UNIVERSITÉ DU BURUNDI
GRENIER DU SAVOIR DU BURUNDI
Dépôt institutionnel officiel
https://repository.ub.edu.bi

Mise en place d’un système automatise de gestion des services d’accueil dans un centre de soins

NIMBONA, Léonidas

2023-02
UB
https://repository.ub.edu.bi/handle/123456789/495

Document téléchargé depuis le **Grenier du Savoir du Burundi** — https://repository.ub.edu.bi

République du Burundi
Ministère de l'Education Nationale
et de la Recherche Scientifique

Université du Burundi
Faculté des Sciences de l'Ingénieur



Master Génie Informatique

Année Académique : 2021-2022

MISE EN PLACE D'UN SYSTEME AUTOMATISE DE GESTION DES SERVICES D'ACCUEIL DANS UN CENTRE DE SOINS

MEMOIRE

Présenté Par

NIMBONA Léonidas

à la

Faculté des Sciences de l'Ingénieur (FSI)

En vue de l'obtention du grade de

MASTER

en

Génie Informatique

Soutenu le 01/Février/2023, devant le jury composé de :

Dr.	**MUKESHIMANA Michèle**	**Président**
Dr.	**NDAYISABA Longin**	**Vice-Président**
Dr.	**NKUNZIMANA Hilaire**	**Secrétaire**
Pr.	**NDIKUMAGENGE Jérémie**	**Directeur**
Dr.	**SAHINGUVU William**	**Membre**

IDENTIFICATION DES MEMBERS DU JURY

Grade Académique	Nom et Prénom	Rôle
Dr.	MUKESHIMANA Michèle	Président
Dr.	NDAYISABA Longin	Vice-Président
Dr.	NKUNZIMANA Hilaire	Secrétaire
Pr.	NDIKUMAGENGE Jérémie	Directeur
Dr.	SAHINGUVU William	Membre

DEDICACES

Je dédie ce travail :

A ma mère, BARAGASIKA Suzanne à laquelle aucune dédicace au monde aussi éloquent soit-elle ne saura rendre toutes les sacrifices qu'elle a consentis pour moi depuis ma naissance à ce jour. Je t'aime Maman tu es mon ange gardien à travers tes bénédictions et je te dois tout,

A mon père, BAMPFUYEKO Joseph, ce travail est le fruit de tes sacrifices, rien au monde ne vaut les efforts que tu as fournis des jours comme des nuits pour mon éducation et mon bien être. Je t'adore papa tu es le papa le plus formidable au monde,

A mes grands (es) Frères et Sœurs qui m'ont servi d'exemple auquel je me suis référé durant mes études et qui m'ont si affectueusement exprimé leurs soutiens en répondant présent à chacune de mes sollicitations,

A toute la Famille de MBOHO Boniface, qui est tout pour moi et qui ne m'a jamais fait manquer de rien, elle est la source de mon énergie, ma forteresse,

A tous mes amis et connaissances.

NIMBONA Léonidas

REMERCIEMENTS

En premier lieu je tiens à présenter ma reconnaissance la plus profonde et ma gratitude la plus sincère à notre DIEU, Eternel Tout Puissant pour SA GLOIRE.

Tout particulièrement mes obligeances distinguées vont à l'endroit de mon directeur de mémoire Professeur Jérémie NDIKUMAGENGE enseignant et responsable de Master de Génie Informatique à l'université du Burundi qui malgré ses responsabilités n'a ménagé aucun effort pour me prodiguer ses prestigieux conseils et recommandations, son dévouement, sa diligence et son accompagnement. Ses qualités scientifiques et humaines ont été une source d'enrichissement et ont largement contribué à l'aboutissement de ce travail.

C'est aussi avec un réel plaisir et un grand honneur que je remercie les membres du jury qui ont accepté de lire et évaluer ce mémoire.

Je n'oserais oublier d'exprimer mes vifs remerciements à tout le corps professoral du département des TIC à la Faculté des Sciences de l'Ingénieur (FSI) de l'Université du Burundi pour la qualité de leurs enseignements et encadrements ainsi que leurs efforts déployés pour nous transmettre les compétences inestimables.

Egalement je profite de l'occasion pour gratifier tous mes collègues de la troisième promotion de Master pour leur esprit d'équipe fraternelle sans pareille avec lesquels ils ont su maintenir une ambiance conviviale au sein du collectif durant ces deux années mémorables.

Enfin je tiens à témoigner mes sincères remerciements à toute personne qui a contribué de près ou de loin à l'élaboration de ce modeste travail.

Résumé

Un centre de soins est une conjonction entre deux concepts qui doivent cohabiter : un lieu de soins et un lieu d'hébergement et qui devrait accueillir tous les patients qui nécessitent une hospitalisation qu'ils soient programmés ou non programmés (issus des services d'urgence). Les ressources techniques et humaines nécessaires sont importantes et forment une organisation complexe et difficile à gérer. La maîtrise des coûts est l'objectif de la plupart des centres de soins, tout en conservant la sécurité et la qualité des soins.

La gestion des services d'accueil et des places d'hospitalisation sont des étapes que l'on retrouve tout au long du parcours du patient hospitalisé. Elle est jugée consommatrice de temps, insatisfaisante à la fois pour les patients, les professionnels et l'établissement. Son informatisation connaît un retard dans la plupart des pays en général et au Burundi en particulier. Ce travail relate un état des lieux dans l'un des centres de soins que possède le Burundi grâce à une étude qualitative des services de soins et propose un outil informatique automatisant la gestion des services d'hospitalisation.

Mots clé : Système automatisé, gestion, centre de soin, place d'hospitalisation, modélisation.

Abstract

A care center is a conjunction between two concepts that must coexist: a place of care and a place of accommodation and that should receive all patients who require hospitalization, whether they are programmed or not (from the emergency services). The technical and human resources required are important and form a complex and difficult organization to manage. Cost control is the objective of most care centers, while maintaining safety and quality of care.

The management of hospitalization places is a step that can be found throughout the hospitalized patient's journey. It is considered time-consuming and unsatisfactory for patients, professionals and the institution. Its computerization is lagging behind in most countries in general and in Burundi in particular. This work relates the state of play in one of the health care centers in Burundi through a qualitative study of the health care services and proposes a computerized tool to automate the management of hospitalization services.

Key words: Automated system, management, care center, hospitalization place, modeling.

Table des Matières

IDENTIFICATION DES MEMBERS DU JURY i
DEDICACES ii
REMERCIEMENTS iii
Résumé iv
Abstract iv
Table des Matières v
Liste des Figures viii
Liste des Tableaux ix
Abréviations et Acronymes x
CHAPITRE I : INTRODUCTION GENERALE 1
I.1. Contexte et justification du projet 1
I.2. Objectifs du projet 2
I.2.1. Objectif global du projet 2
I.2.2. Objectifs spécifiques projet 2
I.3. Problématique 2
I.3.1. Analyse et critique de l'existant au Centre Hospitalo-Universitaire de Kamenge 3
I.3.2. Notion sur les systèmes 4
I.3.3. Système d'information 5
I.4. Solutions proposées 6
I.5. Résultat escompté 6
I.6. Apport scientifique et technologique 7
I.7. Domaine d'application 7
I.8. Méthodologie de recherche utilisée 8
I.9. Outil mathématique et de development utilisé 8
I.10. Plan du mémoire 9
CHAPITRE II. PRESENTATION GENERALE DU CENTRE HOSPITALO UNIVERSITAIRE DE KAMENGE 10
II.1. Présentation générale du CHUK 10
II.2. Historique du CHUK 10
II.3. Mission du CHUK 10
II.4. Organisation administrative du CHUK 11
II.4.1 Conseil d'Administration 11

II.4.2 Conseil de Direction 11
II.4.3 Conseil Médicale Consultatif (CMC) 12
II.5. Organigramme du CHUK 13
CHAPITRE III : MODELISATION MATHEMATIQUE DU NOUVEAU SYSTEME DE GESTION DES SERVICES D'ACCUIL 14
III.1 Introduction 14
III.2 Revue littéraire 14
III.3 Processus stochastiques 15
III.4 Notion des files d'attente 15
III.4.1. Structure de base d'une file d'attente 15
III.4.2. Concepts et terminologies 16
III.4.3. Notations de Kendall et loi de Little 17
III.4.4 Modèles de files d'attentes 18
III.5. Modèle M/M/1 de Kendall dans la modélisation du nouveau système 18
III.5.1. Description des caractéristiques du Modèle M/M/1 18
III.5.2. Modèle conceptuel du nouveau système à l'aide du modèle M/M/s 21
III.5.3 Evaluation des performances du modèle proposé au CHUK 26
CHAPITRE IV. ANALYSE DES BESOINS ET CONCEPTION DU NOUVEAU SYSTEME DE GESTION DES SERVICES D'ACCUEIL 29
IV.1. Introduction 29
IV.2. Analyses et spécification des Besoins 29
IV.2.1. Cahiers des charges 29
IV.2.2. Estimation prévisionnelle des coûts du projet 30
IV .2.3. Estimation de la taille de notre application par la Méthode de calcul de Points de Fonctions (FP). 31
IV.2.4. Calcul des points de fonction brut 34
IV.2.5. Chronogramme des activités de réalisation du projet 36
IV.2.6. Modélisation de notre nouveau système de gestion des places d'hospitalisation 37
IV.2.7. Intérêt de la modélisation 37
IV.3. Modélisation avec langage UML 37
IV.4. Caractéristiques des Diagrammes UML 38
IV.5. Types de Diagrammes UML 39
IV.6. Choix de diagrammes de modélisation du nouveau système 39
IV.7. Diagrammes de séquence pour les cas d'utilisation pertinents 47

IV.8. Diagrammes de Classes 51
CHAPITRE V. DEVELOPPEMENT ET IMPLEMENTATION DU NOUVEAU SYSTEME DE GESTION DES SERVICES D'ACCUEIL 54
V.1. Introduction 54
V.2. Modèles de cycle de vie d'un logiciel 54
V.2.1 Etapes du cycle de vie du logiciel 54
V.2.2 Différents Modèles de cycle de vie du Logiciel 55
V.3. Outils d'implémentation du nouveau système de gestion des services d'accueil 56
V.3.1. Bases de données 56
V.3.2. Système de gestion de base de données MySQL 57
V.3.3. HyperText Processor PHP 57
V.3.4. Framework 59
V.3.5. Technologies de Développement 60
V.4. Environnement de développement Visual Studio Code 62
V.5. Sécurité du Système d'information 62
V.5.1 Principes génériques de sécurité 63
V.5.2 Sécurisation de notre système d'information 63
V.6. Présentation des fonctionalités du nouveau système 64
V.6.1. Interface d'authentification 64
V.6.2. Interface pour la création d'un dossier médical du nouveau patient 65
V.6.3. Interface pour demander un rendez-Vous 66
V.6.4. Interface pour affecter un lit 67
V.6.5. Interface pour la gestion des hospitalisations 67
CONCLUSION GENERALE ET RECOMMANDATIONS 69
Références bibliographique 70
ANNEXE A 73
ANNEXE B 74

Liste des Figures

Figure 1 : Organigramme du Centre Hospitalo-Universitaire de Kamenge 13
Figure 2 : Structure de base d'une file d'attente 15
Figure 3 : Système de file d'attente à n serveurs identiques 21
Figure 4 : Diagramme de Gantt du projet 36
Figure 5 : Treize diagrammes que propose UML 2 39
Figure 6 : Diagramme de cas d'utilisation 41
Figure 7 : Diagramme de séquence pour le cas d'utilisation « S'authentifier » 48
Figure 8 : Diagramme de séquence pour le cas d'utilisation « créer un dossier médical » 49
Figure 9 : Diagramme de séquence pour le cas d'utilisation « demander rendez-vous » 50
Figure 10 : Diagramme de séquence pour le cas d'utilisation « Affecter un lit » 51
Figure 11 : Diagramme de classe du nouveau système 53
Figure 12 : Modèle de cycle de vie en V 56
Figure 13 : Principe de fonctionnement de l'interpréteur de code PHP 59
Figure 14 : Architecture du Modèle Vue Contrôleur 61
Figure 15 : Interface d'authentification du nouveau système 65
Figure 16 : Interface pour l'enregistrement d'un patient 66
Figure 17 : Interface pour demander un rendez-vous 66
Figure 18 : Interface pour affecter un lit 67
Figure 19 : Interface pour la gestion des hospitalisations 68

Liste des Tableaux

Tableau 1 : Synthèse des caractéristiques du modèle M/M/S 25
Tableau 2 : Calcul de la durée d'attente avec plusieurs valeurs de S 27
Tableau 3 : Types de projet selon le modèle COCOMO 30
Tableau 4 : Valeurs des paramètres a, b et c selon le types de projet 31
Tableau 5 : Identification des groupes de données 32
Tableau 6 : Identification des Entrées (ENT) 33
Tableau 7 : Identification des Sorties 33
Tableau 8 : Calcul du nombre de Point de Fonction Brut 34
Tableau 9 : Identification des acteurs et des cas d'utilisation 40
Tableau 10 : Description détaillés du cas d'utilisation « s'authentifier » 42
Tableau 11 : Description détaillés du cas d'utilisation « Créer un dossier Médical » 43
Tableau 12 : Description détaillés du cas d'utilisation « Affecter un lit » 44
Tableau 13 : Description détaillés du cas d'utilisation « Gérer les consultations » 45
Tableau 14 : Description détaillés du cas d'utilisation « demander un rendez-vous » 46
Tableau 15 : Matrice de Complexité pour les ENTR 73
Tableau 16 : Matrice de Complexité pour les SORT 73
Tableau 17 : Matrice de Complexité pour les INT 73
Tableau 18 : Matrice de Complexité pour les GDI 73
Tableau 19 : Matrice de Complexité pour les GDE 73
Tableau 20 : Caractéristiques générales du système (CGS) 74
Tableau 21 : Degrés d'influence et leurs descriptions 75

Abréviations et Acronymes

NTIC : Nouvelles Technologies de l'Information et de la Communication

CHUK : Centre Hospitalo-Universitaire de Kamenge

UML : Unified Modeling Language

HTML : HyperText Markup Language

CSS : Cascading Style Sheet

PHP : HyperText Preprocessing

CMC : Conseil Médicale Consultatif

FIFO : First In First Out

LIFO : Last In First Out

PS : Processor Sharing

COCOMO : Constructive Cost Model

KLOC : Kilo Lines Of Code

ENT : Entrées

SORT : Sorties

INT : Interrogations

GDI : Groupe de Données Internes

GDE : Groupe de Données Externes

DE : Données Elémentaires

SLD : Sous-Ensemble Logique de Données

GRD : Groupes logiques de Données Référencées

PFB : Points de Fonctions Brutes

DI : Degré d'Influence

DIT : Degré d'Influence Total

PF : Point de fonctions

OMG : Object Management Group

OMT : Object Modeling Technique

OOSE : Object Oriented Software Engineering

MVC : Modèle-Vues-Contrôleurs

BD : Bases de Données

SGBD : Système de Gestion des Bases de Données

WAMP : Windows, Apache, MySQL ET Php

LAMP : Linux, Apache, Mysql Et Php

MAMP : MAC, Apache, MySQL et Php

POO : Programmation Orientée Objet

MIT : Massachusetts Institute of Technologies

SSI : Sécurité du Système d'Information

CHAPITRE I : INTRODUCTION GENERALE

I.1. Contexte et justification du projet

Il y a longtemps que les hommes ont essayé de chercher des solutions pour faciliter les activités quotidiennes. L'une de ces solutions a été la création des instruments pour calculer, traiter ou stocker les informations. L'invention des nombres, des systèmes de numération etc. en sont des exemples illustratifs. Les premiers échanges commerciaux ont eu pour conséquence l'apparition des moyens de plus en plus sophistiqués dans la collecte, traitement, transformation, usage, stockage et partage de l'information. Grace aux systèmes d'informations novateurs modernes, tels que les entrepôts de données et les méthodes de gestion de connaissances dont Datamining, l'information est devenue une marchandise, une denrée rare et chère dans ce monde caractérisée par une concurrence économico-financière à outrance. Beaucoup d'institutions, au Burundi, connaissent un retard notoire dans la maitrise, l'usage et utilisation des technologies de l'information et de la communication.

Or, dans le contexte mondial actuel, aucun organisme ne pourrait ignorer le rôle et la place des Nouvelles Technologies de l'Information et de la Communication (NTIC) dans la maitrise de la concurrence et l'amélioration des services ainsi que les dividendes que l'usage de ces dernières apporte dans l'entreprise au quotidien. Cet usage s'avère être un moyen d'optimiser l'usage des ressources disponibles, augmenter ses performances et atteindre les objectifs fixés dans les délais impartis. Ce retard est devenu un handicap pour le développement socio-économique du pays en général et de chacune de ces institutions en particulier.

Dans la plupart des centres de soins, l'accès aux services ainsi que la recherche des places d'hospitalisation des patients n'est pas facile. Ceci est dû aux problèmes causés par l'utilisation d'un système quasi-manuel de traitement de l'information et de données. Dans la première étape de notre travail de recherche qui avait trait à l'exploration et étude de l'existant, nous avons constaté que la gestion de places se fait sans support informatique. L'échange d'information entre les différents services se fait sur papier. Une telle approche de travail ne peut entrainer que la diminution des performances et efficacité de tout le service. De plus, la connaissance des places disponibles d'hospitalisation est difficile due au fait qu'elle est réalisée grâce au parcours physique par un agent d'appui et dénombrement manuel de chambres vacantes. La transmission de patients vers les places qui leur sont attribuées est confrontée aux mêmes problèmes à cause de mouvements de recherche aléatoire de places ou lits. C'est dans cette optique que notre projet de recherche intitule **« Mise en place d'un Système automatisé de Gestion des Services d'accueil dans un centre de soins »** a été choisi.

I.2. Objectifs du projet

I.2.1. Objectif global du projet

L'objectif global de ce projet repose sur l'amélioration de la qualité des services d'accueil, l'hospitalisation et admission dans les centres de soins par la création d'une application de gestion des places d'hospitalisation des patient ainsi que leurs admissions. Il permettra également la diminution de la durée d'attente des patients et du temps de recherche et de transmissions de ces dernières ainsi que la disposition en temps réels des informations sur la disponibilité des ressources nécessaires à la prise en charge des patients.

I.2.2. Objectifs spécifiques projet

La gestion doit permettre aussi bien une optimisation de l'utilisation des ressources que la satisfaction d'une demande de service de plus en plus exigeante. Il s'agit d'apporter aux systèmes de soins les gains significatifs en termes d'efficacité de productivité tout en assurant la qualité des services et satisfactions des patients. Donc les objectifs spécifiques de notre projet sont entre autres :

1) Diminuer les files d'attente des patients dans les services d'accueil et la durée moyenne de séjour des patients à l'hôpital;
2) Créer un système d'informations sur les places et les services d'hospitalisation ;
3) Créer une base de données actualisée et sécurisée des patients;
4) Echanger de données et informations des patients du service d'hospitalisation entre les médecins soignant et les membres des services d'accueil ;
5) Traiter automatiquement des frais de soins et générer des rapports paramétrables temporairement et spatialement ;
6) Aider dans la prise de décision en rapport avec la capacité et conditions d'accueil pour un centre de soin;

Ainsi, le nouvel outil développé sera inséré dans l'organisation du centre-médicale, il sera capable selon des règles et des modes prédéfinis, d'acquérir des données, de les évaluer, de les traiter, de distribuer des informations contenant une forte valeur ajoutée à tous les partenaires internes ou externes de l'établissement collaborant à une œuvre commune orientée vers un but spécifique, à savoir la prise en charge d'un patient et le rétablissement de celui-ci.

I.3. Problématique

Le manque d'informations exactes en temps réel sur les ressources disponibles dans les centres de soins complique la tâche des agents de service d'accueil et du personnel soignant en général,

ce qui entraine une longue durée d'attente des patients avec un impact sur la qualité et la sécurité dans la prise en charge.

En effet, un nombre de problèmes ont été relevés dans les établissements de soins dont :

1) Des patients en attente prolongée sur des brancards ou dans les couloirs ;
2) Echange direct d'informations entre le service d'accueil et personnel soignant est trop lent ;
3) Des complications ou même des décès inutiles dus en partie à l'inefficacité du système existant ;
4) L'identification de place d'hospitalisation demande la présence et le passage physique d'un agent du service d'hospitalisation ;
5) Difficulté de transfert des patients d'une place à l'autre et dans d'autres établissements.

I.3.1. Analyse et critique de l'existant au Centre Hospitalo-Universitaire de Kamenge

La gestion des places d'hospitalisation est une activité principalement confiée à l'encadrement de proximité. Cette activité consiste à affecter un lit pour chaque patient nécessitant une prise en charge hospitalière quel que soit son mode d'admission. C'est une étape que l'on retrouve tout au long du processus du parcours du patient hospitalisé. [9]

Pour bien débuter notre travail, nous nous sommes rendus sur terrain au CHUK afin de pouvoir observer et analyser toutes les procédures suivies par les patients pour être pris en charge. A son arrivé le patient est reçu dans le service d'accueil où il est attribué un numéro d'ordre et la prise des paramètres (poids, taille et tension), puis il se dirige dans le service de recouvrement pour les paiements éventuels et retourne dans le service d'accueil pour être enregistré et orienté dans le cabinet médical disponible pour la consultation.

Certaines de ces étapes se font à l'ordinateur mais nous avons constaté que le processus prend beaucoup de temps dans la mesure où le patient est obligé de retourner encore dans le service d'accueil pour être enregistrer, il n'y a pas du dossier du patient sur lequel chaque service mentionnerait toutes les actes pour éviter des retours inutiles dans le même service.

Pour être hospitalisé, la décision est prise après le résultat des tests ou directement à son arrivé s'il s'agit d'un patient en situation d'urgence.

Malgré un capacitaire de 422 lits que dispose le CHUK, la recherche d'une place ou lit au quotidien est parfois compliquée. Pour chaque service, les lits disponibles sont recensés pendant la matinée

par un agent du service au moyen d'une visite de toutes les places attribuées au service ce qui veut dire qu'il n'y a pas des mis à jour de ceux qui ont été libérées pendant la journée par des sorties ou des décès. Les répercussions sont négatives pour les patients, les professionnels et l'établissement. La méconnaissance des disponibilités réelles et l'absence d'anticipation dans la prise en charge des patients, entrainent des engorgements à différents niveaux, des retards de prise en charge, des hébergements dans d'autres spécialités, des transferts vers d'autres établissements, des déprogrammations.

Face à cette situation, une gestion informatisée et centralisée des services d'accueil et des lits est une solution pour améliorer la prise en charge du patient dans son parcours hospitalier. Notre travail propose la mise en place d'un système automatisant la gestion des services d'accueil et des places d'hospitalisation pour améliorer la qualité de prise en charge des patients.

I.3.2. Notion sur les systèmes

Un système est un ensemble d'éléments qui interagissent entre eux dans un environnement particulier en formant un tout. Il est donc le siège d'échange et de relations plus ou moins complexes, ses caractéristiques permettent de le caractérisé en tant qu'un objet unique.

On le définit aussi comme étant la matérialisation d'une correspondance entre un ensemble de variables d'entrées et un ensemble de variable de sortie et sa réponse dépend de son état et de ses entrées, elle peut évoluer en fonction du temps. [5]



Figure I. 1: Système avec entrées-sorties

Objectifs d'un système

Un système ne peut exister sans objectif, en effet l'ensemble des éléments qui interagissent dans le système sont organisés pour atteindre la raison d'être de tout système, qui n'est autre qu'un objectif bien déterminé. [5]

I.3.3. Système d'information

Un système d'information est l'ensemble des éléments participant à la gestion, au traitement, au stockage, au transport et à la diffusion de l'information au sein d'une organisation. [2]

Peu importe les supports de cette information (Informatiques ou télécommunications), un système d'information réalise toujours quatre grandes fonctions qui sont la collecte, le traitement, la mémorisation et la diffusion.

