

**République Algérienne Démocratique et Populaire**

**Université Abou Bakr Belkaid– Tlemcen**

**Faculté des Sciences**

**Département d'Informatique**

**Mémoire de fin d'études**

**Pour l'obtention du diplôme de Master en Informatique**

*Option : Génie logiciel (G.L)*

**THEME**

**Création d'une Application Mobile de Gestion de  
Notes de Frais et Synchronisation avec Odoo**

*Réalisé par :*

- Sarra ZENNAKI.

Présenté le : 06/07/2019 devant le jury composé de :

- |                           |                    |
|---------------------------|--------------------|
| - Mme. HALFAOUI Amel      | - Présidente.      |
| - M. MATALLAH Houcine     | - Examineur.       |
| - Mme. BENMANSOUR Fazilet | - Encadrant.       |
| - M. BRAHMI Noureddine    | - Maître de stage. |

Année universitaire : 2018-2019

---

## Remerciement

---

*Je remercie allah le tout puissant de m'avoir donné la force d'achever ce travail de fin de d'étude.*

*Je tiens a remercier sincerement Mme benmansour qui en tant qu'encadrant de mémoire, s'est toujours montrée à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'elle est bien voulu me consacrer afin de mener mon travail à bon part.*

*De même je remercie Mr. Maatallah et Mme.Halfaoui d'avoir voulu faire partie du jury.*

*Par ailleurs, je voudrais remercier l'ensemble du personnel de la société SOGESI pour leur accueil. Un grand merci tout particulier à Mr.Brahmi.*

*Enfin je remercie tous ceux qui de près ou de loin ont contribué à la réalisation de ce travail.*

---

Dédicace

---

*Je dédie ce travail à mes chers parents jamais je ne saurais m'exprimer quant aux sacrifices et aux dévouements que vous avez consacré à mon éducation et mes études,*

*A mes chères sœurs Hadjira et Zohra, à mes chers frères Badreddine et Mokhtar, Merci pour vos encouragements et votre aide sincère,*

*A mes chers Beaux-frères, un grand merci pour tout ce que vous avez fait pour moi,*

*A mes chers neveux et nièces, ma source de joie que dieu vous protège,*

*A ma chère Amie et Sœur Fatima, les mots ne suffisent pas pour exprimer l'amour que je port pour toi. Je voulais te dire merci pour ton aide, ta gentillesse sans égal et tes encouragements, et que malgré la distance, tu étais toujours à mes cotes.*

*A mes chères amies Bouchra et Sara, A ma cousine Meryem Merci pour nos bons moments passés ensemble, je vous dédie ce travail en vous souhaitons un meilleur avenir plein de meilleures réalisations,*

*A tous mes proches,*

*Je vous dédie ce travail et je vous remercie.*

*Sarra.*

## Table des matières

<i>Liste des Figures</i> .....	4
<i>Liste des abréviations</i> .....	6
<i>Introduction Générale</i> .....	7
<i>Chapitre 1 : Les ERP_Et Odoo</i> .....	9
<i>I. Introduction</i> .....	9
<i>II. Présentation des ERP</i> .....	9
II.1. Définition d'un ERP :	9
II.2. Fonctionnalités .....	9
II.3. Avantages des ERP .....	10
II.4. Le Choix de l'ERP .....	10
II.5. Types d'ERP .....	11
II.5.1. ERP Open Source .....	11
II.5.2. ERP Propriétaires .....	11
II.5.3. ERP en mode SaaS .....	12
<i>III Présentation d'Odoo</i> .....	12
III.1. Odoo .....	12
III.2. Différentes versions d'Odoo :	13
III.2.1. Online Odoo :	13
III.2.2. Open Community :	13
III.2.3. Odoo entreprise :	14
III.3. Les différentes Licence d'Odoo .....	14
III.4. Architecture logicielle .....	14
III.4.1. Architecture technique :	14
III.4.2 Architecture modulaire .....	15
III.5. Structure d'un module Odoo .....	16
<i>IV. Gestion des ressources humaines</i> .....	16
IV .1. Gestion des Dépenses .....	17
IV.1.1. Définition d'une note de frais .....	17
IV.1.2. Gestion de Dépense dans Odoo .....	17
<i>V. Conclusion</i> .....	18
<i>Chapitre2 : Analyse et Conception</i> .....	19
<i>I. Introduction</i> .....	19

<i>II. Organisme d'accueil</i> .....	19
<i>III. Spécification du besoin</i> .....	20
III.1. Les acteurs du système.....	20
III.2. Exigences fonctionnelles.....	21
<i>IV. Diagramme de Gantt</i> .....	21
<i>V. Conception détaillée du projet</i> .....	22
V.1. Diagrammes de cas d'utilisation.....	22
V.2. Diagrammes de Séquence.....	25
V.2.1. Le Diagramme de séquence des cas d'utilisation de l'employé.....	26
V.2.2. Le diagramme de séquence d'un cas d'utilisation de l'administrateur.....	32
V.3. Diagramme de classe.....	34
<i>VI. Conclusion</i> .....	35
<i>Chapitre 3 : Réalisation de L'application mobile « Expense Manager »</i> .....	36
<i>I. Introduction</i> .....	36
<i>II. Un service web</i> .....	36
II.1. Avantages.....	37
II.2. Fonctionnement.....	37
II.3. Architecture.....	38
<i>II.4. SOAP vs REST</i> .....	39
<i>III. Schéma du protocole Xml-RPC</i> .....	40
<i>IV. Outils utilisés</i> .....	41
<i>V. Langages</i> .....	43
<i>VI. La Reconnaissance optique de caractères (OCR)</i> .....	44
VI.1. Définition.....	44
VI.2. Différence entre OCR et extraction des données.....	44
VI.3. Différents aspects de l'OCR.....	45
<i>VII. Reconnaissance en-ligne et hors-ligne</i> .....	45
<i>VIII. Problèmes liés à l'OCR :</i> .....	46
<i>IX. Adaptation du module SDP</i> .....	46
IX.1. Présentation du module SDP.....	46
IX.2. Gestion des droits d'Access.....	49
<i>X. Présentation de l'application</i> .....	51

<i>XI. Conclusion</i> .....	55
<i>Conclusion Générale</i> .....	56
<i>Références bibliographiques</i> .....	57

## Liste des Figures

Figure 1 : Composition d'un ERP.....	9
Figure 2: Position d'Odoo avec ses concurrents. ....	13
Figure 3:Architecture Client-serveur D'Odoo.[4].....	15
Figure 4: Architecture Modulaire.....	15
Figure 5: Structure d'un module Odoo. ....	16
Figure 6: Les Fonctionnalités assurées par la GRH. ....	17
Figure 7:Les clients de SOGESI. ....	19
Figure 8:Diagramme de Gantt.....	21
Figure 9:Cas d'utilisation de l'employé. ....	22
Figure 10:Cas d'utilisation du manager. ....	23
Figure 11:Cas d'utilisation du comptable. ....	24
Figure 12:Diagramme de Séquence Générer dépenses. ....	25
Figure 13:Diagramme de séquence Soumettre dépense au responsable. ....	28
Figure 14: Diagramme de séquence Suivre le retour des dépenses. ....	30
Figure 15: Diagramme de séquence Gérer les dépenses. ....	32
Figure 16: Diagramme de classe. ....	34
Figure 17:Schéma de fonctionnement d'un web service. ....	37
Figure 18:Schéma du protocole RPC ....	40
Figure 19:Logo d'android. ....	40
Figure 20:Logo de PostgreSQL. ....	40
Figure 21:Logo Sublime Text. ....	41
Figure 22:Logo d'entreprise architect. ....	41
Figure 23:Ms Project.....	41
Figure 24:Logo du langage Python. ....	42
Figure 25:Logo du langage XML.....	42
Figure 26:Logo de Java Android ....	42
Figure 27:Création d'une dépense.....	45
Figure 28:Création d'un rapport (notes de frais).....	47
Figure 30:Exemple d'une catégorie créée pour notre application. ....	47
Figure 31:Activation de la messagerie interne et définition de l'alias domaine ....	49
Figure 32:Configuration de serveur de messagerie entrant.....	50
Figure 33:Page d'authentification ....	51

Figure 34: Ecran splash.....	51
Figure 35: Accueil liste dépenses.....	51
Figure 36: Menu de navigation. ....	51
Figure 37: Afficher les détails d'une dépense et soumettre au responsable.....	52
Figure 38: Creation manuelle.....	53
Figure 39: Creation automatique.....	53

## Liste des abréviations

Abréviations	Désignations
<b>BSD</b>	Berkeley Software Distribution.
<b>CRM</b>	Customer relationship management.
<b>ERP</b>	Enterprise Resource Planning
<b>ETI</b>	Entreprise de taille intermédiaire.
<b>HTTP</b>	HyperText Transfert Protocol.
<b>IDE</b>	Integrated Development Environnement
<b>JSON</b>	JavaScript Object Notation.
<b>JSP</b>	Java Server Pages.
<b>MVC</b>	Model Controller View.
<b>NDF</b>	Note de frais
<b>OCR</b>	Optical Character Recognition
<b>ORM</b>	Object Relational Mapping.
<b>PGI</b>	Progiciel de gestion intégré.
<b>PME</b>	Petites et Moyennes Entreprises.
<b>PMI</b>	Petite ou moyenne entreprise industrielle.
<b>RPC</b>	Remote procedure call.
<b>SAOP</b>	Simple object Access Protocol.
<b>SDK</b>	Software Development Kit.
<b>SGBD</b>	Système de Gestion de Base de Données.
<b>SSII</b>	Société de services et d'ingénierie en informatique
<b>TPE</b>	Très Petites Entreprises.
<b>UML</b>	Unified Modeling Language.
<b>URL</b>	Uniform Resource Locator.
<b>WDSL</b>	Web Services Description Language .
<b>XML</b>	Extensible Markup Language.

---

## Introduction Générale

Avec la prolifération de la technologie, le monde entier cherche à l'utiliser pour répondre à ses besoins professionnels. Cependant, dans le marché du travail, les entreprises ont souvent besoin de plus d'un logiciel pour faciliter leurs activités et rationaliser leurs outils de gestion. Une installation complète combinant tous les logiciels nécessaires sera facile à utiliser et à installer, et de là est venue la notion des ERP (Entreprise Ressource Planning) qui est devenue l'épine dorsale du système d'information des grandes entreprises et des PME. C'est une solution logicielle que toute entreprise souhaitant optimiser et synchroniser la gestion de toutes ses activités devrait mettre en place.

Le marché du travail étant en pleine expansion et soumis à une forte concurrence, les entreprises doivent se concentrer sur leurs tâches principales afin de rivaliser avec leurs concurrents et de les dépasser, d'où vient la nécessité de trouver de nouvelles technologies qui permettent de gagner beaucoup de temps et de les distraire de leurs tâches secondaires.

Parmi ces tâches, nous avons la gestion de note de frais qui s'avère être un besoin important pour les entreprises afin de libérer l'employé des tâches secondaires.

C'est une gestion qui se fait toujours manuellement, ce qui exerce une forte pression sur l'employé.

Notre projet a pour objectif de mettre à la disposition du personnel d'une entreprise une application mobile pour gérer les notes de frais, cette application doit donner la possibilité au personnel de faire du ZERO SAISIE, c'est-à-dire une connaissance intelligente des justificatifs de dépenses (OCR).

Le but de l'application mobile est de gérer les notes de frais et une synchronisation avec un module Odoo existant : le SDP (suivi des dépenses) à adapter. La gestion est faite d'une façon automatique, l'employé capture le reçu et en un seul clique les données sont placées dans les champs qui leurs correspondent, en utilisant la technique ROC (reconnaissance optique de caractères) connue beaucoup plus en anglais sous l'acronyme OCR (Optical character recognition) pour récupérer les données nécessaires à partir de la facture.

Une fois les justificatifs pris en photos, les éléments pertinents sont récupérés automatiquement dans l'application centrale. Ce qui aboutit donc à la création d'une note de frais.

- 1- ZERO PAPIER : les factures sont archivées pendant toute la durée légale.
- 2- Traitement des factures qui arrivent par mail : elles doivent apparaître déjà saisies.
- 3- Gestion des approbations
- 4- Gestion des frais en devises.
- 5- Gérer les frais kilométriques.

Notre travail consiste à apporter (réaliser) une solution à ce besoin réel exprimé par l'entreprise SOGESI (Société de Gestion des Solutions Informatiques) dans le cadre de la préparation de notre projet de fin d'étude de Master informatique, option génie logiciel à l'université de Tlemcen. Le stage s'est déroulé au niveau de l'entreprise du 10 février jusqu'au 18 juin 2019.

Notre manuscrit est constitué de trois chapitres :

- Le premier présente quelques généralités sur les ERP.
- Le deuxième est entièrement consacré à la conception de notre application baptisée « Expense Manager ».
- Le troisième détaille la réalisation de notre application et propose des aperçus de celle-ci.
- Pour finir ce mémoire, une conclusion générale résume le travail réalisé tout en soulignant quelques perspectives futures.

# Chapitre 1 : Les ERP

---

Et Odoo

---

## I. Introduction

Les entreprises disposent de plusieurs services, chacun disposant de son propre système d'information (SI). Ces services ont souvent besoin les uns des autres, et cela nécessite par conséquent de créer un lien entre eux. Cette démarche fait perdre beaucoup de temps aux employés mais aussi elle cause plusieurs erreurs, ce qui entraîne des incohérences entre les différents systèmes. Après l'apparition des ERP, ces problèmes ont été traités, le fait d'adopter cette solution et de l'installer dans les entreprises permet de rassembler tous les services dans un seul logiciel et donc plus d'erreur d'incohérence à cause de l'unicité du SI.

Dans ce chapitre, nous allons commencer par présenter la notion d'ERP, ensuite nous présentons l'ERP qui fait l'objet de notre projet : le progiciel Odoo et à la fin, une brève vue sur la gestion des ressources humaines (GRH).

## II. Présentation des ERP

### II.1. Définition d'un ERP :

Le terme ERP signifie (Entreprise Ressource Planning), traduit en français sous l'acronyme PGI (progiciel de gestion intégré) [1].

Un ERP est un ensemble d'applications relié à une base de données unique. Cela implique que ce système d'information gère l'ensemble des processus opérationnels d'une entreprise en intégrant plusieurs fonctions de gestion.

L'ERP est un outil central pour l'entreprise qui permet d'optimiser ses performances en optimisant le coût et les ressources.

Le point fort des ERP est le fait que ce sont des logiciels évolutifs, ils s'adaptent aux changements de l'entreprise. De nombreux services complémentaires peuvent compléter une solution ERP classique.

### II.2. Fonctionnalités

L'ERP est subdivisé en modules et chaque module couvre un domaine de gestion de l'entreprise. Parmi ces fonctionnalités on a :

- ❖ La gestion comptable et financière (ERP de gestion comptable, ERP de gestion de facturation),
- ❖ La gestion du stock.
- ❖ La gestion du personnel et de la paie.

- ❖ La gestion des fournisseurs.
- ❖ La gestion de la vente.
- ❖ La gestion de l'e-commerce
- ❖ La gestion de la production.

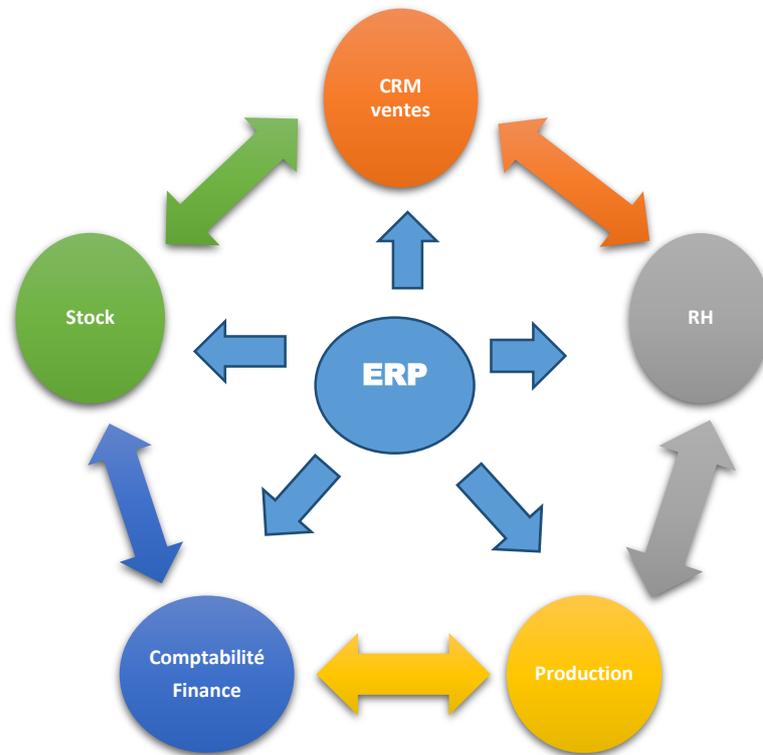


Figure 1 : Composition d'un ERP.

### II.3. Avantages des ERP

Les ERP présentent quelques points forts [2] :

- Une vision d'ensemble sur l'information qui circule au sein de l'entreprise.
- Un gain en temps et en efficacité.
- Le gain de productivité généré par l'utilisation de l'ERP se traduit par une augmentation du chiffre d'affaires.
- Le risque d'erreur est limité.

