

| | | |
|---|---|-----------------------------------|
| <u>Université 20 Août 1955 – Skikda</u> |  | <u>جامعة 20 أوت 1955 - سكيكدة</u> |
| <u>Faculté des Sciences</u> | | <u>كلية العلوم</u> |
| <u>Departement d'informatique</u> | | <u>قسم الإعلام الآلي</u> |

**Mémoire De fin d'étude en vue de l'obtention du
Diplôme de Master en Informatique**

**Option : Systèmes d'Information Avancés et
Applications**

Sujet :

**Conception et réalisation d'un module ERP
pour la gestion des opérations escale au sein de
l'Entreprise Portuaire de Skikda**

Réalisé par l'étudiant :

- BOUCHELIT Fadi.

Encadré par :

- Dr Zeghida Djamel.

Co-Encadré par :

- BELKAHLA
Roumaissa

Année Universitaire 2023-2024

Remerciement

Ce mémoire, ne pourrait exister sans l'aide et l'engagement d'un certain nombre de personnes qui ont décidé de nous accompagner résolument dans notre parcours.

Que tous ceux qui, de près ou de loin, ont contribué par leurs conseils, leurs encouragements et leur assistance à l'aboutissement de ce travail, trouvent ici l'expression de notre profonde gratitude.

Tout d'abord, je tiens à remercier mon Co-encadreur : BELKAHLA Maissa pour les efforts qu'elle a fournis pour nous aider et nous orienter dans notre travail jusqu'à sa finalisation.

Je remercie les membres du jury pour avoir accepté d'examiner et évaluer notre travail.

Merci à mes proches et amis de nous soutenir par leur présence dans les bons comme dans les mauvais moments.

Merci à nos enseignants pour la qualité et le dévouement à la cause éducative. Qu'ils trouvent ici l'expression de toute notre reconnaissance.

Dédicace

Tout d'abord, je tiens à remercier DIEU
De m'avoir donné la force et le courage de mener
à bien ce modeste travail.

Je tiens à dédier ce projet de fin d'étude à:
ma tendre mère ET Mon très Cher père Et mon frère
mes amis ;

Tous ceux qui m'aiment et que j'aime.

Que Ce travail soit l'accomplissement de vos vœux tant allégués, ET
le fruit de votre soutien infaillible.

Fady.

ملخص

الهدف من هذه الأطروحة هو تصميم وإنشاء نظام تخطيط موارد المؤسسات لإدارة ومراقبة عمليات نداء السفن، وهو نتيجة لدراسة متعمقة للنظام الحالي خلال فترة التدريب لدينا في شركة الميناء سكيكدة.

استخدمنا UML للتصميم وبالنسبة للبرمجة اخترنا Odoo كحزمة برمجية مفتوحة المصدر مع لغة النمذجة Python التي تحترم نموذج التصميم MVC بالإضافة إلى PostgreSQL كنظام لإدارة قواعد البيانات .

كلمات مفتاحية: UML، ميناء سكيكدة، تخطيط موارد المؤسسات، أودو، رسو السفن

Résumé

L'objectif de ce mémoire est de concevoir et réaliser un ERP pour la gestion et le suivi des opérations des escales des navires, c'est le résultat d'une étude approfondie d'un système existant lors de notre stage à l'Entreprise Portuaire Skikda.

Nous avons utilisé le langage de modélisation UML pour la conception, et pour la programmation nous avons choisi Odoo comme un progiciel open source avec le langage python qui respecte le modèle de conception MVC, ainsi que PostgreSQL en tant que système de gestion de base de données.

Mots-clés : Entreprise Portuaire de Skikda, ERP, Odoo, Suivi des escales, UML.

Abstract

The objective of this dissertation is to design and create an ERP for the management and monitoring of ship call operations, it is the result of an in-depth study of an existing system during our internship at the Port Company Skikda.

We used the UML modeling language for the design, and for the programming we chose Odoo as an open-source software package with the Python language which respects the MVC design model, as well as PostgreSQL as a database management system.

Keywords: Skikda Port Company, ERP, Odoo, ship calls management, UML.

Table des matières

| | |
|--|----|
| Remerciement..... | 2 |
| Dédicace..... | 3 |
| Résumé..... | 4 |
| Table des matières | 6 |
| Table des figures | 9 |
| Liste des tableaux..... | 11 |
| Table des abréviation | 12 |
| Introduction générale | 13 |
| Chapitre1 : Présentation d'entreprise d'accueil. | 16 |
| 1. Introduction :..... | 16 |
| 2. Entreprise Portuaire de Skikda :..... | 16 |
| 2.1. Aperçu historique d'EPS : | 16 |
| 1962- 1971..... | 16 |
| 1971- 1982..... | 16 |
| 1982- 1988..... | 16 |
| 1888/89-2003..... | 17 |
| Année 1998 | 17 |
| 2.2. Présentation générale :..... | 18 |
| 2.3. L'organigramme de l'entreprise :..... | 20 |
| 2.3.1. Présentation de direction des systèmes | 22 |
| 2.3.2. Présentation de la direction capitainerie | 28 |
| 3. Etude d'existant : | 28 |
| 3.1. Présentation du système actuel :..... | 29 |
| 3.2. Problématique : | 29 |
| 3.3. Solution proposée :..... | 30 |
| 4. Conclusion : | 32 |
| Chapitre 02 : Présentation ERP/Odoo..... | 33 |
| 1. Introduction :..... | 33 |
| 2. C'est quoi un ERP ;..... | 33 |

| | |
|--|----|
| 2.1. Définition du ERP : | 33 |
| 2.2. Caractéristiques d'un ERP : | 34 |
| 2.3. Les composants d'un ERP : | 34 |
| 2.4. L'architecture d'un ERP : | 35 |
| 2.5. Type des ERP : | 36 |
| 2.6. Les Avantages d'un ERP : | 37 |
| 2.7. Les Inconvénients : | 38 |
| 2.8. Les différents environnements de travail d'un ERP : | 38 |
| 2.9. Objectif des ERP : | 39 |
| 2.10. Les ERP open source: | 40 |
| 3. C'est quoi Odoo : | 41 |
| 3.1. Aperçu historique : | 41 |
| 3.2. Définition Odoo : | 42 |
| 3.3. Les caractéristiques de Odoo : | 43 |
| 3.4. Modèle de conception de Odoo : | 44 |
| 3.5. Architecture du système Odoo : | 45 |
| 3.6. Les avantages du système Odoo : | 46 |
| 4. Conclusion : | 47 |
| Chapitre 03 : Analyse et conception | 48 |
| 1. Introduction : | 48 |
| 2. Analyse : | 48 |
| 2.1. Spécification des besoins : | 48 |
| 2.2. Les besoins fonctionnels : | 49 |
| 2.3. Les besoins non fonctionnels : | 49 |
| 3. Conception : | 50 |
| 3.1. Unified Modeling Language (UML) : | 50 |
| 3.2. Diagrammes de cas d'utilisation : | 52 |
| 3.3. Diagramme de classe : | 53 |
| 3.4. Diagrammes de Séquences : | 55 |
| 4. Conclusion | 60 |
| Chapitre 04 : Réalisation..... | 61 |
| 1. Introduction : | 61 |
| 2. Les langages et environnements de développement : | 61 |
| 2.1. Système d'exploitation Windows 11 : | 61 |
| 2.2. PostgreSQL : | 61 |

| | |
|---|----|
| 2.3. Python : | 62 |
| 2.4. PyCharm : | 62 |
| 2.5. Git/GitHub : | 63 |
| 2.6. XML : | 63 |
| 2.7. Odoo : | 63 |
| 3. Les étapes de création d'un module : | 65 |
| 3.1. Installation Odoo16 : | 65 |
| 3.2. Préparation : | 66 |
| 3.3. Exécuter Odoo : | 68 |
| 4. Configuration de Module : | 69 |
| 4.1. Création d'une base de données : | 69 |
| 4.2. Développement de module : | 78 |
| 4.3. Relations entre les attributs : | 85 |
| 5. Conclusion | 86 |
| Conclusion générale | 88 |
| Références | 90 |

Table des figures

| | |
|---|----|
| Figure 1 : Port mixte de Skikda (ancien port)..... | 17 |
| Figure 2 : Position géographique du port de Skikda identité de l'entreprise portuaire de Skikda. | 20 |
| Figure 3 : Organigramme général de l'EPS..... | 21 |
| Figure 4 : Organigramme du centre de système d'informations..... | 22 |
| Figure 5 : Organigramme du centre de système d'informations..... | 28 |
| Figure 6 : Page d'accueil du système SIGNE..... | 29 |
| Figure 7 : ERP logo..... | 33 |
| Figure 8 : organisation d'une entreprise avec et sans ERP..... | 39 |
| Figure 9 : Odoo logo..... | 41 |
| Figure 10 : Pattern Model-View-Controller. | 45 |
| Figure 11 : L'architecture du Système Odoo. | 46 |
| Figure 12 : Les différents types de diagramme. | 51 |
| Figure 13 : diagramme de cas UEREPS..... | 52 |
| Figure 14 : diagramme de cas Admin. | 53 |
| Figure 15 : Diagramme de classe..... | 54 |
| Figure 16 : diagramme de séquence Authentification..... | 56 |
| Figure 17 : diagramme de séquence faire une opération..... | 57 |
| Figure 18 : diagramme d'ajouter admin et utilisateur..... | 58 |
| Figure 19 : diagramme de séquence gestion de comptes. | 59 |
| Figure 20 : Structure d'un Module Odoo. | 64 |
| Figure 21 : Formulaire de création d'une base de données Odoo. | 70 |
| Figure 22 : L'existence de la base de données Odoo16..... | 70 |
| Figure 23 : L'implémentation de fichier manifest .py de notre module EPS. | 71 |
| Figure 24 : L'implémentation de fichier init .py de notre module EPS..... | 71 |
| Figure 25 : Formulaire des détails d'entreprise. | 72 |
| Figure 26 : Le module EPS sur Odoo..... | 72 |
| Figure 27 : Le module EPS sur Odoo..... | 73 |
| Figure 28 : création des admin..... | 73 |

| | |
|--|-----------|
| Figure 29 : page login admin. | 74 |
| Figure 30 : création d'un utilisateur gestionnaire. | 74 |
| Figure 31 : page login utilisateur. | 75 |
| Figure 32 : L'implémentation de fichier ir.model.access.csv dans notre app... | 76 |
| Figure 33 : Liste des droits d'accès de notre utilisateur. | 76 |
| Figure 34 : liste d'user et admin avec leurs droits. | 77 |
| Figure 35 : l'implémentation de fichier 'eps_security.xml'. | 77 |
| Figure 36 : Menu principale de l'application. | 78 |
| Figure 37 : Liste des escales. | 78 |
| Figure 38 : Formulaire des escales. | 79 |
| Figure 39 : liste view opération. | 80 |
| Figure 40 : formulaire opération. | 80 |
| Figure 41 : bouton pour supprimer ou archiver une opération dans le formulaire. | 81 |
| Figure 42 : bouton pour supprimer ou archiver une opération. | 81 |
| Figure 43 : Ajouter date dans Odoo. | 82 |
| Figure 44 : Ajouter heur dans Odoo. | 82 |
| Figure 45 : Ajouter code poste et la designation. | 83 |
| Figure 46 : Sélectionner code poste et la designation. | 83 |
| Figure 47 : Engistrement d'une opération. | 84 |
| Figure 48 : Ajouter une opération favorite. | 84 |
| Figure 49 : Buton télécharger les opérations. | 85 |
| Figure 50 : Un exemple de code many2one. | 86 |

Liste des tableaux

| | |
|---|-----------|
| Tableau 1 : historique des versions d'Odoo [14]..... | 42 |
| Tableau 2 : Les cas d'utilisations assurés par chaque acteur. | 49 |

Table des abréviations

EPS : Entreprise Portuaire de Skikda.

EPE : Entreprise Publique Économique.

SIE : Sûreté Interne d'Établissement.

SIGNE : Système Intégré de Gestion des Navires et Escales.

ERP : Entreprise Ressource Planning.

PGI : Progiciel de gestion intégré.

MVC : Modèle vue contrôleur.

UML : Unified Modeling Language.

CPN : Conférence de Placement des Navires.

EDI : Environnement de Développement Intégré.

XML : eXtensible Markup Language.

Introduction générale

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

Ces ERPs sont des outils de gestion et d'analyse permettant d'optimiser la diffusion des informations en interne, d'améliorer les processus de gestion et d'automatiser les tâches ce qui augmente énormément la réactivité des entreprises et leurs agilités fonctions s'avère de plus en plus complexe et difficile.

Pour cela, moi et mon équipe avons décidé de faire ce projet en collaboration avec l'Entreprise Portuaire de Skikda dans le cadre du développement économique et communication et dans le but de satisfaire les besoins de l'entreprise.

L'adoption d'un ERP est une décision judicieuse pour l'Entreprise Portuaire de Skikda, car cela permet de moderniser son système d'information et de relever les défis auxquels elle est confrontée. Le système précédent, qui n'a pas évolué en fonction des exigences de l'entreprise, a révélé ses limites en ce qui concerne l'efficacité, la traçabilité des opérations et la gestion des flux d'informations.

Il est essentiel pour l'EPS de passer à un ERP afin de faciliter la coordination entre les différents services, d'automatiser davantage les tâches et de gérer les ressources de manière plus efficace. Dans cette perspective, la mise en place

d'un ERP au sein de l'Entreprise Portuaire X marque un tournant crucial dans sa transition numérique.

Nous avons opté pour Odoo comme outil de création de ce système, car il offre de nombreux avantages importants qui répondent aux besoins particuliers de l'entreprise.

Odoo est un logiciel open source qui propose une variété de fonctionnalités modulaires qui abordent tous les aspects essentiels de la gestion d'entreprise, tels que la comptabilité, les ressources humaines, la logistique, et bien d'autres encore. En outre, sa parfaite intégration facilite la circulation des données et offre une vision globale des activités de l'entreprise. Odoo se démarque grâce à sa convivialité et à sa facilité d'utilisation. De plus, sa souplesse et son évolutivité permettront d'ajuster le système en fonction des besoins qui évoluent au fil du temps.

À la demande des dirigeants de l'entreprise portuaire, nous avons débuté l'implémentation de cet ERP en donnant la priorité à la gestion des escales. Cette tâche jugée essentielle par les dirigeants concerne la gestion complète des navires, depuis leur arrivée jusqu'à leur départ, et constitue le point de départ du Workflow du système d'information. Cette étape est interdépendante de toutes les autres étapes du processus, ce qui en fait un élément fondamental à mettre en place en premier, et dans cette partie, on a basé sur les opérations des escales en premier lieu.

En se concentrant sur cette gestion des escales en amont, nous nous assurons de créer une base solide pour la suite de l'implémentation de l'ERP, facilitant ainsi la coordination des opérations portuaires, la gestion des ressources, la planification des activités et l'amélioration générale de l'efficacité opérationnelle de l'entreprise portuaire.

Outre l'introduction et la conclusion générale, ce mémoire est composé de 4 Chapitres :

Chapitre 01: Présentation d'entreprise d'accueil (rapport de stage): une brève présentation de l'entreprise portuaire de Skikda (E.P.Skikda) ainsi que son système informatique actuel.

Dans le chapitre 02 nous explorons le concept des ERP (Enterprise Resource Planning) et nous nous concentrons spécifiquement sur Odoo.

Le chapitre 3 de ce mémoire se concentre sur la conception de notre système basé sur l'ERP Odoo en utilisant le langage de modélisation UML.

Dans le chapitre 4, nous détaillons l'implémentation de notre application. Nous expliquons les outils et langages que nous avons utilisés pour développer l'application. Nous présentons également quelques interfaces graphiques du logiciel réalisé durant ce projet.

Chapitre1 : Présentation d'entreprise d'accueil.

1. Introduction :

L'analyse de l'existant est une étape importante dans le cycle de vie d'un système, il s'agit de connaître la situation actuelle de l'organisation pour pouvoir porter un jugement juste. Ainsi, l'analyse du système existant doit nous fournir toute l'information nécessaire, afin d'établir une bonne conception et de proposer de bonnes solutions.

Dans ce chapitre, nous allons apporter une étude générale sur l'Entreprise Portuaire de Skikda (EPS) et ses besoins.

2. Entreprise Portuaire de Skikda : [1]

Dans cette partie nous allons aborder une présentation générale sur l'EPS :

2.1. Aperçu historique d'EPS :

Avant la création de l'entreprise portuaire de Skikda, le secteur portuaire est passé par Plusieurs phases d'organisation et de régimes administratifs :

- **1962- 1971** : Le port de Skikda était géré par la chambre de commerce et d'industrie
- **1971- 1982** : Création de l'office nationales ports (13 mai 1971) chargé de la Gestion, de l'exploitation et en ce qui la concerne du développement des ports.

Les Ports sont organisés en unités portuaires et exercent le monopole des opérations de Pilotage de lamanage et les prérogatives de police et de sécurité portuaire ;

- **1982- 1988** : Création de l'Entreprise portuaire de Skikda (Décret N°82-284 du 14 août (1982) issue du regroupement de l'office National des ports, de la société

Nationale de Manutention et du service remorquage détenu anciennement par la Compagnie Nationale Algérienne de Navigation. Entreprise socialiste à caractère économique, elle Assure le monopole des dites activités ainsi que les prérogatives de puissance publique De police et de sécurité portuaire. Cette gestion sous tutelle Ministère des transports S'achèvera avec l'avènement de la loi sur l'autonomie des entreprises N°88/01 DU 12 JANVIER 1988.

- **1888/89-2003** : Création de l'Entreprise publique Economique société par Actions (SPA) le 21 mars 1989 (date de passage à l'autonomie), au capital de 135 millions de DA. L'Entreprise conserve les mêmes prérogatives du régime Antérieur. Son portefeuille d'actions est détenu par la société de gestion des Participations des ports (SOGEPORIS).
- **Année 1998** : Réorganisation du secteur portuaire (Transition en cours) Avec la promulgation de la loi N° 98/05 portant code maritime consacrant la séparation Des missions de puissance publique des activités commerciales et du décret exécutif N° 99 – 200 DU 18 août 1999 portant de l'autorité portuaire de l'Est à Skikda.

La position géographique du port de Skikda : Le port de Skikda, est délimité par :
+ 36°.53' 9.93" Nord, + 6°.5' 22,26"Est (' : minute, " : seconde).

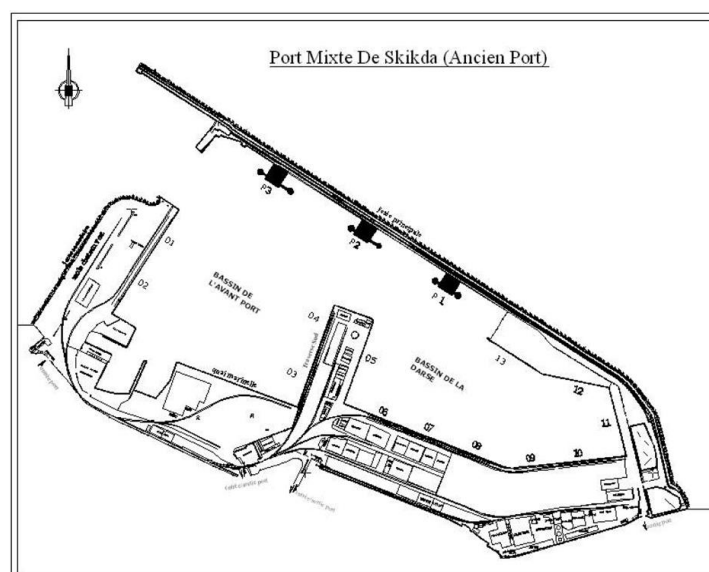


Figure 1 : Port mixte de Skikda (ancien port).

2.2. Présentation générale :

-Dénomination : entreprise portuaire de Skikda, par abréviation E.P. Skikda

-Forme juridique : entreprise publique économique, société par action régie par Leloir et règlements relatifs à l'autonomie des entreprises.

-Date de création : décret n°82-284 du 14 août et modification des statuts en société par actions en date du 21 mars 1989.

-Siège sociale : route îlot des chèvres. BP. 65. Skikda 21000. Algérie

Capitale sociale 900 millions de da détenu par un actionnaire unique, la société de gestion des participations de l'état « port » (SOGEPORT)

-Organisation :

-Organe de gestion de l'entreprise : l'entreprise est administrée par un conseil d'administrations composées de sept (07) membres : deux (02) membres de droit représentant les travailleurs.et cinq (05) membre représentant la société de gestion des participations de l'état « port » par abréviation SOGEPORT.

- Société organisée en direction générale mono – unité : activités et missions.la gestion et l'exploitation de l'outillage et des installations portuaires l'exercice des opérations de remorquage de pilotage, l'exercice des opérations d'acconage et de manutention, l'exercice des missions de police et de sécurité portuaire, l'exécution des travaux d'entretien, d'aménagement et de renouvellement de la superstructure portuaire l'élaboration, en relation avec les autres autorités concernées, de programmes de travaux d'entretien, d'aménagement et de création infrastructures portuaires.

-Compétence territoriale : la compétence territoriale de l'entreprise portuaire de Skikda s'étend aux ports suivants :

- Deux ports de commerce.
- Deux postes de chargement en mer.
- Trois ports de pêche.
- Port sec.

-Capacités portuaires de transit :

-Transit Marchandises générales : 2.500.000 tonnes/an ; 14 postes à quai avec un Total de 2.180 MI de quai.

-Transit hydrocarbures :

30.000.000 tonnes /an ; 10 appontements pour des navires. De 25 à 100.000 tonnes DWT, deux bouées shore de 300.000 Tonnes DWT.

-Surfaces commerciales :

Superficies commerciales couvertes (magasins, hangars modulaires, abris sous Auvents)

-Superficies commerciales terre-pleins :

479069 m2. Port sec 7ha.

-Les parcs engins de servitude et de manutention – transfert :

- **Parc naval et flottant :**

- a. Parc remorqueurs : 09 remorqueurs de 1700 à 3400 CV
- b. Parc vedettes de pilotage : 06 pilotins
- c. Parc canots d'amarrage : 09 canots, 02 pousseurs.
- d. Barrage de récupération : 02 barges (10 m3 et 24.6 m3)

- **Parc grues :**

- a. Grues électroniques de quai : 11 de 3 à 10 tonnes
- b. Grues automotrices : 15 de 30 à 350 tonnes

- **Parc chariots élévateurs :**

- a. Chariots élévateurs et moyen tonnage : 73 chariots de 1 à 10 tonnes
- b. Chariots élévateurs gros tonnages (hors engins manutention conteneurs) : 07 de 13 à 36 tonnes.
- c. Parc chariots élévateurs spécifique manutention conteneur : 11 de 10 à 45 Tonne.

- **Matérielle transfert :**

- a. Tracteurs portuaires : 09 de 25 à 32 tonnes ;
- b. Remorques portuaires : 19 de 40 à 120 tonnes.



Figure 2 : Position géographique du port de Skikda identité de l'entreprise portuaire de Skikda.

