

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB BLIDA-1-
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE



Mémoire de fin d'études pour l'obtention
du diplôme de Master en Informatique

Option : Système d'Information et Réseaux

**Conception et développement d'une solution pour
décentraliser la gestion des multiples clients sur la
plateforme « CTRL-i »**

Réalisé Par : - RAHMANI Mohamed Islam

- SELAHI Ahmed

Encadré par : - Mr Boudjakdji Djamel

Promotrice : - Mme ARKAM Meriem

Jury : - Mme CHICKI. I

- Mme BACHA. S

2018/2019

ملخص

على نحو متزايد، العديد من العملاء يعانون من مشاكل مع العرض وتكلفة التطبيقات المثبتة محليا، لحل هذه المعضلة تم اقتراح حلول جديدة من بينها توفير وتقديم الخدمات لإدارة التطبيقات عن بعد، في مشروعنا، اخترنا اقتراح بنية سحابية لا مركزية من أجل نظام ERP "CTRL-i" المقترح من قبل شركة PC MAX والتي تعتمد على نموذج "SaaS" (البرنامج كخدمة) مع احترام متطلبات جديدة للعملاء لمعرفة تخفيض تكاليف الاستثمار على البرمجيات (اقتناء حزمة البرمجيات، والمواد المادية والبشرية، وكذلك تكلفة الصيانة والتحديث...).

في مذكرتنا، قمنا بتطوير حل سحابة SaaS مع نموذج للتحكم في الوصول مستوحى من نموذج RBAC (التحكم في الدخول على أساس دور المستعمل).

الكلمات الرئيسية: ERP, CTRL-i, Cloud, Saas, التحكم في الدخول على أساس الدور، RBAC.

Résumé

De plus en plus, beaucoup de clients souffrent de problème avec la fourniture et le cout des applications localement installées. Pour résoudre ce problème, de nouvelle solution ont été proposées, parmi elles la prestation et la fourniture de services pour la gestion d'application à distance.

Dans notre projet, nous avons choisi de proposer une architecture cloud pour la décentralisation de l'ERP « CTRL-i » proposé par l'entreprise PC MAX qui est basée sur le modèle 'SaaS' (software as a service) tout en respectant les nouvelles exigences des clients à savoir la réduction des couts d'investissement sur logiciel (acquisition du progiciel, des moyens matériels et humains, ainsi que le cout de maintenance et mise à jour...).

Dans notre mémoire, nous avons développé une solution de cloud computing SaaS munie du modèle de contrôle d'accès inspirer du modèle RBAC (Rôle Based Access Contrôle).

Mots clés : ERP, CTRL-i, Cloud, Saas, Contrôle d'accès, RBAC

Abstract

More and more, many customers are struggling with the provision and cost of locally installed applications. To solve this problem, new solutions have been proposed, among them the provision and provision of services for remote application management.

In our project, we chose to propose a cloud architecture for the decentralization of the ERP "CTRL-i" proposed by the company PC MAX which is based on the model 'SaaS' (software as a service) while respecting the new requirements of the customers to know the reduction of the investment costs on software (acquisition of the software package, the material and human means, as well as the cost of maintenance and update ...).

In our brief, we developed a SaaS cloud computing solution with an access control model inspired by the RBAC (Role Based Access Control) model.

Keywords: ERP, CTRL-i, Cloud, SaaS, Access Control, RBAC.

Remerciement

Tout d'abord, nous rendons grâce à Dieu Tout-Puissant de nous avoir doté du savoir, de la santé et de la volonté.

On tient à remercier sincèrement l'équipe de « ScriptShock », qui ont accepté de nous accueillir et qui s'ont toujours montré à l'écoute et l'aide tout au long de la période de ce stage.

Nous adressons également nos remerciements à notre promotrice Mme ARKAM Meriem.

Nous tenons à exprimer nos sincères remerciements à tous les enseignements de la qualité des cours dispensés dans le Master « SIR » avoir excité nos réflexions, et une profonde satisfaction intellectuelle, merci donc aux enseignants-chercheurs.

Nous tenons à présenter notre reconnaissance aux membres du jury pour nous avoir fait l'honneur d'accepter d'évaluer ce travail.

Nos remerciements vont également à toute personne ayant contribué de près ou de loin à la concrétisation de ce travail.

Merci à toutes et tous.

Dédicaces

Avant tout, je remercie Dieu le tout puissant de m'avoir donné la force, le courage et la volonté nécessaire pour la réalisation de ce mémoire.

Je dédie ce modeste travail à mes chers parents qui m'ont soutenu durant toute la période du travail.

À vous mes frères

À vous mes amis

À tous les membres de ScriptShock

Merci d'être toujours là

Table des matières

INTRODUCTION GENERALE.....	9
Chapitre I : Les ERP.....	11
I.1. Introduction.....	12
I.2. Les ERP	13
I.3. Périmètre de gestion d'un ERP.....	13
I.4. Avantages et Inconvénients d'ERP	13
I.4.1. Les Avantages	13
I.4.2. Les Inconvénients	14
I.5. Architecture des ERP	14
I.5.1. Architecture Technique	14
I.5.2. Architecture Modulaire :	15
I.6. Types d'ERP	16
I.6.1. Les ERP propriétaires.....	16
I.6.2. Les ERP Open-Source	17
I.7. Mode de mise en place des ERP.....	17
I.7.1. Sur site (On-Premise) :	18
I.7.2. ASP :	21
I.7.3. Etude comparative :	23
I.8. L'ERP « CTRL-i »	25
I.9. Conclusion :	26
Chapitre II : Modèle Saas Du Cloud Computing	27
II.1. Introduction.....	28
II.2. Définition Cloud Computing.....	28
II.3. Les modèles du Cloud Computing.....	29
II.3.1. Le modèle Infrastructure comme un service (IaaS)	30
II.3.2. Le modèle Plate-forme comme un service (PaaS)	30
II.3.3. Le Modèle Logiciel comme un service (SaaS).....	31
II.4. Situation actuelle et à venir du modèle SaaS	32
II.5. L'architecture du SaaS	33
II.5.1. Niveaux de maturité	33
II.6. Virtualisation	37
II.7. Les avantages et les inconvénients du SaaS	38
II.7.1. Les avantages	38
II.7.2. Les inconvénients	38

- II.8. Comparaison entre SaaS et les précédentes solutions : 39
- II.9. Conclusion 43
- Chapitre III : Analyse et Conception 44
 - III.1. Introduction : 45
 - III.2. Présentation d’UML 45
 - III.3. Les Diagrammes UML..... 45
 - III.3.1. Diagramme de cas d’utilisation : 46
 - III.3.2. Diagramme de séquence : 50
 - III.3.3. Diagramme de classe : 54
 - III.3.4. Diagramme d’activité du CTRL-i 55
 - III.4. La Sécurité : 57
 - III.4.1. Modèle RBAC : 57
 - III.4.2. Laravel : 57
 - III.5. Conclusion : 61
- Chapitre IV : Implémentation..... 62
 - IV.1. Introduction 63
 - IV.2. Outils et environnement de développement 63
 - IV.2.1. Technologies côtés serveur..... 63
 - IV.2.2. Technologies côtés client..... 63
 - IV.2.3. La base de données..... 66
 - IV.3. Présentation d’Application 67
 - IV.3.1. Page Login client 67
 - IV.3.2. Page Création de compte client..... 67
 - IV.3.3. Page Demande d’Applications 68
 - IV.3.4. Page Login Admin..... 69
 - IV.3.5. Page Welcome Admin 69
 - IV.3.6. Gestion des Rôles..... 70
 - IV.3.7. Gestion des Admins 71
 - IV.3.8. Gestion des Clients..... 71
 - IV.3.9. Gestion des Applications..... 72
 - IV.3.10. Gestion des Permissions d’applications..... 73
 - IV.4. Conclusion : 74
- Conclusion Générale 75

Listes des Tableaux

Tableau I.1 : Etude Comparative entre ASP et Sur site.....	24
Tableau II.1 : Comparaison entre Saas et Sur Site.....	40
Tableau II.2 : Saas vs ASP.....	42

Listes des Figures

Figure I.1 : Architecture Technique	15
Figure I.2 : Architecture Modulaire	16
Figure I.3 : Modèle sur site	19
Figure I.4 : modèle ASP / sur site	25
Figure II.1 : Modèle Saas	29
Figure II.2 : Illustration des modèles du Cloud Computing	30
Figure II.3 : Les Caractéristiques du modèle SaaS	32
Figure II.4 : L'évolution du SaaS.....	32
Figure II.5 : Architecture personnalisable.....	33
Figure II.6 : Architecture d'application SaaS configurable	34
Figure II.7 : Architecture multi-locataire	35
Figure II.8 : Architecture d'application SaaS modulable	37
Figure II.9 : comparaison entre le modèle sur site et le modèle SaaS.....	39
Figure II.10 : diagramme de bénéfice d'une application en mode SaaS et en mode On-Premise	41
Figure III.1 : Diagramme de cas d'utilisation générale.....	46
Figure III.2 : Paquetage de cas d'utilisation décrivant « Gérer les rôles ».....	47
Figure III.3 : Paquetage de cas d'utilisation décrivant « Gestion des applications ».....	48
Figure III.4 : Paquetage de cas d'utilisation décrivant « Gérer les Users Ctrl-i ».....	49
Figure III.5 : Paquetage de cas d'utilisation décrivant « Gestion des clients ».....	50
Figure III.6 : Diagramme de séquence charger le déroulement général du rôle	51
Figure III.7 : Diagramme de séquence charger le déroulement Client-Admin	52
Figure III.8 : Diagramme de séquence charger le déroulement Sup-Admin et Sub-Admin	53
Figure III.9 : Diagramme de Classe de contrôle d'accès utilisateur à base de rôle.....	54
Figure III.10 : Diagramme de Classe de contrôle des droits d'exécutions des composants de 'CTRL_i' inspirer du modèle RBAC.....	54
Figure III.11 : Diagramme d'activité client-admin	56
Figure III.12 : Diagramme d'activité Sup-Admin/Sub-Admin	57
Figure III.13 : Modèle RBAC	58
Figure IV.1 : Capture d'écran « Page Login client »	67
Figure IV.2 : Capture d'écran « Page Création de compte client ».....	68
Figure IV.3 : Capture d'écran « Page Demande d'Applications ».....	69
Figure IV.4 : Capture d'écran « Page Login Admin ».....	69
Figure IV.5 : Capture d'écran « Page Welcome Admin »	70
Figure IV.6 : Capture d'écran « Gestion des Rôles ».....	71
Figure IV.7 : Capture d'écran « Gestion des Admins ».....	71
Figure IV.8 : Capture d'écran « Gestion des Clients »	72
Figure IV.9 : Capture d'écran « Gestion des Applications »	73
Figure IV.10 : Capture d'écran « Gestion des Permissions d'applications ».....	73

INTRODUCTION GENERALE

Le client est généralement la principale source de revenus pour les entreprises. Or, avec le changement de l'économie dû notamment à l'intégration des nouvelles technologies dans les Relations Client-entreprise, la concurrence devient de plus en plus serrée et les clients peuvent ainsi désormais se permettre de choisir leur fournisseur ou d'en changer par un simple clic.

Les critères de choix des clients sont notamment des critères financiers, de réactivité de l'entreprise mais également des critères purement affectifs (besoin de reconnaissance, besoin d'être écoutés ...). Ainsi dans un monde de plus en plus concurrentiel les entreprises souhaitant augmenter leurs bénéfices ont plusieurs alternatives :

- Augmenter la marge sur chaque client

- Augmenter le nombre de clients

- Augmenter le cycle de vie du client c'est-à-dire le fidéliser

Les nouvelles technologies permettent aux entreprises de mieux connaître leur clientèle et gagner leur fidélité en utilisant les informations les concernant de telle manière à mieux cerner leurs besoins et donc pour mieux y répondre.

Les entreprises se concurrencent pour préserver la fidélité de leurs clients et pour attirer de nouveaux clients. En effet, certaines entreprises ont réussi à dominer leurs concurrents et à prouver leur efficacité dans ce domaine grâce à l'implantation du Cloud Computing.

Le Cloud Computing (en français : l'informatique en nuages) est en train de modifier la manière de notre utilisation de l'informatique. Cloud Computing est apparu comme un nouveau modèle de gestion et d'utilisation des systèmes informatiques. Le concept majeur faisant référence à l'utilisation de la puissance de calcul et/ou de stockage de serveurs informatiques distants à travers un réseau, généralement Internet. Le développement remarquable du Cloud Computing, ces dernières années, Les utilisateurs ne sont plus propriétaires de leurs serveurs informatiques mais peuvent ainsi accéder de manière évolutive à de nombreux services en ligne sans avoir à gérer l'infrastructure sous-jacente, souvent complexe. L'accès au service se fait par une application standard facilement disponible, tel

qu'un navigateur web. Cela c'est de profiter à des services et des applications disponibles en ligne à travers le web en mode services. Ce dernier c'est un nouveau modèle économique qui s'appelle le mode SaaS (Software as a service) (en français : Logiciel en tant que service).

La croissance du nombre des clients dans le siège scriptshock d'Entreprise de PCMax qui est considéré comme un acteur de référence dans le domaine des développements et qui veut s'orienter vers la solution du SaaS, a créé une pression de travail fatal et ingérable, suite à sa stratégie de servir pour sa clientèle qui est individuelle, elle doit cloner l'application web dans les différents serveurs des clients à chaque fois, puis les configurer et les suivre individuellement d'où à partir d'un certain nombre de clients, il est devenu impossible de répondre aux besoins techniques du logiciel. C'est dans ce cadre que s'inscrit ce Projet de Fin d'Etudes de Master Académique en SIR consistant à :

- Etudier l'impact de la solution sur l'application web actuel et planifier les solutions optimales dans les aspects SOFTWARE et HARDWARE.
- Développer la partie de la plateforme qui va gérer les clients (abonnement, accès ...).
- Configurer les routes des API en fonction des changements apportés.

Le présent document synthétise tout le travail effectué dans cette perspective. Tout au long de ce projet, on traitera quatre chapitres :

Dans le premier chapitre « Les ERP », nous présentons le principe du ERP pour lequel le progiciel a été développé.

Dans le second chapitre intitulé « *ETAT DE L'ART* », on donnera une vision plus claire sur les différents modèles de ce service et différents modes de son déploiement ainsi que les différentes solutions.

Dans le troisième chapitre « Généralité sur le mode SaaS », on donnera une vision sur la notion de Cloud Computing et nous proposons à traiter les principes de logiciel en mode SaaS ensuite, les avantages et les inconvénients de ce dernier.

Dans le quatrième chapitre « Conception, implémentation et évaluation », décrire en détail les spécifications techniques matérielles et l'architecture réseau ainsi que l'architecture de l'application et à sa conception afin de déterminer les différents modules qui interagissent au cours de l'implémentation.

Chapitre I : Les ERP

I.1. Introduction

Les nouvelles technologies de l'information permettent l'intégration des processus et de toutes les fonctions de l'entreprise. L'accès des entreprises aux nouvelles technologies, à Internet en particulier, tend à modifier la communication entre les différents acteurs du monde des affaires. Notamment entre l'entreprise et ses clients (Business To Consumer, B2C), le fonctionnement interne de l'entreprise (Business To Employées, B2E) et la relation de l'entreprise avec ses différents partenaires et fournisseurs (Business To Business, B2B). Cet outil permet une gestion homogène et cohérente du système d'information (SI) de l'entreprise, en particulier pour la gestion commerciale de la chaîne de production à la vente d'un produit. Un système d'information (SI) est un ensemble organisé de ressources (matériels, logiciels, personnel, données et procédures) qui permet de collecter, regrouper, classer, traiter et diffuser de l'information dans un environnement donné. Ce mouvement a un impact majeur sur la fonction comptable et financière de l'entreprise.

Actuellement, toute entreprise est prête à investir des sommes colossales dans l'implantation des technologies logicielles en vue d'améliorer ses services, d'accroître son agilité et sa flexibilité, de réduire les coûts, d'augmenter la production et de faire face aux défis du marché. Vu la croissance des activités au sein des entreprises, la tâche de gérer efficacement toutes les fonctions s'avère de plus en plus complexe et difficile. Pour surpasser ces difficultés, une entreprise doit utiliser des outils optimisés et adaptés facilitant les tâches et offrant des fonctionnalités riches et utiles. Parmi ces outils nous trouvons les systèmes intégrés de gestion tel que les ERP (Entreprise Ressources Planning).

Les ERP sont des outils de gestion et d'analyse permettant d'optimiser la diffusion des informations en interne, d'améliorer les processus de gestion et d'automatiser les tâches ce qui augmente énormément la réactivité des entreprises et leurs agilités.

Nous verrons tout d'abord une présentation générale des ERP ce qui nous conduira à la description de leur architecture modulaire.

I.2. Les ERP

Une nouvelle catégorie de logiciels d'application packagée est apparue au cours de la dernière décennie, se consolidant apparemment sous une seule bannière, une industrie de plusieurs milliards de dollars qui comprend le quatrième plus grand fournisseur de logiciels au monde, plusieurs autres des plus grandes entreprises de logiciels et les plus grandes organisations de conseil en gestion du monde. Habituellement appelés systèmes de planification des ressources d'entreprise (ERP) [1].

Un système logiciel d'entreprise complet qui permet à une entreprise d'automatiser et d'intégrer la majorité de ses processus d'entreprise, de partager des données et des pratiques communes dans toute l'entreprise et de produire et d'accéder à des informations en temps réel. L'objectif ultime d'un système ERP est que les informations ne doivent être saisies qu'une seule fois.[2].

I.3. Périmètre de gestion d'un ERP

L'ERP a une capacité de couvrir un large périmètre de gestion, c'est-à-dire :

- La gestion des **achats**.
- La gestion des **ventes**.
- La gestion **comptable** : comptabilité client, fournisseur, immobilisations, personnel.
- Le **contrôle de gestion**.
- La gestion de **production** (planification, ...).
- La gestion des **stocks** (logistique).

I.4. Avantages et Inconvénients d'ERP

I.4.1. Les Avantages

Les avantages d'un PGI incluent une meilleure organisation grâce :

- A la mutualisation des données,
- Aux nombreux modules qui composent les logiciels ERP.

Les avantages ERP sont les suivants :

- Pas de doublons ou de saisies multipliées.
- Disponible sur plusieurs langues.
- Possibilité de travailler à plusieurs devises.

- Base de données unique et mise à jour en temps réel.
- Transmission des informations sans erreur.
- Optimisation de la coordination entre les services.
- Meilleure gestion des stocks et des Ressources humaines.
- Un contrôle centralisé de l'entreprise.

I.4.2. Les Inconvénients

Les inconvénients sont moins nombreux que les avantages, ils sont néanmoins à prendre en compte :

- Demande de l'organisation et de la rigueur.
- Système complexe qui nécessite un certain savoir-faire. Il faudra penser à une formation approfondie pour les employés.
- Les difficultés d'apprentissage par le personnel d'entreprise.
- La mise en œuvre complexe.
- La bonne connaissance et la remise en cause des processus de l'entreprise.
- Le coût élevé aux utilisateurs (matériel + licence + intégration + formation + maintenance).

I.5. Architecture des ERP

I.5.1. Architecture Technique

L'architecture d'un ERP se compose principalement d'un serveur ERP sur lequel est présente une base de données unique et disponible pour tous les salariés. Elle implique l'utilisation de différents réseaux :

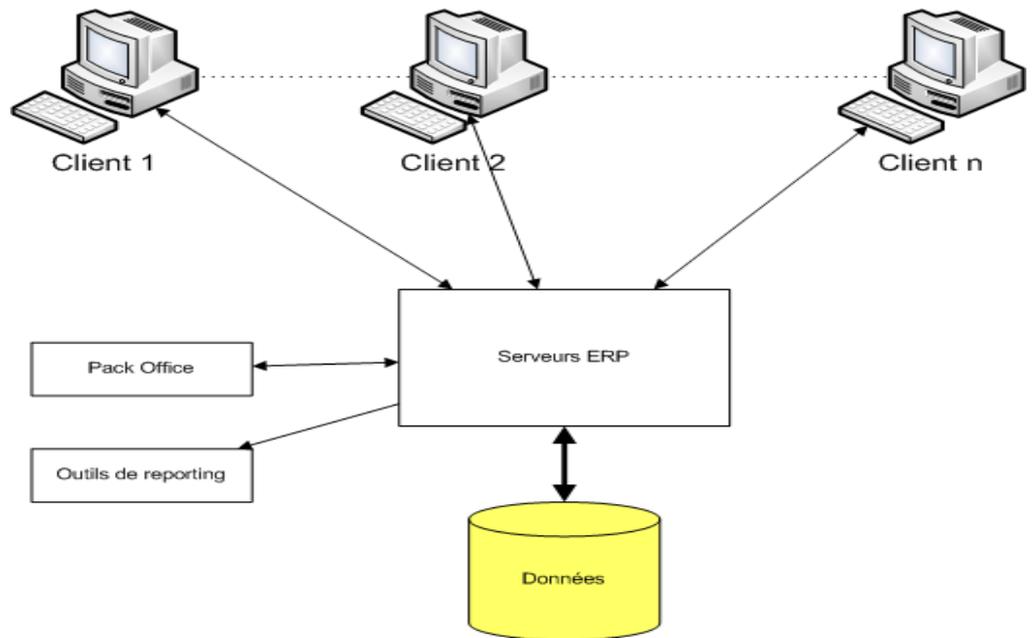


Figure I.1 : Architecture Technique [36]

- Multisites,
- Intranet, extranet
- Serveur ERP à distance.

C'est la mise en réseau des données qui permettra leur gestion décentralisée [5].