### II.4. Le Choix de l'ERP

Choisir un ERP qui répond aux attentes d'une entreprise ne peut être facile mais il y a des critères qui peuvent aider à déterminer l'ERP selon le besoin. Parmi ces critères, on trouve :

- Fixer le domaine d'activité de l'entreprise :

Certains ERP peuvent s'adapter à un spectre d'industrie très large et d'autres sont orientés sur des métiers spécifiques. De préférence, l'entreprise doit avoir un logiciel dans son secteur d'activité qui est prêt à l'emploi.

- Couverture fonctionnelle et agilité :

Le système doit offrir un environnement favorable aux prises de décisions et être souple pour s'adapter à l'entreprise et ses attentes préalablement formulées et ne doit pas la forcer à changer ses processus.

- Coût :

Un ERP est un investissement à long terme, il ne faut pas penser juste aux coûts initiaux mais il faut aussi prendre en considération les coûts à moyen et long terme pour comparer les offres.

- Choisir le bon partenaire :

Intégrer un ERP dans l'entreprise est un engagement pour des années, et du coup, avoir un bon partenaire garantit un bon accompagnement durant toute la période du projet.

## II.5. Types d'ERP

On distingue trois types d'ERP : les ERP open source qui sont "gratuits", les ERP propriétaires, édités par des sociétés, ce qui implique l'achat d'une licence et les ERP en mode SaaS :

### II.5.1. ERP Open Source

Un ERP open source est un logiciel libre, moins cher car il ne nécessite pas l'acquisition d'une licence et donc l'entreprise n'aura aucun engagement. Il est généralement adopté par les PME [21]. Les Top 5 de quelques d'ERP open source dans le marché [3] :

1. Odoo avec 2 000 000 utilisateurs.
2. Dolibarr.
3. Axelor.
4. Openbravo.
5. Compiere avec 9000 utilisateurs.

### II.5.2. ERP Propriétaires

Un ERP propriétaire est un logiciel performant, plus cher que l'ERP open source et plus performant. L'acquisition d'une licence inclut l'accompagnement dans le projet et un service après-vente [4].

Les Top 10 des ERP propriétaires définies par panorama consulting solutions [5] :

1. Epicor.
2. Infor.
3. SAP.
4. IFS.
5. Oracle.
6. NetSuite.
7. Microsoft.
8. Sage.
9. Syspro
10. IQMS.

Les modules et les fonctionnalités d'un ERP open source et d'un ERP propriétaires sont identiques.

#### II.5.3. ERP en mode SaaS :

L'ERP en mode SaaS est un ERP commercialisé sous forme d'un service et non pas d'un logiciel. Contrairement aux ERP propriétaires, le SaaS n'a pas une licence mais un abonnement [4].

### III Présentation d'Odoo

#### III.1. Odoo

 Odoo, anciennement connu sous le nom de TinyERP puis d'OpenERP, est un progiciel open source de gestion intégré, fondé en juillet 2004. Odoo est une solution modulaire dans laquelle chacun de ces modules se rassemble pour créer un système avec des flux automatisés et une centralisation de données. Son ergonomie exceptionnelle au travers de toutes ses applications, a rendu son utilisation hyper simple [6].

Grace à sa large communauté, il est entretenu activement par de nombreux développeurs ; ce qui permet d'adapter ce modèle rapidement aux besoins des clients et leurs fournir de nouvelles applications innovantes [6] ; et cela est témoigné par Marc Peeters le manager général du département vending quand il a dit :

« Nous avons remplacé 14 applications différentes et augmenté nos revenus de 10% du jour au lendemain ». *Marc Peeters, Manager Général du département Vending* [6].

*Fabien pinckaers, fondateur Odoo*, a dit : « Notre mission est de transformer la manière dont les entreprises gèrent leurs affaires, nous rendons l'IT (Technologie de l'Information) simple et totalement intégrée. Nous aidons les organisations à grandir » [6]. C'est pour cette raison qu'il était élu comme le meilleur logiciel de gestion pour diriger une entreprise.

La figure ci-dessous montre la position d'Odoo par rapport aux autres ERP open source et propriétaire selon la performance et la convivialité d'utilisation. On remarque qu'Odoo est le seul ERP qui vérifie les deux conditions :

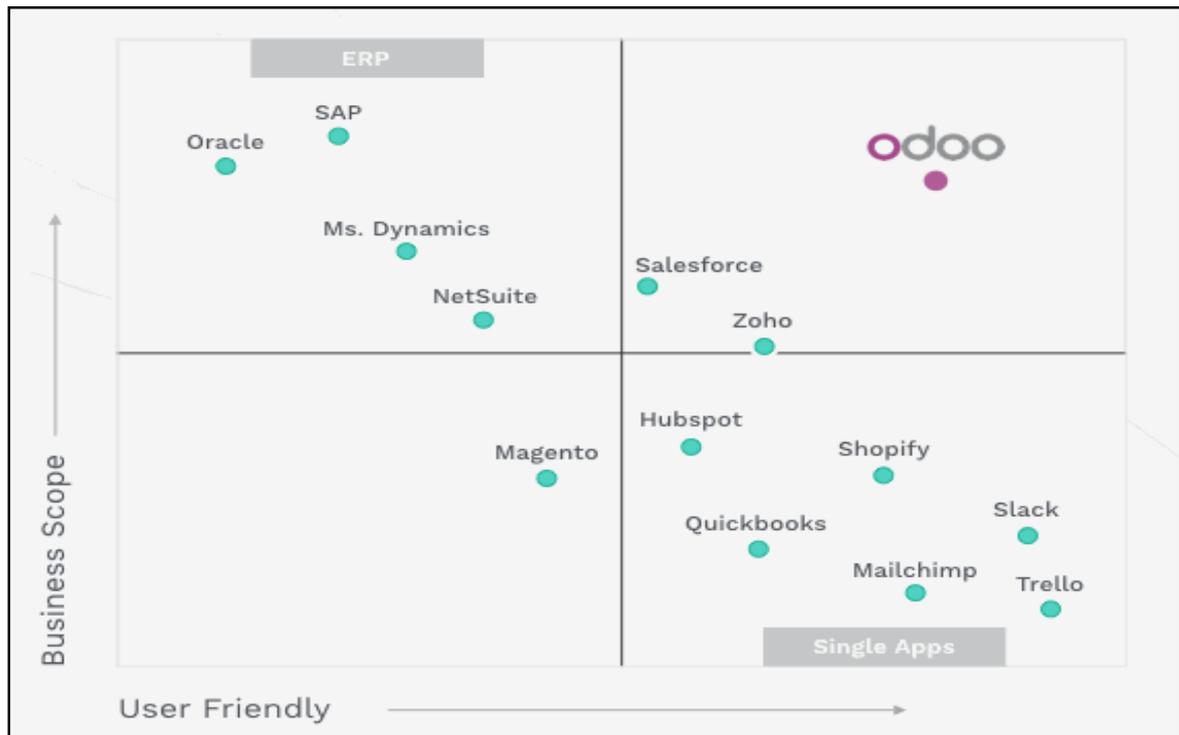


Figure 2: Position d'Odoo avec ses concurrents [6] en juin 2019.

### III.2. Différentes versions d'Odoo :

Odoo vient sous 3 formes :

#### III.2.1. Online Odoo :

La version SaaS offrant une solution non conventionnelle et rapide. Une implémentation idéale pour les PME de moins de 50 utilisateurs [7].

#### III.2.2. Open Community :

C'est la version open source d'Odoo, disponible pour le téléchargement et elle est entièrement gratuite.

### III.2.3. Odoo entreprise :

Cette version est installée localement, payante et contrairement à l'online Odoo elle est recommandée pour les entreprises avec plus de 50 utilisateurs.

### III.3. Les différentes Licences d'Odoo

- ❖ De la première version d'Odoo jusqu'à la 6ème, le serveur et le client GTK+ étaient publiés sous la Licence Publique Générale GNU, et le client web sous la licence publique Open ERP (OEPL) basée sur la version 1.1 de la licence publique Mozilla (MPL) [8].
- ❖ A partir de la 7ème version jusqu'à la 8ème, ont été publiés sous la licence AGPL 3.0 [8].
- ❖ A partir de la version 9, l'édition appelée Odoo-Community est publiée sous la licence LGPLV3 [8].

### III.4. Architecture logicielle

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 [8] ;

#### III.4.1. Architecture technique :

L'architecture d'Odoo est construite autour de trois composants principaux qui communiquent entre eux par les protocoles XML-RPC et NET-RPC, on a [9] :

**1-serveur de base de données PostgreSQL** : il contient toutes les bases de données et chacune contenant toutes les données et les éléments de configuration du système Odoo.

**2-serveur d'applications Odoo** : contient toute la logique de l'entreprise. Le moteur ORM est dédiée à la communication et à l'interface avec la base de données PostgreSQL. Et la couche web qui permet les communications entre le serveur et un navigateur Web.

**3-Clients** : Les clients peuvent communiquer avec un serveur Odoo à l'aide de XML-RPC. Odoo développe deux clients différents : un client de bureau, écrit avec la boîte à outils graphique GTK + largement utilisée, et un client Web devant s'exécuter dans n'importe quel navigateur Web moderne.

Comme la logique d'Open ERP doit entièrement résider sur le serveur, le client est conceptuellement très simple. Il envoie une requête au serveur et affiche le résultat de différentes manières. Lors des actions de l'utilisateur, il enverra les données modifiées au serveur.

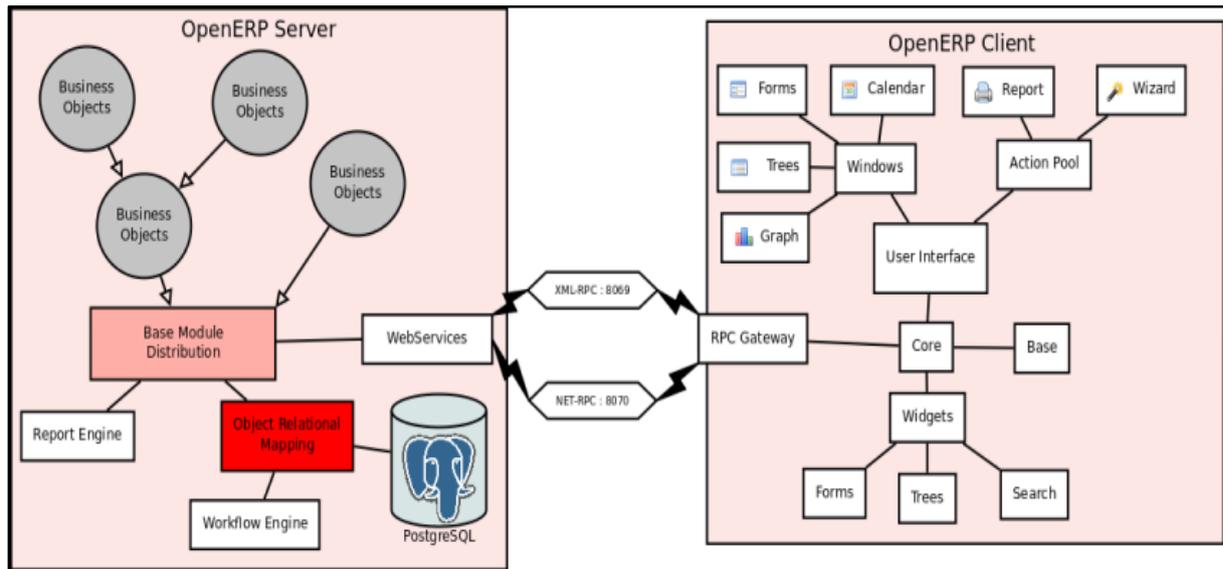


Figure 3: Architecture Client-serveur D'Odoo [9]

### III.4.2 Architecture modulaire

Le noyau d'Open ERP et ses modules sont écrits en Python. La fonctionnalité d'un module est exposée via XML-RPC ou NET-RPC selon de la configuration du serveur. Les modules utilisent généralement l'ORM d'Open ERP pour conserver leurs données dans la base de données relationnelle mentionnée précédemment (PostgreSQL). Les modules peuvent insérer des données dans la base de données lors de l'installation en fournissant des fichiers XML, CSV ou YML [9].

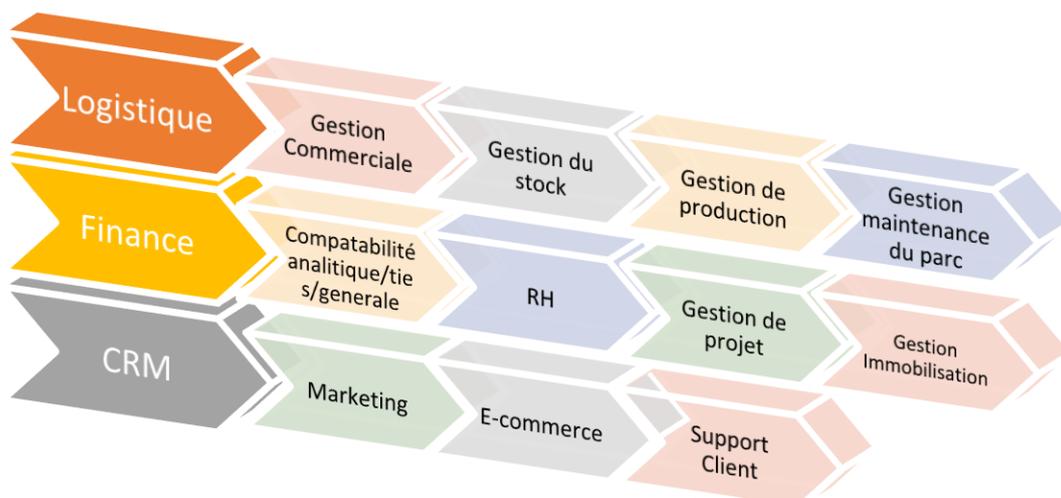


Figure 4: Architecture Modulaire

### III.5. Structure d'un module Odoo

Un module Odoo se compose souvent de 7 dossiers et obligatoirement de 2 fichiers python. Chaque dossier est destiné pour une fonction définie.

Le rôle de chaque composant est défini brièvement ci-dessous :

1. **Security** : le dossier ou sont déclarés les droits d'accès, le fichier porte un nom fixe et prédéfini et vient sous l'extension CSV.
2. **Models** : le dossier ou sont déclarés les objets, le code du fichier est écrit en python. Ce dossier doit contenir obligatoirement un fichier init.
3. **Views** : c'est l'endroit où sont déclarées les vues, ses fichiers sont écrits en XML.
4. **Wizard** : le dossier ou sont créés les alertes.
5. **Static** : dans ce dossier on peut mettre des images.
6. **Init** : init est un fichier python, dans ce fichier on importe tous les objets créés dans le dossier models.
7. **Manifest** : ce fichier, contient le nom du module, sa description, sa catégorie, son auteur, ses dépendances si elles existent, et les déclarations des fichiers XML et CSV.

La figure suivante montre la composition d'un module Odoo :

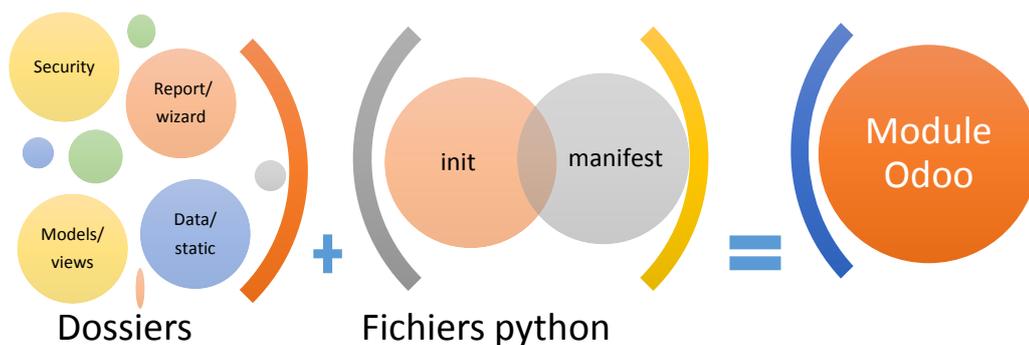


Figure 5: Structure d'un module Odoo

## IV. Gestion des ressources humaines

La gestion des ressources humaines (GRH) ou communément appelée la gestion du personnel, est l'un des services que Odoo offre. La GRH est une fonction transversale dont chaque entreprise a besoin pour centraliser les informations sur ses ressources humaines [10].

Ces ressources humaines sont l'ensemble de salariés de différents types (gestionnaires, développeurs...etc.)

La GRH englobe toutes les gestions nécessaires pour gérer le personnel de l'entreprise, le ministère de l'éducation du Portugal a témoigné à propos de cela en disant que [10] :

« Grâce à Odoo, nous avons unifiés toutes nos applications RH front-office et back-office pour 300 000 utilisateurs de façon rentable. » *Ministère de l'Éducation, gouvernement du Portugal*



Figure 6: Les fonctionnalités assurées par la GRH

Dans notre projet, on s'intéresse à la gestion des dépenses et plus particulièrement aux notes de frais.

## IV .1. Gestion des Dépenses

### IV.1.1. Définition d'une note de frais

La note de frais est un document permettant à un salarié de **se faire rembourser les frais professionnels engagés pour le compte de la société**. Il peut s'agir de frais de repas, de frais de déplacement, de frais de logement, ou tout autre frais engagé dans l'intérêt de l'entreprise.

La note de frais prend le plus souvent la forme d'une fiche remplie par le salarié et remise à son entreprise [11].

### IV.1.2. Gestion de Dépense dans Odoo

Odoo offre une gestion de dépense de sorte qu'il évite The Pain Points, c'est-à-dire qu'il évite à l'employé de perdre du temps avec une tâche administrative fastidieuse sur papier.

