



- PFE
- SI
- SIP

# Rapport de Stage

## Sujet:

**Contribution à la mise en place de la plateforme Requestor du projet Digital Procurement**

## résumé :

*Le projet consiste à participer au développement du portail Requestor qui s'inscrit dans le cadre du projet Digital Procurement. Il s'agit d'une solution web dont l'objectif est de permettre aux chefs de projet du groupe OCP d'avoir plus de transparence sur le processus d'approvisionnement à travers le suivi de leurs demandes d'achat, de leurs commandes et de leurs factures. Cela permettra aux chefs de projet d'évaluer la performance de leurs fournisseurs et de leurs services.*

### Réalisé par :

El Mehdi BOUMHICHA 3GI

Mohamed LMOUSSAOUI 3GI

### Encadré par :

Pr. Malika ADDOU (EHTP)

M. Abdelhaq EL AIBI (OCP SA)

**Année universitaire: 2017- 2018**



**Groupe OCP SA**

2-4, rue Al Abtal, Hay Erraha, 20200, Casablanca

## Dédicace

*Avec immense joie que je dédie ce modeste travail*

### *A Mes parents*

*Pour l'éducation qu'ils m'ont prodiguée ; avec tous les moyens et au prix de tous les sacrifices, pour le sens du devoir qu'ils m'ont enseigné depuis mon enfance, pour leur générosité et leur présence permanente, pour l'inspiration de ma vivacité, et la motivation de mon effort.*

### *A Mes sœurs*

*Pour l'affection et la complicité qui nous lie, pour l'intérêt que vous portez à ma vie, pour vos encouragements et pour tout ce que vous avez fait pour moi. Votre aide, votre générosité extrême et votre soutien ont été pour moi une source de courage, de confiance et de patience. Que Dieu vous protège et vous procure santé, bonheur et réussite, et que notre climat familial s'éternise.*

### *A Mes Chers Amis*

*Pour toute la joie et la bonne humeur que vous insufflez dans ma vie. Puisse notre amitié se pérenniser. Que Dieu vous donne santé et bonheur dans votre vie familiale, personnelle et professionnelle.*

*Trouvez en ce travail le fruit de votre dénouement, de votre patience et l'expression de ma gratitude et mon profond amour.*

*Mohamed LMOUSSAOUI*

## Dédicace

*Avec immense joie que je dédie ce modeste travail*

### *A Mes Chers parents*

*Pour tous vos sacrifices pour moi, mes biens aimés, nul mot ne saura exprimer ma gratitude envers vous. Veuillez trouver ici ma reconnaissance envers vous.*

*Pour votre amour, vos prières, votre éducation, Que Dieu vous protège et vous accorde une longue vie.*

### *A Mes Chers frères*

*Pour l'affection et la complicité qui nous lie, pour l'intérêt que vous portez à ma vie, pour vos encouragements et pour tout ce que vous avez fait pour moi. Votre soutien et votre affection m'ont été d'un grand secours tout au long de ma vie. Que Dieu vous protège et vous procure santé, bonheur et réussite, et que notre climat familial s'éternise.*

### *A Mes Chers Amis*

*Pour toute la joie et la bonne humeur que vous insufflez dans ma vie. Puisse notre amitié se pérenniser. Que Dieu vous donne santé et bonheur dans votre vie familiale, personnelle et professionnelle.*

### *A toute ma famille*

*Pour son aide et soutien pendant toute ma période de formation.*

*Trouvez en ce travail le fruit de votre dénouement, de votre patience et l'expression de ma gratitude et mon profond amour.*

*El Mehdi BOUMHICHA*

## Remerciements

Avant d'entamer ce rapport, nous tenons à témoigner notre profonde gratitude à toutes les personnes qui ont participé de près ou de loin à l'élaboration de ce travail.

Nous souhaitons faire part de notre reconnaissance en premier lieu à notre encadrante à l'EHTP le **Pr. Malika ADDOU** qui a fait preuve de disponibilité à chaque fois que nous avions besoin de son soutien. Son excellent suivi et ses remarques très enrichissantes nous ont été d'un appui considérable.

Nos sincères remerciements s'adressent aussi à **M. Imad Eddine BOUMHI** Delivery Lead au sein de la Digital Factory OCP qui nous a proposé le sujet de stage, nous a accueillis et a facilité notre intégration.

Nous exprimons nos vifs remerciements et respects à notre encadrant au sein de la Digital Factory OCP, **M. Abdelhaq ELAIBI** pour ses conseils fructueux, sa vision critique, son savoir-faire et son savoir être durant toute la période du stage.

Nous tenons à remercier l'ensemble des collaborateurs pour leur accueil très chaleureux et plus particulièrement l'équipe de développement à savoir **Meryem, Ayoub, Khalifa et Omar** pour leur sympathie, leur encouragements et pour nous avoir donné toute information nécessaire.

Nous remercions tout particulièrement les membres du jury, pour l'honneur qu'ils nous ont fait en acceptant d'examiner et juger notre travail, ainsi que toute l'équipe pédagogique du département du génie informatique de l'EHTP, pour la formation prodigieuse et prestigieuse qu'elle nous a prodiguée.

## Résumé

*Le présent document constitue la synthèse de notre projet de fin d'études au sein de la digital factory OCP qui a pour objectif de contribuer à la réalisation du portail Requestor qui s'inscrit dans le cadre du projet Digital Procurement, qui a pour objectif la digitalisation du processus d'approvisionnement du groupe OCP SA et l'optimisation des relations avec ses fournisseurs.*

*Notre mission est de réaliser une application web qui permettra aux chefs de projet de l'OCP d'avoir plus de transparence sur le processus d'approvisionnement appelé « Source To Pay ». En effet, la solution permettra aux chefs de projet de suivre leurs demandes d'achat, leurs commandes, leurs factures et de gérer les attachments en intégrant un système de notifications. Elle offre au client des tableaux de bord pour le suivi et l'évaluation à travers un ensemble d'indicateurs.*

*Pour bien mener notre projet, nous avons adopté la méthode agile de gestion de projet SCRUM, accompagné de la méthode de développement 2TUP. Notre travail s'est étalé sur sept sprints, la durée de chacun étant de deux semaines.*

*L'application est divisée en quatre modules : demandes d'achat, commandes, attachments et factures.*

*En ce qui concerne l'aspect technique, nous nous sommes basés sur une architecture n-tiers orientée Microservices. Sur une plateforme JEE, nous avons utilisé Spring Boot comme framework de développement du Backend et du Middleware avec la librairie ReactJS dans le Frontend.*

### **Mots clés :**

*Digitalisation, Digital procurement, approvisionnement, Source To Pay, Portail Requestor, groupe OCP SA.*

## Abstract

*This document summarizes the work we carried out within the OCP digital factory, which aims to contribute to the creation of the Requestor portal which is a part of the Digital Procurement project, that aims to the digitization OCP SA's procurement process and the optimization of relations with its suppliers.*

*Our mission is to create a web application that will allow OCP project managers to have more transparency and visibility on the procurement process called « Source To Pay ». Indeed, the solution will allow project managers to track their purchase requests, their orders, their invoices and to manage the attachments by integrating a system of notifications. It offers to the client dashboards for monitoring and evaluation through a set of indicators.*

*To successfully carry out our project, we adopted the agile SCRUM project management method, accompanied by the 2TUP development method. Our work was spread over seven sprints, the duration of each being two weeks.*

*The project is divided into four modules. The first concerns purchase requests, the second one purchase orders, the third one attachments and the last one invoices.*

*For the technical aspect, we based ourselves on an n-tier architecture oriented Microservices. On a JEE platform, we used Spring Boot as a Backend and Middleware development framework with the library ReactJS in the Frontend.*

### **Keywords:**

*Digitalization, Digital procurement, Source To Pay, Requestor Portal, OCP SA group.*

## Table des matières

<i>Remerciements</i> .....	3
<i>Résumé</i> .....	4
<i>Abstract</i> .....	5
<i>Table des acronymes</i> .....	9
<i>Liste des figures</i> .....	10
<i>Liste des tables</i> .....	12
<i>Introduction</i> .....	13
<b>Chapitre I : Contexte du projet.....</b>	<b>15</b>
I.    Organisme d'accueil.....	16
1.    Groupe OCP S.A.....	16
II.    Présentation du projet.....	20
1.    Thème du projet .....	20
2.    Objectifs du projet.....	21
III.    Conduite du projet.....	21
1.    Méthode de gestion de projet.....	21
2.    Méthode de développement.....	22
3.    Pratique de développement .....	23
4.    Ressources humaines .....	24
5.    Planification du projet .....	24
Conclusion.....	26
<b>Chapitre II : Etude préliminaire .....</b>	<b>27</b>
I.    Etude de l'existant .....	28
1.    Présentation du processus métier.....	28
2.    Description de l'existant.....	29
3.    Limites de l'existant.....	29
4.    Solution proposée .....	30
II.    Recueil des besoins .....	31
1.    Besoins fonctionnels .....	31
2.    Exigences techniques.....	33
3.    Environnement technologique .....	33
4.    Architecture logicielle .....	34

5. Contraintes techniques.....	34
III. Modélisation du contexte du système .....	34
1. Identification des acteurs du système .....	34
2. Diagramme de contexte .....	35
Conclusion.....	36
Chapitre III : Spécifications générales .....	37
I. Spécifications fonctionnelles.....	38
1. Identification des packages .....	38
2. Identification des cas d'utilisations .....	38
II. Spécifications techniques .....	46
1. Architecture.....	46
2. Design Patterns .....	47
3. Framework.....	48
Conclusion.....	51
Chapitre IV : Analyse et conception.....	52
I. Page des demandes d'achats.....	53
1. Diagramme des classes candidates .....	53
2. Diagrammes de séquences .....	54
II. Page des commandes.....	55
1. Diagramme des classes candidates .....	55
2. Diagrammes de séquences.....	56
III. Page de paiement .....	58
1. Diagramme des classes candidates .....	58
2. Diagrammes de séquences .....	59
IV. Page des attachements.....	60
1. Diagramme des classes candidates .....	60
2. Diagrammes de séquences.....	61
V. Conception générale de l'application:.....	64
Conclusion.....	65
Chapitre V : Réalisation .....	66
I. Outils et technologies utilisés.....	67
1. Environnement de développement .....	67
2. Front End .....	68
3. Back End & Middleware .....	70
4. Outils de versioning .....	71

5. CI/CD .....	72
6. Les tests .....	73
II. Réalisation et mise en œuvre .....	74
1. Structure du projet .....	74
2. Charte graphique.....	78
3. Captures d'écran.....	79
Conclusion.....	92
Webographie .....	94
Annexes .....	95
Annexe 1 : SCRUM .....	96
1. Les principes d'une méthode agile.....	96
2. Au quotidien .....	96
Annexe 2 : Présentation du processus « S2P ».....	101
1. Sourcing: phase AMI/RFI.....	101
3. Sourcing: phase AO .....	102
4. Achat par commande standard .....	102
5. Achat par commande ouverte.....	103
6. Réception .....	104
7. Facturation et paiement .....	104
Annexe 3 : DevOps .....	105
1. Définition du modèle DevOps.....	105
2. Fonctionnement du DevOps .....	105
3. Les avantages du DevOps.....	106
Annexe 4 : Architectures microservices .....	107
1. Définition.....	107
2. Les aspects clés.....	107

## Table des acronymes

<b>2TUP</b>	2 Tracks Unified Process
<b>AC</b>	Appel de commande
<b>AMI</b>	Appel à Manifestation d'Intérêt
<b>AO</b>	Appel d'offre
<b>API</b>	Application Program Interface
<b>CD</b>	Continuos Deployment
<b>CI</b>	Continuos Integration
<b>DA</b>	Demande d'achat
<b>DI</b>	Demande Interne
<b>DNS</b>	Domain Name System
<b>DTO</b>	Data Transfer Object
<b>EB</b>	Expression de Besoin
<b>EBS</b>	E-Business Suit
<b>ERP</b>	Enterprise Resource Planning
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	HyperText Transfer Protocol
<b>IDE</b>	Integrated Development Environment
<b>IOC</b>	Inversion Of Control
<b>JEE</b>	Java Enterprise Edition
<b>JPA</b>	Java Persistence API
<b>JSON</b>	JavaScript Object Notation
<b>LDAP</b>	Lightweight Directory Access Protocol
<b>MOA</b>	Maîtrise d'ouvrage
<b>MOE</b>	Maîtrise d'œuvre
<b>MVP</b>	Model View Presenter
<b>PMBOK</b>	Project Management Body of Knowledge
<b>PMI</b>	Project Management Institute
<b>PMO</b>	Project management office
<b>PO</b>	Purchase Order
<b>PR</b>	Purchase request
<b>RFI</b>	Request For Information
<b>RFQ</b>	Request For Quotation
<b>S2P</b>	Source To Pay
<b>SASS</b>	Syntactically Awesome StyleSheets
<b>SGBD</b>	Système de Gestion de Base de Données
<b>SI</b>	Système d'informations
<b>SQL</b>	Strcutured Query Language
<b>UI</b>	User Interface
<b>UML</b>	Unified Model Language
<b>UX</b>	User eXperience
<b>WS</b>	Web Service

## Liste des figures

Figure 1: logo de OCP SA .....	16
Figure 2: organigramme de OCP SA.....	17
Figure 3: organigramme du Digital Office .....	19
Figure 4: logo de Digital Factory OCP.....	19
Figure 5: une itération selon SCRUM.....	22
Figure 6: illustration du processus de 2TUP .....	23
Figure 7: Schéma DevOps .....	23
Figure 8: planning du projet .....	25
Figure 9: Présentation du processus métier .....	28
Figure 10: logos de Oracle EBS, E-achat et E-invoice.....	29
Figure 11: Architecture logicielle .....	34
Figure 12: Diagramme de contexte dynamique.....	35
Figure 13: Diagramme de packages .....	38
Figure 14: Diagramme des cas d'utilisation - Demandes d'achat .....	39
Figure 15: Diagramme des cas d'utilisation - Commandes.....	40
Figure 16: Diagramme des cas d'utilisation - Factures .....	42
Figure 17: Diagramme des cas d'utilisation - Attachements.....	44
Figure 18: architecture de l'application.....	46
Figure 19: logo de spring boot.....	48
Figure 20: logo de React .....	49
Figure 21: logo de Redux .....	49
Figure 22: logo de Redux-saga.....	50
Figure 23: Diagramme des classes candidates - Demandes d'achat .....	53
Figure 24: Diagramme de séquence - Consulter détail d'une demande d'achat.....	54
Figure 25: Diagramme de séquence - Consulter le Dashboard des demandes d'achats....	54
Figure 26: Diagramme des classes candidates – Commandes .....	55
Figure 27: Diagramme de séquence - Consulter le détail d'une commande.....	56
Figure 28: Diagramme de séquence - Consulter le Dashboard des commandes .....	57
Figure 29: Diagramme de séquence - Voir le profil du fournisseur .....	57
Figure 30: Diagramme de classes candidates - Paiement.....	58
Figure 31: Diagramme de séquences - Détails d'une facture .....	59
Figure 32: Diagramme de séquences - Consulter le Dashboard des factures .....	59
Figure 33: Diagramme de séquences - Filtrer les factures.....	60
Figure 34: Diagramme de classes candidates - Attachements .....	61
Figure 35: Diagramme de séquences - Suivre un attachement.....	62
Figure 36: Diagramme de séquences - Consulter le dashboard des attachments .....	62
Figure 37: Diagramme de séquences - Filtrer les attachments .....	63
Figure 38: Conception générale de l'application.....	64
Figure 39: logo de IntelliJ IDEA.....	67
Figure 40: logo de Visual Studio Code .....	67
Figure 41: logo de ES6 .....	68
Figure 42: logo de Flow.....	68

Figure 43: logo de Eslint .....	69
Figure 44: logo de Sass .....	69
Figure 45: logo de Java EE .....	70
Figure 46: logo de Tomcat .....	70
Figure 47: logo de Gradle.....	70
Figure 48: logo de Oracle.....	71
Figure 49: logo de Git .....	71
Figure 50: logo de Gitlab.....	72
Figure 51: logo de Docker .....	72
Figure 52: logo de Kubernetes.....	73
Figure 53: logo de Mockito.....	73
Figure 54: logo de Jest .....	73
Figure 55: Structure du projet Frontend.....	74
Figure 56: Structure du projet Middleware .....	75
Figure 57: Classes du projet middleware.....	75
Figure 58: Package configuration middleware core .....	76
Figure 59: Structure du projet Backend .....	76
Figure 60: Structure de api oracle-ebs.....	77
Figure 61: structure de api bravo .....	77
Figure 62: Structure des pages de l'application .....	78
Figure 63: page responsive .....	78
Figure 64: page d'authentification .....	79
Figure 65: page des demandes d'achat .....	80
Figure 66: page détail des demandes d'achat .....	81
Figure 67: page suivi de la demande d'achat .....	82
Figure 68: page des commandes .....	83
Figure 69: Détail de la commande .....	84
Figure 70: page suivi de la commande .....	85
Figure 71: page des attachments.....	86
Figure 72: page de détail de l'attachment dans le cas normal .....	87
Figure 73: page de détail de l'attachment dans le cas de dépassement.....	87
Figure 74: page d'évaluation du fournisseur.....	88
Figure 75: page de rejet d'un attachment.....	88
Figure 76: page de paiement .....	89
Figure 77: page détail de la facture .....	89
Figure 78: Moteur de recherche .....	90
Figure 79: Notifications .....	91
Figure 80 : Cycle de vie logiciel .....	97
Figure 81: Backlog Trello .....	98
Figure 82: Ticket Trello.....	99
Figure 83: Processus AMI/RFI.....	101
Figure 84: processus AO.....	102
Figure 85: processus d'achat par commande standard.....	103
Figure 86: processus commande ouverte .....	103
Figure 87: processus de réception.....	104
Figure 88: processus de facturation et paiement .....	104

## Liste des tables

Tableau 1: Fiche signalétique du groupe OCP SA.....	17
Tableau 2: Organisation de l'équipe .....	24
Tableau 3: Fiche descriptive du package "Demandes d'achat" .....	40
Tableau 4: Fiche descriptive du package "Commandes" .....	41
Tableau 5: Fiche descriptive du package "Paiement" .....	43
Tableau 6: Fiche descriptive du package "Attachments" .....	45
Tableau 7: Description des classes candidates du package "Demandes d'achat" .....	53
Tableau 8: Description des classes candidates du package "Commandes".....	55
Tableau 9: Description des classes candidates du package "Paiement" .....	58
Tableau 10: Description des classes candidates du package "Attachments" .....	60

## Introduction

*La transformation digitale, parfois appelée transformation numérique, désigne le processus qui consiste pour une organisation, à intégrer pleinement les technologies digitales dans l'ensemble de ses activités. Le digital entraîne des évolutions comportementales profondes pour les collaborateurs dans le sens où les nouvelles technologies donnent lieu à de nouveaux usages, des façons innovantes de commercer et de communiquer, si bien que, la transformation numérique des entreprises s'avère être plus un sujet culturel que technologique. Et à ce titre, elle impose aux organisations une véritable problématique de conduite du changement de leurs collaborateurs. Avec l'arrivée massive du digital, les entreprises sont en effet amenées à repenser leurs processus, leurs modes de fonctionnement, leurs pratiques managériales et finalement leur culture d'entreprise.*

*C'est ainsi que le digital peut vraisemblablement révolutionner l'approche systémique de l'entreprise. Il donne lieu à de nouveaux usages et des façons innovantes de gestion des systèmes d'information à différentes échelles tout en gardant une importante position par rapport à ses concurrents, en offrant à ses collaborateurs les conditions adéquates de travail et en satisfaisant au maximum les exigences de ses clients.*

*N'étant pas une exception de ces entreprises, le groupe OCP a élaboré des politiques pour lancer le processus de la transformation digitale en restructurant sa direction de systèmes d'information par l'inauguration d'une nouvelle entité « la Digital Factory ».*

*Le but de cette nouvelle entité est d'accélérer la transformation digitale et d'adopter une stratégie digitale qui utilise les nouvelles technologies de l'information et de la communication pour accroître la valeur de l'entreprise, en lui permettant d'atteindre de nouvelles clientèles en ajoutant les canaux numériques à ses canaux de distribution traditionnels.*

*La stratégie digitale du groupe OCP et plus particulièrement de l'entité « Digital Office » se déploie dans les process pour s'adapter aux usages constitutifs de la mécanique organisationnelle de l'OCP. La Digital Factory comme étant une entité dynamique et agile du Digital Office, s'appuie sur l'optimisation des processus pour gagner en efficacité et en compétitivité.*

*Notre projet de fin d'étude s'inscrit dans la continuité de ces actions structurelles et pérennes entreprises au sein du groupe OCP SA et plus précisément pour l'optimisation de ses relations avec ses fournisseurs. Il s'agit d'une plateforme web qui permet la digitalisation du processus « Source to Pay » ou « S2P » englobant initiation et validation du besoin, appel d'offres, commandes, livraisons, réceptions, facturations, paiements, ...*

*Notre objectif dans ce projet est donc de participer au développement de cette application en collaborant avec les différentes parties prenantes du projet, à savoir l'équipe de développement, le Product Owner, l'équipe DevOps et l'équipe UI/UX en adoptant les règles du manifeste agile.*

*Le présent rapport décrit l'essentiel du travail réalisé lors de ce projet. Il comporte cinq chapitres.*

*Le premier définit le contexte général du projet à savoir la présentation de l'organisme d'accueil ainsi que la présentation du projet et la méthode de conduite du projet.*

*Le deuxième chapitre est une étude approfondie de l'existant, de ces limitations et par la suite un recueil des besoins fonctionnels et techniques.*

*Le troisième chapitre présente les spécifications fonctionnelles et techniques au cours desquelles on procèdera à la traduction des besoins fonctionnels en diagrammes de packages, en cas d'utilisation et en scénarii. On tachera également d'y définir notre architecture ainsi que les frameworks et les design patterns utilisés.*

*Le quatrième chapitre est consacré à l'analyse et la conception du système en tenant compte de la répartition du projet.*

*Le cinquième chapitre décrit la phase réalisation et mise en œuvre de notre application où nous transformerons les différents diagrammes conçus au cours des phases précédentes en écrans fonctionnels.*

*A la fin de ce rapport, nous présentons en complément les différentes annexes pour mieux appréhender son contenu.*

# Chapitre I : Contexte du projet

---

*Dans cette première partie nous allons présenter l'organisme d'accueil. Ensuite, nous allons décrire notre projet avant de mettre l'accent sur les grands choix techniques adoptés. Enfin, nous allons expliciter le planning que nous avons élaboré pour mener ce travail dans un cadre plus organisé qui témoigne d'une bonne maîtrise du temps.*

## I. Organisme d'accueil

### 1. Groupe OCP S.A

#### a. Présentation



Figure 1: logo de OCP SA

Première entreprise industrielle du Maroc, l'Office Chérifien des Phosphates (OCP) est un des leviers clé de l'économie marocaine depuis sa création en 1920. Ayant comme activités principales l'extraction et la transformation de phosphate, son rôle s'étend bien au-delà pour contribuer au développement économique et social du Royaume.

L'OCP s'impose comme un des leaders mondiaux de l'industrie de phosphates et de la production d'engrais phosphatés. Sa position de leader lui permet d'être au cœur d'enjeux -planétaires et nationaux- de première importance qui déterminent et orientent en grande partie son développement.

Le Groupe a connu une expansion considérable qui lui a permis de consolider son leadership mondial, il gère aujourd'hui un portefeuille de 160 clients et se distingue par une forte présence à travers le Maroc et à l'étranger.

OCP prévoit d'augmenter sa capacité de production de 30 à 50 millions de tonnes, ainsi que d'augmenter sa production d'engrais en aval à travers des partenariats stratégiques, spécialement à Jorf Phosphate Hub (JPH) où des infrastructures sont en train d'être développées pour accueillir 10 unités supplémentaires. Cette plateforme Plug and Play offrira des infrastructures communes à bas coût, et sera connectée par un slurry pipeline au plus grand gisement de phosphates au monde situé à Khouribga, ce qui assurera un approvisionnement sécurisé.

Présent dans cinq zones géographiques du pays (trois sites d'exploitation minières : Khouribga /Yousoufia, Boucraâ/Laayoun et deux sites de transformation chimique : Safi et Jorf Lasfar), OCP constitue un vecteur de développement régional et national important.

Le tableau ci-dessous représente la fiche signalétique du groupe OCP S.A:

<b>Dénomination sociale</b>	<b>Groupe Office Chérifien des Phosphates (Groupe OCP S.A dès 2008)</b>
<b>Siège social</b>	<b>2, rue Al Abtal, Hay Erraha Casablanca.</b>
<b>Forme juridique</b>	<b>Société anonyme.</b>
<b>Date de création</b>	<b>7 Août 1920.</b>
<b>Produits</b>	<b>Phosphate et Engrais</b>
<b>Chiffre d'affaire</b>	<b>42,7 milliards de MAD (en 2016)</b>
<b>Effectif</b>	<b>Environ 21000 collaborateurs</b>
<b>Directeur général</b>	<b>M. Mostafa Terrab</b>
<b>Tel</b>	<b>+212 5222-30025</b>
<b>Site Web</b>	<b><a href="http://www.ocpgroup.ma">http://www.ocpgroup.ma</a></b>

Tableau 1: Fiche signalétique du groupe OCP SA

## b. Organigramme



Figure 2: organigramme de OCP SA

## 2. Digital Office

### a. Présentation

*Au vu des opportunités stratégiques qu'offre le digital au groupe, L'OCP SA a décidé de créer une nouvelle entité « Digital Office » dirigée par un « Chief Digital Officer ». Cette entité est rattachée à la Direction Générale, et est en charge de conduire la transformation digitale et de renforcer la culture digitale au sein du groupe. Afin de créer les synergies nécessaires pour mener à bien sa mission, l'entité « Systèmes d'Information » lui est rattachée directement.*

### b. Missions du Digital Office

*L'entité « Digital Office » est en charge de :*

- *Elaborer une stratégie digitale du groupe et la décliner en feuille de route, en cocréation avec les différentes entités du groupe.*
- *Conduire la mise en œuvre des initiatives digitales de la feuille de route, en étroite collaboration avec les différentes entités du groupe.*
- *Elaborer et mettre en œuvre une stratégie et un modèle de gouvernance de la « Data » au sein du groupe, en permettant de garantir la disponibilité, la fiabilité et la cohérence des données.*
- *Mettre en place, en collaboration avec les différentes entités du groupe, les infrastructures et moyens nécessaires à la réussite de la transformation digitale du groupe (digital factory et modes de fonctionnement agiles, Data Lake, satffing, ...).*
- *Promouvoir une culture digitale au sein du groupe, et insuffler de nouveaux modes de travail et de collaboration via le Digital.*
- *Mettre en place des plateformes d'incubation permettant l'émergence d'initiatives innovantes de de nouveaux modes de fonctionnement.*

### c. Structure du Digital Office

*En application de ces principes, l'entité « Digital Office » est structurée comme suit :*

- *Une « **Digital Factory** » avec des antennes sur les différents sites du groupe, en charge de livrer les initiatives de la feuille de route, dans des cycles d'itération courts, à travers des méthodes de fonctionnement agile autour d'équipes cross-fonctionnelles.*
- *Une entité « **Systèmes d'Informations** » en charge d'assurer la réalisation des projets SI et télécoms du groupe, ainsi que l'assistance et le support aux utilisateurs du groupe, dans le respect des exigences de qualité et des meilleurs standards.*

- Une entité « **Data Management** » en charge d'élaborer une stratégie et un modèle de gouvernance de la Data, de définir l'architecture Data au sein du groupe et de mettre en œuvre les initiatives data y afférentes.
- Une entité « **Digital Planning & PMO** » en charge de la construction et de la mise à jour de la roadmap digitale du groupe et du suivi de son exécution.
- Une entité « **Business Architecture** » en charge de conseiller et challenger les métiers sur les solutions digitales pertinentes pour adresser leurs besoins, les consolider et suivre leur mise en œuvre.
- Un pool « **Digital Innovation & Change Officers** » en charge d'identifier et mener des projets d'innovation en matière de digital ainsi que la promotion d'une culture digitale et de nouveaux modes de travail au sein du groupe.

L'organigramme de l'entité est comme suit, nous effectuons notre stage fin d'études à l'entité « **Digital Factory** » :

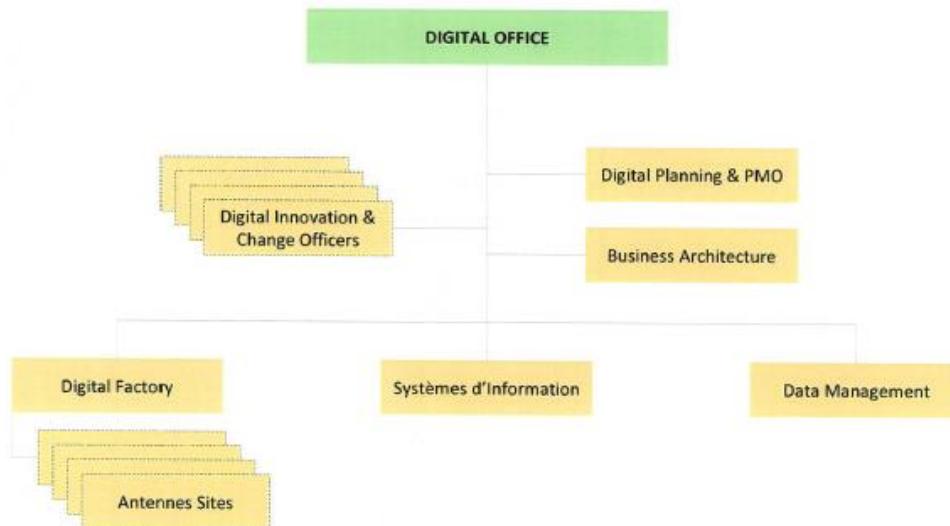


Figure 3: organigramme du Digital Office

#### d. Digital Factory



Figure 4: logo de Digital Factory OCP

*Une digital factory est un lieu collaboratif et inventif, qui a fait son apparition il y a quelques années dans des entreprises de différents secteurs en Europe et aux Etats-Unis.*

*Il s'agit d'un laboratoire dédié à l'innovation qui regroupe des équipes aux profils pluridisciplinaires engagées autour de la transformation globale de la compagnie. Dans une Digital Factory, on ne trouve pas de Manager, Il y'a des gens qui se regroupent autour d'un produit, ils sont autonomes, créatifs et entrepreneurs.*

*Ces collaborateurs exercent plusieurs métiers digitaux avec un seul objectif: développer rapidement ce que veut le client.*

*La force donc d'une Digital Factory, n'est pas seulement ses technologies, mais c'est surtout sa façon de travailler. Pour sa digital factory, le groupe OCP a misé sur deux approches d'organisation du travail: L'agilité et le Scrum.*

*Ces collaborateurs exercent plusieurs métiers qui incluent des Scrum Masters (coaches agiles), des Product Owners (chefs de produit), des Full-Stack développeurs (développeurs de produits de bout en bout capables de livrer rapidement), des Data Scientists (analystes de données), une équipe DevOps (veille sur l'intégration continue des projets) et des Delivery Lead.*

*Aussi travaillent-ils en petites équipes (Squads) qui ne doit pas dépasser 8 personnes, c'est l'optimal pour garder la créativité et ne pas perdre en communication et en proximité.*

*Depuis son lancement, la Digital factory du groupe OCP a montré son engagement en terme de satisfaction de ses clients et a donc lancé comme premiers projets deux plateformes qu'elle a jugé utiles pour le développement du groupe. La plateforme « REQUESTOR » qui fait le sujet de notre projet de fin d'études, et la plateforme « END TO END SUPPLIER » qui vise la digitalisation de la gestion des relations des fournisseurs avec le groupe.*

## II. Présentation du projet

### 1. Thème du projet

*Etant un leader mondial dans l'extraction et la transformation des phosphates, le groupe OCP vise à consolider des relations fiables, pérennes et mutuellement gagnantes avec l'ensemble de ses fournisseurs et partenaires. Chaque jour, le groupe traite des milliers de factures, de demandes d'achat, de données fournisseurs, de contrats, de formulaires de dépenses, etc. Une gestion manuelle du processus nécessite habituellement beaucoup plus de temps, notamment à cause du manque de contrôle. Afin de répondre à ce challenge, la direction des achats se dote de solutions conçues afin de gérer les processus achats stratégiques, opérationnels et transactionnels dans un système unique.*

*Etant dans l'ère du digital, OCP S.A a décidé de digitaliser le processus « Source To Pay » qui gère les différentes étapes du processus achat, en centralisant les informations collectées à travers une suite d'applications intégrées utilisées au sein du groupe à fin de gérer les processus d'approvisionnement dans une seule application.*

## 2. Objectifs du projet

Notre projet de stage consiste à contribuer à la réalisation de la plateforme « Requestor » du grand projet « Digital procurement ». Il entre dans le cadre de la stratégie de digitalisation des processus du groupe OCP et de ses flux d'informations. Il s'agit d'une plateforme web visant l'optimisation des relations d'OCP SA avec ses fournisseurs. Elle est destinée principalement aux chefs de projet qui travaillent au sein du groupe afin de leur permettre d'avoir, en temps réel, plus de transparence sur l'ensemble des étapes du processus « S2P » (Source To Pay) qui comporte l'ensemble des opérations d'initiation et validation du besoin, appel d'offres, commandes, livraisons, réceptions, facturations, paiements, ...

