

Table des matières

Remerciements	
Dédicace	
Introduction générale.....	1
Chapitre I : Contexte du stage	3
Introduction.....	4
I.1 L'entreprise <i>DigitalSide</i>	4
I.2 Présentation de DigitalSide	4
I.2.1 La forme juridique	4
I.2.2 Vocation	4
I.3 L'organisation de DigitalSide.....	5
I.3.1 L'équipe fonctionnelle	5
I.3.2 L'équipe technique	5
I.4 Contexte du stage.....	6
I.4.1 Moyens matériels	6
I.4.2 Moyens logiciels	6
I.5 Objectifs du projet.....	9
I.5.1 Problématique.....	9
I.5.2 Finalité.....	9
Conclusion	9
Chapitre II : Présentation des outils de développement	10
Introduction.....	11
I.1 Modules Odoo	11
II.1.1 modules Front Office	12
II.1.2 modules Back Office.....	12
II.2 Architecture Logiciel.....	13
II.3 Fonctionnalités ajoutées	15
II.4 Versions Odoo	17
II.5 Licence AGPL	18
II.6 L'environnement PostgreSQL	18
II.7 Langage de programmation Python	19

II.8 Langage structurel XML	19
II.9 VOIP	20
II.10 Design Patterns.....	20
II.10.1 Description des design patterns	20
II.10.2 MVC Pattern	21
II.11 Processus UP.....	21
II.11.1 Définition	21
II.11.2 Caractéristiques.....	21
II.11.3 Vie du processus Unifié.....	23
Conclusion	24
Chapitre III : Méthodologies	25
Introduction.....	26
III. Analyse des besoins.....	26
III.1 Rédaction d'un questionnaire	26
III.2 Recueil des besoins	29
III.3 Analyse de l'existant.....	29
III.4 Rédaction du cahier des charges.....	30
III.5 Cahier de conception	35
III.5.1 Méthode de conception.....	35
III.5.2 Conception générale	36
III.5.3 Diagramme de cas d'utilisation.....	37
1 Identification des acteurs	38
2 Fiche descriptive des cas d'utilisation.....	38
III.5.4 Diagramme de classes.....	47
Conclusion	48
Chapitre IV : Implémentation.....	49
Introduction.....	50
IV.1 Structure d'un module	50
IV.1.1 Notion d'objet dans Odoo.....	54
IV.2 Workflow.....	55
IV.3 Captures d'écrans	56

Conclusion générale	65
Références bibliographiques.....	67
Liste des figures.....	68
Liste des tableaux.....	69
Liste des acronymes	69

- INTRODUCTION GENERALE -

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), le fonctionnement interne de l'entreprise (Business To Employees) et la relation de l'entreprise avec ses différents partenaires et fournisseurs (Business To Business, B2B). On appelle aussi "e-Business" l'intégration au sein de l'entreprise d'outils basés sur les technologies de l'information et la communication, en l'occurrence les **Progiciels de Gestion Intégré (PGI)** ou **Enterprise Resource Planning (ERP)**. Cet outil permet une gestion homogène et cohérente du système d'information (SI) de l'entreprise.

Dans ce mémoire, nous présentons la réalisation d'un module call Center (En français centre d'appel). Ce module va être intégré comme un sous module du module CRM (*Customer Relationship Management*) disponible sur Odoo. Notre travail a été réalisé au sein de l'Entreprise DigitalSide, qui nous a encadrés durant notre stage, d'une durée de 15 jours à Alger. L'entreprise nous a même confié un de leur client, Karoura une entreprise qui commercialise dans la production, et la vente de bombonnes d'eau fraîche. Le module call center qu'on a réalisé pour l'entreprise client, va permettre d'assister les utilisateurs des produits de Karoura, pour être plus à l'écoute de leurs clients. A la fin L'entreprise Karoura peut bénéficier d'une bonne gestion commerciale de sa chaine de production d'un côté, et de ces clients d'un autre côté. Pour arriver à la réalisation de cela, il a fallu exercer et mettre notre savoir-faire en Génie logiciel en Pratique, comme tous autres personnels de l'entreprise DigitalSide. Ainsi que l'assistance et le suivi de notre encadrant académique, pour arriver à un résultat correcte et complet.

Dans ce mémoire, nous présentons en détail les étapes que nous avons suivies pour réaliser notre application. Le mémoire est organisé comme suit :

Le premier Chapitre : est un chapitre introductif, concernant la présentation de l'entreprise DigitalSide, ainsi l'environnement du déroulement de notre stage.

Le deuxième Chapitre : est consacré aux définitions propres, des outils et technologie utilisé pour la réalisation de notre application

Le troisième Chapitre : il s'agit d'une prise de connaissance de l'existant, en d'autre terme d'une analyse et spécification des besoins. Suivi de « la Conception » qui présente une étape primordiale, car c'est une phase de modélisation théorique de notre Module Call Center.

Le quatrième et dernier chapitre : nous enchainons avec la présentation de la réalisation du module.

Chapitre I : Contexte du stage

Introduction

Dans ce chapitre, nous allons présenter l'entreprise DigitalSide, et son secteur d'activité. Afin de bien situer le rôle du stage, dans la réalisation de notre mémoire.

I.1. L'entreprise DigitalSide

Un groupe de société de conseils. Il est Composé de « **Iso Système** » Spécialisé en Conseil, et Etude en Management et « **DigitalSide** » Spécialisé en Informatique et Système d'information.

I.2. Présentation de DigitalSide

I.2.1 La forme juridique

Le cabinet est mandaté par le ministre de l'industrie et des mines l'ANDPME (Agence National de Développement de la PME) dans le cadre national de la mise à niveau qui est doté d'une enveloppe de 386 milliard de Dinard.

I.2.2 Vocation

Le cabinet DigitalSide propose divers services :

1. **En Management** dans La réalisation de diagnostic stratégique et plan de mise à niveau, Stratégie d'entreprise et business model, aussi Conseil en organisation et en Gestion des ressources humain. Adapte des Stratégie Marketing et E-Marketing.
2. **En Communication** elle donne des Conseils en stratégie de communication dans les Relation public et en E-Communication, comme le traitement du Web Marketing.
3. **Export** le cabinet possède les moyens d'organiser des salons et foires à l'échelle nationale comme à l'échelle l'internationales.
4. **Technique d'information et de Communication** le service dans lequel nous étions affectés pendant notre stage. C'est un service qui comporte l'Intégration des Solutions ERP/CRM, Et des solutions de GPAO. Comme il fait l'Etude et le Diagnostic des Systèmes D'information, il propose la mise en place d'une structure SI dans les organisations.

I.3. l'organisation de *DigitalSide*

L'équipe de DigitalSide est organisée en deux équipes distinctes :

I.3.1 Equipe Fonctionnel

Cette possède les rôles suivants :

- L'étude du système existant.
- L'analyse des besoins.
- La définition du périmètre et des sous-systèmes, les fonctionnalités, les informations à manipuler.
- L'étude des contraintes de sécurité
- La gestion de la migration des données.
- La distinction les critères d'acceptation et les tests fonctionnels.
- L'estimation des charges et la planification (ressource Humaine et matériel).

Le rôle de cette équipe repose sur la réalisation d'un ouvrage, qui est le ***Cahier de Charges***. Ce dernier doit répondre aux normes (la norme AFNOR), pour définir les besoins métier de manière précise et claire. Ensuite le cahier de charge va être transmis à l'équipe technique.

I.3.2 Equipe Technique :

Cette équipe réalise les études du cahier de charge, pour proposer des solutions techniques. Par la suite représenter ces solutions avec une conception détaillée.

Les rôles de cette équipe consiste à :

- L'établissement de la conception.
- La description des traitements.
- Les contraintes de sécurité.
- La rédaction des tests.
- La rédaction des procédures d'exploitation.
- L'estimation des charges et la planification (coup de temps, et faisabilité).
- La réalisation.

La collaboration entre les deux équipes est très importante, et représente d'un côté la réussite d'un système qui répond à l'attente du client, et d'un autre tout un budget consacré par l'entreprise à la réalisation du projet.

I.4. Contexte Du Stage :

Après avoir présenté l'Entreprise DigitalSide, qui s'est chargé de notre stage, nous allons enchaîner avec à la description du Contexte de notre stage.

Les moyens mis à dispositions : il s'agit des ressources contribuant à la réalisation de notre application :

I.4.1 Les moyens matériels

Pour un début en termes de matériel, on s'est contenté de nos propres moyens (des microordinateurs portables). Le développement du module s'est fait sur le serveur local de nos ordinateurs respectifs.

La récupération des informations qui représentent la base de données de l'entreprise Karoura, s'est faite avec une migration des données dans le serveur Cloud, dont dispose l'entreprise DigitalSide dans l'espace réservé à l'entreprise Karoura. Cela était largement suffisant pour arriver à faire nos tests pour la réalisation.

Les locaux de l'entreprise, qui se trouve dans le siège social de l'entreprise à El Achour à Alger, ont été mis à notre disposition. Un moyen qui nous a permis d'organiser des réunions, pour se retrouver en binôme, et avancer dans notre travail. Vu qu'on était loin de notre ville de résidence.

I.4.2 Les moyens Logiciels

Les moyens logiciels, ont consisté à mettre à notre disposition des comptes sur la plateforme de l'entreprise, pour pouvoir communiquer avec les personnes chargées de notre encadrement pendant le stage. C'était aussi le moyen sur lequel on nous donner des taches à faire avec un délai. Ainsi un compte rendu sur l'état d'avancement de chaque tache sur le travail à rendre.

Pour accéder à la plateforme, il faut avoir un compte avec mot de passe et login. Comme le montre la figure – I.1-

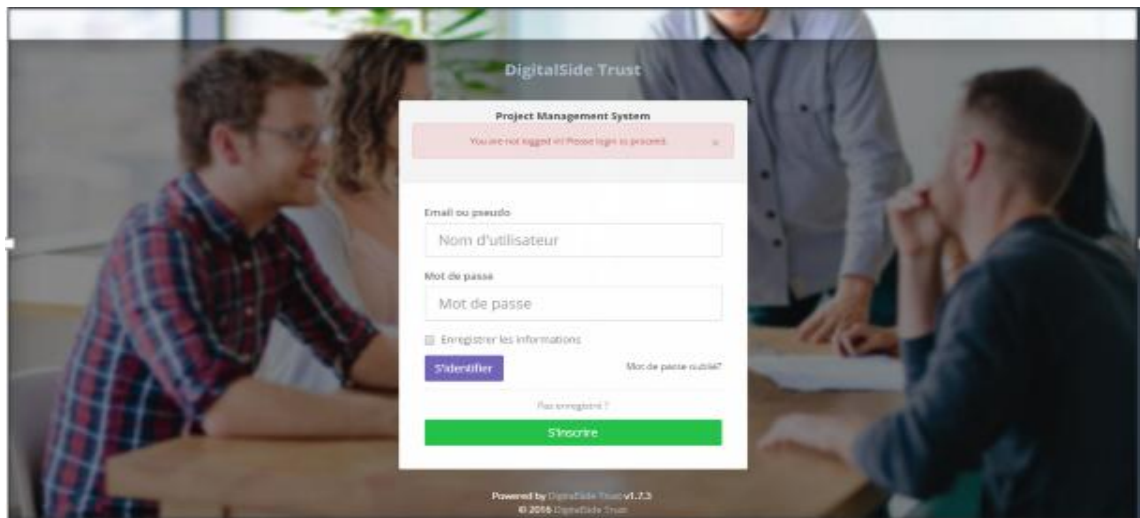


Figure -I.1- Support d'authentification de la plateforme

Une fois connecté sur la plateforme, on pouvait bénéficier d'un forum, pour pouvoir discuter avec les membres concernés par notre PFE. Cela est illustré dans la Figure- I.2-

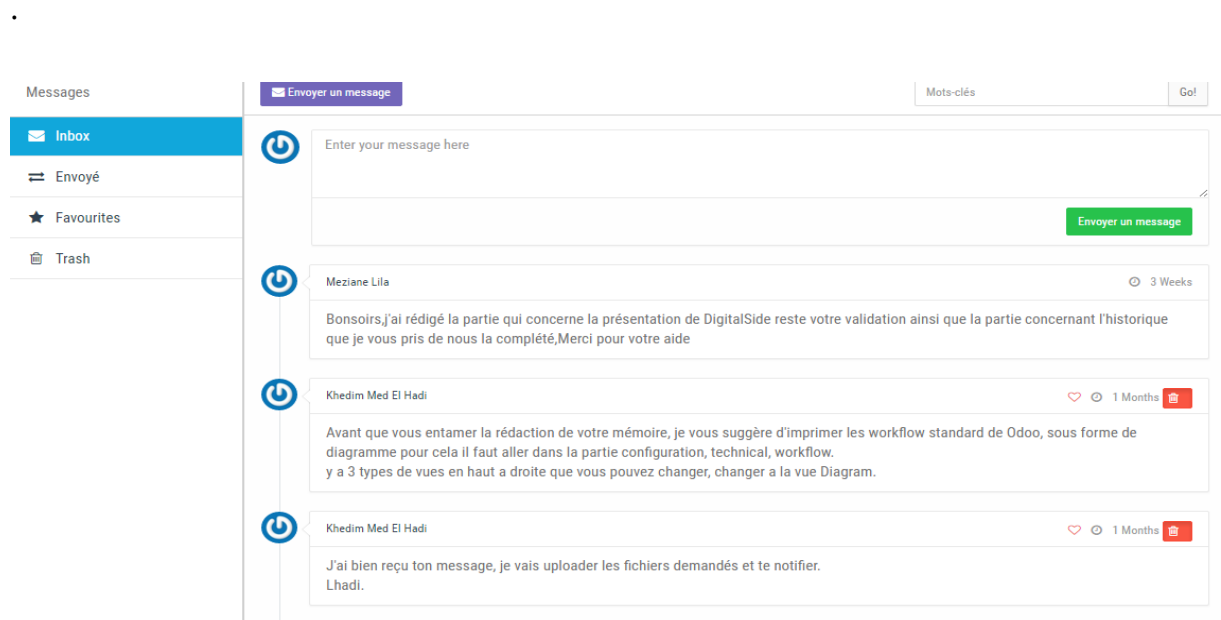


Figure -I.2- Forum de discussion de la plateforme

Pour la réalisation des tâches à effectuer, on reçoit un message de notification dans nos emails. Ensuite le sujet est expliqué et détaillé sur la plateforme, avec une date

d'échéance, et des figures de statistique d'état d'avancement (évalué avec un pourcentage). Comme le présente la Figure -I.3- .

