

RAPPORT DE PROJET ERP ODOO

Ingénierie Informatique et Réseaux

Sous le thème :

**SYSTÈME DE GESTION DE BIBLIOTHÈQUE
AVEC ODOO ERP**

Réalisé par :
OUBEGHDAD Arwa

Encadré par :
Mr. Mohammed Aitdouad

Dédicace

Nous dédions ce travail à toutes les personnes qui ont, de près ou de loin, contribué à la réalisation de ce projet et à la réussite de notre parcours académique.

À nos familles respectives, et tout particulièrement à nos parents, pour leur soutien indéfectible, leur patience, leur compréhension et leurs encouragements constants. Leur présence bienveillante, leurs conseils avisés et leur confiance inébranlable ont été une source permanente de force et de motivation, nous permettant de surmonter les difficultés et d'avancer avec détermination.

À nos enseignants et à notre encadrant académique, **Mr. Mohammed Aitdouad**, pour leur dévouement, leur passion pour la transmission du savoir et leur accompagnement méthodologique et scientifique. Leurs orientations précieuses, leurs conseils éclairés et leur rigueur académique ont grandement enrichi notre travail et contribué à la réussite de ce projet.

Enfin, à toutes les personnes qui nous ont inspirés, soutenus et encouragés, directement ou indirectement, tout au long de notre parcours académique et personnel. Ce travail est le fruit d'un effort collectif et d'un engagement sincère.

Avec toute notre gratitude et notre profonde reconnaissance.

Remerciements

Nous tenons à exprimer notre profonde gratitude à notre encadrant académique, **Mr. Mohammed Aitdouad**, pour la qualité remarquable de son accompagnement tout au long de la réalisation de ce projet. Son écoute attentive, sa disponibilité constante et sa grande générosité intellectuelle ont été des atouts essentiels dans l'avancement et la réussite de notre travail.

Son approche méthodique, sa rigueur scientifique et son expertise approfondie nous ont permis de développer nos compétences techniques et d'adopter une démarche de travail structurée et professionnelle. Grâce à ses remarques pertinentes, à ses orientations précieuses et à la qualité de ses retours, nous avons pu surmonter les difficultés rencontrées et mener ce projet à son terme avec confiance et sérieux.

Son sens de la pédagogie, allié à sa patience et à son ouverture d'esprit, a fait de lui un guide précieux, toujours prêt à nous encourager et à nous pousser à donner le meilleur de nous-mêmes. Les échanges réguliers que nous avons eus ont été particulièrement enrichissants, tant sur le plan académique que personnel.

Nous exprimons à **Mr. Mohammed Aitdouad** notre profonde reconnaissance, notre respect et notre sincère gratitude pour son professionnalisme, sa bienveillance et son engagement exemplaire.

Résumé

Ce laboratoire s'inscrit dans le cadre de l'apprentissage des méthodes de conception et de développement des applications informatiques de gestion. Il porte sur la réalisation d'une application dédiée à la **gestion de bibliothèque**, visant à automatiser et optimiser les différentes opérations liées à la gestion des ouvrages, des lecteurs et des emprunts.

La gestion traditionnelle des bibliothèques, souvent basée sur des méthodes manuelles ou semi-automatisées, présente plusieurs inconvénients tels que la perte d'informations, les erreurs de saisie, la difficulté de suivi des livres empruntés et le manque de visibilité sur la disponibilité des ouvrages. Face à ces limites, la mise en place d'un système informatique fiable et structuré devient indispensable.

L'objectif principal de ce projet est de concevoir et développer un système permettant une gestion efficace et cohérente des ressources d'une bibliothèque. L'application développée offre un ensemble de fonctionnalités essentielles, notamment la gestion des livres (ajout, modification, suppression et consultation), la gestion des lecteurs, ainsi que le suivi des opérations d'emprunt et de retour. Le système permet également de vérifier la disponibilité des ouvrages afin d'éviter les conflits et d'assurer une meilleure organisation du fonds documentaire.

La réalisation de ce laboratoire a suivi une démarche méthodologique bien structurée. Dans un premier temps, une analyse des besoins a été menée afin d'identifier les acteurs du système et de définir les besoins fonctionnels et non fonctionnels. Cette analyse a permis de mieux comprendre les attentes des utilisateurs et les contraintes du système. Ensuite, une phase de conception a été réalisée, incluant la définition de l'architecture générale de l'application, la modélisation des entités principales et l'établissement des relations entre elles. Enfin, la phase d'implémentation a permis de traduire cette conception en une application fonctionnelle respectant les exigences initiales.

Des tests fonctionnels ont été effectués afin de vérifier la validité et la fiabilité des fonctionnalités développées. Les résultats obtenus démontrent que l'application répond correctement aux besoins identifiés et assure une gestion efficace des opérations de la bibliothèque.

Ce laboratoire a constitué une expérience enrichissante permettant de mettre en pratique les connaissances théoriques acquises en cours, notamment en analyse des systèmes, en modélisation et en programmation orientée objet. Il a également permis de développer des compétences pratiques en conception et en développement d'applications informatiques de gestion.

Mots-clés : Gestion de bibliothèque, application de gestion, livres, lecteurs, emprunts, systèmes d'information.

Abstract

This laboratory work focuses on the design and development of a **library management application** aimed at automating and optimizing the management of books, readers, and borrowing operations. Traditional library management methods often suffer from several limitations, such as data inconsistencies, manual errors, and difficulties in tracking borrowed books and their availability.

The main objective of this project is to develop an information system that ensures efficient and reliable management of library resources. The developed application provides essential functionalities, including book management (creation, update, deletion, and consultation), reader management, and the handling of borrowing and return operations. The system also ensures the availability control of books in order to avoid conflicts during the borrowing process.

The project was carried out following a structured methodological approach. Initially, a requirements analysis phase was conducted to identify system actors as well as functional and non-functional requirements. This phase was followed by a design stage, during which the system architecture and data model were defined. Finally, the implementation phase resulted in a functional application that meets the defined requirements.

Functional tests were performed to validate the correct behavior of the implemented features. The obtained results demonstrate that the developed application satisfies the requirements of a library management system.

Keywords : Library management, information system, books, readers, borrowing, software application.

