

Université Abderrahmane Mira de Bejaia
Faculté des sciences exactes
Département d'informatique

Rapport projet fin de cycle (licence)

Thème

**Conception et réalisation d'une application
De suivi de patients dans un établissement
Hospitalier.**

présenté par :

KLOUL Rassim

IBBOU Hamza

Hadj naceur Adlane

Medjani Massinissa

Zouggar Amias

Zouaghi Soumaya

Encadré par :

Table des matières

Introduction	3
1 Généralités	4
Introduction	4
1.1 Présentation de l'UML	4
1.1.1 Définition	4
1.1.2 Diagrammes UML	4
1.2 Quelques définitions relatives aux bases de données	5
1.2.1 Définition d'une base de données	5
1.2.2 Définition d'une base de données informatisée	5
1.2.3 Utilité d'une base de données	5
1.2.4 Les critères d'une base de données	5
1.2.5 Objectifs d'une base de données	6
1.2.6 La gestion des bases de données	6
Conclusion	6
2 Spécification des besoins	7
Introduction	7
2.1 Règles de gestion des dossiers par le système	7
2.2 Exigences du Système	7
2.3 Problématique	8
2.4 L'objectif de l'étude	8
Conclusion	9
3 analyse des besoins	10
Introduction	10
3.1 Diagramme de cas d'utilisation	10

3.1.1 Identification des acteurs	10
3.1.2 Identification des cas d'utilisation	10
3.2 Diagramme de séquence	13
3.3 Diagramme d'activité	22
Conclusion	29
4 Conception	30
Introduction	30
4.1 Diagramme de classe	30
4.1.1 Définition du concept de base.....	30
4.1.2 Identification des classes	31
4.1.3 Réalisation du diagramme de classe	32
4.2 Modèle relationnel de données	34
4.2.1 Quelques définitions	34
4.2.2 Règles de passage	35
Conclusion	36
5 Réalisation	
Introduction	37
5.1 Outils de développement	37
5.1.1 Définitions	37
5.2 Description de l'application	38
5.3 Les interfaces de l'application	39
5.4 Exemples de codes	42
Conclusion	44
Conclusion et perspectives	45
Références	46



Introduction

L'évolution technologique de ces dernières années a conduit la société moderne à adopter de nouvelles habitudes face au travail. La technologie actuelle nous épargne de certaines peines et nous fait gagner beaucoup de temps en mettant à notre disposition le système informatisé.

Avant on enregistrait toutes les informations écrites à la main ou imprimées sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières.

Le programme de médicalisation des systèmes d'informations hospitaliers a relancé l'installation des ordinateurs dans les hôpitaux, en insistant sur la notion de système d'information hospitalier. Comment y intégrer le dossier médical informatisé dont l'utilité est en grande partie liée à la précision des informations qu'il contient, et sachant que cette précision se conjugue mal avec la standardisation ?

Nous proposons un modèle qui tente de répondre à cette question en utilisant la sémantique des informations médicales et hospitalières dans la structuration et à l'utilisation des bases de données représentant les dossiers médicaux. La démarche consiste à repérer la sémantique que l'on incorpore au schéma de la base de données pour aider l'établissement à mieux l'utiliser.

L'intégration de l'outil informatique dans le SIH permet d'améliorer le stockage, la diffusion et le traitement de l'information, ainsi qu'un gain de temps pour le personnel hospitalier. Le système informatique est destiné à faciliter la gestion de l'ensemble des informations médicales et administratives d'un hôpital.

Chapitre

1

Généralités

Introduction

Ce chapitre définit et parle sur certains concepts nécessaires et utiles dans les notions de base de données. Les données permanentes constituent certainement l'objet de base à partir duquel l'utilisateur va élaborer la plupart de ses applications. Certaines applications se réduisent à gérer et consulter les données, ces données sont rangées dans des fichiers et sont structurées en enregistrement. Pour le développement de notre application de gestion d'un patient hospitalisé, nous avons choisi le formalisme UML comme méthode de modélisation.

1.1 Présentation de l'UML

1.1.1 Définition

UML (unified modeling language) est aujourd'hui un standard très largement utilisé dans l'analyse et la conception des systèmes informatiques. La modélisation est incontournable pour permettre aux différents acteurs de coopérer et de dialoguer efficacement. Il est donc important de connaître le langage et les techniques de modélisation et de savoir quels modèles sont les plus appropriés dans chaque situation. Il est enfin important de replacer l'utilisation des modèles dans le processus de développement du projet.

1.1.2 Diagrammes UML

UML s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont repartis en deux grands groupes :

- Six diagrammes structurels :
Diagramme de classe, Diagramme d'objet, Diagramme de package, Diagramme de structure composite, Diagramme de composants et Diagramme de déploiement.
- Sept diagrammes comportementaux :
Diagramme de cas d'utilisation, Diagramme de vue d'ensemble des interactions, Diagramme de séquence, Diagramme de communication, Diagramme de temps, Diagramme d'activité et Diagramme d'états.

1.2 Quelques définitions relatives aux bases de données :

1.2.1 Définition d'une base de données :

Une base de données est un ensemble organisé d'informations pour un objectif commun. Peu importe le support utilisé pour rassembler et stocker les données (papier, fichiers, etc.), dès lors que des données sont rassemblées et stockées d'une manière organisée dans un but spécifique, on parle d'une base de données.

1.2.2 Définition d'une base de données informatisée :

Une base de données informatisée est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur, représentant des informations du monde réel et pouvant être interrogées et mises à jour par une communauté d'utilisateurs.

1.2.3 Utilité d'une base de données :

Une base de données permet de mettre des données à la disposition d'utilisateurs pour une consultation, saisie ou bien une mise à jour, tout en s'assurant des droits accordés à ces derniers. Cela est d'autant plus utile que les données informatiques sont de plus en

Une base de données peut être locale, c'est-à-dire utilisable sur une machine par un utilisateur, ou bien repartie, c'est-à-dire que les informations sont stockées sur des machines distantes et accessible par réseau.

L'avantage majeur de l'utilisation de base de données est la possibilité de pouvoir y accéder par plusieurs utilisateurs simultanément.

1.2.4 Les critères d'une base de données :

- Exhaustivité : implique que l'on dispose de toutes les informations relatives au sujet donné.

- La non redondance : implique l'unicité de l'information dans la base de données. En général on essaie d'éviter la duplication des données car cela pose des problèmes de cohérence lors des mises à jour de ces données.
- La structure : implique l'adaptation du mode de stockage des renseignements aux traitements qui les exploiteront et les mettront à jour, ainsi qu'au coût du stockage de ces renseignements dans l'ordinateur.

1.2.5 Objectifs d'une base de données :

La base de données a beaucoup d'objectifs parmi lesquels nous pouvons citer :

- Eviter les redondances et l'incohérence des données.
- Offrir un langage du haut niveau pour la définition et la manipulation des données.
- Contrôler l'intégrité entre plusieurs utilisateurs et la confidentialité des données.
- Assurer l'indépendance entre les données et les traitements.

1.2.6 La gestion des bases de données :

Afin de pouvoir contrôler les données ainsi que les utilisateurs, le besoin d'un système de gestion s'est vite fait ressentir. La gestion d'une base de données se fait par un système appelé **SGBD** (système de gestion de bases de données). Le SGBD est un ensemble de service (applications logicielles) permettant de gérer les bases de données, c'est-à-dire :

- Permettre l'accès aux données de façon simple.
- Autoriser un accès aux informations à multiples utilisateurs.
- Manipuler les données présentes dans la base de données (insertion, suppression, modification).

Conclusion

Dans ce chapitre, nous avons pu connaître quelques notions de base concernant les bases de données, utilité, objectifs ainsi que la gestion des bases de données.

Chapitre

2

Spécification des besoins

Introduction

Ce projet consiste à concevoir un programme permettant à une clinique de gérer tous ses besoins concernant les patients, en utilisant une base de données permettant la sauvegarde des informations les concernant comme la gestion des entrées, séjours et leurs sorties.

2.1. Règles de gestion des dossiers par le système

Notre système à pour rôle de gérer les dossiers médicaux des patients du centre hospitalier qui sont actuellement des dossiers papiers, en suivant ces règles :

- Le patient qui doit consulter ou se faire hospitaliser doit d'abord passer par l'agent de saisie pour régler les questions administratives et financières. Le patient est ensuite envoyé vers le service qui le prend en charge.
- Parallèlement un dossier médico-administratif du patient est ressorti.
- Lorsqu'un patient quitte le centre hospitalier, le dossier est transmis aux archives.
- Un patient est placé dans un lit situé dans une salle numérotée. Chaque salle est rattachée à un service clinique.

2.2. Exigences du système

Notre système doit répondre aux exigences suivantes : Le système doit pouvoir récupérer des informations de chaque entité à partir de son identifiant pour mettre à jour la base de données de l'application.

- L'insertion de nouveaux patients et d'autres entités et les orienter vers une salle d'un service quelconque.
- La modification des informations à propos du patient et des autres entités.
- Recherche d'informations et de renseignements sur un patient par nom, prénom ou numéro.
- La suppression au niveau de la base de données des informations concernant un patient ayant quitté le centre hospitalier puis l'archivage de ces informations.
- L'impression des documents comme (bulletin d'admission, billet de salle, certificat de séjour, déclaration de décès, etc...).

2.3. Problématique

Notre système consiste à faire le suivi administratif des patients dès leur admission jusqu'à leur sortie (décès, transfert...), après avoir entamé notre étude au sein de la clinique « IFER » et après maintes interviews avec le personnel concerne nous avons pu recenser les problèmes suivants :

- La difficulté de la recherche des informations concernant les patients et les autres entités dans des brefs délais.
- Un taux d'erreur important dans le calcul des statistiques.
- perte de temps considérable dans l'établissement des documents manuellement.
- Détérioration des supports (registres) à cause de leur fréquente utilisation.

2.4. L'objectif de l'étude

La solution informatique à pour objectif :

- Traiter les informations avec rapidité et précision et rendre fiable la gestion de l'information.
- Connaitre à tout moment toutes les informations concernant les patients hospitalisés.
- Réduire la perte de temps dans la recherche des informations ainsi que dans l'établissement des documents.

- Utiliser des supports magnétiques pour le stockage des informations.
- sécuriser les données des patients.
- l'optimisation du temps de traitement et de l'espace de stockage.
- La sauvegarde d'une grande masse de données pour une longue durée.
- Améliorer l'organisation des informations.

Conclusion

Dans ce chapitre, nous avons étudié les besoins des patients et les exigences du système qu'on doit mettre en œuvre et nous avons ainsi noté quelques problèmes et les objectifs à atteindre afin de le réussir.

Chapitre

3

Analyse des besoins

Introduction

Cette phase consiste à déterminer les étapes fondamentales pour le développement de notre application de gestion de patients dans un établissement hospitalier en modélisant avec le formalisme UML comme méthode de conception.

3.1 Diagramme de cas d'utilisation

C'est une représentation graphique fonctionnelle entre les acteurs et le système à l'étude. Le diagramme cas d'utilisation doit permettre de répondre à la question « qui fait quoi ».

Les acteurs qui interagissent avec notre système sont : l'administrateur, l'agent de saisie, le médecin.

3.1.1 Identification des cas d'utilisation :

- **Cas d'utilisation « Authentifier »** : L'application vérifie l'identité de l'utilisateur à savoir administrateur, agent de saisie ou médecin avant de lui attribuer les fonctionnalités correspondantes.
- **Cas d'utilisation « ajouter »** : L'administrateur est le seul qui peut ajouter des médecins, un agent saisie et tout type d'employé, un agent de saisie est le seul qui peut ajouter des patients.
- **Cas d'utilisation « archiver »** : L'administrateur est le seul qui peut déplacer les informations des employés dans la base de données vers l'archive, l'agent de saisie est le seul qui peut déplacer les informations des patients dans l'archive.

