



Spécialité : DUT en Génie Informatique  
(GI)

# Projet de fin d'études

Sous le thème

## Gestion des PFE et des Stagiaires

*Effectué par :*

**LAHSEN AGLAGAL  
AYOUB DERDOURI**

*Encadré par :*

**Pr. GOUSKIR**

Présenté et soutenu le ....., devant le Jury composé de

Pr. ....

Président

Pr. ....

Examineur

Pr. ....

Encadrant

*Année universitaire : 2022-2022*

---

## *Dédicace*

---

Nous dédions ce modeste travail, comme preuve de respect et de reconnaissance à :

**NOS CHERS ET AIMABLES PARENTS :**

Pour les efforts qu'ils ont consentis pour notre éducation et notre formation, pour leur précieux soutien moral et matériel, pour leurs encouragements continus, et pour leurs sacrifices tout au long de notre vie, que nous serons tellement très reconnaissants.

**NOS FRERES ET SŒURS :**

D'être à nos côtés et nous encourager tout le temps.

**NOS FAMILLES :**

Qui nous a soutenus tout au long des études.

**NOS AMIS :**

Qui ont partagé avec nous une période d'étude inoubliable.

**ET A VOUS CHERS LECTEURS**

---

# Remerciements

---

Au nom d'Allah le tout puissant.

On ne peut pas laisser passer l'occasion de la présentation de ce rapport sans exprimer nos remerciements à tous ceux qui ont bien voulu apporter l'assistance nécessaire au bon déroulement de ce projet.

On tient à remercier notre encadrant le Pr. GOUSKIR , pour nous avoir encadré tout au long de cette période, aussi d'être source d'information, de communication, d'encadrement et d'orientation technique pendant toute la durée de réalisé ce projet sans hésiter à aucun moment de nous prodigués, malgré vos obligations professionnelles. Vos encouragements inlassables, votre amabilité, votre gentillesse méritent toute admiration. Nous saisissons cette occasion pour vous exprimer notre profonde gratitude tout en vous témoignant notre respect.

On remercie sincèrement tous les professeurs du département Informatique dans la personne du Pr. A.QUAFFO Chef du département informatique de la ESTBM qui fournit d'énormes efforts pour ses étudiants pour accomplir une bonne formation, dans les conditions les plus favorables.

On adresse aussi nos remerciements les plus sincères à tout le personnel administratif de l'école supérieur de technologie béni Mellal.

---

## *Introduction Générale*

---

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela à l'aide de l'informatique, qui joue un rôle important dans le développement de nombreuses entreprises et organisations.

Avant l'invention de l'ordinateur, toutes les informations étaient enregistrées manuellement sur des supports en papier. Ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières. La nouvelle logique de l'organisation du travail demande aux établissements d'éducation et d'apprentissage d'utiliser essentiellement l'information comme matière première pour pouvoir être plus efficace. Ils doivent donc intégrer le développement du système d'information dans leurs investissements stratégiques, dans la mesure où ils structurent la saisie, le stockage, l'organisation et la communication de l'information.

Aujourd'hui, l'université à laquelle nous rattacherons notre étude, constitue une partie intégrante des établissements universitaires où l'informatique pourra aider à résoudre des problèmes quotidiens de gestion. En effet, la croissance de la population estudiantine nécessite la mise en place d'une gestion efficace, rapide et rationnelle des projets de fin d'étude (PFE), or et jusqu'à ce jour, ce domaine est géré manuellement, d'où la nécessité d'introduire l'informatique afin d'automatiser cette tâche.

## Table des matières

<b>Chapitre I : Analyses des besoins et outils</b> .....	7
I. Introduction.....	8
<b>II. PROBLEMATIQUE</b> .....	8
III. Solution.....	8
<b>IV. OUTILS</b> .....	9
1. La plateforme Odoo .....	9
2. PostgreSQL .....	11
3. Python .....	11
4. XML.....	12
5. Outils de conception .....	12
<b>V. ARCHITECTURE LOGICIELLE</b> .....	13
1. La démarche MVC .....	13
2. Modules d'Odoo .....	14
3. Composition d'un module .....	14
4. Installation d'Odoo 14 Community .....	15
<b>CHAPITRE 2 : ANALYSE ET CONCEPTION</b> .....	19
I. INTRODUCTION .....	20
II. METHODOLOGIES DE DEVELOPPEMENT .....	20
III. LE LANGAGE UML .....	20
IV. DIAGRAMME DE CAS D'UTILISATION .....	21
1. Diagramme des cas d'utilisations d'administrateur.....	22
2. Diagramme des cas d'utilisations d'enseignant.....	22
3. Diagramme des cas d'utilisations d'étudiant.....	23
V. DIAGRAMME DE CLASSES .....	23
VI. LES DIAGRAMMES DE SEQUENCES.....	24
1. Diagramme de séquence « Authentification ».....	25
2. Diagramme de séquence « ajouter sujet PFE ».....	26
3. Diagramme de séquence « valider les choix d'étudiants » .....	27
<b>Chapitre III : Réalisation</b> .....	28
I. PARTIE D'ADMINISTRATEUR .....	29
1. Management des projets .....	29
2. Management des étudiants .....	30
3. Management des enseignants .....	32
4. Management des départements.....	33
• Liste des stagiaires.....	34

<b>II. PARTIE ETUDIANT .....</b>	<b>35</b>
1 Cité un projet.....	35
2 Validation de sujet.....	36
<b>III. PARTIE PROFESSEUR .....</b>	<b>36</b>
<b>IV. PROGRAMMATION : PARTIE CODE .....</b>	<b>37</b>
1. Python .....	37
2. XML.....	38
<b><i>Conclusion générale</i>.....</b>	<b>39</b>
<b><i>Webographie</i> .....</b>	<b>40</b>

---

## table des figures

---

Figure 1 : logo Odoo .....	9
Figure 2: L'historique des versions d'Odoo .....	10
Figure 3 : Architecture MVC .....	13
Figure 4 : Architecture d'un module Odoo.....	15
Figure 5 : Python installer.....	16
Figure 6 : diagramme de cas d'utilisations d'administrateur .....	22
Figure 7 : diagramme des cas d'utilisations d'enseignant .....	22
Figure 8 : diagramme des cas d'utilisations d'étudiant.....	23
Figure 9 : Diagramme de classe global .....	24
Figure 10 : diagramme de séquence « Authentification » .....	25
Figure 11: diagramme de séquence « ajouter sujet PFE » .....	26
Figure 12: diagramme de séquence « valider les choix d'étudiants » .....	27
Figure 13 : Liste des projets.....	29
Figure 14: création d'un nouveau projet.....	29
Figure 15 : détails d'un projet .....	30
Figure 16 : liste des étudiants .....	30
Figure 17 : forme de création d'un étudiant .....	31
Figure 18 : détails d'un étudiant.....	31
Figure 19 : kanban liste représente les professeurs.....	32
Figure 20: forme de création d'un professeur .....	32
Figure 21 : forme représente les détails d'un professeur .....	33
Figure 22 : forme représente les détails d'un professeur .....	33
Figure 23: liste des départements.....	33
Figure 24:forme de création d'un département .....	34
Figure 25:liste des stagiaires .....	34
Figure 26:création d'un stagiaire.....	35
Figure 27:liste des PFE partie étudiant.....	35
Figure 28:Cite un sujet .....	36
Figure 29:Validation PFE.....	36
Figure 30:l'ajoute d'un sujet .....	37
Figure 31:fonction python (l'ajoute d'un utilisateur).....	37
Figure 32:Redéfinition de la fonction crée.....	38
Figure 33:validation d'un fichier PDF .....	38
Figure 34 :Record rule .....	38
Figure 35:Création d'un group .....	38

# *Chapitre I : Analyses des besoins et outils*



# I. Introduction

Dans le cadre de la formation au sein de L'école supérieure de technologie bény Mellal les étudiants sont amenés à réaliser un Projet de Fin d'études. Dans ce contexte, nous avons réalisé notre projet au sein de ESTBM et plus particulièrement dans le département d'informatique. Le travail qui nous a été confié, était de développer une application sous Odoo 14 Community pour la gestion des Projets de Fin d'études des filières Diplôme universitaire de technologie et licence professionnelle.

## II. PROBLEMATIQUE

Nous avons pu constater, en effet, pendant notre observation au sein du département d'informatique de ESTBM que les opérations de collecte des sujets de relatifs aux projets de fin d'étude ainsi que leurs affectations aux étudiants se font de façon manuelle, ce qui engendre un certain nombre de problèmes tels que :

- Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs. Des problèmes dans la sélection des sujets de la part des étudiants le jour du choix des thèmes La distribution des sujets aux étudiants pourrait être injuste. Possibilité d'erreur dans la validation des choix des sujets de PFE.

## III. Solution

Donc, la meilleure solution pour pallier ces problèmes est l'automatisation de la collecte, la validation et l'affectation des sujets.

L'objectif principal de ce projet est de développer une application ergonomique et conviviale qui permet :

- Aux encadrants des projet de suivre et de gérer le déroulement des projets de la recherche jusqu'à la soutenance.
- Aux étudiants de communiquer avec leurs encadrants durant toute la période de PFE. Les étudiants peuvent aussi saisir les informations sur leurs projets et consulter les propositions de projets, les affectations des encadrent et le planning.
- Aux enseignants du département de proposer des projets, de retrouver la trace des projets de fin d'études qu'ils ont encadré ou dans lesquels ils ont participé en tant que membre de jury.
- Au Administrateur ou l'informaticien de gestion des projets, de gestion des étudiants et des enseignants aussi les départements et les filières .

## IV. OUTILS

### 1. La plateforme Odoo

Odoo est un progiciel de gestion d'entreprise (ERP) destiné à intégrer l'ensemble des données opérationnelles et de gestion de l'entreprise dans une base de données unique, accessible par une interface web



Figure 1 : logo Odoo

Cette base de données centrale est associée à une couche fonctionnelle très innovante qui met en relation des informations d'origines diverses et assure un déroulement efficace des processus transversaux de création de valeur ajoutée de l'entreprise

Odoo, anciennement Open ERP et Tiny ERP, est à la base un progiciel libre de gestion intégré comprenant de très nombreux modules permettant de simplifier la gestion d'entreprise dans son ensemble. Le logiciel est sous licence AGPL et est utilisé par plus de 2 millions d'utilisateurs à travers le monde.

