



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Centre Universitaire de Salhi Ahmed –Naâma-

Département : Mathématique et Informatique

## **MEMOIRE DE MASTER**

Domaine: Mathématique et Informatique

Filière: Informatique

Option: Systèmes d'Information

### **Thème:**

# **Intégration et personnalisation d'ODOO au sein du Centre Universitaire de Naâma**

### **Présenté par:**

GUENDOUI Oukacha

SADANI Messaouda

### **Encadré par:**

Mr.BENDIDA Aissam

**Année Universitaire : 2017 /2018**

## *Dédicaces*

*Ce travail n'aurait pu aboutir sans l'aide de plusieurs personnes. En commençant par rendre grâce à Dieu pour m'avoir donné la force de travailler et faire de mon mieux.*

*Je dédie aussi ce travail à tous ceux qui m'ont aidé à dépasser les moments difficiles.*

*À tous ceux qui auraient voulu partager ma joie. . .*

*Qu'ils trouvent en ce travail l'expression de ma gratitude pour l'attention qu'ils portent constamment à mes réussites.*

*Sadani Messaouda*

*Je dédie ce travail à :*

*A la mémoire de mon père,*

*A la mémoire de mon frère*

*A la mémoire de ma sœur,*

*A ma chère mère.*

*A mes sœurs et mes frères.*

*A ma belle-famille.*

**QUENDOUZI Oukacha**

## Remerciements

On remercie Dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de finir ce mémoire.

Je tiens à remercier Monsieur le président de jury qui a agréé d'apprécier ce modeste travail.

Les membres de jury qui ont accepté de juger ce projet à sa propre valeur. Néanmoins s'il ya des hommes à qui je tiens à exprimer mes sentiments de gratitude et respect, c'est bien mon encadrant Monsieur AISSAM BENDIDA pour son suivi, sa disponibilité et ses conseils.

Un remerciement spécial pour notre chef de département, pour le soutien et l'accompagnement tout au long de la formation de Master.

Finalement, j'exprime mes vifs et sincères remerciements à toute personne ayant participé de près ou de loin à la bonne réalisation de ce modeste travail.

# Sommaire

Liste des figures

Liste des tableaux

Introduction générale 1

## Chapitre 1 : Etat de l'art

Introduction 3

1. Introduction au monde des ERP 3

2. Les ERP libres 4

3. Etude comparative entre quelques ERP existants dans le monde de développement 4

3.1. Comparaison des ERPs en utilisation dans le monde 5

4. La plateforme Odoo 7

4.1. Historique des versions 8

4.2. Odoo et ses modules 9

4.3. Architecture fonctionnelle et technique d'Odoo 9

4.4. Structure d'un module Odoo 12

5. Conclusion 13

## Chapitre 2 : Etude préalable

Introduction 14

1.	Définition d'un système de gestion pédagogie :	14
2.	Objectifs	14
3.	Avantages & Caractéristiques	16
4.	Domaines d'application	17
5.	Étude de l'existant	17
5.1.	G-Pédagogie	18
5.2.	SEES	20
5.3.	SCODOC :	21
5.4.	APOGEE	22
6.	Présentation du projet	23
6.1.	Sous objectifs	23
6.2.	Présentation de la problématique :	24
6.3.	Motivation	25
7.	Conclusion	26
 Chapitre 3 : Conception et implémentation		
Introduction		27
1.	Les diagrammes UML ;	27
2.	Le diagramme de cas d'utilisation	27
3.	Diagramme de Séquence	28
4.	Diagramme de classes	31
5.	Diagramme d'état transition<Matière>	31
6.	Partie Implémentation	32
6.1.	Les outils utilisés	32
6.2.	Présentation de module	33
7.	Conclusion	38

Conclusion générale et perspectives

39

Bibliographie

# Liste des figures

1	La fréquence de recherche de quelque ERP open source . . . . .	5
2	Taux d'intérêt de recherches effectuées sur odoo/openerp/tinyerp . . . . .	6
3	Nombre de résultats sur le nombre de recherche Google pour le terme odoo . . . . .	6
4	Modules par défaut d'Odoo . . . . .	9
5	L'architecture technique d'ODOO. . . . .	10
6	Protocole de transport d'Odoo . . . . .	10
7	Architecture de déploiement Odoo [10] . . . . .	11
8	Structure d'un module Odoo . . . . .	11
9	Structure des fichiers d'un module Odoo . . . . .	12
10	L'Interface de système G-Pédagogie . . . . .	19
11	L'interface de logiciel SEES. . . . .	20
12	L'interface de logiciel SCODOC. . . . .	21
13	L'interface de l'APOGEE . . . . .	22
14	Diagramme de cas d'utilisation<<Administrateur>> . . . . .	27
15	Diagramme de cas d'utilisation<<Professeur>>. . . . .	28
16	Diagramme de cas d'utilisation<<Etudiant>>. . . . .	28
17	Diagramme de cas d'utilisation<<Surveillant>>. . . . .	28
18	Diagramme de Séquence<<Ajouter Absence>> . . . . .	29
19	Diagramme de Séquence<<Consulter Absence>>. . . . .	30
20	Diagramme de Séquence<<Affecter un module a un Professeur>>. . . . .	30
21	Diagramme de classe. . . . .	31
22	Diagramme d'état transition<<Matière>> . . . . .	32
23	Les dossiers et les fichiers du module : SGPU_PRINCIPAL . . . . .	34
24	Contenu du fichier : __manifest__.py du module : SGPU_PRINCIPAL . . . . .	35

25	Contenu : sgpu_principal_menu.xml / module : SGPU_PRINCIPAL .....	35
26	Contenu du fichier : etudiant.py / module : SGPU_PRINCIPAL .....	36
27	Contenu du fichier (vue) : filiere_vue.xml / module : SGPU_PRINCIPAL .....	37
28	Capture du module SGPU/ partie Etudiant .....	37
29	Capture du module SGPU/ partie Filière et Modules .....	38
30	Capture du module Gestion assiduité / partie Registre feuille de présence .....	38

## Liste des tableaux

1	Comparaison de quelques ERP open source	5
2	les différentes versions d'Odoo	8
3	Types des SGPS	18

---

# **INTRODUCTION GENERALE**

---

# Introduction générale

Les technologies de l'information changent la communauté entière autour de nous, devenant désormais un coefficient essentiel de la croissance qui influence sur tous les secteurs de notre société. L'enseignement supérieur est parmi les domaines qui bénéficient de l'utilisation de ces nouvelles technologies et assister tous les processus des universités, depuis l'arrivée des nouveaux étudiants à l'édition des diplômes lors de la fin des études.

Un ERP (Enterprise Resource Planning) ou également appelé PGI (Progiciel de Gestion Intégré) en tant qu'un système d'information a pour but de gérer et suivre au quotidien, l'ensemble des informations et des services opérationnels d'une entreprise. Il a connu leur évolution en bénéficiant de l'avancée nécessaire des systèmes d'informations. En conséquence, il était attrayant de changer tous les logiciels de gestion de l'entreprise par un intégré KIT qui offre des mise à jours sur les logiciels actuels par les réformer en utilisant les services web et les systèmes centralisée et les systèmes distribués.

Notre projet de fin d'études basé sur ODOO[1], un PGI open-source extrêmement modulaire grâce à sa flexibilité, et le travail consiste à réaliser un module pour la gestion pédagogique.

Notre idée commence par effectuer d'abord une analyse des besoins, afin de bien souligner les différentes fonctionnalités requises de notre futur module, ensuite à chercher parmi ces fonctionnalités celles qui sont déjà proposées par ODOO. Ceci, afin d'adapter celles qui existent au contexte algérien et surtout d'entamer le développement des fonctionnalités restantes qui n'existaient pas encore.

Notre mémoire est scindé en trois chapitres. Le premier montre en général les concepts de base à propos des ERP. Le second focalise autour de l'étude préalable sur les systèmes de gestion pédagogique et la conception du module. Le dernier développe les étapes du développement du module. En fin, ce mémoire se termine par une conclusion sur l'apport du travail réalisé.

---

# **CHAPITRE 1**

---

## **ETAT DE L'ART**

### **Introduction**

Nous débutons dans cette première partie par une présentation des ERP existantes dans le monde de développement pour les entreprises, par une mise en disposition de notre sujet, en spécifiant le cadre de son élaboration y afférant. Ensuite, nous prévoyons une étude comparative de quelques solutions libres afin de justifier notre choix de travailler avec ERP Odoo.

### **1. Introduction au monde des ERP**

L'acronyme ERP signifie Enterprise Resource Planning traduit en français par Progiciel de Gestion Intégré ou PGI. ERP est le terme le plus couramment utilisé. Un ERP est un progiciel qui permet de gérer d'une manière centralisée l'ensemble des processus d'une entreprise intégrant l'ensemble de ses fonctions comme la gestion des ressources humaines, la gestion financière et comptable, l'aide à la décision, la vente, la distribution, l'approvisionnement, la production ou encore du e-commerce. Le principe fondateur d'un ERP est de construire des applications informatiques correspondant aux diverses fonctions citées précédemment de manière modulaire sachant que ces modules sont indépendants entre eux, tout 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 et qui permet, lorsqu'une donnée est enregistrée dans le système d'information, de la propager dans les modules qui en ont l'utilité, selon une programmation prédéfinie [2].

On peut parler aussi d'un ERP lorsqu'on est en présence d'un système d'information composé de nombreux applications partageant une seule et même base de données, à travers un système automatisé prédéfini et éventuellement paramétrable.

### **2. Les ERP libres**

Un logiciel libre est un logiciel dont l'utilisation, l'étude, la modification et la duplication par autrui en vue de sa diffusion sont permises, techniquement et légalement<sup>1</sup>, ceci afin de garantir certaines libertés induites, dont le contrôle du programme par l'utilisateur et la possibilité de partage entre individus[3].

Pour être libre, un logiciel doit respecter quatre libertés fondamentales :

- La liberté d'exécution,
- La liberté d'étude,
- La liberté de redistribution des copies,
- La liberté d'amélioration.

De plus, comme tout logiciel libre, les ERP libres donnent la garantie de travailler sur des standards ouverts et donc inter opérables qui offre des avantages stratégiques pour beaucoup d'entreprise.

