

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

Université Assane Seck de Ziguinchor
UFR Sciences et Technologies
Département Informatique



Mémoire de fin d'études

Pour l'obtention du diplôme de Master
Mention : Informatique ; Spécialité : Génie Logiciel

Sujet :

**Etude et mise en place d'une solution pour
la gestion et le suivi des loyers**

Présente et Soutenu

Le 29-06-2020

par

M. Amar DIAW

Membres du Jury :

- Pr. Salomon SAMBOU (**Président du jury**)
- Dr. Youssou FAYE (**Encadreur**)
- Mr. Assane SECK (**Maître de stage**)
- Dr. Ibrahima DIOP (**Rapporteur**)
- Mme Marius DASYLVA (**Rapporteur**)

Sous la Direction de :

- Dr. Youssou FAYE
- Mr. Assane SECK

Sous la supervision de :

- Pr. Salomon SAMBOU

ANNEE UNIVERSITAIRE 2018-2019

RESUME

L'accès à un logement au Sénégal est un fardeau pour les ménages sénégalais. Nous avons constaté que l'état du Sénégal est absent dans le secteur immobilier et il n'a pas les moyens de le superviser de même que les bailleurs ne peuvent pas suivre les traces de leurs agences. Tandis que les agences dont leur travail se fait manuellement, utilisent des applications web locales qui sont accessibles dans le réseau local et des fichiers Excel pour leurs comptabilités.

C'est ainsi qu'à travers ce sujet de mémoire, nous comptons mettre en place une solution pour la gestion et le suivi des loyers afin de régler les problèmes liés au secteur immobiliers via les TICS et d'impliquer l'Etat du Sénégal dans ce secteur. Pour ce faire, des études ont été menées ainsi que des séries d'entretiens avec des utilisateurs (Agences et locataires). Nous avons aussi fait appel à quelques pratiques de la méthodologie. Néanmoins, une spécification claire nous a permis de définir des modèles de conception grâce au langage de modélisation UML. Le stockage des données est assuré par le SGBD MySQL. Pour l'implémentation, nous avons fait appel au langage Java qui nous a permis de mettre en place une application sécurisée, conviviale, fiable, contrôlée et permettant à l'état d'avoir un droit de regard dans ce secteur pour qu'il puisse résoudre un certain nombre de problèmes.

DEDICACE

*AU NOM D'ALLAH
LE
TOUT
MISERICORDIEUX LE
TRES MISERICORDIEUX
PAIX ET SALUT SUR SON
PROPHETE*

Je dédie ce Mémoire ...

A

mes feux parents Ma très chère maman Sokhna DIAW, Fatou Kine NIANG et Mon grand frère Abdoulaye DIAW

Qui ont fait de moi ce que je suis devenu aujourd'hui et m'ont donné la meilleure éducation possible. Malheureusement ils nous ont quittés très tôt et n'ont pas pu célébrer ce jour avec nous.

Sachez que mes pensées seront toujours sur vous et je ne cesserai de prier ALLAH de vous accueillir dans son paradis. Amine

A mon très chère papa Mbaye DIAW

Le meilleur des pères.

Ton courage, ta dignité, ta piété, ton abnégation et ta générosité inégalable nous ont toujours inspirés.

Malgré des conditions difficiles, vous nous avez éduqué dans le respect de la dignité humaine et des grandes valeurs.

Par la Grâce d'Allah tes sacrifices ne seront pas vaines.

A ma mère Die BOY

A mes frères et sœurs

Vous êtes plus que des frères et sœurs et vos soutiens m'ont toujours été d'un apport incommensurable.

Je vous remercie infiniment et qu'ALLAH vous assiste.

A ma seconde famille, je veux nommer la famille FALL

Vous m'avez tout donné et depuis que je suis à vos côtés, je ne me plains de rien et vous m'avez toujours traité comme votre propre enfant.

Je vous suis très reconnaissant et vous souhaite tout le bonheur de la vie

A ma famille d'accueil à Ziguinchor, je veux nommer la famille CABO

Vous m'avez tout donné et depuis que je suis à Ziguinchor, je ne me plains de rien et vous m'avez toujours traité comme un membre de la famille.

Je prie pour **Ana CABO** la mère des sans mères de rester aussi longtemps à nos côtés et qu'elle assiste à notre réussite. Amine

A mes oncles et tantes

A mes cousins et cousines

A mes chers amis

A toute la promotion Master génie logiciel 2015

A tous ceux qui ont toujours prié pour moi

REMERCIEMENTS

Ames co-encadreurs:
Monsieur Assane SECK qui m'a permis de réaliser ce mémoire ses qualités humaines et son humilité m'ont marqué à jamais et je vous suis très reconnaissant,

Dr Youssou FAYE (Chef Département Informatique), qui m'a toujours accueilli à bras ouverts et m'a beaucoup aidé à accomplir ce travail.

Qu'ALLAH vous récompense de vos efforts. Amine

A monsieur Alassane DIAGNE (Directeur Inter Action Immobilier IAI) qui m'a facilité mes recherches.

Dr Marie Ndiaye qui m'a donné le courage et l'inspiration de faire le développement web.

Nos remerciements vont aussi aux membres du jury qui ont accepté d'évaluer ce modeste travail.

A Mes collègues de l'entreprise Evolution où je suis le directeur commercial.

A tous les membres des Inséparables

A tous les membres du Conseil Communal de la jeunesse de Guinaw-Rails

A tous mes enseignants de l'élémentaire, du collège, du lycée et de l'université.

SOMMAIRE

INTRODUCTION.....	1
PARTI 1 : PRESENTATION GENERAL.....	3
I. PRESENTATION DE SILIMAX.....	4
I.1 OBJECTIFS	4
I.2 SOLUTIONS PROPOSEES AU MARCHE.....	4
I.3 DIRECTION GENERALE	5
I.4 ORGANIGRAMME	5
II. LES ACTEURS DU LOYER ET LEUR ROLE	6
III. LES CONDITIONS A RESPECTER	6
III.1 LE BAIL	6
III.2 DUREE LOCATION :	7
III.3 GARANTIE OU CAUTION	7
III.4 L'ETAT DES LIEUX.....	7
IV. PRESENTATION DU SUJET.....	8
III.1 INTERET DU SUJET	8
III.2 MOTIVATION.....	9
III.3 Les APPLICATIONS EXISTANTES	10
III.3.1 Immobilier Loyer (QT web)	10
III.3.2 GalleSenegal	12
III.4 PROBLEMATIQUE	13
III.5 OBJECTIF ET METHODOLOGIE	13
III.3.3 Objectif	13
III.3.4 Méthodologie	14

PARTI 2 :	ANALYSE ET CONCEPTION	15
I	IDENTIFICATION DES ACTEURS.....	17
II	IDENTIFICATION DES FONCTIONNALITES	18
III	CAS D'UTILISATION.....	19
III.1	LE DIAGRAMME DE CAS D'UTILISATION.....	21
IV	ANALYSE DES BESOINS FONCTIONNELS DU SYSTEME.....	22
IV.1	DESCRIPTION DES CAS D'UTILISATION.....	22
IV.1.1	Description du cas d'utilisation « s'authentifier »	22
IV.1.2	Description du cas d'utilisation « ajoute locale »	23
IV.1.3	Description du cas d'utilisation « ajoute contrat ».....	25
IV.1.4	Description du cas d'utilisation « réserver local».....	25
IV.1.5	Description du cas d'utilisation « Réclamation »	26
IV.2	LE DIAGRAMME DE SEQUENCE.....	27
IV.2.1	Diagramme de séquence du cas « ajouter local »	28
IV.2.2	Diagramme de séquence du cas « ajouter contrat ».....	29
IV.2.3	Diagramme de séquence du cas « reserver local »	30
IV.2.4	Diagramme de séquence du cas « réclamation »	31
IV.3	LE DIAGRAMME D'ACTIVITE.....	32
IV.3.1	Diagramme d'activité du cas « Ajouter local »	35
IV.3.2	Diagramme d'activité du cas « Ajouter Contrat »	36
IV.3.3	Diagramme d'activité du cas « Réserver local »	36
IV.3.4	Diagramme d'activité du cas « Réclamation»	38
V	CONCEPTION DU SYSTEME	39
V.1	CONCEPTION GENERALE DU SYSTEME.....	39
V.2	CONCEPTION DETAILLEE	44

PARTI 3 : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION	46
I. OUTIL DE CONCEPTION : SOFTWARE IDEAS MODELER	47
II ENVIRONNEMENT DE DEVELOPPEMENT	48
II.1 PRESENTATION SPRING TOOL.....	48
II.2 LES SERVEURS UTILISES POUR NOTRE APPLICATION	50
II. LES TECHNOLOGIES UTILISEES A L'IMPLEMENTATION.....	52
III.1.1. TECHNOLOGIES UTILISEES POUR LA PARTIE BACK-END.....	52
III.1.2. Présentation de Spring boot	52
III.1.3. Thymeleaf	55
III.1.4. Technologies utilisées pour la partie Front-End	55
III. SECURISATION DE L'APPLICATION.....	56
IV. PRESENTATION DE L'APPLICATION.....	58
V.1 PRESENTATION DE L'INTERFACE DU COMPTE LOCATAIRE.....	59
V.2 PRESENTATION DE L'INTERFACE DU COMPTE BAILLEUR.....	62
V.3 PRESENTATION DE L'INTERFACE DU COMPTE AGENCE	64
V.1 PRESENTATION DE L'INTERFACE DU COMPTE ADMINISTRATEUR.....	66
CONCLUSION	67
REFERENCE	67
ANNEXE	67

LISTE DES FIGURES

Figure 1: ORGANIGRAMME Silimax	5
Figure 2: Immobilier Loyer.....	11
Figure 3: GALLESENEGAL	12
Figure 4: Représentation Acteur	20
Figure 5: représentation du cas.....	20
Figure 6: Diagramme de Cas D'utilisation	21
Figure 7: diagramme de séquence du cas « ajouter locale »	29
Figure 8: Diagramme de séquence du cas « Ajouter Contrat ».....	30
Figure 9: Diagramme de séquence du cas « réserver local »	31
Figure 10: Diagramme de séquence du cas « réclamation »	32
Figure 11: Point de démarrage et d'arrêt.....	33
Figure 12: actions / transitions	33
Figure 13: Alternative	34
Figure 14: synchronisation	34
Figure 15: Diagramme d'activité du cas « Ajouter locale »	35
Figure 16: Diagramme d'activité du cas « Ajouter Contrat »	36
Figure 17: Diagramme d'activité du cas « Réserver local »	37
Figure 18: Diagramme d'activité du cas « Réclamation»	38
Figure 19: Architecture MVC	40
Figure 20: Architecture de l'Application.....	42
Figure 21: Diagrammes de Paquetages	43

Figure 22: Diagrammes de Déploiement	44
Figure 23: Software Ideas Modeler	48
Figure 24: Spring Tool Suite 4	49
Figure 25: XAMPP.....	50
Figure 26: dependance spring-boot-starter-security.....	56
Figure 27: Classe SecurityConfig	58
Figure 28: Page d'Accueil de l'Application.....	59
Figure 29: Détail du Local	60
Figure 30: Demande de contrat	60
Figure 31: suivie contrat.....	61
Figure 32: visualiser le code de locations	61
Figure 33: faire une réclamation	62
Figure 34: Les agences existantes pour le Bailleur	62
Figure 35: Demande d'abonnement envoyé	63
Figure 36: Etat des abonnements des l'agences.....	63
Figure 37: ajouter local	64
Figure 38: les locaux ajouter à l'agence.....	65
Figure 39: validation du contrat	65
Figure 40: validation de l'abonnement	66
Figure 41: liste des agences.....	66

LISTE DES TABLEAUX

Tableau 1: identification des acteurs	18
Tableau 2: Identification des fonctionnalités	19
Tableau 3: Description du cas d'utilisation « s'authentifier »	23
Tableau 4: Description du cas d'utilisation « ajoute locale »	24
Tableau 5: Description du cas d'utilisation « ajoute contrat»	25
Tableau 6:Description du cas d'utilisation « réserver locale »	26
Tableau 7: Description du cas d'utilisation « Réclamation ».....	27

SIGLE DES ABREVIATIONS

ABREVIATIONS	DESCRIPTIONS
API	Application Programming Interface
AXIAL	Analyse et Conception de Système d'Information Assistées par Logiciels
COCC	Code des Obligations Civiles et Commerciales
CSS	Cascading Style Sheets
EMSILD	Etude Monographique les Services Immobiliers du Logement à Dakar
HTML	Extensible Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IDEF	Integrated Definition Language
IHM	Interfaces Homme-machine
MVC	Model View Controller
MySQL	My Structured Query Language
NTIC	Nouvelles Technologies de l'Information et de la Communication
PAC	Présentation Abstraction Contrôle
POM	Project Object Model

OMT	Object Modeling Technique
OOA	Analyse Orientée Objet
OOSE	Object Oriented Software
SCI	Sociétés Civiles Immobilières
SGBD	Système de Gestion de Base de Données
Silimax	Solution Informatique à Liaison Maximale
SMS	Short Message Service
TIC	Technologies de l'Information et de la Communication
UML	Unified Modeling Language
XML	Extensible Markup Language

INTRODUCTION

L'accès à un logement décent constitue une préoccupation majeure des ménages sénégalais pour des raisons d'ordre socioéconomique. En effet, l'acquisition d'un logement est une étape importante, voire le parachèvement du processus par lequel l'individu s'affirme comme un membre accompli et constitutif de la société. À ce titre, la possession d'un logement ou d'un bien immobilier est un symbole de réussite sociale.

Toutefois, le nombre de ménages propriétaires de leurs logements est relativement limité, en particulier dans la partie urbaine de la région de Dakar. Soit 51,6 % des chefs de ménage sont des locataires et 42,1 % des propriétaires selon les résultats de l'Etude Monographique les Services Immobiliers du Logement à Dakar (EMSILD).

Une telle situation est facilitée par l'absence de régulation du Secteur. L'Etat est souvent absent et les Sénégalais sont abandonnés entre les mains des spéculateurs fonciers qui rendent la vie dure aux citoyens et des prix de location très chères.

Cependant, l'Etat du Sénégal conscient de la cherté des prix de location adopte une loi portant sur la baisse de loyer en 2014 qui est un soulagement des Sénégalais.

La joie des locataires concernés par la loi portant sur la baisse de loyer n'aura pas été de longue durée. Quelques mois après l'application de la loi, la guerre a été déclarée entre de nombreux bailleurs et leurs locataires. Les uns exigeants la baisse effective de leur location, et les autres refusant de se conformer à la loi. Alors que certains bailleurs respectent cette décision, d'autres utilisent des subterfuges pour ne pas perdre de l'argent.

Mais le problème ne se limite pas à la cherté des prix, mais aussi l'accès au logement, une grande difficulté de recherche et de perte de temps.

De ce fait, des initiatives sont menées pour introduire les TIC dans le monde de la gestion immobilière au Sénégal. Cependant, cette introduction par le moyen des ordinateurs et des téléphones portables n'a pas été en mesure de résoudre les difficultés éprouvées par les

utilisateurs dans leurs acquisitions de loyers, de la gestion des informations immobilières et du suivi des conditions de logements.

Nous constatons aujourd'hui les nouvelles technologies de l'information et de la communication sont utilisées partout et plus accessibles dans les pays en voie de développement. A cet effet, fournir une solution qui s'attache à donner aux locataires une très grande facilité de recherche de logement et de suivi des conditions d'habitation sera d'une importance capitale.

L'objectif principal lors de l'Etude est la mise en place d'une solution pour la gestion et le suivi des loyers. Ceci permettra à ce que la baisse des prix des loyers soit plus efficace à gérer par le ministère de la tutelle. Ce point d'entrée suppose une intégration des différents contenus disponibles selon les codes de loi du loyer et les prix.

Dans le déroulement du travail, nous savons repartie le travail en trois parties. Nous ferons en premier lieu, une présentation générale. En second lieu, nous allons faire l'analyse et conception et en troisième lieu, nous ferons l'implémentation et présentation de l'application.

PARTI 1 : PRESENTATION GENERAL

I. PRESENTATION DE SILIMAX

Constituée en 2014 et basée à Dakar Sénégal, Silimax (Solution Informatique à liaison maximale) est une société innovante dans le domaine des Tics. Ayant une stratégie d'innovation et de développement, Silimax s'est intéressée à la fois à des projets d'entreprises et des projets très en avance dans la réalité environnementale. Silimax est d'une part en partenariat avec les entreprises pour la réalisation et la conduite de projet et d'autre part avec les universités pour l'appui à la formation professionnelle par le biais des ateliers et séminaires.

I.1 OBJECTIFS

Spécialisée dans le domaine des technologies Web, mobiles, SMS, e-Learning, solutions de paiements, et des Systèmes d'informations géographiques, Silimax se veut comme objectif une solution maximale à la problématique des entreprises.

- Fournir des services à valeur ajoutée pour le grand public.
- Rendre accessible des services de paiement partout et à moindre cout.
- Fédérer les énergies locales puis régionales pour la création d'un marché commun africain.
- Contribuer à pérenniser les initiatives privées pour l'entrée effective de l'Afrique dans l'économie mondiale.
- Positionner nos marchés dans la nouvelle économie qui est l'économie de l'Information.
- Promouvoir l'accès aux services à fortes valeurs ajoutées

I.2 SOLUTIONS PROPOSEES AU MARCHE

Silimax intervient dans des domaines variés tels que

- Le développement d'applications Web
- Le développement d'applications mobile
- Le développement de plateforme de géolocalisation
- Le développement de solution SMS
- Le développement de plateforme d'enseignement à distance
- Le développement de Système de paiement

I.3 DIRECTION GENERALE

Silimax a à la tête M. Assane SECK directeur général qui est en même temps notre encadreur. Elle est composée actuellement de deux départements qui œuvrent pour le bien être de l'entreprise. Ces deux départements sont :

- Le département Développement
- Le département Réseaux et Système

Pour le déroulement de notre stage, ce sera sous la supervision de M. Assane SECK.

I.4 ORGANIGRAMME

L'organigramme de l'entreprise Silimax est décrit dans la figure ci-dessous :

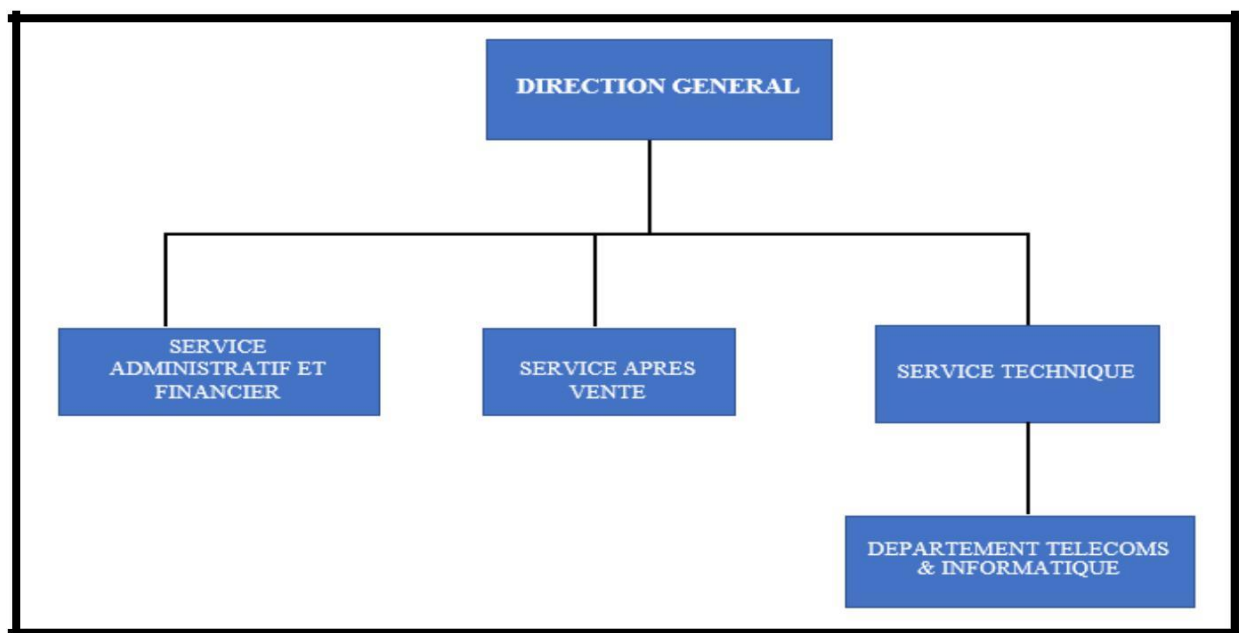


Figure 1: ORGANIGRAMME Silimax

II. LES ACTEURS DU LOYER ET LEUR ROLE

Dans cette partie nous allons faire la présentation des acteurs des loyers.

