

Mémoire de Projet de Fin d'Etudes

Pour l'Obtention du Diplôme Ingénieur d'état en

Ingénierie des Systèmes d'Information

Sujet

Amélioration d'un module de pilotage
opérationnel, la conception et développement d'un
module de pilotage financier des projets

Soutenu par :

Ibrahim ZEROUAL

Sous la direction de :

Mr. Mounir OMARI

Mr. Oussama AMARA

Année Universitaire 2022-2023

Dédicaces

A mes chers parents

Grâce à votre soutien, amour, et sacrifice, que je suis aujourd'hui, à la fin de mon parcours académique. J'espère que vous pouvez trouver dans ce travail un témoignage d'amour et d'affection envers vous. Que le tout puissant vous accorde une bonne santé et une vie heureuse.

A mes chers frères

Vous êtes source de joie et de fraternité. Vous me donnez toujours le bon exemple à suivre, et le vrai soutien dans les moments difficiles. Que Dieu préserve notre rassemblement.

A mes chers amis

Je vous dédie ce travail en témoignage de l'amitié qui nous unie, et les souvenirs des moments que nous avons passés ensemble.

A mes professeurs et enseignants

Depuis le primaire, jusqu'au cycle d'ingénierie, je vous remercie pour toutes les connaissances que vous m'avez apprises et pour chaque précieux conseil.

A mes collègues

Qui m'ont montré le respect, la serviabilité, et la confiance

Remerciements

En premier lieu, Je tiens à remercier Allah le tout puissant de m'avoir donné la santé, le courage et la foi pour réaliser ce travail avec volonté.

Nos encadrants Mme. Asmae El Ghazi et Mr. Mounir OMARI pour leurs aides et conseils précieux qui m'ont aidé dans mon travail et m'ont montré les points importants sur lesquels je dois me concentrer lors de mes séances d'encadrements.

J'exprime mes vifs remerciements à mes encadrants externes M. SAROUI Abdelaziz Expert Technique, et M. AMARA Oussama, Manager et Expert Technique à Novelis, pour m'avoir encadré et encouragé tout au long de ce projet, et prodigué ses directives précieuses et ses conseils pertinents qui m'ont été d'un appui considérable dans ma démarche. Je tiens également à remercier tous les membres de l'équipe Novy pour m'avoir écouté et conseillé durant le processus de développement.

Je remercie aussi ma famille pour leur grande attention, leur grand soutien et encouragement tout au long de l'accomplissement de ce travail en particulier, mais aussi tout au long de mon cursus académique.

Enfin, j'exprime ma grande gratitude envers les membres du jury qui évalueront ce travail qui me tient à cœur.

Résumé

Le présent document constitue la synthèse du travail réalisé dans le cadre de projet de fin d'études et qui a pour objectif d'améliorer Novy, une plateforme qui regroupe plusieurs modules, chacun est dédié à un aspect dans la gestion de l'entreprise (CRM, Recrutement, Collaborateurs, Paie, etc.) et d'améliorer et développer le module pilotage des projets de l'entreprise.

Ce module ainsi nommé Pilotage a pour but de faciliter aux responsables et chefs de projets la supervision et le suivi de l'évolution des indicateurs opérationnels et financiers des projets,

Aussi les actions et les risques opérationnels pour être constamment en conscience de l'écart existant entre ce qui est prévu et ce qui est réalisé.

Pour la réalisation de ce projet, une étude est primordiale pour découvrir le besoin attendu. Puis vient l'étape de l'analyse et l'amélioration de l'existant, et ensuite l'élaboration des spécifications et le développement du module en suivant une architecture micro-services. La méthodologie de développement appliquée est itérative et incrémentale.

Les améliorations et du module ont été réalisés en utilisant la même pile technique utilisée dans les autres modules de la plateforme. Néanmoins, plusieurs choix à faire dans l'aspect technique ainsi que l'aspect fonctionnel ont été nécessaires pour mener à bien le projet.

Mots-clés : Gestion d'entreprises, Pilotage des projets, Amélioration, Architecture micro-services, Java/JEE, Agile, Spring, Kafka, React, Redux.

Abstract

This document is a summary of the work carried out as part of an end-of-studies project aimed at improving Novy, a platform that includes several modules, each dedicated to a specific aspect of company management (CRM, Recruitment, Employees, Payroll, etc.), and enhancing and developing the project management module of the company.

This module, called Pilotage, aims to facilitate the supervision and monitoring of the operational and financial indicators of projects for managers and project leaders. It also tracks operational actions and risks to constantly be aware of the gaps between planned and actual results.

For the implementation of this project, a preliminary study is essential to understand the expected requirements. This is followed by the analysis and improvement of the existing system, and then the development of specifications and the module using a microservices architecture. The applied development methodology is iterative and incremental.

The module's enhancements were implemented using the same technical stack as the other modules in the platform. However, various technical and functional choices had to be made to successfully complete the project.

Keywords : Business management, Project management, Enhancement, Micro-services architecture, Java/JEE, Agile, Spring, Kafka, React, Redux

Liste des abréviations

API	Application Programming Interface
BFF	Backend For Frontend
CAVA	Chiffre d'affaires À Valeur Ajoutée
CD	Continuous Deployment
CI	Continuous Integration
CRM	Customer Relationship Management
CRUD	Create, Read, Update, Delete
DAO	Data Access Object
DNS	Domain Name System
DTO	Data transfer object
HTTP	Hypertext Transfer Protocol
IAM	Identity and Access Management
IP	Internet Protocol
JSON	JavaScript Object Notation
MBth	Marge Brute Théorique
OCR	Optical Character Recognition
POJO	Plain Old Java Object
R&D	Research and Development
RBAC	Role Based Access Control
REST	Representational State Transfer
RH	Ressources Humaines
RPA	Robotic Process Automation
SaaS	Software as a Service
UX/UI	User Experience /User Interface

Table des figures

Figure 1 : Novelis en chiffres	14
Figure 2 : Partenaires en RPA et OCR.....	17
Figure 3 : suivi opérationnel.....	27
Figure 4 : suivi financier	28
Figure 5 : Diagramme de GANTT	31
Figure 6 : Diagramme de contexte statique.....	36
Figure 7 : Diagramme de cas d'utilisation.....	37
Figure 8 : Diagramme de séquence « Gestion des actions »	39
Figure 9 : Diagramme de séquence « Gestion des risques »	40
Figure 10 : Diagramme de class « Gestion des actions »	41
Figure 11 : Diagramme de class « Gestion des risques »	42
Figure 12 : Diagramme de class « Gestion des commandes »	43
Figure 13 : MR de module pilotage du projet	44
Figure 14 : Architecture micro-service de base	46
Figure 15 : Découverte de micro-services.....	48
Figure 16 : Architecture globale du module Pilotage	51
Figure 17 : Schéma de synchronisation de données.....	55
Figure 18 : d'authentification par Keycloak	56
Figure 19 : interface globale des actions	63
Figure 20 : ajout d'une action	64
Figure 21 : vue en Kanban des actions.....	65
Figure 22 : détail d'action dans le vue kanban.....	65
Figure 23 : interface globale des risques	66
Figure 24 : les actions d'un risque	67
Figure 25 : les actions d'un risque	67
Figure 26 : commentaire de clôture.....	68
Figure 27 : La vue Historique des snapshots.....	68
Figure 28 : les indicateurs actuels d'un projet	69
Figure 29 : la vue des commandes d'un projet	70

Figure 30 : la vue des échéanciers d'une commande.....	70
Figure 31 : l'ajout d'une commande.....	71
Figure 32 : l'ajout d'un échéancier.....	72
Figure 33 : vue administration (Type de vente)	72

Liste des tableaux

Tableau 1 : Historique de Novelis	15
Tableau 2 : Table de GANTT.....	29
Tableau 3 : Novy-acteurs	34
Tableau 4 : cas d'utilisation (gestion des actions).....	38
Tableau 5 : cas d'utilisation (gestion des risques)	38
Tableau 6 : Structure du projet Backend	52

Table des matières

Dédicaces	2
Remerciements	3
Résumé	4
Abstract	5
Liste des abréviations.....	6
Table des figures.....	7
Table des matières	10
Introduction générale.....	12
Chapitre 1.....	13
Contexte Général.....	13
1.1 Organisme d'accueil.....	14
1.1.1 Historique	14
1.1.2 Ses valeurs.....	15
1.1.3 Secteur d'activité et structure interne	17
1.2 Définition du projet	21
1.2.1 Cadre général du projet	21
1.3 Problématique.....	23
1.4 Objectifs	23
1.4.1 Amélioration de suivi opérationnel d'un projet	23
1.4.2 Suivi financier d'un projet.....	24
1.5 Besoins fonctionnels	25
1.5.1 Amélioration de suivi opérationnel d'un projet	25
1.5.2 Suivi financier d'un projet.....	27
1.6 Besoins non fonctionnels	28
1.7 Diagramme de GANTT.....	29
Chapitre 2.....	32
Analyse et conception.....	32
1.1 Présentation générale d'UML	33

1.2	Diagramme de cas d'utilisation.....	33
1.2.1	Identification des acteurs.....	33
1.3	Diagramme de séquences	39
1.4	Diagramme de classes	40
1.5	Conclusion.....	44
Chapitre 3.....		45
Mise en œuvre.....		45
1.1	Architecture Micro-services	46
1.1.1	Registre de service	47
1.1.2	Services	49
1.1.3	Problèmes avec une application monolithique.....	50
1.2	Architecture globale du module pilotage	50
1.2.1	Application web	51
1.2.2	API Backend	51
1.2.3	BFF (Backend For Frontend)	52
1.2.4	Services d'Apache Kafka	53
1.2.5	Service de synchronisation de données	54
1.2.6	Authentification.....	56
1.3	Environnement et outils de développement	57
1.3.1	Backend	57
1.3.2	Frontend	59
1.3.3	Bases de données.....	59
1.3.4	Authentification et autorisation	60
1.3.5	DevOps.....	60
1.3.6	Outils de développement.....	62
1.4	Présentation de l'application	62
1.4.1	Améliorations « module pilotage coté opérationnel »	62
1.4.2	Nouvelles fonctionnalités « module pilotage côté financier »	69
1.5	Conclusion.....	73
Conclusion Générale		74
Modèle Bibliographie.....		75

Introduction générale

Survivre dans ce monde concurrentiel dépend en grande partie de la démarche et la stratégie suivie pour piloter les performances de son entreprise. Le pilotage des projets améliore la qualité et la productivité des équipes, et est devenu une nécessité pour exceller dans son domaine, surtout dans le domaine informatique. C'est dans ce sens, que ce projet a été initié.

Le projet de fin d'études, effectué au sein du groupe Novelis, est autour de l'amélioration et la maintenance applicative de Novy, une plateforme de gestion interne composée de plusieurs modules comme la gestion des ressources humaines, de la paie, des collaborateurs et du CRM, et l'amélioration du module pilotage coté opérationnel et le développement d'un nouveau module permettant de piloter les projets côté financier d'entreprises en se basant sur les autres services faisant partie de la plateforme Novy.

La solution développée, ainsi intitulée Novy Pilotage, permet aux responsables de projets de bien suivre l'avancement des projets et des collaborateurs en se basant sur un ensemble d'indicateurs opérationnels et financiers.

Le présent document constitue un rapport exhaustif de l'ensemble des tâches que j'ai réalisé avec l'équipe responsable du projet. Il contient également une synthèse des réunions et débats effectuées afin de mettre en place une meilleure solution. Il est organisé en trois chapitres :

- Le premier chapitre aborde le contexte général du projet en mettant le point sur l'organisme, d'accueil, le cadre général du projet, ces objectifs et ces besoins.
- Le deuxième chapitre intitulé « Analyse et conception », dans ce chapitre je vais présenter les différents diagrammes que j'ai réalisés en utilisant UML.
- Le troisième chapitre intitulé « Mise en œuvre/Réalisation » est consacrée à la présentation de l'architecture de projet, des outils du travail et les interfaces graphiques les plus significatives de l'application.

Chapitre 1

Contexte Général

Ce premier chapitre a pour but de décrire le contexte général du projet proposé et déroulé au sein de Novelis. Dans la première section, je commence par donner un bref aperçu sur l'organisme d'accueil, son organisation et son domaine d'activité. Ensuite, je présente le projet, sa problématique et ses objectifs. Puis, je finis avec la démarche suivie pour la gestion du travail.

1.1 Organisme d'accueil

1.1.1 Historique

Novelis est un cabinet de conseil, fondé par des passionnés, autour de valeurs partagées et avec la volonté de s'inscrire dans une approche tournée résolument vers la recherche et l'innovation. La figure 1 contient quelques chiffres clés de Novelis.



Figure 1 : Novelis en chiffres

Nous vivons dans un monde marqué profondément par les effets de la transformation digitale. Le processus ne fait que commencer, et il est irréversible. Cette mutation n'est pas sans impact sur les entreprises, bien au contraire. Elle conditionne leurs stratégies et leurs compétitivités, dans un monde de plus en plus globalisé.

Novelis propose des technologies de pointe, comme l'intelligence artificielle et la robotisation, au service des enjeux métier des clients. Faire de l'innovation technologique un levier de croissance et d'efficacité opérationnelle est un objectif essentiel. Le laboratoire R&D, qui accueille des docteurs spécialisés, est le moyen pour Novelis d'être un acteur de cette période riche qui va transformer nos vies et ouvrir des perspectives importantes pour les entreprises.

Depuis 2017, Novelis a pu réaliser un grand succès dans le marché grâce à la qualité de son travail et l'expertise de ses collaborateurs. Aujourd'hui, ses racines s'étendent au Maroc, à l'Espagne, et en France.

Le tableau 1 liste les événements marquants dans l'historique de Novelis :

Année	Evènement
2017	Création du cabinet
2018	Création d'une première agence au Maroc
2018	Création d'une Business Unit spécialisée dans les technologies d'automatisation et d'efficacité opérationnelle
2019	Création d'une deuxième agence au Maroc et d'une antenne à Madrid
2019	Arrivée du 50ème collaborateur
2020	Arrivée de nouveaux associés et ouverture de la société à l'actionnariat salarié
2021	Lancement du Smart Roby, Reconnu comme la meilleure solution d'entreprise de l'année par Blue Prism

Tableau 1 : Historique de Novelis

1.1.2 Ses valeurs

Novelis est la résultante d'une longue genèse, qui s'est intensément concentrée sur les valeurs qui structurent l'entreprise. Les valeurs sont le ciment qui permet de construire, progresser et traverser les moments difficiles sans dévier de la trajectoire. Elles permettent de construire un lien qui va au-delà de la simple collaboration, elles permettent de donner un sens à la stratégie de l'entreprise et à chacun de se l'approprier.

Les valeurs fondamentales de Novelis sont les suivantes :

• **Le Collectif** : est la première valeur fondatrice de Novelis. Dans bien des organisations, la concurrence entre les enjeux collectifs et individuels constitue un frein sérieux au développement, à la croissance et à l'épanouissement de nouvelles idées. Il s'agit là d'une réaction 'humaine' et court-termisme. Or, la réussite des enjeux collectifs est le meilleur tremplin pour atteindre et surpasser les enjeux individuels. La notion de Collectif est un vrai état d'esprit, qui implique une vraie culture de partage et d'entraide, y compris concernant les résultats de l'entreprise.

• **L'Innovation** : s'impose aussi comme une valeur incontournable. La recherche permanente d'idées, de solutions et de modèles nouveaux pour répondre efficacement aux enjeux d'aujourd'hui et de demain. Sa mise en œuvre n'est pas un chemin facile et demande une discipline intellectuelle de tous les instants pour accepter une remise en question permanente.

• **La Transparence** : est une autre valeur qui peut sembler s'imposer comme une évidence, mais qui se heurte aux enjeux et comportements humains, car il est tentant de cacher ce qui peut handicaper l'équipe. Au-delà de la valorisation morale, la transparence est un investissement qui peut s'avérer bénéfique avec le temps, car il instaure un climat de confiance, installe une crédibilité qui permet de construire une collaboration durable.

• **L'Audace** : et la prise de risque passe par élaborer, imaginer et mettre en œuvre des solutions peu orthodoxes pour atteindre un objectif ou réaliser une ambition. C'est une approche qui accepte l'échec comme une possibilité ou une opportunité et non une fatalité, et qui y voit les bribes et ingrédients du succès à venir.

• **L'Excellence** : est la cinquième valeur phare chère à Novelis. L'excellence commence par un sens aigu du service, une implication sans faille dans l'exécution de nos tâches, mais aussi un souci permanent du détail : soigner nos interventions du début à la fin, que ce soit sur le fond ou sur la forme. La quête de l'excellence exige aussi de l'humilité et une capacité permanente à se remettre en question, apprendre de ses erreurs et d'autres, dans une démarche d'amélioration continue.

- **L'Agilité :** est adoptée chez Novelis, non seulement comme méthodologie, mais surtout comme philosophie de travail. L'exécution des projets aujourd'hui a beaucoup évolué. Nous faisons souvent face à des projets moins volumineux mais avec un Time to Market plus court et un besoin qui change continuellement. L'agilité s'impose donc comme une évidence pour pouvoir adresser cette nouvelle génération de projets, de la façon la plus efficace pour les clients

1.1.3 Secteur d'activité et structure interne

Depuis sa création, Novelis accompagne ses clients dans leurs projets de transformation digitale. Cet appui couvre une palette riche : Développement de solutions métier, Audits d'architectures, Mise en œuvre de solutions d'automatisation, Projets intégrant des algorithmes complexes et des composants IA.

Pour les sujets portant sur le traitement de documents numérisés (OCR – Optical Character Recognition), l'automatisation des processus front et back office ou encore le déploiement de solutions NO/LOW code, Novelis travaille avec les acteurs les plus avancés sur le marché. Avec plus de 40 collaborateurs certifiés sur ces différentes solutions, Novelis a la capacité d'accompagner ses clients dans des projets de Smart Automation, que leur réflexion soit à ses débuts ou avec une vocation de déploiement massif, incluant une dimension organisationnelle et de gouvernance.

Le figure 2 illustre les différents partenaires de Novelis dans le domaine de RPA et OCR.

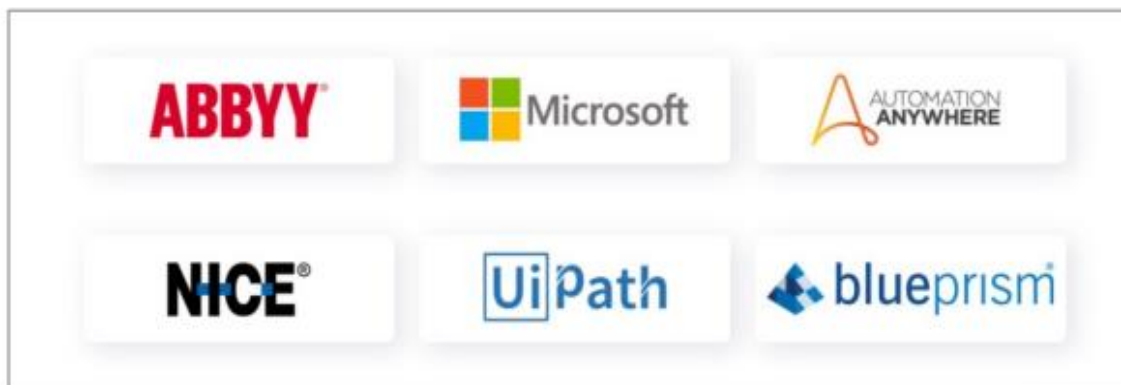


Figure 2 : Partenaires en RPA et OCR

Les six principaux domaines dans lesquels Novelis est active, sont :

Applications Métier : Novelis propose ses consultants, ingénieurs en développement, Experts RPA, Product Owner et Coach Agile à ses clients afin de réussir leurs projets avec la garantie d'un partage de valeurs communes autour de la performance et de la transparence. L'offre Services de Novelis se décline au forfait et en régie afin de répondre à tous les besoins de la façon la plus pertinente.