Table des figures

1	Logo Doocker	13
2	Logo visual studio	14
3	Logo Odoo	15
4	Logo Python	15
5	Logo Git	16
6	Uml logo	18
7	Use case	19
8	Diagramme de classes	20
9	Diagramme de séquence	21
10	Statut odoo	23
11	Page des Applications Odoo	24
12	Interface de création et de gestion des projets	25
13	Interface de suivi et de gestion des projets	26

Table des matières

1	Contexte général	9
2	Problématique	10
3	Objectifs du projet	10
3.1	Objectifs fonctionnels	10
3.2	Objectifs techniques	11
3.3	Objectifs pédagogiques	11
4	Environnement technique	13
4.1	Docker	13
4.2	Visual Studio Code	14
4.3	Odoo	14
4.4	Python	15
4.5	Git	16
5	Formalisme UML	18
6	Use case	19
7	Diagramme de classes	19
8	Diagramme de séquence	20
9	Odoo tourne	23
10	Page des Applications	24
11	Interface de création d'un projet dans Odoo	25
12	Interface de suivi et de gestion des projets	26

Introduction générale

Dans le cadre de notre formation, ce travail pratique a pour objectif de nous initier à la conception, au développement et à la mise en œuvre d'une application informatique de gestion. Ce projet s'inscrit dans une démarche pédagogique visant à renforcer nos compétences en analyse des besoins, modélisation des systèmes d'information et développement d'applications informatiques structurées.

Avec l'évolution des technologies de l'information, les bibliothèques sont confrontées à des exigences croissantes en matière d'organisation et de gestion des ressources documentaires. La gestion manuelle ou partiellement automatisée des bibliothèques présente plusieurs limites, notamment les erreurs de saisie, la difficulté de suivi des ouvrages empruntés, le manque de visibilité sur la disponibilité des livres et l'absence d'un historique fiable des opérations réalisées. Dans ce contexte, la mise en place d'un système informatique de gestion devient indispensable.

Le projet présenté dans ce rapport consiste à concevoir et développer une application dédiée à la **gestion de bibliothèque**. Cette application a pour but de centraliser les informations relatives aux livres, aux lecteurs et aux opérations d'emprunt et de retour, tout en garantissant la cohérence et la sécurité des données. Le système vise à améliorer l'efficacité du processus de gestion, à faciliter la consultation des informations et à assurer un meilleur contrôle des ressources disponibles.

L'objectif principal de ce travail est de proposer une solution informatique capable de gérer de manière efficace et fiable les différentes composantes d'une bibliothèque. Pour atteindre cet objectif, plusieurs objectifs spécifiques ont été définis :

- Identifier et analyser les besoins fonctionnels et non fonctionnels d'un système de gestion de bibliothèque.
- Concevoir un modèle structuré du système, incluant les entités principales, leurs attributs et les relations qui les lient.
- Implémenter les fonctionnalités essentielles telles que la gestion des livres, des lecteurs, ainsi que le suivi des opérations d'emprunt et de retour.
- Tester et valider le bon fonctionnement de l'application afin de garantir sa conformité aux exigences initiales.

La réalisation de ce projet nous a permis de mettre en pratique les notions théoriques abordées au cours de notre formation, notamment la programmation orientée objet, la modélisation des systèmes d'information et le développement d'applications de gestion. Elle nous a également permis de développer des compétences pratiques en organisation du travail, en résolution de problèmes et en validation des solutions informatiques.

Ainsi, ce rapport présente de manière détaillée les différentes étapes de conception, de réalisation et de validation de l'application de gestion de bibliothèque, tout en mettant en évidence les choix techniques et méthodologiques adoptés au cours du projet.

Chapitre 1

Contexte général du projet

1 Contexte général

Dans le cadre de ce projet, nous avons développé un module personnalisé sous Odoo afin de répondre à un besoin de gestion simple, structurée et centralisée des informations. Odoo est un ERP open-source reconnu pour sa capacité à intégrer de manière cohérente plusieurs fonctions de gestion d'entreprise telles que la gestion des ventes, des achats, des stocks, des ressources humaines, de la comptabilité, de la production, et bien d'autres. Sa modularité et sa flexibilité permettent également la création d'applications spécifiques et personnalisées, adaptées aux besoins particuliers des utilisateurs, sans nécessiter de modification du cœur du système. Cette approche garantit une maintenance plus simple, une évolutivité du système et la possibilité d'adapter l'ERP aux besoins futurs de l'organisation.

Le projet a été réalisé sur la version 16 d'Odoo et déployé dans un environnement Docker, composé de deux conteneurs distincts : un conteneur pour le serveur Odoo et un conteneur pour la base de données PostgreSQL. L'utilisation de Docker présente plusieurs avantages significatifs pour ce type de projet : elle permet de disposer d'un environnement entièrement reproductible, assurant que la configuration logicielle reste identique quel que soit le poste de travail ou le serveur utilisé ; elle offre une mise en place rapide et stable de l'application, sans avoir à gérer manuellement les dépendances et les configurations complexes ; elle assure une isolation complète des dépendances et des services, réduisant ainsi les risques de conflits logiciels ou de dysfonctionnements ; enfin, elle simplifie le déploiement et le redémarrage des services, permettant de maintenir un environnement opérationnel avec un minimum d'interventions.

Le module développé, nommé **“Gestion de Bibliothèque”** a été conçu pour mettre en pratique les concepts fondamentaux du développement dans Odoo. Il permet de manipuler et de comprendre les principaux éléments constitutifs d'un module Odoo :

- La structure standard du module, comprenant les fichiers essentiels tels que qui définissent les informations du module, ses dépendances et son organisation.
- La couche métier (`models`), qui définit les objets métier et les champs associés, permettant de structurer et de gérer les données de manière logique et cohérente.
- La couche de présentation (`views`), implémentée via des fichiers XML, offrant à l'utilisateur des interfaces adaptées sous forme de listes, formulaires et vues spécifiques pour la consultation, la création et la modification des données.
- La gestion de la sécurité et des droits d'accès, configurée via des fichiers CSV ('`security/ir.model.access.csv`'), permettant de définir les groupes d'utilisateurs, les permissions associées et d'assurer une utilisation sécurisée et contrôlée du module.
- L'intégration dans l'interface Odoo grâce aux menus, actions et éléments de navigation, permettant à l'utilisateur d'accéder facilement aux fonctionnalités du module depuis le tableau de bord ou le menu principal.