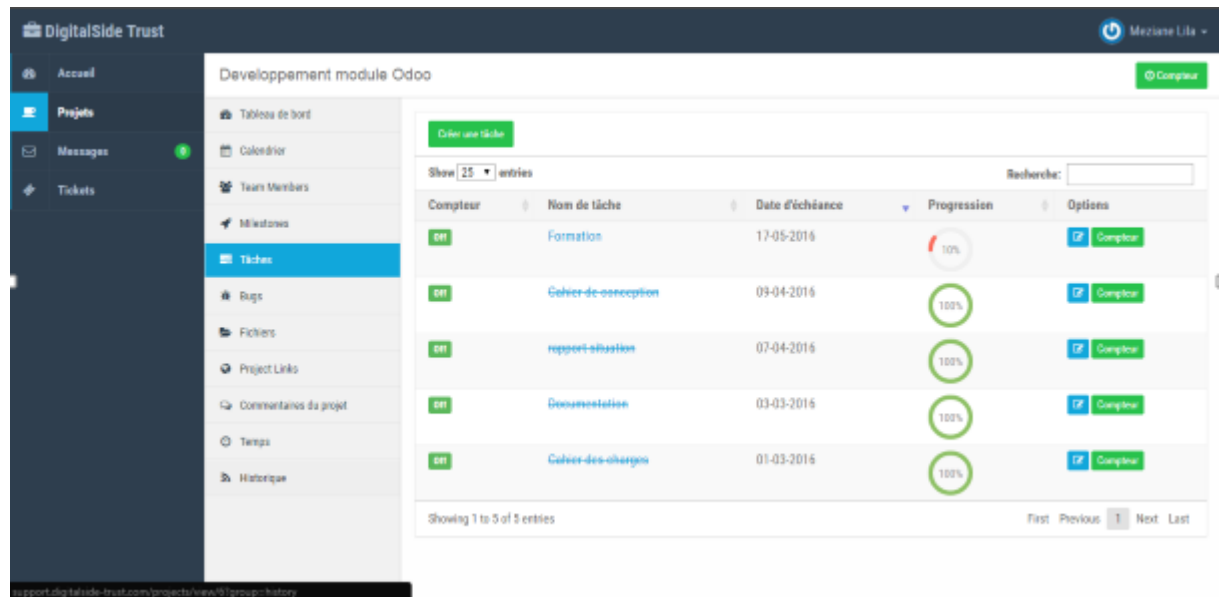


Figure -I.3- l'état d'avancement des tâches

La plateforme nous permet aussi de faire des chargements ou téléchargement de fichier, qui est accessible à tous les membres de la plateforme. Comme On peut le visionner dans la figure -I.4-

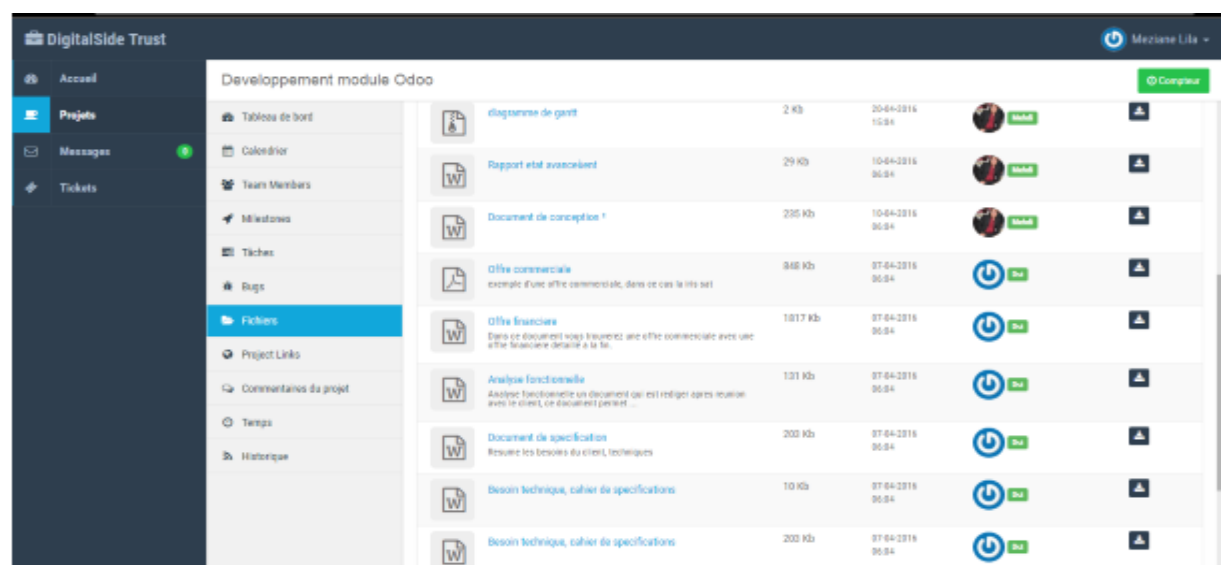


Figure -I.4- Espace fichiers

I.5. Objectifs du Projet

I.5.1 Problématique

Tout d'abord un Call center ou en d'autres termes un centre d'appels, fait partie du CRM c'est-à-dire le « *Customer Relationship Management* », qui constitue l'interface entre un client et les représentants commerciaux, juridiques, ou administratifs d'une entreprise. La gestion du call center peut intégrer la gestion des commandes, la gestion de la comptabilité, ou des plaintes.

L'entreprise Karoura possède un service call center, hébergé dans un serveur externe. Mais ce dernier n'avait pas de relations avec l'ERP Odoo, ce qui posait un problème dans la mesure où une double saisie des données s'effectuait. C'est-à-dire que les services développés avec l'ERP intégrés sur Odoo partagent la seule et même base de données, contrairement au ancien call center, qui avait sa propre base de données. Du coup les problèmes rencontrés avec l'ancien existant, on fait appel au service de l'ERP.

I.5.2 Finalité

La finalité de ce projet, est que l'entreprise Karoura puisse gérer son centre d'appels, à travers un module de call center sur la plateforme Odoo, et enfin l'intégrer avec le reste de ses services.

Conclusion

Nous avons présenté dans ce chapitre, le secteur d'activité, ainsi que l'organisation de l'Entreprise DigitalSide. Une étape dans la quelle par la suite, on saurait exposer le déroulement de ce mémoire, avec la contribution du stage au sein de l'Entreprise.

Chapitre II : Présentation des Outils de Développement

Introduction

Dans ce chapitre nous verrons tout d'abord une présentation générale des ERP et Odoo en particulier, on décrira son évolution, son architecture et les modules qui le composent. Par la suite nous allons donner une description du Langage Python qui gère la communication entre le serveur et la base de données PostgreSQL, puis XML qui gère la communication entre le client et le serveur Odoo. Pour finir nous allons décrire les processus UP qui ont piloté notre travail tout au long de la réalisation du projet.

II. OpenERP Odoo

Odoo, anciennement *OpenERP* et *TinyERP*, est initialement un progiciel open-source de gestion intégré comprenant de très nombreux modules permettant de simplifier la gestion d'entreprise dans son ensemble. Le logiciel est utilisé par plus de deux millions d'utilisateurs pour gérer leurs entreprises à travers le monde. [1]

Odoo touche essentiellement la gestion commercial (points de vente, Stocks, E-Commerce....), aussi bien que le reporting (statique et dynamique). [2]

Odoo donne une image cohérente des données partagées grâce à l'utilisation d'une base de données unique.

Odoo avec l'ancienne appellation OpenERP, grâce à une communauté très développée et étendue à travers le monde a subi de récentes améliorations à différents niveaux qui sont les suivantes :

II.1. Modules d'Odoo

L'aspect libre du logiciel a permis le développement de nombreux modules tiers créés par sa communauté de développeurs. Ces applications sont pour certaines officiellement validées par l'éditeur tandis que d'autres ne sont destinées qu'à des versions spécifiques. [1]

Le progiciel compte 260 modules officiels et certifiés c'est-à-dire que la certification d'un module est obtenue après l'examen du code source, la vérification du suivi des bonnes pratiques par l'équipe Odoo [2].

Il existe aussi des modules qui sont développés par la communauté de développeurs et qui s'élève à environ 4000 modules communautaires.

Jusqu'à l'heure actuelle plus de 20 modules sont développés par mois, plus de 100 demandes d'installations de Odoo par des entreprises à travers le monde.

Tous ces modules sont divisés en deux aspects distincts :

II.1.1 Modules Front Office

Les modules qui ont un aspect Front Office représentent l'interface de l'entreprise vis-à-vis de son environnement et de ses clients, on peut citer quelques exemples :

- Le module de création de sites web avec un système de gestion de contenu intégré.
- Le module de vente en ligne pour permettre les achats via le site est assurer un service e-commerce.
- Le module contenant l'interface de point de vente (PDV). [1]

II.1.2 Modules back-Office

Les modules qui possèdent un aspect Back Office c'est les modules qui ont un rapport direct avec la gestion internes et les ressources de l'entreprise on peut citer les exemples suivants :

- Le module de gestion de relation clients (GRC et CRM (**en**)). avec une étude des opportunités et des statistiques sur les clients.
- Le module de gestion des ventes représente tout ce qui est en rapport avec la facturation et ventes de l'entreprise.
- Le module de gestion de production c'est-à-dire qu'il intègre les fonctionnalités liées à la production dans une usine les quantités de matière premières nécessaires.
- Le module de gestion de projets intègre les diagrammes de Gantt et PERT pour l'affectation, le suivi de la réalisation des tâches.
- Le module de gestion des stocks possède des fonctionnalités liées au suivi des quantités et les variantes des produits et leurs de date de péremption si elle existe.
- Le module de gestion des ressources humaines gère le système de pointage des employés, les congés, recrutement de nouvelles personnes.

- Le module de gestion des achats caractérise les achats de l'entreprise en termes de produits et de fournitures, ce module gère aussi la relation de l'entreprise avec ses fournisseurs.
- Le module de gestion de la comptabilité intègre les fonctionnalités liées aux paiements des factures, la gestion des chèques entrants/sortants, et les documents liés aux impôts. [1]

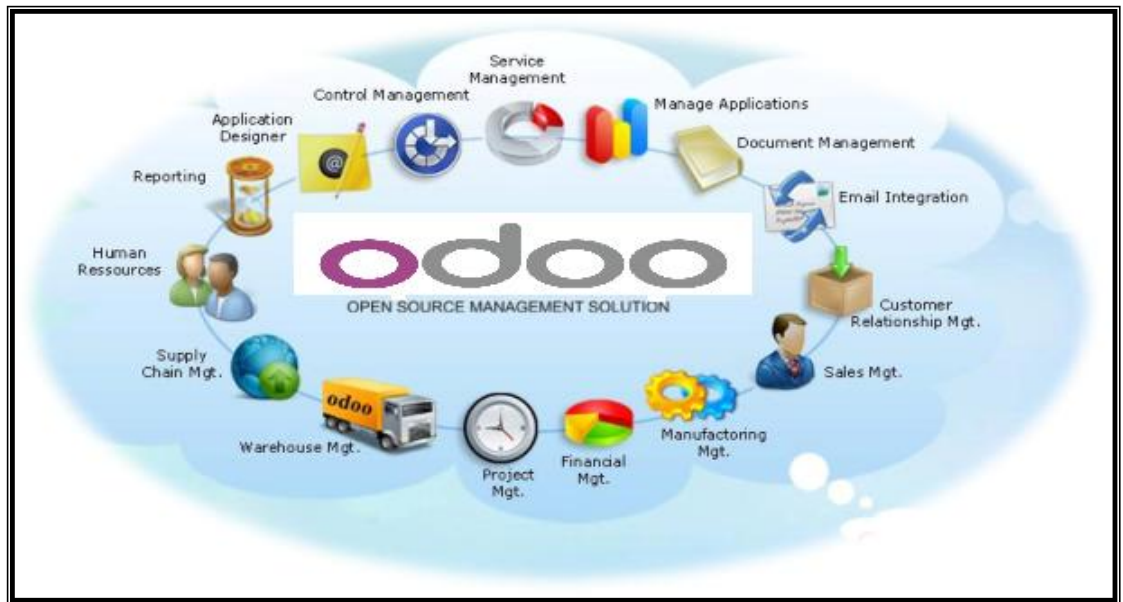


Figure -II.1- Schéma représentant les modules disponibles dans Odoo

La figure -II.1- illustre les différents modules et fonctionnalités de Odoo qui lui sont intégrés.

II.2. Architecture Logiciel

La conception d'Odoo est orientée par une architecture MVC, des flux de travail flexibles, une interface-utilisateur graphique dynamique, une interface de communication interne XML-RPC, et un système personnalisable de comptes rendus.

D'un point de vue de l'architecture technique, Odoo est construit autour de trois composants principaux :

- le serveur *Odoo-server* qui stocke ses données dans une base PostgreSQL
- le client *Odoo mobile* (anciennement *Odoo-client* abandonné depuis la v7) qui s'installe sur le terminal de l'utilisateur
- le serveur web *Odoo-web* qui permet une utilisation depuis un navigateur [3]

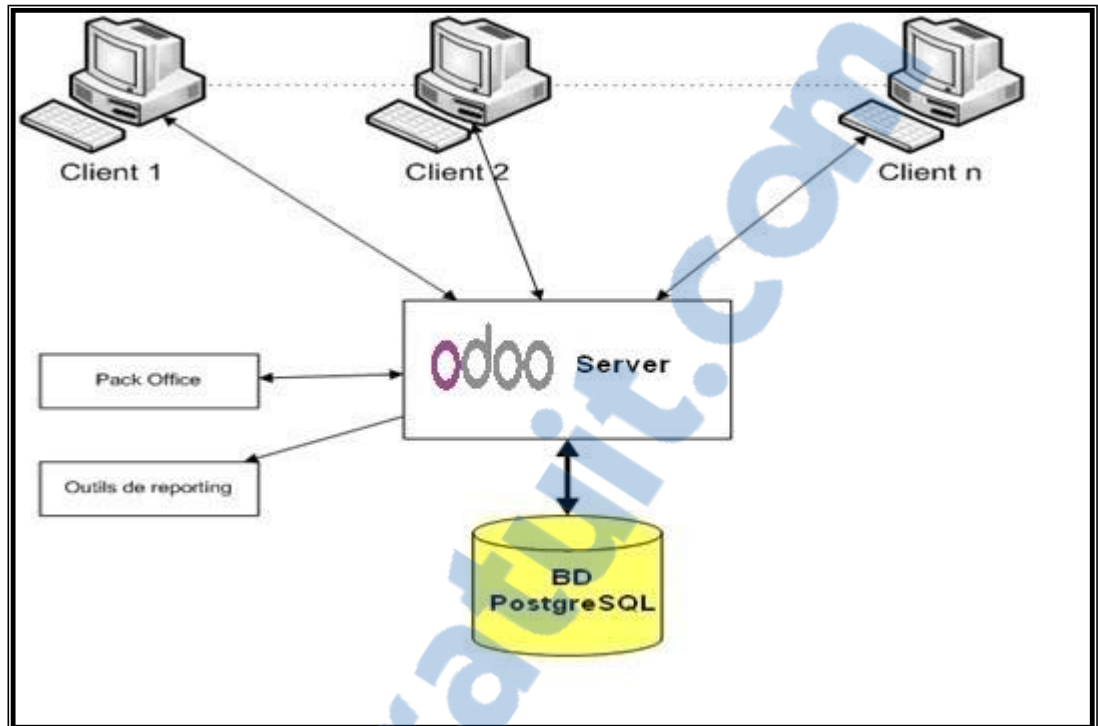


Figure -II.2- Architecture 3 tiers Odoo

La figure -II.2- représente l'architecture 3 tiers de Odoo, on voit que les postes clients communiquent avec le serveur de Odoo en utilisant le langage XML, le serveur d'application Odoo les outils de reporting et le pack Office, le serveur communique aussi avec le serveur de base de données PostgreSQL en utilisant le langage Python.

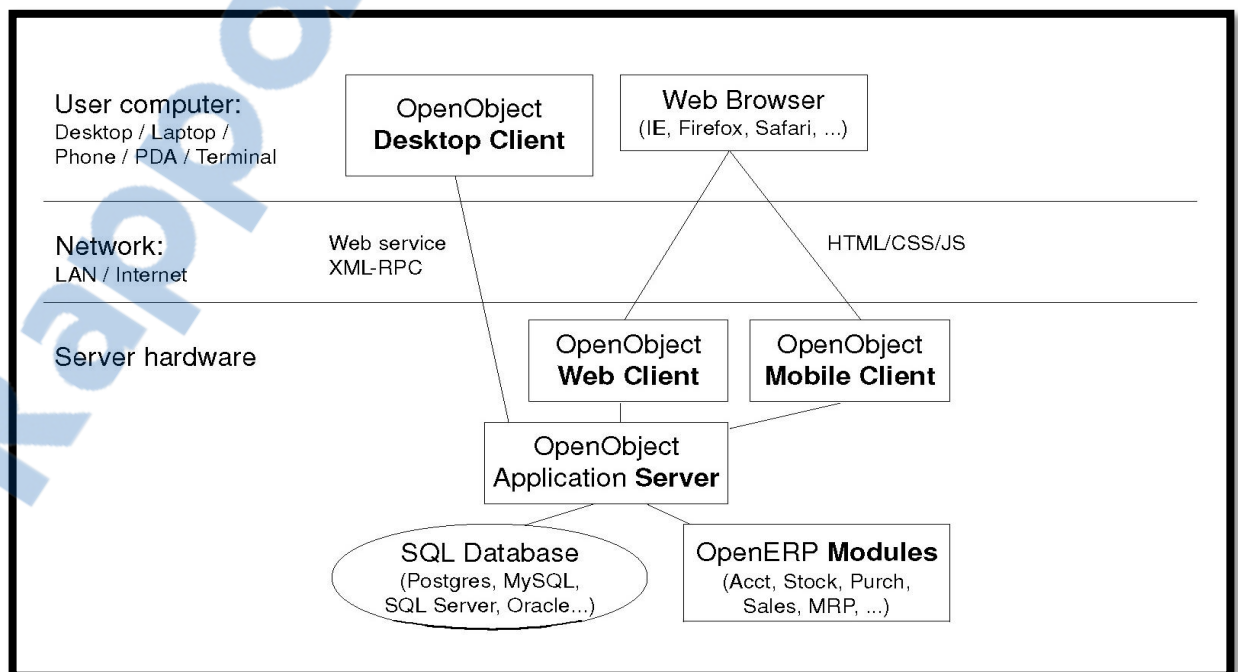


Figure -II.3- Architecture logiciel Odoo

Cette figure explique de manière plus détaillée l'architecture de Odoo avec la décomposition en 3 nouveaux niveaux qui sont le client, le réseau et le serveur.