Parmi les bienfaits que la gestion des notes de frais qu'Odoo apporte à ces utilisateurs :

1. Intégré (plusieurs applications) : le module de gestion de note de frais est relié avec plusieurs applications dans Odoo (CRM et ventes ...etc.).

2. Convivial (utilisation facile) : l'interface du module est simple ce qui facilite l'utilisation, ainsi que le principe de gestion.
3. Traçable : on peut suivre l'état de dépense en tout moment de la création jusqu'à la comptabilisation.
4. Gain de temps : l'intégrité de l'application et la convivialité ainsi que la traçabilité permettent au personnel de gagné du temps.
5. Zéro papier : les factures sont archivées dans Odoo donc il y aura plus la peine de les garder sous format papier.

## V. Conclusion

Nous avons présenté dans ce chapitre la notion d'ERP et particulièrement Odoo, qui est l'open ERP que nous avons utilisé dans notre application « Expense Manager » comme étant l'application centrale. Le chapitre suivant est consacré à la conception de notre dite application.

# Chapitre2 : Analyse et Conception

## I. Introduction

Pendant plusieurs décennies, le monde informatique a toujours rêvé d'un processus qui puisse garantir le développement efficace de logiciels de qualité, valable quel que soit la grandeur et la complexité du projet, et présentant de bonnes pratiques adaptées à la méthode en question, surtout que, de nos jours, les logiciels demandés sont de plus en plus imposants et exigeants qu'auparavant.

Le processus unifié semble être la solution idéale pour remédier à l'éternel problème des développeurs. En effet, il regroupe les activités à mener pour transformer les besoins d'un utilisateur en un système logiciel quel que soit la classe, la taille et le domaine d'application de ce système.

Le processus unifié utilise le langage UML (Unified Modeling Language). Ce langage de modélisation est une partie intégrante du processus unifié [12].

Dans ce chapitre, nous avons commencé par présenter l'organisme dans lequel nous avons effectué notre stage, pour attaquer par la suite la spécification des besoins ou nous avons définis les acteurs et les exigences fonctionnelles de notre système, ensuite, nous avons entamé la conception détaillée du projet ou nous avons utilisé les différents diagrammes UML pour la modélisation.

## II. Organisme d'accueil

SOGESI est une Société de Services en Ingénierie Informatique (SSII), fondée en 2013 à Tlemcen en Algérie, spécialisée dans les services informatiques et d'ingénierie de proximité, la gestion des applicatifs et des infrastructures ainsi que le conseil en technologies. Elle offre plusieurs services y compris des formations sur Odo, python...etc. Elle est spécialisée dans l'intégration de l'ERP Odo.

SOGESI accompagne ses clients en Algérie et en France lors de l'identification de leurs besoins, et assure même des formations pour les utilisateurs, et propose également un contrat de maintenance pour les rassurer [13].

- Services : Les services fournis par SOGESI sont :
  - ❖ GAP\_Analysis.

- ❖ Étude et critique de l'existant.
  - ❖ Rédaction de cahier des charges.
  - ❖ Proposition de solutions sur mesure.
  - ❖ Développement
  - ❖ Intégration
  - ❖ Formation des utilisateurs, support
  - ❖ Maintenance
- Les clients : la figure ci-dessous donne un aperçu de quelques clients (en Algérie et en France) de la société :



Figure 7: Les clients de SOGESI

### III. Spécification du besoin

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques et d'identifier les cas d'utilisation initiaux.

#### III.1. Les acteurs du système

Les acteurs qui sont en interaction avec notre système sont :

- ❖ **Employé** : C'est lui qui va créer la dépense, joindre la facture, soumettre la dépense au manager et suivre le retour des dépenses.

- ❖ **Manager** : est l'administrateur du système, celui qui approuve et refuse les dépenses en demandant plus d'informations en cas de refus.  
En plus de ces fonctionnalités, le manager peut aussi avoir des dépenses et donc il possède toutes les fonctionnalités de l'employé.
- ❖ **Comptable** : Le comptable a deux principales fonctionnalités dont la gestion des comptes de dépenses et le remboursement des employés.  
De même que le manager, le comptable peut aussi avoir des dépenses et par conséquent, il hérite des mêmes fonctionnalités que l'employé.

### III.2. Exigences fonctionnelles

Le système à réaliser doit satisfaire les exigences de la totalité des utilisateurs. Nous présentons dans ce qui suit tous les besoins fonctionnels :

- **Gestion des dépenses** : la saisie de dépenses doit être faite d'une manière automatique, le système permet à l'utilisateur de capturer le reçu, une fois capturé les données nécessaires sont récupérées automatiquement dans l'application centrale, ce qui aboutit donc à la création de la dépense.
- **Gestion des notes de frais (rapports)** : une dépense doit être affectée à un rapport, ce dernier peut contenir une ou plusieurs dépenses de la même catégorie.
- **Gestion des approbations** : le système doit permettre au manager de gérer les procédures de remboursement des frais professionnels.
- **Gestion des frais en devises** : prendre en compte la variété de monnaies.
- **Sauvegarde** : les factures (receipt/invoices) doivent être archivées pendant toute la durée légale de la note de frais.
- **Gestion des frais kilométriques** : le système doit calculer la distance automatiquement à partir d'un lieu de départ et un lieu d'arrivée, ainsi qu'il doit gérer le type de véhicule et la date de dépense.
- **Gestion des emails** : le système doit saisir automatiquement les dépenses qui arrivent par mail.

## IV. Diagramme de Gantt

Le diagramme de Gantt est un outil utilisé en gestion de projet et permettant de visualiser dans le temps les différentes tâches composant un projet [14].

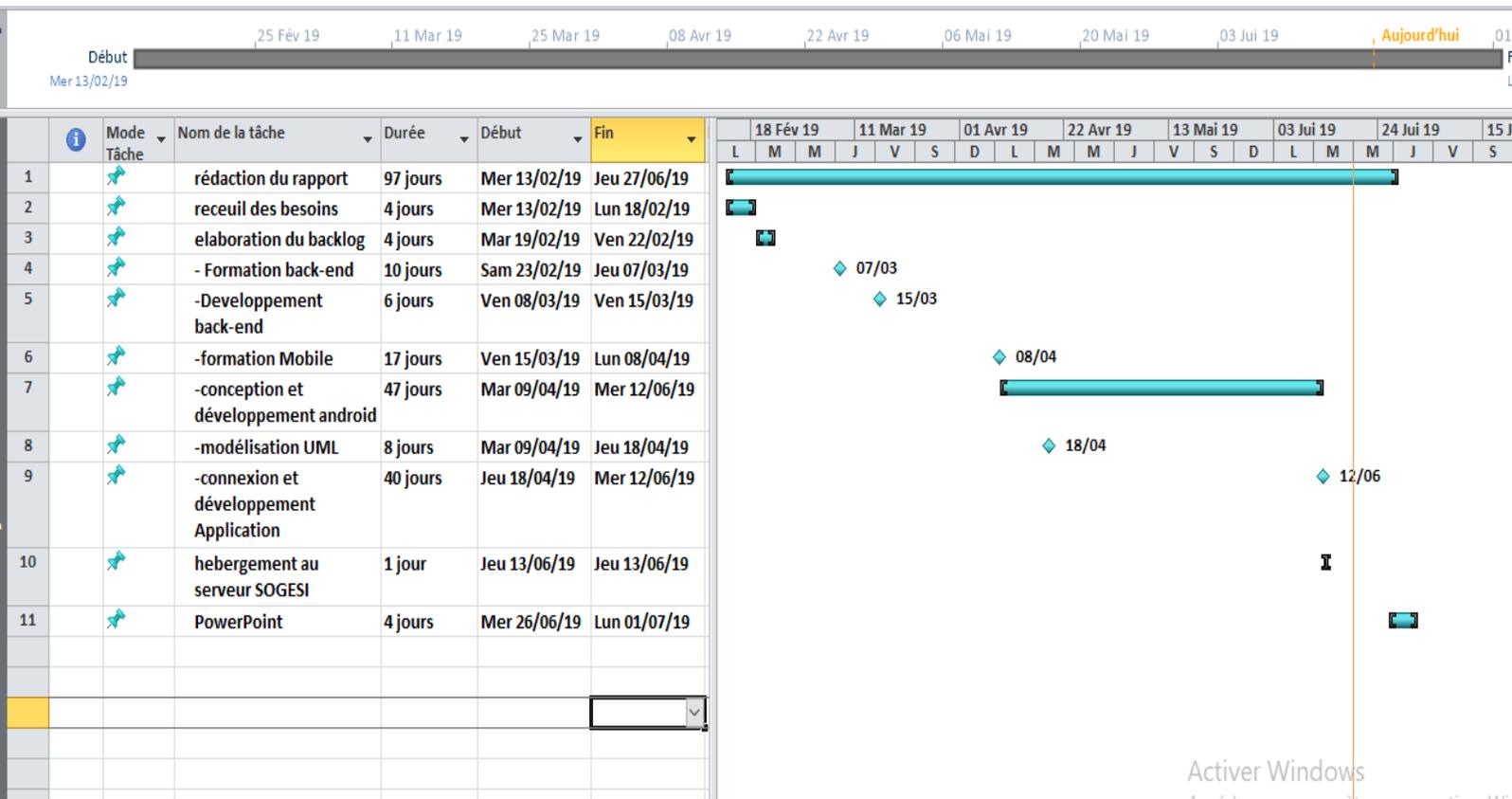


Figure 8: Diagramme de Gantt

## V. Conception détaillée du projet

### V.1. Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ce diagramme représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. C'est une unité significative de travail, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases) [15].

Nous allons présenter par la suite les diagrammes de cas d'utilisation par acteur :

- ❖ Le Diagramme de cas d'utilisation des tâches de l'employé.
- ❖ Le Diagramme de cas d'utilisation des tâches du Manager.
- ❖ Le Diagramme de cas d'utilisation des tâches du comptable.

➤ Diagramme de cas d'utilisation des tâches de l'employé

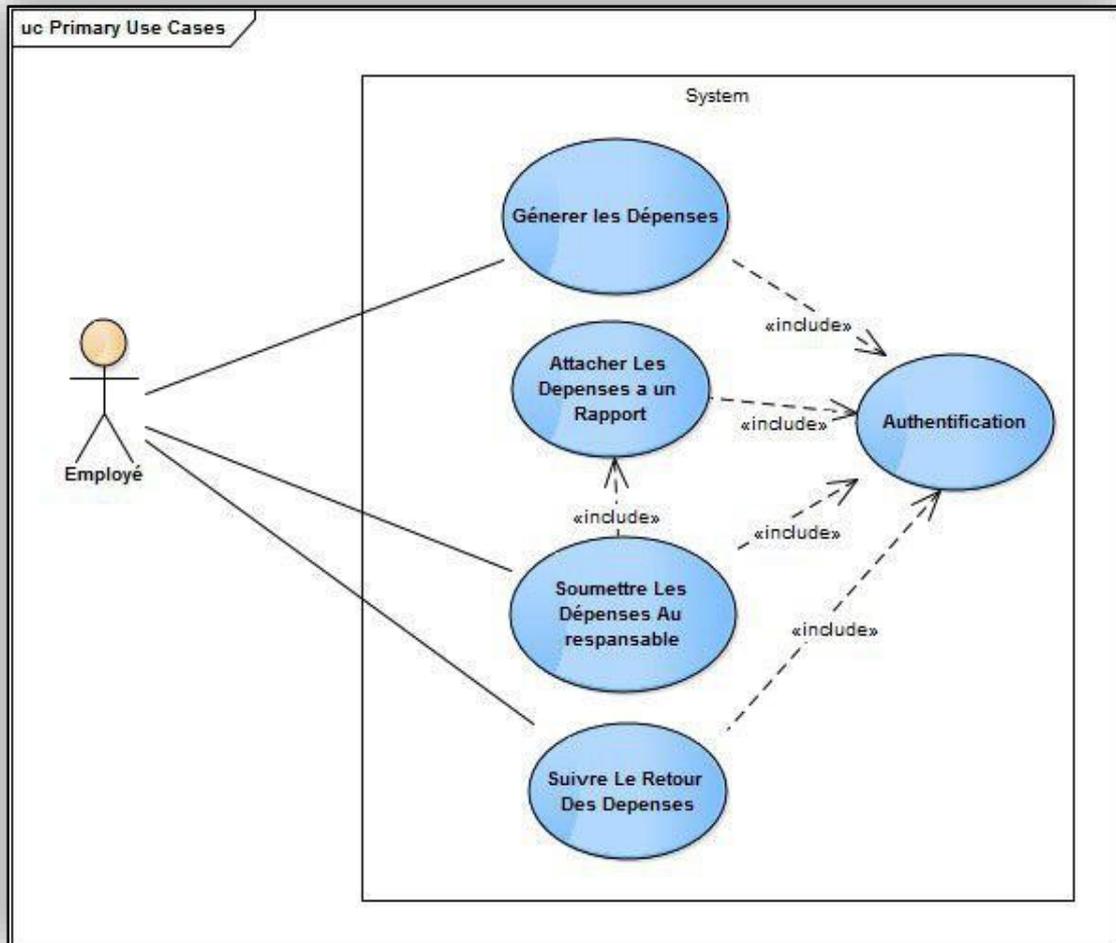


Figure 9: Diagramme de cas d'utilisation de l'employé

➤ Diagramme de cas d'utilisation des tâches du Manager

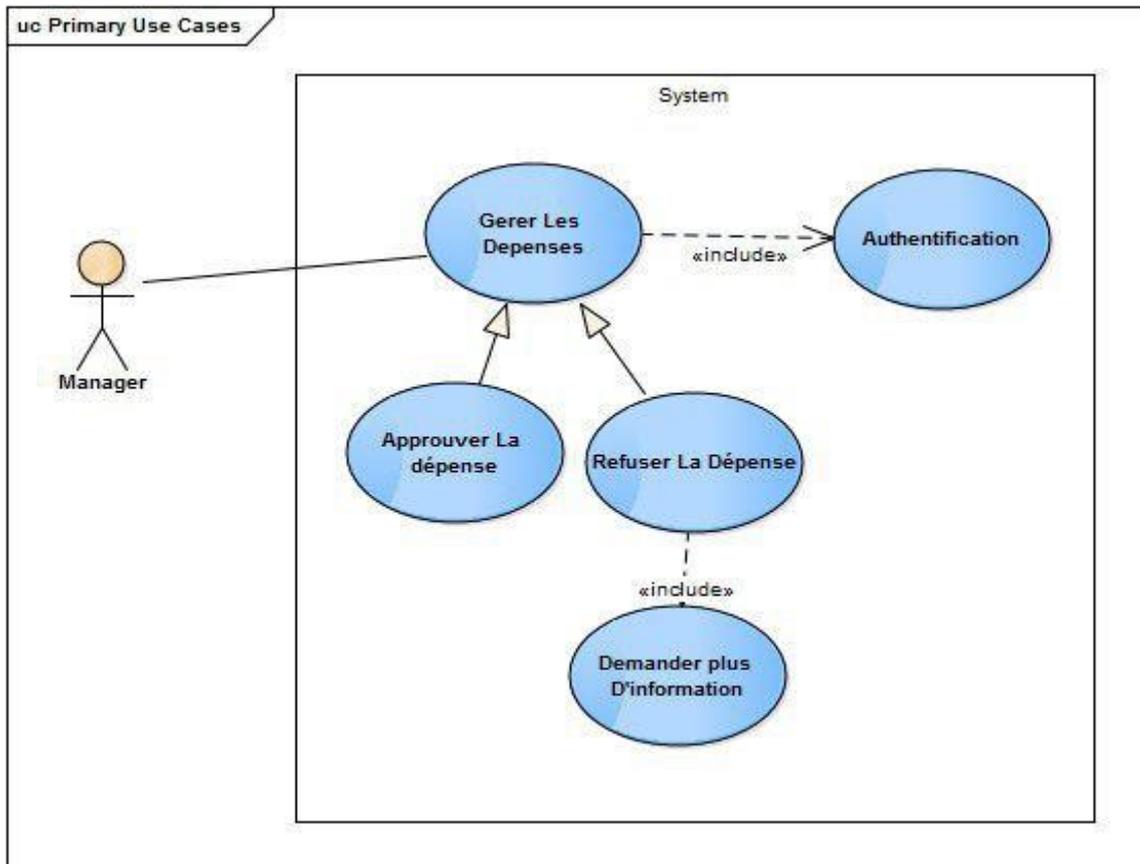
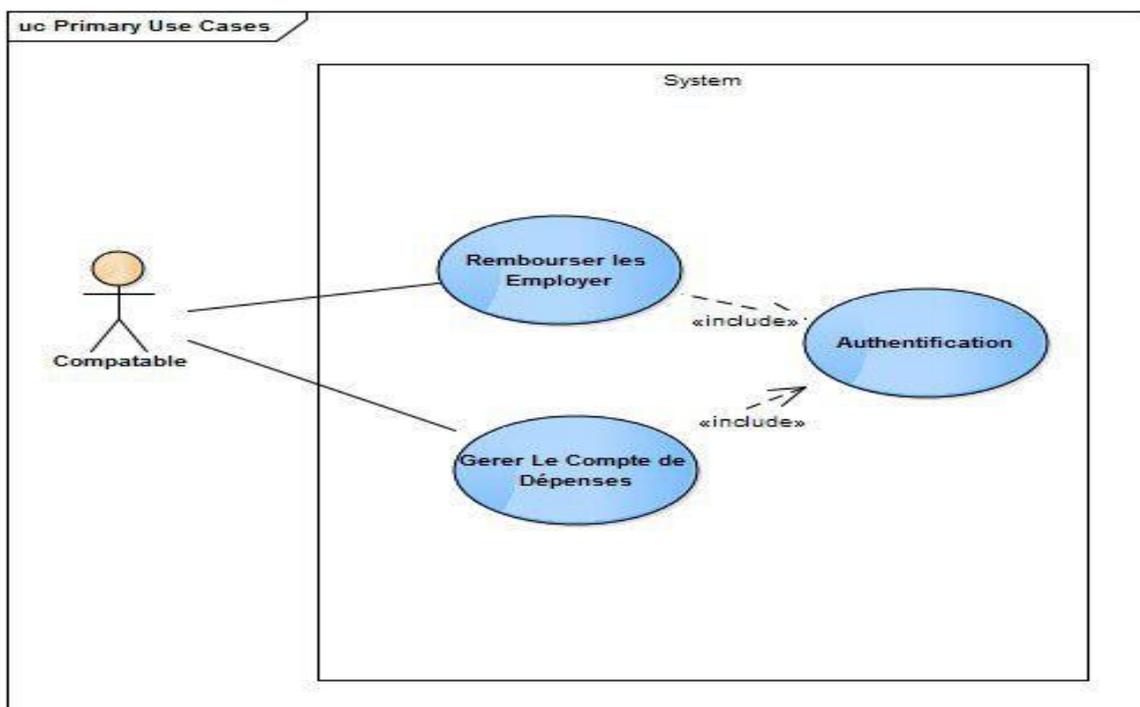


Figure 10 Diagramme de cas d'utilisation du manager

➤ Diagramme de cas d'utilisation des tâches du Comptable



## V.2. Diagrammes de Séquence

Les cas d'utilisation décrivent les interactions des acteurs avec le système que nous voulons spécifier et concevoir. Lors de ces interactions, les acteurs génèrent des messages qui affectent le SI et appellent généralement une réponse de celui-ci. Nous allons isoler ces messages et les représenter graphiquement sur des diagrammes de séquences UML [16].

