

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire
Abdelhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

EN: Informatique

Spécialité : Spécialité: Sciences et Technologies de l'Information et de la
Communication (STIC)

Développement d'une application web Pour la gestion locative au niveau de l'OPGI de Mila

Préparé par :

Safa BENCHAOUI
Sarrah BOULKELIA

Devant le jury

Président	Adel BOURIDAH	M.A.A	Centre Uniersitaire de Mila
Examineur	Souheila KHALFI	M.A.B	Centre Uniersitaire de Mila
Promoteur	Sadek BENHAMMADA	M.C.B	Centre Uniersitaire de Mila

Année Universitaire : 2017/2018

Remerciement

La première personne que nous tenons à remercier est notre encadrant Mr. Sadek BENHAMMADA, pour l'orientation, la disponibilité et la patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené au bon port. Qu'il trouve dans ce travail un hommage vivant à sa haute personnalité.

Nous exprimons notre gratitude aux membres du jury, qui nous ont honorés en acceptant de juger ce modeste travail.

Nos remerciements s'étendent également à tout le personnel de l'office de promotion et de gestion immobilière de la wilaya de Mila pour leur orientation, leurs explications et pour le temps qu'ils nous ont consacré pour répondre à tous nos questions tout au long de cet expérience.

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné surtout Mr. BENCHEIKH ELHOCINE Madjed qui par ses compétences, sa confiance et ses encouragements nous a soutenu dans la poursuite de nos études.

Nous adressons une pensée spéciale à nos parents pour leurs soutiens et leurs attentions sans faille.

Enfin, on remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail et on exprime notre gratitude envers tous ceux qui nous ont accordé leurs soutien, tant par leurs gentillesse que par leur dévouement.

Merci à tous et à toutes.

Résumé

Dans le cadre de notre Projet de Fin d'Etudes, l'OPGI de Mila souhaite automatiser la gestion locative, qui aura besoin de gérer l'ensemble de ses biens immobiliers, locataires, paiements, etc. à travers une interface simple et pratique.

Pour bien mener le développement de ce projet, la méthode générique proposée par Roques (2002) qui est fondée sur l'utilisation d'un sous ensemble nécessaire et suffisant du langage de modélisation UML semble plus adéquate à notre contexte.

Pour ce qui concerne le volet technique, nous avons utilisé le langage de programmation PHP ainsi qu'une multitude de technologie web autour à s'avoir : HTML, CSS, JavaScript. Le SGBD MYSQL été utilisé pour l'implémentation de notre base de données.

Notre mission consiste à développer une application web qui offrira des fonctionnalités permettant aux utilisateurs de réaliser différentes tâches. En utilisant les différentes techniques et outils de développement mentionné auparavant.

Mots clés

Gestion locative, UML, Applications web, VPN, HTML, CSS, PHP, SGBD.

Table des matières

Introduction générale.....	1
1. Contexte	1
2. Motivation	1
3. Objectif du mémoire.....	2
4. Plan du mémoire.....	2
CHAPITRE 1 Étude de l'existant : Gestion locative au sein de l'OPGI de Mila.....	4
1.1 Introduction	4
1.2 Présentation de l'organisme (OPGI de Mila)	4
1.2.1 Statut juridique.....	4
1.2.2 Objet.....	5
1.2.3 Missions de l'office.....	5
1.2.4 Structure interne	5
1.3 Présentation de l'existant.....	8
1.3.1 Gestion locative	8
1.3.2 Gestion locative au niveau de l'OPGI de Mila	8
1.3.3 Informatisation de la gestion locative au sein de l'OPGI de Mila	9
1.4 Critique de l'existant	12
1.5 Conclusion	13
CHAPITRE 2 Proposition d'une nouvelle solution.....	14
2.1 Introduction	14
2.2 Les applications Web.....	14
2.2.1 Notion d'une application web	15
2.2.2 Architectures des applications web.....	16
2.2.3 Types d'architectures des applications web.....	16
2.3 Généralité sur les VPN	18
2.3.1 C'est quoi un VPN	18
2.3.2 Principe de fonctionnement d'un VPN	19
2.3.3 Les différents types de VPN	19
2.3.4 Protocoles utilisés pour réaliser une connexion VPN.....	21

2.3.5	Sécurité du VPN	22
2.4	Solution proposé	22
2.4.1	Présentation de la solution	22
2.4.2	Avantage de la solution.....	24
2.5	Conclusion	24
CHAPITRE 3 Identification des besoins		25
3.1	Introduction	25
3.2	Langage et processus pour l’analyse et la conception	25
3.2.1	Langage UML	26
3.2.2	Processus	26
3.3	Phase d’identification des besoins	28
3.3.1	Cahier des charges	28
3.3.2	Identification des acteurs	30
3.3.3	Diagramme de cas d’utilisation	31
3.3.4	Classement des cas d’utilisation	32
3.3.5	Description textuelles des cas d’utilisation	33
3.3.6	Diagramme de séquence système	44
3.4	Conclusion	56
CHAPITRE 4 Phase d’analyse		57
4.1	Introduction	57
4.2	Modèle du domaine	57
4.2.1	Modèle du domaine.....	59
4.3	Diagrammes de classes participantes	60
4.3.1	S’authentifier.....	61
4.3.2	Créer un compte	61
4.3.3	Consulter les comptes	62
4.3.4	Modifier un compte.....	62
4.3.5	Supprimer un compte	63
4.3.6	Ajouter un bien immobilier.....	63
4.3.7	Consulter les biens immobiliers.....	64
4.3.8	Rechercher les biens immobiliers	64
4.3.9	Modifier un bien immobilier.....	65
4.3.10	Supprimer un bien immobilier	65

4.3.11	Ajouter un locataire	66
4.3.12	Établir un contrat de location	66
4.3.13	Consulter les locataires	67
4.3.14	Rechercher les locataires	67
4.3.15	Modifier un locataire	68
4.3.16	Supprimer un locataire	68
4.3.17	Générer les loyers.....	69
4.3.18	Consulter les Loyers.....	69
4.3.19	Recevoir un paiement.....	70
4.3.20	Établir une quittance	70
4.3.21	Établir un avis d'échéance	71
4.3.22	Établir une mise en demeure	71
4.3.23	Consulter le tableau de bord.....	72
4.4	Diagramme d'activité de navigation.....	72
4.4.1	S'authentifier.....	72
4.4.2	Créer un compte	73
4.4.3	Consulter les comptes	73
4.4.4	Modifier un compte.....	74
4.4.5	Supprimer un compte	74
4.4.6	Ajouter un bien immobilier.....	75
4.4.7	Consulter les biens immobiliers.....	75
4.4.8	Rechercher les biens immobiliers	76
4.4.9	Modifier un bien immobilier.....	76
4.4.10	Supprimer un bien immobilier	77
4.4.11	Ajouter un locataire	77
4.4.12	Établir un contrat de location	78
4.4.13	Consulter les locataires	78
4.4.14	Rechercher des locataires	79
4.4.15	Modifier un locataire	79
4.4.16	Supprimer un locataire	80
4.4.17	Générer les loyers.....	80
4.4.18	Consulter les loyers	81
4.4.19	Recevoir un paiement.....	82
4.4.20	Établir une quittance	83

4.4.21	Établir un avis d'échéance	84
4.4.22	Établir une mise en demeure	85
4.4.23	Consulter le tableau de bord.....	85
4.5	Conclusion	86
CHAPITRE 5 Phase de conception		87
5.1	Introduction	87
5.2	Diagrammes de séquence	87
5.2.1	S'authentifier.....	88
5.2.2	Créer un compte	88
5.2.3	Consulter les comptes	89
5.2.4	Modifier un compte.....	89
5.2.5	Supprimer un compte	90
5.2.6	Ajouter un bien immobilier.....	90
5.2.7	Consulter les biens immobiliers.....	91
5.2.8	Rechercher les biens immobiliers	91
5.2.9	Modifier un bien immobilier.....	92
5.2.10	Supprimer un bien immobiliers.....	92
5.2.11	Ajouter un locataire.....	93
5.2.12	Établir un contrat de location	93
5.2.13	Consulter les locataires	94
5.2.14	Rechercher les locataires.....	94
5.2.15	Modifier un locataire	95
5.2.16	Supprimer un locataire	95
5.2.17	Générer les loyers.....	96
5.2.18	Consulter les loyers	97
5.2.19	Recevoir un paiement.....	98
5.2.20	Établir une quittance	98
5.2.21	Établir un avis d'échéance	99
5.2.22	Établir une mise en demeure	99
5.2.23	Consulter le tableau de bord.....	100
5.3	Diagramme de classes de conceptions.....	100
5.3.1	S'authentifier.....	101
5.3.2	Créer un compte	101

5.3.3	Consulter les comptes	102
5.3.4	Modifier un compte.....	102
5.3.5	Supprimer un compte	103
5.3.6	Ajouter un bien immobilier.....	103
5.3.7	Consulter les biens immobiliers.....	104
5.3.8	Rechercher les biens immobiliers	104
5.3.9	Modifier un bien immobilier.....	105
5.3.10	Supprimer un bien immobilier	105
5.3.11	Ajouter un locataire.....	106
5.3.12	Établir un contrat de location	106
5.3.13	Consulter les locataires	107
5.3.14	Rechercher des locataires	107
5.3.15	Modifier un locataire.....	108
5.3.16	Supprimer un locataire	108
5.3.17	Générer les loyers.....	109
5.3.18	Consulter les loyers	109
5.3.19	Recevoir un paiement.....	110
5.3.20	Établir une quittance	110
5.3.21	Établir un avis d'échéance	111
5.3.22	Établir une mise en demeure	111
5.3.23	Consulter le tableau de bord.....	112
5.4	Conclusion	112
CHAPITRE 6 Implémentation.....		113
6.1	Introduction	113
6.2	Passage du diagramme de classes au modèle relationnel	113
6.2.1	Les règles de passage	113
6.3	Tables de la base de données	114
6.4	Système de gestion de la base de donnée MySQL	115
6.5	Langage Web utilisés.....	116
6.5.1	HTML	116
6.5.2	CSS	117
6.5.3	JAVASCRIPT.....	117
6.5.4	PHP	118

6.6	Outils de développements.....	119
6.6.1	Pacestar UML Diagrammer	119
6.6.2	Editeur de texte Atom	120
6.6.3	Navigateur Web (Google Chrome).....	120
6.6.4	Serveur web Wamp.....	120
6.6.5	Bootstrap	120
6.7	Sécurité de l'application	121
6.7.1	Comment éviter ces attaques d'injection en PHP	121
6.7.2	Cryptage du mot de passe (Password-hash).....	121
6.8	Présentation de l'application	122
6.8.1	Page Authentification.....	122
6.8.2	Page Accueil	123
6.8.3	Page de création d'un compte	124
6.8.4	Page de consultations des bâtiments	125
6.8.5	Page de modification d'un bien immobilier.....	126
6.8.6	Page de suppression d'un locataire	126
6.8.7	Page d'établissement de contrat de location	127
6.8.8	Page de consultation des loyers	127
6.8.9	Page de réception d'un paiement	128
6.8.10	Consulter le tableau de bord.....	129
6.9	Conclusion	129
	Conclusion générale	130
	Bibliographie	132

Liste des tableaux

Tableau 3.1: Classement des cas d'utilisation.....	33
Tableau 4.1: Dictionnaire de données.	59

Liste des figures

Figure 1.1: Organigramme de l'OPGI de Mila	7
Figure 1.2: La gestion locative au niveau de l'OPGI de Mila.....	9
Figure 1.3: Interface de l'application Constatations	10
Figure 1.4: Interface de l'application Recouvrement (Au niveau des caisses)	11
Figure 1.5: Interface de l'application Recouvrement (Au niveau du DFC).....	11
Figure 2.1: Composants d'architecture d'une application Web	15
Figure 2.2: Fonctionnement d'un VPN	19
Figure 2.3: VPN d'accès itinérant	20
Figure 2.4: VPN intranet	20
Figure 2.5: VPN extranet	21
Figure 2.6: VPN intranet au niveau de l'OPGI Mila.....	23
Figure 3.1: Chaîne complète de la démarche de modélisation du besoin jusqu'au code	27
Figure 3.2: Diagramme des cas d'utilisation de l'acteur : «Web master»	31
Figure 3.3: Diagramme des cas d'utilisation de l'acteur «Caissier et Cadre dirigeant»	31
Figure 3.4: Diagramme des cas d'utilisation de l'acteur «Agent de maîtrise»	32
Figure 3.5: Diagramme de séquence système du cas d'utilisation «s'authentifier»	44
Figure 3.6: Diagramme de séquence système du cas d'utilisation «créer un compte»	45
Figure 3.7: Diagramme de séquence système du cas d'utilisation «consulter les comptes» ...	45
Figure 3.8: Diagramme de séquence système du cas d'utilisation «modifier un compte»	46
Figure 3.9: Diagramme de séquence système du cas d'utilisation «supprimerun compte».....	46
Figure 3.10: Diagramme de séquence système du cas d'utilisation «ajouter un bien immobilier»	47
Figure 3.11: Diagramme de séquence système du cas d'utilisation «consulter les biens immobiliers».....	47
Figure 3.12: Diagramme de séquence système du cas d'utilisation «rechercher les biens immobiliers».....	48
Figure 3.13: Diagramme de séquence système du cas d'utilisation «modifier un bien immobilier»	48
Figure 3.14: Diagramme de séquence système du cas d'utilisation «supprimer un bien immobilier»	49
Figure 3.15: Diagramme de séquence système du cas d'utilisation «ajouter un locataire».....	49

Figure 3.16: Diagramme de séquence système du cas d'utilisation «établir un contrat de location»	50
Figure 3.17: Diagramme de séquence système du cas d'utilisation «consulter les locataires»	50
Figure 3.18: Diagramme de séquence système du cas d'utilisation «rechercher les locataires»	51
Figure 3.19: Diagramme de séquence système du cas d'utilisation «modifier un locataire» ..	51
Figure 3.20: Diagramme de séquence système du cas d'utilisation «supprimer un locataire»	52
Figure 3.21: Diagramme de séquence système du cas d'utilisation «générer les loyers»	52
Figure 3.22: Diagramme de séquence système du cas d'utilisation «consulter les loyers»	53
Figure 3.23: Diagramme de séquence système du cas d'utilisation «recevoir un paiement» ..	54
Figure 3.24: Diagramme de séquence système du cas d'utilisation «établir une quittance» ...	54
Figure 3.25: Diagramme de séquence système du cas d'utilisation «établir un avis d'échéance»	55
Figure 3.26: Diagramme de séquence système du cas d'utilisation «établir une mise en demeure»	55
Figure 3.27: Diagramme de séquence système du cas d'utilisation «consulter le tableau de bord»	56
Figure 4.1: Modèle du domaine	60
Figure 4.2: Diagramme de classes participantes du cas d'utilisation «s'authentifier»	61
Figure 4.3: Diagramme de classes participantes du cas d'utilisation «créer un compte»	61
Figure 4.4: Diagramme de classes participantes du cas d'utilisation «consulter les comptes»	62
Figure 4.5: Diagramme de classes participantes du cas d'utilisation «modifier un compte» ..	62
Figure 4.6: Diagramme de classes participantes du cas d'utilisation «supprimer un compte»	63
Figure 4.7: Diagramme de classes participantes du cas d'utilisation «ajouter un bien immobilier»	63
Figure 4.8: Diagramme de classes participantes du cas d'utilisation «consulter les biens immobiliers»	64
Figure 4.9: Diagramme de classes participantes du cas d'utilisation «rechercher les biens immobiliers»	64
Figure 4.10: Diagramme de classes participantes du cas d'utilisation «modifier un bien immobilier»	65
Figure 4.11: Diagramme de classes participantes du cas d'utilisation «supprimer un bien immobilier»	65
Figure 4.12: Diagramme de classes participantes du cas d'utilisation «ajouter un locataire» ..	66

Figure 4.13: Diagramme de classes participantes du cas d'utilisation «établir un contrat de location »	66
Figure 4.14: Diagramme de classes participantes du cas d'utilisation «consulter les locataires»	67
Figure 4.15: Diagramme de classes participantes du cas d'utilisation «rechercher les locataires».....	67
Figure 4.16: Diagramme de classes participantes du cas d'utilisation «modifier un locataire»	68
Figure 4.17: Diagramme de classes participantes du cas d'utilisation «supprimer un locataire»	68
Figure 4.18: Diagramme de classes participantes du cas d'utilisation «générer les loyers»....	69
Figure 4.19: Diagramme de classes participantes du cas d'utilisation «consulter les loyers». 69	
Figure 4.20: Diagramme de classes participantes du cas d'utilisation «recevoir un paiement»	70
Figure 4.21: Diagramme de classes participantes du cas d'utilisation «établir une quittance»	70
Figure 4.22: Diagramme de classes participantes du cas d'utilisation «établir un avis d'échéance ».....	71
Figure 4.23: Diagramme de classes participantes du cas d'utilisation «établir une mise en demeure»	71
Figure 4.24: Diagramme de classes participantes du cas d'utilisation «consulter le tableau de bord».....	72
Figure 4.25: Diagramme d'activité de navigation du cas d'utilisation «s'authentifier»	72
Figure 4.26: Diagramme d'activité de navigation du cas d'utilisation «créer un compte».....	73
Figure 4.27: Diagramme d'activité de navigation du cas d'utilisation «consulter les comptes»	73
Figure 4.28: Diagramme d'activité de navigation du cas d'utilisation «modifier un compte»	74
Figure 4.29: Diagramme d'activité de navigation du cas d'utilisation «supprimer un compte»	74
Figure 4.30: Diagramme d'activité de navigation du cas d'utilisation «ajouter un bien immobilier»	75
Figure 4.31: Diagramme d'activité de navigation du cas d'utilisation «consulter les biens immobiliers».....	75
Figure 4.32: Diagramme d'activité de navigation du cas d'utilisation «rechercher les biens immobiliers».....	76

Figure 4.33: Diagramme d'activité de navigation du cas d'utilisation «modifier un bien immobilier»	76
Figure 4.34: Diagramme d'activité de navigation du cas d'utilisation «supprimer un bien immobilier»	77
Figure 4.35: Diagramme d'activité de navigation du cas d'utilisation «ajouter un locataire»	77
Figure 4.36: Diagramme d'activité de navigation du cas d'utilisation «établir un contrat de location»	78
Figure 4.37: Diagramme d'activité de navigation du cas d'utilisation «consulter les locataires».....	78
Figure 4.38: Diagramme d'activité de navigation du cas d'utilisation «rechercher les locataires».....	79
Figure 4.39: Diagramme d'activité de navigation du cas d'utilisation «modifier un locataire»	79
Figure 4.40: Diagramme d'activité de navigation du cas d'utilisation «supprimer un locataire»	80
Figure 4.41: Diagramme d'activité de navigation du cas d'utilisation «générer les loyers» ...	80
Figure 4.42: Diagramme d'activité de navigation du cas d'utilisation «consulter les loyers»	81
Figure 4.43: Diagramme d'activité de navigation du cas d'utilisation «recevoir un paiement»	82
Figure 4.44: Diagramme d'activité de navigation du cas d'utilisation «établir une quittance»	83
Figure 4.45: Diagramme d'activité de navigation du cas d'utilisation «établir un avis d'échéance».....	84
Figure 4.46: Diagramme d'activité de navigation du cas d'utilisation «établir une mise en demeure»	85
Figure 4.47: Diagramme d'activité de navigation du cas d'utilisation «consulter tableau de bord».....	85
Figure 5.1: Diagramme de séquence du cas d'utilisation «s'authentifier»	88
Figure 5.2: Diagramme de séquence du cas d'utilisation «créer un compte».....	88
Figure 5.3: Diagramme de séquence du cas d'utilisation «consulter les comptes»	89
Figure 5.4: Diagramme de séquence du cas d'utilisation «modifier un compte»	89
Figure 5.5: Diagramme de séquence du cas d'utilisation «supprimer un compte»	90
Figure 5.6: Diagramme de séquence du cas d'utilisation «ajouter un bien immobilier».....	90
Figure 5.7: Diagramme de séquence du cas d'utilisation «consulter les biens immobiliers»..	91

Figure 5.8: Diagramme de séquence du cas d'utilisation «rechercher les biens immobiliers»	91
Figure 5.9: Diagramme de séquence du cas d'utilisation «modifier un bien immobiliers».....	92
Figure 5.10: Diagramme de séquence du cas d'utilisation «supprimer un bien immobilier»..	92
Figure 5.11: Diagramme de séquence du cas d'utilisation «ajouter un locataire».....	93
Figure 5.12: Diagramme de séquence du cas d'utilisation «établir un contrat de location»....	93
Figure 5.13: Diagramme de séquence du cas d'utilisation «consulter les locataires»	94
Figure 5.14: Diagramme de séquence du cas d'utilisation «rechercher les locataires»	94
Figure 5.15: Diagramme de séquence du cas d'utilisation «modifier un locataire»	95
Figure 5.16: Diagramme de séquence du cas d'utilisation «supprimer un locataire».....	95
Figure 5.17: Diagramme de séquence du cas d'utilisation «générer les loyers»	96
Figure 5.18: Diagramme de séquence du cas d'utilisation «consulter les loyers».....	97
Figure 5.19: Diagramme de séquence du cas d'utilisation «recevoir un paiement».....	98
Figure 5.20: Diagramme de séquence du cas d'utilisation «établir une quittance».....	98
Figure 5.21: Diagramme de séquence du cas d'utilisation «établir un avis d'échéance»	99
Figure 5.22: Diagramme de séquence du cas d'utilisation «établir une mise en demeure».....	99
Figure 5.23: Diagramme de séquence du cas d'utilisation «consulter le tableau de bord» ...	100
Figure 5.24: Diagramme de classes de conception du cas d'utilisation «s'authentifier»	101
Figure 5.25: Diagramme de classes de conception du cas d'utilisation «créer un compte» ..	101
Figure 5.26: Diagramme de classes de conception du cas d'utilisation «consulter les comptes»	102
Figure 5.27: Diagramme de classes de conception du cas d'utilisation «modifier un compte»	102
Figure 5.28: Diagramme de classes de conception du cas d'utilisation «supprimer un compter».....	103
Figure 5.29: Diagramme de classes de conception du cas d'utilisation «ajouter un bien immobilier»	103
Figure 5.30: Diagramme de classes de conception du cas d'utilisation «consulter les biens immobiliers».....	104
Figure 5.31: Diagramme de classes de conception e du cas d'utilisation «rechercher les biens immobiliers».....	104
Figure 5.32: Diagramme de classes de conception du cas d'utilisation «modifier un bien immobilier»	105
Figure 5.33: Diagramme de classes de conception du cas d'utilisation «supprimer un bien immobilier»	105

Figure 5.34: Diagramme de classes de conception du cas d'utilisation «ajouter un locataire»	106
Figure 5.35: Diagramme de classes de conception du cas d'utilisation «établir un contrat de location»	106
Figure 5.36: Diagramme de classes de conception du cas d'utilisation «consulter les locataires»	107
Figure 5.37: Diagramme de classes de conception du cas d'utilisation «rechercher les locataires»	107
Figure 5.38: Diagramme de classes de conception du cas d'utilisation «modifier un locataire»	108
Figure 5.39: Diagramme de classes de conception du cas d'utilisation «supprimer un locataire»	108
Figure 5.40: Diagramme de classes de conception du cas d'utilisation «générer les loyers»	109
Figure 5.41: Diagramme de classes de conception du cas d'utilisation «consulter les loyers»	109
Figure 5.42: Diagramme de classes de conception du cas d'utilisation «recevoir un paiement»	110
Figure 5.43: Diagramme de classes de conception du cas d'utilisation «établir une quittance»	110
Figure 5.44: Diagramme de classes de conception du cas d'utilisation «établir un avis d'échéance»	111
Figure 5.45: Diagramme de classes de conception du cas d'utilisation «établir une mise en demeure»	111
Figure 5.46: Diagramme de classes de conception du cas d'utilisation «consulter le tableau de bord»	112
Figure 6.1: Une partie du code HTML de notre application.	116
Figure 6.2: Une partie du code CSS de notre application	117
Figure 6.3: Une partie du code JavaScript de notre application.....	118
Figure 6.4: Une partie du code PHP de notre application	119
Figure 6.5: Page «S'authentifier».....	122
Figure 6.6: Page «Accueil»	123
Figure 6.7: Page «créer un compte».....	124
Figure 6.8: Page «consulter les bâtiments»	125
Figure 6.9: Page «modifier un bien immobilier»	126

Figure 6.10: Page «supprimer un locataire».....	127
Figure 6.11: Page «supprimer un locataire».....	127
Figure 6.12: Page «consulter les paiements»	128
Figure 6.13: Page «recevoir un paiement».....	128
Figure 6.14: Page «consulter le tableau de bord»	129

Introduction générale

1. Contexte

Le système d'information joue un rôle centrale au sein d'une entreprise, il est même indispensable à son bon fonctionnement. Il s'agit de l'ensemble de ressources qui permettent de collecter, stocker, traiter, et diffuser les informations entre les différents intervenants d'un organisme. L'avènement de l'informatique a permis d'automatiser petit à petit les systèmes d'information des entreprises, ce qui a amélioré considérablement le stockage et le traitement. Les réseaux informatiques et les technologies web ont révolutionné encore plus la gestion de l'information au sein de l'entreprise. Les réseaux intranet ont permis d'interconnecter les différents services de l'organisme, cela a rendu possible la mise en œuvre d'applications client/serveur qui permettent d'avoir un accès centralisé et cohérent aux ressources. Finalement, le besoin croissant des entreprises d'interconnecter ses sites distants de façon simple et économique, a favorisé l'apparition de la technologie VPN (Virtual Private Network) permettant à des postes distants de communiquer de manière sûre, tout en empruntant le réseau internet.

C'est dans ce contexte que plusieurs entreprises y compris l'OPGI essayent de profiter au maximum possible de ces technologies afin d'améliorer leurs productivités et de faire face à quelques problèmes pénibles qui peuvent constituer un obstacle de progression.

2. Motivation

L'OPGI de Mila a introduit l'informatique dans la gestion locative depuis des années, ceci a permis une automatisation partielle de cette tâche, grâce à l'exploitation d'un de

certaines nombres d'applications par les différents intervenants chargés de cette mission. Cependant, la gestion locative au niveau de l'office est assurée par plusieurs intervenants réparties géographiquement sur plusieurs sites distants, ce qui explique le recours à des applications monopostes qui ne permettent pas la centralisation des ressources, et obligent les intervenants de partager les données mise à jour via le courrier électronique. La solution informatique actuelle de la gestion locative montre ainsi de nombreuses limites, comme le ralentissement du travail, les difficultés rencontrées par les locataires pour régler leurs loyers, la confidentialité des données échangées sur le réseau internet, etc. D'où la nécessité de mettre en place une nouvelle solution, afin de répondre aux limites de la solution actuelle.

3. Objectif du mémoire

L'objectif de notre travail consiste à l'élaboration d'une nouvelle solution pour la gestion locative au niveau de l'OPGI de Mila. Cette solution consiste principalement à :

- Proposer une solution pour interconnecter les différents intervenants chargés de la gestion locative par un réseau intranet.
- Développer une application web pour la gestion locative accessible sur le réseau intranet. Le principal objectif est la centralisation des ressources (données et traitement), ce qui permettrait de résoudre les limites de la solution actuelle.

4. Plan du mémoire

- Le premier chapitre est principalement consacré à l'étude de la gestion locative au sein de l'OPGI, et à l'identification des limites de la solution actuelle.
- Le deuxième chapitre présente un aperçu des applications web et les réseaux VPN, puis, propose une nouvelle solution pour la gestion locative, basée sur la mise en place d'un réseau intranet VPN, et le développement d'une application web accessible sur ce réseau.
- **Le troisième chapitre** est consacré à la phase d'identification des besoins. La phase décrit le cahier de charge qui exprime les besoins du maître d'ouvrage, elle décrit aussi les acteurs du futur système, les cas d'utilisation et leurs classements ainsi que les descriptions textuelles et graphiques de chaque cas d'utilisation.

- **Dans le quatrième chapitre**, nous nous intéressons à la phase d'analyse. A ce titre nous élaborons le modèle de domaine, nous réalisons les diagrammes de classes participantes et les diagrammes d'activités de navigation.
- **Le cinquième chapitre** est consacré à la phase de conception. Nous construisons les diagrammes de séquences et les diagrammes de classes de conception.
- **Le sixième chapitre** décrit brièvement les outils utilisés pour réaliser l'application, et présente quelques interfaces de l'application réalisée.
- **Enfin**, en conclusion générale, nous résumons nos contributions à l'évolution de l'application et le résultat obtenue, et nous terminons le mémoire par un aperçu des futures perspectives.

Chapitre 1

Étude de l'existant : Gestion locative au sein de l'OPGI de Mila

1.1 Introduction

Afin de mettre en œuvre un nouveau système informatique, il est essentiel de faire une étude de l'existant, pour comprendre la solution actuelle, et identifier ses inconvénients, afin de concevoir une solution qui permettra de résoudre les problèmes posés. Dans ce chapitre nous commençons par un aperçu de l'organisme d'accueil (OPGI de Mila), ses missions et son organisation, ensuite, nous nous intéressons à l'activité de la gestion locative, les différents intervenants concernés par cette activité, et les systèmes informatiques actuellement utilisés pour automatiser cette fonction. Finalement, nous essayons d'identifier les différents inconvénients de la solution informatique actuelle.

1.2 Présentation de l'organisme (OPGI de Mila)

1.2.1 Statut juridique

L'Office de Promotion et de Gestion Immobilière de la wilaya de Mila (OPGI de Mila) a été créé à l'instar des autres offices répartis à travers le territoire national par le

décret 76/93 du 23/10/1976 fixant les conditions de création des OPGI. Sa nature juridique a été transformée par le décret exécutif n°91/147 du 21 mai 1991 portant modification de la nature juridique des statuts des OPGI.

1.2.2 Objet

Conformément à l'article 1 du décret exécutif, l'Office de Promotion et de Gestion Immobilières de la wilaya de Mila est un établissement public doté de la personnalité morale et de l'autonomie financière. Il est réputé commerçant dans leur rapport avec les tiers et est soumis aux règles de droit commercial.

1.2.3 Missions de l'office

Conformément à l'article 4 du décret exécutif n° 91/147 du 12/05/1991, l'OPGI est investi des missions suivantes : Promouvoir le service public en matière de logement notamment pour les catégories sociales les plus démunies, il est chargé: De la promotion immobilière et foncière, de la maîtrise d'ouvrage déléguée de l'état ou pour le compte de tout autre opérateur. Des actions de prestation de services en vue d'assurer l'entretien, la maintenance, la restauration et la réhabilitation des biens immobiliers, de toute action visant l'accomplissement de ses missions. En outre, il est habilité à gérer les biens immobiliers qui lui sont confiés suivant une convention déterminant les conditions et modalités de prise charge de cette mission ainsi que l'administration des biens immobiliers.

1.2.4 Structure interne

L'OPGI de Mila est animée par 259 fonctionnaires dont le directeur, les chefs de départements, les chefs d'unités et les chefs de services, ces fonctionnaires sont regroupés entre sept unités distribués géographiquement et entre cinq principaux départements qui sont cités par la suite(Figure1.1) :

A. Département de ressources humaines et moyennes généraux (DRHMG) :

Il est chargé de la conception et de la mise en œuvre de la politique de gestion des ressources humaines et de la gestion des moyens matériels de l'office.

B. Département finances et comptabilités (DFC) :

Il est chargé essentiellement: de définir et d'appliquer la stratégie en matière financières de développer les investissements les plus rémunérateurs. D'établir le budget, de procéder à la consolidation des bilans d'assister toutes les structures en matières de fiscalité et de procéder à tous les contrôles liés à son activité. Du recouvrement des loyers et des charges locatives ainsi que des produits de la cession des biens immobilières

C. Département de la gestion et de la maintenance du patrimoine (DGMP) :

Il est chargé essentiellement: De la location et/ou de la cession des logements et locaux à usage d'habitation, professionnel, commercial et artisanal et l'établissement de leurs constatations. De l'établissement et de la tenue à jour de l'inventaire des immeubles constituant le parc immobilier dont il assure la gestion ainsi que le contrôle de la nature juridique de l'occupation des logements et des locaux (fichier de biens immobiliers). De gérer pour le compte d'autrui par convention tout bien à usage d'habitation et/ou commercial. De gérer pour le compte d'autrui par convention tout bien à usage d'habitation et/ou commercial. De la maintenance, de la préservation du parc immobilier et de la réhabilitation du cadre bâti.

D. Département de maîtrise d'ouvrage (DMO) :

Il est chargé de : La mise en œuvre de la politique de l'Etat en matière de logement social. La maîtrise d'ouvrage déléguée pour le compte de tout autre opérateur. La promotion immobilière en matière d'habitat et l'équipement.

E. Département du développement de la promotion immobilière et foncière(DPIF) :

Il est chargé essentiellement des missions suivantes: La mise en œuvre d'une stratégie en matière de promotion immobilière. La prospection et l'acquisition des terrains destinés à l'implantation des projets. La prospection et l'acquisition des terrains destinés à l'implantation des projets [1].

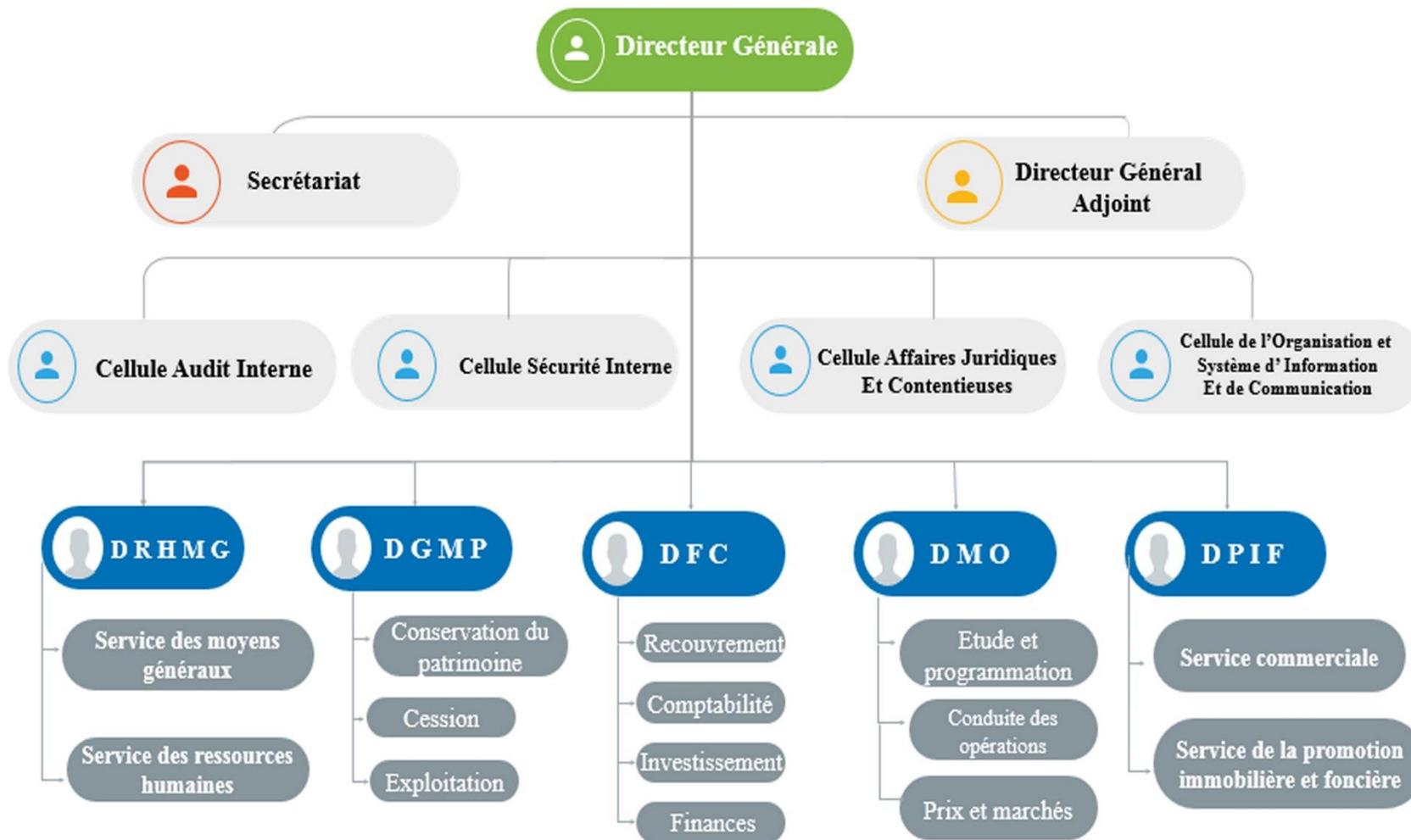


Figure 1.1: Organigramme de l'OPGI de Mila

1.3 Présentation de l'existant

1.3.1 Gestion locative

La gestion locative est l'action pour une société, un organisme ou un individu (qui est désigné sous le nom de mandataire) de gérer les biens immobiliers de son mandant (en général le propriétaire du ou des biens immobiliers) [2]. C'est la prise en charge par un propriétaire ou par un intervenant extérieur de la gestion d'un bien immobilier mis en location par un propriétaire. La gestion locative, est donc un ensemble des démarches à effectuer une fois que le locataire est dans les lieux. Autrement dit, il s'agit de gérer, au quotidien, tout ce qui est nécessaire au bon déroulement de la location. L'action de gérer regroupe tout un ensemble d'autres actions, en résumé les différentes missions du mandataire sont les suivantes :

- La mise en publicité des biens immobiliers,
- La recherche des locataires,
- L'établissement des contrats de location,
- La rédaction des états de lieux d'entrée et de sortie, la perception des loyers et des charges,
- L'envoi des avis d'échéances et quittances aux locataires,
- La régularisation des charges locatives,
- La révision annuelle des loyers, la gestion des impayés et l'envoi des mises en demeure.

1.3.2 Gestion locative au niveau de l'OPGI de Mila

La gestion locative des biens immobiliers fait partie des missions de l'OPGI (voir la section 1.2.3). Les tâches de la gestion locative au sein de l'OPGI sont partagées entre le Département de gestion et de la maintenance du patrimoine (DGMP), le Département des finances et de la comptabilité (DFC) et les caisses (Figure 1.2).

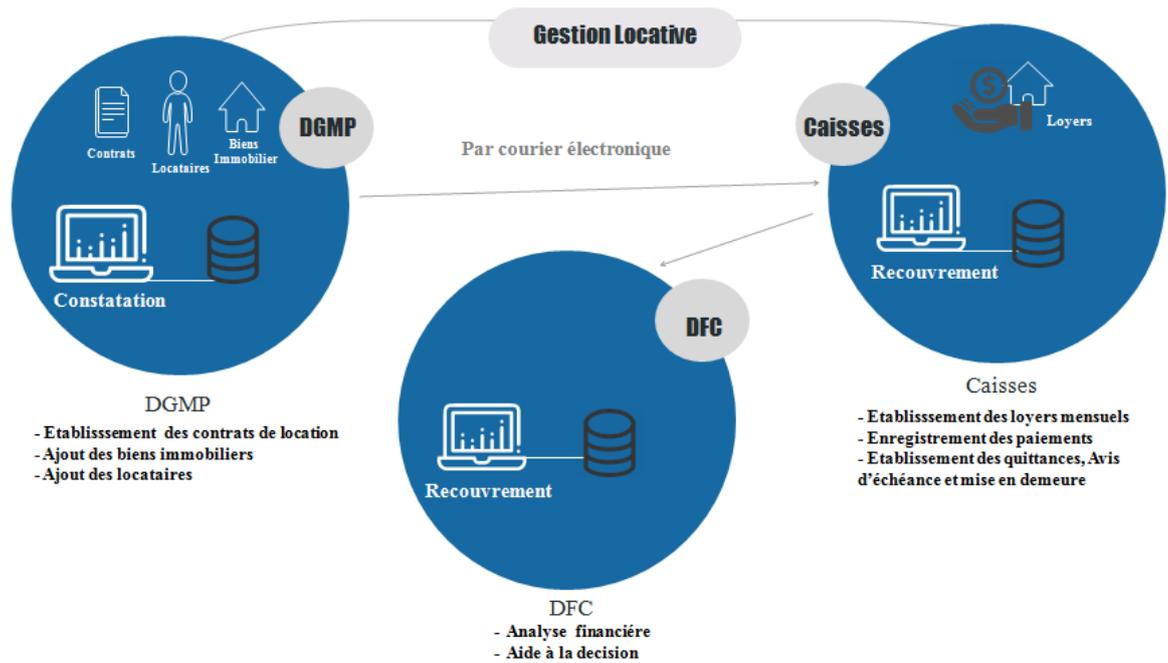


Figure 1.2: La gestion locative au niveau de l'OPGI de Mila

Le **DGMP** prend en charge la gestion des biens immobiliers et des locataires : Ajout des biens immobiliers et des locataires, et l'établissement des contrats de location pour les nouveaux locataires.

Les **CAISSES** sont réparties géographiquement selon l'emplacement du patrimoine de l'office, c'est-à-dire on trouve une ou plusieurs caisses par commune. Le nombre des caisses par commune dépend du patrimoine de l'office. Les caisses prennent en charge la gestion des loyers : Génération des loyers, l'encaissement des loyers, ainsi que l'établissement des avis d'échéance, des quittances et des mises en demeures.

Le **DFC** prend en charge l'analyse des flux financiers de l'OPGI, y compris les paiements de loyers, afin de donner des informations sur la situation financière de l'office. Ces informations servent pour prendre les décisions au niveau de la direction générale de l'office.

1.3.3 Informatisation de la gestion locative au sein de l'OPGI de Mila

Afin d'automatiser la gestion locative, l'OPGI de Mila s'est doté d'un ensemble de logiciels qui sont exploités par les différents départements de l'office qui sont concernés par cette mission.

- **L'application *Constatation***

L'application *Constatation* est exploitée au niveau du DGPM (Figure1.3). Ses principales fonctionnalités sont l'ajout, la modification, et la suppression des biens immobiliers, des locataires et des contrats de location dans une base de données. Les mises à jour de la base de données sont ensuite envoyées aux caisses pour permettre à ces derniers de prendre en charge les paiements des loyers par les locataires concernés.

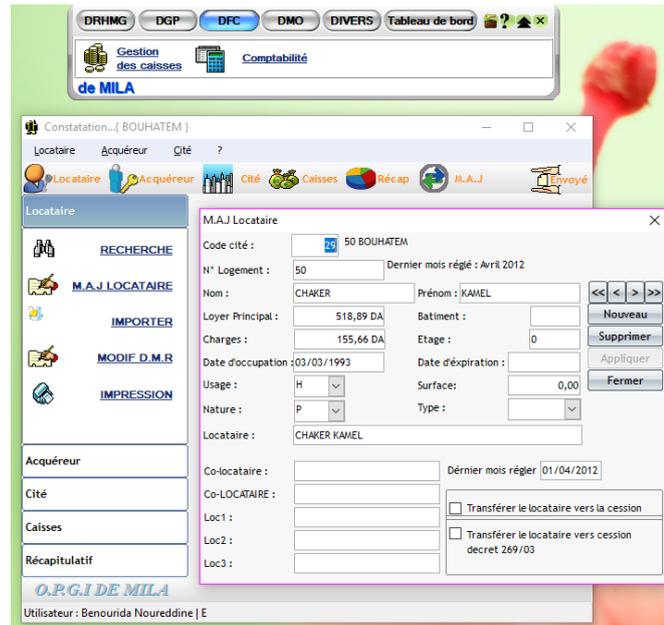


Figure 1.3: Interface de l'application Constatations

- **L'application *Recouvrement***

L'application *Recouvrement* est exploitée au niveau des caisses et du DFC.

Au niveau des caisses, ses principales fonctionnalités sont : la génération automatique des loyers mensuels, l'établissement des avis d'échéance et des mises en demeure, l'enregistrement des paiements des loyers et l'établissement des quittances. Les caisses envoient périodiquement leurs bases de données par courrier électronique au département DFC qui prend en charge l'analyse financière et l'aide à la décision (Figure1.4).

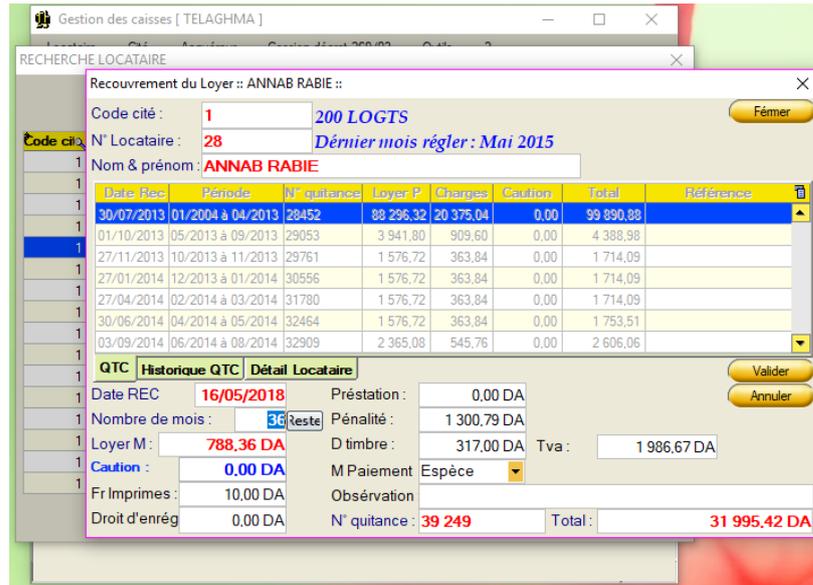


Figure 1.4: Interface de l'application Recouvrement (Au niveau des caisses)

Au niveau du DFC, les principales fonctionnalités de l'application *Recouvrement* sont l'analyse financière et l'aide à la décision, en se basant sur les données qui proviennent des caisses de l'office (Figure 1.5).

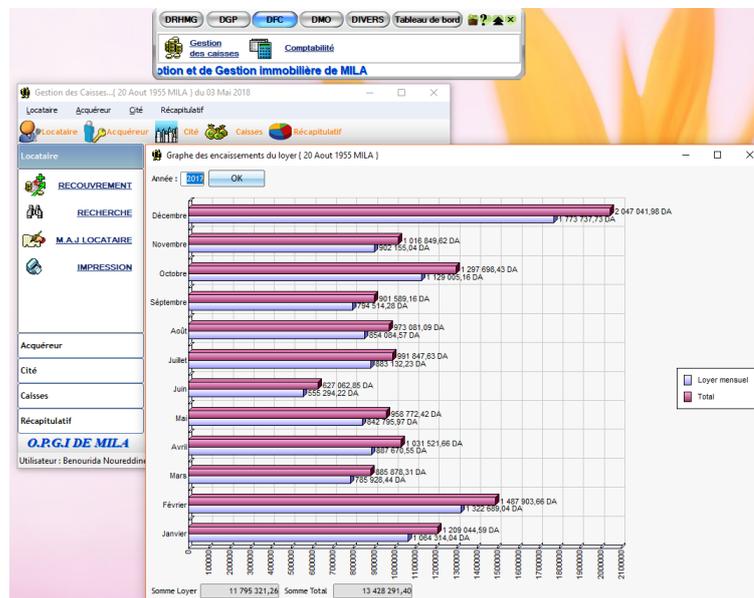


Figure 1.5: Interface de l'application Recouvrement (Au niveau du DFC)

1.4 Critique de l'existant

La gestion locative au sein de l'OPGI de Mila est partiellement automatisée grâce à l'exploitation des logiciels présentés dans la section (1.3.3). Cependant, les structures chargées de la gestion locative au sein de l'OPGI (le DGMP, le DFC, et les caisses) sont réparties géographiquement sur plusieurs sites distants, ce qui rend difficile la liaison de ces sites par un réseau intranet, et ainsi, la mise en œuvre d'un système informatique centralisé pour la gestion locative. Ce qui explique le recours à des applications monopostes réparties sur plusieurs machines et plusieurs endroits. Chaque application manipule une base de données contenant une copie des données nécessaires pour la réalisation des fonctionnalités. Cette solution pose de nombreux problèmes que nous pouvons résumer comme suit:

- Redondance des données : L'existence de plusieurs copies de données exploitées par les applications de la gestion nécessite des mises à jour sur toutes les copies à chaque fois qu'une copie est mise à jour, ce qui peut conduire à l'incohérence de données, et causer des erreurs de traitement en cas de l'utilisation de données qui ne sont pas à jour.
- Les mises à jour de bases de données sont transmises par courrier électronique sur le réseau internet, ce qui pose des problèmes de confidentialité de données pour l'office et ses locataires.
- La perte de temps au niveau du département DFC par exemple qui doit attendre la transmission des paiements des loyers de la part de toutes les caisses de l'office pour pouvoir réaliser l'analyse financière.
- Un locataire ne peut payer ses loyers que dans une seule caisse, ce qui ralentit la collecte des loyers.
- Maintenance difficile : Une maintenance de l'application Recouvrement utilisée au niveau des caisses, entrainera des maintenances de l'application dans toutes les caisses de l'office.

1.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés particulièrement à l'étude de la gestion locative au sein de l'OPGI, et la solution informatique actuellement utilisée pour automatiser cette tâche. L'étude de l'existant réalisée a permis de déceler de nombreux inconvénients qui rendent difficile le recouvrement des loyers, et posent des problèmes de confidentialités des données. Ces problèmes nous motivent pour réfléchir à une nouvelle solution informatique pour la gestion locative au sein de l'OPGI de Mila.

Chapitre 2

Proposition d'une nouvelle solution

2.1 Introduction

Après avoir réalisé une étude de la gestion locative au sein de l'OPGI de Mila, nous avons identifié de nombreuses difficultés qui influencent négativement le bon déroulement de cette activité. Les principaux inconvénients de la solution actuelle découlent de l'absence d'un réseau intranet reliant les différents intervenants concernés par la mission de la gestion locative, et par conséquent, l'impossibilité de mettre en œuvre une application qui centralise les différentes tâches de la gestion locative. Ainsi, nous avons envisagé une solution qui consiste à mettre en œuvre un réseau intranet VPN reliant les différents intervenants de la gestion locative, et le développement d'une application web accessible à travers ce réseau, et qui permettra de centraliser la gestion locative. Dans ce chapitre, nous allons d'abord nous intéresser aux techniques qui seront utilisées pour mettre en œuvre notre solution, à savoir, les applications web et leurs architectures, ainsi que les réseaux VPN. Finalement, nous détaillons la solution proposée, et nous montrons ses avantages par rapport à la solution existante.