La Collecte : C'est le cœur de l'information car c'est ainsi que naît la donnée, à partir d'une saisie directe au clavier, à la souris, par fichiers ou d'une création automatisée, *La Mémorisation* est le stockage des informations brutes ou résultats de traitement. Quant au *traitement*, il consiste à créer de nouvelles informations à partir de celles existantes avec des opérations de tri, de calculs ou de regroupements. Cette fonction utilise des ordinateurs, des serveurs, des systèmes d'exploitation et des logiciels d'application. Ainsi la *diffusion* est l'accès à la mémoire et échange entre les acteurs. [2]

L'an 2019, le Burundi à travers le ministère de la santé publique et de la lutte contre le Sida a procédé au projet d'informatisation des hôpitaux en se procurant de l'outil open source de gestion d'information hospitalière OpenClinic GA. Cet outil permet de gérer la quasi-totalité des flux d'informations hospitaliers. [7]

Comme nous l'avons lu dans le manuel d'utilisateur, le logiciel comprend un nombre de module à savoir: Gestion du dossier du patient, Gestion des contacts, Gestion Financières, Gestion Médicale et Gestion de laboratoires. Mais un module très important qui est la gestion des hospitalisations ne figure pas dans la liste des modules de ce nouveau logiciel ce qui a motivé notre projet de recherche. [10]

Ce module est indispensable dans les systèmes d'informations hospitalières dans la mesure où il comprend plusieurs sous-modules comme les gestions des lits, gestions des transferts, gestion ses admissions selon les modes (programmé ou non programmé), gestions des urgences, gestions des visites, la gestion des rapports d'hospitalisation etc. C'est un sous-système qui gère les patients et le personnel ensemble.

Bien que l'outil OpenClinic existe au CHUK, certains services ne l'utilisent pas y compris le service d'hospitalisation. Le recensement des places et de lits inoccupés est réalisé par l'encadrement sur un décompte manuel puis il est communiqué au service d'urgence et d'accueil par téléphone.

La gestion des lits est réalisée le jour de l'arrivée des patients, sans anticipation entrainant des difficultés.

I.4. Solutions proposées

Par rapport à la problématique précédemment posée, notre travail propose un outil d'automatisation de toutes les activités des services d'accueil et d'hospitalisation. Il s'agit de la création d'une application web qui permettra :

1) Le Partage de données et informations des patients du service d'urgences entre les médecins soignant et les agents du services d'accueil ;
2) La recherche automatique paramétrée selon des critères fixes et pointage de places d'hospitalisation ;
3) Le traitement (calcul) automatique de frais d'hospitalisation ;
4) L'impression de rapports d'hospitalisation et archivages de données y relatifs ;
5) L'existence d'un module d'aide à la prise de décision en rapport avec les capacités et conditions d'accueil et les stratégies à court, moyen et long termes à adopter ;

I.5. Résultat escompté

Selon les objectifs que nous avons évoqués, les résultats attendus à la fin de ce travail sont entre autre:

1) Une application web fonctionnelle permettant le suivi en temps réels des services d'accueil et d'hospitalisation ainsi que l'échange directe des données entre toutes les unités intervenantes dans la prise en charge des patients ;
2) Disposition par les agents du service d'accueil et le personnel soignant d'un réseau informatique doté d'une application réseau qui leur permettent de communiquer à moindre effort physique tout en réduisant le temps de communication;
3) Le nombre de mouvements et de présences dans les enceintes accueil et hospitalisation diminue ;
4) Le service d'accueil fournit en temps opportun des informations utiles aux requérants.

Toutes ces solutions permettront de s'assurer de la cohérence des données, de faciliter l'accès aux informations et de gagner du temps lors de l'accueil d'un patient. Soulignons toutefois que le présent travail de recherche ne se limite qu'à la conception et la réalisation d'une application de gestion des services d'accueil ainsi que leurs hospitalisations.

Par ailleurs, les utilisateurs de notre application bénéficieront d'un outil de travail moderne qui leur permettra de diminuer le temps que prenait la recherche, l`attribution et gestion quasi-manuelle de places pour les patients. Ainsi, la qualité de service se retrouvera améliorée ainsi que l'efficacité des agents du dit service dans sa globalité.

Tout en facilitant la recherche des places d'hospitalisation pour les patients, cette application diminuera d'une part le nombre de cas de complications des patients et le nombre de décès qui était partiellement et éventuellement causé par la lenteur des services d'accueil et l'attribution de places aux patients, et d'autre part, le temps que les patients prennent pour trouver leurs places d'hospitalisation.

I.6. Apport scientifique et technologique

Les nouvelles technologies de l'information et de la communication ne cessent d'envahir les différents domaines des activités humaines.

Cela s'explique par son apport incontestable aux entreprises qui se sont dotées de ce nouvel outil. En effet cet outil permet la résolution des problèmes complexes tels que le partage d'informations sur la capacité d'hospitalisation à tout moment dans un centre de soins, l'automatisation des rapports, la génération automatique des billets de sorties, la transmission des patients d'un centre médical vers un autre (Module d'aide à la prise de décision) ; l'exécution rapide des tâches, etc.

Notre travail se fera dans cette même perspective où le système proposé permettra la mise en application des connaissances acquises tout au long de notre cursus au moyen de la modélisation et minimisation du parcours du patient à l'aide de la théorie des files d'attente. Ainsi, il servira de base pour des recherches ultérieures qui pourrait être menées dans le cadre du développement des systèmes similaires orientés dans le domaine de développement de l'informatique de santé ou analogue.

I.7. Domaine d'application

Le présent travail s'inscrit dans le domaine de santé. Il s'agit de la gestion des services d'accueil et d'hospitalisation notamment la gestion des admissions, l'attribution des places aux patients nécessitant l'hospitalisation dans une institution de soins et le suivi en temps réels des informations y relatives. Le but principal est d'améliorer la qualité des services.

I.8. Méthodologie de recherche utilisée

Un travail de recherche scientifique nécessite la disposition des données sur lesquelles seront menées des études.

Pour collecter les données utiles à la réalisation de notre travail, nous avons procédé par la recherche documentaire qui nous a permis de reconstituer le cadre théorique précédent en s'appuyant sur des revues de documents bibliographiques qui sont des livres, les mémoires, les publications et rapports consultés. Cette approche nous a permis d'effectuer un processus systématique pour connaître en profondeur les détails de ce qui a été étudié et réalisé.

Une autre approche que nous avons utilisée est celle des observations par comptage et chronométrage des activités sur terrain qui ont été effectuées dans des visites organisées en date du 30 mai, du 2juin et du 16 juin 2022 au Centre Hospitalo-Universitaire de Kamenge. Ainsi, nous ne pouvons pas ignorer le rôle de l'internet qui nous a permis de connaitre plus d'informations relatives à notre sujet et d'avoir les documents sur l'outil formel de modélisation UML et le modèle mathématique utilisé dans notre projet.

I.9. Outil mathématique et de development utilisé

L'existence d'un système informatisé découle non seulement de la part des informaticiens mais également des concepts et méthodes mathématiques avec lesquels il doit être préalablement modélisé.

Les mathématiques consistent d'ailleurs en un langage, qui permet de transcrire des problèmes de nature quantitative : c'est la modélisation. Une fois cette transcription faite, des outils sont disponibles pour résoudre ces problèmes partiellement ou complètement, on ramène ensuite la solution dans son contexte d'origine. [8]

Dans le cadre de notre travail, nous avons utilisé la Théorie des files d'attente qui est une technique de la Recherche Opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions. Nous avons choisi de travailler avec le modèle M/M/s qui est un modèle à n serveurs avec stockage infini. [11]

Ainsi pour l'implémentation du système les différents outils ont été utilisés dont MySQL comme système de gestion des bases de données, Microsoft Visio pour la modélisation UML; PHP : Langage de programmation utilisé coté serveur, HTML, CSS, JavaScript et Bootstrap pour le design et programmation coté client; Visual Studio Code : Environnement de développement.

I.10. Plan du mémoire

Hormis la conclusion générale, l'ensemble de ce mémoire s'articule autour de cinq chapitres.

Le premier chapitre consiste en l'introduction générale, la fixation des objectif, identification du problème et les solutions à apporter ainsi que la méthodologie et les outils de travails.

Le deuxième chapitre intitulé « Présentation générale du CHUK » présente le milieu d'étude dans lequel se sont appuyées les recherches menées et l'analyse du système existant.

Le troisième chapitre est intitulé « Modélisation Mathématique du nouveau système de gestion des places d'hospitalisation ». Il introduit une approche mathématique à base des files d'attente qui effectue les calculs des coûts de séjour des patients dans le centre ainsi que les probabilités d'occupations des ressources du centre de soins.

Le quatrième chapitre intitulé « Analyse des besoins et Modélisation du nouveau système de gestion des places d'hospitalisation » consiste en la spécification des besoins pour la mise en place du système et sa représentation par UML.

Le cinquième chapitre intitulé « Développement et implémentation du nouveau système de gestion des places d'hospitalisation » parle sur les différentes technologies de développement ainsi que les outils d'implémentation. C'est la partie qui présente également les différentes interfaces de notre système.

Dans cette partie, il était question de la description du contexte de notre projet et l'exposition du problème tout en menant une analyse critique sur le système existant. Les solutions que nous comptons apporter à la problématique posée ainsi que les objectifs fixés qui seront démontrés dans les chapitres qui vont suivre. Pour cela, procédons à la brève présentation du milieu travail.

CHAPITRE II. PRESENTATION GENERALE DU CENTRE HOSPITALO UNIVERSITAIRE DE KAMENGE

Dans cette partie nous allons faire une brève présentation du centre Hospitalo-Universitaire de Kamenge ainsi que sa structure organisationnelle et administrative donnée dans un organigramme.

II.1. Présentation générale du CHUK

Le CHUK est un établissement sanitaire public qui a ouvert ses portes en 1984. Il est situé au Nord-Ouest de la ville de Bujumbura, capitale économique du Burundi. Il fait partie des quatre grands hôpitaux qui desservent la capitale Bujumbura et les provinces environnantes. Sans pour autant avoir (aujourd'hui) un plateau technique important, il est le centre de référence nationale. [12]

II.2. Historique du CHUK

Le Centre Hospitalo-universitaire de Kamenge a été créé par le Décret N°100/121 du 28 décembre 1984 comme une administration personnalisée dotée de l'autonomie de gestion. Le CHUK, Hôpital National Universitaire, est placé sous l'autorité hiérarchique du Ministère de l'Education Nationale et de la Recherche Scientifique. Il offre jusqu'à présent des possibilités d'hospitalisation grâce à sa capacité d'accueil de 420lits. Il a ouvert ses portes en 1984 avec un effectif de 43 employés, mais jusqu'à nos jours, cet effectif s'élève à 730 employés, toutes catégories confondues.

Le Centre Hospitalo-universitaire de Kamenge (CHUK) héberge en son sein une faculté de Médecine et offre un terrain de stage aux étudiants d'institutions d'enseignement supérieur et aux élèves d'écoles paramédicales. Le CHUK comprend deux types de personnel à savoir : Le personnel hospitalier et le personnel hospitalo-universitaire.

Le personnel hospitalier se subdivise en 3 catégories à savoir : La catégorie des Cadres de Direction qui comprend à son tour des licenciés dans plusieurs domaines, des médecins généralistes ainsi que des médecins spécialistes, la catégorie des Techniciens (Agents de collaboration) ainsi que la catégorie des Agents d'exécution. Signalons que le personnel Hospitalo-universitaire est régi par le statut du personnel de l'Université du Burundi et assure sa prestation au CHUK par une convention le liant avec ce dernier. [13]

II.3. Mission du CHUK

Les missions de cette institution sont :

1) Assurer les soins et les services médico-sanitaires de haut niveau ;
2) Assurer l'approvisionnement en intrants nécessaires à l'accomplissement de ses missions ;

3) Assurer la recherche dans le domaine de la santé ;
4) Assurer l'encadrement des stagiaires dans les limites de ses capacités d'accueil ;
5) Collaborer avec d'autres établissements de soins, d'enseignement et de recherche appartenant à d'autres organisation ou institutions ;
6) Collaborer avec les autres centres Hospitaliers tant nationaux qu'internationaux ;
7) Assurer la formation initiale et continue du personnel hospitalier. [7]

II.4. Organisation administrative du CHUK

Le centre comprend les organes suivants : le Conseil d'Administration, la Direction, le Comité de Direction, le Conseil de Direction et le Conseil Médicale Consultatif (CMC). [7]

II.4.1 Conseil d'Administration

Le conseil d'Administration dispose des pouvoirs étendus de gestion et d'administration du Centre. Sous réserve des instructions du gouvernement, il définit les orientations de l'action du Centre. Il comprend :

1) Un représentant du Ministère ayant la santé publique dans ses attributions : *Président* ;
2) Un représentant du Ministère ayant l'enseignement Supérieur et la recherche scientifique dans ses attributions : *Vice-Président* ;
3) Le Directeur Général du CHUK : *Secrétaire* ;
4) Le Doyen de la Faculté de Médecine de l'Université du Burundi : *Membre* ;
5) Un représentant du Ministère ayant les finances dans ses attributions : *Membre* ;
6) Un Représentant des Bénéficiaires des soins : *Membre*. [7]

II.4.2 Conseil de Direction

Le conseil de direction est un organe interne consultatif. Il est composé par le Directeur Général, les Directeurs, les Chefs de services administratifs et des départements médicaux, le Doyen de la faculté de Médecine et le Chef de service du Nursing.

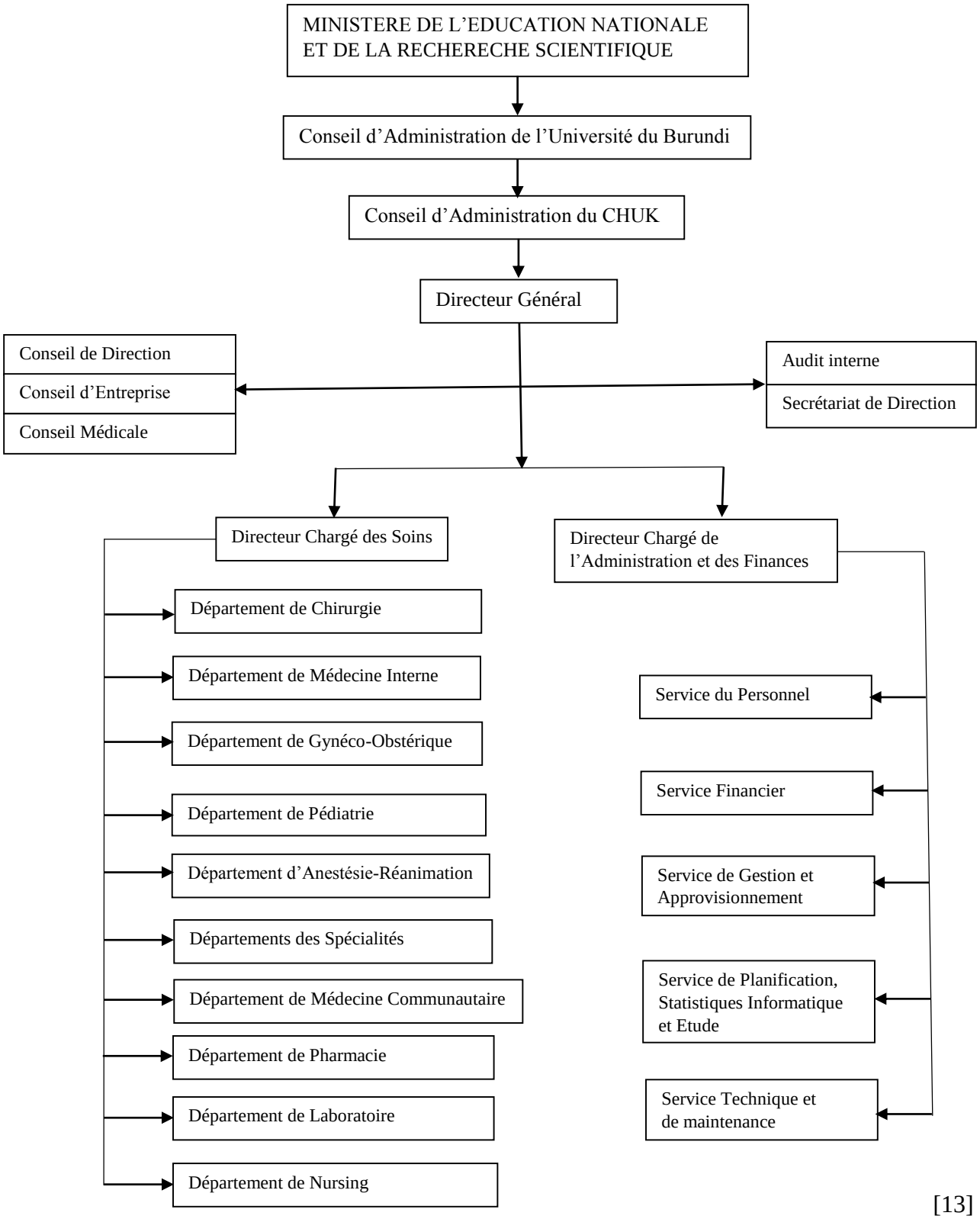
L'administration et la gestion quotidiennes du centre sont assurées par un Directeur Général assisté par deux Directeurs chargés respectivement des soins, de l'Administration et des finances. Quant au comité de direction, il assiste le directeur Général dans la prise des décisions. Il est composé du Directeur Général, des Directeurs et du Doyen de la Faculté de Médecine. Le directeur Général et le directeur chargé de l'administration et des finances en assurent respectivement la présidence et le secrétariat. [7]

II.4.3 Conseil Médicale Consultatif (CMC)

Le conseil Médicale Consultatif est un organe qui assiste la Direction et le Conseil d'Administration dans l'accomplissement de leurs missions d'assurer les soins de qualité, et la recherche médicale. Il est composé du Directeur Chargé des Soins, du Doyen de la Faculté de Médecine, de tous les chefs de départements de soins et services de santé et du Chef de service du Nursing. [7]

II.5. Organigramme du CHUK

Conformément au Décret n°100/09 du 23 janvier 2019 portant révision du Décret n°100/056 du 21 avril 1992 portant réorganisation et fonctionnement du Centre Hospitalo-Universitaire de Kamenge, son organigramme est donné par la figure 1.



[13]

Figure 1 : Organigramme du Centre Hospitalo-Universitaire de Kamenge

CHAPITRE III : MODELISATION MATHEMATIQUE DU NOUVEAU SYSTEME DE GESTION DES SERVICES D'ACCUIL

III.1 Introduction

La Théorie des files d'attente est une technique de la Recherche opérationnelle qui permet de modéliser un système admettant un phénomène d'attente, de calculer ses performances et de déterminer ses caractéristiques pour aider les gestionnaires dans leurs prises de décisions. Généralement les files d'attentes se forment lorsque les clients arrivent de façon aléatoire pour se faire servir. [20]

Notre travail s'inscrit dans le cadre des systèmes de soins de santé où un nombre des patients arrivent selon une certaine fréquence pour se faire soigner. Bien entendu que le flux d'arrivé des patients va en créer des files pour attendre le service.

Dans ce chapitre avec le modèle de file d'attente, nous allons étudier l'évolution de la file par le calcul du temps moyens d'attente, du nombre moyen de patients dans la file d'attente, des probabilités d'avoir un certain nombre de patients dans le système tout en définissant une distribution des arrivées et du taux de services (patients servis) afin d'utiliser le modèle qui convient au problème pour obtenir les bons résultats.

L'idée de base est d'évaluer les coûts nécessaires à la réduction des files d'attentes. Nous allons appliquer la théorie des modèles de files d'attentes au service d'hospitalisation dans plusieurs spécialités afin de les modéliser et de prévoir ainsi leurs évolutions et comportements suite aux certains chocs et changements.

III.2 Revue littéraire

En recherche opérationnelle, l'application des modèles de files d'attentes est très vaste. Afin de mieux comprendre les principes de bases et de voir tout le potentiel, nous nous sommes servis des utilisés deux livres. Le premier est « Introduction to Queuing Theory » de Robert Cooper qui présente la théorie de bases avec toutes les hypothèses nécessaires aux calculs de plusieurs paramètres et résultats. Et "An Introduction to Queueing Theory, Modeling and Analysis in Applications" de U.Narayan Bhat. On nous présente les démonstrations des notions de bases, les calculs nécessaires ainsi que leurs applications.

Ces deux livres nous permettent de voir les différentes distributions applicables aux arrivés et aux temps de service, les conditions qui rendent le système stable ainsi que les conséquences.

Ainsi, nous nous somme servi du troisième livre qui nous présente sommairement l'application des files d'attentes dans le domaine médical. Il s'agit de « Operations research for health planning and administration » de Stephen Duckett. Dans ce dernier livre, on nous présente l'exemple d'une salle d'urgence. Toutes ces notions seront présentées en détails dans les points qui suivent.

III.3 Processus stochastiques

On appelle *processus stochastique* ou *processus aléatoire* toute famille de variables aléatoires X_t. Cela signifie qu'à tout $t \in T$ est associée une variable aléatoire prenant ses valeurs dans un ensemble numérique E.

On note le processus X_t. Si T est dénombrable, on dit que le processus est *discret* ; si T est un intervalle, on dit que le processus est *permanent*.

Autrement dit, un processus stochastique $\{x_t, t \in T\}$ est une collection de variables aléatoires indexées par un paramètre t et définies sur un même espace de probabilités (Ω, F, P)

La variable x_t représente l´état du processus au temps t et l'ensemble de toutes les valeurs possibles pour cette variable est appelée l'espace des états du processus et sera noté N. [16]

III.4 Notion des files d'attente

L'étude mathématique d'un système d'attente, se fait le plus souvent par l'introduction d'un processus stochastique approprie. Généralement, on qualifie de processus stochastique tout phénomène d'évolution temporelle dont l'analyse peut être soumise au calcul des probabilités.

La théorie des files d'attente consiste en l'étude de systèmes où des *clients* se présentent à un dispositif de service, appelé *serveur*. Puisqu'un client occupe le serveur pendant un certain temps, les autres clients doivent attendre avant d'être servis, formant ainsi une *file d'attente* [13]

III.4.1. Structure de base d'une file d'attente

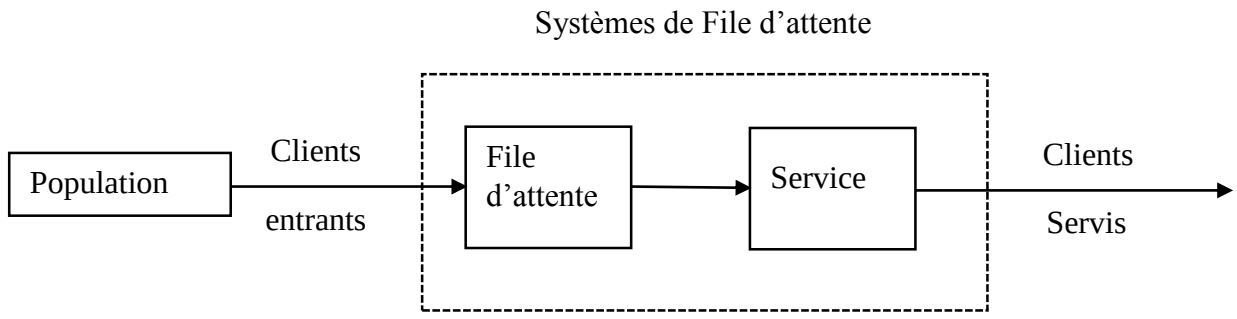


Figure 2 : Structure de base d'une file d'attente

Population: La population constitue la source de clients potentiels. Elle est caractérisée par son nombre d'élément (fini ou infini).