I.5.2. Architecture Modulaire :

Un ERP est modulaire dans le sens où il est possible de n'avoir qu'une ou plusieurs applications, en même temps, ou peu à peu. Les applications modulaires telles que les ERP permettent d'être sûr de la compatibilité des modules entre eux, ils s'imbriquent comme des blocs de Lego et fonctionnent ensemble (Pas de vérification de compatibilité à effectuer) [6].

Le schéma ci-dessous explique l'architecture modulaire qui intègre plusieurs modules retouchant aux grandes fonctions d'une entreprise et on peut citer de la manière suivante : le module finance, logistique et e-commerce.

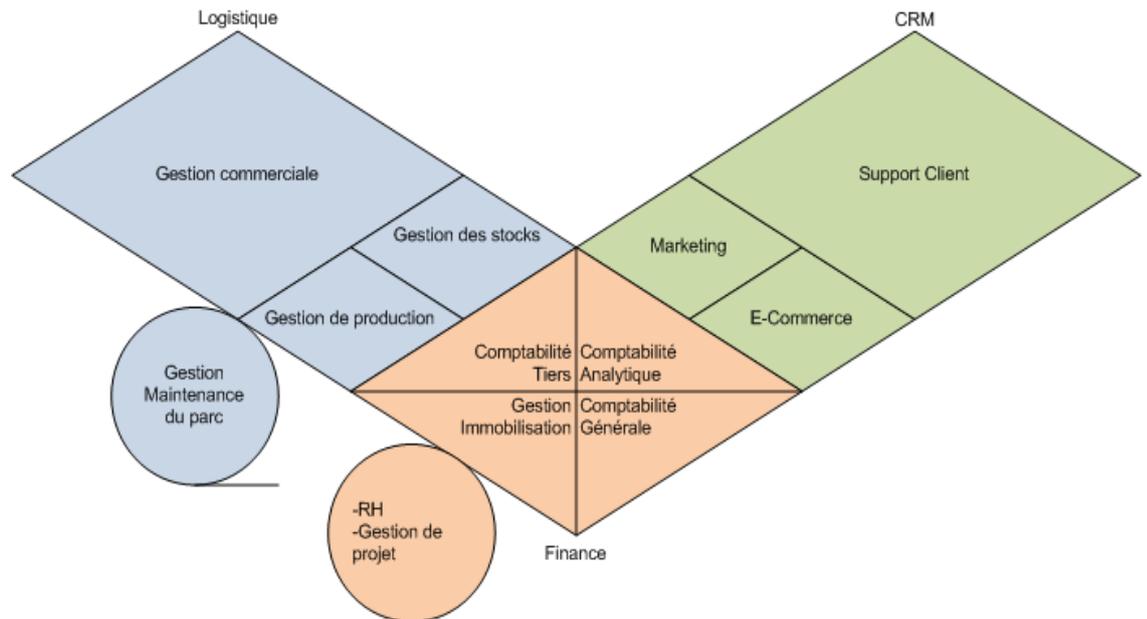


Figure I.2 : Architecture Modulaire [36]

I.6. Types d'ERP

On retrouve deux types d'ERP :

- Les ERP propriétaires édités par des sociétés qui impliquent l'achat d'une licence ou abonnement.
- Les ERP Open-source qui sont gratuits.

I.6.1. Les ERP propriétaires

Parmi les principaux ERP propriétaires, on peut citer :

SAP

SAP est le leader mondial du monde des ERP. Ce progiciel a remporté rapidement un succès important auprès des grandes entreprises en proposant un progiciel multilingue et multidevises. SAP s'intéresse aussi au marché des PME, en pleine croissance en proposant sa suite Business One, pour les entreprises de 2 à 250 salariés.

Il est doté de plusieurs sortes de modules : des modules orientés logistique (Matériel Management, Production Planning, Sales and Distribution, Qualité Management, Plant Maintenance), Finance (FI, CO, TR, IM) et ressources humaines (RH).

ERP SAGE

SAGE vise en particulier le marché des PME où l'éditeur est bien implanté. Les logiciels de SAGE visent en particulier les entreprises de moins de 500 salariés. Cependant, en Novembre 2005, Sage rachète l'éditeur Adonix pour s'ouvrir le marché des PME/PMI de 500 à 2 000 salariés.

Les modules de SAGE sont les suivants :

- La logistique qui inclut la gestion des nomenclatures, des plans de production, l'analyse des coûts de fabrication et la gestion des stocks.
- La comptabilité qui inclut la comptabilité générale et analytique, la gestion de trésorerie, la communication bancaire.
- Les ressources humaines : gestion de la paie, des carrières, des compétences et de la formation.
- Le marketing/CRM/ventes : campagnes de ventes/marketing, gestion des forces de vente, module de e-commerce, configurateur de catalogues, gestion des paiements sécurisés.

I.6.2. Les ERP Open-Source

Un ERP est un système informatique qui comprend plusieurs modules tous reliés ensemble à une seule base de données unique qui est le logiciel. L'Enterprise Resource Planning open source est une solution informatique incontournable pour les petites et les moyennes entreprises (PME), qui permettent aux entreprises de disposer de solutions stables et personnalisables pour la gestion de leur activité. Les spécificités de l'ERP open source sont en effet plus appropriées aux PME parce que comme on peut voir sur ce site ce type de logiciel s'adapte aux besoins particuliers des utilisateurs. La réduction des coûts de développement induite par cette transparence accrue de l'information permet aux entreprises d'économiser l'achat de licences d'utilisation

Comme tout ERP, l'ERP open source est caractérisé par :

- Nombreuses fonctionnalités qui s'adaptent à chaque métier.
- Base de données unique.
- Administration unique.
- La traçabilité.
- Interconnexion des services.

I.7. Mode de mise en place des ERP

À l'heure actuelle, les fournisseurs de services applicatifs sont confrontés à d'immenses défis. Dans un contexte où les applications et les environnements sont plus complexes que jamais, les utilisateurs exigent une haute disponibilité, une vitesse ultra-rapide et un accès instantané sur tous leurs terminaux et en tout lieu, en même temps le grand point qui attire l'utilisateur c'est comment minimiser les dépenses. Pour offrir les meilleures expériences utilisateur, le fournisseur de services applicatifs doit non seulement optimiser et accélérer les applications, mais également maintenir les

plus hauts niveaux de sécurité pour se défendre contre les attaques et éviter l'interruption des activités.

Les chercheurs avec leurs efforts essayent de trouver des solutions pouvant aider les fournisseurs de services applicatifs à définir des moyens d'améliorer la fiabilité, les performances et la sécurité, pour un bon rendement aux entreprises, deux solutions de mise en place sont souvent proposées : acquisition du logiciel avec installation sur site ou encore la fourniture de services avec accès à distance.

I.7.1. Sur site (On-Premise) :

La manière dont les logiciels et les applications sont fournis aux utilisateurs et aux entreprises a beaucoup évolué ces dernières années. Le modèle d'acquisition traditionnel, appelé logiciel sur site.

I.7.1.1. Définition :

En informatique, sur site désigne les logiciels nécessitant d'être installés « sur site » c'est à dire sur les serveurs locaux de l'entreprise. Le terme est traduit en français par « licence lourde » ou « auto-hébergement ». Il s'agit donc d'une implémentation en local du logiciel et de toutes les ressources qui y affèrent.[7]

Les solutions sur site (*on premise*) sont les logiciels dont l'architecture est physiquement présente au sein de l'entreprise (c'est à dire installée sur les serveurs de l'entreprise) qui s'est acquittée d'une licence auprès de l'éditeur. Un logiciel sur site est acquis pour une durée illimitée, il s'agit d'un achat permanent de licences d'utilisation. Généralement, l'hébergement est optionnel : l'entreprise peut choisir si l'hébergement est pris en charge par l'éditeur ou par elle-même. Le client peut en effet stocker ses données et l'environnement technique du logiciel sur ses propres serveurs ou ceux de son prestataire existant. L'hébergement d'un logiciel ou d'une application sur site nécessite des moyens humains et des ressources techniques afin d'assurer la maintenance et l'administration des serveurs.

Dans le cadre d'un contrat de maintenance corrective et évolutive, les mises à jour du logiciel sur site sont effectuées automatiquement par l'éditeur. Le client bénéficie ainsi des dernières améliorations du logiciel sur site. Bien que la plupart des logiciels sur site soient laissés au support et à la maintenance de l'utilisateur, quelques entreprises proposent des services de support supplémentaires avec leur logiciel sur site. [8][9]

Ce type de configuration est populaire parmi les grandes entreprises et les gouvernements qui appliquent des politiques de sécurité et de confidentialité strictes exigeant que les logiciels soient installés en interne sur l'infrastructure de la société. Lorsque le besoin de dimensionner l'application

se fait sentir avec un logiciel sur site, l'entreprise doit supporter les coûts et le temps associés aux mises à niveau.

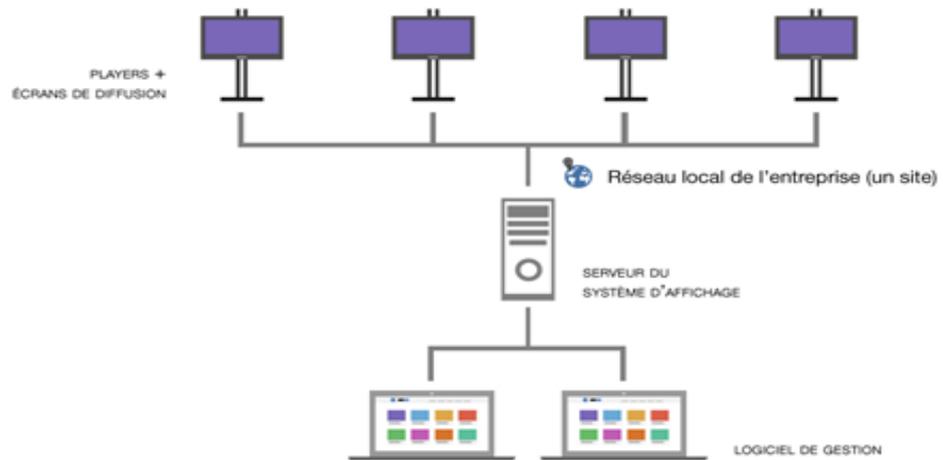


Figure I.3 : Modèle sur site [37]

I.7.1.2. Autres exemples de logiciels sur site

- SharePoint 2013
- Adobe Creative Suite
- Web Trends sur site

I.7.1.3. Avantages du logiciel sur site

Bien que la version sur site nécessite davantage de connaissances techniques et des engagements de maintenance plus importants, elle offre également de nombreux avantages : [10]

Contrôle total

C'est l'un des principaux avantages pour les entreprises et la principale raison pour laquelle les logiciels sur site sont souvent le seul choix, en particulier pour les grandes entreprises et les gouvernements. À une époque où la vie privée des utilisateurs est exploitée et utilisée à mauvais escient, la nécessité pour les entreprises de protéger leurs données et leurs informations n'a jamais été aussi grande. Le stockage de logiciels sur site offre aux entreprises non seulement un contrôle direct et la protection de leurs données, mais également une plus grande souplesse d'utilisation.

Par exemple, une entreprise peut combiner les données brutes avec celles d'autres services, tels que leurs systèmes de gestion de la relation client, ce qui leur permet de créer un profil client complet. Cette liberté et cette capacité à intégrer différents logiciels sur site constituent un avantage très attractif pour les grandes entreprises.

Sécurité accrue

Le contrôle et la flexibilité offerts par les logiciels sur site permettent aux entreprises et aux gouvernements de mettre en œuvre diverses mesures de sécurité pour protéger davantage leurs services contre les attaques.

Personnalisation du logiciel

Selon la licence du logiciel et les droits de propriété, le logiciel sur site permet aux entreprises de personnaliser, d'ajuster et de configurer certains éléments du logiciel afin de l'adapter à leurs besoins et à leurs exigences.

I.7.1.4. Inconvénients du logiciel sur site

Voici les différents inconvénients : [11]

La mise en œuvre

- Cela prend du temps, du personnel et du matériel pour créer un nouvel environnement.
- Des achats supplémentaires de matériel / logiciel peuvent être nécessaires.

Support & Maintenance

- Vous êtes responsable de la maintenance de l'application. Votre service informatique est responsable de la haute disponibilité et de la reprise après sinistre.
- Offre un contrôle sur vos données et procure un plus grand sentiment de propriété.

Cycles de mise à niveau

Vous êtes responsable des mises à niveau, qui sont souvent coûteuses et prennent beaucoup de temps.

Coût

- Coûts d'entrée et d'exploitation élevés comparés au SaaS en raison des besoins en infrastructure et en personnel de soutien.
- Frais d'entretien courants bas.

- Actifs matériels tangibles.
- Les coûts de mise à niveau seront élevés.

I.7.2. ASP :

I.7.2.1. Définition

Le terme ASP est un acronyme pour « Application Service Provider », traduit généralement en français par « Fournisseur d'Applications Hébergées » (FAH). Ce terme désigne à la fois le modèle informatique de fournitures de services applicatifs en ligne et le prestataire qui héberge et fournit ces services (au travers d'un réseau). John McCarthy, chercheur au MIT, est le premier à parler de cette démarche lors d'une présentation en 1961. Il explique qu'il sera possible à des clients d'externaliser certaines applications par l'intermédiaire d'un réseau.

Bien que l'ASP (application service provider) est assez important, la plupart des gens confondent souvent le, car il est « hébergé ». Cependant, le fournisseur de services d'application (ASP) est beaucoup plus proche du logiciel hérité que l'autre logiciel. Pour de nombreuses personnes. En règle générale, ASP est utilisé pour fournir des services informatiques aux clients via un réseau. Il permet aux clients d'accéder à un programme d'application via un protocole standard, par exemple, le CRM via Internet ou via HTTP.

L'un des principaux avantages des ASP est que les petites et moyennes entreprises peuvent obtenir des logiciels spécialisés. Cependant, il est souvent inabordable pour beaucoup. Cela élimine également le besoin physique de distribuer le logiciel, y compris les mises à niveau. En plus du logiciel, les ASP vont également maintenir des services à jour, y compris la sécurité physique et électronique, un support technique 24h/24, 7j/7 et un support intégré pour la continuité des activités et promouvoir le travail flexible. Les ASP offrent aux entreprises un moyen d'externaliser certains ou presque tous les aspects de leurs besoins informatiques.

Les Catégories

Voici 5 sous-catégories d'ASP :

- ASP d'entreprise - fournissent non seulement des applications professionnelles haut de gamme, mais également des solutions à large spectre
- ASP spécialisés - fournissent des applications répondant à un besoin commercial spécifique, tels que les ressources humaines, les services de site Web, le traitement des paiements par carte de crédit, etc.

- ASP locaux / régionaux - fournissent divers services d'application aux petites entreprises d'une zone limitée
- Les ASP métier en volume fournissent généralement des services d'applications préemballés à faible coût en volume, depuis leur propre site, aux petites et moyennes entreprises. PayPal en est un bon exemple
- ASP sur le marché vertical : offre un soutien à un secteur particulier en fournissant un ensemble de solutions pour un certain type de client, tel que le cabinet dentaire ou les soins de santé en général.

Tandis que les ASP tentent de se concentrer sur la gestion et l'hébergement de logiciels ISV tiers. De plus, les ASP fournissent des applications client-serveur plus traditionnelles, nécessitant l'installation de logiciels sur les PC des utilisateurs. De plus, l'architecture logicielle des ASP nécessitait que, pour chaque entreprise, vous deviez gérer une instance distincte de l'application.

I.7.2.2. Les Raisons pour l'échoue

Cependant, ASP est un modèle qui a échoué pour les raisons suivantes :

- Il manque d'évolutivité pour le vendeur
- Pas d'agrégation intégrée de données
- Trop de personnalisation
- Généralement un modèle de revenu unique
- Aucune donnée d'effet de réseau pour la collecte et l'agrégation

Bien que certains fournisseurs aient connu le succès avec le modèle ASP, ce succès a été limité en raison de problèmes d'évolutivité et de personnalisation entre les systèmes.

Une application déployée dans un modèle ASP / locataire unique signifie, dans la plupart des cas, que le produit ne prend pas en charge la colocation multiple et que le fournisseur ne souhaite pas investir son temps dans la refonte de l'architecture du produit. Cela signifie également que le produit n'a pas été correctement commercialisé, mais plutôt considéré comme un logiciel que comme une entreprise, et manque probablement de prise en charge des modèles de revenus intégrés, de facturation, de comptage avancé, etc....

I.7.3. Etude comparative :

La comparaison entre les 2 modes se comporte sur des mesures d'évaluation suivantes :

- **La mise en œuvre** : La mise en œuvre doit répondre à les propres contraintes des utilisateurs : le coût, le temps de mise en œuvre, les performances, la fiabilité.
- **Personnalisation** : personnaliser notre environnement de travail : stockage des données (sur propre serveur de l'entreprise ou sur autres), différentes tâches, mise à jour sur l'application...
- **Support & Maintenance** : La maintenance et le support des applications incluent les compétences et les exigences pour prendre en charge les systèmes d'application, notamment le dépannage, la modification, la maintenance et l'amélioration des systèmes existants. Application Maintenance and Support s'applique également aux applications s'exécutant dans un environnement de production.
- **Coût** : Un montant qui a été payé ou donné afin d'obtenir quelque chose. En affaires, le coût est généralement une évaluation monétaire de (1) l'effort, (2) le matériel, (3) les ressources, (4) le temps et les services consommés, (5) la mise à jour et (6) livraison. Toutes les dépenses sont des coûts.
- **Intégration** : est un outil capable d'extraire les données et de les charger dans un entrepôt de données dans l'intranet ou dans un autre site avec des différentes méthodes.
- **L'évolutivité** : possibilité d'évoluer l'application ou d'une affectation quelconque.

	ASP	Sur site
La mise en œuvre	Rapide à mettre en œuvre car il exploite une plate-forme prête à l'emploi qui a déjà été provisionnée, implémentée et testée par le fournisseur.	Prendre des achats (matériel/logiciel) et du temps, du personnel et du matériel pour créer un nouvel environnement.
Personnalisation	Trop de personnalisation	Option la plus flexible pour la personnalisation. La participation du vendeur est minimale.
Support & Maintenance	Dépendance informatique minimale (matériels) pour la maintenance des applications.	Vous êtes responsable de la maintenance de l'application. Votre service informatique est responsable de la haute disponibilité et de la reprise après sinistre. Offre un contrôle sur vos données et procure un plus grand sentiment de propriété.
Coût	Moins de coût	Coûts plus élevés (Frais d'entretien / matériels / mise à niveau / équipe informatiques)
Intégrations	Pas d'agrégation intégrée de données.	Les intégrations peuvent être relativement simples sur l'intranet. Le transfert de données entre systèmes sera plus rapide.
L'évolutivité	Coté vendeur : manque d'évolutivité (travail avec des logiciels traditionnels)	Nécessite une planification à long terme et l'engagement de ressources pour la mise à l'échelle.

Tableau I.1 : Etude Comparative entre ASP et Sur site

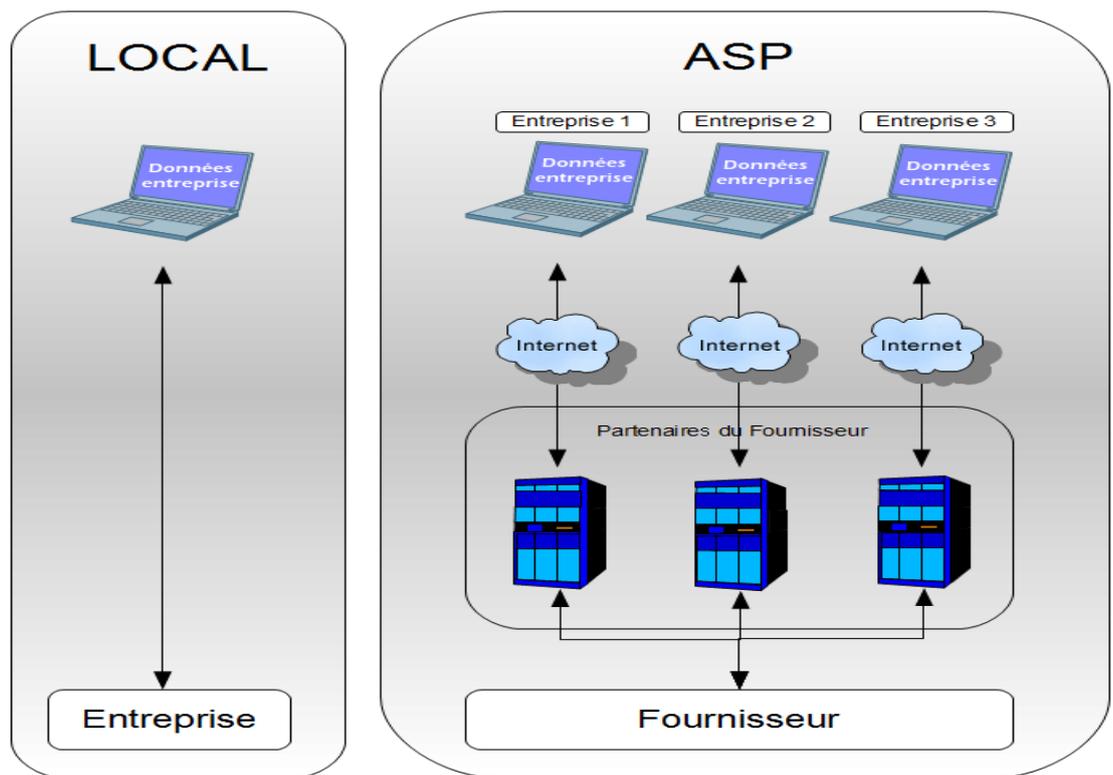


Figure I.4 : modèle ASP / sur site [38]

I.8. L'ERP « CTRL-i »

La planification des ressources d'entreprise (ERP) « CTRL-i » est un processus utilisé par les entreprises pour gérer et intégrer les parties importantes de leurs activités. Il est destiné aux petites et moyennes entreprises (PME).

