

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



N° Réf :.....

Centre Universitaire
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

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

En : Informatique

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

Thème

Un système de gestion optimisé pour
le service d'urgence d'un hôpital
étude de cas : service de Coronavirus

Préparé par : Bouhenniche Khaoula

Harizi Rami

Soutenu devant le jury

Encadré par : M^f Kimouche Abdelkader MAA

Président : M^{me} Afri Faiza MAA

Examineur : M^f Attia Mourad MAA

Année Universitaire : 2021/2022

Remerciement

Nous remercions d'abord et avant tout Allah qui nous a donné la volonté et la patience pour réaliser ce travail.

Un remerciement particulier à notre encadreur M. Kimouche Abdelkader pour sa présence, ses précieux conseils et son aide durant toute la période du travail.

Nous remercions également les membres de jury Pour d'avoir accepté d'examiner notre travail Et de l'enrichir par leurs propositions.

Sans oublier tous les enseignants qui ont contribué à notre formation durant notre vie scolaire particulièrement les enseignants de notre institut.

Nous remercions le personnel qui travaille dans l'établissement public hospitalier Frère Tobal-Mila surtout l'équipe du Service COVID 19 qui offrent les meilleurs conditions et moyens pour suivre notre stage.

Enfin, nous remercions très sincèrement tous nos familles pour leur encouragement sans limite.

Merci à tous

Rami et Khaoula

Dédicace

C'est avec joie que je dédie ce modeste travail :

À mes très chers parents

Pour tout l'amour dont vous m'avez entouré, pour tout ce que vous avez fait pour moi Je ferai de mon mieux pour rester un sujet de fierté à vos yeux avec l'espoir de ne jamais vous décevoir

À Ma chère sœur : Malak

À Mon cher frère : Issam et sa femme, sans oublier la petite nana,

Eline Mayassin

À tout mes collègues du Centre Universitaire de Mila

À Mon binôme Khaoula qui a partagé avec moi les moments difficiles de ce travail.

À tous mes collègues qui travaillent avec moi à la trésorerie et aussi de la direction des impôts de la wilaya de Mila , en particulier mes collègues de la cellule informatique qui m'ont soutenu et m'ont encouragé à terminer mes études

À tous les membres de la famille Harizi et Bouarroudj sans citer les noms.

À tous les amies que je n'ai pas citées.

Rami

Dédicace

*Je remercie dieu parce qu'il est toujours à mes côtés et toujours
me donne des opportunités pour réussir et m'honorer*

*Je dédie ce travail à tous les membres de ma famille
spécialement ma fille qui toujours m'inspire pour continuer*

HANINE

A Mon binôme Rami qu'on a partagé ce travail ensemble.

*À mes collègues de travail de service de la maintenance dans
les transmissions national dans la wilaya de Mila,
spécialement qui m'ont soutenu et m'ont encouragé.*

*À ceux qui m'ont aidé, qui m'ont donné le courage de continuer
et qui m'ont donné la motivation d'aller de l'avant.*

Khaoula

Résumé

Ce mémoire est le résultat d'un stage effectué au sein de l'établissement public hospitalier (EPH) FRERES TOBAL Mila dans le cadre du projet de fin d'étude du Master pour la conception et la réalisation d'un système de gestion optimisé pour le service de Coronavirus. Ce système assure une gestion d'hospitalisation d'un patient de son admission jusqu'à la fin du soin, il garantit une bonne affectation des patients aux médecins à l'aide d'un module basé sur une technique d'optimisation, et en cas de pénurie des médicaments, il assure un partage efficace des quantités disponibles afin de minimiser les pertes humaine.

Pour la réalisation de ce système, on a adopté la méthode 2TUP et le langage UML pour la conception et la modélisation, JAVA comme langage de programmation et SQL pour l'implémentation de la base de données.

Mots clés : Service de COVID, Optimisation, Recuit simulé, 2TUP, UML, JAVA, SQL

ملخص

هذه المذكرة هي نتيجة تربص ميداني تم إجراؤه بالمؤسسة العمومية الاستشفائية الإخوة طوبال ميله في إطار مشروع التحضير لنيل شهادة الماستر بهدف تصميم وتحقيق نظام إدارة محسن لمصلحة كوفيد. هذا النظام يضمن خدمة استشفائية للمريض منذ دخوله إلى المصلحة إلى غاية نهاية فترة العلاج، حيث يتم اقتراح توزيع مناسب للأطعم الطبية على المرضى باستخدام وحدة تعتمد على تقنية التحسين، كما يضمن توزيعا فعالا للأدوية المتاحة في حالة الندرة، من أجل تقليل الخسائر البشرية.

لتحقيق هذا النظام، اعتمدنا طريقة TUP2 و UML كنهج للتصميم و النمذجة، JAVA كلغة برمجة و SQL لتنفيذ قاعدة البيانات.

Table des matières

Introduction général	1
Chapitre 1 : COVID-19 et Le système sanitaire Algérien	
Introduction	3
I. Le système sanitaire en Algérie.....	3
1. La création de système sanitaire en Algérie	3
2. Crises du secteur de la santé	3
3. Problématique.....	3
3.1 Définition.....	3
3.2 Symptômes	4
3.3 Mode de propagation	4
3.4 Le traitement	4
3.5 Prévention.....	4
4. Enseignements tire de la pandémie COVID-19	5
4.1 Un système de santé fortement centralisé	5
4.2 La difficulté des conditions de travail.....	5
4.3 Absence des médecins libéraux	6
4.4 La téléconsultation comme solution de rechange	6
4.5 Des taux de remboursement complètement obsolètes	6
4.6 Pénurie de certains médicaments	6
II. Présentation de l'établissement public hospitalier	7
1. Organisme d'accueil	7
1.1 Définition et création	7
1.2 Classification	7
1.3 Géographie et position de l'établissement hospitalier EPH de Mila	7
1.4 Missions de l'organisme d'accueil	8
1.5 L'organigramme de l'EPH	9
1.6 Etat global du personnel de l'hôpital	11
1.7 Les Ressources humaines et matériaux	11
2. Service de COVID-19	12
2.1 Prise en charge des patients COVID-19.....	12
2.2 Protocole de prise en charge des patients.....	12
2.3 Indications des examens selon la présentation clinique	13
3. Le parcours d'hospitalisation	13
3.1 Au niveau d'accueil : la consultation COVID-19	13
3.2 A l'admission	13
3.3 Au cours de l'hospitalisation.....	13
3.4 Sortie	14
Conclusion	15

Chapitre 2 : La méthodologie de développement 2TUP

Introduction	16
I. La méthodologie de développement 2TUP	16
1. Définition d'un processus de développement logiciel	16
1.1.Le Processus unifié	16
1.2.Le Processus 2TUP	17
1.2.1. La branche gauche (fonctionnelle)	18
1.2.2. La branche droite (architecture technique)	19
1.2.3. La branche du milieu.....	19
2. Un processus de modélisation avec UML	19
II. Etude de cas	21
1. Etude préliminaire.....	21
1.1.Elaboration du cahier de charge	21
1.1.1. la présentation du projet	21
1.1.2. les choix techniques	21
1.1.3. Recueil des besoins fonctionnels.....	22
1.1.4. Recueil des besoins opérationnels.....	23
1.2. Description du contexte	24
1.2.1. Identification des acteurs.....	24
1.2.2. Identification des messages.....	24
1.3. Modélisation du contexte	26
1.3.1. Diagramme de contexte dynamique	26
1.3.2. Signification des messages.....	27
2. Capture des besoins fonctionnels.....	28
2.1.Déterminer les cas d'utilisation	28
2.1.1. Liste préliminaire des cas d'utilisation	28
2.1.2. Description préliminaire des différents cas d'utilisation	29
2.1.3. Diagramme cas d'utilisation des besoins fonctionnels.....	30
2.2. Description détaillée de chaque cas d'utilisation	32
2.2.1. Cas d'utilisation « Gérer les comptes personnels »	32
2.2.1.1.Fiche descriptive	32
2.2.1.2.Diagramme d'activité cas d'utilisation « Gérer les comptes personnels »	33
2.2.1.3.Diagramme de séquence cas d'utilisation « Gérer les comptes personnels »	34
2.2.2. Cas d'utilisation « Faire le planning »	35
2.2.2.1 Fiche descriptive	35
2.2.2.2 Diagramme d'activité cas d'utilisation « Faire le planning »	36
2.2.2.3 Diagramme de séquence cas d'utilisation « Faire le planning »	37
2.2.3. Cas d'utilisation « Affectation des cadres médicaux »	38
2.2.3.1 Fiche descriptive	38
2.2.3.2.Diagramme d'activité cas d'utilisation « Affectation des cadres médicaux »	39
2.2.3.3.Diagramme de séquence cas d'utilisation « Affectation des cadres médicaux » ..	40
2.2.4. Cas d'utilisation « Gérer les lits »	41
2.2.4.2.Fiche descriptive	41
2.2.4.3.Diagramme d'activité cas d'utilisation « Gérer les lits »	42

2.2.4.4.	Diagramme de séquence cas d'utilisation « Gérer les lits »	42
2.2.5.	Cas d'utilisation « Admettre patient »	43
2.2.5.2.	Fiche descriptive	43
2.2.5.3.	Diagramme d'activité cas d'utilisation « Admettre patient »	44
2.2.5.4.	Diagramme de séquence cas d'utilisation « Admettre patient »	45
2.2.6.	Cas d'utilisation « Consulter fiche navette »	46
2.2.6.2.	Fiche descriptive	46
2.2.6.3.	Diagramme d'activité cas d'utilisation « Consulter fiche navette »	46
2.2.6.4.	Diagramme de séquence cas d'utilisation « Consulter fiche navette »	47
2.2.7.	Cas d'utilisation « Consulter planning »	47
2.2.7.2.	Fiche descriptive	47
2.2.7.3.	Diagramme d'activité cas d'utilisation « Consulter planning »	48
2.2.7.4.	Diagramme de séquence cas d'utilisation « Consulter planning »	48
2.2.8.	Cas d'utilisation « Prescrire patient »	49
2.2.8.2.	Fiche descriptive	49
2.2.8.3.	Diagramme d'activité cas d'utilisation « Prescrire patient »	49
2.2.8.4.	Diagramme de séquence cas d'utilisation « Prescrire patient »	50
2.2.9.	Cas d'utilisation « Décision de sortie patient »	50
2.2.9.2.	Fiche descriptive	50
2.2.9.3.	Diagramme d'activité cas d'utilisation « Décision de sortie patient »	51
2.2.9.4.	Diagramme de séquence cas d'utilisation « Décision de sortie patient »	51
2.2.10.	Cas d'utilisation « Faire le bilan »	52
2.2.10.2.	Fiche descriptive	52
2.2.10.3.	Diagramme d'activité cas d'utilisation « Faire le bilan »	52
2.2.10.4.	Diagramme de séquence cas d'utilisation « Faire le bilan »	53
2.2.11.	Cas d'utilisation « Livraison des produits »	53
2.2.11.2.	Fiche descriptive	53
2.2.11.3.	Diagramme d'activité cas d'utilisation « Livraison des produits »	54
2.2.11.4.	Diagramme de séquence cas d'utilisation « Livraison des produits »	55
2.2.12.	Cas d'utilisation « Entrer le stock »	55
2.2.12.2.	Fiche descriptive	55
2.2.12.3.	Diagramme d'activité cas d'utilisation « Entrer le stock »	56
2.2.12.4.	Diagramme de séquence cas d'utilisation « Entrer le stock »	56
2.2.13.	Cas d'utilisation « Gérer produit »	57
2.2.13.2.	Fiche descriptive	57
2.2.13.3.	Diagramme d'activité cas d'utilisation « Gérer produit »	58
2.2.13.4.	Diagramme de séquence cas d'utilisation « Gérer produit »	59
3.	Spécification des besoins techniques	59
3.1.	Capture des spécifications matérielles	60
3.1.1.	Configuration matérielle	60
3.1.2.	Spécification du style d'architecture 2 niveaux	60
3.2.	Capture des spécifications logicielles	61
3.2.1.	Identification des exploitants du système	61
3.2.2.	Identification des cas d'utilisation techniques	61

3.2.3. Description des cas d'utilisation techniques	62
3.2.3.1. Cas d'utilisation « Gérer la sécurité »	62
a. S'authentifier.....	62
b. Gérer intégrité des données.....	63
3.3. Organisation du modèle de spécification logicielle	65
3.3.1. La couche présentation ou IHM (Interface Homme/Machine)	65
3.3.2. La couche traitement.....	65
3.3.3. La couche données	65
4. Analyse.....	65
4.1. Découpage en catégories	65
4.1.1. Répartition des classes candidates en catégorie	66
4.1.2. Elaboration des diagrammes de classes préliminaires par catégorie.....	66
4.1.2.1. La classe préliminaire de la catégorie «Patient»	66
4.1.2.2. La classe préliminaire de la catégorie «Pharmacie»	67
4.1.2.3. La classe préliminaire de la catégorie «Administration»	67
4.1.3. Dépendance entre catégories	68
4.2. Développement du modèle statique	68
4.2.1. Diagramme de classe de la catégorie Patient.....	69
4.2.2. Diagramme de classe de la catégorie Pharmacie	69
4.2.3. Diagramme de classe de la catégorie Administration	70
4.3. Développement du modèle dynamique	70
4.3.1. Diagrammes d'interactions	71
5. Conception préliminaire.....	71
5.1. Développement modèle de déploiement	72
5.1.1. Architecture adoptée.....	72
5.1.2. Déploiement du modèle d'exploitation.....	72
5.2. Définition des interfaces.....	73
6. Conception détaillée	74
6.1. Dictionnaire des données.....	74
6.1.1. Les classes et les attributs	74
6.1.2. Les opérations.....	77
6.2. Diagramme de classe de système	78
6.3. Passage du diagramme de classe au modèle relationnel	79
6.3.1. Règles de passage	79
6.3.2. Les tables de la base de données.....	79
Conclusion	80
Chapitre 3 : Optimisation combinatoire et méthode de résolution	
Introduction	81
I. Généralité sur l'optimisation	81
1. Problèmes d'optimisation combinatoire	81
2. Problèmes célèbres en optimisation combinatoire	82
2.1 Problème d'affectation	82
2.2 Problème d'ordonnancement des tâches.....	82
2.3 Problème du Sac à Dos	82

2.4 Problème de Bin Packing.....	82
2.5 Problème d'Emploi du Temps.....	82
2.6 Problèmes liés à la théorie des graphes.....	83
3. Résolution d'un problème d'optimisation combinatoire	83
4. Les méthodes de résolution des problèmes d'optimisation combinatoire.....	83
4.1 Les méthodes exactes	84
4.1.1 La programmation dynamique.....	84
4.1.2 La méthode de simplexe	84
4.1.3 Branch and Bound.....	84
4.2 Les méthodes approchées	85
4.2.1 Heuristique.....	85
4.2.2 Méta-Heuristique.....	85
4.2.2.1 La recherche Tabou	85
4.2.2.2 Les colonies de fourmis.....	86
4.2.2.3 Les algorithmes génétiques.....	86
4.2.2.4 La Recuit simulé	86
II. La méthode de résolution	86
1. Choix de la méthode.....	86
2. Recuit simulé (Simulated annealing)	86
2.1.Historique	87
2.2.Principe	87
3. Les étapes de l'algorithme	88
3.1.L'algorithme de Metropolis	88
3.2.L'algorithme du recuit simulé	88
3.2.1. L'état initial de l'algorithme.....	89
3.2.2. Paramètre de la température	89
4. Avantages et inconvénients de la méthode.....	89
5. Affectation des médecins aux patients	89
5.1.Présentation du problème	89
5.2.Les données	90
5.3.Les contraintes	90
5.4.La fonction objective	90
5.5.Les étapes de notre algorithme de résolution	91
5.5.1. Etape 1 : Répartition des patients aux médecins dont les maladies chroniques correspond à la spécialité du médecin traitant	91
5.5.2. Etape 2 : Application de la méthode de recuit simulé	92
5.5.2.1.Principe de fonctionnement de l'algorithme proposé	92
5.5.2.2.Construction du voisinage	92
5.5.2.3.La fonction « performance » d'une solution	93
5.5.2.4.Le critère de Metropolis.....	93
5.5.2.5.Algorithme de Recuit Simulé.....	94
5.5.2.6.Affectation des patients aux médecins.....	94
5.5.2.7.Etude expérimentale et résultats	97
6. Affectation des infirmiers aux patients	99

7. Affectation des médicaments aux patients	99
Conclusion	101
Chapitre 4 : Implémentation	
Introduction	102
1. Présentation des outils de développement de l'application	102
1.1 Langage de programmation : JAVA	102
1.2 Netbeans	102
1.3 Implémentation de la base de données	103
2. Les interfaces de notre système.....	103
2.1 Interface authentification.....	103
2.2 Interface Administrateur.....	104
2.3 Interface gestion du personnel	104
2.4 Interface gestion du planning.....	105
2.5 Interface admission d'un patient	105
2.6 Interface consultation de la fiche navette	106
2.7 Interface gestion de prescription	106
2.8 Interface gestion des soins médecins	107
2.9 Interface gestion des bilans	107
2.10Gestion de la pharmacie	108
Conclusion	108
Conclusion générale	109
Références et bibliographiques	110

Table des figures

Chapitre	N° et nom de figure	Page
1	1.1 : L'entrée d'hôpital FRERES TOBAL- Mila.	8
	1.2 : La situation de l'hôpital FRERES TOBAL-Mila.	8
	1.3 : L'Organigramme de l'hôpital FRERE TOBAL Mila.	9
	1.4 : Organigramme d'infrastructure hospitalière.	9
	1.5 : les différents services hospitaliers de l'établissement.	11
	1.6 : Les Secteurs de répartition du personnel dans l'hôpital.	11
	1.7 : Le parcours de soins COVOD-19.	15
2	2.1 : Le système d'information soumis à deux types de contraintes.	18
	2.2 : Le processus de développement en Y.	18
	2.3 : Situation de l'étude préliminaire dans 2TUP.	21
	2.4 : Diagramme de contexte dynamique.	27
	2.5 : Diagramme de cas d'utilisation.	31
	2.6 : Diagramme d'activité de cas d'utilisation «Gérer les comptes du personnels ».	33
	2.7 : Diagramme de séquence de cas d'utilisation «Gérer les comptes du personnels ».	34
	2.8 : Diagramme d'activité de cas d'utilisation «Faire planning».	36
	2.9 : Diagramme de séquence de cas d'utilisation «Faire planning».	37
	2.10 : Diagramme d'activité de cas d'utilisation «Affecter les cadres médicaux ».	39
	2.11 : Diagramme de séquence de cas d'utilisation «Affecter les cadres médicaux ».	40
	2.12 : Diagramme d'activité de cas d'utilisation «Gérer les lits».	42
	2.13 : Diagramme séquence de cas d'utilisation «Gérer les lits».	42
	2.14 : Diagramme d'activité de cas d'utilisation «Admettre patient».	44
	2.15 : Diagramme de séquence de cas d'utilisation «Admettre patient».	45
	2.16 : Diagramme d'activité de cas d'utilisation «Consulter fiche navette».	46
	2.17 : Diagramme de séquence de cas d'utilisation «Consulter fiche navette».	47
	2.18 : Diagramme d'activité de cas d'utilisation «Consulter planning».	48
	2.19 : Diagramme de séquence de cas d'utilisation «Consulter planning».	48
	2.20 : Diagramme d'activité de cas d'utilisation «Prescrire patient».	49
	2.21 : Diagramme de séquence de cas d'utilisation «Prescrire patient».	50
	2.22 : Diagramme d'activité de cas d'utilisation «Décision de sortie patient».	51
	2.23 : Diagramme de séquence de cas d'utilisation «Décision de sortie patient ».	51
	2.24 : Diagramme d'activité de cas d'utilisation «Faire le bilan».	52
	2.25 : Diagramme de séquence de cas d'utilisation «Faire le bilan».	53
	2.26 : Diagramme d'activité de cas d'utilisation «Livraison des produits».	54
	2.27 : Diagramme de séquence de cas d'utilisation «Livraison des produits».	55
	2.28 : Diagramme d'activité de cas d'utilisation «Entré stock».	56
	2.29 : Diagramme de séquence de cas d'utilisation «Entré stock».	56
	2.30 : Diagramme d'activité de cas d'utilisation «Gérer produits».	58
	2.31 : Diagramme de séquence de d'utilisation «Gérer produits».	59
	2.32 : L'architecture 2-tiers.	60

	2.33 : Modèle de spécification logicielle du système.	61
	2.34 : Diagramme d'activité de cas « S'authentifier ».	62
	2.35 : Diagramme de séquence de cas « S'authentifier ».	63
	2.36 : Diagramme d'activité de cas « Gérer l'intégrité des données ».	64
	2.37 : Diagramme de séquence de cas « Gérer l'intégrité des données ».	64
	2.38 : Découpage en catégories.	66
	2.39 : Découpage de la classe de la « catégorie Patient ».	66
	2.40 : Découpage de la classe de la « catégorie Pharmacie ».	67
	2.41 : Découpage de la classe de la « catégorie Administration ».	67
	2.42 : Dépendance entre catégories.	68
	2.43 : Diagramme de classe de la catégorie « Patient ».	69
	2.44 : Diagramme de classe de la catégorie « Pharmacie ».	69
	2.45 : Diagramme de classe de la catégorie « Administration ».	70
	2.46 : Diagramme d'interaction de cas d'utilisation « Admettre patient ».	71
	2.47 : Schéma du modèle de déploiement de notre système.	72
	2.48 : Définition des applications dans le modèle d'exploitation.	73
	2.49 : Diagramme de classe de système.	78
3	3.1 : L'optimisation combinatoire.	81
	3.2 : Classification des méthodes d'optimisation combinatoire.	83
	3.3 : Description recuit simulé.	87
	3.4 : Comparaison entre le recuit simulé et une heuristique classique.	88
	3.5 : Les étapes pour l'affectation des médecins aux patients.	91
	3.6 : Schéma du déroulement de l'affectation des patients aux médecins.	92
	3.7 : Pseudo code de l'algorithme de voisinage.	92
	3.8 : Pseudo code de la performance.	93
	3.9 : Pseudo code de l'algorithme de recuit simulé.	94
	3.10 : Pseudo code affectation des patients aux médecins.	95
	3.11 : L'organigramme de la solution proposée.	96
	3.12 : Capture d'écran des résultats obtenus.	97
	3.13 : La différence entre la performance des affectations des deux méthodes coté médecin.	98
	3.14 : Différence entre la performance des affectations des deux méthodes coté patient.	98
	3.15 : Affectation des infirmiers aux les patients.	99
	3.16 : Le pseudo code de la distribution des médicaments aux patients.	100
	3.17 : Récupération de la liste des patients demandant d'un médicament.	100
	3.18 : Résultat de l'affectation des médicaments aux patients.	101
4	4.1 : Capture interface authentification.	103
	4.2 : Capture interface administrateur.	104
	4.3 : Capture interface gestion du personnel.	104
	4.4 : Capture interface gestion du planning.	105
	4.5 : Capture interface admission d'un patient.	105
	4.6 : Capture interface consultation de la fiche navette.	106
	4.7 : Capture interface gestion de prescription.	106
	4.8 : Capture interface gestion des soins médecins.	107
	4.9 : Capture interface gestion des bilans.	107
	4.10 : Capture interface gestion de la pharmacie.	108

Table des tableaux

Chapitre	N° et nom de tableau	Page
1	1.1 : Etat du Personnel service COVID-19.	12
	1.2 : Etat du matériel médical.	12
	1.3 : Statistique des patients admis et décès pendant année 2021.	12
	1.4 : Les indications des examens selon la présentation clinique.	13
	1.5 : Le traitement des patients COVID-19	14
2	2.1 : Les rôles de chaque acteur.	26
	2.2 : Signification des messages.	27
	2.3 : Liste des acteurs et des messages de chaque cas d'utilisation du système.	29
	2.4 : Fiche descriptive textuelle de cas d'utilisation « Gérer les comptes personnels ».	32
	2.5 : Fiche descriptive textuelle de cas d'utilisation « Faire planning ».	35
	2.6 : Fiche descriptive textuelle de cas d'utilisation « Affecter les cadres médicaux ».	38
	2.7 : Fiche descriptive textuelle de cas d'utilisation « Gérer les lits ».	41
	2.8 : Fiche descriptive textuelle de cas d'utilisation « Admettre patient ».	43
	2.9 : Fiche descriptive textuelle de cas d'utilisation « Consulter fiche navette ».	46
	2.10: Fiche descriptive textuelle de cas d'utilisation « Consulter planning ».	47
	2.11: Fiche descriptive textuelle de cas d'utilisation « Prescrire patient ».	49
	2.12 : Fiche descriptive textuelle de cas d'utilisation « Décision de sortir patient ».	50
	2.13 : Fiche descriptive textuelle de cas d'utilisation « Faire le bilan ».	52
	2.14 : Fiche descriptive textuelle de cas d'utilisation « Livraison des produits ».	53
	2.15: Fiche descriptive textuelle de cas d'utilisation « Entré stock ».	55
	2.16: Fiche descriptive textuelle de cas d'utilisation « Gérer produits ».	57
	2.17 : Fiche descriptive textuelle de cas d'utilisation « S'authentifier ».	62
	2.18 : Fiche descriptive textuelle de cas d'utilisation « Gérer intégrité des données ».	63
	2.19 : Les interfaces de notre système.	73
	2.20 : Tableau des classes et des attribues.	76
	2.21 : Tableau des opérations.	77
	2.22 : Table de la base de données.	80
3	3.1 : Tableau approximatif de niveau de compatibilité entre la spécialité de médecin et la maladie.	90

Introduction général

Introduction général

La propagation de la pandémie de COVID-19 a rapidement plongé la majorité des systèmes de santé à travers le monde dans des situations critiques de crise sanitaire. Le système national de santé Algérien ne fait pas l'exception, la situation étant inédite, les défis à relever sont importants. Certes, l'épreuve est dure même pour les pays les plus développés tels que la France, l'Italie, l'Espagne ou même les Etats Unis. Ces pays se sont retrouvés sous tension face à l'ampleur de la pandémie.

L'augmentation rapide des cas positifs à partir du mois de juillet 2020 en Algérie a mis le système de santé national devant une épreuve rude. Cette crise sanitaire a clairement dévoilé la vulnérabilité de notre système et a créé une saturation dans les services spécialisés dans les hôpitaux, cela conduit à une déstabilisation dans l'utilisation des moyens disponibles surtout l'affectation du cadre médical et la distribution des médicaments, ce qui augmente le taux de risque sur la vie des patients.

Les hôpitaux sont parmi les établissements qui ont besoin d'un système informatique pour bien conduire leur travail et assure une bonne gestion des ressources dans le but de garantir une meilleure hospitalisation.

Dans ce cadre, nous allons proposer de mettre en place un système de gestion optimisé pour le service de Coronavirus au niveau de l'établissement public hospitalier FRERE TOBAL-Mila afin de pallier aux problèmes concernant le partage des ressources disponibles dans l'hôpital.

La recherche opérationnelle est connue pour fournir des outils informatiques d'aide à la décision qui permet une bonne gestion des ressources dans différent secteur. L'optimisation combinatoire occupe une place très importante en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Nous nous sommes intéressés à l'utilisation d'une technique d'optimisation qui est le recuit simulé pour améliorer nos résultats.

Ce mémoire est organisé en quatre chapitres:

Dans le premier chapitre on va présenter le domaine d'étude qui est en fait une description brève du système sanitaire pendant la pandémie de COVID-19 en Algérie, et de donner une présentation de l'établissement public hospitalier (EPH) FERERS TOBAL-Mila où nous avons effectué notre stage.

Dans le deuxième chapitre on va présenter l'analyse et la conception de notre système après une brève description de l'approche de modélisation suivi.

Le troisième chapitre sera consacré à la partie optimisation de notre système, après une présentation de l'optimisation combinatoire et les méthodes de résolution de ses problèmes, on va détailler l'algorithme proposé et les résultats obtenus.

Dans le dernier chapitre on va présenter les différents outils de programmation utilisés pour l'implémentation et la réalisation de notre application suivi d'une présentation de quelques interfaces et fenêtres de notre système.

Nous terminons notre mémoire par une conclusion générale.

Chapitre 1

COVID-19
et le système sanitaire
Algérien

Introduction

Le COVID-19 est une maladie infectieuse avec une caractéristique de propagation élevée, celle-ci a clairement dévoilé la vulnérabilité du système de santé algérien et révèle l'urgence de le réformer, pour son renforcement.

I. Le système sanitaire en Algérie

1. La création de système sanitaire en Algérie

Après l'indépendance, l'Algérie a cherché à essayer l'établissement d'un système de santé équitable et intégré pour tous les Algériens, car ce système national de santé a connu plusieurs réformes et changements, à la fois en termes de structures de santé de toutes sortes, et en termes de nombre de personnes employées, mais ils connaissent maintenant plusieurs contradictions qui ont empêché la réalisation des buts et objectifs qui existaient pour eux [1].

2. Crises du secteur de la santé

Le système de santé Algérien est totalement déconnecté des besoins réels de la population, et dispose d'une faible capacité pour répondre à une crise sanitaire. Et selon le rapport sur l'indice de sécurité sanitaire mondiale, le système de santé algérien est classé parmi ceux des pays les moins préparés à faire face aux épidémies, y compris le COVID-19.

Selon le rapport, les analystes soulignent « la difficulté des pouvoirs publics algériens à définir et élaborer un ensemble de décisions, programmes et actions homogènes et rapides, visant à protéger la population contre tous les dangers et risques pour la santé échappant au contrôle des individus et la nécessité de déployer des mesures de modernisation et une amélioration en toute urgence, du système sanitaire en place qui a démontré son incapacité à répondre aux nouveaux défis sanitaires » [2].

3. Problématique

3.1. Définition

La maladie **COVID-19** baptisée par l'OMS en Février 2020- fait référence à "**CO**rona**VI**rus**D**isease**2019**", est une maladie infectieuse respiratoire qui est caractérisée par des pneumonies sévères pouvant être mortelles causées par le virus **Sars-CoV-2** apparue à Wuhan en Chine en 2019 [3].

Cette maladie aurait été transmise à l'homme suite à la consommation de viandes contaminées (chauve-souris ou pangolin) vendues sur un marché local de Wuhan en Chine. Après ce virus s'est propagé dans le monde entier pour se transformé à une pandémie mondiale- active jusqu'à nos jours.

3.2. Symptômes

La maladie COVID-19 affecte beaucoup plus les adultes qui présenteront des symptômes pouvant aller d'un degré modéré de danger comme la fièvre ,la toux, la fatigue, les courbatures et les douleurs, la diarrhée et la perte du goût ou de l'odorat à un degré plus élevé comme les difficultés respiratoires ou l'essoufflement, perte de la parole, difficultés à se déplacer ou confusion.

Dans certain cas cette maladie deviens fatale spécialement pour les personnes âgées ou souffrant de maladies chroniques tel que le diabète, l'hypertension artérielle, les maladies respiratoires... parce qu'elle provoque des insuffisances respiratoires nécessitant un recours à des moyen de respiration artificielle au sein du service de réanimation.

Il faut noter aussi qu'il y a des cas asymptomatiques, c'est-à-dire que les patients n'ont aucun symptôme apparent malgré la détection du virus.

3.3. Mode de propagation

Le virus peut se propager par l'intermédiaire des gouttelettes de salive ou de sécrétions nasales émises par une personne infectée quand elle tousse, éternue, parle, chante ou respire. Il est donc important d'appliquer les règles d'hygiène respiratoire.

3.4. Le traitement

Il n'existe pour le moment aucun traitement contre ce virus. Les soins prescrits aux patients sont uniquement destinés à traiter les symptômes. Les antibiotiques sont inefficaces contre les infections virales, tout comme certains remèdes traditionnels à base de plantes ou d'aliments.

Dans environ 80 % des cas, les patients guérissent spontanément, sans avoir besoin de traitement particulier. Les cas les plus graves sont pris en charge dans des unités de soins intensifs à l'hôpital où ils font l'objet d'une surveillance étroite.

3.5. Prévention

Pour prévenir la contraction et la propagation de ce virus, l'OMS a publié plusieurs précautions qui doivent être présent tel que :

- Port de masque ou de bavette chirurgicale surtout dans les lieux publics, ou en cas de suspicion de contamination;
- le lavage fréquent des mains au savon ou avec une solution hydro alcoolique;
- éviter les contacts rapprochés, comme faire la bise ou serrer la main;
- se couvrir la bouche avec le pli du coude, ou un mouchoir jetable, lors d'une toux ou d'un éternuement;
- ne pas se toucher les yeux, le nez ou la bouche;
- en cas de symptômes respiratoires et de fièvre, portez un masque et restez confinés pour ne pas contaminer votre entourage. Se faire consulter par un médecin et suivre ses instructions.

4. Enseignements tirés de la pandémie COVID-19

Certes, un fléau de cette ampleur a touché le monde entier, y compris l'Algérie, qui a sans doute tiré des leçons pour l'avenir. Examinons de plus près les principales leçons tirées de la pandémie de COVID-19 pour le secteur de la santé [1].

4.1. Un système de santé fortement centralisé

L'institut Pasteur d'Alger était le seul habilité à effectuer le test de diagnostic spécifique «PCR». Mais, il demeure incapable à lui seul de répondre aux besoins de dépistage croissants. Par conséquent, plusieurs régions se trouvaient dans l'incapacité à mener des analyses pour confirmer ou non des cas suspects. Cette situation a poussé les autorités responsables à prendre la décision de créer des annexes de l'institut dans d'autres wilayas (ex : Oran, Constantine, Boumerdes, Ouargla, Tamanrasset, Béchar, etc...) Egalement, plusieurs laboratoires d'analyse et de dépistage du Coronavirus ont été créés au niveau des universités, à l'exemple de l'université d'Alger, Mostaganem, Tlemcen, et de Tizi Ouzou.

4.2. La difficulté des conditions de travail

Depuis le début de la pandémie, les professionnels de la santé publique se sont engagés à remédier à la situation grave. Tout le personnel est mobilisé pour faire face à une crise sanitaire sans précédent, malgré le manque de protection, d'outils de diagnostic, de gestion des cas diagnostiques, ainsi que de moyens et d'équipements nécessaires à leur activité. Sans oublier l'agression verbale et/ou physique qu'ils subissent de la part des patients et de leurs proches. Bien qu'elles continuent de fonctionner dans ces circonstances inadéquates, la situation continue de se détériorer depuis plusieurs années.

4.3. Absence des médecins libéraux

Si les "combattants en blouses blanches" du secteur public ont fortement fait preuve d'humanité et de professionnalisme, depuis le début de l'épidémie en Algérie, les médecins du secteur privé ont eux aussi contribué à prendre en charge les patients. Toutefois, des médecins ont préféré fermer leurs cabinets malgré une obligation du gouvernement de maintenir leur activité sous peine de poursuites pénales et de sanctions administratives de retrait immédiat et définitif des titres légaux d'exercice de l'activité.

4.4. La téléconsultation comme solution de rechange

En substitution aux consultations dans les cabinets médicaux, et dans le but de désengorger les hôpitaux et les cabinets qui représentent un milieu favorable à la propagation du virus, les médecins du secteur libéral ont essayé de trouver une alternative; la téléconsultation. De fait, ils ont opté pour les moyens de vidéotransmission sur les réseaux sociaux. D'autres médecins ont choisi de joindre des plateformes de téléconsultations en ligne et se sont prêtés volontiers au devoir de répondre aux préoccupations de leurs patients. Ainsi, en collaboration avec une communauté de médecins volontaires, a mis à la disposition des citoyens, des spécialistes pour répondre à leurs questions en ligne via une plateforme regroupant des médecins généralistes et des spécialistes.

4.5. Des taux de remboursement complètement obsolètes

Devant le nombre de nouveaux cas enregistrés chaque jour et le manque de tests de dépistage, et afin de pouvoir entamer le traitement par les personnes présentant des symptômes de contamination par la COVID-19, avant que leur état de santé se détériore, le ministère de la santé a autorisé les spécialistes de recourir à l'utilisation du scanner thoracique comme alternative au dépistage biologique dans les hôpitaux publics, mais reste insuffisant au vu du nombre des demandeurs, avec des délais d'attente souvent longs. Dans les cliniques privées et les centres d'imagerie médicale, l'accès est plus facile, toutefois les tarifs ne sont pas à la portée de tous, surtout que le taux de remboursement par la sécurité sociale de cet acte est dérisoire. Certaines cliniques privées se sont portées volontaires pour offrir leurs services de scanner thoracique à titre gratuit, mais ça reste une réponse temporaire et insuffisante.

4.6. Pénurie de certains médicaments

Après le développement de la pandémie de COVID-19, l'État algérien a pris des mesures de confinement et a décidé de fermer ses frontières. En conséquence, le marché des

médicaments d'ordonnance a souffert de perturbations dans l'approvisionnement de certains médicaments contre les maladies chroniques tels que la "chloroquine", une molécule indispensable pour traiter certains systèmes inflammatoires et auto-immuns, ce qui est très nécessaire dans le protocole de traitement des patients atteints de la COVID-19, depuis l'écllosion de la pandémie.

II. Présentation de l'établissement public hospitalier

1. Organisme d'accueil

1.1. Définition et création

L'établissement public hospitalier FRERE TOBAL Mila est un établissement public à caractère administratif, créé par le décret exécutif n° 07-140 du 2 Joumada El Oula 1428 correspondant au 19 mai 2007, modifié et complète, portant création, organisation et fonctionnement des Etablissements publics hospitaliers et des Etablissements publics de santé de proximité [4].

Doté de la personnalité morale et de l'autonomie financière. Il est placé sous la tutelle du wali Il est constitué d'une structure de diagnostic, de soins, d'hospitalisation et de réadaptation médicale couvrant la population d'une ou d'un ensemble de communes.

1.2. Classification

EPH FRERE TOBAL Mila est classé en catégorie C selon l'arrêté interministériel du 21 Safar 1433 correspondant au 15 janvier 2012, modifié et complète, fixant les critères de classification des Etablissements publics hospitaliers et des Etablissements publics de santé de proximité ainsi que leur classement [5].

1.3. Géographie et position de l'établissement hospitalier EPH de Mila

L'établissement hospitalier public FRERE TOBAL Mila est au cœur de la ville à côté du tribunal, de sorte que la zone bâtie est estimée à 1783 et la zone non bâtie à 24748. Les figures suivantes montrent la situation de l'établissement dans la ville de Mila :



Figure 1.1 : l'entrée d'hôpital FRERES TOBAL- Mila.

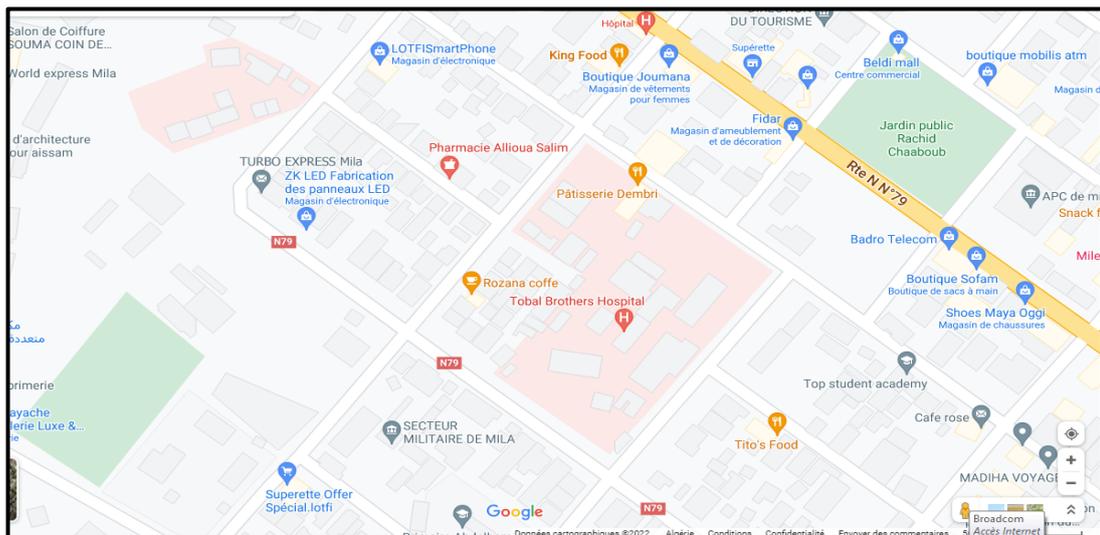


Figure 1.2 : La situation de l'hôpital FRERES TOBAL-Mila dans la ville.

1.4. Missions de l'organisme d'accueil

Ces tâches se résument en la prise en charge des besoins sanitaires de la population à savoir :

- Assurer l'organisation et la programmation de la distribution des soins curatifs, de diagnostic, de réadaptation médicale et d'hospitalisation.
- Appliquer les programmes nationaux de santé.
- Assurer l'hygiène, la salubrité, et la lutte contre les fléaux sociaux.
- Assurer le perfectionnement et le recyclage des personnels des services de la santé, l'EPH peut servir de terrain de formation médicale et paramédicale et en gestion

hospitalière sur la base de conventions signées avec des établissements de formation [4].