2.2 Les applications Web

Pour résoudre les problèmes posés par la gestion locative au niveau de l'OPGI actuellement, nous avons envisagé le développement d'une application web reliant les différents services concernés, et permettra ainsi une gestion locative plus efficace. Ainsi,

nous allons nous intéresser dans cette section aux applications web, et leurs différentes architectures avec leurs avantages et leurs inconvénients, afin de choisir l'architecture adéquate pour notre application.

2.2.1 Notion d'une application web

Une application web (aussi appelée web application, en anglais) désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web (Google Chrome, Mozilla Firefox, etc.), par l'intermédiaire d'un réseau informatique (Internet, intranet, etc.). Contrairement à une application de bureau qui nécessite l'installation d'un logiciel dans chaque environnement dans lequel il est utilisé. L'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur favori (Figure 2.1) [3].

Une application est composée de 3 parties appelées: couches, niveaux ou tiers):

- La présentation des données (L'interface utilisateur) : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- Services métier (couche traitement des données): correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- Persistance (L'accès aux données persistantes) : correspondant aux données qui sont destinées à être conservées.

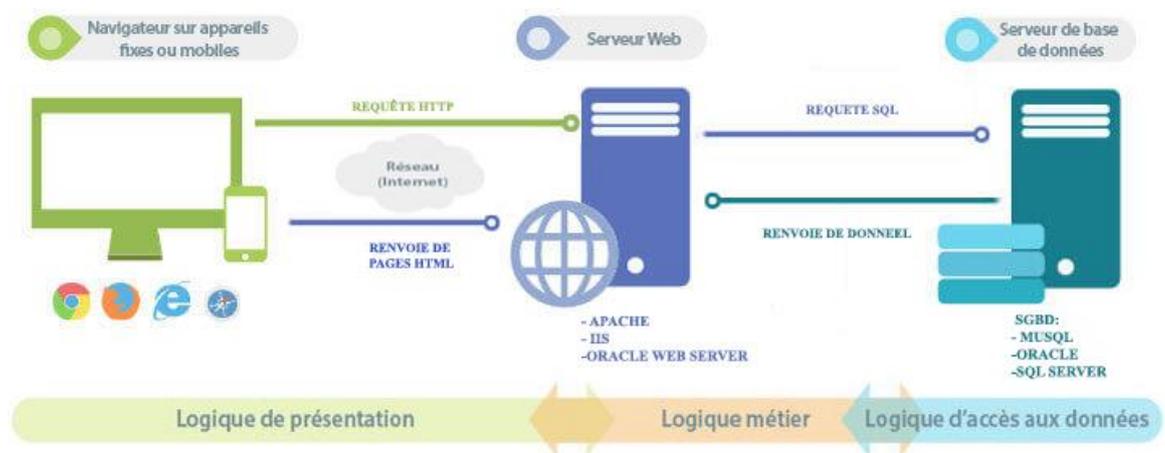


Figure 2.1: Composants d'architecture d'une application Web

2.2.2 Architectures des applications web

Les applications web fonctionnent en mode client/serveur c'est-à-dire qu'il y a des machines dites serveurs qui proposent des ressources et des machines appelées clients qui utilisent ces ressources. Une application web implique au moins l'existence des 3 composants : (1) Client web, (2) Serveur web, (3) Serveur de bases de données (Figure2.1).

2.2.2.1 Client web

Le client web est un navigateur qui peut interagir avec un serveur web via le protocole HTTP, interpréter les pages HTML et XML, et exécuter des programmes (JavaScript, les applets, etc.).

2.2.2.2 Serveur web

Un serveur web est un serveur capable d'exécuter des applications web. C'est-à-dire des applications destinées à traiter des requêtes qui proviennent des clients web, et à générer une réponse adéquate.

2.2.2.3 Serveur de données

Un serveur de données permet de gérer l'accès aux données par le serveur web. Cet accès passera par l'utilisation de bibliothèques standards d'accès aux données, telles que RDO, ADO, ODBC, JDBC, etc.

2.2.3 Types d'architectures des applications web

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc. Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit de courrier électronique)[3].

À un haut niveau, on peut identifier dans les applications web d'aujourd'hui plusieurs architectures. Les plus courants actuellement sont les suivants :

2.2.3.1 Client web très léger

Il est employé pour les applications destinées à Internet, pour lesquelles la configuration du client n'est pas maîtrisable. Le client ne nécessite qu'un navigateur web standard et la logique métier ainsi que la logique de présentation sont intégralement exécutées sur le serveur : Seul le langage HTML est utilisé côté client, ce qui maximise l'accessibilité au site (tous les navigateurs web sont aptes à y naviguer) et la sécurité du poste client. Ce pattern implique toutefois d'innombrables allers-retours entre navigateur et serveur web : cela touche donc l'interactivité.

2.2.3.2 Client web léger

Il correspond également aux applications Internet/intranet pour lesquelles la configuration du client n'est pas maîtrisable, à ceci près que l'on requiert côté client un navigateur web assez récent, comprenant le langage JavaScript. Le client navigue sur des pages dotées d'intelligence (programmée en JavaScript), donc :

- L'interactivité est améliorée (la validation des formulaires est réalisée côté client, avant l'envoi de requêtes sur le serveur web).

- La plupart des navigateurs modernes proposent des options fines sur ce qu'ils autorisent JavaScript à faire ou non (par exemple, lui interdire de désactiver le menu contextuel ou de changer le texte de la barre d'état, d'ouvrir des fenêtres trop fréquemment, etc.).

- La portabilité des pages est assez facile à garantir (par exemple par l'emploi de bibliothèques telles que Prototype).

Cependant, JavaScript doit être utilisé pour le confort : l'accessibilité implique que s'il est désactivé, l'application doit rester utilisable avec une ergonomie raisonnable ou au pire tolérable.

2.2.3.3 Client web alourdi

Il embarque dans les pages web des composants plus complexes : ActiveX, applets Java, plug-ins, ce qui permet à une partie significative de la logique métier d'être exécutée sur le poste client. Du coup :

- L'interface graphique peut être beaucoup plus évoluée (graphiques dynamiques...).
- Le composant peut se rafraîchir spontanément (et télécharger des données brutes du serveur).
- Le déploiement, bien qu'automatique, est beaucoup plus lourd. Les prérequis sur les postes sont plus importants.
- Le risque est beaucoup plus important (typiquement avec les ActiveX et les plug-ins qui s'exécutent sans restriction sur le système d'exploitation).

2.2.3.4 Client web lourd

C'est une application à part entière, qui s'exécute sur le poste client. Ce pattern correspond typiquement aux applications Intranet pour lesquelles on maîtrise la configuration du client :

- La contrainte du déploiement de ces logiciels est moins forte aujourd'hui que par le passé, grâce à l'apparition de techniques de livraison « via le Web » (web delivery).
- La communication client/serveur est moins problématique ; en effet, les protocoles réseau sont pour la plupart persona non grata sur le Web à cause des contraintes imposées par les pare-feu depuis l'avènement des services web [3].

2.3 Généralité sur les VPN

2.3.1 C'est quoi un VPN

VPN signifie Virtual Private Network (en français Réseau Virtuel Privé), est une technique permettant à un ou plusieurs postes distants de communiquer de manière sécurisée, tout en empruntant des réseaux publics (Internet). Ce type de liaison est

apparu suite à un besoin croissant des entreprises à relier des sites distants, et ce de façon simples et économique [4].

2.3.2 Principe de fonctionnement d'un VPN

Un VPN repose sur des protocoles, appelés protocoles de tunnelisation (ou tunneling), qui sont des protocoles permettant aux données passant entre deux réseaux physiques d'être sécurisées par des algorithmes de chiffrement.

On utilise le terme de « tunnel » pour mettre l'accent sur le fait qu'entre l'entrée et la sortie d'un VPN les données sont chiffrées et protégées. Lorsqu'un VPN est établi entre deux réseaux physiques, l'élément qui permet de chiffrer et de déchiffrer les données du côté client (ou utilisateur) est nommé « Client VPN ». On appelle « Serveur VPN » l'élément qui chiffre et qui déchiffre les données du côté de l'organisation (Figure 2.2) [5].

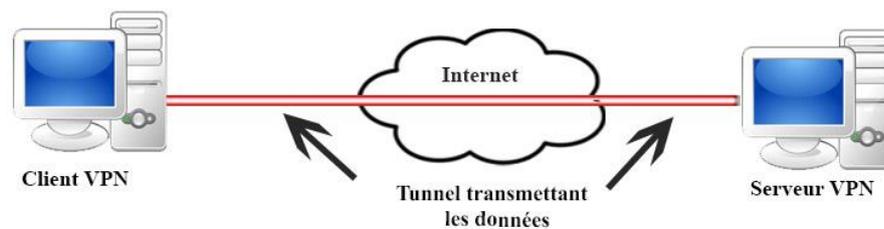


Figure 2.2: Fonctionnement d'un VPN

2.3.3 Les différents types de VPN

Selon le mode d'utilisation, on distingue trois types d'architecture VPN :

- Le VPN d'accès.
- L'intranet VPN.
- L'extranet VPN.

2.3.3.1 VPN pour l'accès itinérant

Ce type de VPN est utilisé pour permettre à des utilisateurs itinérants d'accéder au réseau de leurs entreprises. L'utilisateur se sert d'une connexion Internet afin d'établir une liaison sécurisée (Figure2.3) [6].

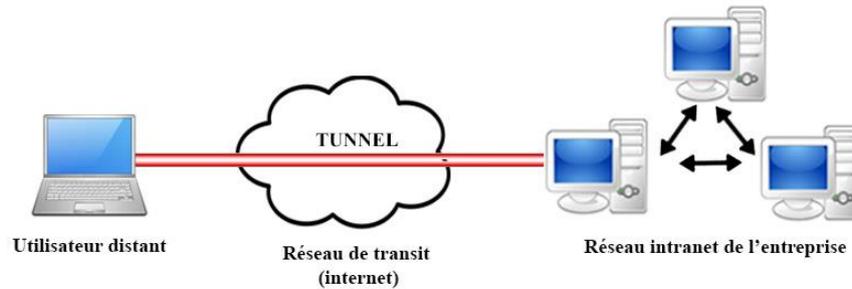


Figure 2.3:VPN d'accès itinérant

2.3.3.2 VPN intranet

L'intranet VPN est utilisé pour relier deux ou plusieurs intranets d'une même entreprise entre eux. Ce type de réseau est particulièrement utile au sein d'une entreprise possédant plusieurs sites distants. Le plus important avec ce type de VPN est de garantir la sécurité et l'intégrité des données. (Figure2.4) [6].

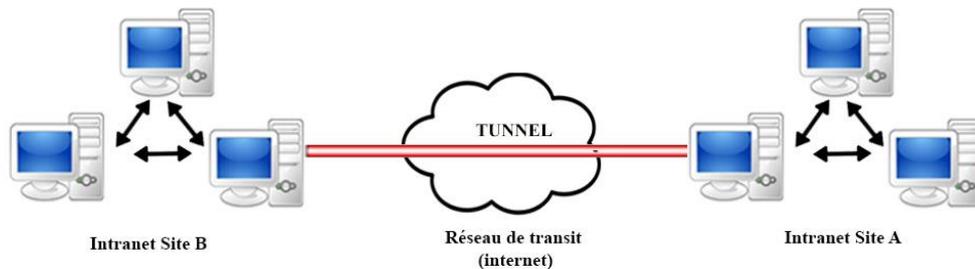


Figure 2.4: VPN intranet

2.3.3.3 VPN extranet

Une entreprise peut utiliser le VPN pour communiquer avec ses clients et ses partenaires. Elle ouvre alors son réseau local à ces derniers. Dans ce cas, il est nécessaire d'avoir une authentification forte des utilisateurs, ainsi qu'une trace des différents accès.

De plus, seule une partie des ressources sera partagée, ce qui nécessite une gestion rigoureuse des espaces d'échange (Figure 2.5) [6].

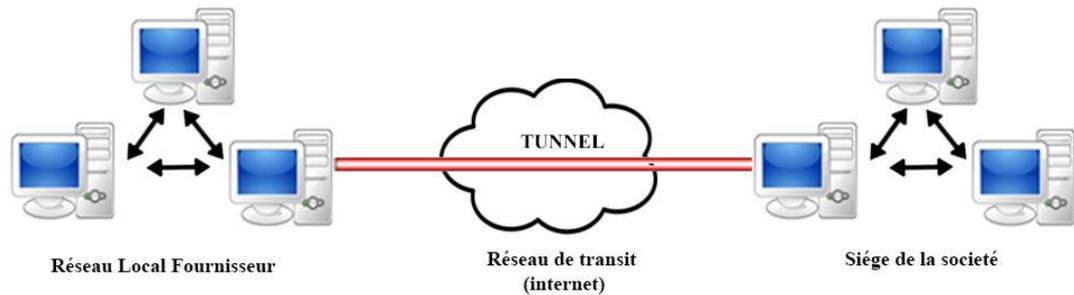


Figure 2.5: VPN extranet

2.3.4 Protocoles utilisés pour réaliser une connexion VPN

Il existe de nombreux protocoles de tunnelisation permettant de réaliser des connexions VPN. Les principaux protocoles de tunnelisation sont:

2.3.4.1 Le protocole PPTP

PPTP (Point to Point Tunneling Protocol) .Microsoft a implémenté ses propres algorithmes afin de l'intégrer dans ses versions de Windows. PPTP est une solution très employée dans les produits VPN. PPTP est ainsi un protocole de niveau 2 qui permet l'encapsulation des données ainsi que leur compression [4].

2.3.4.2 Le protocole L2TP

L2TP (Layer Two Tunneling Protocol) est un protocole combinant les avantages du PPTP de Microsoft et du L2F de Cisco, ce protocole L2TP est aujourd'hui principalement utilisé par les FAI (Fournisseurs d'Accès à Internet). Sur la base des spécifications des protocoles L2F et PPTP. L2TP encapsule les trames PPP qui encapsulent ensuite les protocoles IP et permettent aux utilisateurs d'exécuter à distance des programmes qui sont tributaires de protocoles réseau déterminés [7].

2.3.4.3 Le protocole IP Sec

IP Sec (Internet Protocol Security) est un ensemble de protocoles pour sécuriser les communications IP et garantir le chiffrement, l'intégrité et l'authentification. Ce protocole spécifie les messages nécessaires pour sécuriser les communications du réseau privé. Aujourd'hui, le protocole le plus utilisé pour la mise en place des VPNs est IPSec. Il est

l'un des standards les plus diffusé et le plus ouvert. Effectivement IPSec vise sécuriser les échanges au niveau de la couche réseau [8].

2.3.4.4 Le protocole SSL

(Secure Socket Layer), il offre une très bonne solution de tunnelisation. L'avantage de cette solution est de permettre l'utilisation d'un navigateur Web comme client VPN : on peut accéder à ce type de VPN avec un navigateur web via «https». Dans les faits, il permet aux utilisateurs de mettre en place une connexion sécurisée au réseau depuis n'importe quel navigateur Web [5].

2.3.5 Sécurité du VPN

Pour renforcer la sécurité et la protection de réseau VPN, il est aussi indispensable d'avoir un pare-feu (Firewall) qui est un matériel ou une application qui permet de filtrer les connexions sortantes et entrantes qui sont effectuées depuis et vers votre ordinateur, dans le but de contrôler tout ce qui entre pour limiter les attaques extérieures. Le firewall se situe entre d'un côté l'utilisateur et ses applications et de l'autre la connexion internet.

Le firewall refuse et interdit tout ce qui n'a pas été explicitement autorisé. De ce fait, il défend les ordinateurs contre les virus [19].

2.4 Solution proposé

2.4.1 Présentation de la solution

1. **Mise en place d'un réseau intranet VPN:** La centralisation de la gestion locative nécessite la mise en œuvre d'un réseau intranet reliant les différents sites chargés de cette tâche. Cependant, ces sites sont réparties géographiquement sur l'ensemble du territoire de la Wilaya. Ainsi, la mise en œuvre d'un réseau intranet à l'aide d'une liaison spécialisée est couteuse, et non envisageable par conséquent. Une solution adéquate à cout faible pour ce besoin consiste à mettre en œuvre un réseau intranet VPN (Figure2.6). Cette solution consiste en :
 - L'installation d'un réseau intranet au niveau du DGPM, reliant les postes de travail du département à un serveur Web.

- Mise en place de liaisons entre le serveur et les postes de travail au niveau des caisses et du DFC en utilisant Internet comme support de transmission à l'aide d'un protocole de tunnelisation.

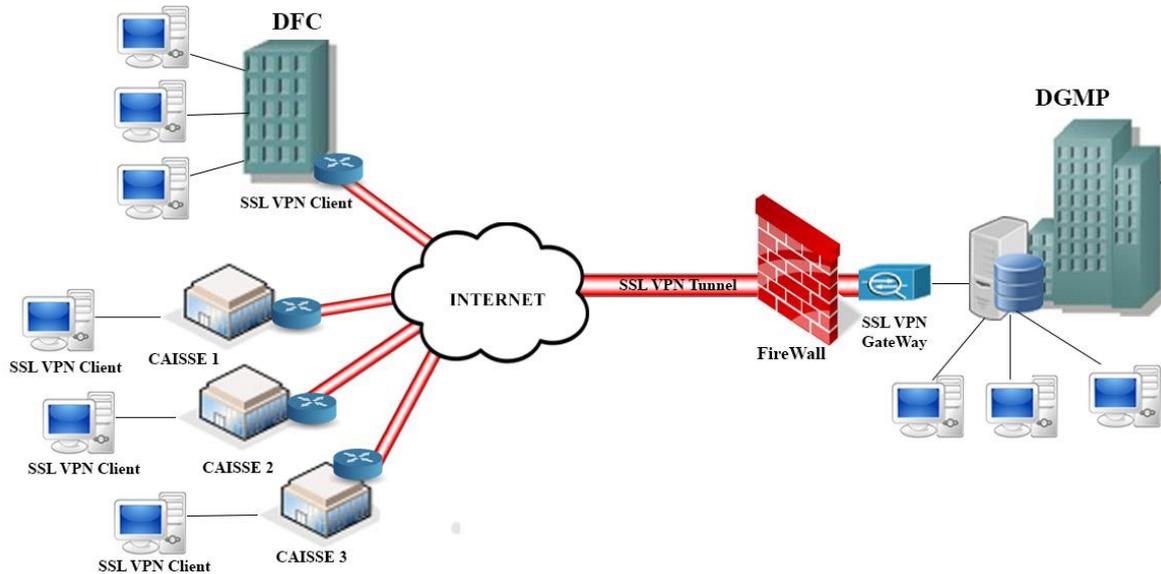


Figure 2.6: VPN intranet au niveau de l'OPGI Mila

2. **Le développement d'une application web pour la gestion locative** : La mise en place du réseau intranet reliant les différentes structures de l'office permettra de mettre en œuvre une application web qui centralise la gestion locative au sein de l'office. Nous avons opté pour une architecture client léger, car :

D'abord l'utilisateur d'un client léger n'a besoin que d'un navigateur internet pour accéder aux services de l'application. Par opposition à un client lourd qui nécessite l'installation du logiciel sur chaque poste utilisateur.

Ensuite, un client léger communique avec un serveur central de traitement, cela permet aux systèmes des utilisateurs d'être gérés de manière centralisée sur un seul serveur central, cela facilite le déploiement et la maintenance. Contrairement à des clients lourds qui ne reposent pas sur un serveur central de traitement parce que le traitement est effectué localement sur le système de l'utilisateur.

Ainsi, un client léger est mieux adapté aux environnements dans lesquels les mêmes informations vont être accessibles par des différents utilisateurs, ce qui présente une meilleure solution pour cet environnement.

Finalement, pour maintenir un client lourd, il est nécessaire de maintenir tous les postes utilisateurs pour le déploiement et les mises à niveau des logiciels, plutôt que de simplement maintenir les applications une seule fois sur un seul serveur centrale.

2.4.2 Avantage de la solution

La solution proposée permettra de résoudre les problèmes de la solution actuelle (Voir la section 1.4). Les principaux avantages de la solution proposée sont:

- **Gestion centralisée des données:** Toutes les données sont stockées dans une base de données centralisée au niveau du serveur, et gérées par une application centralisée, ce qui permettra d'éviter les problèmes de redondance et d'incohérence.
- **Une meilleure sécurité de données:** les données circulent sur le réseau intranet, et sont stockées au niveau du serveur du DGMP, ce qui garantira une meilleure sécurité et confidentialité de données.
- **Facilité de maintenance:** En cas d'une maintenance de l'application Web, seul le serveur sera concerné.
- **Facilité de paiement des loyers:** un locataire peut régler ses loyers dans n'importe quelle caisse, ce qui permettra d'améliorer le recouvrement des loyers.
- **Gain de temps:** Les fonctionnaires du DFC ont la possibilité de réaliser l'analyse financière à tout moment.

2.5 Conclusion

Dans ce chapitre nous avons présenté les concepts de base relatifs aux applications web et leurs architectures, puis, nous avons mis en relief les VPN, leurs types et protocoles. Ceci nous a permis de saisir les connaissances essentielles pour se lancer dans notre projet, et de choisir la solution qui sera utilisée dans notre application.

Chapitre 3

Identification des besoins

3.1 Introduction

Dans le chapitre précédent, nous avons pu proposer une solution appropriée pour interconnecter les intervenants de la gestion locative par un réseau intranet VPN, ce qui ouvre la voie à la mise en œuvre d'une application web qui centralise cette activité au sein de l'OPGI. Ce chapitre est consacré principalement à l'identification des besoins qui est la première phase de réalisation de l'application. Dans un premier temps nous allons présenter brièvement le processus UML utilisée pour la conception de l'application, puis, nous introduisons le cahier des charges qui nous servira de fil conducteur tout au long de la réalisation de l'application. Après, nous identifions les acteurs interagissant avec le système afin d'établir précisément les frontières du système, finalement, nous identifions les différents cas d'utilisation et nous réalisons leurs descriptions textuelles et graphiques.

3.2 Langage et processus pour l'analyse et la conception

Pour passer de l'expression des besoins au code de l'application nous avons besoin d'une méthode, cette dernière doit fournir un langage de modélisation graphique et un processus de développement. Dans notre cas, nous allons adopter UML (Unified Modeling Language) comme un langage de modélisation graphique avec un processus de développement (une démarche) simplifié, orienté par les cas d'utilisations.

3.2.1 Langage UML

UML (Unified Modeling Language) est un langage de modélisation graphique qui permet la spécification, la représentation et la construction des composantes d'un système informatique. UML est né de la fusion de trois méthodes d'analyse orientées objet en 1995: OMT (Object Modeling Technique), OOD (Object Oriented Design) OOSE (Object Oriented Software Engineering). Depuis, il est devenu incontournable pour l'analyse et la conception de système informatique. Pour modéliser les systèmes informatiques, UML offre un ensemble de types de représentations graphiques qu'on appelle « Diagrammes ». Chaque type de diagramme possède ses propres éléments de modélisations, et s'intéresse à un aspect précis du modèle. La combinaison des différents types de diagrammes UML offrent une vue complète des aspects statiques et dynamiques d'un système [3].

3.2.2 Processus

UML n'est qu'un langage de modélisation, ce n'est pas une méthode. En effet, UML ne propose pas une démarche de modélisation explicitant et encadrant toutes les étapes d'un projet, de la compréhension des besoins à la production du code de l'application. Une méthode se doit de définir une séquence d'étapes, partiellement ordonnées. Pour développer notre application, nous allons utiliser une démarche simple et générique qui se situe entre UP (Unified Process), qui constitue un cadre général très complet de processus de développement, et XP (extrême Programming) qui est une approche minimaliste à la mode centrée sur le code. Cette méthode est proposée par Roques (2002) dans son livre «Les cahiers du programmeur UML2 modéliser une application web» qui résulte de plusieurs années d'expérience sur de nombreux projets dans des domaines variés. Elle a donc montré son efficacité dans la pratique [3].

- Conduit par les cas d'utilisation, comme UP, mais beaucoup plus simple.
- relativement légère et restreinte, comme XP, mais sans négliger les activités de modélisation et analyse et conception.
- fondée sur l'utilisation d'un sous ensemble nécessaire et suffisant du langage UML.

Ci-après, nous citons brièvement les phases définies par cette démarche, et les diagrammes élaborés dans chaque phase (Figure3.1) :

Première phase : Identification des besoins

Diagramme de cas d'utilisation.

Diagramme de séquence système.

Maquette d'IHM.

Deuxième phase : Phase d'analyse

Modèle du domaine.

Diagramme de classes participantes.

Diagramme d'activité de navigation.

Troisième phase : Phase de conception

Diagramme d'interaction.

Diagramme de classes de conception.

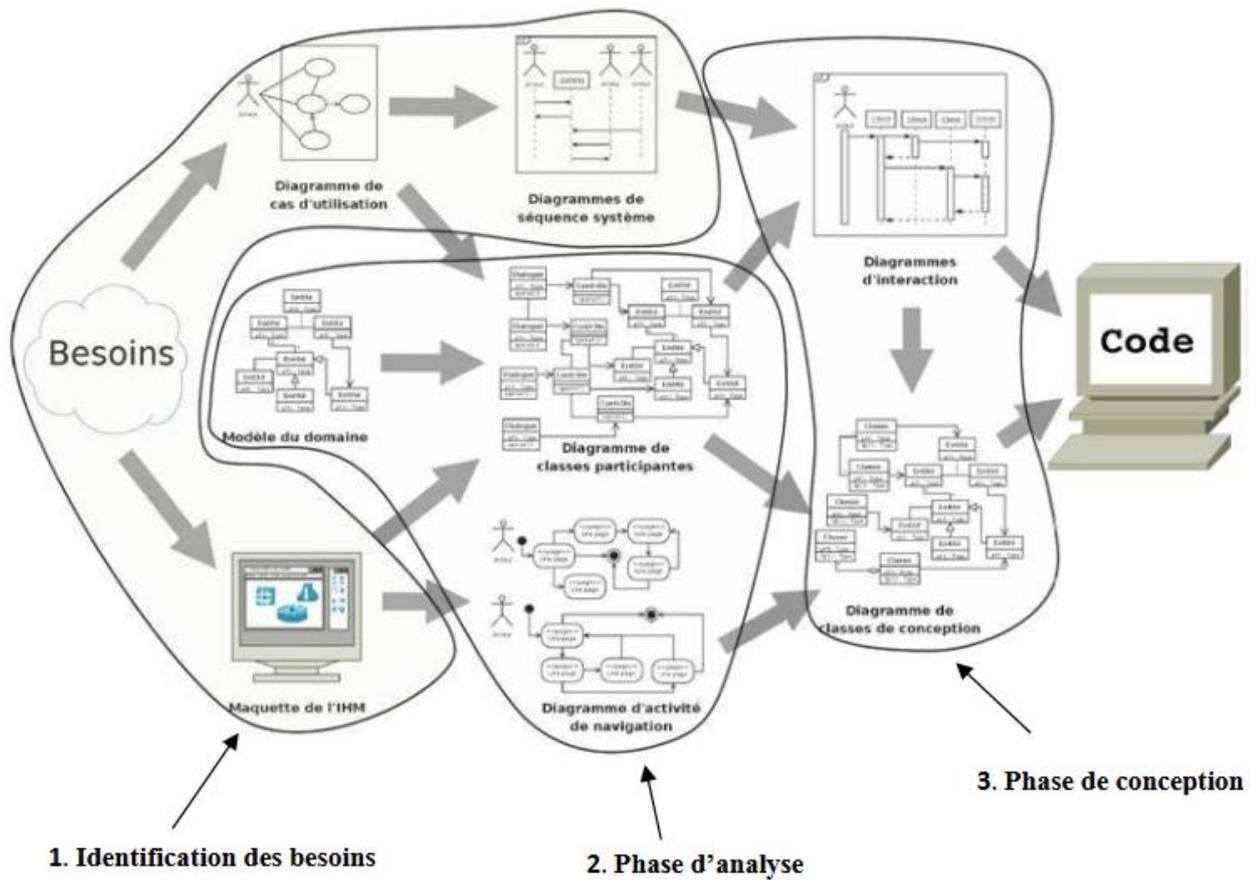


Figure 3.1: Chaîne complète de la démarche de modélisation du besoin jusqu'au code

3.3 Phase d'identification des besoins

3.3.1 Cahier des charges

3.3.1.1 Présentation de projet

L'objectif de ce travail est le développement d'une application de la gestion locative. L'application sera dotée de nombreuses fonctionnalités qui permettront aux utilisateurs de réaliser différentes tâches, entre autres, la gestion des biens immobiliers, la gestion des locataires, la gestion des paiements des loyers ainsi que la consultation des statistiques des recouvrements, etc.

3.3.1.2 Les besoins fonctionnels

Un premier tour d'horizon des besoins exprimés par les experts du domaine permis d'établir le cahier des charges préliminaire suivant :

a. S'authentifier

Un utilisateur possédant un compte dans la base de données peut s'authentifier avec son login et son mot de passe.

b. Gestion des comptes

L'application doit permettre la création d'un nouveau compte dans la base de données pour permettre aux utilisateurs de l'application d'accéder chacun à son espace, elle doit permettre également la consultation de la liste des comptes déjà créés dans la base de données, et puis l'utilisateur aura la possibilité de consulter ou de modifier les informations relatives à chacun des comptes (Nom utilisateur, mot de passe, type et permissions), comme il peut supprimer définitivement un compte déjà existant dans la base de données.

c. Gestion des biens immobiliers

L'application doit permettre la consultation de la liste des biens immobiliers déjà créés dans la base de données, et puis l'utilisateur aura la possibilité de consulter ou de modifier les informations relatives à chacun des biens immobiliers (id et type) ou encore de rechercher un bien. Il peut également ajouter un nouveau bien immobilier, Comme il peut supprimer définitivement un bien déjà existant dans la base de données.

d. Gestion des locataires

L'application doit permettre la consultation de la liste des locataires déjà créés dans la base de données, après, l'utilisateur aura la possibilité de consulter ou de modifier les informations relatives à chacun des locataires(id, nom, prénom, date de naissance, lieu de naissance et adresse) ou encore de rechercher un locataire, la recherche peut être effectuée suivants différents critères : par nom, par prénom ou par date de naissance . Il peut également ajouter un nouveau locataire, comme il peut supprimer définitivement un locataire déjà existant dans la base de données.

e. Gestion des loyers

L'application doit permettre la génération des loyers, la réception et la validation des paiements ainsi que la consultation des loyers, après cela l'utilisateur aura la possibilité d'établir une quittance, un avis d'échéance ou encore une mise en demeure.

f. Consultation du tableau de bord

L'application doit permettre la consultation des statistiques des biens immobiliers payés ou non.

3.3.1.3 Les besoins non fonctionnels (technique)

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

Les principaux besoins non fonctionnels de notre application se résument dans les points suivants :

a. La sécurité

Qui peut définir les niveaux d'accès possibles à l'application pour chacun des rôles métier, chaque acteur possède un nom d'utilisateur et un mot de passe, qui lui accorde des droits d'utilisation bien précis.

L'application web doit être sécurisée contre le piratage, particulièrement les injections SQL.

b. La rapidité de traitement

Vue le nombre important d'interaction quotidien, il est nécessaire que la durée d'exécution des traitements soit acceptable et s'approche le plus possible du temps réel.

c. La convivialité et simplicité d'utilisation

Cela signifie que l'application doit avoir une interface graphique claire, bien organisée et surtout facile à utiliser par un non informaticien.

3.3.1.4 Les choix techniques

Les choix technique que nous avons adoptés sont l'utilisation de :

- Langage de modélisation UML (Unified Modeling Language).
- Serveur web apache.
- Serveur de base de données MySQL.
- Les langages HTML, CSS et Bootstrap pour la structuration du site.
- Le langage PHP pour l'implémentation des fonctions de l'application web.
- L'outil PhpMyAdmin pour l'intégration des bases MySQL.

3.3.2 Identification des acteurs

Un acteur est l'idéalisation d'un rôle joué par une entité externe, (une personne ou un autre système...) qui interagit avec un système.

Notre application possède quatre acteurs humains interagissant avec le système, qui sont:

- **Web master**

C'est l'acteur qui s'occupe de la gestion des comptes des utilisateurs de l'application (la création d'un nouveau compte, la consultation, la modification ou la suppression d'un compte existant) ainsi que la gestion des autorisations pour chacun d'eux.

- **Agent de maitrise**

C'est l'acteur chargé de la gestion des biens immobiliers et celle des locataires, c'est-à-dire l'ajout d'un nouveau bien immobilier ou locataire, la consultation, la recherche, la modification ou encore la suppression des biens immobiliers ou des locataires déjà existants.

- **Caissier**

C'est l'acteur chargé de la génération des loyers mensuels, la réception des paiements, la consultation des loyers ainsi que l'établissement des quittances, des avis d'échéances et celle des mises en demeure.

- **Cadre dirigeant**

C'est l'acteur chargé de la consultation des paiements des loyers ainsi que la consultation du tableau de bord.

3.3.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un des diagrammes UML. Il représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système.

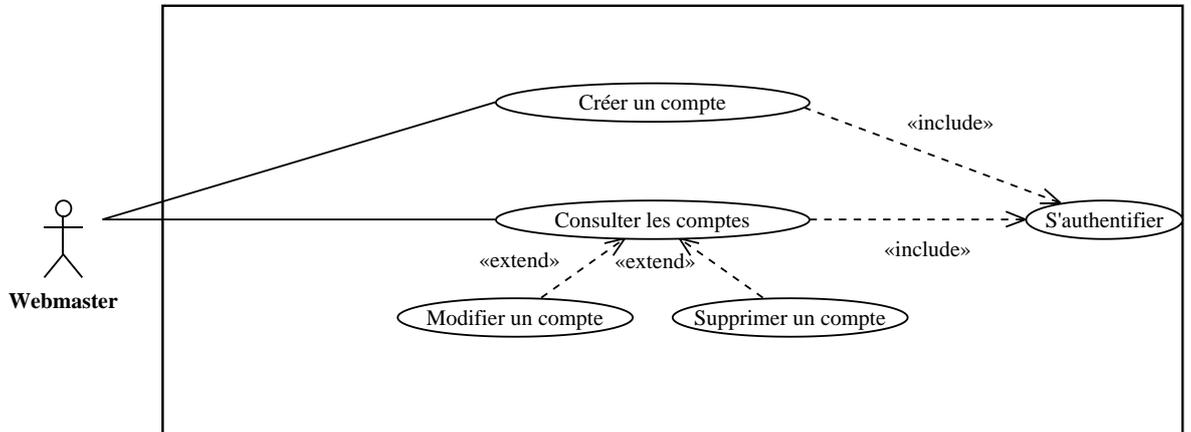


Figure 3.2: Diagramme des cas d'utilisation de l'acteur : «Web master»

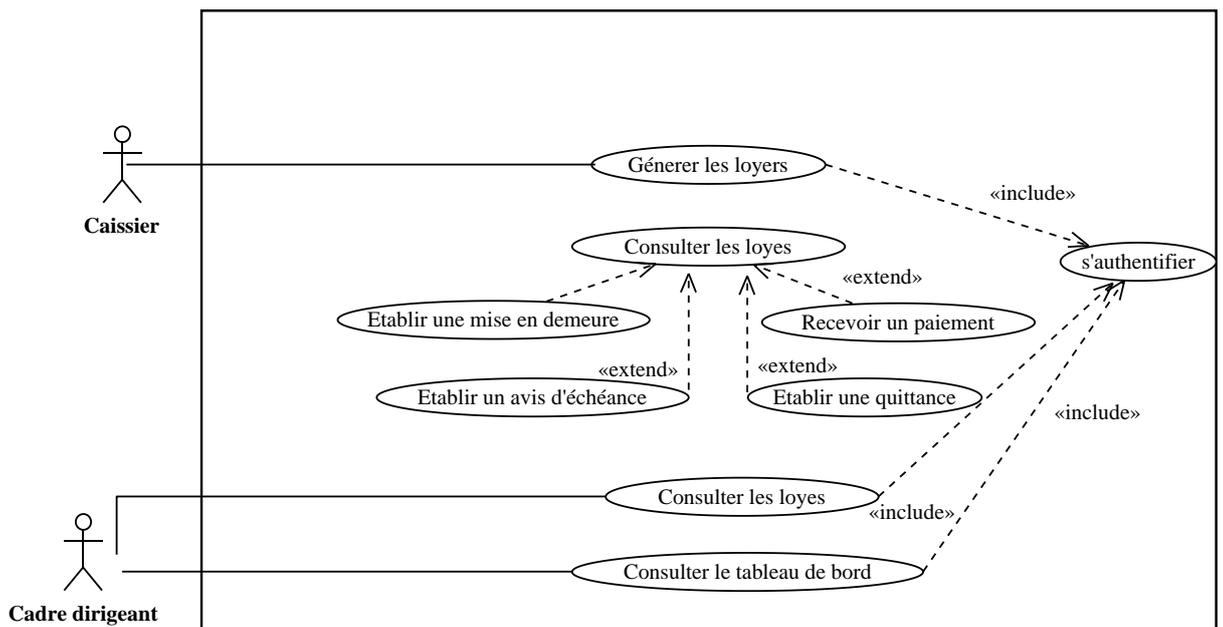


Figure 3.3: Diagramme des cas d'utilisation de l'acteur «Caissier et Cadre dirigeant»

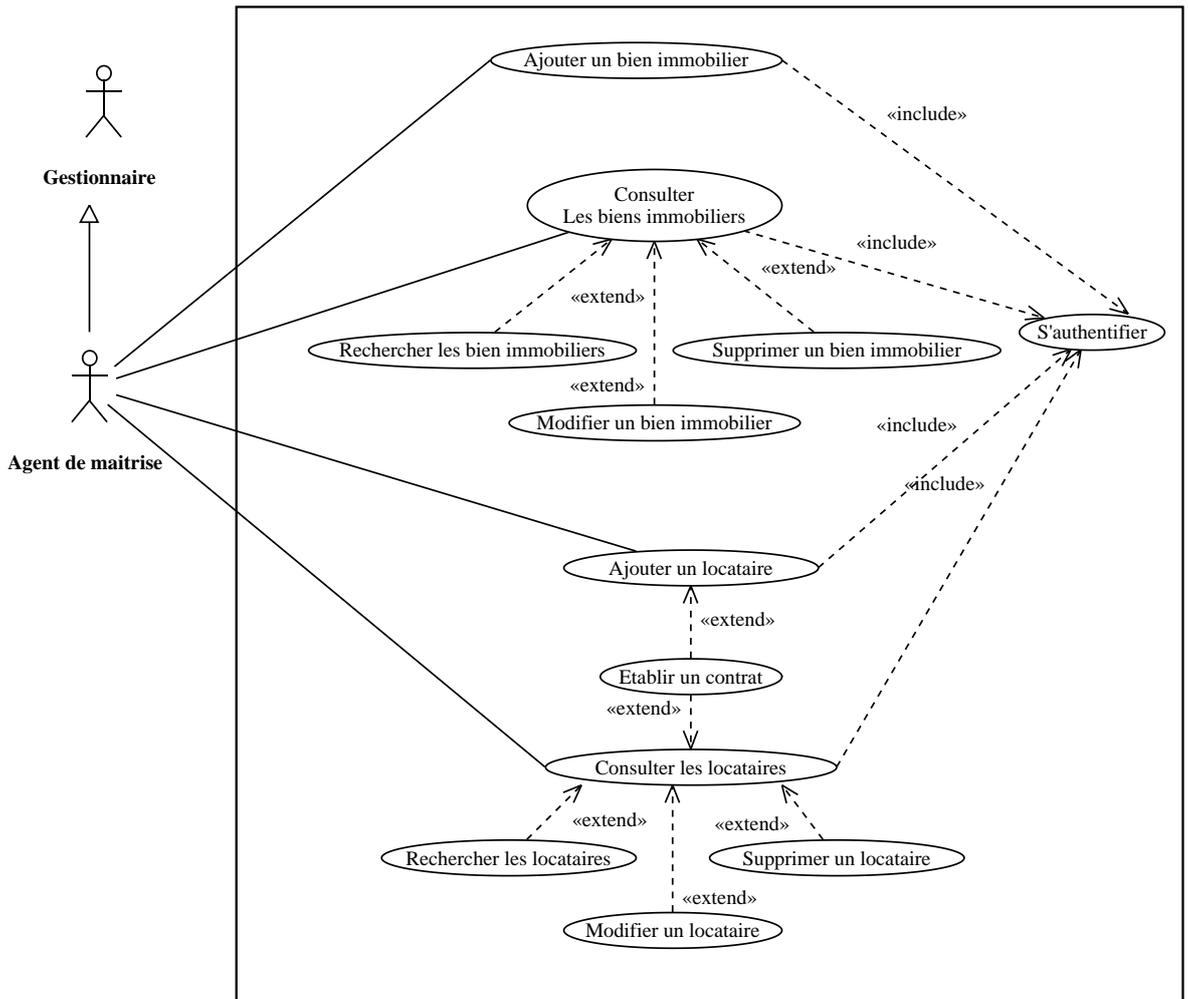


Figure 3.4: Diagramme des cas d'utilisation de l'acteur «Agent de maitrise»

3.3.4 Classement des cas d'utilisation

Après avoir identifié les cas d'utilisation, nous allons les classer en tenant compte de deux facteurs: (1) La priorité fonctionnelle, et (2) le risque technique [3]. Le classement des cas d'utilisation nous permet de découper notre travail à des itérations (Tableau 3.1).

Cas d'utilisation	priorité	Risque	itérations
S'authentifier	Elevé	Faible	1
Créer un compte	Elevé	Faible	2
Consulter les comptes	Moyenne	Faible	17
Modifier un compte	Moyenne	Faible	18
Supprimer un compte	Moyenne	Faible	19
Ajoute un bien immobilier	Elevé	Faible	3
Consulter les biens immobiliers	Moyenne	Faible	4
Rechercher les biens immobiliers	Faible	Faible	20
Modifier les biens immobiliers	Moyenne	Faible	21
Supprimer les biens immobiliers	Moyenne	Faible	22

Ajouter un locataire	Elevé	Faible	5
Etablir un contrat de location	Elevé	Faible	6
Consulter les locataires	Elevé	Faible	7
Rechercher les locataires	Faible	Faible	23
Modifier un locataire	Elevé	Faible	8
Supprimer un locataire	Elevé	Faible	9
Générer les loyers	Elevé	Faible	10
Consulter les paiements des loyers	Elevé	Faible	11
Recevoir un paiement	Elevé	Faible	12
Etablir une quittance	Elevé	Faible	13
Etablir un avis d'échéance	Elevé	Faible	14
Etablir une mise en demeure	Elevé	Faible	15
Consulter le tableau de bord	Moyenne	Moyen	16

Tableau 3.1: Classement des cas d'utilisation

3.3.5 Description textuelles des cas d'utilisation

3.3.5.1 S'authentifier

Cas d'utilisation	S'authentifier.
Acteur	Gestionnaire.
Objectif	Permettre à chaque gestionnaire d'accéder à son compte.
Pré condition	/
Post-condition	Le gestionnaire est authentifié par le système.
Scenario nominal	<ol style="list-style-type: none"> 1. Le gestionnaire demande l'accès à son compte. 2. Le système lui affiche le formulaire d'authentification. 3. Le gestionnaire transmet le login et le mot de passe. 4. Le système ouvre la page d'accueil.
Scenario alternatif	<ol style="list-style-type: none"> 3.1 Le login et/ou le mot de passe sont incorrects. <ol style="list-style-type: none"> 3.1.1 Le système informe le gestionnaire que le login et/ou le mot de passe sont incorrectes. 3.1.2 Le système propose de les renseigner de nouveau. 3.1.3 Reprise de l'enchaînement du scenario nominal à partir du deuxième point(2).

3.3.5.2 Créer un compte

Cas d'utilisation	Créer un compte.
Acteur	Web master.
Objectif	Permettre au web master de créer un compte.
Pré condition	Le web master est authentifié par le système.
Post-condition	Le compte est créé.
Scenario nominal	<ol style="list-style-type: none"> 1. Le web master demande la création d'un compte. 2. Le système lui affiche un formulaire de création du compte. 3. Le web master transmet les informations. 4. Le système confirme la création du compte.
Scenario alternatif	<ol style="list-style-type: none"> 3.1 Les champs sont incorrects ou vides. <ol style="list-style-type: none"> 3.1.1 Le système informe le web master que les champs sont incorrects ou vides. 3.1.2 Le système propose de les renseigner de nouveau. 3.1.3 Reprise de l'enchaînement du scenario nominal à partir du deuxième point(2).

3.3.5.3 Consulter les comptes

Cas d'utilisation	Consulter les comptes.
Acteur	Web master.
Objectif	Permettre au web master de consulter la liste des comptes.
Pré condition	Le web master est authentifié par le système.
Post-condition	Le système affiche la liste des comptes.
Scenario nominal	<ol style="list-style-type: none"> 1. Le web master demande la consultation des comptes. 2. Le système affiche la liste des comptes.
Scenario alternatif 1	<ol style="list-style-type: none"> 2.1. Le web master demande la modification des informations d'un compte. <ol style="list-style-type: none"> 2.1.1. Le système fait référence au cas d'utilisation « Modifier un compte ».
Scenario alternatif 2	<ol style="list-style-type: none"> 2.2. Le web master demande la suppression d'un compte. <ol style="list-style-type: none"> 2.2.1. Le système fait référence au cas d'utilisation « Supprimer un compte ».

3.3.5.4 Modifier un compte

Cas d'utilisation	Modifier un compte.
Acteur	Web master.
Objectif	Permettre au web master de modifier les informations d'un compte.
Pré condition	Le web master est authentifié par le système.
Post-condition	Le compte est modifié.
Scenario nominal	<ol style="list-style-type: none"> 1. Le web master demande la modification d'un compte. 2. Le système lui affiche le formulaire qui contient les informations du compte. 3. Le web master transmet les modifications au système. 4. Le système confirme la modification.
Scenario alternatif	<ol style="list-style-type: none"> 3.1 Les champs sont incorrects ou vides. <ol style="list-style-type: none"> 3.1.1 Le système informe Le web master que les champs sont incorrects ou vides. 3.1.2 Le système propose de les renseigner de nouveau. 3.1.3 Reprise de l'enchaînement du scenario nominal à partir du deuxième point(2).

3.3.5.5 Supprimer un compte

Cas d'utilisation	Supprimer un compte.
Acteur	Web master.
Objectif	Permettre au web master de supprimer un compte.
Pré condition	Le web master est authentifié par le système.
Post-condition	Le compte est supprimé.
Scenario nominal	<ol style="list-style-type: none"> 1. Le web master demande la suppression d'un compte. 2. Le système lui demande la validation de la suppression. 3. Le web master valide la suppression. 4. Le système confirme la suppression du compte.
Scenario d'erreur	<ol style="list-style-type: none"> 3.1. Le web master annule la suppression du compte.

3.3.5.6 Ajouter un bien immobilier

Cas d'utilisation	Ajouter un bien immobilier.
Acteur	Agent de maitrise.
Objectif	Permettre à l'agent de maitrise d'ajouter un bien immobilier.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	Le bien immobilier est ajouté.
Scenario nominal	<ol style="list-style-type: none"> 1. L'agent de maitrise demande l'ajout d'un bien immobilier. 2. Le système lui affiche un formulaire d'ajout d'un bien immobilier. 3. L'agent de maitrise transmet les informations. 4. Le système confirme l'ajout du bien immobilier.
Scenario alternatif	<ol style="list-style-type: none"> 3.1 Les champs sont incorrects ou vides. <ol style="list-style-type: none"> 3.1.1 Le système informe l'agent de maitrise que les champs sont incorrects ou vides. 3.1.2 Le système propose de les renseigner de nouveau. 3.1.3 Reprise de l'enchaînement du scenario nominal à partir du deuxième point(2).

3.3.5.7 Consulter les biens immobiliers

Cas d'utilisation	Consulter les biens immobiliers.
Acteur	Agent de maitrise.
Objectif	Permettre à l'agent de maitrise de consulter les biens immobiliers.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	La liste des biens immobiliers est affichée.
Scenario nominal	<ol style="list-style-type: none"> 1. L'agent de maitrise demande la consultation des biens immobiliers. 2. Le système affiche la liste des biens immobiliers.
Scenario alternatif 1	<ol style="list-style-type: none"> 2.1. L'agent de maitrise demande la recherche des biens immobiliers. <ol style="list-style-type: none"> 2.1.1. Le système fait référence au cas d'utilisation « Rechercher les biens immobiliers ».
Scenario alternatif 2	<ol style="list-style-type: none"> 2.2. L'agent de maitrise demande la modification des informations d'un bien immobilier. <ol style="list-style-type: none"> 2.2.1. Le système fait référence au cas d'utilisation « Modifier un bien immobilier ».
Scenario alternatif 3	<ol style="list-style-type: none"> 2.3. L'agent de maitrise demande la suppression d'un bien immobilier. <ol style="list-style-type: none"> 2.3.1. Le système fait référence au cas d'utilisation « Supprimer un bien immobilier ».

3.3.5.8 Rechercher les biens immobiliers

Cas d'utilisation	Rechercher les biens immobiliers.
Acteur	Agent de maitrise.
Objectif	Permet à l'agent de maitrise de rechercher les biens immobiliers.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	Le résultat de recherche est affiché.
Scenario nominal	1. Le web master transmet la requête rechercher les biens immobiliers. 2. Le système affiche le résultat de la recherche.

3.3.5.9 Modifier un bien immobilier

Cas d'utilisation	Modifier un bien immobilier.
Acteur	Agent de maitrise.
Objectif	Permettre à l'agent de maitrise de modifier les informations d'un bien immobilier.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	Les modifications sont enregistrées.
Scenario nominal	1. L'agent de maitrise demande la modification des informations d'un bien immobilier. 2. Le système lui affiche le formulaire qui contient les informations du bien immobilier. 3. L'agent de maitrise transmet les informations. 4. Le système confirme la modification.
Scenario alternatif	3.1 Les champs sont incorrects ou vides. 3.1.1 Le système informe l'agent de maitrise que les champs sont incorrects ou vides. 3.1.2 Le système propose de les renseigner de nouveau. 3.1.3 Reprise de l'enchainement du scenario nominal à partir du deuxième point(2).