Novelis accompagne notamment ses clients sur les périmètres suivants :

- **Digital Consulting** : Accompagner les clients dans l'émergence ou la déclinaison de nouvelles offres en apportant les expertises Métiers, Techniques et Méthodologiques nécessaires.
- **Digital Delivery** : Mettre à disposition des clients et partenaires des structures de production, adaptables et agiles, capables de mettre en œuvre et de maintenir des projets digitaux selon différents modes d'engagement (périmètre, moyens, engagement capacitaire).
- **Data Engineering** : Analyser et exploiter les données à travers la construction de modèles prédictifs d'algorithmes complexes (intégration IA, DL / ML). Les experts Data travaillent en étroite collaboration avec le Lab R&D.
- **Agile Oriented** : Novelis propose également des profils Scrum Master, Product Owner pour réussir les projets en agilité.

Digitalisation de documents : Novelis accompagne ses clients dans la gestion des données structurées et non structurées. Ces dernières, qu'il s'agisse de documents scannés, numérisés comme des factures, des bons de commandes, des contrats, des documents officiels, nécessitent très souvent de nombreuses actions humaines avant d'être traitées dans le Système d'Informations de l'entreprise ou du service public.

Novelis avec son partenaire ABBYY, leader mondial de l'OCR (Optical Character Recognition), permettra de lire tous ces documents, de les classer, les comprendre puis de les traiter dans les différentes applications de l'entreprise. Au travers de son expertise en NLP

(Natural Language Processing), Novelis va plus loin dans les projets de dématérialisation et permet à ses clients d'intégrer une couche sémantique permettant de gagner du temps et de la réactivité dans le traitement des données non structurées.

Les projets de dématérialisation des documents et d'OCR peuvent ainsi être couplés avec des solutions de Robotisation des processus (RPA), des ChatBots ou des VoiceBots pour améliorer l'expérience client. L'association de ces technologies dans le respect du roadmap IT du client est la « Smart Automation » par Novelis.

L'offre « Digitaliser les documents » c'est :

- **Audi** : Audit des processus intégrant les documents afin de qualifier la meilleure approche OCR / NLP de vos données non structurées.
- **Delivery** : Déploiement de pilotes, de mises en production et gestion de la maintenance des projets délivrés.
- **Distribution** : Distribution des licences ABBYY (Cloud / On-Premise) et installation / maintenance.
- **Training** : Formation à l'utilisation de la solution ABBYY FlexiCapture et à la qualification des projets OCR

Automatisation des processus : Novelis est aux côtés de ses clients dans la gestion des projets d'automatisation des processus, que le projet porte sur la suppression de tâches de processus chronophages, répétitifs, ou encore l'assistance des collaborateurs sur des tâches complexes.

Novelis, avec plus de 40 collaborateurs certifiés avec tous les acteurs clés de ces nouvelles technologies, est en mesure de réaliser des projets RPA de qualité élevée, que ces derniers portent sur des processus back-office finance (gestion des factures, des commandes des notes de frais, des rapprochements...), IT (gestion des mots de passe, rapprochement des données IT...), RH (gestion des entrées, sorties...) ou bien Front-Office (support aux centres de contacts).

Au travers de son expertise en IA, en NLP et OCR, Novelis est capable d'accompagner ses clients pour délivrer des projets Smart Automation. Il s'agit alors de prendre en compte une vision générale de la technologie et ses apports afin de délivrer une solution pertinente pour gérer à la fois les données structurées et non structurées, et c'est toujours avec le souci d'être aligné avec la roadmap IT de ses clients et d'impacter la performance économique (ROI).

A ces robots logiciels, Novelis ajoute également une expertise particulière dans la dimension « Process Mining » afin de permettre l'identification rapide des processus et tâches à automatiser en priorité. Les Business Analystes complètent ce dispositif dans la capacité à proposer une vision haute du potentiel de la technologie, les investissements à mener, les architectures à déployer et bien entendu le ROI attendu.

L'offre « Automatiser les processus » c'est :

- **Audit** : Commercialisation des licences de ses partenaires RPA en Saas/ Cloud ou bien On-Premise en tant que partenaire de référence en Europe.
- **Delivery** : Déploiement de pilotes, de mises en production (installation) et gestion de la maintenance des projets délivrés.
- **Distribution** : Commercialisation des licences de ses partenaires RPA en Saas/Cloud ou bien On-Premise en tant que partenaire de référence en Europe.
- **Organisation** : Déploiement d'organisations de Centres d'Excellences RPA (principes de gouvernances, gestion des demandes, formations...)

CRM / Relation Clients : Novelis propose son expertise dans la gestion de la relation clients dans une approche pragmatique, intégrant à la fois la dimension liée à la gestion des processus clés (réclamation, support, gestion de comptes...) mais également la dimension fonctionnelle (SVI, CRM, Desktop Unifié, analyse des sentiments, de la voix...). Novelis a donc désigné une offre spécifique permettant d'échanger à la fois avec la Direction de la Relation Clients, la Direction Informatique, et la Direction Marketing.

L'offre « CRM / Relation Clients » c'est :

- **Audit** : Analyse et audit des relations clients et des KPI clés (productivité, NPS, satisfaction salariée, parcours clients...).

- **Stratégie « Seamless »** : Création de Business Cases et de recommandations afin de définir des stratégies d'optimisation de la relation clients à distance.
- **Delivery** : Etude et déploiement de solutions de relation clients innovantes (Chatbots, Voicebots, CRM, SVI, Speech Analytics, eRéputation...).
- **Training** : Formations de vos ressources sur des solutions fonctionnelles ou bien des approches métiers permettant de faire la différence

Solutions Métiers SaaS : Novelis a lancé une initiative visant à proposer à ses clients des solutions logicielles adaptées à des besoins particuliers que ses experts ont pu identifier dans le cadre de leurs missions ou bien l'entreprise elle-même dans sa gestion quotidienne.

L'offre « Novelis Solutions » propose aujourd'hui les logiciels suivants accessibles en SaaS / Cloud :

- **Auto-hiring** : Solution pour les cabinets de recrutement et les Directions des Ressources Humaines permettant de créer des campagnes de recrutement de ressources incluant des tests logiques, de questions ouvertes et de gestion de CV / candidats.
- **Exam-Direct** : Solution pour les écoles, universités permettant de gérer des examens à distance pour des candidats et / ou des étudiants. La solution intègre la création de tests simples, complexes et permet de lancer des approches d'apprentissages
- **OWD** : Identification automatique d'objets et de mots dans des vidéos (projet IA /NLP).

1.2 Définition du projet

1.2.1 Cadre général du projet

Le projet du stage intitulé « Amélioration d'un module de pilotage opérationnel, la conception et développement d'un module de pilotage financier des projets » est initié dans le cadre de la solution Novy, premièrement pour l'amélioration de certaines fonctionnalités dans le module existant, et deuxièmement pour le développement d'un nouveau module de pilotage des projets coté financière qui font appels aux données générés dans les autres modules. La

solution Novy est une plateforme web en micro-services composée de plusieurs modules et elle permet de gérer plusieurs aspects d'entreprise, citant :

- Gestion de la Relation Client (CRM)
- Gestion de la Paie
- Gestion des Candidats & Recrutement
- Gestion des Collaborateurs & Projets
- Gestion des Ressources Humaines.

En se basant principalement sur le service « Gestion des Collaborateurs & Projets », il est possible de déduire l'état courant dans le temps de chaque projet en termes de ce qui est réalisé, par rapport à ce qui est attendu. Il est également possible de suivre le rendement de chaque collaborateur dans les projets sur lesquels il travaille. Le pilotage est une étape fondamentale dans la gestion de projets. Il vient après l'étape de l'organisation qui consiste en générale de l'étude des points suivants :

- Périmètre et lotissement du projet
- L'équipe de projet
- Planification
- Budgétisation.

Une fois le projet est budgété, organisé et planifié, le projet démarre. Au cours de son avancement, le pilotage sert à faire la comparaison entre le réalisé et le prévisionnel, et s'assurer que le travail entrepris va dans la bonne direction. Cela implique la détermination de certains indicateurs primordiaux pour une suivie bien détaillée, permettant de mesurer une situation ou un risque, de donner une alerte ou au contraire de signaler qu'on est sur le bon chemin. Le choix des indicateurs dépend des besoins de l'entreprise et du projet lui-même.

1.3 Problématique

La solution existante permet de piloter un projet d'une façon non seulement non détaillée, mais aussi difficile. En fait, le responsable sera mené à exporter des dizaines de fichiers Excel, les combiner pour pouvoir appliquer des fonctions de calcul des indicateurs opérationnels et financiers, et générer des visualisations permettant de connaître la situation actuelle du projet et prendre la décision la plus adéquate. Il sera tellement difficile de répéter ce processus de façon quotidienne, hebdomadaire, ou mensuelle. Aussi, cette solution ne permettra pas de piloter le projet dans le temps de façon à être constamment à jour, afin de déduire l'état courant, l'avancement vis-à-vis le passé, et les risques associés.

1.4 Objectifs

Dans l'optique d'apporter de nouvelles fonctionnalités et remédier aux insuffisances et limites explicitées auparavant. Le service « Pilotage des Projets » permettra de faciliter le suivi opérationnel et financier d'un projet.

1.4.1 Amélioration de suivi opérationnel d'un projet

**** Gestion des actions d'un projet :**

Les actions d'un projet sont les différentes tâches que nous devons accomplir pour atteindre nos objectifs. Cela peut inclure la collecte d'informations, la planification, la réalisation des travaux, et le suivi de notre progression. Chaque membre de l'équipe est responsable de certaines actions et nous devons nous assurer de les accomplir dans les délais fixés. Parfois, nous devons ajuster nos actions en cours de route pour faire face aux changements ou aux problèmes. Il est important de suivre régulièrement nos actions afin de rester sur la bonne voie et d'obtenir les résultats que nous voulons.

**** Gestion des risques d'un projet :**

Les risques d'un projet sont des événements ou des circonstances incertaines pouvant nuire à la réalisation des objectifs, tels que des retards, des dépassements de budget, une baisse de la qualité ou l'échec du projet. Ces risques peuvent provenir de diverses sources, comme des contraintes externes, des facteurs internes, des problèmes techniques, des erreurs de

planification ou des imprévus. Pour minimiser les impacts négatifs et augmenter les chances de réussite, il est crucial d'identifier et d'évaluer ces risques dès le début du projet, de mettre en place des stratégies de prévention, des plans d'action de contingence et de surveiller constamment le projet. Ainsi, l'équipe du projet pourra assurer la livraison des résultats conformément aux attentes.

**** Gestion des risques d'une action :**

La gestion des risques d'une action consiste à identifier, évaluer et gérer les différentes incertitudes qui pourraient affecter cette action. Cela implique d'identifier les risques, d'évaluer leur probabilité et leur impact, puis de mettre en place des stratégies pour les atténuer ou les gérer de manière appropriée. Cela peut inclure des mesures préventives, des plans d'action de contingence et une surveillance continue de l'action pour détecter et répondre aux risques émergents. L'objectif de la gestion des risques est de minimiser les impacts négatifs sur l'action et d'augmenter les chances de réussite en assurant une prise de décision éclairée et une gestion proactive des risques tout au long du processus.

1.4.2 Suivi financier d'un projet

**** Gestion des commandes d'un projet :**

La gestion des commandes d'un projet concerne toutes les actions nécessaires pour prendre en compte, traiter et satisfaire les demandes des clients dans le cadre du projet. Cela implique de mettre en place des processus et des systèmes efficaces pour gérer les demandes des clients, établir des devis, respecter les délais, coordonner avec les fournisseurs et livrer les produits ou services commandés. L'objectif de la gestion des commandes est de s'assurer que les demandes des clients sont traitées correctement, en respectant les contraintes de temps, de coûts et de qualité du projet. Cela nécessite de suivre et de contrôler les commandes, résoudre rapidement les éventuels problèmes ou litiges, afin d'assurer la satisfaction des clients et le succès du projet.

**** Gestion des échéanciers d'une commande :**

La gestion des échéanciers d'une commande consiste à planifier et à respecter les délais associés à la réalisation de cette commande. Cela implique d'établir un calendrier précis des différentes étapes du processus, de déterminer les délais de livraison attendus et de s'assurer que toutes les parties impliquées sont informées des échéances à respecter. La gestion des échéanciers comprend également le suivi régulier de l'avancement de la commande, l'identification des éventuels retards ou obstacles, et la mise en place de mesures correctives pour rétablir le planning initial. L'objectif est de garantir que la commande soit réalisée dans les délais convenus, en minimisant les retards et en assurant la satisfaction du client.

1.5 Besoins fonctionnels

La définition des spécifications fonctionnelles du projet est une étape primordiale au début de chaque démarche de développement des logiciels qui consiste à caractériser les fonctionnalités offertes par un produit pour répondre aux besoins d'un utilisateur. En effet, chacun de ces besoins reflète les attentes des différents utilisateurs du système conçu. Nous présenterons ainsi tous les besoins fonctionnels qui seront implémentés dans notre projet.

On liste ci-dessous, et de façon exhaustive, le résultat de l'étude fonctionnelle et la spécification des besoins, effectués en collaboration avec l'ingénieur tests et automatisation faisant partie de l'équipe, notre concepteur UX/UI, et à travers des réunions avec notre encadrant et le directeur des opérations de Novelis.

1.5.1 Amélioration de suivi opérationnel d'un projet

- **Actions globales** : le tableau des actions qui contient tous les actions des projets :
 - La liste Contient les éléments suivants : nom de l'action, nom du projet, responsable, échéance, priorité, statut, description, type d'action et commentaire de clôture.
 - Pouvoir créer, modifier, et supprimer une action.
 - Pouvoir rechercher, filtrer, et trier les actions.

- **Actions d'un projet** : le tableau des actions que subit un projet :
 - La liste Contient les éléments suivants : nom de l'action, responsable, échéance, priorité, statut, description, type d'action et commentaire de clôture.
 - Pouvoir créer, modifier, et supprimer une action.
 - Pouvoir rechercher, filtrer, et trier les actions.
- **Risques globaux** : le tableau des risques qui contient tous les risques des projets :
 - La liste Contient les éléments suivants : nom du risque, nom du projet, responsable, échéance, priorité, statut, catégorie, description et commentaire de clôture.
 - Pouvoir créer, modifier, et supprimer une action.
 - Pouvoir rechercher, filtrer, et trier les risques.
- **Risques d'un projet** : le tableau des actions que subit un projet :
 - La liste Contient les éléments suivants : nom de l'action, responsable, échéance, statut, catégorie, commentaire de clôture et description.
 - Pouvoir créer, modifier, et supprimer une action.
 - Pouvoir rechercher, filtrer, et trier les risques.
- **L'action d'un risque** : le tableau des actions que subit un risque :
 - La liste Contient les éléments suivants : nom de l'action, responsable, échéance, statut, catégorie, commentaire de clôture et description.
 - Pouvoir créer, modifier, et supprimer une action.
 - Pouvoir rechercher, filtrer, et trier les actions.

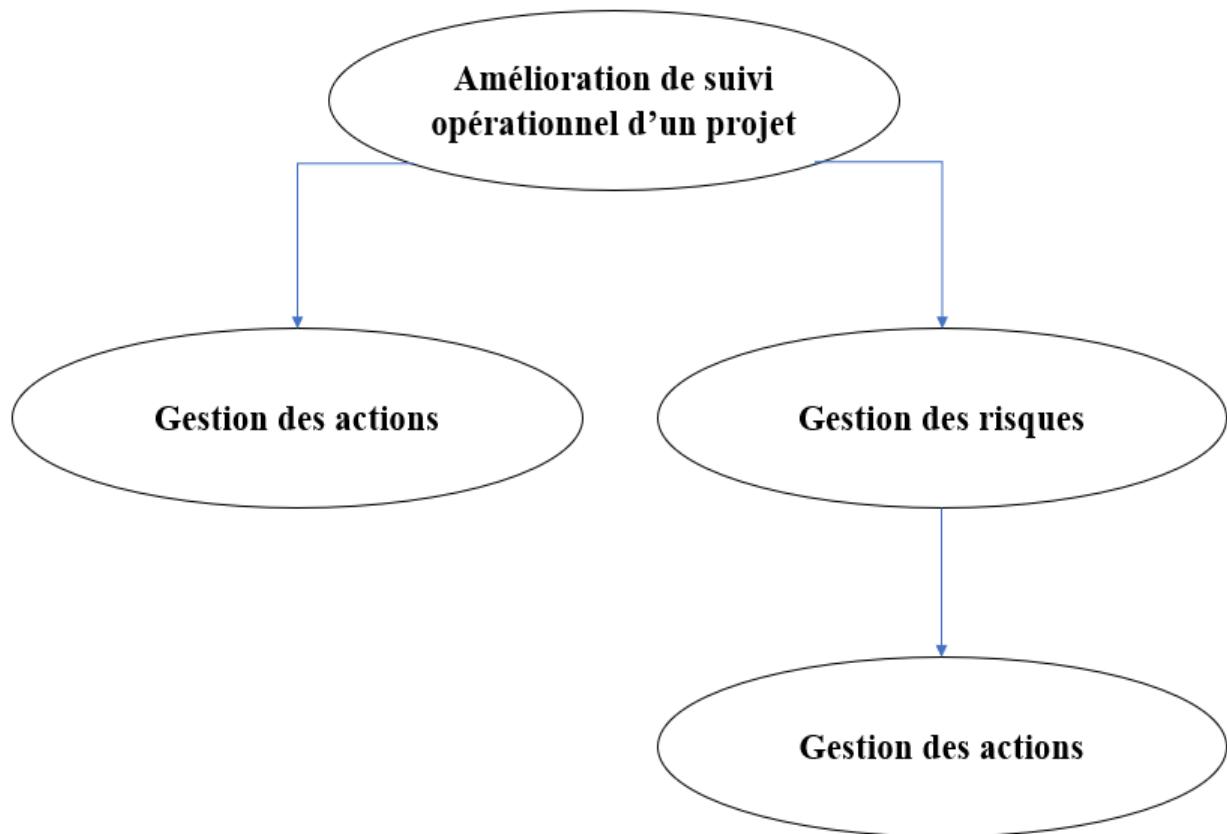


Figure 3 : suivi opérationnel

1.5.2 Suivi financier d'un projet

- **Commandes** : la vue des commandes contient les commandes d'un projet :
 - La liste Contient les éléments suivants : nom de commande, montant, charge vendu, nature de vente, type de vente et la commande est clôturé ou non.
 - Pouvoir créer, modifier, et supprimer une commande.
 - Pouvoir rechercher, filtrer, et trier les commandes.
- **Échéanciers** : la vue des échéanciers contient les échéanciers d'une commande :
 - La liste Contient les éléments suivants : nom de l'échéancier, montant prévisionnel HT, date prévisionnelle de facturation, charge prévisionnelle, statut.
 - Pouvoir créer, modifier, et supprimer une commande.
 - Pouvoir filtrer, et trier les commandes.