- **Cas d'utilisation « modifier »** : L'administrateur est le seul qui peut modifier les informations des employés, l'agent de saisie est le seul qui peut modifier les informations des patients en cas d'erreur ou d'apparition de nouvelle informations.
- **Cas d'utilisation « rechercher »** : L'administrateur peut rechercher des informations sur les employés, l'agent de saisie et les médecins peuvent rechercher des informations sur les patients (dans la base de données ou dans l'archive).
- **Cas d'utilisation « remplir la fiche de suivi »** : Le médecin rempli une fiche de suivi pour chaque patient durant la période de son hospitalisation pour pouvoir suivre son état.
- **Cas d'utilisation « imprimer »** : L'agent de saisie à pouvoir imprimer un bulletin propre à un patient (bulletin d'admission, billet de salle, fiche de suivi...).
- **Cas d'utilisation « calculer les statistiques »** : L'agent de saisie à pour mission de calculer les statistiques (nombre de nouveaux nés, nombre d'accidentés, moyenne de décès et de naissance par mois...) .

D'où la présentation de notre diagramme de cas d'utilisation :

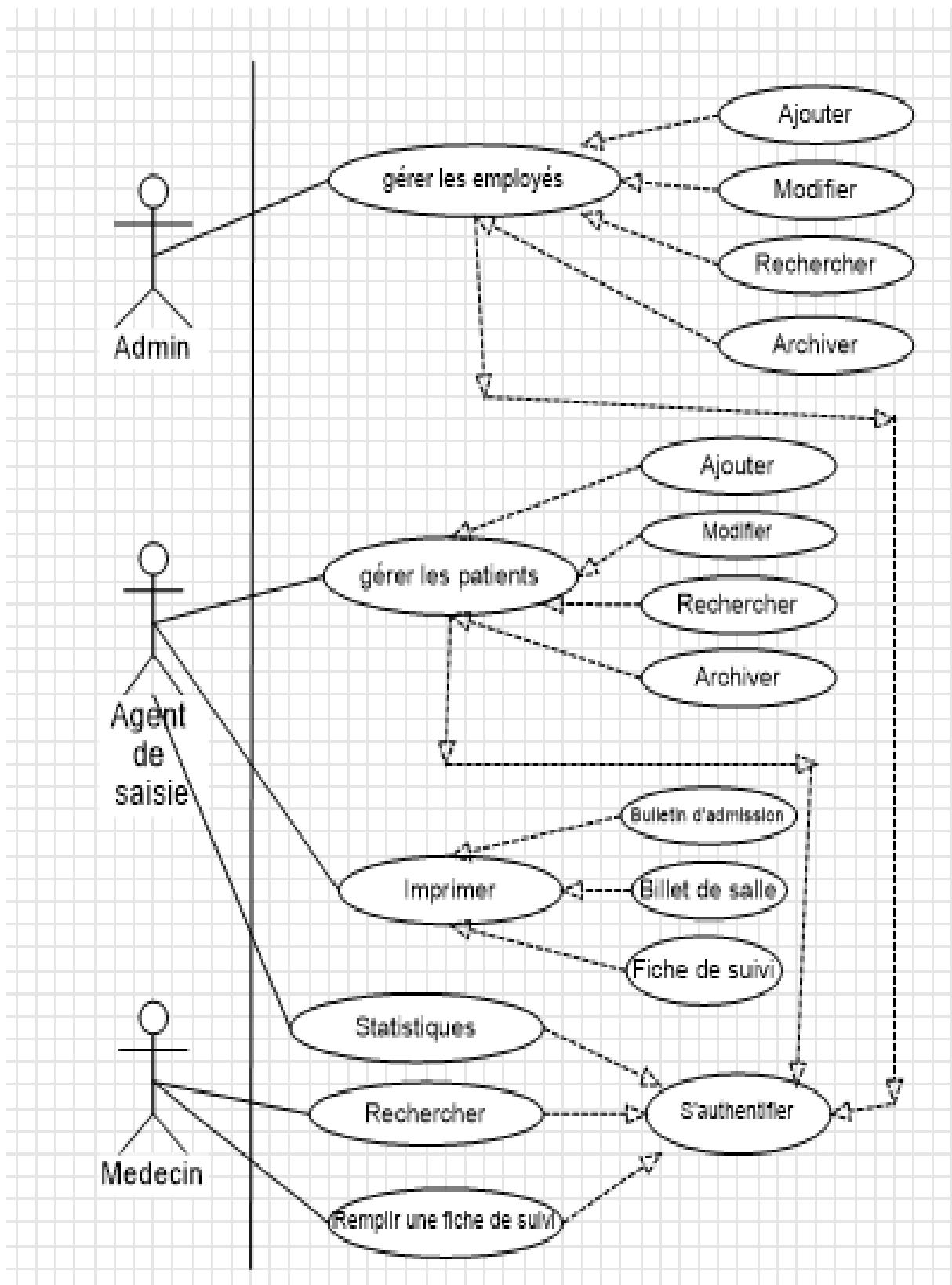


FIGURE 3.1 – Diagramme de cas d'utilisation

3.2 Diagramme de séquence

Les diagrammes de séquence sont utilisés pour illustrer les cas d'utilisation en montrant comment les objets s'utilisent mutuellement. Comme les diagrammes de séquence sont toujours lus du haut vers le bas ils illustrent l'ordre dans lequel les messages sont envoyés entre les objets.

3.2.1 Diagramme de séquence du cas d'utilisation « Authentifier » :

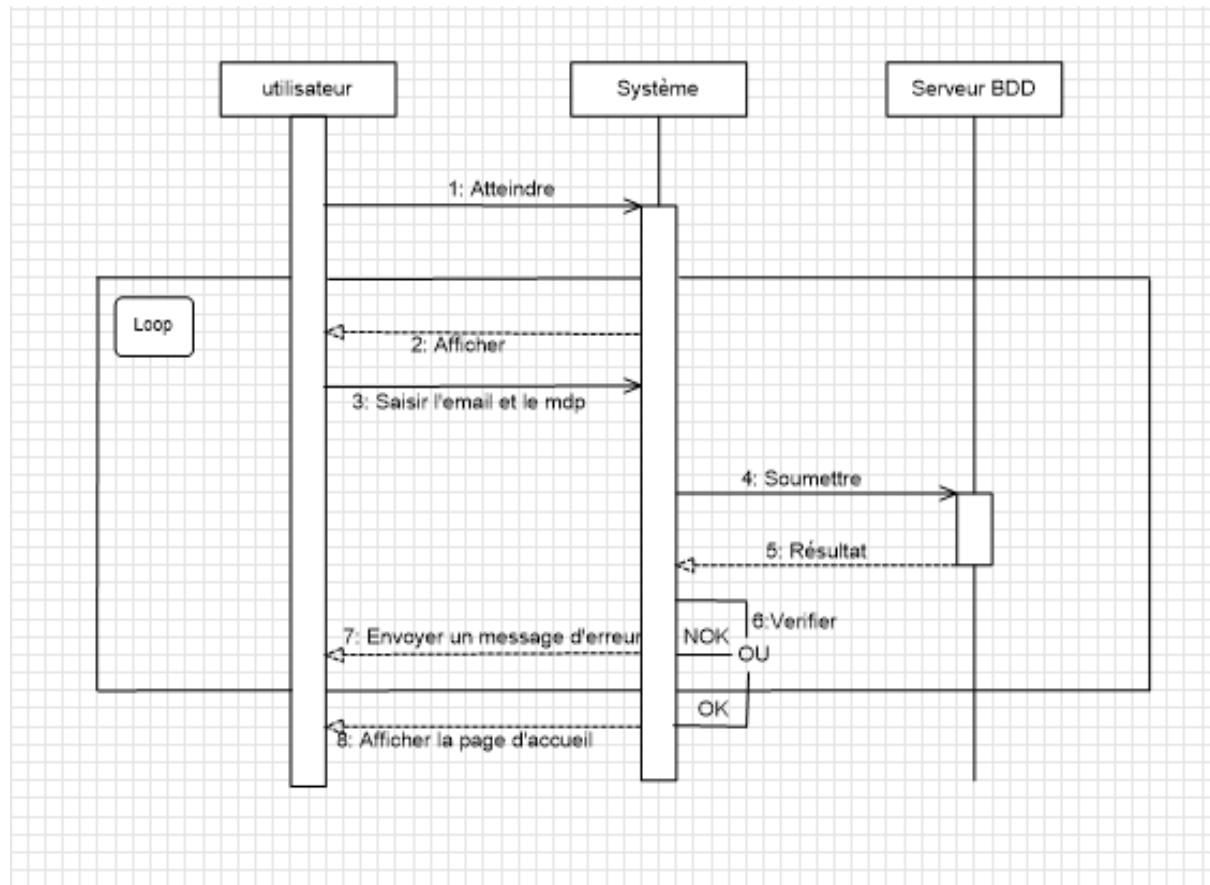


FIGURE 3.2 – Diagramme de séquence « authentifier »

1 :: Atteindre le formulaire d'authentification.

2 : L'application affiche le formulaire d'authentification.

3 : L'utilisateur saisie son email et son mot de passe.

4 : Le système soumet une requête de recherche à la base de données.

5 : Le résultat de la recherche.

6 : Le système vérifie le résultat.

7 : Le système envoie un message d'erreur si l'email ou le mot de passe est erroné.

8 : L'application affiche la page d'accueil.

3.2.2 Diagramme de séquence du cas d'utilisation « ajouter » :

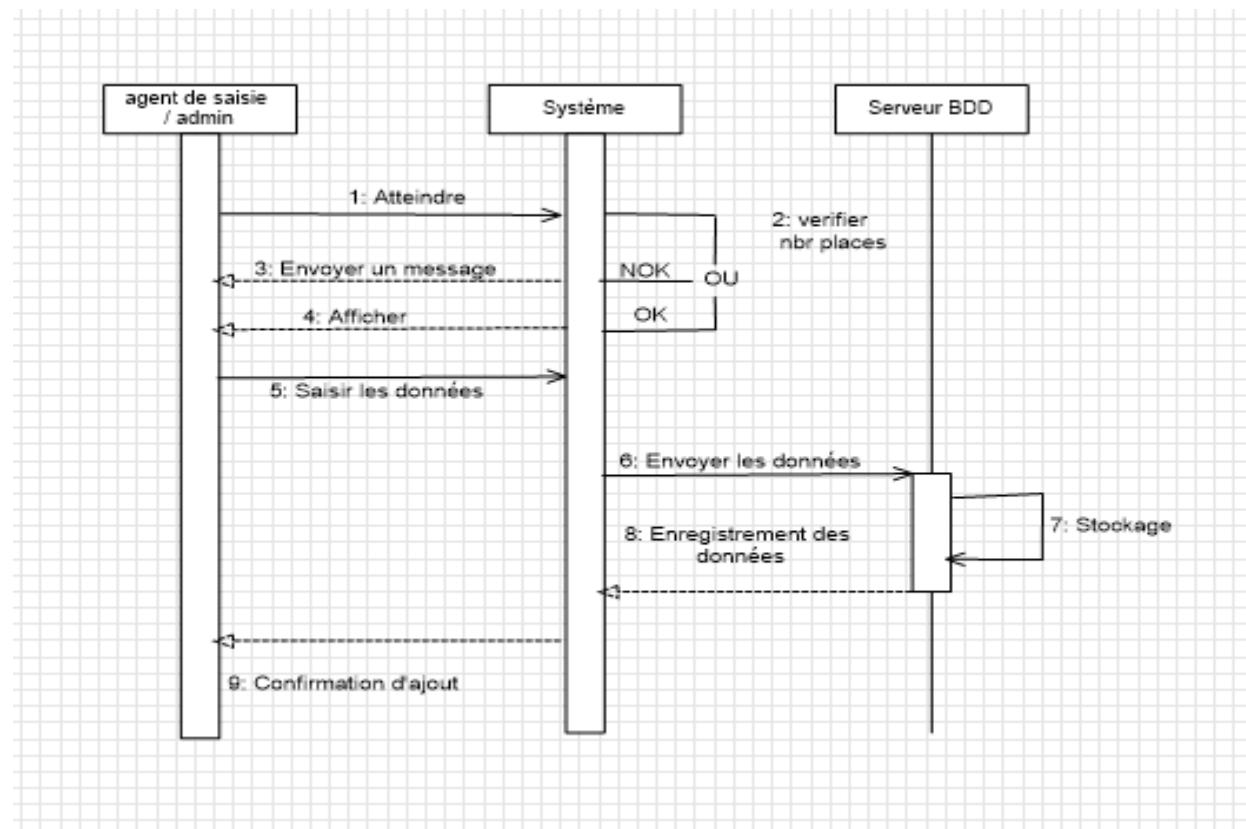


FIGURE 3.3 – Diagramme de séquence « ajouter »

1 : Atteindre le formulaire d'ajout.