Dans un premier temps l'application client communique directement avec le serveur d'application grâce à la technologie XML-RPC et l'affichage se fait en HTML/CSS/JavaScript.

En revanche le navigateur Web passe par le web client ou le mobile client ce qui dépend du type de connexion pour passer ensuite au serveur d'application.

Et enfin le serveur d'application communique avec la base de données et les modules disponibles dans Odoo [3].

II.3. Fonctionnalités ajoutées

➤ Interface Utilisateur plus rapide

Avec l'avènement des nouvelles technologies du Web et l'amélioration des langages HTML/CSS cela a permis d'alléger les pages web qui s'exécutent sur le navigateur pour un meilleur confort de l'utilisateur et un temps de réponse amélioré dans le chargement des interfaces.

➤ Recherche améliorée

La recherche dans Odoo a été améliorée grâce à l'affichage des résultats en temps réel, ainsi que l'ajout de nouveaux filtres pour augmenter la précision des résultats obtenus.

➤ Object Relational Mapping

Un élément clé de Odoo est la couche ORM. Cette couche évite d'avoir à écrire des requêtes SQL à la main et fournit l'extensibilité et des services de sécurité.

➤ Integrated Workflow Engine

Un autre élément clé a été ajouté c'est le moteur de gestion de flux de travail ou le workflow engine, qui est un artefact pour gérer un ensemble de «choses à faire» associés

aux enregistrements d'un modèle. Le flux de travail fournit un niveau supérieur moyen d'organiser des tâches à effectuer avec ou sur un enregistrement.[4]

Plus précisément, un flux de travail est un graphe orienté où les nœuds sont appelés «activités» et les arcs sont appelés «transitions».

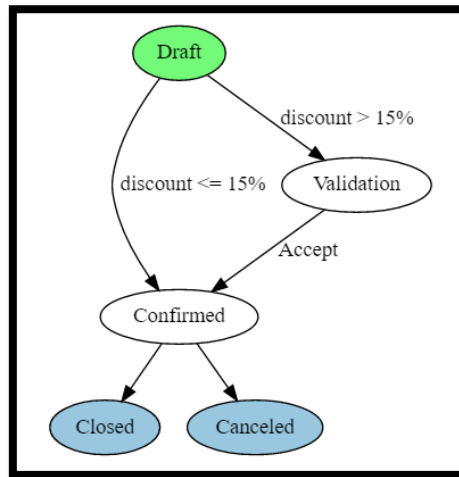


Figure -II.4- Exemple de flux de travail

La figure -II.4- apporte une clarté par rapport graphe orienté lié au flux de travail décrivant le cycle de vie d'une facture d'achat, Elle démarre de «brouillon» et l'état final est soit fermé après confirmation ou dans un état « annulé »

II.4. Versions Odoo

Le tableau ci-dessous représente l'historique de l'évolution de Odoo à travers les années avec les dates liées aux changements de nom et de versions.

Nom du logiciel	Version	Date de lancement
Tiny ERP	1.0	
	2.0	
	3.0	
	4.0	
OpenERP	5.0	
	6.0	Octobre 2009
	6.1	
	7.0	Décembre 2012
Odoo	8.0	Septembre 2014
	9.0	

■ Anciennes versions ou fin de maintenance

■ Anciennes versions avec maintenance étendue

■ Version actuelle

■ Version lancée dernièrement

Figure -II.5- Historique des versions de Odoo

OpenERP a démarré en 2002 en Belgique sous le nom de *TinyERP*. Sa première version a été lancée en 2004, en 2007 une branche indienne spécialisée en développement a été ouverte ce qui permet d'embaucher 80 développeurs à plein temps pour améliorer et développer le *OpenERP*.

Un changement de nom est intervenu en 2010 de *TinyERP* vers *OpenERP* avec noyau, l'arrivée de nouveaux investisseurs l'entreprise a reçu un coup de booste ce qui a poussé vers un changement du nom de *OpenERP* vers *Odoo* en mai 2014 pour un point de vue purement marketing le lancement de la version 8.0 en septembre 2014 avec laquelle on a développé notre module et récemment la dernière version 9.0 a été lancée [3].

Pour la réalisation de notre module sous *Odoo*, on a sélectionné la version 8.0, car c'est la version la plus stable et avancée actuellement, mais aussi elle intègre plusieurs fonctionnalités qui sont devenues payantes dans la version 9.0 due à un changement de stratégie de la part de l'entreprise *OpenERP* pour rendre *Odoo* encore plus commerciale et faire évoluer les bénéfices générés.

II.5. Licence AGPL

Il s'agit d'une nouvelle licence. Basée sur la version 3 de la licence GPL, elle contient une clause supplémentaire qui permet aux utilisateurs qui interagissent avec le logiciel

licencié au travers d'un réseau de recevoir le code source correspondant. Comme l'indique le communiqué, « en publiant cette licence, la Free Software Fondation souhaite encourager les communautés de développeurs et d'utilisateurs qui travaillent sur des logiciels libres orientés vers le réseau ». [5]

En résumé, Les utilisateurs finaux pourront utiliser des modules propriétaires standards publiés par les grandes entreprises qui sont essentiels pour leur entreprise.

II.6. L'environnement PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD. [6]

Les Principales caractéristiques de PostgreSQL sont les suivantes :

- PostgreSQL peut stocker plus de types de données que les types simples traditionnels entiers, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type, etc. [7]
- PostgreSQL fonctionne sur diverses plates-formes matérielles et sous différents systèmes d'exploitation.
- PostgreSQL est largement reconnu pour son comportement stable, proche de *Oracle*. Mais aussi pour ses possibilités de programmation étendues, directement dans le moteur de la base de données, via PL/pgSQL. [6]

Comme cités auparavant le SGBD PostgreSQL présentent beaucoup d'avantages liés à la gestion des accès et leurs nombre.

II.7. Langage de programmation Python

Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions. [8]

Les Principales Caractéristiques du langage Python sont les suivantes :

- Le langage Python est placé sous une licence libre proche de la licence BSD
- Python fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux de Windows, Unix, Linux, Mac OS, Android, iso.
- Python est apprécié par les pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation. [9]

Les responsables de l'entreprise OpenERP ont choisi Python comme technologie pour garantir une communication fluide entre le serveur Odoo et le système de gestion de base de données PostgreSQL.

II.8. Langage structurel XML

Le langage XML (eXtended Markup Language) dérive de SGML (Standard Generalized Markup Language) et de HTML (HyperText Markup Language). Comme ces derniers, il s'agit d'un langage orienté texte et formé de balises qui permettent d'organiser les données de manière structurée. [10]

Les Principales Caractéristiques du langage XML sont les suivantes :

- Séparation stricte entre contenu et présentation.
- Simplicité, universalité et extensibilité.
- Format texte avec gestion des caractères spéciaux. .
- Structuration forte.
- Modèles de documents (DTD et Schémas XML).
- Format libre. [10]

Les responsables de L'entreprise OpenERP ont choisi d'utilisés cette technologie pour assurer la communication entre l'ordinateur client et le serveur Odoo due aux avantages cités auparavant.

II.9. VOIP

La **voix sur IP**, ou « **VOIP** » pour *Voice over IP*, est une technique qui permet de communiquer par la voix (ou *via* des flux multimédia : audio ou vidéo) sur des réseaux

compatibles IP, qu'il s'agisse de réseaux privés ou d'Internet, filaire (câble/ADSL/fibre optique) ou non (Wi-Fi, GSM,...).[11]

Le **VOIP** est utilisé pour par exemple pour appeler d'un PC vers une ligne de téléphone fixe ou d'un PC vers un autre PC gratuitement.

La technologie du VOIP est une partie intégrante de notre projet, car elle garantit aux utilisateurs du module de Call center d'effectuer et recevoir des appels à travers Odoo.

II.10. Les designs patterns

II.10.1 Description des design patterns

Les designs patterns sont utilisés pour améliorer l'architecture du logiciel. Ce sont des modèles de conception réutilisables qui répondent à des problématiques courantes de conception indépendamment de tout langage. Ces modèles de conception fournissent un support fort pour la mise en œuvre de principes chers à l'approche par objets : la flexibilité, la réutilisabilité, la modularité, la maintenabilité

Les designs patterns décrivent une solution simple et élégante à des problèmes spécifiques en conception orienté objet. . Les designs patterns utilise un niveau d'abstraction élevé qui

Permet d'élaborer des constructions logicielles de meilleure qualité.

Les designs patterns sont caractérisés par : Nom, Problème, conséquences

Il existe plusieurs designs patterns, ils se décomposent en 3 groupes :

- Les patterns de création : reliés au processus de création d'objets.
- Les patterns de structure : définis la composition des classes et des objets.
- Les patterns de comportement : caractérise les interactions entre classes/objets, ainsi que la distribution des responsabilités.

Dans chaque groupes, il y des patterns qui traitent les relations entre objets et d'autres qui traitent ceux entre classes. [12]

II.10.2 MVC Pattern

Le design pattern Model View Control est patron de conception utilisé pour séparer le logique métier de l'interface utilisateur, c'est-à-dire que le model représente la logique métier, le View ou la vue représente l'interface et le contrôleur gère le comportement de la vue et du modelé

Odoo utilise le design Pattern MVC (Model View Control) pour construire l'architecture des modules qui le composent. [13]

II.11. Processus UP

II.11.1 Définition

Le Processus Unifié (PU ou UP en anglais pour **Unified Processus**) est un processus de développement logiciel, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. [14]

II.11.2 Caractéristiques

Les principales caractéristiques du processus unifié sont :

- Itérations et incrémentations.

L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes.

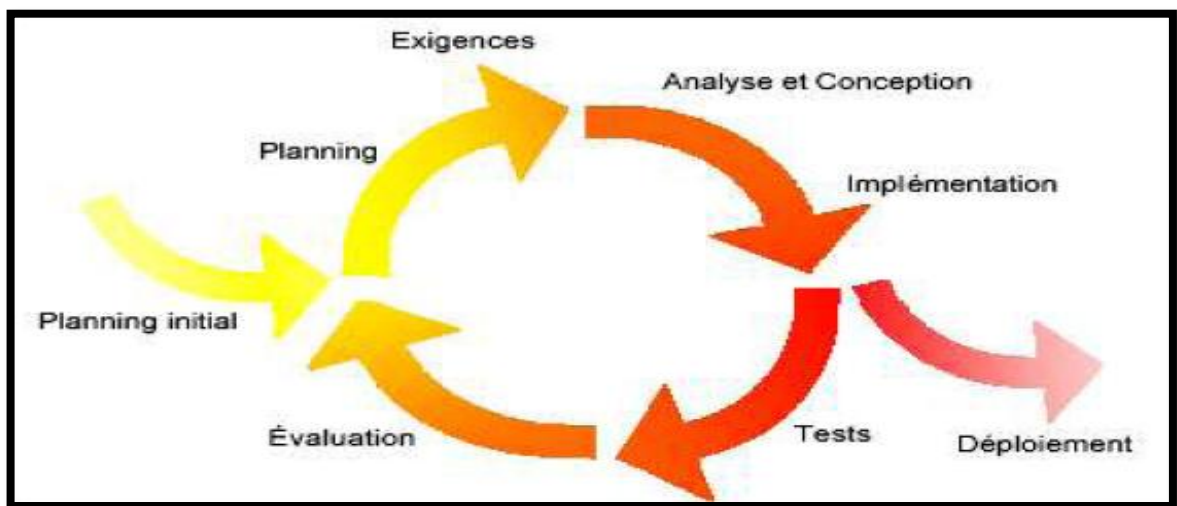


Figure -II.6- Schéma d'itérations du processus UP

Ce schéma illustre les étapes de réalisation d'un projet informatique du planning initial jusqu'au déploiement. De manière itérative. Ces itérations dépendent modifications ou problèmes survenues au cours du cycle de vie du projet.

- **Centré sur l'architecture.**

Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Le processus unifié préconise le modèle des 4+1 vues, de **Philippe Kruchten**, pour représenter l'architecture : [14]

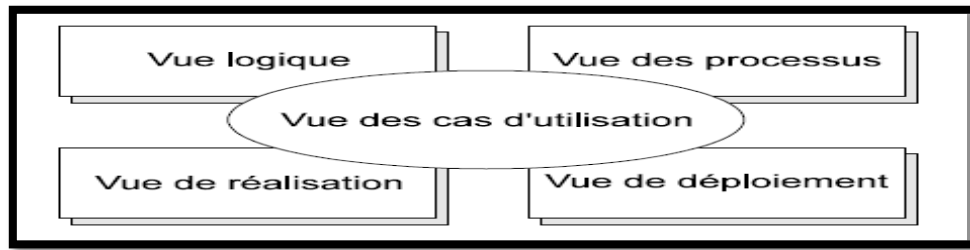


Figure -II.7- Le modele 4+1 vues

Cette figure représente Le modèle « 4+1 » c'est-à-dire qu'il faut étudier un projet des 4 principales vues, puis intégrer à la fin la vue par rapport au cas d'utilisation.

Cette étude garantit plus de clarté par rapport au projet et anticipe les éventuels problèmes qui vont survenir au cours de la réalisation de ce dernier.

➤ **Piloté par les cas d'utilisation UML.**

Le but principal d'un système informatique est de satisfaire les besoins du client.

Le processus de développement sera donc axer sur l'utilisateur [14].

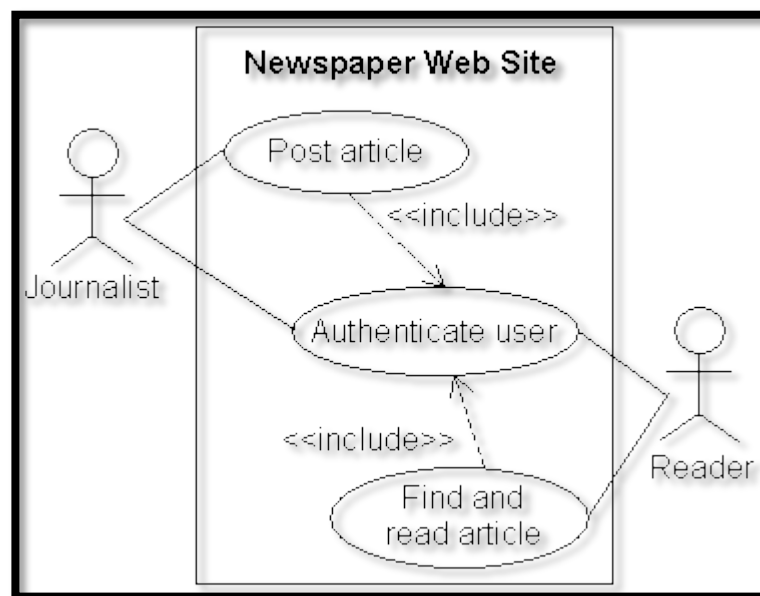


Figure –II.8- Exemple d'un diagramme de cas d'utilisation

Ce diagramme se comporte d'acteurs, des cas d'utilisation et un le périmètre du système. Les acteurs sont représentés par le « Journalist » et le « Reader ». Les cas d'utilisation du système sont « Post article », « Read article » et « Authentification ». Le périmètre du système est défini par le rectangle.

