

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNANEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE



UNIVERSITÉ ABDERAHMANE MIRA DE BÉJAIA  
FACULTÉ DES SCIENCES EXACTES  
DÉPARTEMENT D'INFORMATIQUE

## MÉMOIRE DE FIN DE CYCLE

En vue de l'obtention du diplôme de Master Professionnel en  
Informatique

Option : Administration et Sécurité des Réseaux

### *Thème*

**Adaptation et intégration d'OpenERP pour la gestion  
des stages des enseignants universitaire  
Cas d'étude : Faculté de Biologie**

Présenté par :

Melle LAIFA Yasmina

En cadrée par :

Mme TAHAKOURT Zineb

Devant le jury composé de :

Président Mr MIR Foudil

Maitre assistant (Université UAMB)

Examineur Mr OUZEGGANE Redouane

Maitre assistant (Université UAMB)

Examineur Mr NAFI Mohamed

Maitre assistant (Université UAMB)

*Promotion 2013-2014*

# Remerciements

*Je tiens tout d'abord à remercier le **bon Dieu** qui ma donné la santé et le courage d'accomplir ce travail.*

*Je ne peux pas oublier de présenter ma gratitude à **mes chers parents** pour leur patience et les efforts inlassables qu'ils ne cessent de déployer pour nous.*

*Mes vifs remerciements vont à **Mme TAHAKOURT Zineb**, ma promotrice pour tous ses conseils très précieux, ses encouragements, sa patience et ses orientations qui mon ont été très bénéfiques tout au long de ce travail.*

*Je remercie également les membres de jury qui nous font honneur en acceptant d'examiner et de juger notre travail.*

*Mes remerciements aussi à **Mr OMAR Moualoud**, chef de département informatique, ainsi qu'à l'ensemble des enseignants pour les efforts fournis pour notre bonne formation.*

*Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de mes remerciements les plus chaleureux.*

## Dédicace

*Je dédie ce Modest travail*

*A ma mère ma raison d'être, ma raison de vivre, la lanterne qui éclaire mon chemin et m'illumine de douceur et d'amour.*

*A mon père, en signe d'amour, de reconnaissance et de gratitude pour tous les soutiens et les sacrifices dont il a fait preuve à mon égard.*

*A mes chères frères et sœurs*

*A mon très chère Atmane*

*Aucun mot ne pourra décrire vos dévouements et vos sacrifices.*

*A tous mes amis (Addi, Biba, Celia, firo Lamo, Lati, Nena, Radouche, wawal, Yasmine), En témoignage de l'amitié sincère qui nous a liées et des bons moments passés ensemble. Je vous dédie ce travail en vous*

*Souhaitant un avenir radieux et plein de bonnes promesses.*

*A tous les gens qui ont cru en moi et qui me donnent l'envie d'aller en avant,*

*Je vous remercie tous, votre soutien et vos encouragements me donnent*

*La force de continuer.*

## Table des abréviations

Abréviation	Désignation
<b>PGI</b>	Progiciel de Gestion Intégré
<b>ERP</b>	Entreprise Ressource Planning
<b>PME</b>	Ptittes et Moyennes Entreprises
<b>CRM</b>	Client Relationship Management
<b>UML</b>	Unified Modeling Language
<b>RIP</b>	Rational Unified Process
<b>XP</b>	eXtreme programming
<b>XML</b>	eXtended Markup Language
<b>2TUP</b>	Tow Track Unified Process
<b>BSD</b>	Berkeley Software Distribution
<b>SGBDRO</b>	Système de Gestion de Base de Données Relationnelle et Objet
<b>SQL</b>	Structured Query Language
<b>MVC</b>	Modèle Vue Contrôleur
<b>ORM</b>	Object Relational Mapping
<b>MTT</b>	Multi Information Technology
<b>BPM</b>	Business Precess Management
<b>SAP</b>	Systems, applications, and products for data processing
<b>A, AN</b>	Alpha, Alphanumérique

## Liste des figures

Figure 1 : Architecture technique d'un ERP [Fleur-Anne-Blain].....	7
Figure 2 : Les modules de base d'un ERP [Fleur-Anne-Blain] .....	8
Figure 3 : Architecture modulaire d'open ERP.....	16
Figure 4 : Architecture Client-Serveur d'open ERP.....	17
Figure 5 : Structure d'un module OpenERP Python.....	18
Figure 6 : Modèle vue contrôleur.....	19
Figure 7 : Organigramme de la faculté sciences de la nature et de la vie.....	22
Figure 8 : Axes de changements imposés au système informatique.....	29
Figure 9 : Cycle de développement en Y.....	30
Figure 10 :logo de python.....	47
Figure 11 : logo de postgresQL.....	48
Figure 12: Fonctionnalités ressources humaines - Liste des employés.....	51
Figure 13 : Interface d'ajout d'un employé.....	51
Figure 14 : Interface ajouter un employé - Grade .....	52

---

Figure 15: Interface d'ajouter un contrat.....	52
Figure 16 : Interface d'ajouter un recrutement.....	53
Figure 17 : Interface pour créer une feuille de temps.....	54
Figure 18: Interface pour créer notes de frais.....	54
Figure 19 : Interface pour demander un congé.....	55
Figure 20 : Interface pour demander l'attribution d'un congé.....	55
Figure 21 : Interface pour créer les types de congés.....	56
Figure 22 : Interface pour créer les postes.....	57
Figure 23 : Interface pour créer des départements.....	57
Figure 24 : Connexion au système.....	59
Figure 25 : Accueil OpenERP - client web.....	59
Figure 26 : Liste des utilisateurs.....	60
Figure 27 : Formulaire des droites d'accès.....	61
Figure 28 : Interface d'ajouter un nouveau stage .....	61
Figure 29 : Interface créer les types de stage.....	63
Figure 30 : Interface créer des types de travail.....	63
Figure 31 : Service de messagerie.....	63

---

## Liste des diagrammes

Diagramme 1 : Diagramme de contexte.....	33
Diagramme 2 : Diagramme des cas d'utilisation de l'acteur administrateur système.....	35
Diagramme 3 : Diagramme des cas d'utilisation de l'acteur Responsable RH.....	36
Diagramme 4 : Diagramme des cas utilisation détaillée.....	37
Diagramme 5 : Diagramme de séquence de cas d'utilisation « Authentification » .....	38
Diagramme 6 : Diagramme de séquence de cas d'utilisation « Ajouter stage » .....	39
Diagramme 7 : Diagramme de séquence de cas d'utilisation « modifie stage » .....	40
Diagramme 8 : Diagramme de séquence de cas d'utilisation « supprimer stage » .....	41
Diagramme 9 : Diagramme des classes.....	42

---

## Liste des tableaux

Tableau 1 : Liste des fonctionnalités requises 1 .....	25
Tableau 2 : Comparaison des méthodes de développement.....	28
Tableau 3 : Liste des fonctionnalités requises 2.....	32
Tableau 4 : Dictionnaire de données.....	43

---

---

# Table de matières

Dédicace.....	I
Remerciement.....	II
Liste des abréviations.....	III
Liste des figures.....	III
Liste des Diagrammes.....	V
Liste des tableaux.....	VI
Table de matières .....	VII
<b>Introduction générale .....</b>	<b>1</b>
<b>Chapitre I : Généralités</b>	
Partie I : L'état de l'art des ERP .....	2
1. Introduction.....	3
2. Les progiciels de gestion intégrée (PGI) .....	3
2.1 Historique des ERPP.....	3
2.2 Principe de base d'un ERP.....	4
2.3 Caractéristiques d'un ERP.....	4
2.4 Les avantages et les inconvénients.....	5
2.4.1 Les avantages .....	5
2.4.2 Les inconvénients.....	6
3. Serveur ERP.....	6
4. Architecture d'un ERP.....	7
4.1 Architecture technique d'un ERP.....	7
4.2 Architecture fonctionnelle d'un ERP.....	8
4.2.1 Module comptabilité.....	8
4.2.2 Module achats.....	9
4.2.3 Module ventes.....	9
4.2.4 Module ressources humaines .....	9
5. Les principaux éditeurs d'un ERP.....	9



---

5.1 Les ERP propriétaires .....	9
5.1.1 SAP.....	10
5.1.2 peopleSoft.....	10
5.1.3 SAGE.....	10
5.2 Les ERP Open source.....	10
5.2.1 Aria Maestosa.....	10
5.2.2 Compriere .....	11
5.2.3 ERP5.....	11
5.2.4 FISRERRA.....	11
5.2.5 OFBiz.....	12
5.2.6 OpenERP.....	12
<b>Partie II : Présentation d’OpenERP.....</b>	<b>13</b>
1. Introduction.....	14
2. Pourquoi PGI Opensource.....	14
3. OpenERP.....	14
4. Architecture technique d’OpenERP.....	16
4.1 Architecture modulaire.....	16
4.2 Architecture client/serve.....	17
4.3 Le Framework développement OpenObject.....	18
4.4 Modèle vue contrôle .....	18
5. Conclusion .....	19
<b>Chapitre II : contexte général du projet</b>	
1. Introduction .....	20
2. Présentation de l’organisme d’accueil .....	20
2.1 Historique de l’université de Béjaïa.....	20
2.2 Présentation de la faculté des sciences de la nature et de la vie.....	21
2.3 L’organigramme de la Faculté des Sciences de la Nature et de la Vie.....	22
2.4 Les missions de la Faculté des Sciences de la Nature et de la Vie.....	23
2.5 Présentation du service d’accueil.....	23

---

3. Cadre général du projet.....	23
3.1 Présentation générale de projet.....	23
3.2 Problématique.....	23
3.3 Objectif du projet.....	24
3.4 Elaboration de Cahier des charges fonctionnel.....	25
3.5 Planification du projet.....	25
4. Conclusion .....	26
 <b>Chapitre III : Analyse et conception</b>	
1. Introduction.....	27
2. Analyse et méthodologie de travail.....	27
2.1 Analyse du projet.....	27
2.2 Choix de la méthodologie de conception.....	27
2.3 Présentation du processus 2TUP.....	29
2.3.1 L'axe fonctionnel comporte.....	29
2.3.2 L'axe technique, quant à lui, comporte.....	29
2.3.3 Cycle de vie du modèle 2TUP.....	30
2.4 UML (Unified Modeling Langage) .....	31
3. Etude fonctionnelle.....	31
3.1 Etude de l'existant.....	31
3.2 Le système fonctionnel.....	31
3.3 Diagramme de contexte .....	32
3.3.1 Les acteurs du système .....	32
3.3.2 Analyse des cas d'utilisations .....	33
4. Conception du projet .....	34
4.1 Cas d'utilisations .....	35
4.1.1 Cas d'utilisation de l'administration du système.....	35
4.1.2 Cas d'utilisation de Responsable RH.....	36
4.2 Diagramme de séquence.....	37
4.2.1 Diagramme de séquence d'authentification.....	37

---

4.2.2 Diagramme de séquence de cas d'utilisation « ajouter stage ».....	39
4.2.3 Diagramme de séquence de cas d'utilisation « modifier stage ».....	40
4.2.4 Diagramme de séquence de cas d'utilisation « supprimer stage ».....	41
4.3 Diagramme des classes.....	42
4.4 Description des classes.....	43
4.5 Dictionnaire des données .....	43
4.6 Le modèle logique de données.....	44
5. Conclusion.....	44
 <b>Chapitre IV : Développement du module « Stage »</b>	
1. Introduction .....	46
2. Les technologies mises en œuvre.....	46
2.1 ubuntu .....	46
2.2 python.....	47
2.3 PostgreSQL .....	48
2.4 XML (eXtensible Markup Language) .....	47
2.5 Power AMC .....	49
3. Module ressources humaines sous Openerp.....	50
3.1 Gestion des ressources humaines.....	50
3.1.1Gestion des employés .....	50
3.1.2 Création d'un nouvel employé.....	51
3.1.3 Contrats .....	52
3.2 Gestion de recrutement .....	53
3.2.1 Candidature.....	53
3.3 Gestion du temps .....	53
3.3.1 Feuilles de temps / feuille de présence.....	53
3.4 Gestion notes de frais .....	54
3.5 Gestion des congés.....	54
3.6 Gestion configuration .....	56
3.6.1 Poste .....	56

---

3.6.2 Département.....	57
3.6.3 Contrat.....	57
3.6.4 Recrutement.....	57
4. Module « Stage » .....	58
4.1 Installation .....	58
4.2 Paramétrage .....	60
4.3 Gestion de base.....	60
4.4 Gestion des stages.....	61
4.4.1 Ajouter un nouveau stage.....	61
4.4.2 Menu configuration .....	62
5. Service de messagerie .....	63
6. Conclusion .....	64
<b>Conclusion générale et perspectives.....</b>	<b>65</b>
Webographie et Bibliographie	
ANNEXE :	
ANNEXE A : méthode installation	
ANNEXE B : développement de module openerp avec le framework « open object »	

# Introduction générale

Le présent rapport est le fruit de mon travail effectué au sein de la faculté des sciences de la nature et de la vie, ce stage est le couronnement de ma formation pour obtenir le diplôme de Master professionnel en Administration et sécurité des réseaux à l'Université de Bejaia.

Durant ces dernières décennies, l'informatique de gestion a subi des bouleversements considérables. Les avancées technologiques du traitement de l'information ont eu des conséquences capitales sur le rôle de l'outil informatique. Si les premières applications ont permis d'automatiser les activités opérationnelles des organisations (gestion de production, gestion commerciale et financière, ressources humaines), aujourd'hui les systèmes d'information prennent en charge des niveaux de gestion de plus en plus stratégiques.

Les ERP (Enterprise Resource Planning) ou PGI (Progiciels de Gestion Intégrés) ont connu leur essor en profitant de l'évolution nécessaire des systèmes d'information pour le passage de l'an 2000 puis pour la mise en place de l'euro. En effet, il était séduisant de remplacer tous les logiciels de gestion de l'entreprise par un intégré offrant « l'état de l'art » plutôt que d'engager des corrections des programmes existants plus ou moins anciens.

Mon Projet de Fin d'Etudes s'articule autour d'OpenERP, un PGI open-source extrêmement modulaire est a pour but d'adapter puis intégrer cette solution pour permettre la gestion des stages au sein de la Faculté des sciences de la nature et de la vie.

Notre travail consiste à effectuer d'abord une analyse du besoin afin de bien souligner les différentes fonctionnalités requises ensuite à chercher parmi ces fonctionnalités celles qui sont déjà offertes par OpenERP, comme ça nous pourrons entamer le développement des fonctionnalités restantes qui n'existent pas encore, pour enfin faire des tests de simulation, détecter et corriger les bugs si jamais il y en a.

Ce rapport est scindé en quatre chapitres : le premier chapitre se compose de deux parties, la première présente les ERP en général et la seconde concerne la présentation d'OpenERP. Le deuxième chapitre définit le contexte général du projet puis le troisième chapitre est autour de l'analyse et la conception du projet, alors que le dernier décrit les choix des technologies utilisées pour les différentes étapes de développement de notre application.

En fin, ce rapport se termine par une conclusion sur l'apport du travail réalisé et des perspectives futures où il peut déboucher.

# PARTIE I

« Etat de l'art sur les ERP »

## **1. Introduction**

De nos jours, savoir gérer de plus en plus d'information, sous contrainte de délai est un défi important pour chaque entreprise. Pour relever de tels défis, plusieurs solutions ont été proposées tel que, l'utilisation d'un ERP (Entreprise Ressource Planning ou PGI (Progiciel de gestion intégré), capable d'intégrer tous les processus de gestion en un système d'information unique et cohérent.

## **2. Les progiciels de gestion intégrée (PGI)**

Les entreprises d'aujourd'hui doivent rationaliser leur pilotage vu l'évolution progressive et la compétitivité féroce du marché. Pour ce faire, l'utilisation d'un progiciel de gestion intégrée est une nécessité car il centralise, présente les données de façon pertinente et affecte les bons processus aux bons acteurs.

L'acronyme PGI signifie "Progiciel de Gestion Intégrée" traduit en anglais par Enterprise Resource Planning (ERP). ERP est le terme le plus couramment utilisé. Un ERP est un progiciel qui assure la gestion automatique de l'ensemble des processus d'une entreprise comme la gestion des ressources humaines, la gestion comptable, la gestion des ventes, l'approvisionnement, la production ou encore du e-commerce [1].

Le principe fondateur d'un ERP est l'utilisation des applications informatiques correspondantes aux divers processus métiers de manière modulaire, en partageant une base de données unique et commune au sens logique. L'autre principe qui caractérise un ERP est l'usage de ce qu'on appelle un moteur de Workflow (il permet d'acheminer une donnée qui est enregistrée dans le système d'information(SI) vers les autres modules qui en ont besoin).

### **2.1 Historique des ERP**

Les logiciels d'ERP se sont développés parallèlement à l'introduction de l'informatique dans la gestion d'entreprise durant les années 70. Chaque entreprise développait son propre système de gestion via son service informatique, aucune communication n'était mise en place entre les différents services. Ce qui s'avère un véritable problème pour le pilotage des grandes organisations. Ce fonctionnement engendre de nombreux problèmes tels que la saisie redondante de données, la multiplication de données identiques, l'incohérence de données entre les différents secteurs, les données manquantes, les données non mises à jour ...etc. Dans les années 1980 l'ERP se développe et se place en tant que standard dans les grandes

entreprises. Il évolue vers des solutions d'uniformisation des données pour tous les services et centralise ces informations dans une base commune. Ainsi, ce sont de nombreux problèmes de gestion des données qui sont résolus. L'environnement de travail se standardise aussi et permet une vision et surtout une gestion plus globale de l'ensemble des processus de l'entreprise.

La popularisation d'internet dans les années 1990 apporte de nouvelles évolutions aux domaines de l'ERP. Pour commencer, internet offre un accès à distance aux logiciels de gestion d'entreprise, ainsi la gestion des données peut-être réalisée par une personne externe à l'entreprise, mais partie prenante du circuit commercial (fournisseur, sous-traitant, filiale, client...etc.). De plus, un nouveau type d'entreprise (e-commerce) se développe dont la structure et les besoins opérationnels sont différents des entreprises traditionnelles. En effet, la taille des entreprises d'e-commerce est plus réduite, mais leur champ d'action est planétaire. Par conséquent, la gestion des stocks est importante et le type de relation client est nouveau (relation client à distance). Les ERP ont évolué avec les nouveaux usages des entreprises et se sont allégés pour répondre aux besoins de structure plus petite [2].

## **2.2 Principe de base d'un ERP**

A l'inverse d'une architecture d'application usuelle (applications dédiées), les données sont ici standardisées et partagées, ce qui élimine les saisies multiples et évite l'ambiguïté liée à l'existence de données multiples de même nature assurant ainsi une cohérence de l'information.

Un ERP se caractérise également par l'usage systématique d'un moteur de workflow (système permettant d'automatiser un flux d'information au sein d'une organisation) pas toujours visible par l'utilisateur, il permet, lorsqu'une donnée est entrée dans le système d'information, de la propager dans tous les modules de système qui en ont besoin, selon une programmation prédéfinie.

## **2.3 Les caractéristiques d'un ERP**

Un ERP présente plusieurs caractéristiques, comme [3] :

- Un environnement applicatif unique, quel que soit le domaine : l'interface utilisateur est la même quelle que soit l'application,
- Des référentiels partagés, des traitements qui travaillent en cohérence,



- Une standardisation des processus, des règles de gestion qui s'harmonisent entre les divers services de l'entreprise,
- Une accélération des procédures dans lesquelles interviennent plusieurs décideurs grâce au workflow,
- Une intégration dans l'espace de travail des utilisateurs : outils bureautiques messagerie,
- Une interface utilisateur disponible dans de nombreuses langues,
- Des outils d'analyse (EIS : Executive Information Systems) et de reporting sophistiqués,
- Une ouverture sur le mode extérieur : liaisons directes (d'ordinateur à ordinateur) avec les clients et les fournisseurs, accès direct à Internet.

## **2.4 Les avantages et les inconvénients d'un ERP :**

Un ERP présente aussi bien des avantages que des inconvénients. Nous pouvons les citer comme suit :

### **2.4.1 Les avantages**

Implanter un PGI a ses avantages, parmi lesquels on cite :

- Optimisation des processus de gestion (flux économiques, flux financiers, stocks)
- Cohérence et homogénéité des informations stockées (un seul fichier articles, un seul fichier clients, etc.),
- Intégrité et unicité du système d'information facilitant la communication interne et externe,
- mise à disposition d'un outil multilingue et multidevises (très adapté aux multinationales)
- communication interne et externe facilitée par le partage du même système d'information
- meilleure coordination des services et donc meilleur suivi des processus (meilleur suivi de commande ou meilleure maîtrise des stocks par exemple)
- normalisation de la gestion des ressources humaines (pour les entreprises gérant de nombreuses entités parfois géographiquement dispersées)
- minimisation des coûts (formation et maintenance)
- maîtrise des coûts et des délais de mise en œuvre et de déploiement du logiciel

- mise à disposition, des cadres supérieurs, d'indicateurs nettement plus fiables que lorsqu'ils étaient extraits de plusieurs systèmes différents.

#### **2.4.2 Les inconvénients**

Les ERP ne sont cependant pas exempts d'inconvénients. Nous allons voir quels sont les risques possibles :

- Risques organisationnels : le progiciel et l'organisation de l'entreprise doivent cohabiter.
- Risques de mise en œuvre : les employés et utilisateurs doivent être formés à cet outil.
- Risques fonctionnels : il est possible que les fonctions offertes par le progiciel par rapport aux fonctions attendues par l'entreprise ne soient pas en accord.
- Risques techniques : la mise en place d'un ERP est un projet qui réorganise et bouleverse le SI.
- Risques économiques : du fait de l'investissement financier. Un ERP nécessite des coûts élevés de mise en place.

### **3. Serveur ERP**

Un serveur ERP est un système similaire. On y stocke l'ensemble des informations à mettre à la disposition des utilisateurs des différents modules ERP. Ce serveur est au cœur de l'architecture ERP puisqu'il est utilisé pour :

- administrer le réseau,
- gérer l'accès aux informations,
- gérer les connexions des différents utilisateurs,
- mutualiser les informations,
- assurer la traçabilité.

Son atout majeur : chaque utilisateur peut profiter de la même base de données actualisée en temps réel [2].

## 4. Architecture d'un ERP

L'ERP se base sur une architecture technique client/serveur et sur une architecture modulaire, utilisant une base de données commune, la donnée est donc unique et partagée par les modules qui ont en l'utilité (au moyen d'un moteur workflow). Les informations sont visualisées ou modifiées par les différentes parties.

### 4.1 Architecture technique d'un ERP

Les ERP présentent une structure informatique de type « client/serveur » à trois niveaux :

- ✓ **Le niveau « Présentation »** : il constitue l'interface utilisateur.
- ✓ **Le niveau « Applications »** : il correspond aux fonctions de traitement de l'information.
- ✓ **Le niveau « Base de données »** : il gère les grands volumes de données que l'entreprise conserve.

De plus, les ERP sont compatible pack Office, en particulier pour PowerPoint et Excel. Ils sont aussi adaptés à des outils de reporting.

Les ERP peuvent travailler dans des environnements hétérogènes en ce qui concerne les matériels et les logiciels de base : l'entreprise peut choisir les fournisseurs des matériels, des gestionnaires de bases de données et des systèmes d'information.