File d'attente: La file d'attente est caractérisée par le nombre maximum permis de clients en attente (fini ou infini)

Clients: Les clients (issus de la population) se joignent au système avec un taux moyen d'arrivée.

Service: Le service peut être assuré par un ou plusieurs serveurs. Le temps qui s'écoule entre le début et la fin de service d'un client est dénoté le temps de service suivant une distribution de probabilité. Donc le taux de service est une autre caractéristique du système.

Discipline de service : C'est la règle d'ordonnancement des clients au service, FIFO: first in first out, LIFO : Last in first out, PS : Processor Sharing, un serveur donne à chaque client en attente une *'tranche' de service,* ALEA un serveur libre choisit un client au hasard dans la file. [20]

III.4.2. Concepts et terminologies

Quand nous commençons à analyser un système de file d'attente, l'état de ce dernier dépend beaucoup de l'état initial et du temps écoulé. Nous disons alors que le système est en situation *transitoire,* et son étude est alors très complexe.

C'est pourquoi dans la théorie des files d'attentes, nous préférons faire l'étude une fois le système a atteint sa situation d'équilibre où les états du système sont essentiellement indépendants de l'état initiale et du temps déjà écoulé. On suppose en quelque sorte que le système est en opération depuis un très long moment.

$P_n(t)$ = Probabilité d'avoir n clients dans le système au temps t,

S = Nombre de Serveurs,

λ_n = Taux moyen d'arrivée (espérance mathématique du nombre d'arrivées par unité de temps) de nouveaux clients dans le systèmes lorsque n clients sont dans le système. Paramètre définissant la distribution exponentielle des arrivées lorsque n clients sont dans le système,

$1/\lambda_n$ = Temps moyen entre les arrivées lorsque n clients sont dans le système,

μ_n = Taux moyen de service d'un client lorsque n clients sont dans le systèmes. Paramètre définissant la distribution exponentielle du service d'un client lorsque n clients sont dans le système,

1/ μ_n =Temps moyen de service d'un client lorsque n clients sont dans le système.

P_n = Probabilité qu'il ait n clients dans le système,

L = Nombre moyen (Espérance mathématique) de client dans le système,

L_q = Nombre Moyen de clients dans la file d'attente (excluant ceux dans le service),

W = Temps moyen dans le système,

W_q = Temps moyen dans la file (excluant le temps de service),

S = Nombre de serveurs. [20]

III.4.3. Notations de Kendall et loi de Little

La notation de Kendall se construit à l'aide de 4 concepts abrégée de la forme a/b/c/N,

Où :

- a ∈ {M, G, ...} précise le processus d'arrivée: M (Markovien) pour un processus de Poisson, G (général) pour un processus quelconque non précisé, ...;
- b ∈ {M, G, ...} précise la distribution des durées de service: M (Markovien) pour des durées indépendantes et distribuées exponentiellement avec moyenne identique pour chaque serveur, G pour des durées quelconques, ...;
- c ∈ {1, 2, ..., ∞} indique le nombre de serveurs;
- N ∈ {1, 2, ..., ∞} précise la capacité du système, c'est-à-dire le nombre de clients qui peuvent simultanément s'y trouver.

Soulignons que la notation a/b/c/N est parfois complétée par des champs supplémentaires indiquant par exemple la discipline de service (FIFO, aléatoire, ...), le nombre de clients existant dans l'univers considéré, etc.

Quant à la loi de Little, elle est libérée à base de la notation de Kendall où λ est le taux moyen des arrivées et E(k) le nombre de clients dans le système.

Elle s'applique également à des serveurs à l'intérieur d'un serveur, la seule exigence est que le système soit stable et non-préemptif, ce qui exclut des états de transition comme la mise en route ou l'arrêt d'un serveur.

Voici la formulation explicite de la loi de Little : Lorsqu'un client, ayant terminé son service, quitte le système, il laisse, en moyenne, derrière lui, un nombre de clients égal à E(k). Ce client a trouvé en arrivant E(k) clients déjà présents et a passé dans le système un temps, E(T).

Si on appelle λ, le taux moyen des arrivées on a:

{Nombre moyen de clients arrivés pendant le séjour du client dans le système}
= λ E(t) = {nombre moyen de clients qu'il laisse}

Et en régime permanent, si T est le temps passé dans la file :

$$E(k) = \lambda E(T)$$

D'où $L = \lambda W$ [19]

III.4.4 Modèles de files d'attentes

Selon la notation de Kendall, Il existe plusieurs modèles de files d'attentes, mais les plus importants et couramment utilisées sont : Le modèle *M/M/1* qui est un modèle à serveur unique, où les arrivées suivent une loi de Poisson de taux λ, la loi de service exponentielle de paramètre μ, et une capacité de la file qui peut s'étendre à l'infinie. Le modèle *M/M/s* qui est une extension la plus simple de la file *M/M/1* : les hypothèses probabilistes sont les mêmes, seul change le nombre de serveurs. Un autre modèle de file d'attente est le modèle *M/M/1/N* qui fonctionne aussi comme la file *M/M/1*, sauf que le nombre maximal de clients dans le système est limité à N : un client qui arrive lorsque la file est pleine est rejeté.

Les files G/*G*/1 avec des arrivées et services plus généraux (aléatoires ou non). Par exemple le *M/D*/1 : arrivées poissonniennes et service déterministe (constant) et *D/M*/1 : arrivées déterministes (régulières) et service exponentiel.

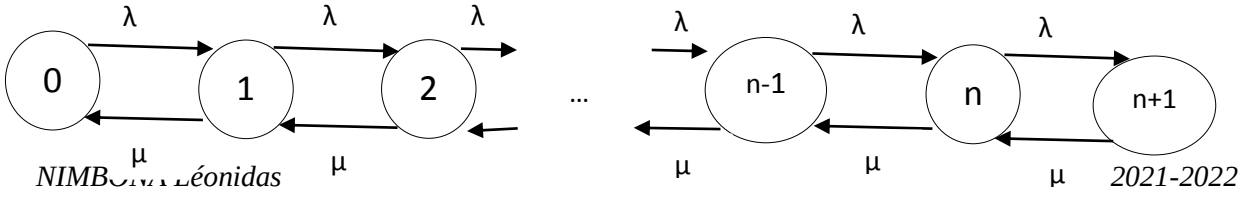
III.5. Modèle M/M/1 de Kendall dans la modélisation du nouveau système

III.5.1. Description des caractéristiques du Modèle M/M/1

La file d'attente M/M/1 est le plus simple des modèles de file d'attente utilisés dans la pratique. Les arrivées sont supposées se produire dans un processus de Poisson avec un taux λ. Cela signifie que le nombre de clients (patients) N(t) arrivant pendant un intervalle de temps (0, t) a une distribution de Poisson. Considérons un modèle de file d'attente où les arrivées et les départs se produisent comme dans un processus de naissance et de mort. On a :

$\left.\begin{array}{l}\lambda_n \equiv \lambda \text{ pour tout n} \\ \mu_n \equiv \mu \text{ pour tout n}\end{array}\right\}$ Signifie, indépendants du nombre de clients dans le système

Diagramme de transition entre les états



Alors

$$P_0 = \frac{1}{\sum_{n\geq 1} \frac{\prod_{i=0}^{n-1} \lambda i}{\prod_{i=1}^{n} \mu i}} = \frac{1}{\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n} \quad (1)$$

Sous l'hypothèse que λ<μ (le taux d'arrive est plus petit que le taux de service) $\frac{\lambda}{\mu} < 1$, et progression géométrique

$$\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu}\right)^n = \frac{1}{1-\frac{\lambda}{\mu}} \quad (2)$$

Notons que la condition $\frac{\lambda}{\mu} < 1$, assure que le système peut atteindre l'équilibre.

Partant de la relation (1) et (2), nous avons,

$$P_0 = \frac{1}{\frac{1}{1-\frac{\lambda}{\mu}}} = 1 - \frac{\lambda}{\mu}$$

De plus $P_n = \frac{\prod_{i=0}^{n-1} \lambda i}{\prod_{i=0}^{ni} \mu i} P_0 = P_0 \left(\frac{\lambda}{\mu}\right)^n = \left(1 - \frac{\lambda}{\mu}\right)\left(\frac{\lambda}{\mu}\right)^n$

Introduisons la notion de facteur d'utilisation $\rho = \frac{\lambda}{\mu}$, ρ représente en quelque sorte la proportion du temps que le serveur est occupé.

D'où $P_n = (1-\rho)\,\rho^n$ avec n = 0,1, 2,…

Calculons maintenant les caractéristiques de la file d'attente *M/M/1*

1) Nombre moyens de clients dans le système

$$L = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n(1-\rho)\rho^n$$

$$= (1-\rho)\rho \sum_{n=1}^{\infty} n\rho^{n-1} = (1-\rho)\,\rho \sum_{n=0}^{\infty} \frac{d}{d\rho}(\rho)^n$$

$$= (1-\rho)\,\rho \frac{d}{d\rho}\left(\sum_{n=0}^{\infty} \rho^n\right)$$

$$= (1-\rho)\,\rho \frac{d}{d\rho}\left(\frac{1}{1-\rho}\right) = (1-\rho)\,\rho \frac{1}{(1-\rho)^2}$$

Donc $L = \frac{\lambda}{\mu-\lambda}$ (3)

2) Nombre moyen de clients dans la file d'attente

$$L_q = \sum_{n=1}^{\infty} (n-1)P_n$$

$$= \sum_{n=1}^{\infty} nP_n - \sum_{n=1}^{\infty} P_n$$

$$= L - (1-P_0)$$

$$= \frac{\lambda}{\mu-\lambda} - \frac{\lambda}{\mu}$$

D'où

$$L_q = \frac{\lambda^2}{\mu(\mu-\lambda)} \tag{4}$$

3) Temps moyen pour un client dans le système

$$W = \frac{L}{\bar{\lambda}} = \frac{L}{\lambda} = \frac{\lambda}{\mu-\lambda}\frac{1}{\lambda} = \frac{1}{\mu-\lambda} \quad \text{(puisque } \bar{\lambda} = \sum_{n=0}^{\infty} \lambda_n P_n = \lambda \sum_{n=0}^{\infty} P_n = \lambda)$$

4) Temps Moyen pour n clients dans la file d'attente

$$W_q = \frac{L_q}{\bar{\lambda}} = \frac{L_q}{\lambda} = \frac{\lambda^2}{\mu(\mu-\lambda)}\frac{1}{\lambda}$$

$$= \frac{\lambda}{\mu(\mu-\lambda)} \tag{5}$$

5) Priorités du Modèle M/M/1

Comme nous modélisons un système de services de soins de santé, il peut y avoir plusieurs priorités qui sont des patient en situation d'urgence. Les priorités qui peuvent interrompre un service en cours pour une personne moins prioritaire est de type préemptive. Pour notre modèle M/M/1, prenons deux classes de patients. La classe 1 a une priorité absolue sur la classe 2. Notons par L_i le nombre de patients par classe dans le système. Définissons encore S_i comme le temps de séjour, c'est-à-dire, le temps dans la file d'attente plus le temps passe dans le service, par classe de patient. Le nombre moyen de patients dans le système et le temps de séjour moyen par classe sont donnés respectivement par :

$$\mathrm{E}(L_1) = \frac{\rho_1}{1-\rho_1} \quad \text{et} \quad \mathrm{E}(L_2) = \frac{\rho_2}{(1-\rho_1)(1-\rho_1-\rho_2)} \quad , \tag{6}$$

$$\text{et} \quad \mathrm{E}(S_1) = \frac{1/\mu}{1-\rho_1} \quad \text{et} \quad \mathrm{E}(S_2) = \frac{1/\mu}{(1-\rho_1)(1-\rho_1-\rho_2)} \quad . \tag{7}$$

Rappelons que ρ est l'intensité du système ou le taux d'occupation(taux d'utilisation). Lorsque le taux d'arrivée est diffèrent entre les classe, nous aurons $\rho_1 = \lambda_1/\mu$ pour la classe 1 et $\rho_1 = \lambda_2/\mu$ pour la classe 2. Par contre si un patient urgente ne peut interrompre un service en cours et doit attendre que le serveur se libère avant de profiter du service, alors il s'agit d'une priorité non préemptive. Les statistiques concernant le nombre moyen de clients dans le système et le temps de séjour moyen se calculent maintenant :

$$\mathrm{E}(L_1) = \frac{(1-\rho_2)\rho_1}{1-\rho_1} \quad \text{et} \quad \mathrm{E}(L_2) = \frac{(1-\rho_1(1-\rho_1-\rho_2))\rho_2}{(1-\rho_1)(1-\rho_1-\rho_2)} \tag{8}$$

$$\text{et } E(S_1) = \frac{(1-\rho_2)/\mu}{1-\rho_1} \quad \text{et } E(S_2) = \frac{(1-\rho_1(1-\rho_1-\rho_2))/\mu}{(1-\rho_1)(1-\rho_1-\rho_2)} \tag{9}$$

En ce qui concerne les services d'hospitalisations, il est clair qu'il y ait des priorités (patients urgents non programmés) qui peuvent interrompre les procédures d'hospitalisations en cours. Dans ce cas nous aurons les types de priorité préemptives. Toutefois, nous ne disposons pas d'assez d'information sur les patients pour pouvoir utiliser un système de priorité dans le modèle.

III.5.2. Modèle conceptuel du nouveau système à l'aide du modèle M/M/s

La file d'attente multiserveur *M/M/s* est le modèle le plus utilisé dans l'analyse des stations de service avec plus d'un serveur. L'arrivée des clients (patients) est supposée suivre un processus de Poisson et les temps de service sont supposés avoir une distribution exponentielle.

Comme nous modélisons les flux des patients dans le service d'hospitalisation, les clients sont les patients qui passent dans les cabinets des médecins pour être soigné, évidemment les serveurs sont les médecins. Enfin, la file d'attente est le nombre de patients dans la salle d'attente. La structure de notre système est composée d'un nombre S de serveurs (cabinets) monté en parallèle, voir la figure III.2 :

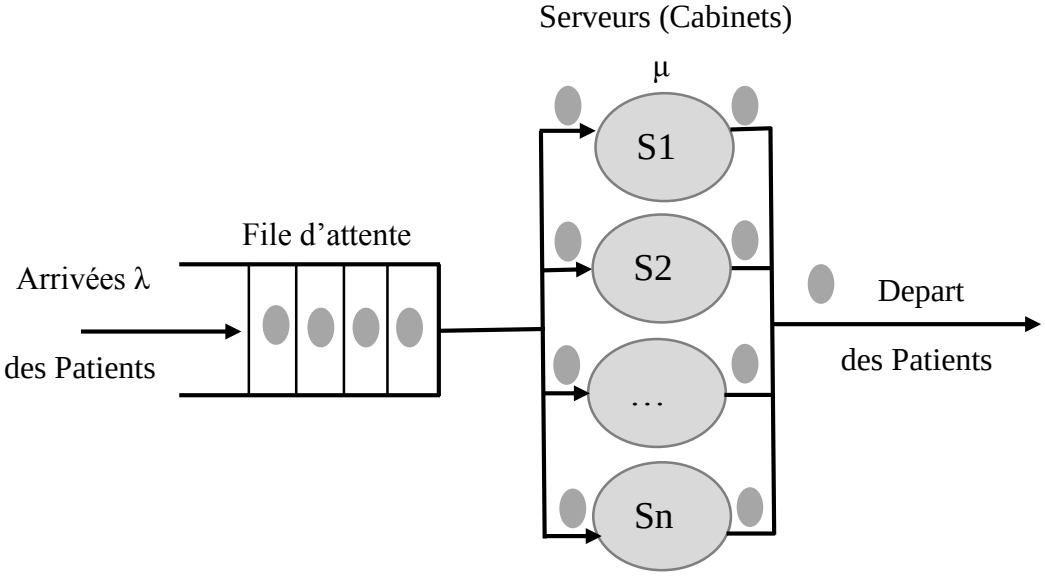


Figure 3 : Système de file d'attente à n serveurs identiques

Prenons alors :

S1 : Cabinet du premier médecin,
S2 : Cabinet du deuxième médecin,
.
.
.
Sn : Cabinet du n^{ième} médecin,

N(t) : Nombre de patients arrivée dans un intervalle de temps (0, t),

Soit λ le taux d'arrivée et μ le taux de service. (Cela signifie que les temps d'inter arrivée et les temps de service ont des distributions exponentielles avec des densités $\lambda e^{-\lambda x}$ (x > 0) et $\mu e^{-\mu x}$(x > 0) respectivement. Notons que pour le cas présent, le taux de service μ est le même pour tous les serveurs.

Soit l'état Ek, où k patients présents dans le système (Hôpital), le taux de naissance et de mort est donné par :

$$\begin{cases} \lambda_k = \lambda & \\ \mu_k = k\mu & \forall k < S \; et \; \in N \\ \mu_k = S\mu & \forall k \geq S \; et \; \in N \end{cases},$$

1) La Probabilité d'avoir K Patients dans le système est donnée par :

$$\begin{cases} P_k = \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k p_0 & si \; k < S \\ P_k = \frac{1}{S^{k-s}\, S!}\left(\frac{\lambda}{\mu}\right)^k p_0 & si \; k \geq S \end{cases} \quad (10)$$

Signalons que le nombre de serveurs S= $\sum_{i=1}^{n} S_i$

2) Calculons maintenant la probabilité qu'il n'y ait aucun patient dans le système P_0 :

$$\sum_{k=0}^{\infty} P_k = 1$$

En remplaçant P_k par les expressions trouvées ci-dessus :

$$P_0 \left[\sum_{k=0}^{s-1}\left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} + \sum_{k=s}^{\infty}\left(\frac{\lambda}{\mu}\right)^k \frac{1}{S^{k-s}S!}\right] = 1$$

$$P_0 \left[\sum_{k=0}^{s-1}\left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} + \frac{S^s}{S!}\sum_{k=s}^{\infty}\left(\frac{\lambda}{S\mu}\right)^k\right] = 1$$

Posons $r = \frac{\lambda}{S\mu}$, on a :

$$P_0 \left[\sum_{k=0}^{s-1}\left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} + \frac{S^s}{S!}\sum_{k=s}^{\infty}(r)^k\right] = 1,$$

Le second terme entre crochets s'écrit :

$$\frac{S^s}{S!}\sum_{k=s}^{\infty}(r)^k = \frac{S^s}{S!}[r^s + r^{s+1} + r^{s+2} + \cdots] = \frac{S^s}{S!} r^s \, [1 + r + r^2 + \cdots]$$

$[1 + r + r^2 + \cdots]$, est une série géométrique de terme général : $r = \frac{\lambda}{S\mu}$

La série est convergente si $\frac{\lambda}{S\mu} \leq 1$ et sa somme est égale à $\frac{1}{1-r}$,

On peut alors écrire :

$$\frac{S^S}{S!}\sum_{k=s}^{\infty} r^k = \frac{S^s}{S!}\ r^S \frac{1}{1-r} = \frac{S^s}{S!}\ \frac{r^s}{1-r}\ ,$$

En remplaçant r par sa valeur, on a :

$$\frac{S^s}{S!}\sum_{k=s}^{\infty} r^k = \frac{S^s\left(\frac{\lambda}{S\mu}\right)^s}{1-\left(\frac{\lambda}{S\mu}\right)} = \frac{1}{S!}\left(\frac{\lambda}{S\mu}\right)^s \frac{1}{1-\left(\frac{\lambda}{S\mu}\right)},$$

$$\text{Soit } P_0 = \frac{1}{\sum_{k=s}^{\infty} \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^s \frac{1}{1-\left(\frac{\lambda}{S\mu}\right)}} \qquad (11)$$

3) Calcul du nombre moyen des patients L_q dans la file

Il existe des patients dans la file d'attente, si k > S. Soient P_0, P_{S+i} les probabilités qu'il y ait respectivement 0 et (S+i) patients dans le système,

$$L_q = \sum_{i=0}^{\infty} i\, P_{S+i} = \sum_{i=0}^{\infty} i \left(\frac{\lambda}{\mu}\right)^{S+1} \frac{S^S}{S!} P_0,$$

$$L_q = P_0 \frac{S^S}{S!}\left(\frac{\lambda}{S\mu}\right)^{S+1} \sum_{i=1}^{\infty} i \left(\frac{\lambda}{S\mu}\right)^{i-1},$$

Posons : $\frac{\lambda}{S\mu} = x$

$$ix^{i-1} = derrivé\ de\ x^i$$

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x},$$

$$\sum_{i=0}^{\infty} ix^i = \frac{\partial}{\partial x}\left(\frac{1}{1-x}\right) = \frac{1}{(1-x)^2},$$

D'où

$$L_q = P_0 \frac{S^S}{S!}\left(\frac{\lambda}{S\mu}\right)^{S+1} \frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2},$$

$$L_q = \frac{1}{S!S}\left(\frac{\lambda}{\mu}\right)^{S+1} \frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2} P_0, \qquad (12)$$

4) Recherchons le temps moyen, W_q que les patients attendent dans la file

Soient L_q le nombre moyen de patients dans la file,

λ, le taux d'arrivé des patients,

W_q, le temps moyen d'attente des patients dans la file.

En utilisant la formule de LITTLE, nous avons :

$$L_q = \lambda W_q,$$

$$.W_q = \frac{L_q}{\lambda},$$

$$.W_q = \frac{1}{\lambda}\frac{1}{S!S}\left(\frac{\lambda}{\mu}\right)^{S+1}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2}P_0$$

$$W_q = \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^{S}\frac{1}{S\mu}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2}P_0 \quad (13)$$

Ainsi la probabilité d'attendre un temps supérieur à t (dans le cas où nous avons S cabinets de médecins) est données par :

$$P(T > t) = P(T > 0)e^{-(\mu S-\lambda)t},$$

Avec $(T > 0) = W_q\mu(1-\rho)$,

Et $\rho = \frac{\lambda}{S\mu}$, taux de Traffic (Occupation, charge d'un médecin)

5) Calcul du temps de séjour d'un patient à l'hôpital

Il est donné par W= $W_q + \frac{1}{\mu}$. En remplaçant W_q par sa valeur nous avons :

$$W = \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^{S}\frac{1}{S\mu}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2}P_0 + \frac{1}{\mu} \quad (14)$$

6) Calcul du nombre de lits Occupés pendant un temps t,

Pour calculer le nombre de lits occupés, nous allons considérer les lits comme des serveurs pour que k patients aient un service d'hospitalisation. Signalons que CHUK à une capacité de 422Lits au totale.

Prenons alors k : nombre de patients à hospitaliser,

S : nombre total de Lits,

$\bar{g}$: nombre moyen de lits occupés.