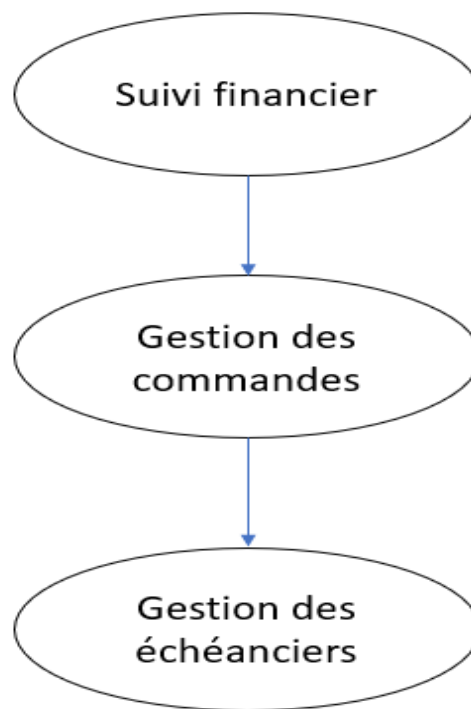


Figure 4 : suivi financier

1.6 Besoins non fonctionnels

Les besoins non fonctionnels décrivent les objectifs liés aux performances du système et aux contraintes de son environnement. Ses exigences techniques sont souvent exprimées sous forme d'objectifs spécifiques que doit atteindre le système.

- **Disponibilité** : Notre application constitue un système de gestion d'entreprise, il est important qu'elle soit disponible à tout moment pour garantir un meilleur contrôle sur les activités de l'entreprise.
- **Sécurité** : Les données traitées par notre application sont de nature pertinente, par conséquent tous les accès doivent être protégés.
- **Performance** : La minimisation du temps de traitement est le cœur de notre application, pour garantir une bonne qualité de service et une meilleure gestion au sein de l'entreprise.

- **Portabilité** : Notre application doit être portable et compatible avec tous les environnements pour des prochaines améliorations.
- **Responsivité** : L'interface utilisateur doit être bien claire et facile à utiliser dans les différents appareils.
- **Maintenabilité** : Les différents modules de l'application doivent être clairs compréhensibles afin de les maintenir et de les mettre à jour rapidement et facilement.
- **Réutilisation** : L'application doit permettre l'ajout et l'édition des extensions futures.

1.7 Diagramme de GANTT

Sprint	Estimation (taille)
Formation	21 jours
Sprint 1	27 jours
Sprint 2	21 jours
Sprint 3	21 jours
Sprint 4	21 jours

Tableau 2 : Table de GANTT

- **Formation :**
 - Formation coté backend et frontend pour bien savoir les différentes technologies utilisées par Novelis.
- **Sprint 1 (l'amélioration du module pilotage coté opérationnel) :**
 - Correction des bugs et des warnings, améliorer la partie de gestion des actions de projet coté backend, BFF (back for front) et frontend.
 - La création des tables de la partie risque de projet.
 - Implémenter le backend pour la partie de risque de projet.
- **Sprint 2 (l'amélioration du module pilotage coté opérationnel) :**
 - Correction des bugs et des warnings,
 - Implémenter le BFF pour la gestion des risques de projet.
 - Implémenter frontend pour la gestion des risques de projet.
- **Sprint 3 :**
 - **L'amélioration du module pilotage côté opérationnel** : correction des bugs et des warnings.
 - **Module pilotage côté financier** : création des tables pour la partie commande et échéancier.
 - Implémenter le backend pour la partie commande et échéancier de projet.
 - Implémenter le BFF pour la partie commande et échéancier de projet.
 - Implémenter le frontend pour la partie commande de projet.
- **Sprint 4 :**
 - **L'amélioration du module pilotage côté opérationnel** : correction des bugs et des warnings.
 - **Module pilotage côté financier** : correction des bugs et des warnings.
 - Implémenter le frontend pour la partie échéancier d'une commande.

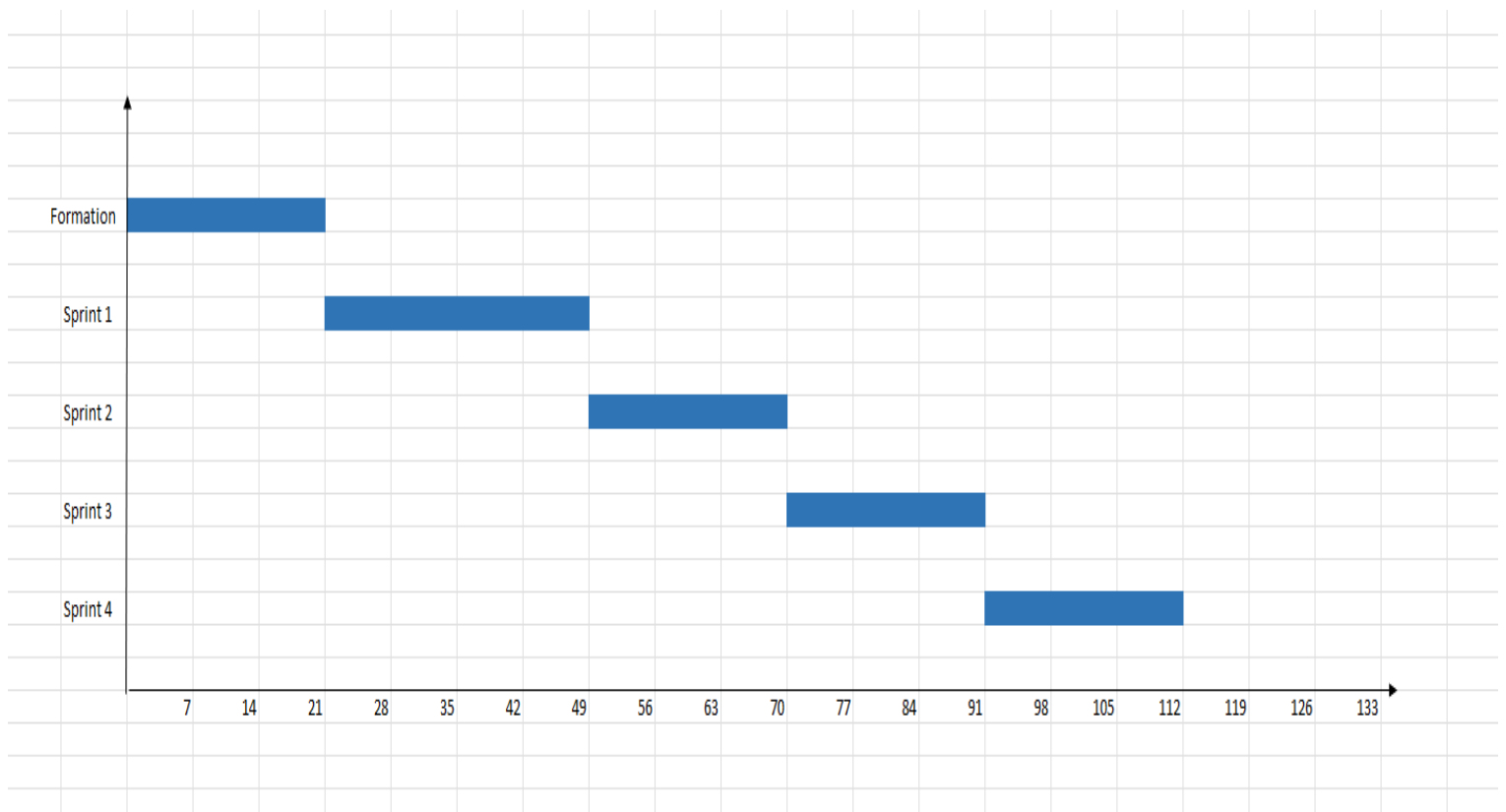


Figure 5 : Diagramme de GANTT

Chapitre 2

Analyse et conception

La phase de conception est l'une des phases les plus importantes pour le développement d'un système d'information. Plusieurs méthodes ont été proposées pour mettre en œuvre cette phase telles que : Merise, UML...etc. Dans cette partie du chapitre, on a suivi les étapes nécessaires pour améliorer la partie opérationnel et concevoir la partie financier de notre module de pilotage des projet . Pour cela, dans cette conception, on a utilisé la méthode UP (processus unifié), cette dernière qui utilise UML comme langage de modélisation et qui offre une souplesse remarquable qui s'exprime par la possibilité d'obtenir des modèles de systèmes reflétant la réalité à l'aide de diagrammes.

1.1 Présentation générale d'UML

UML (Unified Modeling Language) est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancé par l'OMG (Object Management Group) afin de définir une notation standard pour les applications de modélisation construites à partir d'objets. Il est hérité de plusieurs autres méthodes telles que l'OMT (Object Modeling Technique) et OOSE (Object Oriented Software Engineering) et Booch. Les principaux auteurs de la notation UML sont Grady Booch, Ivar Jacobson et Jim Rumbaugh.

1.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est utilisé pour exprimer le comportement d'un système ou la sémantique de toute autre entité sans révéler sa structure interne. Chaque cas d'utilisation spécifie une séquence d'action, y compris des variantes que l'entité réalise, en interagissant avec les acteurs de l'entité. Le rôle d'un cas d'utilisation est de spécifier un ensemble d'instances, où une instance de cas d'utilisation représente une séquence d'actions que le système réalise et qui fournit un résultat observable par l'acteur.

1.2.1 Identification des acteurs

Un acteur représente l'abstraction d'une entité susceptible d'interagir directement avec le système, nous identifions l'acteur principal de l'application qui est le manager, chef de projet ou responsable des ressources humaines au sein de Novelis. Il peut accéder à la plateforme où il peut bénéficier de toutes ces fonctionnalités. Après authentification au serveur central d'authentification de Novy, le service va limiter l'accès uniquement aux personnes ayant les permissions au module Pilotage.

Les acteurs de Novy-solution avec les responsabilités de chacun :

Modules	Manager	Equipe RH	Relai RH	Collaborateurs	Equipe Commerce	Marketing
Collaborateurs	✓	✓	✓	✓	✓	✓
Projets	✓					
Paie	✓	✓				
Recrutement	✓	✓			?	?
Administration	✓	✓			?	?
Client et fournisseurs	✓				?	?
CRM	✓				?	?
Pilotage	✓				?	?

Tableau 3 : Novy-acteurs

Le module Pilotage est constitué de quatre axes principaux dont les rôles du Manager seront répartis comme suit :

**** Les projets**

Le Manager peut :

- Accéder à la liste des projets.
- Filtrer les projets.
- Accéder au Dashboard du projet.
- Visualiser les charges financières du projet (charge vendue, charge initiale, charge consommée, charge actualisée, charge produite).
- Visualiser le graphe représentant l'évolution de la Charge consommée / RAF (Reste à Faire).
- Visualiser l'historique des snapshots pour chaque projet.

**** Les snapshots**

Le Manager a la possibilité de visualiser les snapshots manuels et automatiques de tous les projets avec la possibilité de recherche par nom, client, type et statut de projet.

**** Les actions**

Au niveau des actions, le Manager peut :

- Visualiser les actions (vue en liste / kanban).
- Créer une action.
- Affecter une action à un collaborateur.
- Définir la priorité, le statut, la date d'échéance de l'action.
- Accéder à la liste des actions d'un projet.
- Modifier/supprimer une action

**** Les risques**

Au niveau des risques, le Manager peut :

- Visualiser les risques (vue en liste / kanban).
- Créer / modifier/ supprimer un risque.
- Affecter un risque à un collaborateur.
- Définir la priorité, le statut, la date d'échéance et la catégorie du risque.
- Créer une action et l'affecter à un risque.
- Modifier/ supprimer une action associée à un risque.

**** Les commandes**

Au niveau des commandes, le Manager peut :

- Visualiser les commandes.
- Créer / modifier/ supprimer une commande.
- Affecter une commande à un collaborateur.
- Définir la nature de vente et le type de vente d'une commande.
- Créer un échéancier et l'affecter à une commande.
- Modifier/ supprimer un échéancier associé à une commande.

**** Les échéanciers**

Au niveau des échéanciers, le Manager peut :

- Visualiser les échéanciers (vue en liste).
- Créer / modifier/ supprimer un échéancier.
- Définir le statu d'un échéancier.
- Filtrer la liste des échéanciers par statuts.

La figure suivante représente le diagramme de contexte statique sans rentrer dans le détail des cardinalités. Il montre le module « Collaborateur » qui est utilisé par tout type d'utilisateurs de Novy, et qui gère leurs travaux quotidiens. Cependant, le module Pilotage, qui est basé sur le module Collaborateur, n'est accessible que pour les responsables, chefs de projets, et ceux ayant le droit d'accès « Manager ».

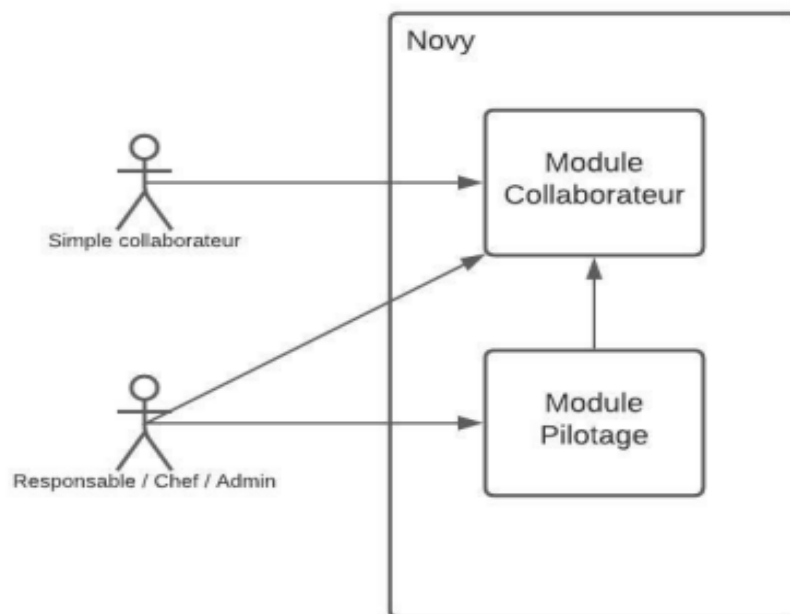


Figure 6 : Diagramme de contexte statique

Dans le diagramme des cas d'utilisation je vais présenter les différentes cases d'utilisations que j'ai la possibilité de soit créé ou modifier :

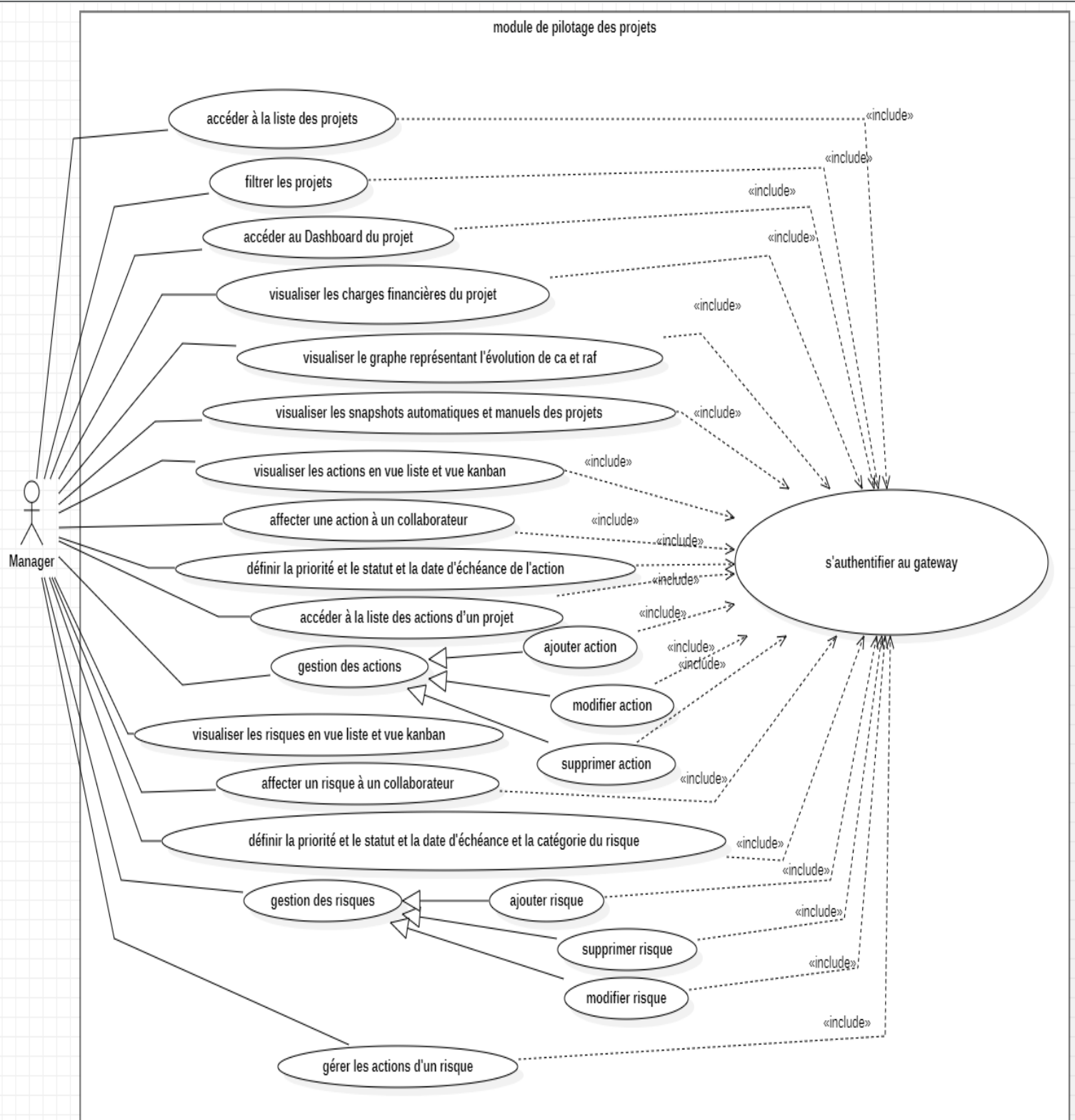


Figure 7 : Diagramme de cas d'utilisation

**** Cas d'utilisation « Gestion des actions » :**

Cas d'utilisation	Gestion des actions
Acteurs	Administrateur/Manager
Description	L'administrateur peut faire ajouter/modifier/supprimer des actions pour un projet et aussi pour un risque
Précondition	S'authentifier
Scénario	L'administrateur doit s'authentifier pour gérer les actions des projets et des risques
Post-condition	Ajout ou modification ou suppression avec succès
Scénario d'exception	Changement échouée, faut vérifier les données saisies

Tableau 4 : cas d'utilisation (gestion des actions)

**** Cas d'utilisation « Gestion des risques » :**

Cas d'utilisation	Gestion des risques
Acteurs	Administrateur/Manager
Description	L'administrateur peut faire ajouter/modifier/supprimer des risques
Précondition	S'authentifier
Scénario	L'administrateur doit s'authentifier pour gérer les risques des projets
Post-condition	Ajout ou modification avec succès
Scénario d'exception	Changement échouée, faut vérifier les données saisies

Tableau 5 : cas d'utilisation (gestion des risques)

1.3 Diagramme de séquences

Il permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.