II.11.3 Vie du processus unifié

Le processus UP gère le processus de développement par deux axes :

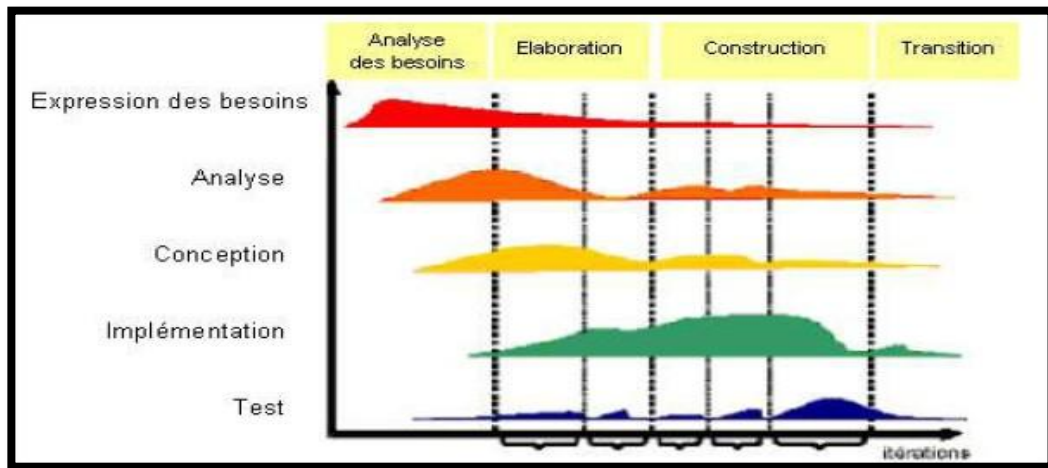


Figure -II.9- Cycle de vie du processus UP

Cette figure illustre la hiérarchie des processus au cours d'un projet, leurs degrés de réalisation et leurs importances à un instant « T ». Si on prend l'exemple du processus « expression du besoins » on remarque qu'elle est présente au début du cycle de vie du projet et son importance est primordiale, mais diminue au fur et à mesure de l'état d'avancement dans le cycle de vie du projet [14].

L'axe vertical représente les principaux enchaînements d'activités, qui les regroupent selon leurs natures.

L'axe horizontal représente le temps et montre le déroulement du cycle de vie du processus ; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons.

Pour conclure cette section, on peut dire que les démarches du processus UP nous ont aidées à mieux cerner le périmètre du projet ainsi de mener à bien la réalisation de notre projet.

Conclusion

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. Parmi les processus cités précédemment, nous avons choisi le processus UP qu'on utilisera pour modéliser notre système. Cette modélisation sera détaillée ainsi que les étapes de recueils de besoins et la rédaction du cahier des charges dans le chapitre suivant.

Chapitre **III** : Méthodologies

Introduction

Dans ce chapitre, nous allons détailler le travail qu'on a pu effectuer depuis l'analyse des besoins du client, jusqu'à la rédaction du cahier de conception.

III. Analyse des besoins

Le recueil des besoins, que nous avons effectué en collaboration avec les représentants de l'entreprise Karoura, est passé par les étapes suivantes :

- Rédaction d'un questionnaire.
- Recueil de besoins grâce à l'organisation de réunions de travail avec le chef de projet. ainsi que les représentants de l'entreprise Karoura.
- Analyse de la solution informatique déjà existante.

Les étapes citées ci-dessus, seront détaillées dans les sections suivantes.

III.1. Rédaction du Questionnaire

Dans le but de la rédaction du questionnaire, nous avons effectués des recherches approfondies portant sur les aspects suivants :

- Le premier, traite le Call center, c'est-à-dire nous avons cherché à identifier les fonctionnalités existantes, ainsi que les profils utilisateurs.
- Le second, consisté à consulter des questionnaires existants, pour s'en inspirer.

Ces démarches, nous ont permis d'avoir une idée plus claire sur les points à aborder pendant la réunion avec le client.

Ce qui nous a conduits à rassembler un ensemble de questions traitant les aspects organisationnel, et fonctionnel d'une entreprise.

Ces questions nous ont permis de mieux connaître les processus métiers, les profils utilisateurs, ainsi que les informations et fonctionnalités à afficher.

La première version du questionnaire a été refusée par les représentants de DigitalSide, pour inadéquation et incomplétude des questions proposées.

Nous avons apporté certaines modifications au questionnaire. Le questionnaire suivant a été validé par le chef de projet de l'entreprise DigitalSide.

Q1- Pouvez-vous nous donner les Détails de l'organisation des agents operateurs pour le Call center ?

Q2- Pouvez-vous nous donner le Nombre d'agents operateurs ?

Q3- Est-ce qu'il y aura une restriction avec des Droits d'accès à un groupe de contacts pour les agents operateurs ?

Q4- Voulez-vous avoir un Accès rapide à la création d'un nouveau client ?

Q5- Pouvez-vous nous donner les Types de contacts que vous voulez qu'ils soient présents dans de l'annuaire ?

- Exemples : (Fournisseurs – Clients –Services des impôts ...).

Q6- Pouvez-vous nous donner les Types de clients ?

- Par catégorie Exemple (Gold Silver Bronze).

Q7- Voulez-vous Afficher la fiche client ?

- Si oui Quels sont les informations que vous voulez affichés ?

Exemples :

- Nom de l'entreprise.
- Nom de l'opérateur.
- Numéros de téléphone.
- Activités commerciales récentes.
- Créances du client.
- Top produit acheté.
- Un champ commentaire pour avoir des informations en plus sur le client.
- Aperçu de l'historique de conversation (Appels – Mails).

Q8- Quels seront Les type de medias utilisé pour le centre de contacts ?

- Mail.
- Appels entrants.
- Appels sortants.
- Messagerie instantanée.

Q9- Voulez-vous Afficher un dialer pour former un numéro de téléphone à appeler ?

- Si oui à quel endroit de l'écran ?

Q10- Faire un accès rapide aux clients les plus actifs ?

- Si oui combien de contacts ?

Q11- Les listes des clients seront basées sur quels critères ?

- Top de la semaine.
- Top du mois.
- Top de l'année.
- Relation particulière avec le client.
- Nouveaux arrivage de produits.
- Catégories des clients.
- Régions.

Q12- Voulez-vous l'Affichage de la liste des clients ?

- Si oui Triés suivant quel ordre : date d'ajout ancienneté, chiffre d'affaire ?

Q13- Voulez-vous un accès rapide à la liste des nouveaux produits arrivés ?

- Si oui à quel endroit de l'écran ?

Q14- Les Notifications d'appels des clients seront à quels endroits de l'écran et suivant quelle sonnerie ?

Q15- Est-ce qu'il y a une priorité d'affectation d'appels par rapport à un agent ?

- Si oui laquelle ?

Q16- Voulez-vous un Rappel d'appels aux clients ?

- Si oui est ce que vous voulez un message accompagnant le rappel ?

Q17- Voulez-vous La mise en place de l'enregistrement des appels ?

- Si oui est-ce que pour tous les contacts ou pour un groupe de contacts ?

Q18- Voulez-vous un Affichage de l'historique des appels et leurs durées ?

Q19- Voulez-vous un Archivage des appels et les mails échangés ?

- Si oui à quelle fréquence ?

Q20- Est-ce qu'il y a une réponse automatique aux mails ?

Q21- Est-ce qu'il y aura un envoi automatique de mail ?

- Si oui dans quels cas ?

Q22- Voulez-vous mettre en place d'une messagerie vocale ?

Q23- Quelles sont les fonctions du service vocal que vous voulez ajoutés ?

Q24- Quelles informations seront diffusées pendant l'attente ?

- Musique.
- Message préenregistré.
- Message publicitaire.

Q25- Voulez-vous mettre en place un service pour la redirection de l'appel vers l'agent concerné ?

Q26- Dans quelles langues seront enregistrés les services de messagerie ?

Q27- Quel est le délai d'attente maximum accepté ?

Q28- Quels seront les périodes où les services vocaux seront actifs ?

- Délai d'attente.
- Automatiquement.
- En dehors d'horaires de travail.

Q29- Voulez-vous une distribution intelligente en fonction des postes disponibles (absent hors ligne ?

Q30- Voulez-vous une analyse statistique des appels ?

- Si oui suivant quels critères ?
 - Nombre d'appels Par Client.
 - Nombre d'appels par operateurs.
 - Top des clients par produits / par période.

III.2. Recueil des besoins

L'organisation de réunions de travail, par le chef de projet, a permis de créer un contact avec les représentants de l'entreprise Karoura, dans le but de créer un environnement de travail convivial.

Au cours de ces réunions, nous avons utilisé le questionnaire préalablement construit, pour posés des questions au fur et à mesure de la réunion. A la fin nous avons collecté l'intégralité des réponses.

Ces réunions nous ont permis aussi de soumettre des suggestions, liés à l'ajout de fonctionnalités au module du call center.

III.3. Analyse de l'existant

Les réunions organisées dans les locaux de la société Karoura nous ont permis de :

- Comprendre le processus de travail de la société Karoura.
- Découvrir la hiérarchie des employés.

- Analyser l'application web de gestion du call center existante. Cette étape nous a permis de comprendre les fonctionnalités disponibles, pour pouvoir les intégrer à notre module de call center.

III.4. Rédaction du cahier des charges

Dans cette section nous allons présenter le cahier des charges qui a été réalisé, et cela grâce à notre étude des besoins. A la fin de la réalisation des deux étapes précédentes, nous avons rédigé le cahier des charges suivant :

Ce cahier des charges a été validé par le chef de projet ainsi que le client.

Dans la prochaine section nous allons expliquer les étapes de la conception du module de

1- Rôle du cahier de charges

Afin de mieux vous servir et de vous fournir un service de qualité, le cahier des charges va nous permettre de définir au mieux les objectifs et l'architecture du module de Call Center au sein de Odoo.

2- Présentation du module de Call center

Dans cette section on va présenter le tableau de bord avec son interface et les fonctionnalités, par la suite la gouvernance d'accès c'est-à-dire le droit d'accès aux fonctionnalités par rapport aux utilisateurs.

2.1-Tableau de bord

2.1. A- Interface

L'interface du call center va se présenter dans le navigateur, sous forme de page web comme tous autres modules, elle contient :

- Des cases de saisie (exemple : nom de clients).
- Bouton (exemple : valider).
- L'affichage des formulaires (exemple : facture).
- Rapport de sortie sous forme de diagrammes (exemple : statistique).
- Zone de recherche et de filtre (exemple : information d'un client en ligne).
- Champ de sélection (exemple : particulier, superette, Entreprise).
- Image (exemple : logo, photo du client).

call center.

- Un clavier de composition d'appels.

2.1. B- Fonctionnalités

Dans cette section on va présenter les fonctionnalités disponibles sur le module de call center, en commençant par l'ajout de clients, ainsi que la gestion des coupons.

Par la suite on va présenter la gestion des plaintes et enfin la gestion des stocks

Les points abordés ci-dessous toucheront les informations à afficher, les fonctionnalités principales du tableau de bord.

- L'interface du module donne la possibilité dans un onglet, de créer une nouvelle fiche client suivant les caractéristiques du client, qui est déjà intégrée dans la solution Odoo.
- Le module offre un annuaire de contacts qui regroupe les clients, fournisseurs, employés ainsi que les partenaires.
- L'aperçu de la fiche client contient les informations suivantes :
 - Nom du client.
 - Nom opérateur.
 - Solde.
 - Achats récents.
 - Paiement de factures.
- Il existe les types de clients suivants :
 - Actif.
 - Dormant.
 - Suspendu.
 - Dépassant le délai de paiement.
- Les informations à entrer pour les fournisseurs sont :
 - Nom.
 - Adresse.
 - Points de repères.
 - Numéro de téléphone 1.
 - Numéro de téléphone 2.
 - Numéro de téléphone 3.
 - Email 1.
 - Email 2.

- Commentaires.

- Le module de Call center dispose d'un dialer, pour pouvoir composer des numéros et effectuer des appels.
- L'opérateur peut appeler et recevoir des appels à travers le module, grâce à la téléphonie IP qui utilise la technologie VOIP.
- Le module permet d'enregistrer les appels, et d'afficher l'historique avec le nom de l'opérateur, la date et la durée de l'appel.

2.1. C- Gestions des coupons

Les coupons sont des tickets qui sont offerts aux clients de la société Karoura et qui permettent à ces clients d'obtenir des réductions sur le montant des factures d'achats

Dans cette section on va développer les fonctionnalités liées à la gestion des coupons c'est-à-dire leurs générations, impressions et validation :

- Les coupons générés contiennent les informations suivantes :
 - Logo de « Karoura »
 - Nom du client
 - Numéro de coupon
 - Une combinaison unique entre un ID à 14 chiffres avec un PIN code à 4 chiffres
 - Un code PIN peut être associé à plusieurs ID
 - Date d'impression
 - Date d'expiration du coupon
 - Montant du coupon
- La validation des coupons se fait à travers les ID pour avoir la réduction du montant sur le montant total de la facture
- La suppression des coupons se fait automatiquement après une durée d'une année

2.1. D- Gestions des plaintes

Dans cette section on va détailler les fonctionnalités liées aux plaintes, et réclamations déposés par les clients auprès des opérateurs commerciaux de l'entreprise Karoura.

- Les plaintes sont enregistrées dans cette rubrique, avec une notification dans le module de CRM (*Customer Relationship Management*) pour les 3 motifs suivants :
 - Plainte sur la qualité de service.
 - Plainte sur les employés.

- Plainte sur la distribution (retard – défaut).

2.1. E- Gestions de stock

Dans cette section on va présenter l'onglet qui est lié à la gestion, c'est-à-dire que le module de call center offre la possibilité d'ajouter une commande, à la réception de l'appel d'un client suivant les caractéristiques du module « sales ».

2.2- Alertes et gestion des exceptions

On va présenter les alertes déclenchées sur Odoo suites aux événements survenus dans le module de Call center.

Les notifications et emails automatiques qui sont envoyés suites aux événements survenus sont gérés par Odoo.

- Le module de Call center notifie l'agent commercial d'appels aux clients en cas de non-paiement ou de retard de rendu des bouteilles.
- Le module de Call center notifie L'opérateur à la réception de la validation de la commande dans l'usine.
- Un Email automatique sera envoyé automatiquement pour les clients qui dépasseront un solde de crédit qui sera définit par le chef de l'entreprise Karoura.
- Une notification sera envoyée à l'opérateur lorsqu'un client aura dépassé la date de paiement, ou un solde supérieur à une limite définit par le chef de l'entreprise Karoura.
- Un Email de notification sera envoyé à l'arrivé d'un contrat d'une année avec un client, pour faire des campagnes de fidélisation.
- Les emails envoyés automatiquement seront enregistrés et visible dans la rubrique historique.

2.3- Statistiques

Dans cette section on va présenter la rubrique « Statistiques », les indicateurs qui la composent, le mode d'affichage de ces données, et enfin le type d'export des vues affichées.

Le module de Call center dispose d'une rubrique 'statistiques' définit par rapport aux critères suivants :

- Nombre d'appels par opérateur.
- Une durée.
- Par rapport aux clients.
- Les statistiques générées peuvent être présentés sous différentes vues comme les diagrammes ou les camemberts.

- Odoo donne la possibilité d'imprimer ces statistiques dans des rapports sous format PDF ou CSV.

2.4- Gouvernance d'accès

Dans cette section on va présenter les profils des utilisateurs, ainsi que les rôles qui leurs sont attribués dans le module de call center, pour permettre la confidentialité des données.

Les rôles de chaque utilisateur ont été définis par le chef de l'entreprise Karoura :

2.4. A- Profils utilisateurs

- Il existe 3 profils qui auront accès au module de call center avec 6 utilisateurs :
 - Un commercial (4 opérateurs).
 - Un superviseur.
 - Le gérant de l'entreprise.

2.4. B- Définition des rôles

Dans cette rubrique on va présenter les fonctionnalités affectées par utilisateur, le rôle de chacun d'eux.

- Le gérant de l'entreprise a le droit d'accéder à toutes les fonctionnalités du module.
- Seul le gérant de l'entreprise a le droit générer et d'imprimer les coupons.
- Le superviseur accéder aux statistiques générés et l'historique des appels, en plus des fonctionnalités disponibles pour l'opérateur commercial.
- Le superviseur peut modifier le statut d'un client.
- L'opérateur commercial a accès aux fonctionnalités suivantes :
 - Passer des commandes.
 - Consulter l'annuaire.
 - Ajouter des contacts dans l'annuaire.
 - Passer et décrocher des appels.
 - Consulter la fiche contact.
 - Ajouter un nouveau client.
 - Validation du coupon.
 - Enregistrement des plaintes.

