

UNIVERSITE IBA DER THIAM de THIES
(U.I.D.T)



Établissement -UFR des Sciences Économiques et
Sociales-UFR des Sciences et Technologie
Département Commun - Management des
Organisations - Mathématiques - Informatique
Sciences Économiques et de Gestion

PROJET BASE DE DONNEES

MBERY SENE GNING
OUMOU BALDE
DIEYNABA AMADOU BA
MOR MBENGUE

MASTER 1 SCIENCE DES DONNÉES ET APPLICATIONS:
OPTION IIA

2 novembre 2024

**SUJET :MISE EN PLACE D'UNE APPLICA-
TION DE GESTION DES PROPRIETES IM-
MOBILIERES**

1 GENERALITE ET ANALYSE DES BESOINS

1.1 INTRODUCTION

1.1.1 CONTEXTE

La gestion des propriétés immobilières, qu'elles soient résidentielles, commerciales ou mixtes, est une tâche complexe nécessitant une coordination minutieuse des opérations quotidiennes. Avec la croissance du marché immobilier et l'augmentation des attentes des locataires, les gestionnaires immobiliers doivent faire face à un volume croissant de données et de processus à gérer. Les méthodes traditionnelles, souvent basées sur des fichiers manuels et des systèmes disparates, deviennent insuffisantes pour répondre aux exigences modernes de précision, de rapidité et d'efficacité. Dans ce contexte, la mise en place d'une application de gestion des propriétés immobilières s'avère être une solution stratégique pour optimiser les opérations et améliorer la qualité du service.

1.1.2 PROBLEMATIQUES

1. Fragmentation des Informations : Les informations relatives aux propriétés, locataires, baux, et transactions sont souvent dispersées entre plusieurs systèmes ou formats, rendant la gestion et l'accès aux données inefficaces.
2. Gestion Inefficace des Taches : Le traitement manuel des tâches telles que la gestion des demandes de maintenance, la facturation et le suivi des paiements entraîne des erreurs et des délais dans la réponse aux besoins des locataires.
3. Suivi des Performances et Analyse : Il est difficile de suivre et d'analyser en temps réel les indicateurs clés tels que le taux d'occupation, les revenus, et les incidents de maintenance, ce qui peut affecter la prise de décision et la performance globale.
4. Satisfaction des Locataires : Une réponse réactive aux demandes des locataires et un manque d'outils pour recueillir des retours peuvent nuire à la satisfaction et entraîner une augmentation des taux de vacance.

1.1.3 OBJECTIFS DU PROJET

1. Centralisation des Données :

Objectif : Développer une application centralisée qui regroupe toutes les informations pertinentes sur les propriétés, les locataires, les baux et les transactions en un seul système intégré.

Bénéfice : Améliorer l'accès aux données, réduire les erreurs et faciliter une gestion cohérente et informée.

2. Automatisation des Processus :

Objectif : Automatiser les processus de gestion tels que la collecte des paiements, la gestion des demandes de maintenance, et la génération de rapports financiers.

Bénéfice : Réduire les tâches manuelles, minimiser les erreurs, et accélérer le traitement des opérations.

3. Suivi et Analyse en Temps Réel :

Objectif : Intégrer des fonctionnalités de suivi en temps réel et de reporting dans l'application pour surveiller les indicateurs de performance clés comme le taux d'occupation, les revenus locatifs, et les demandes de maintenance.

Bénéfice : Fournir une visibilité complète sur les performances du portefeuille immobilier et faciliter une prise de décision rapide et basée sur des données concrètes.

4. Amélioration de l'Expérience Locataire :

Objectif : Offrir des fonctionnalités permettant aux locataires de soumettre des demandes, de payer leur loyer en ligne, et de recevoir des notifications sur l'état de leurs demandes.

Bénéfice : Augmenter la satisfaction des locataires, réduire les temps de réponse aux demandes, et améliorer la fidélisation des locataires.

1.2 ANALYSES DES BESOINS

L'étape de l'analyse des besoins est l'une des étapes les plus importantes à considérer. En effet si les besoins sont mal spécifiés et exprimés tout ce qui suit sera mal traité d'où l'importante attribuée à cette activité. La spécification des besoins constitue la partie primordiale au début de chaque démarche de développement il a pour but de veiller à la mise en place d'une application adéquate, et la description générale des fonctionnalités du système. Dans cette partie, l'objectif est donc d'énoncer les besoins attendus du futur système à concevoir. Ces besoins sont scindés en deux ; ceux fonctionnels c'est à dire ceux indispensable au fonctionnement du système et ceux non fonctionnels à savoir ceux qui sont facultatifs.

1.2.1 Spécification des besoins fonctionnels

Pour ces besoins fonctionnels, nous allons utiliser les cas d'utilisation

- ▶ Gestion des Propriétés
- ▶ Gestion des utilisateurs
- ▶ Gestion des Locataires
- ▶ Gestion des paiements
- ▶ Gestion des baux

1.2.2 Spécification des besoins non fonctionnels

À part les besoins fondamentaux le futur système doit répondre aux critères suivants :

- la rapidité de traitement : vu le nombre important des transactions par jour, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel
- la performance un logiciel doit avant tout être performant c'est à dire ses fonctionnalités, répond à toutes les exigences des utilisateurs d'une manière optimale
- la convivialité : le logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviale c'est à dire simples, ergonomiques et adaptées à l'utilisateur
- l'application doit signaler les erreurs par des messages d'erreur.

En effet, l'objectif de cette partie était de décrire notre projet et faire une étude de l'existant (analyse des besoins) ce qui va nous permettre de passer à notre deuxième partie qui est la phase de modélisation du système

2 MODÉLISATION DU SYSTÈME

2.1 GÉNÉRALITÉ SUR L'UML

UML (Unified Modeling Language) ou (Langage de Modélisation Unifié en français) est un langage de modélisation orienté objet. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent

la limite, la structure et le comportement du système et des objets qui s'y trouvent. L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML.

2.2 REPRÉSENTATION DES DIAGRAMMES

Les différents types de diagrammes

On note deux catégories de diagrammes : les diagrammes de structure et les diagrammes de comportements.

Diagrammes de structure

Les diagrammes de structure représentent les éléments du système, leurs propriétés et leurs relations entre eux :

- Classe
- Composants
- Déploiement
- Objets
- Paquetage
- Structure composite

Diagramme de comportement

Les diagrammes de comportement représentent les processus et les interactions entre les objets :

- Cas d'utilisation
- Activité
- État
- Séquence
- Communication
- Présentation des interactions
- Diagramme de temps

Nous allons utiliser un diagramme de comportement à savoir le cas d'utilisation et un diagramme de structure à savoir le diagramme de classe pour modéliser le système

Les différents acteur du système

Administrateurs

Accès complet à toutes les fonctionnalités et données du système. Responsabilités : Créer et gérer des comptes d'utilisateurs, attribuer des rôles et permissions. Configurer les paramètres du système (comme les politiques de sécurité et de confidentialité). Gérer les sauvegardes et les restaurations des

données. Superviser les mises à jour et la maintenance du système. Générer des rapports d'activité du système et de performance.

Gestionnaires de Propriétés

Accès aux informations des propriétés, locataires, baux, paiements, et entretiens. Responsabilités : Ajouter et mettre à jour les informations sur les propriétés (adresse, caractéristiques, disponibilité). Gérer les contrats de bail (création, modification, renouvellement). Superviser les demandes d'entretien et organiser les réparations nécessaires. Assurer le suivi des paiements de loyer et des charges. Générer des rapports sur les propriétés, les baux, et les paiements.