Ce diagramme met en scène une interaction. En particulier, il montre aussi les objets qui participent à cette même interaction par leur "ligne de vie" et les messages qu'ils échangent présentés sous forme de séquence dans le temps.

Ci-dessous une description détaillée des diagrammes de séquences de quelques cases d'utilisation avec leurs scénarios respectifs :

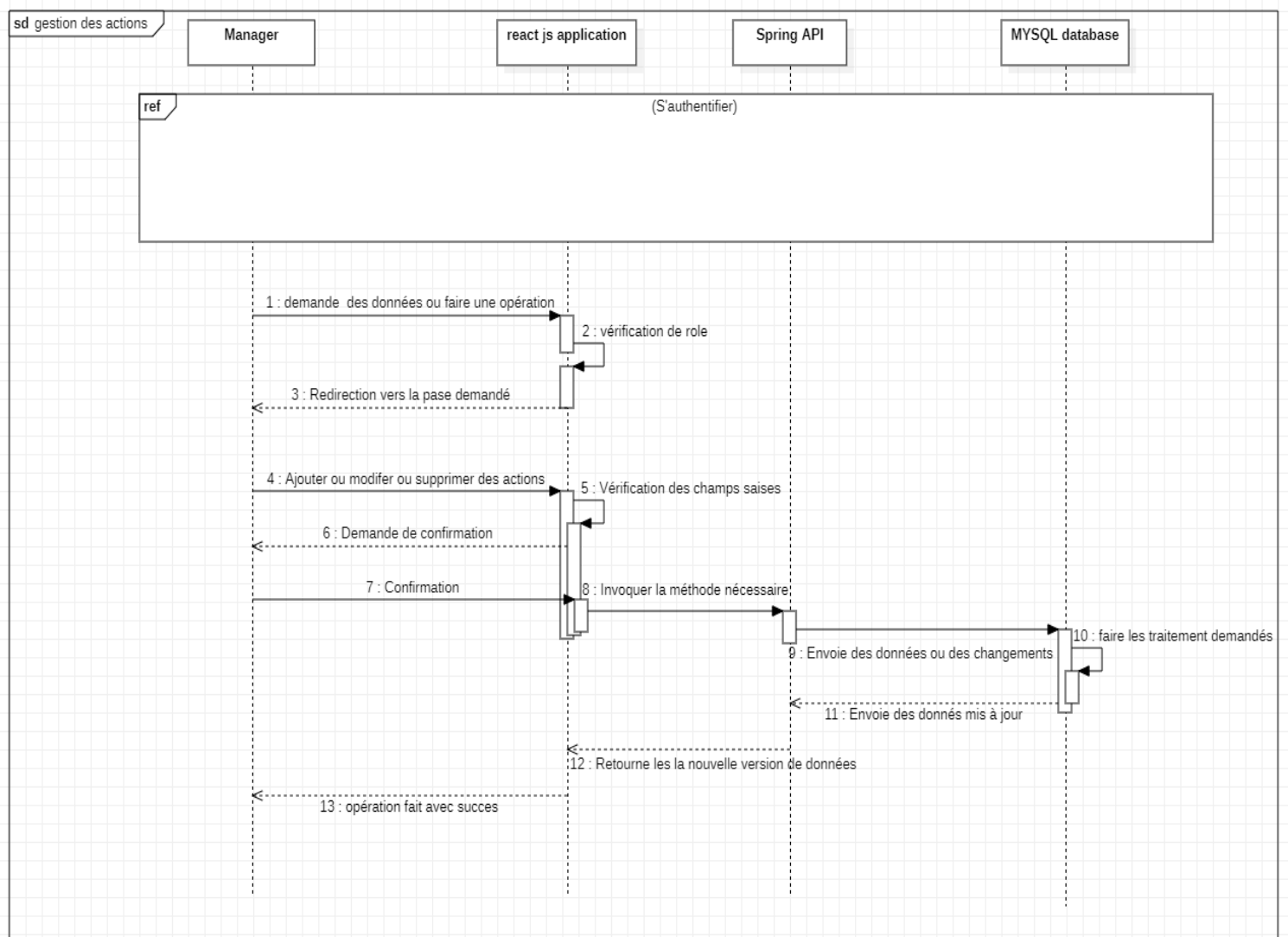


Figure 8 : Diagramme de séquence « Gestion des actions »

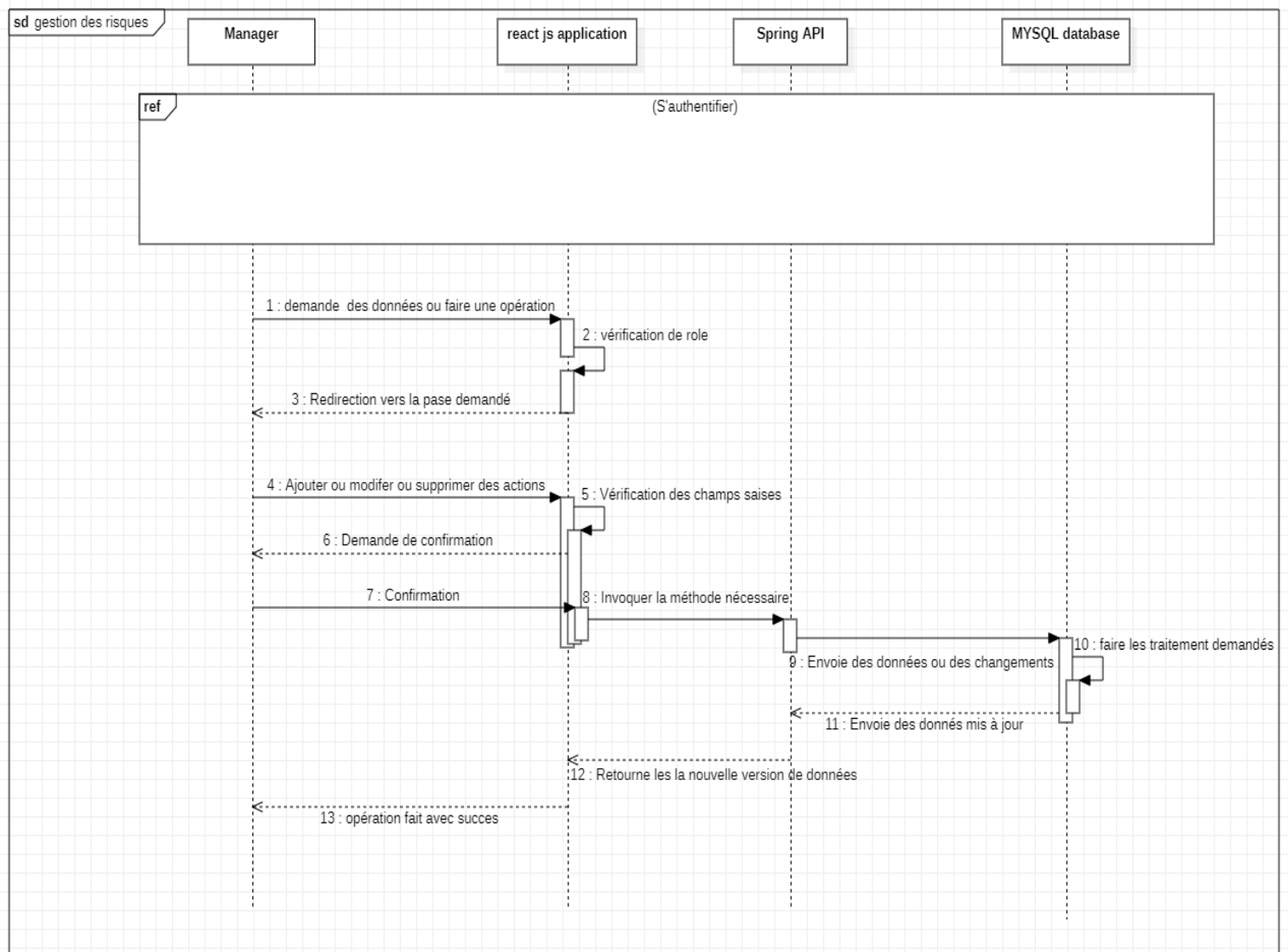


Figure 9 : Diagramme de séquence « Gestion des risques »

1.4 Diagramme de classes

Un diagramme de classe se définit comme étant un ensemble de classes contenant des attributs et des opérations, reliées les unes aux autres par des relations et ceci en ayant des conditions de participation (cardinalités). Le diagramme de classe retenu à la fin de conception est présenté dans la figure suivante :

La figure ci-dessus montre le diagramme de classe pour la partie gestion des actions :

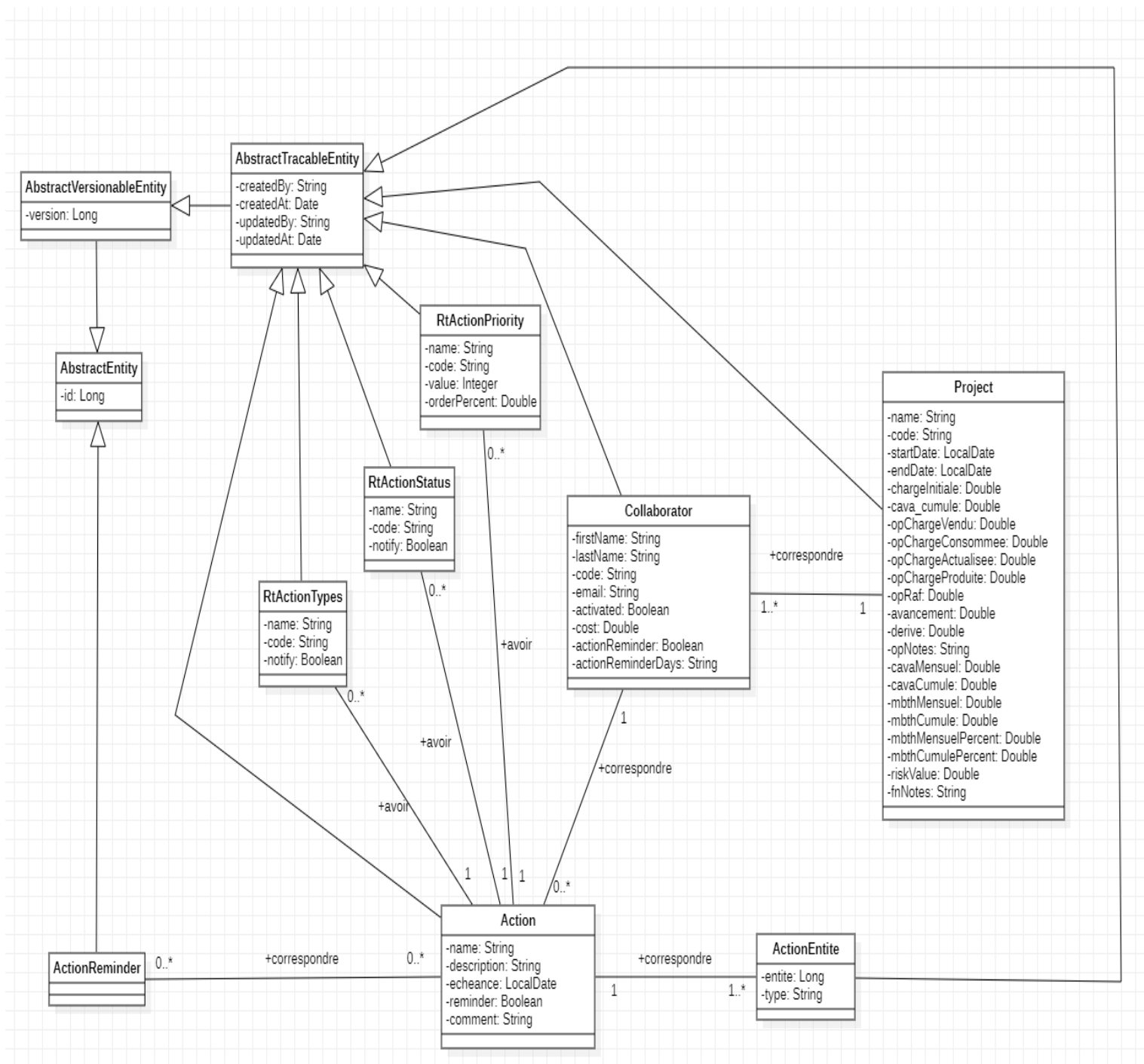


Figure 10 : Diagramme de class « Gestion des actions »

La figure ci-dessus montre le diagramme de classe pour la partie gestion des risques :

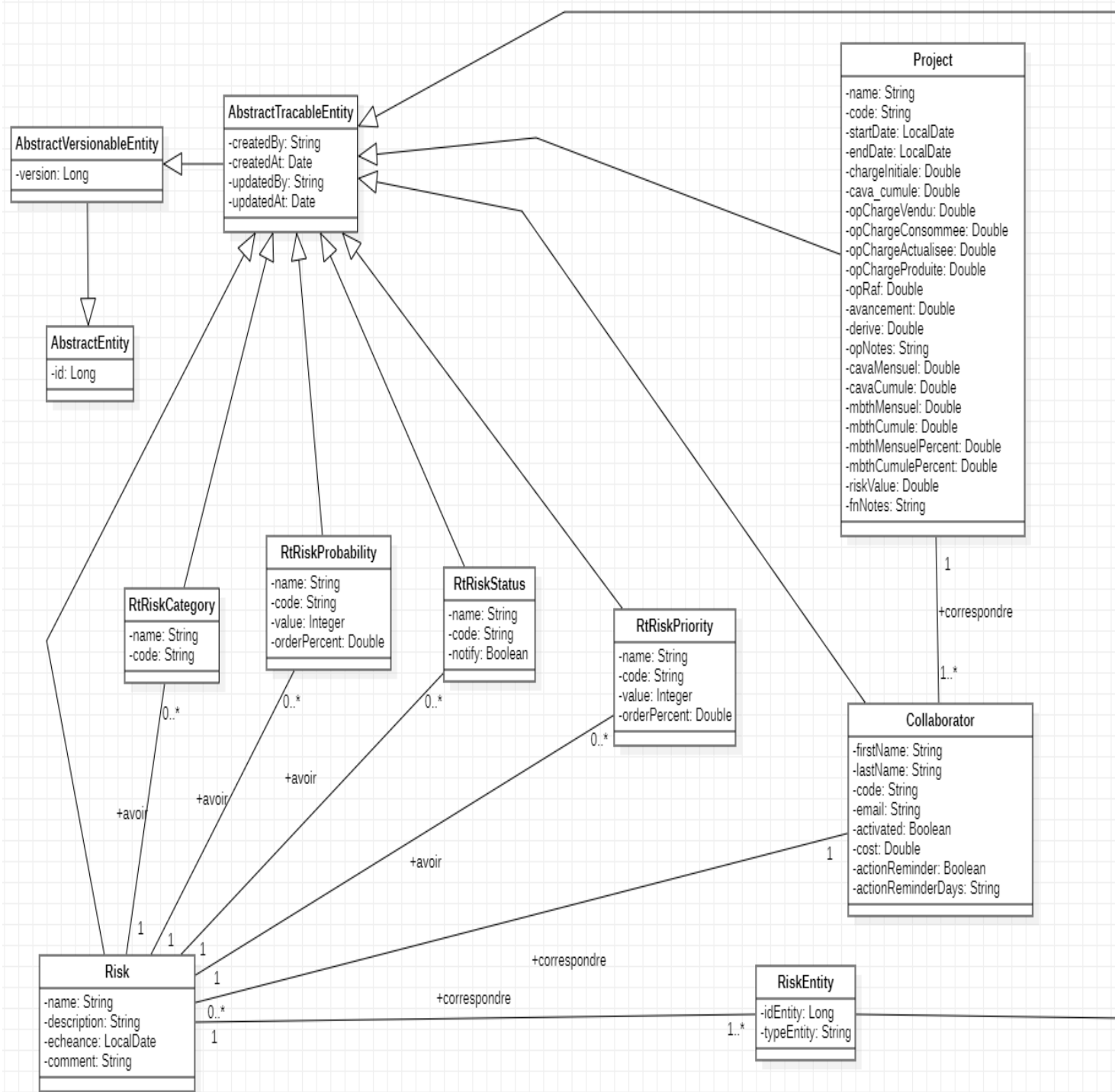


Figure 11 : Diagramme de class « Gestion des risques »

La figure ci-dessus montre le diagramme de classe pour la partie gestion des commandes :

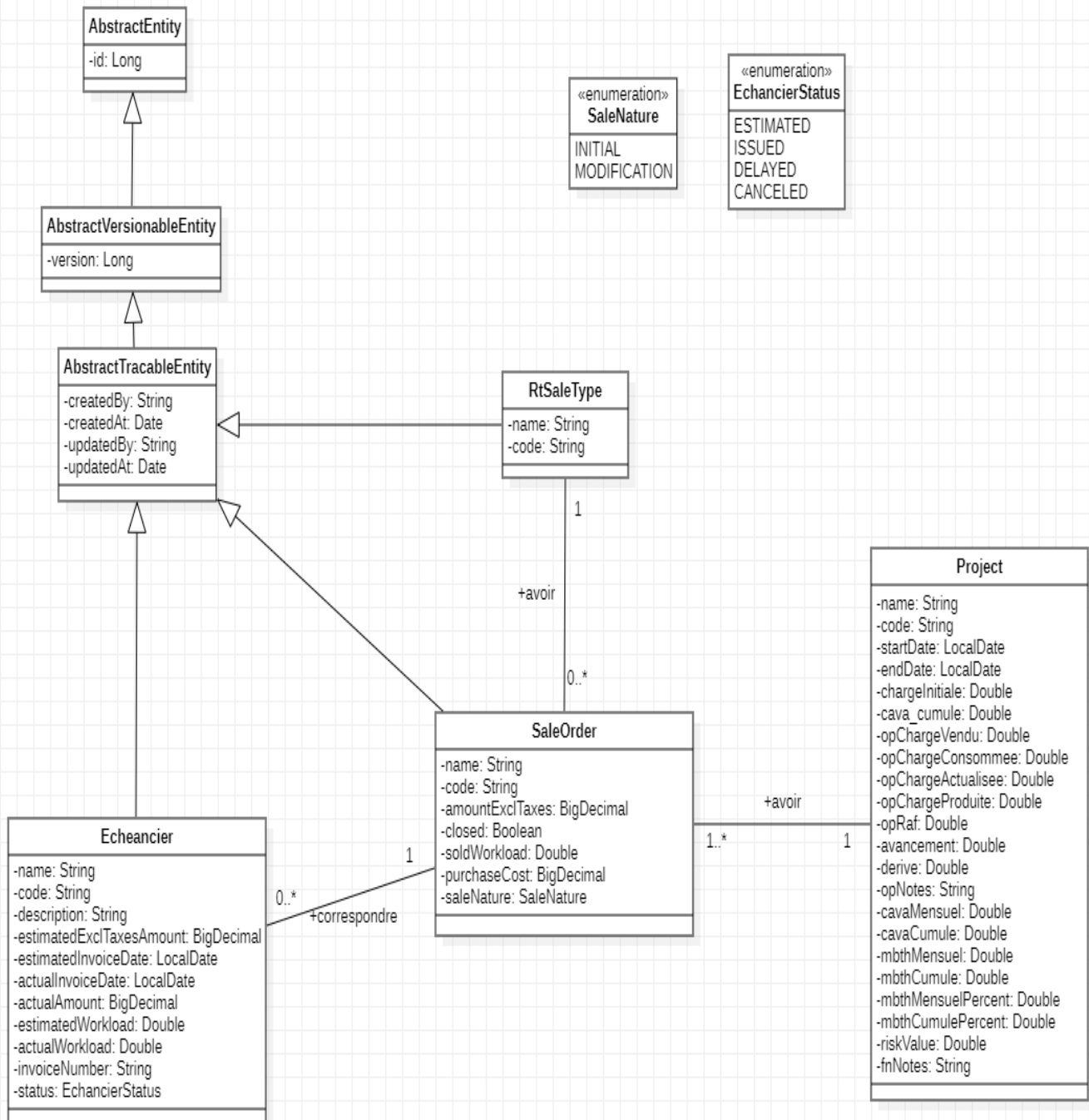


Figure 12 : Diagramme de class « Gestion des commandes »

