d'un module d'indexation et dictionnaire de données pour la solution 'Data Flow Management'** » est réalisé dans le cadre de projet de fin d'études pour l'obtention du diplôme nationale d'ingénieur en informatique au sein de l'École Supérieure des Sciences Appliquées et de Management - SESAME.

Le travail consiste à mettre en pratique les notions théoriques dont nous avons acquis durant le parcours académique. Pour la réalisation de ce projet, j'ai effectué mon stage au sein de la société Talys pour une durée de 5 mois.

Dans ce cadre la société Talys propose un projet : Implémentation d'un module d'indexation et dictionnaire de données pour la solution 'Data Flow Management', qui a pour objectif :

1. Mise en place d'un module 'Dictionnaire de données' : capable de tracer les relations (liaisons) de chaque attribut du SI en se basant sur les flux de données schématisés dans l'application.
2. Produire un modèle de données synthétisé via un outil d'indexation tel que 'Elasticsearch' (choix à challenger par vos propositions)
3. En option compléter le stack indexé par un outil de reporting adéquat pour avoir des reporting dynamique (typiquement 'Kibana' si le choix d'indexation serait 'Elasticsearch')

Problématique

En effet, les obstacles et les problèmes que nous avons confrontés avec les anciennes méthodes de gestion des flux de données sont les suivants : les processus manuels, la complexité des relations entre les attributs du système d'information et la difficulté à évaluer la faisabilité et les écarts lors de la mise en place d'un nouveau système ou d'une nouvelle structure de données.

Imaginez maintenant travailler avec un module d'indexation et un dictionnaire de données efficaces, capables de tracer les relations entre les attributs du SI et de produire un modèle de données synthétisé en utilisant un outil d'indexation tel que "Elasticsearch". Les données indexées permettent de mettre en place un algorithme d'intelligence artificielle capable de calculer la faisabilité et les écarts par rapport aux nouveaux systèmes ou structures de données. De plus, l'ajout d'un outil de reporting adéquat tel que "Kibana" pourrait permettre d'obtenir des rapports dynamiques

Dans ce contexte, la problématique centrale du projet de PFE est la suivante : Comment concevoir et implémenter efficacement un module d'indexation et un dictionnaire de données pour la solution "Data Flow Management", afin de faciliter la gestion des relations entre les attributs du SI, d'optimiser l'évaluation de la faisabilité et des écarts lors de l'introduction de nouveaux systèmes ou structures de données, et d'offrir des rapports dynamiques pour une visualisation claire et pertinente des résultats ?

1.2.2 Analyse et étude de l'existant

La solution DFM (Data Flow Management) actuellement utilisée en version Beta Testing par les clients de la société "Talys Group" est une application qui se base sur les modules liés à la gestion des flux de données, notamment leur historisation, schématisation, et rédaction de toutes transformations ou traitements effectués. Ainsi que la génération automatisée d'un système de ticketing lié à ces traitements effectués.

Pour mieux comprendre l'application DFM existante et afin de pouvoir mieux l'analyser, nous allons présenter une brève description de chaque module de l'application :

- Gestion des utilisateurs + gestion des rôles : englobe les opérations et les éléments de la gestion des profils, hiérarchies profils et groupes d'utilisateurs ainsi que la gestion de leurs rôles et droits d'accès par rapport à chaque IHM
- Gestion des connexions : permet aux utilisateurs de se connecter et de se déconnecter de l'application en toute sécurité. Cette fonctionnalité prend également en charge la récupération de mots de passe et la réinitialisation des comptes utilisateur en cas d'oubli de mot de passe.
- Gestion des flux de données : cette fonctionnalité assure la gestion et le suivi des flux de données entre les différents systèmes de l'entreprise. Elle permet de suivre les transformations effectuées sur les données ainsi que leur traçabilité tout au long de leur parcours.
- Automatisation des tâches : permet de générer des tâches automatiquement dans un tableau de bord et de suivre leur progression. Cette fonctionnalité aide à assurer une exécution efficace et en temps opportun des tâches assignées.
- Cartographie des flux de données : permet de générer automatiquement une cartographie des flux de données entre les différents systèmes de l'entreprise. Cette cartographie aide à visualiser l'infrastructure et les séquences de processus de ces flux.
- Gestion des mappings : assure la gestion des mappings entre les différents systèmes de l'entreprise. Cette fonctionnalité permet de suivre les transformations effectuées sur les données ainsi que leur traçabilité tout au long de leur parcours

En combinant tous ces modules, l'application DFM Data Flow Management offre une solution complète pour la gestion et la traçabilité des flux de données au sein de l'entreprise.

1.2.3 Présentation de quelques interfaces existantes dans le projetDFM

L'interface d'authentification

Permettre à l'utilisateur de s'authentifier et déterminer le rôle et l'interface à ouvrir.

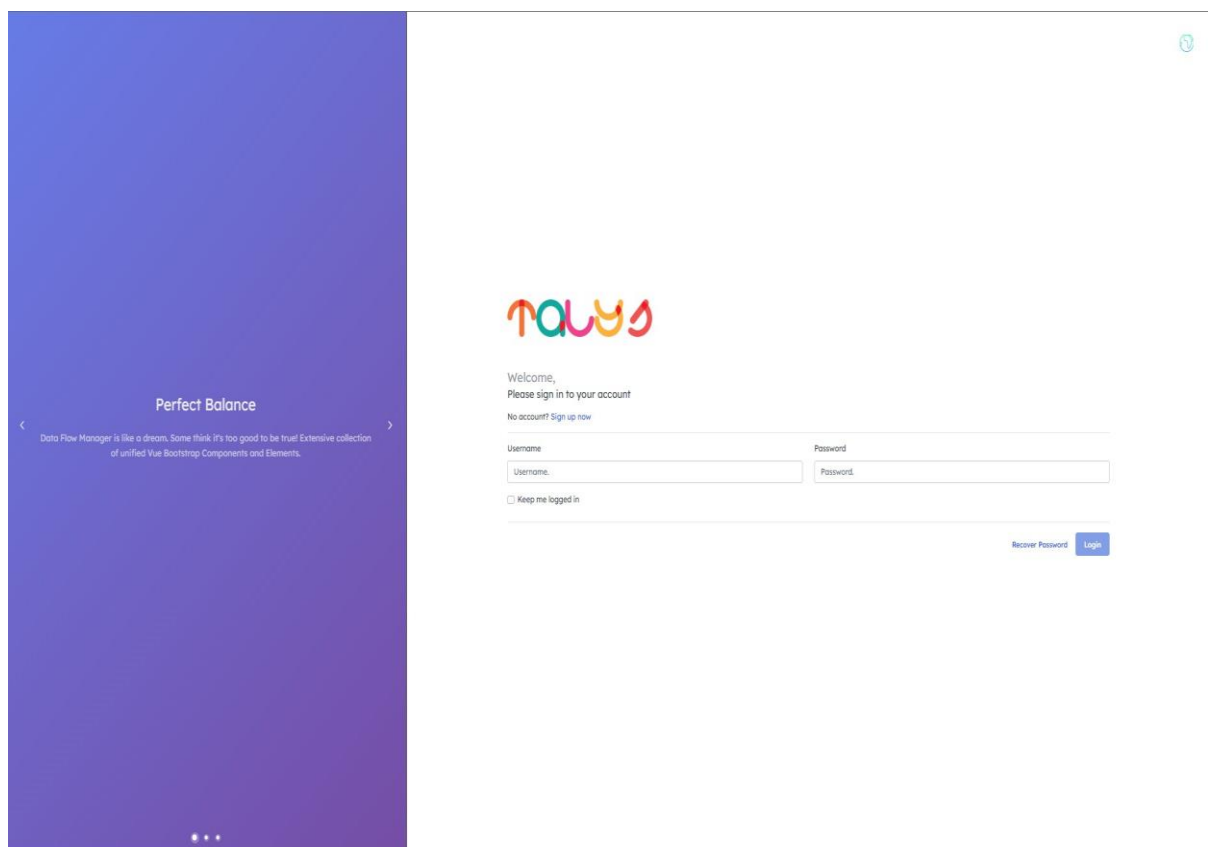


Figure 1.4 – Interface d'authentification

L'interface de dashboarding

Permettre à l'utilisateur de visualiser les données sous forme d'un dashboard. Cette interface subira des changements au cours de notre projet, et l'interface existante est simplement utilisée pour montrer à nos clients :

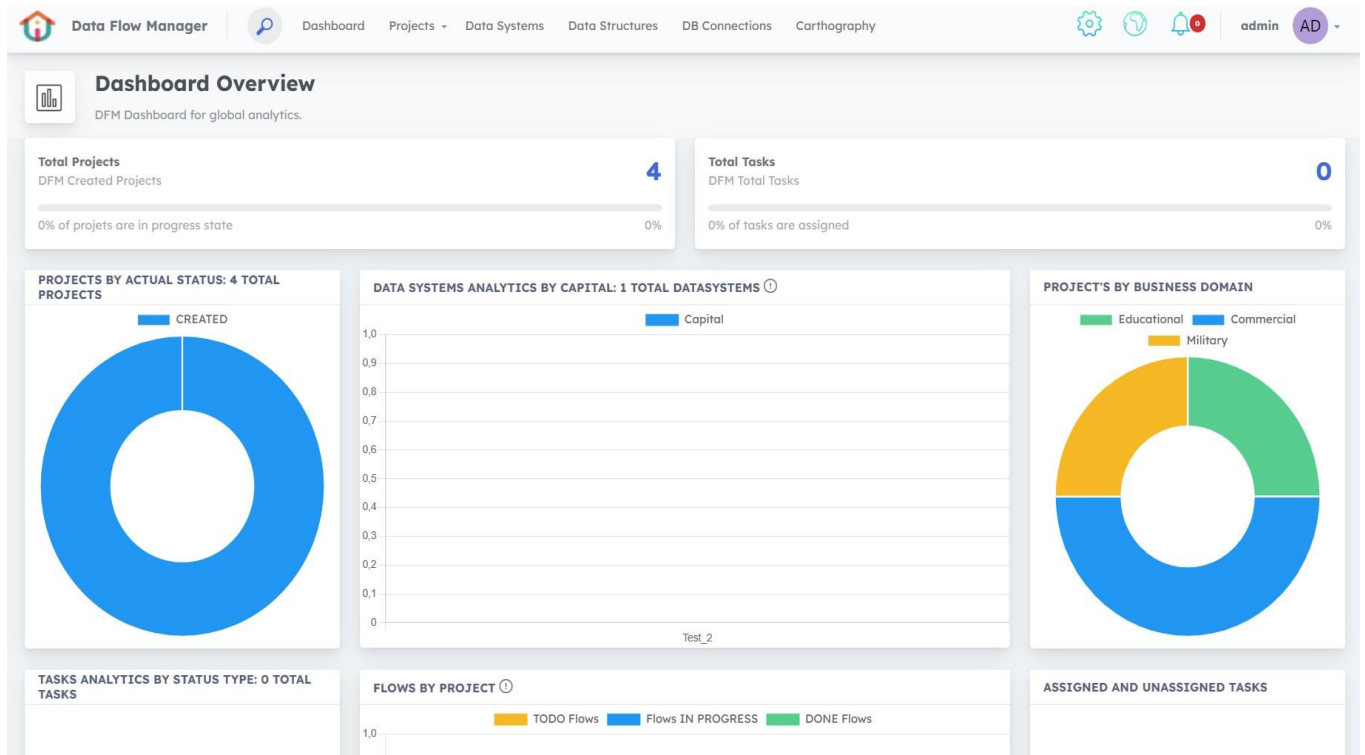


Figure 1.5 – Interface de dashboarding existant

L'interface des Projets

Permettre à l'utilisateur de gérer des projets liés à lui, incluant la création, la modification et la suppression :

Projects' List
The list of projects corresponding to a data flow between a client and a provider.

Refresh New

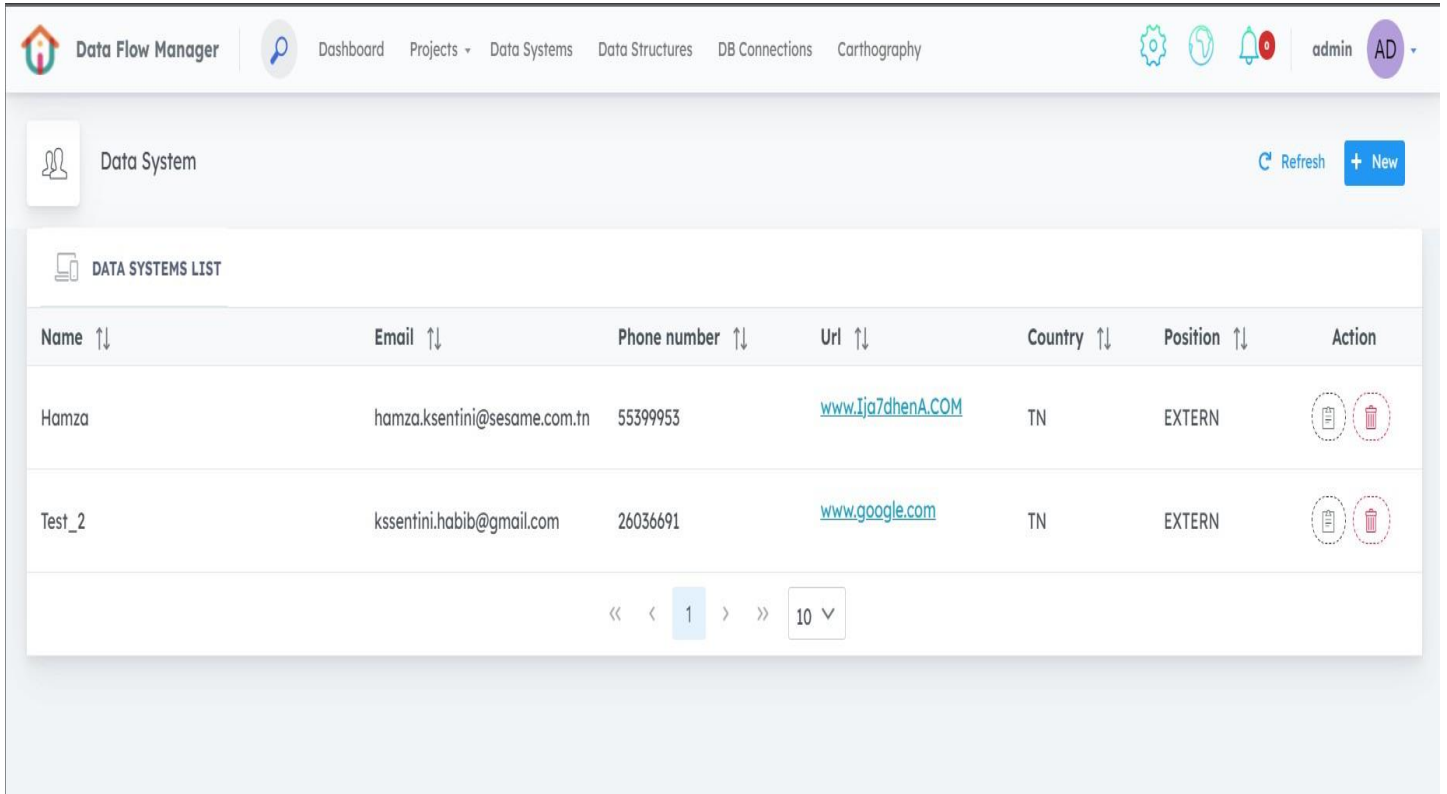
Name	Manager	Business Domain	Budget	Created By	Created On	Actions
War Crimes	Admin	Military	1000000	admin	15/05/2023 16:14:01	
Funding	Admin	Educational	500000	admin	15/05/2023 16:11:41	
Projet Poste	Admin	Commercial	21000	admin	15/05/2023 16:06:44	
Project X	Admin	Commercial	1000	admin	15/05/2023 16:02:19	

<< < 1 > >> 5

Figure 1.6 – Interface des projet

L'interface de de Système de données

Permettre à l'utilisateur de gérer les système de données liés à lui, incluant la création, la modification et la suppression :



The screenshot displays the 'Data Flow Manager' application interface. The top navigation bar includes a home icon, the title 'Data Flow Manager', a search icon, and several menu items: 'Dashboard', 'Projects', 'Data Systems', 'Data Structures', 'DB Connections', and 'Cartography'. On the right side of the navigation bar, there are icons for settings, a globe, a notification bell with a red dot, and a user profile labeled 'admin' with a dropdown arrow.

Below the navigation bar, the main content area is titled 'Data System' with a user icon. To the right of this title are 'Refresh' and '+ New' buttons. The central part of the interface features a 'DATA SYSTEMS LIST' table. The table has seven columns: 'Name', 'Email', 'Phone number', 'Url', 'Country', 'Position', and 'Action'. There are two data rows: one for 'Hamza' and another for 'Test_2'. Each row includes a 'Data System' icon in the 'Action' column. At the bottom of the table, there is a pagination control showing '1' of 10 items.





Name ↑↓	Email ↑↓	Phone number ↑↓	Url ↑↓	Country ↑↓	Position ↑↓	Action
Hamza	hamza.ksentini@sesame.com.tn	55399953	www.tja7dhenA.COM	TN	EXTERN	 
Test_2	kssentini.habib@gmail.com	26036691	www.google.com	TN	EXTERN	 

figure 1.7 – Interface de Système de données existant

L'interface de menu administratif

Permet à l'utilisateur de gérer directement les différents modules de l'application DFM. Cette capture a été prise après l'ajout du module 'référentiel métier' au menu. :

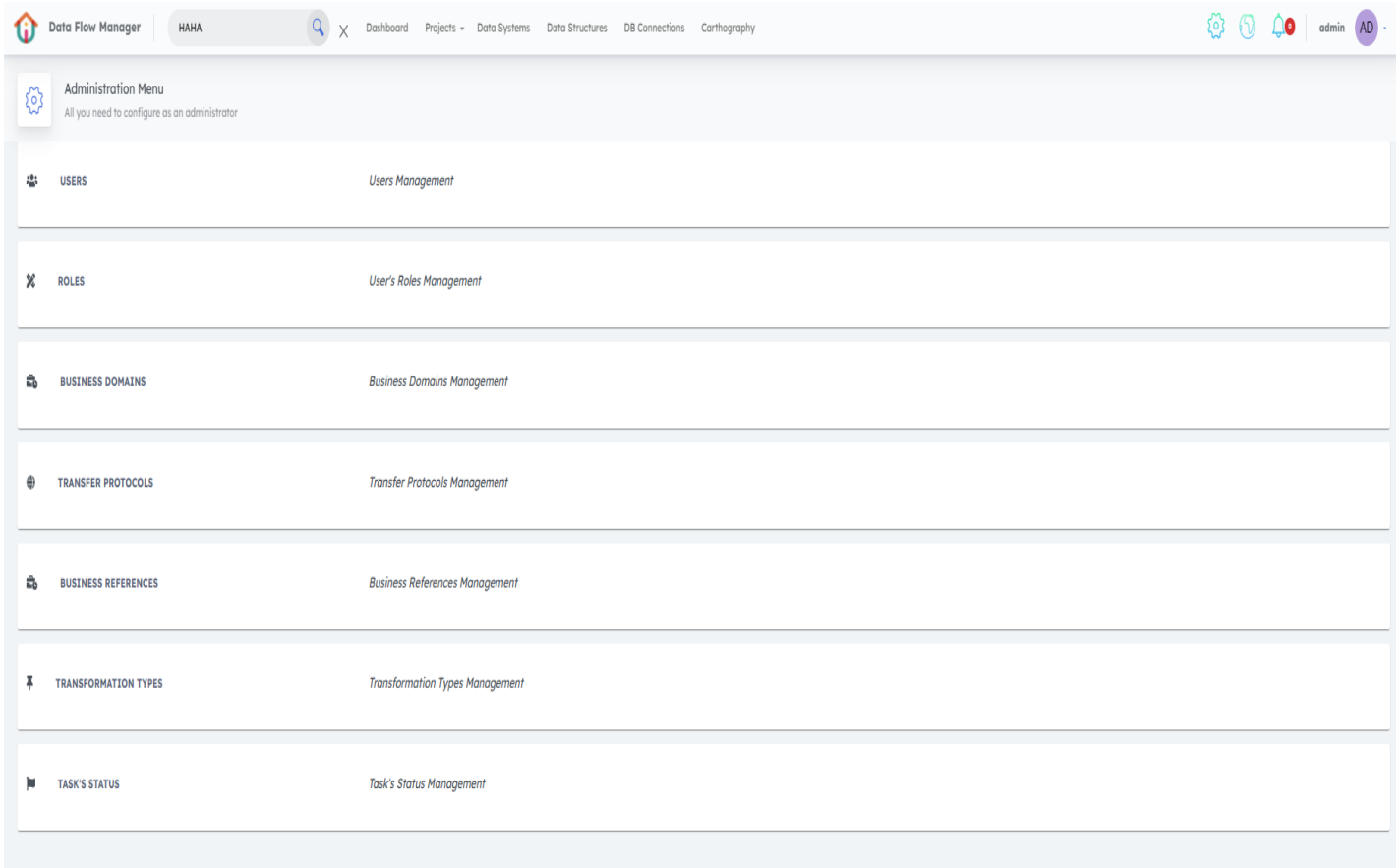


Figure 1.8 – Interface de Menu administratif

1.2.4 Description et Critique de l'existant

Bien que l'application DFM Data Flow Management soit une solution complète pour la gestion et la traçabilité des flux de données au sein de l'entreprise, son absence de module de dictionnaire de données peut entraîner des erreurs de synchronisation et une perte de temps pour les utilisateurs de l'application. De plus, la recherche d'informations précises peut être difficile en raison de la diversité des nomenclatures utilisées dans les différents systèmes de l'entreprise.