Comptables

Accès aux informations financières, aux paiements, et aux factures. Responsabilités : Enregistrer et suivre les paiements de loyers et autres transactions financières. Gérer les dépôts de garantie et les remboursements. Préparer les états financiers et les rapports de performance financière. Assurer la conformité avec les régulations fiscales et comptables.

Agents Immobiliers

Accès aux informations sur les propriétés disponibles à la location ou à la vente. Responsabilités : Inscrire de nouvelles propriétés et mettre à jour les informations existantes. Organiser et coordonner les visites des propriétés pour les potentiels locataires ou acheteurs. Gérer les négociations de contrats de location ou de vente. Maintenir la communication avec les locataires potentiels et les propriétaires. Suivre l'état des demandes et des transactions immobilières.

Locataires

Accès : Accès à leurs informations personnelles, contrats de bail, et informations de paiement. Responsabilités : Consulter les détails de leur bail, les conditions et les dates importantes. Effectuer des paiements de loyer en ligne et consulter l'historique des paiements. Soumettre des demandes de maintenance ou signaler des problèmes dans la propriété. Recevoir et consulter les notifications relatives aux réparations ou aux changements de conditions de bail. Mettre à jour leurs informations personnelles (adresse, coordonnées).

Techniciens d'entretien

Accès : Accès aux demandes d'entretien et aux informations liées aux réparations. Responsabilités : Consulter et accepter les demandes d'entretien assignées. Effectuer les réparations et entretiens selon les demandes reçues. Mettre à jour le statut des travaux (en cours, terminé) et ajouter des notes ou des observations. Communiquer avec les gestionnaires de propriétés sur l'état des réparations et des besoins supplémentaires. Signaler les problèmes nécessitant des ressources supplémentaires ou des pièces de rechange. Chaque utilisateur a des responsabilités spécifiques qui garantissent que toutes les

facettes de la gestion des propriétés sont couvertes de manière efficace.

2.2.1 le diagramme de cas d'utilisation

Est un type de diagramme utilisé en UML pendant la phase d'analyse d'un projet pour identifier la fonctionnalité du système. Il sert à schématiser l'expression des besoins il se présente sous deux formes : schéma regroupant les activités d'un acteur ; et puis pour chaque activité d'un acteur on peut détaillé à travers un tableau les actions qui sont liés à cette activité et ce tableau permet de décrire clairement les différents actions qui devront avoir lieux sur une interface. Il résume certaines des relations entre les cas d'utilisation, les acteurs et les systèmes. Il correspond à une représentation globale (schéma) ou détaillées (tableau) des activités d'un acteur (vision du système par l'acteur).

Ici nous allons représenter quelques diagramme de cas d'utilisation de notre futur système à savoir le diagramme de cas d'utilisation «Authentification», le diagramme pour l'administrateur et le gestionnaire de propriétés

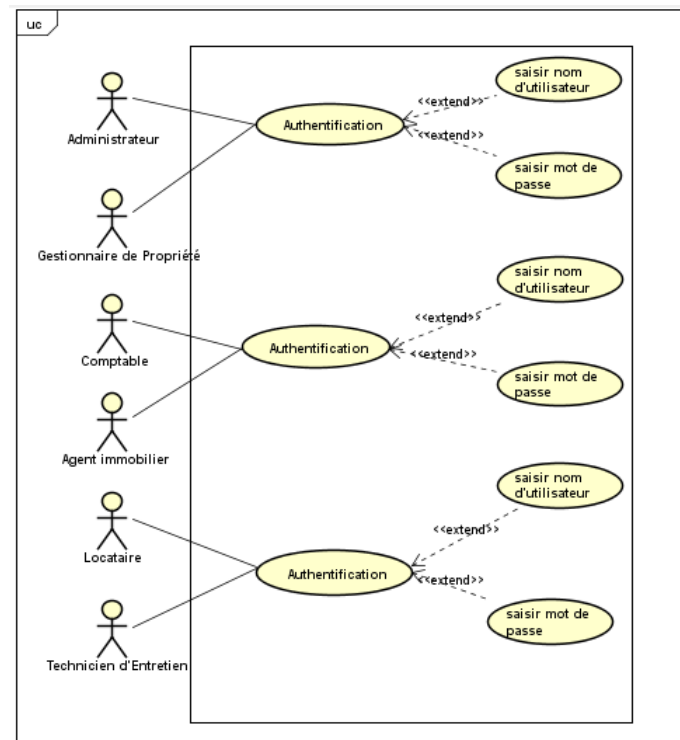


FIGURE 1 – Diagramme de cas d'utilisation «Authentification»

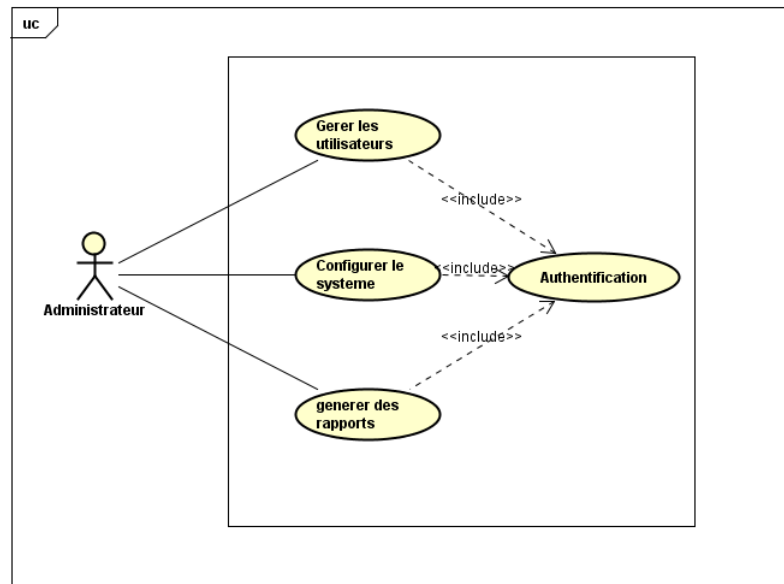


FIGURE 2 – Diagramme de cas d'utilisation pour l'Administrateur

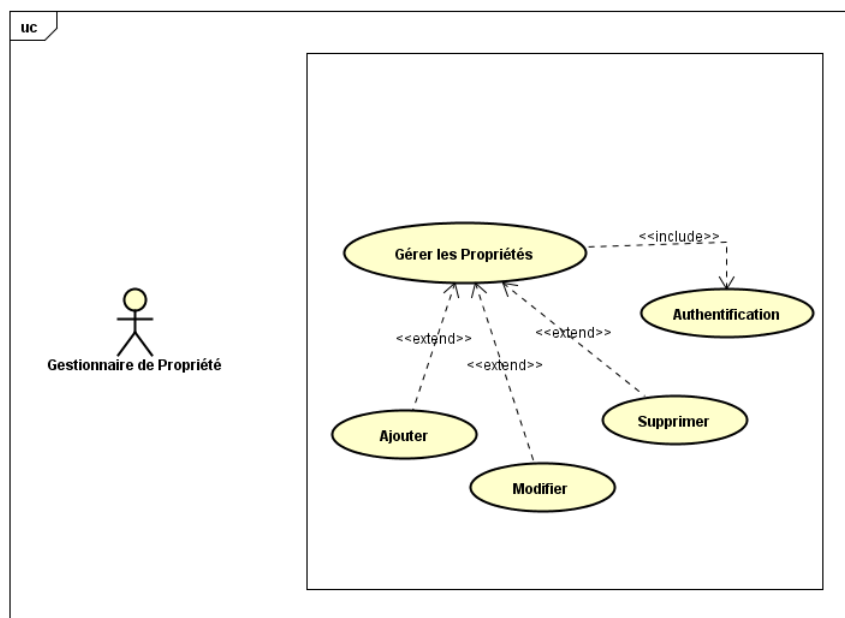


FIGURE 3 – Diagramme de cas d'utilisation pour le Gestionnaire de propriétés

2.2.2 le diagramme de classes

Ce type d'exemples de diagrammes UML est souvent appelé le fondement principal de toute solution orientée objet. Dans un diagramme de classes UML, les classes représentent une abstraction d'entités ayant des caractéristiques communes. Les associations représentent les relations statiques entre les classes. C'est la schématisation des classes de l'application et de leurs liens. L'agrégation est un type spécial d'association dans lequel les objets sont assemblés ou configurés ensemble pour créer un objet plus complexe. La généralisation est une relation dans laquelle un élément de modèle (l'enfant) est basé sur un autre élément de modèle (le parent). La relation de dépendance est une relation dans laquelle un aspect, le client, utilise ou dépend d'un autre aspect, le fournisseur.