De nombreuses applications logicielles ERP sont importantes pour les entreprises car elles les aident à mettre en œuvre la planification des ressources en intégrant tous les processus nécessaires au fonctionnement de leurs entreprises avec un seul système. Le système logiciel ERP « CTRL-i » gère : la planification, les Achats, les ventes, dossiers fournisseurs et clients, produits et Stockage, la gestion des ressources humaines, la gestion des projets, comptabilité et flux monétaire, la gestion de maintenance assistée par ordinateur (GMAO), agence de voyage. D'autres extensions sont en cours de développement.

Pour le moment l'ERP « CTRL-i » utilise le mode sur site, mais la croissance du nombre de clients a posé des problèmes :

- Répondre aux besoins techniques du logiciel est presque impossible.

- La gestion des multiples clients : le besoin est différé pour chaque client, donc lorsque le nombre de clients est grand ce n'est pas facile de gérer ses bases de données, et les problèmes relient dans l'ERP pour chaque client.

- La mise à jour est individuelle : lorsqu'il y a une mise à jour sur l'ERP c'est-à-dire réglage des problèmes et des bugs, il faut déplacer à chaque client pour faire cette mise à jour.

Notre travail est de trouver une solution de migration d'une architecture sur site vers une architecture décentralisée, toutefois la solution ASP ne répond pas vraiment aux besoins de l'entreprise, car dans le mode ASP manque d'évolutivité, et ne répond pas forcément aux besoins du clients.

Pour cela, nous avons opté pour une solution cloud computing avec le modèle Saas.

I.9. Conclusion :

L'objectif de ce chapitre est de définir les différentes méthodes principales pour effectuer la bonne fourniture du service d'application a les entreprises cliente, à travers les définitions et les besoins des clients pour avoir une distribution d'application qui adapté avec de ces besoins. Ainsi que les différentes mesures d'évaluation de ces dernières. Ce qui nous facilitera par la suite de trouver la bonne méthode.

Chapitre II : Modèle Saas Du Cloud Computing

II.1. Introduction

Les applications d'entreprise traditionnelles sont toujours compliquées et onéreuses. Fini l'époque où les gens achetaient des CD / DVD pour charger des logiciels sur leur ordinateur ou devaient télécharger des applications d'installation gigantesques (ou presque). La charge que représente le nombre et la variété des logiciels et matériels requis pour leur exécution est écrasante. Il faut toute une équipe d'experts pour en assurer l'installation, la configuration, les tests, l'exécution, la sécurité et la mise à jour.

Grâce à Internet haut débit, nous pouvons maintenant envoyer rapidement de gros volumes de données entre des ordinateurs locaux et des serveurs externes. Le secteur des technologies évolue progressivement vers le cloud computing, un environnement informatique dans lequel vous n'êtes pas lié par une machine ou un logiciel local.

Donc la possibilité de reproduire les applications traditionnelles hébergées sur le réseau comme des services est né avec les nouveaux modèles des services web tel que SaaS du CLOUD COMPUTING.

II.2. Définition Cloud Computing

Tout système d'information se compose des deux composants suivants :

- Les infrastructures : réseaux, postes de travail, serveurs, outils de développement... En pratique, tout ce qui n'intéresse pas les utilisateurs.
- Les usages : la bureautique, les progiciels de gestion intégrés, le décisionnel, les blogs... Tout ce qui apporte de la valeur aux utilisateurs.

Aujourd'hui, on peut accéder à de multiples applications web à partir de notre ordinateur, de notre appareil mobile ou tout autre objet communicant. Le Cloud Computing est donc un nuage faisant référence à Internet et au web dans lequel nous n'avons aucune idée de l'emplacement physique des infrastructures, qui permettent le bon fonctionnement des applications en ligne.

Le cloud computing est nommé en tant que tel car les informations auxquelles on accède se trouvent à distance dans le cloud ou dans un espace virtuel. Les entreprises qui fournissent des services en nuage permettent aux utilisateurs de stocker des fichiers et des applications sur des serveurs distants, puis d'accéder à toutes les données via Internet. Cela signifie que l'utilisateur n'est pas obligé d'être dans

un endroit spécifique pour y accéder, ce qui lui permet de travailler à distance. Internet devient le nuage et voilà : vos données, votre travail et vos applications sont disponibles à partir de tout appareil avec lequel vous pouvez vous connecter à Internet, partout dans le monde.

Le cloud computing peut être à la fois public et privé. Les services de cloud public fournissent leurs services sur Internet moyennant des frais. Les services de cloud privé, en revanche, ne fournissent des services qu'à un certain nombre de personnes. Ces services sont un système de réseaux fournissant des services hébergés. Il existe également une option hybride, qui combine des éléments des services publics et privés. [12]



Figure II.1 : Modèle Saas [39]

II.3. Les modèles du Cloud Computing

Le Cloud Computing n'est pas une technologie unique comme une micropuce ou un téléphone portable. Il s'agit plutôt d'un système composé principalement de trois services : logiciel en tant que service (SaaS), infrastructure en tant que service (IaaS) et plate-forme en tant que service (PaaS).

Le schéma suivant illustre les trois modèles du Cloud Computing :



Figure II.2 : Illustration des modèles du Cloud Computing [40]

II.3.1. Le modèle Infrastructure comme un service (IaaS)

L'IaaS signifie Infrastructure comme un service. Il s'agit de la partie infrastructure du Cloud, c'est-à-dire les outils serveurs, administrateurs servant à fournir l'infrastructure comme les outils de virtualisation, la console d'administration, le système, les bibliothèques. Les clients peuvent éviter d'avoir à acheter des logiciels ou des serveurs, mais plutôt se procurer ces ressources dans un service à la demande externalisé. Dans le IaaS, on retrouvera donc les composants clés : le réseau (montée en charge, firewall), la partie matérielle, la plate-forme de virtualisation, les outils de facturation et de contrôle de consommation, les niveaux de services. Le IaaS peut prendre plusieurs formes : fournisseur d'outils IaaS (VMware, Eucalyptus, Ubuntu) et les fournisseurs d'infrastructure complète (Amazon EC2, gogrid, etc.).

II.3.2. Le modèle Plate-forme comme un service (PaaS)

L'acronyme PaaS signifie « Platform as a Service » ou « plate-forme comme un service » est considérée comme la plus complexe des trois couches du Cloud Computing. PaaS partage certaines similitudes avec le SaaS, la principale différence étant qu'au lieu de fournir des logiciels en ligne, il s'agit en réalité d'une plate-forme de création de logiciels livrée via Internet. C'est également un ensemble de services d'infrastructure. On peut par exemple utiliser une plateforme PaaS pour exécuter des SaaS.

II.3.3. Le Modèle Logiciel comme un service (SaaS)

SaaS est basé sur le cloud computing, il évite aux entreprises d'installer et d'exécuter des applications sur leurs propres systèmes. Avec le modèle SaaS, le logiciel est exécuté sur une plateforme conçue pour une utilisation simultanée par un grand nombre d'utilisateurs. Ces utilisateurs accèdent au logiciel depuis n'importe quel endroit, à partir d'un navigateur web et d'une connexion Internet. Le logiciel est disponible uniquement sur les serveurs de son opérateur, qui commercialise un service intégré que l'on ne paye qu'à la consommation ou dans le cadre d'un abonnement. Cela a l'avantage élimine ou du moins réduit les coûts associés à l'achat et à la maintenance du matériel, aux logiciels et à l'assistance. Le coût d'installation initial d'une application SaaS est également généralement inférieur à celui d'un logiciel d'entreprise équivalent acheté via une licence de site. De plus, l'entreprise utilisatrice dispose des différentes mises à jour du produit sans coût de licences supplémentaires et surtout de façon transparente. Le SaaS contient en fait tout un éventail de services dédiés aux infrastructures des centres informatiques : réseaux, stockage, systèmes d'exploitation, bases de données, serveurs d'application, serveurs Web, services de restauration et de sauvegarde des données. Il offre, en outre, toute une gamme de services opérationnels : authentification, disponibilité, gestion de l'identité, suivi de la production, gestion des Support Packages, suivi des activités, montées de version et personnalisation du logiciel. Dans ce modèle, la customisation est aussi possible. Par exemple, les sociétés utilisatrices peuvent adapter les modèles de données ou encore personnaliser des fonctionnalités. Pour finir, le SaaS permet de mettre en œuvre des pratiques collaboratives en permettant de travailler à domicile en relation avec des partenaires. Google Documents par exemple, permet de travailler sur des documents partagés en temps réel. Plusieurs utilisateurs peuvent afficher et modifier un document en même temps. [13]

Le schéma suivant regroupe les caractéristiques du modèle SaaS que nous venons d'aborder :



Figure II.3 : Les Caractéristiques du modèle SaaS [41]

II.4. Situation actuelle et à venir du modèle SaaS

L'évolution du marché des applications en mode SaaS s'inscrit dans un cycle de développement comprenant trois vagues principales, comme expliqué ci-dessous : [14]

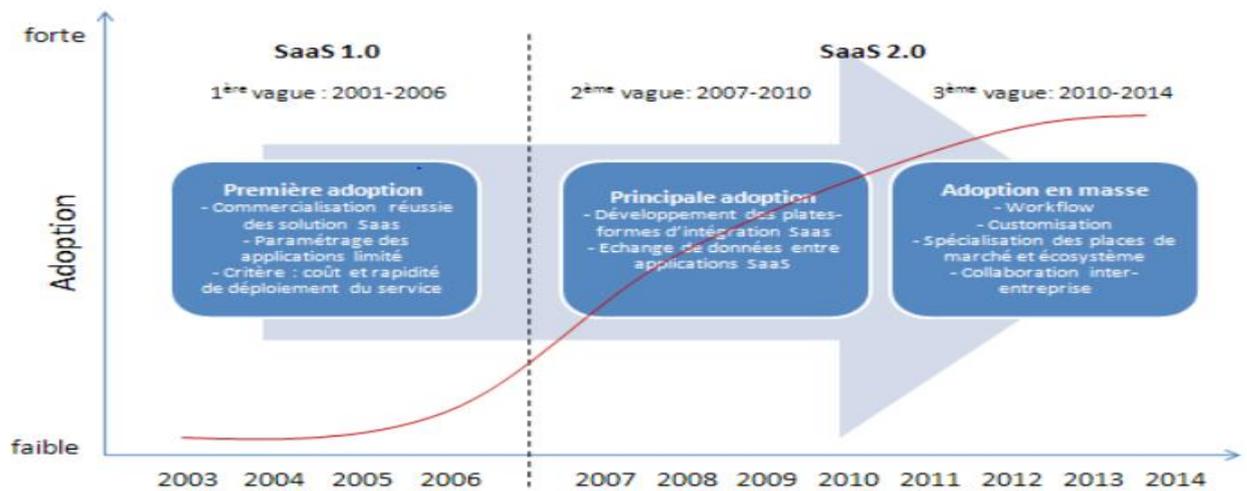


Figure II.4 : L'évolution du SaaS [41]

- La première phase [2001 - 2006] est la phase d'adoption. Elle est marquée par une commercialisation

réussie des solutions SaaS grâce notamment à leur rapidité de développement et leurs coûts peu élevés.

- La deuxième phase [2007 - 2010] est la phase d'adoption principale. Elle se caractérise par le développement des plates-formes d'intégration SaaS et un échange entre les données SaaS.

- La troisième phase [2010 - 2014] marquera une adoption massive des solutions SaaS. Il sera possible de paramétrer ou de personnaliser les applications SaaS jusqu'aux utilisateurs finaux.

II.5. L'architecture du SaaS

On classifie généralement les architecture SaaS sous quatre niveaux de maturité : Le niveau personnalisable, le niveau configurable, le niveau multi-locataires et le niveau modulable (flexibilité/évolutivité). Chaque niveau possède les caractéristiques du niveau précédent et offre des fonctions supplémentaires. [15]

II.5.1. Niveaux de maturité

Niveau 1 : Personnalisable

Au premier niveau de maturité, chaque client dispose de sa propre solution personnalisée d'application hébergée et utilise sa propre instance de l'application sur les serveurs du fournisseur.

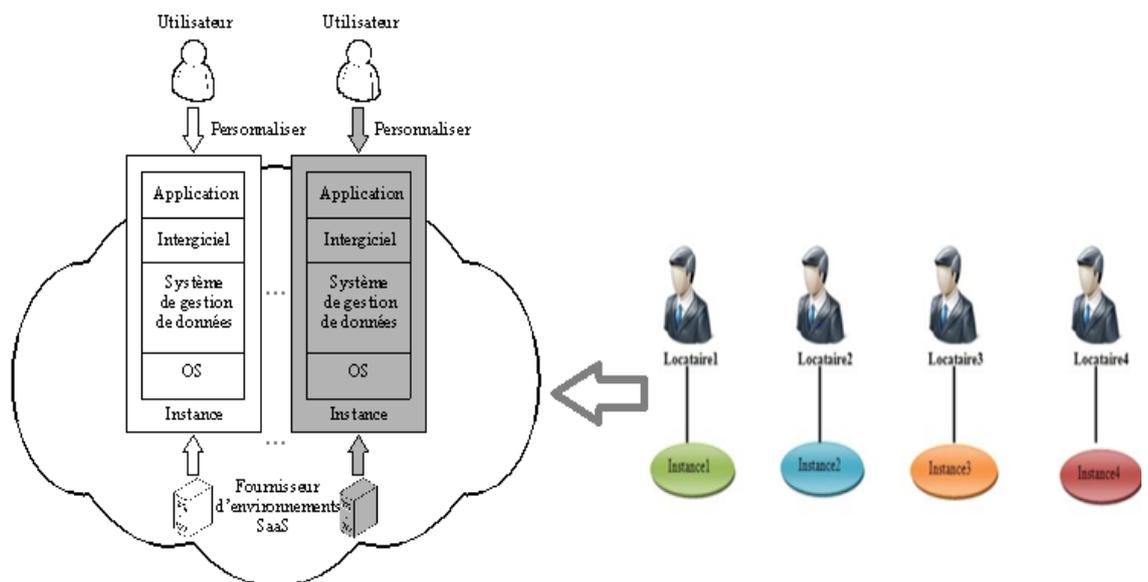


Figure II.5 : Architecture personnalisable [15]

Les applications qui se prêtent bien à ce scénario sont généralement développées pour répondre à des besoins spéciaux de l'entreprise ou de l'organisation. La priorité est donnée à la sécurité au sein du système lui-même. Ce niveau est identique au modèle de fourniture de logiciels traditionnel appelé ASP.

Niveau 2 : Personnalisable et configurable

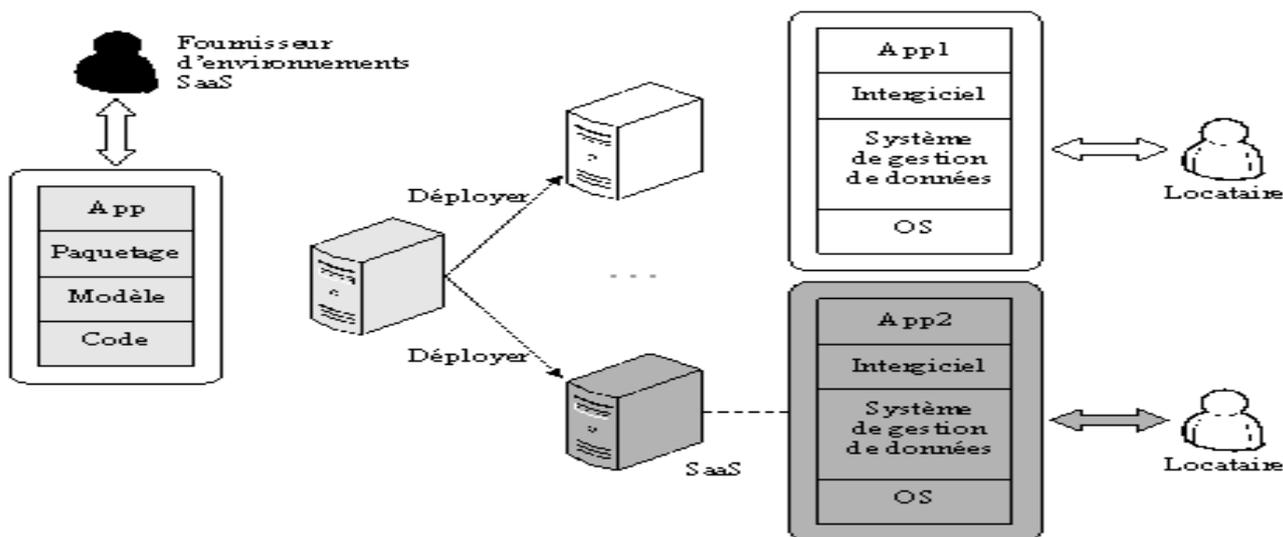


Figure II.6 : Architecture d'application SaaS configurable [15]

Le deuxième niveau de maturité offre la flexibilité par métadonnées configurable. Les fournisseurs d'environnements SaaS proposent des modèles communs pour obtenir à la fin, plusieurs clients peuvent utiliser le même « application de façon différente ». Cela permet au fournisseur de répondre aux besoins de personnalisation (chaque client) via des instructions de configuration (sur des machines virtuelles ou physiques) détaillées tout en simplifiant pour lui la maintenance et mise à jour d'une base de code commune. Chaque instance d'application est isolée aux d'autres. La Figure ci-dessus illustre cette architecture.

Les caractéristiques de l'application SaaS configurable sont :

- L'application est une copie d'un produit standard, puis les locataires configurent l'application selon leurs besoins. Cela dans la mise en service initiale de l'application SaaS configurable.
- Toute modification apportée au code du produit peut être facilement et immédiatement répercutée sur l'ensemble des locataires. Cela du côté du fournisseur d'applications SaaS.
- Chaque locataire mémorise ses données dans sa propre machine virtuelle ou physique, les machines étant isolées les unes des autres.

Niveau 3 : Personnalisable, configurable, et multi-locataire « multi-tenant »

L'architecture multi-locataire est l'un des aspects techniques fondamentaux du Cloud Computing et particulièrement des SaaS, elle signifie qu'il n'y a pas d'infrastructure distincte pour chaque client, mais que tous les utilisateurs travaillent sur la même plateforme. Le fournisseur d'applications SaaS utilise pour ce faire des métadonnées configurables.

Ce modèle nécessite des réglages fins :

- Un équilibrage de charge entre les utilisateurs (pour ne pas donner toute la puissance machine à un seul utilisateur).
- Une conception de tables capables de contenir des informations relatives à plusieurs utilisateurs, voire à plusieurs entreprises.
- Une puissance de calcul plus grande que les systèmes applicatifs de facture classique.
- Un contrôle d'accès multi-niveaux.

Nous vous proposons le schéma explicatif suivant :

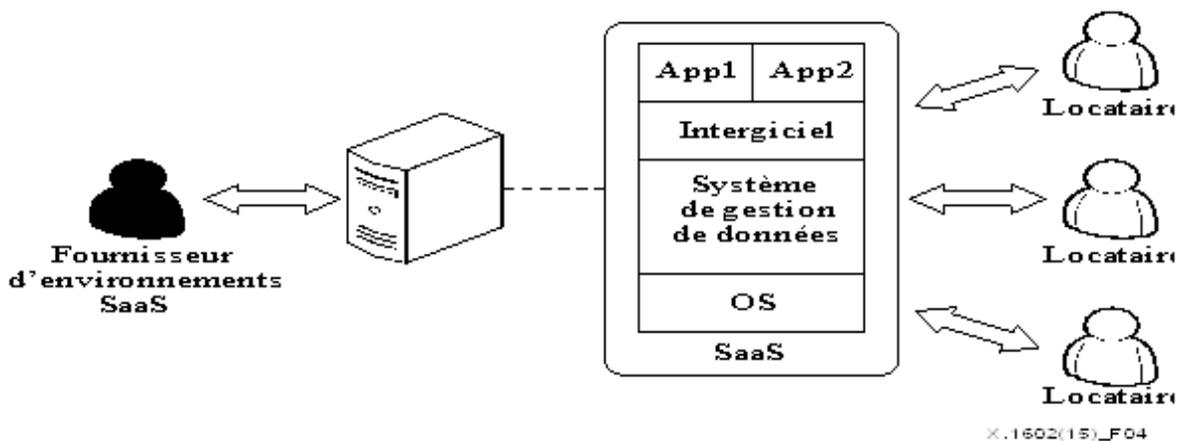


Figure II.7 : Architecture multi-locataire [15]

L'identification

Un système multi-locataires doit être capable d'identifier l'utilisateur par le identifiant/mot de passe pour mémoriser toutes les données de cet utilisateur dans un même ensemble de tables de la base de données. Il doit, dans le cas d'un service accessible sur internet, pouvoir lui associer toutes les informations relatives à son entreprise et à ses rôles et permissions. Une grande partie du système tient à la gestion des droits d'accès.

Les métamodèles

Les modèles de données sont prédéfinis et ne nécessitent qu'une personnalisation minimale. Les locataires n'ont qu'à choisir les éléments de données nécessaires à leurs applications de veille économique et à expliciter le mappage entre d'un côté les sources de données et, de l'autre, l'entrepôt et le modèle. Ce sont ces modèles qui serviront à lancer des requêtes SQL appropriées sur la base de données associée à l'application.

Le contenu de la base de données

Cette base de données comprend les informations relatives à tous les clients ou utilisateurs. Il arrive parfois que des données liées à différents clients soient inscrites dans les mêmes tables. Une vue virtuelle des informations propres à l'entreprise et à la personne qui se connecte est alors constituée.