### **3. Etude comparative entre quelques ERP existants dans le monde de développement**

Les logiciels ERP sont particulièrement adaptés aux petites entreprises qui veulent maîtriser et optimiser leur chaîne de production et avoir de la visibilité sur le projet qu'elles veulent concevoir et développer. En matière de secteur, les logiciels ERP sont particulièrement prisés par les entreprises du domaine des services, des domaines de l'informatique et de l'Internet. Elles doivent en effet gérer des projets sur le long terme. De manière globale, les logiciels ERP s'adressent aux entreprises menant des projets nécessitant de multiples compétences et devant coordonner le travail de plusieurs équipes ou au moins de plusieurs personnes. On peut considérer qu'un logiciel ERP est nécessaire pour conduire un projet où une dizaine de personnes doit intervenir.

Voici un tableau récapitulatif de différentes solutions traitées et ses caractéristiques :

<i>ERP Open source</i>	<b>Couverture fonctionnelle</b>	<b>Environnement de travail</b>
<i>OpenBravo [5] (Année de création : 2001)</i>	Comptabilité, finance, achat, vente, stock, production, CRM, point de vente	J2EE, Postgresql ou oracle
<i>Dolibarr [6] (Année de création : 2003)</i>	Comptabilité, finance, achat, vente, stock, production, CRM	PHP, Postgresql
<i>Odoo [4] (Année de création : 2002)</i>	Comptabilité, finance, achat, vente, stock, production, CRM, point de vente, GED, e-commerce, RH	Python, XML, Postgresql
<i>WebERP [7] (Année de création : 2003)</i>	Comptabilité, finance, achat, vente, stock, production, RH	J2EE, JBoss
<i>ERPNEXT [8] (Année de création : 2008)</i>	Comptabilité, finance, achat, vente, stock, production, J2EE, RH	JBoss

Tableau 1: Comparaison de quelques ERP open source

### 3.1. Comparaison des ERPs en utilisation dans le monde :

Cette étude comparative entre les ERP a été réalisée en utilisant l’outil «Google trends », en traçant la fréquence de recherche des mots clés (Open ERP) sur le moteur de recherche Google dans une période de 12 mois derniers.

La figure 1 montre la fréquence de recherche de quelques ERP open source.

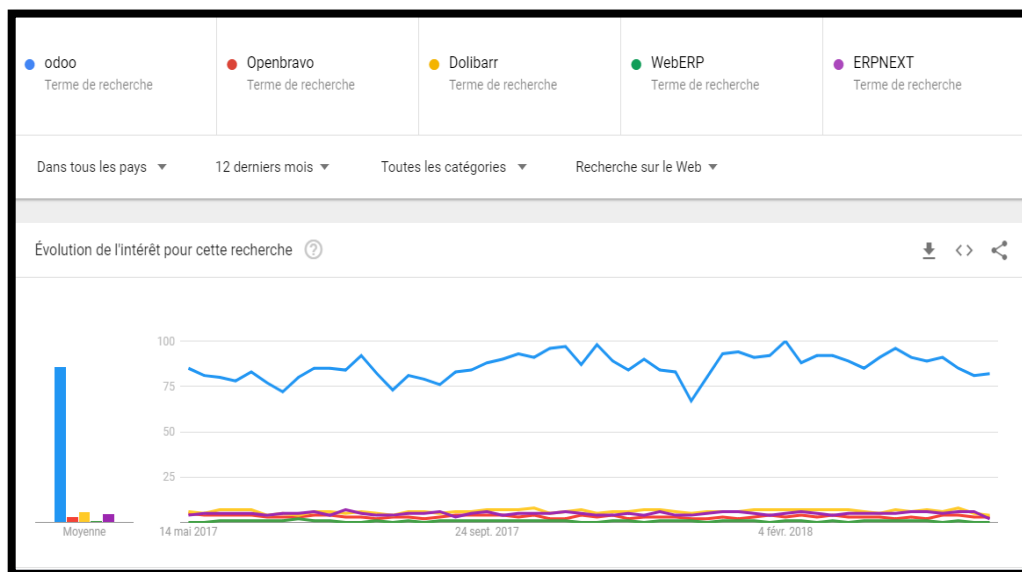


Figure 1: Taux d’intérêt de recherche effectués odoo/ OpenBravo/ Dolibarr/ WebERP/ERPNEXT

La figure présente le changement marque de «TinyERP» à «OpenERP » en 2008 puis vers « Odoo » en 2014, sans aucun impact significatif sur l'activité.

La figure 2 présente taux d'intérêt de recherche effectuée sur odoo/openerp/tinyerp

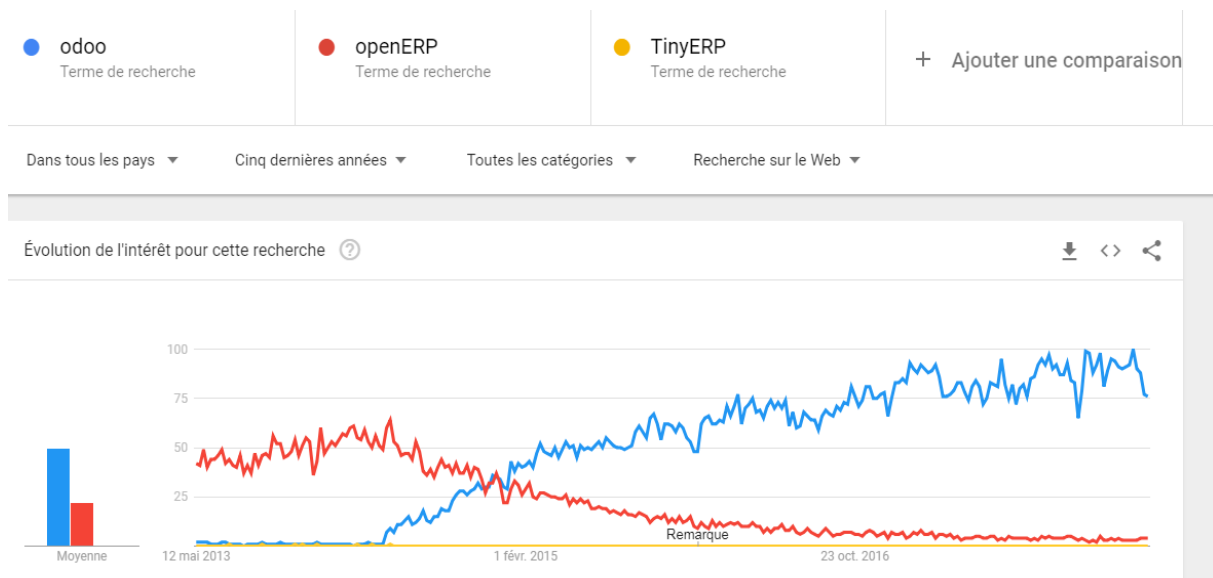


Figure 2: Taux d'intérêt de recherche effectuée sur odoo/openerp/tinyerp

Pour avoir une idée sur le nombre de ressources qui sont archivés dans les serveurs de Google. On a utilisé le moteur de recherche directement et on a obtenu les résultats suivants :

La figure 3 montre le nombre de résultats sur le nombre de recherche Google pour le terme odoo

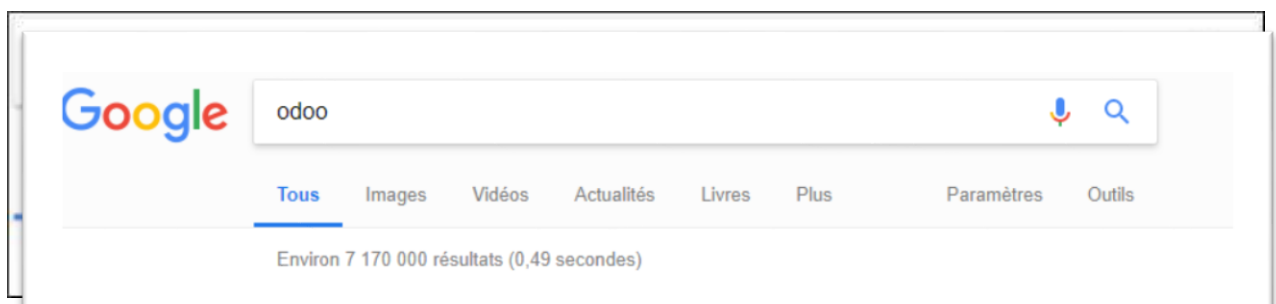


Figure 3: Nombre de résultats sur le nombre de recherche Google pour le terme odoo

Après cette comparaison entre les logiciels mentionnés précédemment, on a opté de travailler sur « Odoo » pour les avantages qu'il possède :

- Son interface conviviale et intuitive,
- La prise en main est immédiate;
- Le fait qu'il n'y a pas de coûts de licence.
- Un accès personnalisé et sécurisé à un portail Web où chaque utilisateur dispose de ses propres fonctionnalités.
- L'open source proposera un module CMS pour la gestion de sites web intégrés à l'ERP, un chat en ligne et de nouvelles fonctionnalités (gestion des achats, gestion de projets, production, gestion d'entrepôt, point de vente, ressources humaines, campagnes marketing, comptabilité & finance, CRM, facturation, générateur d'applications).
- Odoo s'appuie sur un réseau mondial de partenaires présents dans 110 pays. Capacité fonctionnelle, Fiabilité,
- Facilité d'utilisation, Rendement/Efficacité, Maintenabilité.

#### **4. Historique Odoo**

Lorsque Fabien Pinckaers crée en 2005 un logiciel destiné à faciliter et unifier la gestion d'entreprises, il lui donne pour nom : Tiny ERP. C'est alors, et déjà, un ERP open source.

Tiny ERP a grandi, s'est amélioré et a déjà connu 4 versions. Ce n'est plus un "minuscule" ERP, et il convoite désormais des clients qui comptent parmi les plus grandes entreprises mondiales. Par conséquent, il devient urgent de changer de nom, afin aussi d'être crédible devant ces clients.

En 2008, Tiny ERP devient ainsi Open ERP. Un logiciel de gestion intégrée désormais ouvert à tous, quelle que soit la taille de la structure, et gratuit.

ODOO permet de construire des applications informatiques (gestion des commandes, des stocks, de la paie, de la comptabilité, etc), modulaire et intégrée

au niveau des traitements offerts (les différents modules qui le composent sont indépendants mais parfaitement compatibles entre eux), ainsi rigoureux et cohérents au niveau des données gérées (partage d'une base de données unique et commune) et fournir l'ensemble des acteurs de l'entreprise une image unique, en plus il est cohérent et homogène de l'ensemble de l'information, Fusionner l'ensemble des processus de l'entreprise dans chacun des domaines qui la constituent et ce, dans une approche transversale qui optimise sa productivité, logiciel dans lequel le code source est à la disposition du grand public, généralement un effort de collaboration où les programmeurs améliorent ensemble le code source.