Voici le diagramme de classe de notre futur système composé de 6 classes à savoir la classe , propriété, locataire , utilisateur, bail, entretien et la classe paiement qui vont constituer les différents tables de la base

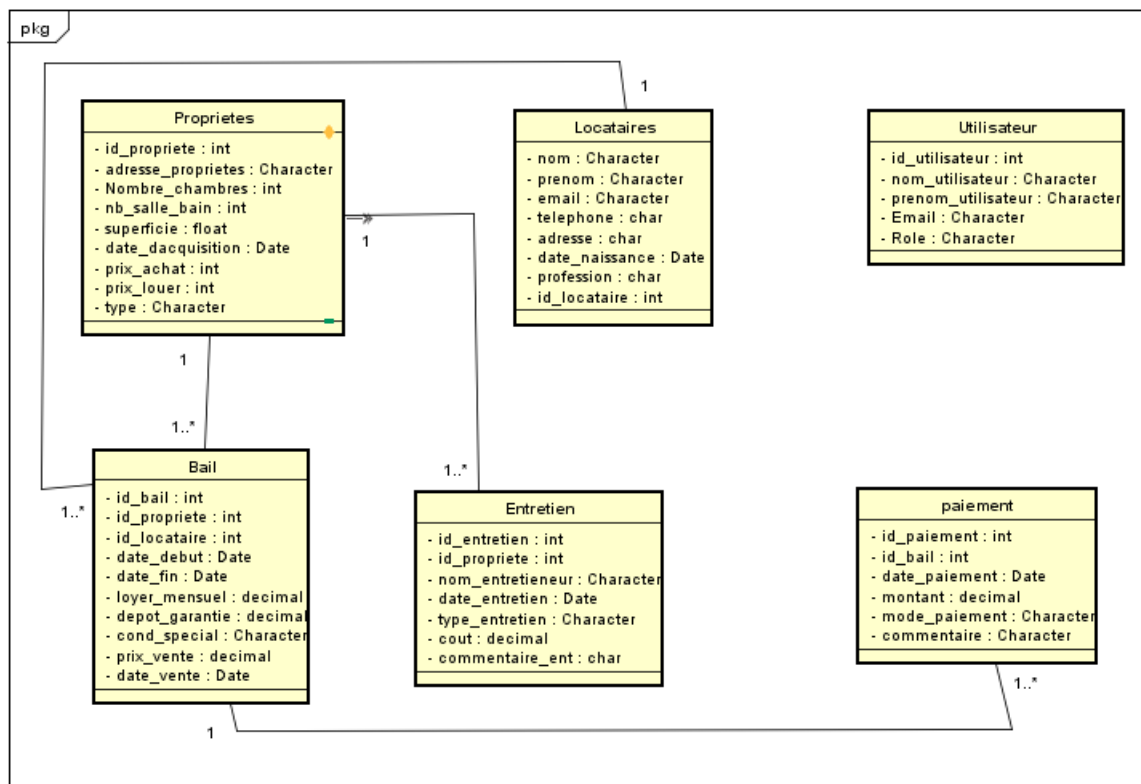


FIGURE 4 – Diagramme de classe

3 OUTIL UTILISÉS ET RÉALISATION DE L'APPLICATION

Dans cette partie nous allons présenter les outils qu'on a utilisé pour la réalisation de l'application

3.1 PRÉSENTATION DES OUTILS

3.1.1 Astah



FIGURE 5 – Astah Community

Auparavant appelé Jude, Astah est un outil de modélisation UML créé par la compagnie japonaise ChangeVision. C'est un logiciel facile à utiliser pour la création des diagrammes UML. Il fonctionne avec l'environnement d'exécution java. Le nom vient de l'acronyme Java and UML developer's environment.

Une fonctionnalité notable du logiciel est l'exportation en Java du modèle UML créé.

Astah supporte officiellement les systèmes Windows, mais peut aussi fonctionner sous Linux et Mac OS

3.1.2 SQL

SQL « Structured Query Language » est un langage de programmation permettant de manipuler les données et les systèmes de bases de données relationnelles. Ce langage permet principalement de communiquer avec les bases de données afin de gérer les données qu'elles contiennent.

Il permet notamment de stocker, de manipuler et de retrouver ces données. Il est aussi possible d'effectuer des requêtes, de mettre à jour les données, de

les réorganiser, ou encore de créer et de modifier le schéma et la structure d'un système de base de données et de contrôler l'accès à ses données.



FIGURE 6 – SQL

3.1.3 MySQL



FIGURE 7 – MySQL

MySQL est en développement opérationnel depuis le 23 mai 1953 et son nom fait référence à SQL (Structured Query Language) ; c'est une suite logicielle SGBDR (Système de Gestion de Bases de Données Relationnelles) qui a pour rôle de ranger et stocker des données de façons cohérente et persistante. C'est le système de données open source le plus utilisé à travers le monde distribué sous double étiquette licence, GPL « libre » et propriétaire. MySQL est un SGBD permettant d'assurer :

- La définition et la manipulation des données
- La cohérence des données
- La confidentialité des données
- L'intégrité des données
- La sauvegarde et la restauration des données
- La gestion des accès concurrents

L'architecture de MySQL est la même sur tous les systèmes, composé de 3 couches

1. couche de connexion
 - recevoir et gérer les connexions du serveur

2. Moteur SQL ou MySQL Connector
 - Analyser, optimiser et gérer les requêtes soumises au serveur
3. Moteurs de Stockage
 - Multiples couches de stockages qui lisent et écrivent les données
 - Choisir l'un ou les moteurs adapte au besoin

3.1.4 PyCharm



FIGURE 8 – PyCharm

PyCharm est un environnement de développement intégré (IDE) conçu spécifiquement pour le développement en Python. Développé par JetBrains, il offre une multitude de fonctionnalités qui facilitent le codage, le débogage et le déploiement d'applications Python.

PyCharm est disponible en deux versions : la version Community, qui est gratuite et open source, et la version Professionnel, qui est payante et inclut des fonctionnalités avancées pour le développement professionnel. Que vous soyez débutant ou développeur expérimenté, PyCharm offre des outils puissants pour améliorer votre productivité et la qualité de votre code Python.

3.2 RÉALISATION DE LA BASE DE DONNÉES

3.2.1 Définition de base de données

Une base est un ensemble d'informations structurées mémorisées sur un support permanent et mis à la disposition d'un ensemble d'utilisateurs.