À l'origine un ERP, le logiciel s'est vu étendre ses fonctionnalités à des applications de front office (CMS, e-Commerce, Blogs, Forums, News, Événements, Live Chat, Job offres, etc.). Il apporte les applications métier dont chacun a besoin dans l'entreprise.

Cette approche modulaire facilite l'intégration de nouvelles fonctionnalités sous la forme de modules complémentaires.

#### Principales applications logicielles front office

- Créateur de site web et système de gestion de son contenu, CMS
- Vente en ligne, E-Commerce
- Interface de point de vente (PDV)

#### Principales applications logicielles back office

- |   |                                   |
|---|-----------------------------------|
| • Gestion de relation clients (CRM & SRM) | • Gestion des ressources humaines |
| • Gestion des stocks                      | • Gestion de projets              |
| • Gestion de production                   | • Gestion des ventes              |
| • Gestion des achats                      | • Gestion de manufactures         |

## Historique et notes des sorties

Le 20 janvier 2011, Open ERP SA annonçait le lancement de la version 6.0 du logiciel, qui comprend une version à la demande (SaaS). Son approche modulaire permet aux utilisateurs de commencer avec une application, puis d'ajouter d'autres modules selon leurs besoins.

En décembre 2012, la version 7.0 d'Open ERP est lancée et peut être testée en ligne, téléchargée ou vue en version de démonstration. Mai 2014: Open ERP change de nom et devient Odoo.

Été 2014, Odoo lance la version 8. Cette version enrichit principalement le logiciel de nouvelles applications qui font d'Odoo un logiciel allant au-delà d'un ERP. Ces applications sont: Marketing (gestion d'événements, d'enquêtes de satisfactions, campagnes de mails auprès de la CRM,...), CMS (construction d'un site internet - frontend lié au backend - grâce au déplacement rapide et simple de "blocs" d'éditions), e-commerce (application pour vente en ligne),...

➤ Première version stable : 2004

➤ Version stable actuelle : 15.0

Nom du logiciel	Version	Date de lancement	Changements significatifs
Tiny ERP	1.0	Février 2005	Première publication
	2.0	Mars 2005	
	3.0	Septembre 2005	
	4.0	Décembre 2006	
OpenERP	5.0		
	6.0	Octobre 2009	Première publication sous AGPL <sup>11</sup> , premier client Web
	6.1		Client web en <a href="#">Ajax</a> , Fin du support pour le <a href="#">client riche</a> (GTK+)
	7.0	Décembre 2012	
Odoo	8.0	Septembre 2014	Support pour le CMS : construction de site internet, e-commerce, point de vente, vente et business intelligence.
	9.0	Novembre 2015	Première publication des éditions Community sous licence LGPLV3 et Enterprise sous licence propriétaire.
	10.0	Octobre 2016	
	11.0 <sup>12</sup>	Octobre 2017 <sup>13</sup>	Nouvelle plateforme Odoo.sh pour le développement
	12.0 <sup>14</sup>	Octobre 2018	
	13.0	Octobre 2019	
	14.0 <sup>15</sup>	Octobre 2020	
	15.0 <sup>16</sup>	Octobre 2021	





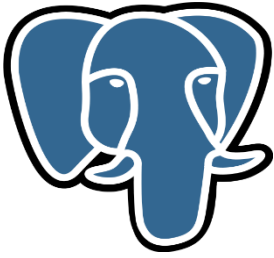
-  Anciennes versions ou fin de maintenance
-  Anciennes versions avec maintenance étendue
-  Version actuelle
-  Versions en cours de développement

Figure 2: L'historique des versions d'Odoo

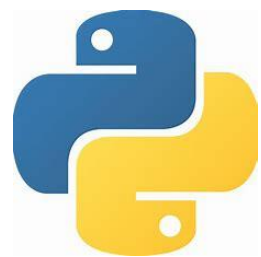
## 2. PostgreSQL



PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outillibre 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 (MariaDB, MySQL et Firebird), ou propriétaires (Oracle, Sybase, DB2, Informix et Microsoft SQL Server). 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.

## 3. Python



Python est un langage de programmation objet, multiparadigme et multiplateformes. Il favorise programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Small talk et Tcl. Le langage Python est placé sous une licence libre proche de la licence BSD2 et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix en passant par GNU/Linux, Mac OS, ou encore Android, iOS, et aussi 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 aisée aux concepts de base de la programmation

#### 4. XML



L'Extensible Markup Language (XMLnote 1, « langage à balise extensible » en français) est un langage informatique de balisage générique qui dérive du SGML. Cette 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, COMME XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (< >) encadrant les balises. L'objectif initial 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.

#### 5. Outils de conception



StarUML est une plateforme de modélisation logicielle qui prend en charge UML (Unified Modeling Language). Il est basé sur UML version 1.4 et fournit onze types de diagrammes différents, et il accepte la notation UML 2.0. Il soutient activement l'approche MDA (Model Driven Architecture) en prenant en charge le concept de profil UML. StarUML excelle dans la personnalisation de l'environnement de l'utilisateur et a une grande extensibilité dans ses fonctionnalités. L'utilisation de StarUML, l'un des meilleurs outils de modélisation logicielle, garantira la maximisation de la productivité et de la qualité de vos projets logiciels.

## V. ARCHITECTURE LOGICIELLE

La conception d'Odoo est orientée par une architecture MVC, des flux de travail flexibles, une interface-utilisateur graphique dynamique, une interface de communication interne XML-RPC, et un système personnalisable de comptes-rendus. D'un point de vue de l'architecture technique, Odoo est construit autour de trois composants principaux qui communiquent entre eux par les protocoles XML-RPC et NET-RPC

- Le serveur odoo-server qui stocke ses données dans une base PostgreSQL ;
- Le client odoo-client qui s'installe sur le poste de l'utilisateur (abandonné depuis la v7) ;
- Le serveur web odoo-web qui permet une utilisation depuis un navigateur.

### 1. La démarche MVC

Modèle d'architecture qui cherche à séparer nettement les couches de présentation (UI : User Interface), métier (BLL : Business Logic Layer) et d'accès aux données (DAL : Data Access Layer). Le but étant d'avoir une dépendance minimale entre les différentes couches de l'application, ainsi les modifications effectuées sur n'importe quelle couche de l'application n'affectent pas les autres couches.

- Modèle – Encapsule le cœur fonctionnel de l'application, le domaine logique.
- Vue – les données sont envoyées, par le modèle, à la vue qui les présente à l'utilisateur.
- Contrôleur – reçoit les données et les transmet au modèle ou à la vue. Une telle architecture est communément appelée architecture 3-tier ou à 3 niveaux.

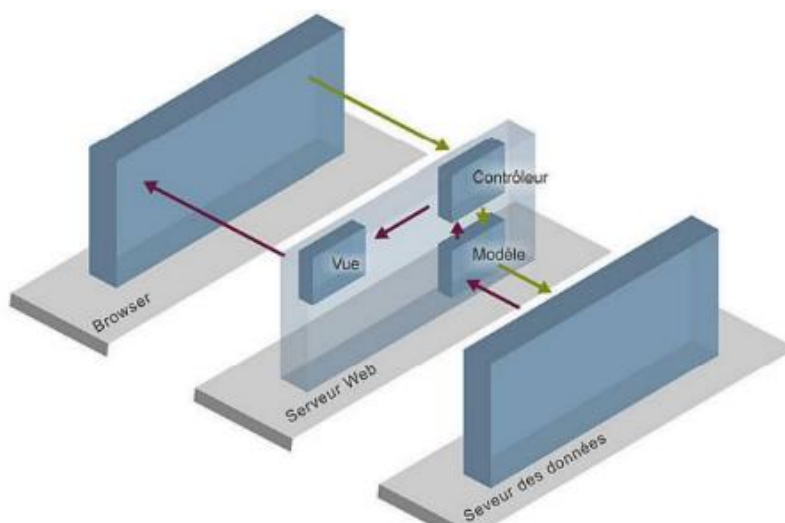


Figure 3 : Architecture MVC

## 2. Modules d'Odoo

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

## 3. Composition d'un module

Un module Odoo peut contenir plusieurs éléments :

### Business objects

Un objet métier (par exemple une facture) est déclaré en tant que classe Python. Les champs définis dans ces classes sont automatiquement mappés aux colonnes de la base de données grâce à la couche ORM.

### Vues d'objet

Définir l'affichage de l'interface utilisateur

### Fichiers de données

Fichiers XML ou CSV déclarant les données du modèle :

- Vues ou rapports,
- Données de configuration (paramétrage des modules, règles de sécurité),
- Données de démonstration

### Web controllers

Traiter les requêtes des navigateurs Web.

### Données Web statiques

Images, fichiers CSS ou JavaScript utilisés par l'interface Web ou le site Web. Aucun de ces éléments n'est obligatoire. Certains modules peuvent uniquement ajouter des fichiers de données (par exemple, une configuration comptable spécifique à un pays), tandis que d'autres peuvent uniquement ajouter des objets métier.

### Structure des modules

Chaque module est un répertoire dans un répertoire de module. Les répertoires de module sont spécifiés à l'aide de l'option `--addons-path`.

Un module Odoo est déclaré par son manifeste.

Lorsqu'un module Odoo inclut des objets métier (c'est-à-dire des fichiers Python), ils sont organisés sous la forme d'un package Python avec un fichier `__init__.py`. Ce fichier contient des instructions d'importation pour divers fichiers Python dans le module.

Voici un répertoire de module simplifié :

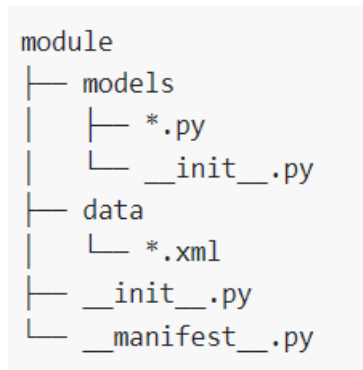


Figure 4 : Architecture d'un module Odoo

#### 4. Installation d'Odoo 14 Community

Il existe plusieurs façons d'installer Odoo, ou de ne pas l'installer du tout, selon le cas d'utilisation prévu.