Une telle digitalisation, permettra également de mieux collaborer en interne avec les différentes parties prenantes du processus S2P et en externe avec les fournisseurs et les prestataires de services.

Notre contribution dans ce projet concerne particulièrement la réalisation des modules demandes d'achat, commandes, attachements et factures en collaborant avec les différentes parties prenantes du projet, à savoir l'équipe de développement, le Product Owner, l'équipe DevOps et l'équipe UI/UX en adoptant les règles du manifeste agile.

## III. Conduite du projet

Notre équipe a adopté une méthode de conduite de projet capable de s'adapter tant au niveau contenu qu'au niveau périmètre du projet, avec les changements qui peuvent avoir lieu durant le processus de développement. En vue de livrer un produit de haute qualité dans les délais convenus avec le client, nous avons travaillé avec la méthode agile de gestion de projet SCRUM, en plus de la méthode de développement 2TUP et avec l'approche DevOps.

### 1. Méthode de gestion de projet

Le projet est un projet critique d'une part par sa grande taille et d'autre part par sa grande rentabilité. Ce qui impose d'être très proche du client et de savoir gérer les différents besoins à temps.

La méthode de gestion de projet adoptée dans ce travail est SCRUM. C'est une méthode agile dédiée à la gestion de projets et issu des méthodes incrémentales qui permettent de maîtriser une production planifiée.

Scrum se différencie des autres méthodes par ses avantages, qui ont fait de ce procédé une réponse pragmatique aux différentes contraintes du projet :

- **Méthode itérative et incrémentielle :** ce qui permet d'éviter de ne voir le résultat qu'à la livraison finale, et de présenter ainsi à chaque fin d'itération une version qui répond à un besoin spécifique.

- **Adaptabilité maximale pour du développement de produits et d'applications :** la composition séquentielle du contenu des sprints permet d'ajouter des modifications ou des fonctionnalités qui n'était pas prévue au départ.
- **Méthode participative :** chaque membre de l'équipe est impliqué et invité à s'exprimer et participer à toutes les décisions prises sur le projet.
- **Augmentation de la communication :** en travaillant dans la même salle de développement, ou en étant connecté avec différents moyens de communication, l'équipe peut communiquer facilement et échanger sur les obstacles afin de les supprimer au plus tôt.
- **Maximisation de la coopération :** les échanges quotidiens entre le client et l'équipe permettent un rapprochement et une entraide se met en place.

La figure ci-dessous, illustre une itération selon la méthode SCRUM :

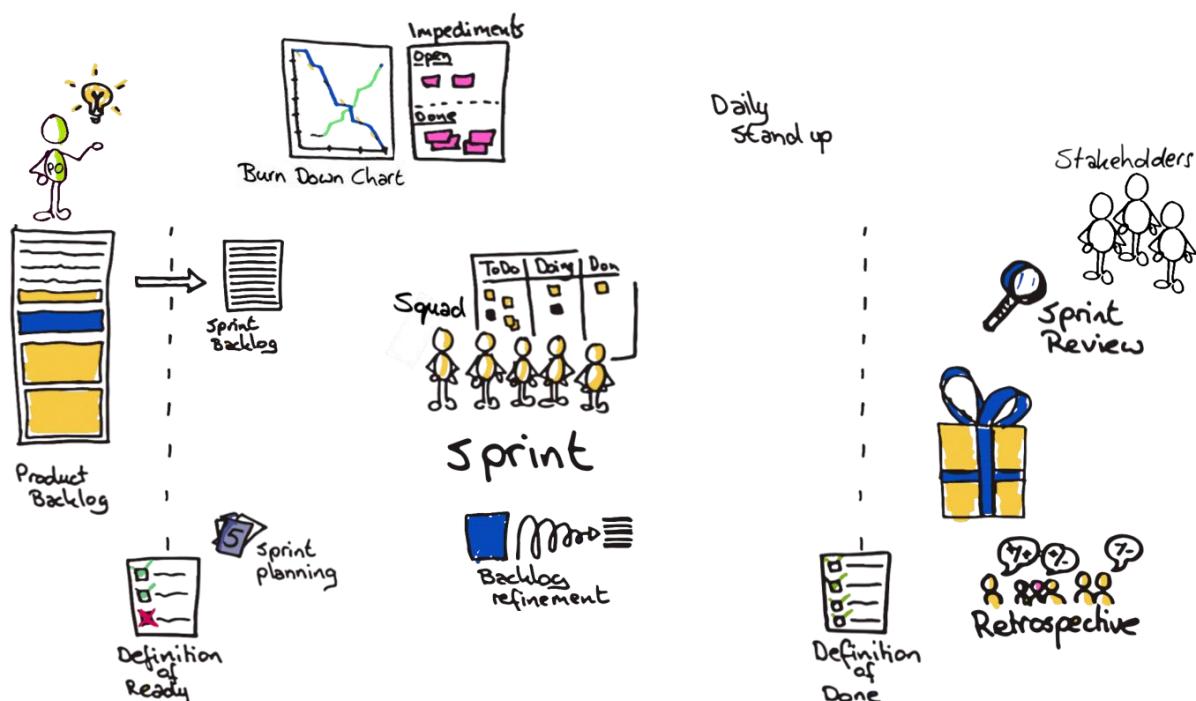


Figure 5: une itération selon SCRUM

## 2. Méthode de développement

La complexité croissante des systèmes informatiques a conduit les concepteurs à s'intéresser aux méthodes de développements.

Pour produire un logiciel de qualité, qui répond aux besoins des utilisateurs dans les temps et les coûts prévisibles, on a opté pour la méthode 2TUP puisque la mise en œuvre du projet nécessite une étude technique approfondie et une vision globale de la solution à mettre en place que ce soit au niveau technique, architectural ou fonctionnel.

Ci-dessous, une illustration du processus 2TUP adopté :

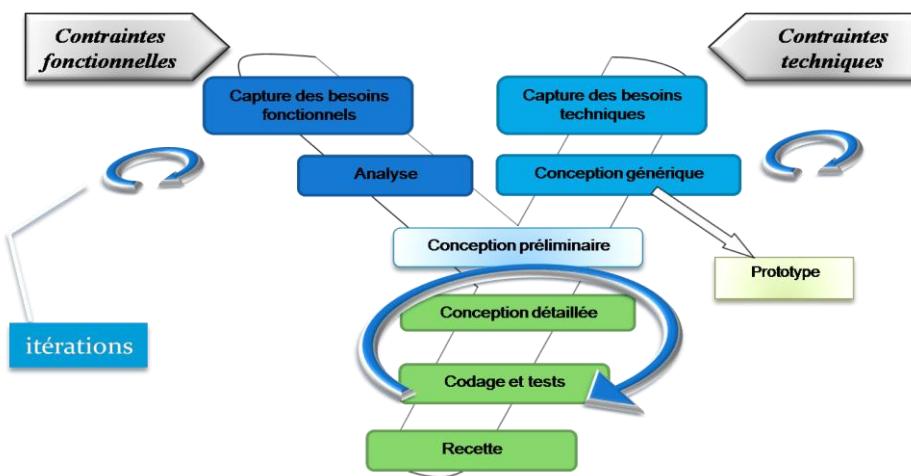


Figure 6: illustration du processus de 2TUP

### 3. Pratique de développement

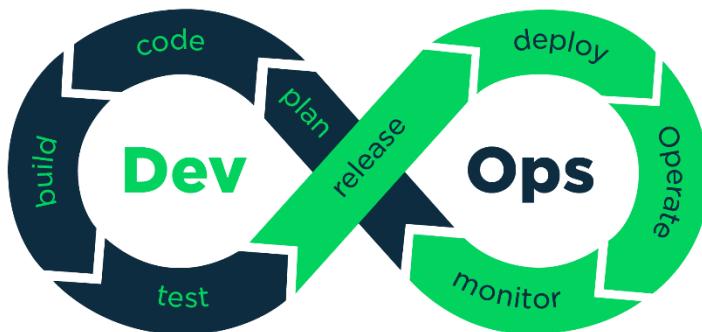


Figure 7: Schéma DevOps

DevOps est un ensemble de pratiques qui automatisent les processus entre les équipes de développement et IT afin de leur permettre de développer, tester et livrer des logiciels plus rapidement et avec plus de fiabilité. Le concept de DevOps repose sur la mise en place d'une culture de la collaboration entre les équipes qui étaient, historiquement, cloisonnées. Parmi les avantages assurés, le gain de confiance, l'accélération des livraisons, la capacité à résoudre les tickets plus rapidement ou encore la gestion plus efficace des tâches non planifiées.

À la base, DevOps est une culture, un courant de pensée, une philosophie. C'est une poignée de main ferme entre les équipes de développement et opérationnelles. L'accent est mis sur le changement de mentalité, la collaboration accrue et l'intégration plus poussée. DevOps associe la méthodologie Agile, l'automatisation, la livraison continue et bien plus.

encore pour aider les équipes de développement et opérationnelles à gagner en efficacité, à innover plus rapidement et à offrir plus de valeur ajoutée aux business et aux clients.

## 4. Ressources humaines

*Le suivi de l'équipe et l'organisation du projet sont illustrés sur le tableau suivant :*

	Nom & prénom	Rôle
Comité de suivi	Pr. Malika ADDOU	Encadrant au sein de l'EHTP
	M. Abdelhaq EL AIBI	Technical Lead, encadrant au sein de la Digital Factory OCP.
	M. Karim KHALFI	Delivery Lead
Maîtrise d'ouvrage	M. Abdelaziz HARRATI	Product Owner
Maîtrise d'œuvre	Mlle Maryam ZIAD	Développeurs full-stack
	M. Ayoub DEQQAQ	
	M. Khalifa RHABI	
	M. Ayoub QABBAL	Coach Agile
	M. Hosni OUELDBELGARNE	DevOps
	Mlle Zainab SOULAYMANI	UX Designer
	M. El Mehdi BOUMHICHA	Elèves Ingénieurs à l'EHTP
	M.Mohamed LMOUSSAOUI	

Tableau 2: Organisation de l'équipe

## 5. Planification du projet

*La planification d'un projet est un outil incontournable dans sa conduite. On a choisi de soumettre la planification de notre projet à la méthodologie SCRUM. En effet, la méthode SCRUM soumet tout d'abord un planning de livraison, résultat d'une réunion de prévision de livraison, qui énonce l'ensemble du projet. Le planning de livraison est ensuite utilisé pour créer les plannings d'itérations pour chacune d'entre elles. Ainsi, une réunion de prévision d'itération se déroule au début de chaque itération pour produire le planning des tâches de développement. Les user-stories, à savoir la liste des scénarios utilisateur, sont alors choisies pour cette itération depuis le planning de livraison par ordre de priorité convenu avec le client. Les tests de validation en échec qui doivent être réparés sont aussi sélectionnés.*

*On présente dans la figure de la page suivante le planning selon lequel s'est déroulé notre projet :*

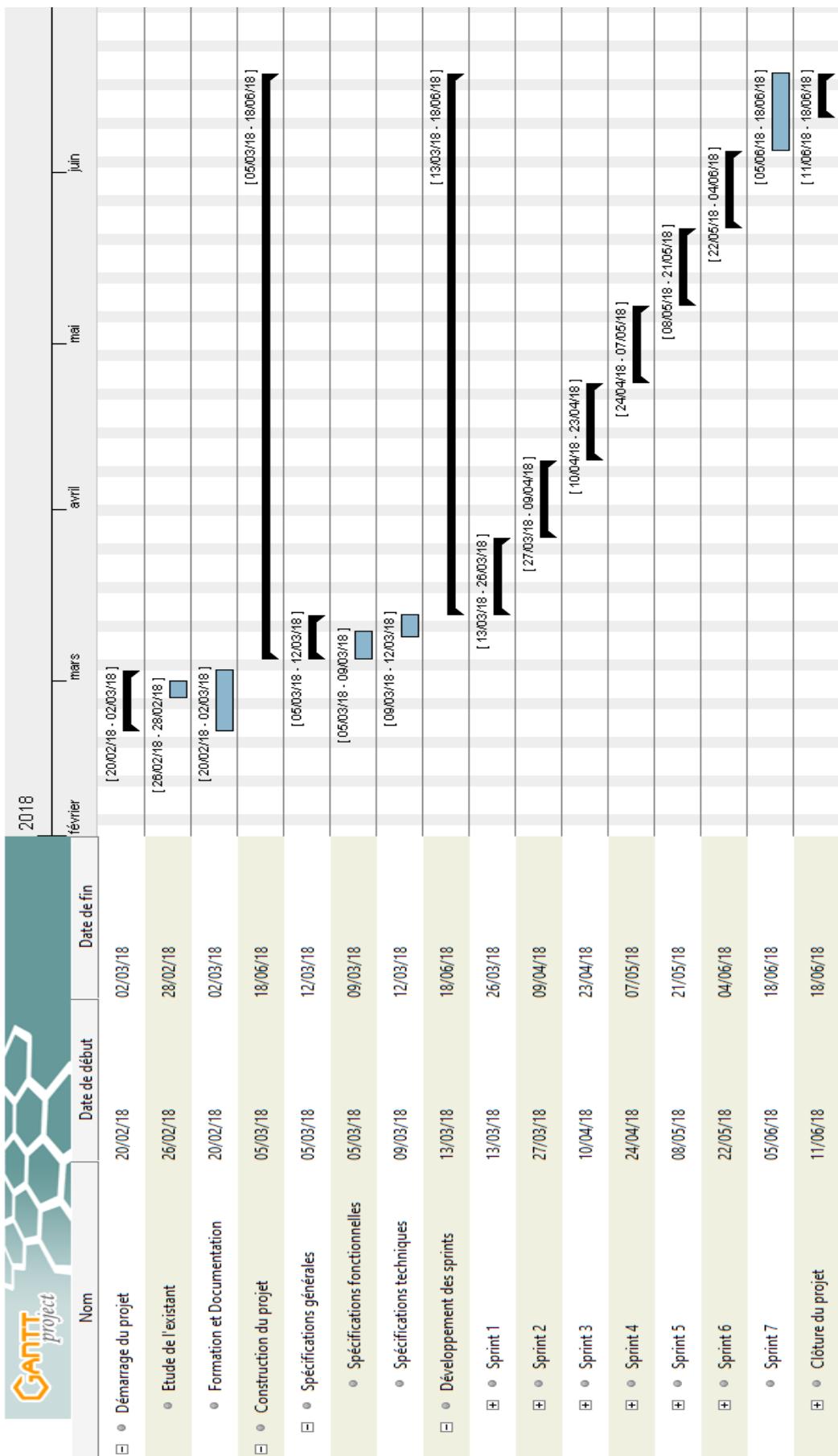


Figure 8: planning du projet

## Conclusion

*Le premier chapitre présentait un point de départ pour l'élaboration du projet dans la mesure où il définit son contexte général à savoir l'organisme d'accueil, les objectifs du projet et la démarche suivie pour sa réalisation.*

*Dans la partie suivante, nous présentons la phase d'étude préliminaire en exposant l'existant afin de mieux cerner les spécifications fonctionnelles et techniques.*

## Chapitre II : Etude préliminaire

---

*L'étude préliminaire est la première étape de notre processus de développement. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels. Dans cette phase nous allons mettre l'accent sur l'étude de l'existant pour recenser par la suite les besoins fonctionnels et ceux techniques.*

## I. Etude de l'existant

### 1. Présentation du processus métier

*Il s'agit du processus Source-to-Pay qui va de l'approvisionnement jusqu'au paiement. Ce processus commence par une expression de besoin du demandeur, cette dernière doit être validé par le service d'approvisionnement pour devenir une demande d'achat.*

*Pour répondre au besoin d'une demande d'achat, un document "Request For Information" (RFI) est élaboré pour donner des informations précieuses sur le futur appel d'offre et permettre une pré-qualification : valeur du contrat, durée du contrat, nature des prestations, nombre de candidats appelés, ...*

*Le fournisseur communiquera selon les demandes exprimées dans le RFI ses données financières, ses rapports de gestion et ses données opérationnelles.*

*Le processus de demande d'achat peut ainsi déclencher plusieurs sous processus tel que la création d'un nouveau budget, la création d'un nouveau contrat ou le référencement d'un nouveau fournisseur, l'action qui suit donc est l'approbation de la demande d'achat qui donnera lieu à la génération d'une ou plusieurs commandes.*

*La commande subit une révision puis un contrôle et vérification de budget et habilitations pour la confirmer, pour que le déroulement de la commande puisse se faire selon les objectifs sur le plan technique et commercial, un système de planification est utilisé pour anticiper la disponibilité de la marchandise ainsi que les délais de livraison pour le matériel nécessaire. Si l'on constate des retards, il est possible de prendre des mesures sans délai.*

*Il existe deux types de produits finis achetés : prestations et équipements. Les prestations sont directement mises à disposition et les achats matériels sont stockés en vue de leur livraison pour subir un contrôle de qualité afin de décider s'il y a des actions correctives à appliquer.*

*Une fois la commande est réceptionnée, une ou plusieurs factures sont établies et subissent au traitement par le service comptable pour vérifier les documents contractuels et attachements et créer les échéanciers de règlement après validation et règlement.*

*Le schéma ci-dessous présente le processus « Source To Pay » :*



Figure 9: Présentation du processus métier

## 2. Description de l'existant

OCP SA opte à utiliser un ensemble de solutions logicielles pour simplifier le processus d'achat pour les fonctionnaires de l'entreprise particulièrement les chefs de projet, réduire les erreurs de facturation et accélérer l'acquisition des factures et des commandes.

Le système courant s'appuie sur trois solutions qui agissent de manière dépendante pour assurer la gestion du processus « Source To Pay » (S2P) qui englobe aussi bien les sous processus « Source To Procure », « Procure To Invoice » et « Invoice To Pay ». Ce processus comporte l'ensemble des opérations d'initiation et validation du besoin, appel d'offres, commande, livraisons, réceptions, facturations ...

Ces opérations sont gérées par les trois systèmes de la manière suivante :

- Oracle EBS gère la phase d'expression et validation du besoin ainsi que la gestion des commandes et le suivi des livraisons et des réceptions.
- E-Achat récupère les demandes d'achats depuis Oracle EBS pour s'occuper de la phase d'appel d'offre ensuite il s'occupe de la gestion des relations avec les fournisseurs.
- E-invoice pour la facturation et paiement, il s'agit d'une solution électronique utilisée par les fournisseurs, pour présenter et contrôler leurs factures et s'assurer du processus de facturation.



Figure 10: logos de Oracle EBS, E-achat et E-invoice

## 3. Limites de l'existant

L'accès aux différents services proposés par ces solutions n'est pas centralisé sur une seule plateforme, ces derniers offrent des services dans leurs propres plateformes et peuvent communiquer entre eux pour garantir le processus d'achat. Cette architecture présente plusieurs inconvénients à savoir :

- Le système courant est complexe car l'infrastructure courante comporte trois logiciels différents avec des points d'accès différents et c'est difficile de bien définir la totalité des fonctionnalités que proposent chaque logiciel et il y a un risque de confusion entre fonctionnalités qui existent dans des logiciels séparés.

- *Les trois solutions électroniques sont interdépendantes car il existe certaines phases où ils doivent partager des informations entre eux pour compléter le déroulement du processus, l'échange des données entre différents logiciels est difficile à maintenir car ils n'ont pas la même structure de donnée et on risquera de violer les règles de structure.*
- *Le système n'est pas fluide puisque l'utilisateur perd trop de temps en naviguant sur trois logiciels. En effet l'utilisateur fait un effort coûteux en termes de temps pour s'adapter à chaque système qui a sa propre représentation de données et ses propres cas d'utilisations.*
- *Le système courant ne couvre pas une parfaite transparence du processus d'achat du niveau le plus général jusqu'au niveau le plus spécifique, ne dispose pas d'un moyen de suivi d'avancement du processus ou d'indicateurs qui représenteront l'état du processus et ne donne pas la possibilité de collecter des feedbacks à propos des services offerts par les fournisseurs pour bien évaluer ces derniers.*
- *Les appareils mobiles permettent aux utilisateurs de rester connectés aux Web d'une façon plus aisée mais le système actuel ne fournit pas une application adaptée aux tablettes ou aux smartphones.*
- *Puisque le processus d'achat déclenche plusieurs événements selon le niveau d'achat, il y a donc un besoin primordial de recevoir l'évènement lors dès son échéance. Cependant le système ne gère pas l'envoi des notifications pour signaler aux utilisateurs les nouveautés de leurs commandes ou de leurs factures.*

#### 4. Solution proposée

*Notre projet de fin d'étude, qui entre dans le cadre de la transformation digitale du groupe OCP, contribue à la réalisation de la plateforme « Requestor » du projet « Digital Procurement ». Son objectif est de digitaliser les processus et les flux d'informations depuis la phase d'expression de besoin jusqu'à la phase de la facturation. Pour répondre à ce besoin l'application va permettre de :*

- *Couvrir le processus d'approvisionnement de bout en bout.*
- *Centraliser l'information sur une seule plateforme.*
- *Proposer une solution responsive adaptable aux appareils mobiles et tablettes.*
- *Envoi de notifications aux chefs de projet lors du déclenchement d'un événement impactant le processus.*
- *Proposer des tableaux de bord avec des indicateurs clés.*

## II. Recueil des besoins

*Pour des fins de simplification et de structuration, les besoins correspondront à un ensemble de pages.*

### 1. Besoins fonctionnels

#### **Page Login :**

- *L'utilisateur est invité à renseigner son identifiant et son mot de passe. Si le login et le mot de passe sont corrects, l'utilisateur est redirigé vers la page d'accueil. Sinon, un message d'erreur est affiché, invitant l'utilisateur à ressaisir les bons identifiants.*
- *Puisque l'OCP SA dispose d'un serveur LDAP, il serait plus judicieux de l'utiliser dans l'authentification.*

#### **Page Accueil :**

- *La page d'accueil comporte un Dashboard avec trois indicateurs : appels d'offres adjugés, demande d'achats approuvées et commandes clôturées.*
- *Elle dispose aussi d'un message d'accueil expliquant le contexte du projet.*

#### **Page demandes d'achats :**

*Les principaux composants de la page sont :*

- *Un tableau de bord qui contient des indicateurs.*
- *Une liste des demandes d'achats divisée par sections : demandes en préparation, demandes en attente d'approbation, demandes approuvées et demandes rejetées.*
- *En cliquant sur une demande on aperçoit le détail de la demande.*
- *Dans la partie détails de la commande on peut consulter l'avancement de la demande sous forme de timeline. Chaque étape est indiquée par une couleur :*
  - *Etape validée : vert*
  - *Etape en retard : orange*
  - *Etape non entamé : gris*

#### **Page commande :**

*Les principaux composants de la page sont :*

- *Un tableau de bord qui contient des indicateurs.*
- *Une liste des commandes divisée par sections : nouvelles commandes, commandes en cours, commandes rejetées. En cliquant sur une commande on aperçoit le détail de la commande.*
- *Dans la partie détails de la commande on peut consulter le profil du fournisseur ainsi que l'avancement sous forme de timeline avec des informations sur le fournisseur, la livraison et la facturation. Chaque étape est indiquée par une couleur :*
  - *Etape validée : vert*
  - *Etape en retard : orange*
  - *Etape non entamé : gris*

### **Page Paiement :**

Les principaux composants de la page sont :

- *Un tableau de bord qui contient des indicateurs.*
- *Une liste des factures.*
- *Des filtres de recherche par statuts et par date.*
- *En cliquant sur une facture, on voit l'avancement des étapes de la facture sur une timeline. Chaque étape est indiquée par une couleur :*
  - *Etape validée : vert*
  - *Etape en retard : orange*
  - *Etape non entamé : gris*

### **Page attachments :**

Les principaux composants de la page sont :

- *Un tableau de bord qui contient des indicateurs.*
- *Une liste des attachments divisée par statuts : nouveaux attachments et les attachments validés ou rejetés.*
- *En cliquant sur un attachment on aperçoit le détail de l'attachment ainsi que la possibilité de valider ou rejeter un attachment s'il s'agit d'un nouvel attachment.*
- *Dans le cas de confirmation, on évalue le fournisseur selon plusieurs critères qui diffèrent selon le type de la commande.*
- *Dans le cas de rejet d'un attachment on spécifie dans un check box le motif de rejet.*

### **Page recherche :**

*Les principaux composants de la page de recherche sont :*

- *Un formulaire de recherche où on saisit le mot à chercher ça peut être un nom de fournisseur, une référence de commande une description de la demande d'achat ...*
- *Le résultat de la recherche filtré par catégorie : demande d'achat, commande, facture ou attachement. En cliquant sur un résultat on accède à la page détail de l'objet concerné.*

## **2. Exigences techniques**

*Le produit doit répondre outre aux besoins fonctionnels, à un certain nombre d'exigences techniques :*

- *Le respect des bonnes pratiques au niveau du codage.*
- *Le développement de composants réutilisables qui permettent l'extensibilité de l'application dans le futur.*
- *Prévoir un design responsive pour chaque page de l'application afin de supporter les smartphones et les tablettes.*
- *Utiliser des requêtes SQL optimisées.*
- *Valider les règles imposées par SONARQUBE : c'est un outil permettant de mesurer la qualité du code source en continu.*
- *Utilisation de l'outil de versioning Git, permettant la conservation de diverses versions du projet ainsi que le partage du code entre membre de la MOE.*
- *Valider tous les tests (Backend, Middleware, Frontend) avant le déploiement.*

## **3. Environnement technologique**

*Pour la réalisation nous avons utilisé les outils suivants :*

- **IDE:** IntelliJ IDEA, Visual Studio Code.
- **Framework et librairies:** Spring Boot, React JS.
- **Languages:** Javascript/JAVA EE.
- **SGBD:** Oracle, MySQL.
- **Outil de versioning:** Git.

## 4. Architecture logicielle

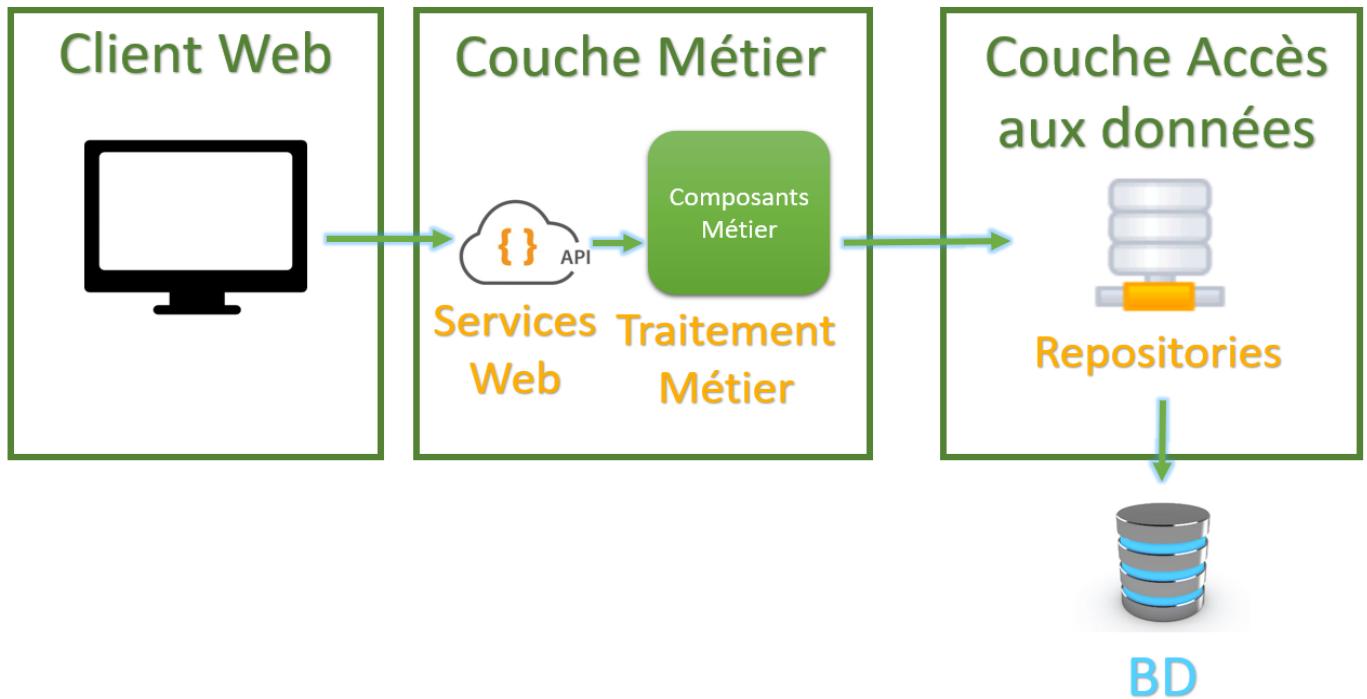


Figure 11: Architecture logicielle

## 5. Contraintes techniques

*Nous avons fait face à quelques difficultés concernant la base de données du système, à partir de laquelle notre solution puise les données. Ces contraintes sont dues à l'absence d'une documentation des champs qui sont utilisés sur la base de données, ainsi que la normalisation de ces données vues qu'elles sont extraites de trois sources différentes.*

### III. Modélisation du contexte du système

#### 1. Identification des acteurs du système

*Selon la notation UML un acteur représente l'abstraction d'un rôle joué par des entités externes qui interagissent directement avec le système étudié. Dans ce qui suit, nous présentons les acteurs qui interagissent avec notre application.*

*Dans le cas de notre projet, on a un seul acteur il s'agit du chef de projet.*

## 2. Diagramme de contexte

Le diagramme de contexte dynamique illustre le système, comme objet central, les acteurs et les liens entre le système et ce dernier. L'objectif de cette représentation est de montrer une vue globale du système développé.

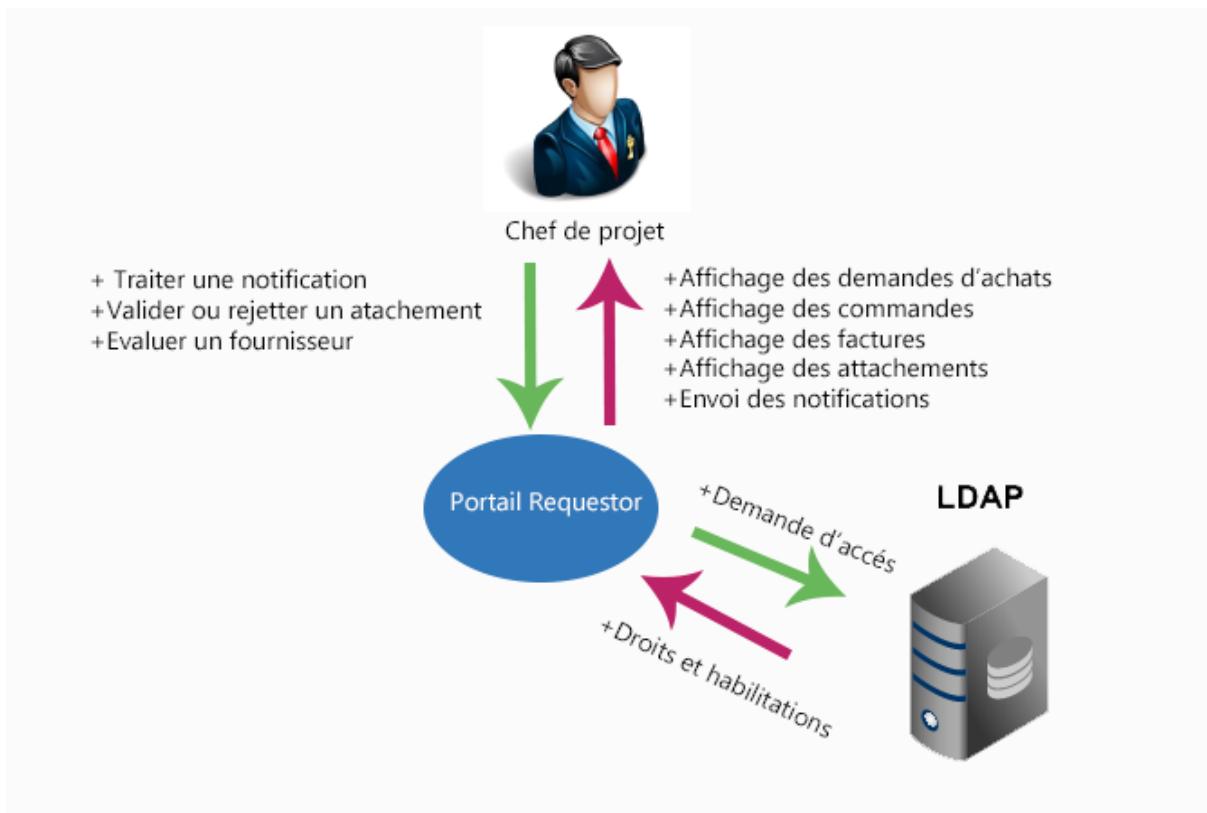


Figure 12: Diagramme de contexte dynamique

## Conclusion

*Pour résumer, la phase d'étude préliminaire du système consiste à présenter l'existant, ses limites, la solution proposée ainsi que les acteurs principaux du système.*

*Dans la partie suivante, nous entamerons les spécifications générales du projet sur le plan fonctionnel et sur le plan technique.*

## Chapitre III : Spécifications générales

---

Ce chapitre sera consacré aux spécifications générales de notre projet. Nous commencerons par les spécifications fonctionnelles dans lesquelles nous allons exposer les différentes fonctionnalités offertes par notre application. Ensuite, nous passerons aux spécifications techniques dans lesquelles nous allons définir l'architecture du projet ainsi que les design patterns et les frameworks utilisés.

