



PROJET ERP ODOO

*5ème Année en Ingénierie Informatique et Réseaux
Option MIAGE*

Création d'un Module Odoo

Module de Réservation des Salles

Réalisée par :

Mohamed amine fouiteh

Encadrée par :

Pr. Mohammed AITDAOUD

Remerciement

Je tiens tout d'abord à exprimer ma profonde gratitude à mon encadrant **Pr. Mohammed AITDAOUD**, pour son accompagnement, ses conseils précieux, sa disponibilité ainsi que son suivi rigoureux tout au long de la réalisation de ce projet.

Je remercie également l'ensemble des enseignants et du personnel de **l'École Marocaine des Sciences de l'Ingénieur**, pour la qualité de la formation dispensée, ainsi que pour les moyens et l'encadrement mis à notre disposition, qui ont grandement contribué à l'aboutissement de ce travail.

Enfin, je tiens à remercier mes camarades et toutes les personnes qui m'ont soutenu de près ou de loin, par leurs encouragements, leurs remarques et leurs contributions, qui ont été d'une grande aide durant la réalisation de ce projet.

Résumé

Dans le cadre de ce projet, nous avons développé un module Odoo dédié à la **gestion et la réservation des salles**, visant à répondre aux besoins d'organisation et de planification des espaces au sein d'un établissement (université, centre de formation ou entreprise). En effet, la réservation manuelle des salles (via des échanges informels ou des tableaux externes) peut entraîner des erreurs fréquentes telles que les **conflits d'occupation**, la perte d'informations, ainsi qu'une mauvaise visibilité sur la disponibilité des ressources.

Le module réalisé permet de **centraliser l'ensemble du processus de réservation** à travers une interface simple et ergonomique intégrée à Odoo. Il offre la possibilité d'ajouter et de gérer les salles, de créer des réservations en spécifiant la date, l'heure de début, l'heure de fin, le responsable et le statut, ainsi que de consulter les réservations existantes via différentes vues (liste et formulaire).

Afin d'améliorer la fiabilité du système, des **règles automatiques de validation** ont été mises en place, notamment pour vérifier la cohérence des heures et éviter toute réservation en conflit. Un workflow (brouillon, confirmée, annulée, terminée) permet également de suivre l'état des réservations et d'assurer la traçabilité des opérations.

Ce projet a été développé en utilisant **Odoo 17**, avec le langage **Python** pour la logique métier, **PostgreSQL** pour la gestion des données et **Docker** pour faciliter le déploiement et l'exécution. La solution obtenue constitue un outil efficace, fiable et évolutif pour optimiser la gestion des salles et améliorer l'organisation des activités.

Mots-clés : Odoo 17, réservation des salles, gestion des ressources, workflow, validation automatique, Python, PostgreSQL, Docker.

.

Abstarct

This project focuses on developing an Odoo module dedicated to room reservation management, aiming to improve the organization and scheduling of shared spaces within an institution. The module centralizes all booking operations through an intuitive interface, allowing users to manage rooms, create reservations by specifying date, start/end time, responsible person, and status, and consult bookings through list and form views.

To ensure reliability, the system includes automated business rules to validate time consistency and prevent booking conflicts. A workflow (draft, confirmed, canceled, completed) guarantees clear tracking of reservations and full traceability of actions within Odoo.

Technically, the solution is built on **Odoo 17**, using **Python** for business logic, **PostgreSQL** for data storage, and **Docker** for reliable deployment. The resulting module is scalable, professional, and ready for real-world use.

Keywords: Odoo 17, room reservation, resource management, workflow, Python, PostgreSQL, Docker.