2 : Le système vérifie le nombre de places disponible.

3 : Le système envoie un message indiquant la saturation.

- 4** : L'application affiche le formulaire d'ajout.
- 5** : L'utilisateur saisit les nouvelles données.
- 6** : Le système envoie une requête à la BDD .
- 7** : Le stockage de données au niveau de la BDD.
- 8** : L'enregistrement de données est effectué.
- 9** : L'application confirme l'ajout par un message.

3.2.3 Diagramme de séquence du cas d'utilisation « rechercher » :

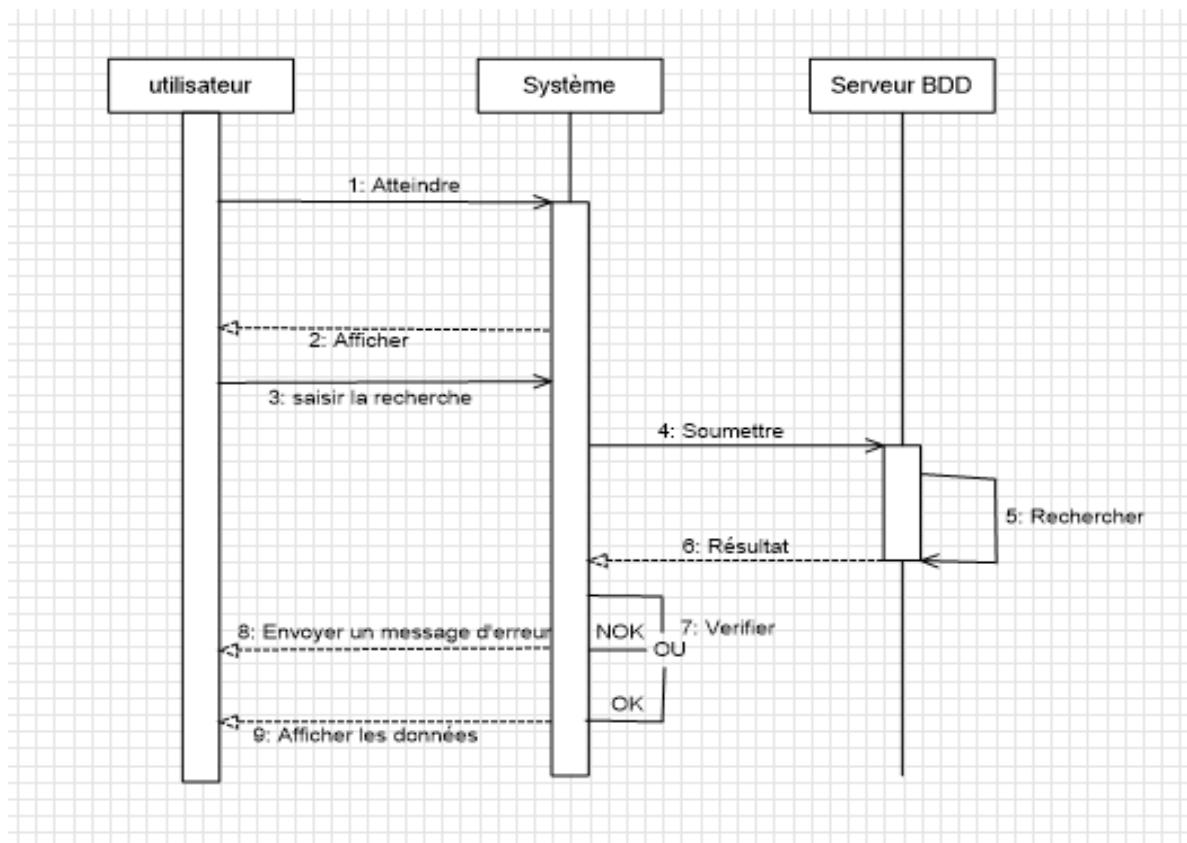


FIGURE 3.4 – Diagramme de séquence « rechercher »

- 1** : Atteindre la barre de recherche.
- 2** : L'application affiche la barre.

3 : L'utilisateur saisie une recherche.

4 : Le système soumet une requête à la BDD.

5 : Une recherche se fait au niveau de la BDD.

6 : Résultat de la recherche.

7 : Le système vérifie le résultat.

8 : Le système envoie un message si les données recherchées n'existent pas.

9 : L'application affiche les données recherchées.

3.2.4 Diagramme de séquence du cas d'utilisation « modifier » :

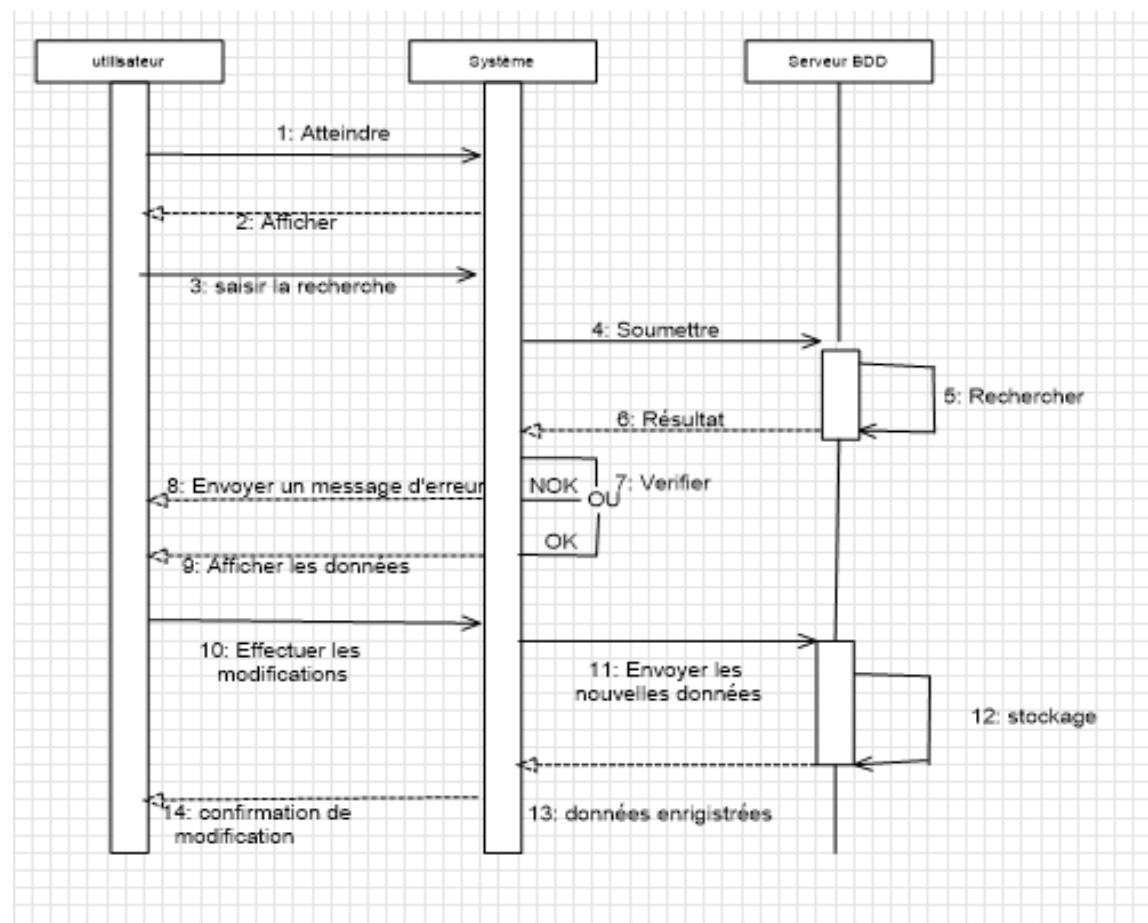


FIGURE 3.5 – Diagramme de séquence « modifier »

- 1** : Atteindre la barre de recherche.
- 2** : L'application affiche la barre.
- 3** : L'utilisateur saisie une recherche.
- 4** : Le système soumet une requête à la BDD.
- 5** : Une recherche se fait au niveau de la BDD.
- 6** : Résultat de la recherche.
- 7** : Le système vérifie le résultat.
- 8** : Le système envoie un message si les données recherchées n'existent pas.
- 9** : L'application affiche les données recherchées.
- 10** : L'utilisateur effectue les modifications.
- 11** : Le système envoie les nouvelles données à la BDD.
- 12** : Un stockage se fait au niveau de la BDD .
- 13** : L'enregistrement de données est effectué.
- 14** : L'application confirme la modification par un message.

3.2.5 Diagramme de séquence du cas d'utilisation « archiver » :

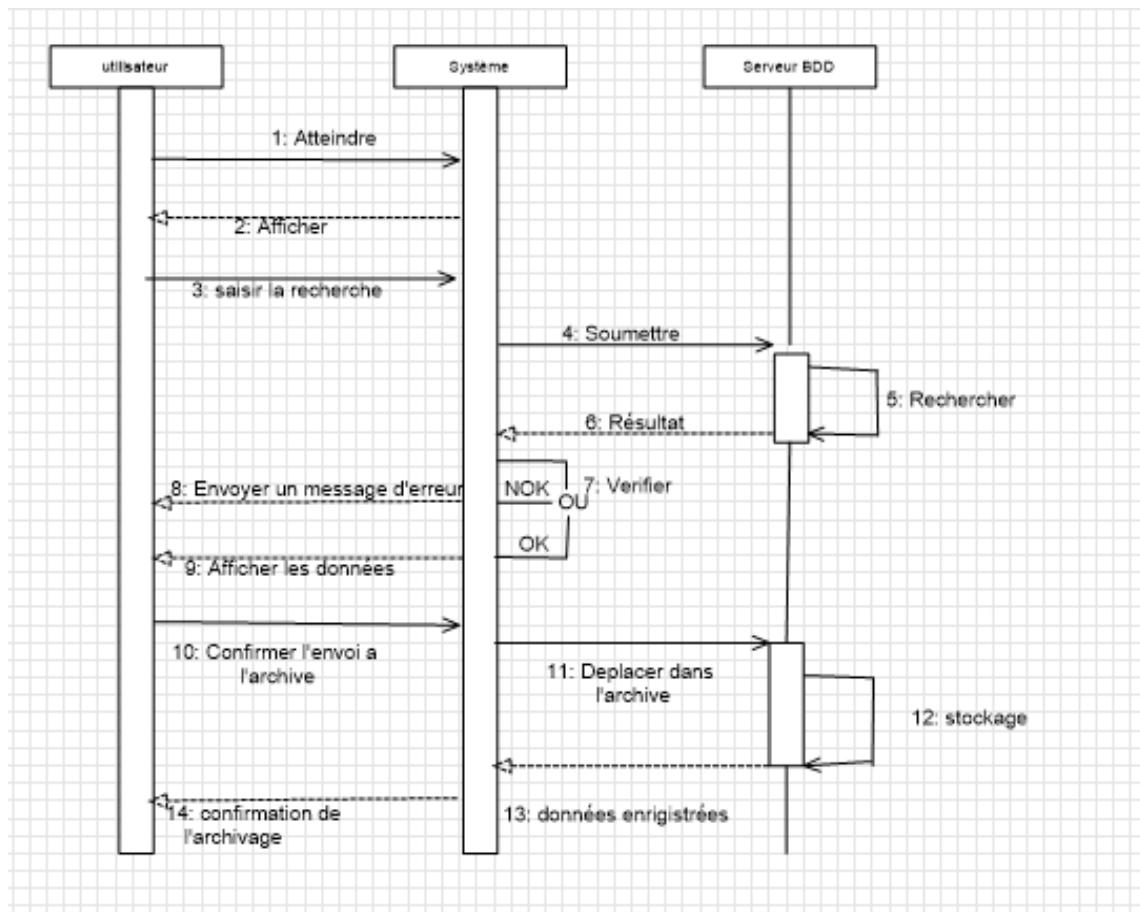


FIGURE 3.6 – Diagramme de séquence « archiver »