## I. Spécifications fonctionnelles

### 1. Identification des packages

*Un package représente un ensemble de fonctionnalités fortement cohérentes, et sont généralement de même nature et de même niveau sémantique.*

*Pour mieux organiser notre projet, on a réparti notre projet sur quatre packages principaux plus ou moins indépendants.*

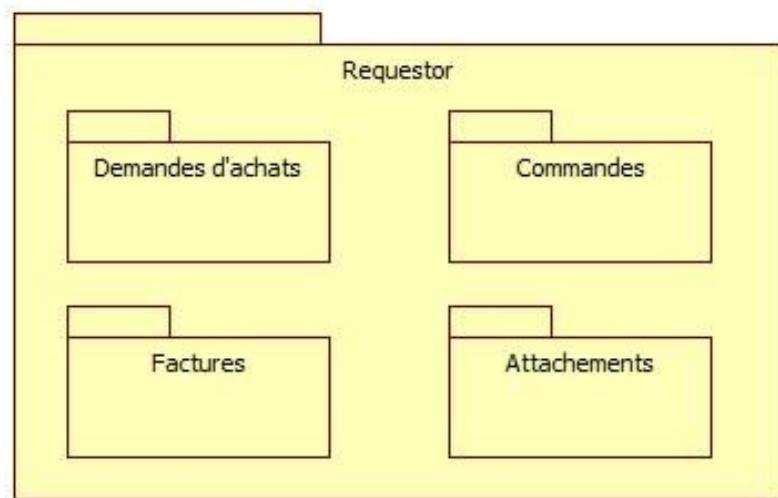


Figure 13: Diagramme de packages

### 2. Identification des cas d'utilisations

*Cette partie a pour objet de décrire les fonctionnalités offertes par notre application. En effet, un diagramme de cas d'utilisation accompagné de la liste des fonctionnalités et d'un enchainement de scénarios seront présentés pour chaque package.*

#### a. Package Demandes d'achat

*Nous présentons ci-après, le diagramme des cas d'utilisation du package « Demandes d'achat » et sa fiche descriptive.*

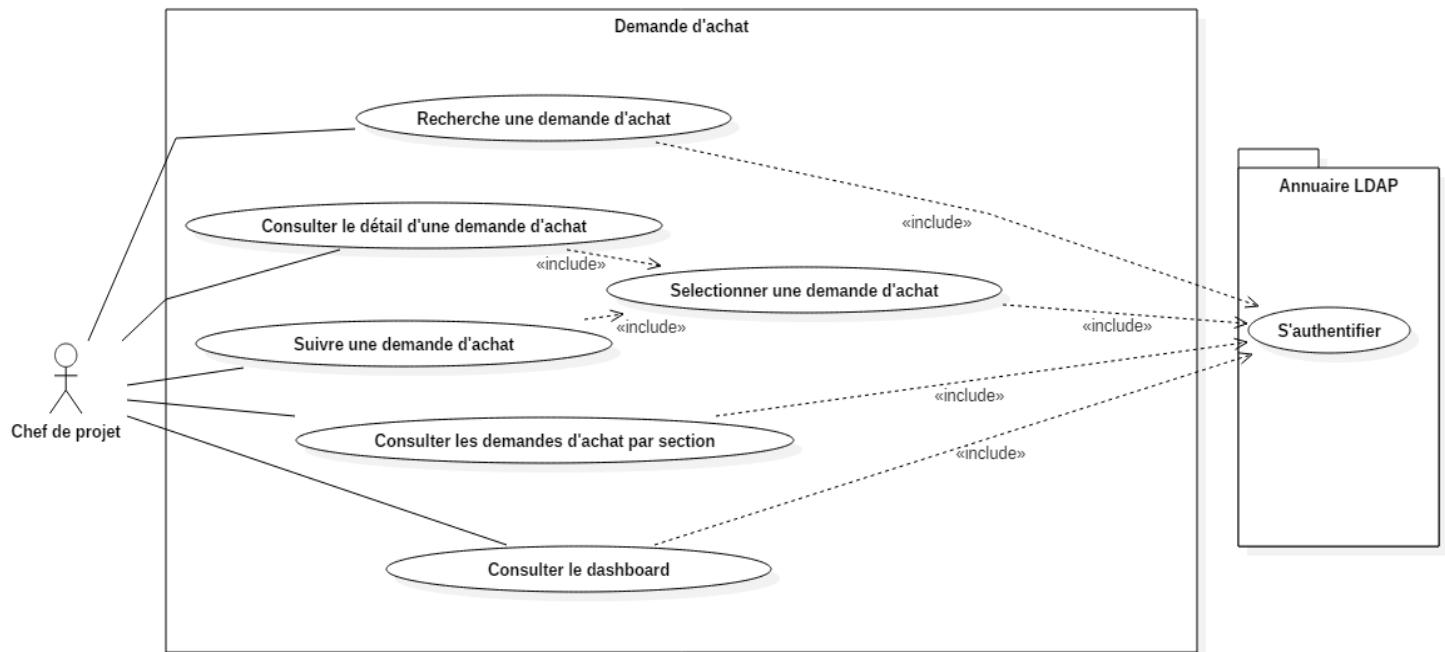


Figure 14: Diagramme des cas d'utilisation - Demandes d'achat

<b>Fonction</b>	<i>Suivi &amp; Détails des demandes d'achats</i>
<b>Objectif</b>	<i>Permettre à un chef de projet, de suivre l'ensemble de ses demandes d'achat triées par section, de consulter le détail de chaque demande, de suivre l'avancement de sa demande dans une timeline et de consulter un tableau de bord contenant des indicateurs clés sur l'ensemble de ses demandes.</i> <i>Ces différentes demandes passent toutes par un workflow.</i>
<b>Acteur</b>	<i>Chef de projet</i>
<b>Préconditions</b>	<i>L'acteur doit se connecter avec son login et son mot de passe</i>
<b>Enchainements</b>	<ul style="list-style-type: none"> <li>● <b>Scénario 1:</b> <i>Consulter les demandes d'achat par section</i> <ol style="list-style-type: none"> <li>1. <i>Le chef de projet se connecte à l'application.</i></li> <li>2. <i>Il choisit la rubrique : « Demandes d'achat ».</i></li> <li>3. <i>L'application renvoie la liste des demandes d'achat propre au chef de projet trié par quatre sections : demandes en préparation, demandes en attente d'approbation, demandes approuvées et demandes rejetées.</i></li> </ol> </li> <li>● <b>Scénario 2:</b> <i>Consulter le Dashboard</i> <ol style="list-style-type: none"> <li>1. <i>Le chef de projet se connecte à l'application.</i></li> <li>2. <i>Il choisit la rubrique : « Demandes d'achat ».</i></li> </ol> </li> </ul>

<p><b>3. L'application renvoie le Dashboard avec trois indicateurs clés : Pourcentage des DA lancées, pourcentage des DA en phase commerciale et pourcentage des DA en phase technique.</b></p> <ul style="list-style-type: none"> <li>● <b>Scénario 3:</b> Consulter le détail d'une demande d'achat</li> </ul> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Demandes d'achat ».</li> <li>3. l'application envoie la liste des demandes d'achat propre au chef de projet.</li> <li>4. Le chef de projet sélectionne une demande d'achat.</li> <li>5. Il visualise la liste des articles de la demande et il peut suivre sa demande d'achat sur une timeline.</li> </ol> <ul style="list-style-type: none"> <li>● <b>Scénario 4:</b> Chercher une demande d'achat</li> </ul> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il clique sur la loupe de recherche et tape un mot clé relatif à la demande d'achat.</li> </ol>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tableau 3: Fiche descriptive du package "Demandes d'achat"

## b. Package Commandes

Nous présentons ci-après, le diagramme des cas d'utilisation du package « Commandes » et sa fiche descriptive.

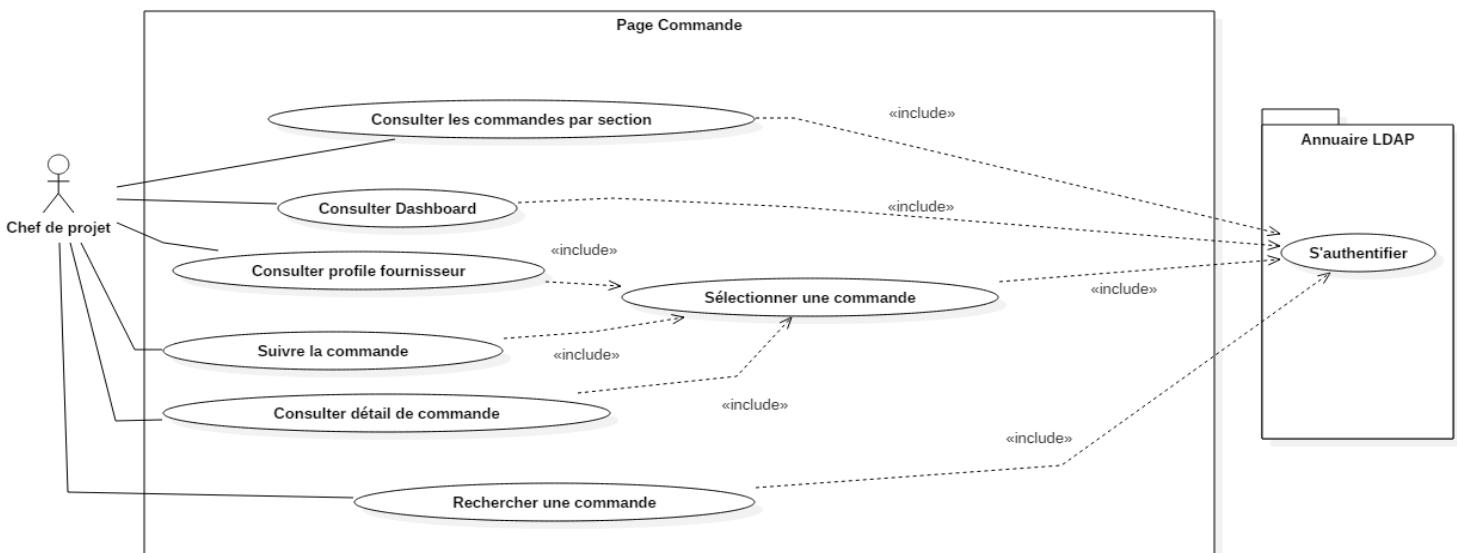


Figure 15: Diagramme des cas d'utilisation - Commandes

Fonction	<i>Suivi &amp; Détails des commandes</i>
<b>Objectif</b>	<p>Permettre à un chef de projet, de suivre l'ensemble de ses demandes commandes par section, de consulter le détail de chaque commande, de suivre l'avancement de sa commande dans une timeline et de consulter un tableau de bord contenant des indicateurs clés sur l'ensemble de ses commandes.</p> <p>Ces différentes commandes passent toutes par un workflow.</p>
<b>Acteur</b>	Chef de projet
<b>Préconditions</b>	L'acteur doit se connecter avec son login et son mot de passe
<b>Enchainements</b>	<ul style="list-style-type: none"> <li>● <b>Scénario 1 : Consulter les commandes par section</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Commandes ».</li> <li>3. L'application renvoie la liste des commandes propre au chef de projet trié par trois sections : nouvelles commandes, commandes en cours, commandes rejetées.</li> </ol> </li> <li>● <b>Scénario 2 : Consulter le Dashboard</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Commandes ».</li> <li>3. L'application renvoie le Dashboard avec trois indicateurs clés : le nombre de nouvelles commandes, nombre des commandes en cours de réception et nombre des commandes en cours de paiement.</li> </ol> </li> <li>● <b>Scénario 3 : Consulter le détail d'une commande</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Commandes ».</li> <li>3. envoie la liste des commandes propre au chef de projet.</li> <li>4. Le chef de projet sélectionne une commande.</li> <li>5. Il visualise la liste des articles de la commande et il peut suivre sa demande d'achat sur une timeline ou afficher le profil du fournisseur octroyé.</li> </ol> </li> <li>● <b>Scénario 4 : Chercher une commande</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il clique sur la loupe de recherche et tape un mot clé relatif à la commande : soit la référence soit le nom du fournisseur.</li> </ol> </li> </ul>

Tableau 4: Fiche descriptive du package "Commandes"

### c. Package Paiement

Nous présentons ci-après, le diagramme des cas d'utilisation du package « Paiement » et sa fiche descriptive.

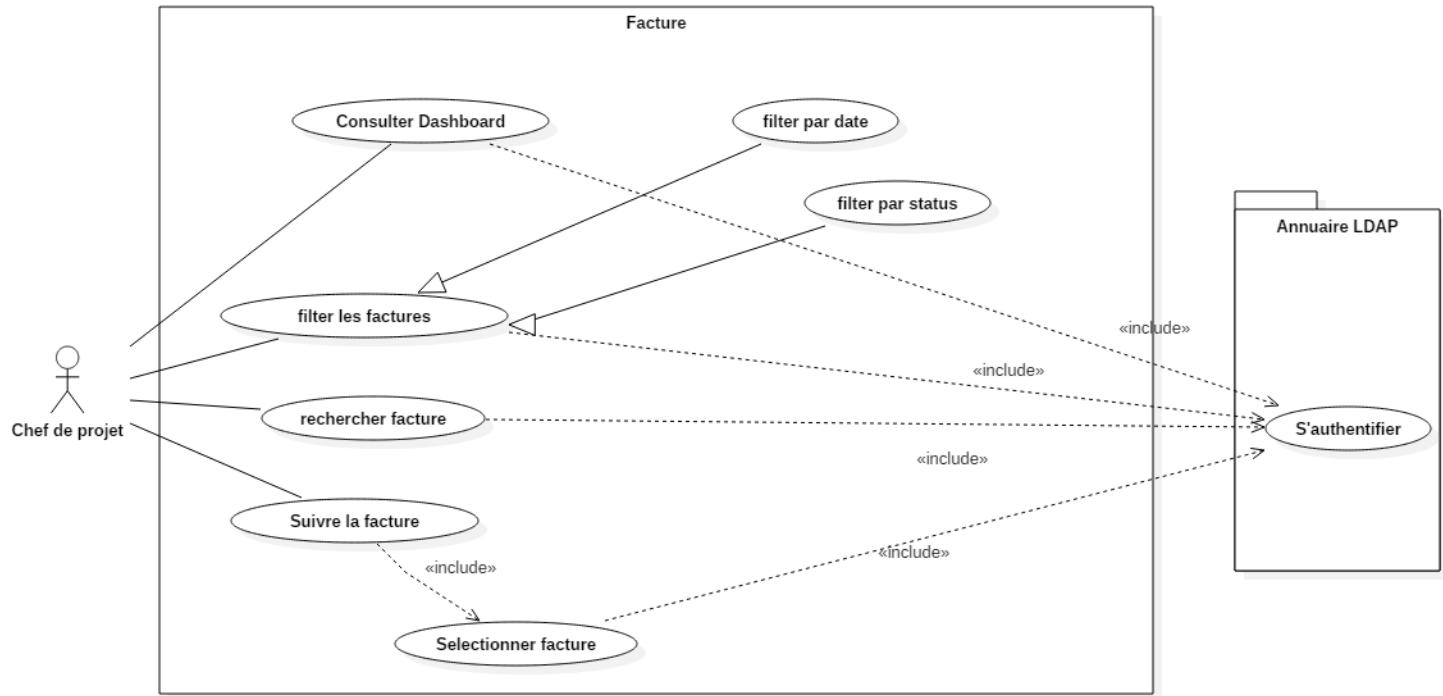


Figure 16: Diagramme des cas d'utilisation - Factures

<b>Fonction</b>	<i>Suivi &amp; Détails des factures</i>
<b>Objectif</b>	<i>Permettre à un chef de projet, de suivre l'ensemble de ses factures, de consulter le détail de chaque facture, de suivre l'avancement de sa facture dans une timeline et de consulter un tableau de bord contenant des indicateurs clés sur l'ensemble de ses factures.</i>  <i>Ces différentes factures passent toutes par un workflow.</i>
<b>Acteur</b>	<i>Chef de projet</i>
<b>Préconditions</b>	<i>L'acteur doit se connecter avec son login et son mot de passe</i>
<b>Enchainements</b>	<ul style="list-style-type: none"> <li>● <b>Scénario 1 : Consulter les factures</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Paiement ».</li> <li>3. L'application renvoie la liste des factures propre au chef de projet.</li> </ol> </li> </ul>

	<ul style="list-style-type: none"> <li>● <b>Scénario 2 : Consulter le Dashboard</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Paiement ».</li> <li>3. L'application renvoie le Dashboard avec six indicateurs clés : le montant et le nombre des factures échues, le montant et le nombre des factures non échues, le montant et le nombre des factures reçues, le montant et le nombre des factures en cours de paiement, nombre des factures réglées.</li> </ol> </li> <li>● <b>Scénario 3 : Consulter le détail d'une facture</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Paiement ».</li> <li>3. L'application envoie la liste des factures propre au chef de projet.</li> <li>4. Le chef de projet sélectionne une facture.</li> <li>5. Le chef de projet consulte l'avancement de sa facture sur une timeline contenant l'ensemble des événements relatifs à la facture ainsi que leurs dates.</li> </ol> </li> <li>● <b>Scénario 4 : Filtrer les factures</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Paiement ».</li> <li>3. L'application envoie la liste des factures propre au chef de projet.</li> <li>4. Le chef de projet peut filtrer ses factures par date et par statuts de facture.</li> </ol> </li> <li>● <b>Scénario 5 : Chercher une facture</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il clique sur la loupe de recherche et tape un mot clé relatif à la facture : soit la référence, le nom du fournisseur ou la référence de la commande.</li> </ol> </li> </ul>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tableau 5: Fiche descriptive du package "Paiement"

## d. Package Attachements

Nous présentons ci-après, le diagramme des cas d'utilisation du package « Attachements » et sa fiche descriptive.

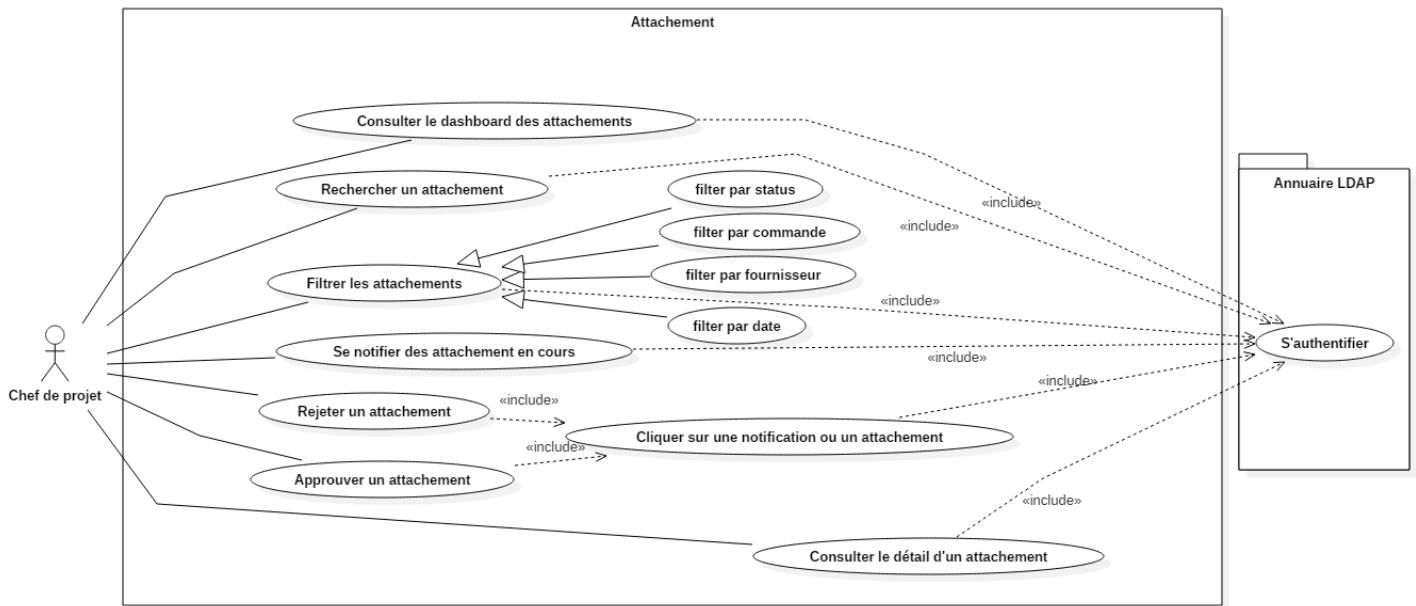


Figure 17: Diagramme des cas d'utilisation - Attachements

<b>Fonction</b>	<i>Lister et gérer les attachments</i>
<b>Objectif</b>	<i>Permettre à un chef de projet, de consulter l'ensemble de ses attachments, de consulter le détail de chaque attachment, de valider ou de rejeter un attachment, d'évaluer le fournisseur selon des critères qui varient selon le type de la commande et de consulter un tableau de bord contenant des indicateurs clés sur l'ensemble de ses attachments.</i>
<b>Acteur</b>	<i>Chef de projet</i>
<b>Préconditions</b>	<i>L'acteur doit se connecter avec son login et son mot de passe</i>
<b>Enchainements</b>	<ul style="list-style-type: none"> <li>● <b>Scénario 1 : Consulter les attachments</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Attachements ».</li> <li>3. L'application renvoie la liste des attachments propre au chef de projet divisée par deux sections : Nouveaux attachments et attachments traités.</li> </ol> </li> <li>● <b>Scénario 2 : Consulter le Dashboard</b> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Attachements ».</li> </ol> </li> </ul>

<p><b>3. L'application renvoie le Dashboard avec trois indicateurs clés : le montant et le nombre des attachments acceptés, le montant et le nombre des commandes acceptées et le montant et le nombre des attachments commandés.</b></p> <ul style="list-style-type: none"> <li>● <b>Scénario 3 : Consulter le détail d'un attachment</b></li> </ul> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Attachements ».</li> <li>3. L'application envoie la liste des attachments propre au chef de projet.</li> <li>4. Le chef de projet sélectionne un attachment.</li> <li>5. Le chef de projet consulte le détail de l'attachment.</li> <li>6. Le chef de projet peut soit rejeter soit confirmer un attachment. Si la quantité attachée dépasse la quantité commandée, l'application notifie le chef de projet et ne lui donne pas la main de valider l'attachment.</li> <li>7. Dans le cas de confirmation de l'attachment le chef de projet évalue le fournisseur selon des critères qui diffèrent selon le type de la commande.</li> </ol> <ul style="list-style-type: none"> <li>● <b>Scénario 4 : Filtrer les attachments</b></li> </ul> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il choisit la rubrique : « Attachements ».</li> <li>3. L'application envoie la liste des attachments propre au chef de projet.</li> <li>4. Le chef de projet peut filtrer ses attachments par date, par commande, par fournisseur et par statuts.</li> </ol> <ul style="list-style-type: none"> <li>● <b>Scénario 5 : Chercher un attachment</b></li> </ul> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. Il clique sur la loupe de recherche et tape un mot clé relatif à l'attachment : soit la référence, le nom du fournisseur ou la référence de la commande.</li> </ol> <ul style="list-style-type: none"> <li>● <b>Scénario 6 : Recevoir des notifications</b></li> </ul> <ol style="list-style-type: none"> <li>1. Le chef de projet se connecte à l'application.</li> <li>2. S'il y a un nouvel attachment à traiter, il reçoit une notification en cliquant dessus, il est dirigé vers le détail de l'attachment.</li> </ol>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tableau 6: Fiche descriptive du package "Attachments"

## II. Spécifications techniques

Durant cette phase nous allons présenter l'architecture logicielle adoptée, citer les Design patterns utilisés par la suite, en fin, nous allons étaler les Frameworks utilisés.

### 1. Architecture

L'architecture en couches est la conséquence inéluctable d'une approche qui s'appuie sur la réalisation de composants réutilisables.

Dans le but de réaliser un système puissant, évolutif et modulaire nous avons adopté une architecture micro service. L'idée étant de découper les grandes fonctionnalités de l'application en petites unités implémentée sous forme de micro-services. Chaque service est responsable d'une fonctionnalité.

Notre application est séparée en trois couches logiques distinctes :

- **Back end** : Il s'agit du niveau de données ou la couche de persistance de l'application. Dans le cas de notre application, le back end est présenté comme un ensemble de services qui exploitent les données présentes dans les bases de données. Ces services sont exposés au Middleware pour les consommer.
- **Middleware** : Essentiellement, Middleware relie le front end et le back end du système en agissant comme un pont. Il est souvent le lien entre les données et l'interface utilisateur. Le middleware consomme les services exposés par le back end et envoie le résultat au front end.
- **Front end** : représente la partie présentation de l'application, son rôle est d'afficher le résultat des requêtes, en consommant les services du middleware.

Nous présentons ci-après une figure illustrant l'architecture utilisée.

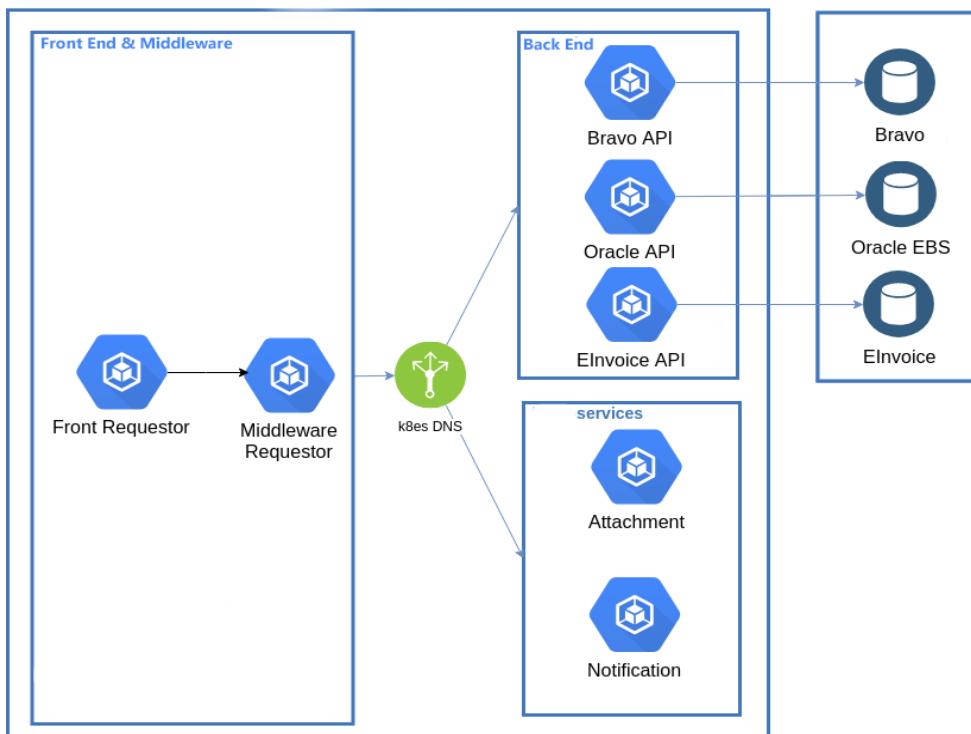


Figure 18: architecture de l'application

## 2. Design Patterns

*Les design patterns se présentent comme étant des solutions de conception communes à un problème récurrent dans un contexte donné. Ainsi, leur usage apporte une évolutivité, une lisibilité et une efficacité au développement. Par ailleurs, ils offrent un transfert de compétence rapide et conception orienté objet, dans la mesure où ils représentent un catalogue des meilleures pratiques à adopter.*

*A ce titre nous présentons ci-dessous les designs patterns dont nous avons fait usage dans la conception de notre projet.*

### a. Inversion de contrôle (IOC)

*L'inversion de contrôle est « un patron d'architecture » visant à réduire le couplage inter ou intra couches applicatives. Ce pattern est aussi connu sous le nom d'injection de dépendance, et utilisé comme pattern principale pour le Framework Spring.*

*Il concerne la manière dont un objet obtient une référence vers ses dépendances. L'approche habituelle pour obtenir ces dépendances au sein d'un objet, que ce soit via un singleton ou un autre mécanisme de Lookup (Service Locator, instantiation directe, ...), a le désavantage d'introduire un couplage fort entre l'objet et ce mécanisme de Lookup.*

*Avec l'inversion de contrôle, les dépendances de l'objet lui sont passées à travers son constructeur ou par ses setters une fois instancié « d'où le nom du pattern injection des dépendances ».*

### b. Objet de transfert de données (DTO)

*Dans la conception d'applications distribuées, et pour satisfaire une seule requête client, l'on se trouve souvent obligé d'émettre un nombre important d'appels vers une interface distante, ce qui accroît le temps de réponse au-delà de l'acceptable.*

*L'objet de transfert de données, de l'anglais Data Transfer Object (DTO), dit aussi « Value Object », est un modèle de conception qui tente de résoudre la problématique suivante : préserver la simplicité de la sémantique d'une interface d'appel de procédure sans être soumis aux problèmes de latence inhérents à la communication distante.*

*La solution consiste à créer un DTO qui contient toutes les données nécessaires à l'appel distant. Modifier la signature de la méthode distante pour qu'elle accepte le DTO en tant que paramètre unique et pour qu'elle renvoie un paramètre DTO unique au client.*

### c. MVP (Model View Presenter)

*Le design pattern MVP, consiste à diviser la couche d'interface graphique en deux. La partie la plus basse, le « Presenter », sera la seule à communiquer avec la couche métier. Ainsi, un système MVP est une version évoluée du MVC (Model View Controller), où la vue*

reçoit les événements de l'interface utilisateur et appelle le « Presenter » au besoin. Le « Presenter » quant à lui est responsable de la mise à jour de la vue avec les nouvelles données générées par la couche métier.

Ainsi, comme le "Presenter" ne se contente pas de répondre à des sollicitations de l'interface graphique mais prend également en charge son pilotage, la dépendance est bidirectionnelle entre les deux parties. Afin de minimiser cette dépendance, le "Presenter" communique avec le "View" au travers d'une interface. L'interface est définie dans le "Presenter" alors que le "View" doit contenir une classe qui l'implémente.

#### d. Decorator

Le design pattern Décorateur appartient à la famille des design patterns structurels, il attache dynamiquement des responsabilités supplémentaires à un objet. Il fournit une alternative souple à la dérivation, pour étendre les fonctionnalités.

Chaque composant peut être utilisé seul ou enveloppé par un décorateur, Les décorateurs ajoutent généralement le nouveau comportement en effectuant un traitement avant ou après une méthode existante dans le composant.

### 3. Framework

#### a. Spring boot



Figure 19: logo de spring boot

Spring Boot est un projet ou un micro Framework qui a notamment pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet.

