

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option: Génie Logiciel (G.L)

Thème

Développement d'un système d'auto provisionnement d'un ERP dans le Cloud «ODOaaS»

Réalisé par :

- Yahia ZEGHBA

Présenté le 06 Juillet 2019 devant le jury composé de :

- Mme. AMRAOUI Asma (Président)
- Mme. HALFAOUI Amel (Encadreur)
- Mme. KHITRI Souad (Examineur)

Année universitaire : 2018-2019

Dédicace

Je dédie ce travail:

A mes très chers parents

Pour tous vos aides, vos encouragements, durant tous mes parcours d'études et qui m'ont permis d'être ce que je suis aujourd'hui.

Que ce modeste travail, soit l'expression des vœux que vous n'avez cessé de formuler dans vos prières.

Qu'allah, le tout puissant, vous préserve et vous procure santé et longue vie.

A mes très chers frères

Youcef sedik, Salim et Taha

Vous occupez une place particulière dans mon cœur. Je vous dédie ce travail en vous souhaitant tout le bonheur et de succès dans votre vie

A mes très chers amis

En souvenir de nos bons moments et tout ce qu'on a vécu ensemble. J'espère de tout mon cœur que notre amitié durera éternellement.

Remerciements

Et lorsque votre Seigneur proclama:

« Si vous êtes reconnaissants, très certainement J'augmenterai Mes bienfaits pour vous » . (Coran - Sourat 14 Verset 7).

Je remercie Dieu le tout puissant de m'avoir donné la volonté et la force pour achever ce travail.

« Ce qui ne remercie pas les gens, ne remercie pas Allah » . (Hadith).

J'exprime mes remerciements à mon responsable encadrante : Mme. Amel Halfaoui pour l'assistance qu'elle m'a témoignée, et d'avoir accepté de m'encadrer pour mon projet de fin d'études.

J' exprime ma sincère gratitude à l'ensemble de l'équipe de l'entreprise IT Solutions pour leur chaleureux accueil et aide, et particulièrement à monsieur .

Je tiens à exprimer toute mes grandes gratitudes aux membres de jury d'avoir accepter de juger ce travail.

Mes vifs remerciements s'adressent également à tous mes enseignants de l'université Abou Bekr Belkaid de Tlemcen, pour la qualité de l'enseignement qu'ils m'ont prodigué au cours de mon parcours de master.

Sans oublier de remercier mes chers parents pour leur soutien, mes frères, mes proches, mes amis et toutes les personnes qui m'ont aidées de près ou de loin, au cours de mes études.

Table des matières

Liste des figures	8
Liste des Tableaux	9
Liste des abréviations	10
Introduction générale	11
1 Contexte général du projet «ODOaaS»	14
1.1 Introduction	14
1.2 Présentation de l'organisme d'accueil	14
1.2.1 Présentation de l'entreprise IT Solutions	14
1.2.2 Domaines d'expertises	14
1.2.3 Quelques solutions proposées	15
1.3 Système d'auto provisionnement d'un ERP dans le cloud «ODOaaS»	16
1.3.1 Architecture fonctionnelle du système «ODOaaS»	16
1.3.2 Architecture de déploiement des systèmes ERP odoo	18
1.3.3 Comparatif entre les outils similaire à «ODOaaS»	19
1.4 Conclusion	21
2 Analyse et conception de l'application «ODOaaS»	22
2.1 Introduction	22
2.2 Spécification des besoins du système	22
2.2.1 Identification des acteurs	22
2.2.2 Besoins fonctionnels	23
2.2.3 Besoins non fonctionnels	24
2.3 Modélisation des besoins fonctionnels	24
2.3.1 Diagramme de cas d'utilisation	24

2.3.2	Diagrammes de séquences	27
2.3.3	Diagramme de classe	40
2.4	Conclusion	41
3	Implémentation du système «ODOaaS»	43
3.1	Introduction	43
3.2	Méthode de gestion de projet	43
3.2.1	Méthode Scrum	43
3.2.2	Diagramme de Gantt	46
3.3	Techonologies utilisées	48
3.3.1	Framework Django	48
3.3.2	Framework Vuejs	48
3.3.3	ERP Odoo	48
3.3.4	Docker	49
3.3.5	Git	50
3.3.6	Latex	50
3.4	Outils utilisées	50
3.4.1	draw.io	50
3.4.2	Trello	51
3.4.3	Github	51
3.4.4	Postman	51
3.4.5	Sublime	51
3.4.6	PhpMyAdmin	52
3.4.7	MySQL Workbench	52
3.5	L'architecture de l'application	53
3.6	Maquettes de l'application	54
3.6.1	Pré-configuration du système ERP	54
3.6.2	Authentification	57
3.6.3	Gérer les demandes des clients	58
3.6.4	Gérer les environnements déployés	61
3.6.5	Gérer les modules marketing	63
3.6.6	Gérer les modules techniques	64
3.6.7	Gérer les versions d'odoo	65
3.7	Conclusion	65
	Conclusion générale et perspectives	66

Références webographiques

70

Table des figures

1.1	Architecture fonctionnelle du système «ODOaaS»	17
1.2	Architecture de déploiement des environnements odoo	19
2.1	Diagramme du cas d'utilisation « Gestion du déploiement et de la configuration des systèmes ERP dans le cloud»	25
2.2	Diagramme du cas d'utilisation « Pré-configuration d'un système ERP» .	26
2.3	Diagramme de séquence « S'authentifier »	27
2.4	Diagramme de séquence « Pré-configurer le Système ERP »	29
2.5	Diagramme de séquence « Valider le déploiement et la configuration d'un environnement »	32
2.6	Diagramme de séquence « Traiter les demandes des clients »	35
2.7	Diagramme de séquence « Ajouter un environnement Demo »	37
2.8	Diagramme de séquence « Gérer les environnements déployées »	39
2.9	Diagramme de classe «application ODOaaS»	41
3.1	Processus de Scrum[29]	44
3.2	Les itérations du projet «ODOaaS»	45
3.3	Listes des Tâches «ODOaaS»	46
3.4	Diagramme de Gantt du Sprint 1 du projet «ODOaaS»	47
3.5	Architecture de l'application «ODOaaS»	54
3.6	Interface de pré-configuration des Coordonnées de L'entreprise	55
3.7	Interface de pré-configuration des Utilisateurs	56
3.8	Interface d'authentification au système «ODOaaS»	57
3.9	Interface de gestion des demandes des clients	58
3.10	Interface de personnalisation de la pré-configuration	59
3.11	Interface de validation de déploiement et configuration d'un ERP	60
3.12	Interface de gestion des environnements déployés	61

3.13	Interface de déploiement d'un nouveau environnement odoo de type démo	62
3.14	Interface de gestion des modules marketing	63
3.15	Interface de gestion des modules techniques	64
3.16	Interface de gestion des versions odoo	65

Liste des tableaux

1.1 Etude comparative entre quelques solutions de Saas d'odoo 20

Liste des abréviations

ERP : Enterprise Resource Planning.

ODOaaS : On Demand Odoo as a Service.

SaaS : Software as a Service.

UML : Unified Modeling Language.

MVC Model View Controller

SPA : Single Page Application

API : Application Programming Interface

SDK : Software Development Kit

LXC : Linux Container

RPC : Remote Procedure Call

XML : Extensible Markup Language

JSON : JavaScript Object Notation

YAML : Yet Another Markup Language

HTML : Hyper Text Markup Language

CSS : Cascading Style Sheets

URL : Uniform Resource Locator

Introduction générale

Contexte du projet

Dans le cadre de réalisation de notre projet de la fin du cycle d'études du Master Génie Logiciel à l'université Abou Bekr Belkaid à Tlemcen, nous avons effectué notre stage au sein de l'entreprise «IT Solutions»[1] à Alger. Il s'insère directement à notre cursus pédagogique(à savoir le second semestre de notre deuxième année d'études de master). Ce stage nous a permis de bénéficier d'une véritable expérience professionnelle qui nous facilitera l'insertion dans le monde de travail. Ce stage nous a permis aussi à concrétiser le savoir acquis durant le parcours universitaire et nous aidé à spécialiser dans le domaine souhaité

IT Solutions est une société de services en ingénierie informatique, s'appuie sur une expérience de 20 ans et des dizaines de projets au service des entreprises locales et internationales, alliant connaissance parfaite du marché local à l'expertise, aux méthodologies et aux standards internationaux les plus éprouvés, ce qui représente une proposition de valeur unique en Algérie.[1]

Elle met à disposition de ses clients un ensemble de compétences et d'experts, pour le déploiement, la maîtrise et l'optimisation des systèmes informations : ERP - CRM - Business Intelligence - Portail et collaboration.[1]

INSIDJAM ERP est une solution maintenu par l'entreprise IT Solutions, c'est une suite de progiciels de gestion intégrés, composée d'un ensemble d'outils technologiques et de services complémentaires ayant comme finalité l'harmonisation et la rationalisation des opérations de gestion des entreprises.

Les outils INSIDJAM sont tous basés sur des technologies ouvertes et robustes, préconçus pour une prise en charge optimale des pratiques de gestion locales et des problématiques sectorielles diverses.

INSIDJAM ERP est un fork du projet Open Source Odoo (Ex Open ERP)[2]. Son objectif est de donner au projet communautaire Odoo une direction stratégique conforme aux attentes et aux contraintes des pays sous-développés en général et à l'Algérie en particulier.

Problématique

IT Solutions propose depuis plusieurs années une version du logiciel intégré de gestion « Insidjam » en mode hébergé dans les nuages.

Les clients qui optent pour ce mode d'hébergement gagent une réduction significatives, de l'ordre de 30% en 4 ans, des coûts liés à l'exploitation de la solution en interne : coûts des serveurs physiques, administration, sauvegarde, maintenance, énergie... etc.

La facturation de la solution en mode abonnement annuelle ou mensuel par utilisateur, permet aussi aux clients de dépenser progressivement et à la hauteur du niveau d'utilisation. L'avènement des offres commerciales « cloud » a permis une réduction significative des barrières à l'acquisition par les petites et moyennes entreprises des solutions de type ERP. Malheureusement, les coûts de déploiement et de configuration non pas connues le même niveau de changement.

La réduction des coûts de déploiement et de configuration initiale passe nécessairement par l'automatisation des tâches de déploiement et de configuration récurrentes en se basant sur un questionnaire, pas à pas, qui permet de préconfigurer le système à 90%. Les 10% des tâches restantes pourront être prises en charge par des agents de support ou des utilisateurs avancés et non pas par des administrateurs système ou des consultants techniques.

Objectifs

Ce projet consiste a développé l'application web «ODOaaS» (**O**n **D**emand **O**odoo **a**s a **S**ervice), ça signifie que l'application fournit des Systèmes ERP Odoo aux clients en tant que service à la demande. Ce projet a pour objectif :

1. L'automatisation du déploiement du système ERP dans le cloud
2. L'automatisation de la configuration des environnements clients
3. Garder la traçabilité de la pré-configuration du système ERP dans la base de données
4. Faciliter la maintenance de la configuration

5. Minimiser le temps de la configuration et de déploiement
6. Faciliter la gérance des environnements déployées

Plan du mémoire

Ce rapport s'articule autour de trois chapitres, organisé comme suit :

- **Chapitre 1** : nous allons présenter dans ce premier chapitre l'entreprise d'accueil dans lequel notre stage pratique s'est déroulé, ensuite nous décrirons notre projet, et on exposera une étude de l'existant.
- **Chapitre 2** : nous allons entamer dans ce deuxième chapitre la phase analyse et conception de notre système, nous commencerons par la spécification des besoins du projet, ensuite nous passerons à la réalisation des diagrammes qui illustrent la vue fonctionnelle, comportementale, structurelle de notre système.
- **Chapitre 3** : est consacré pour l'implémentation de notre application web, en détaillant ses différentes étapes. Nous présenterons l'approche adoptée pour gérer et suivre le déroulement de notre projet, aussi nous exposerons l'environnement de développement, et les technologies et outils utilisés pour la réalisation de notre projet.

Nous clôturerons ce document par une conclusion générale et un résumé du projet réalisé, aussi nous allons proposer des perspectives futures, qui visent à perfectionner notre système, afin qu'il puisse répondre à toutes les exigences du client.

Chapitre 1

Contexte général du projet «ODOaaS»

1.1 Introduction

Dans ce premier chapitre, nous présentons une étude préliminaire de notre projet «ODOaaS». En premier, nous commencerons par la présentation de l'organisme d'accueil «IT Solutions» dans lequel nous avons effectué notre stage de fin d'étude. En suite, nous introduisons notre système «ODOaaS» qu'on vise à réaliser dans ce projet. En fin, nous exposons une étude de l'existant.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation de l'entreprise IT Solutions

IT Solutions est une société de services en ingénierie informatique, elle s'appuie sur une expérience de 20 ans et des dizaines de projets au service des entreprises locales et internationales, alliant connaissance parfaite du marché local à l'expertise, aux méthodologies et aux standards internationaux les plus éprouvés, ce qui représente une proposition de valeur unique en Algérie.[1]

1.2.2 Domaines d'expertises

IT Solutions met à disposition de ses clients un ensemble de compétences et d'experts pour :

- Maîtrise et l'optimisation des systèmes d'informations
- Développement Spécifique
- Intégration de solutions de gestion intégrée ERP
- Intégration de solution de gestion de relation client CRM
- Intégration de Solutions de mobilité : CRM, Field Service et Logistique.
- Expertise sectorielle : Industries, Chaîne de magasin, Distribution, Pharma, auto-mobile
- Business Intelligence
- Portail et collaboration
- Conseils et assistance

1.2.3 Quelques solutions proposées

- **INSIDJAM ERP** : est la nouvelle alternative qui s'offre aux entreprises locales, grâce aux connaissances et à l'expérience de ses experts locaux, INSIDJAM ERP assure aux professionnels des secteurs qu'ils adressent, des solutions personnalisées et adaptées aux pratiques du terrain et législations locales. Il offre aussi, la flexibilité des solutions ouvertes, la stabilité et la garantie d'un éditeur proche de la réalité du terrain, assurant un accompagnement de proximité et un engagement contractuel efficient.[1]
- **SAP S4 HANA** : Résolument orientée expérience utilisateur, S4/HANA de SAP est la suite d'applications de gestion intégrée de nouvelle génération du leader Mondial SAP. Tout en étant dotée d'une couverture fonctionnelle élargie, elle repose sur deux principes fondateurs :
 - Lien naturel entre les visions transactionnelles et décisionnelle grâce à sa technologie révolutionnaire de base de données In Memory qui tient la promesse du « vrai temps réel »
 - Une interface utilisateur « smart », simple d'accès et très visuelle – basée sur SAP FIORI – qui réinvente l'expérience utilisateur et simplifie grandement l'appropriation par les utilisateurs.[1]
- **Oracle JD Edwards EnterpriseOne** : Oracle JD Edwards EnterpriseOne est une suite d'applications intégrée de logiciels complets de planification des ressources d'entreprise associant valeur commerciale, technologie standardisée et expérience

approfondie de l'industrie en une solution assurant un faible coût total d'exploitation. EnterpriseOne est la première solution ERP à exécuter toutes les applications sur Apple iPad. JD Edwards EnterpriseOne fournit également des applications mobiles.[1]

1.3 Système d'auto provisionnement d'un ERP dans le cloud «ODOaaS»

Notre projet consiste à développer un outil de Saas (Software as a Service ou Logiciel en tant que Service). Saas est un modèle de distribution de logiciels permettant aux fournisseurs d'applications de proposer leurs applications ou services tiers aux clients via Internet.[22] Notre solution sert à fournir des systèmes ERP odoo en tant que SaaS, et permet leur industrialisation et leur configuration et déploiement dans le cloud.

1.3.1 Architecture fonctionnelle du système «ODOaaS»

Nous présentons dans cette section les deux parties de notre application web, back office et front office.

1.3.1.1 Description de la partie back office de «ODOaaS»

L'application back office s'agit de l'interface d'administration qui permet à l'entreprise de gérer son application, et qui exige un identifiant et mot de passe pour avoir un accès.[24]

Dans notre système, cette partie permet à l'administrateur de :

- Gérer les demandes des clients.
- Déployer et configurer des systèmes ERP.
- Gérer les environnements déployés.
- Gérer les paramètres de configuration des ERP.
- Gérer les paramètres de déploiement des ERP.