- En ligne : La façon la plus simple d'utiliser Odoo en production ou de l'essayer.
- Installateurs packagés : Convient pour tester Odoo, développer des modules et peut être utilisé pour une utilisation en production à long terme avec des travaux de déploiement et de maintenance supplémentaires.
- Installation source : Offre une plus grande flexibilité : par ex. autoriser plusieurs versions d'Odoo en cours d'exécution sur le même système. Bon pour développer des modules, peut être utilisé comme base pour le déploiement en production.
- Docker : Si vous utilisez habituellement docker pour le développement ou le déploiement, une image de base officielle de docker est disponible.

##### 🔧 L'installation source

Nous avons choisi l'installation source car cela peut être plus pratique pour les développeurs de modules car la source Odoo est plus facilement accessible que d'utiliser l'installation packagée

Il rend également le démarrage et l'arrêt d'Odoo plus flexibles et explicites que les services mis en place par les installations packagées, et permet de remplacer les paramètres à l'aide de paramètres de ligne de commande

Nous vous expliquerons en détail comment installer Odoo dans les prochaines étapes.



## Git

Ce qui suit nécessite que git soit installé sur votre machine et que vous ayez une connaissance de base des commandes git.

Edition communautaire :

```
C:\> git clone https://github.com/odoo/odoo.git
```

Edition pour entreprise :

```
C:\> git clone https://github.com/odoo/enterprise.git
```

## Python

Après avoir installé la source, nous devons installer python. Odoo nécessite Python 3.6 ou version ultérieure pour fonctionner

Visitez « <https://www.python.org/downloads/windows> » pour télécharger et installer la dernière version de Python 3 sur votre machine.

NB : Lors de l'installation, cochez Ajouter Python 3 à PATH, puis cliquez sur Personnaliser l'installation et assurez-vous que pip est coché.

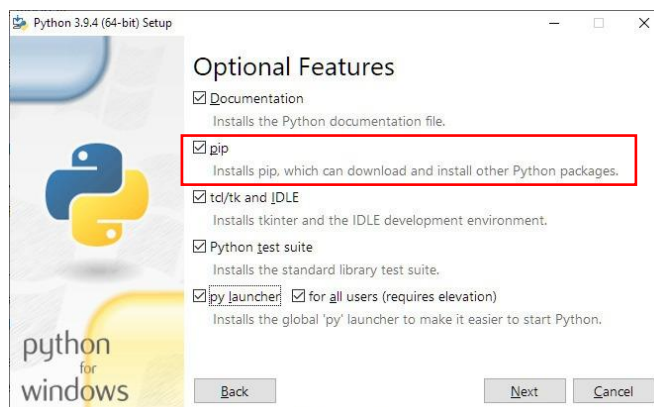


Figure 5 : Python installer

## PostgreSQL

Odoo utilise PostgreSQL comme système de gestion de base de données

Visitez « [https:// www.postgresql.org/download/windows](https://www.postgresql.org/download/windows) » pour télécharger et installer la dernière version de PostgreSQL sur votre machine.

NB: Par défaut, le seul utilisateur est PostgreSQL mais Odoo interdit la connexion en tant que PostgreSQL, vous devez donc créer un nouvel utilisateur PostgreSQL

1. Ajoutez le répertoire bin de PostgreSQL (par défaut : C:\ProgramFiles\PostgreSQL\<version>\bin) à votre PATH.
2. Créer un utilisateur PostgreSQL avec un mot de passe
3. Définir tous les privilèges sur l'utilisateur

## Dépendances

Avant d'installer les dépendances, vous devez télécharger et installer les outils de génération pour Visual Studio. Lorsque vous y êtes invité, sélectionnez les outils de génération C++ dans l'onglet Charges de travail et installez-les.

Les dépendances Odoo sont répertoriées dans le fichier requirements.txt situé à la racine du répertoire de la communauté Odoo.

Accédez au chemin de votre installation Odoo Community (CommunityPath) et exécutez pip sur le fichier requirements dans cmd ou PowerShell avec des privilèges d'administrateur :

En utilisant les commandes suivantes

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

## Exécuter Odoo

Une fois toutes les dépendances configurées, Odoo peut être lancé en exécutant `odoo-bin`, l'interface de ligne de commande du serveur. Il est situé à la racine du répertoire Odoo Community.

Pour configurer le serveur, vous pouvez spécifier des arguments de ligne de commande ou un fichier de configuration.

Les configurations nécessaires courantes sont :

- Utilisateur et mot de passe PostgreSQL. Odoo n'a pas de valeurs par défaut au-delà des valeurs par défaut de `psycopg2` : se connecte via un socket UNIX sur le port 5432 avec l'utilisateur actuel et sans mot de passe.
- Chemins d'accès personnalisés au-delà des valeurs par défaut, pour charger vos propres modules.

En utilisant les commandes suivantes

```
cd /CommunityPath
$ python3 odoo-bin --addons-path=addons -d mydb
```

Où `CommunityPath` est le chemin de l'installation d'Odoo et `mydb` est la base de données par défaut à servir sur `localhost:8069`.

## *CHAPITRE 2 : ANALYSE ET CONCEPTION*

## I. INTRODUCTION

Dans ce chapitre, nous exposerons les objectifs de notre travail, et on explique le contexte du projet et la problématique, puis nous passerons aux méthodes de conception à travers le langage de modélisation UML par la présentation des diagrammes des cas d'utilisation et de séquence et les méthodes d'analyse et de conception, afin de réaliser une implémentation physique de notre application

## II. METHODOLOGIES DE DEVELOPPEMENT

Pour développer notre application nous avons utilisé une démarche simplifiée issue du (TwoTrack Unifié Process) qui est un processus de développement itératif et incrémental basé sur le langage de modélisation UML (Unified Modeling Language). UML se veut une boîte à outils offrant des éléments de modélisation adaptés à l'approche objet. Elle permet aussi de décrire les différents aspects de notre application par une panoplie de diagrammes. Ainsi, nous décrivons notre application en trois grandes étapes :

- Analyse et spécification des besoins : dans cette phase nous modélisons le cahier de charges. Pour ce faire, nous utilisons des diagrammes de cas d'utilisations, les diagrammes de séquences et le diagramme de classes.
- Conception de l'application : dans cette phase nous décrivons l'architecture de l'application, le schéma de la base de données ainsi que les éléments conceptuels 10 que nous avons choisis pour la mise en œuvre des différentes fonctionnalités de notre système.
- Réalisation de l'application : qui consiste à mettre en œuvre les choix conceptuels effectués précédemment.

La boîte à outils qu'on a utilisée pour réaliser les différents modèles Entreprise Architect. C'est un outil d'analyse de création UML, couvrant le développement du logiciel de rassemblement d'exigences, en passant par les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien. Cet outil permet de bien schématiser notre application, pour passer de la conception vers la réalisation. Il facilite la représentation des diagrammes UML tels que le diagramme des cas d'utilisation, des séquences et des classes.

## III. LE LANGAGE UML

Langage de modélisation objet unifié et un support de communication qui facilite la compréhension et la représentation des solutions objet. Il permet de représenter des modèles et peut être intégré à n'importe quel processus de développement logiciel de manière transparente, il permet de définir et de visualiser un modèle, à l'aide des diagrammes

## IV. DIAGRAMME DE CAS D'UTILISATION

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

**Les acteurs :** Ils sont des entités externes qui interagissent avec le système, comme une personne humaine ou un robot.

**Les cas d'utilisation :** est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système.

Les acteurs de notre projet :

Acteur	Description
<b>L'étudiant</b>	C'est un individu, qui va choisir un Project puit crée leur état d'avancement
<b>Enseignant</b>	C'est acteur qui permet d'ajouter des sujets puis, le suivi d'avancement des étudiants.
<b>L'administrateur</b>	C'est celui qui assure le dynamisme du site.

## 1. Diagramme des cas d'utilisations d'administrateur

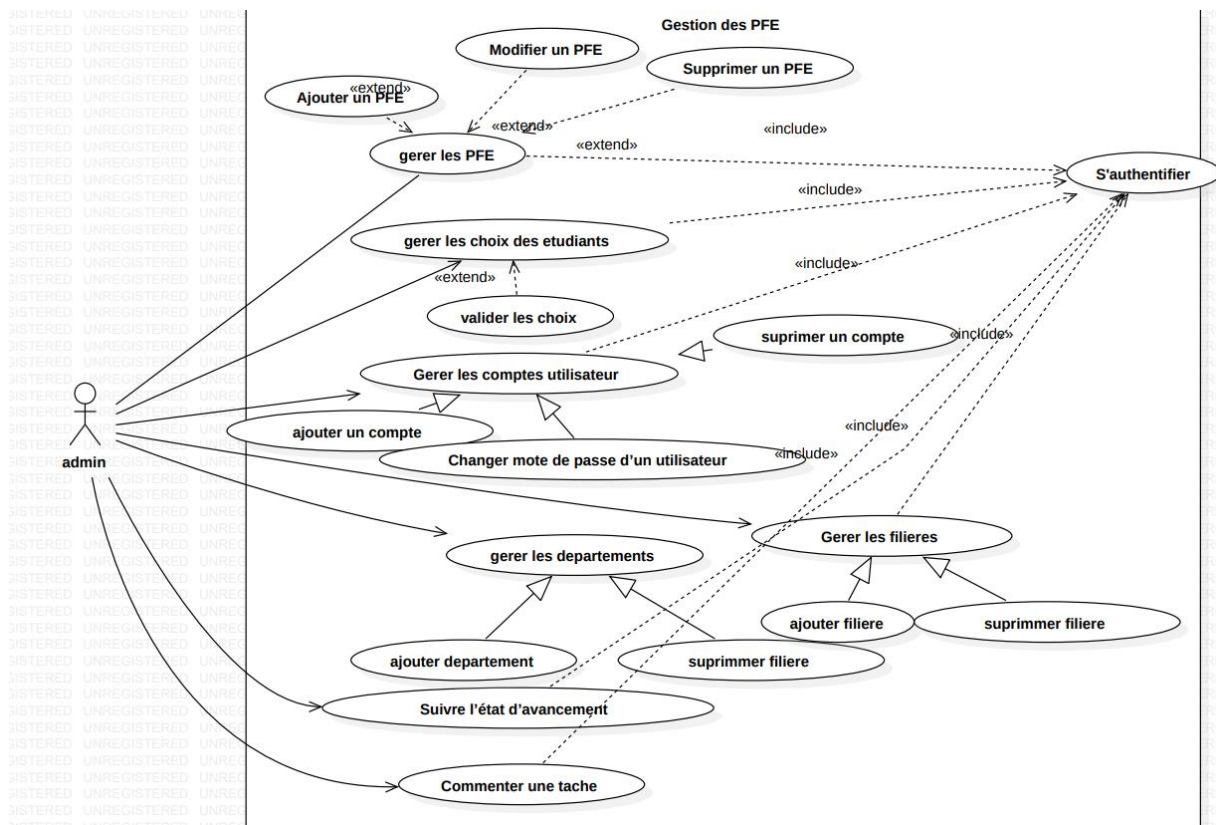


Figure 6 : diagramme de cas d'utilisations d'administrateur

Après l'authentification l'administrateur peut gérer les PFEs (ajouter modifier et supprimer), les utilisateurs soient les comptes d'étudiants ou des enseignants, aussi peut gérer les départements et les filières et finalement l'administrateur peut affecter un projet a un étudiant