Pour les événements propres à un cas d'utilisation, les DSS (diagrammes de séquence système) montrent non seulement les acteurs externes qui interagissent directement avec le système, mais également ce système (en tant que boîte noire) et les événements système déclenchés par les acteurs. L'ordre chronologique se déroule vers le bas et l'ordre des événements doit suivre la séquence décrite dans le cas d'utilisation [16].

Nous allons représenter le DSS d'un scénario représentatif de chacun des cas d'utilisation qui concernent l'application mobile, en commençant par ceux de l'employé ensuite du manager.

V.2.1. Le Diagramme de séquence des cas d'utilisation de l'employé

➤ Le cas d'utilisation « Générer dépenses »

a) Diagramme de séquence :

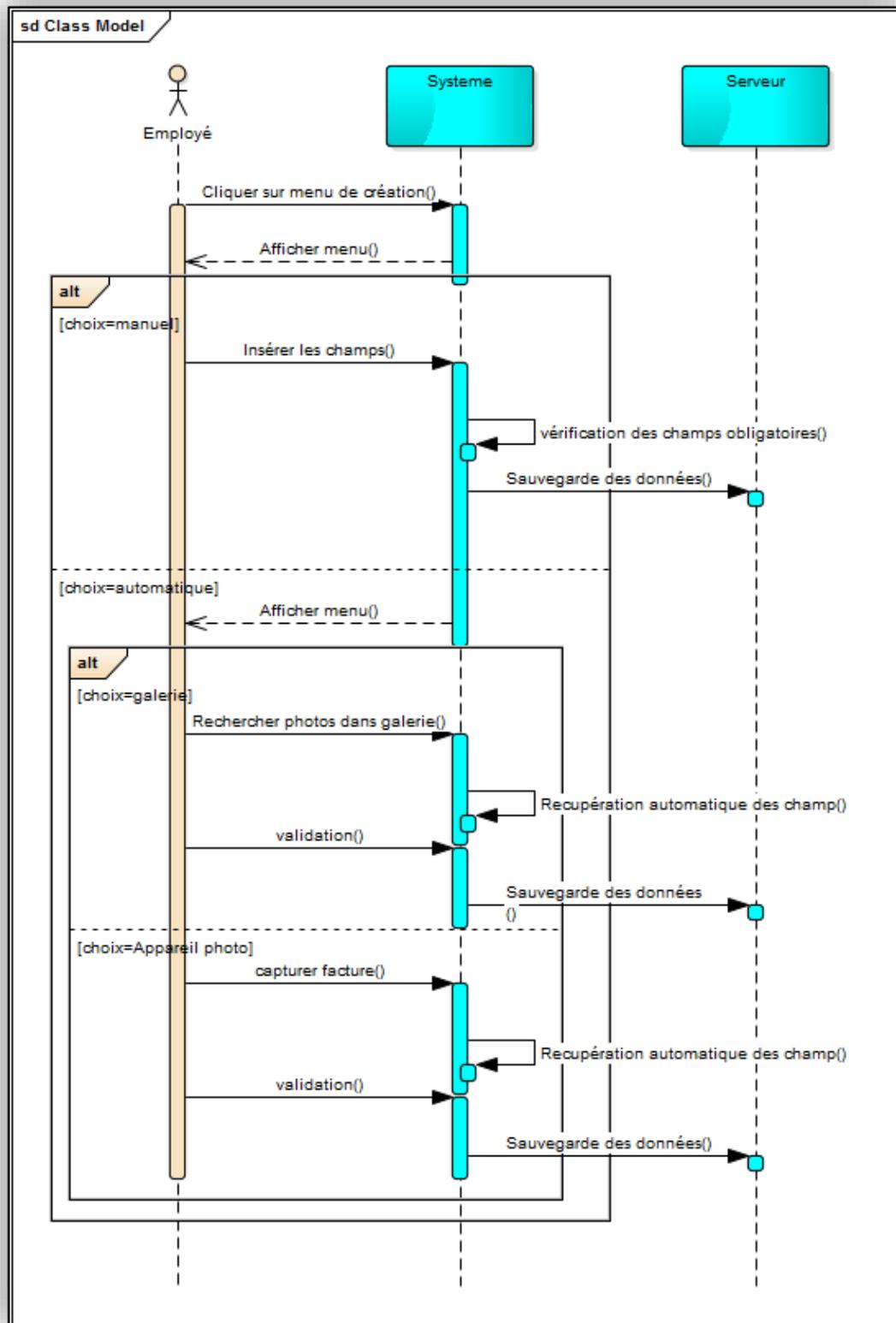


Figure 12: Diagramme de séquence "Generer depense"

b) Description textuelle

<b>Nom</b>	<b>Générer dépenses</b>
<b>Acteur</b>	Employé
<b>Description</b>	L'employé doit pouvoir créer et sauvegarder ses dépenses à partir d'une application mobile.
<b>Auteur</b>	Sarra Zennaki
<b>Date (s)</b>	01/04/2019
<b>Pré-conditions</b>	-L'employé doit être authentifié. -Avoir l'accès au serveur.
<b>Démarrage</b>	L'employé clique sur le menu de création.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur sélectionne un type de création.</li> <li>2. Si le choix est « création manuel », le système passe à la page de création.</li> <li>3. L'utilisateur insère les données nécessaires et attache la facture à la dépense.</li> <li>4. Le système vérifie les champs obligatoires.</li> <li>5. Le système sauvegarde les données dans le serveur.</li> </ol>
<b>Scénario D'exception</b>	<p>2-a. Le choix est « création automatique », le système propose à l'utilisateur deux manières de récupération de photo, soit de prendre une photo ou bien de parcourir une photo de la galerie.</p> <p>2-b. Le système récupère automatiquement les champs nécessaires.</p> <p>2-c. L'utilisateur termine le remplissage des champs qui reste et valide la dépense (retour à l'étape 4).</p> <p>4-a. Les champs ne sont pas valides ou tous remplis, le système notifie l'utilisateur (retour à l'étape 3).</p>

<b>Fin</b>	<ul style="list-style-type: none"> <li>• Scénario nominal : Sur la décision de l'utilisateur, après le point 5.</li> </ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>• La dépense a été enregistrée dans le serveur.</li> </ul>

➤ Le cas d'utilisation « soumettre dépense au responsable »

a) Diagramme de séquence

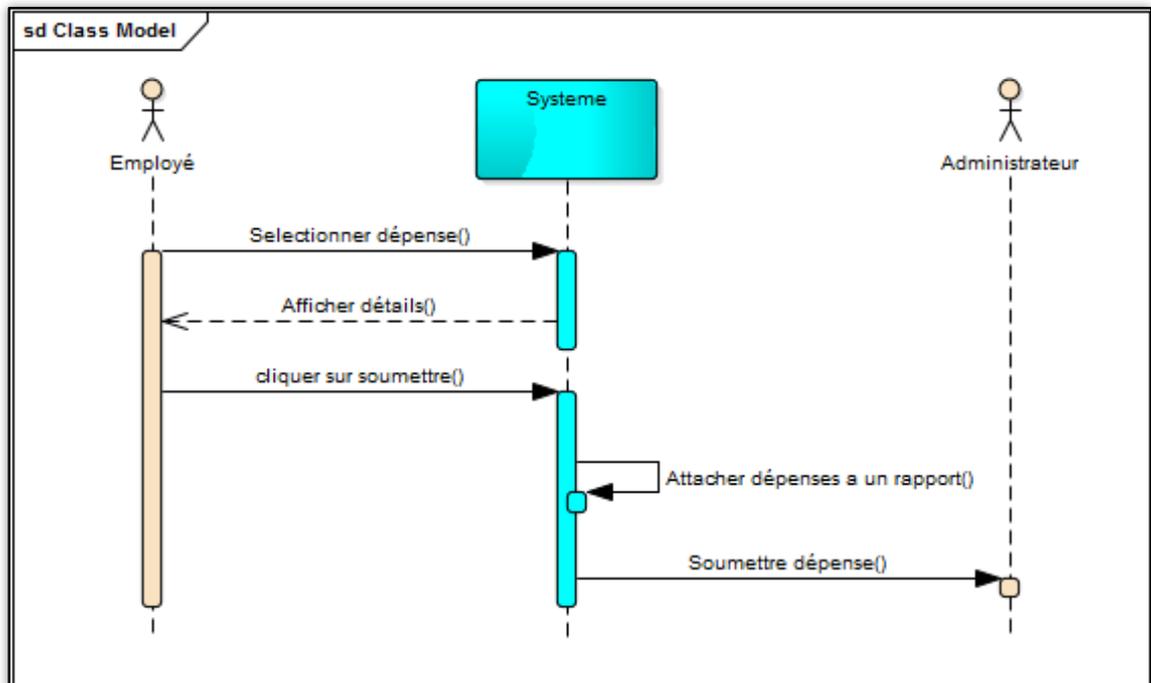


Figure 13:Diagramme de séquence "soumettre au responsable"

b) Description textuelle :

<b>Nom</b>	<b>Soumettre dépense au responsable</b>
<b>Acteur</b>	Employé
<b>Description</b>	L'employé doit envoyer la dépense pour qu'elle soit validé par le responsable.
<b>Auteur</b>	Sarra Zennaki
<b>Date (s)</b>	01/04/2019
<b>Pré-conditions</b>	-L'employé doit être authentifier. -Avoir l'accès au serveur.
<b>Démarrage</b>	L'employé a demandé la liste des dépenses.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système affiche à l'utilisateur la liste de ses dépenses.</li> <li>2. L'utilisateur sélectionne la dépense dont il veut la soumettre.</li> <li>3. Le système affiche une page contenant les détails de la dépense sélectionné et un bouton « soumettre ».</li> <li>4. L'utilisateur clique sur soumettre.</li> <li>5. Le système fait appel au cas d'utilisation interne « attacher dépenses à un rapport ».</li> <li>6. Le système soumet la dépense au manager.</li> </ol>
<b>Fin</b>	<ul style="list-style-type: none"> <li>• Scénario nominal : Sur la décision de l'utilisateur, après le point 6.</li> </ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>• La dépense a été soumise au responsable.</li> </ul>

➤ Le cas d'utilisation « suivre le retour des dépenses »

a) Diagramme de séquence :

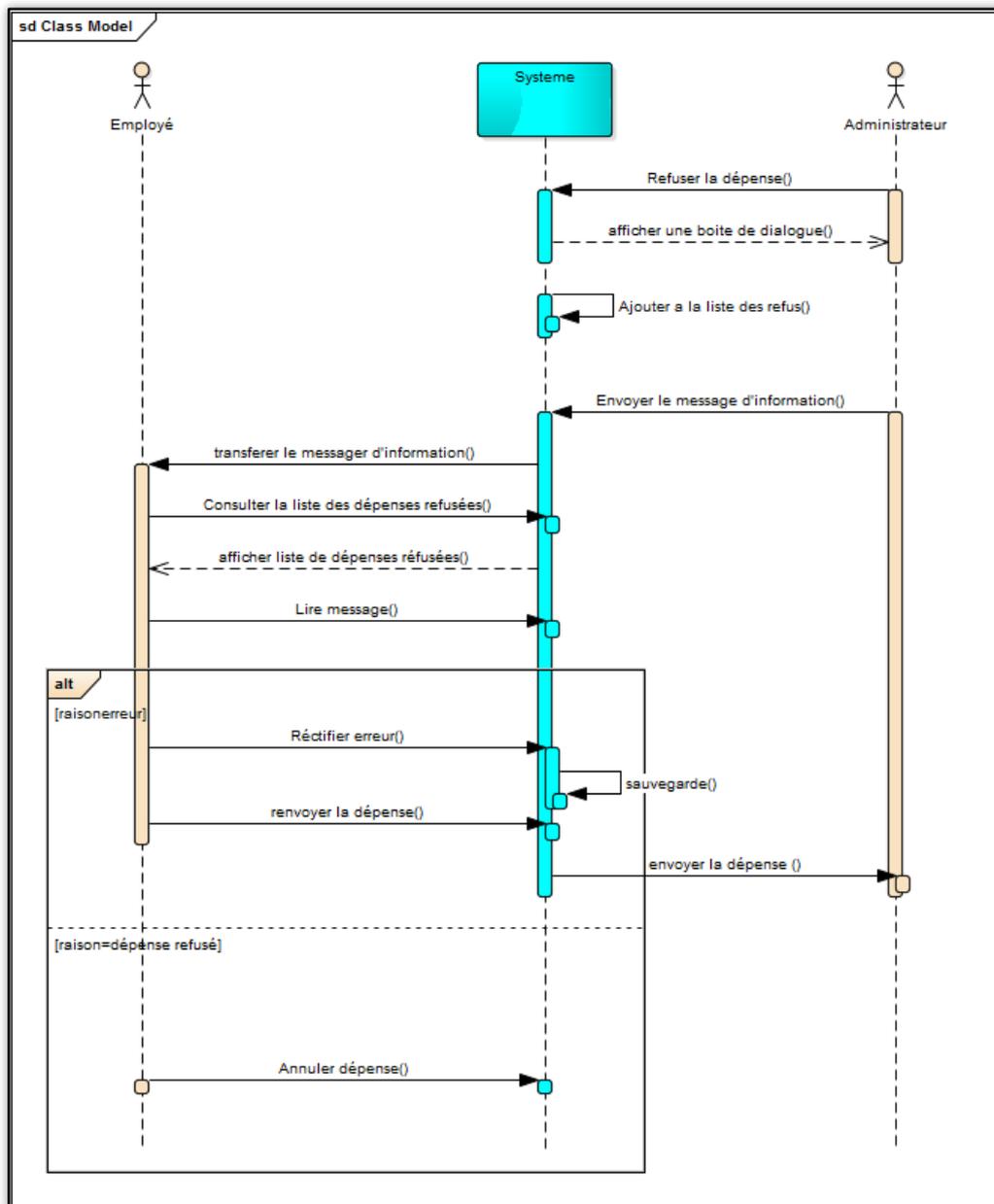


Figure 14: Diagramme de séquence « Suivre le retour des dépenses »

b) Description textuelle :

<b>Nom</b>	<b>Suivre le retour des dépenses</b>
<b>Acteur</b>	Employé
<b>Description</b>	L'employé doit avoir des informations sur la dépense qui a été soumise.
<b>Auteur</b>	Sarra Zennaki
<b>Date (s)</b>	01/04/2019
<b>Pré-conditions</b>	-L'employé doit être authentifié. -Avoir l'accès au serveur. -l'administrateur a déjà géré les dépenses.
<b>Démarrage</b>	L'employé demande la liste des dépenses refusées.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système affiche la liste des dépenses refusées.</li> <li>2. L'utilisateur consulte ses dépenses et lit la raison de refus.</li> <li>3. L'utilisateur rectifie l'erreur et soumet la dépense une autre fois au manager.</li> <li>4. Le système enregistre les changements apportés par l'utilisateur et soumet la dépense.</li> </ol>
<b>Scénario D'exception</b>	2-a l'administrateur n'accepte plus la dépense, l'utilisateur annule la dépense.
<b>Fin</b>	<ul style="list-style-type: none"> <li>• Scénario nominal : la dépense est soumise de nouveau, après le point 4.</li> <li>• Scénario d'exception : sur la décision de l'utilisateur, après le point 2-a.</li> </ul>

<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>• La dépense a été enregistrée dans le serveur.</li> <li>• L'administrateur a pris sa décision.</li> </ul>
------------------------	---

V.2.2. Le diagramme de séquence d'un cas d'utilisation de l'administrateur

➤ Le cas d'utilisation « Gérer les dépenses »

a) Diagramme de séquence :