Liste des abréviations

CRUD :	Create, Read, Update, Delete (Créer, Lire, Mettre à jour, Supprimer)
CSV :	Comma-Separated Values (Valeurs Séparées par des Virgules)
DB :	Database (Base de Données)
ERP :	Enterprise Resource Planning (Progiciel de Gestion Intégré)
XML:	eXtensible Markup Language
ORM:	Object Relational Mapping

Liste des figures

Figure 1: Structure des Fichiers.....	4
Figure 2: Fichier docker-compose.yml.....	5
Figure 3: Fichier : models/reservation.py	6
Figure 4: Fichier ir.model.access.csv.....	6
Figure 5: Code XML	6
Figure 6: page d'authentification	7
Figure 7: mode développeur.....	8
Figure 8:Update du module.....	8
Figure 9: Installation du module	9
Figure 10: Vue Liste	11
Figure 11: Vue Formulaire	11
Figure 12: Création d'un Responsable.....	12
Figure 13: Export Excel	13

Liste des Tableaux

Tableau 1: Technologies Utilisées.....	4
--	---

Introduction Générale

Avec l'évolution des technologies numériques et la transformation digitale des organisations, la gestion des ressources internes devient un enjeu majeur pour garantir une meilleure productivité et une organisation efficace. Parmi ces ressources, les salles (salles de réunion, salles de cours, amphithéâtres, laboratoires, salles informatiques, etc.) occupent une place essentielle dans le fonctionnement quotidien d'un établissement, qu'il soit universitaire ou professionnel.

Cependant, dans plusieurs structures, le processus de réservation des salles reste encore géré de manière manuelle, à travers des échanges informels, des appels téléphoniques ou des tableaux externes. Cette méthode traditionnelle engendre souvent des problèmes tels que les **réservations en double**, les conflits d'occupation, l'absence de visibilité sur la disponibilité des salles, ainsi que la difficulté de conserver un historique fiable des réservations. Ces difficultés impactent directement l'organisation des activités, la planification des événements et la satisfaction des utilisateurs.

Face à ces contraintes, il devient nécessaire de mettre en place une solution centralisée permettant d'automatiser et de sécuriser la gestion des réservations, tout en assurant la traçabilité et le suivi des demandes. Dans ce contexte, les solutions ERP comme **Odoo** représentent un choix pertinent, grâce à leur flexibilité, leur architecture modulaire et leur capacité à intégrer plusieurs fonctionnalités au sein d'un même système.

C'est dans cette optique que s'inscrit ce projet, qui consiste à développer un **module Odoo de réservation des salles**. Ce module vise à offrir une interface simple et intuitive permettant de gérer les salles, de créer des réservations selon des critères précis (date, heure, durée, responsable, motif), et d'appliquer des règles automatiques afin d'éviter tout conflit d'occupation. De plus, un workflow clair (brouillon, confirmée, annulée, terminée) permet un meilleur suivi des réservations et une meilleure organisation des ressources.

Ainsi, ce projet a pour objectif de proposer une solution fiable et évolutive, capable d'améliorer l'organisation interne, de réduire les erreurs humaines et d'optimiser l'utilisation des salles au sein de l'établissement.

Chapitre 1 : Présentation Générale du Projet

INTRODUCTION

Ce chapitre présente l'étude et la conception du module « **Réservation des Salles** » développé dans le cadre du cours de développement d'applications avec Odoo. Il expose le contexte du projet, la problématique rencontrée dans la gestion manuelle des réservations ainsi que la solution proposée. Le chapitre détaille également les objectifs pédagogiques et fonctionnels, l'architecture du module, les technologies utilisées et les étapes d'installation et de configuration. L'objectif est d'établir une base conceptuelle et technique solide avant d'aborder la réalisation et les interfaces.

1.1 Contexte du Projet

La réservation des salles constitue une tâche essentielle dans tout établissement disposant d'espaces partagés. L'objectif de ce projet est de concevoir un module Odoo permettant de gérer efficacement les salles et leurs réservations afin de garantir une meilleure organisation et une meilleure disponibilité des ressources.

1.2 Problématique

Les établissements rencontrent plusieurs difficultés lorsque la réservation se fait de manière manuelle :

- ☒ conflits de réservation (chevauchement),
- ☒ manque de visibilité en temps réel,
- absence d'historique,
- perte de temps dans la coordination.
-

1.3 Solution Proposée

Développement d'un module Odoo nommé "**Réservation des Salles**", offrant :

- gestion des salles,
- gestion des réservations,
- workflow de statut (Brouillon / Confirmée),
- export des réservations,
- interface intuitive (Tree + Form).