La figure suivante illustre l'architecture technique d'un ERP :

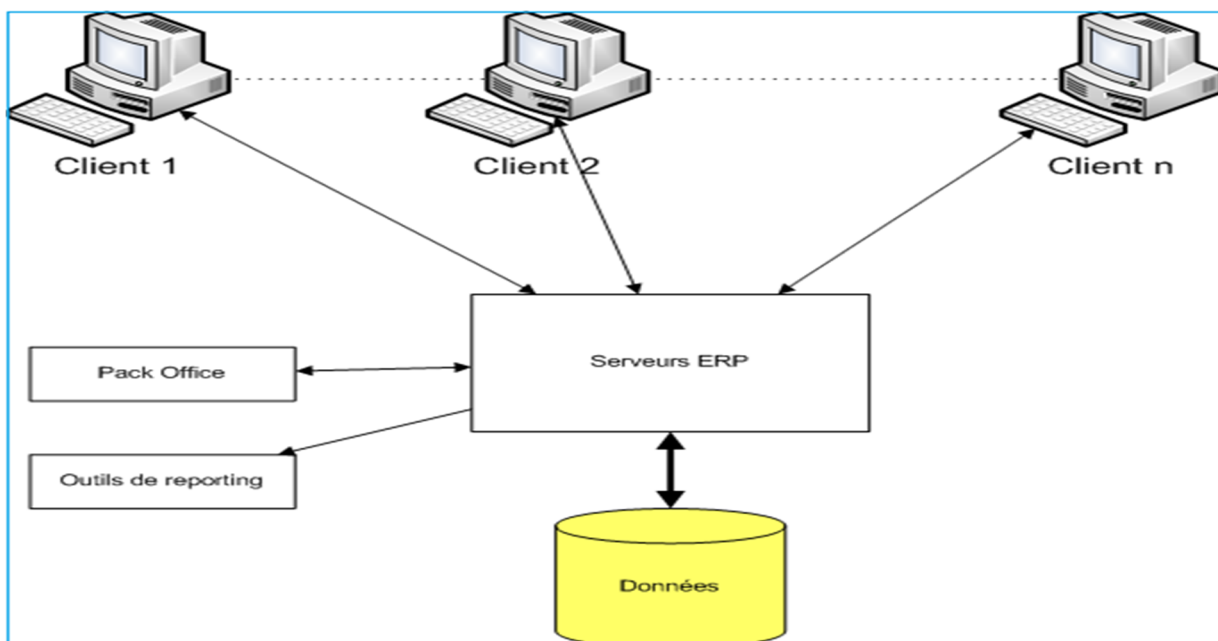


Figure 1 : Architecture technique d'un ERP [Fleur-Anne-Blain]

## 4.2 Architecture fonctionnelle d'un ERP

Un ERP est un ensemble dont toutes les parties fonctionnent les unes avec les autres d'où l'ergonomie et l'unicité des informations et donc la cohérence du système d'information.

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

Voici un exemple d'architecture modulaire qui tend à représenter tous les ERP:

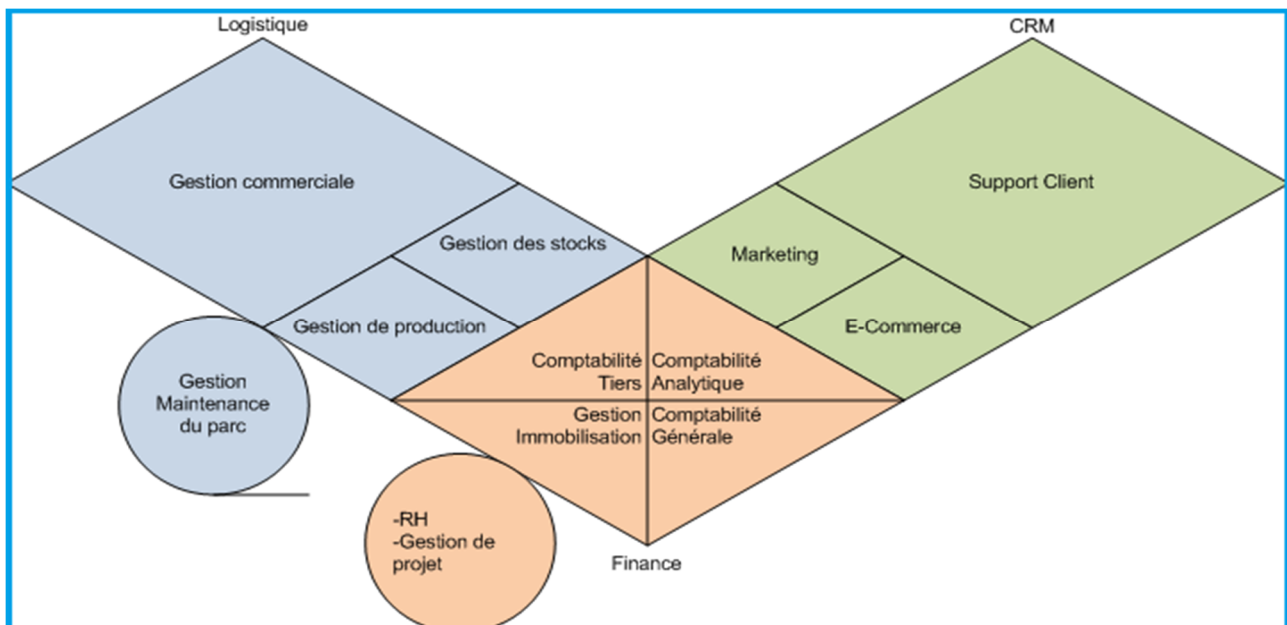


Figure 2 : Les modules de base d'un ERP [Fleur-Anne-Blain]

Dans la suite on présentera les principaux modules couverts par un ERP.

### 4.2.1 Le module comptabilité

Il s'agit au moins de la comptabilité analytique qui s'appuie éventuellement sur une infrastructure de business intelligence embarquée par l'ERP.. Néanmoins, un pont comptable d'export d'écritures peut être mis en place pour utiliser une gestion comptable abordable mais éprouvée tout en conservant les outils d'analyse, de facturation et de gestion commerciale de l'ERP open source.

#### **4.2.2 Le module achats**

Le module d'achat permet de gérer les transactions d'achat et écritures comptables associées, mais aussi les approvisionnements selon des politiques à paramétrer et/ou selon le calcul des besoins déterminés par la gestion de production.

#### **4.2.3 Le module ventes**

Écritures comptables des ventes mais aussi : règles de pricing, devis, factures, paiements, etc. Certains ERP, vont aussi très loin dans le CRM (Customer Relation Management) ou GRC (Gestion de la Relation Client). Dans certains cas, l'ERP peut intégrer une plateforme d'e-commerce native. Mais plus généralement l'ERP disposera de web services et/ou des connecteurs SQL permettant d'interfacer des logiciels d'e-commerce standard. Parfois encore, les ERP s'interfacent nativement avec des solutions de ventes en caisse POS (Point Of Sale) ou encore Point de Vente en français.

#### **4.2.4 Le module ressources humaines**

Le périmètre du module ressources humaines peut varier de la gestion des emplois du temps, au recrutement, en passant par la gestion de la paie. A noter que les modules de paie sont très rares à cause du morcellement législatif d'une part et de la mise en jeu de données très confidentielles d'autre

### **5. Les principaux éditeurs d'un ERP :**

Le marché de l'ERP représente une vraie manne pour les prestataires de services informatiques. Il est devenu aujourd'hui, le marché le plus porteur de l'informatique.

On distingue deux types d'ERP : les ERP propriétaires, édités par des sociétés, ce qui implique l'achat d'une licence, et les ERP open source qui sont « gratuits ».

#### **5.1 Les ERP propriétaires**

Aujourd'hui, il existe de nombreux ERP propriétaires mais seuls quelques éditeurs internationaux restent vraiment dans la course. Ils ne font pas inclure, dans le calcul du coût d'acquisition total, les frais de maintenance et de l'assistance technique. Les trois premiers sont :

### **5.1.1 SAP**

Fondée en 1972, SAP est le premier éditeur mondial de progiciels ERP. SAP emploie plus de 34.000 personnes dans plus de 50 pays. Le progiciel SAP a remporté rapidement un succès important auprès des grandes entreprises en proposant un progiciel multilingue et multidevises. SAP s'intéresse aussi au marché des PME, en pleine croissance en proposant sa suite BusinessOne, pour les entreprises de 2 à 250 salariés.

SAP est une application client-serveur. Ses modules couvrent l'ensemble des fonctions de gestion de l'entreprise et chaque module couvre des besoins complets de gestion. Certaines entreprises implémentent tous les modules fonctionnels de SAP, ou seulement quelques-uns

### **5.1.2 PeopleSoft**

Les applications PeopleSoft d'Oracle sont destinées à répondre aux besoins les plus complexes de l'entreprise. Elles offrent des solutions commerciales et industrielles complètes qui confèrent aux organisations une productivité accrue, un gain de performances et un coût de propriété inférieur.

### **5.1.3 SAGE**

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

## **5.2 Les ERP OpenSource**

Ils sont relayés par des partenaires (SSII, cabinets de Conseil) pour le support. L'implémentation d'un progiciel Open Source revient moins chère, puisqu'il n'y a pas de coût de licence. En revanche il faut inclure, dans le calcul du coût d'acquisition total, les frais de maintenance et de l'assistance technique. Voici la liste des principaux progiciels Open Source :

### **5.2.1 Aria Maestosa**

Est un éditeur MIDI opensource multi-plateformes (Mac, Windows et Linux) qui permet de créer, éditer et lire des fichiers MIDI. Les pistes peuvent être présentées sous forme de partition, clavier, guitare ou batterie.

L'export des partitions se fait dans les formats MIDI ou AIFF.

Ce logiciel sous son apparence encore simple offre les principales possibilités souhaitées de ce type de logiciel :

- \* création de multiples pistes, qui peuvent être nommées ;
- \* choix de l'instrument, du canal, du volume, etc. pour chaque piste ;
- \* choix de la clé, tonalité, de la mesure, du tempo ;
- \* possibilité d'affichage sous forme de tablature de guitare ;
- \* gestion des percussions ;
- \* piste de gestion des contrôleurs MIDI (volume, pan, sustain...).

### **5.2.2 Compiere**

Est un progiciel de gestion intégré (PGI) et gestion de la relation client (GRC) à source ouvert : open source pour les Petites et moyennes entreprises (PME) dans la distribution et le service. L'application est fournie sous double licence GPL et propriétaire<sup>1</sup>. Les sources peuvent être adaptées aux besoins du client. Le support technique et la documentation sont payants.

### **5.2.3 ERP5**

Est un PGI Progiciel de Gestion Intégré libre basé sur le serveur d'application Zope et développé en Python. Il intègre la comptabilité, la gestion de la relation client (CRM), des achats, des ventes, des stocks, des données produits (PDM), de la production (SCM), de la logistique et des ressources humaines etc.

ERP5 gère le variantage de toutes les ressources manipulées dans l'application selon le nombre souhaité d'axes de variantage. Cette gestion du variantage offre des fonctionnalités avancées de configuration des produits, réduit les coûts de mise au point et de maintenance des nomenclatures et gammes opératoires et autorise des approches complexes de tarification.

### **5.2.4 FISTERRA**

Est un projet qui vise à fournir un ERP open-source générique (Enterprise Resource Planning). Aujourd'hui, il prend en charge la facturation, les stocks et le paiement gestion, POS (Point-Of-Sale), travail distribué et la réplication en mode hors connexion. Il utilise Gnome SDK et PostgreSQL.

### **5.2.5 OFBiz (Open for Business)**

Est un projet de logiciel d'automatisation d'entreprise open source sous licence Apache License, version 2.0. Par ouverte automatisation de l'entreprise source, nous entendons: Open Source ERP (Enterprise Resource Planning), CRM Open Source (Gestion de la relation client), Open Source E-Business / e-Commerce Open Source SCM (gestion de la chaîne d'approvisionnement), l'Open Source MRP (Manufacturing la planification des ressources), Open Source GMAO / EAM (gestion de la maintenance / Enterprise Asset Management), Open Source POS (Point Of Sale), et ainsi de suite.

### **5.2.6 OpenERP (TinyERP)**

A été développé principalement pour les petites et moyennes entreprises de 5 à 150 utilisateurs. Grâce à son amélioration permanente et à ses nouvelles fonctionnalités, il peut désormais répondre aux besoins d'entreprises de taille beaucoup plus conséquente. Il est flexible et évolutif. L'objectif est que le logiciel s'adapte aux besoins de l'entreprise et non l'inverse.

Le 1er février 2005, TinyERP commence à offrir un support de deuxième niveau et des formations liées à l'ERP. Un réseau de revendeur est créé, en France notamment.

En 2006 Axelor et Tiny signe un partenariat pour le développement conjoint de la version web de Tiny ERP.

Janvier 2008 : sortie officiel de la version web de TinyERP, coéditée par Axelor et Tiny.

En 2008, TinyERP devient OpenERP et devient l'ERP Open Source le plus adoptés au monde.

Après la présentation globale d'ERP nous nous intéressons à la catégorie des ERP open source et plus particulièrement à OpenERP que nous allons présenter dans le partie suivante.



## PARTIE II

# « Présentation d'OpenERP »

## 1. Introduction

Le marché des ERP pour PME est en pleine expansion. Comme sur tous les marchés, les acteurs se sont d'abord intéressés aux plus grands clients. Concernant les ERP, les éditeurs visaient dans un premier temps les multinationales. Une fois ce marché saturé, ils ont racheté des solutions pour PME afin de mettre un pied dans ce secteur d'activités. Les logiciels libres ont ciblé les petites structures dès le départ. Développés en étroite collaboration avec les utilisateurs et les intégrateurs, ces solutions sont parfaitement adaptées à ces entités.

## 2. Pourquoi un PGI Open Source ?

D'une manière générale, en utilisant un produit open source, nous pouvons s'attendre à des économies de licence en installant un ERP open source. En effet, l'ERP étant un progiciel complexe, les coûts d'intégration et de maintenance représente le coût total de possession de l'ERP. Ainsi, l'économie d'une licence propriétaire représenterait entre 25% et 50% du coût total d'implémentation d'un ERP.

Par ailleurs, les solutions open source arrivent à leur maturité. Ce qui fait, les petites et moyennes entreprises (PME) et les très petites entreprises (TPE) peuvent disposer d'un outil de gestion complet au meilleur coût en leur apportant rapidement un vrai bénéfice en termes de compétitivité [4].

## 3. OpenERP

Anciennement connu sous le nom Tiny ERP, signifiant la fourmi de l'Enterprise Resource Planning) est un progiciel intégré de gestion ouvert, libre de droits comprenant les ventes, la gestion de relation client (CRM), la gestion de projet, la gestion d'entrepôt, la production, la comptabilité et les ressources humaines. OpenERP à trois composants séparés :

- ✓ Le serveur openerp-server qui stocke ses données dans une base postgresql,
- ✓ Le client openerp-client qui s'installe sur le poste de l'utilisateur,
- ✓ Le serveur web openerp-web qui permet une utilisation depuis un navigateur.

Ces trois composants communiquent par les protocoles xml-rpc et net-rpc.

Le logiciel est basé sur une forte architecture MVC, des flux de travail flexibles, une interface-utilisateur graphique dynamique, une interface XML-RPC, et un système personnalisable de comptes rendus avec une intégration pratique d'OpenOffice.org.

Dans la classification des logiciels, OpenERP, comme tout autre PGI sur le marché est un package destiné, a priori, à tous les secteurs, à toutes les fonctions, les adaptations nécessaires se faisant par paramétrage.

Il dispose de forts arguments commerciaux pour séduire les dirigeants, Cette offre séduisante par sa qualité et sa cohérence se révèle à l'usage plus risquée que l'on avait pu l'imaginer : elle ne peut être efficace que si l'on accepte les contraintes qu'elle impose. Sa mise en œuvre comporte des difficultés et des pièges.

Comme tous les ERP, OpenERP possède des avantages et des inconvénients que nous résumerons afin de vous permettre une meilleure visualisation

### **Avantages**

- Éditeur très dynamique
- Communauté dynamique et expérimentée
- Périmètre fonctionnel inégalé avec ses quelques 500 modules et des 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.
- Vrai ORM 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.
- Grand souplesse générale, notamment grâce à la scriptabilité des rapports.
- Les coûts d'intégration les plus faibles

### **Inconvénients :**

- Lourdeur et rigidité de mise en œuvre
- Difficultés d'apprentissage pas le personnel
- Progiciel souvent sous-utilisé, son panel fonctionnel est le plus souvent plus grand que les besoins de l'entreprise
- Besoin d'une maintenance régulière

### 4. Architecture technique d'OpenERP

Open ERP est caractérisé par son architecture modulaire et technique.

#### 4.1 Architecture modulaire :

Un module OpenERP est la définition d'une gestion informatisée d'un domaine. Cette architecture n'est pas propre à OpenERP. Elle est en fait partagée par tous les ERP. Il s'agit de la faculté de construire des applications informatiques de manière modulaire (modules indépendants entre eux) tout en partageant une base de données unique. Ceci apporte une importance significative puisque les données sont maintenant standardisées et partagées. Ce qui élimine les saisies multiples et évite l'ambiguïté des données de même nature. L'architecture modulaire d'open ERP lui permet de couvrir plusieurs domaines illustrés dans la figure ci-dessous :

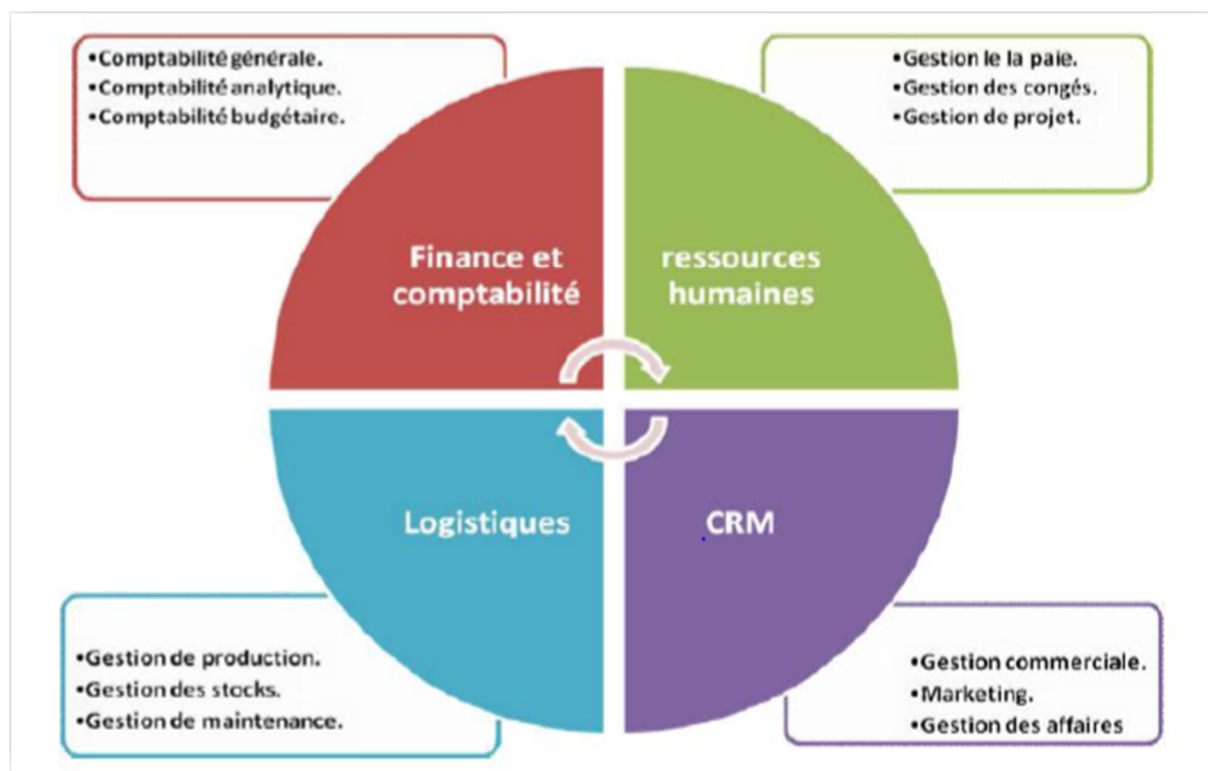


Figure 3 : Architecture modulaire d'open ERP

## 4.2 Architecture Client/serveur

OpenERP est basé sur une architecture client/serveur. Le serveur et le client communiquent via le protocole XML-RPC (Remote Procedure Call). C'est un simple protocole qui permet au client de faire des appels de méthodes à travers un réseau. Une fois la fonction est appelée, ses arguments et ses résultats sont envoyés par le protocole http, eux-mêmes sont encodés par le langage XML.

OpenERP est couplé à une base de données PostgreSQL. De plus, il est compatible au pack OpenOffice et aussi avec des outils de reporting (ReportLab) pour produire des rapports en PDF ou en HTML.

La logique d'open ERP est entièrement du côté serveur. La tâche du client se résume à demander les données (formulaire ou listes) au serveur et de les renvoyer. Avec cette approche, presque tout le développement est fait du côté serveur. Ce qui rend OpenERP plus simple au développement et à la maintenance.

L'opération client est très simple. Quand un utilisateur exécute une action (sauvegarder un formulaire, ouvrir un menu, imprimer, ...) il envoie cette action au serveur. Le serveur envoie alors la nouvelle action pour s'exécuter côté client. Il y a trois types d'actions :

- Ouvrir une fenêtre (formulaire, listes),
- Imprimer un document,
- Exécuter un wizard.

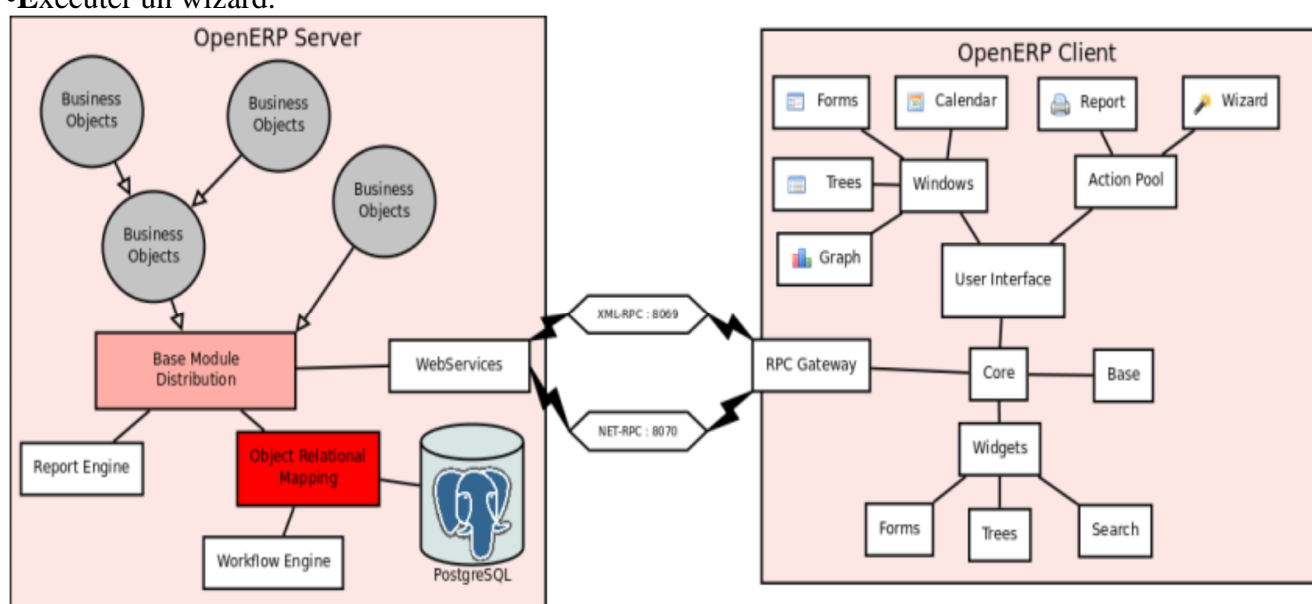


Figure 4: Architecture Client-Serveur d'openERP

## 4.3 Le Framework développement OpenObject

OpenERP offre un cadre de développement, c'est à dire des services techniques informatiques :

- Un serveur de base de données objet pour représenter et mémoriser les objets de gestion et les rendre accessible via le réseau.
- Un "workflow" qui contrôle l'évolution des objets suivant une procédure.
- Des formulaires et écrans pour l'interaction avec l'utilisateur.
- Des états imprimables des objets.