En plus de ces aspects techniques, le module a été développé dans une perspective pédagogique, visant à familiariser l'utilisateur avec les bonnes pratiques de développement sur Odoo. Cela inclut la compréhension du cycle de vie d'un module, depuis sa conception et son développement jusqu'à son installation, sa mise à jour et son exploitation. Ce projet permet également de se familiariser avec les mécanismes de déploiement dans un environnement conteneurisé Docker, renforçant ainsi la maîtrise de la gestion de services ERP modernes et de l'administration d'applications modulaires.

2 Problématique

- **Erreurs de saisie et doublons** : L'absence de contrôles automatiques favorise les incohérences de données et la redondance des informations entre différents supports.
- **Difficulté d'accès à l'information** : L'éparpillement des données ralentit la recherche, rendant l'accès aux informations critiques laborieux, voire impossible en situation d'urgence.
- **Manque de traçabilité et d'intégrité** : Sans historique des modifications (audit trail), il devient difficile d'identifier l'auteur d'une saisie ou de garantir que le document consulté est la version la plus récente.
- **Perte de productivité** : Le temps consacré à la manipulation manuelle, au classement et à la réconciliation des fichiers réduit considérablement le temps disponible pour les tâches à plus haute valeur ajoutée.
- **Risques liés à la sécurité et à la conformité** : Le stockage sur des supports physiques ou des fichiers locaux limite le contrôle des accès et expose l'organisation à des pertes de données définitives ou à des non-conformités réglementaires (type RGPD).
- **Obstacles à la collaboration** : La dispersion des fichiers empêche le travail simultané et crée des silos d'information, nuisant à la communication entre les différents services.
- **Incapacité d'analyse décisionnelle** : L'absence de centralisation rend l'extraction de statistiques ou de rapports globaux complexe, empêchant une vision stratégique en temps réel.

3 Objectifs du projet

Les objectifs du projet visent à transformer une gestion archaïque en un système structuré et évolutif, déclinés selon trois axes stratégiques :

3.1 Objectifs fonctionnels

- **Centralisation des données** : Assurer la persistance et l'intégrité des informations à travers un cycle CRUD (Create, Read, Update, Delete) complet et sécurisé.
- **Optimisation de l'expérience utilisateur (UX)** : Concevoir des interfaces intuitives (vues Tree et Form) incluant des filtres de recherche et des regroupements pour une extraction rapide de l'information.
- **Intégration écosystémique** : Garantir une navigation fluide en intégrant le module dans l'arborescence native d'Odoo, permettant une interaction cohérente avec les autres fonctionnalités de l'ERP.
- **Automatisation des processus** : Réduire les tâches manuelles répétitives par la mise en place de valeurs par défaut et de contraintes de saisie.

3.2 Objectifs techniques

- **Architecture Modulaire** : Respecter scrupuleusement les standards de développement Odoo (séparation modèles/vues/contrôleurs) pour garantir la maintenabilité et l'évolutivité du code.
- **Sécurisation des données** : Implémenter une gestion fine des privilèges via le fichier `ir.model.access.csv` pour restreindre l'accès aux données sensibles selon le profil utilisateur.
- **Conteneurisation et Déploiement** : Maîtriser l'environnement système en utilisant Docker pour isoler l'instance Odoo et la base de données PostgreSQL, assurant la portabilité du projet.
- **Qualité du code** : Appliquer les conventions de nommage et de structure, `description`, `depends`) pour faciliter le débogage et les futures mises à jour.

3.3 Objectifs pédagogiques

- **Maîtrise de l'ORM Odoo** : Comprendre l'interaction entre les classes Python et la base de données relationnelle sans écrire de requêtes SQL complexes.
- **Compétence en Développement Full-Stack ERP** : Appréhender le lien entre le back-end (Python/Modèles) et le front-end (XML/Vues) spécifique à l'architecture Odoo.
- **Gestion du cycle de vie logiciel** : Expérimenter les phases d'installation, de mise à jour à chaud (`-u module`) et de gestion des dépendances.
- **Culture DevOps** : Se familiariser avec l'orchestration de services via `docker-compose` et la gestion des volumes pour la persistance des données.

Conclusion

En somme, ce chapitre a permis de mettre en exergue les limites critiques d'une gestion manuelle et de définir une feuille de route claire pour le développement du module. En alignant les impératifs fonctionnels sur les bonnes pratiques techniques d'Odoo, ce projet ne se limite pas à une simple numérisation de données ; il vise à instaurer un environnement de travail structuré, sécurisé et évolutif. La maîtrise de cet écosystème, combinée à l'utilisation de Docker, constitue le socle technique sur lequel s'appuiera la phase de conception présentée dans le chapitre suivant.

Chapitre 2

Environnement technologique

4 Environnement technique

Le développement du module `note_partage` repose sur un écosystème technologique moderne, garantissant à la fois la fiabilité du code et la rapidité du déploiement. Ce chapitre détaille les outils et frameworks qui constituent le socle de notre travail.

Nous aborderons notamment l'utilisation de **Docker** pour la conteneurisation et l'isolation des services, l'éditeur **Visual Studio Code** pour l'optimisation du flux de développement, ainsi que l'ERP **Odoop**, plateforme cible de notre solution. La maîtrise de cette architecture est une étape indispensable pour assurer la cohérence technique et la maintenabilité du module au sein de son environnement d'exécution.

4.1 Docker

Docker est une plateforme de conteneurisation open source qui permet de simplifier le développement, le déploiement et l'exécution des applications informatiques. Elle repose sur l'utilisation de conteneurs, qui encapsulent une application ainsi que l'ensemble de ses dépendances, notamment les bibliothèques, les frameworks, les outils système et les fichiers de configuration nécessaires à son bon fonctionnement. Cette approche assure une exécution uniforme et reproductible de l'application, indépendamment de l'environnement d'exécution, qu'il s'agisse d'un poste de développement, d'un serveur de test ou d'un environnement de production. Contrairement aux machines virtuelles traditionnelles, Docker partage le noyau du système d'exploitation hôte, ce qui permet de réduire la consommation des ressources matérielles et d'améliorer les performances ainsi que le temps de démarrage des applications. Par ailleurs, Docker facilite l'automatisation des processus de déploiement et d'intégration continue, favorisant ainsi l'adoption des architectures microservices. Grâce à sa flexibilité, sa portabilité et sa légèreté, Docker est devenu un outil incontournable dans le domaine du cloud computing et du développement logiciel moderne, contribuant à renforcer la fiabilité, la scalabilité et la maintenabilité des applications.