2.3. L'organigramme de l'entreprise :

L'EPS est organisée selon des directions fonctionnelles et opérationnelles dirigées par une direction générale qui est chargée de concevoir, coordonner et contrôler les actions liées à la gestion et au développement de l'entreprise. La figure suivante montre la structure générale de l'entreprise portuaire de Skikda :

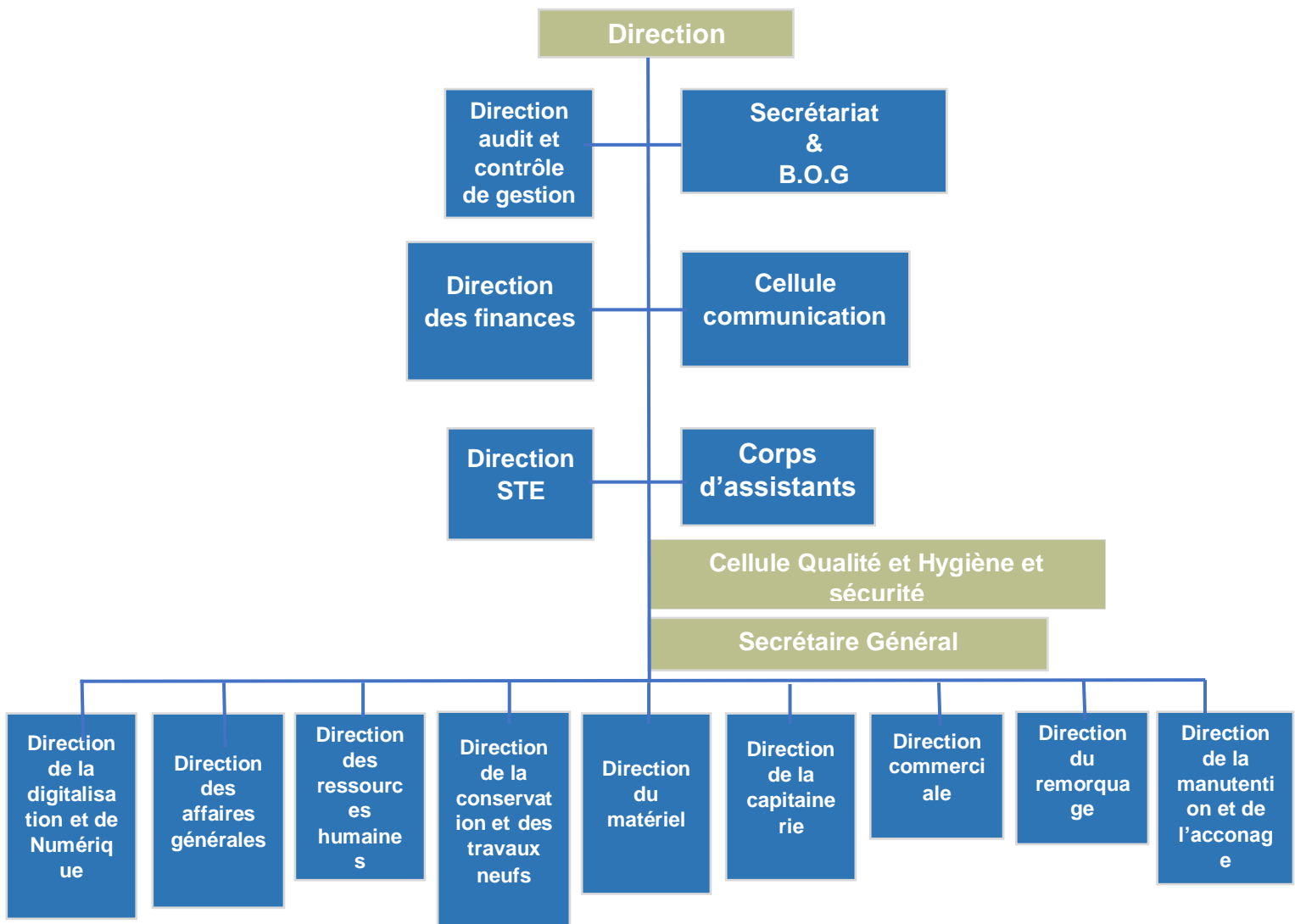


Figure 3 : Organigramme général de l'EPS.

2.3.1. Présentation de direction des systèmes d'information :

Le centre de système d'informations, se constitue de trois départements principaux : Le département « Infrastructures », le département « Programmes et méthodes » et le département « Génie logiciel » où se déroulera principalement notre stage. Chaque département regroupe un nombre de ressources humaines chargé de tâches spécifiques, La figure suivante illustre cette constitution :

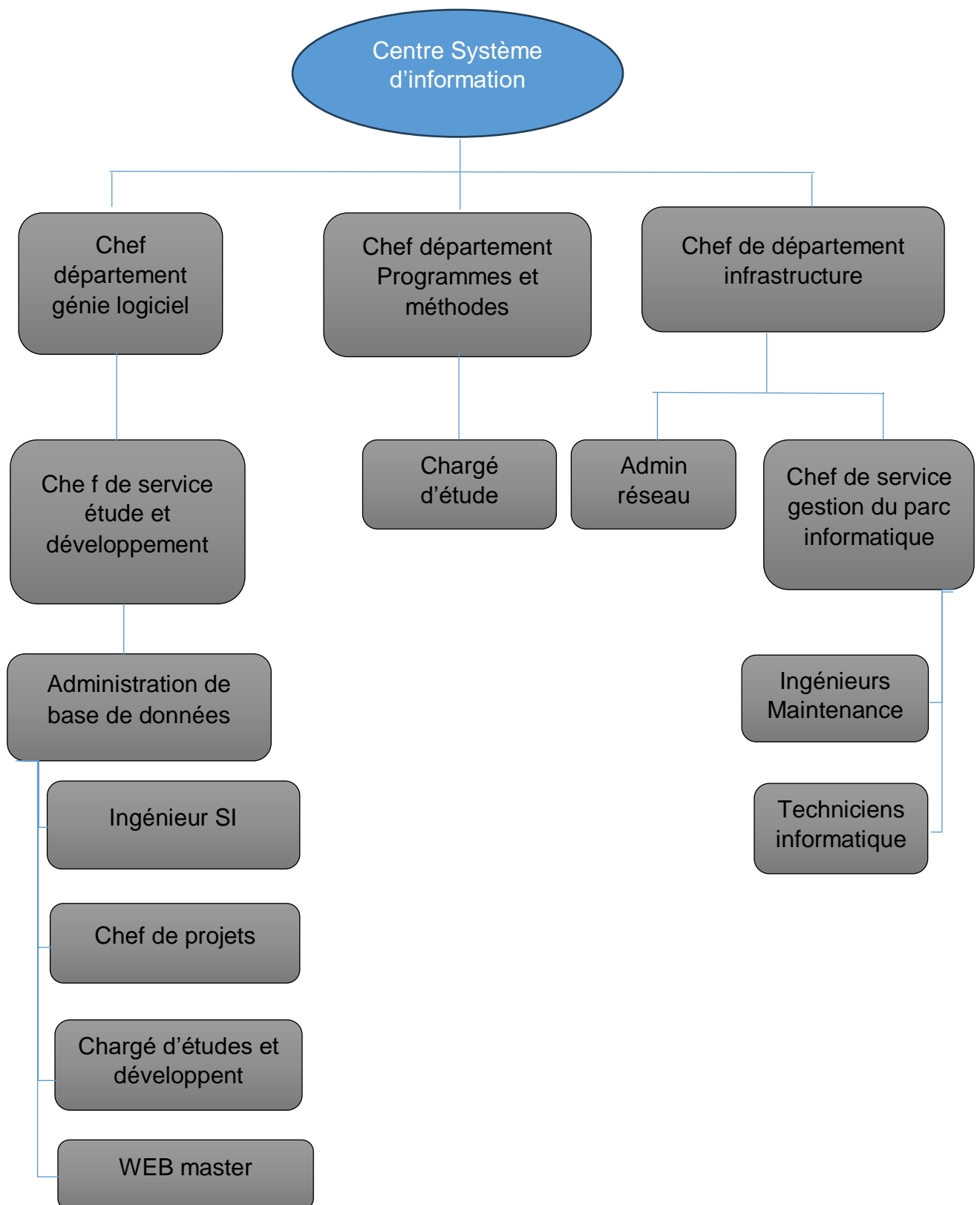


Figure 4 : Organigramme du centre de système d'informations.

1) Les missions du département informatique :

Le département a pour charges de :

*Elaborer et mettre en œuvre le plan directeur informatique.

*Entreprendre les travaux nécessaires qui s'inscrivent dans la démarche d'informatisation de l'ensemble du système d'information de l'Entreprise (logistique, gestion des ressources humaines, gestion financière et comptable, gestion des stocks etc.)

*Développer et mettre en œuvre les systèmes d'échange d'informations à travers notamment la réalisation, la gestion et la maintenance du réseau INTRANET de l'Entreprise.

*Etablir le plan informatique annuel ainsi que les budgets correspondants

*Superviser le travail du personnel du département en coordination avec les deux chefs de service.

*Effectuer un suivi régulier du plan d'action de la structure

*Orienter le développement du site web de l'entreprise en collaboration avec les parties concernées

*Gérer les dossiers d'acquisition de matériel et de logiciels ainsi que ceux concernant la fourniture et installation du réseau en collaboration avec ses subordonnés directs

*Communiquer en permanence avec le personnel sur les objectifs et résultats à atteindre

*Assurer le pilotage du processus « Gérer le système d'information ».

▪ Le Département informatique est composé de deux (02) services :

- **Service Maintenance du Réseau et des équipements informatiques.**

- **Service Maintenance et Développement du Système d'information.**

2) Service Maintenance du Réseau et des Equipements Informatiques :

Il est chargé de :

*L'entretien et la réparation du matériel informatique et l'installation de nouveaux équipements

*Formation des utilisateurs sur le bon fonctionnement des équipements dont ils ont la charge

*Mise en place de systèmes de sécurité permettant la protection des équipements, installations réseau etc.

*Animer et coordonner toutes les actions concourant au développement de l'outil informatique au sein de l'Entreprise.

*Veiller à la bonne gestion du courrier électronique à travers le site internet.

*Le Service Maintenance du Réseau et des Equipements informatiques est organisé en deux 02 sections :

- **Section Maintenance du Réseau Informatique :**

Ingénieur chargé du développement et de la maintenance du système d'information.

- **Section Maintenance des Equipements Informatiques :**

Technicien en maintenance informatique.

3) Service Maintenance et Développement du Système d'Information :

Il est chargé de :

- Etudier et entreprendre les démarches opérationnelles appropriées visant l'exploitation en réseau de l'échange d'informations avec les opérateurs portuaires.
- Etudes et développement de nouvelles applications informatiques selon les besoins exprimés par les utilisateurs.
- La mise en service et la maintenance des applications développées
- Assister les utilisateurs en vue d'aboutir à une exploitation efficace des applications développés en conformité avec les besoins de l'Entreprise.
- Entreprendre des actions d'encadrement et de formation des utilisateurs à chaque fois que le besoin est exprimé par les structures.
- Mise en place d'un planning de sauvegarde (logiciels et bases de données) et sécurité du système informatique de l'Entreprise

- Veiller au bon fonctionnement du réseau informatique de l'Entreprise et l'amélioration de ses performances
- S'assure que le plan de développement garantit une cohérence globale du système d'information et ce notamment via des interfaces et passerelles entre les différents modules ;
- Répartit le plan de charge sur les différents éléments du service et suit le plan d'action établi pour chacun en conséquence ;
- Effectue un contrôle régulier sur les différentes bases de données de l'entreprise et notamment en ce qui concerne le volet sécurité ;
- Développe elle-même des applicatifs de gestion et assure la maintenance des logiciels qu'elle a précédemment créés ;
- Superviser le travail du chargé de la maintenance du site web ;
- Le Service Maintenance et développement du Système d'information est organisé en :
 - **Section Maintenance et développement du Système d'Information**
:
 - Ingénieur chargé du développement et de la maintenance du système d'information.
 - **Section et développement du Système d'Information.**

2.3.2. Présentation de la direction capitainerie :

La direction capitainerie a pour mission d'exercer dans les limites du domaine public maritime et portuaire, la police et la sécurité des plans d'eau, des terres pleines et des installations, et ce dans le but d'exercer, en permanence, la conservation des ouvrages, la circulation et la sécurité des personnes et des biens, et de la contribution de l'élaboration de la politique générale en matière d'hygiène, sécurité et environnement ainsi que de la mise en œuvre des mesures de prévention. Elle est également responsable de l'ensemble des actions relatives à la navigation, du pilotage, du lamanage et du placement des navires. Elle incarne l'autorité portuaire. La direction de la capitainerie est composée de deux départements, et une station de pilotage, le directeur de la capitainerie assume les missions de PFSO.

- **PFSO** : Port Facilities Office est chargé de coordonner les activités de sûreté, sécurité au sens de la préservation du domaine portuaire et maritime (infrastructures, personnels, matériels, navires et marchandises) contre les risques d'intrusion, sabotage, vol, terrorisme et ce conformément aux dispositions du décret exécutif 04-418 du 20 décembre 2004, et du code ISPS. Le bureau de sûreté portuaire est placé sous l'autorité du PFSO, et comprend :
 - Un adjoint PFSO (chef de bureau sûreté portuaire).
 - Officiers de port chargés de la sûreté (assistants PFSO).

1) Département police et sécurité :

- Exécuter les décisions de la conférence de placement des navires.
- Assurer les relations de la capitainerie avec les commandants de navires et les consignataires.
- Assurer la gestion des équipements de radio télécommunication.
- Tenir à jour les dossiers navires.
- Gérer les statistiques en matière de mouvements particuliers de trafic.
- Tenir à jour les informations relatives aux infrastructures portuaires (tirant d'eau, longueur, des postes à quai, distance entre les bollards).
- Traiter les réclamations des amateurs ou leurs représentants.
- Veillez à l'application des règlements en matière de police et sécurité dans l'ensemble du domaine public maritime et portuaire : plan d'eau, terre-plein, docks-installations.
- Veiller au respect de la réglementation réagissent le transport et la manutention des marchandises dangereuses.
- Dresser des PV de constatation et verbaliser tout contrevenant.

Le personnel « officiers et surveillants de port » est assermenté, conformément aux dispositions du code maritime. Il comprend :

- Chefs de quart.
- Officiers de port.
- Officiers Radio.
- Section police et sécurité.
- Surveillants de port.

- Ambulanciers et accompagnateurs.

2) Département aides à la navigation :

Le département aide à la navigation est chargé des activités d'assistance à la navigation à l'entrée et sortie du port dans les bassins portuaires. Le département aide à la navigation est composé d'un service et d'une section environnement et lutte antipollution, et une station de pilotage.

- **Service lamanage :** sa mission est d'assurer la coordination et la supervision des opérations de lamanage et de largage des navires dans l'enceinte portuaire. Il comprend une section de lutte antipollution. Il assure aussi :
 - L'exploitation et l'armement des engins de servitude.
 - Suivi des engins de servitude en matière d'hygiène et de sécurité , ainsi que leurs entretiens internes et externes.
 - Elaboration du tableau de position mensuel des engins de servitude.
 - Veiller à la validité et le renouvellement des titres de navigation et de sécurité.
 - Assurer les services d'urgences telles que le déversement d'hydrocarbures ou d'autres incidents graves.
- **Station de pilotage :** la station de pilotage placée sous l'autorité du Directeur de la Capitainerie, elle comprend :
 - Un chef pilote, et son adjoint, le corps des pilotes maritimes, et fonctionne selon le régime de travail permanent.
 - Le pilotage maritime est obligatoire dans les limites administratives du port pour l'ensemble des navires.
 - Le pilotage est effectué par les pilotes dotés de qualifications et d'expériences professionnelles prévues par la réglementation.

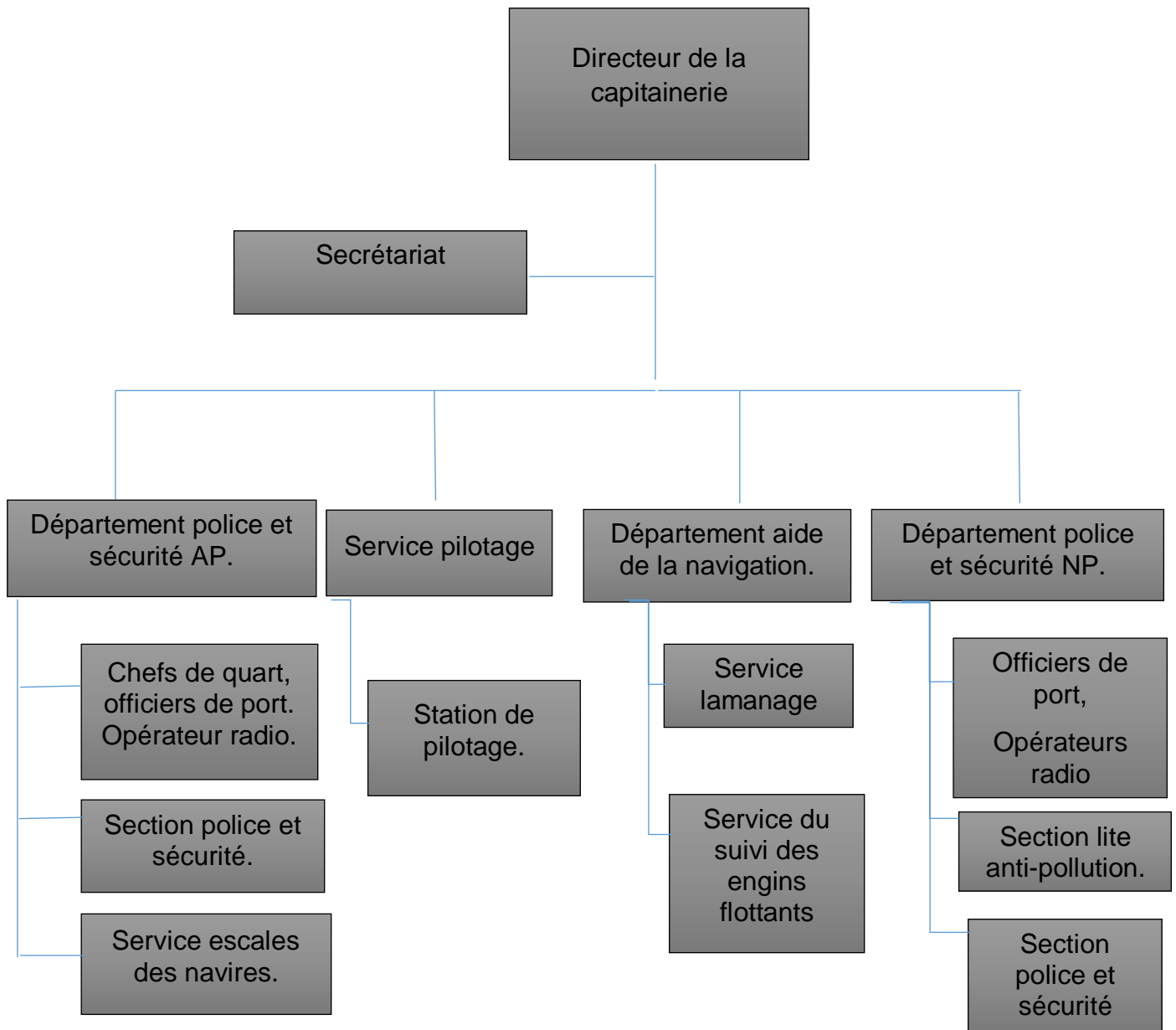


Figure 5 : Organigramme du centre de système d'informations.

3. Etude d'existant :

Environ 15 systèmes d'information sont utilisés par l'entreprise portuaire de Skikda pour diverses tâches et services, tels que SIGNE, Planetecom, BIGFinances, MEDSTATS, SIDOM et GPRPO.

Au sein de notre projet, notre objectif est d'analyser le système d'information SIGNE, de mettre en évidence ses lacunes et de proposer une solution qui répond aux nouvelles exigences des utilisateurs.

3.1. Présentation du système actuel :

Le Système Intégré de Gestion des Navires et Escales est un système d'information permettant la gestion des navires accostés au niveau du port et de toutes les activités liées aux prestations de services, offertes aux clients (importateurs, transitaires, consignataire, armateur etc.).

Le système SIGNE est basé sur une architecture client/serveur. Il est conçu et implémenté d'une façon modulaire, permettant ainsi de l'adapter aux nouveaux besoins des utilisateurs.

Il constitue un système sécurisé grâce à un verrouillage des accès.

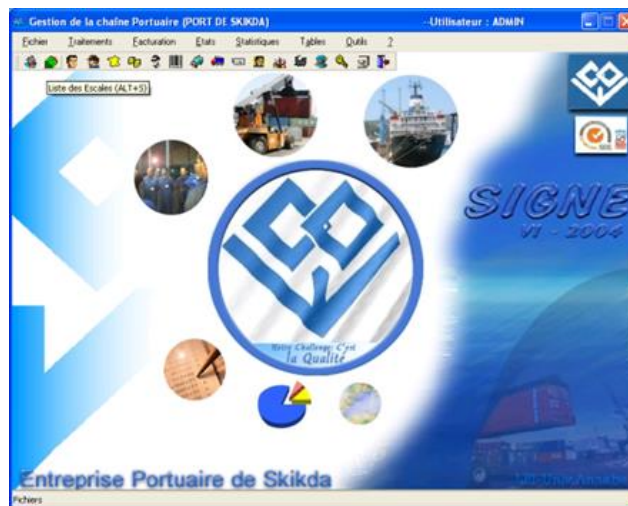


Figure 6 : Page d'accueil du système SIGNE.

3.2. Problématique :

Nous avons constaté un certain nombre de difficultés et d'incohérences sur le système existant tel que :

- Le partage et l'échange d'informations avec d'autres systèmes se font manuellement ou via des passerelles, ce qui fait perdre du temps et augmente le risque d'erreur.

- Absence d'interactions entre les systèmes ce qui provoque l'incohérence des données.
- La base de données (BDD) utilisée est incomplète en termes de relations entre les données.
- Manque des fonctionnalités dans les systèmes pour faciliter la manipulation.

3.3. Solution proposée :

Pour résoudre les problèmes posés par le système existant, nous avons proposé de concevoir et implémenter un ERP avec Odoo.

La solution projetée devra être en mesure de satisfaire certaines fonctionnalités de base :

- **Restriction des accès et personnalisation des menus** : L'accès aux différentes sessions sera contrôlé par l'identification de l'utilisateur (Login) assorti d'un mot de passe. Les menus et le mode d'accès aux sessions du progiciel seront personnalisables par utilisateur (lecture, écriture, suppression). Certaines données jugées confidentielles telles que les prix, les tarifs seront masqués à tous les utilisateurs non autorisés.
- **Import et export des données** : Le système permettra d'importer et d'exporter des données. L'import des données sera contrôlé pour interdire la mise en cause de l'intégrité des tables. Les imports pourront être réalisés à partir de fichiers Excel mis en forme. Des exports sur Excel seront facilement réalisables.
- **Editions et états personnalisés** : Des outils simples permettront d'effectuer des requêtes paramétrables dans les tables pour éditer des états (numériques ou graphiques), afficher des listes à l'écran ou exporter des données sur Excel. Des outils permettront aussi de personnaliser des éditions. Ces personnalisations pourront comporter des mises en page avec zones topologiques (première page, en tête, pied de page, saut de page, etc.) et permettront l'intégration de données issues de n'importe quel champ enregistré dans le système, sous réserve du respect de l'intégrité des requêtes effectuées au sein d'une même page. Des champs calculés (fonctions dépendantes d'autres champs) pourront également être mis en

œuvre. L'utilisation de ces outils sera à la portée d'utilisateurs spécialement formés, mais qui ne seront pas nécessairement des informaticiens spécialistes en programmation.

- **Tables et champs libres** : Dans les tables existantes, des "champs libres" seront paramétrables pour gérer des informations spécifiques. Par ailleurs, des "tables libres" seront paramétrables pour gérer des informations indépendantes.
- **Gestion du Workflow** : Pour chaque processus mettant en œuvre un enchaînement d'opérations, un Workflow pourra être paramétré qui décrira :
 - La nature des opérations,
 - Leur enchaînement,
 - Les prérequis et les attendus de l'opération,
 - Les utilisateurs concernés
 - Les délégations de signature
 - La validation (signature électronique) d'une opération générera un message d'invitation (email) à l'utilisateur suivant du processus. La validation de l'opération suivante sera conditionnée par la validation de la précédente.
- **Historique** : Toutes les saisies et transactions enregistrées feront l'objet d'un enregistrement historique comportant le login, la date, l'heure et la nature de la transaction. Certains champs (par ex. prix), à définir lors de la mise en œuvre, doivent garder l'historique des modifications/suppressions. Il sera possible d'interroger l'historique sur n'importe lequel des champs précédemment mentionnés.
- **Autres fonctionnalités** : Sans constituer une liste exhaustive, les fonctionnalités suivantes seront couvertes :
 - Multidevises
 - Multilingue : Le système doit supporter plusieurs langues applicables sur les éditions ;
 - Synchronisation des utilisateurs : La solution doit permettre une intégration des annuaires d'utilisateurs (ex : AD).
 - Gestion automatique des sauvegardes et mise en mode sûr en cas de coupure d'énergie de longue durée.

- Gestion simple des procédures de sauvegarde et de restauration.

4. Conclusion :

Au cours de ce chapitre, nous avons exposé la société portuaire de Skikda. Par la suite, nous avons examiné le système actuel utilisé par l'EPS pour la gestion des escales afin de mettre en évidence des limites et ses faiblesses. Notre objectif était de proposer une solution sous forme d'un ERP qui regroupe toutes les fonctionnalités impliquées dans la gestion des escales et inclus de nouvelles fonctionnalités essentielles.