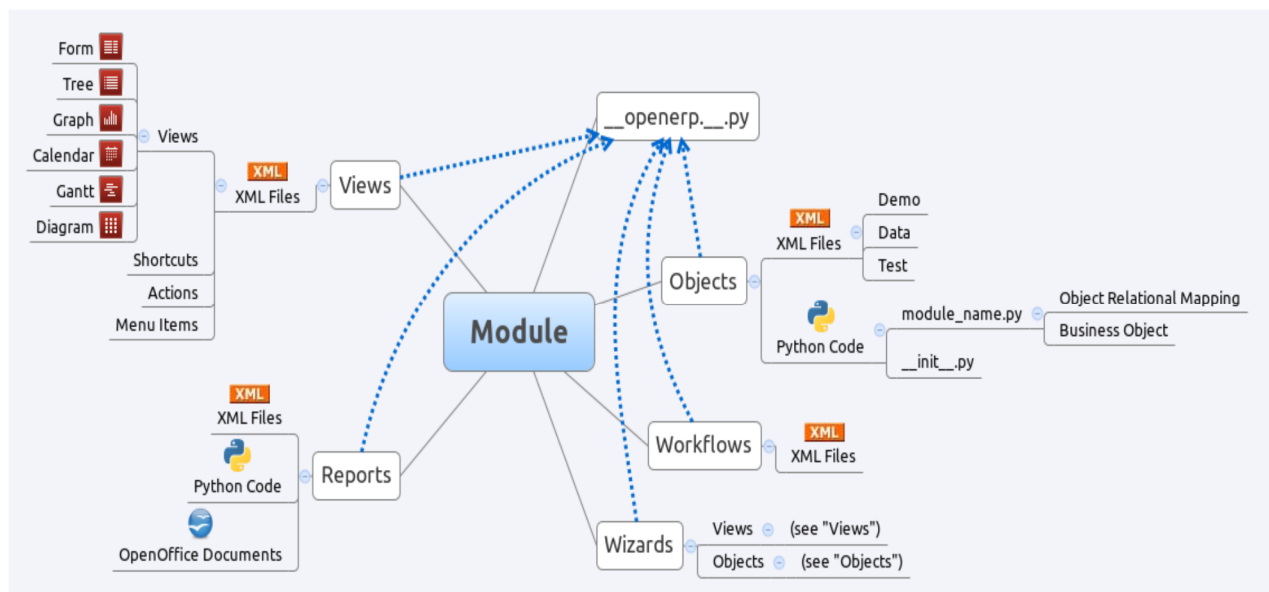


Figure 5 : Structure d'un module OpenERP Python

## 4.4 MVC : Modèle- Vue-Contrôleur

MVC est une architecture de modèles utilisée en génie logiciel. Dans des applications complexes qui présentent des lots de données aux utilisateurs, on souhaite souvent séparer les données (modèle) et l'interface utilisateur (vue), de sorte que les changements à l'interface utilisateur n'affectent pas le traitement des données, et que les données peuvent être réorganisées sans changer l'interface utilisateur. Le MVC résout ce genre de problème en découplant l'accès des données et la logique des applications de la présentation des données et de l'interaction utilisateur, en introduisant un composant intermédiaire : « le contrôleur ».

Dans open ERP, on peut appliquer cette sémantique de MVC avec :

**Model** : les modèles sont les objets déclarés dans OpenERP. Ils sont également des tables PostgreSQL.

**View** : les vues sont définies en fichiers XML dans OpenERP.

**Controller** : le contrôleur est les classes Python qui contrôle OpenERP [5].

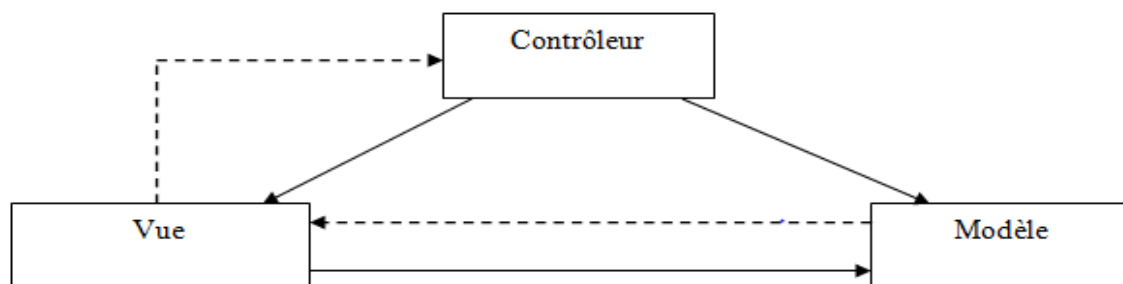


Figure 6 : Modèle vue contrôleur

## 5. Conclusion

L'avenir des ERP passe par le respect des standards qui s'imposent aujourd'hui en informatique. L'enjeu est de permettre à tous les logiciels développés dans des environnements et des langages différents de pouvoir échanger des informations. Cette interopérabilité permettra l'intégration de plus en plus prononcée des systèmes d'information au sein de l'entreprise et entre les entreprises. Les éditeurs travaillent également à augmenter la qualité des interfaces ainsi qu'à diminuer les coûts de maintenance liés aux innovations technologiques.

Dans ce chapitre, nous avons présenté deux parties : la première partie est l'état de l'art sur les ERP. Nous avons donné une définition au progiciel de gestion intégrée, son historique, nous avons cité quelques caractéristiques ainsi que les avantages et les inconvénients suivis par ses architectures. Enfin, les principaux éditeurs.

La deuxième partie est la présentation d'openerp, nous avons précisé pourquoi choisir un PGI Open Source. Nous avons donné une définition à OpenERP, de ses avantages et ses inconvénients, suivi par ses différentes architectures.

## **1. Introduction**

La gestion des ressources humaines au sein de l'Université de Bejaïa est une opération rigoureuse, qui mérite d'être perfectionnée et analysée soigneusement ; mais avant d'essayer de porter une solution informatique pour ce processus, la présentation de l'organisme d'accueil en général et le service qui gère ces mouvements au niveau de la Faculté des Sciences de la nature et de la vie en particulier est nécessaire, et c'est ce qui est conseillé d'ailleurs dans toute démarche informatique de Génie Logiciel. Toute fois, il est important de signaler à l'attention qu'afin de mieux réaliser les prochaines étapes de notre plan de travail, la spécification et la précision de notre sujet et champ d'étude doivent être bien comprises, cernées et clarifiées.

## **2. Présentation de l'organisme d'accueil**

### **2.1 Historique de l'université de Bejaia**

L'université de Bejaia a été créée en faveur de la décentralisation initiée au début des années 80 pour désengorger les pôles universitaires d'Alger, Constantine, Tizi-Ouzou et de Sétif. Bien que née tardivement (1983) par rapport à son passé scientifique et culturel de renom, l'Université de Bejaia, établissement public de formation supérieure sous tutelle du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, a pu s'agrandir et passer d'un effectif de 205 étudiants encadrés par 40 enseignants à son ouverture à un effectif de 22 792 étudiants et 698 enseignants en 2006.

Cet essor n'a pu se réaliser qu'avec une diversification judicieuse de ses filières et options. L'Université de Bejaia regroupe huit Facultés (Faculté des Sciences exactes, Faculté de technologie, Faculté de science de la nature et de la vie, faculté de droit, faculté des lettres et des sciences humaines, faculté des sciences économiques, sciences de gestion et sciences commerciales et faculté de médecine). Elle se déploie sur deux sites principaux (Targa Ouzemmour et Aboudaou). L'Université de Bejaia dispose d'une bibliothèque centrale pluridisciplinaire, d'une annexe opérationnelle depuis 2002 et d'une bibliothèque au campus d'Aboudaou ouverte en 2003. Elle couvre plus de 35 spécialités avec un fond documentaire riche et dépassant en ouvrage 146 000 volumes, 2500 titres de thèses et mémoires, 300 titres de revue dont 79 en cour d'abonnement et environ 400 CD ROM. La diversification de filière de formation, la valorisation de la recherche et le transfert de compétences lui ont permis d'être parmi les universités les plus performantes du pays.



Devant l'importance grandissante de l'informatique dans les domaines de la vie courante et sa large utilisation dans toutes les filières (technologie, économie, médecine, etc.), l'ouverture de cette spécialité est devenue indispensable. La volonté et le dynamisme des différents responsables ont conduit à l'ouverture d'une Ecole Doctorale en Informatique option Réseaux et Systèmes Distribués : c'est la première du genre dans le pays. L'université de Bejaia, à l'instar des autres établissements de l'enseignement supérieur, a mis en place le nouveau dispositif d'enseignement LMD des la rentrés 2004/2005, avec comme première étape une proposition de 33 licences dont 19 professionnelles et 14 académiques.

### **2.2 Présentation de la faculté des Sciences de la Nature et de la Vie**

La faculté des Sciences de la Nature et de la Vie a été créée le 18 septembre 2001 par le Décret N°01-268 au sein de l'université Abderrahmane Mira de Bejaia (Campus de Targa Ouzamour). Elle se situe dans le prolongement de l'ex Institut des Sciences de la Nature (1983-2001); elle est dirigée par un doyen, assisté dans ses fonctions par deux vices doyens et un secrétaire général.

2.3 L'organigramme de la Faculté des Sciences de la Nature et de la Vie

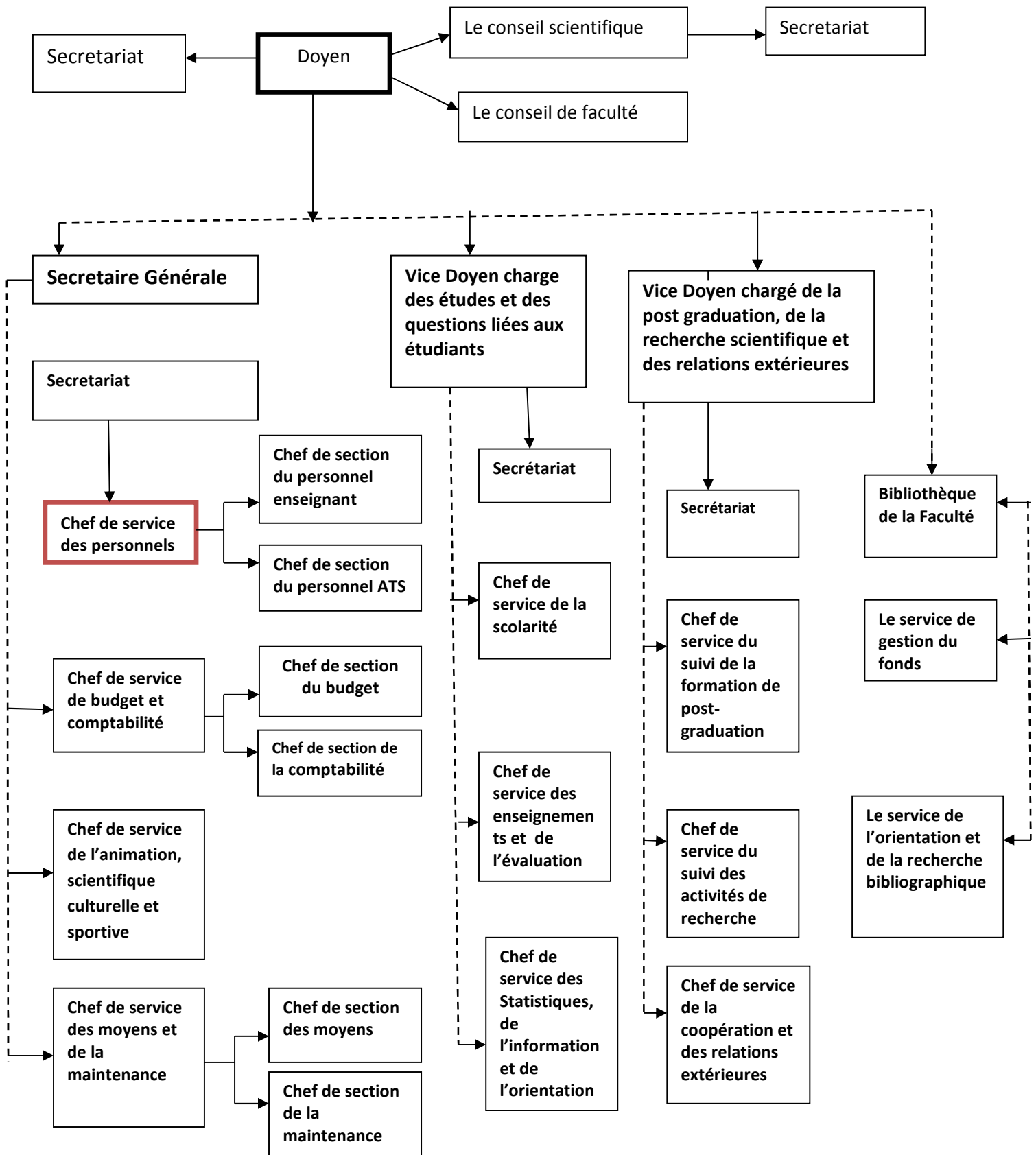


Figure 7 : Organigramme de la faculté des sciences de la nature et de la vie

## **2.4 Les missions de la Faculté des Sciences de la Nature et de la Vie**

La faculté des Sciences de la Nature et de la Vie de l'Université Abderrahmane Mira de Bejaïa se caractérise par un fort potentiel d'encadrement aussi bien en graduation qu'en post graduation. La volonté politique de la faculté des Sciences de la Nature et de la Vie est de développer l'accueil d'étudiants et enseignants nationaux et étrangers, d'aider à la mobilité des étudiants et enseignants chercheurs de la Faculté dans le cadre de conventions interuniversitaires et de les faire participer aux différents projets. Ces missions sont :

- Formations des cadres (Graduation et post graduation),
- Réalisation des recherches et des études scientifiques,
- Promotion des activités culturelles, scientifiques et sportives.

## **2.5 Présentation du service d'accueil**

Le service des personnels de la faculté des sciences de la Nature et de la Vie se subdivisent en deux structures: section du personnel ATS et section du personnel enseignant auquel sont affiliés les enseignants de la faculté, dans lequel on a effectué notre stage.

# **3. Cadre général du projet**

## **3.1 Présentation générale du projet**

Le système d'information coordonne grâce à l'information des activités de l'organisation, et lui permet ainsi d'atteindre ses objectifs. Il est le véhicule de communication de l'organisation. Il représente l'ensemble des ressources (humaines, matérielles et logicielles) organisées pour collecter, stocker, traiter et communiquer les informations au sein de la faculté.

Notre projet s'inscrit dans le cadre d'une solution optimisée de la gestion des stages des enseignants à l'étranger, à l'aide de l'ERP open source particulièrement OpenERP dans lequel nous nous basons pour adapter nos besoins et de développer des nouvelles fonctionnalités.

## **3.2 Problématique**

Le service des personnels souhaite s'ouvrir sur les nouvelles technologies et les utiliser pour avoir un meilleur rendement et ainsi intégrer des solutions de gestion complètes, paramétrables et flexibles pour la gestion de tout ce qui se rapporte au bon fonctionnement de son service.

Le service personnel de la faculté des sciences de la nature et de la vie, ne compte aucune application informatique pour gérer ses ressources humaines, que ce soit leurs carrières, activité, congés, stages, etc. Ceci génère non seulement une difficulté d'échange de données entre les différentes structures de la faculté comme le service de budget et de la comptabilité,...etc, mais également une incohérence de l'information interne.

### 3.3 Objectif du projet

Le travail demandé est d'adapter un progiciel libre, OpenERP, aux besoins de la faculté, en essayant de rajouter de nouvelles fonctionnalités que les autres solutions ne peuvent offrir.

Afin de garantir une modernisation de service des personnels, il est indispensable d'adopter les nouvelles technologies et les exploiter pour en tirer le meilleur profit.

Les PGI sont connus pour leur intégration des principales fonctions nécessaires à la gestion des flux et procédures de l'entreprise (ressource humaine, comptabilité, achats, ...), tous ces outils accèdent à des ressources communes, en particulier des bases de données.

Parmi les autres avantages des PGI, c'est qu'ils sont conçus de telle sorte qu'un "simple" paramétrage suffit à les adapter à l'organisation de l'entreprise. Il n'est normalement pas nécessaire d'effectuer des développements spécifiques, sauf en cas de nécessité, lorsque ce paramétrage ne permet pas de prendre en compte des particularités liées au processus métier de la faculté.

Ainsi, le système d'information à développer aura pour objectifs :

- Amélioration et simplification du travail administratif
- La gestion des éléments de bases, à savoir les employés, les utilisateurs avec une interface ergonomique qui facilite la navigation entre les fonctionnalités du système.
- L'optimisation de la gestion des processus métiers : gestion ressource humaine, gestion des stages.
- Permettre le suivi des employés, voir leur situation en tout moment.
- La cohérence et l'homogénéité de l'information interne.
- La possibilité de développer de nouvelles fonctionnalités.

### 3.4 Elaboration du Cahier des charges fonctionnel

Afin de mener à bien notre projet, il est bien donc nécessaire de faire une l'analyse préliminaire des besoins et de faire une liste des différentes fonctionnalités à mettre en œuvre.

Les fonctionnalités requises sont présentées dans le tableau ci-dessous :

Objectif	Fonctionnalité
Gestion de base	-Gestion des employés -Gestion des utilisateurs
Gestion de la ressource humaine	-Consulté la liste des employés -Gérer les employés -Gérer les congés
Les rapports (état de sortie)	-Imprimer les rapports (fiche de stage)
Le service de messageries	-Gestion des emails des partenaires
Gestion des stages	- Consulté les stages - Gérer les stages

Tableau 1 : Liste des fonctionnalités requises 1

### 5.3 Planification du projet

La planification consiste à prévoir le déroulement du projet tout au long des phases constituant le cycle de vie prévu. C'est ainsi qu'on peut diviser le projet de mise en place d'une solution de gestion intégrée en quatre grandes parties :

- **Description et analyse des besoins spécifiques** : les objectifs de cette phase sont de bien cerner le sujet d'une part, et de délimiter le périmètre du projet d'autre part au niveau fonctionnel.
- **L'étude comparative des solutions existantes sur le marché**: cette étude doit aboutir au choix d'une solution de gestion intégrée tout en prenant en considération l'environnement de l'entreprise et son cahier de charge.
- **La mise en production du progiciel de gestion intégrée (PGI)**: L'objectif de cette phase est de construire le système PGI cible, puis d'en vérifier la fiabilité des différents traitements. Les tâches typiques associées à cette étape sont :
  - La personnalisation et le paramétrage.
  - La mise à jour de la base de données.

- La fermeture des trous fonctionnels (si un bouton ne fait pas son rôle correctement on doit changer son fonctionnement)
  - Le développement spécifique des fonctionnalités non traitées par le progiciel.
  - La mise à jour finale de la documentation du système final.
- **La bascule** : L'objectif de cette phase est le démarrage opérationnel du progiciel sur le site. Les tâches typiques associées à cette étape sont :
- La formation des utilisateurs finaux et la documentation des utilisateurs.
  - Le pilotage : Test d'intégration sur la base des processus et validation des modes opératoires par les utilisateurs.

#### 4. Conclusion

Dans ce chapitre nous avons décrit le contexte général dans lequel s'inscrit notre projet. Au début, nous avons présenté l'organisme d'accueil de la faculté sciences de la Nature et de la Vie puis nous avons déterminé la problématique et les objectifs du projet qui se résument à la mise en place d'un système de gestion intégrée. Enfin, la description et l'élaboration du cahier des charges.

### 1. Introduction

L'étape de conception est très importante pour la réussite d'un projet informatique car elle vise à définir une feuille de route du projet, le concevoir et le valider avant de passer au développement du système. Elle permet aussi d'avoir une bonne réflexion avant de passer à l'action, une bonne organisation du travail et une bonne communication entre les différents intervenants dans le projet.

### 2. Analyse et méthodologie

#### 2.1 Analyse du projet

La complexité croissante des systèmes informatiques a conduit les concepteurs à s'intéresser aux méthodes de développement. On a comptabilisé en 1994 jusqu'à 50 méthodes [11]. Chaque méthode se définit par une notation et un processus spécifique. UML a ouvert le terrain en fusionnant la notation. Il reste cependant à définir le processus pour réellement capitaliser des règles dans le domaine du développement logiciel. Les groupes de travail UML ont donc travaillé à unifier non pas les processus mais plus exactement les meilleures pratiques de développement objet. Ces processus se distingueront par le générique «UNIFIED PROCESS» [11].

Un processus définit une séquence d'étapes, en parties ordonnées qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. Pour produire des logiciels de qualité qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles. On découpe le processus en deux axes :

- L'axe de développement technique qui concentre principalement sur la qualité de production.
- L'axe de gestion du développement qui permet la mesure et la prévision des coûts et des délais.

#### 2.2 Choix de la méthodologie de conception

Avant d'adopter une méthode, il faut d'abord faire une comparaison entre les différentes méthodes existantes, voir les points forts et les points faibles de chacune puis déterminer celle qui va mieux dans le contexte du projet. Ci-dessous un tableau qui résume cette comparaison.

Méthode	Description	Points forts	Points faibles
<b>Rational Unified Process (RUP)</b>	<ul style="list-style-type: none"> <li>-Méthodologie centrée sur l'architecture et couplée aux diagrammes UML</li> <li>-Concerne des projets de +10 personnes</li> <li>-Processus complet assisté par des outils exhaustifs</li> </ul>	<ul style="list-style-type: none"> <li>-Itératif -Spécifie le dialogue entre les différents intervenants du projet : les livrables, plannings et prototypes...</li> <li>-Propose des modèles de documents, et des canevas pour des projets types</li> <li>-Rôles bien définis, modélisation</li> </ul>	<ul style="list-style-type: none"> <li>-Coûteux à personnaliser</li> <li>-Très axé processus, au détriment du développement</li> <li>-Lourd, largement étendu, il peut être difficile à mettre en œuvre de façon spécifique</li> <li>-Convient pour les grands projets qui génèrent beaucoup de documentation</li> </ul>
<b>eXtreme Programming (XP)</b>	<ul style="list-style-type: none"> <li>-Développement guidé par les besoins du client</li> <li>-Equipes réduites, centrées sur les développeurs</li> </ul>	<ul style="list-style-type: none"> <li>-Itératif</li> <li>-Simple à mettre en œuvre</li> <li>-Laisse une large place aux aspects techniques</li> <li>-Builds journaliers</li> <li>-Amélioration constante adaptation aux modifications</li> </ul>	<ul style="list-style-type: none"> <li>-Ne couvre pas les phases en amont et en aval du développement</li> <li>-Assez flou dans sa mise en œuvre : quels intervenants ? Quels livrables ?</li> <li>-Focalisé sur l'aspect individuel du développement, au détriment d'une vue globale et des pratiques de management ou de formalisation.</li> </ul>
<b>Two Track Unified Process (2TUP)</b>	<ul style="list-style-type: none"> <li>-Articulé autour de l'architecture</li> <li>-Cycle de développement en Y</li> <li>-Convient aux projets de toutes tailles</li> </ul>	<ul style="list-style-type: none"> <li>-Itératif, laisse une large partie à la technologie et à la gestion du risque</li> <li>-Définit les profils des intervenants, les livrables, les prototypes</li> </ul>	<ul style="list-style-type: none"> <li>- Superficiel sur les phases en amont et en aval du développement</li> <li>-Aucune proposition de document type</li> </ul>

Tableau 2 : Comparaison des méthodes de développement

Pour atteindre les objectifs, nous avons suivi la méthode 2TUP (2Track Unified Process), car il apporte une réponse aux contraintes de changements continuels imposés aux systèmes d'information. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes.