1.5. L'organigramme de l'EPH

L'organigramme de l'EPH FRERE TOBAL à sa tête le directeur qui représente le chef hiérarchique de tout le personnel, il est à l'origine de toutes les décisions majeurs au sein de cet EPH, le bureau d'ordre général et de communication et aussi un nouveau bureau qui est un centre de calcul sont rattachés directement en direction générale dans le but de mettre l'équilibre entre les différentes sous directions (figure 3) , selon les branches de cet EPH, il existe des sous directions qui sont gérées par des sous directeurs (figure 4), on peut les citer dans les organigrammes suivants [6]:

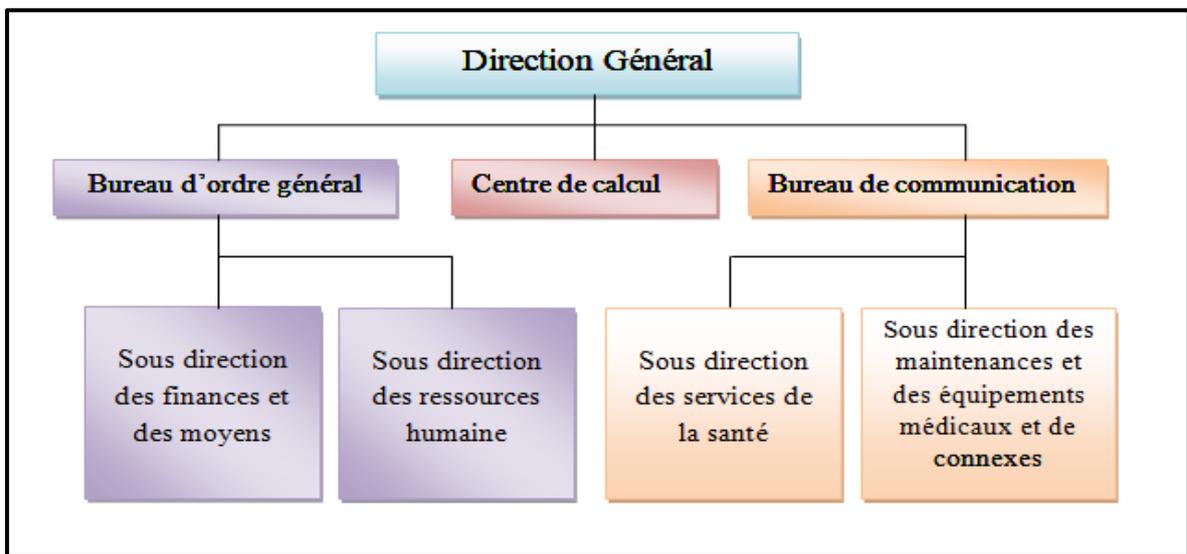


Figure 1.3 : L'Organigramme de l'hôpital FRERE TOBAL Mila.

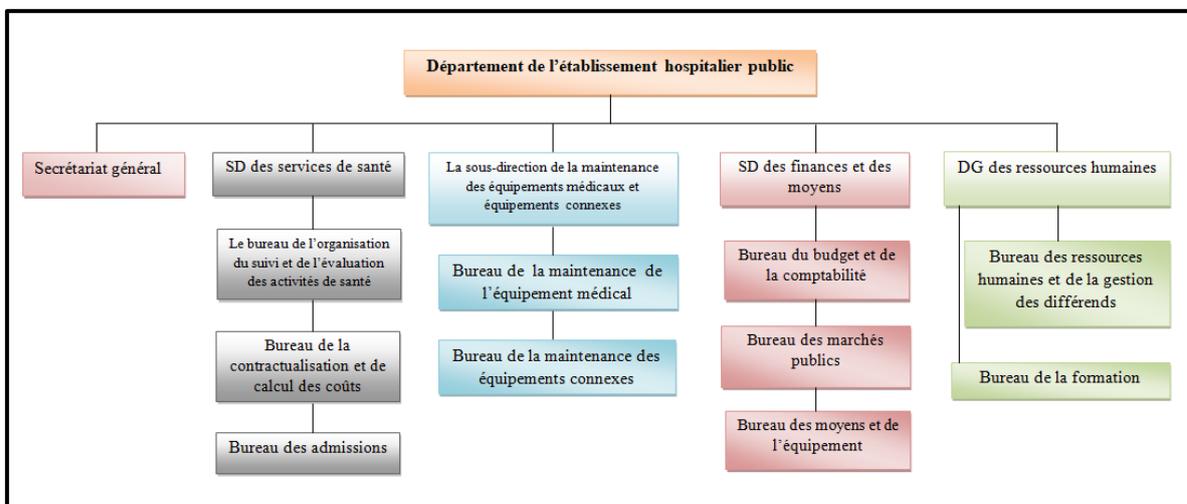


Figure 1.4: Organigramme d'infrastructure hospitalière.

A noter aussi que chaque sous-direction est répartie en bureaux de travail et cela selon la nature et le type de tâches effectuées dans ces derniers comme montré dans la figure précédente de gauche à droite :

- ❖ **Secrétaire général.**
- ❖ **La sous-direction des services de santé:** trois (3) bureaux constituent cette sous-direction:
 - Bureau de l'organisation du suivi et de l'évaluation des activités de santé ;
 - Bureau de la contractualisation et de calcul des coûts ;
 - Bureau des admissions.
- ❖ **La sous-direction de la maintenance des équipements médicaux et équipements connexes:** qui est composée de deux (2) bureaux;
 - Bureau de la maintenance de l'équipement médical;
 - Bureau de la maintenance des équipements connexes.
- ❖ **La sous-direction des finances et des moyens:** elle se divise en trois (3) bureaux qui sont :
 - Bureau du budget et de la comptabilité;
 - Bureau des marchés publics;
 - Bureau des moyens et des équipements.
- ❖ **La sous-direction des ressources humaines:** elle est composée que de deux (2) bureaux qui sont:
 - Bureau de la gestion des ressources humaines;
 - Bureau de la formation.

La création des services et de leurs unités constitutives au sein de l'établissement (Arrêté ministériel N° 2825 du 03 Mars 2008), la figure suivante présente les différents services hospitaliers qui existent dans l'EPH :

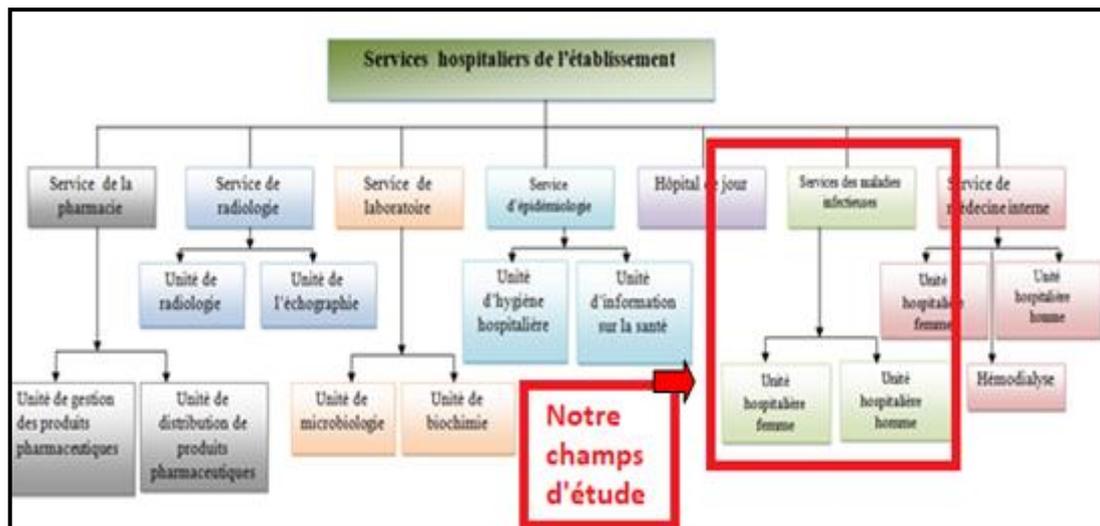


Figure 1.5 : Les différents services hospitaliers de l'établissement.

1.6. Etat global du personnel de l'hôpital

La figure ci-dessous montre la répartition du personnel dans l'établissement hospitalier FRERE TOBAL Mila :

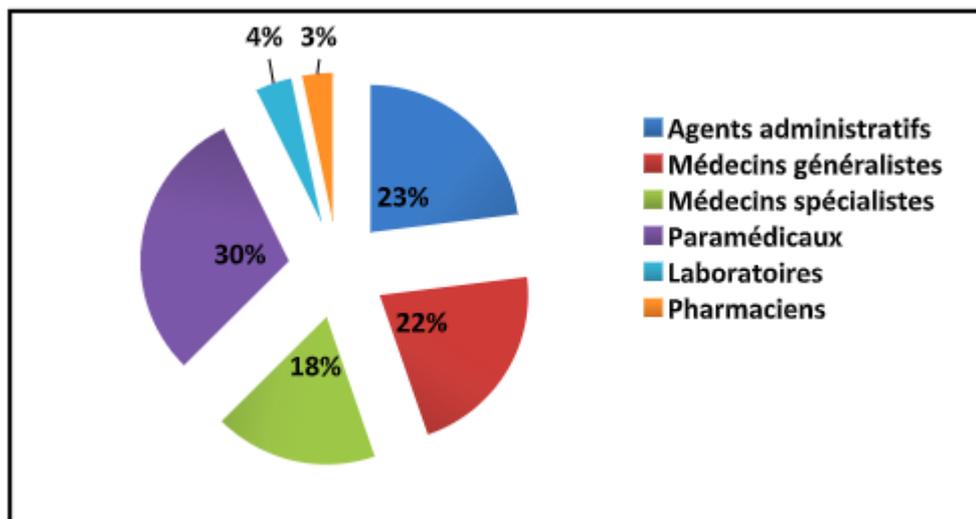


Figure 1.6 : Les Secteurs de répartition du personnel dans l'hôpital.

1.7. Les Ressources humaines et matériaux

Le totale des personnels qui travaillent dans l'EPH sont repartie comme suit :

Personnel	Nombre
Agents administratifs	35
Médecins généralistes	33
Médecins spécialistes	27

Chapitre 1 : COVID-19 et le système sanitaire Algérien

Paramédicaux	46
Laboratoires	06
Pharmaciens	05

Tableau 1.1 : Etat du Personnel service COVID-19.

Matériel médicaux	Nombre	Capacité
Lits	40	06 Réanimation
		34 Normal
Réservoir d'oxygène	1 Central	5500 Litre

Tableau 1.2 : Etat du matériel médical.

2. Service de COVID-19

Le service de coronavirus fait partie des nouveaux services qui se sont créés dans les hôpitaux, et ce depuis l'émergence de la pandémie de COVID 19. Le secteur de la santé a joué un rôle primordial dans la lutte contre la pandémie émergente de Corona, car il était dans les premières lignes de défense telle qu'elle était.

Le nombre de patients infectés (cas confirmé) qui arrivent au service COVID à l'hôpital dans la période de **janvier à décembre 2021** est :

Type	Nombre
Patients Admis	802
Patients Décès	182

Tableau 1.3 : Statistique des patients admis et décès pendant année 2021.

2.1. Prise en charge des patients COVID-19

En vue d'organiser la prise en charge de l'épidémie de COVID-19 au niveau de l'EPH, un protocole a été rédigé, basé sur les instructions du ministère de la santé et de la réforme hospitalière et les données de la littérature scientifique internationale notamment celles de l'organisation mondiale de la santé.

C'est un protocole qui sera mis à jour régulièrement en fonction de l'évolution de la situation. L'objectif actuel est la prise en charge des cas, et de l'atténuation de la propagation du virus [5].

2.2. Protocole de prise en charge des patients

Le but principal est la prise en charge précoce des cas COVID-19. Cinq objectifs :

1. Repérer précocement les cas suspects parmi les consultants symptomatiques ;
2. Etablir le diagnostic en tenant compte des éléments cliniques, radiologiques, microbiologiques ;
3. Classer en formes asymptomatique ou bénignes, modérées, et graves en fonction des signes cliniques, imagerie ;
4. Décider de la prise en charge ambulatoire ou en hospitalisation conventionnelle selon le degré de gravité (cliniques et imagerie) ;
5. Confirmer par la PCR à partir de la forme bénigne [5];

2.3. Indications des examens selon la présentation clinique [9]

Indications des examens selon la présentation clinique	Tableau clinique	Examens à faire
Forme bénigne	1. signes cliniques, 2. sans facteurs, 3. de risque, 4. sans signes de gravité, 5. SpO ₂ ≥ 95%.	TDM > 30%. +PCR Positif
Forme modérée	1. signes cliniques, 2. avec au moins un facteur de risque, 3. pas de signes de gravité. 4. SpO ₂ < 92%.	TDM 30 - 70 % +PCR Positif
Forme grave	1. au moins un signe de gravité, 2. SpO ₂ < 80 % ne répondant pas à l'oxygénothérapie (15L), 3. Quelle que soit l'étendue.	TDM + 80% ou quelle que soit l'étendue + PCR Positif

Tableau 1.4 : Les indications des examens selon la présentation clinique.

3. Le parcours d'hospitalisation

3.1. Au niveau d'accueil : la consultation COVID-19

Une première consultation doit être effectuée pour confirmer l'état du patient, avec les résultats des analyses effectuées précédemment (TDM, PCR, et sérologie).

3.2. A l'admission

Après l'acceptation du patient, une fiche d'admission va être remplie, elle contient toutes les informations nécessaires à l'hospitalisation.

3.3. Au cours de l'hospitalisation

Une consultation périodique sera effectuée par les médecins suivant le protocole proposé par le ministère de la santé et les infirmiers assurent l'application de ce dernier. [8]

Le détail de ce protocole est présenté dans le tableau suivant :

Traitement	Indications	Modalités
Hydroxychloroquine		Hydroxychloroquine Cp 200mg X3 de 1 ^{er} jour à dixième jours
Azithromycine		Azithromycine 250 mg de 2 ^{ème} jour à 5 jours
Anticoagulants	Doses préventives et curatives	Lovenox 0,4 UI/J Si BMI < 30 /24h Lovenox 0,6 UI/J Si BMI > 30 /24h Lovenox 0,8 UI/J Si BMI > 40 /24h
Oxygénothérapie	SpO2 < 95%	/
Corticoïdes	Aggravation	Dexaméthasone 6 mg/kg/J

Tableau 1.5 : Le traitement des patients COVID-19.

3.4. Sortie

Au cours de l'hospitalisation, des prélèvements périodiques seront effectués pour définir la situation sanitaire de patient, une fois le niveau d'oxygène est stable (SPO2 > 95%) et les résultats d'analyses donnent un PCR et sérologie négatifs, le médecin peut décider la fin d'hospitalisation de ce patient, la figure suivante récapitule ce parcours.

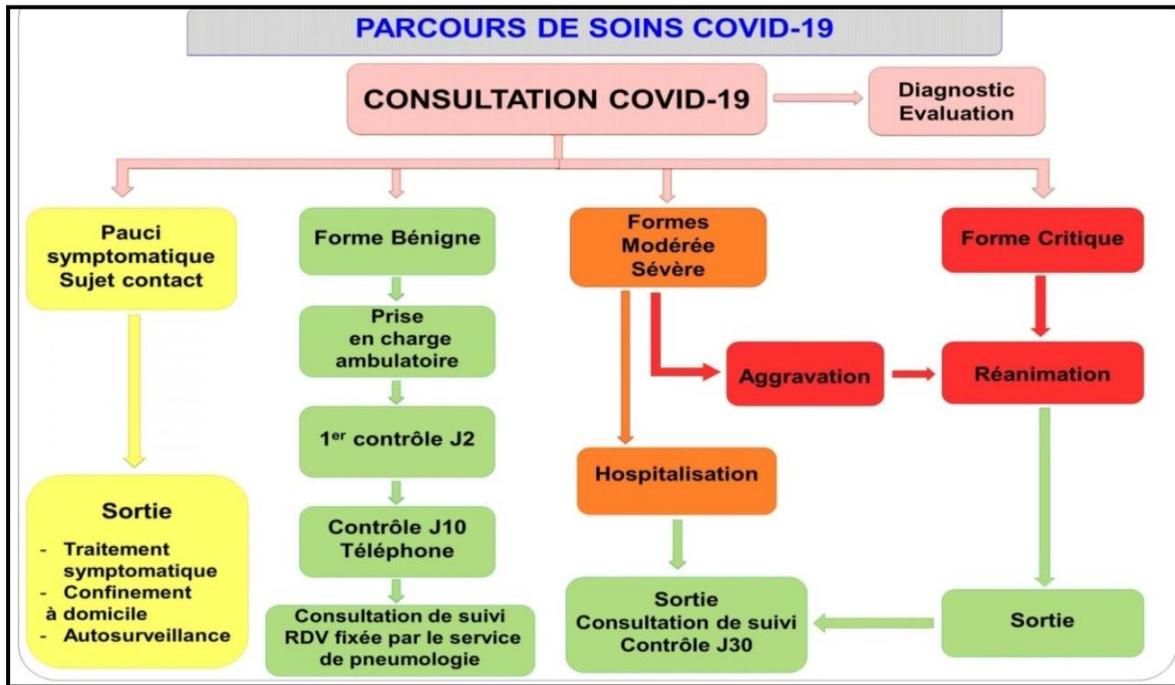


Figure 1.7 : Le parcours de soins COVID-19

Conclusion

Dans ce chapitre on a mis l'accent sur la situation sanitaire algérienne à travers l'établissement public hospitalier FRERES TOBAL – Mila lieu où on a effectué notre stage pratique pour lequel on va concevoir notre système de gestion.

Dans le chapitre suivant on va présenter la méthodologie de développement adoptée pour la réalisation de notre système.

Chapitre 2

*La méthodologie de
développement 2TUP*

Introduction

Le terme développement logiciel représente tous les processus impliqués collectivement dans la création de programmes et incarnant chacun une des phases du cycle de développement des systèmes.

Nous allons suivre la méthode 2TUP pour le développement de notre système, ce chapitre est structuré selon les phases de cette méthode.

I. La méthodologie de développement 2TUP

1. Définition d'un processus de développement logiciel

1.1. Le Processus unifié

Le Processus unifié (PU ou UP en anglais pour Unified Process) est une méthode de prise en charge du cycle de vie d'un logiciel orienté objets. C'est une méthode générique, itérative et incrémentale. Il vient compléter la systémique des modèles UML. Elle est le résultat final d'une évolution.

- **Itérative et incrémentale:** la méthode est itérative dans le sens où elle propose de faire des itérations lors de ses différentes phases, ceci garantit que le modèle construit à chaque phase ou étape soit affiné et amélioré. Chaque itération peut servir aussi à ajouter de nouveaux incréments.
- **Conduite par les cas d'utilisation :** elle est orientée utilisateur pour répondre aux besoins de celui-ci.
- **Centrée sur l'architecture :** les modèles définissent tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide.
- **Pilotée par les risques :** en définissant des priorités pour chaque fonctionnalité, on peut minimiser les risques d'échec du projet.

La gestion d'un tel processus est organisée en quatre (04) phases :

- **La pré-étude :** c'est ici qu'on évalue la valeur ajoutée du développement et la capacité technique à le réaliser (étude de faisabilité).
- **L'élaboration :** sert à confirmer l'adéquation du système aux besoins des utilisateurs et à livrer l'architecture de base.
- **La construction :** sert à livrer progressivement toutes les fonctions du système.
- **La transition :** déployer le système sur des sites opérationnels.

Chapitre 2 : La méthodologie de développement 2TUP

Chaque phase est elle-même décomposée séquentiellement en itérations limitées par le temps. Le résultat de chacune d'elles est un système testé, intégré et exécutable. L'approche itérative est fondée sur la croissance et l'affinement successifs d'un système par le biais d'itérations multiples. Le système croît avec le temps de façon incrémentale, itération par itération, et c'est pourquoi cette méthode porte également le nom de développement itératif et incrémental. Il s'agit là du principe le plus important du Processus Unifié.

Ces activités de développement sont définies par 6 disciplines fondamentales qui décrivent la capture des besoins, la modélisation métier, l'analyse et la conception, l'implémentation, le test et le déploiement.

Notons que ces différentes étapes (ou disciplines) peuvent se dérouler à travers plusieurs phases.

Le processus unifié doit donc être compris comme une trame commune des meilleures pratiques de développement.

1.2.Le Processus 2TUP

On dit de la méthode UP qu'elle est générique c.à.d. qu'elle définit un certain nombre de critères de développement, que chaque société peut par la suite personnaliser afin de créer son propre processus plus adapté à ses besoins. **2TUP** signifie « 2 Track Unified Process»; c'est un processus qui répond aux caractéristiques du **Processus Unifié**. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes [11].

« 2 Track» signifie littéralement que le processus suit deux chemins. Il s'agit du « chemin fonctionnel » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.

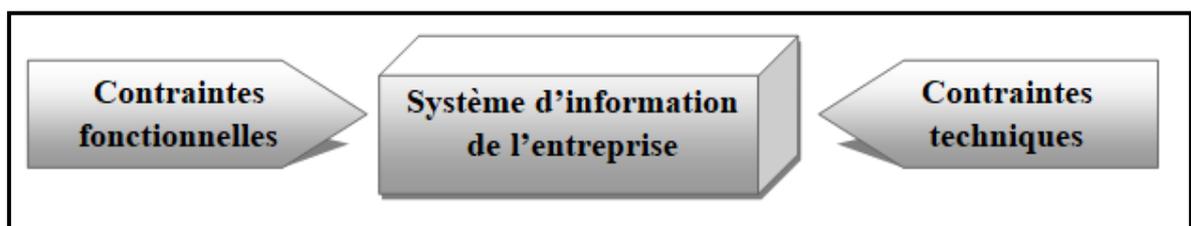


Figure 2.1: Le système d'information soumis à deux types de contraintes.

A 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 suivante.

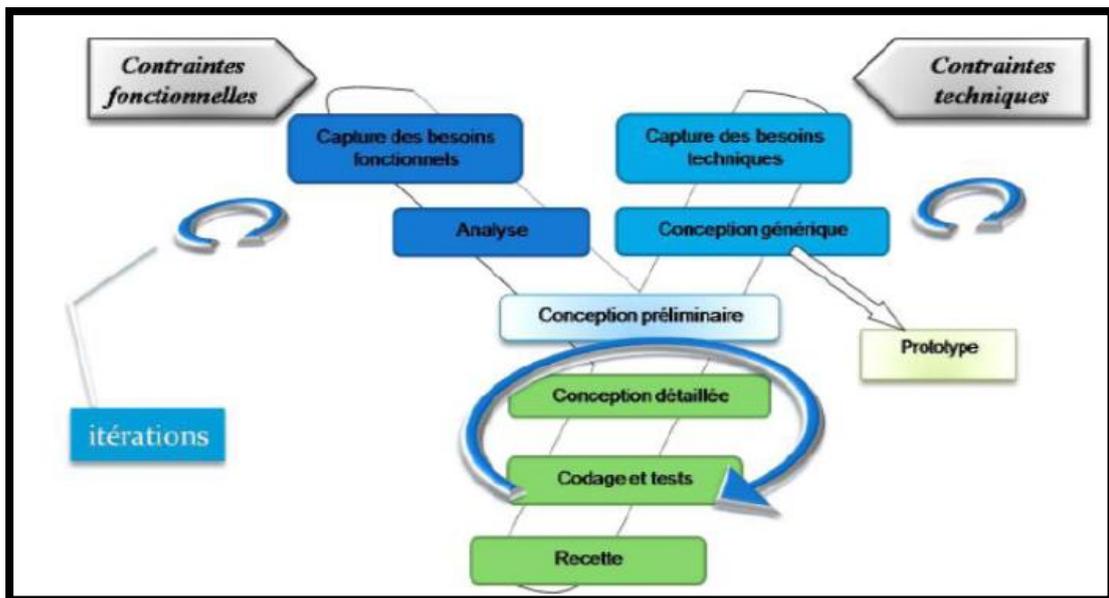


Figure 2.2 : Le processus de développement en Y.

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.

Cette branche comporte les étapes suivantes :

- La capture des besoins fonctionnels : qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux 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.

Cette branche comporte les étapes suivantes :

- La capture des besoins techniques : Les outils et le matériel sélectionné ainsi que la prise en compte des contraintes d'intégration avec l'existant.

- La conception générique : qui définit ensuite les composants nécessaires à la construction de l'architecture technique.

1.2.3. La branche du milieu

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

Cette branche comporte les étapes suivantes :

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

2. Un processus de modélisation avec UML : Le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de par leur facilité et clarté, de bien modéliser le système à chaque étape. « Unified Modeling Language » : UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. il ne s'agit pas d'une simple notation, mais les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage, c'est pour ça qu'UML est présenté parfois comme une méthode alors qu'il ne l'est absolument pas.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage [10].

On va voir une présentation des différents diagrammes UML qui utilisera dans le projet :

- **Le diagramme des cas d'utilisation** : représente la structure des fonctionnalités nécessaires aux utilisateurs du système. Il est normalement utilisé lors des étapes de capture des besoins fonctionnels et techniques.
- **Le diagramme d'activités** : représente les règles d'enchaînement des activités et actions dans le système. Il peut être assimilé comme un algorithme mais schématisé.
- **Le diagramme de séquence et de communication** : représente les échanges de messages entre objets, dans le cadre d'un fonctionnement particulier du système.
- **Le diagramme de classes** : sûrement l'un des diagrammes les plus importants dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs.
- **Le diagramme d'états** : représente le cycle de vie d'un objet. Il spécifie les états possibles d'une classe et leur enchaînement. Ce diagramme est utilisé lors des étapes d'analyse et de conception.
- **Le diagramme de composants** : représente les concepts connus de l'exploitant pour installer et dépanner le système. Il s'agit dans ce cas de déterminer la structure des composants d'exploitation que sont les bibliothèques dynamiques, les instances de bases de données, les applications, les progiciels, les objets distribués, les exécutables, etc.
- **Le diagramme de déploiement** : correspond à la fois à la structure du réseau informatique qui prend en charge le système logiciel, et la façon dont les composants d'exploitation y sont installés.

II. Etude de cas

1. Etude préliminaire

L'étude préliminaire est la toute première étape de notre processus de développement. Il est important pour démarrer l'analyse par le positionnement le plus précis possible du système à étudier.

Ainsi, pour recueillir les besoins des utilisateurs et de situer le contexte du système.

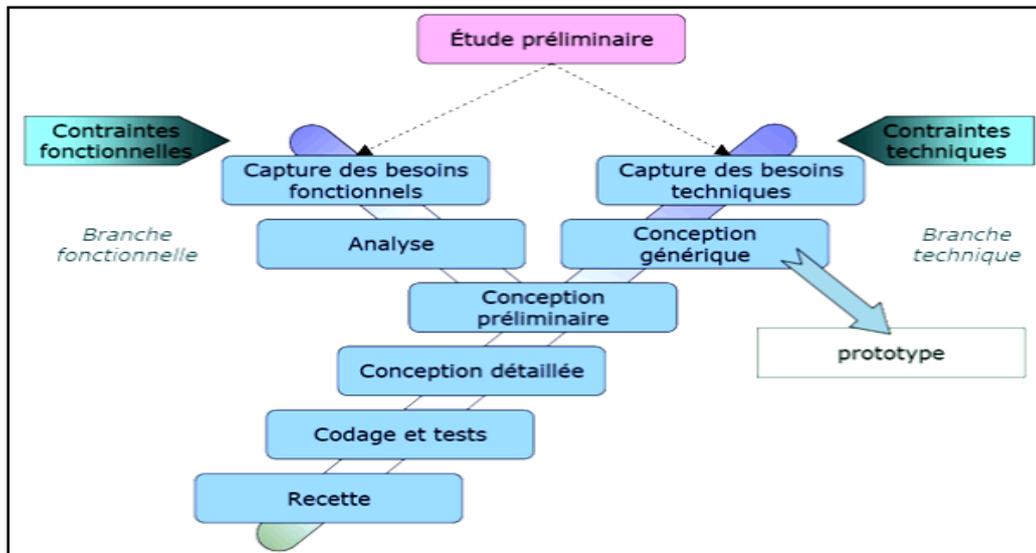


Figure 2.3: Situation de l'étude préliminaire dans 2TUP.

Dans cette étape il devra décrire les besoins et les acteurs qui vont intégrer avec notre futur système. Ces besoins vont des fonctionnalités jusqu'aux aspects techniques du système.

1.1. Elaboration du cahier de charge

1.1.1. la présentation du projet

Le but de notre projet de fin d'étude est la conception et la réalisation d'un système de gestion optimisé qui permet le partage des ressources d'hôpital entre les patients de coronavirus au moment de leur admission au service jusqu'à leur sortie.

1.1.2. les choix techniques

On souhaite utiliser une approche itérative et incrémentale, fondée sur le processus en Y (méthode 2TUP) qui utilise le langage standard de modélisation UML, nous avons affranchi le choix d'un certain nombre de techniques clés pour ce projet :

- La modélisation avec UML 2.0.
- Le serveur XAMPP Control Panel pour le stockage et la gestion des données.
- Netbeans JavaScript.

1.1.3. Recueil des besoins fonctionnels

Un premier tour d'horizon des besoins exprimés par les personnels affectés au service de COVID a permis d'établir le cahier des charges préliminaire suivant :

- **La gestion du personnel**

Permet à l'administrateur de faire les différentes opérations sur les comptes de personnel, le planning de travail, la gestion des commandes et lits pour assurer le bon déroulement de travail.

- **La gestion d'admission**

La décision d'hospitalisation d'un patient se fait à base des résultats de ses analyses, après la consultation, le patient doit suivre la procédure d'entrée.

- **La consultation des patients**

Permet au médecin de donner le feu vert pour hospitaliser un patient, enregistrer les différentes consultations, le traitement de patient COVID et ces besoins de médicaments et de finaliser l'hospitalisation des patients.

- **Tâches de l'infirmier**

Permet à l'infirmier d'enregistrer le suivi quotidien des patients COVID-19, l'application des traitements, ainsi que la mise à jour de ses états selon le bilan.

- **Gestion des lits**

Permet de définir les types et l'état (occupé ou libre) des lits.

- **La gestion de la pharmacie**

Permet au pharmacien d'assurer le bon contrôle du stock (mouvements des médicaments) selon les besoins du service.

- **La consultation de la fiche navette**

Le suivi des patients COVID au niveau du service se résume à travers une fiche qui s'appelle la fiche navette, elle assure la communication entre les médecins et les infirmiers et marque une trace de tous les traitements de la journée.

- **La gestion du planning**

Permet de faire un planning du cadre médical pour assurer le bon déroulement de travail dans le service.

- **Affectation des ressources**

Permet l'affectation journalier des ressources humaines (médecins et Infirmiers) et médicaux aux patients.

1.1.4. Recueil des besoins opérationnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, et ergonomiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. En ce qui concerne notre application, nous avons dégagé les besoins suivants :

- **La disponibilité**

L'application doit être disponible pour être utilisée par tous les utilisateurs intégrés dans la base de données.

- **La sécurité**

La sécurité des données est mise en exergue par un certain nombre des besoins techniques.

- ✓ **La gestion des utilisateurs**

- L'ajout d'un compte : consiste à ajouter un nouveau compte avec les informations concernant l'utilisateur (numéro utilisateur, nom et prénom d'utilisateur, date de naissance, adresse, la qualité, affectation au service COVID et le numéro de téléphone et n'oublie pas les informations d'authentications qui sont le username et le mot de passe).
- La modification des comptes : elle permet de modifier les informations d'un utilisateur.
- La suppression des comptes : la mise à jour de la base de données.

- ✓ **Authentification**

Chaque utilisateur se connecte au système par son username et son mot de passe.

- **Gérer l'intégrité des données**

Une contrainte d'intégrité est une clause permettant de contraindre la modification de tables, faite par l'intermédiaire de requêtes d'utilisateurs, afin que les données saisies dans la base soient conformes aux données attendues. Ces contraintes sont :

- ✓ Définir un attribut par défaut.
- ✓ Forcer la saisie obligatoire des champs.
- ✓ Emettre des conditions sur les champs.

- **La convivialité de l'interface graphique**

L'application doit fournir une interface conviviale et simple pour tout type d'utilisateurs, car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci on découvrira ses fonctionnalités.

- **Une solution ouverte et évoluée**

L'application doit pouvoir être améliorée par l'ajout d'autre fonctionnalité pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.

1.2. Description du contexte

1.2.1. Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système en émettant et /ou recevant des messages éventuellement porteurs de données.

Nous avons identifié les acteurs suivants :

- **Administrateur** : Il se charge essentiellement de la gestion des comptes, faire le planning journalier, l'affectation des cadres médicaux, et la gestion des lits.
- **Agent d'admission** : chargé d'admettre les patients par la gestion, la classification et l'organisation des dossiers médicaux.
- **Médecin** : Consulte les malades du service et prend les décisions concernant les traitements médicaux aux patients et leur sortie.
- **Infirmier** : il est chargé d'écrire les conseils et les prescriptions des traitements décidés par les médecins, et également de donner les soins et de rédiger le bilan de patient selon les besoins.
- **Pharmacien** : Répond aux besoins du service par les médicaments, les outils et les matériaux médicaux, ainsi que la gestion (l'entrée et la sortie) du stock.

1.2.2. Identification des messages

Un message représente la spécification d'une communication entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur. Ainsi, pour chaque acteur cité auparavant, il faut chercher les messages qui déclenchent un comportement du système attendu par l'acteur dans le cadre de son activité et de l'autre côté, chercher les messages émis par le système à l'intention d'un acteur particulier.

On va détailler les différents messages échangés entre le système et l'extérieur :

- **Le système émet les messages suivants**
 - Compte d'un fonctionnaire.
 - Fiche navette.
 - Confirmation de la mise à jour de fiche navette.
 - Prescription.
 - Le bilan.
 - Mise à jour de stock.
 - Commandes.

- Le planning.
- Plan d'affectation.
- Les lits.
- **Le système reçoit les messages suivants**
- Gérer compte () :
 - ✓ Créer compte ().
 - ✓ Modifier compte ().
 - ✓ Supprimer compte ().
- Admission patient ().
 - ✓ Créer dossier patient ().
 - ✓ Modifier dossier patient ().
 - ✓ Supprimer compte ().
- Décision de la sortie d'un patient ().
- Mise à jour de fiche navette ().
- Consulter la fiche navette ().
- Livraison des produits.
- Gérer le stock ().
- Envoyer les commandes ().
- Faire le bilan ().
- Faire planning ().
- Affectation des cadres médicaux ().
- Affectation des médicaments ().
- Gérer les lits.

1.3. Modélisation du contexte

A partir des informations obtenues lors des deux précédentes étapes, 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 :

Utilisateurs finaux	Description des besoins fonctionnels
Administrateur	L'application doit permettre à l'administrateur de : <ul style="list-style-type: none">- S'authentifier.- Gérer les comptes des fonctionnaires.- Faire le planning de travail.- Définir les types des lits.- Affectation des ressources humaine et médicaments.- Envoyer des commandes.- Consulter le travail des autres acteurs.
Agent d'admission	L'application doit permettre à l'agent d'admission de : <ul style="list-style-type: none">- S'authentifier.- Admettre un patient.- Gérer les dossiers des patients.- Consulter la fiche navette.
Médecin	L'application doit permettre au médecin de : <ul style="list-style-type: none">- S'authentifier.- Prescrire soins.- Consulter et modifier la fiche navette.- Décision de la sortie d'un patient.- Consulter le planning.
Infirmier	L'application doit permettre à l'infirmier de : <ul style="list-style-type: none">- S'authentifier.- Consulter et modifier la fiche navette d'un patient.- Faire le bilan.- Consulter le planning.
Pharmacien	L'application doit permettre au pharmacien de : <ul style="list-style-type: none">- S'authentifier.- Livraison des produits.- Entré stock.- Gérer le stock.

Tableau 2.1 : Les rôles de chaque acteur.

1.3.1. Diagramme de contexte dynamique

Il permet de positionner le système dans son environnement à travers les communications, il précise les échanges d'informations entre le système et les différents acteurs (description physique et logique du système vu en diagramme de contexte statique). Chaque numéro correspond à un message dans la table de message suivante.

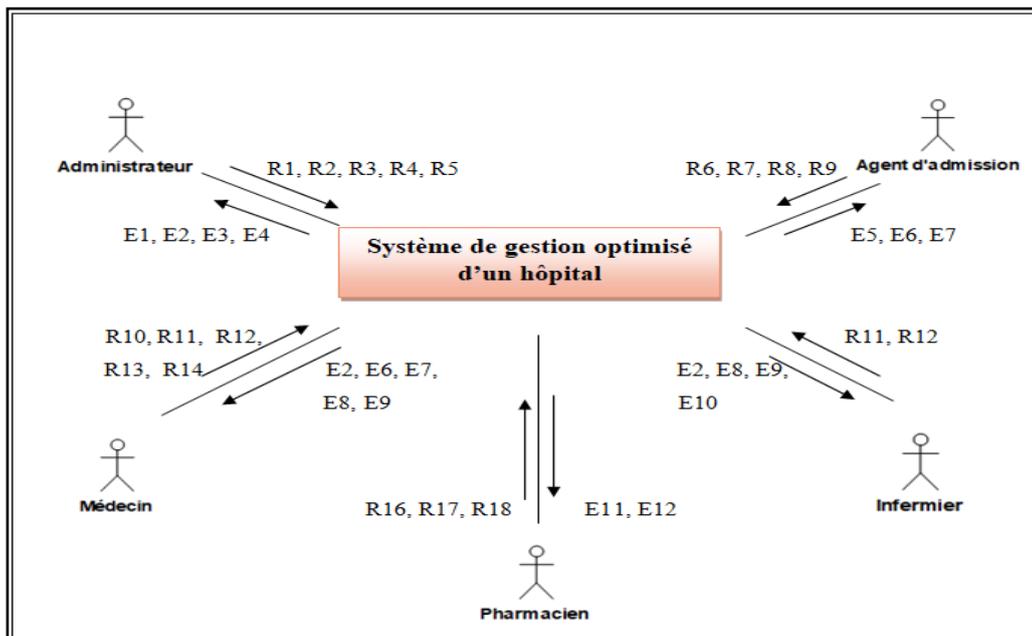


Figure 2.4 : Diagramme de contexte dynamique.

1.3.2. Signification des messages

Les messages émis	Les messages reçus
E1 : Compte.	R1 : Gérer comptes fonctionnaires ().
E2 : Le planning des fonctionnaires.	R2 : Faire le planning ().
E3 : Identifier les lits.	R3 : Affectation ().
E4 : commande.	R4 : Gestion des lits ().
E5: Admission patient.	R5 : Envoyer commandes ().
E6 : Dossier patient.	R6 : Admission patient ().
E 7 : Mise à jour de dossier patient.	R7 : Gérer dossier patient ().
E6 : prescription.	R 8: Consultation dossier patient ().
E7 : Fiche navette.	R9: Chercher dossier patient ().
E8 : Mise à jour de fiche navette.	R10: Finaliser hospitalisation ().
E9 : PV de sortie.	R11 : Consulter patient ().
E10: Le bilan d'un patient.	R12: Mise à jour fiche navette ().
E11 : Mise à jour des produits.	R13 : Consulter planning ().
E12 : Mise à jour de stock.	R14 : Consulter fiche navette ().
	R15 : Faire le bilan ().
	R16 : Livraison des produits ().
	R17 : Entré stock.
	R18 : Gérer le stock ().

Tableau 2.2: Signification des messages.

2. Capture des besoins fonctionnels

Cette phase est la première étape de la branche gauche du cycle en Y. Elle représente un point de vue « fonctionnel » de l'architecture de système. Chaque cas d'utilisation sera identifié, décrit, et organisé, classé en fonction de son importance dans le projet.

2.1. Déterminer les cas d'utilisation

2.1.1. Liste préliminaire des cas d'utilisation

En considérons l'intention fonctionnelle de l'acteur par rapport au système dans le cadre de l'émission ou de la réception de chaque message. Chaque intention fonctionnelle est modélisée par un cas d'utilisation

Cas d'utilisations	Acteurs	Message(s) émis / reçus par les acteurs
Gérer les comptes personnels	Administrateur	Emet : Saisir les informations (). Reçoit : Compte du fonctionnaire.
Créer compte	Administrateur	Emet : Créer compte () Reçoit : Mise à jour des comptes.
Modifier compte	Administrateur	Emet : Modifier compte (), Reçoit : Mise à jour des comptes.
Supprimer compte	Administrateur	Emet : Supprimer compte (), Reçoit : Mise à jour des comptes.
Faire le planning	Administrateur	Emet: Créer planning () / modifier planning () / supprimer planning (). Reçoit : Le planning de travail.
Affectation des ressources	Administrateur	Emet : Récupérer les informations (). Reçoit : Tables des ressources.
Gérer les lits	Administrateur	Emet : Saisir les informations (). Reçoit : Mise à jour les lits.
Admettre patient	Agent d'admission	Emet : Saisir les informations (). Reçoit : Admission patient.
Modifier dossier	Agent d'admission	Emet : Modifier dossier (), Reçoit : Fiche navette.
Supprimer dossier	Agent d'admission	Emet : Supprimer dossier (), Reçoit : La mise à jour de la table des dossiers.
Chercher patient	Agent d'admission	Emet : Saisir les informations (). Reçoit : Fiche navette patient.
Consulter fiche navette	Agent d'admission/ Médecin/infirmier	Emet : Saisir les informations (). Reçoit : Fiche navette.
Consulter planning	Médecin / infirmier	Emet : Saisir les informations (). Reçoit : le planning de travail.
Prescrire patient	Médecin	Emet : Saisir les informations (). Reçoit : Mise à jour fiche navette.
Mise à jour fiche navette	Médecin / infirmier	Emet : Saisir ()/ Modifier ()/ Enregistrer (). Reçoit : Mise à jour de la fiche navette.

