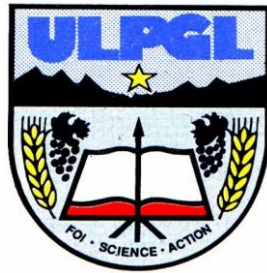


RÉPUBLIQUE DÉMOCRATIQUE DU CONGO  
ENSEIGNEMENT SUPÉRIEUR ET UNIVERSITAIRE  
UNIVERSITÉ LIBRE DE PAYS DE GRANDS LACS  
U.L.P.G.L  
B.P 368 /Goma



**CONCEPTION D'UN SYSTEME DE GESTION INFORMATISEE  
DES PATIENTS DANS UN HOPITAL**

Par KAKASU KANDANDA Christian  
En vue de l'obtention du diplôme d'ingénieur A1  
en Faculté des Sciences et Technologies Appliquées  
Génie Electrique-Informatique.

Directeur : CT. Ir Antoine KAYISU  
Encadreur : Ass.Ir Henri SHABANI

**Année Académique : 2014-2015**

## DÉDICACE

A mes très chers parents Jean-Paul KANDANDA et Chantal KANKU,  
A tous ceux qui me sont chers.

KAKASU KANDANDA Christian

## REMERCIEMENTS

Le présent travail sanctionne la fin de notre premier cycle à l'université, dans la faculté des sciences et technologies appliquées. Il est un produit de la conjugaison des efforts et de l'assistance de plusieurs personnes. Ainsi, qu'elles daignent trouver à travers ce travail l'expression de notre profonde gratitude. Nos remerciements sont adressés à tous ceux qui, de loin ou de près ont contribué à la réalisation de ce travail.

Nous rendons premièrement grâce à l'Eternel Dieu pour tous ses bienfaits à notre endroit. Aux corps académique et administratif de l'Université Libre des Pays des Grands Lacs (ULPGL), qu'ils veuillent accepter nos très sincères remerciements pour tout leur travail appréciable accompli pour notre encadrement tout au long de notre premier cycle.

Nous témoignons particulièrement notre profonde gratitude au Chef des Travaux (CT) Antoine KAYISU et à l'Assistant Henri SHABANI qui, en dépit de leurs multiples occupations, ont accepté de diriger ce travail avec une sollicitude paternelle et un souci du travail bien fait.

Que mes parents trouvent en ce travail notre reconnaissance indéfectible pour tout ce qu'ils représentent pour nous. Ils n'ont cessé de céder à nos caprices et répondre aux exigences qu'a imposé notre formation pour ce premier cycle afin que nous soyons utiles ; ce travail est le fruit de leur sueur et des sacrifices qu'ils continuent à consentir pour notre formation.

Nous remercions aussi certains aînés, amis, camarades et collègues qui nous ont apporté leur soutien tant moral, matériel que spirituel au cours de notre parcours, à savoir MURHABAZI Espoir, RUBAMBURA Daniel, WABO Marcelin, DJANI Elie, Joviane UDJU,....

La liste étant longue, il ne nous est pas facile d'énumérer tous ceux qui ont apporté leur pierre pour que finalement nous arrivions à produire ce travail. Qu'ils daignent trouver tous à travers ce travail l'expression de nos remerciements cordiaux.

KAKASU KANDANDA Christian

## SIGLES ET ABRÉVIATIONS

**SGBD** : Système de gestion de base des données.

**BD** : Base des données

**IDE** : Environnement de Développement Intégré

**LDD** : Langage de description des données

**LMD** : Langage de description des données

**LCD** : Langage de contrôle des données

**LCT** : Langage de contrôle de Transaction

**JVM** : Machine Virtuelle Java

**UML** : Unified Modeling Language

**SQL** : Structured Query Language (Langage des requêtes structurées)

## Table des Matières

### Contenu

<b>DÉDICACE</b> .....	<b>1</b>
<b>REMERCIEMENTS</b> .....	<b>2</b>
<b>SIGLES ET ABRÉVIATIONS</b> .....	<b>3</b>
<b>Table des Matières</b> .....	<b>4</b>
<b>CHAPITRE 1 : INTRODUCTION GÉNÉRALE</b> .....	<b>6</b>
<b>1.1. Problématique</b> .....	<b>6</b>
<b>1.2. Hypothèses de recherche</b> .....	<b>6</b>
<b>1.3. Choix et intérêt de l'étude</b> .....	<b>7</b>
<b>1.4. Délimitation spatio-temporelle du Travail</b> .....	<b>7</b>
<b>1.5. Méthodologie du travail</b> .....	<b>7</b>
<b>1.6. Subdivision du travail</b> .....	<b>7</b>
<b>CHAPITRE 2 : REVUE LITTÉRAIRE</b> .....	<b>8</b>
<b>2.1 Introduction</b> .....	<b>8</b>
<b>2.2 Information</b> .....	<b>8</b>
<b>2.3 Base des données</b> .....	<b>9</b>
<b>2.4 Système de gestion de base de données</b> .....	<b>9</b>
<b>2.4.1 Langage SQL</b> .....	<b>11</b>
<b>2.5 Langages de programmation</b> .....	<b>11</b>
<b>2.5.1 Éléments du langage de programmation</b> .....	<b>12</b>
<b>2.5.2 Les paradigmes</b> .....	<b>12</b>
<b>2.5.3 Le langage JAVA</b> .....	<b>13</b>
<b>2.6 Environnements de développement</b> .....	<b>13</b>
<b>2.7 Architecture</b> .....	<b>14</b>
<b>CHAPITRE 3 : MÉTHODOLOGIE DE RECHERCHE</b> .....	<b>15</b>
<b>3.1 Définitions</b> .....	<b>16</b>
<b>3.2 Choix et intérêt des méthodes et techniques</b> .....	<b>16</b>
<b>CHAPITRE 4 : COLLECTE DES DONNÉES, ANALYSE ET DISCUSSION DES</b> <b>RÉSULTATS.</b> .....	<b>Erreur ! Signet non défini.</b>
<b>4.1 Population et échantillon de l'étude</b> .....	<b>17</b>
<b>4.2 Résultats de l'interview</b> .....	<b>19</b>
<b>CHAPITRE 5 : ÉTUDE DES CAS</b> .....	<b>20</b>
<b>5.1 Étude du système existant</b> .....	<b>20</b>
<b>5.2 Contexte du système</b> .....	<b>20</b>
<b>5.3 Circuit de déroulement</b> .....	<b>21</b>
<b>5.3.1 Condition d'entrée dans le système</b> .....	<b>21</b>
<b>5.3.2 Condition d'admission dans différents services</b> .....	<b>21</b>
<b>5.3.2 Condition de sortie du système</b> .....	<b>21</b>
<b>5.3.3 Identification des lacunes du système existant</b> .....	<b>21</b>
<b>5.4 Proposition</b> .....	<b>22</b>
<b>5.5 MCC du système d'information actuel</b> .....	<b>22</b>
<b>5.5.1 Identification des acteurs</b> .....	<b>22</b>
<b>5.5.2 Schéma du MCC du système d'information</b> .....	<b>23</b>
<b>5.5.3 Schéma du MCC du système Informatique</b> .....	<b>23</b>
<b>5.5.4 Liste de flux</b> .....	<b>24</b>