On a:

$$\bar{g} = \sum_{k=0}^{\infty} kP_k + \sum_{k=S+1}^{\infty} S\,P_k$$

$$\bar{g} = \sum_{k=1}^{S} k\frac{\left(\frac{\lambda}{\mu}\right)^k}{S!}P_0 + \sum_{k=S+1}^{\infty} S\frac{\left(\frac{\lambda}{\mu}\right)^k}{S!S^{k-s}}P_0$$

$$\bar{g} = P_0\frac{\lambda}{\mu}\left[\sum_{k=1}^{S}\frac{\left(\frac{\lambda}{\mu}\right)^k}{(k-1)!} + \sum_{k=S+1}^{\infty}\frac{\left(\frac{\lambda}{\mu}\right)^{k-1}}{S!S^{k-s-1}}\right]$$

Posons j = k-1, nous avons :

$$\bar{g} = P_0 \frac{\lambda}{\mu}\left[\sum_{j=0}^{s-1}\frac{\left(\frac{\lambda}{\mu}\right)^j}{j!} + \frac{S^S}{S!}\sum_{j=s}^{\infty}\frac{\left(\frac{\lambda}{\mu}\right)^j}{S^j}\right]$$

$$\bar{g} = P_0 \frac{\lambda}{\mu}\left[\sum_{j=0}^{s-1}\left(\frac{\lambda}{\mu}\right)^j \frac{1}{j!} + \frac{S^S}{S!}\sum_{j=s}^{\infty}\left(\frac{\lambda}{S^j\mu}\right)^j\right]$$

Le terme entre crochets représente l'expression $\frac{1}{P_0}$, il reste alors :

$$\bar{g} = \frac{\lambda}{\mu} \quad (15)$$

Le flux d'arrivées des patients est λ, personne ne disparaît dans le système, le flux de sortie doit être μ, hors saturation. Ce nombre est égal au nombre moyen de lits actifs par μ (le taux d'hospitalisation) de chacun.

7) Calculons en fin le nombre moyen des patients dans le système

Ce nombre est donnés par la somme du nombre de patients hospitalisés (occupants les lits) et du nombre de patients en attente.

Ainsi nous avons :

$$L = W_q + \bar{g}$$

$$L = \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S \frac{1}{S\mu}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2} P_0 + \frac{\lambda}{\mu} \quad (16)$$

Synthèse de caractéristiques calculées avec notre modèle :

Tableau 1 : Synthèse des caractéristiques du modèle M/M/S

Mesure de performance	Equation
Probabilité qu'il n'y ait aucun patient dans le système	$P_0 = \frac{1}{\sum_{k=0}^{s-1}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S \frac{1}{1-\left(\frac{\lambda}{S\mu}\right)}},$
Probabilité qu'il y ait k patients dans le système	$\begin{cases} P_k = \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k p_0 & si\ k < S \\ P_k = \frac{1}{S^{k-s}S!}\left(\frac{\lambda}{\mu}\right)^k p_0 & si\ k \geq S \end{cases}$
Nombre moyen de patients dans la file d'attente	$L_q = \frac{1}{S!S}\left(\frac{\lambda}{\mu}\right)^{S+1}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2} P_0,$

Nombre moyen de patients dans le système	$L = \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S \frac{1}{S\mu}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2} P_0 + \frac{\lambda}{\mu}$
Temps moyen d'attente des patients en file	$.W_q = \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S \frac{1}{S\mu}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2} P_0.$
Temps moyen de séjour d'un patient à l'hôpital	$W = \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S \frac{1}{S\mu}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2} P_0 + \frac{1}{\mu}$

III.5.3 Evaluation des performances du modèle proposé au CHUK

Pour parvenir à l'évaluation des performances et pouvoir calculer certaines caractéristiques selon les besoins et objectifs du centre de soins, il nous faut obligatoirement la disposition des données y relatives.

Comme nous n'avons pas eu les données exactes du CHUK, nous allons formuler les objectifs et besoins du centre et prendre les valeurs au hasard pour se servir d'exemple.

Prenons les activités dans le service de médecine interne et évaluons les performances : Prenons que le service de médicine interne comptes 3 médecins généralistes (3 cabinets fonctionnels) qui prennent en charge chaque jour les patients. Les patients arrivent selon un processus de Poissons de paramètre λ= 7 patients par heure. La durée de consultation d'un patient est une variable aléatoire exponentielle de moyenne 0.4h (μ=1/0.4).

Calculons :

- Taux d'occupation (la charge) d'un médecin,
- Nombre de patients dans la file,
- Temps d'attente des patients dans la file.

- Connaissant le taux d'arrive des patients λ et la durée de consultation d'un patient, le (Taux d'occupation) d'un médecin est donné par :

$\rho = \frac{\lambda}{S\mu}$, puisque S=3cabinets,

$\rho = \frac{7}{3*\frac{1}{0,4}}$ = 0,93 soit 93%

- Calculons maintenant le nombre des patients en attente de service,

$L_q = \frac{1}{S!S}\left(\frac{\lambda}{\mu}\right)^{S+1} \frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2} P_0$, avec $\rho = \frac{\lambda}{S\mu}$, nous avons

$$L_q = \frac{1}{S!S}\left(\frac{\lambda}{\mu}\right)^{S+1}\frac{1}{(1-\rho)^2}P_0,$$

Cherchons d'abord P_0 avec S=3 serveurs

$$P_0 = \frac{1}{\sum_{k=0}^{2}\frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k + \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S\frac{1}{1-\rho}} = \frac{1}{1+\left(\frac{7}{1/0,4}\right)+\frac{1}{2}*\left(\frac{7}{1/0.4}\right)^2+\frac{1}{6}\left(\frac{7}{1/0,4}\right)^3*\frac{1}{1-0.93}}$$

$$= \frac{1}{1+(7*0.4)+\frac{1}{2}*(7*0.4)^2+\frac{1}{6}(7*0.4)^3*\frac{1}{0.07}} = 0.0166$$

Le nombre moyen de patient en attente devient alors,

$$L_q = \frac{1}{3!3}\left(\frac{7}{1/0.4}\right)^{3+1}\frac{1}{(1-0.93)^2} * 0.0166,$$

$$= \frac{1}{27*3}(7*0.4)^4 * \frac{1}{(0.07)^2} * 0.0166,$$

= 2.57 soit 3 patients en attente

- Et le temps d'attente est donnée par :

$$W_q = \frac{1}{S!}\left(\frac{\lambda}{\mu}\right)^S\frac{1}{S\mu}\frac{1}{\left(1-\frac{\lambda}{S\mu}\right)^2}P_0.$$

$$= \frac{1}{3!}\left(\frac{7}{1/0.4}\right)^3\frac{1}{(3/0.4)}\frac{1}{(1-0.93)^2} * 0.0166$$

$$= \frac{1}{6}(7*0.4)^3\frac{0.4}{(3)} * \frac{1}{(0.07)^2} * 0.0166$$

= 1.65h

Les médecins sont occupés à 93%, mais les patients doivent attendre plus d'une heure 30minutes. Si on décide d'ajouter d'autres cabinets dans le service avec objectifs que le temps d'attente des patients soit inférieur à 45min. Cherchons combien de médecins doit-on y affecter ?Pour trouver le nombre de médecins nécessaire à y affecter, nous essayons avec plusieurs valeurs de S et calculer W_q et le nombre en files :

Tableau 2 : Calcul de la durée d'attente avec plusieurs valeurs de S

Valeur de S	P_0	L_q (patients)	W_q(heures)
S=4	0.0208	7.61 soit 8	1.08
S=5	0.0297	1.73 soit 2	0.69
S=6	0.0401	0.91 soit 1	0.36

Si on y affecte encore 2 médecins la contrainte d'attendre moins de 45min est satisfaite, par ailleurs le nombre de patients en file diminue aussi.

On pouvait y ajouter 3cabinet mais le coût d'un médecin peut être considéré par l'hôpital. Sinon c'est 6medecins qui pourraient minimiser le temps moyen d'attente des patients en file.

L'analyse des files d'attente a été importante dans conception de notre système. Dans les centres de soins, les files d'attentes ont tendances à se former, les arrivées aléatoires des patients combinées à la variabilité des temps de service créent temporairement des congestions dans le système, d'où la création des files d'attente.

Dans cette partie, nous avons analysé l'évolution de la file dans les centres de soins tout en posant l'hypothèse que les taux d'arrivées des patients **(λ)** suivent une distribution de Poisson alors que les temps de services suivent une loi exponentielle de paramètre **μ**. Ainsi nous avons constaté qu'à partir des calculs des performances de la file, nous pouvons contribuer dans la prise de décision du centre visant à améliorer la qualité de service de la prise en charge des patients.

CHAPITRE IV. ANALYSE DES BESOINS ET CONCEPTION DU NOUVEAU SYSTEME DE GESTION DES SERVICES D'ACCUEIL

IV.1. Introduction

Pour programmer une application (développer un logiciel), il ne convient pas de se lancer tête baissée dans l'écriture du code, il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation définissant les modèles et étapes de la réalisation. C'est cette démarche antérieure à l'écriture du code que l'on appelle modélisation et son produit est un modèle. [22]

Dans cette section, nous allons mener une étude détaillée sur notre système tout en exprimant les besoins fonctionnels. Cette modélisation va nous permettre de produire une représentation abstraite qui exclut certains détails du monde réel ce qui conduit à la réduction de la complexité du problème en éliminant les détails qui n'influencent pas son comportement de manière significative, c'est-à-dire visualiser le système comme il doit être, valider le modèle vis-à-vis des utilisateurs, spécifier les structures des données et le comportement du système et donc de fournir un guide pour la construction de notre système . Et cette représentation reflète ce qui est important pour la compréhension et la prédiction de notre problème dépendamment des objectifs fixes.

Ainsi, nous allons présenter les diagrammes UML utilisés dans la modélisation, leurs types et leurs rôles dans la conception des systèmes. Et pour parvenir au développement de notre application, la modélisation UML nous a été importante. Pour ce faire, nous allons représenter quelques diagrammes UML comme le diagramme de séquence, le diagramme de classe et le diagramme cas d'utilisation.

IV.2. Analyses et spécification des Besoins

Le processus de définition des besoins est une étape qui vise à établir les fonctionnalités du futur système et à exprimer les attentes des futurs utilisateurs. C'est la première des étapes du cycle de vie du logiciel. Elle permet de décrire à la fois le système et l'environnement dans lequel le système sera exécuté. Cette étape comprend également une étude de faisabilité de la part des experts. C'est sur la base de cette étape que le contrat est signé avec un cahier des charges. [26]

IV.2.1. Cahiers des charges

Le cahier des charges est, dans un projet informatique, le document dans lequel le client exprime ses besoins, c'est-à-dire les caractéristiques fonctionnelles et techniques attendues de la solution informatique [40]. Ces derniers jours, le secteur de santé a introduit l'usage des NTIC, dans la gestion des centres de soins. Il s'agit du logiciel OpenClinic qui permet principalement la gestion des services de comptabilités des centres et la gestion des patients.

En plus de ces fonctionnalités, le présent travail consiste en la mise en place d'un système automatisant la gestion des places d'hospitalisation qui est un domaine large et complexe de l'hôpital. Il permettra d'effectuer les opérations de transfert et d'affectation des places d'hospitalisation (lits) aux patients qui en ont besoins ainsi que la mise à jour et la diffusion des informations sur l'occupation de la capacité totale d'accueil du centre selon les différents départements. Le système permettra également la génération automatique des rapports ainsi que l'impression de quelques documents médicaux comme les fiches ou dossiers médicaux et les billets de sorties pour les patients hospitalisés. Il y aura aussi une interface de dialogue pour les patients qui sollicitent un rendez-vous à un des docteurs ou spécialistes du centre. Le logiciel sera une application web accessible sur le réseau du centre de soins par authentification préalable afin de limiter les accès aux utilisateurs non autorisés.

IV.2.2. Estimation prévisionnelle des coûts du projet

Dans la gestion des projets, l'estimation est une étape qui permet de connaître le coût d'une "vue de l'esprit (vision) " qui deviendra réalité au bout d'un temps fini. Elle concerne: l'effort de développement (coût), la durée du projet (temps), et autre (équipement, voyage, formation), et ajouter (la logique des calculs, les hypothèses). [41]

Bien qu'il existe différents méthode d'estimations, le modèle le mieux documenté, dont les paramètres sont adaptables à l'environnement, est le modèle COCOMO (Constructive COst Model) décrit par Barry Boehm en 1981. Son Objectif est d'évaluer les critères de projet comme l'effort, la durée, l'effectif et la productivité [41]. Cette méthode suppose l'existence d'une corrélation entre la taille (en instructions source) d'un programme et la charge consommée. Ainsi Barry Boehm distingue trois types de projets selon la taille qui sont donne dans le tableau IV.1.

Tableau 3 : Types de projet selon le modèle COCOMO

Type	**Taille**	**Description**
Organique	< 50 KLOC	projets simples menés avec de petites équipes
Moyen (Semi-detached)	< 300 KLOC	projets intermédiaires menés avec des équipes mixtes
Imbriqué (Embedded)	> 300 KLOC	projets complexes devant obéir à des ensembles de contraintes

KLOC représente le nombre, en milliers, de lignes de code (LOC = Lines Of Code) ; en fait il s'agit du nombre d'instructions source.

Pour le calcul de l'effort (charge), le temps de développement, ainsi que la taille de l'équipe et la productivité, Barry Boehm propose les formules suivantes :

$$Effort(charge) = a \times (KLOC)^b \text{ en homme-mois(HM)}$$
$$TDEV = 2.5 \times (effort)^C \text{ en mois}$$
$$Taille\ de\ l'équipe = \frac{Effort}{TDEV}$$

Ainsi, sur base des résultats d'analyse statistique sur un échantillon significatif de projet, Boehem a pu déterminer les valeurs des paramètres a, b et c suivant le type de projet comme elles sont données dans le tableau IV.2.

Tableau 4 : Valeurs des paramètres a, b et c selon le types de projet

Type de projet	**a**	**b**	**c**
Organique	*2.4*	*1.05*	*0.38*
Semi-detached	3.0	1.12	0.35
Imbriqué	3.6	1.20	0.32

IV .2.3. Estimation de la taille de notre application par la Méthode de calcul de Points de Fonctions (FP).

Etant donné que la taille du logiciel n'est connue qu'à la fin de sa réalisation, utilisons la méthode de calcul de point de fonctions (FP) qui a été développée à la fin des années 1970 par Allan J. Albrecht chez IBM. Ainsi, il propose les formules suivantes :

$$PF = FA * PFA$$
$$FA = 0.65+0.01\ Somme(DI)$$
$$ISL = 118.7 \times PFA - 6490$$

Avec

FP: Points de Fonction,
PFA: Point de Fonction Ajusté
ISL: (Instruction Sources Livrable) [42]

Pour calculer le nombre de point de fonction, Cette Méthode introduit cinq composants fonctionnels à savoir :

1. **ENT**: Entrée de donnée ou de contrôle qui requiert un traitement comme écrans, transactions, fichier de données, etc.
2. **SORT**: Sortie de donnée ou de contrôle après un traitement du système comme écrans, transactions, fichier de données, etc.
3. **INT**: Entrée ou sortie d'une requête demandant une réponse immédiate du système comme interruptions, appels, etc.

4. **GDI**: Regroupement logique de données ou de contrôle interne au système comme bases de données, répertoires, etc.
5. **GDE**: Fichier ou exécutable qui sortent des limites du système comme bibliothèques, bases de données externes, paquetages génériques, etc.

Ainsi que les trois niveaux de complexité de ces composants : complexité **faible**, **moyenne** ou **élevée.** Signalons que la complexité d'un GDI ou d'un GDE est déterminée par le nombre de Sous-ensembles logiques de donnée (SLD) et le nombre de données élémentaires (DE) qui constituent les champs de données. Une DE est un attribut au sens de MERISE.

La complexité des ENTR, INT et SORT est déterminée par le groupe logique de données référencées (GRD) qui correspondant aux GDI ou GDE traités et par les DE manipulées. Les INT et les SORT peuvent sembler similaires, mais la distinction est que les INT génère des données non traitées, tandis que SORT génère des données traitées. Chacune des transactions et chacun des types de fichiers ci-dessus sont classés en trois niveaux de complexité : **faible**, **moyen** ou **élevé**.

Pour notre projet identifions les ENT, SORT, INT, GDI et GDE et leurs degrés de complexité dans le tableau suivant :

1) Indentification des groupements de données internes et externes (GDI &GDE)

Tableau 5 : Identification des groupes de données

Nom du groupe de donnée	GDI ou GDE	Nbre de SLD	Nbre de DE	Complexité
1. Docteur ;	GDI	1	5	faible
2. Patient ;	GDI	1	9	faible
3. Personnel ;	GDI	1	6	faible
4. Lit ;	GDI	1	3	faible
5. Département ;	GDI	1	3	faible
6. Facture ;	GDI	1	5	faible
7. Place ;	GDI	1	4	faible
8. Service ;	GDI	1	2	faible
9. Rapport ;	GDI	1	4	faible
10. Utilisateur ;	GDI	1	5	faible
11. Rendez-vous ;	GDI	1	3	faible
12. Horaires ;	GDI	1	4	faible
13. Consultation ;	GDI	1	3	faible
14. Réception.	GDI	1	3	faible

Le tableau ci-dessus a été établi sur référence du tableau 4 de l'annexe A, page 73. Comme notre application est un système de gestion, il contrôle les données dont il est capable de créer, de mettre

à jour ou de supprimer, il n'aura pas besoins des données au-delà de ses limites. Pour ce, on conclut d'après le tableau que le groupes de données externes GDE n'existe pas pour ce cas présent.

2) Identification des entrées ENT et leurs complexités

Le dénombrement des entrées s'appuie sur les GDI, et leurs DE. Pour notre système, elles sont constituées par les différents écrans comme:

Tableau 6 : Identification des Entrées (ENT)

Nom de du composant	Nbre de SLD	Nbre de DE	Complexité
1. Ecran d'enregistrement d'un docteur	1	5	faible
2.Ecran de création d'un dossier médical	1	9	faible
3.Ecran de demande de rendez-vous avec	1	3	faible
4.Ecran d'enregistrement d'un personnel	1	6	faible
5.Ecran de saisie des informations d'un horaire	1	4	faible
6.Ecran d'enregistrement d'un nouvel utilisateur	1	5	faible
7.Ecran de saisie des information d'authentification	1	2	faible
8.Ecran d'affectation d'un lit	1	3	faible
9.Ecran d'enregistrement d'un nouveau service	1	2	faible
10.Ecran de saisie d'enregistrement d'une nouvelle place	1	4	faible
11.Ecran de création d'une nouvelle facture	1	5	faible
12.Ecran d'enregistrement d'un nouveau département	1	3	faible

La complexité de ces entrées ci-haut énumérées a été déterminée sur référence du tableau 1 de l'annexe A, page 66.

3) Identification des sorties SORT

Tableau 7 : Identification des Sorties

Nom de composant (SORT)	Nbre de GRD	Complexité
1. Impression du dossier Médical	2	Moyen
2. Décharge du patient	3	Moyen
3. Impression du Rapports	4	Elevé
4. Affectation du patient	3	Moyen
5. Impression du facture	2	Faible
6. Impression du billet de sortie	2	Moyen

De même la complexité de ces entrées ci-haut énumérées a été déterminée sur référence sur le tableau 2 de l'annexe A, page 73.

4) Identification des interrogations INT

Il s'agit de l'ensemble des requêtes envers le groupes de données GDI. Chaque entités docteur, patient, personnel, lit, département, facture, place, service, rapport, utilisateur, rendez-vous, horaires, consultation et Réception doivent pouvoir être consultées indépendamment les unes des autres ou en listes croisées. Donc les INT possèdent 14GRD. En se référant au tableau IV.1, on déduit que le niveau de complexité des ENT est **faible** avec un poids égal à 3.

IV.2.4. Calcul des points de fonction brut

Les Points de fonction bruts (PFB) sont calculés en fonction des valeurs de complexité que nous venons d'établir dans les étapes précédentes comme le montre le tableau IV.6.

Tableau 8 : Calcul du nombre de Point de Fonction Brut

Types	Complexité	Nbre de composants	Poids	Nbre de Point de fonction
ENT	Faible Moyen Elevé	12	3	36
SORT	Faible Moyen Elevé	1 4 1	4 5 7	4 20 7
INT	Faible Moyen Elevé	14	3	42
GDI	Faible Moyen Elevé	14	3	42
GDE	Faible Moyen Elevé	0 0 0		0
Nombre total de point de fonction brut (PFB)				**151**

Maintenant que nous avons le PFB, calculons alors le point de fonction ajusté PFA en utilisant l'ensemble de 14 facteurs de complexité auxquels l'on attribue une valeur de zéro à cinq en fonction du Degré d'influence (DI). Signalons que le degré d'influence totale (DIT), permet de corriger le nombre de PFB avec un facteur allant de 65% à 135% pour trouver le nombre de Points de fonction ajustés (PFA). Comme énoncé par Allan J. Albrecht, utilisons les formules suivantes :

Le degré d'influence totale DIT = $\sum_{i=1}^{14} DI$ *;*

*Le facteur d'ajustement FA = 0.65 + 0.01*DIT ;*

Et les points de fonction ajustés PFA = PFB × FA ;

Soient F_1, $F_2, F_3 \dots, F_{14}$ les 14 facteurs d'influences qui sont les Caractéristiques Générales du Système(CGS) voir l'annexe B, Page 74, calculons le DIT de notre application dans le tableau suivant :

FI	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}
DI	2	0	0	3	4	5	5	4	0	1	2	4	0	2

Sa somme totale des DI est donnée par DIT=2+0+0+3+4+5+5+4+0+1+2+4+0+2= **32,**

D'après les formules de Allan J. Albrecht, nous avons DIT=32

FA= 0.65+(0.01*32) = 0.97

D'où PFA= 151*0.97= 146.47

Et pour établir une correspondance entre la taille fonctionnelle et la taille du logiciel, le PFA permet de calculer le nombre d'instructions source (ISL pour instruction source lines) utile pour COCOMO avec la formule suivante :

ISL = 118.7 × PFA – 6490

= (118.7*146.47) - 6490

= 10923

Maintenant que nous avons le nombre de ligne de code de notre application qui sera égale à 10923 donc inférieur à 50000 (d'après le tableau IV.1) ce qui implique que notre projet est de type *Organique* avec les paramètres *a =2.4, b=1.05 et c=0.38.*

Calculons alors la charge, le temps de développement, la taille de l'équipe et la productivité ainsi que les coûts que pourrait prendre le projet. Comme le projet est de type organique, nous avons d'après les formules d'Allan J. Albrecht :

Effort(charge) = 2.4 x $(10923)^{1.05}$ *= 41,732.1031HM = 41HM ;*

TDEV = 2.5 x $(41)^{0.38}$ *= 10.2517=10Mois ;*

Taille de l'équipe= $\frac{41}{10}$*=4.1= 4 personnes ;*

Et la productivité = $\frac{Taille(en\ LOC}{Effort}$ $= \frac{10923}{41}$ *= 266 LOC/HM ;*

Si on considère les jours ouvrables, soit 22 jours par mois, ce projet nécessiterait donc dix mois de travail, soit 220 jours pour une équipe de quatre personnes. Et en se référant à l'ordonnance ministérielle no 570/540/1299 du 02 décembre 2008 portant mesures d'application du décret no 100/322 du 21 novembre 2007 portant fixation des paliers de recrutement et des grilles indicateurs des fonctionnaires en ce qui concerne la transposition, et conformément à l'ordonnance ministérielle no 540/570/2491 du 12 février 2016 portant octroi d'une indemnité d'ajustement des disparités salariales dans le secteur publique, le salaire mensuel brut octroyé à une personne détenant un diplôme de maîtrise est de 374 383 FBu. Avec 22 jours ouvrables par mois, le salaire journalier est de 17017,409090909 Fbu. Dans le cadre de ce projet, avec donc une équipe de 4 personnes, le coût serait : Coût = 17017,409090909 × 4 × 220 = 14 975319,99999992FBU, soit **14 975 320FBU**

IV.2.5. Chronogramme des activités de réalisation du projet

Comme définie par Chemuturi Murali, dans son livre intitule « Software estimation best practices, tools, & techniques : a complete guide for software project estimation » L'estimation logicielle est l'estimation de la taille du logiciel, de l'effort de développement du logiciel, du coût de développement du logiciel et du calendrier de développement du logiciel pour un projet logiciel spécifique dans un environnement spécifique, en utilisant des méthodes, des outils et des techniques définis.