1 : Atteindre la barre de recherche.

2 : L'application affiche la barre.

3 : L'utilisateur saisie une recherche.

4 : Le système soumet une requête à la BDD.

5 : Une recherche se fait au niveau de la BDD.

6 : Résultat de la recherche.

7 : Le système vérifie le résultat.

8 : Le système envoie un message si les données recherchées n'existent pas.

9 : L'application affiche les données recherchées.

10 : L'utilisateur confirme l'envoie à l'archive (appuie sur archiver).

11 : Le système déplace les données dans la table archive de la BDD.

12 : Un stockage se fait au niveau de la BDD .

13 : L'enregistrement de données est effectué.

14 : L'application confirme l'archivage par un message.

3.2.6 Diagramme de séquence du cas d'utilisation « remplit la fiche de suivi » :

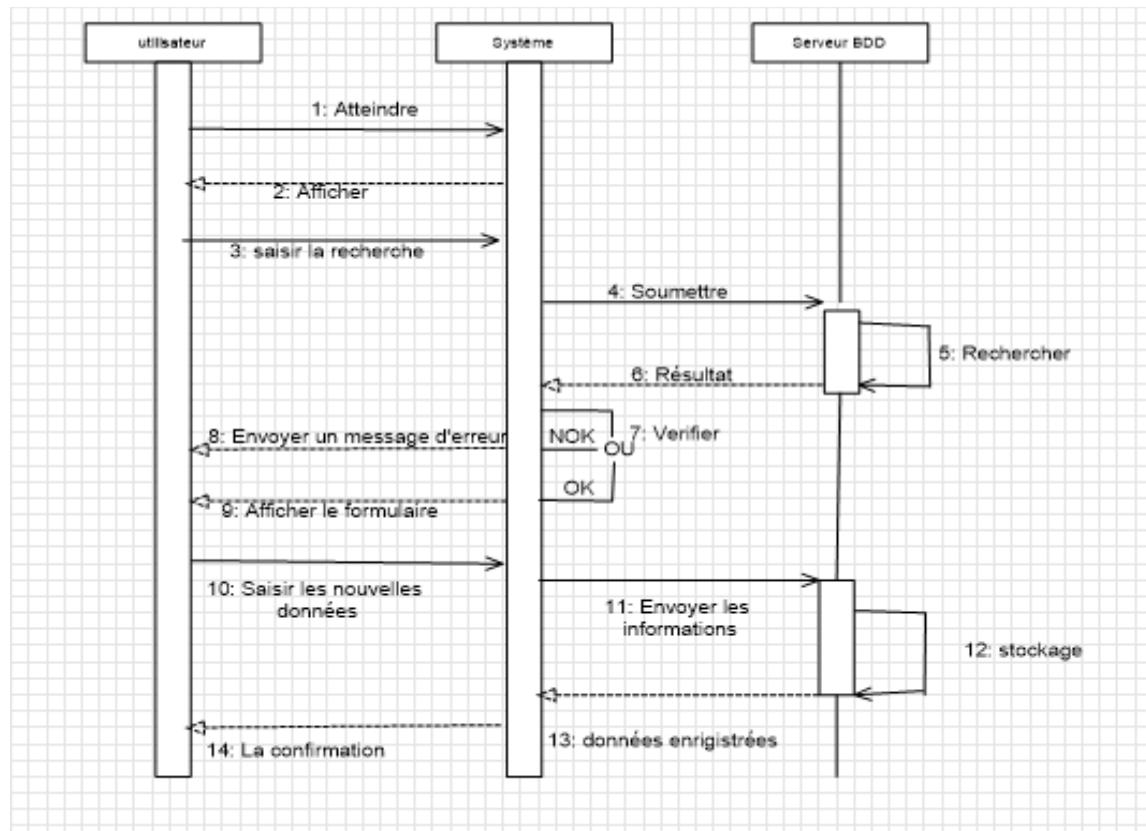


FIGURE 3.7 – Diagramme de séquence « remplir la fiche de suivi »

1 : Atteindre la barre de recherche.

2 : L'application affiche la barre.

3 : Le médecin saisie une recherche.

- 4 :** Le système soumet une requête à la BDD.
- 5 :** Une recherche se fait au niveau de la BDD.
- 6 :** Résultat de la recherche.
- 7 :** Le système vérifie le résultat.
- 8 :** Le système envoie un message si les données recherchées n'existent pas.
- 9 :** L'application affiche le formulaire (fiche de suivi).
- 10 :** Le médecin saisit les données.
- 11 :** Le système soumet les données à la BDD.
- 12 :** Un stockage des données au niveau de la BDD .
- 13 :** L'enregistrement de données est effectué.
- 14 :** L'application confirme par un message.

3.2.7 Diagramme de séquence du cas d'utilisation « imprimer » :

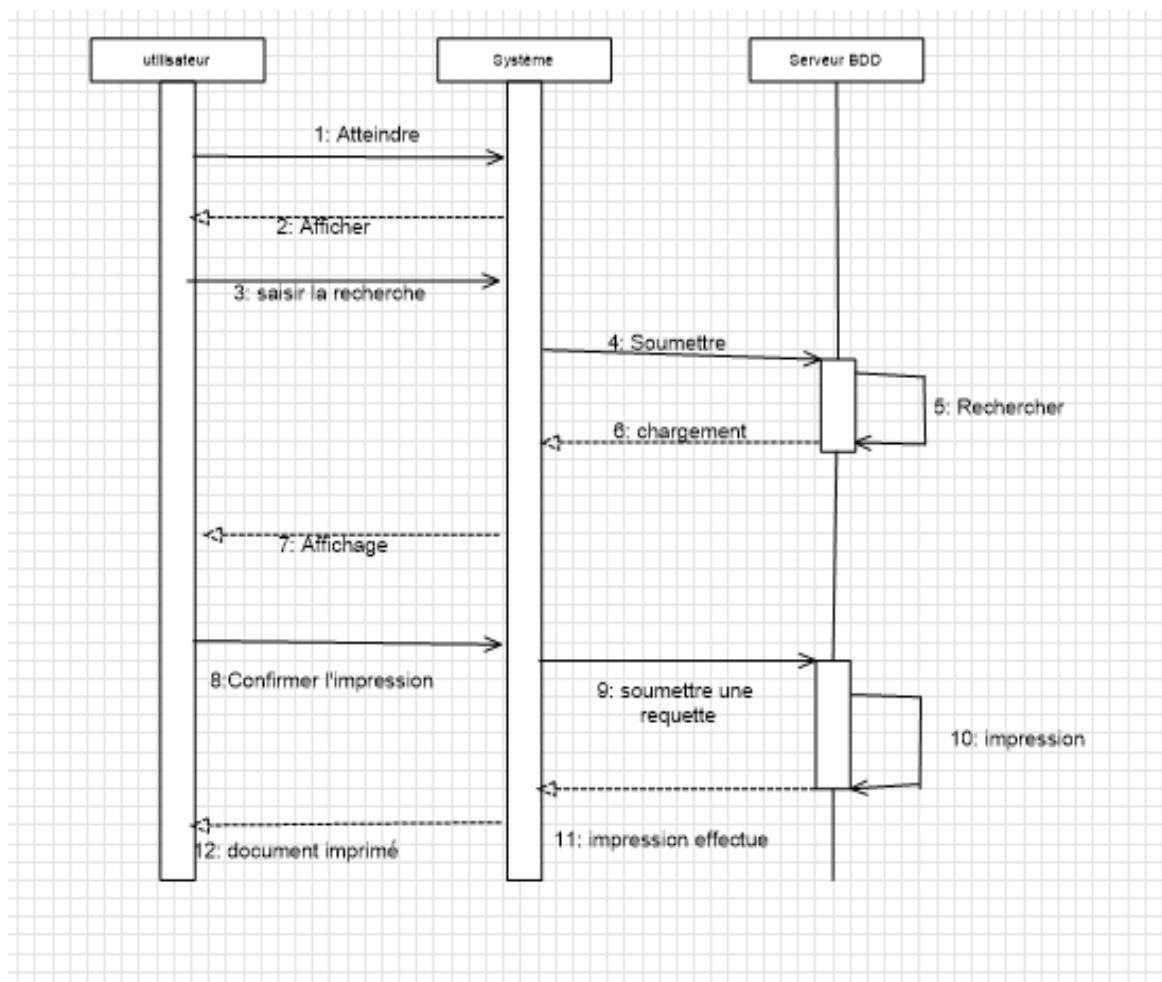


FIGURE 3.8 – Diagramme de séquence « imprimer »

1 : Atteindre la barre de recherche.

2 : L'application affiche la barre.

3 : Le médecin saisie une recherche.

4 : Le système soumet une requête à la BDD.

5 : Une recherche se fait au niveau de la BDD.

6 : chargement du résultat de la recherche.

7 : Afficher le formulaire à imprimer.

8 : L'agent de saisie confirme l'impression (appuie sur imprimer).

9 : Le système soumet une requête d'impression.

10 : L'impression s'effectue dans la BDD.

11 : Impression terminé.

12 : Confirmation d'impression.

3.3 Diagramme d'activité

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Le but de ce diagramme est de mettre en évidence les contraintes de séquentialité et de parallélisme qui pèsent sur la tache globale.