Spring Boot offre plusieurs avantages notamment en termes de déploiement applicatif. Habituellement, le déploiement d'une application Spring nécessite la génération d'un fichier .war qui doit être déployé sur un serveur comme un Apache Tomcat. Spring Boot simplifie ce mécanisme en offrant la possibilité d'intégrer directement un serveur Tomcat dans votre exécutable. Au lancement de celui-ci, un Tomcat embarqué sera démarré afin de faire tourner votre application.

Enfin, Spring Boot met à disposition des opérationnels, des métriques qu'ils peuvent suivre une fois l'application déployée en production. Pour cela Spring Boot utilise « Actuator » qui est un système qui permet de monitorer une application via des URLs spécifiques ou des commandes disponibles via SSH.

## b. React JS

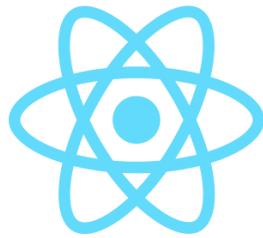


Figure 20: logo de React

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un Framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.

Par ailleurs, React n'utilise pas de système de templates et ne fonctionne qu'avec du Javascript, permettant une encapsulation complète du composant au sein d'une unique classe. Pour faciliter l'écriture de la vue, l'équipe initiale chez Facebook a développé un langage, JSX, qui permet de générer des objets Javascript avec une notation similaire à HTML.

## c. Redux



Figure 21: logo de Redux

*Redux est une bibliothèque JavaScript open-source pour gérer l'état de l'application. Il est le plus souvent utilisé avec des librairies telles que React ou Angular pour créer des interfaces utilisateur. Similaire à (et inspiré par) l'architecture Flux de Facebook, il a été créé par Dan Abramov et Andrew Clark.*

*Redux est une petite bibliothèque avec une API simple et limitée conçue pour être un conteneur prévisible pour l'état de l'application. Il fonctionne de la même manière qu'une fonction réductrice, un concept de programmation fonctionnelle.*

*Il est influencé par le langage de programmation fonctionnel.*

*Redux résout un problème qui peut ne pas être clair au début : il aide à donner à chaque composant React l'état exact dont il a besoin. Redux soutient l'état dans un seul endroit. De même, avec Redux, la logique de récupération et de gestion de l'état ne dépend pas de React.*

#### d. Redux-Saga



Figure 22: logo de Redux-saga

*Redux-saga est une bibliothèque qui vise à rendre les effets secondaires de l'application (par exemple, les tâches asynchrones telles que l'extraction de données et l'accès à la mémoire) plus faciles à gérer, plus efficaces à exécuter et simples à tester.*

*Saga est comme un fil séparé dans votre application qui est seul responsable des effets secondaires. Redux-saga est un middleware redux, ce qui signifie que ce thread peut être démarré, suspendu et annulé à partir de l'application principale avec des actions de redux normales, il a accès à l'état redux de l'application et peut aussi distribuer des actions redux.*

## Conclusion

*Au cours de chapitre, nous avons énoncé les spécifications fonctionnelles et techniques de notre système au cours desquelles les besoins fonctionnels ont été traduits en diagrammes de packages, cas d'utilisation et en scénarios. Les besoins techniques par l'architecture de l'application ainsi que les Frameworks et les design patterns utilisés.*

*Le chapitre suivant portera sur l'analyse et la conception de notre projet où nous présenterons nos diagrammes de classes en plus des différents scénarios.*

## Chapitre IV : Analyse et conception

---

*Dans ce chapitre, nous allons présenter les diagrammes de classes candidates, le diagramme de classes détaillé ainsi que les différents scénarios accompagnés des diagrammes de séquence pour chaque partie de l'application.*

## I. Page des demandes d'achats

### 1. Diagramme des classes candidates

Nous allons commencer par décrire nos classes candidates dans le tableau suivant avant de présenter notre diagramme de classes candidates.

Classe	Description
<b>Requestor</b>	Désigne le chef de projet
<b>PurchaseRequest</b>	Désigne la demande d'achat
<b>PurchaseRequestEvent</b>	Désigne à un événement sur la timeline de la demande d'achat
<b>PurchaseRequestDashboard</b>	Désigne le tableau de bord des demandes d'achat
<b>Items</b>	Désigne les articles de la demande d'achat

Tableau 7: Description des classes candidates du package "Demandes d'achat"

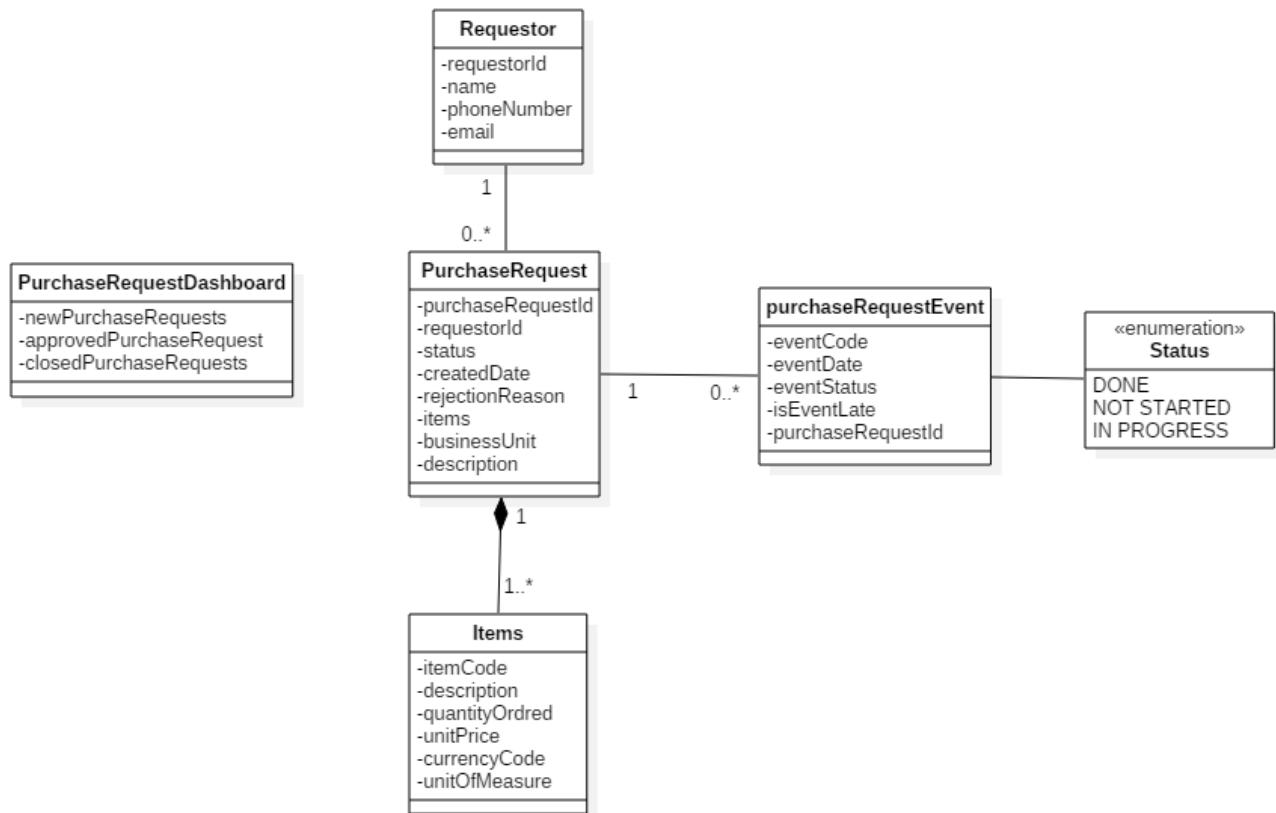


Figure 23: Diagramme des classes candidates - Demandes d'achat

## 2. Diagrammes de séquences

Nous présentons ci-après la liste des scénarios pour le cas d'utilisation pour le package demandes d'achat :

- **Scénario 1 : Consulter les demandes d'achat par section**
- **Scénario 2 : Consulter le Dashboard**
- **Scénario 3 : Consulter le détail d'une demande d'achat**
- **Scénario 4 : Chercher une demande d'achat**

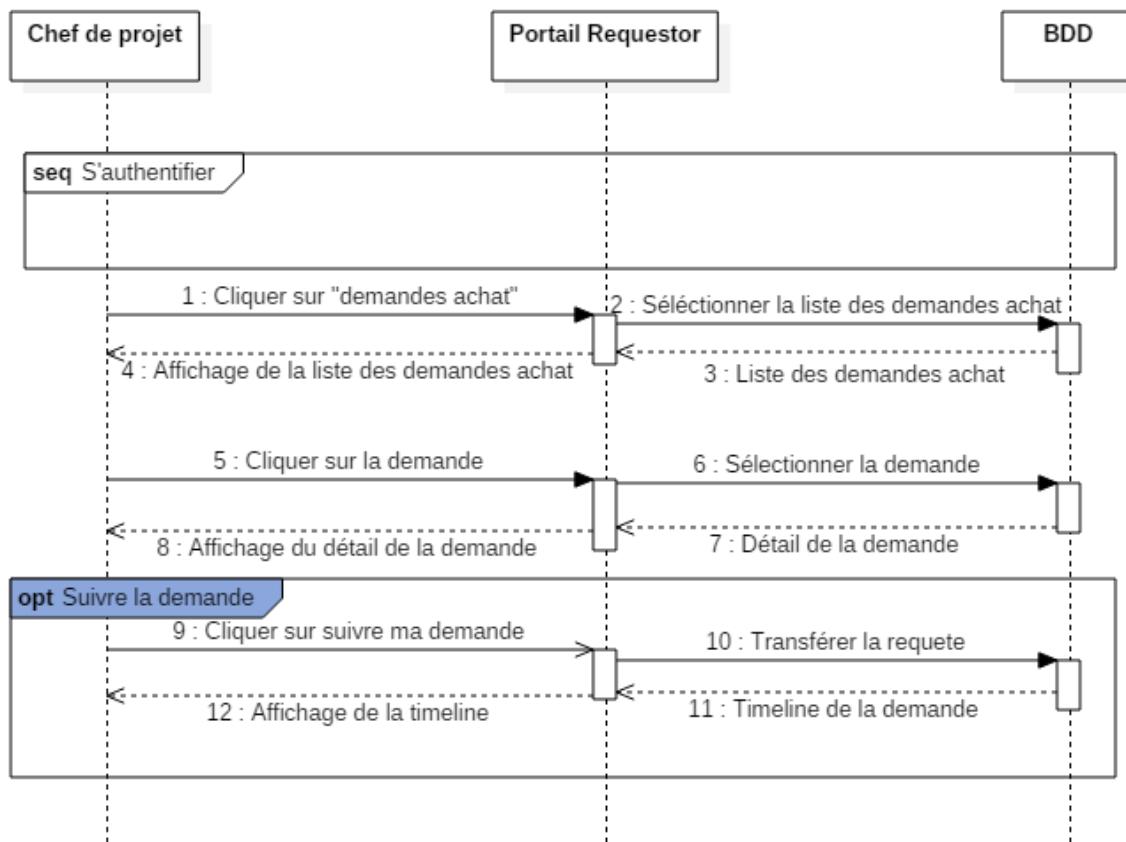


Figure 24: Diagramme de séquence - Consulter détail d'une demande d'achat

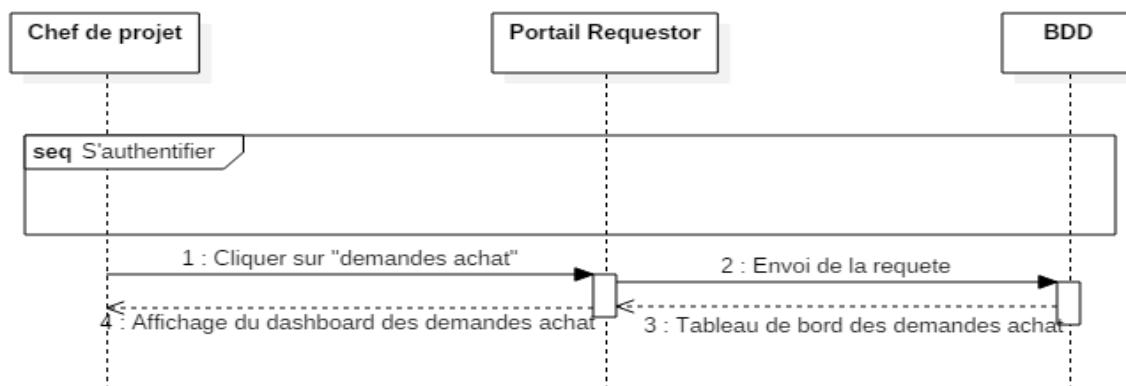


Figure 25: Diagramme de séquence - Consulter le Dashboard des demandes d'achats

## II. Page des commandes

### 1. Diagramme des classes candidates

Classe	Description
<b>Requestor</b>	Désigne le chef de projet
<b>PurchaseRequest</b>	Désigne la demande d'achat
<b>PurchaseOrder</b>	Désigne la commande liée à la demande d'achat
<b>PurchaseOrderEvent</b>	Désigne à un événement sur la timeline de la commande
<b>Supplier</b>	Désigne le fournisseur octroyé à la commande.
<b>PurchaseOrderDashboard</b>	Désigne le tableau de bord des commandes
<b>Items</b>	Désigne les articles de la commande

Tableau 8: Description des classes candidates du package "Commandes"

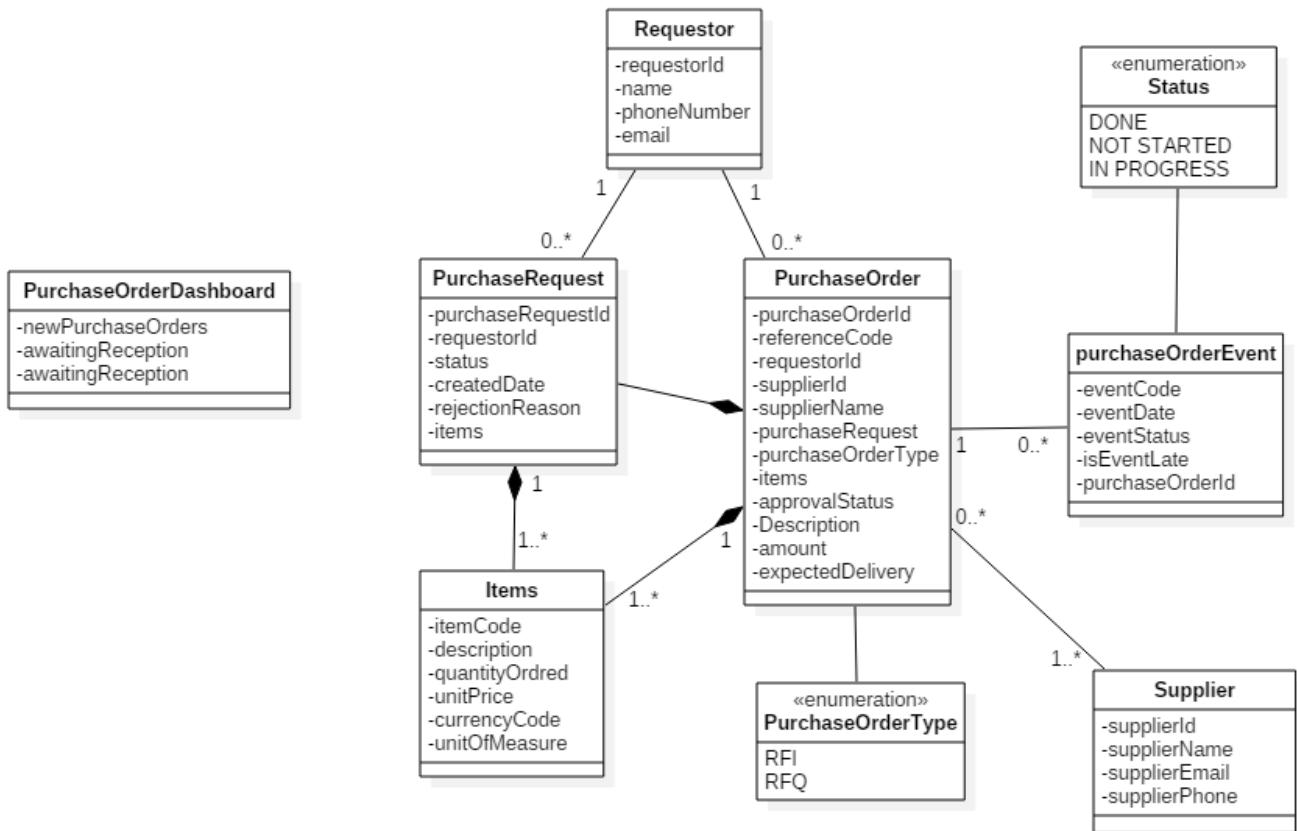


Figure 26: Diagramme des classes candidates – Commandes

## 2. Diagrammes de séquences

Nous présentons ci-après la liste des scénarios pour le cas d'utilisation pour le package commande :

- **Scénario 1 : Consulter les commandes par section**
- **Scénario 2 : Consulter le Dashboard**
- **Scénario 3 : Consulter le détail d'une commande**
- **Scénario 4 : Chercher une commande**

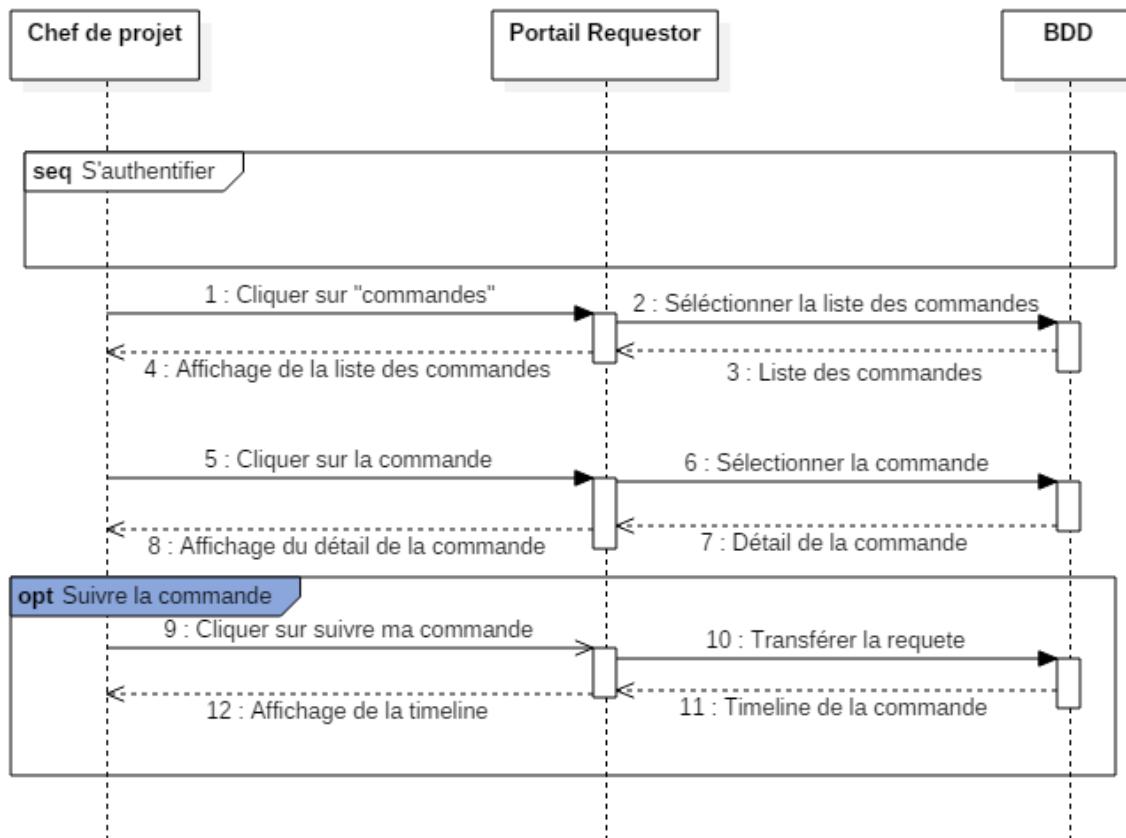


Figure 27: Diagramme de séquence - Consulter le détail d'une commande

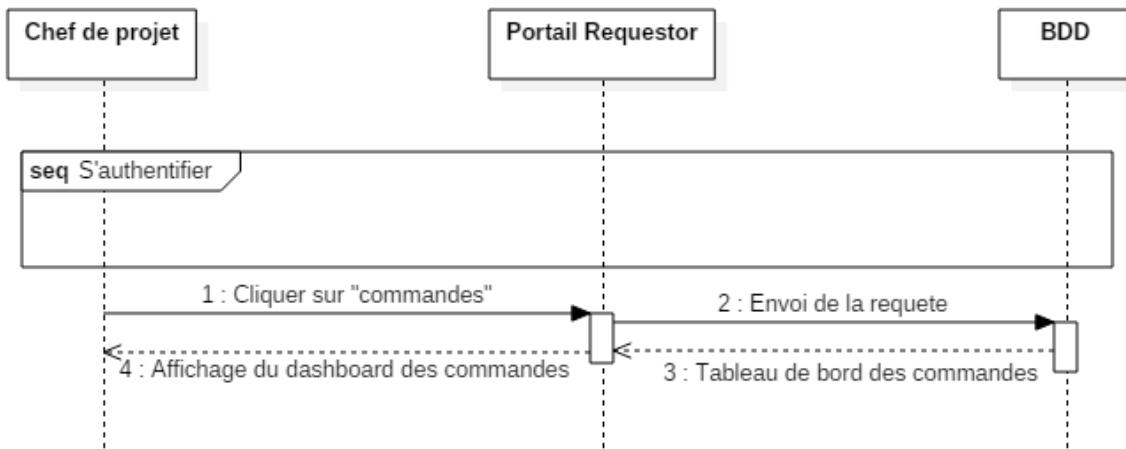


Figure 28: Diagramme de séquence - Consulter le Dashboard des commandes

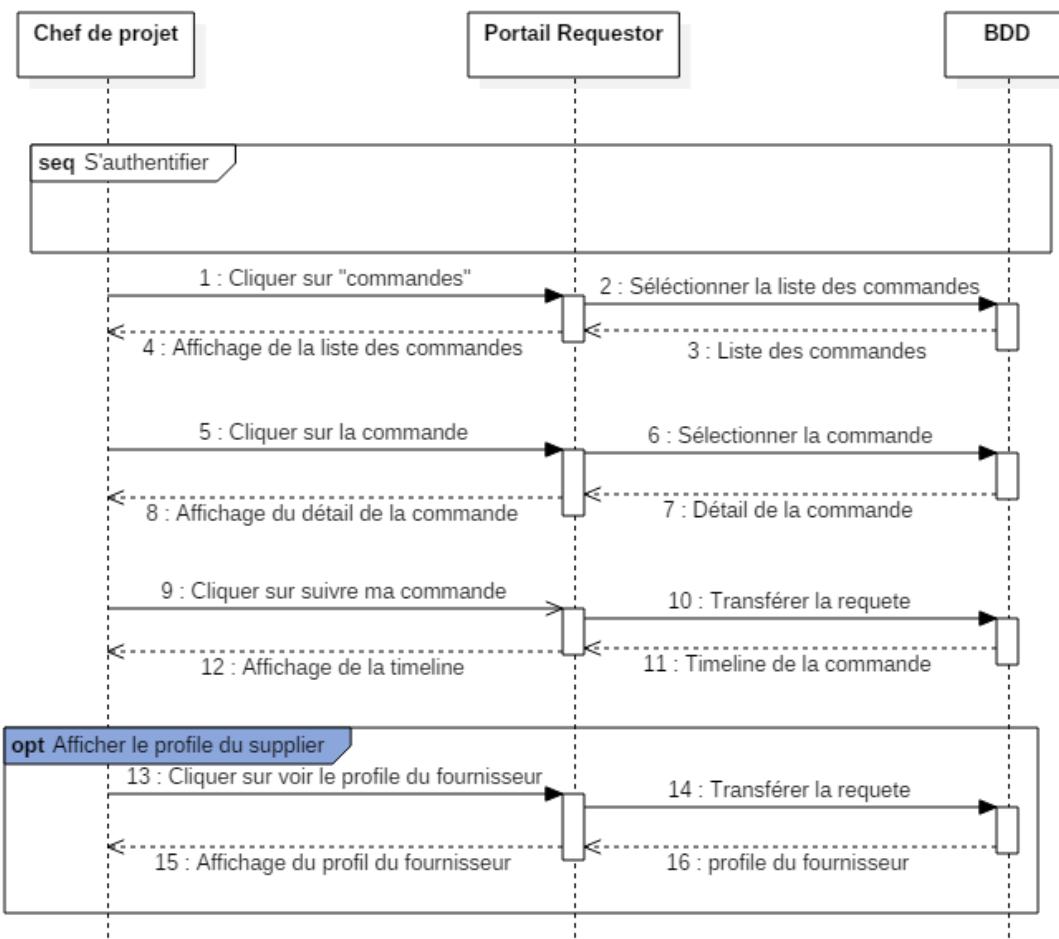


Figure 29: Diagramme de séquence - Voir le profil du fournisseur

### III. Page de paiement

#### 1. Diagramme des classes candidates

Classe	Description
<b>Requestor</b>	Désigne le chef de projet
<b>Invoice</b>	Désigne la facture issue d'Oracle EBS
<b>EInvoice</b>	Désigne la facture issue de E-Invoice
<b>PurchaseOrder</b>	Désigne la commande liée à la facture
<b>InvoiceEvent</b>	Désigne à un événement sur la timeline de la facture
<b>InvoiceFilter</b>	Désigne le filtre à appliquer sur la liste des factures
<b>InvoiceDashboard</b>	Désigne le tableau de bord des factures

Tableau 9: Description des classes candidates du package "Paiement"

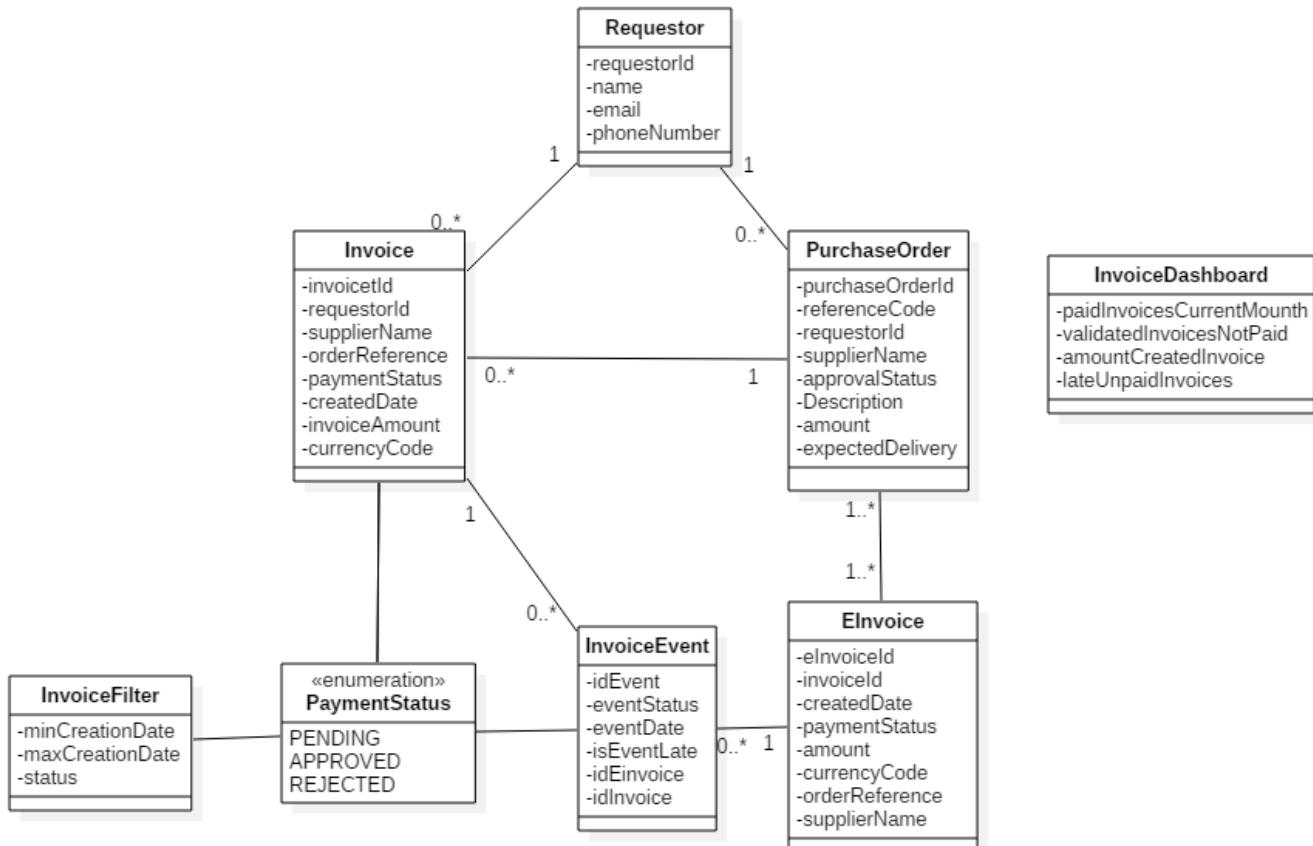


Figure 30: Diagramme de classes candidates - Paiement

## 2. Diagrammes de séquences

Nous présentons ci-après la liste des scénarios pour le cas d'utilisation pour le package facture :

- **Scénario 1 : Consulter les factures**
- **Scénario 2 : Consulter le Dashboard**
- **Scénario 3 : Consulter le détail d'une facture**
- **Scénario 4 : Chercher une facture**
- **Scénario 5 : Filtrer les factures**