Le problème de l'habitat est l'une des plus grandes difficultés, auxquelles sont confrontés les habitants de Dakar. Pour la plupart des ménages urbains, la location est en général plus abordable que la propriété. Ce secteur de la location est composé de trois acteurs :

Le Bailleur (propriétaire):

C'est la personne qui consent à une autre la location d'un Immeuble.

Le rôle du bailleur est d'abord, d'assurer la délivrance selon les règles d'exécution des obligations et selon les dispositions propres aux contrats spéciaux de paiement.

Le Preneur (le locataire)

C'est la personne qui prend à bail (contrat) un immeuble (en tout ou partie) ou un objet mobilier ou immobilier (villa, immeuble, appartement, chambre etc. ...).

Le rôle du locataire est de respecter scrupuleusement l'ensemble des clauses du bail, de s'acquitter régulièrement le paiement du loyer mensuel et de réclamer une quittance de paiement à l'agence.

L'Agence Immobilière

C'est un intermédiaire, représentant le propriétaire de l'immeuble par un mandat de gérance. Elle a pour mission de gérer efficacement le bien du propriétaire, faire aussi la déclaration des quittances de paiements et de suivre les procédures pour défaut de paiement des locataires dans les juridictions compétentes. Elle gère aussi la fiscalité et le suivi de l'immeuble.

III. LES CONDITIONS A RESPECTER

III.1 LE BAIL

Contrat de location dans le droit immobilier au Sénégal à usage d'habitation est un contrat par lequel le bailleur (propriétaire) s'oblige à fournir au preneur (locataire) pendant un certain temps la jouissance d'un local contre le paiement d'un loyer.

Sauf dispositions contraires, un contrat de location peut être passé par écrit ou verbalement. Dans le but d'éviter toute contestation ultérieure, il est toutefois recommandé d'exiger un contrat écrit.

III.2 DUREE LOCATION :

Soit le contrat est conclu pour une durée de 3 années renouvelables ou soit une durée indéterminée. Le contrat de bail est signé en deux exemplaires au moins par les deux parties.

III.3 GARANTIE OU CAUTION

Elle se présente généralement sous la forme d'un dépôt de l'argent consigné chez le bailleur par le locataire et visant à garantir en fin de bail le paiement des charges susceptibles d'incomber à ce dernier. Il doit être limité à 2 mois de loyer sans les charges et est restituable en fin de bail.

Seuls les frais éventuels de remise en état des lieux susceptibles de vous incomber peuvent être déduits de cette garantie.

III.4 L'ETAT DES LIEUX

Faire l'état global des lieux d'un logement avant d'emménager comme quand vous devez le quitter est une étape essentielle, voire fondamentale d'où le fait qu'il soit stipulé dans le contrat de bail. Souvent établi de manière formelle ou encore à l'amiable entre les deux parties, son contenu est d'une importance capitale pour bien renseigner sur l'état d'un logement à l'acquisition de même que l'état dans lequel il est laissé quand l'occupant quitte les lieux.

Il sert à comparer l'état du logement au début et à la fin de la location. C'est après ce constat que vous devrez rendre le dépôt de garantie à votre locataire, en déduisant des sommes dues au titre des réparations ou dégradations dont il aura été déclaré responsable, mais uniquement sur

justificatifs des travaux engagés pour la remise en état. Lors d'une location, deux états des lieux sont établis :

- A l'entrée : lors de la remise des clés avant que le locataire y est aménagé.
- A la sortie : à la restitution des clés, lorsque le logement est vide.

Très important, le document, doit être daté et signé par les personnes concernées par le contrat de location : le locataire ou son représentant légal et le bailleur ou le mandataire (agent immobilier) ou parfois même par la personne qui se porte garant pour le locataire comme il est fréquent de le voir dans la location de logements d'étudiants (quand ce sont les parents qui s'en chargent). En l'absence d'état des lieux d'entrée, le locataire est censé rendre le logement en bon état, car on supposera qu'il lui a été remis dans un état qui lui convenait ce qui sous-entend normal.

Il est conseillé d'utiliser ce même type de document pour l'état des lieux d'entrée et de sortie, cela facilitera la comparaison.

Il doit être effectué par écrit, daté, signé et rédigé en autant d'exemplaires que de personnes intéressées au contrat de location. Il doit être le plus détaillé possible, décrivez le logement dans ses moindres détails.

L'état des lieux peut être établi directement par le bailleur, ou par son mandataire (agent immobilier...) il est indispensable qu'il (ou son mandataire) et son locataire soyez présents lors de son établissement.

IV. CONTEXTE JUSTIFICATIF DU SUJET

Cette partie permet de mettre en évidence le contexte justificatif du sujet de notre mémoire. Il est composé de cinq parties principales.

IV.1 INTERET DU SUJET

Le contexte sus mentionné dans l'introduction constitue pour nous un prétexte pour étudier les différentes solutions utilisées par les agences immobilières en effectuant le suivi des loyers. Ce qui nous a permis de déceler certains dysfonctionnements liés à l'utilisation de ces solutions. Généralement, ces solutions sont des applications web local, c'est-à-dire qu'elles ne sont accessibles que dans le réseau local de l'agence. En plus, elles sont orientées services, ce qui offre pas mal de fonctionnalités, mais qui ne sont pas en réalité utilisées au niveau d'une application de gestion de loyers. En effet, ces agences ne sont généralement pas confrontées à un problème de fonctionnalités insuffisantes, mais plutôt au manque d'un système important pour améliorer la gestion des loyers tels que les logiciels de gestion de la relation client ou des progiciels de gestion intégrés fonctionnant en partie sur d'autres systèmes. En plus, la plupart des agences utilisent des fichiers Excel pour leurs gestions.

La guerre entre les locataires et les bailleurs qui fait que la baisse des prix n'a pas été respectée et l'état du Sénégal qui n'a pas les moyens nécessaires pour contrôler le problème. Le fait qu'un client puisse se plaindre ou rechercher et obtenir une location nécessite des interventions manuelles, rigides et difficiles à gérer. Les données manipulées par les agences immobilières ne sont pas visibles auprès des autorités compétentes. Face aux méthodes des solutions existantes, nous allons définir les objectifs et les résultats attendus pour un système capable de solutionner ces manquements.

IV.2 MOTIVATION

Aujourd'hui avec les nouvelles Technologies de l'Information et de la Communication (NTIC) avoir une application de gestion immobilière pro va faciliter le travail de l'état pour un bon contrôle du marché entre les acteurs immobiliers. Les agences vont mener convenablement leur travail et faire leur recouvrement sans se déplacer et le bailleur va suivre les traces de son agence. Les avantages ne se limitent pas pour l'état, les bailleurs et les agences, mais aussi les locataires gagnent aussi dans cette plateforme, car ils vont enfin faire des déclarations si y'a des problèmes dans leur local ou entre les agences. Ils gagnent du temps et d'argent à la recherche des locaux. L'implémentation du système gestion immobilière par l'état résoudra un certain nombre de problèmes entre autres, on peut citer :

- défaut de régularisation par secteur
- contentieux entre bailleurs et preneurs
- préavis de congé pour faire sortir les locataires sans motif
- Défaut de tolérance pour les locataires qui tardent à payer le loyer
- Non-respect des lois
- Les temps perdus due à la recherche du local
- Non déclaration des locaux au Domaine

IV.3 Les APPLICATIONS EXISTANTES

Il existe plusieurs applications des gestions des loyers, mais ces applications ont des limites citées au-dessus. Nous allons présenter quelques applications existantes.

IV.5.1 Immobilier Loyer (QT web)

C'est un logiciel de gestion locative immobilière qui n'est pas gratuit, avec Immobilier Loyer, vous gérer immédiatement le suivi des loyers et des paiements, quittances, régularisation des charges, relances... Un outil simple et efficace pour gérer un patrimoine immobilier locatif de manière autonome : appartements, maisons, garages, immeubles...

Ce logiciel s'adresse :

- aux particuliers "propriétaires bailleurs"
- aux SCI (Sociétés Civiles Immobilières)

La version Pro s'adresse aux professionnels ayant une activité de gestion locative :

- gestionnaires de biens,
- agences immobilières,
- gestionnaires de résidences.

IMMOBILIER LOYER

Vous parcourez la démo du logiciel de gestion locative Immobilier Loyer. Cliquez pour plus d'informations.

Loyers et paiements

Page 1

Ajouter un Nouveau Loyer Exporter

Recherche: Recherche

Locataire : M. LOCATAIRE Exemple Paiements à jour

Locataire	Local	Loyer	Montant total	Echéance ↓	Paiement
M. LOCATAIRE Exemple	Appartement EXEMPLE1 01 rue EXEMPLE 10 000 VILLE	février 2018	557,00€	01/02/2018	En attente (557,00€)
M. LOCATAIRE Exemple	Appartement EXEMPLE1 01 rue EXEMPLE 10 000 VILLE	janvier 2018	557,00€	01/01/2018	Payé
M. LOCATAIRE Exemple	Appartement EXEMPLE1 01 rue EXEMPLE 10 000 VILLE	décembre 2017	557,00€	01/12/2017	Payé

Figure 2: Immobilier Loyer

L'application Immobilier Loyers (QT Web) présente certains avantages et inconvénients

Les Avantages :

- Suivre des paiements et arriérés
- Gestion des contrats
- Etablissement de statistiques
- Convivialité et gain de temps
- Etc...

Les inconvénients :

- Application Desktop
- Inaccessibilité au client
- Application payantes
- Pas contrôle de l'état
- Etc...

IV.5.2 GalleSenegal

C'est une plateforme de gestion locative permettant de gérer des biens locatifs en toute simplicité. Avec GalleSenegal, vous gérez immédiatement les appels des loyers et la saisie des annonces.

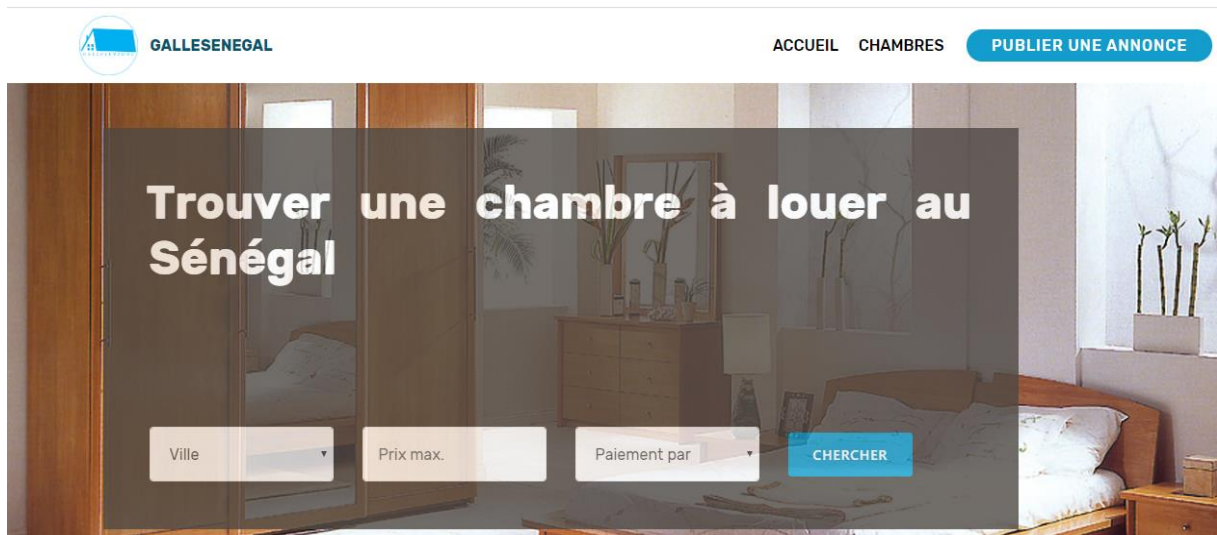


Figure 3: GALLESENEGAL

Les Avantages :

- Accessibilité des clients
- Application Web
- Gain de temps des clients
- Etc...

Les inconvénients

- Application payante
- Pas de gestion des contrats
- Pas des paiements
- Pas Contrôle de l'état
- Accessibilité de toute personne pour faire des annonces
- Etc...

IV.4 PROBLEMATIQUE

Après l'expansion époustouflante du couplage des technologies Web-Mobile, cette dernière occupe maintenant sa place aux services associés.

En effet, force est de constater que la croissance exponentielle des nouvelles technologies de l'information et de la communication, singulièrement dans le domaine de la téléphonie associée au web a suscité une mutation dans la création de logiciels. Des résultats encourageants obtenus avec les Smartphones et les ordinateurs de calculateurs très puissants avec le concept de parallélisme, ayant suscité un réel engouement, auprès des sociétés créatrices de solutions logicielles. Cependant, ce nouveau domaine demeure largement inexploré pour prendre en charge les activités liées au secteur immobilier dans nos pays africains. Le domaine immobilier est l'un des secteurs les plus dynamiques pour l'économie sénégalaise. Ce serait une valeur ajoutée si l'on arrivait à introduire les services permettant de faciliter aux locataires la recherche de location, la gestion des conditions de logements définie dans les contrats, la centralisation et la supervision des données composites d'un tel système auprès des différents intervenants.

IV.5 OBJECTIF ET METHODOLOGIE

IV.5.1 Objectif

Notre objectif vise principalement à mettre en place une solution pour la gestion et le suivi des loyers afin :

- ✓ De permettre un suivi entre les acteurs
- ✓ De Mettre en place un système d'information centralisant les données des agences immobilières
- ✓ Une recherche de locaux rapidement sans se déplacer
- ✓ Mobilité réduite
- ✓ Contrôler les agences et bailleurs

Ces objectifs atteints, nous permettront de disposer d'un logiciel de gestion avec des interfaces conviviales correspondantes aux résultats suivants :

- L'authentification des utilisateurs pour la sécurité applicative
- L'ajout d'un nouveau local
- L'ajout d'un contrat de location
- Le règlement des paiements de factures
- La recherche de local suivant différents critères
- La remonté de suggestion ou de réclamation
- La gestion des utilisateurs
- La Mise à jour sur les contrats en vigueur
- La Géolocalisation des logements des bailleurs suivant les sites

IV.5.2 Méthodologie

Pour atteindre les objectifs que nous nous sommes fixés, nous allons suivre la méthodologie suivante :

- ✚ Nous allons dans un premier temps faire une spécification et analyse des besoins fonctionnels ;
- ✚ Ensuite conception du système ;
- ✚ Et enfin nous terminerons par la mise en œuvre de la plateforme.

PARTI 2 : ANALYSE ET CONCEPTION

Dans cette partie, il sera question d'analyser l'ensemble des besoins exprimés par les clients et de proposer des solutions en mettant en place des modèles d'analyse nécessaires pour la réalisation de notre application.

Notre travail s'appuiera essentiellement sur le langage UML (Unified Modeling Language) dont la traduction française est « langage de modélisation unifié ». C'est un langage de modélisation purement orienté objet. Cependant, certaines de ces méthodes sont les suivantes :

Quelques méthodes à objets dites « Orientées objet »:

- ✓ **OOD (G. Booch) 1991**
- ✓ **OOA/OOD (T. Coad & E. Yourdon) 1991**
- ✓ **OMT (J. Rumbaugh et al.) 1991**
- ✓ **OOSE (I. Jacobson et al.) 1992**
- ✓ **OOM (M. Bouzeghoub, A. Rochfeld) 1994**

La notation UML (Booch, Jacobson, Rumbaugh) 1998 La notation UML repose sur deux concepts essentiels :

- ✚ **La modélisation du monde réel au moyen de l'approche orientée objet ;**
- ✚ **L'élaboration d'une série de diagrammes facilitant l'analyse et la conception du système d'information, et permettant de représenter les aspects statiques et dynamiques du domaine à modéliser et à informatiser.**

L'approche orientée objet a pour but de représenter les objets du monde réel sous forme de classes et objets. Une classe est une abstraction du monde réel qui regroupe un ensemble d'objets ayant des caractéristiques et des comportements communs. Chaque classe est définie par un ensemble d'attributs (qui représentent les caractéristiques ou propriétés de la classe), et un ensemble d'opérations qu'elle peut exécuter. Ces opérations représentent un ensemble de traitements. Un objet est une unité élémentaire d'une classe. Il hérite des propriétés de la classe à laquelle il appartient, et il est caractérisé par les valeurs affectées à ces propriétés.

Classes et objets sont reliés entre eux par différents types de relations :

- ✓ La relation d'instanciation : un objet est une instance d'une classe.
- ✓ La relation d'association qui représente les connexions entre les objets de différentes classes. Cette relation précise combien d'objets d'une classe peuvent être liés à l'objet d'une classe associée. Ces associations peuvent être bidirectionnelles.
- ✓ La relation de généralisation / spécialisation qui permet d'identifier, parmi les objets d'une classe, un sous-ensemble d'objets (sous-classe) ayant des propriétés spécifiques.
- ✓ Les relations d'agrégation et de composition : L'agrégation est une sorte d'association qui met en relation une classe agrégat et une classe agrégée. Cette dernière étant considérée comme une partie de la classe agrégat. La composition est un cas particulier d'agrégation.

L'utilisation des méthodes de conception ci-dessus que soient, au préalable, clairement identifiés les besoins des applications et que l'on ait la maîtrise de leur évolution. Or, en matière d'analyse financière ou de données complexes, les besoins s'identifient le plus souvent en fonction des données dont on dispose, il existe aussi d'autres méthodes d'analyse qui s'alignent à UML comme la méthode Shlaer et Mellor.

I IDENTIFICATION DES ACTEURS

Un acteur est l'archétype de l'utilisateur (personne, processus externe, ...) qui interagit avec le système. Par défaut, c'est un acteur principal, c'est-à-dire qu'il agit directement sur le système et en attend des résultats ou bien, il peut être un acteur secondaire qui est souvent sollicité pour des informations supplémentaires.

Dans notre cas, le tableau ci-dessous détaille l'ensemble de nos acteurs agissant sur le système.

Tableau 1: identification des acteurs

Acteurs	Descriptions	Rôles
Administrateurs	une agence du ministère ou un démembrement	Il gère la partie administration de la plateforme
Responsable agence	C'est lui qui gère l'agence immobilière	Créer compter des bailleurs, gérer le paiement des locataires, suivre la déclaration des locataires, suivre l'état des locaux,
Bailleurs	Le propriétaire des locaux	ajoute locaux, suivi le paiement des locataires
Locataires	Le preneur des locaux	Recherches, logements, visualisation de leurs factures pour le suivi et réclamations

II IDENTIFICATION DES FONCTIONNALITES

Les fonctionnalités sont les actions ou services pouvant être rendus par le système en cas de sollicitation de l'acteur concerné. Ces services peuvent être externes (nécessitant un acteur) ou bien, interne, c'est-à-dire non-visible, mais fonctionnant en interne s'il y a le démarrage d'un processus donné. Dans notre étude, nous avons pu lister un ensemble de fonctionnalités pouvant mener à bien l'organisation et le processus de gestion du loyer dans notre système. Le tableau ci-dessous montre l'ensemble des fonctionnalités.

Tableau 2: Identification des fonctionnalités

Fonctionnalités	Acteur(s)
S'authentifier	Administrateur, Responsables Agence, Bailleurs, Locataires
Insère le COCC	Administrateur
Créer compte responsable agence	Responsables Agence
gérer le paiement des locataires	Responsables Agence
Valider compte Bailleur et client	Responsables Agence
Insère surface corrige	Responsables Agence
Faire l'état des lieux à l'entrée et à la sortie du preneur	Responsables Agence
Ajoute local	Bailleur
suivre la déclaration des locataires	Administrateur, Responsables Agence, Bailleur
visualisation des quittances	Locataires
Réclamations	Locataires
Réserver logement	Locataires
Créer compte client	Locataires

III CAS D'UTILISATION

Bien souvent, la maîtrise d'ouvrage et les utilisateurs ne sont pas des informaticiens. Il leur faut donc un moyen simple pour exprimer leurs besoins. C'est précisément le rôle des diagrammes de cas d'utilisation qui permettent de recueillir, d'analyser et d'organiser les besoins, et de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML d'analyse d'un système.