1.3.1.2 Description de la partie front office de «ODOaaS»

L'application front Office désigne la partie frontale de l'entreprise, accessible aux utilisateurs finaux et aux clients.[23]

Dans notre système, cette partie permet aux clients de pré-configurer leur système ERP et d'envoyer leur demande à l'administrateur pour avoir une version démo ou production de l'ERP.

Dans la figure suivante, nous montrons l'architecture fonctionnelle de notre système, qui consiste à schématiser les fonctionnalités globales décrites précédemment.

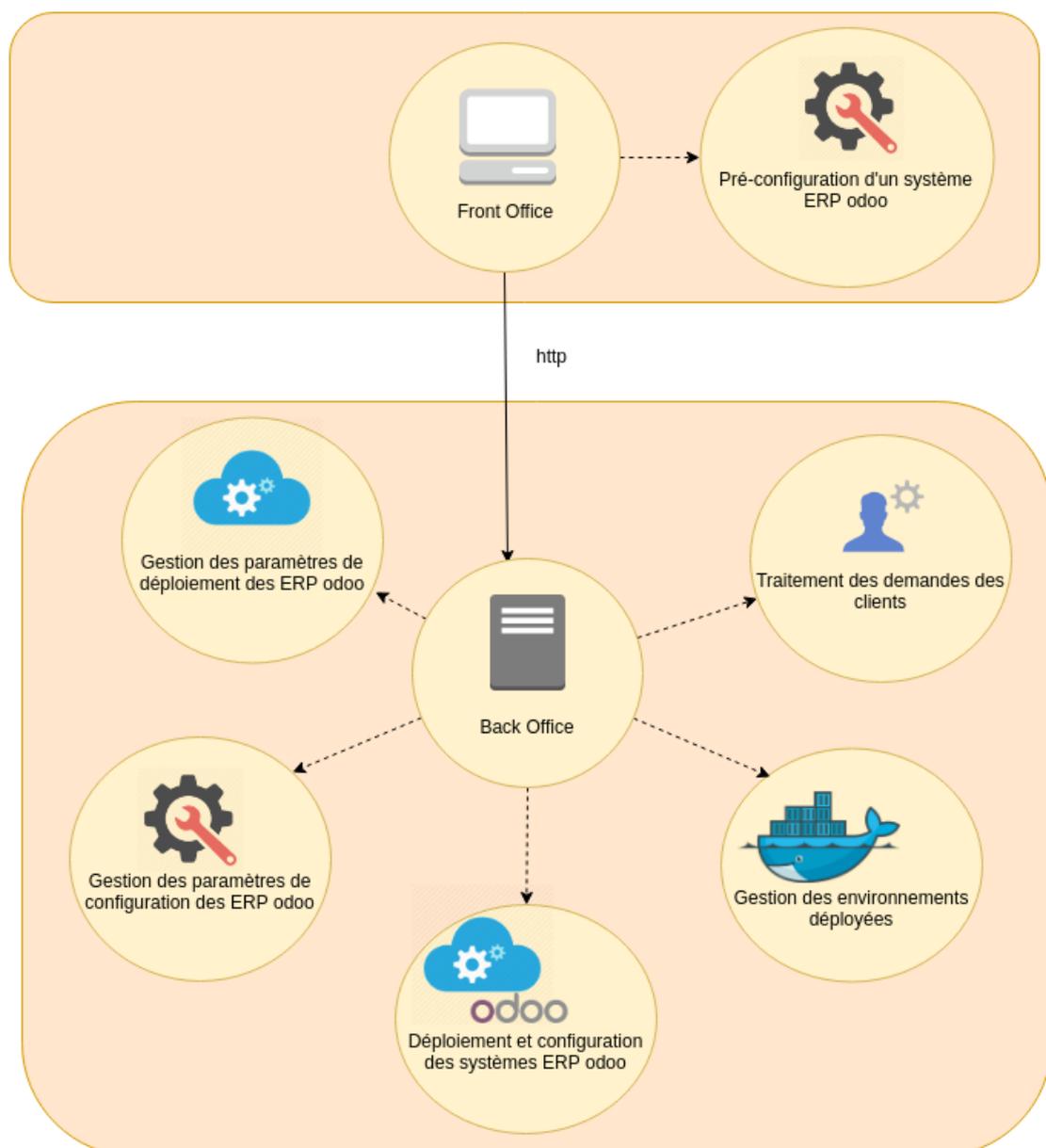


FIGURE 1.1 – Architecture fonctionnelle du système «ODOaaS»

1.3.2 Architecture de déploiement des systèmes ERP odoo

Dans cette partie, nous nous intéressons à la phase de déploiement, qui est la plus importante de notre solution, et qui consiste à automatiser les processus de déploiement et de la configuration des systèmes ERP dans le cloud.

1.3.2.1 Description de l'architecture d'un environnement odoo

Nous avons utilisé pour le déploiement des systèmes ERP odoo la technologie docker, qui sert à déployer et gérer des conteneurs d'applications virtualisées sur un système d'exploitation.[12]

Dans notre système, nous avons défini un environnement Odoo qui est sous forme d'une composition de deux conteneurs connectés par un réseau privé de type bridge, l'un des conteneurs est pour déployer l'application Odoo, et l'autre pour déployer le SGBD PostgreSQL qui contient des instances du système ERP, représentées par des bases de données. Ces conteneurs communiquent par des ports exposés à l'intérieur du réseaux privé défini entre eux. Tous les conteneurs d'application odoo sont connectés avec l'outil de saas dans un seul réseau privé, afin d'avoir la possibilité de configurer les différentes instances à travers l'API odoo.

Pour la sécurité des systèmes ERP fournis par notre solution saas, nous avons mis en place un serveur nginx qui sert à faire le reverse proxy, c'est à dire, il représente un intermédiaire entre les clients et les systèmes odoo déployés, en canalisant les requêtes reçus par les clients, en les transmettant aux conteneurs d'application odoo ciblés

1.3.2.2 Processus de déploiement d'un environnement odoo

le déploiement d'un environnement odoo est constitué d'un ensemble des opérations, qui sont comme suit :

- **Choix de version d'application odoo à déployer.**
- **Choix de type d'environnement odoo :** nous distinguons deux types d'environnements odoo, qui désignent le mode d'exploitation du produit ERP :
 - **Environnement démo :** contient un nombre limité d'instances ERP de plusieurs clients, pendant une certaine période d'essai.
 - **Environnement de production :** est spécialement conçu pour un client en mode de production, contient ses instances d'ERP
- **Création d'un instance ERP dans l'environnement odoo sélectionné.**

- **Configuration de l'instance ERP créée**

Nous montrons dans la figure suivante l'architecture de déploiements des environnements odoo expliqués au-dessus.

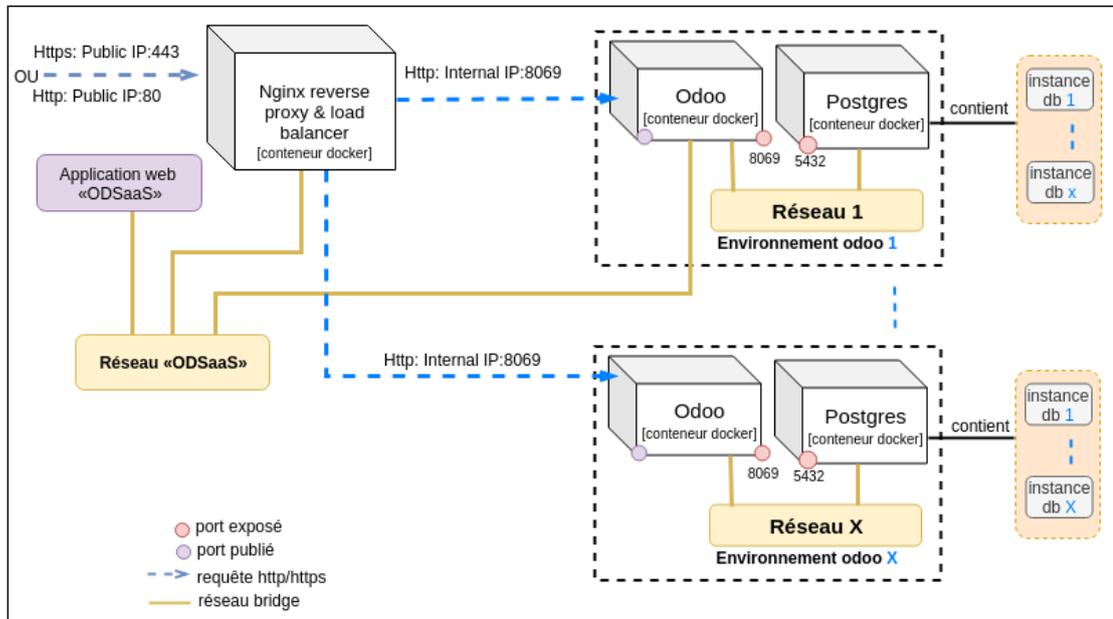


FIGURE 1.2 – Architecture de déploiement des environnements odoo

1.3.3 Comparatif entre les outils similaire à «ODOaaS»

Dans cette partie, nous faisons une étude comparative entre la solution «ODOaaS» que nous avons implementé dans ce projet et quelques solutions existantes sur le marché, comme : Odoo SaaS Kit, Odoo Saas Tool.

- **Odoo SaaS Kit** : c'est une solution du saas qui fournit odoo en tant que service, développé par l'organisation de Webkul Software Pvt. Ltd.[25]
- **Odoo Saas Tools** : c'est un projet open source qui consiste à vendre et gérer des bases de données odoo.[26]
- **Odoo Multi Instance Control(TrendERP Saas)** : c'est un module qui permet de déployer rapidement des instances Odoo pour des clients. Gérer les licences utilisateur et les applications pour Odoo[27]

Nous présentons dans le tableau suivant une comparaison entre les différentes solutions.

	Odoo Saas kit	Odoo Saas tools	Odoo Multi Instance Control	ODOaaS version alpha
Pré-configuration d'instance ERP côté client	NON	NON	NON	OUI
Pré-configuration d'instance ERP côté administrateur	NON	NON	NON	OUI
Ouvert pour les modules personnalisés	OUI	OUI	OUI	OUI
Versions d'odoo supportées	11, 12	8, 9, 10, 11	11	11
Supporte plusieurs versions d'odoo en un seul instance du saas	NON	NON	NON	OUI
Fourniture des environnements virtualisés séparés, en mode production ou d'essai, pour la création d'instance	NON	NON	NON	OUI
Filtrer des instances en créant des URL de sous-domaines pour empêcher les utilisateurs de voir d'autres instances	OUI	OUI	OUI	NON
Création d'instance sur demande du client	OUI	OUI	OUI	OUI
Création et suppression manuelle d'instances à partir du panneau d'administration	OUI	OUI	OUI	NON
migration d'instances entre hôtes	NON	NON	OUI	NON
Fourniture de divers plans d'abonnement	OUI	OUI	NON	NON
Intégration de passerelle de paiement	OUI	NON	NON	NON
surveiller les ressources système allouées à l'instance	NON	NON	OUI	NON

TABLE 1.1 – Etude comparative entre quelques solutions de Saas d'odoo

Nous visons par cette comparaison, à déterminer la valeur ajoutée de notre solution par rapport à d'autres, et planifier à réaliser les fonctionnalités manquantes, pour avoir un système complet qui répond à tous les besoins des fournisseurs de solution ERP odoo.

1.4 Conclusion

Durant ce premier chapitre, nous avons présenté l'entreprise d'accueil dans laquelle notre projet s'est déroulé, puis nous avons introduit notre système «ODOaaS», Enfin, nous avons terminé par une étude comparative entre les systèmes existants similaire au notre.

Chapitre 2

Analyse et conception de l'application «ODOaaS»

2.1 Introduction

Ce chapitre est consacré à la définitions des grandes fonctionnalités de notre système et à l'identification des rôles des utilisateurs. En premier, nous entamons la spécification des besoins fonctionnels et non fonctionnels, et l'explication des rôles des acteurs impliqués dans notre système, ensuite on passera à la phase modélisation et conception qui consiste à donner une idée plus claire de l'analyse de notre système en utilisant le langage UML, et en adoptant la méthodes Scrum, qui est une méthode Agile, pour la gestion du cycle de vie de notre application.

Les diagrammes UML utilisés sont :

- * Les diagrammes de cas d'utilisation qui spécifient les besoins de notre système.
- * Les diagrammes de séquence qui détaillent quelques cas d'utilisation.
- * Un diagramme de classes qui décrit la structure de notre système.

2.2 Spécification des besoins du système

2.2.1 Identification des acteurs

Les acteurs que nous avons identifié sont :

- * **Administrateur** : a pour rôle principal, la validation des demandes des clients en validant la configuration et le déploiement du système ERP
- * **Client** : a pour rôle d'envoyer une demande d'obtention de sa propre version démo du système ERP en pré-configurant les parties nécessaires du système

2.2.2 Besoins fonctionnels

Cette partie consiste à présenter l'ensemble des besoins fonctionnels auxquels notre solution doit répondre, en les classifiant par chaque acteur qui lui sont reliés.

* Le système permet aux clients de :

- **Pré-configurer le système ERP** : le système permet aux clients de pré-configurer leur système ERP démo à partir d'un wizard, qui contient un ensemble de formulaires en plusieurs étapes.

* Le système permet à l'administrateur de :

- **Traiter les demandes des clients** : le système doit permettre à l'administrateur de modifier et personnaliser la pré-configuration du système d'un client.
- **Valider la configuration et le déploiement d'un environnement client** : le système doit permettre à l'administrateur de valider la configuration et le déploiement d'un environnement client et paramétrer les options de validation.
- **Ajouter un environnement démo** : le système doit permettre à l'administrateur de déployer un nouveau environnement démo du système ERP, où on peut créer des instances du système ERP.
- **Gérer les environnements déployés** : le système doit permettre à l'administrateur de démarrer, stopper, détruire ou consulter les détails d'un environnement du système ERP déployé.
- **Gérer les modules marketing** : le système doit permettre à l'administrateur d'ajouter, supprimer, modifier un pack ou une option.
- **Gérer les versions du système ERP** : le système doit permettre à l'administrateur d'ajouter, modifier une version du système ERP.
- **Gérer les modules techniques** : le système doit permettre à l'administrateur d'ajouter, supprimer, modifier un module technique.

- **Gérer les rôles des utilisateurs** : le système doit permettre à l'administrateur d'ajouter, supprimer, modifier un rôle d'utilisateur.

2.2.3 Besoins non fonctionnels

Dans cette partie, on présente les besoins non fonctionnels qui caractérisent notre système, et qui nous aident à fournir un système performant et de bonne qualité.

L'ensemble des fonctionnalités à implémenter doivent répondre aux besoins distingués :

- **Ergonomie** : l'application doit offrir des interfaces faciles à utiliser et adaptées à l'utilisateur.
- **Performance** : l'application doit fournir des fonctionnalités optimales en terme de temps de réponse, et de temps de traitement.
- **Sécurité** : l'application doit satisfaire la contrainte de la sécurité et respecter les règles qui concernent la sécurité des systèmes informatiques.
- **Extensibilité** : l'application doit faciliter les modifications ou l'ajouter des nouvelles fonctionnalités.

2.3 Modélisation des besoins fonctionnels

2.3.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre[3].

Dans cette partie nous identifions et modéliser les besoins des utilisateurs de notre système «ODOaaS» à partir des diagrammes de cas d'utilisations.

2.3.1.1 Diagramme de cas d'utilisations global

La figure suivante représente le diagramme de cas d'utilisation global de notre application «ODOaaS».