---

5.5.6	Modèle Organisationnel de Traitement. ....	25
5.5.7	Tableaux des procédures fonctionnelles. ....	25
5.5.7.1	Processus de réception .....	25
5.5.7.2	Tableaux de déroulement des procédures fonctionnelles.....	26
5.6	Conception et modélisation du système informatique.....	33
5.6.1.	Modèle Conceptuel des Données. ....	33
5.6.2	Graphe des dépendances fonctionnelles.....	34
5.7	Réalisation .....	34
5.7.1	Diagramme de cas d'utilisation .....	35
5.7.3	Schéma fonctionnel des interfaces.....	36
<b>CHAPITRE 6 : CONCLUSION.....</b>		<b>40</b>
<b>Bibliographie .....</b>		<b>41</b>
<b>Annexe : Code Java implémentant les classes du logiciel .....</b>		<b>42</b>

## CHAPITRE 1 : INTRODUCTION GÉNÉRALE

### 1.1. Problématique

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs, et cela grâce à l'informatique qui est une science étudiant les techniques du traitement automatique de l'information. Depuis toujours, la gestion de l'organisation pose de sérieux problèmes de surveillance relatifs à la saisie, à la collecte, au traitement et à la diffusion des informations.

Avant l'invention de l'ordinateur, on enregistrait toutes les informations manuellement sur des supports en papier ; ce qui engendrait beaucoup de problèmes tel que la perte considérable de temps dans la recherche de ces informations ou la dégradation de ces dernières, etc. ...

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de données des entreprises ; c'est aujourd'hui la partie essentielle dans leur développement.

Les cliniques médicales font partie intégrante des établissements que l'informatique pourra beaucoup aider. Jusqu'à ce jour, la gestion manuelle de l'information y est encore dominante ; d'où la nécessité d'y introduire l'informatique.

Considérant les principaux besoins à couvrir au sein des cliniques médicales, une tendance se dessine pour l'harmonisation des supports utilisés, ceci pour palier des problèmes fastidieux d'organisation et d'archivage des données. L'objectif du projet présenté dans ce travail est de concevoir et réaliser une application simple en vue d'informatiser les dossiers médicaux et gérer de patients dans une structure de santé.

Le système traditionnel (fiches des malades, registres de consultation, papiers écrits) ne peut, d'une manière automatique :

- Donner la liste des patients et leur suivi pendant une période donnée;
- Classer les patients par catégorie ;
- Produire le cas échéant une fiche du patient ;
- Sortir une ordonnance médicale et une facture de soins.

La plupart des gestionnaires se retrouvent alors face au défi consistant à mettre sur pied un système d'information prenant en charge les éléments ci-haut cités. Mais comment s'y prendre ? Quels outils et techniques doivent être utilisés ? Que dire de la configuration matérielle pouvant être adoptée pour ce système ? Voilà bien des questions....

Est-ce donc possible d'anticiper la solution d'automatisation pour résoudre ces différents problèmes.

### 1.2. Hypothèses de recherche

Étant donné que les hypothèses sont des réponses anticipées aux questions spécifiques d'une recherche, il sied d'évoquer les hypothèses ci-dessous, considérant que l'hypothèse relative à la question principale est que l'informatisation du système traditionnel actuel influencerait de manière considérable la clinique médicale dans le sens qu'elle lui ferait gagner en temps et en sécurité, lesquels sont par ailleurs parmi des éléments fondamentaux pour un bon fonctionnement d'une entreprise :

- L'informatisation aiderait la comptabilité dans le sens à lui éviter des calculs fastidieux et à produire des états financiers en peu de temps et à moindre coût ;
- Elle permettrait aussi l'organisation, la centralisation et la conservation de l'information ; ce

qui faciliterait l'accès rapide aux données et un gain de temps considérable.

### 1.3. Choix et intérêt de l'étude

Le temps est un des facteurs importants dans la productivité d'une entreprise. L'automatisation des tâches participe ainsi au gain du temps, et permet à l'entreprise d'être compétitive.

Théoriquement ce travail fera partie de la littérature et des travaux antérieurs qui serviront de base aux futurs chercheurs dans le domaine de la gestion informatique.

Pratiquement, il a pour objectif la création d'une application pouvant prendre en charge la gestion des patients dans une clinique médicale. Cette application devra globalement permettre l'enregistrement des données sur les patients et leur suivi, la génération de leurs fiches de consultation et de laboratoires, le suivi de caisse de consultation. Elle devra alors intégrer les services suivants : réception, triage, consultation et Laboratoire.

### 1.4. Délimitation spatio-temporelle du Travail

L'hôpital sur lequel se porte l'étude est l'hôpital militaire de Goma. Il comporte en son sein les services suivants :

1. Réception
2. Triage
3. Consultation
4. Laboratoire
5. Gynécologie obstétrique
6. Médecine interne
7. Pédiatrie
8. Pharmacie.

- a) En ce qui concerne l'hôtellerie, elle est constituée de : Salles communes ;
- b) Chambres privées ;
- c) Chambres semi-privées.

### 1.5. Méthodologie du travail

En vue d'atteindre les objectifs nous assignés pour ce travail, nous avons utilisé la comparaison et l'analyse du système de fonctionnement de la structure de santé comme méthodes. Les méthodes UML et MERISE nous ont fournis les outils indispensables pour la conception principalement de la base des données et des modèles qui vont permettre la réalisation de l'application. Les méthodes sus mentionnées ont été soutenues par les techniques suivantes de la revue documentaire nous a permis de cerner le problème et des interviews.

### 1.6. Subdivision du travail

Ce travail se subdivise en 6 chapitres qui sont : Introduction générale, revue littéraire, Méthodologie de recherche, Collecte des données- analyse et discussion des résultats, étude des cas et conclusion.

## CHAPITRE 2 : REVUE LITTÉRAIRE

### 2.1 Introduction

Ce chapitre porte sur la présentation de différents concepts et Technologies qui vont intervenir dans la suite de ce travail et pour la conception, la modélisation et l'implémentation du système qui sera utilisé pour la réalisation de l'application.