Chapitre 02 :

Présentation ERP/Odoo

1. Introduction :

L'environnement dans lequel les entreprises évoluent a beaucoup changé ces dernières années en raison de la compétitivité sans cesse croissante à laquelle sont confrontées. Par conséquent, les organisateurs sont toujours sollicités à innover en matière d'organisation de leurs entreprises et doivent prendre des décisions intervenant à plusieurs niveaux dans des délais de plus en plus limités. Dans ce chapitre nous allons introduire les concepts de base indispensables à la compréhension de la notion des ERPs, ainsi que le framework Odoo.

2. C'est quoi un ERP ;



Figure 7 : ERP logo [2].

2.1. Définition du ERP :

Un progiciel de gestion intégrée (PGI) dit ERP (Enterprise Resource Planning) est défini comme : « Un ERP est un ensemble de moules applicatifs généralement signé par un même éditeur et travaillant en mode native sur une base de données

unique, au sens du terme » « Un ERP est un logiciel qui a pour objectif de fournir une colonne vertébrale informationnelle à un processus. Un processus est une suite de tâches ordonnées avec entrées / sorties bien définies. Cette notion de processus est fondamentale pour comprendre les impacts d'un tel système sur une entreprise. » Nous retenons de ces définitions que l'ERP est un logiciel paramétrable construit autour d'une base de données unique et couvrant la majorité des besoins des différentes fonctions de l'entreprise grâce à un ensemble de modules [2].

2.2. Caractéristiques d'un ERP : [3]

On définit par ERP un sous-ensemble du système d'information qui intègre les caractéristiques globales suivantes :

- Gestion effective de plusieurs domaines de l'entreprise par des modules intégrés sous des logiciels susceptibles d'assurer une collaboration des processus.
- Existence d'un référentiel unique des données. Le référentiel est défini comme étant l'ensemble des références des données, ainsi que les indications.
- Adaptations rapides aux règles de fonctionnement (professionnelles, légales ou résultant de l'organisation interne de l'entreprise et règles dictées par le marché); unicité d'administration du sous-système applicatif (les applications).
- Uniformisation des Interfaces Homme-Machine (IHM) : même ergonomie des écrans, mêmes boutons, même famille de barres menu, mêmes touches de fonctions et de raccourcis. Existence d'outils de développement ou de personnalisation de compléments applicatifs.

2.3. Les composants d'un ERP : [4]

L'ERP représente l'élément central (pivot) entre les quatre entités Clients, Employées, partenaires et Fournisseurs, et prend en compte les six composants fondamentaux qui sont : Ils sont tous articulés autour du référentiel de données de l'entreprise (Base de données) qui garantit l'unicité, un accès, un partage, une visibilité, une sécurisation, une consistance et une cohérence des informations.

2.4. L'architecture d'un ERP : [5]

Les ERP utilisent une architecture de type client/serveur. Ils comportent un nombre variable de modules fonctionnels partageant les mêmes données, Celles-ci sont contenues au sein d'une base de données centralisée et unique. Les modules communiquent et échangent en utilisant cette même base qui assure la cohérence de l'information. Les modules applicatifs d'un ERP recouvrent toutes les fonctionnalités de l'entreprise celle destiné au client comme du coté fournisseur mais aussi celles qui sont en relation avec les partenaires. L'ERP possède une structure organiser selon deux axes d'organisation :

2.4.1. L'axe horizontal :

Cet axe représente la classification des modules de l'ERP, selon Jean-Louis Laqueux un ERP est définit comme une application ayant au minimum les modules suivants :

- Gestion commerciale.
- GPAO (Gestion de Production Assistée par Ordinateur).
- Gestion comptable et financière.
- Ressources humaines.
- Gestion client.
- Gestion planning.

2.4.2. L'axe vertical :

Où l'ERP semble organisé en trois couches d'après Baglin et al (2005), qui sont :

- **Le niveau présentation** : il constitue l'interface utilisateur, et donc toute tâche intervenant dans la communication avec ce dernier, afin d'acquérir les données nécessaires au traitement et lui fournir les résultats qu'il espère.
- **Le niveau applications** : il correspond aux fonctions de traitement de l'information.
- **Le niveau Base de données** : contient les procédures qui gèrent les grands volumes de données que peut avoir une entreprise, et les différentes manipulations et opérations à utiliser sur ces données.

2.5. Type des ERP : [6]

Il existe 4 types d'ERP qui sont :

- **Généralistes** : Ce type d'ERP est un outil complet ayant des fonctionnalités basiques qui répondent à tous les besoins classiques des entreprises et il s'adapte à n'importe quel secteur d'activité. Ces ERP ont plusieurs paramétrages possibles que chaque entreprise peut adapter à ces besoins et il possède également plusieurs modules qui sont sélectionnables selon les besoins du métier. Le déploiement le plus utilisé dans ce type d'ERP est le cloud.
- **Spécialisés** : Ce genre d'ERP est intégralement pensé pour un métier particulier, ce sont des logiciels qui répondent parfaitement aux problèmes complexes et particuliers du secteur d'activité pour lesquels ils ont été développés, les fournisseurs choisissent un métier dans lequel ils sont spécialisés pour développer un logiciel ERP parfaitement adapté dans les moindres détails à celui-ci. Parmi les secteurs d'activité les plus couverts, on trouve la santé, la biologie, la logistique et le commerce, etc. La difficulté de ces ERP est de prévoir les changements et les évolutions futurs des entreprises du secteur d'activités choisi.
- **Open-source** : Ce sont des logiciels libres et leur utilisation n'impose pas l'acquisition d'une licence, le but de ces logiciels est de permettre la modification légalement et la rendre techniquement possible grâce notamment à la réutilisation du code source qui est à la disposition des utilisateurs pour les ajuster selon leurs besoins, en ajoutant ce qui manque et retranché ce qui alourdit.
- **En mode SAAS (Software As Service)** : Les ERP en mode SAAS (Software As Service) ne sont pas commercialisés sous forme de logiciels mais sous forme de services, la différence de ces ERP est qu'ils utilisent des serveurs distants et donc il n'est pas nécessaire d'avoir des serveurs puissants au sein de l'entreprise pour y installer les ERP et les manipuler. Les ERP en mode SAAS sont accessibles à partir de n'importe quel ordinateur, téléphone ou tablette ayant une connexion internet, ils sont évidemment déployés avec le type cloud.

2.6. Les Avantages d'un ERP : [7]

Les progiciels de gestion intégrés permettent à l'entreprise une meilleure maîtrise de ses activités de production, tout en se basant sur une méthode stratégique garantissant l'optimisation de l'utilisation des ressources humaines et matérielles à la fois.

Les objectifs d'un ERP se focalisent sur la réduction des coûts afin d'aboutir à la croissance de l'entreprise. Ainsi les ERP présentent plusieurs avantages :

- La facilité de la gestion des flux d'informations dans une entreprise.
- La cohérence et l'homogénéité des informations.
- La réduction des coûts d'exploitation et des goulots d'étranglement.
- L'intégrité et l'unicité du système d'information.
- Différents niveaux de sécurité pour éviter l'utilisation non autorisée des données.
- L'optimisation des processus de gestion.
- Fournit un outil de configuration ayant pour rôle l'adaptation des services aux besoins spécifiques de chaque utilisateur la réduction des erreurs et des délais coûteux.
- Une communication interne et externe facile.
- Minimisation des coûts d'interfaçage puisqu'ils ne contiennent pas d'interfaces entre les composants du progiciel.
- Minimisation des coûts en temps d'exécution grâce à la synchronisation des traitements.
- Minimisation des coûts de maintenance évolutive comme l'amélioration des fonctionnalités et l'évolution des règles de gestion, de même pour la maintenance corrective.
- L'augmentation de l'efficacité opérationnelle.
- La simplification de la planification stratégique.
- L'amélioration du service à la clientèle.
- Permet la portée d'une entreprise hors ses murs, jusqu'aux fournisseurs, aux clients et aux partenaires.

Malgré le taux d'avantages que fournit un ERP, n'empêche qu'il souffre de quelques inconvénients que nous allons présenter par la suite.

2.7. Les Inconvénients : [8]

Comme tout projet, les ERP souffrent de quelques problèmes du fait qu'ils sont difficiles et longs à mettre en œuvre ; ils sont aussi relativement rigides et délicats à modifier, Par conséquent les inconvénients qui surgissent sont les suivants :

- Il est plus une boîte à outil puisqu'il ne peut fonctionner sans qu'il soit personnalisé coût élevé quoi qu'il existe quelques-uns qui sont des logiciels libres, les coûts dans ce cas se réduisent dans le coût de formation des utilisateurs et le service assuré par le fournisseur.
- Le progiciel est parfois sous-utilisé lourdeur et rigidité de mise en œuvre, difficultés d'adaptation du personnel de l'entreprise à l'utilisation de l'ERP.
- Nécessité d'une bonne connaissance des processus de l'entreprise, nécessité d'une maintenance continue ce qui rend l'entreprise dépendante du fournisseur.

Le choix d'une solution ERP est souvent structurant pour l'entreprise et un changement vers un autre ERP peut être extrêmement lourd à gérer.

Donc, les systèmes ERP sont toutefois difficiles à gérer et à mettre en œuvre et peuvent même avoir des effets perturbateurs sur le fonctionnement d'une entreprise. Si le déploiement n'est pas proprement planifié et implémenté avec la vision et l'aide de professionnels, de telles initiatives peuvent entraîner une augmentation des coûts et un mécontentement général sans réels avantages pour l'organisation.

2.8. Les différents environnements de travail d'un ERP : [9]

Un ERP contient généralement trois environnements de travail :

- **Un environnement de développement** : qui permet d'adapter le progiciel standard à des besoins spécifiques de l'entreprise.
- **Un environnement de test** : dit encore environnement de recette qui permet de réaliser des simulations. Ces simulations permettent de tester de nouveaux paramétrages et de contrôler le fonctionnement correct du progiciel par rapport à un processus de gestion donné (une vente, un achat, une sortie de stock, etc.)

- **Un environnement de production** : qui correspond au progiciel utilisé par les gestionnaires de l'entreprise au quotidien.

2.9. Objectif des ERP : [10]

Les ERP visent plusieurs objectifs lors de son intégration au sein d'une entreprise, parmi lesquels nous citons :

- Optimiser la gestion des données et faciliter leur contrôle.
- Assurer l'intégration entre les différentes fonctions de l'entreprise.
- Contrôler la gestion grâce à une base de données unique.
- Améliorer les performances globales de l'entreprise.
- Clarifier les objectifs visés par l'entreprise.
- Fournir un accès facile aux données sans compromettre les exigences réglementaires et de sécurités.
- L'ERP améliore l'organisation interne de l'entreprise. Au lieu d'avoir plusieurs bases de données associées aux différents services l'ERP permet la centralisation des données en un seul endroit ce qui facilite la circulation de données et une base de travail unique engendrant ainsi, une meilleure compréhension des tâches de chaque service et une organisation fluide de travail.

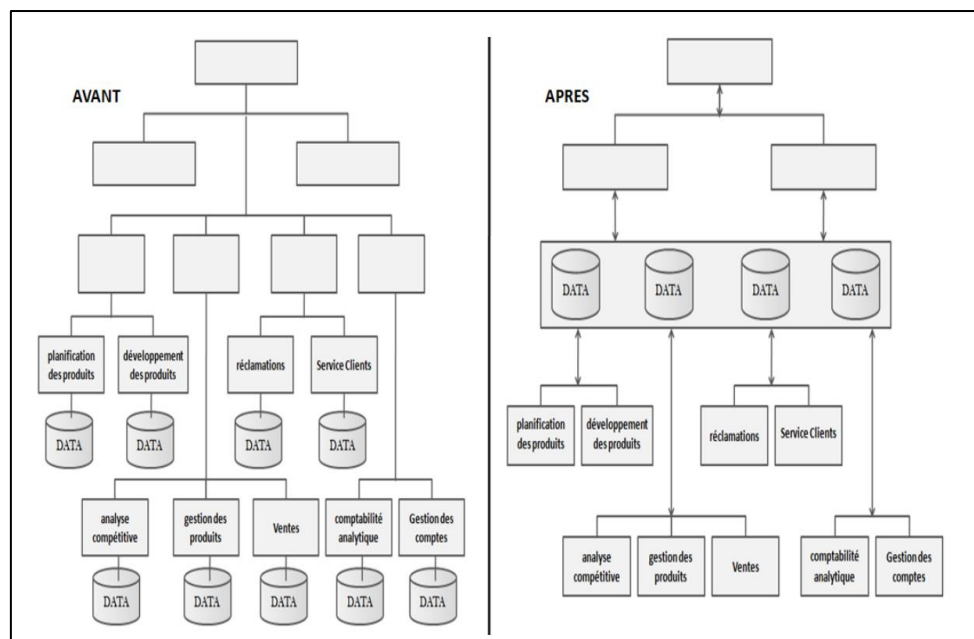


Figure 8 : organisation d'une entreprise avec et sans ERP.

2.10. Les ERP open source: [11]

Il existe un certain nombre de systèmes ERP open-source flexibles, riches en fonctionnalités et rentables. En voici quelque types:

- **ERPNext:** ERPNext est l'un de ces projets open source classiques; en fait, il a été présenté sur Opensource.com en 2014. Il a été conçu pour répondre à un besoin particulier, en l'occurrence le remplacement d'une implémentation ERP propriétaire grinçante et coûteuse.

ERPNext a été conçu pour les petites et moyennes entreprises. Il comprend des modules de comptabilité, de gestion des stocks, de vente, d'achat et de gestion de projet. Les applications qui composent ERPNext sont basées sur des formulaires: vous remplissez un ensemble de champs et laissez l'application faire le reste. L'ensemble de la suite est facile à utiliser.

- **Tryton:** Tryton's based on a EPR system called TinyERP and has been around since 2008. Over its lifetime, Tryton has grown both in popularity and flexibility.

Tryton is aimed at businesses of all sizes, and has a range of modules. Those include accounting, sales, invoicing, project management, shipping, analytics, and inventory management. Tryton's not all or not, though. The system is modular, so you can install only the modules your business needs. While the system is web based, there are desktop clients for Windows and MacOS.

- **Dolibarr:** Dolibarr offre une gestion de bout en bout pour les petites et les moyennes entreprises du suivi des factures, des contrats, des stocks, des commandes et des paiements à la gestion des documents et au soutien du système de point de vente électronique. Le tout est présenté dans une interface relativement claire.

En plus d'une démo en ligne, Dolibarr dispose également d'une boutique de modules complémentaires où vous pouvez acheter des logiciels qui étendent ses fonctionnalités. Vous pouvez consulter son code source sur GitHub ; il est sous licence GPLv3 ou toute version ultérieure.

- **Odoo:**

Odoo est une suite intégrée d'applications qui comprend des modules de gestion de projet, de facturation, de comptabilité, de gestion des stocks, de fabrication et d'achat. Ces modules peuvent communiquer entre eux pour échanger des informations de manière efficace et transparente.

Alors que l'ERP peut être complexe, Odoo le rend plus convivial grâce à une interface simple, presque spartiate. L'interface rappelle celle de Google Drive, avec seulement les fonctions dont vous avez besoin visibles. Vous pouvez essayer Odoo avant de vous inscrire.

3. C'est quoi Odoo :



Figure 9 : Odoo logo.[12]

3.1. Aperçu historique :

L'idée de Odoo a germé il y a maintenant 13 ans, en 2005. Fabien Pinckaers, le fondateur, émette rêve de devenir le leader sur le marché du management d'entreprise avec un logiciel de gestion complet Open Source. Il crée alors Tin ERP, qui était le premier nom d'Odoo.

Lorsqu'il stance, Tin ERP est un outil informatique qui permet d'accompagner le travail de gestion des entreprises [13].

| Historique des versions | | | |
|-------------------------|--------------|-------------------|--|
| Nom du logiciel | Versio n | Date de lancement | Changements significatifs |
| Tiny ERP | 1.0 | Février 2005 | Première publication |
| | 2.0 | Mars 2005 | |
| | 3.0 | Septembre 2005 | |
| | 4.0 | Décembre 2006 | |
| Open ERP | 6.0 | Octobre 2009 | Première publication sous AGPL, premier client Web |
| | 7.0 | Décembre 2012 | Client web en Ajax , Fin du support pour le client riche (GTK+) |
| Odoo | 8.0 | Septembre 2014 | Support pour le CMS : construction de site internet, e-commerce, point de vente, vente et business intelligence. |
| | 9.0 | Novembre 2015 | Première publication des éditions Community sous licence LGPLV3 et Enterprise sous licence propriétaire. |
| | 10.0 | Octobre 2016 | |
| | 11.0 | Octobre 2017 | Nouvelle plateforme Odoo.sh pour le développement |
| | 12.0 | Octobre 2018 | |
| | 13.0 | Octobre 2019 | |
| | 14.0 | Octobre 2020 | |
| | 15.0 | Octobre 2021 | |
| | 16.0 | Octobre 2022 | |
| 17.0 | Octobre 2023 | | |

Tableau 1 : historique des versions d'Odoo [14]

3.2. Définition Odoo : [15]

Odoo est un système de gestion d'entreprise open source et intégré qui offre une suite complète d'applications pour répondre aux besoins variés des entreprises. Il permet de gérer efficacement divers aspects de l'entreprise tels que la comptabilité, la gestion des ventes, des achats, des stocks, la gestion de projet, les ressources humaines, le marketing, le commerce électronique, et bien plus encore. Odoo est connu pour sa modularité, sa flexibilité et sa convivialité, ce qui en fait une solution populaire pour les entreprises de toutes tailles, des petites et moyennes entreprises aux grandes corporations. Son architecture modulaire permet aux utilisateurs de sélectionner les applications spécifiques dont ils ont besoin et de les intégrer facilement pour créer un système personnalisé répondant à leurs exigences métier.

3.3. Les caractéristiques de Odoo : [16]

Caractéristiques font d'Odoo une solution ERP populaire pour les entreprises de toutes tailles, car elle offre une flexibilité, une intégration et une personnalisation exceptionnelles pour répondre aux besoins diversifiés des entreprises. Pour ce la voici cinq caractéristiques clés d'Odoo :

3.3.1. Modularité :

Odoo est modulaire, ce qui signifie que vous pouvez choisir les modules spécifiques dont vous avez besoin pour votre entreprise. Vous n'êtes pas obligé d'installer l'ensemble du système, ce qui permet une personnalisation précise en fonction des besoins de votre entreprise. Les modules couvrent divers aspects de la gestion, tels que la comptabilité, la gestion des ventes, la gestion des stocks, les ressources humaines, le CRM (Customer Relationship Management), et bien plus encore.

3.3.2. Open source :

Odoo est une solution open source, ce qui signifie que son code source est accessible et personnalisable. Cela offre une grande flexibilité pour adapter le logiciel aux besoins spécifiques de votre entreprise sans dépendre d'un éditeur de logiciels propriétaires.

3.3.3. Intégration :

Odoo propose une intégration native entre ses modules. Cela signifie que les données circulent de manière transparente entre les différentes parties de votre entreprise, ce qui permet une meilleure coordination et une gestion plus efficace des opérations. Par exemple, les ventes, la comptabilité et le stock peuvent être liés pour une gestion plus fluide des commandes et de la facturation.

3.3.4. Convivialité :

Odoo se distingue par son interface utilisateur conviviale et moderne. L'interface est intuitive, ce qui facilite l'adoption par les utilisateurs finaux. De plus, il offre des fonctionnalités telles que la personnalisation des tableaux de bord pour permettre aux utilisateurs de voir rapidement les informations importantes.

3.3.5. Communauté et écosystème :

Odoo bénéficie d'une vaste communauté d'utilisateurs, de développeurs et de partenaires dans le monde entier. Cette communauté contribue à l'amélioration constante du logiciel, à la création de modules complémentaires et à la résolution de problèmes. Il existe également un écosystème de partenaires qui proposent des services de conseil, de personnalisation et de support pour Odoo.

3.4. Modèle de conception de Odoo : [17]

La conception de Odoo est basée sur le design pattern MVC (Model-View-Controller). Ce dernier est composé de trois parties :

3.4.1. Modèle (Model) :

- Les modèles dans Odoo sont des classes Python qui définissent la structure et le comportement des données dans la base de données. Chaque modèle correspond généralement à une table dans la base de données. Ils sont définis dans des fichiers Python avec une extension '.py' dans le répertoire 'models' de votre module.
- Les modèles définissent les champs de données, les contraintes, les relations avec d'autres modèles, les méthodes de calcul, etc.

3.4.2. Vue (View) :

- Les vues dans Odoo sont des fichiers XML qui définissent la présentation des données à l'utilisateur. Ils contrôlent la manière dont les données sont affichées et organisées dans l'interface utilisateur.
- Les vues sont définies dans des fichiers XML avec une extension '.xml' dans le répertoire views de votre module.
- Les vues peuvent inclure des éléments tels que des formulaires, des listes, des graphiques, des rapports, etc.

3.4.3. Contrôleur (Controller) :

- Dans Odoo, le contrôleur est principalement géré par le framework lui-même. Cependant, vous pouvez étendre le comportement standard en ajoutant des contrôleurs personnalisés.

- Les contrôleurs personnalisés peuvent être des classes Python qui étendent les contrôleurs existants ou des scripts JavaScript pour ajouter des fonctionnalités côté client.

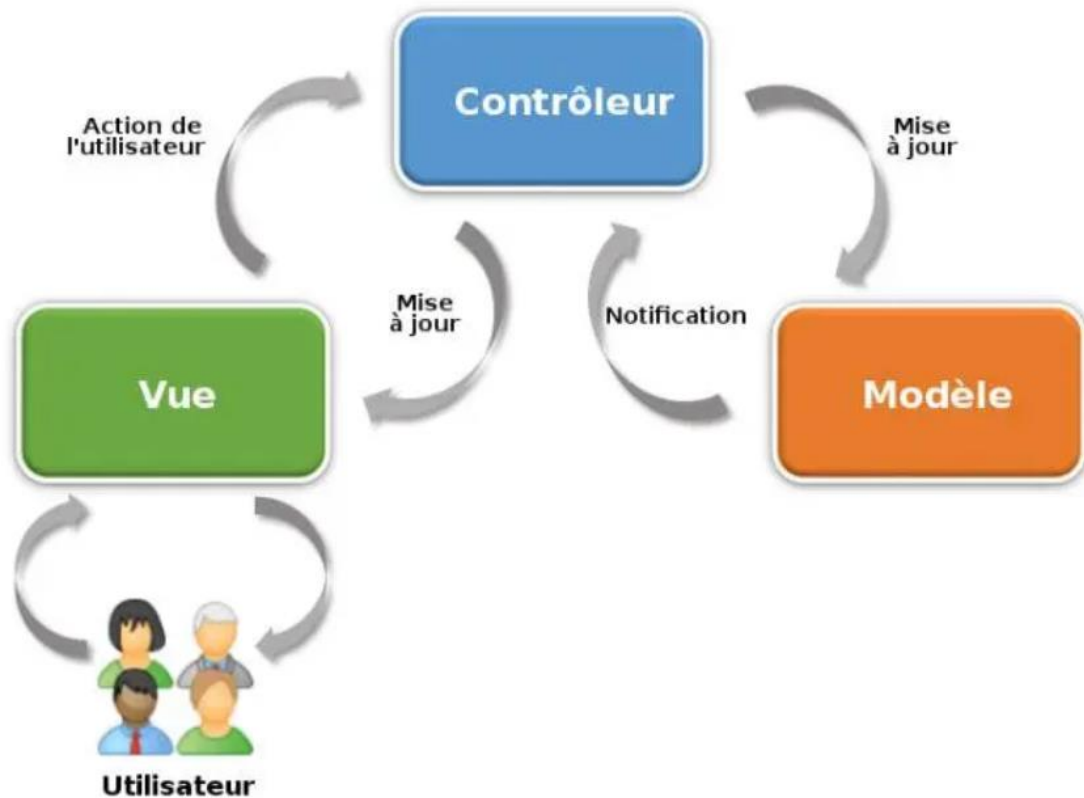


Figure 10 : Pattern Model-View-Controller.

3.5. Architecture du système Odoo : [18]

L'architecture du système Odoo est composée de 3 tiers (Voir la Figure) :

- Un serveur de base de données PostgreSQL : Odoo utilise une couche ORM (Object relational Mapping) pour la persistance de ses objets métier et la gestion de la base de données.
- Un serveur d'application : contenant les objets de gestion, le moteur de workflow, le générateur d'édition, etc.
- Un serveur de présentation : qui permet à l'utilisateur de se connecter à Odoo avec n'importe quel navigateur internet (Google chrome, Firefox, etc.).