C'est pourquoi le développement d'un module de dictionnaire de données pour l'application DFM est une étape importante pour améliorer la qualité et l'efficacité de la gestion des flux de données au sein de l'entreprise. Ce module permettra de standardiser les informations en utilisant un référentiel métier, de faciliter la synchronisation des profils métiers et techniques, et de faciliter la recherche d'informations précises grâce à l'indexation et à la recherche par barre de recherche avec Elasticsearch.

Inconvénients :

- Absence d'un module de dictionnaire de données pour tracer les relations et les liens
- entre les différents attributs des systèmes de l'entreprise.

- Manque de synchronisation entre les profils métiers et les profils techniques, ce qui peut entraîner des erreurs et des conflits lors de la gestion des données.
- Difficulté à identifier les informations qui sont présentées sous des nomenclatures différentes mais qui représentent en réalité la même information.
- Indexation des données limitée, ce qui rend la recherche et l'exploration des données plus difficile.
- Absence d'un référentiel métier pour standardiser les informations.
- Absence d'un système de dashboarding pour présenter les données de manière claire et efficace.

Solution proposée :

Notre solution proposée pour DFM consiste en l'ajout d'un module de dictionnaire de données capable de tracer les relations et les liaisons de chaque attribut système de notre existant au sein de notre application. L'objectif de ce module est la synchronisation des profils métiers et profils techniques, afin d'assurer une standardisation des données et une meilleure traçabilité de leur parcours.

Nous avons également ajouté un référentiel métier, qui permet de faire la standardisation des informations et d'assurer la cohérence des données dans l'ensemble des systèmes de l'entreprise. Ce référentiel facilite également la maintenance et la mise à jour des données.

Enfin, nous avons mis en place un système de dashboarding, qui permet de présenter les données de manière claire et intuitive. Ce système facilite la compréhension et l'analyse des flux de données, et permet de prendre des décisions éclairées en temps réel.

Avantages :

— Meilleure traçabilité des données : grâce au module de dictionnaire de données et à la gestion des liens entre les attributs des différents systèmes, la solution proposée permet une meilleure traçabilité des données. Les utilisateurs peuvent ainsi retracer les transformations effectuées sur les données et suivre leur parcours tout au long du processus.

— Standardisation des données : la mise en place d'un référentiel métier permet une standardisation des données et une uniformisation des nomenclatures utilisées pour désigner les mêmes informations. Cela facilite la communication entre les différents départements et assure une meilleure qualité des données.

- Recherche plus efficace : l'indexation des données dans Elasticsearch permet une recherche rapide et efficace des informations. Les utilisateurs peuvent ainsi retrouver rapidement les données dont ils ont besoin et gagner du temps dans leur travail quotidien.
- Dashboarding : la solution proposée inclut un système de dashboarding pour la présentation des données. Cela permet aux utilisateurs de visualiser les données sous forme de graphiques et de tableaux de bord personnalisés, facilitant ainsi la prise de décision et la planification.
- Amélioration de l'efficacité : la mise en place d'un système automatisé pour la génération des tâches permet une exécution rapide et efficace des tâches assignées, ce qui améliore l'efficacité de l'ensemble du processus.
- Sécurité renforcée : la gestion des connexions assure une sécurité renforcée pour les utilisateurs de l'application. La récupération de mots de passe et la réinitialisation des comptes utilisateur en cas d'oubli de mot de passe garantissent un accès sécurisé à l'application.

1.3 Méthodologie de travail

Pour garantir un niveau de qualité optimale, pour la performance de notre solution et pour satisfaire les exigences du projet et éviter tout débordement au niveau des délais, l'adoption d'une méthodologie de développement est une nécessité. Nous avons adopté la méthode agile tout au long de la réalisation de notre projet. C'est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif. Le terme "agile" définit une approche de gestion de projet qui prend le contre-pied des approches traditionnelles prédictives et séquentielles de type cycle en V ou waterfall (en cascade). La notion même de "gestion de projet" est remise en question au profit de "gestion de produit". De façon à raisonner davantage "produit" que "projet". Après tout, l'objectif d'un projet consiste bien à donner naissance à un produit. Les méthodes agiles partent du principe que spécifier et planifier dans les détails l'intégralité d'un produit avant de le développer (approche prédictive) est contre-productif.

Une fois qu'une organisation décide d'adopter une gestion de développement Agile, il reste encore à choisir la méthodologie la plus adaptée à son projet. En effet, les méthodes Agiles disponibles sont nombreuses et peuvent être source de confusion. La méthode Agile offre plusieurs cadres de travail :

- Dynamics System Development Method (DSDM, 1995).
- Scrum (1996).
- Rapide Application Développement (RAD, 1991).

- Extreme Programming (XP, 1999).
- Adaptive Software Development (ASD, 2000).

Dans le cadre de notre projet de fin d'études et afin d'assurer le bon déroulement des différentes phases développement, test et déploiement, nous avons opté le cadre SCRUM.

Pourquoi SCRUM ? [1]

Avec une méthode de gestion de projet classique, la maîtrise d'ouvrage qui définit les besoins et la maîtrise d'œuvre qui définit les solutions techniques, sont généralement séparées. Par contre avec les méthodes agiles, comme SCRUM, brisent cette séparation et invitent tous les intervenants à travailler ensemble.

Cette proximité est à la fois dynamique et productive, elle engendre la réalisation des fonctionnalités par ensemble des itérations en incluant la participation du client. Chaque itération « sprint » peut durer entre deux à quatre semaines. A la fin de chaque itération ou sprint un produit fonctionnel doit être livré.

La méthode SCRUM présente plusieurs avantages qui s'adaptent parfaitement à la décomposition de la réalisation de notre projet :

- Plus de souplesse et de réactivité.
- Grande capacité d'adaptation au changement grâce à des itérations courtes.
- Satisfaire au mieux les besoins du client.

Caractéristiques du Scrum

- Itératif, lié à des processus
- incrémentaux. Approche basée sur l'équipe.
- Assure le développement des produits/applications nécessitant une grande adaptabilité.
- Contrôler le chaos résultat de conflits d'intérêt et des différents besoins.
- Augmenter la communication et maximiser la coopération.
- Protéger l'équipe des éléments externes perturbateurs.
- Un moyen d'augmenter la productivité.

Les rôles dans Scrum

La méthodologie SCRUM est une méthode où certains rôles sont attribués à des membres du projet qui vont permettre de superviser des parties précises. On distingue donc trois rôles principaux comme le montre la figure 1.9 :

- Le « Responsable Produit (PO) » : qui porte la vision du produit à réaliser (représentant généralement le client).
- Le « Scrum Master » : garant de l'application de la méthodologie Scrum.

- « L'équipe de développement » : regroupe tous les rôles habituellement nécessaires à un projet, comme l'architecte, le concepteur, le développeur, le testeur, etc...

Les rôles Scrum: Une approche différente pour atteindre le succès
En collaborant, une équipe Scrum livre plus de valeur d'affaires, plus rapidement.



Figure 1.9 – Les rôles Scrum

La figure suivante illustre le cycle durant la réalisation d'un projet en adoptant le cadre du travail scrum :

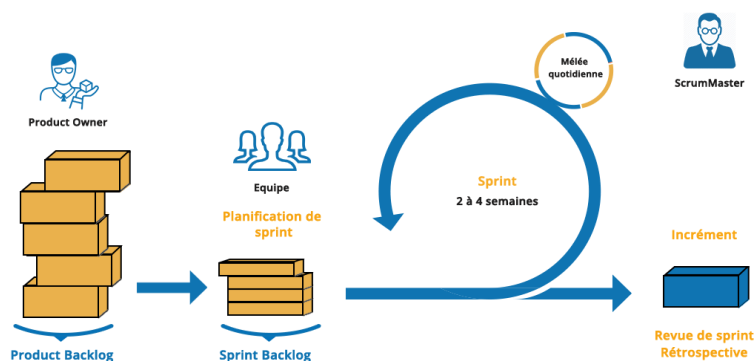


Figure 1.10 – Le cadre de travail SCRUM

1.4 Langages de modélisation

Le langage de modélisation unifié « UML » est un langage formel et normalisé. C'est le langage le plus populaire pour la modélisation graphique des solutions informatiques

pour faciliter la compréhension de la fonctionnalité du tel système informatique et leurs besoins dans le cadre de la conception orientée objet. L'UML offre des composants et des diagrammes pour la modélisation d'un projet tout au long de son cycle de vie. Ces diagrammes sont :

- Diagrammes Structure ou statique :
 - Diagramme de classes
 - Diagramme d'objets
 - Diagramme de composants
 - Diagramme de déploiement
- Diagrammes Comportementaux :
 - Diagramme de cas d'utilisation
 - Diagramme d'états transitions
 - Diagramme d'activité
- Diagrammes d'interaction ou dynamique :
 - Diagramme de séquence
 - Diagramme de communication

Conclusion

Ce chapitre a permis de situer le projet dans son cadre général en présentant l'organisme d'accueil, le contexte, la problématique, l'étude et la critique de l'existant ainsi que la solution proposée en précisant la méthodologie adoptée et le langage de modélisation utilisé. Le chapitre suivant sera consacré à l'analyse et aux spécifications des besoins du projet

Chapitre 2

Analyse et spécification des besoins

Introduction

Cette partie, nous procédons à la spécification des besoins fonctionnels et non fonctionnels. Nous commençons par définir nos acteurs, par la suite nous identifions également le backlog de produit et la planification des sprints, enfin, nous présentons l'architecture logicielle de l'application.

2.1 Capture des besoins globaux

L'identification des besoins est une étape primordiale et importante afin de déterminer les besoins et les attentes exactes du système. Ces besoins se déclinent en deux types de besoins : Fonctionnels et non fonctionnels.

2.1.1 Spécifications des Besoins Fonctionnels

— Identification des acteurs du système informatique :

nous allons présenter dans cette partie les acteurs de notre système et leurs responsabilités



Figure 2.1 – User

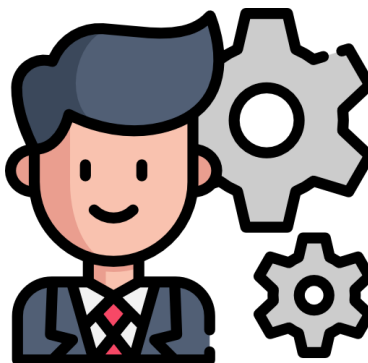


Figure 2.2 – Admin



Figure 2.3 – Système

— Identification des acteurs externes du système informatique :

- **User** : c'est l'utilisateur qui se bénéficie des services DFM.
- **Admin** : Cet utilisateur a le droit d'administration de l'application DFM
- **Système** : Cet utilisateur représente le système informatique qui exécute les tâches nécessaires pour le bon fonctionnement de l'application DFM

2.1.2 Présentation des différents modules

Dans cette section, nous présentons l'ensemble des besoins fonctionnels dans les différents modules de notre application, nous allons également présenter les différents acteurs interagissant avec le système. En effet, un acteur peut consulter ou modifier directement l'état du système, en émettant ou en recevant des messages éventuellement porteurs de données, pour mieux Comprendre le projet nous allons présenter les acteurs et leur interaction avec le système DFM existant :

Module	Action	Acteurs
Gestion des systèmes d'informations	Extraction des structures de données Gestion des structures de données	Admin et User
Gestion de projets	Gestion des projets Gestion des tableaux kanban Gestion des Mappings Rédaction et simulation des expressions au niveau des champs de la structure ciblent Schématisation des Flow Chart	Admin et User
Gestion des tickets	Affectation et suivi des tickets Modification de la description du ticket Approbation des tâches du ticket Écriture de commentaires au niveau du ticket	Admin et User
Visualisation des données	Visualisation de la cartographie	Admin et User
Module administratif	Gestion des utilisateurs Gestion des rôles Gestion des domaines de business Gestion des types de transformations Gestion du statut des tickets	Admin

TABLE 2.1 – Acteurs en interaction avec le système existant DFM

On presente maintenant le tableau des Acteur et interaction avec le système de notre projet :

Module	Action	Acteurs
Dictionnaire de données	Création d'un referentiel métier Liasions d'une structure de donnéesavec un referentiel métier Utilisation des structures de donnéesindexés dans un mapping	Admin et User
Indexation des données	Creation des modules elastic search	Système
Exploitation du dictionnaire	La recherche des données indexés Dashboarding des données	Adminet User
Gestion du dictionnaire	Gérer les référentiels métiers Gérer les mappings indexés. Gérer les liasions des structures dedonnées et les référentiels métiers	Admin

TABLE 2.2 – Acteurs en interaction avec les modules du projet

2.1.1 Spécifications des Besoins non fonctionnels

Les besoins non fonctionnels représentent les exigences implicites auxquelles le système doit répondre. Parmi ces besoins on cite :

Ergonomie et souplesse : l'application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support tenu.

Sécurité : La solution proposée permet à l'utilisateur une navigation sécurisée. Elle n'est accessible qu'avec une authentification. Aussi elle doit assurer la confidentialité et la sécurité des données des utilisateurs et les systèmes externes.

Maniabilité et scabilité : le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins du marché.

Interopérabilité : Permet une communication fluide et efficace entre différentes plateformes, applications et technologies, favorisant ainsi l'intégration et la collaboration entre les systèmes.

2.2 Pilotage du Projet avec Scrum

2.2.1 Equipe et rôle

L'élément fondamental de Scrum est une petite équipe de personnes, la Scrum Team. La Scrum Team se compose d'un Scrum Master, d'un Product Owner et des développeurs. Au sein de cette équipe, il n'y a pas de sous-équipes ou de hiérarchies. Il s'agit d'un groupe uni de professionnels focalisés sur un seul objectif à la fois, l'Objectif de Produit. Les Scrum Teams sont pluridisciplinaires, ce qui signifie que les membres ont toutes les compétences nécessaires pour créer de la valeur à chaque Sprint. Ils sont également autogérés, ce qui signifie qu'ils décident en interne qui fait quoi, quand et comment. Toute la Scrum Team est responsable de la création d'un Incrément de valeur et utile durant chaque Sprint. Scrum définit trois responsabilités spécifiques au sein de la Scrum Team : développeurs, le Product Owner et le Scrum Master. Voici Notre Equipe SCRUM :

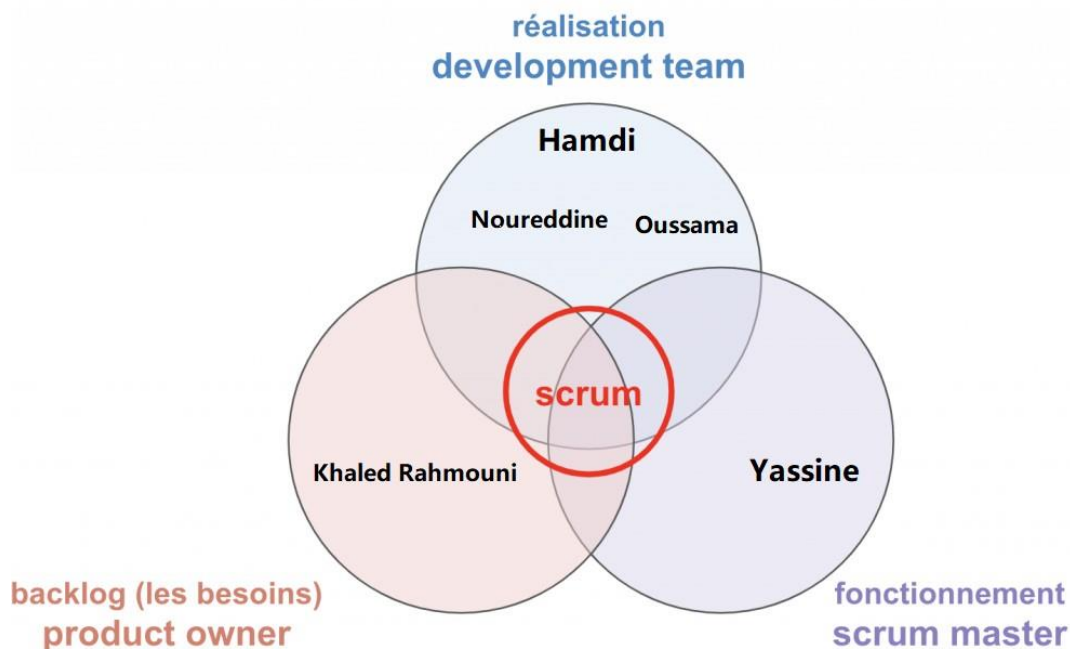


Figure 2.4 – Equipe Scrum

2.2.2 Backlog produit

Après avoir étudié les besoins de notre application, nous allons présenter le « PRO- DUCT BACKLOG » que nous allons suivre dans notre projet.

Module	ID	Acteur	Story	Priorité	Complexité
Dictionnaire de données	0	En tant qu'un User / Admin	Je dois pouvoir Créer un référentiel métier).	1	Forte
	1	En tant qu'un User / Admin	Je dois pouvoir lier une structure de données avec un référentiel métier.	1	Forte
	2	En tant qu'un User / Admin	Je dois pouvoir utiliser les structures des données indexés dans des mappings.	1	Forte
Exploitation du dictionnaire de données	3	En tant qu'un User / Admin	Je dois pouvoir chercher champs utilisés et avoir toutes les informations nécessaires incluant si ce champ utilisé dans un mapping ou non	2	Forte
	4	En tant qu'un User / Admin	Je peux visualiser mes données (Dashboarding des données).	3	Moyenne
Gestion de dictionnaire de données	5	En tant qu'un Admin	Je peux gérer les référentiels métiers.	3	Faible
	6	En tant qu'Admin	Je peux gérer les liaisons des structures de données et les référentiels métiers.	3	Faible

TABLE 2.3 – Backlog produit

Nous avons défini dans la figure ci-dessus la complexité de chaque User Story selon 3 catégories :

-Complexité Forte : attribuée aux user stories qui nécessitent une conception complexe avec un développement spécifique et complexe.

-Complexité Moyenne : attribuée aux user stories qui nécessitent un développement spécifique et complexe.

-Complexité Faible : attribuée aux user stories qui présentent un CRUD.

2.2.3 Planification des sprints

Un sprint est une unité de temps définie dans la méthodologie Agile, utilisée pour la gestion de projet. Il s'agit d'une période de travail itérative et incrémentale au cours de laquelle une équipe de développement se concentre sur la réalisation de fonctionnalités spécifiques du projet.

1) Structuration des Sprints

Dans cette section, nous allons entamer la planification de nos sprints qu'on a fixés auparavant en se basant sur les priorités des tâches et leurs complexités.

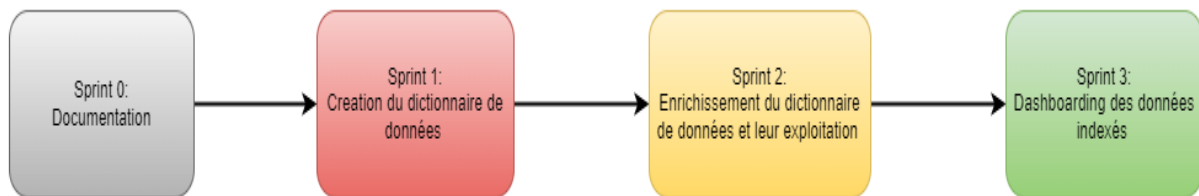


Figure 2.5 – Structurations des sprints

2.3 Analyse globale

2.3.1 Diagramme de cas d'utilisation globale

Les besoins et les acteurs que nous avons identifiés se traduisent dans ce diagramme qui sera notre repère tout au long du développement. Ce diagramme nous donne une vision globale sur l'application web. Au cours des sprints, on construira une conception plus détaillée d'une façon incrémentale.