Décision de sortie patient	Médecin	Emet : Saisir les informations (). Reçoit : Mise à jour de la fiche navette.
Faire le bilan	Infirmier	Emet : Saisir information (). Reçoit : Mise à jour de fiche navette.
Gérer produit	Pharmacien	Emet : Saisir information (), Reçoit : Mise à jour des produits.
Entrer le stock	Pharmacien	Emet : Ajouter ()/modifier()/ supprimer (). Reçoit : Mise à jour de stock.
Livraison des produits	Pharmacien	Emet : Saisir information (), Reçoit : Produit.

Tableau 2.3 : Liste des acteurs et des messages de chaque cas d'utilisation du système.

2.1.2. Description préliminaire des différents cas d'utilisation

Voici une description préliminaire des cas d'utilisations énumérés précédemment :

- **Gérer les comptes :**
 - Ajouter un nouveau fonctionnaire dans le système.
 - Modifier et sauvegarder les informations d'un fonctionnaire.
 - Entrer les informations et supprimer le dossier d'un fonctionnaire.
- **Faire le planning :**
 - Saisir les informations et choisir le type cadre médical de planning.
 - Créer et supprimer le planning des cadres médicaux.
- **Gérer les ressources du personnel et médicaments:**
 - Affecter chaque cadre médical a chaque patient selon le planning.
 - Affecter les médicaments aux patients selon la disponibilité des produits dans la pharmacie.
- **Gérer les lits :**
 - L'ajout et la suppression des lits selon le type.
- **Admettre patient :**
 - Saisir et sauvegarder les informations d'un patient.
- **Modifier dossier:**
 - Saisir et sauvegarder les informations modifiant d'un dossier patient.
- **Supprimer dossier:**
 - Saisir les informations en supprimant le dossier patient.
- **Chercher patient:**
 - Enter le motif de recherche et sélectionner le patient.
 - Afficher les détails et sélectionner le patient.

- **Consulter fiche navette :**
 - Enter le motif de recherche et sélectionner le patient.
 - Consulter les informations relatives au patient (identification, admission, sortie, médicaments et bilan).
- **Consulter planning :**
 - Entrer les informations et consulter le planning.
- **Consulter patient :**
 - Enter le motif de recherche et sélectionner le patient.
 - Saisir et sauvegarder les informations modifiant la fiche navette de patient.
- **Décision de sortie patient :**
 - Enter le motif de recherche et sélectionner le patient.
 - Saisir et sauvegarder les informations modifiant la fiche navette de patient.
 - Finaliser l'hospitalisation.
- **Faire le bilan :**
 - Enter le motif de recherche et sélectionner le patient.
 - Saisir et sauvegarder les nouveaux en modifiant la fiche navette de patient.
- **Gérer produit :**
 - Saisir et sauvegarder les nouveaux médicaments.
 - Consulter les mouvements des produits.
- **Entrer le stock :**
 - Enter le motif de recherche et sélectionner le produit.
 - La réception des nouvelles quantités.
- **Livraison des produits :**
 - Réceptionner la commande envoyée par le service.
 - Faire sortir des produits à partir du stock selon la disponibilité.

2.1.3. Diagramme cas d'utilisation des besoins fonctionnels

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 :

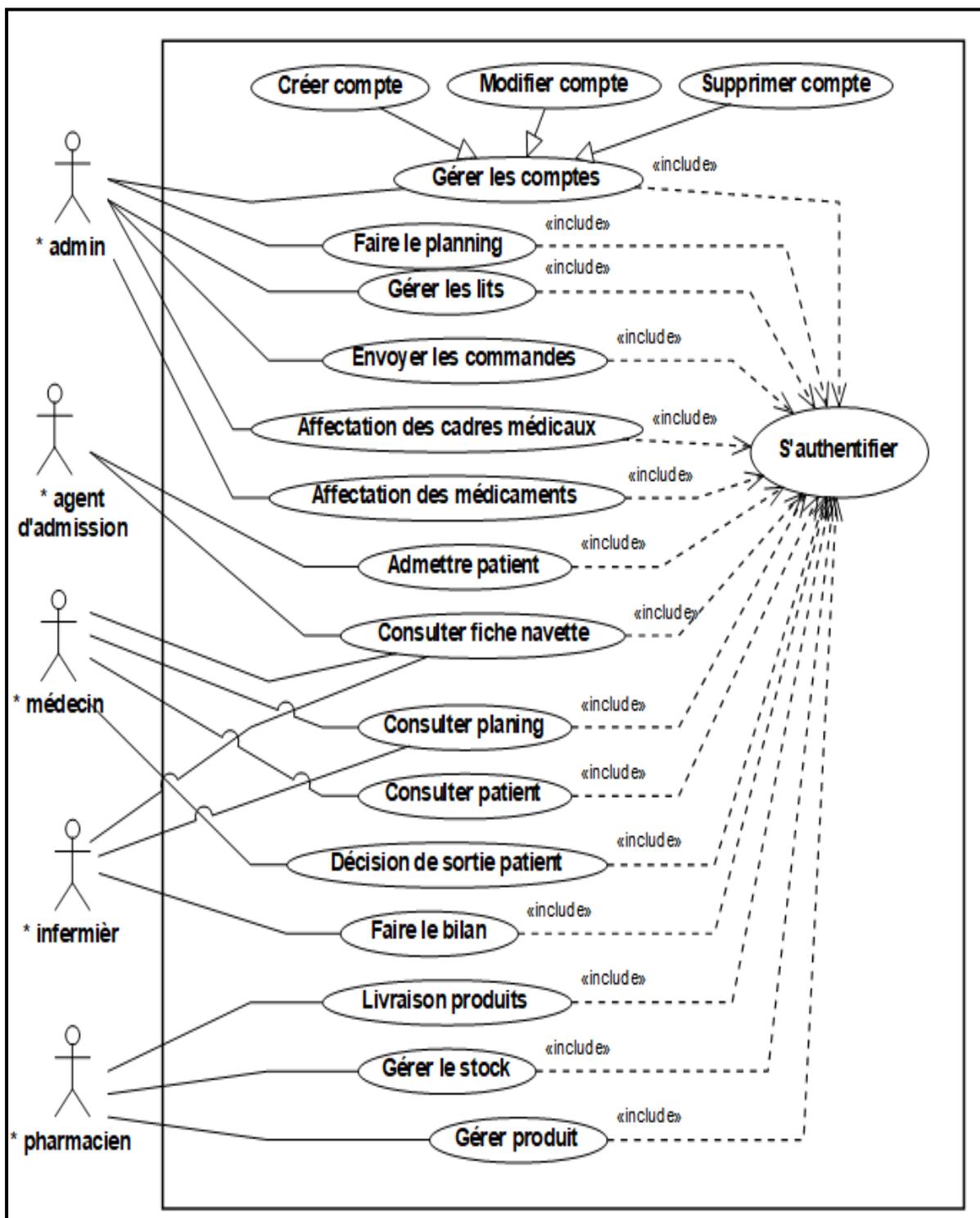


Figure 2.5:Diagramme de cas d'utilisation.

2.2. Description détaillée de chaque cas d'utilisation

2.2.1. Cas d'utilisation « Gérer les comptes du personnels »

2.2.1.1. Fiche descriptive

Titre : Gérer les comptes des personnels.
Acteurs : Administrateur.
But : La gestion des comptes personnels.
Pré-condition : Acteur s'authentifie.
Post-condition : La mise à jour des comptes personnels.
Scénario nominal : <ol style="list-style-type: none">1. Administrateur demande le formulaire « gestion de personnel ».2. Le système affiche le formulaire « gestion de personnel ».<ul style="list-style-type: none">• Cas ajouter un nouveau compte :<ol style="list-style-type: none">2.1 L'administrateur saisit les informations de compte.2.2 Le système lit les informations saisies.2.3 Le système vérifie les informations et crée le nouveau compte et affiche une notification « Compte personne ajouté avec succès ».• Cas modification d'un compte :<ol style="list-style-type: none">2.4 Le système affiche la liste des comptes.2.5 L'administrateur sélectionne le compte.2.6 Le système affiche les détails.2.7 L'administrateur saisit les nouvelles modifications.2.8 Le système lit les informations saisies.2.9 Le système vérifie et demande la confirmation.2.10 Le système modifie et affiche une notification « compte personne modifié avec succès ».• Cas suppression un compte :<ol style="list-style-type: none">2.11 Le système affiche la liste des comptes.2.12 L'administrateur sélectionne le compte.2.13 Le système affiche les détails.2.14 L'administrateur supprime le compte.2.15 Le système vérifie l'affectation des personnes dans le planning.2.16 Le système demande la confirmation.<ol style="list-style-type: none">a. Si le personne a des affectations afficher une notification « suppression interdit il y a des affectations sur la personne »b. Sinon le système supprime et affiche une notification « compte personne supprimé avec succès ».
Scénario d'exception : la valeur de champs date de naissance invalide.

Tableau 2.4: Fiche descriptive textuelle de cas d'utilisation « Gérer le comptes du personnels ».

2.2.1.2. Diagramme d'activité de cas d'utilisation « Gérer les comptes du personnels »

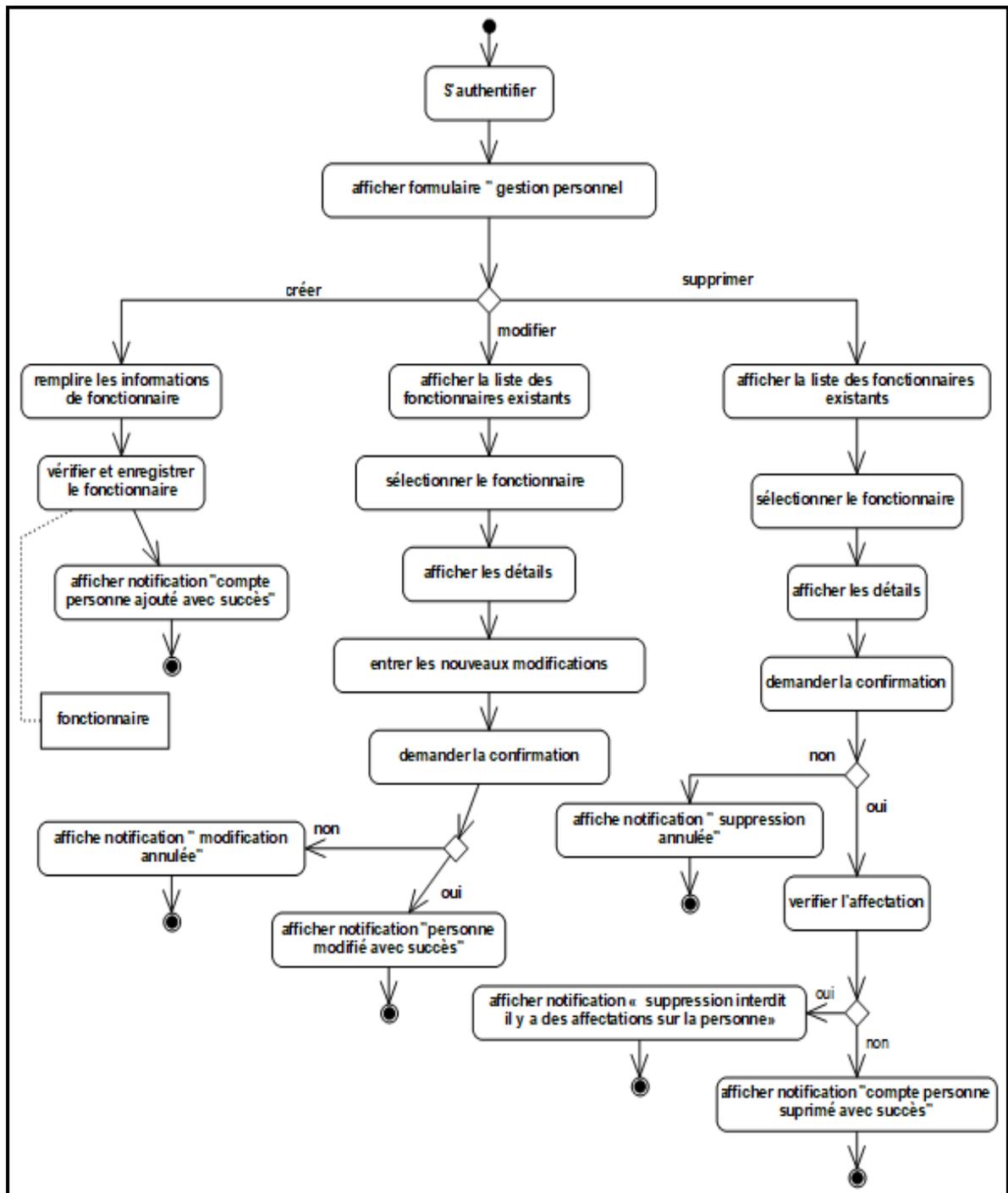


Figure 2.6: Diagramme d'activité de cas d'utilisation « Gérer les compte du personnels ».

2.2.1.3. Diagramme de séquence de cas d'utilisation « Gérer les comptes du personnels »

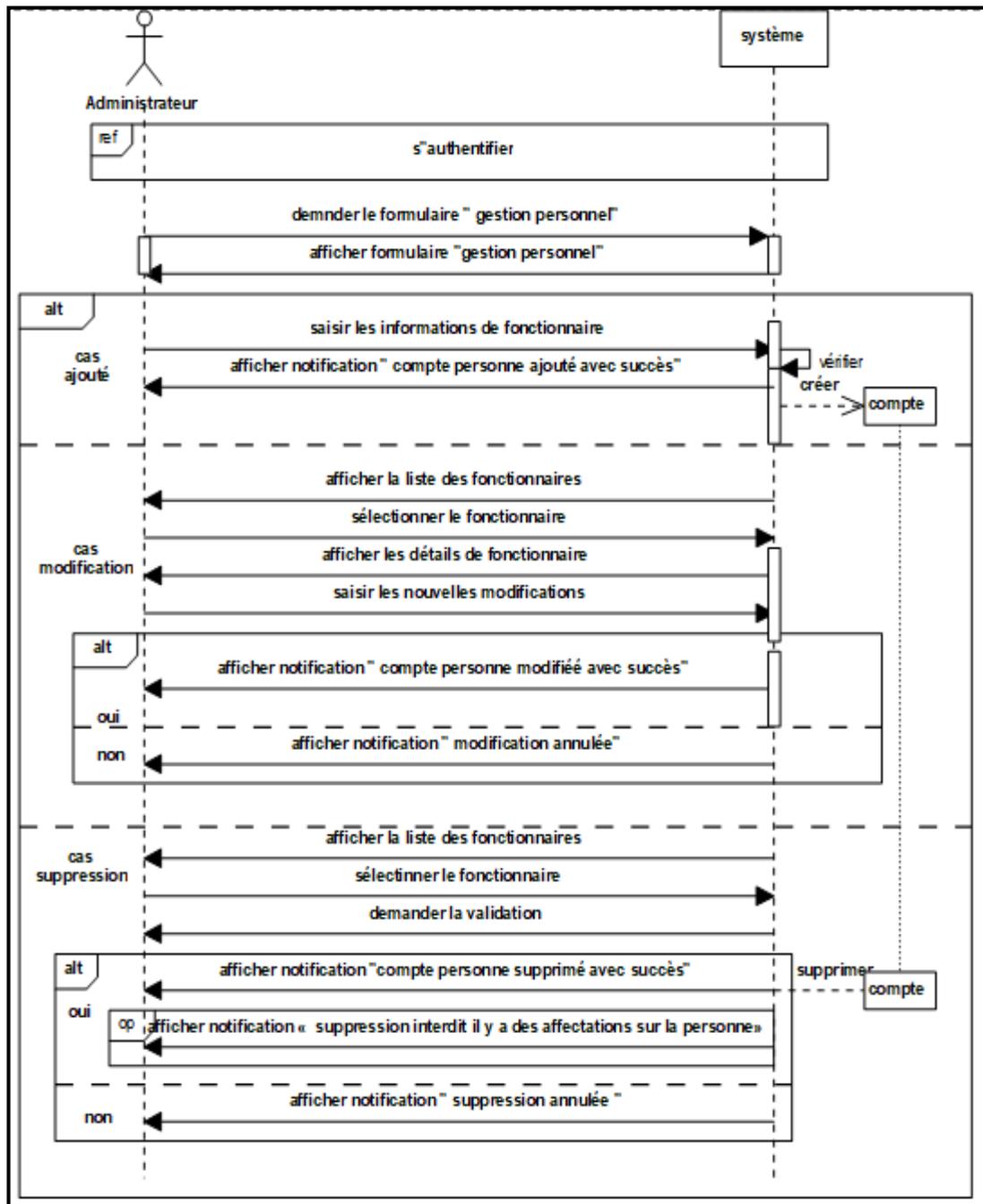


Figure 2.7: Diagramme de séquence de cas d'utilisation « Gérer les compte du personnels ».

2.2.2. Cas d'utilisation « Faire le planning »

2.2.2.1. Fiche descriptive

Titre : Faire le planning.
Acteurs : Administrateur.
But : Faire un planning pour la bonne gestion de travail.
Pré-condition : S'authentifier.
Post-condition : Le planning de chaque fonctionnaire.
Scénario nominal : <ol style="list-style-type: none">1. Administrateur demande le formulaire « gestion planning ».2. Le système affiche le formulaire « gestion planning».3. L'administrateur choisit la date et sélectionne la catégorie des fonctionnaires.4. Le système lit les informations et affiche la liste de catégorie des fonctionnaires.<ul style="list-style-type: none">• Cas de créer planning5. L'administrateur sélectionne la liste des fonctionnaires à programmer.6. Le système affiche une notification « planning du journée enregistré avec succès ».<ul style="list-style-type: none">• Cas modifier planning7. L'administrateur demande le formulaire.8. Le système affiche la liste des personnes de la journée.9. L'administrateur sélectionne le fonctionnaire.10. Système affiche les informations de planning.11. L'administrateur saisit l'état de la présence.12. Le système met à jour le planning.<ul style="list-style-type: none">• Cas suppression planning13. Le système affiche la liste des fonctionnaires.14. L'administrateur sélectionne personnes.15. Le système vérifie l'affectation des personnes dans le planning.16. Le système demande la confirmation.<ol style="list-style-type: none">16.1 Si le personne a des affectations affiche une notification « suppression interdit il y a des affectations sur la personne ».16.2 Sinon le système supprime et affiche une notification « planning de la personne supprimé avec succès».
Scénario d'exception : /

Tableau 2.5 : Fiche descriptive textuelle de cas d'utilisation « Faire planning ».

2.2.2.2. Diagramme d'activité de cas d'utilisation « Faire le planning »

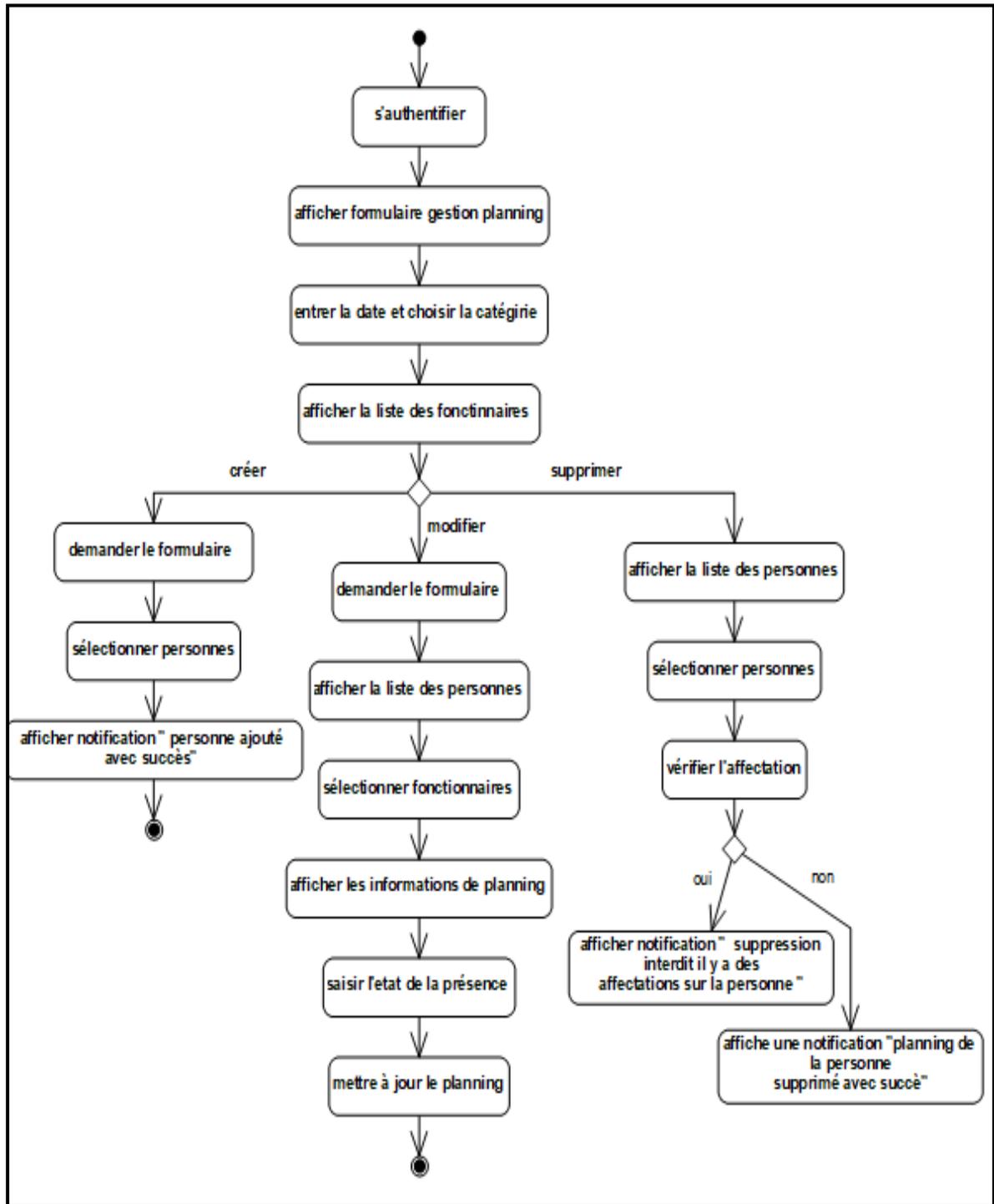


Figure 2.8: Diagramme d'activité de cas d'utilisation « Faire le planning ».

2.2.2.3. Diagramme de séquence de cas d'utilisation « Faire le planning »

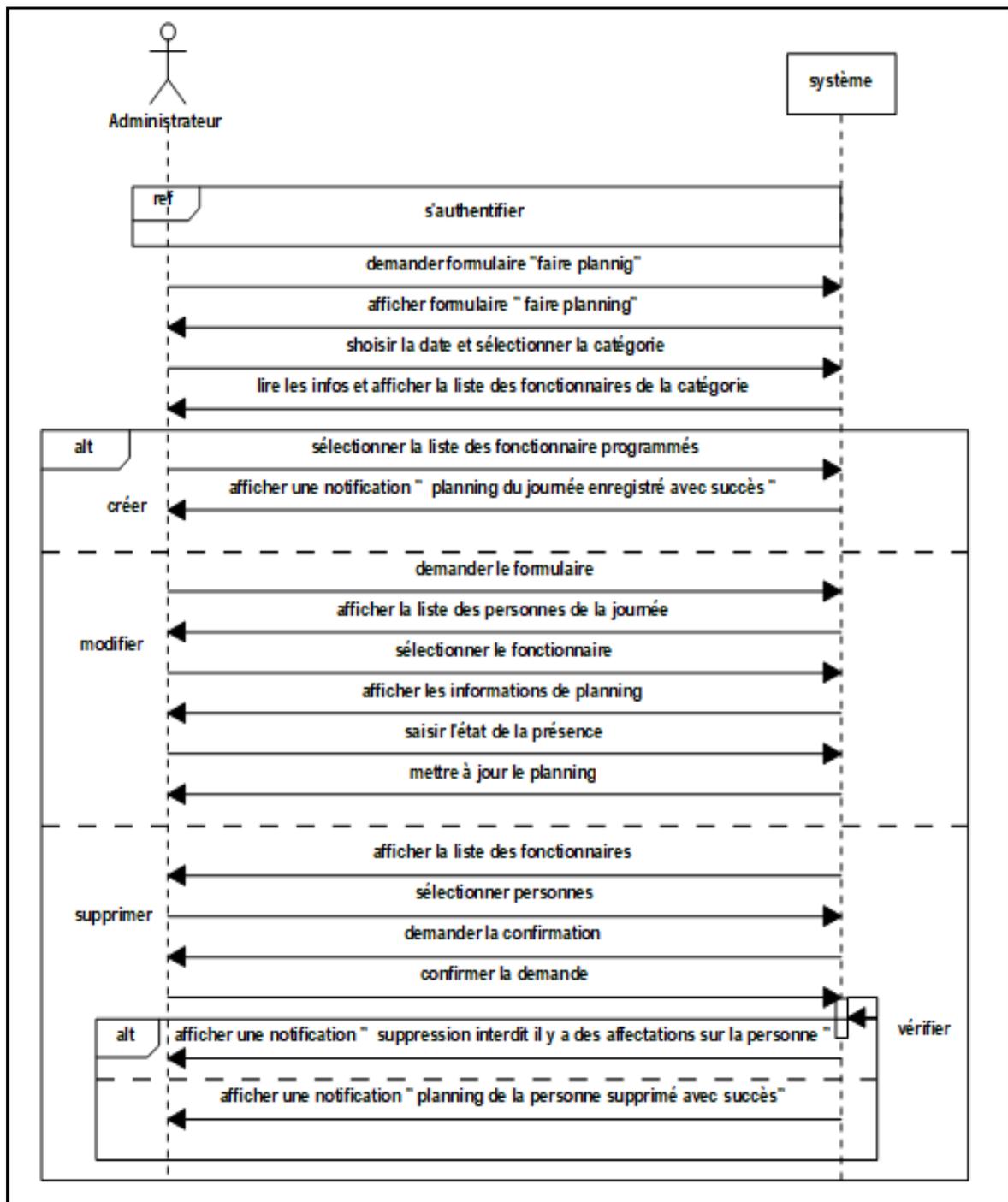


Figure 2.9: Diagramme de séquence de cas d'utilisation « Faire le planning ».

2.2.3. Cas d'utilisation « Affectation des cadres médicaux »

2.2.3.1. Fiche descriptive

Titre : Affectation des cadres médicaux.
Acteurs : Administrateur.
But : Affecter les ressources humaines aux patients.
Pré-condition : S'authentifier.
Post-condition : Chaque patient a ses propres besoins.
Scénario nominal : <ol style="list-style-type: none">1. Administrateur demande le formulaire « affectation personnels ».2. Le système affiche le formulaire « affectation personnels ».3. L'administrateur choisit la journée d'affectation.4. Le système lit les informations saisit et affiche les personnels (par catégorie) selon le planning et la liste des patient en cours d'hospitalisation.<ul style="list-style-type: none">• Cas générer et affecter5. L'administrateur demande le formulaire « générer l'affectation ».6. Le système affiche les listes des médecins, infirmiers et patients.7. L'administrateur génère une affectation.8. Système vérifie l'affectation.9. Le système affiche une notification « affectation génère avec succès ».<ul style="list-style-type: none">• Cas régénérer10. L'administrateur demande le formulaire « régénérer l'affectation ».11. Système affiche le formulaire.12. L'administrateur régénère une autre affectation.13. Système vérifie l'affectation.<ol style="list-style-type: none">a. Si le médecin a des prescriptions affiche une notification « suppression interdit il y a des prescription ».b. Sinon le système supprime l'affectation et affiche une notification « affectation supprimé avec succès ».
Scénario d'exception : /

Tableau 2.6: Fiche descriptive textuelle de cas d'utilisation « Affecter les cadres médicaux ».

2.2.3.2. Diagramme d'activité de cas d'utilisation « Affectation des cadres médicaux »

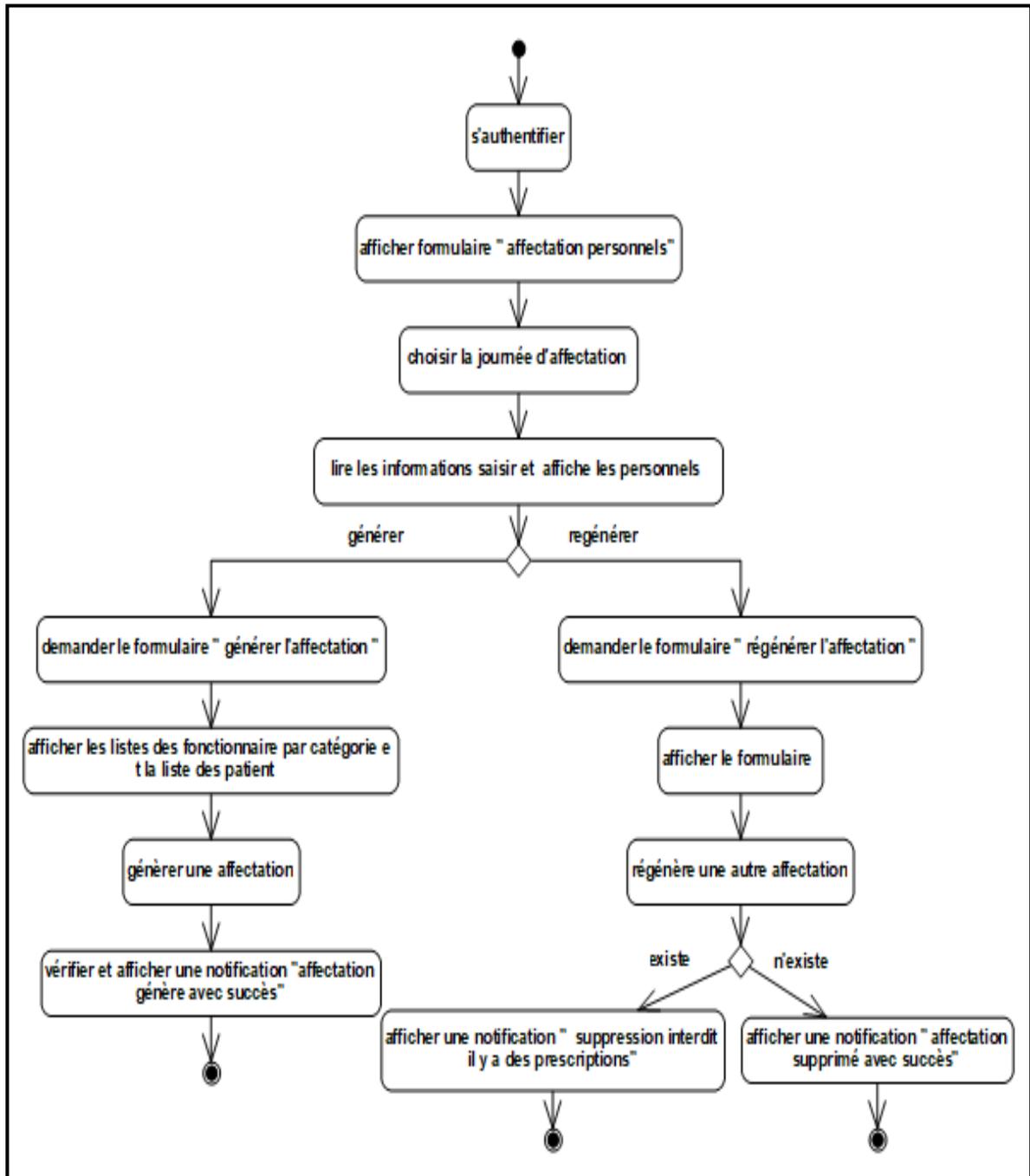


Figure 2.10: Diagramme d'activité de cas d'utilisation« Affecter les cadres médicaux».

2.2.3.3. Diagramme de séquence de cas d'utilisation « Affectation des cadres médicaux»

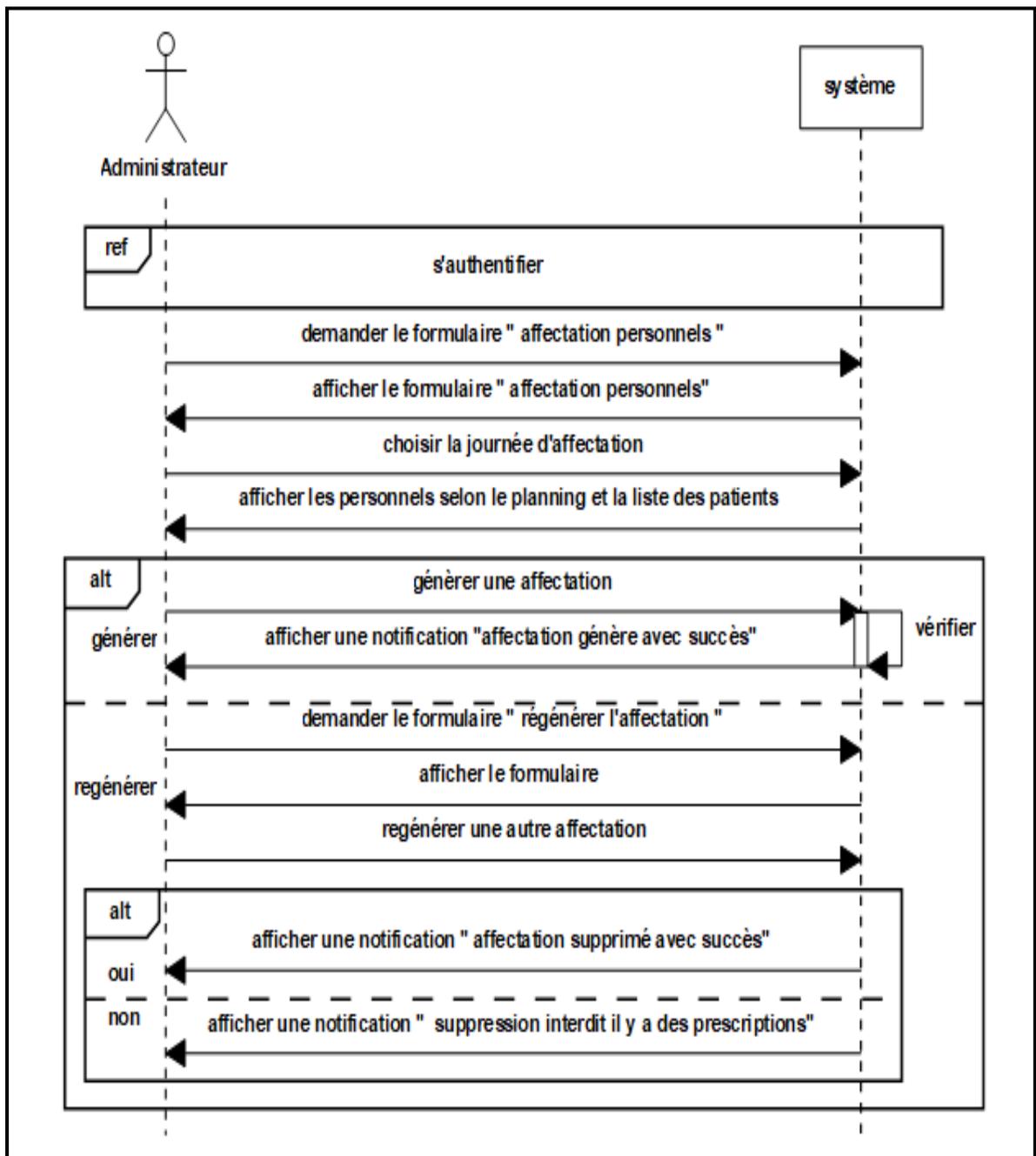


Figure 2.11: Diagramme de séquence de cas d'utilisation « Affecter les cadres médicaux».

2.2.4. Cas d'utilisation « Gérer les lits »

2.2.4.1. Fiche descriptive

Titre : Gérer les lits.
Acteurs : Administrateur.
But : Identifier les nouveaux lits au système.
Pré-condition : S'authentifier.
Post-condition : Connaître les lits dans l'établissement.
Scénario nominal : <ol style="list-style-type: none">1. L'administrateur demande le formulaire «gestion des lits ».2. Le système affiche le formulaire «gestion des lits ».<ul style="list-style-type: none">• Cas ajouter lit3. L'administrateur sélectionne le type de lit.4. Le système affiche une notification «lit ajouté avec succès».<ul style="list-style-type: none">• Cas supprimer lit5. L'administrateur sélectionne le lit.6. Le système demande la confirmation.7. L'administrateur confirme la suppression.8. Le système vérifie le lit.<ol style="list-style-type: none">a. Si e lit est occupé affiche une notification « lit occupé suppression interdit ».b. Sinon système affiche une notification « lit supprimé avec succès ».
Scénario d'exception : /

Tableau 2.7: Fiche descriptive textuelle de cas d'utilisation « Gérer les lits ».

2.2.4.2. Diagramme d'activité de cas d'utilisation « Gérer les lits »

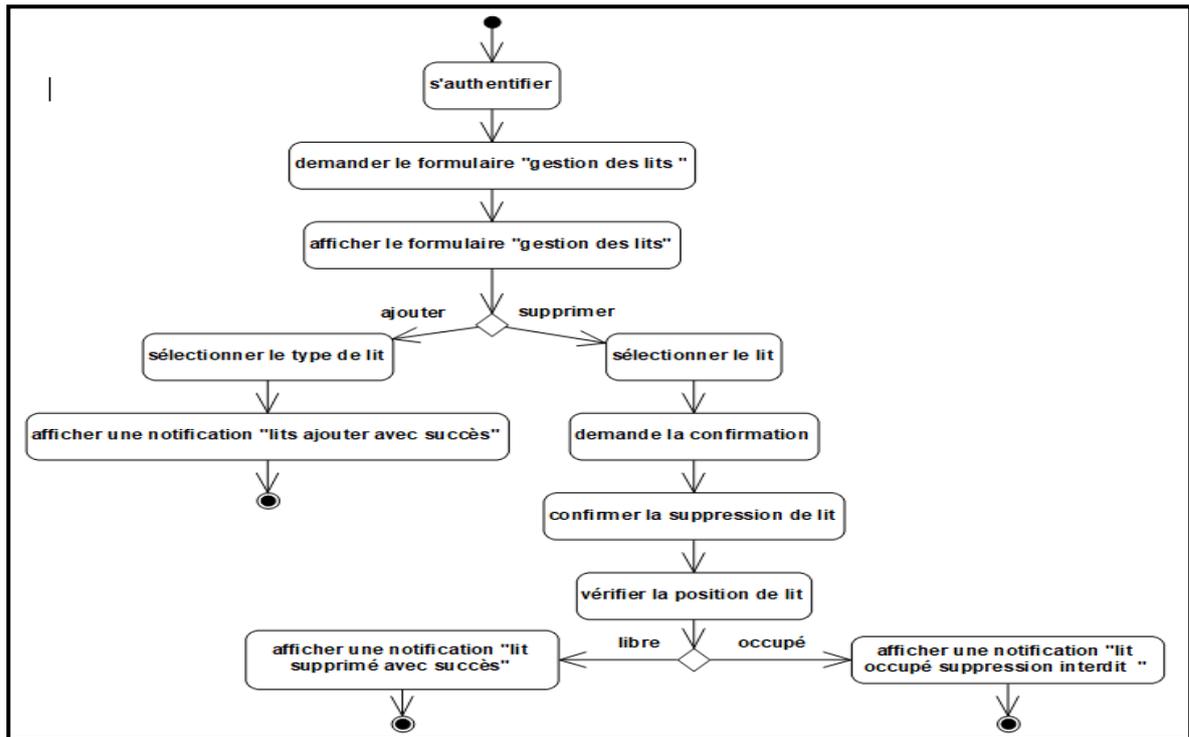


Figure 2.12: Diagramme d'activité de cas d'utilisation « Gérer les lits ».