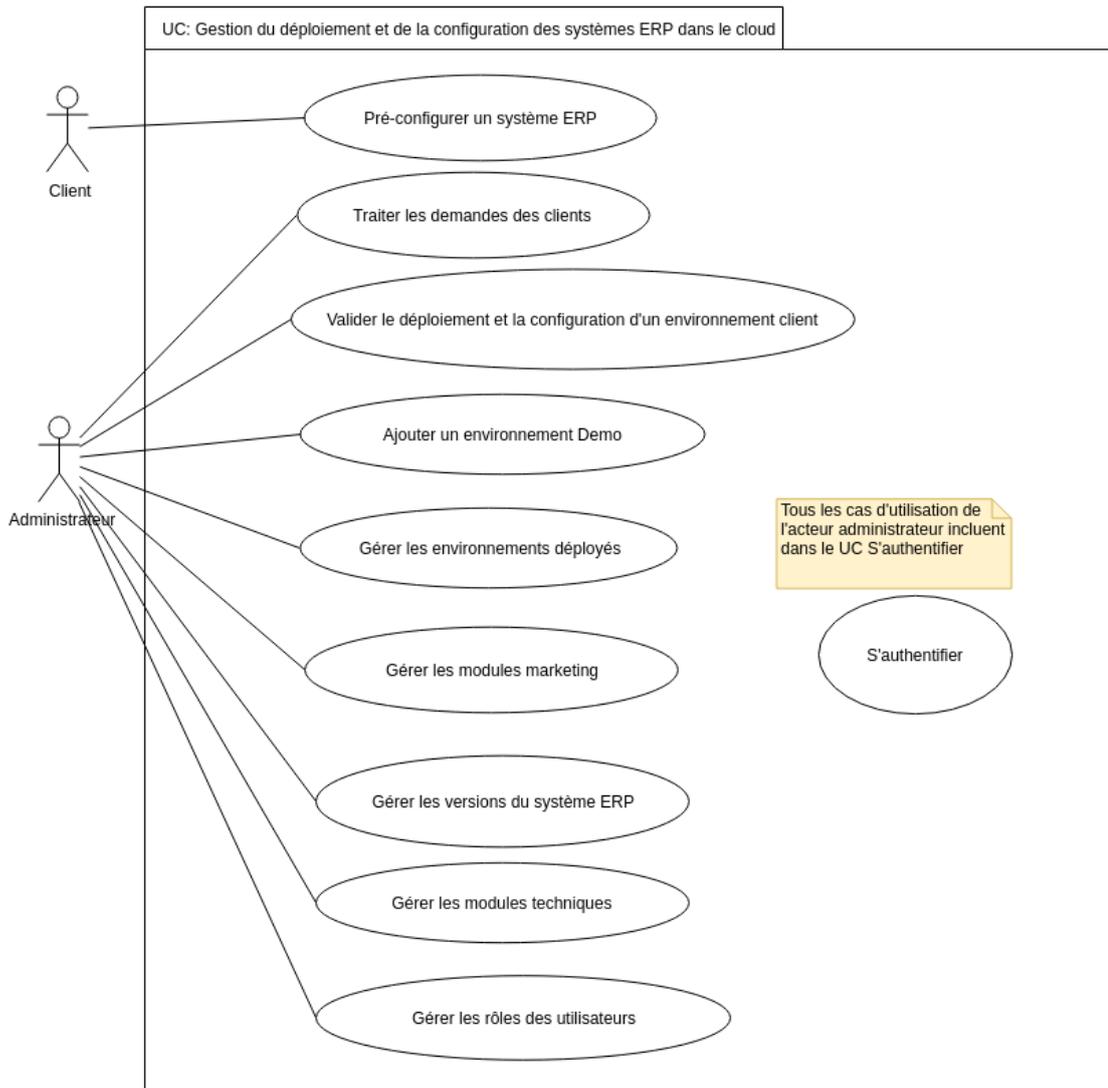


FIGURE 2.1 – Diagramme du cas d'utilisation « Gestion du déploiement et de la configuration des systèmes ERP dans le cloud»

2.3.1.2 Diagramme de cas d'utilisations « Pré-configurer un système ERP »

Nous détaillons dans la figure suivante le cas d'utilisation « Pré-configurer un système ERP » par sous diagramme de cas d'utilisations.



FIGURE 2.2 – Diagramme du cas d'utilisation « Pré-configuration d'un système ERP »

2.3.2 Diagrammes de séquences

Le diagramme de séquence système représente la vue comportementale du système, et qui permet de présenter les différentes interactions entre les acteurs et le système à partir des messages dans un ordre chronologique.

Nous détaillons dans la partie ci-dessous l'ensemble des diagrammes de séquences les plus importants de notre système.

2.3.2.1 Diagramme de séquence « S'authentifier »

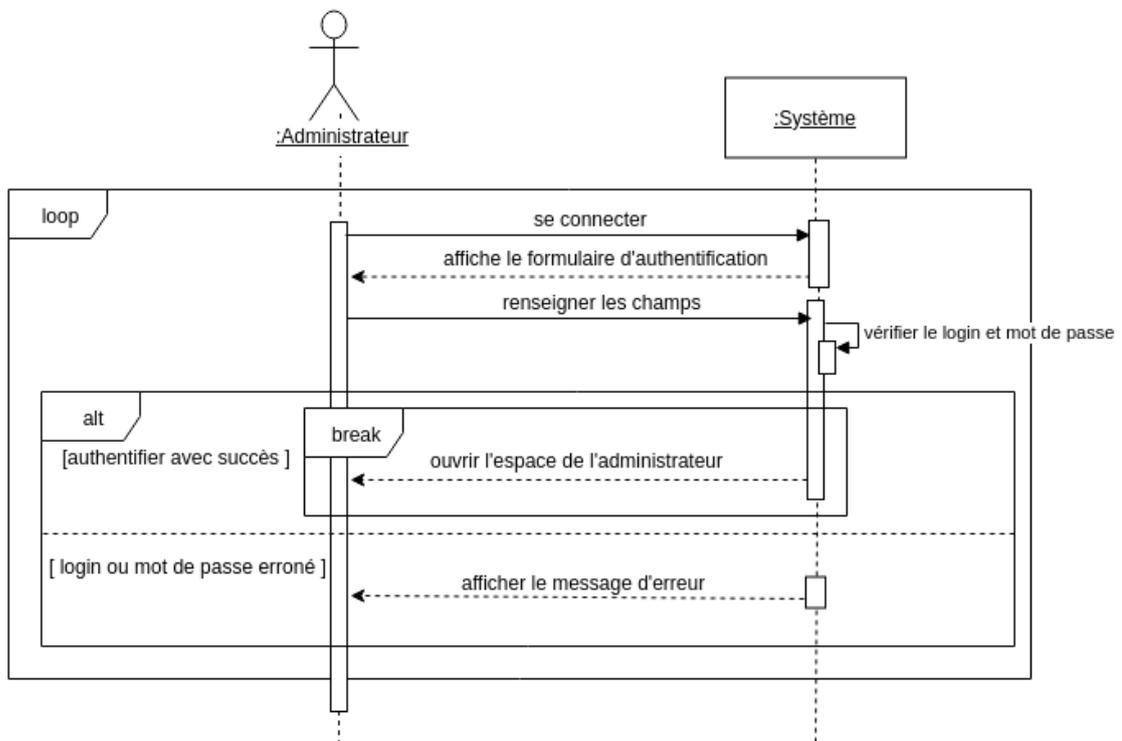


FIGURE 2.3 – Diagramme de séquence « S'authentifier »

Description textuelle du diagramme de séquence «S'authentifier»

Acteur : Administrateur.

Précondition : L'utilisateur possède un profil.

Postcondition : Sessoin personnelle ouverte.

Description des scénarios :

a) **Scénario nominale :**

1. L'administrateur demande le formulaire d'authentification.
2. Le système affiche le formulaire d'authentification.
3. L'administrateur renseigne les champs.
4. Le système vérifie le login et mot de passe.
5. Le système ouvre l'espace administrateur.

b) Scénarios d'exception :

- 4.a. L'administrateur saisit des informations fausses.
 - * Le système affiche le message d'erreur.
 - * Aller à 1.

2.3.2.2 Diagramme de séquence « Pré-configurer le Système ERP »

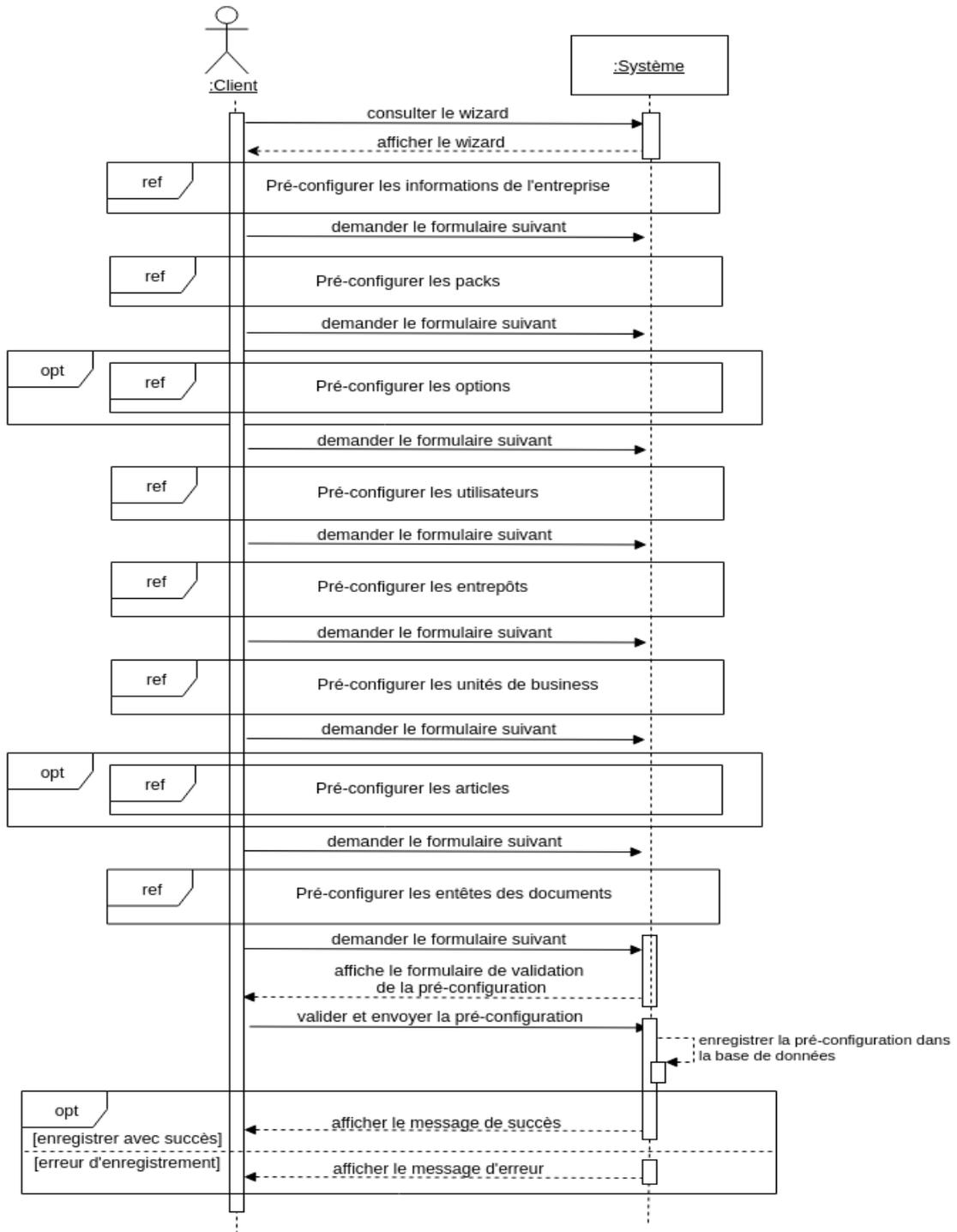


FIGURE 2.4 – Diagramme de séquence « Pré-configurer le Système ERP »

Description textuelle du diagramme de séquence «Pré-configurer le Système ERP»

Acteur : Administrateur.

Précondition : Accès à la page du wizard.

Postcondition : Sauvegarder la pré-configuration du client dans la base données.

Description des scénarios :

a) **Scénario nominale :**

1. Le client demande l'affichage du wizard.
2. Le système affiche le wizard.
3. Le système fait appel au cas d'utilisation «Pré-configurer les informations de l'entreprise».
4. Le client demande le formulaire suivant.
5. Le système fait appel au cas d'utilisation «Pré-configurer les packs».
6. Le client demande le formulaire suivant.
7. Le système fait appel au cas d'utilisation «Pré-configurer les options».
8. Le client demande le formulaire suivant.
9. Le système fait appel au cas d'utilisation «Pré-configurer les utilisateurs».
10. Le client demande le formulaire suivant.
11. Le système fait appel au cas d'utilisation «Pré-configurer les entrepôts».
12. Le client demande le formulaire suivant.
13. Le système fait appel au cas d'utilisation «Pré-configurer les unités de business».
14. Le client demande le formulaire suivant.
15. Le système fait appel au cas d'utilisation «Pré-configurer les articles».
16. Le client demande le formulaire suivant.
17. Le système fait appel au cas d'utilisation «Pré-configurer les entêtes des documents».
18. Le client demande le formulaire suivant.
19. Le système affiche le formulaire de validation de la pré-configuration.
20. Le client valider et envoyer la pré-configuration.
21. Le système enregistre la pré-configuration dans la base de données.

22. Le système affiche le message de succès.

b) Scénarios d'exception :

21.a. Le système rencontre une erreur d'enregistrement de la pré-configuration.

- * Le système affiche le message d'erreur.
- * Aller à 19.

2.3.2.3 Diagramme de séquence « Valider le déploiement et la configuration d'un environnement »

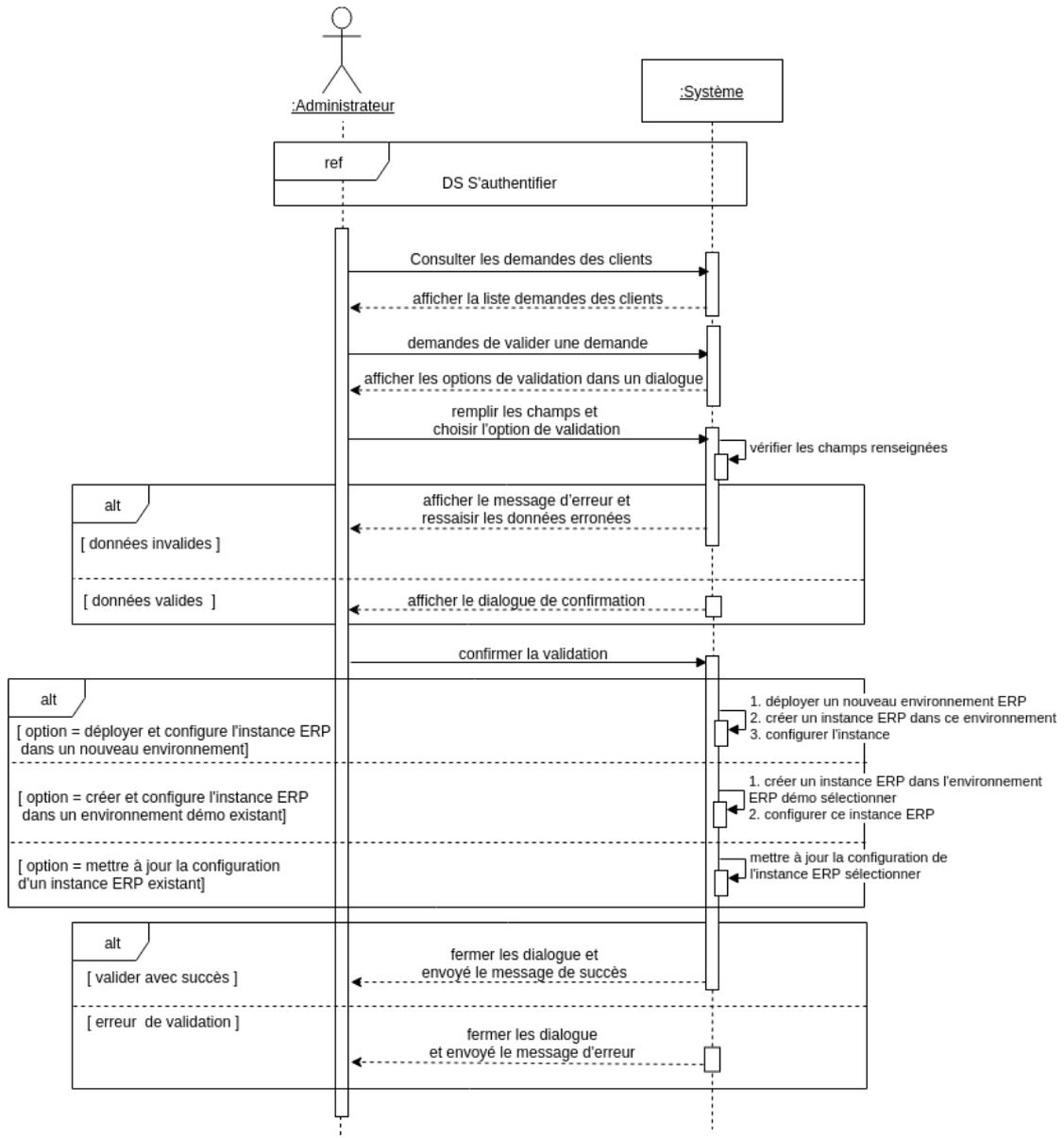


FIGURE 2.5 – Diagramme de séquence « Valider le déploiement et la configuration d'un environnement »

Description textuelle du diagramme de séquence «Valider le déploiement et la configuration d'un environnement»

Acteur : Administrateur.