Diagramme de cas d'utilisation du Projet PFE :

Ce diagramme est la représentation des fonctionnalités et les rôles de chaque acteur dans notre application :

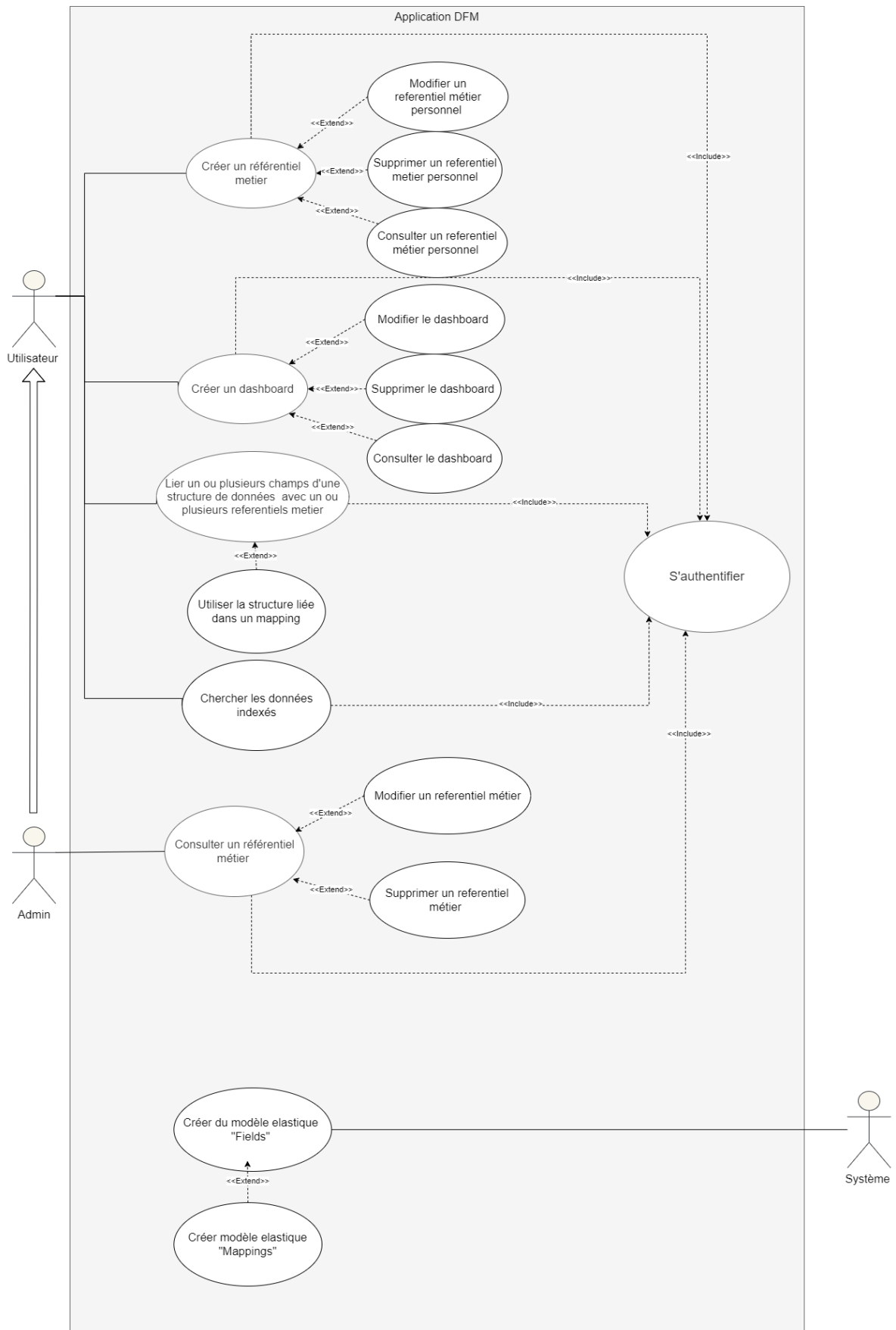


Figure 2.6 – Diagramme de cas d'utilisation globale de projet

Nous allons par la suite présenter le diagramme de classe général de DFM puis le diagramme de classe de notre projet, un diagramme de classes est un type de diagramme de modélisation qui représente les classes, les relations entre les classes, les attributs et les méthodes dans un système logiciel. Il est utilisé pour visualiser la structure statique d'un système et les interactions entre les différentes entités :

1) Diagramme de classe de projet DFM :

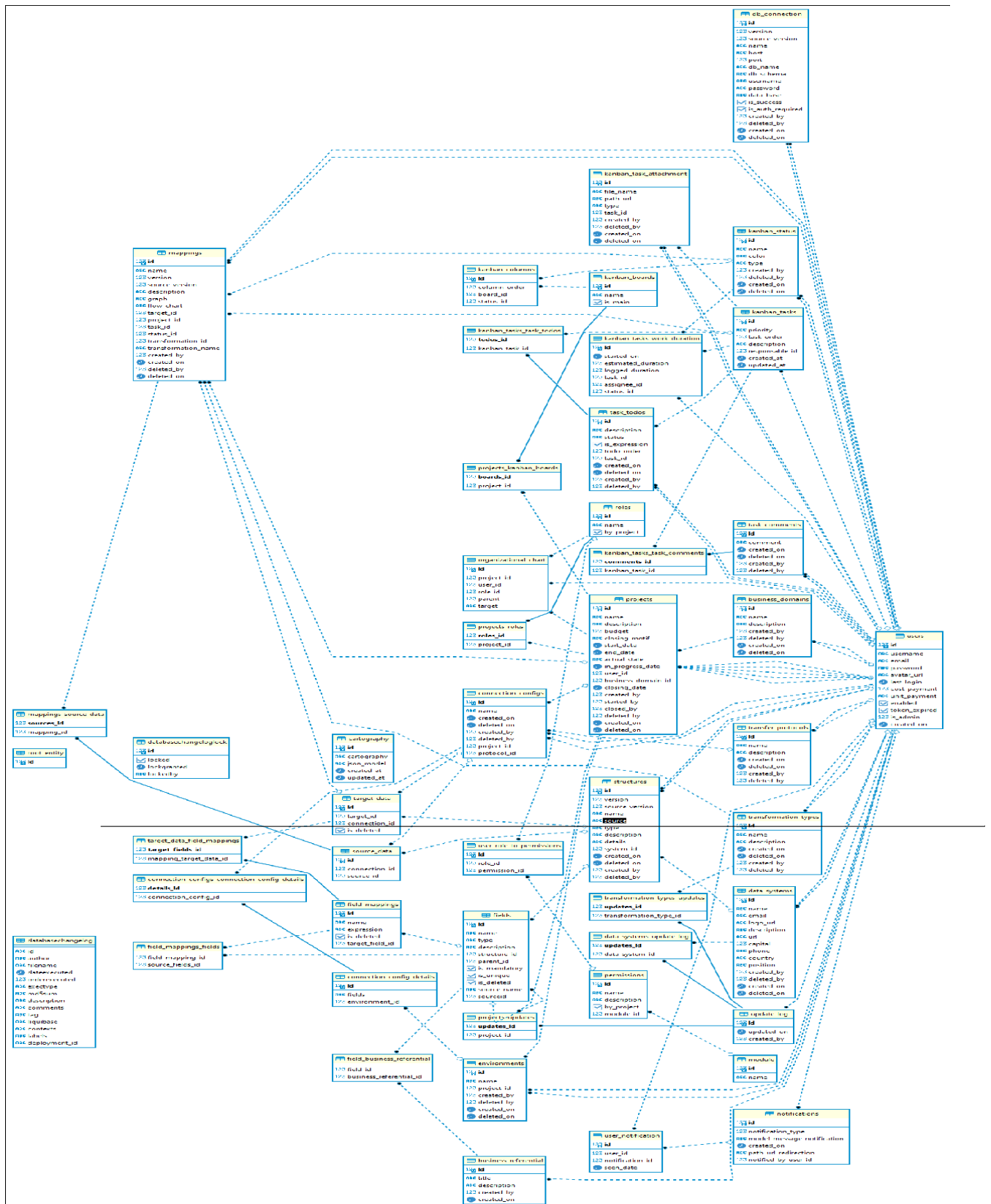


Figure 2.7 – Diagramme de classes d'analyse global de DFM

2) Diagramme de classe de projet pfe :

Ce diagramme de classes présente les entités clés sur lesquelles nous nous concentrerons afin de créer notre dictionnaire.

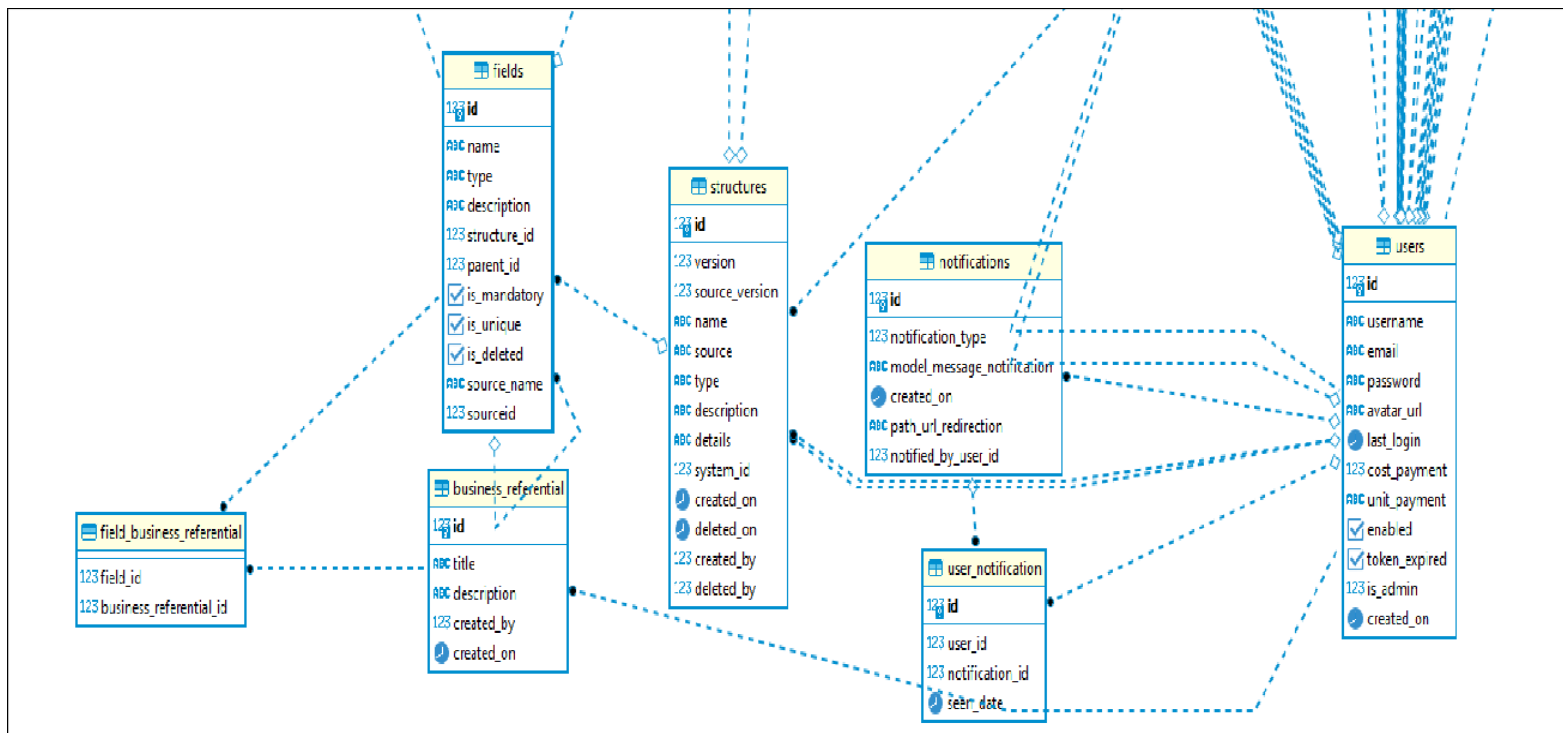


Figure 2.8 – Diagramme de classes d’analyse global de notre projet

2.4 Architecture logique globale de l’application

2.4.1 Architecture logique BACKEND et FRONTEND

Pour mieux comprendre l’architecture globale, nous allons présenter l’architecture logique entre le Front-end et le Back-end avec la liaison entre ces deux environnements

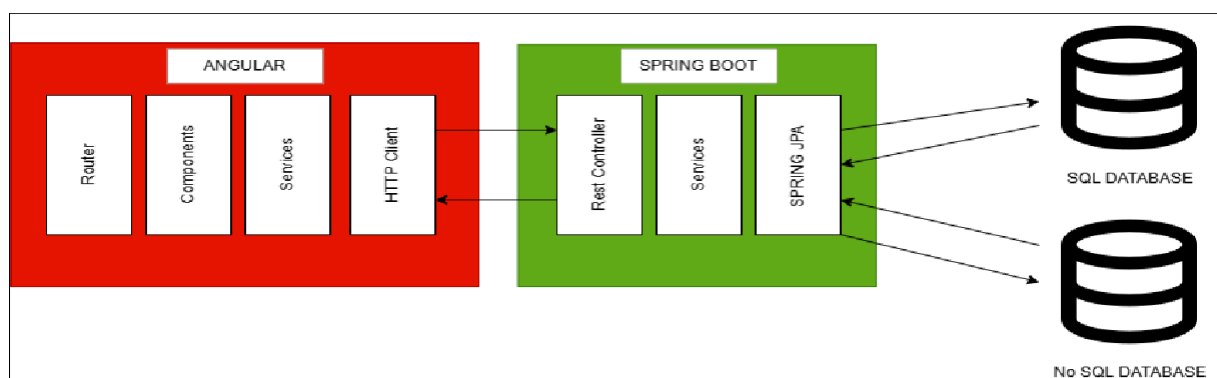


Figure 2.9 – Architecture logique BACKEND et FRONTEND

2.4.2 Architecture logique -FRONTEND-

Ensuite, nous allons présenter l'architecture logique de la partie du Front End qui se base sur l'architecture MVC :

Model (modèle) : gère les données et la logique métier.

View (vue) : gère la disposition et l'affichage.

Controller (contrôleur) : achemine les commandes des parties "model" et "view".

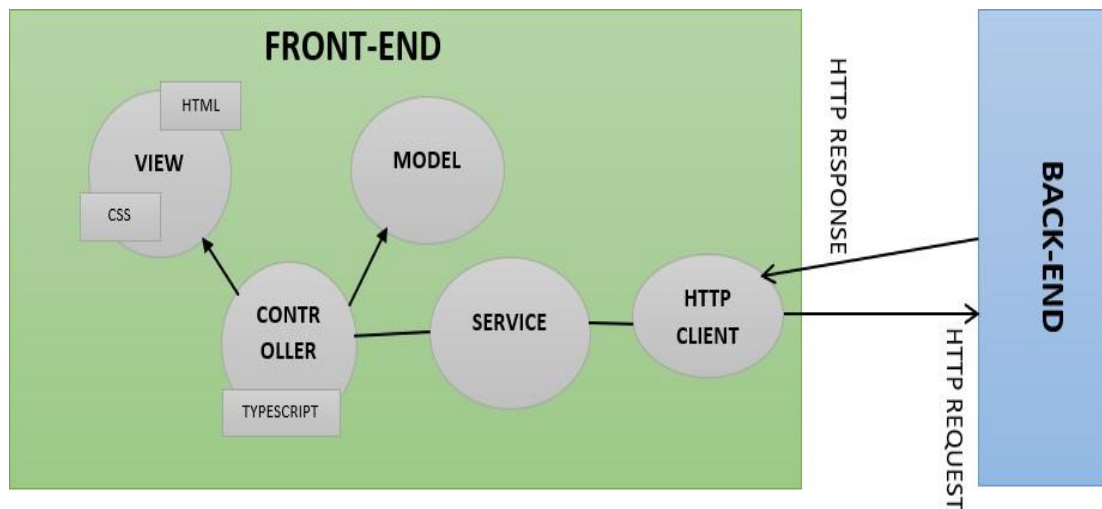


Figure 2.10 – Architecture logicielle -FRONTEND

2.4.3 Architecture logique -BACKEND-

Dans la partie Back -End nous allons travailler sur une application avec une architecture Mono-service qui se traduit dans la figure suivante :

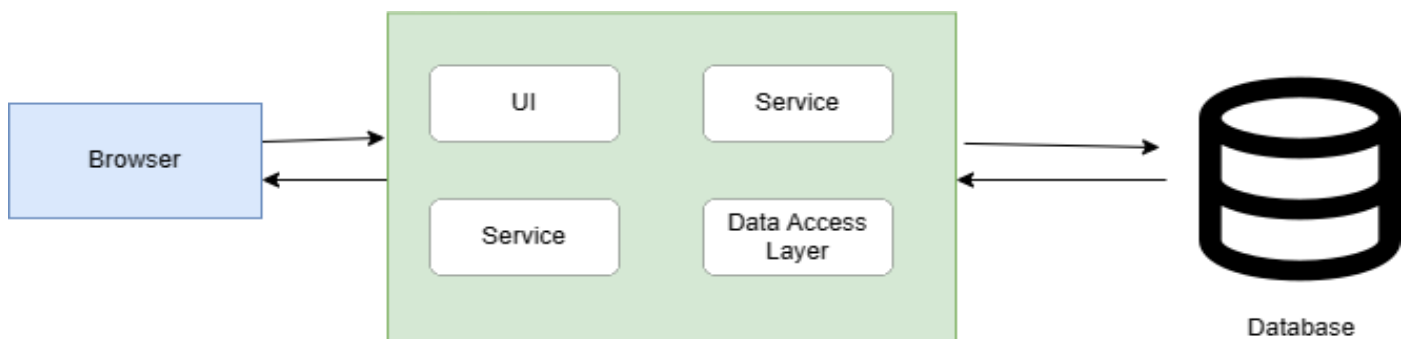


Figure 2.11 – Architecture logicielle -FRONTEND

L'architecture monolithique est un modèle d'architecture de logiciel dans lequel toutes les fonctionnalités d'une application sont combinées dans un seul programme ou un seul module. Dans cette architecture, toutes les couches de l'application (interface utilisateur, logique métier, stockage de données) sont intégrées dans un seul bloc, avec une base de code unique et une base de données centrale.

2.5 Étude des outils

Afin de réussir à développer notre projet, nous avons besoins de plusieurs outils et environnements logiciels. Nous allons maintenant exposer les outils utilisés avec une description de chaque outil :

Outils	DESCRIPTION
JDK 1.8(Java Development Kit)[2]	C'est l'ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.
SpringToolSuite4 (IDE)[3]	C'est l'environnement de développement basé sur Eclipse. Il fournit un environnement prêt à l'emploi pour implémenter, exécuter, déployer et déboguer l'application. Il valide notre application et fournit des correctifs rapides pour les applications.

Apache Maven[4]	<p>C'est un outil de gestion et d'automatisation de production des projets logiciels Java. Maven est géré par l'organisation Apache Software Foundation. Il permet notamment :</p> <ul style="list-style-type: none"> — D'automatiser certaines tâches : compilation, tests unitaires et déploiement des applications qui composent le projet — De gérer des dépendances vis-à-vis des bibliothèques nécessaires au projet — De générer des documentations concernant le projet
Spring boot[5]	<p>Spring Boot est un framework qui facilite le développement d'applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en jar, totalement autonome. Ce qui nous intéresse particulièrement, puisque nous essayons de développer un monoservice</p>
JPA[6]	<p>JPA est une spécification qui décrit l'interface des map-pages objet-relationnel et de la gestion des objets persistants. Au-delà de la définition de l'API, il définit l'ensemble de règles et de directives relatives à la mise en œuvre du mappage objet-relationnel et à la conformité de ces spécifications avec tous les JPA providers. C'est une spécification d'interface conçue pour créer une norme pour l'implémentation ORM afin de cibler les exigences de la couche d'accès aux données. Hibernate est le framework JPA le plus populaire.</p>
Hibernate[7]	<p>C'est un framework ORM (Object / Relational Mapping) pour le langage de programmation Java qui concerne la persistance des données. Il s'agit simplement d'une solution de mappage objet-relationnel open-source qui mappe les classes Java aux tables de bases de données dans les bases de données relationnelles et des types de données Java à SQL. Il s'agit d'un framework ORM léger distribué sous licence GNU qui vise à fournir aux développeurs un mécanisme transparent permettant d'obtenir de la persistance. Il utilise un puissant langage de requête semblable au SQL appelé HQL (abréviation de Hibernate Query Language), qui s'intègre rapidement à la fois aux applications existantes et nouvelles sans modifier le reste de l'application.</p>

ElasticSearch[8]	C'est un système de recherche et d'analyse de données open source, conçu pour stocker, récupérer, gérer et analyser des données à grande échelle en temps réel. Il utilise une architecture de type "moteur de recherche" et est basé sur Apache Lucene, une bibliothèque de recherche et d'indexation écrite en Java. Elasticsearch est couramment utilisé pour des applications de recherche, de surveillance de logs, de surveillance d'infrastructures, d'analyse de données et de recherche de texte intégral. Il est connu pour sa rapidité, sa scalabilité, sa flexibilité et sa facilité d'utilisation.
Kibana[9]	IDE qui fournit une interface graphique pour se connecter et travailler avec le serveur Microsoft SQL. Il a été lancé avec Microsoft SQL Server 2005 et est utilisé pour configurer, gérer et administrer tous les composants de Microsoft SQL Server.
Star UML[10]	StarUML est un logiciel de modélisation UML, qui a été « cédé comme open source » par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL. StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0.
Angular[11]	framework côté client, open source, basé sur TypeScript, et co-dirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète d'AngularJS, un framework construit par la même équipe. Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif.
Internationalisation (I18n) ngx-translate librairie[12]	Internationalisation est le processus de prise en charge de plusieurs langues dans notre application. Ce processus est implémenté à l'aide de la librairie ngx-translate.
Visual Studio Code (IDE)	Editeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Il permet le débogage, la mise en évidence de la syntaxe, la complétion intelligente du code et Git intégré.

Node.js[13]	Plateforme logicielle libre en JavaScript, orientée vers les applications réseau évènementielles hautement concurrentes qui doivent pouvoir monter en charge (extensibilité ou scalabilité désigne la capacité d'un produit à s'adapter à un changement d'ordre de grandeur de la demande).
npm[14]	Le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, npm appartient à l'environnement, il est donc automatiquement installé par défaut. Npm gère les dépendances pour une application et fonctionne à l'aide d'un terminal. Il permet également d'installer des applications Node.js qui existent sur le dépôt npm.
Angular CLI[15]	Outil d'interface de ligne de commande que vous utilisez pour initialiser, développer, échafauder et maintenir des applications Angular directement à partir d'un shell de commande.
Bootstrap[16]	Collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.
Typescript[17]	Il s'agit d'un langage de programmation open source et libre développée par Microsoft afin d'améliorer et de sécuriser la production de code JavaScript. C'est un sur-ensemble syntaxique strict de JavaScript. Le code TypeScript est transcompilé en JavaScript, et peut être interprété par n'importe quel moteur JavaScript ou navigateur web.
Html[18]	Le HyperText Markup Langage, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.
Sourcetree[19]	Simplifie la façon dont vous interagissez avec vos référentiels Git afin que vous puissiez vous concentrer sur le codage. Visualisez et gérez vos référentiels via la simple interface graphique Git de Sourcetree.

Bitbucket[20]	Service web d'hébergement et de gestion de développement logiciel utilisant le logiciel de gestion de versions Git (et par le passé également le logiciel Mercurial). Hébergé sur un serveur Bitbucket ou sur Bitbucket Cloud, le repository en question pourra accueillir les fichiers et commits du projet, et orchestrera les versions de développement... Le tout pourra ensuite être cloné du serveur Bitbucket ou de Bitbucket Cloud vers un poste de développement local. Bitbucket permet également la gestion des git repositories, des branches, des commits . . . et facilite la synchronisation automatique entre l'équipe de développement.
JIRA[21]	Système de suivi de bugs, de gestion des incidents et de gestion de projets développé par Atlassian et publié pour la première fois en 2002. Il propose des solutions à la fois à destination des développeurs et des intervenants non développeurs. Destiné aux sociétés désireuses de mettre en place un fonctionnement en méthode agile et facilite le travail des utilisateurs concernés dans leur organisation. Il permet notamment la création et la planification de tâches via un système de rédaction et de gestion des récits utilisateurs.