### 2.3 Présentation du processus 2TUP

**2TUP** (*2 tracks unified process*) est un processus de développement logiciel qui implémente le Processus Unifié.

Le **2TUP** propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs qui vont interagir avec le système à construire, les messages qu'échangent les acteurs et le système, à produire le cahier des charges et à modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui, sur l'axe qui lie un acteur au système on met les messages que les deux s'échangent avec le sens) [10].

Ce processus suit s'articule ensuite autour de deux chemins :

- Architecture fonctionnelle
- Architecture technique

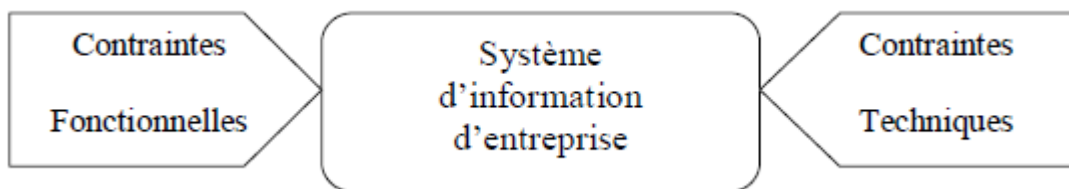


Figure 8 : Axes de changements imposés au système informatique

#### 2.3.1 L'axe fonctionnel comporte

- Capture des besoins fonctionnels qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs.
- Analyse qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

#### 2.3.2 L'axe technique comporte

- Capture des besoins techniques qui recensent toutes les contraintes et les choix dimensionnant la conception du système.
- Conception générique qui définit ensuite les composants nécessaires à la conception de l'architecture technique. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système informatique et écarte la plupart des risques du niveau technique.

- Conception préliminaire qui représente une étape délicate car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- Conception détaillée qui étudie ensuite comment réaliser chaque composant.
- Codage qui produit ses composants et teste au fur et à mesure les unités de code réalisées.
- Recette qui consiste enfin à valider les fonctionnalités du système développé.

### 2.3.3 Cycle de vie du modèle 2TUP

Le modèle 2TUP repose sur cinq principes fondamentaux :

- ✓ Séparer les aspects fonctionnels et les aspects techniques
- ✓ Travailler selon deux points de vue qui se complètent et s'enrichissent mutuellement ; celui de l'entreprise, celui des applications.
- ✓ Modéliser l'activité de l'entreprise et des applications aux moyens d'objets en utilisant UML.
- ✓ Faire des maquettes et des prototypes pour affiner les besoins fonctionnels et les aspects techniques.
- ✓ Effectuer la réingénierie des applications dans le sens de la réutilisation.

Le schéma suivant montre bien le développement en « Y » caractéristique de 2TUP :

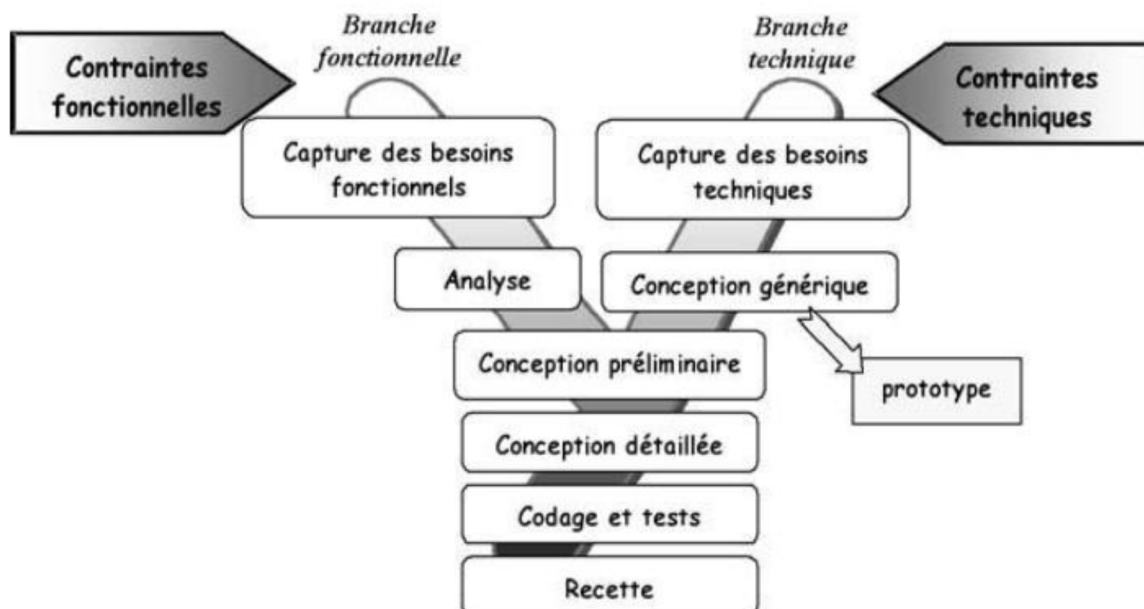


Figure 9: Cycle de développement en Y

### **2.4 UML (Unified Modeling Language)**

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie est à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. UML a une dimension symbolique et ouvre une nouvelle voie d'échange de visions systémiques précises. Ce langage est certes issu du développement logiciel mais pourrait être appliqué à toute science fondée sur la description d'un système. Dans l'immédiat, UML intéresse fortement les spécialistes de l'ingénierie système [11].

### **3. Etude fonctionnelle**

Avant de spécifier les besoins il est nécessaire de réaliser une étude de l'existant. L'étude de l'existant est une étape clé dans la réalisation de n'importe quelle application informatique quel que soit le domaine concerné. Il s'agit d'une étude permettant de comprendre la problématique du projet et de détecter les avantages et inconvénients des solutions proposées actuellement sur le marché afin d'en profiter pour la réalisation de notre projet.

#### **3.1 Etude de l'existant**

Comme tous les établissements de l'enseignement supérieur l'Université de Bejaïa offre à ses enseignants la possibilité d'effectuer des stages dans des entreprises publics ou privés à l'étranger.

Le processus de suivi des stages se fait manuellement. En effet l'échange de l'information entre le service et l'enseignant se fait par le mailing ou par papiers manuscrite. Ce qui rend la tâche du service fastidieuse et complexe et ne peut entièrement les assumer. La fatigue, l'oubli et l'erreur ont de grandes répercussions sur la qualité du travail et par conséquent, un effet négatif sur les résultats attendus de l'organisation.

#### **3.2 Le système fonctionnel**

L'ensemble des activités de la faculté est géré par des traitements manuels ou bien par le biais des applications diversifiées tout en utilisant des supports papiers.

Les besoins de la faculté s'articulent sur les modules fonctionnels suivants : la gestion comptable, la gestion de la ressource humaine. A cela, s'ajoutera le module de gestion des stages.

### 3.3 Diagramme de contexte

Le domaine étudié :

ADAPTATION ET INTEGRATION D'OPENERP POUR LA GESTION DES STAGES  
DES ENSEIGNANTS UNIVERSITERE

Les activités internes du domaine

- Gestion de ressource humaine
- Gestion de comptabilité
- Gestion des stages

#### 3.3.1 Les acteurs du système

Sur le diagramme de contexte, on constate que ce système est en interaction avec des divers acteurs tel que:

Acteur	Description
Le responsable RH	Peut gérer les employés, les congés, les stages, comme ajouté, modifié et supprimé.
Administrateur	Superviseur et administrateur système.

Tableau 3: Liste des fonctionnalités requises 2

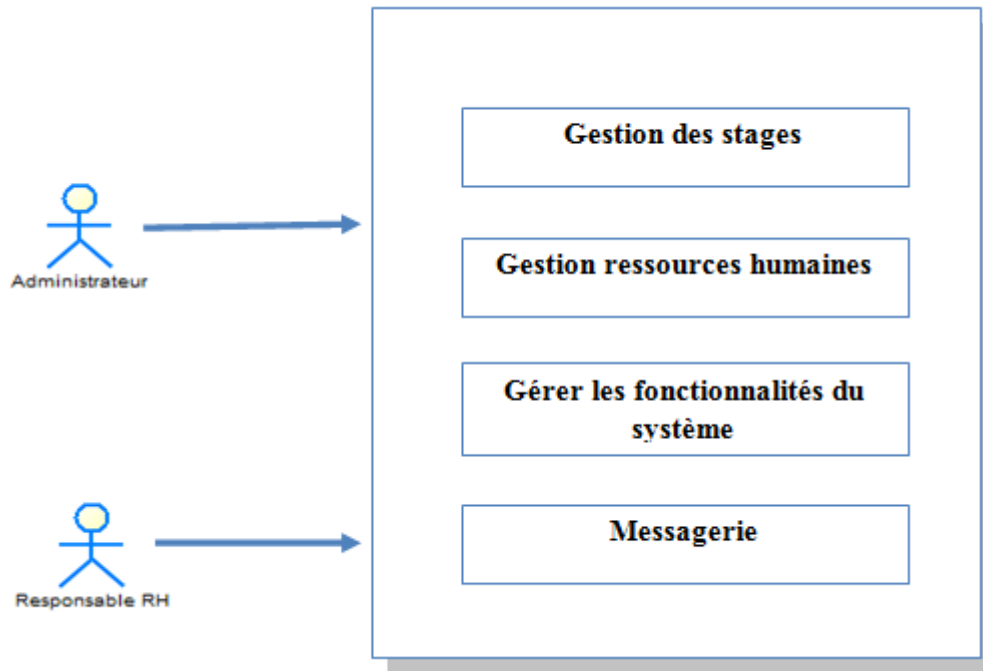


Diagramme 1 : Diagramme de contexte.

### 3.3.2 Analyse des cas d'utilisations

#### ❖ Description des scénarios

##### Gestion des ressources humaines

Ce module permet de gérer l'ensemble des données nécessaires concernant les employés

**Ajouter un employé** (enseignant) : Ce cas permet d'ajouter un employé on saisissant la description relative (Nom, prénom, adresse, grade ....).

**Modifier un employé** : Ce cas permet de modifier les données relatives à l'employé.

**Supprimer un employé** : Ce cas effectue la suppression d'un employé dans le cas où il est démissionné.

##### Gestion des droits d'accès :

#### 🚦 Création d'utilisateurs

Cette fonction permet de créer les différentes fiches du personnel à utiliser le système.

Il suffit de saisir dans le module les données correspondant à l'identification de l'utilisateur concerné Nom, Login, Mot de passe...

**Modification** : Le système permet de modifier toutes les données relatives à l'utilisateur.

**Suppression** : le système permet également de supprimer un utilisateur dans le cas où il n'est plus habilité à utiliser l'application.

### **Création de groupes d'utilisateurs :**

Le système permet de créer un groupe selon la fonction de l'utilisateur et de spécifier les droits d'accès pour chaque groupe.

**Modification** : cette fonction permet de modifier les droits d'accès d'un groupe.

**Suppression** : Le système permettra de supprimer un groupe d'utilisateur préalablement créé.

### **Gestion des stages**

**Ajouter** : L'application permet d'ajouter un stage à un employé en saisissant la description relative (N° stage, type stage, responsable, nom et prénom, la date début et la date fin, manifestation scientifique, type travail à présenté, itinéraire)

**Modification** : L'application permet de modifier les données relatives d'un stage enregistré.

**Suppression** : L'application permet de supprimer un stage déjà enregistré et non attribuer.

### **Serveur d'email**

Le système doit configurer un serveur mail qui assure le flux de communication entre les entités de la faculté.

## **4. Conception du projet**

Dans la démarche de Processus Unifié, la phase de conception suit immédiatement la phase d'Analyse, par ailleurs la conception de logiciel est un art qui nécessite de l'expérience, et elle consiste à traduire les besoins en spécifiant comment l'application pourra les satisfaire avant de procéder à sa réalisation.

## 4.1 Cas d'utilisations

### 4.1.1 Cas d'utilisation de l'administration du système

L'administration complète du système est assurée par l'administrateur. Celui-ci se charge de contrôler l'accès des utilisateurs et de gérer l'ensemble des fonctionnalités du système. Ceci est modélisé par le diagramme de cas d'utilisation suivant :

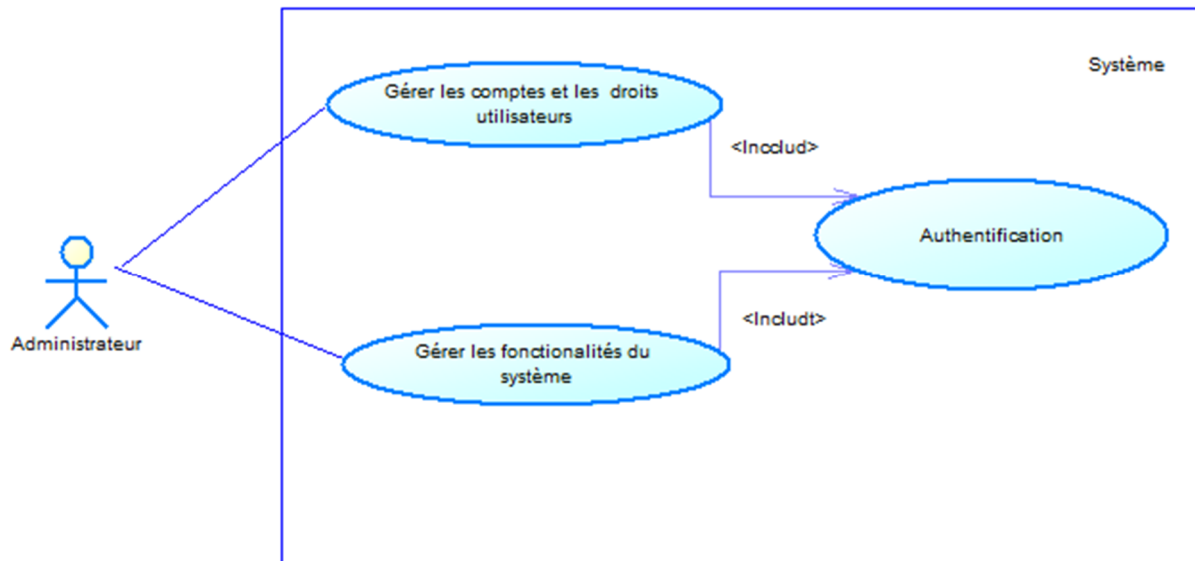


Diagramme 2: Diagramme des cas d'utilisation de l'acteur « administrateur système »

**Authentification** : avant de se connecter au système, chaque acteur doit être identifié par un login et un mot de passe afin d'avoir les permissions d'accès au système.

**Gérer les comptes et les droits des utilisateurs** : ce cas permet à chaque utilisateur du système d'avoir son compte qui comporte l'ensemble des processus métiers dont lesquels il intervient. Comme, il attribue les droits d'accès à l'utilisateur afin de garantir la fiabilité du système.

**Gérer les fonctionnalités du système** : ce cas permet à l'administrateur de gérer la totalité des fonctions métiers du système, comme il peut paramétrer des nouvelles fonctionnalités en cas de besoin.

### 4.1.2 Cas d'utilisation de l'acteur « Responsable RH »

La gestion de la ressource humaine consiste à gérer les employés, les congés, comme elle assure la gestion des stages.

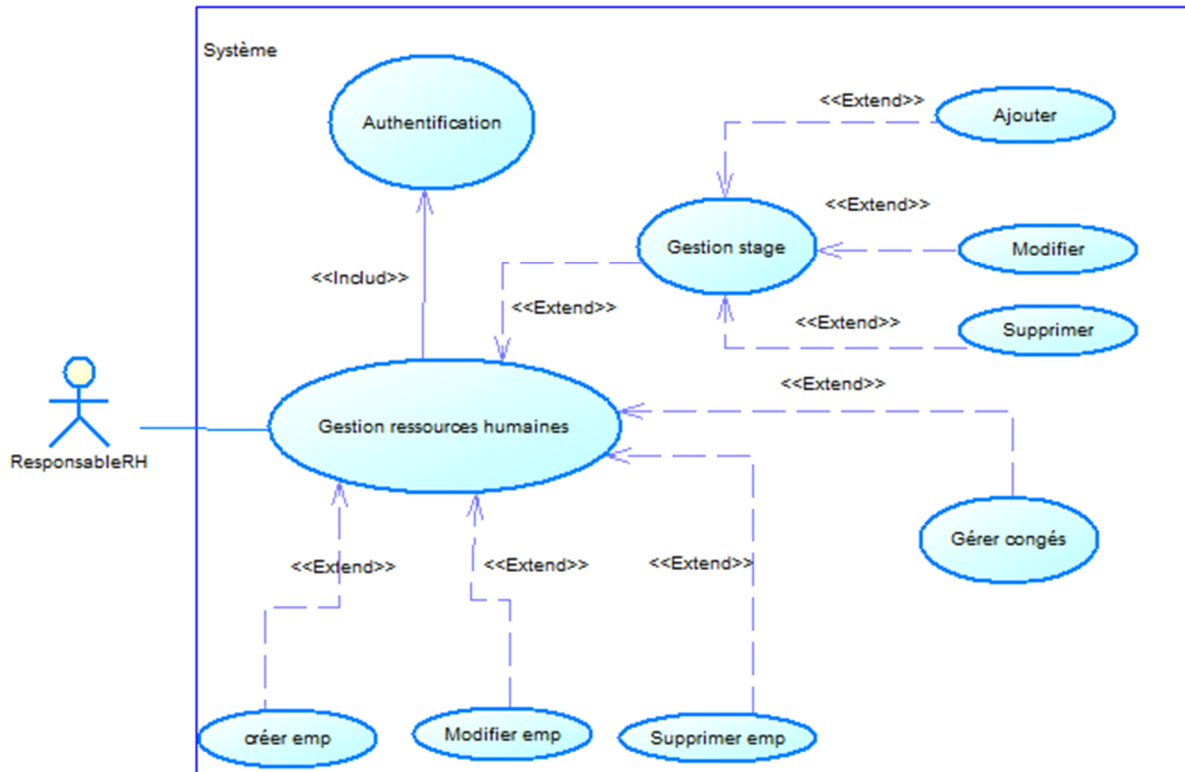


Diagramme 3 : Diagramme des cas d'utilisation de l'acteur « Responsable RH »

Pour cela nous obtiendrons un diagramme des cas de utilisation détaillé pour les deux acteurs comme suit :



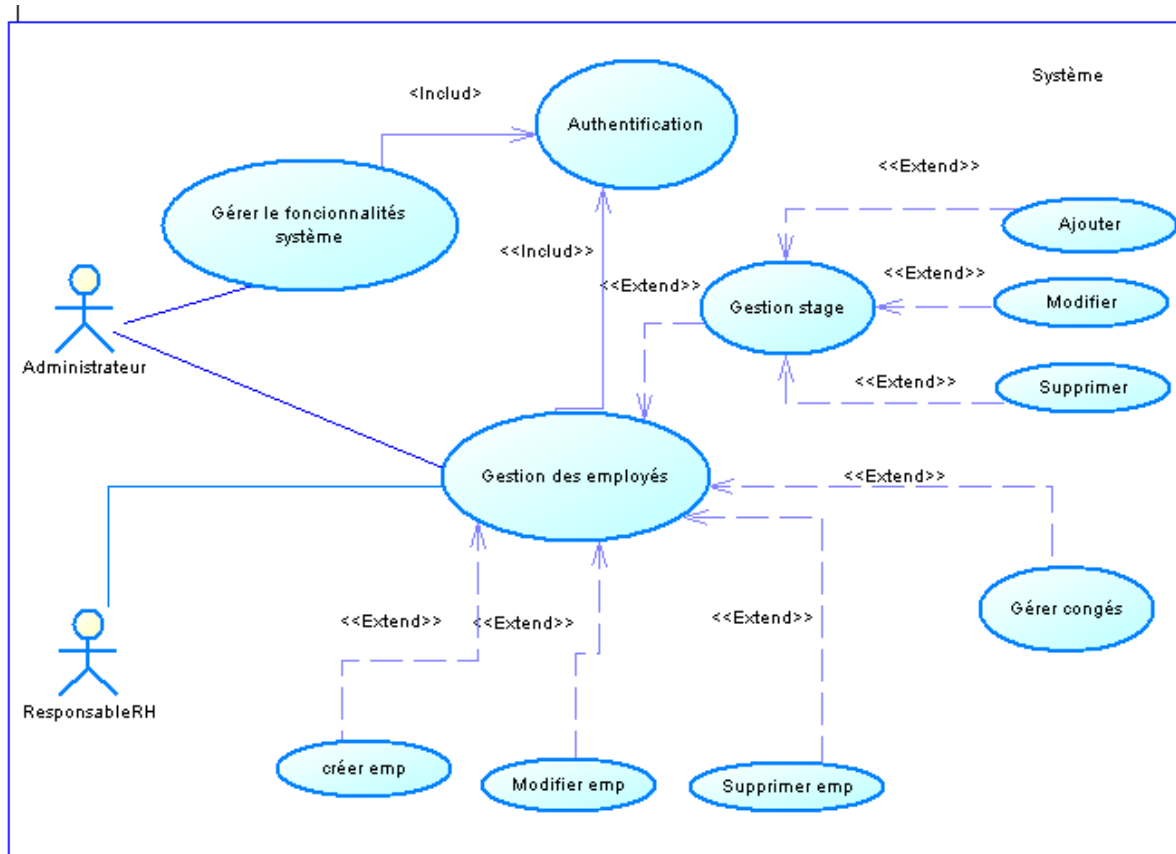


Diagramme 4 : diagramme des cas d'utilisation détaillée

## 4.2 Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.

### 4.2.1 Diagramme de séquence d'authentification

Lorsque un utilisateur souhaite accéder à sa session, une page d'accueil lui sera affichée, dans laquelle saisit ses propres coordonnées d'authentification (login et mot de passe). par la suite le système procède à la vérification des informations introduites et l'envoyé a son tour les cordonnées(login et mot de passe) a la base de donnée, si le login et le mot de passe sont valides sa session lui sera alors ouverte, sinon un message d'erreur est affiché le sollicitant de, réintroduire ses coordonnées. Ce processus de vérification ce répétera autant de fois que l'utilisateur communique des informations erronées.

