

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° Ref :.....

## Centre Universitaire de Mila

Institut des Sciences et de la Technologie Département de Mathématiques et Informatique

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

**En :**

**Filière : informatique générale**

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

# Réalisation d'une application client-serveur pour la gestion du patrimoine communal de la wilaya de Mila

**Préparé par : - Belhadj Meriem**

**- Benmekhlouf Amal**

**Soutenu devant le jury :**

**- Président : Fatiha Benabderahmane .**

**Grade : MAA, C.U.Mila.**

**- Examineur : Faiza Afri.**

**Grade : MAB, C.U.Mila.**

**- Promoteur : Nardjes Bouchemal.**

**Grade : MAA, C.U.Mila.**

**Année universitaire : 2013/2014**

## *Remerciements*

*Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.*

*Un remerciement particulier à notre encadreur Mme Bouchemal Nardjes pour sa présence, son aide et surtout pour ses précieux conseils qui nous ont assistés pour l'accomplissement de notre projet.*

*Nos vifs remerciements vont également aux membres du jury Mme Benabderrahmane Fatiha et Mme Afri Faiza, pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.*

*Nos remerciements à nos très chers parents, frères particulièrement Riyad Benmekhlouf, sœurs, amis particulièrement Radouane, Nassim, Amine, respectives qui nous ont encouragés, soutenu durant tout notre parcours. Nous tenons à exprimer nos sincères remerciements à tout le personnel de l'institut de sciences et de la technologie du centre universitaire de Mila surtout les enseignants qui nous ont enseigné durant toutes nos années d'étude.*

*Nous tenons à remercier tous les employeurs de la wilaya de Mila et la commune de Mila particulièrement Mr Tabbakh Abdalhamid, Mr Dagha Boudjemâa.*

*Nous remercions les étudiants de la promotion 2013/2014 pour avoir été liés et unis tout au long de cette année et tous ceux qui ont collaborés de près ou de loin à l'élaboration de ce travail. Qu'ils acceptent nos humbles remerciements.*

*Meriem L. Amal*

## *Dédicace*

*Je tiens en tout premier lieu à remercier le dieu je voudrais  
dédier ce modeste travail*

*A mon père qu'il à été toujours pour moi et qui m'a soutenu  
toute sa vie, que dieu le protège.*

*A celle qui a cru en moi, et qui m'a toujours soutenu  
pendant toutes ces longues années avec ses conseils et  
orientations ; à toi chère et adorable mère j'aime de plus  
profond de mon cœur, que dieu la protège.*

*A mes chères sœurs : Selma & Amina*

*A mes chères frères : Tarek & Ayoub & Ziad & Lokman*

*A toute mes oncles et mes tantes.*

*A mon ami Benmekhlouf Amal et sa famille.*

*A toute mes amis avec les quelles j'ai partagé les bons  
moments : Sabra, Safia, Sarra, Khawla, Samra, Soumia,  
Fadila, Asma, .....*

*A tout Personne qui me connaît.*

*Meriem*

## *Dédicace*

*Je dédie ce modeste travail aux êtres les plus chers dans ma vie, mes parents à qui je dois les remercier beaucoup pour leur soutien moral et matériel, leur compréhension, leur tendresse, leur amour et leur sacrifice.*

*À mon père Brahim, ma très chère maman Zahia.*

*À ma sœur Sabrina.*

*À mes frères Riyad, Ishaq, Fayçal et Zineddine.*

*À mes très chères amies Ahlam, Karima, Imane.*

*À tous mes enseignants du primaire jusqu'aujourd'hui.*

*À mon binôme Meriem ainsi que toute sa famille.*

*À toute ma famille.*

*À tous mes collègues de la section particulièrement*

*Radouane, Nassim, Amine, Djamal.*

*À tous ceux qui me connaissent.*

*Amal*

# *Table des matières*

Remerciements.....	II
Dédicace.....	III
Dédicace.....	IV
Introduction .....	XVIII
1. Motivation.....	XVIII
2. Organisation de travail.....	XIX

## **Chapitre 01 : Etude de l'existant et objectif du projet**

Introduction .....	2
Première Partie : présentation de la wilaya .....	3
1. Présentation .....	3
2. Organigramme .....	3
2.1. Direction de l'administration locale (DAL).....	4
2.2. Direction des transmissions nationales (DTN) .....	4
3. Moyens matériels.....	5
3.1. Espaces de travail.....	5
3.2. Equipements informatique .....	5
Deuxième Partie : présentation de la commune .....	7
1. Présentation .....	7
2. Organigramme .....	7
3. Le patrimoine communal .....	8

3.1.	Les différents types de patrimoine.....	8
3.1.1.	Patrimoines mobile.....	8
3.1.2.	Patrimoines immobiliers.....	8
<b>Troisième Partie : Gestion du patrimoine communal.....</b>		<b>9</b>
<b>1. Au niveau de la commune.....</b>		<b>9</b>
1.1.	Création de cahier de charge.....	9
1.2.	Suivi du patrimoine.....	10
1.2.1.	L'ajout d'un nouveau patrimoine communal.....	10
1.2.2.	Suppression d'un patrimoine communale.....	10
1.2.3.	Location d'un patrimoine communal.....	11
1.2.4.	Renouvellement d'un contrat de location.....	11
1.2.5.	Vente d'un patrimoine communal.....	11
1.3.	Création et transmission de l'annexe C29 et l'annexe C32.....	11
<b>2. Au niveau de la Wilaya.....</b>		<b>12</b>
2.1.	Validation de cahier de charge et les actes.....	12
2.2.	Contrôle des patrimoines.....	12
2.3.	Préparation des statistiques du patrimoine communal.....	12
<b>3. Les critiques.....</b>		<b>12</b>
<b>4. Objectifs de ce projet.....</b>		<b>13</b>
<b>Conclusion.....</b>		<b>14</b>

## **Chapitre 02 : Outils de conception et modélisation**

<b>Introduction.....</b>	<b>16</b>
<b>Première Partie : Architecture client-serveur.....</b>	<b>17</b>
<b>1. Les acteurs principaux de l'architecture client-serveur.....</b>	<b>17</b>
1.1. Client.....	17

1.2.	Serveur .....	17
1.2.1.	Serveur de fichier.....	17
1.2.2.	Serveur de bases de données .....	18
1.2.3.	Serveur d'application web .....	18
1.3.	Middleware .....	19
<b>2.</b>	<b>  Fonctionnement d'architecture client-serveur .....</b>	<b>19</b>
<b>3.</b>	<b>  Type d'architecture de client-serveur .....</b>	<b>19</b>
3.1.	Architecture à 2 niveaux .....	19
3.2.	Architecture à 3 niveaux .....	20
3.3.	L'architecture multi niveaux.....	20
<b>Deuxième partie : La méthodologie 2TUP .....</b>		<b>22</b>
<b>1.</b>	<b>  Processus de développement .....</b>	<b>22</b>
1.1.	Le Processus Unifié .....	22
1.2.	Le processus 2TUP .....	23
1.2.1.	La branche gauche (fonctionnelle) .....	24
1.2.2.	La branche droite (architecture technique).....	24
1.2.3.	La branche du milieu (conception).....	24
<b>Conclusion .....</b>		<b>25</b>

## **Chapitre 03 : Analyse et conception**

<b>Introduction .....</b>	<b>27</b>
<b>Première Partie : Etude Préliminaire .....</b>	<b>28</b>
<b>1.  Elaboration du cahier des charges : .....</b>	<b>28</b>
1.1.  Présentation du projet .....	28
1.2.  Les choix techniques.....	28
1.3.  Recueil des besoins fonctionnels .....	28

1.4.	Recueil des besoins opérationnels .....	29
1.5.	Identification des acteurs .....	30
1.6.	Identification des messages .....	30
1.6.1.	Entre le système et l'administrateur .....	30
1.6.2.	Entre le système et l'utilisateur de la commune .....	31
1.6.3.	Entre le système et l'utilisateur de la wilaya .....	31
1.7.	Modélisation de contexte .....	32
1.8.	Le diagramme de contexte dynamique de système.....	32
1.9.	La signification des messages .....	33
<b>Deuxième Partie : Capture Des Besoins Fonctionnels .....</b>		<b>35</b>
<b>1.</b>	<b>Identification des cas d'utilisation .....</b>	<b>35</b>
<b>2.</b>	<b>Le diagramme de cas d'utilisation.....</b>	<b>37</b>
<b>3.</b>	<b>Description des cas d'utilisation .....</b>	<b>39</b>
3.1.	Au niveau de la commune .....	39
3.1.1.	Gestion des patrimoines.....	39
3.1.2.	Gestion des propriétaires .....	46
3.1.3.	Location .....	52
3.1.4.	Vente.....	59
3.1.5.	Recherche Multicritères commune.....	62
3.1.6.	Statistique commune.....	70
3.2.	Au niveau de la wilaya.....	75
3.2.1.	Validation des contrats .....	75
3.2.2.	Recherche Multicritères wilaya .....	77
3.2.3.	Statistique wilaya.....	84
<b>4.</b>	<b>Les règles de gestion.....</b>	<b>89</b>
<b>5.</b>	<b>Le modèle de domaine .....</b>	<b>90</b>
<b>Troisième partie : Capture des besoins technique.....</b>		<b>91</b>

<b>1. Spécification technique du point de vue matériel .....</b>	<b>91</b>
1.1. Configuration matérielle .....	91
1.2. Spécification du style d'architecture 2 niveaux .....	92
<b>2. Spécification technique du point de vue logicielle.....</b>	<b>92</b>
2.1. Les exploitants .....	93
2.2. Identification des cas d'utilisation techniques .....	93
2.3. Description des cas d'utilisation techniques .....	94
2.3.1. Cas d'utilisation technique : «Gérer la sécurité » .....	94
2.4. Organisation du modèle de spécification logicielle .....	99
<b>Quatrième partie : La phase d'analyse.....</b>	<b>100</b>
<b>1. Découpage en catégories.....</b>	<b>100</b>
1.1. Répartition des classes candidates en catégorie.....	100
1.2. Elaboration des diagrammes de classes préliminaires par catégorie .....	101
1.3. Dépendance entre catégorie .....	101
<b>2. Le développement du modèle statique .....</b>	<b>102</b>
2.1. Diagramme de classe pour la catégorie localisation.....	102
2.2. Diagramme de classe pour la catégorie gestion patrimoine .....	102
2.3. Diagramme de classe pour la catégorie gestion contrat.....	103
<b>3. Le développement du modèle dynamique.....</b>	<b>103</b>
3.1. Diagrammes de séquences détaillés.....	103
<b>Cinquième partie : Conception Préliminaire.....</b>	<b>108</b>
<b>1. Développement du modèle de déploiement.....</b>	<b>108</b>
1.1. Architecture adopté.....	108
<b>2. Développement du modèle d'exploitation.....</b>	<b>109</b>

<b>3. Définition des interfaces .....</b>	<b>110</b>
<b>4. Diagramme de composants .....</b>	<b>111</b>
<b>Sixième partie : Conception Détaillée.....</b>	<b>112</b>
<b>1. Conception des classes .....</b>	<b>112</b>
<b>2. Les opérations.....</b>	<b>114</b>
<b>3. Diagramme de classe détaillée .....</b>	<b>115</b>
<b>4. Passage vers le modèle relationnel.....</b>	<b>115</b>
4.1. Règles de passage .....	115
<b>5. Les tables de la base de données .....</b>	<b>116</b>
<b>Conclusion .....</b>	<b>117</b>

## **Chapitre 04 : Le dossier technique**

<b>Introduction .....</b>	<b>119</b>
<b>1. Présentation des outils de développement de l'application.....</b>	<b>119</b>
1.1. Le langage de programmation java .....	119
1.2. Implémentation de la base de données (Oracle) .....	119
1.3. Environnement de développement de l'application .....	120
<b>2. Description de l'application .....</b>	<b>120</b>
2.1. Application de administrateur.....	120
2.1.1. Fenêtre d'authentifier .....	120
2.1.2. Fenêtre d'accueil de l'administrateur .....	121
2.1.3. Fenêtre de gestion des utilisateurs .....	122
2.2. Application de la Wilaya .....	126
2.2.1. Fenêtre d'authentifier .....	126
2.2.2. Fenêtre d'accueil de l'utilisateur wilaya.....	126

2.3.	Application da la commune .....	127
2.3.1.	Fenêtre d’authentifier .....	127
2.3.2.	Fenêtre d’accueil de l’utilisateur commune.....	127
2.3.3.	Fenêtre de gestion du patrimoine.....	128
2.3.4.	Fenêtre de la création un contrat administratif de vente.....	128
2.3.5.	Fenêtre de la création un contrat administratif de location .....	129
2.3.6.	Fenêtre de recherche patrimoine.....	129
2.3.7.	Fenêtre de recherche contrat.....	130
2.3.8.	Fenêtre de statistique patrimoine.....	130
<b>Conclusion</b>	.....	<b>131</b>
<b>Conclusion générale</b>	.....	<b>133</b>

# Liste des figures

Figure 1 : Organigramme de la wilaya. ....	3
Figure 2: Organigramme de la direction de l'administration locale. ....	4
Figure 3: Organigramme de la direction des transmissions nationales. ....	5
Figure 4: Organigramme des Services de commune. ....	7
Figure 5: Client-serveur avec serveur de fichiers. ....	17
Figure 6: Client-serveur avec serveur de bases de données. ....	18
Figure 7: Client-serveur avec serveur d'application web. ....	18
Figure 8: Fonctionnement de l'architecture Client-serveur. ....	19
Figure 9:L'architecture à deux niveaux. ....	20
Figure 10 :l'architecture à 3 niveaux.....	20
Figure 11: architecture multi niveaux.....	21
Figure 12: Le système d'information soumis à deux natures de contraintes. ....	23
Figure 13: Le processus de développement en Y. ....	23
Figure 14: Diagramme de contexte dynamique .....	33
Figure 15: Diagramme de cas d'utilisation. ....	38
Figure 16: Diagramme d'activité «Gestion_patrimoine». ....	43
Figure 17: Diagramme de séquence «Gestion_patrimoine». ....	44
Figure 18: Diagramme de séquence «Ajouter_patrimoine». ....	45
Figure 19: Diagramme de séquence «Modifier_patrimoine».....	45
Figure 20: Diagramme de séquence «Supprimer_patrimoine».....	45
Figure 21: Diagramme d'activité «Gestion_propriétaires».....	49
Figure 22: Diagramme de séquence «Gestion propriétaire».....	50
Figure 23: Diagramme de séquence «Ajouter propriétaire».....	50
Figure 24: Diagramme de séquence «Modifier propriétaire».....	51
Figure 25: Diagramme de séquence «Supprimer propriétaire».....	51
Figure 26: Diagramme d'activité «Gestion_contrats_location».....	56
Figure 27: Diagramme de séquence «Location».....	57
Figure 28: Diagramme de séquence «Ajouter contrat_location».....	57
Figure 29: Diagramme de séquence «Modifier contrat_location».....	58
Figure 30: Diagramme de séquence «Supprimer contrat_location».....	58
Figure 31: Diagramme d'activité «Recherche_Multicritères_commune».....	66
Figure 32: Diagramme d'activité «Recherche_propriétaire».....	67
Figure 33: Diagramme de séquence «Recherche_Multicritères_commune».....	67

Figure 34: Diagramme de séquence «Rechercher_patrimoine_commune».	68
Figure 35: Diagramme de séquence «Rechercher_contrat_commune».	69
Figure 36: Diagramme de séquence «Rechercher_propriétaire_commune».	70
Figure 37: Diagramme d'activité «Statistique_commune».	73
Figure 38: Diagramme de séquence «Statistique_commune».	73
Figure 39: Diagramme de séquence «Statistique_patrimoine».	74
Figure 40: Diagramme de séquence «Statistique_contrat».	74
Figure 41: Diagramme d'activité «Validation des contrats».	76
Figure 42: Diagramme de séquence «Valider des contrats».	76
Figure 43: Diagramme d'activité «Recherche_Multicritères_wilaya».	81
Figure 44: Diagramme d'activité «Recherche_propriétaire».	81
Figure 45: Diagramme de séquence «Recherche_Multicritères_wilaya».	82
Figure 46: Diagramme de séquence «Rechercher_patrimoine_wilaya».	82
Figure 47: Diagramme de séquence «Recherche_contrat_wilaya».	83
Figure 48: Diagramme de séquence «Recherche_propriétaire».	84
Figure 49: Diagramme d'activité «statistique_wilaya».	87
Figure 50: Diagramme de séquence «statistique_wilaya».	88
Figure 51: Diagramme de séquence «statistique_Patrimoine».	88
Figure 52: Diagramme de séquence «statistique_contrat».	89
Figure 53: le modèle de domaine.	90
Figure 54: Architecture 2 niveaux de notre système.	92
Figure 55: Modèle de spécification logicielle du système (diagramme de cas technique).	94
Figure 56: Diagramme d'activité « Gérer la sécurité (s'authentifier)»	95
Figure 57: Diagramme de séquence « Gérer la sécurité (s'authentifier)».	95
Figure 58: Diagramme d'activité «gestion_utilisateurs».	99
Figure 59: Organisation du modèle de spécification logicielle.	99
Figure 60: Le découpage en catégories.	100
Figure 61: Diagramme de classes préliminaires par catégorie.	101
Figure 62: Dépendance entre catégorie.	101
Figure 63: Diagramme de classe pour la catégorie localisation.	102
Figure 64: Diagramme de classe pour la catégorie gestion patrimoine.	102
Figure 65: Diagramme de classe pour la catégorie gestion contrat.	103
Figure 66: Diagramme d'interaction « Ajouter patrimoine».	104
Figure 67: Diagramme d'interaction « Modifier patrimoine ».	105
Figure 68: Diagramme d'interaction « Supprimer patrimoine ».	106
Figure 69: Diagramme d'interaction « Consulter patrimoine ».	107
Figure 70: schéma du modèle de déploiement de notre système.	109

Figure 71: Définition des applications dans le modèle d'exploitation. ....	109
Figure 72: Diagramme de composants. ....	111
Figure 73: Diagramme de classes détaillé de notre système. ....	115
Figure 74: Fenêtre d'identification administrateur. ....	120
Figure 75: Message d'erreur. ....	121
Figure 76 : Fenêtre d'accueil administrateur. ....	121
Figure 77 : Fenêtre de gestion des utilisateurs. ....	122
Figure 78 : Fenêtre d'ajouter un utilisateur. ....	123
Figure 79 : Fenêtre de modifier un utilisateur. ....	124
Figure 80 : Fenêtre de supprimer un utilisateur. ....	125
Figure 81 : Message de confirmation. ....	125
Figure 82 : Fenêtre d'identification utilisateur wilaya. ....	126
Figure 83 : Fenêtre d'accueil utilisateur wilaya. ....	126
Figure 84 : Fenêtre d'identification utilisateur commune. ....	127
Figure 85 : Fenêtre d'accueil utilisateur commune. ....	127
Figure 86: Fenêtre de gestion du patrimoine. ....	128
Figure 87: Fenêtre de la création un contrat de vente. ....	128
Figure 88: Fenêtre de la création un contrat de location. ....	129
Figure 89 : Fenêtre de recherche patrimoine. ....	129
Figure 90 : Fenêtre de recherche contrat. ....	130
Figure 91 : Fenêtre de statistique patrimoine. ....	130

# Liste des Tableaux

Tableau 1: Espaces de travail. ....	5
Tableau 2: Les serveurs du service informatique. ....	5
Tableau 3: les micros ordinateurs du service informatique.....	6
Tableau 4: les imprimantes du service informatique .....	6
Tableau 5 : modélisation de contexte. ....	32
Tableau 6: Signification des messages. ....	34
Tableau 7: Liste des acteurs et des messages par cas d'utilisation.....	37
Tableau 8: Description textuelle du cas d'utilisation « gestion patrimoine».....	39
Tableau 9: Description textuelle du cas d'utilisation «Ajout_patrimoine».....	40
Tableau 10: Description textuelle du cas d'utilisation «Modification_patrimoine».....	41
Tableau 11: Description textuelle du cas d'utilisation «Suppression_patrimoine».....	42
Tableau 12: Description textuelle du cas d'utilisation «Consultation_patrimoine».....	42
Tableau 13: Description textuelle du cas d'utilisation «gestion_propriétaires ».....	46
Tableau 14: Description textuelle du cas d'utilisation «Ajout_propriétaire».....	47
Tableau 15: Description textuelle du cas d'utilisation«Modification_propriétaire ».....	47
Tableau 16: Description textuelle du cas d'utilisation «Suppression_propriétaire ».....	48
Tableau 17: Description textuelle du cas d'utilisation «Consultation_propriétaire».....	49
Tableau 18: Description textuelle du cas d'utilisation «Location».....	52
Tableau 19: Description textuelle du cas d'utilisation «Création_contrat_location».....	53
Tableau 20: Description textuelle du cas d'utilisation «Modification_contrat_location».....	54
Tableau 21: Description textuelle du cas d'utilisation «Suppression_contrat_location».....	54
Tableau 22: Description textuelle du cas d'utilisation «Consultation_contrat_location».....	55
Tableau 23: Description textuelle du cas d'utilisation «Vente».....	59
Tableau 24: Description textuelle du cas d'utilisation «Création_contrat_vente».....	60
Tableau 25: Description textuelle du cas d'utilisation «Modification_contrat_vente».....	61
Tableau 26: Description textuelle du cas d'utilisation «Suppression_contrat_vente».....	61
Tableau 27: Description textuelle du cas d'utilisation «Consultation_contrat_vente».....	62
Tableau 28: Description textuelle du cas d'utilisation «Recherche_Multicritères_commune».....	63
Tableau 29: Description textuelle du cas d'utilisation «Recherche_patrimoine».....	64
Tableau 30: Description textuelle du cas d'utilisation «Recherche_contrat».....	65
Tableau 31: Description textuelle du cas d'utilisation «Recherche_contrat».....	66
Tableau 32: Description textuelle du cas d'utilisation « statistique commune».....	70
Tableau 33: Description textuelle du cas d'utilisation « statistique patrimoine».....	71
Tableau 34: Description textuelle du cas d'utilisation « statistique contrat».....	72

Tableau 35: Description textuelle du cas d'utilisation « validation des contrats».....	75
Tableau 36: Description textuelle du cas d'utilisation «Recherche_Multicritères_wilaya».....	77
Tableau 37: Description textuelle du cas d'utilisation «Recherche_patrimoine_wilaya».....	78
Tableau 38: Description textuelle du cas d'utilisation «Recherche_contrat_wilaya».....	79
Tableau 39: Description textuelle du cas d'utilisation «Recherche_propriétaire».....	80
Tableau 40: Description textuelle du cas d'utilisation « statistique wilaya».....	85
Tableau 41: Description textuelle du cas d'utilisation « statistique patrimoine».....	86
Tableau 42: Description textuelle du cas d'utilisation « statistique contrat».....	87
Tableau 43: Description textuelle du cas d'utilisation « authentification».....	95
Tableau 44: Description textuelle du cas d'utilisation «gestion_utilisateurs ».....	96
Tableau 45: Description textuelle du cas d'utilisation «Création_utilisateur».....	97
Tableau 46: Description textuelle du cas d'utilisation «Modification_utilisateur ».....	98
Tableau 47: Description textuelle du cas d'utilisation «Suppression_utilisateur ».....	98
Tableau 48: Les interfaces de notre système.....	110
Tableau 49: Conception des classes.....	113
Tableau 50: Les opérations.....	114

# Introduction générale

## Introduction

Les développements dans l'ère actuelle , surtout ces derniers temps qui a touché tous les domaines de la vie économique , politique et sociale , où il a contribué à ce développement spectaculaire et significative dans le développement des entreprises et des administrations de différents types et domaines de spécialisation , qui a également conduit à la mise au point et les fonctions de soutien administratif , surtout après la découverte de l'Internet , et c'est la dernière révolution dans l'accès à l'information, qui nécessite de trouver un système rapide et sophistiquées dans le domaine des médias.

Durant le stage pratique que nous avons effectué dans le cadre du projet de fin d'études qui s'est déroulé au sein de la Wilaya de Mila, et après les entretiens avec le personnel du service de l'animation communale des marchés, nous avons constaté que tout le travail est réalisé de façon manuelle pour la gestion du patrimoine communal. C'est pourquoi nous avons proposé de réaliser une application client-serveur pour la gestion du patrimoine au niveau de la wilaya du Mila pour obtenir la plus grande quantité d'informations de manière organisée dans les plus brefs délais et à moindre coût et effort.

Notre projet consiste à réaliser une application serveur située à la wilaya et des applications clientes situées dans les trente-deux communes. Nous avons utilisé le langage de modélisation **UML**, qui est un outil très puissant dans le domaine de développement, ainsi on a utilisé le processus **2TUP** comme méthode de développement pour représenter les différents aspects de notre application selon la conception spécifiée.

Pour l'implémentation du projet, nous avons adopté le langage **JAVA**, qui est l'un des langages de développement les plus répandus et le **SGBD ORACLE** pour implémenter la base de données.

### 1. Motivation

Plusieurs facteurs peuvent inciter le service de l'animation communale des marchés à la mise en place d'une application client/serveur :

- Améliorer l'opération de gestion et suivi du patrimoine communale, et réduire le temps d'envoi des différentes informations par chaque commune.
- Faciliter la mise à jour du patrimoine (ajout, modification, suppression).

- Faciliter l'accès aux informations et fournir une protection nécessaire pour établir une base de données sécurisée.
- Réduire le temps de traitement et la possibilité de faire des erreurs.
- Faciliter l'opération de recherche (contrats, patrimoines,...etc.) au niveau de la Wilaya et la commune.

## 2. Organisation de travail

Notre mémoire est constitué de quatre chapitres :

### **Chapitre 01 : Etude de l'existant et objectif du projet**

Ce chapitre est structuré en trois parties, la première partie pour la présentation de la structure de la Wilaya de Mila. La deuxième partie pour la structure de la commune, et la troisième partie est consacrée à la gestion du patrimoine communal entre la wilaya et les communes.

### **Chapitre 02 : Outils de conception et modélisation**

Ce chapitre est structure en deux parties, dans la première partie nous présenterons l'architecture client-serveur et la deuxième partie pour les concepts fondamentaux de processus de développement 2TUP que nous avons adapté pour modéliser notre système.

### **Chapitre 03 : Analyse et conception**

Ce chapitre englobe toutes les phases d'analyse et conception de notre étude de cas. Nous l'avons organisé en parties en fonction des activités d'ingénierie préconisées dans la méthode.

- La première partie **étude préliminaire** représente la première étape du processus 2TUP, dans cette partie, nous allons élaborer le cahier des charges qui contient les activités formelles de capture des besoins fonctionnels et de capture des besoins techniques.
- La deuxième partie **la capture des besoins fonctionnel** représente la branche gauche du cycle en Y puisqu'elle décrit les différentes fonctionnalités du système et la façon de les utiliser. Dans cette partie, nous complétons le recueil des besoins ébauchés lors de l'étude préliminaire.
- La troisième partie c'est **la capture des besoins techniques**. nous avons cité les spécifications techniques du système (l'architecture physique ainsi que les cas d'utilisation techniques).

- La quatrième partie est **la phase d'analyse**. Dans cette phase, nous découpons notre système en catégories et définissons les classes d'analyse du système, le modèle statique et le modèle dynamique.
- La cinquième partie est **la conception préliminaire**. Dans cette partie, nous allons fusionner les résultats des études fonctionnelles et techniques pour présenter le modèle de conception.
- La sixième partie est **la conception détaillée** qui élabore les spécifications des classes et des associations qui construiront notre système.

#### **Chapitre 04 : Le dossier technique**

Ce dernier chapitre décrit la phase réalisation de notre projet, nous allons présenter les différents outils et techniques utilisées pour l'implémentation et exposer quelques interfaces de notre l'application.

# Chapitre

*ETUDE DE L'EXISTANT*

*ET*

*OBJECTIF DU PROJET*

1

## **Introduction**

Dans ce chapitre, nous allons présenter dans la première partie la structure de la Wilaya de Mila. Pour cela, nous commencerons par une présentation générale, ensuite l'organigramme de la Wilaya et les moyens humains et matériel. Dans la deuxième partie, nous présenterons la structure de la commune. Nous commencerons par une présentation générale, l'organigramme de la commune et la définition et l'utilisation des patrimoines de commune. Enfin, nous allons représenter dans la troisième partie la gestion du patrimoine communal entre la wilaya et les communes.

## Première Partie : présentation de la wilaya

### 1. Présentation

La wilaya de Mila est créée lors du dernier découpage administratif Algérien de 1984, avec la ville de MILA comme chef-lieu de la wilaya 43. La wilaya de Mila est située dans le Nord-est Algérien à 464 m d'altitude, et à 33 KM de la mer Méditerranée.

La wilaya de Mila a connu plusieurs walis depuis sa création. En plus, elle est formée de 13 daïras et de 32 communes [7].

### 2. Organigramme

La wilaya de Mila est constituée de 3 directions et une inspection :

- DAL : Direction de l'administration locale.
- DRAG : Direction de la réglementation et des affaires générales.
- DTN : Direction des transmissions nationales.
- L'inspection Général (voir figure 1).

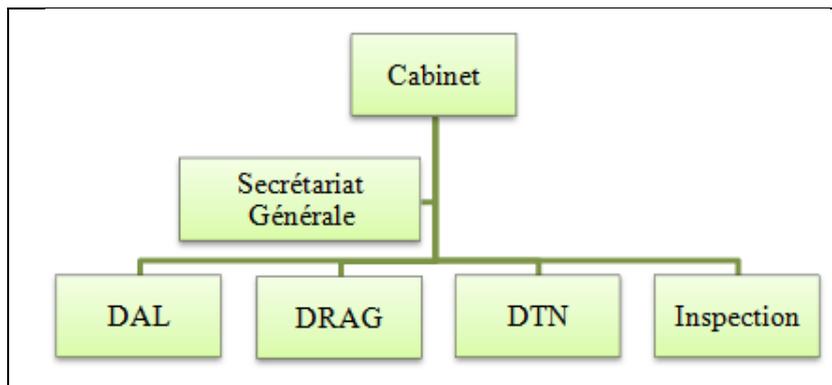


Figure 1 : Organigramme de la wilaya.

Notre stage s'est déroulé dans le bureau du patrimoine communal du service de l'animation communale des marchés de la direction de l'administration locale (DAL). En plus, nous avons utilisé les moyens matériels (Serveur base de données, équipement réseau, etc.) du service informatique de la DTN (Direction des Transmission Nationales) de la wilaya de Mila.

## 2.1. Direction de l'administration locale(DAL)

Direction de l'administration locale (DAL) est composée de trois services :

1. Services des budgets et du patrimoine.
2. Services de personnels de la wilaya.
3. Services de l'animation communale des marchés (voir figure 2).

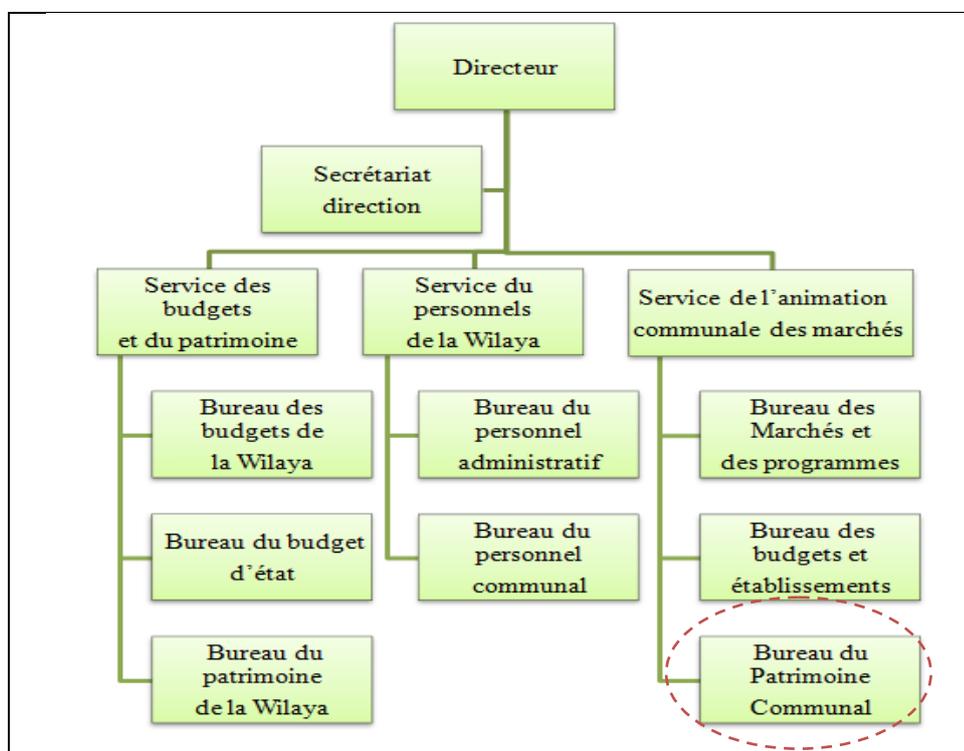


Figure 2: Organigramme de la direction de l'administration locale.

## 2.2. Direction des transmissions nationales(DTN)

Direction des transmissions nationales (DTN) est composée de quatre services:

1. Services informatique.
2. Services technique.
3. Services exploitation.
4. Services de l'administration et de la logistique(voir figure 3).

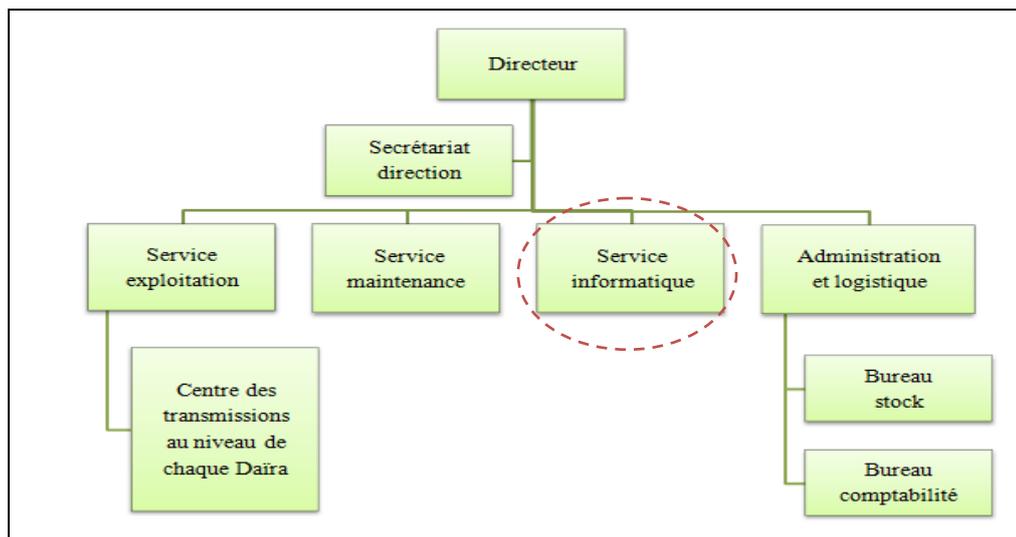


Figure 3: Organigramme de la direction des transmissions nationales.

### 3. Moyens matériels

Dans ce qui suit nous allons présenter les moyens matériels du service informatique.

#### 3.1. Espaces de travail

Espaces de travail	Quantités	Surface
Salle	01	41,50 m <sup>2</sup>
Salle	01	22,50 m <sup>2</sup>

Tableau 1: Espaces de travail.

#### 3.2. Equipements informatique

- Serveur

Marque	Quantités	Capacité	Ressource
<b>Compaq ML370</b>	01	Bi-processeur Pentium III 1Go et 128Ram	Disquedur 54Go stramers 20/40Go
<b>Hp Proliant DL 380 G6</b>	02	Intel(R) XEM(R) 2.77Go et 4RAM	4 disques durs 146 Go
<b>Hp Proliant ML 350</b>	01	Intel(R) XEM(R) 2Go et 2RAM	3 disques durs 72 Go

Tableau 2: Les serveurs du service informatique.

- **Micro-ordinateurs**

Modèle et marque	Pentium IV	Pentium IV
<b>Processeur</b>	2 GHz	2.71 GHz
<b>RAM</b>	1 Go	1 Go
<b>Disque dur</b>	150 Go	140 Go
<b>Système d'exploitation</b>	WINDOWS XP	WINDOWS 7
<b>Quantité</b>	8	2

**Tableau 3: les micros ordinateurs du service informatique**

- **Les imprimantes**

Marque	Quantités
Manesmanne Tallay MT 621	2
HP Deskjet 1050	1
Canon	2

**Tableau 4: les imprimantes du service informatique**

- **Réseaux informatiques**

- ✓ Matériel actif : 3 Switch Cisco CATALIST 24 ports 10/100 BTX.
- ✓ Firewall Cisco PIX 515 (2) ports 10/100 BTX.
- ✓ Routeur Cisco 2620 (02) ports Wan n\*64K 02MBPS 08lignes asynchrones modem intégrés.
- ✓ Serveur d'accès distant FTP.
- ✓ Serveur d'accès distant pour le web.
- ✓ Serveur de base de données.
- ✓ Réseau haut débit.

- **Logiciels**

- ✓ Windows server 2000, 2003,2008.
- ✓ Logiciels de programmation utilisés : Visual Basic 6, Delphi 7.
- ✓ SGBD: SQL Server 2000, Firebird.
- ✓ Microsoft office 2003, 2007, 2010.

## Deuxième Partie : présentation de la commune

### 1. Présentation

La commune est la collectivité territoriale de base de l'état algérien, à la fois collectivité disposant de la personnalité morale, dotée de ses propres organes, délibératif et exécutif, et plus petite subdivision administrative de l'organisation territoriale de l'Algérie. Cette double compétence de la commune est exercée par le président de l'Assemblée Populaire Communale (A.P.C), qui est conjointement le représentant de la commune et le représentant de l'État au niveau communal. [8]

### 2. Organigramme

Chaque commune est constituée de 5 services :

- Service règlementation générale.
- Service d'administration et moyens généraux.
- Service des affaires sociales culturelles et sportives.
- Service urbanisme et construction et équipement.
- Service des routes et réseau divers biens (voir figure 4).

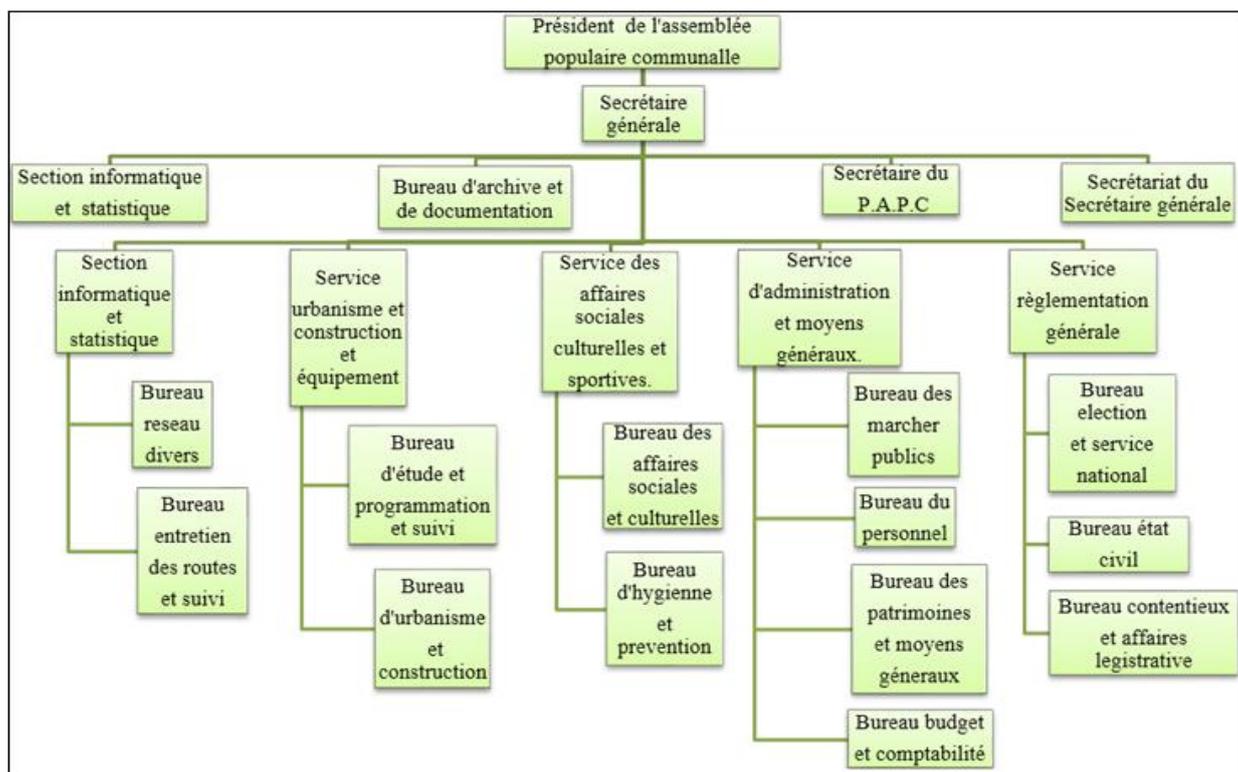


Figure 4: Organigramme des Services de commune.

### **3. Le patrimoine communal**

Le patrimoine est défini comme l'ensemble des biens possédés à un moment donné (par une personne, une famille, une entreprise, une collectivité, une nation) et ayant une valeur économique.

#### **3.1. Les différents types de patrimoine**

##### **3.1.1. Patrimoines mobile**

Les patrimoines mobiles sont des patrimoines physiques qui peuvent être transportés d'un lieu à un autre.

Comme exemple de patrimoines mobiles communaux, nous pouvons citer: Le matériel roulant, les engins de travaux publics, le matériel informatique, les équipements de bureau etc.

##### **3.1.2. Patrimoines immobiliers**

Sont dits immobiliers, les patrimoines qui ne peuvent être déplacés. Les patrimoines immobiliers communaux sont habituellement constitués des terrains et des ouvrages tels que les logements, les abattoirs, les hangars de marchés, les boutiques, etc.

Chaque type de patrimoines (mobiles, immobiliers) est divisé en deux catégories selon sa productivité :

- **Le patrimoine produisant des bénéfices**
  - Comme exemple de patrimoine immobile produisant des bénéfices : les maisons, les marchés communaux, les abattoirs, les parcs de stationnement, etc.
  - Comme exemple de patrimoine mobile produisant des bénéfices: les véhicules, les engins des travaux publics, divers équipements etc.
- **Le patrimoine non productif des bénéfices**
  - Comme exemple de patrimoine immobile non productif: Les écoles, les salles de soins, les maisons de jeunes, les bibliothèques, les monuments historiques et les statues, le réseau routier communale, les espaces verts, l'éclairage public etc.