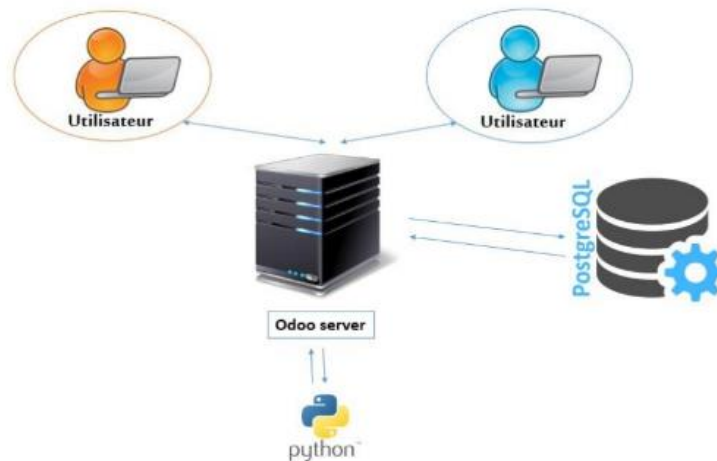


Figure 11 : L'architecture du Système Odoo.

3.6. Les avantages du système Odoo : [19]

Il y a plusieurs avantages pour choisir Odoo parmi eux :

- Périmètre fonctionnel inégalé avec ses nouveaux modules tous les mois.
- Conception très intelligente. Souvent jusqu'à 10 fois moins de code que les ERP en Java pour offrir les mêmes fonctionnalités.
- Interface web très compétitive et dynamique.
- Vrai ORM (Object-rational Mapping) qui fait le pont entre la base relationnelle et le code objet proche des spécifications fonctionnelles.
- Tout le data model et les méthodes métier sont nativement exposés en web services, c'est un gage d'interopérabilité facile.
- Moteur BPM intégré très efficace.
- Grande souplesse générale, notamment grâce à la scriptable des rapports.
- Les coûts d'intégration les plus faibles.

4. Conclusion :

Dans ce chapitre, nous avons fourni le contexte général du projet à travers la définition des ERPs et la définition du progiciel ODOO. Ce dernier est actuellement une alternative de marché libre aux systèmes de gestion des entreprises avec toutes leurs solutions conçues pour satisfaire le plus grand nombre de personnes, les entreprises en intervenant dans plusieurs domaines.

Chapitre 03 : Analyse et conception

1. Introduction :

Dans le but de développer un système cohérent et complet, une phase de spécification des besoins est jugée très importante. En effet, elle permet de recenser les fonctionnalités du système et de définir son architecture fonctionnelle et c'est l'objectif de ce chapitre.

2. Analyse :

Dans cette partie on va analyser les besoins de l'entreprise portuaire de Skikda en commençant par la spécification des besoins fonctionnels et non-fonctionnels.

2.1. Spécification des besoins :

Le but de cette partie est de décrire de manière précise et détaillée notre application. Pour ce faire, nous commencerons par établir les exigences opérationnelles que notre application finale doit satisfaire, puis nous déterminerons les utilisateurs potentiels de l'application ainsi que les contextes dans lesquels ils l'utiliseront. L'analyse commencera par la spécification des besoins opérationnels du système, en identifiant les parties prenantes et leurs diverses responsabilités ainsi que les scénarios associés.

En conclusion, nous exposerons les conclusions de cette analyse à travers des diagrammes de cas d'utilisation et plusieurs exemples de scénarios standards impliquant différents cas d'utilisation et leurs utilisateurs.

1) Définition des acteurs :

Notre application comporte deux acteurs :

- **Administrateur** : le service de département informatique qui a la possibilité de gérer l'application, d'ajouter, supprimer, ou de modifier une opération et d'assurer l'administration complète du système. Celui-ci se charge de contrôler l'accès des utilisateurs.
- **Utilisateur** : un employé qui a la possibilité d'authentifier, d'ajouter ou supprimer une opération et consulter l'état de l'opération.

2.2. Les besoins fonctionnels :

Les besoins fonctionnels représentent les actions que le système doit exécuter.

Selon notre application, nous avons identifié les besoins fonctionnels pour L'administrateur, d'une part et pour l'utilisateur (Employé), d'autre part.

| | |
|-------------------------|---|
| L'administrateur | <ul style="list-style-type: none">- Authentification.- Gérer utilisateur :<ul style="list-style-type: none">○ Ajouter un compte.○ Modifier un compte.○ Supprimer un compte.- Droit de création. |
| Utilisateur | <ul style="list-style-type: none">- Authentification.- Droit de création une escale ou des opérations.- Imprimer ou sauvegarder les données |

Tableau 2 : Les cas d'utilisations assurés par chaque acteur.

2.3. Les besoins non fonctionnels :

Dans notre application, on a défini les besoins fonctionnels qui sont indispensables pour le bon fonctionnement de notre système. Il existe d'autres besoins non fonctionnels qui sont exprimés en matière de performance et du type d'opération escale utilisé. Dans notre projet nous pouvons citer :

- **Ergonomie** : L'interface de l'application doit être simple et utilisable pour les utilisateurs pour qu'ils puissent l'exploiter.
- **L'authentification** : Notre application devra assurer une bonne sécurité, chaque utilisateur doit avoir un identifiant.
- **L'extensibilité** : Notre application doit être extensible qui signifie la possibilité d'ajout de nouvelles fonctionnalités ou modification de l'existant.
- **L'intégrité** : Notre application doit être intégrée à d'autres systèmes et utilisable par d'autres applications.
- **La disponibilité** : Notre application puisse fonctionner dans n'importe quel système d'exploitation.

3. Conception :

3.1. Unified Modeling Language (UML) :

3.1.1. Définition du UML :

L'UML est un langage graphique de modélisation informatique. Ce langage est constitué de diagrammes qui servent à visualiser et décrire la structure et le comportement des objets qui se trouvent dans un système. Il permet de présenter des systèmes logiciels complexes de manière plus simple et compréhensible qu'avec du code informatique. L'UML a des applications dans le développement logiciel, mais aussi dans l'industrie (pour modéliser les flux de processus par exemple), dans l'ingénierie ou le marketing [20].

3.1.2. L'objectif principal d'UML :

L'objectif principal d'UML est de définir une manière standard de visualiser la façon dont un système a été conçu et aide les équipes de projet à communiquer et à explorer les conceptions potentielles ainsi qu'à valider la conception architecturale du logiciel [21].

3.1.3. Type de diagrammes UML : [22]

Il y a deux catégories qui représente les types des diagrammes :

- **Diagrammes comportementaux** : il montre le comportement dynamique des objets dans un système UML propose les diagrammes comportementaux suivants :
 - Diagramme de cas d'utilisation.
 - Diagramme d'activité.
 - Diagramme de la machine d'état.
 - Les diagrammes d'interaction qui regroupent :
 - Diagramme de séquence.
 - Diagramme de communication.
 - Diagramme de timing.
 - Diagramme de présentation des interactions.
- **Diagrammes structurels** : qui montrent la structure statique du système. Cette catégorie regroupe les diagrammes suivants :
 - Diagramme de classe.
 - Diagramme des composants.
 - Diagramme de déploiement.
 - Diagramme d'objets.
 - Diagramme de package.
 - Diagramme de structure composite.
 - Diagramme de profil.

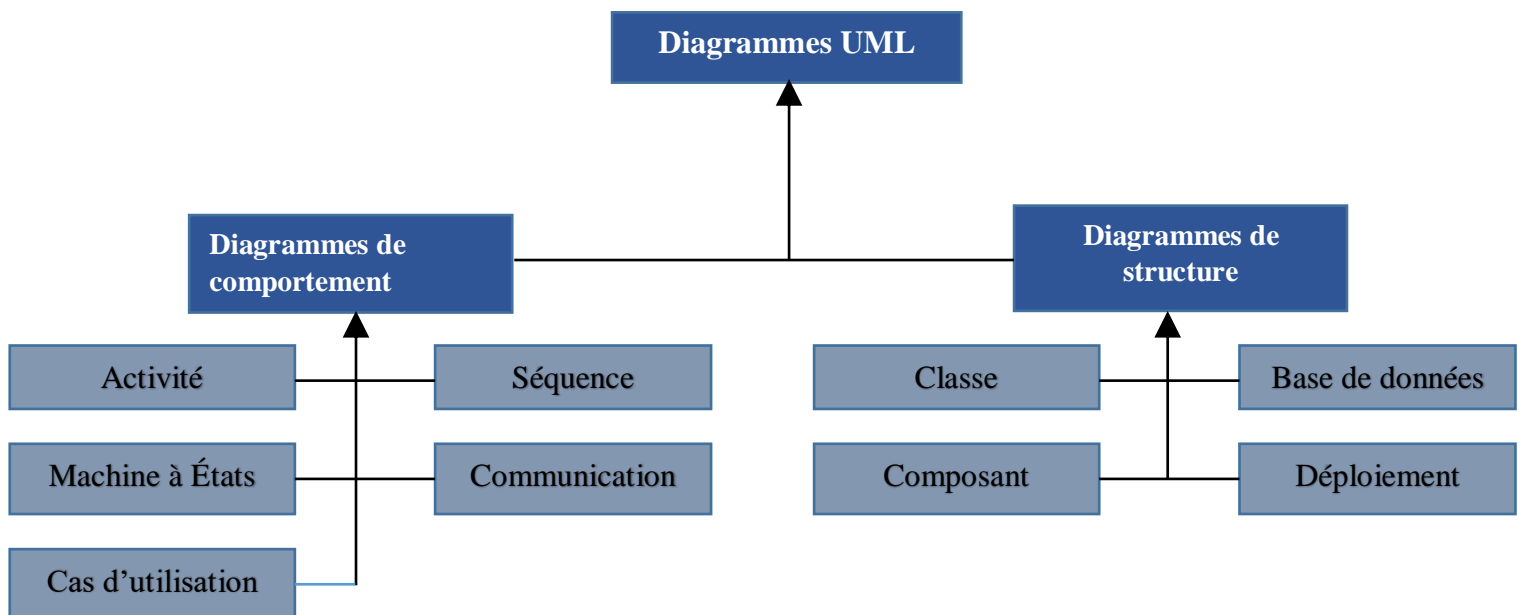


Figure 12 : Les différents types de diagramme [23].

3.2. 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. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases) [24].

- **Cas d'utilisation (UEREPS) :**

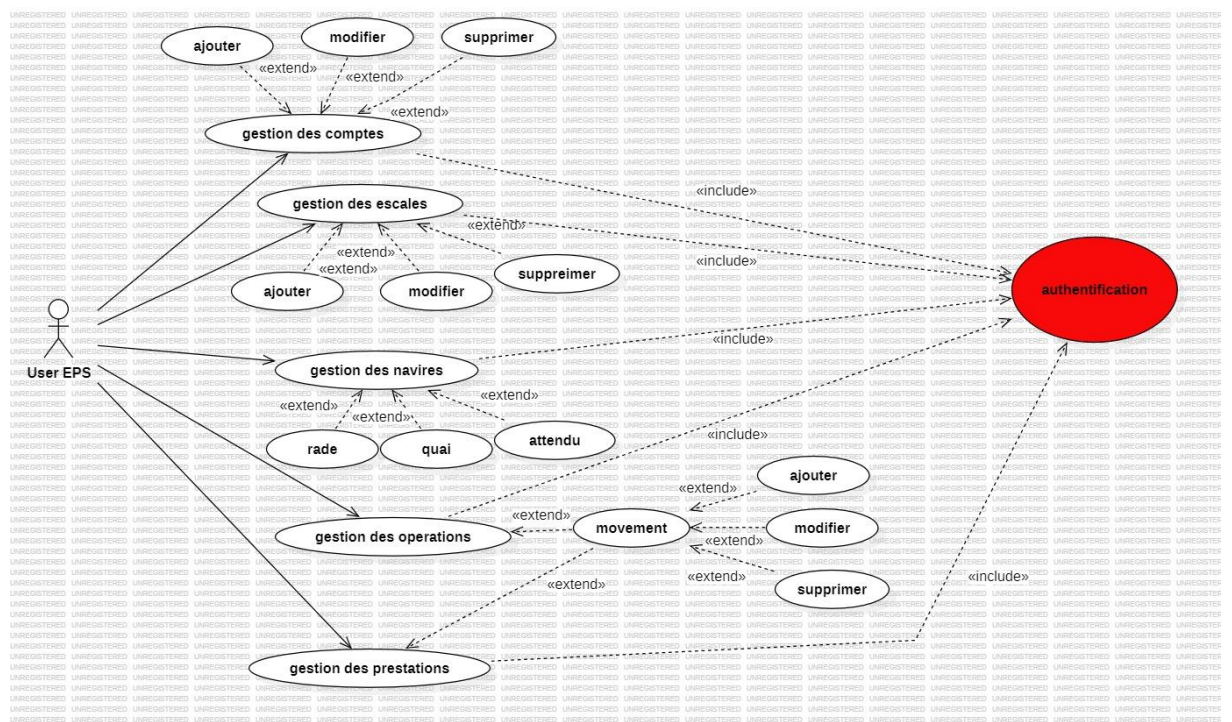


Figure 13 : diagramme de cas UEREPS.

- **Cas d'utilisateur (ADMIN) :**

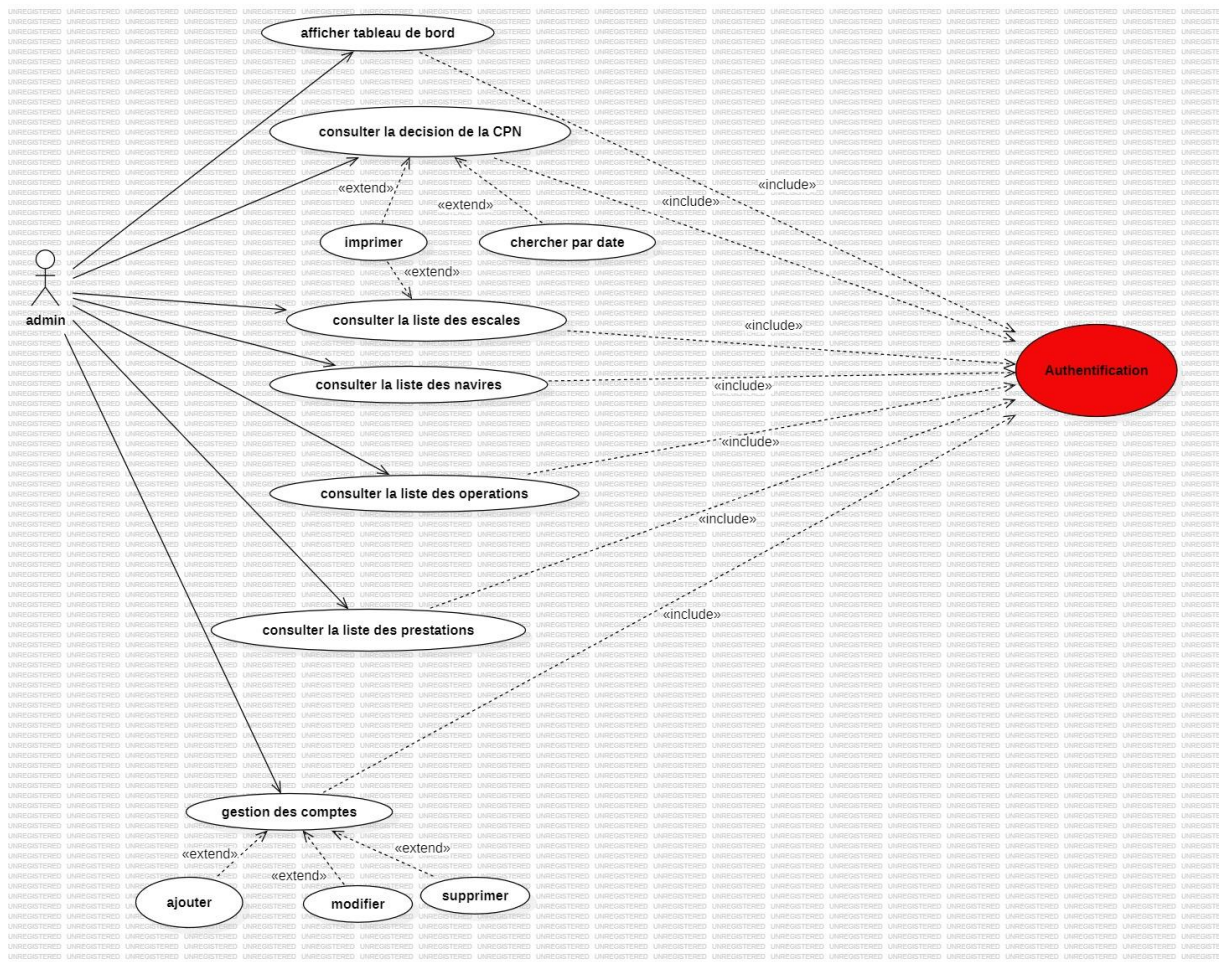


Figure 14 : diagramme de cas Admin.

3.3. Diagramme de classe :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une partition des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans

le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Les principaux éléments de cette vue statique sont les classes et leurs relations : association, généralisation et plusieurs types de dépendances, telles que la réalisation et l'utilisation.

La figure illustre le diagramme de classe de notre projet.

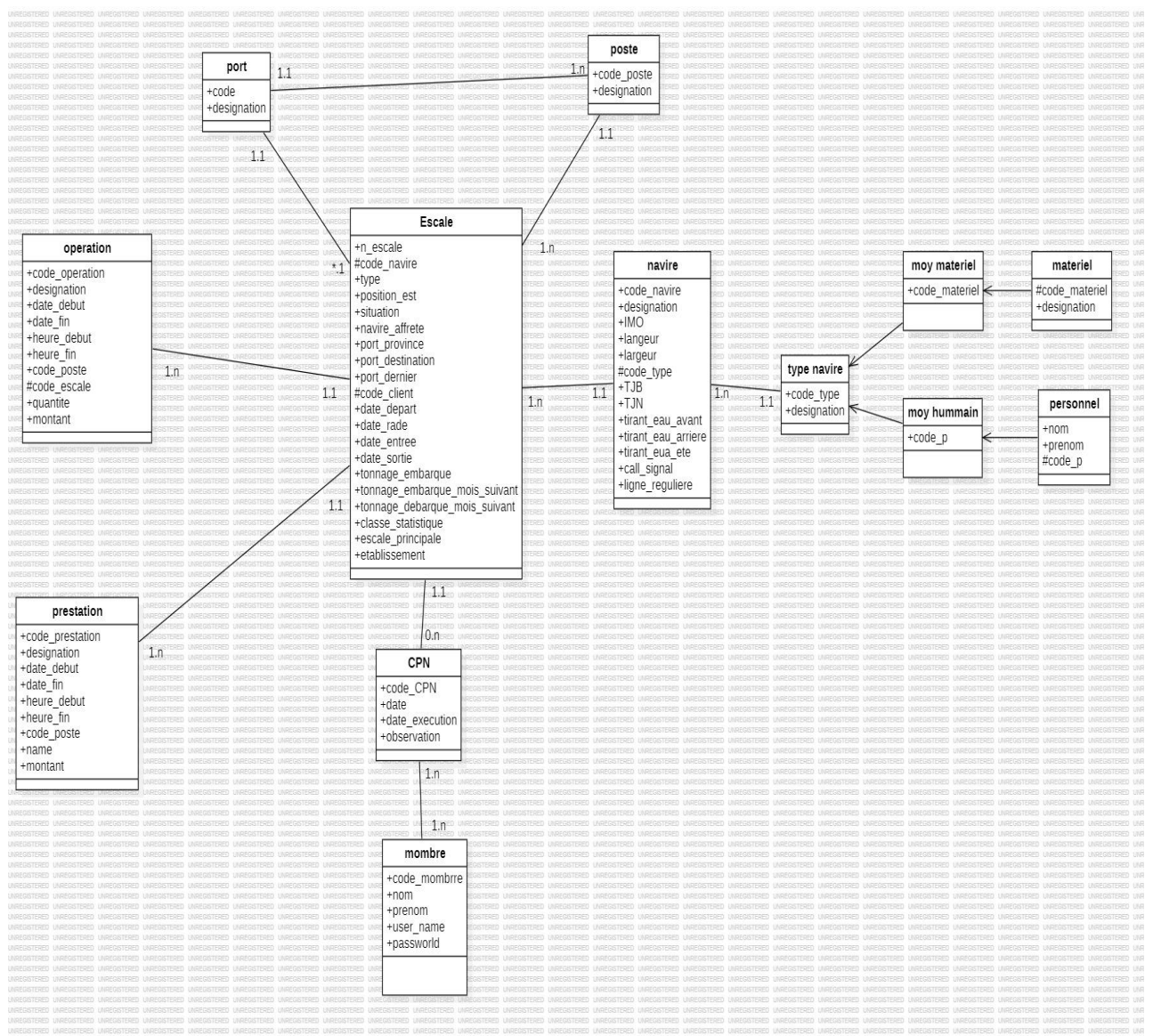


Figure 15 : Diagramme de classe.

3.3.1. Modèle Relationnel De Données :

- **T_port** : (#code_port, Désignation).
- **T_poste** : (#code_poste, Désignation).
- **T_Escale** : (#n_escale, #code_navire, type, position_est, situation, navire_affrete, port_province, port_destination, port_dernier, date_depart, date_rade, date_entree, date_sortie, tonnage_enbarque, tonnage_embarque_mois_suivant, tonnage_debarque_mois_suivant, classe_statique, escale_principale, etablissement).
- **T_navire** : (#code_navire, designation, IMO, longueur, largeur, #code_type, TJB, TJN, tirant_eau_avant, tirant_eau_arriere_tirant_eau_ete, call_signal, ligne_reguliere).
- **T_type_navire** : (#code_type, designation).
- **T_moy_materiel** : (#code_materiel).
- **T_materiel** : (#code_materiel, designation).
- **T_moy_hummain** : (#code_p).
- **T_personnel** : (nom, prenom, #code_p).
- **T_CPN** : (#code_CPN, date, date_execution, observation).
- **T_membre** : (#code_membre, nom, prenom, user_name, password).
- **T_operation** : (#code_operation, designation, date_debut, date_fin, geure_debut, heure_fin, #code_poste, #code_escale, quantite, montant).
- **T_prestation** : (#code_prestation, designation, date_debut, date_fin, heure_debut, heure_fin, #code_poste, name, montant).

3.4. Diagrammes de Séquences :

Afin de bénéficier des services de l'application, chaque utilisateur doit passer en premier lieu par un formulaire d'authentification où il saisit son identifiant et son mot de passe. Ces données sont transférées vers le contrôleur d'authentification qui sera chargé de vérifier l'existence de l'utilisateur et la crédibilité des informations saisies. Par la suite, si les informations entrées sont valides, l'utilisateur sera redirigé vers la page d'accueil, sinon un message d'erreur sera affiché.

3.4.1. Scénario d'authentification :

Lorsque l'utilisateur demande l'accès à l'application, il doit tout d'abord saisir son nom utilisateur et son mot de passe, le système va afficher le menu principal de l'application après vérification a la base de données, sinon il affiche un message d'erreur. Ce scénario est présenté par le diagramme de séquence de la figure 3-3.