### 2.2 Information

Au sens étymologique, l'information est ce qui donne forme à l'esprit. Elle vient du verbe latin *informare*, qui signifie «donner forme à» ou «se former une idée de».

L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire ; elle utilise un code de signes porteurs de sens tels qu'un alphabet de lettres, une base de chiffres, des idéogrammes ou pictogrammes. Hors contexte, elle représente le véhicule des données comme dans la théorie de l'information ; et, hors support, elle représente un facteur d'organisation<sup>1</sup>.

Une information est un élément qui permet de compléter la connaissance sur une personne, un objet, un événement. C'est aussi un ensemble d'éléments de connaissance susceptibles d'être codés pour être conservés, traités ou communiqués.

Elle est aussi l'émission ou la réception de signaux oraux, ou écrits, sonores, visuels ou multimédia dont le but est de déclencher les processus alimentant l'échange, base naturelle et indispensable de l'animation de l'organisation.

Dans certains cas, un ensemble d'informations peut se référer à un domaine précis ; cela constitue alors ce qu'on appelle un système d'informations.

La plupart de systèmes actuels se basent sur la technologie de l'informatique. Cependant il existe encore de systèmes dans lesquels les informations sont stockées, manipulées et communiquées traditionnellement ; on peut citer parmi eux : les armoires, les fiches en papier ou les registres.

Il existe une différence entre information et données comme cela peut être illustré par l'exemple suivant :

*Dans une librairie, un client demande au vendeur si le livre " Apprendre à développer un site web avec PHP et MySQL (Livre de Olivier Rollet) est disponible en stock. Le vendeur consulte la base de données de la librairie à l'aide de son ordinateur et confirme au client que le livre est disponible. Le vendeur a donc donné au client l'information que le livre est en stock. Afin de pouvoir donner cette information, le vendeur a dû consulter les données qui représentent le stock de la librairie. Le fait de consulter le stock constitue un traitement sur les données du stock<sup>2</sup>.*

Dans les technologies de l'information, une donnée est la représentation d'une information dans un programme : soit dans le texte du programme (*code source*), soit en mémoire durant l'exécution. Les données, souvent codées, décrivent les éléments du logiciel tels qu'une entité (chose), une interaction, une transaction, un événement, un sous-système, etc.<sup>3</sup>

Les données peuvent être conservées et classées sous différentes formes : textuelles (chaîne), numériques, images, sons, etc

---

<sup>1</sup> <https://fr.m.wikipedia.org/wiki/donne>

<sup>2</sup> Claude TAKENGA, informatique de gestion 8, ULPGL/FSTA, 2015

<sup>3</sup> <https://fr.m.wikipedia.org/wiki/donne>

## 2.3 Base des données

Une base des données est un ensemble structuré organisé d'informations avec un objectif commun permettant le stockage de grandes quantités d'informations afin d'en faciliter les différentes manipulations ; c'est notamment le cas de l'ajout, de la mise à jour et éventuellement la recherche de données.

Le support utilisé pour rassembler et stocker les données (papier, fichiers, etc.) peut varier ; mais dès lors que des données sont rassemblées et stockées d'une manière organisée dans un but spécifique, on parle de base de données. Plusieurs bases de données constituées sous forme de collection représentent une banque de données.

**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.

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le Système de gestion de base de données (SGBD).

Un SGBD est caractérisé par le modèle de description des données qu'il supporte (hiérarchique, réseau, relationnel, objet). Les données sont alors décrites sous la forme de ce modèle, grâce à un Langage de Description des Données (LDD). Cette description est appelée schéma.

Une fois la base de données spécifiée, on peut y insérer des données, les récupérer, les modifier et les détruire. C'est la manipulation des données et celles-ci peuvent être manipulées non seulement par un Langage spécifique de Manipulation des Données (LMD), mais aussi par des langages de programmation classiques (Java,...).

Les modèles de bases de données sont :

- **Modèle hiérarchique** : désigne une forme de système de gestion des bases de données dans laquelle les enregistrements sont liés dans une arborescence de façon à ce que chaque enregistrement n'ait qu'un seul possesseur.
- **Modèle réseau** : améliore celui ci-haut cité en introduisant des liaisons de type multiple (n-n) ; la recherche d'une donnée nécessite alors la connaissance de liens et chemins d'accès, ce qui rend le programme dépendant de la structure des données.
- **Modèle relationnel** : est basé sur les principes de l'algèbre relationnel, notamment sur la théorie des ensembles et la logique des prédicats du premier ordre pour résoudre des difficultés telles que la redondance des données, l'intégrité des données ou l'indépendance de la structure de la base de données avec sa mise en œuvre physique.
- **Modèle objet-relationnel** ou objet plus récent.

## 2.4 Système de gestion de base de données

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le Système de gestion de base de données (SGBD). Un SGBD permet l'ajout, la modification et la recherche de données. Il héberge généralement plusieurs bases de données, qui sont destinées à des logiciels ou des thématiques différentes.

La plupart des SGBD actuels fonctionnent selon un mode client-serveur. Le serveur étant la machine stockant les données, recevant des requêtes de plusieurs clients de manière concurrente.

Le serveur analyse la requête, la traite et retourne le résultat au client.

Le modèle client-serveur est assez souvent implémenté au moyen de l'interface des sockets.

Quel que soit le modèle, un des problèmes fondamentaux à prendre en compte est la cohérence des données. Certaines politiques sont mises en œuvre pour définir l'accès aux données, ceci dans le sens de gérer la concurrence d'accès aux données.

Le niveau externe correspond à la perception de tout ou partie de la base par un groupe donné d'utilisateurs, indépendamment des autres. On appelle cette description le schéma externe ou vu.

Il peut exister plusieurs schémas externes représentant différentes vues sur la base de données avec des possibilités de recouvrement. Le niveau externe assure l'analyse et l'interprétation des requêtes en primitives de plus bas niveau et se charge également de convertir éventuellement les données brutes, issues de la réponse à la requête, dans un format souhaité par l'utilisateur.

Le niveau conceptuel décrit la structure de toutes les données de la base, leurs propriétés (i.e. les relations qui existent entre elles : leur sémantique inhérente), sans se soucier de l'implémentation physique ni de la façon dont chaque groupe de travail voudra s'en servir.

Dans le cas des SGBD relationnels, il s'agit d'une vision tabulaire où la sémantique de l'information est exprimée en utilisant les concepts de relation, attributs et de contraintes d'intégrité. C'est le schéma conceptuel.