3.3.1 Diagramme d'activité « authentifier » :

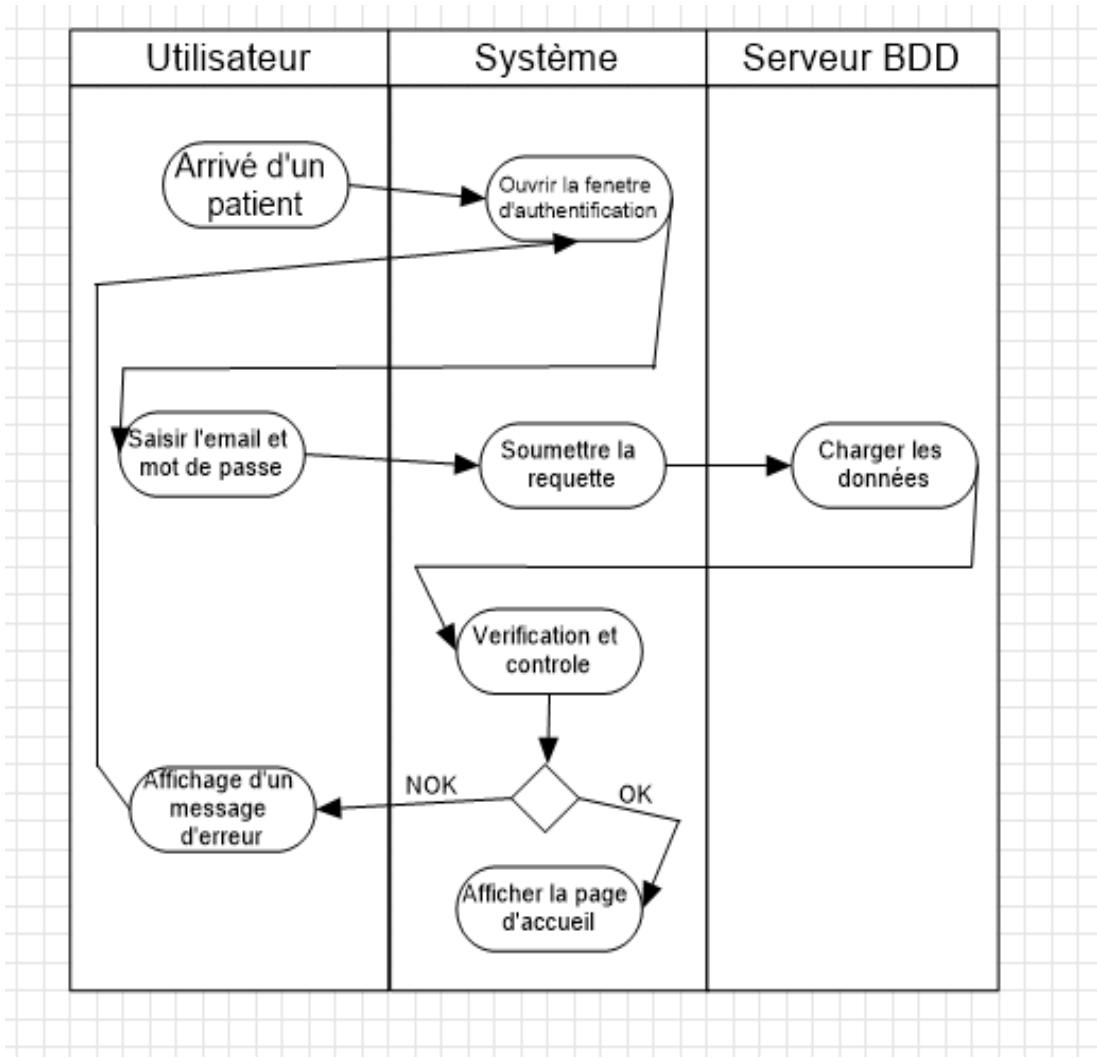


FIGURE 3.9 – Diagramme d'activité « authentifier »

3.3.2 Diagramme d'activité « ajouter » :

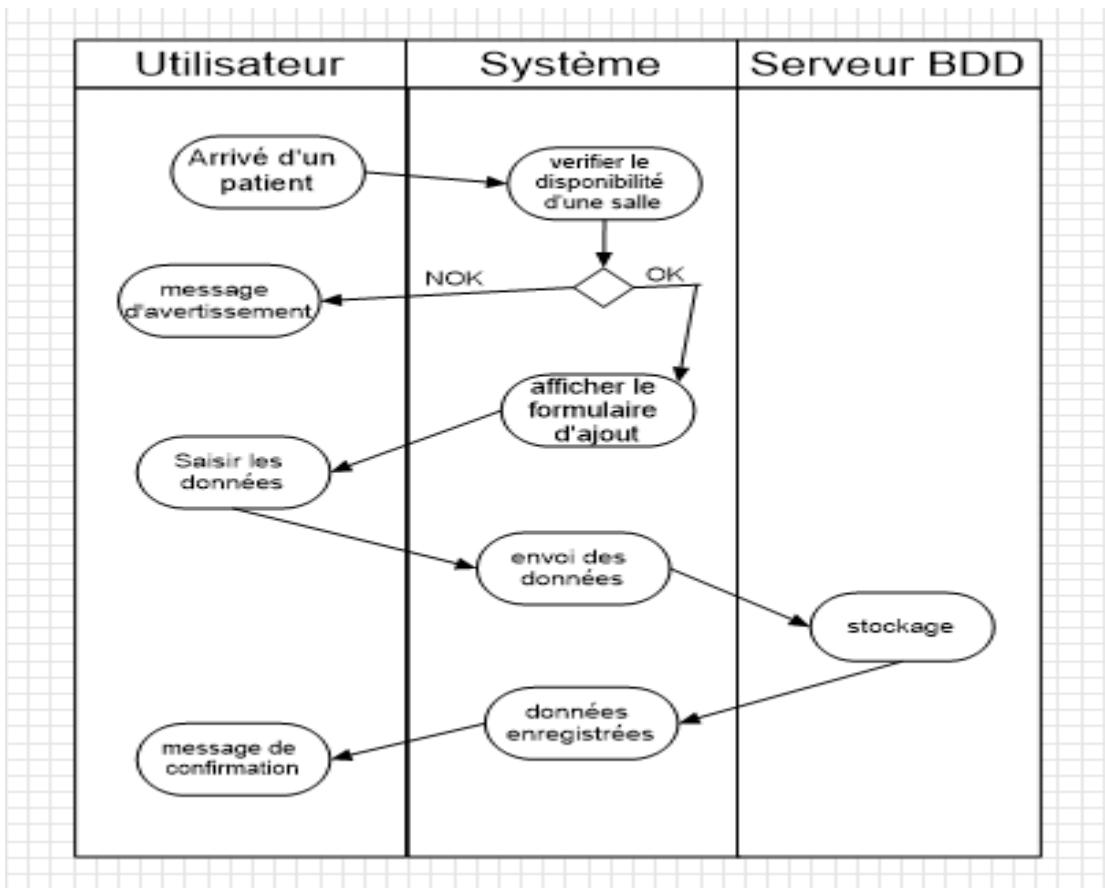


FIGURE 3.10 – Diagramme d'activité « ajouter »

3.3.3 Diagramme d'activité « modifier » :

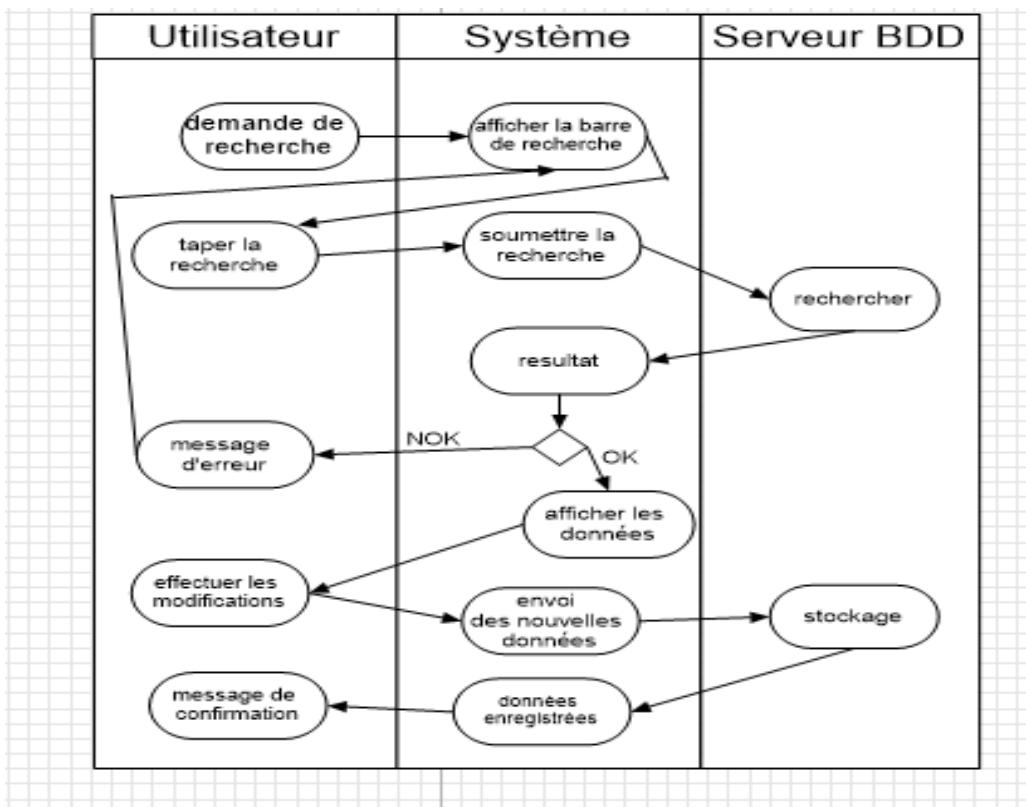


FIGURE 3.11 – Diagramme d'activité « modifier »

3.3.4 Diagramme d'activité « archiver » :

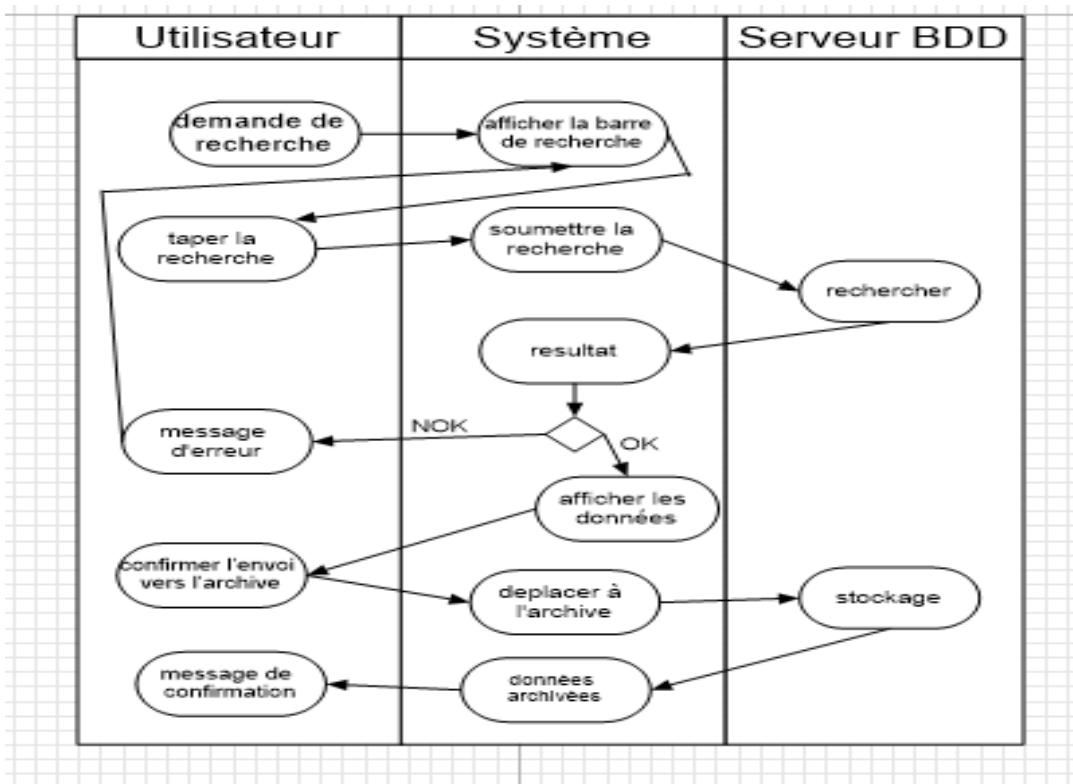


FIGURE 3.12 – Diagramme d'activité « archiver »