Précondition : Utilisateur authentifié.

Postcondition : Validation du déploiement et de la configuration d'un environnement client

Description des scénarios :

a) **Scénario nominale :**

1. L'administrateur consulte la liste des demandes des clients.
2. Le système affiche la liste des demandes des clients.
3. L'administrateur demande de valider une demande.
4. Le système affiche les options de validation dans un dialogue.
5. L'administrateur remplit les champs et choisit les options de validation.
6. Le système vérifie les champs renseignés.
7. Le système affiche le dialogue de confirmation
8. L'administrateur confirme la validation
9. Le système exécute les fonctionnalités qui correspond à l'option choisie
10. Le système ferme les dialogues
11. Le système affiche le message de succès

b) **Scénarios alternatifs :**

- 9.a. L'administrateur choisit l'option : déployer et configurer l'instance ERP dans un nouveau environnement de production.
- * Le système déployer un nouveau environnement ERP de production.
 - * Le système créer un instance ERP dans ce environnemnet.
 - * Le système configure ce instance ERP.
 - * Aller à 2.
- 9.b. L'administrateur choisit l'option : créer et configure l'instance ERP dans un environnement démo existant.
- * Le système créer un instance dans l'environnement ERP démo sélectionner.
 - * Le système configure ce instance ERP.
 - * Aller à 2.
- 9.c. L'administrateur choisit l'option : mettre à jour la configuration d'un instance ERP existant.

- * Le système met à jour la configuration de l'instance sélectionnée.
- * Aller à 2.

c) **Scénarios d'exception :**

4.a. L'administrateur a saisi des fausses informations.

- * Le système affiche le message d'erreur.
- * Aller à 2.

7.a. Le système rencontre un problème de déploiement d'environnement ERP ou la création d'instance ou de la configuration.

- * Le système affiche le message d'erreur.
- * Aller à 2.

2.3.2.4 Diagramme de séquence « Traiter les demandes des clients »

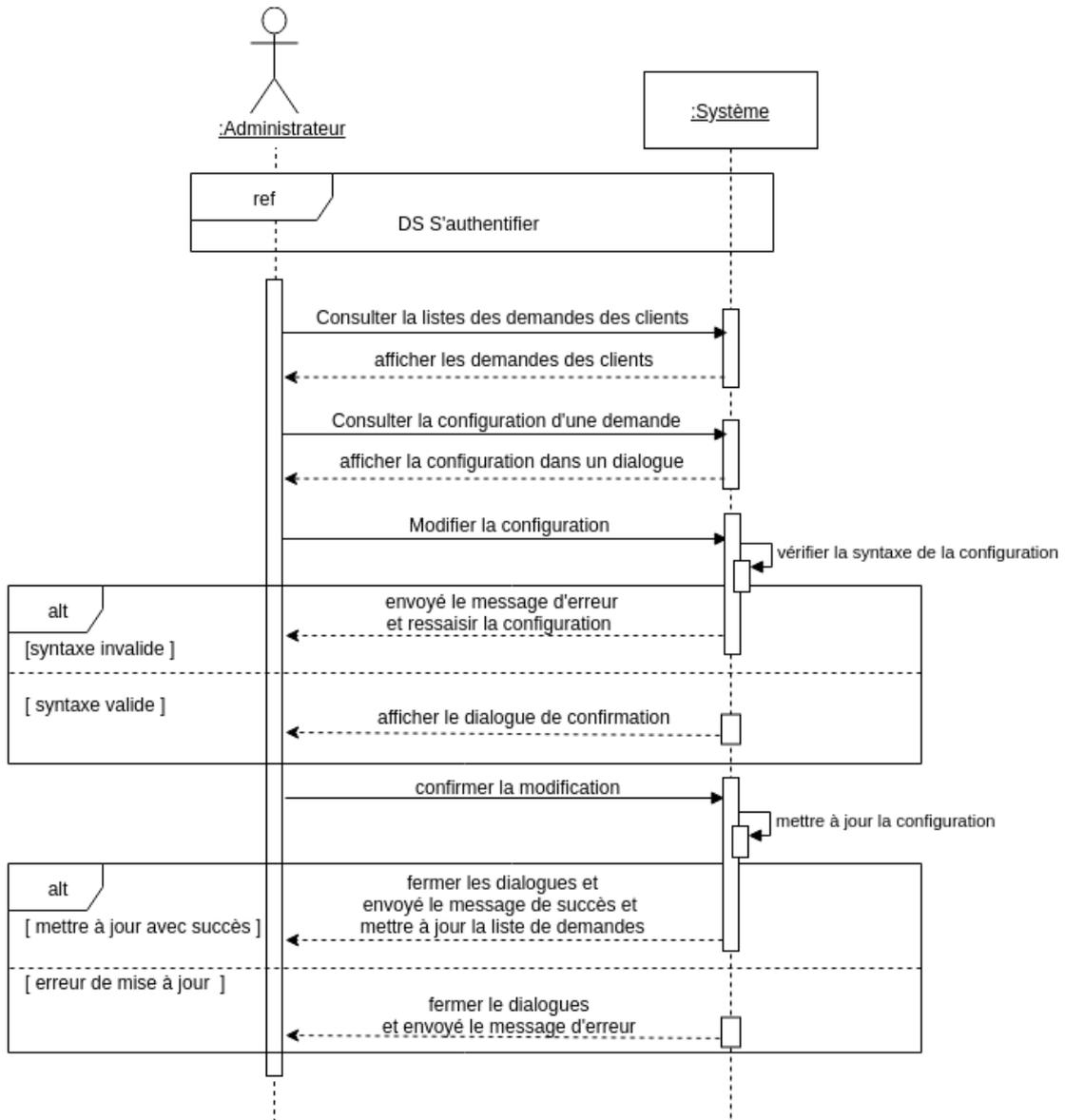


FIGURE 2.6 – Diagramme de séquence « Traiter les demandes des clients »

Description textuelle du diagramme de séquence «Traiter les demandes des clients»

Acteur : Administrateur.

Précondition : Utilisateur authentifié.

Postcondition : Pré-configuration d'un système ERP mis à jour

Description des scénarios :

a) **Scénario nominale :**

1. L'administrateur consulte la liste des demandes des clients.
2. Le système affiche la liste des demandes.
3. L'administrateur consulte la configuration d'une demande.
4. Le système affiche la pré-configuration du client dans un dialogue.
5. L'administrateur modifier la pré-configuration du système ERP.
6. Le système vérifie la syntaxe de la pré-configuration.
7. Le système affiche le dialogue de confirmation de modification.
8. L'administrateur confirme la modification de la pré-configuration.
9. Le système mis à jour la pré-configuration dans la base de données.
10. Le système ferme les dialogues.
11. Le système affiche le message de succès dans un dialogue.
12. Le système recharge la liste des demandes.

b) Scénarios d'exception :

- 6.a. L'administrateur commet une erreur dans la syntaxe de la configuration.
 - * Le système affiche le message d'erreur.
 - * Aller à 4.
- 9.a. Le système rencontre une erreur de mise à jour de la pré-configuration dans la base de données.
 - * Le système affiche le message d'erreur.
 - * Aller à 2.

2.3.2.5 Diagramme de séquence « Ajouter un environnement Demo »

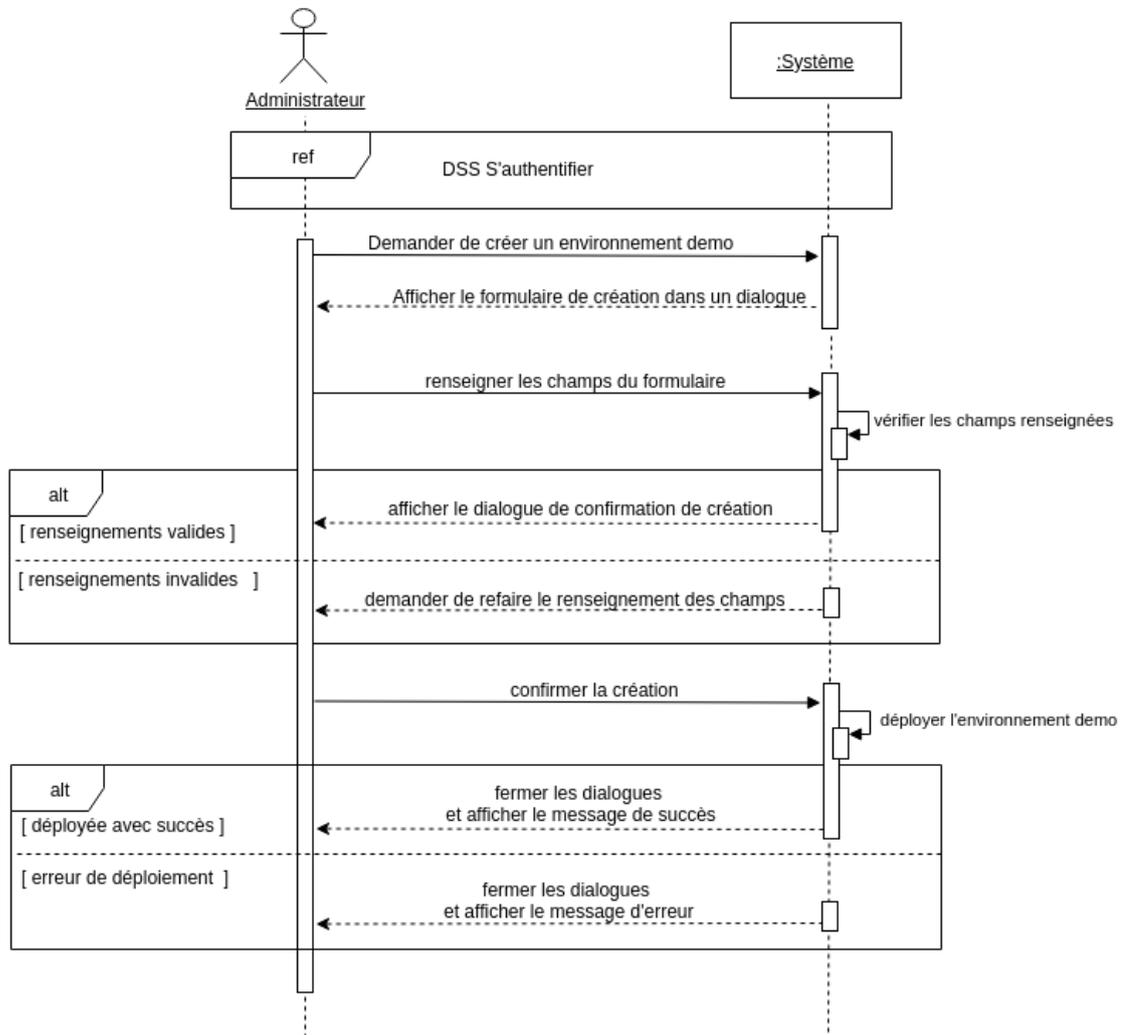


FIGURE 2.7 – Diagramme de séquence « Ajouter un environnement Demo »

Description textuelle du diagramme de séquence «Ajouter un environnement Demo»

Acteur : Administrateur.

Précondition : Utilisateur authentifié.

Postcondition : Création d'un environnement odoo pour le déploiements des instances démonstratifs du système ERP

Description des scénarios :

a) **Scénario nominale :**

1. L'administrateur demande la création d'un environnement demo.

2. Le système affiche le formulaire de création dans un dialogue.
3. L'administrateur renseigne les champs et envoie le formulaire
4. Le système vérifie les champs renseignés.
5. Le système affiche le dialogue de confirmation de création.
6. L'administrateur confirme la création de l'environnement demo.
7. Le système déploie un nouveau environnement demo et stocke ses renseignements dans la base de données.
8. Le système ferme les dialogues.
9. Le système affiche le message de succès dans un dialogue.

b) Scénarios d'exception :

- 4.a. L'administrateur a saisi des fausses renseignements.
 - * Le système affiche le message d'erreur.
 - * Aller à 2.
- 7.a. Le système rencontre un problème de déploiement.
 - * Le système affiche le message d'erreur.
 - * Aller à 1.

2.3.2.6 Diagramme de séquence « Gérer les environnements déployés »

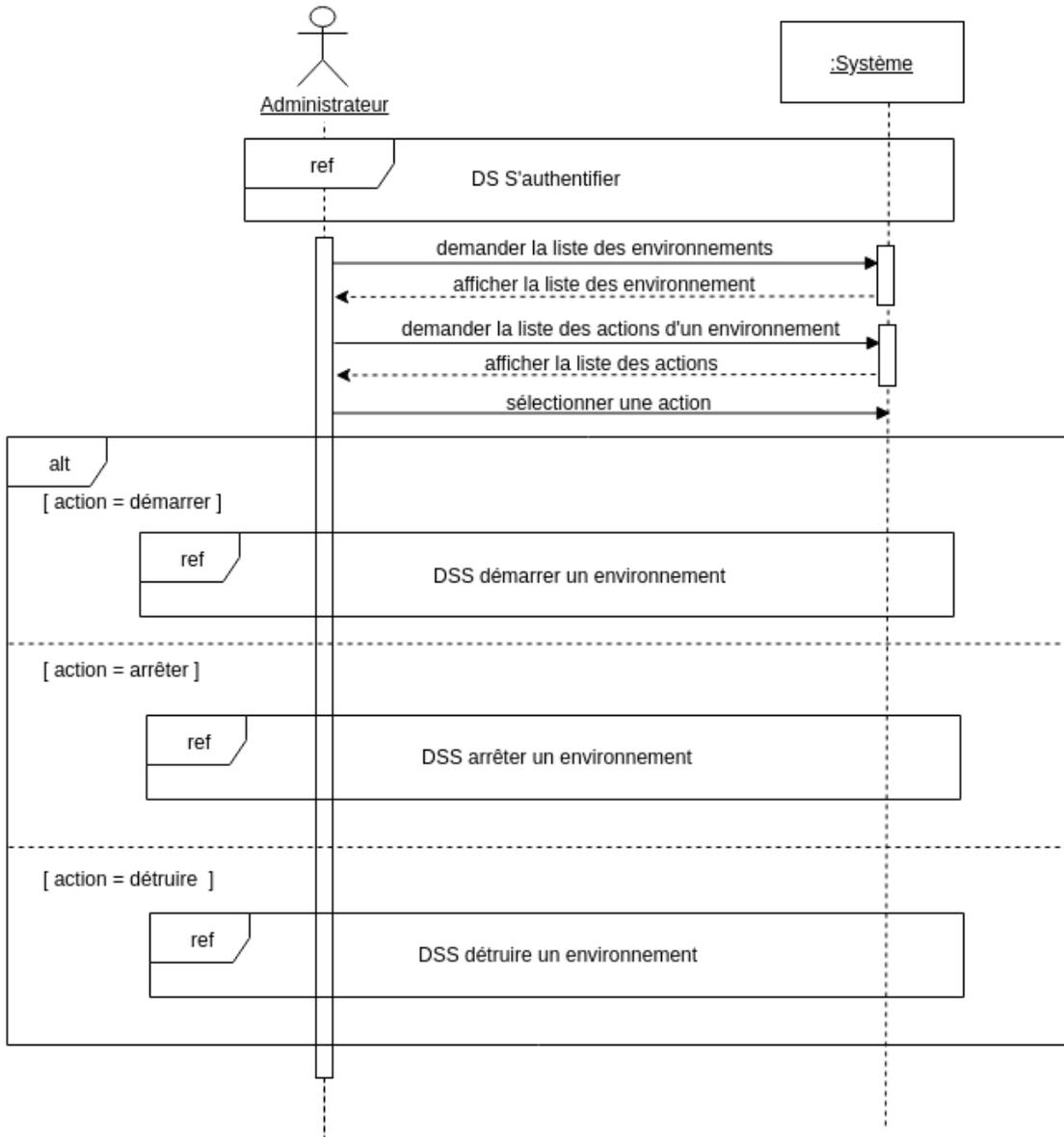


FIGURE 2.8 – Diagramme de séquence « Gérer les environnements déployés »

Description textuelle du diagramme de séquence «Gérer les environnements déployés»

Acteur : Administrateur.