2.5- Choix des indicateurs :

Dans cette section on va présenter les indicateurs qui peuvent aider les responsables de l'entreprise Karoura à la prise de décision.

III.5. Cahier de conception :

Après avoir déterminé l'activité sur laquelle l'étude décisionnelle va se porter, il faut relever des indicateurs clés génériques permettant d'extraire des rapports, contenant des statistiques de l'entreprise Karoura. Ces indicateurs ont été définis par le chef de l'entreprise lui-même.

Les indicateurs choisis pour répondre aux besoins prédéfinis dans le rapport, sont les suivants :

- Le nombre d'appels effectués par chaque opérateur.
- La durée d'appels pour chaque opérateur.
- Le chiffre d'affaires généré par chaque opérateur.
- Le chiffre d'affaire total par rapport à une période.
- Le chiffre d'affaires par client.

Dans cette section on va présenter la méthode de conception utilisée, concernant la réalisation des diagrammes UML.

III.5.1 Méthode de Conception

Les méthodes de conception sont utilisées afin d'améliorer la qualité de la conception finale.

La méthode de conception **UML** a été utilisée pour mettre en place le **Modèle Métier** du système,

On a utilisé deux diagrammes UML pour la conception de notre projet :

- Le diagramme de cas d'utilisation : il contient les acteurs du système ainsi que ses fonctions.
- Le diagramme de classes : représente les classes qui vont être développées, ainsi que les interactions entre elles.

Nous avons choisi ces deux diagrammes dans la mesure où on estime qu'ils représentent au mieux notre système,

Odoo s'appuie sur le design pattern MVC (Model View Control) pour son architecture, ce qui offre une meilleure flexibilité et maintenabilité.

Pour garantir ces deux critères, ce design pattern permet à Odoo de séparer le code en 3 niveaux :

- Le modèle : c'est là où sont stockées les données.
- Les vues : leurs tâches est de présenter les résultats renvoyés par le modèle, ainsi que de recevoir toute action de l'utilisateur
- Les classes contrôleurs : permettent de faire le lien entre le modèle et les vues, ainsi que toutes les fonctions du système (synchronisation, changement de vues, actions dans le système...).

III.5.2 Conception générale

Le call center (le centre d'appel) est avant tout une organisation humaine, chargée de prendre en compte les demandes de toute personne ou organisation en relation avec l'entreprise Karoura (client, Fournisseur, etc). Les opérateurs sont chargés dans un premier temps d'identifier leurs interlocuteurs à l'aide de quelques questions (numéro de client, adresse, numéro de téléphone, etc).

Dans l'onglet réservé à l'affichage des informations des clients, l'opérateur ou l'opératrice ouvre manuellement la fiche client, pour prendre connaissance de son dossier

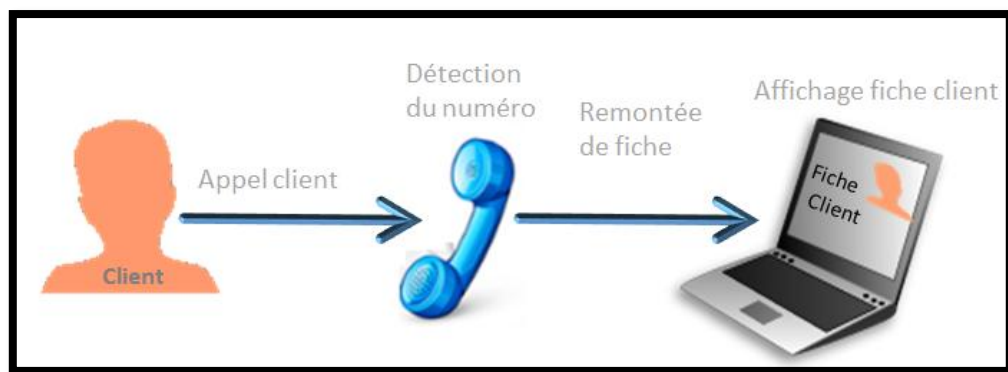


Figure -III.1- Identification des interlocuteurs

La figure -III.1- montre les étapes d'identification d'un client, à la réception d'un appel sur la ligne téléphonique fixe, l'opérateur enclenche une procédure d'identification d'appel. Cette procédure consiste à ce que l'opérateur recherche, sélectionne la fiche client correspondante et l'affiche sur l'interface du module de call center partir de l'appel téléphonique le système.

Lorsque le premier niveau d'opérateurs (on parle de *Front-Line*) n'est pas capable d'apporter une réponse à son interlocuteur, le dossier est transmis à des opérateurs de niveau 2, plus spécialisés. Le mécanisme qui consiste à transmettre le dossier à un niveau supérieur est appelé **escalade d'appels**.

III.5.3 Diagramme de cas d'utilisation

Dans cette section on va présenter le diagramme de cas d'utilisations, et cela en montrant les différents acteurs du système, ainsi que les cas d'utilisation résumant les fonctionnalités disponibles.

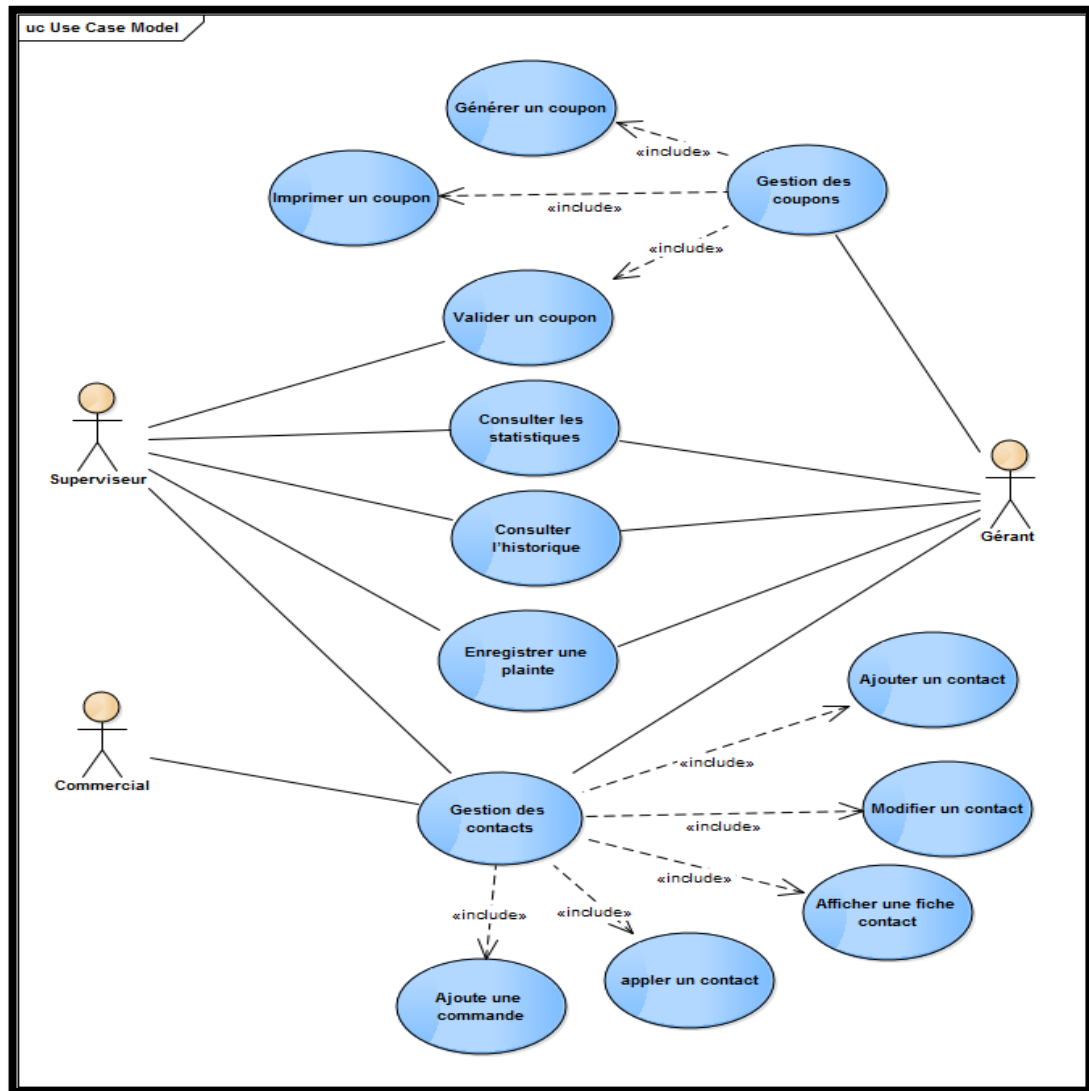


Figure –III.2- Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représenté dans la figure -III.2- illustre les cas d'utilisations repartis sur les Acteurs du module de call center sur Odoo.

Les détails du diagramme de cas d'utilisation seront présentés dans la section suivante.

1. Identification des profils acteur :

Il existe quatre profils qui vont utiliser le module du Call Center au sein de Karoura :

- **Gérant de l'entreprise** : c'est la personne qui a accès à toutes les fonctionnalités du module.
- **Superviseur** : il est chargé des vérifications des statistiques, de l'historique ainsi que de manager la relation client.
- **Agent commercial** : c'est le contact direct avec le client, il a accès aux rubriques de la gestion de stock/comptabilité et de la relation client.

Dans la suite de la section, on va détailler les cas d'utilisation du système, en expliquant le but de chacun, les pré conditions nécessaires et enfin les acteurs intervenants dans le système.

2. Fiche de description des cas d'utilisations :

Dans cette section on va détailler les cas d'utilisations, et ses acteurs, pour passer ensuite au diagramme de cas d'utilisation.

- Gérer les contacts

La gestion des contacts, est un ensemble de cas d'utilisation tel que l'ajout, la modification ainsi que l'affichage des contacts de Karoura. Plus détaillé dans le tableau dessus

Tableau 1: Gérer les contacts

But	Il regroupe tous les cas d'utilisations liés aux contacts pour avoir toutes informations nécessaires
Acteur	Gérant de l'entreprise Superviseur Agent Commercial
Pré Condition	Authentification avec un login et un mot de passe
Scénario	- L'utilisateur entre dans la rubrique contact

nominal	- l'annuaire est affiché - l'utilisateur fait une recherche
----------------	--

Par la suite on va détailler l'ensemble de cas d'utilisation, sur la gestion des contacts :

- **Ajouter un contact**

L'ajout un contact représente la saisit des informations dans la création d'un nouveau Contact (client ou partenaire). Plus détaillé dans le tableau dessus :

Tableau 2: Ajouter un contact

But	I Avoir les informations nécessaires pour contacter une personne ou un organisme
Acteur	Gérant de l'entreprise Superviseur Agent Commercial
Pré Condition	Authentification avec un login et un mot de passe
Scénario nominal	- L'utilisateur ouvre la rubrique Contact - L'utilisateur appuie sur le bouton ajouter un contact - L'utilisateur remplis le formulaire - L'utilisateur valide le les informations du contact

L'entreprise peut effectuer un ensemble d'opérations sur la liste de ses clients qui représentent le cas d'utilisation suivant :

- **Modifier un contact**

Le tableau ci-dessous va présenter les étapes de modifications des informations des contacts, en cas d'erreur ou de changement (exemple changement d'adresse). La suppression d'un contact est intégrée aussi. Plus de détails dans le tableau dessus :

Tableau 3: Modifier un contact

But	En cas d'erreurs, la possibilité est donnée pour modifier les informations
Acteur	Gérant de l'entreprise Superviseur Agent Commercial
Pré Condition	Authentification avec un login et un mot de passe
Scénario nominal	- Accéder à la rubrique contact - Sélectionner un contact - Cliquer sur le bouton modifier - Modifier les informations - Appuyer sur valider

- **Afficher une fiche contact**

L'affichage de la fiche Client est le point fort du Call Center, car il va permettre de visualiser les informations nécessaires de l'interlocuteur en ligne, afin développer une identification spontanée. Plus détaillé dans le tableau dessus :

Tableau 4: Afficher une fiche contact

But	Visualiser les informations nécessaires pour contacter une personne ou un organisme
Acteur	Gérant de l'entreprise Superviseur Agent Commercial
Pré Condition	Authentification avec un login et un mot de passe
Scénario	- Accéder à la rubrique contact

nominal	- Sélectionner un contact - La fiche contenant les informations s'affiche
----------------	--

- **Appeler un contact**

L'appel d'un contact représente la fonctionnalité basique d'un call center, l'opérateur peut effectuer un appel directement à travers l'interface de Odoo. Plus détaillé dans le tableau dessus :

Tableau 5: Appeler un contact

But	Contacter les clients pour des informations ou une commande
Acteur	Gérant de l'entreprise Superviseur Agent Commercial
Pré Condition	Authentification avec un login et un mot de passe
Scénarios nominal	<u>Scenario 1</u> - Accéder à la rubrique contact - Sélectionner un contact - Cliquer sur le bouton appeler <u>Scenario 2</u> - Afficher le Dialer - Composer un numéro - Cliquer sur appeler

- **Ajouter une commande**

L'interface mise en place va permettre de faciliter l'ajout d'une commande pour un client identifié en ligne lors de l'appel, Plus détaillé dans le tableau dessus :

Tableau 6: Ajouter une commande

But	Lors de la réception d'appels du client l'opérateur saisit sa commande contenant la quantité des bombonnes d'eau et de ses accessoires
Acteur	Gérant de l'entreprise Superviseur Agent Commercial
Pré Condition	Authentification avec un login et un mot de passe
Scénario nominal	- Sélectionner le contact identifié - Aller sur commande - Ajoute des produits la fiche de commande - Valider la commande

- **Enregistrer une plainte**

Lorsqu'un problème survient pour différentes raisons, l'opérateur enclenche une procédure de saisit de plainte. Les responsables seront notifiés afin de régler le problème. Le tableau ci-dessous illustre ce cas de figure.

Tableau 7: Enregistrer une plainte

But	Faire le suivi pour améliorer le service
Acteur	Gérant de l'entreprise Superviseur Agent commercial
Pré Condition	Authentification avec un login et un mot de passe
Scénario nominal	- Sélectionner le client concerné par la plainte - Aller sur ajouter une nouvelle plainte

	- Entrer les paramètres de la plainte - Valider la plainte
--	---

- **Générer un rapport**

La gestion des rapports représente les différentes utilisations décrites selon le besoins de l'Entreprise Karoura. Plus détaillé dans le tableau dessus :

Tableau 8: Générer un rapport

But	Garder une traçabilité et faire circuler l'information Odoo grâce à la configuration et la génération des rapports
Acteur	Gérant de l'entreprise Superviseur
Pré Condition	Authentification avec un login et un mot de passe
Description	La génération des rapports est gérée automatiquement Par Odoo, nous avons configuré les rapports, pour que les utilisateurs puissent y intégrer les statistiques générés. Ces rapports peuvent être exportés sous format PDF / Excel.

- **Gestion des coupons**

Le tableau ci-dessous présente le cas d'utilisation général pour la gestion des coupons, qui est un ticket de réduction sur le montant d'une facture. Il est offert par l'entreprise Karoura à des clients particuliers. Un cas d'utilisation délicat, qui ne peut s'effectuer que par le gérant de l'entreprise.

Tableau 9: Gérer les coupons

But	Est un cas d'utilisation intermédiaire qui regroupe tous les cas d'utilisations liés aux coupons
Acteur	Gérant de l'entreprise
Pré Condition	S'authentifier sur Odoo avec un login et un mot de passe

La Gestion des coupons inclus un ensemble de cas d'utilisation. Dans le détail d'opération effectué sur le concept des coupons, il s'agit de :

- **Générer un coupon**

Le tableau va présenter le cas d'utilisation générer un coupon. Pour la mise en place d'un ensemble de profils, qui permettent de profiter d'un ticket de réduction, et reste relatifs avec les commandes effectuer.