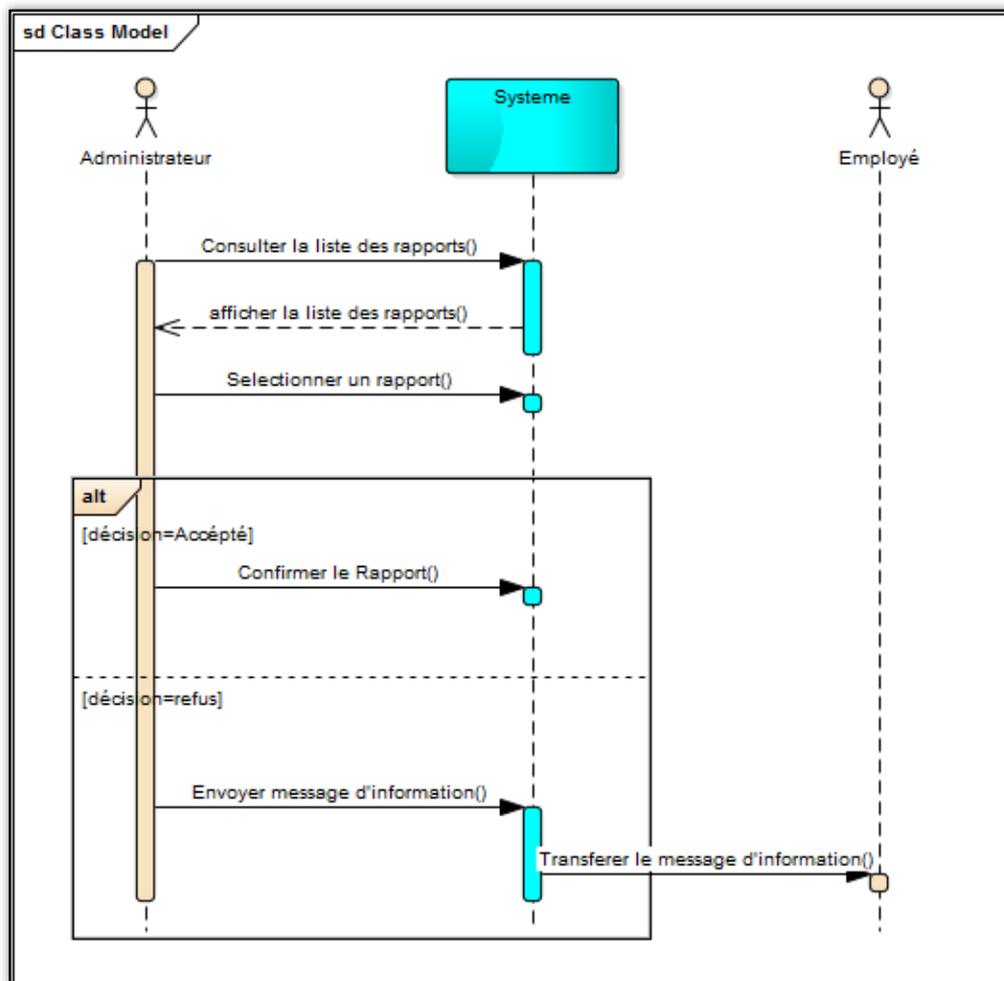


Figure 15: Diagramme de séquence « Gérer les dépenses »

b) Description textuelle

<b>Nom</b>	<b>Gérer les dépenses</b>
<b>Acteur</b>	Administrateur (manager)
<b>Description</b>	L'administrateur doit pouvoir accepter ou refuser les dépenses.
<b>Auteur</b>	Sarra Zennaki
<b>Date (s)</b>	01/04/2019
<b>Pré-conditions</b>	-L'utilisateur doit être authentifié. -Avoir l'accès au serveur. -les dépenses doivent être soumises par l'employé.
<b>Démarrage</b>	Le manager a demandé la liste des rapports.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système affiche la liste des rapports.</li> <li>2. L'utilisateur consulte les rapports.</li> <li>3. Si l'utilisateur accepte la note de frais, il confirme le rapport.</li> <li>4. Le système envoie la note de frais au comptable.</li> </ol>
<b>Scénario D'exception</b>	3-a. L'utilisateur refuse la note de frais, et envoie un message pour informer l'employé sur la raison de refus.
<b>Fin</b>	<ul style="list-style-type: none"> <li>• Scénario nominal : la note de frais est transmise au comptable, après le point 4.</li> <li>• Scénario d'exception : sur la décision de l'administrateur après le point 3-a.</li> </ul>
<b>Post-conditions</b>	<ul style="list-style-type: none"> <li>• La dépense a été enregistrée dans le serveur.</li> </ul>

### V.3. Diagramme de classe

Le diagramme de classes dans le langage UML est un type de diagramme de structure statique qui décrit la structure d'un système / application.

En pratique, le diagramme de classes UML est utilisé pour visualiser, décrire et documenter divers aspects d'une application, mais également pour construire le code exécutable de l'application logicielle. Il contient les classes du système, les attributs, les opérations et les relations entre les classes.

Le diagramme de classes UML est également appelé diagramme structurel. C'est le point de départ le plus courant pour les architectes logiciels et les développeurs lors de la conception d'un système [17].

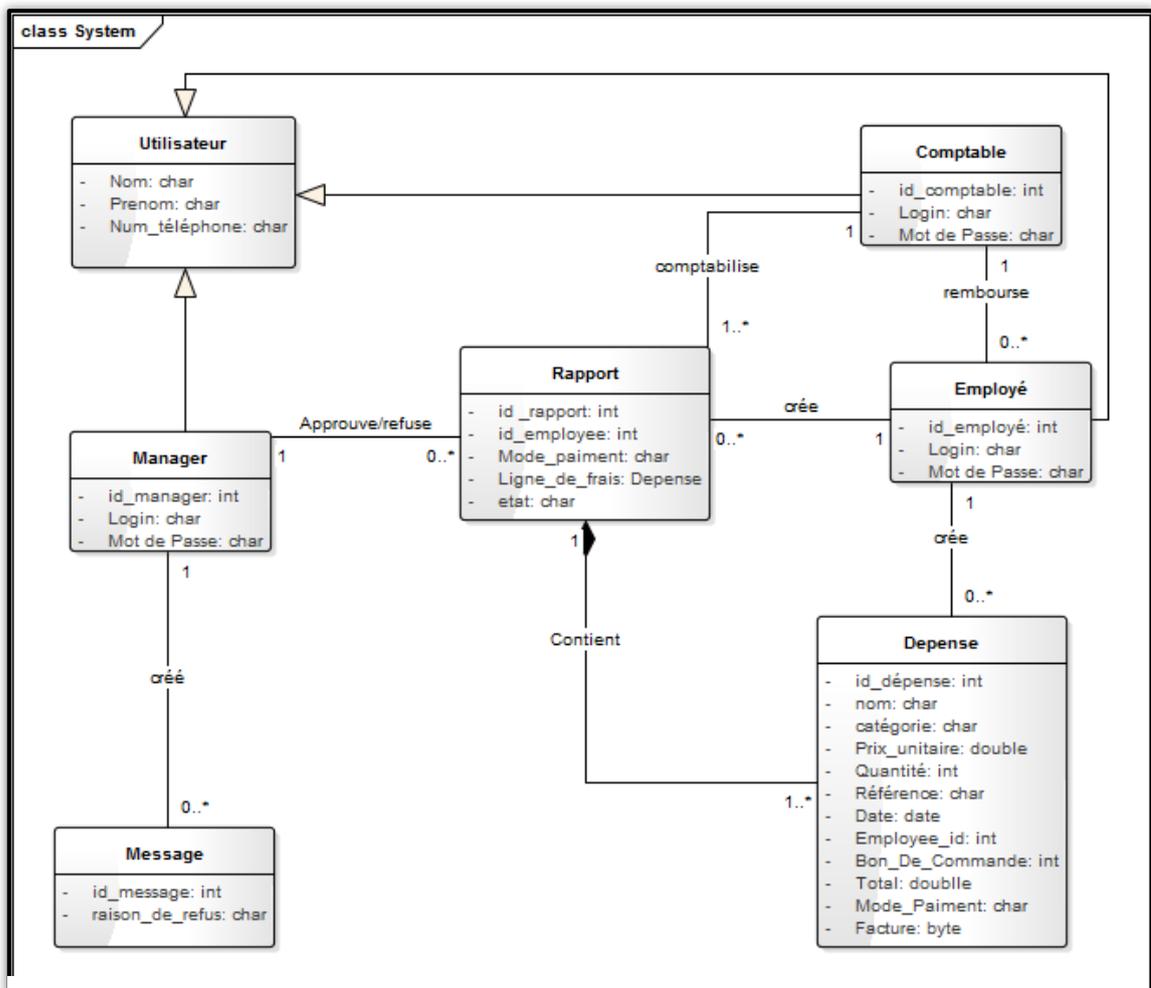


Figure 16: Diagramme de classe.

## VI. Conclusion

La conception est une phase très importante dans la gestion de projet, elle donne des opportunités de valeur pour les entreprises. Une bonne conception peut apporter certains avantages importants à l'entreprise.

Ce chapitre a donné une vision de notre travail. Nous avons donné l'aspect conceptuel de l'application "Expense Manager" à travers les différents diagrammes décrits en UML. Les diagrammes de cas d'utilisation illustrent les fonctionnalités du système, les diagrammes de séquence illustrent le processus de visualisation des cas d'utilisation et le diagramme de classe qui représente les différentes classes métiers de notre système.

Nous allons voir par la suite l'implémentation de notre application dans le chapitre suivant.

Chapitre 3 : Réalisation de

---

L'application mobile

---

« Expense Manager »

---

## I. Introduction

L'intérêt d'une application fonctionnant à distance peut à première vue sembler inutile dans la mesure où les applications fonctionnent fort bien en local (sur le poste de l'utilisateur), néanmoins une application distante peut répondre aux problématiques suivantes :

- Les données peuvent être présentes uniquement sur le serveur distant (par exemple un catalogue produit, un classement en temps réel, etc.) ;
- Le serveur distant peut disposer d'une puissance de calcul ou de capacités de stockage dont l'utilisateur local ne dispose pas ;
- L'application distante peut être utilisée simultanément par un grand nombre d'utilisateurs et sa mise à jour n'intervient qu'à un seul endroit.

Pour toutes ses raisons, une interaction entre des programmes distants peut être utile. Les Web Services apportent une solution à ce problème en définissant une manière standard d'invoquer une application distante et d'en récupérer les résultats à travers le web [18].

Dans ce chapitre nous avons présenté les services web le mécanisme qui relie entre le coté mobile (Android) et le coté web (Odo) de notre solution, ensuite le protocole utilisé dans notre cas l'XML-RPC, après nous avons définis les outils et les langages utilisés dans notre projet. Puis nous avons entamé la réalisation de l'application Expencc Manager.

## II. Un service web

Un service web est un module logiciel qui permet à différentes applications distantes écrites dans des langages de programmation différents de communiquer entre elles via le réseau internet, en :

- Utilisant le protocole HTTP comme moyen de transport. Ainsi, les communications s'effectuent sur un support universel, maîtrisé et généralement non filtré par les pare-feux ;
- Employant une syntaxe basée sur la notation XML pour décrire les appels de fonctions distantes et les données échangées ;
- Organisant les mécanismes d'appel et de réponse [18].

Grâce aux services web, les applications peuvent être vues comme un ensemble de services métiers, structurés et correctement décrits, dialoguant selon un standard international plutôt qu'un ensemble d'objets et de méthodes entremêlés [18].

## II.1. Avantages

Le premier bénéfice de ce découpage est la facilité de maintenance de l'application, ainsi que l'interopérabilité permettant de modifier facilement un composant (un service) pour le remplacer par un autre, éventuellement développé par un tiers. Qui plus est, les services web permettent de réduire la complexité d'une application car le développeur peut se focaliser sur un service, indépendamment du reste de l'application.

Les services web facilitent non seulement les échanges entre les applications de l'entreprise mais surtout permettent une ouverture vers les autres entreprises. Les premiers fournisseurs de services web sont ainsi les fournisseurs de services en ligne (météo, bourse, planification d'itinéraire, pages jaunes, etc.), mettant à disposition des développeurs des API (Application Programmable Interface) payantes ou non, permettant d'intégrer leur service au sein d'applications tierces [18].

## II.2. Fonctionnement

Le fonctionnement des services web repose sur un modèle en couches, dont les trois couches fondamentales sont les suivantes [18] :

- **Invocation**, visant à décrire la structure des messages échangés par les applications.
- **Découverte**, pour permettre de rechercher et de localiser un service web particulier dans un annuaire de services décrivant le nom de la société, l'objectif de chaque service, ...etc.
- **Description**, dont l'objectif est la description des interfaces (paramètres des fonctions, types de données) des services web.

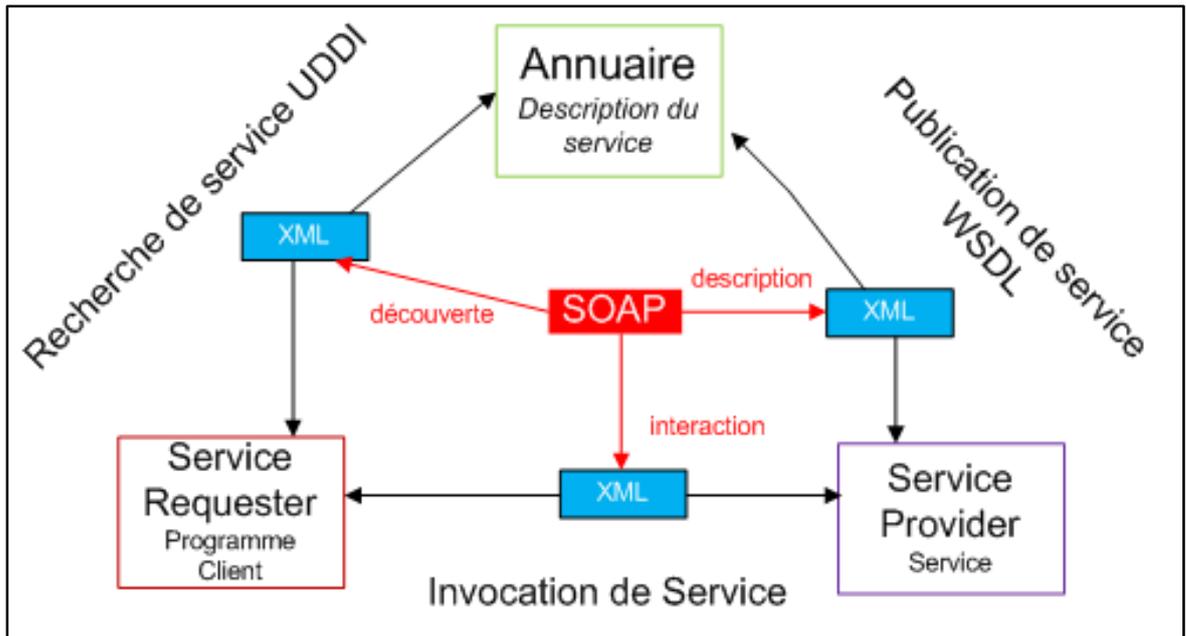


Figure 17 : Schéma de fonctionnement des web services [19]

### II.3. Architecture

Les principales technologies utilisées par les web services sont :

#### ➤ XML-RPC

Xml-RPC (Extensible Markup Language -Remote procedure call) est un protocole simple utilisant XML pour effectuer des messages RPC. Les requêtes sont écrites en XML et envoyées via HTTP POST. Les requêtes sont intégrées dans le corps de la réponse HTTP. XML-RPC est indépendant de la plate-forme, ce qui lui permet de communiquer avec diverses applications. Par exemple, un client Java peut parler de XML-RPC à un PerlServer [20].

#### ➤ SOAP

SOAP (*Simple object Access Protocol*) est un protocole d'accès aux services Web basé sur des normes qui existe depuis un certain temps et qui bénéficie de tous les avantages d'une Utilisation à long terme. Développé à l'origine par Microsoft, SOAP n'est pas aussi simple que L'acronyme le suggère [20].

➤ **REST**

REST (*Representational State Transfer*) est une architecture de services Web. Élaborée en l'an 2000 par *Roy Fielding*, l'un des créateurs du protocole HTTP, du serveur Apache httpd et d'autres travaux fondamentaux. REST est une manière de construire une application pour les systèmes distribués comme le World Wide Web [20].

➤ **WSDL**

WSDL (*Web Services Description Language*) est un langage de description standard. C'est L'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées [20].

➤ **UDDI**

UDDI (*Universal Description, Discovery and Integration*) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services web aux clients. Les informations qu'il contient peuvent être séparées en trois types [20] :

- les pages blanches qui incluent l'adresse, le contact et les identifiants relatifs au service Web
- les pages jaunes qui identifient les secteurs d'affaires relatifs au service web ;
- les pages vertes qui donnent les informations techniques.

## **II.4. SOAP vs REST**

Les protocoles SOAP et REST sont deux réponses à la même question : comment accéder aux services Web. Le choix initial peut sembler facile, mais il peut parfois être étonnamment difficile.

❖ **SOAP Vs Rest**

SOAP est définitivement le choix poids lourd pour l'accès aux services Web. Il offre les avantages suivants par rapport à REST [21] :

- Indépendant de la langue, de la plate-forme et du transport (REST nécessite l'utilisation de HTTP)
- Fonctionne bien dans les environnements d'entreprise distribués (REST suppose une communication directe point à point)
- Normalisé
- Offre une extensibilité significative avant la construction sous la forme des normes WS
- Traitement d'erreur intégré
- Automatisation lorsqu'il est utilisé avec certains produits de langage.