3.3.5.10 Supprimer un bien immobilier

Cas d'utilisation	Supprimer un bien immobilier.
Acteur	Agent de maitrise.
Objectif	Permettre à l'agent de maitrise de supprimer un bien immobilier.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	Le bien immobilier est supprimé.
Scenario nominal	<ol style="list-style-type: none"> 1. L'agent de maitrise demande la suppression d'un bien immobilier. 2. Le système demande la validation de la suppression. 3. L'agent de maitrise valide la suppression. 4. Le système confirme la suppression du bien immobilier.
Scenario d'erreur	3.1. L'agent de maitrise annule la suppression.

3.3.5.11 Ajouter un locataire

Cas d'utilisation	Ajouter un locataire.
Acteur	Agent de maitrise.
Objectif	Permettre à l'agent de maitrise d'ajouter un locataire.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	Le locataire est ajouté.
Scenario nominal	<ol style="list-style-type: none"> 1. L'agent de maitrise demande l'ajout d'un locataire. 2. Le système lui affiche un formulaire d'ajout du locataire. 3. L'agent de maitrise transmet les informations du locataire au système. 4. Le système confirme l'ajout du locataire.
Scenario alternatif 1	<ol style="list-style-type: none"> 3.1 Les champs sont incorrects ou vides. <ol style="list-style-type: none"> 3.1.1 Le système informe l'agent de maitrise que les champs sont incorrects ou vides. 3.1.2 Le système propose de les renseigner de nouveau. 3.1.3 Reprise de l'enchainement du scenario nominal à partir du deuxième point (2).
Scenario alternatif 2	<ol style="list-style-type: none"> 4.1 L'agent de maitrise demande l'établissement d'un contrat. <ol style="list-style-type: none"> 4.2.1 Le système fait référence au cas d'utilisation « Établir un contrat de location ».

3.3.5.12 Établir un contrat de location

Cas d'utilisation	Établir un contrat de location.
Acteur	Agent de maîtrise.
Objectif	Permettre à l'agent de maîtrise d'établir un contrat.
Pré condition	Le caissier est authentifié par le système.
Post-condition	Le contrat est établi par le système.
Scenario nominal	1. L'agent de maîtrise demande l'établissement d'un contrat. 2. Le système affiche le formulaire du contrat.
Scenario alternatif 1	2.1 L'agent de maîtrise demande l'impression d'un contrat. 2.1.1 Le système imprime le contrat.

3.3.5.13 Consulter les locataires

Cas d'utilisation	Consulter les locataires.
Acteur	Agent de maîtrise.
Objectif	Permet à l'agent de maîtrise de consulter les locataires.
Pré condition	L'agent de maîtrise est authentifié par le système.
Post-condition	La liste des locataires est affichée.
Scenario nominal	1. L'agent de maîtrise demande la consultation des locataires. 2. Le système affiche la liste des locataires.
Scenario alternatif 1	2.1. L'agent de maîtrise demande la recherche des locataires. 2.1.1. Le système fait référence au cas d'utilisation « Rechercher les locataires ».
Scenario alternatif 2	2.2. L'agent de maîtrise demande la modification des informations d'un locataire. 2.2.1. Le système fait référence au cas d'utilisation « Modifier un locataire ».
Scenario alternatif 3	2.3. L'agent de maîtrise demande la suppression d'un locataire. 2.3.1. Le système fait référence au cas d'utilisation « Supprimer un locataire ».

3.3.5.14 Rechercher les locataires

Cas d'utilisation	Rechercher les locataires.
Acteur	Agent de maitrise.
Objectif	Permet à l'agent de maitrise de rechercher les locataires.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	Le résultat est affiché.
Scenario nominal	1. L'agent de maitrise transmet la requête rechercher les locataires. 2. Le système affiche le résultat de la recherche.

3.3.5.15 Modifier un locataire

Cas d'utilisation	Modifier un locataire.
Acteur	Agent de maitrise.
Objectif	Permet à l'agent de maitrise de modifier les informations d'un locataire.
Pré condition	L'agent de maitrise est authentifié par le système.
Post-condition	Les informations du locataire sont modifiées.
Scenario nominal	1. L'agent de maitrise demande la modification des informations du locataire. 2. Le système lui affiche le formulaire de modification. 3. L'agent de maitrise transmet les modifications. 4. Le système confirme la modification.
Scenario alternatif	3.1 Les champs sont incorrects ou vides. 3.1.1 Le système informe l'agent de maitrise que les champs sont incorrects ou vides. 3.1.2 Le système lui propose de les renseigner de nouveau. 3.1.3 Reprise de l'enchaînement du scenario nominal à partir du deuxième point(2).

3.3.5.16 Supprimer un locataire

Cas d'utilisation	Supprimer un locataire.
Acteur	Agent de maîtrise.
Objectif	Permettre à l'agent de maîtrise de supprimer un locataire.
Pré condition	L'agent de maîtrise est authentifié par le système.
Post-condition	Le locataire est supprimé.
Scenario nominal	<ol style="list-style-type: none"> 1. L'agent de maîtrise demande la suppression d'un locataire. 2. Le système demande la validation de la suppression. 3. L'agent de maîtrise valide la suppression. 4. Le système confirme la suppression du locataire.
Scenario d'erreur	3.1. L'agent de maîtrise annule la suppression.

3.3.5.17 Générer les loyers

Cas d'utilisation	Générer loyers
Acteur	Caissier.
Objectif	Permettre à l'acteur de générer les loyers mensuels d'une date d'échéance.
Pré condition	L'acteur est authentifié par le système.
Post-condition	Les loyers mensuels sont générés
Scenario nominal	<ol style="list-style-type: none"> 1. Le caissier demande de générer le loyer 2. Le système générer loyer.
Scenario d'erreur	<ol style="list-style-type: none"> 2.1. Le système a déjà généré les loyers mensuels de l'échéance. <ol style="list-style-type: none"> 2.1.1. Le système informe l'acteur que le loyer a été déjà généré.

3.3.5.18 Consulter les loyers

Cas d'utilisation	Consulter les loyers
Acteur	Gestionnaire paiements
Objectif	Permettre au caissier de consulter les loyers.
Pré condition	Le caissier est authentifié par le système.
Post-condition	Les paiements de loyers sont affichés
Scenario nominal	1. Le caissier demande la consultation des loyers. 2. Le système lui affiche les loyers et leurs états (payé, en attente, en retard).
Scenario alternatif 1	1.1. Le caissier demande la consultation des loyers en attente 2.2.1. Le système affiche les loyers en attente.
Scenario alternatif 2	2.2. Le caissier demande la consultation des loyers en retard 2.2.1. Le système affiche les loyers en retard.
Scenario alternatif 3	2.3. Le caissier demande la consultation des loyers payés 2.3.1. Le système affiche les loyers payés.
Scenario alternatif 4	2.4. Le caissier demande la consultation des loyers d'un locataire. 2.4.1. Le système affiche les loyers du locataire.
Scenario alternatif 5	2.5. Le caissier demande l'établissement d'une quittance 2.5.1. Le système fait référence au cas d'utilisation « Établir un avis d'échéance ».
Scenario alternatif 6	2.6. Le caissier demande l'établissement d'une mise en demeure 2.6.1. Le système fait référence au cas d'utilisation « Établir une mise en demeure ».
Scenario alternatif 7	2.7. Le caissier demande la réception d'un paiement. 2.7.1. Le système fait référence au cas d'utilisation « Recevoir paiement ».

3.3.5.19 Recevoir un paiement

Cas d'utilisation	Recevoir un paiement.
Acteur	Caissier.
Objectif	Permettre au caissier de sauvegarder un paiement de loyer par un locataire.
Pré condition	Le caissier est authentifié par le système.
Post-condition	Le paiement du loyer est sauvegardé.
Scenario nominal	1. Le caissier demande la réception d'un paiement. 2. Le système affiche le formulaire de paiement. 3. Le caissier transmet les informations du paiement. 4. Le système confirme la réception du paiement.
Scenario alternatif	4.1. Le caissier demande l'impression d'une quittance de paiement. 4.1.1. Le système fait appel au cas d'utilisation «Établir une quittance».

3.3.5.20 Établir une quittance

Cas d'utilisation	Établir une quittance.
Acteur	Caissier.
Objectif	Permettre au caissier d'établir une quittance.
Pré condition	Le caissier est authentifié par le système.
Post-condition	La quittance est établit par le système.
Scenario nominal	1. Le caissier demande l'établissement d'une quittance 2. Le système lui affiche le formulaire de la quittance.
Scenario alternatif 1	2.1 L'agent de maitrise demande l'impression d'une quittance. 2.1.1 Le système imprime la quittance.

3.3.5.21 Établir un avis d'échéance

Cas d'utilisation	Établir un avis d'échéance.
Acteur	Caissier.
Objectif	Permettre au caissier d'établir un avis d'échéance.
Pré condition	Le caissier est authentifié par le système.
Post-condition	L'avis d'échéance est établit par le système.
Scenario nominal	1. Le caissier demande l'établissement d'un avis d'échéance. 2. Le système lui affiche l'avis d'échéance.
Scenario alternatif 1	2.1 L'agent de maitrise demande l'impression d'un avis d'échéance. 2.1.1 Le système imprime l'avis d'échéance.

3.3.5.22 Établir une mise en demeure

Cas d'utilisation	Établir une mise en demeure.
Acteur	Caissier.
Objectif	Permettre au caissier d'établir une mise en demeure.
Pré condition	Le caissier est authentifié par le système.
Post-condition	La mise en demeure est établit par le système.
Scenario nominal	1. Le caissier demande l'impression d'une mise en demeure. 2. Le système lui affiche la mise en demeure.
Scenario alternatif 1	2.1 L'agent de maitrise demande l'impression d'une mise en demeure. 2.1.1 Le système imprime la mise en demeure.

3.3.5.23 Consulter le tableau de bord

Cas d'utilisation	Consulter le tableau de bord
Acteur	Cadre dirigeant.
Objectif	Permettre au cadre dirigeant de consulter le tableau de bord.
Pré condition	Le cadre dirigeant est authentifié par le système.
Post-condition	Le tableau de bord est affiché par le système.
Scenario nominal	1. Le cadre dirigeant demande la consultation du tableau de bord. 2. Le système affiche le tableau de bord

3.3.6 Diagramme de séquence système

3.3.6.1 S'authentifier

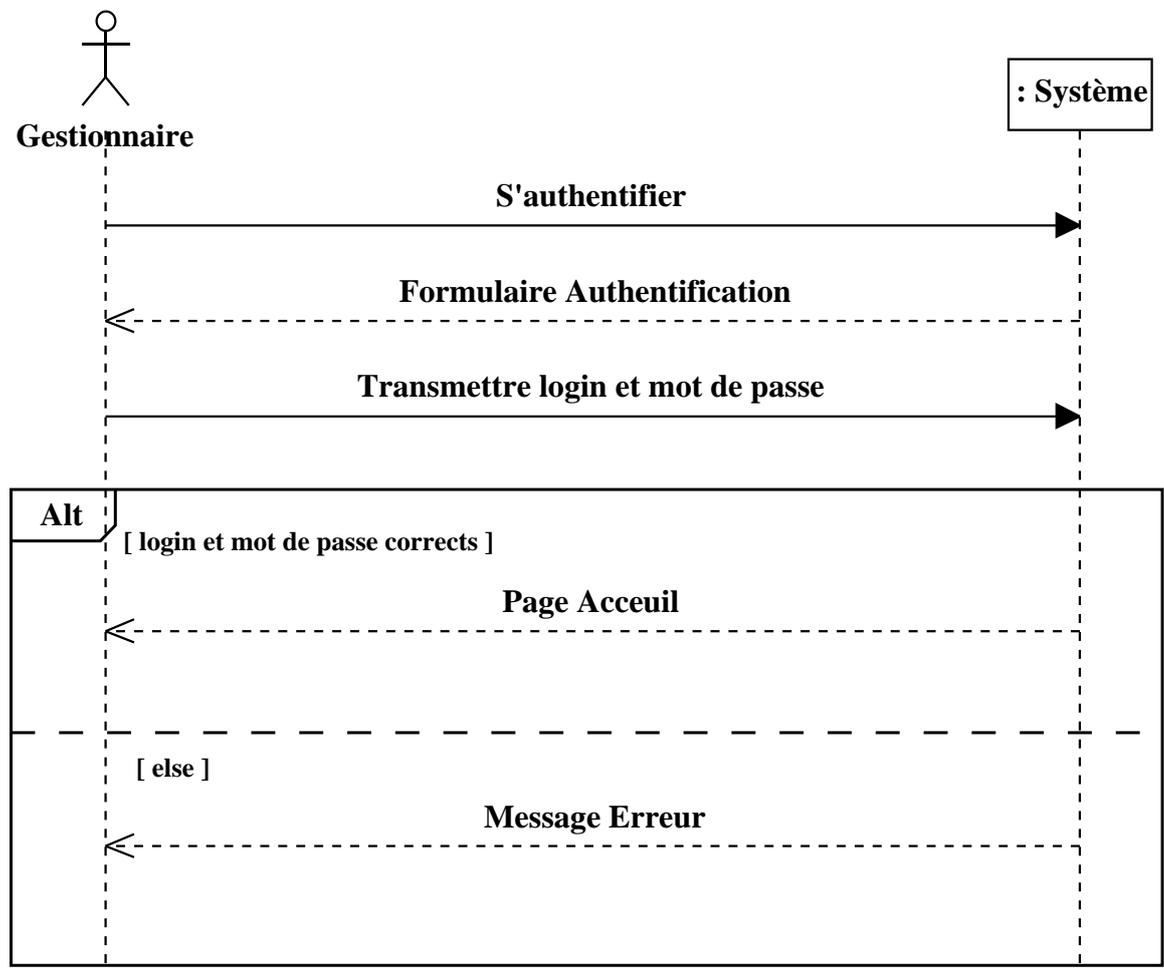


Figure 3.5: Diagramme de séquence système du cas d'utilisation «s'authentifier»

3.3.6.2 Créer un compte

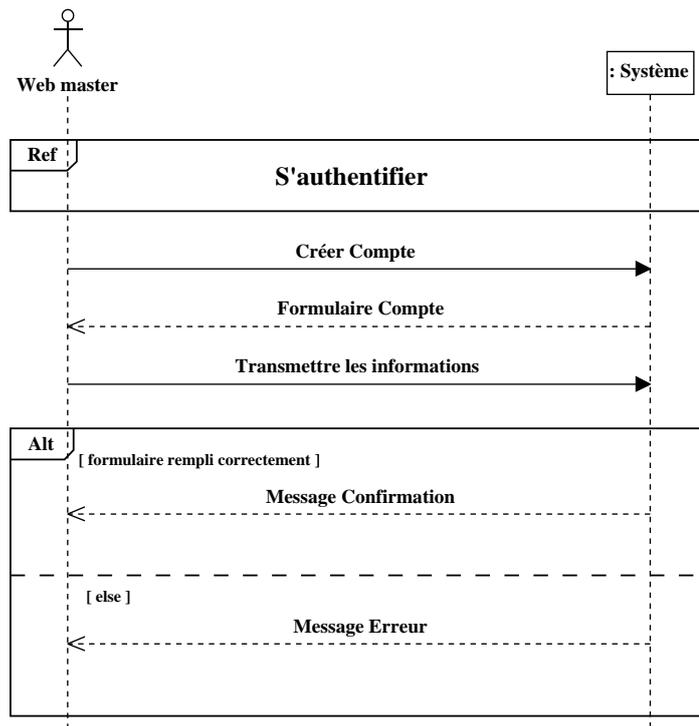


Figure 3.6: Diagramme de séquence système du cas d'utilisation «créer un compte»

3.3.6.3 Consulter les comptes

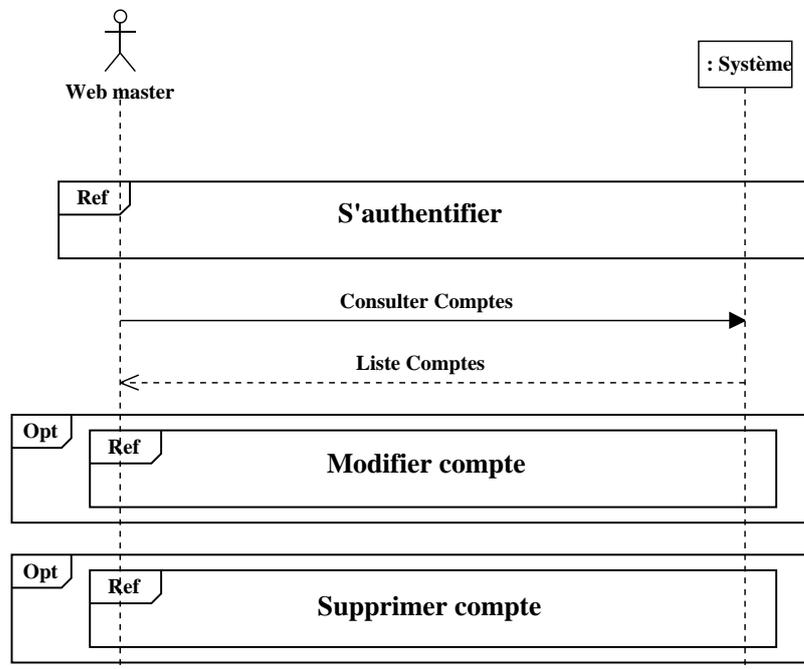


Figure 3.7: Diagramme de séquence système du cas d'utilisation «consulter les comptes»

3.3.6.4 Modifier un compte

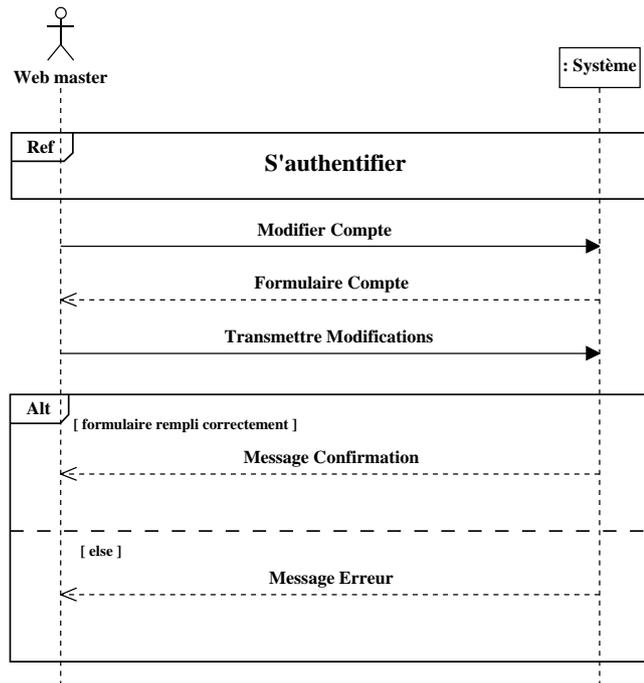


Figure 3.8: Diagramme de séquence système du cas d'utilisation «modifier un compte»

3.3.6.5 Supprimer un compte

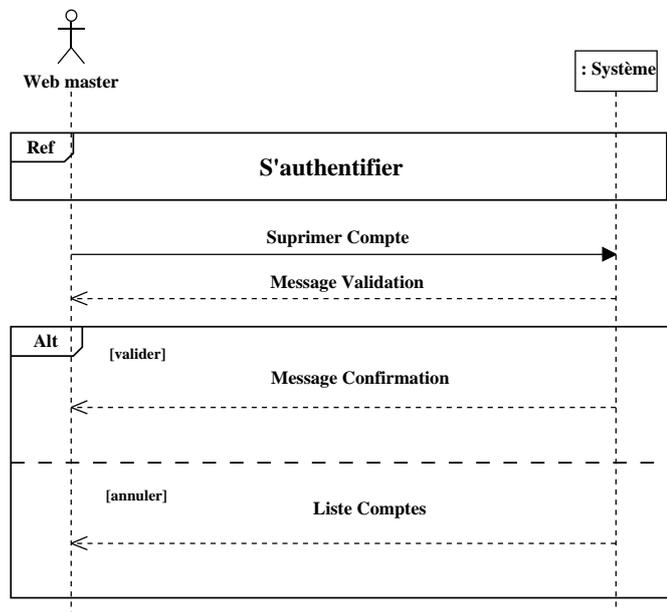


Figure 3.9: Diagramme de séquence système du cas d'utilisation «supprimer un compte»

3.3.6.6 Ajouter un bien immobilier

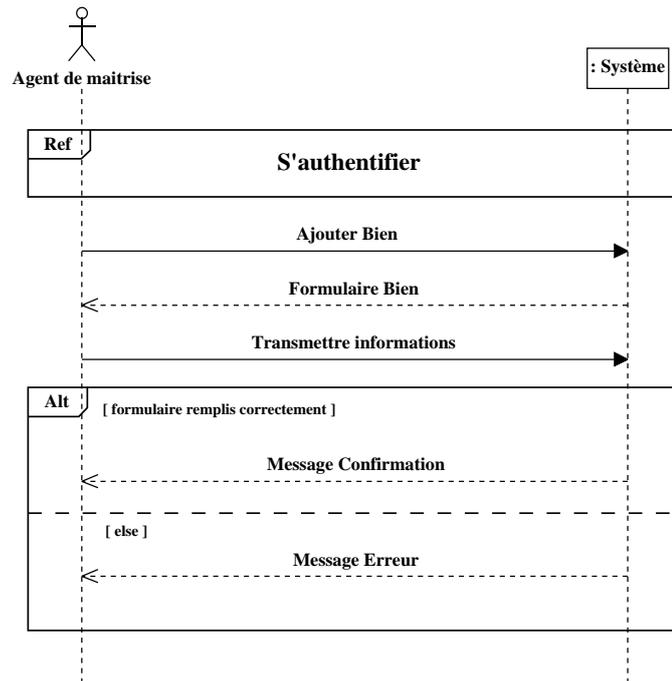


Figure 3.10: Diagramme de séquence système du cas d'utilisation «ajouter un bien immobilier»

3.3.6.7 Consulter les biens immobiliers

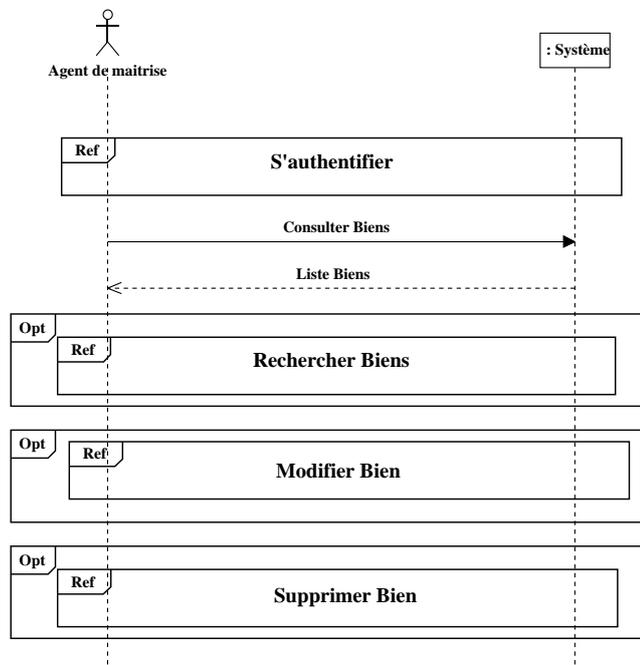


Figure 3.11: Diagramme de séquence système du cas d'utilisation «consulter les biens immobiliers»

3.3.6.8 Rechercher les biens immobiliers

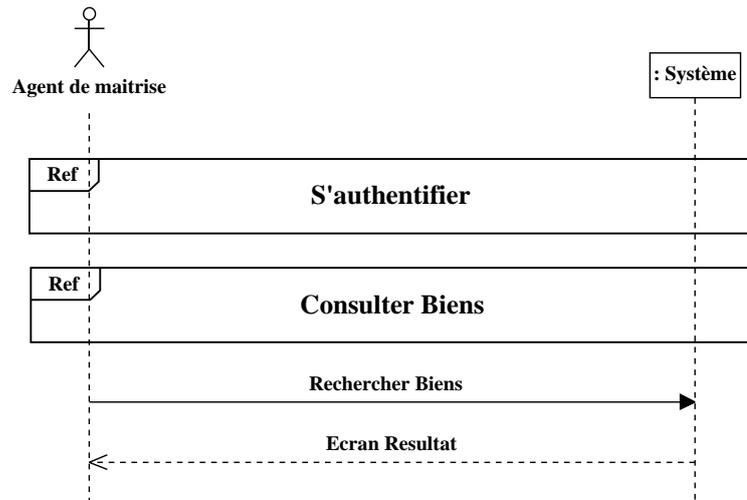


Figure 3.12: Diagramme de séquence système du cas d'utilisation «rechercher les biens immobiliers»

3.3.6.9 Modifier un bien immobilier

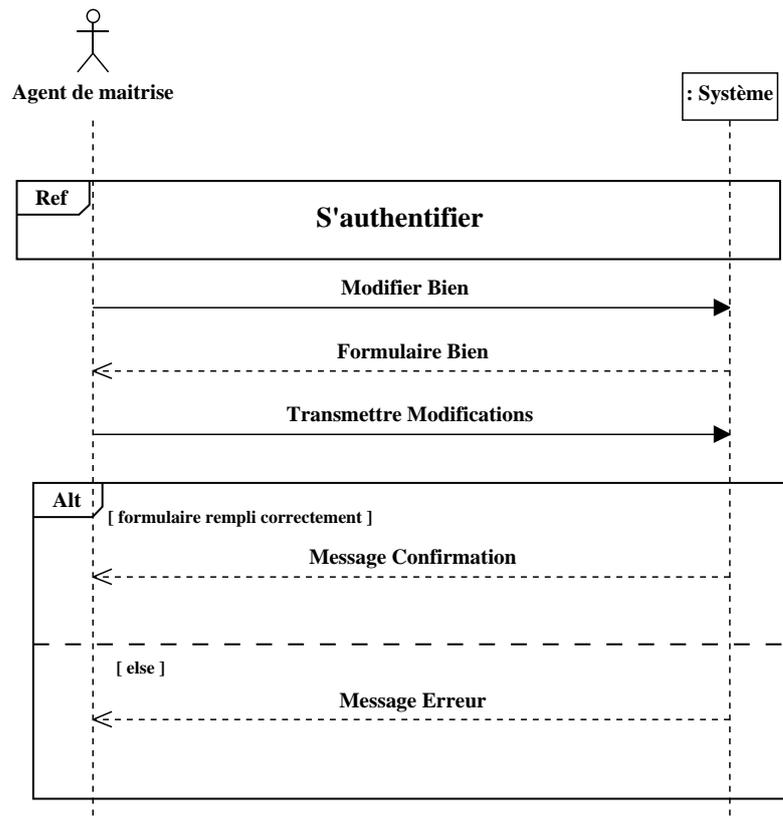


Figure 3.13: Diagramme de séquence système du cas d'utilisation «modifier un bien immobilier»

3.3.6.10 Supprimer un bien immobilier

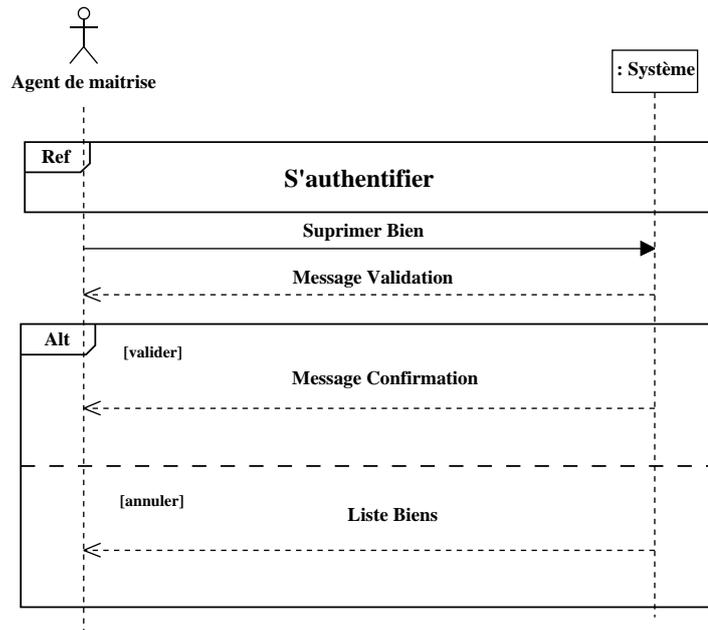


Figure 3.14: Diagramme de séquence système du cas d'utilisation «supprimer un bien immobilier»

3.3.6.11 Ajouter un locataire

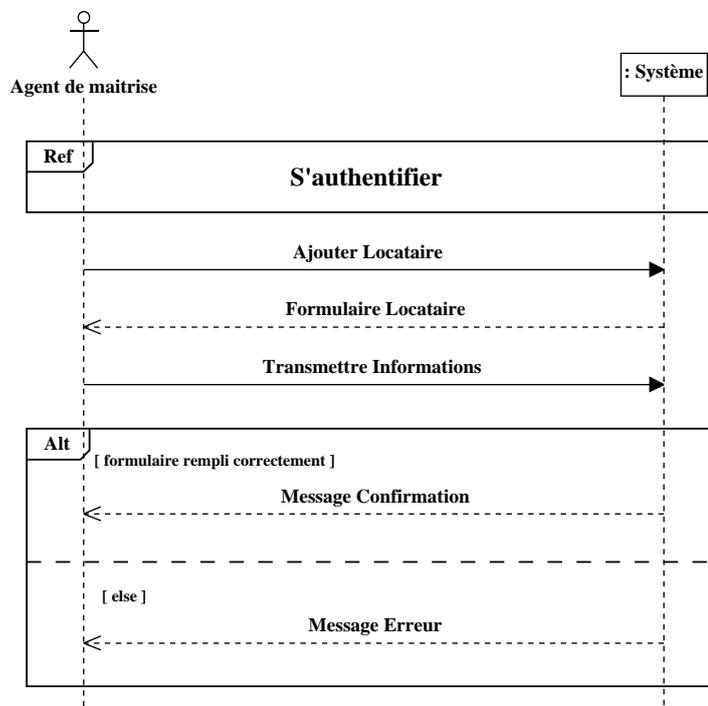


Figure 3.15: Diagramme de séquence système du cas d'utilisation «ajouter un locataire»

3.3.6.12 Établir un contrat de location

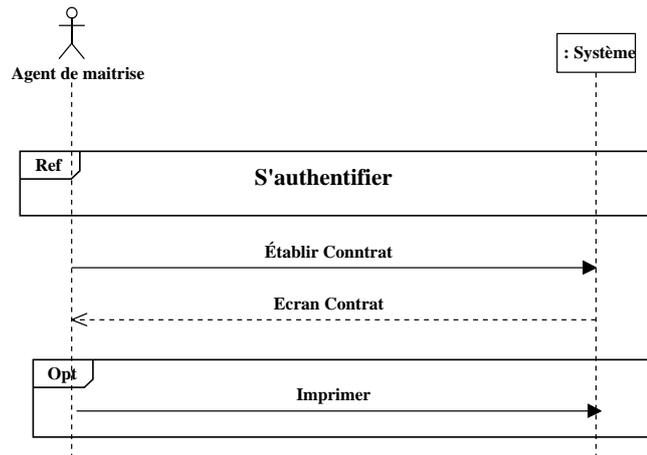


Figure 3.16: Diagramme de séquence système du cas d'utilisation «établir un contrat de location»

3.3.6.13 Consulter les locataires

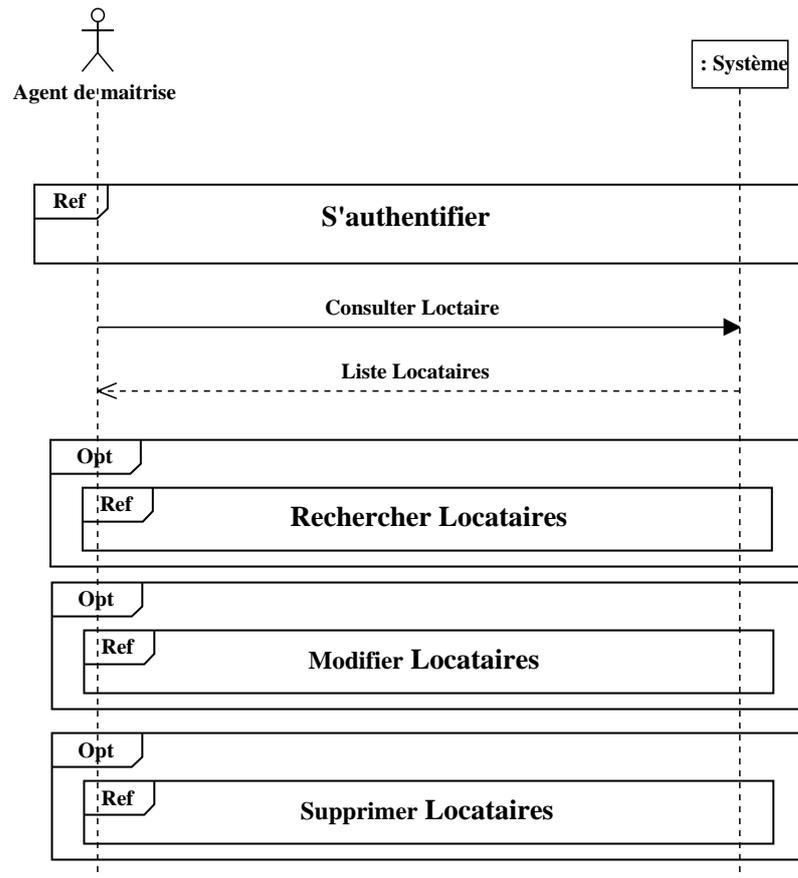


Figure 3.17: Diagramme de séquence système du cas d'utilisation «consulter les locataires»

3.3.6.14 Rechercher les locataires

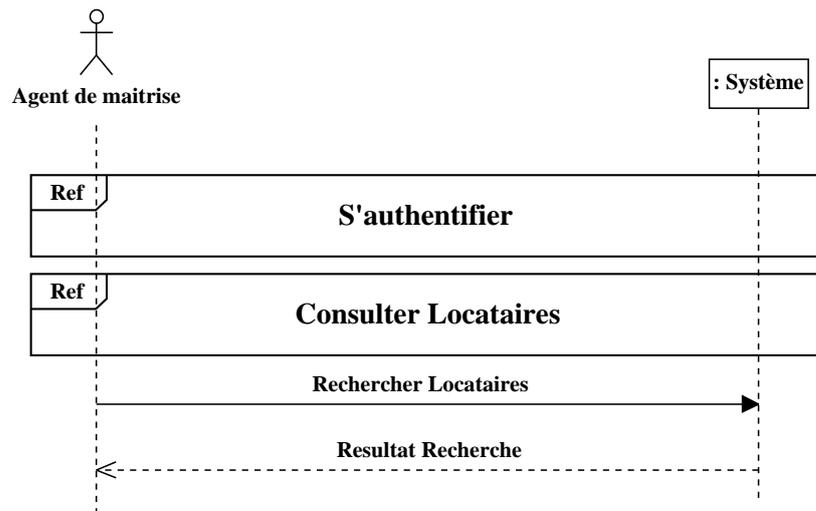


Figure 3.18: Diagramme de séquence système du cas d'utilisation «rechercher les locataires»

3.3.6.15 Modifier un locataire

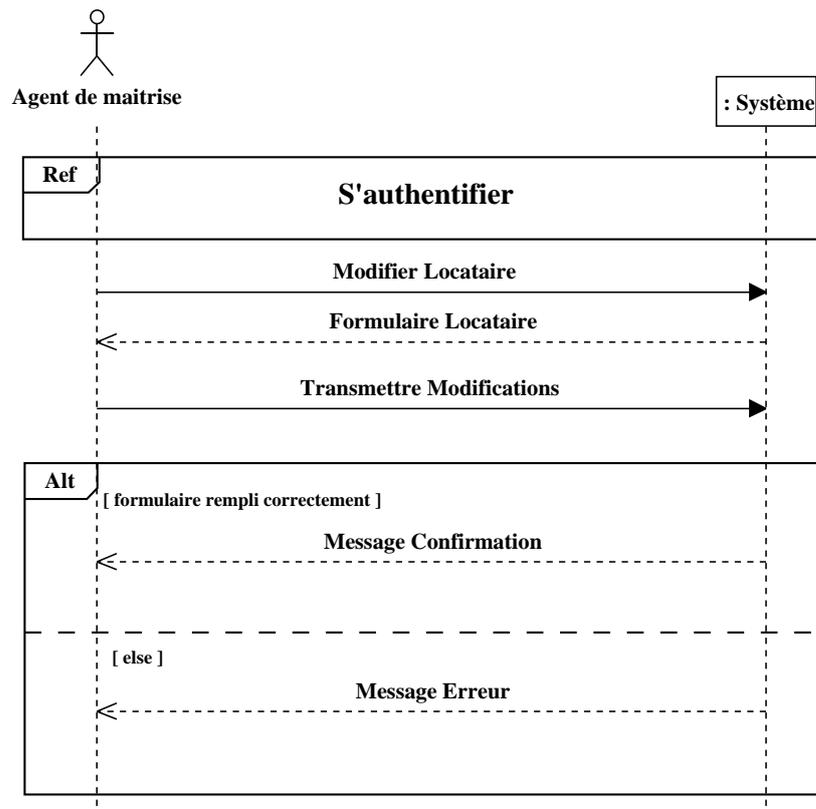


Figure 3.19: Diagramme de séquence système du cas d'utilisation «modifier un locataire»

3.3.6.16 Supprimer un locataire

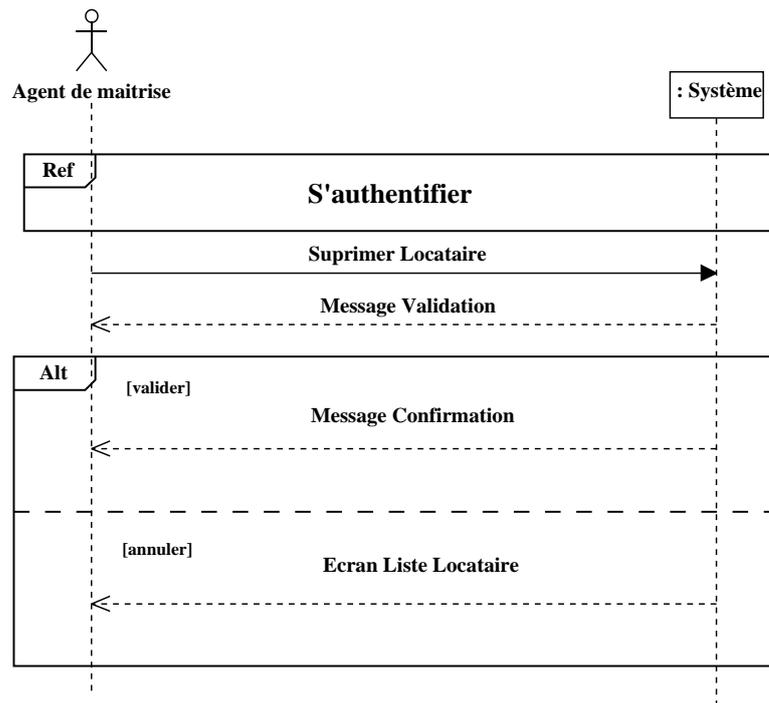


Figure 3.20: Diagramme de séquence système du cas d'utilisation «supprimer un locataire»

3.3.6.17 Générer les loyers

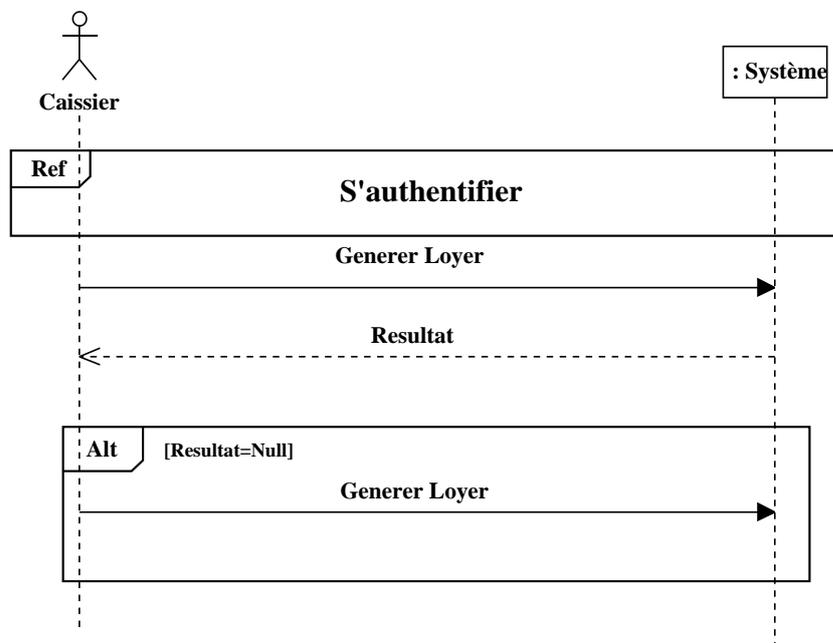


Figure 3.21: Diagramme de séquence système du cas d'utilisation «générer les loyers»

3.3.6.18 Consulter les loyers

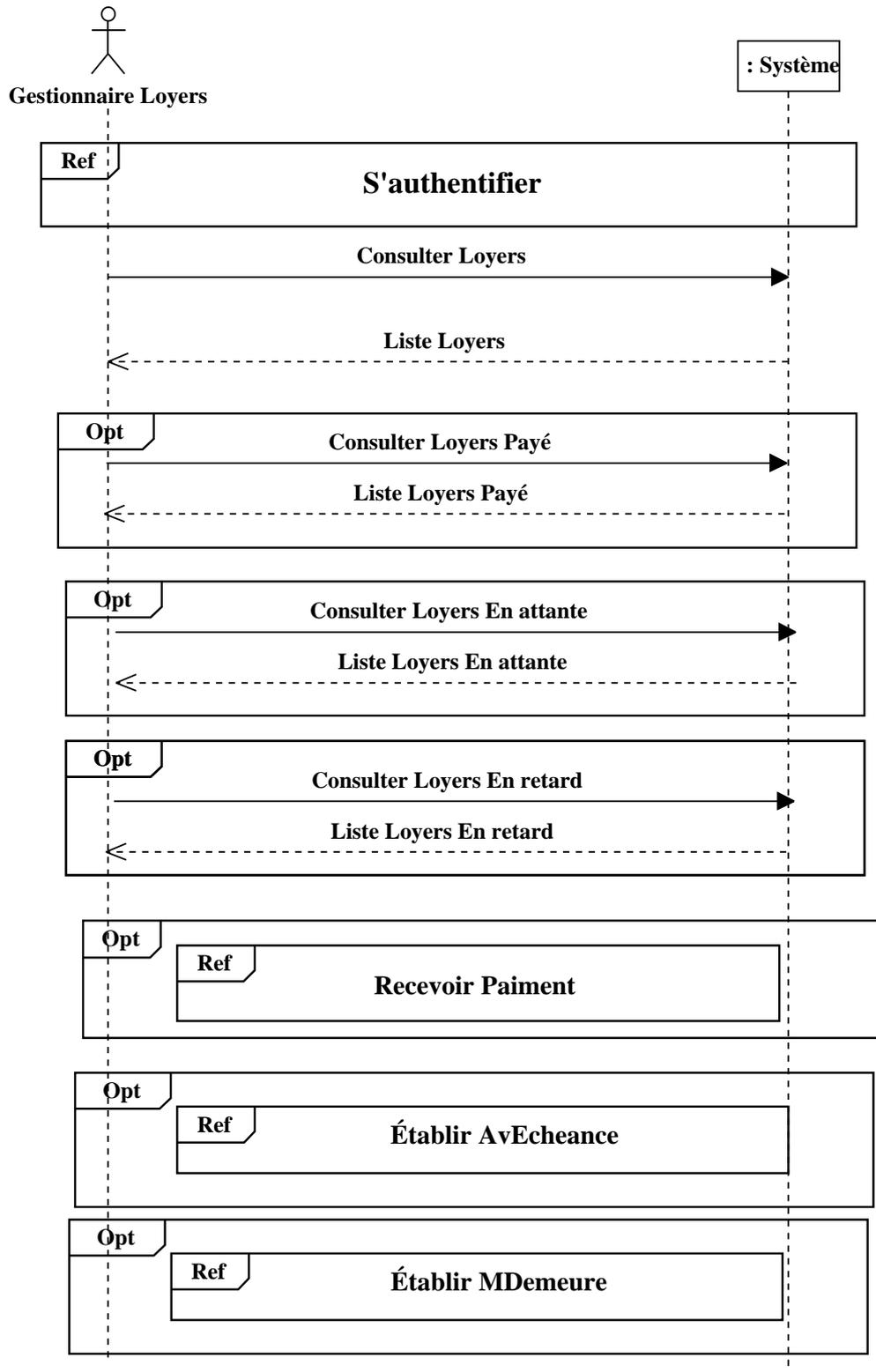


Figure 3.22: Diagramme de séquence système du cas d'utilisation «consulter les loyers»

3.3.6.19 Recevoir un paiement

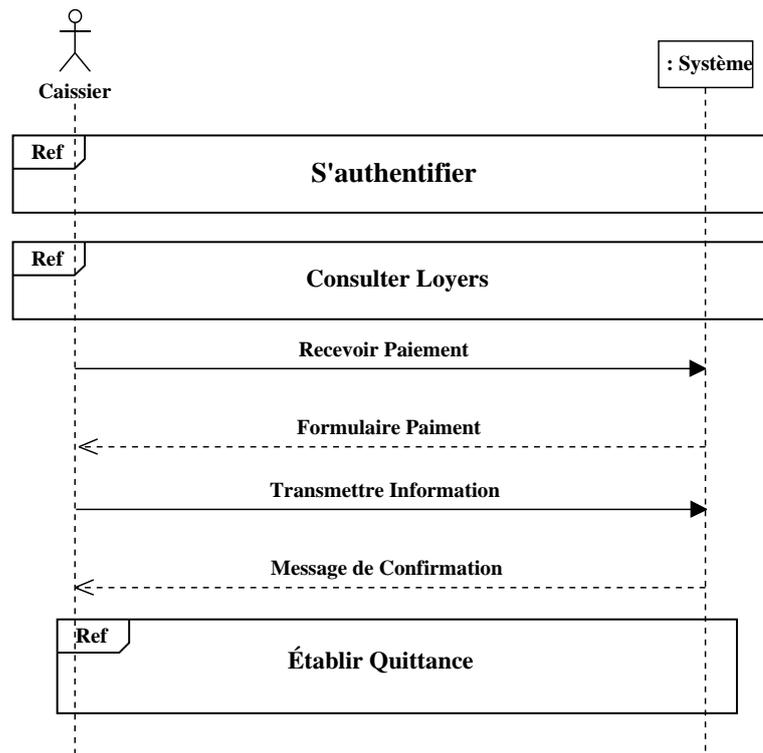


Figure 3.23: Diagramme de séquence système du cas d'utilisation «recevoir un paiement»

3.3.6.20 Établir une quittance

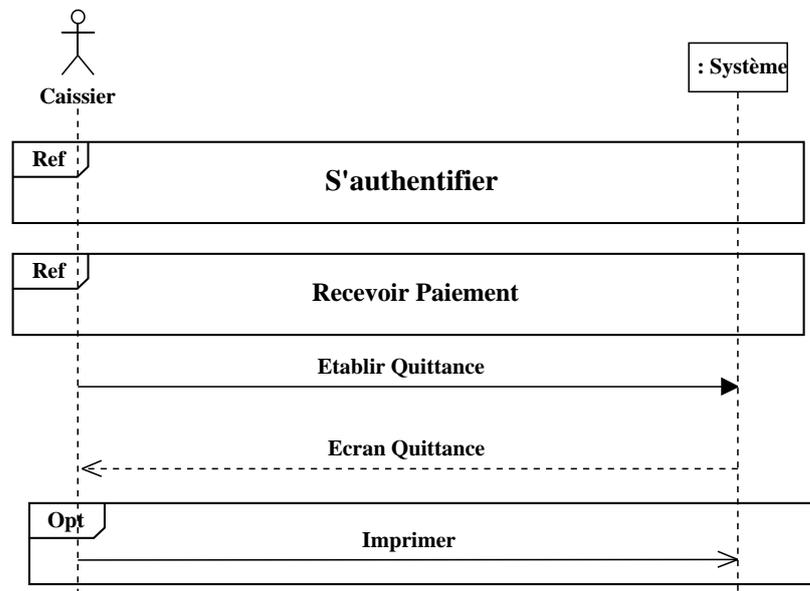


Figure 3.24: Diagramme de séquence système du cas d'utilisation «établir une quittance»

3.3.6.21 Établir un avis d'échéance

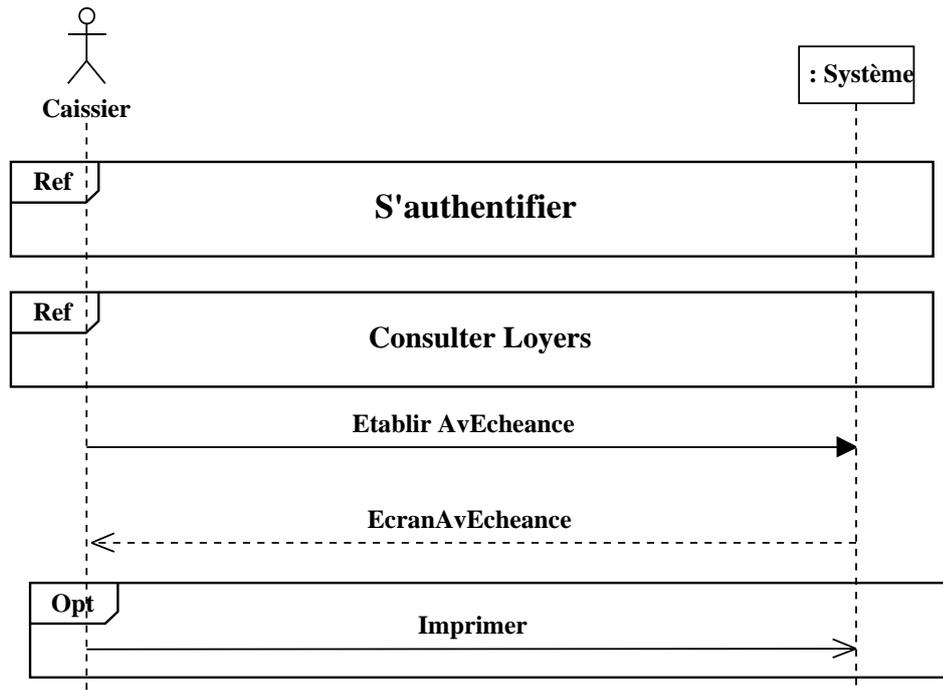


Figure 3.25: Diagramme de séquence système du cas d'utilisation «établir un avis d'échéance»

3.3.6.22 Établir une mise en demeure

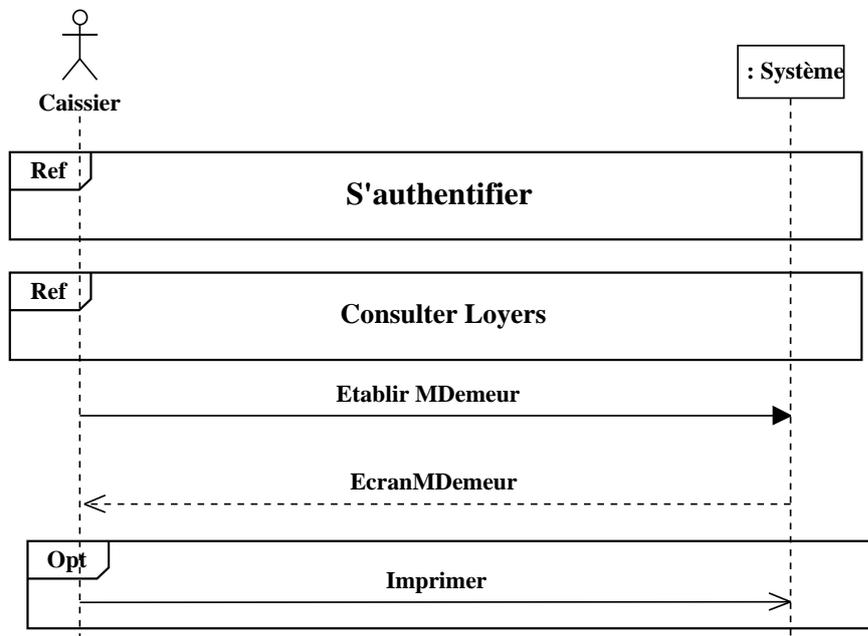


Figure 3.26: Diagramme de séquence système du cas d'utilisation «établir une mise en demeure»

3.3.6.23 Consulter le tableau de bord

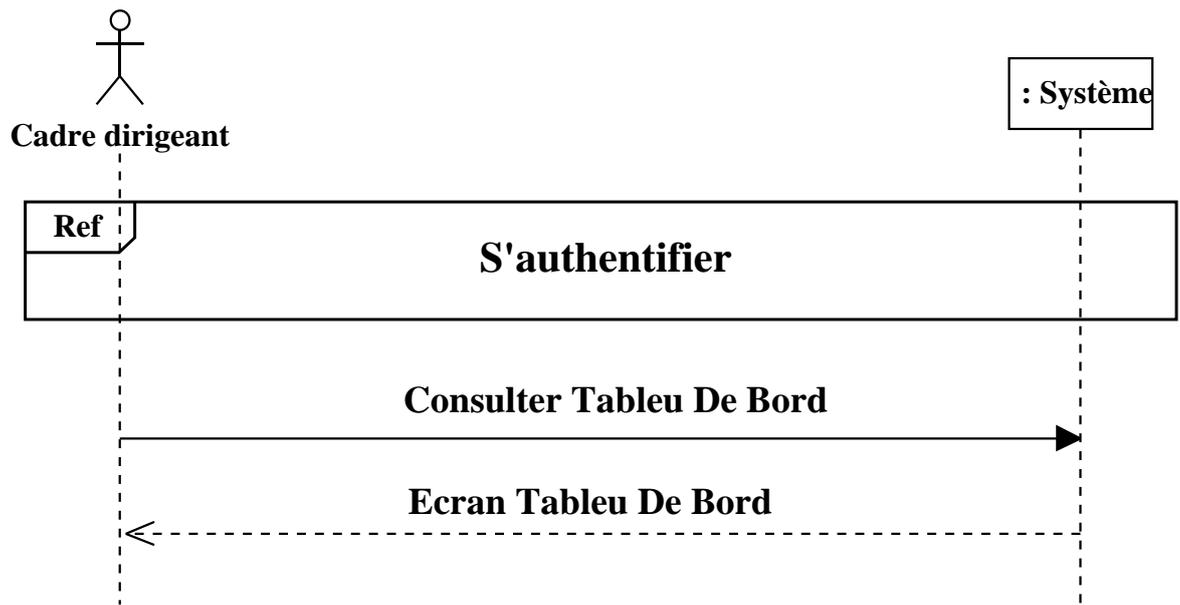


Figure 3.27: Diagramme de séquence système du cas d'utilisation «consulter le tableau de bord»

3.4 Conclusion

Dans ce chapitre, nous avons mis l'accent sur le langage et le processus utilisés pour l'analyse et la conception de l'application. Puis nous avons identifié les besoins du maître d'ouvrage (L'OPGI) dans le cahier des charges. Ensuite, nous avons élaboré le diagramme de cas d'utilisation. Et nous avons fini par l'élaboration des descriptions textuelles et des diagrammes de séquence système pour chacun des cas d'utilisation.