Précondition : Utilisateur authentifié.

Postcondition : Démarrer, arrêter ou détruire un environnement.

Description des scénarios :

a) **Scénario nominale :**

1. L'administrateur demande la liste des environnements.
2. Le système affiche la liste des environnements.
3. L'administrateur demande la liste des actions d'un environnement.
4. Le système affiche la liste des actions.
5. L'administrateur sélectionne une action
6. Le système effectuer l'action sélectionner.

b) **Scénarios alternatifs :**

6.a. L'administrateur choisit l'action "démarrer l'environnement".

- * Le système fait appel au cas d'utilisation «démarrer un environnement»
- * Aller à 2.

6.b. L'administrateur choisit l'action "arrêter l'environnement".

- * Le système fait appel au cas d'utilisation «arrêter un environnement»
- * Aller à 2.

6.c. L'administrateur choisit l'action "détruire l'environnement".

- * Le système fait appel au cas d'utilisation «détruire un environnement»
- * Aller à 2.

2.3.3 Diagramme de classe

Nous montrons dans la figure ci-dessous le diagramme de classes qui représente la structure statique de notre système.

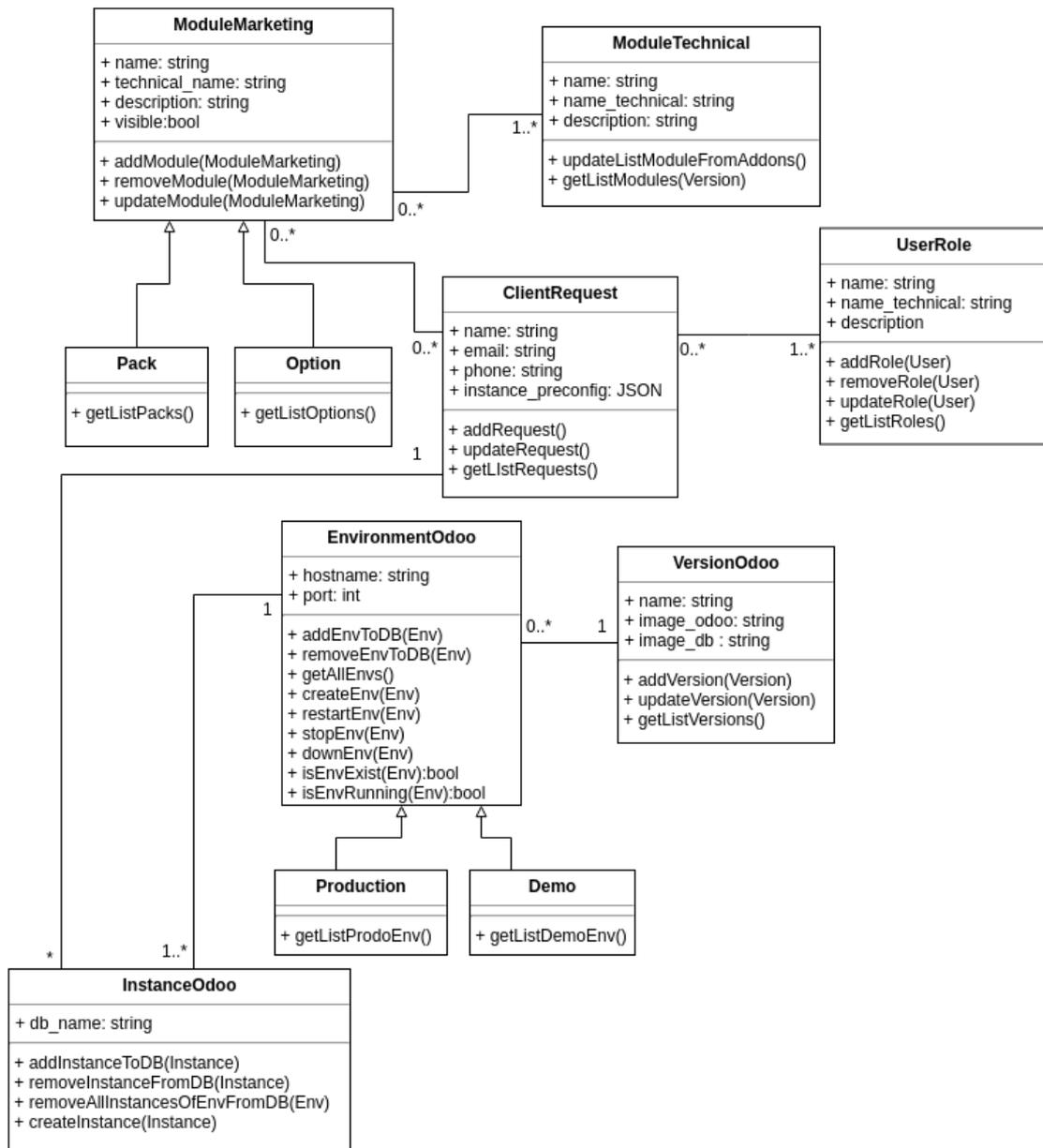


FIGURE 2.9 – Diagramme de classe «application ODOAAS»

2.4 Conclusion

Au cours de ce chapitre, nous avons détaillé notre projet en faisant une analyse profonde de notre système, qui consiste à identifier et décrire les besoins fonctionnels et non fonctionnels, et l'explication des rôles des utilisateurs, ensuite nous avons fait une étude conceptuelle en réalisant les différents diagrammes qui donnent une vision plus claire sur notre système, en utilisant le langage UML.

Nous avons commencé par la vue fonctionnelle du système en réalisant les diagrammes de cas d'utilisation, ensuite les diagrammes de séquence qui illustrent la vue comportementale du système, enfin le diagramme de classe qui représente la structure de notre système.

Dans le chapitre qui suit, nous allons entamer la phase de l'implémentation du projet.

Chapitre 3

Implémentation du système «ODOaaS»

3.1 Introduction

Dans ce dernier chapitre, nous détaillons la phase d'implémentation de notre application web par ses différentes étapes. En premier, nous commencerons par la présentation de la méthode adoptée pour le suivi et la gestion de notre projet, Ensuite, nous allons décrire les technologies et outils utilisés pour la réalisation de notre projet, Enfin, nous allons terminer par la présentation de l'architecture de l'application et ses maquettes.

3.2 Méthode de gestion de projet

3.2.1 Méthode Scrum

Pour la gestion du processus de développement de notre application web, nous avons adopté une méthode agile, qui s'appelle Scrum, pour s'assurer une livraison d'un produit de bon qualité.

Scrum est un cadre de travail simple pour une collaboration d'équipe efficace sur des produits complexes. Durant la réalisation du projet, le «**Scrum Master**» qui garantit l'application de la méthodologie Scrum, effectue une réunion de planification avant chaque itération «**Sprint Planification**», dans cette réunion l'équipe scrum «**Scrum Team**» commence par sélectionner les éléments prioritaires du «**Product Backlog**». Le Product Backlog est la liste ordonnée des exigences du projet qu'elle pense pouvoir réaliser au cours du sprint en accord avec «**Product Owner**» qui porte la vision du produit à réaliser (représentant généralement le client).[28]

Une itération «**Sprint**» est constituée d'un ensemble d'exigence appelé histoire d'utilisateur «**User Story** »,et d'un «**Sprint Backlog**». Le Sprint Backlog contient une liste des tâches à réaliser durant une itération.[28]

Au cours du sprint, une réunion de synchronisation de l'équipe de développement qui se fait quotidiennement «**Daily meeting**», 15 minutes maximum au cours desquelles chacun répond principalement à 3 questions[28] :

- Qu'ai-je terminé depuis la dernière mêlée ?
- Qu'aurai-je terminé d'ici la prochaine mêlée ?
- Quels obstacles me retardent ?



FIGURE 3.1 – Processus de Scrum[29]

Nous avons organisé notre projet en 4 itérations, à la fin de chaque itération, les fonctionnalités réalisées seront prêt à livrer et d'être exploitable par le client.

Nous présentons dans la figure suivante le contenu des itérations.



FIGURE 3.2 – Les itérations du projet «ODOaaS»

Afin d'avoir un aperçu complet du projet et le gérer d'une façon optimale, nous avons opté pour l'utilisation de l'outil web Trello pour mieux suivre le déroulement des tâches de chaque itération, en les mobilisant selon leur état d'avancement dans des listes ordonnées qui sont comme suit :

- **TODO** : c'est la liste qui contient les tâches à réaliser durant le sprint, s'appelle le Sprint Backlog.
- **In Progress** : c'est la liste qui contient les tâches en cours de réalisation.
- **To Test** : c'est la liste qui contient les tâches à tester.
- **Done** : c'est la liste qui contient les tâches déjà réalisées et validées par les testeurs.

Nous montrons dans la figure suivante un aperçu du Sprint 1.

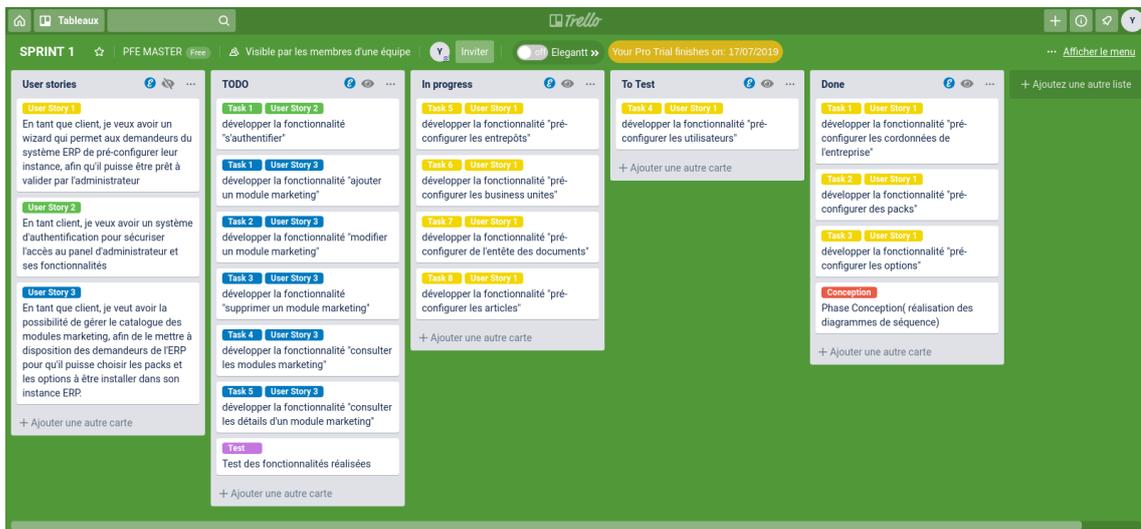


FIGURE 3.3 – Listes des Tâches «ODOaaS»

3.2.2 Diagramme de Gantt

Pour avoir une vue graphique sur l'avancement du projet, nous avons réalisé un diagramme de Gantt en utilisant le plugin «Elegantt» de l'outil Trello.

Nous montrons dans la figure ci-dessous le diagramme de Gantt du Sprint 1.

3.3 Technologies utilisées

3.3.1 Framework Django

Nous avons choisit d'utiliser pour le développement de notre application web pour la partie back-end(côté serveur) la technologie «Django», qui est un framework python qui permet de créer des applications web.



Le but principal de ce framework est de rendre le développement web plus simple et rapide, et d'avoir un code bien structuré et claire, il fournit presque tous les outils nécessaires à sécuriser notre application, à gérer la structure de nos modèles avec un ORM et d'autres librairies. Django s'inspire du patron MVC, et cette structuration vise à séparer les données (models) du traitement(controller), qui sont eux-mêmes séparés de la vue (view/template)[4].

3.3.2 Framework Vuejs

Pour la partie front-end(côté client) de notre application nous avons choisit d'utiliser la technologie «Vuejs», qui est un framework javascript, et celui-ci a été créé par Evan You en 2014. La particularité de ce genre de framework est que tout code HTML, CSS, JavaScript est exécuté directement par Javascript et cela offre alors la possibilité d'optimiser les performances de notre application web.[5]



VueJs se veut être un framework visant à simplifier et organiser le développement d'applications web en implémentant un bon nombre de fonctionnalités telles que les : Templates, Composants, Transitions,Routage, ...Etc.[5]

3.3.3 ERP Odoo

Le système ERP Odoo est la partie essentielle de notre projet, où nous avons automatisé son déploiement dans le cloud et sa configuration grâce à notre application web que nous avons développée.



Odoo, anciennement OpenERP et Tiny ERP est initialement un progiciel open-source de

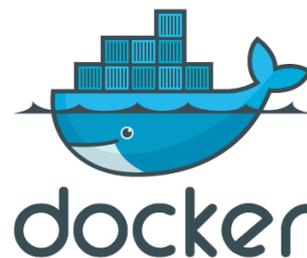
gestion intégré comprenant de très nombreux modules permettant de simplifier la gestion d'entreprise dans son ensemble. Le logiciel est utilisé par plus de deux millions d'utilisateurs pour gérer leurs entreprises à travers le monde.[2]

3.3.3.1 Odoo API

Odoo est étendu de l'extérieur, où de nombre de ses fonctionnalités et de ses données sont disponibles pour une analyse ou une intégration avec divers outils. Une partie de l'API de référence de modèle est facilement disponible via XML-RPC et accessible à partir de plusieurs langages de programmation,[6] nous avons utilisé ce API dans notre projet pour automatisé la configuration de nos instance du système ERP, aussi nous avons utilisé l'outil OdooRPC, qui est un package Python offrant un moyen facile de piloter nos environnements client via RPC.[7]

3.3.4 Docker

Docker est un outil conçu pour faciliter la création, le déploiement et l'exécution d'applications à l'aide de conteneurs isolés. Le conteneur permet d'empaqueter une application avec tous les dépendances et les bibliothèques dont il a besoin.[8] Docker utilise LXC, cgroups et le noyau Linux, avec une API de haut niveau fournissant une solution pratique de virtualisation qui exécute les processus de façon isolée, il s'appuie sur les fonctionnalités du système d'exploitation fournies par la machine hôte, Cette approche permet d'accroître la flexibilité et la portabilité d'exécution d'une application, laquelle va pouvoir tourner sur une grande variété de machines hôtes.[12]



3.3.4.1 Docker SDK

Docker SDK est une bibliothèque Python pour l'API Docker Engine, permet de faire tout ce que fait la commande docker comme la gérance des conteneurs, volumes, réseaux, images etc.[10]

3.3.4.2 Docker Compose

Docker Compose est un outil permettant de définir et d'exécuter des applications Docker à conteneurs multiples. Avec Compose, vous utilisez un fichier YAML pour configurer les services de votre application. Ensuite, avec une seule commande, vous créez et démarrez tous les services de votre configuration.[11]

3.3.5 Git

Git est un système de contrôle de version distribué permettant de suivre les modifications du code source au cours du développement du logiciel. Il est conçu pour coordonner le travail des programmeurs, mais il peut être utilisé pour suivre les modifications dans n'importe quel ensemble de fichiers. Ses objectifs incluent la vitesse, l'intégrité des données et la prise en charge de flux de travail distribués non linéaires.[13]



3.3.6 Latex

LaTeX est un langage de description donnant à l'auteur les moyens d'obtenir des documents mis en page de façon professionnelle sans avoir à se soucier de leur forme. La priorité est donnée à l'essentiel : le contenu.[14] Pour l'éditeur de code de latex, nous avons utilisé TexStudio pour la rédaction de notre mémoire



3.4 Outils utilisées

3.4.1 draw.io

Draw.io est un outil web de création de diagrammes les plus riches en fonctionnalités. Ses nombreuses barres d'outils, réunissant des centaines d'éléments visuels et de fonctions, per-



mettent de créer des diagrammes pour toutes les applications imaginables. Positionnement libre des formes et objets sur la grille, déplacement, redimensionnement et modification de ceux-ci en fonction des besoins. Il est aussi compatible avec Google Drive et entièrement gratuite, permettant de dessiner :Des organigrammes, Des diagramme UML, Des schémas réseaux, Des circuits électroniques, Maquetage d'interface, Etc.[15]

3.4.2 Trello

Trello est un outil de gestion de projet en ligne gratuit, Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre.[16]



3.4.3 Github

GitHub est un outil en ligne gratuit pour héberger de répertoires de code open source placés sous le contrôle de Git, proposant de nombreuses fonctionnalités, à titre d'exemple : Communiquer avec d'autres développeurs et signaler des problèmes de code en déclarant des "issues", Proposer des modifications de code à d'autres repos en faisant des "pull requests", Etc.[18].