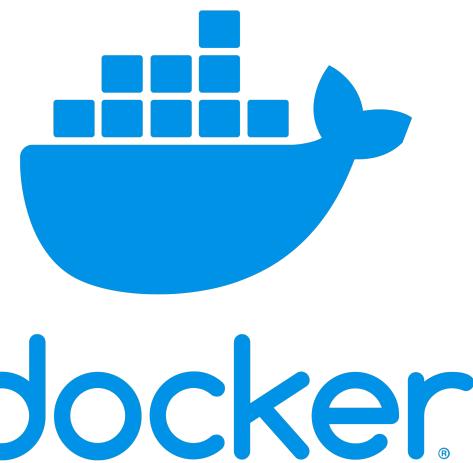


FIGURE 1 – Logo Docker

4.2 Visual Studio Code

Visual Studio Code est un éditeur de code source léger, open source et multiplateforme, développé par Microsoft, destiné à faciliter le développement d'applications modernes. Il prend en charge un large éventail de langages de programmation tels que Java, Python, C/C++, JavaScript et bien d'autres, ce qui en fait un outil polyvalent adapté à différents domaines du développement logiciel. Visual Studio Code propose de nombreuses fonctionnalités avancées, notamment la coloration syntaxique intelligente, l'auto-complétion basée sur IntelliSense, le débogage intégré, ainsi que la gestion du contrôle de version grâce à l'intégration native de Git. Son système d'extensions permet d'ajouter facilement de nouvelles fonctionnalités, comme des frameworks, des outils de tests ou des environnements de conteneurisation tels que Docker. De plus, son interface intuitive et sa légèreté garantissent de bonnes performances même sur des machines aux ressources limitées. Grâce à sa flexibilité, sa richesse fonctionnelle et sa facilité de prise en main, Visual Studio Code est aujourd'hui un outil incontournable pour améliorer la productivité et la qualité du développement logiciel.

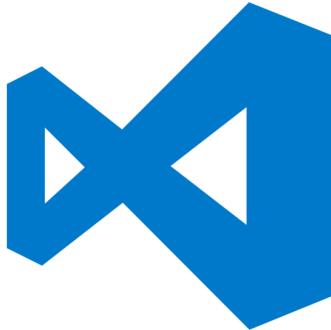


FIGURE 2 – Logo visual studio

4.3 Odoo

Odoo est une suite de logiciels de gestion intégrée (ERP) open source qui permet aux entreprises de centraliser et de gérer efficacement l'ensemble de leurs processus métiers au sein d'une seule plateforme. Il propose un large ensemble de modules couvrant différents domaines tels que la gestion commerciale, la comptabilité, les ressources humaines, la gestion des stocks, la production, la relation client (CRM) et le commerce électronique. Grâce à son architecture modulaire et flexible, Odoo peut être facilement personnalisé et adapté aux besoins spécifiques de chaque organisation. Développé principalement en Python et reposant sur des technologies web modernes, Odoo offre une interface utilisateur intuitive et accessible via un navigateur. Sa nature open source, combinée à une grande communauté de développeurs, permet une évolution continue du système et une intégration aisée avec d'autres applications, faisant d'Odoo une solution ERP complète et performante pour les entreprises de toutes tailles.



FIGURE 3 – Logo Odoo

4.4 Python

Python est un langage de programmation interprété, open source et de haut niveau, reconnu pour sa simplicité, sa lisibilité et sa flexibilité. Il se distingue par une syntaxe claire, concise et proche du langage naturel, ce qui facilite grandement son apprentissage pour les débutants tout en accélérant le développement des applications pour les professionnels. Python est utilisé dans une grande variété de domaines, tels que le développement web avec des frameworks comme Django et Flask, l'automatisation des tâches répétitives, l'analyse et la visualisation de données, l'intelligence artificielle, le machine learning, le traitement d'images et le calcul scientifique. Grâce à son vaste écosystème de bibliothèques et de frameworks (NumPy, Pandas, Matplotlib, TensorFlow, PyTorch, etc.), il permet de créer des applications performantes, modulaires et facilement maintenables avec un minimum de code. De plus, Python est multiplateforme et fonctionne sur la plupart des systèmes d'exploitation, ce qui assure une grande portabilité des programmes. Sa communauté active et dynamique contribue également à l'enrichissement continu du langage, à travers des mises à jour régulières, la création de nouvelles bibliothèques et un support étendu pour les développeurs de tous niveaux. Grâce à toutes ces caractéristiques, Python est devenu un langage incontournable dans le développement logiciel moderne et continue de gagner en popularité dans le monde entier, tant pour des projets académiques que pour des applications professionnelles complexes.



FIGURE 4 – Logo Python

4.5 Git

Git Bash est une interface en ligne de commande pour Windows qui permet d'utiliser les fonctionnalités de Git, le système de contrôle de version distribué, dans un environnement proche de celui de Linux. Git Bash fournit un terminal Unix complet sur Windows, permettant d'exécuter des commandes Git ainsi que des commandes shell classiques comme ls, cd, rm ou mkdir. Cet outil est particulièrement utile pour les développeurs travaillant sur des projets collaboratifs, car il facilite la gestion du code source, le suivi des modifications, la fusion des branches et la résolution des conflits. Git Bash est souvent utilisé avec des plateformes de gestion de code telles que GitHub, GitLab ou Bitbucket, offrant ainsi un environnement flexible pour le développement moderne. Sa combinaison de Git et d'un terminal Unix rend l'expérience de développement plus cohérente pour les utilisateurs venant de systèmes Linux ou macOS, tout en restant accessible aux développeurs Windows.



FIGURE 5 – Logo Git

Conclusion

En résumé, Docker, Visual Studio Code, Odoo, Python et Git Bash sont des outils essentiels dans le développement logiciel moderne. Docker assure la portabilité des applications, Visual Studio Code offre un environnement de développement flexible, Odoo centralise la gestion des processus métiers, Python fournit un langage polyvalent et puissant, et Git Bash facilite la gestion du code source. Ensemble, ils permettent de créer des applications robustes et maintenables tout en améliorant la productivité et la collaboration dans les projets logiciels.

Chapitre 3

Analyse et conception

5 Formalisme UML

UML (Unified Modeling Language) est un langage de modélisation graphique normalisé, basé sur l'utilisation de pictogrammes, initialement conçu pour faciliter la conception et l'analyse des systèmes orientés objet dans le domaine du génie logiciel. Toutefois, son champ d'application s'est rapidement élargi, et UML peut aujourd'hui être utilisé pour modéliser tout type de système, qu'il soit informatique, organisationnel, industriel ou même métier.