Chapitre 4

Phase d'analyse

4.1 Introduction

Dans cette phase nous allons modéliser les concepts du domaine par un diagramme de classe qu'on appelle «Modèle du domaine», puis, nous développerons les diagrammes de classes participantes qui effectuent la jonction entre, d'une part les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de séquence et les diagrammes de classes de conception. Finalement, nous représenterons l'activité de navigation dans l'interface de la future application en produisant des Diagrammes d'activités de navigations.

4.2 Modèle du domaine

Classe	Attribut	Code	Type
Gestionnaire	L'identificateur du gestionnaire.	id_U	Numérique
	Le nom d'utilisateur du gestionnaire.	Username	Chaine de caractère
	Le mot de passe du gestionnaire.	Password	Chaine de caractère
	Le type du gestionnaire (Web Master, Agent de maitrise, Cadre dirigeant, Caissier).	Type	Chaine de caractère
	Si l'acteur à le droit de gérer les compte ou non	gererComptes	Texte

	Si l'acteur à le droit de gérer les biens immobiliers ou non	gererBiens	Texte
	Si l'acteur à le droit de gérer les locataires ou non	gererLocataires	Texte
	Si l'acteur à le droit de gérer les paiements ou non	gererPaiements	Texte
	Si l'acteur à le droit de consulter le tableau de bord ou non	TableauDeBord	Texte
Bâtiment	L'identificateur de bâtiment.	id_B	Numérique
	Numéro de bâtiment.	NBat	Numérique
	La Daïra dans laquelle se trouve le bâtiment.	Daira	Chaine de caractère
	La Commune dans laquelle se trouve le bâtiment.	Commune	Chaine de caractère
	La Rue dans laquelle se trouve le bâtiment.	Rue	Chaine de caractère
Bien Immobilier	L'identificateur du bien immobilier.	id_Bn	Numérique
	L'étage dans lequel se trouve le bien immobilier	Etage	Numérique
	Le numéro du bien immobilier	NumLog	Numérique
	La superficie du bien immobilier	Sprf	Réel
	Le Type du bien immobilier (F1, F2, F3, F4 ou F5).	Type	Chaine de caractère
	Usage du bien immobilier habitation ou professionnelle.	Us	Chaine de caractère
Locataire	L'identificateur du locataire	id_L	Numérique
	Le nom du locataire.	Nom	Chaine de caractère
	Le prénom du locataire.	Prénom	Chaine de caractère
	La date de naissance du locataire.	DateN	Date
	Le lien de la naissance de locataire.	LienN	Chaine de caractère
Contrat	L'identificateur du contrat de location.	id_C	Numérique
	L'identificateur du bien immobilier.	id_Bn	Numérique
	Date de début de contrat.	DateD	Date
	Date de la fin de contrat	DateF	Date
	Caution à payer.	Caut	Réel
	Loyer mensuel à payer	LoyerM	Réel
Loyer	L'identificateur du loyer.	id_Lo	Numérique
	Le montant du loyer.	MontL	Numérique

	Début de période du loyer.	DebutP	Date
	Fin de période du loyer.	FinP	Date
	L'échéance.	Echeance	Date
	L'état du loyer (Payé, En attente, En retard).	Etat	Chaine de caractère
Paiement	L'identificateur du paiement.	id_P	Numérique
	Le montant a payé.	MontP	Double
Quittance	L'identification de la quittance.	id_Q	Numérique
	La date de la quittance.	DateQ	Date
	Le montant	MontTotal	Réel
Avis d'échéance	L'identificateur de l'avis.	id_A	Numérique
	La date de l'avis.	DateA	Date
	Le montant total.	MontTotal	Réel
Mise en demeure	L'identificateur de la mise en demeure.	id_M	Numérique
	La date de la mise.	DateM	Date
	Le montant.	MontTotal	Réel

Tableau 4.1: Dictionnaire de données.

4.2.1 Modèle du domaine

Le modèle du domaine définit les classes qui modélisent les concepts du monde réel présents dans le domaine de l'application [20]. Ainsi, nous avons élaboré un modèle du domaine de la gestion locative adapté, et qui servira plus tard pour une implémentation sous forme d'une base de données (Figure 4.1).

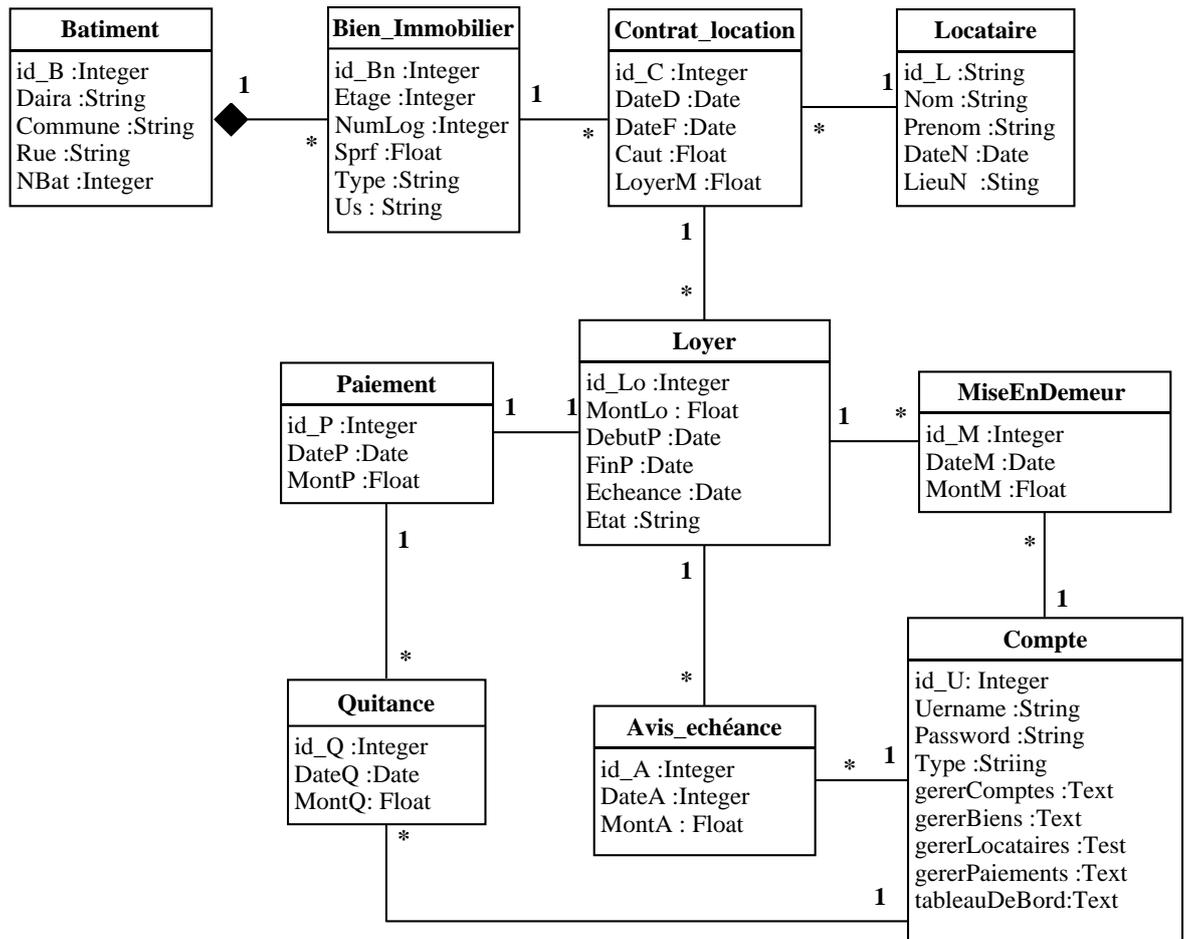


Figure 4.1: Modèle du domaine

4.3 Diagrammes de classes participantes

Le diagramme de classes participantes est particulièrement important puisqu'il effectue la jonction entre, d'une part, les cas d'utilisation, le modèle du domaine et la maquette, et d'autre part, les diagrammes de conception (diagrammes d'interaction et les diagrammes de classes de conception [20]). Ainsi, nous avons réalisé un diagramme de classes participantes pour chaque cas d'utilisation.

4.3.1 S'authentifier

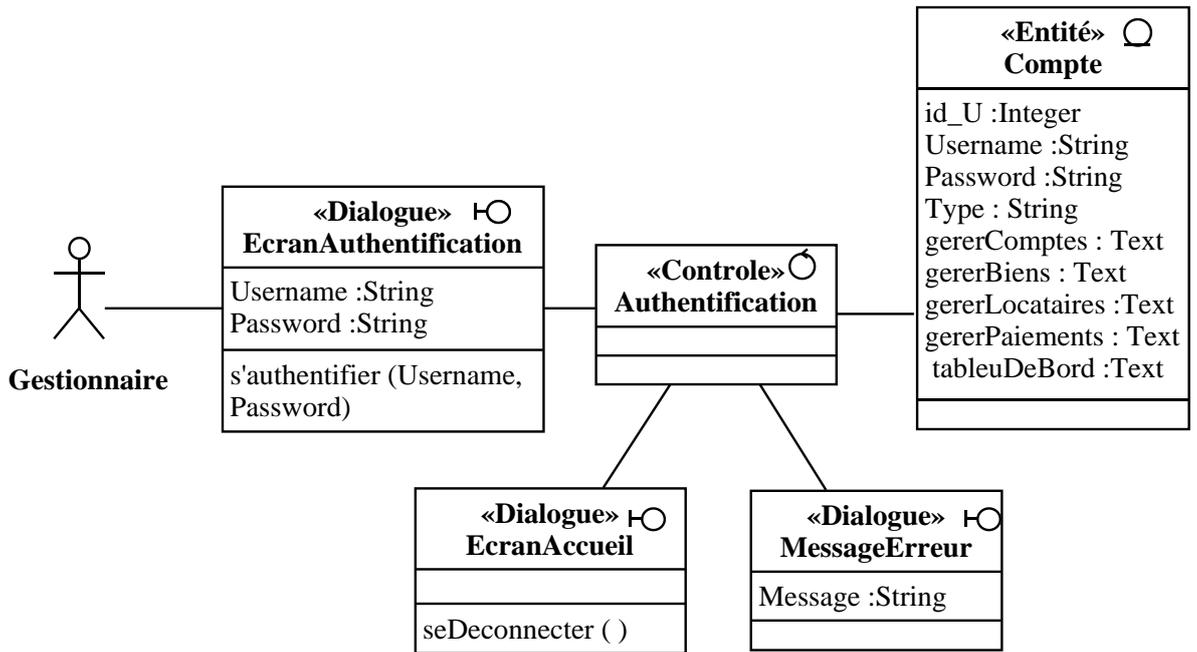


Figure 4.2: Diagramme de classes participantes du cas d'utilisation «s'authentifier»

4.3.2 Créer un compte

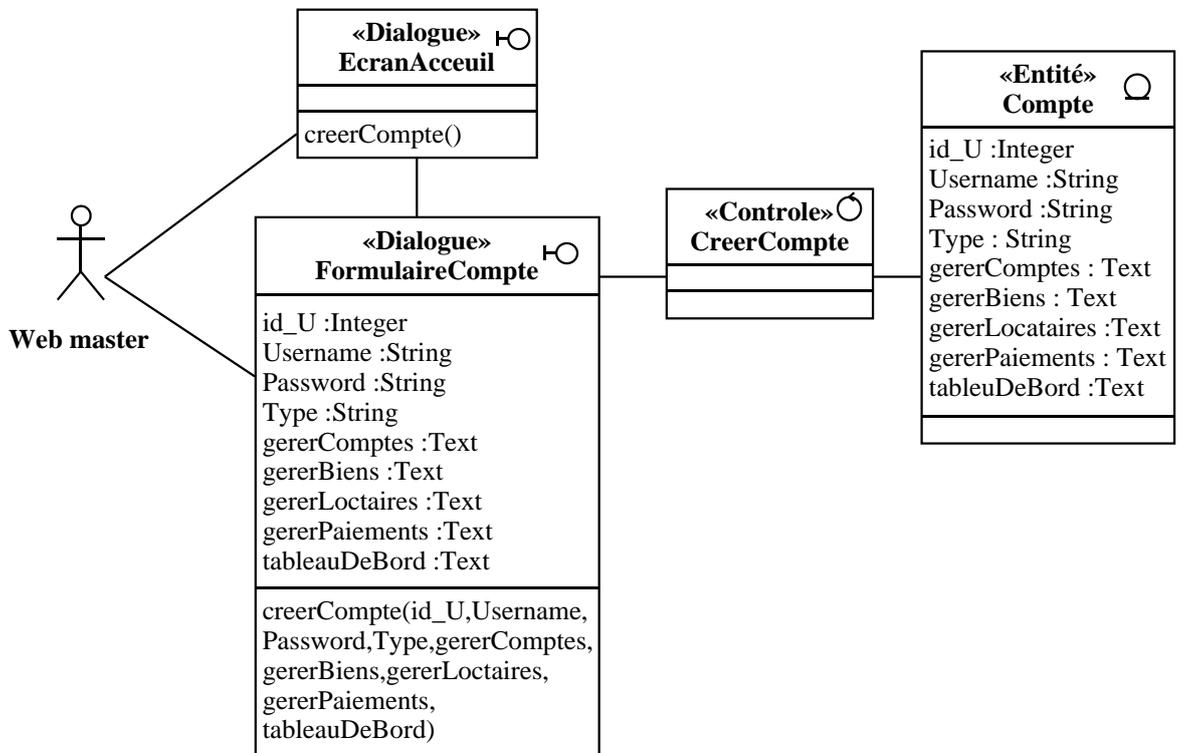


Figure 4.3: Diagramme de classes participantes du cas d'utilisation «créer un compte»

4.3.3 Consulter les comptes

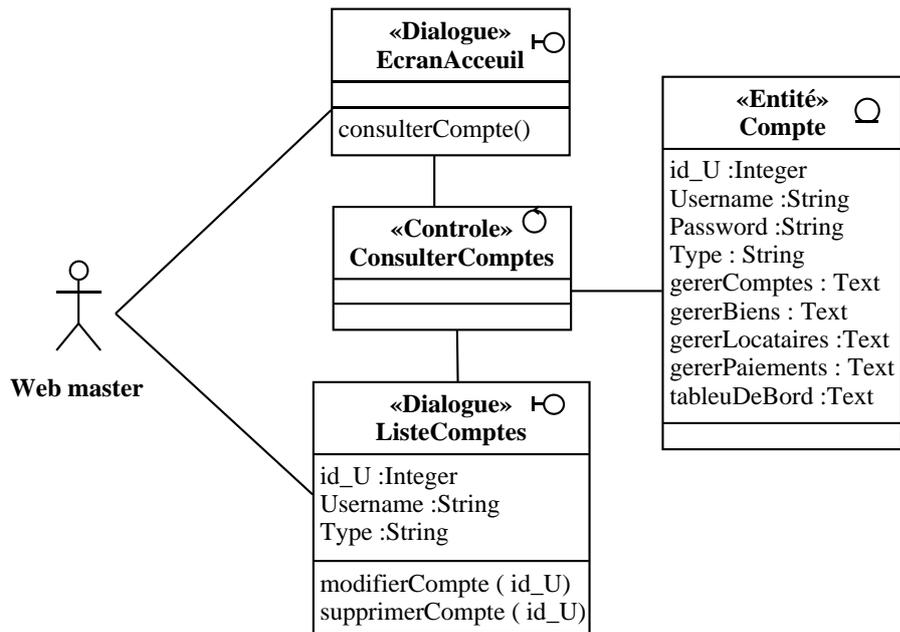


Figure 4.4: Diagramme de classes participantes du cas d'utilisation «consulter les comptes»

4.3.4 Modifier un compte

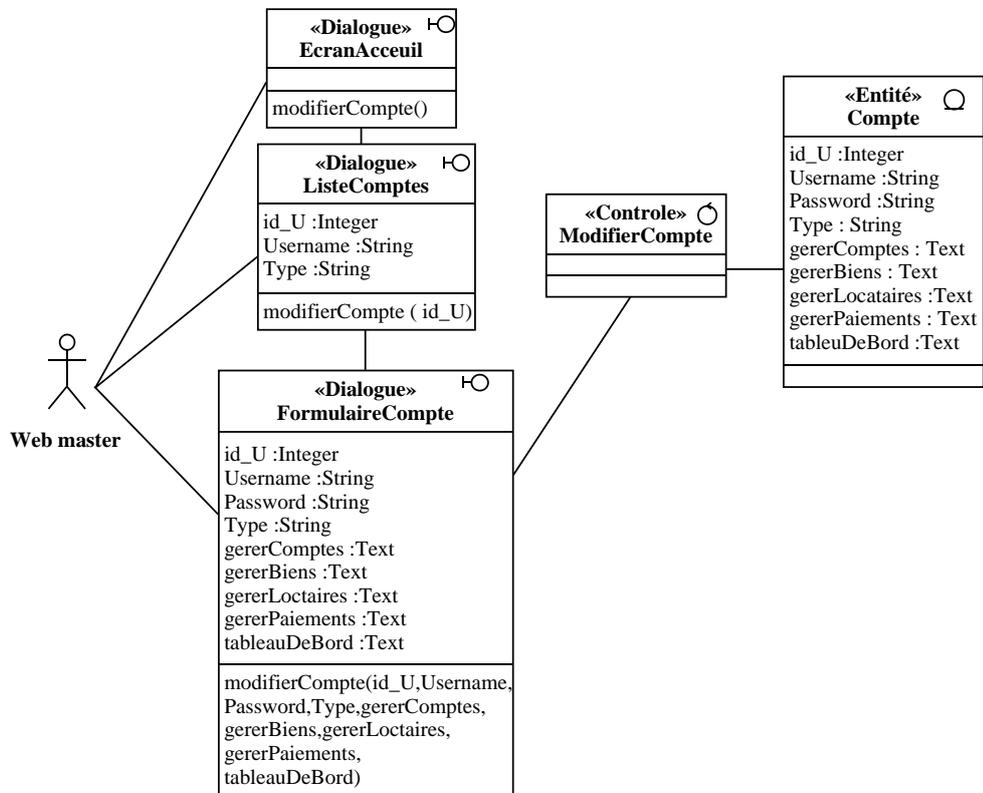


Figure 4.5: Diagramme de classes participantes du cas d'utilisation «modifier un compte»

4.3.5 Supprimer un compte

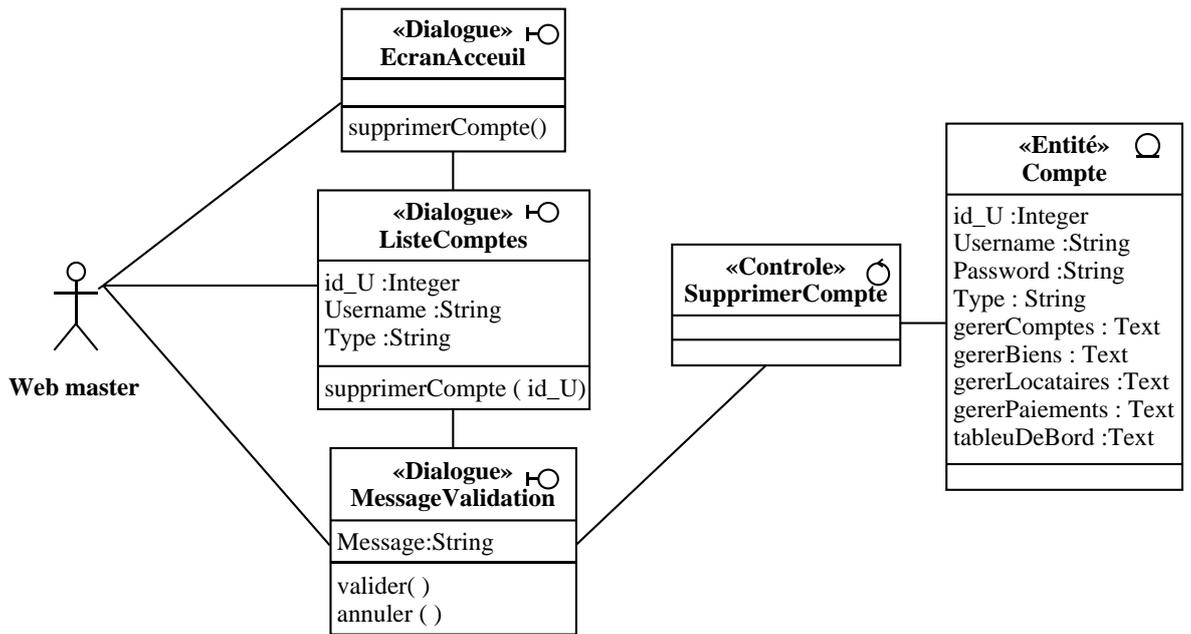


Figure 4.6: Diagramme de classes participantes du cas d'utilisation «supprimer un compte»

4.3.6 Ajouter un bien immobilier

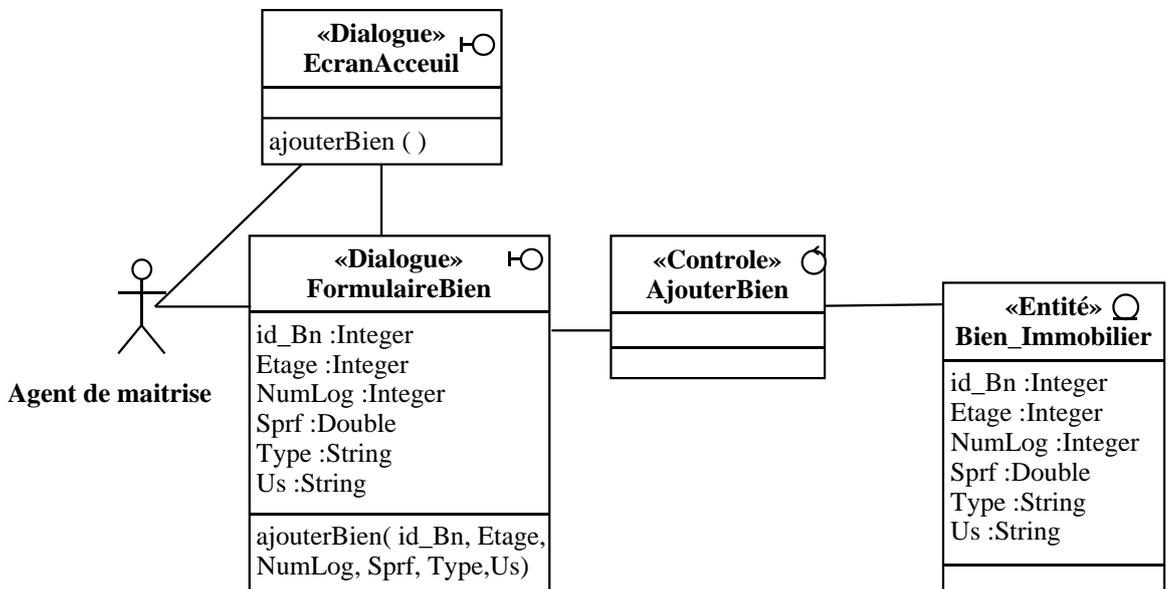


Figure 4.7: Diagramme de classes participantes du cas d'utilisation «ajouter un bien immobilier»

4.3.7 Consulter les biens immobiliers

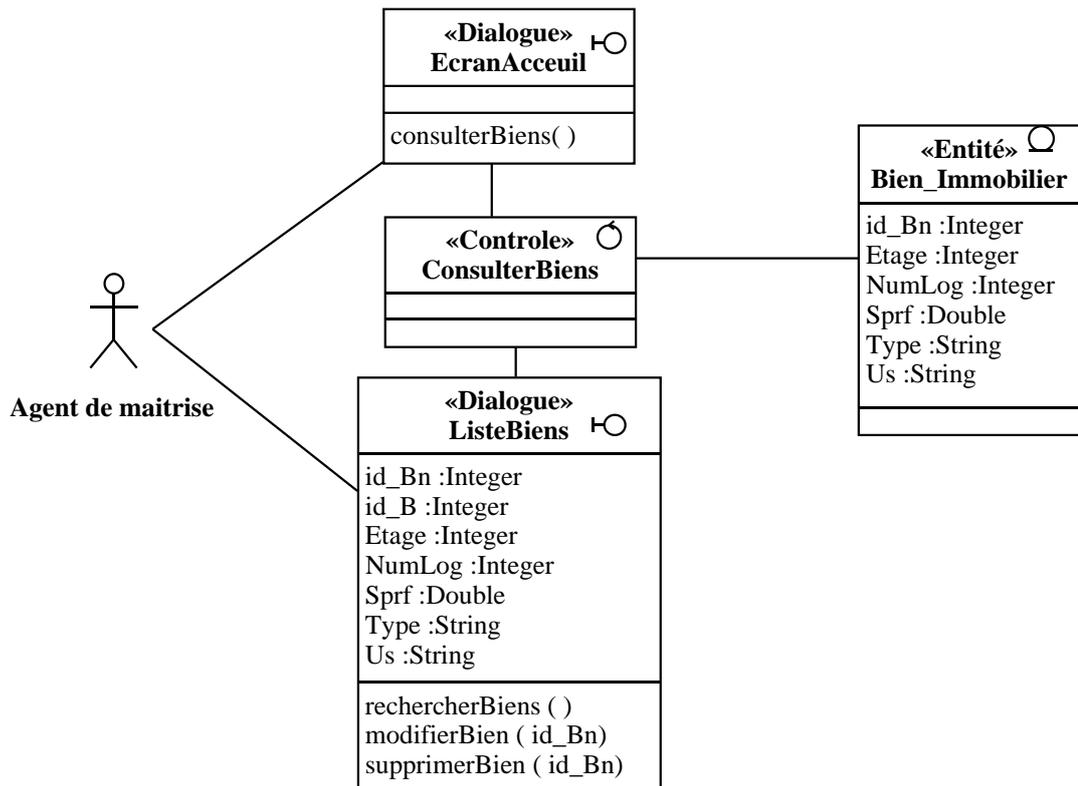


Figure 4.8: Diagramme de classes participantes du cas d'utilisation «consulter les biens immobiliers»

4.3.8 Rechercher les biens immobiliers

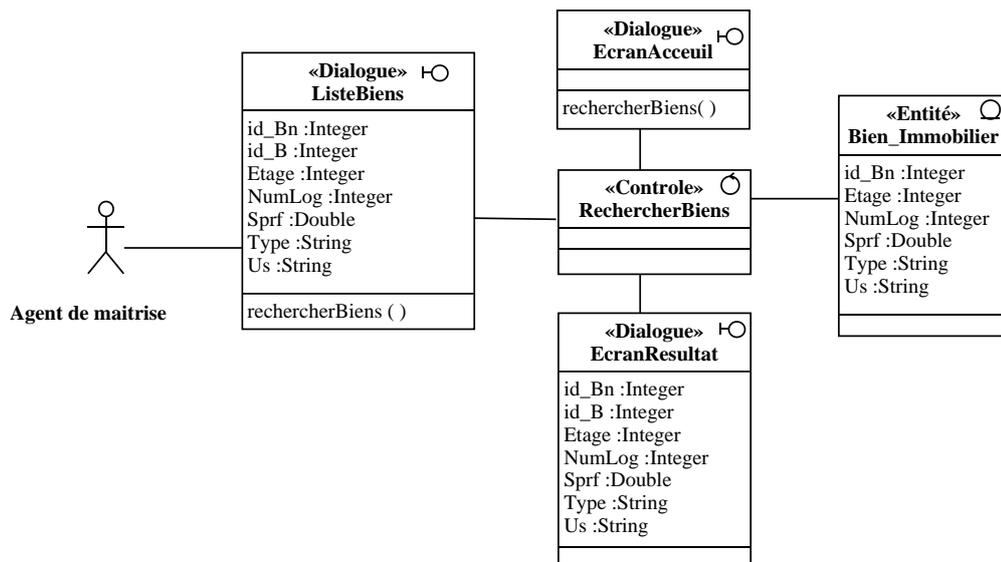


Figure 4.9: Diagramme de classes participantes du cas d'utilisation «rechercher les biens immobiliers»

4.3.9 Modifier un bien immobilier

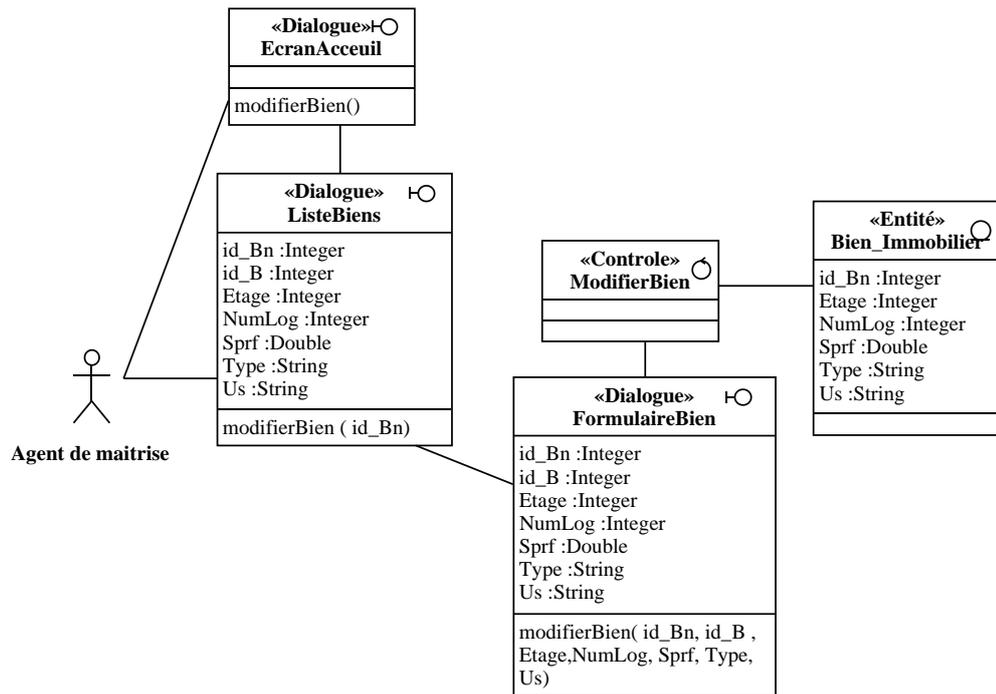


Figure 4.10: Diagramme de classes participantes du cas d'utilisation «modifier un bien immobilier»

4.3.10 Supprimer un bien immobilier

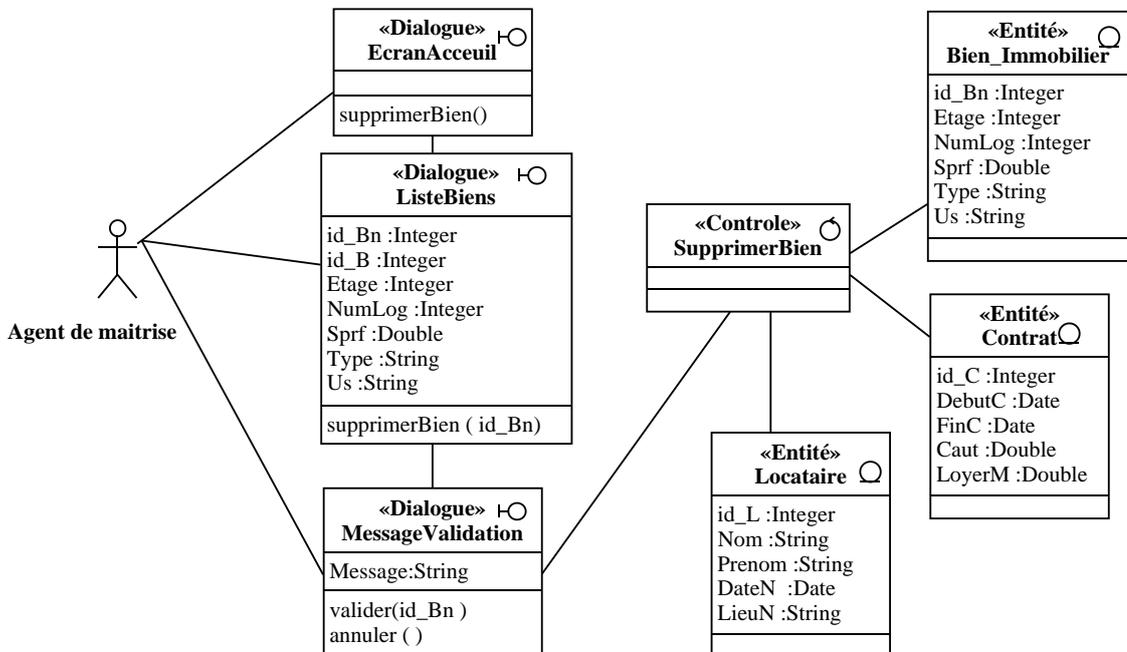


Figure 4.11: Diagramme de classes participantes du cas d'utilisation «supprimer un bien immobilier»

4.3.11 Ajouter un locataire

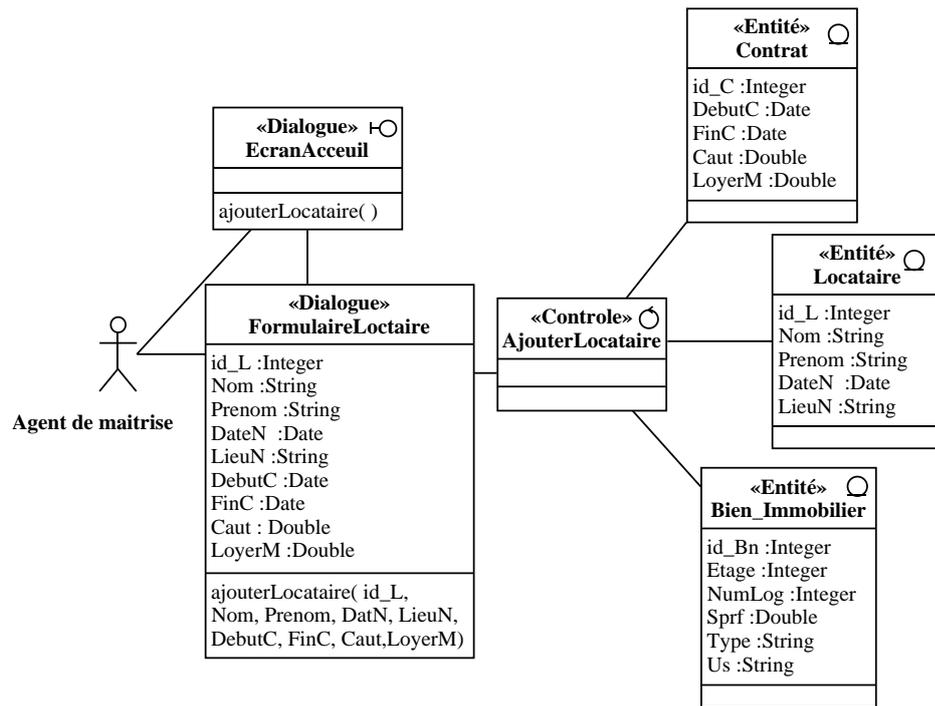


Figure 4.12: Diagramme de classes participantes du cas d'utilisation «ajouter un locataire»

4.3.12 Établir un contrat de location

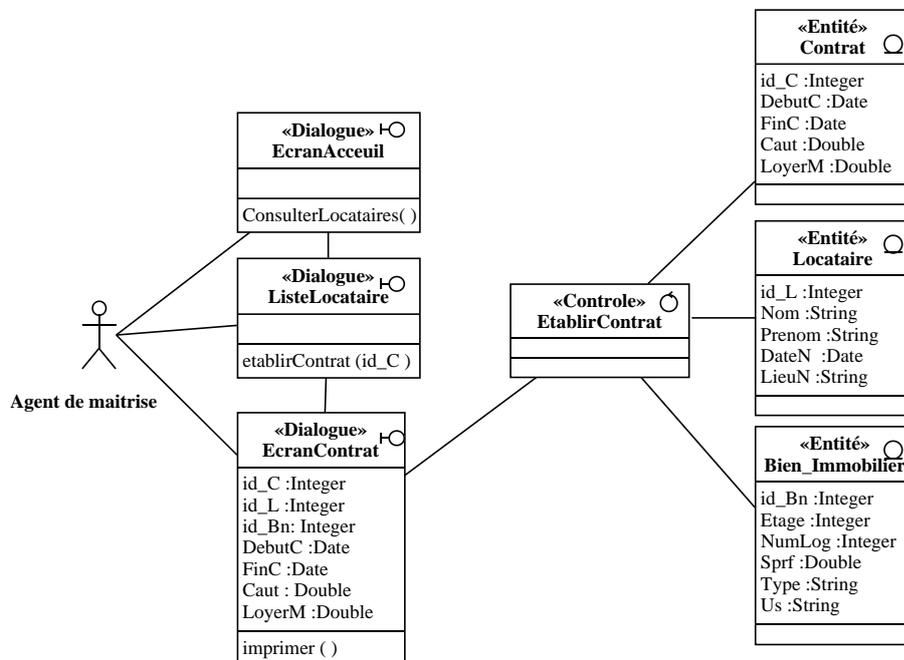


Figure 4.13: Diagramme de classes participantes du cas d'utilisation «établir un contrat de location »

4.3.13 Consulter les locataires

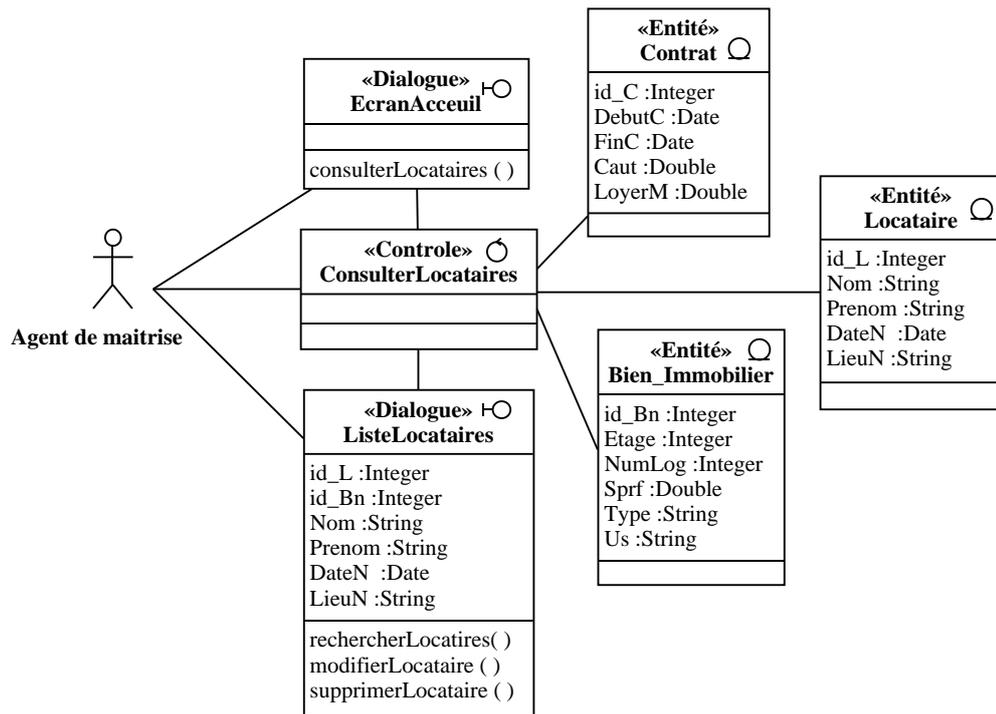


Figure 4.14: Diagramme de classes participantes du cas d'utilisation «consulter les locataires»

4.3.14 Rechercher les locataires

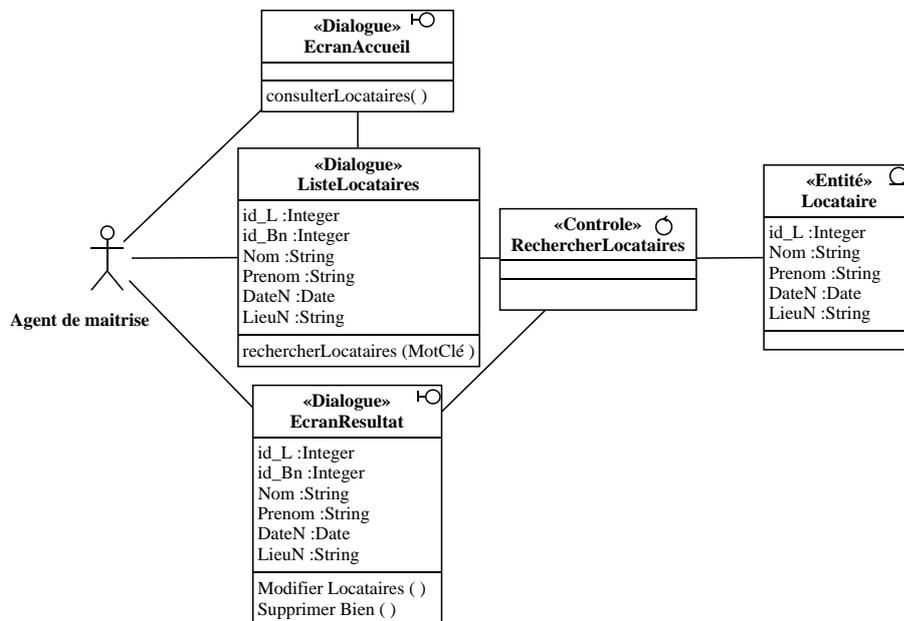


Figure 4.15: Diagramme de classes participantes du cas d'utilisation «rechercher les locataires»

4.3.15 Modifier un locataire

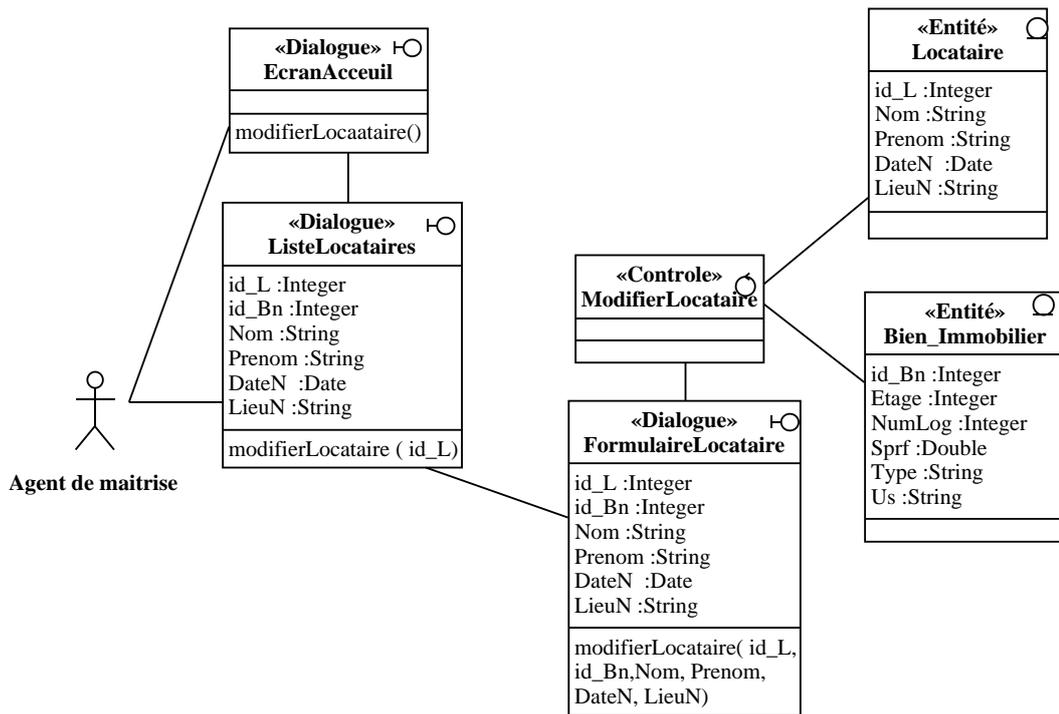


Figure 4.16: Diagramme de classes participantes du cas d'utilisation «modifier un locataire»