Les différent acteurs du système ils seront regroupés sous l'acteur unique « utilisateurs »

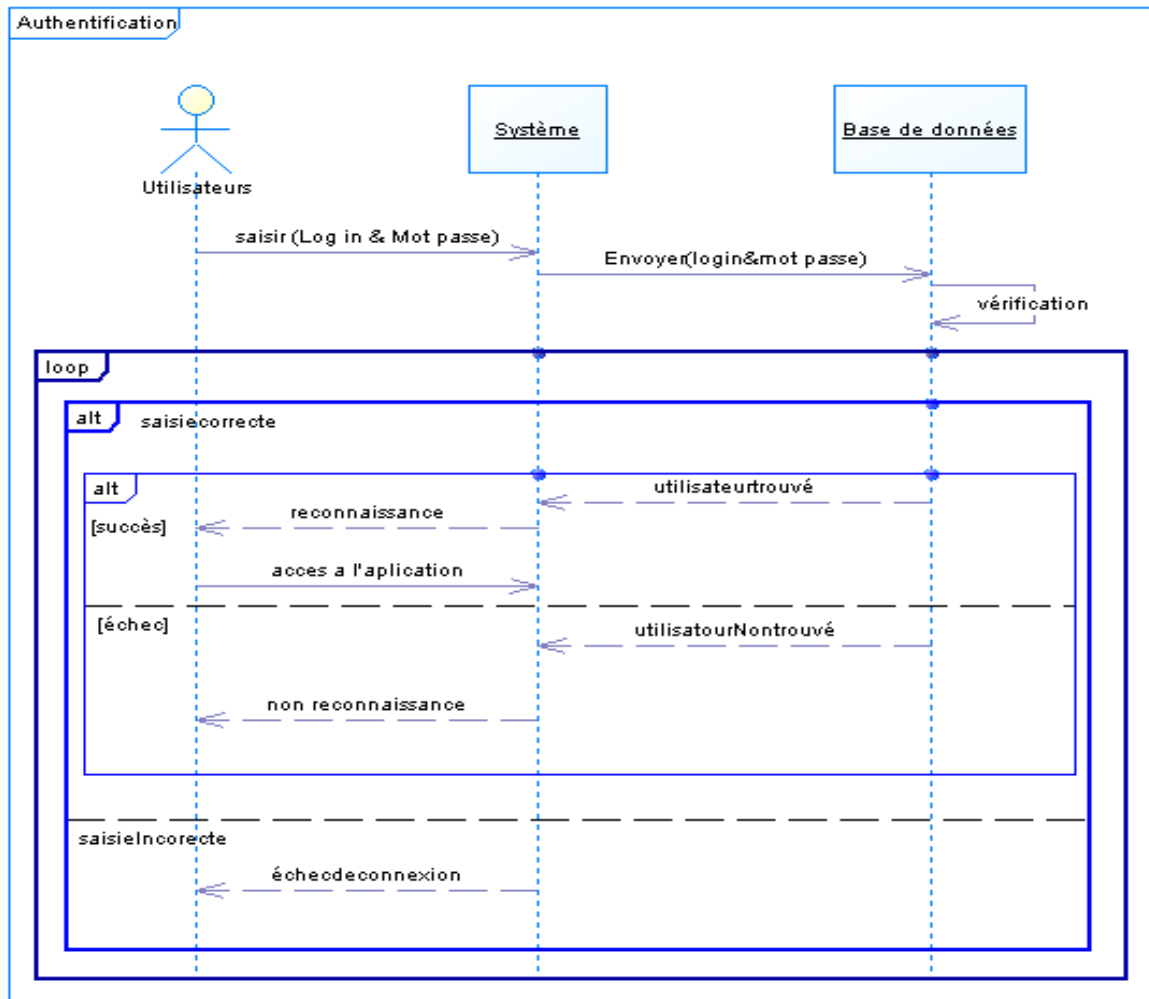


Diagramme 5 : Diagramme de séquence de cas d'utilisation « Authentification »

4.2.2 Diagramme de séquence de cas d'utilisation « ajouter stage »

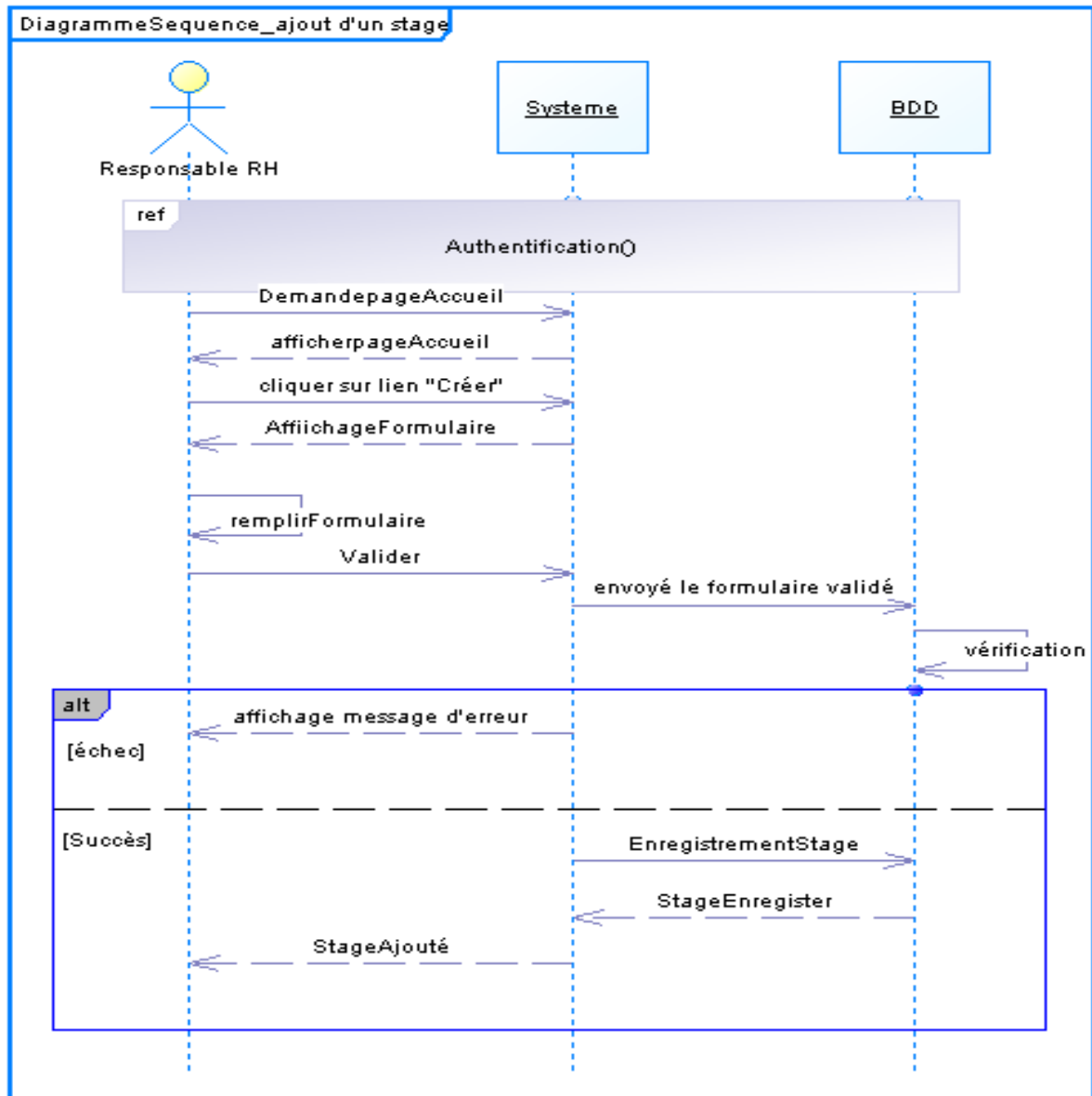


Diagramme 6 : Diagramme de séquence de cas d'utilisation « Ajouter stage »

4.2.3 Diagramme de séquence de cas d'utilisation « modifier stage »

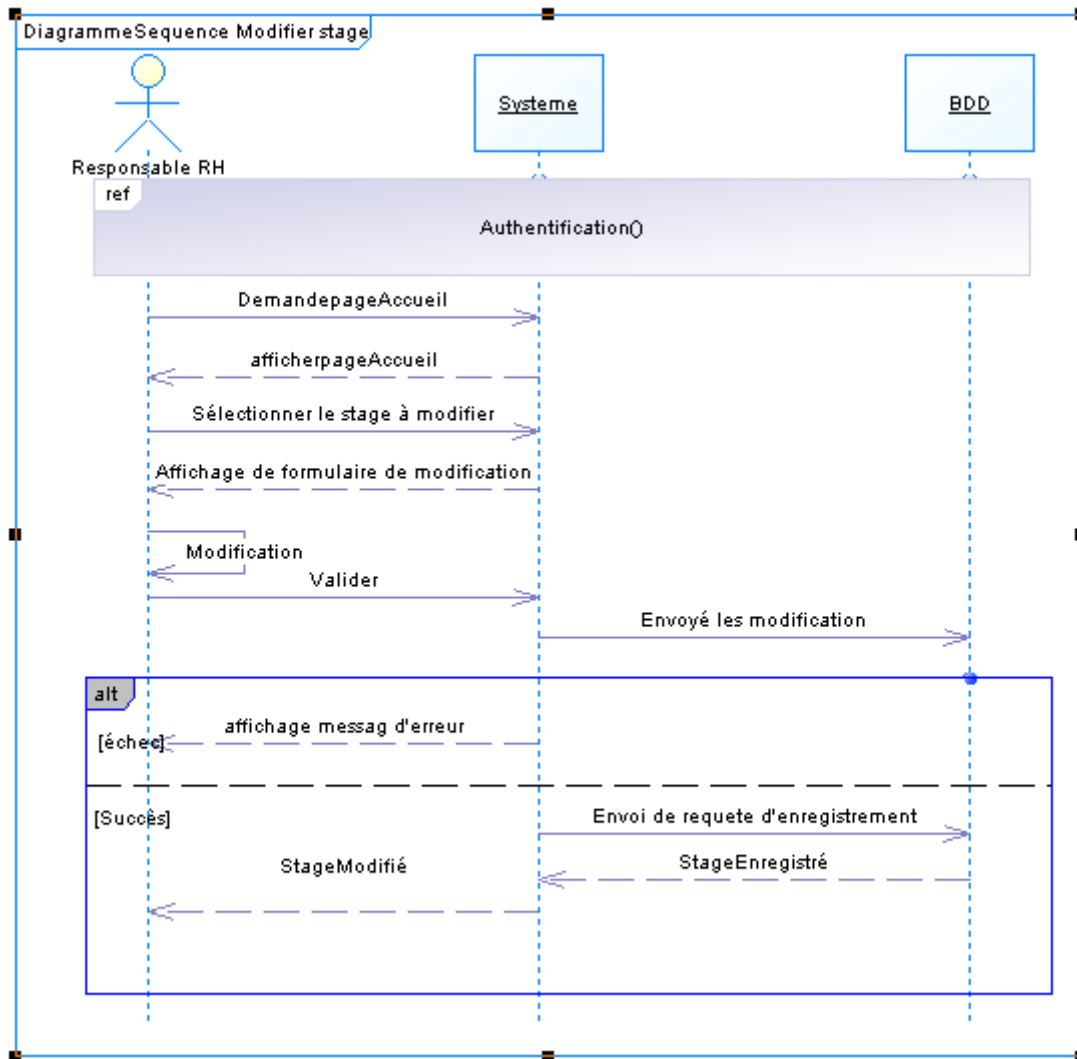


Diagramme 7 : Diagramme de séquence de cas d'utilisation « modifier stage »

4.2.4 Diagramme de séquence de cas d'utilisation « supprimer stage »

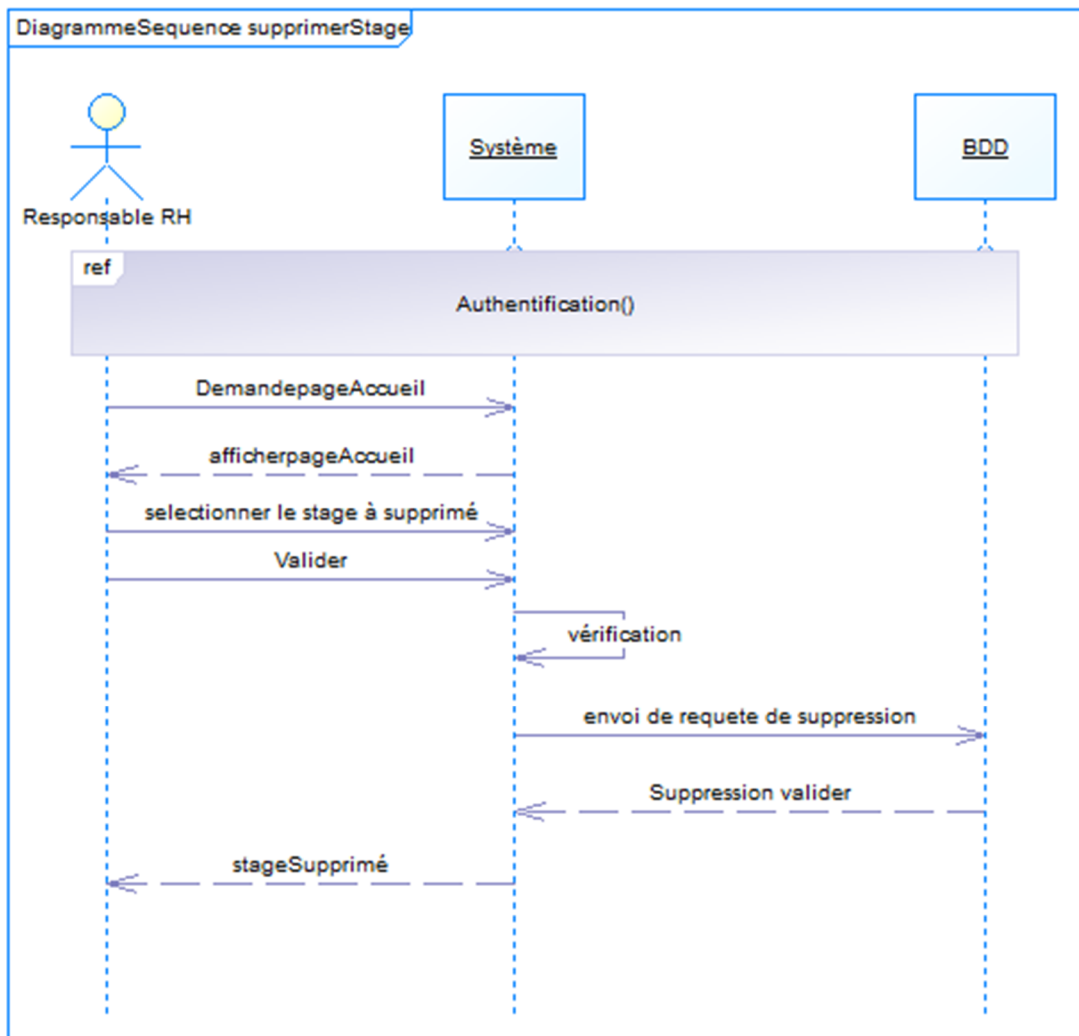


Diagramme 8 : Diagramme de séquence de cas d'utilisation « supprimer stage »

### 4.3 Diagramme de classe :

Le diagramme suivant est le diagramme de classes « fonctionnel » que nous avons réalisé avant de nous lançons dans le développement.

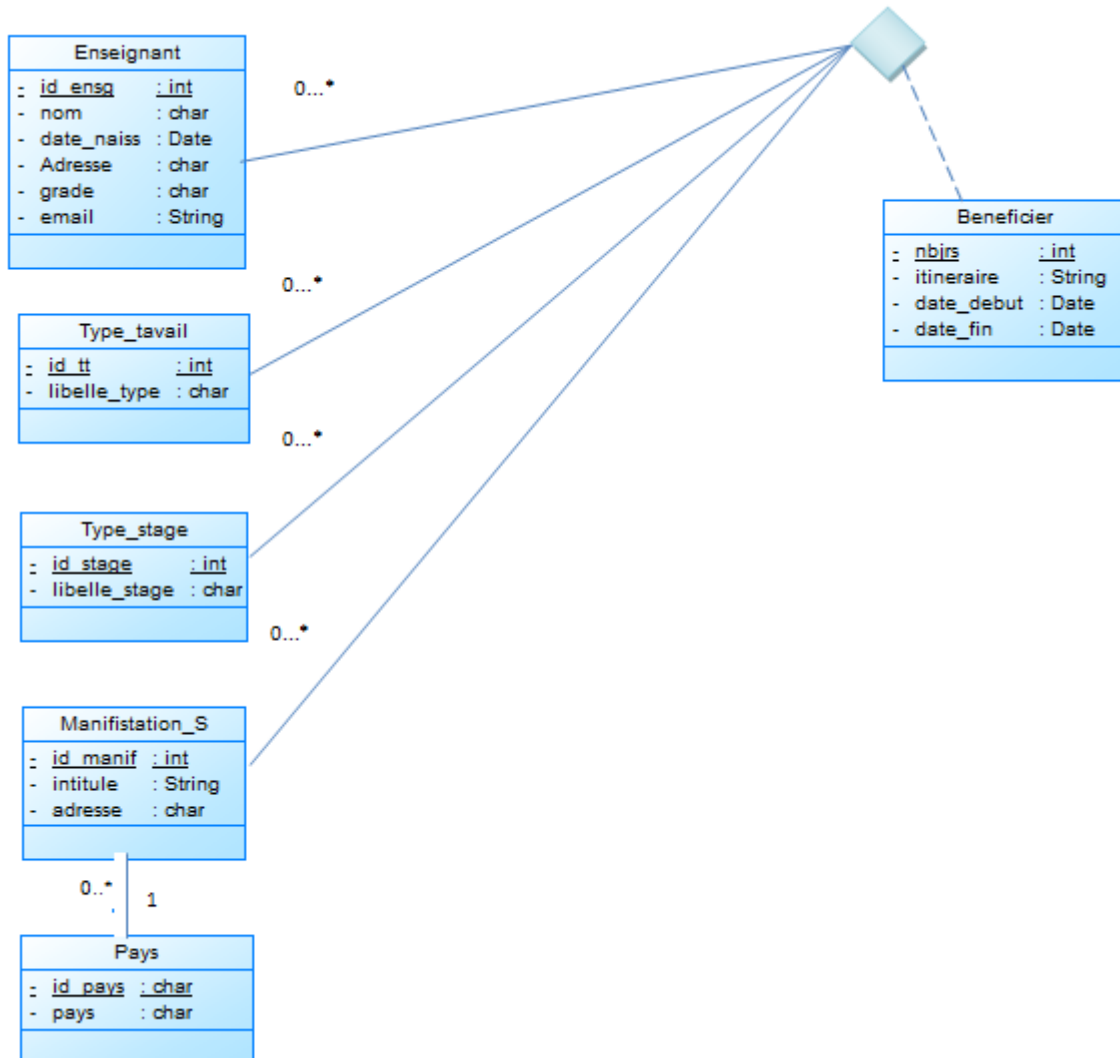


Diagramme 9: Diagramme des classes

Ce diagramme représente les classes nécessaires pour assurer un bon fonctionnement du système à mettre en œuvre.

#### 4.4 Description des classes

Une classe représente la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui le composent. Elle est constituée d'attributs dont les valeurs représentent l'état de l'objet et des méthodes qui sont les opérations applicables aux objets.

⇒ Notre application comporte les classes suivantes :

**Classe enseignant** : classe qui est hérite de la classe hr.employé et qui contient les informations de l'enseignant.

**Classe type de travail à présenter** : contient les différents types de travail à présenter (résumé, article, poster).

**Classe type de stage** : contient les différents types d'un stage (stage de courte durée, congé scientifique).

**Classe manifestation scientifique** : contient les informations concernant la manifestation.

**Classe pays** : contient les différents pays.

**Classe bénéficié** : indique pour chaque stage la durée et itinéraire.

#### 4.5 Dictionnaire des données

Nom de la classe	codification	Désignation	Type	Longueur	Observation
Enseignant	Id_ensg	Identifiant	N	4	
	nom	Le nom d'enseignant	A	25	
	date_naiss	Date de naissance	Date	8	jj/mm/aaaa
	adresse	Adresse personnelle	A	50	
	nationalite	Nationalité	A	25	
	grade		A	30	
	email	L'email d'enseignant	AN	30	
Type_tarvail	Id_tt	identifiant	AN	4	
	Libelle_travail	Type de travail qui va présenter l'enseignant	A	50	
Type_stage	Id_stage	identifiant	AN	4	

	Libelle_stage	Le type de stage qui va effectuer l'enseignant	A	50	
Manifestation _S	Id_manif	identifiant	AN	4	
	intitule	sujet	A	50	
	adresse	Adresse de la manifestation	A	50	
Pays	Id_pays	identifiant	AN	4	
	pays	pays	A	50	
Beneficier	nbjrs	Nombre de jours	AN	75	
	intineraire	intineraire	A	50	
	Date_debut	Date debut	date	8	Jj/mm/aaaa
	Date_fin	Date fin	date	8	Jj/mm/aaaa

Tableau 4 : Dictionnaire de données

#### 4.6 Le modèle logique de données

Après l'application des règles de transformations et de passage du diagramme de classe vers le modèle logique de données, nous avons dégagé les différentes tables relatives au diagramme de classe, elles sont comme suit :

**Enseignant** ( id\_ensg, nom, adresse, date\_naiss, nationalite, grade, email )

**Type de travail a présenter** ( id\_tt, libelle\_type )

**Type stage** ( id\_stage, libelle\_stage )

**Manifestation scientifique** ( id\_manif, intitule, adresse, #id\_pays )

**Pays** ( id\_pays, pays )

**Beneficier** ( nbjrs, inteneraire, date\_debut, date\_fin ,#id\_ensg,# id\_manif, #id\_satge,# id\_tt)

#### 5. Conclusion

Après la description et l'analyse des besoins de la faculté, nous avons utilisé la solution adéquate pour assurer la bonne gestion des processus métiers interne de la faculté. Pour ce faire, nous avons utilisé un ensemble des outils et technologies pour assurer la mise en place



de la solution. Dans le prochain chapitre on aborde la partie réalisation ainsi que la description de quelques choix techniques effectués pendant le développement.

### 1. Introduction

A ce stade du processus, les cas d'utilisation sont terminés, le problème a été analysé en profondeur ; nous avons défini une conception mieux appropriée aux besoins de l'application. Nous pouvons alors entreprendre la dernière activité du Processus Unifié qu'est de même composé de deux parties (implémentation et test), ayant comme objectif d'aboutir à un produit final, exploitable par les utilisateurs. Dans cette phase nous allons présenter l'environnement de développement que nous avons utilisé, l'architecture matérielle mise en place, implémenter tout les cas d'utilisation, et enfin les tester.

### 2. Les technologies utilisées

#### 2.1 Ubuntu

Ubuntu est un système d'exploitation libre commandité par la société Canonical et une marque déposée par cette même société.

Fondé sur la distribution Linux Debian et utilisant le bureau Unity, Ubuntu se veut « convivial, intuitif et sûr ». Il est constitué de logiciels libres, est disponible gratuitement y compris pour les entreprises, et bénéficie d'une nouvelle version (appelée « mise à niveau ») tous les six mois.

Avec une utilisation globale estimée à plus de 25 millions d'utilisateurs, il est principalement conçu pour une utilisation sur des ordinateurs personnels (portables et fixes), bien que d'autres versions consacrées aux netbooks et aux serveurs existent aussi. Depuis Ubuntu 11.04, la version Netbook a fusionné avec la version Desktop. Cette dernière étant passée à l'interface Unity, il n'y avait plus de raison de maintenir deux branches distinctes [7].

#### 2.2 Python

Python est un langage de programmation multi-paradigme. Il favorise la programmation impérative structurée, et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Schème, Smalltalk et Tcl. Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix en passant par Linux et Mac OS, avec Java ou encore .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser. Il est également apprécié par les pédagogues qui y trouvent un langage où la syntaxe, clairement

séparée des mécanismes de bas niveau, permet une initiation plus aisée aux concepts de base de la programmation.

Python est un langage :

- conçu pour produire du code de qualité, portable et facile à intégrer : grâce à sa syntaxe claire, cohérente et concise, Python permet aux développeurs de produire du code de qualité, lisible et maintenable.
- Fourni des le départ avec des modules de tests, Python est un langage agile. Le terme agile est originellement issu de la méthodologie de programmation agile, très proche de la programmation itérative. Cette méthodologie, qui réduit les risques liés à la conception de logiciels, introduit entre autres des principes de tests continus du code.
- De haut niveau, oriente objet et totalement libre : même si elle n'est pas imposée, Python permet la programmation orientée objet. Tous les mécanismes objet essentiels sont implémentés et toutes les données manipulées sont des instances de classes, comme pour les langages SmallTalk ou Ruby.
- Dynamique : dans la plupart des implémentations, le code source n'est pas compilé contrairement à des langages comme C ou Pascal, mais exécute à la volée. On parle alors de langage interprète.

Ce mode de fonctionnement rend la programmation beaucoup plus souple puisqu'il est possible de changer un programme en cours d'exécution [13].