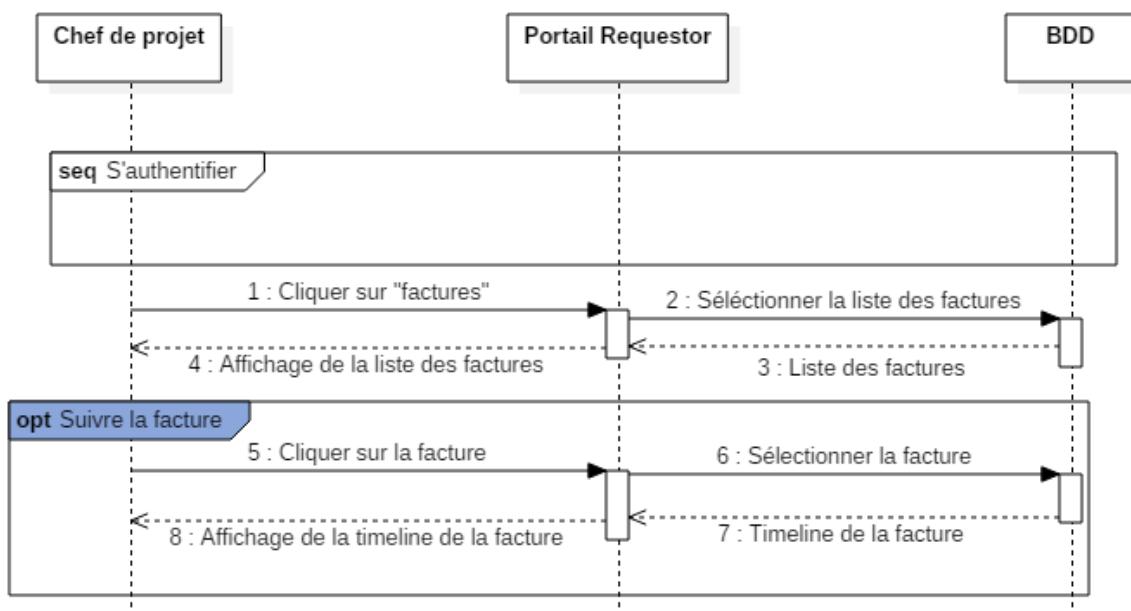


Figure 31: Diagramme de séquences - Détails d'une facture

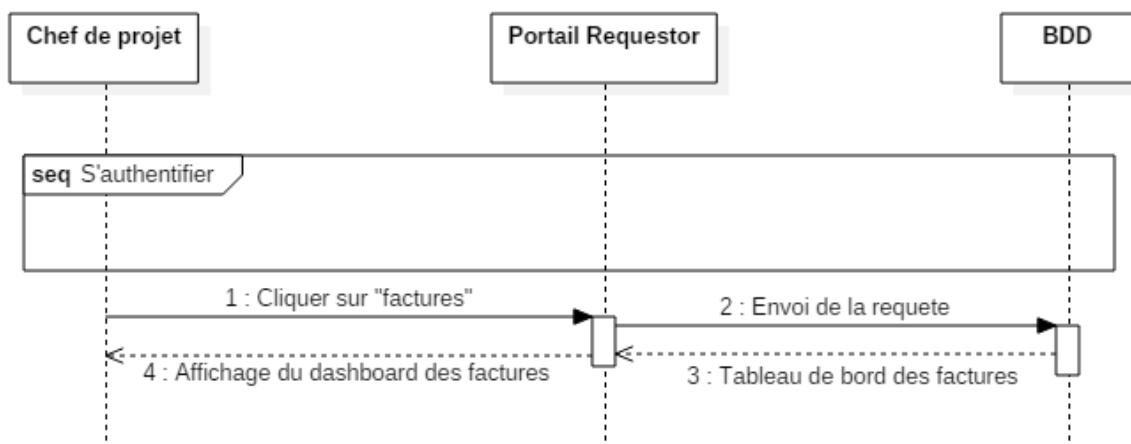


Figure 32: Diagramme de séquences - Consulter le Dashboard des factures

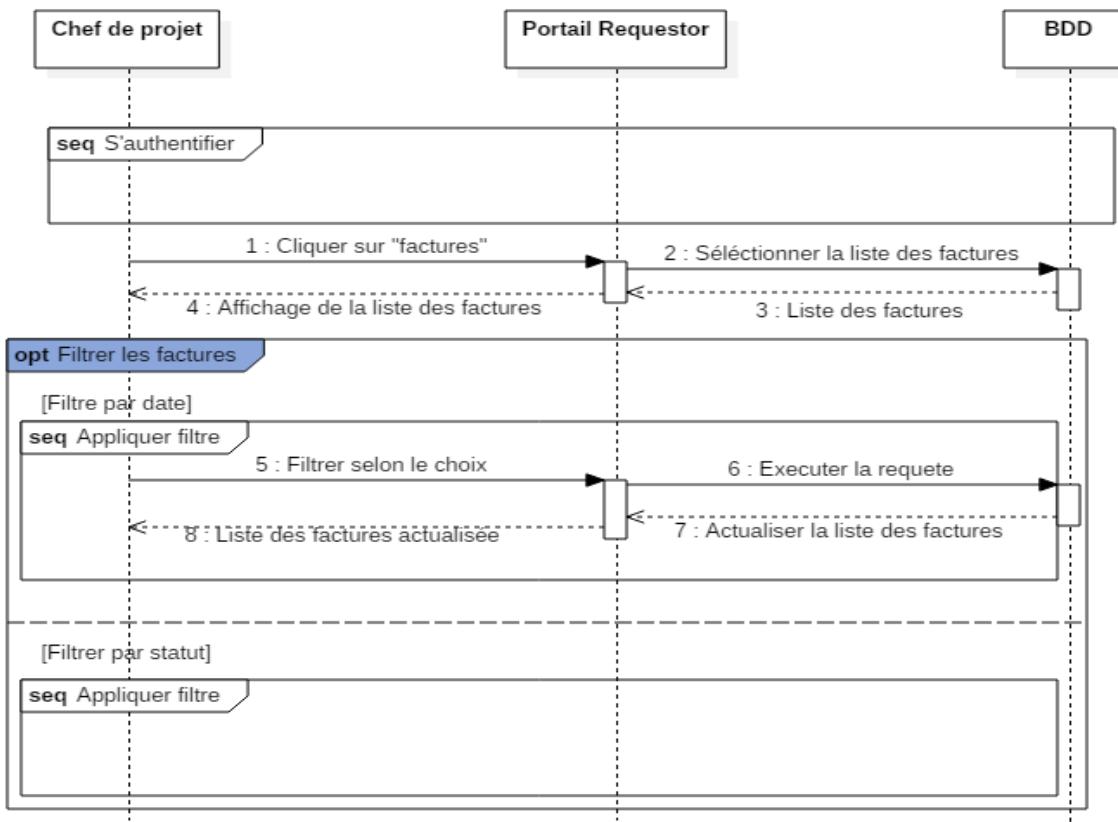


Figure 33: Diagramme de séquences - Filtrer les factures

## IV. Page des attachments

### 1. Diagramme des classes candidates

Classe	Description
<b>Requestor</b>	Désigne le chef de projet
<b>Attachment</b>	Désigne l'attachment lié à la commande
<b>PurchaseOrder</b>	Désigne la commande
<b>AttachmentItem</b>	Désigne les articles de l'attachment
<b>Notification</b>	Désigne la notification
<b>Criterea</b>	Désigne les critères d'évaluation du fournisseur
<b>Evaluation</b>	Désigne l'évaluation du fournisseur
<b>AttachmentDashboard</b>	Désigne le tableau de bord des commandes
<b>AttachmentFilter</b>	Désigne les filtres à appliquer sur la liste des attachments

Tableau 10: Description des classes candidates du package "Attachments"

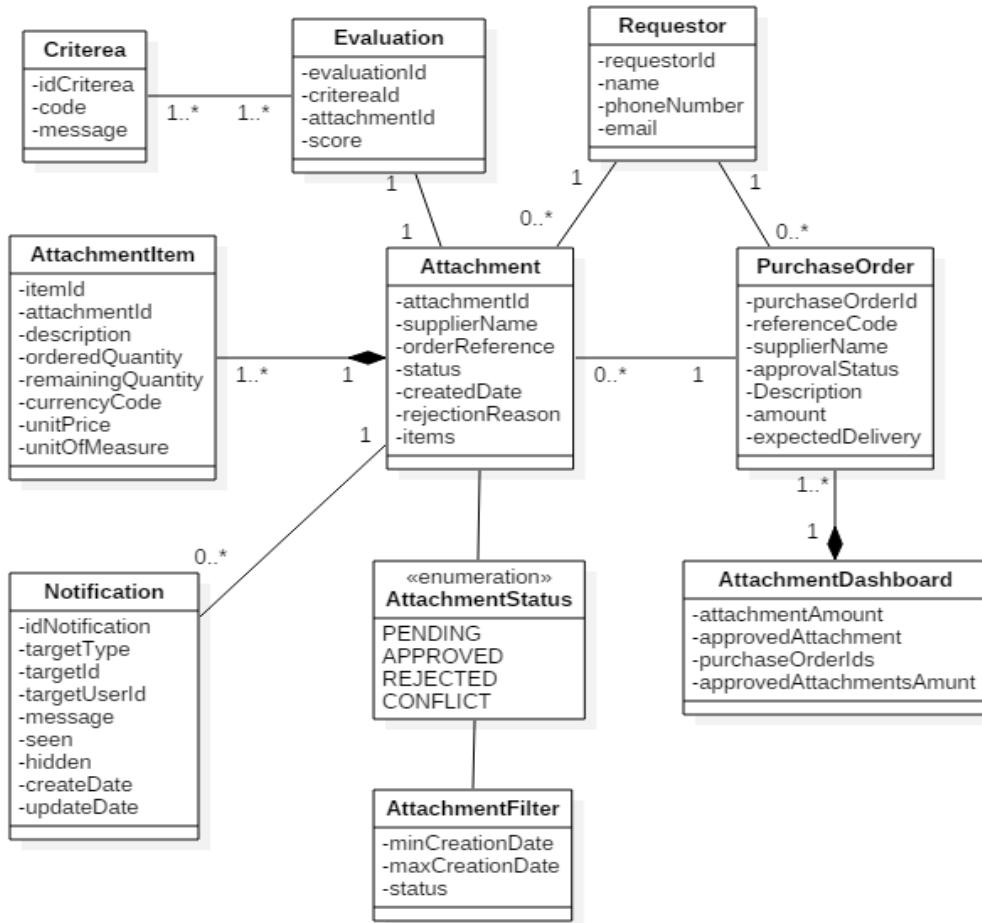


Figure 34: Diagramme de classes candidates - Attachements

## 2. Diagrammes de séquences

Nous présentons ci-après la liste des scénarios pour le cas d'utilisation pour le package facture :

- **Scénario 1 : Consulter les attachments**
- **Scénario 2 : Consulter le Dashboard**
- **Scénario 3 : Consulter le détail d'un attachment**
- **Scénario 4 : Chercher un attachment**
- **Scénario 5 : Filtrer les attachments**

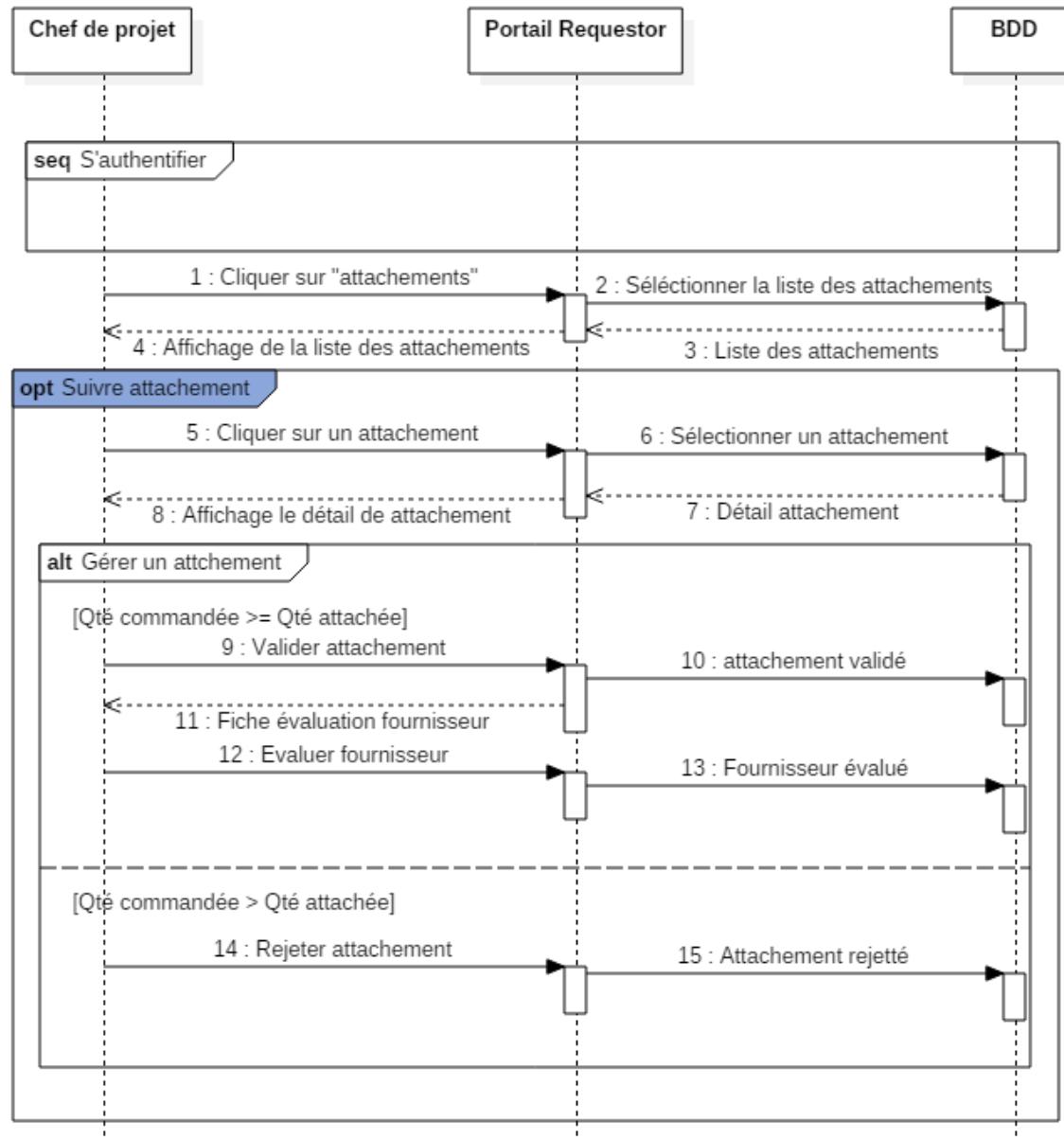


Figure 35: Diagramme de séquences - Suivre un attachement

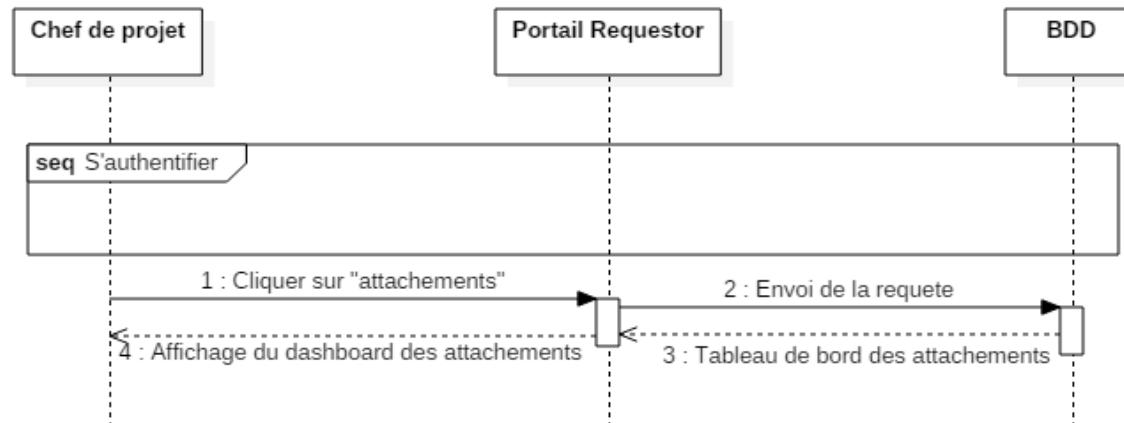


Figure 36: Diagramme de séquences - Consulter le dashboard des attachments

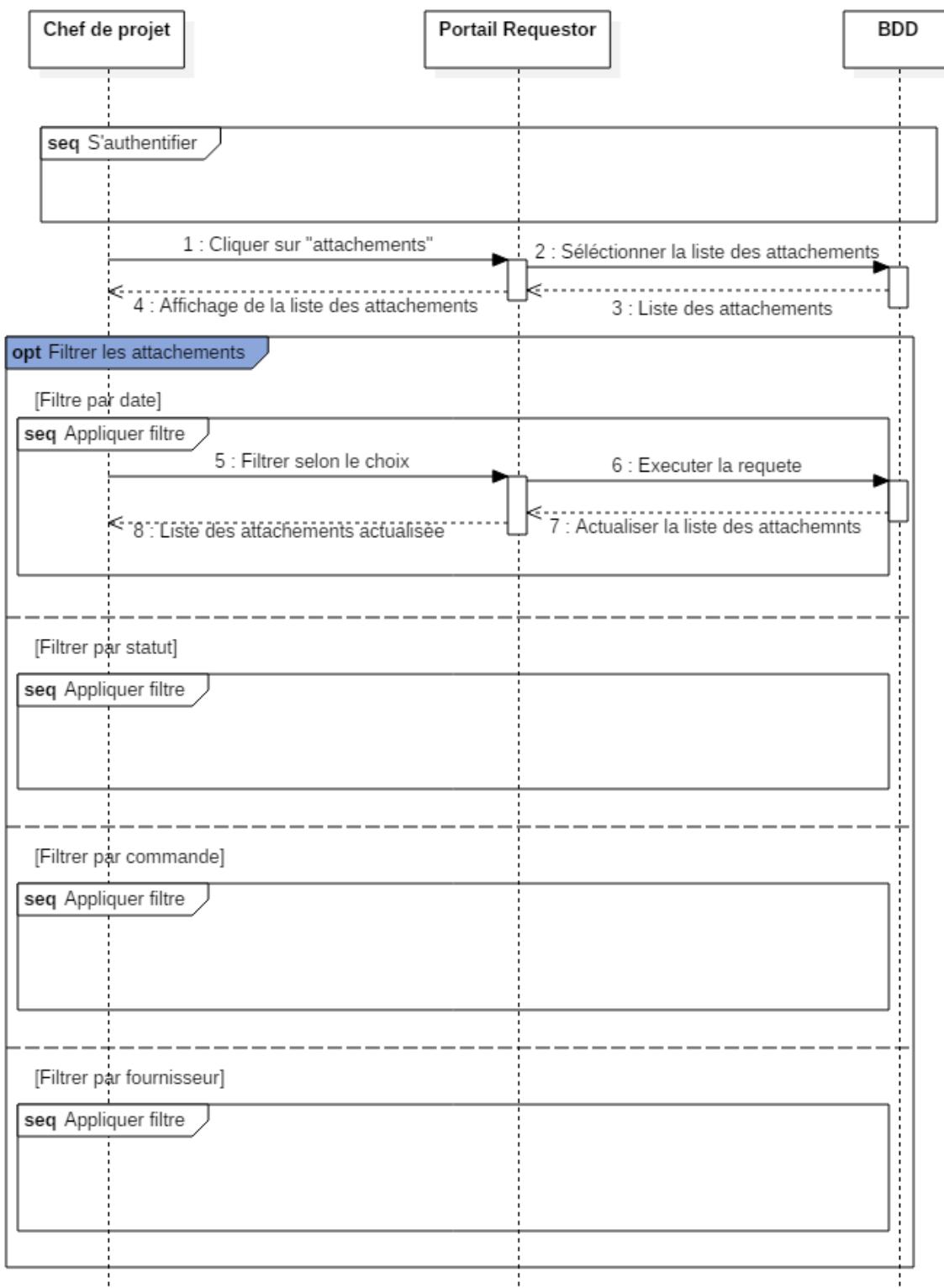


Figure 37: Diagramme de séquences - Filtrer les attachments

## V. Conception générale de l'application:

A ce niveau, on va projeter la partie fonctionnelle sur l'architecture logicielle selon les couches qu'elle présente à savoir : la couche présentation, la couche métier, la couche accès aux données.

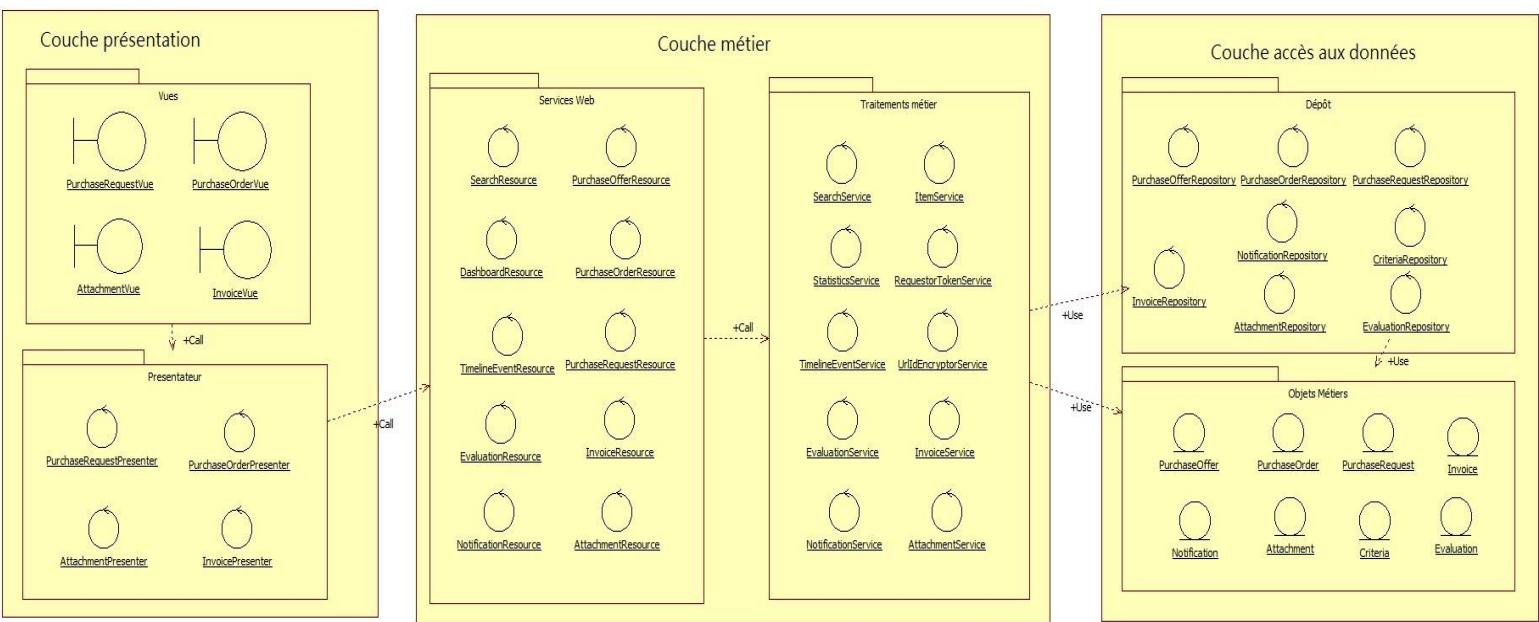


Figure 38: Conception générale de l'application

## Conclusion

Ce chapitre est scindé en quatre parties représentant chacune un module. Dans chaque module, nous commençons par détailler les diagrammes de classes candidates puis nous présentons les différents scénarii en les illustrant par des diagrammes de séquence et enfin on a conclu par un diagramme représentant la conception générale de l'application.

Dans le chapitre suivant, il s'agira de transformer ces diagrammes en écrans fonctionnels.

## Chapitre V : Réalisation

---

Ce chapitre qui clôture notre rapport est dédié à la réalisation et la mise en œuvre de l'application. Dans cette partie, on va commencer par parcourir en détail l'environnement technique du projet. On étalera par la suite les différentes étapes de mise en œuvre de l'application. Enfin, quelques IHM seront exposées pourachever cet élément.

## I. Outils et technologies utilisés

### 1. Environnement de développement

#### a. IntelliJ IDEA



Figure 39: logo de IntelliJ IDEA

*IntelliJ IDEA est un environnement de développement intégré (IDE) Java pour le développement de logiciels informatiques. Il est développé par JetBrains (anciennement connu sous le nom d'IntelliJ), et est disponible sous la forme d'une édition communautaire sous licence Apache 2, et dans une édition commerciale exclusive. Les deux peuvent être utilisés pour le développement commercial.*

#### b. Visual Studio Code

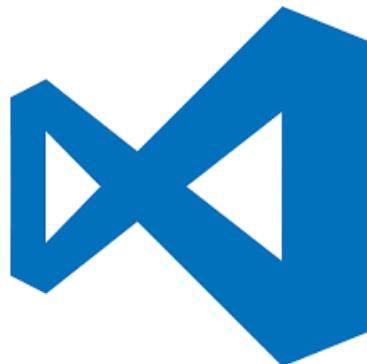


Figure 40: logo de Visual Studio Code

*Visual Studio Code est un éditeur de code source développé par Microsoft. Il inclut la prise en charge du débogage, du contrôle Git intégré, de la mise en évidence de la syntaxe, de l'achèvement de code intelligent, des extraits et du refactoring de code. Il est gratuit et open-source, bien que le téléchargement officiel soit sous licence exclusive.*

## 2. Front End

### a. ECMAScript 6



Figure 41: logo de ES6

*ECMAScript 6 plus connu sous le nom ES6, est la nouvelle version du langage JavaScript. Elle apporte un lot conséquent de nouveautés, de l'ajout de nouvelles fonctions à la bibliothèque standard dont les classes, les modules et bien plus encore.*

*ES6 a été pensé pour créer des applications web facilement maintenables, tout en restant compatibles avec le code existant. L'idée a été d'ajouter de nouvelles fonctionnalités au langage. Ainsi, la bibliothèque standard s'enrichit de nouvelles méthodes, mais surtout, le langage adopte de nouvelles syntaxes, permettant d'avoir du code beaucoup plus structuré et lisible.*

### b. Flow



Figure 42: logo de Flow

*Flow est un vérificateur de type statique open-source pour JavaScript. Il ajoute du typage statique à JavaScript pour améliorer la productivité des développeurs et la qualité du code. En particulier, le typage statique offre des avantages tels que la vérification précoce des erreurs, qui vous aide à éviter certains types de défaillances d'exécution et l'intelligence du code, ce qui facilite la maintenance, la navigation, la transformation et l'optimisation.*

### c. Eslint

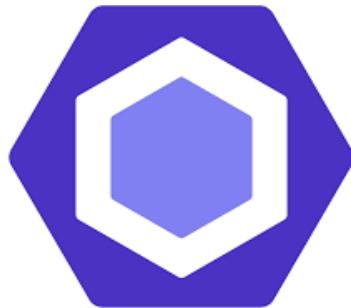


Figure 43: logo de Eslint

*ESLint est un utilitaire de code JavaScript open source créé en juin 2013. Le Linting de code est un type d'analyse statique fréquemment utilisé pour trouver des modèles problématiques ou du code qui ne respecte pas certaines directives de style. Il y a des linters de code pour la plupart des langages de programmation et les compilateurs intègrent parfois du Linting dans le processus de compilation.*

*JavaScript, étant un langage dynamique et mal typé, est particulièrement vulnérable aux erreurs de développeur. Sans le bénéfice d'un processus de compilation, le code JavaScript est généralement exécuté afin de trouver la syntaxe ou d'autres erreurs. Les outils de Linting comme ESLint permettent aux développeurs de découvrir des problèmes avec leur code JavaScript sans l'exécuter.*

### e. SaaS



Figure 44: logo de SaaS

*Sass est un langage de feuilles de style en cascade (CSS). C'est un langage de description qui est compilé en CSS. SassScript est un langage de script pouvant être utilisé à l'intérieur du code Sass. Deux syntaxes existent. La syntaxe originale, nommée « syntaxe indentée », est proche de Haml. La nouvelle syntaxe se nomme SCSS. Elle a un formalisme proche de CSS.*

### 3. Back End & Middleware

#### a. Java EE



Figure 45: logo de Java EE

Java EE est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants. On parle généralement de «plate-forme Java EE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution.

#### b. Tomcat

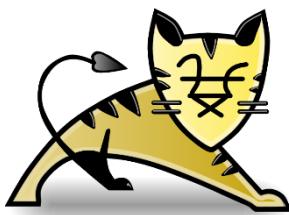


Figure 46: logo de Tomcat

Tomcat est un Serveur d'application Java, c'est à dire un programme tournant sur un serveur (Apache) qui gère d'autres servlets permettant de générer des pages web dynamiques. Tomcat a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant. Il est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion.

#### c. Gradle



Figure 47: logo de Gradle

Gradle est un moteur de production fonctionnant sur la plateforme Java. Il permet de construire des projets en Java, Scala, Groovy voire C++.

Gradle allie les atouts de Apache Maven et Apache Ant : il allie l'utilisation de conventions à la manière de Maven (convention plutôt que configuration) avec la flexibilité de Ant pour décrire les tâches de constructions, avec une cohérence forte dans l'interface de programmation des tâches.

#### d. Oracle



Figure 48: logo de Oracle

Oracle Database est un système de gestion de base de données relationnelle (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné entre autres, de Bob Miner et Ed Oates.

### 4. Outils de versioning

#### a. Git



Figure 49: logo de Git

Git est un système de contrôle de version pour suivre les changements dans les fichiers informatiques et coordonner le travail sur ces fichiers entre plusieurs personnes. Il est principalement utilisé pour la gestion de code source dans le développement de logiciels, mais il peut également être utilisé pour suivre les changements dans n'importe quel ensemble de fichiers. En tant que système de contrôle de révision réparti, il vise la vitesse, l'intégrité des données et la prise en charge des flux de travail distribués non linéaires.

## b. Gitlab



Figure 50: logo de Gitlab

Gitlab est une application de gestion de dépôt git libre. Il est un clone open source de Github. C'est un outil permettant à chacun de travailler à son rythme, de façon indépendante des autres collaborateurs, de s'échanger leurs travaux respectifs. Il existe réellement plusieurs dépôts pour un même projet.

## 5. CI/CD

### a. Docker

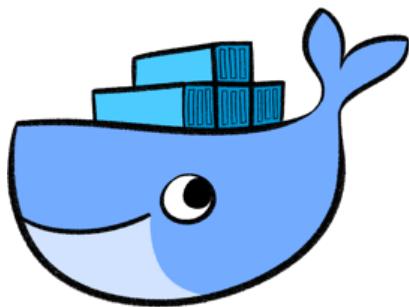


Figure 51: logo de Docker

Docker vous aide à créer et à déployer des logiciels dans des conteneurs. C'est une collection open source d'outils qui vous aident à «créer, expédier et exécuter n'importe quelle application, n'importe où».

Avec Docker, vous créez un fichier spécial appelé Dockerfile. Les Dockerfiles définissent un processus de construction qui, lorsqu'il est ajouté à la commande 'docker build', génère une image docker immuable. Vous pouvez considérer cela comme un instantané de votre application, prêt à être animé à tout moment. Lorsque vous voulez le démarrer, utilisez simplement la commande 'docker run' pour l'exécuter partout.

Docker fournit également un référentiel basé sur le cloud appelé Docker Hub. Vous pouvez y penser comme GitHub pour Docker Images. Vous pouvez utiliser Docker Hub pour stocker et distribuer les images conteneur que vous créez.