Un diagramme de cas d'utilisation est un diagramme UML qui fournit une représentation graphique des exigences du système, il aide à identifier la façon dont les utilisateurs interagissent avec ce dernier.

Il permet ainsi de recueillir, d'analyser et d'organiser les besoins, et de recenser les principales fonctionnalités d'un système. Il montre aussi les interactions existant entre acteurs et cas d'utilisation, entre cas d'utilisation en liaison directe ou indirecte avec les acteurs.

- ❖ Les acteurs : Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Ce rôle décrit les besoins et les capacités de l'acteur. L'activité du système a pour but de satisfaire les besoins de l'acteur. Il se représente par un petit bonhomme sous-titré par le nom de l'acteur.



Figure 4: Représentation Acteur

- ❖ Cas d'utilisation : Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service, c'est un classificateur qui modélise une fonctionnalité d'un système ou d'une classe. Il se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom.

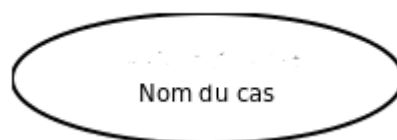


Figure 5: représentation du cas

III.1 LE DIAGRAMME DE CAS D'UTILISATION

Le diagramme de cas d'utilisation permet de représenter les fonctionnalités auxquelles les acteurs ont accès à notre système suivant selon le droit de chacun. Il est important de noter que l'exécution de tout cas d'utilisation nécessite, au préalable, une authentification sauf la recherche de local.

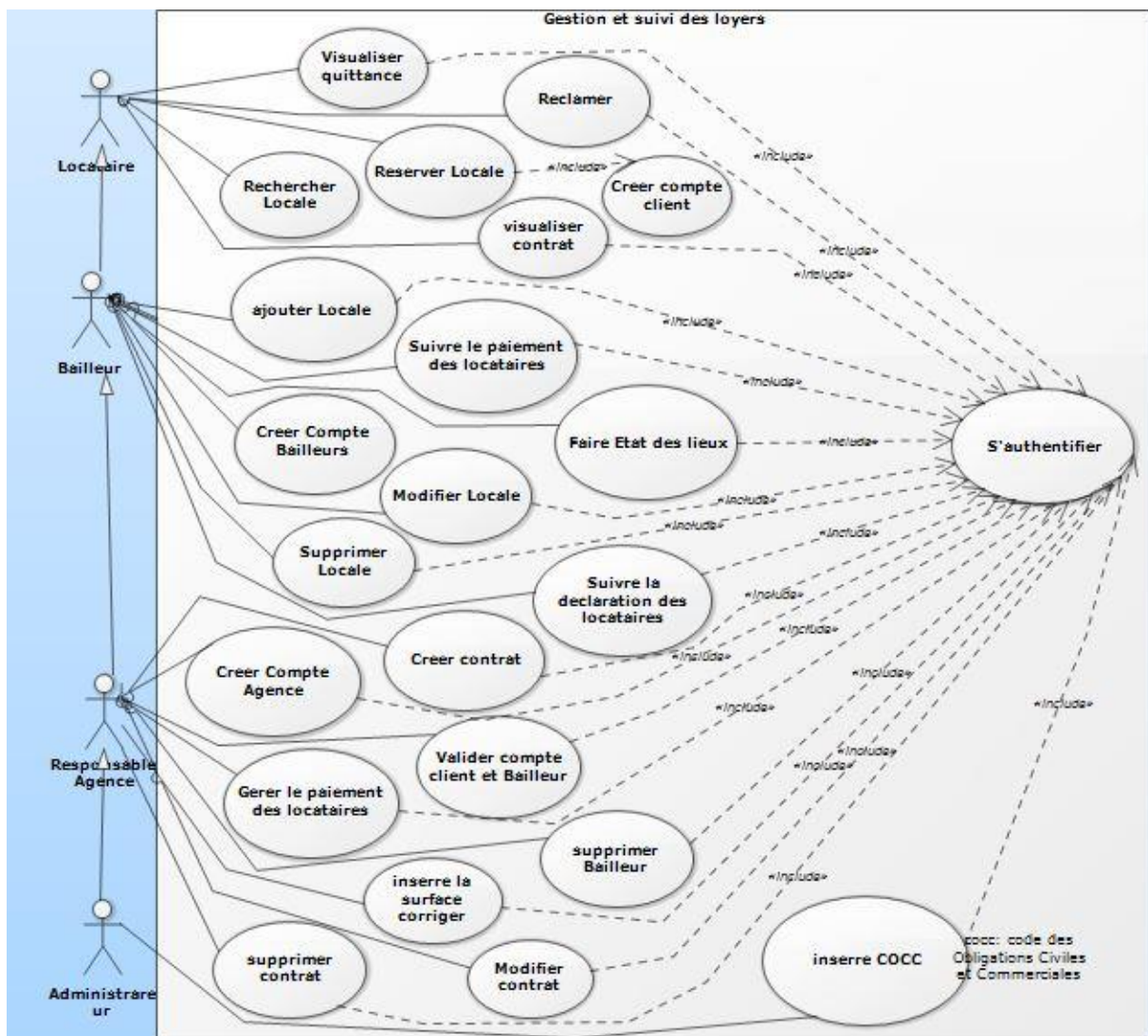


Figure 6: Diagramme de Cas D'utilisation

IV ANALYSE DES BESOINS FONCTIONNELS DU SYSTEME

L'analyse fonctionnelle est une démarche qui consiste à analyser les fonctions remplies par le système compte tenu du but à atteindre. Chaque cas d'utilisation présente tout un ensemble de processus dans son fonctionnement. Dans cette partie, nous analysons les besoins fonctionnels qui semblent les plus complexes. D'abord, nous avons fait la description des cas d'utilisation et en suite la représentation des diagrammes de séquence et d'activité de quelques cas d'utilisation.

IV.1 DESCRIPTION DES CAS D'UTILISATION

La description des cas d'utilisations permet de clarifier le déroulement de la fonctionnalité et décrire la chronologie des actions qui devront être réalisées. Dans cette partie, la description des cas sera faite sous forme de tableau.

IV.1.1 Description du cas d'utilisation « s'authentifier »

Dans cette partie, nous avons la description du cas d'utilisation s'authentifier

Tableau 3: Description du cas d'utilisation « s'authentifier »

Description du cas d'utilisation « s'authentifier »	
Nom	<i>S'authentifier</i>
Acteurs	Administrateur, Responsable Agence, Bailleur, Locataire
Description	Ce cas d'utilisation permet de vérifier l'accès avant manipulation du système.
Préconditions	Avoir un compte d'utilisateur
Scénario nominal	
1	Lancer l'application pour saisir son login et mot de passe
2	Le système vérifie les informations saisies
3	Le système lui envoie un menu correspondant
Scénarios alternatifs	
2a	L'utilisateur saisit un login et mot de passe incorrect où il devra réessayer la saisie
3a	L'accès sera refusé à toute personne qui n'a pas de compte dans le système
Post condition (aucun)	
Fin (fin à l'étape 3, 2a, 3a)	

IV.1.2 Description du cas d'utilisation « ajoute locale »

Tableau 4: Description du cas d'utilisation « ajoute locale »

Description du cas d'utilisation « ajoute locale »	
Nom	<i>Ajoute Locale</i>
Acteurs	Bailleur
Description	Ce cas d'utilisation nous permet d'ajouter un local
Préconditions	S'authentifier
Scénario nominal	
1	L'authentification réussie, l'utilisateur sera redirigé vers le menu Bailleur
2	L'utilisateur click sur le bouton ajoute locale
3	Le système lui envoi le formulaire de ajoute locale
4	L'utilisateur remplit le formulaire et valide
5	Le système enregistre les données sur la base de données
Scénario alternatifs	
1a	En cas ou l'authentification invalide, l'utilisateur n'accédera pas au menu
Post condition (aucun)	
Fin (fin à l'étape 5)	

IV.1.3 Description du cas d'utilisation « ajoute contrat »

Tableau 5: Description du cas d'utilisation « ajoute contrat»

Description du cas d'utilisation « ajoute contrat»	
Nom	<i>Ajoute Contrat</i>
Acteurs	Responsable Agence
Description	Ce cas d'utilisation nous permet d'ajouter un contrat
Préconditions	S'authentifier
Scénario nominal	
1	L'authentification réussie, l'utilisateur sera redirigé vers le menu Agence
2	L'utilisateur click sur le bouton ajoute contrat
3	Le système lui envoi le formulaire d'ajoute contrat
4	L'utilisateur click sur le bouton valider
5	Le système enregistre les données sur la basse de donne
Scénario alternatifs	
1a	En cas ou l'authentification invalide, l'utilisateur n'accédera pas au menu
Post condition (aucun)	
Fin (fin à l'étape 5)	

IV.1.4 Description du cas d'utilisation « réserver local»

Tableau 6:Description du cas d'utilisation « réserver local »

Description du cas d'utilisation « réserver locale »	
Nom	<i>Réserver locale</i>
Acteurs	Locataire
Description	Ce cas d'utilisation nous permet de réserver un local
Préconditions	Aucun
Scénario nominal	
1	L'utilisateur click sur le bouton réserver locale
2	Le système lui envoi le formulaire de création de compte
3	L'utilisateur remplit le formulaire et click sur le bouton valider
4	Le système enregistre les données sur la base de données, lui valide le local réservé et lui envoie le contrat
5	L'utilisateur imprime le contrat
Scénario alternatif	
3a	Le système envoie un message d'erreur
4a	Le système invalide le locale réserver
Post condition (aucun)	
Fin (fin à l'étape 5)	

IV.1.5 Description du cas d'utilisation « Réclamation »

*Tableau 7: Description du cas d'utilisation « Réclamation »***Description du cas d'utilisation « réclamation »**

Nom	<i>Réclamation</i>
Acteurs	Locataire
Description	Ce cas d'utilisation nous permet d'envoyer une réclamation
Préconditions	S'authentifier

Scénario nominal

1	L'utilisateur clique sur le bouton réclamation
2	Le système lui envoie le formulaire de la réclamation
3	L'utilisateur remplit le formulaire et clique sur le bouton valider
4	Le système enregistre le message dans la base de données et lui envoie succès

Scénario alternatif

1a	En cas où l'authentification est invalide, l'utilisateur n'accèdera pas au menu
----	---

Post condition (aucun)**Fin** (fin à l'étape 4)

IV.2 LE DIAGRAMME DE SEQUENCE

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs, en montrant les messages qu'ils échangent entre eux ordonnés dans le temps, les composants d'un diagramme de séquence sont les suivants :

- ❖ Les objets au cœur d'un système interagissent en s'échangeant des messages. Ils apparaissent toujours dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction.
- ❖ Le message élément de communication unidirectionnel entre objets qui déclenche une activité dans l'objet destinataire. La réception d'un message provoque un événement dans l'objet récepteur. La flèche pointillée représente un retour au sens UML. Cela signifie que le message en question est le résultat direct du message précédent.
- ❖ Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-machine).

IV.2.1 Diagramme de séquence du cas « ajouter local »

La figure ci-dessous nous présente les étapes à suivre pour l'ajout d'un local par le bailleur. Il s'authentifie en saisissant son login et mot de passe d'abord après vérification des informations, si tout est correct le système lui redirige vers son menu. Le bailleur va choisir l'option ajouté local ensuite un formulaire lui sera envoyé ; il va saisir les informations du local et valider.

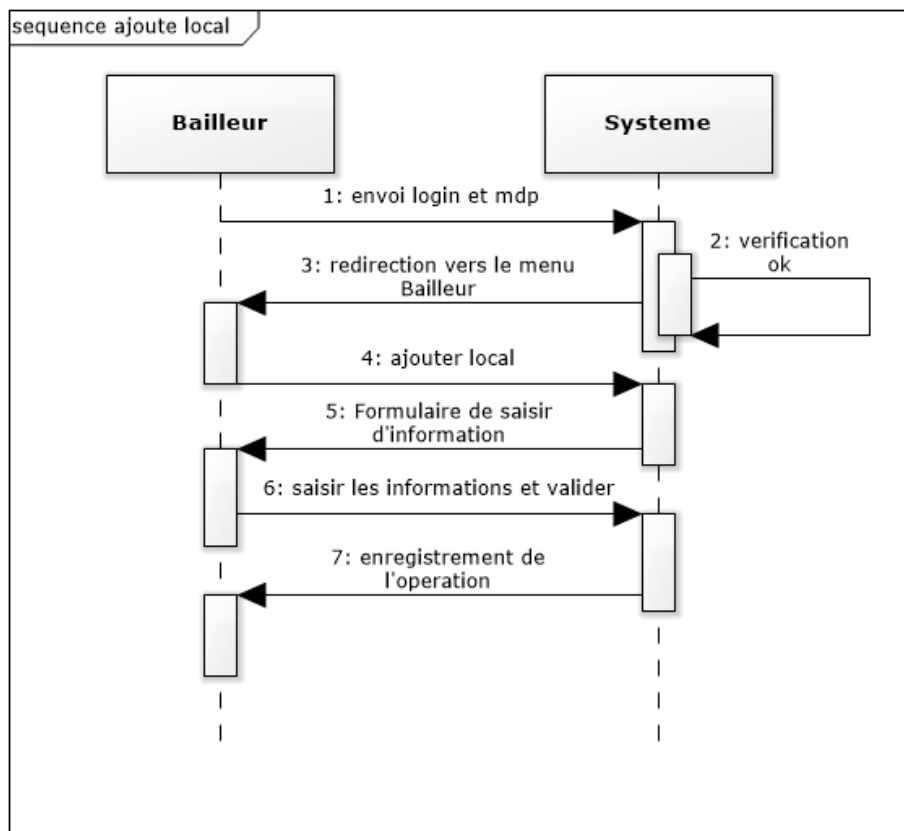


Figure 7: diagramme de séquence du cas « ajouter locale »

IV.2.2 Diagramme de séquence du cas « ajouter contrat »

La figure ci-dessous nous présente les étapes à suivre pour l'ajoute contrat par le responsable Agence. Il s'authentifie en saisissant son login et mot de passe d'abord après vérification des informations, si tout est correct le système lui redirige vers son menu agence. Il va choisir l'option ajoutée contrat ensuite un formulaire lui sera envoyé ; il insère le contrat du local et valide.

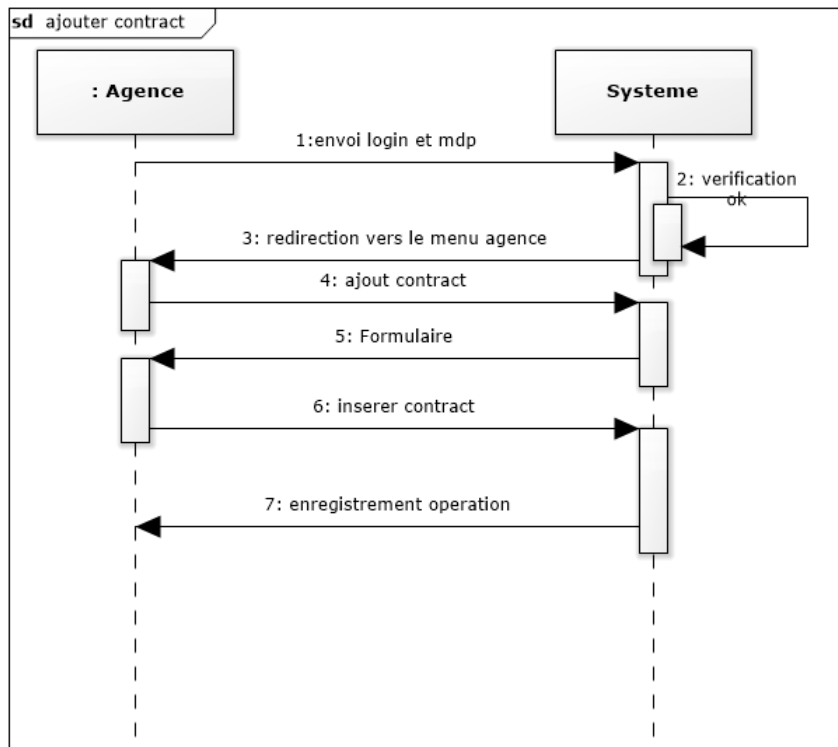


Figure 8: Diagramme de séquence du cas « Ajouter Contrat »

IV.2.3 Diagramme de séquence du cas « reserver local »

La figure ci-dessous nous présente les étapes à suivre pour faire une réservation par le client. Il visualise le local, clique le bouton détail ensuite faire la réservation et un formulaire de création de compte est envoyé par le système et ce formulaire sera remplie par le client une fois terminer valider. Il va s'authentifier en saisissant son login et mot de passe d'abord après vérification des informations, si tout est correct le système lui redirige vers le menu locataire. Le client va choisir l'option envoi réservation ensuite le système valide la réservation et lui envoie le contrat.

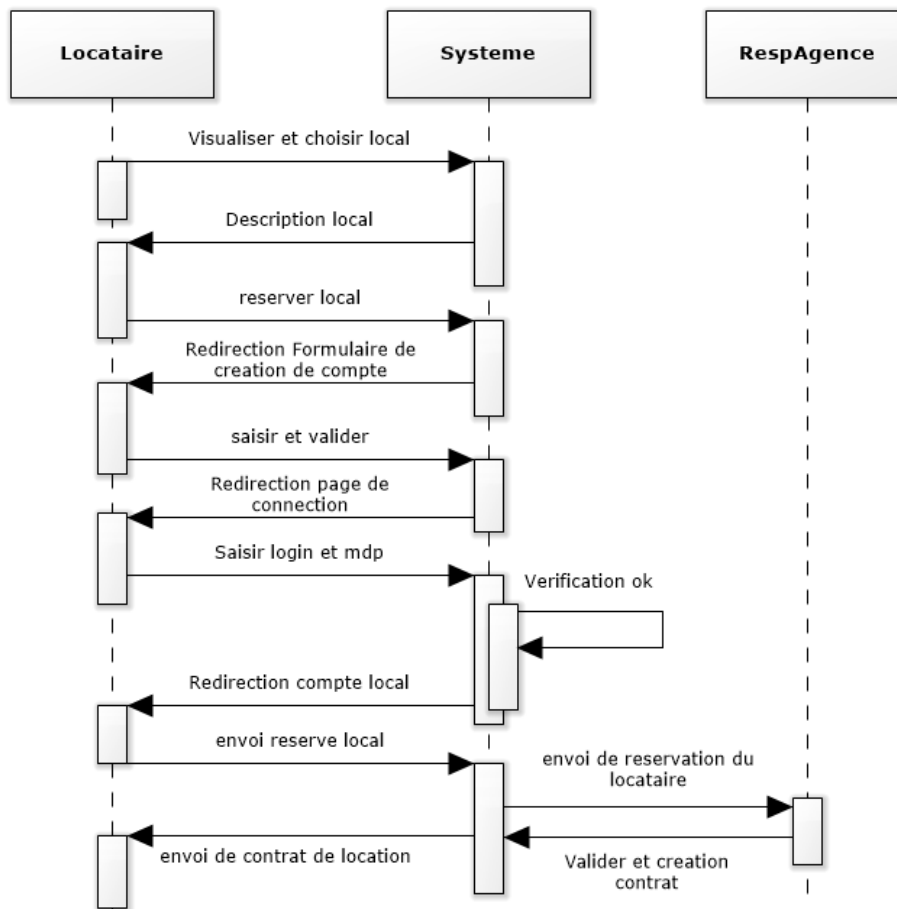


Figure 9: Diagramme de séquence du cas « réserver local »

IV.2.4 Diagramme de séquence du cas « réclamation »

La figure ci-dessous nous présente dans cette partie les étapes à suivre pour faire une réclamation par le locataire. Il s'authentifie en saisissant son login et mot de passe d'abord après vérification des informations, si tout est correct le système lui redirige vers son menu. Le locataire va choisir l'option réclamation ensuite un formulaire lui sera envoyé ; il saisir son message puis envoyer.