Figure 10: logo de python

### 2.3 PostgreSQL

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

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme 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, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.

PostgreSQL peut stocker plus de types de données que les types traditionnels entiers, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type etc.

PostgreSQL est pratiquement conforme (de plus en plus conforme) aux normes ANSI SQL 89, SQL 92 (SQL 2), SQL 99 (SQL 3), SQL:2003 et SQL:2008. Il fonctionne sur diverses plates-formes matérielles et sous différents systèmes d'exploitation.



Figure : logo de postgresQL

PostgreSQL fonctionne sur Solaris, SunOS, Mac OS X, HP-UX, AIX, Linux, IRIX, Digital Unix, BSD, NetBSD, FreeBSD, OpenBSD, SCO unix, NeXTSTEP, UnixWare et toutes sortes d'Unix. Depuis la version 8.0, PostgreSQL fonctionne également nativement sur Windows. Avant la version 8, il fallait un émulateur de type cygwin pour faire fonctionner PostgreSQL sur ce système d'exploitation.

PostgreSQL est largement reconnu pour son comportement stable, proche de Oracle. Mais aussi pour ses possibilités de programmation étendues, directement dans le moteur de la base de données, via PL/pgSQL. Le traitement interne des données peut aussi être couplé à d'autres modules externes compilés dans d'autres langages [8].

### **2.4 XML (eXtensible Markup Language)**

XML est en quelque sorte un langage HTML amélioré permettant de définir de nouvelles balises. Il s'agit effectivement d'un langage permettant de mettre en forme des documents grâce à des balises (Markup).

Contrairement à HTML, qui est considéré comme un langage défini et figé (avec un nombre de balises limité), XML peut être considéré comme un métalangage permettant de définir d'autres langages, c'est-à-dire définir de nouvelles balises permettant de décrire la présentation d'un texte (Qui n'a jamais désiré une balise qui n'existait pas ?). La force de XML réside dans

sa capacité à pouvoir décrire n'importe quel domaine de données grâce à son extensibilité. Il va permettre de structurer, poser le vocabulaire et la syntaxe des données qu'il va contenir.

En réalité les balises XML décrivent le contenu plutôt que la présentation (contrairement à HTML). Ainsi, XML permet de séparer le contenu de la présentation, ce qui permet par exemple d'afficher un même document sur des applications ou des périphériques différents sans pour autant nécessiter de créer autant de versions du document que l'on nécessite de représentations.

XML a été mis au point par le XML Working Group sous l'égide du World Wide Web Consortium (W3C) dès 1996. Depuis le 10 février 1998, les spécifications XML 1.0 ont été reconnues comme recommandations par le W3C, ce qui en fait un langage reconnu. XML est un sous ensemble de SGML (Standard Generalized Markup Language), défini par le standard ISO8879 en 1986, utilisé dans le milieu de la Gestion Electronique Documentaire (GED). XML reprend la majeure partie des fonctionnalités de SGML, il s'agit donc d'une simplification de SGML afin de le rendre utilisable sur le web , XML fait partie du code des modules composants OpenERP, les vues par lesquelles sont représentés les différents objets sont écrites en XML, ainsi nous y trouvons la description détaillée de l'affichage des arbres, formulaires, menus et autres actions [6].

### **2.5 Power AMC**

PowerAMC est un logiciel de modélisation créé par la société SDP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

Créé par SDP sous le nom AMC\*Designor, racheté par Powersoft, ce logiciel est produit par Sybase depuis leur achat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque PowerDesigner

PowerAMC permet de réaliser tous les types de modèles informatiques. Il reste un des seuls qui permet de travailler avec la méthode Merise. Selon Riff News, cela permet d'améliorer la modélisation, les processus, le coût et la production d'applications [9].

### **3. Module ressources humaines sous Openerp**

La Gestion des Ressources Humaines a bien évolué depuis le temps où la fonction était dénommée Administration du Personnel. Il ne s'agit plus d'administrer mais bien de manager le capital humain. La Gestion des RH comporte de multiples aspects, tous aussi importants les uns que les autres.

Nous allons découvrir les différentes et principales fonctionnalités RH du logiciel Openerp V7.0 .la structure du progiciel se veut dynamique et inductive .l'utilisateur n'éprouve généralement pas de grandes difficultés à travers le système. Cependant, il est toujours intéressant d'avoir une vision globale des outils proposé à fin d'optimiser l'utilisation du logiciel. La liste ci-dessous présente les différents menus et sous-menu du module Ressources humaines .en se référant aux modules proposés dans la section RH, les principaux outils correspondants nous y seront détaillés.

#### **3.1 Gestion des ressources humaines**

La gestion de la ressource humaine englobe :

- Gestion des employés
- Gestion de recrutement
- Gestion du temps
- Gestion notes de frais
- Gestion des congés

Qu'on va les détails par la suite.

##### **3.1.1 Gestion des employés :**

Dans ce volet nous pouvons consulter la liste des employés ainsi que les détails de chaque un et autre action en relation.

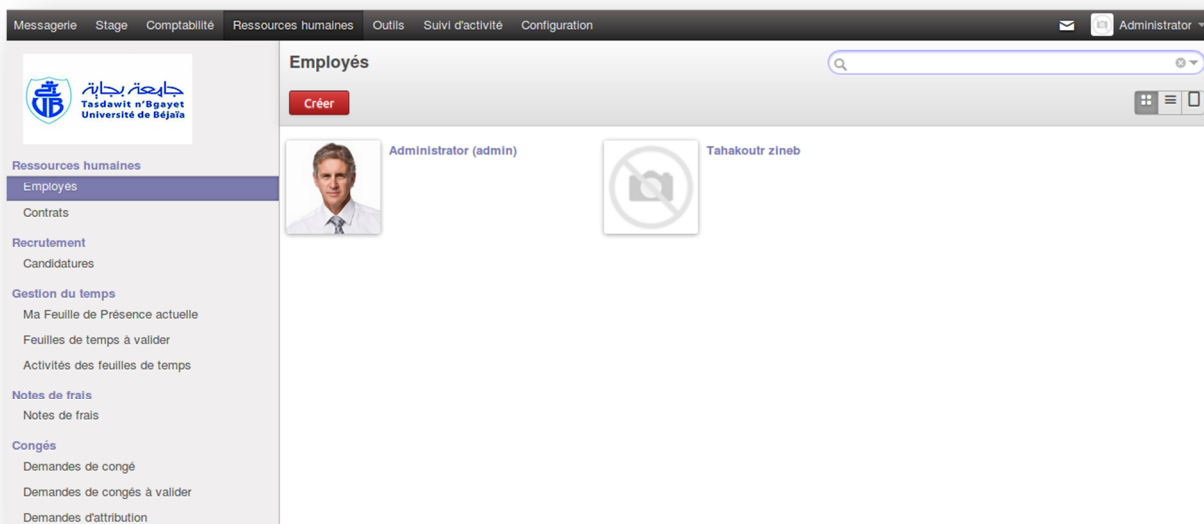


Figure 12 : Fonctionnalités ressources humaines - Liste des employés

### 3.1.2 Création d'un nouvel employé :

Pour créer et compléter un dossier administratif pour un nouvel employé, nous utilisons la section le sous-menu ressources humaines >employé du module ressources humaines

Les principaux renseignements liés au nouvel employé que nous pouvons encoder sont : le nom, les comptes bancaires, les numéros sociaux (sécurité social, passeport), les coordonnées (téléphone, adresse principale, adresse courriel, etc.), les informations relatives à son emploi (Département, poste, etc.), une photo peut être téléchargée.

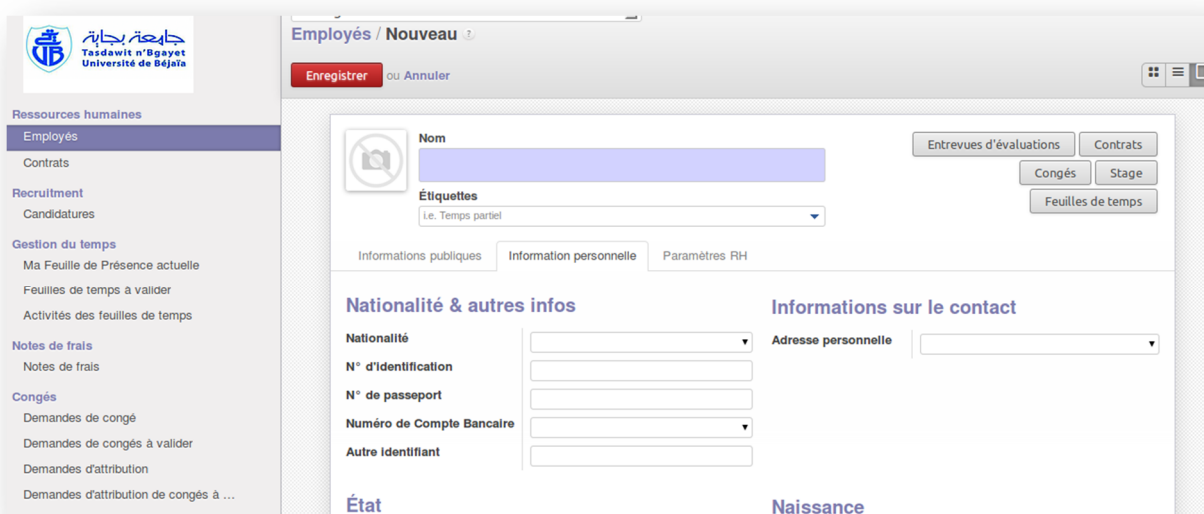


Figure 13: Interface d'ajout d'un employé

Pour le formulaire « Informations publiques » de création d'un nouvel employé nous avons ajouté un champ qui est « **Grade** » permet d'encoder les différents types de grade existant exemple maître des conférences, professeur, maître assistant...etc.

The screenshot shows the 'Ajouter un employé' form. On the left is a navigation menu with categories like 'Ressources humaines', 'Recrutement', 'Gestion du temps', 'Notes de frais', 'Congés', and 'Evaluation'. The main form has tabs for 'Informations publiques', 'Information personnelle', and 'Paramètres RH'. The 'Informations sur le contact' section includes fields for 'Adresse professionnelle', 'Adresse électronique professionnelle', 'Téléphone professionnel', 'Tél. portable professionnel', and 'Localisation du Bureau'. The 'Poste' section includes 'Département', 'Poste', 'Responsable', 'Mentor', and 'Grade' (highlighted with a red box). There is also a checkbox for 'Est un responsable' and a 'Utilisateur lié' dropdown.

Figure 14 : Interface ajouter un employé - Grade

### 3.1.3 Contrats :

Nous devons définir un contrat pour chaque employé. Ce contrat sera utilisé lors de la génération des bulletins de salaire. Pour créer un contrat, il faut aller à : **Ressources humaines > Ressources humaines > Contrats**.

The screenshot shows the 'Ajouter un contrat' form. The top navigation bar includes 'Messagerie', 'stage', 'Ventes', 'Comptabilité', 'Entrepôt', 'Ressources humaines', 'Outils', 'Suivi d'activité', and 'Configuration'. The left menu is under 'Ressources humaines' with 'Contrats' selected. The form has a header with 'Contrats / Nouveau' and buttons for 'Enregistrer' and 'Annuler'. The main form has a 'Référence du contrat' field and a 'Type de Contrat' dropdown set to 'Employee'. Below are sections for 'Salaire et avantages' (with 'Salaire' at 0,00 and 'Structure salariale' dropdown) and 'Durée' (with 'Durée de la période d'essai' and 'Durée' date pickers, and 'Heures de travail' and 'Paie planifiée' dropdowns).

Figure 15 : Interface d'ajouter un contrat



## 3.2 Gestion de recrutement

### 3.2.1 Candidature

Au travers de cette section, nous avons l'opportunité de prédéfinir l'ensemble des étapes de notre processus de recrutement. Chaque démarche de notre dotation peut être directement affectée à un service. **Ressources humaines > ressources humaines > candidature > recrutement**, encodons les étapes de notre planification en RH. Dans la même section, nous pouvons également enregistrer les types des diplômes auxquels nous avons les plus confronté ainsi que les canaux de communication par lesquelles les candidats auront à notre annonce.

The screenshot displays a web interface for managing recruitment. On the left is a navigation menu with categories like 'Ressources humaines', 'Recrutement', 'Gestion du temps', 'Congés', and 'Evaluation'. The main area shows a form titled 'Embaucher et créer l'employé' with a progress bar indicating steps: 'Qualification initiale', 'Premier entretien', 'Second entretien', 'Contrat proposé', and 'Contrat signé'. The form fields are organized into sections: 'Sujet' (Subject), 'Nom du candidat' (Candidate Name), 'Contact' (Contact), 'Courriel' (Email), 'Téléphone' (Phone), 'Portable' (Mobile), 'Niveau d'étude' (Education Level), 'Poste' (Position), 'Job concerné' (Job), 'Département' (Department), 'Disponibilité' (Availability), 'Responsable' (Responsible), 'Action suivante' (Next Action), 'Appréciation' (Appreciation), 'Origine' (Origin), 'Referred By', 'Contract', 'Salaire demandé' (Requested Salary), and 'Salaire proposé' (Proposed Salary). A 'Planifier un rendez-vous' button is located at the top right of the form.

Figure 16 : Interface d'ajouter un recrutement

## 3.3 Gestion du temps

### 3.3.1 feuilles de temps / feuille de présence

Le temps est la clé de l'achèvement des projets, et sa bonne gestion est un point essentiel pendant l'exécution d'un projet jusqu'à son déploiement. Une feuille de temps est un moyen pour déterminer et enregistrer le temps effectué par un employé sur chaque tâche.

Le module de feuille de temps, **hr\_timesheet**, sous OpenERP nous permet saisir et de vérifier le temps consacré par les employés sur un projet ou une activité particulière.

Le module de feuille de temps est entièrement personnalisé. Il nous donne plusieurs aperçus (Par jour, Par semaine...etc). Pour accéder aux feuilles de présence, il faut suivre le chemin

**Ressources humaine > Gestion du temps > Ma feuille de présence actuelle**

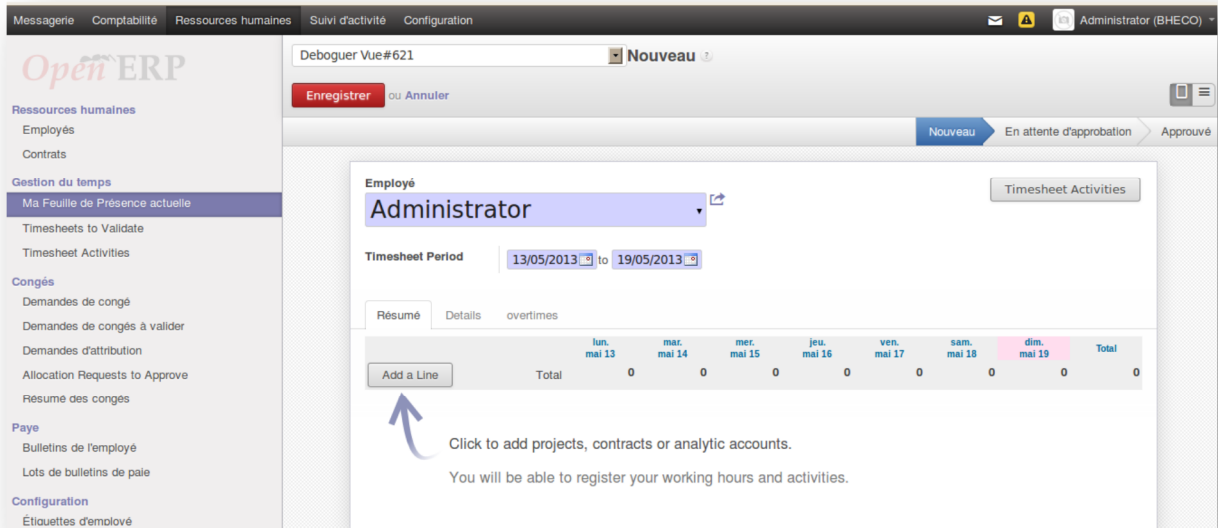


Figure 17 : Interface pour créer une feuille de temps

### 3.4 Gestion notes de frais

Ce formulaire permet aux employés de déposer leurs notes de frais afin qu'elles soient remboursées après approbation.

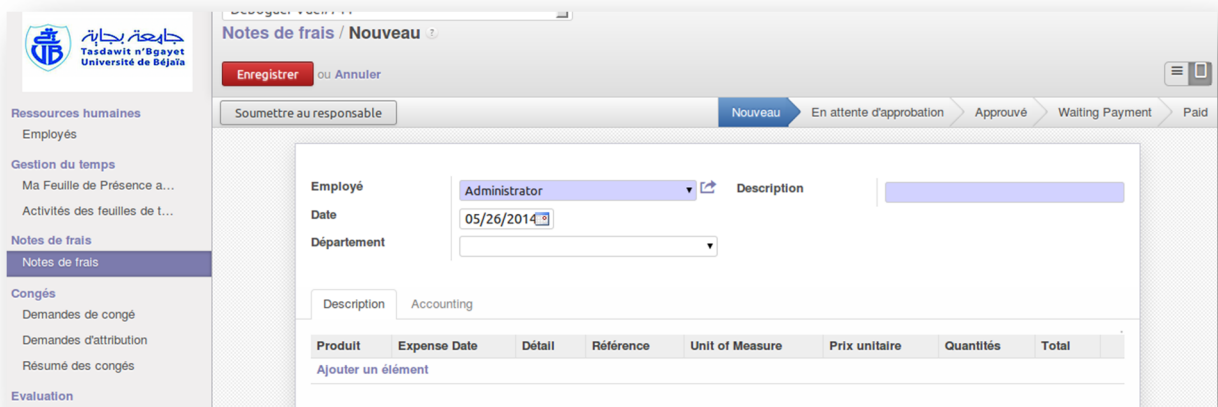


Figure 18 : Interface pour créer notes de frais

### 3.5 Gestion des congés

Nous pouvons gérer les congés pris par les employés en utilisant le module d'OpenERP `hr_holidays`. A l'aide du menu **Ressources humaines > Congés > Demandes de congé** l'employé peut demander un congé.

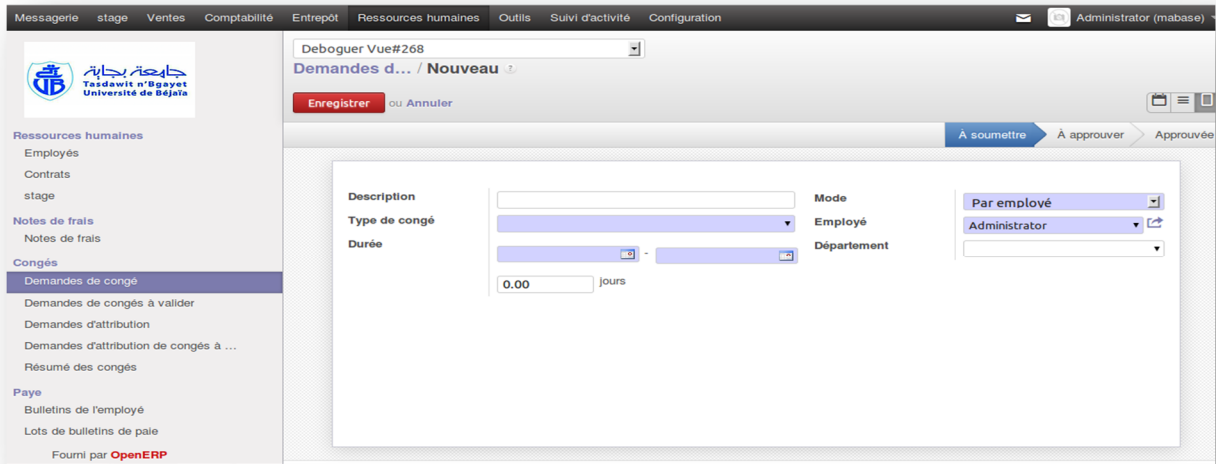


Figure 19 : Interface pour demander un congé

Les demandes de congés peuvent être enregistrées et validées par le responsable. Une fois la demande de congé est validée, elle apparaît automatiquement dans l’agenda de l’employé. D’un autre côté, les responsables peuvent déterminer, au préalable, les différents types de congés (congés payés, maladie...etc) et gérer les allocations par type.

Un employé peut demander aux responsables de lui attribuer plus de jours de congés en faisant une nouvelle demande d’attribution sur **Ressources humaines > Congés > Demandes d’attribution**

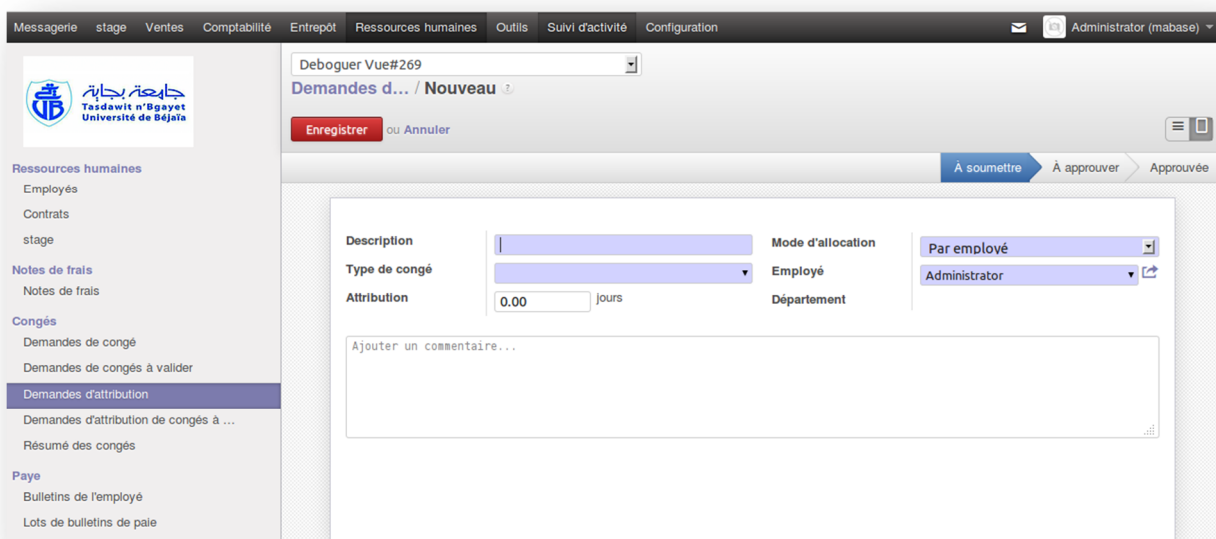


Figure 20 : Interface pour demander l’attribution d’un congé

Nous pouvons configurer, dès le départ, les types de congés partant à **Ressources humaines** > **type de congé**.