2. OBJECTIFS DU TP

2.1 Objectifs Pédagogiques

- Comprendre l'architecture d'Odoo (MVC + ORM)
- Créer des modèles et relations
- Développer des vues (Tree / Form)

- Implémenter un workflow métier
- ☑ Gérer la sécurité et les droits

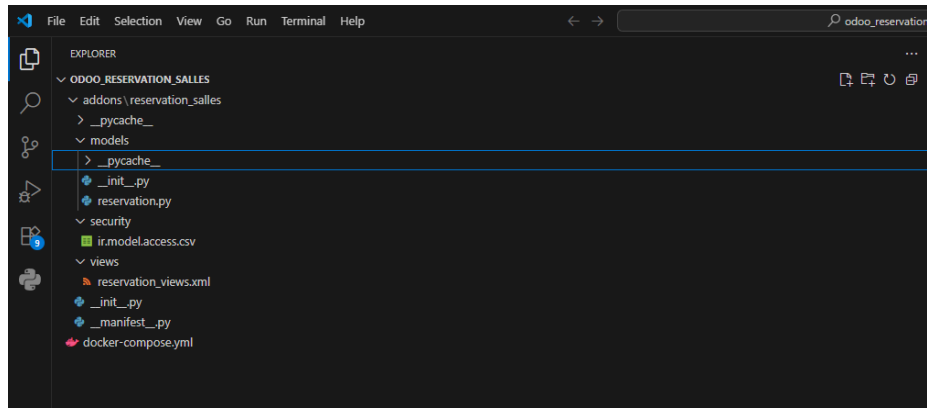
2.2 Objectifs Fonctionnels

- Ajouter et gérer les salles
- Créer une réservation (date + heure début/fin)
- Affecter un responsable
- Gérer le statut (Brouillon / Confirmée)
- Rechercher et filtrer
- Export Excel/CSV

3. ARCHITECTURE DU MODULE

3.1 Structure des Fichiers

Figure 1: Structure des Fichiers



3.2 Technologies Utilisées

Tableau 1: Technologies Utilisées

Technologie	Version	Utilisation
Odoo	17.0	Framework principal
Python	3.10+	Logique métier/backend
PostgreSQL	16	Base de données
XML	1.0	Vues / interface
Docker	Latest	Conteneurisation

4. INSTALLATION ET CONFIGURATION

4.1 Prérequis Système

Matériel requis :

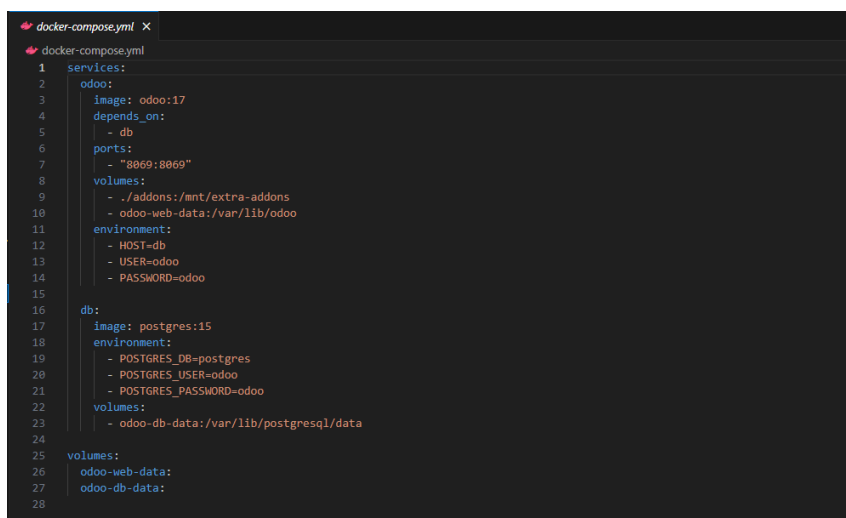
- CPU : 2 cores minimum
- RAM : 4 GB minimum
- Stockage : 10 GB minimum

Logiciels requis :