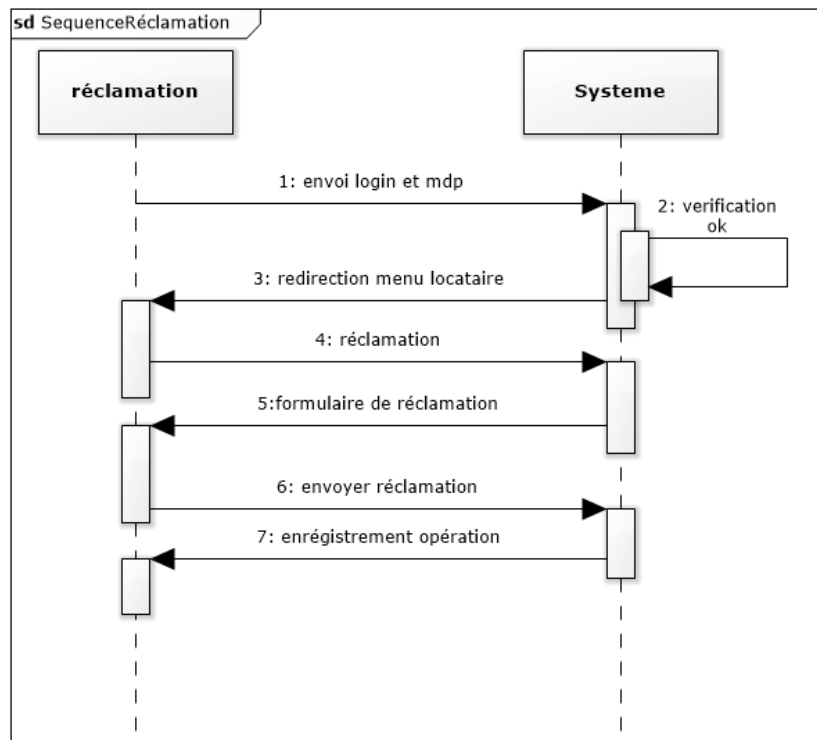


Figure 10: Diagramme de séquence du cas « réclamation »

IV.3 LE DIAGRAMME D'ACTIVITE

Dans cette partie, nous allons brièvement voir l'alternative visuelle des descriptions détaillées des cas d'utilisation. Il s'agit du diagramme d'activité.

La description d'un cas d'utilisation peut contenir plusieurs scénarii alternatifs et/ou d'exception. Il est alors difficile d'avoir une vision de l'ensemble des actions. Le diagramme d'activité est un moyen graphique pour donner cette vision d'ensemble.

Il est donc particulièrement adapté à la modélisation du cheminement de flots de contrôle et de flots de données.

De plus, leur représentation sous forme d'organigrammes les rend facilement intelligibles et beaucoup plus accessibles que les diagrammes d'états-transitions. La modélisation du flot d'objets est souvent importante dans ce type d'utilisation des diagrammes d'activités.

Les composants d'un diagramme d'activité sont les suivants :

❖ Point de démarrage et d'arrêt

Le diagramme est composé d'un point de démarrage, d'un point arrêt et d'action, qui est représenté par des cercles.

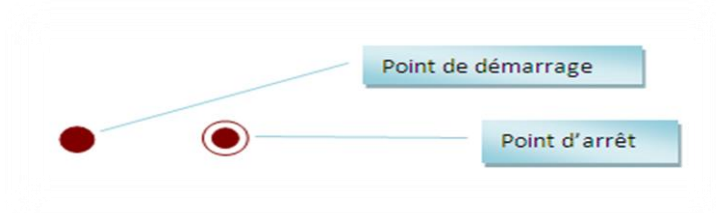


Figure 11: Point de démarrage et d'arrêt

❖ Les actions et les transitions

Le diagramme est donc organisé en actions réalisées soit par un acteur, soit par le système, relié par une flèche indiquant l'enchaînement des actions.

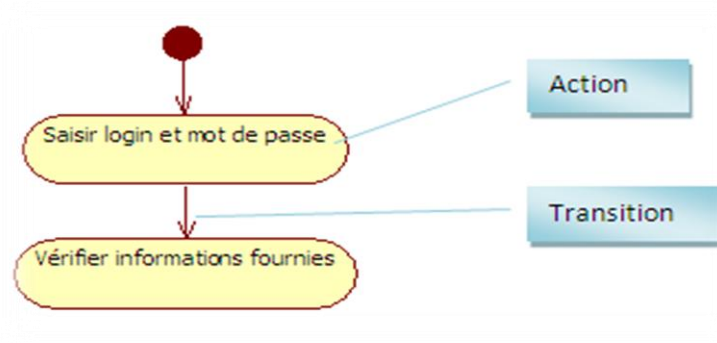


Figure 12: actions / transitions

❖ L'alternative

Elle permet d'indiquer les différents scénarii du cas d'utilisation dans un même diagramme. Il s'agit de la condition d'après laquelle le cas d'utilisation serait appelé.

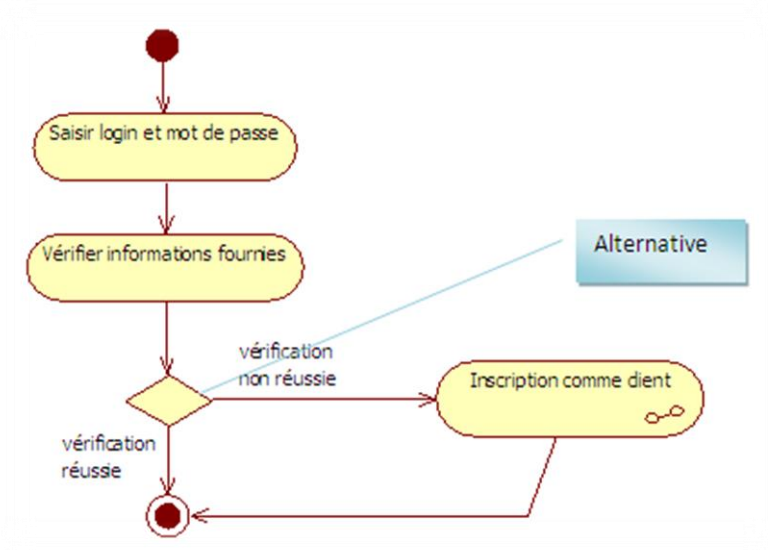


Figure 13: Alternative

❖ La synchronisation

Elle indique qu'il faut avoir réalisé deux actions pour pouvoir réaliser la troisième en dessous

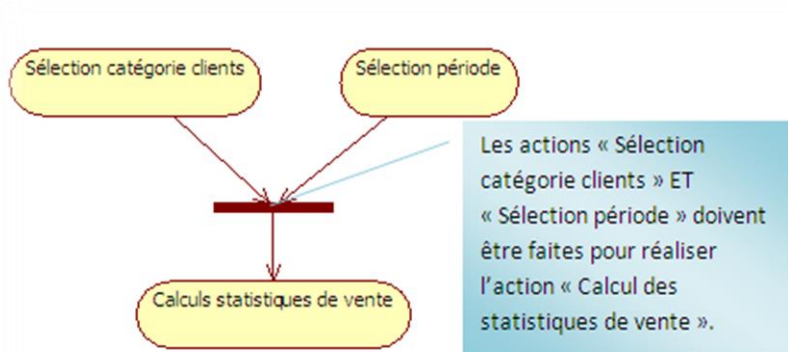


Figure 14: synchronisation

IV.3.1 Diagramme d'activité du cas « Ajouter local »

Dans cette partie de ce cas d'utilisation, nous commençons à s'authentifier en saisissant le login et le mot de passe, si les informations sont correctes, le système sera dirigé vers le «menu Bailleur». Une fois, cela est fait, « cliquer sur le bouton ajouter local ». Après l'exécution de ces activités, l'utilisateur met en ligne le local. Dans ce cas l'activité « ajouter local » s'exécute puis celle de l'enregistrement du local. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

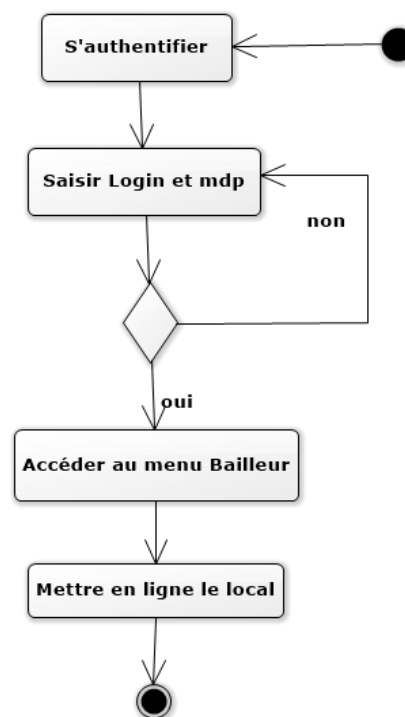


Figure 15: Diagramme d'activité du cas « Ajouter local »

IV.3.2 Diagramme d'activité du cas « Ajouter Contrat »

Dans cette partie, l'utilisateur s'authentifie en saisissant le login et le mot de passer, si les informations sont correctes, l'utilisateur sera dirigé vers le « menu agence ». Une fois, cela est fait, « cliquer sur le bouton ajouter contrat ». Après l'exécution de ces activités, l'utilisateur met en ligne le contrat. Dans ce cas l'activité « ajouter contrat » s'exécute puis celle de l'enregistrement du contrat. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

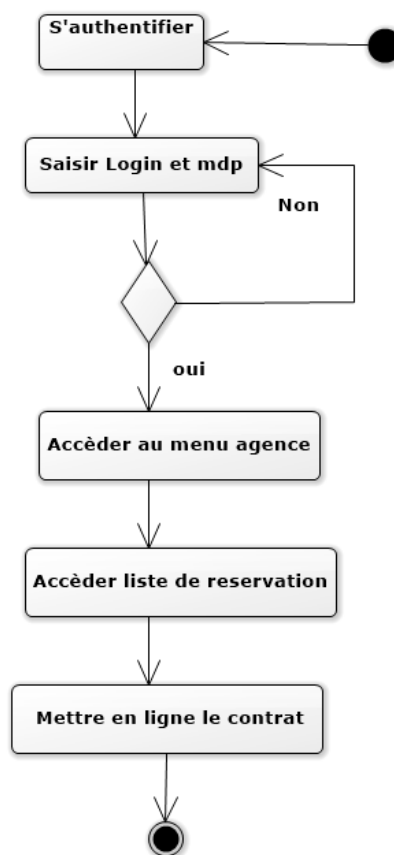


Figure 16: Diagramme d'activité du cas « Ajouter Contrat »

IV.3.3 Diagramme d'activité du cas « Réserver local »

Dans cette partie, l'utilisateur visualise le local et voit les détails ensuite il procède à réserver le local et il sera dirigé vers le formulaire de création compte local, une fois terminé il sera redirigé vers la page de connexion puis il s'authentifie en saisissant le login et le mot de passe de son compte, si les informations sont correctes, l'utilisateur sera dirigé vers le « menu

locataire ». Une fois, cela est fait, « cliquer sur le bouton ajouter réserver ». Après l'exécution de ces activités, le système valide la réservation et lui envoie un contrat. Dans ce cas l'activité « Réserver local » s'exécute puis l'utilisateur visualise son contrat. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

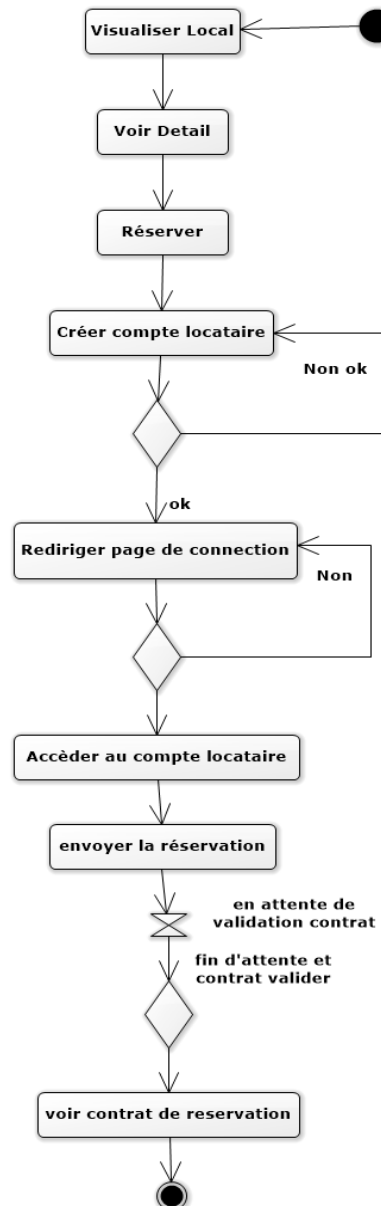


Figure 17: Diagramme d'activité du cas « Réserver local »

IV.3.4 Diagramme d'activité du cas « Réclamation »

Dans cette partie, l'utilisateur s'authentifie en saisissant le login et le mot de passe, si les informations sont correctes, le système sera dirigé vers le « menu locataire ». Une fois, cela est fait, « cliquer sur le bouton réclamation ». Après l'exécution de ces activités, l'utilisateur saisit son message. Dans ce cas l'activité « réclamation » s'exécute. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité ci-dessous.

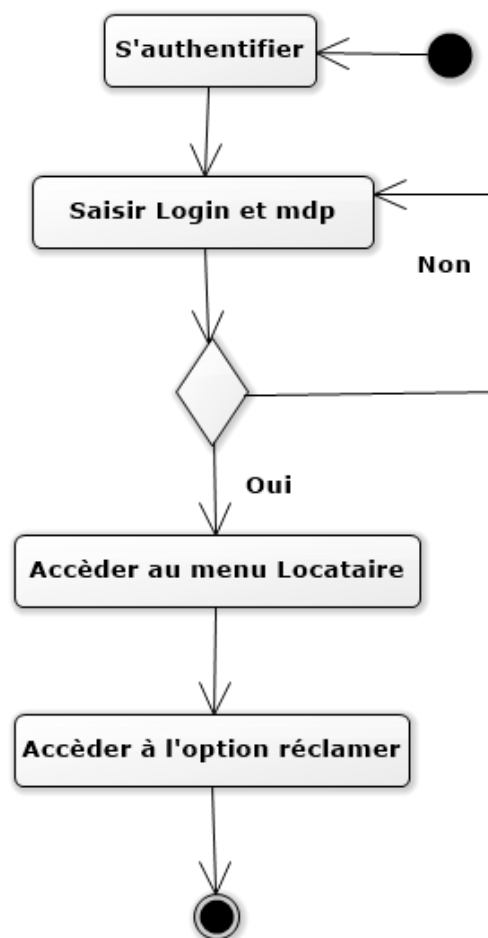


Figure 18: Diagramme d'activité du cas « Réclamation »

V CONCEPTION DU SYSTEME

Le travail d'un informaticien, est celui d'architecte, c'est-à-dire qu'il commence par une conception pour s'achever par un accomplissement. Il est capital de comprendre les différents termes liés à notre application.

La conception est une étape très importante pour la réalisation d'une application en s'appuyant sur des modèles de conception. Plusieurs méthodes de conception coexistent et sont exploitées différemment selon les pays, dont **Merise**, **UML**, **AXIAL**, **IDEF**. Dans notre cas, nous utilisons **UML**. La conception du système comprend deux étapes que sont la conception générale et la conception détaillée.

- ✚ Conception générale, il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel, le diagramme de packages et le diagramme de déploiement.
- ✚ Conception détaillée, consistant à définir précisément chaque sous-ensemble du logiciel comme les différentes classes intervenant et le diagramme de classe.

V.1 CONCEPTION GENERALE DU SYSTEME

Dans cette étape, nous allons présenter l'architecture du système ainsi le diagramme de packages et le diagramme de déploiement.

V.1.1 Architecture De L'application

L'architecture logicielle est l'ensemble des décisions qui, lorsqu'elles sont mal prises, font que votre projet échoue. Elle décrit d'une manière symbolique et schématique les différents éléments d'un ou de plusieurs systèmes informatiques, leurs interrelations et leurs interactions. Contrairement aux spécifications produites par l'analyse fonctionnelle, le modèle d'architecture, produit lors de la phase de conception, ne décrit pas ce que doit réaliser un système informatique, mais plutôt comment il doit être conçu de manière à répondre aux spécifications. L'analyse décrit le « quoi faire » alors que l'architecture décrit le « comment le faire ». Ainsi, nous avons trois principaux modèles architecturaux qui sont le modèle **MVC**, modèle **PAC**, modèle **PAC-Amodeus**. Dans notre cas, nous choisirons le modèle MVC qui est utilisé par de nombreux Framework pour applications web.

1. Modèle MVC

Le modèle MVC (Modèle, Vue, Contrôleur) a été défini par les travaux de (Krasner et Pope 1988) et a pour objectif de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

- **Le modèle** : il contient les données fonctionnelles ainsi que les objets ayant un comportement complexe. Il modifie la vue associée à son état à chaque fois que celui-ci se trouve modifié par le noyau fonctionnel de l'application ou par le contrôleur.
- **Vue** : elle maintient une représentation de l'état du modèle perceptible par l'utilisateur et met celle-ci à jour lors des changements d'état du modèle.
- **Le Contrôleur** : il reçoit et interprète les événements correspondants aux actions de l'utilisateur en les transmettant sur le modèle en modifiant son état ou son vue en effectuant le feed-back.

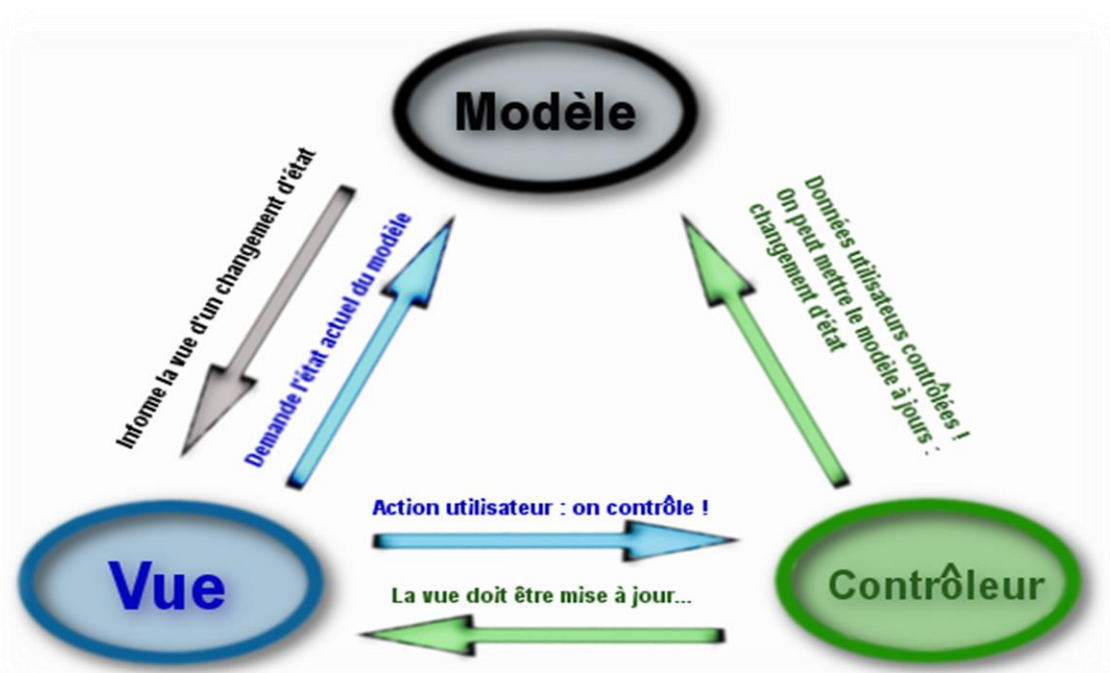


Figure 19: Architecture MVC

2. Architecture 3-tier

L'architecture trois niveaux (3-tier) est un modèle en couches, c'est-à-dire que chaque couche communique seulement avec ses couches adjacentes (supérieures et inférieures) et le flux de contrôle traverse le système de haut en bas. Les couches supérieures contrôlent les couches inférieures, c'est-à-dire que les couches supérieures sont toujours sources d'interaction (clients) alors que les couches inférieures ne font que répondre à des requêtes (serveurs).