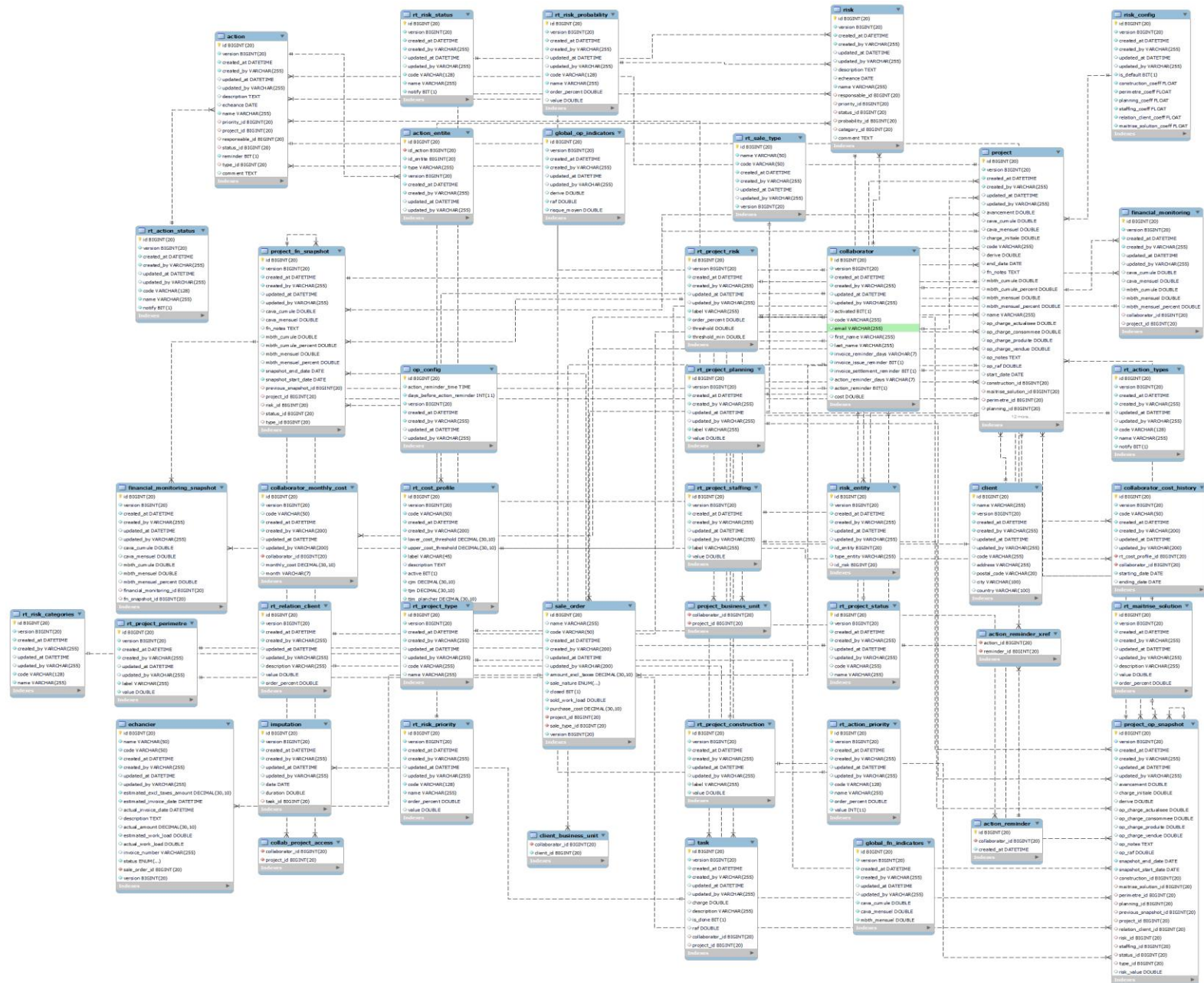


Figure 13 : MR de module pilotage du projet

1.5 Conclusion

Dans ce chapitre, nous avons analysé le fonctionnement de notre système grâce à des diagrammes UML. Nous avons décrit d'abord les cas d'utilisation en recensant de manière textuelle toutes les interactions entre les acteurs et le système suivi par un diagramme des cas d'utilisation. Par la suite, Nous avons complété cette description textuelle par des diagrammes des séquences. Et à la fin, définissant les relations entre les entités, nous sommes parvenus à concevoir le diagramme de classes. Ce qui nous permet de commencer le développement de l'application, ainsi que la base de données optée.

Chapitre 3

Mise en œuvre

Ce chapitre aborde l'implémentation de l'application. Dans un premier temps, je présenterai l'architecture existante du projet, l'environnement de développement utilisé ainsi que les outils qui ont été employés. Ensuite, je décrirai les fonctionnalités que j'ai implémenté en entier ou j'ai amélioré où je présenterai quelques exemples d'interfaces utilisateur.

1.1 Architecture Micro-services

L'architecture de micro-services est un style qui structure une application comme un ensemble de services séparés physiquement, et qui sont généralement :

- Hautement maintenables et testables.
- Faiblement couplés.
- Déployables indépendamment.
- Organisés autour des capacités de l'entreprise.
- Gérés par une petite équipe.

L'architecture de micro-services permet la livraison rapide, fréquente et fiable d'applications larges et complexes. Il permet également à une organisation de faire évoluer sa pile technologique indépendamment des autres services développés auparavant et déployés en production. Les micro-services n'impliquent pas le choix d'une pile technologique commune entre tous les services, chaque service conduit vers une étude de technologies qui prend en considération la nature du métier à développer, et qui donnera le maximum en termes de performance, fiabilité, mais également la simplicité en cas d'une maintenance ou d'une amélioration dans le futur.

La figure suivante présente les éléments de base faisant partie de l'architecture micro-services. Elle est constituée des trois éléments suivants : Service Registry ou registre de service, Gateway ou passerelle, et les services.

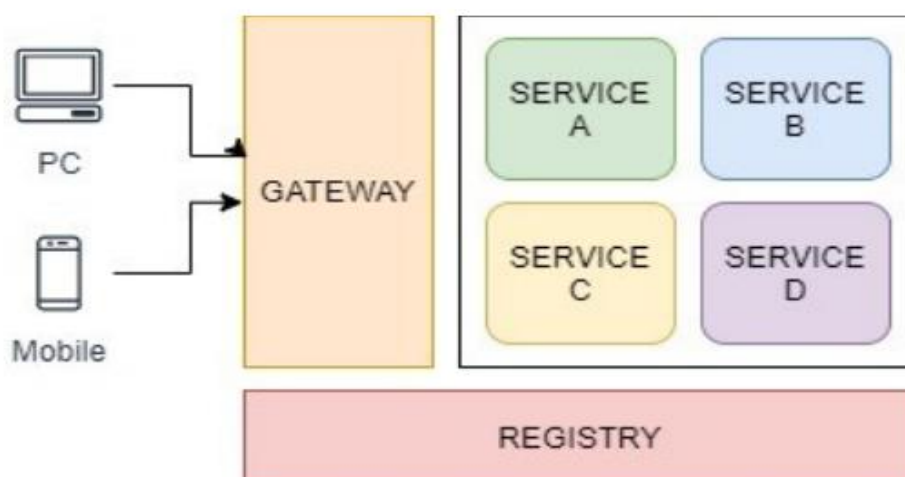


Figure 14 : Architecture micro-service de base

1.1.1 Registre de service

C'est une sorte de base de données de tous les services, de leurs instances et de leurs localisations physiques. Les instances des services sont inscrites auprès du registre de service au démarrage et désinscrites à l'arrêt. Les clients du service interrogent le registre de service pour trouver les instances disponibles d'un service. Un registre de service peut appeler l'API de vérification de l'état d'une instance de service pour vérifier qu'il est en bon état et qu'il est capable de gérer les demandes des clients.

- **Les avantages du modèle de registre de service sont :**

- ** Le client du service et/ou les routeurs peuvent découvrir de façon dynamique l'emplacement des instances de service.

- **Il y a aussi quelques inconvénients :**

- ** Il s'agit d'un autre composant qui doit être installé, configuré et géré. De plus, si le registre échoue ou tombe en panne, toutes les informations sur les services finiront par devenir obsolètes. Par conséquent, le registre de service doit être hautement disponible.

- **Il existe deux manières pour que les services soient inscrits correctement au registre :**

- ** Modèle d'auto-enregistrement : les services s'enregistrent eux-mêmes.

- ** Modèle d'enregistrement tiers : un tiers enregistre les instances de service auprès du registre de service.

Les clients du registre de service doivent connaître l'emplacement ou les emplacements des instances du registre de service. Ces dernières doivent être déployées sur des adresses IP fixes et bien connues.

Dans le diagramme de la figure suivante, le client obtient l'emplacement des instances d'un service en interrogeant le registre de service (Service Discovery). Ensuite, il fait la répartition de charge côté client (Client Side Load Balancing) pour déterminer l'instance à utiliser :

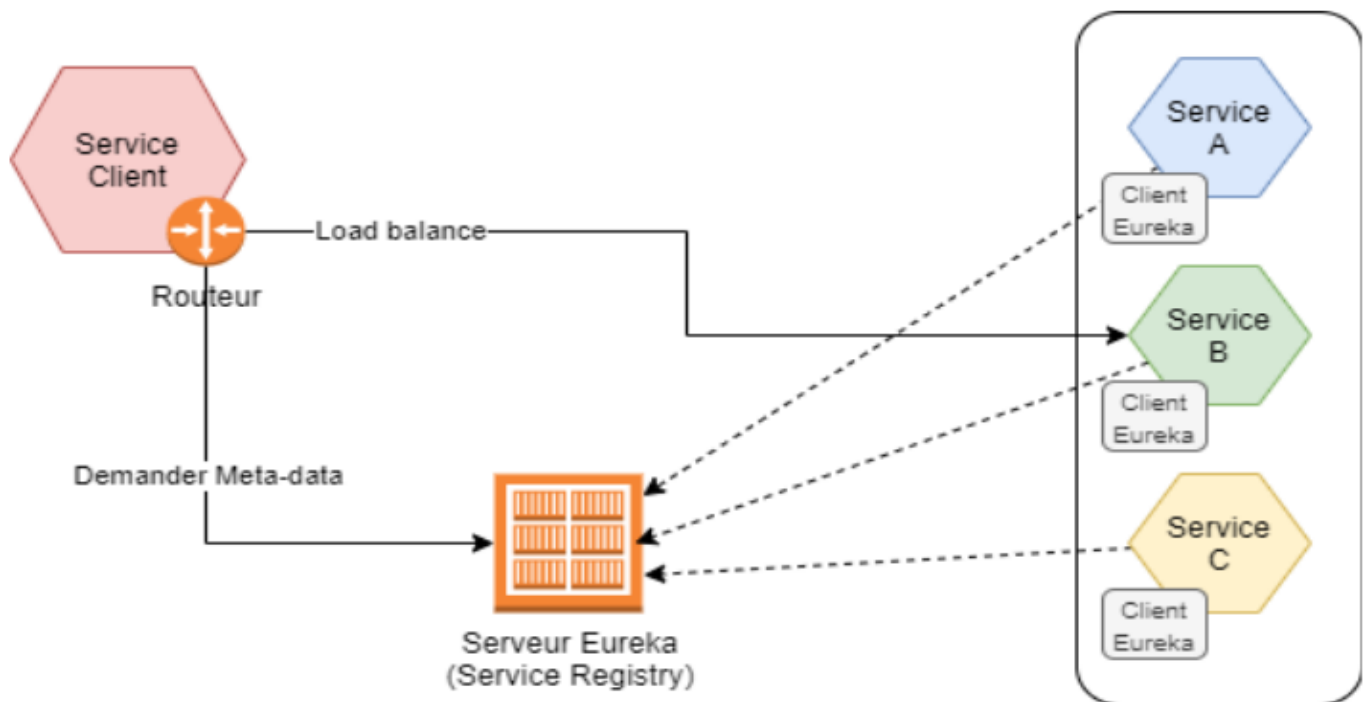


Figure 15 : Découverte de micro-services

- **L'utilisation de telle architecture offre plusieurs avantages, en effet, cette architecture :**

** Isole les clients de la façon dont l'application est partitionnée en micro-services.

** Isole les clients du problème lié à la découverte des emplacements des services (Service Discovery).

** Fournit l'API optimale pour chaque client à travers un adaptateur qui détermine l'API à interroger selon des critères bien déterminés.

** Réduit le nombre de requêtes/allers-retours. Par exemple, l'API Gateway permet aux clients de récupérer des données de plusieurs micro-services avec une seule requête. Moins de demandes signifie également moins de consommation de ressources, et améliore l'expérience utilisateur. Elle est essentielle pour les applications mobiles.

** Simplifie le client en déplaçant la logique d'appel de plusieurs services vers l'API Gateway.

- **Le modèle de l'API Gateway présente certains inconvénients :**

- ** l'API Gateway est un autre composant qui doit être développé, déployé et géré.

- ** Temps de réponse accru en raison du saut de réseau supplémentaire via l'API Gateway (généralement négligeable dans les architectures larges).

1.1.2 Services

Les services contiennent le traitement métier (Business Logic), et doivent être suffisamment petits pour être développés par une petite équipe et être facilement testés. La conception des services nécessite de choisir la manière de décomposer le métier en plusieurs services indépendants, de façon que les changements et améliorations affectent un seul service. En effet, les changements qui affectent plusieurs services nécessitent de la coordination entre plusieurs équipes, ce qui ralentit le développement.

- **Les services doivent être décomposés de manière à garantir les points suivants :**

- ** L'architecture doit être stable.

- ** Les services doivent être cohérents. Un service doit implémenter un petit ensemble de fonctions fortement couplées.

- ** Les services doivent être conformes au principe « Common Closure Principle » les fonctionnalités qui changent ensemble doivent être regroupées ensemble

- ** Les services doivent être faiblement couplés - chaque service expose son API qui encapsule son implémentation

- ** Un service doit être testable.

- ** Chaque service doit être suffisamment petit pour être développé par une petite équipe de 6 à 10 personnes.

- ** Chaque équipe doit être autonome et capable de développer et de déployer ses services avec une collaboration minimale avec les autres équipes.

1.1.3 Problèmes avec une application monolithique

Lorsqu'une application est emballée et déployée sous la forme d'un seul fichier, nous l'appelons une application monolithique. Bien qu'elle puisse avoir une structure de code modulaire dans laquelle les différentes classes sont regroupées sous différents paquets ou modules. Mais à moins qu'elles ne puissent être déployées et exécutées en tant qu'applications séparées ou indépendantes, nous nous dirigeons progressivement vers un enfer monolithique.

Au fil du temps, une application monolithique devient si complexe et si grande que les développeurs eux-mêmes trouvent qu'il est assez difficile de développer, construire et exécuter une telle application.

Le temps de vérification du code est long, le temps de construction de l'application est deux fois plus long et enfin, le temps de démarrage est également affecté en raison d'un référentiel de code aussi important.

Un autre problème de l'application monolithique est que nous ne pouvons pas faire évoluer les modules individuellement car l'application entière est déployée comme une seule unité sur un serveur. Et s'il est nécessaire de faire évoluer seulement 1 ou 2 modules, il n'est pas possible de le faire parce qu'il s'agit d'un artefact de déploiement unique.

En résumé, de telles applications sont difficiles à faire évoluer et ne sont pas fiables. Et par conséquent, le développement et les pratiques agiles qui s'ensuivent sont déréglés.

1.2 Architecture globale du module pilotage

Le module Pilotage dans la plateforme Novy doit respecter et implémenter l'architecture micro-services déjà établie et utiliser les mêmes outils et technologies pour les services et fonctionnalités similaires à celles des modules existants. Le nouveau module lui-même est composé de plusieurs services et outils, le figure 11 illustre l'architecture globale du module Pilotage, ses composants sont décrits en détail ci-après.

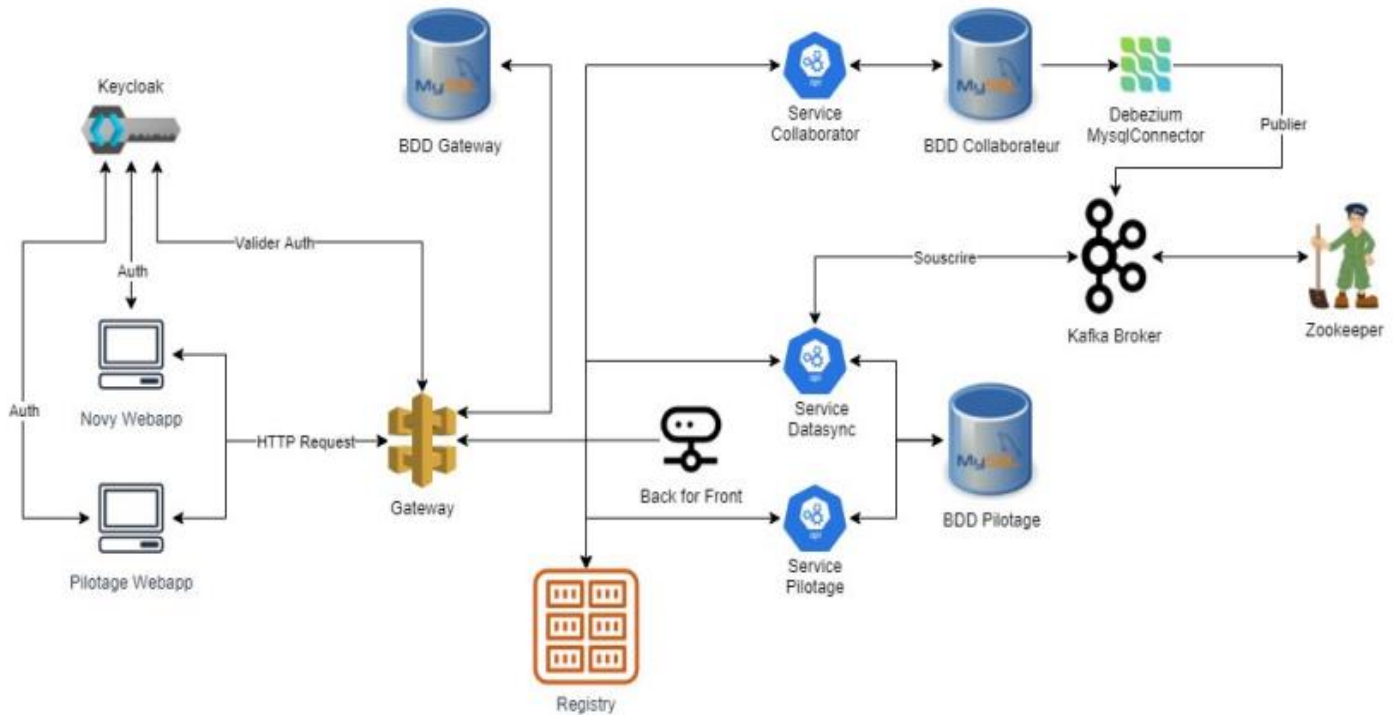


Figure 16 : Architecture globale du module Pilotage

1.2.1 Application web

Le module Pilotage a une seule application client qui est une application web. Pour son développement, les mêmes technologies Frontend React et Redux sont utilisées. Pour le développement des composants de l'interface web Material UI est utilisée, qui est une librairie open source et robuste qui implémente le système de design Material Design de Google comme des composants React facile à utiliser et personnaliser.