3.3.5 Diagramme d'activité « rechercher » :

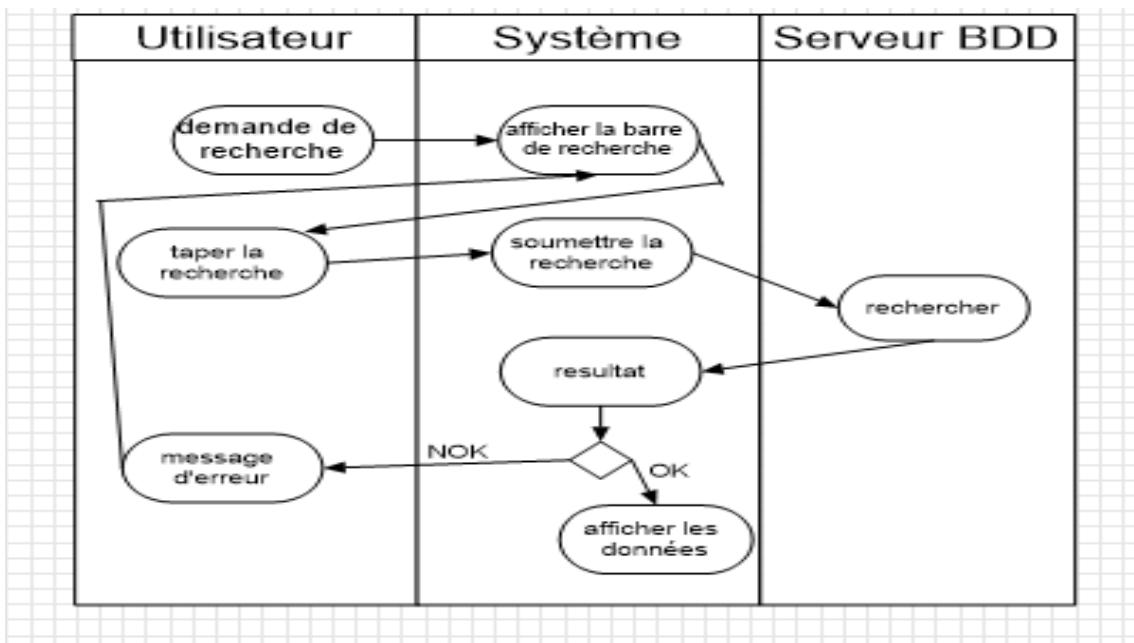


FIGURE 3.13 – Diagramme d'activité « rechercher »

3.3.6 Diagramme d'activité « remplir la fiche de suivi » :

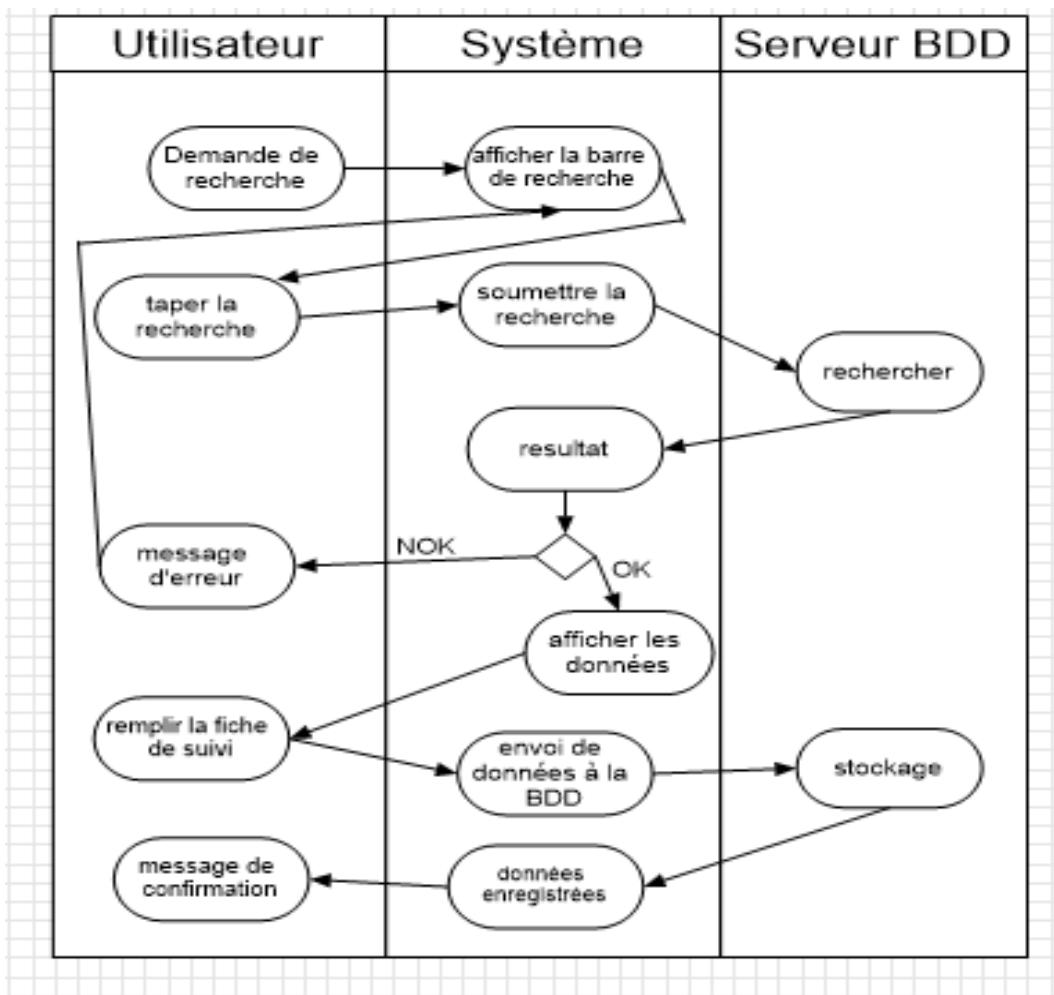


FIGURE 3.14 – Diagramme d'activité « remplir la fiche de suivi »

3.3.7 Diagramme d'activité « imprimer » :

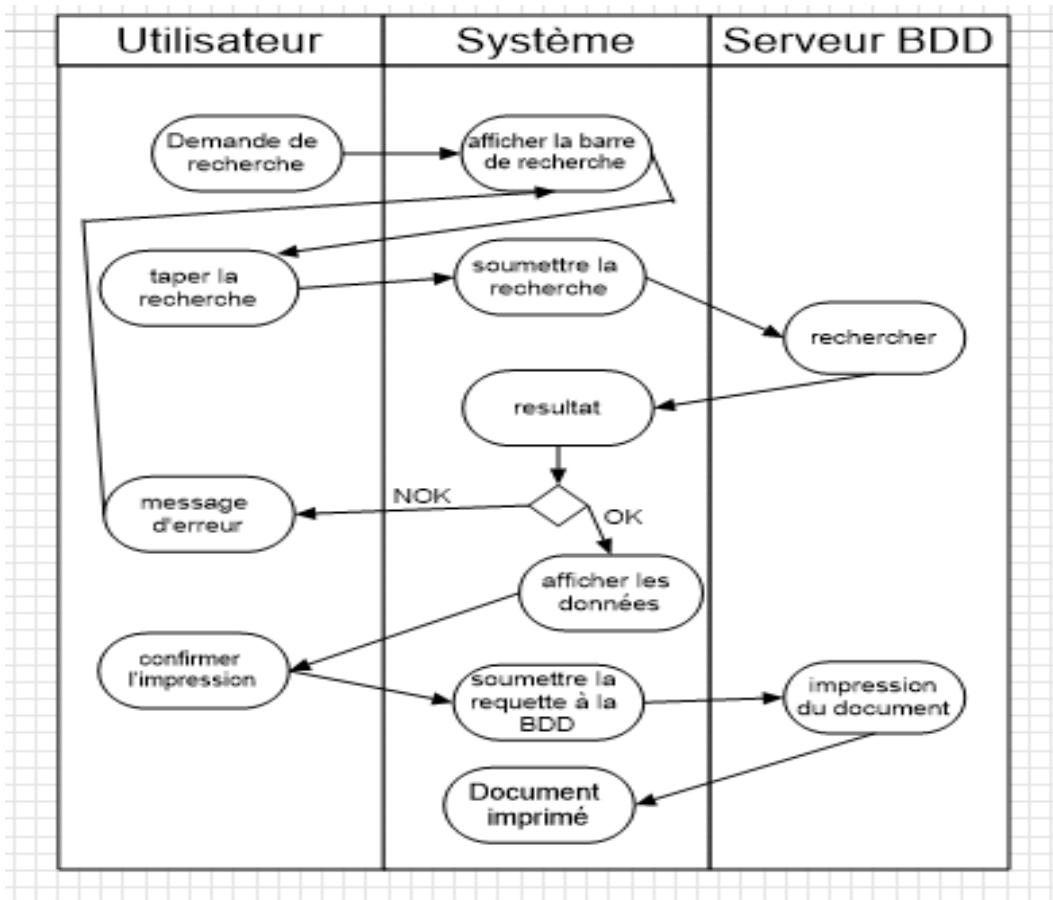


FIGURE 3.15 – Diagramme d'activité « imprimer »

Conclusion

Ce chapitre nous a permis de concevoir un système pour une gestion du patient dans un établissement hospitalier en utilisant le formalisme de modélisation UML .

Chapitre

4

Conception

Introduction

L'étude conceptuelle est un processus qui consiste à définir des principes de gestion. Elle se définit généralement comme une étude de premier niveau et d'évaluation préliminaire d'un projet. Une étude conceptuelle a pour but de développer les caractéristiques essentielles et de décrire les principes des solutions indépendamment de moyens utilisés. Une étude conceptuelle peut être entreprise sur plusieurs variantes d'un projet pour identifier les scénarios potentiels et le temps requis pour entreprendre des études plus poussées.

4.1 Diagramme de classe

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre elles. L'intérêt du diagramme de classe est de modéliser les entités du système d'information.

4.1. Définition des concepts de base

Une **classe** est une description abstraite (condensée) d'un ensemble d'objet du domaine de l'application : elle définit leur structure, leur comportement et leur relations.

Une **propriété** peut être définie comme étant une information élémentaire permettant l'identification d'un objet ou d'une entité.

Un **identifiant** ou une **propriété clé** est une propriété particulière permettant de caractériser de façon unique une entité ou un objet.

Une propriété clé ou identifiant peut être primaire ou secondaire (étrangère) :

- Elle est primaire lorsqu'elle répond aux critères d'unicité et de non nullité.
- Elle est secondaire lorsqu'elle utilisée dans une autre entité pour référence à une entité ou elle est définie comme clé primaire.

Une **association** est un lien sémantique qui peut exister entre deux entités c'est-à-dire c'est le lien qui permet de mettre en relation deux entités en donnant l'ensemble de caractéristiques du système.

Une **cardinalité** est une modélisation des participations minimum et maximum d'une entité ou un objet. Une cardinalité permet de caractériser le lien qui existe entre une relation et la relation à laquelle elle est située. Elle est composée d'un couple comportant une borne maximale (1 ou n : décrit le nombre maximum de fois qu'une entité peut participer à une relation) et une borne minimale (0 ou 1 : décrit le nombre minimum de fois qu'une entité peut participer à une relation).