#### 4.1. Historique des versions

Le tableau 2 montre les différentes versions d'ODOO

Nom du logiciel	Version	Date de lancement
Tiny ERP	1.0	Février 2005
	2.0	Mars 2005
	3.0	Septembre 2005
	4.0	Décembre 2006
OpenERP	5.0	
	6.0	Octobre 2009
	6.1	
	7.0	Décembre 2012
Odoo	8.0	Septembre 2014
	9.0	Novembre 2015
	10.0	Octobre 2016
	11.0	Octobre 2017
	12.0	Prévue pour Octobre 2018

Tableau 2: les différentes versions d'Odoo

## 4.2. Odoo et ses modules

Grâce à la communauté open source, le catalogue de logiciels d'Odoo s'était développé bien plus rapidement que pour un éditeur de logiciels dits «propriétaires» (comme ceux de SAP[9], Microsoft[10], Sage[11], Oracle[12]) pas moins de 500 modules étaient déjà à disposition des entreprises. On notait déjà plus de 1000 installations par jour, devenant le logiciel de gestion le plus installé au monde.[13]

La figure 4 présente les modules par défaut d'Odoo



Figure 4: Modules par défaut d'Odoo

## 4.3. Architecture fonctionnelle et technique d'Odoo

Odoo possède une structure modulaire qui permet d'ajouter de nouveaux modules facilement pour étendre les fonctionnalités. Un module est un dossier avec une structure prédéfinie contenant du code Python et des fichiers XML, qui permet de définir la structure de données, formulaires, rapports, menus, procédures, flux de travail, etc. Lors de la première installation, on installe le

noyau d'Odoo avec un certain nombre de modules selon le profil d'installation choisit.

Odoo est basé sur une architecture client/serveur. Le serveur et le client communiquent via le protocole XML-RPC. XML-RPC c'est un simple protocole qui permet au client de faire des appels aux Procédures. 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[14] .

L'architecture d'ODOO se base sur l'architecture client/serveur comme présenté sur la figure suivante :

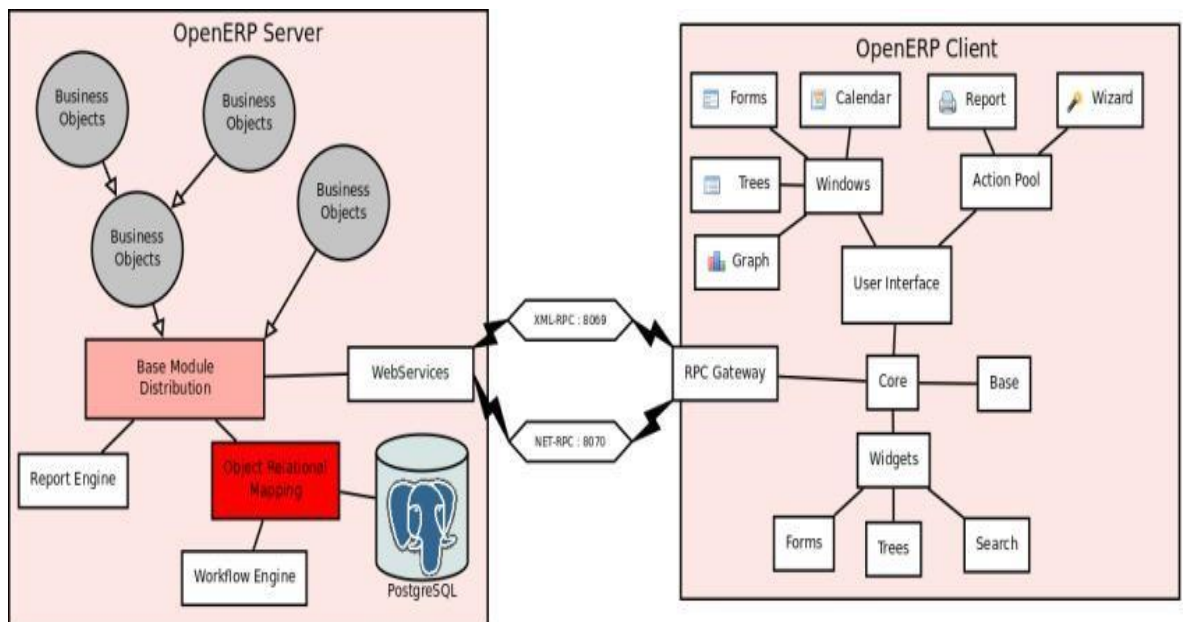


Figure 5. : l'architecture technique d'ODOO

Chaque composante possède un rôle bien défini :

**Base Module Distribution :** ce composant assure la communication distribuée au sein du serveur. En effet, une fois une requête arrive au serveur, ce mécanisme fait appel aux différents objets reliés à cette requête et rassemble toutes les informations afin de construire la réponse.

**Mapping objet-relationnel :** constitue le pont entre les objets métier et la base de données relationnelle. En effet, il crée une base de données objet qui peut être utilisé dans un langage de programmation orienté objet sans faire appel à la base

de données relationnelle. Cette couche permet d'assurer la consistance et la sécurité (gestion de droits par utilisateur et par groupe) des données.

**Web Services** : permet la connexion du serveur ODOO avec un client web via un navigateur web.

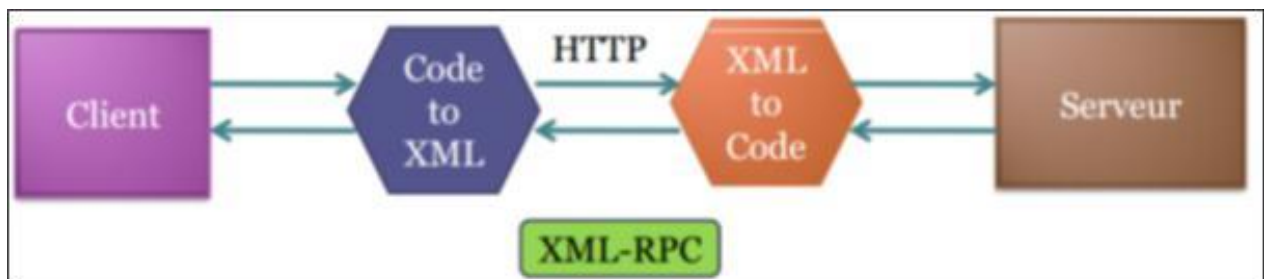
**Moteur de workflow** : ce moteur assure le suivi des flux métiers et l'enchaînement des tâches au sein d'un processus métier.

**Moteur de reporting** : prend en charge la génération des rapports dans OpenERP en faisant appel à des outils comme Open Report qui génère des rapports en PDF.

**Business Object** : représente les objets métiers d'ODOO construites, ils peuvent être représentés par différentes vues.

**Serveur de base de données PostgreSQL** : présenté par la suite, il assure le stockage et la persistance des données d'ODOO.

La figure 6 présente le protocole de transport d'Odoo



**Figure 6: Protocole de transport d'Odoo**

La logique d'Odoo 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 Odoo plus simple au développement et à la maintenance.

La figure 7 présente l'architecture de déploiement Odoo.

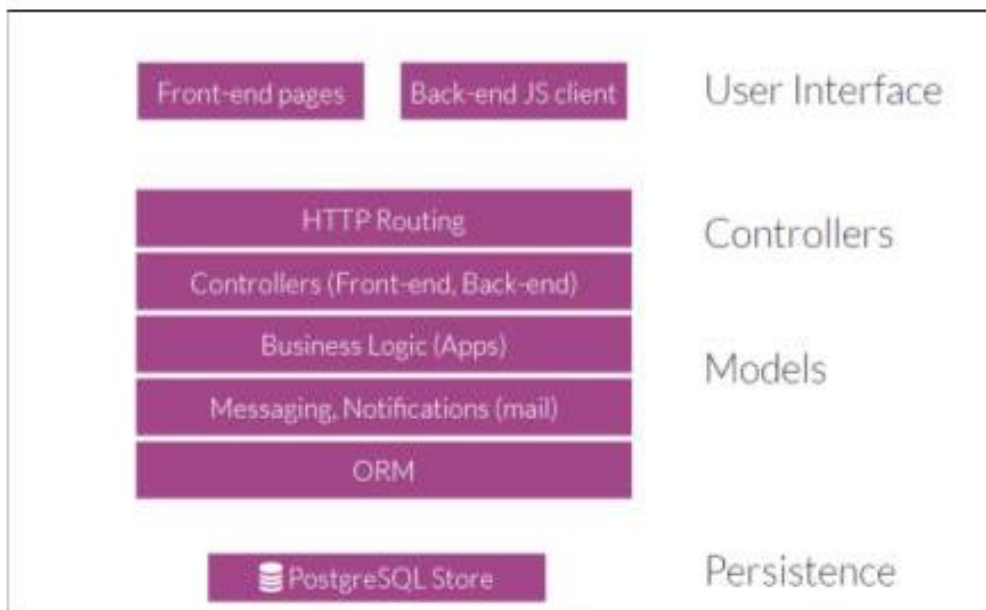


Figure 7 Architecture de déploiement Odoo

#### 4.4. Structure d'un module Odoo

La figure 8 présente la structure d'un module Odoo

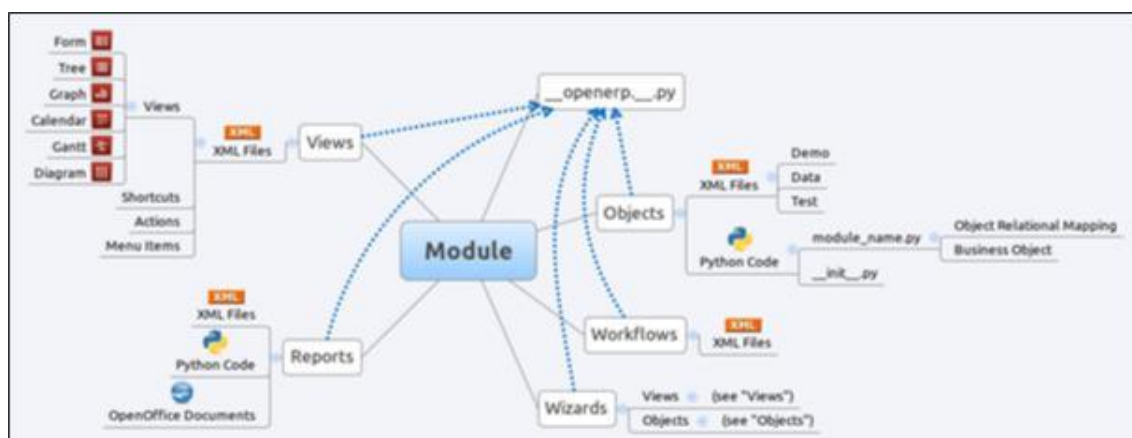


Figure 8: Structure d'un module Odoo

Pour créer un module Odoo, il y'a des étapes essentielles à suivre :

- Créer un package python dans le répertoire /addons portant le nom de votre

module, Créer un fichier de description du module openerp.py,

- Créer le fichier Python contenant les modèles (Classes + Méthodes),
- Créer des fichiers .XML pour définir les menus, les vues et les actions,
- Créer éventuellement des rapports, des assistants (Wizard) ou des flux de travail(Workflow)[15].