Dans le modèle MVC, il est généralement admis que la vue puisse consulter directement le modèle (lecture) sans passer par le contrôleur. Par contre, elle doit nécessairement passer par le contrôleur pour effectuer une modification (écriture). Ici, le flux de contrôle est inversé par rapport au modèle en couches, le contrôleur peut alors envoyer des requêtes à toutes les vues de manière qu'elles se mettent à jour.

Dans l'architecture trois niveaux (3-tier), si une vue modifie les données, toutes les vues concernées par la modification doivent être mises à jour, d'où l'utilité de l'utilisation du MVC au niveau de la couche de présentation. La couche de présentation permet donc d'établir des règles du type « mettre à jour les vues concernant X si Y ou Z sont modifiés ». Mais ces règles deviennent rapidement trop nombreuses et ingérables si les relations logiques sont trop élevées. Dans ce cas, un simple rafraîchissement des vues à intervalle régulier permet de surmonter aisément ce problème. Il s'agit d'ailleurs de la solution la plus répandue en architecture trois niveaux (3-tier), l'utilisation du MVC étant moderne et encore marginale.

3. Architecture de notre Application

Pour que le client puisse se connecter à une SGBD MySQL, la solution la plus répandue est d'écrire et d'exécuter des services web à l'aide d'un Protocole http. Dans notre cas, l'application a comme l'architecture le client, le serveur web et le serveur de base de données forment une architecture 3-tiers.

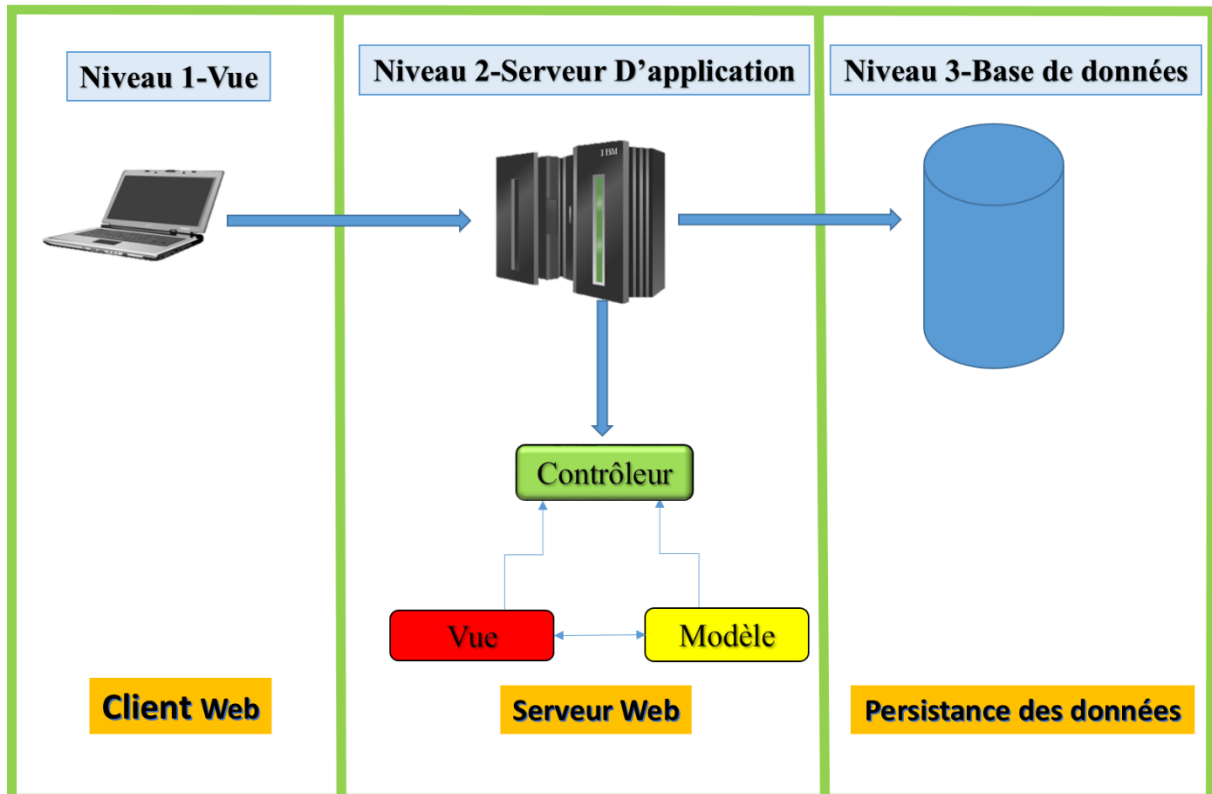


Figure 20: Architecture de l'Application

V.1.2 Diagrammes de Paquetages

Ils sont utilisés pour refléter l'organisation de paquetages et de leurs éléments. Lorsqu'il est utilisé pour représenter des éléments de classe, diagrammes de paquetages permettent de visualiser les espaces de noms. L'utilisation la plus courante pour les diagrammes de paquetages est d'organiser des Diagrammes de Cas d'Utilisation et des Diagrammes de Classes. Bien que l'utilisation des Diagrammes de Paquetages ne se limite pas à ces éléments UML.

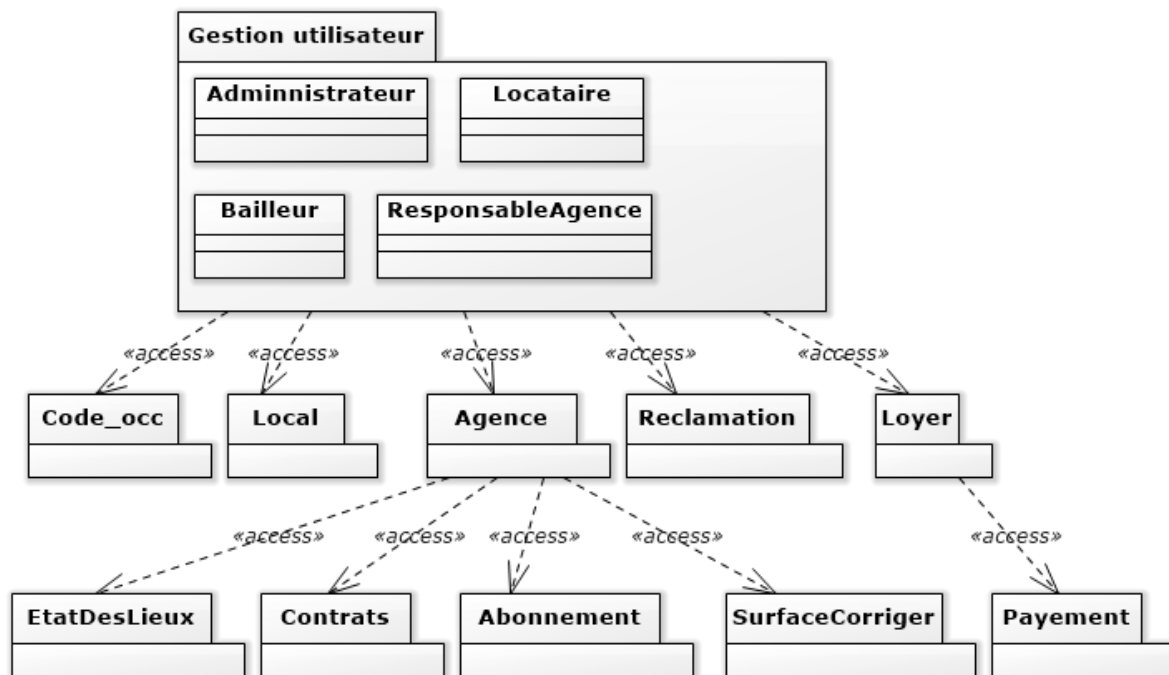


Figure 21: Diagrammes de Paquetages

V.1.3 Diagrammes de Déploiement

Un diagramme de déploiement modélise l'architecture physique de temps d'exécution d'un système. Il affiche les relations entre les composants logiciels et matériels du système, d'une part, et la distribution physique du traitement, d'autre part. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les nœuds représentent des périphériques matériels tels que des ordinateurs, des détecteurs et des imprimantes, ainsi que d'autres périphériques qui prennent en charge l'environnement d'exécution d'un système. Les chemins de communication et les relations de déploiement modélisent les connexions dans le système.

L'application doit être légère et performante ; elle peut tourner sur des machines de 2 Go de RAM et 1.80 GHZ de fréquence processeur. Les ordinateurs tiers pourront accéder au serveur contenu dans une machine d'au moins 8 Go de RAM. Ils empruntent le réseau internet.

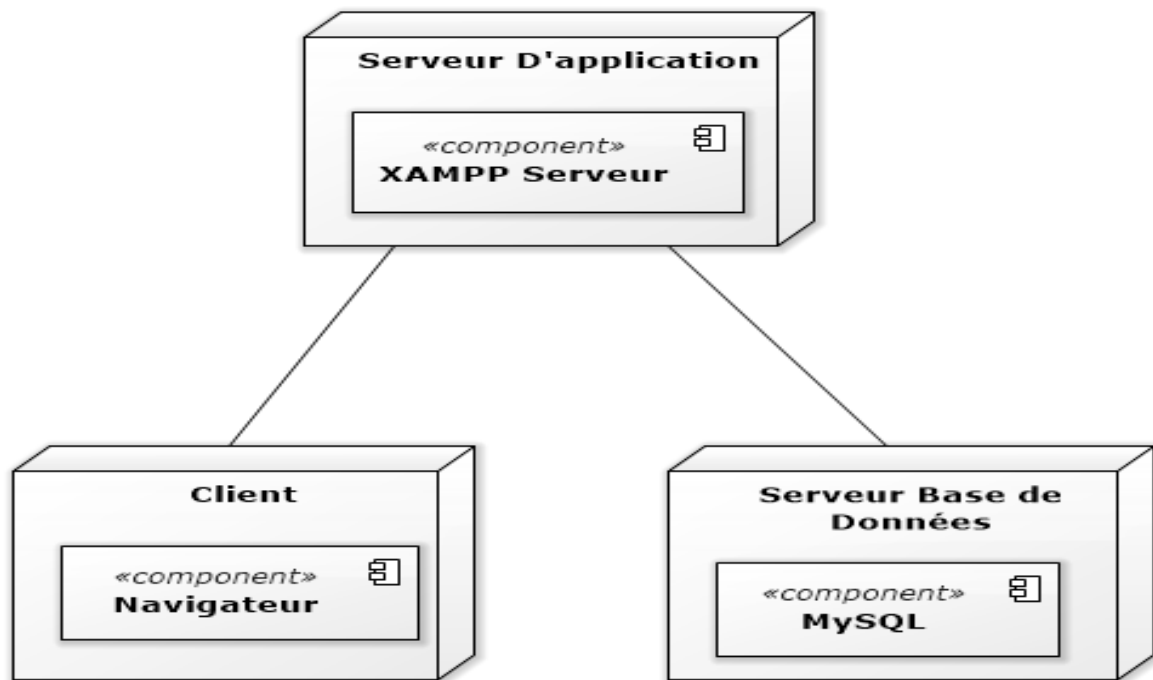


Figure 22: Diagrammes de Déploiement

V.2 CONCEPTION DETAILLEE

La conception détaillée est une phase ultime de la conception de notre système d'information. Elle consiste à construire les classes qui interviennent dans le codage de l'application. Pour cela, nous abordons le diagramme de classes ainsi que le dictionnaire de données de notre système.

V.2.1 Diagramme de classes

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une partition des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier. Chaque application qui va mettre en œuvre le système sera une instance des différentes classes qui le compose.

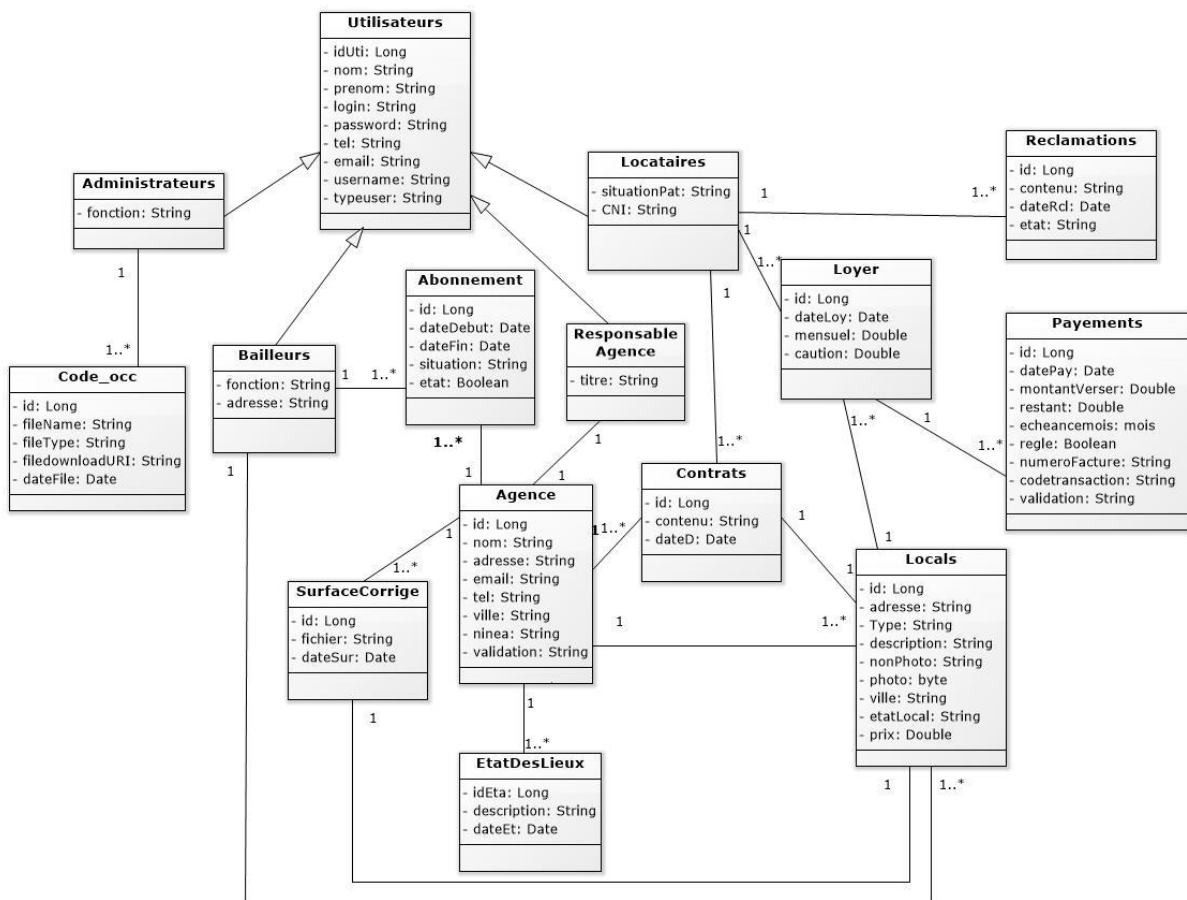


Figure 23: diagramme des classes

PARTI 3 : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir complété l'étude conceptuelle, nous allons aborder la partie implémentation dans ce qui suit. Nous commençons à présenter les outils utilisés pour notre application.

I. OUTIL DE CONCEPTION : SOFTWARE IDEAS MODELER

Les modélisations UML ne se dessinent bien pas avec Pain. Ce sont les **outils UML** qui vous aident à utiliser le langage de modélisation. Mais trouver le bon outil n'est pas si facile. Il existe d'innombrables fournisseurs de programmes UML sur Internet, mais tous n'offrent pas les mêmes fonctions. Bien que certains outils nécessitent moins de mémoire, ils possèdent également moins de fonctionnalités. Certains outils de modélisation UML peuvent dessiner et exporter n'importe quel type de modèle vers divers langages de programmation et importer un modèle à partir du code existant. Mais beaucoup de ces programmes n'offrent aucune fonction pour échanger des informations sur les projets au sein de l'équipe. Parmi ces outils, il existe de nombreuses options, des outils de dessin de base plus spécialisés comme ClickCharts de NCH Software, ArgoUML, Papyrus, UML Designer, Software Ideas Modeler. Dans notre cas, nous avons utilisé le Software Ideas Modeler qui est un créateur de diagrammes incroyablement compétent qui est livré avec des fonctionnalités professionnelles et permettant de gagner du temps, et piétine la concurrence gratuite (et de nombreux concurrents commerciaux aussi). Il supporte également entre autres les diagrammes et standards, comme les organigrammes, les diagrammes de flux de données, etc.

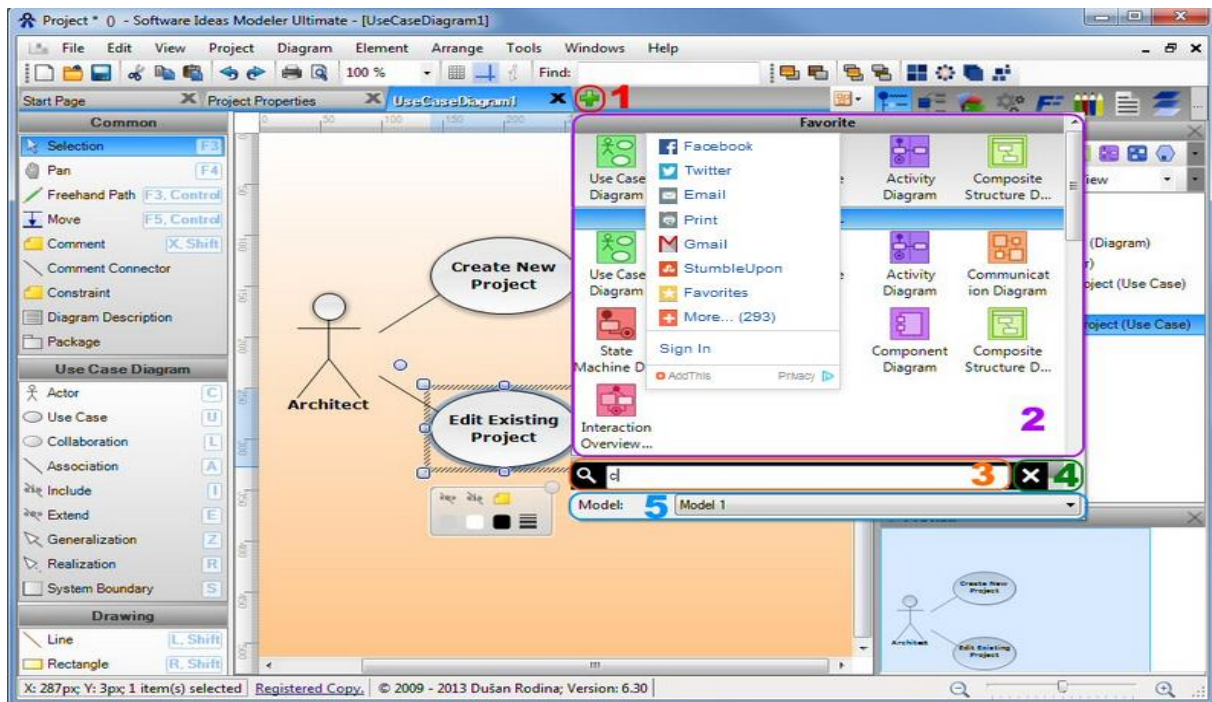


Figure 24: Software Ideas Modeler

II ENVIRONNEMENT DE DEVELOPPEMENT

II.1 PRESENTATION SPRING TOOL

Adaptée au développement d'applications d'entreprise à l'aide de Spring Framework et Spring Boot, la nouvelle génération de Spring Tools offre un support de développement de classe mondiale pour vos applications Spring. Nos outils ont une connaissance approfondie de Spring intégré. Les tout nouveaux outils Spring peuvent être utilisés dans divers environnements de codage, allant d'Eclipse en tant qu'environnement de développement intégré complet à Visual Studio Code et Theia en tant qu'éditeurs de code légers. Continuez à utiliser votre environnement préféré et ajoutez-y d'excellents outils Spring. La nouvelle génération de Spring Tools est en grande partie construite à partir de zéro, intégrant des technologies modernes et des architectures d'outils de développement. Il s'exécute dans des processus distincts, est conçu avec des performances à l'esprit dès le départ et connaît les dernières technologies Spring. De ce fait, nous utilisons Spring Tool Suite 4.