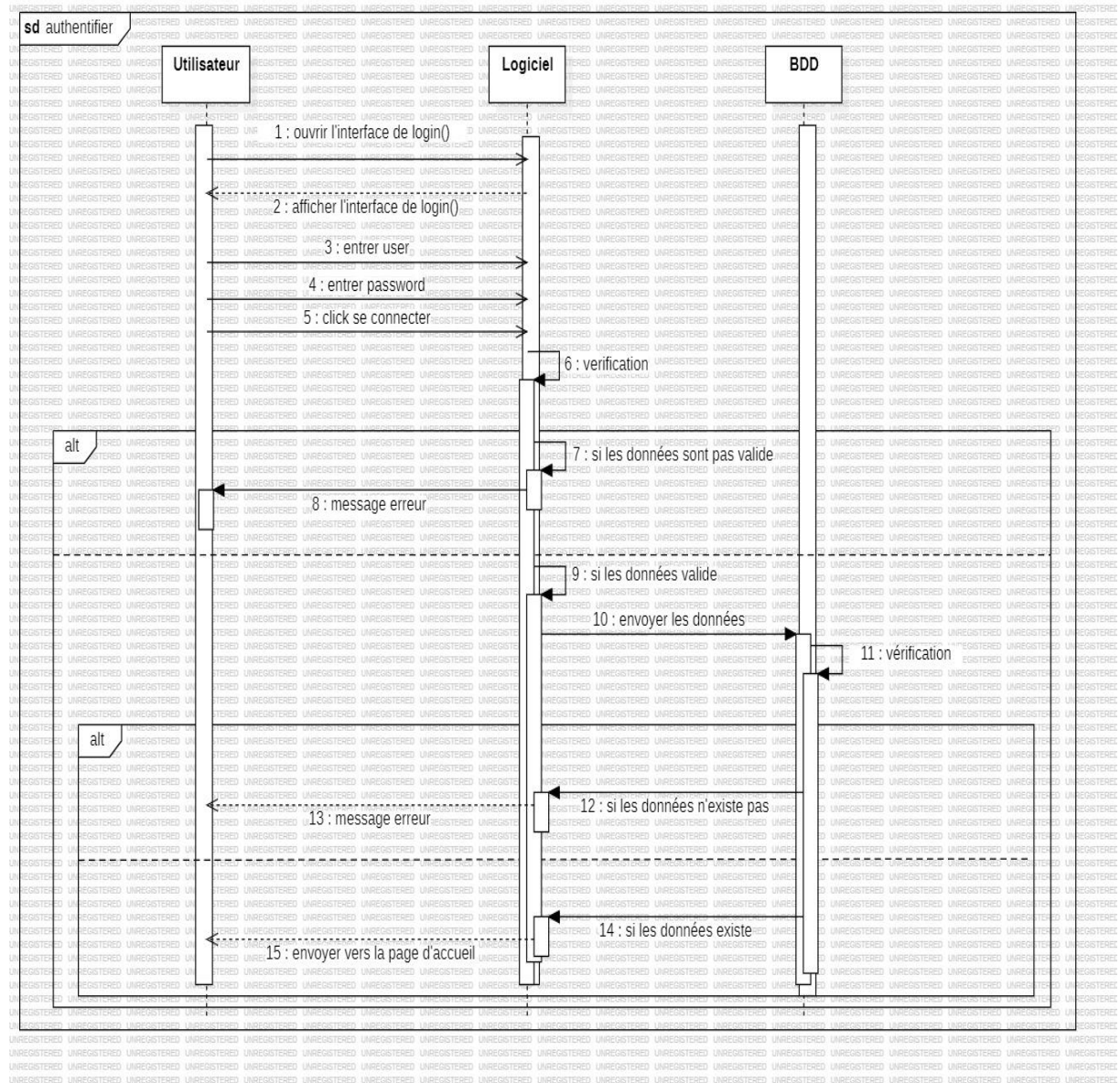


Figure 16 : diagramme de séquence Authentification.

3.4.2. Scénario d'établir une opération :

Lorsque l'utilisateur fait une opération, il doit tout d'abord choisir le type de l'opération, après il vas choisir la case 'new opération', le système va afficher le formulaire, après le gestionnaire va remplir le formulaire après le système va vérifier l'interface, sinon il affiche un message d'erreur. Ce scénario est présenté par le diagramme de séquence de la figure.

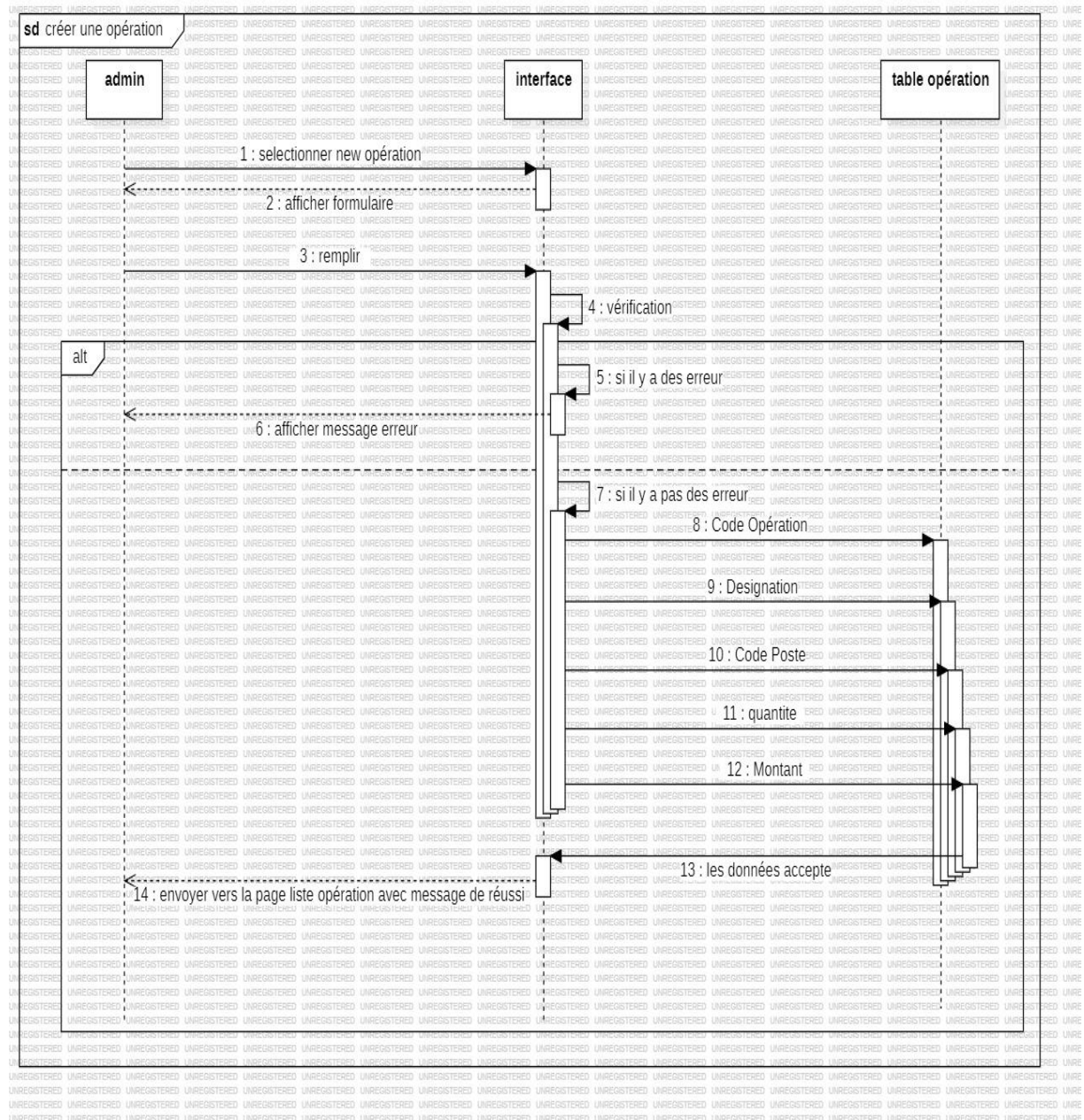


Figure 17 : diagramme de séquence faire une opération.

3.4.3. Scénario de gestion des comptes :

- **Ajouter un utilisateur :**

L'administrateur ouvre la fenêtre 'paramètre' dans l'application, il clique sur utilisateur après il saisit les informations d'un utilisateur (nom, prénom, le rôle, et les tâches de travail sur l'application...). Ensuite il clique sur le bouton ajouter. Si l'identifiant existe, un message d'erreur sera affiché.

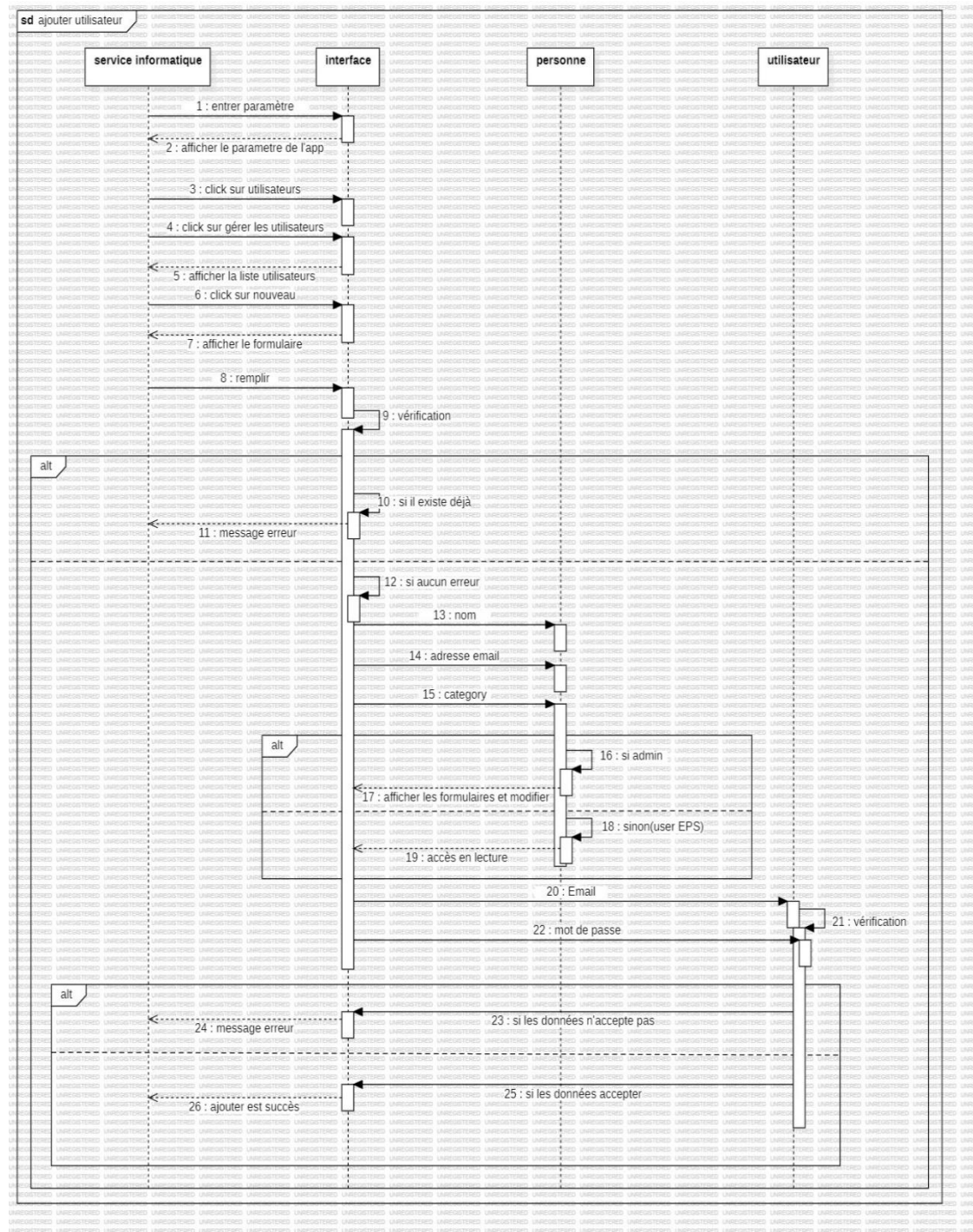


Figure 18 : diagramme d'ajouter admin et utilisateur.

• **Modifier et suppression d'utilisateur :**

- Action de modification d'un utilisateur : sélectionner l'employé à modifier à partir du formulaire, après vérification modifier les champs existants et sauvegarder, en cas d'erreur le message d'erreur sera affiché.
- Action suppression d'un utilisateur : sélectionner l'utilisateur à supprimer, puis appuyer sur un bouton de suppression, après confirmation les données seront supprimées et il va rediriger vers la page d'accueil.

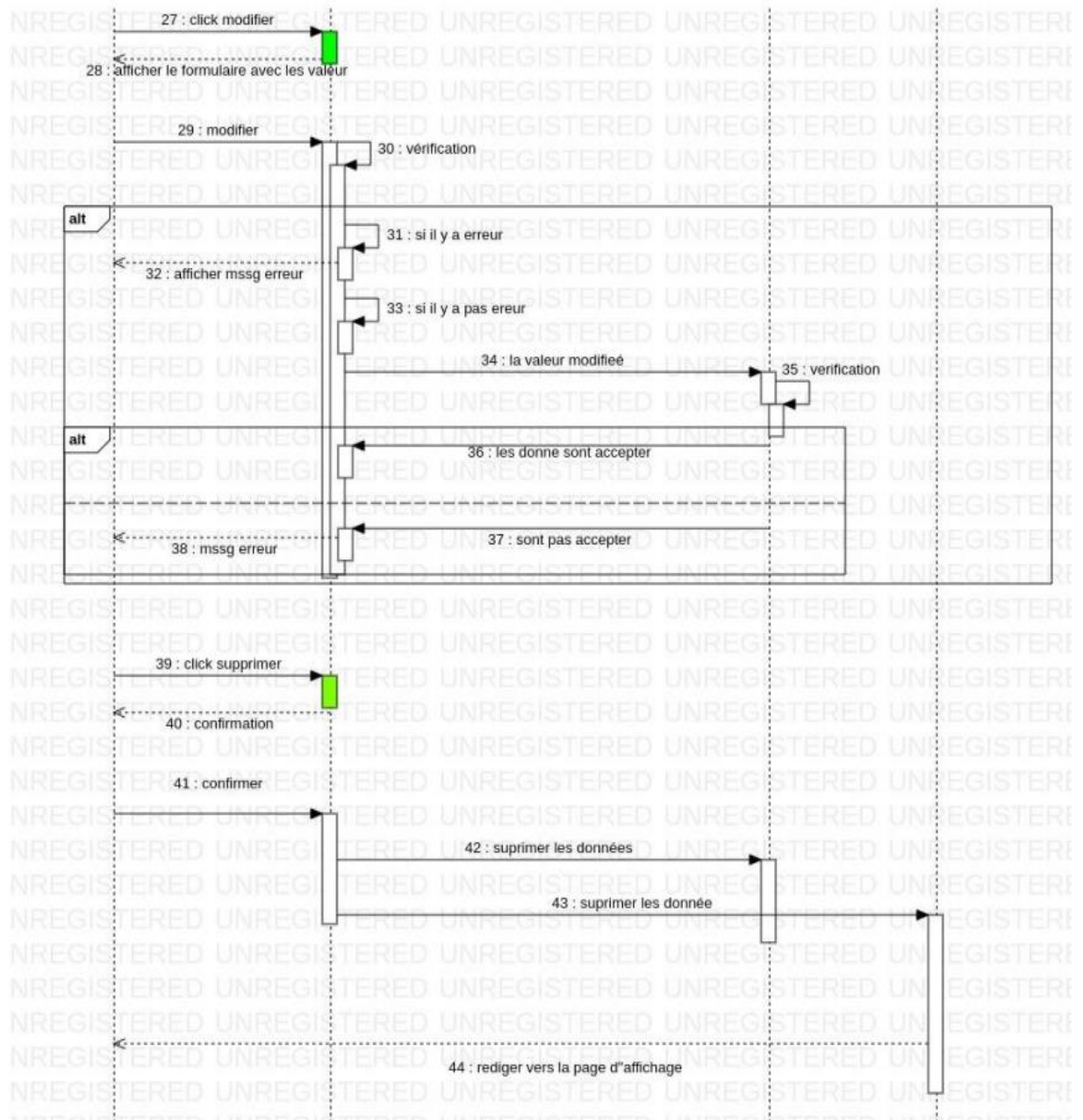


Figure 19 : diagramme de séquence gestion de comptes.

4. Conclusion

Pour conclure, la phase de conception est une phase indispensable pour modéliser un système. En fait, ladite conception de notre application enrichit la compréhension de la méthodologie de développement. Ce qui nous permettra dans le chapitre suivant d'aborder la phase de développement et simulation de notre application.

Chapitre 04 : Réalisation.

1. Introduction :

Ce chapitre constitue le dernier volet de cette partie du rapport. Il a pour objet d'exposer le travail réalisé. Nous présenterons les outils de développement utilisés, Nous montrerons ensuite les différents aperçus d'écran illustrant les fonctionnalités de notre application. Pour l'implémentation nous avons utilisé Microsoft Visual Studio.

2. Les langages et environnements de développement :

Dans cette partie nous allons présenter les outils utilisés pour réaliser ce projet :

2.1. Système d'exploitation Windows 11 :



Windows 11 est une version majeure du système d'exploitation Windows développé par Microsoft, exploitant le noyau Windows NT en version 10. Windows 11 a été annoncé lors du Microsoft Event le 24 juin 2021 [25].

2.2. 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 du

type BSD. Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB, MySQL et Firebird), ou propriétaires (comme Oracle, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, il est fondé sur une communauté mondiale de développeurs et d'entreprises [26].

2.3. Python :



Python est un langage de programmation informatique généraliste. Contrairement à HTML, CSS ou JavaScript, son usage n'est donc pas limité au développement web. Il peut être utilisé pour tout type de programmation et de développement logiciel. On s'en sert notamment pour le développement back end d'applications web ou mobile, et pour le développement de logiciels et d'applications pour PC. Il permet également d'écrire des scripts système, afin de créer des instructions pour un système informatique [27].

2.4. PyCharm :



Est un environnement de développement intégré (EDI) utilisé pour programmer en Python. Il permet l'analyse de code et contient un débogueur graphique. Il permet également la gestion des tests unitaires, l'intégration de logiciel de gestion de version [28].

2.5. Git/GitHub :



Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvald, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2[32]. Les objectifs que Git devait atteindre à sa conception étaient les suivants : être rapide, avoir un design simple, permettre le développement non-linéaire, être distribué et pouvoir gérer efficacement de gros projets [29].

2.6. XML :



XML (extensible Markup Language) est un langage simple et puissant de description et d'échange de documents structurés de n'importe quel domaine de données grâce à son extensibilité. Il décrit cette structure à l'aide d'un système de balises [30].

2.7. Odoo : [31]

L'idée fondamentale d'Odoo repose sur l'utilisation d'une architecture modulaire et autonome. Cela signifie qu'on peut améliorer facilement les modules déjà en place et également ajuster ou supprimer ceux dont on n'a pas besoin sans perturber l'intégralité du système.

Les modules intégrés à Odoo sont regroupés dans un répertoire appelé "/adonns". On peut aussi concevoir nos propres modules en créant un nouveau dossier personnalisé.

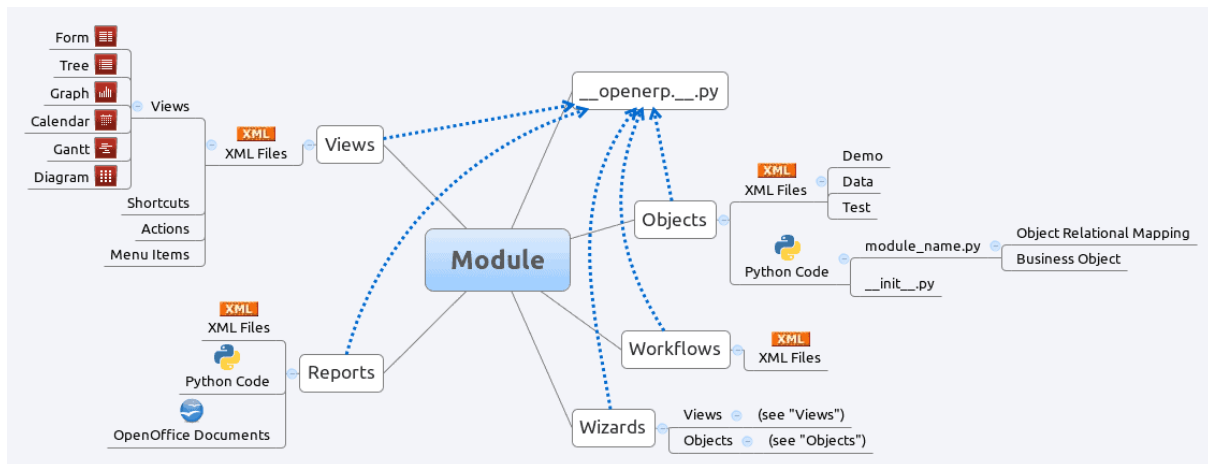


Figure 20 : Structure d'un Module Odoo.

Un module dans Odoo est symbolisé par un dossier qui contient des sous-dossiers et des fichiers suivant une convention spécifique.

Afin de créer et développer ses propres modules Odoo, il est essentiel de saisir la signification des fichiers et répertoires constituant l'architecture technique d'un module Odoo :

- **__init__.py** : Le fichier Python d'initialisation, crucial pour la création d'un module, renferme toutes les autres dépendances Python à importer.
- **__manifest__.py** : C'est le fichier de manifeste du module, où toutes les données relatives au module sont répertoriées, telles que son nom, sa description, sa version, l'auteur, le site web, ainsi que les fichiers de données, les démos et les sécurités.
- **Views** : C'est le dossier qui rassemble les fichiers XML décrivant la couche de présentation ou d'interface, comprenant notamment les vues liste (tree), les formulaires (form), les graphiques, les tableaux croisés dynamiques, les actions, les menus et les recherches.
- **Models** : C'est un répertoire qui contient tous les fichiers pythons du module, ce répertoire doit être déclaré dans le fichier d'initialisation `__init__.py`.
- **Static** : Ce répertoire renferme tous les fichiers liés au site web, tel qu'es fichiers image, js (javascript), css, font...
- **Data** : C'est un répertoire qui contient les données prédéfinies nécessaire lors de l'installation du module.

- **Démo** : Ce répertoire contient les données de démonstrations, qui vont être chargées lors de l'installation du module.
- **Security** : Répertoire contenant les fichiers de base définissant les règles de sécurités, les droits d'accès, les groupes, etc...

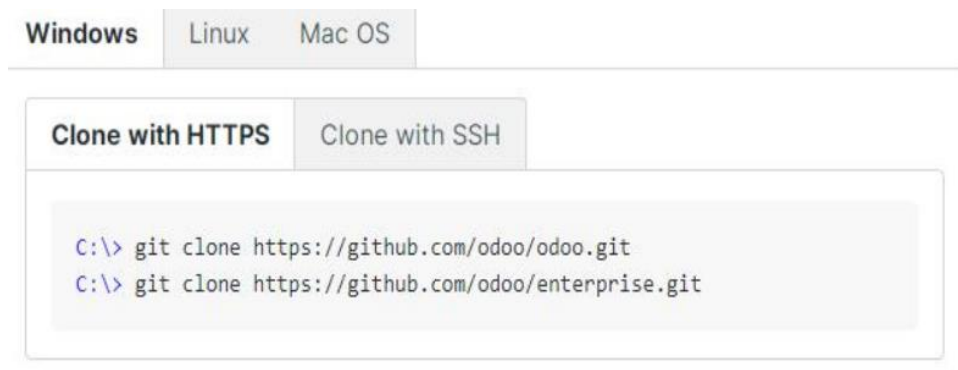
3. Les étapes de création d'un module : [32]

3.1. Installation Odoo16 :

L'installation par la source implique de ne pas installer Odoo de manière traditionnelle, mais plutôt de l'exécuter directement à partir de sa source. Cette approche peut être plus pratique pour les développeurs de modules, car la source d'Odoo est plus facilement accessible que lors d'une installation via un package. Elle offre une flexibilité accrue pour démarrer et arrêter Odoo, contrairement aux services configurés lors d'une installation via un package, et permet de contourner les paramètres en utilisant des options en ligne de commande sans avoir à modifier un fichier de configuration.

Enfin, cela permet un meilleur contrôle sur la configuration du système et facilite la conservation et l'exécution de plusieurs versions d'Odoo simultanément.

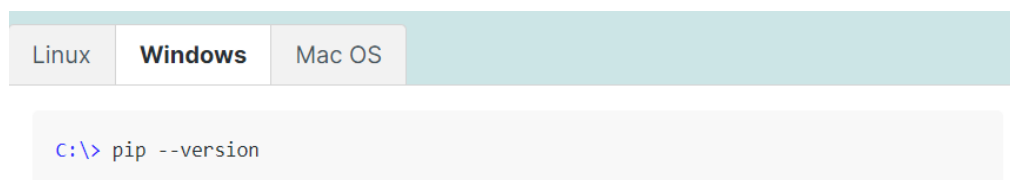
- **Récupérer les sources** : Il existe deux façons d'obtenir le code source d'Odoo sous forme d'archive Zip ou via Git. Nous avons opté pour l'utilisation de Git.
 - **Git** : La procédure suivante nécessite que Git soit installé sur votre appareil et que vous ayez une connaissance de base des commandes Git. Pour cloner un dépôt Git, vous devez choisir entre le clonage avec HTTPS ou SSH. Si vous ne connaissez pas la différence entre les deux, la meilleure option est probablement HTTPS. Toutefois, si vous suivez le tutoriel pour les développeurs ou prévoyez de contribuer au code source d'Odoo, SSH serait plus approprié.