## b. Kubernetes



Figure 52: logo de Kubernetes

*Kubernetes est un système open source permettant d'automatiser le déploiement, la montée en charge et la gestion des applications conteneurisées, conçu à l'origine par Google. Il vise à fournir une plate-forme permettant d'automatiser le déploiement, la montée en charge et la mise en œuvre de conteneurs d'applications sur des clusters de serveurs. Il fonctionne avec toute une série de technologies de containers, et est souvent utilisé avec Docker.*

## 6. Les tests

### a. Mockito



Figure 53: logo de Mockito

*Mockito un framework Java très connu permettant de générer automatiquement des objets ‘mockés’. Couplé avec JUnit, il permet de tester le comportement des objets réels associés à un ou des objets ‘mockés’ facilitant ainsi l’écriture des tests unitaires.*

### b. Jest

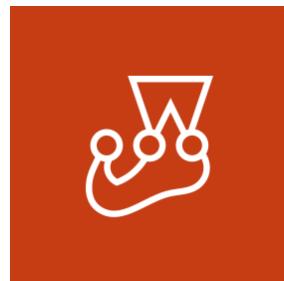


Figure 54: logo de Jest

*Jest est un excellent framework de test JavaScript développé par Facebook. Il est souvent utilisé pour tester les composants React, mais c'est aussi un bon framework de test général. Jest parallélise les tests effectués entre les travailleurs pour optimiser les performances. Les messages de la console sont mis en mémoire tampon et imprimés avec les résultats des tests.*

## II. Réalisation et mise en œuvre

### 1. Structure du projet

#### a. Structure du Frontend

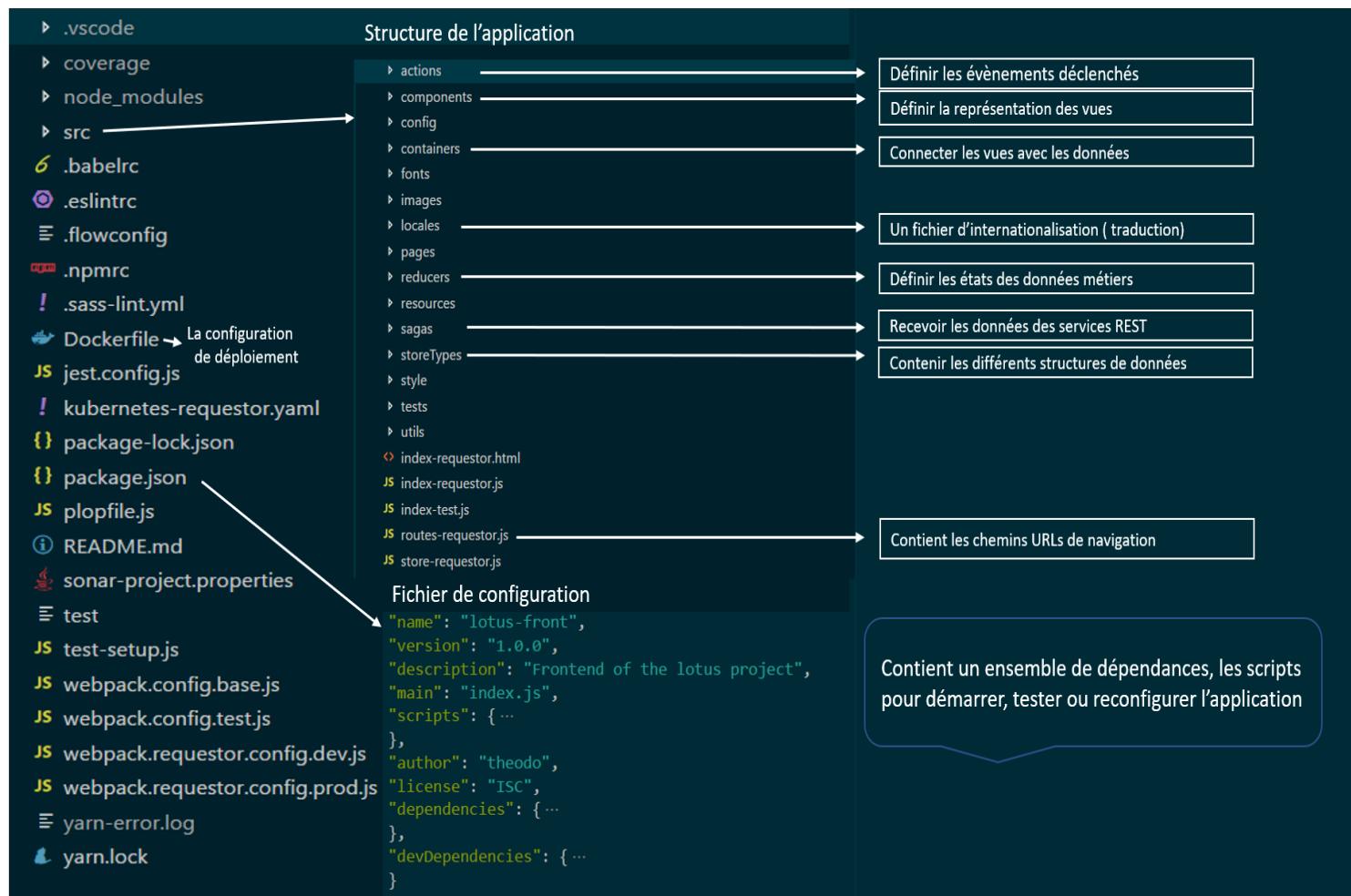


Figure 55: Structure du projet Frontend

## b. Structure Middleware

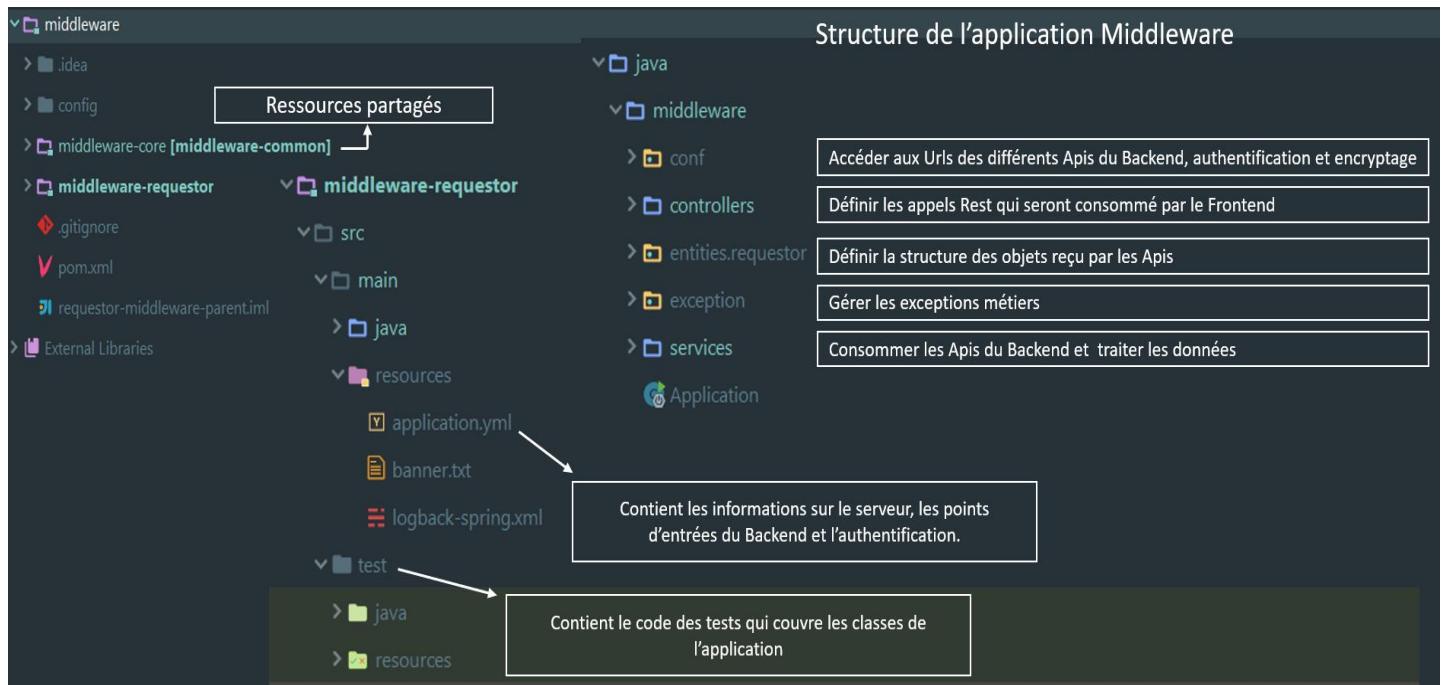


Figure 56: Structure du projet Middleware

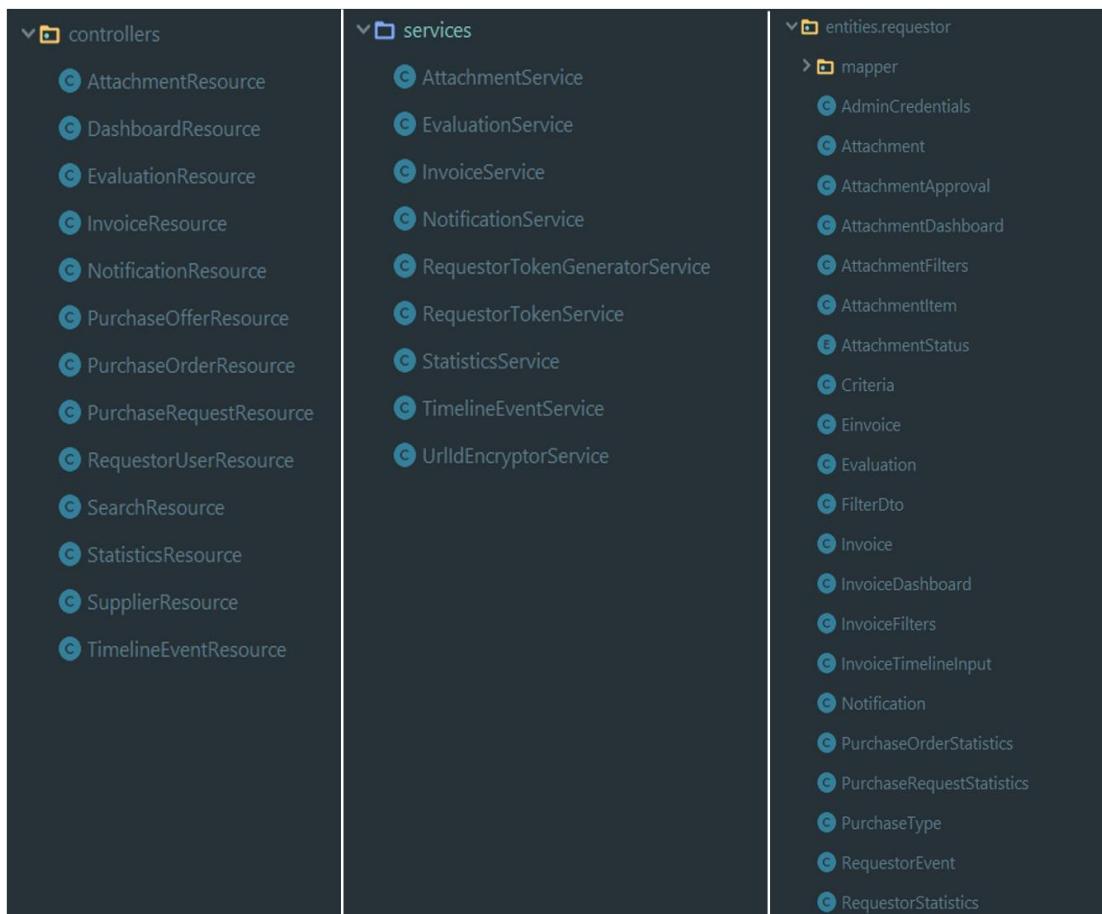


Figure 57: Classes du projet middleware

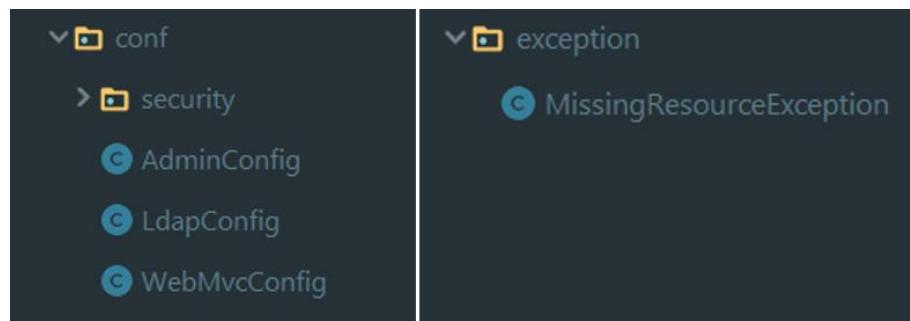


Figure 58: Package configuration middleware core

## c. Structure Backend

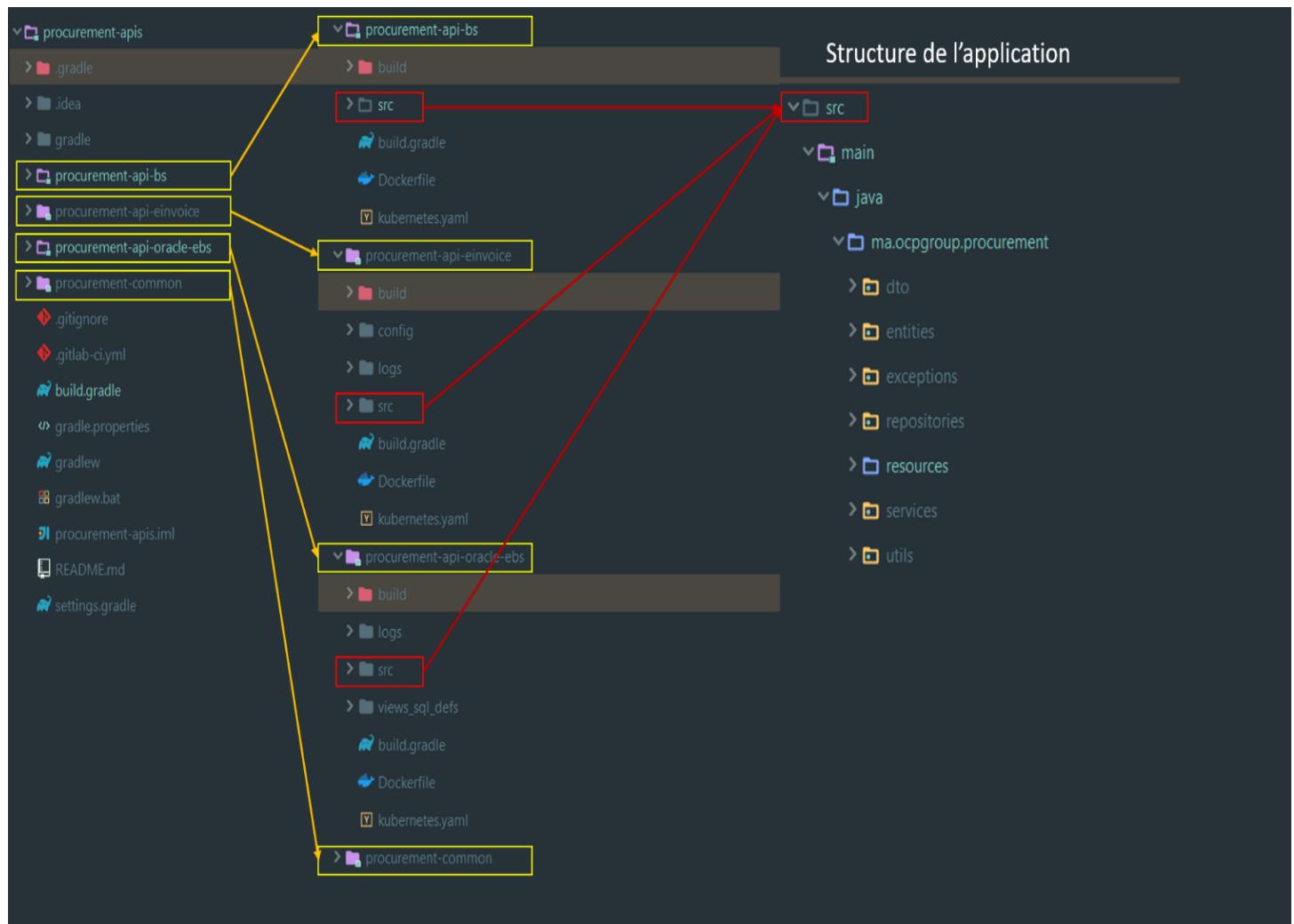


Figure 59: Structure du projet Backend

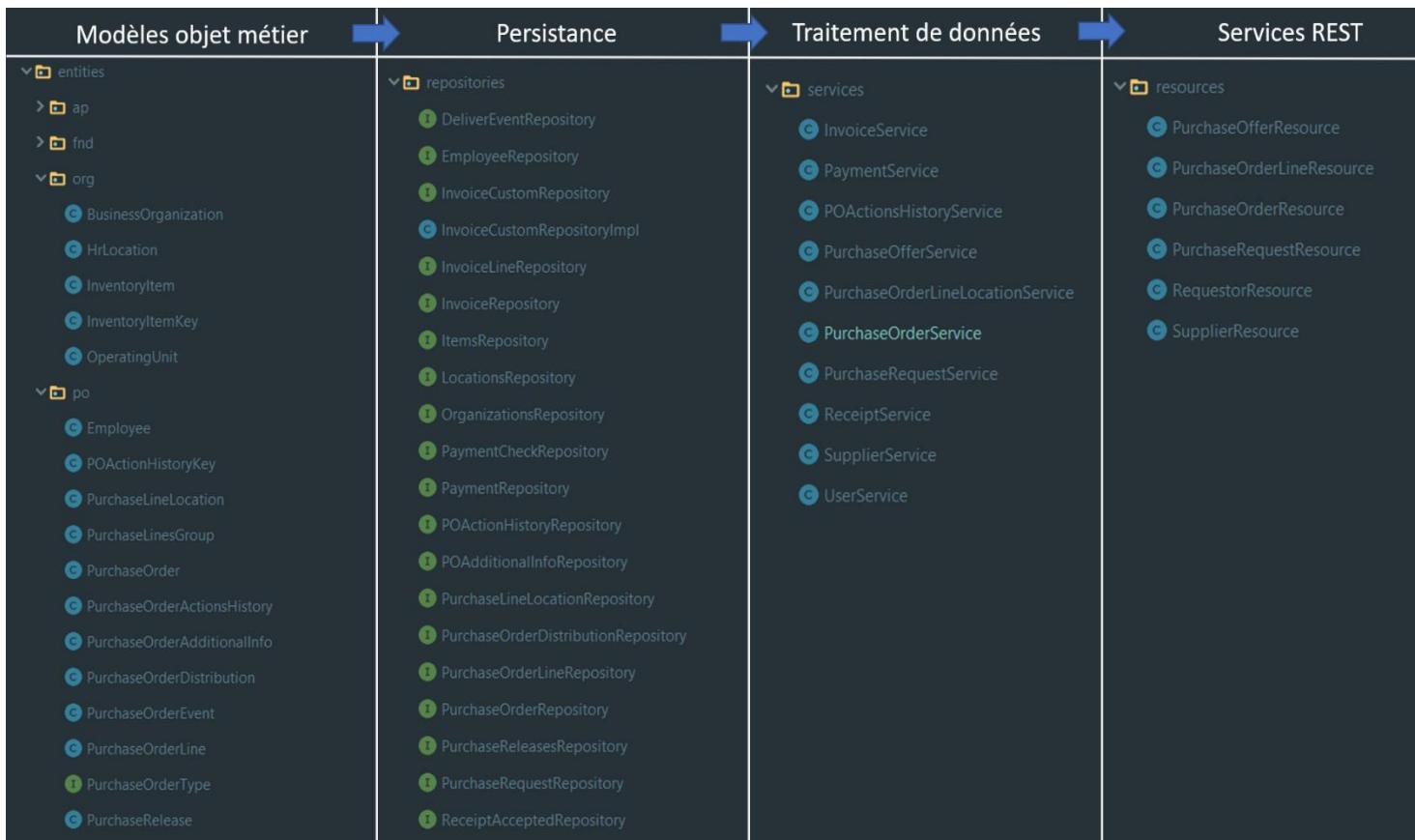


Figure 60: Structure de api oracle-ebs

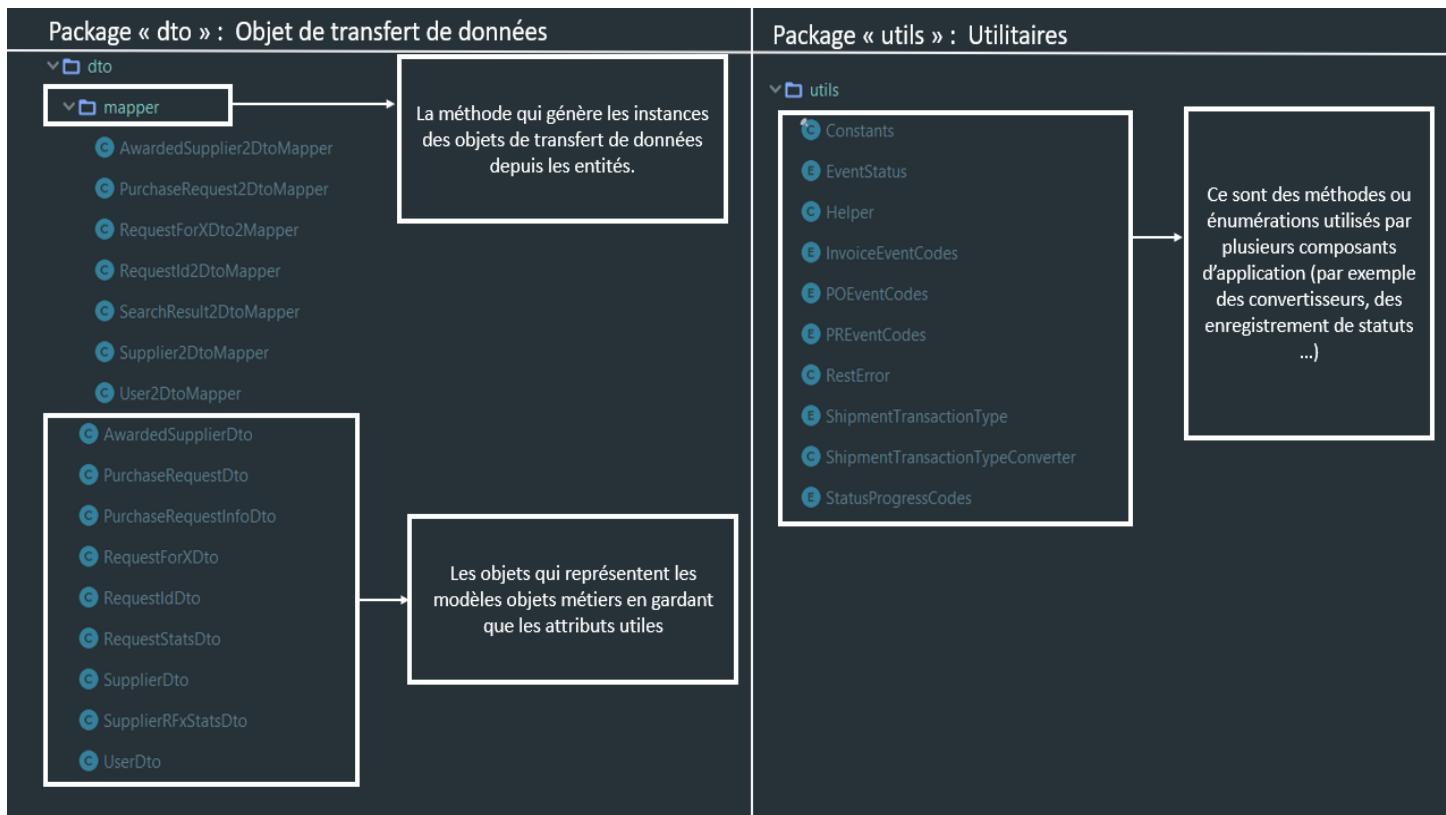


Figure 61: structure de api bravo

## 2. Charte graphique

On présente sur la figure ci-dessous, la structure qui a été adoptée pour toutes les pages de l'application.

En effet, toutes les pages de l'application sont divisées en trois sections :

- Section 1 : Le header qui contient le profil de l'utilisateur, la recherche et le menu lié à ce qui sera affiché.
- Section 2 : c'est le corps de l'application qui contient le contenu demandé par l'utilisateur.
- Section 3 : le footer qui contient des informations sur les droits d'auteur.

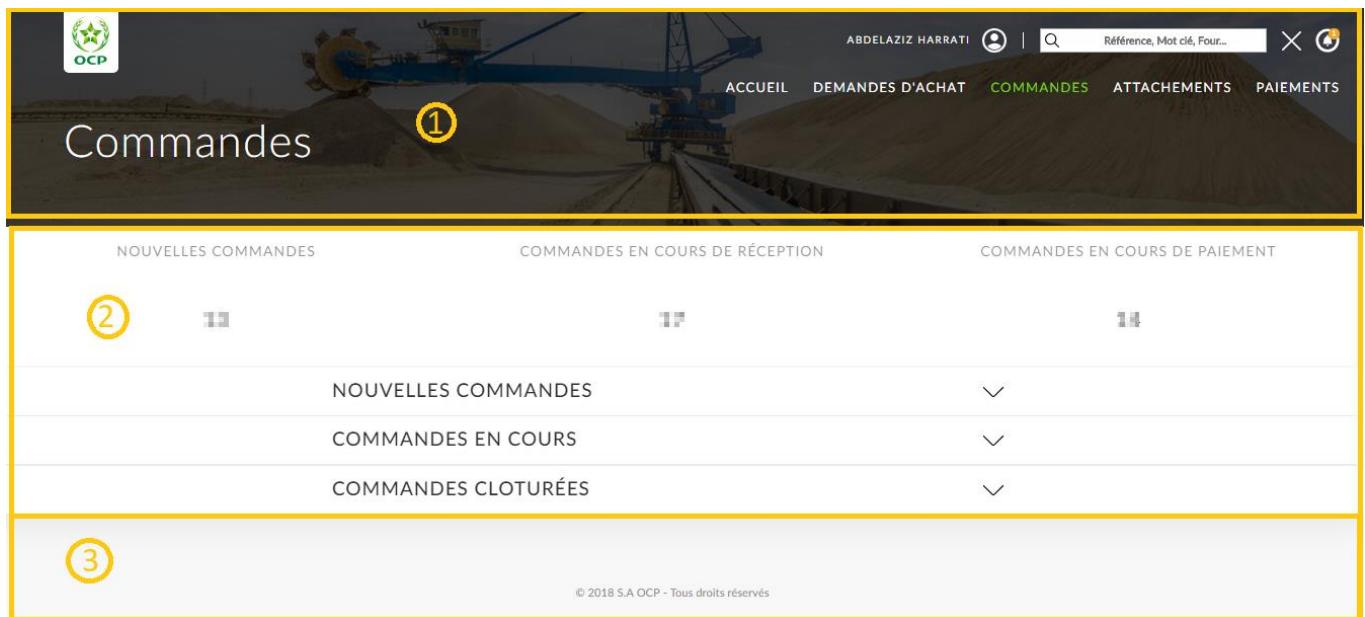


Figure 62: Structure des pages de l'application



Figure 63: page responsive

### 3. Captures d'écran

Dans cette partie on présente des exemples des IHM de l'application. Pour des raisons de confidentialité, des données ont été grisesées.

#### a. Authentification

L'authentification à l'application est assurée par le serveur d'annuaire LDAP qui fournit un token au chef de projet qui contient lui permet d'accéder à toute les pages de l'application.

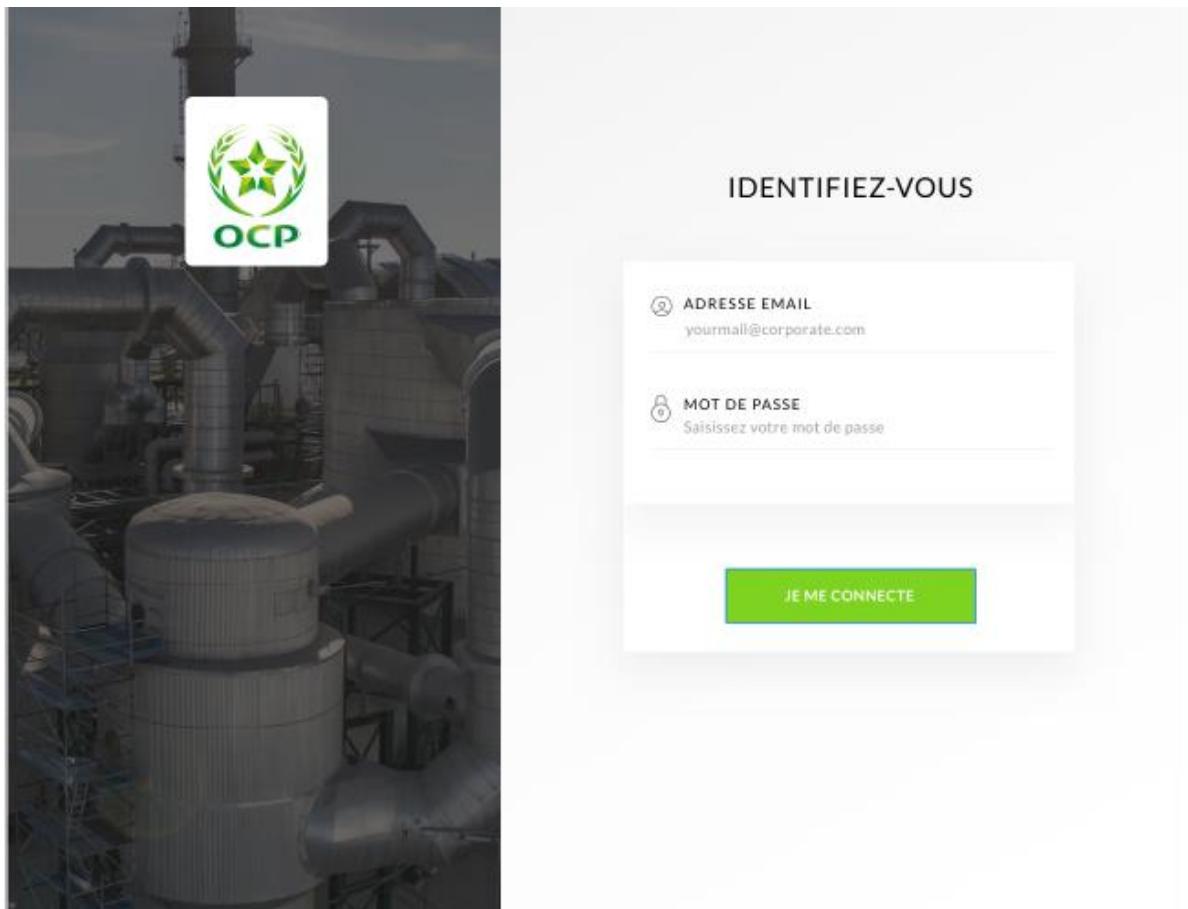


Figure 64: page d'authentification

#### b. Page des demandes d'achat

En choisissant le menu « Demandes d'achat », l'utilisateur est redirigé vers une page qui contient la liste de toutes ses demandes séparée par sections : demandes en préparation, demandes en attente d'approbation, demandes approuvées et demandes rejetées, ainsi qu'un tableau de bord avec trois indicateurs : Pourcentage des DA lancées, pourcentage des DA en phase commerciale et pourcentage des DA en phase technique.

Ci-dessous une capture de la page des demandes d'achat :

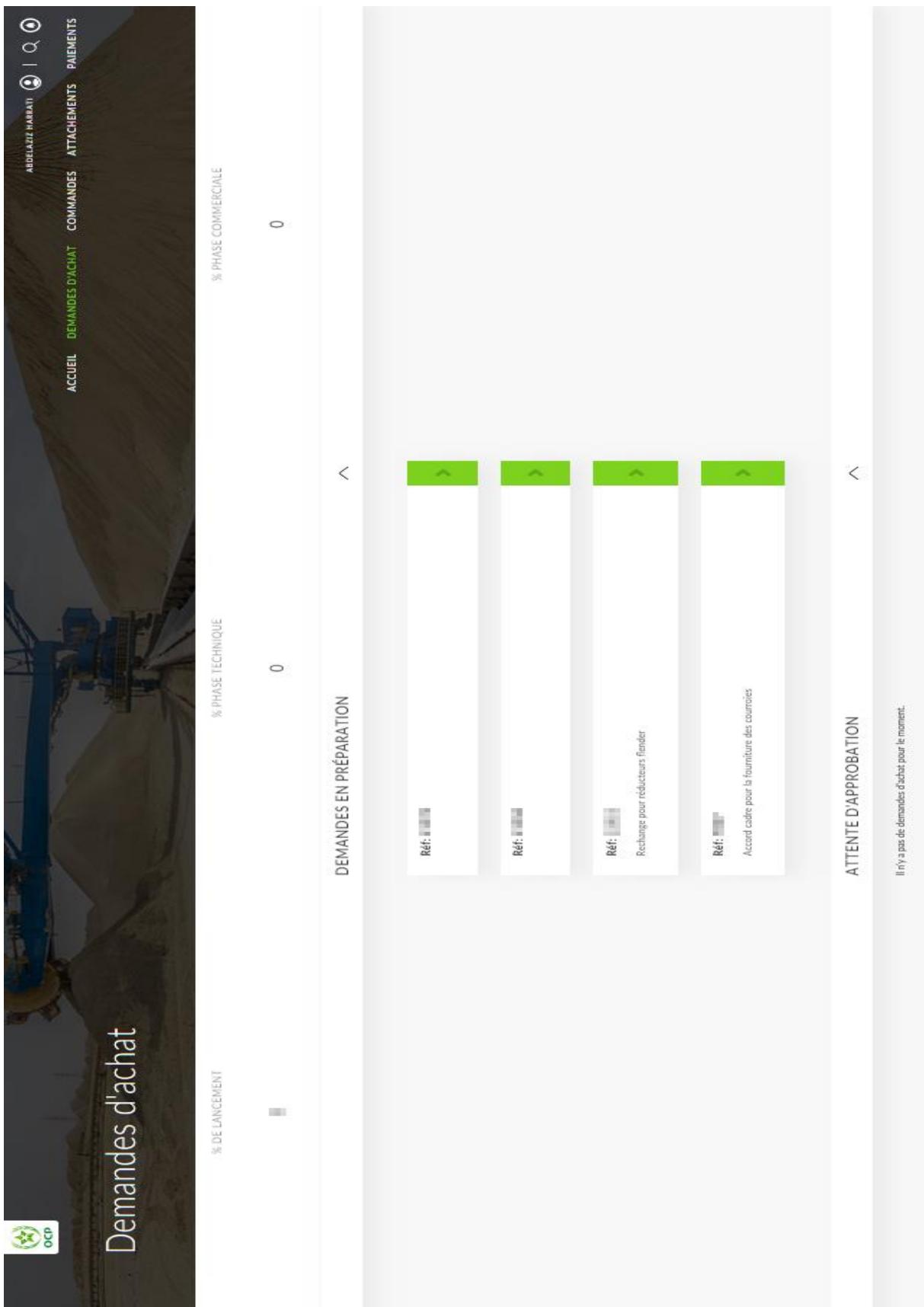


Figure 65: page des demandes d'achat

En cliquant sur une demande d'achat, on est redirigé vers la page de détail de la demande d'achat qui contient des informations sur les articles demandés à savoir le code de l'article, la quantité, le prix unitaire et la description :

The screenshot displays two pages of a web application. The left page is a dark-themed dashboard titled "Demandes d'achat" (Purchase Requests) with a background image of industrial cranes. It features a search bar with placeholder text "Référence, Mot clé, Fourn...". Below the search bar are navigation links: "ACCUEIL", "DEMANDES D'ACHAT", "COMMANDES", "ATTACHEMENTS", and "PAIEMENTS". A green button labeled "SUIVRE MA DA" (Follow my DA) is positioned on the right side. The right page is a detailed view of a specific purchase request. It shows a table with one row of data:

Code article :	Q'té : 1 PIÈCE	PU :
Détecteur de flamme uv / ir		

At the bottom right of the detail page, there is a small copyright notice: "© 2018 S.A.OCP - Tous droits réservés".