## 2. Diagramme des cas d'utilisations d'enseignant

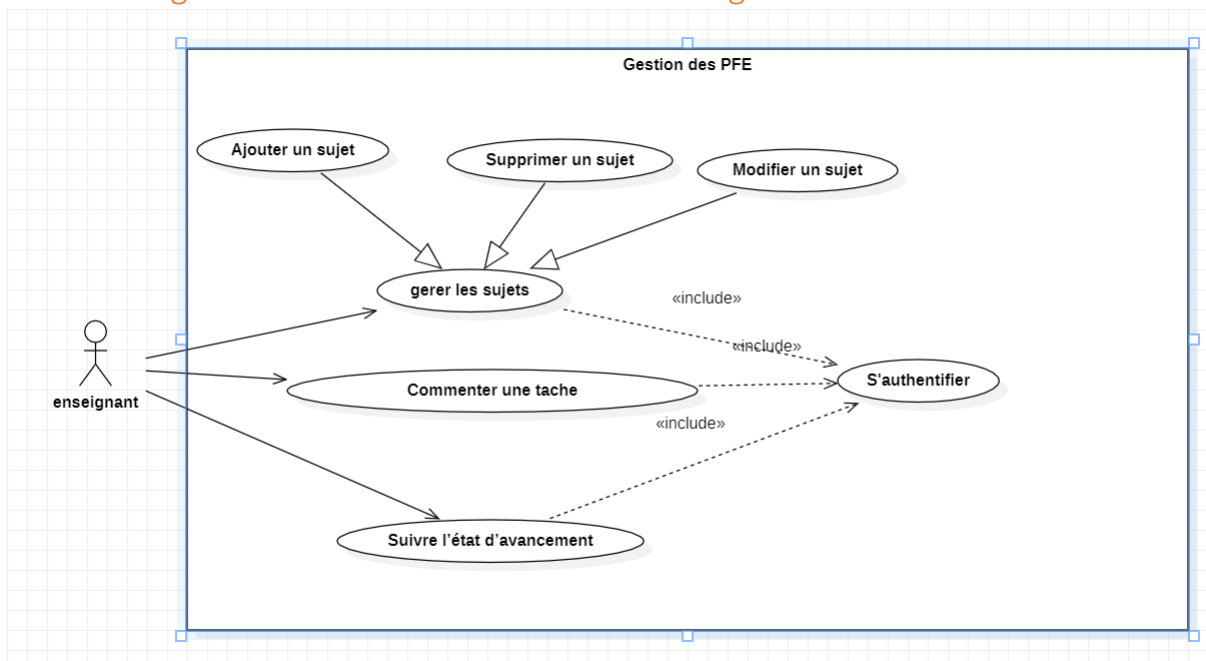


Figure 7 : diagramme des cas d'utilisations d'enseignant

Après l'authentification l'enseignant peut gérer sons PFEs (ajouter modifier et supprimer), suivre l'état d'avancement d'un projet

### 3. Diagramme des cas d'utilisations d'étudiant

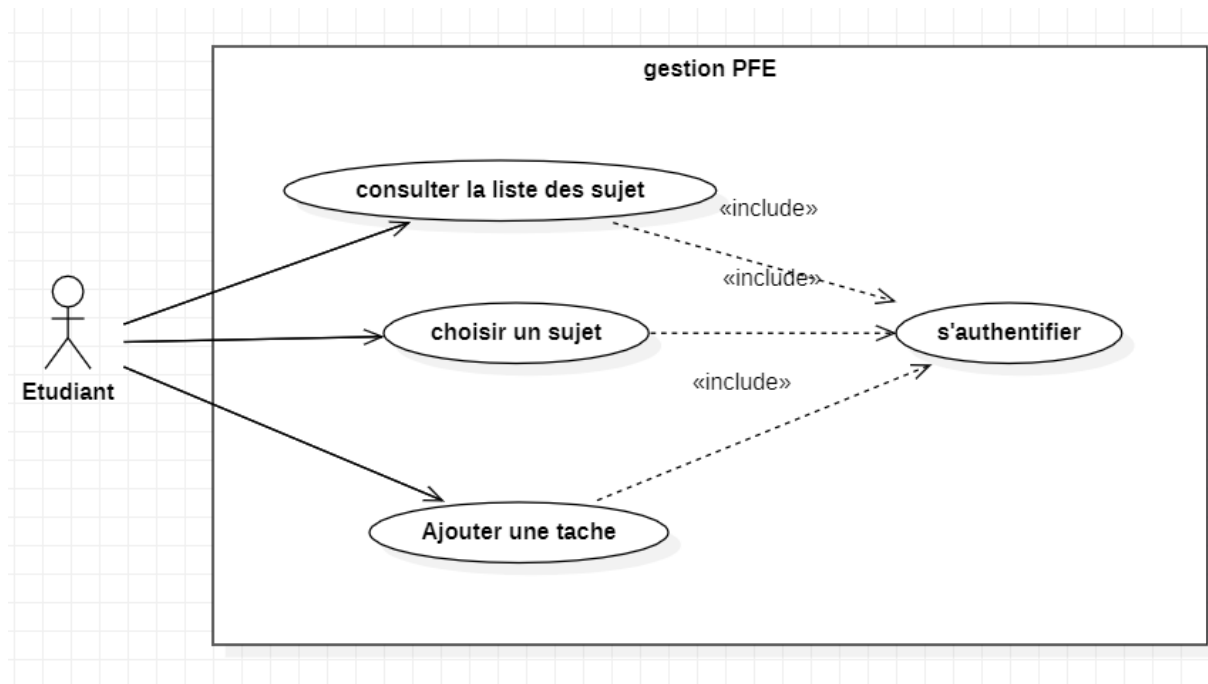


Figure 8 : diagramme des cas d'utilisations d'étudiant

Après l'authentification l'étudiant peut consulter la liste des PFEs , choisir un ou trois projets , mettre la priorité de chaque choix , gérer l'état d'avancement de sons projet .

## V. 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.

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

Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation orienté objet donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.



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

## 1. Diagramme de classe global

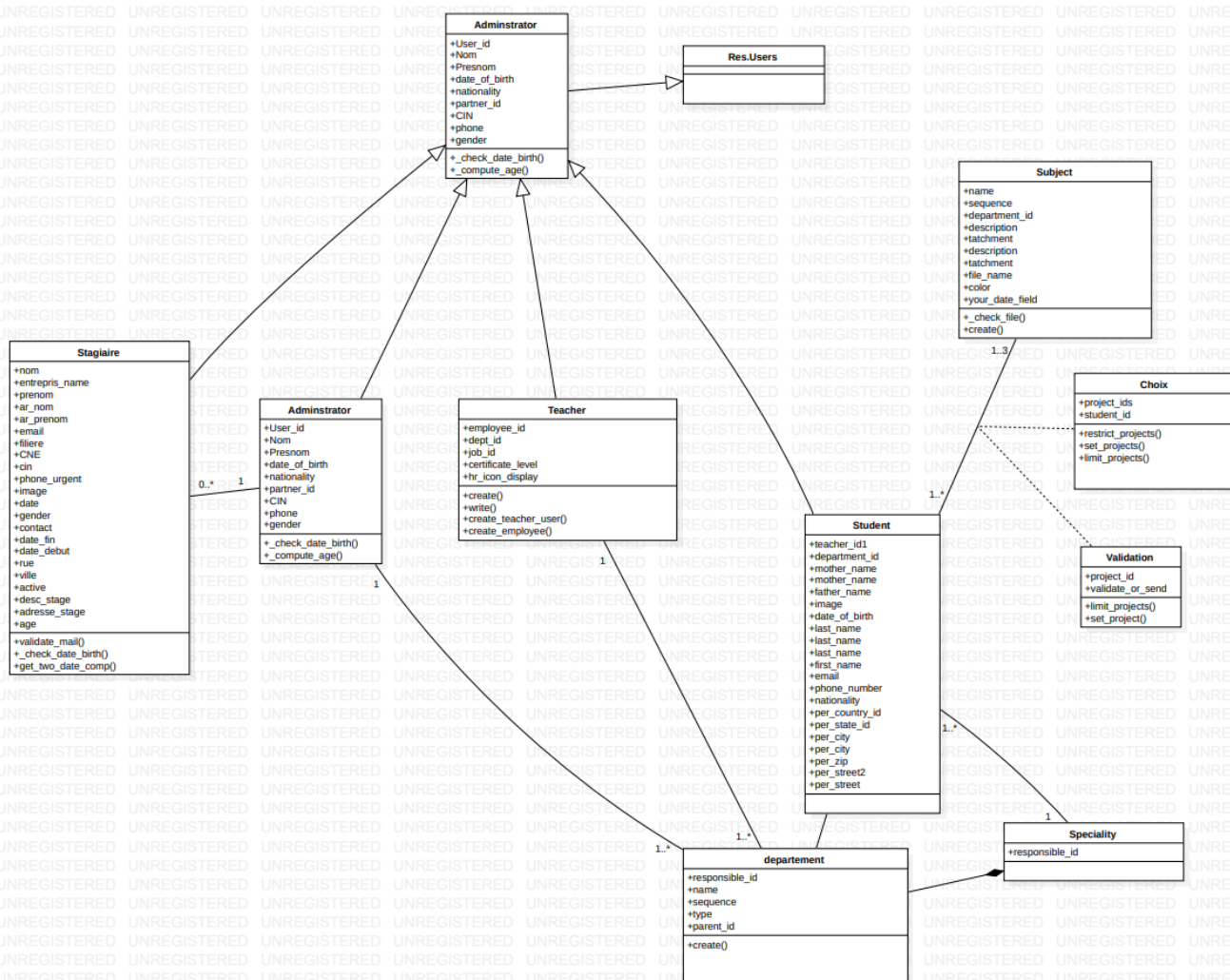


Figure 9 : Diagramme de classe global

## VI. LES DIAGRAMMES DE SEQUENCES

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

Le diagramme de séquences permet de cacher les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages.