3.2. Préparation :

- **Python :**

Pour faire fonctionner Odoo, vous aurez besoin de Python 3.7 ou d'une version ultérieure. Vous pouvez télécharger la dernière version de Python 3 à partir de la page de téléchargement de Python. Pendant l'installation, assurez-vous de cocher l'option "Ajouter Python 3 au chemin", puis cliquez sur "Personnaliser l'installation" et assurez-vous également de cocher pip. Cela vous permettra d'installer facilement les packages Python nécessaires pour Odoo.



- **PostgreSQL :**

Odoo utilise PostgreSQL comme système de gestion de base de données. Vous pouvez télécharger et installer PostgreSQL à partir du site officiel. Assurez-vous de télécharger une version prise en charge, recommandée à partir de la version 12.0. Par défaut, PostgreSQL crée un utilisateur appelé "postgres", mais Odoo restreint l'accès à la connexion en tant que "PostgreSQL".

Voici les étapes pour créer un nouvel utilisateur PostgreSQL :

- A. Ajoutez le répertoire bin de PostgreSQL (par défaut : C:\Program Files\PostgreSQL\<version>\bin) à votre PATH.
- B. Créez un utilisateur PostgreSQL avec un mot de passe en utilisant le guide pg Admin :

- Ouvrez **pgAdmin**.
- Double-cliquez sur le serveur pour créer une connexion.
- Sélectionnez **Objet** ▶ **Créer** ▶ **Rôle Login/Groupe**.
- Saisissez le nom d'utilisateur dans le champ **Nom de rôle** (Odo).
(Odo).
- Ouvrez l'onglet Définition et saisissez le mot de passe (Odo), cliquez ensuite sur **Sauvegarder**.
- Ouvrez l'onglet **Privilèges** et définissez **Peut se connecter ?** sur Oui et **Créer une base de données ?** sur oui.

- **Dépendances :**

Avant d'installer les dépendances, assurez-vous de télécharger et d'installer les **Build Tools** pour **Visual Studio**. Lorsque vous êtes invité, sélectionnez "**C++_build_tools**" dans l'onglet "Charges de travail" et procédez à l'installation. Les dépendances d'Odo sont répertoriées dans le fichier

requirements.txt situé à la racine du répertoire de la communauté Odo.

Dirigez-vous vers le répertoire d'installation d'Odo Community (CommunityPath) et exécutez la commande **pip** sur le fichier requirements.txt dans un terminal avec des **privilèges d'administration** :

```
C:\> cd \CommunityPath
C:\> pip install setuptools wheel
C:\> pip install -r requirements.txt
```

Pour les langues utilisant une interface de droite à gauche (comme l'arabe ou l'hébreu), vous aurez besoin du package rtlcss :

- A. Télécharger et installer nodejs.
- B. Installer rtlcss :

```
C:\> npm install -g rtlcss
```

- C. Éditez la variable PATH de l'environnement système pour ajouter le fichier où se trouve rtlcss.cmd :
C:\Users\AppData\Roaming\npm\.

3.3. Exécuter Odoo :

Une fois que toutes les dépendances sont installées, Odoo peut être démarré en exécutant Odoo-bin, l'interface en ligne de commande du serveur. Elle se trouve à la racine du répertoire de la communauté Odoo. Pour paramétrer le serveur, vous avez le choix entre définir des paramètres en ligne de commande ou via un fichier de configuration.

Les ajustements les plus fréquents comprennent :

- Identifiants utilisateur et mot de passe PostgreSQL.
- Adaptation des chemins d'accès pour répondre à vos besoins spécifiques, au-delà des emplacements par défaut, pour charger vos propres modules.

Une façon typique d'exécuter le serveur serait :

Linux**Windows**Mac OS

```
C:\> cd CommunityPath/  
C:\> python odoo-bin -r dbuser -w dbpassword --addons-path=addons -d mydb
```

Where `CommunityPath` is the path of the Odoo Community installation, `dbuser` is the PostgreSQL login, `dbpassword` is the PostgreSQL password, and `mydb` is the name of the PostgreSQL database.

Si le serveur à démarrer, Accédez à **http://localhost:8069** dans votre navigateur web et connectez-vous en utilisant le compte administrateur de la base de données : Utilisez "Admin" pour l'email et de nouveau "Admin" pour le

mot de passe. Voilà, vous venez de vous connecter à votre propre base de données Odoo.

4. Configuration de Module :

Avant de créer un module on devra d'abord commencer par la création d'une base de données sur Odoo, ensuite en va créer les comptes nécessaires à chaque utilisateur en affectant certain rôles et paramétrage.

4.1. Création d'une base de données :

Les applications Odoo reposent sur le modèle de conception MVC, comme mentionné précédemment, pour segmenter les technologies appropriées pour chaque partie de ce modèle. Pour la gestion des données, Odoo utilise le SGBDR (robuste et libre système de gestion de base de données relationnelle) PostgreSQL, avec une couche ORM pour sécuriser et faciliter l'accès aux données stockées, simplifiant ainsi les différentes requêtes sur les données. La création d'une base de données sur Odoo est effectuée après authentification avec l'utilisateur Admin par défaut.

La figure suivante présente le formulaire de création d'une base de données Odoo :

The screenshot shows a 'Create Database' dialog box with the following fields and values:

- Master Password: [Redacted]
- Database Name: EPS
- Email: fadybouchelit
- Password: [Redacted]
- Phone number: [Empty]
- Language: English (US)
- Country: Algeria
- Demo data:

At the bottom right, there is a blue 'Continue' button. Below the fields, there is a note: 'To enhance your experience, some data may be sent to Odoo online services. See our [Privacy Policy](#).'

Figure 21 : Formulaire de création d'une base de données Odoo.

La figure suivante démontre l'existence de la base de données Odoo16 dans l'instance PostgreSQL.

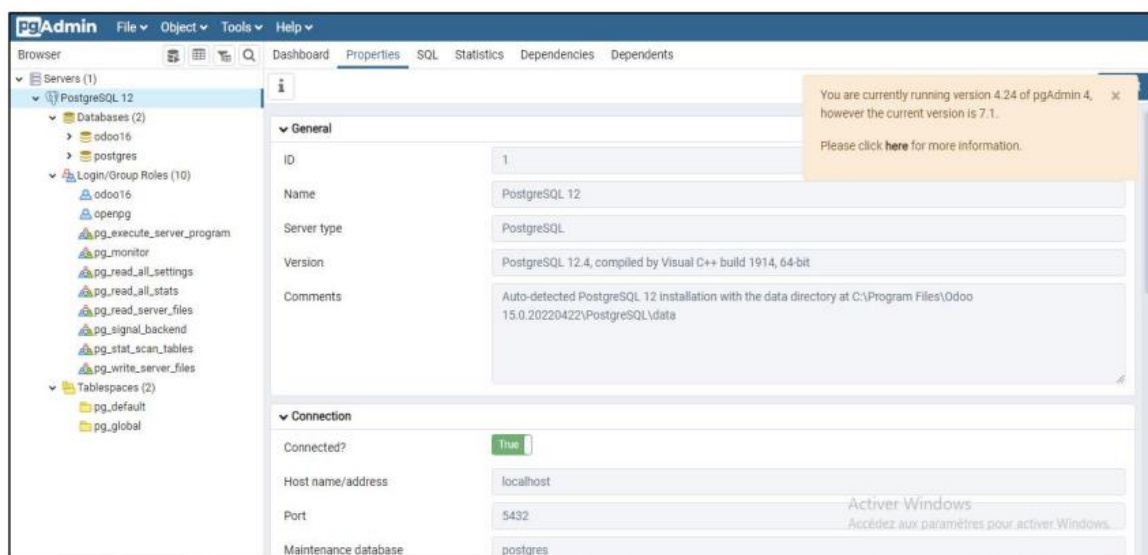
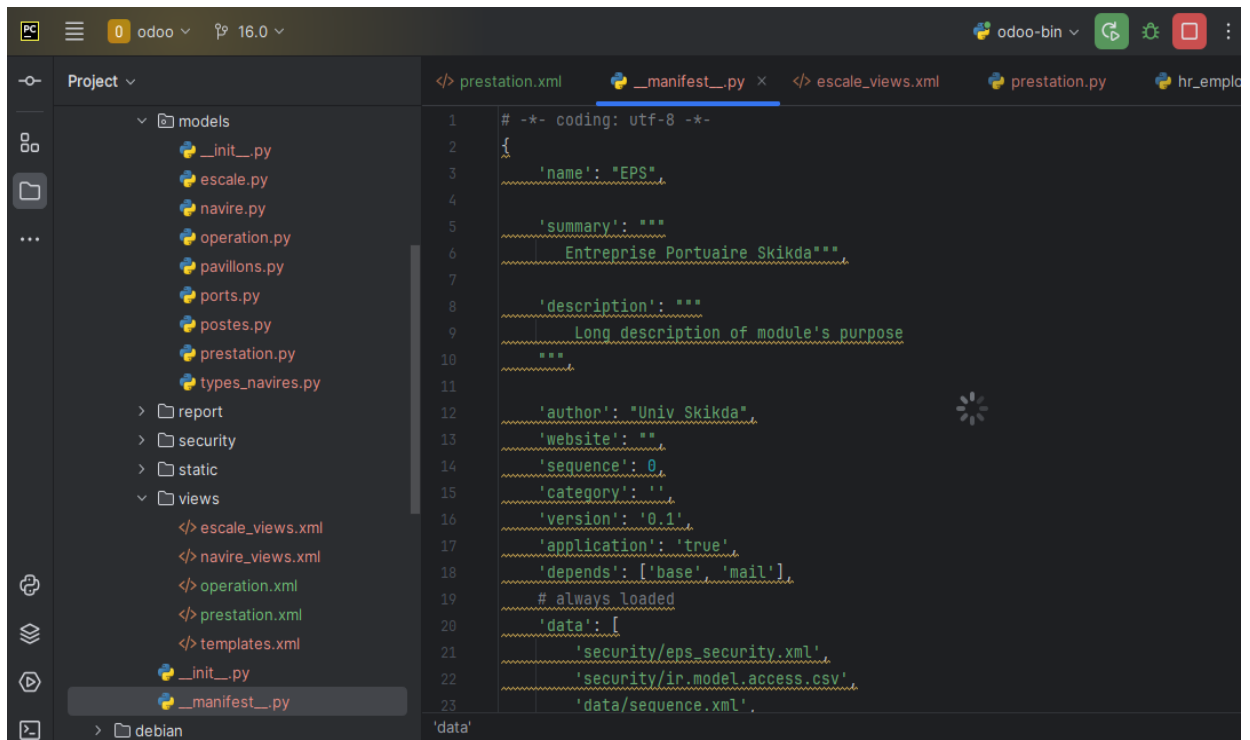


Figure 22 : L'existence de la base de données Odoo16.

Ensuite la première étape est l'implémentation de fichier **manifest .py**.

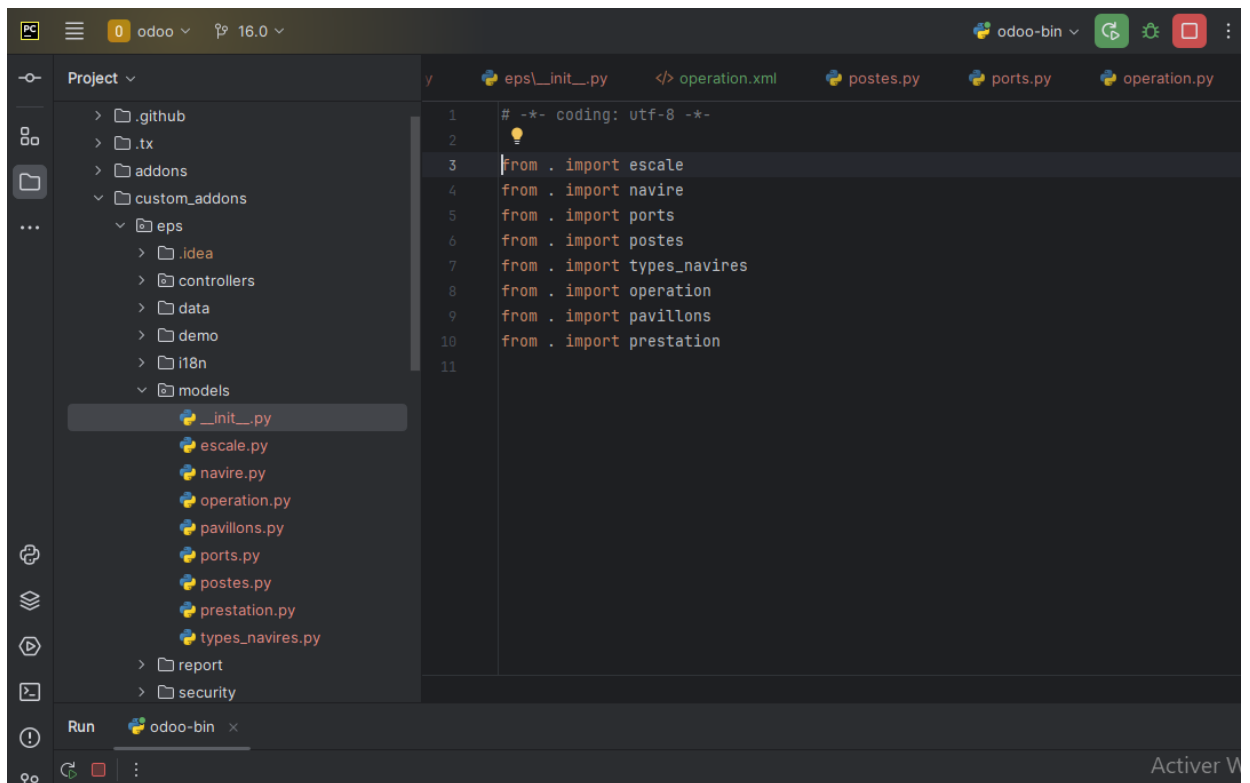


The screenshot shows a code editor with a project tree on the left and a manifest file being edited. The project tree includes folders for models, report, security, static, and views, with various Python and XML files listed. The manifest file contains the following code:

```
1 # -*- coding: utf-8 -*-
2 {
3     'name': "EPS",
4     'summary': """
5         Entreprise Portuaire Skikda"""
6     'description': """
7         Long description of module's purpose
8         """
9     'author': "Univ Skikda",
10    'website': ""
11    'sequence': 0,
12    'category': 'base',
13    'version': '0.1',
14    'application': 'true',
15    'depends': ['base', 'mail'],
16    # always loaded
17    'data': [
18        'security/eps_security.xml',
19        'security/ir_model.access.csv',
20        'data/sequence.xml',
21    ]
22 }
```

Figure 23 : L'implémentation de fichier manifest .py de notre module EPS.

Et le fichier `init .py` ainsi que l'arborescence des fichiers nécessaires.



The screenshot shows a code editor with a project tree on the left and an init file being edited. The project tree includes folders for .github, .tx, addons, custom_addons, eps, .idea, controllers, data, demo, i18n, and models, with various Python files listed. The init file contains the following code:

```
1 # -*- coding: utf-8 -*-
2
3 from . import escale
4 from . import navire
5 from . import ports
6 from . import postes
7 from . import types_navires
8 from . import operation
9 from . import pavillons
10 from . import prestation
11
```

Figure 24 : L'implémentation de fichier init .py de notre module EPS.

Et après remplir le formulaire de paramètre d'entreprise :

The screenshot shows the 'Informations générales' (General Information) tab of the Odoo company configuration form. The company name is 'Entreprise Portuaire Skikda'. The contact information includes 'Entreprise Portuaire Skikda' and phone number '038752254'. The address is 'Petite zone industrielle près embuchure oued SAFSAF', 'Rue 2...', 'SKIKDA', 'California (US)', '21000', 'Algérie'. The email is 'epsskikda@skikda-port.com' and the website is 'http://www.skikda-port.com'. The tax ID and company ID fields are empty. The currency is 'DZD'. The company logo is displayed in the top right and bottom right corners.

Figure 25 : Formulaire des détails d'entreprise.

La Figure suivante présenter le module EPS avec l'inventaire des modules Odoo à modifier.

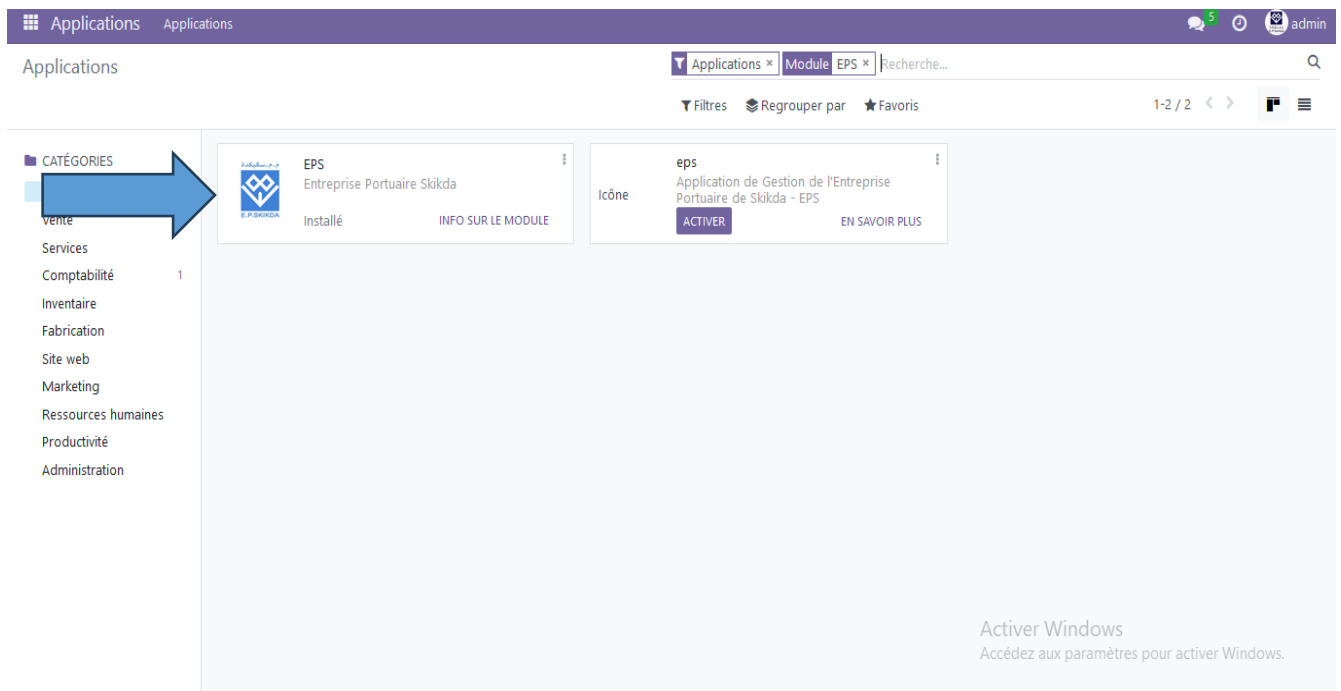


Figure 26 : Le module EPS sur Odoo.

La figure suivante représente le Login pour accéder à notre module sous Odoo16.

Email

Password

Log in

[Reset Password](#)

[Log in as superuser](#)

Figure 27 : Le module EPS sur Odoo.

- **Création authentification admin :**

Paramètres / Utilisateurs / Mitchell Admin

ENVOYER LES INSTRUCTIONS DE RÉINITIALISATION DU MOT DE PASSE

JAMAIS CONNECTÉ CONFIRMÉ

15 Groupes 338 Droits D'accès 31 Règles Sur Les ...

Nom ?
fady

Adresse email ?
fady@gmail.com

Partenaire associé ? YourCompany, Mitchell Admin

Droits d'accès Préférences Sécurité du compte

USER TYPE

Types d'utilisateur ?
 Utilisateur interne
 Portail
 Public

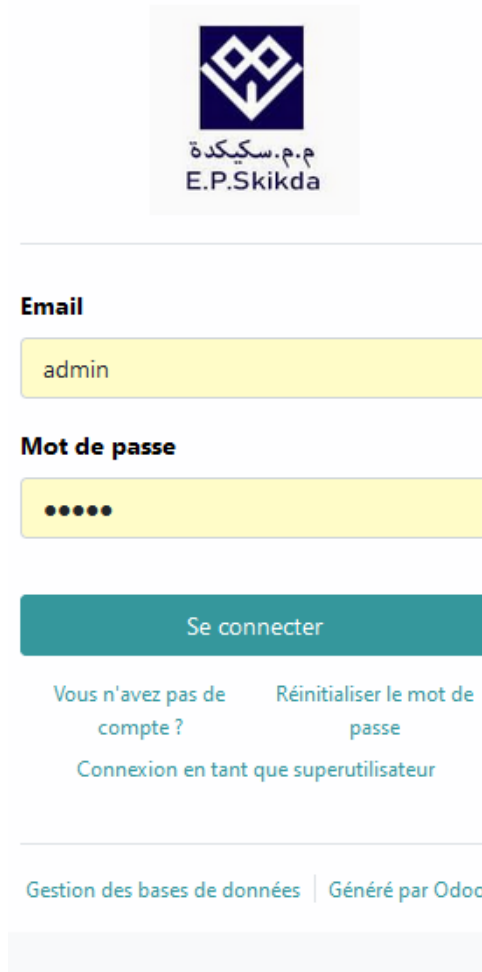
SALES
Vente ? Administrateur

ACCOUNTING
Comptabilité ? Paramétrage facturation

ADMINISTRATION
Administration ? Paramètres

Activer Windows
Accédez aux paramètres pour activer Windows.

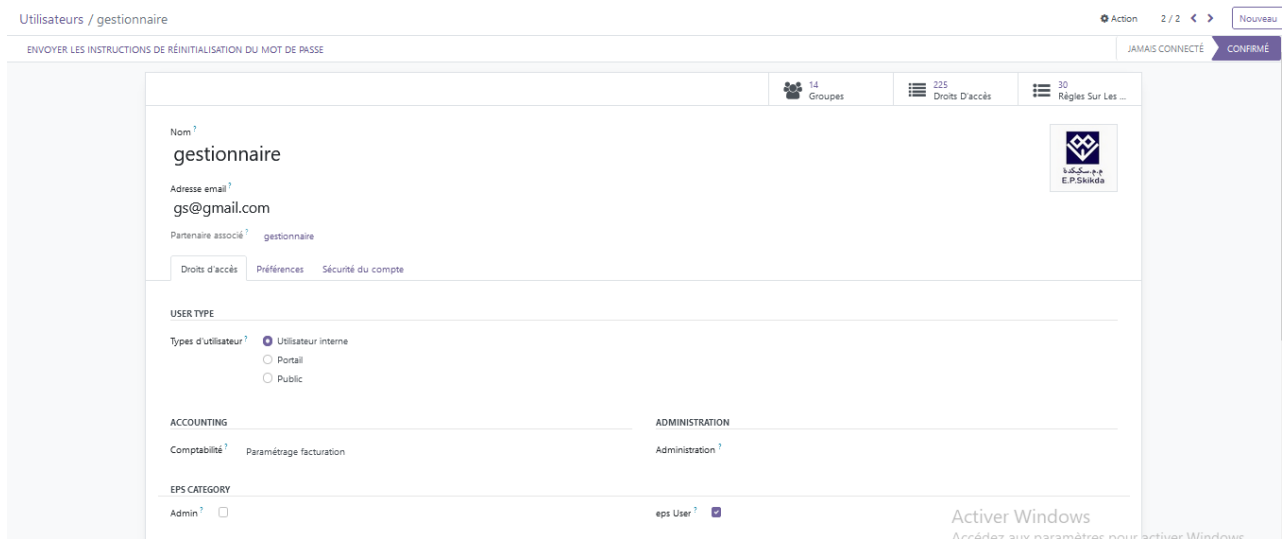
Figure 28 : création des admin.



The image shows the admin login page for E.P. Skikda. At the top is the logo of E.P. Skikda, which consists of a stylized geometric design in blue and white, with the text 'م.م.سككدة' and 'E.P.Skikda' below it. Below the logo, there are two input fields: one for 'Email' containing the text 'admin' and one for 'Mot de passe' (password) represented by six dots. A teal button labeled 'Se connecter' is positioned below the password field. Underneath the button, there are two links: 'Vous n'avez pas de compte ?' and 'Réinitialiser le mot de passe', with the text 'Connexion en tant que superutilisateur' centered below them. At the bottom of the page, there are two links: 'Gestion des bases de données' and 'Généré par Odoo'.

Figure 29 : page login admin.