La figure 9 présente la structure d'un module Odoo

<b>Obligatoires</b>	<code>__init__.py</code>	Initialisation du package Python
	<code>__openerp__.py</code>	La fiche descriptive du module
	<code>module.py</code>	Les classes Python, les objets du module
	<code>module_view.xml</code>	Les vues (Formulaires, listes, graph,...), menus et actions
<b>Optionnels</b>	<code>demo</code>	Données de test et de démonstration
	<code>i18n</code>	Fichiers de traduction – Internationalisation
	<code>report</code>	Définition des rapports
	<code>security</code>	Déclaration des groupes et des droits d'accès
	<code>workflow</code>	Définition des workflow
	<code>wizard</code>	Définition des assistants

**Figure 9: Structure des fichiers d'un module Odoo**

## 5. Conclusion

Dans ce chapitre, nous avons présenté une étude comparative de quelques ERP et spécifié le choix de ces derniers pour réaliser notre application.

Nous passons par la suite au chapitre suivant où nous allons faire l'étude préalable de notre projet.

---

## **CHAPITRE 2**

---

### **ETUDE PREALABLE**

### **Introduction**

Au sein des établissements universitaires il y a énormément de systèmes informatiques et des plates formes déployées pour gérer les différentes tâches et activités administratives, financières, techniques ou pédagogiques.

Dans ce projet de fin d'études nous sommes intéressés aux systèmes de gestion des opérations pédagogiques. Et afin d'éviter le conflit entre ce dernier et les autres systèmes, nous présenterons à la suite de ce chapitre les différents objectifs et concepts essentiels liés à ce type de système.

#### **1. Définition d'un système de gestion pédagogie :**

Le système de gestion pédagogiques universitaire est un logiciel de gestion pédagogique (Multi-utilisateurs) permet le suivi administratif et pédagogique des étudiants des universités dès l'obtention du bac jusqu'au l'obtention du doctorat troisième cycle.

Le système de gestion pédagogique propose toutes les fonctionnalités nécessaires pour le suivi et le passage d'une année à une autre ou d'un cycle à un autre. [16]

#### **2. Objectifs**

Les systèmes de gestion pédagogiques sont intégrés dans l'espace numérique universitaire pour assurer la liste non exhaustive des objectifs suivants ;

- **Gain de temps :**

Le volume des données pédagogiques au sein des campus universitaires est devenu de plus en plus volumineuse grâce à la croissance rapide de nombre des étudiants, enseignants et les différentes formations ouvertes.

Ce dernier fait un handicap major en terme de temps face aux solutions traditionnelles proposées dans le marché pour traiter ce genre de problème. C'est pour ça la mise en place et le déploiement des systèmes de gestion pédagogique

est devenu plus d'une nécessité afin de faciliter les tâches et gagner beaucoup de temps par les différents acteurs de l'université.

- **Gestion des emplois du temps :**

L'optimisation des ressources physiques de l'université notamment les locaux pédagogiques (les amphis, salles et les laboratoires pédagogiques). En plus l'incohérence et les chevauchements dans la programmation des horaires découlant de l'utilisation des moyens traditionnels. Ces deux derniers invitent les nouveaux systèmes de placer la gestion d'emploi du temps comme un objectif principal dans ses architectures de fonctionnement.

- **Suivi d'assiduité :**

Les règlements intérieurs de gestion de l'université invoque des décisions importantes basé sur l'absence / présence des étudiants dans les activités pédagogiques et en particulier les TDs et les TPs charge à l'exclu de l'étudiant d'un module ou de la formation entière.

Donc il est important de mettre en place un moyen automatisé comme les systèmes de gestion pédagogique pour éviter n'importe quel erreur pour le suivi et traitement d'assiduité des étudiants ; et ne laisse pas l'être humain faire des graves fautes influant négativement sur la carrière des étudiants.

- **Saisie des notes, PVs et résultats de délibérations et publication / édition des bulletins... :**

Le bon déroulement des taches du fin semestre, fin d'année ou la fin de cycle dans l'ancien système nécessite des grands efforts et implique une réquisition presque total de toutes les ressources pédagogiques de l'université, que se soient les enseignants, les étudiants, toute le staff administratif et les différentes instances scientifiques (comités et conseils)

En effet la saisie des notes et préparation des PVs de délibération (calcul des moyens de module, unité et semestre, la liste des admis et ajournés et les pourcentages nécessaires...) sont des opérations lourdes et consomment

beaucoup des ressources humaines et matériels avec les solutions traditionnelles, et en plus ne donnent pas des bonnes résultats sur le plan pédagogique ou technique. Pour pallier ces inconvénients, le déploiement d'un système de gestion qui nous permettons de gérer ces derniers opérations est une mission fondamentale et principale dans les établissements de l'enseignement supérieur.

### **3. Avantages & Caractéristiques**

Les systèmes de gestion pédagogiques offrent des avantages importants aux institutions scientifiques et éducatives, ces qui leur permet d'exploiter ses ressources d'une façon optimale, parmi ces avantages et caractéristiques, on a listé les suivants :

- **Collaboration :**

L'objectif d'un système de gestion pédagogique (SGP) est de fournir un espace de travail collaboratif, ce dernier a pour but de faciliter et optimiser la communication entre les individus (services de scolarité, étudiants et enseignants) dans le cadre du travail ou d'une tâche non liée au travail.

- **Visibilité**

La visibilité d'une université fait un défi major pour l'université que se soit au contexte interne ou externe. Le SGP donne une grande visibilité interne entre les différents acteurs de l'université grâce à son disponibilité et les espaces associés pour chacun d'eux.

- **Transparence**

La confiance entre le staff administratif et les enseignants d'un part et avec les étudiants de l'autre part représente une valeur ajoutée pour l'université.

Parmi les critères techniques d'un système pour garantir cette caractéristique est d'être un système transparent. C'est pour ça les SGPs fournissent des efforts énormes pour assurer un degré élevé de transparence à travers la précision pour chacun de ses acteurs leurs rôle et leurs taches en détails.

- **Fiabilité**

Fiabilité d'un système et son aptitude à accomplir une fonction requise dans des conditions données et pour une durée déterminée.

*Rst(t) : fiabilité du système à l'instant t = probabilité pour qu'il n'y ait pas de défaillance sur l'intervalle [0, t]*

La fiabilité implique donc un contexte donné et un temps donné. C'est l'objectif principal de tous les systèmes informatiques. Alors les SGPs sont obligés d'augmenter leurs niveau de fiabilité pour garder leurs place dans les plateformes numériques de l'université.

Optimisation des ressources, mobilité, intégration dans des plateformes existantes et interaction avec d'autres systèmes sont aussi d'autres caractéristiques qui donnent une particularité exclusive pour les systèmes de gestion pédagogiques par rapport à d'autres systèmes informatiques.[17]

#### **4. Domaines d'application**

Selon les caractéristiques et les objectifs cités ci-dessus, ce type de système est très intéressant pour pas mal des domaines d'activités notamment les secteurs liés à l'éducation et l'enseignement

La taille des établissements éducatifs aujourd'hui est important (les primaires, CEMs et les lycées) rends l'usage de ces systèmes (SGPs) comme un outil nécessaire et indispensable pour leurs développement et leurs croissance.

L'enseignement supérieur, la formation professionnelle et le E-learning font l'un des domaines gourmands à l'application et l'utilisation de ce type du système de gestion pédagogique.

#### **5. Étude de l'existant**

Pour parvenir à présenter une bonne spécification des besoins, nous sommes tenus d'organiser et de bien analyser les fonctionnalités et les résultats attendus par notre projet. Dans cette optique, nous allons commencer par une étude des

outils et des solutions existantes pour pouvoir en dégager les défauts et mettre en valeur l'apport de notre solution.

Nous allons présenter et analyser un ensemble des systèmes de gestion pédagogiques locaux et internationaux, afin d'éviter leurs inconvénients et leurs limites.

Le choix de ces prototypes est basé sur la documentation disponible qui explique chacun d'eux, et l'accès possible à l'information pour avoir ses avantages et ses limites.

Les systèmes que nous présenterons à la suite sont :

Voici un tableau présente les types des (SGPS) :

<b>Locale (interne)</b>	<b>International (externe)</b>
<b>G-Pédagogie</b>	ScoDoc
<b>SEES</b>	APOGEE.

**Tableau 3 :Types des SGPS**

### **5.1. G-Pédagogie**

Le système G-Pédagogie est un logiciel de Gestion pédagogique (Multi-utilisateurs) conçu et développé par DEVLOSOFT, il permet le suivi administratif et pédagogique des étudiants des universités algériennes de nouveau système LMD dès l'obtention du bac jusqu'au l'obtention du doctorat troisième cycle.

Le système G-Pédagogie propose toutes les fonctionnalités nécessaires pour le suivi et le passage d'une année à une autre ou d'un cycle à un autre selon les lois

et les règlements du ministère de l'enseignement supérieur et de la recherche scientifique. [17]

Le système G-Pédagogie fonctionne en deux environnements:

- Environnement Mono-Utilisateurs (Poste à Poste)

Dans cet environnement le logiciel est installé avec la base de données sur un seul ordinateur. Toutes les opérations pédagogiques seront exécutées sur cet ordinateur (saisie des notes, délibérations, impression des PV, Impression des relevés des notes ...etc.)

- Environnement Multi-Utilisateurs (Client / Serveur)

Dans cet environnement la base de données est installée sur un serveur dédié situé à l'administration de la faculté au le Rectorat (inaccessible sauf par des administrateurs), et le logiciel est installé dans chaque poste utilisateur bien évidemment avec des identifiants et des privilèges, toutes les opérations pédagogiques seront exécutées parallèlement sur les ordinateurs du réseau liées est l'accès à la base des données est gérer par le serveur. [18]

La figure 10 présente l'interface de système G-Pédagogie



Figure 10: interface de système G-Pédagogie

## 5.2. SEES

SEES (Suivi des Etudiants et Enseignement Supérieurs) est un système de gestion pédagogique avec licence exploité au sein de notre université (SALHI AHMED Naama ) et d'autres universités algériennes, on peut dire que c'est une application mono poste conçu pour traiter uniquement les évaluations standards (examen, interrogation TD et contrôle TP).