Maintenant que nous avons déjà calculé la taille, l'effort et le cout procédons à l'estimation du calendrier de travail dans le cadre de ce projet. La date de début du projet étant le 1er juin 2022, le calendrier des activités de ce projet est donné dans le diagramme de GANTT ci-après :

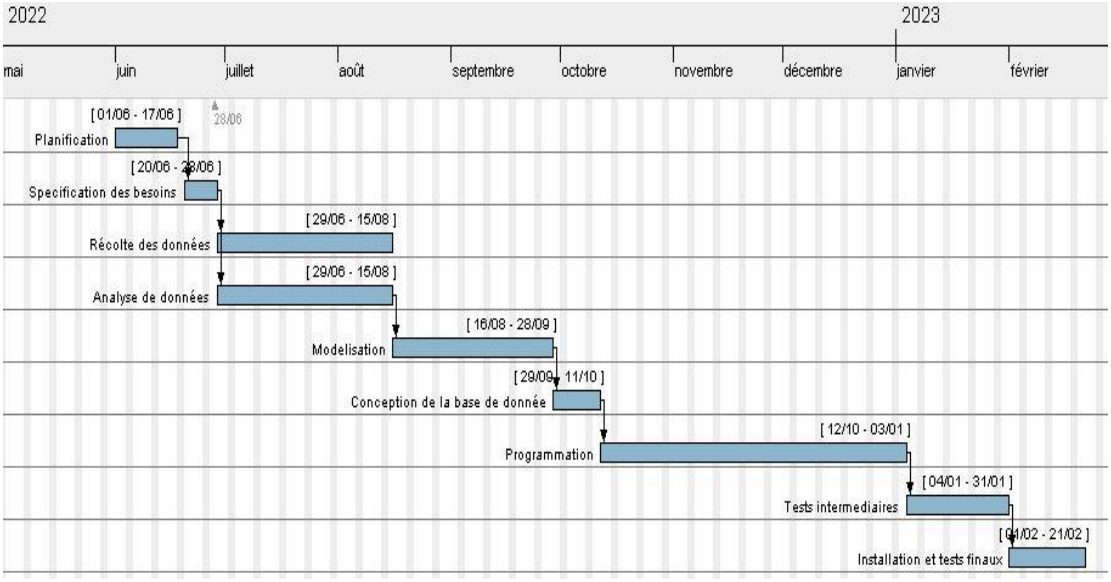


Figure 4 : Diagramme de Gantt du projet

IV.2.6. Modélisation de notre nouveau système de gestion des places d'hospitalisation

Le processus de modélisation vise à obtenir une solution acceptable du système informatique. La solution finalement retenue n'est pas obtenue en une seule itération. Plusieurs étapes sont nécessaires ; ces étapes successives permettent de raffiner le niveau de détails du système à réaliser.

Les premières étapes donnent une vision à très gros grains et permettent d'avancer dans la compréhension du problème. À partir d'une certaine taille et complexité, l'informatisation d'un système nécessite un processus de modélisation. [21]

Quel que soit la taille du problème, une phase de modélisation est une aide au développement du système informatique. Cependant, souvent, le nombre de personnes qui participent à la résolution du problème (clients, équipe de développement, équipe de maintenance) est un des éléments jouant fortement dans la décision de passer par une phase de modélisation. [21]

La pérennité de l'informatisation réalisée est un autre élément justifiant la décision de modéliser le système. En effet, le développement mais aussi la maintenance corrective et la maintenance évolutive du système bénéficient de l'existence du modèle en tant que documentation de référence. Le modèle permet donc de spécifier le système à réaliser/réalisé, de valider le modèle vis-à-vis des clients, de fournir un guide pour la construction du système pour organiser les structures de données et le comportement du système, et de documenter le système et les décisions prises.

IV.2.7. Intérêt de la modélisation

Un modèle est une représentation partielle de la réalité, une abstraction de ce qui est intéressant pour un contexte donné et une vue subjective et simplifiée d'un système.

Elle permet de prédire un fonctionnement ou des événements à venir, de mieux comprendre, d'expliquer et de maîtriser le fonctionnement d'un système (un dessin vaut) mieux que mille discours) et par conséquent de faciliter sa réalisation puis sa maintenance, dans certains cas d'automatiser certaine tâches (génération automatique du squelette du programme à partir d'un modèle), de disposer d'un langage commun entre les différents intervenants et de la maîtrise d'ouvrage **(MOA)** à la maîtrise d'œuvre **(MOE).** [26]

IV.3. Modélisation avec langage UML

UML (*Unified Modeling Language*) est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancé par l'OMG (*Object Management Group*) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets.

Elle est héritée de plusieurs autres méthodes telles que OMT (*Object Modeling Technique*) et OOSE (*Object Oriented Software Engineering*) et Booch. Les principaux auteurs de la notation UML sont Grady Booch, Ivar Jacobson et Jim Rumbaugh. [21]

Elle est utilisée pour spécifier un logiciel et/ou pour concevoir un logiciel. Dans la spécification, le modèle décrit les classes et les cas d'utilisation vus de l'utilisateur final du logiciel.

Le modèle produit par une conception orientée objet est en général une extension du modèle issu de la spécification. Il enrichit ce dernier de classes, dites techniques, qui n'intéressent pas l'utilisateur final du logiciel mais seulement ses concepteurs. Il comprend les modèles des classes, des états et d'interaction. UML est également utilisée dans les phases terminales du développement avec les modèles de réalisation et de déploiement. [21]

UML est une méthode utilisant une représentation graphique. L'usage d'une représentation graphique est un complément excellent à celui de représentions textuelles. En effet, l'une comme l'autre sont ambiguës mais leur utilisation simultanée permet de diminuer les ambiguïtés de chacune d'elles. Un dessin permet bien souvent d'exprimer clairement ce qu'un texte exprime difficilement et un bon commentaire permet d'enrichir une figure. [21]

Il est nécessaire de préciser qu'une méthode telle que UML ne suffit pas à produire un développement de logiciel de qualité à elle seule. En effet, UML n'est qu'un formalisme, ou plutôt un ensemble de formalismes permettant d'appréhender un problème ou un domaine et de le modéliser, ni plus ni moins. Un formalisme n'est qu'un outil. Le succès du développement du logiciel dépend évidemment de la bonne utilisation d'une méthode comme UML mais il dépend surtout de la façon dont on utilise cette méthode à l'intérieur du cycle de développement du logiciel. [21]

IV.4. Caractéristiques des Diagrammes UML

Comme UML est un langage graphique qui permet de modéliser des systèmes sous la forme d'une collection d'objets. Un seul type de modèle n'étant pas suffisant pour décrire correctement un système, UML dispose de plusieurs types de modèles appelés diagrammes (13 en tout), chaque diagramme représentant une vue distincte du système. Il n'est bien sûr pas nécessaire d'utiliser tous les diagrammes pour modéliser un système, nous allons choisir parmi tous les diagrammes disponibles ceux qui sont les plus pertinents et les mieux adaptés à la modélisation de notre système [22].

Un diagramme UML est une représentation graphique qui s'intéresse à un aspect précis du modèle. Chaque type de digramme UML possède une structure (les types des éléments de modélisation qui le compose sont prédéfinis). Un type de diagramme UML offre toujours la même vue d'un système (il véhicule une sémantique précise). Combines, les différents types de diagrammes ML offrent une vue complète des aspects statistiques et dynamiques d'un système. [26]

IV.5. Types de Diagrammes UML

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles à savoir:

Les diagrammes structurels ou statiques *(structure diagrams)* et Les diagrammes comportementaux ou dynamiques *(behavior diagrams).* [22]

Ils sont donnés dans la figure ci-dessous :

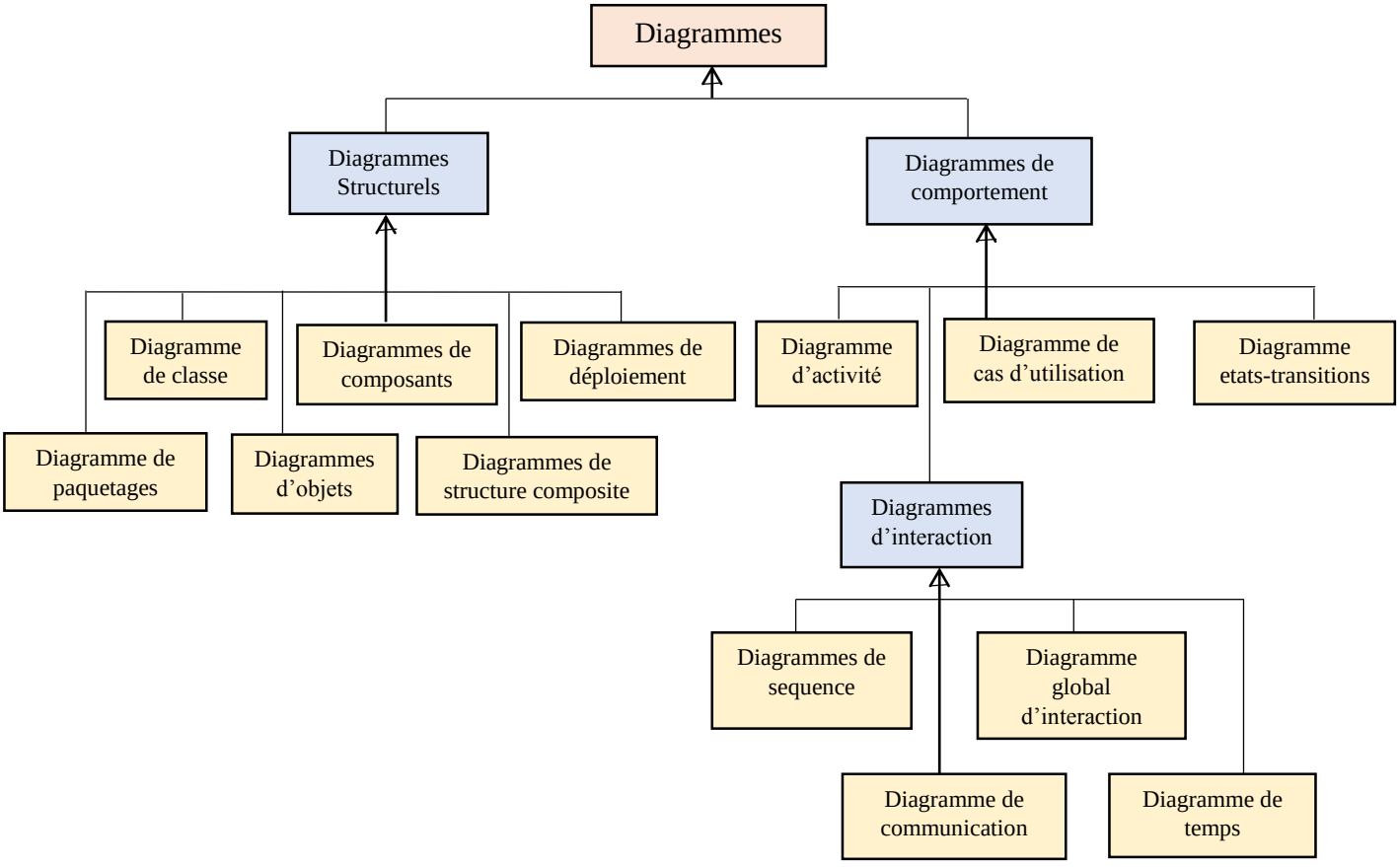


Figure 5 : Treize diagrammes que propose UML 2

IV.6. Choix de diagrammes de modélisation du nouveau système

Cette section donne une description détaillée de quelques diagrammes utilisés pour la modélisation de notre système de gestion des places d'hospitalisation ainsi que leurs rôles. Il s'agit de diagrammes de cas d'utilisation, diagrammes de séquence et le diagramme de classe.

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (Human ou Machine) et un system. Il est une entité significative de travail. Sa représentation met en jeu trois concepts dont l'acteur, l'interaction entre l'acteur et le cas d'utilisation [23].

Un diagramme de séquence est la représentation graphique de la partie comportementale (interaction) d'un modèle UML. Toute les informations (objets, messages, etc.) sont contenues dans le modèle et représentées graphiquement à l'aide des diagrammes. Il est donc possible de représenter une même information dans différents diagrammes. Ainsi *Le diagramme de classe* permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. [23]

Tout système peut être décrit par un certain nombre de cas d'utilisation correspondant aux besoins exprimés par l'ensemble des utilisateurs. À chaque utilisateur, vu comme acteur, correspondra un certain nombre de cas d'utilisation du système. L'ensemble de ces cas d'utilisation se représente sous forme d'un diagramme. [23] Ainsi, procédons à la représentation globale du diagramme de notre système.

Tableau 9 : Identification des acteurs et des cas d'utilisation

Acteurs	**Cas d'utilisation**
Tous les acteurs	- Authentification
Administrateur du système	- Gérer tous les utilisateurs du système.
Agents d'accueil	- Gérer les fiches des patients (ouverture du dossier médicale), - Attribuer un numéro d'ordre; - Gérer les billets de sorties (patients hospitalisés)
Docteur	- Gérer les consultations ; - Gérer les admissions (hospitalisations) ; - Gérer les rendez-vous.
Personnel Administratif	- Gérer les places (capacité d'accueil) ; - Gérer les rapports ; - Gérer les paiements.
Patient	- Consulter son dossier médical ; - Télécharger le dossier Médical ; - Consulter son historique d'admission ; - Demander un rendez-vous.

Maintenant que nous venons d'identifier les différents acteurs et leurs cas d'utilisation, procédons à la construction du diagramme de cas d'utilisation du nouveau système de gestion des places d'hospitalisation. A ce niveau, nous présentons une vue d'ensemble du fonctionnement de notre système en reliant les acteurs à leurs cas d'utilisation comme le montre la figure IV.4.

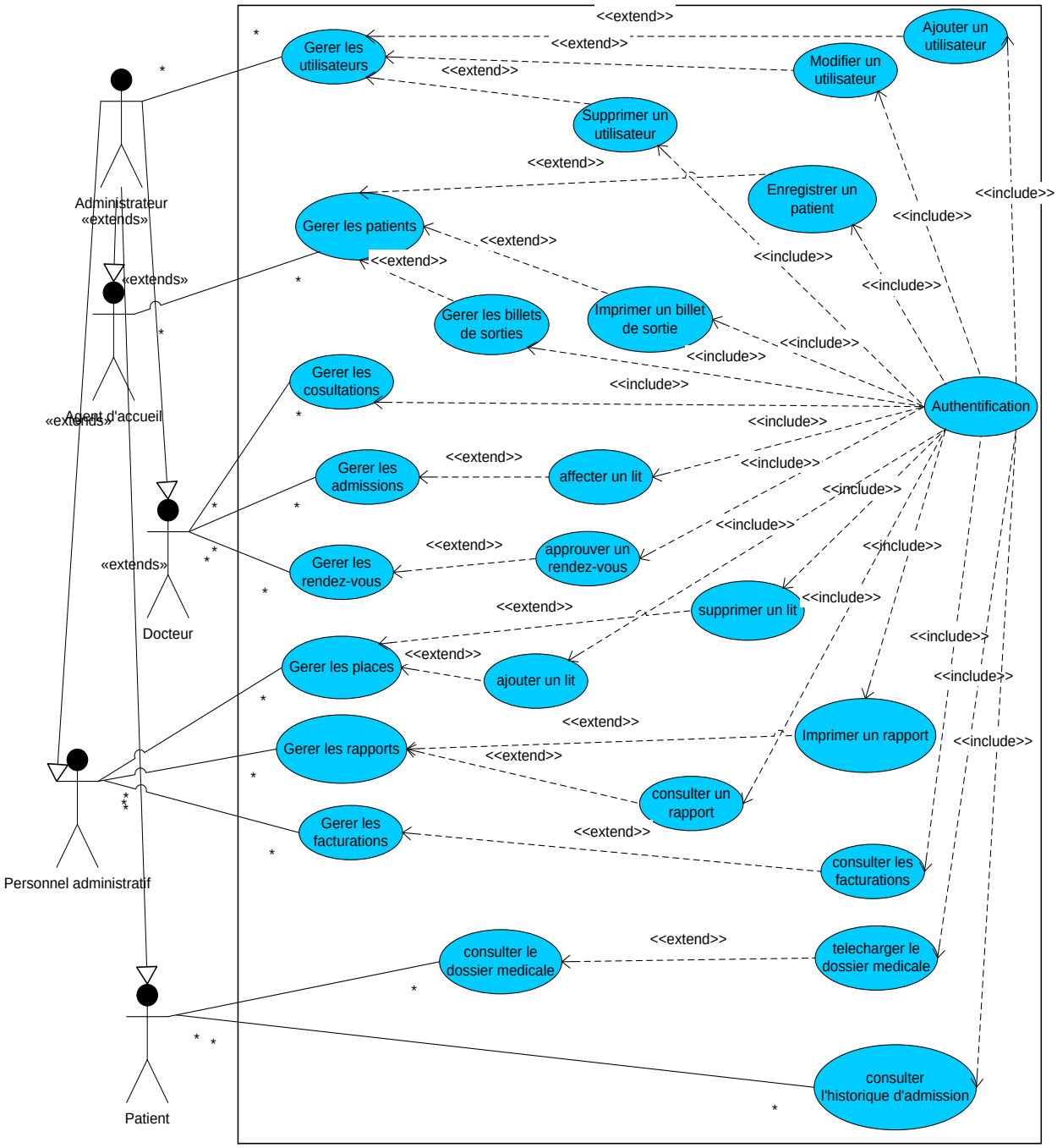


Figure 6 : Diagramme de cas d'utilisation

Nous allons maintenant procéder à la description de chaque cas d'utilisation en précisant un objectif du cas d'utilisation et les acteurs intervenants ainsi que les scénarios nominaux.

1. Cas d'utilisation « S'authentifier »

Nom du cas d'utilisation : S'authentifier

Objectif : Permettre aux utilisateurs de s'authentifier avant d'accéder au système.

Tableau 10 : Description détaillés du cas d'utilisation « s'authentifier »

Sommaire	
Nom du Cas d'utilisation	**S'authentifier**
Acteur	- Tous les utilisateurs
Précondition	- Le système est en cours d'exécution ; - Apparition de la page d'authentification; - Avoir ses paramètres de connexion.
Post-condition	- L'utilisateur est authentifié ;
Scenario Principale	- Le système demande d'abord à l'utilisateur de choisir le type de compte avec lequel il veut se connecter ; - Le système demande à l'utilisateur de saisir ses paramètres de connexion (email et le mot de passe) - L'utilisateur saisit son email et son mot de passe et appuie sur le bouton <connexion> ou sur la touche entrée du clavier
Scenario alternatif	- L'utilisateur a mal saisi ses identifiants ou l'utilisateur s'est connecté avec le compte qu'il n'en a pas le droit; - Le système lui renvoie un message d'échec de connexion lui demandant de vérifier ses paramètres de connexion; - L'utilisateur vérifie son email et son mot de passe puis recommence.

2. Cas d'utilisation « Créer un dossiers médical »

Nom du cas d'utilisation : Créer un dossier médical

Résumé : Ce cas d'utilisation permet d'ouvrir un nouveau dossier du patient au centre de soins.

Tableau 11: Description détaillés du cas d'utilisation « Créer un dossier Médical »

Sommaire	
Acteurs concernés	- Agents d'accueil
Précondition	- L'utilisateur doit être authentifié et autorisé
Post-condition	- Le dossier du nouveau patient est créé avec succès.
Scénario principale	- L'agent demande la page de gestion des patients dans la barre de navigation de sa page d'accueil ; - Le système lui retourne automatiquement la liste des patients déjà ajoutés; - L'agent sélectionne dans l'entête de la table <ajouter un patient> pour créer son dossier ; - Le système lui renvoie un formulaire de création; - L'agent remplit les champs du formulaire ; - Le système vérifie les informations saisies, enregistre les données et affiche un message de confirmation.
Scénario Alternatif	- Les informations saisies existent déjà: le système signale l'agent que le patient existe dans le système - Tous les champs ne sont pas remplis, formulaire vide : un message d'erreur sera affiché.

3. Cas d'utilisation « Affecter un lit »

Nom du cas d'utilisation : Affecter un lit

Résumé : Ce cas d'utilisation permet l'affectation de places d'hospitalisation aux patients qui en ont besoins.

Tableau 12 : Description détaillés du cas d'utilisation « Affecter un lit »

Sommaire	
Acteurs concernés	- Docteur
Précondition	- Docteur authentifié et autorisé
Post-condition	- Allocation effectuée avec succès
Scénario principale	- Le docteur demande la page d'allocation des lits; - Le système lui retourne premièrement les allocations déjà faites; - Le docteur sélectionne dans l'entête de la table <affecter un lit>; - Le système lui renvoie un formulaire de sélection du nom du patient et du type de lit; - Le docteur recherche le nom du patient, le type de lit et son emplacement; - Le système vérifie les informations saisies, enregistre les données et met à jour la liste des allocations déjà faites.
Scénario Alternatif	- Le lit choisi est déjà alloué par un autre patient, - Le nom du patient et/ou le type de lit n'est pas sélectionné : un message d'erreur indiquant que le formulaire est vide est affiche.

4. Cas d'utilisation « Gérer les consultations »

Nom du cas d'utilisation : Gérer la consultation

Résumé : Ce cas d'utilisation permet au docteur de pouvoir ajouter un patient (lui consulter)

Tableau 13 : Description détaillés du cas d'utilisation « Gérer les consultations »

Sommaire	
Acteurs concernés	- Docteur
Précondition	- Le docteur doit être authentifié et autorisé
Post-condition	- Le patient consulté et ajouté
Scénario principale	- Le docteur demande la page de consultation dans son tableau de bord (Dashboard); - Le système lui retourne automatiquement la liste des patients déjà consultés; - Le docteur sélectionne dans l'entête de la table <consulter un patient>; - Le système lui renvoie un formulaire de consultation; - L'agent remplit les champs du formulaire ; - Le système vérifie les informations saisies, enregistre les données et affiche un message de confirmation.
Scénario Alternatif	- Le patient existe déjà dans les consultations antérieures: le système affiche un message d'erreur, - Tous les champs ne sont pas remplis, formulaire vide : un message d'erreur sera affiché.

5. Cas d'utilisation « Demander un rendez-vous »

Nom du cas d'utilisation : Demandez un rendez-vous

Résumé : Ce cas d'utilisation permet aux patients d'envoyer un message de demande de rendez-vous.

Tableau 14 : Description détaillés du cas d'utilisation « demander un rendez-vous »

Sommaire	
Acteurs concernés	- Patient
Précondition	- Le patient doit être authentifié et autorisé
Post-condition	- La demande envoyée.
Scénario principale	- Le patient demande page d'accueil ; - Le système lui retourne automatiquement page; - Le patient sélectionne dans la barre de navigation <rendez-vous> - Le système affiche la liste des réponses aux rendez-vous envoyés; - Le patient clique sur <Envoyer une demande>; - Le système lui retourne une zone de texte pour saisir le message ; - Le patient écrit le message et sélectionne le destinataire (docteur) et clique sur <envoyer> - Le système enregistre les données et affiche un message de confirmation de l'envoie.
Scénario Alternatif	- La zone de texte ou le destinataire est vide, le système affiche un message d'erreur indiquant que tous les champs doivent être obligatoirement remplis.

IV.7. Diagrammes de séquence pour les cas d'utilisation pertinents

L'objectif du **diagramme de séquence** est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés. [23]

Ligne de vie: Une **ligne de vie** représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour. [23]

Message synchrone et asynchrone : Dans un diagramme de séquence, deux types de messages peuvent être distingués : **Message synchrone** – Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message [23].