- **Création authentification gestionnaire :**



The image shows the user creation page in Odoo. The page title is 'Utilisateurs / gestionnaire'. At the top right, there are navigation links for 'Action', '2 / 2', and 'Nouveau'. Below the title, there is a link to 'ENVOYER LES INSTRUCTIONS DE RÉINITIALISATION DU MOT DE PASSE'. The main content area is divided into several sections. The first section contains the user's name 'gestionnaire', email 'gs@gmail.com', and associated partner 'gestionnaire'. Below this are tabs for 'Droits d'accès', 'Préférences', and 'Sécurité du compte'. The 'USER TYPE' section has three radio buttons: 'Utilisateur interne' (selected), 'Portail', and 'Public'. The 'ACCOUNTING' section has a field for 'Comptabilité' with the value 'Paramétrage facturation'. The 'ADMINISTRATION' section has a field for 'Administration'. The 'EPS CATEGORY' section has a field for 'Admin' with a checkbox that is unchecked. At the bottom right, there is a watermark for 'Active Windows' and a link to 'Accédez aux paramètres pour activer Windows'.

Figure 30 : création d'un utilisateur gestionnaire.

The image shows a user login interface. At the top center is a logo consisting of a stylized blue and white geometric design above the text 'E.P.Skikda' and 'م.م.سكيدة'. Below the logo is a horizontal line. Underneath, the label 'Email' is followed by a yellow input field containing the text 'gs@gmail.com'. Below that, the label 'Mot de passe' is followed by a yellow input field containing two black dots. A teal button with the text 'Se connecter' is centered below the password field. Below the button, there are three links: 'Vous n'avez pas de compte?' on the left, 'Réinitialiser le mot de passe' in the middle, and 'Connexion en tant que superutilisateur' on the right. At the bottom of the page, there is a footer with the text 'Gestion des bases de données | Généré par Odoo'.

Figure 31 : page login utilisateur.

- **Les droits d'accès des utilisateurs :**

Pour faire les droits d'accès de chaque utilisateur, il faut tout d'abord ajouter les autorisations dans 'ir.model.access.csv' comme été présenter dans la figure suivante :

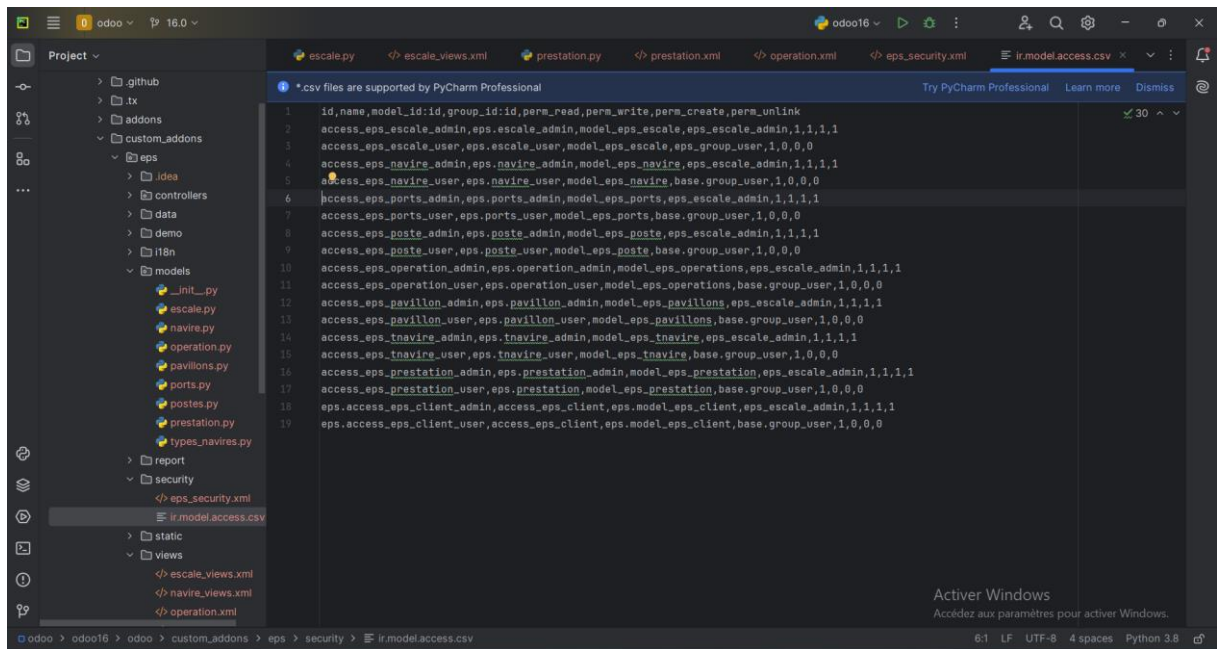


Figure 32 : L'implémentation de fichier `ir.model.access.csv` dans notre app.

Ensuite, on va aller sur l'application et sélectionner les droits de chaque utilisateur, et la figure suivante présente cette action :

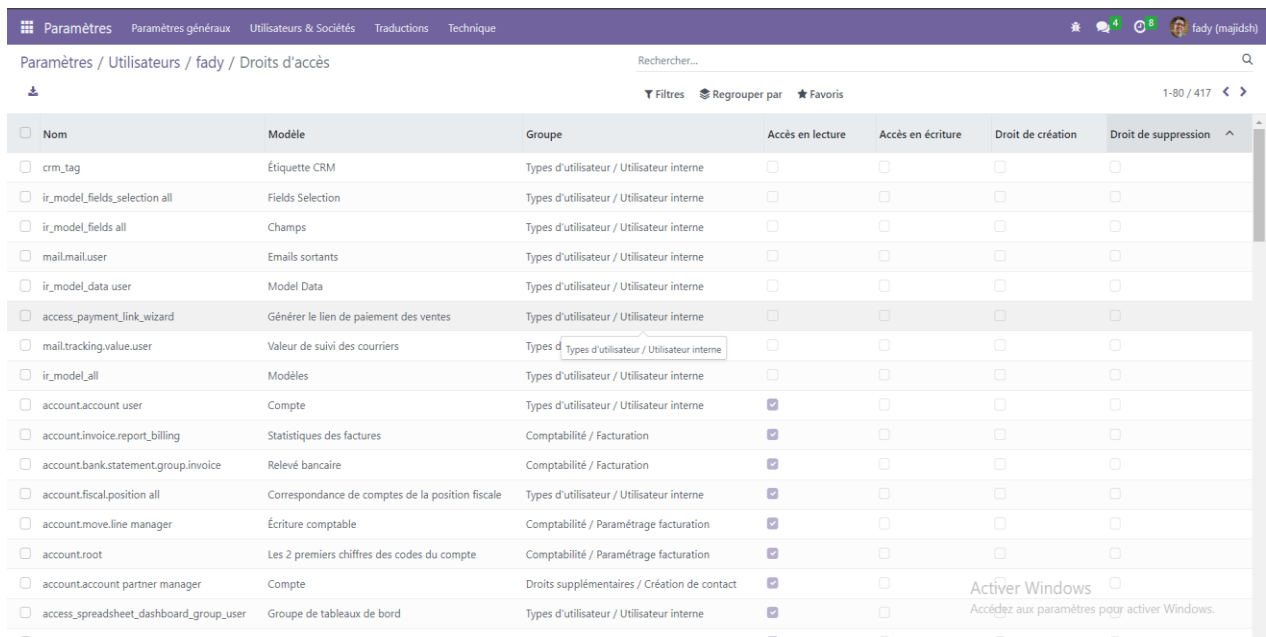
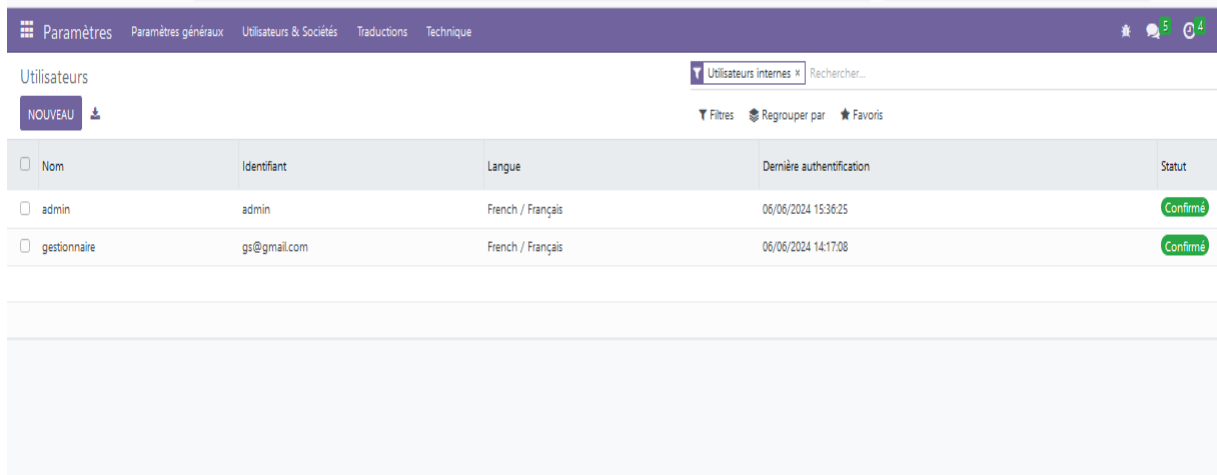


Figure 33 : Liste des droits d'accès de notre utilisateur.

La figure suivante représente liste des utilisateurs et des admins avec leur tâche :



| <input type="checkbox"/> | Nom | Identifiant | Langue | Dernière authentification | Statut |
|--------------------------|--------------|--------------|-------------------|---------------------------|----------|
| <input type="checkbox"/> | admin | admin | French / Français | 06/06/2024 15:36:25 | Confirmé |
| <input type="checkbox"/> | gestionnaire | gs@gmail.com | French / Français | 06/06/2024 14:17:08 | Confirmé |

Figure 34 : liste d'user et admin avec leurs droits.

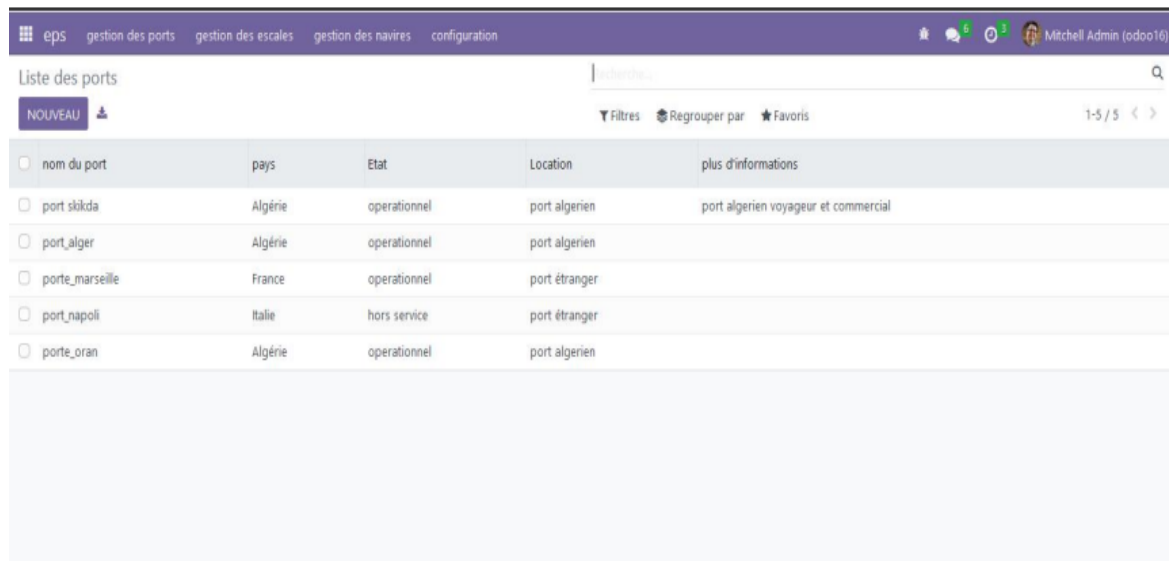
Dans la figure suivante on a représenté comment ajouter l'admin et user dans l'application à partir le code source :

```
3 <odoo>
4
5 <record model="ir.module.category" id="eps_category">
6   <field name="name">eps category</field>
7 </record>
8
9 <record id="eps_escale_admin" model="res.groups">
10  <field name="name">Admin</field>
11  <field name="category_id" ref="eps_category"/>
12 </record>
13
14 <record id="eps_group_user" model="res.groups">
15  <field name="name">eps User</field>
16  <field name="category_id" ref="eps_category"/>
17 </record>
18
19 </odoo>
20
```

Figure 35 : l'implémentation de fichier 'eps_security.xml'.

4.2. Développement de module :

- **Menu principal de l'application :**

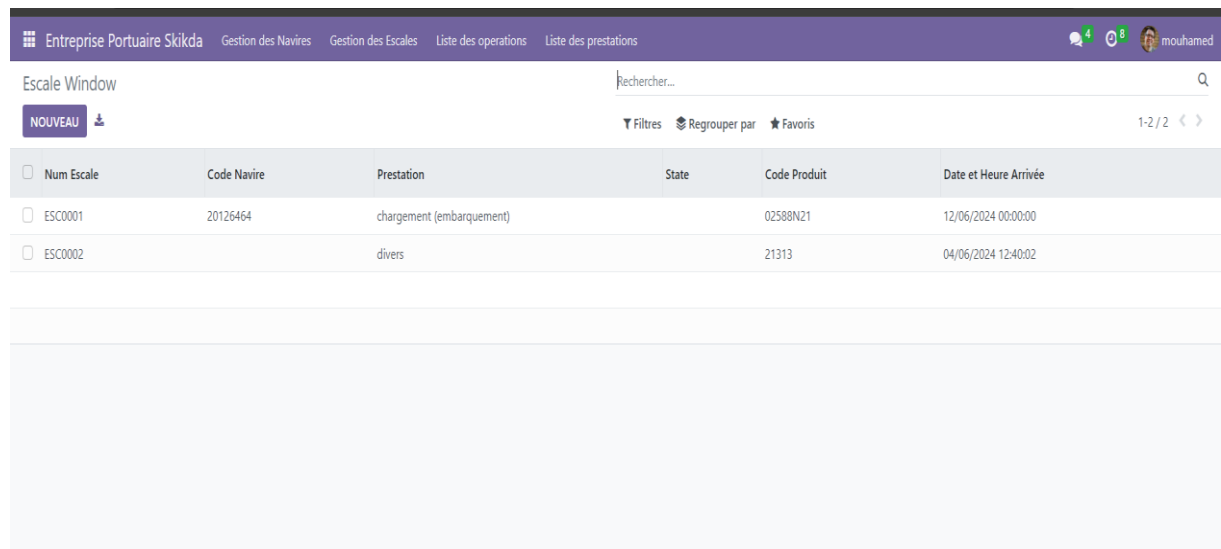


The screenshot shows the main menu of the application. The header includes navigation links: 'eps', 'gestion des ports', 'gestion des escales', 'gestion des navires', and 'configuration'. The user is logged in as 'Mitchell Admin (odoo16)'. The main content area is titled 'Liste des ports' and features a search bar, a 'NOUVEAU' button, and options for 'Filtres', 'Regrouper par', and 'Favoris'. The table below lists several ports with columns for 'nom du port', 'pays', 'Etat', 'Location', and 'plus d'informations'.

| <input type="checkbox"/> | nom du port | pays | Etat | Location | plus d'informations |
|--------------------------|-----------------|---------|--------------|---------------|--------------------------------------|
| <input type="checkbox"/> | port skikda | Algérie | operationnel | port algerien | port algerien voyageur et commercial |
| <input type="checkbox"/> | port_alger | Algérie | operationnel | port algerien | |
| <input type="checkbox"/> | porte_marseille | France | operationnel | port étranger | |
| <input type="checkbox"/> | port_napoli | Italie | hors service | port étranger | |
| <input type="checkbox"/> | porte_oran | Algérie | operationnel | port algerien | |

Figure 36 : Menu principale de l'application.

- **Liste view des escales :**



The screenshot shows the 'Liste des escales' view. The header includes navigation links: 'Entreprise Portuaire Skikda', 'Gestion des Navires', 'Gestion des Escales', 'Liste des operations', and 'Liste des prestations'. The user is logged in as 'mouhamed'. The main content area is titled 'Escale Window' and features a search bar, a 'NOUVEAU' button, and options for 'Filtres', 'Regrouper par', and 'Favoris'. The table below lists scale operations with columns for 'Num Escale', 'Code Navire', 'Prestation', 'State', 'Code Produit', and 'Date et Heure Arrivée'.

| <input type="checkbox"/> | Num Escale | Code Navire | Prestation | State | Code Produit | Date et Heure Arrivée |
|--------------------------|------------|-------------|---------------------------|-------|--------------|-----------------------|
| <input type="checkbox"/> | ESC0001 | 20126464 | chargement (embarquement) | | 02588N21 | 12/06/2024 00:00:00 |
| <input type="checkbox"/> | ESC0002 | | divers | | 21313 | 04/06/2024 12:40:02 |

Figure 37 : Liste des escales.

- **Formulaire listes des escales :**

Entreprise Portuaire Skikda | Gestion des Navires | Gestion des Escales | Liste des opérations | Liste des prestations

Escale Window / Nouveau | Imprimer | Action | Nouveau | Envoyer un message | Enregistrer une note | Activités | Suivre

Caractéristiques d'Escale

VERS ETAT RADE | VERS ETAT QUAI | VERS ETAT ATTENDU | RADE | QUAI | ATTENDU

0 EN RADE | 0 EN QUAI | 0 ATTENDU

New

| | | | |
|------------------------|-------------------------------------|--------------------------------|-------------------------------------|
| Code Navire? | | Notes? | |
| Situation? | 0 | Date et Heure Depart? | |
| Prestation? | | Produit Danger? | <input checked="" type="checkbox"/> |
| Code Produit? | | Type Marchandise? | 1 |
| Date et Heure Arrivée? | | Code Armateur? | |
| Type? | 0 | Date et Heure Rade? | |
| Position Est? | | Date et Heure Entrée? | |
| Navire Affrété? | <input checked="" type="checkbox"/> | Date et Heure Sortie? | |
| Port Province? | | Tonnage Embarqué? | 0,00 |
| Port Destination? | | Classe Statistique? | 0 |
| Port Dernier? | | Tonnage Embarqué Mois Suivant? | 0,00 |
| Tonnage? | 0,00 | Tonnage Debarqué Mois Suivant? | 0,00 |
| Atp? | | | |

Aujourd'hui

fady
Création d'un nouvel enregistrement...

Activer Windows
Accédez aux paramètres pour activer Windows.

Figure 38 : Formulaire des escales.

- **Liste view des opérations escales : La liste des opérations déjà créées .**

| <input type="checkbox"/> | Code Opération | Designation | date debut | date fin | Heure Debut | Heure Fin | Code Poste | Designation Poste |
|--------------------------|----------------|----------------------------|------------|------------|---------------------|---------------------|------------|----------------------------|
| <input type="checkbox"/> | 035N0 | Dérangement Pilote | 19/06/2024 | 20/06/2024 | 19/06/2024 13:44:33 | 20/06/2024 13:45:36 | 0109 | Dérangement Pilote |
| <input type="checkbox"/> | N524J | Mouillage a Bord du Pilote | 11/06/2024 | 12/06/2024 | 11/06/2024 13:46:19 | 12/06/2024 13:47:39 | 0107 | Mouillage a Bord du Pilote |
| <input type="checkbox"/> | 120LIB | Libération | 13/06/2024 | 14/06/2024 | 13/06/2024 13:48:19 | 14/06/2024 13:49:02 | 0104 | Libération |
| <input type="checkbox"/> | 25N200F | Garde à Bord du Pilote | 17/06/2024 | 18/06/2024 | 17/06/2024 13:49:44 | 18/06/2024 13:49:44 | 0106 | Garde à Bord du Pilote |

Figure 39 : liste view opération.

- **formulaire opération : ci-dessous le formulaire et les champs nécessaires pour créer une opération**

Code Opération ?

Designation ?

date debut ?

date fin ?

Heure Debut ?

Heure Fin ?

Code Poste ?

Designation Poste ?

quantité ?

Montant ?

Figure 40 : formulaire opération.

4.2.1. Archiver un enregistrement ou supprimer (une opération) :

Après remplir le formulaire d'une opération dans notre application et pour archiver un enregistrement on ajoute un Field actif de type Boolean dans la classe 'escale.py' : `active=fields.Boolean(default=True)`.

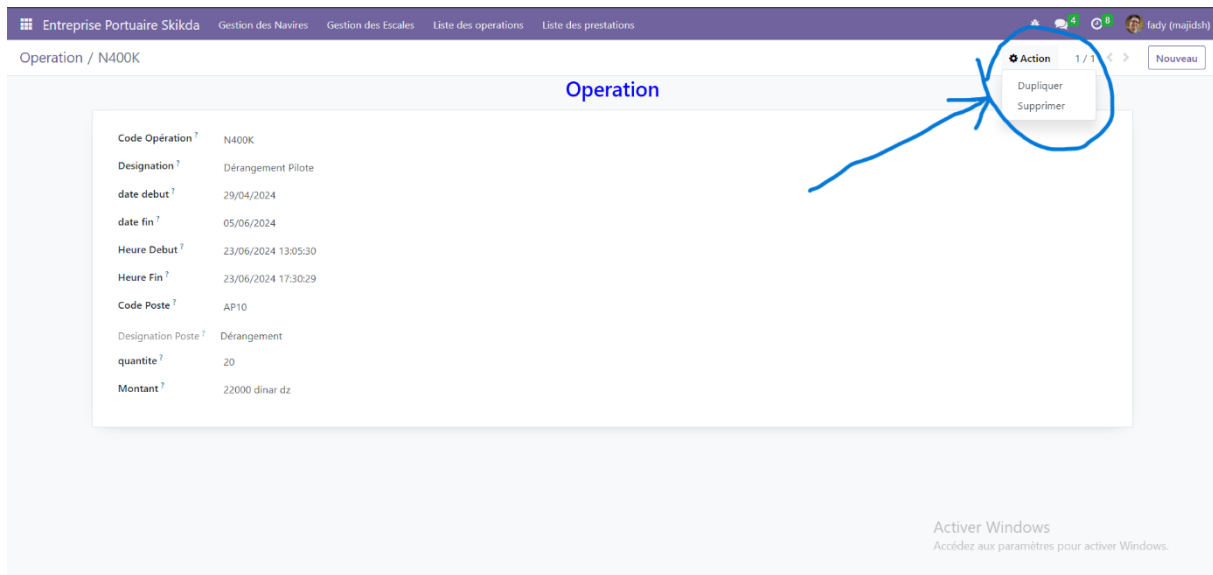


Figure 41 : bouton pour supprimer ou archiver une opération dans le formulaire.

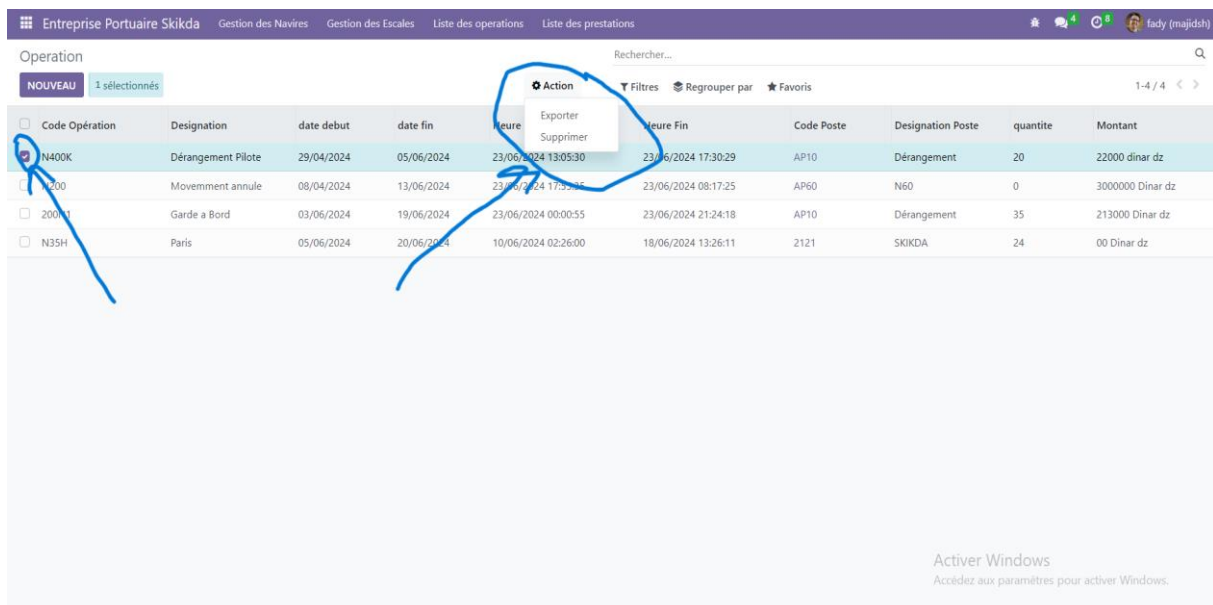


Figure 42 : bouton pour supprimer ou archiver une opération.