3.4.4 Postman

Postman est un outil qui permet de tester des web services et API. Il permet aussi de construire et d'exécuter des requêtes HTTP, de les stocker dans un historique afin de pouvoir les rejouer.[17]



3.4.5 Sublime

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Il intègre la plupart des fonctionnalités de base d'un éditeur de texte, dont la coloration syntaxique personnalisable, l'auto complétion, un système de plugins, et d'autres fonctions plus avancées.[19]



3.4.6 PhpMyAdmin

phpMyAdmin est un logiciel gratuit écrit en PHP, destiné à gérer l'administration de MySQL sur le Web. phpMyAdmin prend en charge un large éventail d'opérations sur MySQL et MariaDB. Les opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des autorisations, etc.) peuvent être effectuées via l'interface utilisateur, tout en vous permettant d'exécuter directement toute instruction SQL.[20]



3.4.7 MySQL Workbench

MySQL Workbench est un outil visuel unifié pour les architectes de bases de données, MySQL Workbench fournit des outils de modélisation de données, de développement SQL et d'administration complète pour la configuration du serveur.[21]



3.5 L'architecture de l'application

Nous allons présenter dans cette partie l'architecture du système «ODOaaS». Notre système est composé de deux applications : vuejs et django :

- **Application Frontend** : cette application basé sur le framework javascript vuejs qui represente la partie frontale du système, visible par les utilisateurs via le navigateur web. Ce type d'application conçu en tant qu'une SPA (Single Page Application), c'est à dire le code d'application(HTML,CSS, Javascript) se charge qu'une seule fois dans une page HTML, et lorsque les utilisateurs interagissent avec l'application en demandant du JSON ou effectuent une action sur le serveur, Javascript intercepte les évènements du navigateur et réécrit de manière dynamique la page en cours, plutôt que de charger de nouvelles pages entières à partir d'un serveur.[30]
- **Application Backend** : cette application est basée sur le framework python django qui represente la couche d'accès aux données et traitement des requêtes des utilisateurs. Cette partie est invisible pour les utilisateurs. Lorsqu'un utilisateur appelle un service de l'application Django, le framework se charge d'exécuter la vue correspondante via les règles de routage URL définies. Une vue est responsable de récupérer les données des modèles et de les envoyer en réponse à l'utilisateur sous format JSON, afin d'être afficher sur le navigateur web par l'intermédiaire de l'application VueJS.

Nous montrons dans la figure suivante l'architecture de notre application expliquée au-dessus.

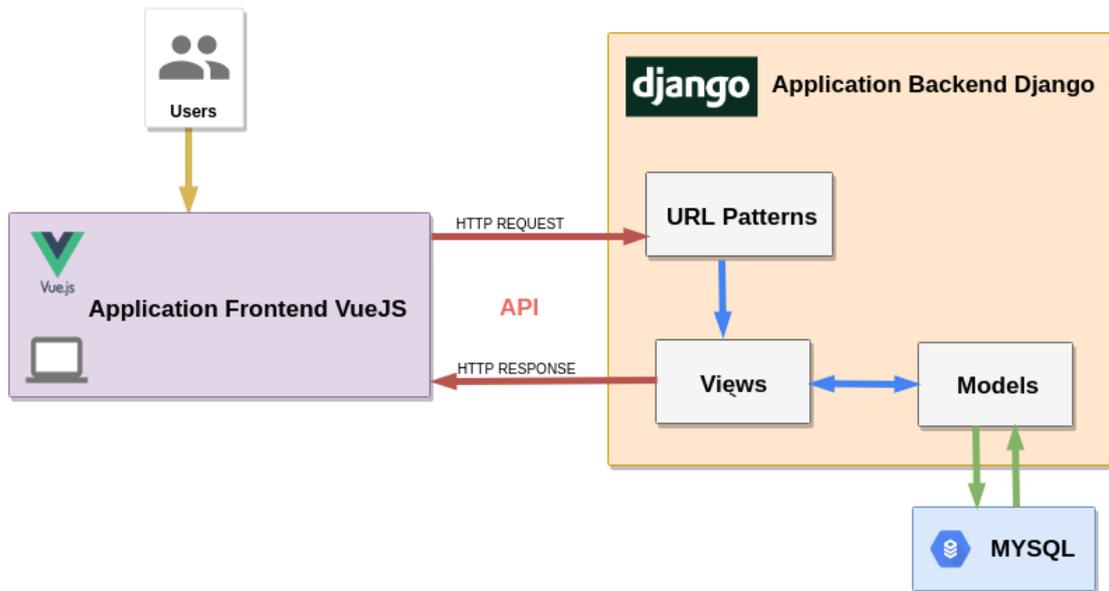


FIGURE 3.5 – Architecture de l’application «ODOaaS»

3.6 Maquettes de l’application

Dans cette partie, nous allons présenter les maquettes les plus importantes qui illustrent les fonctionnalités globales de notre système «ODOaaS» que nous avons déjà vues dans le chapitre précédent.

3.6.1 Pré-configuration du système ERP

Les figures suivantes représentent l’interface de wizard qui contient un ensemble de formulaires. Ce wizard permet aux clients de pré-configurer leur système ERP. Nous présentons deux formulaires à titre d’exemple : le formulaire de pré-configuration des coordonnées de l’entreprise et le formulaire de pré-configuration des utilisateurs du système ERP.

La figure suivante représente l'interface qui permet aux clients de pré-configurer les coordonnées de leur entreprise.

The screenshot shows the ODOaaS demo interface for configuring company coordinates. The header includes the ODOaaS logo, a search bar, and navigation links for Demo, A Propos, and Contact. The main content area is titled "DEMO DE INSIDJAM ERP" and contains a message: "Vous pouvez avoir une version demo de la solution INSIDJAM ERP en remplissant ces formulaires." Below this message is a progress bar with 9 steps: 1. Coordonnées de l'Entreprise (highlighted), 2. Choix des Packs, 3. Choix des Options, 4. Utilisateurs, 5. Entrepôts, 6. Unités de business, 7. Articles, 8. Entêtes de documents, and 9. Validation. The form fields are as follows:

Nom de la société	Site Web
<input type="text" value="It solutions"/>	<input type="text" value="https://itsolutions.com"/>
Adresse	Téléphone
<input type="text" value="rue boumerdes"/>	<input type="text" value="0556360146"/>
<input type="text" value="Rue 2..."/>	Courriel
Ville	<input type="text" value="itsolutions@gmail.com"/>
Wilaya	Registre de la société
<input type="text" value="boumerdes"/>	<input type="text" value="12B4585655"/>
Code postal	
<input type="text" value="16445"/>	

At the bottom right of the form is a "Suivant" button. At the bottom of the page, the footer reads: "2019 © ODOaaS, - www.odoaas.com - developed By Yahia ZEGHBA".

FIGURE 3.6 – Interface de pré-configuration des Coordonnées de L'entreprise

La figure suivante représente l'interface qui permet aux clients de pré-configurer les utilisateurs de leur système ERP.

ODOaasS

Search...

Demo A Propos Contact

DEMO DE INSIDJAM ERP

Vous pouvez avoir une version demo de la solution **INSIDJAM ERP** en remplissant ces formulaires.

1 2 3 4 5 6 7 8 9

Coordonnées de l'Entreprise Choix des Packs Choix des Options **Utilisateurs** Entrepôts Unités de business Articles Entêtes de documents Validation

Nom

Email

Rôles

Administrateur Fonctionnaire

Ajouter

Import un fichier des utilisateurs

Choose File No file chosen Importer

Précédent Suivant

Liste des Utilisateurs

Nom	Email	#	#
ZEGHBA Yahia	zeghbayahia@gmail.com		

2019 © ODOaasS. - www.odoaas.com - developed By Yahia ZEGHBA

FIGURE 3.7 – Interface de pré-configuration des Utilisateurs

3.6.2 Authentification

La figure suivante représente l'interface d'authentification qui permet l'accès au panel d'administration du système «ODOaaS»

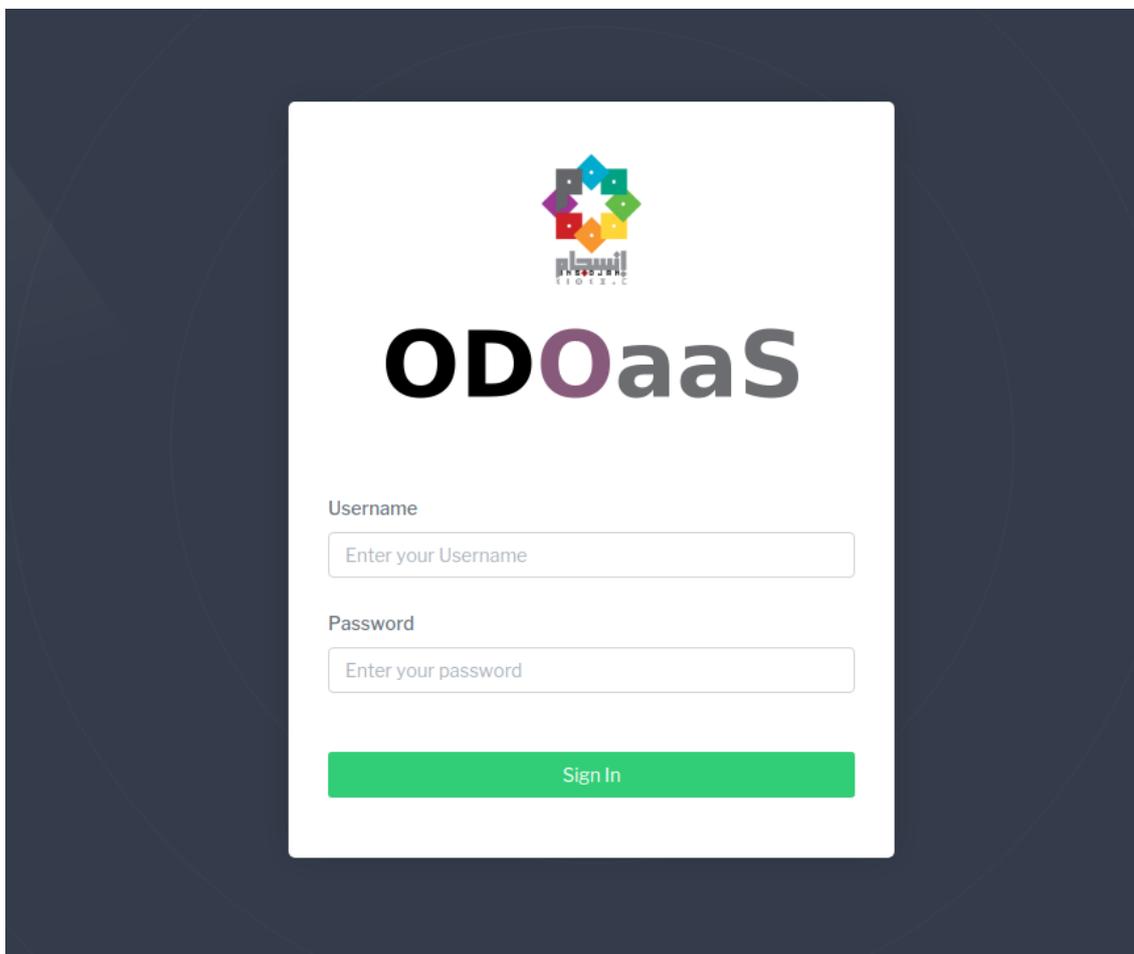
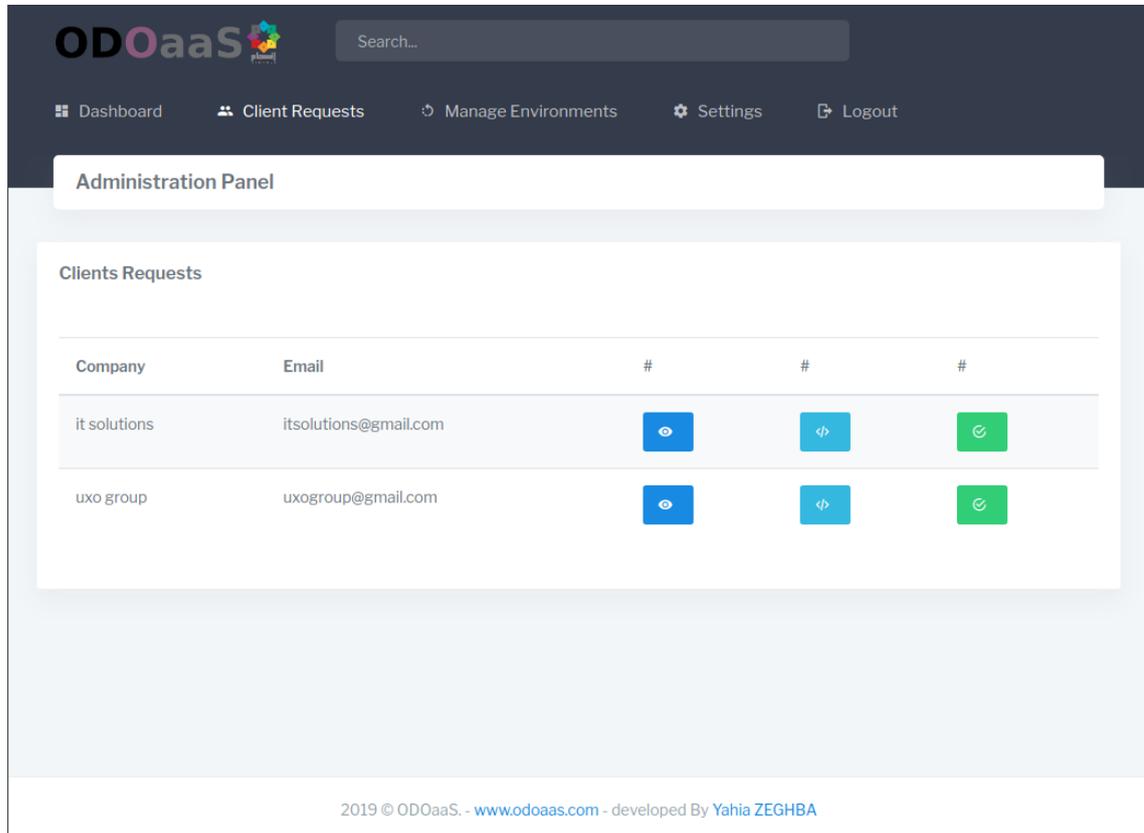


FIGURE 3.8 – Interface d'authentification au système «ODOaaS»

3.6.3 Gérer les demandes des clients

La figure suivante représente l'interface qui permet à l'administrateur de visualiser les demandes des clients, afin de les traiter.



The screenshot displays the ODOaaS Administration Panel. At the top, there is a search bar and a navigation menu with options: Dashboard, Client Requests, Manage Environments, Settings, and Logout. Below the navigation, the 'Administration Panel' is titled, and a section for 'Clients Requests' is shown. This section contains a table with two rows of client data. Each row includes the company name, email address, and three action buttons: a blue eye icon, a blue left-pointing arrow, and a green circular refresh icon.

Company	Email	#	#	#
it solutions	itsolutions@gmail.com			
uxo group	uxogroup@gmail.com			

2019 © ODOaaS. - www.odoaas.com - developed By Yahia ZEGHBA

FIGURE 3.9 – Interface de gestion des demandes des clients

La figure suivante représente l'interface qui permet à l'administrateur de personnaliser une pré-configuration d'un client.