Tableau 10: Générer un coupon

But	Est de créer une liste de coupons d'achats pour permettre aux clients de les utiliser pour faire des achats
Acteur	Gérant de l'entreprise
Pré Condition	S'authentifier sur Odoo avec un login et un mot de passe
Scénario nominal	- Le gérant entre dans la rubrique générer les coupons - Sélectionne le nombre de coupons à générer - Clique sur le bouton générer

- **Imprimer un coupon**

Le tableau ci-dessous va présenter les étapes pour imprimer un coupon, Il s'agit d'un exemplaire de contrat, qui va être donné au client bénéficiant d'un coupon.

Tableau 11: Imprimer un coupon

But	Imprimer les coupons afin de les délivrer aux clients
Acteur	Gérant de l'entreprise
Pré Condition	S'authentifier sur Odoo avec un login et un mot de passe
Scénario nominal	- Accéder à rubrique gestion des coupons - Sélectionner un nombre à imprimer - Cliquer sur imprimer

- **Valider un coupon**

La validation du coupon qui a été généré, va être prise en considération lors de l'achat. Le but de la validation est d'offrir au client une réduction sur le montant total des créances. Plus détaillé dans le tableau dessus

Tableau 12: Valider un coupon

But	Est de faire des réductions à des clients par les coupons générés auparavant
Acteur	Gérant de l'entreprise Superviseur
Pré Condition	Authentification avec un login et un mot de passe.
Scénario nominal	- Entrer dans la rubrique gestion des coupons - Entrer le ID à 14 chiffres - Entrer le code pin à 4 chiffres - Entrer le nom du client - Appuyer sur « valider »

Après la gestion des coupons, On enchaîne avec le reste des cas d'utilisations :

- **Consulter les statistiques**

La Consultation des statistiques va permettre la visualisation de résultats, citées dans le cahier de charge. Un cas d'utilisation qui va permettre de faire un bon suivi. Plus détaillé dans le tableau dessus :

Tableau 13: Consulter les statistiques

But	Est de connaître les fréquences de l'activité par période et par opérateur afin de prendre des décisions
Acteur	Gérant de l'entreprise Superviseur
Pré Condition	Authentification avec un login et un mot de passe
Scénario nominal	- L'utilisateur ouvre la rubrique Statistiques. - Le système affiche les rapports autorisés et paramétrés - L'utilisateur sélectionne un rapport. - Le système affiche les valeurs de paramètres saisies par date.

- **Consulter l'historique**

La consultation de l'historique, permet d'examiner les traces de toutes les actions effectuées pour chaque client, et chaque opérateur. Les paramètres de l'historique (date, heure...) sont exprimé dans le cahier de charge. Plus détaillé dans le tableau dessus :

Tableau 14: Consulter l'historique

But	Pour garder une traçabilité des actions effectuées au sein du module et leurs émetteurs
Acteur	Gérant de l'entreprise Superviseur
Pré	Authentification avec un login et un mot de passe

Condition	
Scénario nominal	- L'utilisateur accède à la rubrique historique - Il sélectionne la période dans laquelle il veut afficher l'historique - L'historique s'affiche

Après la description des différentes interactions avec le module du Call Center, On va enchaîner avec le diagramme de classe.

III.5.4 Diagramme de classes

Dans cette section on va présenter le diagramme de classes de notre système.

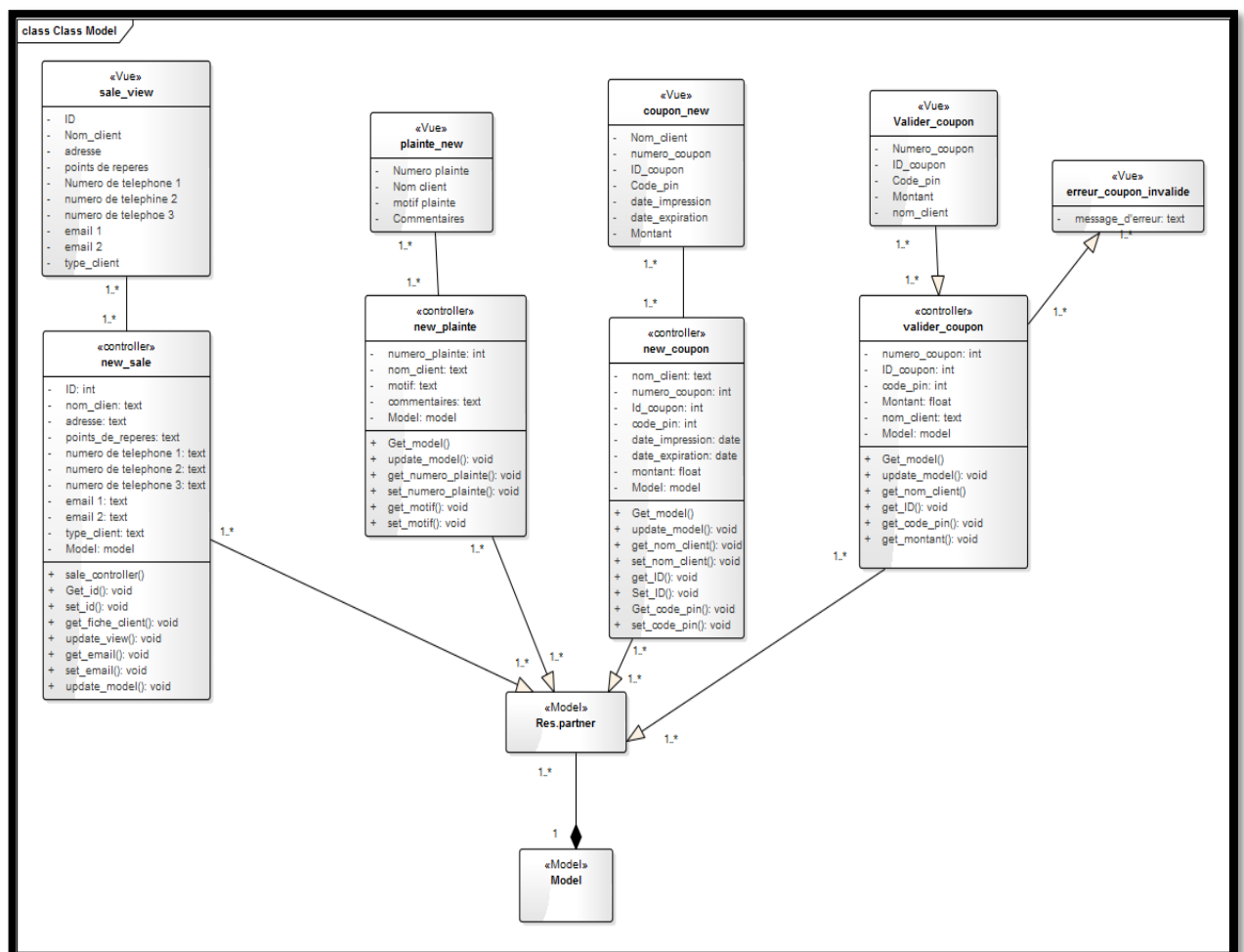


Figure -III.3- Diagramme de classes

Le diagramme de classe respecte le pattern MVC, les classes sont réparties en 3 stéréotypes

- Le stéréotype « Vue » regroupe les classes représentant les vues du module.
- Le stéréotype « Modèle » regroupe les classes représentant les modèles du module.
- Le stéréotype « Contrôle » regroupe les classes représentant les contrôleurs du module.

Le diagramme est composé de classes existantes dans Odoo, des classes qu'on a ajoutées.

Les classes existantes sont :

- Les vues représentants les formulaires à remplir (Nouveau contact, Plainte, Coupon),
- La vue qui représente la fiche client.
- Les contrôleurs responsables de chaque Vue pour l'affichage, récupération des données et les changements de vues.
 - Ces contrôleurs sont aussi le lien entre les vues et le modèle.

Conclusion

Nous avons réalisé dans ce chapitre le recueil de besoins auprès de es représentants de la société Karoura pour passer ensuite à la rédaction du cahier des charges et le cahier de conception la conception UML. Cette conception est une étape nécessaire et très importante pour pouvoir réaliser le module, le prochain chapitre détaillera les étapes de l'implémentation du module de Call center sur Odoo.

Chapitre IV : Implémentation

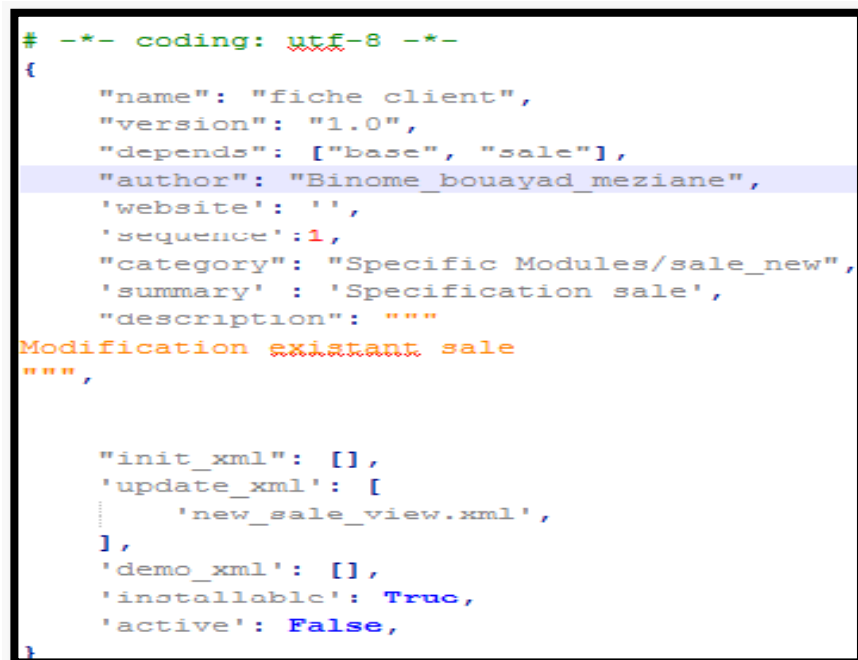
IV. Introduction

Dans ce chapitre nous allons détailler les étapes de l'implémentation du module de Call Center sur Odoo en expliquant la structure d'un module, ainsi que les workflows du système. Enfin nous allons montrer des captures d'écrans illustrant tout le travail réalisé.

IV.1. Structure d'un module

La composition d'un module est structurée d'une manière à appliquer le patron de conception MVC, c'est-à-dire qu'il y aura les fichiers suivants :

- Les fichiers de configuration du module qui sont « `__openerp__` » et « `_init_` », écrits en Python.
- Les fichiers contrôleurs et les modèles écrits en Python
- Les fichiers représentant les vues sont écrits en XML.



```
# -*- coding: utf-8 -*-
{
    "name": "fiche client",
    "version": "1.0",
    "depends": ["base", "sale"],
    "author": "Binome bouayad meziane",
    "website": '',
    "sequence": 1,
    "category": "Specific Modules/sale_new",
    'summary' : 'Specification sale',
    "description": """
Modification existant sale
""",

    "init_xml": [],
    'update_xml': [
        'new_sale_view.xml',
    ],
    'demo_xml': [],
    'installable': True,
    'active': False,
}
```

Figure –IV.1- Fichier de configuration "`_openerp_`"

La figure ci-dessus représente le fichier de configuration du module, où on déclare :

- Son nom
- Sa version.
- Le nom de l'auteur.
- La catégorie du module.
- Sa description.
- Les fichiers qui ont une relation avec le module.
- Sa dépendance par rapport aux autres modules

Dans ce cas le module a une dépendance avec « sales » et « base », qui sont des modules existants dans Odoo.

- Le module « sales » gère tout ce qui est en rapport avec les ventes (commande, facture, produits,...)
- Le module « base » représente le noyau de Odoo, il gère toutes les fonctions du système.

```
# -*- coding: utf-8 -*-
from openerp import fields, models, api

class res_partner(models.Model):
    _name = "res.partner"
    _inherit = "res.partner"

    Nom = fields.Char(string="Nom", required=True, size=30)
    Adresse = fields.Char(string="Adresse", required=True, size=30)
    Point_de_repere = fields.Char(string="Point de repere", required=True, size=200)
    Numero_de_telephone1 = fields.Char(string="Numero de telephone 1", size=30)
    Numero_de_telephone2 = fields.Char(string="Numero de telephone 2", size=30)
    Numero_de_telephone3 = fields.Char(string="Numero de telephone 3", size=30)
    Email1 = fields.Char(string="Email 1", size=30)
    Email2 = fields.Char(string="Email 2", size=30)
    Commentaires = fields.Char(string="Commentaires", size=200)
    registre_commerce = fields.Char(string = 'N° Registre commerce', size=30)
    id_fiscal = fields.Char(string = 'N° Identification fiscale', size=30)
    articl_impo = fields.Char(string = "Article d'imposition", size=30)
```

Figure -IV.2- Exemple de fichier modèle

Du dossier openerp on va importer les Fields, les modèles et API. Le model de la classe res.partner se composent de champs qui vont stocker les données du formulaire.

Ce modèle hérite du modèle res.partner, on fait la déclaration de la variable avec son nom et le type de son contenu.

Les champs peuvent être de différents types :

- **Char.**
- **Date.**
- **Float.**

Il existe d'autres types de champs qu'on appelle les champs relationnels qui lient entre les enregistrements d'un même modèle ou de différents modèles.

Les champs relationnels sont :

- **Many2One** : ce type permet de relier plusieurs enregistrements d'un modèle vers un seul enregistrement d'un autre modèle. On prend l'exemple suivant
 - L'ID d'un opérateur est relié à plusieurs ID clients, car un opérateur possède un ensemble de clients à gérer.
- **One2Many** : c'est l'inverse d'un Many2One. l'accès d'un enregistrement d'un modèle il en résulte un (éventuellement vide) ensemble d'enregistrements d'un autre modèle. On prend l'exemple suivant
 - Plusieurs ID produits sont relié à une seule monnaie,
- **Many2Many** : Tout enregistrement d'un modèle peut être lié à un certain nombre d'enregistrements d'un autre modèle. Cette relation peut être aussi bidirectionnelle. On prend l'exemple suivant pour mieux expliquer :
 - Un ensemble de taxes peut être relié à un ensemble de produits.

Ces types représentent les clés étrangères entre les différents modèles

Dans l'exemple présenté de fichier qu'on a réalisé nous avons déclaré juste des champs de type Char.

Dans les autres fichiers on a utilisé les autres types de champs suivant le besoin par exemple :

- Lorsqu'on a déclaré dans la partie des coupons un code pin, il peut valider plusieurs ID on a utilisé le One2Many.

```
?xml version="1.0" encoding="UTF-8"?>
openerp>
<data>
  <record id="view_sale_new_form" model="ir.ui.view">
    <field name="name">view.sale.new.form</field>
    <field name="model">res.partner</field>
    <field name="inherit_id" ref="base.view_partner_form"/>
    <field name="arch" type="xml">
      <xpath expr="//page[1]" position="before">
        <page string="Informations">
          <group>
            <group col="2">
              <field name="Nom"/>
              <field name="Adresse"/>
              <field name="Points_de_reperes"/>
              <field name="Numero_de_telephone 1"/>
              <field name="Numero_de_telephone 2"/>
              <field name="Numero_de_telephone 3"/>
              <field name="Email_1"/>
              <field name="Email_2"/>
              <field name="Commentaires"/>
              <field name="registre_commerce"/>
              <field name="id_fiscal"/>
              <field name="artici_l_impo"/>
            </group>
          </group>
        </page>
      </xpath>
    </field>
  </record>
```

Figure -IV.3- Exemple de fichier représentant une vue

Ce fichier représente la vue du model, c'est le fichier XML, représentant les données du fichier new_sale.py. On définit un enregistrement qui représente la vue qui contient les champs, ces derniers sont :

- Le nom de la vue.
- Le type de la vue (dans ce cas nous avons déclaré que c'est un formulaire).
- On relie le model « res.partner » à cette vue
- On déclare l'arborescence du model.

Dans Odoo, une page représente une des interfaces du système, elle regroupe un ensemble de « Groups ».

Les « groups » séparent l'interface utilisateur en blocs d'affichage des données, c'est-à-dire qu'on peut regrouper un ensemble de données pour les afficher dans un endroit.

Dans notre travail, nous avons déclaré les informations dans un seul groupe.