TABLE 2.4 – Étude des outils

Conclusion

Durant ce chapitre, nous avons organisé le terrain pour le développement de notre projet. Nous avons commencé par l'identification des besoins fonctionnels et non fonctionnels ainsi que les acteurs du système. Nous avons également formé le Backlog produit et les diagrammes d'analyse, de plus une présentation de l'architecture de l'application a été présentée. Pour terminer nous avons énumérer les outils utilisés avec leurs utilités et planifier les sprints du projet.

Chapitre 3

**Sprint 1 : La Création du dictionnaire
de données**

Introduction

Dans ce chapitre, nous allons exposer le premier Sprint de la gestion des événements et rendez-vous ainsi que la gestion de la boîte de messagerie. Au début, nous allons donner les spécifications et l'analyse des besoins, ensuite, nous allons préparer les diagrammes de conception puis nous allons présenter la réalisation du premier sprint avec les captures d'écrans de l'application. Enfin Les méthodes de tests travaillées et leur validation seront présentées.

3.1 Spécification et analyse des besoins

3.1.1 Backlog du sprint 1

ID	User Story	Taches	Temps estimé	Fonctionnalités
1	En tant qu'User / Admin, Je dois pouvoir créer consulter modifier supprimer mes référentiel métiers	Préparer la conception du module Referentielmetier	1w	<ul style="list-style-type: none"> • Afficher un champ pour la création d'un referentiel métier • Afficher un message s'il existe • Afficher les référentiels métier crée par l'utilisateur • Modifier et supprimer le référentiel métier
		Préparer les requêtes SQL de création des tables	1w	
		Préparer le champ de Création (Front end)	1w	
		Préparer les services de CRUD.	2w	
2	En tant qu'Admin / User, Je dois pouvoir lier les champs d'une structure de données avec un où plusieurs référentiel metier	Modifier la conception des module des champs et des structures de données	2w	<ul style="list-style-type: none"> • Afficher un champ pour insérer un referentiel métier dans le menu de structure de données. • Afficher la liste des référentiel métiers enregistrés dans la base • Mettre à jour la structure de donnée
		Préparer les requêtes SQL pour la liaison des différentes entités		
		L'ajout d'un champ dans le menu de structure de données pour l'ajout d'un référentiel métier		

		Preparer les services CRUD		
3	Le système doit Automatisement indexer les champs qui on été liés avec un ou plusieurs referentiel métier dans elasticSearch .	Configurer et liée Élastique search avec notre environnement SPRING Création de model elastique.	2w	<ul style="list-style-type: none"> Création d'un elastique model simplifié pour les champs indexés

TABLE 3.1 – BACKLOG SPRINT 1

Le Backlog du Sprint est l'un des artefacts de SCRUM, chaque nouveau sprint aura son propre backlog de manière que nous, en tant qu'équipe de développement nous travaillons qu'avec un seul backlog lors d'un sprint. Au cours de la réunion de planification du sprint, nous avons choisi un certain nombre d'éléments du « Product Backlog » puis nous avons distingué les tâches à faire durant ce sprint. La figure suivante détermine le sprint Backlog du premier sprint.

3.1.2 Diagramme raffiné du cas d'utilisation : Sprint 1

1) Diagramme de cas d'utilisation : Créer un référentiel métier

Dans cette partie, nous allons présenter le diagramme de cas d'utilisation du premiersprint ainsi que la description textuelle des cas d'utilisation :

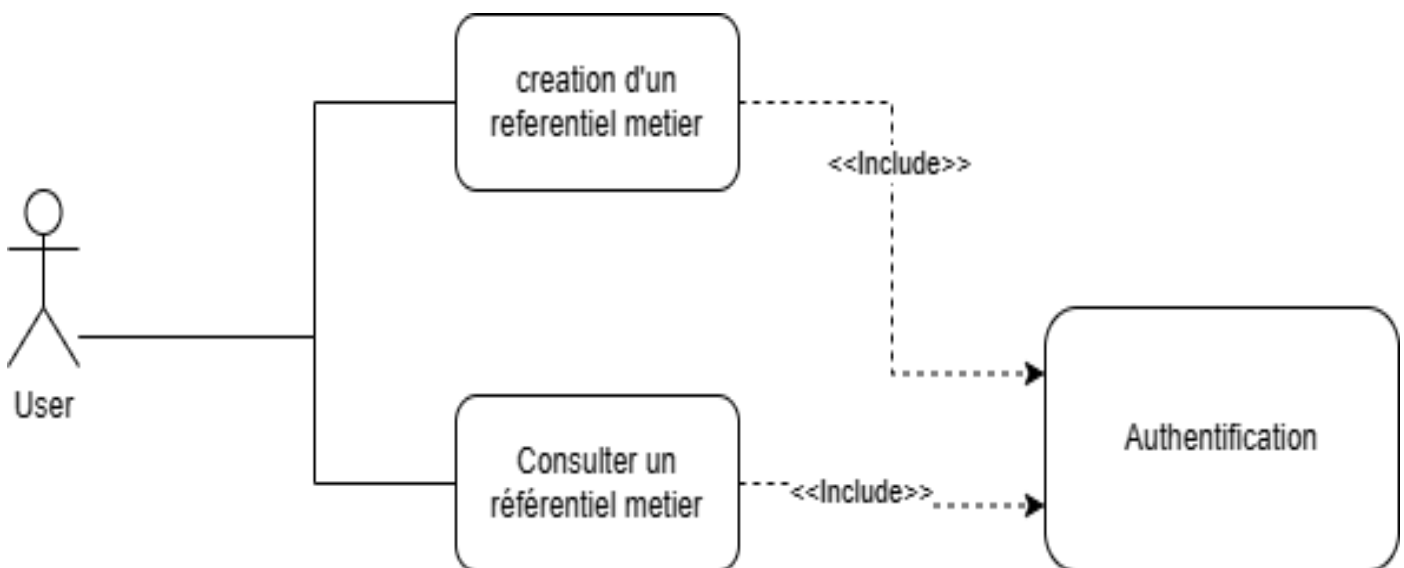


Figure 3.1 – Diagramme de cas d'utilisation création d'un referentiel métier

2) Raffinement des cas d'utilisation :

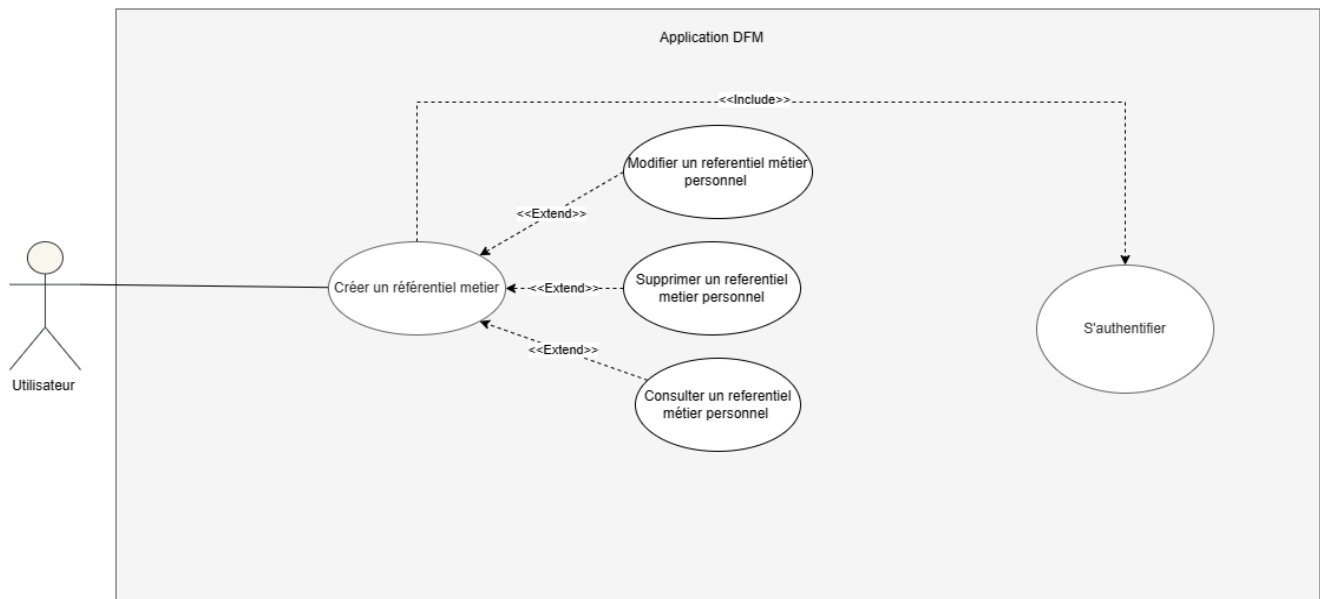


Figure 3.2 – Cas d'utilisation "Création d'un référentiel métier "

3) Diagramme raffiné du cas d'utilisation : Création du référentiel métier

La création d'un référentiel métier est la première étape pour la création d'un dictionnaire de données on va utiliser cette entité dans la liaison entre un référentiel et un champs d'une structure de données.

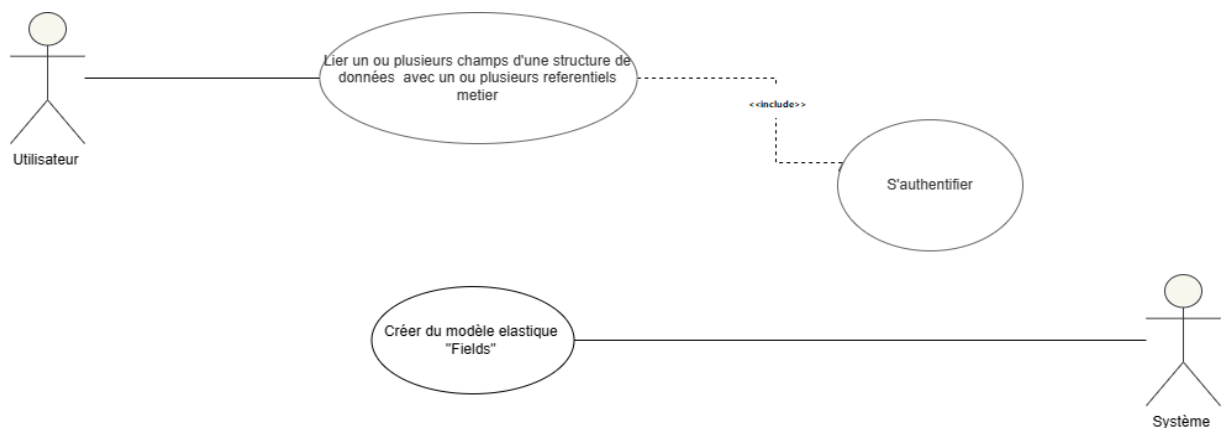


Figure 3.3 – Cas d'utilisation "Création du dictionnaire des données "

3.1.3 Description textuelle des cas d'utilisations Sprint 2

1) Description de cas d'utilisation : Création d'un référentiel métier

Durant cette section nous allons raffiner le cas d'utilisation (Création du référentiel métier) en donnant la description textuelle :

Cas d'utilisation « Création d'un référentiel métier »	
Acteurs	User, Admin
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale Scénario alternatif	<ul style="list-style-type: none"> • L'utilisateur fait la création d'un référentiel métier. • Si le référentiel métier n'est pas enregistré dans la base, il va être créé directement après la confirmation d'input.
Exception : Si le référentiel métier est déjà créé il ne va pas être créé une autre fois	

TABLE 3.2 – Description textuelle du cas d'utilisation "Création du référentiel métier"

2) Description de cas d'utilisation : Création du dictionnaire de données

Cas d'utilisation « Création du dictionnaire de données »	
Acteurs	User, Admin, Système
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur fait la liaison d'un champ d'une structure de données avec un référentiel métier créé ou non.

Exception : Si le referentiel métier et déjà sélectionné il ne sera pas sélectionné une autre fois

TABLE 3.3 – Description textuelle du cas d'utilisation "Création du dictionnaire de données"

3.1.1 Diagrammes de séquence sprint 2 :

Afin de mieux étudier les interactions entre les acteurs et le système selon un ordre chronologique, nous allons présenter les diagrammes de séquence d'analyse du premier sprint qui englobe le module création du dictionnaire de données.

1) Diagramme de séquence pour la création d'un nouveau référentiel métier :

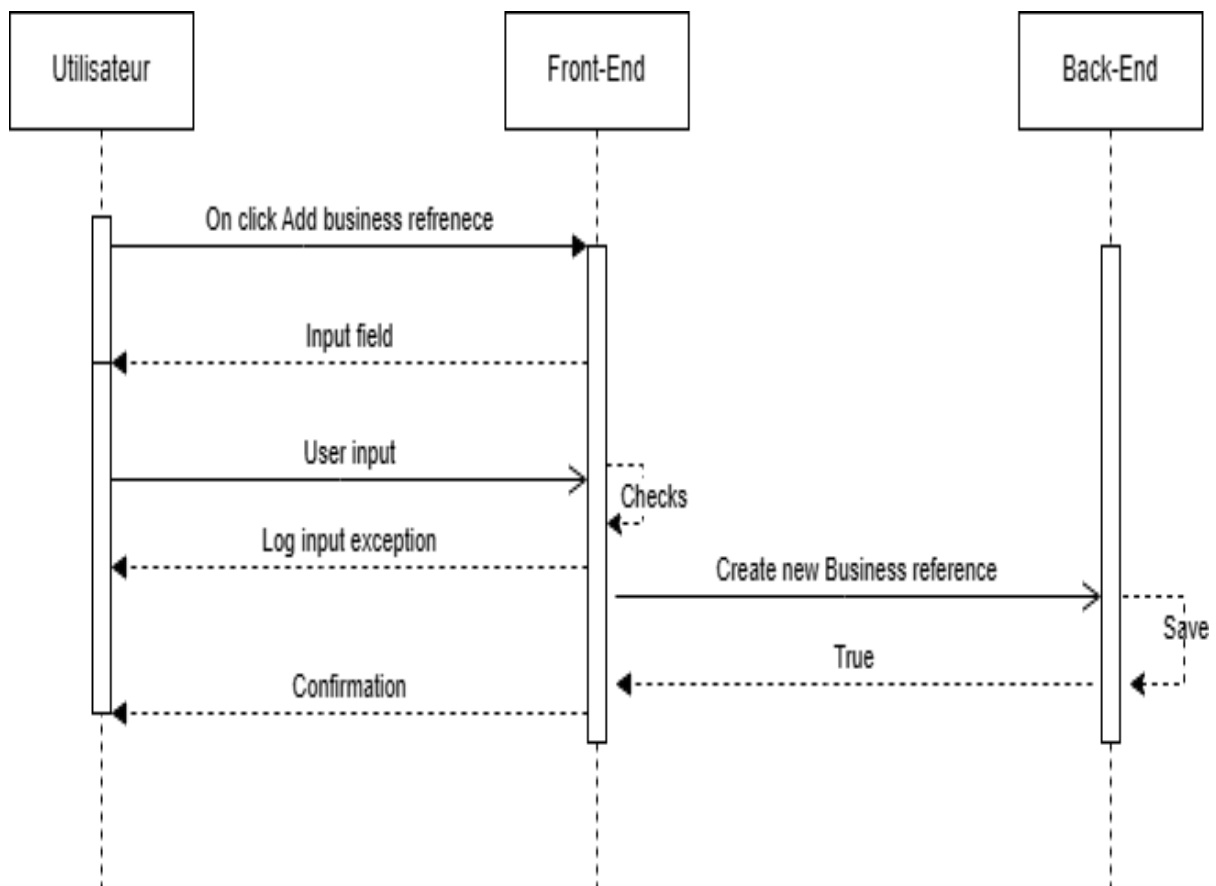


Figure 3.4 – Diagramme de sequence de creation d'un nouveau référentiel métier.

2) Diagramme de séquence pour la consultaion des référentiels métiers :

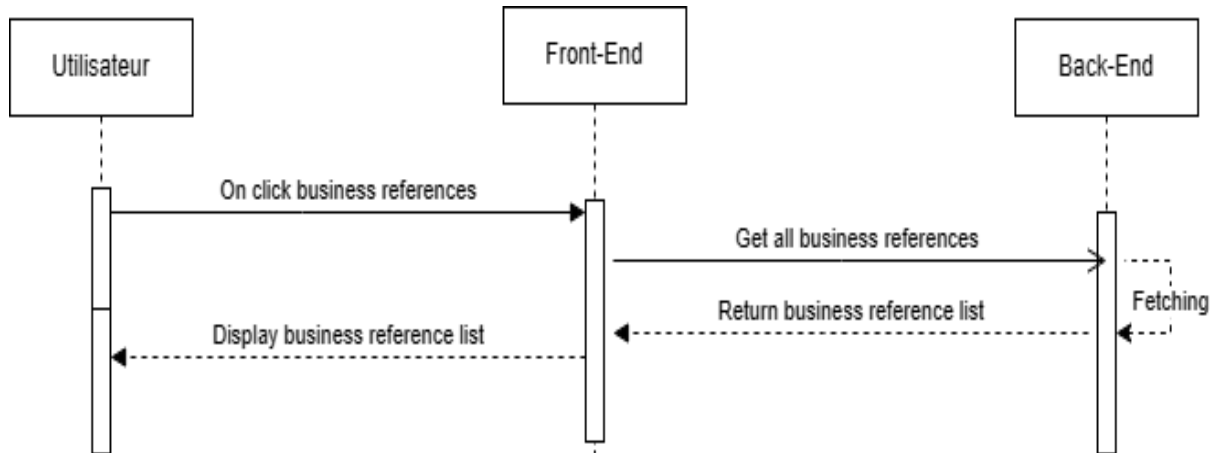


Figure 3.5 – Diagramme de séquence de consultation des référentiels métiers.

3) Diagramme de séquence pour la modification des référentiels métiers :

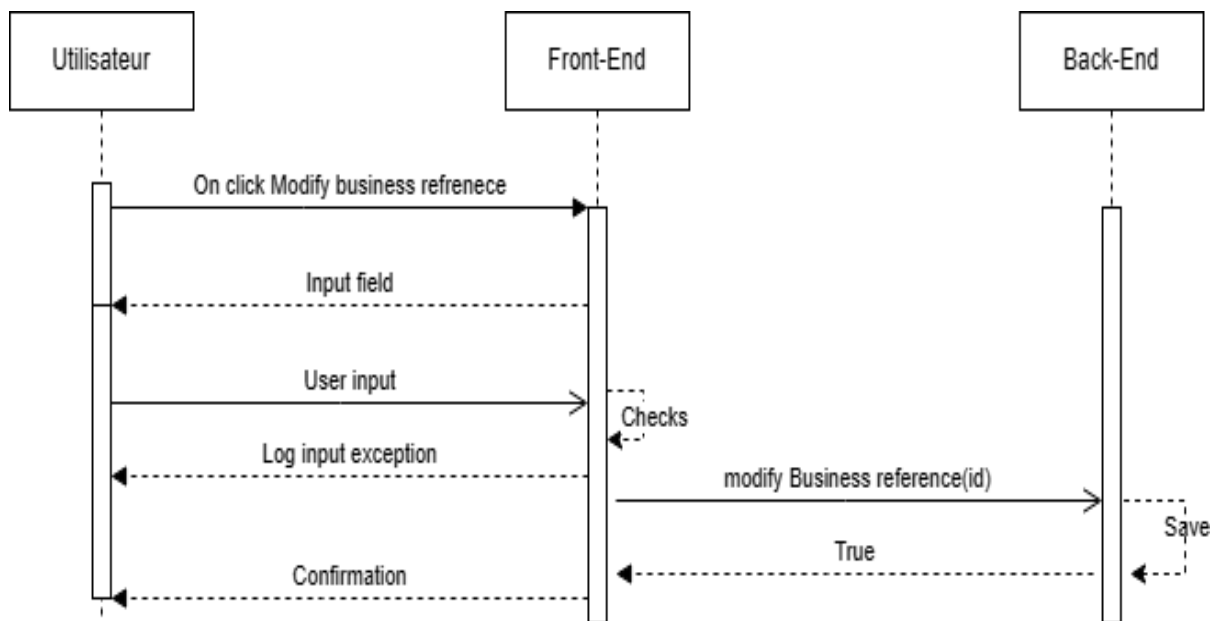


Figure 3.6 – Diagramme de séquence de modification des référentiels métiers.

4) Diagramme de séquence pour la suppression des référentiels métiers :

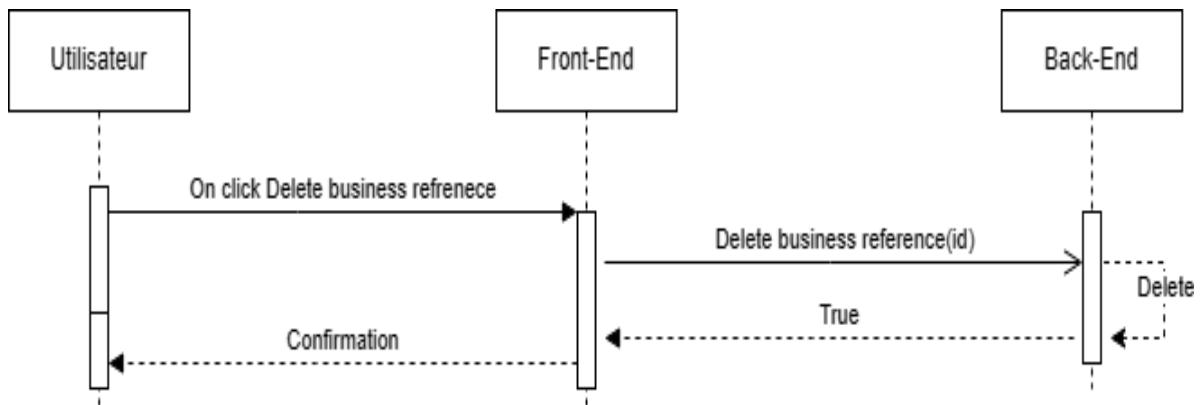


Figure 3.7 – Diagramme de séquence de la suppression des référentiels métiers.