4.3.16 Supprimer un locataire

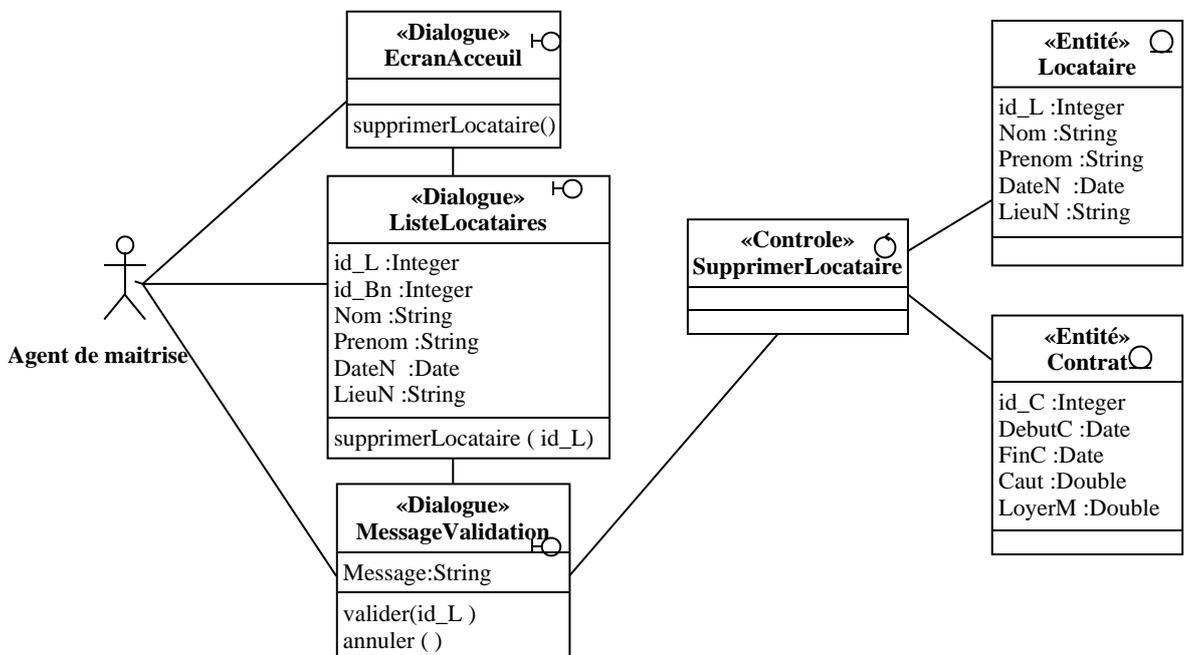


Figure 4.17: Diagramme de classes participantes du cas d'utilisation «supprimer un locataire»

4.3.17 Générer les loyers

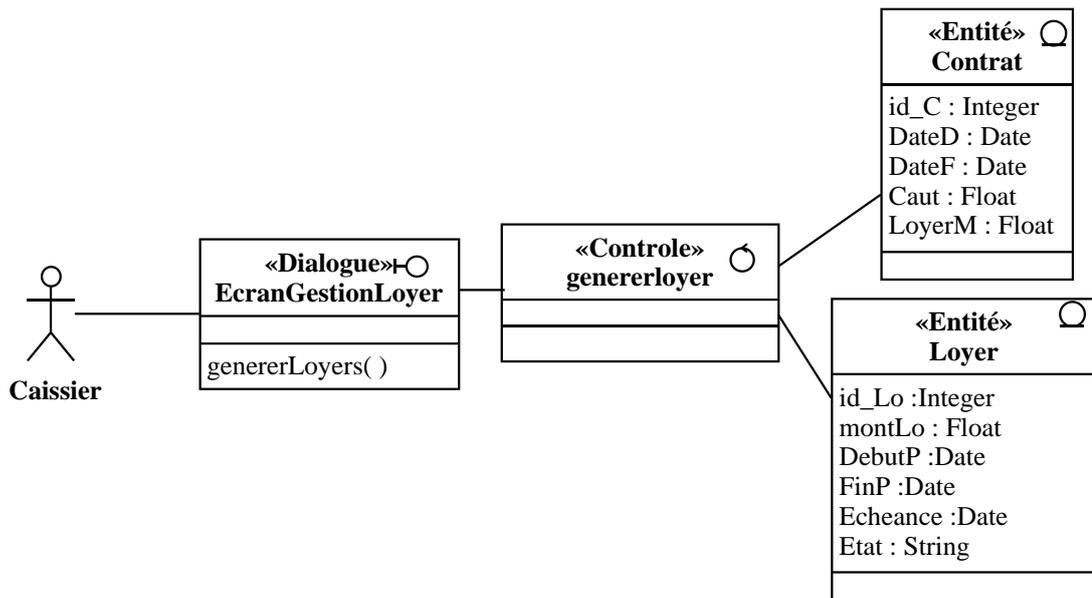


Figure 4.18: Diagramme de classes participantes du cas d'utilisation «générer les loyers»

4.3.18 Consulter les Loyers

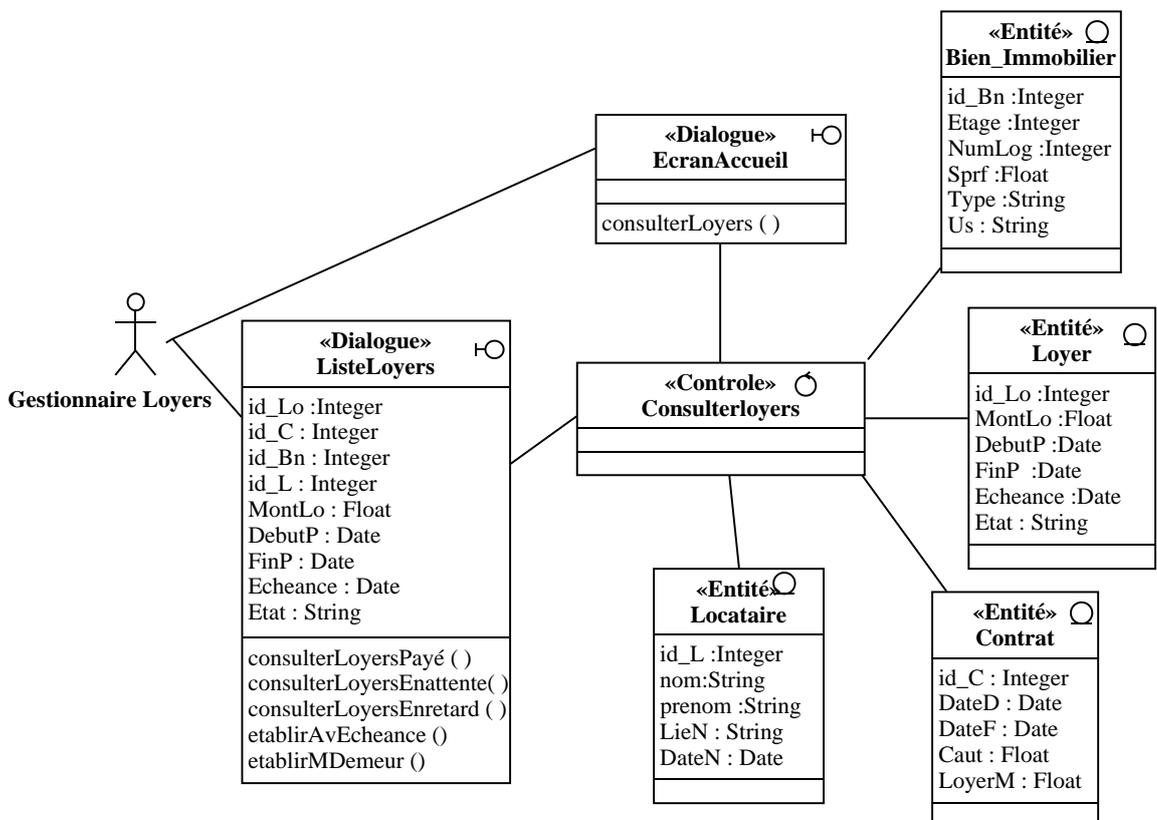


Figure 4.19: Diagramme de classes participantes du cas d'utilisation «consulter les loyers»

4.3.19 Recevoir un paiement

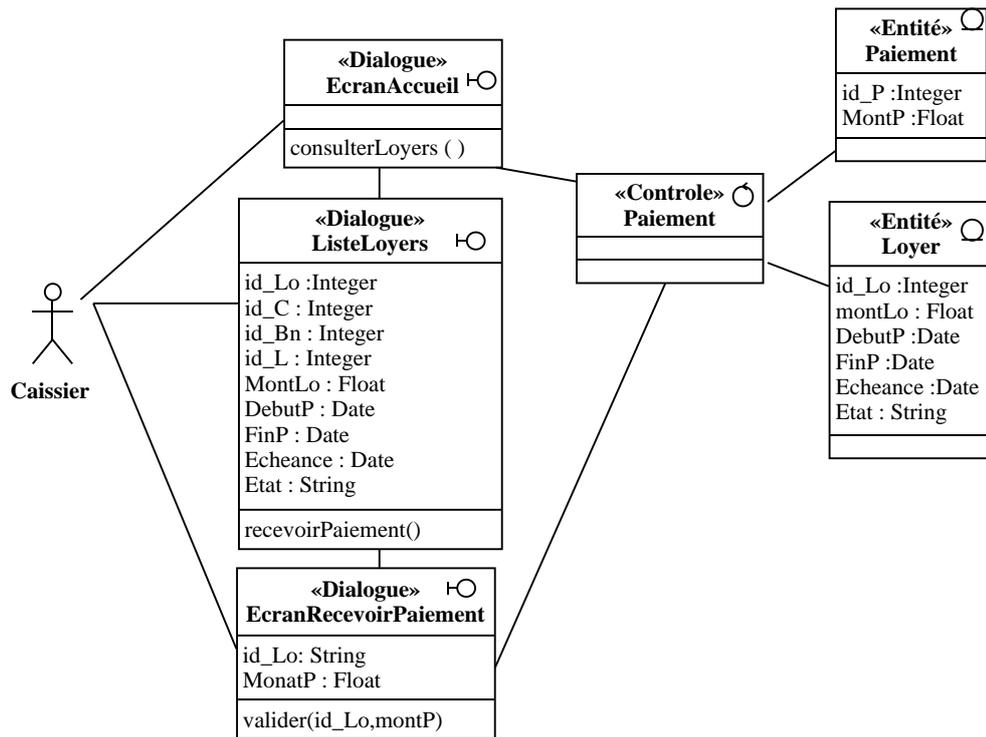


Figure 4.20: Diagramme de classes participantes du cas d'utilisation «recevoir un paiement»

4.3.20 Établir une quittance

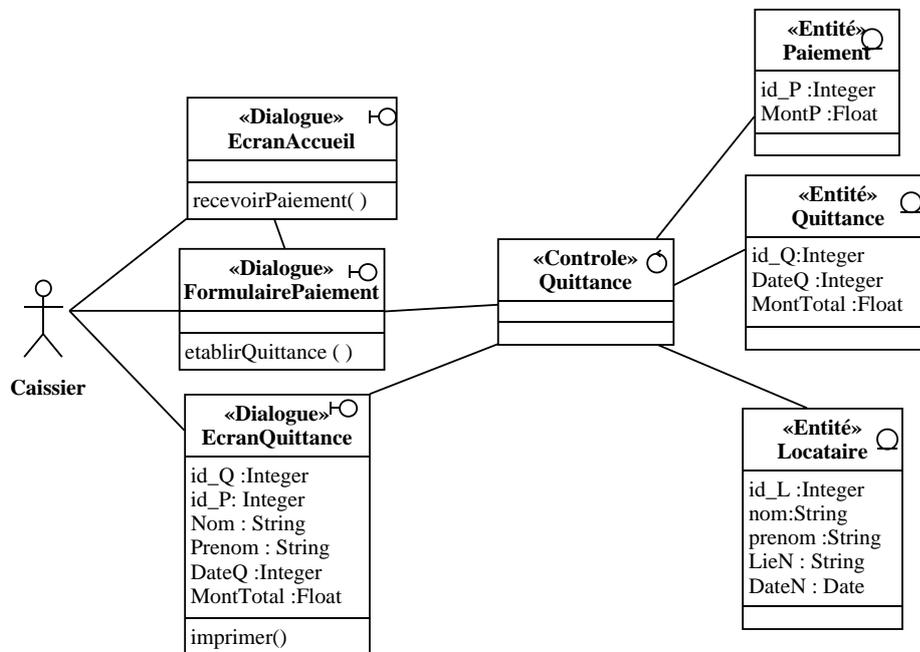


Figure 4.21: Diagramme de classes participantes du cas d'utilisation «établir une quittance»

4.3.21 Établir un avis d'échéance

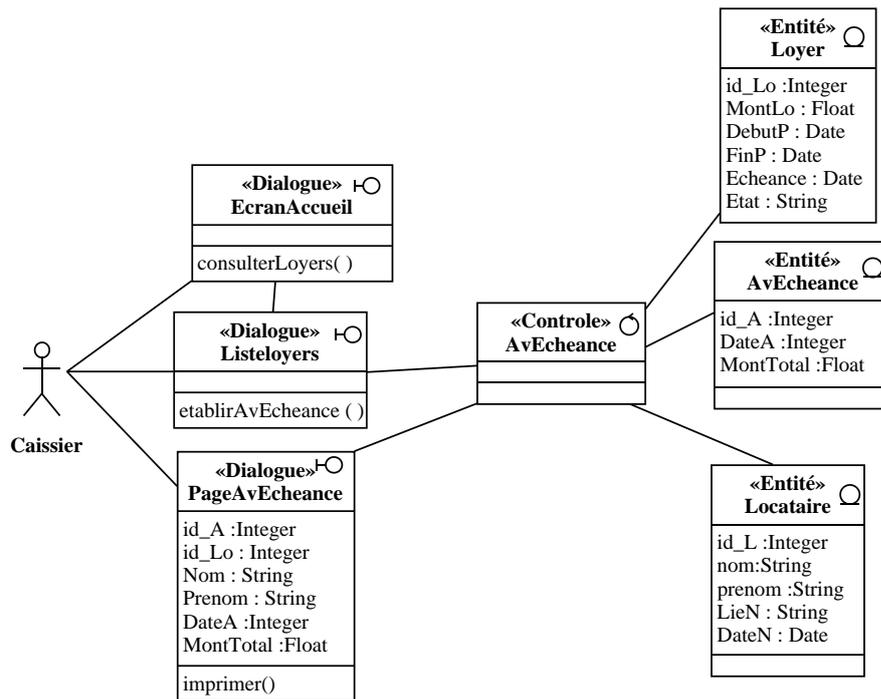


Figure 4.22: Diagramme de classes participantes du cas d'utilisation «établir un avis d'échéance »

4.3.22 Établir une mise en demeure

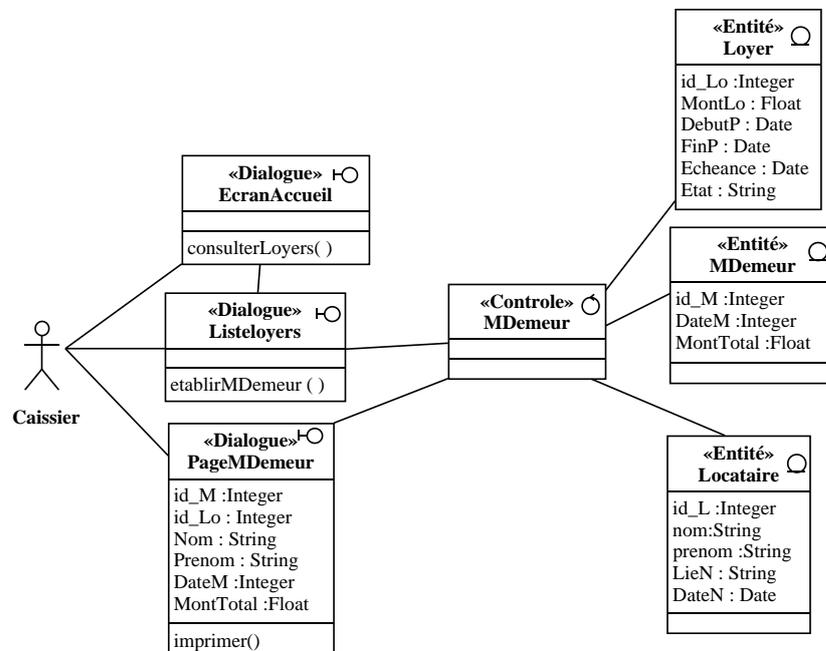


Figure 4.23: Diagramme de classes participantes du cas d'utilisation «établir une mise en demeure»

4.3.23 Consulter le tableau de bord

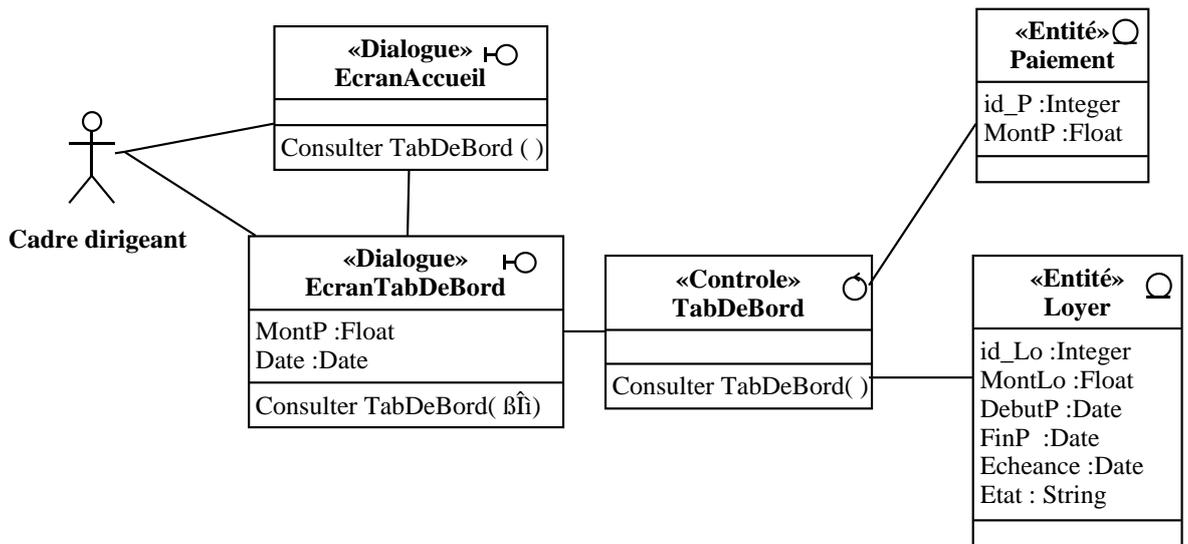


Figure 4.24: Diagramme de classes participantes du cas d'utilisation «consulter le tableau de bord»

4.4 Diagramme d'activité de navigation

Les diagrammes d'activité de navigation représentent graphiquement l'activité de navigation dans l'IHM de l'application, ainsi, il fournit la possibilité de décrire précisément et exhaustivement les aspects dynamiques de l'interface utilisateur [20], [3].

4.4.1 S'authentifier

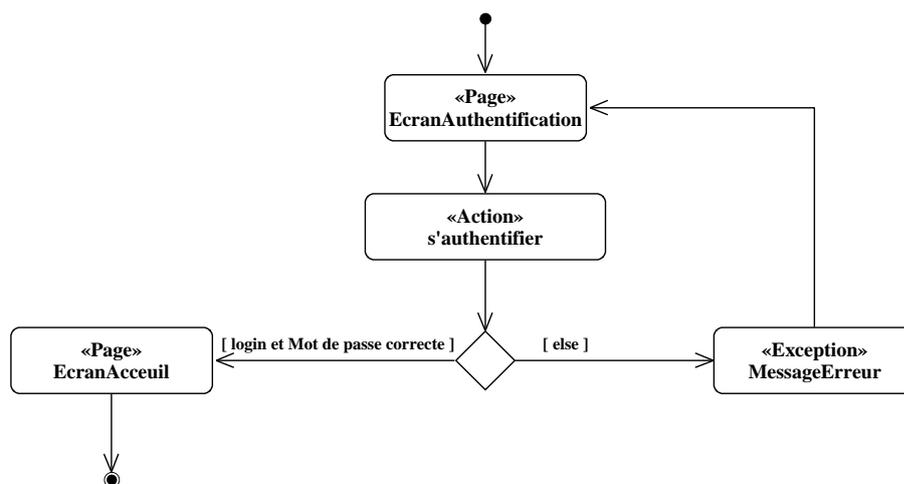


Figure 4.25: Diagramme d'activité de navigation du cas d'utilisation «s'authentifier»

4.4.2 Créer un compte

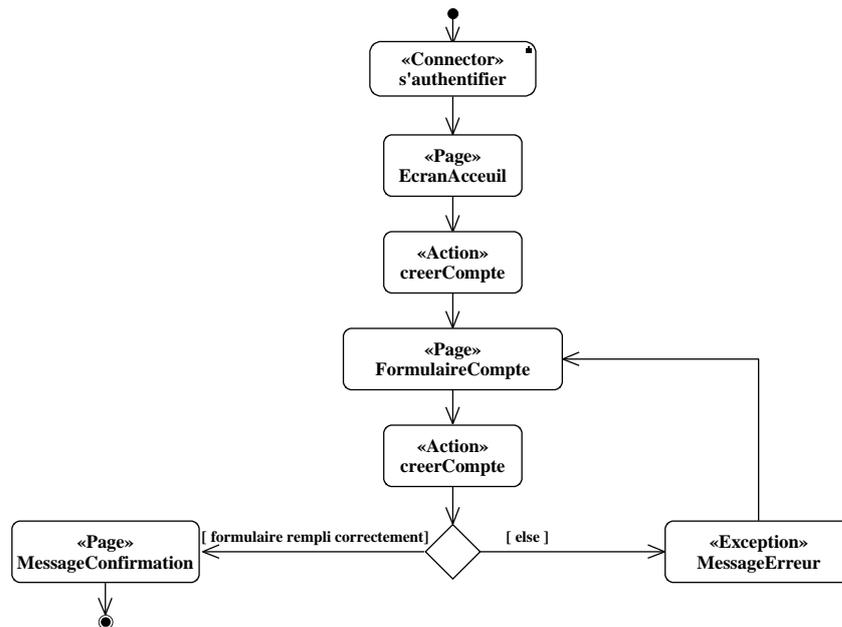


Figure 4.26: Diagramme d'activité de navigation du cas d'utilisation «créer un compte»

4.4.3 Consulter les comptes

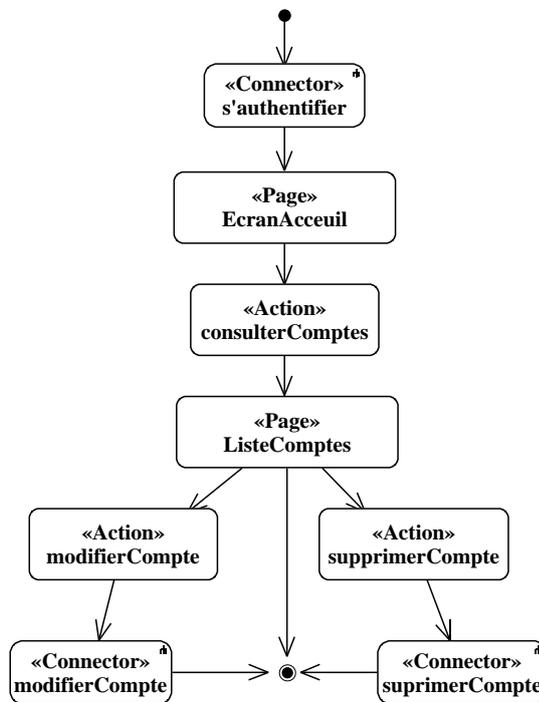


Figure 4.27: Diagramme d'activité de navigation du cas d'utilisation «consulter les comptes»

4.4.4 Modifier un compte

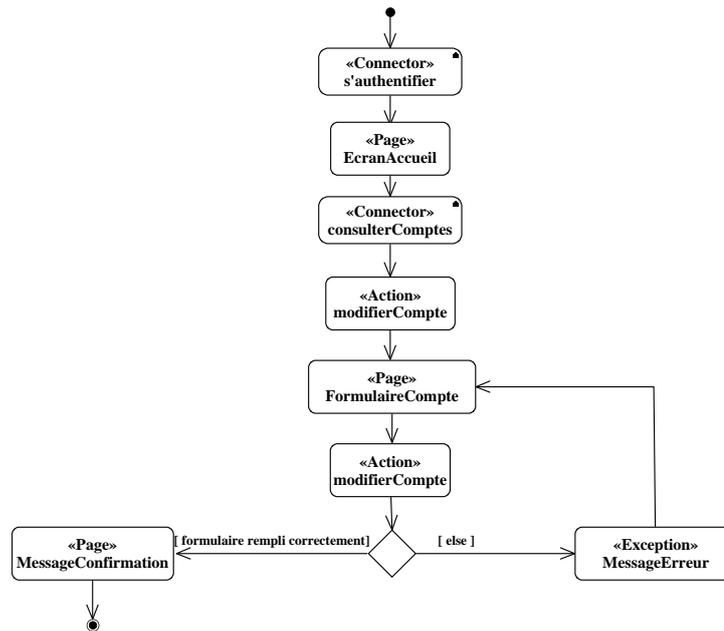


Figure 4.28: Diagramme d'activité de navigation du cas d'utilisation «modifier un compte»

4.4.5 Supprimer un compte

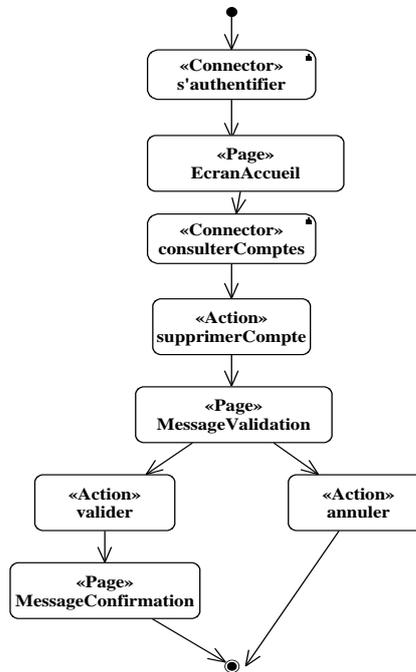


Figure 4.29: Diagramme d'activité de navigation du cas d'utilisation «supprimer un compte»

4.4.6 Ajouter un bien immobilier

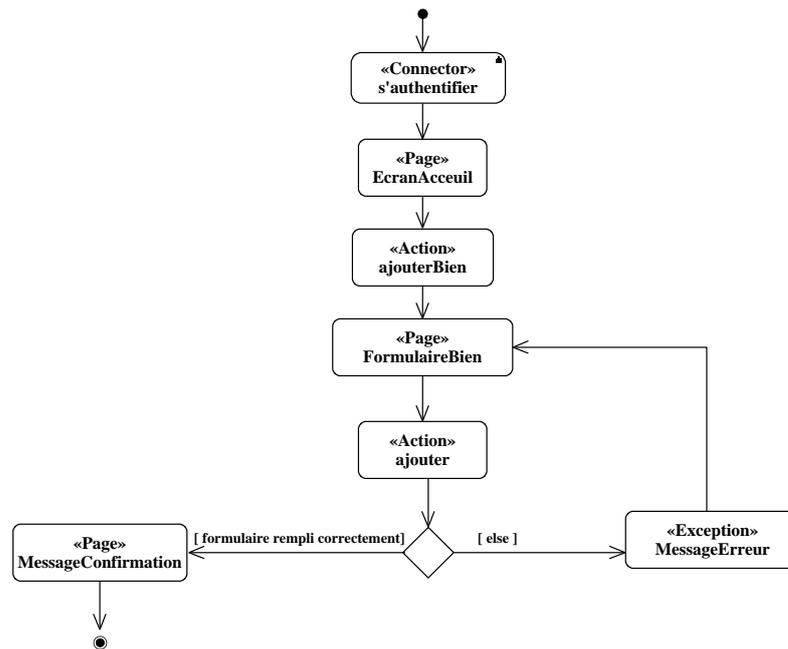


Figure 4.30: Diagramme d'activité de navigation du cas d'utilisation «ajouter un bien immobilier»

4.4.7 Consulter les biens immobiliers

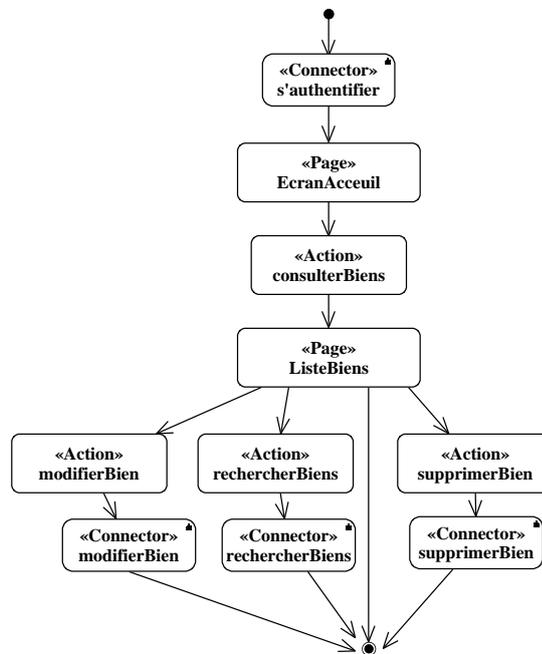


Figure 4.31: Diagramme d'activité de navigation du cas d'utilisation «consulter les biens immobiliers»

4.4.8 Rechercher les biens immobiliers

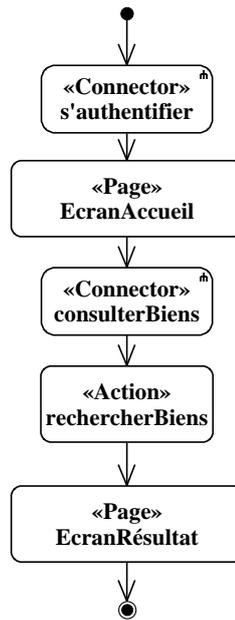


Figure 4.32: Diagramme d'activité de navigation du cas d'utilisation «rechercher les biens immobiliers»

4.4.9 Modifier un bien immobilier

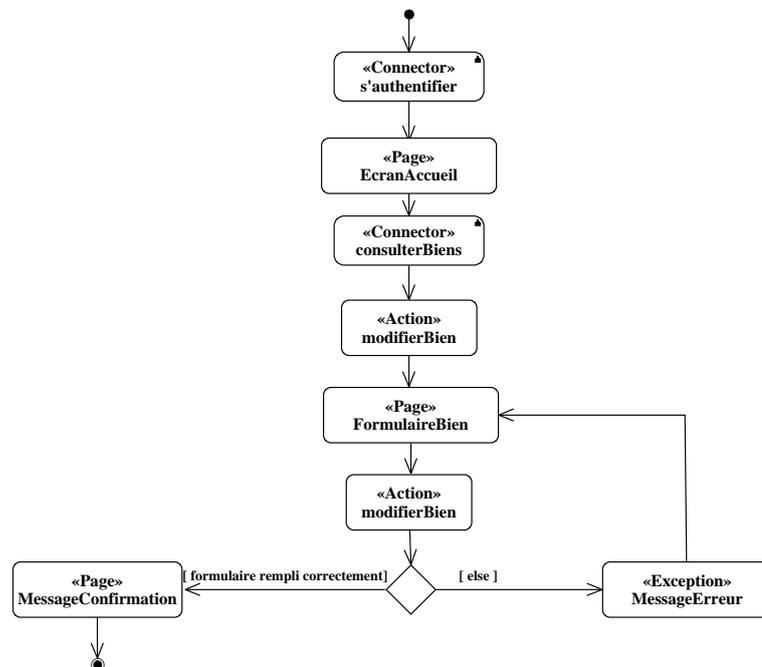


Figure 4.33: Diagramme d'activité de navigation du cas d'utilisation «modifier un bien immobilier»

4.4.10 Supprimer un bien immobilier

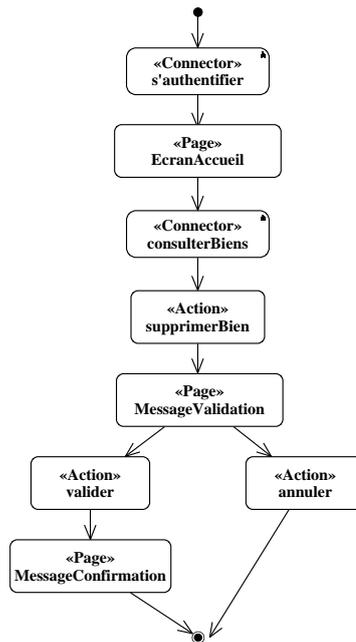


Figure 4.34: Diagramme d'activité de navigation du cas d'utilisation «supprimer un bien immobilier»

4.4.11 Ajouter un locataire

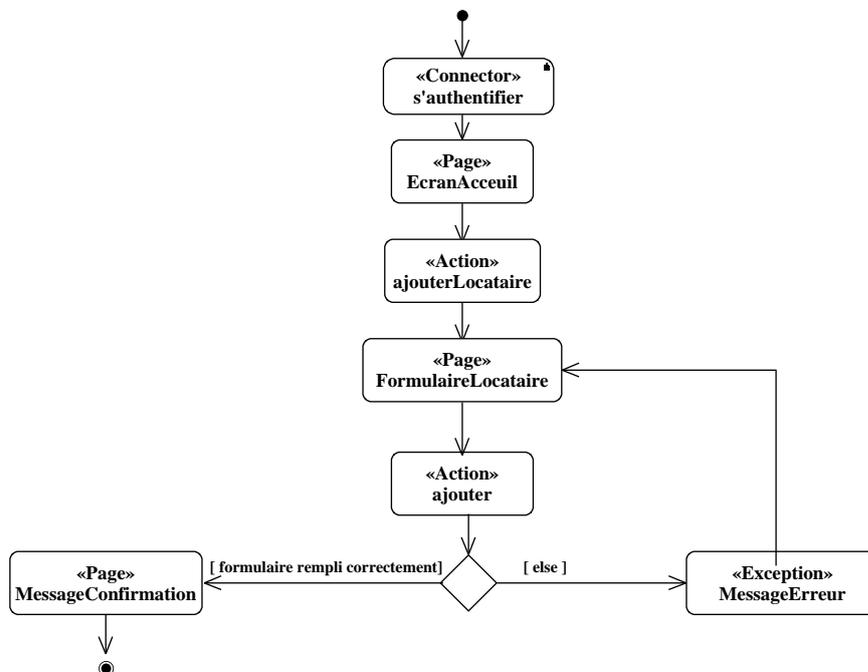


Figure 4.35: Diagramme d'activité de navigation du cas d'utilisation «ajouter un locataire»

4.4.12 Établir un contrat de location

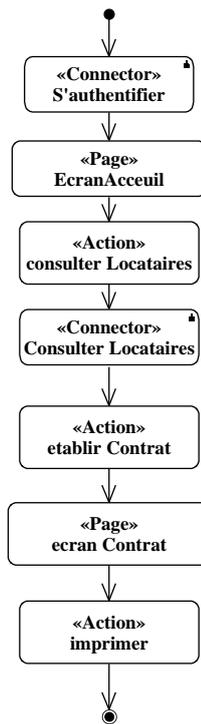


Figure 4.36: Diagramme d'activité de navigation du cas d'utilisation «établir un contrat de location»

4.4.13 Consulter les locataires

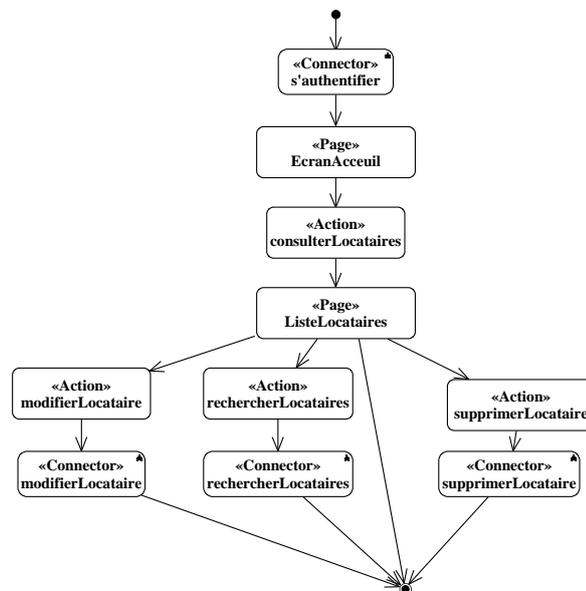


Figure 4.37: Diagramme d'activité de navigation du cas d'utilisation «consulter les locataires»

4.4.14 Rechercher des locataires

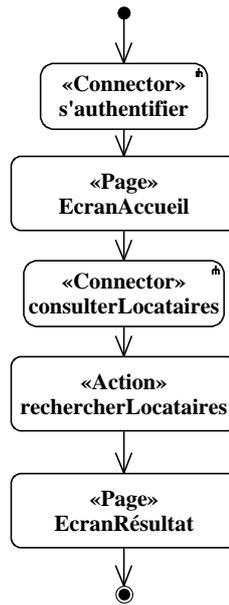


Figure 4.38: Diagramme d'activité de navigation du cas d'utilisation «rechercher les locataires»

4.4.15 Modifier un locataire

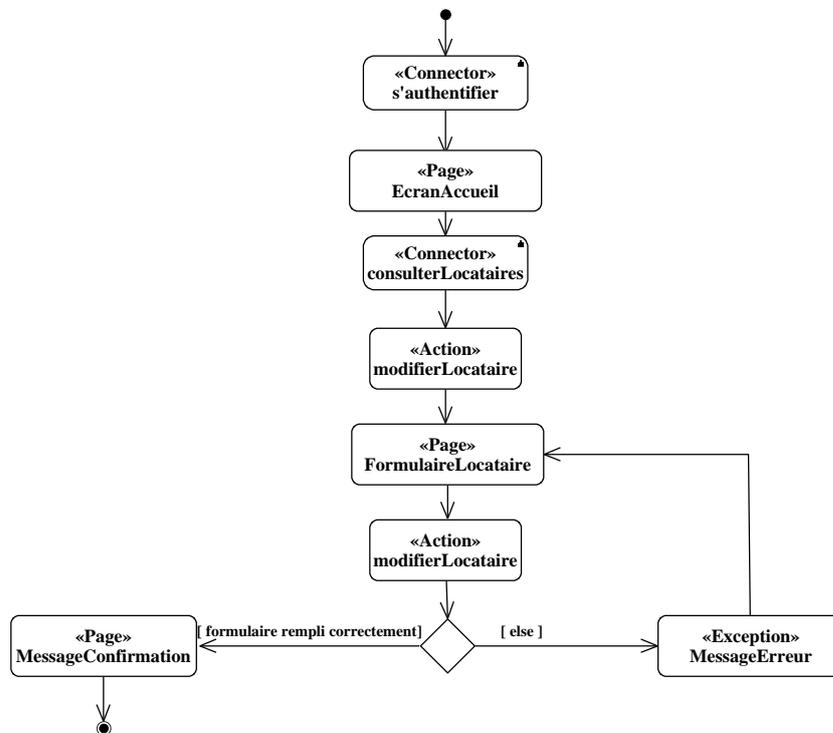


Figure 4.39: Diagramme d'activité de navigation du cas d'utilisation «modifier un locataire»

4.4.16 Supprimer un locataire

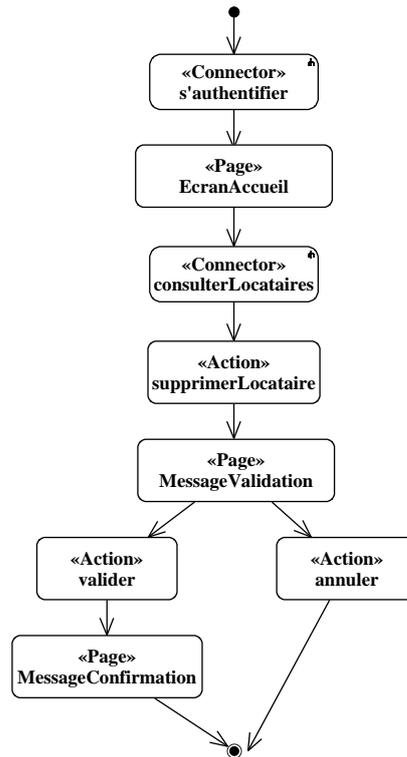


Figure 4.40: Diagramme d'activité de navigation du cas d'utilisation «supprimer un locataire»

4.4.17 Générer les loyers

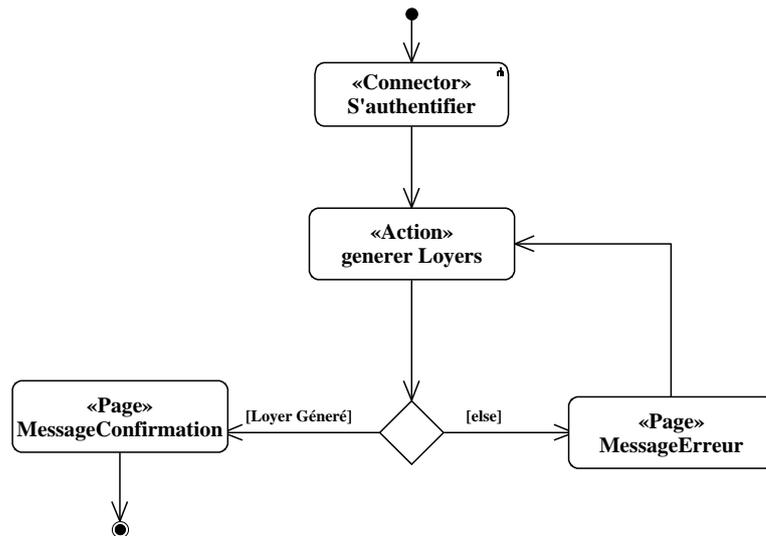


Figure 4.41: Diagramme d'activité de navigation du cas d'utilisation «générer les loyers»

4.4.18 Consulter les loyers

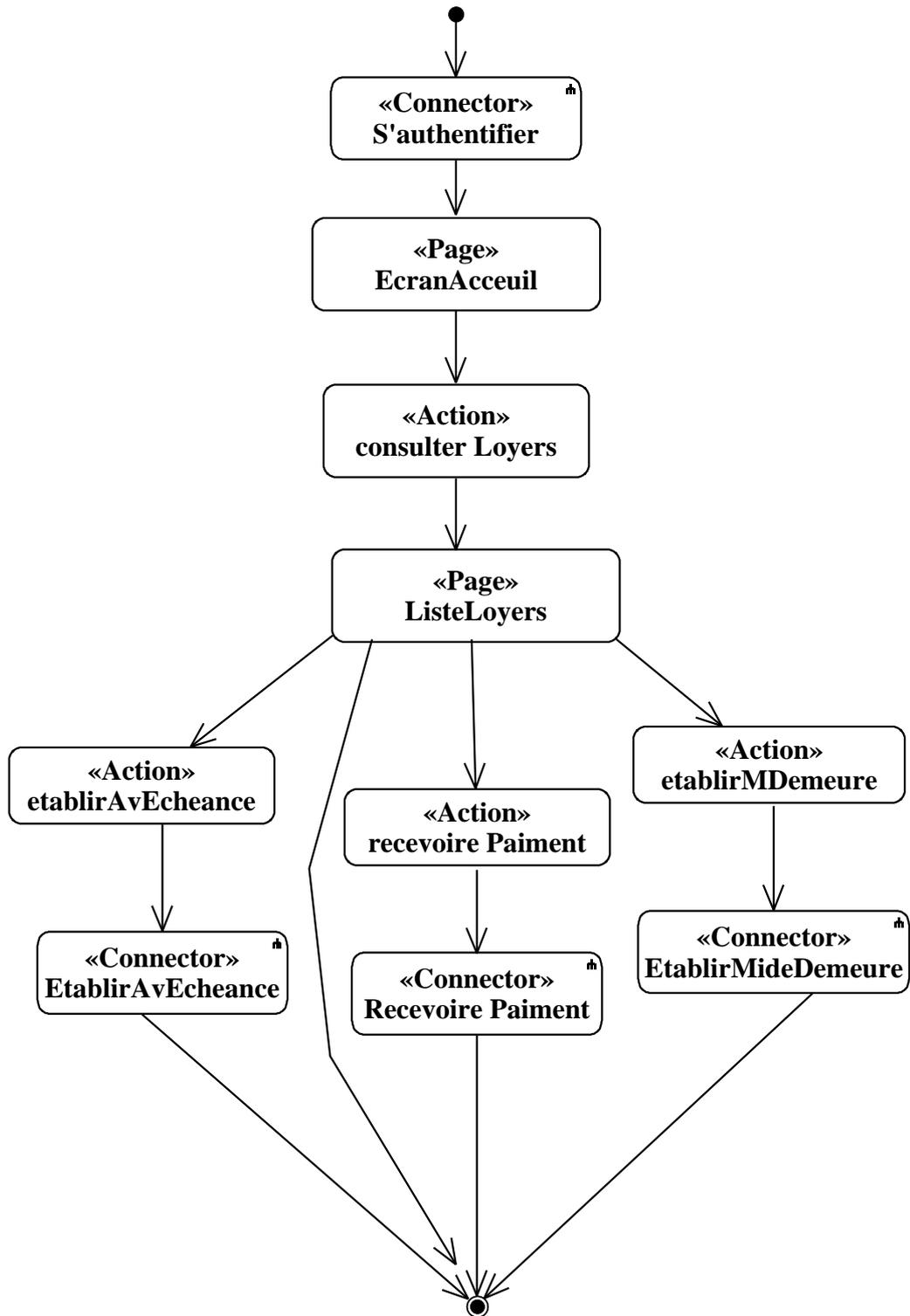


Figure 4.42: Diagramme d'activité de navigation du cas d'utilisation «consulter les loyers»

4.4.19 Recevoir un paiement

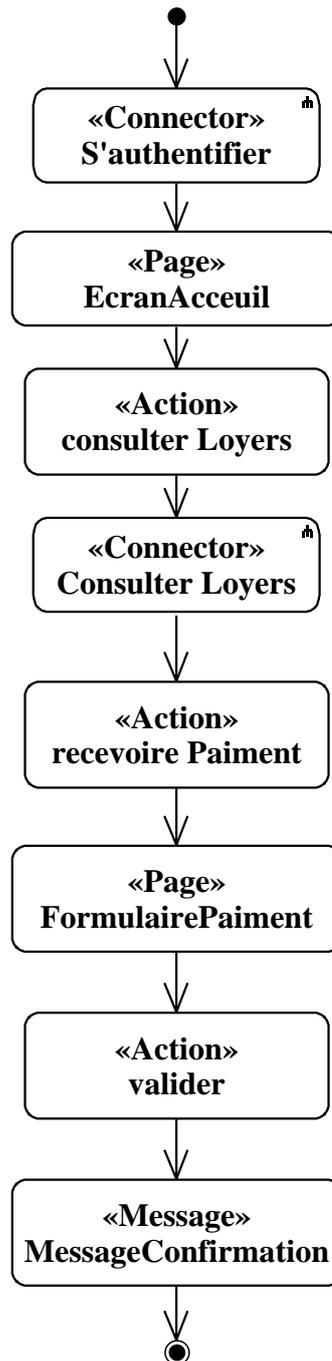


Figure 4.43: Diagramme d'activité de navigation du cas d'utilisation «recevoir un paiement»

4.4.20 Établir une quittance

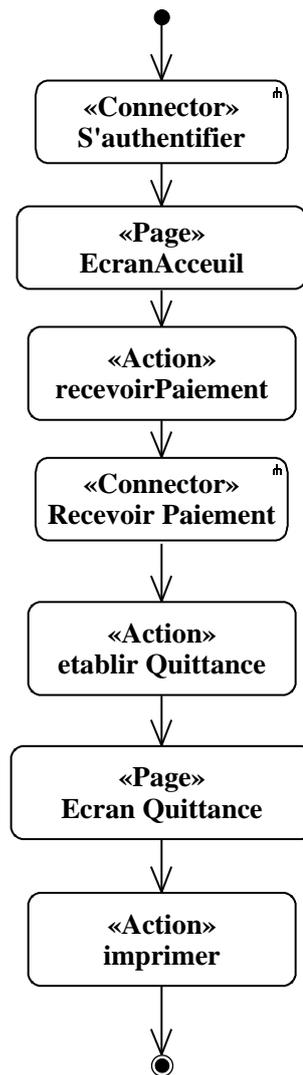


Figure 4.44: Diagramme d'activité de navigation du cas d'utilisation «établir une quittance»

4.4.21 Établir un avis d'échéance

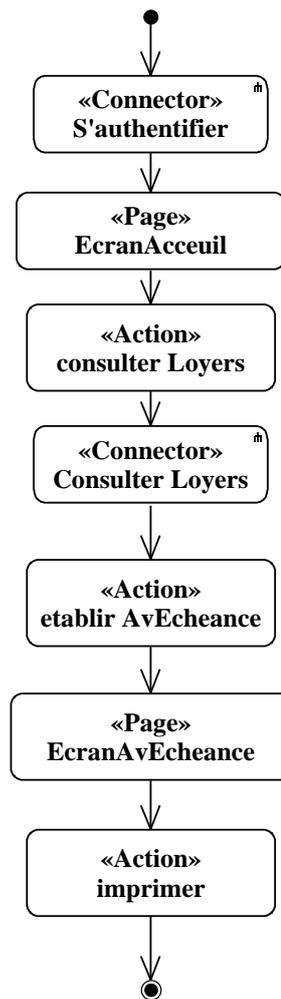


Figure 4.45: Diagramme d'activité de navigation du cas d'utilisation «établir un avis d'échéance»

4.4.22 Établir une mise en demeure

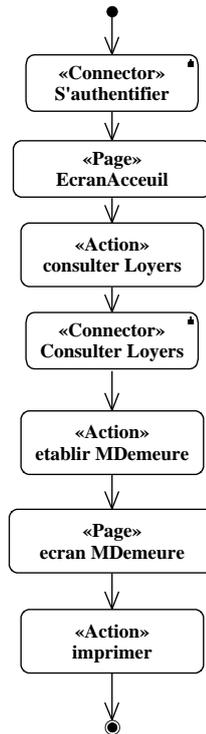


Figure 4.46: Diagramme d'activité de navigation du cas d'utilisation «établir une mise en demeure»

4.4.23 Consulter le tableau de bord

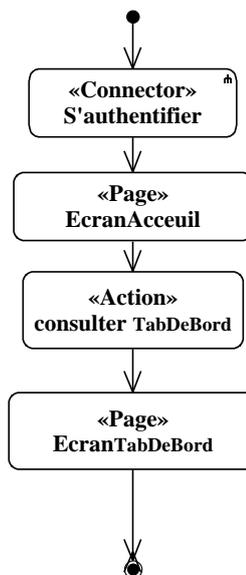


Figure 4.47: Diagramme d'activité de navigation du cas d'utilisation «consulter tableau de bord»

4.5 Conclusion

Dans ce chapitre, nous avons commencé par l'élaboration du modèle du domaine. Ensuite nous sommes passés à l'élaboration des diagrammes de classes participantes. Et nous avons fini par l'élaboration de diagrammes d'activité de navigation. Ce chapitre prépare la phase de conception qui est l'objet du chapitre suivant.