#### ❖ **REST vs SOAP**

REST est généralement plus facile à utiliser et plus flexible. Il présente les avantages suivants par rapport à SOAP [21] :

- Aucun outil coûteux ne nécessite d'interagir avec le service Web
- Courbe d'apprentissage réduite
- Efficace (SOAP utilise XML pour tous les messages, REST peut utiliser des formats de message plus petits)
- Rapide (aucun traitement approfondi requis)
- Plus proche des autres technologies Web dans la philosophie de conception.

### **III. Schéma du protocole Xml-RPC**

Les principaux composants d'un service web sont les données transférées entre le client et le serveur, elles sont écrites en xml et envoyées entre les applications à l'aide d'un protocole, dans notre cas c'est l'XML-RPC.

La figure ci-dessous représente les procédures de communication entre deux applications à l'aide de XML-RPC :

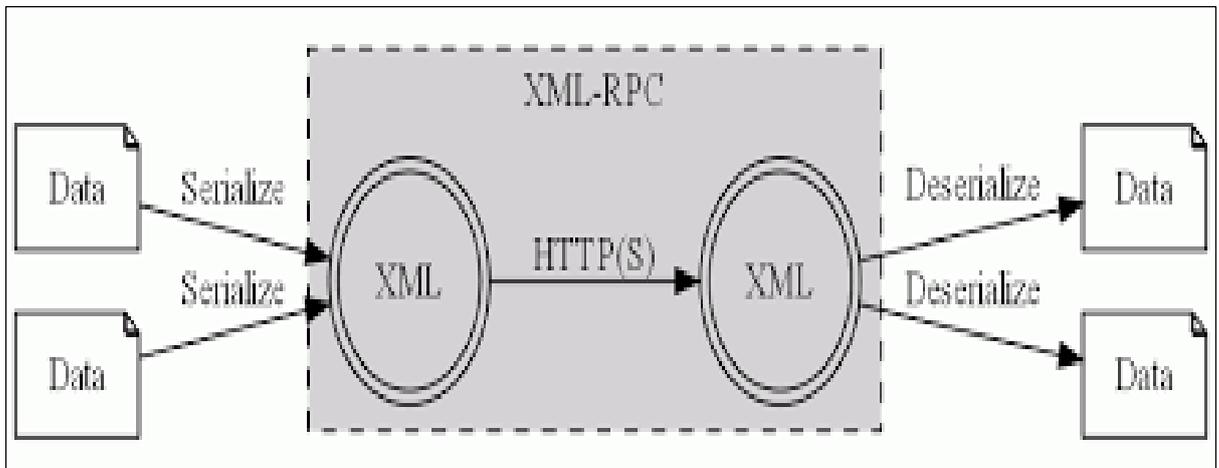


Figure 18: schéma du Protocole XML-RPC [22]

#### IV. Outils utilisés

- **Android studio**



Figure 19: logo d'Android studio [23]

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS et Linux [23].

- **PostgreSQL**



PostgreSQL

Figure 20: logo de PostgreSQL [24]

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. Ce système est concurrent à d'autres systèmes de gestion de base de données, qu'ils soient libres comme (MariaDB et Firebird), ou propriétaire comme (Oracle, MySQL, Sybase, DB2, Informix et Microsoft SQL Server).

Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises [24].

- **Sublime text 3**



Figure 21: logo de sublime text [25]

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.

Depuis la version 2.0, sortie le 26 juin 2012, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares [25].

Dans notre projet nous l'avons utilisé dans le côté Application centrale, pour apporter la modification nécessaire à Odoo (droit d'accès, ajout des champs...etc.).

- **Entreprise Architect**



Figure 22: logo d'entreprise architect [26]

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus [26].

Cet outil nous a servi dans la modélisation des diagrammes du projet.

- **Ms Project**



Figure 23: logo de Ms project [27]

Microsoft Project (ou MS Project ou MSP) est un logiciel de gestion de projets édité par Microsoft. Il permet aux chefs de projet et aux planificateurs de planifier et piloter les projets, de gérer les ressources et le budget, ainsi que d'analyser et communiquer les données des projets [27].

C'est l'outil qui nous a aidé à faire la gestion du projet et de créer le diagramme de Gantt.

## V. Langages

- **Python**



Figure 24: logo du langage python [28]

Python est un langage de programmation interprété, 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 de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl [28].

- **Xml**



Figure 25: logo du langage XML [29]

L'*Extensible Markup Language*, généralement appelé XML, « langage de balisage extensible » en français, est un métalangage informatique de balisage générique qui est un sous-ensemble du Standard Generalized Markup Language (SGML). Sa syntaxe est dite « extensible » car elle permet de définir différents langages avec chacun leur vocabulaire et leur grammaire [29].

- **Java android**



Figure 26: logo de java android [30]

Java est un langage de programmation et une plateforme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts [30].

## VI. La Reconnaissance optique de caractères (OCR)

### VI.1. Définition

La reconnaissance optique de caractères est la conversion mécanique ou électronique d'images de textes dactylographiés, imprimés ou même manuscrits, en textes codés par machine. Le texte peut provenir d'un document scanné, d'une image, ou de textes sous-titrés superposés à une image (comme dans une émission télévisée) [31].

### VI.2. Différence entre OCR et extraction des données

L'OCR est une technique qui permet la conversion d'une image en texte. La brique suivante est l'extraction intelligente de données qui reconnaît et traite le texte extrait de l'OCR pour le transformer ensuite en données pertinentes [31].

Il existe trois principales technologies d'extraction de données [31] :

1. Extraction vérifiée par l'homme ou externalisée.
2. Extraction par zones à partir de modèles prédéfinis.
3. Systèmes reposant sur l'intelligence artificielle (IA) ou le Machine Learning.

L'OCR seul ignore ce qu'il doit faire des informations qu'il lit, nous avons besoin d'une solution qui offre une technologie complète, une solution qui combine à la fois OCR, l'extraction intelligente de données et Machine Learning [31].

### VI.3. Différents aspects de l'OCR

Il n'existe pas de système universel d'OCR qui permet de reconnaître n'importe quel caractère dans n'importe quelle fonte. Tout dépend du type de données traitées et bien évidemment de l'application visée [32].

Il existe plusieurs modes de classification des systèmes OCR parmi lesquels on peut citer [33] :

- Les systèmes qualifiés de « en-ligne » ou « hors-ligne » suivant le mode d'acquisition.
- Les approches globales ou analytiques selon que l'analyse s'opère sur la totalité du mot, ou par segmentation en caractères.
- Les approches statistiques, structurelles ou stochastiques relatives aux traits caractéristiques extraits des formes considérées.

## VII. Reconnaissance en-ligne et hors-ligne

Ce sont deux modes distincts, chacun a ses propres outils d'acquisition et ses algorithmes correspondants de reconnaissance [33].

- **La reconnaissance en-ligne (on-line) :**

Ce mode est destiné généralement à l'écriture manuscrites, et s'opère pendant l'écriture du contenu. L'écriture est représentée comme un ensemble de points dont les coordonnées sont en fonction du temps [34] [35].

La reconnaissance en-ligne présente un avantage majeur c'est la possibilité de correction et de modification de l'écriture de manière interactive vu la réponse en continu du système [36]. Il faut évidemment un matériel spécifique pour saisir un tel échantillon (stylos numériques, des stylets sur tablettes...etc.) [37].

- **La reconnaissance hors-ligne (off-line) :**

Ce mode est réservé aux documents imprimés et les manuscrits déjà rédigés, et peut être considéré comme le cas le plus général de la reconnaissance de l'écriture. Dans ce contexte, il est impossible de savoir comment ont été tracés les différents motifs. Il est seulement possible d'extraire des formes à partir de l'image, en s'appuyant sur les technologies de reconnaissance de forme [38]. L'interprétation de l'information est indépendante de la source de génération [39].

## VIII. Problèmes liés à l'OCR :

L'OCR est une technique très délicate et exigeante, une simple erreur peut tout changer. Parmi les choses qui peuvent influencées sur la performance d'OCR nous avons :

- **Qualité du document** : un simple pli peut changer le résultat ou même causer un dysfonctionnement, aussi si le document est une copie le traitement sera difficile par rapport à l'original, les mots peuvent n'apparaître pas claires et même être effacés.
- **Impression** : un document imprimé par l'imprimante est mieux par rapport à un document écrit avec un dactylo.
- **Qualité d'image** : l'angle de prise de photos a une grande influence sur le résultat, ainsi la stabilité de la main lors de la capture de photo, la luminosité aussi a une influence.
- **Structure de contenu et style de police** : la structure de contenu diffère d'un document à un autre surtout dans notre cas avec la facture, le choix des mots et leur emplacement, le type de la police.

Tous ces facteurs peuvent influencer sur le fonctionnement de l'OCR, ce sont les problèmes que nous avons rencontrés dans l'application mais il existe d'autres problèmes qui sont liés à cette technique.

## IX. Adaptation du module SDP

### IX.1. Présentation du module SDP

Le module donne la possibilité de :

- ❖ Créer les dépenses :

La création de Dépenses se fait soit via Le menu « Dépenses à soumettre », ou bien lors de la création d'un rapport (Note de frais).

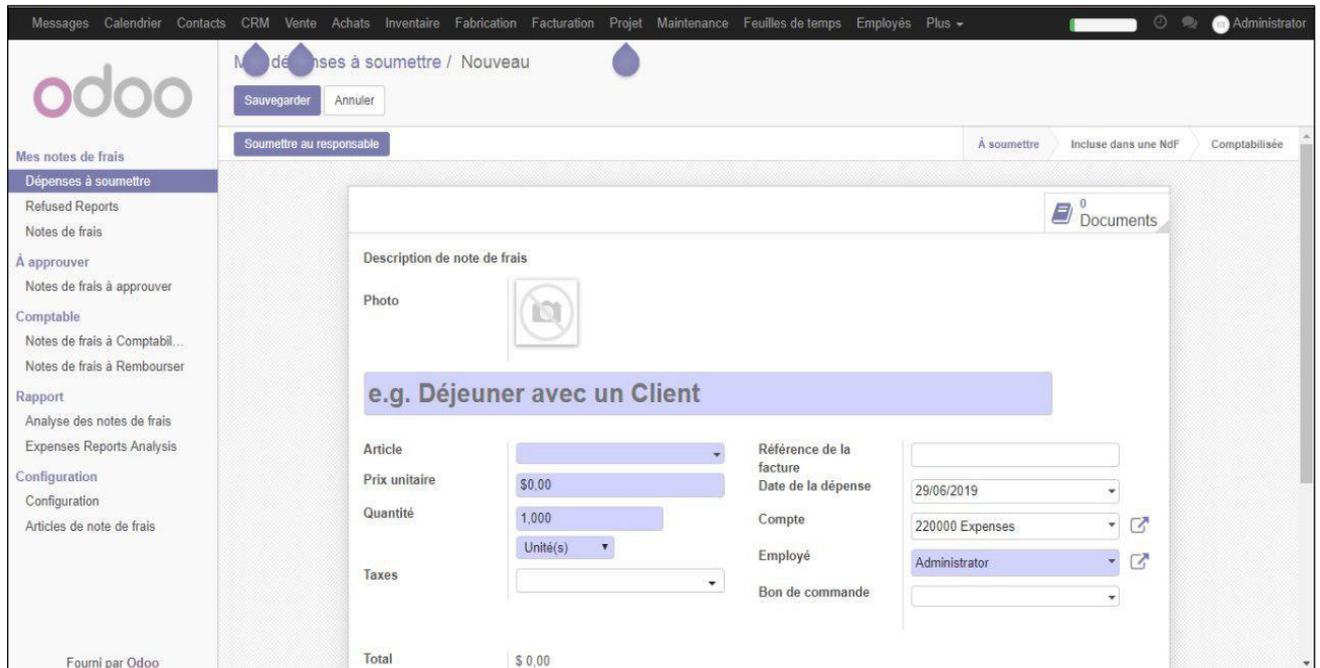


Figure 27 : Création d'une Dépense

❖ Créer un rapport (Note de frais) :

Dans cette partie, la dépense est déjà soumise au responsable et donc incluse dans un rapport, dans ce cas le responsable soit il accepte la dépense soit il la refuse.

Mes Rapports / Nouveau

Sauvegarder Annuler

Approuver Refuser

Soumis Confirmé par responsable Comptabilisée Payée

0 Documents

Résumé de la note de frais

**e.g. Voyage à New-York**

Employé Administrator

Paielement par

Ligne de frais Autres informations

Date de la dépense	Description de note de frais	Bon de commande	Taxes	Total	Montant hors taxe
Ajouter un élément					
				0,00	0,00

Figure 28:Création d'un rapport (note de frais)

❖ Créer des nouvelles catégories de note de frais :

Chaque dépense appartient à une catégorie, dans notre cas une catégorie est un article de note de frais, on peut créer un nouvel article à partir du menu « Articles de notes de frais ».

Articles de note de frais / [NDFB] Petites Fournitures bureau

Modifier Créer

Imprimer Action

2 / 10

Actif/ve

**Petites Fournitures bureau**

**Information Générale**

Type d'article: Consommable

Prix de vente: 0,00

Coût: 0,00

Unité de mesure: Unité(s)

Référence interne: NDFB

**Comptabilité**

Compte de dépenses: Taxes fournisseur

**Facturation**

Politique de facturation: Quantités commandées

Re-factorer les dépenses: Non

Figure 29:Exemple d'une catégorie créer pour notre application

## IX.2. Gestion des droits d'Access

Chaque employé à ces propres fonctions et certaines fonctions ne sont autorisés que pour l'administrateur. Odoo offre une gestion de droit d'accès simple soit à partir d'Odoo ou bien d'un fichier csv dans le module courant.

```

access_hr_expense_user,hr.expense.user,model_hr_expense,hr_expense.group_hr_expense_user,1,1,1,1
access_hr_expense_employee,hr.expense.employee,model_hr_expense,base.group_user,1,1,1,1
access_hr_expense_sheet_user,hr.expense.sheet.user,model_hr_expense_sheet,hr_expense.group_hr_expense_user,1,1,1,1
access_hr_expense_sheet_employee,hr.expense.sheet.employee,model_hr_expense_sheet,base.group_user,1,1,1,1
access_product_product_hr_expense_user,product.product.hr.expense.user,product.model_product_product,hr_expense.group_hr_expense_user,1,1,1,1
access_product_template_hr_expense_user,product.template.hr.expense.user,product.model_product_template,hr_expense.group_hr_expense_user,1,1,1,1
access_product_uom_hr_expense_user,product.uom.hr.expense.user,product.model_product_uom,hr_expense.group_hr_expense_user,1,1,1,1
access_account_journal_user,account.journal.user,account.model_account_journal,hr_expense.group_hr_expense_user,1,1,1,1
access_account_journal_employee,account.journal.employee,account.model_account_journal,base.group_user,1,0,0,0
access_account_invoice_user,account.invoice.user,account.model_account_invoice,hr_expense.group_hr_expense_user,1,1,1,1
access_account_invoice_line_user,account.invoice.line.user,account.model_account_invoice_line,hr_expense.group_hr_expense_user,1,1,1,1
access_account_invoice_tax_user,account.invoice.tax.user,account.model_account_invoice_tax,hr_expense.group_hr_expense_user,1,0,0,0
access_account_move_user,account.move.user,account.model_account_move,hr_expense.group_hr_expense_user,1,0,0,0
access_account_move_line_user,account.move.line.user,account.model_account_move_line,hr_expense.group_hr_expense_user,1,0,0,0
access_account_analytic_line_user,account.analytic.line.user,account.model_account_analytic_line,hr_expense.group_hr_expense_user,1,1,1,1
    
```

Figure 29:Droits d'accès du module SDP

## IX.3. Gestion des emails

La gestion des emails consiste à gérer les notes de frais envoyées par mail et qui arrivent à la messagerie d'Odoo. Pour cela, il est nécessaire de configurer un alias domaine ou les emails arrivent et de configurer le serveur de messagerie entrant. Ci-dessous des captures d'écran des étapes de configuration avec détails :

- Configuration de l'Alias domaine :

Après l'activation de messagerie externe dans Odoo, il est recommandé d'utiliser notre propre serveur mail.