Et les **Message asynchrone** – Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message [23]. UML propose aussi des messages de **création** et de **suppression** (ou de destruction) d'objets afin de gérer le cycle de vie des objets participant à une interaction. Dans une interaction UML, les objets peuvent soit exister au début de l'interaction, soit être créés par d'autres objets pendant l'interaction. Il est aussi possible de spécifier des suppressions d'objets. Celles-ci sont initiées par des objets participant à l'interaction. Un objet détruit ne peut plus recevoir de message. [24]

Tous les utilisateurs du système doivent s'authentifier pour accéder à l'application, le système envoie le formulaire d'authentification, l'utilisateur saisit les paramètres nécessaires et les valide. Le système effectue une vérification et affiche la page réservée à l'utilisateur authentifié ou le message d'erreur si les informations fournies ne correspondent pas à celles qui existent dans la base de données.

Le cas d'utilisation <s'authentifier> de notre nouveau système est représenté dans la figure 7.

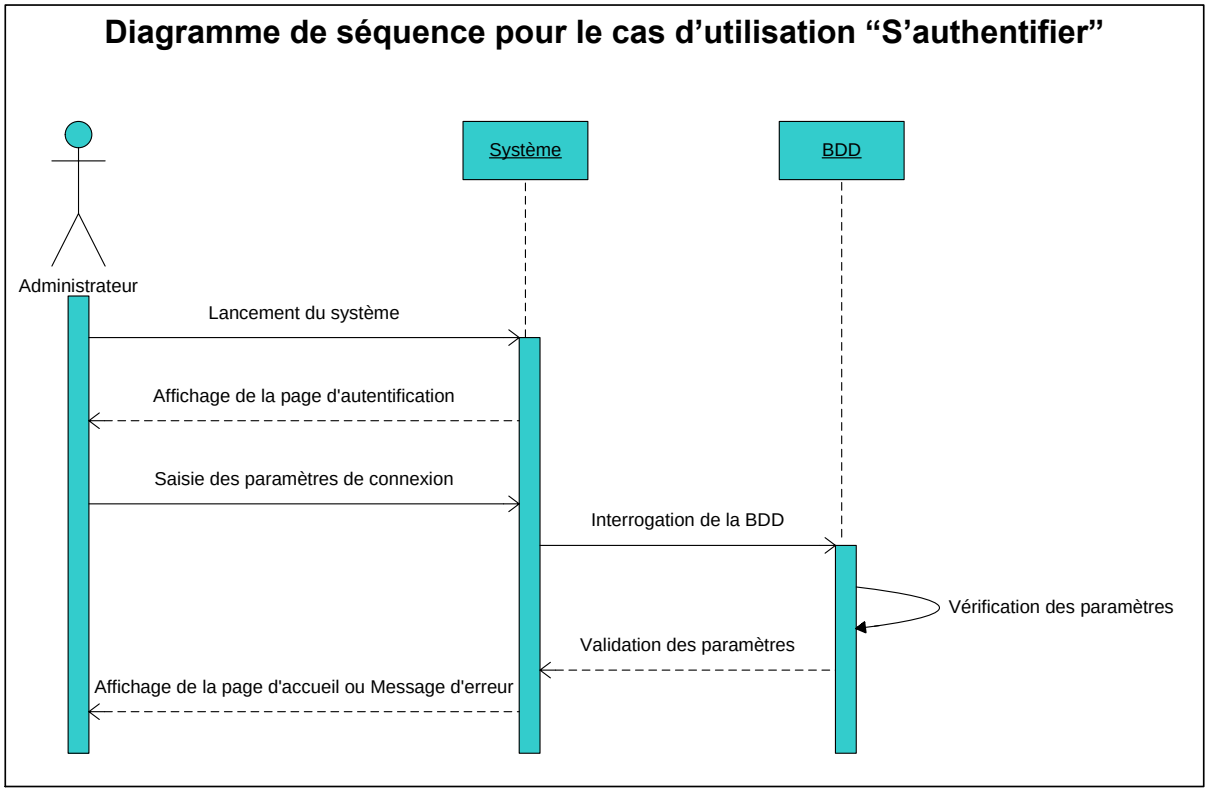


Figure 7 : Diagramme de séquence pour le cas d'utilisation « S'authentifier »

Après l'authentification, l'agent d'accueil peut sélectionner le menu <créer un dossier du dossier médical>. La page contenant la création du dossier du patient apparait, il saisit toutes les informations du patient, puis clique sur le bouton valider et le système vérifie les informations saisies par l'agent et si elles sont valides un message de confirmation de la création du dossier est affiché. Ce cas d'utilisation est illustré dans la figure 8.

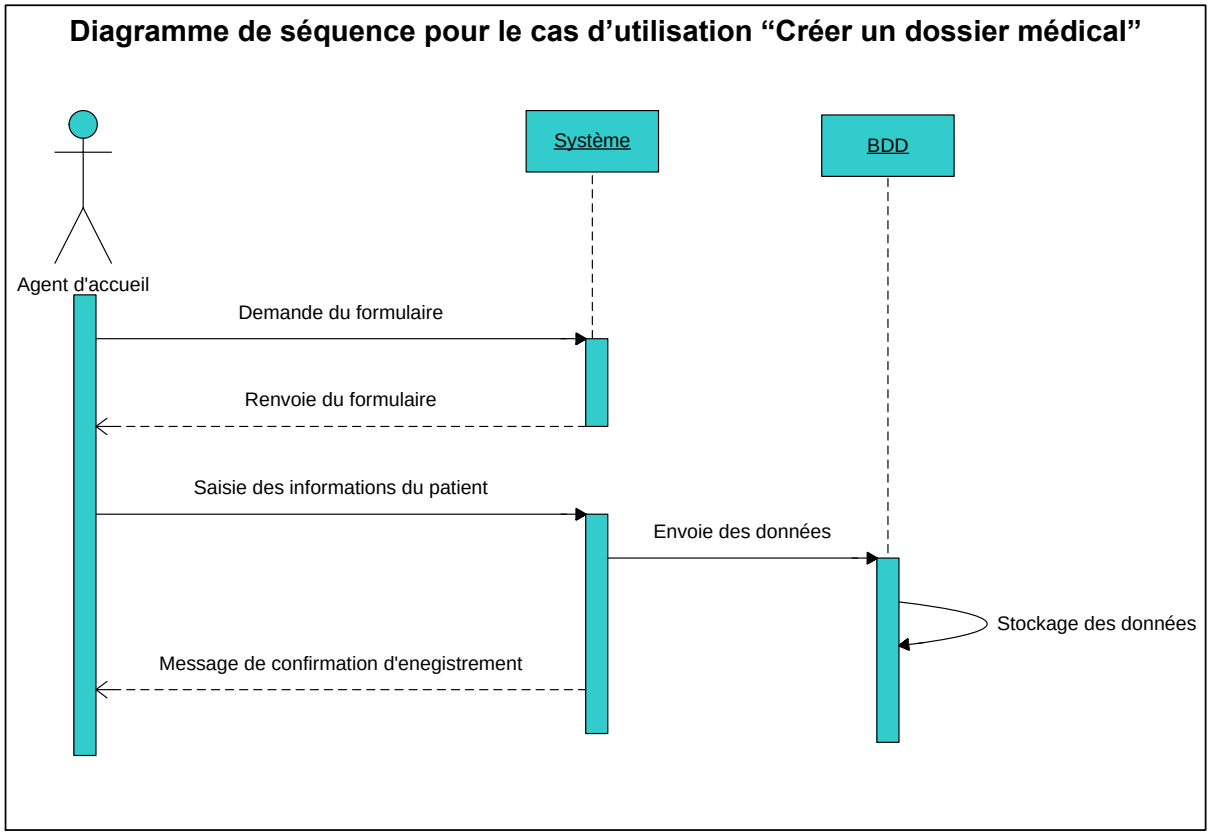


Figure 8 : Diagramme de séquence pour le cas d'utilisation « créer un dossier médical »

De même, après authentification, un patient peut avoir besoin d'un rendez-vous envers un docteur dans l'un des départements du centre, il ouvre la page de demande, le système lui retourne un formulaire où il choisit le département, le système lui affiche automatiquement la liste des docteurs de ce département avec leurs spécialités. Il complète le formulaire en choisissant le docteur souhaité et envoie le message, le système lui retourne un message de confirmation de l'envoie.

Le diagramme de séquence pour le cas d'utilisation <demander un rendez-vous> est alors représenté dans la figure 9.

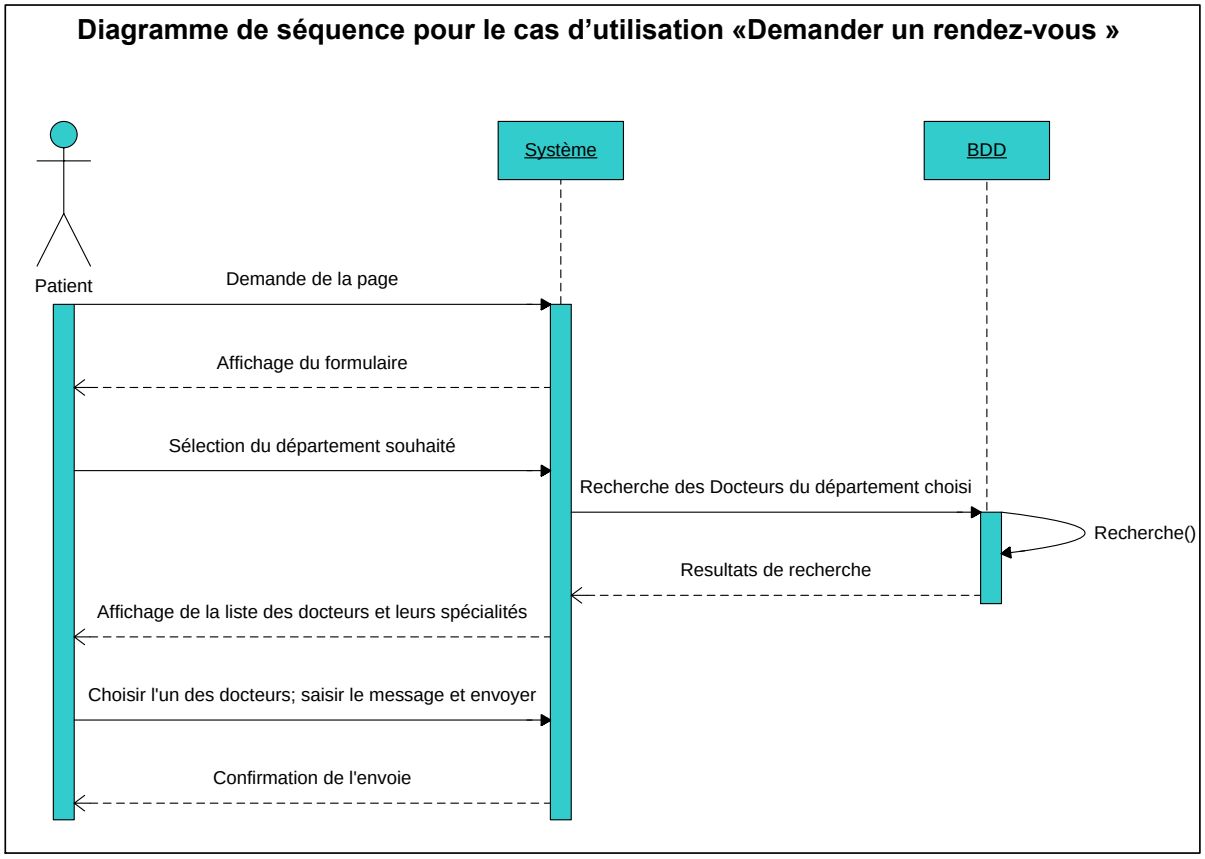


Figure 9 : Diagramme de séquence pour le cas d'utilisation « demander rendez-vous »

Ainsi un docteur, après son authentification peut demander le formulaire pour hospitaliser un patient. Le système renvoie le formulaire, le docteur vérifié s'il y a une place disponible. S'il y en a, il saisit l'identification du patient pour la sélectionner et valide. Le système retourne un message de confirmation de l'affectation.

Le diagramme de séquence pour « Affecter un lit » est donné par la figure 10.

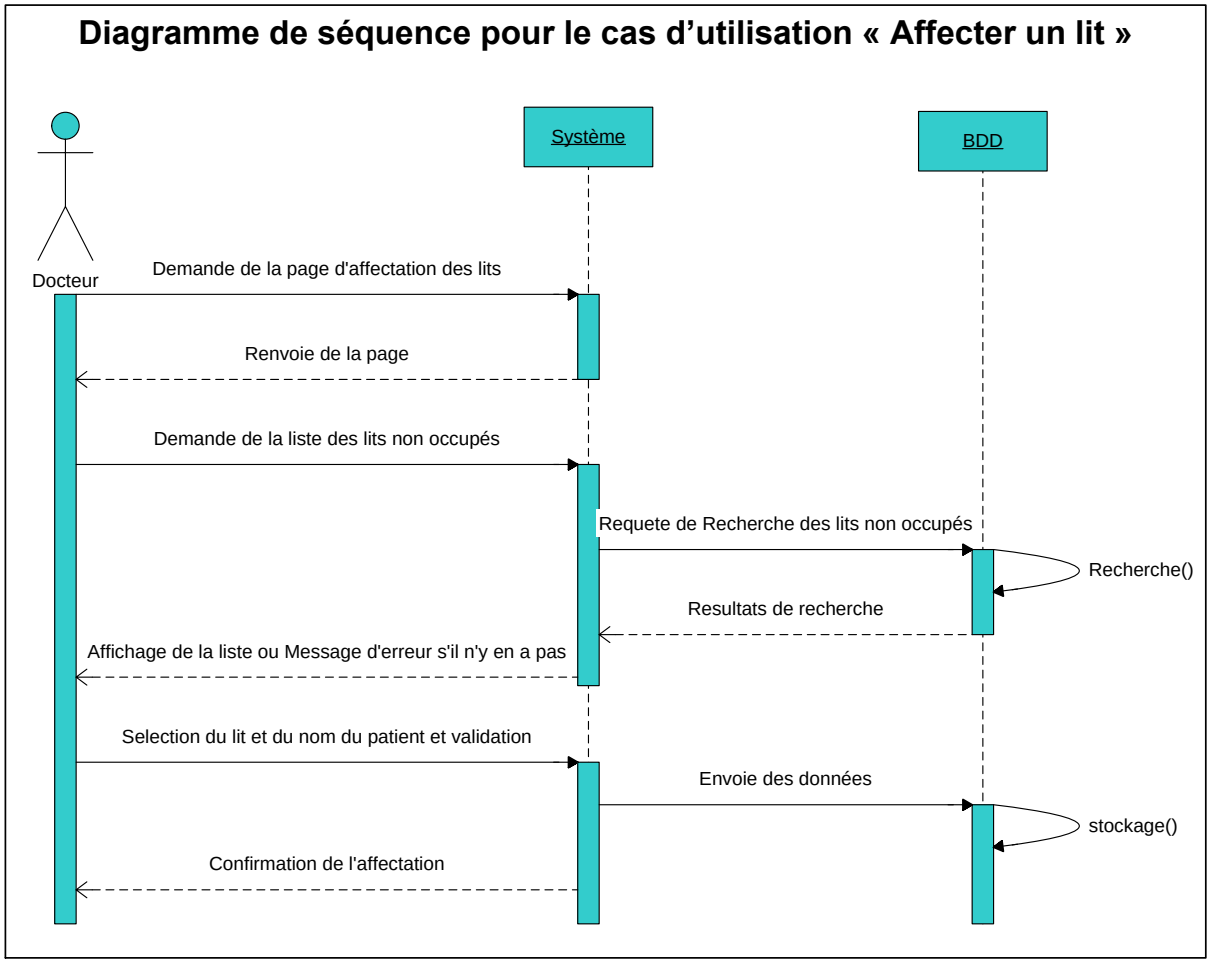


Figure 10 : Diagramme de séquence pour le cas d'utilisation « Affecter un lit »

IV.8. Diagrammes de Classes

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec *UML. En* effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe ; seul l'algorithme général et le pseudo-code correspondant peuvent être associés à la modélisation. [23]

La description du diagramme de classe est fondée sur :

1. Le concept d'objet,
2. Le concept de classe comprenant les attributs et les opérations,
3. Les différents types d'association entre classes. [23]

Un **objet** est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés. Il est caractérisé par les valeurs de ses propriétés qui lui confèrent des états significatifs suivant les instants considérés [24].

Une **classe** décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition). Un objet est une **instance** d'une classe. La classe représente l'abstraction de ses objets. Au niveau de l'implémentation, c'est-à-dire au cours de l'exécution d'un programme, l'identificateur d'un objet correspond une adresse mémoire [25]. Un **attribut** est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur (sauf cas d'attributs multivalués) [24]. Une **opération** est une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet. Une **méthode** est l'implémentation d'une opération [25].

La figure 11 nous donne la représentation du diagramme de classe de notre système de gestion des places d'hospitalisation.

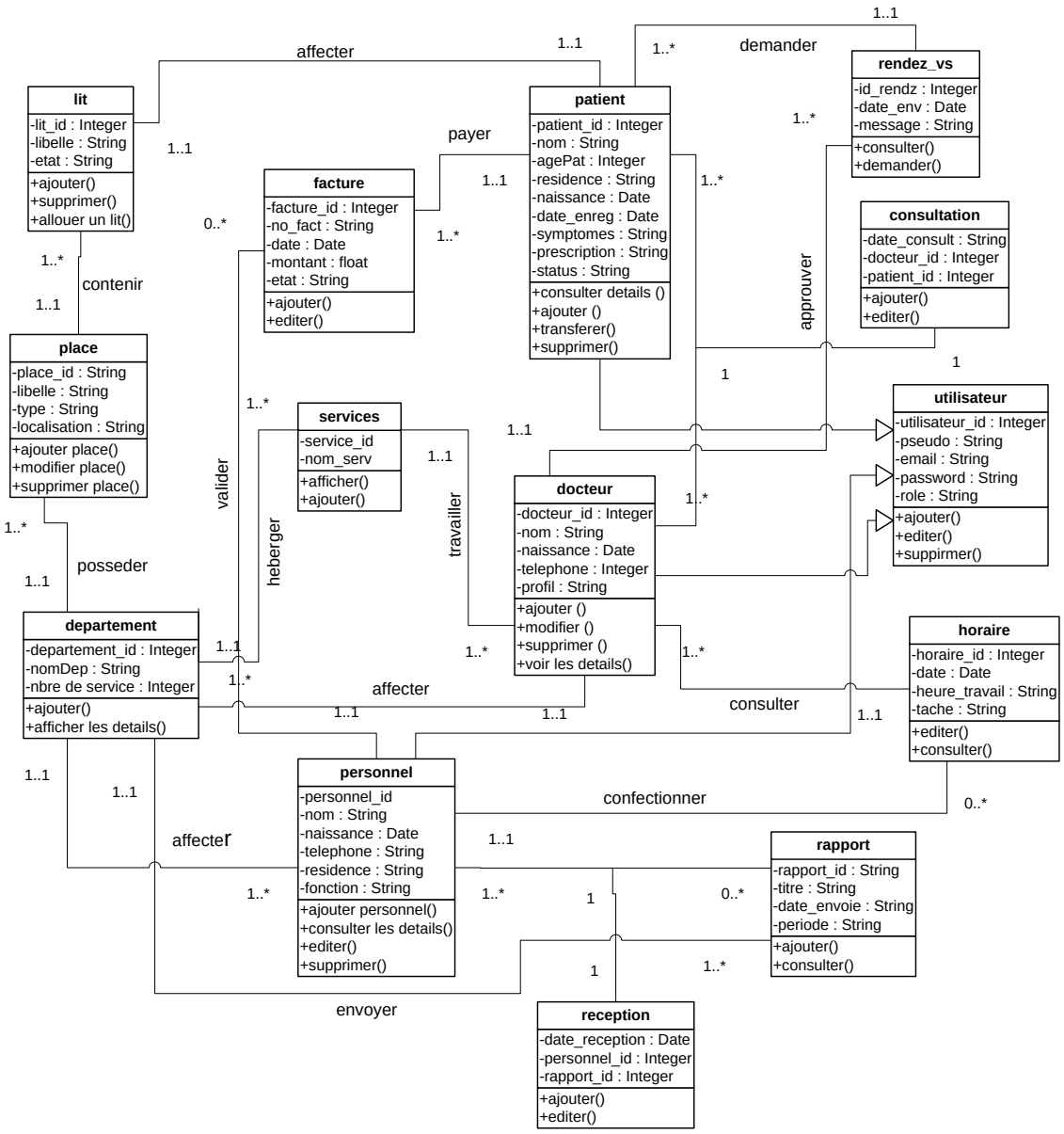


Figure 11 : Diagramme de classe du nouveau système

La modélisation avec un outil de modélisation comme UML est l'une des étapes primordiales du développement de l'application. Au cours de cette partie nous avons défini les concepts de bases d'UML et nous avons présenté la façon dont s'articulent toutes les parties du modèle UML d'une application lors de la réalisation d'un développement avec UML. Sur base de diagrammes d'UML qui sont expliqués dans ce chapitre, nous avons utilisé quelques-uns pour donner la structure de notre nouvelle application de gestion des places d'hospitalisation qui sera développée dans la suite. Il s'agit du diagramme de classe, diagramme de cas d'utilisation et le diagramme de séquence.

CHAPITRE V. DEVELOPPEMENT ET IMPLEMENTATION DU NOUVEAU SYSTEME DE GESTION DES SERVICES D'ACCUEIL

V.1. Introduction

Maintenant que nous avons terminé toutes les étapes nécessaires de modélisation et de conception, nous pouvons procéder à l'étape de développement de notre application en écrivant le code sur bases des résultats d'analyse obtenus dans les chapitres précédents. Dans cette partie, nous allons présenter quelques outils d'implémentation en comparant les uns aux autres et nous allons choisir l'un d'entre eux pour l'implémentation de notre système de gestion des places d'hospitalisation.

De même, le développement d'une application fait référence à un ensemble des modèles, c'est ainsi que nous allons présenter dans ce chapitre les différents modèles de développement des applications et choisir l'un des modèles à adopter pour notre application. Aussi cette partie consiste à la présentation des résultats obtenus (les principales fonctionnalités de notre application) et les aspects de sécurité de notre système ainsi qu'une petite estimation du coût de ce projet.

V.2. Modèles de cycle de vie d'un logiciel

Le développement de logiciel est une activité complexe, qui est loin de se réduire à la programmation. Le développement de logiciels nécessite d'adopter une méthode de développement, qui permet d'assister une ou plusieurs étapes du cycle de vie de logiciel. Parmi les méthodes de développement, les approches objet, issues de la programmation objet, sont basées sur une modélisation du domaine d'application. Cette modélisation a pour but de faciliter la communication avec les utilisateurs, de réduire la complexité en proposant des vues à différents niveaux d'abstraction, de guider la construction du logiciel et de documenter les différents choix de conception.

Le cycle de vie d'un logiciel désigne alors un ensemble structuré d'activités nécessaires pour développer un logiciel. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

V.2.1 Etapes du cycle de vie du logiciel

Le développement d'un logiciel impose d'effectuer un certain nombre d'étapes. Il ʻagit de : *Analyse et spécification des besoins, Spécification du système, Conception de l'architecture, Conception détaillée, Mise en œuvre, Validation et Evolution et maintenance.*

L'étape d'analyse et définition des besoins consiste à d´exterminer les attentes des futurs utilisateurs, par exemple avec un cahier des charges. Il faut décrire à la fois le système et l'environnement dans lequel le système sera exécuté. Cette étape comprend également une étude de faisabilité de la part des experts. C'est sur la base de cette étape que le contrat est signé.