5) Diagramme de séquence pour la création du dictionnaire de données :

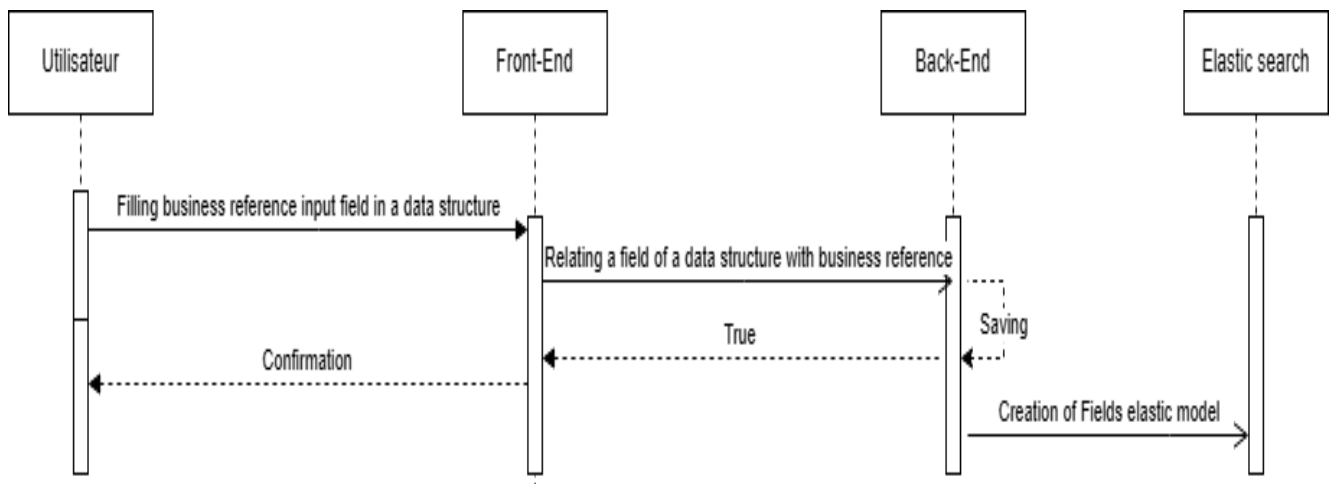


Figure 3.8 – Diagramme de séquence de la création du dictionnaire de données.

3.2 Réalisation

3.2.1 Réalisation du cas d'utilisation Créer un référentiel métier

Dans cette partie nous allons présenter le travail déjà réalisé à l'aide des captures d'écrans et des schémas explicatifs des parties fonctionnelles de notre application développée pendant le sprint 1.

1) Menu d'utilisateur des settings

Nous avons ajouté une nouvelle section intitulée "Référentiel Métier" dans notre menu, offrant un accès direct à la liste complète des référentiels métier.

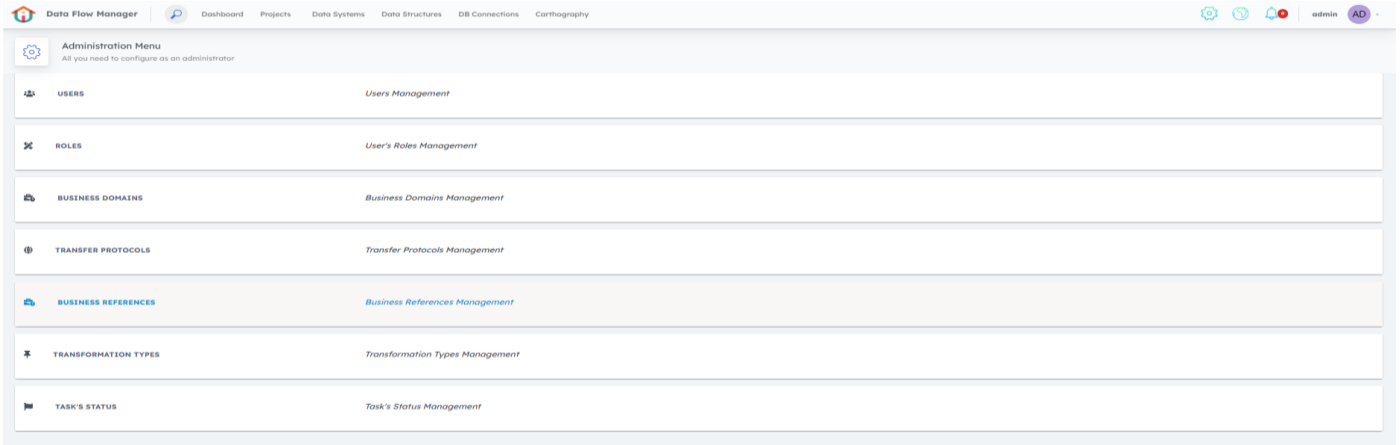


Figure 3.9 – Page de menu settings

2) Créer un nouveau referentiel metier :

En cliquant sur "New", un pop-up apparaît, vous permettant de créer un nouveau référentiel métier en saisissant les informations pertinentes.

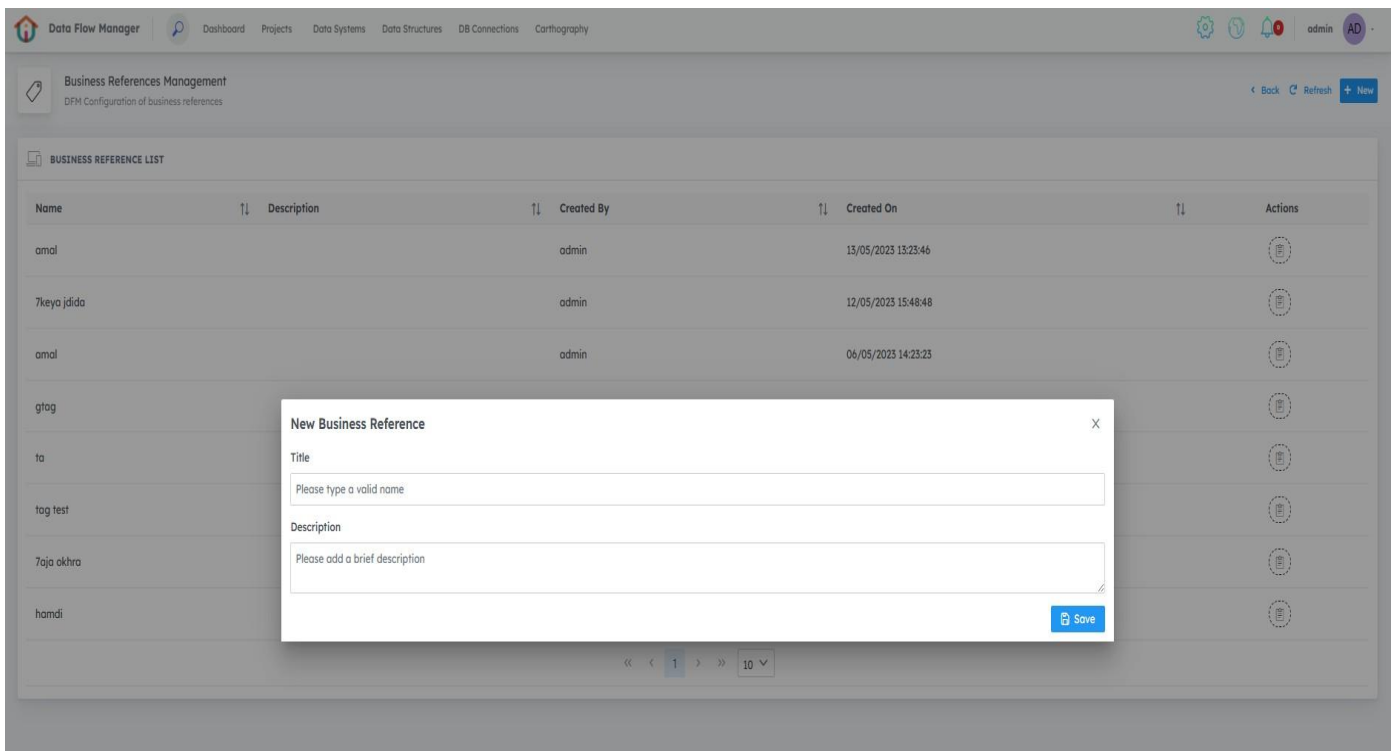


Figure 3.10 – Interface de création du référentiel métier

3.2.1 Liaison des champs de structure de données avec un/des référentiels métiers

1) Liste des structures de données :

Dans cette interface, vous trouverez la liste des structures de données, y compris les informations suivantes : nom, version, type, créé par, créé le.

	Name	Current Version	Data System	Type	Created By	Created On	Actions
<input type="checkbox"/>	Test_csv	5	Test_2	Csv	admin	12/05/2023 03:48:48	
<input type="checkbox"/>	Annual-enterprise-survey-2021-financial-year-provisional-csv	6	Test_2	Csv	admin	05/05/2023 04:42:59	

Figure 3.11 – Liste des structures de données

2) Liaison d'un champ d'une structure avec le référentiel métier :

Lorsque vous cliquez pour modifier une structure de données, ce pop-up s'ouvre et vous permet d'ajouter une nouvelle colonne nommée "Référentiel Métier" où vous pouvez lier la structure à un référentiel métier. Cette action déclenchera la création d'un nouveau modèle élastique dans l'index des champs de notre dictionnaire de données.

On peut créer directement des référentiel métier qui n'existe pas dans la base pour donner à l'utilisateur plus de liberté

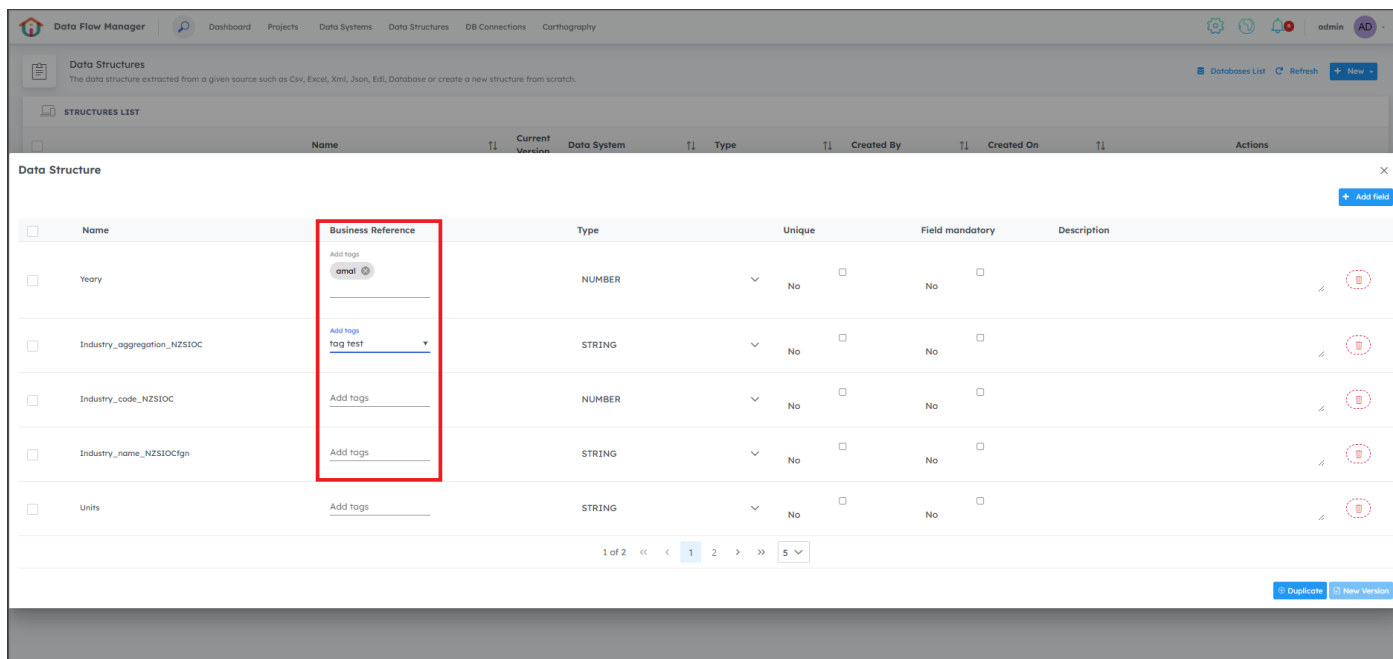


Figure 3.12 – Liaison des champs de structure de données avec un/des référentiels métiers

3) Visualisation des modelés elastique créés sur kibana :

Dans cette capture, Kibana est ouvert, offrant une visualisation des index, des modèles élastiques créés et des attributs associés à chaque index.

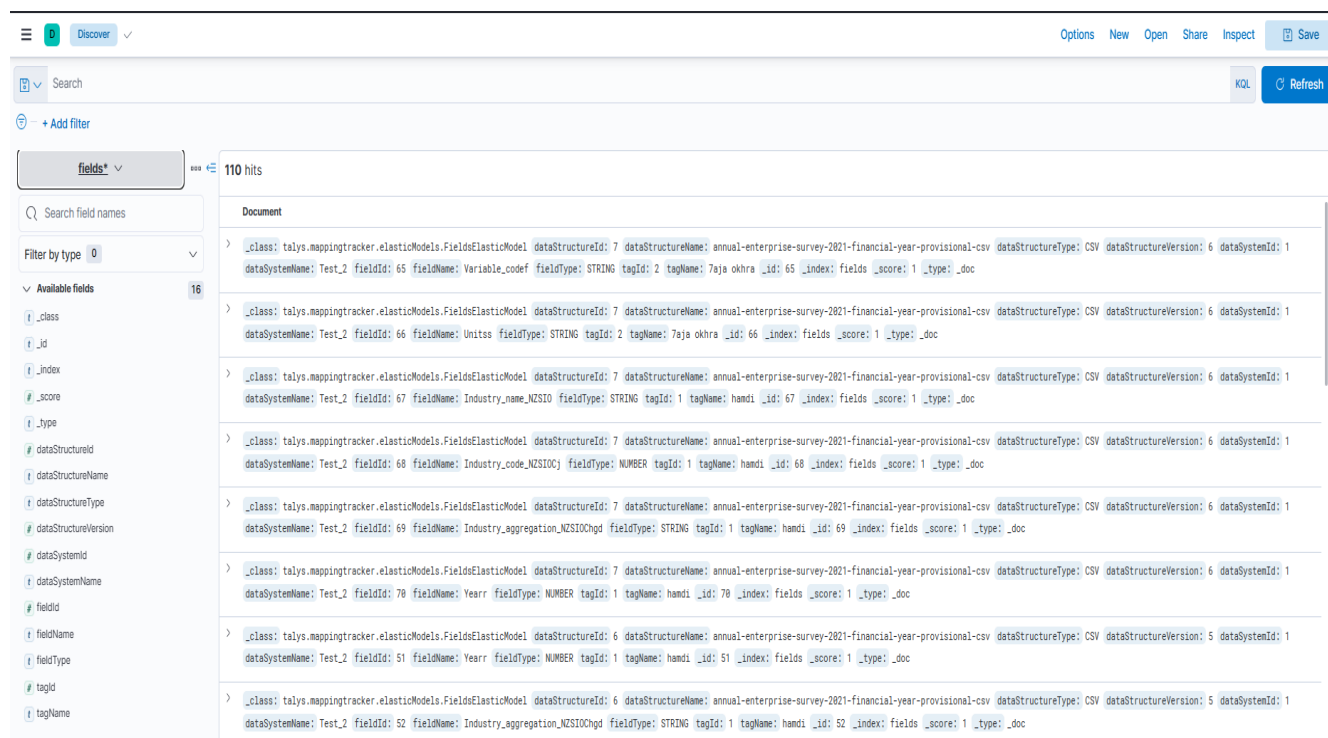


Figure 3.13 – Visualisation des objets élastique

Conclusion

Tout au long de ce chapitre, nous avons développé la conception du premier sprint afin d'avoir le passage souple et facile à l'étape suivante. Pour ce faire, nous avons présenté en premier lieu le backlog de Sprint. En second lieu, l'authentification des différents acteurs,. Ensuite nous avons présenté les diagrammes des classes et de séquence. Finalement nous avons présenté quelques captures et schémas explicatifs pour la partie réalisation. Le chapitre suivant sera consacré à la présentation du travail réalisé suite à la première Sprint.

Chapitre 4

**Sprint 2 : Enrichissement du dictionnaire
de données et leur exploitation.**

Introduction

Dans ce chapitre, nous allons exposer le deuxième Sprint au début, nous allons donner les spécifications et l'analyse des besoins, ensuite, nous allons préparer les diagrammes de conception puis nous allons présenter la réalisation du deuxième sprint avec les captures d'écrans de l'application.

L'enrichissement du dictionnaire de données consiste à ajouter le modèle élastique "Mappings". Dans notre application, le concept de mapping implique l'utilisation d'une structure de données appelée "target" qui utilise les autres structures de données en tant que sources.

La partie exploitation des données comprend la capacité de rechercher les champs indexés et de déterminer s'ils sont utilisés dans des mappings ou non, ainsi que de déterminer s'ils sont utilisés en tant que cible (target) ou source.

4.1 Spécification et analyse des besoins

4.1.1 Backlog du sprint 2 : enrichissement du dictionnaire de données

ID	User Story	Taches	Temps estimé	Fonctionnalités
1	En tant que User / Admin, Je dois pouvoir utiliser les champs liés avec un référentiel métier dans un mapping.	Préparer la conception du module élastique « mappings »	1w	<ul style="list-style-type: none"> — L'exécution de la méthode de synchronisation quand un mapping avec une structure indexé se déplace vers PROD — Le mapping élastique doit être lié avec les modèles Fields élastique utilisé dans la structure métier
		Préparer l'entité élastique « mappings »	1w	
		Préparer la fonction de création de module élastique « mappings » et les modules sprint nécessaire pour assurer le bon fonctionnement	1w	

2	En tant qu'Admin / User, Je dois pouvoir chercher les champs liés avec un référentiel métier et connaître s'ils sont utilisés dans un mapping ou non et les informations complémentaires à ce champs	Préparer les requêtes elastic search de recherche	2w	<ul style="list-style-type: none"> Afficher un champ pour l'insertion d'utilisateur Afficher l'interface de résultats de recherche Afficher les mappings associés avec ce champ s'ils existent et les informations nécessaires
		Implémenter les requêtes dans une fonction et l'implémenter dans l'application Back et front .		
		Préparer l'interface et leur design		

TABLE 4.1 – BACKLOG SPRINT 2

4.1.2 Les Diagrammes de cas d'utilisations : Sprint 2

1) Diagramme de cas d'utilisation : Créer un modèle élastique « Mappings »

Dans cette partie, nous allons présenter le diagramme de cas d'utilisation créer un modèle élastique « Mappings » :



Figure 4.1 – Diagramme de cas d'utilisation créer un modèle élastique « Mappings »

2) Diagramme de cas d'utilisation : Chercher les données indexées

Dans cette partie, nous allons présenter le diagramme de cas d'utilisation Chercher les données indexées :

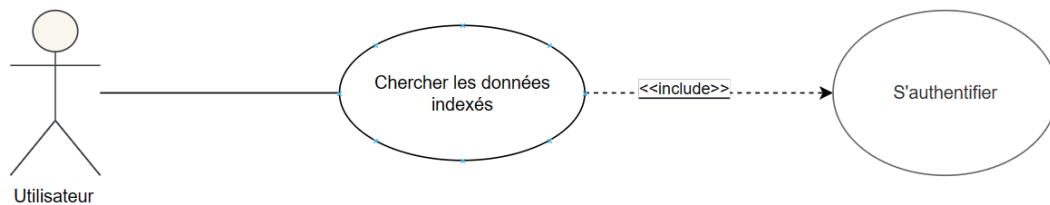


Figure 4.2 – Diagramme de cas d'utilisation Chercher les données indexées

4.1.3 Description textuelle des cas d'utilisation : Sprint 2

Durant cette section nous allons raffiner les cas d'utilisation du sprint 2 en donnant leur description textuelle :

1) Description Textuelle de cas d'utilisation : Créer un modèle élastique « Mappings »

Cas d'utilisation « Créer un modèle élastique 'Mappings' »	
Acteurs	User, Admin, Système
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur utilise une ou plusieurs structures de données indexées dans un mapping et le mettre en prod
Exception : Si le mapping n'est pas en prod la création ne sera pas déclenché.	

TABLE 4.2 – Description textuelle du cas d'utilisation " créer un modèle élastique « Mappings »"

2) Description Textuelle de cas d'utilisation : Chercher les données indexés

Cas d'utilisation « Chercher les données indexé »	
Acteurs	User, Admin, Système
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale Scénario Alternative	<ul style="list-style-type: none"> • L'utilisateur saisit une chaîne de caractères dans le champ de recherche, puis appuie sur Entrée pour effectuer la recherche. Les résultats affichent une liste des champs indexés correspondant à la recherche. • Si le champ est utilisé dans un mapping, nous pouvons le savoir et avoir un bouton "Row Expand" qui nous permet d'afficher les mappings correspondants et les informations associées à ces mappings.
Exception : La recherche effectuée ne correspond à rien.	

TABLE 4.3 – Description textuelle du cas d'utilisation " Chercher les données indexées "

4.1.4 Les diagrammes de séquence de sprint 2 :

Afin de mieux étudier les interactions entre les acteurs et le système selon un ordre chronologique, nous allons présenter les diagrammes de séquence d'analyse du premier sprint qui englobe le module enrichissement du dictionnaire de données.