Il est à noter que le niveau interne ou physique s'appuie sur un système de gestion de fichiers pour définir la politique de stockage ainsi que le placement des données. Le niveau physique est donc responsable du choix de l'organisation physique des fichiers ainsi que de l'utilisation de telle ou telle méthode d'accès en fonction de la requête. C'est le schéma interne.

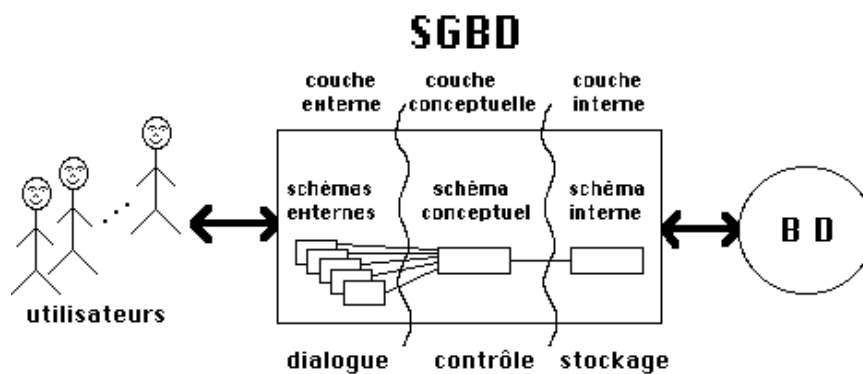


Figure 1 : Schéma d'un système de gestion de bases des données.

Source : K Kyamakya, *Cours de base des données et systèmes d'information p4*, Université de Kinshasa

Il existe de nombreux systèmes de gestion de bases de données ; quelques uns de ces systèmes sont : PostgreSQL, MySQL, Oracle, IBM DB2, Microsoft SQL, Sybase, Informix,...

La présentation et l'utilisation de différentes fonctionnalités des Bases des données se font à travers des interfaces écrites et programmées dans certains langages de programmation, c'est le côté application. Il est à noter que les systèmes de gestion des bases des données utilisent plusieurs langages dont le langage SQL (Structured Language Query), lequel peut être combiné à un langage classique.

### 2.4.1 Langage SQL

SQL est une abréviation anglaise de «Structured Query Language». En français, il s'agit du langage de requête structurée informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

Outre le langage de manipulation des données, la partie langage de définition des données permet de créer et de modifier l'organisation des données dans la base de données ; la partie langage de contrôle de transaction permet de commencer et de terminer des transactions, et la partie langage de contrôle des données permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes.

Les instructions SQL s'écrivent de la manière qui ressemble à celle de phrases ordinaires en anglais. Cette ressemblance voulue vise à faciliter l'apprentissage et la lecture. On classe les ordres suivant 4 domaines :

- LDD (Langage de description des données),
- LMD (Langage de manipulation des données),
- LCD (Langage de contrôle des données),
- LCT (Langage de contrôle de Transaction).

Le langage SQL est un langage déclaratif, c'est-à-dire qui permet de décrire le résultat escompté, sans décrire la manière de l'obtenir.

Les SGBD sont équipés d'optimiseurs de requêtes, des mécanismes qui déterminent automatiquement la manière optimale d'effectuer les opérations.

Les instructions de manipulation du contenu de la base de données commencent par les mots clés SELECT, UPDATE, INSERT ou DELETE qui correspondent respectivement aux opérations de recherche de contenu, modification, ajout et suppression.

## 2.5 Langages de programmation

Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent.

D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire et de significations.

Les langages de programmation permettent de décrire d'une part les structures des données qui seront manipulées par l'appareil informatique, et d'autre part d'indiquer comment sont effectuées les manipulations, selon quels algorithmes. Ils servent de moyens de communication par lesquels le programmeur communique avec l'ordinateur, mais aussi avec d'autres programmeurs, les programmes étant d'ordinaire écrits, lus, compris et modifiés par une équipe de programmeurs.

Un langage de programmation est mis en œuvre par un traducteur automatique : compilateur ou interpréteur.

Un compilateur est un programme informatique qui transforme dans un premier temps un code source écrit dans un langage de programmation donné en un code cible qui pourra être directement exécuté par un ordinateur, à savoir un programme en langage machine ou en code intermédiaire, tandis que l'interpréteur réalise cette traduction 'à la volée'.

Ces langages offrent différentes possibilités d'abstraction et une notation proche de l'algèbre, permettant de décrire de manière concise et facile à saisir les opérations de manipulation de données et l'évolution du déroulement du programme en fonction des situations.

Chaque langage de programmation reflète un paradigme. On entend ainsi par paradigme un ensemble de notions qui orientent le travail de réflexion du programmeur, sa technique de programmation et sa manière d'exprimer le fruit de ses réflexions dans le langage choisi.

### 2.5.1 Éléments du langage de programmation

Un langage de programmation est construit à partir d'une grammaire formelle, qui inclut des symboles et des règles syntaxiques, auxquels on associe des règles sémantiques. Ces éléments sont plus ou moins complexes selon la capacité du langage.

Les modes de fonctionnement et de définition de la complexité d'un langage de programmation sont généralement déterminés par leur appartenance à l'un des degrés de la Hiérarchie de Chomsky.

Les règles de syntaxe sont définies par une grammaire formelle. Elles régissent les différentes manières dont les éléments du langage peuvent être combinés pour obtenir des programmes et autorisent également les identifiants Unicode.

### 2.5.2 Les paradigmes

Un paradigme est un ensemble des connaissances et techniques orientant le travail d'un programmeur et sa façon de travailler.

Chaque langage de programmation reflète un paradigme. Il existe plusieurs types de paradigmes dont :

- **Paradigme Impératif** : basé sur l'exécution des instructions étape par étape, un ensemble des structures contrôlant l'ordre dans lequel sont exécutées les commandes, une procédure principale faisant appel à d'autres.  
Exemple : C, FORTRAN...
- **Paradigme Orienté objet** : chaque objet est une entité active, qui communique avec d'autres objets par échange de messages. Les procédures agissent sur les données et le tout est cloisonné dans des objets.  
Les objets sont groupés en classes. Les objets d'une même classe sont similaires du point de vue de leur structure et de leur comportement. Cette programmation consiste à décrire les classes. Ces dernières sont organisées selon une structure hiérarchique où il y a de l'héritage : de nouveaux objets peuvent être créés sur la base d'objets existants.  
Il est important de noter que ce paradigme aide le programmeur à créer un modèle organisé du problème à traiter et à gérer les interactions. Il permet d'associer fortement les données avec des procédures adaptées pour réaliser des fonctions abstraites ; ce que ne peut faire le paradigme procédural.  
Exemple : C++, Java.