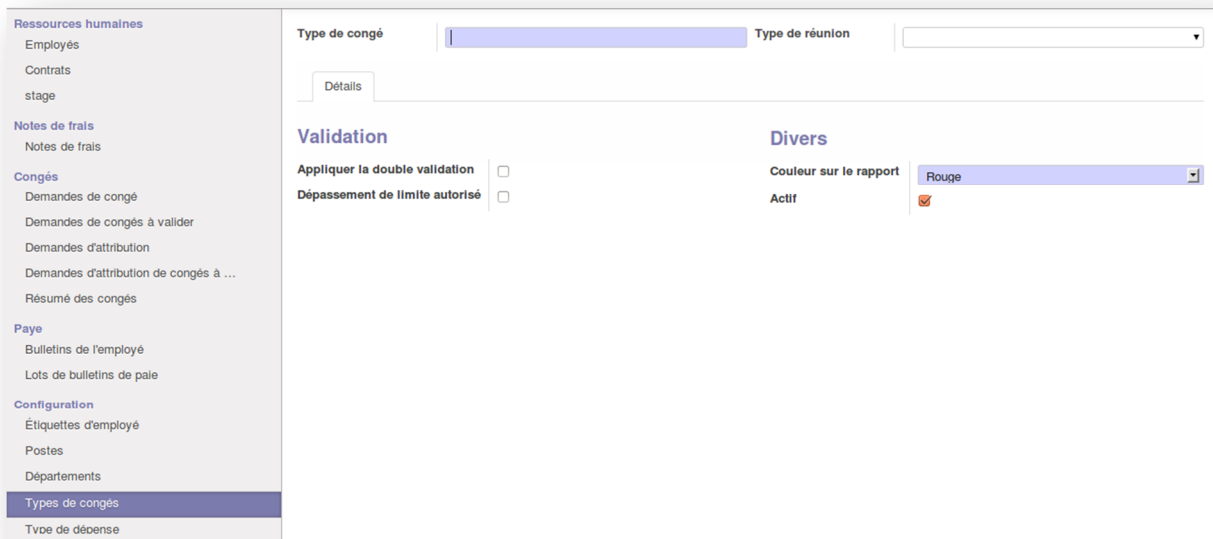


Figure 21 : Interface pour créer les types de congés

### 3.6 Gestion configuration

Ce n'est pas dans le but de nous déstabiliser d'emblée que ce travail commence par le menu configuration qui est pourtant le dernier de la liste proposée par le menu à la gauche de notre écran. Cependant, il est nécessaire d'effectuer une série de paramétrages avant de se lancer dans la navigation des fonctionnalités RH proposés par le logiciel. Le menu configuration de module ressources humaines permet de structurer l'ensemble des caractéristiques intrinsèques de l'entreprise. En effet, nous pouvons prédéfinir la structure hiérarchique de la société, les différents postes recensés, les processus d'évaluation soumis aux salariés, les différents types de congés, poste, département, etc.

#### 3.6.1 Poste

Grâce au menu **configuration** > **poste**, nous pouvons avoir un aperçu des postes recensés au sein de la organisation. Les postes peuvent être classés par leur dénomination, le département auquel ils sont affectés, le nombre d'employés actuel, attendu en recrutement, .....etc.

Figure 22 : Interface pour créer les postes

### 3.6.2 Département

Ce menu permet d'enregistrer les noms des départements dans le but de structurer l'organisation interne de l'entreprise.

Figure 23 : Interface pour créer des départements

### 3.6.3 Contrat :

Ce menu nous permet d'encoder les différents types de contrats en vigueur au sein d'une entreprise (par exemple, contrat à durée indéterminée, contrat à durée déterminée, consultant, etc.).

### 3.6.4 Recrutement :

Au travers de cette section, nous avons l'opportunité de prédéfinir l'ensemble des étapes de processus recrutement. Chaque démarche de dotation peut être directement affectée à un service. Via le sous-menu recrutement, nous encodons les étapes de planification en RH. Dans la même section, nous pouvons également enregistrer les types de diplômes auxquels nous

sommes le plus confrontés ainsi que les canaux de communication par les lesquelles les candidats auront répondu aux annonces.

### 4. Module « Stage »

Après avoir analysé les différents composants du module RH, nous avons constaté qu'il faut créer les objets suivants :

**Un menu** : stage, manifestation scientifique, itinéraire, configuration qui contient type stage et type de travail.

Nous avons ajouté un champ « Grade » au formulaire concernant l'employé dont on a besoin.

Ces nouveaux objets sont regroupés dans un nouveau module à ajouter, qui n'est autre que « Stage ». Le module est un dossier composé de fichiers python (.py) contenant les définitions des classes et d'autres XML contenant les détails des vues de ces dernières, en plus de certains dossiers comme les wizards (assistants se lançant dans certaines conditions) ou report (contient les modèles de rapport du module).

Après la création des différentes classes nouvelles et classes filles, nous devons créer deux fichiers spéciaux pour OpenERP afin de pouvoir installer le module. Le premier fichier est « `__init__.py` » où on importe tous les fichiers python qu'il nous faut pour faire fonctionner le module. Le deuxième est « `__openerp__.py` » (anciennement appelé `__terp__.py`), dans ce fichier se trouve la fiche technique du module, à savoir son nom, sa description, le nom de l'auteur, les noms des modules auxquels il dépend et les fichiers des vues XML à inclure pour pouvoir afficher nos objets créés ou étendus.

#### 4.1 Installation

Maintenant que le codage est terminé, nous pouvons passer à l'installation du module « Stage », qui installera d'abord les modules auxquels il est lié, ensuite ajoutera ses propres fonctionnalités.

Avant de lancer le serveur d'OpenERP, on doit copier le dossier du module « Stage » dans le dossier « Addons » d'OpenERP, ensuite nous lançons le serveur (fichier `openerp-server.py`), et nous pourrions à ce stade, installer notre nouveau module.

Bien évidemment, nous devons d'abord se connecter puis accéder aux paramètres.



Figure 24 : connexion au système

Une fois connecté, nous nous rendons aux paramètres puis dans le menu modules, nous lançons une mise à jour de la liste des modules, afin que nous puissions trouver celui que nous venons d'ajouter parmi la liste, là nous lançons l'installation des modules concernés.

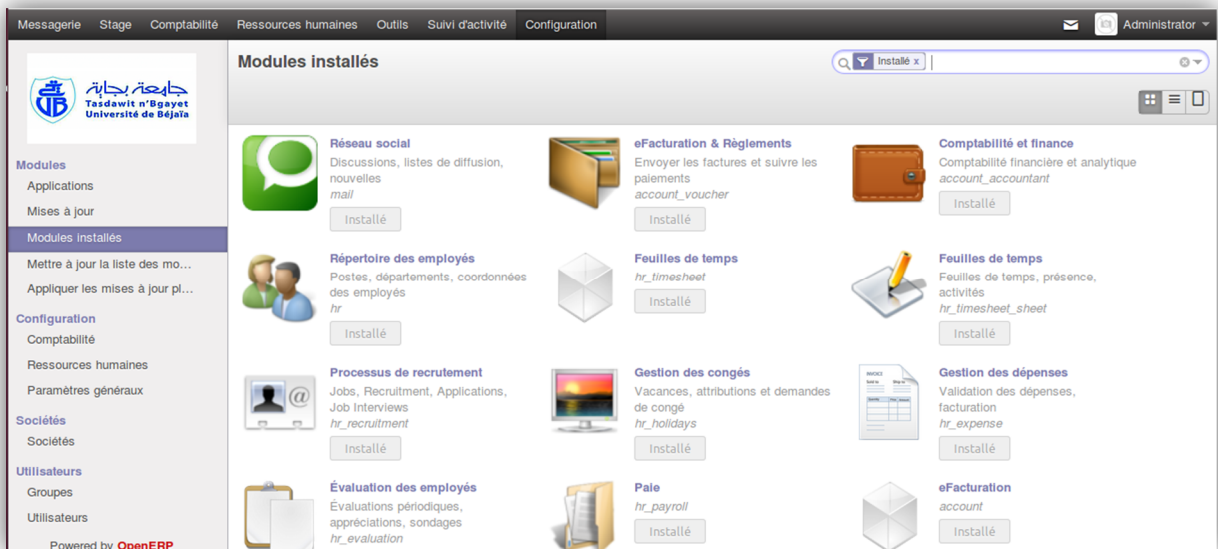


Figure 25 : Accueil OpenERP - client web

## 4.2 Paramétrage

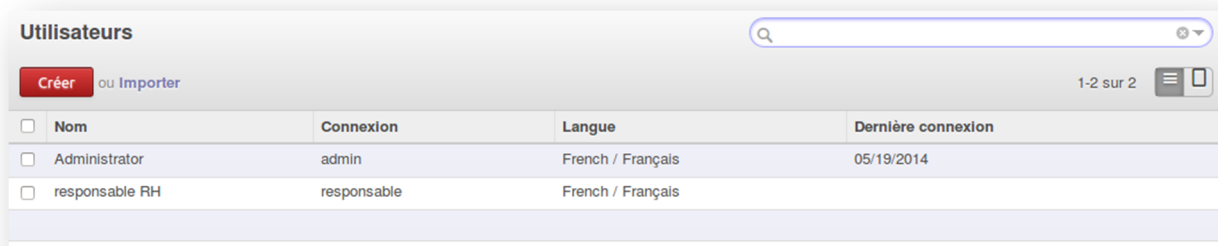
Maintenant que nous avons installé les modules, nous pouvons commencer à paramétrer le PGI et démarrer une simulation pour s'assurer de son bon fonctionnement.

### ✓ *Paramétrage Général*

Avant de se lancer dans le paramétrage de l'application, nous devons commencer par un paramétrage général, où il faut saisir les détails de l'entreprise, son adresse....etc.

## 4.3 Gestion de base

La gestion de base (ajouter, modifier, supprimer) concerne les objets élémentaires, comme les utilisateurs, les employés et groupes. Ci-dessous la liste des utilisateurs qui doivent interagir avec le système :



<input type="checkbox"/>	Nom	Connexion	Langue	Dernière connexion
<input type="checkbox"/>	Administrator	admin	French / Français	05/19/2014
<input type="checkbox"/>	responsable RH	responsable	French / Français	

Figure 26 : Liste des utilisateurs

Pour bien contrôler les droits d'accès et sécuriser notre système, **openerp** doté d'une fonction des groupes qui permet de gérer et d'affecter les utilisateurs a des groupes distincts avec leurs tâches à réaliser.

Plus spécifiquement déterminer les menus accessibles par chaque utilisateur comme il montre la figure suivant.



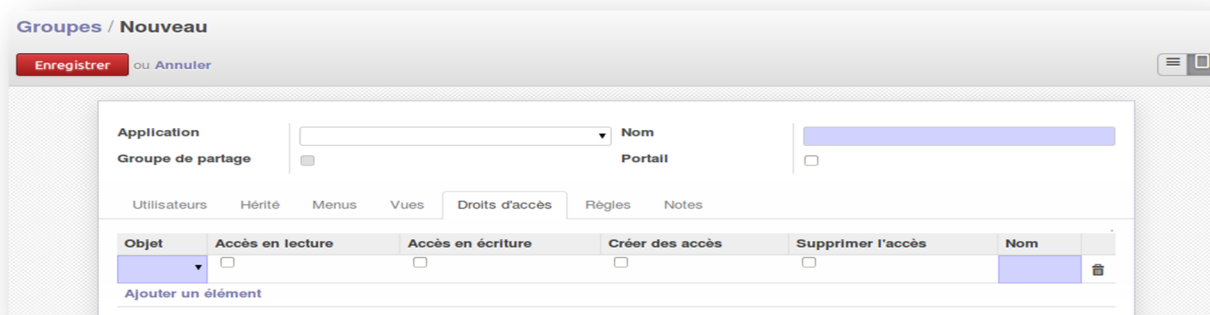


Figure 27 : Formulaire des droit s d'accès

Pour faciliter l'utilisation de l'application, des info-bulles s'affichent lorsque nous survolons les champs, fournissant des explications concernant la fonction du champ pour aider à bien le renseigner.

#### 4.4 Gestion des stages

La gestion des stages consiste à attribuer des stages aux enseignants, par exemple un enseignant besoin d'effectuer un stage à l'étranger il doit le demander au service, une fois la demande de l'enseignant arrivée, le responsable lui accorde le stage, le valide et l'envoi par email la confirmation.

##### 4.4.1 Ajouter un nouveau stage :

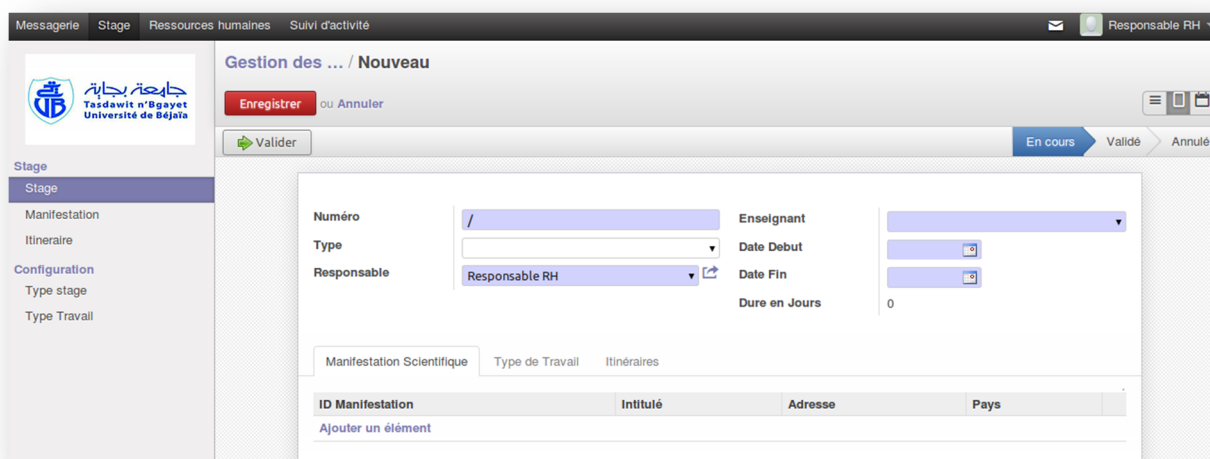


Figure 28: Interface d'ajouter un nouveau stage

Nous avons divisé ce formulaire en quatre parties :

**Partie I**

**Décision de stage N°** : c'est le numéro d'enregistrement de la décision de stage,

**Type stage** : le type de stage qui va effectuer (il ya deux type : congés scientifique, stage courte durée),

**Responsable** : le responsable qui valide le stage (champ obligatoire),

**Pays et ville** : le pays et la ville où il va effectuer son stage ;

**Enseignant** : le nom et prénom de l'enseignant qui est déjà enregistré dans module ressources humaine (champ obligatoire),

**Date début et date fin et Durée en jours** : c'est un calendrier pour définir la durée de stage

(Un message d'erreur s'affiche si la date début supérieure à la date fin).

**Partie II :**

**Manifestation scientifique** : il ya quatre champs à remplir

**Id manifestation** : c'est un numéro automatique,

**Intitulé** : c'est le sujet de stage,

**Pays, Adresse** : l'adresse où il va effectuer le stage,

**Partie III :**

**Type de travail** : c'est le type de travail qui va présenter (il ya trois type : résumé, article, poster).

**Partie IV :**

**Itinéraire** : il ya quatre champs à remplir,

**Id itinéraire** : numéro d'itinéraire,

**Pays** : pays,

**Ville et l'adresse** : la ville où ça se passe le stage et l'adresse exacte.

#### 4.4.2 Menu configuration

Dans ce menu nous pouvons faire la configuration de type de stage et type de travail.

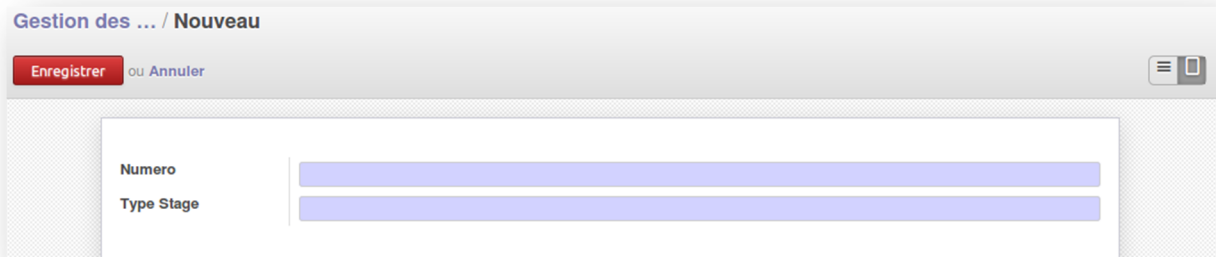


Figure 29 : Interface créer les types de stage

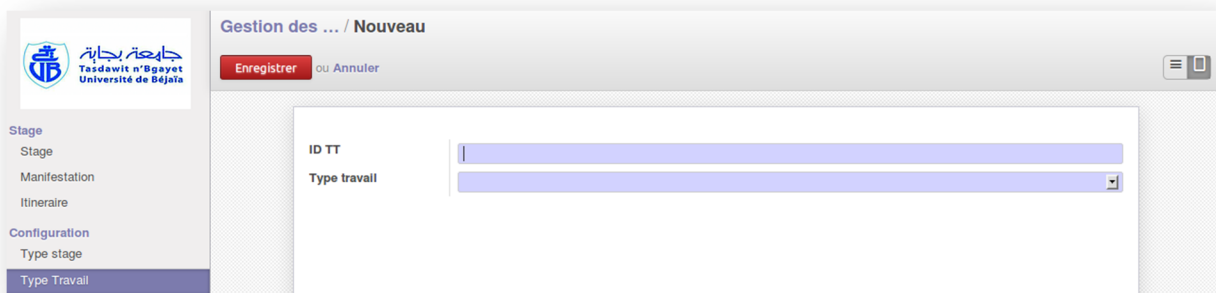


Figure 30 : Interface créer des types de travail

## 5. Services de messagerie

Le service de messagerie constitue l'un des éléments les plus importants dans notre ERP c'est lui qui permet de faire communiquer les documents entre le service et les enseignants ainsi que le partage d'information entre les utilisateurs du système puis l'envoi des emails.

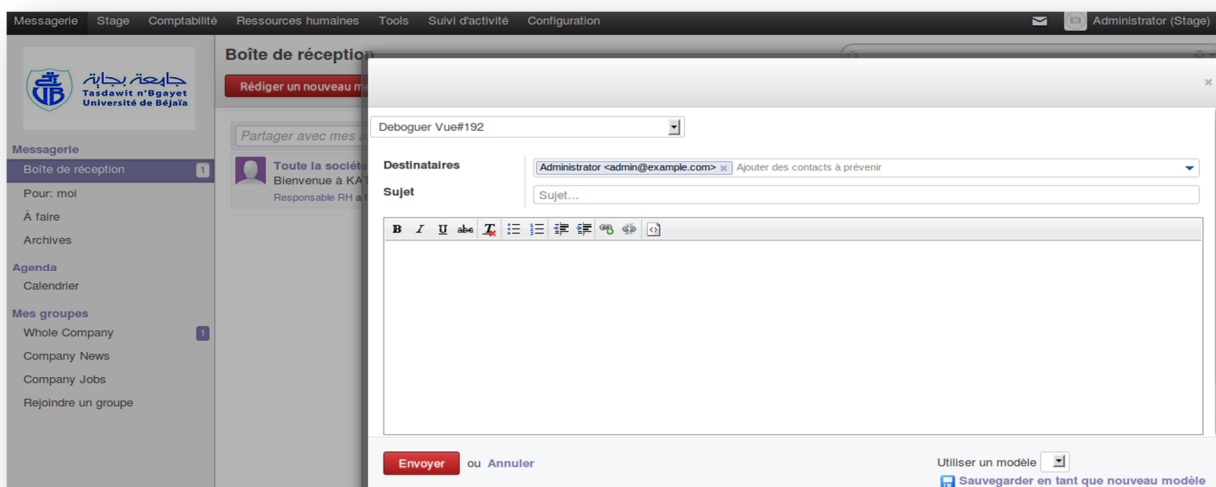


Figure 31 : service de messagerie

### **6. Conclusion**

Dans ce chapitre nous avons décrit les différentes technologies adoptées et utilisées pour la réalisation de ce projet, on a commencé par le système d'exploitation Linux, le système de gestion de bases de données PostgreSQL, les langages nécessaires pour le développement, à savoir le langage Python et XML, puis nous avons parlé sur les principales fonctionnalités de la gestion ressources humaines sous OpenERP, ensuite nous avons montré les étapes de développement de module « stage » et ses différents fonctionnalités, enfin nous avons configuré le service messagerie.

## Conclusion générale et perspectives

Ce stage de fin de cycle a été une occasion pour nous de côtoyer le monde des ERP et de l'entreprise, mais avec plus de responsabilités.

L'objectif de notre travail est de réaliser au profit du service du personnel de la faculté des sciences de la nature et de la vie de l'université de Bejaia. Dans ce cadre, nous avons mis en place un système de gestion intégrée qui est OpenERP et développée une nouvelle fonctionnalité « module stage » au sein de la faculté.

Pour ce faire, nous avons commencé par la description des besoins des futurs utilisateurs du système. Par la suite, nous nous sommes basés sur une étude bibliographique sur les différentes solutions existantes à savoir, les progiciels de gestion intégrée dans le marché, afin de choisir une solution qui respecte les exigences de la faculté comme : le gain de temps, l'efficacité dans la gestion de l'information partagée et la maîtrise des coûts d'implémentation. L'ERP open-source et plus particulièrement OpenERP est la solution élue parmi les différents progiciels de gestion intégrée que nous avons étudié. La phase suivante à consisté de paramétrer les modules d'OpenERP avec les modules fonctionnels de la faculté (Ressources humaines). Comme, la faculté possède des modules fonctionnels non prise en charge par OpenERP à savoir la gestion des stages des enseignants. Donc, pour le développement de ce module, nous avons débuté par une analyse conceptuelle et par la suite nous avons développé l'ensemble des vues et des objets. Finalement, nous avons implémenté ces modules au sein d'OpenERP afin d'obtenir un système globale de la faculté.

Coté personnel, travailler dans le monde de l'open source, et plus précisément sur un PGI tel que OpenERP. Qui nous a permit d'acquérir des connaissances dans une large partie des technologies notamment le langage python et l'utilisation du framework « OpenObject », et de rejoindre ainsi une communauté mondiale de plus de 1500 individus impliqués eux aussi dans la recherche et le développement de nouveaux modules, afin de faciliter l'intégration d'une telle solution dans tous les domaines professionnels et sociaux.

Comme perspective, nous proposons la amélioration de module « stage » et de continuer le paramétrage du module de la comptabilité afin de garantir une meilleur gestion des finances de la faculté de des sciences de la nature et de la vie.

## **Webographie**

- [1] <http://fablain.developpez.com/tutoriel/presenterp/> date de la dernière consultation 05/2014
- [2] <http://www.erp-logiciel-gestion.com/> date dernière consultation 05/2014
- [3] [http://tics-supplychain.over-blog.com/pages/Presentation\\_generale\\_des\\_ERP-4903708.html](http://tics-supplychain.over-blog.com/pages/Presentation_generale_des_ERP-4903708.html)  
dernier accès : 04/2014
- [4] <http://www.openerp.com/> date dernière consultation 06/2014
- [5] <https://doc.openerp.com/> dernier accès : 06/2014
- [6] <http://flashq.free.fr/xml/xmlintro.htm> dernier accès : 05/2014
- [7] <http://www.moselotteinformatique.fr/ordinateurs-pc/ubuntu.html>
- [8] <http://www.postgresql.org/> dernier accès : 05/2014
- [9] <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc31018.1653/doc/html/rad1232633073807.html> 05/2014 dernier accès :/2014
- [10] [http://fr.wikipedia.org/wiki/Two\\_Tracks\\_Unified\\_Process](http://fr.wikipedia.org/wiki/Two_Tracks_Unified_Process) dernier accès

## **Bibliographie**

- **Ouvrage :**

- [11] ROQUES Pascal, Franck VALLÉE, *UML 2 en action, De l'analyse des besoins à la conception*, Paris, Edition Eyrolles, 2007,394 pages.
- [12] GREGORY Smith, *Bases de données PostgreSQL, 9.0 : Gestion des performances*, Pearson 2011, 444 pages.
- [13] GERARD Swinnen, *Apprendre à programmer avec Python3*, Edition Eyrolles, Paris, Février 2012, 473 pages.

- **Projet fin d'étude :**

- [14] Ait mlouk Addi, *Conception et réalisation d'une application de gestion, intégrée au sein de la société Eone Group basée sur OpenERP, 2013*.université Cadi Ayyad.
- [15] HAML I Marouane, *Analyse de besoin, adaptation et intégration d'OpenERP pour la gestion d'officine, 2012*, université Abdelmalek Essaâdi.