Le contenant

La vision de l'utilisateur est totalement personnalisée d'où chaque utilisateur a une vision personnalisée sur son poste de travail. L'objectif des architectures multi-locataires est donc de trouver le bon compromis entre la customisation et la mutualisation des ressources.

Niveau 4 : Personnalisable, configurable, multi-locataire et modulable (évolutif/flexible)

A ce niveau en ajoute l'évolutivité et la flexibilité à travers une architecture dite « multi-tiers » qui supporte un nombre indéfini d'instances ou un grand nombre de clients de l'application tournant sur un nombre indéfini de serveurs. Les clients sont ensuite redirigés par un système de répartition de charge en fonction des besoins. Le fournisseur peut ainsi augmenter ou réduire les capacités du système en ajoutant ou supprimant des serveurs sans avoir à modifier son application. Lorsque le système doit prendre en charge un nouveau locataire, deux possibilités se présentent au niveau de la couche applicative : soit une ou plusieurs instances d'application sont créées en fonction des besoins spécifiques du locataire, soit une instance existante est choisie sur la base du mécanisme de répartition de la charge.

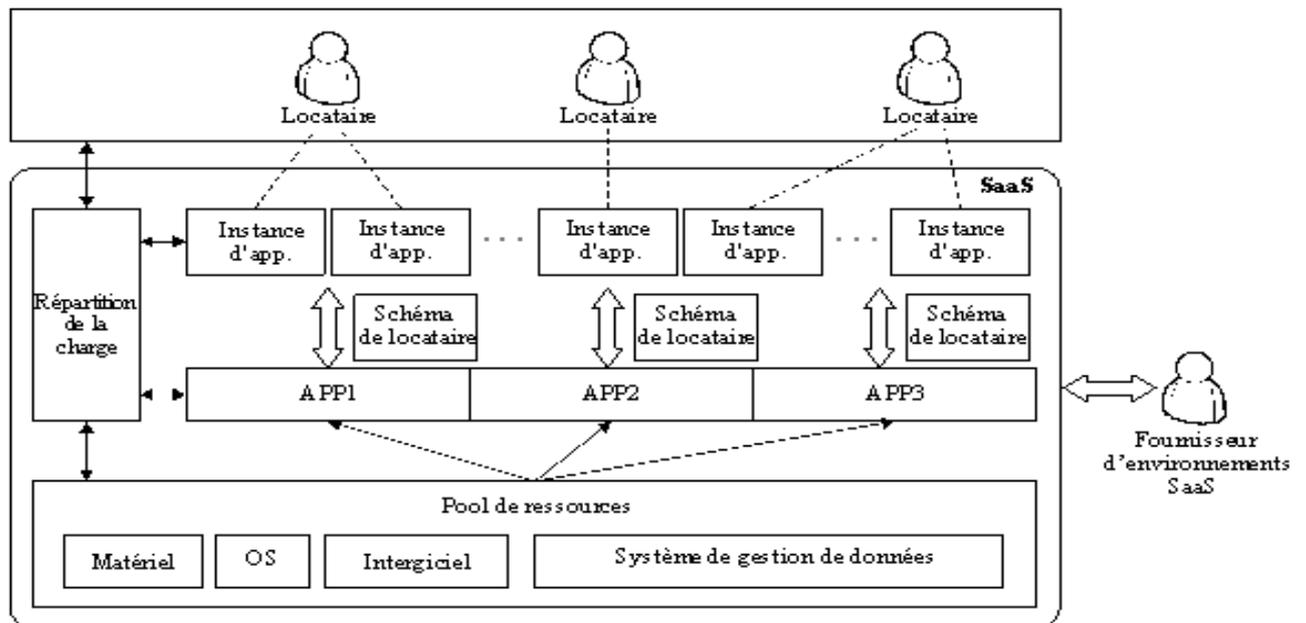


Figure II.8 : Architecture d'application SaaS modulable [15]

De même, les ressources sur lesquelles reposent les applications SaaS modulables doivent être configurables de façon très souple. Les matériels, intergiciels, logiciels et données doivent tous être gérés dans le pool de ressources. Les applications puisent dans le pool, de manière dynamique, toutes les ressources dont elles ont besoin. Lorsque cela est nécessaire, de nouvelles ressources peuvent être ajoutées sans recombinaison ni réarchitecture.

II.6. Virtualisation

Les architectures SaaS peuvent également faire usage de la virtualisation, soit en complément du caractère multi-locataire de l'application. Le principal avantage de la virtualisation est que cela peut augmenter les capacités du système sans pour autant avoir à reprogrammer quoi que ce soit. D'un autre côté, une quantité considérable de travail peut être nécessaire pour construire une application plus efficace.

Combiner le multi-locataire avec la virtualisation procure une meilleure flexibilité au niveau des réglages du système pour une performance optimale. [16]

II.7. Les avantages et les inconvénients du SaaS

II.7.1. Les avantages

- Le coût d'acquisition de matériel est ainsi éliminé, au même titre que les coûts d'approvisionnement et de maintenance, de licence de logiciel, d'installation et de support (Grâce à un logiciel SaaS, les entreprises n'ont plus besoin d'installer et de lancer des applications sur leurs propres ordinateurs ou sur leurs Data Centers).
- L'offre se présente sous la forme d'un abonnement mensuel dont le tarif est proportionnel à l'utilisation (Au lieu d'investir dans un logiciel à installer, et dans un équipement permettant de le prendre en charge, les utilisateurs souscrivent à une offre SaaS, grâce à cette flexibilité les entreprises peuvent organiser leur budget avec plus de précision et de facilité. De plus, il est possible de résilier l'abonnement à tout moment pour couper court aux dépenses).
- L'utilisateur peut accéder à plus ou moins de services et à des fonctionnalités à la demande (Le Logiciel en tant que Service est donc adapté aux besoins propres à chaque business).
- Les utilisateurs peuvent compter sur le fournisseur SaaS pour effectuer des mises à jour automatiquement et gérer l'ajout de patches correctifs (L'entreprise a donc moins besoin d'une équipe d'informaticiens internes et éviter d'acheter de nouveaux logiciels).
- Les utilisateurs peuvent y accéder depuis n'importe quel appareil connecté et n'importe quelle position géographique (L'accessibilité est l'un des grands points forts de ce modèle).
- Une application SaaS peut être utilisée par des milliers (voire des millions d'utilisateurs finaux simultanément puisqu'elle est stockée sur le Cloud).

II.7.2. Les inconvénients

Les entreprises doivent compter sur des vendeurs externes pour fournir des logiciels, maintenir ces logiciels, proposer une facturation détaillée et garantir un environnement sécurisé pour les données de l'entreprise. Pour éviter le problème en cas d'interruption de service, de changements de l'offre, de faille de sécurité ou tout problème, il est nécessaire pour les entreprises d'analyser soigneusement les conditions du service proposé par le fournisseur.

II.8. Comparaison entre SaaS et les précédentes solutions :

- SaaS vs sur site (On-Premise) :

La manière dont les logiciels et les applications sont fournis aux utilisateurs et aux entreprises a beaucoup évolué ces dernières années. Les clients nous demandent généralement de les aider à décider si un modèle de déploiement SaaS (logiciel en tant que service) ou sur site serait le meilleur choix.

Comparé à la méthode traditionnelle d'installation du logiciel directement sur un ordinateur ou un serveur, SaaS est fourni à l'utilisateur via Internet.

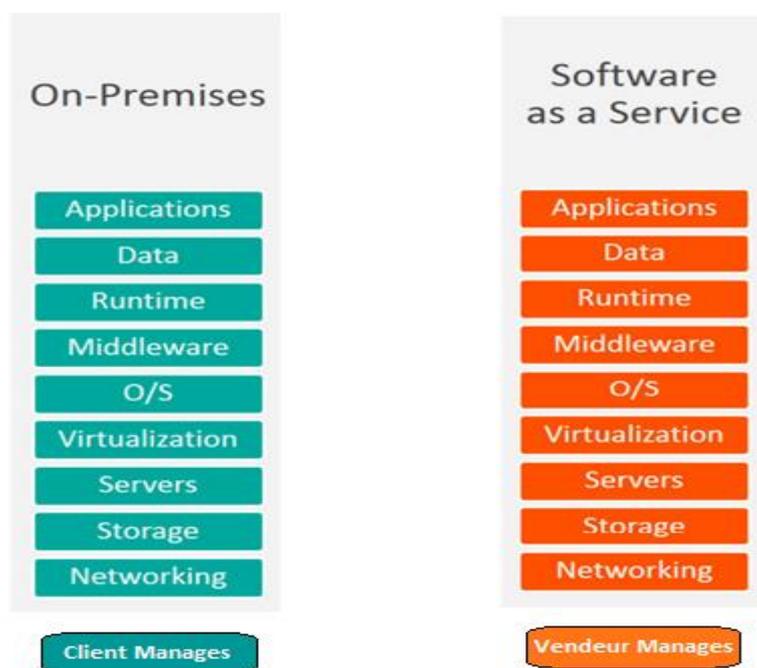


Figure II.9 : comparaison entre le modèle sur site et le modèle SaaS [42]

SaaS	Sur site
Peu ou pas de connaissances techniques requises (Maintenance et mise à jour assurées par le prestataire)	Le stage des connaissances techniques élevé (Maintenance et mise à jour dépendent de l'entreprise cliente lui-même)
Évolutivité dans un peu de temps et d'efforts	Nécessite une planification à long terme
Le fournisseur SaaS prend en charge les risques d'infrastructure pour assurer une haute disponibilité au client	L'entreprise cliente prend en charge les risques de son infrastructure
Abonnement mensuel payant (moins chère)	Achat de licence par poste (beaucoup de gaspillage d'argent)
Abonnement auprès d'un hébergeur	Pas de dépendance à un hébergeur
Frais d'exploitation réduits (économiser sur RH et le matériel de l'infrastructure)	Frais d'exploitation plus élevé (plus de RH et plus de hardware)
La mise en œuvre plus rapide	La mise en œuvre prend du temps, du personnel et du matériel pour créer un nouvel environnement
Données délocalisées (centralisé les données dans le serveur de fournisseur)	Données hébergées sur site (saturé les serveurs)

Tableau II.1 : Comparaison entre Saas et Sur Site

On-Premise vs. Software as a Service

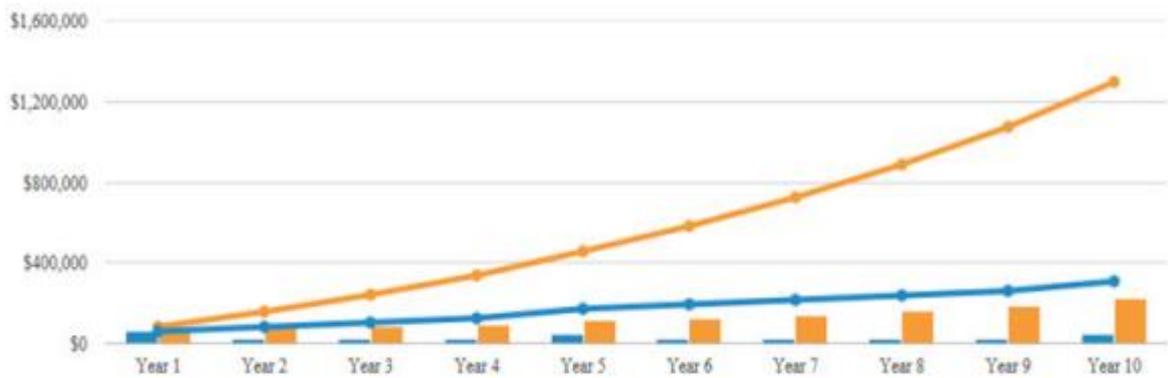


Figure II.10 : diagramme de bénéfice d'une application en mode SaaS et en mode On-Premise [43]

Récemment, de nombreuses entreprises ont opté pour les applications SaaS en raison de leur coût et de leur facilité d'utilisation.

- SaaS vs ASP :

Bien que la différence entre ASP (application service provider) et SaaS (software as a service) soit assez importante, la plupart des gens confondent souvent les deux modèles, car ils sont tous deux « hébergés ».

SaaS	ASP
Évolutivité dans un peu de temps et d'efforts	Il manque d'évolutivité pour le vendeur
Le fournisseur SaaS prend en charge les risques d'infrastructure pour assurer une haute disponibilité au client	Le fournisseur demande de paiement couteux pour avoir des mises à jour
Fournisseurs de logiciels appliquent toujours des mises à jour, des opérations de maintenance et des améliorations récentes	Fournissent des applications client-serveur plus traditionnelles
Réduire les complexités communes aux personnalisations	Personnalisation entre les systèmes
Multi-location intégrée (permettant des ressources et une infrastructure partagée)	Uni-locataire
Une architecture multi-locataire (Chaque entreprise gérer des instances partagé)	Une architecture uni-locataire (Chaque entreprise, vous deviez gérer une instance distincte de l'application)

Tableau II.2 : Saas vs ASP

Entre SaaS et ASP, la réalité sur le terrain est que les applications à locataire unique, telles que celles des modèles ASP ne sont pas correctement architecturées pour prendre en charge les exigences commerciales exigeantes relatives à l'architecte d'entreprise SaaS.

Globalement, entre SaaS et ASP, une application conçue et créée spécifiquement comme une offre SaaS est plus sûre si vous souhaitez utiliser une application Web, car elle sera également facile à faire évoluer sans coûts supplémentaires.

II.9. Conclusion

Dans ce chapitre, nous avons étudié SaaS, un modèle de déploiement de logiciel dans lequel les applications sont accessibles par l'Internet, propose des solutions aux entreprises de toutes tailles et de toutes natures gérées par des fournisseurs et utilisées par des clients à distance. SaaS est une nouvelle tendance très prometteuse.

Chapitre III : Analyse et Conception

III.1. Introduction :

L'analyse d'une application consiste à définir, représenter, et documenter les tâches effectuées dans l'entreprise, elle tenir compte des contraintes, des risques et de tout autre élément pertinent afin de développer un système répondant aux besoins du client. Elle permet de modéliser le projet à travers ses diagrammes, à partir des besoins recueillis auprès des utilisateurs. Dans ce chapitre nous permet d'identifier toutes les fonctionnalités de notre future application pour chaque type d'utilisateurs y compris ces besoins fonctionnels. Ceci se fera par l'identification des acteurs et la définition de tous les besoins qui seront par la suite modélisés par un diagramme de cas d'utilisation, suivi de leurs diagrammes de séquence. Les principaux concepts de modélisation sont :

- Chaque processus ou sous processus peut être décomposé en un graphe d'enchaînement de tâches.
- Chaque tâche est réalisée par un acteur (humain ou machine/logiciel).

Pour mener une bonne analyse pour cela on a opté pour la démarche UML qui offre une diversité de diagramme.

III.2. Présentation d'UML

UML (Langage de modélisation unifié), se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins. UML est également utilisée dans les phases terminales du développement avec les modèles de réalisation et de déploiement. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc... En clair, UML comme une boîte d'outils qui sert à améliorer les méthodes de travail. [20]

III.3. Les Diagrammes UML

UML prend en charge de 14 diagrammes y compris les diagrammes UML pour les Schémas XML et les autres pour modéliser les tables dans des bases des données relationnelles, chacun d'eux est dédié à la représentation d'un système logiciel suivant un point de vue particulier. Ces diagrammes sont regroupés dans deux grands ensembles : structurels et les diagrammes de comportement. Nous nous intéressons aux diagrammes utilisés pour l'analyse du processus sélectionné.

III.3.1. Diagramme de cas d'utilisation :

Ce diagramme est utilisé pour représenter les besoins des utilisateurs par rapport au système et donner une vision globale du comportement fonctionnel d'un système logiciel. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système.[21]

III.3.1.1. Diagramme du cas d'utilisation du processus

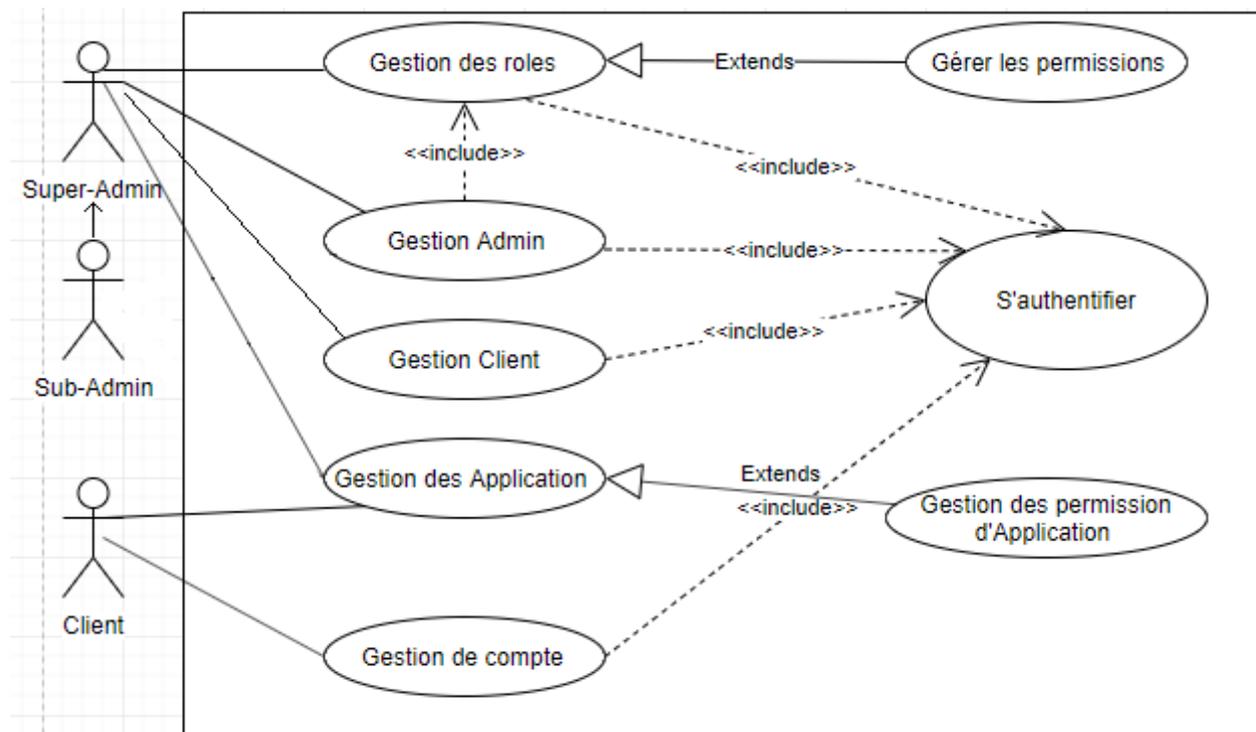


Figure III.1 : Diagramme de cas d'utilisation générale

III.3.1.2. Les paquetages des cas d'utilisations

CU-Gestion des rôles	
L'administrateur pourrait :	- Créer un rôle personnalisé pour l'affecter a d'autres utilisateurs.
	- Modifier un rôle lorsque on veut ajouter ou supprimer des permissions.
	- Supprimer un rôle.
	- Consulter toutes les informations des rôles exister.

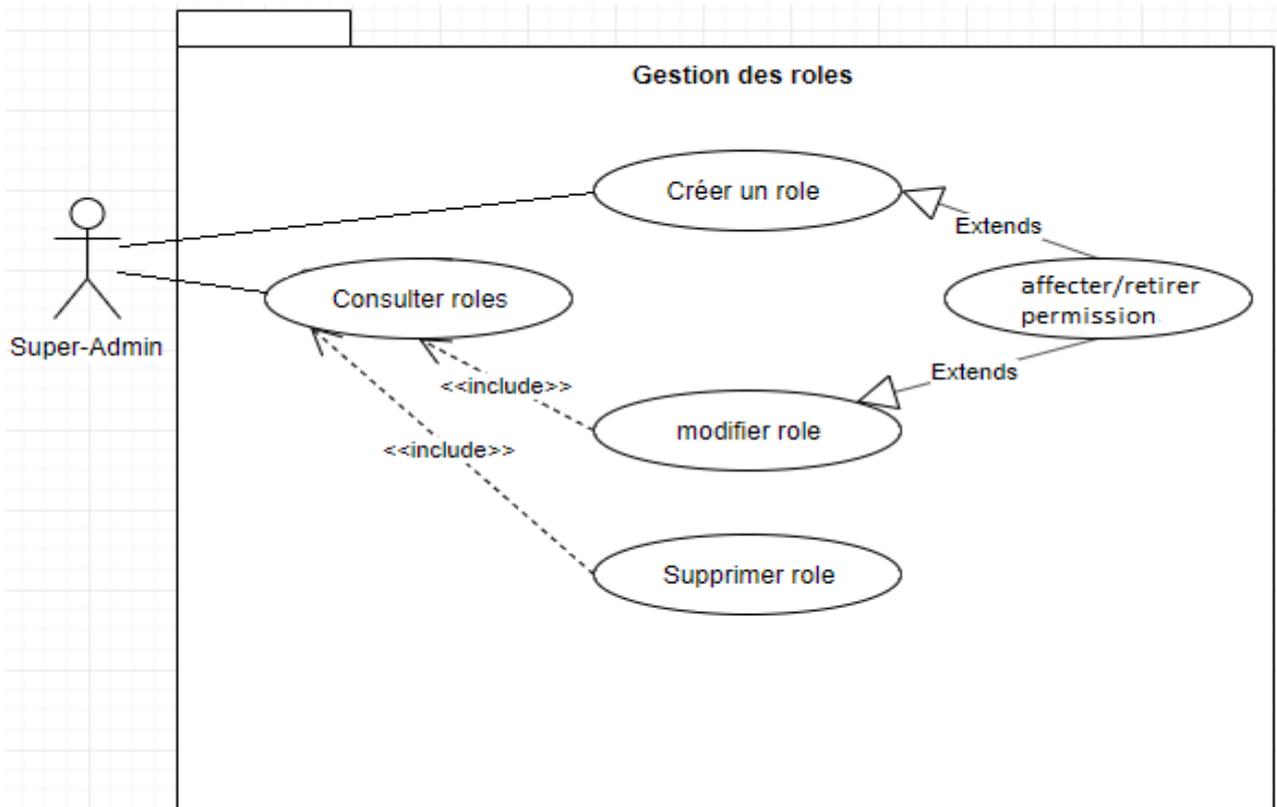


Figure III.2 : Paquetage de cas d'utilisation décrivant « Gérer les rôles »

CU- Gestion des applications	
L'administrateur pourrait :	- Consulter application : ajouter un abonnement mensuel, désactiver l'abonnement lorsque le client n'a pas payé leurs abonnements ou il avait des dettes.
	- Gérer les demandes d'applications des clients (demande espace) : accepter la demande par la création d'une base de données à cette application ou la refuser.
	- Supprimer une application quand un client veut désabonner.
	- Modifier les informations d'une application (par exemple changer le nom d'entreprise cliente).
	- Consulter toutes les informations des applications exister.
Le client peut :	- Lancer des demandes pour créer des nouvelles applications et consulter l'état de ces demandes.