Certains des langages de programmation sont multi-paradigmes, c'est le cas du C++. Le langage Java sera choisi grâce à sa simplicité, sa facilité de trouver de l'aide et son paradigme qui correspond à celui que nous avons adopté.

Il reprend en grande partie la syntaxe du langage C++, très utilisée néanmoins. Java a été épuré des concepts les plus subtils et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche "orientée objet" de sorte qu'en Java tout est objet, à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Java permet de développer des applications client-serveur. Du côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout du côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (Java Server Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

### 2.5.3 Le langage JAVA

Le langage Java sera utilisé au cours du projet. C'est un langage de programmation informatique orienté objet. Sa particularité et son objectif central sont tels que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que GNU/Linux, Mac OS, UNIX ou Windows, avec peu ou pas de modifications. Pour cela, divers plates-formes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

## 2.6 Environnements de développement

Un environnement de développement est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels.<sup>4</sup>

Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction. Certains environnements sont dédiés à un langage de programmation en particulier.

Dans un environnement de développement « intégré » (abrégé EDI en français ou IDE en anglais pour Integrated Development Environment), les outils sont prévus pour être utilisés ensemble (le produit d'un outil peut servir de matière première pour un autre outil) et peuvent être intégrés dès le départ, c'est-à-dire qu'ils sont construits dans le but d'être utilisés ensemble. Il peut aussi s'agir d'un ensemble d'outils développés sans lien entre eux et intégrés a posteriori.

L'objectif d'un environnement de développement est alors d'augmenter la productivité des programmeurs en automatisant une partie des activités et en simplifiant les opérations.

Un environnement de développement comporte toujours un éditeur de texte incorporé comme souligné ci-haut, avec souvent la possibilité de le substituer par un autre éditeur. L'éditeur comporte typiquement des fonctions de mise en évidence alignées avec le langage de programmation: indentation automatique des blocs de code, marquage des délimiteurs (parenthèses ou accolades), et mise en évidence des mots clés du langage par de la couleur ou des caractères gras, mais aussi un outil de création d'interface graphique, outil permettant au programmeur de gagner un temps significatif dans la construction de l'interface graphique de son programme.

Jusqu'à l'arrivée de la technologie Java, de tels outils ciblaient toujours un système d'exploitation en particulier. Ces environnements ont aussi des fonctions telles que :

- un moteur de recherche qui tient compte du langage de programmation: il permet par exemple de rechercher le nom d'une fonction en évitant les commentaires et les expressions littérales, et en se limitant au cadre d'un module ou d'une classe.
- des outils destinés à assister aux opérations préliminaires à la programmation, par exemple des outils de modélisation ou d'analyse des exigences.
- un outil de contrôle de versions. Un tel outil permet à plusieurs programmeurs de travailler simultanément sur les fichiers de code source du programme.

Netbeans et Eclipse sont des environnements de développement populaires.

Netbeans sera utilisé pour notre projet, car il constitue une plate-forme permettant le développement d'applications spécifiques notamment grâce à sa bibliothèque "Swing" sur laquelle il s'appuie. Il

---

4 (en)Donald G. Firesmith - B. Henderson- Sellers, *The OPEN Process Framework: An Introduction*, Pearson Education – 2002, (ISBN 9780201675108)

comprend en outre un explorateur de base des données supportant les bases des données. L'explorateur comprend un éditeur des requêtes, un gestionnaire intégré de bases des données MySQL.

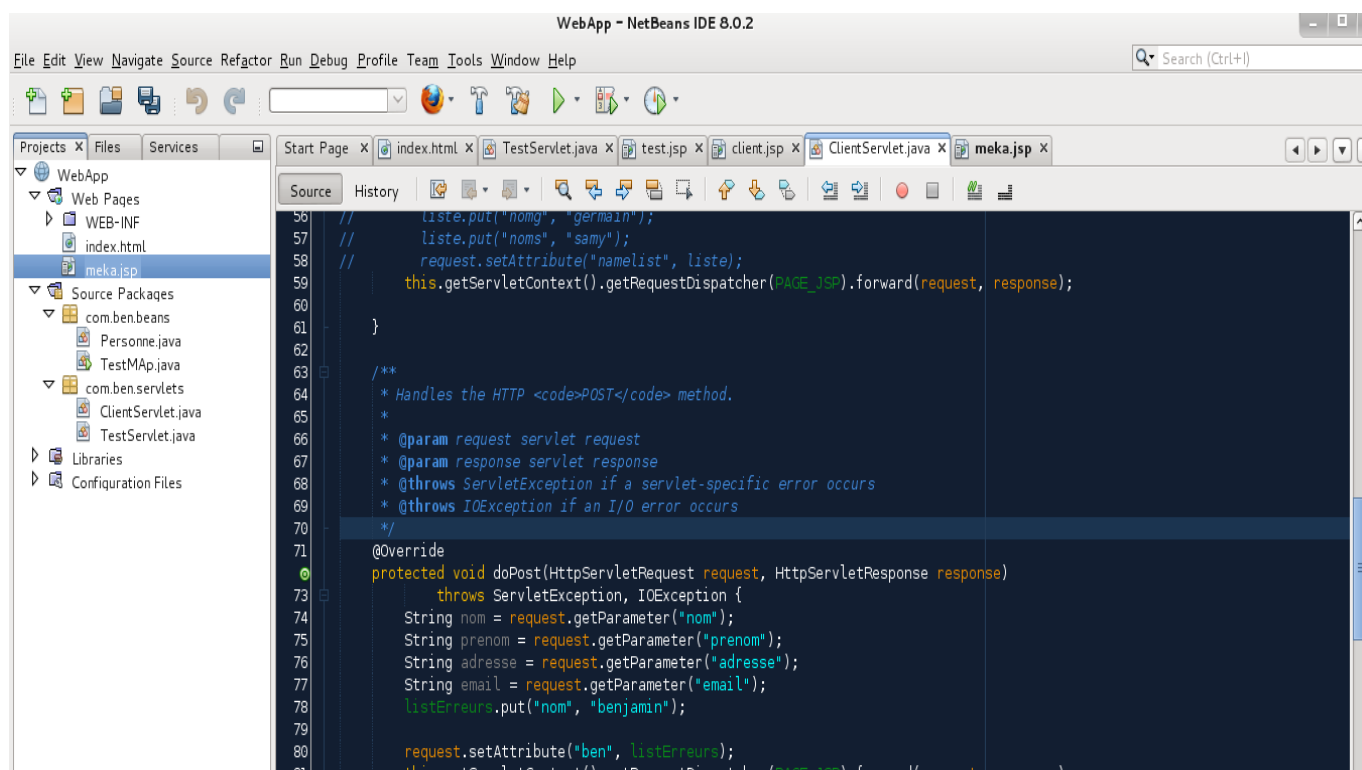


Figure 2 : Netbeans IDE

## 2.7 Architecture

L'architecture désigne la structure générale inhérente à un système informatique, l'organisation des différents éléments du système (logiciels et/ou matériels et/ou humains et/ou informations) et des relations entre les éléments.