# ANNEXE

## **ANNEXE A : Méthode installation**

### **1. La démarche d'installation d'OpenERP Sous linux (distribution ubuntu):**

#### **1.1 Construire un serveur**

J'installe uniquement le strict minimum de la routine d'installation, Après le redémarrage du serveur pour la première fois j'installe le paquetage **openssh-server** et **denyhosts** pour ajouter un degré de protection contre les attaques par force brute.

```
sudo apt-get install openssh-server denyhosts  
sudo apt-get update  
sudo apt-get dist-upgrade
```

#### **1.2 créer un utilisateur openerp**

Après le redémarrage du serveur pour la première fois j'installe le paquetage **openssh-server** et **denyhosts** pour ajouter un degré de protection contre les attaques par force brute. Dans un premier temps on se connecte en tant que root sous le terminal linux et on crée un nouveau utilisateur « openerp » :

```
sudo adduser --system --home=/opt/openerp --group openerp
```

#### **1.3 installer le package PostgreSQL**

Après l'ajout de l'utilisateur « openerp » on installe le serveur de base de données postgresql

```
sudo apt-get install postgresql
```

#### **1.4 Créer l'utilisateur de base de données openerp**

Dans le terminal on se connecte en tant que postgres.



```
sudo su postgres
```

On créer l'utilisateur de base de donnée « openerp »

```
createuser --createdb --username postgres --no-creatorole --no-superuser --pwprompt openerp
```

## 1.5 Installer les dépendances

L'utilisation d'OpenERP nécessite un ensemble des bibliothèques python on les installe comme suit :

```
sudo apt-get install python-dateutil python-docutils python-feedparser python-gdata \  
python-jinja2 python-ldap python-libxslt1 python-lxml python-mako python-mock python-openid \  
python-psycopg2 python-psutil python-pybabel python-pychart python-pydot python-pyparsing \  
python-reportlab python-simplejson python-tz python-unittest2 python-vatnumber python-vobject \  
python-webdav python-werkzeug python-xlwt python-yaml python-zsi
```

Sinon l'ors de l'installation d'OpenERP toutes les dépendances python sont installer automatiquement (c'est l'avantage d'utilisation d'Openerp sous linux par rapport à Windows qui oblige l'installation de ces dépendances manuellement).

## 1.6 Installez le serveur OpenERP

```
wget http://nightly.openerp.com/7.0/nightly/src/openerp-7.0-latest.tar.gz
```

```
sudo tar xvf ~/openerp-7.0-latest.tar.gz
```

```
sudo chown -R openerp: *
```

```
sudo cp -a openerp-7.0 server
```

## 1.7 Création d'une base de données

Pour fonctionner il faut créer une base de données qui correspondra à une entreprise, il est possible de créer plusieurs bases de données. En saisissant `http://localhost:8069` dans un navigateur.

- Le mot de passe est enregistré
- Donnez un nom à la base de données
- NE PAS CHARGER la démonstration
- Choisissez Français/Français
- Saisissez le mot de passe choisi lors de l'installation de PostgreSQL

The screenshot shows the 'Create Database' interface in OpenERP. The page has a dark header with 'Back to Login' on the right. The main content area is divided into a left sidebar and a main form. The sidebar, titled 'Database Management', contains a 'Create' button and a list of actions: Duplicate, Drop, Backup, Restore, and Password. The main form, titled 'Create Database', has a red 'Create' button at the top left. It contains several input fields: 'Master password' (masked with dots), 'New database name', 'Load Demonstration data' (checkbox), 'Default language' (dropdown menu set to 'English (US)'), 'Admin password', and 'Confirm password'.

## 2. Configuration de service messagerie

The screenshot shows the 'Serveurs de ... / (Google Mail)' configuration page. At the top, there are buttons for 'Enregistrer' (red) and 'ou Annuler'. Below this, the configuration is organized into sections: 'Description' with 'Google Mail' and 'Priorité' set to '10'; 'Informations sur la connexion' with 'Serveur SMTP' as 'smtp.googlemail.com', 'Port SMTP' as '465', and 'Débogage' as an unchecked checkbox; and 'Sécurité et authentification' with 'Sécurité de la connexion' set to 'SSL/TLS', 'Nom d'utilisateur' as 'yaslaifa.24@gmail.com', and 'Mot de passe' masked with dots. A 'Test de connexion' button is located at the bottom left.

## ANNEXE B: Développement de modules OpenERP avec le Framework « Open Object »

Toutes les données OpenERP sont accessibles à travers des "objets". Par exemple, il y a un objet «formation.stage» qui accède aux données du stage. Pour définir un nouvel objet, on doit définir une nouvelle classe dans Python puis l'instantier. Cette classe doit hériter de la classe osv module osv. Le fichier: addons/module\_name/module\_name.py :

```
class object_name (osv.osv) :
    _name = "module_name.object_name"
    _columns = {
        'field_name' : fields.field_type('Libellé du champs', ...),
        'other_field_name' : fields.other_field_type(...),
    }
    object_name()
```

Exemple:

```
class formation_stage (osv.osv):
    _name = 'res.partner'
    _columns = {
        'name': fields.char (u 'Name', size=100, required=True),
        'city': fields.char (u'Ville',size=100),
        'country' : fields.many2one ('res.country', u'pays', size=64),
    }
    formation_stage()
```

Deux éléments sont obligatoires (*\_name* et *\_columns*), le reste est optionnels. Les noms des champs doivent respecter les normes suivantes :

- Tout doit être minuscule.
- Pas d'espace.
- Pas de caractères spéciaux.

Voici un petit exemple d'un module standard qui vous aidera à mieux comprendre comment ça marche:

## 1.1 Structure de l'exemple de module :

```
_init_.py  
_openerp_.py  
Stage.py  
Stage_view.xml
```

Créer le dossier stage dans \server\bin\addons

- Créer le fichier « \_init\_.py » dans le dossier stage ce fichier contiendra l'instruction suivante :

```
Import stage
```

- Cette instruction permet d'importer le module (les classes) stage
- Créer le fichier « \_openerp\_.py » dans le dossier stage qui correspond au fichier config de module.

```
{  
    "name": u"Gestion stage ",  
    'complexity': "normal",  
    "version": "1.0",  
    "depends": ["base", "hr", "mail"],  
    "author": " Laifa Yasmina ",  
    "website" : "",  
    'icon': '/stage/static/src/img/icon.png',  
    "category": "Vertical functionality",  
    'summary': 'stage, type du stage, manifestation, itinéraire',  
    'description': ""  
}
```

Créer le fichier « stage.py » dans le dossier :

- Ce fichier contiendra toutes les classes du module.
- On commence par une classe permettant de gérer les stages

```

from osv import fields, osv
class formation_stage (osv.osv) :
    _name = 'formation.stage'
    _columns = {
        'name' : fields.char(u'Numéro',size=100,required=True),
        'responsable' : fields.many2one('hr.employee',u'Enseignant',required=True),
        'country' : fields.many2one('res.country',u'Pays',size=100),
        'city' : fields.char(u'Ville',size=100),
        'user' : fields.many2one('res.users',u'Responsable',required=True),
        'date_start' : fields.date(u'Date Debut',required=True),
        'date_end' : fields.date(u'Date Fin',required=True),
        'subject' : fields.float(u'Sujet'),
        'type_stage' : fields.many2one('formation.stage.type',u'Type'),
        'work_ids' :fields.many2one('formation.work.type',u'Type Travail'),

        'manif_ids' :fields.one2many('stage.manifest','manifest',u'Manifestation'),
        'duration' : fields.function(_get_duration, type='integer', string='Dure en Jours'),
        'iten_ids' :fields.one2many('stage.itin','iten',u'Etineraire'),
        'state' : fields.selection([('draft',u'En
cours'),('done',u'Validé'),('cancel',u'Annulé']],u'Statut',required=True),
    }

formation_stage ()

```

## 1.2 Les interfaces graphiques

Après la création des classe il nous reste a créé des interfaces graphique désiré avec la syntaxe XML Et des balises propre pour openerp comme il présente le doc suivant :

Créer le fichier « *stage\_view.xml* » dans le dossier Stage qui correspond au fichier interface de module, voici une partie :

```

<openerp>
  <data>

  <!-- search -->

  <record model="ir.ui.view" id="formation_stage_search">
    <field name="name">formation.stage.search</field>
    <field name="model">formation.stage</field>
    <field name="type">search</field>
    <field name="arch" type="xml">
      <search string="Recherche travaux">
        <group>
          <filter icon="terp-stage" string="En cours" domain="[(('state','=', 'draft'))]"
help=""/>
          <filter icon="terp-stage" string="Terminé" domain="[(('state','=', 'done'))]"
help=""/>
          <filter icon="terp-stage" string="Annulé" domain="[(('state','=', 'cancel'))]"
help=""/>
        </group>
        <newline/>
        <field name="name"/>
        <field name="date_start"/>
        <field name="user"/>
        <field name="responsable"/>
        <newline/>
        <group expand="0" string="Grouper par ...">
          <filter string="Utilisateur" icon="stock_person" domain="[]"
context="{ 'group_by': 'user' }"/>
          <filter string="Responsable" icon="terp-go-year" domain="[]"
context="{ 'group_by': 'responsable' }"/>
          <filter string="Date Debut" icon="stock_person" domain="[]"
context="{ 'group_by': 'date_start' }"/>
          <filter string="Pays" icon="stock_person" domain="[]"
context="{ 'group_by': 'country' }"/>
        </group>
      </search>
    </field>
  </record>

```

### 1.3 Un rappel important pour les programmeurs :

- Les objets d'OpenERP sont appelés classes si on parle au terme de la programmation orientée objet
- Et une ressource est appelée objet, l'instance de la classe.

#### ➤ Définitions des objets en OpenERP

Pour définir un nouvel objet, on définit une nouvelle classe et après on l'instancie, cette classe doit hériter de la classe **OSV** du module **OSV** d'OpenERP.

Ainsi, la première ligne de la définition de l'objet sera toujours comme suivant:

```
class nom_objet(osv.osv):
    _name = 'nom de l'objet'
    _columns = {...}
    ...
    nom_objet()
```

Donc, l'objet se définit en déclarant quelques attributs avec des noms prédéfinis dans la classe, deux de ces attributs sont obligatoires à savoir **\_columns** et **\_name**, les autres sont optionnels.

Les attributs prédéfinis sont:

- **\_auto**: Prend True /False et détermine si la table correspondante dans postgres doit se générer automatiquement à partir de cet objet. Mettre `_auto=False` pourra être utile dans les cas où on veut créer des objets basés sur des vues de postgresql sans créer des tables.
- **\_columns(obligatoire)**: on y définit les champs (les attributs de la classe si on parle au terme d'orienté objet) de l'objet
- **\_constraints**: on y détermine des restrictions sur l'objet (on en parlera ultérieurement)
- **\_defaults**: on définit les valeurs par défaut pour quelques champs de l'objet.
- **\_inherit**: on met l'objet père que l'actuel objet hérite. On va le détailler dans deux types:
  - o ORM - Object Declaration - `_inherit (1)` : Réalise un héritage d'un objet existant, mais crée un nouvel objet, exemple :

```
class formateur(osv.osv):
    _name = 'formateur'
    _inherit = 'res.partner'
    _columns = {
        'lang_ids': fields.many2many('res.lang', 'res_lang_partner_rel',
        'partner_id', 'lang_id', 'Languages')}
    formateur()
```

- o ORM - Object Declaration - `_inherit (2)` : Réalise un héritage d'un objet existant, ajoute des caractéristiques à l'objet dont nous héritons, exemple :

```
class res_partner_add_langs(osv.osv):
    _name = 'res.partner'
    _inherit = 'res.partner'

    _columns = {
        'lang_ids' : fields.many2many('res.lang', 'res_lang_partner_rel',
        'partner_id', 'lang_id', 'Languages')}
    res_partner_add_langs()
```

- **\_name(obligatoire)**: le nom de l'objet, la valeur par défaut est None
- **\_order**: on met le nom du champ de l'objet actuel qui sera comme critère de tri des résultats des méthodes search et read. Sa valeur par défaut est 'id'
- **\_sql**: on met le code sql pour l'exécuter dans la création de l'objet (seulement si `_auto=True`)
- **\_table**: nom de la table sql correspondante a cet objet, sa valeur par défaut est celle de `_name` en remplaçant les points (.) par des tirets (\_)

### ➤ Les champs des objets

Les objets peuvent contenir différents types de champs. Ces types s'articulent dans trois catégories: types **simples**, types **relationnelles** et champs **fonctionnels**.

Les types simples englobent les entiers, les réels, les booléens et les chaînes de caractère..., les champs de type relationnel permettent de représenter les relations entre les objets (**one2one**, **many2one**, **many2many**). Les champs fonctionnels sont des champs spéciaux parce qu'ils ne sont pas enregistrés dans la base de données mais plutôt sont calculables à partir d'autres champs dans le temps réel.

La signature de la méthode d'initialisation de la classe dont hérite tout champ déclaré dans OpenERP( ..... /osv/fields.py).

```
def init (self, string='unknown', required=False, readonly=False, domain=[], context="",
states={}, priority=0, change_default=False, size=None, ondelete="set null",
translate=False, select=False, **args) :
```

Ainsi les paramètres optionnels communs pour tous les types de champs sont comme suit:

- **change\_default: (True/False)** quand ce champ prend la valeur booléenne **True**, le programme permet à l'utilisateur d'établir des valeurs par défaut dans autres champs qui dépendent de la valeur de ce champ. Sa valeur par défaut est: **False**. Exemple:(dans `res.partner.adress`)



- **'zip': fields.char('Zip', change\_default=True, size=24)**, dans ce cas, les utilisateurs peuvent mettre des valeurs par défaut dans les champs de la vue qui dépendent du champ 'Zip'(code postale). Par exemple, l'utilisateur peut programmer que le programme remplit automatiquement le champ **'city'** à Barcelone si le code postale prend la valeur '08000'.

- **help:** montre un message d'aide lorsque la souris est au-dessus de ce champ. Exemple:

*'name': fields.char('name', help='le nom s'affiche.')*

- **Readonly:(True/False)** détermine si le champ sera éditable ou non. Valeur par défaut=False

- **required: (True/False)** détermine si le champ est obligatoire ou non, signalons que OpenERP refuse d'enregistrer une ressource si un champ obligatoire est vide. Donc il est obligatoire de remplir tous les champs obligatoires avant d'enregistrer une ressource. Valeur par défaut=False

- **states:** ce paramètre permet de définir des autres attributs pour ce champs qui seront disponibles selon des états déterminés de la ressource. Format: **states= {'état de la ressource': [liste des attributs]}** Avec **liste des attributs** est une liste de tuples de la forme: **[('nom de l'attribut', valeur), ...]**. Valeur par défaut ={}  
Exemple: (dans **account.transfer**)

*'partner\_id': fields.many2one('res.partner', 'Partner', states={'posted [('readonly',True)]})*,

Dans cet exemple, lorsque l'état de la ressource est **'posted'**, l'attribut 'partner\_id' sera en lecture seule.

- **String:** c'est l'étiquette du champ. Valeur par default=unknown

- **translate:** (True/False) détermine si ce champ doit être traduit.(géré par le système de traduction). Valeur par défaut=False

Finalement, voici les paramètres optionnels spécifiques pour quelques types de champs :

- **domain:** sert à restreindre le domaine des ressources, ce paramètre est valide seulement pour les champs de type relationnel. Valeur par défaut=[]

Exemple: *domain=[('nom','=',valeur)]*

- **invisible:** (True/False) pour cacher la valeur du champ dans les formulaires (mots de passes...). Valeur par défaut=False

- **selection:** sert à sélectionner le niveau de recherche par défaut dans les vues. si ce paramètre prend la valeur '1', ceci signifie que le champs toujours apparaît dans le filtre de recherche dans la vue qui manifeste ce champs. Et s'il prend '2', le champ apparaît seulement dans la recherche avancée qui se manifeste lorsqu'on clique sur le symbole +

- **on\_change:** lance une fonction (défini sur l'objet ORM qui contient ce champ) quand la valeur de ce champ change dans une vue. Exemple: *on\_change = "onchange\_shop\_id(shop\_id)"*

➤ **Types de champs :**

▪ **Champs simples:**

- **boolean:** le champ boolean prend deux valeurs (True/False)

**Syntaxe:** `fields.boolean('Nom du champs' [, Parametres optionnels])`,

- **integer :** si la valeur voulue est un entier.

**Syntaxe:** `fields.integer('Nom du champs' [, Parametres optionnels])`

- **Float :** si la valeur voulue est un nombre flottant (réel).

**Syntaxe:** `fields.float('Nom du champs' [, Parametres optionnels])`,

Note: il y'a un paramètre spécifique `digit` pour le champ de type float, ce paramètre précise le nombre de chiffres après la virgule et précise le nombre des chiffres significatif c'est à dire le nombre de chiffres après et avant la virgule.

Exemple:

*'rate' : fields.float('le taux', digits=(12,6) [, Parametres optionnels])*

- **char:** Une chaîne de caractères de longueur limitée. Le paramètre obligatoire **size** détermine sa longueur

**Syntaxe :** `fields.char('Nom du champs', size=n [, Parametres optionnels])`, ou n est un entier.

- **text:** un texte de longueur illimitée

**Syntaxe :** `fields.text('Nom du champs' [, Parametres optionnels])`,

- **Date :** ce champs détermine la date .

**Syntaxe :** `fields.date('Nom du champs' [paramètres optionnels, ])`,

- **datetime** : Permet l'édition de la date et l'heure dans le même champ.

**Syntaxe** : `fields.datetime('Nom du champs' [paramètres optionnels, ])`,

- **Binary**: pour l'image, Exemple : `'image': fields.binary('Image')`

- **selection**: ce champ permet à l'utilisateur de sélectionner plusieurs valeurs prédéfinis.

**Syntaxe**: `fields.selection([( 'n', 'non confirmé'), ('c', 'Confirmé')], 'Nom du champs' [, paramètres optionnels])`,

- ✚ **Remarque**: le format du paramètre de sélection est une liste de tuples: `[('clé', 'chaîne de caractère à montrer'), ...]` ou la clé s'enregistre dans la ligne de la table (correspondante à l'objet qui contient le champ) comme une valeur prédéfinie.

- **Champs fonctionnels:**

Le champ fonctionnel est un champ dont la valeur est calculée par une fonction (au lieu de s'enregistrer dans la base de données). Dans les cas spéciales (pour faciliter la recherche ou la consultation) les champs de type fonctionnel peuvent être sauvegardés dans la base de données avec le paramètre **STORE** (il va prendre la valeur booléenne True). Ils sont actualisés par les fonctions et non par les utilisateurs.

**Paramètres**: `fncnt, arg=None, fncnt_inv=None, fncnt_inv_arg=None, type="float,`

`fncnt_search=None, obj=None, method=False,`

- `fncnt_inv`: Fonction inverse pour écrire

- `fncnt_search`: Fonction permettant de réaliser une recherche sur le champ `method`: Si True, il s'agit d'une méthode d'instance d'`openErp`, sinon, fonction

- `type`: Indique le type d'élément (char, integer, float, boolean, date, datetime)

- `store`: Si True, stocke dans la base de données la valeur du champ (par défaut False)

`fncnt`: c'est la fonction qui calcule la valeur du champs avec une méthode d'`openErp` ou une fonction globale, Si `method=True`, la signature de méthode sera comme suit:

```
def fncnt(self, cr, uid, ids, field_name, arg, context)
```

Exemple:

```
_columns = {
'avg': fields.function(_get_average_value, fncnt_inv=_something_write,
fncnt_search = _something_search, method=True, string="Fields", type="float", store=True)
}
```

➤ **champs relationnels**

- ✓ **ORM - Champs - Relationnel - one2many, exemple : Un instructeur donne plusieurs formations**

```
class openacademy_instructor(osv.osv):
    _name = 'openacademy.instructor'
    _columns = {
        'training_ids': fields.one2many('openacademy.training',
        'instructor_id', 'Trainings')
    }
```

- ✓ **ORM - Champs - Relationnel - many2one, exemple : plusieurs formations sont données par un instructeur**

```
_columns = {
    'instructor_id': fields.many2one('openacademy.instructor', 'Instructor')
}
```

- ✓ **ORM - Champs - Relationnel - many2one & one2many**

Comment retenir la différence ?

- Champ **many2one**: plusieurs (*many*) de ces objets sont possédés par un (*one*) objet parent
- Champ **one2many**: l'objet (*one*) possède plusieurs (*many*) enfants

- ✓ **ORM - Champs - Relationnel - many2many, exemple :**

```
_columns = {
    'participant_ids': fields.many2many('openacademy.participant',
    'openacademy_training_participant_rel', 'training_id', 'participant_id', 'Participants')
}
```

- Un participant peut suivre plusieurs formations
- Une formation est donnée à plusieurs participants

- ✓ **ORM – Méthodes spéciales – Méthodes prédéfinies**

Méthodes prédéfinies :

- **create**: Création d'un enregistrement
- **write**: Mise à jour d'un enregistrement
- **unlink**: Suppression d'un enregistrement
- **read**: Lecture de champs d'un enregistrement
- **copy** : Duplication d'un enregistrement
- **search**: Recherche d'enregistrement

- **browse**: Récupération d'enregistrement via critères de recherche
- **name\_get**: Retourne uniquement que le nom et l'identifiant du record
- **name\_search**: Réalise une recherche du nom sur base d'un domaine
- **init** : Permet de créer une vue si `_auto = False`
- **\_auto\_init** : Permet de créer un index ou un objet SQL si nécessaire

# Résumé

Pour améliorer sa performance, l'entreprise d'aujourd'hui vise à automatiser la gestion interne de ses activités en faisant appel à des technologies informatiques. D'ailleurs c'est le cas de la faculté qui souhaite optimiser la totalité de sa gestion autour d'un même système d'information à l'aide des progiciels de gestion intégrée connu sous l'acronyme ERP.

Notre projet consiste à identifier et analyser les besoins liés à la faculté qui s'articulent autour des modules fonctionnels à savoir : la gestion des ressources humaines, puis la gestion des stages.

Pour y arriver, on choisi le types d'ERP open-source et plus particulièrement OpenERP.

A l'aide de ce système unifié, les utilisateurs de différents métiers travaillent dans un environnement applicatif identique qui repose sur une base de données unique. Ce modèle permet d'assurer l'intégrité des données, la non-redondance de l'information, ainsi que la réduction des temps de traitement.

La réalisation de ce projet, quant à elle, est composée de deux parties essentielles : le paramétrage et le développement spécifique des modules fonctionnels non pris en charge par OpenERP à savoir la gestion des stages.

L'étape finale consistée à faire des simulations et tests pour s'assurer que le travail a bien été fait, et corriger les bugs qui peuvent arriver, si jamais il y en a.

## Mots clés:

- ERP open source , OpenERP, Python, XML, module stage.

# Abstract

To improve its performance, today's business is to automate the internal management of its activities involving information technology. Moreover it is the case of the faculty wishes to optimize its entire management around a single information system using ERP systems known by the acronym ERP.

Our project is to identify and analyze the needs related to the faculty that revolve around functional modules namely: the management of human resources and management training.

To achieve this, we chose the open-source kind of ERP and especially OpenERP.

Using this unified system, users of different trades working in the same application environment based on a single database. This model ensures data integrity, non-redundant information and reducing processing time.

The realization of this project, in turn, is composed of two essential parts: the setup and development of specific unsupported by OpenERP management courses functional modules.

The final stage comprised to make simulations and tests to ensure that the work has been done, and fix bugs that can happen, if ever there was.

## Keywords:

- ERP open source, OpenERP, Python, XML, placement Module.