2.2.4.3. Diagramme de séquence de cas d'utilisation « Gérer les lits »

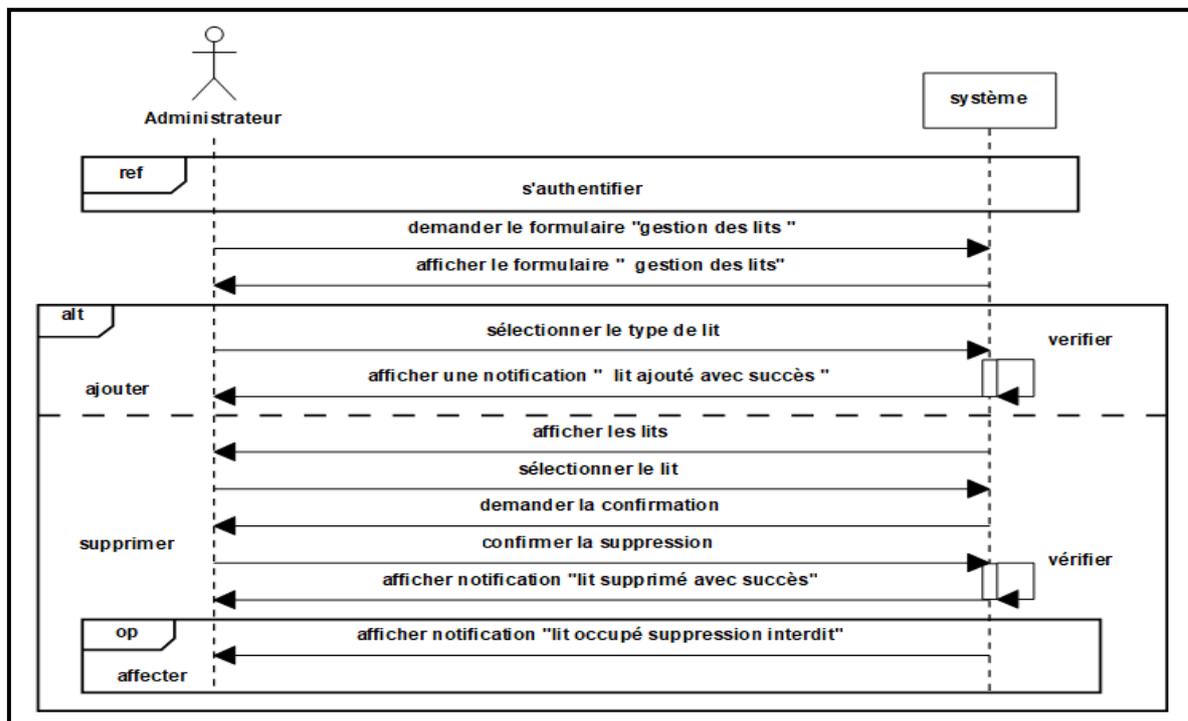


Figure 2.13: Diagramme séquence de cas d'utilisation « Gérer les lits ».

2.2.5. Cas d'utilisation « Admettre patient »

2.2.5.1. Fiche descriptive

Titre : Admettre un patient.
Acteurs : Agent d'admission.
But : Enregistrer l'admission d'un patient en créant éventuellement un dossier.
Pré-condition : L'acteur s'authentifie.
Post-condition : Les informations de l'admission sont enregistrées et un dossier est créé en cas d'un nouveau patient.
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1. Agent d'admission demande le formulaire « gestion d'admission ». 2. Le système affiche le formulaire « gestion d'admission ». <ul style="list-style-type: none"> • Cas ajouté nouveau patient : 3. L'agent d'admission saisit les informations du patient. 4. Le système lit les informations saisies. 5. L'agent d'admission saisie l'état de patient 6. Le système affiche les types des lits selon l'état du patient. <ol style="list-style-type: none"> 6.1 S'il y a des lits libres le système affiche une notification « patient ajouté avec succès ». 6.2 Sinon affiche une notification « service complet ». <ul style="list-style-type: none"> • Cas modification patient : 7. Le système affiche la liste des patients. 8. L'agent d'admission sélectionne le patient. 9. Le système affiche les détails. 10. L'agent d'admission saisie les nouvelles modifications. 11. Le système lit les informations saisies 12. Le système vérifie et demande la confirmation. 13. L'agent d'admission confirme la modification. 14. Le système modifie et affiche une notification « patient modifié avec succès». <ul style="list-style-type: none"> • Cas suppression patient : 15. Le système affiche la liste des patients. 16. L'agent d'admission sélectionne le patient. 17. Le système affiche les détails. 18. L'agent d'admission demande la suppression du patient. 19. Le système vérifie l'existence des prescriptions et demande la confirmation. 20. L'agent d'admission confirme la suppression. 21. Le système supprime et affiche une notification « patient supprimé avec succès». <ol style="list-style-type: none"> 22.1 Le système modifie l'affectation de lit. 22.2 Sinon le système affiche une notification « il y a des prescriptions sur le patient, suppression interdit ».
Scénario d'exception : /

Tableau 2.8: Fiche descriptive textuelle de cas d'utilisation « Admettre patient ».

2.2.5.2. Diagramme d'activité de cas d'utilisation « Admettre patient »

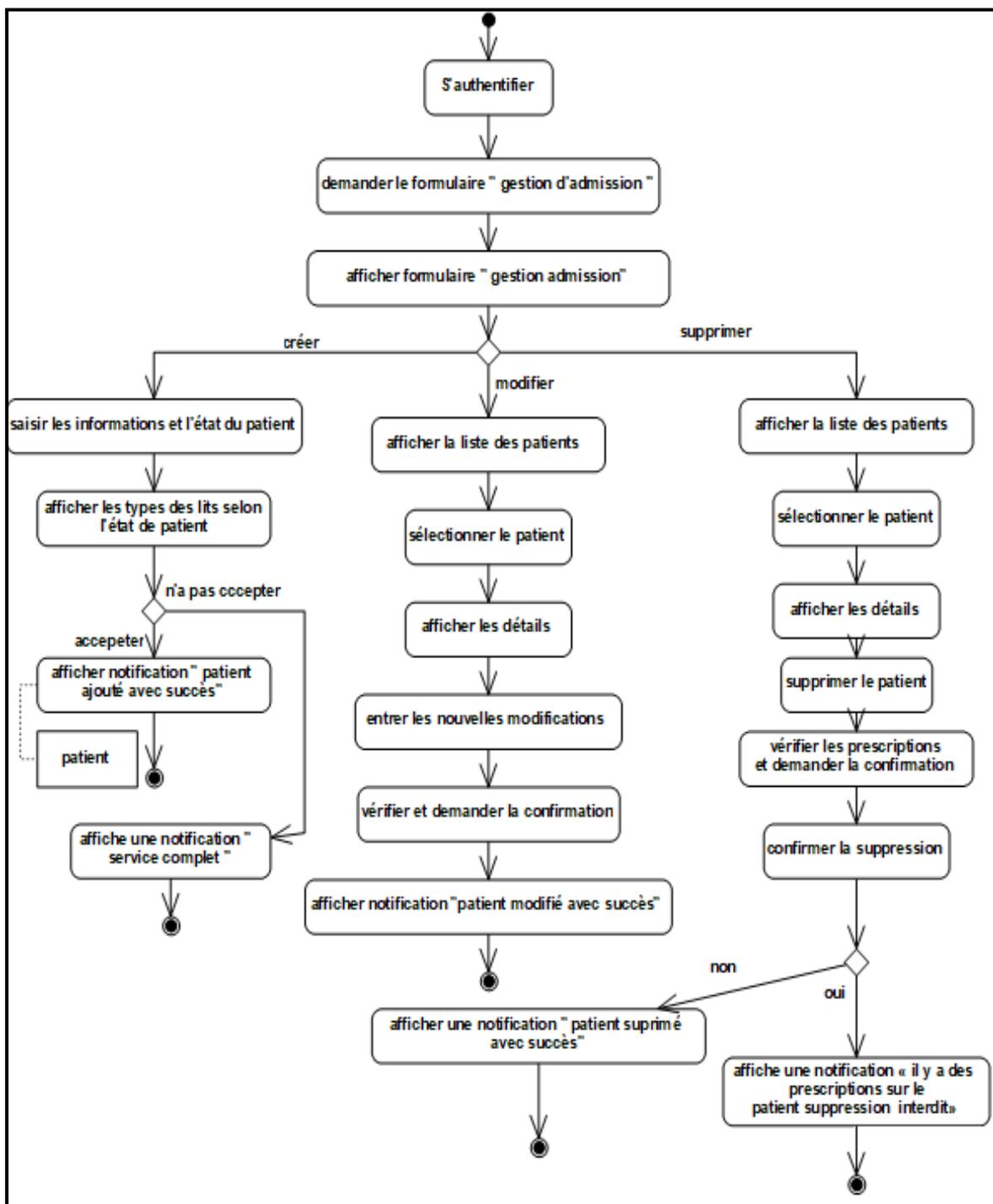


Figure 2.14: Diagramme d'activité de cas d'utilisation « Admettre patient ».

2.2.5.3. Diagramme de séquence de cas d'utilisation « Admettre patient »

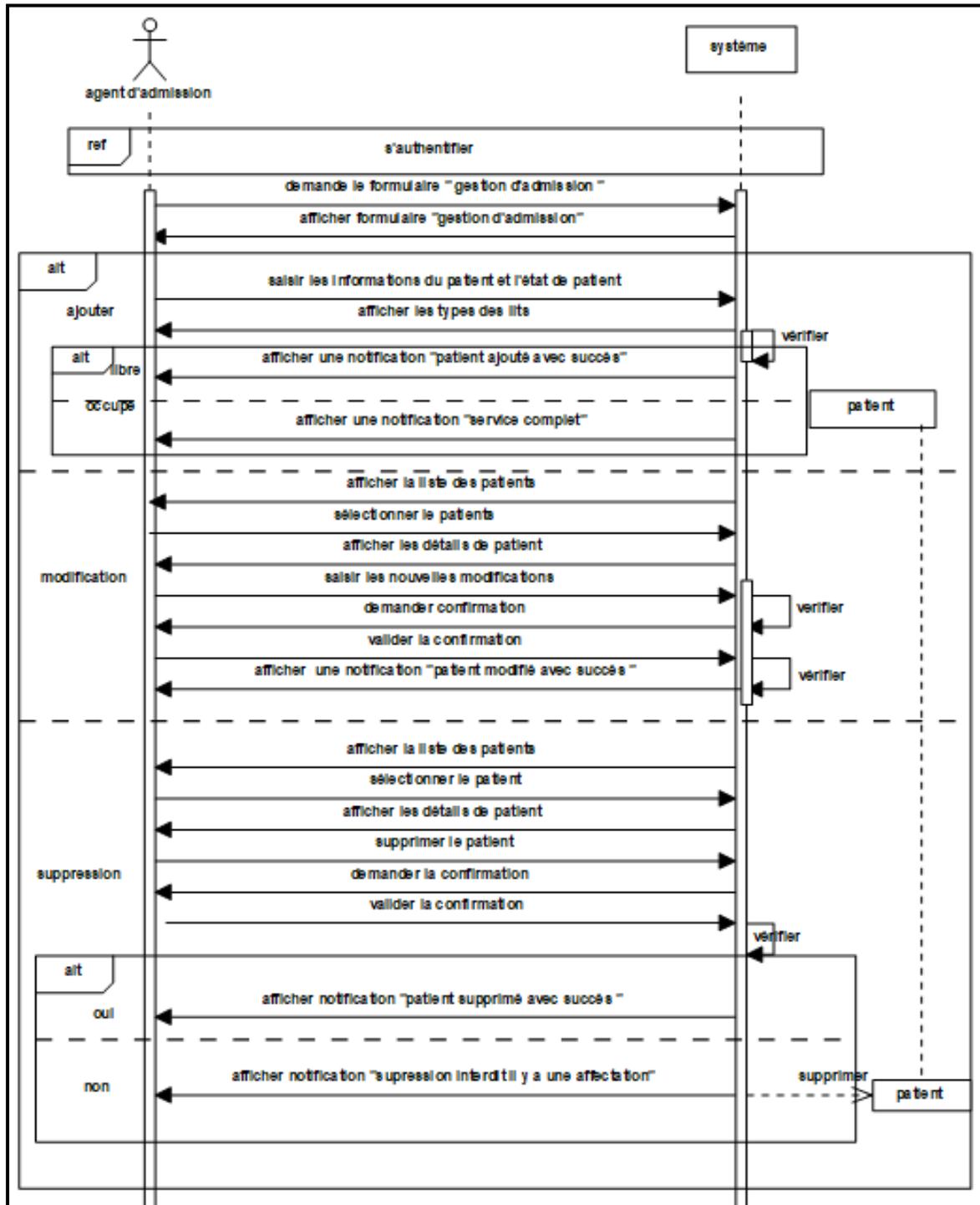


Figure 2.15: Diagramme de séquence de cas d'utilisation « Admettre patient ».

2.2.6. Cas d'utilisation « Consulter fiche navette »

2.2.6.1. Fiche descriptive

Titre : Consulter fiche navette.
Acteurs : Médecin, infirmier, agent d'admission.
But : Consulter les détails de patient.
Pré-condition : L'acteur s'authentifié.
Post-condition : Les informations de patient sont affichées.
Scénario nominal : <ol style="list-style-type: none">1. L'acteur demande le formulaire «fiche navette ».2. Le système affiche le formulaire «fiches navette ».3. L'acteur sélectionne le motif de recherche.4. Le système affiche la liste des patients.5. L'acteur choisit le patient.6. Le système affiche les détails du patient sélectionné.7. L'acteur consulte les informations de la fiche navette du patient sélectionné.
Scénario d'exception : /

Tableau 2.9: Fiche descriptive textuelle de cas d'utilisation « Consulter fiche navette ».

2.2.6.2. Diagramme d'activité de cas d'utilisation « Consulter fiche navette »

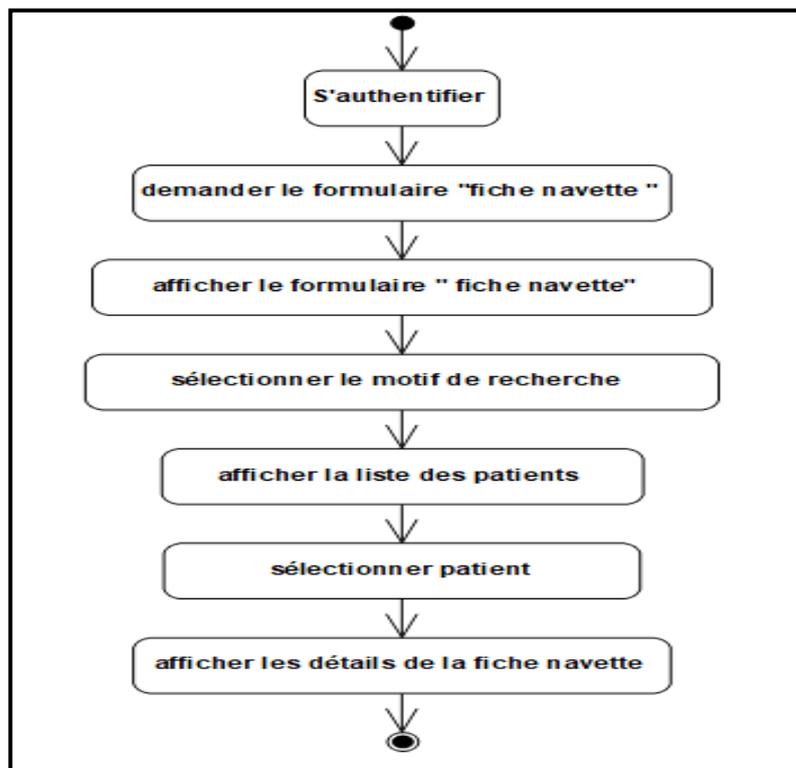


Figure 2.16: Diagramme d'activité de cas d'utilisation « Consulter fiche navette ».

2.2.6.3. Diagramme de séquence de cas d'utilisation « Consulter fiche navette »

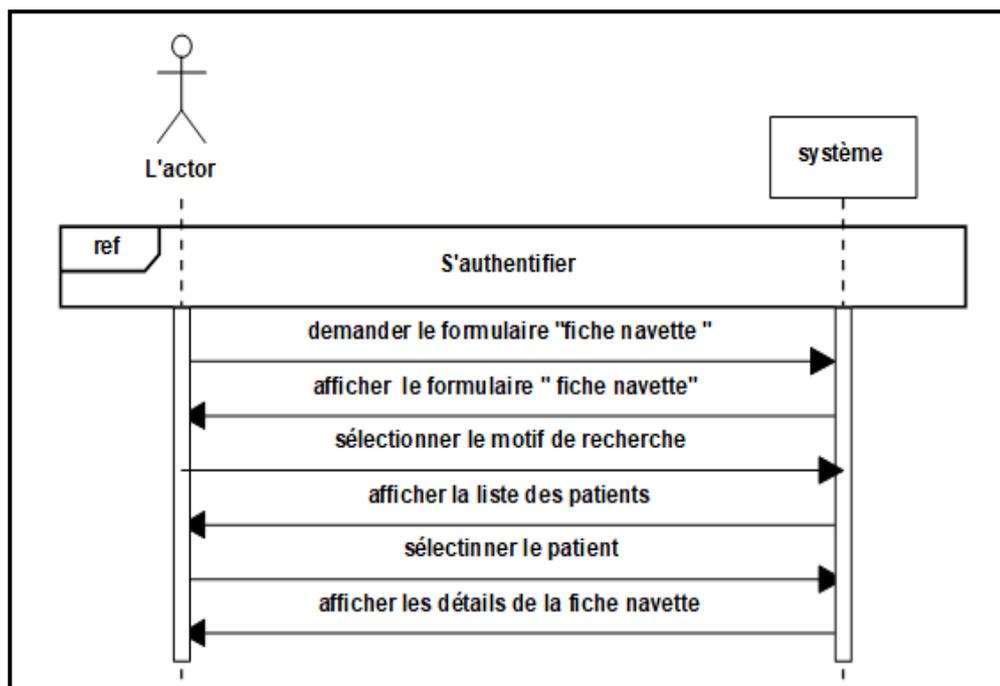


Figure 2.17: Diagramme de séquence de cas d'utilisation« Consulter fiche navette ».

2.2.7. Cas d'utilisation « Consulter planning »

2.2.7.1. Fiche descriptive

Titre : Consulter planning.
Acteurs : Médecin, infirmier.
But : Connaitre les horaires de travail.
Pré-condition : L'acteur s'authentié.
Post-condition : Les informations du planning sont affichées.
Scénario nominal : <ol style="list-style-type: none"> 1. L'acteur demande le formulaire «consulter planning». 2. Le système affiche le formulaire «consulter planning». 3. L'acteur saisit la date de planning. 4. Le système affiche les détails de programme de tell date. <ol style="list-style-type: none"> 4.1 Sinon affiche une notification « il n'y a aucun programme ».
Scénario d'exception : /

Tableau 2.10: Fiche descriptive textuelle de cas d'utilisation « Consulter planning ».

2.2.7.2. Diagramme d'activité de cas d'utilisation « Consulter planning »

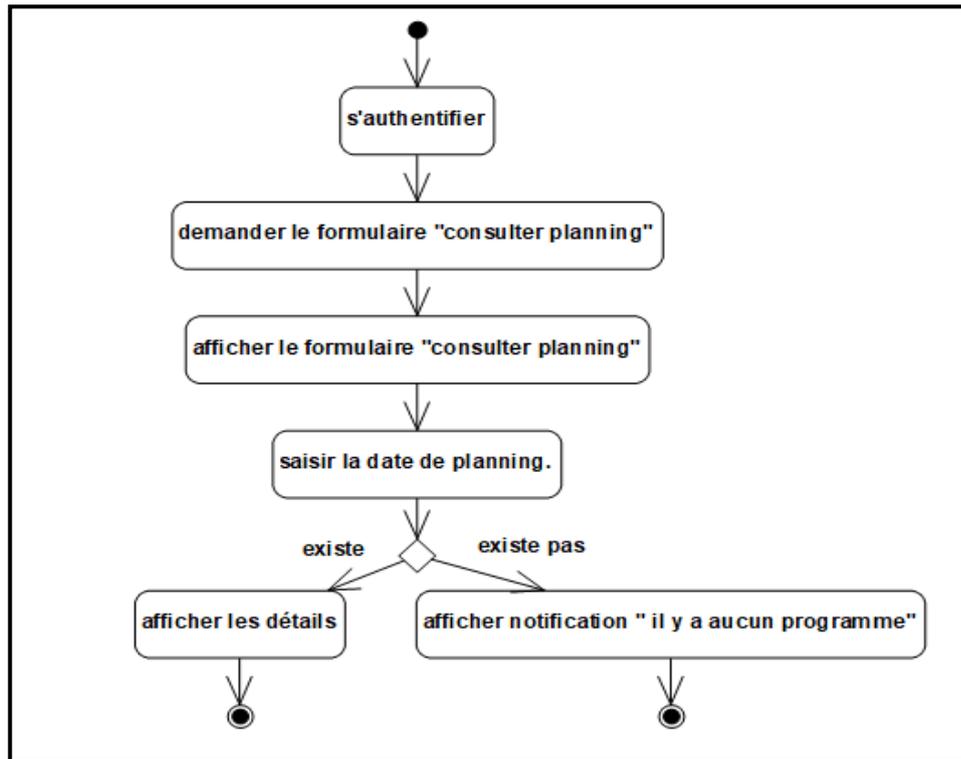


Figure 2.18: Diagramme d'activité de cas d'utilisation « Consulter planning ».

2.2.7.3. Diagramme de séquence de cas d'utilisation « Consulter planning »

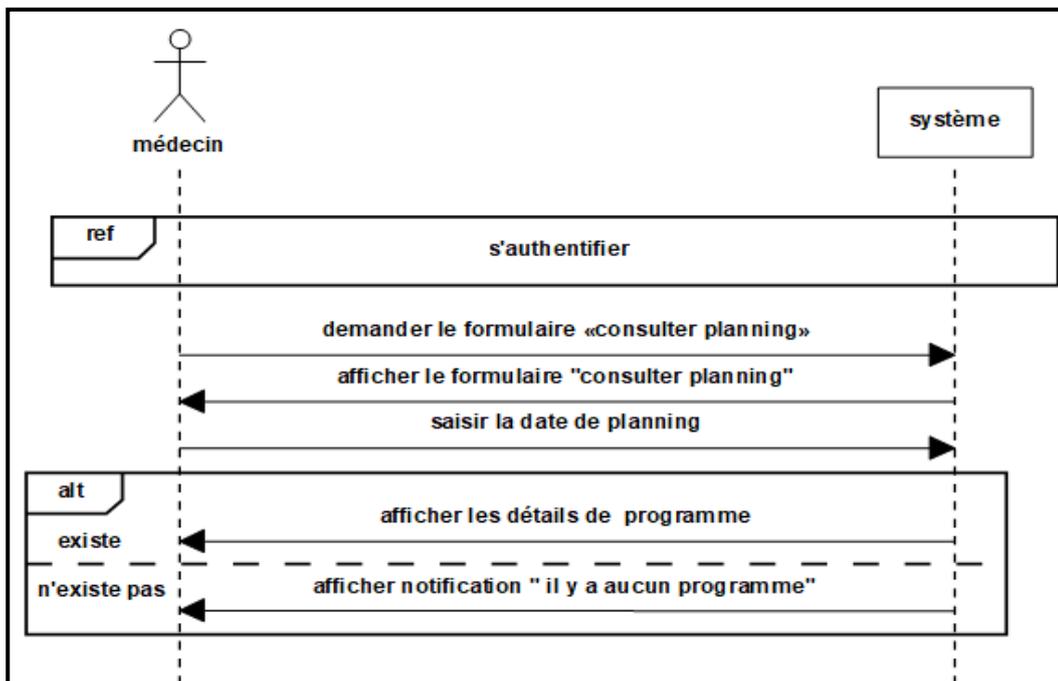


Figure 2.19: Diagramme de séquence de cas d'utilisation « Consulter planning ».

2.2.8. Cas d'utilisation « Prescrire patient »

2.2.8.1. Fiche descriptive

Titre : Prescrire patient.
Acteurs : Médecin.
But : Mise à jour l'état de patient.
Pré-condition : L'acteur s'authentifia.
Post-condition : Les nouvelles informations sur l'état du patient sont enregistrées.
Scénario nominal : <ol style="list-style-type: none">1. Le médecin demande le formulaire «prescription».2. Le système affiche le formulaire «prescription».3. Le système affiche la liste des patients affecté.4. Le médecin saisie le motif de recherche.5. Le système lit les informations saisies.6. Le système affiche le patient sélectionné.7. Le médecin saisit prescription.8. Le système vérifie les informations saisies.9. Le système affiche une notification « prescription ajouté avec succès ».
Scénario d'exception : /

Tableau 2.11: Fiche descriptive textuelle de cas d'utilisation« Prescrire patient ».

2.2.8.2. Diagramme d'activité de cas d'utilisation « Prescrire patient »

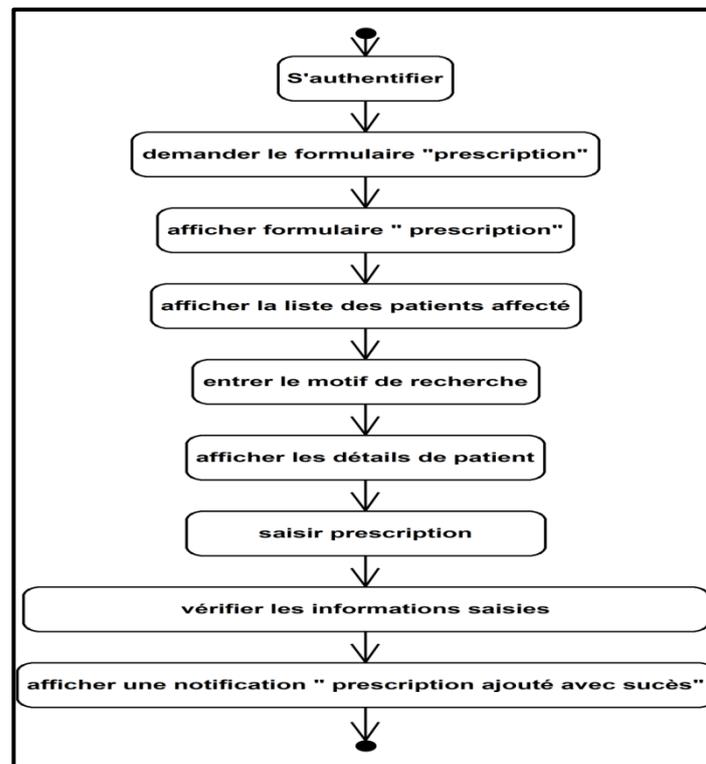


Figure 2.20:Diagramme d'activité de cas d'utilisation« Prescrire patient ».

2.2.8.3. Diagramme de séquence de cas d'utilisation « Prescrire patient »

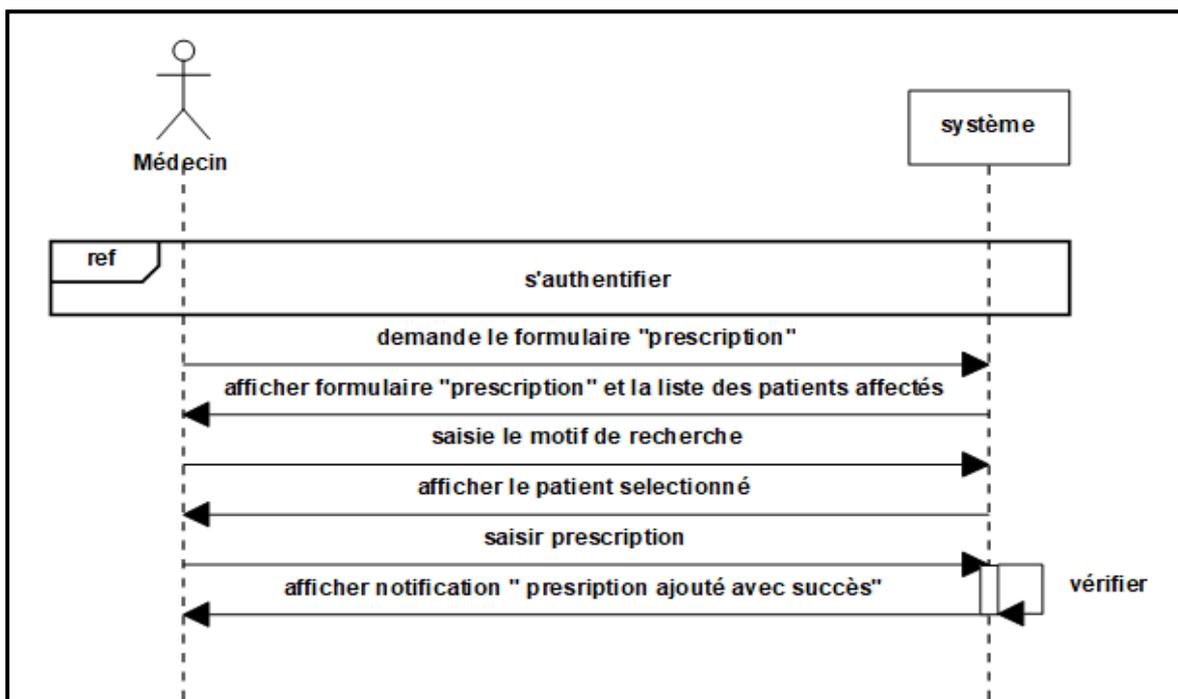


Figure 2.21: Diagramme de séquence de cas d'utilisation « Prescrire patient ».

2.2.9. Cas d'utilisation « Décision de la sortie d'un patient »

2.2.9.1. Fiche descriptive

Titre : Décision de sortir patient.
Acteurs : Médecin.
But : Permission de sortir patient.
Pré-condition : L'acteur s'authentié.
Post-condition : faire sortir un patient.
Scénario nominal : <ol style="list-style-type: none"> 1. Le médecin demande le formulaire « décision de sortie». 2. Le système affiche le formulaire « décision de sortie». 3. Le système affiche la liste des patients affecté. 4. Le médecin saisie le motif de recherche. 5. Le système affiche les détails du patient. 6. Le médecin saisit la date et le mode de sortie du patient. 7. Le système lit les informations saisies et demande la confirmation. 8. Le médecin confirme la demande. 9. Le système affiche une notification « décision de faire sortir du patient est ajouté avec succès ». 10. Le système met à jour l'état d'occupation du lit.
Scénario d'exception : /

Tableau 2.12: Fiche descriptive textuelle de cas d'utilisation « Décision de sortir patient ».

2.2.9.2. Diagramme d'activité de cas d'utilisation « Décision de sortie patient »

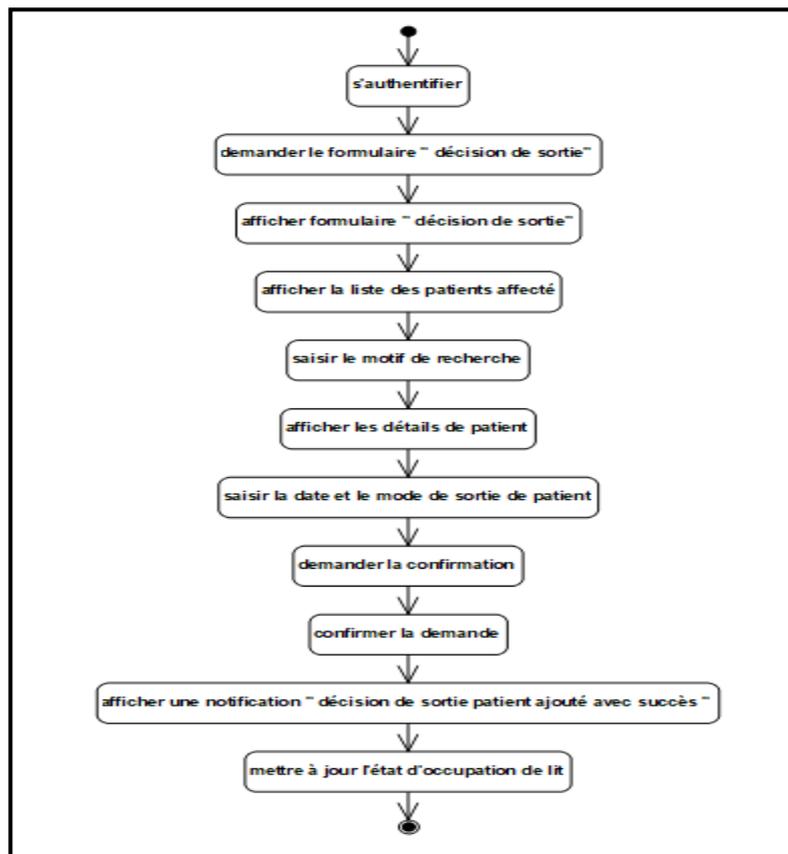


Figure 2.22:Diagramme d'activité de cas d'utilisation « Décision de sortir patient ».

2.2.9.3. Diagramme de séquence de cas d'utilisation « Décision de sortie patient »

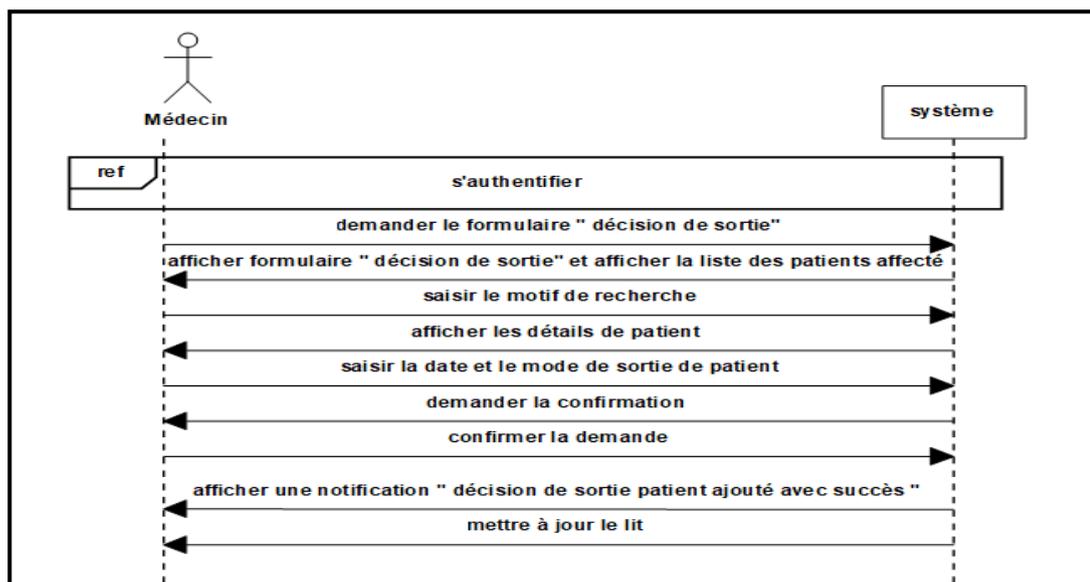


Figure 2.23: Diagramme de séquence de cas d'utilisation « Décision de sortir patient ».

2.2.10. Cas d'utilisation « Faire le bilan »

2.2.10.1. Fiche descriptive

Titre : Faire le bilan
Acteurs : Infirmier.
But : sauvegarder l'état du patient.
Pré-condition : S'authentifier.
Post-condition : L'état du patient est enregistré.
Scénario nominal : <ol style="list-style-type: none">1. L'infirmier demande le formulaire « Bilans».2. Le système affiche le formulaire « Bilans».3. Le système affiche la liste des patients affecté.4. L'infirmier sélectionne le patient et saisit les résultats de bilan.5. Le système lit les informations saisies.6. Le système vérifie et demander la confirmation.7. Le système affiche une notification « bilan ajouté avec succès ».
Scénario d'exception : annuler la demande.

Tableau 2.13: Fiche descriptive textuelle de cas d'utilisation « Faire le bilan ».

2.2.10.2. Diagramme d'activité de cas d'utilisation « Faire le bilan »

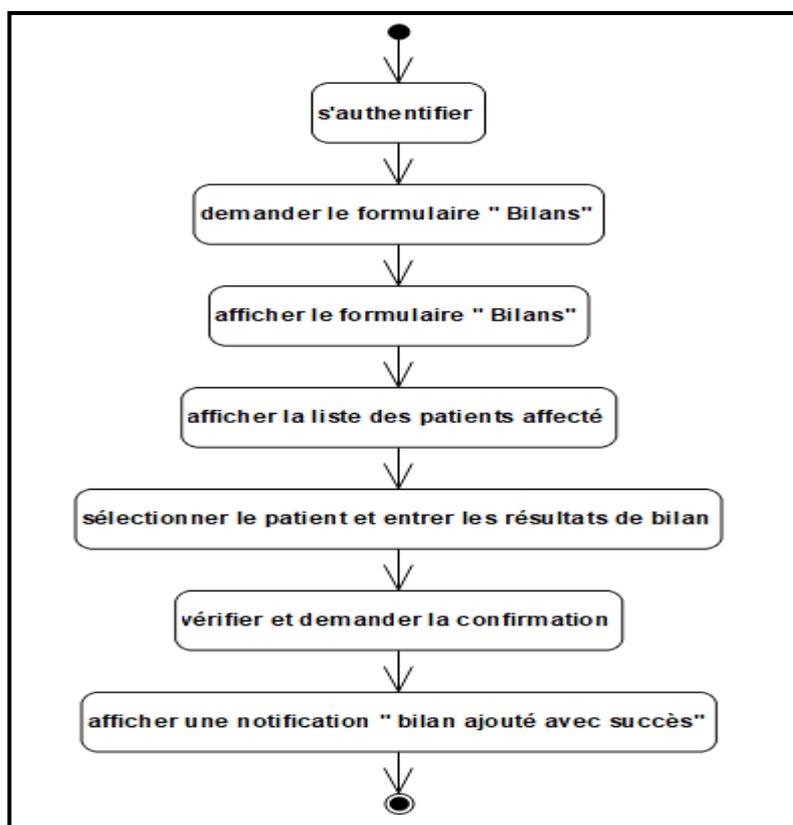


Figure 2.24: Diagramme d'activité de cas d'utilisation « Faire le bilan ».

2.2.10.3. Diagramme de séquence de cas d'utilisation « Faire le bilan »

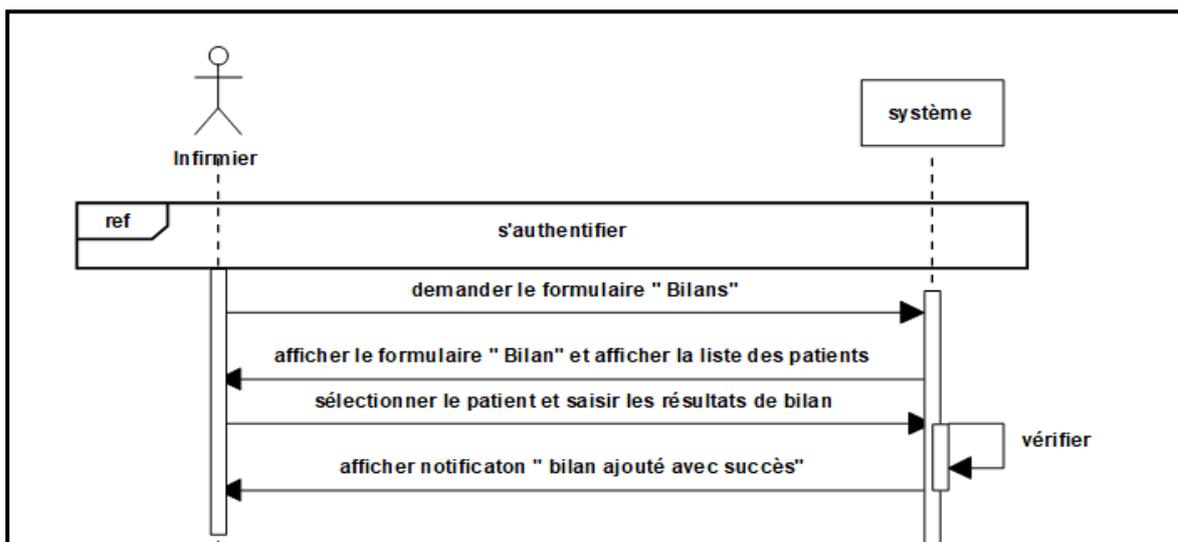


Figure 2.25: Diagramme de séquence de cas d'utilisation « Faire le bilan ».

2.2.11. Cas d'utilisation « Livraison des produits »

2.2.11.1. Fiche descriptive

Titre : livraison des produits.
Acteurs : Pharmacien.
But : Mise à jour de stock.
Pré-condition : S'authentifie.
Post-condition : Mis à jour de l'état du stock.
<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1. Le pharmacien demande le formulaire « livraison produit ». 2. Le système affiche le formulaire « livraison produit ». 3. Le système lit les informations saisies. <ul style="list-style-type: none"> • Cas il y a de commande <ol style="list-style-type: none"> 3.1 Le système affiche la commande de la journée. 3.2 Le pharmacien livre les commandes envoyées. 3.3 Le système lit l'information saisie. 3.4 Le système vérifie l'existence de produit. <ol style="list-style-type: none"> a. Si le produit existe : <ol style="list-style-type: none"> a.1 Le système affiche une notification « commande livrée ». a.2 le système diminue la quantité de produit en stock. b. Sinon afficher une notification « stock de produit insuffisant ». • Cas pas de commande 3.5 Le système affiche une notification « il n'y a aucune commande ».
Scénario d'exception : /

Tableau 2.14: Fiche descriptive textuelle de cas d'utilisation « Livraison des produits ».