La décision du choix d'une architecture doit découler de l'exploration de l'éventail des besoins d'un client ainsi que de l'éventail des moyens technologiques à sa disposition, d'identifier les points clés parmi les besoins du client et d'y apporter une solution théorique, puis de décrire schématiquement la solution en question sous forme de diagrammes.

Pour un problème donné, il peut exister plusieurs solutions ; on est donc amené à choisir la mieux adaptée en tenant compte des qualités, coûts et défauts de différentes solutions envisageables.

Pour cela, on se base sur le savoir-faire tel que les styles et patrons (design pattern en anglais), l'expérience et les recommandations en rapport avec les différentes solutions envisageables.

Les patrons d'architecture (en anglais : design patterns) sont des modèles de référence de résolution des problèmes courants d'architecture<sup>5</sup>. Ils sont utilisés comme source d'inspiration dans de nombreux produits informatiques. L'héritage d'un patron se reconnaît par le style caractéristique du diagramme d'architecture du produit.

Dans le style à filtres et tubes, une série de programmes sont reliés entre eux par des tubes. Le

<sup>5</sup> <http://wikipedia.fr/architectures.html>

produit de l'exécution d'un programme est transmis par le tube où il servira de matière première pour le programme suivant, et ainsi de suite. Un programme donné n'a pas besoin d'attendre que le précédent ait terminé le travail, et commence dès lors que des informations lui sont envoyées par le tube. Ce style est utilisé dans l'architecture de nombreuses applications de manipulation du son et de la vidéo.

Dans l'architecture de style client-serveur, l'application informatique est décomposée en deux sous-systèmes différents prévus pour résider sur des ordinateurs différents. Les deux sous-systèmes communiquent selon des protocoles réseau normalisés.

Le patron "modèle-vue-contrôleur" sera utilisé pour notre projet, il est un modèle souvent utilisé dans l'architecture de l'interface graphique des logiciels. Il est fait de trois composants : le modèle, la vue et le contrôleur.

Le contrôleur prend en charge les opérations effectuées par l'utilisateur à la souris et au clavier (pour cette application, les différentes manipulations), et les transforme en messages destinés au modèle et à la vue.

Le modèle (base de données) stocke les informations à traiter et envoie des messages à la vue lors de modifications. La vue reçoit alors les messages et effectue les modifications nécessaires de l'interface graphique<sup>6</sup>.

Dans une architecture en couches, les composants sont regroupés en sous-systèmes placés les uns au-dessus des autres. Chaque composant d'un sous-système donné est en relation uniquement avec des composants des sous-systèmes placés immédiatement au-dessus ou en dessous.

Dans le style dit architecture trois tiers, les composants sont regroupés en trois couches qui concernent respectivement l'affichage sur l'interface graphique, la logique et le stockage en base de données. Chaque couche peut résider sur un ordinateur différent.

L'architecture orientée services est souvent considérée comme un patron. Elle est souvent associée au concept de "Enterprise Service Bus" (abrégé ESB), une ligne de communication publique à laquelle se connectent à la demande, différents logiciels fournisseurs et consommateurs de services.

---

<sup>6</sup> (en) Frank F. Tsui & Orlando Karam, Essentials of software engineering, Jones & Bartlett Publishers, 2006 (ISBN 9780763735371)

## CHAPITRE 3 : MÉTHODOLOGIE DE RECHERCHE

### 3.1 Définitions

Dans les disciplines scientifiques, un chercheur doit concevoir sa recherche et penser aux moyens à utiliser à chacune des étapes, c'est sa méthodologie. C'est donc un ensemble des méthodes et techniques orientant l'élaboration d'une recherche visant à guider la démarche scientifique. La méthode quant à elle représente l'ensemble des règles et principes organisant le mouvement d'ensemble de la connaissance, c'est-à-dire les relations entre l'objet de la recherche et le chercheur, mais aussi entre les informations rassemblées à l'aide des techniques et le niveau de théorie des concepts.

Les méthodes sont nombreuses mais pour dégager un élément commun à toutes, la plupart d'entre elles sont un ensemble concerté d'opérations mises en œuvre visant à atteindre un ou plusieurs objectifs, un corps de principes présidant à toute recherche scientifique organisée, un ensemble de normes permettant de sélectionner et coordonner les techniques. Elles constituent ainsi de façon plus ou moins abstraite ou concrète, précise ou vague, un plan de travail en fonction d'un but.

### 3.2 Choix et intérêt des méthodes et techniques

Pour atteindre nos objectifs, nous recourons à un certain nombre de méthodes et techniques reprises dans les lignes qui suivent.

Les méthodes que nous avons utilisées sont :

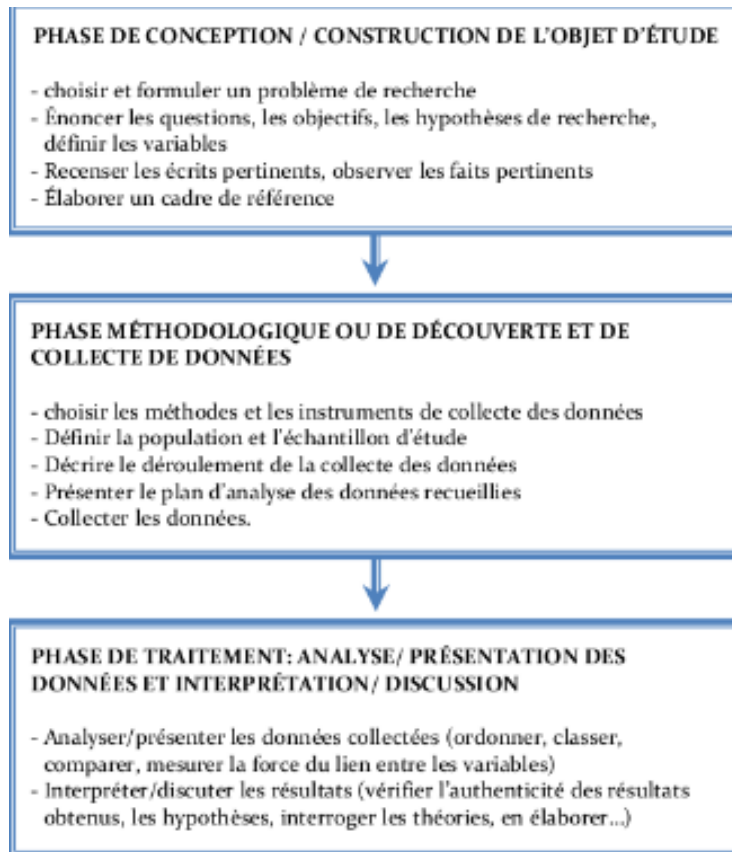
- a) **Méthode analytique** : utile pour l'analyse des données récoltées au cours de la descente sur terrain.
- b) **MERISE** : pour le traitement des données en vue de la conception en passant du modèle conceptuel au modèle physique de données, mais aussi de représenter différents modèles du système existant de la gestion de dossiers médicaux, de concevoir nos propres modèles (système future) dans les mêmes domaines, centrés sur l'architecture du logiciel que nous allons mettre en place et guidés sur les besoins des utilisateurs de notre système.