II.1.1 *Spring Tool Suite 4*

Facilite le démarrage Une intégration directe et facile à utiliser du Spring Initializr et des célèbres Spring Guides vous permet de passer de rien à une application Spring Boot en cours d'exécution en quelques secondes. La compréhension et la navigation rapide du code source sont essentielles pour le codage. Le nouveau Spring Tools 4 comprend votre code source basé sur Spring et vous permet d'obtenir rapidement une vue d'ensemble et de naviguer vers les éléments importants de vos applications Spring. Trouver des éléments Spring et y accéder n'ont jamais été aussi simple. L'achèvement du code est une partie critique du travail avec le code source. Le tout nouveau Spring Tools 4 fournit des compléments de code intelligents pour les éléments Spring de votre application. Spring Tools 4 comble désormais l'écart entre votre code source et les applications Spring Boot en cours d'exécution. En tirant parti des Spring Boot Actuators, nous avons enrichi le code source avec des informations détaillées de l'application en cours d'exécution (par exemple, les informations exactes sur le câblage du bean, les rapports conditionnels, la configuration, les détails, etc.).



Figure 25: Spring Tool Suite 4

II.2 LES SERVEURS UTILISES POUR NOTRE APPLICATION

Beaucoup de personnes ont constaté, d'après leur propre expérience, qu'il n'est pas aisé d'installer un serveur web Apache. Et c'est encore plus difficile si vous voulez ajouter MySQL, PHP et Perl. Le but recherché par XAMPP est de fournir une distribution facile à installer. Ainsi, les développeurs peuvent se lancer facilement dans le monde d'Apache. Pour le rendre encore plus pratique, XAMPP est la distribution Apache avec toutes les fonctionnalités activées. Elle contient MySQL, PHP et Perl. Il suffit de télécharger et de lancer l'installateur est livré avec si vous désirez l'utiliser dans un but commercial, merci de prendre connaissance des différentes licences. XAMPP lui-même peut être utilisé librement, dans un but commercial. Il existe actuellement des distributions pour Windows, Linux et OS X.



Figure 26: XAMPP

II.2.1 Serveur d'application Apache

Apache est un logiciel de serveur web gratuit et open-source qui **alimente environ 46% des sites web** à travers le monde. Le nom officiel est **Serveur Apache HTTP** et il est maintenu et développé par Apache Software Foundation.

Bien que nous appelions Apache un serveur web, ce n'est pas un serveur physique, mais plutôt un logiciel qui s'exécute sur un serveur. Son travail consiste à établir une connexion entre un serveur et les navigateurs des visiteurs du site web (Firefox, Google Chrome, Safari, etc.) tout en délivrant des fichiers entre eux (structure client-serveur). Apache est un logiciel multiplateforme, il fonctionne donc à la fois sur les serveurs Unix et Windows. Apache est hautement personnalisable, car il a une structure basée sur des modules. Les modules permettent aux administrateurs de serveur d'activer ou de désactiver des fonctionnalités supplémentaires. Apache possède des modules pour la sécurité, la mise en cache, la réécriture d'URL, l'authentification par mot de passe et encore plus. Vous pouvez également configurer vos propres configurations du serveur via un fichier appelé **.htaccess**, qui est un fichier de configuration Apache pris en charge avec tous les plans **Hostinger**.

II.2.2 SGBD (MySQL)

Un système de gestion de base de données (SGBD) est le logiciel qui permet à un ordinateur de stocker, récupérer, ajouter, supprimer et modifier des données. Un SGBD gère tous les aspects primaires d'une base de données, y compris la gestion de la manipulation des données, comme l'authentification des utilisateurs, ainsi que l'insertion ou l'extraction des données. Un SGBD définit ce qu'on appelle le schéma de données ou la structure dans laquelle les données sont stockées. Il existe des principaux SGBD : MySQL, MariaDB, Apache Derby, SQLite. Dans notre cas, nous avons utilisé le MySQL.

MySQL est un serveur de base de données relationnelles SQL qui fonctionne sur de nombreux systèmes d'exploitation (dont Linux, Mac OS X, Windows, Solaris, FreeBSD...) et qui est accessible en écriture par de nombreux langages de programmation, incluant notamment PHP, Java, Ruby, C, C++, NET, Python.

L'une des spécificités de MySQL c'est qu'il inclut plusieurs moteurs de bases de données et qu'il est par ailleurs possibles au sein d'une même base de définir un moteur différent pour les tables qui composent la base. Cette technique est astucieuse et permet de mieux optimiser les performances d'une application. Les 2 moteurs les plus connus étant **MyISAM** (moteur par défaut) et **InnoDB**.

La réplication est possible avec MySQL et permet ainsi de répartir la charge sur plusieurs machines, d'optimiser les performances ou d'effectuer facilement des sauvegardes des données.

III. LES TECHNOLOGIES UTILISEES A L'IMPLEMENTATION

Dans notre application, nous avons repartir l'implémentation en deux parties Back-end et Front-End.

III.1 TECHNOLOGIES UTILISEES POUR LA PARTIE BACK-END

Le Back-End, c'est un peu comme la partie immergée de l'iceberg. Elle est invisible pour les visiteurs, mais représente une grande partie du développement d'un projet web. Sans elle, l'application reste une coquille vide. Pour l'implémentation de la partie Back-end de notre application, nous avons utilisé des technologies comme Spring boot et Thymeleaf.

III.1.1. Présentation de Spring boot

Spring Boot permet de créer facilement des applications Spring autonomes de niveau production que vous pouvez «simplement exécuter». Il fournit une bonne plate-forme aux développeurs Java pour développer une application packagée en jar, totalement autonome.

Nous adoptons une vue avisée de la plate-forme Spring et des bibliothèques tierces afin que vous puissiez commencer avec un minimum d'agitation. La plupart des applications Spring Boot nécessitent très peu de configuration Spring.

Vous pouvez commencer avec des configurations minimales sans avoir besoin d'une configuration de configuration Spring complète. Le Spring Boot a un but bien déterminé et des avantages.

1. Buts

- ✚ Spring Boot est conçu avec les objectifs suivants
- ✚ Pour éviter une configuration XML complexe dans Spring
- ✚ Pour développer plus facilement des applications Spring prêtes pour la production

- ✚ Pour réduire le temps de développement et exécuter l'application indépendamment
- ✚ Offrez un moyen plus simple de démarrer avec l'application.

2. Les avantages

Spring Boot offre les avantages suivants à ses développeurs

- Facile à comprendre et à développer des applications de ressort
- Augmente la productivité.
- Réduit le temps de développement

Il évite d'écrire plusieurs codes d'expression standard, des Annotations et des configurations XML.

Il est très facile d'intégrer des applications Spring Boot avec ses écosystèmes de **Spring JDBC**, **Spring ORM**, **Spring Data**, **Spring Security** etc.

3. Pourquoi le choix de Spring Boot ?

Vous pouvez choisir Spring Boot en raison des fonctionnalités et des avantages qu'il offre, comme indiqué ici :

- ✚ Il fournit un moyen flexible de configurer les beans Java, les configurations XML et les transactions de base de données.
- ✚ Il fournit un traitement en lots puissant et gère les points de terminaison REST.
- ✚ Dans Spring Boot, tout est configuré automatiquement ; aucune configuration manuelle n'est nécessaire.
- ✚ Il offre une application de ressort basée sur des annotations
- ✚ Facilite la gestion des dépendances
- ✚ Il comprend un conteneur de servlet intégré

En utilisant les outils Spring Boot dans une application, il sera bordé de fichiers XML qui indiquent les configurations des servlets, des vues, des contenus statiques, etc. Ces fichiers de configuration deviennent un vrai challenge lorsque vous avez une application complexe. Pour

simplifier cette configuration, Spring Boot propose deux fonctionnalités principales qui sont l'auto-configuration et les starters.

L'auto-configuration

Cette fonctionnalité est la plus importante de Spring Boot. Elle permet de configurer automatiquement votre application en fonction des dépendances que vous avez ajoutées au projet à l'aide de l'annotation **@EnableAutoConfiguration**. Par exemple, si la base de données **MySQL** se trouve sur votre chemin de classe, mais que vous n'avez pas configuré aucune connexion à la base de données, Spring Boot configure automatiquement une base de données en mémoire.

Le point d'entrée de l'application Spring Boot est la classe contenant l'annotation **@SpringBootApplication** et la méthode principale.

Spring Boot, analyse automatiquement tous les composants inclus dans le projet à l'aide de l'annotation **@ComponentScan**.

En d'autres termes, si vous avez importé des dépendances, Spring Boot ira consulter cette liste puis produira la configuration nécessaire pour que tout fonctionne correctement.

Les Starters

Les starters viennent compléter l'auto-configuration et font gagner énormément de temps, notamment lorsqu'on commence le développement d'un **Microservice**. Un starter va apporter à votre projet un ensemble de dépendances, communément utilisées pour un type de projet donné. Ceci va vous permettre de créer un "squelette" prêt à l'emploi très rapidement. L'autre énorme avantage est la gestion des versions de la dépendance.

Spring Boot résout ce problème en fournissant un ensemble de dépendances pour la commodité des développeurs. Plus besoin de chercher quelles versions sont compatibles puis de les ajouter une à une dans le pom.xml ! Il vous suffit d'ajouter une simple dépendance au starter de votre choix. Cette dépendance va alors ajouter, à son tour, les éléments dont elle dépend, avec les bonnes versions.

III.1.2. Thymeleaf

Thymeleaf est une bibliothèque Java. Il s'agit d'un moteur de Template Java de modèle XML, XHTML, HTML5 capable d'appliquer un ensemble de transformations aux fichiers de modèle afin d'afficher les données et/ou le texte produits par vos applications.

Il est mieux adapté pour servir XHTML ou HTML5 dans des applications Web, mais il peut traiter n'importe quel fichier XML, que ce soit sur le Web ou dans des applications autonomes. L'objectif principal de Thymeleaf est de fournir une manière élégante et bien formée de créer des modèles. Pour ce faire, il est basé sur des balises et des attributs XML qui définissent l'exécution de la logique prédéfinie sur le DOM (Document Object Model), au lieu d'écrire explicitement cette logique sous forme de code dans le modèle.

III.2 Technologies utilisées pour la partie Front-End

Lorsque l'on parle de « Front-End », il s'agit finalement des éléments de l'application que l'on voit à l'écran et avec lesquels on peut interagir. Ces éléments sont composés de HTML, CSS et de Javascript contrôlés par le navigateur web de l'utilisateur pour l'implémentation de la partie **Front-End** de notre application nous avons utilisé des technologies comme *Bootstrap et JQuery*.

III.2.1. Bootstrap

Bootstrap est un framework frontal gratuit pour un développement web plus rapide et plus facile. Il comprend des modèles de conception HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les modèles, les styles d'images et bien d'autres, ainsi que des plugins JavaScript en option. Bootstrap vous donne également la possibilité de créer facilement des conceptions réactives.

La conception Web réactive consiste à créer des sites Web ou application qui s'ajustent automatiquement pour bien paraître sur tous les appareils, des petits téléphones aux grands ordinateurs de bureau.

III.2.2. jQuery

jQuery est une bibliothèque JavaScript rapide, petite et riche en fonctionnalités. Il rend les choses comme la traversée et la manipulation de documents HTML, permettant en quelques lignes de code de dynamiser un site web, de créer de petites animations, des interactions avec les formulaires, de programmer des appels Ajax. La gestion des événements, l'animation et Ajax beaucoup plus simples avec une API facile à utiliser qui fonctionne sur une multitude de navigateurs. Avec une combinaison de polyvalence et d'extensibilité, jQuery a changé la façon dont des millions de personnes écrivent JavaScript. C'est un ensemble organisé d'interactions, d'effets, de widgets et de thèmes d'interface utilisateur, construit au-dessus de la bibliothèque jQuery.

jQuery est conçu pour les concepteurs et les développeurs. Avec tous nos plugins pour permettre d'être opérationnel rapidement tout en étant suffisamment flexible pour évoluer avec les besoins et résoudre une surcharge de cas d'utilisation. L'approche modulaire de la bibliothèque jQuery permet la création de pages Web dynamiques et d'applications Web puissantes.

IV. SECURISATION DE L'APPLICATION

La sécurité est très importante pour la protection contre les éventuelles attaques. Notre application est développée avec la technologie Spring Boot. Les applications spring sont sécurisées par **Spring Security** qui fournit une authentification et un support d'autorisation. Pour renforcer la sécurité de notre application Spring Boot, nous pouvons utiliser la dépendance *spring-boot-starter-security* qui inclut *spring-security-config*. Avec *spring-boot-starter-security* **le mot de passe utilisateur sera encrypté** afin d'éviter de le stocker en clair dans la base de données.

```
i3  
i4 <dependency>  
i5     <groupId>org.springframework.boot</groupId>  
i6     <artifactId>spring-boot-starter-security</artifactId>  
i7 </dependency>  
i8
```

Figure 27: dependence spring-boot-starter-security

Pour personnaliser la configuration, nous allons créer un package *security*. Dans ce package, nous allons gérer l'ensemble de la sécurité de l'application. L'authentification, les contrôles d'accès, les redirections à partir des rôles qui sont gérées à ce niveau. Avec ce package une classe nommée **SecurityConfig** qui hérite de **WebSecurityConfigurerAdapter**. Cette classe doit avoir les annotations :





@EnableWebSecurity est une annotation qui permet d'activer toutes les fonctionnalités nécessaires à la configuration.

@Configuration est une annotation qui permet de configurer une classe comme une source de définition des beans Spring.

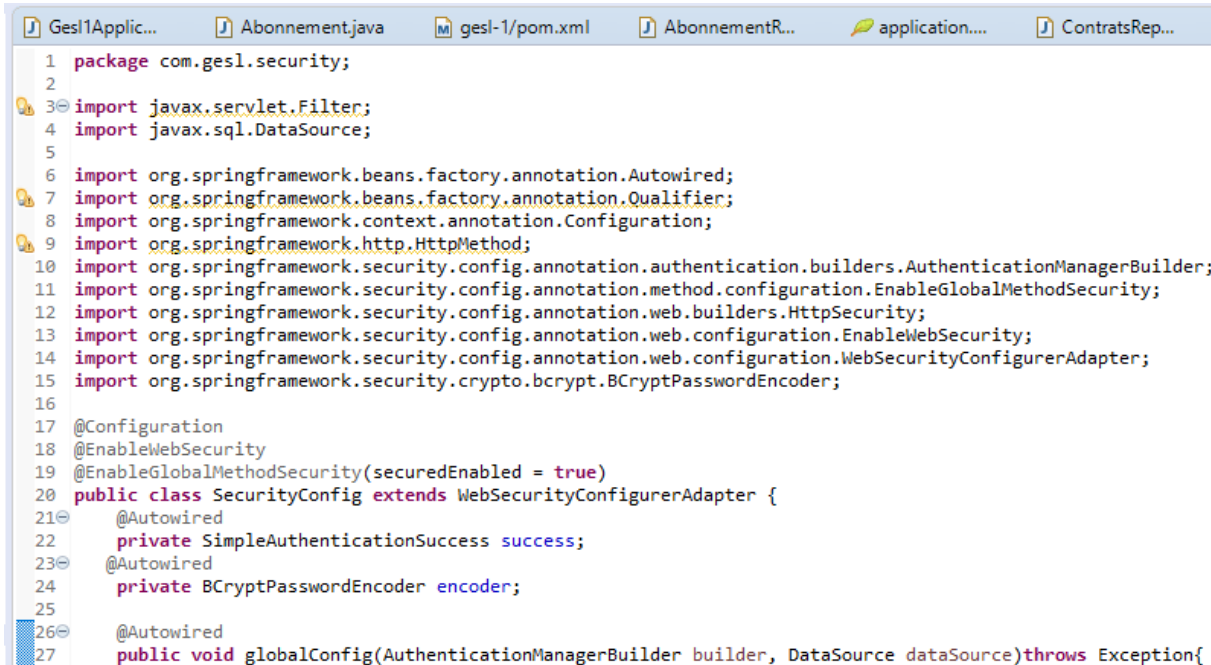
@EnableGlobalMethodSecurity est une annotation qui permet de gérer l'accès à des méthodes grâce au rôle de l'utilisateur avec l'argument « **SecuredEnable = True** ».

L'extension à la classe « **WebSecurityConfigurerAdapter** » fournit tous les services nécessaires.

Elle exige aussi une authentification à chaque URL de votre application, Générez un formulaire de connexion, Autoriser l'utilisateur avec l'**utilisateur Username** et le **mot de passe Password** à s'authentifier avec une authentification basée sur un formulaire, Autoriser l'utilisateur à se déconnecter, Prévention des attaques CSRF, Protection de fixation de session , Intégration de l'en-tête de sécurité comme : **HTTP Strict Transport Security** pour les requêtes sécurisées, Intégration **X-Content-Type-Options**, Contrôle du cache (peut être remplacé ultérieurement par votre application pour permettre la mise en cache de vos ressources statiques), Intégration de **X-XSS-Protection**, Intégration de **X-Frame-Options** pour empêcher le détournement de clics et Intégration avec les méthodes d'API Servlet suivantes :

-  HttpServletRequest # getRemoteUser ()
-  HttpServletRequest.html # getUserPrincipal ()
-  HttpServletRequest.html # isUserInRole (java.lang.String)
-  HttpServletRequest.html # login (java.lang.String, java.lang.String)

 HttpServletRequest.html # logout ()



```

1 package com.gesl.security;
2
3 import javax.servlet.Filter;
4 import javax.sql.DataSource;
5
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.beans.factory.annotation.Qualifier;
8 import org.springframework.context.annotation.Configuration;
9 import org.springframework.http.HttpMethod;
10 import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;
11 import org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;
12 import org.springframework.security.config.annotation.web.builders.HttpSecurity;
13 import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
14 import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
15 import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
16
17 @Configuration
18 @EnableWebSecurity
19 @EnableGlobalMethodSecurity(securedEnabled = true)
20 public class SecurityConfig extends WebSecurityConfigurerAdapter {
21     @Autowired
22     private SimpleAuthenticationSuccess success;
23     @Autowired
24     private BCryptPasswordEncoder encoder;
25
26     @Autowired
27     public void globalConfig(AuthenticationManagerBuilder builder, DataSource dataSource) throws Exception{

```

Figure 28: Classe SecurityConfig

V. PRESENTATION DE L'APPLICATION

Dans cette partie nous allons faire une présentation du résultat obtenu de notre application. Notre présentation se fera par des captures d'écran de quelques pages important de notre application. Comme tous les acteurs de notre application ont leur interface, nous allons présenter d'abord l'interface des locataires ensuite celle des bailleurs et des agences et enfin celle de l'administrateur.

C'est l'interface qui s'affiche lors du lancement de notre application c'est la page d'accueil de notre application.