2.2.11.2. Diagramme d'activité de cas d'utilisation « Livraison des produits »

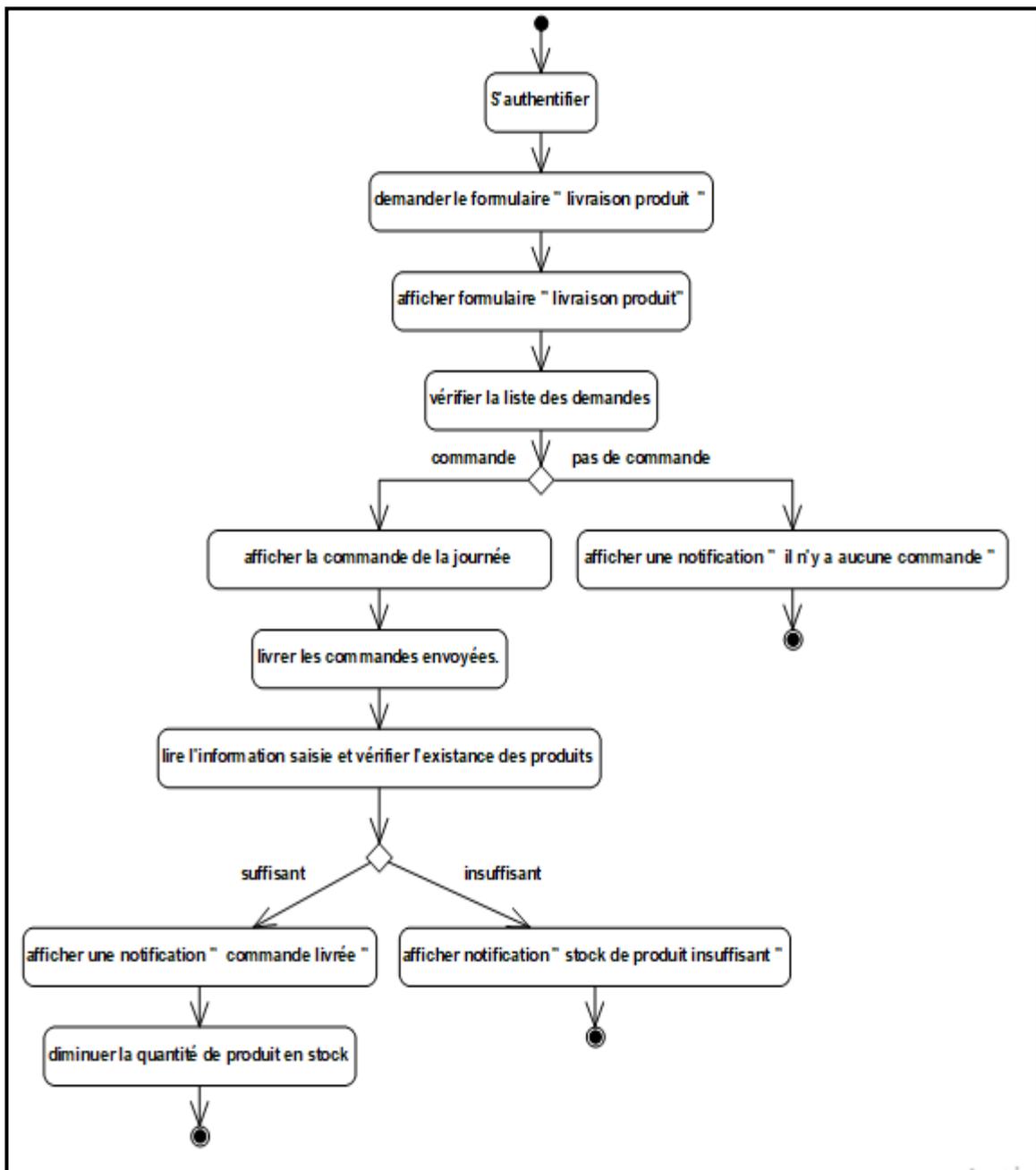


Figure 2.26:Diagramme d'activité de cas d'utilisation « Livraison des produits ».

2.2.11.3. Diagramme de séquence de cas d'utilisation « Livraison des produits »

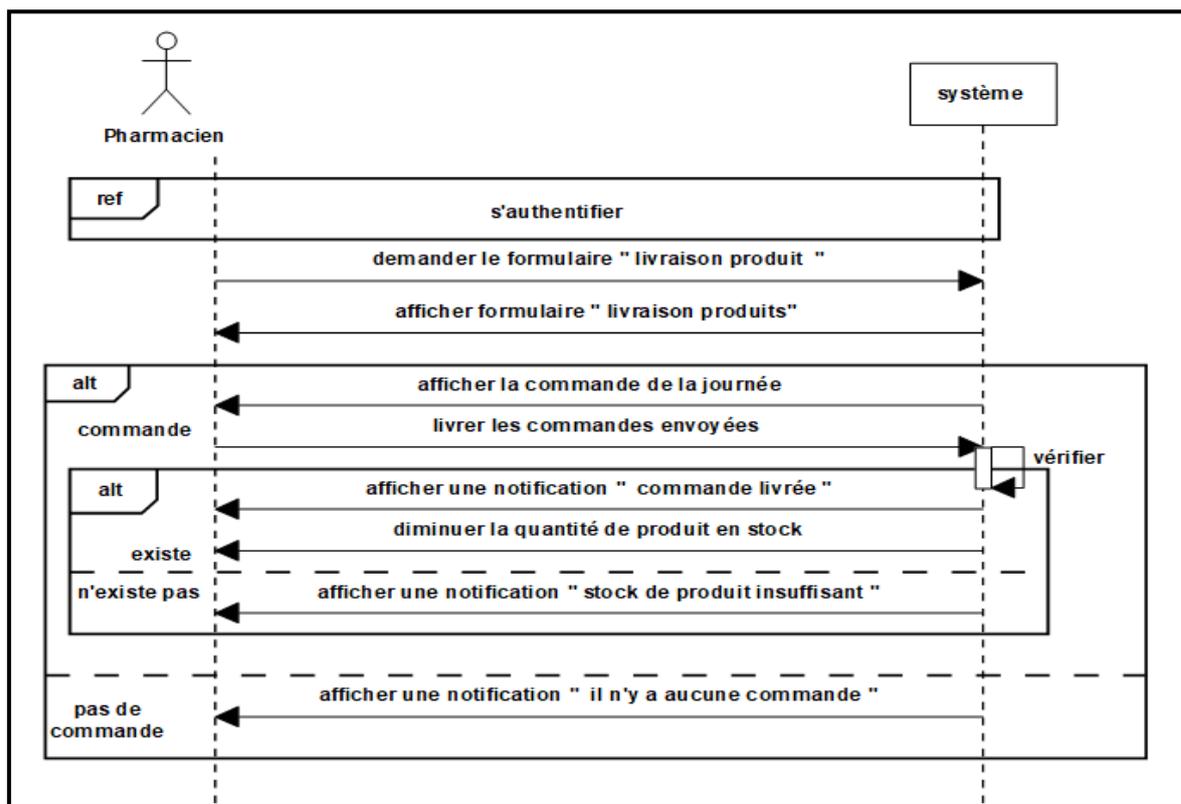


Figure 2.27: Diagramme de séquence de cas d'utilisation « Livraison des produits ».

2.2.12. Cas d'utilisation « Entrer le stock »

2.2.12.1. Fiche descriptive

Titre : Entrer stock.
Acteurs : Pharmacien.
But : L'ajout de la quantité des produits.
Pré-condition : S'authentifié.
Post-condition : la quantité en stock.
Scénario nominal : <ol style="list-style-type: none"> 1. Le pharmacien demande le formulaire « Entré stock ». 2. Le système affiche le formulaire « Entré stock». 3. Le système affiche la liste des produits. 4. Le pharmacien sélectionne le produit et saisie la nouvelle quantité. 5. Le système lit les informations saisies. 6. Le système vérifie et affiche notification « mouvement ajouté avec succès ». 7. Le système met à jour la quantité du produit.
Scénario d'exception : /

Tableau 2.15: Fiche descriptive textuelle de cas d'utilisation « Entré stock ».

2.2.12.2. Diagramme d'activité de cas d'utilisation « Entré stock »

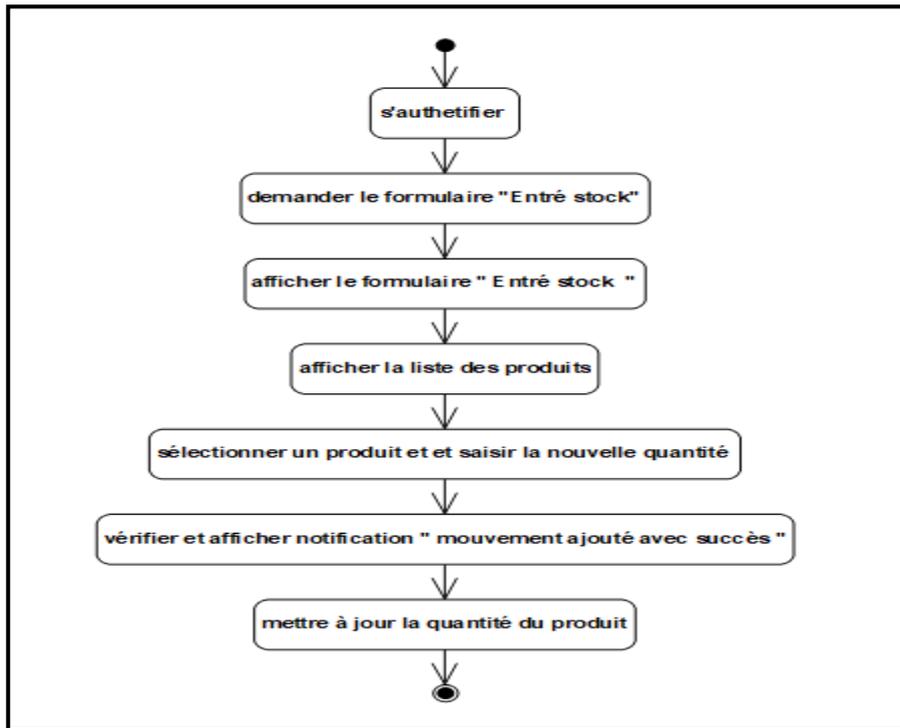


Figure 2.28: Diagramme d'activité de cas d'utilisation « Entré stock ».

2.2.12.3. Diagramme de séquence de cas d'utilisation « Entré stock »

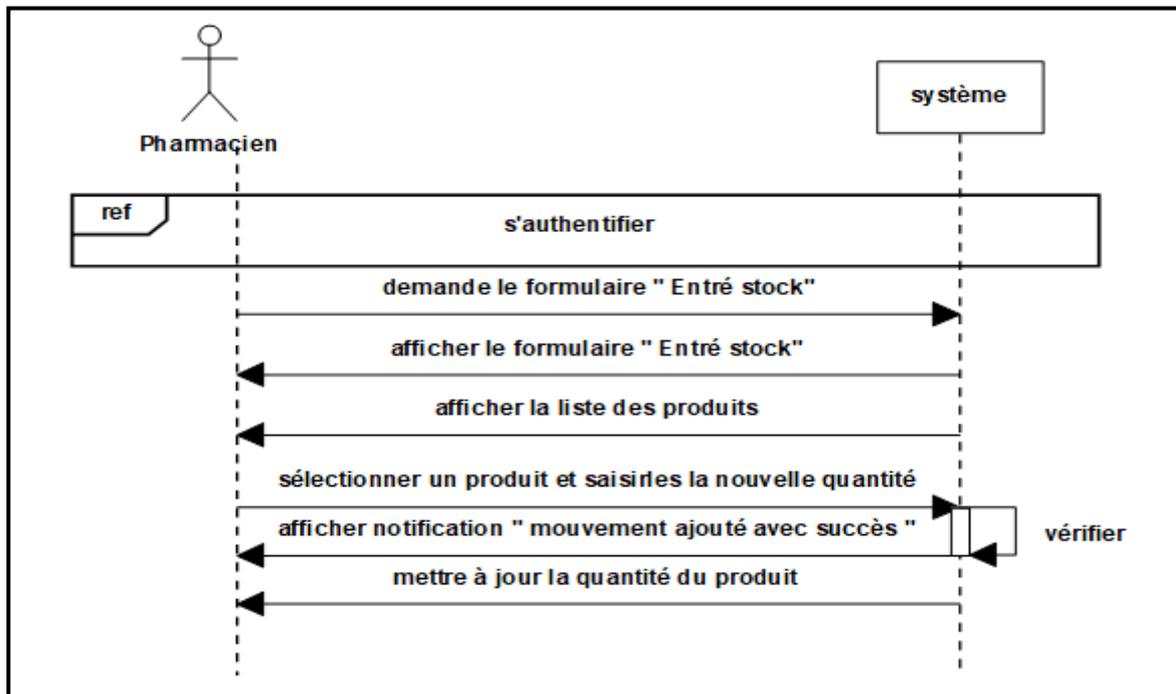


Figure 2.29: Diagramme de séquence de cas d'utilisation « Entré stock ».

2.2.13. Cas d'utilisation « Gérer produit »

2.2.13.1. Fiche descriptive

Titre : Gérer produit.
Acteurs : Pharmacien.
But : Identifier un produit.
Pré-condition : S'authentifie.
Post-condition : Identifier le produit.
Scénario nominal : <ol style="list-style-type: none">1. Le pharmacien demande le formulaire « gestion produit ».2. Le système affiche le formulaire « gestion produit ».• Cas de l'ajout d'un produit<ol style="list-style-type: none">3. Le pharmacien saisie les informations du produit.4. Le système vérifie les informations saisies et affiche une notification « produit ajouté avec succès ».• Cas de modification<ol style="list-style-type: none">5. Le système affiche la liste des produits.6. Pharmacien sélectionne le produit.7. Le système affiche les détails du produit.8. Pharmacien saisit les modifications.9. Le système vérifie et demande la confirmation.10. Pharmacien confirme les modifications.11. Le système affiche une notification « produit modifié avec succès ».• Cas de suppression<ol style="list-style-type: none">12. Le système affiche la liste des produits.13. Pharmacien sélectionne un produit.14. système affiche les détails du produit.15. Pharmacien supprime le produit.16. Le système vérifie et demande la confirmation.17. Pharmacien confirme la suppression.18. Le système affiche une notification « produit supprimé avec succès ».19. Sinon afficher une notification « interdit de supprimer le produit il y a des mouvements ».
Scénario d'exception : /

Tableau 2.16: Fiche descriptive textuelle de cas d'utilisation « Gérer produits ».

2.2.13.2. Diagramme d'activité de cas d'utilisation « Gérer produits »

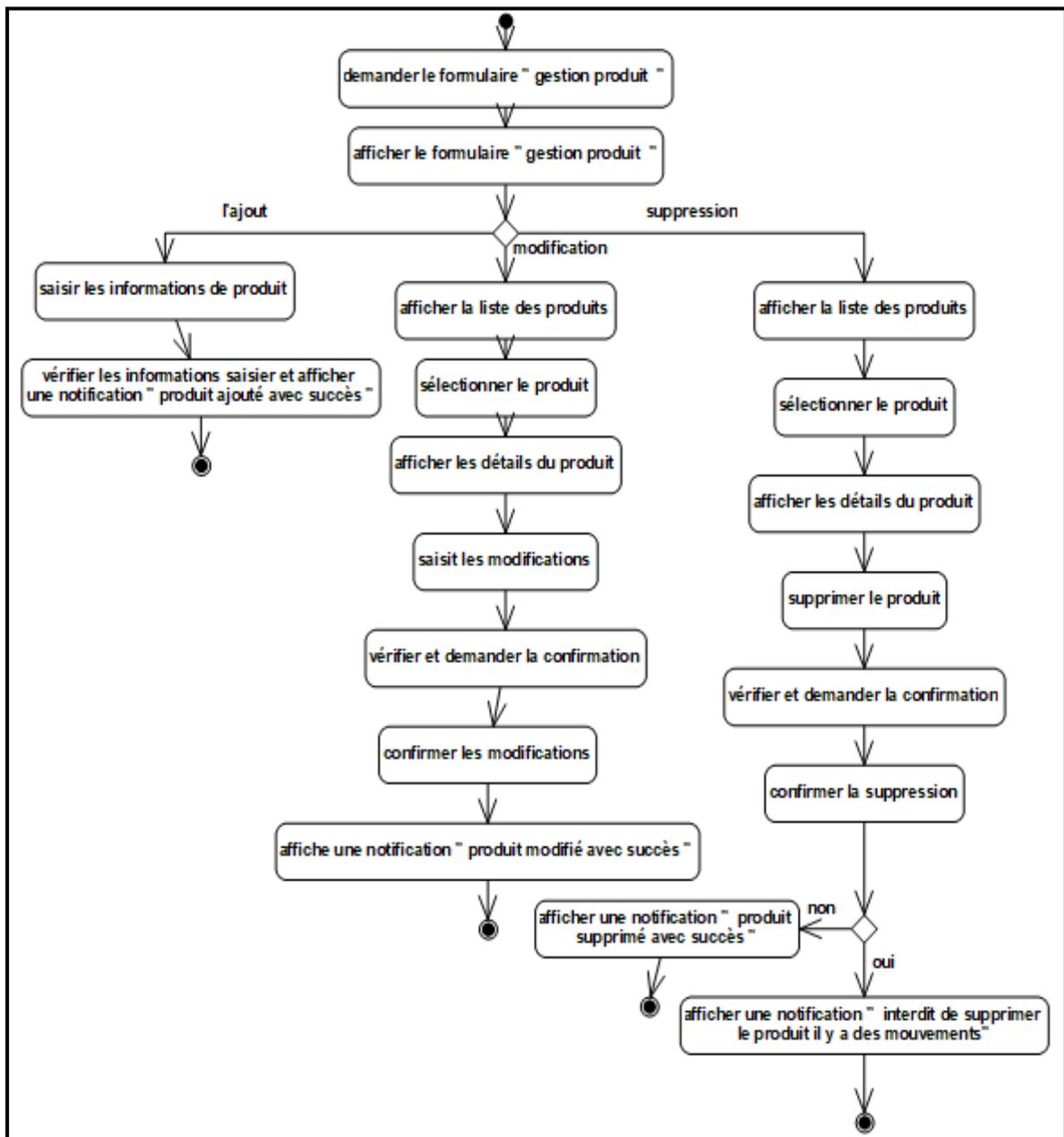


Figure 2.30: Diagramme d'activité de cas d'utilisation « Gérer produits ».

2.2.13.3. Diagramme de séquence de cas d'utilisation « Gérer produits »

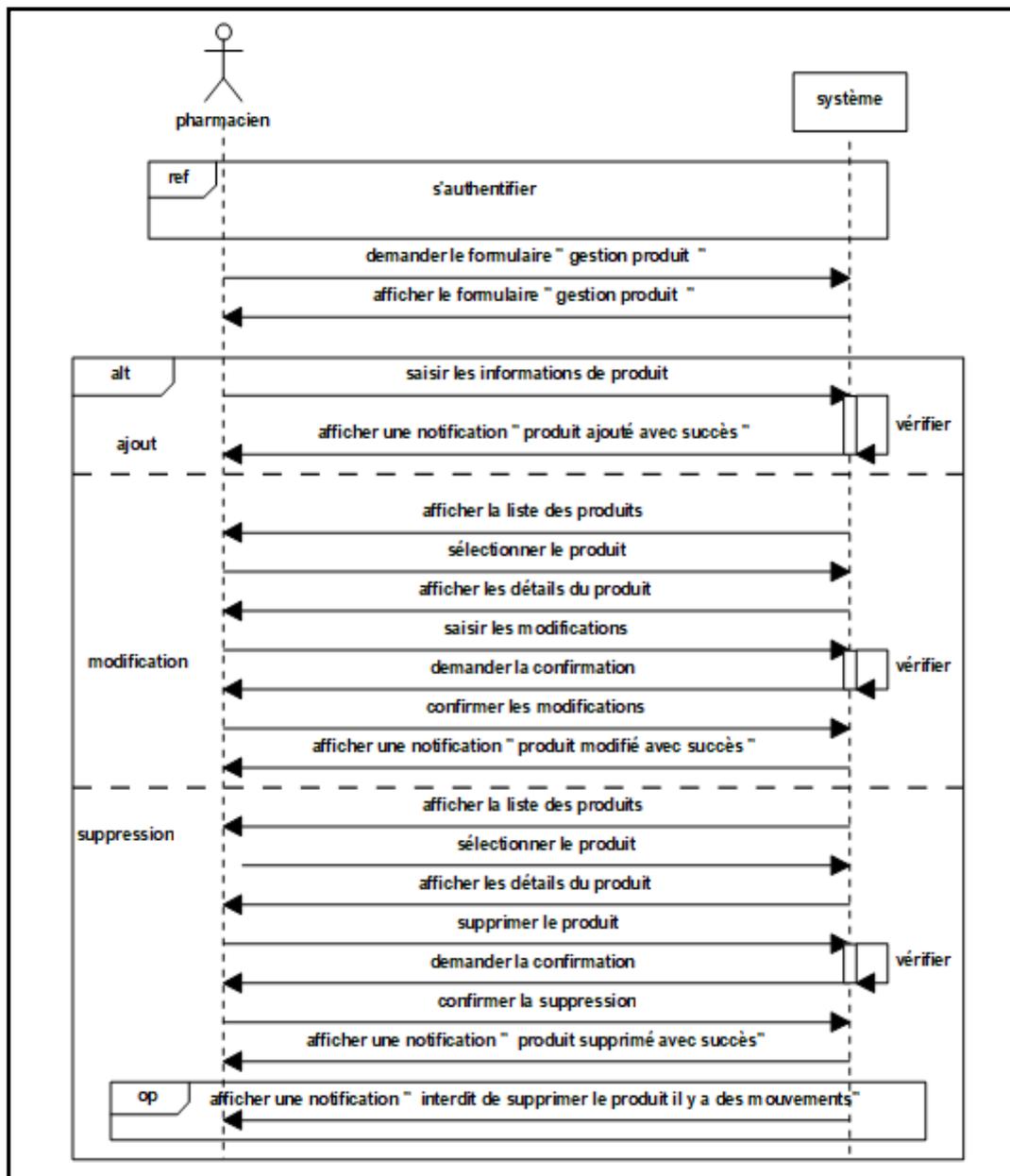


Figure 2.31: Diagramme de séquence de cas d'utilisation « Gérer produits ».

3. Spécification des besoins techniques

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

Cette étape peut être appréhendée lorsque les concepteurs de l'application ont obtenu suffisamment d'informations sur l'environnement de travail ainsi que l'architecture globale utilisée pour le système et les près requis technique. La capture des besoins techniques se présente comme suit :

- Capture des spécifications matérielles.
- Capture des spécifications logicielles.

3.1. Capture des spécifications matérielles

3.1.1. Configuration matérielle

Caractéristiques techniques du :

- ❖ Serveur :
 - MySQL.
- ❖ Client :
 - N postes.
 - Windows work station (7, 8, 9,10).
 - Imprimante.
- ❖ Réseau :
 - Routeur CISCO
 - Réseau local LAN 10/100mbps.

3.1.2. Spécification du style d'architecture 2 niveaux

L'architecture à deux niveaux (aussi appelée architecture 2-tiers) caractérise les systèmes clients/serveurs dans lesquels le client demande un service et le serveur le lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.

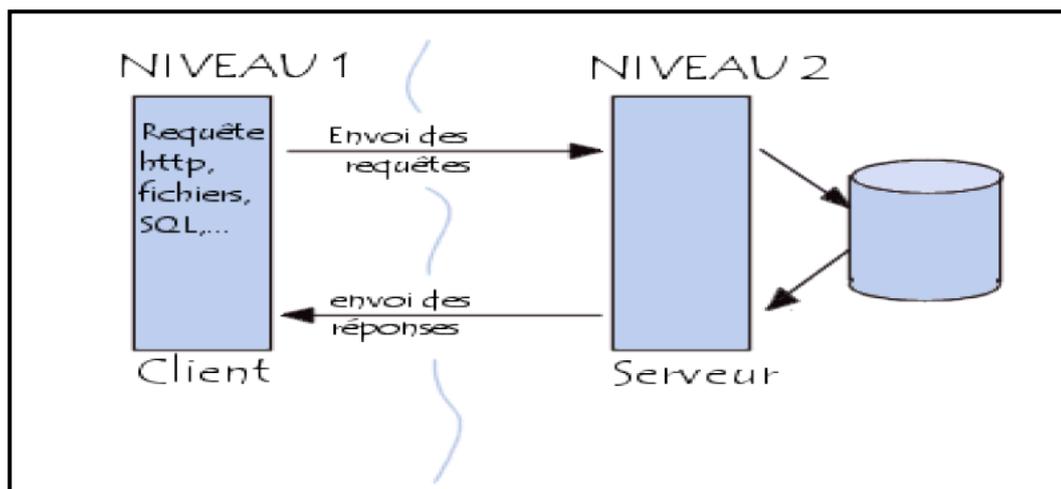


Figure2.32 : L'architecture 2-tiers.

3.2. Capture des spécifications logicielles

Une fois que les spécifications techniques et d'architecture 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'utilisations de manière différente que pour la spécification fonctionnelle. C'est pourquoi nous avons introduit le concept d'exploitant et de cas d'utilisation technique.

- **Exploitant:** c'est un utilisateur dans son sens le plus général qui bénéficie des fonctionnalités techniques du système.
- **Cas d'utilisation technique (CUT):** un CUT destiné à l'exploitant est une séquence d'actions produisant une valeur ajoutée opérationnelle ou technique mais pas fonctionnelle.

3.2.1. Identification des exploitants du système

Les exploitants de notre système sont la majorité des acteurs de la branche fonctionnelle. Ce sont : l'administrateur, agent d'admission, médecin, infirmier et le pharmacien.

3.2.2. Identification des cas d'utilisation techniques

Les cas d'utilisation techniques de notre système sont essentiellement :

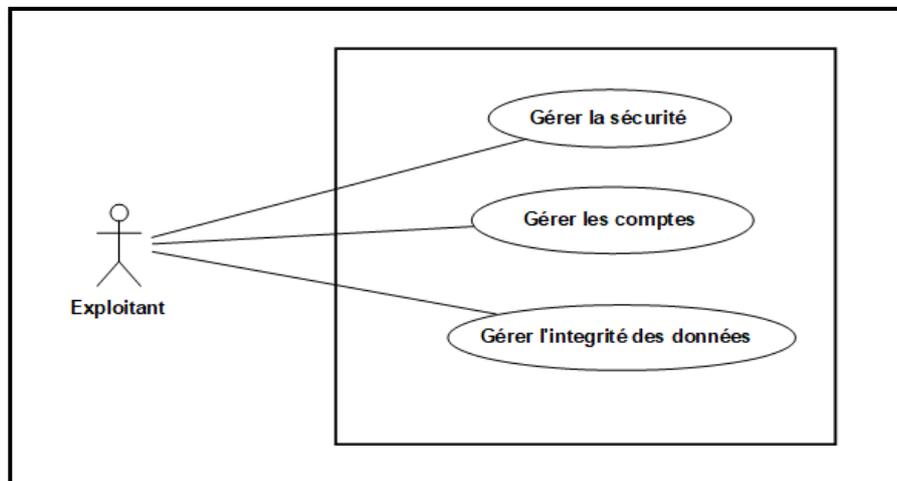


Figure 2.33: Modèle de spécification logicielle du système.

3.2.3. Description des cas d'utilisation techniques

3.2.3.1. Cas d'utilisation « Gérer la sécurité »

a. S'authentifier

- Fiche descriptive

Titre : S'authentifier.
But : Vérifier l'autorisation d'accéder au système.
Pré condition : L'utilisateur possède un compte.
Post condition : L'utilisateur est authentifié par le système.
Scénario nominal : <ol style="list-style-type: none">1. L'utilisateur demande au système d'ouvrir son compte pour entrer dans son espace de travail.2. Le système affiche le formulaire pour saisir les informations d'accès (le mot de passe et nom utilisateur).3. L'utilisateur saisie son username et mot de passe et valide.4. Le système vérifie la validité du mot de passe et de nom utilisateur et ouvre le compte demandé.
Scénario alternatif : <ul style="list-style-type: none">- Le username ou Le mot de passe est erroné.- Le système affiche une notification « Username ou mot de passe ou rôle est incorrect.- Le système propose à l'utilisateur de renseigner une nouvelle fois le mot de passe.- Reprise de l'enchaînement du scénario nominal au point 2.
Exception : /

Tableau 2.17 : Fiche descriptive textuelle de cas d'utilisation « S'authentifier ».

- Diagramme d'activité cas d'utilisation « S'authentifier »

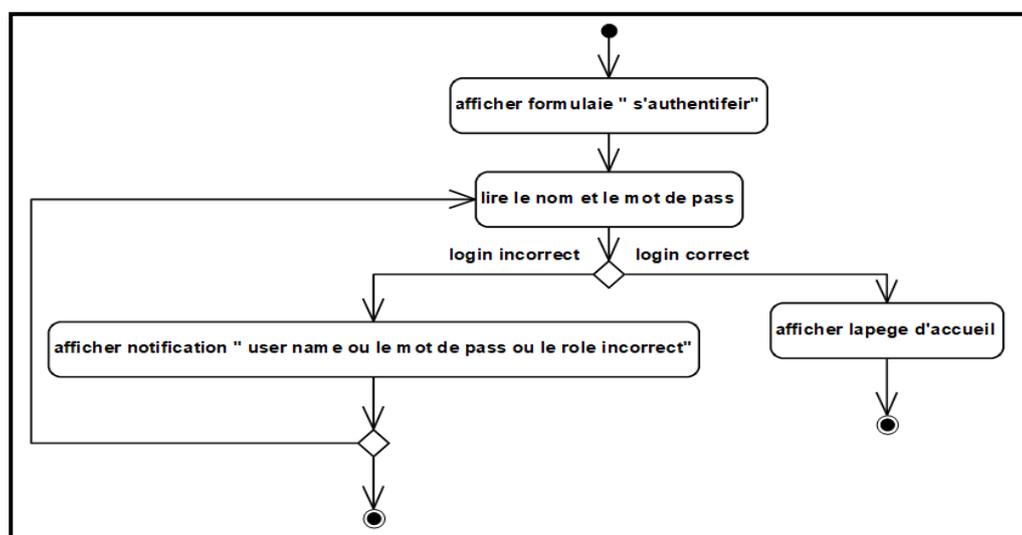


Figure 2.34: Diagramme d'activité de cas d'utilisation « S'authentifier ».

- Diagramme de séquence de cas d'utilisation « S'authentifier »

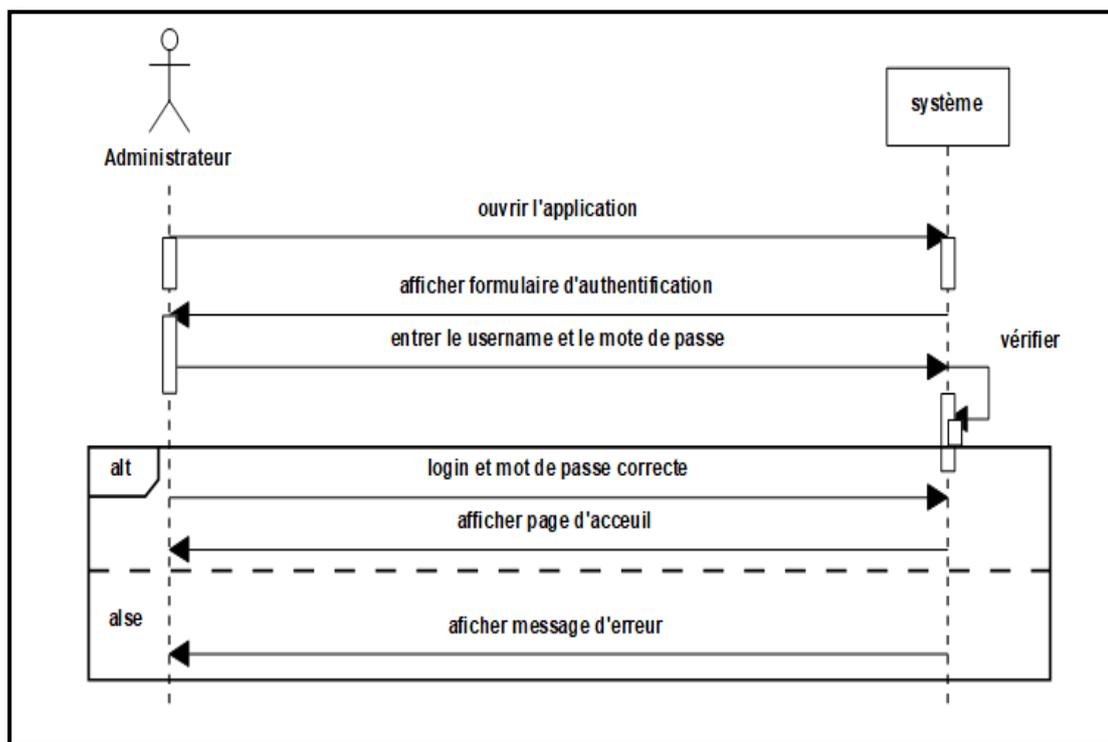


Figure 2.35: Diagramme de séquence de cas d'utilisation « S'authentifier ».

b. Gérer intégrité des données

- Fiche descriptive

Titre : Gérer intégrité des données.
Pré condition : L'utilisateur est authentifié et accède à un formulaire.
Scénario nominal : <ol style="list-style-type: none">1. L'utilisateur saisit les données.2. le système contrôle la validité de chaque champ.
Scénario alternatif : <ol style="list-style-type: none">1. Le système affiche une notification « donnée invalide ».2. Le scénario reprend au point 1.

Tableau 2.18 : Fiche descriptive textuelle de cas d'utilisation « Gérer intégrité des données ».

- Diagramme d'activité de cas d'utilisation « Gérer intégrité des données »

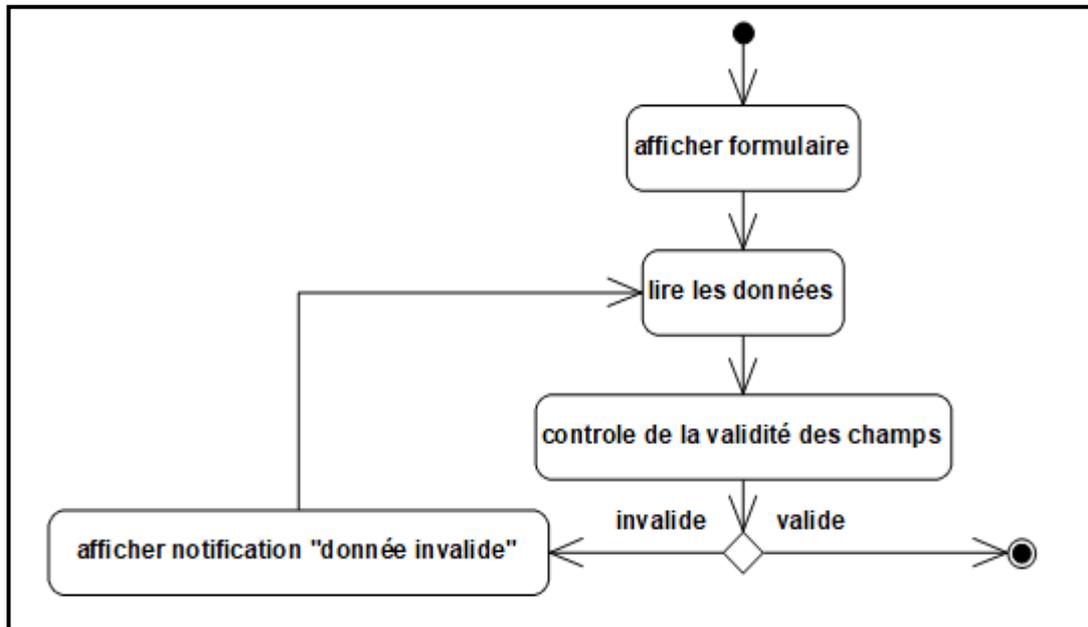


Figure 2.36: Diagramme d'activité de cas d'utilisation «Gérer l'intégrité des données ».

- Diagramme de séquence de cas d'utilisation « Gérer intégrité des données »

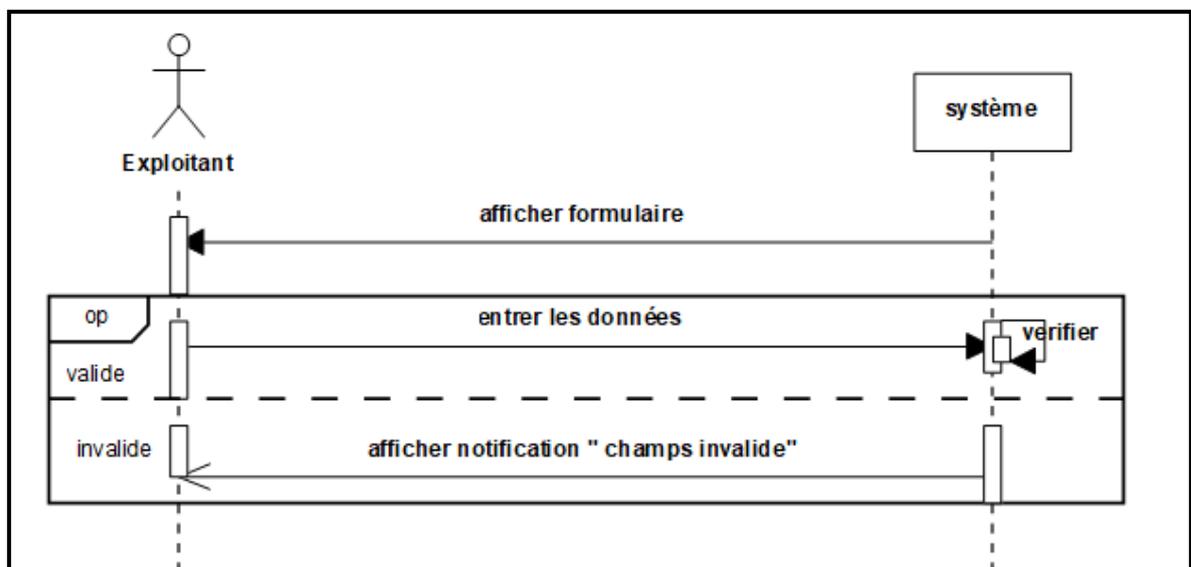


Figure 2.37: Diagramme de séquence de cas d'utilisation «Gérer l'intégrité des données ».

3.3. Organisation du modèle de spécification logicielle

La couche logicielle représente un ensemble de spécifications ou de réalisations qui respectivement expriment ou mettent en œuvre un ensemble de responsabilités techniques et homogènes pour un système logiciel.

Les couches s'empilent en niveaux pour couvrir des transformations logicielles successives, de sorte que la couche d'un niveau ne puisse utiliser que les services des couches de niveaux inférieurs.

Le modèle de spécification logiciel est conditionné par l'architecture déployée. Dans notre cas avec l'architecture 2-tiers, notre système est séparé en trois couches logicielles.

3.3.1. La couche présentation ou IHM (Interface Homme/Machine) : gère les interactions utilisateurs/machine (la présentation).

3.3.2. La couche traitements

- Locaux : contrôles effectués au niveau du dialogue avec l'IHM.
- Globaux : L'application elle-même.

3.3.3. La couche donnée: Gère le stockage des données et l'accès à ces dernières.

- Les couches présentation et traitements sont implantées sur le client.
- La couche donnée est implantée sur le serveur.

Au cours de ce titre, l'architecture physique a été choisie selon l'environnement adopté, on a pris en compte toutes les contraintes techniques et logicielles pour le choix le plus adapté de notre architecture. Prochainement nous allons présenter la phase d'analyse.

4. Analyse

La phase d'analyse est consacrée à l'analyse objet de notre système, elle vient juste après l'étude préliminaire et la capture des besoins fonctionnels et techniques. Elle est constituée de trois (03) activités :

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

4.1. Découpage en catégorie

Le découpage en catégories constitue la première activité de l'étape d'analyse. Il 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.