UML est devenu un standard international défini et maintenu par l'Object Management Group (OMG). Depuis novembre 2007, l'OMG diffuse la version UML 2.1.2, qui a apporté des améliorations notables en termes de précision sémantique et de capacité d'expression. L'organisme poursuit encore ses travaux afin d'enrichir et d'adapter UML aux besoins évolutifs des systèmes modernes et complexes.

Le langage UML propose un ensemble riche de diagrammes complémentaires, chacun permettant de représenter le système sous un angle spécifique. Parmi eux, le diagramme de classes occupe une place centrale, car il décrit la structure statique du système, les entités principales, leurs attributs, leurs méthodes ainsi que les relations qui les lient. Selon les méthodologies de développement adoptées, telles que l'Unified Process (UP), l'analyse peut également commencer par des diagrammes de cas d'utilisation (Use Case), qui permettent de capturer les besoins fonctionnels du système du point de vue des acteurs et des utilisateurs finaux.

UML se compose de plusieurs sous-ensembles complémentaires :

- **Les vues** : elles représentent le système selon différents points de vue (organisationnel, fonctionnel, dynamique, temporel, architectural ou logique). Chaque vue met en évidence un aspect particulier du système, et leur combinaison permet d'obtenir une description globale, cohérente et complète.
- **Les diagrammes** : ce sont des représentations graphiques concrètes des vues. Ils traduisent les concepts abstraits du système sous forme visuelle afin de faciliter la compréhension, la communication entre les parties prenantes et la documentation du projet.



FIGURE 6 – Uml logo

6 Use case

Le diagramme de cas d'utilisation présente de manière structurée les fonctionnalités principales du module Gestion de Bibliothèque et les interactions entre les acteurs et le système. Deux acteurs sont représentés : le Bibliothécaire, qui dispose de l'ensemble des droits d'administration et peut gérer les livres, les membres, les emprunts ainsi que consulter le catalogue, et le Membre, qui dispose de droits limités lui permettant uniquement de consulter les livres disponibles et de gérer ses propres emprunts. Les cas d'utilisation mettent en évidence les processus essentiels de la bibliothèque, tels que l'ajout, la modification ou la suppression de livres et de membres, ainsi que l'enregistrement des emprunts et des retours. Ce diagramme permet non seulement de visualiser les rôles et responsabilités de chaque acteur, mais aussi de clarifier les interactions fonctionnelles avec le système, facilitant ainsi la conception et le développement du module dans Odoo, tout en respectant les principes de gestion des droits d'accès et de sécurité. Il constitue un outil précieux pour comprendre l'architecture fonctionnelle et organiser le développement autour des besoins des utilisateurs.

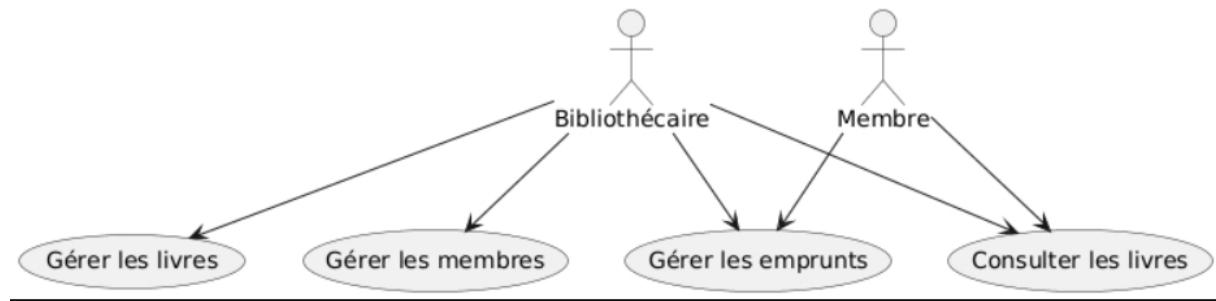


FIGURE 7 – Use case

7 Diagramme de classes

Le diagramme de classes UML représente la structure statique du module Gestion de Bibliothèque, en montrant les entités principales, leurs attributs et leurs relations. Les classes principales incluent Livre, Auteur, Membre, Emprunt et Catégorie, chacune contenant les informations essentielles pour gérer les données de la bibliothèque. Par exemple, la classe Livre contient des attributs tels que le titre, l'ISBN, la date de publication et le nombre d'exemplaires, et est liée à un Auteur et à une Catégorie. La classe Emprunt relie un Livre à un Membre, en enregistrant les dates d'emprunt et de retour ainsi que l'état du prêt. Ce diagramme illustre clairement les dépendances et associations entre les différentes entités, facilitant la compréhension de la structure de données et la conception du module Odoo. Il constitue un outil essentiel pour la modélisation de la base de données, la gestion des relations entre objets et l'organisation des fonctionnalités, permettant ainsi un développement cohérent et structuré du module.