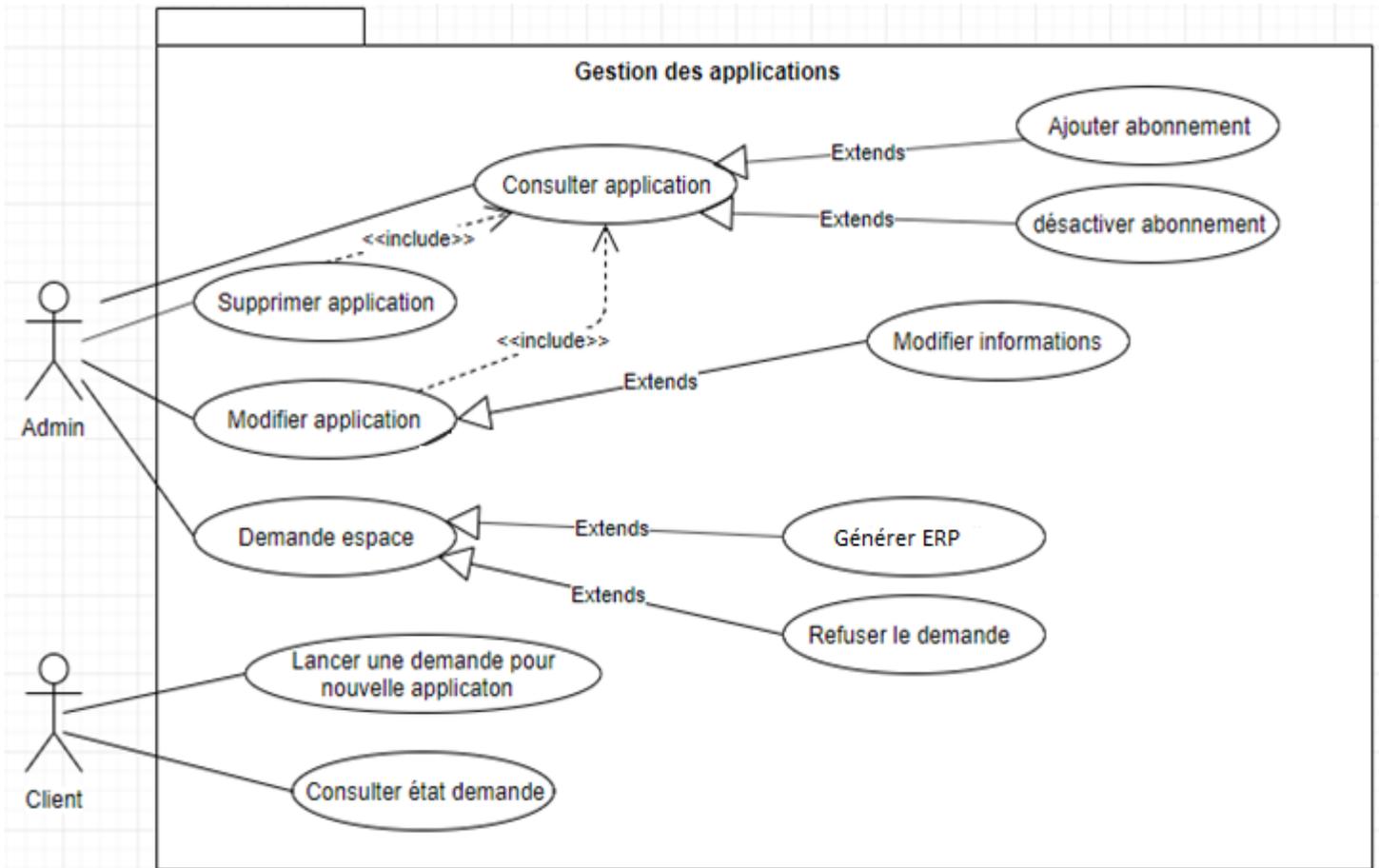


Figure III.3 : Paquetage de cas d'utilisation décrivant « Gestion des applications »

CU-Gestion admins	
Le super administrateur pourrait :	- Créer un utilisateur qui travaille dans le Ctrl-i Manager,
	- Donner le des taches bien défini (par définir son rôle et ses permissions).
	- Modifier les taches ou les rôles de l'utilisateur Ctrl-i Manager.
	- Supprimer un utilisateur Ctrl-i Manager.
	- Consulter toutes les informations des utilisateurs Ctrl-i Manager.
Le Sub-administrateur peut :	- Consulter son compte.

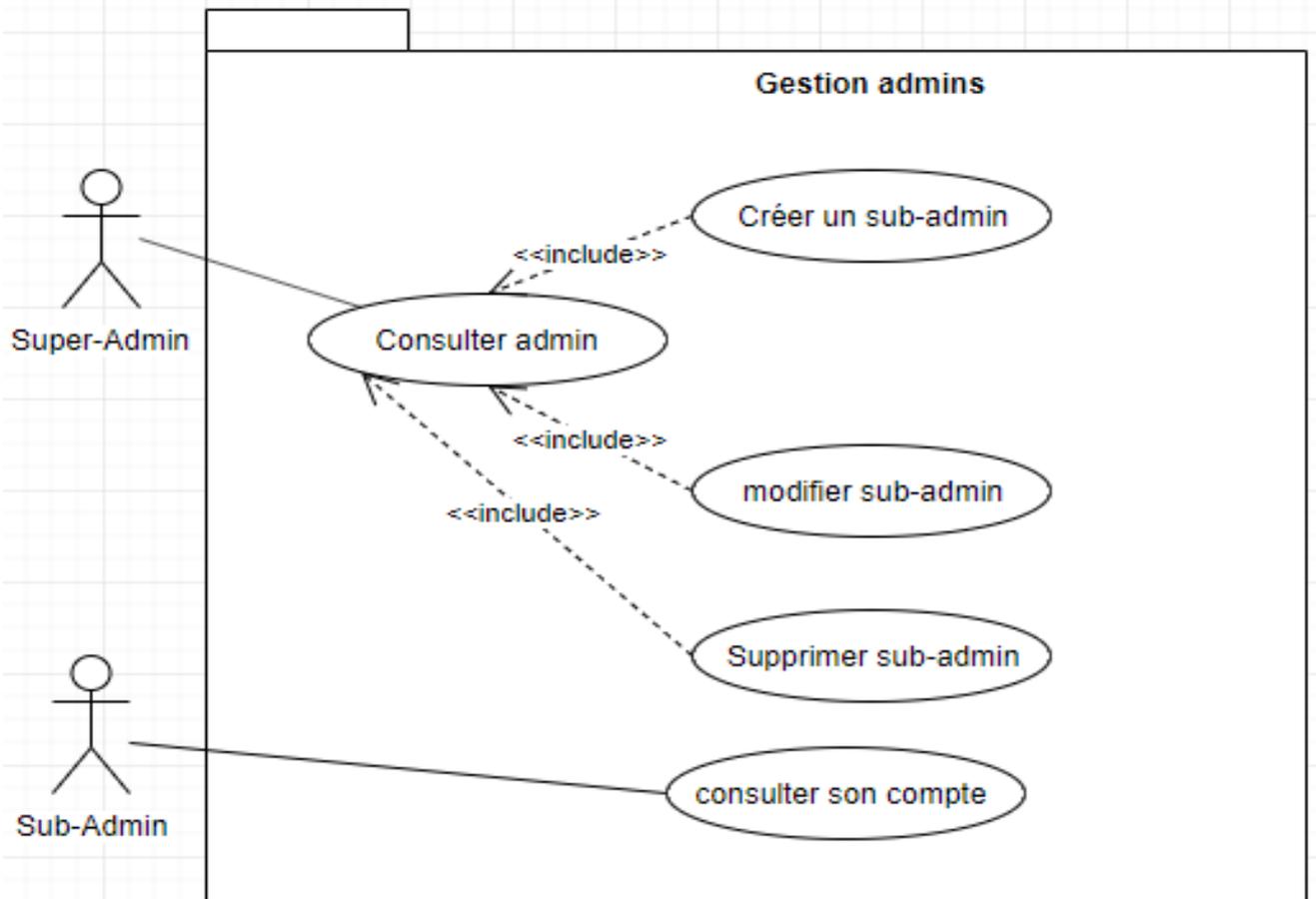


Figure III.4 : Paquetage de cas d'utilisation décrivant « Gérer les Users Ctrl-i »

CU-Gestion des clients	
L'administrateur pourrait :	- Créer un client par créer sa propre base de données et son propre ERP.
	- Modifier les informations du client.
	- Supprimer un client.
	- Consulter toutes les informations des clients.

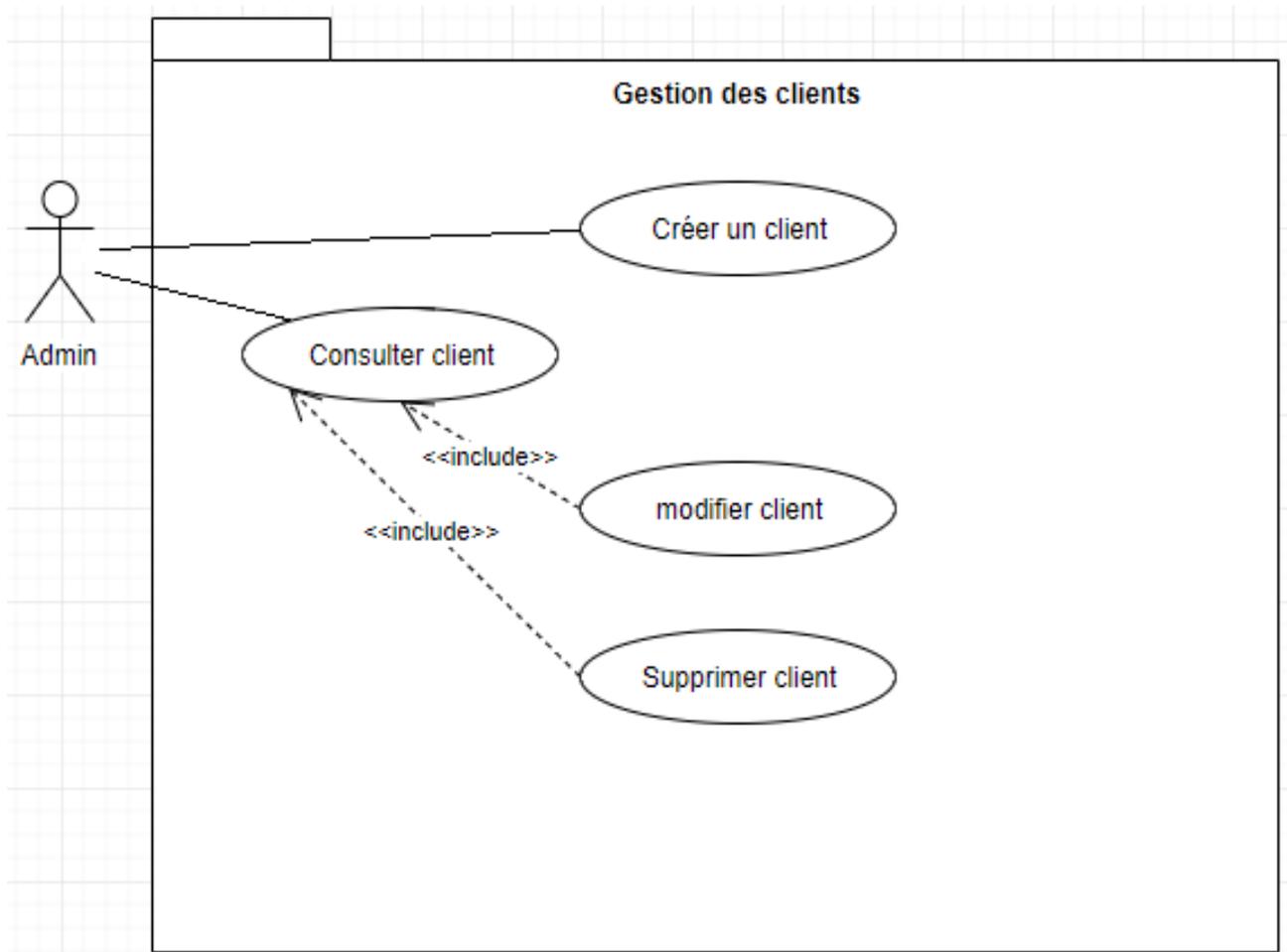


Figure III.5 : Paquetage de cas d'utilisation décrivant « Gestion des clients »

III.3.2. Diagramme de séquence :

Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des messages échangés entre les objets et avec les acteurs.[22]

III.3.2.1. Diagramme de séquence charger le déroulement du Progiciel

- Diagramme de séquence charger le déroulement du client-admin :

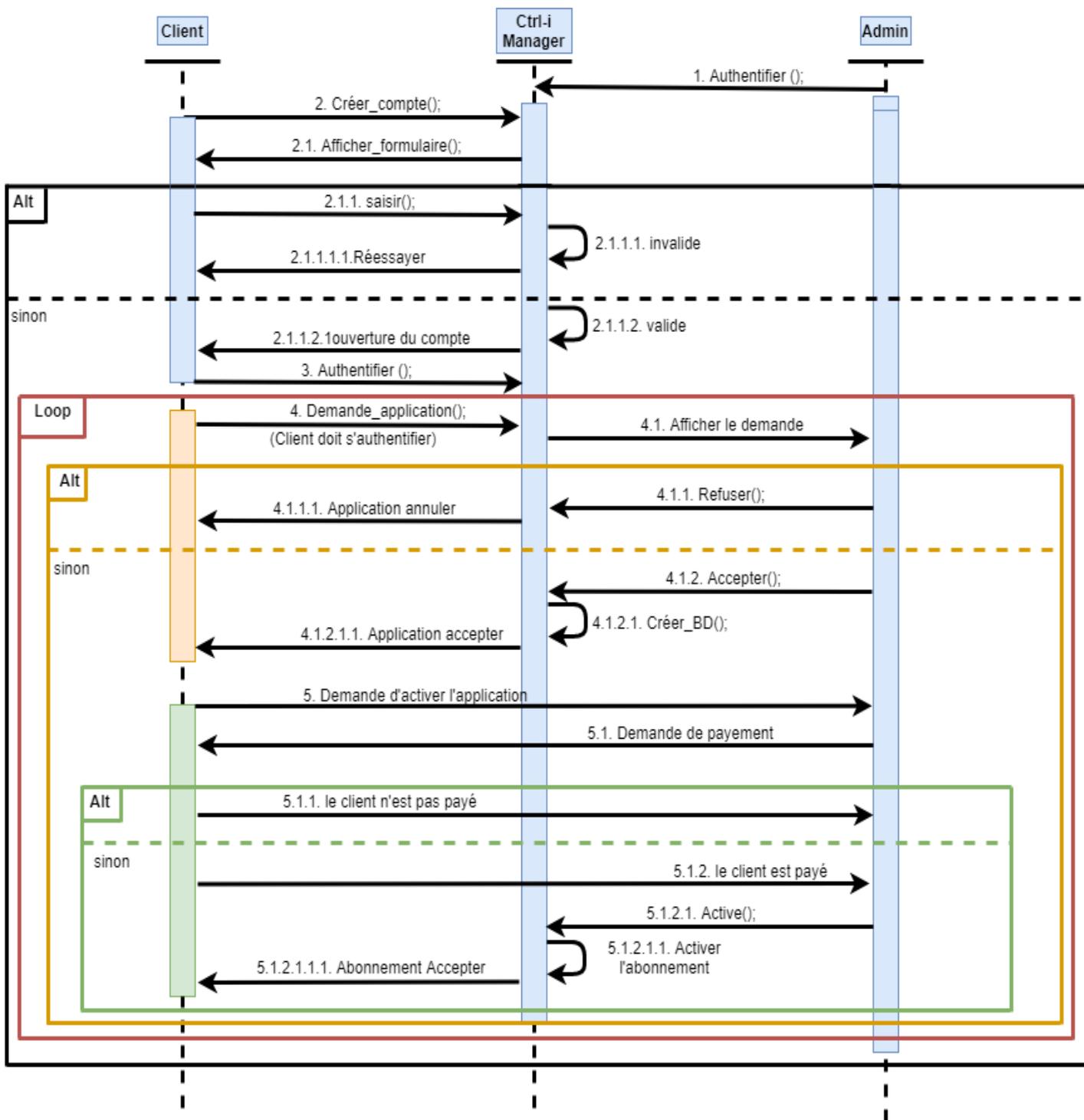


Figure III.6 : Diagramme de séquence charger le déroulement général du rôle

- Diagramme de séquence charger le déroulement général du rôle :

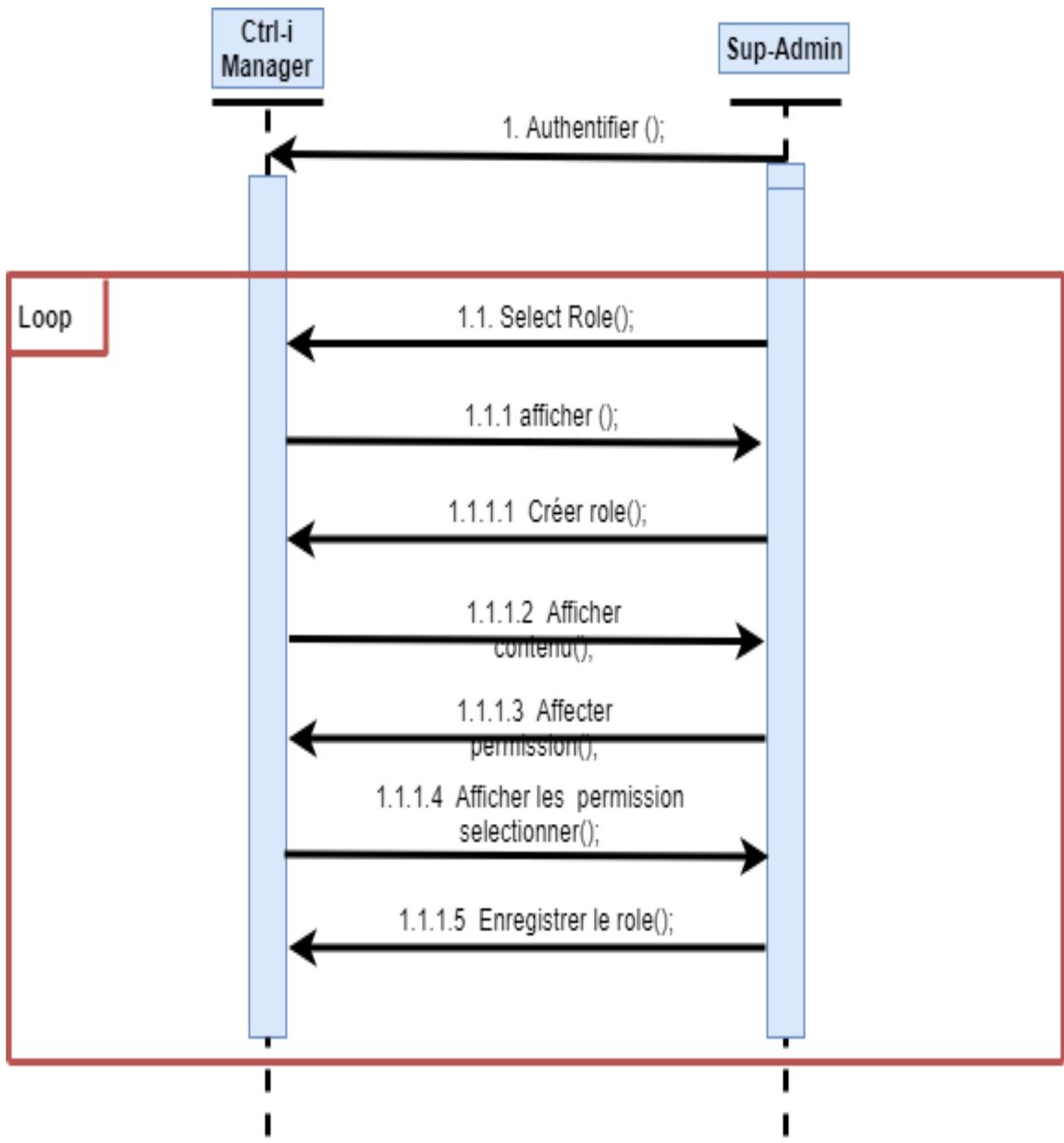


Figure III.7 : Diagramme de séquence charger le déroulement Client-Admin

- Diagramme de séquence charger le déroulement Sup-Admin et Sub-admin :