## Troisième Partie : Gestion du patrimoine communal

Nous allons présenter dans cette partie les opérations effectuées par l'utilisateur de la commune (création de cahier de charge, suivi du patrimoine, création et transmission des annexes C29, C32) et les opérations effectuées par l'utilisateur de la wilaya (validation de cahier de charge, control de patrimoine, préparation des statistiques du patrimoine communal).

### 1. Au niveau de la commune

La gestion du patrimoine communal fait par les agents du service de l'administration et moyen généraux et exactement bureau des patrimoines et moyens généraux section d'inventaire et gestion patrimoine de la commune sous la supervision du chef service.

#### 1.1. Création de cahier de charge

Un cahier des charges est un document contractuel définissant de façon exhaustive ce que le propriétaire attend de la réalisation d'un produit ou d'un service. Il est composé de plusieurs éléments correspondant aux différents aspects du projet :

- La description du contexte (politique, stratégique, commercial...) de réalisation du projet et ses objectifs.
- Les modalités d'exécution.
- Le périmètre du projet : les ressources et le nombre de personnes à mobiliser pour sa réalisation.
- Les contraintes techniques : économiques (budget), environnementales, humaines (contraintes particulières dans le cas d'un objet pour enfant...), industrielles et matérielles.

La création du cahier de charge est faite après chaque location ou vente d'un patrimoine, et passe par les étapes suivantes :

- Etablir le cahier de charge qui doit contenir toutes les informations du patrimoine qui peut être loué ou vendu.
- La validation du cahier de charge par l'assemblée populaire communale élargie présidée par le président de l'APC.

- Envoi du cahier de charge à la wilaya et attente d'une période de temps (30 jours) à partir de dépôt cahier de charge dans la wilaya.

S'il n'y a aucune réponse, la commune considère que le cahier de charge est validé.

## **1.2. Suivi du patrimoine**

Pour un bon suivi du patrimoine communal, il est indispensable de tenir les registres légaux à jour (Registre d'inventaire, Sommier de consistance, Registre des contrats administratifs).

- **Sommier de consistance (annexe n°29)**

C'est un registre qui contient toute les informations sur les patrimoines immobiliers. Il est dévissé en deux tableaux : le premier tableau spécial pour le patrimoine produisant des bénéfices et le deuxième pour le patrimoine non productif des bénéfices. Ce registre doit être numéroté, signé par le PAPC (Président Assemblé Populaire Communal).

- **Registre d'inventaire (annexe n°32)**

C'est un registre qui contient toute les informations sur les patrimoines mobiles. Ce registre doit être numéroté, signé par le PAPC (Président Assemblé Populaire Communal).

- **Registre des contrats administratifs**

C'est un registre qui contient toutes les informations sur les contrats de vente et location des patrimoines de la commune.

Les principales opérations pour un bon suivi des patrimoines sont :

### **1.2.1. L'ajout d'un nouveau patrimoine communal**

L'ajout d'un nouveau patrimoine communal se fait manuellement où on ajoute toutes les informations du patrimoine dans le sommier de consistance si ce patrimoine est de type immobile, il est classifié dans deux tableaux selon la catégorie du patrimoine produisant des bénéfices et non productif ou dans le registre d'inventaire si ce patrimoine est de type mobile.

### **1.2.2. Suppression d'un patrimoine communale**

Cette opération ne peut être effectuée sauf dans le cas du total dommage du patrimoine communal et devient inutilisable.

### **1.2.3. Location d'un patrimoine communal**

La location est l'action qui consiste à louer un logement, un véhicule, ou tout autre patrimoine mobile ou immobile à un propriétaire, et elle est faite après la création et la validation du cahier de charge.

- **Création et inscription d'un nouveau contrat de location d'un patrimoine**

La création du contrat se fait manuellement où on ajout toutes les informations sur le patrimoine à louer ainsi que la personne qui veut le louer dans le contrat.

On doit aussi ajouter toutes les informations du nouveau contrat de location d'un patrimoine dans le registre des contrats administratifs.

### **1.2.4. Renouvellement d'un contrat de location**

Le renouvellement du contrat de location se fait à la fin d'une période du contrat de location convenu, où la commune notifie le locataire avant 30 jours de la date de fin de location.

Si le locataire veut renouveler le contrat de location la commune créé un autre contrat de location.

On doit aussi ajouter toutes les informations du nouveau contrat de renouvellement d'un patrimoine dans le registre des contrats administratifs.

### **1.2.5. Vente d'un patrimoine communal**

Cette opération est faite sur les patrimoines tel que les terres, après la création et la validation du cahier de charge.

- **Création et inscription d'un nouveau contrat de vente d'un patrimoine**

La création du contrat se fait manuellement où on ajoute toutes les informations concernant le patrimoine à vendre et la personne acheteuse du patrimoine dans le contrat.

On doit aussi ajouter toutes les informations de nouveau contrat de vente d'un patrimoine dans le registre des contrats administratifs.

### **1.3. Création et transmission de l'annexe C29 et l'annexe C32**

La création et la transmission de l'annexe C29 et l'annexe C32 est faite de façon périodique chaque année tel que :

- L'annexe C29 est une copie de sommier de consistance (annexe n°29), est une partie de budget communal.

- L'annexe C32 est une copie de registre d'inventaire (annexe n°32), est une partie de budget communal.
- Le budget communal est l'état de prévisions des recettes et des dépenses annuelles de la commune.

## **2. Au niveau de la Wilaya**

La gestion du patrimoine communal est faite manuellement par le chef bureau de patrimoine communale du service de l'animation locale des marchés de la direction de l'administration locale (DAL).

### **2.1. Validation de cahier de charge et les actes**

Après la réception du cahier de charge, le chef du bureau l'étudie et le valide, s'il trouve des erreurs dans le contenu, il le renvoi à la commune pour préparer un autre cahier de charge. Sinon, il le stocke dans l'archive de la wilaya.

### **2.2. Contrôle des patrimoines**

Cette opération est faite par un comité de control composé de plusieurs personnes et son rôle est de contrôler les informations des registres d'inventaire et le sommier de consistance pour s'assurer qu'il est correct.

### **2.3. Préparation des statistiques du patrimoine communal**

Cette opération est faite manuellement pour calculer les statistiques de chaque patrimoine, et les mettre dans des tableaux selon plusieurs critères (par type, par date,...etc.) on utilise l'annexe C29, C32 pour extraire les informations.

## **3. Les critiques**

De notre étude analytique du domaine de la gestion du patrimoine, faite au niveau de la wilaya de Mila ainsi que quelques communes de références (Ferdjioua, Mila), nous avons acquis un ensemble d'informations d'une part et extrait les obstacles et les inconvénients de la procédure de la gestion du patrimoine communal faite actuellement entre la wilaya et les différentes communes, d'autre part. Parmi ces problèmes, on peut noter:

- **Au niveau de la commune**

- L'opération de gestion et de suivi patrimoine communale est une opération faite manuellement ce qui incite une grande lenteur de traitement (recherche d'un patrimoine, d'un contrat ou d'un propriétaire, ajout de contrat, etc.) et nécessite beaucoup de matériel et de personnel (espace d'archivage, une grande de masse de papier, etc.). Cela est dû au grand nombre de communes dans la wilaya où chacune possède son propre patrimoine.
- L'échange d'informations et de documents entre la wilaya et les commune prend beaucoup de temps, car on n'utilise pas les moyens technologique que dispose la wilaya (Matériel informatique, réseau local, fibre optique, serveurs, etc.).

- **Au niveau de la Wilaya**

- Difficulté d'accès aux documents avec la possibilité de perte ou de destruction en raison de la distance entre la wilaya et quelques communes (Exemple : MinarZareza se situe à 80 Km du siège de la wilaya) et aussi des conditions climatiques.
- Difficulté de recherche et d'obtention d'informations sur un patrimoine spécifique pour une commune donnée.
- Perte de documents due à l'absence d'organisation et d'archivage structuré.
- Problème de calcule et statistiques.

À partir des critiques mentionnés ci-dessus, nous proposons un ensemble de propositions pour faciliter la gestion du patrimoine aussi bien au niveau des communes qu'au niveau de la wilaya.

#### **4. Objectifs de ce projet**

Le but de cette étude est de réaliser une application client/serveur de gestion du patrimoine communal pour faciliter l'opération de gestion et gagner du temps. De ce fait, nous allons proposer et implémenter trois applications :

- La **1<sup>ere</sup>** c'est l'application cliente commune devant être installée au niveau de chaque commune de wilaya de Mila, elle servira à effectuer la gestion du patrimoine (ajout, modification, suppression, consultation), et la gestion des contrats administrative de vente ou de location (ajout, modification, suppression, renouvellement, consultation), et la consultation des statistiques des patrimoines et des contrats de la commune, et la recherche d'un patrimoine, d'un contrat ou d'un propriétaire .
- La **2<sup>eme</sup>** c'est l'application client wilaya devant être installée au niveau de la Wilaya, elle servira à effectuer la consultation des statistiques des patrimoines, des contrats des

communes de la Wilaya et la recherche d'un patrimoine, propriétaire ou contrat et la validation des contrats administratifs de vente et location.

- La 3<sup>eme</sup> c'est l'application administrateur devant être installée au niveau de la Wilaya, elle servira à effectuer la gestion des utilisateurs (création, modification, suppression, consultation) etc.

## **Conclusion**

Dans ce chapitre, nous avons présenté l'organisation et les moyens humains et matériels de la wilaya de Mila et l'organisation de la commune qui est l'environnement de notre système, ensuite nous avons expliqué le système de gestion du patrimoine communale.

Pour développer notre application de gestion du patrimoine communale en ligne on a choisi d'utiliser l'architecture client-serveur afin d'assurer une meilleure sécurité et fiabilité ainsi que l'incité des informations car ces derniers sont situées juste dans le serveur. Pour cela, dans ce le chapitre qui suit, nous présentons l'architecture client-serveur et la méthodologie 2TUP.

# Chapitre

*OUTILS DE CONCEPTION*

*ET*

*MODELISATION*

2

## Introduction

Ce chapitre est divisé en deux parties, Dans la première partie nous allons présenter l'architecture client-serveur. Pour cela, nous commencerons par la définition et les caractéristiques et le fonctionnement de l'architecture client-serveur, ensuite on présentera les différents types de serveur, d'architecture client-serveur. Et dans la deuxième partie nous allons expliquer la méthodologie de développement 2TUP. Dans cette partie nous commencerons par une présentation de processus de développement 2TUP, ensuite en présenter chaque étape du ce processus.

## Première Partie : Architecture client-serveur

L'architecture client-serveur est une architecture de réseau dans laquelle les traitements sont répartis entre les clients qui demandent aux serveurs les informations dont ils ont besoin. C'est une architecture qui désigne un mode de communication entre plusieurs ordinateurs d'un réseau qui distingue un ou plusieurs postes clients du serveur. [1]

### 1. Les acteurs principaux de l'architecture client-serveur

Les acteurs principaux de l'architecture client-serveur sont au nombre de trois :

#### 1.1. Client

Un client est un système (programme, ordinateur) demandant l'exécution d'une opération à un fournisseur des services par l'envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour [5].

#### 1.2. Serveur

Un serveur est un système (programme, ordinateur de grande capacité) détenant des ressources qu'il met à la disposition des autres ordinateurs (clients) d'un réseau.

Nous distinguons plusieurs types de serveur en fonction des services rendus [5]:

##### 1.2.1. Serveur de fichier

Le client (généralement un PC) requiert des enregistrements de fichiers en émettant des requêtes sur le réseau en direction d'un serveur de fichier (voir figure 5).

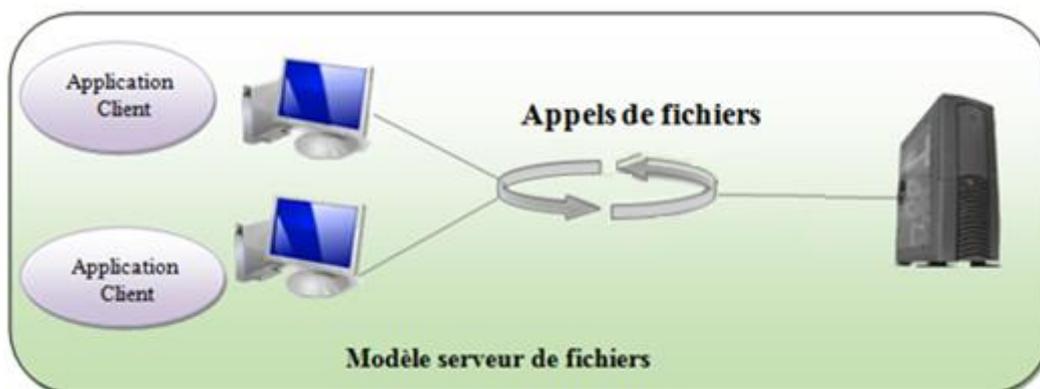


Figure 5: Client-serveur avec serveur de fichiers.

### 1.2.2. Serveur de bases de données

Le client émet des requêtes SQL sous forme de message. Le serveur renvoie le résultat de chaque requête (voir figure 6).

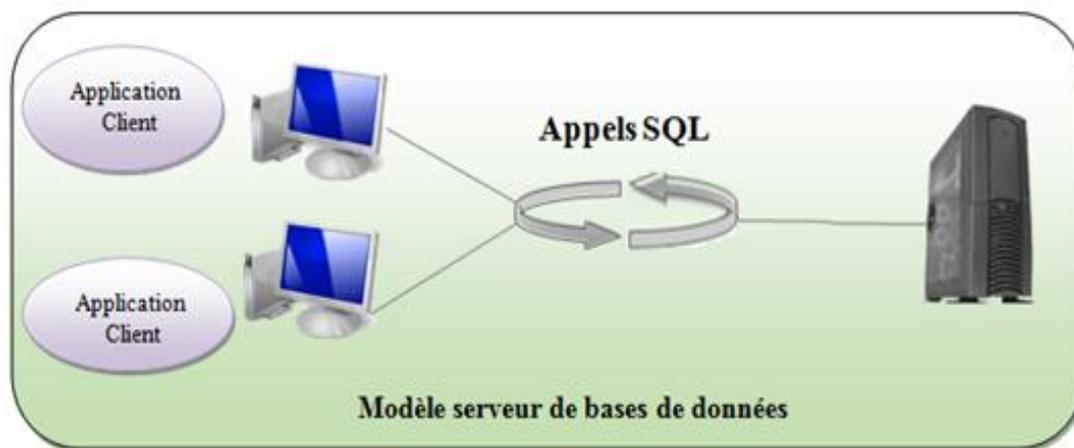


Figure 6: Client-serveur avec serveur de bases de données.

### 1.2.3. Serveur d'application web

Le World Wide Web est la première véritable application client-serveur intergalactique. Ce nouveau modèle consiste en des clients qui communiquent avec des très gros serveurs. Dans sa concrétisation la plus simple, un serveur web renvoie des documents lorsque le client les demande par leur nom. (Voir figure 7). [6]

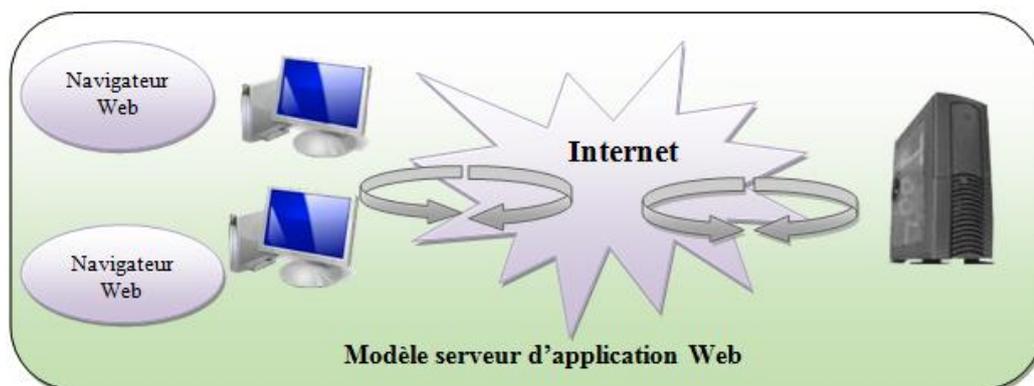


Figure 7: Client-serveur avec serveur d'application web.

### 1.3. Middleware

Le middleware est l'ensemble des services logiciels qui assurent l'intermédiaire entre les applications et le transport de données dans le réseau afin de permettre les échanges des requêtes et des réponses entre client et serveur de manière transparente. [1]

## 2. Fonctionnement d'architecture client-serveur

Un système client-serveur fonctionne selon le schéma suivant (voir figure 8) :

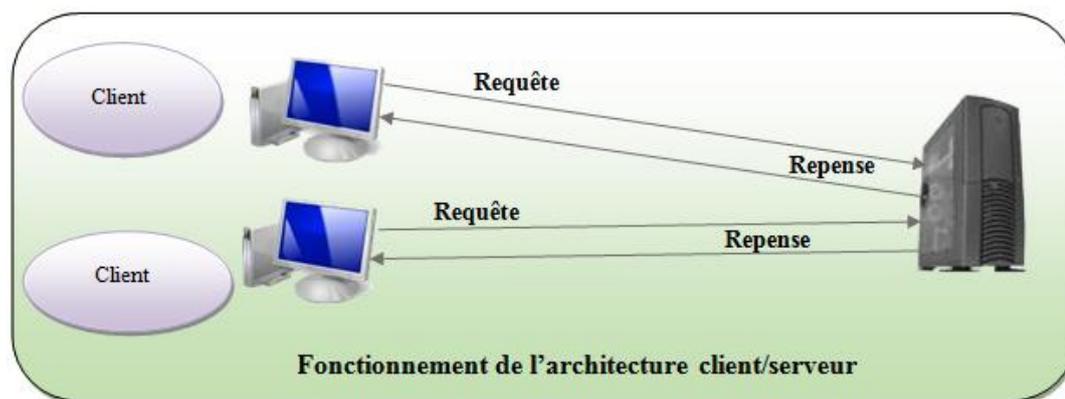


Figure 8: Fonctionnement de l'architecture Client-serveur.

- Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port. [3]

## 3. Type d'architecture client-serveur

### 3.1. Architecture à 2 niveaux

L'architecture à deux niveaux (aussi appelée architecture 2-tiers) caractérise les systèmes clients-serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service (voir figure 9). [3]

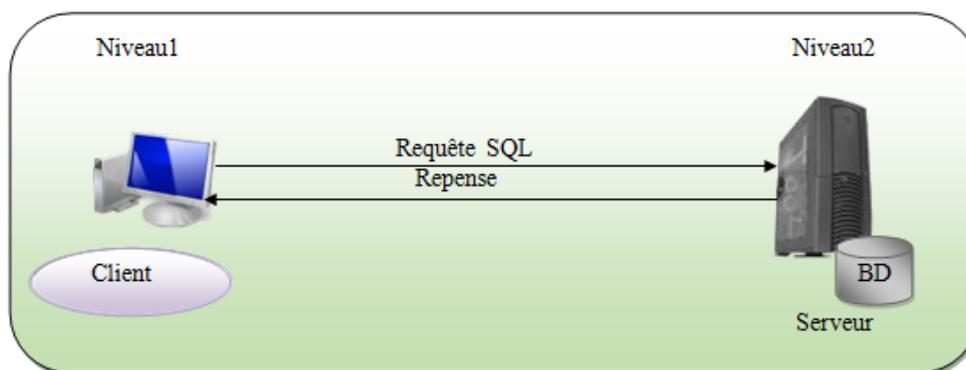


Figure 9:L'architecture à deux niveaux.

### 3.2. Architecture à 3 niveaux

Dans l'architecture à 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

- Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
- Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur ;
- Le serveur de données, fournissant au serveur d'application les données dont il a besoin (voir figure 10). [3]

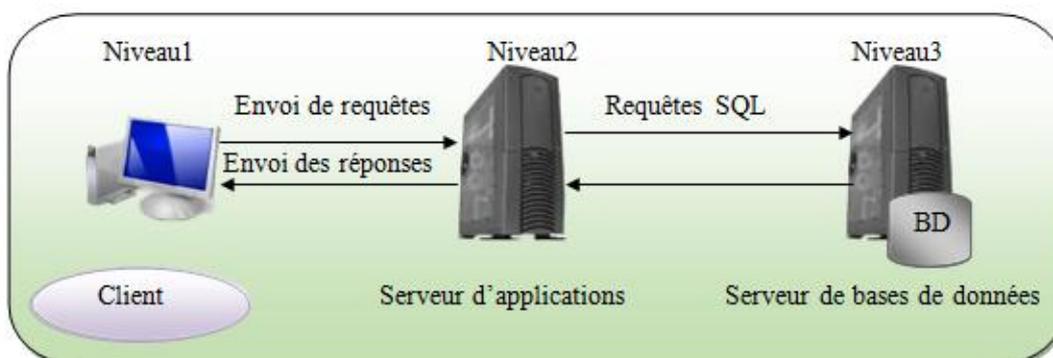
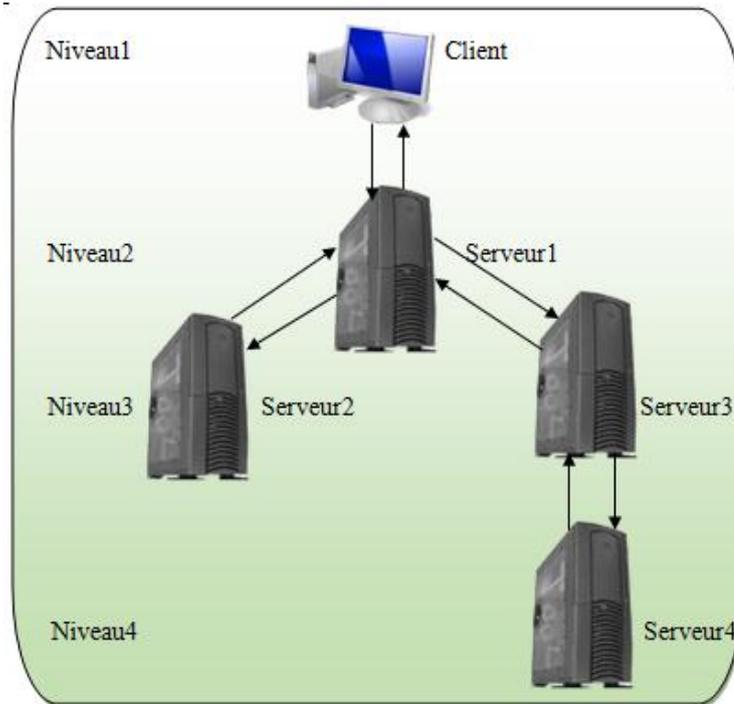


Figure 10:l'architecture à 3 niveaux.

### 3.3. L'architecture multi niveaux

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux (voir figure 11). [3]



**Figure 11: architecture multi niveaux.**

## Deuxième partie : La méthodologie 2TUP

La complexité croissante des systèmes informatiques a conduit les concepteurs à s'intéresser aux méthodes. Chaque méthode se définit par une notation et un processus spécifique. Le groupe UML tente à unifier non pas les processus, mais plus exactement les meilleures pratiques de développement objet. Ces processus se distingueront par le générique Unified Process.

Dans ce chapitre, nous commençons d'abord par une présentation du processus de développement. Les étapes du processus de développement du système seront guidées par le processus unifié 2TUP (Two Tracks Unified Process). [4]

### 1. Processus de développement

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles. [4]

#### 1.1. Le Processus Unifié

Le Processus Unifié (UP, pour Unified Process) est un processus de développement logiciel, itératif, incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques.

- **Itératif et incrémental.** Le projet est découpé en itérations de courte durée qui aident à mieux suivre l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
- **Centré sur l'architecture.** Le système doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture doit être modélisée en UML.
- **Piloté par les risques.** Les risques majeurs du projet doivent être identifiés au plus tôt.
- **Conduit par les cas d'utilisation.** Le projet est mené en tenant compte des besoins et des exigences des utilisateurs.

## 1.2. Le processus 2TUP

Le 2TUP (pour 2 Track Unified Process). C'est un processus UP qui répond aux caractéristiques que nous venons de citer. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise.

Ce processus suit deux chemins. Il s'agit des chemins « fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système informatique (voir figure 12). [4]



Figure 12: Le système d'information soumis à deux natures de contraintes.

À l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette fusion conduit à l'obtention d'un processus de développement en forme de Y, comme illustré par la (figure 13).

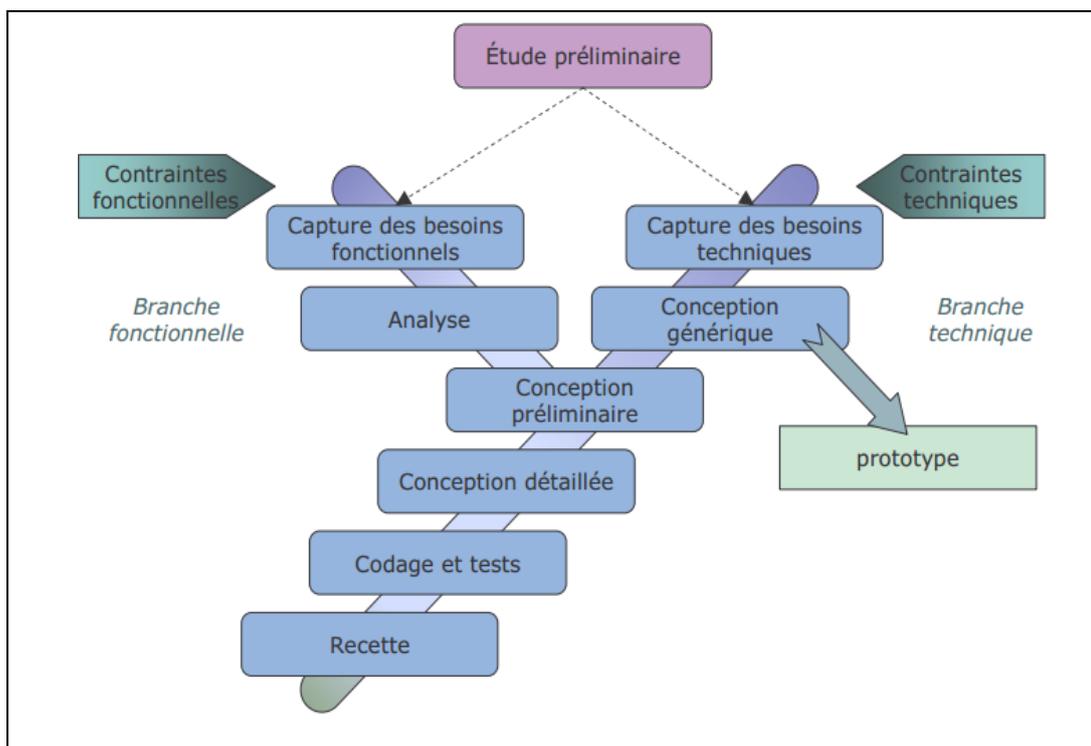


Figure 13: Le processus de développement en Y.

Nous allons expliquer en bref chaque étape du processus de développement 2TUP.

### 1.2.1. La branche gauche (fonctionnelle)

Capitalise la connaissance du métier de l'entreprise. Elle constitue généralement un investissement pour le moyen et le long terme. Les fonctions du système d'information sont en effet indépendantes des technologies utilisées. [4]

Cette branche comporte les étapes suivantes :

- **La capture des besoins fonctionnels.** Qui produit un modèle des besoins focalisé. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs sur le métier des utilisateurs.
- **L'analyse.** Qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

### 1.2.2. La branche droite (architecture technique)

Capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser. [4]

Cette branche comporte les étapes suivantes :

- **La capture des besoins techniques.** Qui recense toutes les contraintes sur les choix de dimensionnant et la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique).
- **La conception générique.** Qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnel. Elle a pour objectif de d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.

### 1.2.3. La branche du milieu (conception)

À l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats de 2 branches. Cette fusion conduit à l'obtention d'un processus en forme de **Y**. [4]

Cette branche comporte les étapes suivantes :

- **La conception préliminaire.** Qui représente une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- **La conception détaillée.** Qui étudie ensuite comment réaliser chaque composant.
- **L'étape de codage.** Qui produit ses composants et teste au fur et à mesure les unités de code réalisées.
- **L'étape de recette.** Qui consiste enfin à valider les fonctionnalités du système développé.

## Conclusion

Pour réaliser notre projet, et d'après les différentes catégories de serveurs et les différentes types d'architecture de client-serveur vus dans la première partie de ce chapitre, nous avons choisi d'utiliser l'architecture à deux niveaux (appelée architecture 2-tiers) et le serveur de base de données existant et opérationnel au niveau de la Wilaya, où les demandes seront émises sous forme de requêtes SQL et le traitement de ces requêtes s'effectue automatiquement selon des contraintes suivies lors du développement du logiciel installé au niveau de la commune, d'une part, et au niveau de la Wilaya d'une autre part.

Le processus de développement 2TUP permet la conception d'un système, en tenant compte les étapes de développement et les points de vue de modélisation préconisée.

Dans le troisième chapitre du mémoire, nous allons mettre en œuvre cette méthodologie pour notre étude de cas.

# Chapitre

*ANALYSE*  
*ET*  
*CONCEPTION*

3

## Introduction

Dans ce chapitre on va suivre les étapes du processus de développement 2TUP pour le développement de notre logiciel. Premièrement, on va faire l'étude préliminaire pour identifier les acteurs qui interagiront avec le système, on va développer un premier modèle UML de niveau contexte, pour pouvoir préciser les frontières du système. Deuxièmement, on va faire une capture des besoins fonctionnels, dans cette phase on va élaborer le diagramme de cas d'utilisation, le diagramme d'activité, le diagramme de séquence et le diagramme de domaine . Troisièmement, on va faire une capture des besoins techniques, dans cette phase on va faire la capture des spécifications logicielles et capture des spécifications liées à la configuration matérielle. Quatrièmement la phase d'analyse, constituée de trois activités : le découpage en catégorie, le développement du modèle statique et le développement du modèle dynamique. Cinquièmement la phase de la conception préliminaire, et enfin la phase de conception détaillé qui consiste à concevoir précisément le code qui va être produit.

## Première Partie : Etude Préliminaire

L'étude préliminaire ou (pré-étude) est la toute première étape de notre processus de développement. Elle survient à la suite d'une décision de démarrage de projet, et consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou des diagrammes très simples. Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture techniques.[4]

### 1. Elaboration du cahier des charges :

Le cahier des charges est un document essentiel à la réalisation d'un projet. Il décrit précisément les besoins auxquels les intervenants doivent répondre : objectifs, cibles, concurrence, spécifications techniques, etc.

#### 1.1. Présentation du projet

Le but de ce projet de fin d'étude, est de faire la conception, et la réalisation d'une application client-serveur pour la gestion du patrimoine communal de la wilaya de Mila.

#### 1.2. Les choix techniques

Les choix techniques que nous avons faits pour la réalisation de notre application sont :

- le processus développement 2TUP qui utilise le langage standard de modélisation UML.
- Le langage JAVA pour la programmation de la logique applicative de ce projet dans une plateforme NetBeans.
- Oracle pour le stockage et la gestion des données.

#### 1.3. Recueil des besoins fonctionnels

Nous avons effectué plusieurs recherches pour identifier au mieux les besoins de l'application, et ceci afin de répondre aux attentes des utilisateurs. Nous sommes allés chercher les informations au sein de la direction de l'administration locale de la wilaya(DAL) et le service d'administration et moyens généraux de la commune. Cette phase correspond à une recherche sur le terrain pour bien définir le cadre de notre système. Nous nous sommes aussi procuré quelques documents, qui expliquent le mode de fonctionnement de la gestion de

patrimoine communale entre la commune et la wilaya, ce qui nous a permis d'établir le cahier des charges préliminaire suivant :

- **Au niveau de la commune**

- **Gestion patrimoine.** Consiste en l'ajout, la modification, la suppression ou consultation d'un patrimoine communal exemple : marchés communaux, véhicules.
- **Gestion propriétaire.** Consiste en l'ajout, la modification, la suppression ou consultation d'un propriétaire.
- **Location.** Consiste en la création, la modification, la suppression, la consultation ou le renouvellement d'un contrat administratif de location correspondant à un patrimoine communal.
- **Vente.** Consiste en la création, la modification suppression ou la consultation d'un contrat administratif de vente correspondant à un patrimoine communal.
- **Recherche Multicritères de la commune.** Elle s'effectue selon plusieurs critères (type du patrimoine, type du contrat etc.) pour obtenir les différentes informations des patrimoines, des contrats et des propriétaires.
- **Statistique commune.** Elle s'effectue selon plusieurs critères (type du patrimoine, type du contrat etc.) pour obtenir et calculer les différentes statistiques des patrimoines, des contrats, des propriétaires, etc.

- **Au niveau de la wilaya**

- **Recherche Multicritères de la wilaya.** Elle s'effectue selon plusieurs critères (daïra, commune, type de patrimoine, type de contrat etc.) pour obtenir les différentes informations des patrimoines et des contrats des communes et des propriétaires.
- **Statistique wilaya.** Elle s'effectue selon plusieurs critères (daïra, commune, type de patrimoine, type de contrat etc.) pour obtenir et calculer les différentes statistiques des patrimoines et des contrats des communes.
- **Validation des contrats administratifs.** Consiste en la validation, la vérification et le contrôle des contrats (de location et de vente) créent par les communes.
- **Gestion des utilisateurs.** Consiste en la création, modification, consultation ou la suppression d'un utilisateur (commune ou wilaya).

#### 1.4. Recueil des besoins opérationnels

Les besoins opérationnels représentent les besoins non fonctionnels, qui caractérisent le système comme la performance ainsi que la sécurité et l'ergonomie du système.

- **La Sécurité.** Lors de démarrage, chaque utilisateur doit être reconnu du système par un login, et un mot de passe. Un administrateur système est chargé de définir les profils des utilisateurs pour garantir l'intégrité et la confidentialité de ses données.
- **L'ergonomie des interfaces.** L'interface de cette application doit être simple et claire, la manipulation de l'interface ne doit pas nécessiter des connaissances (facile à manipuler, compréhensible), l'application doit être compatible avec n'importe quel système d'exploitation, les interfaces doivent être bien organisées du point de vue graphique, le choix des couleurs, et des styles.

### 1.5. Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateurs, dispositif matériel ou autre système) qui interagissent directement avec le système étudiés. [4]

Dans notre système il existe trois (03) acteurs :

- **Administrateur.** Utilisateur au niveau de la wilaya. Il effectue la gestion des utilisateurs de la wilaya, et des communes.
- **Utilisateur de la wilaya.** Il effectue la recherche selon les différents critères (patrimoine, contrat administratif), et fait les statistiques et la validation des contrats administratifs.
- **Utilisateur de la commune.** Il effectue la gestion du patrimoine, la gestion du propriétaire, location et vente de patrimoine après la création des contrats administratifs pour louer et vendre, et fait la recherche selon différents critères (type patrimoine, contrat administratif), ainsi que les statistiques.

### 1.6. Identification des messages

On va détailler les différents messages échangés entre le système et le monde extérieur (les acteurs).

#### 1.6.1. Entre le système et l'administrateur

- Le système émet les messages suivants :
  - La confirmation de mise à jour des utilisateurs.
  - Liste des utilisateurs.

- Le système reçoit les messages suivants :
  - Création (), modification (), suppression (), consultation des utilisateurs.

### 1.6.2. Entre le système et l'utilisateur de la commune

- Le système émet les messages suivants :
  - La confirmation de mise à jour du patrimoine.
  - La confirmation de mise à jour du propriétaire.
  - La confirmation de mise à jour des contrats administratifs pour vente et location.
  - Liste des patrimoines.
  - Liste des propriétaires.
  - Liste des contrats administratifs de vente et location.
  - Résultat de la recherche.
  - Résultat des statistiques.
  - Notification de validation.
- Le système reçoit les messages suivants :
  - Ajout (), modification (), suppression (), consultation () des patrimoines.
  - Ajout (), modification (), suppression (), consultation () des propriétaires.
  - Création (), modification (), suppression (), consultation () des contrats administratifs de vente et location.
  - Statistique\_commune ().
  - Recherche\_Multicritère\_commune ().

### 1.6.3. Entre le système et l'utilisateur de la wilaya

- Le système émet les messages suivant :
  - La confirmation de la validation des contrats administratifs.
  - Liste des patrimoines.
  - Liste des contrats administratifs de vente et location.
  - Résultat de la recherche.
  - Résultat des statistiques.
  - Notification de validation.
- Le système reçoit les messages suivant :
  - Validation () des contrats administratifs.
  - Statistiques\_wilaya ().

- Recherche\_Multicritère\_wilaya ().

### 1.7. Modélisation de contexte

Après les étapes précédentes, et à partir des informations obtenues, nous allons modéliser le contexte de notre application. Ceci va nous permettre dans un premier temps, de définir le rôle de chaque acteur dans le système. Voir le tableau (5) suivant :

Les utilisateurs finaux	Description des besoins fonctionnels
<b>Administrateur</b>	L'application doit permettre à l'administrateur : <ul style="list-style-type: none"> <li>- Authentification,</li> <li>- Gestion des utilisateurs.</li> </ul>
<b>Utilisateur Wilaya</b>	L'application doit permettre à utilisateur de la wilaya : <ul style="list-style-type: none"> <li>- Authentification,</li> <li>- Recherche Multicritères du patrimoine, les contrats et les propriétaires de chaque commune de la wilaya,</li> <li>- Statistiques du patrimoine et les contrats par communes,</li> <li>- Validation des contrats administratifs.</li> </ul>
<b>Utilisateur Commune</b>	L'application doit permettre à utilisateur de commune : <ul style="list-style-type: none"> <li>- Authentification.</li> <li>- Recherche Multicritères du patrimoine, les contrats administratifs et les propriétaires de la commune.</li> <li>- Gestion des patrimoines.</li> <li>- Gestion des propriétaires.</li> <li>- Location d'un patrimoine.</li> <li>- Vente d'un patrimoine.</li> <li>- Statistiques du patrimoine et des contrats de la commune.</li> </ul>

**Tableau 5 : modélisation de contexte.**

### 1.8. Le diagramme de contexte dynamique de système

Les messages de transaction entre les acteurs et le système identifiés précédemment, peuvent être représentés de façon synthétique, grâce au diagramme de contexte dynamique illustré par la figure 14.

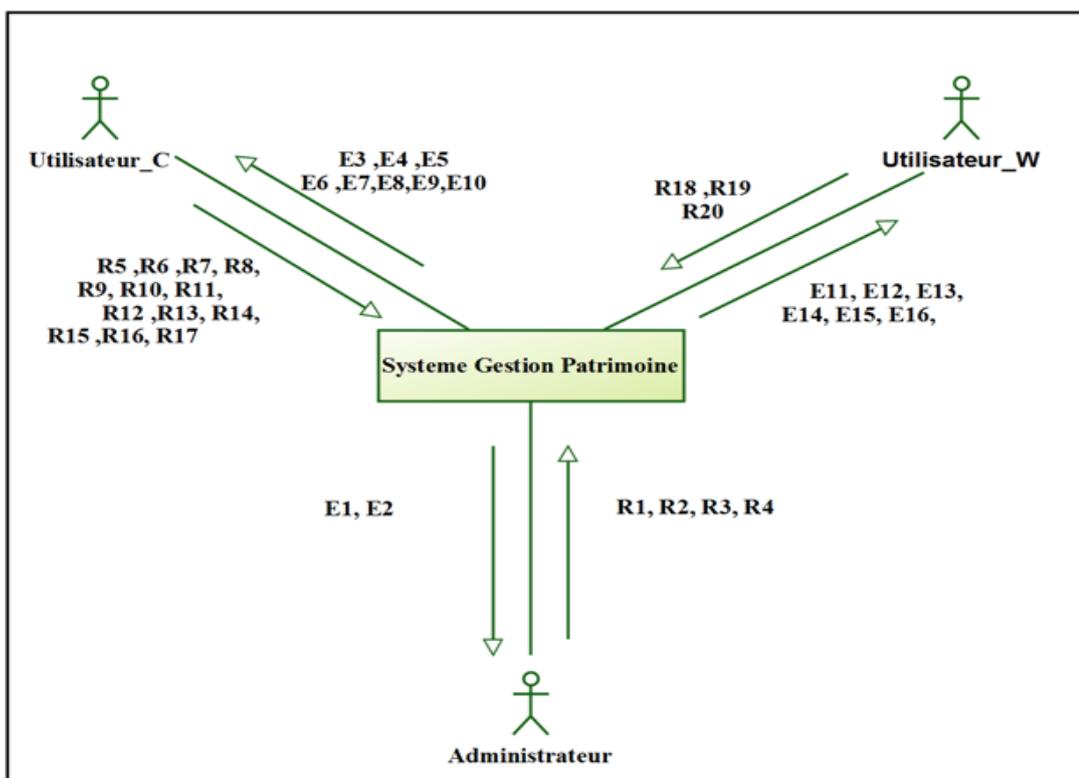


Figure 14: Diagramme de contexte dynamique

### 1.9. La signification des messages

Les Acteurs	Système	
	Les messages émettent	Les messages reçus
<b>Administrateur</b>	E1 : La confirmation de mise à jour des utilisateurs. E2 : Liste des utilisateurs	R1:Création_utilisateur (). R2 :Modification_utilisateur (). R3:Suppression_utilisateur (). R4 :Consultation_utilisateur ().
<b>Utilisateur_C</b>	E3 : La confirmation de mise à jour de patrimoine. E4 : La confirmation de mise à jour des contrats administratifs (vente ou location). E5 : Liste des contrats de vente ou location. E6 : Résultat de recherche. E7 : Résultat des statistiques. E8 : Notification de validation.	R5: Ajout_patrimoine (). R6:Modification_patrimoine(). R7: Suppression_patrimoine (). R8:Consultation_patrimoine (). R9: Création_contrat (). R10 : Modification_contrat (). R11 : Suppression_contrat (). R12 : Statistique_commune (). R13:Recherche_Multicritères (). R14: Ajout_propriétaire ().

	E9:Confirmation de mise à jour des propriétaires. E10 : Liste des propriétaires.	R15:Modification_propriétaire(). R16:Suppression_propriétaire (). R17:Consultation_propriétaire().
<b>Utilisateur_W</b>	E11:Confirmation de la validation des contrats administratifs. E12: Liste des patrimoines. E13:Liste des contrats administratifs de vente ou location. E14 : Résultat de recherche. E15 : Résultat des statistiques. E16 : Notification de validation.	R18: Validation_contrats (). R19: Statistique_wilaya (). R20:Recherche_Multicritères ().

**Tableau 6: Signification des messages.**

## Deuxième Partie : Capture Des Besoins Fonctionnels

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y de la méthode 2TUP. Elle formalise et détaille ce qui a été recueilli lors de l'étude préliminaire. Cette phase représente un point de vue « fonctionnel » de l'architecture système. Par le biais des cas d'utilisation, nous nous baserons sur les données recueillies et nous resterons en contact continu avec les acteurs du système en vue de définir les limites de celui-ci, et ainsi éviter de trop s'éloigner des besoins réels des utilisateurs finaux.