L'analyse et conception consiste à analyser, spécifier et effectuer les choix de conception du système. Cette étape comporte plusieurs sous-étapes. Quant à la spécification du système, c'est une description des fonctionnalités. Il s'agit de décrire ce que le système doit faire, sans préciser comment ces fonctionnalités seront implémentées.

La conception de l'architecture consiste à décrire la structure générale du système et la conception détaillée du système consiste en une description des composants, des algorithmes et des structures de données. L'étape de mise en œuvre consiste à programmer le logiciel, en suivant les choix effectués lors de l'analyse et la conception.

La validation consiste à s'assurer que le programme est de qualité. Il existe plusieurs techniques de validation : Analyse statique (typage, conventions de programmation, détection d'erreurs pouvant survenir à l'exécution), preuve formelle (couteuse, peu utilisée), revue de code (efficace), tests qui constituent la principale méthode de validation. On distingue les tests unitaires, les tests d'intégrations, les tests systèmes et les tests d'acceptation.

L'étape d'évolution et de maintenance consiste à effectuer des modifications du logiciel après sa livraison. On distingue plusieurs types de maintenance : Maintenance corrective (correction de défauts), maintenance évolutive (ajout de nouvelles fonctionnalités), maintenance adaptative (portage sur une nouvelle plate-forme). [26]

V.2.2 Différents Modèles de cycle de vie du Logiciel

Comme nous avons déjà expliqué les différentes étapes de développement du logiciel, cette section nous permet de représenter ces étapes sous forme de processus dans des modèles. Leurs avantages et inconvénients ainsi que leur complexité marquent leur différence dans le développement. Bien qu'il en existe d'autres, dans cette partie nous allons décrire le modèle en cascade, le modèle en V et le modèle en spirale.

Dans le *modèle en cascade*, on effectue les différentes étapes du logiciel de façon séquentielle. Les interactions ont lieu uniquement entre étapes successives : on s'autorise des retours en arrière uniquement sur l'étape précédente. Il est simple et logique, facilite la planification des étapes et des délais et adapté à des petits systèmes [27].

Le *modèle en* V du cycle de vie du logiciel précise la conception des tests. Il est facile à comprendre, il segmente clairement le projet et effectue des vérifications adaptées à chaque étape. Par conséquent il Manque de souplesse [27].

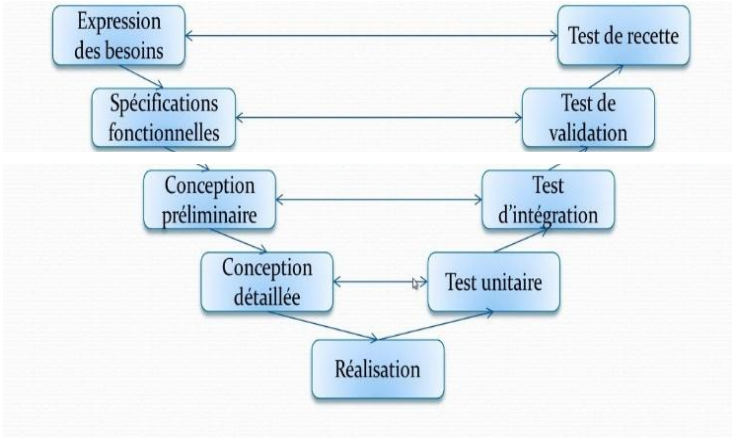


Figure 12 : Modèle de cycle de vie en V

Et quant au modèle en spirale, Chaque cycle est composé de 4 étapes : Définition des objectifs alternatives contraintes, identification des risques et solutions pour les réduire, développement et vérification et planification des cycles suivants.

V.3. Outils d'implémentation du nouveau système de gestion des services d'accueil

Pour parvenir à une application fonctionnelle, un ensemble d'outils doivent être mis en jeu. Il s'agit principalement du langage de programmation, d'un système de gestion des bases de données (SGBD), d'un serveur de données, d'un éditeur ainsi que les technologies qui en découlent.

Notre système de gestion des bases de données est une application web, nous avons jugé bon d'utiliser PHP comme langage de programmation coté serveur, HTML et CSS pour le design et la programmation coté client, MySQL comme système de gestion des bases de données, Apache comme serveur Web et Visual Studio Code comme éditeur.

Du point de vue technologie, nous avons utilisé la programmation Orienté-objet dans l'architecture MVC (Modèle-Vues-Contrôleurs) du Framework CodeIgniter du langage PHP. Dans cette partie alors, nous allons donner un bref aperçu sur ces différents outils et technologies de programmation.

V.3.1. Bases de données

Lors du développement de notre système la première des choses a été la création de la base de données qui servira notre application. Une base de données est donc une collection de données stockées dans des fichiers et accessibles à la demande pour plusieurs utilisateurs et des besoins divers. [30]

Les bases de données (BD) sont nées à la fin des années 1960 pour combler les lacunes des systèmes de fichiers et faciliter la gestion qualitative et quantitative des données informatiques. [30]

Les BD sont gérées par les Systèmes de gestion de bases de données (comme Oracle PostgreSQL ou MySQL par exemple). Un SGBD est un logiciel système qui permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données. Il permet d'effectuer des comptes rendus des informations enregistrées et comporte des mécanismes pour assurer la cohérence des informations, éviter des pertes d'informations dues à des pannes, assurer la confidentialité et permettre son utilisation par d'autres logiciels1. Selon le modèle, le SGBD peut comporter une simple interface graphique jusqu'à des langages de programmation sophistiqués. [31]

V.3.2. Système de gestion de base de données MySQL

MySQL est un Système de Gestion de Base de Données (SGBD) parmi les plus populaires au monde. Il est distribué sous double licence, un licence publique générale GNU et une propriétaire selon l'utilisation qui en est faites. La première version de MySQL est apparue en 1995 et l'outil est régulièrement entretenu. [32]

Ce système est particulièrement connu par des développeurs pour faire partie des célèbres quatuors: WAMP (Windows, Apache, MySQL et PHP), LAMP (Linux) et MAMP (Mac). Ces packages sont si populaires et simples à mettre en œuvre que MySQL est largement connu et exploité comme système de gestion de base de données pour des applications utilisant PHP. C'est d'ailleurs pour cette raison que la plupart des hébergeurs web proposent PHP et MySQL. [32]

L'une des spécificités de MySQL c'est qu'il inclut plusieurs moteurs de bases de données et qu'il est par ailleurs possibles au sein d'une même base de définir un moteur différent pour les tables qui composent la base. Cette technique est astucieuse et permet de mieux optimiser les performances d'une application. Les 2 moteurs les plus connus étant MyISAM (moteur par défaut) et InnoDB. La réplication est possible avec MySQL et permet ainsi de répartir la charge sur plusieurs machines, d'optimiser les performances ou d'effectuer facilement des sauvegardes des données. [32]

V.3.3. HyperText Processor PHP

PHP (officiellement, ce sigle est un acronyme récursif pour PHP: HyperText Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web. Il peut être intégré facilement au HTML. [33]

Le langage PHP a été mis au point au début d'automne 1994 par Rasmus Lerdorf. Ce langage de script lui permettait de conserver la trace des utilisateurs venant consulter son CV sur son site,

grâce à l'accès à une base de données par l'intermédiaire des requêtes SQL. Ainsi, étant donné que de nombreux internautes lui demandèrent ce programme, Rasmus Lerdorf mit en ligne en 1995 la première version de ce programme qu'il baptisa Personal Sommaire Page Tools, puis Hypertext Preprocessor V1.0 (Traduisez page personnelle version 1.0). Etant donné le succès de PHP1.0, Rasmus Lerdorf décida d'améliorer ce langage en y intégrant des structures plus avancées telles que les boucles, des structures conditionnelles, et y intégra un package permettent d'interpréter les formulaires qu'il avait développé (FI, Form Interpreter) ainsi que le support de MySQL. C'est de cette façon que la version 2du langage, 72 baptisée pour l'occasion PHP/FI version2, vit le jour durant l'été 1995. Il fut rapidement utilisé sur de nombreux sites (15000 fin 1996, puis 50000 en milieu d'année 1997). A partir de 1997, Zeev Suraski et Andi Gurmans rejoignirent Rasmus pour former une équipe de programmeurs afin de mettre au point PHP3 (Stig Bakken, Shane Caraveo et Jim Winstead les rejoignirent par la suite). C'est ainsi que la version3.0 de PHP fut disponible le 6 juin 1998. A la fin de l'année 1999, une version de PHP, baptisé PHP4, est apparue. [34]

Ce qui distingue PHP des langages de script comme le Javascript est que le code est exécuté sur le serveur. Si vous avez un script similaire sur votre serveur, le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Vous pouvez configurer votre serveur web afin qu'il analyse tous vos fichiers HTML comme des fichiers PHP. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques. [33]

Le grand avantage de PHP est qu'il est extrêmement simple pour les néophytes, mais offre des fonctionnalités avancées pour les experts. Ne craignez pas de lire la longue liste de fonctionnalités PHP. Vous pouvez vous plonger dans le code, et en quelques instants, écrire des scripts simples. [33]

Le principe de fonctionnement de l'interpréteur de code PHP est illustré dans la figure 13.

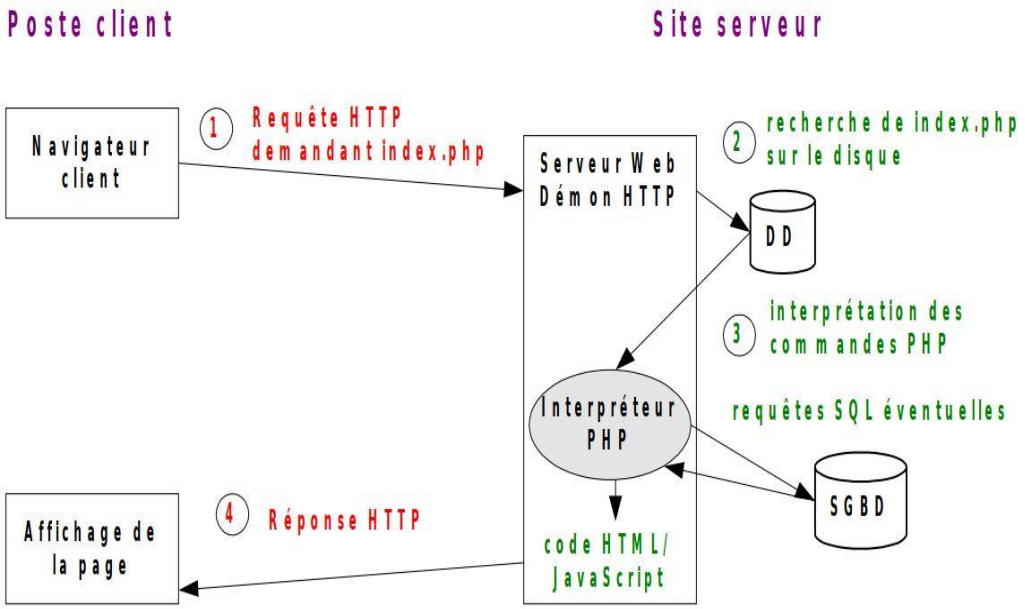


Figure 13 : Principe de fonctionnement de l'interpréteur de code PHP

V.3.4. Framework

En programmation informatique, un *framework* (appelé aussi infrastructure logicielle, infrastructure de développement, environnement de développement, socle d'applications, cadre d'applications ou cadriciel) est un ensemble cohérent de composants logiciels structurels qui sert à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel, c'est-à-dire une architecture.

Un *framework* se distingue d'une simple bibliothèque logicielle principalement, d'une part par son caractère générique, faiblement spécialisé, contrairement à certaines bibliothèques ; un *framework* peut à ce titre être constitué de plusieurs bibliothèques, chacune spécialisée dans un domaine. Un *framework* peut néanmoins être spécialisé dans un langage particulier, une plateforme spécifique, un domaine particulier : communication de données, data mapping, etc.. D'autre part, il impose un cadre de travail, dû à sa construction même, guidant l'architecture logicielle voire conduisant le développeur à respecter certains patrons de conception ; les bibliothèques le constituant sont alors organisées selon le même paradigme. [35]

Le langage PHP comme d'autres langages de programmation possède plusieurs framework comme CodeIgniter, Laravel, Symphonie, Pour développer notre application nous avons utilisé le framework CodeIgniter. **CodeIgniter** est alors un environnement cadre de développement d'application, un ensemble d'outils permettant de structurer et de construire des sites Web en utilisant PHP.

Son objectif est de vous permettre de développer des projets beaucoup plus rapidement que si vous partiez de zéro, en fournissant un ensemble fourni de bibliothèques pour les tâches habituellement nécessaires, ainsi que d'une interface simple et une structuration logique d'accès à ces bibliothèques. CodeIgniter vous permet de vous concentrer sur votre créativité en minimisant la quantité de code nécessaire pour réaliser une tâche donnée. [35]

V.3.5. Technologies de Développement

Comme nous venons de le signaler dans les points qui précèdent, dans le cadre de notre projet, nous avons utilisé le langage de programmation PHP, et pour un langage de programmation il existe plusieurs techniques de l'écriture de code en l'occurrence la programmation procédurale, l'Orienté-Objet…Pour notre application nous avons adopté la programmation Orienté-objet avec l'architecture MVC qu'implémente le Framework CodeIgniter. Cette partie nous présente une brève description sur ces techniques utilisées.

La programmation orientée objet (POO) consiste en la définition et l'interaction de briques logicielles appelées objets ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait interagir avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; l'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues, de mieux résoudre le ou les problèmes. [37]

Cette technologie nous a été utile dans la mesure où nous avons plusieurs modules à développer pour notre application. Elle nous a permis de créer différentes classes selon l'architecture MVC qu'implémente le Framework CodeIgniter. Les classes que nous avons créées héritent soit du modèle, soit du contrôleur. Par exemple la classe <Utilisateur> pour les utilisateurs de notre système qui est elle aussi un contrôleur selon le MVC est hérité du contrôleur du Framework codeIgniter.

Le MVC, acronyme de Model View Controller (Modèle Vue Contrôleur), est une technique de développement avancée devenue un design pattern, qui découpe l'application en 3 couches principales, nommées Modèle, Vue et Contrôleur.

La distinction de ces couches : Facilite l'organisation des sources du projet, permet à chaque corps de métier de travailler en parallèle sur les sources qui leur sont dédiées et réduit l'impact des modifications pour minimiser les risques d'erreur et les régressions.

Nous pouvons représenter l'architecture MVC dans la figure 14.

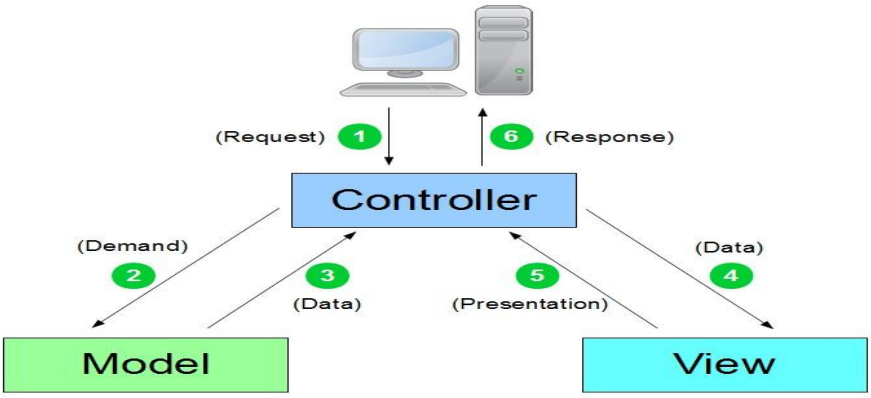


Figure 14: Architecture du Modèle Vue Contrôleur

L'objectif du MVC est de bien séparer les données, les traitements et la présentation de ces données. La règle est « divisée pour mieux régner ». Chaque couche est spécialisée, donc est dédiée aux tâches qu'elles savent faire le mieux. Ainsi :

1) Si une anomalie est découverte, il sera aisé de deviner la partie de l'application devant être corrigée sans impacter obligatoirement les autres ;
2) Si une évolution doit être apportée, alors les modifications ne concerneront que certaines couches et les composants bien spécifiques liés à cette évolution. [36]

Le **Contrôleur :** C'est la couche centrale entre le Modèle et la Vue. Son rôle consiste à coordonner les tâches à exécuter, et à gérer les erreurs. Ces dernières peuvent apparaître lors d'un contrôle de paramètres ou de l'exécution d'un traitement par le Modèle. Le Contrôleur est en fait responsable du bon déroulement de la requête de l'utilisateur, de sa réception au renvoi de la réponse à l'utilisateur. [36]

Le **modèle** correspond à la logique métier et utilise des composants dits métiers. Le Modèle désigne le traitement à exécuter demandé par l'utilisateur. Il est appelé par le Contrôleur après que les paramètres de la requête aient été une première fois vérifiés. Ces derniers sont des données d'entrée nécessaires à la bonne exécution du processus métier. [36]

Le rôle du Modèle consiste à traiter l'information reçue. Après le traitement, il peut envoyer ou non un résultat, ou indiquer qu'une erreur est apparue au cours de son exécution. C'est le seul contrat que le Modèle doit honorer avec son appelant : le Contrôleur ou autre (test unitaire). [36]

La **Vue** est la partie visible de l'utilisateur. Son rôle est d'afficher les données résultant du traitement du Modèle. La Vue récupère ces informations depuis le Contrôleur, formate les données si besoin puis construit la présentation avant de la renvoyer au Controller. Cette présentation, ou la réponse à la requête doit être sous le format de données attendu par l'utilisateur (client). [36]

V.4. Environnement de développement Visual Studio Code

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto-complétion, la coloration syntaxique, le débogage, et les commandes git. Il a été annoncé le 29 avril 2015 par Microsoft lors de la conférence Build 2015. Une version préliminaire a été publiée peu de temps après. Le 18 novembre 2015, Visual Studio Code a été publié sous la licence MIT et son code source publié sur GitHub. Le support d'extensions a également été annoncé. Le 14 avril 2016, Visual Studio Code est sorti de phase bêta et a été publié sur le Web. [38] [39]

Le code source de Visual Studio Code provient du projet logiciel libre et open source VS Code de Microsoft publié sous la licence MIT (Massachusetts Institute of Technologie) permissive, mais les binaires compilés constituent un freeware, c'est-à-dire un logiciel gratuit pour toute utilisation mais privateur. [39]

Le code source est fourni sous la licence libre MIT sur le site du projet sur GitHub. En revanche, l'exécutable est proposé sur le site officiel de Microsoft sous une licence propriétaire5. Le projet VSCodium6 propose une compilation du logiciel sans les outils de télémétrie inclus dans les binaires fournis par Microsoft. [39]

Ainsi, il est à signaler que cet éditeur a été beaucoup utile lors du développement de notre application car ses fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code et la factorisation du code. Les possibilités qu'il offre aux utilisateurs de modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires nous a permis de travailler aisément.

V.5. Sécurité du Système d'information

Un système d'information est constitué d'un ensemble d'entités interconnectées de façon permanente ou temporaire. Celui-ci est fermé (intranet et extranet) ou ouvert (Internet).

La sécurité d'un système d'information a pour finalité d'en garantir le fonctionnement et la continuité dans le temps. Une politique sécuritaire garantit la confidentialité des informations transmises vis-à-vis à de personnes non concernées ainsi que l'intégrité du système d'information.

Tenter de sécuriser un système d'information revient à essayer de se protéger contre les menaces intentionnelles et d'une manière plus générale contre tous les risques pouvant avoir une influence sur la sécurité de celui-ci ou des informations qu'il traite.

V.5.1 Principes génériques de sécurité

La sécurité des systèmes d'information se focalise généralement à garantir les droits d'accès aux données et ressources d'un système, en mettant en place des mécanismes d'authentification et de contrôle. Ces mécanismes permettent d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés.

La sécurité informatique doit toutefois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, c'est-à-dire : Élaborer des règles et des procédures, installer des outils techniques dans les différents services de l'organisation (autour de l'informatique), définir les actions à entreprendre et les personnes à contacter en cas de détection d'une intrusion, sensibiliser les utilisateurs aux problèmes liés à la sécurité des systèmes d'informations, préciser les rôles et responsabilités. Pour ce faire, nous avons définis des règles de sécurités de notre application afin de garantir quelques objectifs de la sécurité tels qu'ils sont expliqués dans le point suivant.

V.5.2 Sécurisation de notre système d'information

Le système de gestion des places d'hospitalisation mise en place est une application web accessible via le réseau ce qui veut dire qu'il doit être protéger contre les accès non autorisés. Pour garantir la sécurité de notre système nous avons défini les objectifs suivants :

1) *L'authentification* : L'authentification permet de vérifier l'identité annoncée et de s'assurer de la non usurpation de l'identité d'une entité. Pour cela, l'entité devra produire une information spécifique. Notre application garantit l'authenticité dans la mesure où chaque utilisateur doit être en possession de ses propres paramètres de connexion (email et mot de passe) pour avoir accès au système.

2) *La confidentialité* : La confidentialité est la protection des données contre une divulgation non autorisée. Etant donné qu'il existe 2 actions complémentaire pour assurer la confidentialité des données qui sont la limitation des accès par un mécanisme de contrôle d'accès et la transformation des données par des procédures de chiffrement.

Dans notre application, nous sommes parvenus à limiter les accès au système par un module de gestion des utilisateurs et leur droit d'accès contrôlé par un administrateur du système.

3) *La disponibilité* : Pour un utilisateur, la disponibilité d'une ressource est la probabilité de pouvoir mener correctement à terme une session de travail. Pour les utilisateurs qui ont accès, les ressources de notre système sont disponibles et utilisables avec un temps de réponses acceptable.

4) *Intégrité* : Le critère d'intégrité est relatif au fait que des ressources, données, traitements, transactions ou services n'ont pas été modifiés, altérés ou détruits tant de façon intentionnelle qu'accidentelle. Pour le système mis en place seuls les utilisateurs qui en ont le droit peuvent modifier ou supprimer une donnée.

V.6. Présentation des fonctionalités du nouveau système

Par des captures d'écran, cette partie nous présente quelques interfaces de notre système de gestion des places d'hospitalisation. Nous présentons également les fonctions principales car notre application possède plusieurs fenêtres.

V.6.1. Interface d'authentification

Lors de l'ouverture de l'application, la première fenêtre qui s'affiche est celle d'authentification. Et pour bénéficier ses services, l'utilisateur doit être obligatoirement en possession des paramètres de connexion qui sont l'adresse email et son mot de passe. Selon la catégorie d'utilisateur le système lui redirige à sa page d'accueil. Elle est donnée dans la figure 15.