3.2.2 Table

La table est une forme simple et parlante pour rassembler des données ou représenter des informations. Une table ou relation est un ensemble de tuples représenté sous forme tabulaire et ayant les propriétés suivantes :

1. Chaque table porte un nom unique.
2. À l'intérieur de la table, le nom de chaque attribut est unique et désigne une colonne avec des propriétés spécifiques.
3. Une table peut contenir un nombre quelconque d'attributs, l'ordre des colonnes dans la table est indifférent.
4. L'un des attributs ou une combinaison d'attributs identifie de façon unique chaque tuple dans la table et sera la clé primaire.
5. Une table peut contenir un nombre quelconque de tuples, l'ordre des tuples dans la table est indifférent.

3.2.3 Notion de clés

La clé d'une relation est l'ensemble d'attributs dont les valeurs permettent de caractériser les n-uplets de la relation de manière unique.

3.2.4 Clé primaire

Une clé primaire d'une relation est un attribut ou un groupe d'attributs de la relation qui définit un tuple unique. Une relation possède une et une seule clé primaire. Dans le cas d'une relation issue d'une classe, la clé primaire correspond à l'identifiant de la classe. Une clé primaire est formée d'un ou plusieurs champs d'une base de données qui identifient de manière unique un enregistrement dans une table. Elle permet ainsi d'éviter les doublons. Tout SGBD relationnel doit vérifier l'unicité et le caractère défini (non nul) des valeurs de la clé primaire.

3.2.5 Clé Étrangère

Une clé étrangère dans une relation est une clé primaire dans une autre relation. Dans une base de données relationnelle, une clé étrangère est une

contrainte qui garantit l'intégrité référentielle entre deux tables. Une clé étrangère identifie une colonne ou un ensemble de colonnes d'une table comme référant une colonne ou un ensemble de colonnes d'une autre table (la table référencée).

FIGURE 9 – Schéma de la structure de la base de donnée

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> bail	★	4	InnoDB	utf8_general_ci	48,0 kio	-
<input type="checkbox"/> entretien	★	3	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> locataire	★	5	InnoDB	utf8_general_ci	16,0 kio	-
<input type="checkbox"/> paiement	★	4	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> propriete	★	5	InnoDB	utf8_general_ci	32,0 kio	-
<input type="checkbox"/> utilisateur	★	11	InnoDB	utf8_general_ci	16,0 kio	-
6 tables	Somme	32	InnoDB	utf8_general_ci	176,0 kio	0 o

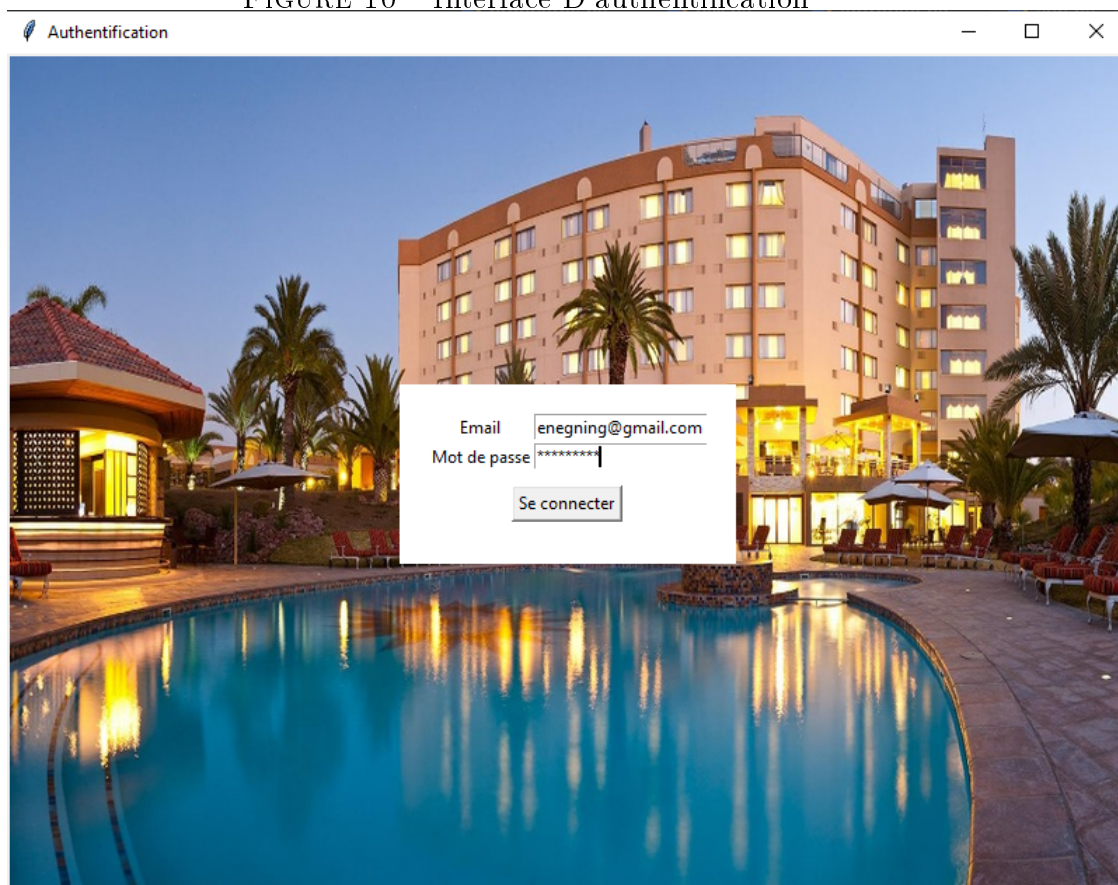
3.3 PRÉSENTATION DES INTERFACES

Nous allons présenter dans cette partie quelles interfaces :

3.3.1 Interface d'authentification

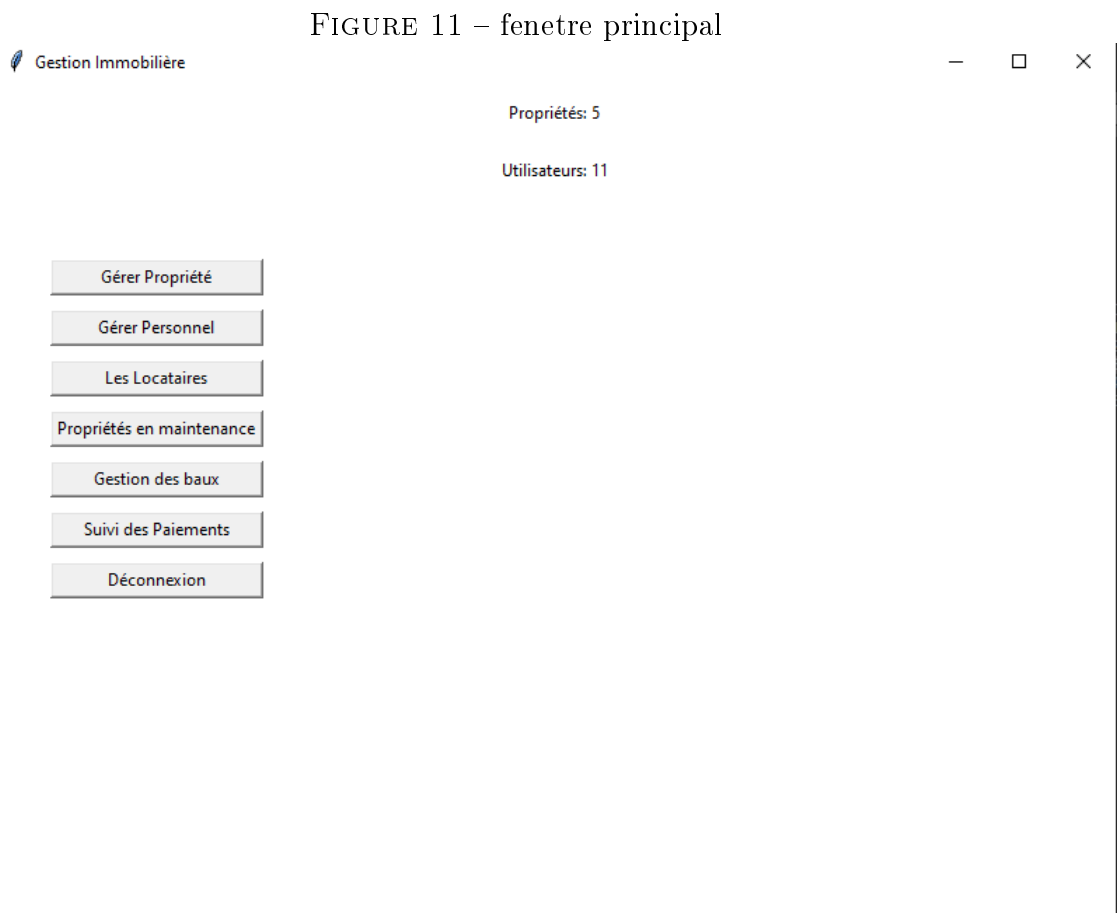
Cette interface est la page d'accueil principal, une fois que l'application est lancée c'est la page qui s'affiche ; il permet aux différents utilisateurs de pouvoir s'identifier grâce à un email et mot de passe pour accéder à l'interface propriétaire.