Chapitre 5

Phase de conception

5.1 Introduction

Dans ce chapitre nous allons élaborer les diagrammes de séquence qui nous permet d'attribuer précisément les responsabilités de comportement aux classes d'analyse du diagramme de classes participantes. Parallèlement, une première ébauche de la vue statique de conception, c'est-à-dire du diagramme de classes de conception, est construite et complétée.

5.2 Diagrammes de séquence

Le diagramme de séquences attribue précisément les responsabilités dégagées par les diagrammes de séquence système aux classes d'analyse du diagramme de classes participantes [20]. Dans la suite, nous réalisons les diagrammes de séquences correspondants aux cas d'utilisation, en se basant sur les diagrammes de séquences système, et les diagrammes de classes de conception.

5.2.1 S'authentifier

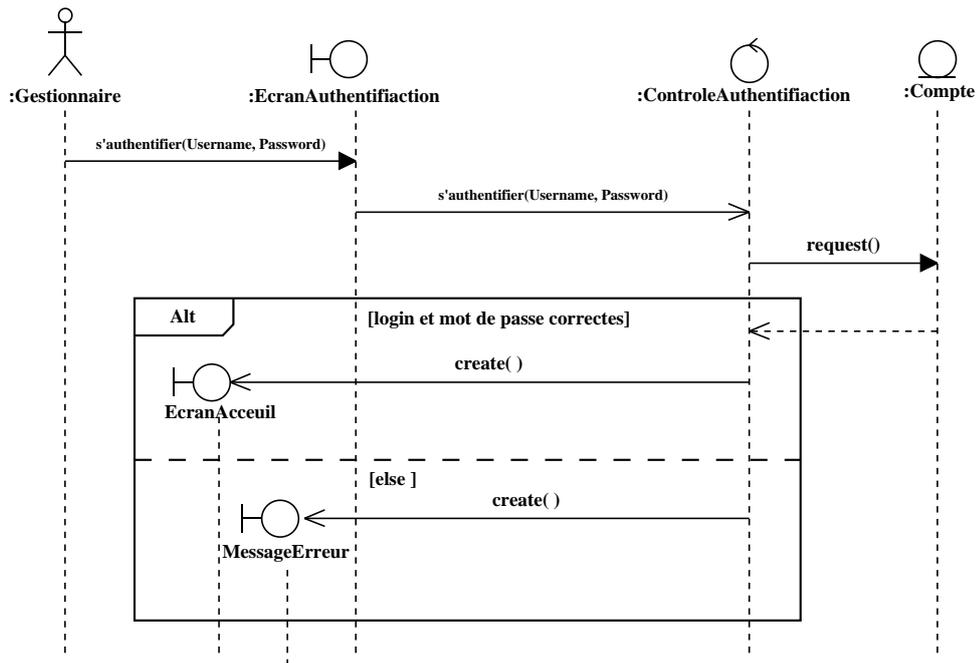


Figure 5.1: Diagramme de séquence du cas d'utilisation «s'authentifier»

5.2.2 Créer un compte

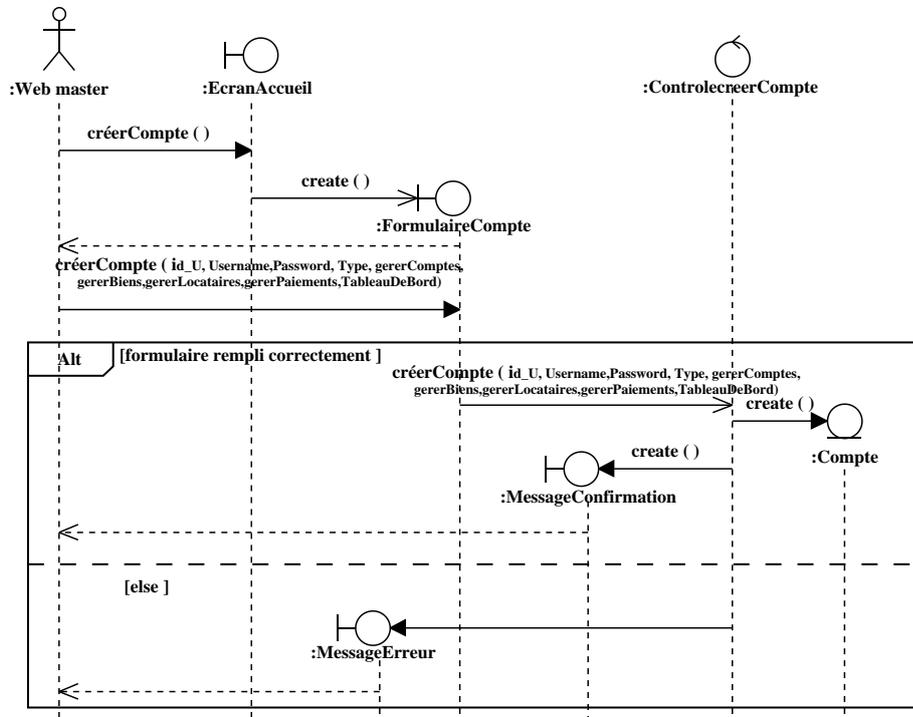


Figure 5.2: Diagramme de séquence du cas d'utilisation «créer un compte»

5.2.3 Consulter les comptes

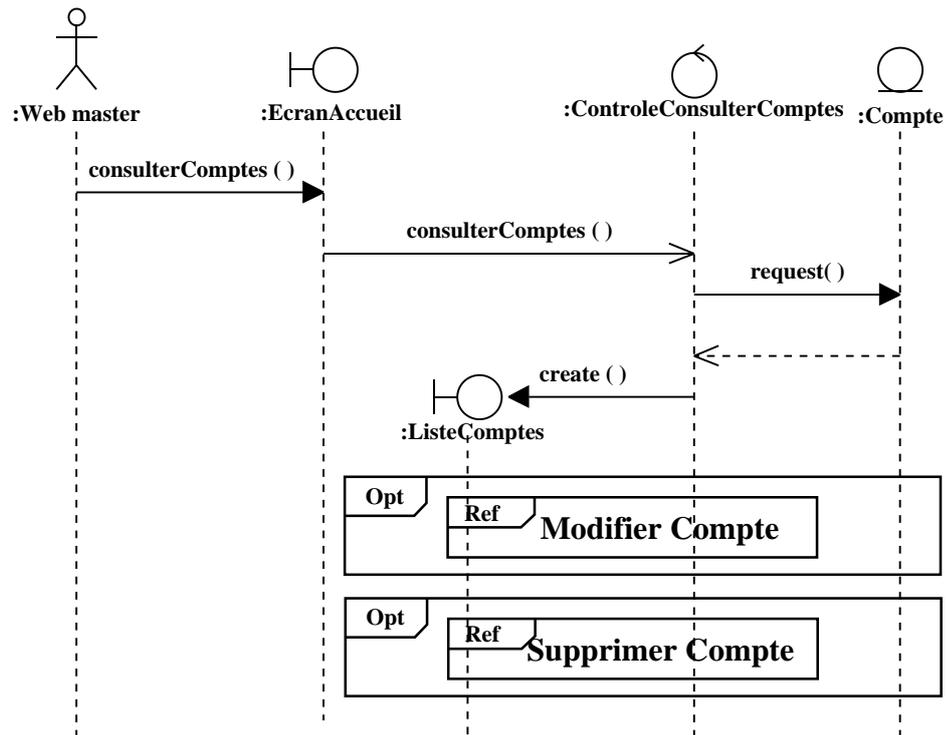


Figure 5.3: Diagramme de séquence du cas d'utilisation «consulter les comptes»

5.2.4 Modifier un compte

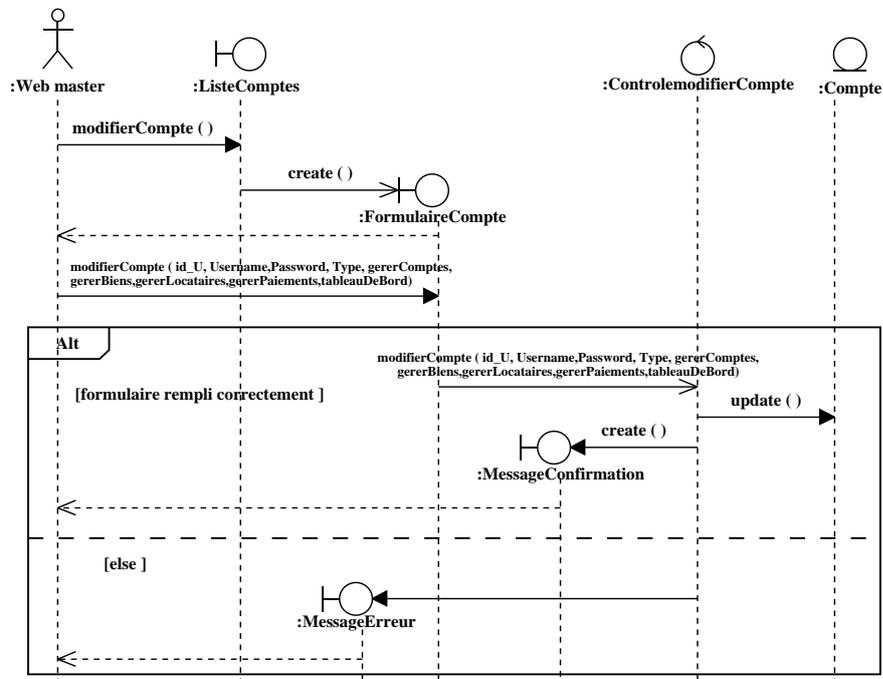


Figure 5.4: Diagramme de séquence du cas d'utilisation «modifier un compte»

5.2.5 Supprimer un compte

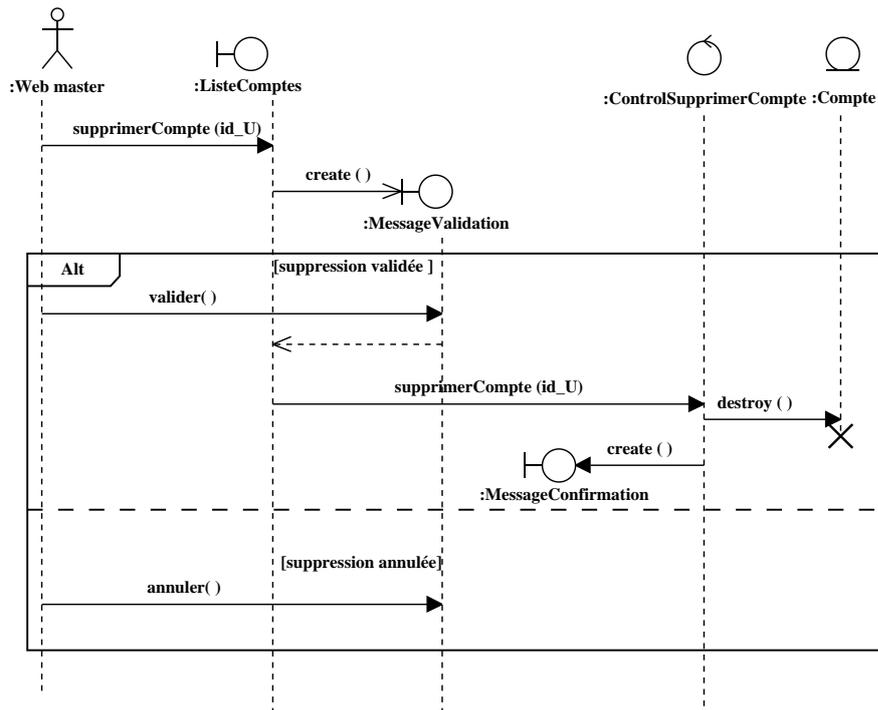


Figure 5.5: Diagramme de séquence du cas d'utilisation «supprimer un compte»

5.2.6 Ajouter un bien immobilier

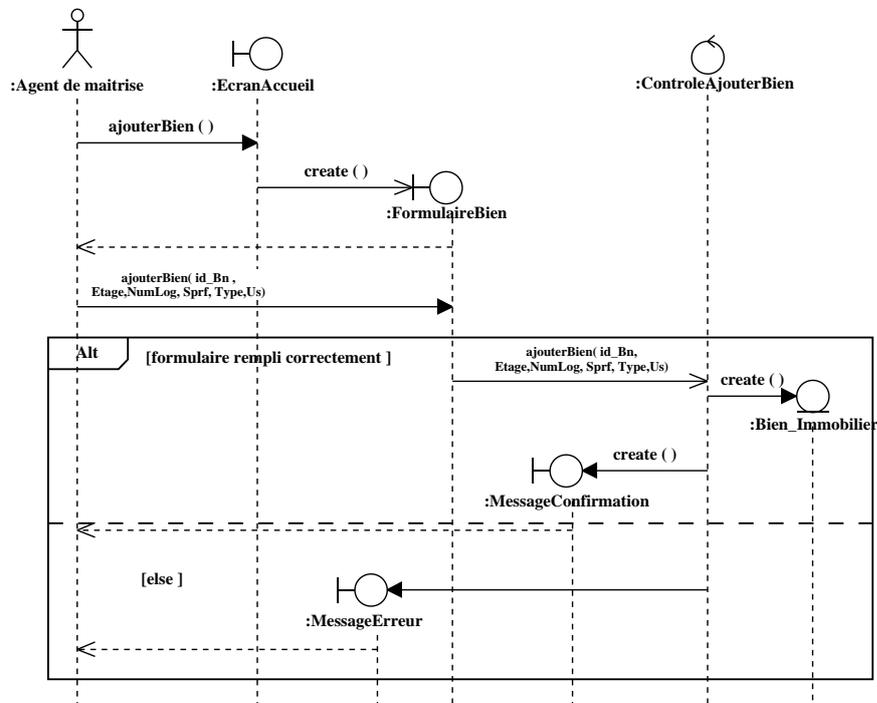


Figure 5.6: Diagramme de séquence du cas d'utilisation «ajouter un bien immobilier»

5.2.7 Consulter les biens immobiliers

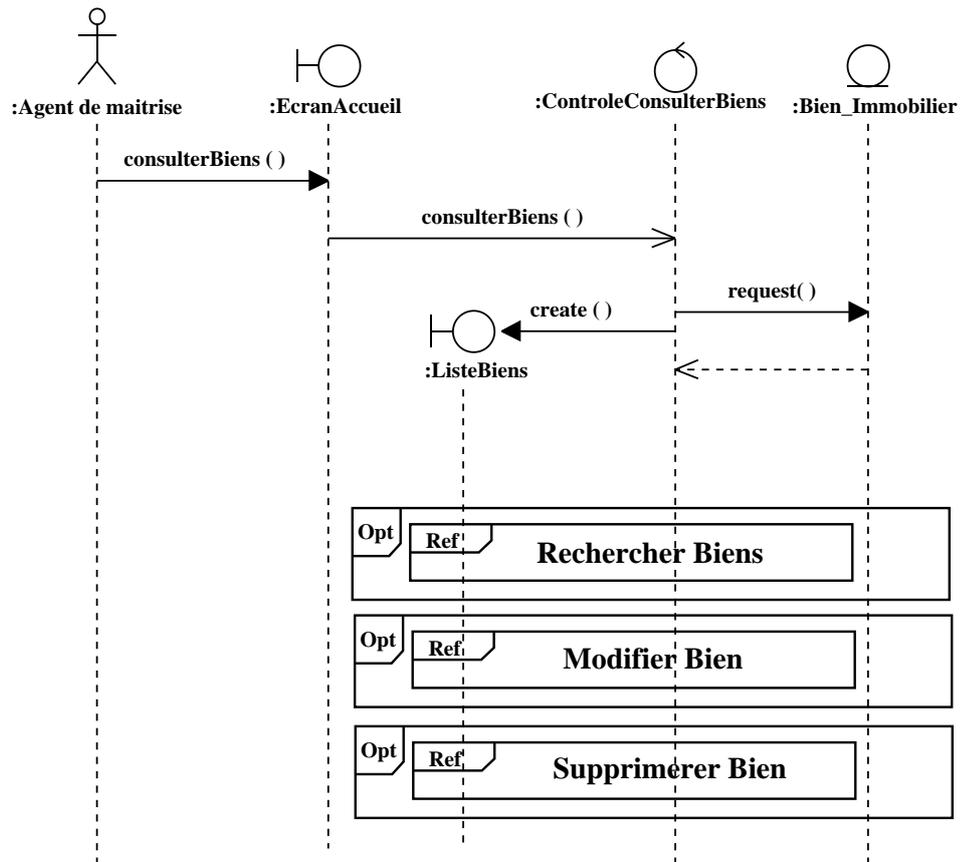


Figure 5.7: Diagramme de séquence du cas d'utilisation «consulter les biens immobiliers»

5.2.8 Rechercher les biens immobiliers

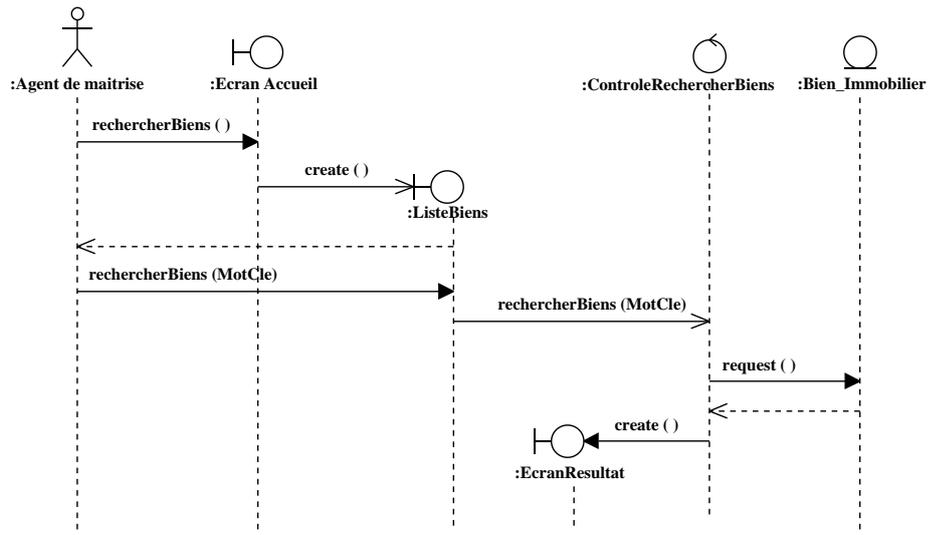


Figure 5.8: Diagramme de séquence du cas d'utilisation «rechercher les biens immobiliers»

5.2.9 Modifier un bien immobilier

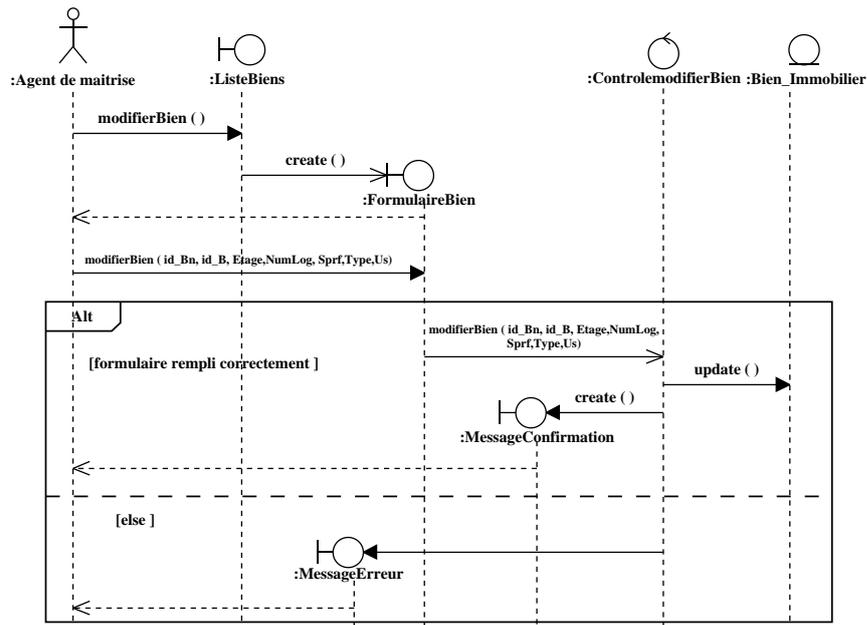


Figure 5.9: Diagramme de séquence du cas d'utilisation «modifier un bien immobiliers»

5.2.10 Supprimer un bien immobiliers

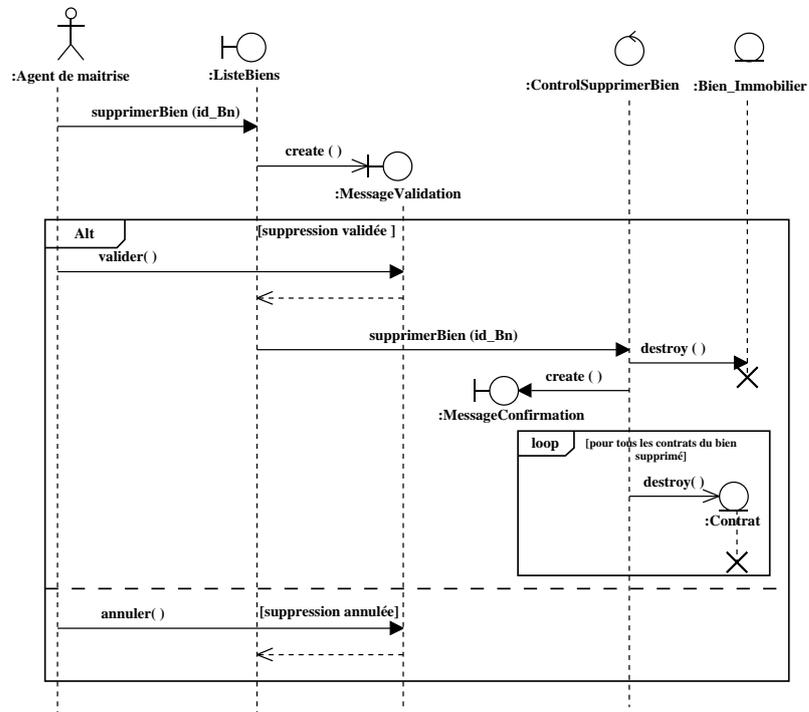


Figure 5.10: Diagramme de séquence du cas d'utilisation «supprimer un bien immobilier»

5.2.11 Ajouter un locataire

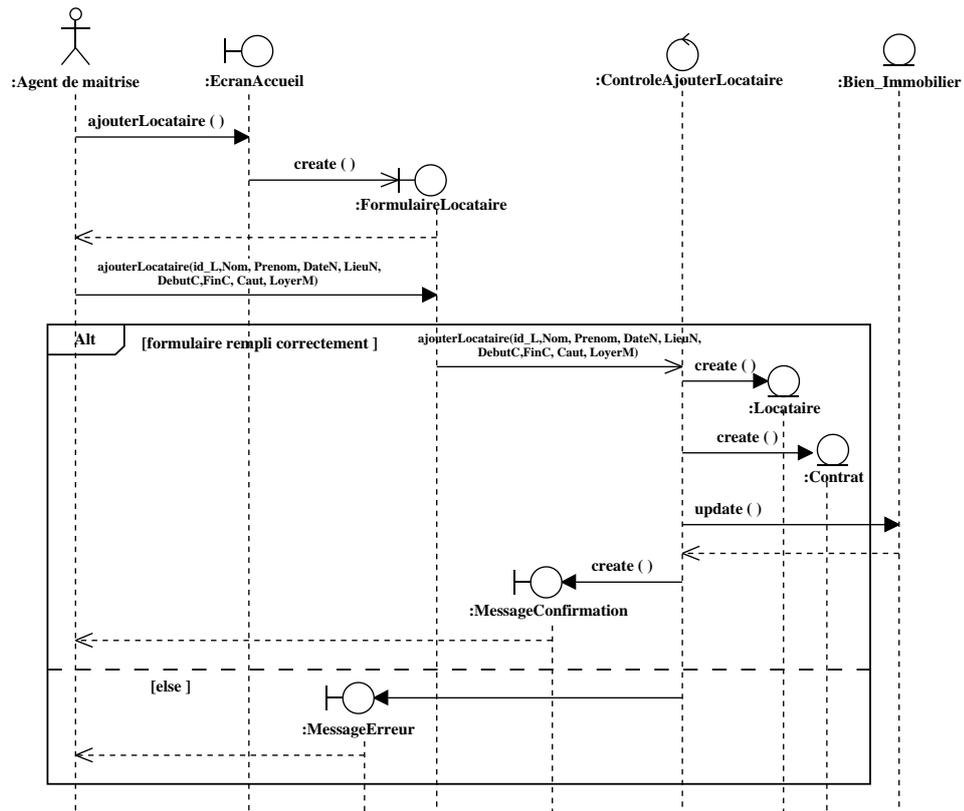


Figure 5.11: Diagramme de séquence du cas d'utilisation «ajouter un locataire»

5.2.12 Établir un contrat de location

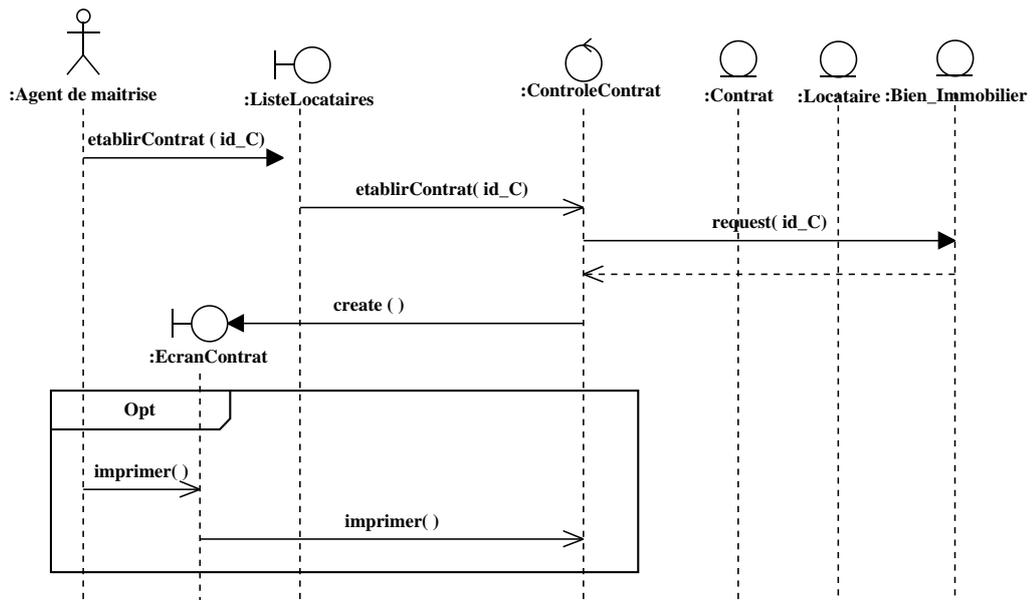


Figure 5.12: Diagramme de séquence du cas d'utilisation «établir un contrat de location»

5.2.13 Consulter les locataires

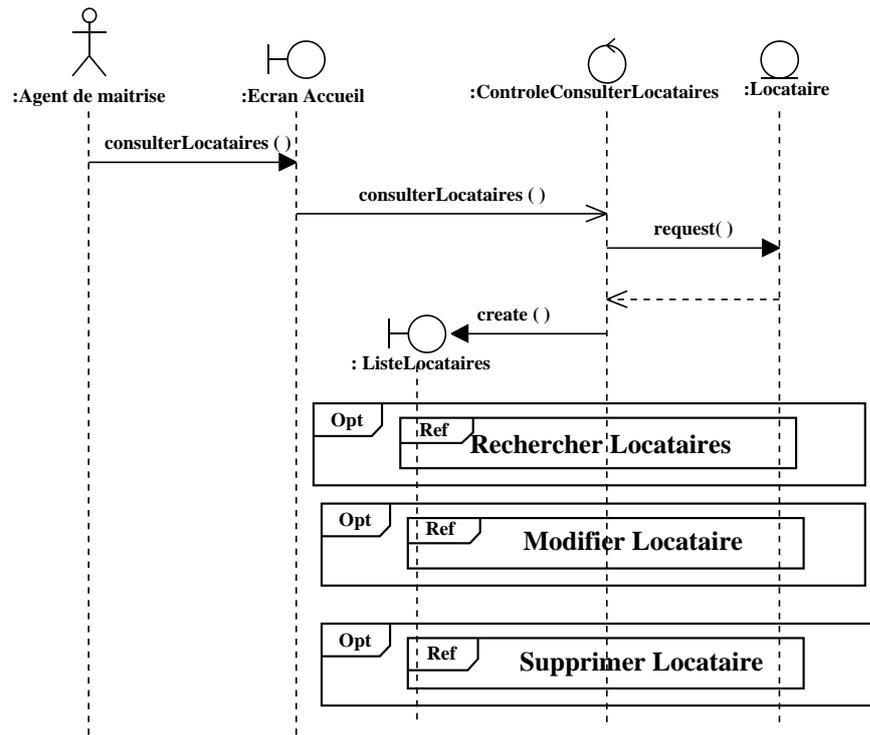


Figure 5.13: Diagramme de séquence du cas d'utilisation «consulter les locataires»

5.2.14 Rechercher les locataires

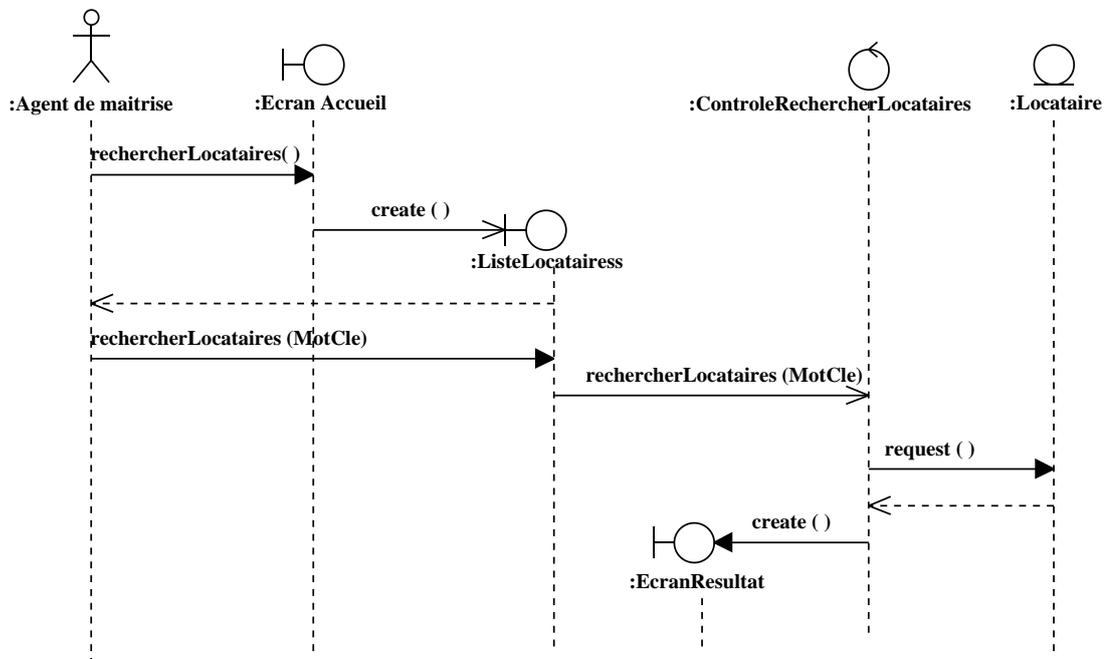


Figure 5.14: Diagramme de séquence du cas d'utilisation «rechercher les locataires»

5.2.15 Modifier un locataire

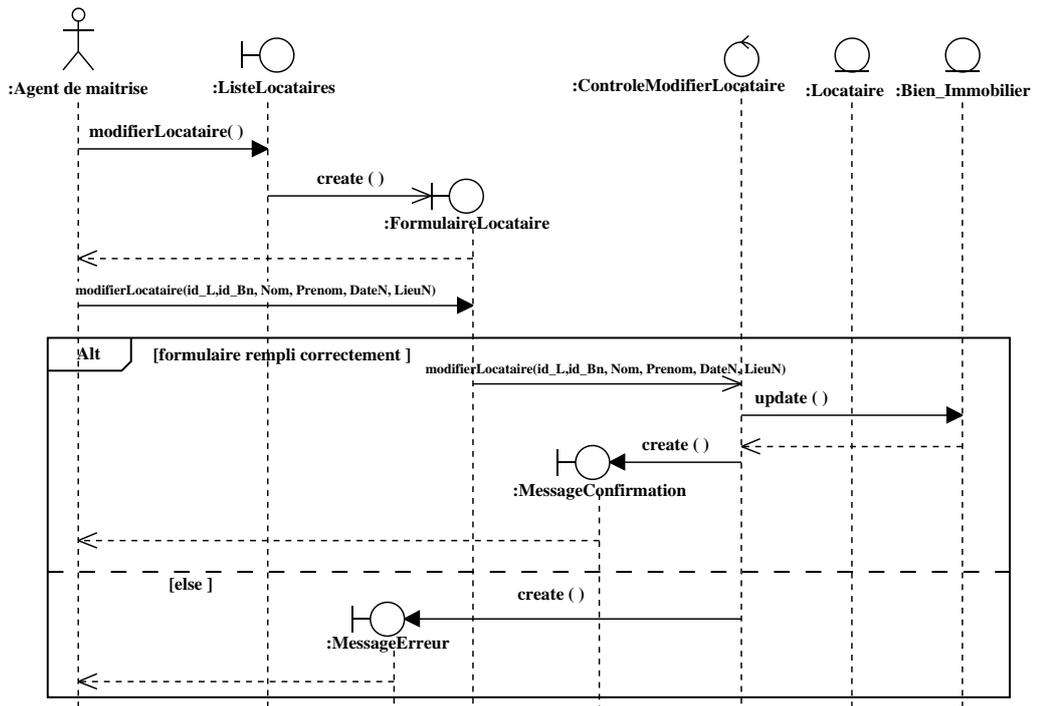


Figure 5.15: Diagramme de séquence du cas d'utilisation «modifier un locataire»

5.2.16 Supprimer un locataire

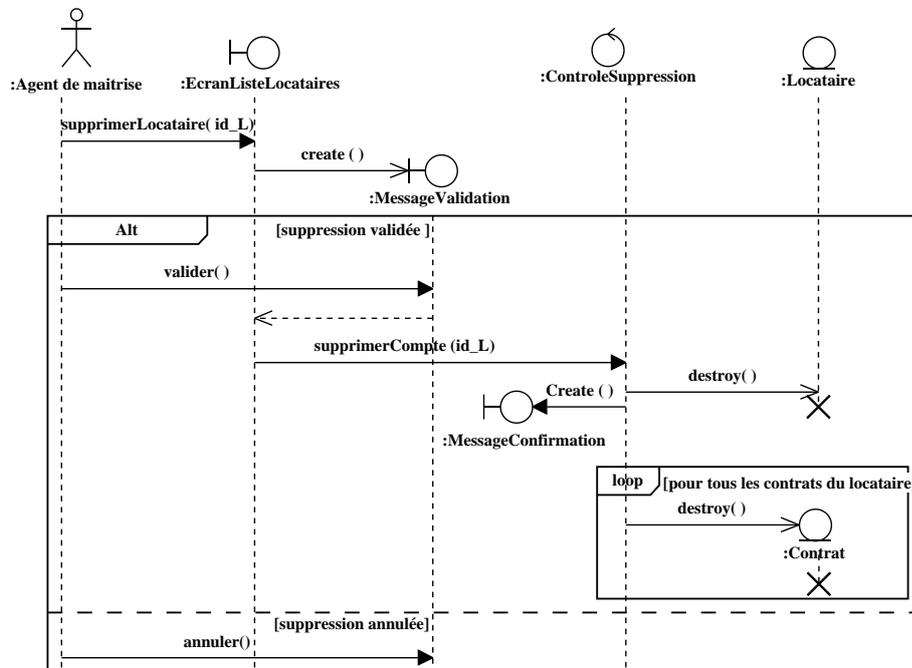


Figure 5.16: Diagramme de séquence du cas d'utilisation «supprimer un locataire»

5.2.17 Générer les loyers

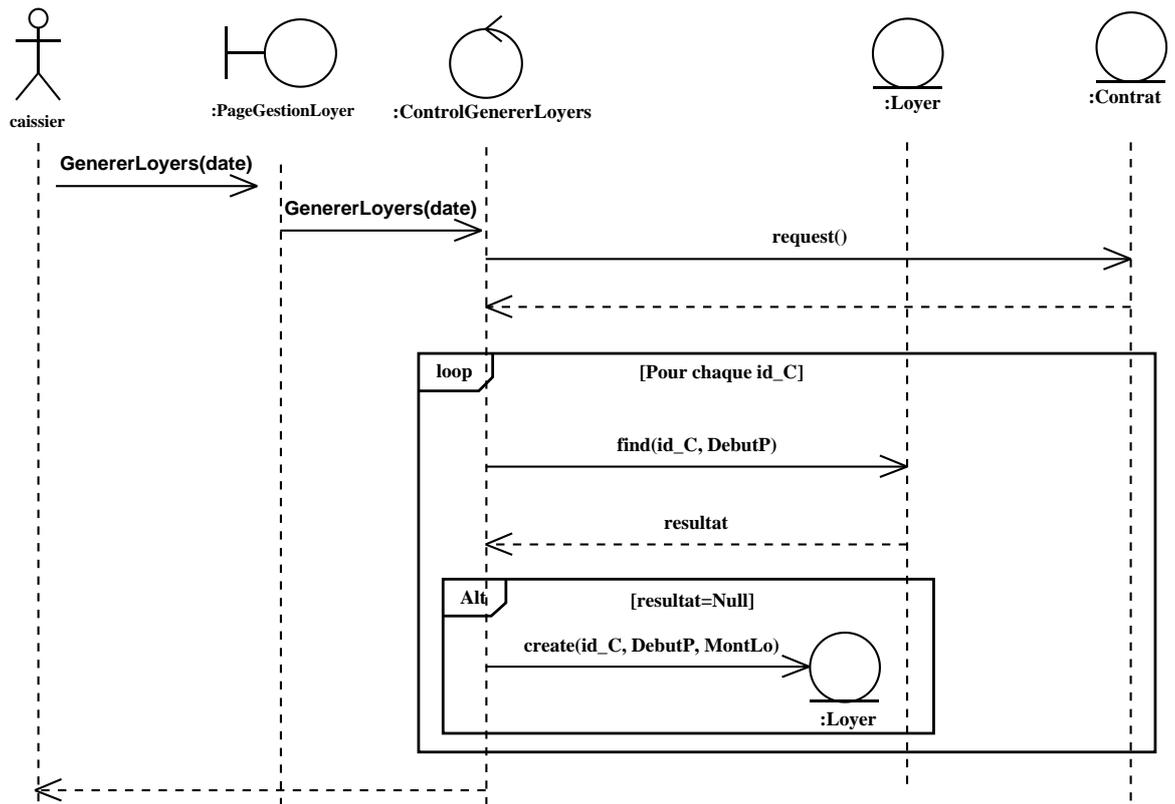


Figure 5.17: Diagramme de séquence du cas d'utilisation «générer les loyers»

5.2.18 Consulter les loyers

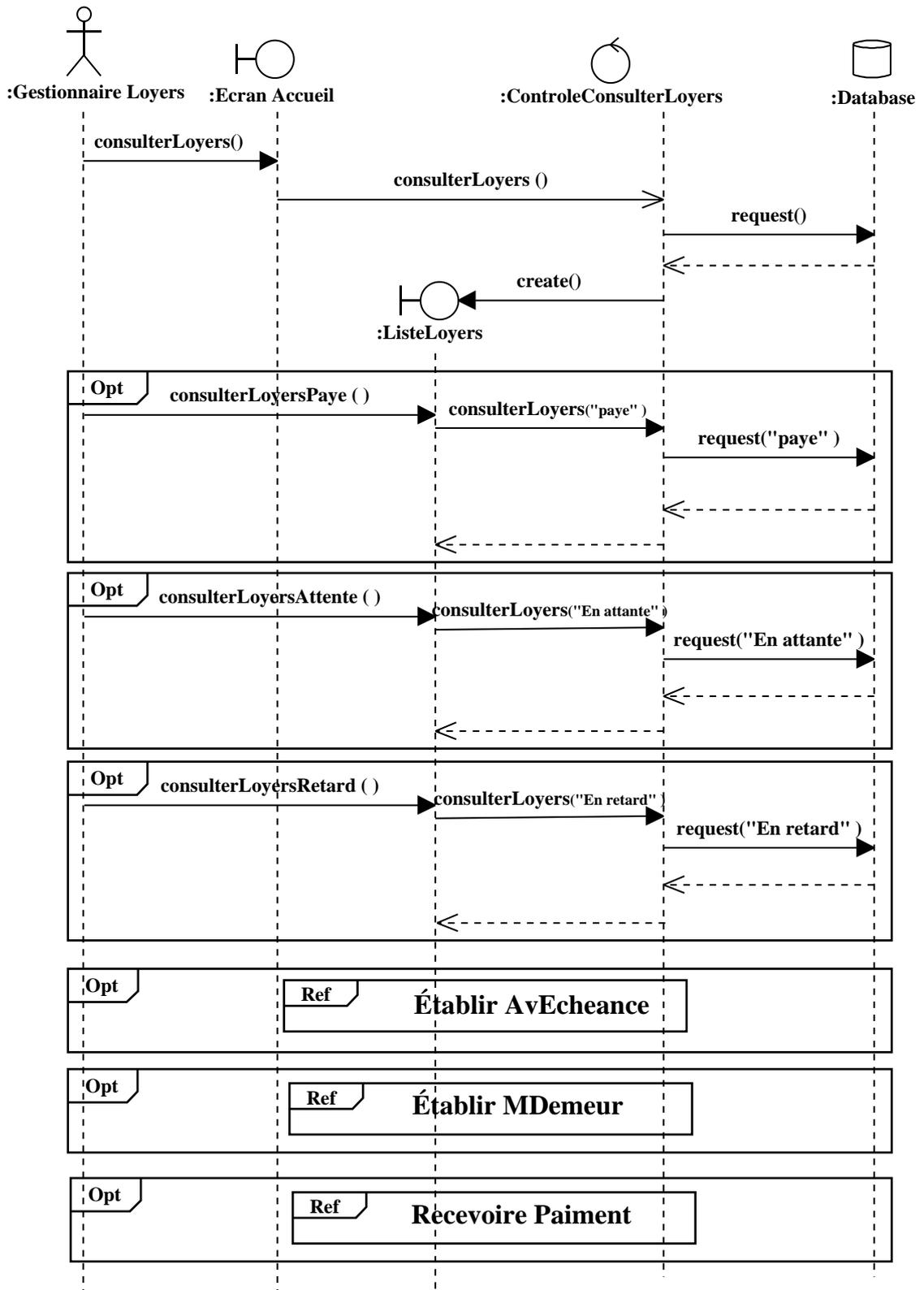


Figure 5.18: Diagramme de séquence du cas d'utilisation «consulter les loyers»

5.2.19 Recevoir un paiement

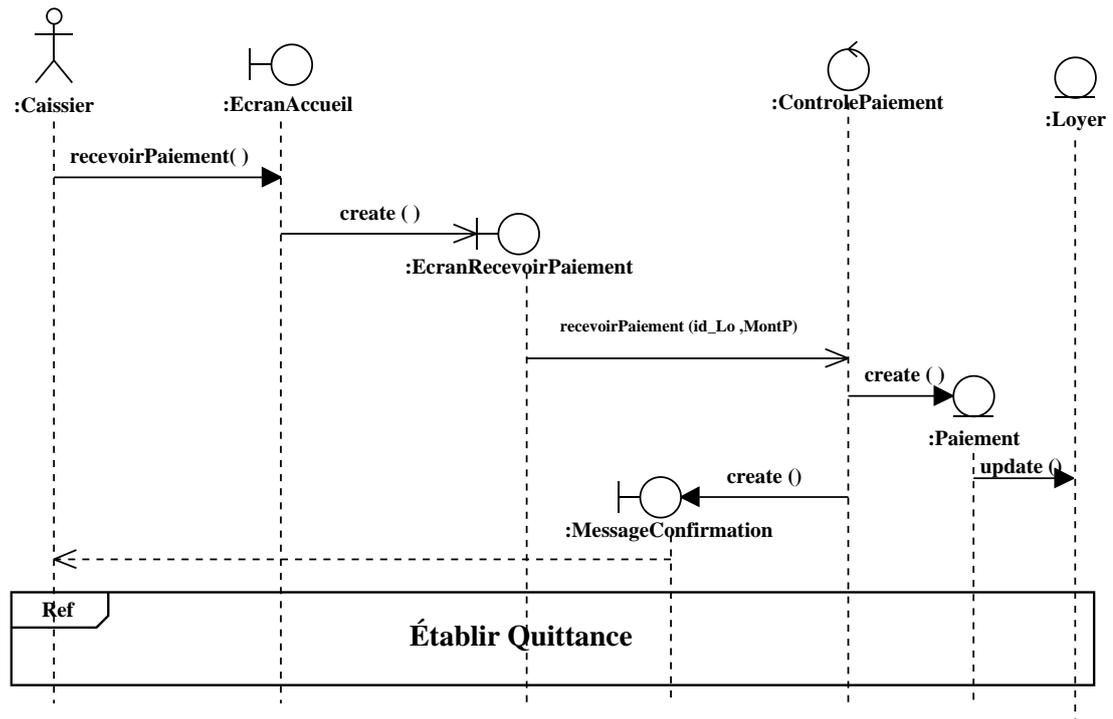


Figure 5.19: Diagramme de séquence du cas d'utilisation «recevoir un paiement»

5.2.20 Établir une quittance

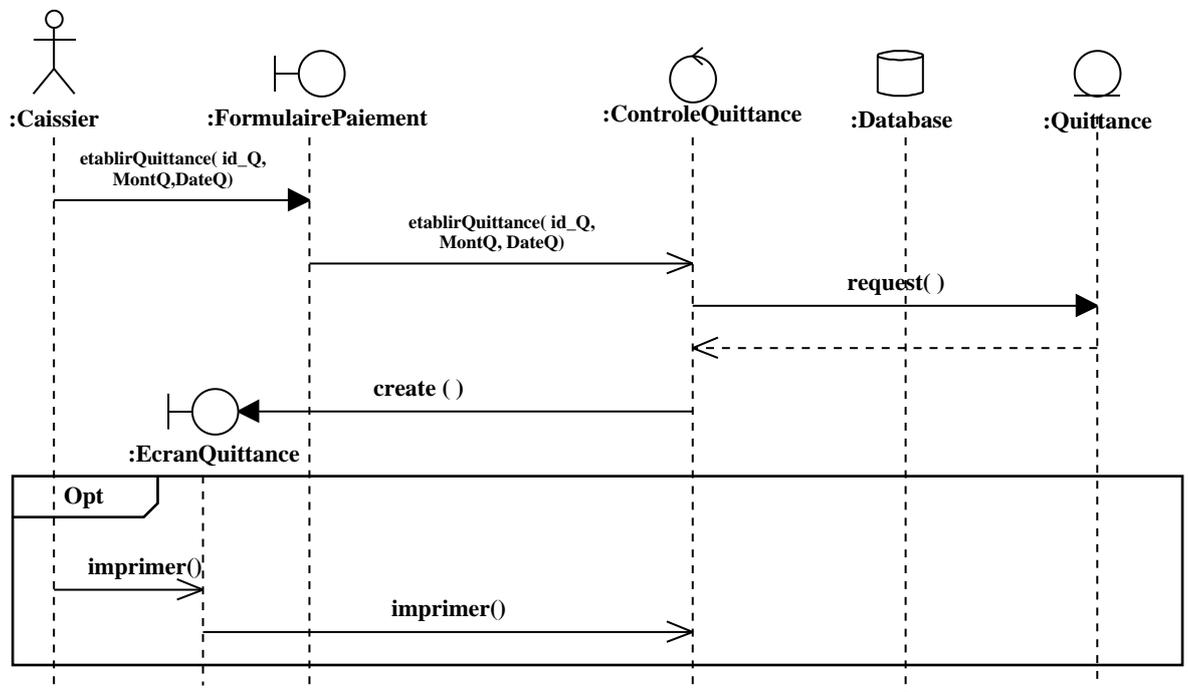


Figure 5.20: Diagramme de séquence du cas d'utilisation «établir une quittance»

5.2.21 Établir un avis d'échéance

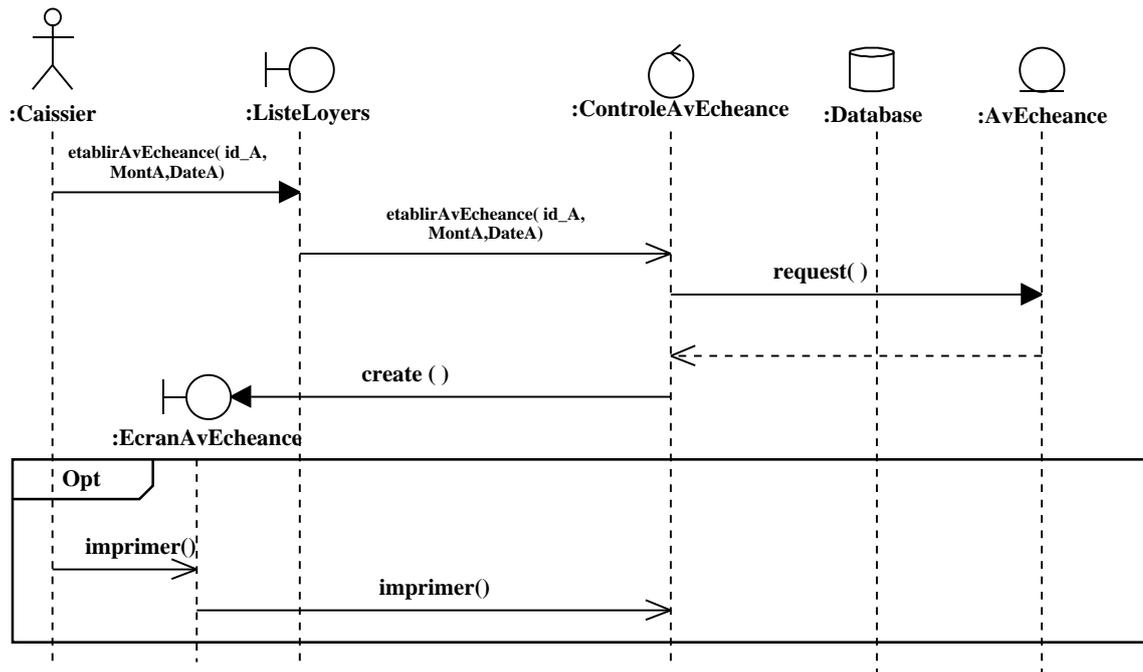


Figure 5.21: Diagramme de séquence du cas d'utilisation «établir un avis d'échéance»

5.2.22 Établir une mise en demeure

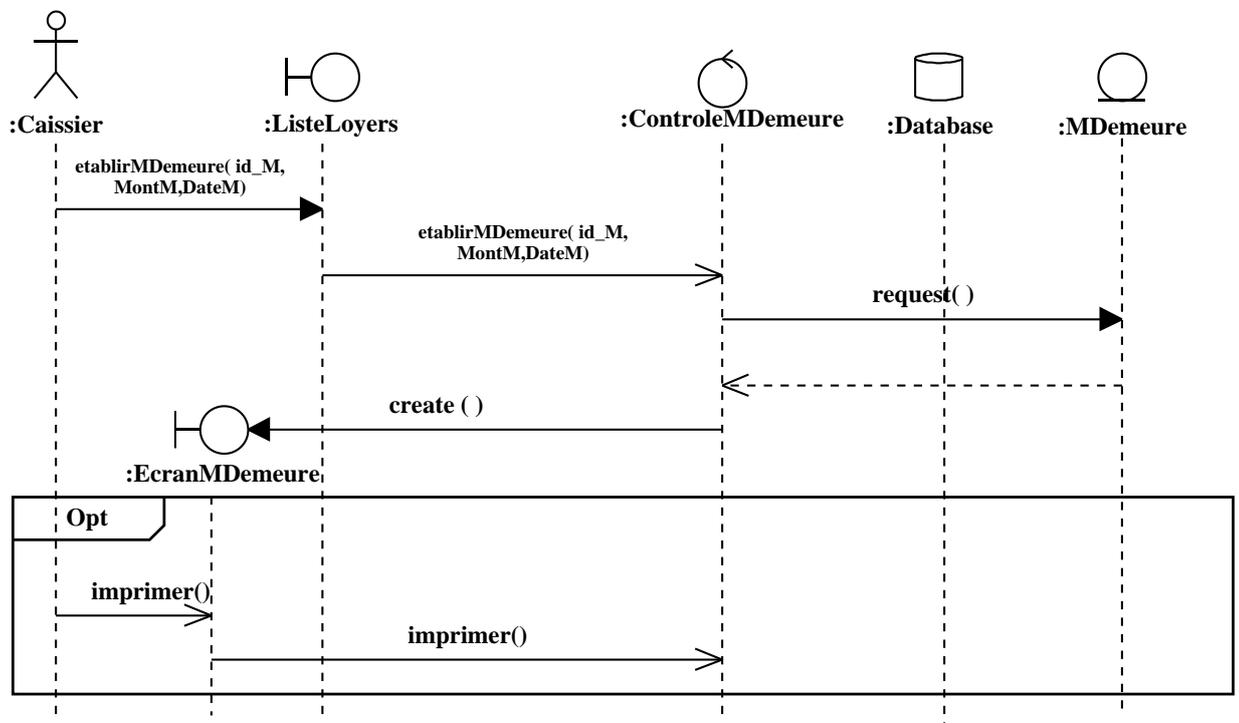


Figure 5.22: Diagramme de séquence du cas d'utilisation «établir une mise en demeure»

5.2.23 Consulter le tableau de bord

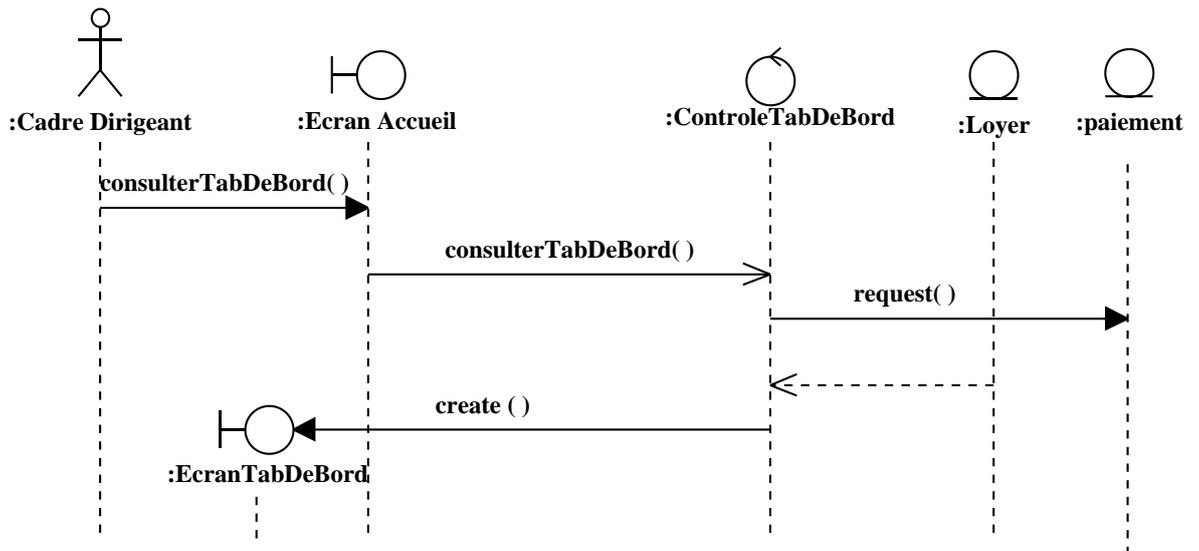


Figure 5.23: Diagramme de séquence du cas d'utilisation «consulter le tableau de bord»

5.3 Diagramme de classes de conceptions

L'objectif du diagramme de classes de conception est de servir de guide pour l'implémentation. Une première ébauche du diagramme de classes de conception a déjà été élaborée en parallèle des diagrammes d'interaction. Il faut maintenant le compléter en précisant les opérations privées des différentes classes. Il faut prendre en compte les choix techniques, comme le choix du langage de programmation, le choix des différentes bibliothèques utilisées (notamment pour l'implémentation de l'interface graphique), etc.