### 1. Identification des cas d'utilisation

- **Qu'est qu'un cas d'utilisation ?**

Un cas d'utilisation représente un ensemble de séquence d'actions réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il modélise un service rendu par le système, et permet de décrire ce que le futur système devra faire sans spécifier comment il le fera.

Le tableau (7) suivant montre les liens entre les cas d'utilisation identifiés, les acteurs principaux et secondaires et les messages provenant du contexte.

Cas d'utilisation	Les Acteurs	Message émis/reçu
<b>Gestion patrimoine</b>  - Ajout - Modification - Suppression - Consultation		<b>Emet</b> :Ajout_patrimoine, Modification_patrimoine, Suppression_patrimoine.  <b>Reçoit</b> : Notification de mise à jour du patrimoine.  <b>Emet</b> :Consultation_patrimoine.  <b>Reçoit</b> :Liste des patrimoines.
<b>Gestion propriétaire</b>  - Ajout		<b>Emet</b> :Ajout_propriétaire, Modification_propriétaire,      Suppression_

<ul style="list-style-type: none"> <li>- Modification</li> <li>- Suppression</li> <li>- Consultation</li> </ul>	<p><b>Utilisateur_C</b></p>	<p>propriétaire.</p> <p><b>Reçoit :</b> Notification de mise à jour dupropriétaire.</p> <p><b>Emet :</b>Consultation_ propriétaire.</p> <p><b>Reçoit :</b>Liste des propriétaires.</p>
<p><b>Location</b></p> <ul style="list-style-type: none"> <li>- Création</li> <li>- Modification</li> <li>- Suppression</li> <li>- Consultation</li> </ul>		<p><b>Emet :</b>Création_contrat_location, Modification_contrat_location, Suppression_contrat_location.</p> <p><b>Reçoit :</b> Notification de mise à jour descontrats administratifs de location.</p> <p><b>Emet :</b>Consultation_contrat_location.</p> <p><b>Reçoit :</b>Liste des contrats de location.</p>
<p><b>Vente</b></p> <ul style="list-style-type: none"> <li>- Création</li> <li>- Modification</li> <li>- Suppression</li> <li>- Consultation</li> </ul>		<p><b>Emet :</b>Création_contrat_vente, Modification_contrat_vente, Suppression_contrat_vente.</p> <p><b>Reçoit :</b> Notification de mise à jour descontrats administratifs de vente.</p> <p><b>Emet :</b>Consultation_contrat_vente.</p> <p><b>Reçoit :</b>Liste des contrats de vente.</p>
<p><b>Recherche</b></p> <ul style="list-style-type: none"> <li>- Patrimoine</li> <li>- Contrat</li> <li>- Propriétaire</li> </ul>		<p><b>Emet :</b>Recherche_Multicritères_commune.</p> <p><b>Reçoit :</b> différent états de sortie selon le critère de recherche.</p>
<p><b>Statistique commune</b></p> <ul style="list-style-type: none"> <li>- Patrimoine</li> </ul>		<p><b>Emet :</b> statistique_commune.</p> <p><b>Reçoit :</b> différent états de sortie selon le</p>

- Contrat		critère de demande.
<b>Recherche</b> - Patrimoine - Contrat - Propriétaire	Utilisateur_W	<b>Emet</b> :Recherche_Multicritères_wilaya.  <b>Reçoit</b> : différent états de sortie selon le critère de recherche.
<b>Statistique wilaya</b> - Patrimoine - Contrat		<b>Emet</b> :Statistique_wilaya.  <b>Reçoit</b> : différent états de sortie selon le critère de demande.
<b>Validation contrat</b>		<b>Emet</b> :Validation_contrat.  <b>Reçoit</b> :Notification de validation.
<b>Gestion utilisateurs</b>	Administrateur	<b>Emet</b> :Création_utilisateur, Modification_utilisateur,      Suppression_utilisateur.  <b>Reçoit</b> : Notification de mise à jour des utilisateurs.  <b>Emet</b> :Consultation_utilisateur.  <b>Reçoit</b> :Liste des utilisateurs.

**Tableau 7: Liste des acteurs et des messages par cas d'utilisation.**

## 2. Le diagramme de cas d'utilisation

Maintenant que nous avons identifié les cas d'utilisation et leurs acteurs, nous allons les représenter graphiquement sur un diagramme de cas d'utilisation qui représente le fonctionnement du système vis-à-vis de l'utilisateur :

- Notre diagramme de cas d'utilisation :

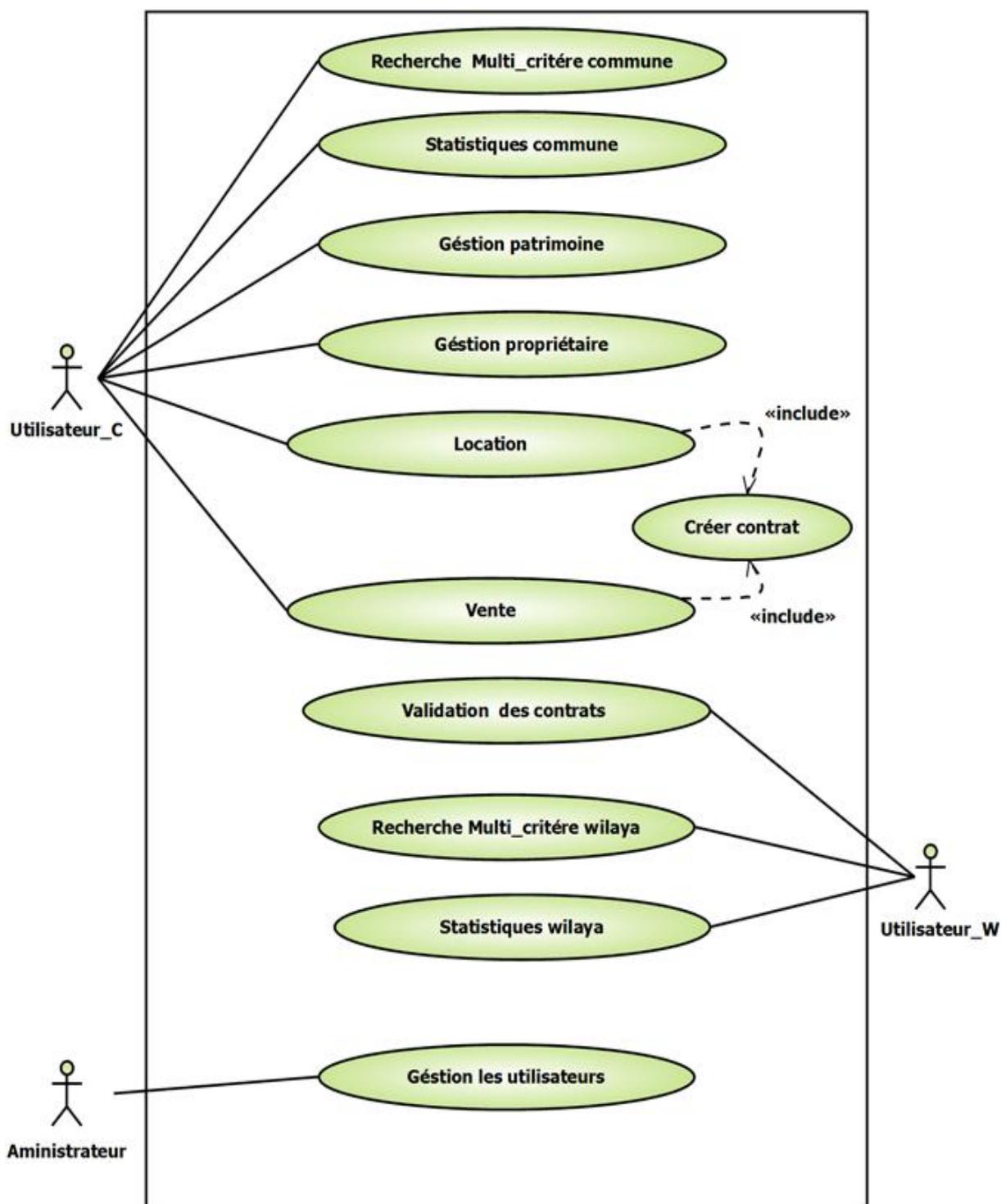


Figure 15: Diagramme de cas d'utilisation.

### 3. Description des cas d'utilisation

Nous allons maintenant décrire chaque cas en lui associant un nom, un objectif, les acteurs, les pré-conditions, les post-conditions et des scénarios (nominal, alternatif, exceptionnel) sont illustrés par les diagrammes d'activité et de séquence.

#### 3.1. Au niveau de la commune

##### 3.1.1. Gestion des patrimoines

Ce cas commence lorsque l'utilisateur accède au système et choisit «Gestion du patrimoine ».

Cas d'utilisation	Gestion_patrimoine
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Consiste à ajouter, modifier, supprimer ou consulter un patrimoine.
<b>Pré condition</b>	L'utilisateur doit être Authentifié.
<b>Post condition</b>	/
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la gestion du patrimoine.</li> <li>2. Le système affiche menu de la gestion du patrimoine.</li> <li>3. L'utilisateur choisit l'une des actions suivantes : <ul style="list-style-type: none"> <li>- Ajout d'un patrimoine.</li> <li>- Modification d'un patrimoine.</li> <li>- Suppression d'un patrimoine.</li> <li>- Consultation d'un patrimoine.</li> </ul> </li> </ol>

**Tableau 8: Description textuelle du cas d'utilisation « gestion patrimoine».**

##### 3.1.1.1. Ajout patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Ajout\_patrimoine » pour ajoute un patrimoine.

Cas d'utilisation	Ajout_patrimoine
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Permettre à l'utilisateur d'ajout un nouveau patrimoine
<b>Pré condition</b>	L'utilisateur doit être Authentifié.
<b>Post condition</b>	Le nouvel patrimoine enregistré par le système.

<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Utilisateur demande l'ajout d'un patrimoine.</li> <li>2. Le système affiche le formulaire d'ajout.</li> <li>3. L'utilisateur remplit le formulaire par les informations de patrimoine qui veut ajouter.</li> <li>4. Le système lit et enregistre les informations de patrimoine.</li> <li>5. Le système affiche une notification «ajout avec succès».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le patrimoine existe déjà ou les informations sont mal remplies, le système affiche une notification («le patrimoine existe déjà», ou «le formulaire est mal rempli»).</li> <li>2. Reprise de l'enchaînement du scénario nominal au point 2.</li> </ol>

**Tableau 9: Description textuelle du cas d'utilisation «Ajout\_patrimoine».**

### 3.1.1.2. Modification patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Modification\_patrimoine» pour modifier les informations relatives d'un patrimoine.

Cas d'utilisation	Modification_patrimoine
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Permettre à l'utilisateur de modifier les informations d'un patrimoine
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un patrimoine dans la base.</li> </ul>
<b>Post condition</b>	Les modifications enregistrées par le système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la modification d'un patrimoine.</li> <li>2. Le système affiche la liste des patrimoines.</li> <li>3. L'utilisateur consulte la liste des patrimoines, et choisit un patrimoine à modifier.</li> <li>4. Le système affiche le formulaire de modification avec les informations sur le patrimoine.</li> </ol>

	<ol style="list-style-type: none"> <li>5. L'utilisateur modifie les informations souhaitées et confirme.</li> <li>6. Le système lit et enregistre les nouvelles informations saisies.</li> <li>7. Le système affiche une notification «modification avec succès».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où les informations sont mal remplis, le système affiche une notification « le formulaire est mal rempli ».</li> <li>2. Reprise de l'enchaînement du scenario nominal au point 4.</li> </ol>

**Tableau 10: Description textuelle du cas d'utilisation «Modification\_patrimoine».**

### 3.1.1.3. Suppression patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Suppression\_patrimoine » pour supprimer un patrimoine.

Cas d'utilisation	Suppression_patrimoine
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Permettre à l'utilisateur de supprimer un patrimoine
<b>Pré condition</b>	L'utilisateur doit être Authentifié. Il existe au moins un patrimoine dans la base.
<b>Post condition</b>	La suppression effectuée par le système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la suppression d'un patrimoine.</li> <li>2. Le système affiche la liste des patrimoines.</li> <li>3. L'utilisateur consulte la liste des patrimoines, et choisit le patrimoine à supprimer.</li> <li>4. Le système affiche un message de confirmation.</li> <li>5. L'utilisateur confirme la suppression.</li> <li>6. Le système supprime le patrimoine sélectionné.</li> <li>7. Le système affiche une notification « suppression avec succès ».</li> </ol>

<b>Scénario Alternatif</b>	1. Le cas où l'utilisateur annule la suppression, le système ignore la suppression.
----------------------------	---

**Tableau 11: Description textuelle du cas d'utilisation «Suppression\_patrimoine».**

#### 3.1.1.4. Consultation patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Consultation\_patrimoine» pour consulter les informations d'un patrimoine.

Cas d'utilisation	Consultation_patrimoine
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Permettre à l'utilisateur de consulter les informations d'un patrimoine.
<b>Pré condition</b>	L'utilisateur doit être Authentifié. Il existe au moins un patrimoine dans la base.
<b>Post condition</b>	Les informations du patrimoine sont affichées.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la consultation.</li> <li>2. Le système affiche la liste des patrimoines.</li> <li>3. L'utilisateur choisit un patrimoine pour consulter les informations.</li> <li>4. Le système affiche les informations de ce patrimoine.</li> </ol>

**Tableau 12: Description textuelle du cas d'utilisation «Consultation\_patrimoine».**

• Diagramme d'activité : Gestion patrimoine

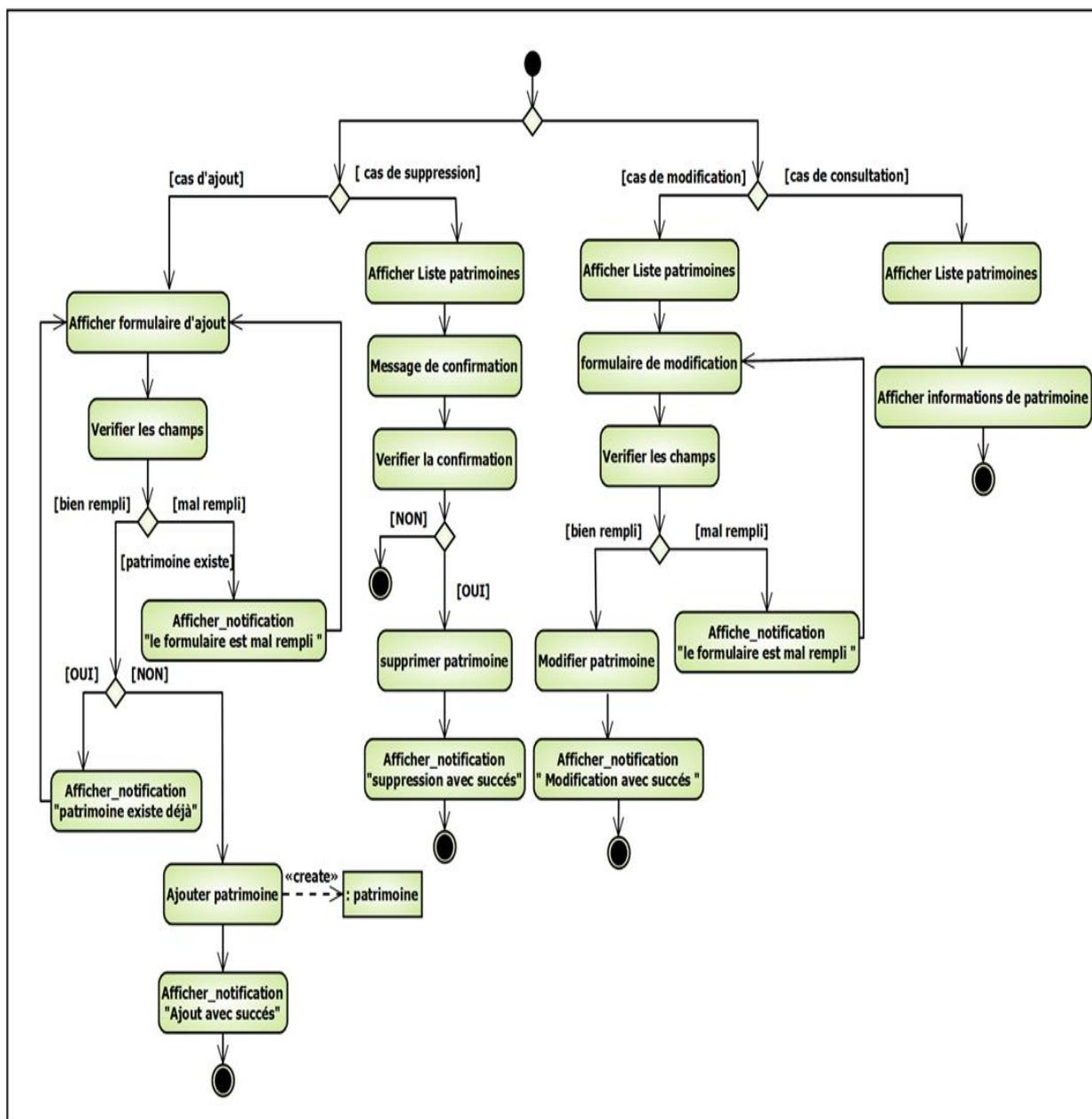


Figure 16: Diagramme d'activité «Gestion\_patrimoine».

- Diagramme de séquence : Gestion patrimoine

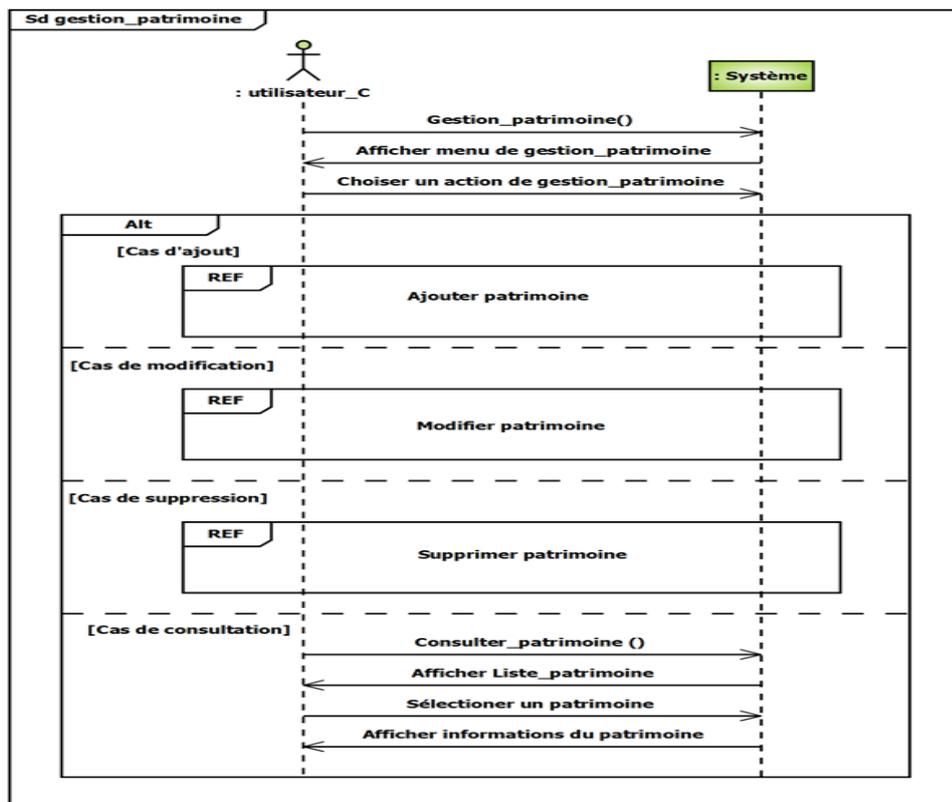


Figure 17: Diagramme de séquence «Gestion\_patrimoine».

- Diagramme de séquence : Ajouter patrimoine

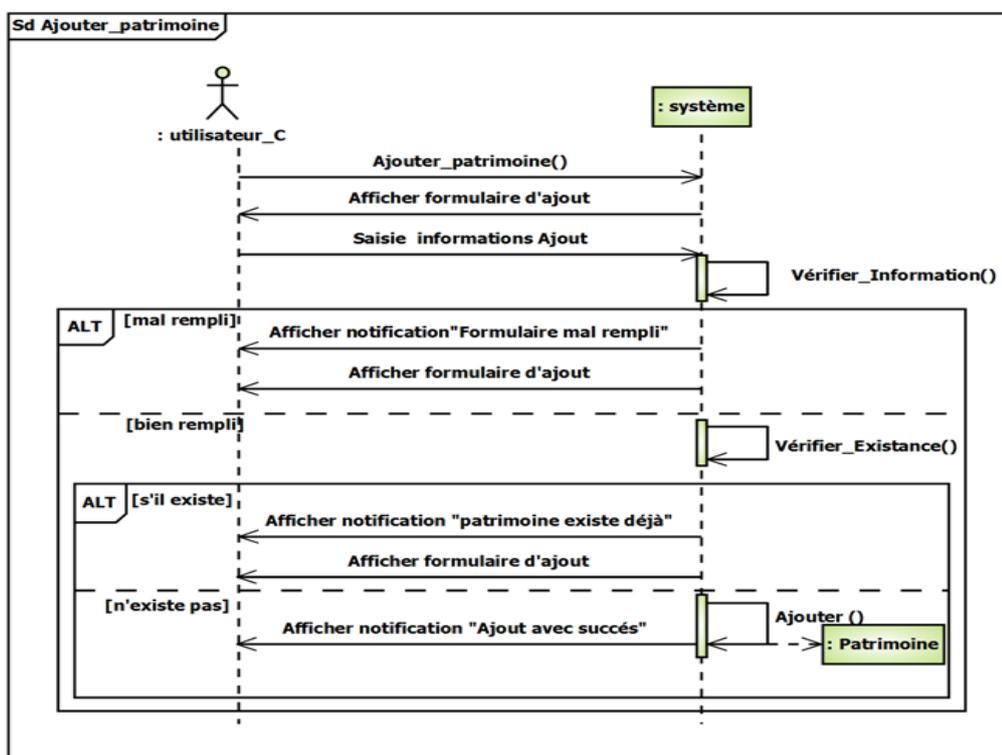


Figure 18: Diagramme de séquence «Ajouter\_patrimoine».

• Diagramme de séquence : Modifier patrimoine

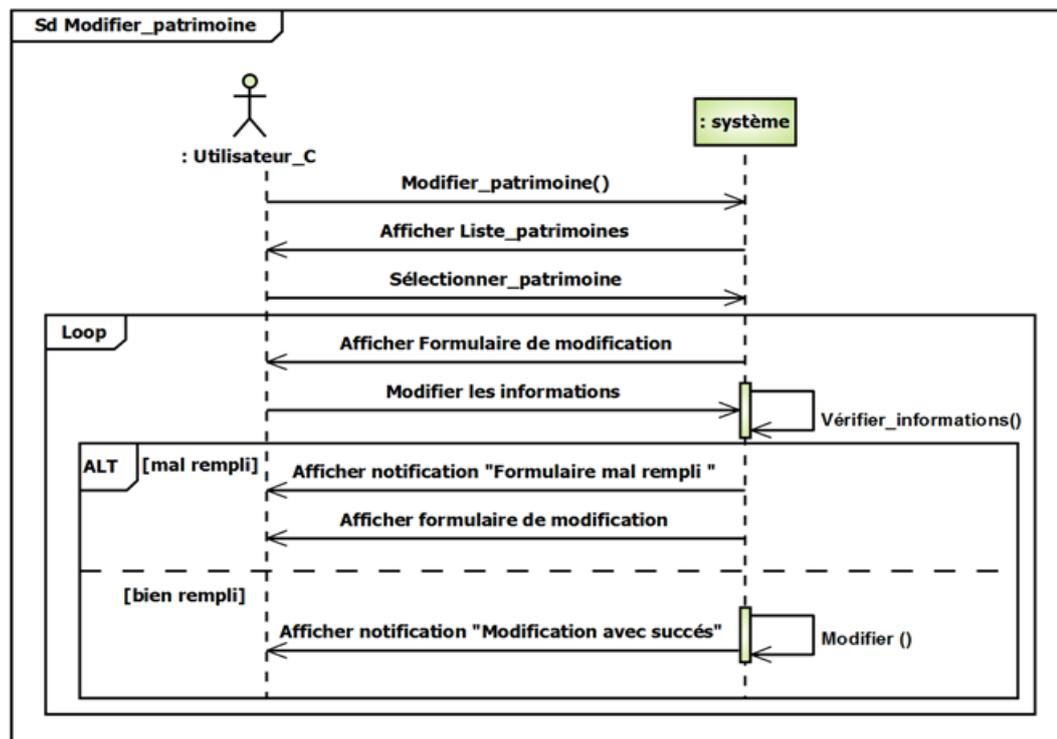


Figure 19: Diagramme de séquence «Modifier\_patrimoine».

• Diagramme de séquence : Supprimer patrimoine

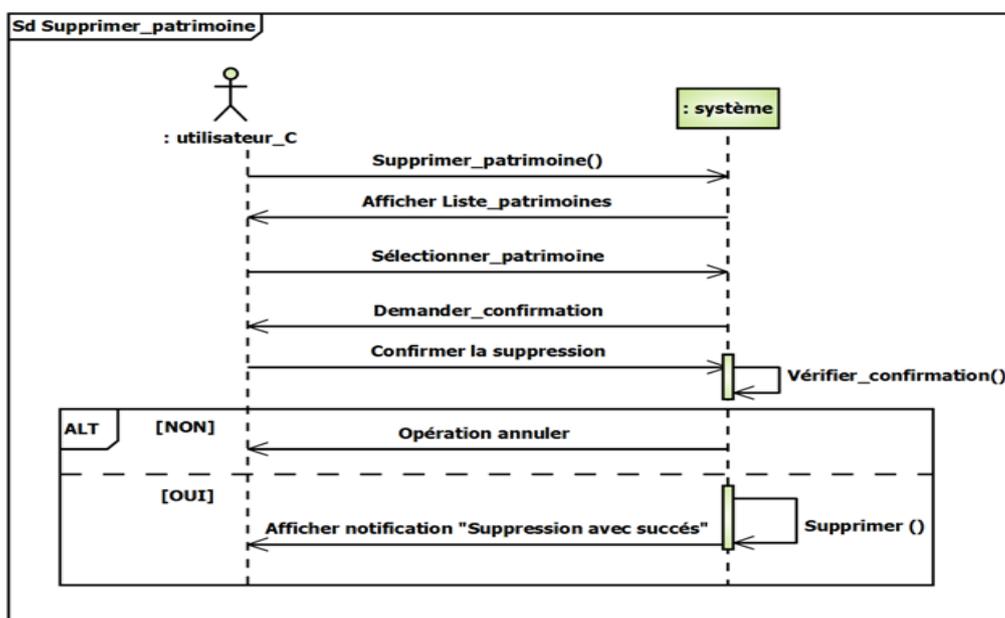


Figure 20: Diagramme de séquence «Supprimer\_patrimoine».

### 3.1.2. Gestion des propriétaires

Ce cas commence lorsque l'utilisateur de commune accède au système et choisit «Gestion\_propriétaires ».

Cas d'utilisation		Gestion_propriétaires
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Consiste à créer, modifier, supprimer ou consulter un propriétaire.	
<b>Pré condition</b>	L'utilisateur doit être authentifié.	
<b>Post condition</b>	/	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la gestion des propriétaires.</li> <li>2. Le système affiche menu de la gestion des propriétaires.</li> <li>3. L'utilisateur choisi de l'action suivante : <ul style="list-style-type: none"> <li>- Ajout d'un propriétaire.</li> <li>- Modification d'un propriétaire.</li> <li>- Suppression d'un propriétaire.</li> <li>- Consultation d'un propriétaire.</li> </ul> </li> </ol>	

Tableau 13: Description textuelle du cas d'utilisation «gestion\_propriétaires ».

#### 3.1.2.1. Ajout d'un propriétaire

Ce cas commence lorsque l'utilisateur accède au système et choisit «Ajout\_propriétaire »

Cas d'utilisation		Ajout_propriétaire
<b>Acteur</b>	L'utilisateur_C	
<b>But</b>	Permettre à l'utilisateur d'ajouter un propriétaire.	
<b>Pré condition</b>	L'utilisateur doit être authentifié.	
<b>Post condition</b>	L'ajout d'un propriétaire est effectué.	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande l'ajout d'un propriétaire.</li> <li>2. Le système affiche le formulaire d'ajout.</li> <li>3. L'utilisateur remplit le formulaire.</li> <li>4. Le système lit et enregistre les informations du propriétaire.</li> </ol>	

	5. Le système affiche une notification «Ajout avec succès ».
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le formulaire est mal rempli ou ce propriétaire existe, le système affiche une notification (« propriétaire existe déjà », ou « formulaire est mal rempli »).</li> <li>2. Reprise de l'enchaînement du scénario nominal au point 2.</li> </ol>

**Tableau 14: Description textuelle du cas d'utilisation «Ajout\_propriétaire».**

### 3.1.2.2. Modification d'un propriétaire

Ce cas commence lorsque l'utilisateur accède au système et choisit «Modification\_propriétaire»

Cas d'utilisation Modification_propriétaire	
<b>Acteur</b>	L'utilisateur_C
<b>But</b>	Modification des informations d'un propriétaire.
<b>Pré condition</b>	L'utilisateur doit être authentifié.
<b>Post condition</b>	La modification est effectuée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la modification d'un propriétaire.</li> <li>2. Le système affiche la liste des propriétaires.</li> <li>3. L'utilisateur consulte la liste des propriétaires, et choisit le propriétaire à modifier.</li> <li>4. Le système affiche le formulaire de modification avec les informations de se propriétaire.</li> <li>5. L'utilisateur modifie les informations souhaitées et confirme.</li> <li>6. Le système lit et enregistre les nouvelles informations.</li> <li>7. Le système affiche une notification «modification avec succès ».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le formulaire est mal rempli, le système affiche une notification « formulaire est mal rempli ».</li> <li>2. Reprise de l'enchaînement du scénario nominal au point 4.</li> </ol>

**Tableau 15: Description textuelle du cas d'utilisation«Modification\_propriétaire ».**

### 3.1.2.3. Suppression d'un propriétaire

Ce cas commence lorsque l'administrateur accède au système et choisit «Suppression\_propriétaire»

Cas d'utilisation		Suppression_propriétaire
<b>Acteur</b>	L'utilisateur_C	
<b>But</b>	Suppression d'un propriétaire.	
<b>Pré condition</b>	L'utilisateur doit être authentifié.	
<b>Post condition</b>	La suppression est effectuée.	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la suppression d'un propriétaire.</li> <li>2. Le système affiche la liste des propriétaires.</li> <li>3. L'utilisateur consulte la liste des propriétaires, et choisit le propriétaire à supprimer.</li> <li>4. Le système affiche le message de confirmation.</li> <li>5. L'utilisateur confirme la suppression.</li> <li>6. Le système supprimer le propriétaire sélectionner.</li> <li>7. Le système affiche une notification «suppression avec succès ».</li> </ol>	
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où l'administrateur annuler la suppression, le système ignorer la suppression.</li> </ol>	

Tableau 16: Description textuelle du cas d'utilisation «Suppression\_propriétaire ».

### 3.1.2.4. Consultation propriétaire

Ce cas commence lorsque l'utilisateur accède au système et choisit «Consultation\_propriétaire» pour consulter les informations d'un propriétaire.

Cas d'utilisation		Consultation_propriétaire
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Permettre à l'utilisateur de consulter les informations d'un propriétaire.	
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un propriétaire dans la base.</li> </ul>	

<b>Post condition</b>	Les informations du propriétaire sont affichées.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la consultation.</li> <li>2. Le système affiche la liste des propriétaires.</li> <li>3. L'utilisateur choisit un propriétaire pour consulter les informations.</li> <li>4. Le système affiche les informations de ce propriétaire.</li> </ol>

Tableau 17: Description textuelle du cas d'utilisation «Consultation\_ propriétaire».

• Diagramme d'activité : Gestion propriétaires

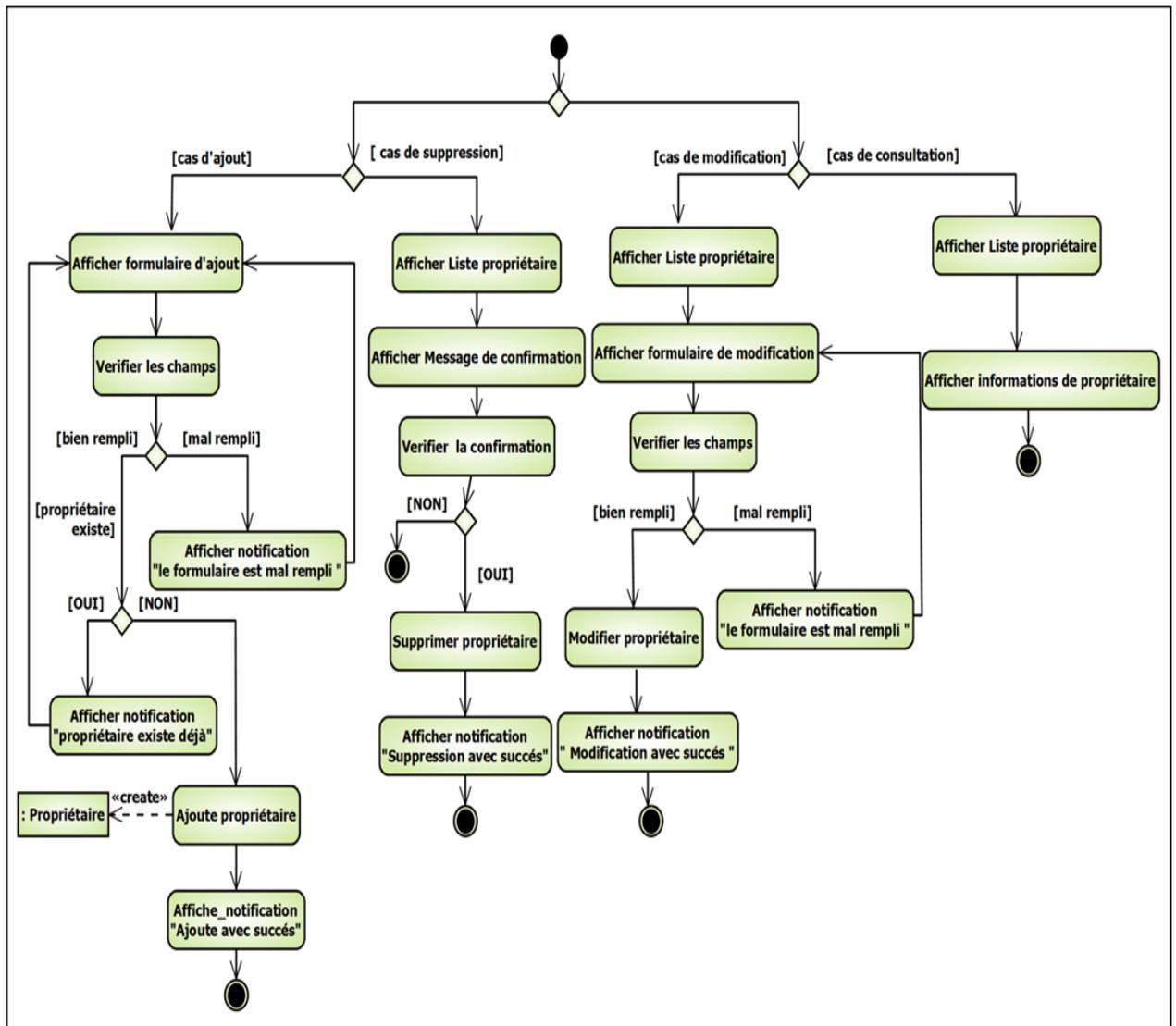


Figure 21: Diagramme d'activité «Gestion propriétaires».

• Diagramme de séquence : Gestion propriétaire

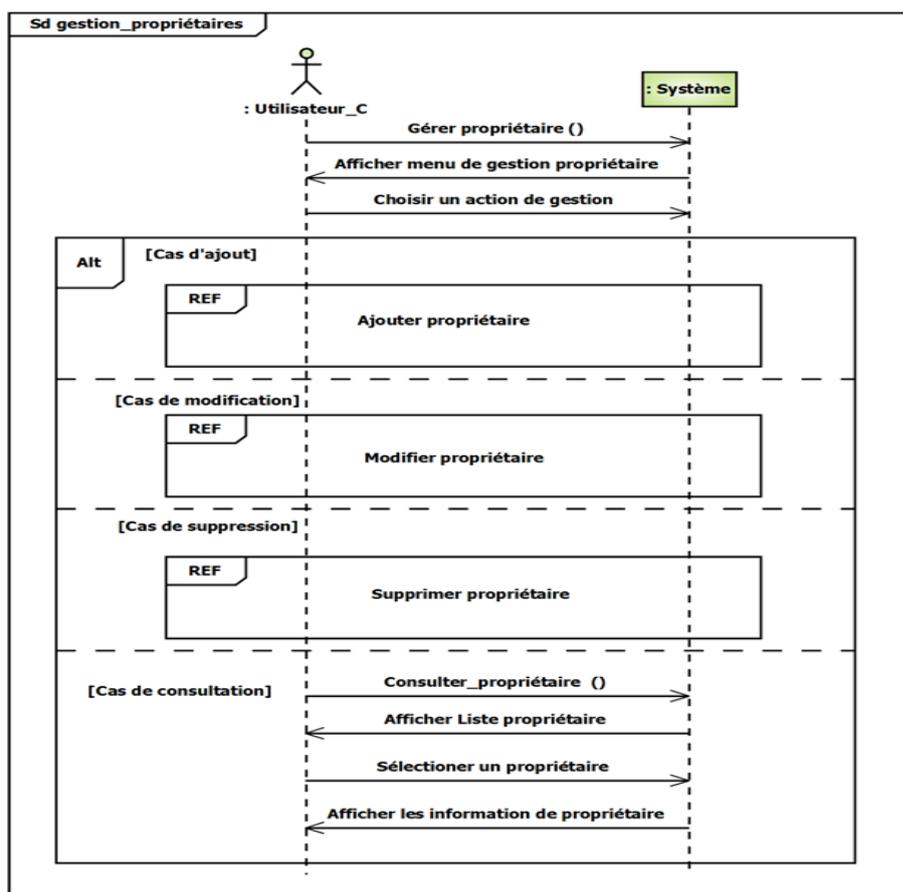


Figure 22: Diagramme de séquence «Gestion propriétaire».

• Diagramme de séquence : Ajouter propriétaire

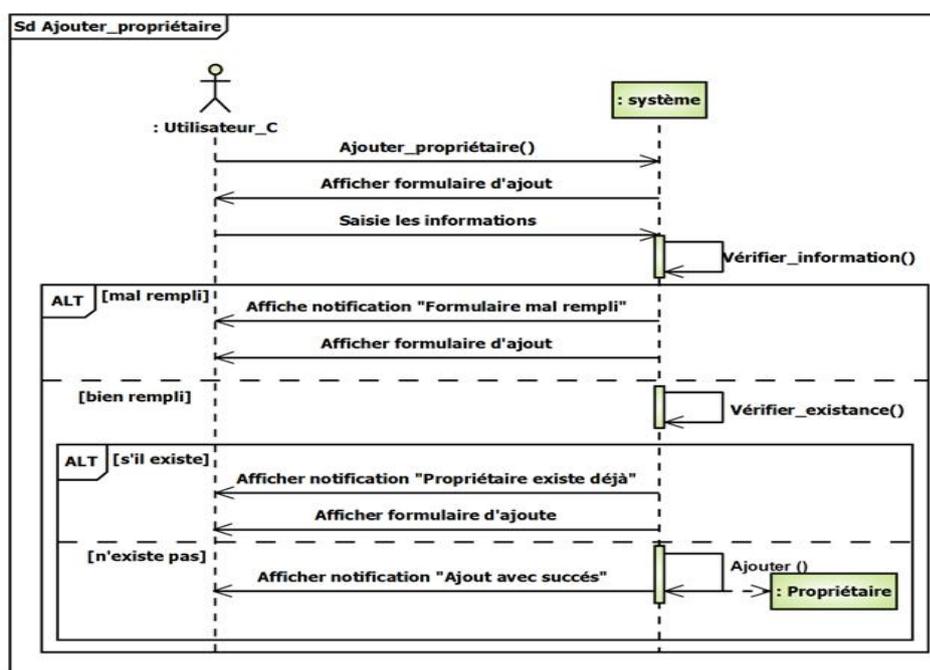


Figure 23: Diagramme de séquence «Ajouter propriétaire».

- Diagramme de séquence : Modifier propriétaire

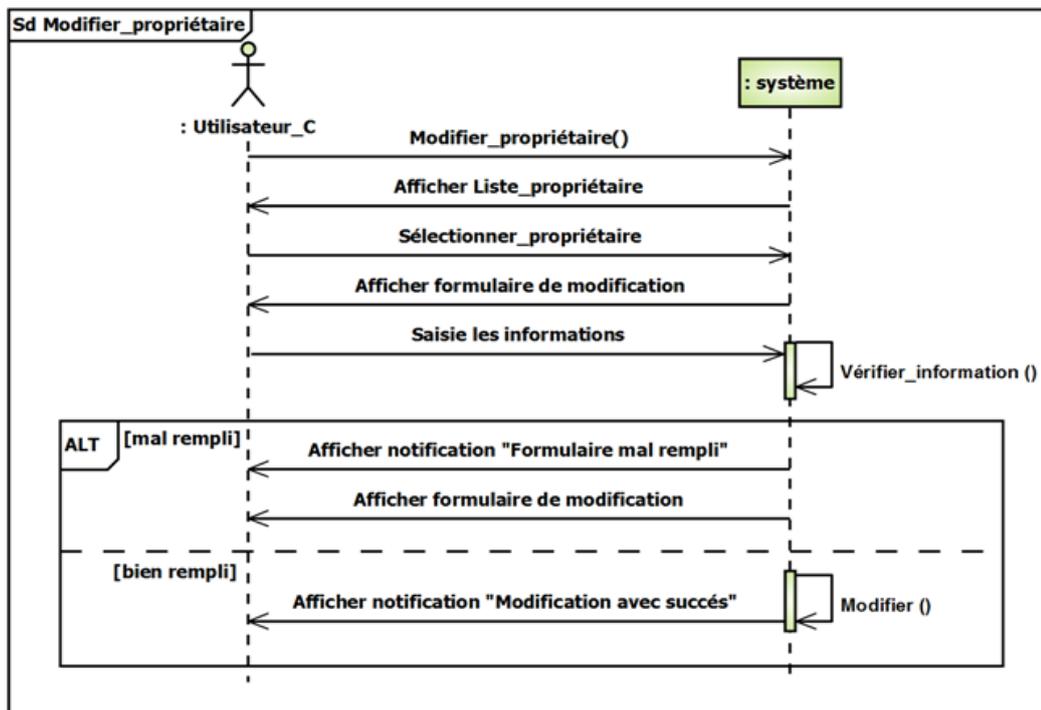


Figure 24: Diagramme de séquence «Modifier propriétaire».