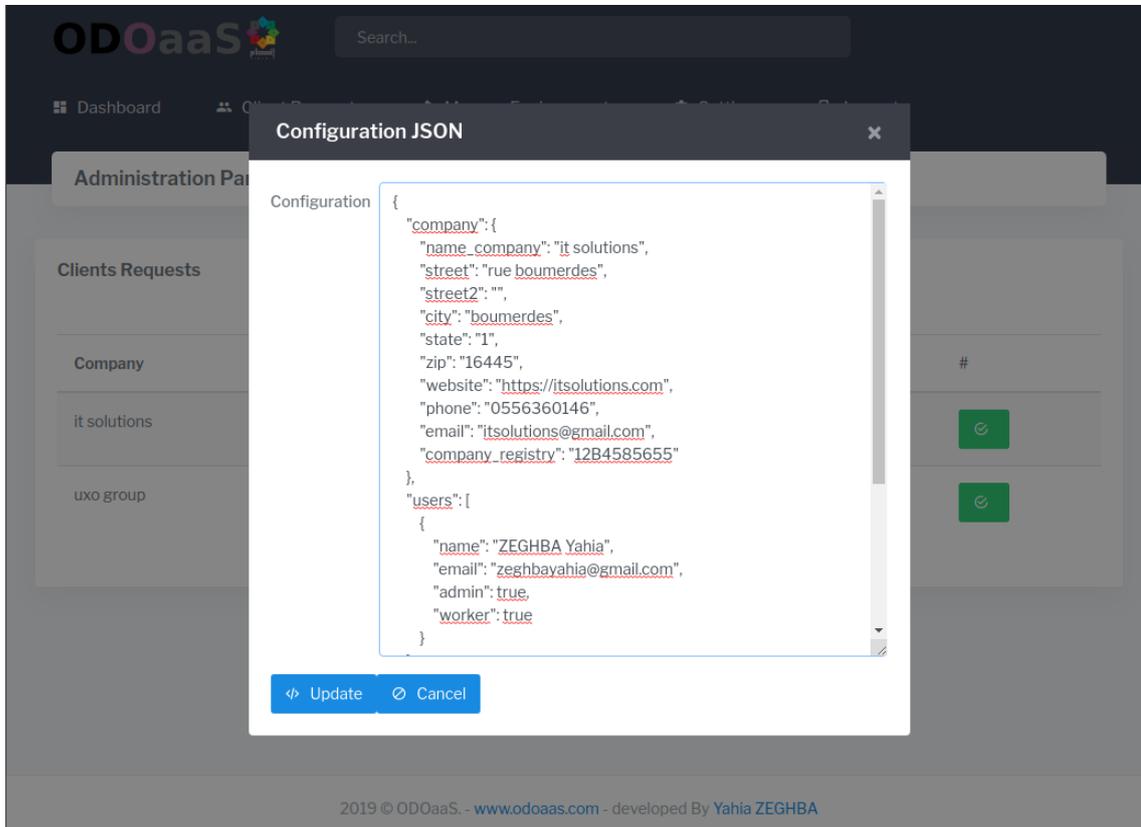


FIGURE 3.10 – Interface de personnalisation de la pré-configuration

La figure suivante représente l'interface qui permet à l'administrateur de valider le déploiement et la configuration d'un instance ERP.

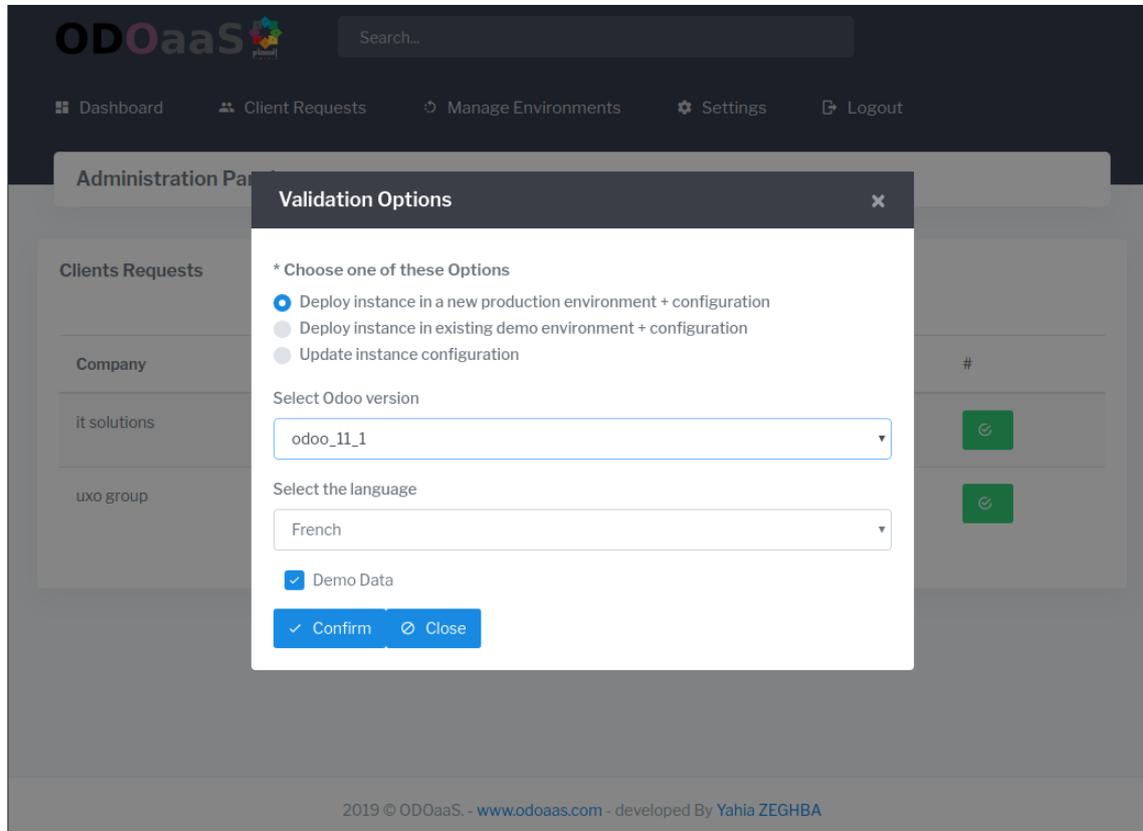
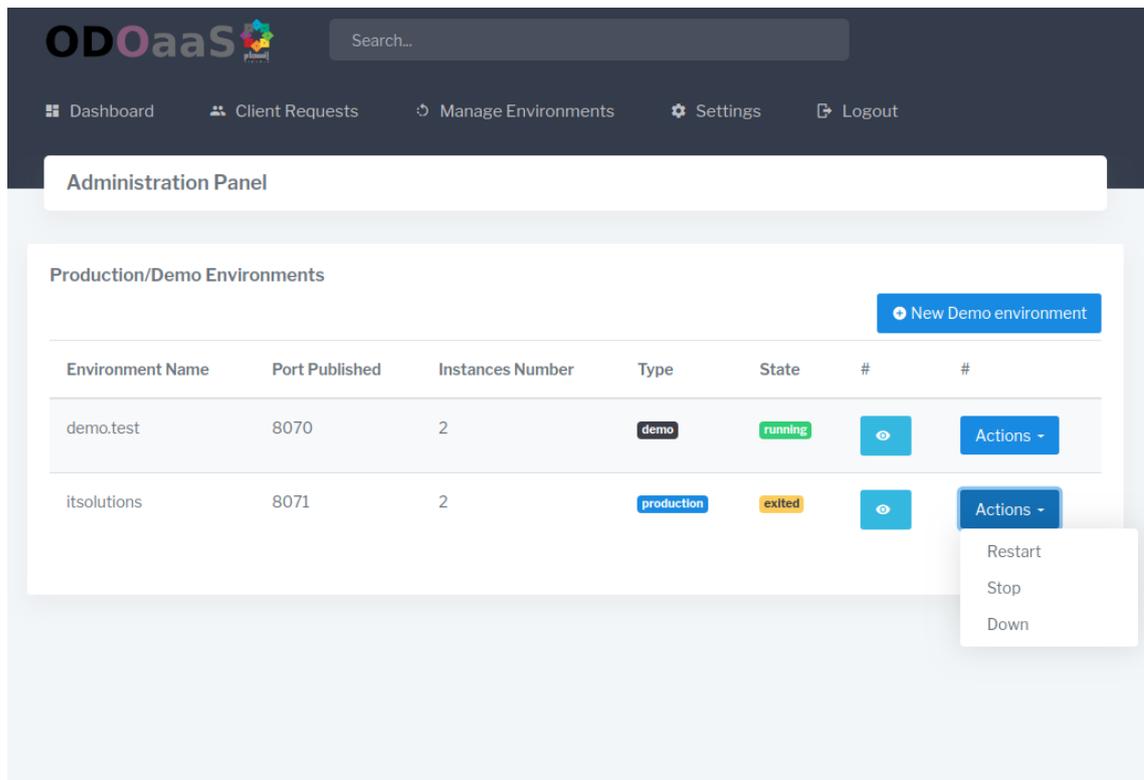


FIGURE 3.11 – Interface de validation de déploiement et configuration d'un ERP

3.6.4 Gérer les environnements déployés

La figure suivante représente l'interface qui permet à l'administrateur de gérer les environnements déployés, en exécutant l'opération démarrer, stopper ou détruire un environnement.



The screenshot displays the ODOaaS Administration Panel. At the top, there is a search bar and navigation links for Dashboard, Client Requests, Manage Environments, Settings, and Logout. Below this is the Administration Panel header. The main content area is titled 'Production/Demo Environments' and features a 'New Demo environment' button. A table lists the following environments:

Environment Name	Port Published	Instances Number	Type	State	#	#
demo.test	8070	2	demo	running		Actions -
itsolutions	8071	2	production	exited		Actions -

The 'Actions' dropdown menu for the 'itsolutions' environment is open, showing the following options: Restart, Stop, and Down.

2019 © ODOaaS. - www.odoaas.com - developed By Yahia ZEGHBA

FIGURE 3.12 – Interface de gestion des environnements déployés

La figure suivante représente l'interface qui permet à l'administrateur de déployer un environnement odoo de type démo.

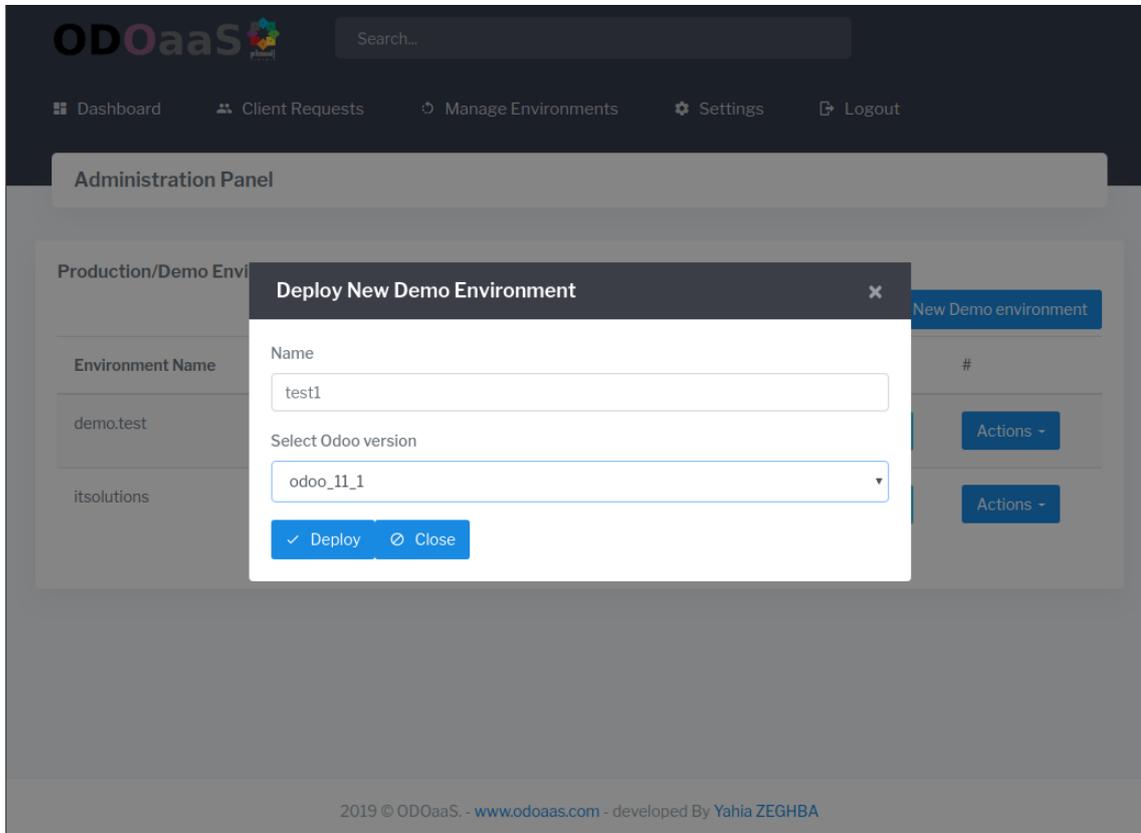


FIGURE 3.13 – Interface de déploiement d'un nouveau environnement odoo de type démo

3.6.5 Gérer les modules marketing

La figure suivante représente l'interface qui permet à l'administrateur de gérer des packs et des options, en effectuant les différentes opérations : ajout, suppression, modification et voir les détails d'un module marketing.

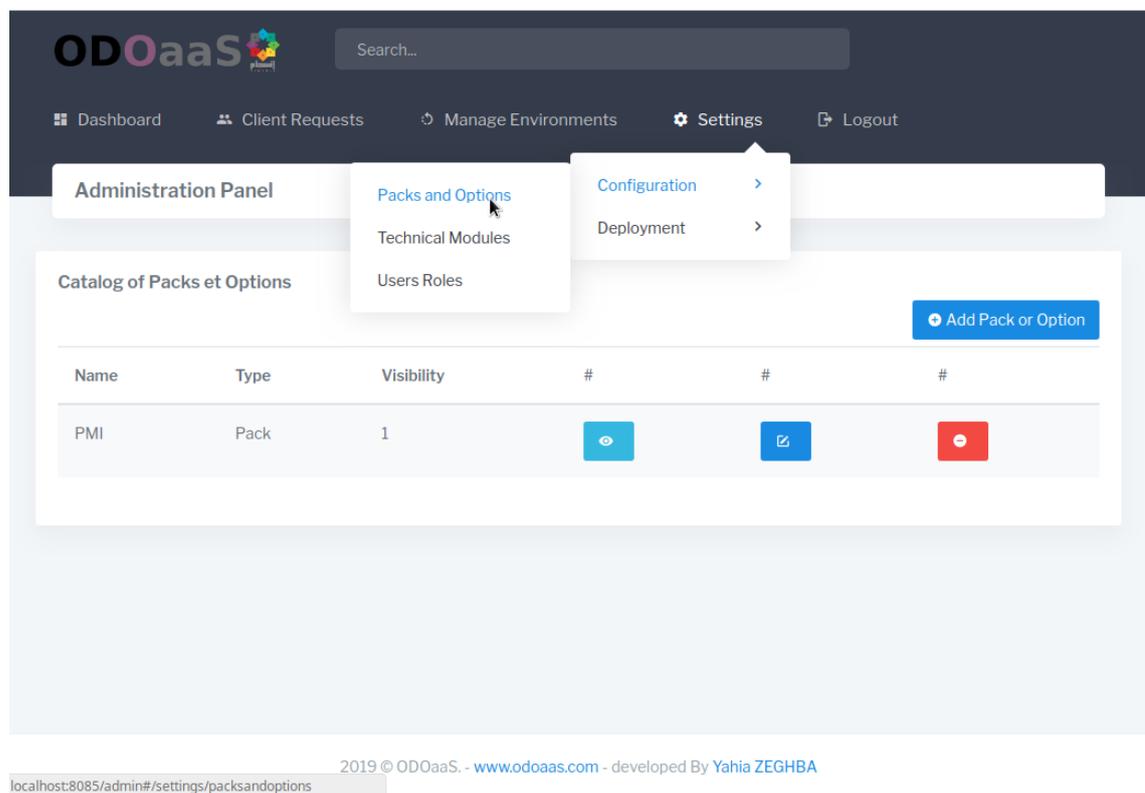


FIGURE 3.14 – Interface de gestion des modules marketing

3.6.6 Gérer les modules techniques

La figure suivante représente l'interface qui permet à l'administrateur de gérer des modules techniques, en effectuant les différentes opérations : ajout, suppression, modification et voir les détails d'un module technique.