4.2.2. Calendaire et heure view :

La vue du calendrier dans Odoo est un outil de gestion des événements et des tâches qui permet d'afficher les informations dans un format de calendrier. Cette vue permet aux utilisateurs d'organiser, de planifier et de suivre les activités dans Odoo de manière visuelle et intuitive.

Pour cela, on a utilisé cet outil dans notre projet pour ajouter une date de début et une date de fin d'opération et la même chose pour heur comme été définie dans les figures suivantes :

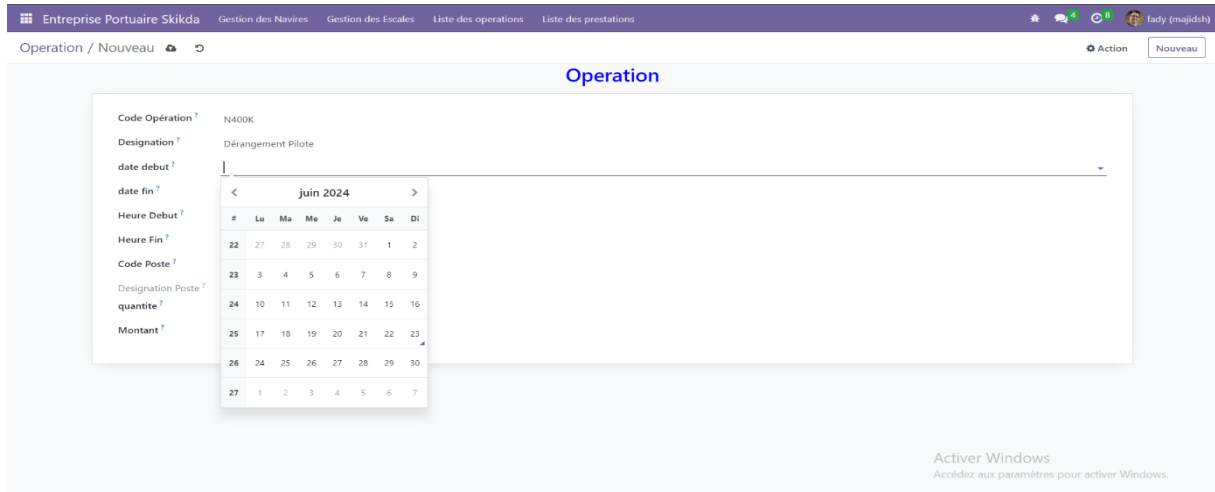


Figure 43 : Ajouter date dans Odoo.

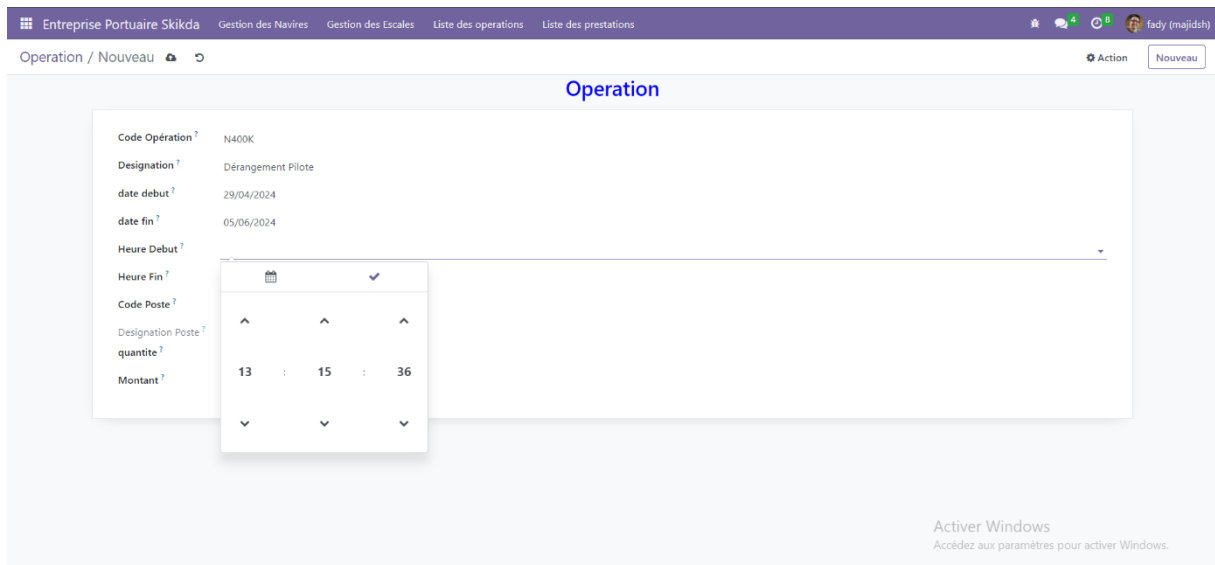


Figure 44 : Ajouter heur dans Odoo.

4.2.3. Code poste et designation :

Après remplir le formulaire d'opération on doit il faut entrer le code poste et ça designation comme été présenter dans la figure suivante :

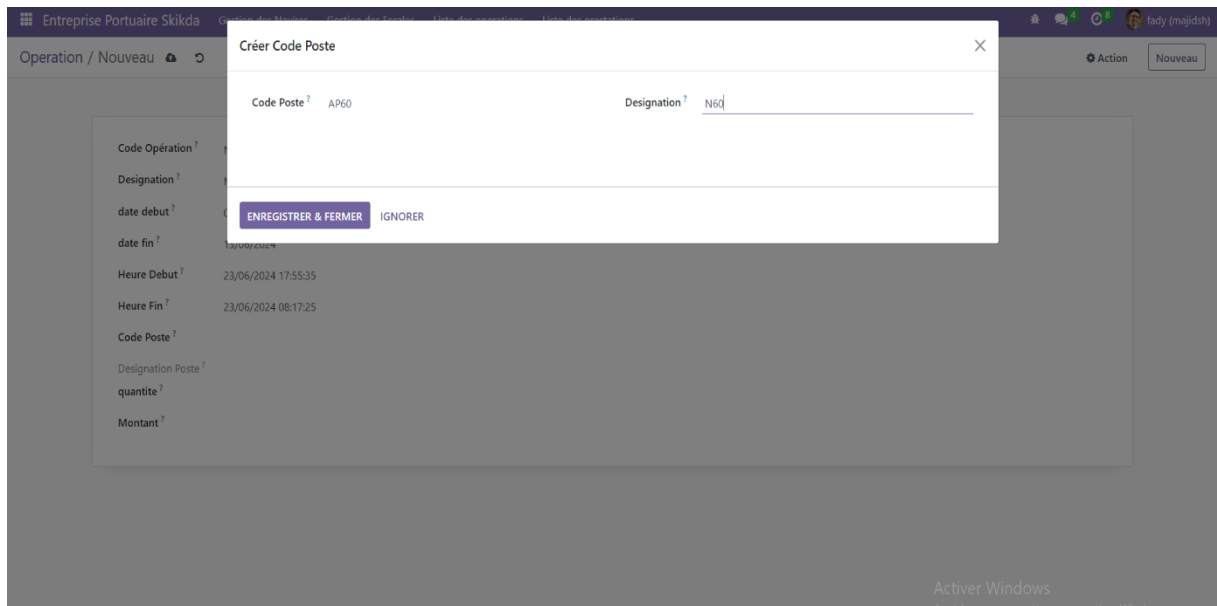


Figure 45 : Ajouter code poste et la designation.

Si le 'code poste' existe auparavant on peut le choisir comme cela :

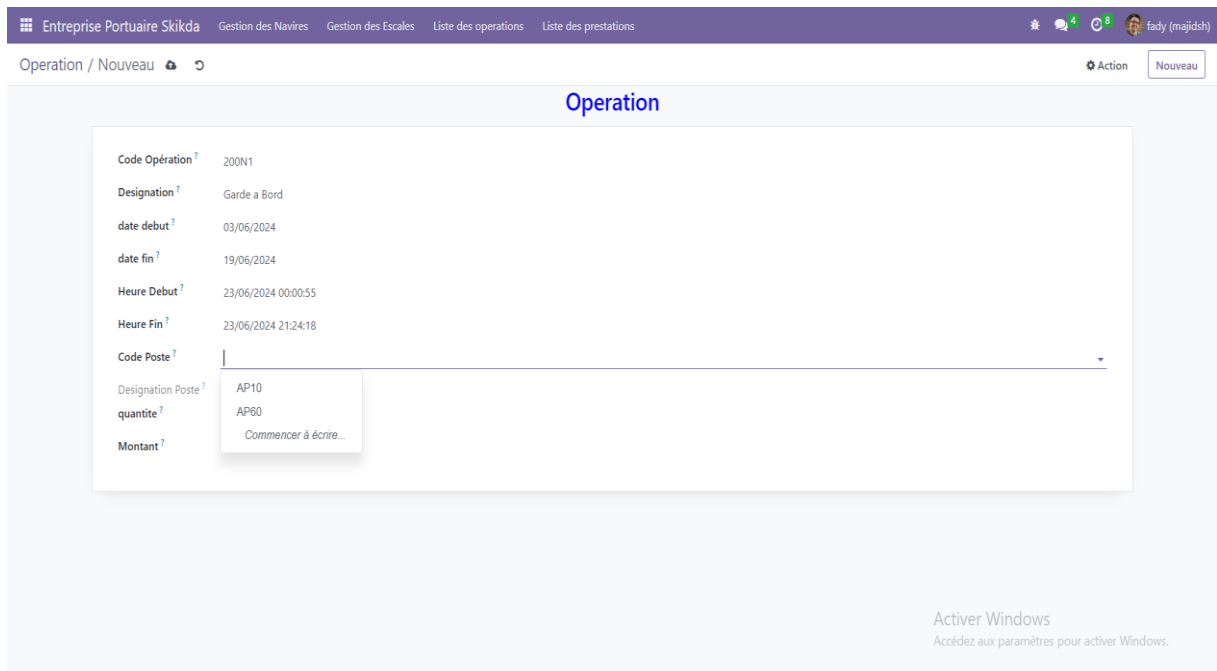


Figure 46 : Sélectionner code poste et la designation.

4.2.4. Enregistrement d'une opération :

Pour enregistrer une opération il y a un bouton :

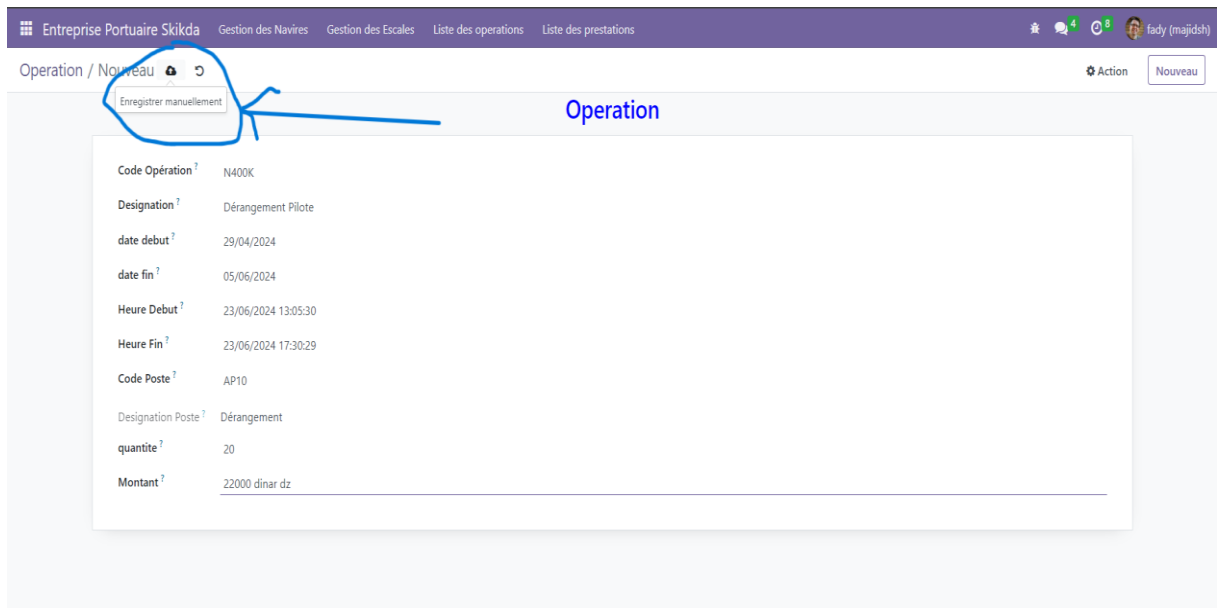


Figure 47 : Engistrement d'une opération.

4.2.5. Ajouter une opération comme favoris :

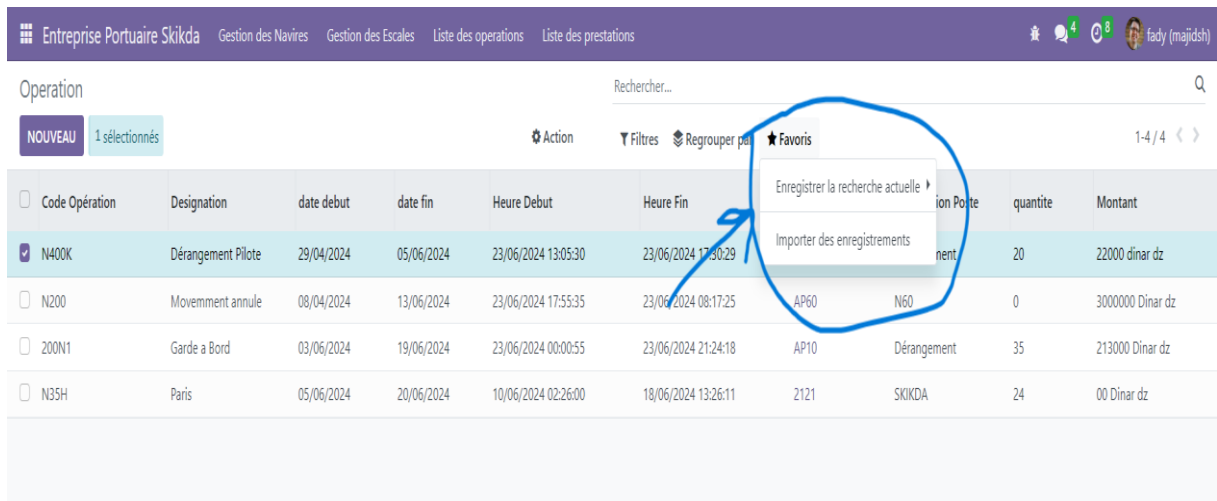
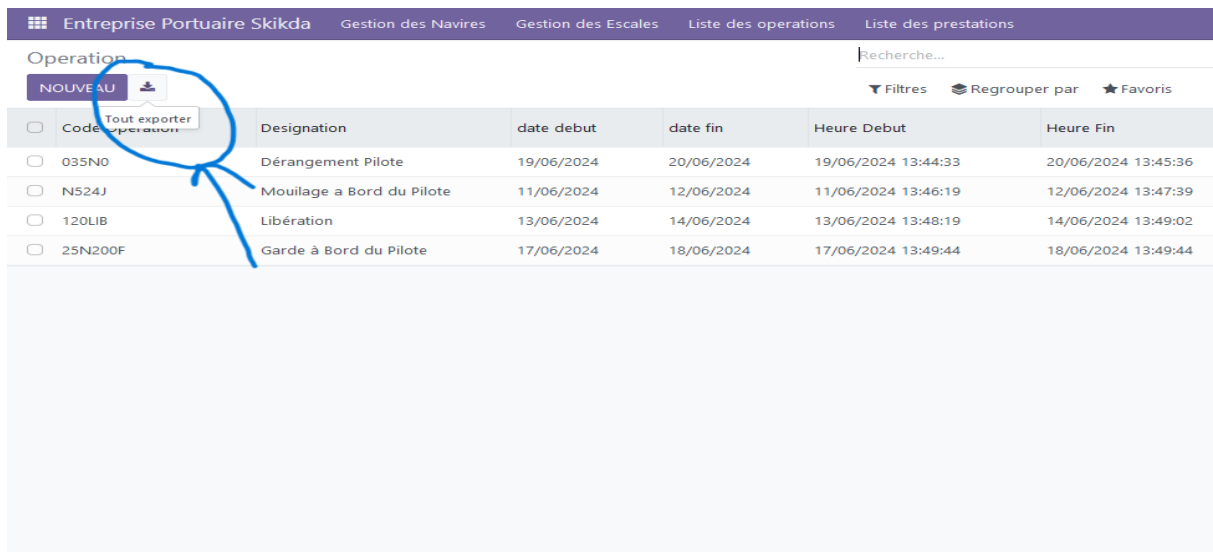


Figure 48 : Ajouter une opération favorite.

4.2.6. Télécharger les opérations :



| Code | Designation | date debut | date fin | Heure Debut | Heure Fin |
|---------|----------------------------|------------|------------|---------------------|---------------------|
| 035NO | Dérangement Pilote | 19/06/2024 | 20/06/2024 | 19/06/2024 13:44:33 | 20/06/2024 13:45:36 |
| N524J | Mouillage a Bord du Pilote | 11/06/2024 | 12/06/2024 | 11/06/2024 13:46:19 | 12/06/2024 13:47:39 |
| 120LIB | Libération | 13/06/2024 | 14/06/2024 | 13/06/2024 13:48:19 | 14/06/2024 13:49:02 |
| 25N200F | Garde à Bord du Pilote | 17/06/2024 | 18/06/2024 | 17/06/2024 13:49:44 | 18/06/2024 13:49:44 |

Figure 49 : Bouton télécharger les opérations.

4.3. Relations entre les attributs :

Les relations One2Many, Many2One et Many2Many sont des types de relations de base de données utilisées dans Odoo pour établir des liens entre différents modèles.

- **Many2One (plusieurs à un) :**

- C'est une relation où plusieurs enregistrements d'un modèle sont liés à un seul enregistrement d'un autre modèle.
- Dans le modèle source (la table principale), vous utilisez un champ de type Many2one pour établir cette relation.
- Dans le modèle cible (la table liée), vous utilisez un champ de type One2many pour indiquer la relation inverse.

Exemple : il peut exister plusieurs code poste dans une seule opération (la même opération mais différant code poste).

```
<field name="code"/>
<field name="designation"/>
<field name="date_debut" widget="date"/>
<field name="date_fin" widget="date"/>
<field name="h_debut"/>
<field name="h_fin"/>
<field name="code_poste" widget="many2one"/>
<field name="designation_poste"/>
```

Figure 50 : Un exemple de code many2one.

- **Many2Many (plusieurs à plusieurs) :**
 - C'est une relation où plusieurs enregistrements d'un modèle sont liés à plusieurs enregistrements d'un autre modèle.
 - Pour établir une relation Many2Many, vous devez utiliser une table de liaison qui relie les deux modèles.
 - Dans le modèle source, vous utilisez un champ de type Many2many pour établir cette relation et vous spécifiez la table de liaison.
 - Dans le modèle cible, vous utilisez également un champ de type Many2many pour indiquer la relation inverse.
- **One2Many (un à plusieurs) :**
 - C'est une relation où un enregistrement d'un modèle est lié à plusieurs enregistrements d'un autre modèle.
 - Dans le modèle source (la table principale), vous utilisez un champ de type One2many pour établir cette relation.
 - Dans le modèle cible (la table liée), vous utilisez un champ de type Many2one pour indiquer la relation inverse.

Exemple : une escale peut faire un ou plusieurs opérations, une opération est assurée par une seule escale.

5. Conclusion

Dans ce chapitre nous avons décrit l'environnement de développement, ainsi que les différents outils et techniques informatiques qui ont été utilisés. Nous avons présenté les étapes détaillées d'implémentation, de configuration et d'installation de

notre module « la conception et la gestion des opération au sein d'un escale » sous Odoo 16. Nous avons illustré également les différentes captures d'écran de notre module et qui présente une partie des fonctionnalités du système conçu.

Conclusion générale

Tout au long de la préparation de notre mémoire, nous avons essayé de mettre en pratique les connaissances que nous avons acquies durant nos études universitaires pour la conception et réalisation d'un module ERP pour la gestion des opérations des escales de l'Entreprise Portuaire de Skikda.

En effet, nous avons fait une étude approfondie du système SIGNE utilisé par le service capitainerie de l'EPS pour la gestion des escales. Nous avons par la suite critiqué ce système et fait ressortir ses failles, faiblesses et limites dans le but de proposer une solution adéquate en accord avec l'organisme d'accueil. Notre logiciel est développé en utilisant un ensemble d'outil et de techniques innovantes .

Contrairement aux solutions existantes, l'ERP proposée permet non seulement de créer des navires et nouveaux escales , mais aussi d'établir les différents opérations liées aux escales, d'alerter les membres du personnel de la capitainerie en cas de modification de toute information en relation avec ces opérations , et aussi permettre d'avoir une discussion entre les admins et les gestionnaires du plateforme .

Comme tout projet, notre travail peut être compléter et amélioré par l'intégration de nouvelles fonctionnalités. Ainsi, nous envisageons comme perspectives, d'une part de relier le logiciel à la base nationale APCS.dz, et d'autre part, de programmer des alertes sur les smartphones des utilisateurs en cas de création d'une nouvelle notification.

En plus du service capitainerie, nous espérons élargir le champ d'utilisation de notre proposition à d'autres services de l'EPS par l'intégration de nouveaux modules permettant de prendre en charge la facturation, la gestion des manifestes, la gestion des remorqueurs etc.

Ce projet a fait l'objet d'une expérience très intéressante, car elle nous a permis de nous familiariser avec de nouvelles notions d'une part, et d'améliorer nos connaissances et nos compétences dans le domaine de la programmation, et d'autre

part avoir le sens de responsabilité et gestion de projets qui permet et faciliter l'insertion dans le domaine professionnel .

Références

- [1] : source Entreprise Portuaire SKIKDA (EPS) (Accès le 22/03/2023).
- [2] : <https://www.choisirmonerp.com/erp/definition-d-un-erp>.
- [3] : <https://www.sqorus.com/logiciel-erp-definition/>.
- [4] : <https://www.choisirmonerp.com/erp/>.
- [5] : <https://www.isdigi.ma/blog/articles-2/architecture-du-erp-odoo-2>.
- [6] : <https://erp.ooreka.fr/comprendre/logiciel-erp>.
- [7] : https://fr.wikipedia.org/wiki/Progiciel_de_gestion_intégré.
- [8] : <http://www.quadrans.fr/index.php/fr/savoir-faire/prestation-a-la-carte/erp>.
- [9] : Parth GAJJAR, et al, Odoo 12 développement Cookbook, 4ème édition Packt 2019.
- [10] : <https://www.flowlineintegration.com>
- [11] : <https://opensource.com/tools/enterprise-resource-planning>.
- [12] : <http://grh2017.byethost14.com/wp/2017/04/24/la-plate-forme-odoo/>.
- [13] : <https://fr.wikipedia.org/wiki/Odoo>.
- [14] : <https://www.xperdoo.fr/blog/expert-comptable-odoo-1/>.
- [15] : <https://www.isdigi.ma/blog/articles-2/architecture-du-erp-odoo-2>.
- [16] : <https://odooskills.com/>.
- [20] : https://www.captivea.com/fr_FR/.

- [21] : <https://www.altova.com/fr/umodel>.
- [22] : <https://www.gleek.io/blog/uml-diagram-types>.
- [23] : <https://web.maths.unsw.edu.au/~lafaye/CCM/uml/uml-use-cases.htm>.
- [24] : <https://www.lucidchart.com>
- [25] : https://fr.wikipedia.org/wiki/Windows_11.
- [26] : <https://www.postgresql.org/>.
- [27] : <https://www.python.org/>.
- [28] : <https://fr.wikipedia.org/wiki/PyCharm>.
- [29] : <https://fr.wikipedia.org/wiki/GitHub>.
- [30] : <https://www.w3.org/XML/>.
- [31] : <https://odooskills.com/bien-comprendre-architectue-technique-odoo>.
- [32] : https://www.odoo.com/documentation/16.0/administration/on_premise/source.html.