- Diagramme de séquence : Supprimer propriétaire

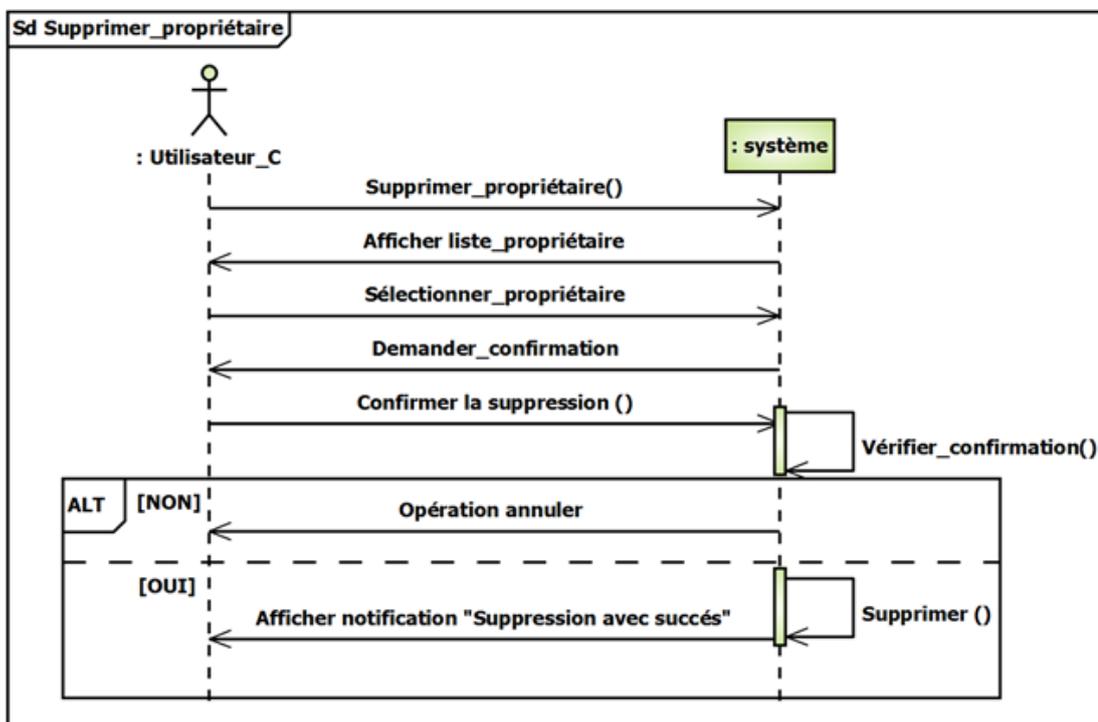


Figure 25: Diagramme de séquence «Supprimer propriétaire».

### 3.1.3. Location

Ce cas commence lorsque l'utilisateur accède au système et choisit «Location».

Cas d'utilisation	Location
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Consiste à créer, modifier, supprimer ou renouveler un contrat administratif de location pour un patrimoine donné.
<b>Pré condition</b>	L'utilisateur doit être Authentifié, Il existe au moins un patrimoine et un propriétaire.
<b>Post condition</b>	/
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la location.</li> <li>2. Le système affiche menu de la gestion du contrat location.</li> <li>3. L'utilisateur choisi l'une des actions suivante : <ul style="list-style-type: none"> <li>- Création d'un contrat de location.</li> <li>- Modification d'un contrat de location.</li> <li>- Suppression d'un contrat de location.</li> <li>- Consultation d'un contrat de location.</li> </ul> </li> </ol>

Tableau 18: Description textuelle du cas d'utilisation «Location».

#### 3.1.3.1. Création contrat de location

Ce cas commence lorsque l'utilisateur accède au système et choisit «Création\_contrat\_location».

Cas d'utilisation	Création_contrat_location
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Permettre à l'utilisateur la création d'un nouveau contrat de location.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un patrimoine et un propriétaire.</li> </ul>
<b>Post condition</b>	Le nouveau contrat enregistré par le système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la création d'un contrat de location.</li> <li>2. Le système affiche le formulaire de création d'un contrat administratif pour location.</li> </ol>

	<ol style="list-style-type: none"> <li>3. L'utilisateur remplit le formulaire par les informations de contrat qui veut créer.</li> <li>4. Le système lit et enregistre les informations de contrat.</li> <li>5. Le système affiche une notification «Création avec succès».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le contrat existe déjà ou les informations sont mal remplies, le système affiche une notification («le contrat existe déjà», ou « le formulaire est mal rempli »).</li> <li>2. Reprise de l'enchaînement du scénario nominal au point 2.</li> </ol>

**Tableau 19: Description textuelle du cas d'utilisation «Création\_contrat\_location».**

### 3.1.3.2. Modification contrat de location

Ce cas commence lors que l'utilisateur accède au système et choisit «Modification\_contrat\_location».

Cas d'utilisation	Modification_contrat_location
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Permettre à l'utilisateur de modifier les informations d'un contrat administratif de location.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un contrat de location dans la base.</li> </ul>
<b>Post condition</b>	Les modifications enregistrées par le système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la modification d'un contrat de location.</li> <li>2. Le système affiche la liste des contrats des locations.</li> <li>3. L'utilisateur consulte la liste des contrats administratifs de location, et choisit un contrat à modifier.</li> <li>4. Le système affiche le formulaire de modification avec les informations du contrat.</li> <li>5. L'utilisateur modifie les informations souhaitées et confirme.</li> <li>6. Le système enregistre les nouvelles informations saisies.</li> </ol>

	7. Le système affiche une notification «modification avec succès».
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où les informations sont mal remplies, le système affiche une notification « le formulaire est mal rempli ».</li> <li>2. Reprise de l'enchaînement du scenario nominal au point 4.</li> </ol>

**Tableau 20: Description textuelle du cas d'utilisation «Modification\_contrat\_location».**

### 3.1.3.3. Suppression contrat de location

Ce cas commence lorsque l'utilisateur accède au système et choisit «Suppression\_contrat\_location».

Cas d'utilisation		Suppression_contrat_location
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Permettre à l'utilisateur de supprimer un contrat de location.	
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un contrat de location dans la base.</li> </ul>	
<b>Post condition</b>	La suppression effectuée par le système.	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la suppression d'un contrat de location.</li> <li>2. Le système affiche la liste des contrats de location.</li> <li>3. L'utilisateur consulte la liste des contrats de location, et choisit un contrat à supprimer.</li> <li>4. Le système affiche un message de confirmation.</li> <li>5. L'utilisateur confirme la suppression.</li> <li>6. Le système supprime le contrat sélectionné.</li> <li>7. Le système affiche une notification « suppression avec succès ».</li> </ol>	
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où l'utilisateur annule la suppression, le système ignore la suppression.</li> </ol>	

**Tableau 21: Description textuelle du cas d'utilisation «Suppression\_contrat\_location».**

### 3.1.3.4. Consultation contrat de location

Ce cas commence lorsque l'utilisateur accède au système et choisit «Consultation\_contrat\_location » pour consulter les informations d'un contrat.

Cas d'utilisation		Consultation_contrat_location
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Permettre à l'utilisateur de consulter les informations d'un contrat.	
<b>Pré condition</b>	- L'utilisateur doit être Authentifié.	
<b>Post condition</b>	Les informations du contrat sont affichées.	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la consultation.</li> <li>2. Le système affiche la liste des contrats location.</li> <li>3. L'utilisateur choisit un contrat pour consulter les informations.</li> <li>4. Le système affiche les informations de ce contrat.</li> </ol>	

Tableau 22: Description textuelle du cas d'utilisation «Consultation\_contrat\_location».

• Diagramme d'activité : Location

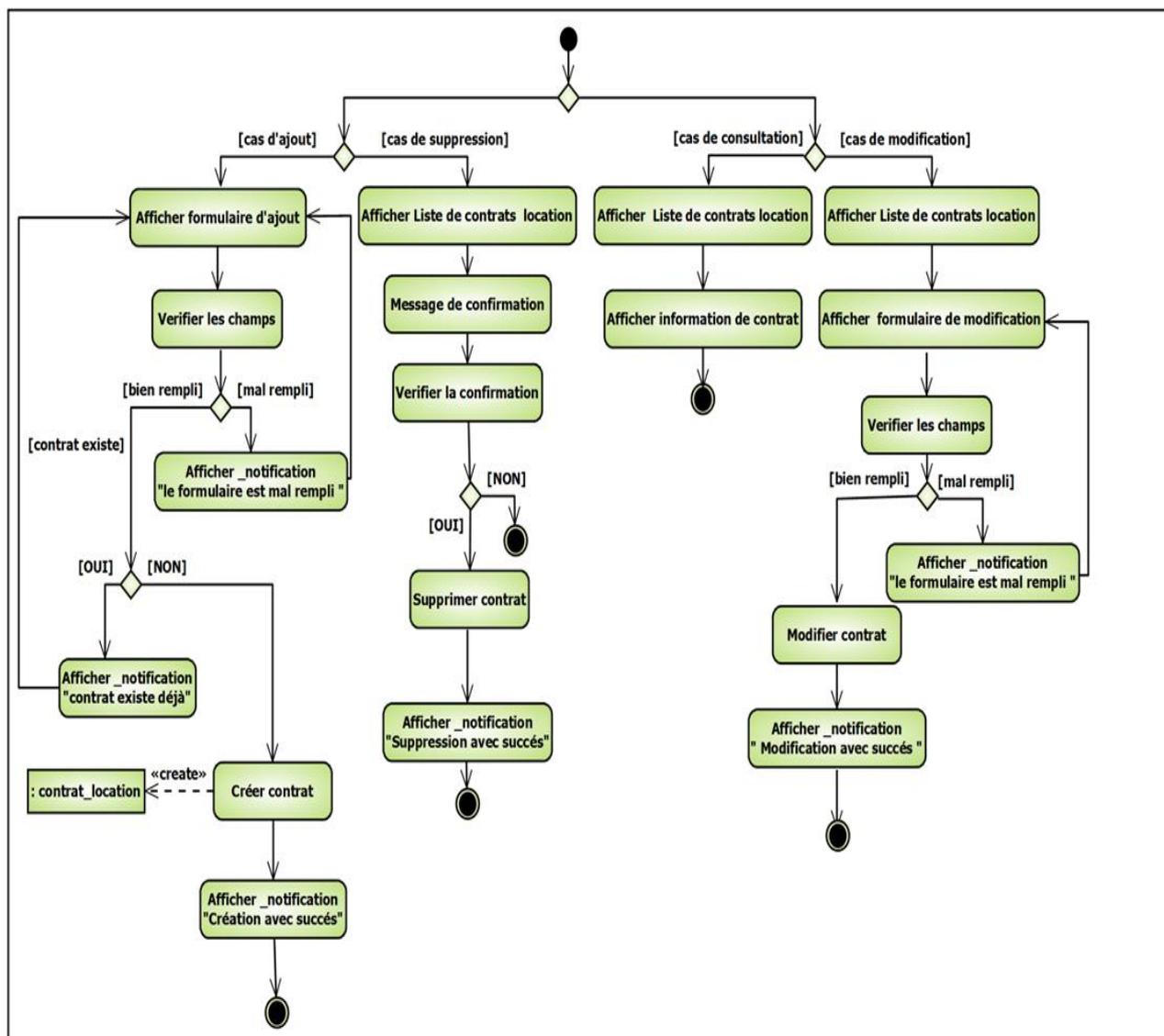


Figure 26: Diagramme d'activité «Gestion\_contracts\_location».

• Diagramme de séquence : Location

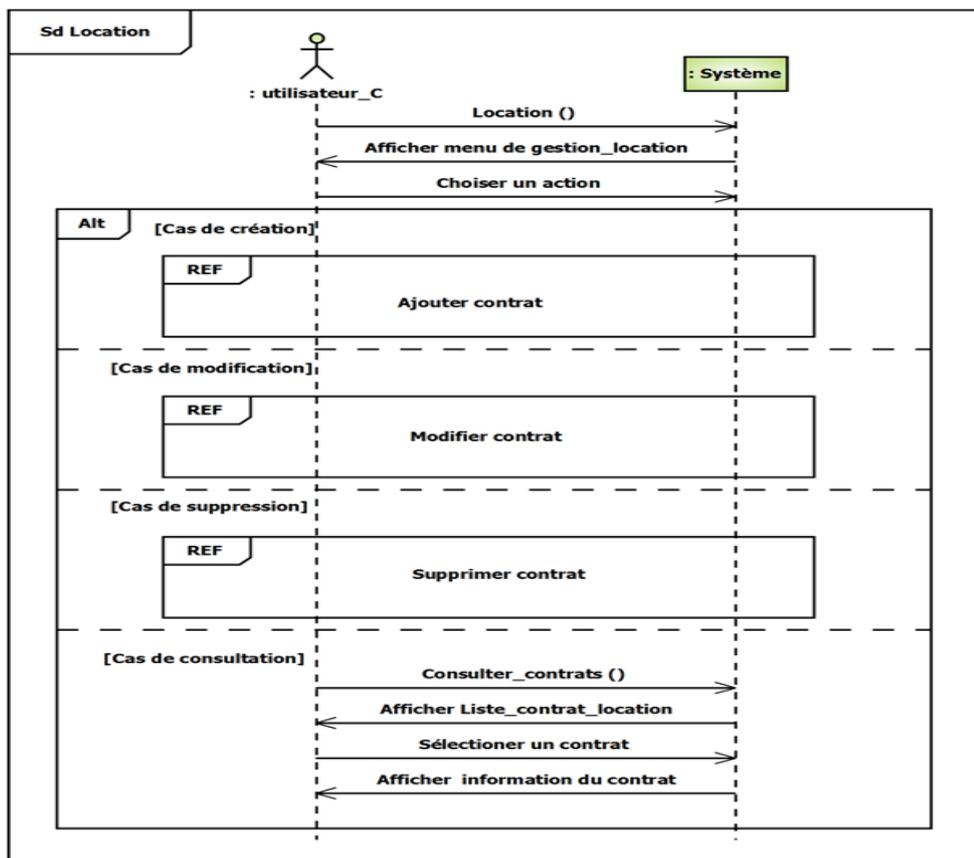


Figure 27: Diagramme de séquence «Location».

• Diagramme de séquence : Ajouter contrat de location

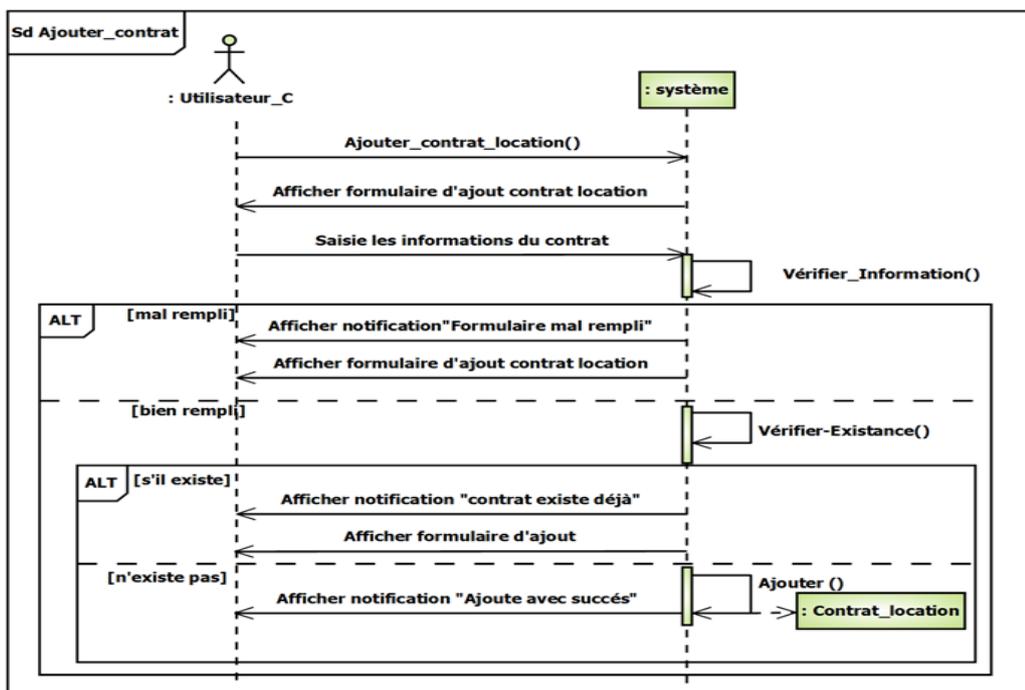


Figure 28: Diagramme de séquence «Ajouter\_contrat\_location».

• Diagramme de séquence : Modifier contrat de location

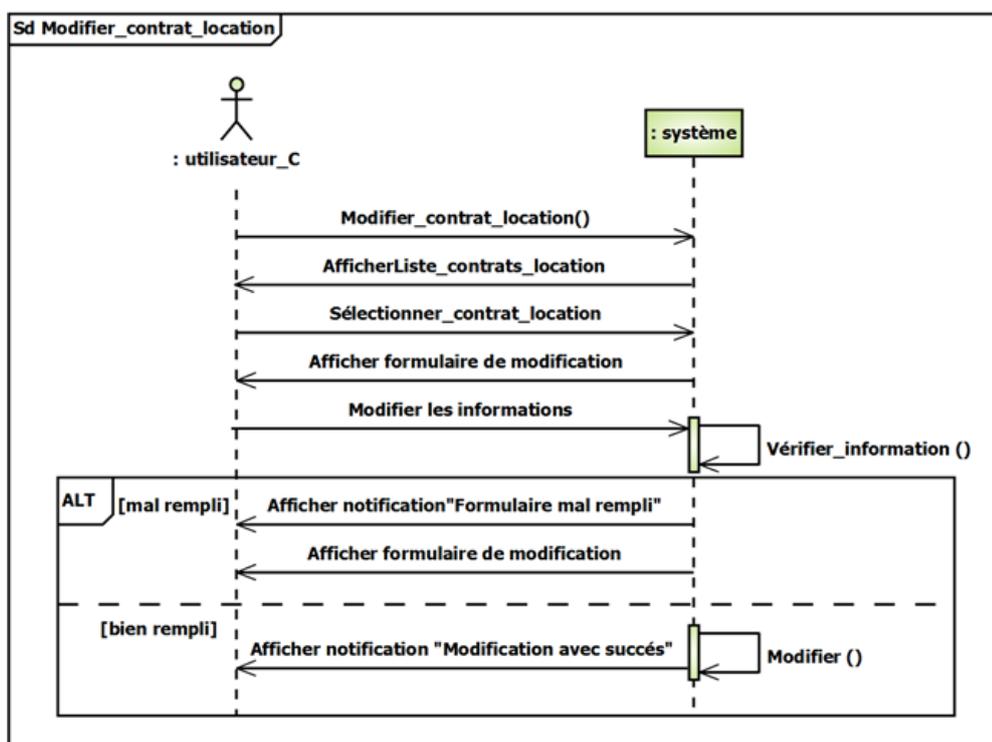


Figure 29: Diagramme de séquence «Modifier\_contrat\_location».

• Diagramme de séquence : Supprimer contrat

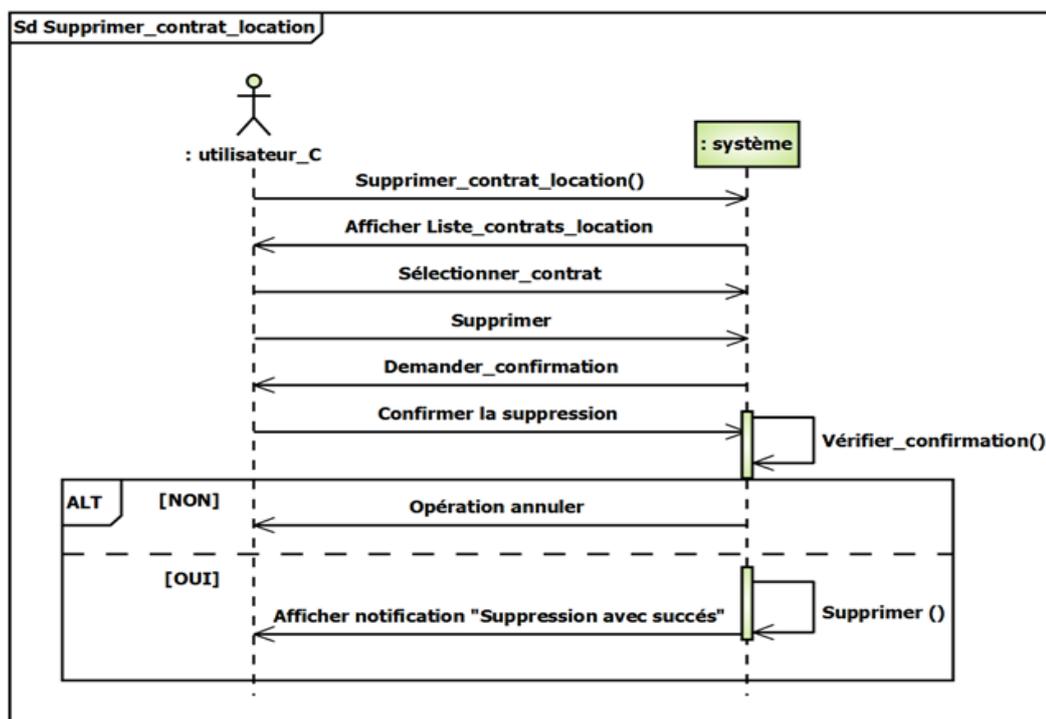


Figure 30: Diagramme de séquence «Supprimer\_contrat\_location».

### 3.1.4. Vente

Ce cas commence lorsque l'utilisateur accède au système et choisit «Vente».

Cas d'utilisation		Vente
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Consiste à créer, modifier, supprimer ou consulter un contrat administratif de vente pour un patrimoine donné.	
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un patrimoine et un propriétaire.</li> </ul>	
<b>Post condition</b>	/	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système l'opération de vente.</li> <li>2. Le système affiche menu de la gestion du contrat vente.</li> <li>3. L'utilisateur choisit l'une des actions suivante : <ul style="list-style-type: none"> <li>- Création d'un contrat de vente.</li> <li>- Modification d'un contrat de vente.</li> <li>- Suppression d'un contrat de vente.</li> <li>- Consultation d'un contrat de vente.</li> </ul> </li> </ol>	

Tableau 23: Description textuelle du cas d'utilisation «Vente».

#### 3.1.4.1. Création contrat vente

Ce cas commence lorsque l'utilisateur accède au système et choisit «Création\_contrat\_Vente».

Cas d'utilisation		Création_contrat_vente
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Permettre à l'utilisateur la création d'un nouveau contrat de vente.	
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un patrimoine dans la base.</li> </ul>	
<b>Post condition</b>	Le nouveau contrat enregistré par le système.	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la création d'un contrat de vente.</li> <li>2. Le système affiche le formulaire de création d'un contrat administratif pour vente.</li> <li>3. L'utilisateur remplit le formulaire par les informations de</li> </ol>	

	<p>contrat qui veut créer.</p> <ol style="list-style-type: none"> <li>4. Le système enregistre les informations de contrat.</li> <li>5. Le système affiche une notification «Création avec succès».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le contrat existe déjà ou les informations sont mal remplies, le système affiche une notification («le contrat existe déjà», ou « le formulaire est mal rempli »).</li> <li>2. Reprise de l'enchaînement du scénario nominal au point 2.</li> </ol>

**Tableau 24: Description textuelle du cas d'utilisation «Création\_contrat\_vente».**

### 3.1.4.2. Modification contrat vente

Ce cas commence lors que l'utilisateur accède au système et choisit «Modification\_contrat\_vente».

Cas d'utilisation	Modification_contrat_vente
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Permettre à l'utilisateur de modifier les informations d'un contrat administratif de vente.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il exister au moins un contrat de vente dans la base.</li> </ul>
<b>Post condition</b>	Les modifications enregistrées par le système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la modification d'un contrat de vente.</li> <li>2. Le système affiche la liste des contrats des ventes.</li> <li>3. L'utilisateur consulte la liste des contrats administratifs de vente, et choisit le contrat à modifier.</li> <li>4. Le système affiche le formulaire de modification avec les informations de ce contrat.</li> <li>5. L'utilisateur modifie les informations souhaitées et confirme.</li> <li>6. Le système lit et enregistre les nouvelles informations saisies.</li> </ol>

	7. Le système affiche une notification «modification avec succès».
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où les informations sont mal remplies, le système affiche une notification « le formulaire est mal rempli ».</li> <li>2. Reprise de l'enchaînement du scénario nominal au point 4.</li> </ol>

**Tableau 25: Description textuelle du cas d'utilisation «Modification\_contrat\_vente».**

### 3.1.4.3. Suppression contrat de vente

Ce cas commence lorsque l'utilisateur accède au système et choisit «Suppression\_contrat\_vente ».

Cas d'utilisation		Suppression_contrat_vente
<b>Acteur</b>		Utilisateur_C
<b>But</b>		Permettre à l'utilisateur de supprimer un contrat de vente.
<b>Pré condition</b>		<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un contrat de vente dans la base.</li> </ul>
<b>Post condition</b>		La suppression effectuée par le système.
<b>Scénario nominal</b>		<ol style="list-style-type: none"> <li>1. L'utilisateur demande la suppression d'un contrat de vente.</li> <li>2. Le système affiche la liste des contrats de vente.</li> <li>3. L'utilisateur consulte la liste des contrats de vente, et choisit le contrat à supprimer.</li> <li>4. Le système affiche un message de confirmation.</li> <li>5. L'utilisateur confirme la suppression.</li> <li>6. Le système supprime le contrat sélectionné.</li> <li>7. Le système affiche une notification « suppression avec succès ».</li> </ol>
<b>Scénario alternatif</b>		<ol style="list-style-type: none"> <li>1. Le cas où l'utilisateur annule la suppression, le système ignore la suppression.</li> </ol>

**Tableau 26: Description textuelle du cas d'utilisation «Suppression\_contrat\_vente».**

### 3.1.4.4. Consultation contrat de vente

Ce cas commence lorsque l'utilisateur accède au système et choisit «Consultation\_contrat\_vente » pour consulter les informations d'un contrat.

Cas d'utilisation		Consultation_contrat_vente
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Permettre à l'utilisateur de consulter les informations d'un contrat de vente.	
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être identifié.</li> <li>- Il existe au moins un contrat de vente dans la base.</li> </ul>	
<b>Post condition</b>	Les informations du contrat sont affichées.	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la consultation d'un contrat de vente.</li> <li>2. Le système affiche la liste des contrats vente.</li> <li>3. L'utilisateur choisit un contrat pour consulter les informations.</li> <li>4. Le système affiche les informations de ce contrat.</li> </ol>	

Tableau 27: Description textuelle du cas d'utilisation «Consultation\_contrat\_vente».

### 3.1.5. Recherche Multicritères commune

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_Multicritères\_commune».

Cas d'utilisation		Recherche_Multicritères_commune
<b>Acteur</b>	Utilisateur_C	
<b>But</b>	Consiste à rechercher un patrimoine ou rechercher un contrat.	
<b>Pré condition</b>	L'utilisateur doit être Authentifié.	
<b>Post condition</b>	/	
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la recherche.</li> <li>2. Le système présente les actions de la recherche.</li> <li>3. L'utilisateur choisit l'une actions suivantes :</li> </ol>	

	<ul style="list-style-type: none"> <li>- Recherche patrimoine.</li> <li>- Recherche contrat.</li> <li>- Recherche propriétaire.</li> </ul>
--	--

**Tableau 28: Description textuelle du cas d'utilisation «Recherche\_Multicritères\_commune».**

### 3.1.5.1. Recherche patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_patrimoine» pour rechercher un patrimoine.

Cas d'utilisation	Recherche_patrimoine
<b>Acteurs</b>	Utilisateur_C
<b>But</b>	Obtenir les informations d'un patrimoine.
<b>Pré condition</b>	L'utilisateur doit être Authentifié.
<b>Post condition</b>	Affiche les informations d'un patrimoine.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la recherche d'un patrimoine.</li> <li>2. Le système affiche le formulaire de recherche qui contient les critères de recherche (type de patrimoine, catégorie de patrimoine, code de patrimoine, type de contrat, date d'enregistrement de patrimoine).</li> <li>3. L'utilisateur choisi un ou plusieurs critères de recherche               <ol style="list-style-type: none"> <li>3.1. Recherche par (type de patrimoine, catégorie de patrimoine, date d'enregistrement, type de contrat).                   <ul style="list-style-type: none"> <li>- Le système afficher la liste des patrimoines selon les critères.</li> <li>- L'utilisateur sélectionne un patrimoine.</li> <li>- Le système affiche les informations du patrimoine sélectionné.</li> </ul> </li> <li>3.2. Recherche par code de patrimoine.                   <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de patrimoine.</li> <li>- Le système affiche les informations de ce patrimoine.</li> </ul> </li> </ol> </li> </ol>

<b>Scénario alternatif</b>	1. Le cas où le patrimoine n'existe pas, le système notifie la non existence, et reprise de l'enchaînement du scénario nominal au point 2.
----------------------------	--

**Tableau 29: Description textuelle du cas d'utilisation «Recherche\_patrimoine».**

### 3.1.5.2. Recherche contrat

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_contrat» pour rechercher un contrat.

Cas d'utilisation	Recherche_contrat
<b>Acteurs</b>	Utilisateur_C
<b>But</b>	Obtenir les informations d'un contrat.
<b>Pré condition</b>	L'utilisateur doit être Authentifié.
<b>Post condition</b>	Affiche les informations d'un contrat.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la recherche d'un contrat.</li> <li>2. Le système affiche le formulaire de recherche qui contient les critères de recherche (type de contrat, type et/ou la catégorie du patrimoine, code du contrat, date d'enregistrement du contrat, le code du propriétaire).</li> <li>3. L'utilisateur choisi un ou plusieurs critères de recherche               <ol style="list-style-type: none"> <li>3.1. Recherche par (type de patrimoine, catégorie de patrimoine, date d'enregistrement, type de contrat).                   <ul style="list-style-type: none"> <li>- Le système afficher la liste des contrats selon les critères.</li> <li>- L'utilisateur sélectionne un contrat.</li> <li>- Le système affiche les informations de contrat.</li> </ul> </li> <li>3.2. Recherche par code de contrat.                   <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de contrat.</li> <li>- Le système afficher les informations de ce contrat.</li> </ul> </li> <li>3.3. Recherche par le code de propriétaire.                   <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de propriétaire.</li> <li>- Le système afficher la liste des contrats de ce propriétaire.</li> </ul> </li> </ol> </li> </ol>

	<ul style="list-style-type: none"> <li>- L'utilisateur sélectionne un contrat.</li> <li>- Le système affiche les informations de contrat.</li> </ul>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le contrat n'existe pas, le système notifie le non existence, et reprise de l'enchaînement du scénario nominal au point 2.</li> <li>2. Le cas où le propriétaire n'existe pas, le système notifie la non existence, et reprise de l'enchaînement du scénario nominal au point 2.</li> </ol>

**Tableau 30: Description textuelle du cas d'utilisation «Recherche\_contrat».**

### 3.1.5.3. Recherche propriétaire

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_propriétaire» pour rechercher un propriétaire.

Cas d'utilisation Recherche_propriétaire	
<b>Acteurs</b>	Utilisateur_C
<b>But</b>	Obtenir les informations d'un propriétaire.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur est authentifié.</li> </ul>
<b>Post condition</b>	Affiche les informations d'un propriétaire.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la recherche d'un propriétaire.</li> <li>2. Le système affiche le formulaire de recherche qui contient les critères de recherche (type de contrat, code de propriétaire).</li> <li>3. L'utilisateur choisi un ou plusieurs critères de recherche               <ol style="list-style-type: none"> <li>3.1.Recherche par type de contrat.                   <ul style="list-style-type: none"> <li>- L'utilisateur sélectionne un type de contrat.</li> <li>- système afficher la liste des propriétaires selon le type de contrat.</li> <li>- L'utilisateur sélectionne un propriétaire.</li> <li>- Le système affiche les informations de propriétaire.</li> </ul> </li> </ol> </li> </ol>

	<p>3.2.Recherche par code de propriétaire.</p> <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de propriétaire.</li> <li>- Le système afficher les informations de ce propriétaire.</li> </ul>
<p><b>Scénario alternatif</b></p>	<p>1. Le cas où le propriétaire n'existe pas, le système notifie la non existence, et reprise de l'enchainement du scénario nominal au point 2.</p>

Tableau 31: Description textuelle du cas d'utilisation «Recherche\_contrat».

• Diagramme d'activité : Recherche\_Multicritères\_commune

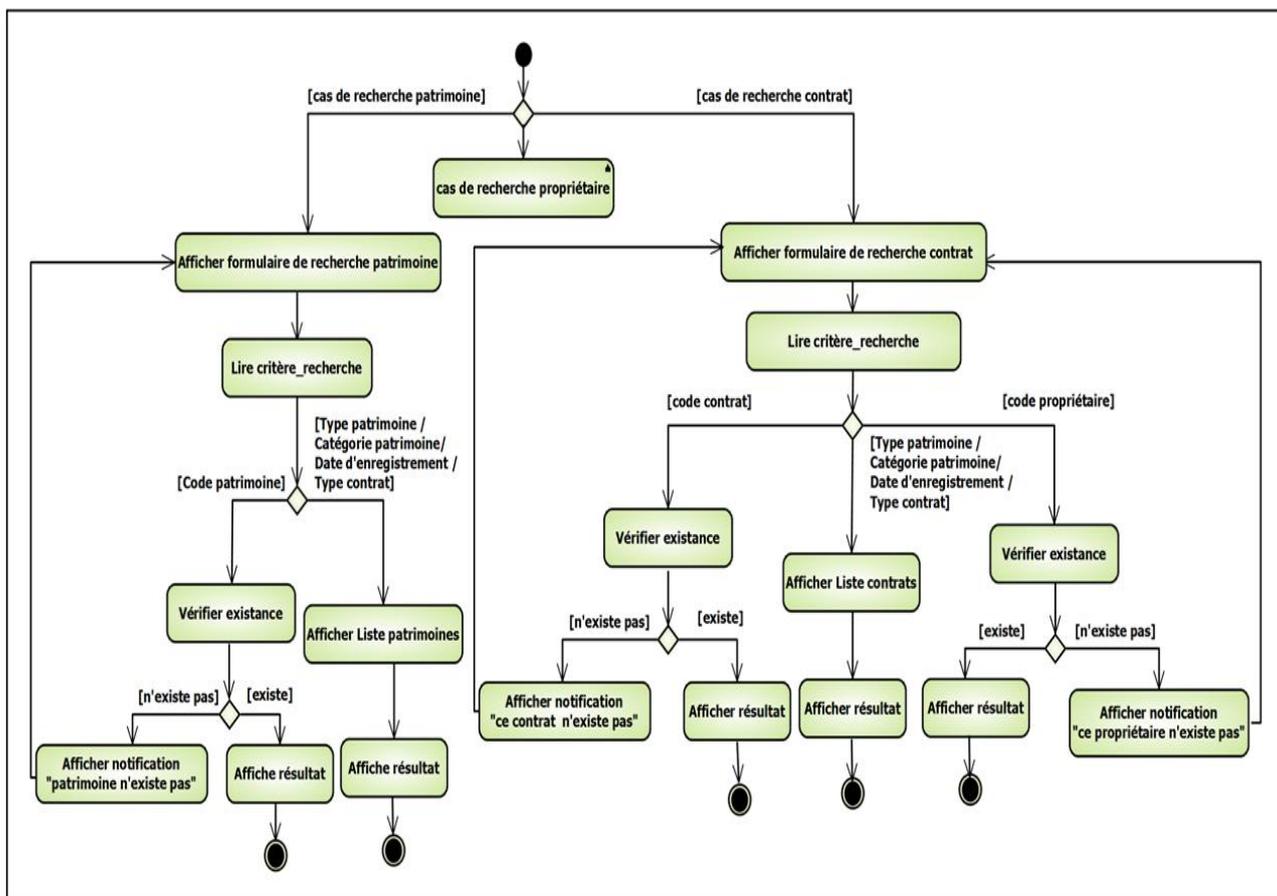


Figure 31: Diagramme d'activité «Recherche\_Multicritères\_commune».

- Diagramme d'activité : Cas de recherche propriétaire

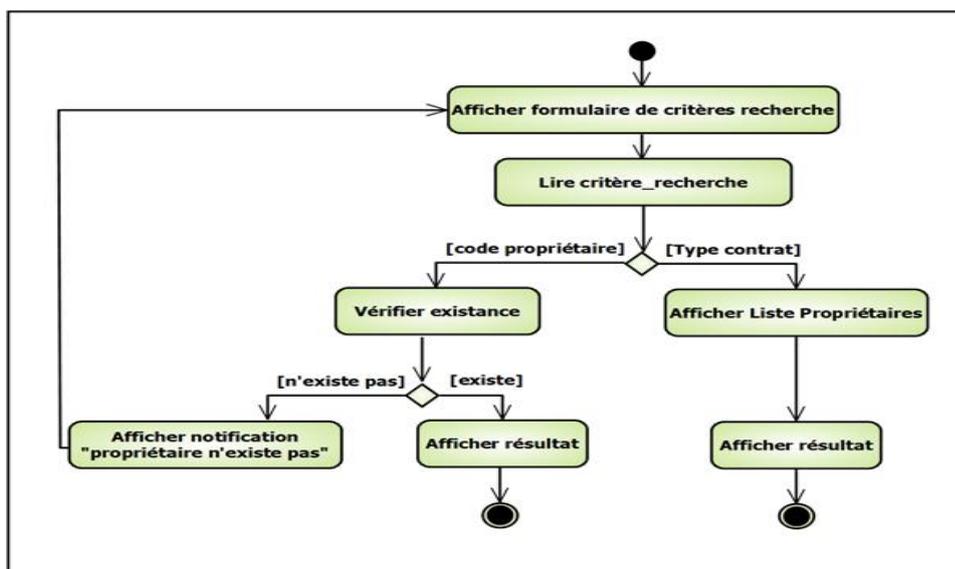


Figure 32: Diagramme d'activité «Recherche\_propriétaire».

- Diagramme de séquence : Recherche\_Multicritères\_commune

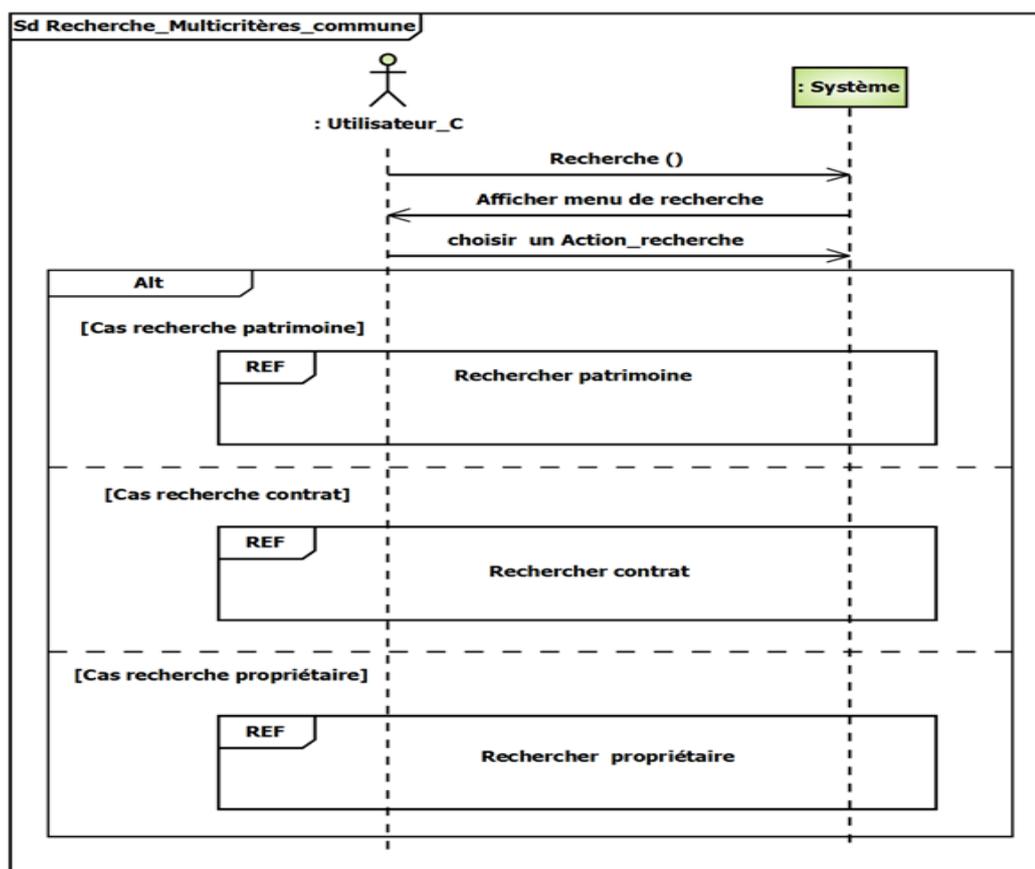


Figure 33: Diagramme de séquence «Recherche\_Multicritères\_commune».

• Diagramme de séquence : Rechercher patrimoine commune

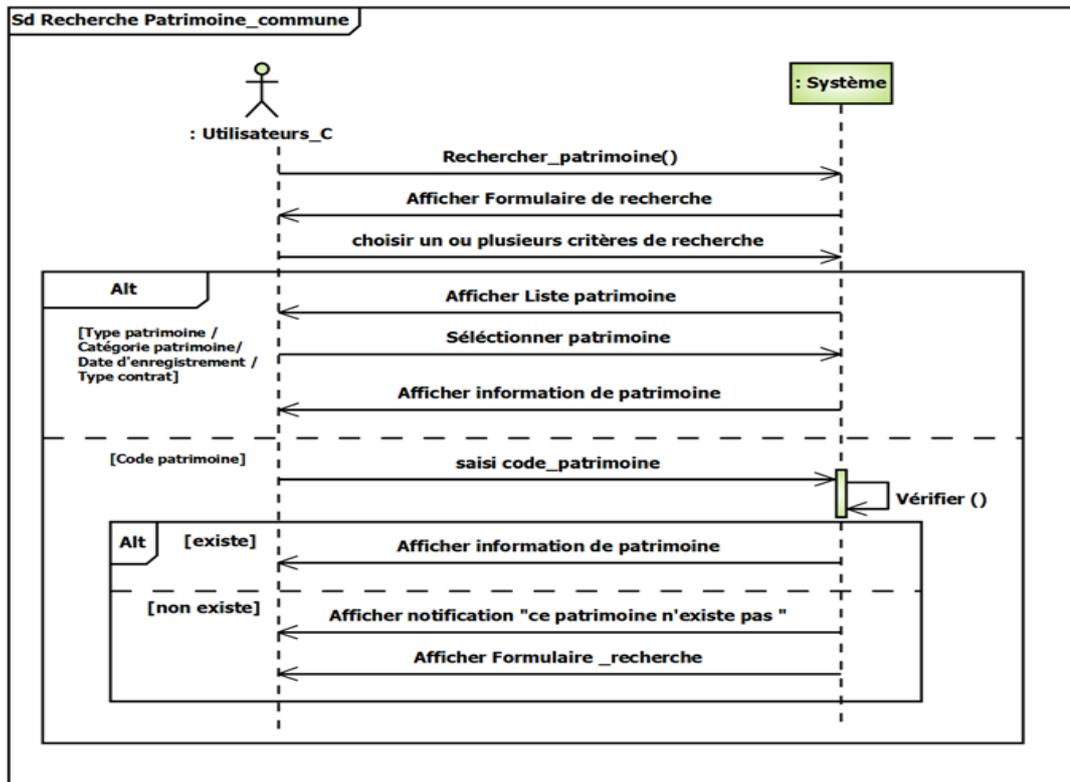


Figure 34: Diagramme de séquence «Rechercher\_patrimoine\_commune».