IV.1.1 La notion de l'objet dans Odoo

Tous les composants de données dans Odoo sont accessibles par le biais des «objets». A titre d'exemple, il est un objet de **res.partner** pour accéder aux données concernant les partenaires, un objet **account.invoice** pour les données concernant les factures..

Un objet est relié à un type de ressource, et non pas un objet par ressource. Nous avons donc un objet **res.partner** pour gérer l'ensemble des partenaires et non pas un objet **res.partner** par partenaire.

Dans Odoo le modèle de données est décrit et manipulé par les classes et objets. Ces actions sont gérées par l'ORM.

IV.2. Workflow

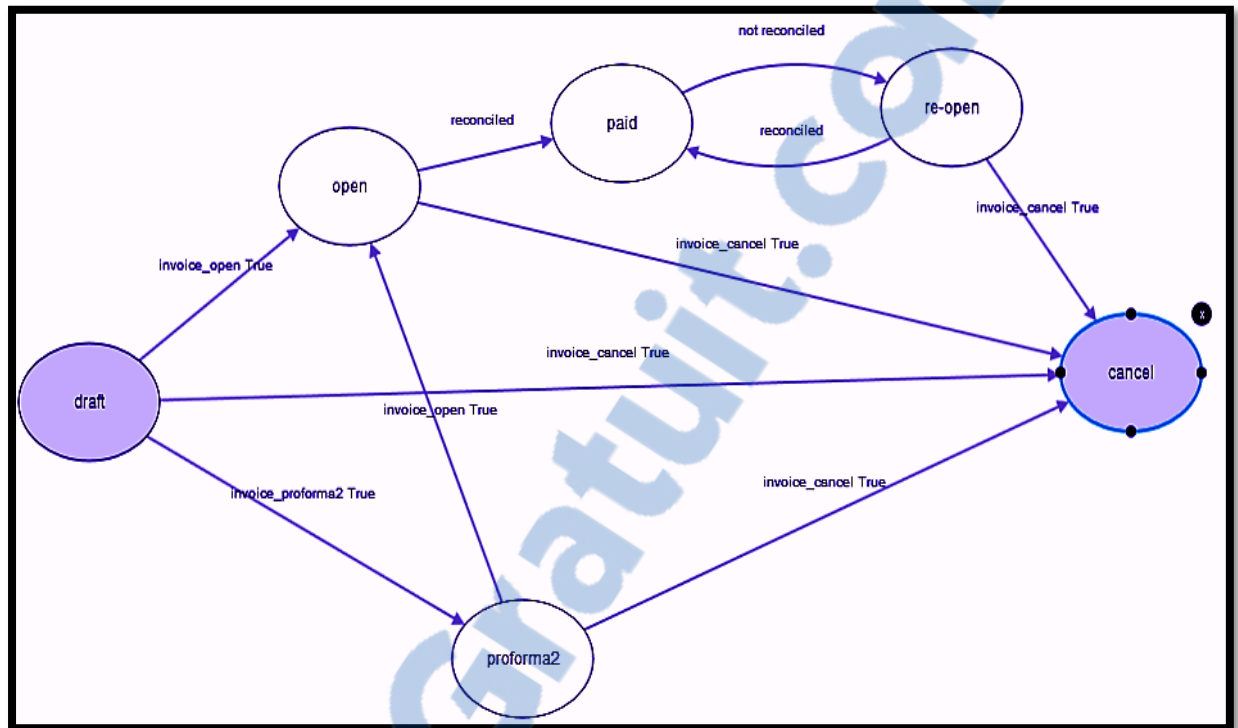


Figure -IV.4- Workflow d'une commande

Dans cette section on va présenter les workflow de notre système. Ce workflow Standard montre les états d'une commande.

Le workflow a un état initial ou la commande est « draft » et suivant les déclencheurs l'état de la commande évolue.

- Le brouillon devient une facture pro-forma si la condition `invoice_proforma` est satisfaite.
- Le brouillon devient une facture ouverte si la condition `Invoice_open` est vraie
- Et à la fin la facture passe à l'état payer.
- A n'importe quel état on peut annuler une commande si la condition `invoice_cancel` est vraie.

Nous avons appliqué ce workflow sur notre système, pour gérer les commandes faites par les clients.

IV.3. Captures d'écrans

Dans cette section nous allons présenter les différentes captures d'écrans de notre module.

Odoo partage la même base de données pour tous les modules disponibles.

Lorsqu'on installe le module le système affiche la liste des clients déjà existants de l'entreprise Karoura.

Il existe un bouton ajouter en rouge, en haut pour ajouter un nouveau client

Lorsqu'on clique sur le bouton la vue suivante s'affiche :

Figure -IV.5- Ajout d'un nouveau client

Le formulaire d'ajout d'un client ci-dessus existe déjà dans Odoo. Nous l'avons adapté aux caractéristiques émises par le client dans le cahier des charges.

La vue change en fonction du type de client s'il est un « particulier » ou une « entreprise », en cochant le bouton à côté du nom.

Cet affichage représente le module qu'on a présenté auparavant.

Lors de la saisi des informations, nous avons réalisé un autre module qui vérifie la validité du numéro de téléphone saisi, et transforme le « 0 » du début en « +213 »

A la validation de ce formulaire la fiche client suivante s'affiche :

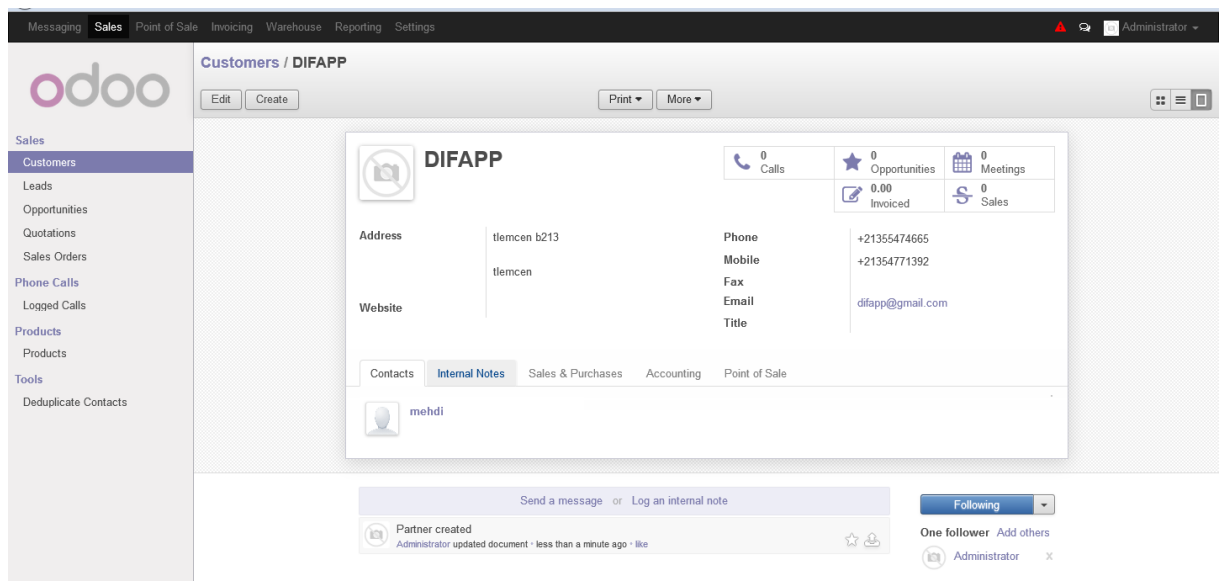


Figure -IV.6- Fiche client

Cette fiche représente les informations du client que ce soit un particulier ou une entreprise.

En bas de la fiche client il y a des onglets pour consulter les détails des commandes passées par le client.

Cette étape existe déjà dans Odoo et nous l'avons adapté à nos besoins.

- En haut à droite il y a un bouton pour Appeler le client. Au click du bouton un Pop-up s'affiche pour permettre d'appeler le contact via la technologie VOIP.

Le pop-up qui s'affiche est gérée par un module qu'on a développé

Dans la partie qui va suivre nous allons donner un bout de code de ce module et l'expliquer.

```

# -*- coding: utf-8 -*-

from openerp.osv import orm, fields
from openerp.tools.translate import _
import logging

logger = logging.getLogger(__name__)

class phone_common(orm.AbstractModel):
    _inherit = 'phone.common'

    def _prepare_incall_pop_action(
        self, cr, uid, record_res, number, context=None):
        action = False
        if record_res:
            obj = self.pool[record_res[0]]
            action = {
                'name': obj._description,
                'type': 'ir.actions.act_window',
                'res_model': record_res[0],
                'view_mode': 'form,tree',
                'views': [[False, 'form']],
                'target': 'new',
                'res_id': record_res[1],
            }
        else:
            action = {
                'name': _('Number Not Found'),
                'type': 'ir.actions.act_window',
            }

```

Figure -IV.7- Bout de code du pop-up

Dans cette partie nous allons expliquer le module gérant les pop-up

- Tout d'abord nous allons commencer par importer les champs et ORM
- Nous allons importer les méthodes de traduction de la langue anglaise vers la langue française grâce à **openerp.tools.translate**
- Nous allons importer le logging permettant d'avoir une trace des actions survenues dans le système.
- On définit notre logger pour l'enregistrement des actions
- On définit la classe **phone_comon** qui va contenir les méthodes gérant les actions du pop-up et le contenu à afficher.
- La méthode **_prepare_incall_pop_action** gère l'affichage du contenu du pop-up
- Les paramètres de cette méthode sont :
 - L'objet lui-même.
 - L'identifiant de l'utilisateur.
 - L'enregistrement.

- Numéro de l'enregistrement.
- si l'enregistrement correspond à l'instance self. Pool enregistré dans pool, le système va afficher une fenêtre contenant les informations du contact à appeler :
 - Object description : représente la description de fenêtre qui est un objet.
 - Action : c'est le type d'affichage qui est une fenêtre
 - Cette fenêtre contient l'enregistrement du contact à appeler dans le modèle.
 - Les informations du contact sont affichées dans un formulaire.
- Si la condition précédente n'est pas vérifiée une fenêtre s'affiche avec le message « numéro introuvable ».

Par la suite nous allons montrer la fiche du produit.

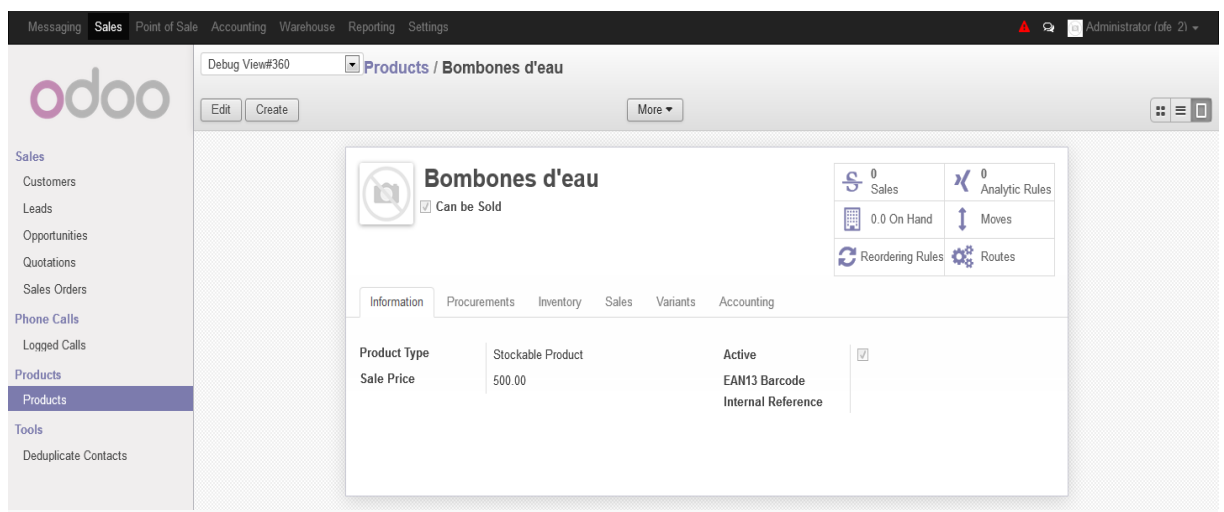


Figure -IV.8- Fiche du produit « Bonbonne d'eau »

La fiche produit avec ses caractéristiques (nom, prix, quantité....) est gérée par le module Product. Ce module existe déjà dans Odoo, nous l'avons intégré à notre module pour permettre à l'agent commercial de saisir les produits dans une commande.

Dans ce qui va suivre nous allons décrire le processus d'une commande comme détaillé auparavant dans la section des workflow.

Figure –IV.9- Ajout d'une facture « Brouillon »

La facturation est gérée par deux méthodes principales qui sont :

- **action_invoice_create** est la méthode dans le module de vente (vente / sale.py) qui crée la facture.
- La méthode « **invoice_line_create** » ainsi que « **prepare_order_line_invoice_line** » crée des lignes de facturation.

Après que la facture « brouillon » soit créée elle va contenir les informations suivantes :

- Les produits qu'on saisit dans le tableau avec leurs quantités.
- Les informations du client.

Il existe un module qui intègre les taxes aux prix des produits, Odoo calcule automatiquement le prix définitif de chaque produit.

A la fin le montant total de la facture « brouillon » est calculé automatiquement par Odoo.

Le module de « Sale » est un des modules que nous avons intégrés à notre module de call center.

Debug View#559 Customer In... / SAJ/2016/0001

Send by Email Print Register Payment Refund Invoice

Invoice SAJ/2016/0001

Customer: DIFAPP tiemcen b213 tiemcen Invoice Date: 05/17/2016 Journal: Sales Journal (DZD) Account: 110200 Debtors Currency: DZD

Fiscal Position

Invoice Lines Other Info Payments

Product	Description	Account	Analytic Distribution	Quantity	Unit Price	Taxes	Amount
Bombones d'eau	Bombones d'eau	200000 Product Sales		50.000	500.00	17%	25000.00

Figure –IV.10- Ajout d'une facture

Cette figure représente la validation d'une facture. Cette étape survient après la saisie de la facture « brouillon » on clique sur le bouton valider pour générer la facture.

En se basant sur le workflow de la vente c'est la dernière étape avant que la facture soit à l'état « payé »

Le serveur Odoo récupère la facture « brouillon » du modèle, après sa validation la commande passe de l'état facture « brouillon » à l'état facture.

Debug View#557 Customer Invoices

Create or Import

Customer	Invoice Date	Number	Responsible	Due Date	Source Document	Currency	Balance	Subtotal	Total	Status
DIFAPP	05/17/2016	SAJ/2016/0001	Administrator	05/17/2016		DZD	25000.00	25000.00	25000.00	Open
							25000.00	25000.00	25000.00	

Figure -IV.11- Liste de factures effectuées par client

Une autre fonctionnalité de notre module c'est la consultation des factures effectuées par clients. On y accède grâce au menu qui est à gauche. Cette fonctionnalité fait partie du module « sales » qu'on a intégré.

Le server Odoo récupère la liste des factures du modèle grâce au module « sale » pour que chaque ligne de ce tableau contienne les informations suivantes

- Nom du client.
- Date de la facture.
- Le numéro de la facture.
- La monnaie (en cas il y a des clients étrangers).
- Le montant de la facture.