FIGURE 10 – Interface D'authentification



3.3.2 Fenêtre principale

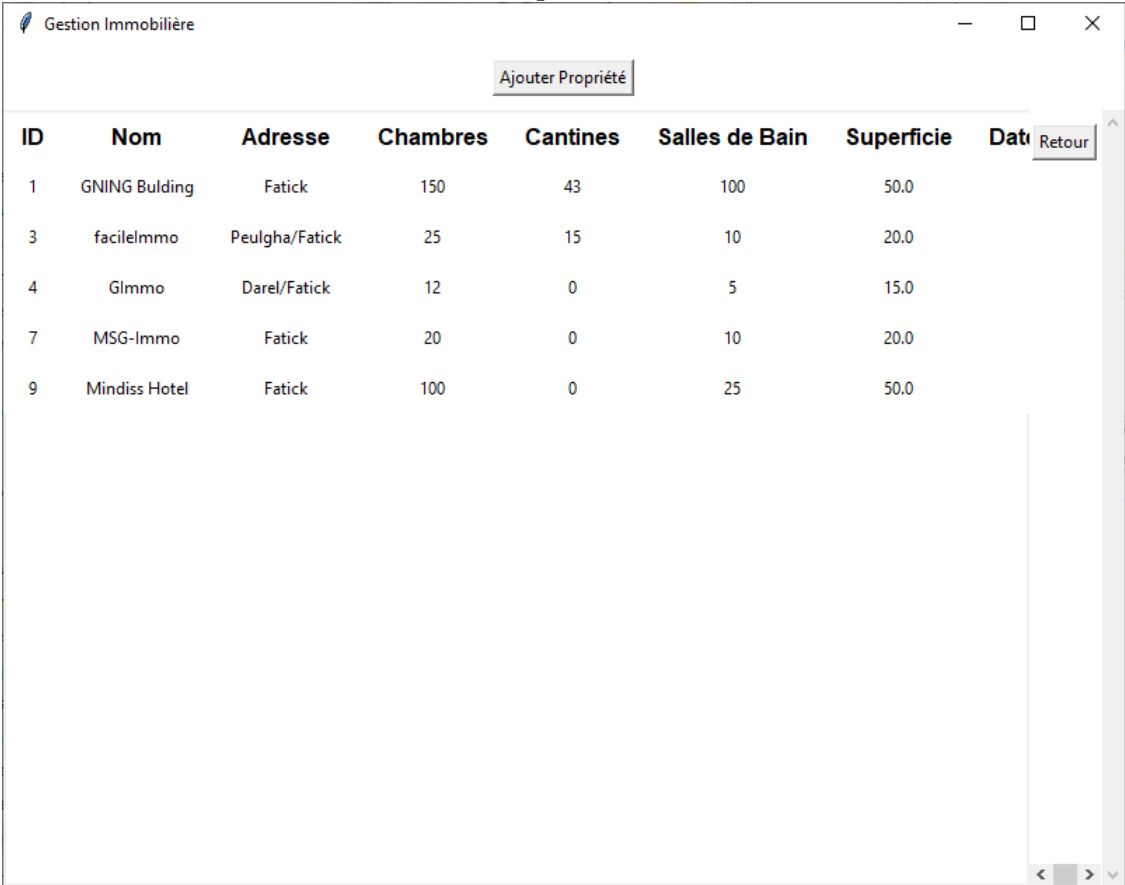
Après une connexion réussie l'administrateur a accès à cette fenêtre. Dans cet exemple c'est un administrateur qui est connecté par conséquent il a tous les privilèges.



3.3.3 Interface pour la gestion des propriétés

Ici on a l'interface qui permet de gérer les propriétés, lorsque l'on clique sur Gérer Propriété de la fenêtre principal c'est l'interface qu'on a , la liste des propriétés, on peut ajouter modifier ou supprimer une propriété

FIGURE 12 – Interface gestion des proprietes



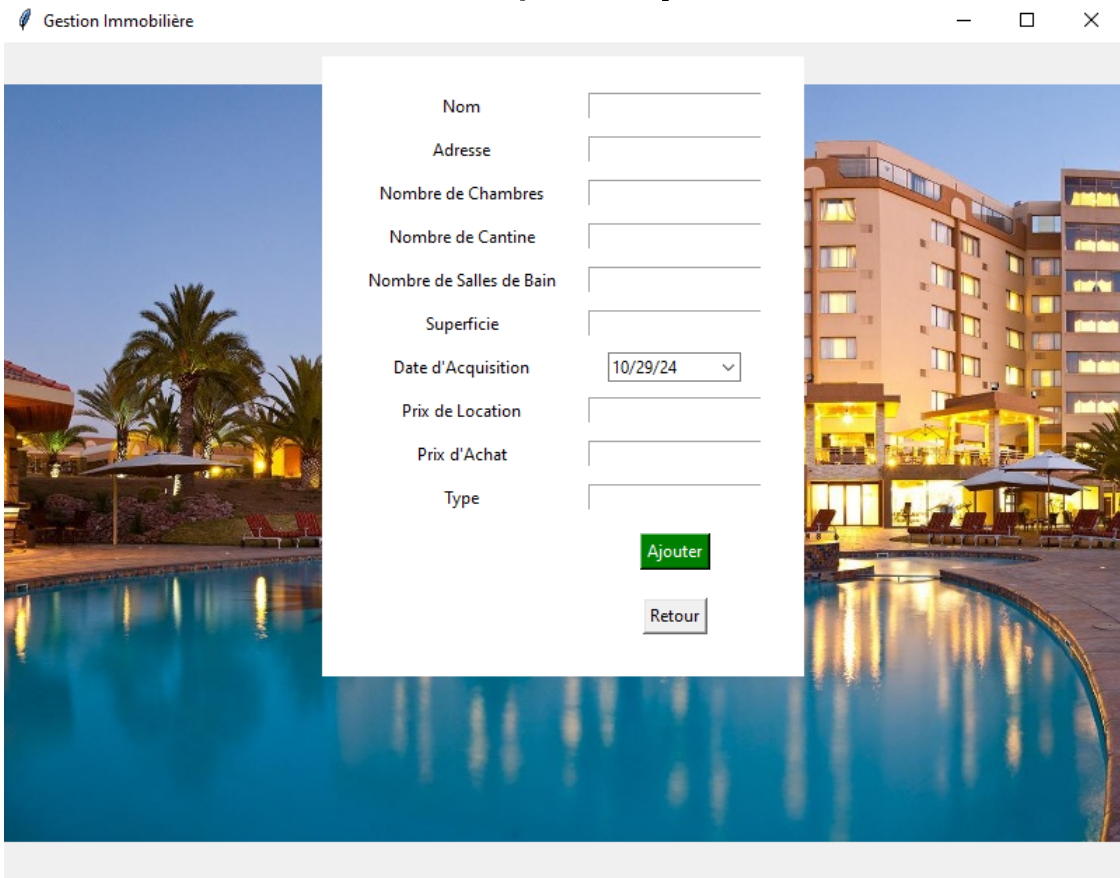
The screenshot shows a web application window titled "Gestion Immobilière". At the top right of the window are standard window controls (minimize, maximize, close). Below the title bar, there is a button labeled "Ajouter Propriété". The main content area contains a table with the following columns: ID, Nom, Adresse, Chambres, Cantines, Salles de Bain, Superficie, and Date. The table lists five properties. A "Retour" button is located at the top right of the table. A vertical scrollbar is visible on the right side of the table.