• Diagramme de séquence : Rechercher contrat commune

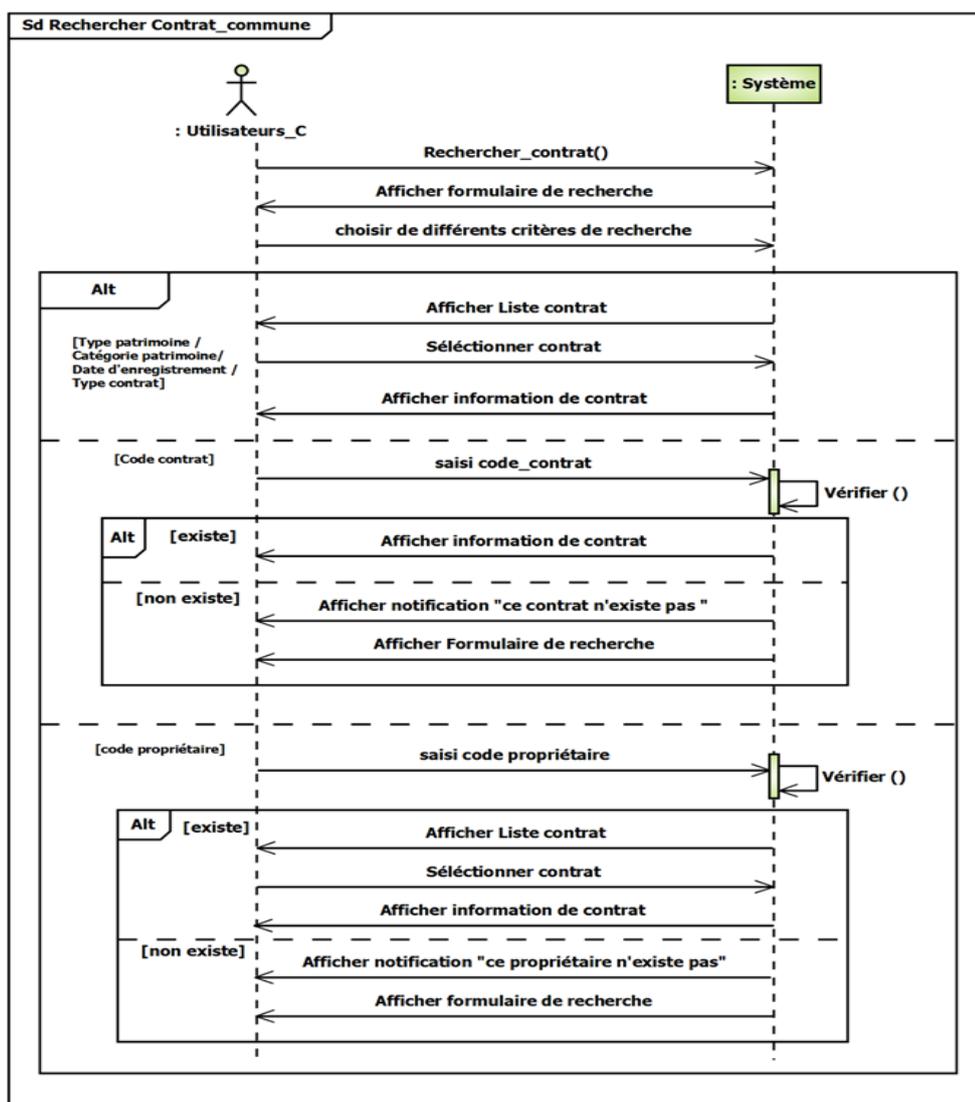


Figure 35: Diagramme de séquence «Rechercher\_contrat\_commune».

• Diagramme de séquence : Rechercher propriétaire de commune

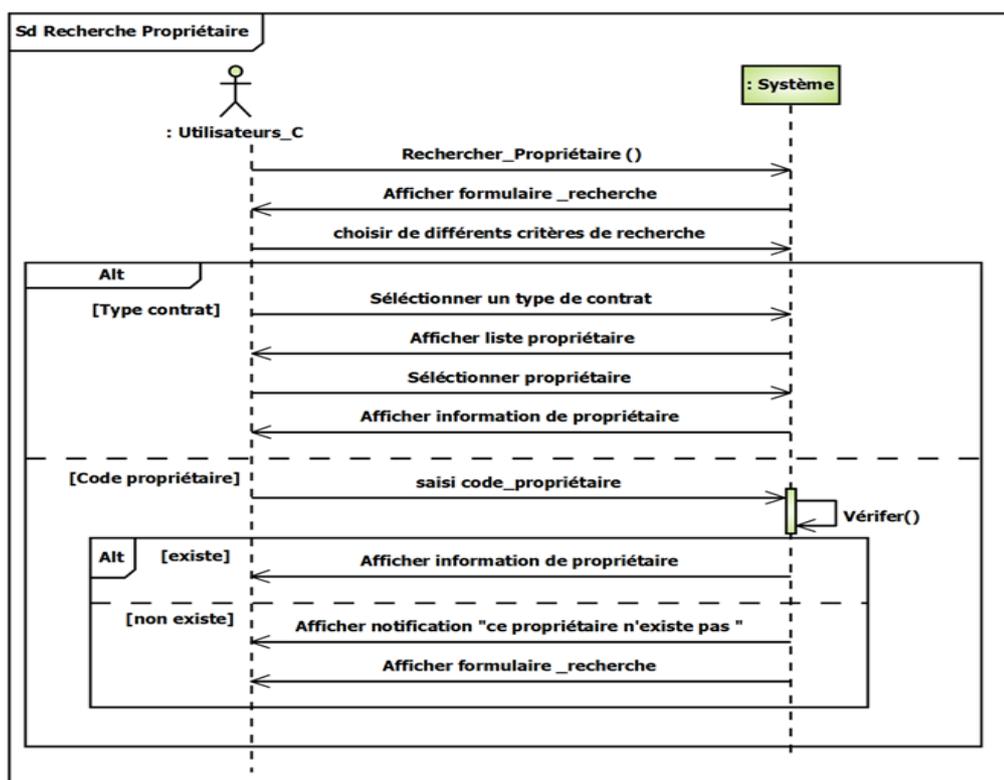


Figure 36: Diagramme de séquence «Rechercher\_propriétaire\_commune».

3.1.6. Statistique commune

Ce cas commence lorsque l'utilisateur accède au système et choisit «Statistique commune».

Cas d'utilisation	Statistique_commune
<b>Acteur</b>	Utilisateur_C
<b>But</b>	Consiste à calculer les statistiques d'un patrimoine ou un contrat.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un patrimoine et un contrat dans la base de données.</li> </ul>
<b>Post condition</b>	/
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système le calcul des statistiques.</li> <li>2. Le système affiche menu de calcul des statistiques.</li> <li>3. L'utilisateur choisit l'une des actions suivantes :                             <ul style="list-style-type: none"> <li>- Statistique patrimoine.</li> <li>- Statistique contrat.</li> </ul> </li> </ol>

Tableau 32: Description textuelle du cas d'utilisation « statistique commune».

### 3.1.6.1. Statistique patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Statistique\_patrimoine» pour calculer les statistiques d'un patrimoine.

Cas d'utilisation	Statistique_patrimoine
<b>Acteurs</b>	Utilisateur_C
<b>But</b>	Obtenir les statistiques d'un patrimoine.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être Authentifié.</li> <li>- Il existe au moins un patrimoine dans la base de données.</li> </ul>
<b>Post condition</b>	Affiche les statistiques d'un patrimoine.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système le calcul des statistiques d'un patrimoine.</li> <li>2. Le système affiche le formulaire de statistique qui contient les critères de le calcul (type de patrimoine, catégorie de patrimoine, nature de patrimoine, revenus de patrimoine).</li> <li>3. L'utilisateur choisi un ou plusieurs critères de calcul.               <ol style="list-style-type: none"> <li>3.1. Calcule par type de patrimoine et/ou la catégorie de patrimoine.                   <ul style="list-style-type: none"> <li>- L'utilisateur sélectionne un type et /ou une catégorie.</li> <li>- Le système fait le calcul et affiche le résultat.</li> </ul> </li> <li>3.2. Calcule par la nature de patrimoine.                   <ul style="list-style-type: none"> <li>- L'utilisateur saisi la nature de patrimoine.</li> <li>- Le système fait le calcul et affiche le résultat de calcul.</li> </ul> </li> </ol> </li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le patrimoine n'existe pas, le système notifie le non existence, et reprise de l'enchaînement du scénario nominal au point 2.</li> </ol>

Tableau 33: Description textuelle du cas d'utilisation « statistique patrimoine».

### 3.1.6.2. Statistique contrats

Ce cas commence lorsque l'utilisateur accède au système et choisit «Statistique\_contrat» pour calculer les Statistique des contrats.

Cas d'utilisation	Statistique_contrat
<b>Acteurs</b>	Utilisateur_C
<b>But</b>	Obtenir les statistiques des contrats.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur est authentifié.</li> <li>- Il existe au moins un contrat dans la base de données.</li> </ul>
<b>Post condition</b>	Affiche les statistiques d'un contrat.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système le calcul des statistiques d'un contrat.</li> <li>2. Le système affiche le formulaire de statistiques qui contient le critère de calcul (type du contrat (vente, location), code du propriétaire).</li> <li>3. L'utilisateur choisit un ou plusieurs critères pour le calcul. <ol style="list-style-type: none"> <li>3.1. Calcul par type de contrat. <ul style="list-style-type: none"> <li>- L'utilisateur choisit le type de contrat.</li> <li>- Le système calcule et affiche le résultat.</li> </ul> </li> <li>3.2. Calcul par code du propriétaire. <ul style="list-style-type: none"> <li>- L'utilisateur saisit le code du propriétaire.</li> <li>- Le système calcule et affiche le résultat.</li> </ul> </li> </ol> </li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le propriétaire n'existe pas, le système notifie la non existence, et reprend l'enchaînement du scénario nominal au point 2.</li> </ol>

Tableau 34: Description textuelle du cas d'utilisation « statistique contrat ».

• Diagramme d'activité : Statistique\_commune

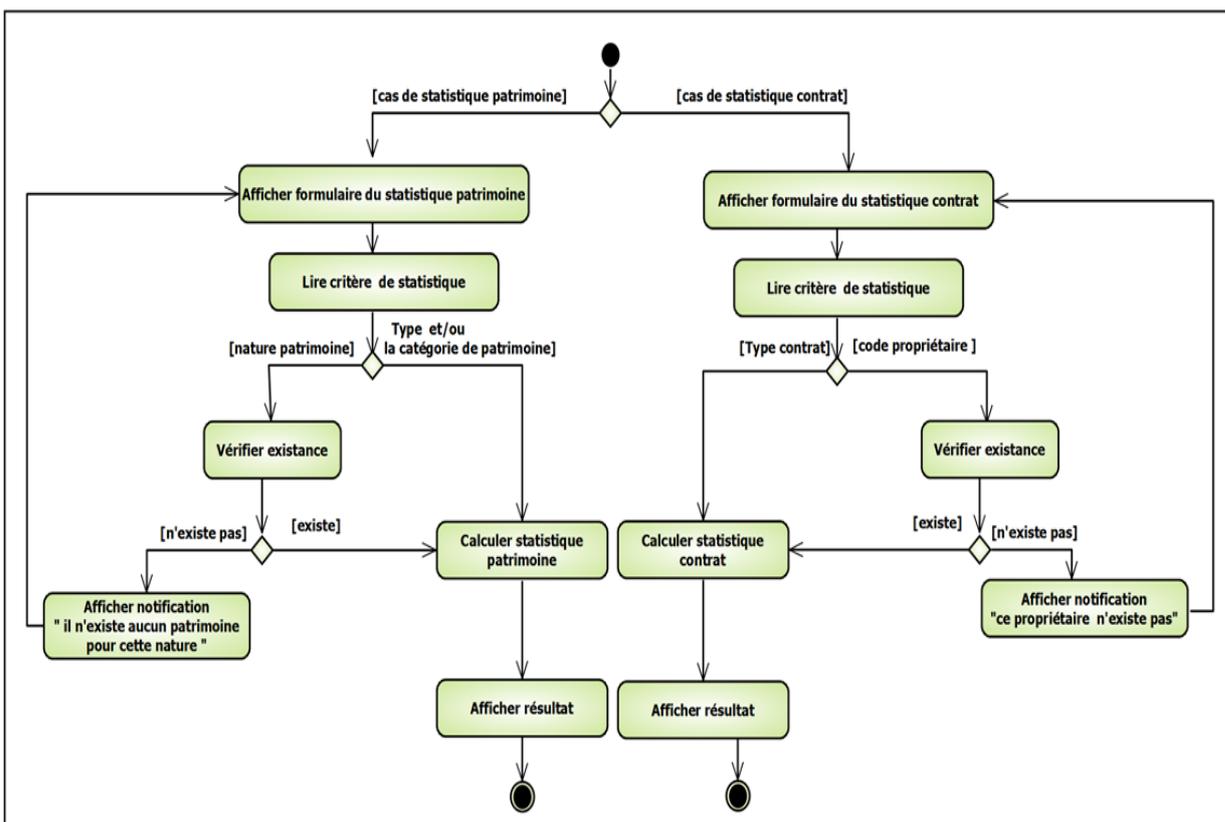


Figure 37: Diagramme d'activité «Statistique\_commune».

• Diagramme de séquence :Statistique\_commune

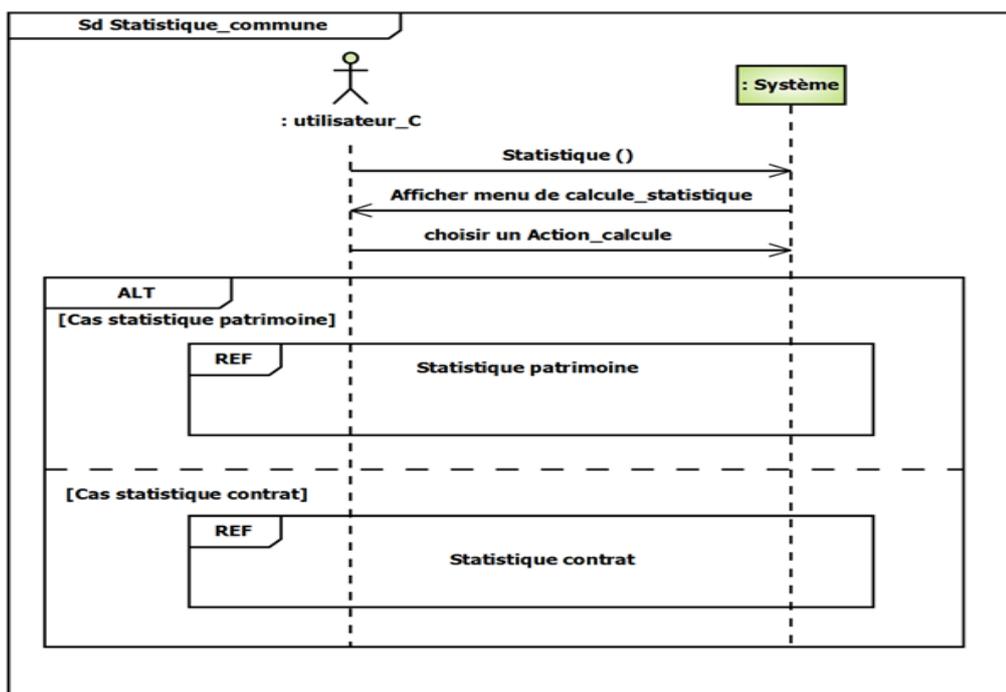


Figure 38: Diagramme de séquence «Statistique\_commune».

- Diagramme de séquence : Statistique patrimoine

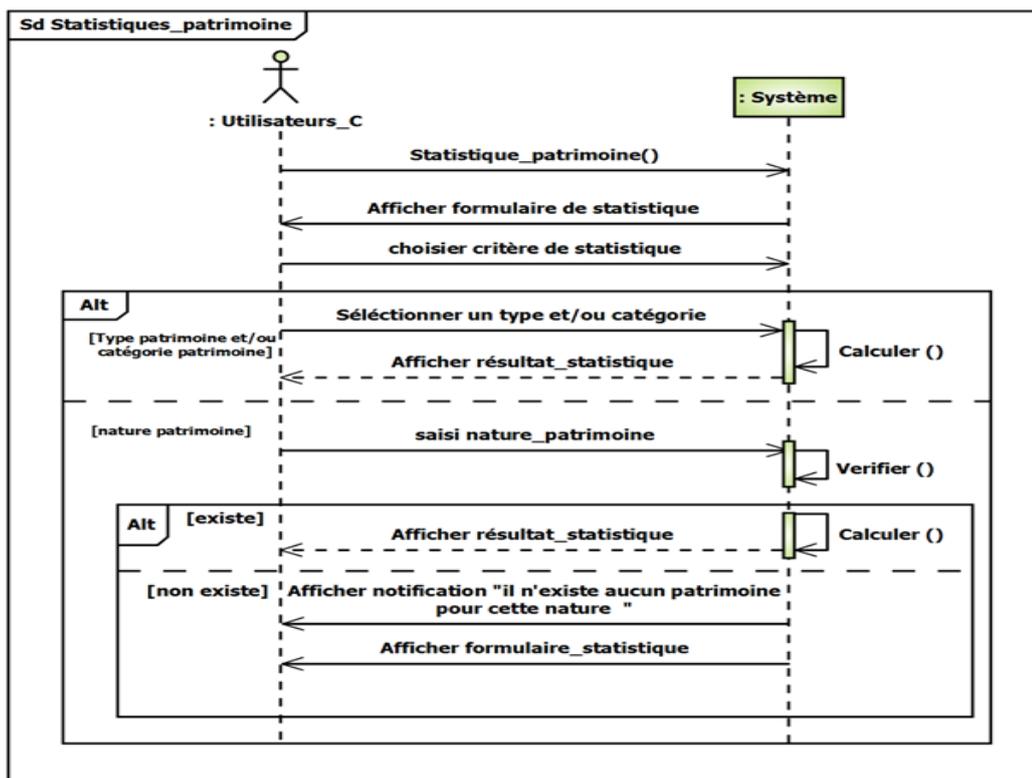


Figure 39: Diagramme de séquence «Statistique\_patrimoine».

- Diagramme de séquence : Statistique contrat

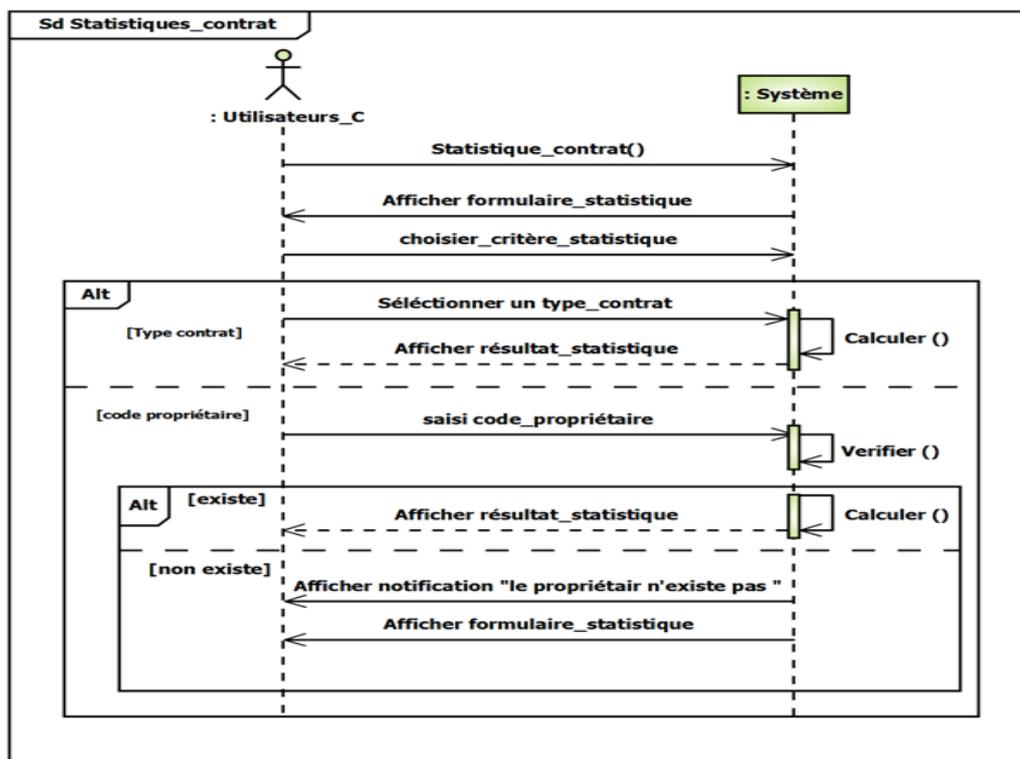


Figure 40: Diagramme de séquence «Statistique\_contrat».

## 3.2. Au niveau de la wilaya

### 3.2.1. Validation des contrats

Ce cas commence lorsque l'utilisateur accède au système et choisit «Validation\_contrats».

Cas d'utilisation	Validation_contrat
<b>Acteurs</b>	Utilisateur_W
<b>But</b>	Permettre à l'utilisateur de valider les contrats administratifs des communes.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur est authentifié.</li> <li>- Il existe au moins un contrat dans la base.</li> </ul>
<b>Post condition</b>	Validation des contrats est effectuée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la validation des contrats.</li> <li>2. Le système affiche la fenêtre de la validation.</li> <li>3. L'utilisateur sélectionne la commune et le type de contrat.</li> <li>4. Le système affiche la liste des contrats non validés.</li> <li>5. L'utilisateur sélectionne un contrat pour la validation.</li> <li>6. Le système affiche les informations du contrat.</li> <li>7. L'utilisateur consulte et vérifie les informations du contrat pour valider.</li> <li>8. Le système enregistre la validation et affiche la notification « le contrat est validée ».</li> <li>9. Le système envoie la notification à la commune.</li> </ol>
<b>Scénario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où les informations du contrat sont incorrects l'utilisateur refuse ce contrat</li> <li>2. Le système affiche la notification « le contrat est refusé ».</li> <li>3. Le système envoie la notification à la commune.</li> </ol>

Tableau 35: Description textuelle du cas d'utilisation « validation des contrats».

• Diagramme d'activité : Validation des contrats

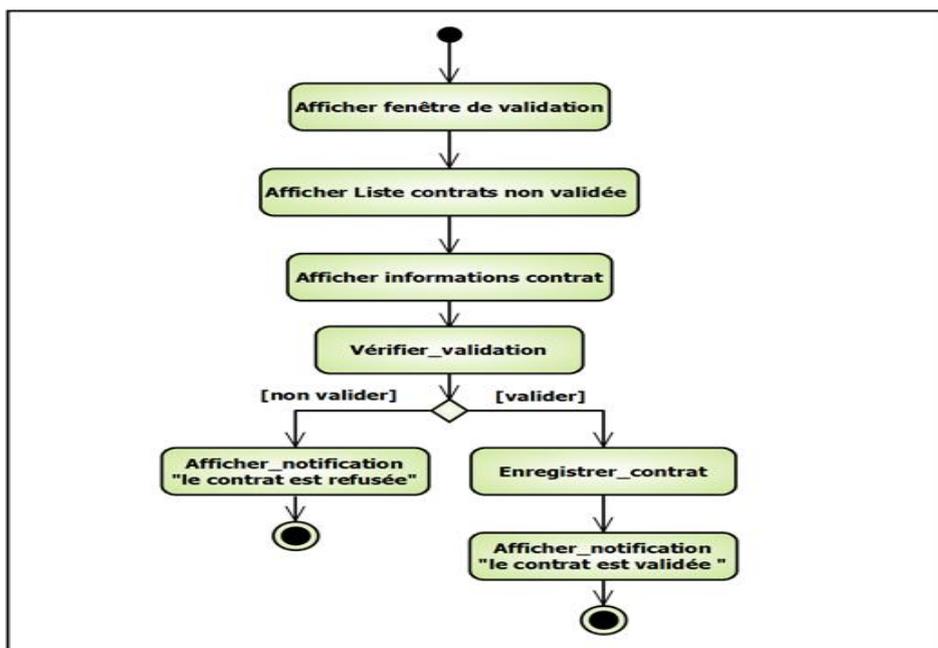


Figure 41: Diagramme d'activité «Validation des contrats».

• Diagramme de séquence : Validation des contrats

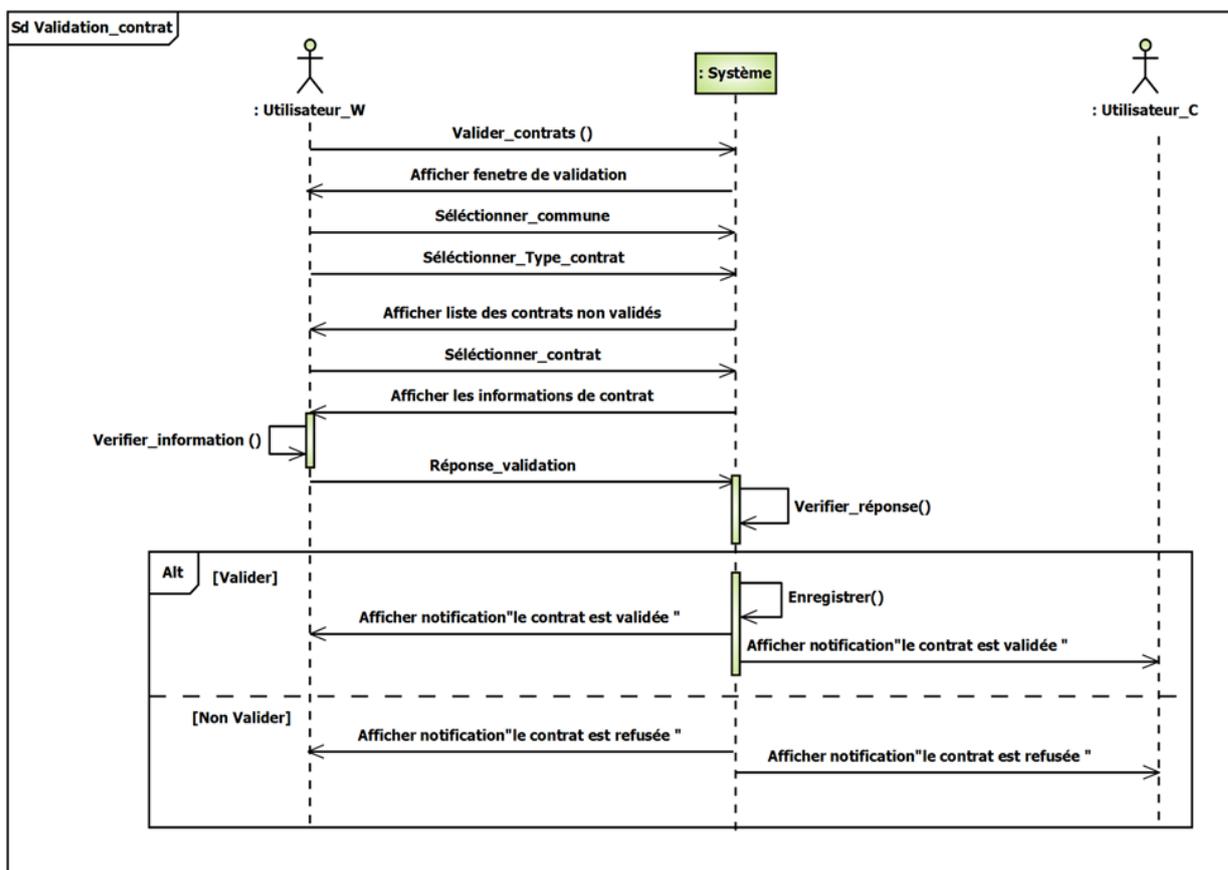


Figure 42: Diagramme de séquence «Valider des contrats».

### 3.2.2. Recherche Multicritères wilaya

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_Multicritères\_wilaya».

Cas d'utilisation	Recherche_Multicritères_wilaya
<b>Acteur</b>	Utilisateur_W
<b>But</b>	Consiste à rechercher un patrimoine ou rechercher un contrat.
<b>Pré condition</b>	L'utilisateur doit être authentifié.
<b>Post condition</b>	/
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la recherche.</li> <li>2. Le système présente les choix de la recherche.</li> <li>3. L'utilisateur choisi de l'action suivante : <ul style="list-style-type: none"> <li>- Recherche patrimoine.</li> <li>- Recherche contrat.</li> <li>- Recherche propriétaire.</li> </ul> </li> </ol>

Tableau 36: Description textuelle du cas d'utilisation «Recherche\_Multicritères\_wilaya».

#### 3.2.2.1. Recherche patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_patrimoine» pour recherche un patrimoine.

Cas d'utilisation	Recherche_patrimoine
<b>Acteurs</b>	Utilisateur_W
<b>But</b>	Obtenir les informations d'un patrimoine.
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Post condition</b>	Affiche les informations d'un patrimoine.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la recherche d'un patrimoine.</li> <li>2. Le système affiche le niveau de recherche (par commune, par daïra, par wilaya).</li> <li>3. L'utilisateur choisi un niveau de recherche.</li> <li>4. Le système affiche le formulaire qui contient les critères de</li> </ol>

	<p>recherche.</p> <p>5. L'utilisateur choisit un ou plusieurs critères de recherche</p> <p>5.1. Recherche par (type de patrimoine, catégorie de patrimoine, date d'enregistrement, type de contrat).</p> <ul style="list-style-type: none"> <li>- Le système affiche la liste des patrimoines selon les critères.</li> <li>- L'utilisateur sélectionne un patrimoine.</li> <li>- Le système affiche les informations de patrimoine sélectionné.</li> </ul> <p>5.2. Recherche par code de patrimoine.</p> <ul style="list-style-type: none"> <li>- L'utilisateur saisit le code de patrimoine.</li> <li>- Le système affiche les informations de ce patrimoine.</li> </ul>
<b>Scénario alternatif</b>	<p>1. Le cas où le patrimoine n'existe pas, le système notifie la non existence, et repris de l'enchaînement du scénario nominal au point 4.</p>

**Tableau 37: Description textuelle du cas d'utilisation «Recherche\_patrimoine\_wilaya».**

### 3.2.2.2. Recherche contrat

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_contrat » pour rechercher un contrat.

Cas d'utilisation Recherche_contrat	
<b>Acteurs</b>	Utilisateur_W
<b>But</b>	Obtenir les informations d'un contrat.
<b>Pré condition</b>	L'utilisateur est authentifié.
<b>Post condition</b>	Affiche les informations d'un contrat.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la recherche d'un contrat.</li> <li>2. Le système affiche le niveau de recherche (par commune, par daïra, par wilaya).</li> <li>3. L'utilisateur choisit un niveau de recherche.</li> <li>4. Le système affiche le formulaire qui contient les critères de</li> </ol>

	<p>recherche.</p> <p>5. L'utilisateur choisi un ou plusieurs critères de recherche.</p> <p>5.1. Recherche par (type de patrimoine, la catégorie de patrimoine, type de contrat).</p> <ul style="list-style-type: none"> <li>- Le système afficher la liste des contrats selon les critères.</li> <li>- L'utilisateur sélectionne un contrat.</li> <li>- Le système affiche les informations de contrat.</li> </ul> <p>5.2. Recherche par code de contrat.</p> <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de contrat.</li> <li>- Le système afficher les informations de ce contrat.</li> </ul> <p>5.3. Recherche par le code de propriétaire.</p> <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de propriétaire.</li> <li>- Le système afficher la liste des contrats de ce propriétaire.</li> <li>- L'utilisateur sélectionne un contrat.</li> <li>- Le système affiche les informations de contrat.</li> </ul>
<p><b>Scénario alternatif</b></p>	<ol style="list-style-type: none"> <li>1. Le cas où le contrat n'existe pas, le système notifie le non existence, et reprise de l'enchaînement du scénario nominal au point 4.</li> <li>2. Le cas où le propriétaire n'existe pas, le système notifie la non existence, et reprise nominal au point 4.</li> </ol>

**Tableau 38: Description textuelle du cas d'utilisation «Recherche\_contrat\_wilaya».**

### 3.2.2.3. Recherche propriétaire

Ce cas commence lorsque l'utilisateur accède au système et choisit «Recherche\_propriétaire » pour rechercher un propriétaire.

Cas d'utilisation	Recherche_propriétaire
<b>Acteurs</b>	Utilisateur_W
<b>But</b>	Obtenir les informations d'un propriétaire.
<b>Pré condition</b>	- L'utilisateur est authentifié.
<b>Post condition</b>	Affiche les informations d'un propriétaire.
<b>Scénario nominal</b>	1. L'utilisateur demande au système la recherche d'un

	<p>propriétaire.</p> <ol style="list-style-type: none"> <li>2. Le système affiche le niveau de recherche (par commune, par daïra, par wilaya).</li> <li>3. L'utilisateur choisit un niveau de recherche.</li> <li>4. Le système affiche le formulaire qui contient les critères de recherche (type de contrat, code de propriétaire).</li> <li>5. L'utilisateur choisi un ou plusieurs critères de recherche.</li> </ol> <ol style="list-style-type: none"> <li>5.1. Recherche par type de contrat. <ul style="list-style-type: none"> <li>- L'utilisateur sélectionne un type de contrat.</li> <li>- système afficher la liste des propriétaires selon le type de contrat.</li> <li>- L'utilisateur sélectionne un propriétaire.</li> <li>- Le système affiche les informations de propriétaire.</li> </ul> </li> <li>5.2. Recherche par code de propriétaire. <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de propriétaire.</li> <li>- Le système afficher les informations de ce propriétaire.</li> </ul> </li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le propriétaire n'existe pas, le système notifie le non existence, et reprise de l'enchaînement du scénario nominal au point 4.</li> </ol>

**Tableau 39: Description textuelle du cas d'utilisation «Recherche\_propriétaire».**

• Diagramme d'activité : Recherche Multicritères wilaya

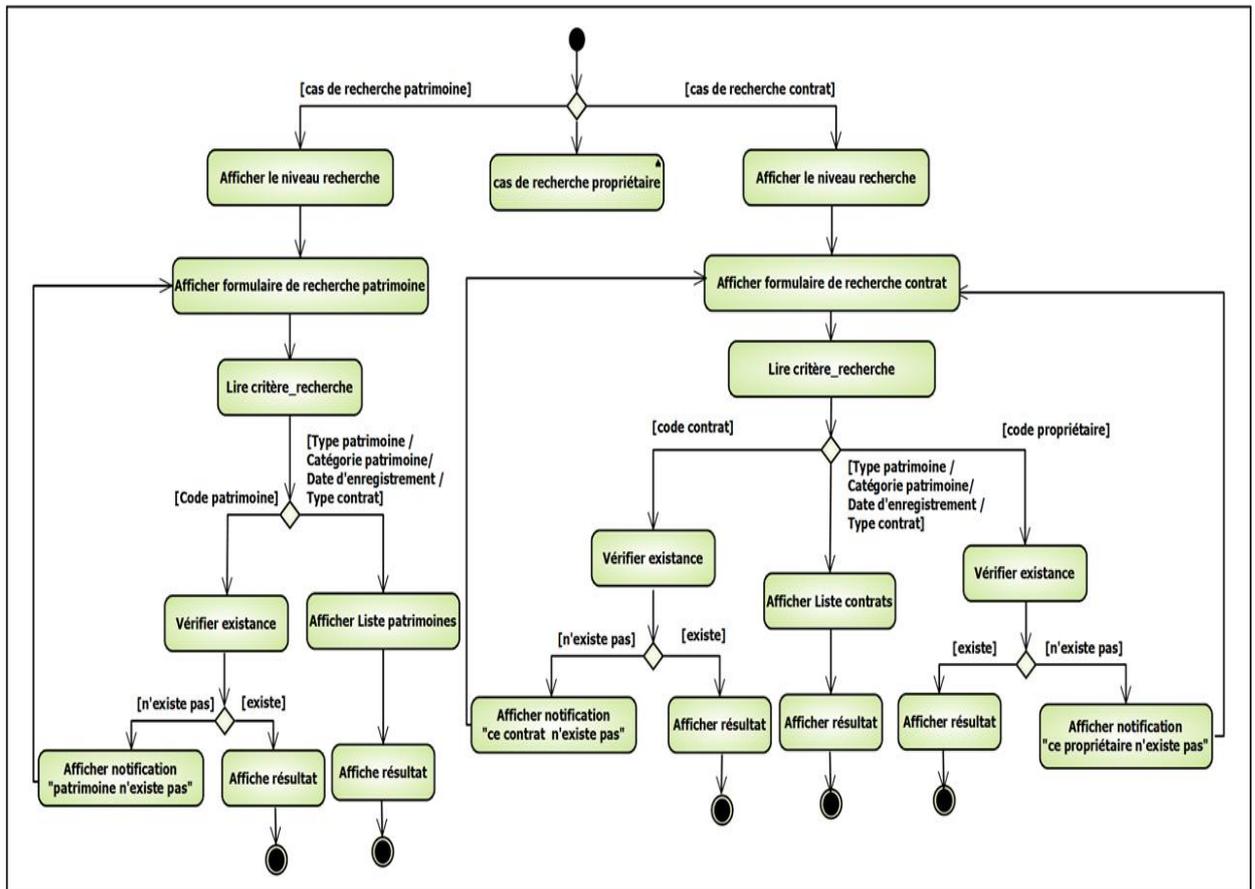


Figure 43: Diagramme d'activité «Recherche\_Multicritères\_wilaya».

• Diagramme d'activité : Recherche Multicritères (recherche propriétaire)

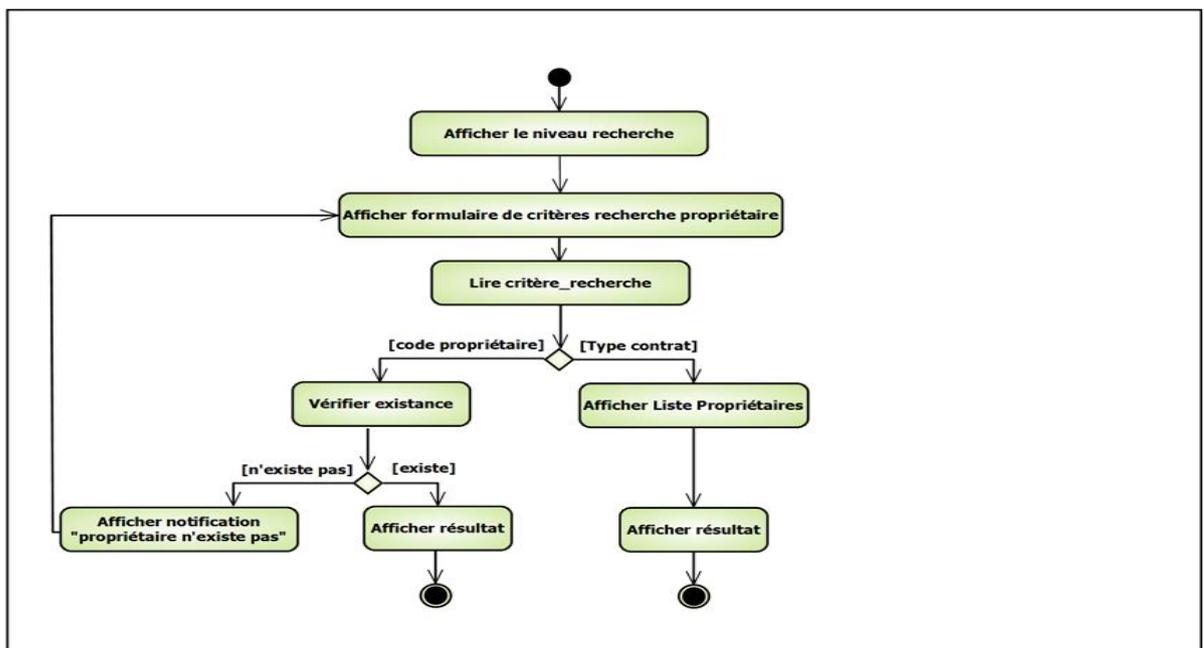


Figure 44: Diagramme d'activité «Recherche\_propriétaire».

• Diagramme de séquence : Recherche Multicritères wilaya

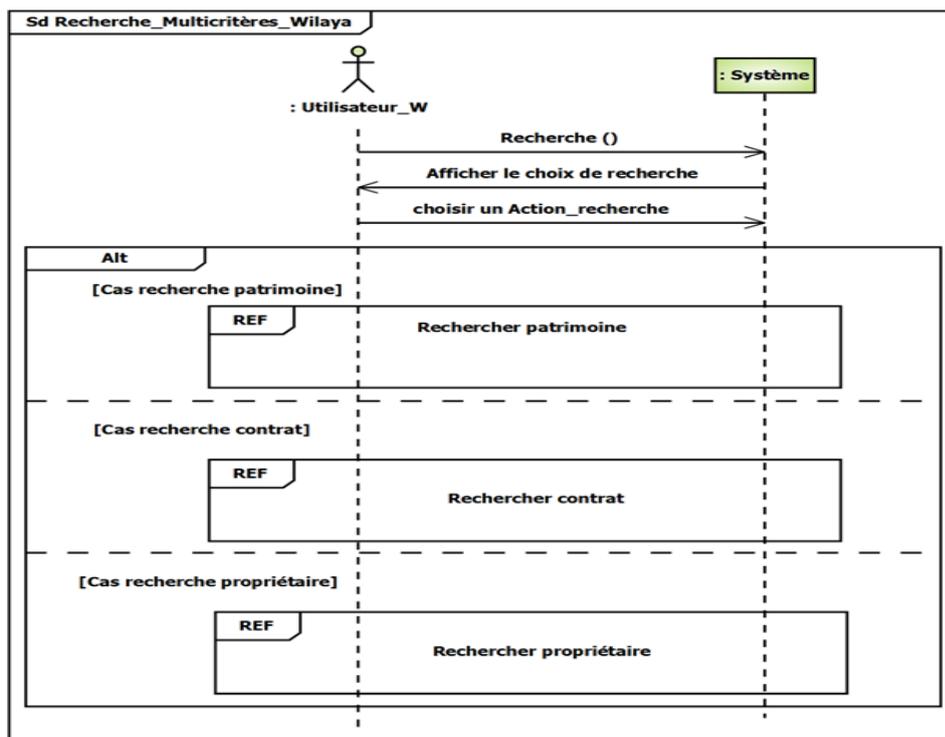


Figure 45: Diagramme de séquence «Recherche\_Multicritères\_wilaya».