Les techniques utilisées sont :

- a) **Technique d'interview** : a consisté à entrer en contact direct avec le personnel médical pour s'imprégner de la façon dont ce domaine est organisé, la manière dont les patients sont gérés et les difficultés auxquelles ils font face pour en sortir les fonctionnalités à implémenter dans le logiciel. Celle-ci permet de recueillir les informations sur les différentes étapes par lesquelles le malade doit passer pour aboutir aux soins médicaux commençant par son arrivée pour finir par l'enregistrement des informations qui lui concernent ; et aussi les différents procédés utilisés pour aboutir aux différents documents de synthèse.
- b) **Technique de la revue documentaire** : il s'est agi de s'informer davantage en consultant les notes des cours, les livres, les tutoriels et en surfant sur internet.
- c) **Technique d'observation** : avec cette technique, il s'est agi de se livrer à la constatation des faits, des activités liées à la gestion des données des patients pour en déduire les fonctions et procédures de gestion et d'autres informations importantes à l'implémentation du logiciel.

### 3.3 Étapes clés de la méthodologie :

Pour la méthodologie choisie, les étapes clés ont été recensées et regroupées en 3 phases de cette manière :



## CHAPITRE 4 : COLLECTION DES DONNÉES, ANALYSE ET DISCUSSION DES RÉSULTATS.

Ce chapitre porte sur la collecte des données, leur analyse ainsi que la discussion. La collecte des données porte sur ce que l'on a eu des différentes interviews, de différentes recherches dans les documents et sur internet ensuite l'analyse se portera sur le système actuel et une éventuelle prédiction du comportement du système futur en fonction des données recueillies.

Aujourd'hui, la gestion informatisée du dossier patient est une priorité dans le domaine de santé.

Pour améliorer l'accueil et les soins des patients, pour faciliter et sécuriser le travail du personnel soignant, pour optimiser la gestion administrative et financière ou pour s'adapter en temps réel aux normes réglementaires, tous les établissements sont concernés par cette démarche.

### 4.1 Population et échantillon de l'étude

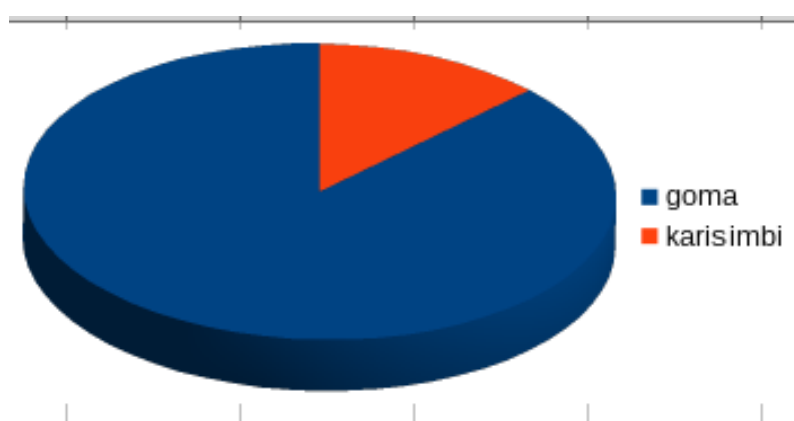
Lamoureux A. (2000 p. 169) définit la population comme étant l'ensemble de tous les éléments auxquels on espère généraliser les résultats d'une recherche. Ainsi, notre population dans la présente étude est constituée par tous les hôpitaux de la ville de Goma. Elle est donc finie et exhaustive car on peut connaître exactement leur nombre.

Cependant, le thème évoqué étant un thème très complexe, il s'avère indispensable de recourir à un échantillon occasionnel se contentant ainsi des personnels disposés à accorder une interview. Ainsi, comme il peut être constaté dans le tableau ci-dessous, notre échantillon est constitué de huit (8) hôpitaux et centres de Santé de la ville parmi lesquels six (6) grands hôpitaux et deux (2) Centres de Santé. L'échantillon occasionnel selon F.PELTEAU (2000, P .128) est composé des unités rencontrées au hasard.

### Description de l'échantillon selon les communes

Communes	Nombre d'hôpitaux ou centres
Goma	7
Karisimbi	1

Graphiquement la situation se présente de la manière suivante :



La lecture des données de ce tableau et de ce graphique montre que de 8 hôpitaux et centres de santé, 7 sont issus de la commune de Goma, 1 de la commune de karisimbi.

Après enquête sur la gestion de ces hôpitaux et centres de santé par un software, les résultats se présentent de la manière suivante :

Communes	Hôpitaux et Centres gérés par un software	Hôpitaux et Centres non gérés par un software
Goma	3	4
Karisimbi	0	1

Taux d'utilisation=  $\{(3/8)*100\} = 37.5 \%$

L'interprétation de ce tableau montre un faible taux d'utilisation d'un software soit 37.5 % pour tous les hôpitaux et centres.

#### 4.2 Résultats de l'interview

L'interview s'est portée sur le déroulement des activités au sein de l'hôpital et l'interaction entre les différents acteurs du système et les différents services.