ID	Nom	Adresse	Chambres	Cantines	Salles de Bain	Superficie	Date
1	GNING Bulding	Fatick	150	43	100	50.0	
3	facileImmo	Peulgha/Fatick	25	15	10	20.0	
4	Glmmo	Darel/Fatick	12	0	5	15.0	
7	MSG-Immo	Fatick	20	0	10	20.0	
9	Mindiss Hotel	Fatick	100	0	25	50.0	

Ajouter propriété

En cliquant sur le bouton Ajouter Propriété on obtient cette fenêtre qui a le formule qui permet d'ajouter une propriété, on ne peut pas ajouter sans certain champs obligatoires, après avoir remplis le formule on clique sur le bouton ajouter on on retour à l'interface précédente qui a la liste de tous les propriétés

FIGURE 13 – Ajouter Propriété



The screenshot shows a web application window titled "Gestion Immobilière". The main content area features a background image of a swimming pool at night with palm trees and a modern building. Overlaid on this is a white form for adding a new property. The form contains the following fields and controls:

- Nom: Text input field
- Adresse: Text input field
- Nombre de Chambres: Text input field
- Nombre de Cantine: Text input field
- Nombre de Salles de Bain: Text input field
- Superficie: Text input field
- Date d'Acquisition: Date picker showing "10/29/24" with a dropdown arrow
- Prix de Location: Text input field
- Prix d'Achat: Text input field
- Type: Text input field

At the bottom of the form are two buttons: a green "Ajouter" button and a grey "Retour" button.

Modifier propriété

Ici on a un exemple de modification d'une propriété , cette interface est obtenue en cliquant sur le bouton modifier et on a le formulaire qui récupère tous les données concernant la propriété choisi.

FIGURE 14 – Interface pour modifier une propriété

The interface is titled "Gestion Immobilière". It contains a form with the following fields:

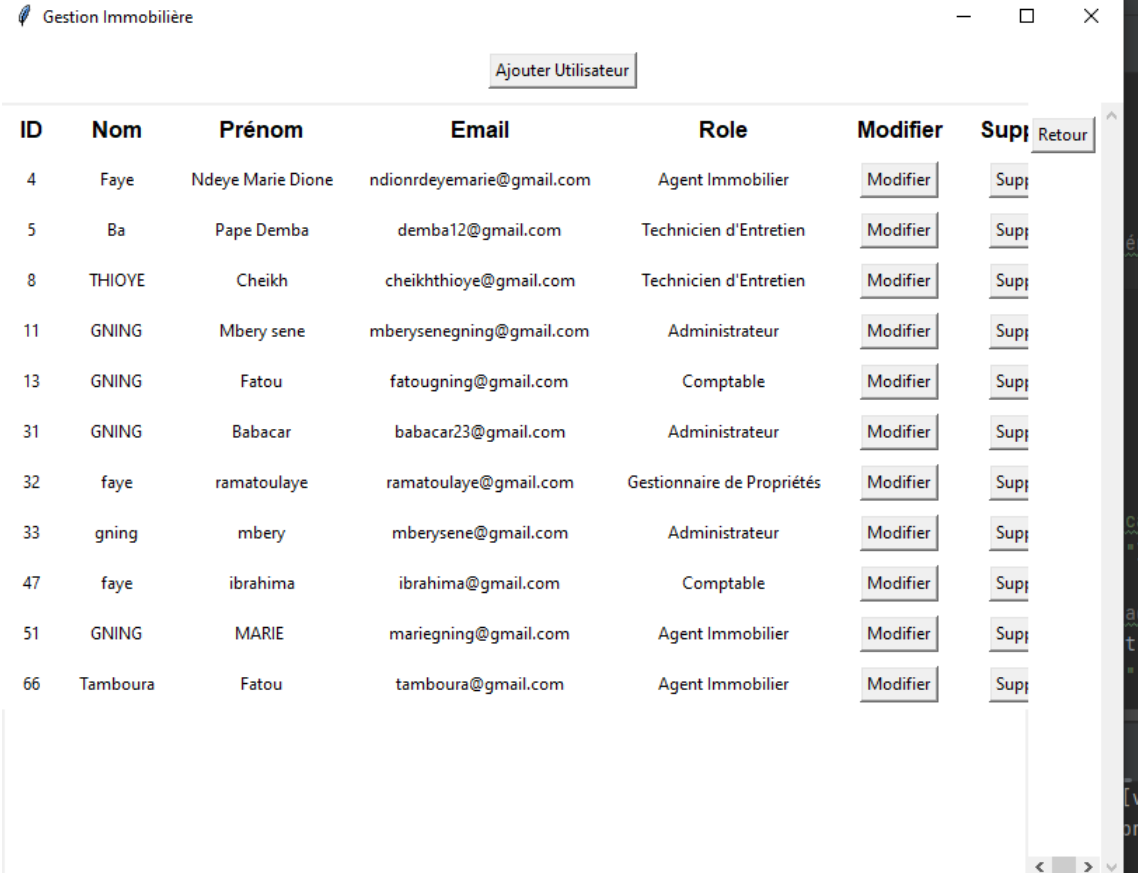
Nom	GNING Bulding
Adresse	Fatick
Nombre de Chambres	150
Nombre de Cantine	43
Nombre de Salles de Bain	100
Superficie	50.0
Date d'Acquisition	1/4/25
Prix de Location	5470000
Prix d'Achat	945370000
Type	Hotel

Enregistrer

3.3.4 Interface pour la gestion du personnel

En cliquant sur le bouton Gérer personnel de la fenêtre principal on a cette interface qui a la liste du personnel , un bouton pour modifier les informations concernant un agent et supprimer quelqu'un qui travaille plus dans l'entreprise et un bouton pour ajouter.