• Diagramme de séquence : Rechercher patrimoine wilaya

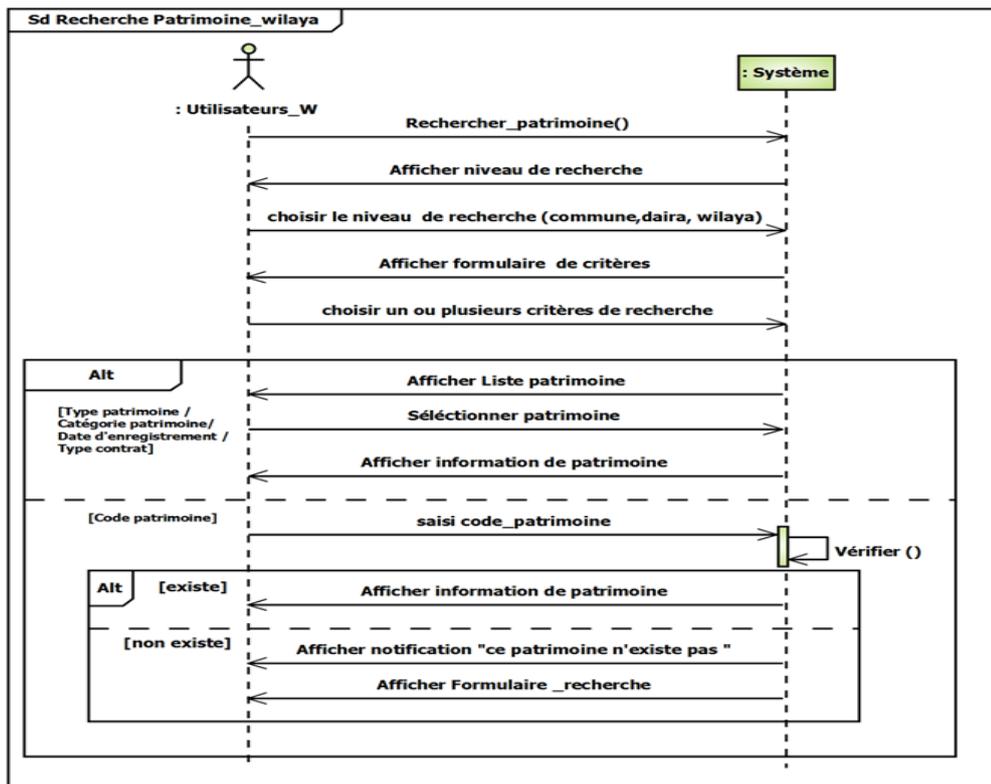


Figure 46: Diagramme de séquence «Rechercher\_patrimoine wilaya».

• Diagramme de séquence : Rechercher contrat wilaya

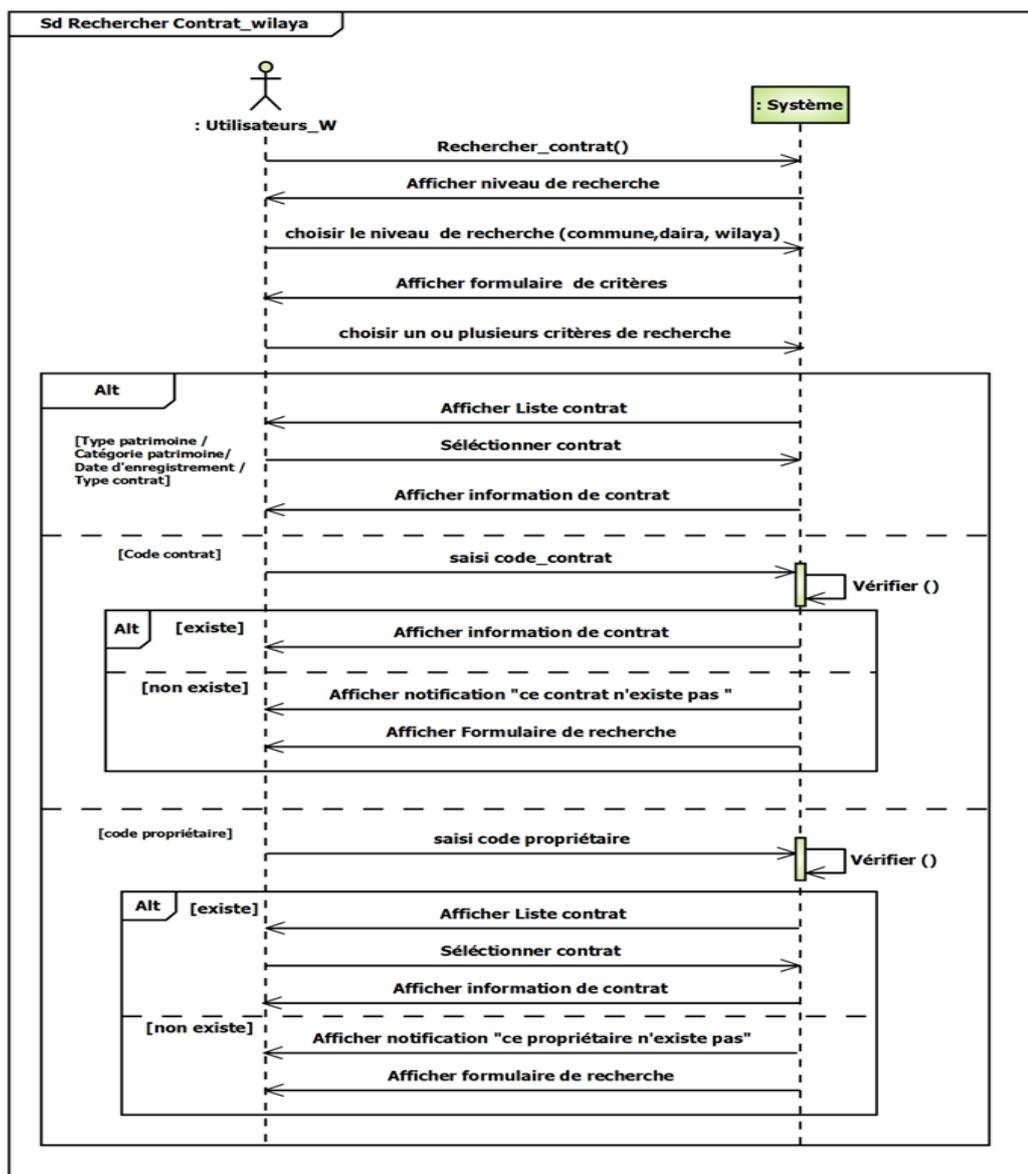


Figure 47: Diagramme de séquence «Recherche\_contrat wilaya».

• Diagramme de séquence : Recherche propriétaire des communes

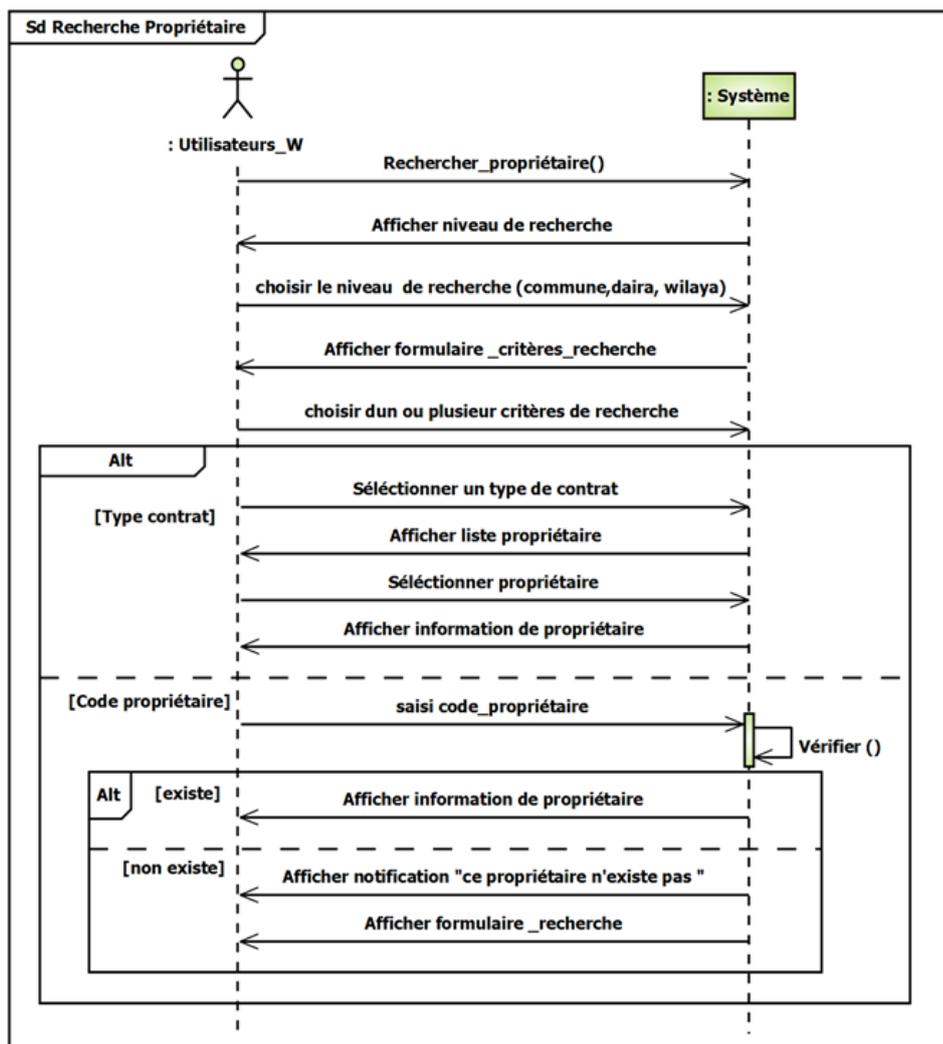


Figure 48: Diagramme de séquence «Recherche propriétaire».

3.2.3. Statistique wilaya

Ce cas commence lorsque l'utilisateur accède au système et choisit «Statistique\_wilaya».

Cas d'utilisation	Statistique_wilaya
<b>Acteur</b>	Utilisateur_W
<b>But</b>	Consiste à calculer les statistique d'un patrimoine ou un contrat.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur doit être authentifié.</li> <li>- Il existe au moins un daïra et une commune et un patrimoine et un contrat dans la base de données.</li> </ul>
<b>Post condition</b>	/

<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système le calcul des statistiques.</li> <li>2. Le système présente les actions de calcul des statistiques.</li> <li>3. L'utilisateur choisit de l'action suivante : <ul style="list-style-type: none"> <li>- Statistique patrimoine.</li> <li>- Statistique contrat.</li> </ul> </li> </ol>
-------------------------	---

**Tableau 40: Description textuelle du cas d'utilisation « statistique wilaya ».**

### 3.2.3.1. Statistique patrimoine

Ce cas commence lorsque l'utilisateur accède au système et choisit «Statistique\_patrimoine» pour calculer les statistiques d'un patrimoine.

Cas d'utilisation	Statistique_patrimoine
<b>Acteurs</b>	Utilisateur_W
<b>But</b>	Obtenir les statistiques d'un patrimoine.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur est authentifié.</li> <li>- Il existe au moins un daïra et une commune et un patrimoine et un contrat dans la base de données.</li> </ul>
<b>Post condition</b>	Affiche les statistiques d'un patrimoine.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système le calcul des statistiques d'un patrimoine.</li> <li>2. Le système affiche le niveau de statistique (par commune, par daïra, par wilaya).</li> <li>3. L'utilisateur choisit un niveau de statistique.</li> <li>4. Le système affiche le formulaire qui contient les critères de statistique (type de patrimoine, catégorie de patrimoine, code de patrimoine, type de contrat, date d'enregistrement de patrimoine, revenus de patrimoine).</li> <li>5. L'utilisateur choisit un ou plusieurs critères de statistique <ol style="list-style-type: none"> <li>5.1. Calcule par le type de patrimoine et/ou la catégorie de patrimoine. <ul style="list-style-type: none"> <li>- L'utilisateur sélectionner un type et /ou une catégorie.</li> <li>- Le système fait le calcul et affiche le résultat de calcul.</li> </ul> </li> </ol> </li> </ol>

	<p>5.2. Calcule par la nature de patrimoine.</p> <ul style="list-style-type: none"> <li>- L'utilisateur saisi la nature de patrimoine.</li> <li>- Le système fait le calcul et affiche le résultat de calcul.</li> </ul>
<b>Scénario alternatif</b>	<p>1. Le cas où le patrimoine n'existe pas, le système notifie le non existence, et reprise de l'enchaînement du scénario nominal au point 4.</p>

**Tableau 41: Description textuelle du cas d'utilisation « statistique patrimoine».**

### 3.2.3.2. Statistique contrats

Ce cas commence lorsque l'utilisateur accède au système et choisit «Statistique\_contrat» pour calculer les Statistique des contrats.

Cas d'utilisation	Statistique_contrat
<b>Acteurs</b>	Utilisateur_W
<b>But</b>	Obtenir les statistiques des contrats.
<b>Pré condition</b>	<ul style="list-style-type: none"> <li>- L'utilisateur est authentifié.</li> <li>- Il existe au moins un contrat et un daïra et une commune dans la base de données.</li> </ul>
<b>Post condition</b>	Affiche les statistiques d'un contrat.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur demande au système la statistique d'un contrat.</li> <li>2. Le système affiche le niveau de statistique (par commune, par daïra, par wilaya).</li> <li>3. L'utilisateur choisi un niveau de statistique (par commune, par daïra, par wilaya).</li> <li>4. Le système affiche le formulaire qui contient les critères de statistique (type du patrimoine, catégorie du patrimoine, code de patrimoine, type de contrat, date d'enregistrement de patrimoine).</li> <li>5. L'utilisateur choisit un ou plusieurs critères pour le calcul.</li> </ol> <p>5.1. Calcule par type de contrat.</p> <ul style="list-style-type: none"> <li>- L'utilisateur choisit le type de contrat.</li> </ul>

	<ul style="list-style-type: none"> <li>- Le système calculer et afficher le résultat de calcul.</li> </ul> <p>5.2. Calcule par code de propriétaire.</p> <ul style="list-style-type: none"> <li>- L'utilisateur saisi le code de propriétaire.</li> <li>- Le système calculer et afficher le résultat de calcul</li> </ul>
<b>Scénario alternatif</b>	<p>1. Le cas où le propriétaire n'existe pas, le système notifie le non existence, et reprise de l'enchaînement du scénario nominal au point 4.</p>

Tableau 42: Description textuelle du cas d'utilisation « statistique contrat».

• Diagramme d'activité: Statistique wilaya

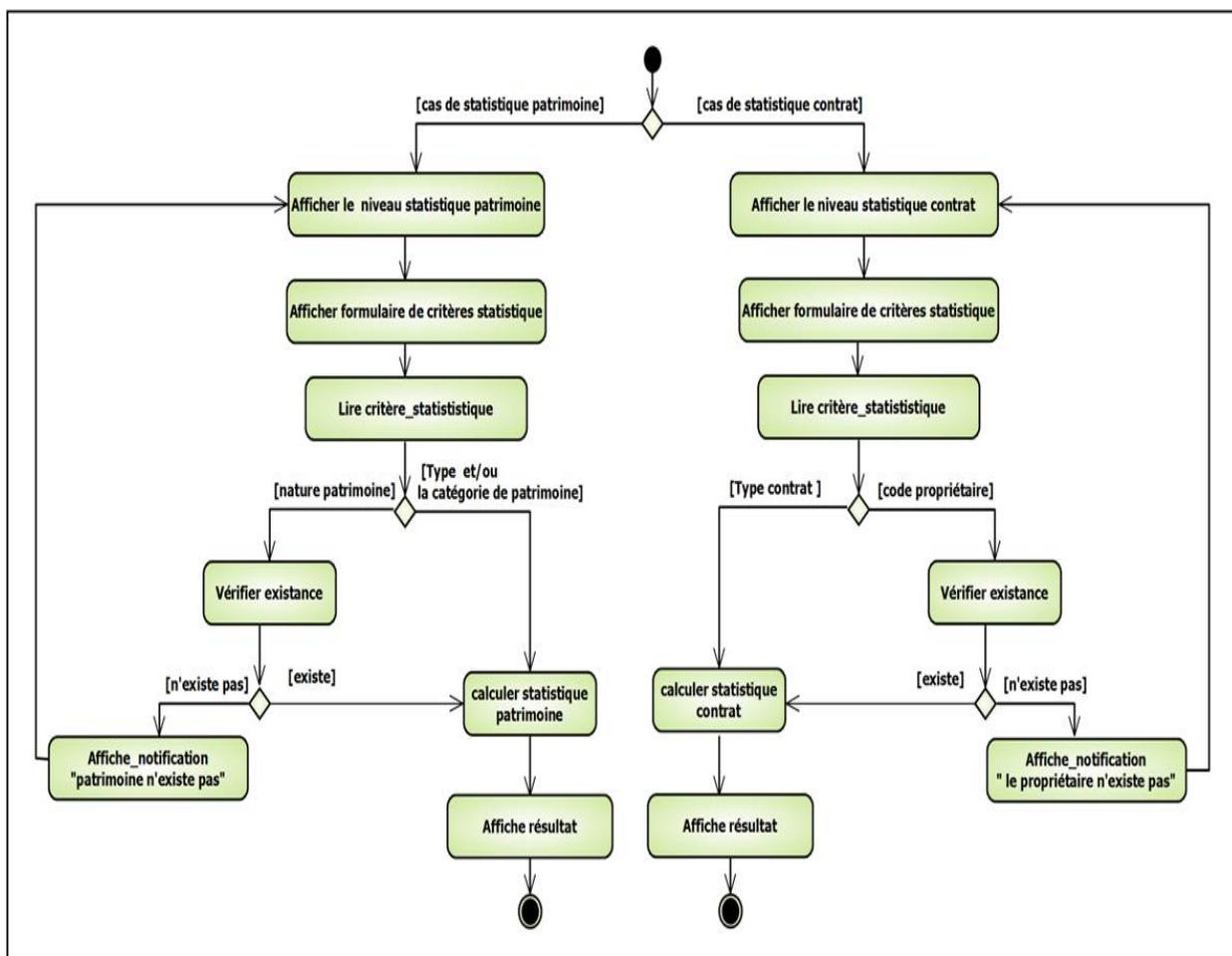


Figure 49: Diagramme d'activité «statistique\_wilaya».

- Diagramme de séquence : Statistique wilaya

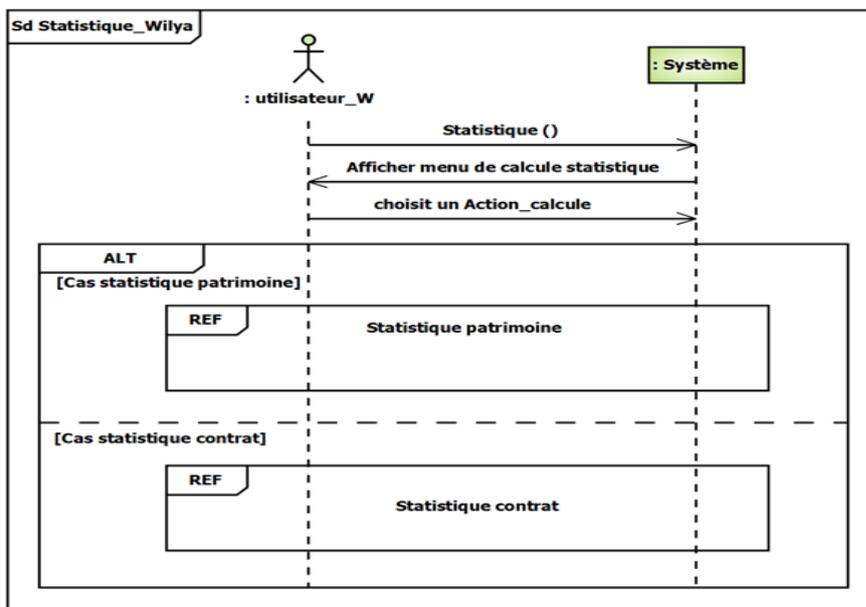


Figure 50: Diagramme de séquence «statistique\_wilaya».

- Diagramme de séquence : Statistique patrimoine wilaya

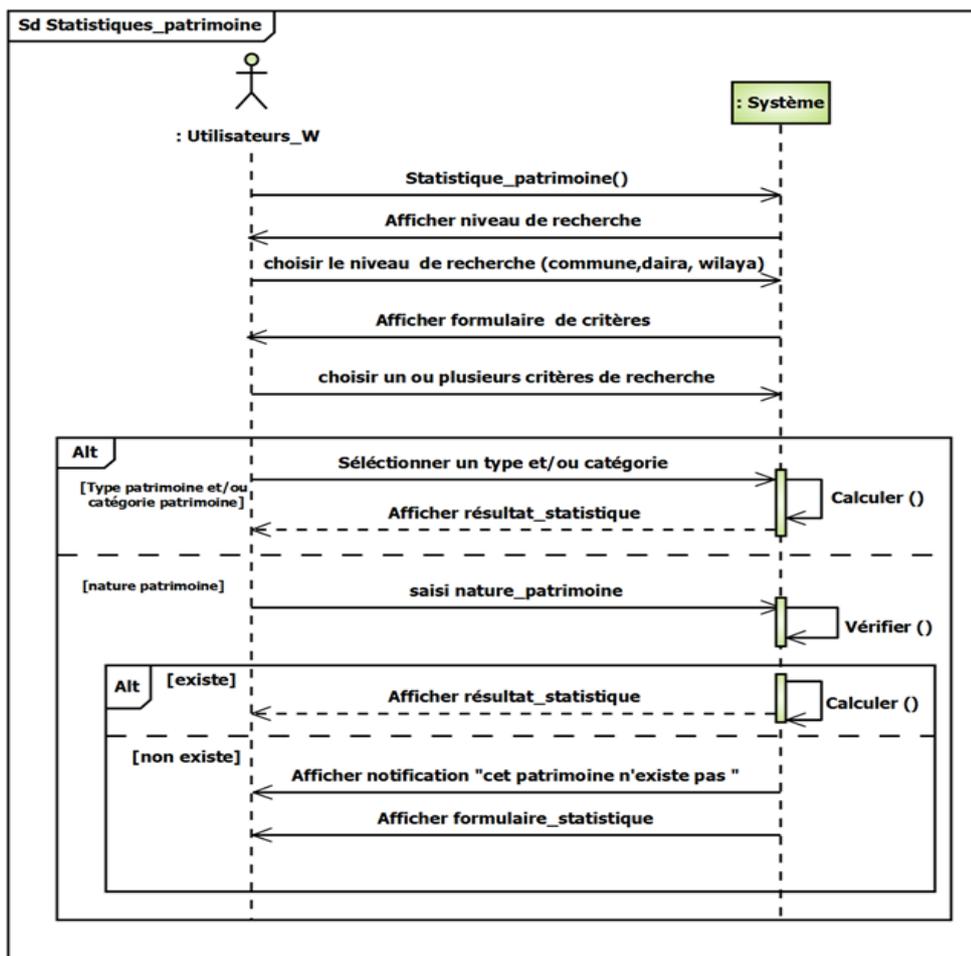


Figure 51: Diagramme de séquence «statistique\_Patrimoine».

- **Diagramme de séquence : Statistique contrat wilaya**

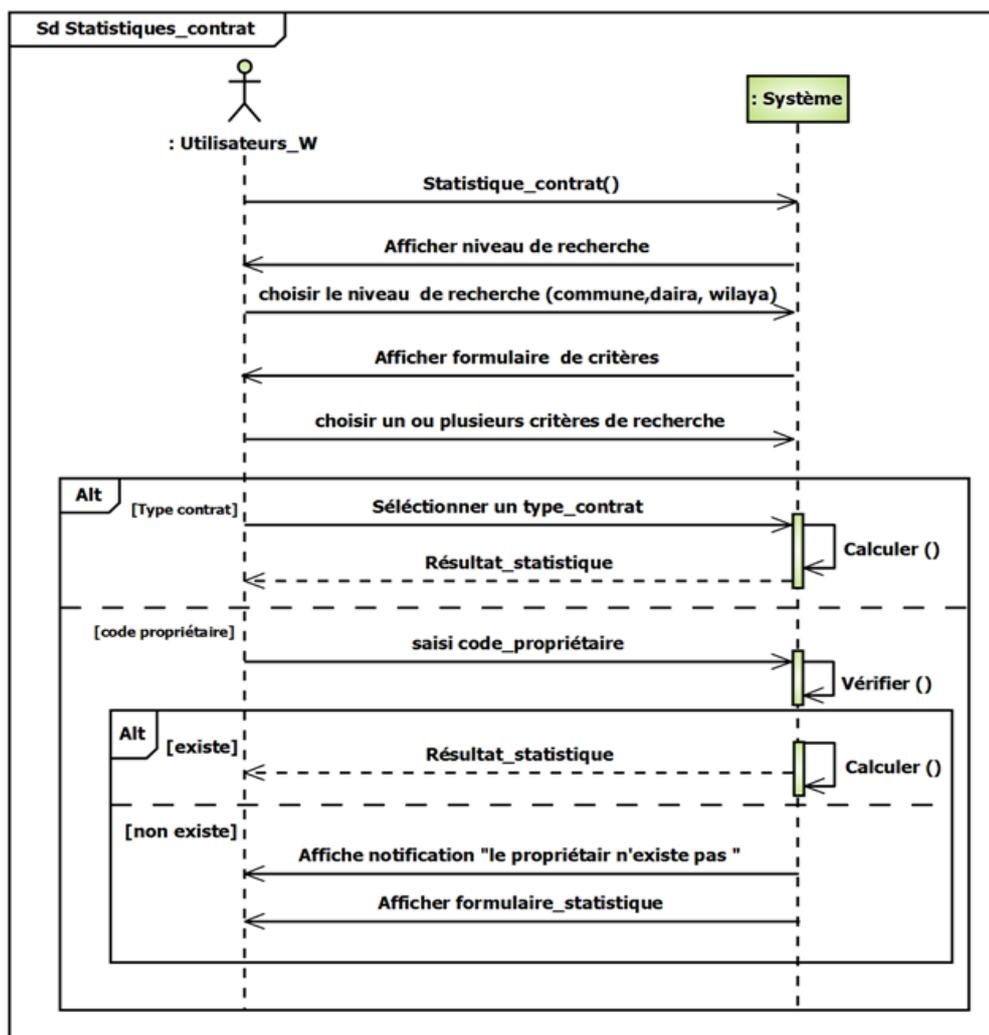


Figure 52: Diagramme de séquence «statistique\_contrat».

#### 4. Les règles de gestion

- La daïra est composée de plusieurs communes.
- La commune est associée à une seule daïra.
- La commune possède plusieurs patrimoines.
- Un patrimoine est associé à une commune.
- Un patrimoine possède plusieurs contrats.
- Un contrat associé à un seul patrimoine.
- Un propriétaire possède plusieurs contrats.
- Un contrat associé à un seul propriétaire.

## 5. Le modèle de domaine

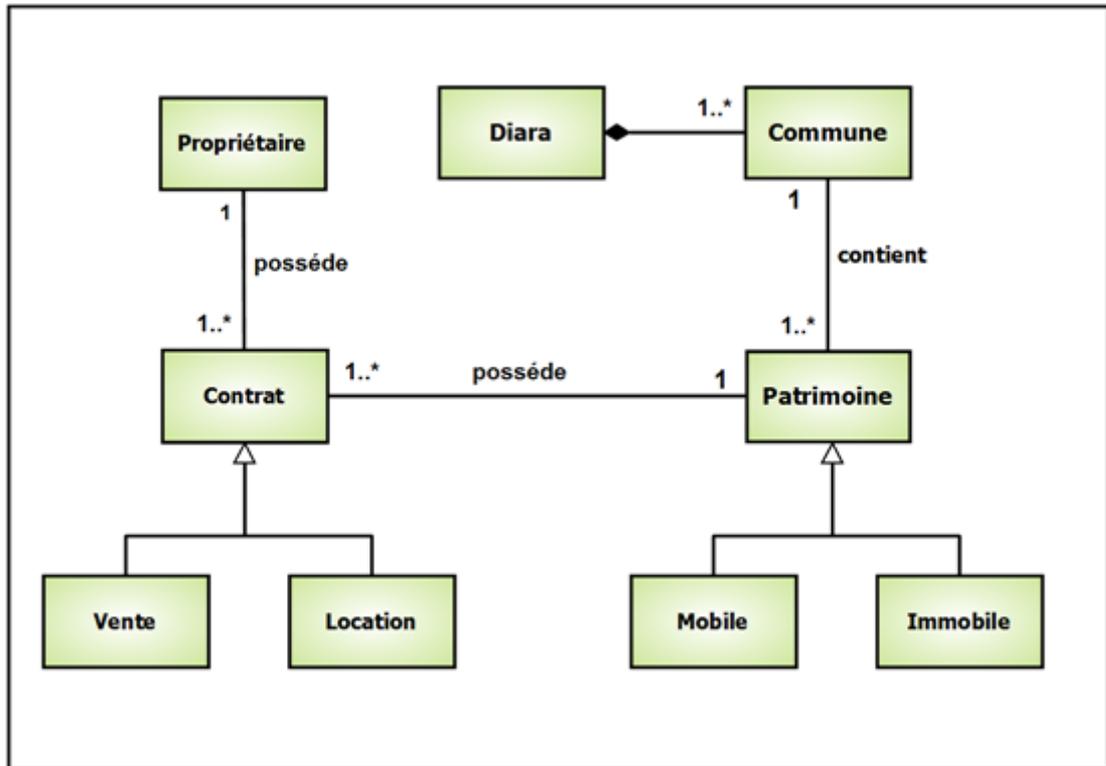


Figure 53: le modèle de domaine.

## Troisième partie : Capture des besoins technique

On va s'intéresser à la branche droite du cycle en Y qui est « la capture des besoins techniques» en couvrant avec celle des besoins fonctionnels les contraintes qui ne traitent pas la description applicative, C'est la phase qui sert de compléter la capture des besoins fonctionnels.

Cette étape est très importante lorsque les architectes ont obtenu suffisamment d'informations sur l'environnement de travail ainsi que l'architecture globale utilisée pour notre système et les prés requis techniques. La capture des besoins techniques se présente comme suit:

- Capture des spécifications logicielles.
- Capture des spécifications liées à la configuration matérielle. [4]

### 1. Spécification technique du point de vue matériel

Les choix des prés requis techniques déjà mentionnés dans l'étude préliminaire, lors de l'expression des besoins opérationnels, impliquent des contraintes relatives à la configuration du réseau matériel, les performances d'accès aux données ainsi que la sécurité du système, l'intégration des applications, la volumétrie et le mode d'utilisation du système.

#### 1.1. Configuration matérielle

Pour finaliser la réalisation de notre système, nous avons besoin du matériel et des postes de travail suivants. Notez qu'un poste de travail représente un ou plusieurs acteurs pouvant être localisés sur une machine d'un type particulier et remplissant une fonction identifié l'application.

- Caractéristiques techniques du serveur
  - Marque : HP Proliant DL380G5.
  - Intel® Xeon® CPU.
  - 3.25 GO de RAM.
  - 3 Disque de 146 GO (SCSI).
  - Lecteur DVD.
  - Lecteur HP DAT 72 USB.
  - Disque externe de 250 GO.
  - 2 Cartes réseau HP Gigabit.

- Onduleur MGE Evolution S 3000.
- Un réseau haut débit on note que le terme de haut débit (ou large bande) fait référence à des capacités d'accès à internet supérieures à celle de l'accès analogique par modem (typique : 56 kb/s) et à l'accès numérique (typique 1× ou 2× 64 kb/s). pour la wilaya de Mila, la capacité d'accès est 2MB, elle donne pour les grandes communes une capacité d'accès égal à 256 KB/S, et pour les autres communes une capacité d'accès supérieur ou égal à 64KB/S.
- Répartition des postes
  - L'administrateur (serveur): 1 poste.
  - Les clients : nous avons besoin de deux types des clients :
    - ✓ Client commune : nous avons besoin d'un nombre de postes égales aux nombre de commune, à savoir : 32.
    - ✓ Client wilaya : nous avons besoin de 1 ou plusieurs postes dans la wilaya.

## 1.2. Spécification du style d'architecture 2 niveaux

La spécification d'une architecture à composants métier 2 niveaux implique les contraintes sur le modèle d'exploitation. Une solution « client-serveur » 2 niveaux entraîne en effet de faire la répartition des composants d'exploitation comme suit :

- Chaque partie indique une responsabilité technique à la quelle souscrivent les différents composants d'exploitation d'un système :

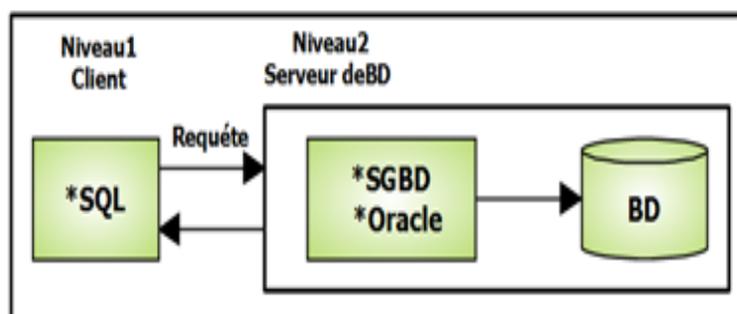


Figure 54: Architecture 2 niveaux de notre système.

## 2. Spécification technique du point de vue logicielle

Une fois que les spécifications technique et architecturale sont exprimées, on peut s'intéresser aux fonctionnalités propres du système technique en procédant à une spécification logicielle. Pour ceci, on propose d'utiliser les cas d'utilisation de

manière différente que pour la spécification fonctionnelle. C'est pourquoi nous avons introduit les concepts :

- **L'exploitant.** est un acteur au sens d'UML, si ce n'est qu'il ne bénéficie que des fonctionnalités techniques du système. Tout système informatique possède au minimum un exploitant qui est «l'utilisateur du système».
- **Un cas d'utilisation technique.** est destiné à l'exploitant. C'est une séquence d'actions produisant une valeur ajoutée opérationnelle ou purement technique. [4]

## 2.1. Les exploitants

Les exploitants de notre système sont la majorité des acteurs de la branche fonctionnelle (utilisateur commune, utilisateur wilaya, administrateur) sont donc utilisateur dans la dimension technique.

## 2.2. Identification des cas d'utilisation techniques

Les cas d'utilisation technique de notre système sont d'abord identifiés en considérant l'attente opérationnelle de chaque exploitant. Nous pouvons recenser les contraintes techniques suivant :

- Le système doit être gérer la distribution des données des patrimoines entre les communes et la wilaya.
- Le système doit garantir l'intégrité et la cohérence des données à chaque mise à jour (Exemple lors des mis a jours des patrimoines, des marques, des utilisateurs).
- Le système doit être capable de certifier la sécurité des données par l'authentification de chaque utilisateur qui veut avoir l'accès au système et la gestion des utilisateurs.
- Le système doit gérer les erreurs pour faciliter la maintenance (Voir figure 55).

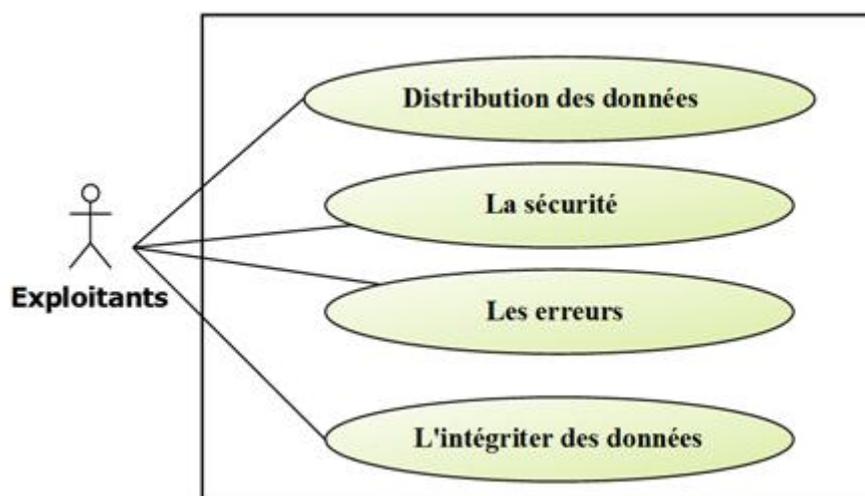


Figure 55: Modèle de spécification logicielle du système (diagramme de cas technique).

### 2.3. Description des cas d'utilisation techniques

Pour des raisons de délais du projet et d'espace dans ce mémoire, nous nous contentons de décrire l'aspect technique «Gérer la sécurité ».

#### 2.3.1. Cas d'utilisation technique : «Gérer la sécurité »

##### 2.3.1.1. S'authentifier

Cas d'utilisation	S'authentifier
<b>Acteur</b>	L'exploitant
<b>But</b>	L'exploitant est authentifié par le système.
<b>Pré condition</b>	L'exploitant est inscrit dans le système.
<b>Post condition</b>	L'exploitant est identifié par le système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système affiche le formulaire d'authentification.</li> <li>2. L'exploitant saisi ses identifiants.</li> <li>3. Le système lit et vérifie la validité du nom et du mot de passe et ouvre le système.</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le nom d'exploitant et/ou mot de passe est erroné, le système affiche une notification « le nom d'exploitant et /ou mot de passe incorrects».</li> </ol>

	2. Reprise de l'enchaînement du scénario nominal au point 1.
<b>Scénario exceptionnel</b>	1. le nom d'exploitant ou mot de passe est erroné pour la 3 <sup>ème</sup> fois il sera déconnecte du système.

Tableau 43: Description textuelle du cas d'utilisation « authentification ».

• Diagramme d'activité : Gérer la sécurité (authentifier)

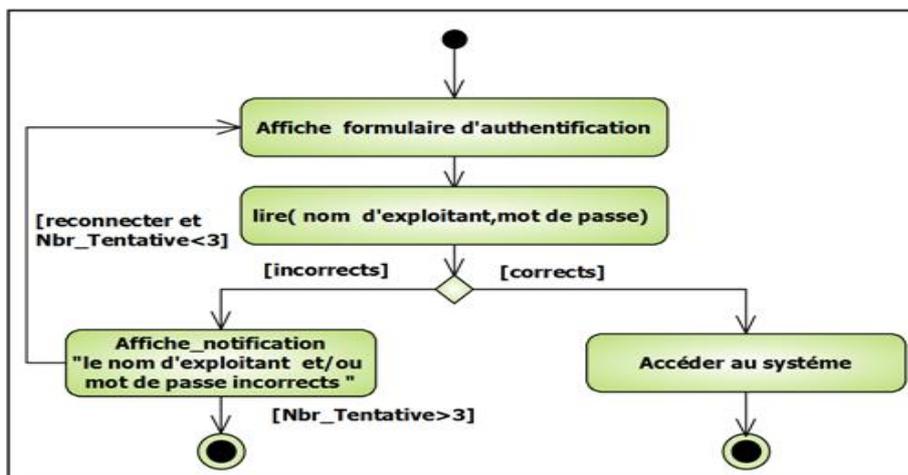


Figure 56: Diagramme d'activité « Gérer la sécurité (s'authentifier)»

• Diagramme de séquence : Gérer la sécurité (authentifier)

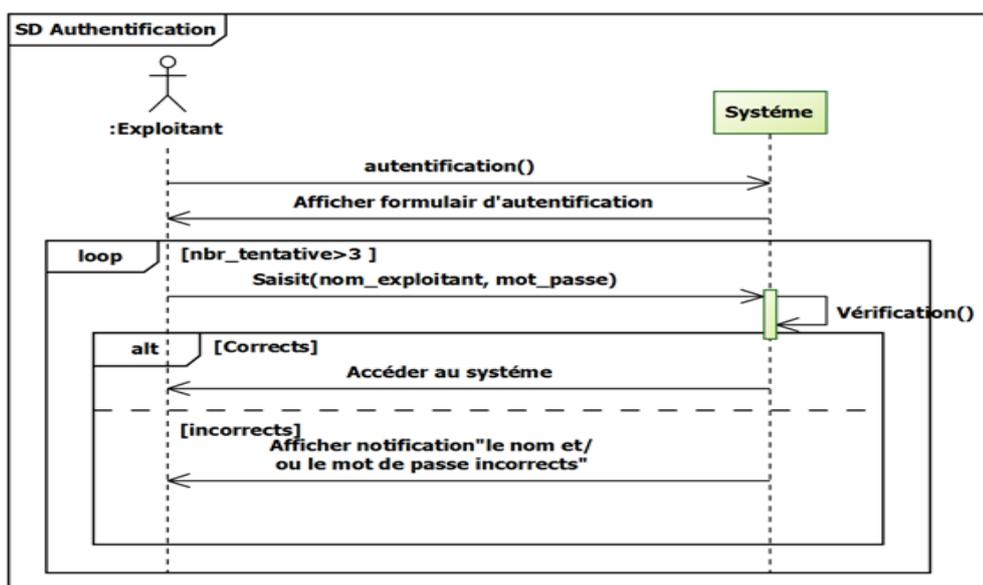


Figure 57: Diagramme de séquence « Gérer la sécurité (s'authentifier)».

### 2.3.1.2. Gestion des utilisateurs

Ce cas commence lorsque l'administrateur accède au système et choisit «Gestion\_utilisateurs ».

Cas d'utilisation	Gestion_utilisateurs
<b>Acteur</b>	Administrateur
<b>But</b>	Consiste à créer, modifier, supprimer ou consulter un utilisateur.
<b>Pré condition</b>	- L'administrateur doit être authentifié.
<b>Post condition</b>	/
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur demande au système la gestion d'utilisateur.</li> <li>2. Le système affiche le menu de la gestion d'utilisateur.</li> <li>3. L'administrateur choisi de l'action suivante : <ul style="list-style-type: none"> <li>- Création d'un utilisateur.</li> <li>- Modification d'un utilisateur.</li> <li>- Suppression d'un utilisateur.</li> <li>- Consultation d'un utilisateur.</li> </ul> </li> </ol>

Tableau 44: Description textuelle du cas d'utilisation «gestion\_utilisateurs ».