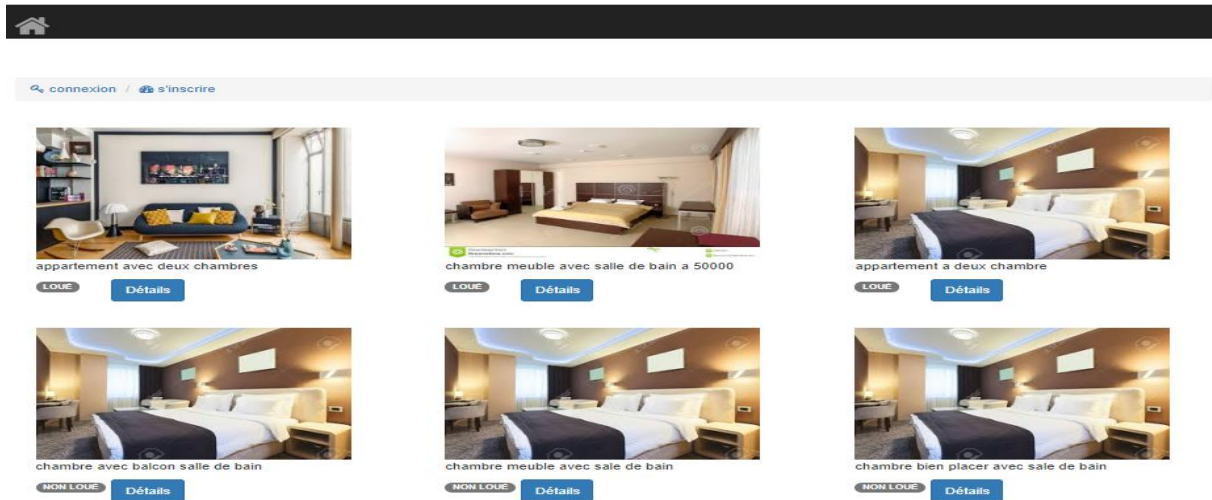


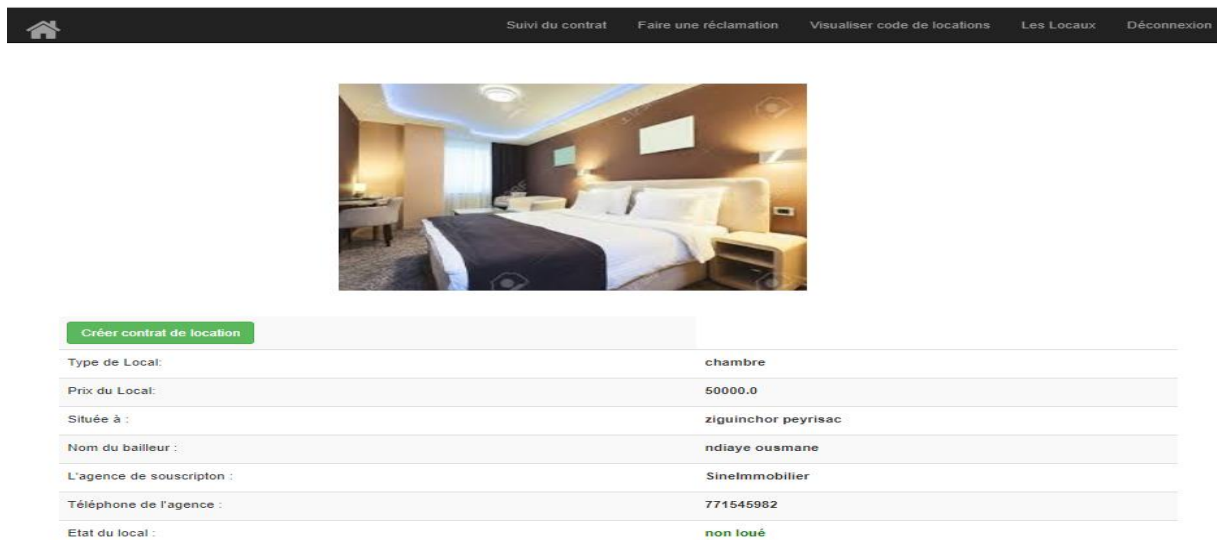
Figure 29: Page d'Accueil de l'Application

Dans cette page d'accueil, l'utilisateur peut créer son compte pour se connecter ou se connecter directement s'il a un compte déjà.

V.1 PRESENTATION DE L'INTERFACE DU COMPTE LOCATAIRE

Une fois que l'utilisateur saisit son login et mot de passe pour l'ouverture de son compte. Il visualise les locaux existants dans la plateforme, fait une réservation, suit le contrat pour savoir si la réservation est valide, fait une réclamation et télécharge son code occ.

- La figure ci- dessous nous montre le détail du local pour pouvoir visualiser la description et le prix et envoyer la demande de contrat.



Type de Local:	chambre
Prix du Local:	50000.0
Située à :	ziguinchor peyrisac
Nom du bailleur :	ndiaye ousmane
L'agence de souscription :	Sinemmobilier
Téléphone de l'agence :	771545982
Etat du local :	non loué

Figure 30: Détail du Local

1. Cette figure nous permet d'envoyer la création de contrat entre l'agence immobilière et le locataire.

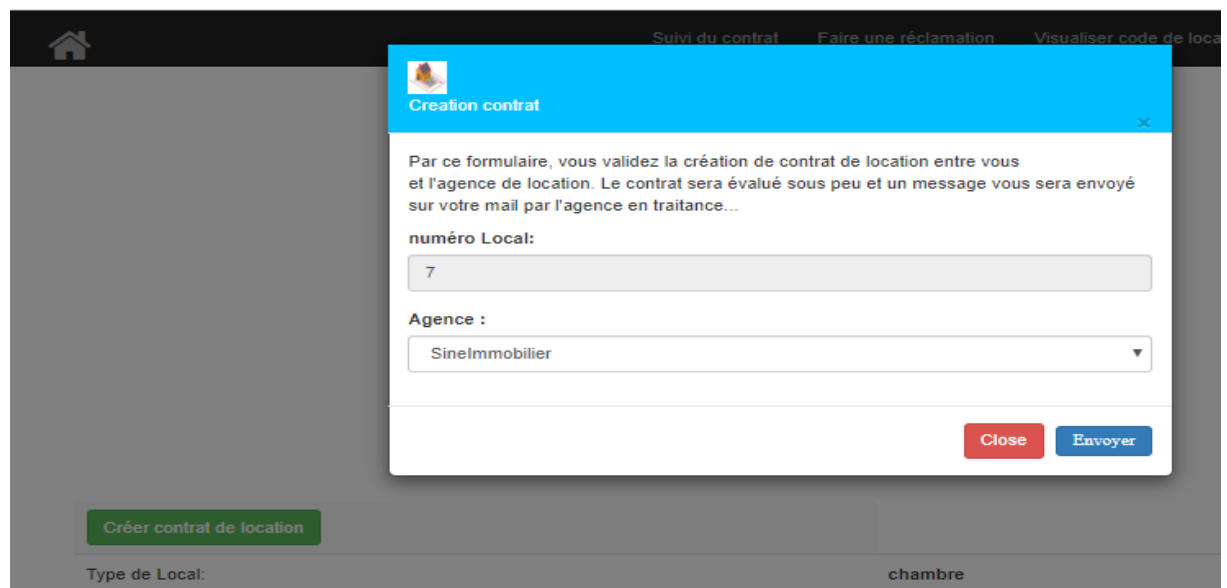


Figure 31: Demande de contrat

Cette figure nous montre le suivi du contrat. Elle permet de visualiser et imprimer le contrat après avoir envoyé une demande de validation de son réserve qui sera validée par son agence.

The screenshot shows a web interface for tracking contracts. At the top, there is a navigation bar with a home icon and links for 'Suivi du contrat', 'Faire une réclamation', 'Visualiser code de locations', and 'Déconnexion'. Below this is a green header for 'Les contrats suivis'. The interface includes a 'Show' dropdown set to '10', a search box, and a table of contract entries. The table has columns for 'Id', 'numero Local souscrit', 'Type', 'Agence', 'Date de creation', 'Etude de validation', 'locataire', and 'Option'. One entry is visible with 'Etude de validation' set to 'loué'. Below the table, there are pagination controls showing 'Showing 1 to 1 of 1 entries' and 'Previous 1 Next'.

Id	numero Local souscrit	Type	Agence	Date de creation	Etude de validation	locataire	Option
1	2	appartement	Billi kwint	2019-09-19 21:27:52.0	loué	SEYE daba	Imprimer Annuler

Figure 32: suivie contrat

Dans cette figure nous avons le Code OCC de location, Elle permet au locataire de télécharger le Code OCC défini par l'état pour savoir les lois du loyer au Sénégal.

The screenshot shows a web interface for viewing location rules. At the top, there is a navigation bar with a home icon and links for 'Suivi du contrat', 'Faire une réclamation', 'Visualiser code de locations', and 'Déconnexion'. Below this is a green header for 'Regle de location'. The interface includes a 'Show' dropdown set to '10', a search box, and a table of location rules. The table has columns for 'Nom fichier', 'Date d'entree', and 'lien telechargement'. Two entries are visible, each with a 'télécharger' button. Below the table, there are pagination controls showing 'Showing 1 to 2 of 2 entries' and 'Previous 1 Next'.

Nom fichier	Date d'entree	lien telechargement
article1_du_code_location.pdf	2019-08-14 00:00:00.0	télécharger
loyer.pdf	2020-01-15 16:58:22.0	télécharger

Figure 33: visualiser le code de locations

Cette figure est la partie de Réclamation son objectif est de permettre au locataire de déposer des messages sur le manquement, des suggestions ou des confrontations.

Id	Date du	Contenu	Locataire	Option
2	2020-01-07 14:34:15.0	mon chambre a des fissures	gal sam	Annuler
3	2020-02-20 15:13:10.0	je suis vois vous ne respecter pas le contrat	gal sam	Annuler

Figure 34: faire une réclamation

V.2 PRESENTATION DE L'INTERFACE DU COMPTE BAILLEUR

Dans cette partie, nous présentons le menu bailleur comme on peut le voir ci-dessous, les bailleurs peuvent voir si l'agence a validé son demande, ajouter des locaux, visualiser les agences, abonner et faire leur abonnement dans les agences et visualiser le code OCC.

Dans cette partie, nous avons la figure qui nous montre les agences qui existent dans la plateforme et elle permet au bailleur de s'abonner à l'un d'eux.

Nom Agence	Adresse	Ville	adresse mail	Telephone	Responsable	Option
Billi kwin	nema cadior	ziguinchor	billi@gmail.com	332854525	siguiri	s'abonner
imo	Point E		taf@imo.gs	33991206	sagna	s'abonner
Sineimmobilier	castor	Dakar	sine@gmail.com	332850225	gaye	s'abonner

Figure 35: Les agences existantes pour le Bailleur

Dans cette figure, c'est la partie ou le bailleur qui envoie sa demande de souscription dans une agence de la place.

Figure 36: Demande d'abonnement envoyé

Cette figure ci-dessous nous montre les agences abonnées, elle permet aux bailleurs de consulter si sa demande d'abonnement est validée par l'agence.

Nom Agence	nom bailleur	Date début	Date fin	Etude de validation	Option
Billi kvin	Sane	2020-03-04 00:00:00.0	2020-12-31 00:00:00.0	en cours	Annuler
gallembobilier	Sane	2020-03-10 00:00:00.0	2020-06-20 00:00:00.0	validé	Annuler
Inter Action Immobilière	Sane	2020-03-11 00:00:00.0	2020-12-31 00:00:00.0	validé	Annuler
Sinembobilier	Sane	2020-03-04 00:00:00.0	2020-12-31 00:00:00.0	en cours	Annuler

Figure 37: Etat des abonnements des agences

Cette figure est la partie des locaux du bailleur, elle permet aux bailleurs d'ajouter des locaux dans les agences.

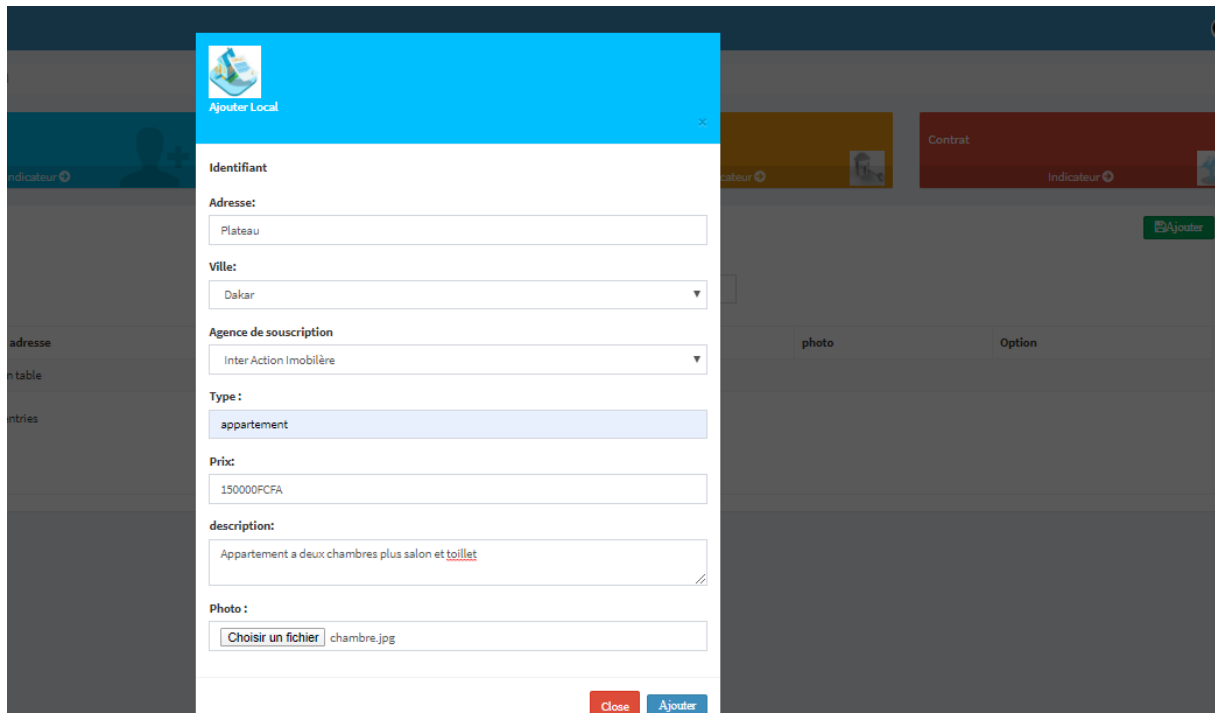


Figure 38: ajouter local

V.3 PRESENTATION DE L'INTERFACE DU COMPTE AGENCE

Comme nous l'avons annoncé dans la présentation de l'application, voici dans le menu agence qui permet au responsable de l'agence de valider la demande des bailleurs, visualiser les locaux des bailleurs et valider la demande des locataires

Cette figure ci-dessous nous montre les locaux, elle permet aux responsables agences de gérer les locaux des bailleurs qui sont dans son agence.

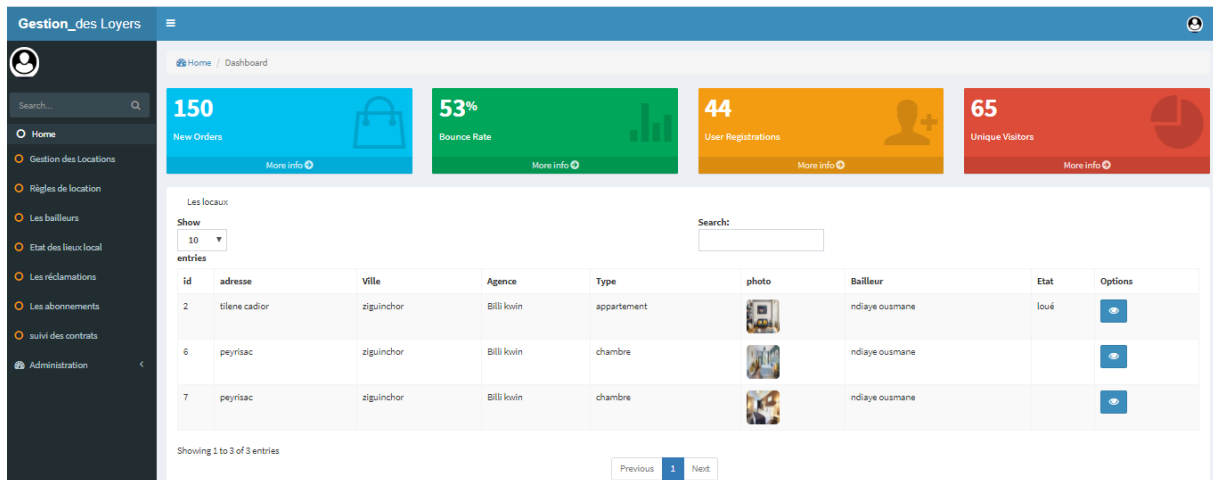


Figure 39: les locaux ajouter à l'agence

Dans cette figure nous avons la partie des locataires, elle permet de valider et envoyer le contrat.

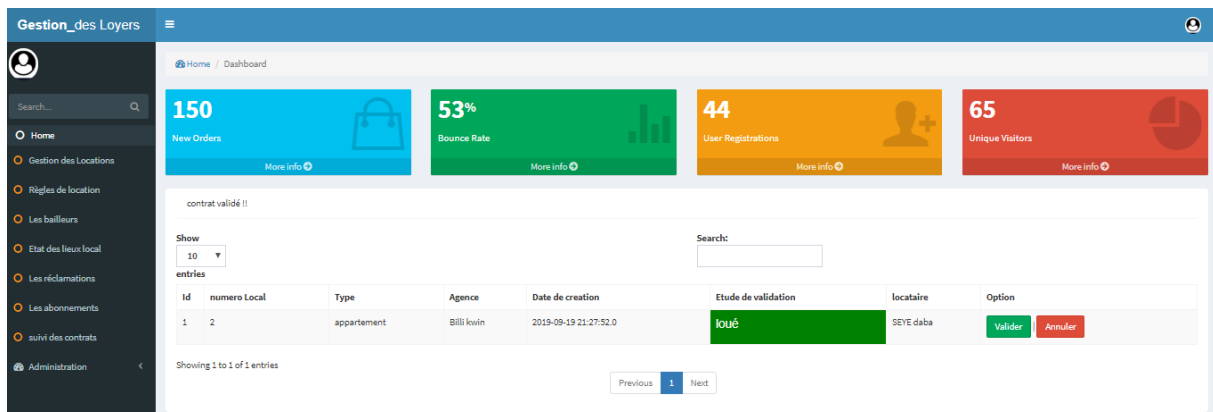


Figure 40: validation du contrat

Cette figure c'est la partie des abonnements, elle permet aux agences de valider la demande des bailleurs dans son agence.

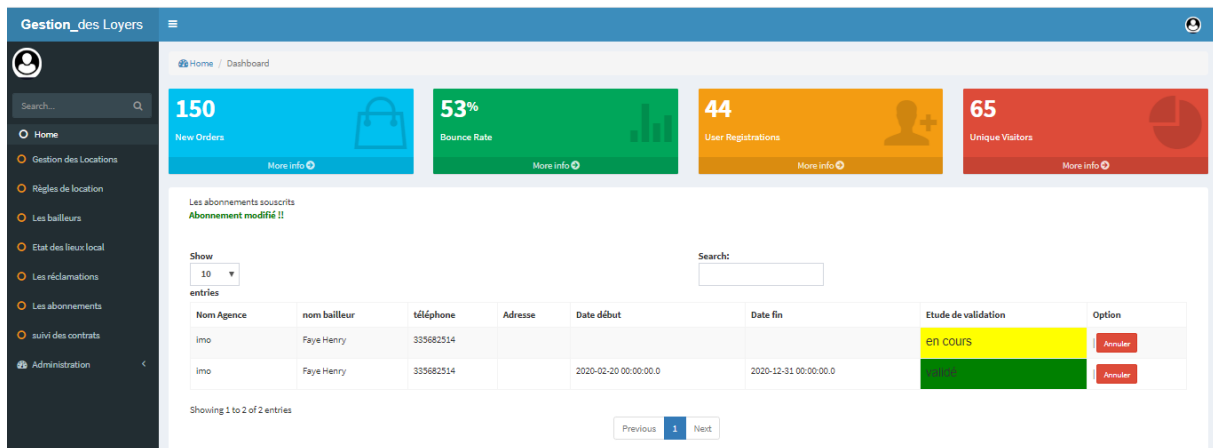


Figure 41: validation de l'abonnement

V.4 PRESENTATION DE L'INTERFACE DU COMPTE ADMINISTRATEUR

Dans cette partie vous avons l'interface de l'administrateur qui permet de visualiser les agences, les responsable et gérer le code OCC.

Cette figure nous présente les agences, elle permet à l'administrateur de visualiser les agences et leurs responsables.