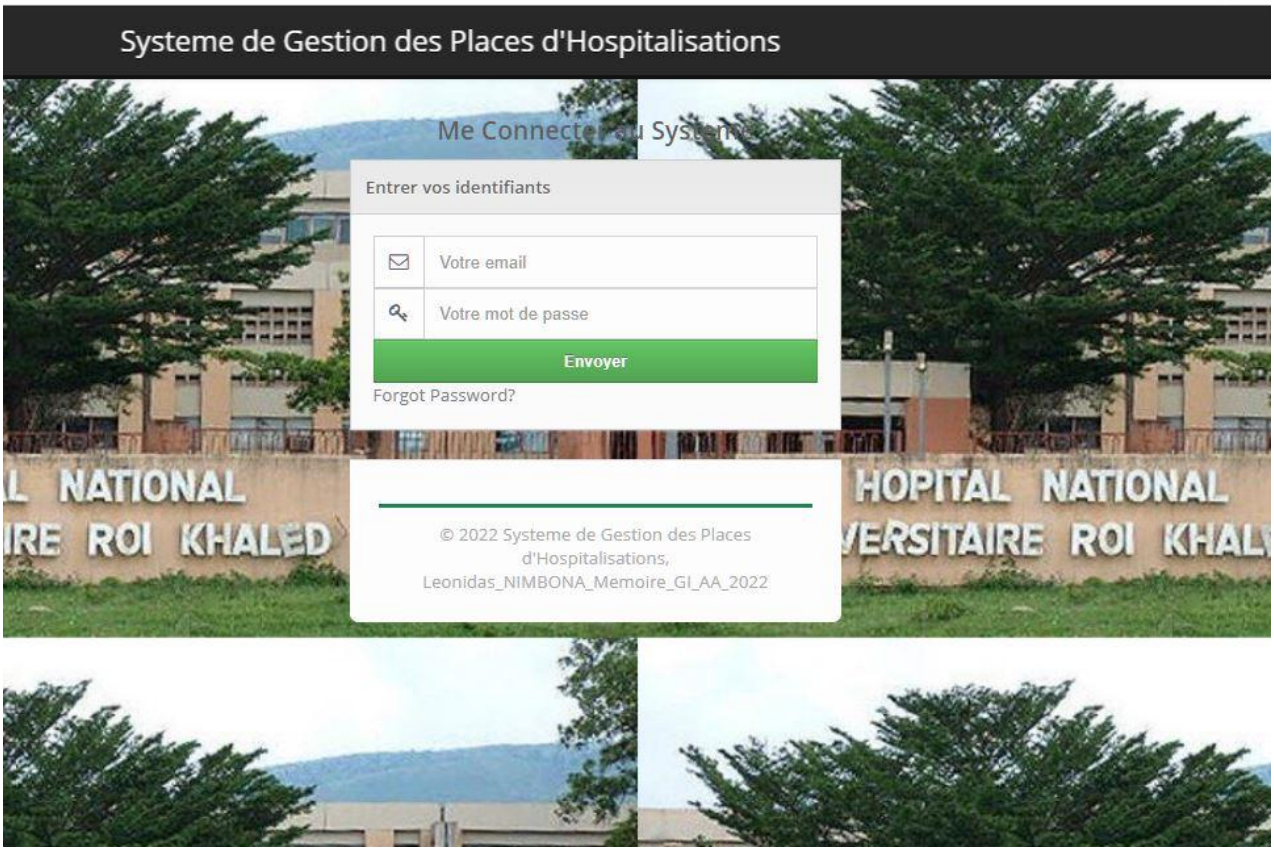


Figure 15: Interface d'authentification du nouveau système

V.6.2. Interface pour la création d'un dossier médical du nouveau patient

A son arrive, le patient doit être enregistrer par un agent d'accueil qui lui ouvre son dossier médical au sein du centre si c'est la première fois qu'il fréquente le centre de soin. L'interface d'enregistrement de ce patient est donnée par la figure 16.

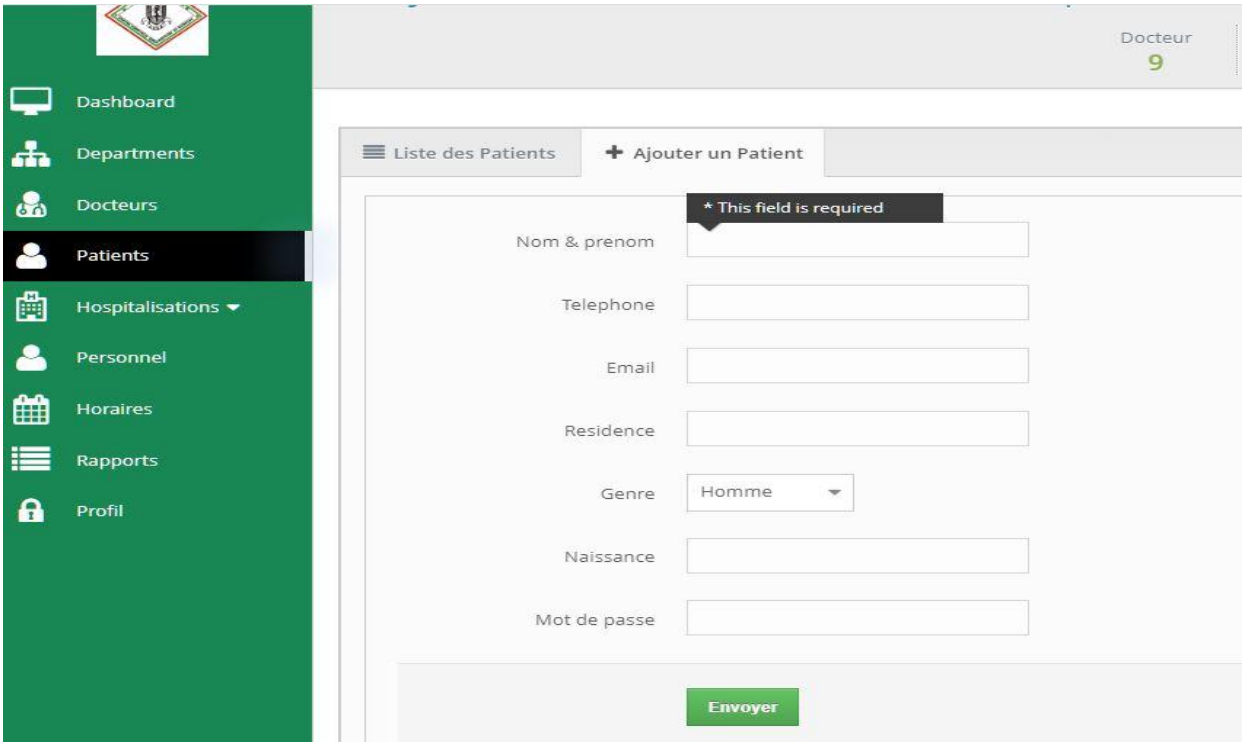


Figure 16 : Interface pour l'enregistrement d'un patient

V.6.3. Interface pour demander un rendez-Vous

Ayant déjà fréquenté le centre, le patient possède un mot de passe pour se connecter au système et en plus de la consultation de certaines informations, il peut aussi demander un rendez-vous à n'importe quel docteur selon sa spécialité.

Elle est donne dans la figure 17.

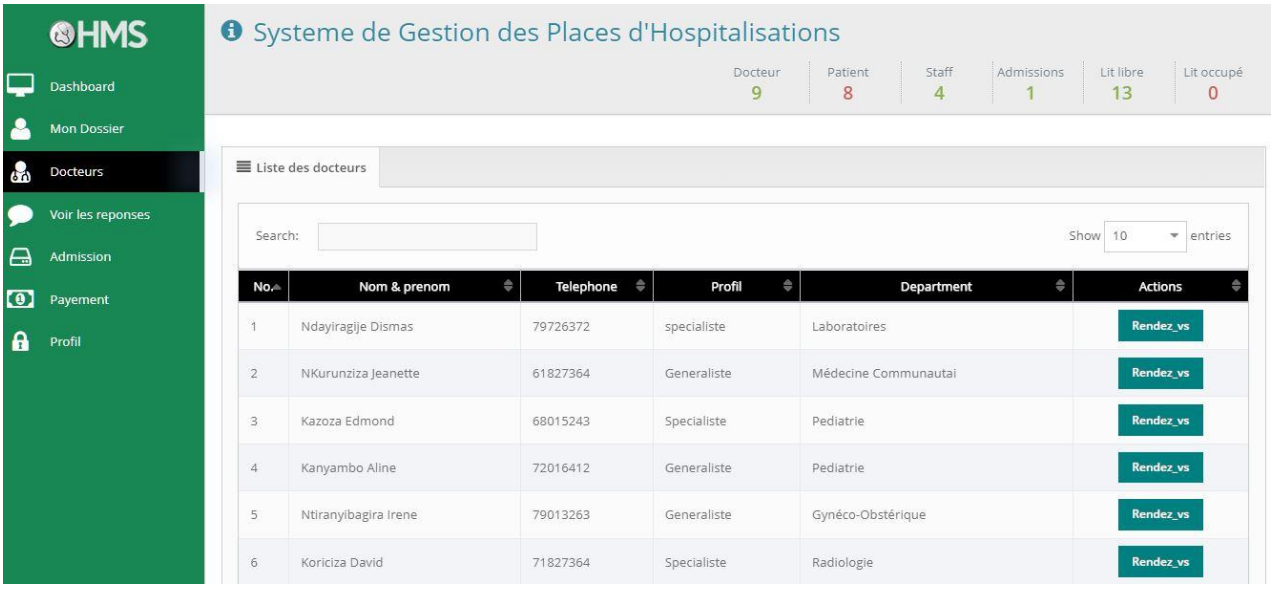


Figure 17 : Interface pour demander un rendez-vous

V.6.4. Interface pour affecter un lit

Après avoir consulté un patient, le docteur peut décider de l'hospitaliser, c'est ainsi qu'il lui attribue un lit selon leur disponibilité dans le département comme le montre la figure ci-dessous :

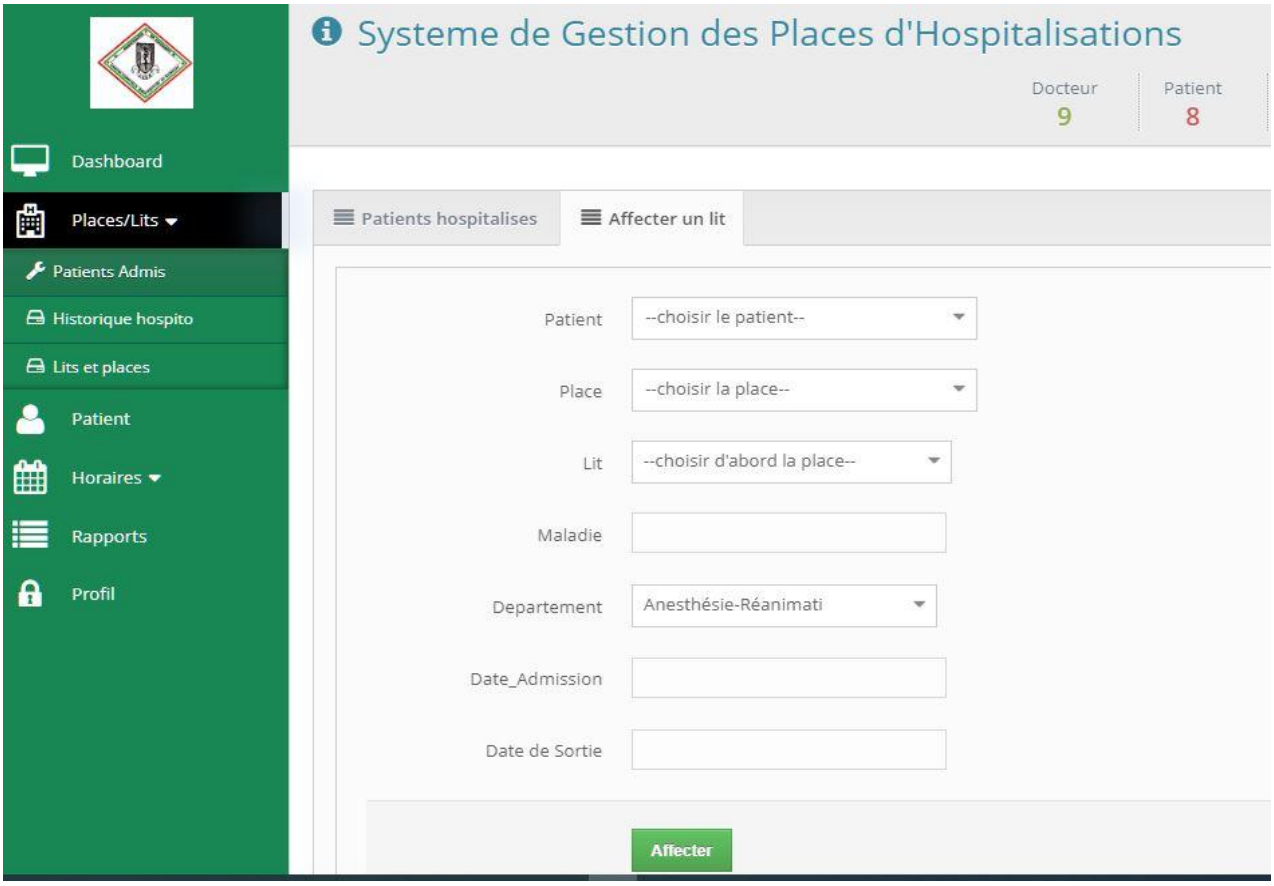


Figure 18 : Interface pour affecter un lit

V.6.5. Interface pour la gestion des hospitalisations

Elle permet de pouvoir localiser automatiquement un patient hospitalise sans parcourir tous les places. C'est dans ce sous-menu aussi qu'on peut voir la liste des lits et leur disponibilité, l'horaire pour le garde-malade, ainsi que les facturations au seins des hospitalisations.



Figure 19: Interface pour la gestion des hospitalisations

CONCLUSION GENERALE ET RECOMMANDATIONS

Ce travail de mémoire, intitulé Mise en place d'un système automatisé de gestion des places d'hospitalisation dans un centre de soins a porté sur le management des services d'hospitalisations notamment l'affectation des lits, le décharge des patients, les informations sur l'occupation des places etc. Les présentes recherches ont abouti à l'obtention d'une application permettant aux agents et au personnel œuvrant dans le service d'hospitalisation de suivre facilement tous les processus d'hospitalisation et d'en améliorer la qualité de service offerte.

Pour atteindre ce résultat, la première des étapes a été la correcte de données par l'approche de questionnement, l'observation ainsi que l'entretien libre. Il a été aussi important de prendre connaissance des systèmes déjà existant pour mettre en place un système informatique tenant compte des besoins des centres de soins dans la gestion de leurs services d'hospitalisation.

La modélisation basée sur le langage UML a été la deuxième étape qui portait sur l'identification des besoins des utilisateurs futurs du système. Avec le logiciel Microsoft Visio nous avons élaborer le diagramme de cas d'utilisation pour décrire les fonctionnalités du système tel que vues par les utilisateurs, le diagramme de classe pour représenter les objets manipulés par les utilisateurs et le diagramme de séquence pour voir comment s'effectuent les différents processus.

De même, à l'aide de la théorie des files d'attente, nous avons pu développer l'aspect mathématique de notre système en déterminant certaines caractéristiques de la file qui pourra aider les décideurs des centres de soins à prendre des meilleures décisions visant à améliorer la qualité de leurs services.

Le présent travail de recherche consistait en la mise en place d'une application qui gère spécialement les services d'hospitalisation. Bien qu'il possède d'autres fonctionnalités, il existe des aspects importants de l'étude qui n'ont pas été abordés pour des raisons diverses notamment la contrainte temporelle. C'est ainsi que nous voudrions adresser nos recommandations :

Aux futurs chercheurs d'améliorer le présent travail afin que le système soit centralisé et distribué d'un centre de soins à un autre. Aussi, il serait intéressant de développer un système qui gère toutes les services du centre ainsi que son personnel.

Et aux centres de soins d'intégrer ce système dans leurs activités quotidiennes en vue d'améliorer la qualité de leurs services notamment les services d'hospitalisation.

Références bibliographique

[1] Berbain X., Minvielle E. (2001). *L'informatique dans la gestion quotidienne des unités de soins : la barrière de l'apprentissage*, Revue Sciences sociales et Santé,

[2] https://ingenierie-creations.fr/ consulté le 28/06/2022

[3] HeaL., Chalil Sreenath M., Amrita O., Servis G., Khasawneha M. (2019). *A systematic review of research design and modeling techniques in patient bed management*. Computers & Industrial Engineering,

[4] Romeyer C. (2004). Les obstacles à la mise en œuvre des Systèmes d'Informations Centrés sur les Activités Hospitalières, *Revue Logistique et Management,*

[5] Mémoire : *Systèmes d'information hospitalier : Admission et planification des blocs opératoires.*

[6] Waring T.S., Alexander M. (2015). Innovations in patient flow and bed management: An action research project in a UK acute care hospital. *International Journal of Operations & Production Management*, 35(5): 751-781

[7] Décret N°100/865 du 08 mai 2020 portant révision du décret N°100/09 du 23 janvier 2019 portant réorganisation et fonctionnement du centre Hospitalo-Universitaire de Kamenge.

[8] *Société de Calcul Mathématique, SA Outils d'aide à la décision depuis 1995* sur http://www.scmsa.com/simul.htm consulté le 29/06/2022

[9] Auguste Rakotondranaivo, Sophie Pottier. *Gestion informatisée des lits : état des lieux et analyse a priori des impacts multicritères*. 10ème conférence Francophone en Gestion et Ingénierie des Systèmes Hospitaliers, GISEH2020, Oct. 2020, Valenciennes, France. Ffhal-03198921

[10] http://www.minisante.bi/*Plan National de Développement de l'Informatique de Santé 2020-2024*

[11] CHUK : Manuel d'utilisateur de l'outil Open clinic

[12] Mémoire : *Contribution à la Correction et l'Amélioration de la Qualité de Service dans une Entreprise Publique, en utilisant les Réseaux de Files d'Attente*

[13] https://www.gfmer.ch/Activites_internationales_Fr/CHUK.htm / consulté le16/06/2022

[14] https://www.chuk.bi / consulté le 16/06/2022

[15] U. Narayan Bhat. *An Introduction to Queueing Theory, Modeling and Analysis in Applications*

[16] https://www.universalis.fr/encyclopedie/ consulté le 05/08/2022

[17] *Introduction to Queuing Theory.* Second edition Robert B. Cooper

[18] Stephen Duckett. *Operations research for health planning and administration*

[19] OMI (Organisation Maritime Internationale). Évaluation de performances et sûreté de fonctionnement. *Techniques de modélisation: Méthodes analytiques*

[20] Aimé LACHAL *Modélisation en univers aléatoire*

[21] Cours : Bruno Bouzy https://www.cours-gratuit.com/cours-uml/cours-uml-pdf-complet

[22 pour programmer.Fig.Diagr.] Mohamed Nemiche *Modélisation UML : Diagramme de cas d'utilisation*

[23] Joseph Gabay, David Gabay Uml 2 : *Analyse Et Conception*

[24] Xavier Blanc, Isabelle Mounier : *UML 2 pour les developpeurs, Cours avec exercices corrigés*

[25] Christian Soutou : UML 2 Pour les bases de données

[26] Cours : *Introduction à UML 2*, Eric Cariou

[27] INP Grenoble. ENSIMAG, Cours : *Analyse, Conception et Validation de Logiciels*

[28] Bailly Gabriel, Bacchiocchi Julien, Moly Corentin, Verny Thiago et Glasson Emma :*Les cycles du développement logiciel.*

[29] Julie Vachon : *Génie logiciel du commerce électronique,* Hiver 2003.

[30] Stéphane Crozat (stph.scenari-community.org) : *Introduction aux bases de données.*

[31] https://sql.sh/sgbd/mysql, Consulté le 27/10/2022

[32] https://sql.sh/sgbd/mysql Consulté le 28/10/2022

[33] http://www.nexen.net *Manuel PHP 4.3.8 - Version Française* Consulté le 29/10/2022

[34] Cours PHP sur developpez.com 7-Cours PHP sur le site du zéro

[35] https://www.techno-science.net/definition/1471.html Consulté le 29/10/2022

[36] https://adventy.org/fr/mvc : *Introduction au MVC* Consulté le 25/10/2022

[37] https://www.futura-sciences.com/tech/definitions/informatique-programmation-orientee-objet 19301/

[38] https://framalibre.org/content/visual-studio-code Consulté le 29/10/2022

[39] https://code.visualstudio.com/ Consulté le 29/10/2022

[40] https://www.baumann-avocats.com/definition/cahier-des-charges.php Consulté le 31/10/2022

[41] Cours Master1: *Gestion des projets* par Dr. Michèle MUKESHIMANA

[42] Chemuturi, Murali, 1950-*Software estimation best practices, tools, & techniques: a complete guide for software project estimators*

[43] https://portail-ie.fr/resource/glossary/56/securite-des-systemes-dinformation. Consulté le 19/11/2022

[44] Cours Master1 : *Sécurité des réseaux* par Dr. Longin NDAYISABA

ANNEXE A

Tableau 15: Matrice de Complexité pour les ENTR

GRD	**DE**		
	1 à 4	**5 à 15**	**Plus de 15**
< 2	Simple (3)	Simple (3)	Moyen (4)
2	Simple (3)	Moyen (4)	Complexe (6)
Plus de 2	Moyen (4)	Complexe (6)	Complexe (6)

Tableau 16: Matrice de Complexité pour les SORT

GRD	**DE**		
	1 à4	**5 à 15**	**Plus de 15**
< 2	Simple (4)	Simple (4)	Moyen (5)
2 ou 3	Simple (4)	Moyen (5)	Complexe (7)
Plus de 3	Moyen (5)	Complexe (7)	Complexe (7)

Tableau 17: Matrice de Complexité pour les INT

GRD	**DE**		
	1 à 5	**6 à 19**	**Plus de 19**
< 2	Simple (3)	Simple (3)	Moyen (4)
2 ou 3	Simple (3)	Moyen (4)	Complexe (6)
Plus de 3	Moyen (4)	Complexe (6)	Complexe (6)

Tableau 18: Matrice de Complexité pour les GDI

SLD	**DE**		
	1 à 19	**20 à 50**	**51 ou plus**
1	Simple (7)	Simple (7)	Moyen (10)
2 à 5	Simple (7)	Moyen (10)	Complexe (15)
Plus de 5	Moyen (10)	Complexe (15)	Complexe (15)

Tableau 19: Matrice de Complexité pour les GDE

SLD	**DE**		
	1 à 19	**20 à 50**	**51 ou plus**
1	Simple (5)	Simple (5)	Moyen (7)
2 à 5	Simple (5)	Moyen (7)	Complexe (10)
Plus de 5	Moyen (7)	Complexe (10)	Complexe (10)

Notez que les valeurs données entre parenthèse représentent le poids attribué à chaque degré de complexité.

ANNEXE B

14 Caractéristiques Générale du Système (CGS)

Tableau 20: Caractéristiques générales du système (CGS)

CGS	**Description**
1. Communications de données	Le système en cours de développement devra-t-il gérer les aspects liés à la communication des données ?
2. Traitement distribué des données	Le système en cours de développement devra-t-il prévoir le traitement distribué des données ?
3. Performance	Dans quelle mesure les délais d'exécution sont-ils critiques ?
4. Configuration fortement utilisée	Le système en cours de développement doit-il optimiser l'utilisation du matériel ?
5. Taux de transaction	Le système est-il appelé à effectuer de nombreuses transactions ?
6. Saisie des données en ligne	Toutes les données seront-elles saisies en ligne par les utilisateurs ?
7. Efficacité pour l'utilisateur final	Le système en cours de développement devra-t-il être conçu pour être efficace aux utilisateurs finaux ?
8. Mise à jour en ligne	Combien de GDI seraient-ils mis à jour en ligne par le système ?
9. Traitement complexe	Le système devra-t-il traiter des équations mathématiques ou des logiques complexes ?
10. Réutilisabilité	Le système en cours de développement devra-t-il être conçu de manière à pouvoir répondre aux besoins d'utilisateurs disparates ?
11. Facilité d'installation	L'application développée doit-elle être conçue de manière à faciliter son installation ?
12. Facilité d'exploitation	L'application contiendrait-elle des modules destinés à faciliter la tâche du personnel d'exploitation ?
13. Sites multiples	Est-il prévu que le système soit installé sur plusieurs sites ?
14. Faciliter le changement	Serait-il nécessaire de garder à l'esprit la maintenabilité du logiciel lors du développement de l'application ?

Degrés d'Influence (DI)

Tableau 21 : Degrés d'influence et leurs descriptions

Degré d'Influence	Description
0	Nulle
1	Mineure
2	Modérée
3	Moyenne
4	Significative
5	Essentielle