SEES offre les fonctionnalités suivantes :

- La saisie des notes
- Traitement et calcul des moyens : module, unité, semestre et annuel.
- Traiter la compensation entre les semestres pour les formations de licence
- Editer des états de sorties notamment les PVs de délibérations. [19]

### Les limites

- Est une solution mono poste.
- Ne possède pas d'un espace dédié aux enseignants ou les étudiants.
- Ne suivent pas l'assiduité
- Comporte des limites et des lacunes pour les traitements des évaluations
- N'assure pas la gestion d'emploi du temps.
- Ayant une interface non conviviale

La figure 11 présente l'interface de système SEES



Figure 11: L'interface de logiciel SEES.

### 5.3. SCODOC :

ScoDoc est un logiciel libre développée par Emmanuel Viennet (Université Paris 13) pour le suivi de la scolarité des étudiants d'université : suivie précis des étudiants (assiduité, notes). Son objective est de faciliter le travail des enseignants et personnels (facilité de saisie des informations, éditions de bordereaux et de bulletins simplifiée).

ScoDoc est un logiciel purement web: tous les accès se font via un navigateur web standard, à travers l'internet (connexions sécurisées). [20]

La figure 12 présente l'interface de système SCODOC

**Figure 12: L'interface de SCODOC**

ScoDoc offre les fonctionnalités suivantes :

- Suivi de l'assiduité (saisie et visualisation des absences)
- Gestion des notes
- Fichiers étudiant (coordonnées, annotations, suivi du parcours individuel)
- Suivi des relations avec les entreprises (stages, projets)
- Analyses et statistiques (rapports d'activité, etc.)

## Les limites de Scodoc :

- Il ne respecte pas les lois réglementant les universités algériennes
- Ne gère pas l'emploi du temps

## 5.4. APOGEE

APOGEE est un logiciel étranger élaboré par l'AMUE (Agence de Modernisation des Universités et Etablissements) qui est utilisé dans environ 90 Universités (ou établissements).

Son objectif principal est d'assurer la gestion du dossier étudiant qui va de l'inscription administrative jusqu'aux notes en passant par les stages et les thèses. [21]

APOGEE offre les fonctionnalités suivantes :

- Accueil et renseignement des étudiants et stagiaires ;
- Suivi de l'inscription des étudiants et stagiaires ;
- Organisation des examens et jurys ;
- Gestion spécifique de la formation continue
- La figure 13 présente l'interface de l'APOGEE

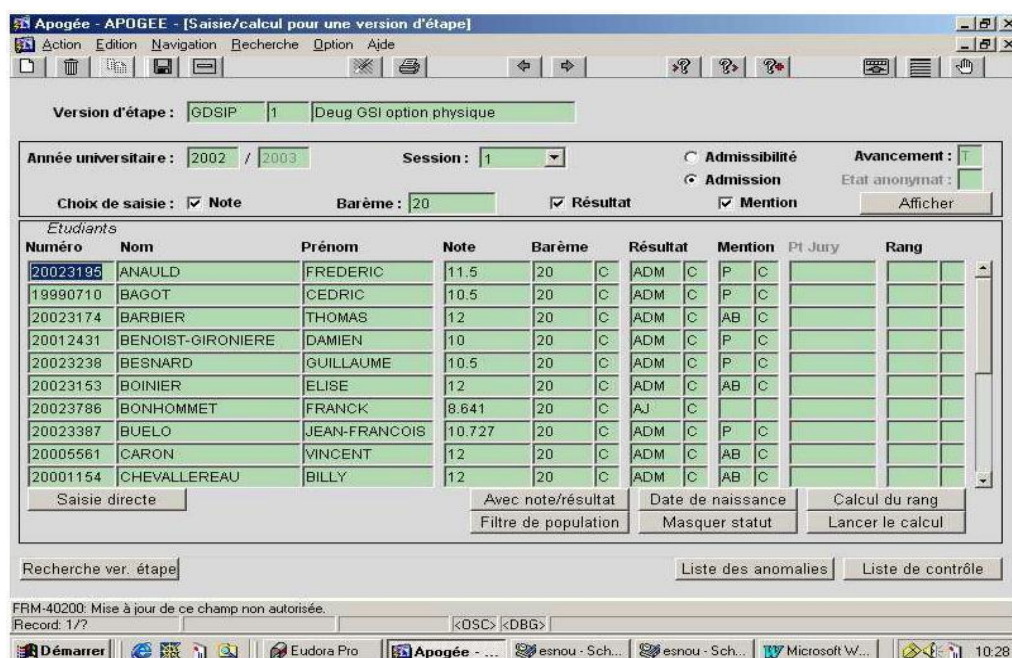


Figure 13: L'interface de l'APOGEE

### **Les limites :**

- Aussi APOGEE comme ScoDoc ne respecte pas les lois réglementant les universités algériennes.
- Ne gère pas l'emploi du temps
- Les notes ayant une saisie unique sur le logiciel qu'empêchent les erreurs de la saisie.
- Ne suivre pas l'assiduité
- Difficulté à calculer la charge d'enseignement.
- Ne prend pas en considération la charge de la patrouille de l'étudiant.

Après de cette étude analytique pour chaque système : G-pédagogie, SEES, ScoDoc et APOGEE, nous concluons qu'il n'y a pas un système parfait et complet pour assurer tous les besoins pédagogiques.

## **6. Présentation du projet**

Notre mission dans ce travail est d'élaborer un système open source fiable, cohérent, convivial, sécurisé et facile à utiliser à concevoir et à développer des modules sous Odoo ERP, pour la gestion pédagogique des universités.

Toutefois, ce système est adaptable à l'environnement algérien que ce soit les règlements internes qui régissent la gestion des universités ou bien les conditions de travail.

L'objectif principal est la gestion des étudiants, des enseignants des universités.

### **6.1. Sous objectifs**

Pour atteindre cet objectif cité ci-dessus, nous le détaillerons à des sous objectifs pour préciser le travail demandé et planifier les étapes à respecter et à suivre pendant la conception et l'implémentation de ce système.

Ces sous objectifs sont les suivants :

- Gestion des évaluations :

Création des évaluations, saisie des notes localement ou à distance, établir les PVs de délibérations, publication des bulletins, consultation des résultats à distance par les étudiants ; tout cela représente des opérations les plus importantes réalisées par cet objectif

- Suivi d'assiduité :

L'automatisation d'enregistrement de présence et l'absence des étudiants ont un impact significatif pour donner une crédibilité au processus de la gestion pédagogique.

- Gestion d'emploi du temps

La mise en place d'un processus de gestion d'emploi du temps, fait un objectif très important pour l'optimisation des ressources humaines et matérielles au sein de l'université d'un côté, et faciliter le travail pour le staff administratif, les enseignants et les étudiants à l'autre coté.

### **6.2. Utilités du notre système**

- Pour l'administration
  - La gestion des utilisateurs
  - La gestion des notes et des bulletins scolaires
  - les absences
- Pour l'enseignant
  - L'emploi de temps
  - la gestion des notes et des absences
  - des outils de communication et travail collaboratif
- Pour l'étudiant
  - L'emploi du temps
  - la consultation de ses notes et de ses absences
  - des outils de travail de groupe
  - l'accès des ressources numériques...

### 6.3. Motivation :

Après une petite analyse rapide et simple aux outils utilisés par les différents services pédagogiques des universités algériennes au niveau national, nous constatons que ces derniers sont classifiés sur trois parties :

- Des universités n'ont aucun outil informatique pour la gestion de leurs services de scolarité notamment les opérations pédagogiques.
- Des universités utilisent des logiciels libres ou bien avec licence chers et importés de l'étranger ne sont pas compatibles parfaitement avec le contexte interne.
- D'autres universités développent des logiciels localement ne sont pas généralisés et ayant d'un nombre important des lacunes techniques.

Donc, la problématique est d'avoir un système purement algérien fiable et sécurisé pour la gestion des services de pédagogique, ce dernier est répondu aux contraintes imposées par les règlements et les lois qui gèrent les universités algériennes

- ce système au minimum assure le bon déroulement des tâches suivantes :
- gestion des évaluations (examen fin de semestre, évaluation continue : les interrogations TD, contrôle TP et les exposés...etc) à partir de la saisie des notes et les délibérations jusqu'à l'affichage des résultats finals.
- suivi d'assiduité pour tous les séances pédagogiques (cours, TD et TP) surtout le calcul de nombre des absences autorisés et justifiés et l'extraction des étudiants exclus des modules.
- la gestion d'emploi du temps affin
  - d'éviter les chevauchements entre les salles / les amphis, les enseignants ou les groupes et les sections des étudiants) .
  - d'assurer le partage de charge hebdomadaire ou quotidienne pour

les enseignants et les étudiants.

- et optimiser les ressources physiques disponibles.

### **7. Conclusion**

On espère qu'à la fin de ce chapitre, on a bien clarifiés le sujet de travail par la détermination et la sélection des objectifs souhaités pour palier la problématique présenté ci-dessus.

Et afin que nous puissions atteindre ces objectifs, nous détaillerons dans le chapitre suivant les différentes étapes suivies (conception et implémentation) pour la mise en place de notre système.

---

## **CHAPITRE 3**

---

# **CONCEPTION ET IMPLEMENTATION**

## Introduction

Dans cette section, nous allons répondre aux différentes exigences citées auparavant sous la forme de diagramme UML.

UML (Unified Modeling Language) est un langage de modélisation graphique à base de pictogramme. Il est paru dans le univers de génie logiciel, dans le cadre de la conception orienté objet. Or le langage UML est une synthèse de tous les concepts et formalisme et méthodologiques les plus utilisés, pouvant être utilisés, grâce à sa simplicité et à son globalité comme langage de modélisation pour la plus part des systèmes devant être développés[22].