#### a. Création d'un utilisateur

Ce cas commence lorsque l'administrateur accède au système et choisit «création\_utilisateur »

Cas d'utilisation	Création_utilisateur
<b>Acteur</b>	Administrateur
<b>But</b>	Création d'un compte utilisateur.
<b>Pré condition</b>	- L'administrateur doit être identifié.
<b>Post condition</b>	La création d'un compte est effectuée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur demande la création d'un compte d'utilisateur.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Le système affiche un formulaire de création.</li> <li>3. L'administrateur remplit le formulaire.</li> <li>4. Le système lit et enregistre les informations du compte d'utilisateur.</li> <li>5. Le système affiche une notification «création avec succès».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le formulaire est mal rempli ou cet utilisateur existe, le système affiche une notification (« utilisateur existe déjà », ou « formulaire est mal rempli »).</li> <li>2. Reprise de l'enchaînement du scénario nominal au point 2.</li> </ol>

**Tableau 45: Description textuelle du cas d'utilisation «Création\_utilisateur».**

### b. Modification d'un utilisateur

Ce cas commence lorsque l'administrateur accède au système et choisit «Modification\_utilisateur»

Cas d'utilisation	Modification_utilisateur
<b>Acteur</b>	Administrateur
<b>But</b>	Modification des informations d'un utilisateur.
<b>Pré condition</b>	- L'administrateur doit être authentifié.
<b>Post condition</b>	La modification de compte est effectuée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur demande la modification d'un compte d'utilisateur.</li> <li>2. Le système affiche les types d'utilisateurs (Utilisateur_C, Utilisateur_W).</li> <li>3. L'administrateur sélectionne un type d'utilisateur.</li> <li>4. Le système affiche la liste des utilisateurs.</li> <li>5. L'administrateur consulte la liste des utilisateurs, et choisit un compte à modifier.</li> <li>6. Le système affiche le formulaire de modification avec les informations de cet utilisateur.</li> <li>7. L'administrateur modifie les informations souhaitées et confirme.</li> </ol>

	<ol style="list-style-type: none"> <li>8. Le système lit et enregistre les nouvelles informations saisies.</li> <li>9. Le système affiche une notification «modification avec succès».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où le formulaire mal remplir, le système affiche une notification « formulaire est mal rempli ».</li> <li>2. Reprise de l'enchaînement du scenario nominal au point 6</li> </ol>

Tableau 46: Description textuelle du cas d'utilisation «Modification\_utilisateur ».

### c. Suppression d'un utilisateur

Ce cas commence lorsque l'administrateur accède au système et choisit «Suppression\_utilisateur»

Cas d'utilisation	Suppression_utilisateur
<b>Acteur</b>	Administrateur
<b>But</b>	Suppression d'un compte utilisateur.
<b>Pré condition</b>	- L'administrateur doit être identifié.
<b>Post condition</b>	La suppression d'un compte est effectuée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'administrateur demande la suppression d'un compte d'utilisateur.</li> <li>2. Le système affiche les types d'utilisateurs (Utilisateur_C, Utilisateur_W).</li> <li>3. L'administrateur sélectionne un type d'utilisateur.</li> <li>4. Le système affiche la liste des utilisateurs selon le type.</li> <li>5. L'administrateur consulte la liste des utilisateurs, et choisit un compte à supprimer.</li> <li>6. Le système affiche le message de confirmation.</li> <li>7. L'administrateur confirme la suppression.</li> <li>8. Le système supprimer le compte sélectionner.</li> <li>9. Le système affiche une notification «suppression avec succès ».</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. Le cas où l'administrateur annuler la suppression, le système ignorer la suppression.</li> </ol>

Tableau 47: Description textuelle du cas d'utilisation «Suppression\_utilisateur ».

• Diagramme d'activité : Gestion\_utilisateurs

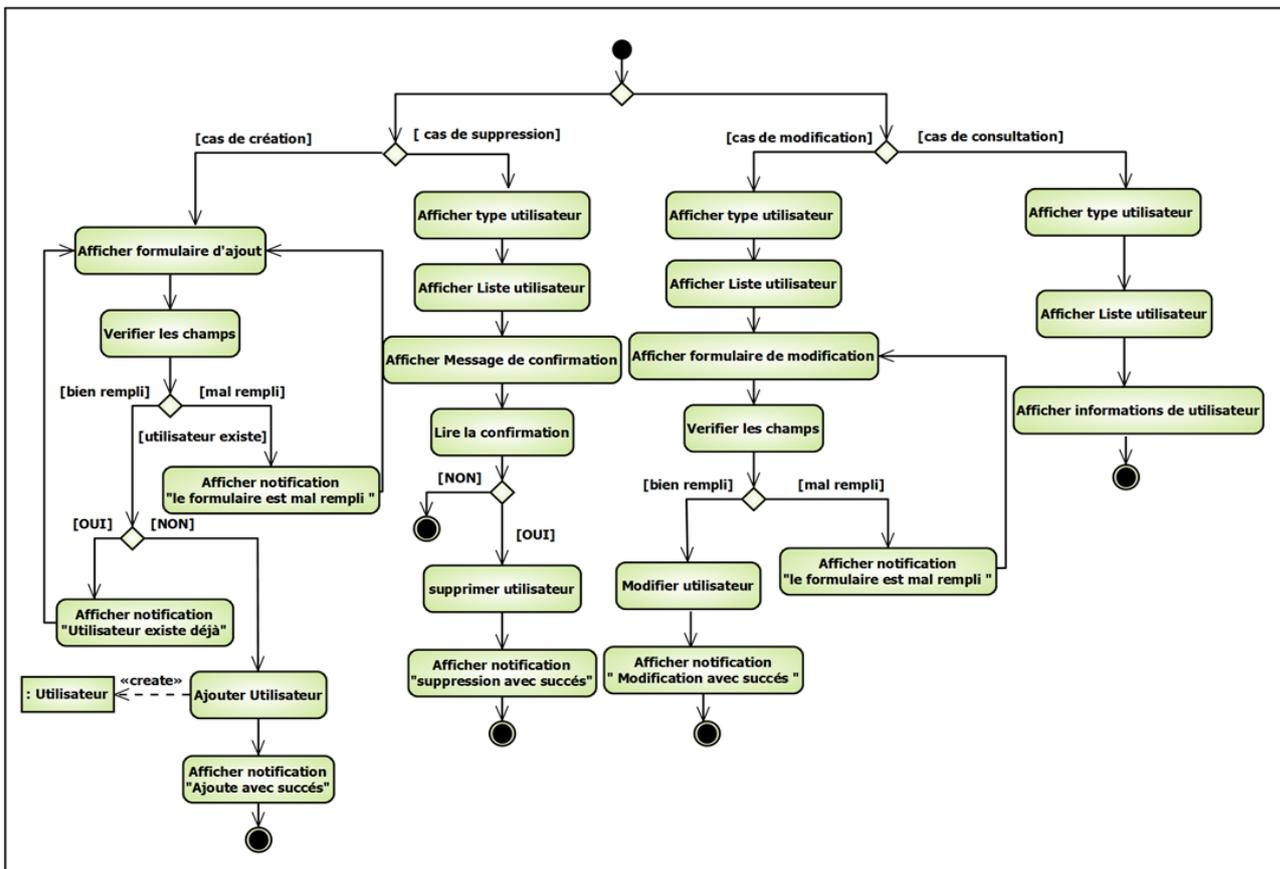


Figure 58: Diagramme d'activité «gestion\_utilisateurs».

2.4. Organisation du modèle de spécification logicielle

L'architecture 2-tiers vise à séparer trois couches logicielles au sein d'une même application.

1. Présentation et traitements sont sur le client.
2. Les données sur le serveur.
3. Contexte Multiutilisateurs avec accès aux données centralisées (Voir figure 59).

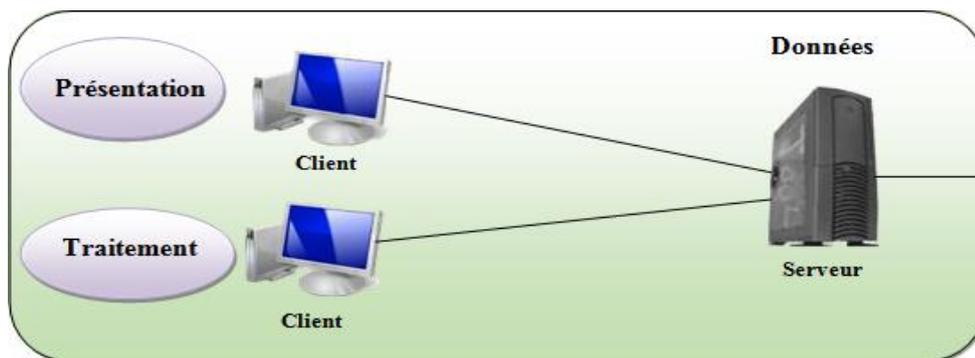


Figure 59: Organisation du modèle de spécification logicielle.

## Quatrième partie : La phase d'analyse

La phase d'analyse est consacrée à l'analyse objet de notre système est-il vient juste après l'étude préliminaire et la capture des besoins fonctionnels et technique. Elle est constituée de 3 activités :

- Le découpage en catégorie.
- Le développement du modèle statique.
- Le développement du modèle dynamique.

### 1. Découpage en catégories

Le découpage en catégories constitue la première activité de l'étape d'analyse. Il se situe sur la branche gauche du cycle en Y et succède à la capture des besoins fonctionnels. Cette phase utilise la notion de package pour définir des catégories de classes d'analyse et découper le modèle UML en blocs logiques les plus indépendants possibles.

Le découpage en catégories se fait en 3 étapes :

#### 1.1.Répartition des classes candidates en catégorie

Le découpage en catégories de notre projet a donné le résultat suivant :

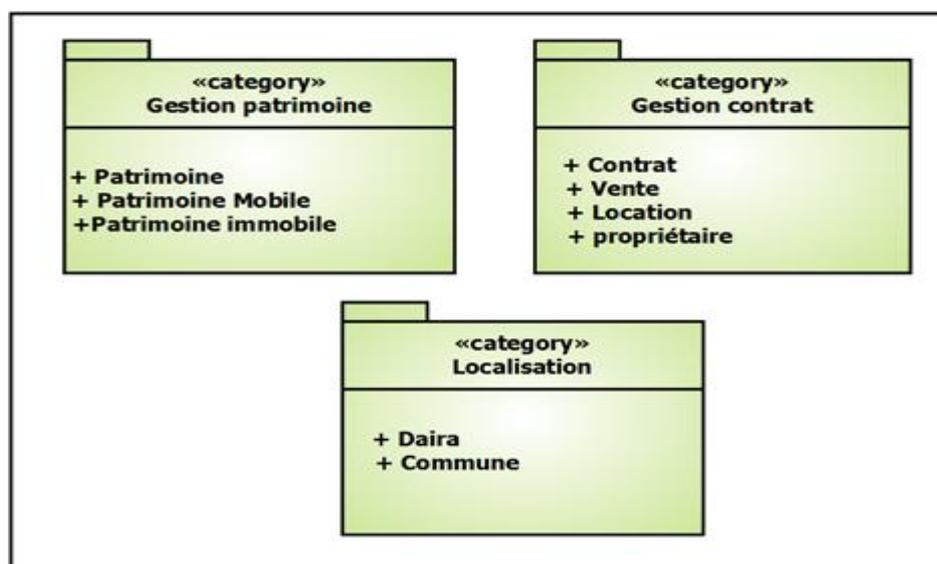


Figure 60: Le découpage en catégories.

### 1.2.Elaboration des diagrammes de classes préliminaires par catégorie

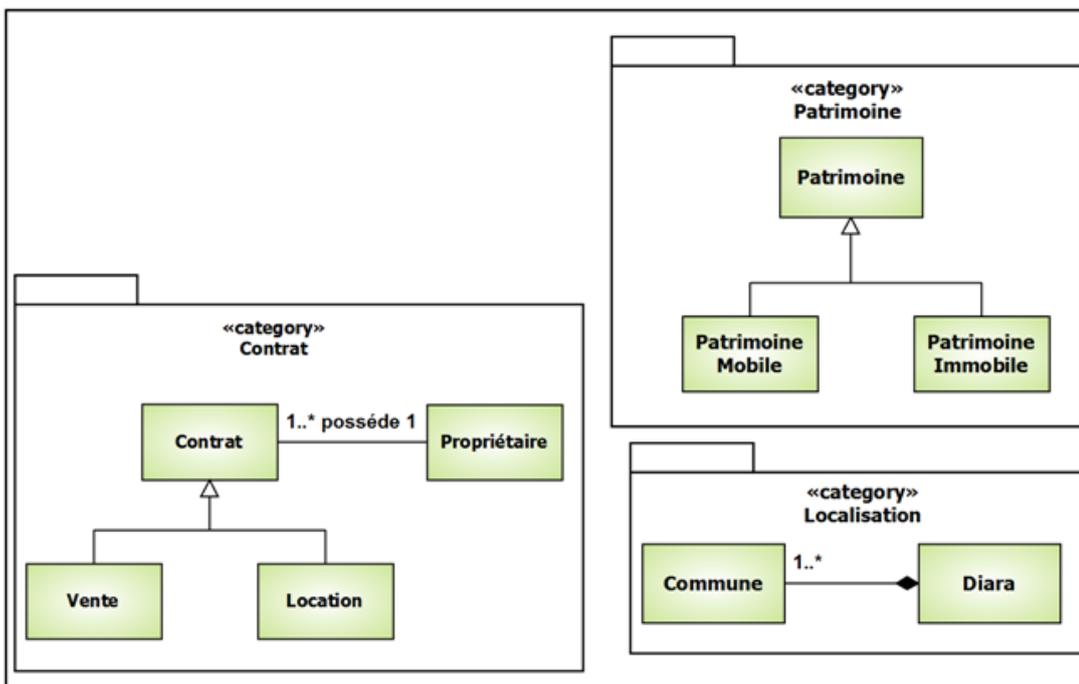


Figure 61: Diagramme de classes préliminaires par catégorie.

### 1.3.Dépendance entre catégorie

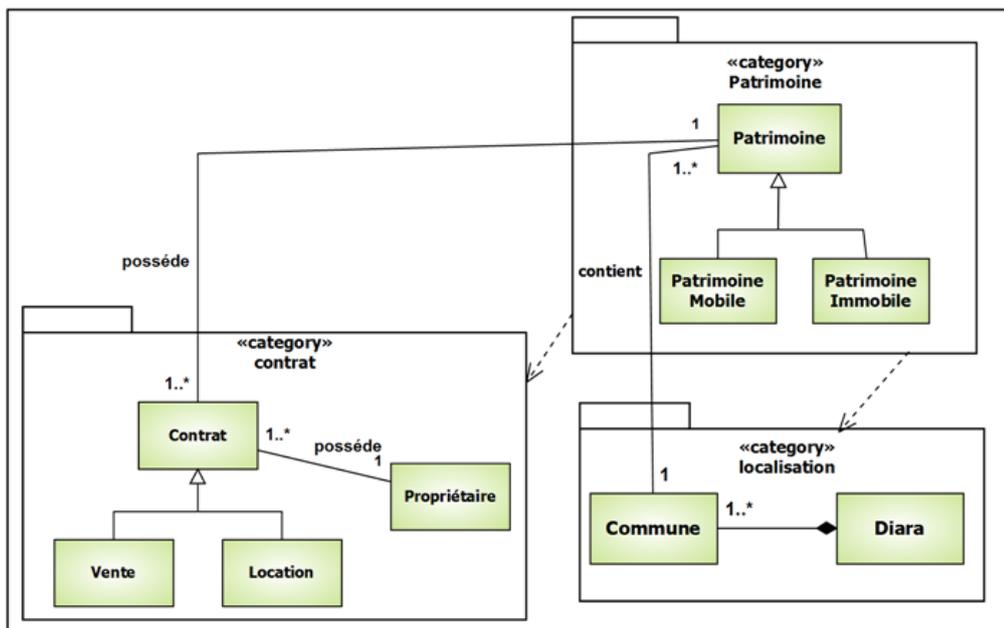


Figure 62: Dépendance entre catégorie.

## 2. Le développement du modèle statique

Le développement du modèle statique constitue la deuxième activité de l'étape d'analyse. Elle se situe sur la branche gauche du cycle en Y et succède au découpage en catégories.

### 2.1. Diagramme de classe pour la catégorie localisation

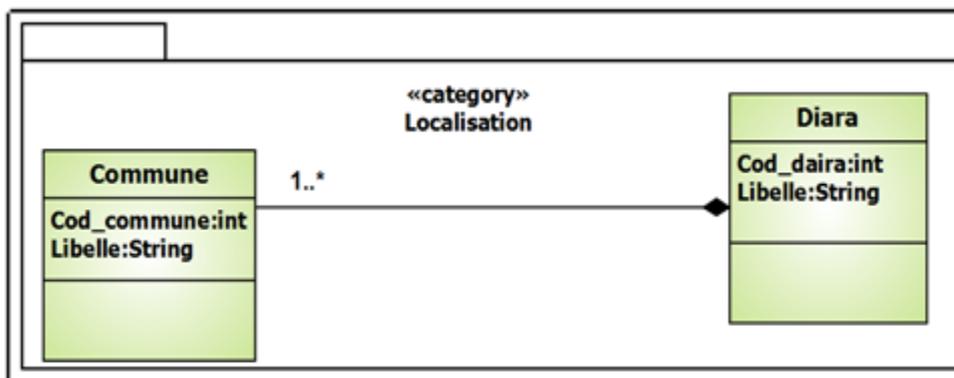


Figure 63: Diagramme de classe pour la catégorie localisation.

### 2.2. Diagramme de classe pour la catégorie gestion patrimoine

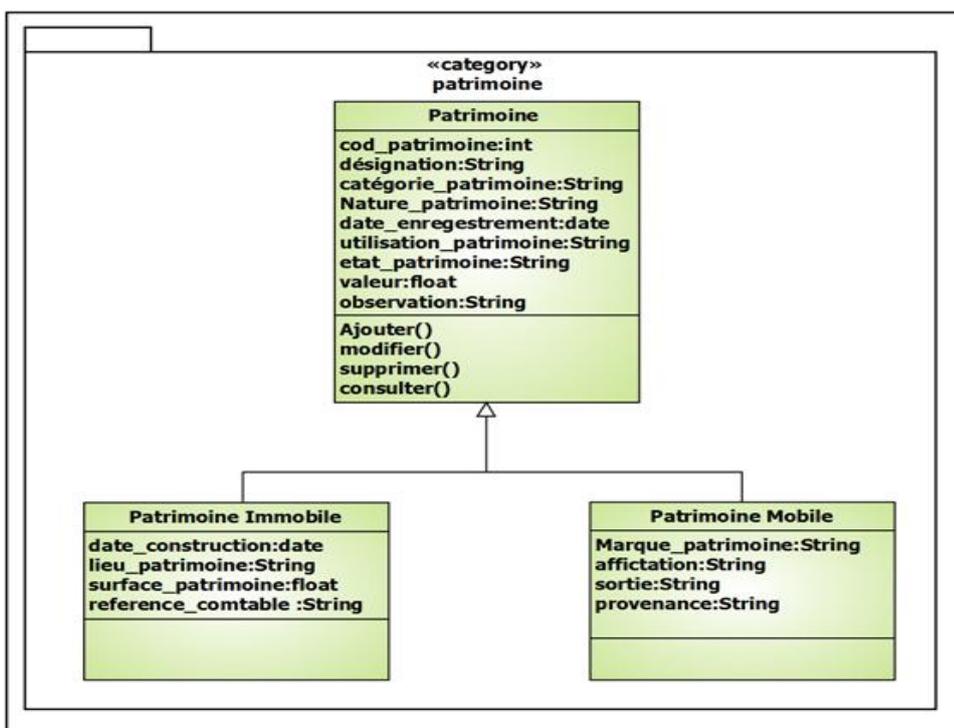


Figure 64: Diagramme de classe pour la catégorie gestion patrimoine.

### 2.3. Diagramme de classe pour la catégorie gestion contrat

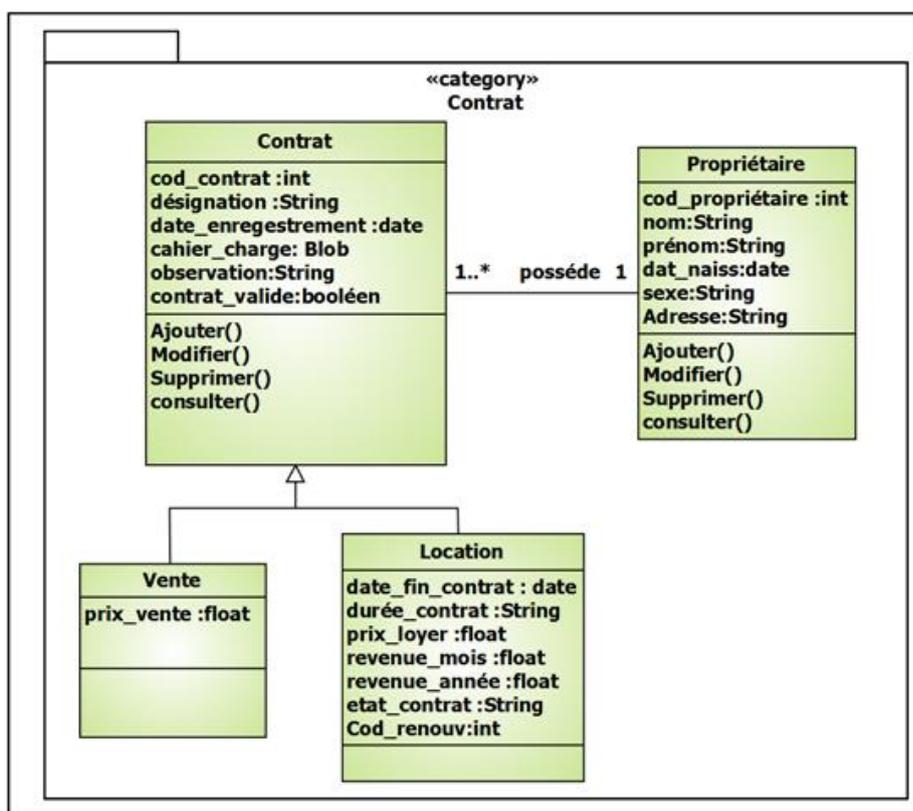


Figure 65: Diagramme de classe pour la catégorie gestion contrat.

## 3. Le développement du modèle dynamique

Le développement du modèle dynamique est la troisième activité de l'étape d'analyse. Cette activité est en relation avec l'activité de modélisation statique. Lors de cette étape, nous décrivons les différentes interactions entre les objets de notre application. En effet, nous avons utilisés le modèle dynamique : le diagramme d'interaction.

### 3.1. Diagrammes de séquences détaillés

Le diagramme de séquence est un diagramme d'interaction entre les objets, qui met l'accent sur le classement des messages par ordre chronologique durant l'exécution du système.

En effet dans cette phase, et après identification des cas d'utilisation, nous représentons à l'aide des diagrammes de séquences les différents cas d'utilisation.

• Diagramme d'interaction de cas d'utilisation ajouter patrimoine

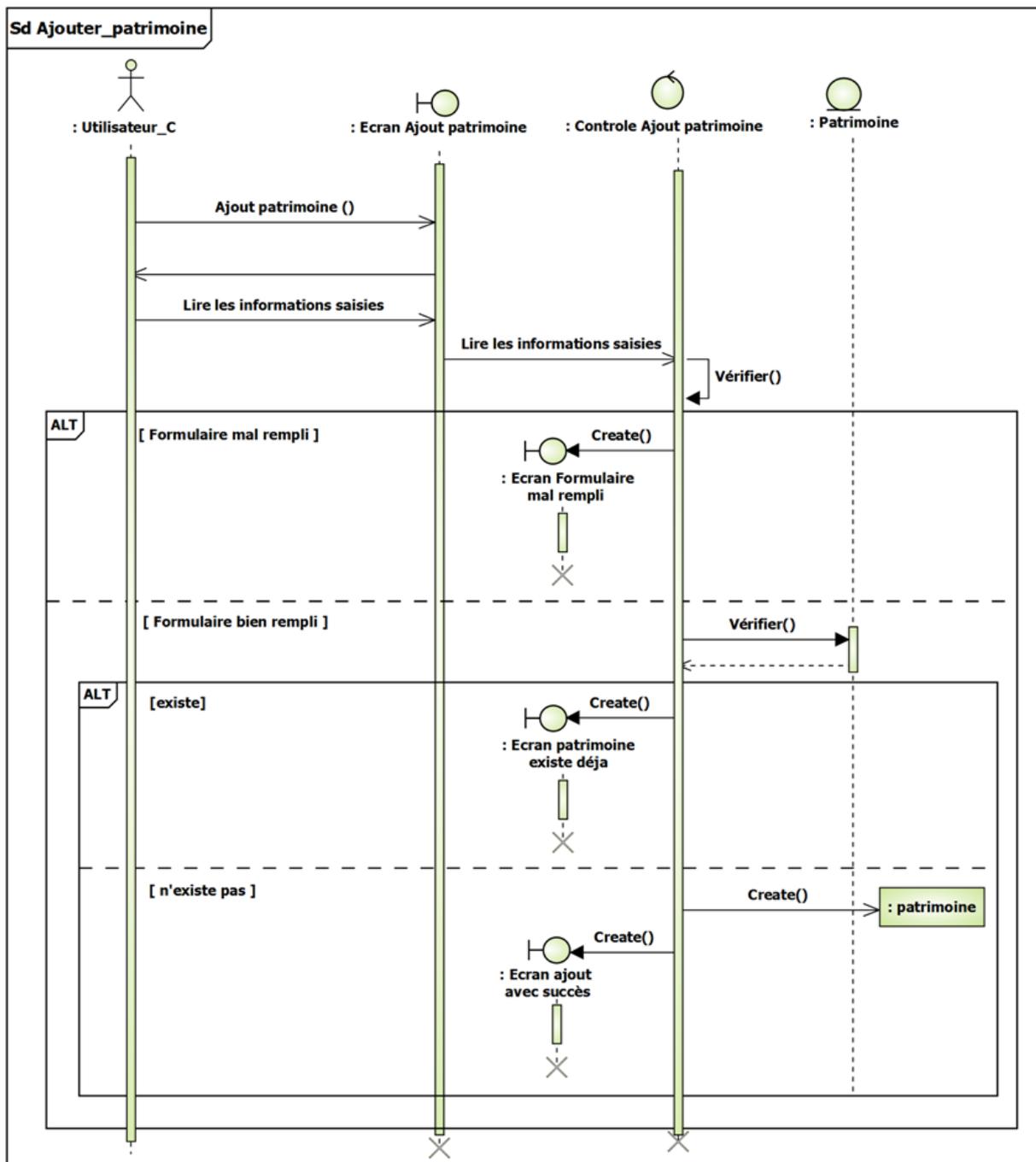


Figure 66: Diagramme d'interaction « Ajouter patrimoine ».

• Diagramme d'interaction decas d'utilisation modification patrimoine

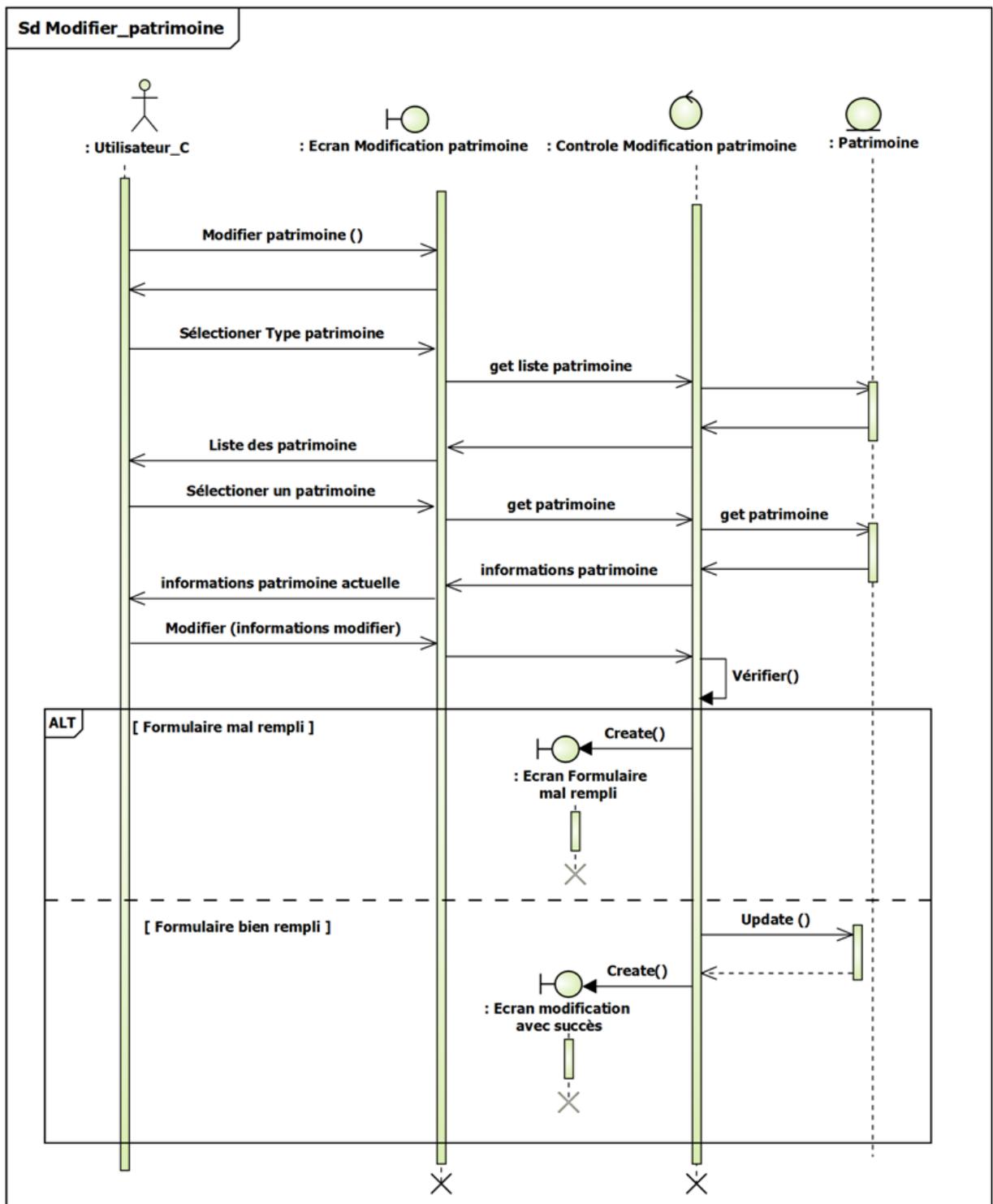


Figure 67: Diagramme d'interaction « Modifier patrimoine ».

• Diagramme d'interaction de cas d'utilisation suppression patrimoine

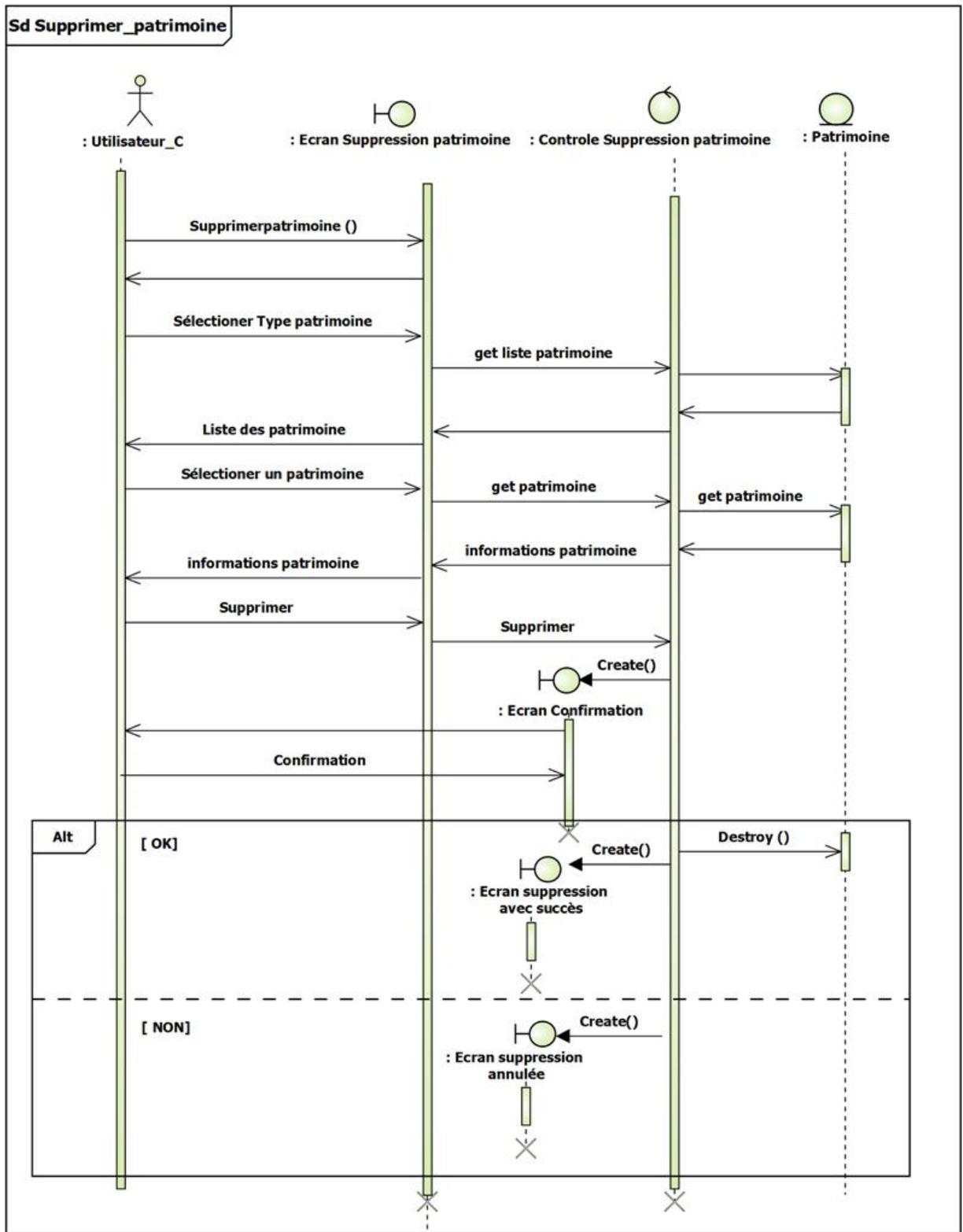


Figure 68:Diagramme d'interaction « Supprimer patrimoine ».

- Diagramme d'interaction de cas d'utilisation consultation patrimoine

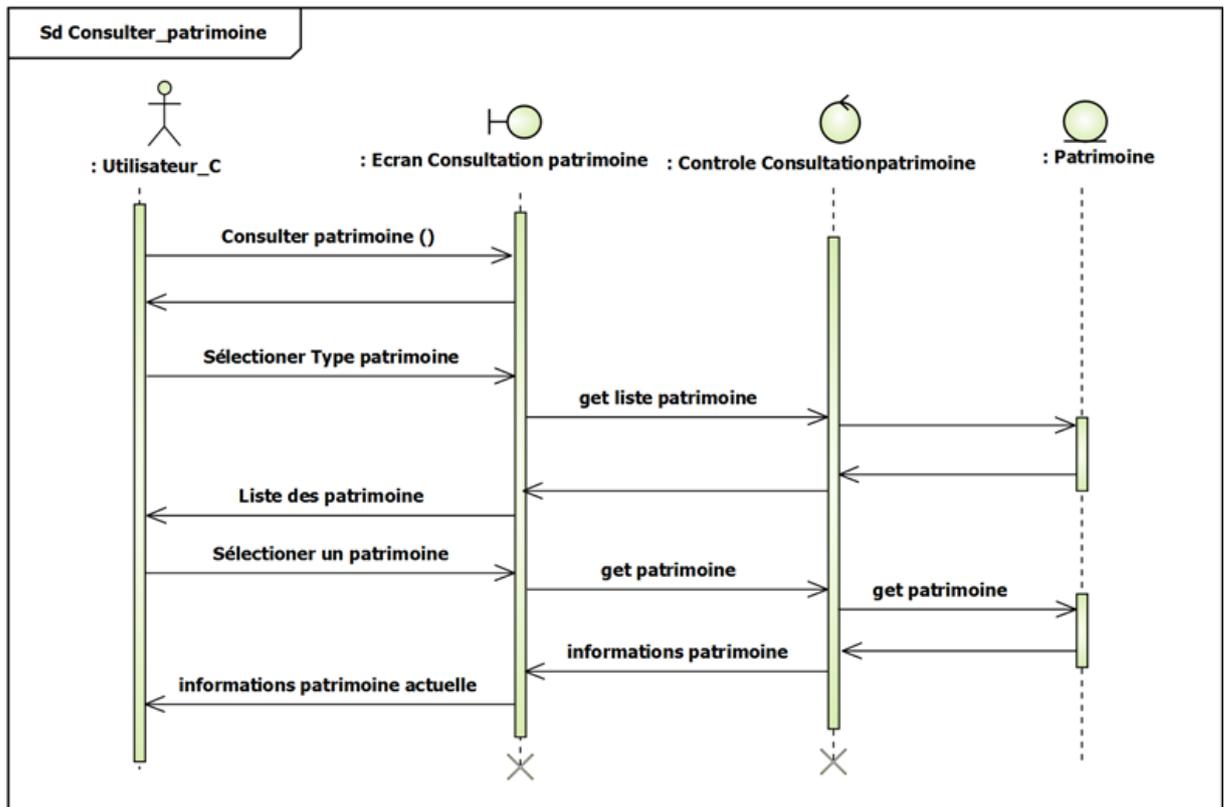


Figure 69: Diagramme d'interaction « Consulter patrimoine ».

## Cinquième partie : Conception Préliminaire

La conception préliminaire est certainement l'étape la plus délicate du processus 2TUP car elle en représente le cœur. C'est en effet à cette occasion que s'effectue la fusion des études fonctionnelles et techniques. En conséquence, plusieurs activités doivent coexister.

Il convient de :

- Passer de l'analyse objet à la conception.
- Intégrer les fonctions métier et applicatives du système dans l'architecture technique.
- Adapter la conception générique aux spécifications fournies par l'analyse. [4]

### 1. Développement du modèle de déploiement

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondant aux supports physiques (serveurs, routeurs) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables...) sur ces nœuds, le déploiement d'une solution client-serveur se construit sur la définition des postes de travail. [4]

#### 1.1. Architecture adopté

Le choix de notre solution c'est porté sur une architecture 2 tiers (Client-serveur) c'est -à-dire plusieurs clients, connectés à un serveur (BDD, Application), et cela en implémentant un réseau LAN. Ce choix repose sur les arguments suivants :

- Le choix des responsables qui ont exclu l'utilisation du net, ce qui élimine des architectures qui apporte le plus à notre solution.
- L'ensemble des utilisateurs se trouve dans différents zones géographiques.
- Le nombre des utilisateurs est restreint l'acquisition d'un serveur puissant exclut le risque de surcharge de réseau qui peut être engendré par le choix d'une architecture différente.

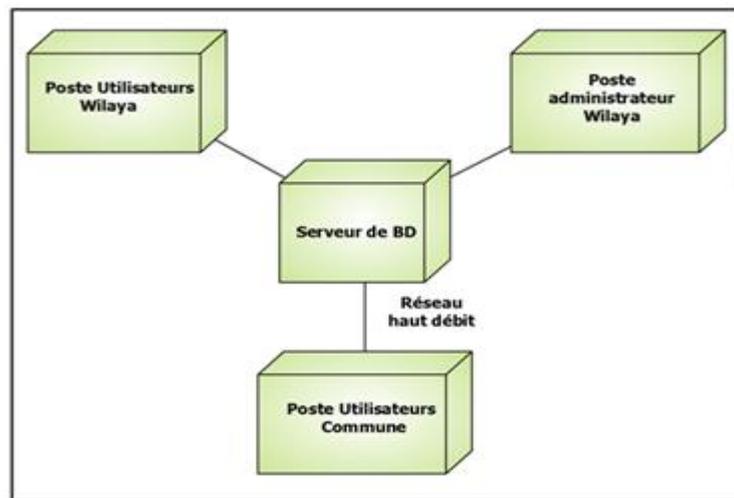


Figure 70: schéma du modèle de déploiement de notre système.

## 2. Développement du modèle d'exploitation

Par le biais du modèle d'exploitation nous définissons les applications installées sur les postes de travail, les composants métier déployés sur les serveurs et les instances de base de données implantées sur les serveurs également.

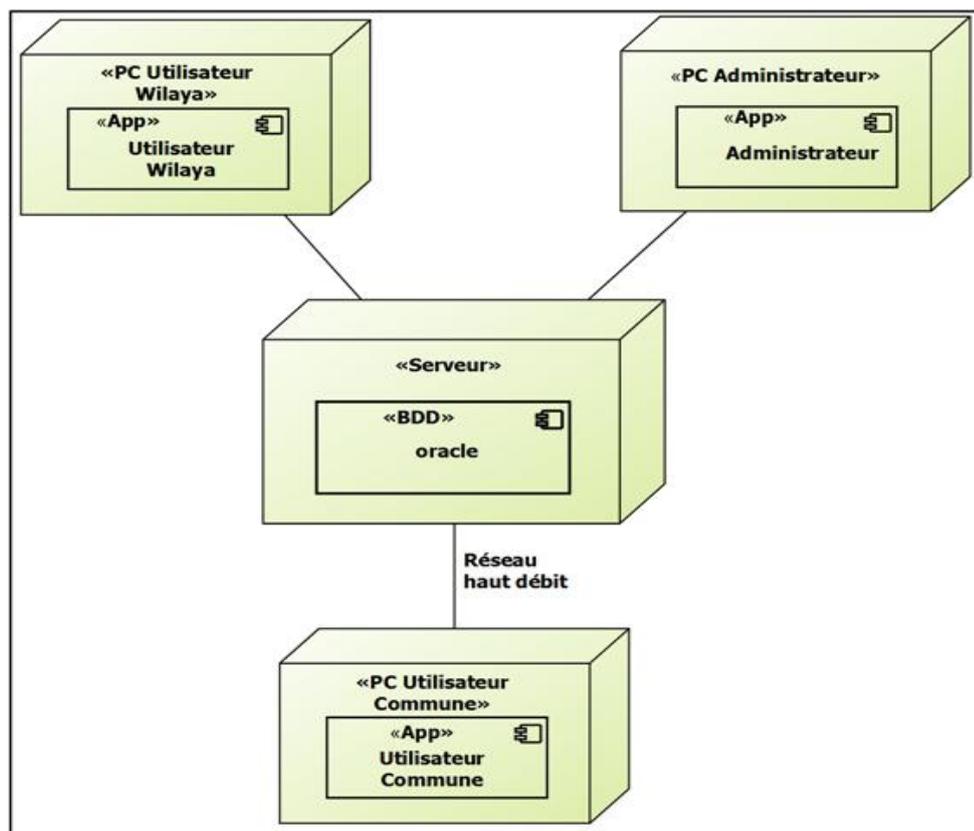


Figure 71: Définition des applications dans le modèle d'exploitation.