La démonstration de cette étape est dans la figure ci-dessous :

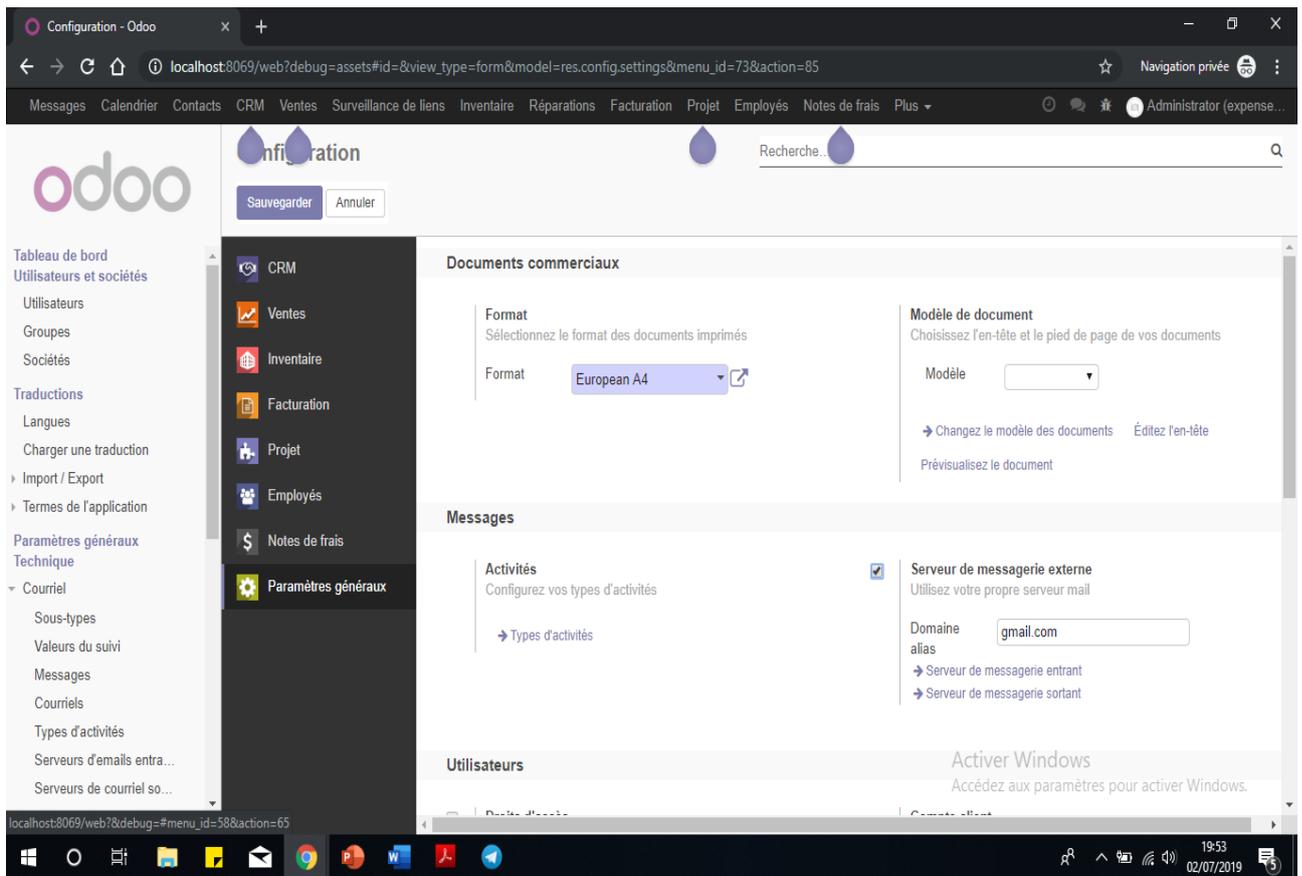


Figure 30: Activation de la messagerie interne et définition de l'alias domaine

➤ Configuration de serveur de messagerie entrant :

Dans cette étape nous avons configuré le serveur d'emails entrants, le choix du serveur est selon le besoin. Après le choix du serveur le nom du serveur doit être de la forme (nomServeur.notre propre serveur mail), après l'activation du SSL/TLS le port est renseigné automatiquement.

Dans l'étape suivante nous indiquons l'action à effectuer à l'arrivée des emails, dans notre cas on a besoin de créer une nouvelle dépense.

Dans la dernière étape nous testons la connexion si elle n'est pas échouée, l'état devient « confirmé ».

La figure suivante montre un exemple de configuration de serveur des messageries entrant :

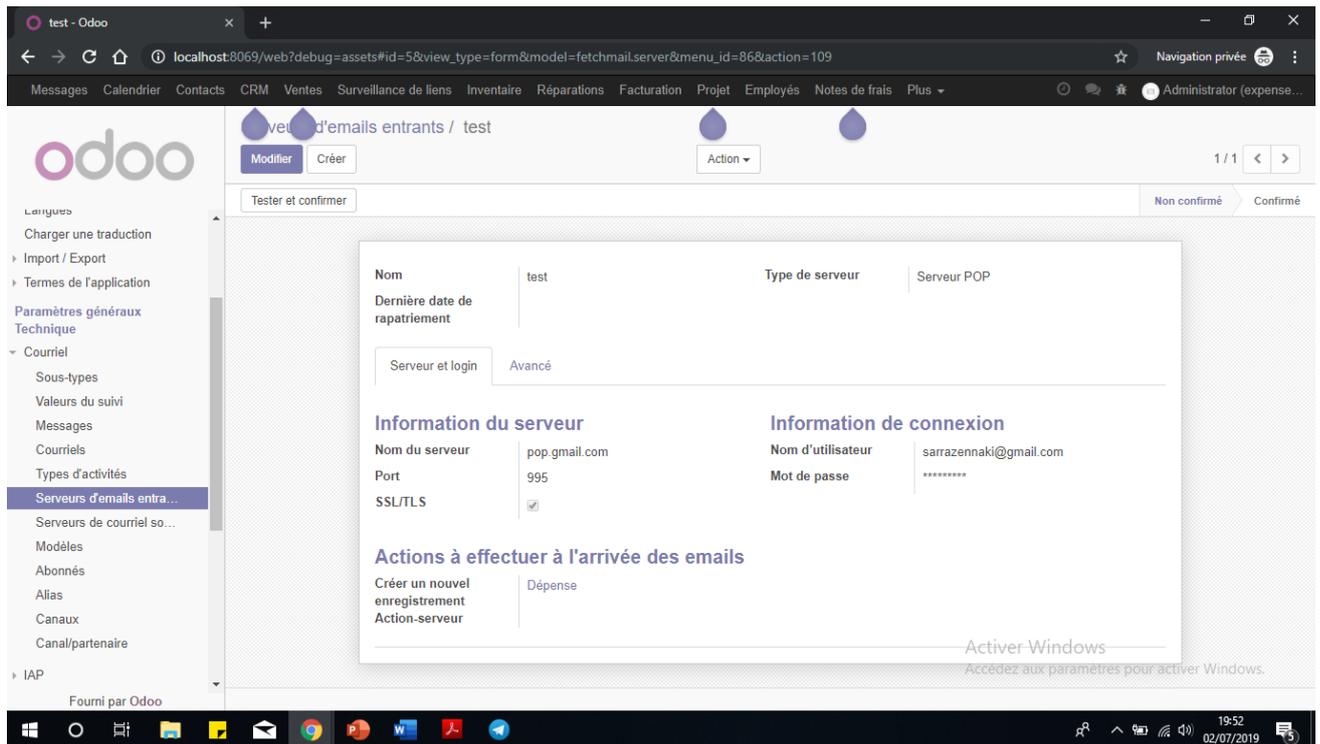


Figure 31: Configuration de serveur de messagerie entrant

## X. Présentation de l'application

Nous présentons dans cette partie notre application « Expense Manager » :

- **Ecran splash et authentification**

Au premier lancement de l'application on obtient une page qui contient le logo de l'application en attendant le lancement de la page d'authentification. Après l'écran splash, la page de connexion s'ouvre, et donne la main à l'utilisateur pour se connecter.



Figure 32: Ecran splash

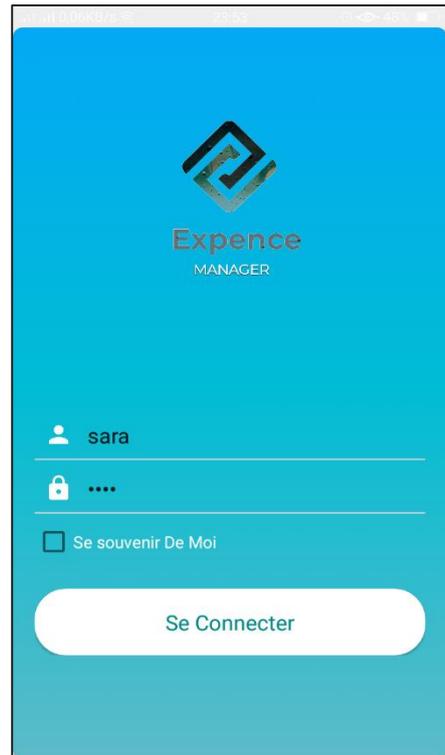


Figure 33:Page d'authentification

Après la vérification de l'adresse et du mot de passe, on obtient la page d'accueil qui contient la liste des dépenses et le menu principal de navigation :

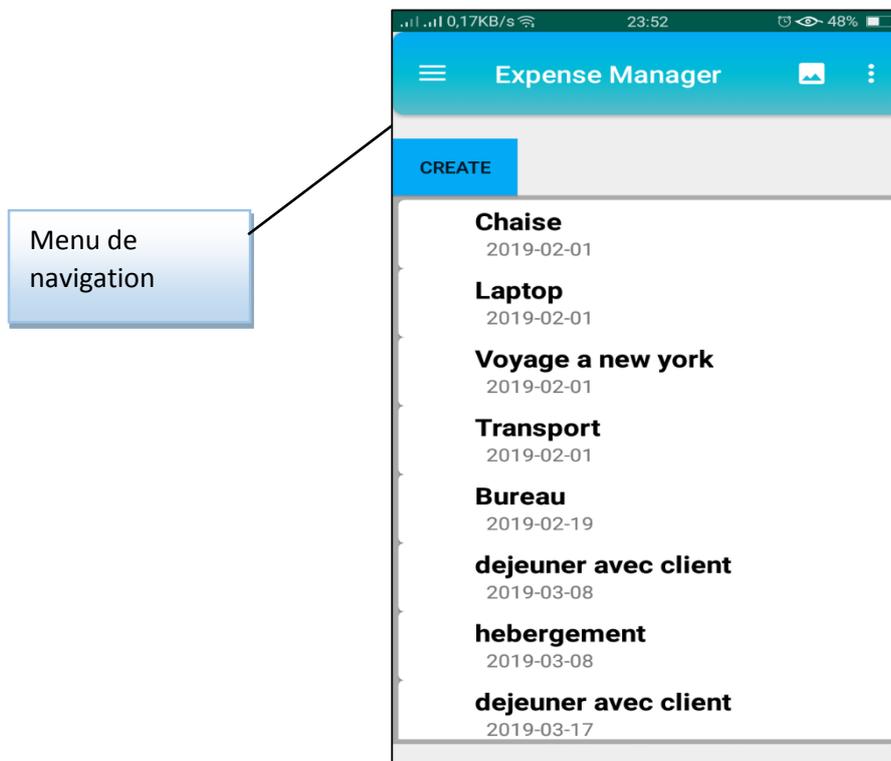


Figure 34:Accueil Liste dépenses

- **Menu**

Le menu de navigation contient plusieurs éléments et chaque élément a une fonction :

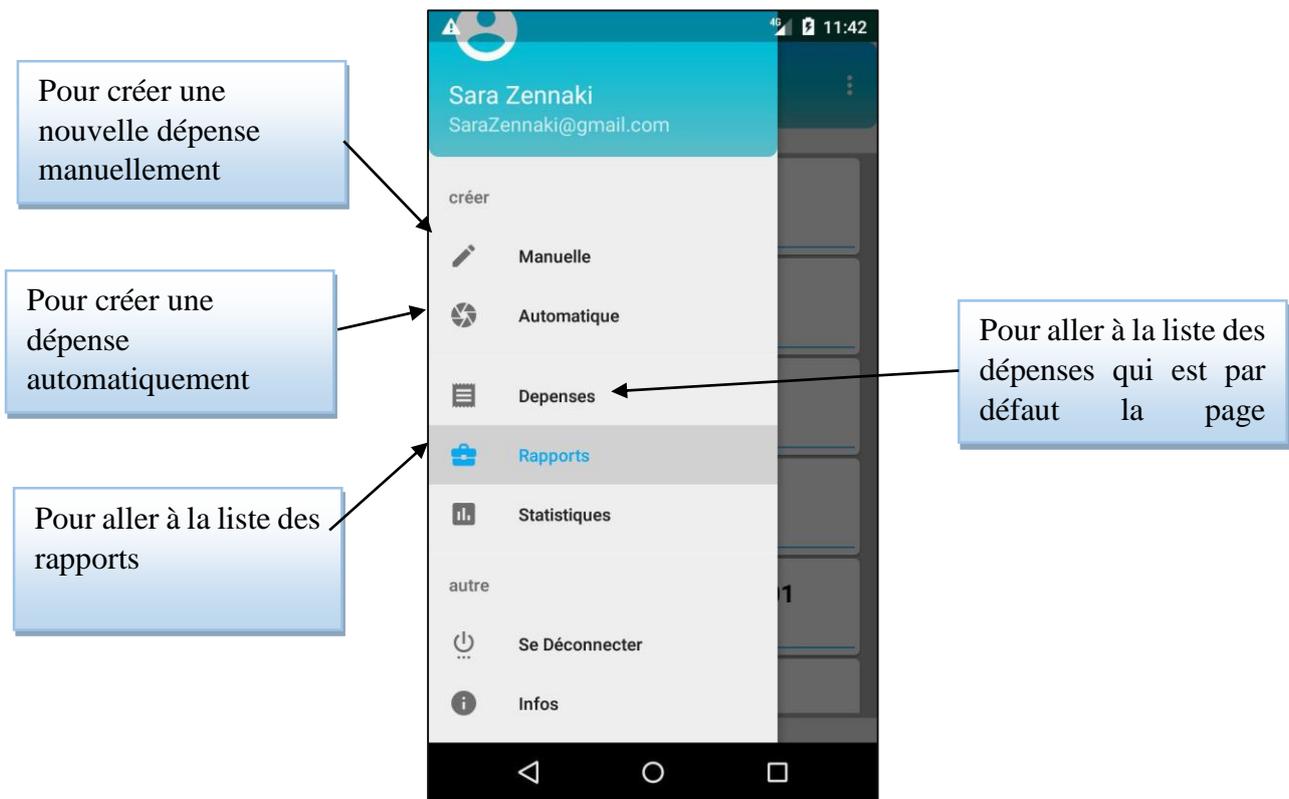


Figure 35: Menu de navigation

- **Aperçu d'une Dépense**

Lors de sélection d'une dépense à partir de la liste nous obtenons une page qui affiche ses détails, et qui contient un Button permettant de soumettre la dépense au responsable.

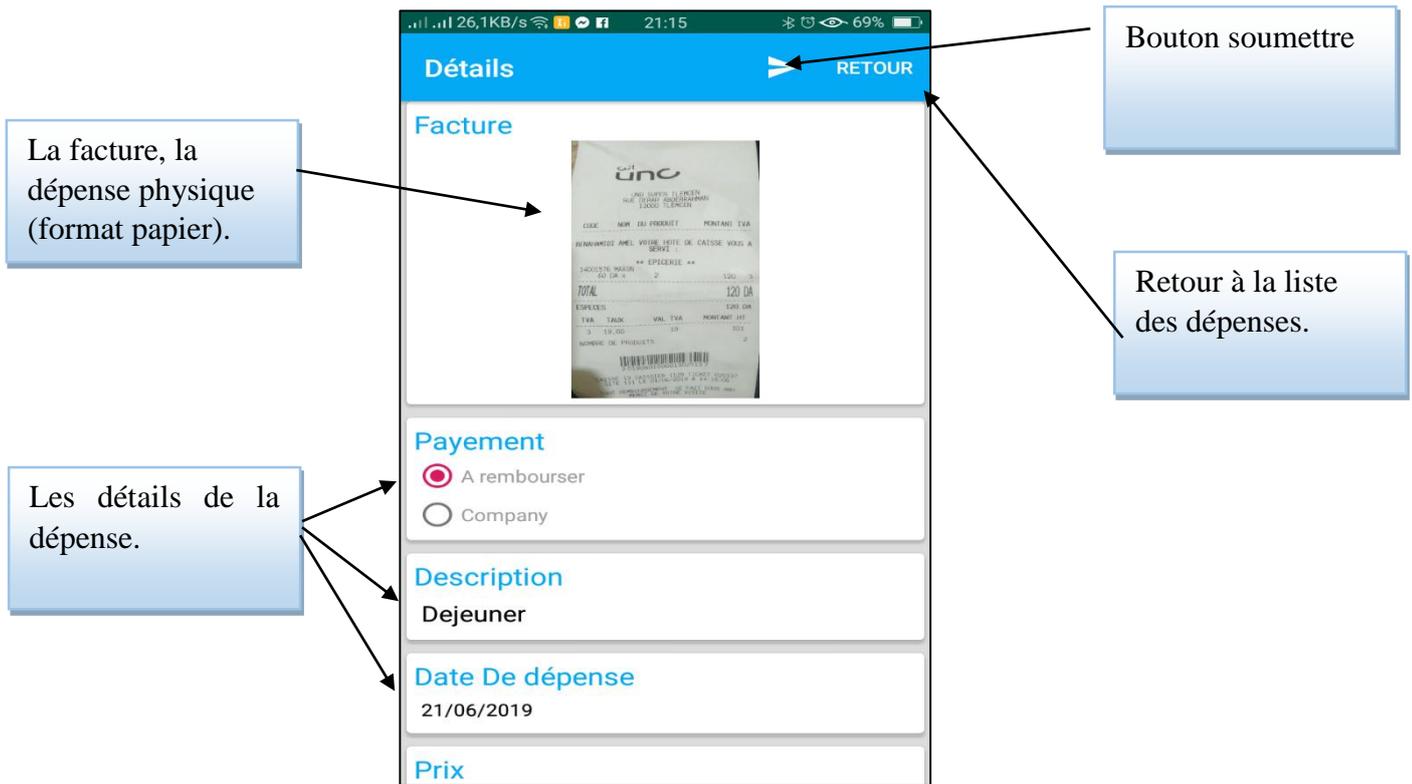


Figure 37:Afficher les détails d'une dépense et soumettre au responsable

- **Création de note de frais**

Les fonctions principales dans notre application sont la création automatique de la dépense et l'archivage.