- Docker Desktop
- Git
- VS Code recommandé

4.2 Configuration du projet

Fichier docker-compose.yml :



```
1 services:
2   odoo:
3     image: odoo:17
4     depends_on:
5       - db
6     ports:
7       - "8069:8069"
8     volumes:
9       - ./addons:/mnt/extra-addons
10      - odoo-web-data:/var/lib/odoo
11     environment:
12       - HOST=db
13       - USER=odoo
14       - PASSWORD=odoo
15
16   db:
17     image: postgres:15
18     environment:
19       - POSTGRES_DB=postgres
20       - POSTGRES_USER=odoo
21       - POSTGRES_PASSWORD=odoo
22     volumes:
23       - odoo-db-data:/var/lib/postgresql/data
24
25 volumes:
26   odoo-web-data:
27   odoo-db-data:
```

Figure 2: Fichier docker-compose.yml

Modèle Principal : gestion.formation

Fichier : models/reservation.py:

```

reservation.py 1 X
addons > reservation_salles > models > reservation.py > ...
1 from odoo import models, fields
2
3 class ReservationSalle(models.Model):
4     _name = 'reservation.salle'
5     _description = 'Réservation de Salle'
6
7     name = fields.Char(string="Nom de la salle", required=True)
8     date_reservation = fields.Date(
9         string="Date de réservation",
10        required=True
11    )
12     heure_debut = fields.Float(
13         string="Heure de début"
14     )
15     heure_fin = fields.Float(
16         string="Heure de fin"
17     )
18     responsable_id = fields.Many2one(
19         'res.users',
20         string="Responsable",
21         default=lambda self: self.env.user
22     )
23     state = fields.Selection(
24         [
25             ('brouillon', 'Brouillon'),
26             ('confirnee', 'Confirmée'),
27             ('annulee', 'Annulée')
28         ],
29         string="Statut",
30         default='brouillon'
31     )
32

```

Figure 3: Fichier : models/reservation.py

Fichier ir.model.access.csv :

```

ir.model.access.csv X
addons > reservation_salles > security > ir.model.access.csv
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_reservation_salle,access.reservation.salle,model_reservation_salle,,1,1,1,1
3

```

Figure 4: Fichier ir.model.access.csv

VUES ET INTERFACE

Objectif : Afficher toutes les formations sous forme de tableau

Code XML :

```

reservation_views.xml X
addons > reservation_salles > views > reservation_views.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <odoo>
3
4 <!-- Vue LISTE -->
5 <record id="view_reservation_salle_tree" model="ir.ui.view">
6     <field name="name" res="reservation.salle.tree"/>
7     <field name="model" res="reservation.salle"/>
8     <field name="arch" type="xml">
9         <tree string="Réservations de salles">
10             <field name="name"/>
11             <field name="date_reservation"/>
12             <field name="heure_debut"/>
13             <field name="heure_fin"/>
14             <field name="state"/>
15             <field name="responsable_id"/>
16         </tree>
17     </field>
18 </record>
19
20 <!-- Vue FORMULAIRE -->
21 <record id="view_reservation_salle_form" model="ir.ui.view">
22     <field name="name" res="reservation.salle.form"/>
23     <field name="model" res="reservation.salle"/>
24     <field name="arch" type="xml">
25         <form string="Réservation de salle">
26             <sheet>
27                 <group>
28                     <field name="name"/>
29                     <field name="date_reservation"/>
30                 </group>
31                 <group>
32                     <field name="heure_debut"/>
33                     <field name="heure_fin"/>
34                 </group>
35                 <group>
36                     <field name="responsable_id"/>
37                     <field name="state"/>
38                 </group>
39             </sheet>
40         </form>
41     </field>
42 </record>

```

Figure 5: Code XML

Fonctionnalités :

- Tri par colonne

- Recherche rapide
- Coloration conditionnelle (vert = terminé, bleu = en cours)
- Export Excel/CSV

4.3 Étapes d'Installation

Étape 1 : Démarrer Docker

`docker-compose up -d`

Étape 2 : Créer la base de données

- Accéder à <http://localhost:8069>
- Email admin : `fouitehdb@gmail.com`
- Mot de passe : `admin`