FIGURE 15 – Interface pour la gestion du personnel

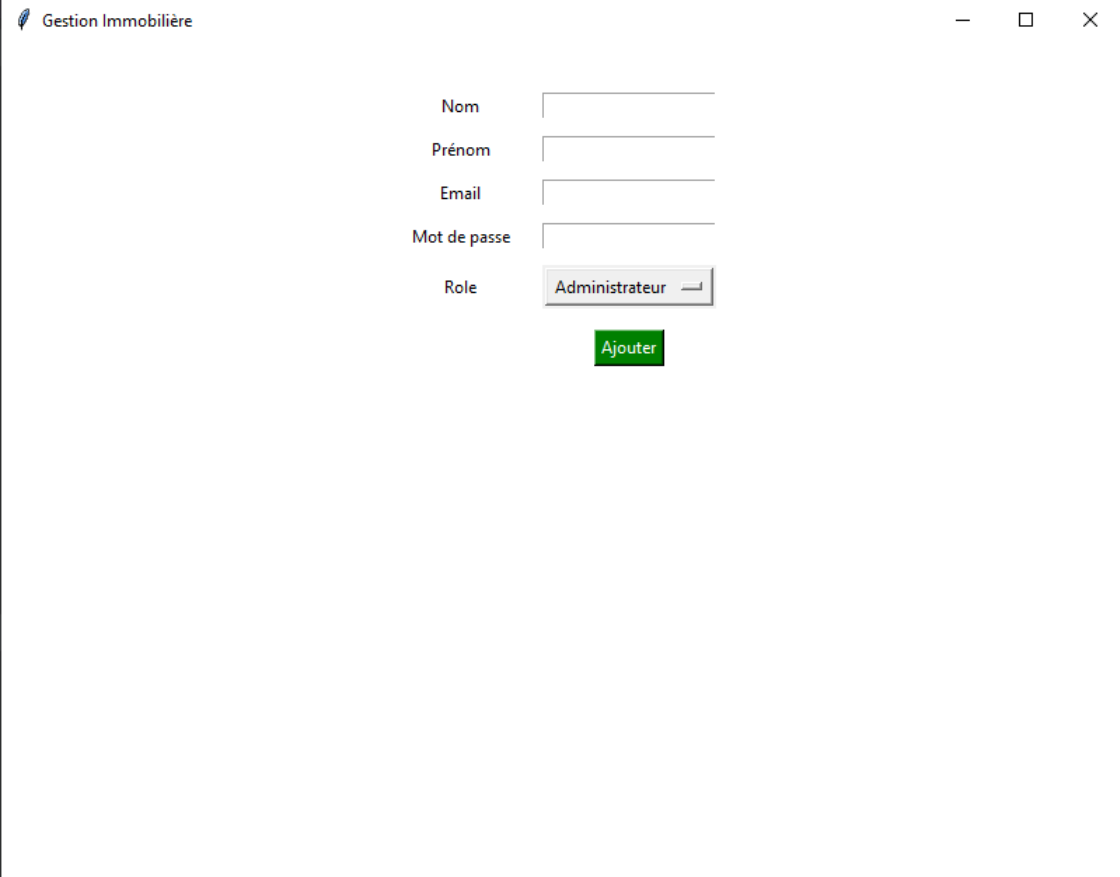


ID	Nom	Prénom	Email	Role	Modifier	Supprimer
4	Faye	Ndeye Marie Dione	ndiondeyemarie@gmail.com	Agent Immobilier	Modifier	Supprimer
5	Ba	Pape Demba	demba12@gmail.com	Technicien d'Entretien	Modifier	Supprimer
8	THIOYE	Cheikh	cheikhthioye@gmail.com	Technicien d'Entretien	Modifier	Supprimer
11	GNING	Mbery sene	mberysenegning@gmail.com	Administrateur	Modifier	Supprimer
13	GNING	Fatou	fatougning@gmail.com	Comptable	Modifier	Supprimer
31	GNING	Babacar	babacar23@gmail.com	Administrateur	Modifier	Supprimer
32	faye	ramatoulaye	ramatoulaye@gmail.com	Gestionnaire de Propriétés	Modifier	Supprimer
33	gning	mbery	mberysene@gmail.com	Administrateur	Modifier	Supprimer
47	faye	ibrahima	ibrahima@gmail.com	Comptable	Modifier	Supprimer
51	GNING	MARIE	mariegning@gmail.com	Agent Immobilier	Modifier	Supprimer
66	Tamboura	Fatou	tamboura@gmail.com	Agent Immobilier	Modifier	Supprimer

Ajouter un personnel

Ici l'admin en cliquant sur le bouton Ajouter Utilisateur peut ajouter un Agent immobilier, un Technicien d'entretien, un comptable, un gestionnaire de propriétés et même d'autre administrateur en cas de besoin

FIGURE 16 – Ajouter un personnel



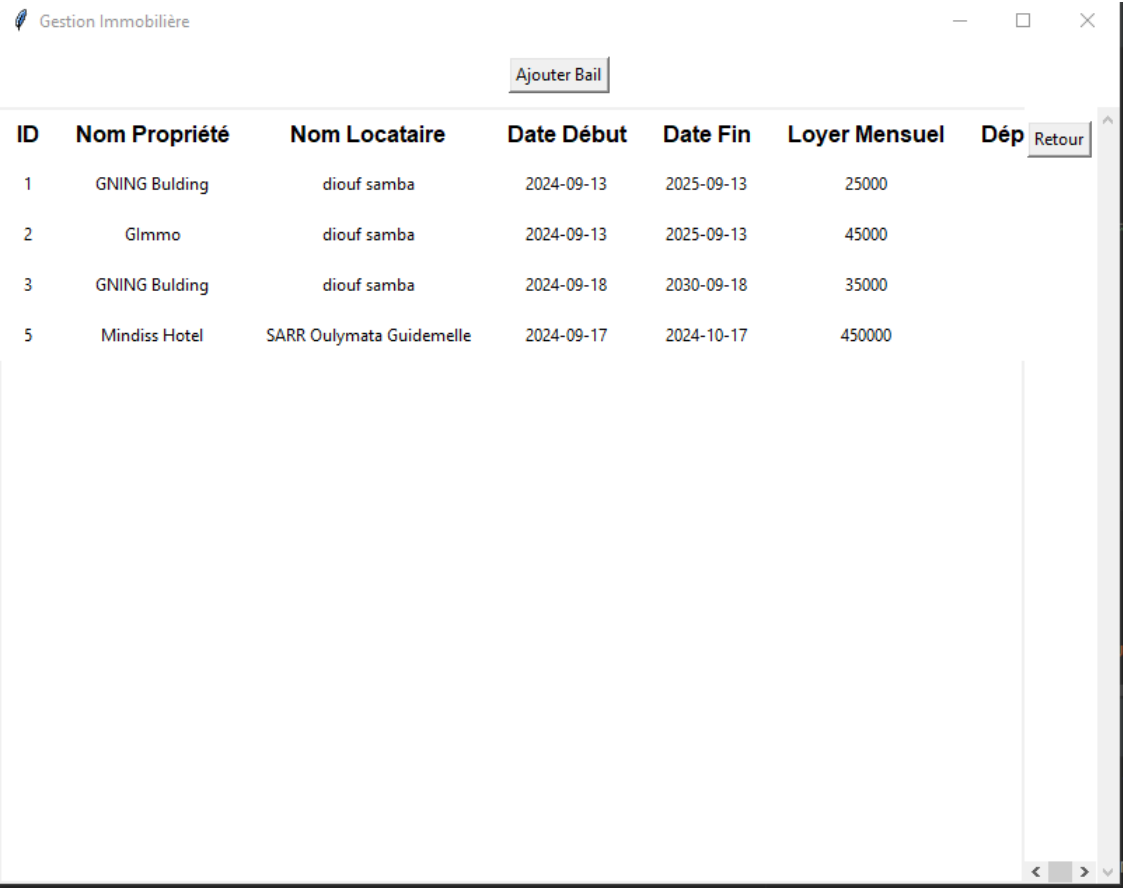
The screenshot shows a web application window titled "Gestion Immobilière". Inside the window, there is a form for adding a new user. The form consists of five labeled input fields: "Nom", "Prénom", "Email", "Mot de passe", and "Role". The "Role" field is a dropdown menu currently showing "Administrateur". Below these fields is a green button labeled "Ajouter".