4.1.1. Répartition des classes candidates en catégorie

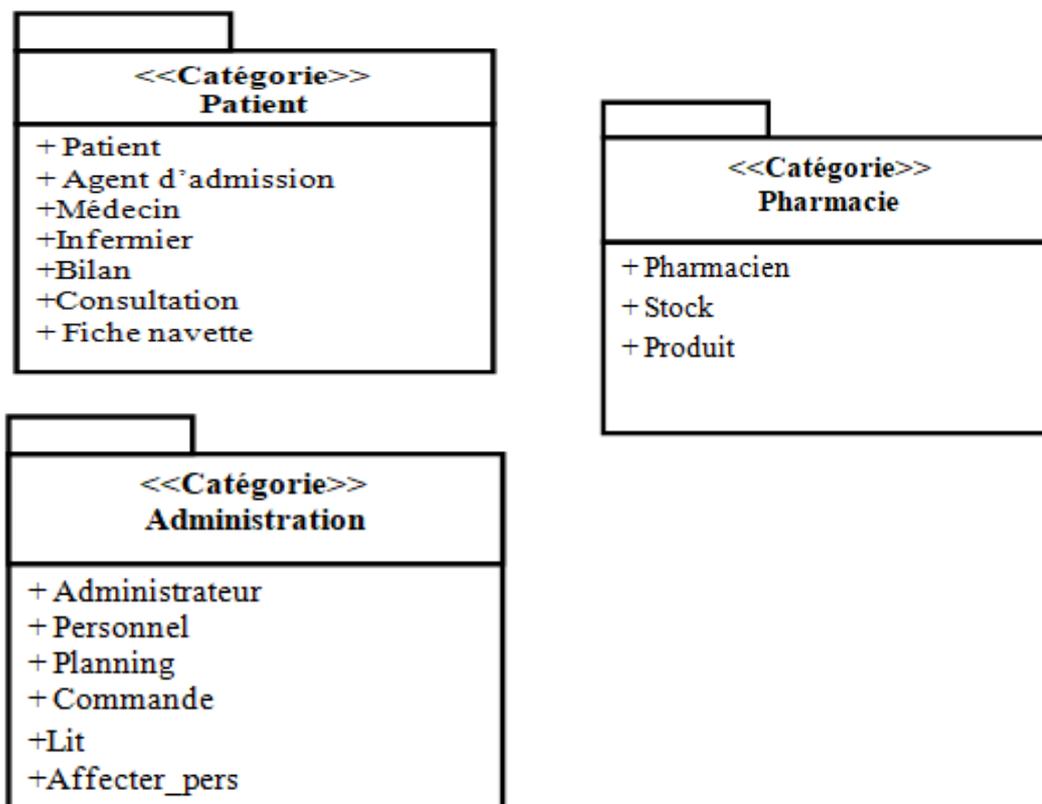


Figure 2.38: Découpage en catégories.

4.1.2. Elaboration des diagrammes de classes préliminaires par catégorie

4.1.2.1 La classe préliminaire de la catégorie «Patient»

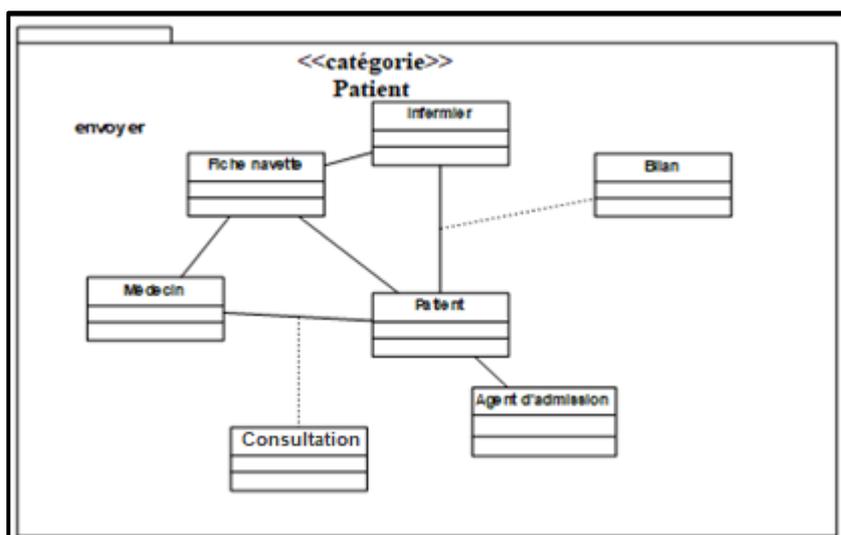


Figure 2.39 : Découpage de la classe de la « catégorie Patient ».

4.1.2.2 La classe préliminaire de la catégorie «Pharmacie»

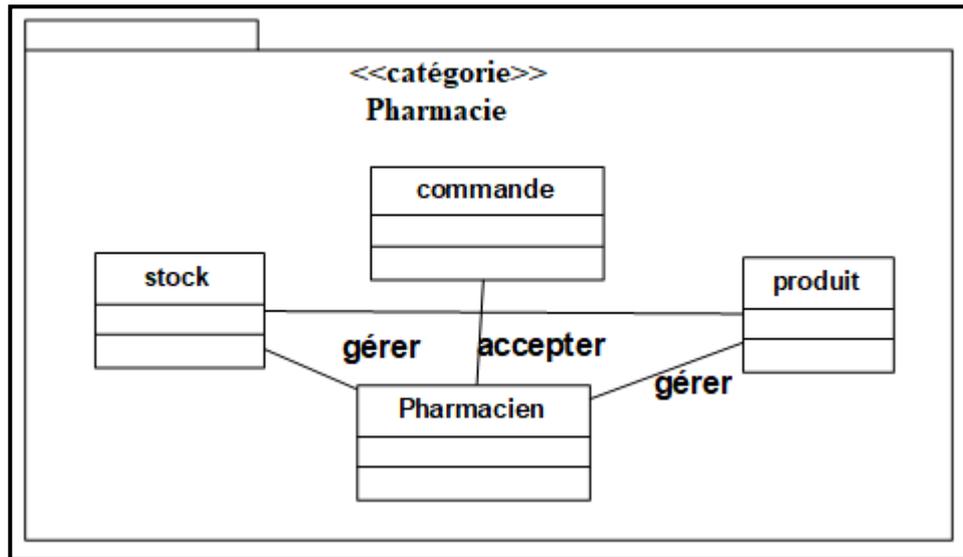


Figure 2.40 : Découpage de la classe de la « catégorie Pharmacie ».

4.1.2.3 La classe préliminaire de la catégorie «Administration»

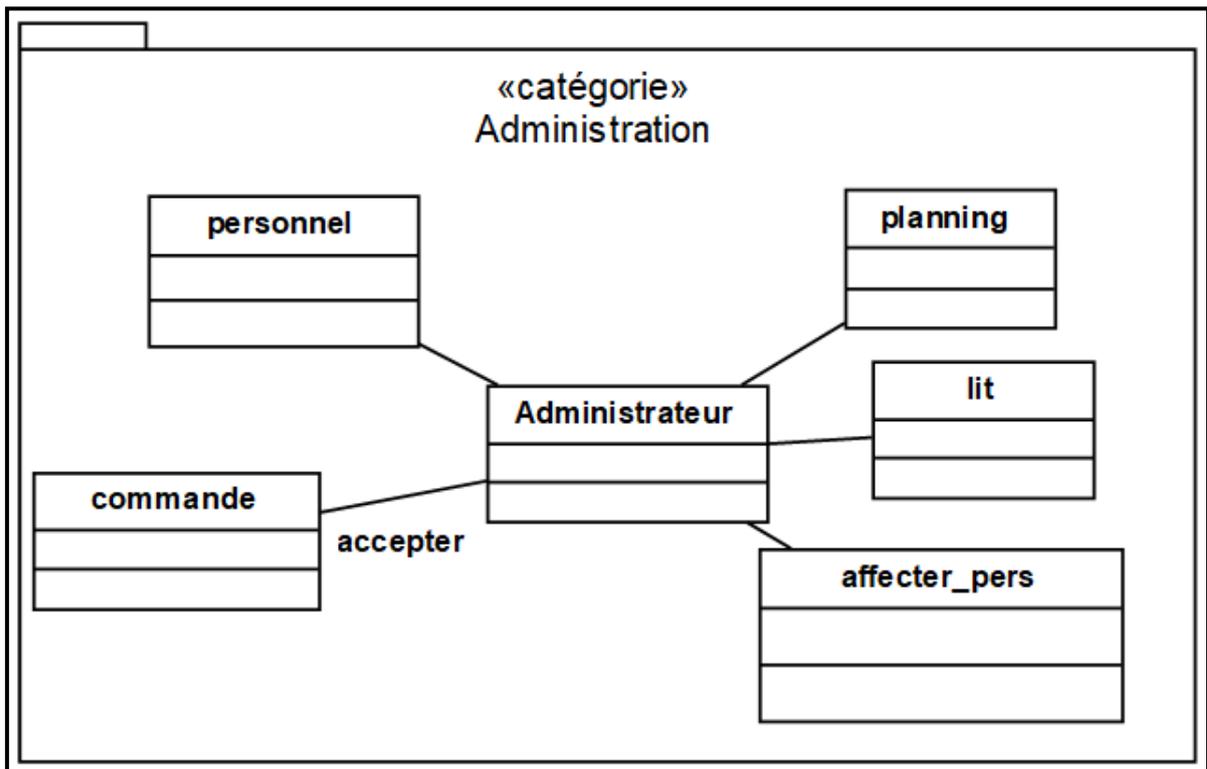


Figure 2.41 : Découpage de la classe de la « catégorie Administration ».

4.1.3. Dépendance entre catégorie

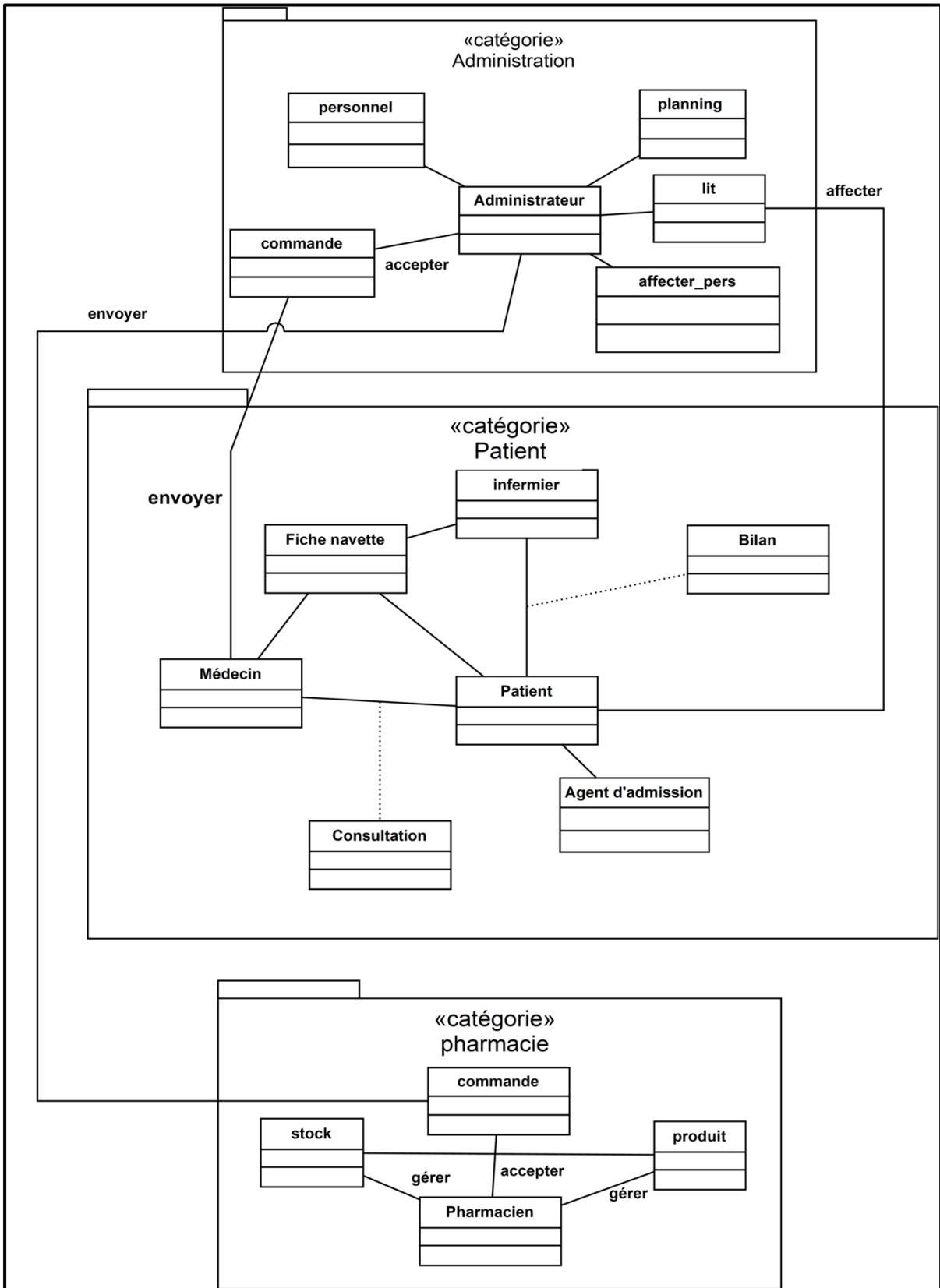


Figure 2.42 : Dépendance entre catégories.

4.2. Développement du modèle statique

Le développement en modèle statique représente la deuxième activité de l'étape d'analyse. Lors de cette étape, nous reprenons les diagrammes organisés lors du découpage en catégories afin de les affiner en leur ajoutant des attributs.

4.2.1. Diagramme de classe de la catégorie Patient

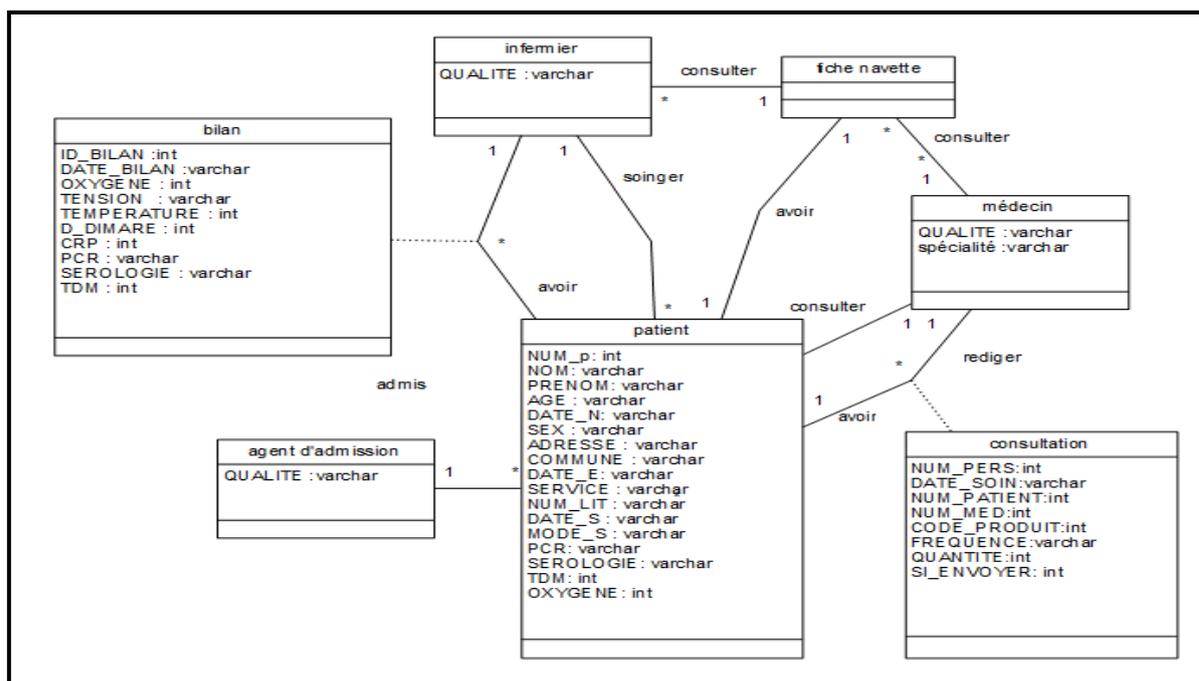


Figure 2.43: Diagramme de classe de la catégorie « Patient ».

4.2.2. Diagramme de classe de la catégorie Pharmacie

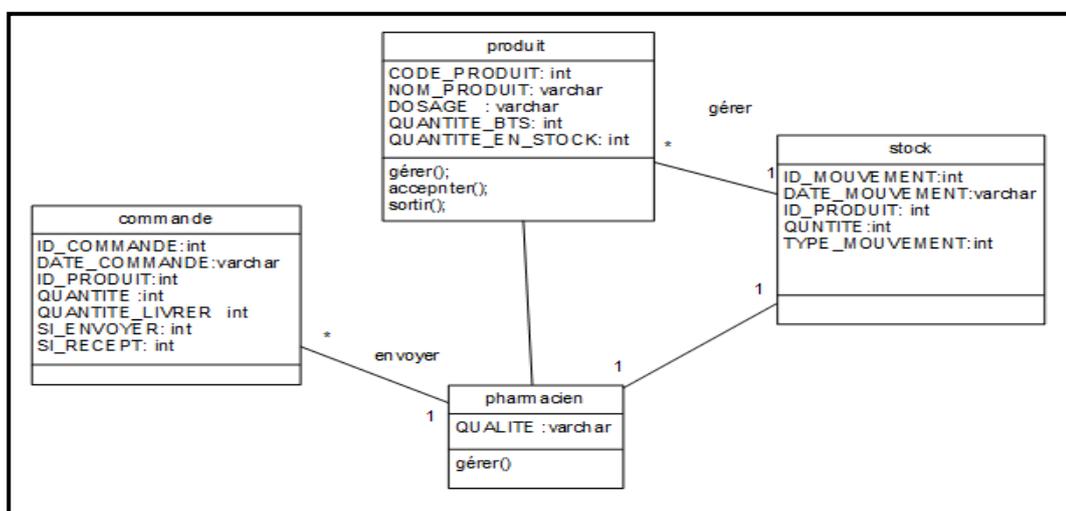


Figure 2.44 : Diagramme de classe de la catégorie « Pharmacie ».

4.2.3. Diagramme de classe de la catégorie Administration

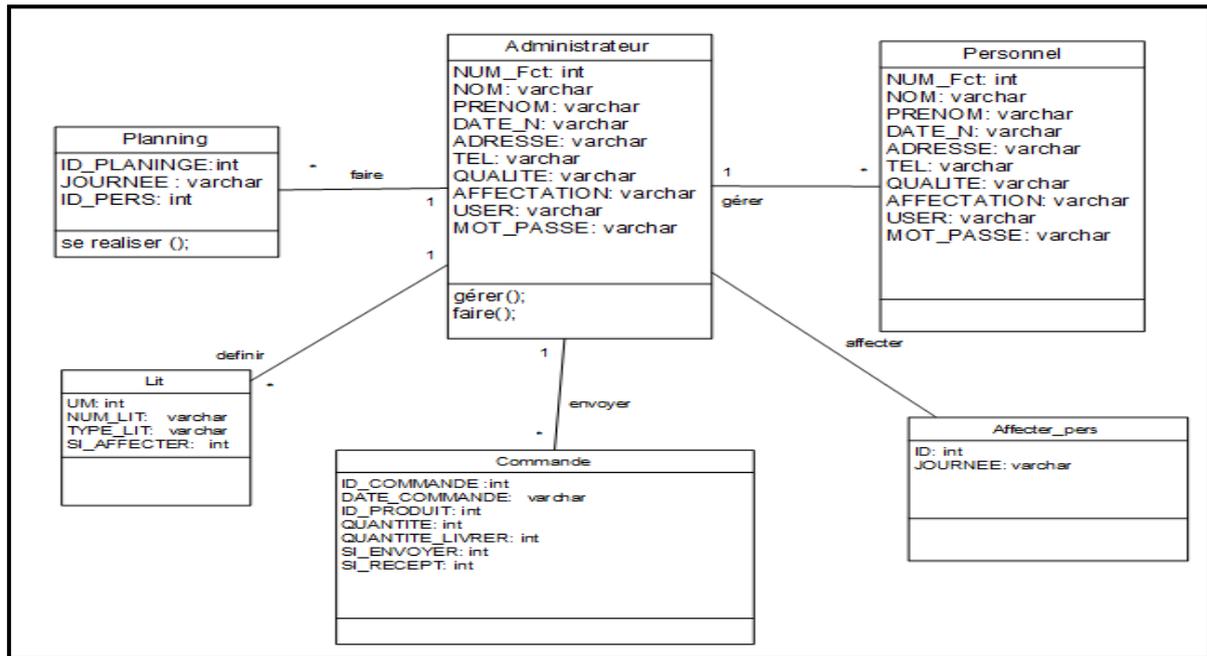


Figure 2.45 : Diagramme de classe de la catégorie « Administration ».

4.3. Développement du modèle dynamique

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

4.3.1. Diagrammes d'interactions

- Diagramme d'interaction de cas d'utilisation « Admettre patient »

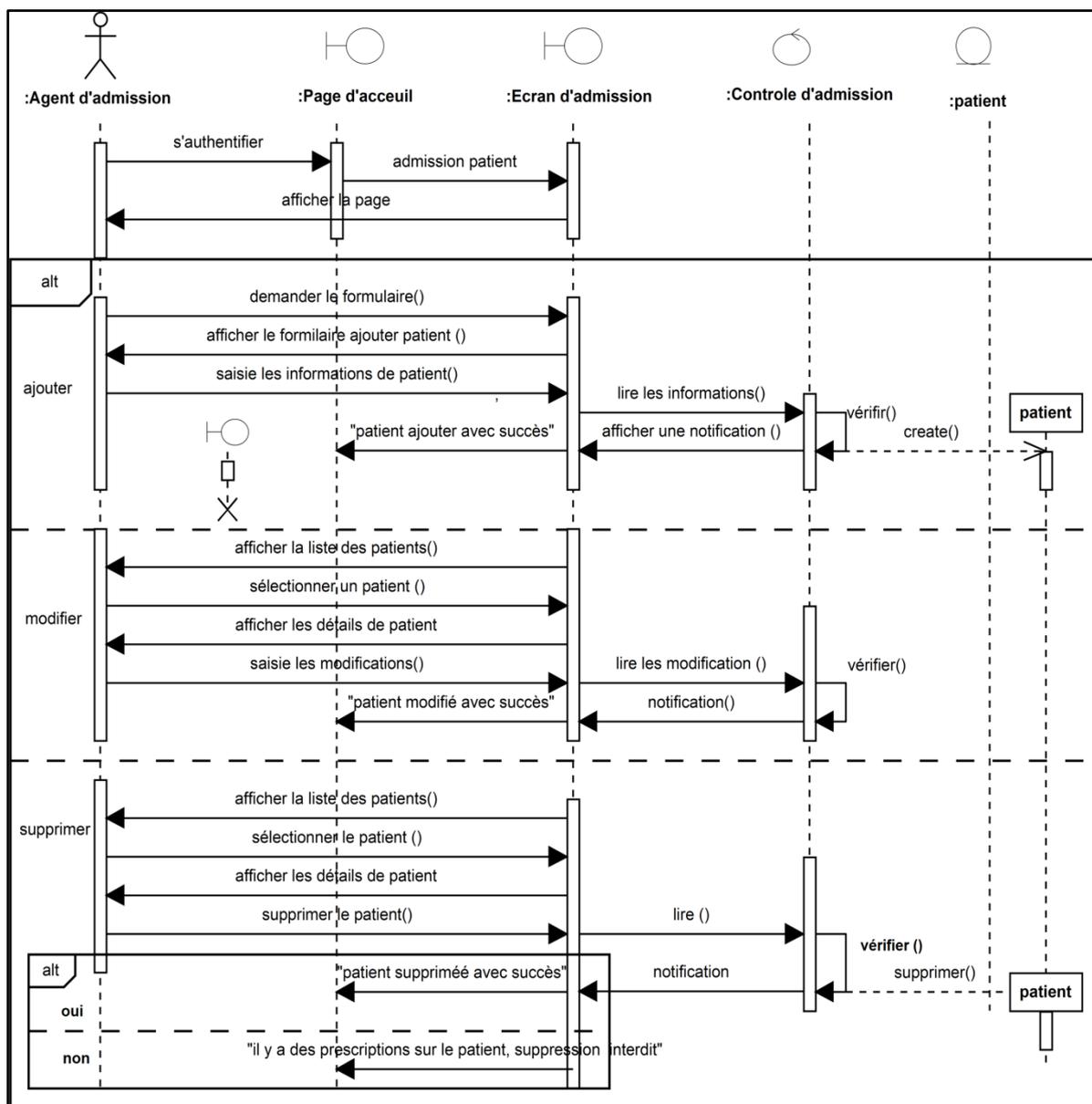


Figure 2.46 : Diagramme d'interaction de cas d'utilisation «Admettre patient ».

5. Conception préliminaire

La conception préliminaire est 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 technique. En conséquence, cette étape permet de passer de l'analyse objet à la conception, c'est-à-dire adapter la conception aux spécifications fournies par l'analyse et intégrer les fonctions métier et applicatives du système dans l'architecture technique.

5.1. Développement du modèle de déploiement

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système, le déploiement d'une solution client/serveur se construit sur la définition des postes de travail.

5.1.1. Architecture adoptée

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 :

- L'ensemble des utilisateurs se trouve dans la même zone géographique ;
- 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.

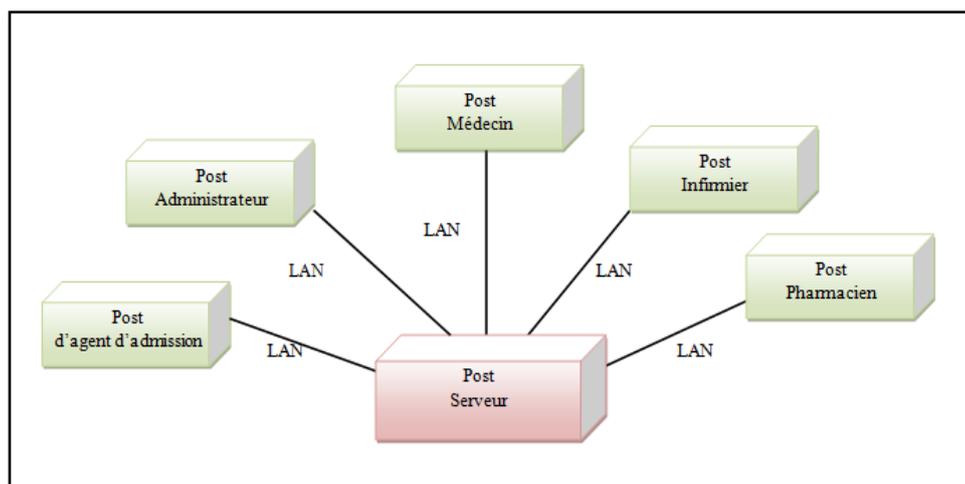


Figure 2.47 : Schéma du modèle de déploiement de notre système.

5.1.2. Déploiement du modèle d'exploitation

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

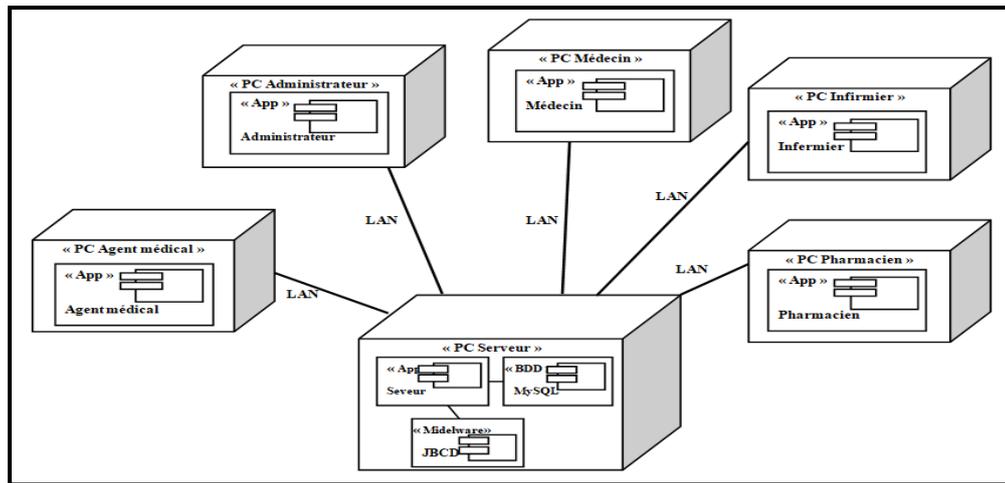


Figure 2.48: Définition des applications dans le modèle d'exploitation.

5.2. Définition des interfaces

Espace	Interface	Description
Administrateur	Gestion compte	Ajouter, modifier, et supprimer un compte
	Gestion planning	Définir le planning
	Gestion lits	Définir les lits
	Envoyer commandes	Envoyer la demande de service à la pharmacie
	Affectation du cadres médicaux	Affectation des ressources humaines au patient
	Affectation des médicaments	Affectation de ressources médicales au patient
Agent médical	Gestion d'admission	Mise à jour des dossiers patients
	Consulter la fiche navette	La fiche navette
Médecin	Prescription	Les soins des patients
	Consulter fiche navette	Mise à jour de la fiche navette
	Consulter planning	Consulter le planning propre
	Décision de sortie	Finaliser hospitalisation d'un patient
Infirmier	Faire le bilan	Le bilan quotidien de chaque patient
	Consulter la fiche navette	Mise à jour de la fiche navette
	Consulter planning	Consulter le planning propre
Pharmacien	Gestions produits	Mouvement des produits
	Entré stock	Entrer les nouveaux médicaments au système
	Livraison produit	Demande des médicaments

Tableau 2.19 : Les interfaces de notre système.

6. Conception détaillée

La conception détaillée est la phase ultime de la modélisation avec UML. Après la modélisation des besoins puis l'organisation de la structure de la solution, la conception détaillée vient construire et documenter précisément les classes, les interfaces, les tables et les méthodes qui constituent le codage de la solution.

6.1. Dictionnaire des données

6.1.1. Les classes et les attributs

Classe	Description	Code	Type	Taille
Administrateur	Numéro de fonctionnaire	NUM_Fct	int	10
	Nom de fonctionnaire	NOM	varchar	50
	Prénom de fonctionnaire	PRENOM	varchar	50
	Date de naissance de fonctionnaire	DATE_N	Date	10
	Adresse de fonctionnaire	ADRESSE	varchar	200
	Téléphone de fonctionnaire	TEL	varchar	10
	La spécialité	QUALITE	varchar	50
	Affectation	AFFECTATION	varchar	50
	Utilisateur de compte	USER	varchar	20
	Mot de passe de compte utilisateur	MOT_PASSE	varchar	10
Patient	Numéro de patient	NUM_p	int	11
	Nom de patient	NOM	varchar	50
	Prénom de patient	PRENOM	varchar	50
	Age de patient	AGE	varchar	3
	Date de naissance de patient	DATE_N	date	10
	Sexe de patient	SEX	varchar	10
	Adresse de patient	ADRESSE	varchar	100
	Commune de résidence de patient	COMMUNE	varchar	50
	Date d'entrée	DATE_E	date	10
	Service affecté	SERVICE	varchar	50
	Numéro de lit affecté	NUM_LIT	varchar	20
	Date de sortie	DATE_S	date	10
	Mode de sortie	MODE_S	varchar	20
	Analyse	PCR	varchar	10
	Analyse	SEROLOGIE	varchar	10
Analyse	TDM	int	11	

Chapitre 2 : La méthodologie de développement 2TUP

	Analyse	OXYGENE	int	11
Produit	Le code de produit	CODE_PRODUIT	int	11
	Le nom de produit	NOM_PRODUIT	varchar	50
	Dosage de produit	DOSAGE	varchar	50
	Quantité de produit	QUANTITE_BTS	int	11
	Quantité de produit dans le stock	QUANTITE_EN_STOCK	int	11
Personnel	Numéro de fonctionnaire	NUM_Fct	int	10
	Nom de fonctionnaire	NOM	varchar	50
	Prénom de fonctionnaire	PRENOM	varchar	50
	Date de naissance de fonctionnaire	DATE_N	date	10
	Adresse de fonctionnaire	ADRESSE	varchar	200
	Numéro de téléphone	TEL	varchar	10
	Le poste occupé	QUALITE	varchar	50
	Spécialité	TYPE	varchar	50
	Service	AFFECTATION	varchar	50
	Le nom de compte de l'utilisateur	USER	varchar	20
	Mot de passe	MOT_PASSE	varchar	10
Planning	Id de planning	ID_PLANINGE	int	11
	Identifier le jour	JOURNEE	varchar	10
Bilan	Id de bilan	ID_BILAN	int	11
	La date de bilan	DATE_BILAN	date	10
	Oxygène	OXYGENE	int	11
	Tension	TENSION	varchar	10
	Température	TEMPERATURE	int	11
	Analyse	D_DIMARE	int	11
	Analyse	CRP	int	11
	Analyse	PCR	varchar	20
	Analyse	SEROLOGIE	varchar	20
	Analyse	TDM	int	11
Commande	Id de commande	ID_COMMANDE	int	(11)
	La date d'envoi de commande	DATE_COMMANDE	varchar	(10)
	Id de produit	ID_PRODUIT	int	11
	Nom de produit	PRODUIT	varchar	50
	Dosage	DOSAGE	varchar	20
	Quantité	QUANTITE	int	11
	Quantité livré	QUANTITE_LIVRER	int	11
	Commande envoyée	SI_ENVOYER	int	11

	Commande réceptionnée	SI_RECEPT	int	11
	Numéro de lit	NUM_LIT	varchar	5
	Le type	TYPE_LIT	varchar	20
	Affecter	SI_AFFECTER	int	11
Prescription	Numéro de Prescription	NUM_PRESC	int	11
	La date de soins	DATE_SOIN	date	10
	Numéro de patient	NUM_p	int	11
	Numéro de médecin	NUM_Fct	int	10
	Code de produit	CODE_PRODUI	int	11
	Fréquence	FREQUENCE	varchar	20
	Quantité	QUNTITE	int	11
	Prescription envoyée	SI_ENVOYER	int	11
Stock	Id mouvement	ID_MOUVEMENT	int	11
	Date de mouvement	DATE_MOUVEMENT	date	10
	Id de produit	ID_PRODUI	int	11
	Quantité	QUNTITE	int	11
	Type de mouvement	TYPE_MOUVEMENT	varchar	10
Affectation_p ers	Id de l'affectation	ID_aff	int	11
	La journée affectée	JOURNEE	varchar	10

Tableau 2.20 : Tableau des classes et des attribues.

6.1.2. Les opérations

Classe	Opération	Description
Fonctionnaire (compte)	Créer	Ajouter un nouveau compte.
	Modifier	Modifier les informations d'un compte.
	Supprimer	Supprimer les informations d'un compte.
Patient	Créer	Ajouter un nouveau patient.
	Modifier	Modifier les informations d'un patient.
	Supprimer	Supprimer les informations d'un patient.
	Consulter	Consulter dossier patient.
	Chercher	Chercher dossier patient.
Admission	Enregistrer	Enregistrer les informations d'un patient.
Planning	Créer	Ajouter un nouveau planning.
	Modifier	Modifier le planning.
	Supprimer	Supprimer le planning.
	Chercher	Chercher le planning d'un fonctionnaire.
Produit	Ajouter	Ajouter un nouveau produit.

	Modifier	Modifier la quantité de produit.
	Supprimer	Supprimer le produit.
	Chercher	Chercher l'état de la liste des produits.
	Envoyer	Livrer produit au service.
Soins	Ajouter	Ajouter un nouveau soin.
	Modifier	Modifier le soin.
	Supprimer	Modifier le soin.
Lit	Ajouter	Ajouter un nouveau soin.
	Modifier	Modifier le type de lit.
	Supprimer	Modifier le type de lit.
Bilan	Créer	Ajouter un nouveau bilan.
	Modifier	Modifier le bilan.
	Supprimer	Supprimer le bilan.
	Chercher	Chercher le bilan d'un patient.
/	Annuler	Annuler l'opération

Tableau 2.21: Tableau des opérations.

6.3. Passage du diagramme de classe au modèle relationnel

6.3.1. Règles de passage

Les règles de transformation des associations dépendent des multiplicités maximales des associations. Nous distinguons trois familles d'associations :

- **Règle 1** : Toute classe devient une relation ayant pour clé primaire son identifiant. Chaque propriété se transforme en attribut.
- **Règle 2 Association 1..*** Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. On peut se rappeler cette règle de la manière suivant : la clé de la relation père migre dans la relation fils.
- **Règle 3 : Association *.* et classes-associations** L'association/classe association devient une relation. La clé primaire de cette relation est la concaténation des clés primaires des classes connectées à l'association. Chaque attribut devient clé étrangère. Les attributs de l'association /classe-association doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.
- **Règle 4 : 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.
- **Règle 5** : En cas de héritage, transformer chaque sous-classe en une relation, la clé primaire de la super classe devient clé primaire de chaque sous classe.

6.3.2. 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 de données, Les tables générées sont :

Classes	Attributs
Administrateur	(# <u>NUM_pers</u> , NOM, PRENOM, DATE_N, ADRESSE, TEL, QUALITE, AFFECTATION, USER, MOT_PASSE).
Patient	(# <u>NUM_p</u> , NOM, PRENOM, AGE, DATE_N, SEX, ADRESSE, COMMUNE, DATE_E, SERVICE, + <u>NUM_LIT</u> , DATE_S, MODE_S, PCR, SEROLOGIE, TDM, OXYGENE, M_CHRONIQUE).
Personnels	(# <u>NUM_pers</u> , NOM, PRENOM, DATE_N, ADRESSE, TEL, QUALITE, TYPE, AFFECTATION, USER, MOT_PASSE).
Produit	(# <u>CODE_PRODUIT</u> , NOM_PRODUIT, DOSAGE, QUANTITE_BTS, QUANTITE_EN_STOCK).
Planning	(# <u>ID_PLANNING</u> , JOURNEE, + <u>NUM_pers</u>).
Prescription	(# <u>NUM_Presc</u> , DATE_SOIN, + <u>NUM_p</u> , + <u>NUM_pers</u> , + <u>CODE_PRODUIT</u> , FREQUENCE, QUNTITE, SI_ENVOYER).

Commande	(# ID COMMANDE , DATE_COMMANDE, + ID_PRODUIT , PRODUIT , DOSAGE, QUANTITE, QUANTITE_LIVRER, SI_ENVOYER, SI_RECEPT).
Bilan	(# D_BILAN , DATE_BILAN, + ID_PATIENT , NOM_PATIENT , PRENOM_PATIENT , AGE, + ID_INFIRMIER , NOM_INFIRMIER , PRENOM_INFIRMIER , OXYGENE, TENSION, TEMPERATURE, D_DIMARE, CRP, PCR, SEROLOGIE, TDM).
Lit	(# NUM_LIT , TYPE_LIT, SI_AFFECTER).
Stock	(# ID MOUVEMENT , DATE_MOUVEMENT, + CODE_PRODUIT , QUANTITE, TYPE_MOUVEMENT).
Affectation_pers	(# ID aff , JOURNEE, + NUM p , + NUM pers).

Tableau 2.22: Table de la base de données.

Conclusion

Dans ce chapitre on a présenté dans une première partie la méthode de développement 2TUP qu'on a suivi pour la conception et la réalisation de notre système, ensuite on a détaillé les résultats d'application de ses phases.

Chapitre 3

*Optimisation combinatoire
et méthode de résolution*

Introduction

L'optimisation combinatoire est un outil indispensable combinant diverses techniques de la mathématique discrète et de l'informatique afin de résoudre des problèmes d'optimisation combinatoire de la vie réelle. Un problème d'optimisation combinatoire consiste à trouver la meilleure solution dans un ensemble discret de solutions appelé ensemble des solutions réalisables. En général, cet ensemble est fini mais de cardinalité très grande [12]. Il s'agit, en général, de maximiser (problème de maximisation) ou de minimiser (problème de minimisation) une fonction objective sous certaines contraintes. Le but est de trouver la solution optimale dans un temps d'exécution raisonnable. Dans ce chapitre nous présentons certaines méthodes parmi les plus représentatives, développées pour résoudre les problèmes d'optimisation combinatoire.

I. Généralité sur l'optimisation

1. Problèmes d'optimisation combinatoire

L'optimisation combinatoire consiste à minimiser ou maximiser une fonction souvent appelée fonction coût, d'une ou plusieurs variables soumises à des contraintes. L'optimisation combinatoire occupe une place très importante en recherche opérationnelle, en mathématiques discrètes et en informatique. Son importance se justifie d'une part par la grande difficulté des problèmes d'optimisation et d'autre part par de nombreuses applications pratiques pouvant être formulées sous la forme d'un problème d'optimisation combinatoire. Bien que les problèmes d'optimisation combinatoire soient souvent faciles à définir, ils sont généralement difficiles à résoudre. En effet, la plupart de ces problèmes appartiennent à la classe des problèmes NP-difficiles et ne possèdent donc pas à ce jour de solution algorithmique efficace valable pour toutes les données [13].