1.2.2 API Backend

Le Backend du module Pilotage est une API Rest utilisant les mêmes technologies que les autres modules de Novy (Java, Spring Boot, Spring Data, Spring Cloud). Elle est structurée en sous-projets pour une bonne expérience de développement et pour une maintenance facile du code. Le tableau 6 décrit cette structure de sous-projets.

Sous-projet	Description
Domain	Couche persistance, contient les classes des entités JPA de l'application
DAO	Objets d'accès aux données
Service	Couche services métier et DTO (objets de transfert de données)
API	Couche web, contient les contrôleurs REST
Boot	Configurations, utilitaires et classe de démarrage de l'application

Tableau 6 : Structure du projet Backend

1.2.3 BFF (Backend For Frontend)

Un BFF est une application intermédiaire entre la couche Backend métier et le navigateur. Ce backend supplémentaire rendra spécifique les données de l'API uniquement pour une interface utilisateur (un site web, une application mobile...). Un autre intérêt est de réduire le nombre de requêtes entre le navigateur et le serveur, afin de s'épargner les ralentissements lors des problèmes de latence réseau. Également il y aura moins de traitement des réponses et donc moins de complexité côté navigateur. De plus, il est préférable de réduire la taille de la réponse entre serveur et client. Comme le BFF est lié à une expérience utilisateur, il doit être maintenu par la même équipe qui est en charge du Frontend. Ceci permet de définir et d'adapter plus facilement les requêtes HTTP suivant le besoin du Frontend. Même si, pour certaines données, le BFF fera peut-être passe-plat sur certaines routes, il est important que chaque appel de l'application Frontend soit effectué au BFF, et lui seul communique avec l'API pour renvoyer les bonnes réponses.

Pour le développement du BFF, nous avons besoin d'une technologie légère, simple à maintenir et optimisée pour le traitement de JSON. C'est pourquoi nous avons choisi d'utiliser Node.js avec le Framework Express.

- **Node.js :**



Node (ou plus formellement Node.js) est un environnement d'exécution open-source, multi-plateforme, qui permet aux développeurs de créer toutes sortes d'applications et d'outils côté serveur en JavaScript. Cet environnement est destiné à être utilisé en dehors du navigateur. Aussi, Node ne permet pas d'utiliser les API JavaScript liées au navigateur mais des API plus traditionnellement utilisées sur un serveur dont notamment celles pour HTTP ou la manipulation de systèmes de fichier.

1.2.4 Services d'Apache Kafka

Les données gérées dans le module Pilotage dépendent directement des données générées dans le module Collaborateur. L'importation de ces données doit se faire automatiquement et en temps réel pour avoir des indicateurs de pilotage qui reflètent la situation à jour des projets. La solution qui est déjà utilisée et testée sur d'autres projets pour ce genre de situations au sein de Novelis est Apache Kafka.

- **Apache Kafka :**



Apache Kafka est une plateforme de streaming de données distribuées qui peut publier, s'abonner, stocker et traiter des flux d'enregistrements en temps réel. Elle est conçue pour traiter des flux de données provenant de plusieurs sources et les transmettre à plusieurs consommateurs. En bref, elle permet de déplacer des quantités massives de données, non seulement d'un point A à un point B, mais aussi d'un point A à un point Z et partout ailleurs, le tout en même temps.

Apache Kafka est une alternative aux systèmes de messagerie d'entreprise traditionnels. Il s'agissait à l'origine d'un système interne développé par LinkedIn pour traiter 1,4 milliard de

messages par jour, mais il s'agit aujourd'hui d'une solution de streaming de données open source qui répond à de nombreux besoins des entreprises.

Kafka est utilisé dans le module Pilotage pour lire les changements dans la base de données du module Collaborateur et les mettre dans des topics pour que le service de synchronisation de données met à jour la base de données Pilotage. Le connecteur MySQL de Debezium a été choisi contre aux autres projets au sein de Novelis qui utilisent le connecteur Confluent car le connecteur Debezium, contrairement à Confluent détecte l'évènement de suppression sur la base de données qui est très important pour notre projet.

MySQL dispose d'un journal binaire (binlog) qui enregistre toutes les opérations dans l'ordre dans lequel elles sont effectuées dans la base de données. Cela inclut les modifications des schémas de tables ainsi que les modifications des données des tables. MySQL utilise le binlog pour la réplication et la récupération.

Le connecteur MySQL Debezium lit le binlog, produit des événements de changement pour les opérations INSERT, UPDATE et DELETE au niveau des lignes, et émet les événements de changement vers les topics Kafka.

Comme MySQL est généralement configuré pour purger les binlogs après une période de temps spécifiée, le connecteur Debezium effectue une capture initiale cohérente de la base de données. Le connecteur Debezium lit le binlog à partir du moment où la capture a été réalisée.

1.2.5 Service de synchronisation de données

La solution Apache Kafka nous permet de synchroniser les données entre les bases de données des modules Pilotage et Collaborateur, mais dans Pilotage nous n'utilisons pas les données provenant de Collaborateur directement sauf pour certains cas comme la liste des collaborateurs, les types et les statuts des projets. Ces données servent donc principalement à réaliser plusieurs calculs pour déduire les indicateurs de pilotage à enregistrer. Le service de synchronisation de données est un projet qui est composé de plusieurs fichiers de

configuration pour lancer et gérer les services de Kafka et une application Spring pour effectuer les calculs.

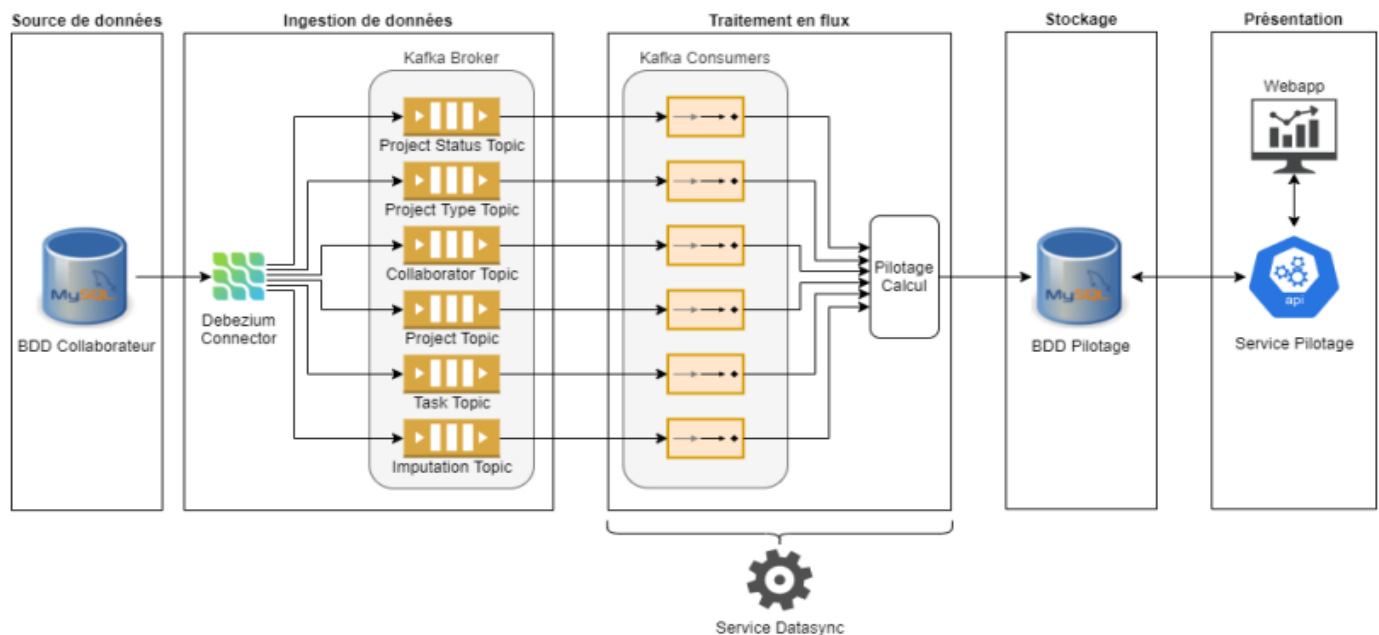


Figure 17 : Schéma de synchronisation de données

Le schéma de déploiement sur la figure suivante la collecte en temps réel de données provenant de la base de données des collaborateurs. Il les traite périodiquement en appliquant un ensemble de calculs de pilotage, puis stocke ces données et calculs dans une nouvelle base de données prête à être interrogée par la couche de présentation.

- **Couche “Source de données”** : La première couche est celle de la source de données, à partir de laquelle les données en continu sont générées. Dans cette solution, les données des collaborateurs, des tâches et projets résident dans des bases de données.
- **Couche “Ingestion de données”** : Les données sont injectées dans Kafka à travers le connecteur de Debezium, dont chaque table correspond à un Topic de Kafka.
- **Couche “Traitement en flux”** : Les données injectées sont consommées depuis les Topics de Kafka, et passent par une boîte noire de calculs ou elles seront transformées

- **Couche “Stockage”** : Après le traitement en flux, les résultats de calculs sont stockés dans une base de données final.
- **Couche “Présentation”** : Dans laquelle, le service de pilotage interroge la base de données finale et envoie les résultats vers tableau de bord web pour la visualiser et interagir avec ces données.

1.2.6 Authentification

Fournir une authentification pour les composants non publics d’une application est une partie importante de nombreux systèmes. Keycloak est une solution de gestion du SSO (Single Sign-On). Le SSO est une propriété du contrôle d'accès de plusieurs systèmes logiciels connexes, mais indépendants.

Cela signifie que nous pouvons utiliser un seul système d'authentification pour permettre aux utilisateurs de se connecter à plusieurs services avec les identifiants, sans avoir besoin de fournir de mot de passe au service auquel ils se connectent.

Comme dans les autres modules de Novy, Pilotage utilise Keycloak pour l’authentification et l’autorisation. La figure 13 illustre le schéma d’authentification par Keycloak.

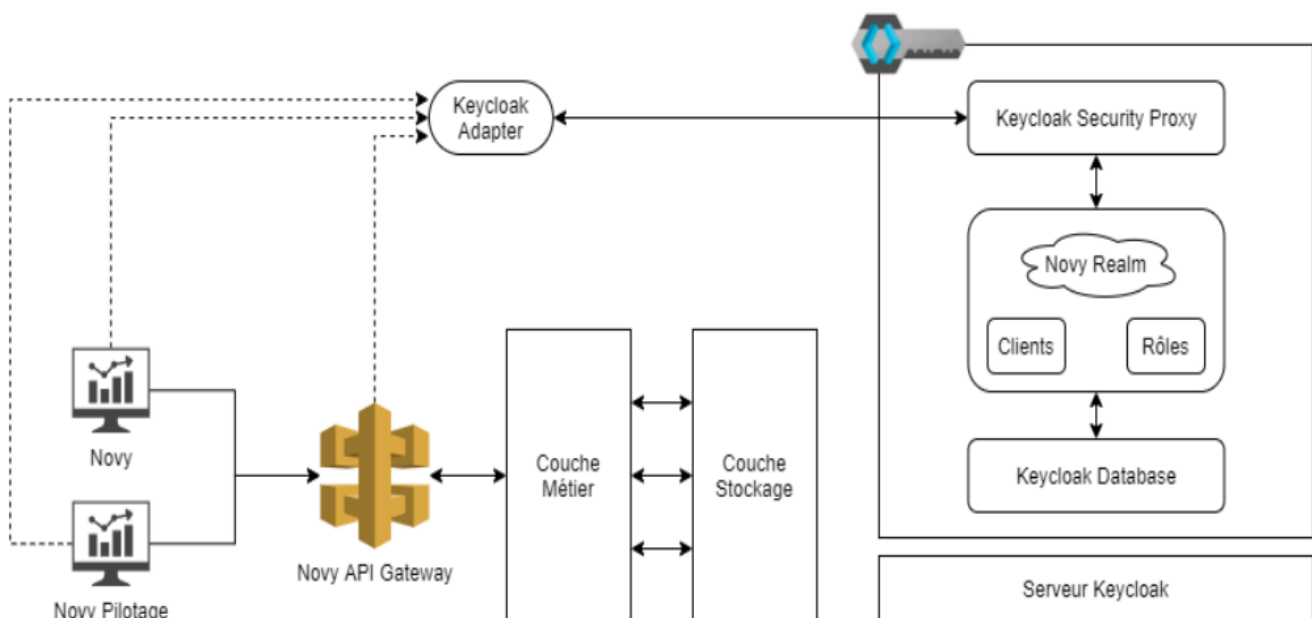


Figure 18 : d'authentification par Keycloak

1.3 Environnement et outils de développement

Au sein de Novelis, plusieurs piles techniques sont utilisées selon le type de projet. Pour les projets internes et plus particulièrement pour le projet Novy, les outils et technologies listés dans ce qui suit sont utilisés. Cette pile technique est choisie pour les exigences de sécurité, diversification et l'évolution de la plateforme. Pour le développement du module de pilotage des projets, les mêmes outils et technologies seront utilisés.

1.3.1 Backend

- **Java / Jakarta EE :**



Il s'agit d'un ensemble de spécifications pour écrire des applications d'entreprise avec Java. Ces spécifications sont des documents définissant l'API et l'interaction d'une technologie. En langage Java, les documents de spécification sont comme des interfaces avec des définitions formelles et l'implémentation réelle de ces interfaces est faite par un fournisseur de serveur d'application (par exemple WildFly, Open Liberty, Payara, TomEE).

Depuis de nombreuses années, Java EE est le moyen standard d'écrire des applications d'entreprise. Java EE (appelé J2EE à l'époque) a longtemps été maintenu par Oracle. En 2017, Oracle a décidé de ne plus faire évoluer le standard d'entreprise sous son égide et l'a remis à la Fondation Eclipse. Pour des raisons juridiques, ils ont dû changer de nom et ont décidé de le nommer Jakarta EE.

- **Spring :**



Spring est le cadre de développement d'applications Java le plus populaire pour les entreprises. Des millions de développeurs dans le monde utilisent Spring Framework pour créer un code performant, facilement testable et réutilisable.

Les fonctionnalités de base du Spring Framework peuvent être utilisées pour développer n'importe quelle application Java, mais il existe des extensions pour la création d'applications Web au-dessus de la plate-forme Java EE. Spring Framework vise à rendre le développement J2EE plus facile à utiliser et à promouvoir les bonnes pratiques de programmation en permettant un modèle de programmation basé sur les POJO.

Pour la plateforme Novy, les projets de Spring suivants sont utilisés : Spring Boot, Spring Data, Spring Cloud, Spring Security

- **Liquibase:**



Liquibase est un outil open-source de migration de base de données et Spring Boot est un framework populaire pour le développement d'applications Java. Lorsqu'ils sont utilisés ensemble, Liquibase et Spring Boot offrent un moyen pratique de gérer les modifications de schéma de base de données et le contrôle des versions au sein des applications Spring Boot.

Liquibase vous permet de définir des changements de schéma de base de données en utilisant des formats XML, YAML ou SQL. Il garde une trace des modifications que vous apportez à votre schéma de base de données au fil du temps et vous aide à appliquer ces modifications de manière contrôlée et cohérente.

En utilisant Liquibase avec Spring Boot, vous pouvez vous assurer que le schéma de base de données de votre application évolue en même temps que votre code, ce qui facilite la gestion et la maintenance de la structure de la base de données dans différents environnements et versions de votre application.

1.3.2 Frontend

- **React.js :**



React (parfois appelé React.js ou ReactJS) est une bibliothèque JavaScript pour créer des interfaces utilisateur. Il est maintenu par Facebook et une communauté de développeurs et d'entreprises. Le plus grand avantage revendiqué de React est l'accessibilité - à la base tout développeur connaissant HTML peut créer des applications React. Un autre avantage revendiqué est la possibilité de créer des applications Web et mobiles en utilisant la même pile technologique. Plusieurs entreprises utilisent la bibliothèque React with Redux, qui permet aux développeurs de créer des applications Web complexes mais évolutives.

- **Redux :**



Redux est un conteneur d'état prévisible pour les applications JavaScript. Il aide à écrire des applications qui se comportent de manière cohérente, s'exécutent dans des environnements différents (client, serveur et natif) et sont faciles à tester. En plus de cela, il offre une excellente expérience de développement, telle que l'édition de code en direct combinée à un débogueur dans le temps. On peut utiliser Redux avec React, ou avec n'importe quelle autre bibliothèque de vues.

1.3.3 Bases de données

- **MySQL :**



MySQL, initialement développé par MySQL AB, maintenant soutenu par Oracle, est un système de gestion de base de données relationnelle (SGBDR), basé sur le modèle client-serveur.

1.3.4 Authentification et autorisation

- **Keycloak :**



Keycloak est un outil de gestion des identités et des accès (IAM). L'IAM vise généralement à vérifier l'identité d'un utilisateur ou d'un système qui demande l'accès à notre environnement, et évalue un ensemble de règles qui indiquent à quelles fonctionnalités et à quels actifs cet utilisateur/ce système a accès. La gestion des identités et la gestion des accès sont deux entités différentes. Dans les situations classiques, la gestion de l'accès s'appuie sur un fournisseur d'identité pour vérifier l'utilisateur, et s'appuie sur celui-ci pour faire correspondre l'identité de l'utilisateur/du système avec ce qu'il peut faire dans un environnement. En général, le contrôle d'accès basé sur les rôles (RBAC) est un concept standard qui est suivi pour établir une couche de contrôle d'accès. Une couche de support est ajoutée pour utiliser cette gestion d'accès, essentiellement un ensemble de protocoles pour l'utiliser avec d'autres applications.