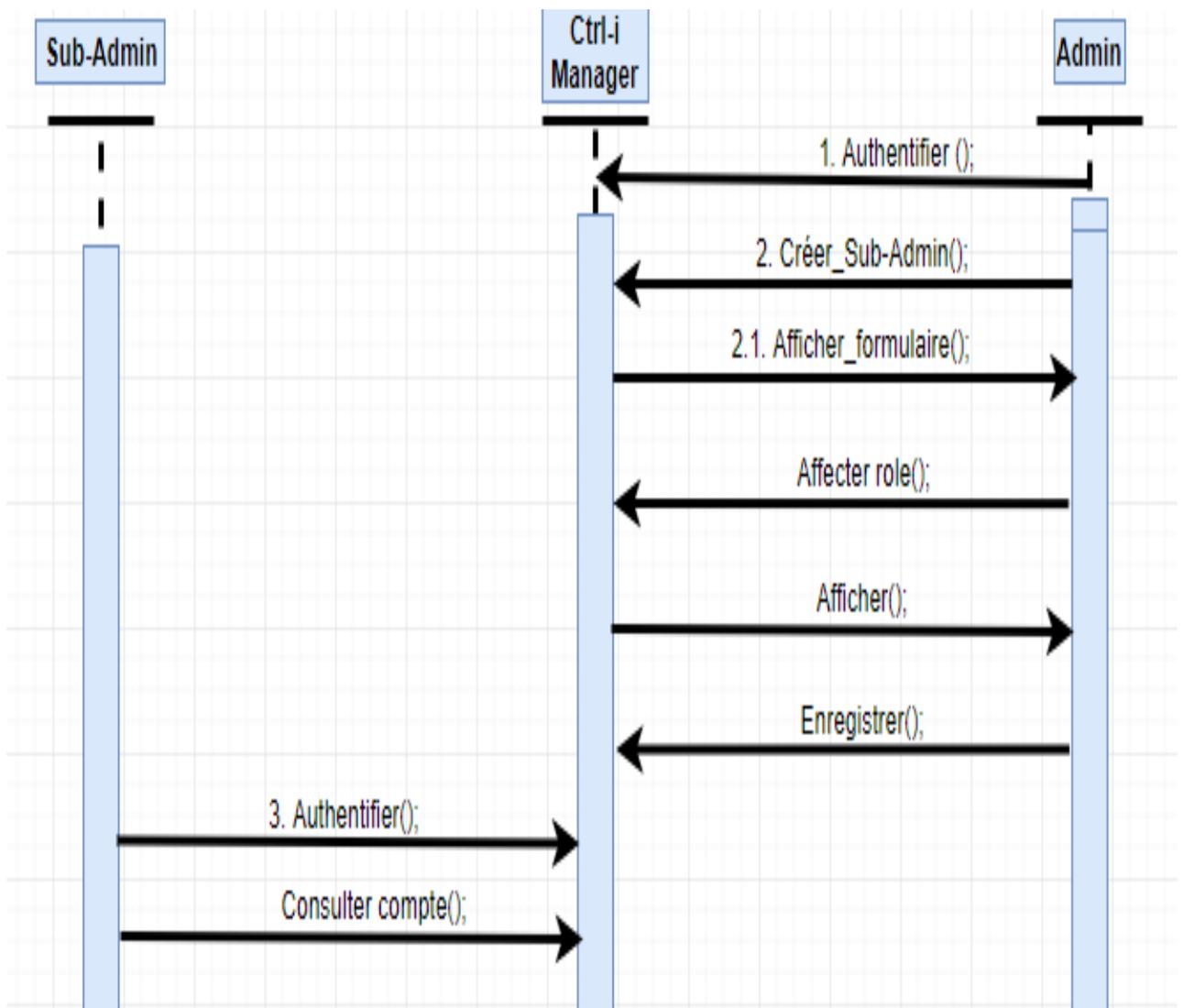


Figure III.8 : Diagramme de séquence charger le déroulement Sup-Admin et Sub-Admin

III.3.3. Diagramme de classe :

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. En conception, le diagramme de classes représente la structure d'un code orienté.[23]

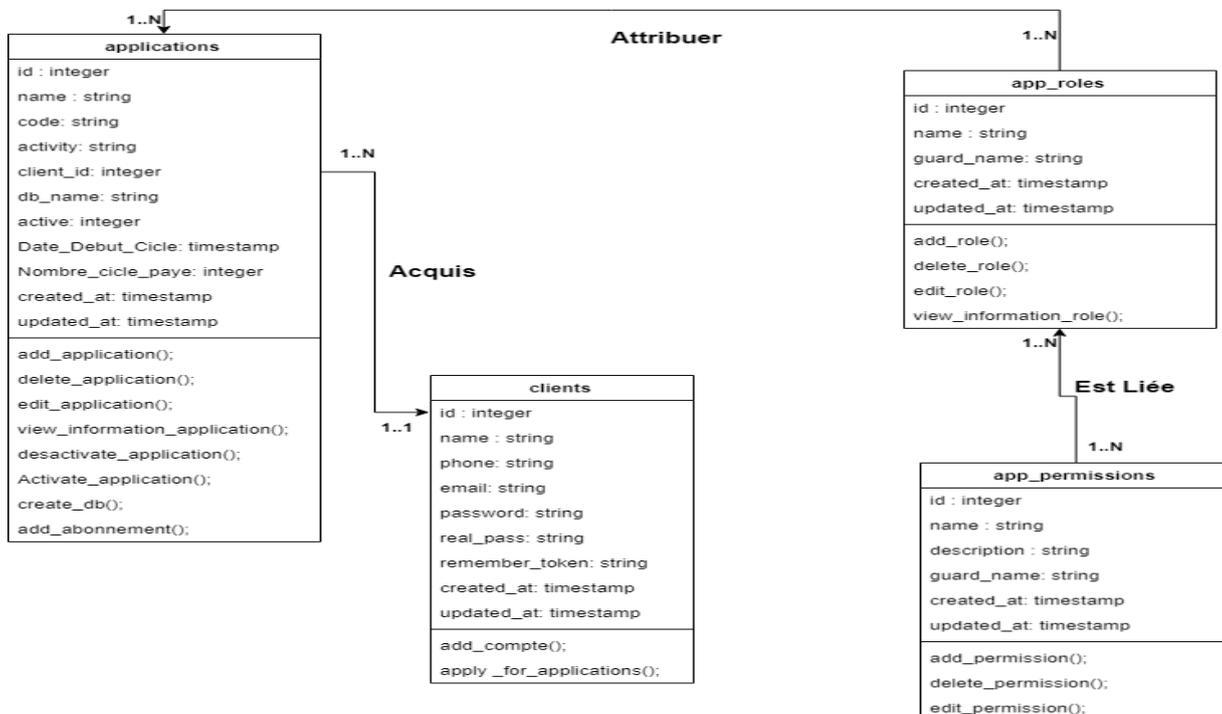


Figure III.10 : Diagramme de Classe de contrôle d'accès utilisateur à base de rôle



Figure III.9 : Diagramme de Classe de contrôle des droits d'exécutions des composants de 'CTRL_i' inspirer du modèle RBAC

III.3.4. Diagramme d'activité du CTRL-i

- Client-Admin (Créer un ERP)

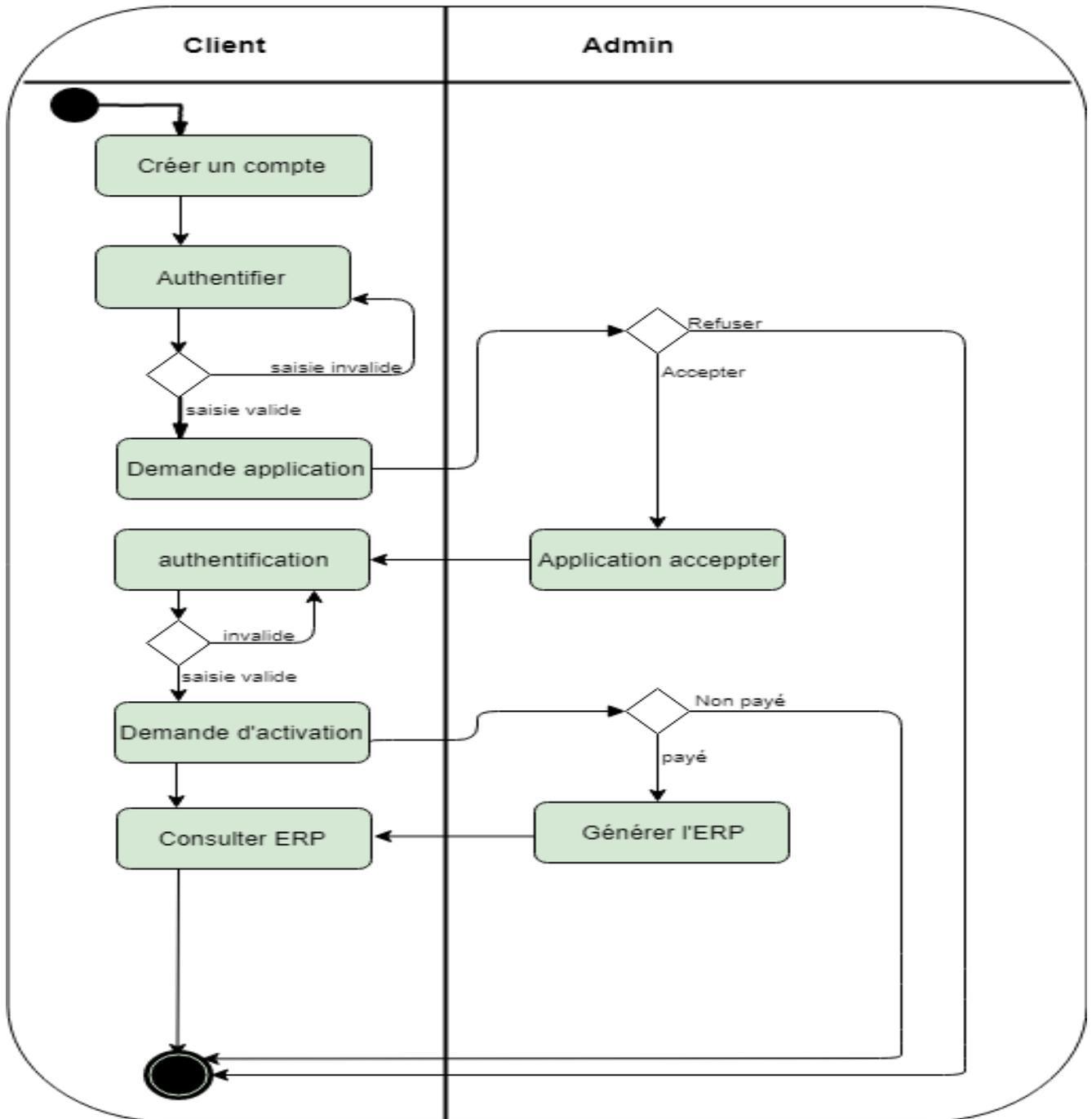


Figure III.11 : Diagramme d'activité client-admin

- Sup-Admin/Sub-Admin:

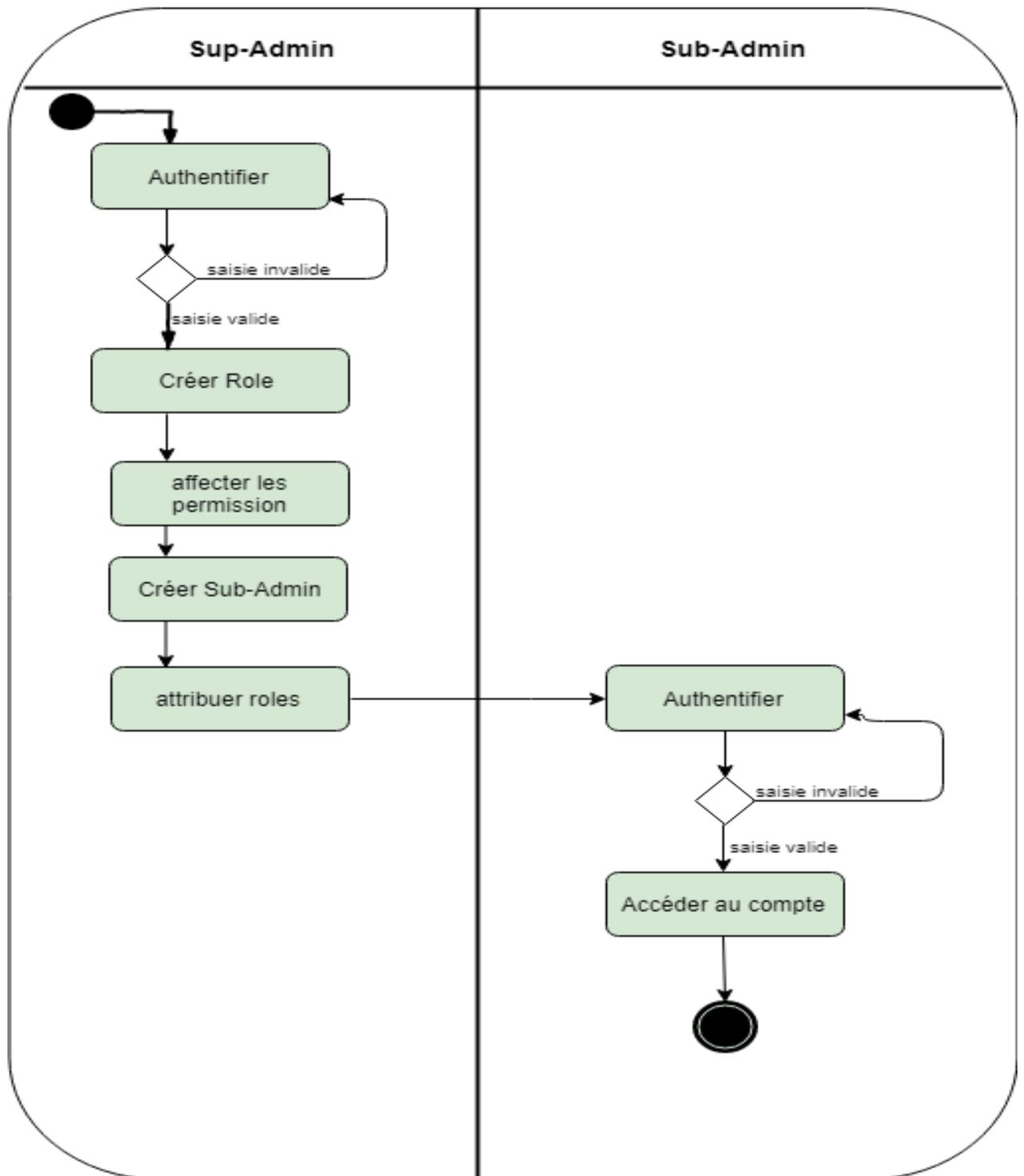


Figure III.12 : Diagramme d'activité Sup-Admin/Sub-Admin

III.4. La Sécurité :

III.4.1. Modèle RBAC :

La modification des contrôles d'accès n'est pas nécessaire chaque fois qu'une personne se joint à une organisation ou la quitte. Par cette caractéristique, RBAC est considéré comme un système « idéal » pour les entreprises dont la fréquence de changement du personnel est élevée. Le contrôle d'accès basé sur le rôle (RBAC) a pris beaucoup d'attention, en particulier, dans le domaine des applications commerciales grâce à son potentiel de réduire la complexité et le coût de la gestion de sécurité sur le réseau. Sous RBAC, la gestion de la sécurité est simplifiée par l'utilisation des rôles, par la hiérarchie et les contraintes pour organiser les privilèges d'accès. Avec RBAC, le concept central des politiques d'accès est le rôle. Les permissions sont affectées à des rôles et les rôles sont affectés à des utilisateurs. Le modèle RBAC manipule cinq éléments administratifs : Utilisateur, Rôle, Opération, Objet et la Permission. RBAC montre des limites surtout qu'il ne permet pas de définir les obligations et les recommandations.

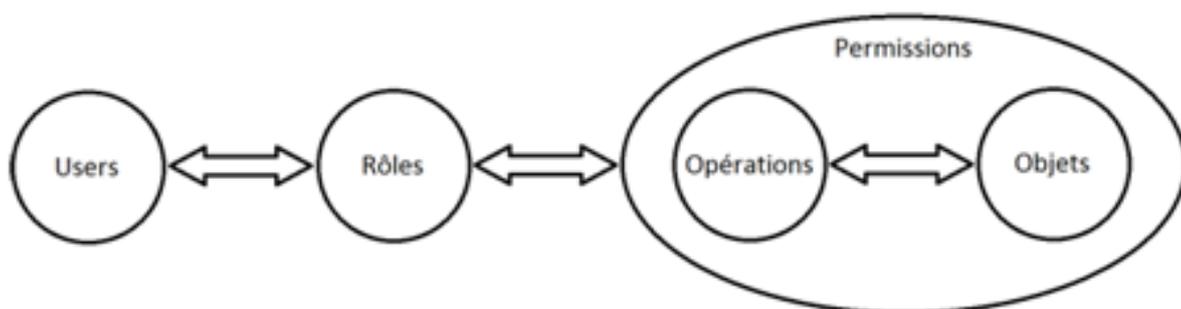


Figure III.13 : Modèle RBAC [44]

III.4.2. Laravel :

Laravel est un Framework web open-source écrit en PHP. Il respecte le modèle MVC (modèle vue contrôleur). C'est un Framework orienté objet distribué sous licence MIT. Laravel offre les fonctionnalités de sécurité suivantes pour permettre aux développeurs de réduire les vulnérabilités de Laravel dans l'application. [17]

Authentication :

Les fonctions d'authentification de Laravel sont essentiellement constituées de gardes (en anglais Guards) et de fournisseurs (en anglais Providers). Les gardes définissent comment les utilisateurs sont authentifiés pour chaque demande. Par exemple, Laravel est livré avec une session gardes qui maintient l'état en utilisant le stockage de session et les cookies.

Les fournisseurs définissent comment les utilisateurs sont extraits de votre stockage persistant. Laravel fournit un support pour récupérer les utilisateurs à l'aide d'Eloquent et du constructeur de requêtes de base de données.

Les Injections SQL :

Une injection SQL est une attaque qui consiste à ajouter à une requête inoffensive un complément qui peut être dangereux pour l'application.

On a utilisé Eloquent ORM pour être automatiquement immunisé contre ces attaques. Il fournit une belle et simple implémentation d'Active Record pour travailler avec la base de données. Chaque table de base de données à un "Modèle" correspondant qui est utilisé pour interagir avec cette table. Les modèles permettent de rechercher des données dans les tables, ainsi que d'insérer de nouveaux enregistrements dans la table.

Eloquent ORM de Laravel utilise une liaison PDO qui protège des injections SQL. Cette fonctionnalité garantit qu'aucun client ne peut modifier l'intention des requêtes SQL.

CSRF :

Laravel facilite la protection de votre application contre les attaques de type CSRF (Cross Site Request Forgery) (en français falsification de requêtes entre sites). Les falsifications de requêtes intersites sont un type d'exploit malveillant par lequel des commandes non autorisées sont exécutées pour le compte d'un utilisateur authentifié.

Laravel génère automatiquement un "jeton" CSRF pour chaque session d'utilisateur actif gérée par l'application. Ce jeton est utilisé pour vérifier que l'utilisateur authentifié est celui qui envoie réellement les demandes à l'application. Chaque fois que vous définissez un formulaire HTML dans votre application, vous devez inclure un champ de jeton CSRF masqué dans le formulaire afin que le middleware de protection CSRF puisse valider la demande. Utiliser la « @csrf » directive Blade pour générer le champ de jeton. Le « VerifyCsrfToken » (Vérifier le jeton Csrf) middleware, qui est inclus

dans le web groupe de middleware, vérifiera automatiquement que le jeton dans l'entrée de requête correspond au jeton stocké dans la session.

XSS (Cross-Site Scripting) :

Le script inter site (XSS) est une attaque par injection de code côté client. L'attaquant vise à exécuter des scripts malveillants dans un navigateur Web de la victime en incluant un code malveillant dans une page Web ou une application Web légitimes.

Par défaut, les instructions Blade sont automatiquement envoyées via la fonction PHP pour empêcher les attaques XSS.

Blade : est le moteur de gabarit simple mais puissant fourni avec Laravel. Contrairement aux autres moteurs de des modèles (templates) PHP populaires, Blade ne vous empêche pas d'utiliser du code PHP simple dans vos vues. En fait, toutes les vues Blade sont compilées dans du code PHP brut et mises en cache jusqu'à ce qu'elles soient modifiées, ce qui signifie que Blade n'a pratiquement aucune surcharge de travail pour l'application.

Chiffrement :

Le crypteur de Laravel utilise OpenSSL pour fournir le cryptage AES-256 et AES-128. Nous avons utilisé les fonctions de cryptage intégrées de Laravel sur notre application et à l'aide de « encrypt » pour chiffrer une valeur. Toutes les valeurs cryptées de Laravel sont signées à l'aide d'un code d'authentification de message (MAC), de sorte que leur valeur sous-jacente ne peut pas être modifiée une fois cryptée.

Pour déchiffrer les valeurs, on utilise « decrypt » de l'assistant.

Hachage :

La Hash façade de Laravel fournit un hachage sécurisé « Bcrypt » pour le stockage des mots de passe des utilisateurs.