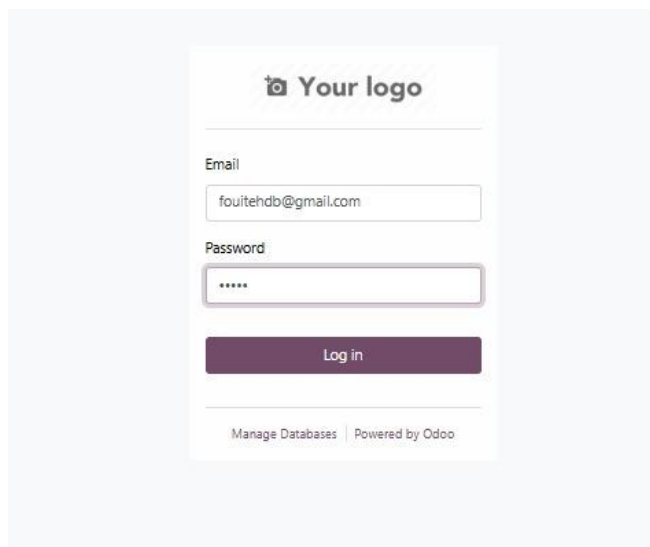


Figure 6: page d'authentification

Étape 3 : Installer le module

- Menu Apps → Rechercher "Réservation de Salles"
- Cliquer sur "Installer"

4.4 Configuration Initiale

Activation du mode développeur :