4.1.2 Identification de classes

Les classes de notre application sont les suivantes :

Patient : c'est le noyau de l'application.

Admission : c'est la façon dont le patient est orienté vers un service précis selon son mode d'entrée à l'hôpital (consultation, hospitalisation, naissance, ...).

Sortie : Cette classe détermine la façon avec laquelle le patient va quitter l'hôpital (décès, évacuation, rétablissement, ...)

Salle/Service : elle détermine le lieu où séjournera le patient durant son hospitalisation.

Garde-patient : c'est la personne accompagnant le patient tout au long de son séjour.

Médecin : cette classe représente les personnes chargées de suivre le patient durant son hospitalisation (traitement, diagnostic, ...).

Traitements : elle constitue les traitements effectués sur le patient.

Naissance : c'est la classe des nouveaux nés.

4.1.3 Réalisation du diagramme de classe

Règles de gestion

- Un patient peut avoir un type d'admission et une admission peut ne pas avoir ou avoir plusieurs patients.
- Un patient est affecté à une salle qui se trouve dans un service déterminé et une salle peut ne pas contenir ou contenir plusieurs patients.
- Chaque patient peut être gardé par un seul garde-patient et un garde-patient peut ne pas garder ou garder plusieurs patients.
- Un patient peut avoir plusieurs naissances et une naissance concerne un seul patient.
- Un médecin suit un ou plusieurs patients et un patient peut être suivi par un ou plusieurs médecins.
- Un médecin est affecté à un seul service et un service contient un ou plusieurs services.
- Un patient prend un ou plusieurs traitements et un traitement est pris par aucun ou plusieurs patients.
- Un patient doit sortir par un seul mode de sortie et un mode de sortie peut ne pas avoir ou avoir plusieurs patients.

Dictionnaire de données

Codification	Désignation	Type	Observation
Cod-patient	Le code du patient	N	
Nom-patient	Le nom du patient	A	
Prenom-patient	Le prénom du patient	A	
Sexe-patient	Le sexe du patient	A	
Date-naissance	La date de naissance du patient	N	
Lieu-naissance	Le lieu de naissance du patient	A	
Adr-patient	L'adresse du patient	A	
Contact	Le contact du patient	N	
Cod-médecin	Le code du médecin	N	
Nom-médecin	Le nom du médecin	A	
Prenom-médecin	Le prénom du médecin	A	
Specialité-méd	La spécialité du médecin	A	
Cod-admission	Le code de	N	

	l'admission		
Type-adm	Le type de l'admission	A	
Date-adm	La date de l'admission	N	
Heure-adm	L'heure de l'admission	AN	
Cod-sortie	Le code de la sortie	N	
Type-sortie	Le type de la sortie	A	
Date-sortie	La date de la sortie	N	
Heure-sortie	L'heure de la sortie	AN	
Code-traitement	Le code du traitement	N	
Type-traitement	Le type du traitement	A	
Durée-traitement	La durée du traitement	AN	
Cod-naissance	Le code de la naissance	N	
Date-naissance	La date de la naissance	N	
Heure-n	L'heure de la naissance	AN	
Sexe-né	Le sexe du nouveau-né	A	
Poids-né	Le poids du nouveau-né	AN	
Prenom-né	Le prénom du nouveau-né	A	
Cod-gp	Le code du garde-patient	N	
Nom-gp	Le nom du garde-patient	A	
Prenom-gp	Le prénom du garde-patient	A	
Adr-gp	L'adresse du garde-patient	A	
Num-salle	Le numéro de la salle	N	
Cod-service	Le code du service	N	
Nom-service	Le nom du service	A	

N : numérique

A : alphabétique

AN : alphanumérique

Le diagramme de classe :

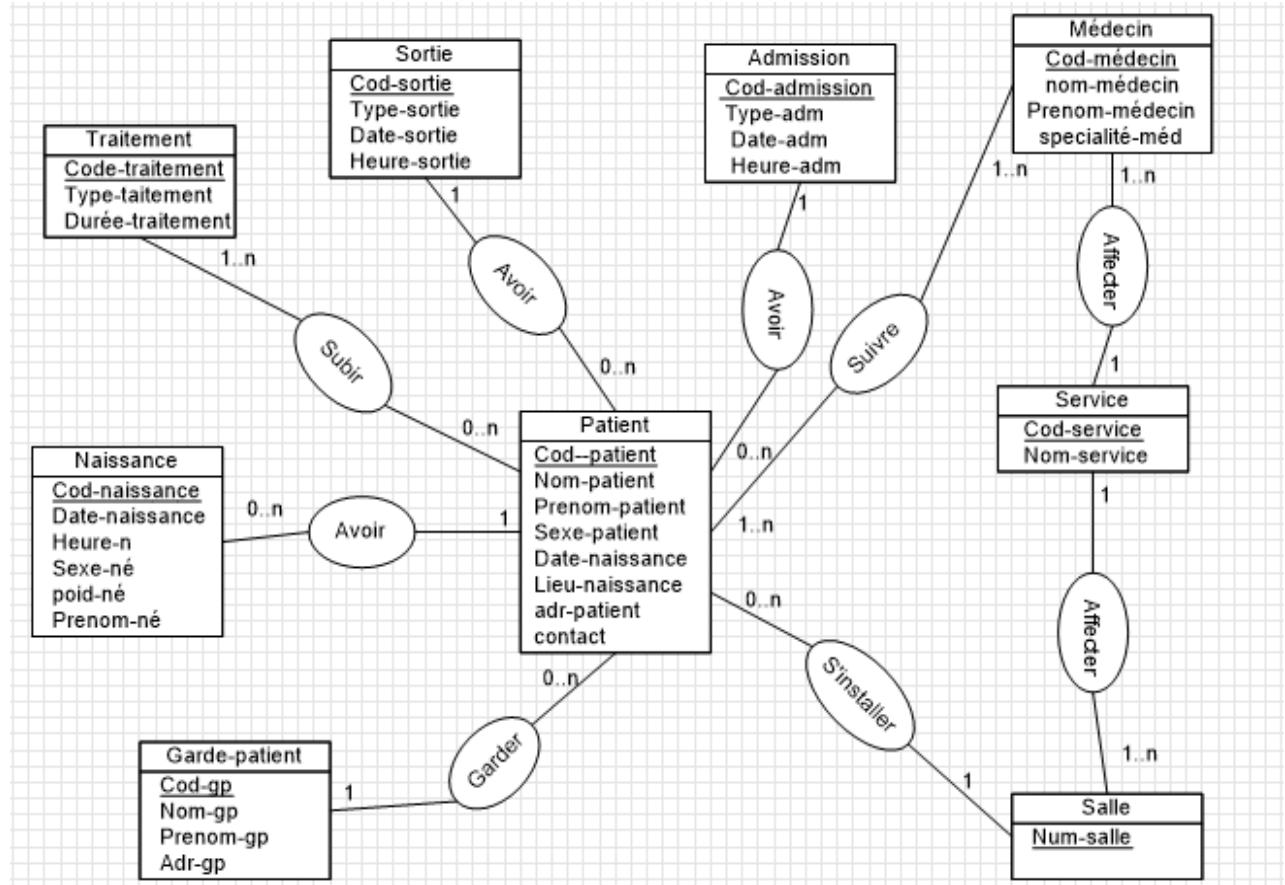


FIGURE 4.1 – Diagramme de classe

4.2 Modèle relationnel de données

4.2.1 Quelques définitions

- Une table est un tableau composé de lignes et de colonnes.
- Une ligne est appelée enregistrement.
- Une colonne est appelée champ.

- A l'insertion d'une ligne ou d'une colonne on a une valeur.
- Clé : on distingue deux types de clés :
 - Clé primaire : elle identifie d'une manière unique chaque enregistrement de la table.
 - Clé étrangère : elle matérialise une liaison avec une autre table, elle fait référence à une clé primaire d'une autre table.

4.2.2 Règles de passage

- Toute entité donne naissance à une table dont la clé primaire est l'identifiant de l'entité.
- Toutes les associations binaires de type (x..n) et (x..1), l'identifiant de l'entité avec la cardinalité (x..1) reçoit l'identifiant de l'entité (x..n).
- Les associations binaires de type (x..n) et (x..n) deviennent une table supplémentaire comportant les identifiants des deux autres entités.

Dans ce qui suit nous avons souligné les clés primaires et nous avons mis un « # » avant les clés étrangères.

Notre schéma relationnel :

Patient(Cod-patient, Nom-patient, Prenom-patient, Sexe-patient, Date-naissance, Lieu-naissance, Adr-patient, Contact, #Code-naissance)

Suivre(#Cod-patient, #Cod-médecin)

Subir(#Cod-patient, #Code-traitement)

Medecin(Cod-médecin, Nom-médecin, Prenom-médecin, Specialité-méd)

Admission(Cod-admission, Type-adm, Date-adm, Heure-adm, #cod-patient)

Sortie(Cod-sortie, Type-sortie, Date-sortie, Heure-sortie, #cod-patient)

Traitement(Code-traitement, Type-traitement, Durée-traitement)

Naissance(Cod-naissance, Date-naissance, Heure-n, Sexe-né, Poid-né, Prenom-né)

Garde-patient(Cod-gp, Nom-gp, Prenom-gp, Adr-gp, #Cod-patient)

Salle(Num-salle, #Cod-patient)

Service(Cod-service , Nom-service, #Cod-médecin, #Num-salle)

Conclusion

Dans ce chapitre, nous avons réalisé une étude conceptuelle de notre système et nous avons pu représenter les différentes relations entre les acteurs du système grâce un diagramme de classe duquel nous avons réalisé un modèle relationnel afin de pouvoir l'implémenter.

Chapitre

5

Réalisation

Introduction

A ce stade du processus les cas d'utilisation sont déterminés, le problème a été analysé en profondeur et nous avons défini une conception la plus appropriée aux besoins de l'application. Nous pouvons alors entreprendre la réalisation. Nous avons utilisé des outils de développement comme le serveur xampp et phpMyAdmin pour la base données ainsi les langages de développements web HTML, CSS, Java script et PHP que nous avons implémenté via l'éditeur de textes « visual studio code ».

5.1 Outils de développement

5.1.1 définitions

Xampp : c'est un ensemble de logiciels permettant de mettre en place un serveur web local

phpMyAdmin : c'est une application web de gestion pour les systèmes de gestion de base de données MySQL, réalisé principalement en PHP.

HTML : « Hyper Text Markup » qu'on peut traduire langage de balise pour l'hypertexte, c'est un langage conçu pour représenter et structurer les pages web.

CSS : « cascading style sheets » ou bien les feuilles de style en cascade, c'est un langage informatique qui décrit la présentation et qui met du style aux documents HTML et XML.

Java script: c'est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web, java script est au cœur des langages utilisés par les développeurs web. Une grande majorité des sites web l'utilisent.

PHP: « Hypertext Preprocessor », est un langage de programmation libre, principalement utilisé pour produire des pages web dynamiques via un serveur http, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale .PHP est un langage impératif orienté objet.

Visual studio code: c'est un éditeur de code extensible développé par Microsoft pour Windows, linux et macOS. Il peut être utilisé avec une variété de langages de programmation, notamment java, java script, HTML, CSS, PHP....

5.2 Description de l'application

L'application que nous avons conçu permet d'effectuer les tâches suivantes :

- **Les admissions :**
 - De patient
- **Les sorties :**
 - De patients
- **L'affichage :**
 - Des patients
 - Des médecins
 - Des gardes patients
 - Agent de saisie
- **Les statistiques :**
 - Nombre de patients hospitalisés
 - Nombre de patients archivés
 - Nombre de naissances
 - Nombre de décès
- **Les imprimés :**
 - Bulletin d'admission

5.3 Les interfaces de l'application

5.3.1 La fenêtre d'authentification

Au lancement de notre application, une fenêtre s'affiche à l'écran qui demande d'introduire l'email et le mot de passe pour se connecter et accéder au menu principal en tant qu'admin ou médecin ou agent de saisie.