En général, le protocole finit par être OAuth, qui est une norme responsable de la gestion des paramètres de contrôle d'accès du côté du logiciel d'autorisation, ainsi que des directives à suivre par les applications utilisant le service d'autorisation.

1.3.5 DevOps

- **Docker :**



Docker est une plateforme qui regroupe une application et toutes ses dépendances sous forme de conteneurs. Cet aspect de conteneurisation de Docker garantit que l'application fonctionne dans n'importe quel environnement.

Chaque application s'exécute sur des conteneurs séparés et possède son propre ensemble de dépendances et de bibliothèques. Cela permet de s'assurer que chaque application est indépendante des autres applications, ce qui rassure les développeurs quant à la possibilité de créer des applications qui n'interfèrent pas entre elles.

Un développeur peut donc créer un conteneur sur lequel sont installées différentes applications et le remettre à l'équipe d'assurance qualité. Il suffira alors à l'équipe d'assurance qualité d'exécuter le conteneur pour reproduire l'environnement du développeur.

- **Jenkins :**



Jenkins est un serveur d'automatisation populaire, autonome et open-source, qui permet de réaliser l'intégration continue et l'automatisation de la construction. Sa fonctionnalité élémentaire consiste à exécuter une liste prédéfinie de phases ou de tâches. En d'autres termes, chaque changement dans un référentiel déclenche un pipeline de travaux qui évalue et exécute différentes tâches pour accomplir ce qui a été préalablement défini.

Chaque phase est surveillée et permet d'arrêter l'ensemble du processus et le changement sera signalé à l'utilisateur par Jenkins. Dans les grandes entreprises, il est courant que plusieurs équipes travaillent sur un même projet sans savoir ce que les autres équipes font sur la même base de code. Ces changements peuvent créer des bogues qui ne seront révélés que lorsque les deux codes seront intégrés dans la même branche. Puisque Jenkins peut exécuter ses tâches prédéfinies pour chaque commit, il sera en mesure de détecter et de notifier aux développeurs que quelque chose ne va pas et où cela se trouve.

1.3.6 Outils de développement

- **IntelliJ IDEA :**



IntelliJ IDEA est un environnement de développement intégré (IDE) utilisé principalement pour la programmation en Java, bien qu'il supporte également d'autres langages de programmation tels que Kotlin, Scala, Groovy et bien d'autres. Développé par JetBrains, IntelliJ IDEA offre une gamme de fonctionnalités avancées qui aident les développeurs à améliorer leur productivité et à simplifier leur flux de travail.

- **Git & GitLab :**



Git est un logiciel permettant de suivre les modifications dans n'importe quel ensemble de fichiers, généralement utilisé pour coordonner le travail entre les programmeurs développant en collaboration le code source pendant le développement du logiciel. Ses objectifs incluent la vitesse, l'intégrité des données et la prise en charge des flux de travail distribués et non linéaires.

GitLab est une plateforme web basée sur Git qui permet la gestion collaborative de projets logiciels. Elle offre un ensemble complet d'outils, tels que la gestion des dépôts Git, le suivi des problèmes, les demandes de fusion (merge requests), l'intégration continue (CI/CD), et bien d'autres fonctionnalités. GitLab peut être utilisé en tant que service cloud hébergé ou être déployé sur des serveurs locaux pour une utilisation en interne.

1.4 Présentation de l'application

1.4.1 Améliorations « module pilotage coté opérationnel »

Novy est une plateforme importante pour la gestion interne chez Novelis, elle est utilisée chaque jour par les managers et les collaborateurs. Cela pousse la suggestion des nouvelles

fonctionnalités, corrections et améliorations régulièrement. La première étape de notre travail sur le sujet consistait à effectuer des améliorations sur Novy et plus particulièrement sur le module pilotage des projets coté opérationnel. Autre que le but de réaliser les corrections et les améliorations, le travail sur le module existant nous a permis premièrement de mieux comprendre l'architecture de la plateforme et de se familiariser encore plus avec la pile technique utilisée au sein de Novelis, et deuxièmement d'effectuer des améliorations nécessaires pour l'implémentation du nouveau module de pilotage des projets côté financier. Je cite ci-dessous quelques améliorations effectuées dans la partie opérationnel :

1.4.1.1 La vue Actions

La section Actions contient la liste des actions de tous les projets. La liste est paginée et offre la fonctionnalité de recherche des tris et des filtres par responsable, échéance, statut, type et priorité des actions. On peut modifier ou supprimer une action rapidement sur la liste sans avoir à entrer dans la vue de détails de l'action, on peut aussi ajouter des actions ou exporter les actions en format Excel.

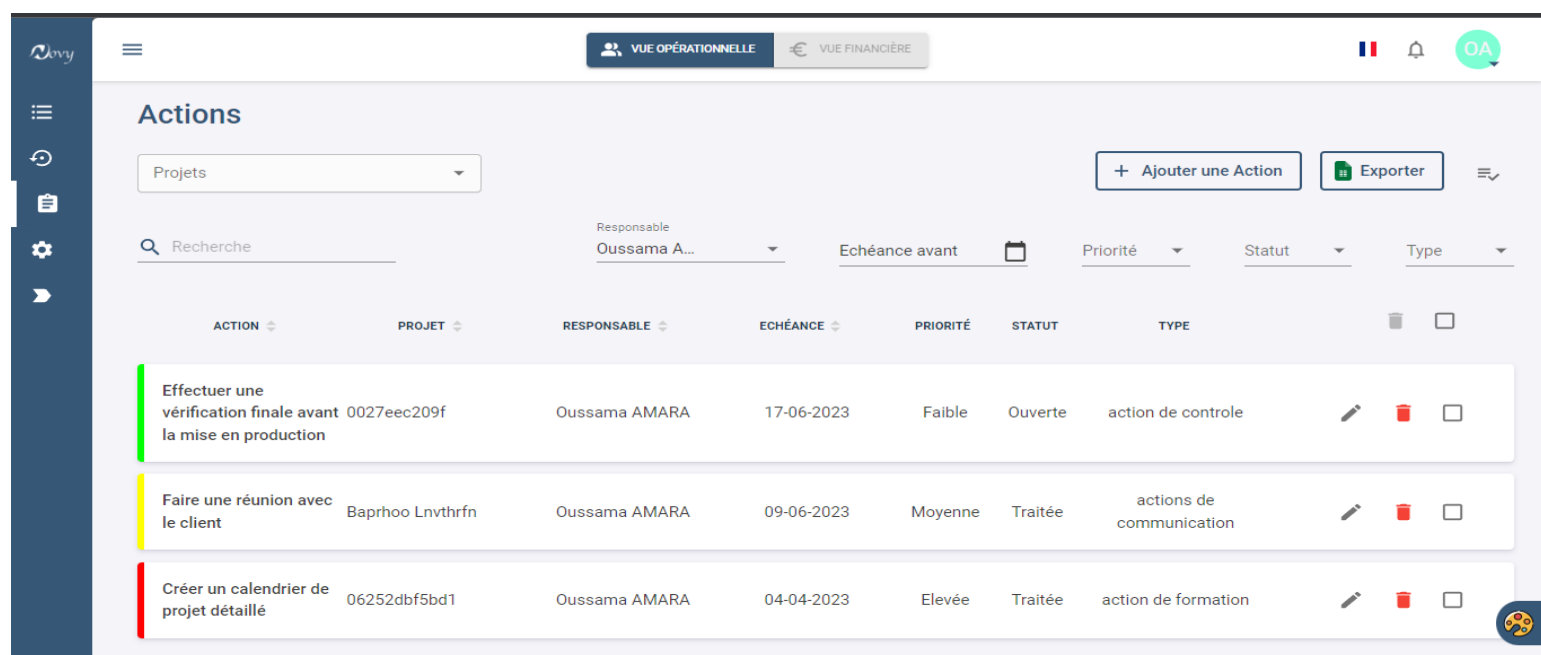


Figure 19 : interface globale des actions

** Zoom sur le bloc des filtres



- Le filtre **Responsable** : le filtre “Responsable” est prérempli avec le nom du manager connecté. Le manager a ensuite le droit de vider le filtre et saisir le(s) nom(s) de(s) responsable(s).
- Le filtre **Echéance_avant** : permet de filtrer les actions en choisissant la date limite pour l’accomplir.
- Le filtre **Priorité** : permet de filtrer les actions selon leur priorité : Faible, Moyenne ou Elevée (choix multiple inclus).
- Le filtre **Statut** : permet de filtrer les actions en se basant sur leur statut : Ouverte, En cours, Traitée (choix multiple inclus).
- Le filtre **Type** : permet de filtrer les actions en se basant sur leur type : Projet ou risque (choix multiple inclus).

** Ce que j'ai effectué

- Ajouter le filtre en utilisant la période.
- Correction des bugs.
- L’ajout du colonne type d’action et la possibilité de filtrer avec cette colonne.

1.4.1.2 Ajout d’une action

Par défaut, le responsable sera informé de l’action avant sa date d’échéance afin qu’il puisse la réaliser dans les délais impartis.

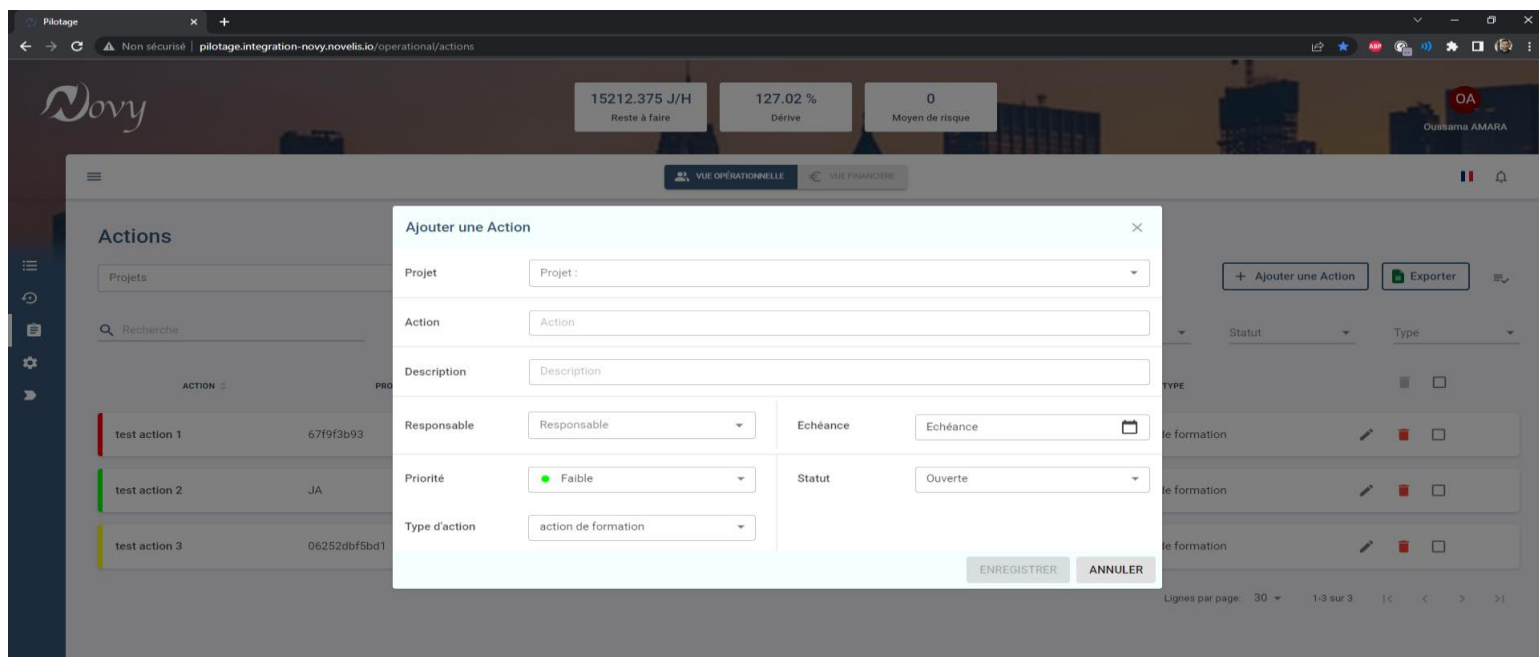


Figure 20 : ajout d'une action

** Ce que j'ai effectué

- Ajouter la colonne type d'action dans le modal d'ajout
- Correction des bugs

1.4.1.3 La vue en Kanban

La vue kanban des actions offre la possibilité de changer le statut d'une action tout simplement en utilisant Drag & Drop.

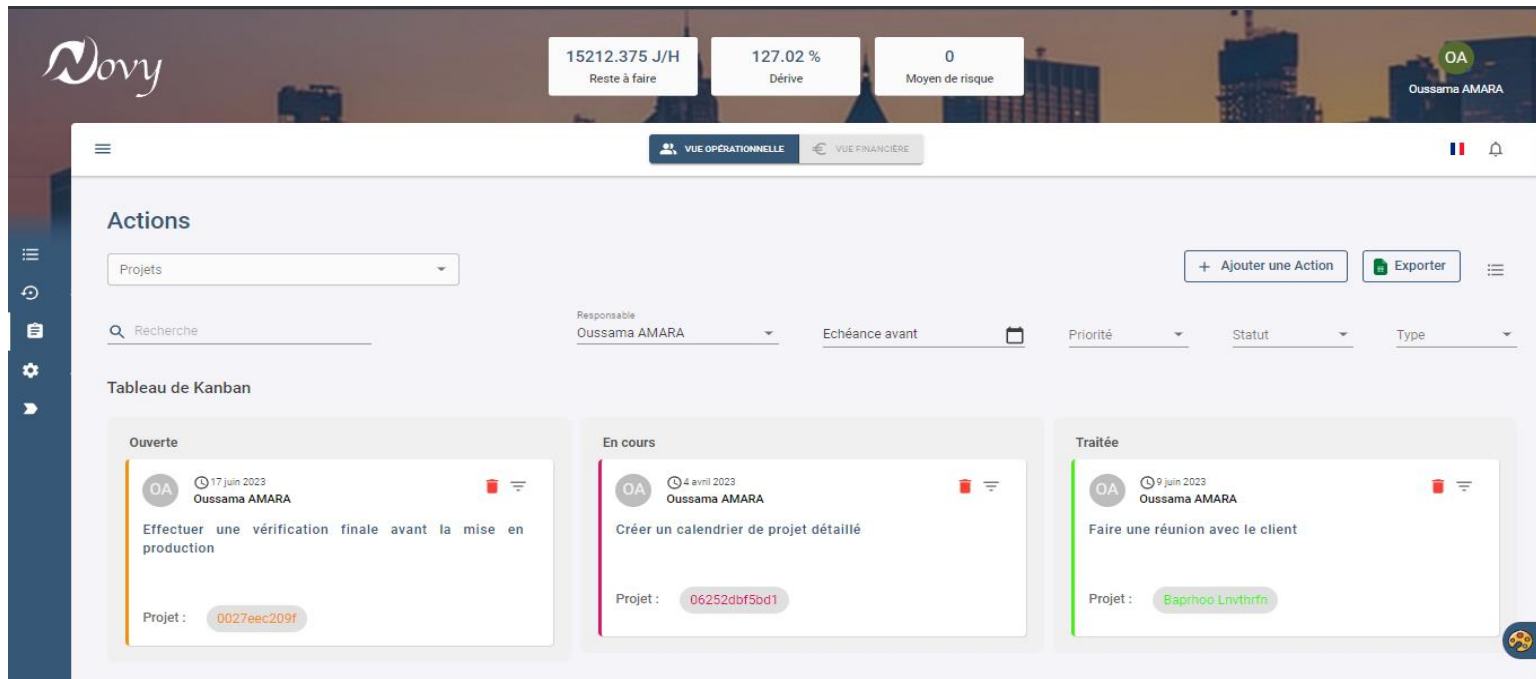


Figure 21 : vue en Kanban des actions

Lorsqu'une action est cliquée, Une fenêtre apparaît au centre de l'écran affichant les informations sur l'action.

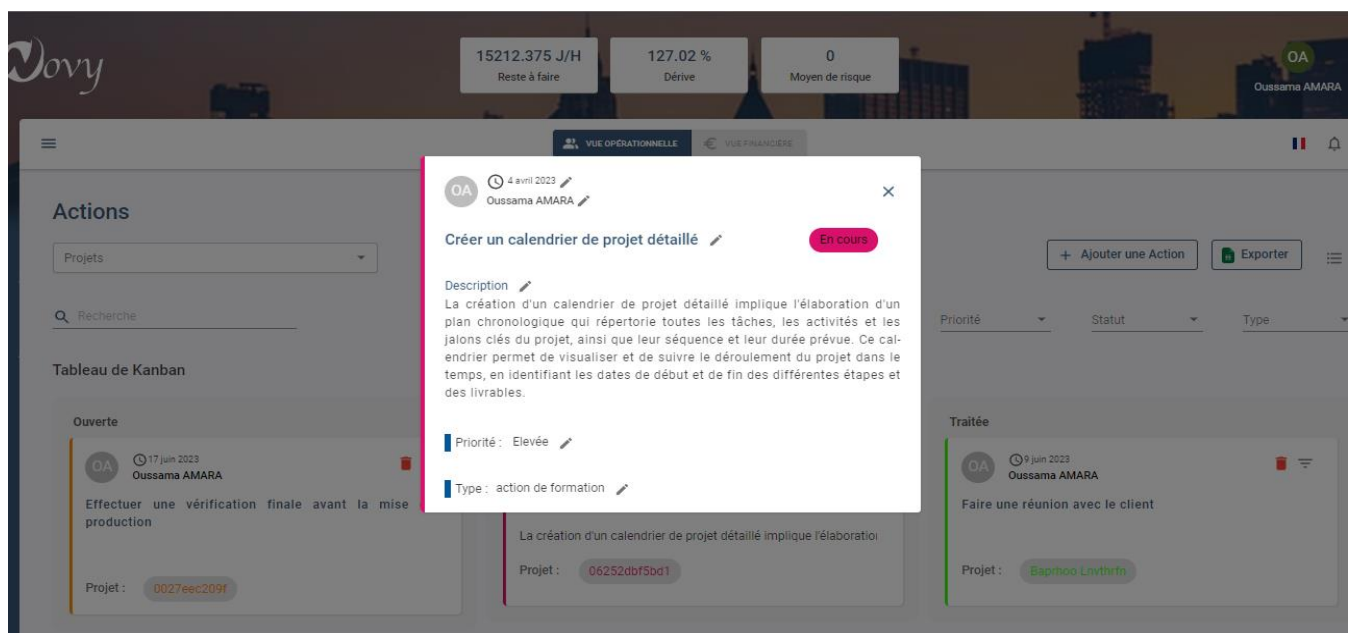


Figure 22 : détail d'action dans le vue kanban

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.1.4 La vue Risques

Cet écran permet au manager de visualiser les risques associés aux différents projets. En accédant à l'écran "Risques", le filtre "Responsable" est prérempli avec le nom du manager connecté. Le manager a ensuite le droit de vider le filtre pour afficher tous les risques.