1. Activer le mode développeur (si nécessaire)
2. Ou via URL : `http://localhost:8069/web?debug=1`



Figure 7: mode développeur

Mettre à jour la liste des applications

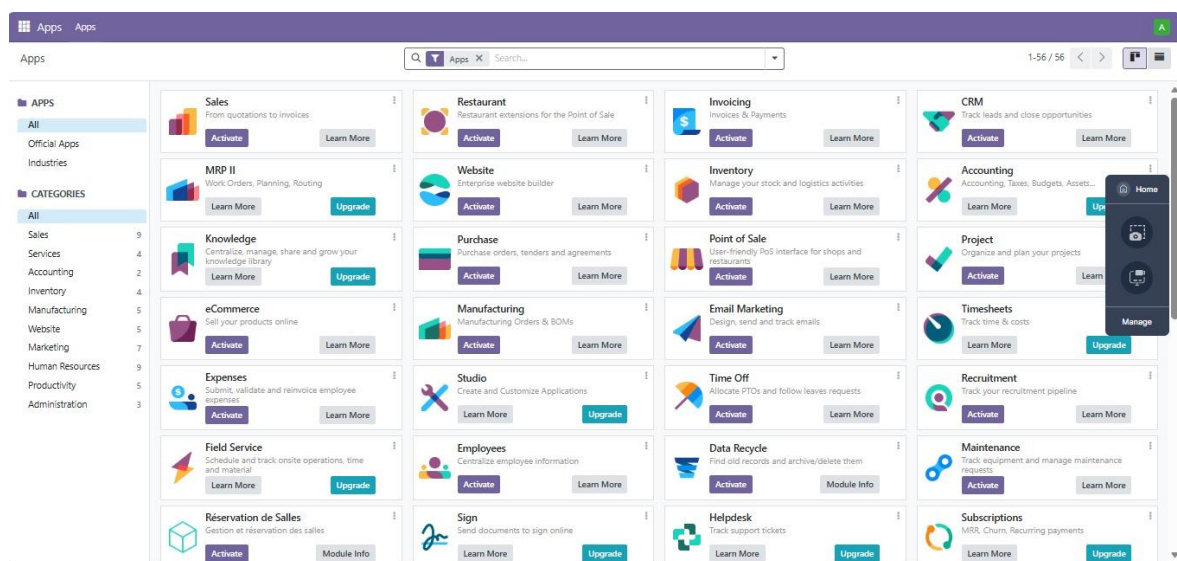


Figure 8: Update du module

Activation du module

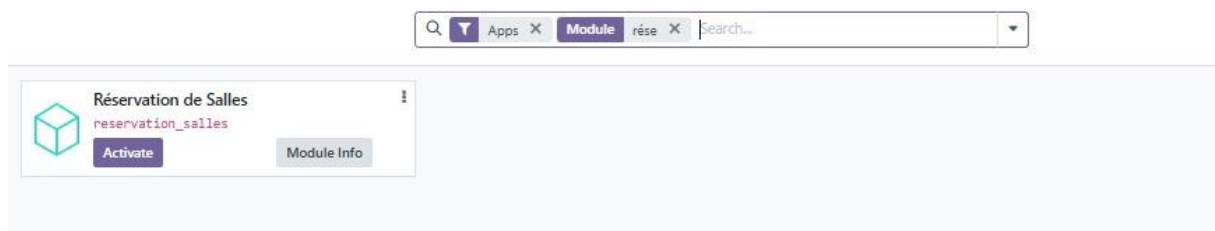


Figure 9: Installation du module

Conclusion

Ce chapitre a présenté le contexte, la problématique, les objectifs et la solution proposée à travers le module **Réservation des Salles**. L'architecture du module ainsi que l'environnement technique ont été décrits afin de préparer la phase de réalisation. Le chapitre suivant présentera les interfaces utilisateur, les fonctionnalités implémentées et les démonstrations du module.

Chapitre 2 : Réalisation et Interfaces du Projet

Introduction

Ce chapitre présente la réalisation du module, notamment les interfaces utilisateur (vues liste et formulaire), ainsi que les principales fonctionnalités : création de réservation, affectation d'un responsable, gestion des statuts, et export des données.

2.1 Développement des Interfaces Utilisateur

2.1.1. Vue Liste (Tree View)

La vue liste affiche un tableau récapitulatif des réservations. Les colonnes principales sont :

- Nom de la salle
- Date de réservation
- Heure de début
- Heure de fin
- Statut
- Responsable

Cette vue permet la recherche rapide, le tri, ainsi que l'export des réservations en Excel/CSV.

New

Réervations de salles

Q Search...

1-2/

<input type="checkbox"/> Nom de la salle	Date de réservation	Heure de début	Heure de fin	Statut	Responsable
<input type="checkbox"/> Salle de réunion A	05/02/2026	10:00	12:00	Brouillon	Administrator
<input type="checkbox"/> Salle de formation	08/02/2026	14:00	17:00	Confirmée	fouteh

Figure 10: Vue Liste

2.1.2. Vue Formulaire

The screenshot shows a web application interface for room reservations. At the top, there is a header bar with the text 'Reservations de salles' and a 'New' button. Below this, a form is displayed with the following fields and values:

Nom de la salle	Salle de réunion A
Date de réservation	05/02/2026
Heure de début	10.00
Heure de fin	12.00
Responsable	Administrator
Statut	Brouillon

Figure 11: Vue Formulaire

Cette capture montre le formulaire de création/modification d'une formation intitulée "Python Avancé". En haut se trouve la barre de statut avec les boutons d'action (Planifier en violet, Annuler en

rouge) et les états du workflow (Brouillon, Planifié, En cours, Terminé). Le formulaire est organisé en deux sections : Informations Générales (Type de formation : Formation Interne, Catégorie : Formation Continue, Responsable : Mitchell Admin, Lieu : Salle informatique A) et Planification (Date de début : 12/28/2025, Date de fin : 12/31/2025, Durée : 24.00 heures calculée automatiquement, Capacité maximale : 30). En bas, cinq onglets permettent d'accéder aux détails : Description, Formateurs, Inscrits, Documents, et Évaluation. À droite, le chatter affiche l'historique avec "Mitchell Admin - Creating a new record..."