### 3. Définition des interfaces

Application	Interface	Description
Administrateur	Gestion d'utilisateur	Ajouter, modifier, supprimer, consulter un utilisateur.
Utilisateur commune	Gestion patrimoine	Ajouter, modifier, supprimer, consulter un patrimoine.
	Location	Créer, modifier, supprimer, consulter un contrat de location.
	Vente	Créer, modifier, supprimer, consulter un contrat de vente.
	Recherche multicritère	Rechercher d'un patrimoine, un contrat, un propriétaire par plusieurs critères au niveau de la commune.
	Statistique	Calculer les statistiques par plusieurs critères d'un patrimoine, contrat au niveau de la commune.
	Gestion propriétaire	Ajouter, modifier, supprimer, consulter un propriétaire.
Utilisateur Wilaya	Validation	Valider un contrat.
	Recherche multicritère	Rechercher d'un patrimoine, un contrat, un propriétaire par plusieurs critères au niveau de la Wilaya.
	Statistique	Calculer les statistiques par plusieurs critères d'un patrimoine, contrat au niveau de la wilaya

Tableau 48: Les interfaces de notre système.

### 4. Diagramme de composants

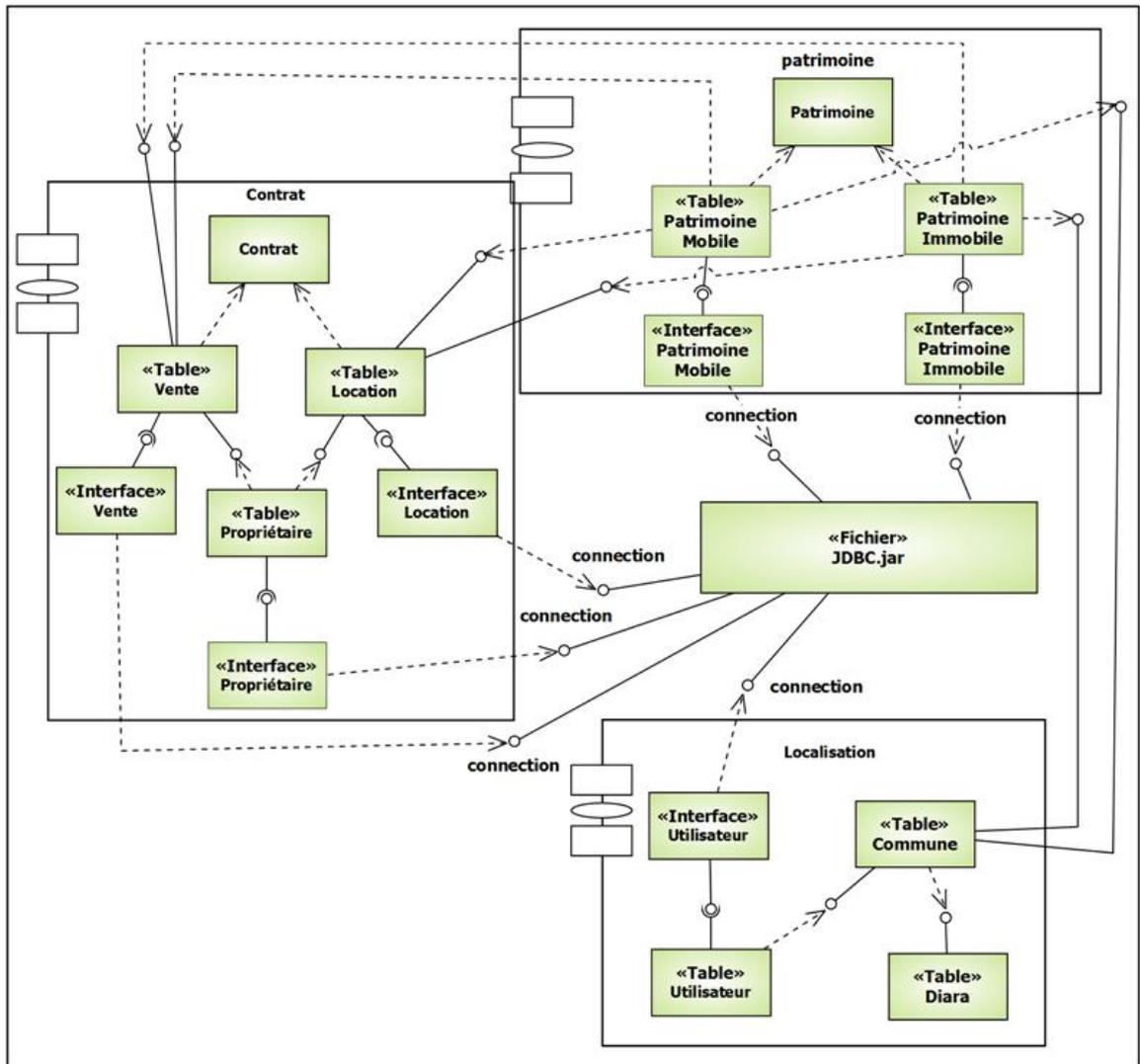


Figure 72: Diagramme de composants.

## Sixième partie : Conception Détaillée

La conception détaillée qui vient juste après est une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. Le modèle logique Y est particulièrement important dans la mesure où c'est dans cette étape qu'on génère le plus grand nombre d'informations. Il est ainsi possible de confier les catégories à des personnes différentes, qui pourront travailler indépendamment les uns des autres.[4]

### 1. Conception des classes

Classe	Description	Code	Type
<b>Daïra</b>	Code de daïra	Cod_daira	Number (15)
	Libelle de daïra	Libelle_daira	Varchar2(20)
<b>Commune</b>	Code de commune	Cod_commune	Number (15)
	Libelle de commune	Libelle_commune	Varchar2(20)
<b>Utilisateur</b>	Code d'utilisateur	Cod_utilisateur	Number (15)
	Nom d'utilisateur	Nom_utilisatuer	Varchar2(20)
	Type d'utilisateur	Typ_utilisateur	Varchar2(20)
	Mot de passe d'utilisateur	Mot_passe	Varchar2(20)
<b>Patrimoine</b>	Code du patrimoine	cod_patrimoine	Number (15)
	Désignation du patrimoine	Désignation	Varchar2(20)
	Catégorie du patrimoine	catégorie_patrimoine	Varchar2(20)
	Nature du patrimoine	Nature_patrimoine	Varchar2(20)
	Date d'enregistrement de patrimoine	date_enregistrement	Date
	Etat du patrimoine (venter/ louer)	Etat_patrimoine	Varchar2(20)
	Utilisation du patrimoine	utilisation_patrimoine	Varchar2(20)
	Valeur du patrimoine	Valeur	Number (15.5)
	Observation	Observation	Varchar2(3000)
		Marque du patrimoine	Marque_patrimoine
	Affectation du patrimoine	Affectation	Varchar2(20)

<b>Patrimoine Mobile</b>	Sortie de patrimoine	Sortie	Varchar2(20)
	Provenance	Provenance	Varchar2(20)
<b>Patrimoine Immobilie</b>	adresse du patrimoine	lieu_patrimoine	Varchar2(20)
	Surface du patrimoine	surface_patrimoine	Number (15.5)
	Date de contractions	Date_contractions	Date
	Référence de comptable	référence_comptable	Varchar2(20)
<b>Contrat</b>	Code du contrat	Cod_contrat	Number (15)
	Désignation du contrat	Désignation	Varchar2(20)
	Date d'enregistrement du contrat	Date_enregistrement	Date
	Cahier de charge du contrat	Cahier_charge	Blob
	Observation	Observation	Varchar2(3000)
	Validation de contrat	contrat_valide	Booléen
<b>Vente</b>	Prix de vente	prix_vente	Number (15.5)
<b>Location</b>	Date fin du contrat	date_fin_contrat	Date
	Durée du contrat	durée_contrat	Number (15)
	Prix du loyer	prix_loyer	Number (15.5)
	Revenue par mois du contrat	revenue_mois	Number (15.5)
	Revenue par année du contrat	revenue_année	Number(15.5)
	Etat de contrat de location	etat_contrat	Varchar2(20)
	Code de contrat renouveler	Cod_renov	Number (15)
<b>Propriétaire</b>	Code dupropriétaire	cod_propriétaire	Number(15)
	Nom dupropriétaire	Nom	Varchar2(20)
	Prénom dupropriétaire	Prénom	Varchar2(20)
	Date de naissance dupropriétaire	dat_naiss	Date
	Sexe dupropriétaire	Sexe	Varchar2(20)
	Adresse dupropriétaire	Adresse	Varchar2(20)

Tableau 49: Conception des classes.

## 2. Les opérations

Classe	Opérations	Description
<b>Utilisateur</b>	Ajouter	Ajouter un nouvel utilisateur
	Modifier	Modifier un utilisateur
	Supprimer	Supprimer un utilisateur
	Consulter	Consulter un utilisateur
<b>Patrimoine</b>	Ajouter	Ajouter un nouvel patrimoine
	Modifier	Modifier un patrimoine
	Supprimer	Supprimer un patrimoine
	Consulter	Consulter un patrimoine
	Rechercher	Rechercher sur un patrimoine
	Calculer les statistiques	Calculer les statistiques d'un patrimoine
<b>Contrat</b>	Créer	Créer un nouvel contrat
	Modifier	Modifier un contrat
	Supprimer	Supprimer un contrat
	Consulter	Consulter un contrat
	Rechercher	Rechercher sur un contrat
	Calculer les statistiques	Calculer les statistiques d'un contrat
	Valider	Valider les contrats
<b>Propriétaire</b>	Ajouter	Ajouter un propriétaire
	Modifier	Modifier un propriétaire
	Supprimer	Supprimer un propriétaire
	Consulter	Consulter un propriétaire

Tableau 50: Les opérations.

### 3. Diagramme de classe détaillée

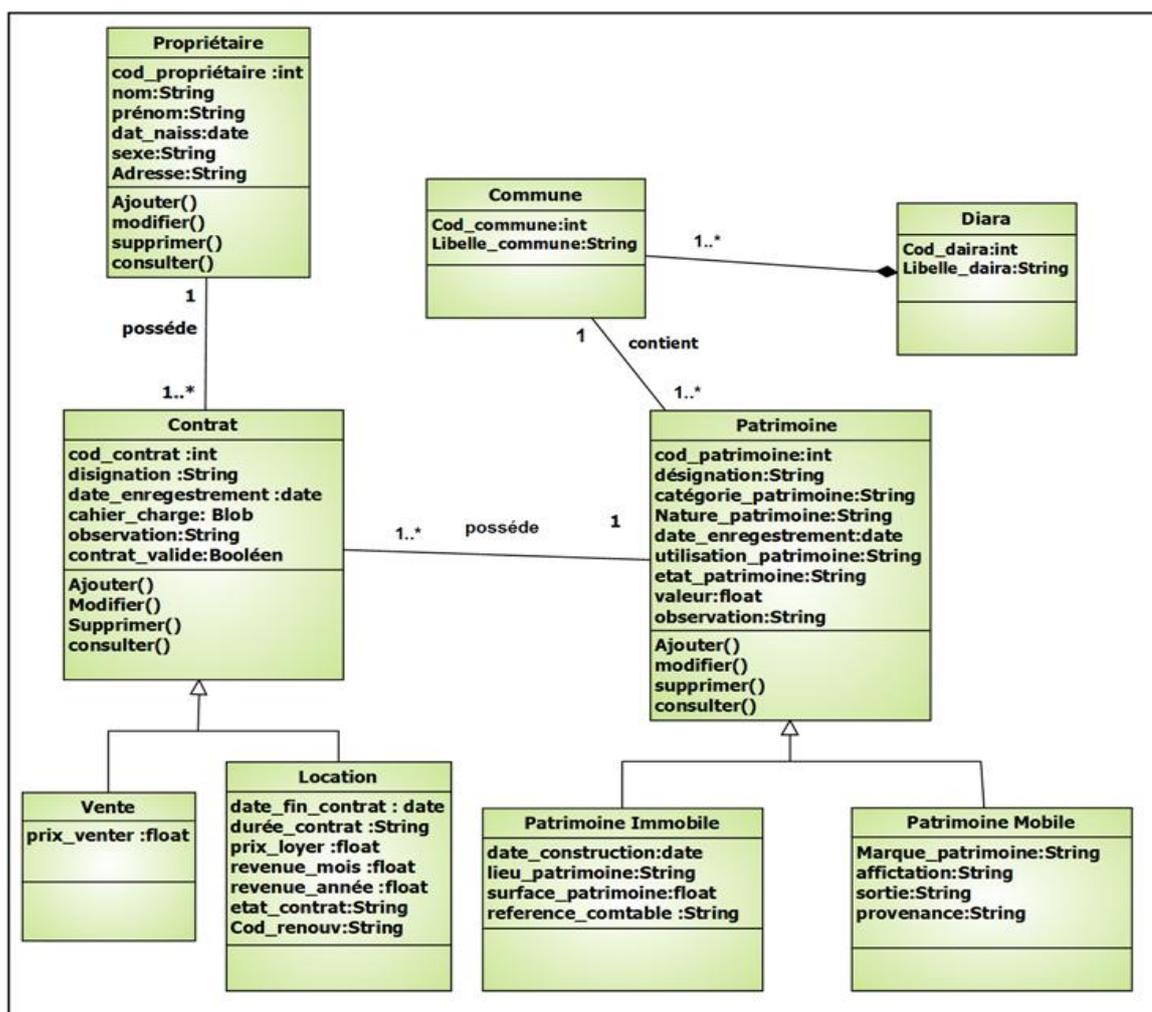


Figure 73: Diagramme de classes détaillé de notre système.

### 4. Passage vers le modèle relationnel

#### 4.1. Règles de passage

**Règle 1.**(Transformation des classes) chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.

**Règle 2.** (Association 1..\*) il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

**Règle 3.**(Association 1..1) il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. Si les deux

multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.

**Règle 4.** Cas héritage,

- L'héritage entre les sous-classes C1, C2..Cn et la classe mère C sera traduit par la répétition des attributs représentant C au niveau de chacune des relations représentant C1,C2..Cn.
- L'héritage entre les sous-classes C1, C2..Cn et la classe mère C sera traduit par une relation unique comportant la traduction de la classe C avec les attributs de C1,C2..Cn.

**Règle 5.** Cas de composition, migration de clé primaire de la super classe composé clé étranger de la classe composant.

## 5. Les 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 :

- **Daïra** (cod\_daira, libelle\_daïra).
- **Commune** (cod\_commune, libelle\_commune, #cod\_daïra).
- **Patrimoine** (cod\_patrimoine, typ\_patrimoine, désignation, catégorie\_patrimoine, Nature\_patrimoine, date\_enregistrement, utilisation\_patrimoine, etat\_patrimoine, valeur, observation, date\_construction, lieu\_patrimoine, surface\_patrimoine, reference\_comtable, Marque\_patrimoine, affectation, sortie, provenance, #cod\_commune).
- **Utilisateur** (cod\_utilisateur, type\_utilisateur, nom\_utilisateur, mot\_passe, #cod\_commune).
- **Propriétaire** (cod\_propriétaire, nom, prénom, dat\_ress, sexe, adresse).
- **Vente** (Cod\_contrat, Désignation, contrat\_valider, Date\_enregistrement, observation, Cahier\_charge, prix\_vente, # cod\_propriétaire, #cod\_commune, #cod\_patrimoine).
- **Location** (Cod\_contrat, Désignation, Date\_enregistrement, observation, Cahier\_charge, date\_fin\_contrat, durée\_contrat, prix\_loyer, revenue\_mois, revenue\_année, etat\_contrat, cod\_renov, contrat\_valider, # cod\_propriétaire, #cod\_commune, #cod\_patrimoine).

## **Conclusion**

Durant ce chapitre de conception et analyse, nous avons conçu et documenté le code que nous devons produire. Dans cette phase, toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (ensemble de diagrammes) prêt à être codé. Dans le prochain chapitre, nous allons montrer comment nous avons traduit cette étude conceptuelle et à l'aide de quelques outils nous avons pu mettre en place notre application.

# Chapitre

*LE DOSSIER*

*TECHNIQUE*

4

## Introduction

Dans ce chapitre, consacré à la réalisation et la mise en œuvre de notre application client-serveur pour la gestion du patrimoine communal de la wilaya de Mila, nous allons présenter les outils de développement adoptés; soit le système de gestion de base de données **Oracle**, le langage de programmation **Java**, ainsi que l'environnement utilisé qui est **Netbeans** et enfin nous montrer les principales interfaces et fenêtres de l'application.

### 1. Présentation des outils de développement de l'application



#### 1.1. Le langage de programmation java

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises et mobiles. [10]



#### 1.2. Implémentation de la base de données (Oracle)

Pour créer la base de données de notre application, nous avons utilisé l'SGBD oracle. Oracle est un système de gestion de base de données relationnel. C'est un bon choix si vous souhaitez avoir une base de données pour une application multiposte, de grande taille mais facilement gérable, ou que vous connaissez grande chose aux SGBD.

- **Les fonctionnalités d'Oracle.** Oracle permettant d'assurer :
  - La définition et la manipulation des données
  - La cohérence des données
  - La confidentialité des données
  - L'intégrité des données
  - La sauvegarde et la restauration des données
  - La gestion des accès concurrents.



### 1.3. Environnement de développement de l'application

Pour réaliser notre application, nous avons utilisé l'environnement NetBeans. Est un outil de développement puissant et rapide pour la programmation d'applications.

Netbeans est un IDE qui supporte une large variété de langages de programmation et d'outils de collaboration. L'environnement de base comprend les fonctions générales suivantes:

- configuration et gestion de l'interface graphique des utilisateurs,
- support de différents langages de programmation,
- Comprend un explorateur de bases de données qui supporte toutes les bases relationnelles.

## 2. Description de l'application

Dans cette section nous présentons quelque fenêtre de notre application :

### 2.1. Application administrateur

#### 2.1.1. Fenêtre d'authentification

Cet interface d'authentification permet à l'administrateur d'accéder au système.



Figure 74: Fenêtre d'identification administrateur.

Si l'administrateur entre le nom ou mot de passe incorrect cette fenêtre est affichée.

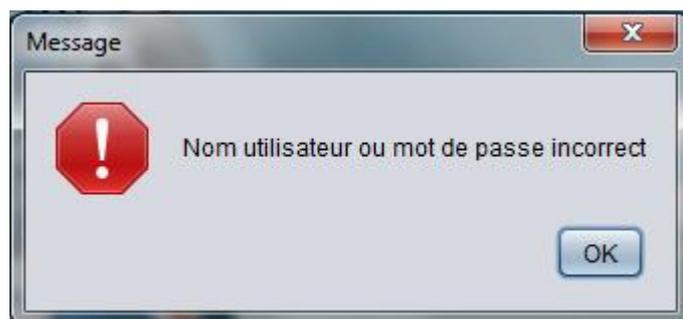


Figure 75: Message d'erreur.

### 2.1.2. Fenêtre d'accueil de l'administrateur

Après l'authentification, une fenêtre d'accueil s'affiche pour permettre à l'administrateur d'accéder aux différents services offerts par l'application en cliquant sur le bouton correspondant.



Figure 76 : Fenêtre d'accueil administrateur.

### 2.1.3. Fenêtre de gestion des utilisateurs

Cette interface de gestion des utilisateurs permet à l'administrateur de réaliser les mises à jour.

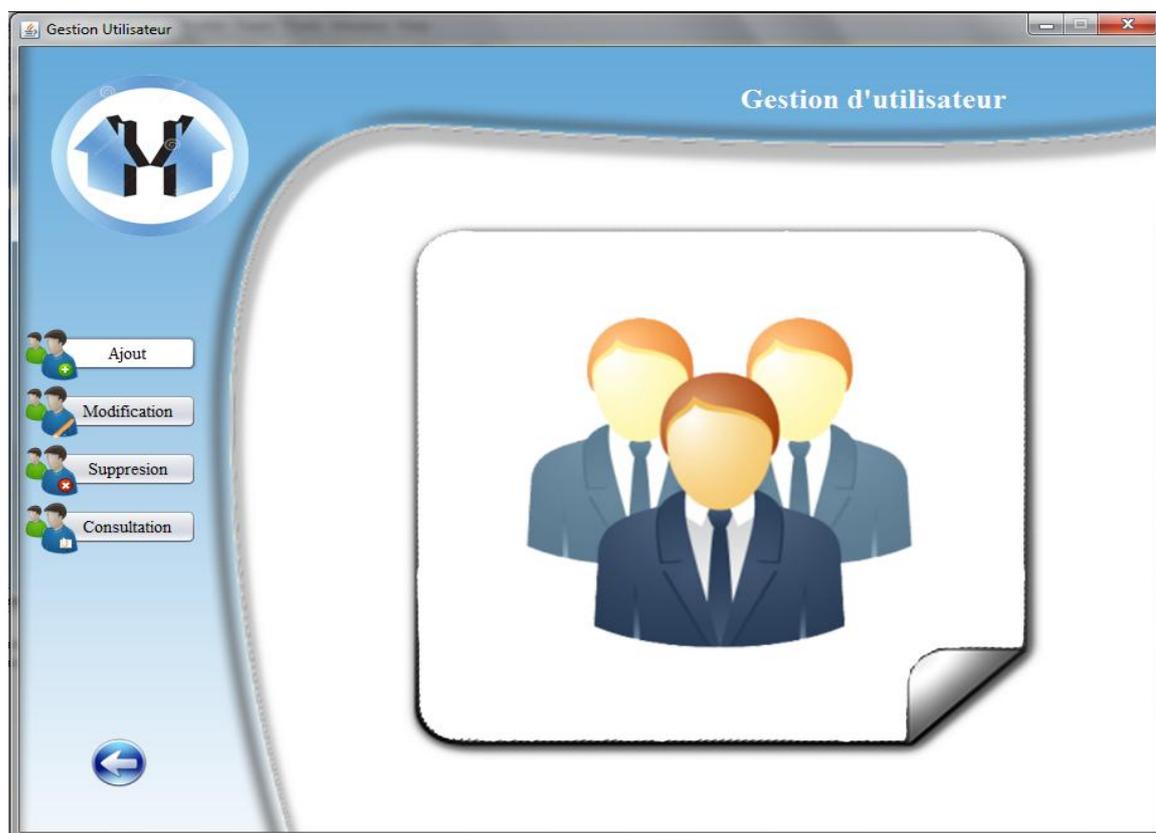


Figure 77 : Fenêtre de gestion des utilisateurs.

### 2.1.3.1. Fenêtre d'ajouter un utilisateur

Cette interface d'ajout un utilisateur permet à l'administrateur de réaliser l'ajout de l'utilisateur.

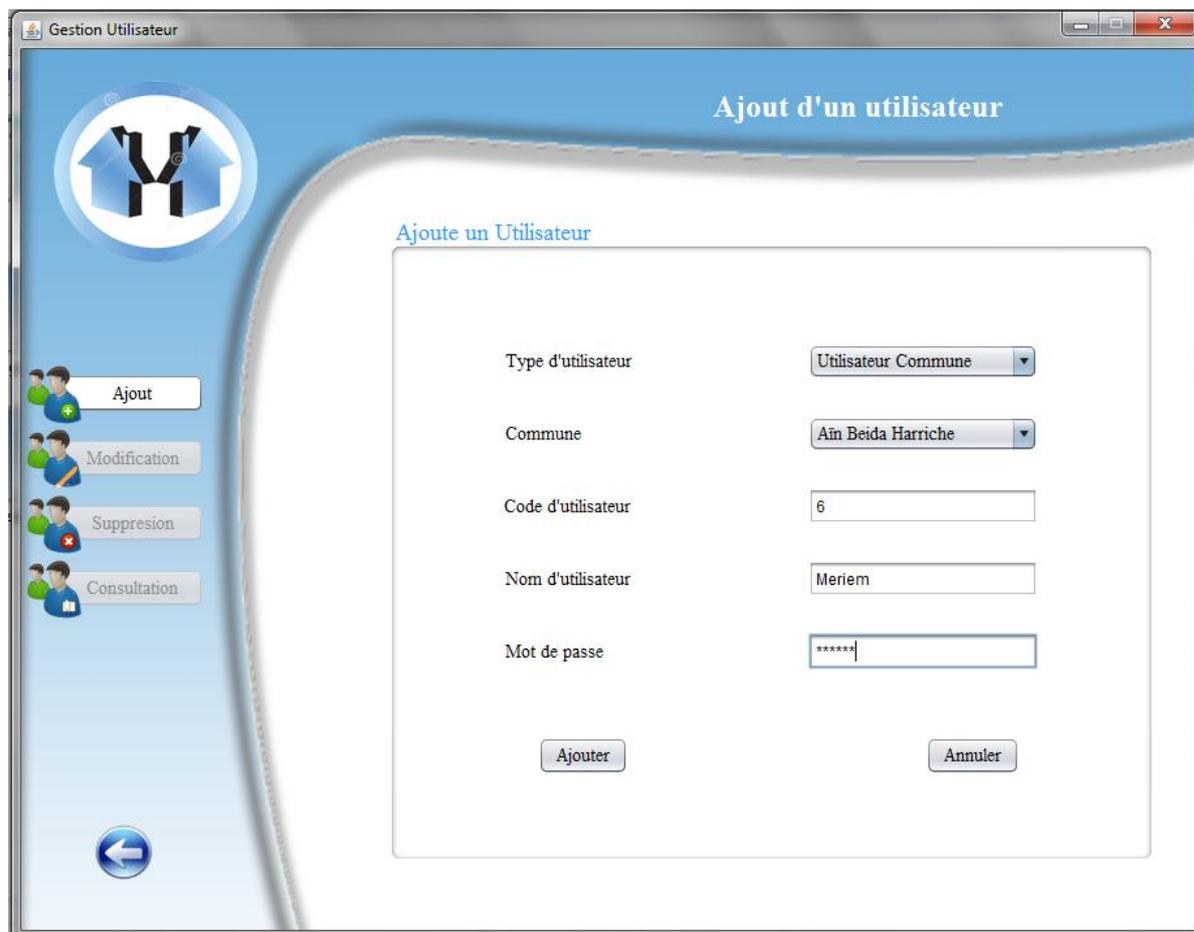


Figure 78 : Fenêtre d'ajouter un utilisateur.

### 2.1.3.2. Fenêtre de modifier un utilisateur



Figure 79 : Fenêtre de modifier un utilisateur.

### 2.1.3.3. Fenêtre de supprimer un utilisateur



Figure 80 : Fenêtre de supprimer un utilisateur.

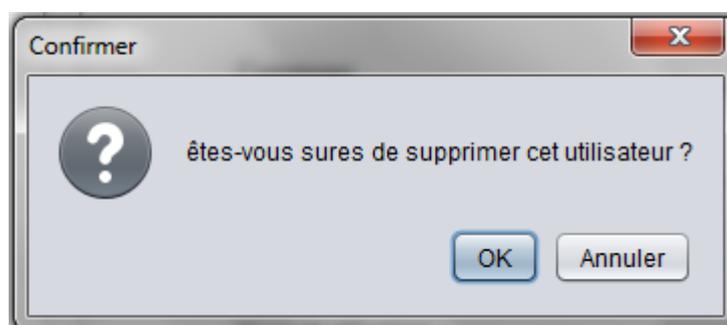


Figure 81 : Message de confirmation.

## 2.2. Application de la Wilaya

### 2.2.1. Fenêtre d'authentification

Cette interface d'authentification permet aux utilisateurs de la wilaya d'accéder au système.



Figure 82 : Fenêtre d'identification utilisateur wilaya.

### 2.2.2. Fenêtre d'accueil aux l'utilisateur wilaya

Après l'authentification, une fenêtre d'accueil s'affiche pour permettre aux d'utilisateurs wilaya d'accéder aux différents services offerts par l'application en cliquant sur le bouton correspondant.



Figure 83 : Fenêtre d'accueil utilisateur wilaya.

## 2.3. Application da la commune

### 2.3.1. Fenêtre d'authentification

Cet interface d'authentification permet aux utilisateurs des communes d'accéder au système.



Figure 84 : Fenêtre d'identification utilisateur commune.

### 2.3.2. Fenêtre d'accueil aux l'utilisateur commune

Après l'authentification, une fenêtre d'accueil s'affiche pour permettre aux d'utilisateurs commune d'accéder aux différents services offerts par l'application en cliquant sur le bouton correspondant.



Figure 85 : Fenêtre d'accueil utilisateur commune.

### 2.3.3. Fenêtre de gestion du patrimoine

Cette interface de gestion du patrimoine permet à aux utilisateurs des commune de réaliser les mises à jour.



Figure 86: Fenêtre de gestion du patrimoine.

### 2.3.4. Fenêtre de la création un contrat administratif de vente

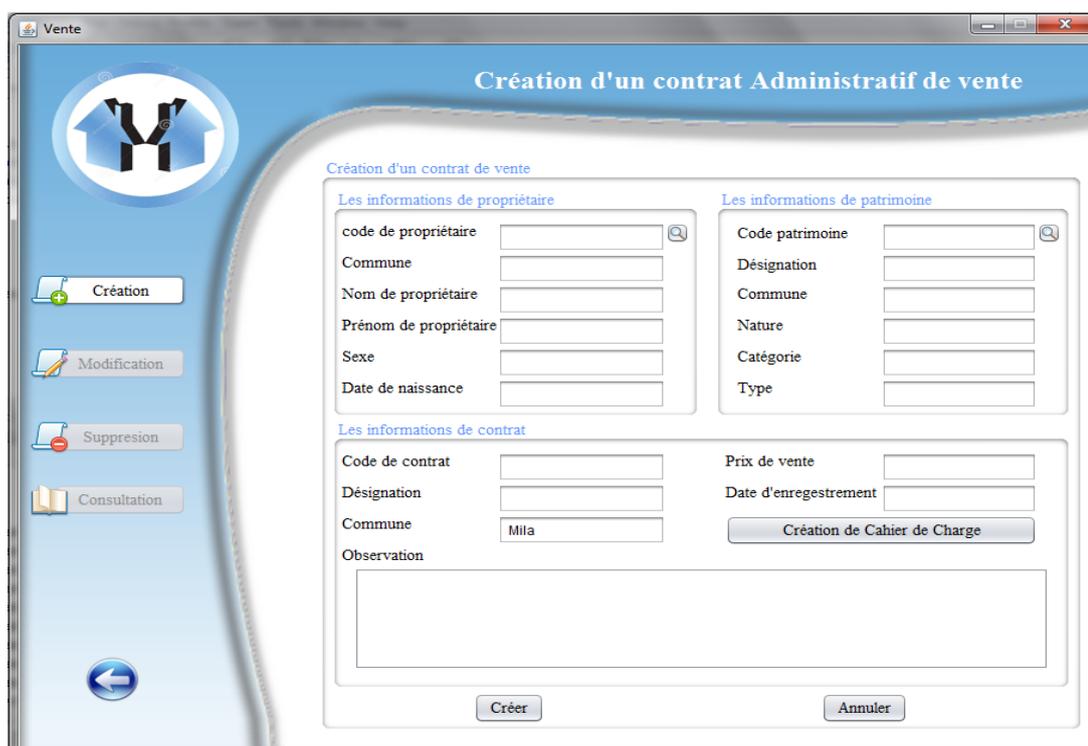


Figure 87: Fenêtre de la création un contrat de vente.

### 2.3.5. Fenêtre de la création un contrat administratif de location

Location

### Création du contrat Administratif de location

Création d'un contrat de location

**Les informations de propriétaire**

code de propriétaire

Commune

Nom de propriétaire

Prénom de propriétaire

Sexe

Date de naissance

**Les informations de patrimoine**

Code patrimoine

Désignation

Commune

Nature

Catégorie

Type

**Les informations de contrat**

Code de contrat

Désignation

Commune

Observation

Prix de location

Date d'enregistrement

Durée de contrat

Date fin de contrat

Revenu de mois

Revenu de l'année

Création de Cahier de Charge

Créer Annuler

Figure 88: Fenêtre de la création un contrat de location.

### 2.3.6. Fenêtre de recherche patrimoine

Recherche

### Recherche Patrimoine

Recherche Patrimoine

Choisit un ou plusieurs critères de recherche

- Recherche par type de patrimoine
- Recherche par catégorie de patrimoine
- Recherche par date d'enregistrement
- Recherche par code de patrimoine
- Recherche par type de contrat

Saisit et/ou choisit les critères de recherche

Type de patrimoine

Catégorie de patrimoine

Type de contrat

Code de patrimoine

Date d'enregistrement

Recherche Annuler

Figure 89 : Fenêtre de recherche patrimoine.

### 2.3.7. Fenêtre de recherche contrat

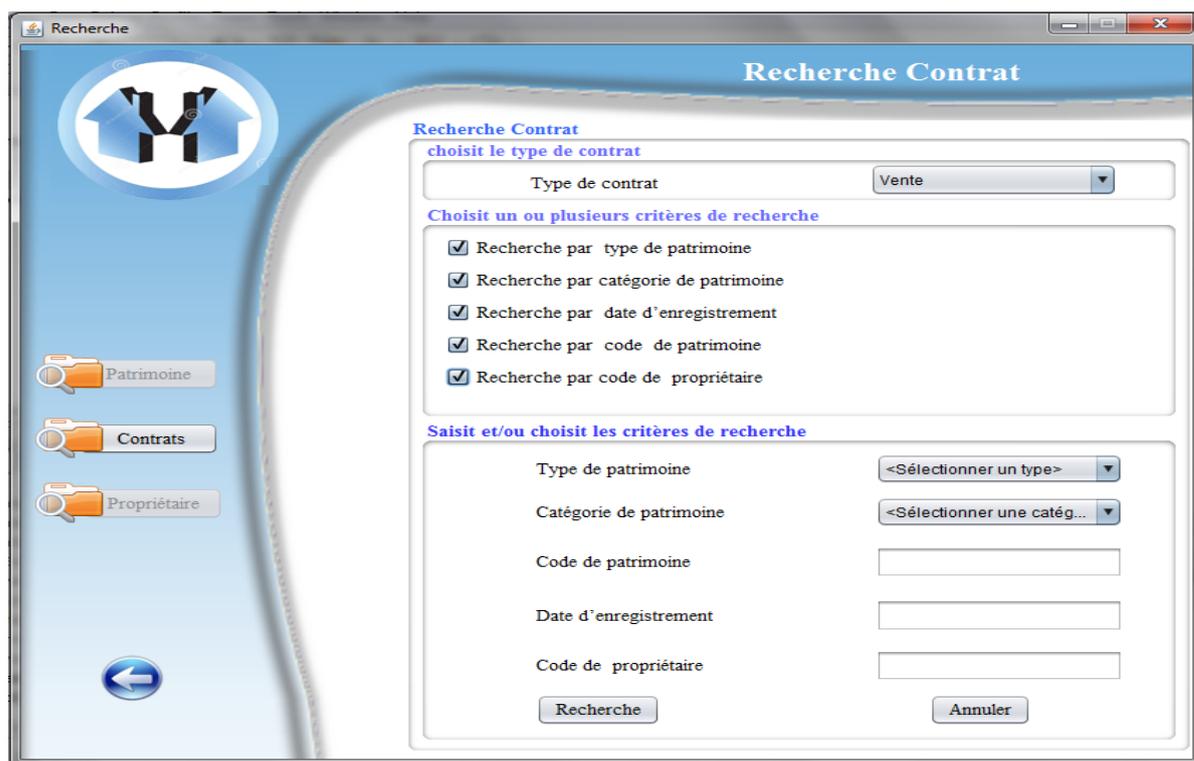


Figure 90 : Fenêtre de recherche contrat.

### 2.3.8. Fenêtre de statistique patrimoine

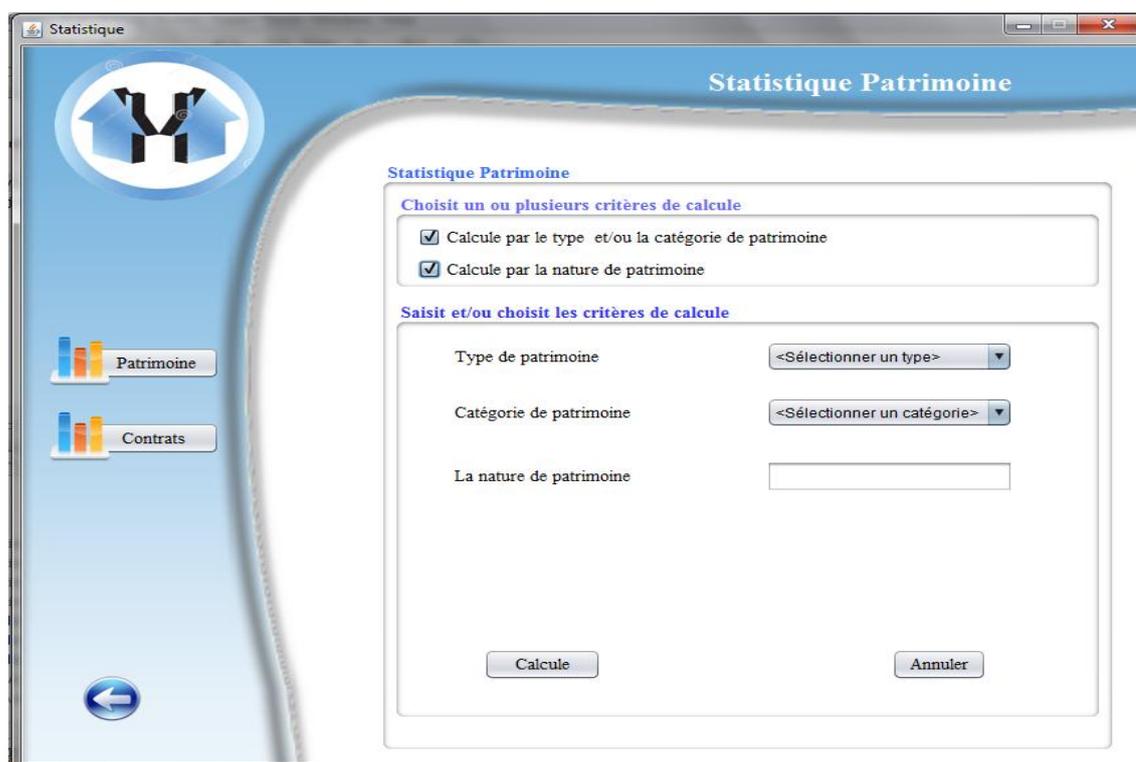


Figure 91 : Fenêtre de statistique patrimoine.

## **Conclusion**

Dans cette dernière partie de notre projet, nous avons présentés les différents outils de développement de notre application ainsi que ses interfaces essentielles.

# Conclusion générale

## Conclusion générale

Notre projet a été réalisé dans le cadre d'un projet de fin d'études. Il a pour objectif l'automatisation la gestion du patrimoine communal au sein de la wilaya de Mila dans le service de l'animation communale des marchés.

Notre travail se résume en la réalisation d'une application client-serveur permettant la gestion du patrimoine communal. Dans ce contexte, nous avons cherché à développer une application flexible et évolutive permettant son amélioration par la suite afin d'anticiper les changements continus des besoins des utilisateurs.

Le projet s'est déroulé selon trois axes principaux afin de passer par les étapes essentielles de tout projet : l'analyse, la conception et la réalisation.

Pour la conception de notre application, nous avons eu recours à la méthode 2TUP. Cette approche nous a permis de bien comprendre la problématique et de bien modéliser les objectifs à atteindre. Ce qui nous a donné la possibilité de réaliser un système stable et évolutif.

Pour la réalisation, nous avons utilisé JAVA comme langage de programmation et ORACLE comme système de gestion de base de données.

En outre, ce projet était une opportunité pour enrichir nos connaissances dans des domaines variés comme : L'orienté objet, UML, 2TUP, le langage JAVA, ...

D'autre part, ce projet a été une expérience réussie qui nous a permis de bénéficier d'un stage pratique. Ce stage a contribué à l'augmentation de nos connaissances dans le domaine de la recherche d'information. Il a également eu le rôle de nous rapprocher au monde du travail.

# *Résumé*

Le projet réalisé au sein de ce mémoire de fin d'études, consiste à la réalisation d'une application client- serveur pour la gestion du patrimoine communal. C'est une application qui permet l'automatisation du travail et l'amélioration de la documentation, la communication et l'échange de l'information de gestion du patrimoine communal entre le service de l'animation communal des marchés de la wilaya et le service de l'administration et moyens généraux de la commune.

Les objectifs majeurs de cette application sont la gestion, le suivi et le contrôle du patrimoine. Pour atteindre ces objectifs, nous avons eu recours à la création d'un système basé sur une architecture 2 tiers. Ainsi, nous avons mené une étude conceptuelle en suivant la démarche 2TUP, utilisé UML comme langage de modélisation, NetBeans comme IDE et Oracle comme système de gestion de bases de données.

**Les mots clés :** Orienté Objet, UML, JAVA, Oracle, 2TUP, IDE.

# *Summary*

The project carried out in this dissertation study is to achieve a client-server application for managing municipal assets. This is an application that enables the automation of work and improving the documentation, communication and the exchange of management information between the municipal heritage service of communal entertainment markets and the wilaya service administration and general resources of the municipality.

The major objectives of this application are the management, monitoring and control of assets. To achieve these objectives, we used to create a report based on a 2-tier architecture system. Thus, we conducted a conceptual study following the 2TUP approach used UML as modeling language, such as IDE and NetBeans as Oracle management system databases.

# Bibliographiques

## *Ouvrage :*

- [4] P.Roques et F.Vallée, UML en action, 4 iémé édition 2007.
- [9] La loi n°90-09 du 07 avril 1990 relative à la wilaya.
- [10] Jean-Michel DOUDOUX, Développons en Java.

## *@ Sitesweb:*

- [1] [http://www.memoireonline.com/10/12/6180/m\\_Etude-dune-replication-symetrique-asynchrone-dans-une-base-de-donnees-repartie-Application-12.html](http://www.memoireonline.com/10/12/6180/m_Etude-dune-replication-symetrique-asynchrone-dans-une-base-de-donnees-repartie-Application-12.html).
- [2] [http://www.memoireonline.com/05/12/5812/m\\_La-realisation-dune-application-de-contrle-total-des-processus-dun-ordinateur-distant3.html](http://www.memoireonline.com/05/12/5812/m_La-realisation-dune-application-de-contrle-total-des-processus-dun-ordinateur-distant3.html).
- [3] <http://www.commentcamarche.net/contents/222-environnement-client-serveur>.
- [5] <http://fr.scribd.com/doc/19215590/Architecture-Client-Serveur>.
- [7] <http://www.wilaya-de-mila.sitew.com>
- [8] [http://fr.wikipedia.org/wiki/Commune\\_\(Alg%C3%A9rie\)](http://fr.wikipedia.org/wiki/Commune_(Alg%C3%A9rie)).

## *Mémoires de fin d'étude :*

- [6] H Amel, M Seloua ; Modélisation et réalisation d'un site web dynamique d'un agence commerciale avec le langage de modélisation UML et le processus de développement 2TUP ; université menturi Constantine ; 2007.