Bcrypt est un excellent choix pour le hachage des mots de passe car son "facteur de travail" est ajustable, ce qui signifie que le temps nécessaire pour générer un hachage peut être augmenté à mesure que la puissance du matériel augmente.

Le throttle :

Dans l'authentification une sécurisation de la connexion contre les attaques « brute force ». On trouve le trait Connexion aux manettes (Throttles Logins).

C'est ce trait qui assure la sécurité. À partir d'un certain nombre de connexions avortées (4 connexions pour notre application), on bloque l'accès pour le couple « e-mail et mot de passe » pendant une minute.

III.5. Conclusion :

Ce chapitre a donné une vision sur notre travail, on a analysé les différents cas d'utilisation avec bien détaillé ses paquetages, le diagramme de séquence générale et diagramme de classe qui facilite la compréhension de notre système.

Le chapitre suivant fera l'objet de l'activité d'implémentation de notre application.

Chapitre IV : Implémentation

IV.1. Introduction

Dans ce chapitre, nous allons présenter notre application et décrire : les différents outils qui nous ont servi lors de l'étape de l'implémentation, l'environnement de développement utilisé et les composantes applicatives réalisées d'application, et aussi les différentes tâches.

IV.2. Outils et environnement de développement

Dans Cette section on présente l'environnement logiciel, les différents outils utilisé pour le développement d'application et aussi les langages de programmation que nous avons utilisés. Ensuite, nous illustrons quelques interfaces graphiques de l'application. A la fin nous finissons le chapitre par la phase de test.

IV.2.1. Technologies côtés serveur

IV.2.1.1. PHP :

PHP (officiellement, ce sigle est un acronyme récursif pour PHP HyperTextPreprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML [18].

IV.2.1.2. Framework Laravel :

Laravel est un Framework d'application Web à la syntaxe élégante et expressive. Le développement doit être une expérience agréable et créative pour être réellement épanouissant. Laravel tente de simplifier le développement en simplifiant les tâches courantes utilisées dans la majorité des projets Web, telles que l'authentification, le routage, les sessions et la mise en cache [19].

IV.2.2. Technologies côtés client

IV.2.2.1. HTML5

Le HTML5, pour HyperText Markup Language 5, est une version du célèbre format HTML utilisé pour concevoir les sites Internet et les applications web. Celui-ci se résume à un langage de balisage qui sert à l'écriture de l'hypertexte indispensable à la mise en forme d'une page Web [20].

IV.2.2.2. CSS3

Le terme CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade". Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.

Bien que l'HTML puisse être mis en forme à l'aide de balises prévus à cet effet, de nos jours il est plus judicieux d'utiliser le CSS et de n'utiliser le XHTML que pour le contenu.

L'avantage de l'utilisation d'un fichier CSS pour la mise en forme d'un site réside dans la possibilité de modifier tous les titres du site en une seule fois en modifiant une seule partie du fichier CSS. Sans ce fichier CSS, il serait nécessaire de modifier chaque titre de chaque page du site (difficilement envisageable pour les énormes sites de plusieurs milliers de pages) [21].

IV.2.2.3. JavaScript

JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. À chaque fois qu'une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, etc... — JavaScript a de bonnes chances d'être impliqué. C'est la troisième couche des technologies standards du web, les deux premières (HTML et CSS) [22].

IV.2.2.4. Bootstrap

Bootstrap est une infrastructure de développement front-end ouverte pour la création de sites Web et d'applications Web . Le cadre Bootstrap est construit sur HTML , CSS et JavaScript (JS) pour faciliter le développement de réactifs , mobile premiers sites et applications. Bootstrap comprend des composants d'interface utilisateur, des mises en page et des outils JS, ainsi que le cadre de mise en œuvre.

La conception adaptative permet à une page Web ou à une application de détecter la taille et l'orientation de l'écran du visiteur et d'adapter automatiquement l'affichage en conséquence [23].

IV.2.2.5. JQuery

JQuery est une bibliothèque JavaScript concise et rapide qui peut être utilisée pour simplifier la gestion d'événements, la traversée de documents HTML, les interactions Ajax et l'animation pour le développement rapide de sites Web. JQuery simplifie les scripts côté client HTML, simplifiant ainsi le développement d'applications Web.

JQuery est une bibliothèque libre, à code source ouvert et à double licence sous licence publique générale GNU. Elle est considérée comme l'une des bibliothèques JavaScript (JS) préférées actuellement disponibles [24].

IV.2.2.6. Ajax

L'architecture informatique ajax (acronyme d'asynchrones JavaScript and XML : JavaScript et XML asynchrones) permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web entre 1995 et 2005.

Ajax combine JavaScript, les requêtes de type XMLHttpRequest, les manipulations du DOM, ainsi qu'un format de données (XML ou JSON), afin d'améliorer maniabilité et confort d'utilisation des applications internet riches [25].

IV.2.2.7. DataTables

DataTables est un plug-in pour la bibliothèque Javascript de jQuery. C'est un outil extrêmement flexible, construit sur les fondements d'une amélioration progressive, qui ajoute toutes ces fonctionnalités avancées à n'importe quel tableau HTML. [26].

Voici quelques fonctionnalités de ce plugin :

- La recherche instantanée, le tri.
- La pagination.
- La possibilité de travailler avec plusieurs sources de données : DOM, JavaScript, Ajax.

IV.2.3. La base de données

IV.2.3.1. MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server [27].

IV.2.3.2. XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus [28].

IV.2.3.3. Serveur Apache

Apache est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows. Le pack PHPdev (désormais EasyPHP) est ainsi téléchargeable, il regroupe les applications suivantes : [29]

- Le serveur web Apache
- Le serveur de bases de données MySQL
- Le serveur d'application PHP
- L'outil phpMyAdmin permettant de gérer des bases MySQL

IV.3. Présentation d'Application

IV.3.1. Page Login client

La page Login du client contient deux champs, l'un pour l'email et l'autre pour le mot de passe, et à cause de ces deux champs le client peut accéder à son compte.

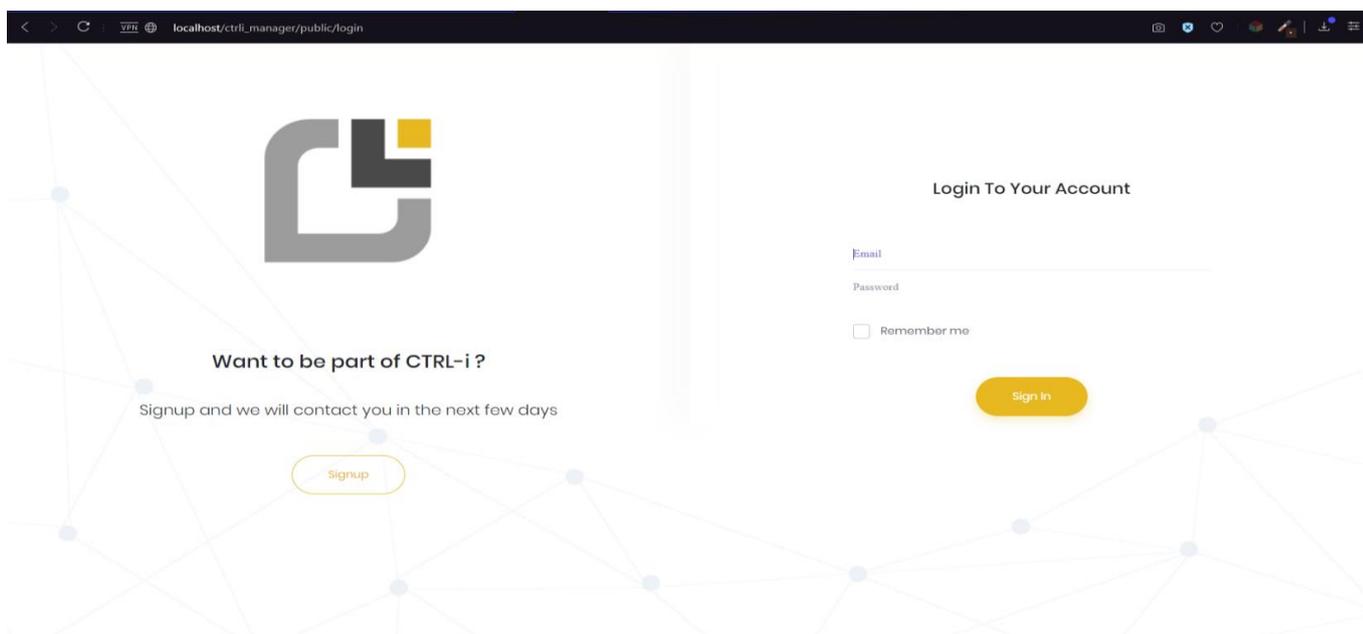


Figure IV.1 : Capture d'écran « Page Login client »

IV.3.2. Page Création de compte client

Vu que le client pour puisse demander un espace d'ERP, il faut avoir un compte, et pour cela cette page a la possibilité de créer un compte à travers de remplir un formulaire qui contient 6 champs.

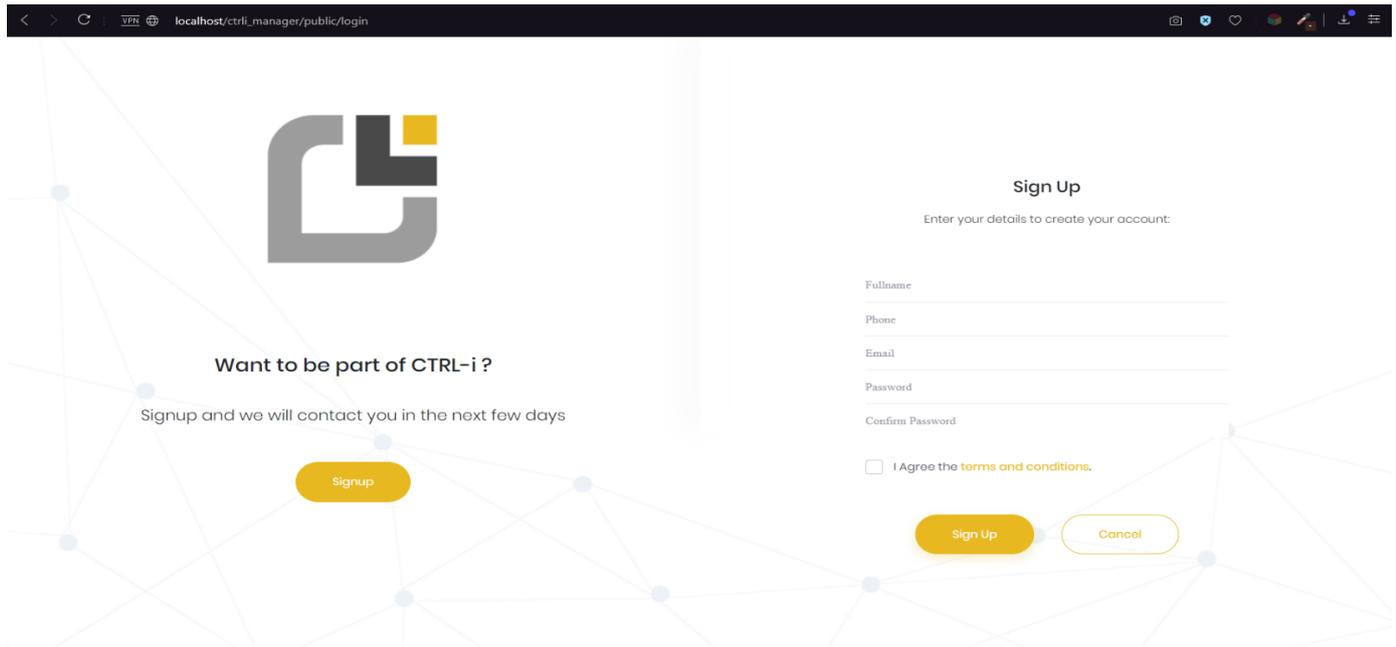


Figure IV.2 : Capture d'écran « Page Création de compte client »

IV.3.3. Page Demande d'Applications

Lorsqu'un client rentre dans son compte, il peut consulter ses demandes d'espace d'application (ERP CTRLi). Cette page contient 2 fonctionnalités :

- Permettre au client de se déconnecter à travers le bouton « Logout ».
- Demande un ou plusieurs espaces d'ERP.

Si le client veut un autre espace il suffit de cliquer sur le bouton « Apply for new one ».

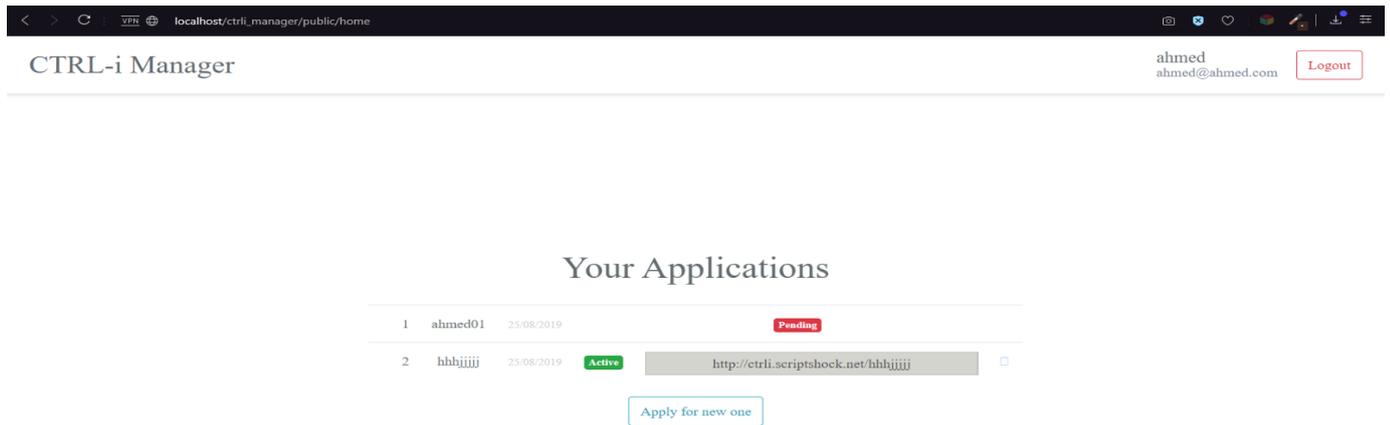


Figure IV.3 : Capture d'écran « Page Demande d'Applications »

IV.3.4. Page Login Admin

Avec cette page qui implique un champ d'email et un champ mot de passe, vous pouvez rentrer et utiliser l'application CTRL-i Manager.

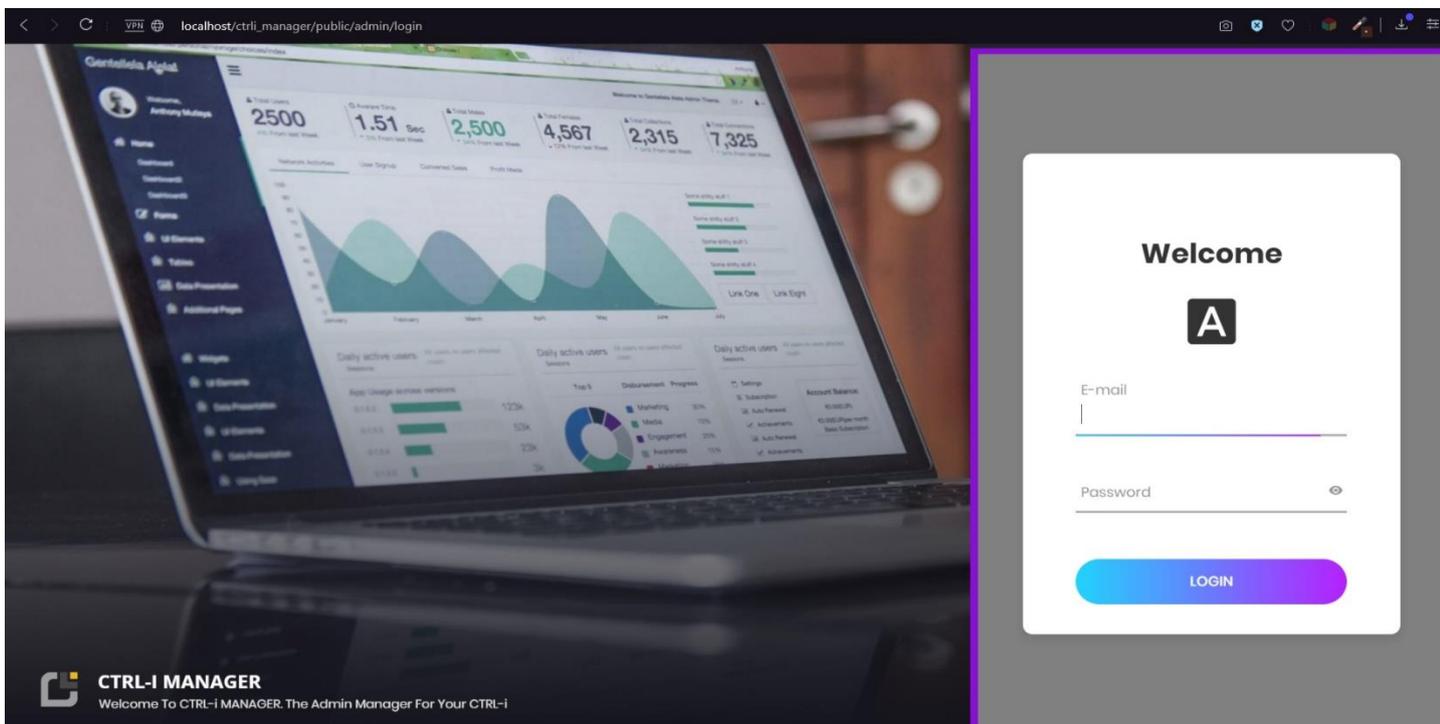


Figure IV.4 : Capture d'écran « Page Login Admin »

IV.3.5. Page Welcome Admin

Une page d'accueil avec une simple phrase de salutation lorsque les utilisateurs accèdent à l'application « CTRL-i Manager ». Elle a plusieurs fonctionnalités tel que :

- Permettre aux utilisateurs de se déconnecter à travers le bouton « Logout ».
- Naviguer dans le menu qui se présente à gauche de la page.

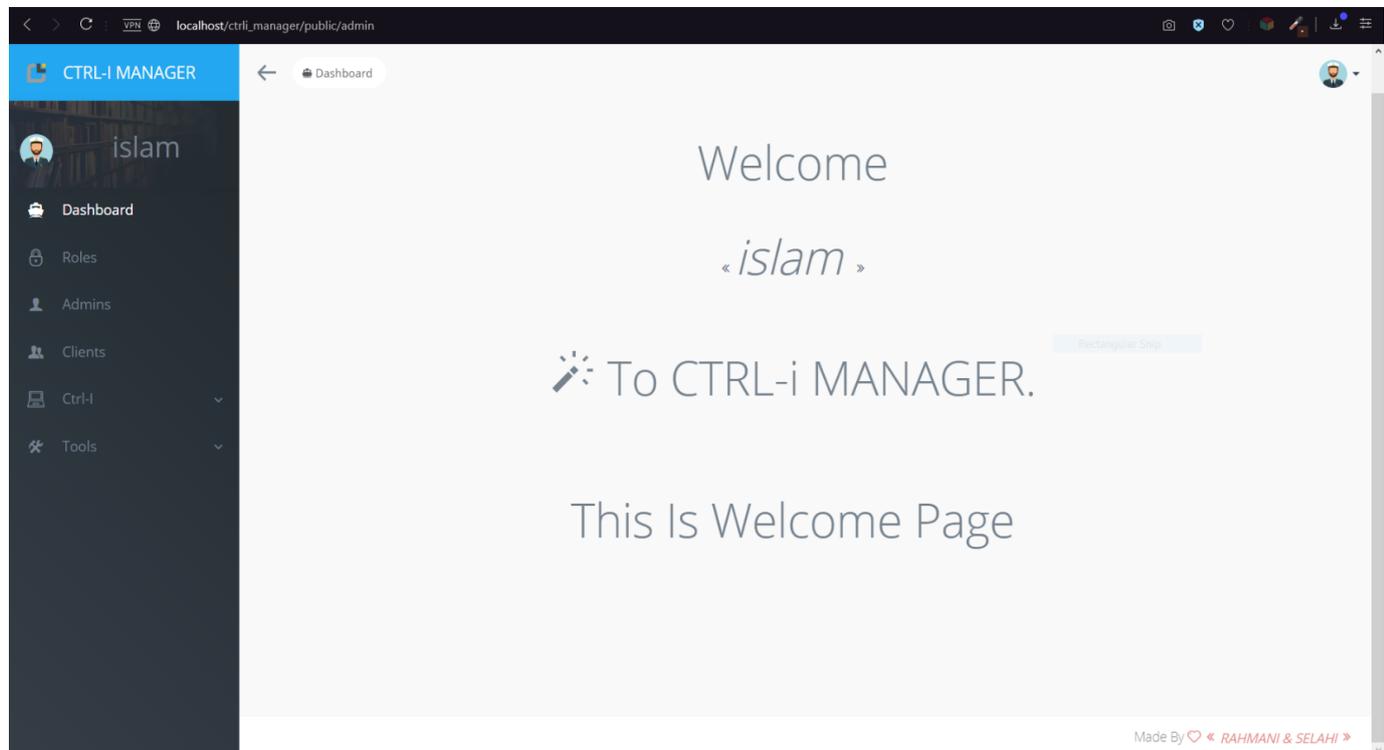


Figure IV.5 : Capture d'écran « Page Welcome Admin »

IV.3.6. Gestion des Rôles

Il y'a plusieurs fonctions qui peut seule admin (ou les sub-admin, s'ils ont les permissions) d'utiliser tel que :

- Ajouter ou modifier les rôles (affecter ou retirer des permissions).
- Chercher ou supprimer un rôle.
- Afficher seulement 10,25,100,500 rôles dans le tableau.