2.1.3. Vue Formulaire - Création d'un responsable

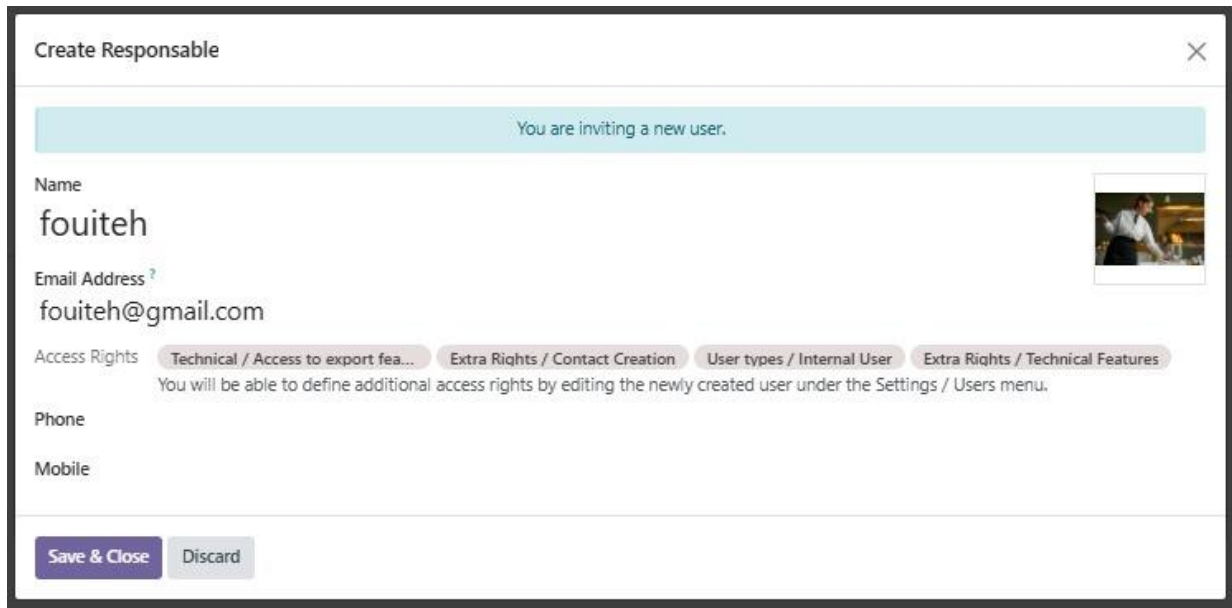


Figure 12: Création d'un responsable

Lors de la sélection du responsable, Odoo permet de créer un nouvel utilisateur/contact via une popup. Cette fonctionnalité facilite la gestion des responsables directement depuis la réservation.

2.1.4. Export Excel/CSV

Le module permet d'exporter la liste des réservations via le menu d'action d'Odoo (Export). Cela permet l'exploitation des données dans Excel.

A1	✕	✓	fx	Nom de la salle	▼
	A	B	C	D	
1	Nom de la salle	Date de réservation	Heure de début	Heure de fin	Statut
2	Salle de réunion A	2026-05-02	10,00	12,00	Brouillon
3	Salle de formation	2026-08-02	14,00	17,00	Confirmée
4					
5					
6					

Figure 13: Export Excel

Conclusion

Ce chapitre a présenté les interfaces développées et leur fonctionnement. La vue liste offre une vision globale et rapide des réservations, tandis que la vue formulaire permet la création, la gestion des statuts et la traçabilité. L'intégration de la création du responsable et l'export Excel renforcent la flexibilité et l'utilité du module.

Conclusion générale

Au terme de ce projet, un module Odoo complet et opérationnel dédié à la **réservation des salles** a été conçu et développé avec succès. La solution répond aux objectifs initiaux en centralisant les réservations et en améliorant la visibilité sur la disponibilité des salles. Les vues développées permettent une gestion simple et professionnelle, tandis que les fonctionnalités de tri, recherche et export offrent une meilleure exploitation des données.

Sur le plan technique, le projet a permis de consolider des compétences en **développement Odoo**, notamment la création de modèles via ORM, la création de vues XML, la gestion des droits d'accès, et l'utilisation de Docker pour un déploiement maîtrisé.

Plusieurs perspectives peuvent enrichir le module : ajout d'une vue calendrier, notifications automatiques par email, gestion des équipements liés aux salles, ou encore un système d'approbation multi-niveaux.

BIBLIOGRAPHIE

[1] **Odoo S.A.** (2024). *Odoo 17.0 Documentation Officielle*. Disponible en ligne :

<https://www.odoo.com/documentation/17.0/>

[2] **Odoo S.A.** (2024). *Odoo ORM API Reference*. Disponible en ligne :

<https://www.odoo.com/documentation/17.0/developer/reference/backend/orm.html>

[3] **Docker Inc.** (2024). *Docker Documentation*. Disponible en ligne :

<https://docs.docker.com/>