5.3.1 S'authentifier

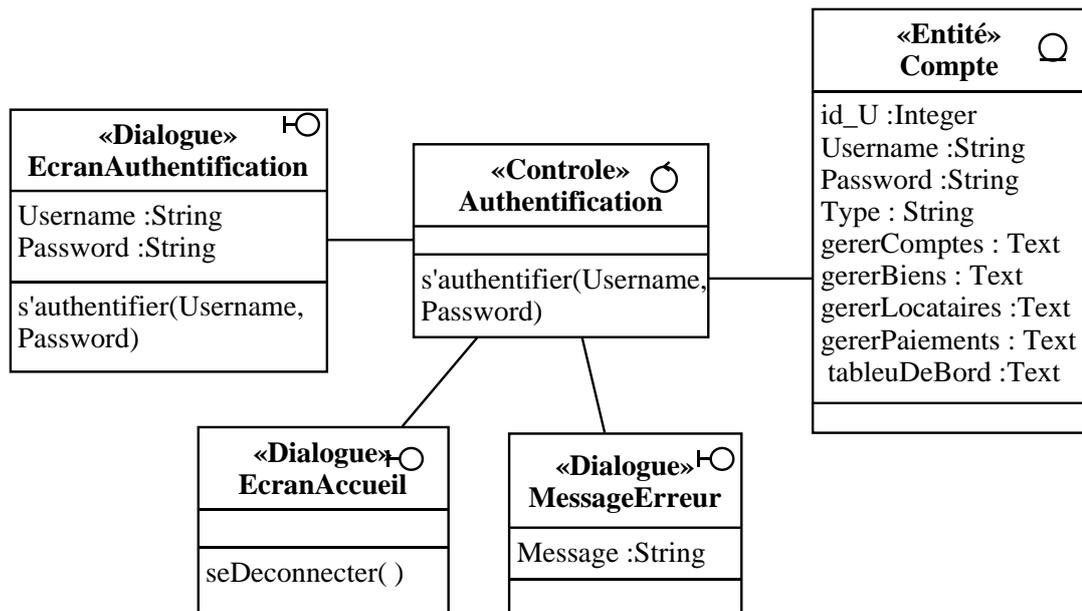


Figure 5.24: Diagramme de classes de conception du cas d'utilisation «s'authentifier»

5.3.2 Créer un compte

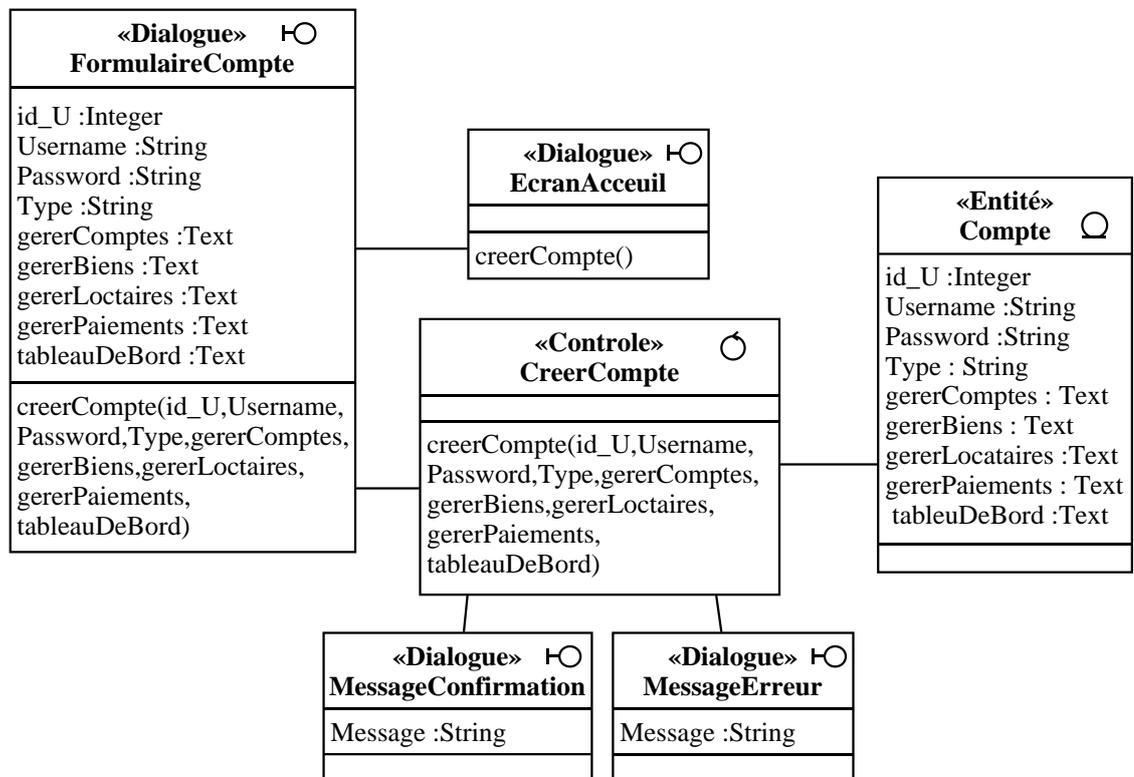


Figure 5.25: Diagramme de classes de conception du cas d'utilisation «créer un compte»

5.3.3 Consulter les comptes

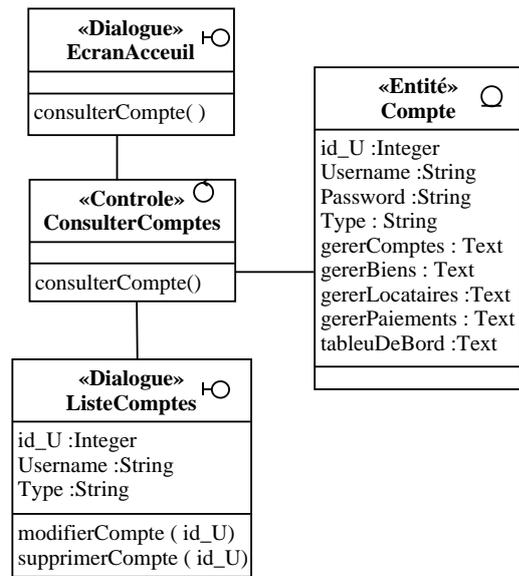


Figure 5.26: Diagramme de classes de conception du cas d'utilisation «consulter les comptes»

5.3.4 Modifier un compte

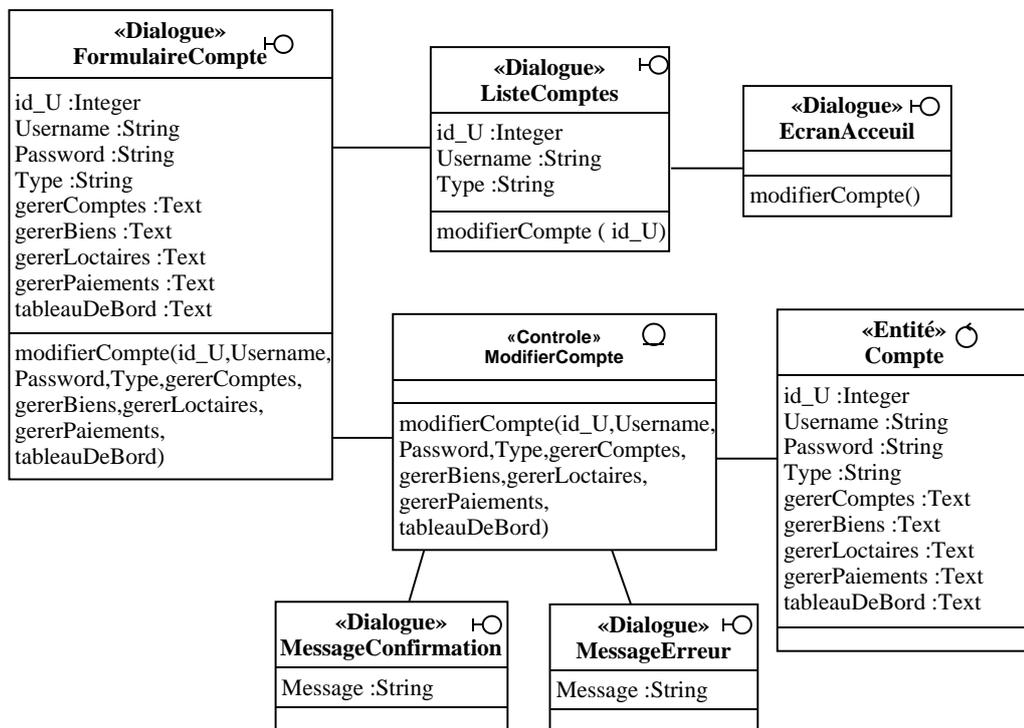


Figure 5.27: Diagramme de classes de conception du cas d'utilisation «modifier un compte»

5.3.5 Supprimer un compte

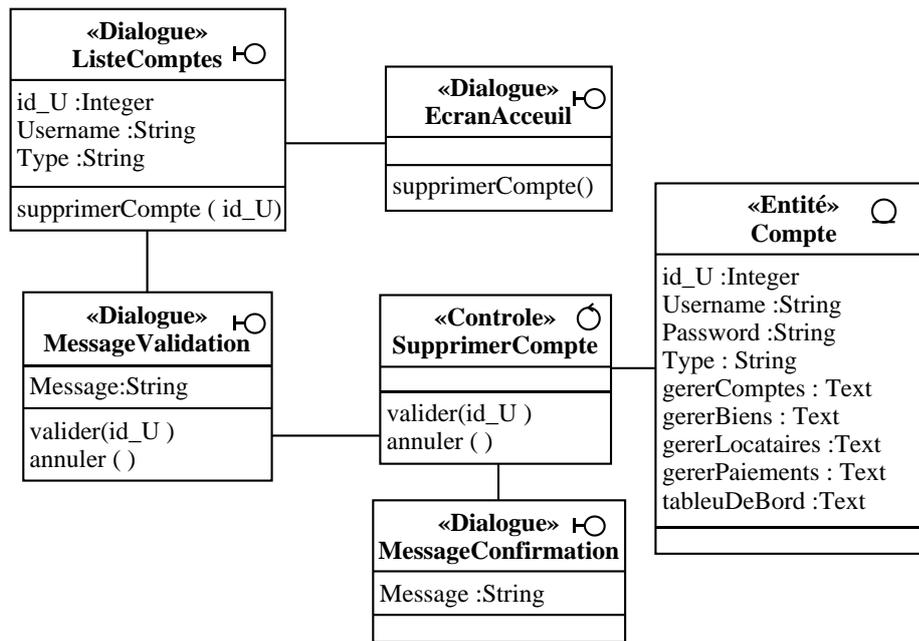


Figure 5.28: Diagramme de classes de conception du cas d'utilisation «supprimer un compte»

5.3.6 Ajouter un bien immobilier

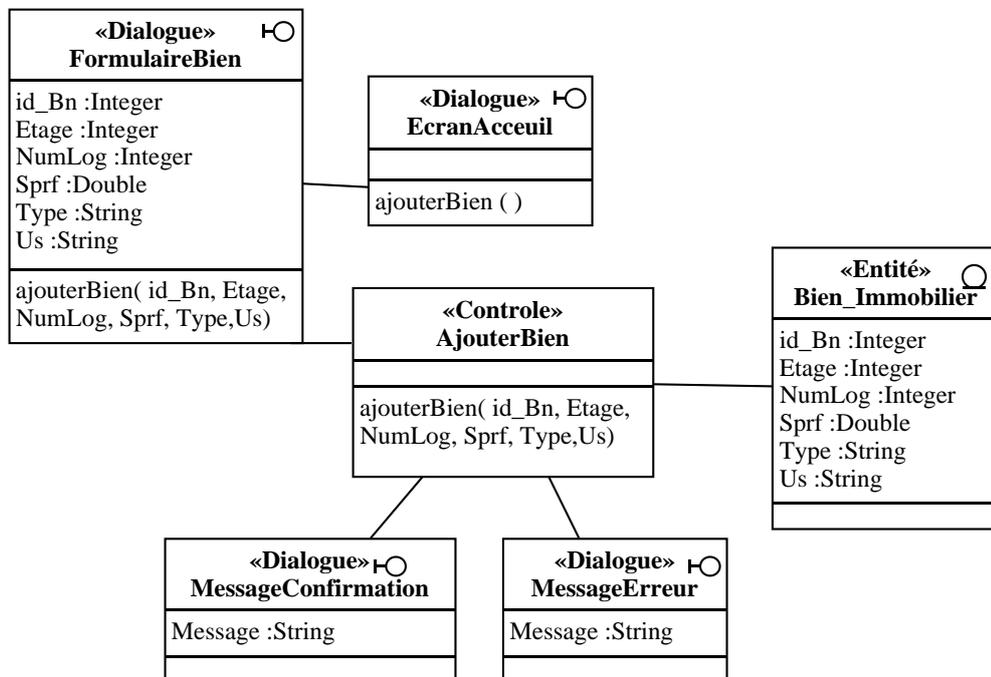


Figure 5.29: Diagramme de classes de conception du cas d'utilisation «ajouter un bien immobilier»

5.3.7 Consulter les biens immobiliers

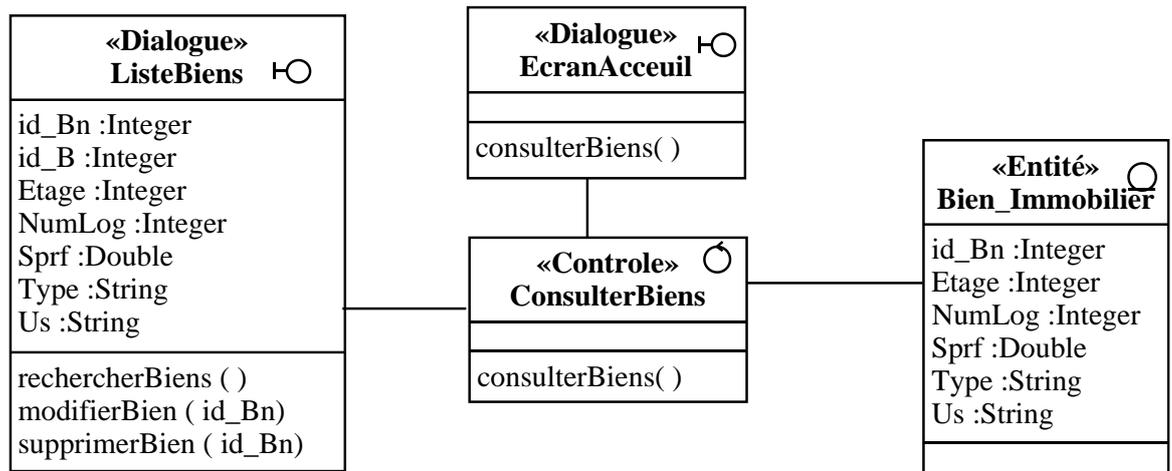


Figure 5.30: Diagramme de classes de conception du cas d'utilisation «consulter les biens immobiliers»

5.3.8 Rechercher les biens immobiliers

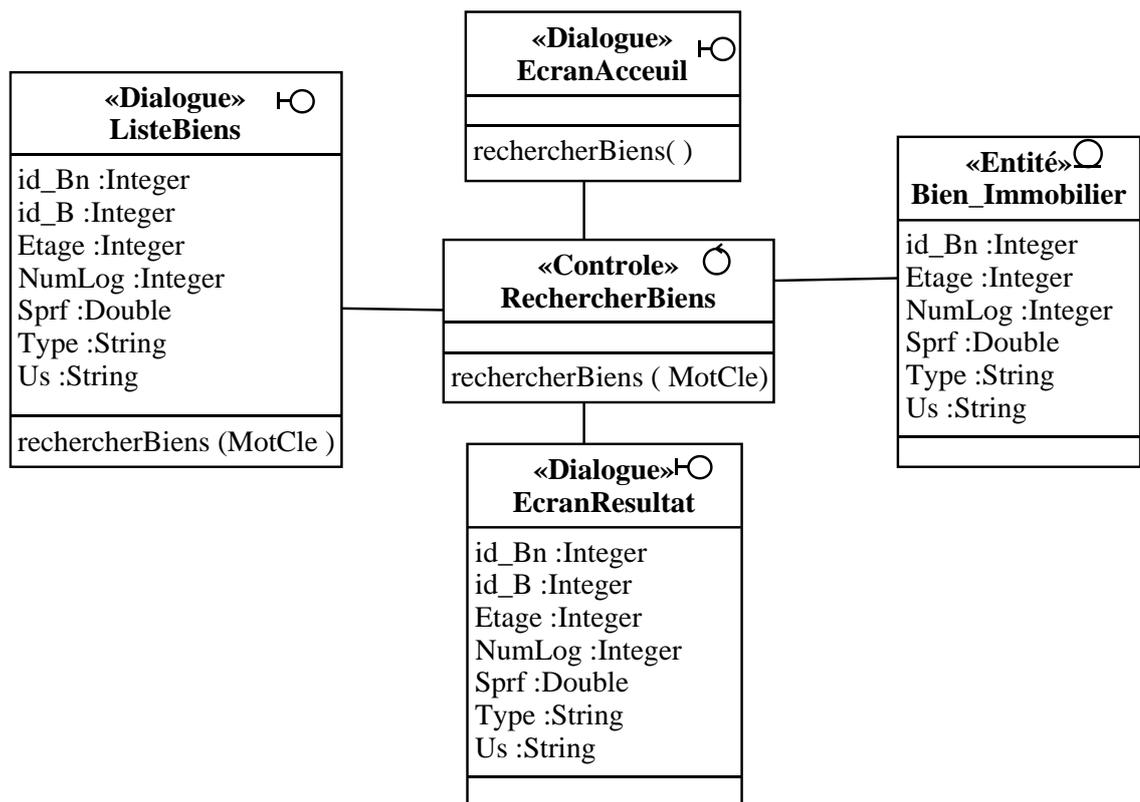


Figure 5.31: Diagramme de classes de conception e du cas d'utilisation «rechercher les biens immobiliers»

5.3.9 Modifier un bien immobilier

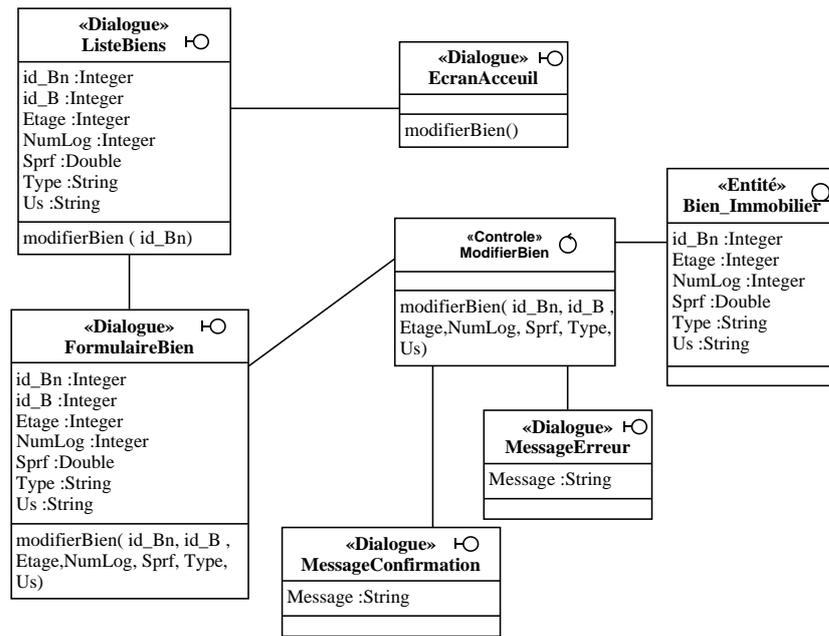


Figure 5.32: Diagramme de classes de conception du cas d'utilisation «modifier un bien immobilier»

5.3.10 Supprimer un bien immobilier

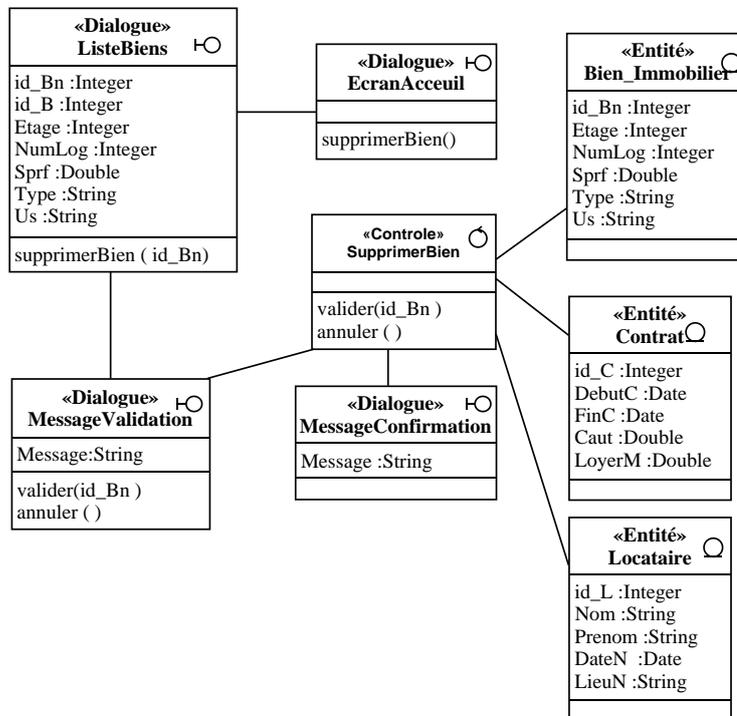


Figure 5.33: Diagramme de classes de conception du cas d'utilisation «supprimer un bien immobilier»

5.3.11 Ajouter un locataire

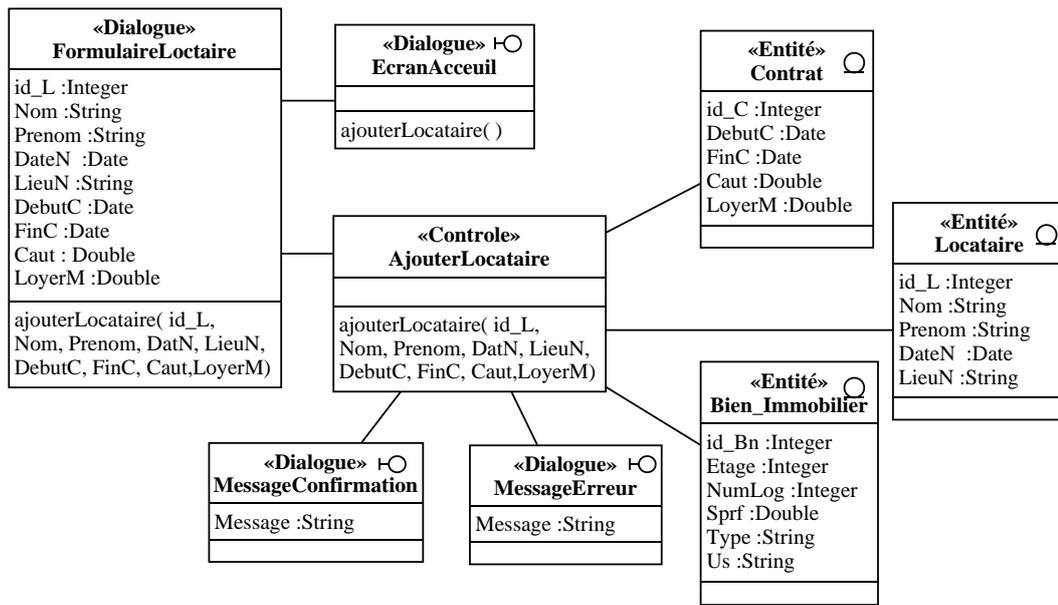


Figure 5.34: Diagramme de classes de conception du cas d'utilisation «ajouter un locataire»

5.3.12 Établir un contrat de location

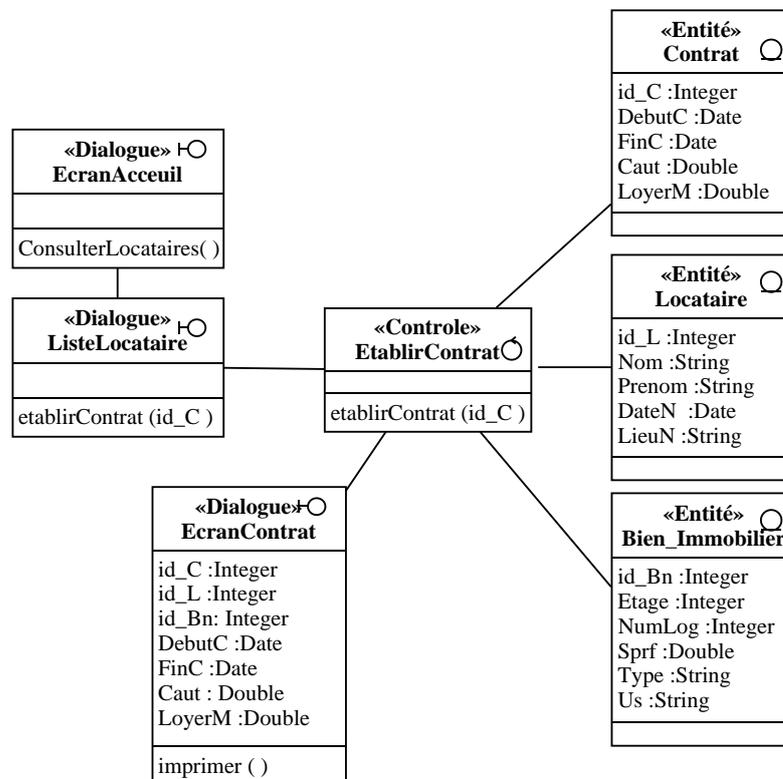


Figure 5.35: Diagramme de classes de conception du cas d'utilisation «établir un contrat de location»

5.3.13 Consulter les locataires

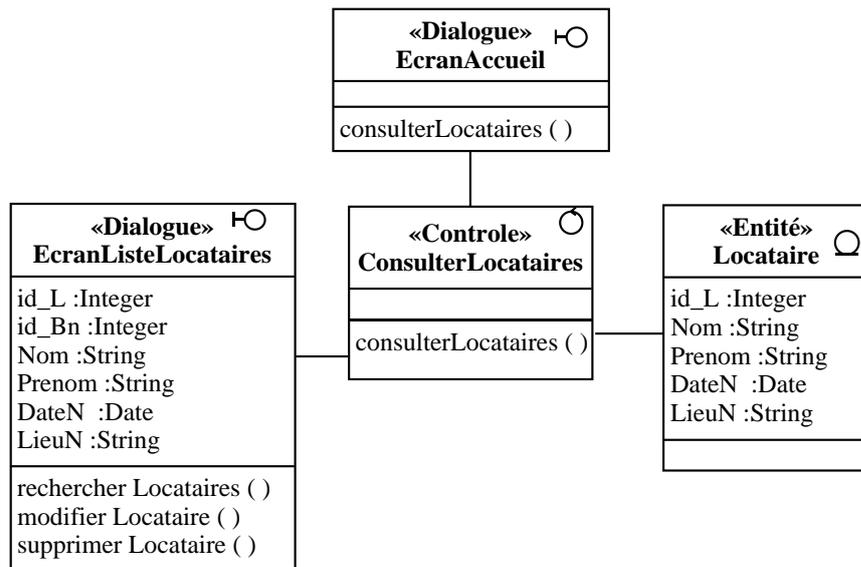


Figure 5.36: Diagramme de classes de conception du cas d'utilisation «consulter les locataires»

5.3.14 Rechercher des locataires

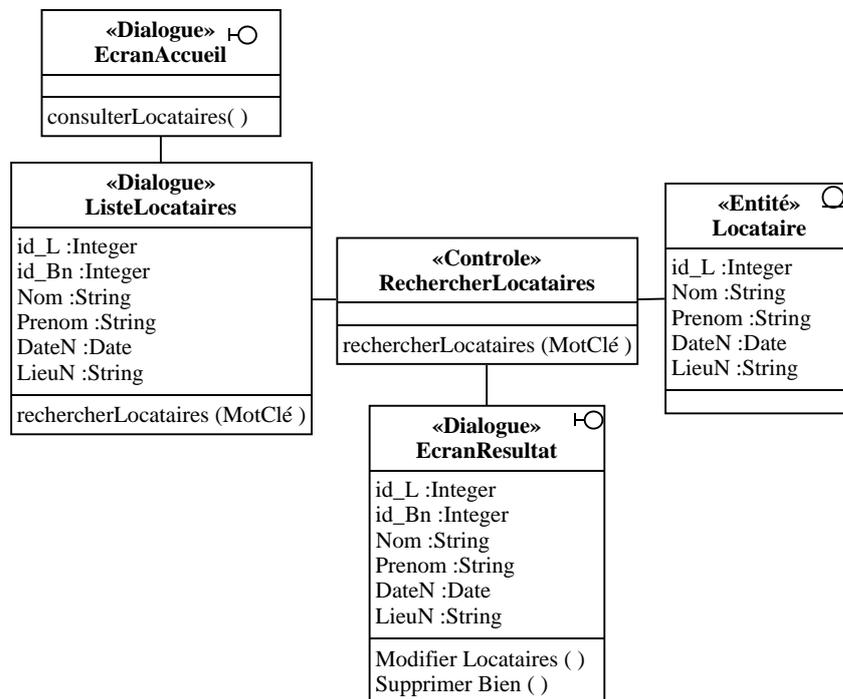


Figure 5.37: Diagramme de classes de conception du cas d'utilisation «rechercher les locataires»

5.3.15 Modifier un locataire

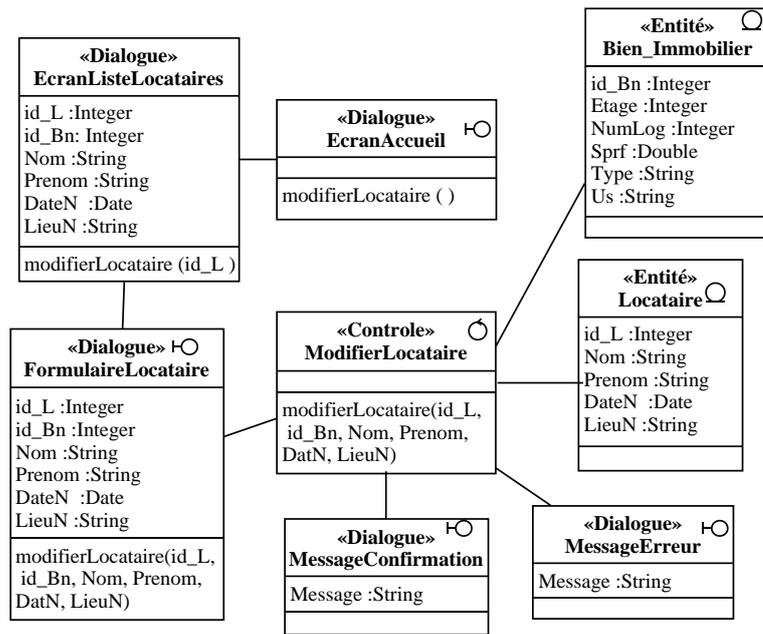


Figure 5.38: Diagramme de classes de conception du cas d'utilisation «modifier un locataire»

5.3.16 Supprimer un locataire

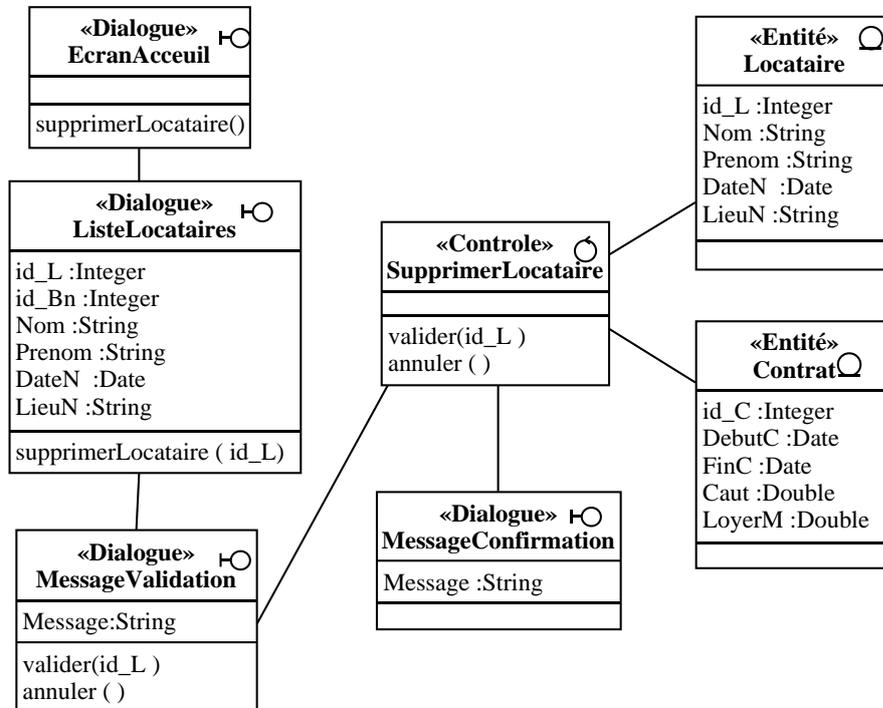


Figure 5.39: Diagramme de classes de conception du cas d'utilisation «supprimer un locataire»

5.3.17 Générer les loyers

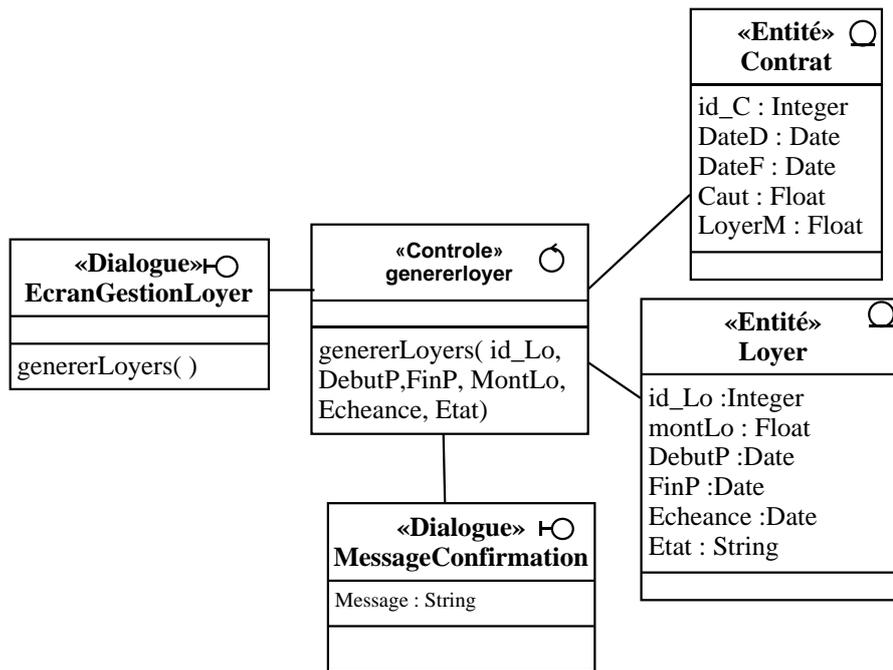


Figure 5.40: Diagramme de classes de conception du cas d'utilisation «générer les loyers»

5.3.18 Consulter les loyers

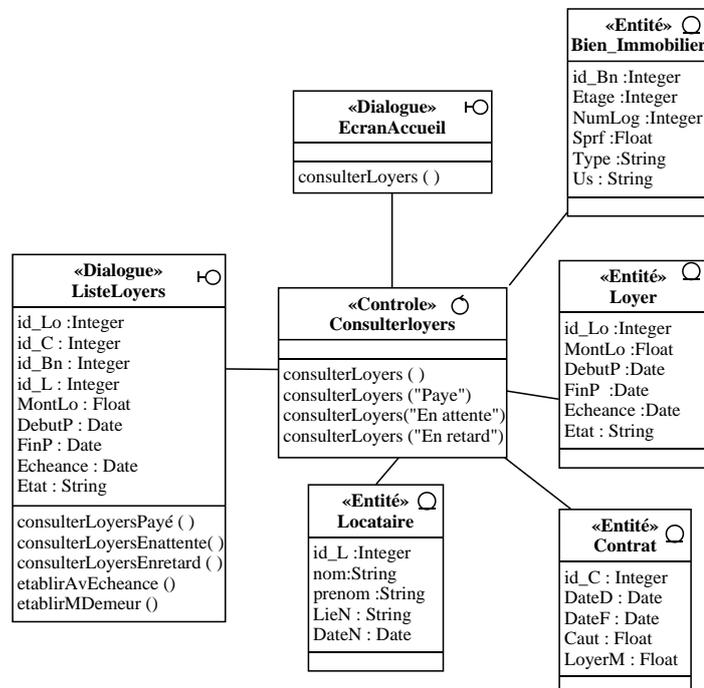


Figure 5.41: Diagramme de classes de conception du cas d'utilisation «consulter les loyers»

5.3.19 Recevoir un paiement

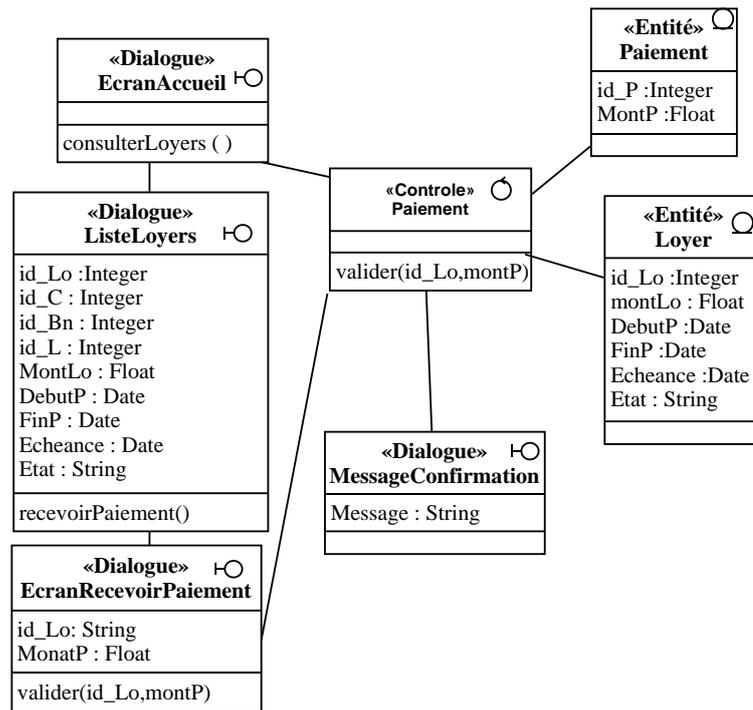


Figure 5.42: Diagramme de classes de conception du cas d'utilisation «recevoir un paiement»

5.3.20 Établir une quittance

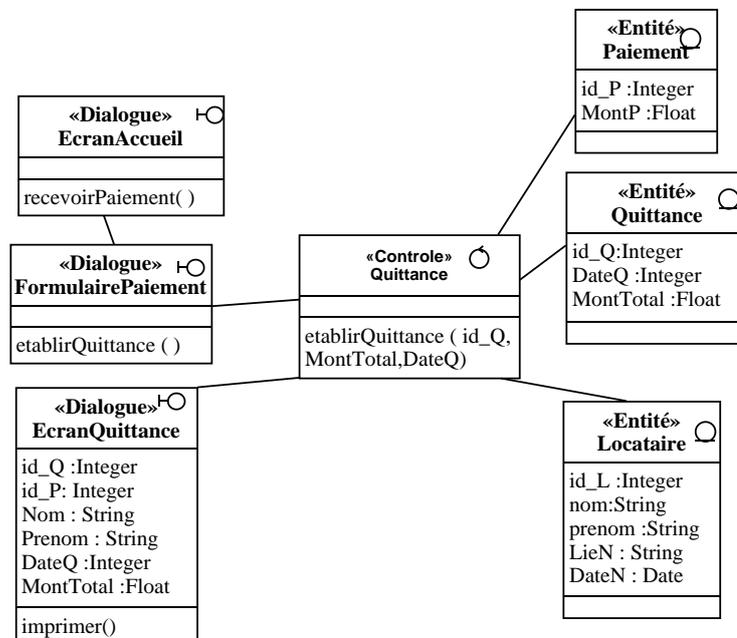


Figure 5.43: Diagramme de classes de conception du cas d'utilisation «établir une quittance»

5.3.21 Établir un avis d'échéance

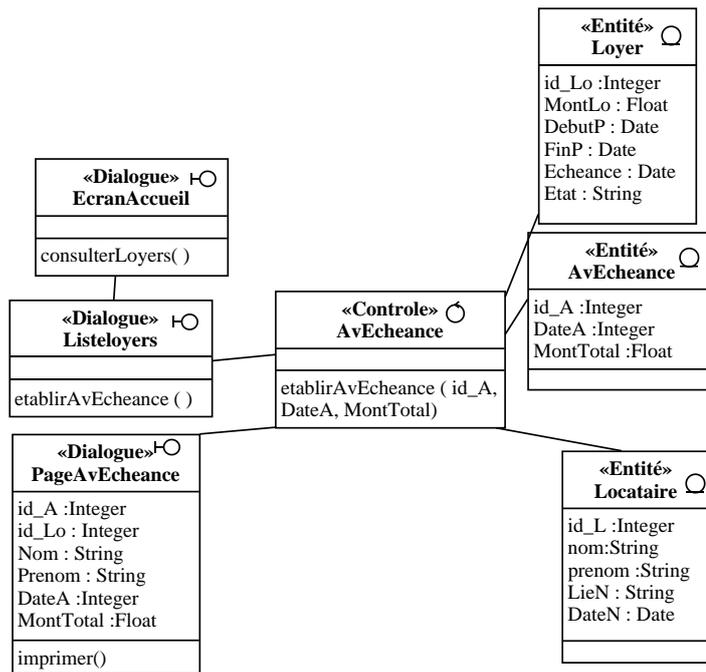


Figure 5.44: Diagramme de classes de conception du cas d'utilisation « établir un avis d'échéance »

5.3.22 Établir une mise en demeure

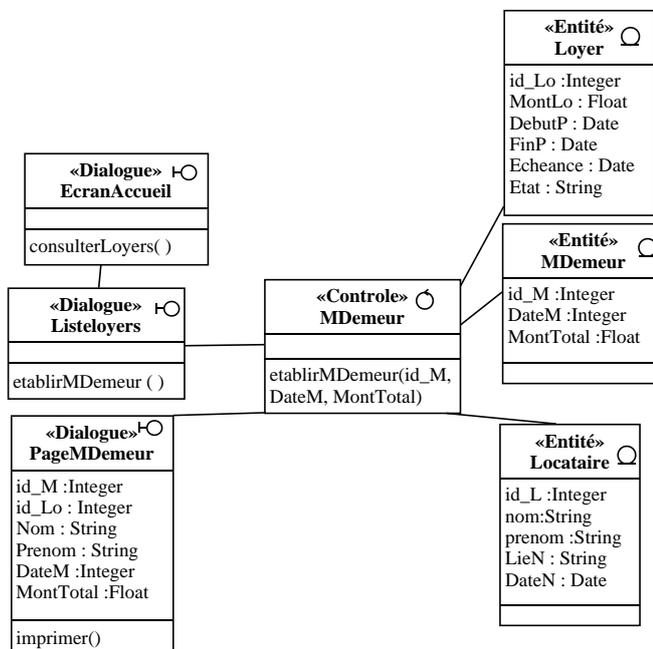


Figure 5.45: Diagramme de classes de conception du cas d'utilisation «établir une mise en demeure»

5.3.23 Consulter le tableau de bord

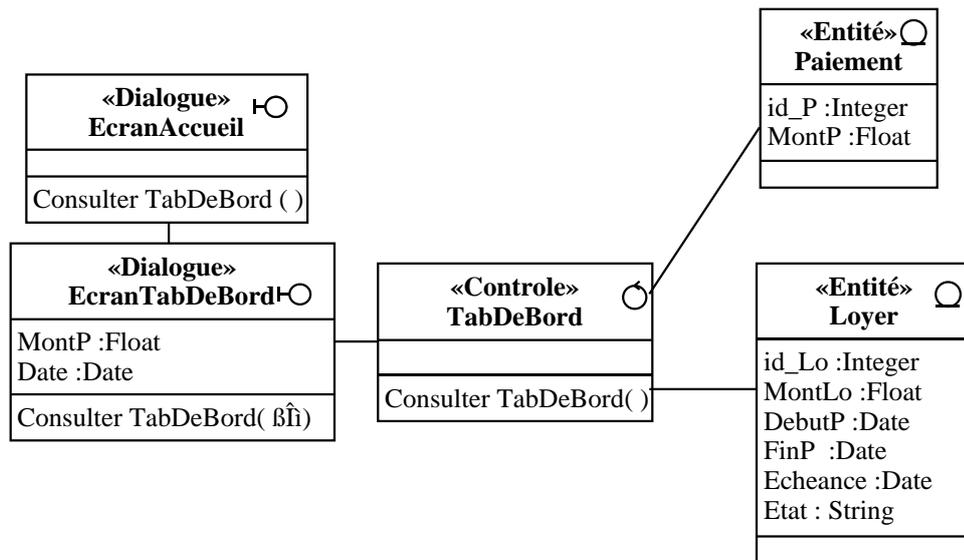


Figure 5.46: Diagramme de classes de conception du cas d'utilisation «consulter le tableau de bord»

5.4 Conclusion

Dans ce chapitre, nous avons développé les diagrammes d'interactions ainsi que les diagrammes de classes de conceptions des cas d'utilisations qui représentent la phase de conception de notre développement de l'application. Le chapitre suivant présentera la phase d'implémentation, les langages et les outils utilisés et donnera une première vision des différentes interfaces de l'application.