Après la description des cas d'utilisation, nous allons élaborer le modèle dynamique dans lequel nous allons décrire les scénarios de quelques cas d'utilisations, les plus importants dans ce sprint, sous forme de diagrammes de séquence.

### 1. Diagramme de séquence « Authentification »

Le diagramme de séquence représenté dans la figure (10) représente le scénario d'authentification d'un utilisateur de l'application Web. Afin d'accéder aux fonctionnalités de notre application, tous les acteurs (Administrateur, Enseignant et étudiant) doivent s'authentifier.

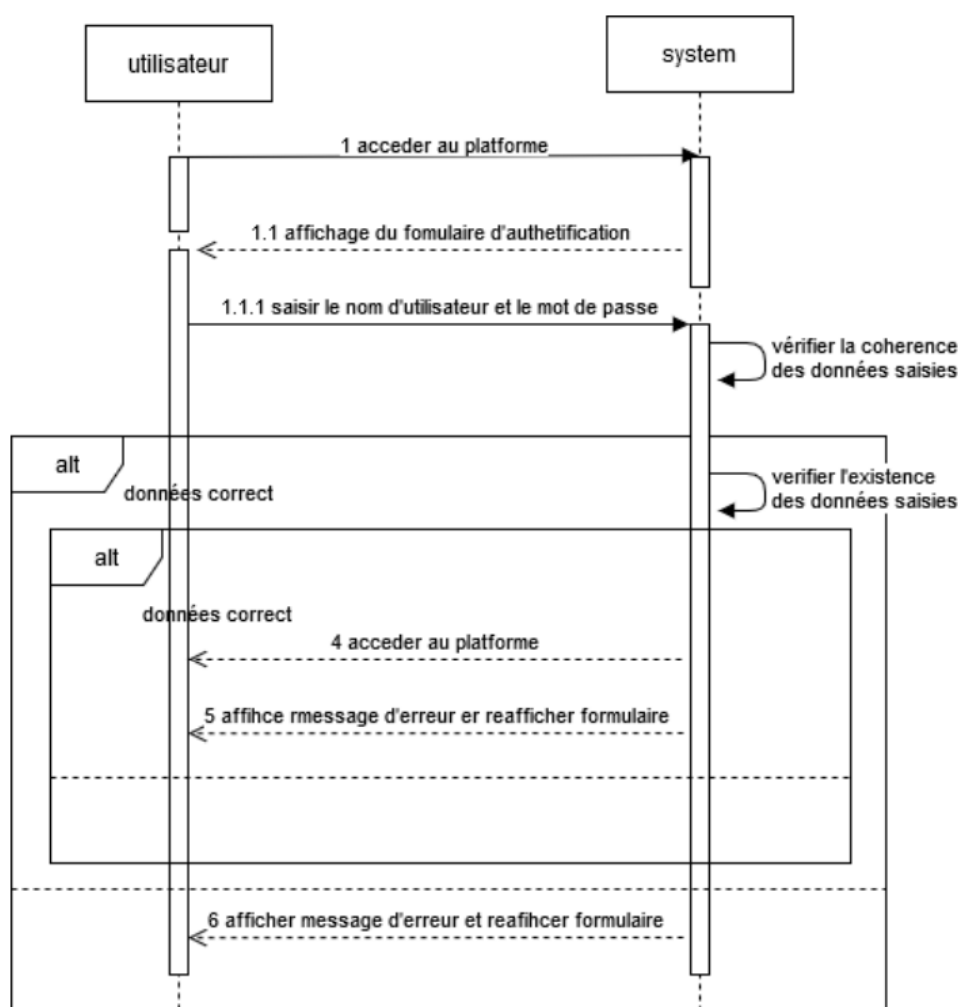


Figure 10 : diagramme de séquence « Authentification »

L'utilisateur doit tout d'abord remplir le formulaire afin de s'authentifier, et ceci en introduisant son login et son mot de passe puis valider l'envoi.

Le système vérifie ensuite si les deux champs ont été bien saisis et bien conformes. Cependant, si l'utilisateur en oublie un champ ou qu'il est mal introduit, le système lui demande de ressaisir. Finalement, et après avoir vérifié que l'utilisateur possède les droits d'accès, le système renvoie la page correspondante sinon il affiche un message d'erreur.

## 2. Diagramme de séquence « ajouter sujet PFE »

Le diagramme de séquence représenté dans la figure (11) représente le scénario d'ajout d'un sujet PFE par l'enseignant. Afin d'initialiser le sujet la plateforme pour qu'il apparaisse aux administrateurs ainsi que les étudiants.

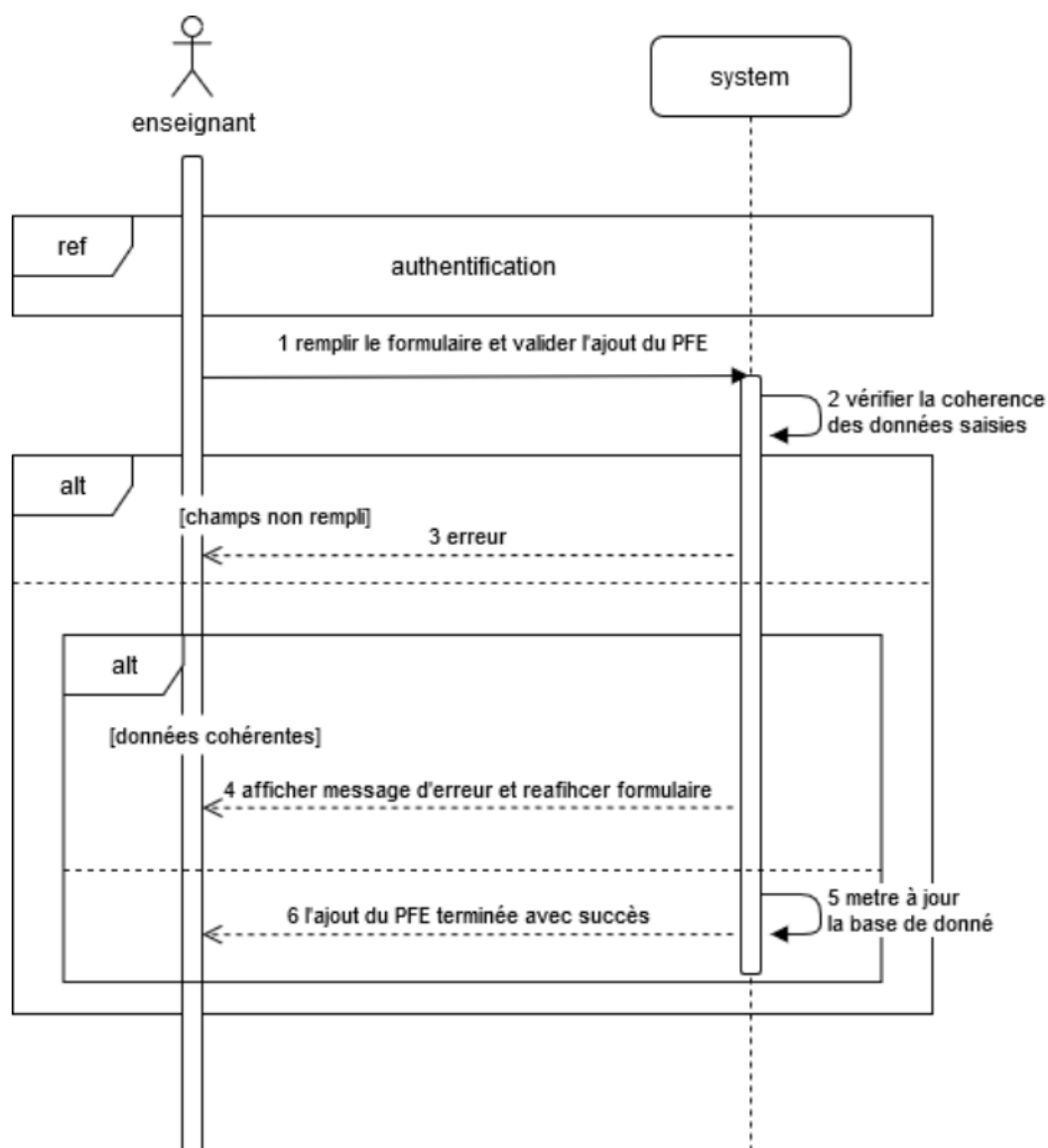


Figure 11: diagramme de séquence « ajouter sujet PFE »

L'enseignant doit tout d'abord remplir le formulaire afin d'ajouter le sujet, et ceci en introduisant les informations concernant le sujet et valider, Le système vérifie ensuite si les champs ont été bien saisis et bien conformes.

Cependant, si l'enseignant en oublie un champ, le système lui demande de ressaisir.

### 3. Diagramme de séquence « valider les choix d'étudiants »

Ce diagramme de séquence (Figure 12) représente le scénario de l'affectation des sujet PFE pour les étudiants de la part de l'administrateur. Après authentification, l'administrateur clique sur le Button lancer l'affectation. Le système fait une mise à jour de la base de données et après, affiche un message de succès.