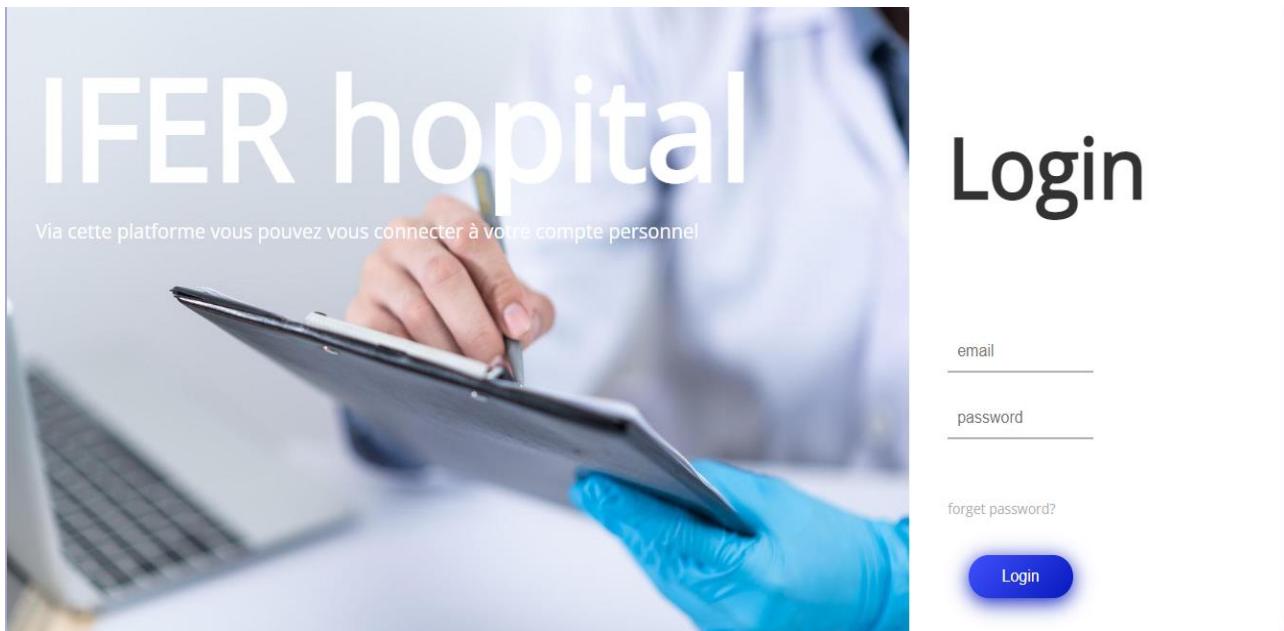


FIGURE 5.1 – Fenêtre d'authentification

5.3.2 Le menu principal

Selon le mode d'accès validé, une fenêtre s'affiche, elle comporte le menu principal ou l'utilisateur (admin, médecin ou agent de saisie) pourra sélectionner la tâche à effectuer.

En mode admin, la fenêtre principale comporte trois bouton « Médecins », « Agent de saisie » et « Garde patients » .

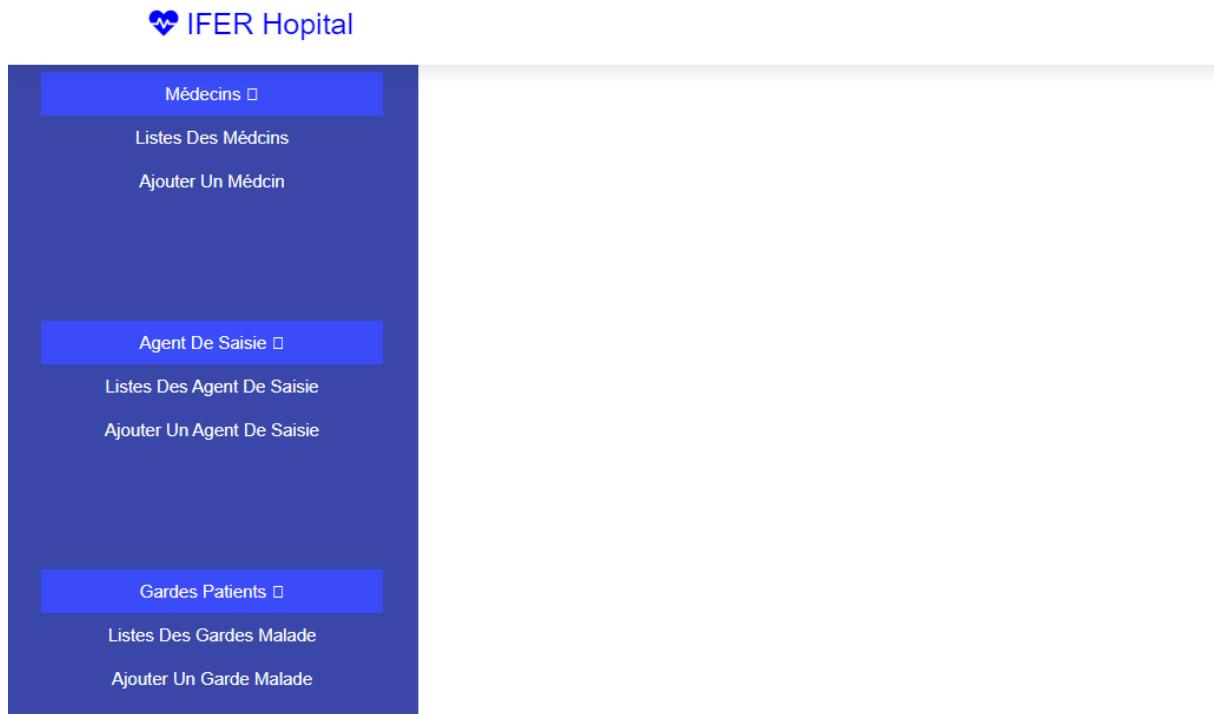


FIGURE 5.2 – menu principale en mode admin

Médecins

- Listes Des Médecins
- Ajouter Un Médecin

Agent De Saisie

- Listes Des Agent De Saisie
- Ajouter Un Agent De Saisie

	CODE Médecin	NOM	Prénom	Spécialité	
1		Raymond REDINCTON	Tizi Ouzou	Cardiologie	details
2		Elisabette Keane	Bejaia	Pneumologie	details
3		Walter White	Bejaia	Cardiologie	details
4		Pink Man	Tizi Ouzou	Phsycologie	details

FIGURE 5.3 – liste des médecins

Ajout De Médecin

Code De Médecin

Nom

Adresse

Spécialité

[ajouter](#)

FIGURE 5.4 – ajouter un médecin

En mode médecin, la fenêtre principale comporte un bouton « Liste des patients » auquel il peut rechercher un patient et des informations le concernant.

CODE PATIENT	NOM ET PRENOM	DATE DE NAISSANCE	SEX	LIEU DE NAISSANCE	ADRESSE	CONTACT	Fiche De Suivi

FIGURE 5.5 – menu principal en mode médecin

En mode agent de saisie, la fenêtre principale comporte un bouton « patients » qui permet de consulter la liste des patients ou ajouter un patient.

The screenshot shows a web-based application interface. On the left, there is a sidebar with a blue header containing a heart icon and the text 'JFER Hopital'. Below the header, the sidebar has two items: 'Patients' and 'Listes Des Patients'. The main area is titled 'Ajout D'un Agent De Saisie' (Add Agent). It contains several input fields: 'Code Patient' (with a placeholder 'Code Agent'), 'Nom' (placeholder 'Nom Et Prenom'), 'Sexe' (placeholder 'Sexe'), 'Date De Naissance' (placeholder 'Jj/MM/AAAA' with a calendar icon), 'Lieu De Naissance' (placeholder 'Lieu De Naissance'), and 'Adresse'.

FIGURE 5.6 – menu principal en mode agent de saisie

5.4 Exemple de codes :

HTML :

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <title>Login Page in HTML with CSS Code Example</title>
    <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">

    <link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">
    <link rel="stylesheet" href="stylelog.css">

</head>

<body>
    <!-- partial:index.partial.html -->

    <div class="box-form">
        <div class="left">
            <div class="overlay">
                <h1>JFER hospital</h1>
                <p>Via cette plateforme vous pouvez vous connecter à votre compte personnel </p>
            </div>
        </div>

        <div class="right">
            <h5>Login</h5>

            <div class="inputs">
                <input type="text" placeholder="email">
                <br>
                <input type="password" placeholder="password">
            </div>
        </div>
    </div>
</body>
```

FIGURE 5.7 – exemple HTML

CSS :

```
1  body {
2      display: flex;
3      flex-wrap: wrap;
4      gap: 0px;
5      font-family: 'Poppins', sans-serif;
6  }
7
8  * {
9      font-family: 'Poppins', sans-serif;
10     margin: 0;
11     padding: 0;
12     box-sizing: border-box;
13     outline: none;
14     border: none;
15     text-transform: capitalize;
16     transition: all .2s ease-out;
17     text-decoration: none;
18 }
19
20 html {
21     font-size: 62.5%;
22     overflow-x: hidden;
23     scroll-padding-top: 7rem;
24     scroll-behavior: smooth;
25 }
26
27 .heading {
28     text-align: center;
29     padding-bottom: 2rem;
30     text-shadow: var(--text-shadow);
31     text-transform: uppercase;
32     color: var(--black);
33     font-size: 1.5em;
```

FIGURE 5.8 – exemple CSS

PHP :

```
<?php
$servername = "localhost";
$database = "hopital";
$username = "root";
$password = "";

// Create connection

$connection = mysqli_connect($servername, $username, $password, $database);

// Check connection

if ($connection->connect_error) {
die("Connection failed: " . $connection->connect_error);
}

$getmed = "SELECT * FROM medecin ;";
$result= mysqli_query($connection, $getmed);
if (mysqli_num_rows($result) > 0)
{
    // OUTPUT DATA OF EACH ROW
    while($row = mysqli_fetch_assoc($result))
    {
        echo" <tr> ";
        echo"     <td>$row[id]</td>";
        echo"     <td>$row[nomPrenom]</td>";
        echo"     <td>$row[adresse]</td>";
        echo"     <td><button>details</button></td>";
        echo"   </tr>";
    }
}
```

FIGURE 5.9 – exemple PHP

Conclusion :

Dans ce chapitre, on a décrit brièvement le processus de réalisation de notre application de suivi des patients en spécifiant l'environnement de développement et la démarche suivie pour la réalisation.



Conclusion et perspectives

Le travail exposé dans ce rapport s'intéresse au suivi des patients au sein de l'établissement hospitalier « IFER ».

On a fait beaucoup de recherches et on a travaillé avec beaucoup de dévouement pour réaliser ce travail et atteindre cet objectif.

Nous tenons à remercier notre enseignante qui nous a guidé tout au long du travail et tous ceux qui ont attribué à faire ce travail.

Nous espérons que cette modeste contribution sera d'un apport certain à tous ceux qui auront un jour à l'utiliser.



Références

- [1] Olivier sigaud. Introduction à la modélisation orienté objet avec UML
- [2] M.Laurent AUDIBERT. Base de données et langage SQL.
- [3] Johan Sievering. Les cahiers du programmeur juin 2010
- [4] Bernard Guisiano et Veronique Lidoray. Cours fondamentaux février 2008
- [5] <http://fr.wikipedia.org/> HTML, CSS, Java script, PHP