The screenshot displays the 'Risques' (Risks) management interface. At the top, there are tabs for 'VUE OPERATIONNELLE' (selected) and 'VUE FINANCIERE'. A user profile 'OA' is visible in the top right. The main section is titled 'Risques' and includes a 'Projets' dropdown menu. Below this is a search bar with a 'Recherche' button. A filter bar shows 'Responsable' set to 'Oussama AMARA'. The main table lists risks with columns: RISQUE, PROJET, RESPONSABLE, ECHÉANCE, PRIORITÉ, STATUT, and CATÉGORIE. Four risks are listed, all with a due date of 15-06-2023. The first risk is 'Risque de perte de données' (High priority, Identified status, Financial risk). The second is 'Risque de cyberattaques' (Low priority, Closed status, Operational risk). The third is 'Risque de retard dans le développement' (High priority, Closed status, Financial risk). The fourth is 'Risque de manque de compétences techniques' (Medium priority, In progress status, Financial risk). The interface also includes a 'Lignes par page: 30' dropdown and a '1-4 sur 4' indicator.

Figure 23 : interface globale des risques

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.1.5 Actions d'un risque

Sur la vue en liste, le manager est en mesure de voir les actions liées à un risque.

** Ce que j'ai effectué

- Implémenter la vue en entier

The screenshot shows a web application interface with a sidebar on the left and a main content area. The main content area is titled 'VUE OPERATIONNELLE' and displays a table of risks. Below the table, there is a section for 'Actions' with a table of actions associated with the selected risk.

Risque	ID	Responsable	Echéance	Priorité	Statut	Type
Risque de cyberattaques	0027eec209f	Oussama AMARA	15-06-2023	Faible	Cloturé	risque opérationnel
Risque de retard dans le développement	0027eec209f	Oussama AMARA	15-06-2023	Elevée	Cloturé	Risque financier
Risque de manque de compétences techniques	0027eec209f	Oussama AMARA	15-06-2023	Moyenne	En cours	Risque financier

Action	Responsable	Echéance	Priorité	Statut	Type
Communication efficace	Ali Sadeq		Faible	Ouverte	action de formation
Formation et développement	ABED SOUBRA	10-06-2023	Faible	Ouverte	action de formation
Évaluation des compétences existantes	Abdelaziz SAROUI	07-06-2023	Faible	Traitée	action de formation

Figure 24 : les actions d'un risque

1.4.1.6 Détail d'action associé à risque

Cet écran permet de voir les informations d'une action associé à un risque.

The screenshot shows a modal window titled 'Formation et développement' with a close button (X). The modal contains a table with details about the action and its associated risk.

Action	Risque
Formation et développement	Risque de manque de compétences techniques

Description	L'action de formation et de développement vise à renforcer les compétences et les connaissances des membres de l'équipe afin de réduire le risque de manque de compétences techniques dans le projet. Cette action implique la mise en place de programmes de formation et de développement appropriés pour combler les lacunes identifiées et améliorer les compétences de l'équipe.		
Responsable	ABED SOUBRA	Echéance	10-06-2023
Priorité	Faible	Statut	Ouverte
Type d'action	action de formation		

🔔 Rappels d'échéance activés

MODIFIER SUPPRIMER

Figure 25 : les actions d'un risque

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.1.7 Commentaire de clôture



Figure 26 : commentaire de clôture

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.1.8 La vue Historique

La vue historique des snapshots des projets englobe à la fois les instantanés pris par l'utilisateur et ceux générés automatiquement par le système.

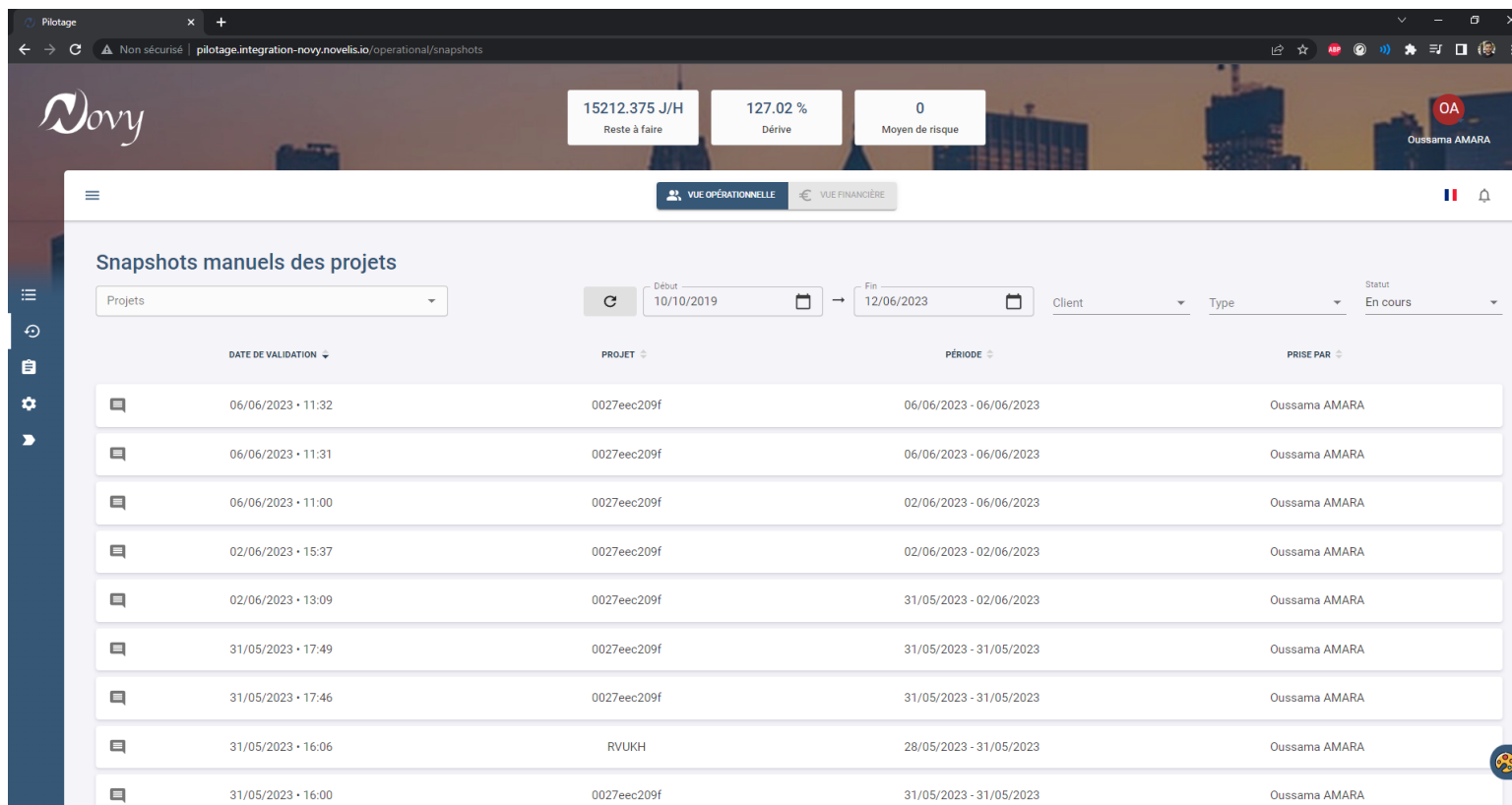


Figure 27 : La vue Historique des snapshots

** Ce que j'ai effectué

- L'ajouter le filtre par période
- Corrections des bugs

1.4.1.9 La vue Synthèse

C'est la vue par défaut, elle présente tous les indicateurs d'un projet sous forme d'un tableau de bord. En utilisant cet écran le manager peut voir les indicateurs d'un projet en temps réel.

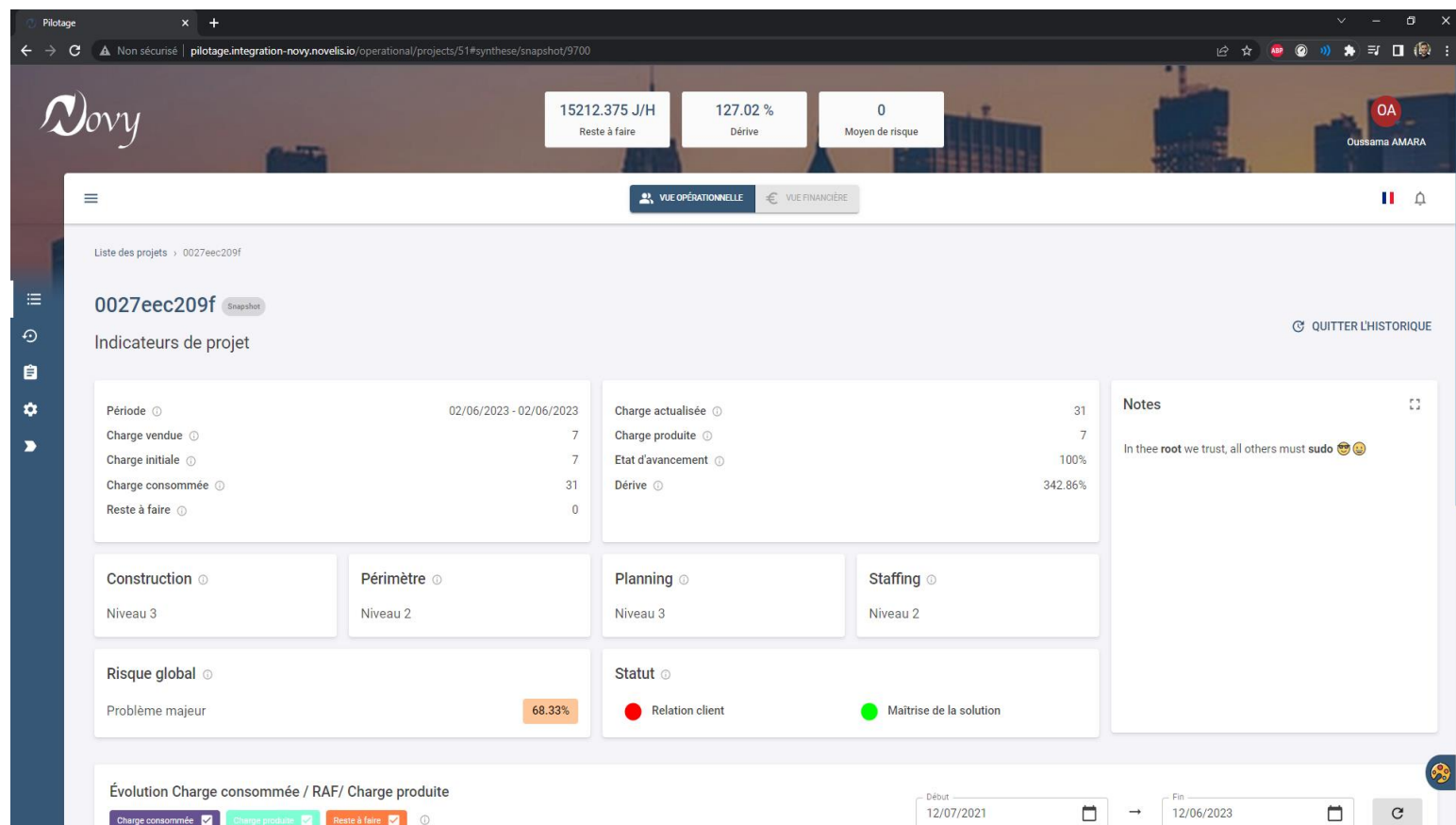


Figure 28 : les indicateurs actuels d'un projet

1.4.2 Nouvelles fonctionnalités « module pilotage côté financier »

Après avoir présenté quelques interfaces concernant le suivi opérationnel d'un projet maintenant je vais présenter les nouvelles fonctionnalités dans le module de pilotage des projets qui est le suivi financier.

1.4.2.1 Gestion des commandes

Cette vue offre un aperçu des commandes exécutées dans un projet, aussi il y a la possibilité de faire des modifications sans accéder au détail d'une commande.

0027eec209f

Commandes

+ Ajouter une commande

Recherche

Nature de vente Type de vente

NOM	MONTANT	CHARGE VENDU	NATURE DE VENTE	TYPE DE VENTE	FERMÉE
Conception initiale du logiciel	12345	12345	INITIAL	SERVICE	<input type="checkbox"/>
Développement du noyau du logiciel	12345	12345	INITIAL	SERVICE	<input type="checkbox"/>
Intégration des fonctionnalités principales	12345	12345	INITIAL	SERVICE	<input type="checkbox"/>

Lignes par page: 30 1-3 sur 3

Figure 29 : la vue des commandes d'un projet

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.2.2 Les échéanciers d'une commande

Cette vue donne la main au manager de voir les échéanciers qui sont associé à une commande

Et pouvoir les manipuler facilement en utilisant les icônes à coté de chaque échéancier

0027eec209f

Commandes

+ Ajouter une commande

Recherche

Nature de vente Type de vente

NOM	MONTANT	CHARGE VENDU	NATURE DE VENTE	TYPE DE VENTE	FERMÉE
Conception initiale du logiciel	12345	12345	INITIAL	SERVICE	<input type="checkbox"/>

+ AJOUTER UN ÉCHÉANCIER

Échéanciers

NOM	MONTANT PRÉVISIONNEL HT	DATE PRÉVISIONNELLE DE FACTURATION	CHARGE PRÉVISIONNEL	STATUT
Traitement de la commande	12345	2023-06-03	12345	ESTIMATED
Livraison de la conception initiale	12345	2023-06-10	12345	ESTIMATED
Développement du noyau du logiciel	12345	12345	INITIAL	SERVICE

Figure 30 : la vue des échéanciers d'une commande

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.2.3 Ajout d'une commande

Le manager doit remplir tous les champs afin d'ajouter une commande, lorsqu'il entre des données valides, le Button d'enregistrer va être apparaitre pour faire l'ajout

Figure 31 : l'ajout d'une commande

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.2.4 Ajout d'un échéancier

Le manager doit remplir tous les champs prévisionnels afin d'ajouter un échéancier pour une commande, lorsqu'il entre des données valides, le Button d'enregistrer va être apparaitre pour faire l'ajout. Noté que le nom de commande est prés remplis par la commande concerné.

Ajouter un Echéancier
0027eec209f

Nom	<input type="text" value="Nom"/>	Commande de vente	<input type="text" value="Commande de vente"/>
Code	<input type="text" value="Code"/>	Description	<input type="text" value="Description"/>
Montant prévisionnel HT	<input type="text" value="Montant prévisionnel HT"/>	Montant réel	<input type="text" value="Montant réel"/>
Date prévisionnelle de facturation	<input type="text" value="Date prévisionnelle de facturation"/>	Date réelle de facturation	<input type="text" value="Date réelle de facturation"/>
Charge prévisionnelle	<input type="text" value="Charge prévisionnel"/>	Charge réelle	<input type="text" value="Charge réelle"/>
Numéro de facture	<input type="text" value="Numéro de facture"/>	Statut	<input type="text" value="Statut"/>

ENREGISTRER ANNULER

Figure 32 : l'ajout d'un échéancier

** Ce que j'ai effectué

- Implémenter la vue en entier

1.4.2.5 Vue d'administration (Type de vente)

Administration

Tables de référence Paramètres Gestion d'accès

trise de la solution Priorité d'action Statut d'action Type d'action Statut de risque Priorité de risque Probabilité de risque Catégorie de risque Profile de coût Type de vente

NOM	CODE	Importer	Exporter	Ajouter
LICENCE	LICENCE			
SERVICE	SERVICE			
SOUS_TRAITANCE	SOUS_TRAITANCE			
MATERIEL	MATERIEL			

Figure 33 : vue administration (Type de vente)

**** Ce que j'ai effectué**

- Ajouter la CRUD pour les tables de référence :
 - Type d'action
 - Statut de risque
 - Priorité de risque
 - Probabilité de risque
 - Catégorie de risque
 - Type de vente

1.5 Conclusion

Ce chapitre a décrit la dernière étape du développement qui est celle de la réalisation et la mise en œuvre du projet. J'ai présenté dans un premier temps l'architecture existant du projet, l'environnement et les outils de développement choisis. En fin, on a présenté le résultat final de la réalisation de l'application.

Conclusion Générale

Dans le cadre de mon projet de fin d'études, j'ai intégré une équipe au sein de la société Novelis pour réaliser des améliorations au niveau d'un outil de pilotage des projets coté opérationnel et le développement d'une nouvelle partie qui est le suivi financier des projets.

Le présent rapport détaille toutes les étapes par lesquelles nous sommes passés pour arriver au résultat obtenu. Nous avons commencé par comprendre le contexte général du projet et identifier ses différentes exigences, puis nous avons fait une analyse de l'existant pour comprendre ses fonctionnalités, son architecture et les technologies utilisées. Ensuite nous avons réalisé plusieurs améliorations sur les modules existants de la plateforme avant de faire l'étude et la réalisation du nouveau module

Ce travail m'a été très instructif, puisqu'il m'a permis de découvrir un nouveau domaine de travail qui est la gestion d'entreprise et ses processus en général et particulièrement le pilotage des projets, qui est d'une grande importance pour les entreprises. Outre, ce projet m'a permis d'approfondir mes connaissances dans les bonnes pratiques de la gestion de projet vu que nous avons eu l'opportunité d'organiser son déroulement dès les premières phases.

Notre travail est encore d'actualité et ne s'arrête pas à ce niveau. En effet, plusieurs perspectives s'offrent à ce projet pour ajouter plusieurs fonctionnalités et mieux répondre aux exigences de la société. Le module développé restera donc un sujet à développer et à explorer davantage.

Finalement, vu l'accomplissement du projet, je souhaite très fortement qu'il soit le fruit du progrès, de l'évolution et qu'il soit à la hauteur des exigences de la société, j'espère par ailleurs avoir gagné la satisfaction de ma société d'accueil Novelis et de tous ses responsables, ainsi que les membres de jury.

Modèle Bibliographie

[Material-UI] *Material-UI React components library [Software]*, v4.12.1, <https://mui.com/>, Disponible sur : <https://mui.com/getting-started/installation/>, Last updated on June 25, 2021

[Eich, AA] Brendan Eich , *javascript*, 1995, [En ligne] Date de dernière mise à jour : juin 2020
Disponible sur : <https://www.w3schools.com/js/> 25 juin 2021

[IntelliJ IDEA] *IntelliJ IDEA IDE [Software]*, v2023, <https://www.jetbrains.com/idea/>,
Disponible sur : <https://www.jetbrains.com/idea/download/>, June 13, 2023

[UML2] *Unified Modeling Language: Superstructure*, version 2.1.1. *OMG Specification*, formal/07-02- 05, <http://www.uml.org/> , Disponible sur : <http://www.omg.org/docs/formal/07-02-05.pdf> , 01 juin 2021

[React] *JavaScript Library [Software]*, v17.0.2, <https://reactjs.org/>, Disponible sur : <https://reactjs.org/docs/getting-started.html>, Last updated on March 18, 2021

[Java] *Development Kit (JDK) [Software]*, v17, <https://www.oracle.com/>, Disponible sur : <https://www.oracle.com/java/technologies/javase-jdk17-downloads.html>, April 12, 2022

[Docker] *Containerization Platform [Software]*, v24.0.2, <https://www.docker.com/>, Disponible sur : <https://www.docker.com/get-started>, Last updated on May 26, 2023

[Spring Framework] *spring framework for java [software]*, v5.3.10, <https://spring.io/projects/spring-framework>, Disponible sur : <https://spring.io/projects/spring-framework#learn>, Last updated on Avril 13, 2023