Figure 66: page détail des demandes d'achat

*En cliquant sur suivre ma DA dans la page de détail, on est dirigé vers la page de suivi de la demande qui contient une timeline avec différents événements liés à la DA :*

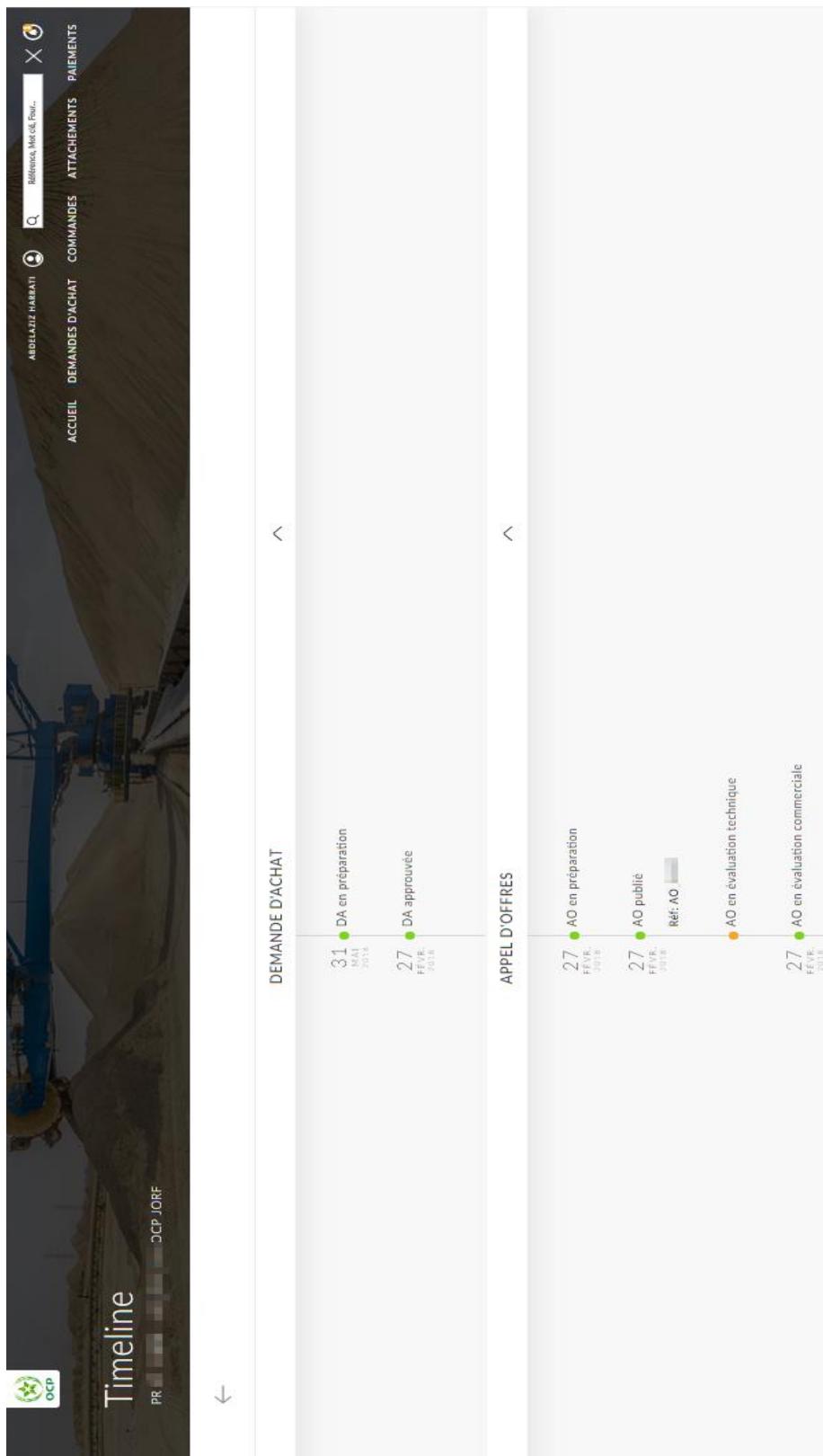


Figure 67: page suivi de la demande d'achat

## c. Page des commandes

*En choisissant le menu « Commandes », l'utilisateur est redirigé vers une page qui contient la liste de toutes ses commandes triées par sections : nouvelles commandes, commandes en cours, commandes rejetées, ainsi qu'un tableau de bord avec trois indicateurs : le nombre de nouvelles commandes, nombre des commandes en cours de réception et nombre des commandes en cours de paiement.*

Section	Description	Count
NOUVELLES COMMANDES	Livraison roulement	13
COMMANDES EN COURS	Avenir de réutilisation	17
COMMANDES EN REJET	Accessoires de roulements .	14

Figure 68: page des commandes

*En cliquant sur une commande, on est redirigé vers la page de détail de la commande qui contient des informations sur les articles demandés à savoir le code de l'article, la quantité, le prix unitaire, la date de la livraison prévue et la description:*

The screenshot shows the OCP Commandes application interface. On the left, there is a sidebar with the title "Commandes" and a logo. The main area displays a list of orders with columns for "Code article", "Qté", "Prix unit.", and "Date de livraison prévue". A green button labeled "SUIVRE MA COMMANDE" is visible. An arrow points from the list to a detailed view of order #10 PIECE, which includes fields for "Code article", "Qté", "Prix unit.", "Description", and "Date de livraison prévue". The detailed view also features a green "SUIVRE MA COMMANDE" button.

Code article	Qté	Prix unit.	Date de livraison prévue
PU_1 [REDACTED] MAD	10 PIECE	100,00 MAD	30/12/2015
PU_1 [REDACTED] MAD	5 PIECE	100,00 MAD	30/12/2015

Figure 69: Détail de la commande

*En cliquant sur suivre ma commande dans la page de détail, on est dirigé vers la page de suivi de la commande qui contient le profil du fournisseur et une timeline avec différents événements liés à la commande :*

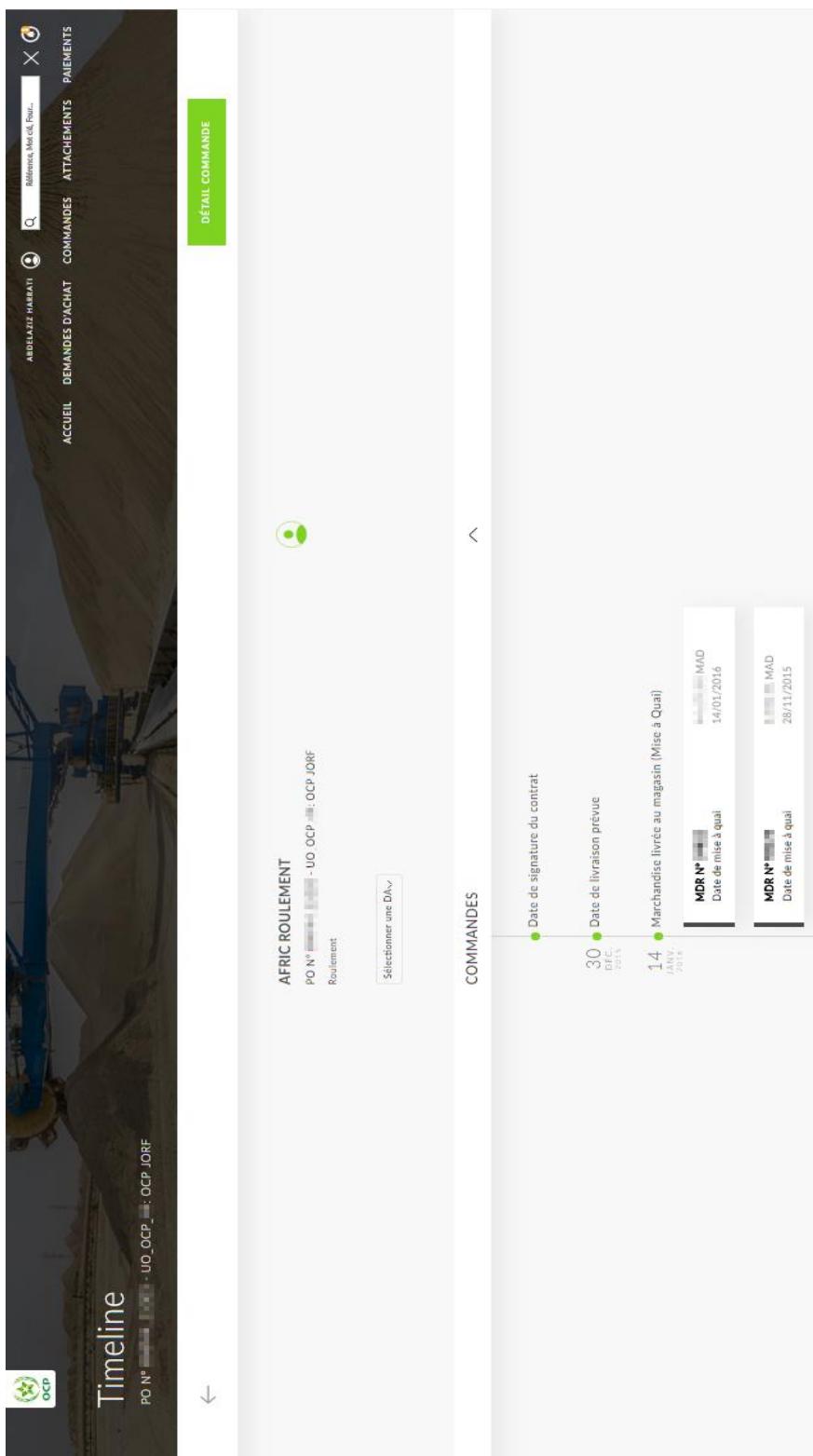


Figure 70: page suivi de la commande

## d. Page des attachments

*En choisissant le menu « Attachements », l'utilisateur est redirigé vers une page qui contient la liste de tous ses attachments triés par deux sections : nouveaux attachments et attachments traités, ainsi qu'un tableau de bord avec trois indicateurs : le montant et le nombre des attachments acceptés, le montant et le nombre des commandes acceptées et le montant et le nombre des attachments commandés.*

The screenshot shows a web application interface for managing attachments. At the top, there is a header bar with the OCP logo, user information (ABDELAZIZ HABIBI), and search/filter icons. Below the header, a navigation menu includes 'ACCUEIL', 'DEMANDES D'ACHAT', 'COMMANDES', 'ATTACHEMENTS', and 'PIEMENTS'. The main content area has two main sections:

- ATTACHEMENTS**: This section displays a table of attachments with columns: REF (reference number), DATE (date), STATUS (status), and TOTAL MAROC (total amount). There are three rows of data:
 

REF	DATE	STATUS	TOTAL MAROC
REF 1	DATE 30/04/2018	STATUS À traiter	TOTAL MAROC
REF 2	DATE 30/04/2018	STATUS À traiter	PROFINIK MAROC
REF 3	DATE 27/04/2018	STATUS Accepté	TOTAL MAROC
- COMMANDÉS (EN EXÉCUTION)**: This section displays a table of orders with columns: REF, DATE, STATUS, and TOTAL MAROC. There are three rows of data:
 

REF	DATE	STATUS	TOTAL MAROC
REF 1	DATE 30/04/2018	STATUS À traiter	TOTAL MAROC
REF 2	DATE 30/04/2018	STATUS À traiter	PROFINIK MAROC
REF 3	DATE 27/04/2018	STATUS Accepté	TOTAL MAROC

On the right side of the dashboard, there are three large green indicator boxes with the numbers 2, 3, and 5, representing different metrics. The background of the dashboard features a blurred image of industrial equipment.

Figure 71: page des attachments

*En cliquant sur un attachement, on est redirigé vers la page de détail de l'attachement, si la quantité attachée est inférieur à la quantité commandée, on pourra donc soit valider ou rejeter l'attachement sinon on est averti que la quantité attachée dépasse la quantité commandée et l'utilisateur ne pourra que rejeter l'attachement.*

The screenshot shows the 'Détail de l'attachement' (Attachment Detail) page. At the top, there is a header with the OCP logo, user information (ABDELAZIZ HARRATI), and navigation links (ACCUEIL, DEMANDES D'ACHAT, COMMANDES, ATTACHEMENTS, PAIEMENTS). Below the header, the main title 'Détail de l'attachement' is displayed, along with the date '27/04/2018' and a command reference number. On the left, there is a back arrow and a link to 'ARTICLES: 1'. On the right, there are two buttons: 'REJETER' (Reject) and 'APPROUVER' (Approve). The central content area is titled 'Articles de l'Attachement' and shows a table with columns 'REF' and 'PU'. A note below the table states: 'Mise à disposition d'un monteur en régie pour la réalisation des travaux de maintenance à EMAPHOS.' To the right of the table, there are numerical values: QTÉ\_REST 157 Unité, QTÉ\_ATT 20 Unité, and QTÉ\_CDE 200 Unité.

Figure 72: page de détail de l'attachement dans le cas normal

The screenshot shows the same 'Détail de l'attachement' page as Figure 72, but with a red warning banner at the top. The banner contains the text 'Attachement à rejeter' and 'La quantité attachée d'au moins un des articles dépasse la quantité de la commande!'. Below the banner, the page structure is identical to Figure 72, showing the 'ARTICLES: 1' link, the 'REJETER' button, and the detailed table of attached items.

Figure 73: page de détail de l'attachement dans le cas de dépassement

*Dans le cas de validation d'un attachement, l'utilisateur est amené à évaluer le fournisseur selon des critères qui diffèrent selon le type d'achat.*

AR345678 P.U. 18 000 MAD

Evaluation du fournisseur X

L'attachement à la référence de commande [REDACTED] sera approuvé, et le fournisseur SYDEN sera notifié.

Avant de continuer l'opération, veuillez évaluer ce fournisseur:

Respect des échéances de livraison	<span style="color: orange;">★ ★ ★ ★ ★</span>
Conformité des équipements réceptionnés aux exigences	<span style="color: orange;">★ ★ ★ ★ ★</span>
Qualité des moyens techniques déployés pendant la livraison et la mise en service (mise en service, formation, support technique,...)	<span style="color: orange;">★ ★ ★ ★ ★</span>

Annuler Confirmer

Figure 74: page d'évaluation du fournisseur

*Si le chef de projet choisit de rejeter l'attachement, il est mené à préciser la raison du rejet :*

MA JETE IP2 É\_CDE

Motif du rejet X

Vous avez rejeté l'attachement soumis par le fournisseur TOTAL MAROC. Nous vous invitons à justifier la raison de votre rejet.

Non conformité du budget  
 Autre

Veuillez renseigner la raison de votre rejet.

ANNULER CONFIRMER

Figure 75: page de rejet d'un attachement

## e. Page de paiement

*En choisissant le menu « Paiement », l'utilisateur est redirigé vers une page qui contient la liste de toutes ses factures, ainsi qu'un tableau de bord avec six indicateurs clés : le montant et le nombre échues, le montant et le nombre non échues, le montant et le nombre reçues, le montant et le nombre en cours de paiement, nombre réglées.*

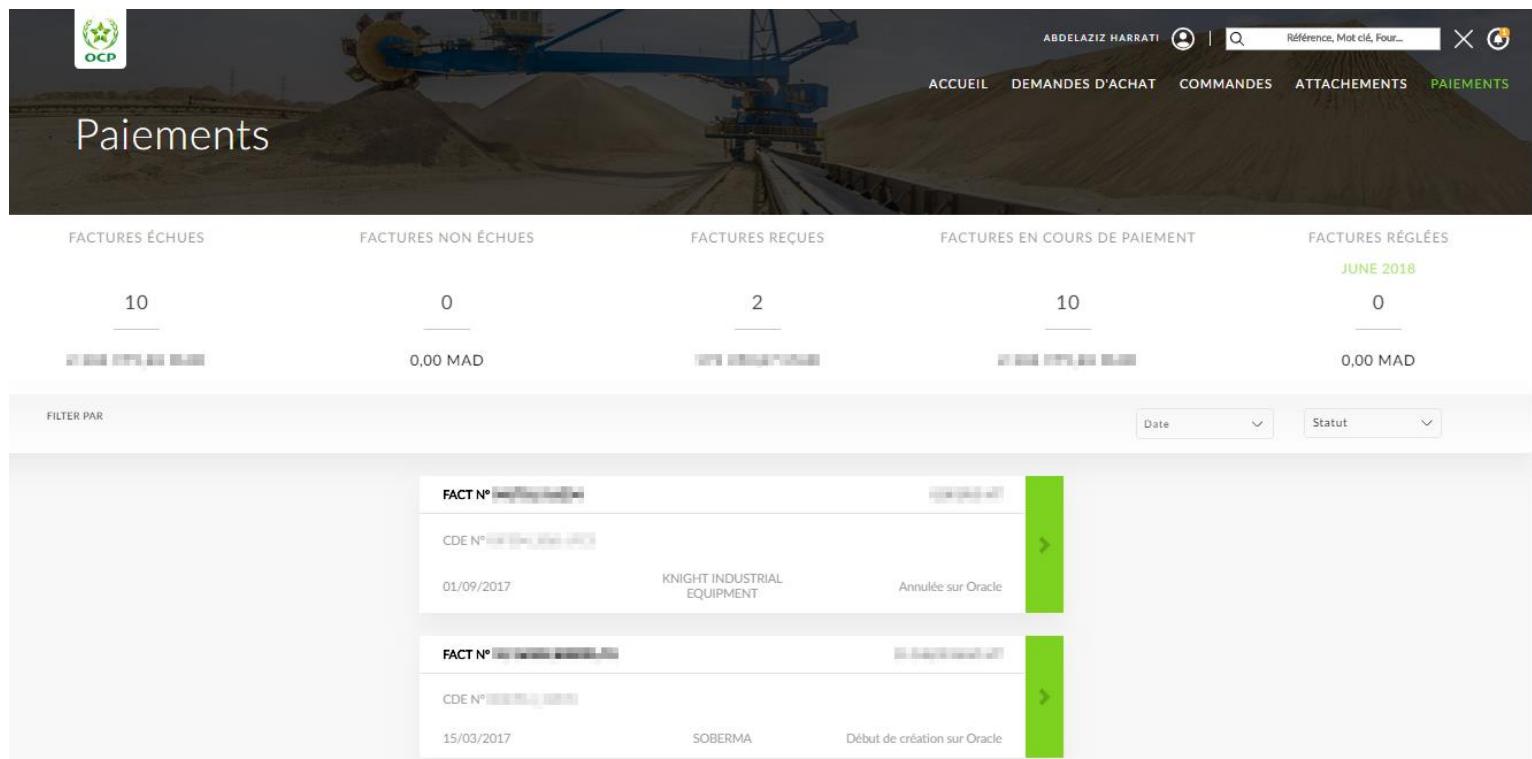


Figure 76: page de paiement

*En cliquant sur une facture, on est redirigé vers la page de détail de suivi de la facture, qui contient une timeline avec différents événements liés à la facture :*

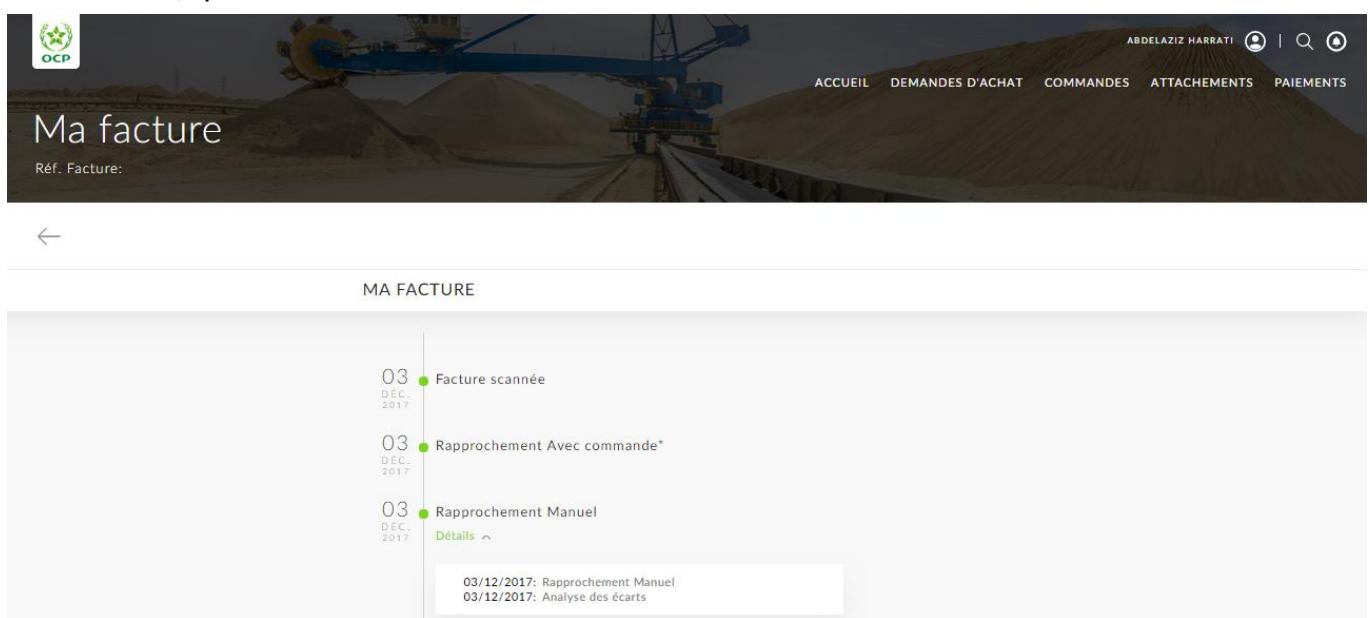


Figure 77: page détail de la facture

## f. Moteur de recherche

L'utilisateur peut accéder au moteur de recherche à n'importe quel moment en cliquant sur la loupe de recherche en haut du header de la page, et peut taper un mot clé et le moteur retournera les résultats adéquats dans les pages demandes d'achat, commandes, attachements et paiement. En cliquant sur un résultat de la recherche, on est redirigé vers la page détail correspondante.

The screenshot shows a search interface with a search bar containing 'pompe'. Below the search bar, there is a button labeled 'X' and a green search icon. A grey banner indicates '8 RÉSULTATS'. The results are organized into three sections:

- DEMANDES D'ACHAT**
  - Réf. : XXXXXXXXXXXX  
Pompe centrifuge multi-étages orem ipsum dolor sit amet, consectetur pompe
  - Réf. : XXXXXXXX  
Pompe centrifuge multi-étages orem ipsum dolor sit amet
  - Réf. : XXXXXX  
Pompe centrifuge multi-étages
- COMMANDES**
  - Réf. : XXXXXXXXX  
Pompe centrifuge multi-étages orem pompe
  - Réf. : XXXXXXXXXXXX  
Pompe centrifuge multi-étages consectetur adipiscing elit, sed
- DEMANDES D'ACHAT**
  - Réf. : XXXX  
Pompe centrifuge multi-étages orem ipsum dolor sit amet, consectetur adipiscing elit, sed pompe
- FOURNISSEURS**
  - Réf. : XXXXXXXXX  
Pompe centrifuge multi-étages

Figure 78: Moteur de recherche

## e. Notifications

Le chef de projet reçoit des notifications à chaque fois qu'un attachement a été créé par un fournisseur, il peut accéder au détail de l'attachement en cliquant sur la notification où il pourra soit valider ou rejeter l'attachement. L'ordre d'affichage des notifications est chronologique, et une fois elle est lue sa couleur devient gris et atterrit en bas de la liste.

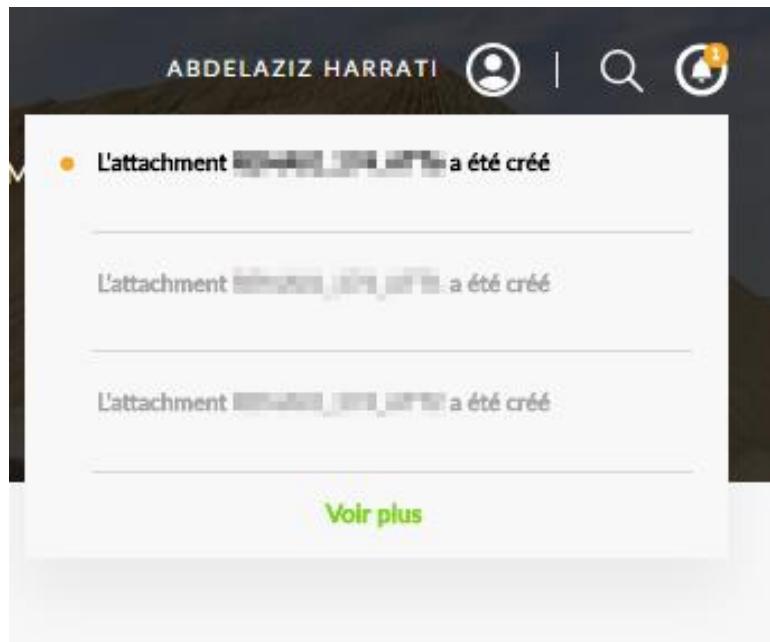


Figure 79: Notifications

## Conclusion

Ce projet de fin d'études a réellement été le fruit d'une expérience très enrichissante. En effet, ce fut l'occasion pour nous de travailler sur un projet réel qui nous a permis de mettre en pratique la méthode agile SCRUM et l'approche technique DevOps en utilisant des technologies récentes telles que ReactJS et Spring boot. C'était aussi une opportunité de discerner un métier nouveau, celui de l'approvisionnement au sein d'un organisme aussi bien connu comme l'OCP SA.

Le projet consiste en la contribution au développement d'un portail web pour le compte des chefs de projets au sein du groupe OCP SA qui permet d'avoir plus de transparence sur le processus d'approvisionnement. Il s'agit de mener une étude avec analyse, conception et réalisation dans le but de fournir au client une application web particulièrement adaptée à ses besoins.

Ce grand projet a été lancé dans le cadre de la transformation digitale menée au sein du groupe OCP SA, afin d'optimiser les relations de ce dernier avec ses fournisseurs et de donner plus de visibilité sur l'ensemble des opérations liées au processus « Source To Pay ».

C'est dans cette perspective que nous avons rejoint une équipe de développement au sein de la Digital Factory OCP, pour collaborer au développement de l'application de manière professionnelle avec toutes les parties prenantes du projet.

Dès le début, Nous avons effectué des formations tant bien dans les technologies avec lesquelles nous devons travailler que dans la méthode SCRUM et l'approche DevOps. Nous avons été amenés par la suite à comprendre au maximum le processus métier malgré sa complexité.

Nous avons commencé par donner les spécifications fonctionnelles et techniques de notre application en traduisant les besoins fonctionnels en diagrammes de packages, cas d'utilisation et scénarii et en présentant les besoins techniques sur une architecture microservices utilisant les frameworks Spring Boot, React JS, Redux, ... et les design patterns DTO, IOC, ...

L'analyse du projet consiste surtout à déterminer les classes candidates. La conception définit les procédures de test correspondantes aux scénarii déjà produits à travers des diagrammes de séquence.

Le projet est planifié en des sprints de deux semaines. Chaque sprint commence par une réunion qui permet de planifier ce qui doit être réalisé durant le sprint et s'achève sur la démonstration de la réalisation et une rétrospective pour discuter les points forts et les erreurs à éviter dans les prochains sprints. L'approche DevOps nous a permis de procéder à deux types de déploiement de l'application d'une part dans un environnement de test et développement accessible à l'équipe de projet et au product owner et d'autre part dans un environnement de production accessible aux utilisateurs de l'application à savoir les chefs de projet.

*Les objectifs planifiés lors de notre période de stage concernant la réalisation du portail ont été atteints et le product owner est tout à fait satisfait du travail réalisé. Le produit est prêt à être lancé en production en attente des feedbacks des utilisateurs qui vont générer d'autres besoins à réaliser dans l'application.*

*Il faut dire que la proximité du product owner nous a été très utile pour la réalisation du projet. La méthode agile SCRUM suivie pour la gestion de ce projet s'est avérée la mieux adaptée à son contexte surtout que, au début, le client n'avait pas une vision claire sur la solution (beaucoup de besoins ont été exprimés au cours du développement de la solution).*

*L'architecture microservices, facilement maintenable, a également grandement contribué dans la rapidité de la réalisation du projet et la satisfaction des besoins du client. Elle nous a permis de réduire considérablement l'effort de développement.*

*Nous avons rencontré certaines difficultés durant le projet, à savoir les dépendances externes avec d'autres entités conduisant à des retards incontournables. Nous avons aussi rencontré des problèmes d'accès à la base de données que nous avons résolus temporairement en effectuant les tests de requêtes dans les machines de nos collègues bénéficiant de cet accès.*

*Comme il s'agit d'un projet de grande envergure, de nouveaux modules seront intégrés comme extension à ce projet par d'autres collaborateurs.*

*Finalement et comme perspective importante, ce projet est conçu et développé d'une manière générique pour qu'il soit facilement adaptable aux besoins de tous les chefs de projets de l'organisme d'accueil au niveau national et continental.*

## Webographie

- <http://www.ocpgroup.ma/fr/group/financial-information>
- <http://www.ocpgroup.ma/fr/investors/governance/committees>
- <https://fr.atlassian.com/devops>
- <http://www2.fii.tstuba.sk/~bielik/courses/msi-slov/reporty/pmbok.pdf>
- <https://www.negotiations.com/articles/procurement-terms/>
- <http://igm.univ-mlv.fr/~dr/XPOSE2008/SCRUM/presentation.php>
- <https://blog.containerization.io/k8svsdocker>
- <https://www.docdoku.com/blog/2015/03/23/tests-unitaires-avec-mockito/>
- [https://fr.wikipedia.org/wiki/Sass\\_\(langage\)](https://fr.wikipedia.org/wiki/Sass_(langage))
- <http://www.tutorialspoint.com/spring/>
- <https://www.jetbrains.com/idea/documentation/>
- <https://spring.io/guides/gs/gradle/>
- <https://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>
- <https://flow.org/en/docs/>
- <https://eslint.org/docs/>
- <https://facebook.github.io/jest/>

## Annexes

---

*Dans ce qui suit seront présentés les documents annexes proposant plus de détails sur quelques concepts et définitions utilisés pour concevoir et réaliser ce projet.*

## Annexe 1 : SCRUM

### 1. Les principes d'une méthode agile

*Les 12 principes d'une méthode agile :*

- *Satisfaire le client est la priorité.*
- *Accueillir les demandes de changement « à bras ouverts ».*
- *Livrer le plus souvent possible des versions opérationnelles de l'application.*
- *Assurer une coopération permanente entre Client et Equipe projet.*
- *Construire des projets autour d'individus motivés.*
- *Privilégier la conversation en face à face.*
- *Mesurer l'avancement du projet en termes de fonctionnalités de l'application.*
- *Faire avancer le projet à un rythme soutenable et constant.*
- *Porter une attention continue à l'excellence technique et à la conception.*
- *Favoriser la simplicité.*
- *Responsabiliser les équipes.*
- *Ajuster, à intervalles réguliers, son comportement, ses processus pour être plus efficace.*

### 2. Au quotidien

*Chaque journée de travail commence par une réunion de 15 minutes maximum appelée mêlée quotidienne (Daily Scrum).*

*Seuls l'équipe, le directeur de produit et le Scrum Master peuvent parler, tous les autres peuvent écouter mais pas intervenir (leur présence n'est pas obligatoire).*

*A tour de rôle, chaque membre répond à 3 questions :*

- *Qu'est-ce que j'ai fait hier ?*
- *Qu'est-ce que je compte faire aujourd'hui ?*
- *Quelles sont les difficultés que je rencontre ?*

*Le tour de parole doit être scrupuleusement respecté pour éviter que le Scrum dérive sur des discussions techniques et déborde des 15 minutes. Si le besoin s'en fait sentir, des discussions sont alors menées librement après le Scrum.*

*Cette réunion a un but de synchronisation pour l'équipe et ne doit pas être vécue comme un Reporting d'activité.*

### 3. Les sprints

*À la fin du sprint, tout le monde se réunit pour effectuer la revue de sprint, qui dure au maximum 4 heures.*

*L'objectif de la revue de sprint est de valider le logiciel qui a été produit pendant le sprint. Le responsable du produit commence par rappeler le Sprint précédent et ensuite d'énoncer le plan de la revue.*

*Une présentation puis une démonstration du produit réalisé est entamée. C'est sur la base de cette démonstration que le directeur de produit valide chaque fonctionnalité planifiée pour ce sprint.*

*Une fois le bilan du sprint réalisé, l'équipe et le directeur de produit proposent des aménagements sur les Backlogs du produit et sur la planification provisoire de la release.*

*Il est probable qu'à ce moment, des items soient ajoutés, modifiés ou ré estimés, en conséquence de ce qui a été découvert.*

## 4. Cycle de vie logiciel

*Un cycle de vie d'un logiciel est un ordonnancement des différentes étapes du processus de développement. Un cycle de vie définit les étapes du processus, leur ordonnancement ainsi que les critères de passage d'une étape à une autre.*

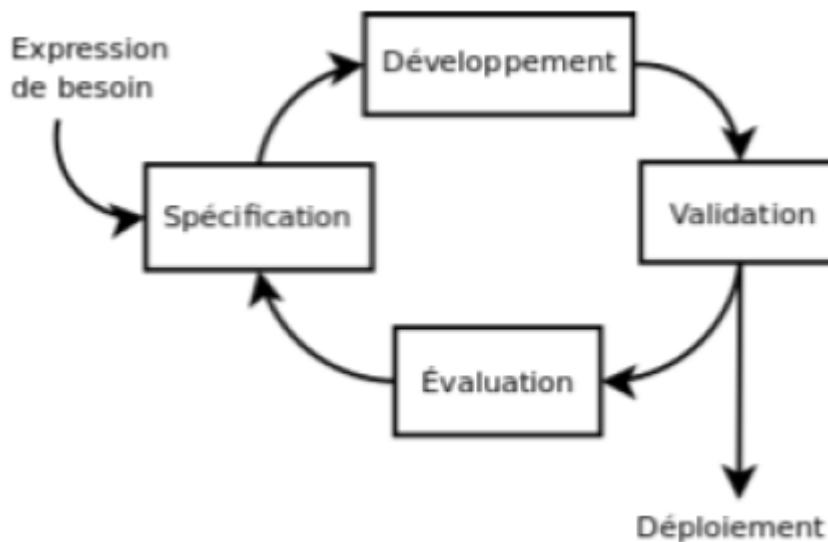


Figure 80 : Cycle de vie logiciel

*Puisque SCRUM exige l'organisation de la réalisation du projet en Sprints, chaque sprint sera représenté par une itération de 2 semaines concernera des charges à réaliser et aura comme produit une nouvelle version de l'application contenant de nouvelles fonctionnalités.*

## 5. Spécification du Backlog

Au démarrage, le Scrum master met en place le Backlog du projet qui est une liste des fonctionnalités attendues du produit final (identification des Key-users, création des User stories, cas de tests) ses fonctionnalités sont classés par priorité par le chef de projet et répartis par les sprints de projet. Ce qui permet de définir l'ordre de réalisation.