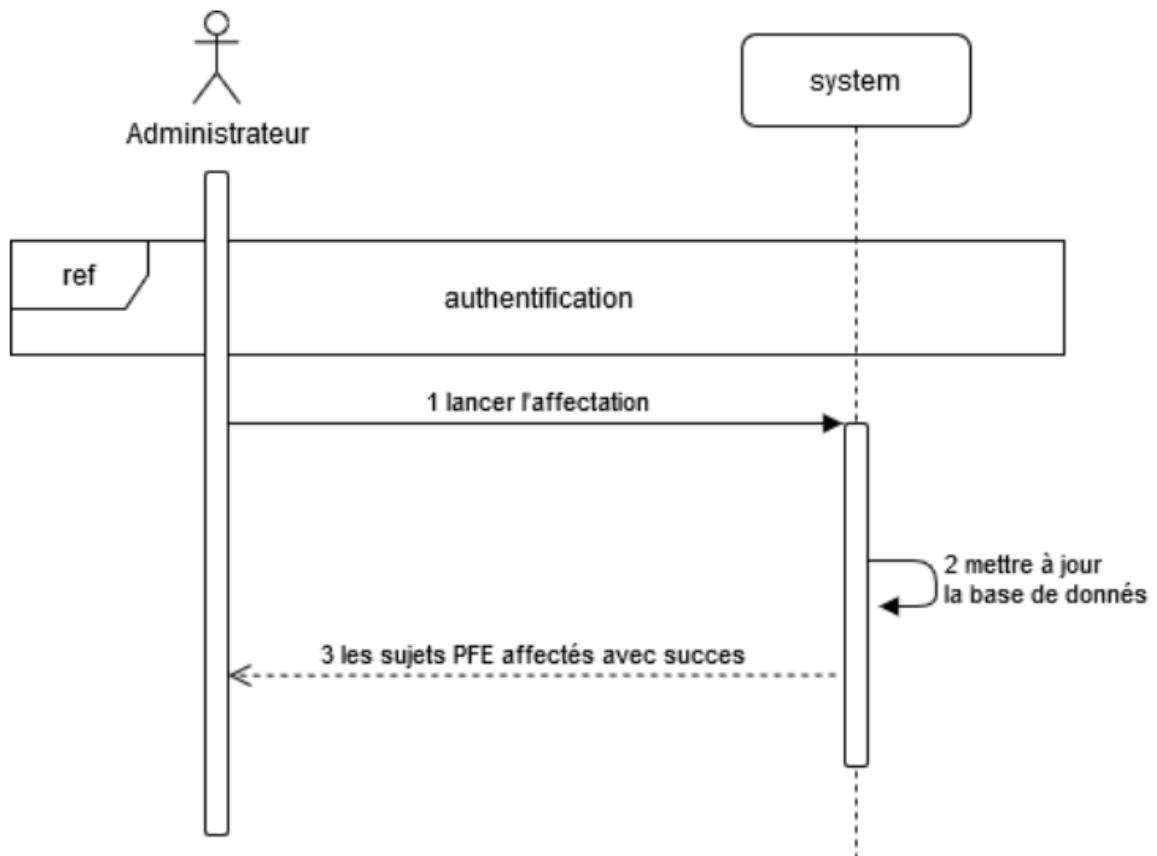


Figure 12: diagramme de séquence « valider les choix d'étudiants »

## *Chapitre III : Réalisation*

# I. PARTIE D'ADMINISTRATEUR

## 1. Management des projets

### Liste des projets

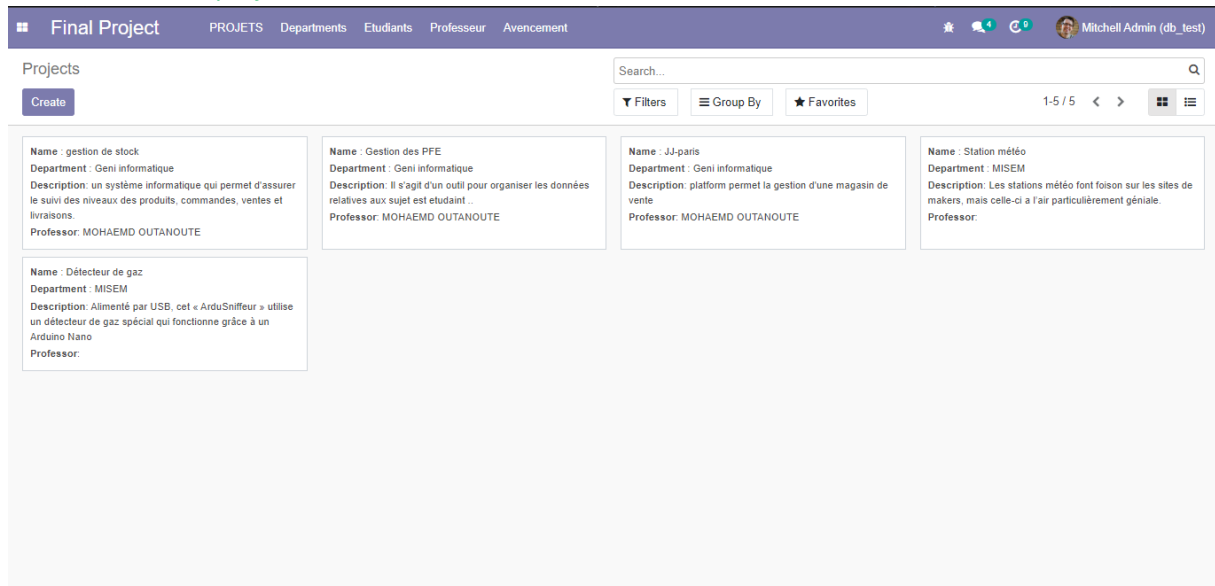


Figure 13 : Liste des projets

Cette kanban List représente l'ensemble des projets réalisés.

### Création d'un nouveau projet

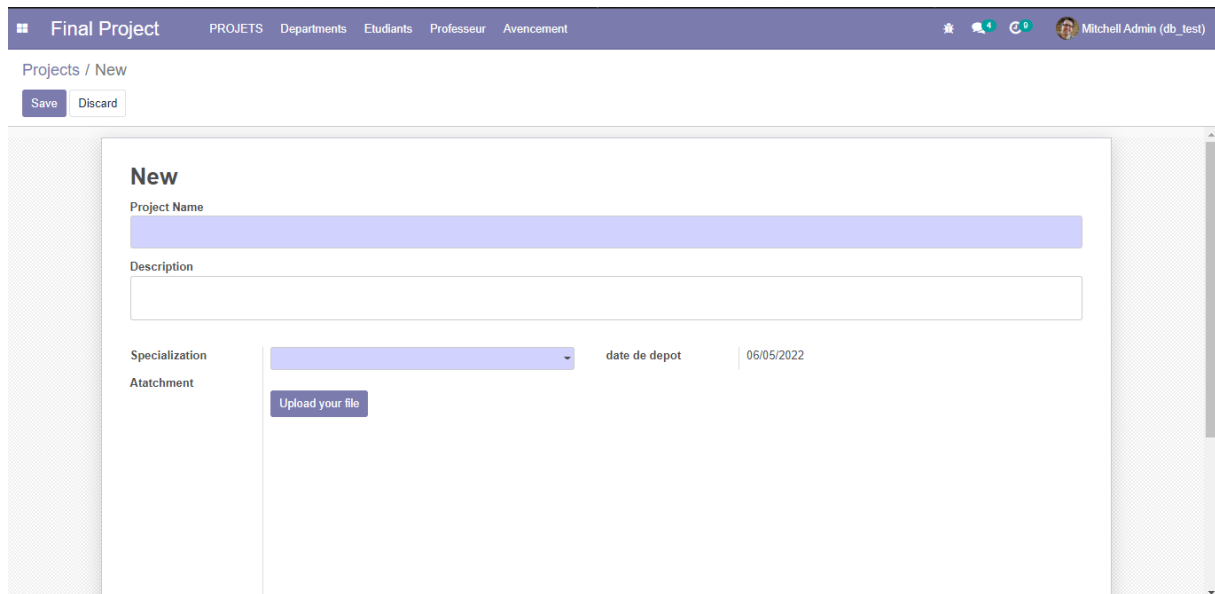


Figure 14: création d'un nouveau projet

Le professeur peu crée un nouveau projet et uploader un fichier (PDF) pour donner plus d'informations a l'étudiant sur le projet, on a cité comme autre information non du sujet et description et date de dépôt et département

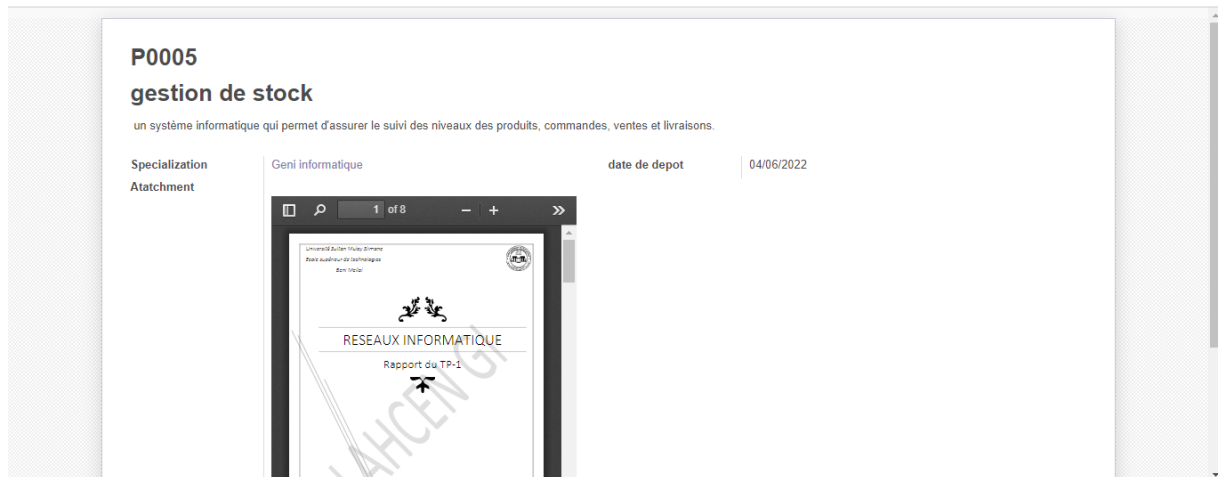


Figure 15 : détails d'un projet

Cette forme montre les informations liées à un projet déjà enregistré sur la base de données. Le professeur ou l'administrateur a le droit de faire des modifications avec un emplacement qui affiche le PDF de façon que l'étudiant puisse lire le PDF sans le télécharger.