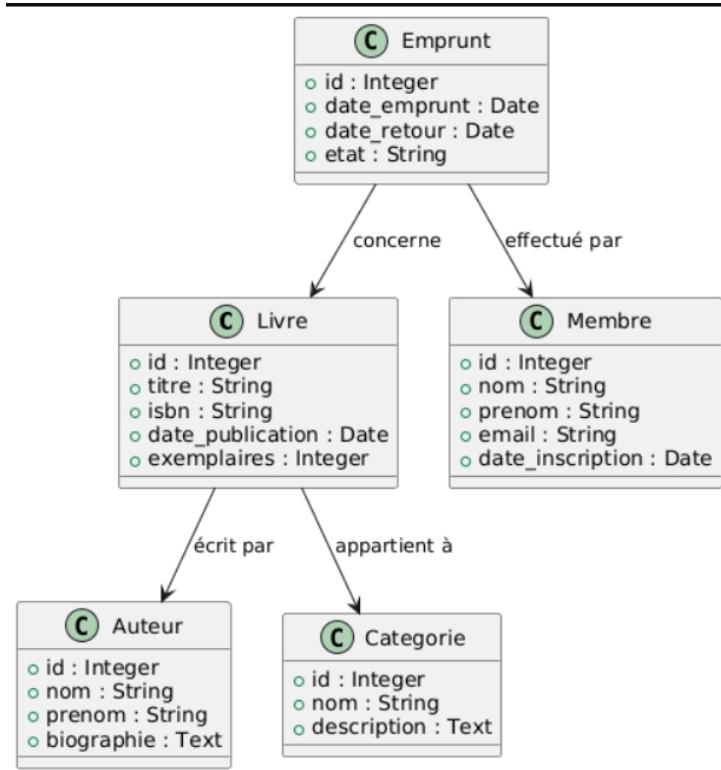


FIGURE 8 – Diagramme de classes

8 Diagramme de séquence

Le diagramme de séquence UML illustre le déroulement dynamique d'un processus dans le module Gestion de Bibliothèque, en mettant en évidence l'ordre précis des interactions entre les acteurs et les objets du système. Il permet de visualiser comment les différentes entités collaborent pour accomplir une fonctionnalité spécifique et de suivre le flux des informations dans le temps. Par exemple, lors de l'emprunt d'un livre, le Membre envoie une demande au Bibliothécaire, qui vérifie la disponibilité du Livre dans la base de données et crée un enregistrement correspondant dans la classe Emprunt, avant de confirmer la transaction et de communiquer la date de retour au membre. Ce type de diagramme rend explicite la séquence des opérations, clarifie les responsabilités et interactions de chaque acteur et de chaque objet, et permet de détecter d'éventuels points de blocage ou améliorations dans le processus. Il est particulièrement utile pour comprendre le comportement fonctionnel du module et la manière dont les fonctionnalités sont orchestrées dans Odoo, notamment pour la gestion des emprunts, des retours et des mises à jour des données. De plus, ce diagramme constitue un outil précieux pour la conception et la documentation technique, facilitant le développement, la validation, la maintenance et l'optimisation des processus tout en garantissant que les flux d'information respectent la logique métier et les droits d'accès définis.

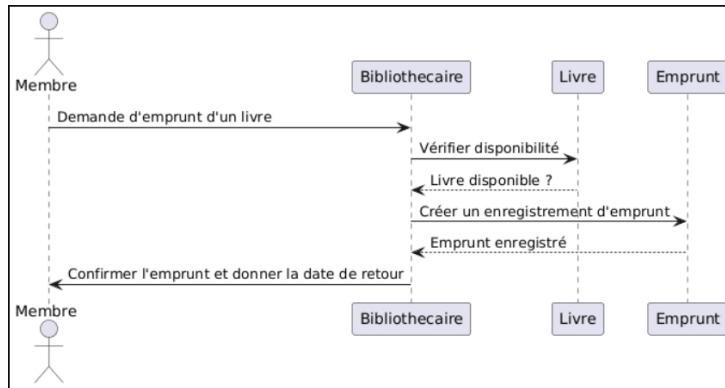


FIGURE 9 – Diagramme de séquence

Conclusion

En conclusion, les diagrammes UML du module Gestion de Bibliothèque permettent de représenter clairement la structure des données, les fonctionnalités et les interactions entre les acteurs et le système. Le diagramme de classes montre les entités et leurs relations, le diagramme de cas d'utilisation illustre les rôles et droits d'accès, et le diagramme de séquence décrit le déroulement des processus comme l'emprunt d'un livre. Ces outils facilitent la conception, le développement et la compréhension globale du module dans Odoo.

Chapitre 4

Réalisation

Introduction

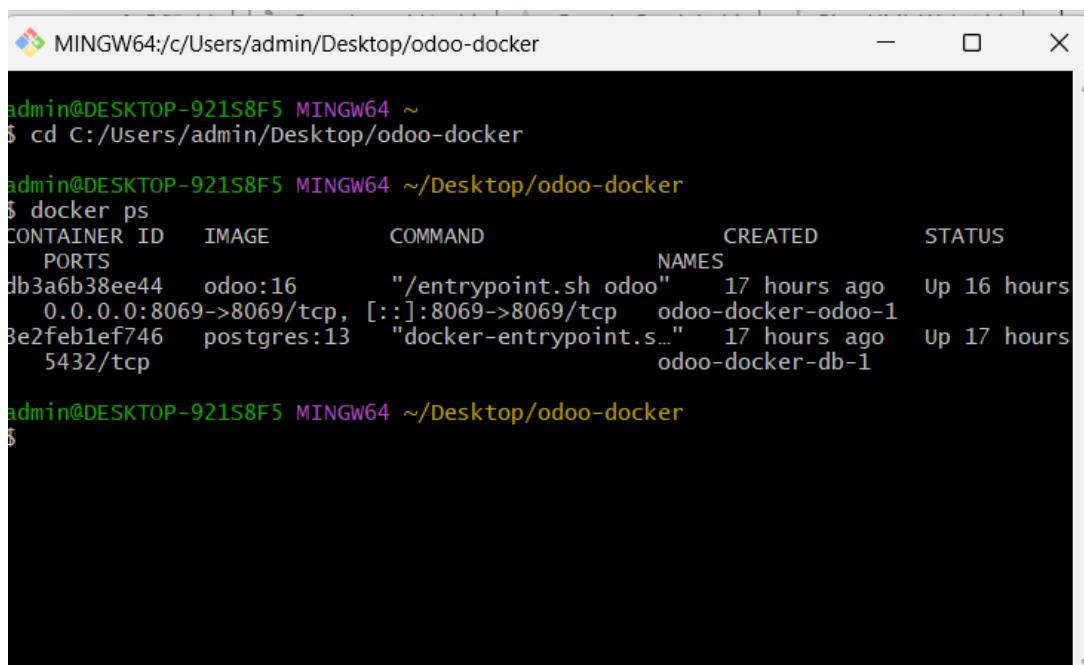
Ce chapitre présente la réalisation technique du module Gestion de Bibliothèque développé sous Odoo 16. Il décrit les principales étapes mises en œuvre pour passer du besoin initial à une solution fonctionnelle intégrée à l'ERP, en suivant l'architecture standard d'Odoo et les bonnes pratiques de développement.

Nous détaillerons d'abord l'environnement de travail basé sur Docker (conteneur Odoo et conteneur PostgreSQL), choisi pour garantir un déploiement simple et reproductible. Ensuite, nous expliquerons la structure du module (manifest.py, organisation des dossiers models/, views/, security/) ainsi que la mise en place des composants essentiels : définition des modèles (logique métier), création des vues XML (listes et formulaires), intégration via les menus/actions, et configuration des droits d'accès pour sécuriser l'utilisation.