### 1. Les diagrammes UML [23]:

Dans cette phase de conception nous sommes intéressés aux digrammes de conception suivants :

### 2. Le diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet



Figure 14 :Diagramme de cas d'utilisation<<Administrateur>>

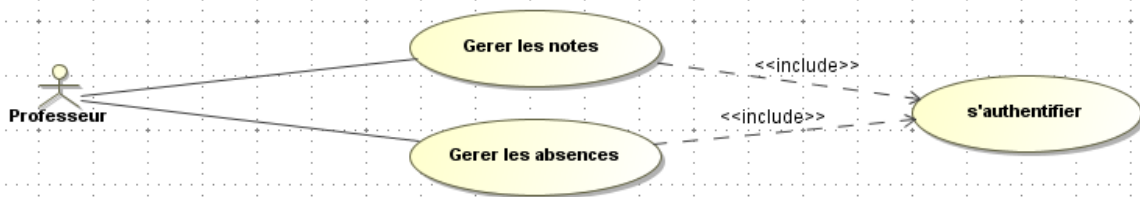


Figure 15 :Diagramme de cas d'utilisation<<Professeur>>

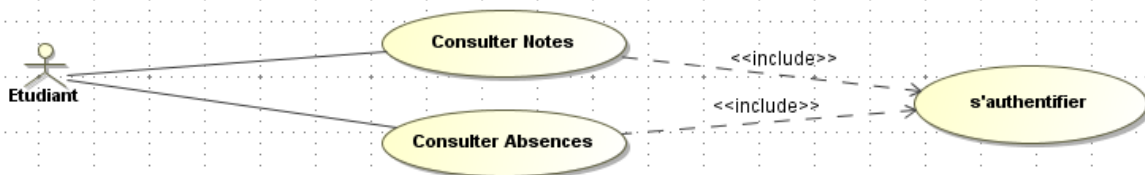


Figure 16 : Diagramme de cas d'utilisation<<Etudiant>>

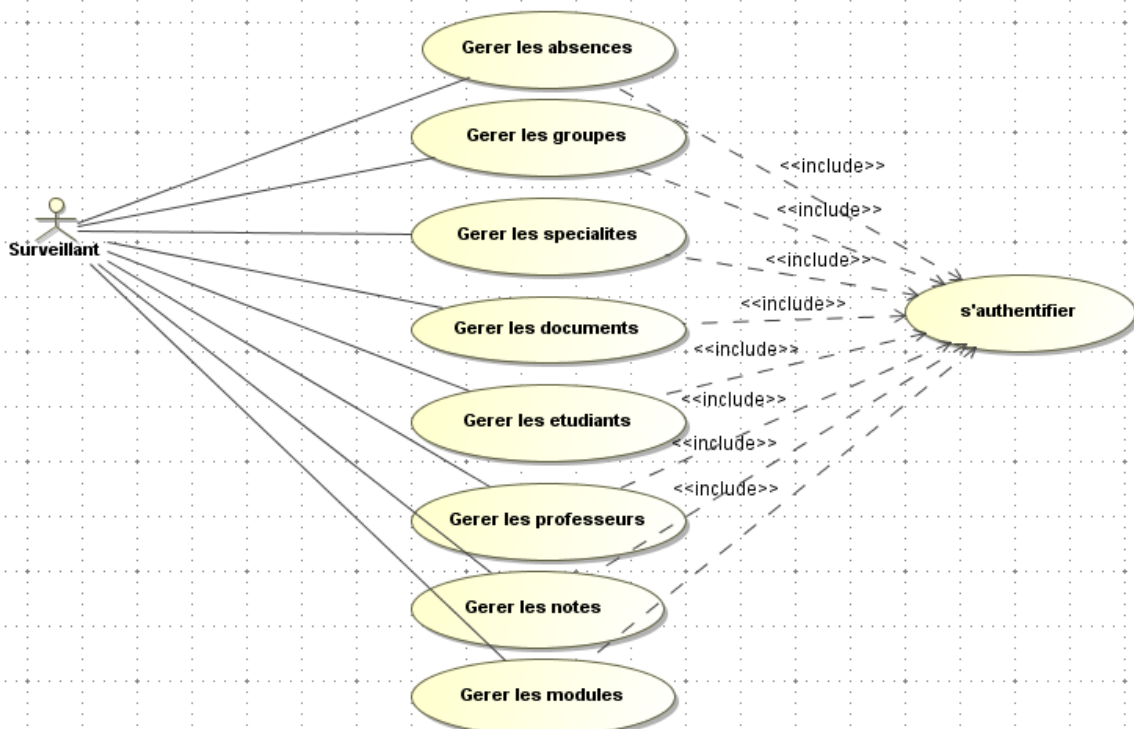


Figure 17 :Diagramme de cas d'utilisation<<Surveillant>>

### 3. Diagramme de Séquence

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Ainsi, contrairement au diagramme de communication, le temps

y est représenté explicitement par une dimension (la dimension verticale) et s'écoule de haut en bas.

Voici quelques diagrammes de séquences du système :