Nous avons deux types de création de dépense, manuelle et automatique. Dans les deux cas la notion d'archivage est préservée.

- Dans le cas de la création automatique en prenant une photo ou bien en parcourant une photo de la galerie, l'application récupère automatiquement les données nécessaires et les met dans leur endroit correspondant. Ces données peuvent être : la tva, la date et le total. Les autres champs sont saisis manuellement par l'utilisateur.
- Dans le cas de la création manuelle, les champs sont renseignés par l'utilisateur.

- La page de création contient aussi un menu qui permet de prendre une photo ou bien de la rechercher à partir de la galerie. Le bouton permet de valider la dépense.

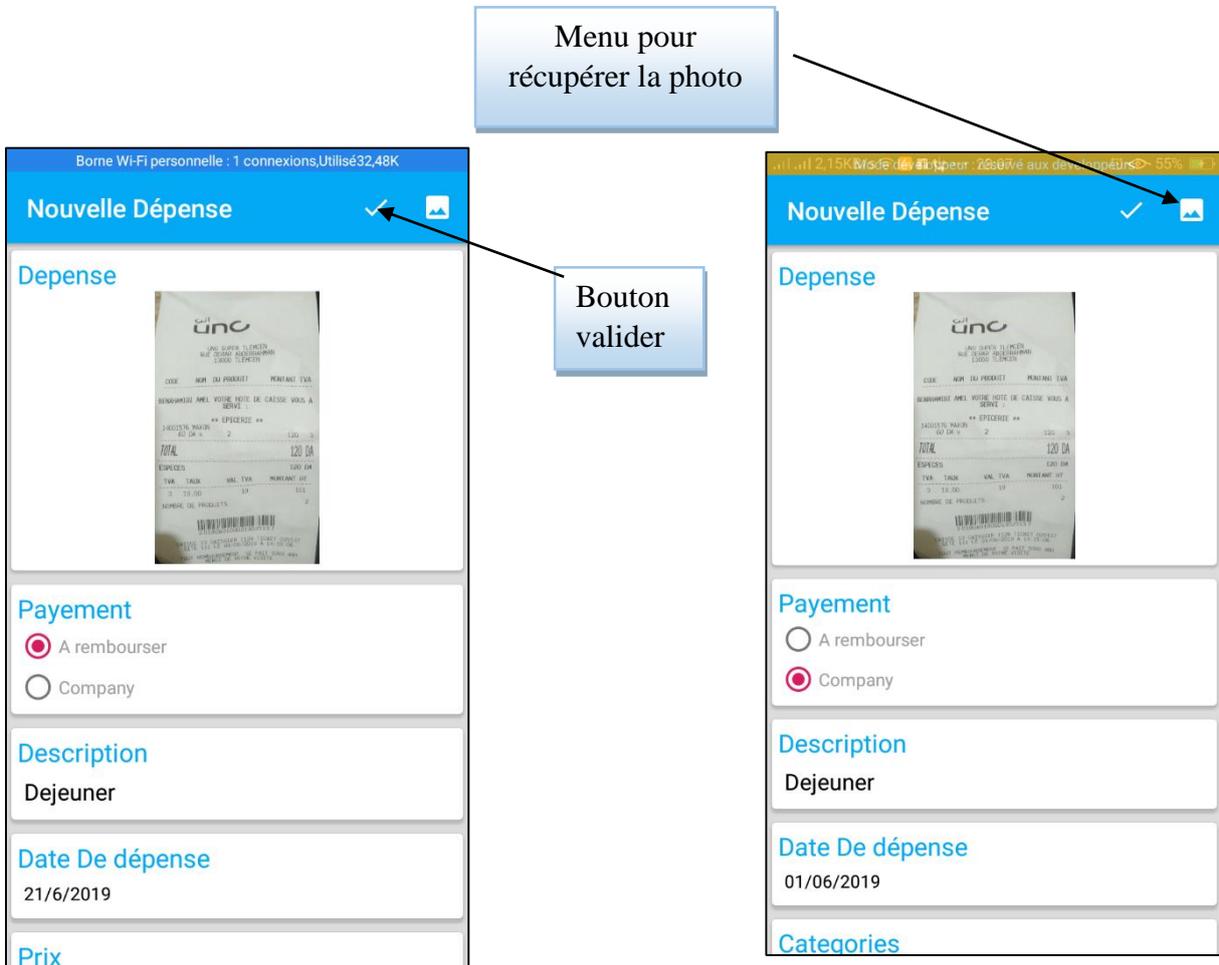


Figure 38:Création manuelle

Figure 39:Création avec l'OCR

## XI. Conclusion

Dans ce chapitre nous avons commencé par présenter les services web et les différents protocoles de connexions, ensuite nous avons spécifié le protocole utilisé pour relier entre notre Application Mobile et le coté web (Odo), après nous avons donné une vision sur la technique OCR en donnant quelques principes et aspects qui peuvent initier le lecteur a cette technique, et à la fin nous avons présenté notre application avec des aperçus de celle-ci.

---

# Conclusion Générale

Dans ce projet de fin d'étude, nous avons proposé une application mobile de gestion de note de frais et synchronisation avec un module Odoo. Nous sommes passés par plusieurs phases au cours de cette période, nous avons commencé par la première phase d'analyse des besoins pour connaître le contexte, déterminer les besoins et les contraintes, et pour préparer le cahier de charges.

Ensuite nous avons fait une période de formation sur les différents outils, langages, protocoles que nous devons utiliser dans notre application, ainsi que l'installation de l'environnement de travail (Odoo et ses modules, Android). Après la conception détaillée où nous avons modéliser notre solution nous avons passé à la réalisation du notre système.

Ce projet a été, pour nous, une occasion de compléter notre formation universitaire, une opportunité pour se familiariser avec le monde professionnel et d'appliquer ce que nous avons appris durant toutes ces années sur un projet réel. Cela nous a permis d'avoir une expérience qui va nous servir dans notre carrière dans le futur. Aussi nous avons appris beaucoup de choses intéressantes dont une grande partie sur les nouvelles technologies, en passant par l'intelligence artificielle dans la partie OCR, puis par l'outil de gestion dans les entreprises l'ERP notamment Odoo...etc.

Nos perspectives sont de répondre à tous les besoins du client, vu que nous étions limités par le temps, nous n'avons pas pu intégrer d'autres fonctionnalités qui peuvent améliorer les performances et la qualité de notre application.

Comme perspectives, nous pouvons citer :

- Gestion des frais kilométriques.
- Gestion des devises.

## Références bibliographiques

Code	Référence	Consulter le
[1]	<a href="https://www.choisirmonerp.com/erp/definition-d-un-erp">https://www.choisirmonerp.com/erp/definition-d-un-erp</a> . Consulter le	05/05/2019
[2]	<a href="https://www.celge.fr/comparer-les-logiciels-erp">https://www.celge.fr/comparer-les-logiciels-erp</a> . Consulter le	05/05/2019
[3]	<a href="https://www.appvizer.fr/magazine/operations/erp/top-5-erp-gratuit-open-source">https://www.appvizer.fr/magazine/operations/erp/top-5-erp-gratuit-open-source</a> . Consulter le	01/05/2019
[4]	<a href="https://erp.ooreka.fr/comprendre/marche-erp">https://erp.ooreka.fr/comprendre/marche-erp</a> . Consulter le	02/05/2019
[5]	<a href="https://www.solutions-numeriques.com/top-10-erp-sap-et-oracle-bien-places/">https://www.solutions-numeriques.com/top-10-erp-sap-et-oracle-bien-places/</a> . Consulter le	01/05/2019
[6]	<a href="https://www.odoo.com/fr_FR/">https://www.odoo.com/fr_FR/</a> . Consulter le	06/05/2019
[7]	<a href="https://www.celge.fr/editeurs/odoo-logiciel-de-gestion-dentreprise-crm-erp-facturation-comptabilite-gpao-cms-ecommerce">https://www.celge.fr/editeurs/odoo-logiciel-de-gestion-dentreprise-crm-erp-facturation-comptabilite-gpao-cms-ecommerce</a> . Consulter le	11/05/2019
[8]	<a href="https://fr.wikipedia.org/wiki/Odoo">https://fr.wikipedia.org/wiki/Odoo</a> . Consulter le	19/03/2019
[9]	<a href="https://doc.odoo.com/v6.0/developper/1_2_module_development/1_server_module/">https://doc.odoo.com/v6.0/developper/1_2_module_development/1_server_module/</a> . Consulter le	05/06/2019
[10]	<a href="https://www.odoo.com/fr_FR/page/employees">https://www.odoo.com/fr_FR/page/employees</a> . Consulter le	06/05/2019
[11]	<a href="https://www.evoliz.com/aide/notes-de-frais/460-decouverte-notes-de-frais.html">https://www.evoliz.com/aide/notes-de-frais/460-decouverte-notes-de-frais.html</a> . Consulter le	07/06/2019
[12]	<a href="https://www.memoireonline.com/07/08/1363/m_conception-realisation-application-gestion-centre-kinesie9.html">https://www.memoireonline.com/07/08/1363/m_conception-realisation-application-gestion-centre-kinesie9.html</a> . Consulter le	18/06/2019
[13]	<a href="http://www.sogesi-dz.com/">http://www.sogesi-dz.com/</a> . Consulter le	04/05/2019
[14]	<a href="https://fr.wikipedia.org/wiki/Diagramme_de_Gantt">https://fr.wikipedia.org/wiki/Diagramme_de_Gantt</a> . Consulter le	20/06/2019
[15]	<a href="https://fr.wikipedia.org/wiki/Diagramme_des_cas_d'utilisation">https://fr.wikipedia.org/wiki/Diagramme_des_cas_d'utilisation</a> . Consulter le	18/05/2019
[16]	Pascal Roques, UML Modéliser un site e-commerce,2 <sup>e</sup> , Paris, Editions Eyrolles, 2002, Les Cahiers Du Programmeurs, 152p	
[17]	<a href="https://www.genmymodel.com/fr/uml-class-diagram">https://www.genmymodel.com/fr/uml-class-diagram</a> . Consulter le	01/06/2019
[18]	<a href="https://web.maths.unsw.edu.au/~lafaye/CCM/web-services/web-service.htm">https://web.maths.unsw.edu.au/~lafaye/CCM/web-services/web-service.htm</a> . Consulter le	18/06/2019
[19]	<a href="http://souadatoui.blogspot.com/p/chapitre-1.html">http://souadatoui.blogspot.com/p/chapitre-1.html</a> . Consulter le	18/06/2019

[20]	<a href="http://www.pfl-cepia.inra.fr/uploads/images/GestionDonneesImages/unites/GdpDoc/OpenClassrooms-servicesWeb.pdf">http://www.pfl-cepia.inra.fr/uploads/images/GestionDonneesImages/unites/GdpDoc/OpenClassrooms-servicesWeb.pdf</a>	
[21]	<a href="https://smartbear.com/blog/test-and-monitor/understanding-soap-and-rest-basics/">https://smartbear.com/blog/test-and-monitor/understanding-soap-and-rest-basics/</a> . Consulter le	
[22]	<a href="http://xmlrpc.scripting.com/">http://xmlrpc.scripting.com/</a> . Consulter le	18/06/2019
[23]	<a href="https://fr.wikipedia.org/wiki/Android_Studio">https://fr.wikipedia.org/wiki/Android_Studio</a> . Consulter le	19/06/2019
[24]	<a href="https://fr.wikipedia.org/wiki/PostgreSQL">https://fr.wikipedia.org/wiki/PostgreSQL</a> . Consulter le	19/06/2019
[25]	<a href="https://fr.wikipedia.org/wiki/Sublime_Text">https://fr.wikipedia.org/wiki/Sublime_Text</a> . Consulter le	19/06/2019
[26]	<a href="https://fr.wikipedia.org/wiki/Enterprise_Architect">https://fr.wikipedia.org/wiki/Enterprise_Architect</a> . Consulter le	19/06/2019
[27]	<a href="https://fr.wikipedia.org/wiki/Microsoft_Project">https://fr.wikipedia.org/wiki/Microsoft_Project</a> . Consulter le	19/06/2019
[28]	<a href="https://fr.wikipedia.org/wiki/Python_(langage)">https://fr.wikipedia.org/wiki/Python_(langage)</a> . Consulter le	19/06/2019
[29]	<a href="https://fr.wikipedia.org/wiki/XML">https://fr.wikipedia.org/wiki/XML</a> . Consulter le	19/06/2019
[30]	<a href="https://fr.wikipedia.org/wiki/Android">https://fr.wikipedia.org/wiki/Android</a> . Consulter le	19/06/2019
[31]	<a href="https://cloud.getyooz.com/fr/blog/dematerialisation-ocr...[30]">https://cloud.getyooz.com/fr/blog/dematerialisation-ocr...[30]</a> . Consulter le	20/06/2019
[32]	N. Benamara , « Utilisation des modèles de Markov cachés planaires en reconnaissance de l'écriture arabe imprimée »,Thèse de doctorat, spécialité Génie Electrique, Université des sciences, des Techniques et de médecine de Tunis II, 1999	
[33]	<a href="https://www.memoireonline.com/09/09/2694/m_Systeme-de-reconnaissance-hors-ligne-des-mots-manuscrits-arabe-pour-multi-scripteurs0.html">https://www.memoireonline.com/09/09/2694/m_Systeme-de-reconnaissance-hors-ligne-des-mots-manuscrits-arabe-pour-multi-scripteurs0.html</a> . Consulter le	20/06/2019
[34]	E.Lecolinet, O. Barrett : « Cursive word recognition : Methods and strategies ». In NATO/ASI , Fundamentals in handwriting recognition, Bonas, France June 21-july 3, 1993	
[35]	B. Al-Badr , S.A. Mahmoud : « Survey and bibliography of Arabic optical text recognition ». Signal processing , vol. 41, pp. 49-77, 1995	
[36]	P.M. Lallican, C. Viarp-Gaudin, S. Knerr : « From off-line to on-line handwriting recognition ». Proc. 7 <sup>th</sup> workshop on frontiers in handwriting recognition, pp. 303-312, Amsterdam 2000	

[37]	<a href="http://www.mcours.com/telecharger/memoire-en-ligne-segmentation-de-textes-en-caracteres-pour-la-reconnaissance-optique-de-lecriture-arabe/">http://www.mcours.com/telecharger/memoire-en-ligne-segmentation-de-textes-en-caracteres-pour-la-reconnaissance-optique-de-lecriture-arabe/</a> . Consulter le	20/06/2019
[38]	<a href="https://fr.wikipedia.org/wiki/Reconnaissance_de_l%27%C3%A9criture_manuscrite">https://fr.wikipedia.org/wiki/Reconnaissance de l%27%C3%A9criture manuscrite</a> . Consulter le	20/06/2019
[39]	I.R. Tsang : «Pattern recognition and complex systems». Thèse de doctorat, université d'Anterwerpen, 2000	

---

## Résumé

Afin d'améliorer leurs performances, les entreprises visent à automatiser la gestion interne de leurs activités en se servant des nouvelles technologies. La tendance ces jours-ci est aux ERP qui deviennent de plus en plus réponsus. La solution Open source Odoo permet de réduire le coût et d'apporter un gain en matière de de productivité et d'efficience.

A travers notre projet, nous avons réussi à optimiser la gestion de note de frais de l'employé. Notre solution combine L'ERP Open source Odoo avec Android. Nous avons opéré des modifications dans le module de gestion des dépenses SDP d'Odoo, qui est une partie de la GRH. Ce module est relié à une application mobile pour permettre à l'employé d'avoir une gestion portable simple et facile de la NDF en ajoutant la technique OCR pour la gestion automatique.

**Mots clés :** Odoo, ERP, NDF, Android, OCR, GRH, SDP.

## Abstract

In order to improve their performance, companies aim to automate the internal management of their activities by using new technologies. The trend these days is with ERP becoming more and more answered. The Odoo open source solution reduces cost and brings a gain in productivity and efficiency.

Through our project, we have managed to optimize the management of the employee's expense report. Our solution combines Open source ERP Odoo with Android. We have made changes to Odoo's SDP spend management module, which is a part of HRM. This module is connected to a mobile application to allow the employee to have simple and easy portable management of the NDF by adding the OCR technique for automatic management.

**Keywords:** Odoo, ERP, NDF, Android, OCR, HRM, SDP.

## ملخص

من أجل تحسين أدائها، تهدف الشركات إلى أتمتة الإدارة الداخلية لأنشطتها باستخدام تقنيات جديدة. الميول والرغبات تتجه في هذه الأيام نحو ERP والتي أصبحت شيئاً فشيئاً أكثر استجابة.

SOGESI هي الشركة التي اقترحت علينا موضوع عملنا و الذي يهدف لتحسين مهام البرامج المختلفة باستخدام أدوات تكنولوجيا المعلومات.

من خلال مشروعنا، تمكنا من تحسين إدارة دفتر مصاريف الموظف. حلنا يدمج L'ERP Open source Odoo مع Android. لقد أجرينا تغييرات على وحدة إدارة الإنفاق SPD d'Odoo والذي هو جزء من GRH. هذه الوحدة متصلة بتطبيق على الهاتف المحمول بهدف السماح للموظف القيام بإدارة سهلة وبسيطة لل NDF وذلك باستخدام OCR من أجل الإدارة التلقائية.

**كلمات البحث :** OCR, ERP, SDP, Odoo, GRH, NDF