Enfin, ce chapitre conclut sur la procédure d'installation et de mise à jour du module dans Odoo (mise à jour de la liste des applications, installation/upgrade), afin de valider le bon fonctionnement de l'application et son intégration dans l'interface utilisateur. Etape1 : vérifier que odoo

9 Odoo tourne

Le bon fonctionnement d'Odoo a été vérifié en contrôlant le démarrage des services et l'accessibilité de l'application. L'accès à l'interface web via le navigateur sur le port 8069 a permis de confirmer que le serveur Odoo était correctement lancé. En parallèle, l'état des conteneurs Docker Odoo et PostgreSQL a été vérifié afin de s'assurer qu'ils étaient actifs et fonctionnaient sans erreur. L'analyse des journaux d'exécution a également permis de confirmer l'absence de problèmes lors du démarrage. Ces vérifications garantissent que l'environnement Odoo est opérationnel et prêt pour l'installation et l'utilisation du module Gestion de Bibliothèque.



The screenshot shows a terminal window titled 'MINGW64:/c/Users/admin/Desktop/odoo-docker'. The user is in their home directory and changes to the 'odoo-docker' folder. Then, they run the 'docker ps' command to list running containers. The output shows two containers:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
db3a6b38ee44	odoo:16	"/entrypoint.sh odoo"	17 hours ago	Up 16 hours
0.0.0.0:8069->8069/tcp, [::]:8069->8069/tcp				odoo-docker-odoo-1
3e2feb1ef746	postgres:13	"docker-entrypoint.s..."	17 hours ago	Up 17 hours
		5432/tcp		odoo-docker-db-1

FIGURE 10 – Statut odoo

10 Page des Applications

Cette interface correspond à la page des applications (*Apps*) d’Odoo 16, qui constitue un élément central dans l’administration et la gestion fonctionnelle de la plateforme. Elle offre à l’administrateur une vue globale de l’ensemble des modules disponibles, incluant à la fois les modules standards fournis par Odoo et les modules personnalisés développés pour répondre à des besoins spécifiques. À travers cette interface, il est possible de rechercher des applications, de consulter leurs informations, ainsi que de procéder à leur installation, activation, désactivation ou mise à jour, assurant ainsi une gestion cohérente et centralisée des fonctionnalités du système.

Le module personnalisé **Gestion de Bibliothèque (gestion_bibliotheque)** apparaît dans la liste des applications disponibles, ce qui confirme qu’il a été correctement développé et intégré dans l’environnement Odoo. Cette reconnaissance par le système indique que la structure du module respecte les normes et conventions imposées par Odoo, notamment l’organisation des répertoires, la présence des fichiers obligatoires (`__init__.py`, `__manifest__.py`) ainsi que la définition correcte des métadonnées du module. La visibilité du module dans cette interface démontre également que l’environnement Odoo a bien détecté le module après son ajout dans le répertoire des modules personnalisés.

La présence du bouton *Activate* indique que le module n’est pas encore installé dans la base de données et qu’il peut être activé par l’administrateur en une seule action. Cette phase constitue une étape clé dans le cycle de développement d’un module Odoo, car elle permet de vérifier la validité du fichier `__manifest__.py`, la compatibilité avec la version d’Odoo utilisée, ainsi que la prise en compte correcte des dépendances définies. Elle garantit également que l’environnement technique est stable et prêt à accueillir de nouvelles fonctionnalités sans provoquer d’erreurs lors de l’installation.

Une fois le module activé, Odoo procède automatiquement au chargement de l’ensemble de ses composants, notamment les modèles de données, les vues XML, les menus, ainsi que les règles de sécurité et de gestion des droits d’accès. Le module devient alors pleinement opérationnel et accessible via l’interface web d’Odoo. Cette étape marque la transition entre la phase de conception et de développement technique du module et sa phase d’exploitation fonctionnelle.

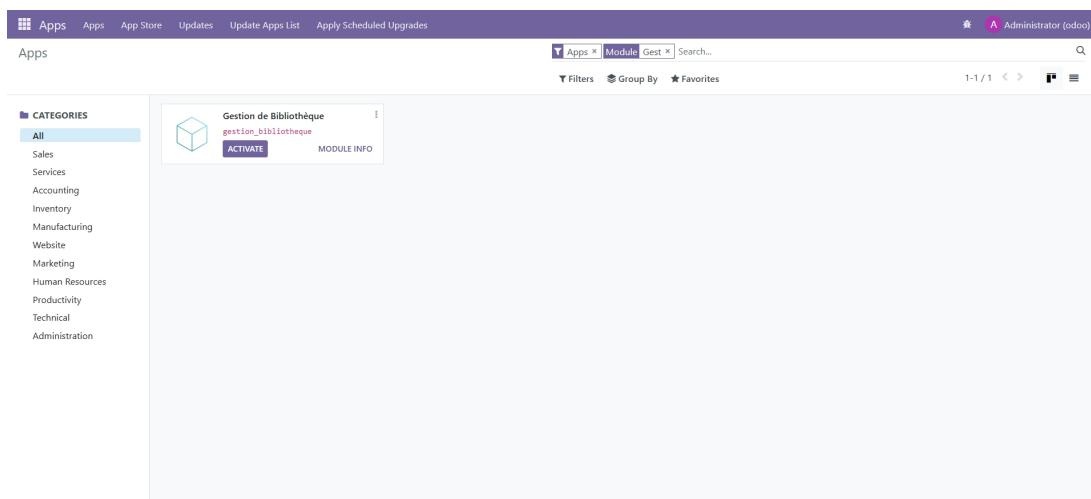


FIGURE 11 – Page des Applications Odoo

11 Interface de création d'un projet dans Odoo

Cette image illustre l'interface du progiciel de gestion intégré **Odoo**, et plus précisément le module « Projets TP », qui constitue un outil central pour la création, la gestion et le suivi des projets. L'écran présenté correspond à la vue formulaire dédiée à l'ajout d'un nouveau projet, offrant à l'utilisateur un espace structuré pour la saisie et l'organisation des informations essentielles liées au projet. Cette vue permet de garantir une cohérence des données et facilite leur exploitation tout au long du cycle de vie du projet.