1) Diagramme de séquence pour la création d'un nouveau référentiel métier :

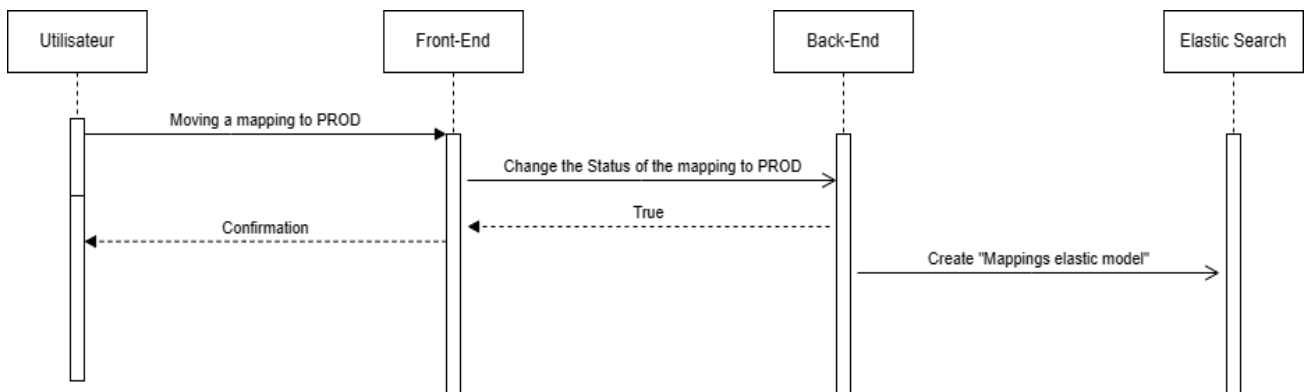


Figure 4.3 – Diagramme de cas d'utilisation création du modèle élastique « Mappings »

2) Diagramme de séquence pour la Recherche des données indexés :

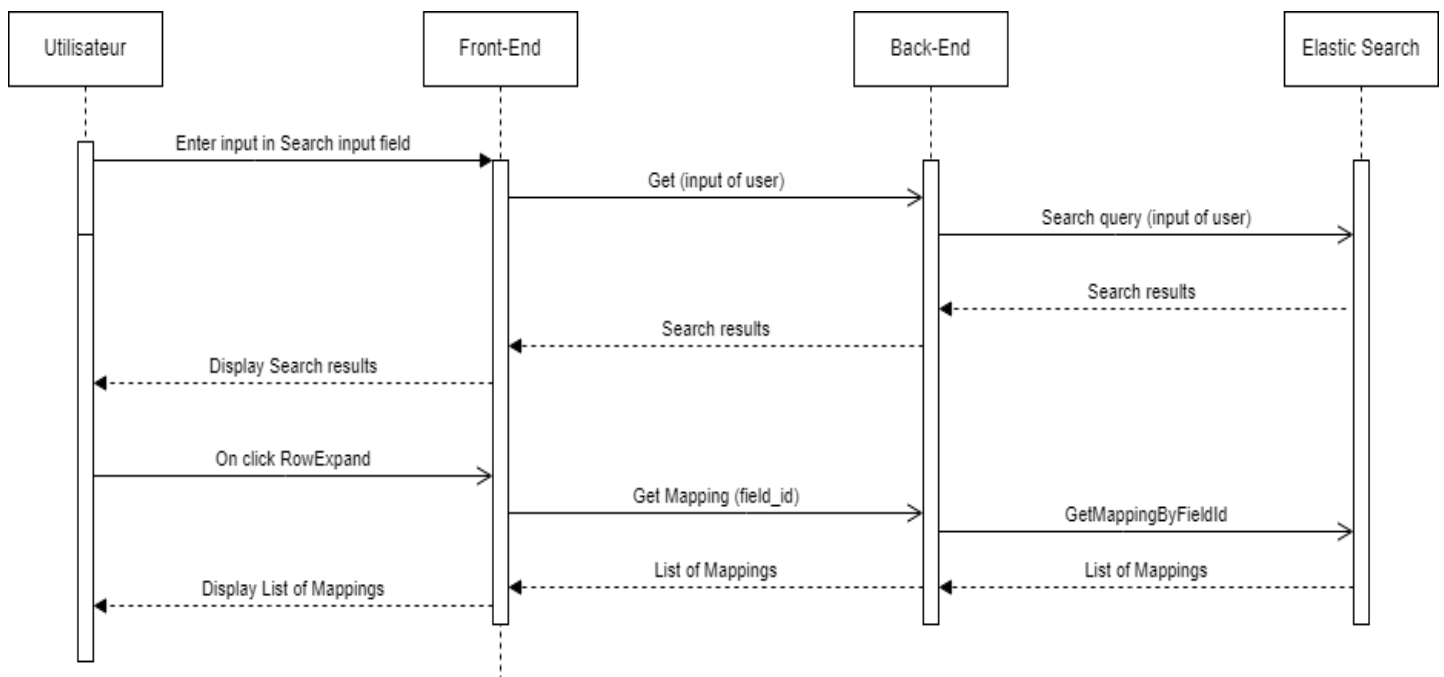


Figure 4.4 – Diagramme de cas d'utilisation " Chercher les données indexées "

4.2 Réalisation

Dans cette partie nous allons présenter le travail déjà réalisé à l'aide des captures d'écrans et des schémas explicatifs des parties fonctionnelles de notre application développée pendant le sprint 2

4.2.1 Réalisation du cas d'utilisation Créer modèle « Mappings »

1) Interface des Flowcharts

Pour que le modèle élastique (Mappings) soit créé, il est nécessaire d'établir une relation de mappage entre les structures d'index et les référentiels métier. Dans ce projet, une capture réelle a été donnée avec une structure appelée "Cruise reservation" et les autres structures sources telles que "Staff cruise" et "Bateau" ont été mentionnées pour illustrer la création du mappage. Cela permet à l'application de déclencher la création du modèle élastique (Mappings) une fois que le projet est mis en production.

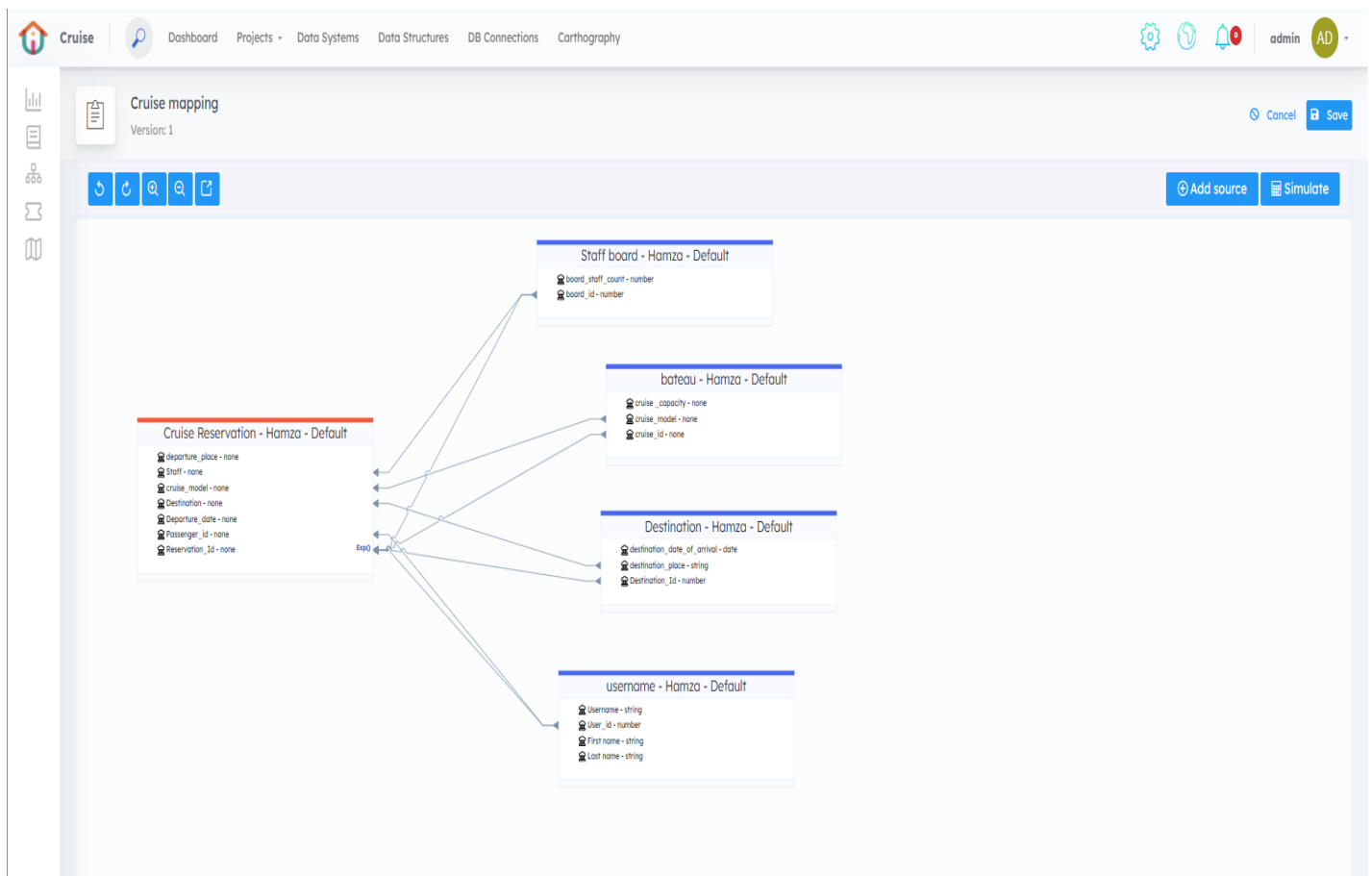


Figure 4.5 – Interface des flowcharts

2) Mettre un mapping en Prod

Cette interface affiche la liste des mapping et les informations associées à chacun, comprenant : nom, version, date de création, créé par et statut.

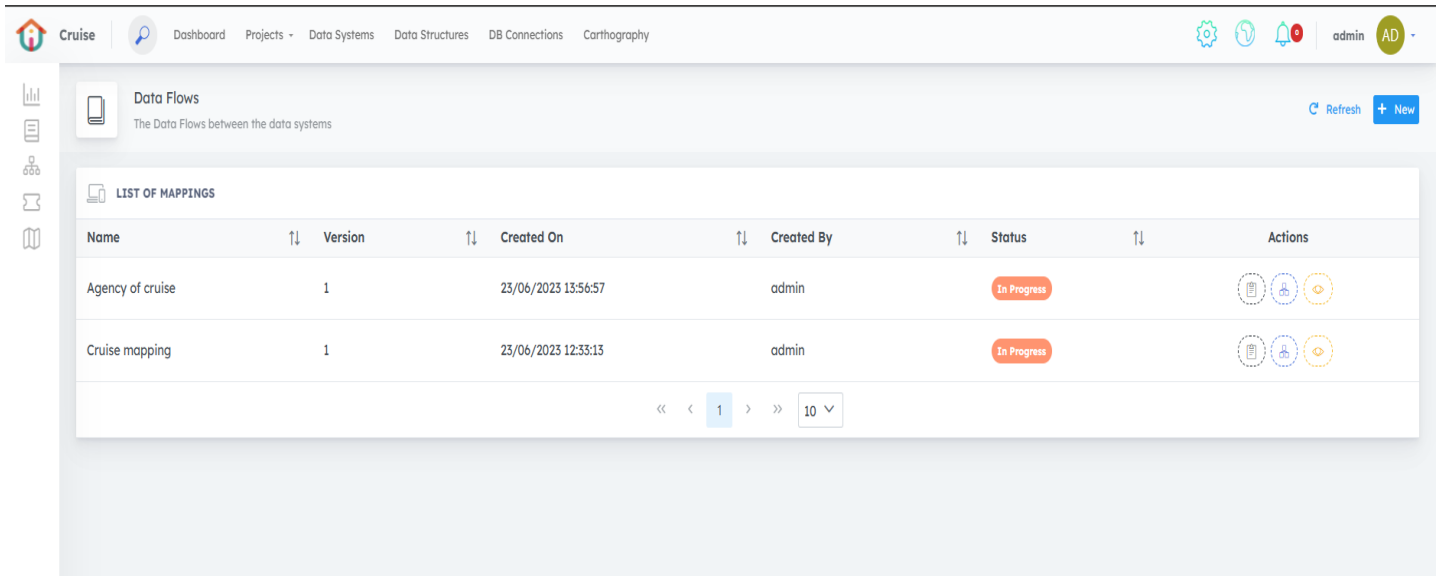


Figure 4.6 – Interface de la liste de mappings

On va changer le statut d'un des deux mappings affiché vers PROD pour qu'on déclenche la création du modèle élastique « Mappings » :

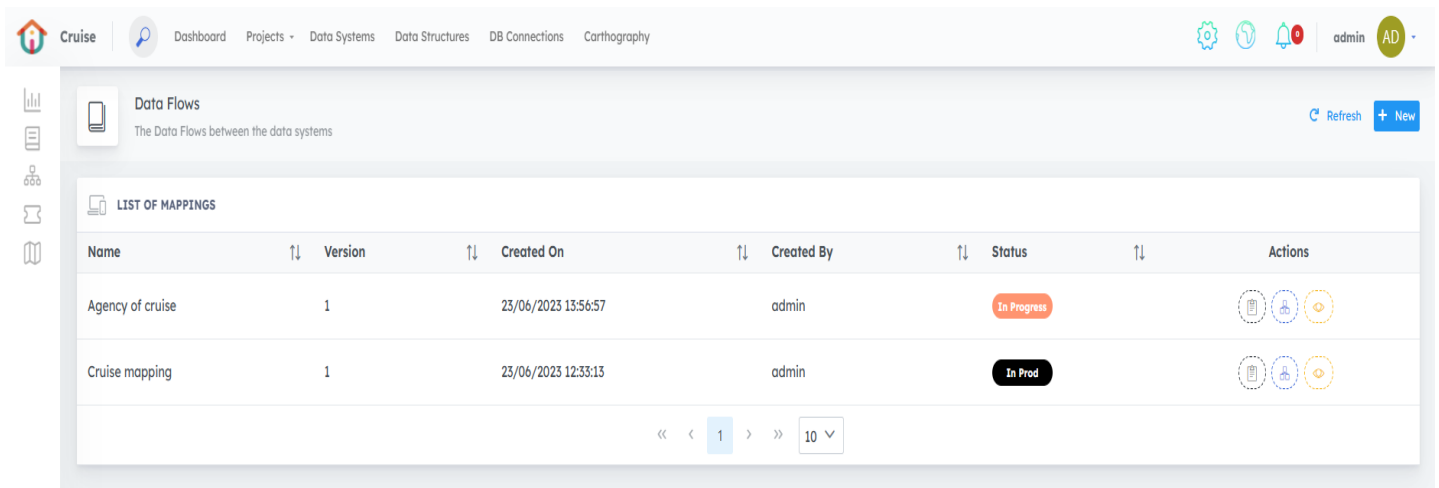


Figure 4.7 – Interface de la liste de mappings 2

3) Visualisation sur Kibana :

Dans cette capture, nous avons basculé vers Kibana pour visualiser les résultats de la création du modèle élastique (Mappings). À gauche, nous pouvons voir les attributs de l'index "Mappings", tandis qu'à droite, nous avons la liste des modèles élastiques créés

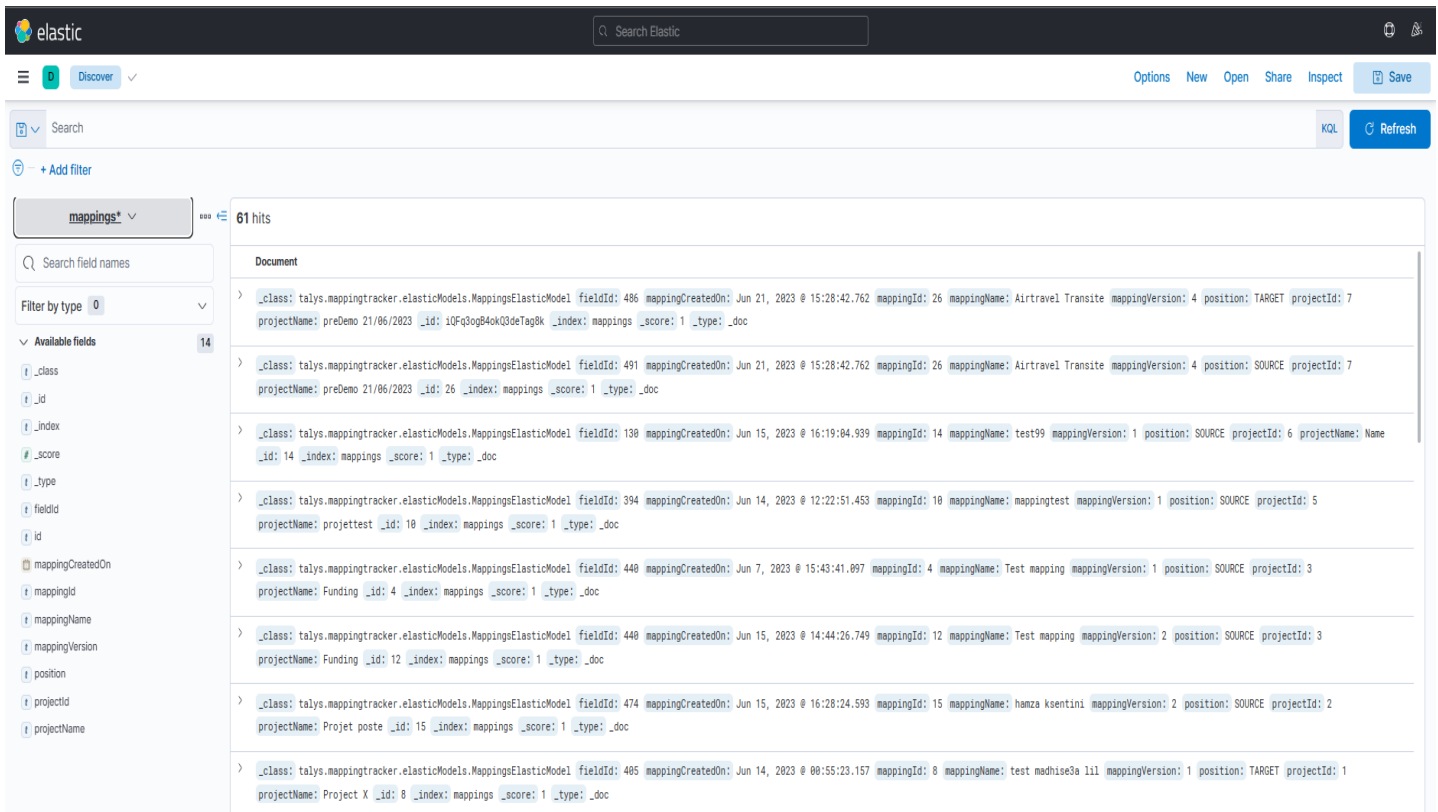


Figure 4.8 – Interface des listes de « Mappings » créés dans Kibana

4.2.2 Realisation de cas d'utilisation : rechercher les données indexées

1) Champ de la recherche :

Ce champ de recherche permet à l'utilisateur de spécifier des critères de recherche en entrant une chaîne de caractères, et ces critères seront utilisés pour interroger les modèles élastiques ("Fields") préalablement créés dans Elasticsearch. Ainsi, l'utilisateur peut effectuer des recherches précises et ciblées pour trouver les résultats correspondants à ses besoins.

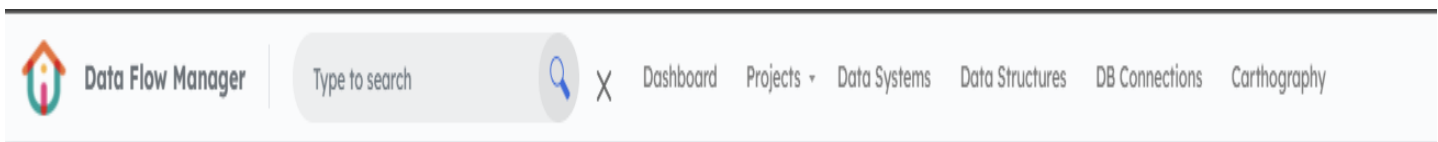
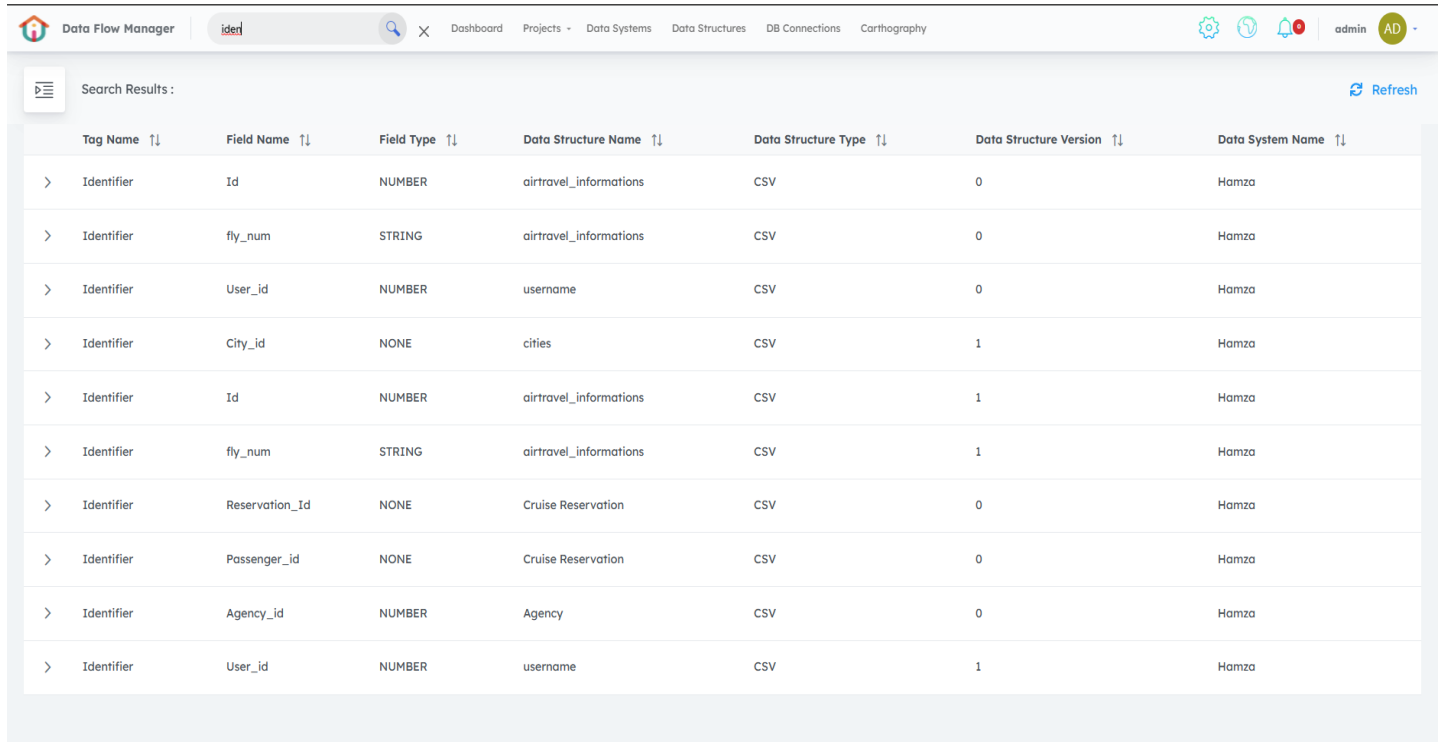


Figure 4.9 – Diagramme de cas d'utilisation Chercher les données indexées

2) L'interface du résultat de recherche :

Cette interface affiche les résultats de la recherche effectuée sur l'index "Fields" dans Elasticsearch. Les résultats incluent les champs qui correspondent à la recherche, ainsi que leurs informations associées telles que le nom du tag, le nom du champ, le type de champ, le nom de la structure de données, le type de structure de données, la version de la structure de données et le nom du système de données. Ces informations fournissent une vue détaillée des champs correspondants, permettant à l'utilisateur de mieux comprendre et d'analyser les résultats de la recherche.