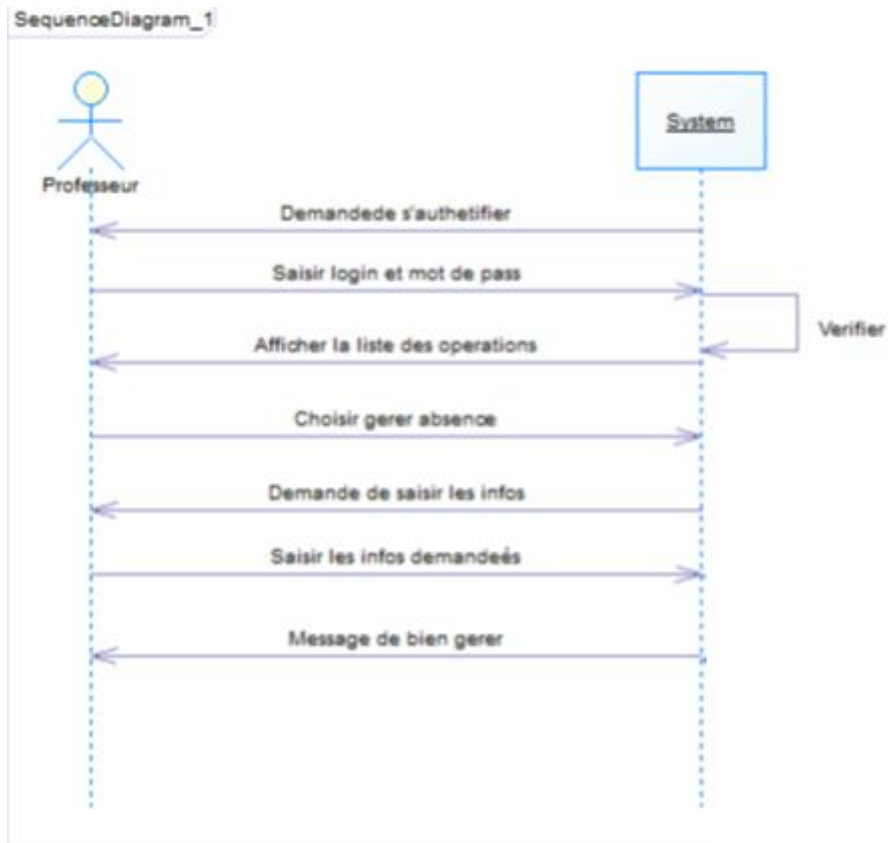


Figure 18 :Diagramme de Sequence<<Ajouter Absence>>

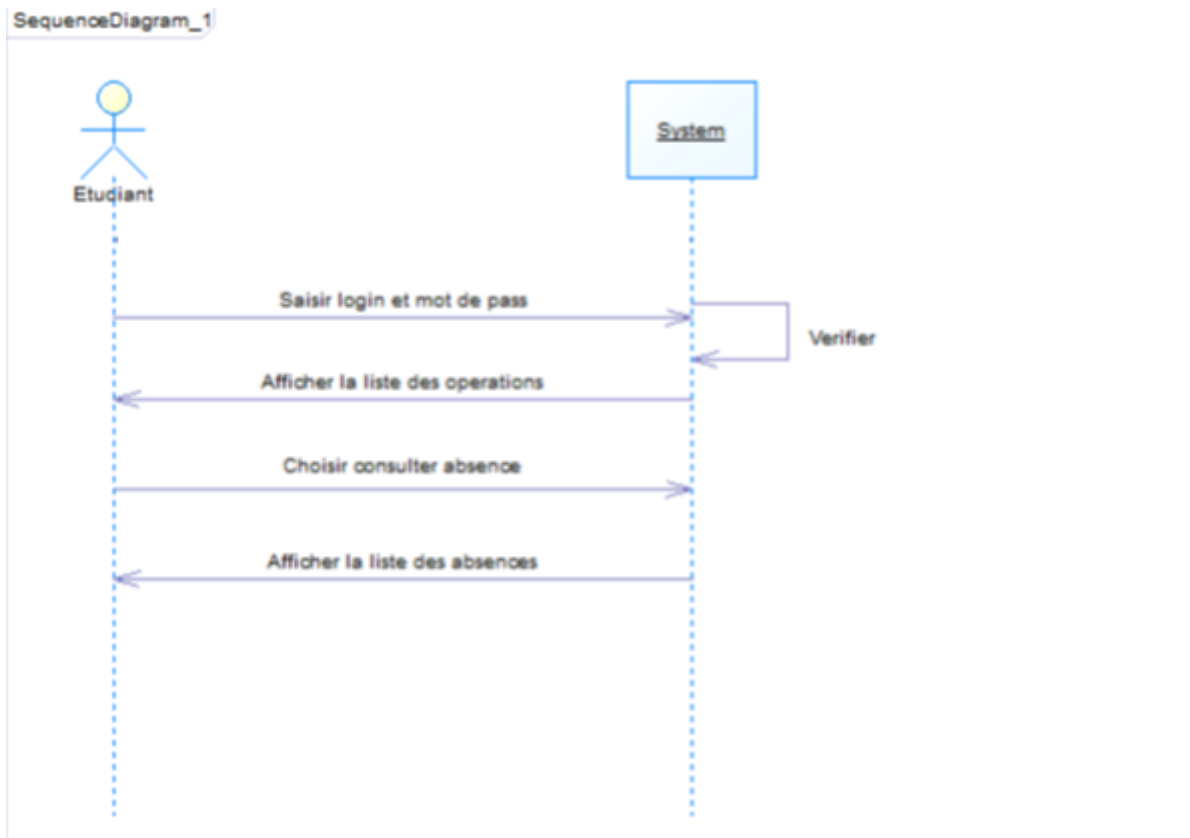


Figure 19 : Diagramme de Séquence<<Consulter Absence>>



Figure 20 : Diagramme de Séquence<<Affecter un module a un Professeur>>

## 4. Diagramme de classes

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

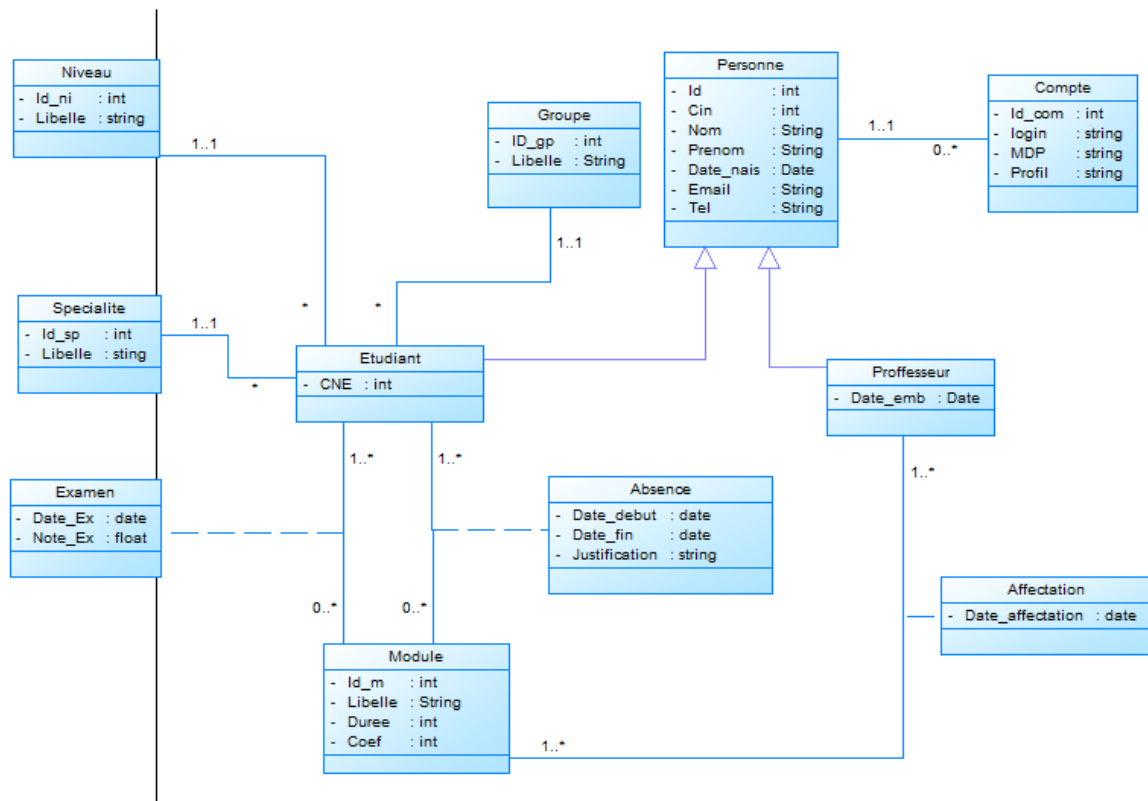


Figure 21 : Diagramme de classe

## 5. Diagramme d'état transition<Matière>

Les diagrammes d'états-transitions permettent de décrire les changements d'états d'un objet ou d'un composant, en réponse aux interactions avec d'autres objets/composants ou avec des acteurs.

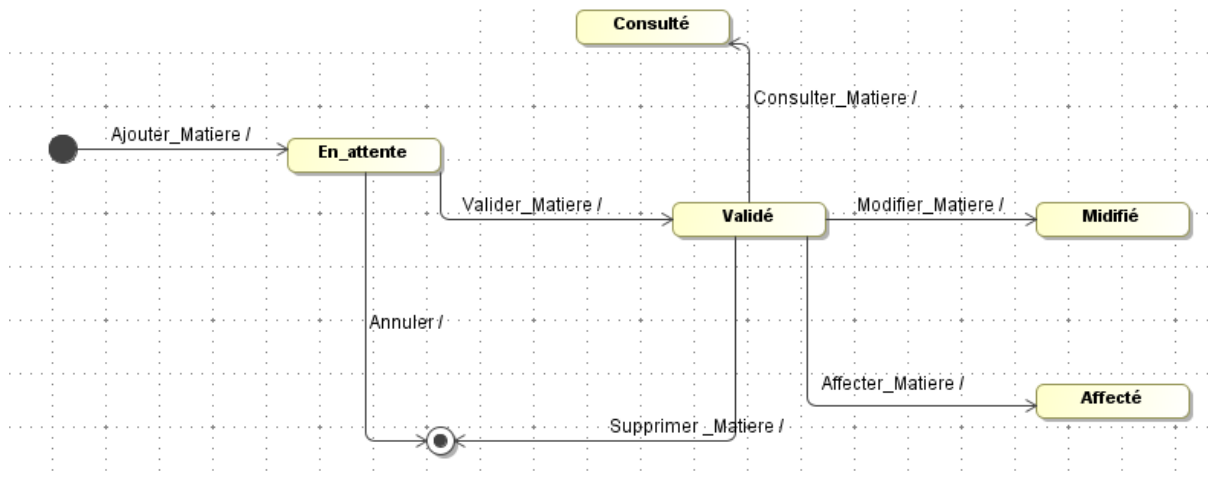


Figure 22 :Diagramme d'état transition&lt;&lt;Matière&gt;&gt;

## 6. Partie Implémentation

Après avoir terminé la partie conceptuelle, on va décrire maintenant la façon d'implémenter mon module de gestion de pédagogie dédié à l'université appelé SGPU ; Système de Gestion Pédagogique Universitaire ;

### 6.1. Les outils utilisés

- 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'un type BSD. « *La licence BSD (Berkeley Software Distribution License) est une licence libre utilisée pour la distribution de logiciels. Elle permet de réutiliser tout ou une partie du logiciel sans restriction, qu'il soit intégré dans un logiciel libre ou propriétaire.* »

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).

- 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 gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Le langage Python est placé sous une licence libre proche de la licence BSD et fonctionne sur la plupart des architectures.

### • XML (eXtensible Markup Language)

L'Extensible Markup Language « langage de balisage extensible » est un métalangage informatique de balisage générique. Sa syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, Elle est reconnaissable par son usage des chevrons (<, >) encadrant les noms des balises. L'objectif initial de XML est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes (interopérabilité). Avec ses outils et langages associés, une application XML respecte généralement certains principes :

- la structure d'un document XML est définie et validable par un schéma ;
- un document XML est entièrement transformable dans un autre document XML.

## 6.2. Présentation de module

Structure d'un module

Un module est un package python qui peut contenir les éléments suivants ;

- Objets métier : classes python
- Fichiers de données (xml ou json): métadonnées pour les vues, workflows et fichiers de configuration
- Contrôleurs web:
- Fichiers statiques: images, css, javascripts
- Un module est un dossier stocké dans un dossier contenant les modules :
- Fichier obligatoires

## CHAPITRE 3

---

- Manifest: `__manifest__.py`
- `__init__.py`

Il y a principalement quatre fichiers requis pour créer un nouveau module (`.py` ou `.xml` peuvent être exclus selon le besoin). Les quatre principaux fichiers sont,

```
* __init__.py
* __manifest__.py
* model.py
* view.xml
```

Dans le `__manifest__.py`, nous devons mentionner le nom du module, le nom de l'auteur, la version, la description, l'entreprise, la catégorie, etc. Voyons maintenant à quoi servent tous ces paramètres,

```
* Name - Nom du module à deployer
* Author - Nom de l'auteur du module
* Version - version du module (1.0 p. ex.)
* company - La société qui développe ce module
* website - Adresse du site
* Category - Catégorie du module
* depends - Les dépendances
* data - Les fichiers xml du module
```

Appliquons maintenant, ces modèles sur notre Module, nous retrouvant les captures suivants ;













 <b>controllers</b>	03/06/2018 00:35	Dossier de fichiers	
 <b>demo</b>	06/06/2018 13:41	Dossier de fichiers	
 <b>menu</b>	06/06/2018 19:04	Dossier de fichiers	
 <b>models</b>	06/06/2018 15:03	Dossier de fichiers	
 <b>report</b>	06/06/2018 13:41	Dossier de fichiers	
 <b>security</b>	06/06/2018 13:41	Dossier de fichiers	
 <b>static</b>	26/02/2018 22:02	Dossier de fichiers	
 <b>views</b>	06/06/2018 10:56	Dossier de fichiers	
 <b>wizard</b>	06/06/2018 15:03	Dossier de fichiers	
 <b>__init__.py</b>	26/02/2018 22:09	Fichier PY	1 Ko
 <b>__manifest__.py</b>	06/06/2018 02:31	Fichier PY	2 Ko
 <b>README.rst</b>	26/02/2018 22:06	Fichier RST	1 Ko

Figure 23: Les dossiers et les fichiers du module : `SGPU_PRINCIPAL`

```

1  █
2  'name': 'SGPU Principal',
3  'version': '1.0.0.0',
4  'license': 'LGPL-3',
5  'category': 'Education',
6  "sequence": 1,
7  'summary': 'Gestion des étudiants , Enseignants , ...',
8  'author': 'Universite NAAMA',
9  'depends': ['board', 'document', 'hr', 'web', 'website'],
10 █ 'data': [
11     'security/op_security.xml',
12     'security/ir.model.access.csv',
13     'views/etudiant_view.xml',
14     'views/groupe_view.xml',
15     'views/filiere_view.xml',
16     'views/domaine_view.xml',
17     'views/module_view.xml',
18     'views/enseignant_view.xml',
19     'views/sgpu_template.xml',
20     'menu/sgpu_principal_menu.xml',
21     'menu/enseignant_menu.xml',
22     'menu/etudiant_menu.xml',
23 ],
24
25 'css': ['static/src/css/base.css'],
26 'js': [],
27 █ 'images': [
28     'static/description/sgpu_principal_banner.jpg',
29 ],
30 'installable': True,
31 'auto_install': False,
32 'application': True,
33 █
34

```

Figure 24 : Contenu du fichier : `__manifest__.py`  
du module : `SGPU_PRINCIPAL`

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  █<odoo>
3  █  <data>
4  █    <!-- sgpu Menu -->
5  █    <menuitem id="menu_racine" name="SGPU"
6  █      sequence="1" />
7  █
8  █
9  █    <menuitem id="menu_principal" name="Etudiants" parent="menu_racine"
10 █      sequence="10" action="act_ouvrir_etudiant"
11 █      groups="group_enseignant" />
12 █
13 █
14 █    <menuitem id="menu_principal_enseignant" name="Enseignants"
15 █      parent="menu_racine" sequence="20" action="action_ouvrir_enseignant"
16 █      groups="sgpu_principal.group_enseignant" />
17 █
18 █
19 █    <menuitem id="menu_principal_etudiant" name="etudiant"
20 █      parent="menu_principal" sequence="40"
21 █      action="action_ouvrir_etudiant" />
22 █
23 █    <menuitem id="menu_principal_filiere" name="Gestion des specialites" parent="menu_principal" />
24 █  .
25 █  .
26 █  .
27 █</data>
28 █</odoo>
29 █

```

Figure 25 : Contenu : `sgpu_principal_menu.xml` / module :  
`SGPU_PRINCIPAL`