Parmi les champs proposés, le champ relatif au nom du projet joue un rôle fondamental dans l'identification claire et explicite du travail à réaliser. Le champ responsable du projet permet d'assigner une personne chargée de la supervision, de la coordination des tâches et du suivi de l'avancement, favorisant ainsi une meilleure répartition des responsabilités. La date de début constitue un élément clé pour la planification temporelle du projet, puisqu'elle sert de référence pour l'organisation des activités et le respect des délais. Le statut du projet informe sur son état d'avancement, tel que *en cours*, *terminé* ou *en attente*, ce qui permet d'avoir une vision globale et synthétique de l'ensemble des projets gérés au sein du système. Un champ de description est également mis à disposition afin de fournir des informations complémentaires concernant les objectifs, le contexte général, les fonctionnalités attendues ou encore les contraintes du projet.

En complément des champs de saisie, l'interface propose plusieurs boutons d'action accessibles à l'utilisateur connecté, offrant la possibilité d'effectuer diverses opérations telles que la création de nouveaux projets, la modification ou la mise à jour des informations existantes, ainsi que la gestion des données associées. Ces fonctionnalités contribuent à rendre l'outil interactif et flexible, tout en améliorant l'efficacité du processus de gestion. L'organisation visuelle claire, intuitive et structurée de cette interface renforce son ergonomie et facilite sa prise en main, même pour des utilisateurs novices.

Ainsi, le module « Projets TP » s'inscrit dans une logique de centralisation et de structuration de la gestion des projets au sein de la plateforme Odoo. Il permet d'améliorer l'organisation des informations, d'assurer une traçabilité efficace des données et de soutenir un suivi rigoureux et cohérent des projets, aussi bien dans un cadre académique, tel que les travaux pratiques et les projets de fin d'études, que dans un contexte professionnel.

The screenshot shows the Odoo Project Management interface. At the top, there's a dark header bar with the Odoo logo and the text "Gestion des Projets". Below it, a sub-header bar shows "Projets TP / New" and has "Action" and "New" buttons. The main area is a white form for creating a new project. It contains the following fields:

Nom du projet ?	Gestion de bibliothèques
Responsable ?	Oubeghdad Arwa
Date de début ?	01/22/2026
Statut ?	En cours
Description ?	Siiir

FIGURE 12 – Interface de création et de gestion des projets

12 Interface de suivi et de gestion des projets

Cette interface correspond au module **Gestion des Projets** de l'ERP **Odoo** et constitue un élément central pour l'organisation, la planification et le suivi des projets au sein de l'organisation. Elle offre une vue globale, synthétique et structurée de l'ensemble des projets enregistrés, présentés sous forme de tableau afin de faciliter la lecture, l'analyse et la prise de décision. Chaque projet est décrit à travers des informations essentielles telles que le nom du projet, le responsable chargé de sa supervision, la date de début ainsi que le statut d'avancement, ce qui permet aux utilisateurs d'avoir une vision immédiate et précise sur l'état général des projets en cours et terminés.

Cette interface joue un rôle important dans la gestion opérationnelle des projets, car elle permet de centraliser toutes les informations liées aux activités dans un seul espace. Elle facilite le suivi de l'évolution des projets, le contrôle du respect des délais et l'identification rapide des projets nécessitant une attention particulière. Le statut d'avancement permet notamment de distinguer les projets actifs de ceux déjà achevés, contribuant ainsi à une meilleure organisation du travail et à une répartition plus efficace des ressources.

Par ailleurs, l'interface offre une gestion complète du cycle de vie des projets, depuis leur création jusqu'à leur finalisation. Grâce aux fonctionnalités intégrées, l'utilisateur peut créer de nouveaux projets, consulter les projets existants, mettre à jour leurs informations et assurer un suivi continu de leur progression. Les outils de recherche, de filtrage et de regroupement permettent d'accéder rapidement aux données souhaitées et d'organiser les projets selon différents critères, tels que le responsable ou le statut. Ces fonctionnalités améliorent la productivité des utilisateurs, renforcent la coordination entre les différents intervenants et favorisent une gestion plus efficace, transparente et structurée des projets au sein de l'organisation.

Nom du projet	Responsable	Date de début	Statut
Gestion de bibliothèques	Oubeghdad Arwa	01/22/2026	En cours
les emprunts	oubeghdad	01/23/2026	Terminé
projet odoo	arwa oubeghdad	02/18/2026	Terminé

FIGURE 13 – Interface de suivi et de gestion des projets

Conclusion

En conclusion, le chapitre de réalisation a permis de présenter toutes les étapes de développement du module **gestion_bibliotheque**. De la configuration de l'environnement technologique à la structuration des dossiers, en passant par la définition des modèles, la conception des vues et la gestion des droits d'accès, chaque étape a contribué à construire un outil fonctionnel et efficace. Ce travail a non seulement permis de mettre en pratique les connaissances acquises sur Odoo et le développement modulaire.

Conclusion Générale et Perspectives

Le présent projet a permis le développement d'un module `gestion_bibliotheque` pour l'ERP Odoo, visant à centraliser et organiser efficacement les activités d'une bibliothèque. À travers ce travail, nous avons pu appliquer les concepts théoriques étudiés en ERP, en développement modulaire et en gestion de projets informatiques. La mise en place d'un environnement technologique complet, incluant Docker, Visual Studio Code, Git et Odoo, a facilité le développement, le test et la gestion des versions du module.

La réalisation du module a consisté à structurer les dossiers, définir les modèles (livres, auteurs, emprunteurs, prêts), concevoir les vues et gérer les droits d'accès. Ces étapes ont permis de créer un outil fonctionnel, intuitif et adaptable aux besoins de la bibliothèque, capable de suivre et d'organiser toutes les informations relatives aux ressources et aux utilisateurs. Ce projet a également renforcé nos compétences techniques et organisationnelles, en particulier dans le développement Odoo et la gestion d'un environnement de travail complet.

En termes de perspectives, plusieurs améliorations peuvent être envisagées. Il serait intéressant d'intégrer un système de notifications pour les retards de prêts, de mettre en place des statistiques avancées sur l'utilisation des ressources, ou encore d'ajouter une interface mobile pour faciliter l'accès aux informations depuis n'importe quel appareil. De plus, l'extension du module pour gérer plusieurs bibliothèques ou la connexion avec d'autres modules Odoo (comptabilité, ressources humaines) pourrait améliorer encore davantage son efficacité et sa valeur ajoutée pour l'organisation.

Ainsi, ce projet constitue une base solide pour la gestion numérique d'une bibliothèque et ouvre la voie à de nombreuses extensions et optimisations futures, permettant de répondre aux besoins évolutifs des utilisateurs et de l'organisation.