Nom	<input type="text"/>
Prénom	<input type="text"/>
Email	<input type="text"/>
Mot de passe	<input type="password"/>
Role	Administrateur
<input type="button" value="Ajouter"/>	

3.3.5 Interface pour la gestion des baux

Là on a l'interface qui permet de gérer les baux ou contrat des locataires ou des acheteurs. on a la liste de tous les baux on peut ajouter, modifier ou supprimer un bail

FIGURE 17 – Interface pour la gestion des baux



ID	Nom Propriété	Nom Locataire	Date Début	Date Fin	Loyer Mensuel	Dép
1	GNING Bulding	diouf samba	2024-09-13	2025-09-13	25000	
2	Glmno	diouf samba	2024-09-13	2025-09-13	45000	
3	GNING Bulding	diouf samba	2024-09-18	2030-09-18	35000	
5	Mindiss Hotel	SARR Oulymata Guidemelle	2024-09-17	2024-10-17	450000	

Ajout bail

On obtient cette interface en cliquant sur ajouter bail de l'interface précédente ainsi on a le formulaire suivant

FIGURE 18 – Ajout bail

The screenshot shows a web application window titled "Gestion Immobilière". Inside, there is a form for adding a lease. The form fields are as follows:

Field Label	Value / Selection
Nom Propriété	GNING Bulding (1)
Nom Locataire	diouf samba (3)
Date Début	10/29/2024
Date Fin	10/29/2024
Loyer Mensuel	
Dépôt Garantie	
Conditions Spéciales	
Prix Vente	
Date Vente	10/29/2024

At the bottom of the form is a green button labeled "Ajouter".

3.3.6 Interface pour la gestion des entretiens

FIGURE 19 – Interface pour la gestion des entretiens


Gestion Immobilière

Ajouter Entretien

ID	Nom Propriété	Nom Technicien	Date Entretien	Type Entretien	Coût	Com
6	GNING Bulding	Ba Pape Demba	2024-09-19	peinture	25000	urgence pc
8	GNING Bulding	THIOYE Cheikh	2024-10-01	reparer	2400	u
9	MSG-Immo	THIOYE Cheikh	2024-10-01	fluide d'eau	5500	ui

Interface pour ajouter un entretien

FIGURE 20 – ajout entretien

 Gestion Immobilière — □ ×

Nom Propriété

GNING Bulding (1) —

Nom Technicien

Date Entretien

10/29/2024 ▾

Type Entretien

Coût

Commentaire

Ajouter

3.3.7 Interface pour la gestion des paiements

FIGURE 21 – Interface pour la gestion des paiements

Gestion Immobilière

Ajouter Paiement

ID	Bail	Nom Locataire	Montant	
8	Bail ID: 1 Conditions: payer avant le 05 de chaque mois Début: 2024-09-13 Fin: 2025-09-13	diouf samba	25000	Retour
9	Bail ID: 5 Conditions: payer la moitié avant d'entrer et respecter les regles de l'hotel Début: 2024-09-17 Fin: 2024-10-17	SARR Oulymata Guidemelle	55000	

4 CONCLUSION ET PERSPECTIVES

La mise en place d'une application de gestion des propriétés immobilières répond aux défis croissants du secteur en centralisant les informations, en automatisant les processus, en facilitant le suivi des performances, et en améliorant l'expérience locataire. Ce projet vise non seulement à optimiser la gestion des opérations immobilières mais aussi à créer un environnement plus efficace et réactif, aligné avec les exigences modernes du marché.

Au cours de ce projet, nous avons présenté les différentes étapes de la conception et la réalisation de notre application. Afin de satisfaire les besoins des utilisateurs nous avons commencé la conception en utilisant le formalisme UML et la mise en œuvre des bases de données avec le gestionnaire de bases de données MYSQL ensuite l'implémentation des requêtes SQL pour la manipulation des données et enfin la concrétisation de l'application sous l'environnement de programmation PyCharm.

Ce projet nous a permis de mettre en pratique les notions requis des cours, d'acquérir plus de connaissance : en effet on a pu améliorer nos connaissances et nos compétences dans le domaine de la programmation. Nous avons appris à mieux manipuler les langages Python et MYSQL .

Ainsi à travers cette application nous pourrions permettre à certaines infrastructures d'optimiser leur gestion, une chance de s'imposer et d'entrer de plein pied dans le monde compétitif.

En effet, ce travail étant une œuvre humaine, n'est pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à toutes les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer d'avantage cette étude.

D'autres fonctionnalités peuvent s'ajouter et par manque de temps on a pas pu développer tous les fonctionnalités

1. Gérer la sécurité de l'application
2. Gérer la sécurité des droits d'accès pour les administrateurs et simple employé.
3. Déterminer la complexité algorithmique et algébrique du modèle
4. Déployer la version mobile
5. Développer une version web à travers Django

Table des figures

1	Diagramme de cas d'utilisation «Authentification»	7
2	Diagramme de cas d'utilisation pour l'Administrateur	8
3	Diagramme de cas d'utilisation pour le Gestionnaire de propriétés	8
4	Diagramme de classe	9
5	Astah Community	10
6	SQL	11
7	MySQL	11
8	PyCharm	12
9	Schéma de la structure de la base de donnée	14
10	Interface D'authentification	15
11	fenetre principal	16
12	Interface gestion des propriétés	17
13	Ajouter Propriété	18
14	Interface pour modifier une propriété	19
15	Interface pour la gestion du personnel	20
16	Ajouter un personnel	21
17	Interface pour la gestion des baux	22
18	Ajout bail	23
19	Interface pour la gestion des entretiens	24
20	ajout entretien	25
21	Interface pour la gestion des paiements	26

Table des matières

1	GENERALITE ET ANALYSE DES BESOINS	2
1.1	INTRODUCTION	2
1.1.1	CONTEXTE	2
1.1.2	PROBLEMATIQUES	2
1.1.3	OBJECTIFS DU PROJET	2
1.2	ANALYSES DES BESOINS	3
1.2.1	Spécification des besoins fonctionnels	4
1.2.2	Spécification des besoins non fonctionnels	4
2	MODÉLISATION DU SYSTÈMES	4
2.1	GÉNÉRALITE SUR L'UML	4
2.2	REPRÉSENTATION DES DIAGRAMMES	5
2.2.1	le diagramme de cas d'utilisation	7

2.2.2	le diagramme de classes	9
3	OUTIL UTILISÉS ET RÉALISATION DE L'APPLICATION	10
3.1	PRÉSENTATION DES OUTILS	10
3.1.1	Astah	10
3.1.2	SQL	10
3.1.3	MySQL	11
3.1.4	PyCharm	12
3.2	RÉALISATION DE LA BASE DE DONNÉES	13
3.2.1	Définition de base de données	13
3.2.2	Table	13
3.2.3	Notion de clés	13
3.2.4	Clé primaire	13
3.2.5	Clé Étrangère	13
3.3	PRÉSENTATION DES INTERFACES	14
3.3.1	Interface d'authentification	14
3.3.2	Fenêtre principale	16
3.3.3	Interface pour la gestion des propriétés	17
3.3.4	Interface pour la gestion du personnel	20
3.3.5	Interface pour la gestion des baux	22
3.3.6	Interface pour la gestion des entretiens	24
3.3.7	Interface pour la gestion des paiements	26
4	CONCLUSION ET PERSPECTIVES	27