Optimisation = modélisation + résolution

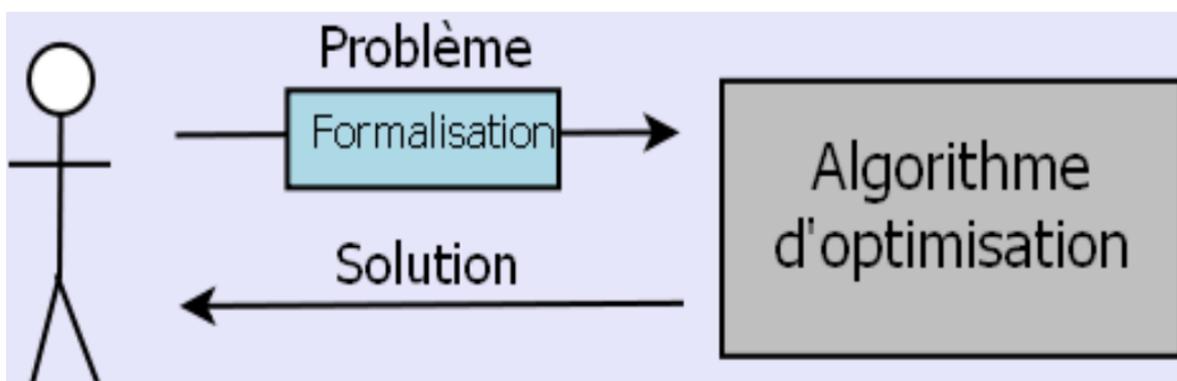


Figure 3.1 : L'optimisation combinatoire.

2. Problèmes célèbres en optimisation combinatoire

L'optimisation combinatoire englobe une classe importante des problèmes de la recherche opérationnelle. Il nous est impossible de les énumérer de manière exhaustive mais nous en citons les plus fréquents.

2.1.Problème d'affectation

L'objectif d'un problème d'affectation est d'affecter un ensemble d'éléments (objets, tâches, endroits, . . .) à un ensemble d'individus (machines, endroits, . . .) de sorte que le produit du flot circulant entre ces différents endroits et la distance les séparant soit minimisée. Par exemple, la répartition de bâtiments dans un espace donné en fonction du nombre de personnes amenées à circuler entre ces bâtiments, ou la répartition des modules électroniques sur une carte en fonction du nombre de connexions les liant les uns aux autres.

2.2.Problème d'ordonnancement des tâches

Il consiste à définir dans quel ordre, exécuter un ensemble de tâches T sur un ensemble de machines M, de sorte qu'une machine exécute une et une seule tâche et que l'ensemble des tâches soit exécuté en un minimum de temps.

2.3.Problème du Sac à Dos

L'énoncé de ce problème fameux en optimisation combinatoire est simple : étant donné un ensemble d'objets chacun possédant un poids et une valeur et étant donné une capacité pour le sac, quels objets doit on mettre dans le sac de manière à maximiser la valeur totale sans dépasser la capacité du sac.

2.4.Problème de Bin Packing

Il s'agit de trouver le nombre minimum de boîtes (bins) de capacité fixe pour emballer un ensemble d'éléments de tailles différentes sans que la capacité des boîtes soit excédée.

2.5.Problème d'Emploi du Temps

Ce problème a plusieurs variantes, selon le domaine d'application, leur point commun est d'élaborer la planification de l'activité de l'individu au cours du temps. La solution est affichée sous forme d'un tableau à deux dimensions dont lequel on associe les lignes aux personnes et les colonnes aux périodes horaires.

2.6. Problèmes liés à la théorie des graphes

Nous citons les problèmes de couplage, de couverture de sommets, de coloration de graphes, de stable maximum, le problème de voyageur de commerce (PVC) et le problème de tournées de véhicules (VRP), qui a pour objectif le service d'un ensemble des clients en parcourant une longueur minimale.

3. Résolution d'un problème d'optimisation combinatoire

Résoudre un problème d'optimisation combinatoire nécessite l'étude de trois points particuliers :

- la définition de l'ensemble des solutions réalisables.
- l'expression de l'objectif à optimiser.
- le choix de la méthode d'optimisation à utiliser.

Les deux premiers points relèvent de la modélisation du problème, le troisième de sa résolution. Afin de définir l'ensemble des solutions réalisables, il est nécessaire d'exprimer l'ensemble des contraintes du problème. Ceci ne peut être fait qu'avec une bonne connaissance du problème sous étude et de son domaine d'application.

4. Les méthodes de résolution des problèmes d'optimisation combinatoire

Les méthodes d'optimisation peuvent être réparties en deux grandes classes de méthodes pour la résolution des problèmes :

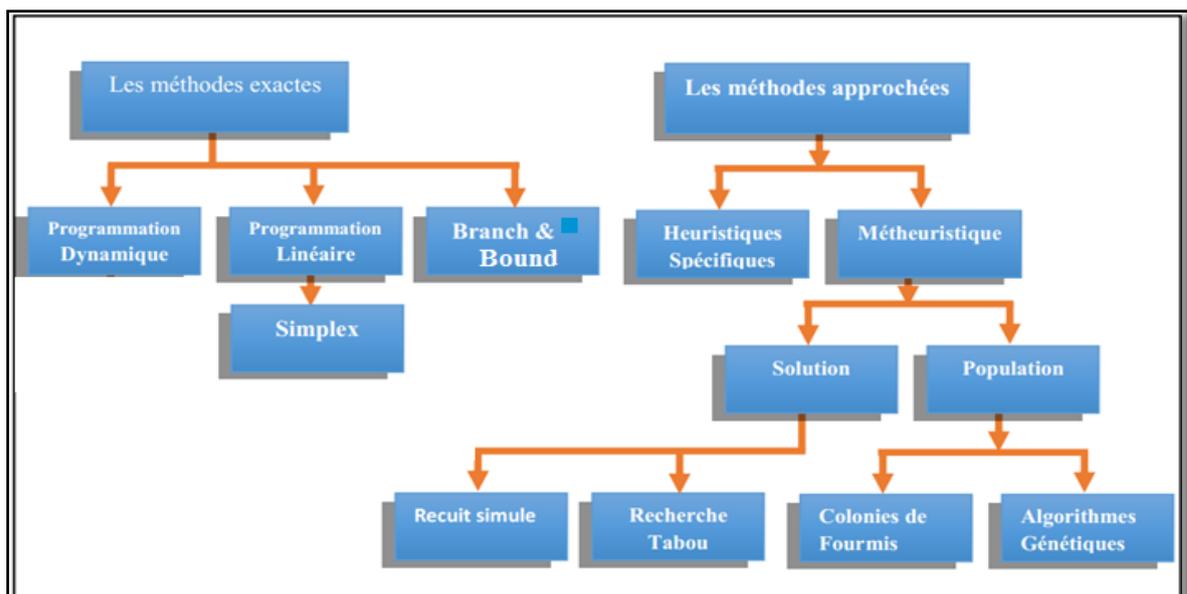


Figure 3.2 : Classification des méthodes d'optimisation combinatoire.

4.1. Les méthodes exactes

Une méthode de résolution est dite exacte si elle garantit l'obtention d'une solution optimale du problème. En effet, le temps de résolution de certains problèmes par les méthodes exactes connues augmente d'une manière exponentielle avec leurs tailles. Nous pouvons distinguer trois grands types de méthodes exactes :

4.1.1. La programmation dynamique

En informatique, la programmation dynamique est une méthode algorithmique pour résoudre des problèmes d'optimisation. Le concept a été introduit au début des années 1950 par Richard Bellman. À l'époque, le terme « programmation » signifie planification et ordonnancement. La programmation dynamique consiste à résoudre un problème en le décomposant en sous-problèmes, puis à résoudre les sous-problèmes, des plus petits aux plus grands en stockant les résultats intermédiaires. Elle a d'emblée connu un grand succès, car de nombreuses fonctions économiques de l'industrie étaient de ce type. [13]

4.1.2. La méthode de simplexe

L'algorithme du simplexe est un algorithme de résolution des problèmes d'optimisation linéaire. Il a été introduit par George Dantzig à partir de 1947. C'est probablement le premier algorithme permettant de minimiser une fonction sur un ensemble défini par des inégalités. De ce fait, il a beaucoup contribué au démarrage de l'optimisation numérique. L'algorithme du simplexe a longtemps été la méthode la plus utilisée pour résoudre les problèmes d'optimisation linéaire. Depuis les années 1985 à 1990, il est concurrencé par les méthodes de points intérieurs, mais garde une place de choix dans certaines circonstances (en particulier si l'on a une idée des contraintes d'inégalité active en la solution). [13]

4.1.3. Branch and Bound

La technique du Branch & Bound est une méthode algorithmique classique pour résoudre un problème d'optimisation combinatoire. Il s'agit de chercher une solution optimale dans un ensemble combinatoire de solutions possibles. La méthode repose d'abord sur la séparation (Branch) de l'ensemble des solutions en sous-ensembles plus petits. L'exploration de ces solutions utilise ensuite une évaluation optimiste pour majorer (bound) les sous-ensembles, ce qui permet de ne plus considérer que ceux susceptibles de contenir la solution potentiellement meilleure que la solution courante. [15]

4.2. Les méthodes approchées

4.2.1. Heuristiques

Les heuristiques constituent une alternative très intéressante pour traiter les problèmes d'optimisation de grande taille en un temps de calcul très petit, une heuristique représente un bon compromis entre le temps de recherche et la qualité de solution. Feigenbaum et Feldman (1963) définissent une heuristique comme une règle d'estimation, une stratégie, une astuce, une simplification, ou toute autre sorte de système qui limite drastiquement la recherche des solutions dans l'espace des configurations possibles. [13]

4.2.2. Méta-Heuristiques

Une méta-heuristique est un algorithme d'optimisation visant à résoudre des problèmes d'optimisation difficiles (souvent issus des domaines de la recherche opérationnelle, de l'ingénierie ou de l'intelligence artificielle) pour lesquels on ne connaît pas de méthodes classiques plus efficace.

Les méta-heuristiques sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global, c'est-à-dire l'extremum global d'une fonction, par échantillonnage d'une fonction objectif. Elles se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation).

- ✓ Les algorithmes évolutionnistes, parmi lesquels :
 - L'algorithme génétique.
 - le recuit simulé.
 - L'algorithme de colonies de fourmis.
 - la recherche Tabou.

La recherche dans le domaine étant très active, il est impossible de produire une liste exhaustive des différentes méta-heuristiques d'optimisation. La littérature spécialisée montre un grand nombre de variantes et d'hybridations entre méthodes, particulièrement dans le cas des algorithmes évolutionnaires.

4.2.2.1. La recherche Tabou

La recherche Tabou (Tabou Search) est une méthode de recherche locale combinée avec un ensemble de techniques permettant d'éviter d'être piégé dans un minimum local ou la répétition d'un cycle. La recherche Tabou est introduite principalement par Glover &

Hansen 1986. Cette méthode a montré une grande efficacité pour la résolution des problèmes d'optimisation difficiles [14].

4.2.2.2. Les colonies de fourmis

Cette méta-heuristique a été proposée pour la première fois par Dorigo en 1992. Les colonies de fourmis sont basées sur le comportement réel et la communication chez les fourmis. Il est connu que les fourmis sont capables de déterminer le chemin le plus court entre leur nid et une source de nourriture grâce à la phéromone déposés sur les trajets parcourus. Cela peut paraître surprenant au premier abord mais un chemin plus court reçoit plus de phéromones qu'un chemin plus long. Cet algorithme a été appliquée sur le problème du voyageur de commerce (TSP : Travelling Salesman problem). [16]

4.2.2.3. Les algorithmes génétiques

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes. Leur but est d'obtenir une solution approchée à un problème d'optimisation, lorsqu'il n'existe pas des méthodes exactes (ou que la solution est inconnue) pour le résoudre en un temps raisonnable. Les algorithmes génétiques utilisent la notion de sélection naturelle et l'appliquent à une population de solutions potentielles au problème donné.

4.2.2.4. Le recuit simulé

C'est une méthode basée sur le principe de refroidissement du métal. On a choisi Cette méthode pour la résolution de notre problème d'affectation des médecins, on va la détaillée dans la partie suivante.

II. La méthode de résolution

1. Choix de la méthode

On a choisi la méthode de recuit simulé par ce qu'elle est généralement connue par deux caractéristiques : la facilité de son implémentation, et sa rapidité d'exécution. En outre, elle fournit des bonnes résultats pour des nombreux problèmes d'optimisation combinatoire, on commence avec une seule solution initiale et s'en éloignant progressivement, en construisant une trajectoire dans l'espace de recherche.

2. Recuit simulé (Simulated annealing)

En algorithmique, le recuit simulé est une méthode empirique (méta-heuristique) d'optimisation, inspirée d'un processus utilisé en métallurgie. On alterne dans cette dernière

des cycles de refroidissement lent et de réchauffage (recuit) qui ont pour effet de minimiser l'énergie du matériau. Cette méthode a transmis à l'optimisation pour trouver les extrema d'une fonction [17].

Cette méthode vient de l'observation que le refroidissement naturel de certains métaux ne permet pas aux atomes de se mettre dans la composition la plus solide. La configuration la plus stable est obtenue en contrôlant le refroidissement et en le ralentissant par une entrée thermique externe ou par isolation.

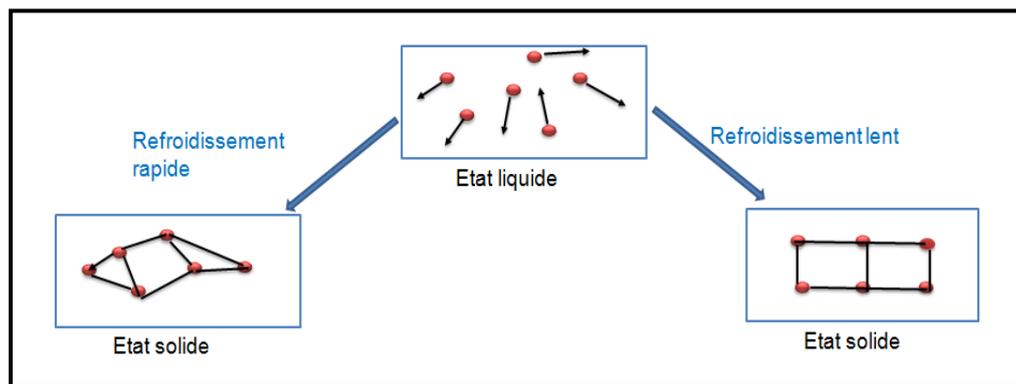


Figure 3.3 : Description recuit simulé.

2.1. Historique

La méthode de recuit simulé réalisées par Metropolis et al. (1953) pour simuler l'évolution de ce processus de recuit physique. Elle a été mise au point par trois chercheurs de la société IBM, Kirkpatrick, C.D. Gelatt et M.P. Vecchi en 1983 au Etats-Unis, et indépendamment par V. Černý en 1985 en Slovaquie, l'utilisation de cette méthode pour la résolution des problèmes d'optimisation combinatoire est beaucoup plus récente.

2.2.Principe

- Explorer l'espace d'état de manière aléatoire afin d'éviter les minimums (ou maximums) locaux.
- Diminuer progressivement la température $T(n)$ pour stabiliser l'algorithme sur un minimum (ou maximums) global.
 - Si le refroidissement est trop rapide, il y a un risque de rester bloqué dans un minimum local (configuration sous-optimale).

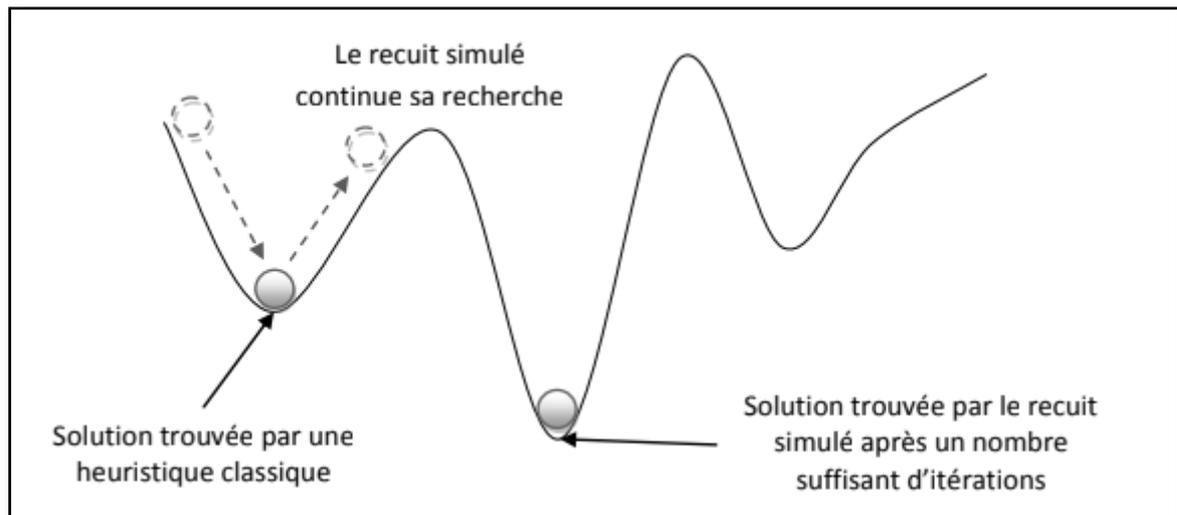


Figure 3.4 : Comparaison entre le recuit simulé et une heuristique classique.

3. Les étapes de l'algorithme

3.1.L'algorithme de Metropolis

Dans l'algorithme de Metropolis, on part d'une configuration donnée, et on lui fait subir une modification aléatoire. Si cette modification fait diminuer la fonction objectif (ou énergie du système), elle est directement acceptée ; Sinon, elle n'est pas acceptée qu'avec une probabilité égale à $\text{Exp}(\Delta E/T)$, cette règle est appelé critère de Metropolis [18].

3.2.L'algorithme du recuit simulé

Le recuit simulé applique itérativement l'algorithme de Metropolis, pour engendrer une séquence de configurations qui tendent vers l'équilibre thermodynamique :

1. Choisir une température de départ T et une solution initiale $s = s_0$.
2. Générer une solution aléatoire dans le voisinage de la solution actuelle. $s_0 \rightarrow s'$, $s' \in V(s)$.
3. On calcule la variation de coût $\Delta f = f(s') - f(s_0)$
4. Si $\Delta f \leq 0$, le coût diminue et on effectue la transformation améliorante $s_0 := s'$.
5. Si $\Delta f > 0$, le coût remonte. On calcule une probabilité d'acceptation $a = e^{-\Delta f/T}$, puis on tire au sort p dans $[0,1]$. Si $p \leq a$, la transformation est déclarée acceptée, bien qu'elle dégrade le coût, et on fait $s_0 := s'$. Sinon, la transformation est rejetée: on garde s_0 pour l'itération suivante.
6. répéter 2 et 3 jusqu'a ce que l'équilibre statistique soit atteint ;
7. décroître la température et répéter jusqu'a ce que le système soit gelé.

3.2.1. L'état initial de l'algorithme

La solution initiale peut être prise au hasard dans l'espace des solutions possibles. A cette solution correspond une énergie initiale $E=E_0$. Cette énergie est calculée en fonction du critère que l'on cherche à optimiser. Une température initiale $T=T_0$ élevée est également choisie.

3.2.2. Paramètre de la température

- Le paramètre T (température) est un réel positif.
- La température permet de contrôler l'acceptation des dégradations :
 - ✓ Si T est grand, les dégradations sont acceptées avec une probabilité plus grande.
 - ✓ A la limite, quand T tend vers l'infini, tout voisin est systématiquement accepté.
 - ✓ Inversement, pour $T=0$, une dégradation n'est jamais acceptée.
- La température varie au cours de la recherche : T est élevée au début, puis diminue et finit par tendre vers 0.

4. Avantages et inconvénients de la méthode

Avantages

- Facile à implémenter.
- Donne généralement de bonnes solutions par rapport aux algorithmes de recherche classiques.
- Peut être utilisé dans la plupart des problèmes d'optimisation.

Inconvénients

- La difficulté de déterminer la température initiale.
- L'impossibilité de savoir si la solution trouvée est optimale.
- Dégradation des performances pour les problèmes où il y a peu de minimums locaux.

5. Affectation des médecins aux patients

5.1. Présentation du problème

Notre travail traite la problématique des affectations des patients aux médecins disponibles dans le service COVID-19 dans le but d'améliorer la qualité du service et assurer une meilleure hospitalisation aux patients surtout pendant la période où le nombre des cas augmente et le nombre des médecins est limité, cela crée un déséquilibre dans la répartition des médecins.

5.2. Les données

LP: liste des patients.

LM: liste des médecins.

5.3. Les contraintes

- Garantir que chaque patient a traité par un médecin pendant la journée.
- Répartir équitablement les patients entre les médecins.
- Assurer le traitement des patients qui ont des maladies chroniques par des médecins de la même spécialité.

5.4. La fonction objective

A travers notre stage effectué dans le service du COVID, la majorité des patients qui souffrent des maladies chroniques sont parmi les maladies suivantes : Hyper-tension, Asthme, Diabète, les maladies infectieuses.

On a constaté que pendant la crise de la pandémie, le service ne prend pas en considération la spécialité des médecins par rapport aux maladies chroniques ce qui diminue la qualité de soins.

Pour maximiser la valeur de la performance de médecin qui suit les maladies chronique des patients on a défini une matrice d'estimation de degré de la compatibilité des maladies chroniques des médecins traitant le plus proche de son soin.

Maladie/spécialité	Généraliste	Cardiologue	Pneumologue	Inféctiologue	Diabétologue
Néant	10	8	4	2	6
Tension	8	10	2	4	6
Asthmatique	4	8	10	6	2
Infection	6	8	4	10	2
Diabète	6	8	2	4	10

Tableau 3.1 : Tableau approximatif de niveau de compatibilité entre la spécialité de médecin et la maladie.

5.5. Les étapes de notre algorithme de résolution

Pour résoudre ce problème on a décomposé notre approche en deux étapes comme illustré dans la figure suivante.

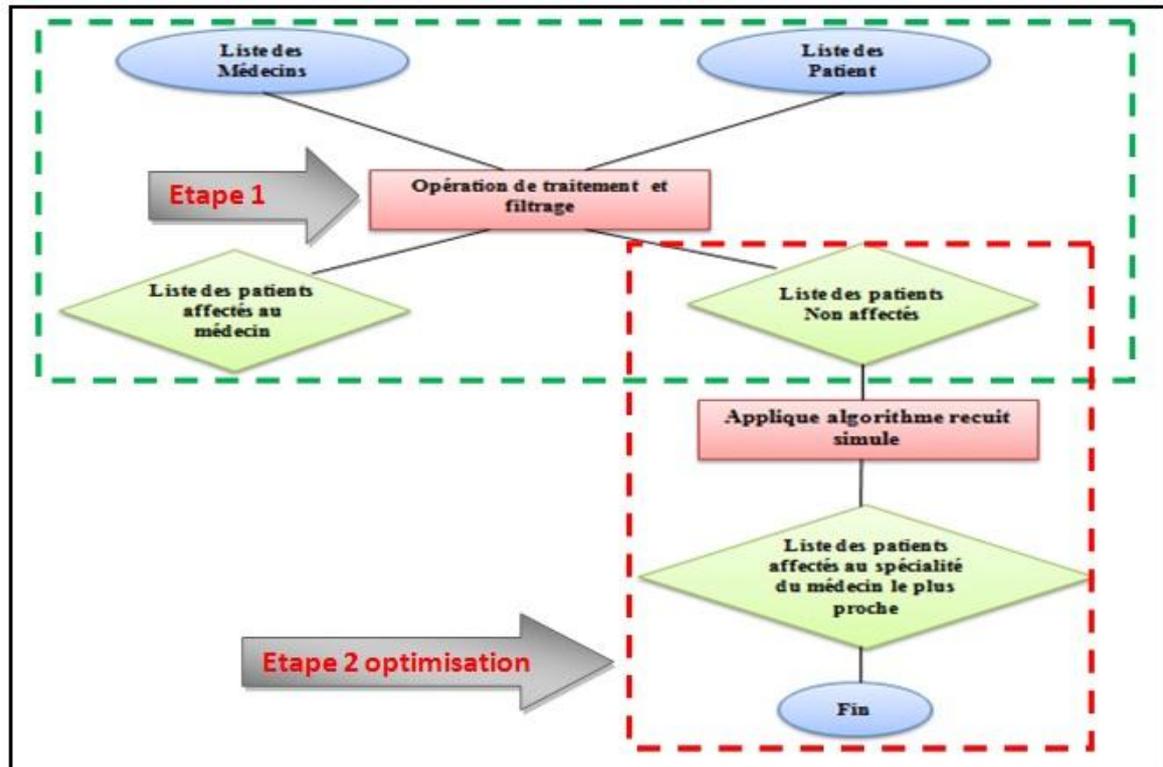


Figure 3.5 : Les étapes pour l'affectation des médecins aux patients.

5.5.1. Etape 1 : Répartition des patients aux médecins dont les maladies chroniques correspond à la spécialité du médecin traitant

- Récupérer la liste des patients hospitalisés (admis).
- Récupérer la liste des médecins présents selon le planning de la journée.
- Partager les patients dont les maladies correspondent à la spécialité des médecins appropriés (selon le Tableau 3.1).
- Prendre en considération la capacité du médecin pendant la journée.

❖ Résultat de l'étape 1

Après le tri, on obtient :

- Une liste contient les patients non affectés.
- Une liste des médecins ayant une charge incomplète (Figure 3.6).

Nous récupérons ces deux listes, puis on fait une répartition selon le médecin dont la spécialité est la plus proche (selon le tableau 3.1) du patient en appliquant **la méthode du recuit simulé**.

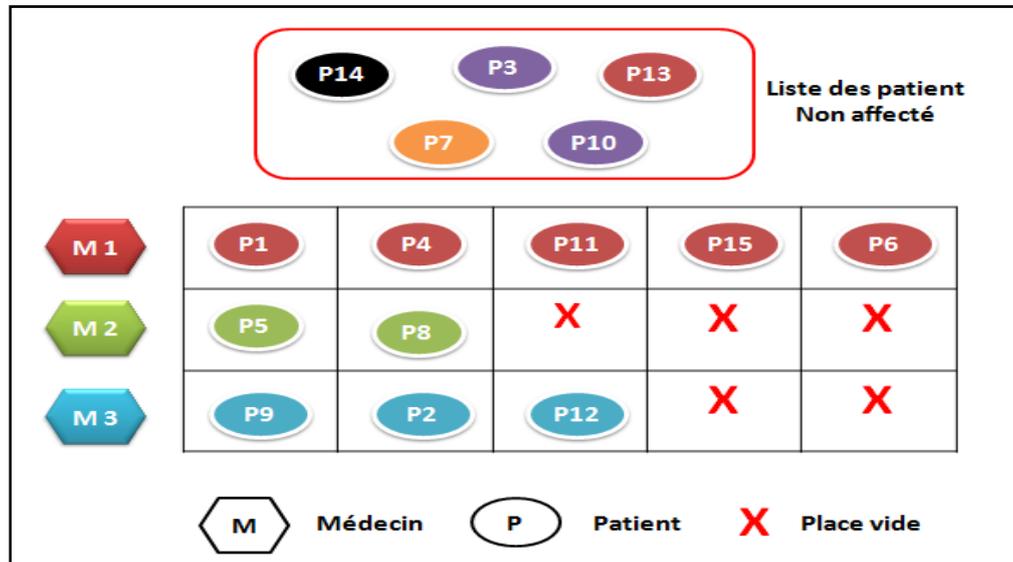


Figure 3.6 : Schéma du déroulement de l'affectation des patients aux médecins.

5.5.2. Etape 2 : Application de la méthode de recuit simulé

5.5.2.1. Principe de fonctionnement de l'algorithme proposé

Nous avons appliqué la méthode de recuit simulé pour la résolution de notre problème ; partant d'une solution initiale S_0 ($S = S_0$), qui est un ensemble de patient choisi aléatoirement à partir de la liste des patients pas encore affectés pour compléter la charge minimale d'un médecin. On considère S_0 comme meilleure solution, ensuite on initialise la température T à une valeur de départ, on génère une solution S' voisine de S et on compare la performance de deux solution. Si la solution S' est plus performante on la sauvegarde comme meilleure solution est on l'utilise pour la prochaine itération sinon on passe au critère de Metropolis (**voir titre critère de Metropolis**). On diminue la température T par une valeur Δ pour chaque itération et on s'arrête dé que T atteint la valeur 0.

5.5.2.2. Construction du voisinage

Le voisinage d'une solution est le remplacement d'un patient par un autre à partir de la liste des patients pas encore affectés, l'algorithme suivant illustre cette tache :

```
Début  
.....  
Données :  
    S = solution de départ ;  
    LP liste des patients pas encore affectés;  
    Choisir au hasard un patient de S et le remplacé par un autre de la liste LP ;  
Fin
```

Figure 3.7 : Pseudo code de l'algorithme de voisinage.

5.5.2.3. La fonction « performance » d'une solution

La qualité d'une solution est mesurée par le niveau d'adaptation des maladies chroniques des patients avec la spécialité du médecin traitant, la fonction suivante illustre cette mesure :

```
Début  
.....  
Données :  
    S = solution;  
    Pour chaque patient de la solution  
        Accumuler les valeurs d'adaptation de la maladie chronique avec la  
        Spécialité de médecin // a partir de tableau 3.1 ;  
    Fin pour  
    Retourner la somme () ;  
Fin
```

Figure 3.8 : Pseudo code de la performance.

5.5.2.4. Le critère de Metropolis

Si la performance du voisin est moins que la performance de la solution en cours, on fait appel au critère de Metropolis; cela peut donner une chance à ce voisin selon une probabilité. On génère une probabilité P aléatoirement et faire comparer $Exp(-\Delta/T)$;

Si la probabilité est inférieure on garde la solution S' pour prochaine itération sinon on élimine cette solution et on génère un autre voisin.

5.5.2.5. Algorithme de Recuit Simulé

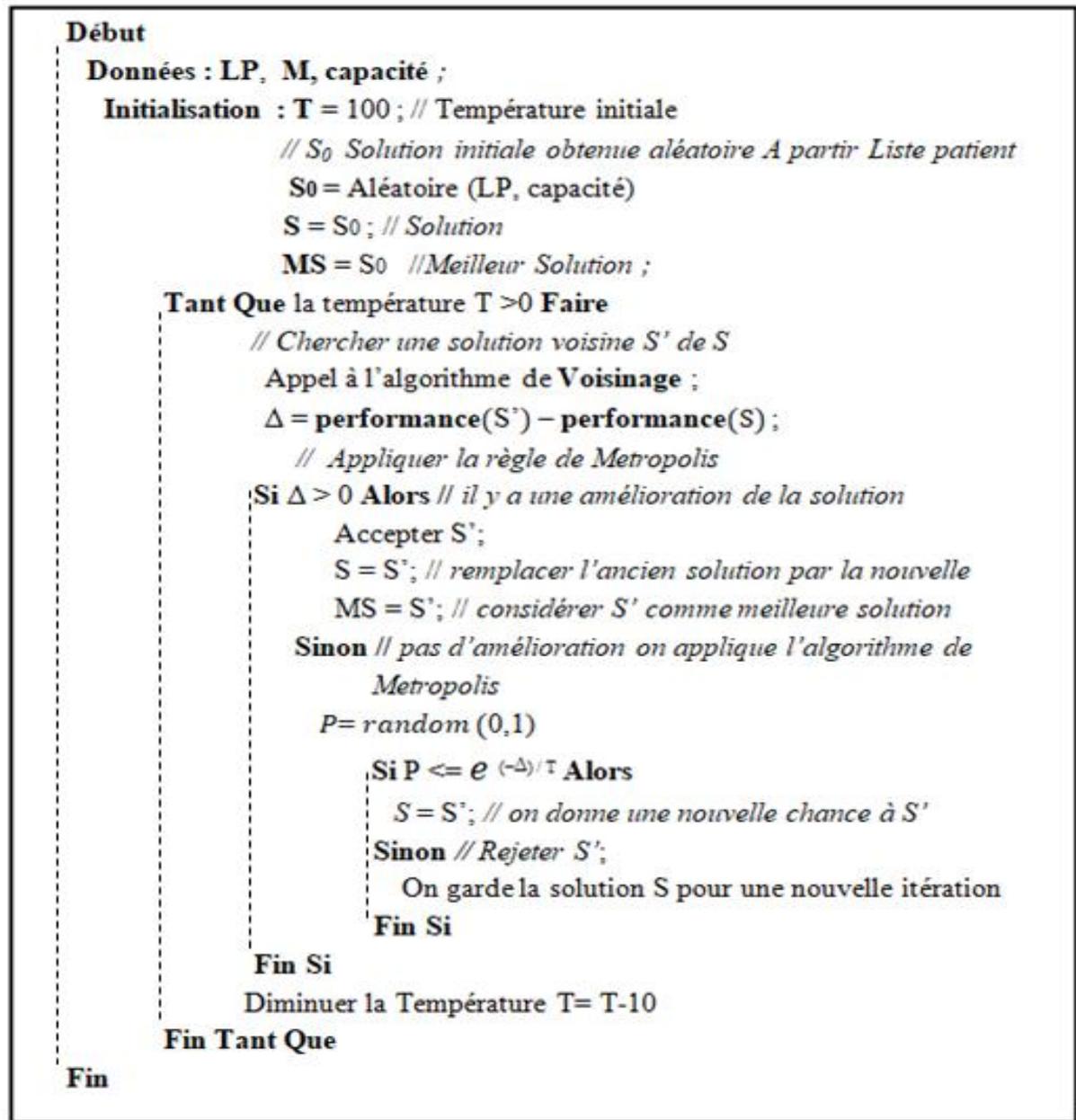


Figure 3.9 : Pseudo code de l'algorithme de recuit simulé.

5.5.2.6. Algorithme d'affectation des patients aux médecins

Le pseudo code suivant présent notre solution globale du problème, il applique la technique de recuit simulé pour chaque médecin pour obtenir la meilleure combinaison :

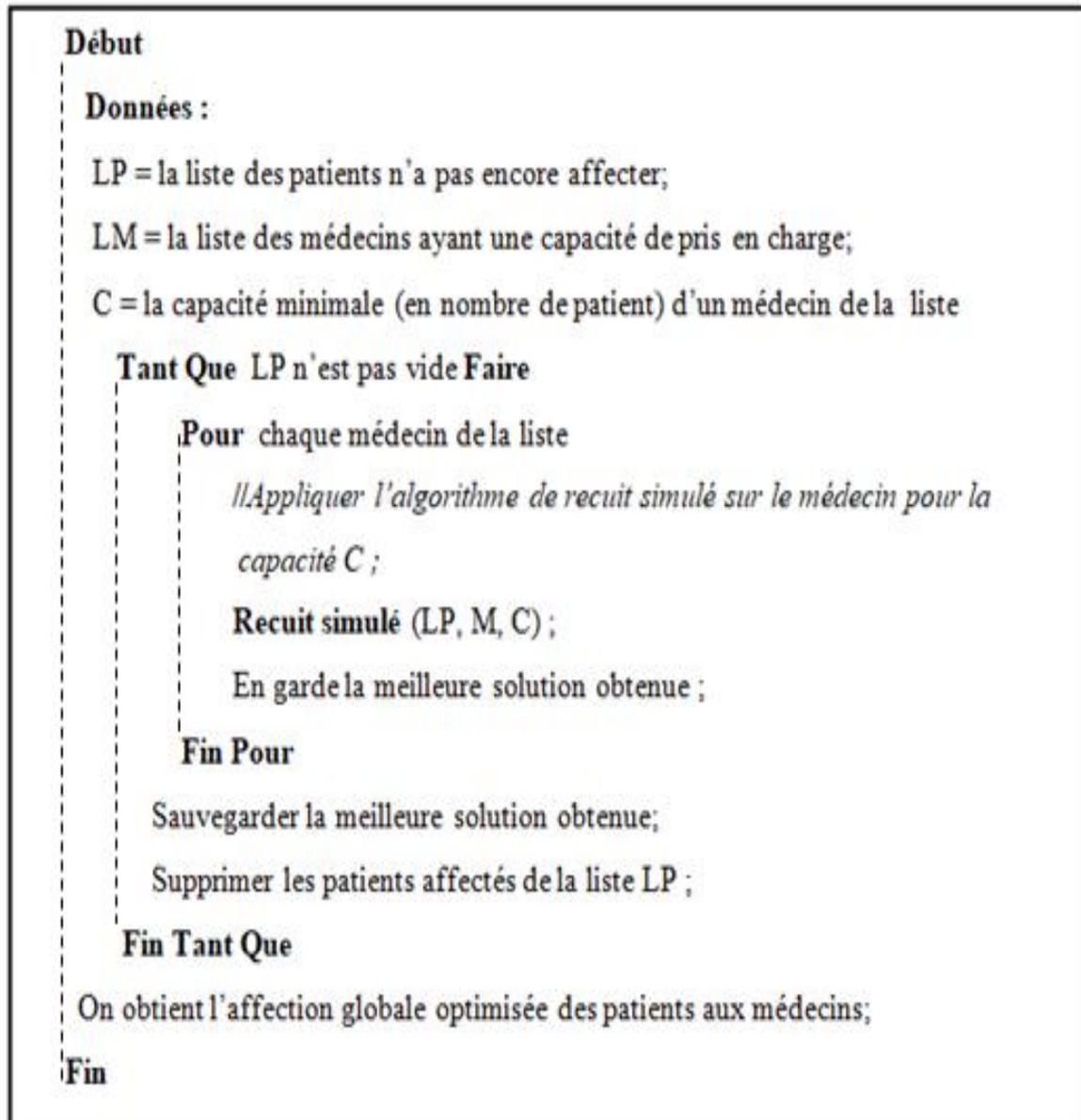


Figure 3.10 : Pseudo code affectation des patients aux médecins.

La figure 3.11 représente un organigramme de l'algorithme d'affectation des patients aux médecins proposés :

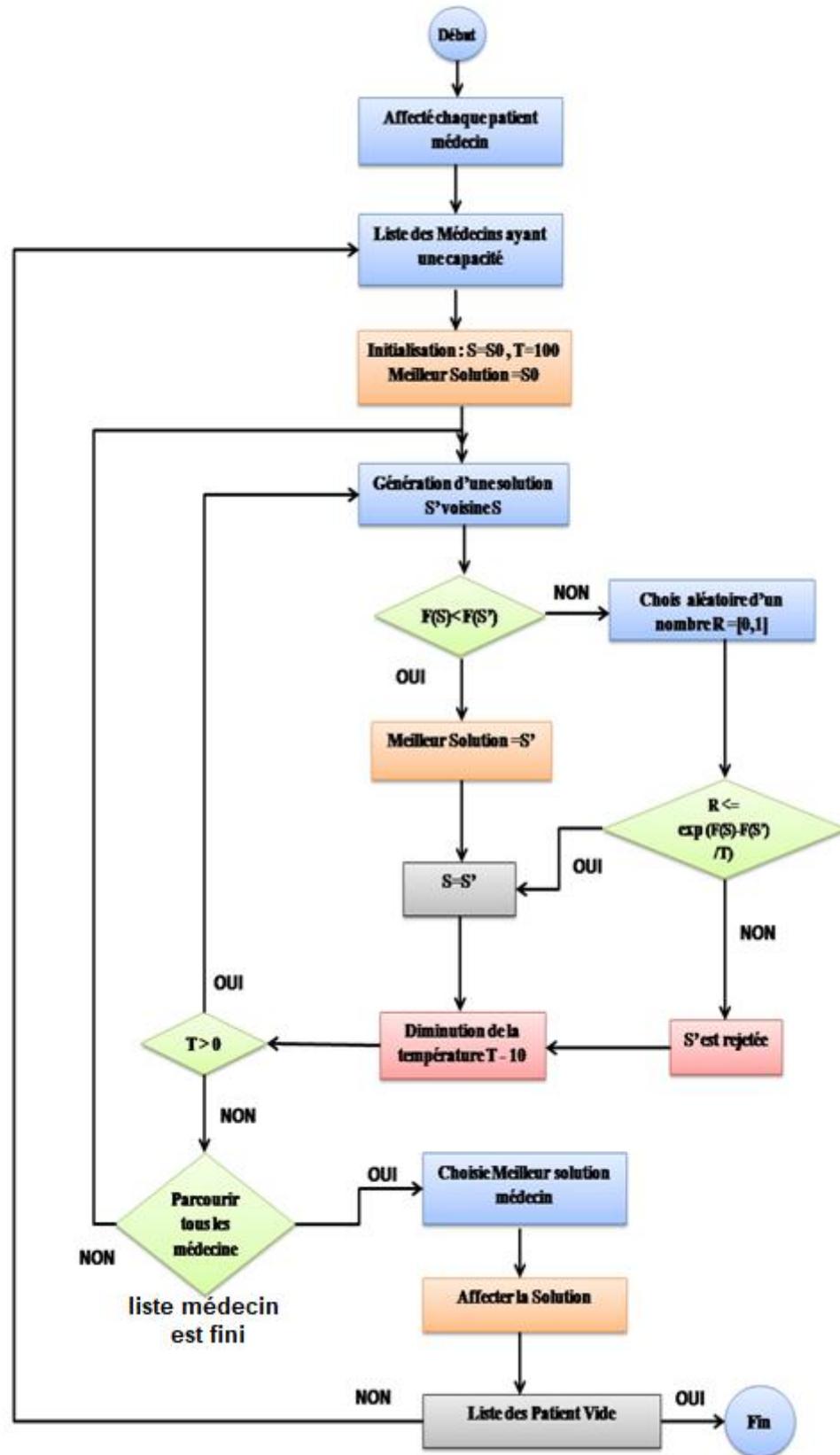


Figure 3.11 : L'organigramme de la solution proposée.