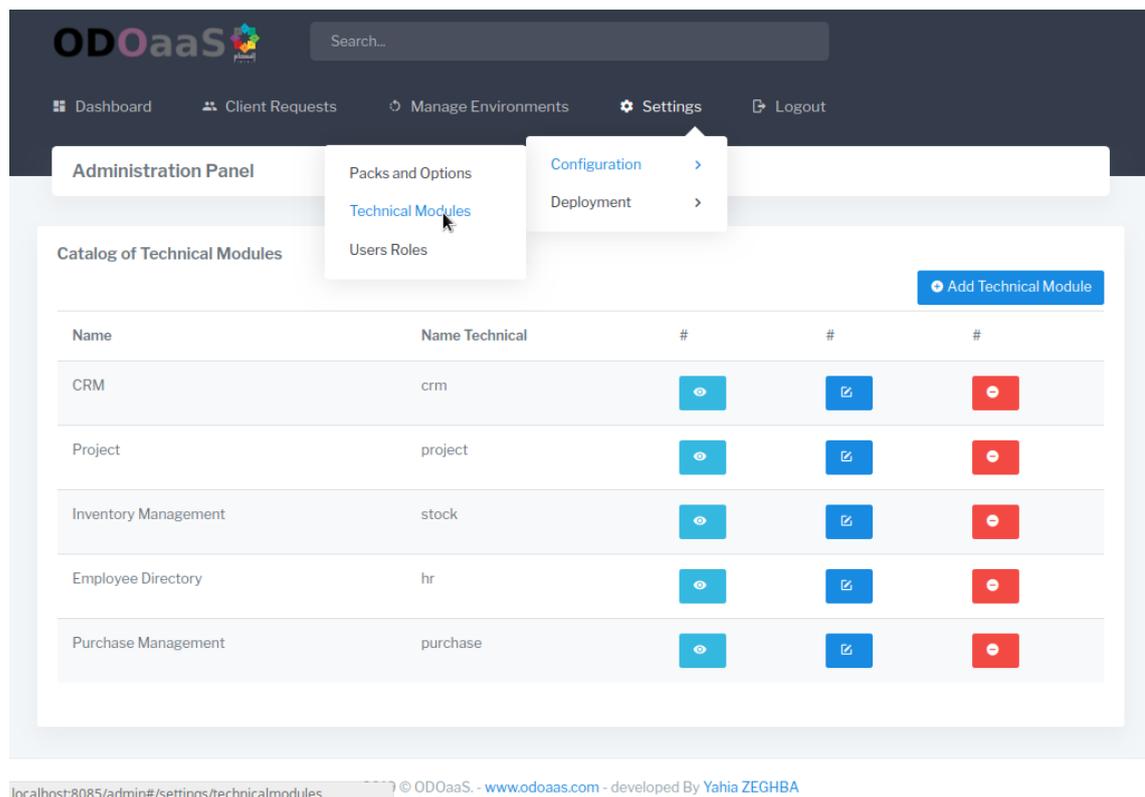


FIGURE 3.15 – Interface de gestion des modules techniques

3.6.7 Gérer les versions d'odoo

La figure suivante représente l'interface qui permet à l'administrateur de gérer les versions odoo, en effectuant les opérations suivantes : ajout, suppression, modification, et voir les details d'une version odoo

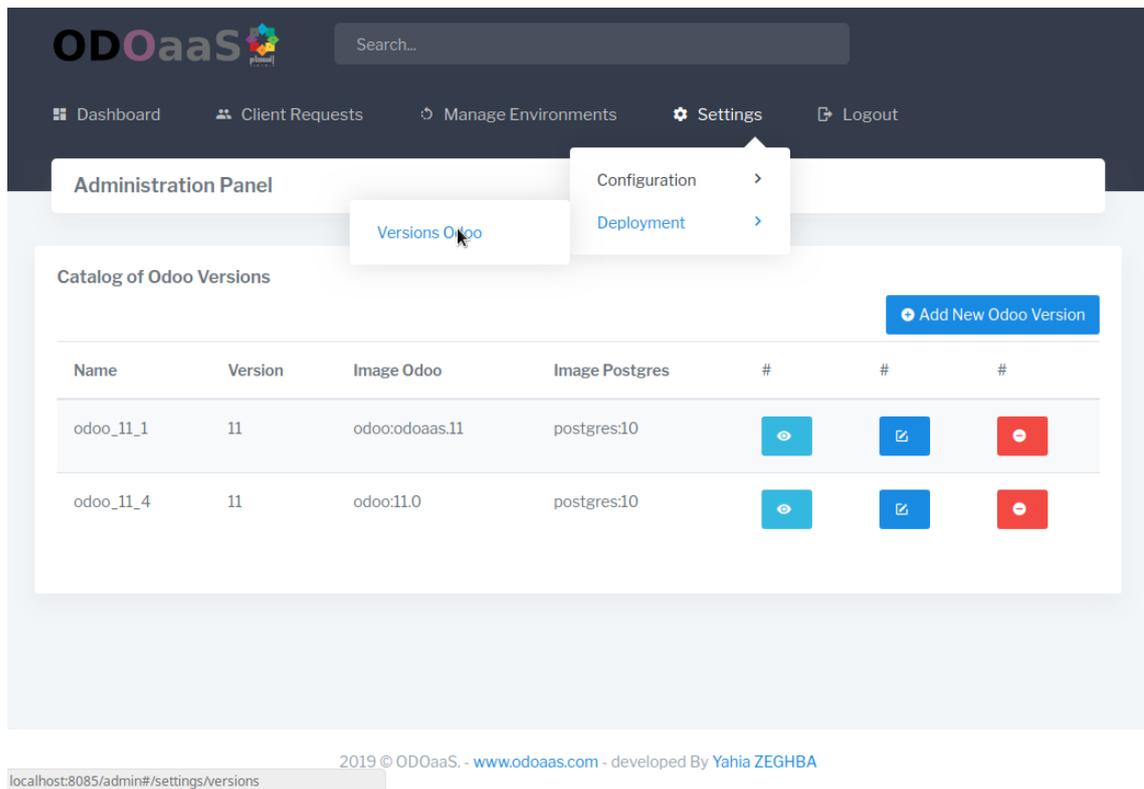


FIGURE 3.16 – Interface de gestion des versions odoo

3.7 Conclusion

Durant ce dernier chapitre, nous avons décrit la méthode adoptée pour la gestion de notre projet. Ensuite, nous avons présenté les technologies et les outils utilisés pour la réalisation de notre solution. Enfin, nous avons terminé par la présentation de l'architecture et les maquettes de notre application.

Conclusion générale et perspectives

Ce projet de fin d'étude avait pour objectif d'implémenter l'application web «ODOaaS», qui permet d'automatiser le déploiement et la configuration d'un système ERP dans le cloud. Nous avons développé ce projet lors d'un stage qui s'est déroulé au sein de l'entreprise IT Solutions.

Nous avons établi une étude comparative de plusieurs systèmes existants, afin d'arriver à cadrer et préciser les finalités de notre système, et déterminer les fonctionnalités importantes à implémenter. En suite, nous avons entamé la phase d'analyse et conception de notre projet.

Au cours de ce projet, nous avons rencontré plusieurs difficultés liées aux contraintes du temps, d'apprendre des nouvelles technologies et d'analyse des besoins du client, mais nous sommes arrivés à la fin de réaliser la plupart des fonctionnalités prévues pour notre application, en répondant aux besoins du client et en respectant les objectifs cités au début de projet.

Ce projet nous a été très bénéfique, car il nous a aidés à découvrir et d'apprendre de nombreuses technologies innovantes. Nous avons permis d'épanouir nos connaissances sur le modèle de logiciel SaaS, et d'avoir l'occasion d'exploiter la méthode Scrum pour la gestion du déroulement de notre projet.

Pour s'assurer une continuation de développement de notre solution et d'aboutir à un produit complet, performant et riche de fonctionnalités, et qui répond à nos attentes, nous avons proposé quelques perspectives pour améliorer et faire évoluer notre travail. En premier, nous devons continuer à implémenter les fonctionnalités qui ont été déjà définies dans la conception mais non encore réalisées, comme :

- Gestion des rôles des utilisateurs.
- Créer les formulaires manquants de la partie pré-configuration du système ERP.
- Mettre à jour la configuration d'une instance déjà déployée.

En suite, nous citons les futures fonctionnalités à concevoir et implémenter dans nos pro-

chaines versions de l'application «ODOaaS», comme :

- Filtrer des instances en créant des URL de sous domaines pour empêcher les utilisateurs de voir d'autres instances ERP du même environnement odoo.
- Création, suppression et duplication manuelle d'instances d'un client.
- Création, importation manuelle ou automatique du backup d'un instance ERP.
- Migration d'instances entre hôtes.
- Fourniture de divers plans d'abonnement.
- Intégration de passerelle de paiement.
- Surveiller les ressources système allouées à l'instance.
- Définir le nom d'hôte de l'environnement odoo automatiquement à l'aide de la fonctionnalité de reverse proxy de nginx.
- Automatiser le processus de facturation des instances fournissent aux clients.

Références webographiques

- [1] IT Solutions. [En ligne]. Disponible sur : <http://www.itsolutions.dz>. Consulté le 16 mars 2019. <http://itsolutions.dz/applications/erp-sap-s4-hana/>
- [2] Odoo. [En ligne]. Disponible sur : <https://fr.wikipedia.org/wiki/Odoo>. Consulté le 16 mars 2019.
- [3] Laurent AUDIBERT, 2009. [En ligne]. Disponible sur : <https://laurent-audibert.developpez.com/Cours-UML>. Consulté le 20 mars 2019.
- [4] Django. [En ligne]. Disponible sur : <https://python-django.dev/page-django-introduction-python>. Consulté le 27 mai 2019.
- [5] Vuejs. [En ligne]. Disponible sur : <https://www.supinfo.com/articles/single/6959-une-premiere-approche-vuejs>. Consulté le 27 mai 2019.
- [6] Odoo API. [En ligne]. Disponible sur : <https://www.odoo.com/documentation/11.0/webservices/odoo>. Consulté le 27 mai 2019.
- [7] OdooRPC. [En ligne]. Disponible sur : <https://odoorpc.readthedocs.io/en/latest/>. Consulté le 27 mai 2019.
- [8] Docker. [En ligne]. Disponible sur : <https://opensource.com/resources/what-docker>. Consulté le 27 mai 2019.
- [9] Docker. [En ligne]. Disponible sur : <https://www.lebigdata.fr/docker-definition>. Consulté le 27 mai 2019.
- [10] Docker sdk. [En ligne]. Disponible sur : <https://dockerpy.readthedocs.io/en/stable/>. Consulté le 27 mai 2019.
- [11] Compose. [En ligne]. Disponible sur : <https://docs.docker.com/compose/overview>. Consulté le 27 mai 2019.
- [12] Docker. [En ligne]. Disponible sur : [https://fr.wikipedia.org/wiki/Docker_\(logiciel\)](https://fr.wikipedia.org/wiki/Docker_(logiciel)). Consulté le 27 mai 2019.

- [13] Docker. [En ligne]. Disponible sur : <https://en.wikipedia.org/wiki/Git>. Consulté le 29 mai 2019.
- [14] Latex. [En ligne]. Disponible sur : <https://openclassrooms.com/fr/courses/1617396-redigez-des-documents-de-qualite-avec-latex/1617565-quest-ce-que-latex>. Consulté le 29 mai 2019.
- [15] Draw.io. [En ligne]. Disponible sur : <https://blog.valiantys.com/fr/apps-fr/draw-io-addon-confluence/>. Consulté le 29 mai 2019.
- [16] Trello. [En ligne]. Disponible sur : <https://fr.wikipedia.org/wiki/Trello>. Consulté le 29 mai 2019.
- [17] Postman. [En ligne]. Disponible sur : <https://blog.webnet.fr/presentation-de-postman-outil-multifonction-pour-api-web/>. Consulté le 29 mai 2019.
- [18] Github. [En ligne]. Disponible sur : <https://openclassrooms.com/fr/courses/2342361-gerez-votre-code-avec-git-et-github/2433636-github-quest-ce-que-cest>. Consulté le 29 mai 2019.
- [19] Sublime. [En ligne]. Disponible sur : https://fr.wikipedia.org/wiki/Sublime_Text. Consulté le 29 mai 2019.
- [20] PhpMyAdmin. [En ligne]. Disponible sur : <https://www.phpmyadmin.net/>. Consulté le 29 mai 2019.
- [21] Workbench. [En ligne]. Disponible sur : <https://www.mysql.com/fr/products/workbench/>. Consulté le 29 mai 2019.
- [22] Saas. [En ligne]. Disponible sur : <https://www.lebigdata.fr/definition-saas>. Consulté le 08 juin 2019.
- [23] Front office. [En ligne]. Disponible sur : <https://www.commentcamarche.net/contents/318-front-office-et-back-office>. Consulté le 08 juin 2019.
- [24] Back office. [En ligne]. Disponible sur : <https://www.commentcamarche.net/contents/318-front-office-et-back-office>. Consulté le 08 juin 2019.
- [25] Saas Kit. [En ligne]. Disponible sur : https://apps.odoo.com/apps/modules/11.0/odoo_saas_kit/. Consulté le 13 juin 2019.
- [26] Saas tools. [En ligne]. Disponible sur : <https://it-projects-llc.github.io/odoo-saas-tools/getting-started/>. Consulté le 13 juin 2019.
- [27] TrendERP. [En ligne]. Disponible sur : https://www.odoo.com/apps/modules/11.0/multi_instance_c. Consulté le 13 juin 2019.

[28] Scrum. [En ligne]. Disponible sur : <https://www.bettercalldave.io/methodologie/definition-scrum/>. Consulté le 13 juin 2019.

[29] Scrum. [En ligne]. Disponible sur : <https://blog.myagilepartner.fr/index.php/2018/08/17/ceremonie-sprint-scrum/>. Consulté le 18 juin 2019.

[30] SPA. [En ligne]. Disponible sur : https://en.wikipedia.org/wiki/Single-page_application. Consulté le 20 juin 2019.

Résumé

Ce présent projet a été réalisé durant le stage effectué au sein de l'entreprise IT Solutions. Il s'inscrit dans le cadre du projet de fin d'études, pour l'obtention du diplôme de master en Informatique option Génie Logiciel.

Avec l'avènement des offres commerciales « cloud », les coûts liés à l'acquisition des solutions de type ERP par les petites et moyennes entreprises, ont connues des réductions significatives. Malheureusement, les coûts de déploiement et de configuration non pas connues le même niveau de réduction, et pour répondre à cette problématique, nous avons développé une application web qui vise à automatiser le déploiement et la configuration d'un système ERP dans le cloud.

Mots Clés: Système ERP, Cloud, Application Web, Docker.

Abstract

This project realized during the internship at IT Solutions Company. For obtaining the master's degree in Computer Science option Software Engineering.

With the advent of "cloud" commercial offers, the costs associated with the acquisition of ERP solutions by small and medium-sized companies have seen significant reductions. Unfortunately, the costs of deployment and configuration not known the same level of reduction, and to address this problem, we have developed a web application that aims to automates the deployment and configuration of an ERP system in the cloud.

Keywords: ERP System, Cloud, Web Application, Docker.

ملخص

تم إنجاز هذا المشروع خلال فترة التدريب في إطار شركة IT Solutions. وهو بمثابة مشروع تخرج للحصول على درجة الماستر في الإعلام الآلي تخصص هندسة برمجيات. مع ظهور العروض التجارية "السحابية"، شهدت التكاليف المرتبطة باكتساب أنظمة ERP من قبل الشركات الصغيرة والمتوسطة تخفيضات كبيرة. لسوء الحظ، تكاليف النشر والتكوين لم تشهد نفس التخفيضات، ولمعالجة هذه المشكلة، قمنا بتطوير تطبيق ويب يعمل على أتمتة نشر وتكوين أنظمة ال ERP في السحابة.

الكلمات المفتاحية: نظام الإي آر بي، سحابة، تطبيق الويب، دوكر