## 2. Mangement des étudiants

### Liste des étudiants

Final Project									
PROJETS Departments Etudiants Professeur Avancement									
Students									
<div> <div>Create</div> <div>Search...</div> <div> <div>Filters</div> <div>Group By</div> <div>Favorites</div> </div> <div>1-4 / 4 &lt; &gt;</div> </div>									
<input type="checkbox"/> Name	Date of Birth	Specialization	Supervisor	Identity Number	Gender	Country	State	Kanban State	
<input type="checkbox"/> Zakaria Dinari	05/05/2003	Geni informatique	MOHAEMD OUTANOUTE	Y777790	Male			En cours	
<input type="checkbox"/> Jihan Atouil	05/11/2000	Geni informatique	ABDELLAH AMIN	U89	Female	Morocco		choisi	
<input type="checkbox"/> Mohamed amin	04/29/2003	Mathematique	ABDELLAH AMIN	88u8	Male			choisi	
<input type="checkbox"/> Lahsen aglalal	04/24/2000	MISEM	MOHAEMD OUTANOUTE	Y97883	Male	Morocco		choisi	

Figure 16 : liste des étudiants

Dans cette liste nous avons montré tous les étudiants avec leur spécialisation et un autre état de champ, ce dernier contient trois valorisateurs :

- En attend de validation: l'étudiant a choisi un sujet et à l'attend de validation
- En cous : l'étudiant il n'a pas encore choisi un sujet
- Projet Valide : l'enseignante validée choix d'étudiant

## Création d'un nouvel étudiant

Final Project PROJETS Departments Etudiants Professeur Avancement Mitchell Admin (db\_test)

Students / New

Save Discard

Create Student User

Related User Name

First Name

Last Name

Date of Birth

Age

Father

Mother

Specialization

Supervisor

Identity Number

Kanban State

Gender

Address

Male

Street...

Street 2...

City

Country

State

Nationality

Phone Number

Email

Projects Selected

Accepted Project

Figure 17 : forme de création d'un étudiant

L'administrateur et le professeur ayons le droit de crée un étudiant. La bouton Cérate user permet de crée un nouveau étudiant et avoir l'accès à tous les fonctionnels d'un user du type étudiant

## Détails de l'étudiant

Final Project PROJETS Departments Etudiants Professeur Avancement Mitchell Admin (db\_test)

Students / Lahsen aglagal

Edit Create Action

4 / 4

Related User Name

Name

First Name

Last Name

Date of Birth

Age

Father

Mother

Specialization

Supervisor

Identity Number

Kanban State

Lahsen

Lahsen aglagal

Lahsen

aglagal

04/24/2000

22

omar

aicha

MISEM

MOHAEMD OUTANOUTE

Y97883

Gender

Address

Male

النواحي المسجد العطاوية قلعة السراغنة

attaouia

attouia

Morocco

Morocco

0680372722 SMS

lahsen.aglagal20@usms.ac.ma

Projects Selected

Station météo Détecteur de gaz

Accepted Project

Détecteur de gaz

Figure 18 : détails d'un étudiant

Cette forme contient toutes les informations d'un étudiant, et seul l'administrateur et le professeur peut les modifier ou supprimer un étudiant



### 3. Management des enseignants

#### Liste des enseignants

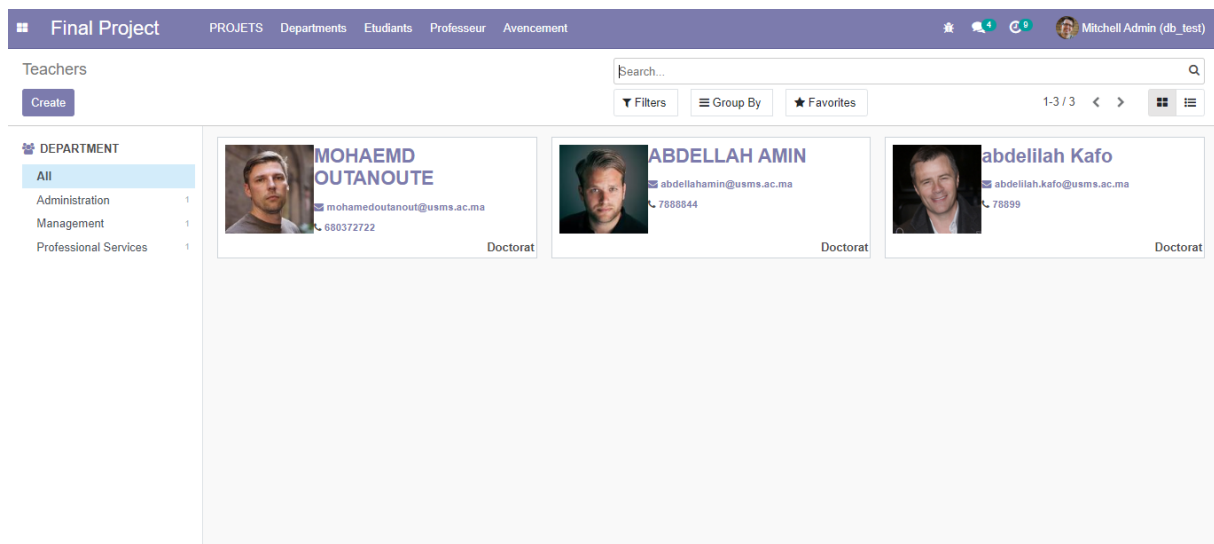


Figure 19 : kanban liste représente les professeurs

Cette liste représente les professeurs sur une kanban liste et aussi une tree liste, groupe par le département et seul l'administrateur peut faire des modifications sur les données des professeurs

#### Création d'un nouvel enseignant

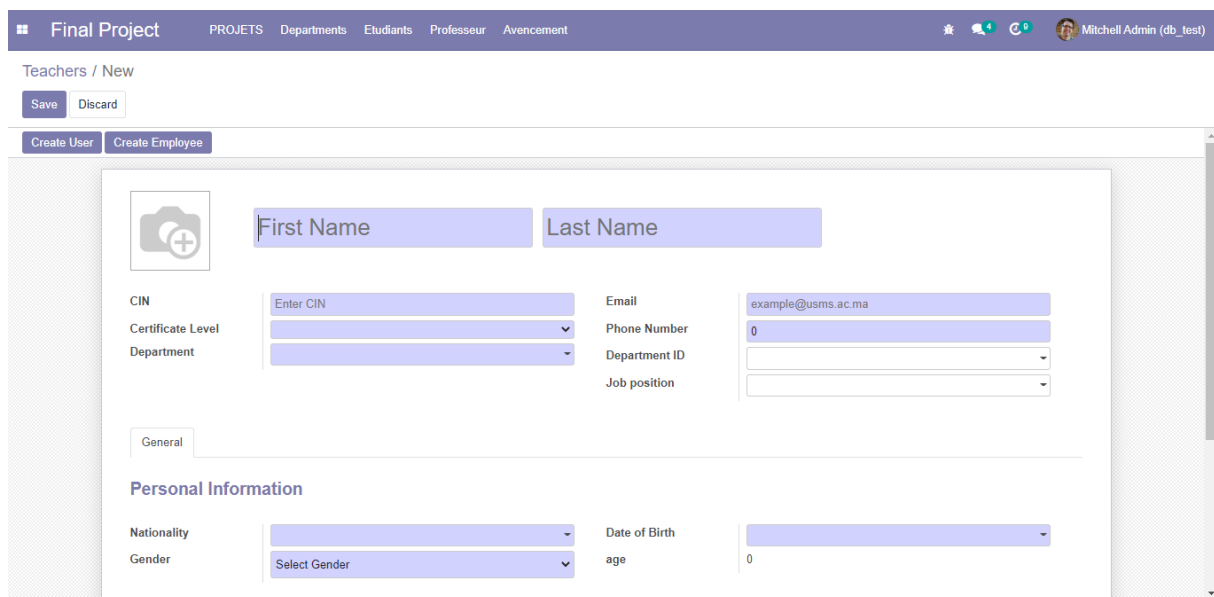


Figure 20: forme de création d'un professeur

Seulement l'administrateur peut créer un professeur. Le bouton 'Créer employé' permet d'ajouter cet utilisateur dans la liste des employés (modèle hr.employees),

Le bouton 'Créer utilisateur' permet de créer un professeur puis l'ajouter dans le modèle res.users d'Odoo afin de donner les privilèges à ce dernier pour se connecter et avoir l'accès à tous les fonctionnels des professeurs.

## Détails d'un enseignant

**Final Project** PROJETS Departments Etudiants Professeur Avancement Mitchell Admin (db\_test)

Teachers / MOHAEMD OUTANOUTE

**MOHAEMD OUTANOUTE**

**General**

CIN: YO9988  
Employee ID: MOHAEMD OUTANOUTE  
Certificate Level: Doctorat  
Department: Megatronique

Email: mohamedoutanout@usms.ac.ma  
Phone Number: 680.372.722  
Department ID: Management  
Job position: Experienced Developer

**Personal Information**

Nationality: Algeria  
Gender: Male  
Date of Birth: 01/04/1999  
age: 23  
User: MOHAEMD OUTANOUTE

Figure 21 : forme représente les détails d'un professeur

Figure 22 : forme représente les détails d'un professeur

Cette forme contient toutes les informations d'un professeur, et seul l'administrateur peut les modifier ou supprimer ce professeur

## 4. Management des départements

### Liste des départements

**Final Project** PROJETS Departments Etudiants Professeur Avancement Mitchell Admin (db\_test)

Departments

Create

Search...

Filters Group By Favorites

1-6 / 6

Sequence	Department Name	Department Manager	Type
<input type="checkbox"/> D0006	Informatique	Mitchell Admin	Parent
<input type="checkbox"/> D0007	Megatronique	Mitchell Admin	Parent
<input type="checkbox"/> D0009	Geni informatique	Mitchell Admin	Child
<input type="checkbox"/> D0010	MISEM	Mitchell Admin	Child
<input type="checkbox"/> D0012	Math	ABDELLAH AMIN	Parent
<input type="checkbox"/> D0013	Mathematique	ABDELLAH AMIN	Child

Figure 23: liste des départements

Cette liste représente l'ensemble des départements, l'admin peut supprimer un filière/département a partir de sélectionner les champs à supprimer puis cliquer sur supprimer sur l'image suivant :

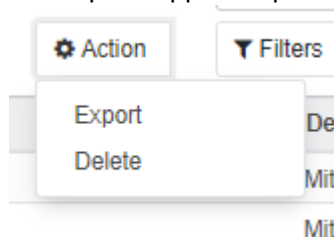


Figure 23 : menu de suppression

## Création d'un nouveau département

The screenshot shows the 'New Department' form in the 'Final Project' application. The form is titled 'New' and contains three input fields: 'Department Name', 'Department Manager', and 'Type'. The 'Type' field is a dropdown menu with 'Parent' selected. Below the dropdown, a list of options is visible: 'Parent' and 'Child'. The form is set against a light gray background with a subtle grid pattern. At the top of the application window, there is a navigation bar with the title 'Final Project' and several menu items: 'PROJETS', 'Departments', 'Etudiants', 'Professeur', and 'Avancement'. On the right side of the navigation bar, there are icons for a star, a speech bubble, a refresh button, and a user profile for 'Mitchell Admin (db\_test)'. Below the navigation bar, there is a sub-header 'Departments / New' and two buttons: 'Save' and 'Discard'.