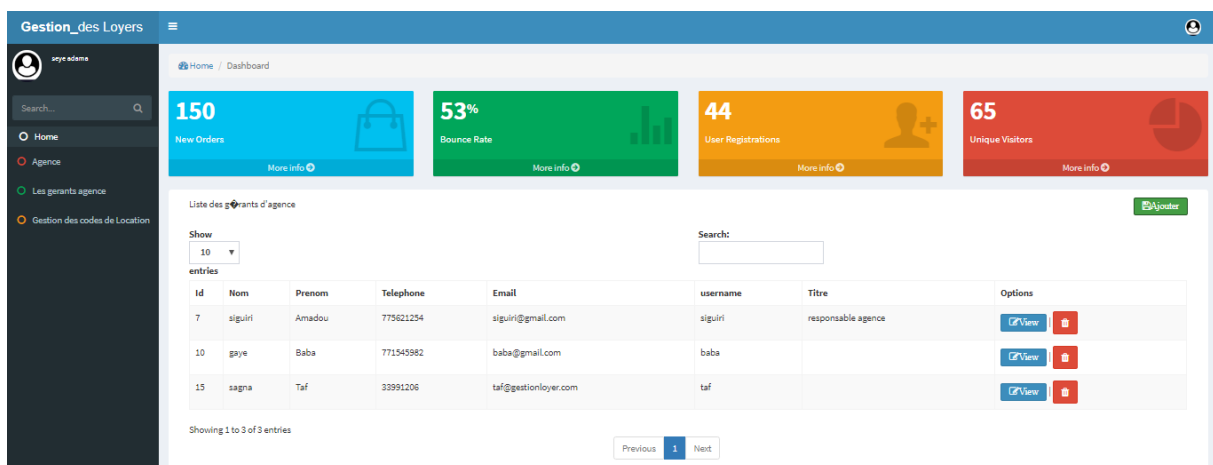


Figure 42: liste des agences

CONCLUSION

Comme toute innovation technologique qui se respecte, l'application pour la gestion et le suivi des loyers simplifie non seulement le travail des agences et bailleurs, rendre les locataires une gestion de leur temps sur la recherche des locaux, mais permet aussi à l'état du Sénégal d'avoir une visibilité sur ce que l'on fait dans le loyer. L'étude et mise en place d'une solution pour la gestion et le suivi des loyers a pour objectif la baisse des prix des loyers soit plus efficace à gérer, et que les agences déclarent tous les locaux qu'ils ont dans leur agence pour pouvoir payer leurs fiscalités normalement et régler les conflits des locataires.

Au cours de la phase de réalisation de notre application, nous avons divisé ce travail en trois principales parties. La première partie a fait l'objet de la présentation Générale, la deuxième partie est axée sur l'analyse et la conception de l'application et la troisième partie a parlé sur l'implémentation et présentation.

Ce mémoire nous a permis d'avoir une mise en pratique des outils et fonctionnalités des technologies Spring Boot et Thymeleaf et concrètement de pouvoir l'appliquer à toutes les applications.

Il a été enrichissant, aussi bien au niveau recherche que professionnel et sera un atout pour notre insertion dans la vie active. Il nous a apporté de nouvelles connaissances aussi bien sur le plan méthodologique, organisationnelle que technique et nous a permis d'approfondir les compétences que nous avons acquises tout au long de notre cursus universitaire et basculer dans le monde des développeurs.

Dans le futur, il serait intéressant de mettre en perspective :

- d'apporter une touche de sécurité concernant l'accès à ce système en production ;
- l'étendre en mettant en place une solution adaptée aux smartphones ;
- Implémenter une API pour le paiement en ligne
- Implémenter une géolocalisation.

REFERENCE

1. https://www.google.com/search?rlz=1C1CHBF_frFR862FR862&tbm=isch&sxsrf=A_CYBGNRmHna9PGE0ZdvjZRbl3Kcx4PS9zQ%3A1580175285917&sa=1&ei=tY8vXp3RN6rjsAf-jY
2. https://www.tutorialspoint.com/spring_boot/
3. <https://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html>
4. <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2111926-le-diagramme-d-activite>
5. https://www.pressafrik.com/Crise-du-loyer-au-Senegal-Les-faiblesses-d-une-loi-aux-relents-populistes-Acte-II_a118684.html
6. <https://w3tutoriels.com/bootstrap/qu-est-ce-que-bootstrap/#:~:text=Bootstrap%20est%20le%20framework%20HTML,plus%20facile%20et%20plus%20rapide.>
7. <https://adminlte.io/>
8. http://www.lycee-ferry-versailles.fr/si-new/1_1_analyse_fonctionnelle_structurale/1_cours_analyse_fonctionnelle.pdf
9. <https://www.peoi.org/Courses/Coursesfr/mic/mic10.html>
10. <http://www.servicepublic.gouv.sn/e-citoyen/vos-idees/consommation-commerce/article/harmoniser-le-prix-des-loyers-et-de-l-acquisition-de-biens-immobiliers-a-dakar>
11. https://fr.wikipedia.org/wiki/Mod%C3%A8le_vue_contr%C3%B4leur
12. https://www.sparxsystems.fr/resources/uml2_tutorial/uml2_packagediagram.html
13. <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/outils-uml/>
14. <https://spring.io/tools>
15. <https://docs.spring.io/spring-security/site/docs/current/guides/html5/helloworld-javaconfig.html>

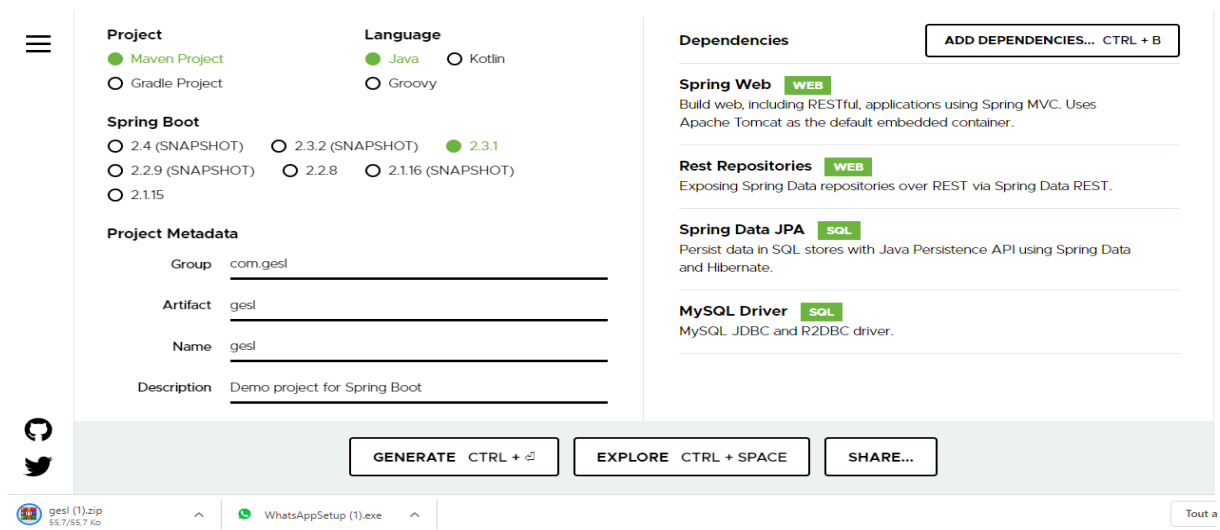
16. https://youtu.be/il_gXsZngUs

17. https://youtu.be/_1welXC8gEo

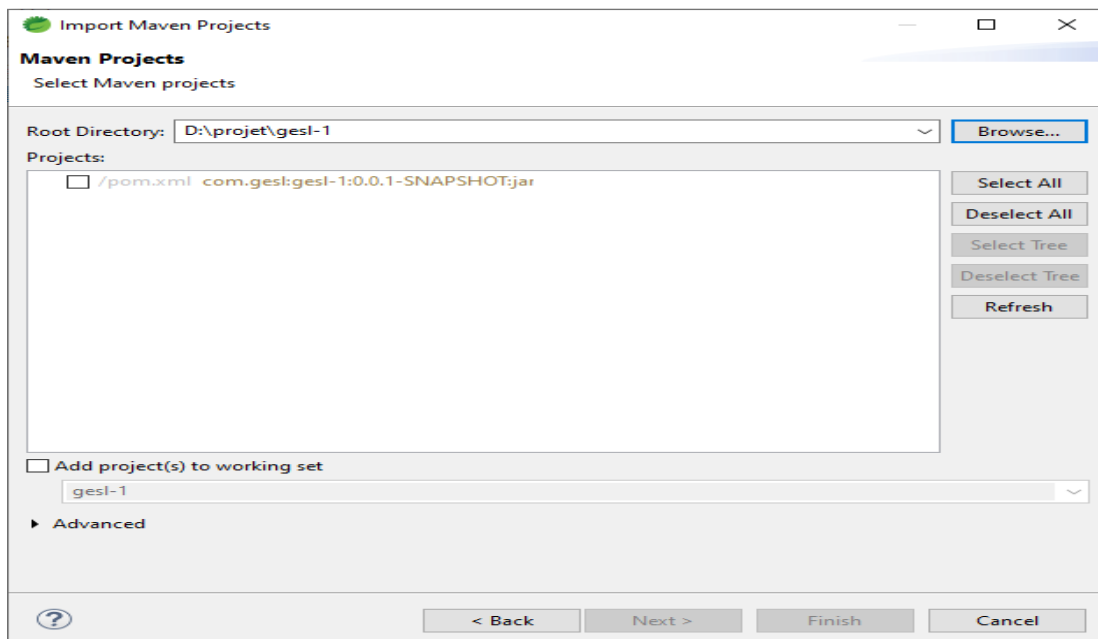
ANNEXE

Annexe 1 : Création du projet

Création de l'application dans spring initializr en ligne

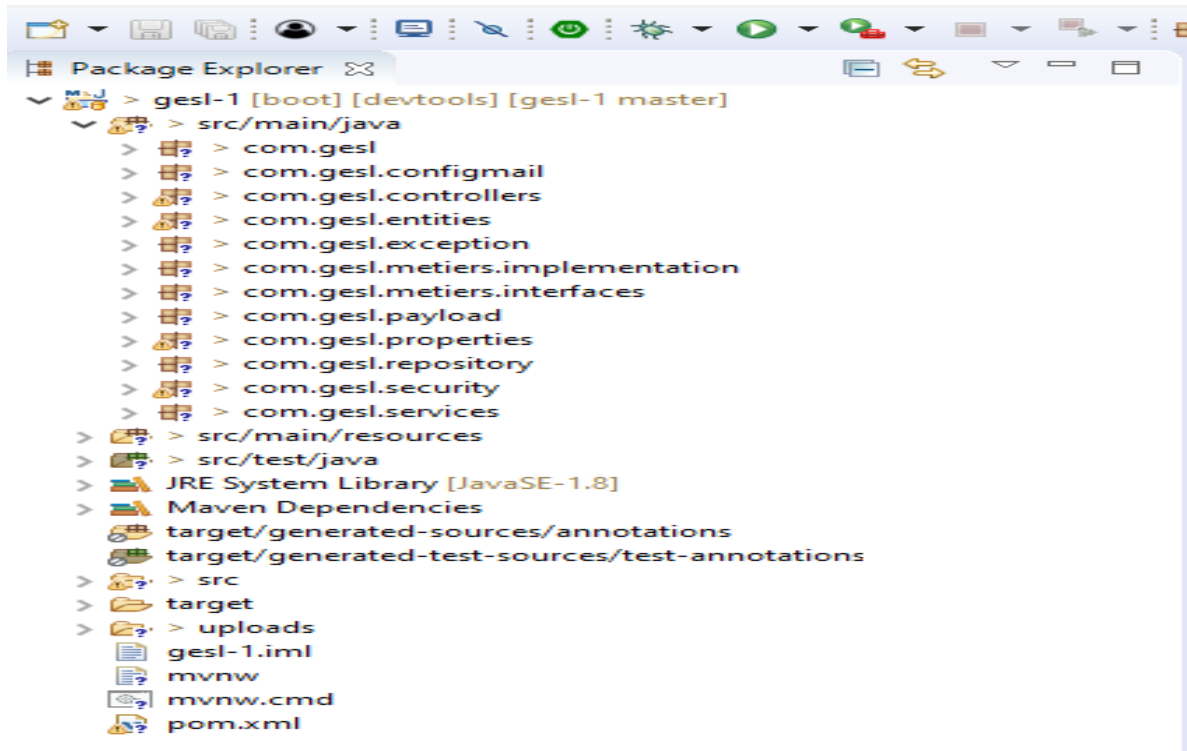


Après avoir créé le projet puis le télécharger. On importer le code directement dans notre IDE Spring Tool Suite



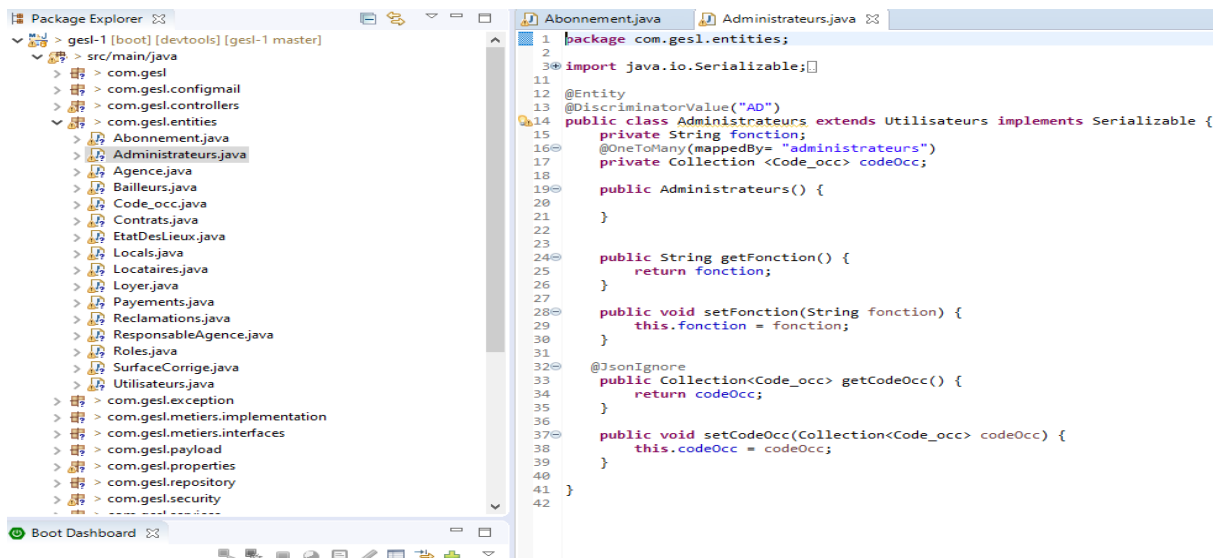
Annexe 2 : les Packages

L'arborescence des packages de l'application



Annexe 3 : les entités

Développement de l'entité Administrateur (table Administrateur)



Développement de l'entité abonnement (table Abonnement)

```

1 package com.gesl.entities;
2
3 import java.io.Serializable;
4
5 @Entity
6 @JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
7 public class Abonnement implements Serializable {
8     @Id
9     @GeneratedValue
10    private Long id;
11    @DateTimeFormat(pattern = "yyyy-MM-dd")
12    private Date dateDebut;
13    @DateTimeFormat(pattern = "yyyy-MM-dd")
14    private Date dateFin;
15    private String situation;
16    private boolean etat;
17
18    public boolean isEtat() {
19        return etat;
20    }
21
22    public void setEtat(boolean etat) {
23        this.etat = etat;
24    }
25
26    @ManyToOne
27    @JoinColumn(name = "idbail")
28    private Bailleurs bailleurs;
29
30    @ManyToOne
31    @JoinColumn(name = "idAg")
32    private Agence agence;
33
34    public Abonnement() {
35    }
36
37    public String getSituation() {
38        return situation;
39    }
40
41
42
43
44
45
46
47
48
49
50

```

Développement de l'entité Agence (table Agence)

```

1 package com.gesl.entities;
2
3 import java.io.Serializable;
4
5 @Entity
6 @JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
7 public class Agence implements Serializable {
8     @Id
9     @GeneratedValue(strategy=GenerationType.IDENTITY)
10    private Long id;
11    private String nom;
12    private String adresse;
13    private String email;
14    private String tel;
15    private String ville;
16    public String getVille() {
17        return ville;
18    }
19
20    public void setVille(String ville) {
21        this.ville = ville;
22    }
23
24    @JsonIgnore
25    public Collection<Locals> getLocals() {
26        return locals;
27    }
28
29    public void setLocals(Collection<Locals> locals) {
30        this.locals = locals;
31    }
32
33    @OneToOne
34    @JoinColumn(name="id_Resp")
35    private ResponsableAgence responsableAgence;
36    @OneToMany(mappedBy = "agence")
37    private Collection <Abonnement> abonnement = new ArrayList<>();
38    @OneToMany(mappedBy = "agence")
39    private Collection <EtatDesLieux> etatDesLieux;
40    @OneToMany(mappedBy = "agence")
41
42
43
44
45
46
47
48
49
50
51
52

```

Développement de l'entité Bailleur (table Bailleur)

```

1 package com.gesl.entities;
2
3 import java.io.Serializable;
12
13 @Entity
14 @DiscriminatorValue("Ba")
15 public class Bailleurs extends Utilisateurs implements Serializable {
16
17     private String fonction;
18     private String adresse;
19     @OneToMany(mappedBy = "bailleurs")
20     private Collection<Locals> locals = new ArrayList<>();
21     @OneToMany(mappedBy = "bailleurs")
22     private Collection<Abonnement> abonnement = new ArrayList<>();
23
24     public Bailleurs() {
25     }
26
27
28     public String getFonction() {
29         return fonction;
30     }
31
32     public void setFonction(String fonction) {
33         this.fonction = fonction;
34     }
35
36     public String getAdresse() {
37         return adresse;
38     }
39
40     public void setAdresse(String adresse) {
41         this.adresse = adresse;
42     }
43
44     @JsonIgnore
45     public Collection<Locals> getLocals() {
46         return locals;
47     }

```

Développement de l'entité Utilisateur (table Utilisateur)

```

1 package com.gesl.entities;
2
3 import java.io.Serializable;
17
18
19 @Entity
20 @Inheritance(strategy = InheritanceType.SINGLE_TABLE)
21 @DiscriminatorColumn(name="type_utilisateurs",discriminatorType = DiscriminatorType.STRING,length = 3)
22 public class Utilisateurs implements Serializable {
23
24     @Id
25     @GeneratedValue(strategy=GenerationType.IDENTITY)
26     private Long idUtili;
27     private String nom;
28     private String prenom;
29     private String login;
30     private String password;
31     private String tel;
32     private String email;
33     private String username;
34     private String typeuser;
35
36
37     public String getUsername() {
38         return username;
39     }
40
41     public void setUsername(String username) {
42         this.username = username;
43     }
44
45     @ManyToMany(fetch=FetchType.EAGER)
46     private Collection<Roles> roles=new ArrayList<>();
47
48     public Utilisateurs() {
49     }
50
51 }

```

Développement de l'entité Rôle (table Roles)

```

1 package com.gesl.entities;
2
3 import java.io.Serializable;
4
5 @Entity
6 public class Roles implements Serializable{
7     @Id
8     @GeneratedValue
9     private Long id;
10    private String nom;
11
12    public Roles() {
13        super();
14    }
15
16    public Long getId() {
17        return id;
18    }
19
20    public void setId(Long id) {
21        this.id = id;
22    }
23
24    public String getNom() {
25        return nom;
26    }
27
28    public void setNom(String nom) {
29        this.nom = nom;
30    }
31
32 }

```

Développement de l'entité Payements (table Payements)

```

1 package com.gesl.entities;
2
3 import java.io.Serializable;
4
5 @Entity
6 @JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
7 public class Payements implements Serializable{
8     @Id
9     @GeneratedValue(strategy=GenerationType.IDENTITY)
10    private Long id;
11    private Date datePay;
12    private String quittance;
13    private double montantVerser;
14    private double restant;
15    private String echeancemois;
16    private boolean regle;
17    private String numeroFacture;
18
19    public String getNumeroFacture() {
20        return numeroFacture;
21    }
22
23    public void setNumeroFacture(String numeroFacture) {
24        this.numeroFacture = numeroFacture;
25    }
26
27    public boolean isRegle() {
28        return regle;
29    }
30
31    public void setRegle(boolean regle) {
32        this.regle = regle;
33    }
34
35    public String getEcheancemois() {
36        return echeancemois;
37    }
38
39 }

```