5.5.2.7. Etude expérimentale et résultats

Notre algorithme a été programmé et testé sur Windows avec l'environnement de développement NetBeans, en utilisant le langage Java sur un ordinateur portable équipé d'un processeur Intel(R) dual Core CPU T 3200 @2.400GHz 2.00 GHz.

Dans cette étude, un ensemble d'expériences numériques a été réalisé à partir d'un ensemble de données extraites du système d'information, notamment le nombre de patients à l'hôpital et le nombre de médecins selon leurs spécialités et selon le planning de travail défini par l'administration. Après l'application de l'algorithme on a obtenu les résultats présentés dans la figure suivante :

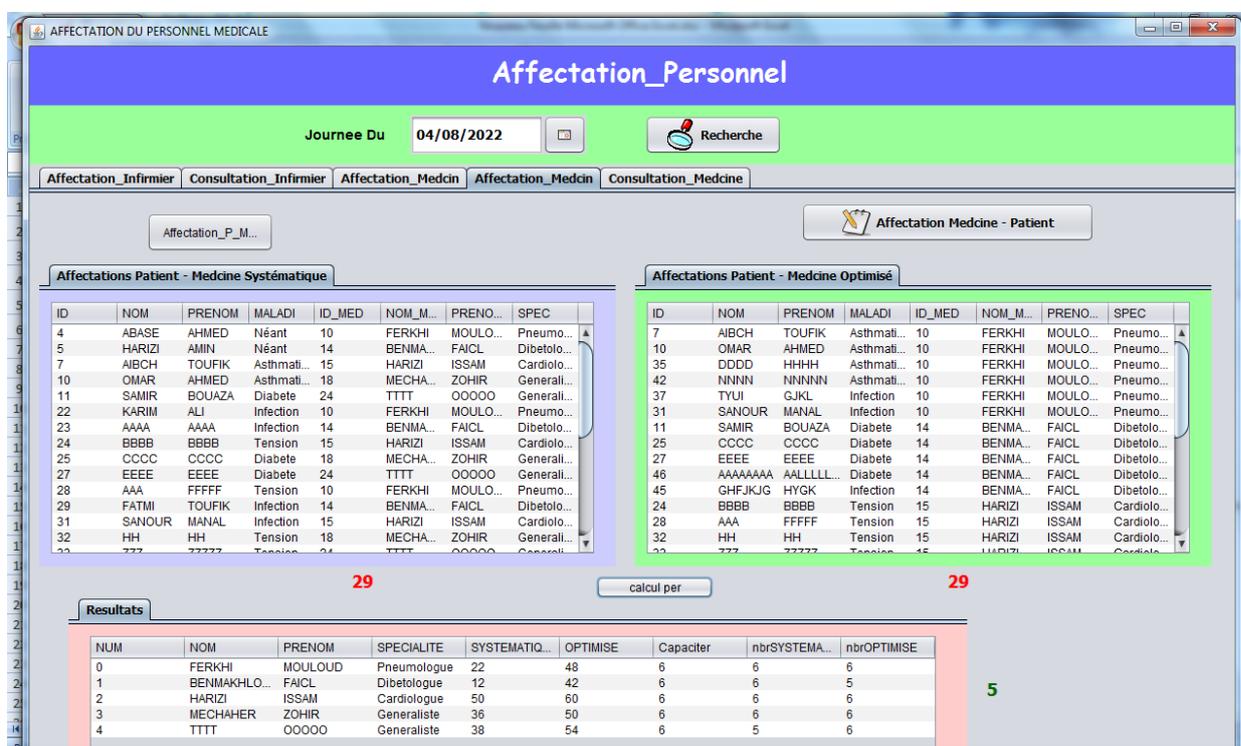


Figure 3.12: Capture d'écran des résultats obtenu.

D'après notre expérience dans l'hôpital dans le cadre de stage on a trouvé que les affectations des patients aux médecins sont faites de façon pseudo aléatoire (c'est -à -dire sans prise en compte de la spécialité des médecins avec les maladies chronique des patients surtout dans la période de pic de la pandémie, pour cela on a généré des solutions aléatoire pour faire les comparer avec les solutions obtenu par application de notre algorithme proposé.

Chapitre 3 : Optimisation combinatoire et méthode de résolution

Pour bien montrer la différence entre les deux solutions on a fait des représentations graphiques. (La figure 3.13) elle montre la performance de la solution de notre système par rapport à la méthode appliqué réellement.

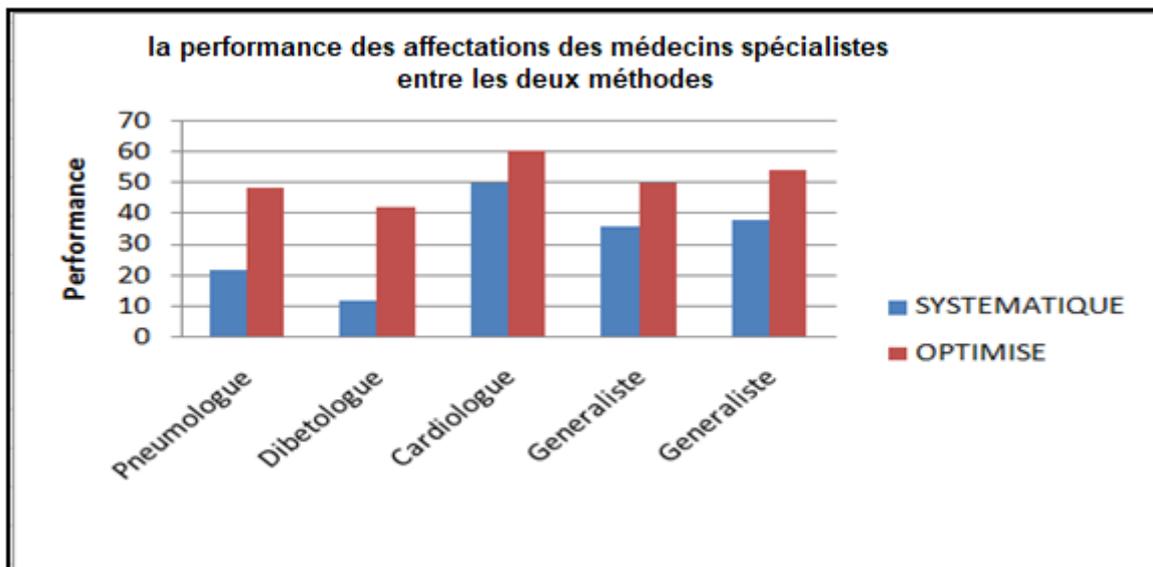


Figure 3.13: La différence entre la performance des affectations des deux méthodes coté médecin.

Si on focalise notre vue coté patient on voit une amélioration de la qualité d'hospitalisation des patients dans la solution de notre algorithme d'optimisation par rapport à la solution appliquée réellement. La figure suivante illustre cette amélioration:

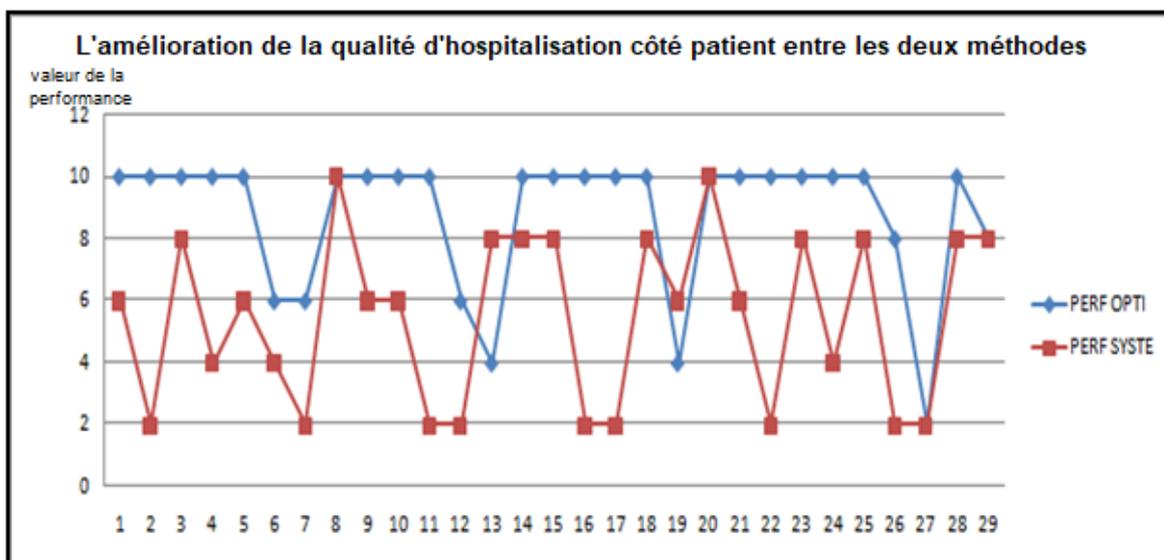


Figure 3.14: Différence entre la performance des affectations des deux méthodes coté patient.

6. Affectation des patients aux infirmiers

La répartition des patients aux infirmiers se fait d'une façon équitable puisque les infirmiers ont le même profil il suffit d'assurer un infirmier pour chaque patient pour garantir le soin.

Pour réaliser cette affectation on a fait juste une divisions de nombre de patient sur le nombre des infirmiers

Voici une capture d'écran des résultats de cette affectation:

The screenshot shows a software window titled 'Affectation_Personnel' with a date field set to '04/08/2022'. It features two main tables: 'Liste des Patient' and 'Liste des Infirmiers'. Below these is a table titled 'Affectations Patient - Infirmiers' which shows the final assignment of nurses to patients.

ID	NOM	PRENOM	M_CHRONIQUE
4	ABASE	AHMED	Néant
5	HARIZI	AMIN	Néant
7	AIBCH	TOUFIK	Asthmatique
10	OMAR	AHMED	Asthmatique
11	SAMIR	BOUAZA	Diabete
22	KARIM	ALI	Infection
23	AAAA	AAAA	Infection
24	BBBB	BBBB	Tension
25	CCCC	CCCC	Diabete
27	EEEE	EEEE	Diabete
28	AAA	FFFF	Tension

JOURNEE	NUM	NOM	PRENOM	QUALITE
04/08/2022	11	BELKACMI	ROUMAISA	INFIRMIER
04/08/2022	16	BOULAKROUN	SADEK	INFIRMIER
04/08/2022	19	BOUSOUF	AMINA	INFIRMIER

ID	NOM	PRENOM	MALADI	ID_INF	NOM_INF	PRENOM_INF
4	ABASE	AHMED	Néant	11	BELKACMI	ROUMAISA
5	HARIZI	AMIN	Néant	16	BOULAKROUN	SADEK
7	AIBCH	TOUFIK	Asthmatique	19	BOUSOUF	AMINA
10	OMAR	AHMED	Asthmatique	11	BELKACMI	ROUMAISA
11	SAMIR	BOUAZA	Diabete	16	BOULAKROUN	SADEK
22	KARIM	ALI	Infection	19	BOUSOUF	AMINA
23	AAAA	AAAA	Infection	11	BELKACMI	ROUMAISA
24	BBBB	BBBB	Tension	16	BOULAKROUN	SADEK
25	CCCC	CCCC	Diabete	19	BOUSOUF	AMINA
27	EEEE	EEEE	Diabete	11	BELKACMI	ROUMAISA
28	AAA	FFFF	Tension	16	BOULAKROUN	SADEK

Figure 3.15 : Affectation des infirmiers aux les patients.

7. Affectation des médicaments aux patients

Au moment de la pandémie, il y a avait une déstabilisation dans la distribution et même la disponibilité de certain médicaments dans les services se qui pose la situation des patients dans un véritable danger pour cela on a proposé un algorithme qui classe les patients demandant d'un médicament selon les analyses correspondant et on donne une dose aux patients les plus prioritaires, le pseudo code suivant résume cette idée :

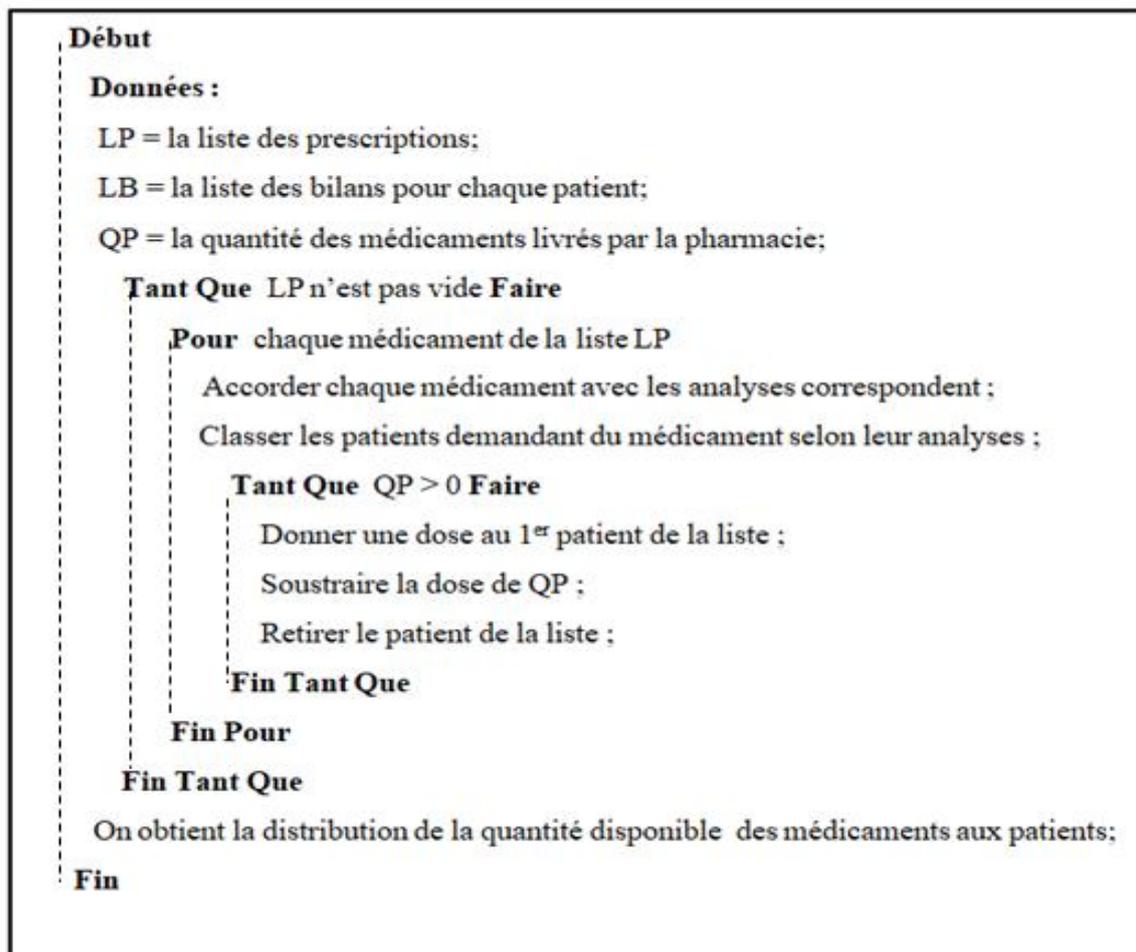


Figure 3.16 : Le pseudo code de la distribution des médicaments aux patients.

Par application de l'algorithme sur des données générées aléatoirement on a obtenu les résultats suivants (Figure 3.17 et Figure 3.18):

Liste des Patient Et Medicaments

DATE_S	ID_M	Medicam...	Qunlité	ID_P	NOM_P	PRENOM...
17/06/20...	2	LOVENO...	3	5	HARIZI	AMIN
17/06/20...	4	PARACIT...	2	5	HARIZI	AMIN
17/06/20...	6	Asitromy...	2	5	HARIZI	AMIN
17/06/20...	7	Hydroxyc...	3	5	HARIZI	AMIN
17/06/20...	8	Dexamét...	1	5	HARIZI	AMIN
17/06/20...	10	OXYGENE	15	5	HARIZI	AMIN
17/06/20...	2	LOVENO...	2	10	OMAR	AHMED
17/06/20...	9	Dexamét...	3	10	OMAR	AHMED
17/06/20...	10	OXYGENE	8	10	OMAR	AHMED
17/06/20...	4	PARACIT...	3	23	AAAAA	AAAAA
17/06/20...	9	VITAMIN...	3	23	AAAAA	AAAAA
17/06/20...	10	OXYGENE	7	23	AAAAA	AAAAA
17/06/20...	2	LOVENO...	2	31	SANOUR	MANAL
17/06/20...	4	PARACIT...	3	31	SANOUR	MANAL
17/06/20...	7	Hydroxyc...	2	31	SANOUR	MANAL
17/06/20...	8	Dexamét...	3	31	SANOUR	MANAL
17/06/20...	9	VITAMIN...	3	31	SANOUR	MANAL
17/06/20...	10	OXYGENE	11	31	SANOUR	MANAL
17/06/20...	2	LOVENO...	2	37	O0000	O0000
17/06/20...	4	PARACIT...	2	37	O0000	O0000
17/06/20...	6	Asitromy...	3	37	O0000	O0000

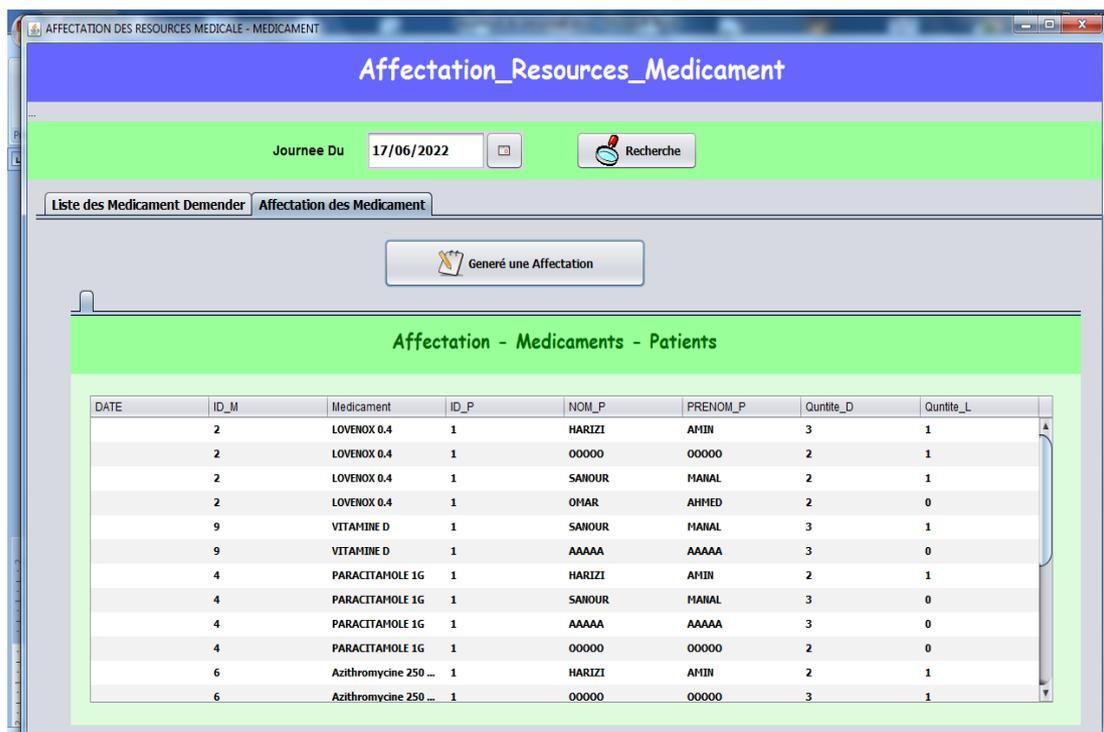
Liste des Medicaments Livrer

DATE	ID_M	Medicament	Qunlité_D	Qunlité_L	Manque
17/06/2022	2	LOVENOX 0.4	9	3	-6
17/06/2022	4	PARACITAMO...	10	1	-9
17/06/2022	6	Asitromycine ...	5	2	-3
17/06/2022	7	Hydrocyclo...	5	1	-4
17/06/2022	8	Dexaméthaso...	7	4	-3
17/06/2022	9	VITAMINE D	6	1	-5
17/06/2022	10	OXYGENE	53	37	-16

Liste des Bilans

DATE	OXYG...	TENSL	TEMPR	DIAMRE	CRP	TDM	ID_P	NOM_P	PREN...
17/06/...	93	11	41	734	35	37	5	HARIZI	AMIN
17/06/...	87	10	42	875	43	53	7	AIBCH	TOUFIK
17/06/...	88	14	39	512	9	0	10	OMAR	AHMED
17/06/...	78	12	37	568	13	0	23	AAAAA	AAAAA
17/06/...	91	13	41	678	23	31	31	SANO...	MANAL
17/06/...	96	12	36	687	23	17	37	O0000	O0000

Figure 3.17 : Récupération de la liste des patients demandant d'un médicament.



DATE	ID_M	Medicament	ID_P	NOM_P	PRENOM_P	Qunlité_D	Qunlité_L
	2	LOVENOX 0.4	1	HARIZI	AHIN	3	1
	2	LOVENOX 0.4	1	00000	00000	2	1
	2	LOVENOX 0.4	1	SANOUR	HAJIAL	2	1
	2	LOVENOX 0.4	1	OMAR	AHMED	2	0
	9	VITAMINE D	1	SANOUR	HAJIAL	3	1
	9	VITAMINE D	1	AAAAA	AAAAA	3	0
	4	PARACETAMOLE 1G	1	HARIZI	AHIN	2	1
	4	PARACETAMOLE 1G	1	SANOUR	HAJIAL	3	0
	4	PARACETAMOLE 1G	1	AAAAA	AAAAA	3	0
	4	PARACETAMOLE 1G	1	00000	00000	2	0
	6	Azithromycine 250 ...	1	HARIZI	AHIN	2	1
	6	Azithromycine 250 ...	1	00000	00000	3	1

Figure 3.18 : Résultat de l'affectation médicaments aux patients.

Conclusion

On a présenté dans ce chapitre notre solution pour améliorer le système de gestion du service de corona virus dans l'hôpital, on a focalisé notre travail sur l'affectation des patients aux médecins puisque elle représente la tâche la plus critique surtout dans le cas de la crise. Dans ce contexte on a proposé un algorithme basé sur la méthode de recuit simulé pour atteindre une affectation optimisée. Pour les médicaments on a donné une solution basée sur le principe de classement par priorité dont le but de sauver le patient qui a l'état est le plus grave.

Pour les infirmiers on a proposé une distribution équitable aux patients pour garantir leurs soins.

On a essayé de récupérer les données réelles du service pour faire nos expérimentations mais malheureusement l'administration a refusé de nous donner cette autorisation en raison de confidentialité, donc on a testé l'efficacité de nos algorithmes sur des données générées aléatoirement.

Chapitre 4

Implémentation

Introduction

Après avoir présenté la conception de notre système, nous entamons l'étape de réalisation, c'est la dernière étape dans le cycle de développement des systèmes d'informations, on va commencer par une présentation de l'environnement de travail et les outils de développement utilisés en suite on va terminer par une présentation des différentes interfaces du système réalisé.

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

1.1 Langage de programmation : JAVA

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès [19] :

- ✓ Java est interprétée : la source est compilé en pseudo code ou byte code puis exécuté par un interpréteur Java (Java Virtual Machine (JVM)).
- ✓ Java est portable : il n'y a pas de compilation spécifique pour chaque plate-forme. Le code reste indépendant de la machine sur laquelle il s'exécute.
- ✓ Java est simple :le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs,
- ✓ Java est fortement typée : toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données.



1.2 Netbeans :

NetBeans est à l'origine un EDI (environnement de développement intégré) Java. NetBeans fut développé à l'origine par une équipe d'étudiants à Prague, racheté ensuite par Sun

Microsystems. Quelque part en 2002, Sun a décidé de rendre NetBeans open-source, sa conception est complètement modulaire. Ce qui fait de NetBeans une boîte à outils facilement améliorable ou modifiable.



NetBeans comprend les fonctions générales suivantes :

- ✓ Configuration et gestion de l'interface graphique des utilisateurs,
- ✓ Support de différents langages de programmation
- ✓ Traitement du code source (édition, navigation, formatage, inspection..),
- ✓ Fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder,
- ✓ Accès et gestion de bases de données, serveurs Web.

1.3 Implémentation de la base de données

Pour l'implémentation de la base de données que notre système utilise, nous avons sélectionné MySQL , qu'est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées .



2. Les interfaces de notre système

L'application est dotée d'une interface graphique composée de fenêtres, permettant à l'utilisateur d'accéder aux différents services d'une façon conviviale. Dans ce qui suit, nous passons en revue les différentes fenêtres de notre application

2.1 Interface authentification

Chaque utilisateur de notre système doit passer par l'étape d'authentification pour accéder à son espace de travail. La figure suivante illustre cette étape :



Figure 4.1 : Capture interface authentification.

Chapitre 4: Implémentation

2.2 Interface Administrateur

L'administrateur a tous les privilèges sur le système. Ses tâches principales sont présentées dans l'interface suivante:



Figure 4.2 : Capture interface administrateur.

2.3 Interface gestion du personnel

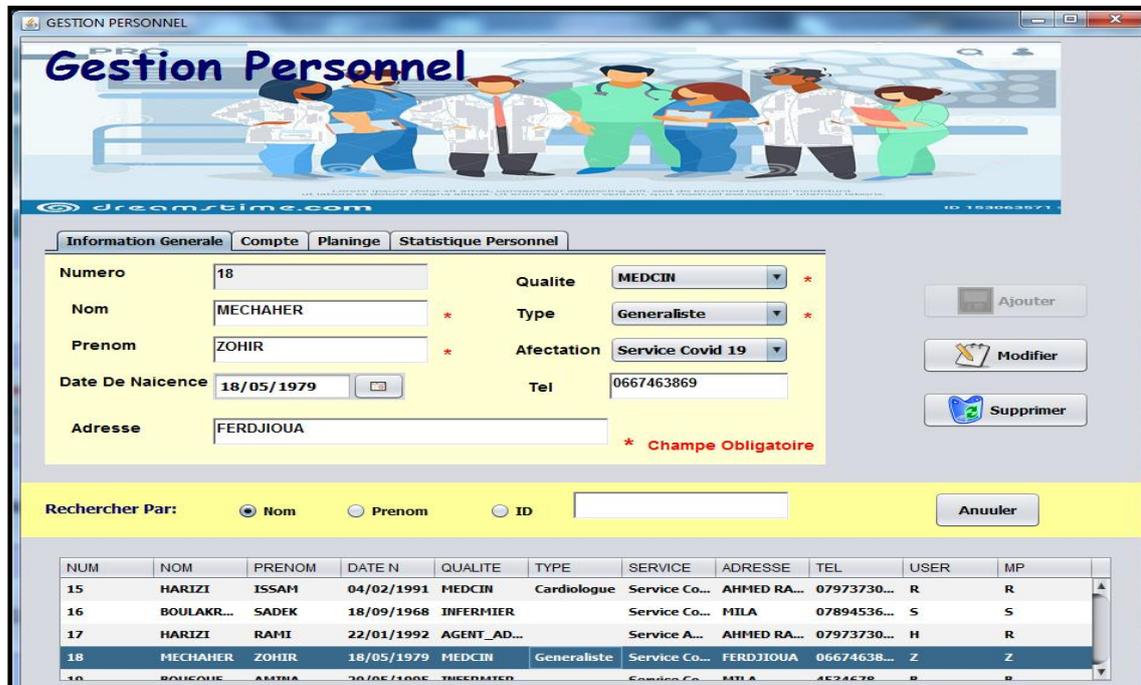


Figure 4.3 : Capture interface gestion du personnel.

Chapitre 4: Implémentation

2.4 Interface gestion du planning

ID_P	JOURNEE	NUM	NOM	PRENOM	QUALITE	TYPE	SI_PRESENT
51	20/06/2022	10	FERKHI	MOULOUD	MEDCIN	Pneumologue	1
52	20/06/2022	14	BENMAKHOUL	FAICL	MEDCIN	Dibetologue	0
53	20/06/2022	15	HARIZI	ISSAM	MEDCIN	Cardiologue	1
54	20/06/2022	24	DALI	NADJIB	MEDCIN	Generaliste	1
55	20/06/2022	23	BENSLAMA	HOUA	MEDCIN	Infectiologue	1
56	20/06/2022	18	MECHAHER	ZOHIR	MEDCIN	Generaliste	0
57	20/06/2022	25	SAMIR	MOUHAMED	MEDCIN	Cardiologue	1

Figure 4.4 : Capture interface gestion du planning.

2.5 Interface admission d'un patient

NUM	NOM	PRE...	AGE	DATE...	SEX	ADR...	COM...	DATEE	SER...	NLIT	DAT...	MOD...	PCR	SERO	TDM	OXG	M_C...
10	OMAR	AHM...	46	23/0...	Hom...	MILA	Oued...	14/0...	Covi...	1 Re...			Posti...	Posti...	83	63	Asth...
11	SAMIR	BOU...	39	14/0...	Hom...	OUED...	Mila	27/0...	Medc...	4 Nor...			Posti...	Posti...	54	78	Diab...
20	AIBC...	TAHRE	43	04/1...	Hom...	SANA...	Mila	20/0...	Covi...	5 Nor...			Posti...	Posti...	45	77	Néant
22	KARIM	ALI	52	17/0...	Hom...	AAA		07/0...	Covi...	2 Re...			Posti...	Posti...	90	67	Infec...

Figure 4.5 : Capture interface admission d'un patient.

2.6 Interface consultation de la fiche navette

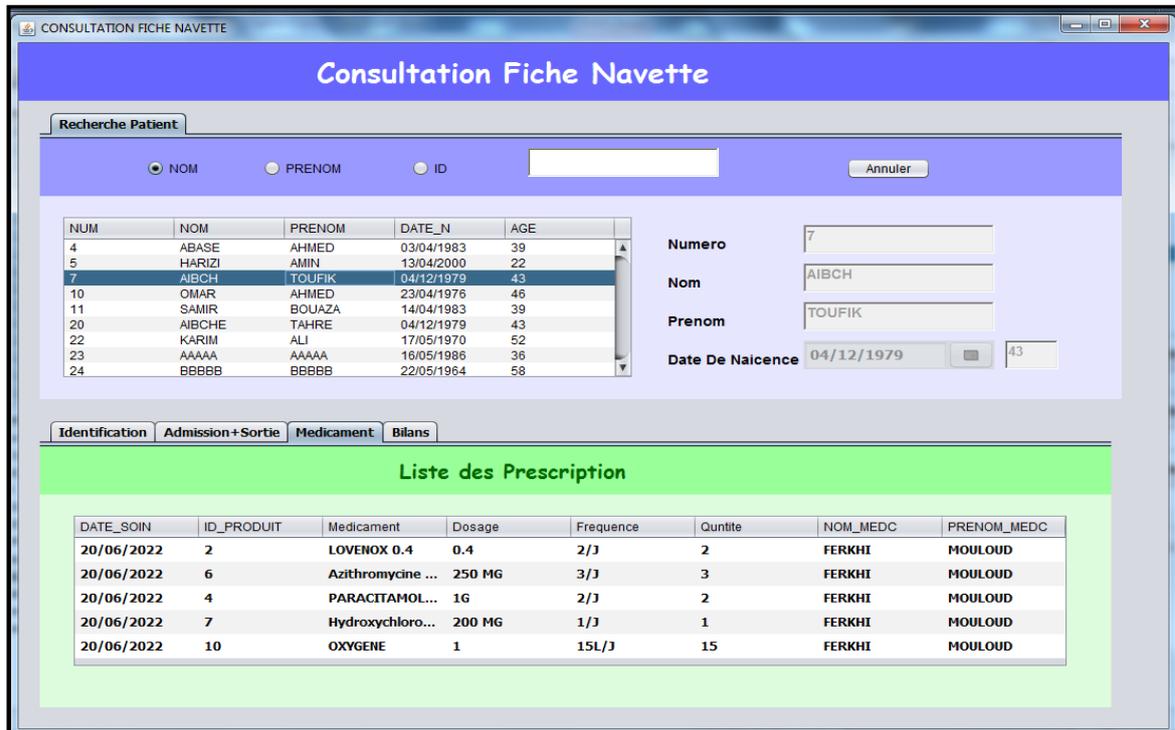


Figure 4.6 : Capture interface consultation de la fiche navette.

2.7 Interface gestion de prescription



Figure 4.7 : Capture interface gestion de prescription.

2.8 Interface gestion des soins médecins

NUM	NOM	PRENOM	DATE_N	AGE	M_CHRONI...
10	OMAR	AHMED	23/04/1976	46	Asthmatique
35	DDDD	HHHH	06/06/1951	71	Asthmatique
42	NNNN	NNNNN	18/06/1948	74	Asthmatique
7	AIECH	TOUFIK	04/12/1979	43	Asthmatique
10	OMAR	AHMED	23/04/1976	46	Asthmatique
35	DDDD	HHHH	06/06/1951	71	Asthmatique
42	NNNN	NNNNN	18/06/1948	74	Asthmatique
41	JJJJ	MMMM	15/06/1962	60	Néant

ID_PRODUIT	Medicament	Dosage	Frequence	Quntite
2	LOVENOX 0.4	0.4	2U	2
6	Azithromycine ...	250 MG	3U	3
10	OXYGENE	1	12L	12

Figure 4.8 : Capture interface gestion des soins médecins.

2.9 Interface gestion des bilans

Menu_Gestion Bilans -Infirmier-

COVID-19

Gestion Bilans -Infirmier-

Etablissement Public Hospitalier Frère Tobal Mila

Bilans

Fiche Navette

Consulter Planning

Déconnexion

Figure 4.9 : Capture interface gestion des bilans.

2.10 Gestion de la pharmacie

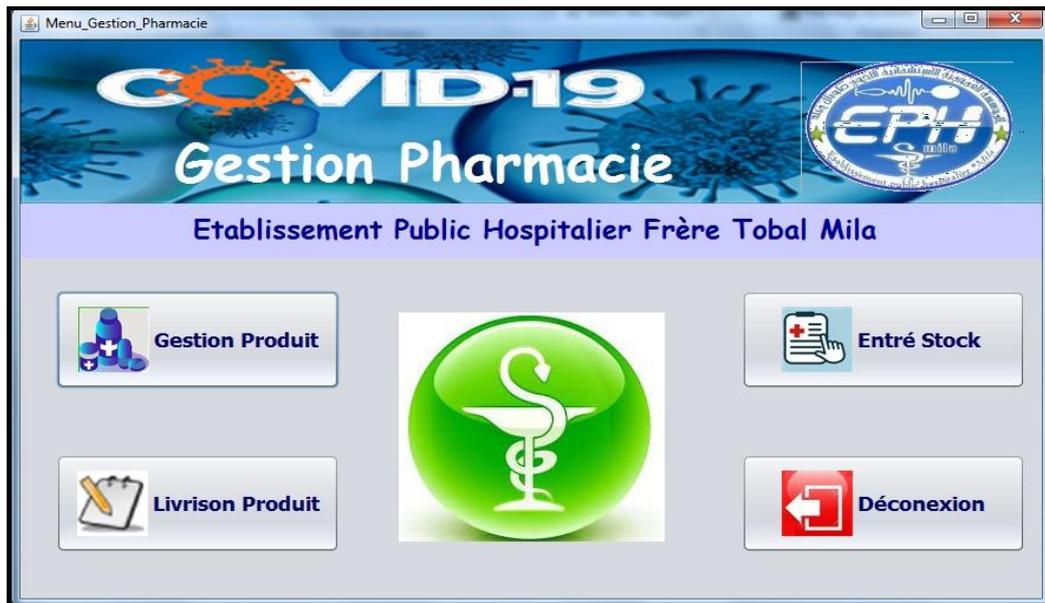


Figure 4.10 : Capture interface gestion de la pharmacie.

Conclusion

Dans ce chapitre nous avons présenté l'environnement de développement et quelques interfaces du système. Cependant cette phase ne présente pas la fin du processus de développement. C'est au tour des deux phases test et validation.

Il faut remettre le système aux utilisateurs testeurs dans le but de détecter les éventuels bugs et anomalies pour les rectifier et assurer la stabilité et la fiabilité du système.

Conclusion général

Conclusion:

La pandémie de Coronavirus a mis le système sanitaire dans une situation critique, surtout pendant la période d'augmentation du nombre de cas positifs, où on a vu une saturation dans les services hospitaliers, ce qui a poussé les autorités à installer des hôpitaux de campagne pour pouvoir sauver les patients. Dans cette situation, on a dû utiliser la technologie de l'information pour pouvoir mieux gérer les ressources disponibles afin de pouvoir dépasser cet état avec un minimum de perte.

Notre projet de fin d'étude s'inscrit dans ce contexte, après un stage fait au service de Coronavirus dans l'EPH Frères Tobal Mila, on a réalisé un système de gestion optimisé des ressources de ce service, notre système permet la gestion et le suivi du soin d'un patient depuis son admission jusqu'à la sortie de l'hôpital. A base d'un module basé sur une technique d'optimisation, ce système permet au chef service d'affecter les patients ayant une maladie chronique aux médecins en prend en considération la compatibilité entre la maladie et la spécialité du médecin dans le but d'assurer une meilleur hospitalisation.

Dans le cas d'une insuffisance des quantités disponibles des médicaments, on a proposé un module d'aide à la décision, il permet de classer les patients demandeurs d'un médicament selon leur état et de donner une dose aux patients les plus prioritaire, cette technique permet de sauver un nombre maximal de patients.

Dans ce projet, nous avons exploité nos efforts et nos connaissances acquises pendant tout le cursus d'étude en informatique pour pouvoir bien analyser le sujet, et réaliser un produit qui répond aux besoins du service. Ce système reste ouvert à toutes évolutions et améliorations des fonctionnalités pour qu'il réponde le maximum possible aux exigences.

Références bibliographiques

Références

- [1] Article publié dans Africaine Journal OnLine (AJOL) le 15/12/2020 sur le système de santé algérien face à la crise sanitaire du COVID-19.
- [2] L'état du système de santé de l'Algérie est déplorable, selon un rapport – Barlamane .
- [3] Conseils de l'OMS concernant la pandémie en cours de maladie à coronavirus 2019 (COVID-19).
- [4] Décret exécutif n° 07-140 du 19 mai 2007 portant création, organisation et fonctionnement des établissements publics hospitaliers et des établissements publics de santé de proximité , Journal Officiel De La République Algérienne N°33.
- [5] Arrêté interministériel du 21 Safar 1433 correspondant au 15 janvier 2012, modifié et complète, fixant les critères de classification des Etablissements publics hospitaliers.
- [6] Documents accordés par la direction de l'EPH Frère Tobal Mila qui comprennent la structure organisationnelle et les tâches de l'établissement.
- [7] Protocole prise en charge COVID 19 cellule de veille du ministère de la santé et de la réforme hospitalière.
- [8] Instruction N° 09 / DGSSRH DU 16 Avril 2020 relative à la démarche diagnostique et thérapeutique du COVID-19.
- [9] Instruction N° 20 / DGSSRH du 23 juillet 2020 relative à L'oxygénothérapie dans la prise en charge de la COVID-19.
- [10] P. Roques et F.Vallé, « UML en action », Livre, 4 IIème éditions ,2007.
- [11] Kazi Aouel Bassim et Rostane Zakaria, « Suivie des enseignements du LMD par application de la méthode 2TUP», mémoire fin d'étude, 2007.
- [12] Baptiste Mille, « Méthodes approchées pour la résolution d'un problème d'ordonnancement avec travaux interférents », mémoire fin d'étude, École Polytechnique de l'Université de Tours.
- [13] Sidi Mohamed Douiri et autre, «Cours des Méthodes de Résolution Exactes Heuristiques et Méta heuristiques », Université Mohammed V RABAT.

- [14] Michel Nabaa, « Approche décentralisée pour résoudre le problème du transport à la demande » Thèse de doctorat de l'université de France, 2009.
- [15] Jamal Rebaine, « Technique de branch and bound », cours hiver département d'informatique et mathématique Université du Québec, 2005.
- [16] M. Dorigo et autres, « Ant colony optimisation » IEEE computationnel intelligence magazine volume 1, 2006.
- [17] Kendall Graham, « AI Methods - Simulated Annealing » Rapport complément de cours, CS - Nottingham Université.
- [18] S. Bouajaja S. et N. Dridi, « Méthode de Recuit Simulé pour l'optimisation de l'affectation d'opérateurs sur une ligne de production », Operational Research: International Journal of Business & Economic Strategy (IJBES) Vol.5, Issue 2, pp. 145-151.
- [19] www.jmdoudoux.fr./java/dej/chap-presentation.htm.