Tag Name	Field Name	Field Type	Data Structure Name	Data Structure Type	Data Structure Version	Data System Name
Identifier	Id	NUMBER	airtravel_informations	CSV	0	Hamza
Identifier	fly_num	STRING	airtravel_informations	CSV	0	Hamza
Identifier	User_id	NUMBER	username	CSV	0	Hamza
Identifier	City_id	NONE	cities	CSV	1	Hamza
Identifier	Id	NUMBER	airtravel_informations	CSV	1	Hamza
Identifier	fly_num	STRING	airtravel_informations	CSV	1	Hamza
Identifier	Reservation_Id	NONE	Cruise Reservation	CSV	0	Hamza
Identifier	Passenger_id	NONE	Cruise Reservation	CSV	0	Hamza
Identifier	Agency_id	NUMBER	Agency	CSV	0	Hamza
Identifier	User_id	NUMBER	username	CSV	1	Hamza

Figure 4.10 – Interface de la Résultats de la recherche

Conclusion

En conclusion, ce chapitre a permis de renforcer notre dictionnaire de données et d'exploiter ses fonctionnalités pour une meilleure utilisation des informations. Le prochain chapitre mettra l'accent sur la création de tableaux de bord pour une visualisation avancée et une analyse approfondie des données du dictionnaire.

Chapitre 5

Sprint 3 : Dashboarding des données

Introduction

Dans ce chapitre, nous allons exposer le troisième Sprint Le Dashboarding des données. Au début, nous allons donner les spécifications et l'analyse des besoins, ensuite, nous allons préparer les diagrammes de conception puis nous allons présenter la réalisation du deuxième sprint avec les captures d'écrans de l'application.

En utilisant Kibana, une plateforme d'analyse et de visualisation de données puissante, nous pouvons créer des tableaux de bord personnalisés qui affichent des informations pertinentes extraites des données indexées.

Grâce à la combinaison d'Elastic Search et de Kibana, nous pouvons regrouper, filtrer et agréger les données pour obtenir des insights précieux. Les fonctionnalités de visualisation avancées de Kibana, telles que les graphiques interactifs, les tableaux de bord dynamiques et les indicateurs clés de performance, permettent de présenter les données de manière visuellement attrayante et facilement compréhensible.

5.1 Spécification et analyse des besoins

5.1.1 Backlog du sprint 3 : Dashboarding des données et la préparation du module administratif

ID	User Story	Taches	Temps esti mé	Fonctionnalités
1	En tant que User / Admin, Je dois pouvoir utiliser les données indexées via Elastic Search dans un Dashboard	Préparer les patterns du dashboard dans Kibana .	1w	<div><div>— Le display du dashboard Kibana dans notre application</div><div>— Pouvoir modifier les dashboard directement sur notre application</div></div>
		Assurer la connexion entre Kibana et notre application	1w	
		Préparer l’interface de l’intégration du Kibana dans notre application	1w	
		Implémenter les requêtes dans une fonction et l’implémenter dans l’application Back et front	1w	
		Préparer l’interface et leur design		

TABLE 5.1 – BACKLOG SPRINT 3

5.1.2 Les Diagrammes de cas d'utilisations : Sprint 2

1) Diagramme de cas d'utilisation : Créer un Dashboard

Dans cette partie, nous allons présenter le diagramme de cas d'utilisation Créer un Dashboard :

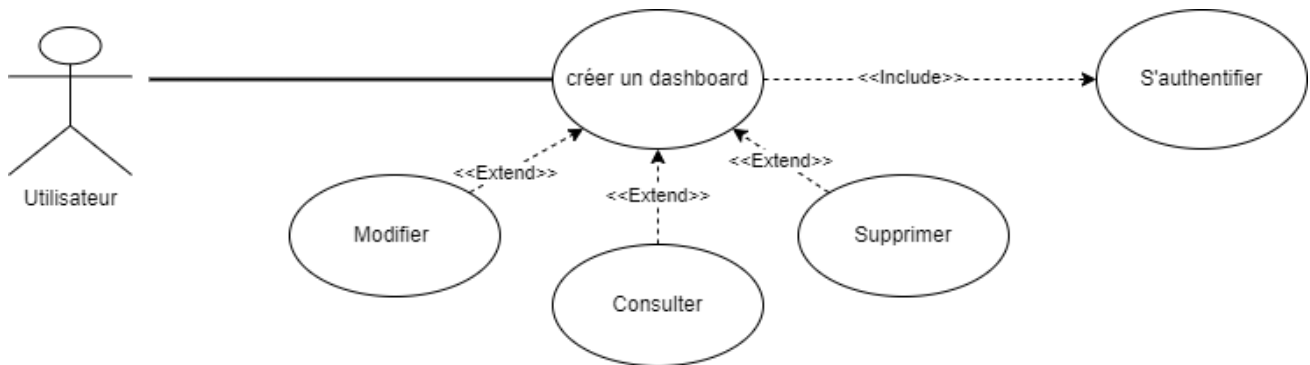


Figure 5.1 – Diagramme de cas d'utilisation Créer un Dashboard

5.1.3 Les Diagrammes de cas d'utilisations : Sprint 3

Durant cette section nous allons raffiner les cas d'utilisation du sprint 3 en donnant leur description textuelle :

1) Description Textuelle de cas d'utilisation : Créer un Dashboard

Cas d'utilisation « Créer un Dashboard »	
Acteurs	User, Admin
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale	<ul style="list-style-type: none"> L'utilisateur utilise notre application pour créer un Dashboard via l'outil Kibana.
Scenario alternative	<ul style="list-style-type: none"> L'utilisateur fait la creation du Dashboard directement du Kibana
Exception : Si Kibana sevice n'est pas lancé le Dashboard ne s'affiche pas	

TABLE 5.2 – Description textuelle du cas d'utilisation " Créer un Dashboard "

2) Description Textuelle de cas d'utilisation : Supprimer un Dashboard

Cas d'utilisation « Supprimer un Dashboard »	
Acteurs	User, Admin
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur utilise notre application pour supprimer un des Dashboard qu'il a créés.
Scénario alternative	<ul style="list-style-type: none"> • L'utilisateur fait la Suppression directement du Sur Kibana
Exception : Si Le Dashboard est créé par une autre personne seule cette personne et l'admin peut le supprimer	

TABLE 5.3 – Description textuelle du cas d'utilisation " Supprimer un Dashboard "

1) Description Textuelle de cas d'utilisation : Consulter un Dashboard

Cas d'utilisation « Consulter un Dashboard »	
Acteurs	User, Admin
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur utilise notre application pour Consulter un des Dashboard
Scénario alternative	<ul style="list-style-type: none"> • L'utilisateur consulte le Dashboard sur Kibana
Exception : Pas d'exception ici	

TABLE 5.4 – Description textuelle du cas d'utilisation " Consulter un Dashboard"

2) Description Textuelle de cas d'utilisation : Modifier un Dashboard

Cas d'utilisation « Modifier un Dashboard »	
Acteurs	User, Admin
Précondition	Utilisateur authentifié
Post-condition	Utilisateur est connecté.
Scénario nominale	<ul style="list-style-type: none"> L'utilisateur utilise notre application pour Modifier un des Dashboard qu'il a créé.
Scenario alternative	<ul style="list-style-type: none"> L'utilisateur fait les modifications de Dashboard sur Kibana directement
Exception : Seule l'admin peut modifier tous les Dashboards.	

TABLE 5.5 – Description textuelle du cas d'utilisation " Modifier un Dashboard "

5.1.4 Les diagrammes de séquence de sprint 3 :

1) Diagramme de séquence de cas d'utilisation : Créer un Dashboard

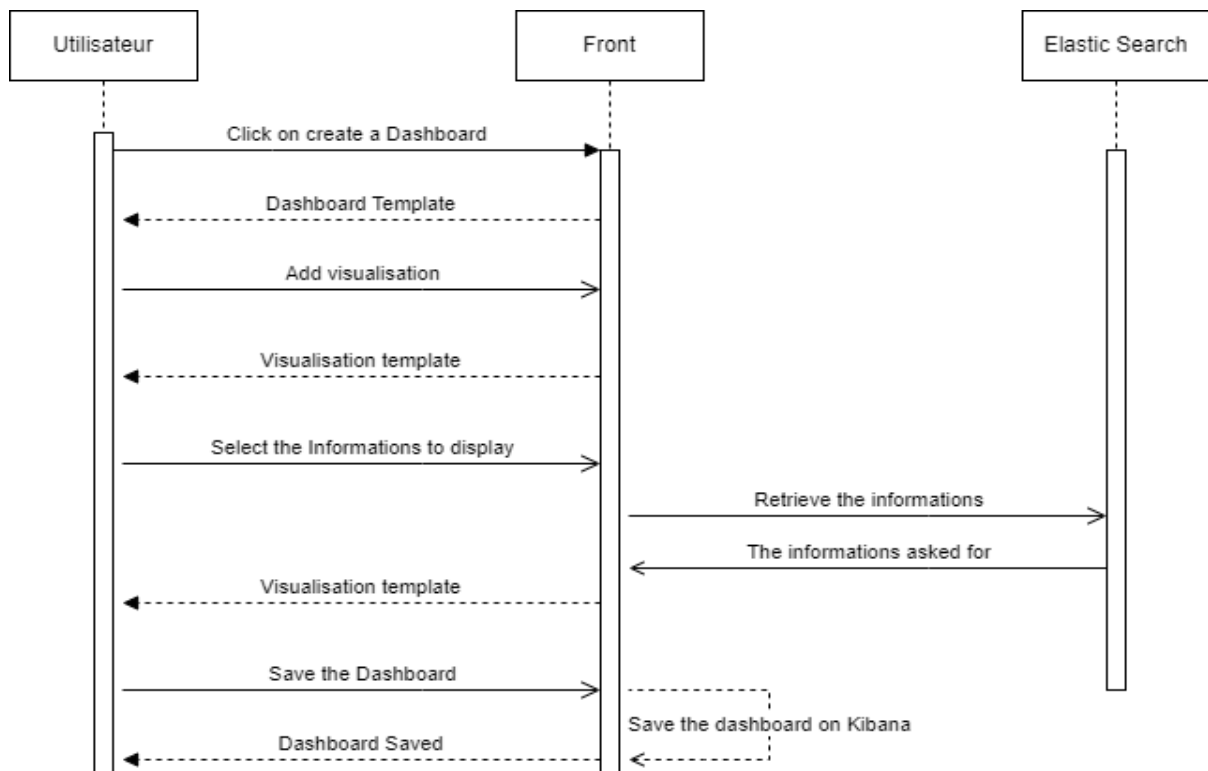


Figure 5.2 – Diagramme de séquence de cas d'utilisation Créer un Dashboard

2) Diagramme de séquence de cas d'utilisation : Supprimer un Dashboard

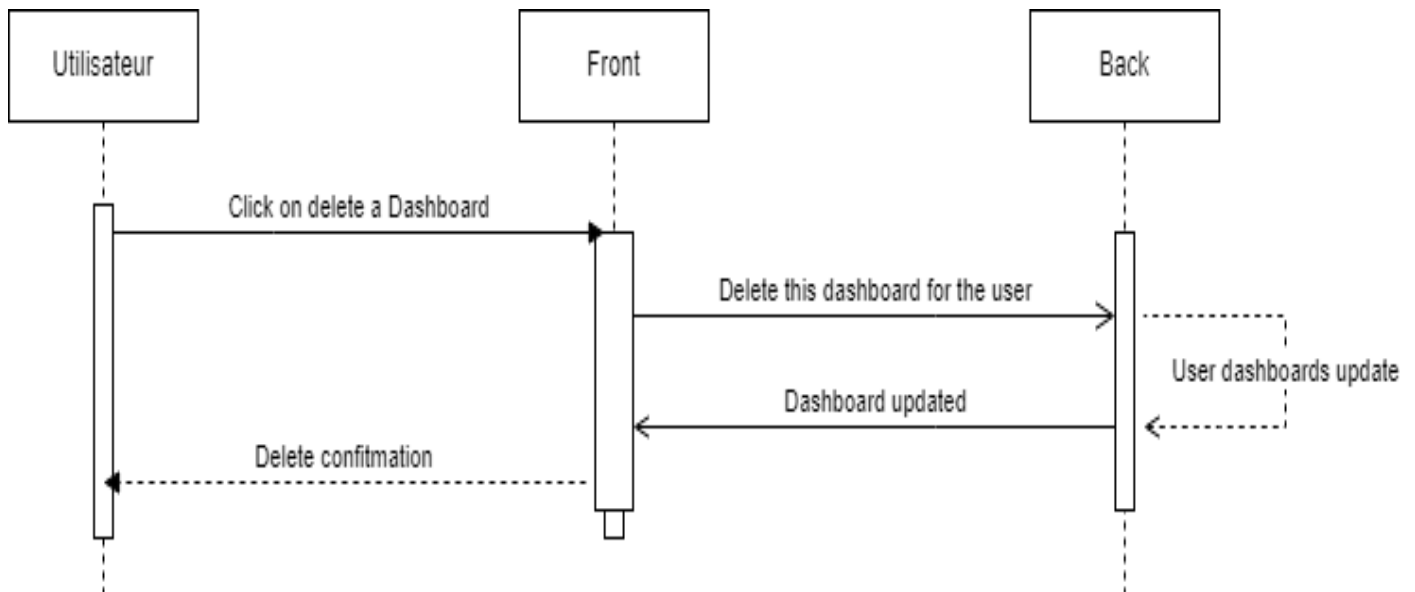


Figure 5.3 – Diagramme de séquence de cas d'utilisation Supprimer un Dashboard

3) Diagramme de séquence de cas d'utilisation : Consulter un Dashboard

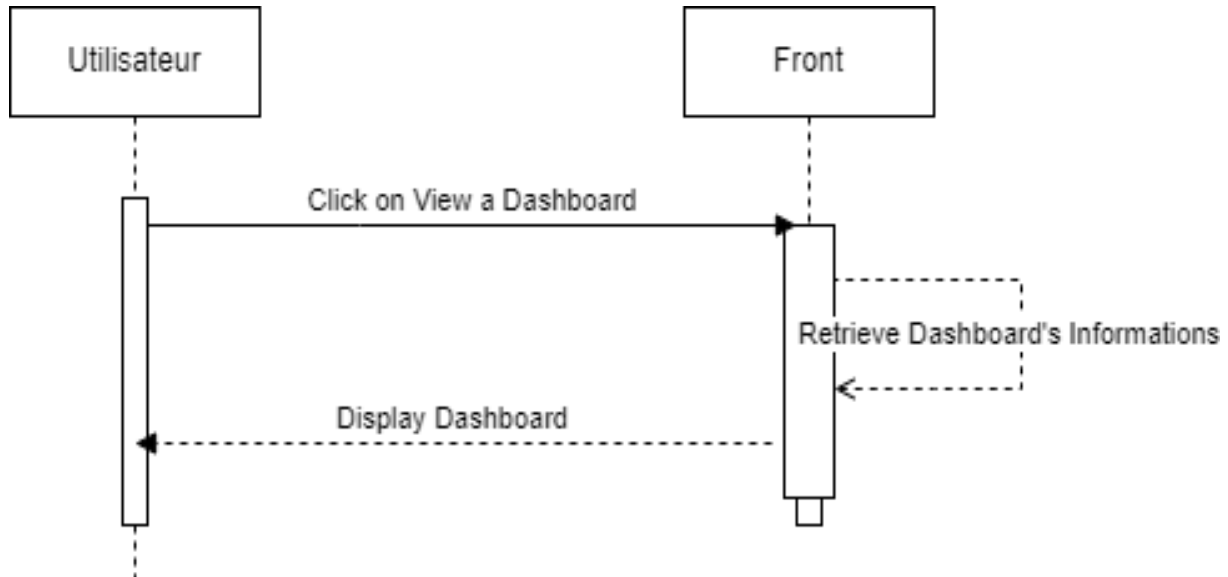


Figure 5.4 – Diagramme de séquence de cas d'utilisation Consulter un Dashboard

4) Diagramme de séquence de cas d'utilisation : Modifier un Dashboard

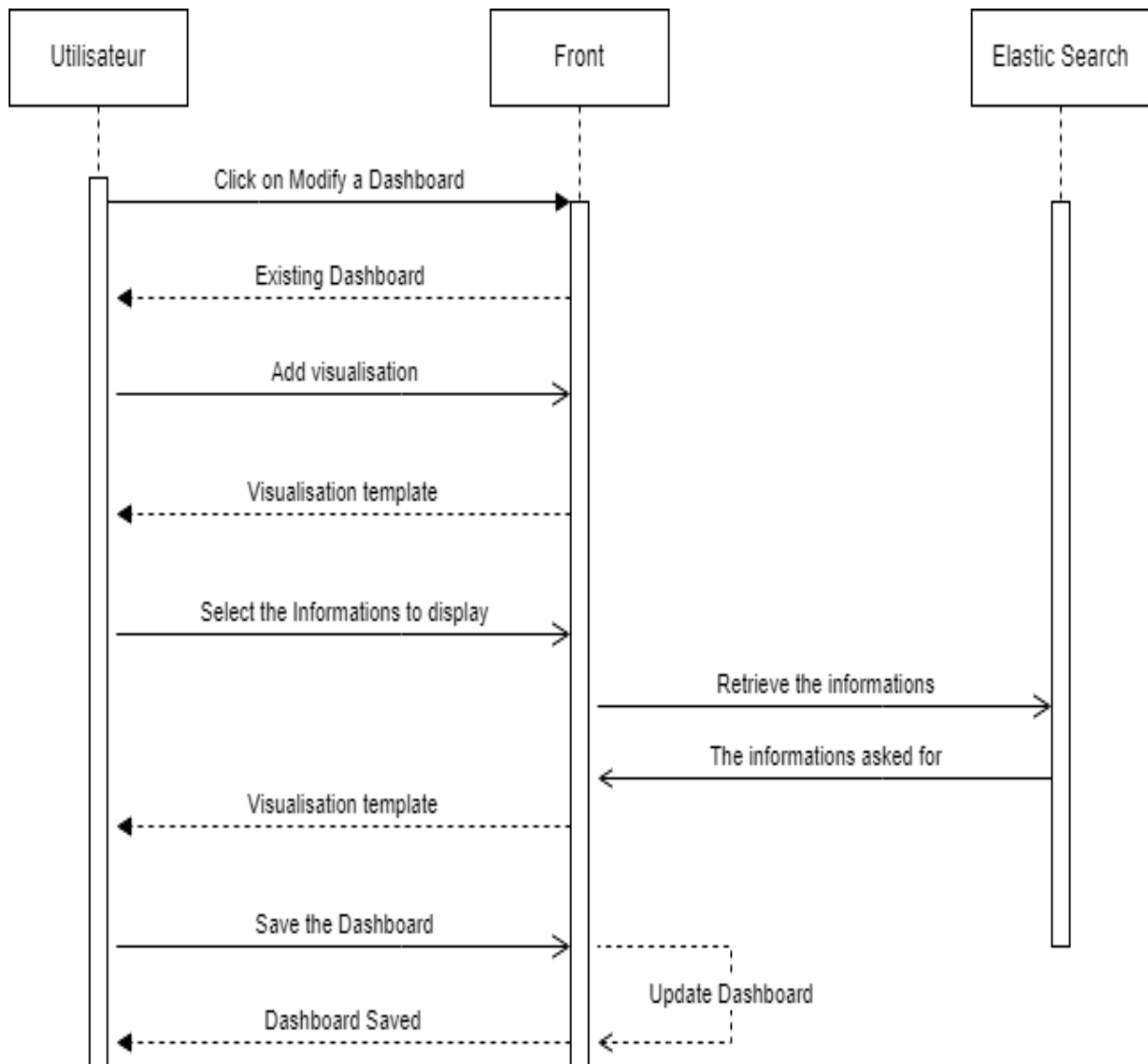


Figure 5.5 – Diagramme de séquence de cas d'utilisation Modifier un Dashboard

5.2 Réalisation

Dans cette partie nous allons présenter le travail déjà réalisé à l'aide des captures d'écrans et des schémas explicatifs des parties fonctionnelles de notre application développée pendant le sprint 3.