Figure 24: forme de création d'un département

Un administrateur peut créer un département à partir de cité le choix Parent, il peut aussi créé une filière à partir de cité le choix Child et liait cette dernière avec un département

## Management des stagiaires

### Liste des stagiaires

The screenshot shows the 'Stagiaires' (Interns) Kanban list in the 'Final Project' application. The interface features a top navigation bar with the title 'Stagiaires' and the same user profile as the previous screenshot. Below the navigation bar, there is a sub-header 'kanban test window' and a 'Create' button. The main area displays a Kanban board with three cards, each representing an intern. Each card includes a profile picture, a name, an email address, and a phone number. The cards are arranged horizontally. Above the cards, there is a search bar and a set of controls including 'Filters', 'Group By', 'Favorites', and a pagination indicator '1-3 / 3'.

Figure 25: liste des stagiaires

Cette liste représente la partie des stagiaires et leurs coordonnées sur une liste Kanban, l'administrateur a le droit de crée un nouveau stagiaire il suffit de cliquer sur le bouton Creat

## Création d'un nouveau stagiaire

Figure 26:création d'un stagiaire

Pour crée un stagiaire l'administrateur doit remplir tous les champs, on a cité comme information nom prénom on Arabe et en Français et un \*notebook qui contient d'information sur les stagiaires\*.

## II. PARTIE ETUDIANT

### 1 Cité un projet

#### Liste des sujets

Figure 27:liste des PFE partie étudiant

Dans cette liste l'étudiant peut voir juste les sujets qui ont dans leur département

## Cite un sujet :

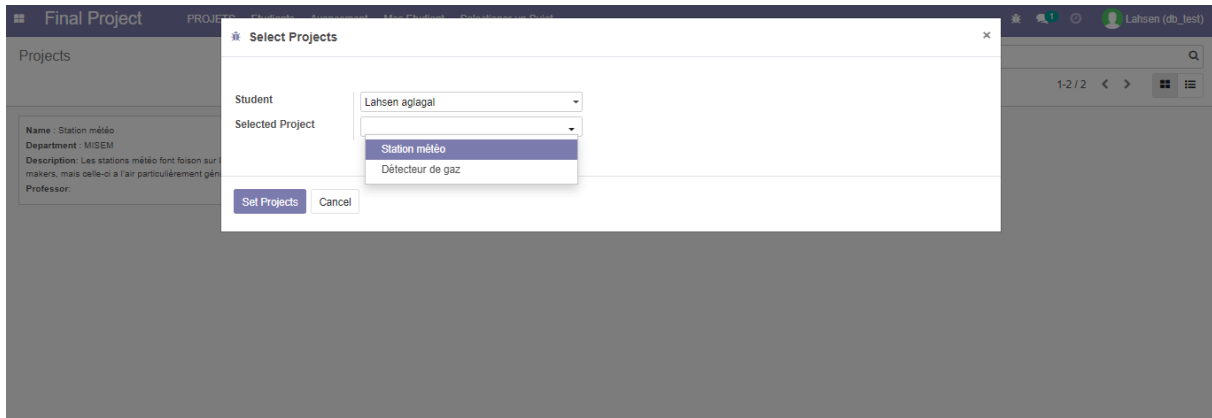


Figure 28:Cite un sujet

Dans cette Pop-up (Wizard) l'étudiant peut citer un sujet à partir de montrer leur nom dans le premier champ, après citer un sujet. Si l'étudiant choisit plus de 3 sujets, une alerte d'erreur s'affiche pour l'informer qu'il a dépassé le nombre possible des sujets.

## 2 Validation de sujet



			
<b>Related User</b>	Lahsen	<b>Gender</b>	Male
<b>Name</b>	Lahsen aglagal	<b>Address</b>	النواحي المسجد العطاوية قلعة السراغنة
<b>First Name</b>	Lahsen		attaouia
<b>Last Name</b>	aglagal		attouia
<b>Date of Birth</b>	04/24/2000		Morocco
<b>Age</b>	22	<b>Nationality</b>	Morocco
<b>Father</b>	omar	<b>Phone Number</b>	0680372722 SMS
<b>Mother</b>	aicha	<b>Email</b>	lahsen.aglagal20@usms.ac.ma
<b>Specialization</b>	MISEM	<b>Projects Selected</b>	Station météo
<b>Supervisor</b>	MOHAEMD OUTANOUTE		
<b>Identity Number</b>	777	<b>Accepted Project</b>	Détecteur de gaz
<b>State</b>			

Figure 29:Validation PFE

Dans le cas où un choix d'un sujet est validé, le champ (accepted project) reçoit la valeur du projet valide, pour l'affecter directement à l'étudiant.

## III. PARTIE PROFESSEUR

Pour la partie des professeurs, c'est tout à fait la même que les pages précédentes. En bas, on a une image qui contient le menu de professeur, qui contient seulement (Project : pour consulter / ajouter / supprimer/modifier un projet) et (mes étudiants : pour consulter / ajouter / supprimer/modifier les étudiants).

Final Project PROJETS Avancement Mes Etudiant

Projects / Nouveau

Sauver Ne pas sauvegarder

**Nouveau**

Project Name  
Construire un indicateur de stationnement de voiture

Description  
Garer une voiture peut être difficile. Cela demande de la précision et beaucoup de pratique. Vous devez utiliser MATLAB pour simplifier les problèmes pour le conducteur.

Specialization  
Mathématique

Attachment  
date de depot 06/06/2022

INTELLCAP

Figure 30: l'ajoute d'un sujet

## IV. PROGRAMMATION : PARTIE CODE

### 1. Python

```
def create_related_user(self):
    # user_group = self.env.ref("graduation_project.group_student") or False
    tab = []
    tab.append(self.env.ref('graduation_project.group_student').id)
    tab.append(self.env.ref('base.group_user').id)
    user_group = self.env['res.groups'].search([('id', 'in', tab)])
    users_res = self.env['res.users']
    for record in self:
        if not record.related_user_id:
            user_id = users_res.create({
                'name': record.first_name,
                'login': record.email,
                'student_id': record.id,
                'groups_id': user_group,
                'graduatio_department_id': record.department_id.id,
                'password': str(record.first_name) + 'odoo',
                'tz': self._context.get('tz'),
            })
            record.related_user_id = user_id.id
```

Figure 31: fonction python (l'ajoute d'un utilisateur)

Dans la Forme de la création d'une étudiant/enseignant en haut il y a un bouton create user, Après avoir cliqué sur ce bouton cette fonction s'exécute :

- Il joute de l'utilisateur crée au groupe des étudiants
- Inséré les données de l'utilisateur dans la table des utilisateurs odoo (res.users).
- Génère un mot de passe automatiquement (le nom de l'utilisateur + odoo)

```
@api.model
def create(self, vals):
    vals['sequence'] = self.env['ir.sequence'].next_by_code(
        'project.sequence')
    res = super(FinalProject, self).create(vals)
    return res
```

Figure 32: Redéfinition de la fonction crée

Le champ séquencé ne peut pas être inséré depuis l'utilisateur, pour cela nous avons redéfini la fonction créer (Ouvrride create fuction) pour envoyer une séquence automatiquement avec les autres données

```
@api.constrains('tatchment')
def _check_file(self):
    if str(self.file_name.split(".")[1]) != 'pdf':
        raise ValidationError("Cannot upload file different from .pdf file")
```

Figure 33: validation d'un fichier PDF

Cette fonction garantit que l'enseignant peut uniquement charger des fichiers PDF sur la plateforme, si l'enseignant tente d'ajouter d'autres fichiers, une alerte d'erreur s'affichera.

## 2. XML

```
<record id="rule_professor_project" model="ir.rule">
    <field name="name">professor see his projects</field>
    <field name="model_id" ref="model_final_project"/>
    <field name="groups" eval="[(4, ref('graduation_project.group_professor')))]"/>
    <field name="perm_read" eval="True"/>
    <field name="perm_write" eval="True"/>
    <field name="perm_create" eval="True"/>
    <field name="perm_unlink" eval="True"/>
    <field name="domain_force">[('create_uid','=',user.id)]</field>
</record>
```

Figure 34 :Record rule

Ce record garantit que l'enseignant peut voir uniquement ces sujets

```
<record id="group_professor" model="res.groups">
    <field name="name">Professor</field>
    <field name="implied_ids" eval="[(4,ref('hr.group_hr_user')))]"/>
    <field name="category_id" ref="graduation_project.module_category_graduation_project"/>
</record>
```

Figure 35:Création d'un group

Ce record représente la création d'un group professeur, c'est tout à fait la même chose pour les autre groups (student & admin)

# *Conclusion générale*

Dans le cadre de notre projet fin d'étude, nous avons participé à la conception et la réalisation d'un module Odoo pour la gestion de choix de projet de fin d'études :

Le travail sur cette mission, nous à aider à approfondir et élargir nos connaissances et nos acquises durant le cursus universitaire, et tous en suivant des méthodes de modélisation et de conception.

Nous sommes assez certains que cette application sera bénéfique et qui va permettre d'améliorer le système de gestion des choix de projet fin d'études pour notre école.

Enfin, nous espérons que ce travail pourra aider et servir de bonnes références à toutes personne qui voudra s'engager dans un travail similaire.



# Webographie

- ✓ [Odoo Documentation — Odoo 14.0 documentation](#)
- ✓ [OpenERP Documentation v14\(odoo.com\)](#)
- ✓ [Odoo Apps | Odoo Apps Store \(openerp.com\)](#)
- ✓ [Odoo Help - Forum](#)