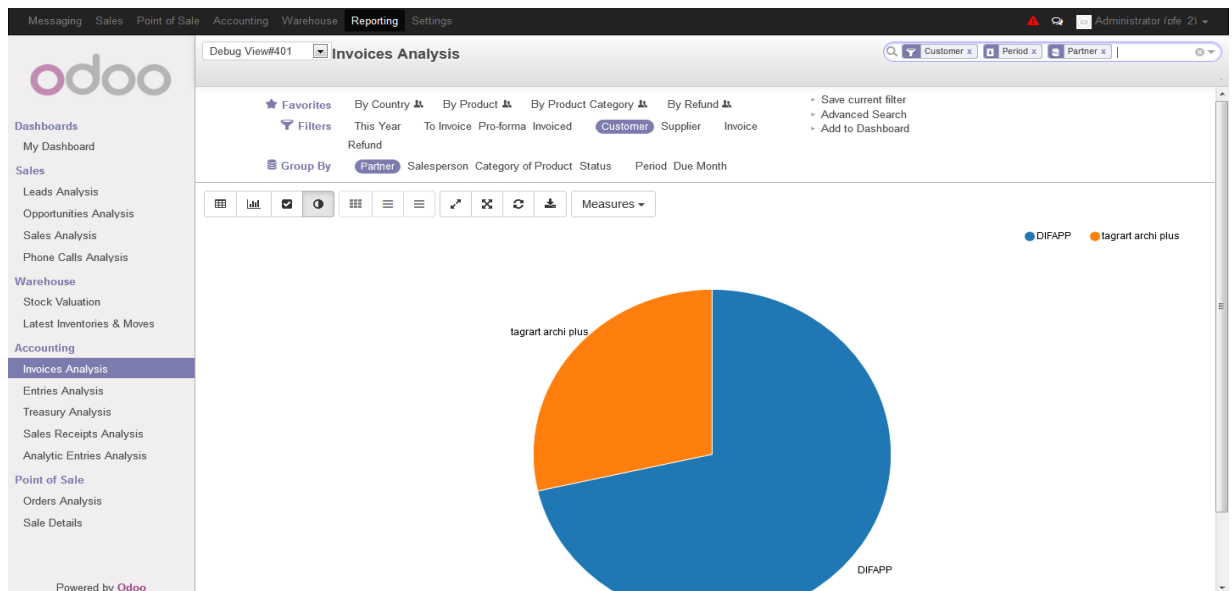


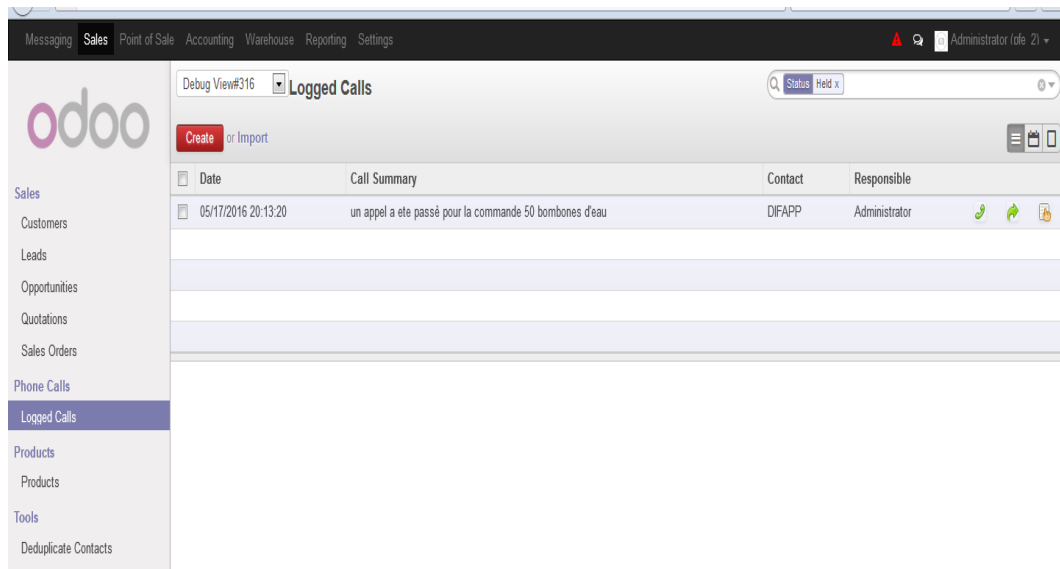
Figure -IV.12- Statistiques de factures effectuées par client

Odoo offre la possibilité aux utilisateurs d'afficher les statistiques suivant plusieurs formats. Nous avons choisi le graphique sous forme de camembert

Odoo donne aussi la possibilité d'avoir différents filtres pour les données à afficher grâce au module de « reporting »

Pour calculer les statistiques réelles sur les objets, on définit un objet statistique qui est une vue de PostgreSQL. Ensuite on crée une arborescence et une vue graphique sur cet objet

Nous avons intégrés cette fonctionnalité à notre module, pour donner la possibilité aux dirigeants de Karoura d'avoir les statistiques permettant une prise de décision efficace.



The screenshot shows the 'Logged Calls' interface in Odoo. The left sidebar contains a menu with 'Sales' selected, and sub-items like 'Customers', 'Leads', 'Opportunities', 'Quotations', 'Sales Orders', 'Phone Calls', 'Logged Calls' (highlighted), 'Products', 'Tools', and 'Deduplicate Contacts'. The main area displays a table of logged calls.

Date	Call Summary	Contact	Responsible
05/17/2016 20:13:20	un appel a ete passé pour la commande 50 bombones d'eau	DIFAPP	Administrator

Figure -IV.13- Détails d'appel avec un client

Cette capture d'écran montre la liste des appels effectués par client. Chaque ligne contient :

- Nom du client.
- Le nom de l'opérateur.
- Date et heure de l'appel.
- Une description de la conversation lors de l'appel.

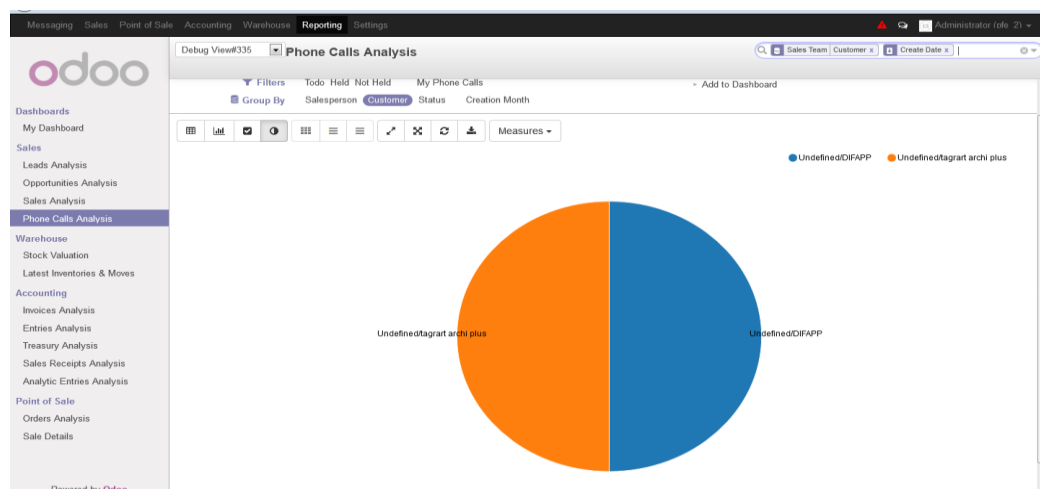


Figure -IV.14- Statistiques par nombre d'appels effectués par client

Cette figure se base le tableau précédant d'appel par client. Ces statistiques sont présentées sous forme de camembert.

Conclusion

Nous avons présenté dans ce chapitre la structure d'un module Odoo, les flux de travail qui existent dans notre module. Pour finir nous montrés quelques captures d'écrans de notre module.

- CONCLUSION GENERALE-

Au cours de ce stage de projet de fin d'étude nous avons eu la chance d'intégrer l'entreprise DigitalSide, nous nous sommes impliqués dans les différentes étapes cycle de vie de réalisation du Call center. Nous avons été incorporés à l'équipe fonctionnelle puis technique. De ce fait nous avons pu suivre les différentes étapes : du recueil de besoin jusqu'au développement du module Odoo en passant par la rédaction du cahier des charges.

Le travail en entreprise nous a permis de nous développer sur le plan humain, professionnel, et technique. Nous avons acquis des compétences dans plusieurs domaines, on peut citer quelques-unes :

- La communication interne/externe c'est-à-dire avec les employés de DigitalSide et Karoura.
- Le travail d'équipe.
- La planification des tâches.
- La gestion des projets informatiques.
- La rédaction des cahiers des charges.

Nous avons eu la possibilité de nous déplacer au sein des locaux de l'entreprise Karoura à Alger. A travers les réunions que nous avons effectuées avec leurs représentants, nous avons pu connaître les processus de travail de l'entreprise, ainsi que leurs besoins en ce qui concerne le module de call center.

Ce projet nous a permis aussi de nous perfectionner en améliorant nos connaissances en conception et d'acquérir des nouvelles compétences en programmation.

Nous avons essayé d'appliqué au maximum possible les règles de bases permettant d'avoir une application performante. Nous avons commencé par la modélisation de notre système en utilisant le langage UML grâce à la mise en place de diagrammes de cas d'utilisation et de classes.

Cela s'est fait en suivant les étapes du processus UP. Notre démarche était pilotée par les cas d'utilisations

Le développement du module de call center nous a offert la possibilité de connaître la logique de développement de Odoo. Nous avons appliqué les bonnes pratiques de la programmation à travers le développement de sous modules pour former le module principal.

Cet aspect de la programmation nous a permis de mieux localiser et gérer les erreurs survenues dans le système, sans que cela n'affecte le reste des modules.

Odoo utilise le design pattern MVC, cela nous a donné la possibilité de mettre en pratique les connaissances acquises lors de notre cursus.

Nous avons eu la chance d'avoir une formation sur Odoo et en langage Python de la part d'un membre de l'équipe technique de DigitalSide. L'apprentissage d'un nouveau langage de programmation dans un laps de temps aussi court était un challenge en soi cela nous a montré qu'on pouvait se surpasser pour mener à bien ce projet.

Au terme de ce projet de fin d'études, nous estimons avoir atteint les principaux objectifs que nous nous étions fixés, ainsi que ceux qui nous été assignés de la part de l'entreprise DigitalSide. Néanmoins, quelques améliorations sont proposées, nous citons :

- L'appel vidéo.
- L'envoi des factures pro-forma aux clients directement via l'interface du module.
- Intégration de la gestion de mise en attente des appelants en cas où l'opérateur soit indisponible ou occupé.

Nous avons été confrontés à un certain nombre de difficultés que nous n'avions pas envisagées au départ. En effet, nous avons découvert lors de l'installation de Odoo que Windows en tant que système d'exploitation n'offre pas toutes les autorisations à Odoo. Un autre problème que nous avons rencontré c'est la mauvaise gestion de la mémoire cache de Windows, c'est-à-dire que même après l'enregistrement des modifications d'un des modules, le changement ne s'opérait pas sur le système, il fallait redémarrer le service Odoo à nouveau pour obtenir les modifications.

Néanmoins, ces difficultés nous ont permis d'élargir nos perspectives et nos connaissances et de mieux orienter nos axes de travail futurs ainsi que nous forger une bonne expérience de travail.

- REFERENCES BIBLIOGRAPHIQUES -

- [1] <https://fr.wikipedia.org/wiki/Odoo> consulté le 04 mars 2016
- [2] https://www.odoo.com/fr_FR/ consulté le 20 Avril 2016
- [3] Maza TAKIEDIINE, Présentation OpenERP de DigitalSide consulté le 20 Avril 2016
- [4] <https://www.odoo.com/documentation/8.0/reference/workflows.html> consulté le 2 Mai 2016
- [5] <https://linuxfr.org/news/publication-de-la-licence-gnu-affero-general-public-licence-ve> consulté le 20 mars 2016
- [6] <https://fr.wikipedia.org/wiki/PostgreSQL> consulté le 20 mars 2016
- [7] http://doc.fedora-fr.org/wiki/Installation_et_configuration_de_PostgreSQL consulté le 20 mars 2016
- [8] [https://fr.wikipedia.org/wiki/Python_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)) consulté le 21 Mars 2016
- [9] <http://www.labsolution.lu/solutions-fr/technologies/11492/> consulté le 21 Mars 2016
- [10] www.irif.univ-paris.diderot.fr/~carton/Enseignement/XML/Cours/Annexes/presentation.html consulté le 1 Avril 2016
- [11] https://fr.wikipedia.org/wiki/Voix_sur_IP consulté le 1 avril 2016
- [12] Seladji Yassamine Cours Université de Tlemcen (Design Pattern) consulté le 15 Avril 2016
- [13] Seladji Yassamine Cours Université de Tlemcen (MVC) consulté le 15 Avril 2016
- [14] <http://sabricole.developpez.com/uml/tutoriel/unifiedProcess/> consulté le 04 mars 2016



Liste des Figures

Chapitre I

Figure -I.1- Support d'authentification de la plateforme	7
Figure -I.2- Forum de discussion de la plateforme	7
Figure -I.3- L'état d'avancement des tâches	8
Figure -I.4- Espace fichiers	8

Chapitre II

Figure -II.1- Schéma représentant les modules disponibles dans Odoo	13
Figure -II.2- Architecture 3 tiers Odoo	14
Figure -II.3- Architecture logiciel Odoo	14
Figure -II.4- Exemple de flux de travail	16
Figure -II.5- Historique des versions de Odoo	17
Figure -II.6- Schéma d'itérations du processus UP	21
Figure -II.7- Le modele 4+1 vues	22
Figure -II.8- Exemple d'un diagramme de cas d'utilisation	22
Figure -II.9- Cycle de vie du processus UP	23

Chapitre III

Figure -III.1- Identification des interlocuteurs	35
Figure -III.2- Diagramme de cas d'utilisation	36
Figure -III.3- Diagramme de classes	46

Chapitre IV

Figure -IV.1- Fichier de configuration "_openerp_"	49
Figure -IV.2- Exemple de fichier modèle	50
Figure -IV.3- Exemple de fichier représentant une vue	52
Figure -IV.4- Workflow d'une commande	54
Figure -IV.5- Ajout d'un nouveau client	55
Figure -IV.6- Fiche client	56
Figure -IV.7- Bout de code du pop-up	57
Figure -IV.8- Fiche du produit « Bonbonne d'eau »	58
Figure -IV.9- Ajout d'une facture « Brouillon »	59
Figure -IV.10- Ajout d'une facture	60
Figure -IV.11- Liste de factures effectuées par client	60
Figure -IV.12- Statistiques de factures effectuées par client	61
Figure -IV.13- Détails d'appel avec un client	62

Figure -IV.14- Statistiques par nombre d'appels effectués par client	62
---	-----------

Liste des Tableaux

Tableau 1 : Gérer les coupons	43
Tableau 2 : Générer un coupon	43
Tableau 3 : Imprimer un coupon	44
Tableau 4 : Valider un coupon	44
Tableau 5 : Consulter les statistiques.....	45
Tableau 6 : Consulter l'historique	45
Tableau 7 : Gérer les contacts	37
Tableau 8 : Ajouter un contact.....	38
Tableau 9 : Modifier un contact	39
Tableau 10 : Afficher une fiche contact	39
Tableau 11 : Appeler un contact	40
Tableau 12 : Ajouter une commande	41
Tableau 13 : Enregistrer une plainte	41
Tableau 14 : Générer un rapport	42

Liste des Acronymes

UML : Unified Modelling Language
XML : eXtensible Markup Language
AGPL : Affero GNU General Public Licence
VOIP : Voix via IP
ORM /Object Relationnal Mapping

- RESUME -

L'intérêt de ce travail est d'automatiser la gestion du call center pour une entreprise. Cette automatisation se traduit par la mise en œuvre d'un système qui intègre les fonctionnalités d'un Customer Relationship Management

L'utilité de ce système est de faciliter la gestion de l'entreprise pour l'administratif et garder un contact permanent avec le client.

La modélisation de notre module a été faite par UML en s'appuyant sur le processus UP. La base de données et le module ont été respectivement gérés par Odoo.

Odoo utilise comme SGBD PostgreSQL et les langages Python, XML réputés par leurs puissances, dynamismes et efficacités.

Abstract

In the context of presenting a final project study, efforts have been devoted to the implementation of a system to facilitate the management of a Call center for an Algerian company.

To achieve this goal, we have developed a module in the ERP Odoo.

The modeling our system was made by following, the UML UP steps the process. The implementation of the application was carried out under the languages used by Odoo: Python, XML and PostgreSQL as DBMS.

ملخص

في إطار تقديم مشروع نهاية الدراسة لي ماستر تم بذل مجهودات كبيرة لإنجاز برنامج معلوماتي لإدارة مركز الاتصالات لمؤسسة جزائرية في هذا نحو تم إنجاز موديول في نظام تخطيط المؤسسات Odoo

قمنا بدراسة تحليلية وتم نمذجة نظامنا عن طريق UML وإتباع مراحل UP. بعده تم تطوير موديول عن طريق لغات الحاسوب Python، XML، التي يستعملها Odoo و PostgreSQL كنظام إدارة قواعد البيانات.