```

1 from odoo import models, fields, api, _
2 from odoo.exceptions import ValidationError
3
4 # Version 4 jour ramdane
5 class OpEtudiant(models.Model):
6     _name = 'op.etudiant.Filiere'
7     _description = 'Details formation des etudiants'
8     etudiant_id = fields.Many2one('op.etudiant', 'Etudiant', ondelete="cascade")
9     Filiere_id = fields.Many2one('op.Filiere', 'Filiere', required=True)
10    domaine_id = fields.Many2one('op.domaine', 'Domaine', required=True)
11    numero = fields.Char('Numero dans la table')
12    subject_ids = fields.Many2many('op.subject', string='Module')
13
14    _sql_constraints = [
15        ('unique_nom_numero',
16         'unique(numero,Filiere_id,domaine_id,etudiant_id)',
17         'N etudiant doit être unique !'),
18    ]
19
20
21 class Opetudiant(models.Model):
22     _nom = 'op.etudiant'
23     NomF = fields.Char('Nom de famille', size=128)
24     date_naissance = fields.Date('Date de naissance')
25     groupSang = fields.Selection(
26         [('A+', 'A+ve'), ('B+', 'B+ve'), ('O+', 'O+ve'), ('AB+', 'AB+ve'),
27          ('A-', 'A-ve'), ('B-', 'B-ve'), ('O-', 'O-ve'), ('AB-', 'AB-ve')],
28         'Groupe sanguine')
29     genre = fields.Selection(
30         [('m', 'Masculin'), ('f', 'F&eacuteminin')], 'Genre')
31     id_numero = fields.Char('Numero Etudiant', size=64)
32     groupe_id = fields.Many2one('op.groupe', 'Groupe')
33
34     @api.multi
35     @api.constrains('date_naissance')
36     def _check_birthdate(self):
37         for record in self:
38             if record.date_naissance > fields.Date.today():
39                 raise ValidationError(_('
40                     "date de naissance erronee!")
41

```

Figure 26 : Contenu du fichier : etudiant.py / module : SGPU\_PRINCIPAL

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <odoo>
3     <data>
4
5         <record id="vue_filiere_tree" model="ir.ui.view">
6             <field name="nom">op.filiere.tree</field>
7             <field nom="model">op.filiere</field>
8             <field nom="priorite" eval="8" />
9             <field nom="arch" type="xml">
10                <tree string="Filiere">
11                    <field nom="nom" />
12                    <field nom="code" />
13                    <field nom="section" />
14                    <field nom="evaluation_type" />
15                    <field nom="parent_id" />
16                    <field nom="module_ids" />
17                </tree>
18            </field>
19        </record>
20

```

```

21 <record id="vue_filiere_form" model="ir.ui.view">
22   <field nom="nom">op.filiere.form</field>
23   <field nom="model">op.filiere</field>
24   <field nom="priorite" eval="8" />
25   <field nom="arch" type="xml">
26     <form string="Filiere">
27       <sheet>
28         <group colspan="4" col="4">
29           <field nom="nom" />
30           <field nom="code" />
31           <field nom="section" />
32           <field nom="evaluation_type" />
33           <field nom="parent_id" />
34           <field nom="min_unite"/>
35           <field nom="max_unite"/>
36         </group>
37         <group string="Modules" nom="module">
38           <field nom="module_ids" nolabel="1"/>
39         </group>
40       </sheet>
41     </form>
42   </field>
43 </record>

```

Figure 27; Contenu du fichier (vue) : `filiere_vue.xml` / module : `SGPU_PRINCIPAL`

Passant maintenant à revoir quelques captures des modules exécutés :

Figure 28: Capture du module SGPU/ partie Etudiant

Créer : Filiere x

---

Nom	<input type="text"/>	Code	<input type="text"/>
Section	<input type="text"/>	Type d evaluation	Normal ▼
Filiere Parent	<input type="text"/>	Minimum Unite	0,00
Maximum Unite	0,00		

**Modules**

Nom de Module	Code de Module	Type de Module	Nature de Module	Filiere	Coefficient
Ajouter un élément					

Sauvegarder Annuler

Figure 29 : Capture du module SGPU/ partie Filière et Modules

Feuilles de presence / Nouveau

Sauvegarder Annuler

Rigistre	<input type="text"/>	Name	<input type="text"/>
Session	<input type="text"/>	Date	07/06/2018 ▼

**Etudiants**

Etudiant	Present ?	Remarque
Ajouter un élément		

Total Present	0	Total Absent	0
---------------	---	--------------	---

Figure 30 : Capture du module Gestion assiduité / partie Registre feuille de présence

## 7. Conclusion

Nous avons présenté dans ce chapitre la structure d'un module Odoo, les flux de travail qui existent dans notre module. Et terminant par montrer quelques captures d'écrans de notre module ;

---

## **CONCLUSION GENERALE**

---

## Conclusion générale et perspectives

Ce présent travail, s'intégrant dans le cadre du projet de fin d'études au sein du centre universitaire de Naâma, avait pour objectif de proposer une solution de gestion pédagogique universitaire .

Ce travail nous a été très instructif de point de vue des connaissances acquises, il nous a donné l'occasion d'avoir une idée approfondie sur le développement sous Odoo ERP .

Au début nous avons commencé par comprendre le contexte général de notre application et identifier les différentes exigences de ce système qui se base sur le développement d'une application sous l'ERP libre ODOO, c'est un nouveau environnement de développement auquel nous avons travaillé et réalisé un module pour la gestion pédagogique.

Nous avons présenté l'essentiel des notions et concepts de la littérature liés aux systèmes de gestion pédagogiques pour le but d'éviter le confondre avec les autres systèmes et les plateformes informatiques. Afin de nous citer une liste non exhaustive des projets similaires ou prototypes locaux et étrangers pour extraire ses limites et les éviter dans notre solution. Qui nous a permis d'identifier les différentes fonctions de notre futur module comme nous avons montré dans ce mémoire et tous qui concernent la gestion pédagogique et la conception du module et son développement pour rendre les tâches plus faciles.

## Conclusion générale et perspectives

---

Notre système SGPU (Système de gestion pédagogique universitaire) se compose de l'espace principal s suivant :

- Espace administrateur conçu essentiellement pour la mise à jour des différents acteurs (enseignant, étudiant) , création des formations, association étudiant-module ou enseignant-module et la surveillance de toute le système.
- Espace pour le service de scolarité pour la saisie des notes et l'assiduité.
- Espace pour l'enseignant pour définir ses évaluations des examens de fin semestre, interrogations TDs ou les contrôles TPs. Aussi il peut consulter ses modules, les notes de ses étudiants et son emploi du temps.
- Espace pour l'étudiant qui le permet de consulter ses notes de différentes évaluations et son emploi du temps, en plus le suivi de son assiduité et son influence sur son résultat final.

En fin, pour finaliser ce système nous espérons de réaliser et implémenter les fonctionnalités suivants qui représentent nos perspectives :

- Validation des délibérations et publication des résultats finales des examens
- Gestion d'emploi du temps
- Insérer un module d'importation : Importer la liste des étudiants ou des enseignants à partir d'un fichier Excel

## Conclusion générale et perspectives

---

- Importer des formations à partir des fichiers XML.
- Assurer l'interaction avec le système SEES le plus utilisé actuellement.
- Intégration dans l'espace numérique de l'université.

Le travail que nous avons réalisé nous a procuré beaucoup de satisfactions car les principaux objectifs tracés ont pu être atteints.

# Bibliographie

[1] [https://www.odoo.com/fr\\_FR/](https://www.odoo.com/fr_FR/) Visité le 06/06/2018

[2] Stallman R. 2009, p. 20 : « Computer users should be [...] free to share software because helping other people is the basis of society ».

[3] F. pour le logiciel libre (FSF). (18 novembre 2016). Catégories de logiciels libres et non libres adresse : <https://www.gnu.org/philosophy/categories.fr.html>.

Visité le 06/06/2018

[4] [https://www.odoo.com/fr\\_FR/page/docs](https://www.odoo.com/fr_FR/page/docs) Visité le 06/06/2018

[5] Openbravo. (mars 2017). Openbravo. adresse : <http://www.openbravo.com/fr/>

Visité le 06/06/2018

[6] Dolibarr. (2014). Dolibarr. adresse : <https://www.dolibarr.fr>. Visité le 06/06/2018

[7] WebERP. (2013). Weberp. adresse : <http://www.weberp.org>. Visité le 06/06/2018

[8] ERPNext. (2014). adresse : <p://frappe.github.io/erpnext/> Visité le 06/06/2018

[9] <https://www.sap.com/index.html>. Visité le 20/06/2018

[10] <https://www.microsoft.com/fr-dz>. Visité le 01/06/2018

[11] <https://www.sage.com/fr-fr/>. Visité le 06/06/2018

[12] <https://www.oracle.com/fr/index.html> Visité le 06/06/2018

[13] <https://www.solutions-numeriques.com> Visité le 29/05/2018

[14] OpenERP. (29 Novembre 2009). Architecture technique et modulaire adresse :

<http://myopenerp.blogspot.com/2009/11/architecture-technique-et-modulaire.html>.

Visité le 29/05/2018

[15] D. Reis. (Jun 10, 2014). Improving the performance of odoo deployments.

[16] <http://www.devlosoft.com/> DevloSoft Informatique Visité le 29/05/2018

[17] Un système de gestion pédagogique pour les université algériennes ; mémoire Master université Ourgla ; Ben Azza et Djedai 2015

[19] Foire aux Questions Système SEES-LMD : source ; <https://www.mesrs.dz>

Visité le 29/05/2018

[20] Viennet, Emmanuel. ScoDoc logiciel libre pour le suivi de scolarité.

[21] Agence de mutualisation des universités et établissements (Apogée - logiciel)

[22] UML 2 et les design patterns : Analyse et conception orientées objet et développement itératif Auteur(s) : Craig Larman

[23] Cas d'utilisation, une introduction, Olivier Capuozzo Travaux de Relecture: Christine Gaubert-Macon, Valérie Emin