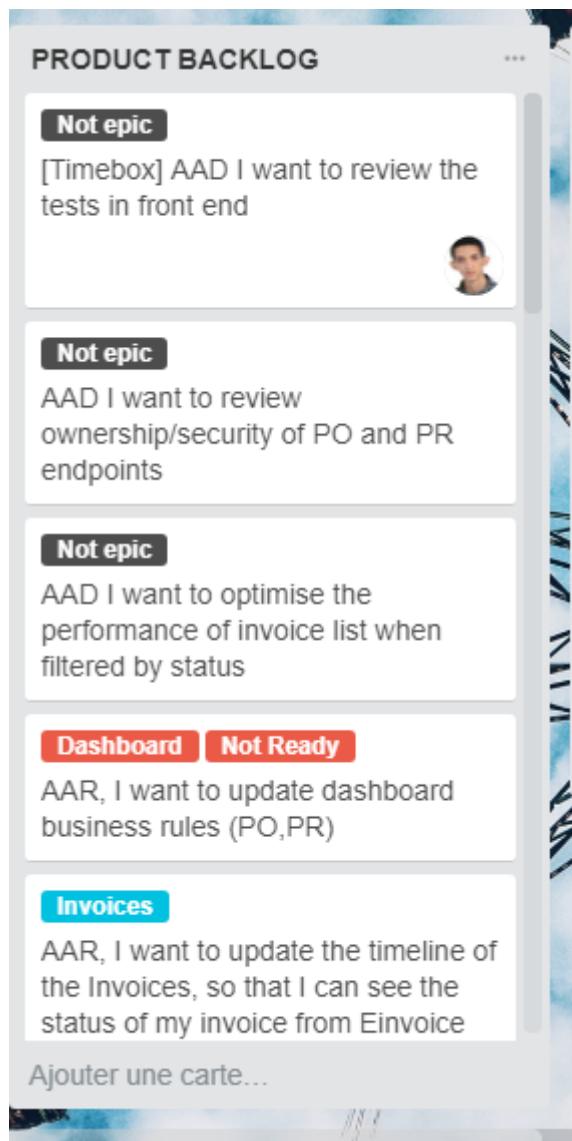


Figure 81: Backlog Trello

## 6. Avant le sprint planning

Au début de chaque sprint le chef de projet et le Scrum master se réunissent pour discuter les fonctions à produire et les user stories les plus importantes qui doivent obligatoirement être réalisées au cours du sprint, par la suite le chef de projet valide les

développements nécessaires à faire avec son équipe de projet, le Scrum master, essaye de son tour de traduire ces fonctionnalités en des interfaces faciles à utiliser, c'est-à-dire la réalisation de maquettes.

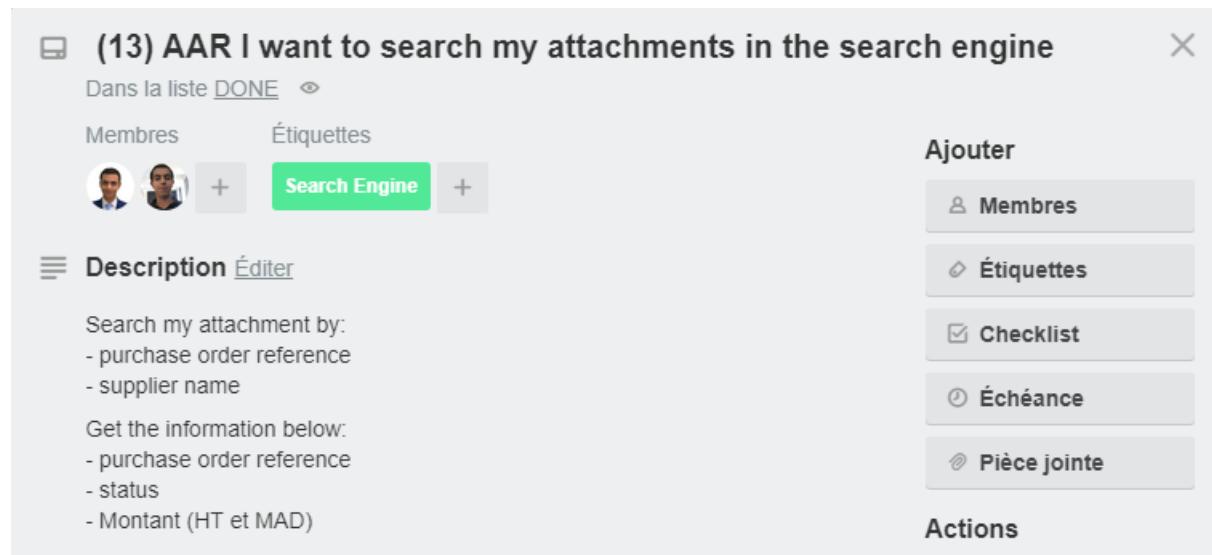
Au cas où il y a une complexité sur un point mal traité, l'équipe de projet consulte à tout moment le Scrum Master pour que l'interface soit à l'image exacte que le client l'a voulue.

## 7. Durant le sprint planning

Au début de chaque sprint nous nous réunissons avec la présence du chef de projet et toute l'équipe de développement pour établir le sprint planning. C'est une réunion où nous estimons, en points de complexité ou d'effort, les tâches à réaliser pendant le sprint.

Ensuite nous parcourons chaque histoire (user story : élément du Backlog) et nous ajoutons des sous tâches pour chaque histoire, chaque sous tâche correspondant à une tâche technique de cette histoire et estime le temps qu'il faudra pour la réaliser, en commençant par la plus importante.

Pour pouvoir fournir une estimation globale du sprint. Chaque fois qu'une tâche doit être estimée (en degré de sa difficulté et non pas en nombre de jours de travail), chacun de nous l'équipe représente son estimation de temps. De cette manière chacun de nous est forcé à réfléchir par lui-même au lieu de s'appuyer sur l'estimation de quelqu'un d'autre. S'il y a un gros écart entre deux estimations, nous discutons les différences et nous tentons d'élaborer une vision commune du travail impliquée par l'histoire. Ils peuvent même faire une sorte de décomposition en tâches. Après, nous estimons de nouveau. Cette boucle est répétée jusqu'à ce que les estimations convergent, c'est-à-dire que toutes les estimations soient approximativement les mêmes pour cette histoire.



**(13) AAR I want to search my attachments in the search engine**

Dans la liste DONE

Membres Étiquettes

+ +

**Description** Éditer

Search my attachment by:

- purchase order reference
- supplier name

Get the information below:

- purchase order reference
- status
- Montant (HT et MAD)

**Ajouter**

- Membres
- Étiquettes
- Checklist
- Échéance
- Pièce jointe

**Actions**

Figure 82: Ticket Trello

## 8. Le stand up

*Pour la mêlée quotidienne nous nous mettons d'accord pour le faire chaque matinée à 9h45 de durée de 15min à 20 min. chacun de nous répond à la première question du Scrum : Qu'as-tu fais hier ? Il s'agit, pour chaque membre de l'équipe, de parler des tâches sur lesquelles il a travaillé. Pour chacune de ces tâches, il va indiquer si la tâche est en cours ou si elle est finie. Ensuite chaque participant répond à la deuxième question du Scrum : Que prévois-tu de faire jusqu'à aujourd'hui ? Il s'agit de parler des tâches sur lesquelles il prévoit de travailler. Pour chacune de ces tâches, il va indiquer s'il estime que la tâche sera finie ou pas de la prochaine mêlée. En dernier lieu chacun de nous répond à la troisième question du Scrum : Qu'est-ce qui te gêne pour réaliser ton travail aussi efficacement que possible ? L'objectif de cette question est l'identification des problèmes.*

## Annexe 2 : Présentation du processus « S2P »

### 1. Sourcing: phase AMI/RFI

Comme son nom l'indique, le département des achats utilise les RFI pour recueillir des informations afin de décider la prochaine étape avant de s'engager dans des négociations.

Une RFI est une sollicitation envoyée à une large base de fournisseurs potentiels pour conditionner, recueillir des renseignements, préparer une demande de prix, élaborer une stratégie ou créer une base de données qui sera utile aux négociations ultérieures avec les fournisseurs sur:

- Les fournisseurs, y compris: les installations, les finances, les attitudes et les motivations.
- L'état du marché de l'offre.
- Tendances et facteurs de changement.

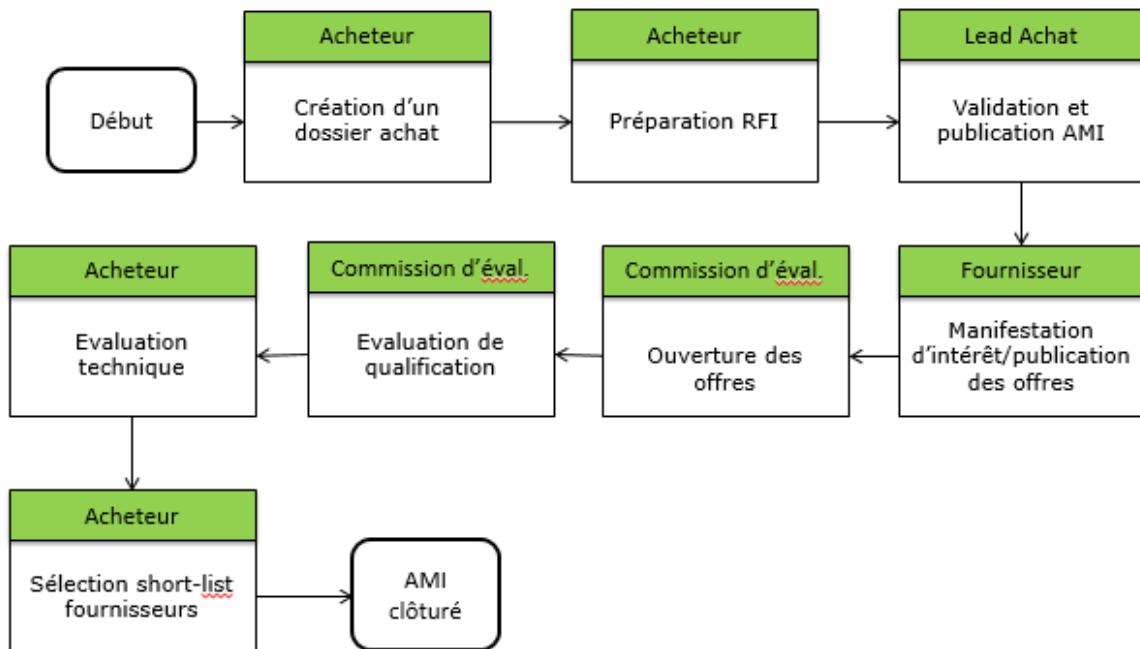


Figure 83: Processus AMI/RFI

### 3. Sourcing: phase AO

*Un appel d'offre est une procédure qui permet à un commanditaire (le maître d'ouvrage), de faire le choix de l'entreprise (le soumissionnaire qui sera le fournisseur) la plus à même de réaliser une prestation de travaux, fournitures ou services. Le but est de mettre plusieurs entreprises en concurrence pour fournir un produit ou un service.*

*Les appels d'offres sont devenus une pratique courante de tous les processus d'achat pour des ventes dès lors que leur montant est significatif. Ils traduisent le poids des directions des achats et l'intensité de la concurrence.*

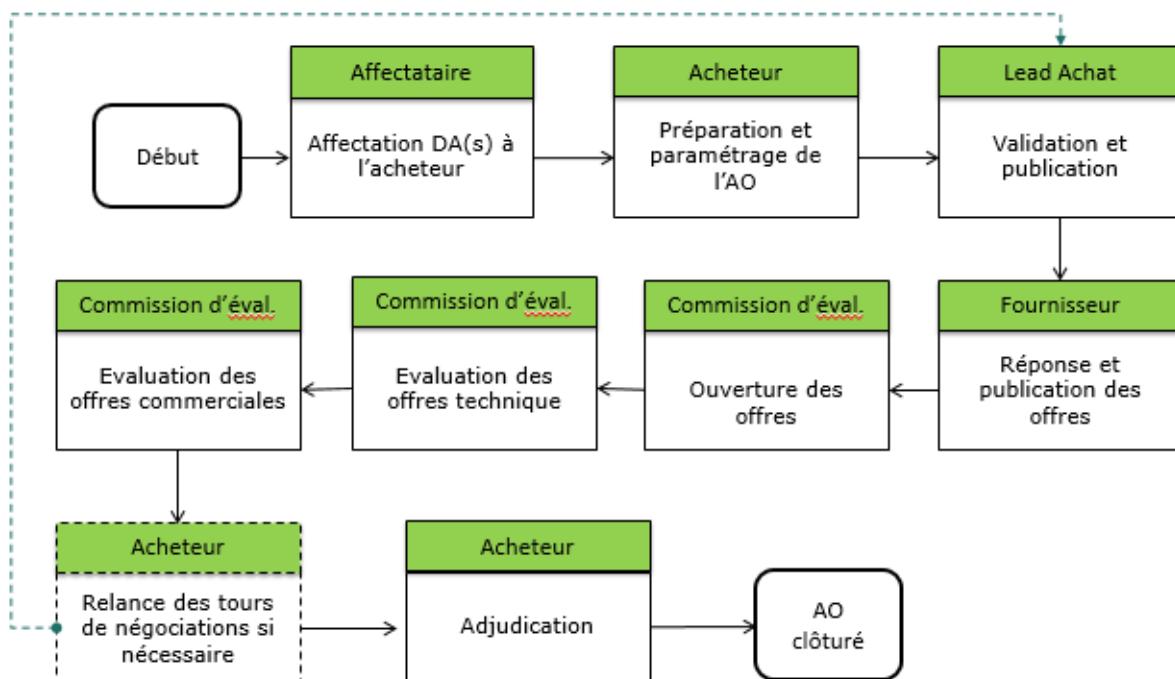


Figure 84: processus AO

### 4. Achat par commande standard

*Une commande standard est le plus basique et le plus utilisé parmi les différents types de commandes, il est créé lorsqu'un acheteur est sûr des détails de la commande tels que l'article, le prix, le calendrier de livraison, les conditions de paiement, etc.*

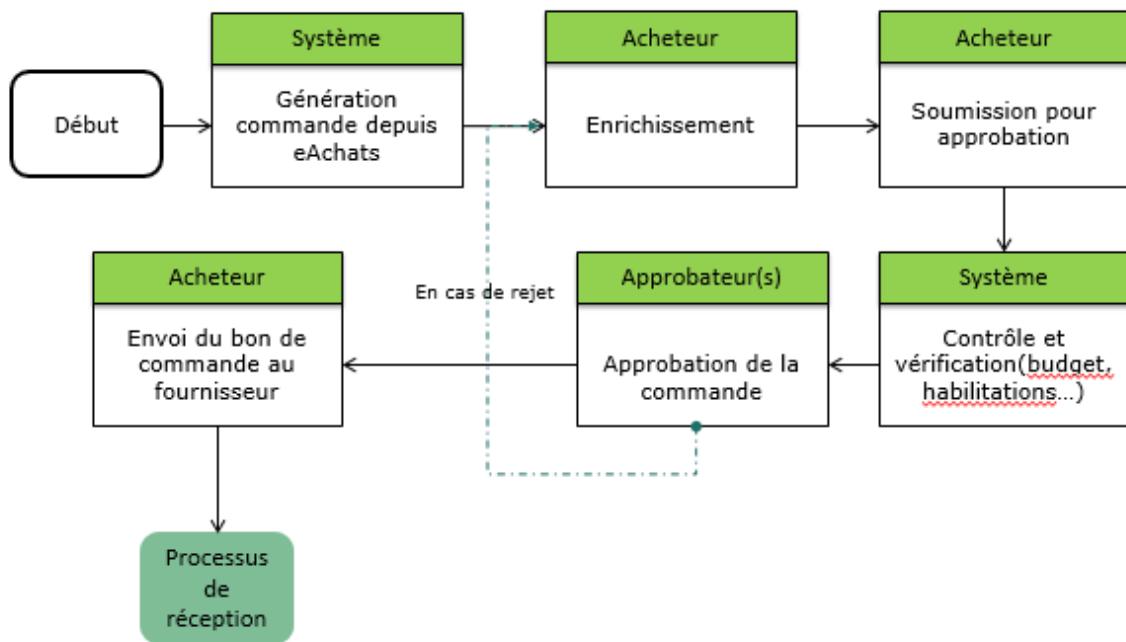


Figure 85: processus d'achat par commande standard

## 5. Achat par commande ouverte

Commande ouverte est une commande du client ou fournisseur qui correspond à un contrat portant généralement sur un an pour la fourniture de pièces à un prix déterminé. Cette commande ne spécifie pas les dates de livraison. Les appels de livraison provoquent l'approvisionnement ou l'expédition des pièces.

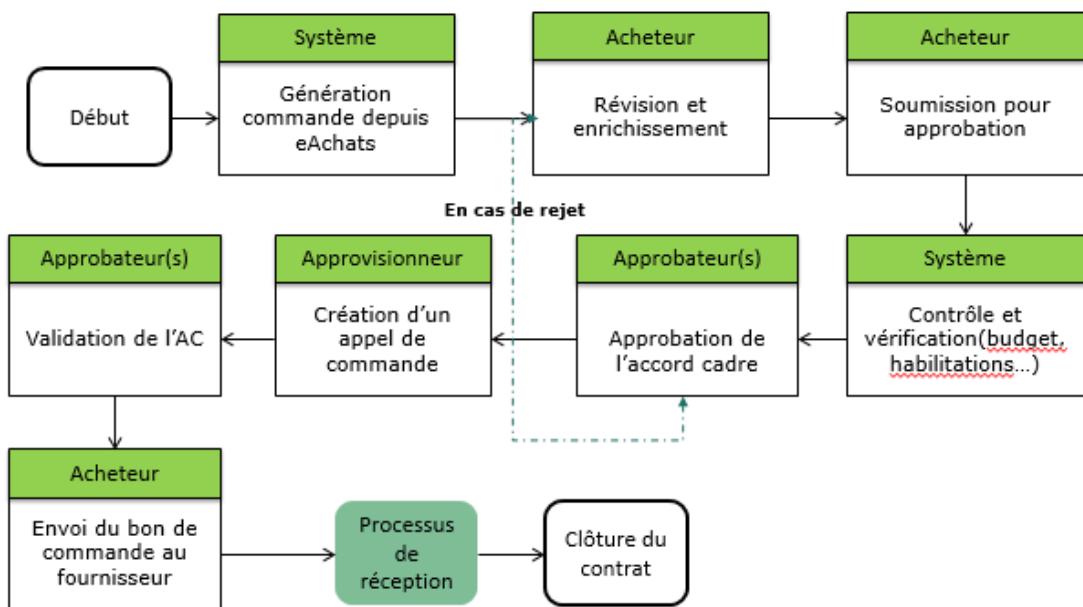


Figure 86: processus commande ouverte

## 6. Réception

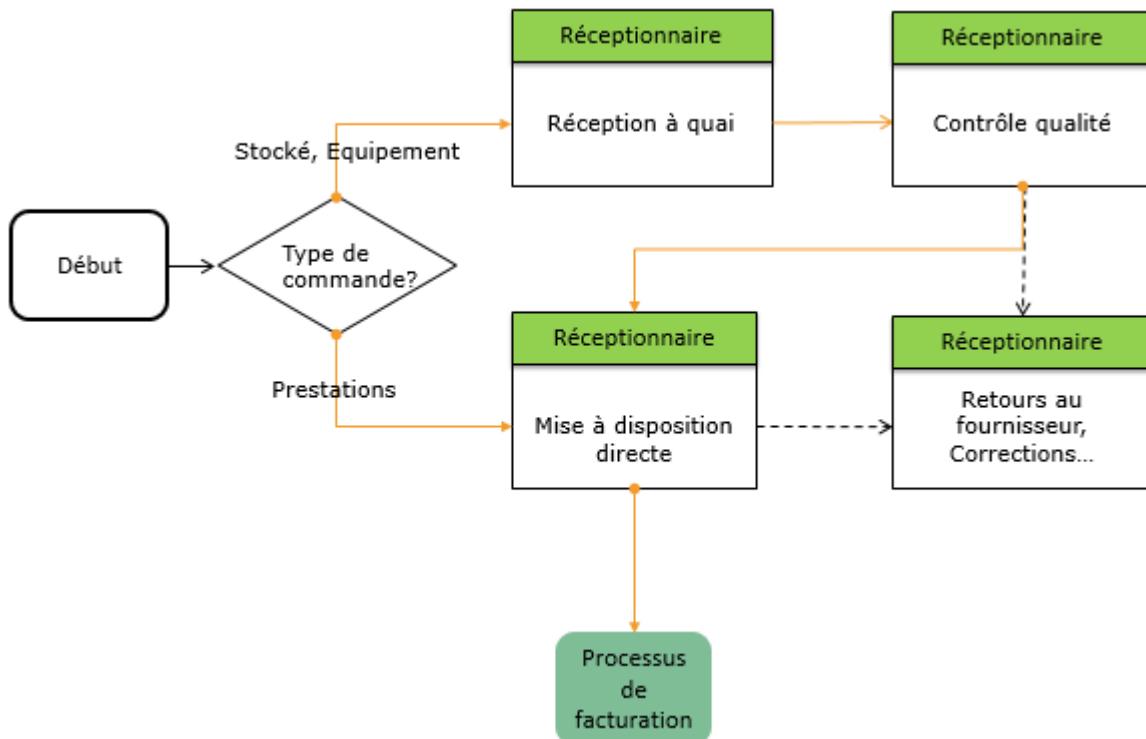


Figure 87: processus de réception

## 7. Facturation et paiement

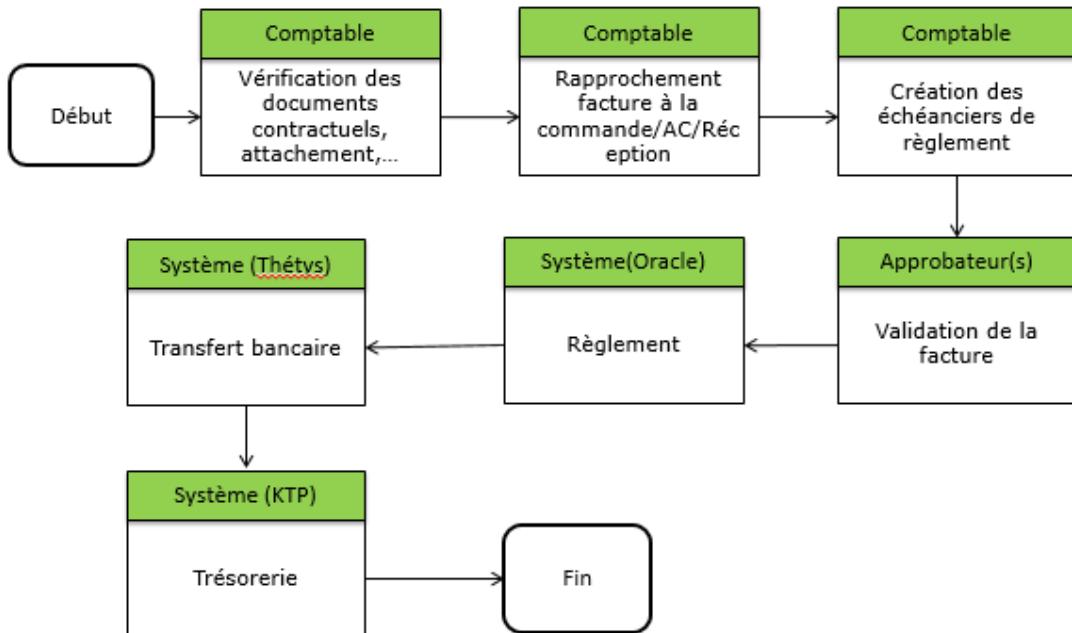


Figure 88: processus de facturation et paiement

## Annexe 3 : DevOps

### 1. Définition du modèle DevOps

Le DevOps est une combinaison de philosophies culturelles, de pratiques et d'outils qui améliore la capacité d'une entreprise à livrer des applications et des services à un rythme élevé, pour des produits qui évoluent et s'améliorent plus rapidement que ceux des entreprises utilisant des processus traditionnels de développement de logiciel et de gestion d'infrastructure. Cette vitesse permet aux entreprises de mieux servir leurs clients et de gagner en compétitivité.

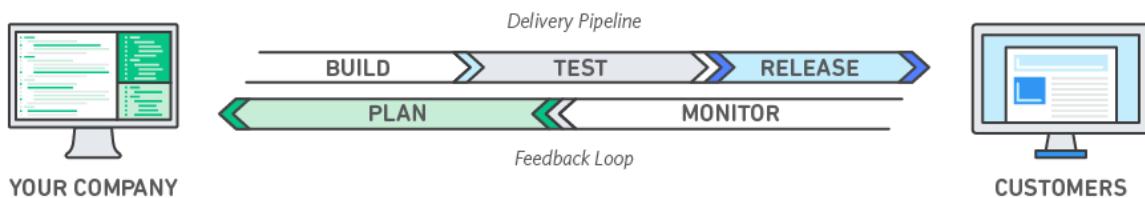


Figure 89: DevOps

### 2. Fonctionnement du DevOps

Dans un modèle axé sur DevOps, les équipes de développement et d'exploitation ne sont plus isolées. Il arrive qu'elles soient fusionnées en une seule et même équipe. Les ingénieurs qui la composent travaillent alors sur tout le cycle de vie d'une application, de sa conception et son test jusqu'au déploiement et son exploitation, et développent toute une gamme de compétences liées à différentes fonctions.

Dans certains modèles DevOps, les équipes d'assurance qualité et de sécurité peuvent également s'intégrer davantage au développement, à l'exploitation et au reste du cycle de vie d'une application. Lorsque chaque personne d'une équipe DevOps se concentre sur la sécurité, il est parfois question de DevSecOps.

Ces équipes utilisent des pratiques pour automatiser des processus qui étaient autrefois manuels et lents. Elles exploitent un stack technologique et des outils qui les aident à opérer et à faire évoluer des applications de façon rapide et fiable. Ces outils aident également les ingénieurs à accomplir de façon autonome des tâches (par exemple, le déploiement de code ou la mise en service d'infrastructure) qui nécessiteraient normalement l'aide d'autres équipes, ce qui augmente encore davantage leur productivité.

### 3. Les avantages du DevOps

- **Rapidité :** Avancer plus rapidement, accélérer le rythme des innovations pour vos clients, améliorer votre capacité d'adaptation au marché et gagner en efficacité. Avec le modèle DevOps, ces objectifs sont à la portée de vos équipes de développement et d'exploitation. Par exemple, les microservices et la livraison continue permettent aux équipes de s'approprier les services et de les mettre à jour plus rapidement.
- **Livraison rapide :** Augmentez le rythme et la fréquence des publications de façon à améliorer et à innover plus rapidement. Plus vite vous publiez de nouvelles fonctionnalités et corrigez des bogues, plus vite vous pouvez répondre aux besoins de vos clients et gagner en compétitivité. L'intégration continue et la livraison continue sont des pratiques qui automatisent le processus de publication de logiciel, de la conception jusqu'au déploiement.
- **Fiabilité:** Assurez la qualité des mises à jour d'application et des changements d'infrastructure, afin de livrer un produit fiable à un rythme accéléré tout en continuant de proposer une expérience positive aux utilisateurs finaux. Utilisez des pratiques comme l'intégration continue et la livraison continue pour vous assurer que chaque changement est fonctionnel et sûr. Des pratiques de supervision et de journalisation vous aident à rester informé sur les performances en temps réel.
- **Evolutivité:** Opérez et gérez vos processus d'infrastructure et de développement à grande échelle. L'automatisation et la consistance vous aident à gérer des systèmes complexes ou changeants de manière efficace et moins risquée. Par exemple, l'infrastructure en tant que code vous aide à gérer vos environnements de développement, de test et de production de façon constante et plus efficace.
- **Collaboration améliorée:** Créez des équipes plus efficaces avec un modèle culturel axé sur le DevOps, qui insiste sur des principes comme la prise de responsabilité. Les équipes de développement et d'exploitation collaborent en étroite collaboration, partagent de nombreuses responsabilités et combinent leurs workflows. Cela leur permet de limiter les pertes d'efficacité et de gagner du temps (par exemple en réduisant les délais de transfert entre les équipes de développement et d'exploitation et en écrivant du code prenant en compte l'environnement dans lequel il est exécuté).

## Annexe 4 : Architectures microservices

### 1. Définition

*Les microservices sont un style d'architecture logicielle à partir duquel un ensemble complexe d'applications est décomposé en plusieurs processus indépendants et faiblement couplés, souvent spécialisés dans une seule tâche. Les processus indépendants communiquent les uns avec les autres en utilisant des API langage-agnostiques.*

*Des API REST sont souvent employées pour relier chaque microservice aux autres. Un avantage avancé est que lors d'un besoin critique en une ressource, seul le microservice lié sera augmenté, contrairement à la totalité de l'application dans une architecture classique, par exemple une architecture trois tiers.*

### 2. Les aspects clés

- Chaque micro service peut être conçu à l'aide de n'importe quel outil et développé avec n'importe quel langage et technologie.
- Ils sont faiblement couplés puisque chaque micro service est physiquement séparé des autres,
- Indépendance relative entre les différentes équipes qui développent les différents microservices (en partant du principe que les APIs qu'ils exposent sont définis à l'avance).
- Facilité des tests et du déploiement ou de la livraison continue.