IV.3.7. Gestion des Admins

L'application offre au admin d'ajouter des sub-admins ou les supprimer ainsi que consulter ses détails, et elle offre aussi le choix du rôle pour le sub-admin.

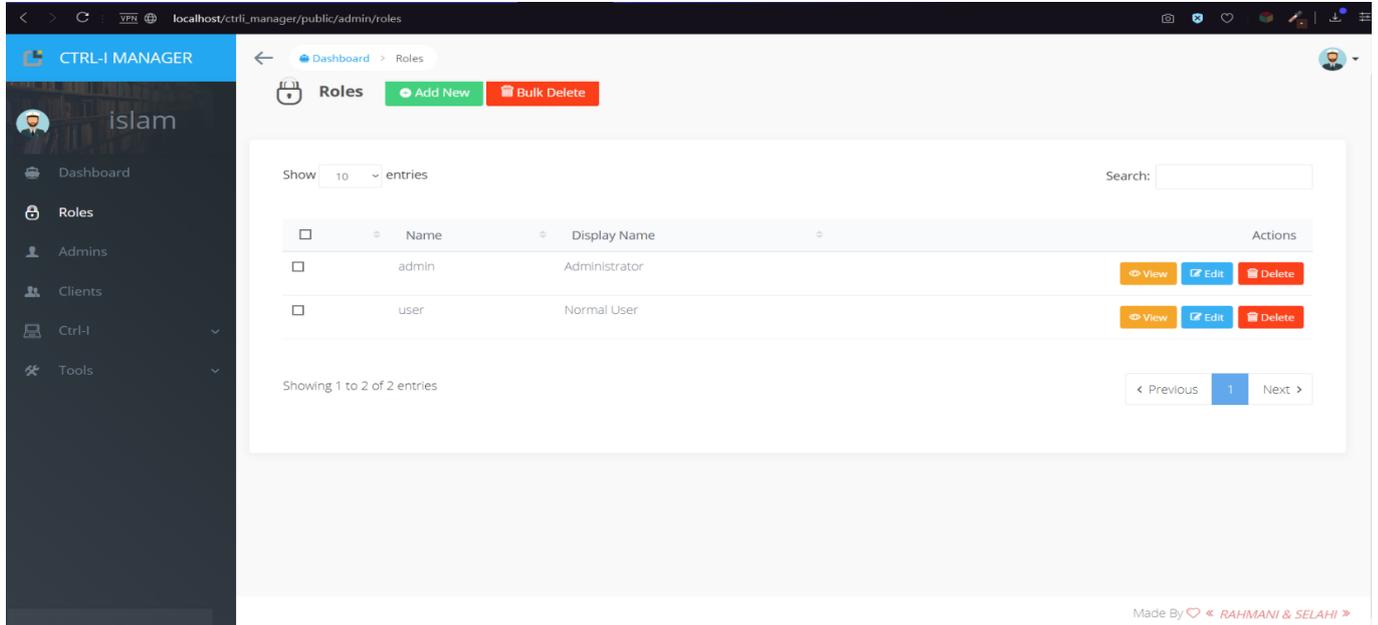


Figure IV.6 : Capture d'écran « Gestion des Rôles »

IV.3.8. Gestion des Clients

Cette page contient toutes les informations des clients lesquels : le nom, l'email, le numéro de téléphone, les applications de chaque client.

L'admin a le pouvoir de supprimer des clients et aussi modifier ses informations.

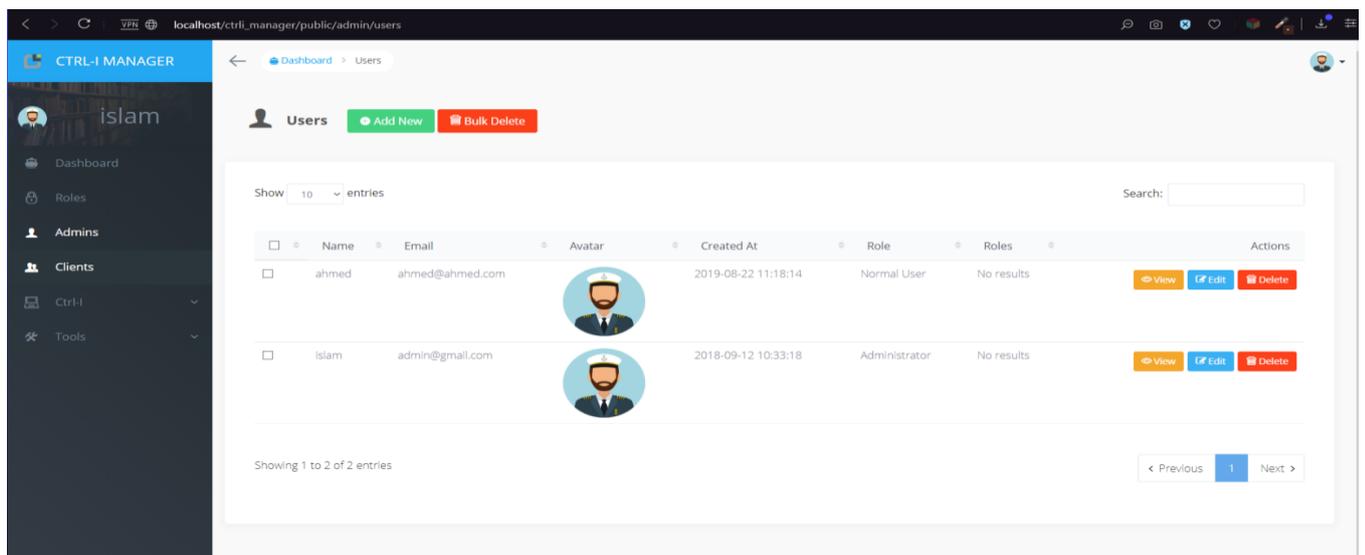


Figure IV.7 : Capture d'écran « Gestion des Admins »

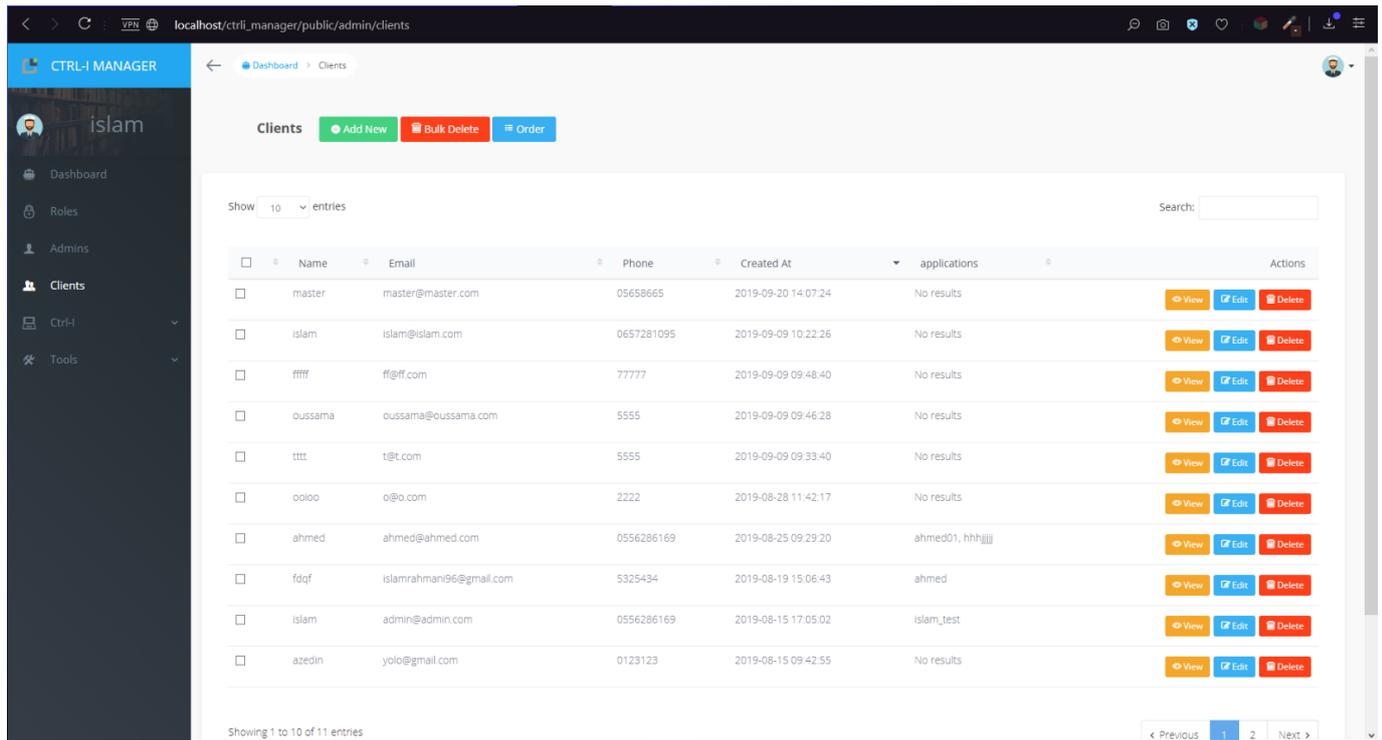


Figure IV.8 : Capture d'écran « Gestion des Clients »

IV.3.9. Gestion des Applications

Afin de garder un œil sur les états des applications du client, CTRL-i Manager offre un certain nombre d'actions sous formes des boutons qui manipule l'état de la base de données et aussi l'abonnement de chaque client.

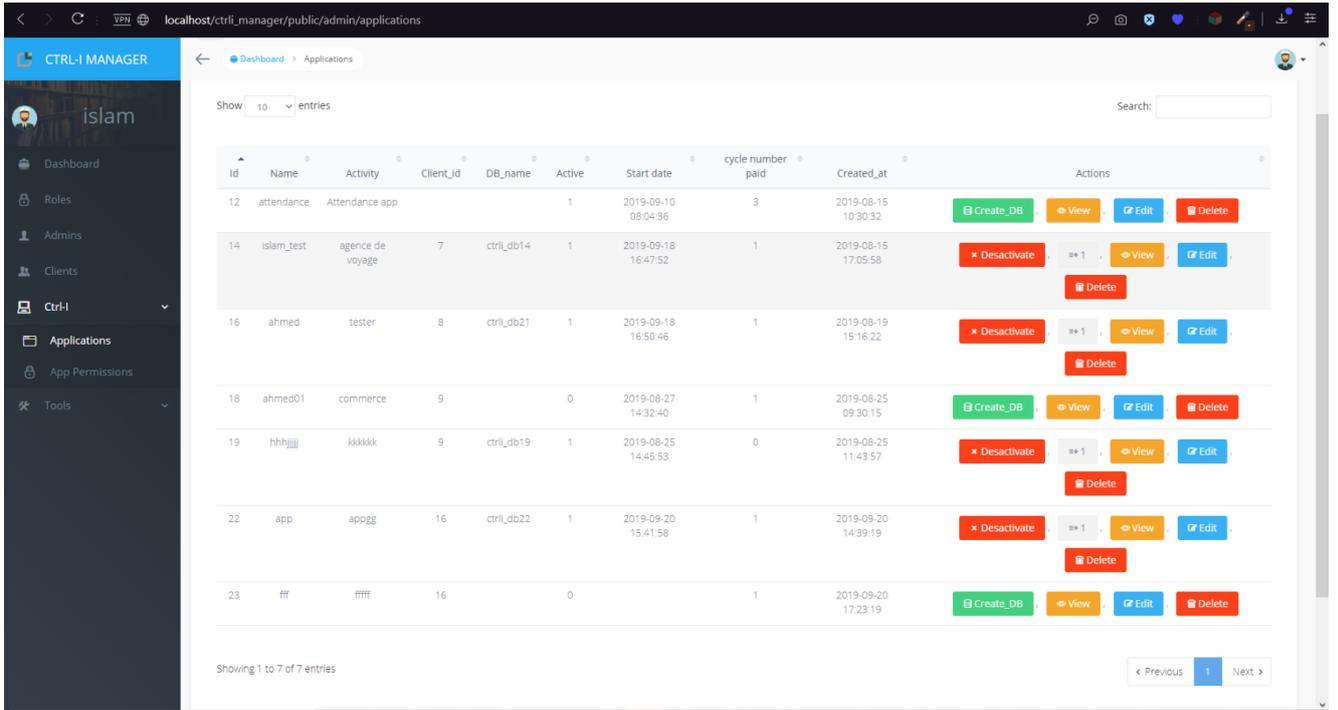


Figure IV.9 : Capture d'écran « Gestion des Applications »

IV.3.10. Gestion des Permissions d'applications

Le titre de menu « App Permissions » permet de contrôler les permissions de chaque application (ERP), selon les besoins de client.

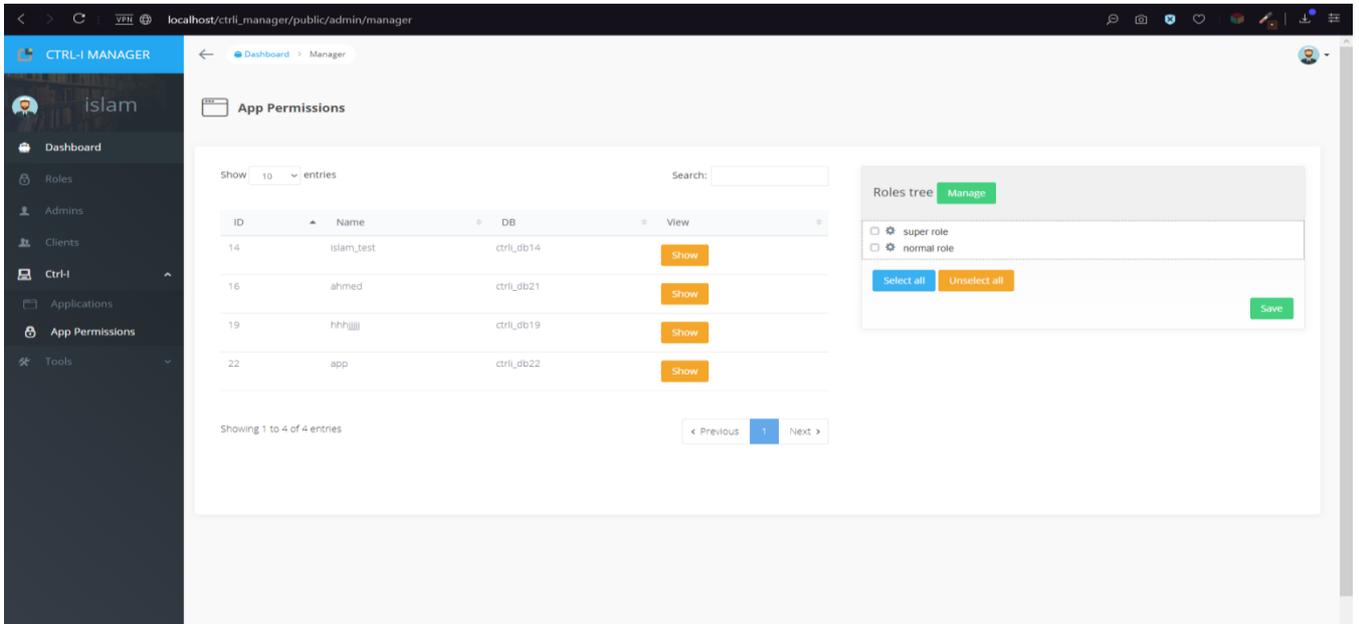


Figure IV.10 : Capture d'écran « Gestion des Permissions d'applications »

IV.4. Conclusion :

Ce chapitre a bien expliqué les outils et les méthodes utilisés pour l'implémentation de l'application « CTRL-i MANAGER », les différentes interfaces donne une vue complète sur l'utilité de cette application.

Conclusion Générale

La problématique de décentraliser la gestion des multiples clients sur la plateforme « CTRL-i » abordée dans ce mémoire s'est cristallisée autour de deux points : le premier concerne les définitions et les besoins des clients pour avoir une distribution d'application qui adapté avec de ces besoins. Le deuxième point se focalise sur le choix d'approche qui peuvent aider les fournisseurs de services applicatifs à définir des moyens d'améliorer la fiabilité, les performances et pour un bon rendement aux entreprises clientes. Pour comprendre et simplifier le sujet, nous avons présenté dans le premier chapitre une généralité sur l'ERP et les différents modèles d'où des solutions de notre problématique. Dans le deuxième chapitre une généralité sur le Cloud Computing et une étude sur notre solution SaaS avec un état comparatifs entre les modèles du chapitre 1 et le modèle SaaS. Dans le chapitre trois identifier toutes les fonctionnalités de notre future application pour chaque type d'utilisateurs y compris ces besoins fonctionnels. Dans le chapitre quatre nous allons présenter notre application et décrire : les différents outils qui nous ont servi lors de l'étape de l'implémentation, l'environnement de développement utilisé et les composantes applicatives réalisées d'application, et aussi les différentes tâches.

En conclusion, ce travail montre que les modèles appliqués sur la fourniture aux différents clients ne répondent pas aux besoins et compliquent la vie du travail du client par contre le modèle SaaS traite automatiquement les besoins clients.

Dans notre application toutes les données échangées sont envoyées en clair. Ainsi, tout le monde pourrait facilement se faire voler entre la transmission. Par conséquent, pour protéger ces informations nous proposons comme perspectives on travaille sur ce problème qui peut être résolu avec le passage vers HTTPS afin de protéger les informations sensibles.

Références :

- [1] : Klaus, H., Rosemann, M., & Gable, G. G. (2000). What is ERP? Information systems frontiers, 2(2), 141-162.
- [2] : Marnewick, C., & Labuschagne, L. (2005). A conceptual model for enterprise resource planning (ERP). Information management & computer security, 13(2), 144-155.
- [3] : Hendricks, K. B., Singhal, V. R., & Stratman, J. K. (2007). The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations. Journal of operations management, 25(1), 65-82.
- [4] : Topi, H., Lucas, W. T., & Babaian, T. (2005, May). Identifying Usability Issues with an ERP Implementation. In ICEIS (pp. 128-133).
- [5] : <https://erp.ooreka.fr/comprendre/architecture-erp>
- [6] : Blain, F. A. (2006). Présentation générale des ERP et leur architecture modulaire.
- [7] : <https://www.celge.fr/article-conseil/logiciel-on-premise>
- [8] : <https://www.inasoft.fr/blog/actualites-logiciel-rh/logiciel-mode-saas-et-on-premise-quel-sont-les-differences/>
- [9] : <https://www.ionos.fr/digitalguide/serveur/know-how/logiciels-on-premise-vs-cloud/>
- [10] : <https://www.celge.fr/article-conseil/on-premise-preferable-au-saas>
- [11] : <https://www.planzone.fr/blog/logiciel-saas-vs-on-premise-lequel-choisir-pour-vos-projets>
- [12] : <https://www.investopedia.com/terms/c/cloud-computing.asp>
- [13] : <https://www.infoworld.com/article/3226386/what-is-saas-software-as-a-service-defined.html>
- [14] : TONIC François, Livre Blanc « les enjeux du SaaS » disponible sur le site : <http://mySaaS.fr>
- [15] : https://www.itu.int/dms_pub/itu-t/md/13/sg17/r/T13-SG17-R-0052!!MSW-F.docx
- [16] : http://www.nicolasgrevet.com/files/mr09_ngrevet_cloudcom.pdf
- [17] : <https://laravel.com/docs/5.8>
- [20] : [https://fr.wikipedia.org/wiki/UML_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique))

- [21] : https://www.ibm.com/support/knowledgecenter/fr/SS8PJ7_9.5.0/com.ibm.xtools.modeler.doc/topics/ucd.html
- [22] : <http://remy-manu.no-ip.biz/UML/Cours/coursUML5.pdf>
- [23] : <http://remy-manu.no-ip.biz/UML/Cours/coursUML3.pdf>
- [24] : <https://www.php.net/manual/fr/intro-what-is.php>
- [25] : <https://laravel.com/docs/4.2/introduction>
- [26] : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction/>
- [27] : <https://fr.wikipedia.org/wiki/MySQL>
- [28] : <https://desgeeksetdeslettres.com/web/xampp-plateforme-pour-heberger-son-propre-site-web>
- [29] : <https://www.commentcamarche.net/contents/8-apache-introduction>
- [30] : <http://glossaire.infowebmaster.fr/css/>
- [31] : https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [32] : <https://whatis.techtarget.com/definition/bootstrap>
- [33] : <https://www.techopedia.com/definition/3977/jquery>
- [34] : [https://fr.wikipedia.org/wiki/Ajax_\(informatique\)](https://fr.wikipedia.org/wiki/Ajax_(informatique))
- [35] : <https://datatables.net/>

Les nouveaux references pour les figures:

- [36] : <http://erp-als.over-blog.com/article-fonctionnement-de-l-erp-59051450.html>
- [37] : <https://blog.cenareo.com/bien-comprendre-laffichage-dynamique>
- [38] : <http://www.mail-tracking.ch/logiciels-emailing/quelle-solution-emailing-choisir.html>
- [39] : <https://go4hosting.com/blog/cloud-hosting/what-is-cloud-computing>
- [40] : <https://newsapisecurityonbigdata.wordpress.com/>
- [41] : http://democritique.org/IT/Documents/Cloud_Computing/Martin_Sarazin_M%C3%A9moire_Master_MIAGE.pdf
- [42] : <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>
- [43] : <http://www.sharepointcrmtemplate.com/blog/comparing-crm-total-cost-of-ownership-cloud-vs-on-premise/>
- [44] : <https://www.synbioz.com/blog/tech/role-based-access-control>