5.2.1 Réalisation du Dashboard

1) Interface de Dashboard

Lorsque vous ouvrez notre application, vous serez accueilli par un tableau de bord créé via Kibana. Ce tableau de bord utilise les données indexées dans Elasticsearch comme source de données pour les différentes visualisations. Vous pourrez explorer et interagir avec ces visualisations pour obtenir des informations précieuses et des insights sur vos données.

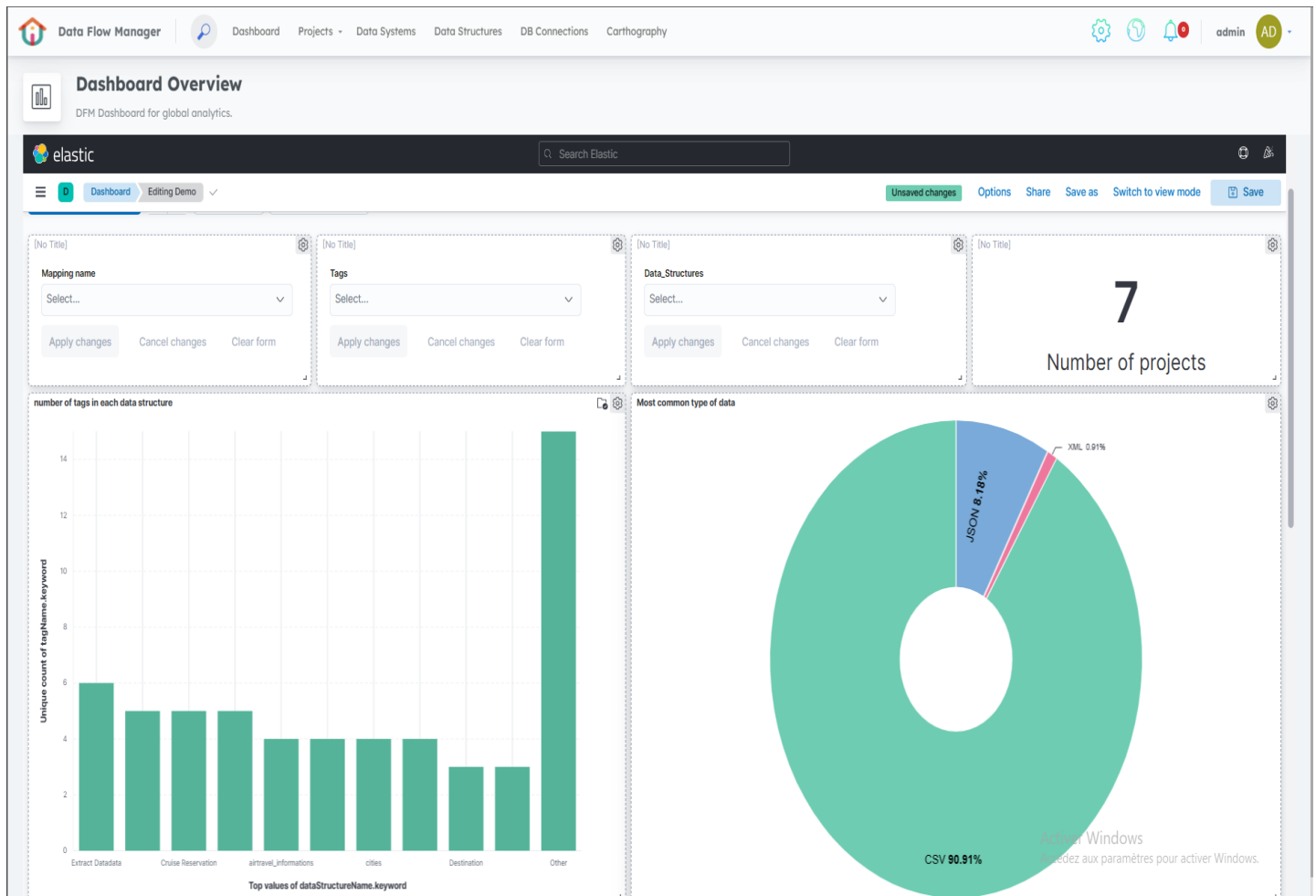


Figure 5.6 – Interface du Dashboard principale 1

2) Interface de Dashboard 2

Dans le même tableau de bord précédent, vous avez utilisé la barre de défilement pour voir davantage d'exemples de visualisations disponibles, offrant ainsi une plus grande variété de choix pour représenter vos données de manière visuellement attrayante et informative.

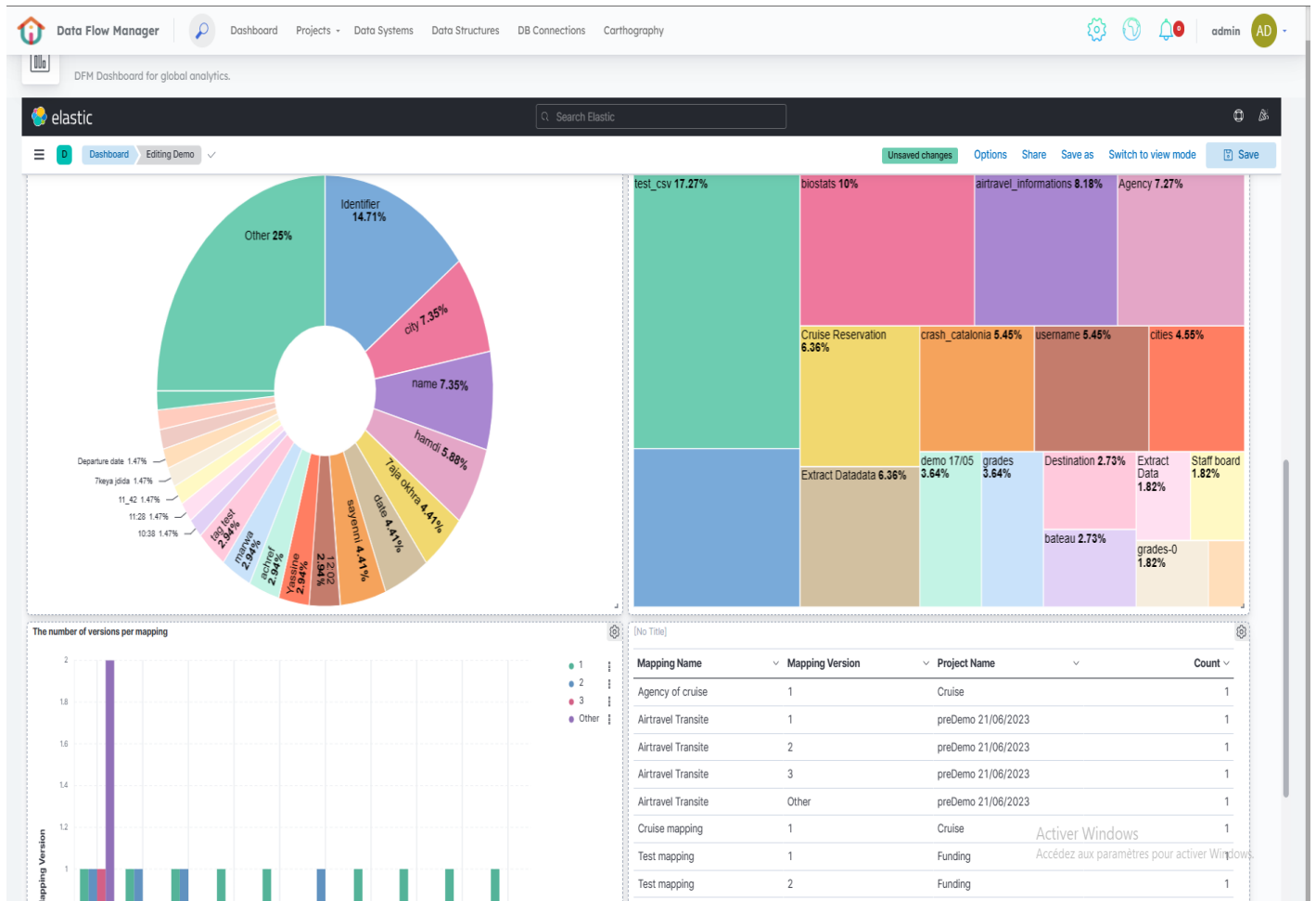


Figure 5.7 – Interface du Dashboard principale 2

3) Interface de Creation d'une Visualisation

En cliquant sur "Create Visualization", vous avez la possibilité de créer une nouvelle visualisation des données. Vous pouvez sélectionner l'index à partir duquel vous souhaitez utiliser les données pour votre visualisation. De plus, Kibana offre une vaste collection d'options de visualisation parmi lesquelles vous pouvez choisir. Vous pouvez explorer différents types de visualisations tels que des graphiques, des diagrammes à barres, des cartes géographiques, des tableaux de bord interactifs, et bien d'autres encore. Cela vous permet de personnaliser et de présenter vos données de manière significative et percutante.

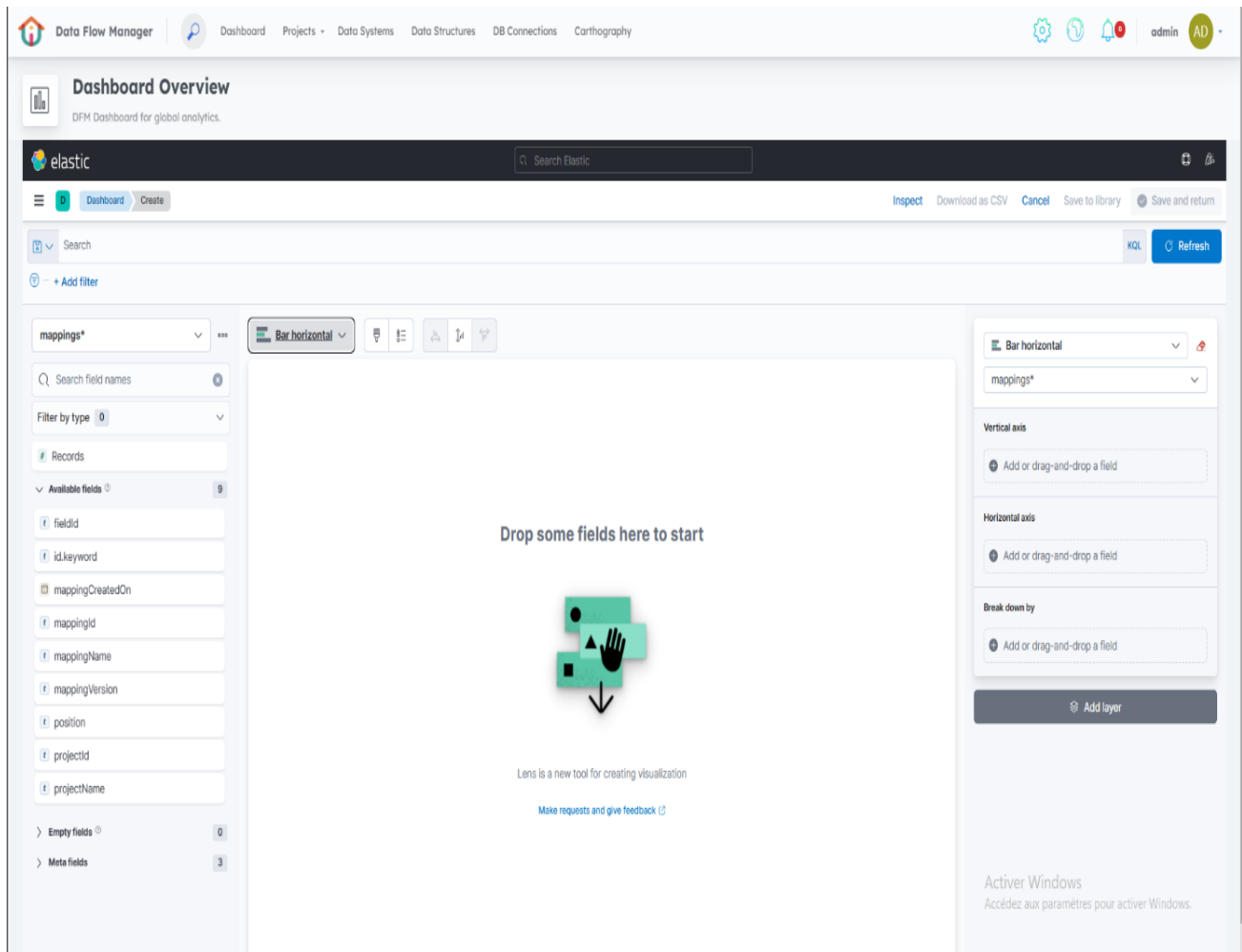


Figure 5.8 – Interface de la création d'une nouvelle visualisation

4) Exemple de Création d'une Visualisation

Dans cet exemple, nous avons utilisé la visualisation "Donut" en sélectionnant l'index "Mappings" comme source d'informations. Nous avons choisi de visualiser le nombre de positions pour chaque mapping. Cette visualisation en forme de donut vous permet de voir la répartition des positions entre les différents mappings de manière claire et concise.

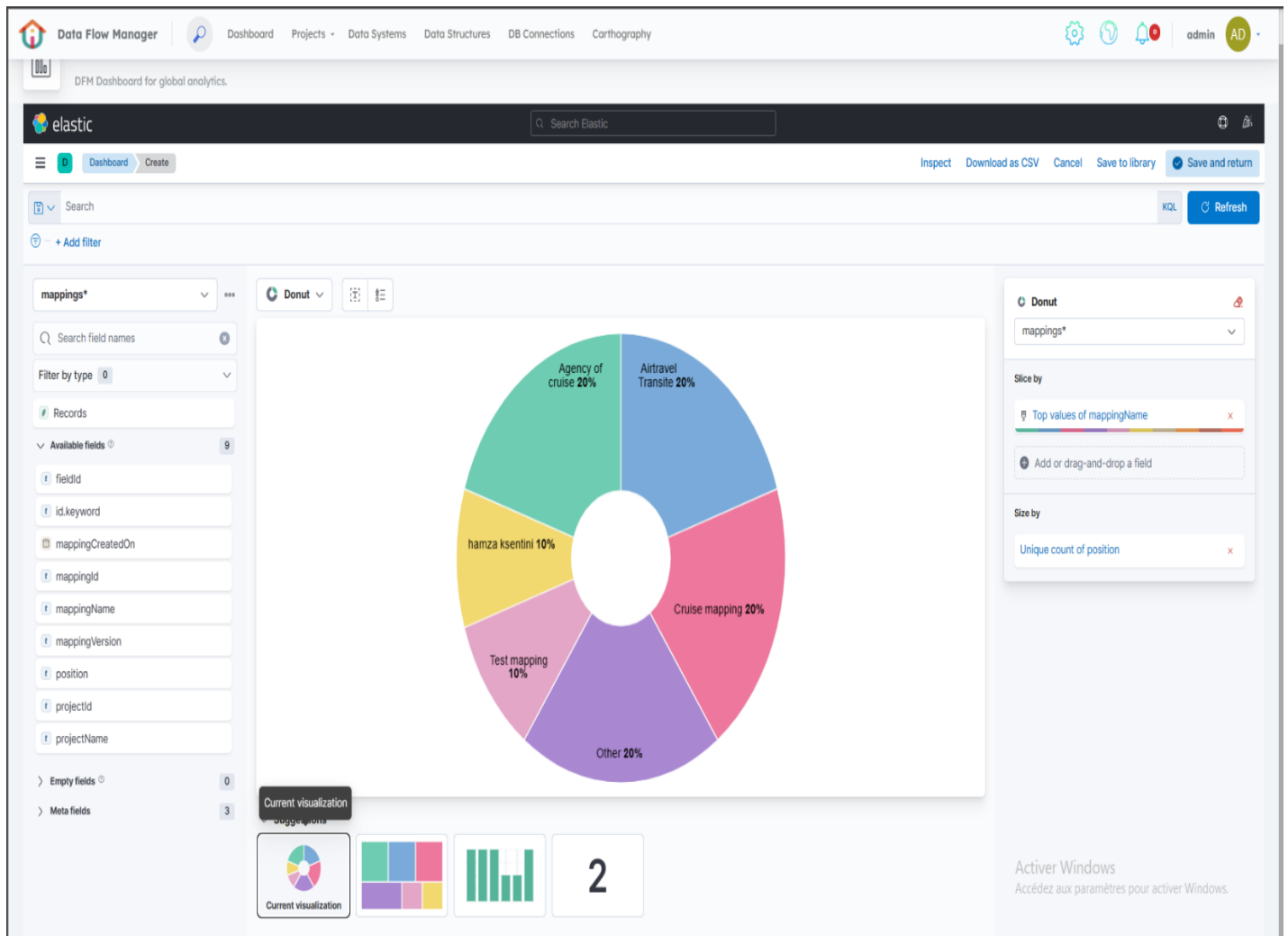


Figure 5.9– Exemple de visualisation des données

Conclusion

En conclusion, ce chapitre a mis en évidence l'importance du dashboarding des données et l'utilisation de Kibana pour créer des tableaux de bord personnalisés. Grâce à la combinaison d'Elasticsearch et de Kibana, nous avons pu regrouper, filtrer et visualiser nos données de manière efficace. Les fonctionnalités avancées de visualisation offertes par Kibana nous ont permis de présenter les informations de manière attrayante et facilement compréhensible. Ce troisième sprint a été essentiel pour exploiter pleinement nos données et prendre des décisions éclairées.

Conclusion générale et perspective :

Conclusion générale

L'application DFM (Data Flow Management) actuellement utilisée en version Beta Testing par les clients de la société "Talys Group" offre une solution complète pour la gestion et la traçabilité des flux de données au sein de l'entreprise. Les différents modules de l'application, tels que la gestion des utilisateurs, la gestion des connexions, la gestion des flux de données, l'automatisation des tâches, la cartographie des flux de données et la gestion des mappings, permettent d'assurer une gestion efficace et sécurisée des données.

Cependant, malgré ses fonctionnalités avancées, l'application DFM présente certaines limites. L'absence d'un module de dictionnaire de données peut entraîner des erreurs de synchronisation et une perte de temps pour les utilisateurs. De plus, la diversité des nomenclatures utilisées dans les différents systèmes de l'entreprise rend la recherche d'informations précises difficile. L'indexation des données est également limitée, ce qui rend l'exploration et l'analyse des données plus complexes.

Pour remédier à ces limitations, notre solution propose l'ajout d'un module de dictionnaire de données, la mise en place d'un référentiel métier pour standardiser les informations, et l'intégration d'un système de dashboarding pour une présentation claire et intuitive des données. Ces améliorations permettront une meilleure traçabilité des données, une standardisation des informations, une recherche plus efficace, une prise de décision éclairée et une amélioration globale de l'efficacité de la gestion des flux de données.

En adoptant cette solution, l'entreprise pourra bénéficier d'une gestion des flux de données plus robuste, d'une meilleure collaboration entre les différents départements, d'une réduction des erreurs et des conflits, ainsi que d'une meilleure sécurité des données.

En conclusion, l'application DFM, avec l'ajout des améliorations proposées, représente une solution complète et performante pour la gestion et la traçabilité des flux de données au sein de l'entreprise "Talys Group". Elle contribuera à optimiser les processus métier, à améliorer l'efficacité opérationnelle et à prendre des décisions éclairées basées sur des données fiables et précises.

Perspective

En outre, notre projet présente également des perspectives d'évolution prometteuses. Nous envisageons d'intégrer un module d'intelligence artificielle qui assisterait les utilisateurs dans la création des "Mappings". Cette fonctionnalité permettrait d'automatiser une partie du processus de gestion des flux de données, rendant ainsi l'application encore plus efficace et intuitive.

Par ailleurs, nous prévoyons de séparer les services de notre application actuellement monoservice en deux versions distinctes. La première version serait une version de base sans l'intégration d'Elasticsearch, tandis que la deuxième version inclurait Elasticsearch pour améliorer l'indexation et faciliter l'exploration et l'analyse des données.

Webographie

- [1] <https://www.supinfo.com/articles/single/2912-pourquoi-utiliser-methode-scrum-votre-projet>.
- [2] [Java Archive Downloads - Java SE 8 \(oracle.com\)](#)
- [3] [Spring | Tools](#)
- [4] [Maven – Welcome to Apache Maven](#)
- [5] [Spring Boot](#)
- [6] [Spring Data JPA](#)
- [7] [Hibernate. Everything data.](#)
- [8] [Découvrez Elastic, l'entreprise à qui l'on doit Elasticsearch et Kibana | Elastic](#)
- [9] [Kibana : Exploration, visualisation et découverte des données | Elastic](#)
- [10] [StarUML](#)
- [11] [Angular](#)
- [12] [NGX-Translate: The i18n library for Angular 2+](#)
- [13] [Visual Studio Code - Code Editing. Redefined](#)
- [14] [Node.js \(nodejs.org\)](#)
- [15] [npm \(npmjs.com\)](#)
- [16] [Angular - CLI Overview and Command Reference](#)
- [17] [Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](#)
- [18] [TypeScript: JavaScript With Syntax For Types. \(typescriptlang.org\)](#)
- [19] [HTML Tutorial \(w3schools.com\)](#)
- [20] [Sourcetree | Free Git GUI for Mac and Windows \(sourcetreeapp.com\)](#)
- [21] [Bitbucket | Git solution for teams using Jira](#)
- [22] [Jira | Issue & Project Tracking Software | Atlassian](#)