Chapitre 6

Implémentation

6.1 Introduction

Après l'analyse et la conception de l'application, nous allons passer à la phase d'implémentation. Nous allons utiliser les résultats de la phase de conception (les diagrammes de classes de conception, et les diagrammes de séquence) pour implémenter notre application dans un langage de programmation. Nous allons également créer une base de données en convertissant les classes entités du modèle du domaine à des tables relationnelles. Dans ce chapitre nous allons passer en revue les langages et les outils de développement utilisés pour l'implémentation, puis, nous présentons quelques interfaces de notre application.

6.2 Passage du diagramme de classes au modèle relationnel

Il existe plusieurs solutions de transformation mais ces règles qu'on va utiliser sont les plus simples et les plus opérationnelles :

6.2.1 Les règles de passage

Nous allons traduire le modèle du domaine (voir la figure 4.1) sous forme de tables relationnelles.

Pour ce faire, nous appliquons des règles de passages où chaque classe devient une relation. Les attributs de la classe deviennent des attributs de la relation. Si la classe possède un identifiant, il devient la clé primaire de la relation, sinon, il faut ajouter une clé primaire arbitraire.

A. Association 1 vers 1

Pour représenter une association 1 vers 1 entre deux relations, la clé primaire de l'une des relations doit figurer comme clé étrangère dans l'autre relation.

B. Association 1 vers plusieurs

Pour représenter une association 1 vers plusieurs, on procède comme pour une association 1 vers 1, excepté que c'est forcément la relation du côté plusieurs qui reçoit comme clé étrangère la clé primaire de la relation du côté 1.

C. Association plusieurs vers plusieurs

Pour représenter une association du type plusieurs vers plusieurs, il faut introduire une nouvelle relation dont les attributs sont les clés primaires des relations en association et dont la clé primaire est la concaténation de ces deux attributs.

D. Classe-association plusieurs vers plusieurs

Ce cas est proche de celui d'une association plusieurs vers plusieurs, les attributs de la classe association étant ajoutés à la troisième relation qui représente, cette fois ci, la classe-association elle-même [9].

6.3 Tables de la base de données

En se basant sur les règles ci-dessus, nous avons converti les classes entités et leurs Associations, à des tables dans la base données. Les tables générées sont les suivantes :

Compte (id_U, Username, Password, Type, GererComptes, GererBiens, GererLocataires, GererPaiements, tableauDeBord).

Bâtiment (id_B, NBat, Daira, Commune, Rue).

Bien_Immobilier (id_Bn, id_B, Etage, NumLog, Sprf, Type, Us).

Locataire (id_L, id_Bn, Nom, Prenom, DateN, LieuN).

Contrat(id_C, id_L, id_Bn, DateD, DateF, Caut, LoyerM).

Loyer (id_Lo, MontLo, DebutP, FinP, Echeance, Etat).

Paielement (id_P, id_Lo, MontP).

Quittance (id_Q, id_Lo, DateQ, MontTotal).

Avis_echéance (id_A, id_Lo, DateA, MontTotal).

MiseEnDemeur (id_M, id_Lo, DateM, MontTotal).

6.4 Système de gestion de la base de donnée MySQL

Les alternatives de MySQL sont DB conçu par l'université de Berkeley, PostgreSQL et Oracle etc. De nombreux fournisseurs d'accès internet, et hébergeurs de sites proposent un accès gratuit à une plateforme PHP/MySQL, ceci soutient le choix de MySQL. D'autres facteurs d'ordre technique existent :

A. Vitesse de traitement

MySQL se distingue par une vitesse de traitement accrue résultant de la réécriture de routine ISMA de gestion des entrées/sorties, et de l'adoption d'algorithmes d'appariement de tables optimisés et organisés autour de threads.

B. Fiabilité

Des listes de diffusion active contribuent à la correction rapide d'éventuels bogues de MySQL. De plus, des utilitaires de teste de cohérence automatique existent, comme MYIAMCHK.

C. Compatibilité SQL

MySQL est compatible avec SQL, il respecte la norme ANSI SQL 92.

D. Sécurité

L'administrateur d'une base de données contrôle finement les catégories de personnes et les machines autorisées à se connecter, et l'authentification repose sur des

mots de passe cryptés. D'autre part, MySQL montre des limitations. Il n'intègre pas de mécanisme de transaction ou de réplication des bases de données existantes, ainsi que les instructions SELECT ne peuvent pas être imbriquées etc. [22].

6.5 Langage Web utilisés

6.5.1 HTML

HTML, signifie « Hypertext Markup Language », c'est-à-dire langage de marquage hypertexte. C'est un langage de description permettant de structurer et d'afficher différents objets sur un écran (qu'on appellera abusivement "page"). Ces objets peuvent être du texte, des tableaux, des images voire de la vidéo et des sons. Ils peuvent être aussi les éléments devenus classiques des environnements graphiques, à savoir les boutons, pop-listes, listes à ascenseurs, boîtes de dialogue... Sur la plupart de ces objets, il est possible de placer des liens qui vont permettre de se connecter à d'autres pages. Ce langage est donc à l'origine essentiellement statique. (Figure 6.1) [10].

```
<html>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title> Gestion Locative | OPGI Mila</title>
<head>

<link rel="stylesheet" href="../../dist/css/Opgi.min.css">
</head>
<body>
  <div class="content">
    <div class="box-header with-border">
      <h1 class="box-title">Modifier Bien Immobilier</h1>
    </div>
    hv
  </div>

</body>

<div class="footer">

</div>

</html>
```

Figure 6.1: Une partie du code HTML de notre application.

6.5.2 CSS

CSS signifie « Cascading Style Sheets », ce qui se traduit en français par feuilles de style en cascade. La feuille de style fournit la mise en forme des éléments de la page, qui auront été écrits en XHTML. Elle s'applique à une ou plusieurs pages du site. Le terme « en cascade » indique que la mise en forme d'une page peut faire appel à plusieurs feuilles de style. Les différentes propriétés affectées à un même élément s'ajoutent alors pour lui donner sa mise en forme finale. Lorsque deux propriétés se contredisent, des règles de priorité s'appliquent et c'est généralement le dernier style défini qui est pris en compte. (Figure 6.2) [11].

```
body{
  background: url(../images/bg1.jpg) no-repeat 0px 0px;
  font-family: 'Open Sans', sans-serif;
  background-size:cover;
  -webkit-background-size:cover;
  -moz-background-size:cover;
  -o-background-size:cover;
  min-height:1050px;
}
.Content{
  margin: 0 auto;
  width: 80%;
}
h1 {
  font-family: 'Exo 2', sans-serif;
  text-align: center;
  padding-top: 4em;
  font-weight: 400;
  color: #2B2B36;
  font-size: 2em;
}
.login-form {
  background: #2b2b36;
  position: relative;
  width: 30%;
  margin: 3% auto 0 auto;
```

Figure 6.2: Une partie du code CSS de notre application.

6.5.3 JAVASCRIPT

Ce langage de programmation est conçu pour traiter localement des événements provoqués par le lecteur (par exemple, lorsque le lecteur fait glisser la souris sur une zone de texte, cette dernière change de couleur). C'est un langage interprété, c'est-à-dire que le

texte contenant le programme est analysé au fur et à mesure par l'interprète, partie intégrante du browser, qui va exécuter les instructions. (Figure 6.3) [10].

```
<script src="../../dist/js/demo.js"></script>
<!-- page script -->
< <script>
  $(document).ready(function(){
    $('#suivre_data').DataTable();
  });
</script>
```

Figure 6.3: Une partie du code JavaScript de notre application.

6.5.4 PHP

PHP (PHP Hypertext PreProcessor) est un langage de programmation. Sa principale application se situe au niveau de la gestion des sites web dynamiques. On peut par exemple lui faire créer le contenu de pages HTML suivant différents paramètres : l'âge d'un visiteur, sa catégorie socioprofessionnelle, des mots-clés qu'il aura indiqués dans un moteur de recherche, des actualités du jour, etc. Les capacités de PHP ne s'arrêtent pas à la création de pages web. Il est aussi possible de manipuler des images, de créer des fichiers PDF, de se connecter à des bases de données. (Figure 6.4) [21].

```
<?php
while($row = mysqli_fetch_array($result))
{
    echo '
    <tr>
        <td>' . $row["username"] . '</td>
        <td>' . $row["password"] . '</td>
        <td>' . $row["type"] . '</td>
        <td>' . $row["Permi"] . '</td>

    </tr>
    ';
}
?>
```

Figure 6.4: Une partie du code PHP de notre application.

6.6 Outils de développements

6.6.1 Pacestar UML Diagrammer



Pacestar UML Diagrammer est un logiciel qui crée des diagrammes d'UML, développe, documente, et communique vos conceptions dans une notation généralement comprise systématique pourtant flexible. Il soutient l'activité, la classe et l'objet, le cas d'utilisation, l'ordre, l'état, le paquet, et les diagrammes composants de déploiement. Ceci des aides spécialisées d'outil de schéma et de conception vous exploitez le meilleur aspect d'UML, le processus visuel normalisé de conception sans imposer une méthodologie encombrante de projet. Et il simplifie ce processus de sorte que vous puissiez être productif immédiatement avec peu ou pas de formation [12].

6.6.2 Editeur de texte Atom



Présenté comme l'éditeur de texte du XXIème siècle par ses développeurs, Atom est un puissant éditeur de code open source dont la structure repose sur les meilleurs éléments d'autres éditeurs de texte également open source. Doté d'une interface pratique et moderne, Atom s'articule autour de deux blocs, l'un représentant l'arborescence des dossiers et facilitant la navigation, l'autre dédié à l'ouverture des fichiers. Multifonctionnel, l'éditeur vous permet d'ouvrir et de gérer d'autres documents que des fichiers texte. Ainsi, vous pouvez vous servir d'Atom pour travailler sur une image ou du code source. Notez d'ailleurs que le logiciel supporte la fonction de surlignage pour aider à la programmation [13].

6.6.3 Navigateur Web (Google Chrome)



Un navigateur web est un logiciel informatique qui permet d'utiliser le web. Pour être plus précis, ce type de logiciel permet de consulter le World Wide Web (WWW). L'utilisation la plus répandue de ces logiciels étant de visualiser les pages web et d'utiliser les liens hypertextes dans le but d'aller de pages en pages. Le navigateur web qu'on a utilisé est : Google Chrome qui est développé par Google, ce logiciel est gratuitement mis à disposition du grand public depuis fin 2008 [14].

6.6.4 Serveur web Wamp



WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données [15].

6.6.5 Bootstrap



Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de

navigation et bien d'autres encore. On appelle ce type de framework un "Front-End Framework" [16].

6.7 Sécurité de l'application

La sécurité est un facteur principal dans les sites web pour cela nous avons sécurisé notre site contre les injections SQL. Une injection SQL est un type d'exploitation d'une faille de sécurité d'une application interagissant avec une base de données, en injectant une requête SQL non prévue par le système et pouvant compromettre sa sécurité. Il y a deux types d'injections SQL:

- L'injection dans les variables qui contiennent des chaînes de caractères.
- L'injection dans les variables numériques.

6.7.1 Comment éviter ces attaques d'injection en PHP

Les « magic quotes » étaient utilisées par défaut dans la configuration de PHP. Elles permettaient qu'un caractère d'échappement soit automatiquement placé devant les apostrophes et guillemets dangereux. Certaines failles de sécurité et la disparition de magic quotes dans PHP incitent à remplacer cette option par la fonction `mysql_real_escape_string` [17].

6.7.2 Cryptage du mot de passe (Password-hash)

La fonction `password_hash()` crée un nouveau hachage en utilisant un algorithme de hachage fort et irréversible. La fonction `password_hash()` est compatible avec la fonction `crypt()`. Aussi, un mot de passe haché par la fonction `crypt()` peut être utilisé avec la fonction `password_hash()`.

Cette fonction supporte plusieurs algorithmes (Password-DEFAULT, Password-BCRYPT, Password-ARGON2I), dans notre cas nous avons utilisée :

PASSWORD_BCRYPT : Utilisation de l'algorithme CRYPT_BLOWFISH pour créer la clé de hachage. Ceci va créer une clé de hachage standard `crypt()` utilisant l'identifiant

"\$2y\$". Le résultat sera toujours une chaîne de 60 caractères, ou FALSE si une erreur survient [18].

6.8 Présentation de l'application

L'application que nous avons développée possède pratiquement toutes les fonctionnalités décrites dans le cahier des charges. L'application est dotée d'une interface graphique composée de fenêtres, permettant à l'utilisateur d'accéder aux différents services d'une façon conviviale. Dans ce qui suit, nous passons en revue les différentes interfaces de notre application :

6.8.1 Page Authentification

La page d'authentification permet aux utilisateurs de l'application de s'identifier par leurs login et leurs mots de passe (Figure 6.5).

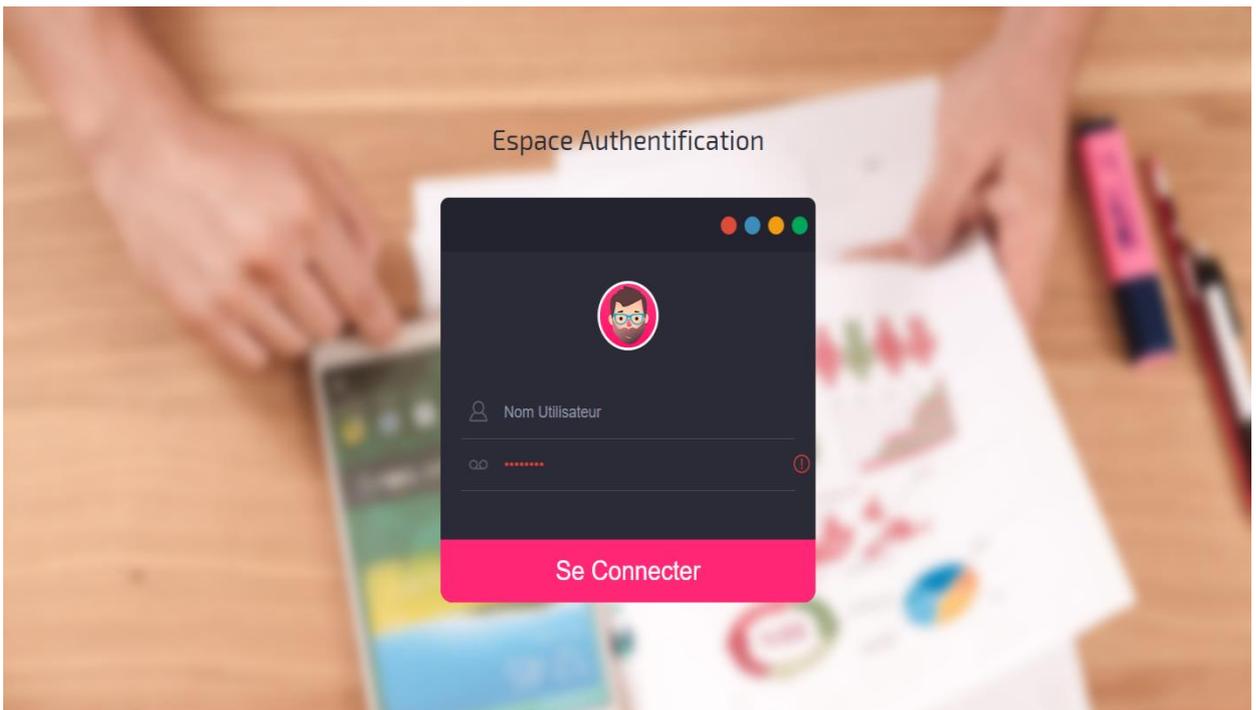


Figure 6.5: Page «S'authentifier»

6.8.2 Page Accueil

C'est la page d'accueil qui s'affiche dès l'accès à notre application. Elle permet à l'utilisateur d'accéder aux différentes fonctionnalités de l'application en cliquant sur le bouton correspondant de la barre de menu horizontal contenant des liens à toutes les pages de notre application (Figure 6.6).

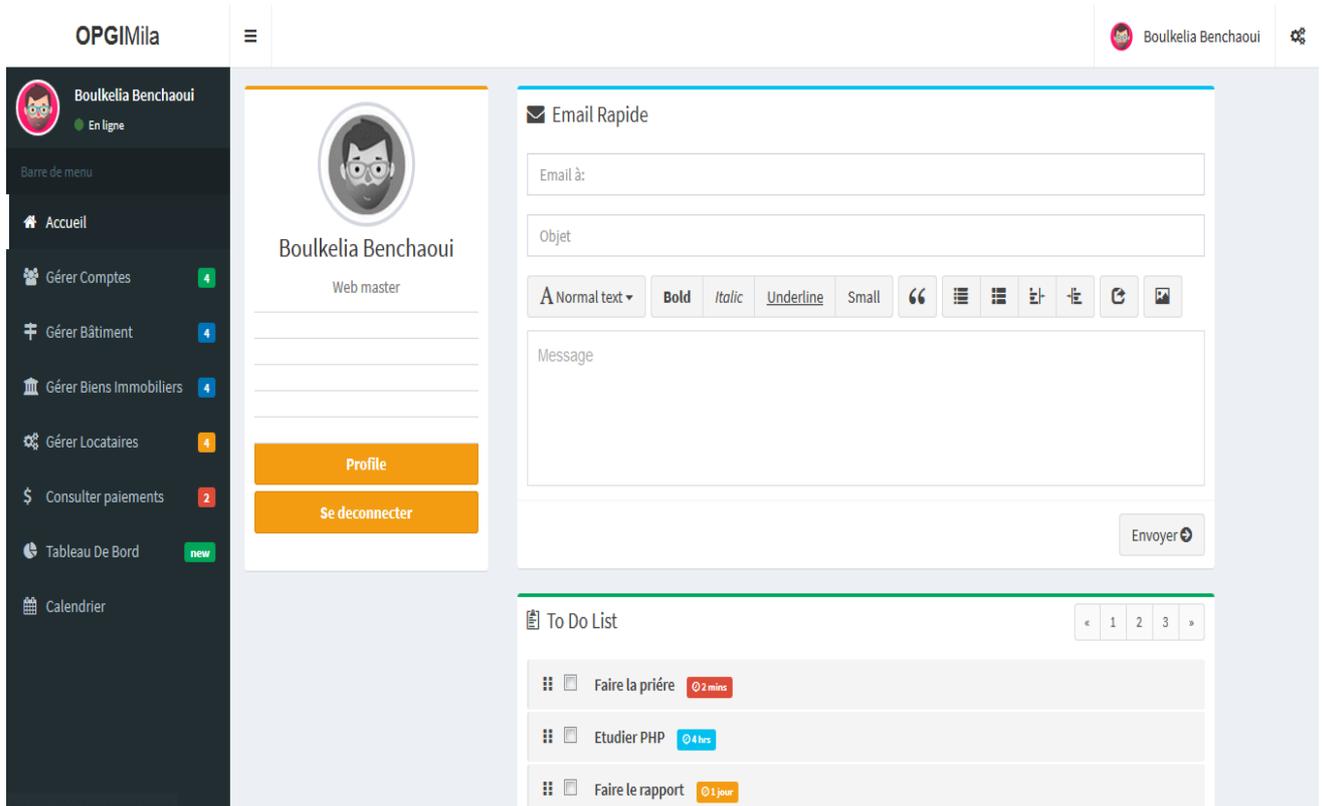


Figure 6.6: Page «Accueil»

6.8.3 Page de création d'un compte

Permet au web master de créer des comptes pour chacun des employés ou utilisateurs de l'application, elle permet également de préciser les permissions a les quelles l'utilisateur a le droit d'accéder (Figure 6.7).

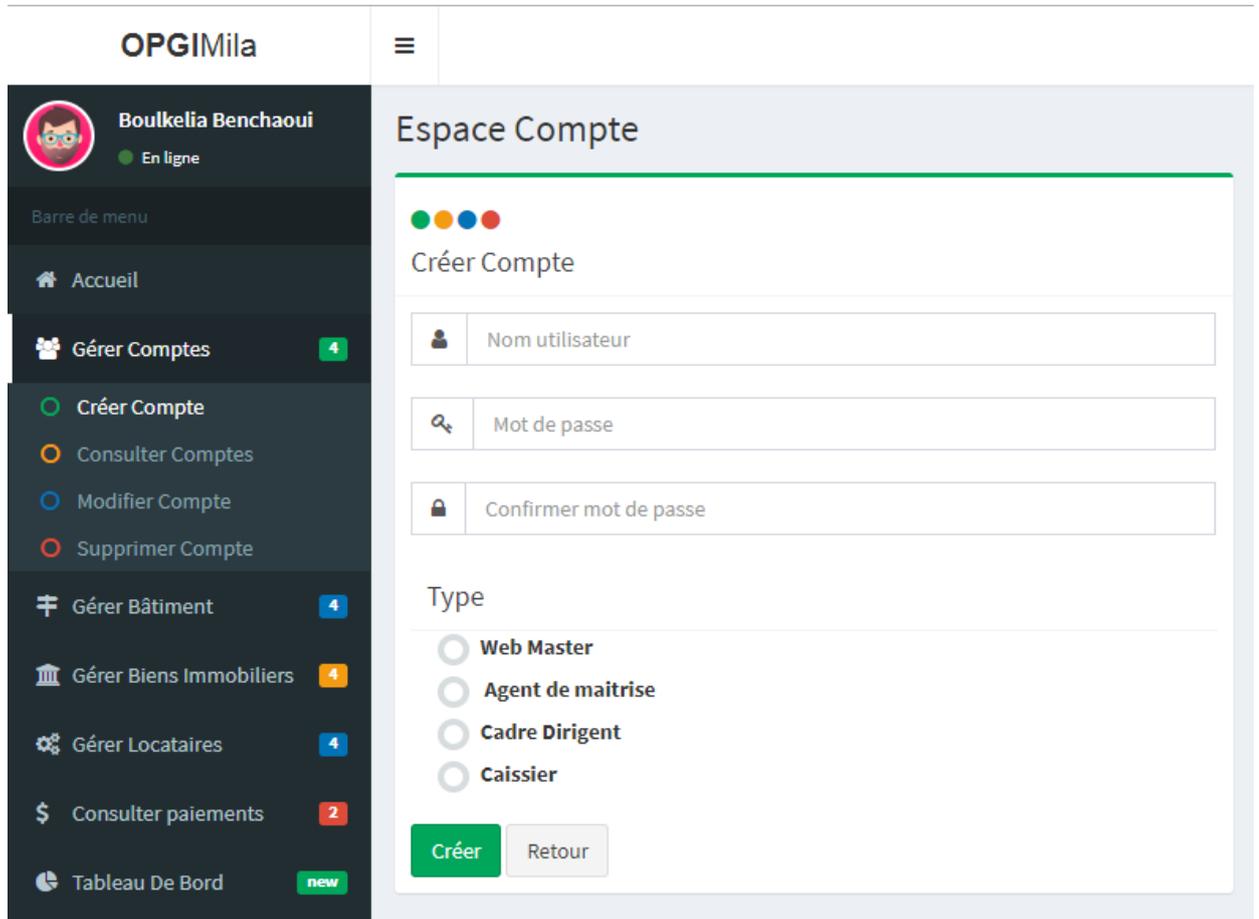


Figure 6.7: Page «créer un compte»

6.8.4 Page de consultations des bâtiments

Permet à l'agent de maitrise de consulter la liste des bâtiments appartenant au patrimoine de l'office, à travers cette interface, l'agent de maitrise aura la possibilité de rechercher, modifier ou encore supprimer un bâtiment (Figure 6.8).

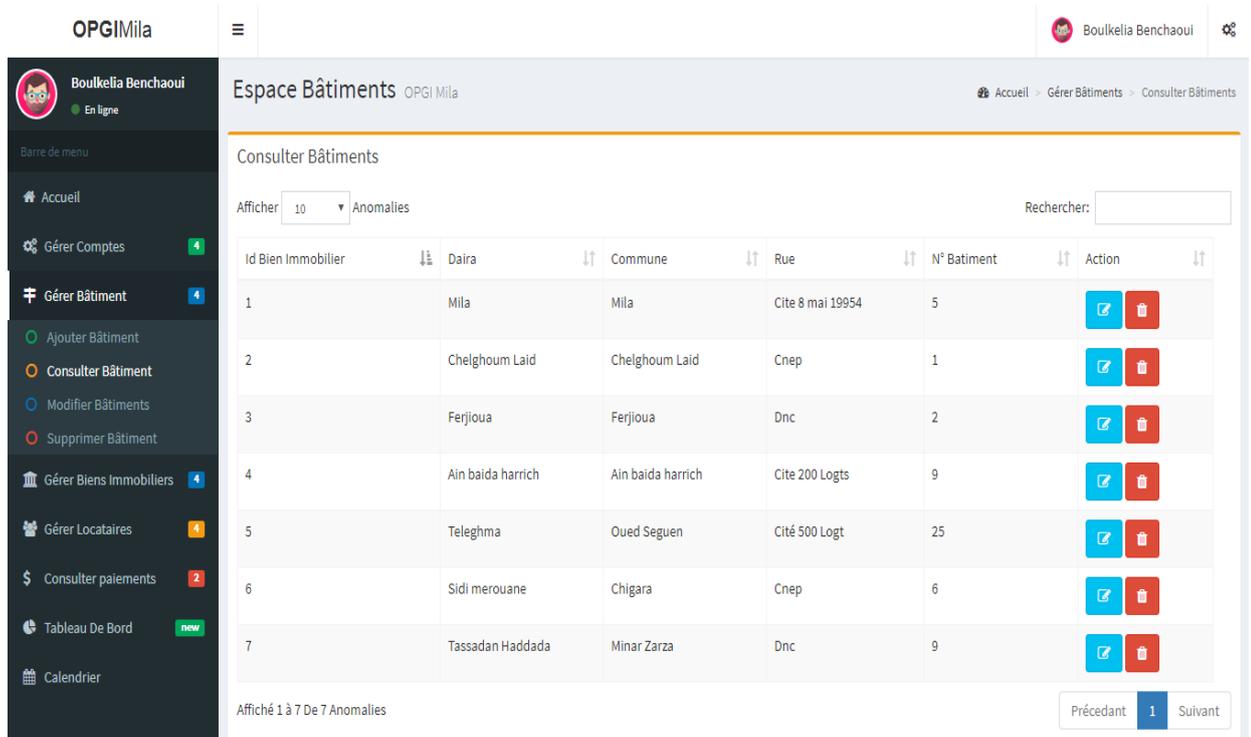


Figure 6.8: Page «consulter les bâtiments»

6.8.5 Page de modification d'un bien immobilier

Permet à l'agent de maitrise de modifier les informations relatives à un bien immobilier, Après la modification, un message de confirmation sera affiché (Figure 6.9).

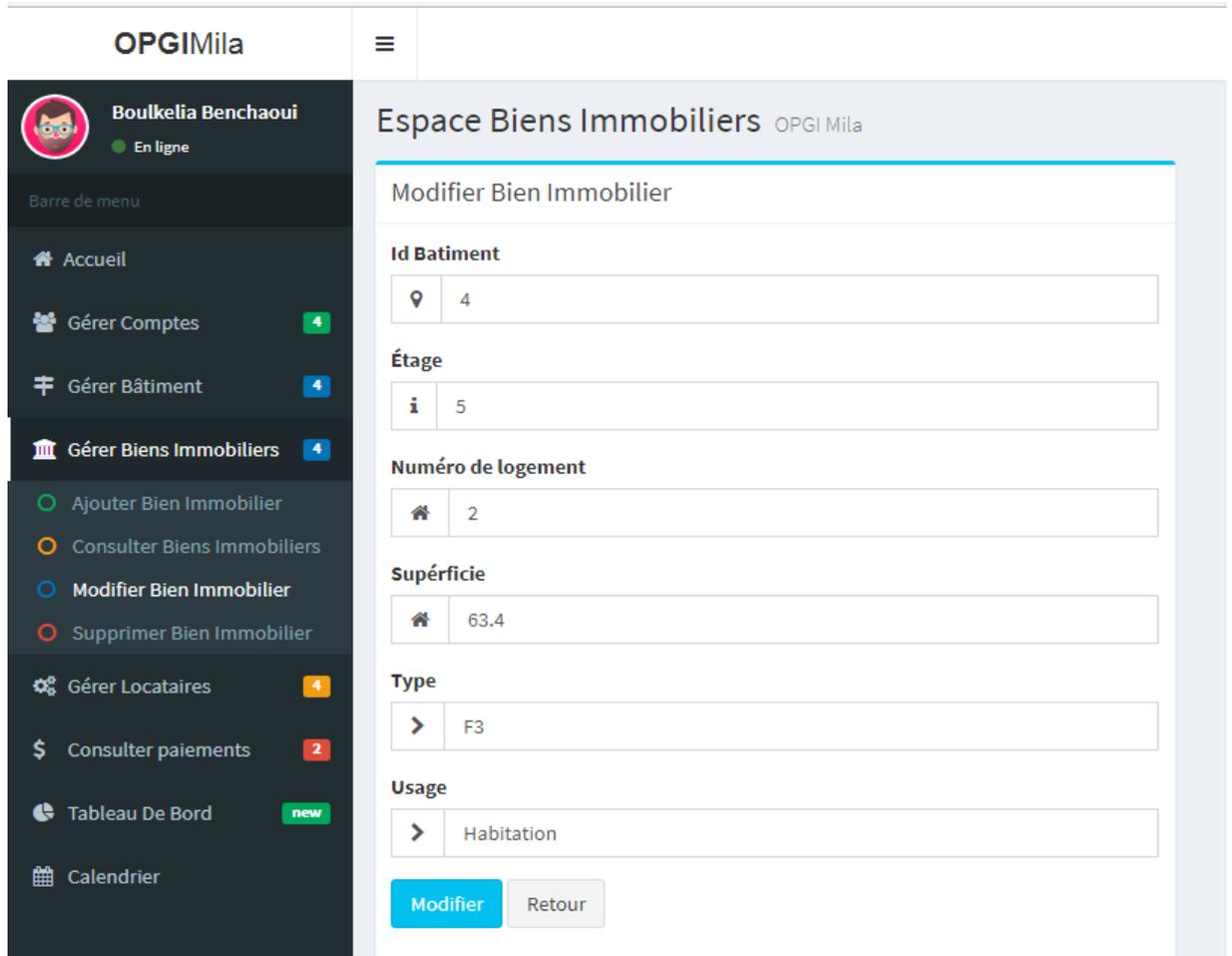


Figure 6.9: Page «modifier un bien immobilier»

6.8.6 Page de suppression d'un locataire

Permet à l'agent de maitrise la suppression d'un locataire. Après la suppression, un message de confirmation sera affiché (Figure 6.10).

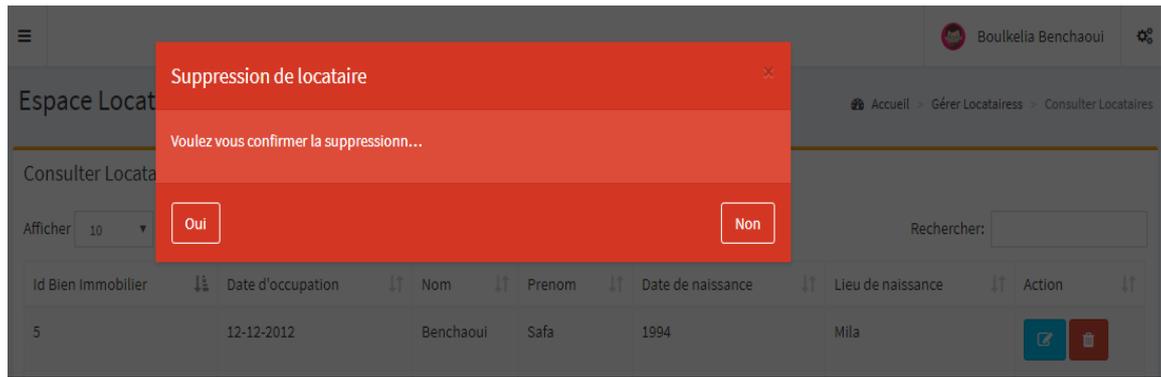


Figure 6.10: Page «supprimer un locataire»

6.8.7 Page d'établissement de contrat de location

Permet à l'agent de maîtrise d'établir un contrat de location pour chaque locataire (Figure 6.11).

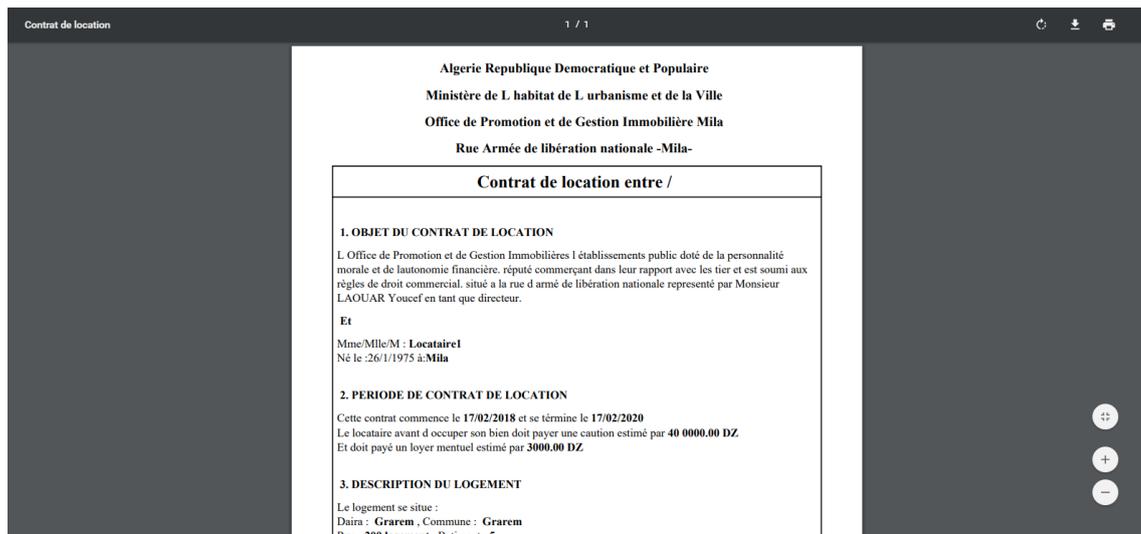


Figure 6.11: Page «établir contrat de location»

6.8.8 Page de consultation des loyers

Cette page permet au caissier de consulter la liste des loyers déjà générés, avec toutes leurs informations principales, elle permet également de consulter les loyers payés, les loyers en attente et ceux en retard, et en fonction de cela l'utilisateur aura la possibilité d'établir ou bien un avis d'échéance ou une mise en demeure (Figure 6.12).

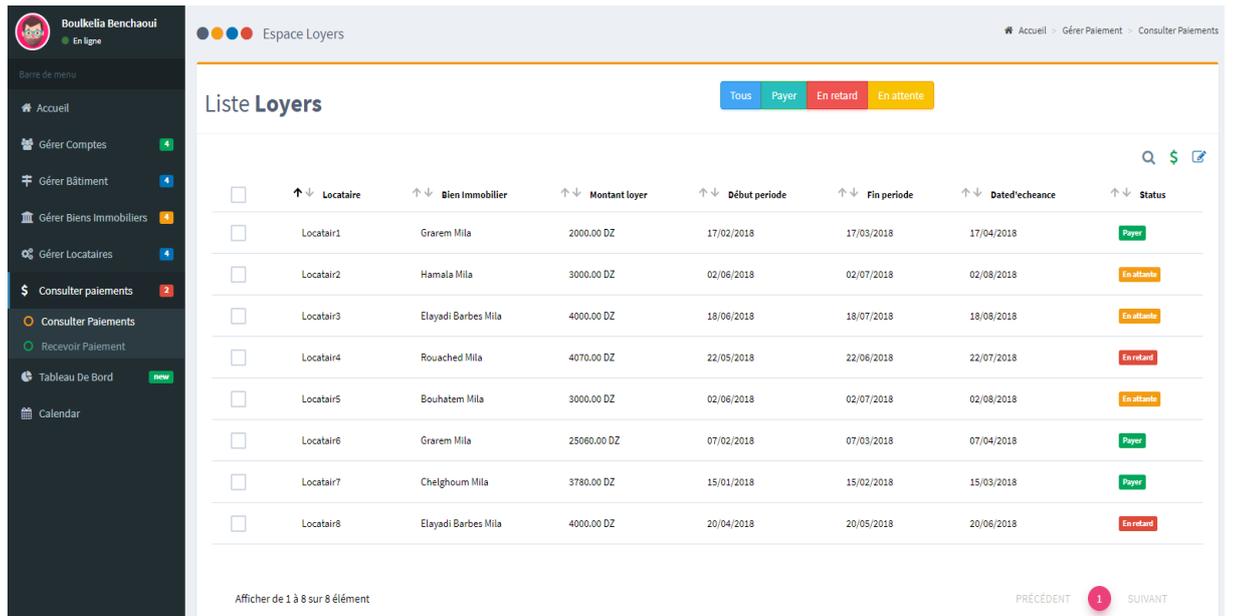


Figure 6.12: Page «consulter les loyers»

6.8.9 Page de réception d'un paiement

Permet au caissier d'enregistrer un paiement en spécifiant le montant payé (Figure 6.13).

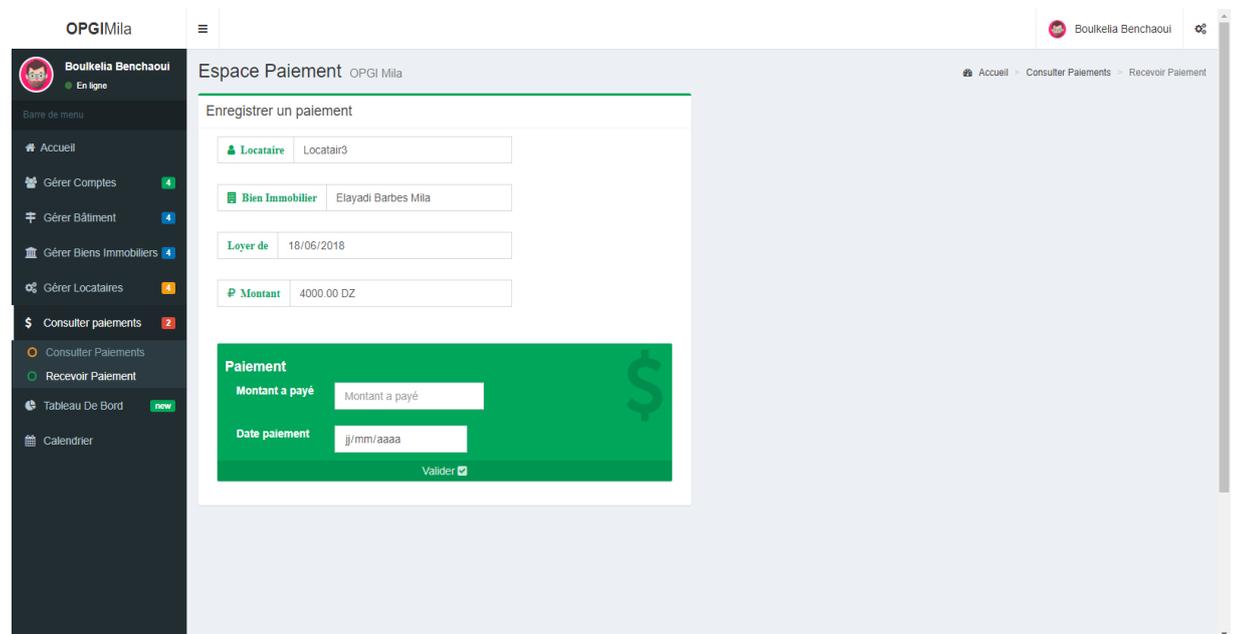


Figure 6.13: Page «recevoir un paiement»

6.8.10 Consulter le tableau de bord

A travers cette interface le cadre dirigeant peut effectuer l'analyse financière en consultant les recouvrements ainsi que les dettes de chaque année (Figure 6.14).

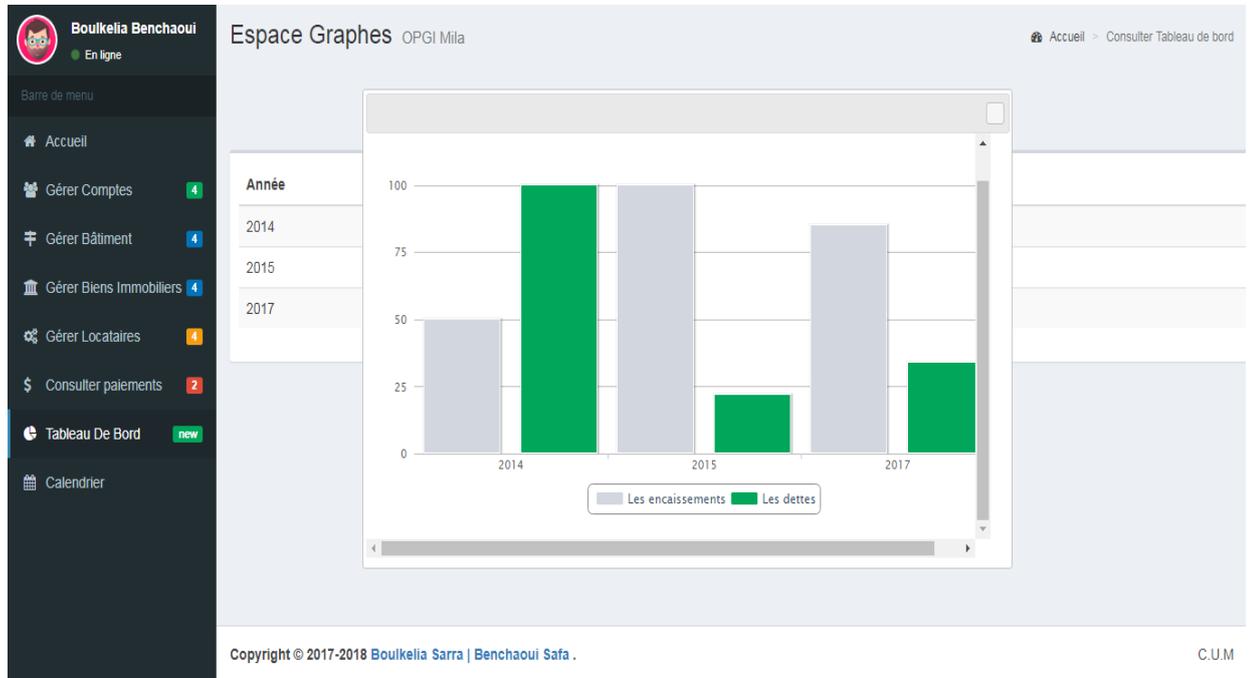


Figure 6.14: Page «consulter le tableau de bord»

6.9 Conclusion

Dans ce chapitre, nous avons présenté les outils et les logiciels utilisés pour le développement de notre application. Ensuite, nous avons présenté quelques interfaces. La phase d'implémentation, nous a permis de matérialiser les résultats des phases précédentes. Elle nous a permis également d'apprendre des langages et des outils nécessaires pour le développement des applications web.

Conclusion générale

L'objectif de notre projet de fin d'études était la contribution à l'élaboration d'une nouvelle solution de gestion locative au niveau de l'OPGI de Mila. Le point de départ de la réalisation de ce projet était une récolte d'informations nécessaires pour dresser une étude de l'existant, présenter un aperçu sur l'organisme d'accueil et les logiciels utilisées, ainsi que les limites de l'existant. Ceci nous a permis d'élaborer une nouvelle solution pour répondre à ces limites. Cette solution consiste à : (1) établir un réseau intranet VPN reliant les différents intervenants qui sont répartis géographiquement sur plusieurs sites, et (2) le développement d'une application web pour la gestion locative, accessible sur le réseau intranet de l'organisme.

A ce stade, nous avons réussi à trouver une solution économique et rapide à mettre en œuvre pour créer un réseau intranet reliant les différentes structures concernées, et nous avons réussi à développer une application qui peut offrir les fonctionnalités les plus importantes pour la gestion locative. La période que nous avons passée à la réalisation de notre projet de fin d'étude, nous a permis d'enrichir nos connaissances dans de domaines très variés, entre autres: Les réseau VPN, le langage UML et le processus UP, le langage web (HTML,PHP, CSS, etc.), et Le SGBD MYSQL.

Perspective

La solution que nous avons proposée n'a pas été tous réalisé, il est possible de mettre en place un réseau intranet VPN pour pouvoir relier les différent intervenants de la gestion locative et renforcer la sécurité. En ce qui concerne l'application la plus part des

fonctionnalités mentionnées dans le cahier des charges ont été implémenté, mais il est encore possible de les améliorer, par exemple par l'envoi des quittances, des mise en demeure et des avis d'échéance aux locataires concernés par email, ainsi sauvegarder l'historiques des quittances imprimées.

Bibliographie

[1] MHUV. Ministère de l'Habitat, et de l'Urbanisme et de la Ville [en ligne] (page consultée mars 2018). <http://www.mhuv.gov.dz/Pages/Article.aspx?a=17>.

[2] Publicis Regicom. Annoncelagegestionlocative [en ligne] (page consultée mars 2018). <http://www.immo-fourmier.fr/gestion-locative/la-gestion-locative-2288>.

[3] Roques, P. (2008). UML 2: Modéliser une application web. Editions Eyrolles.

[4] Archier, J. P. (2013). Les VPN-Fonctionnement, mise en oeuvre et maintenance des Réseaux Privés Virtuels [2ième édition]. Éditions ENI.

[5] EDUCINVEST Belgium. Les VPN [en ligne] (page consultée avril 2018). <https://www.supinfo.com/articles/single/41-vpn-virtual-private-network-fonctionnement-interets>.

[6] Haddouche, W., Hadjrioua, K., & Aloui, A. (2017). Mise en oeuvre d'une solution VPN sous linux cas: université de béjaia (Doctoral dissertation, universite abderrahmane mira Bejaia).

[7] Cours. Les VPN. [En ligne] (Page consultée avril 2018). [Http : //perso.modulonet.fr/placurie /Ressources/BTS2-AMSI/Chap-8-Les-VPN.pdf](http://perso.modulonet.fr/placurie/Ressources/BTS2-AMSI/Chap-8-Les-VPN.pdf).

[8] LANDRY William. Mise en place d'une architecture vpn mpls avec gestion du temps de connexion et de la bande passante utilisateur. Institut d'ingénierie d'informatique de Limoge, ISTD - Master Europeen en Informatique OPTION : Administration systèmes réseaux, 2009.

[9] Audibert, L. (2009). UML 2: de l'apprentissage à la pratique... Ellipses.

[10] Charnay, D., & Chaléat, P. (1998). HTML et Javascript (p. 450). Eyrolles.

[11] Draillard, F. (2017). Premiers pas en CSS3 et HTML5: 7e édition mise à jour. Editions Eyrolles.

[12] CBS Interactive Inc.PACESTAR UML DIAGRAMMER [en ligne] (page consultée avril 2018). http://telecharger.cnet.com/Pacestar-UML-Diagrammer/3000-2383_4-10212093.html.

[13] 01telecharger.Editeur de texteAtom[en ligne] (page consultée mai 2018).http://www.01net.com/telecharger/windows/Bureautique/editeur_de_texte/fiches/130958.html.

[14] InfoWebMaster.Navigateur web [en ligne] (page consultée avril 2018).
<http://glossaire.infowebmaster.fr/navigateur-web/>.

[15] ALTER WAY.WampServer [en ligne] (page consultée février 2018).
<http://www.wampserver.com/>.

[16] JDN.Bootstrap [en ligne] (page consultée Mai 2018).
<https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap/>.

[17]Wikipedia.Injection SQL [en ligne] (page consultée Avril 2018).
https://fr.wikipedia.org/wiki/Injection_SQL.

[18]T he PHP group.Password-hash [en ligne] (page consultée Avril 2018).
<http://php.net/manual/fr/function.password-hash.php>.

[19] Le Crabe Info. C'est quoi un firewall [en ligne] (page consultée Mai 2018).
<https://lecrabeinfo.net/cest-quoi-firewall-faut-installer-pare-feu-windows-7-8-10.html?highlight=firewall>.

[20] Audibert, L. (2008). UML 2.0. Cours. Institut Universitaire de Technologie de Villetaneuse–Département Informatique, Villetaneuse, France.

[21] Daspet, E., & Geyer, C. P. (2000). PHP5 avancé 4e edition. Book, EYROLLES.

[22] C. Notion. L'essentiel pour concevoir un site web dynamique. PHP et MY SQL en ligne.