Le processus de traitement du malade se déroule de la manière suivante :

- Le malade se présente à la réception pour se faire enregistrer et paie la fiche de consultation ;
- le réceptionniste oriente le malade vers la salle de triage où il sera consulté ;
- le malade parle de ses problèmes au consultant et répond aux différentes questions lui posées ;
- le consultant envoie le malade au laboratoire (si les examens de laboratoire sont demandés) accompagné de la fiche de demande des examens de laboratoire ;
- le malade se présente au laboratoire avec sa fiche ;
- le laborantin fait de prélèvements, le malade attend que les résultats de tests soient inscrits sur le bon des examens de laboratoire avant que ce bon lui soit remis ;
- après le laboratoire, le malade revient au cabinet de consultation avec les résultats des examens de laboratoire pour interprétation par le consultant ; le consultant diagnostique la pathologie du malade et lui prescrit un traitement ;
- le malade est envoyé à la pharmacie si son cas ne nécessite pas l'hospitalisation, pour être servi après facturation à la comptabilité et paiement à la caisse. Si le cas nécessite une hospitalisation, il est envoyé dans le service d'hospitalisation pour sa prise en charge par les infirmiers ;
- pour le cas hospitalisé, lorsque sa sortie est autorisée, la fiche passe au service de comptabilité pour facturation. Ce n'est qu'après la facturation que le malade va se présenter à la caisse pour le paiement de sa facture.

Après cette étape, suit celle concernant l'analyse et l'étude des cas qui fera l'objet du chapitre suivant.

## CHAPITRE 5 : ÉTUDE DES CAS

Après la récolte et l'analyse des données, il s'avère indispensable d'analyser le système actuel et de concevoir un système pour le futur. Ce chapitre porte donc sur l'étude de ces 2 systèmes.

### 5.1 Étude du système existant

Selon le dictionnaire Robert, un système est un dispositif formé par la réunion d'éléments. La définition du Larousse semble plus explicite : c'est la «Combinaison de parties qui se coordonnent pour concourir à un résultat, de manière à former un ensemble». Tout système fonctionne en transformant des flux d'entrée en flux de sortie selon des processus plus ou moins complexes. Un système doit comporter généralement 3 parties distinctes<sup>7</sup> :

- **Le Système de Pilotage ou Système de Gestion (SP)** : procède au pilotage du Système Opérant (SO) en décidant sur le comportement de celui-ci en fonction des objectifs fixés.
- **Le Système d'Information(SI)** : Il sert à véhiculer les informations venant du SO vers le SP et vice versa, mais rien n'empêche que le SP peut passer directement les informations vers le SO.
- **Le Système Opérant (SO)**: Est celui qui agit sur les activités de l'organisation pour atteindre les objectifs fixés par le SP. Le système de cadres ou système de pilotage procède au pilotage, à la régulation et au contrôle du système opérant.

En ce qui concerne l'hôpital militaire de Goma, le processus du parcours du patient se fait au travers les différents services liés entre eux ; l'information circulant dans ces différents services est la fiche du patient. Cette dernière contient toutes les informations concernant le patient.

Les activités liées à l'entrée, au suivi et à la sortie des malades sont inscrites dans des registres spécifiques.

### 5.2 Contexte du système

Le système qu'il nous faut mettre en place devra implémenter toutes les fonctionnalités liées à la gestion des patients dans l'hôpital. Il s'agit principalement de :

- L'enregistrement des patients ;
- L'attribution des fiches aux patients,
- Le suivi des patients ;
- L'admission des patients dans différents services via cette fiche.

Dans les lignes qui suivent, nous allons présenter les différents traitements liés à ces opérations.

---

<sup>7</sup> Jean Luc Baptiste, *MERISE Guide pratique, modélisation des données*, langage sql, p1,eni éditions

### 5.3 Circuit de déroulement

#### 5.3.1 Condition d'entrée dans le système

Pour notifier l'entrée d'un patient ou son enregistrement, il faut s'assurer au préalable qu'il n'est répertorié dans la liste des patients figurant dans les registres. Si ce n'est pas le cas, on procédera d'abord à l'enregistrement de celui-ci, puis à la prise en compte de ses informations notamment le nom, le post-nom, le genre, l'âge et l'adresse, puis l'octroi d'un jeton sur lequel figure un numéro que le patient pourra user dans le futur s'il arrivait qu'il revienne pour un autre problème de santé. Ce numéro ou jeton va aider lorsqu'il reviendra, afin de retrouver sa fiche ou son dossier médical beaucoup plus facilement. La prise en charge du malade devant tenir compte de son passé, de l'historique de son état de santé à travers le jeton qui permet de retrouver sa fiche.

#### 5.3.2 Condition d'admission dans les différents services

Pour être admis dans les différents services, le patient doit avoir une fiche sur laquelle sont inscrits les informations actualisées le concernant, relatives à son état de santé, fiche lui remise à la réception, et qui lui a permis d'accéder au service de triage avant de se présenter au cabinet de consultation. Après la consultation, il peut maintenant accéder à d'autres services selon le cas et prescription du consultant.

#### 5.3.2 Condition de sortie du système

Lorsque la situation du malade ne nécessite pas l'hospitalisation, après consultation médicale, examens de laboratoire éventuellement ou autres examens prescrits et la prescription de médicaments, le malade sort du système après paiement de sa facture pour les soins de santé reçus.

Si le malade a été hospitalisé, le médecin devra d'abord autoriser sa sortie après analyse de sa situation. Ce n'est qu'après avis favorable du médecin qu'il pourra passer au service de comptabilité pour sa facturation et à la caisse pour paiement de sa facture relative aux soins lui administrés.

#### 5.3.3 Identification des lacunes du système existant

Après avoir fini l'étude du contexte du domaine d'étude, nous avons approfondi la compréhension étendue du fonctionnement du système actuel. A présent il est possible de relever certains écarts sur les plans informationnel, technique et organisationnel.

Plan	Insuffisances	Suggestions
Informationnel	<ul style="list-style-type: none"> <li>• Collecte pas toujours exhaustive des données</li> <li>• Redondance des données</li> <li>• Perte de certaines informations</li> <li>• Perte de temps considérable suite aux recherches</li> </ul>	Sauvegarde des données dans une base des données facilitant l'accès aux données.

<b>Plan</b>	<b>Insuffisances</b>	<b>Suggestions</b>
Technique	<ul style="list-style-type: none"> <li>• Pas de modèle d'organisation des données en vue de leur traitement</li> <li>• Traitement lent des données</li> <li>• Calculs lents</li> </ul>	Automatisation des calculs avec possibilité de modification des données sans surcharge.
Organisationnel	<ul style="list-style-type: none"> <li>• Pas de compétences avérées en informatique</li> <li>• Pas de logiciel commercial disponible</li> <li>• Pas d'outils informatiques appartenant à l'institution</li> </ul>	Disposer d'un logiciel pour l'enregistrement des données et de l'outil informatique

**Tableau : Identification des insuffisances du système existant**

#### **5.4 Proposition**

Le système d'information actuel de l'Hôpital Militaire de Goma est non encore automatisé, les tâches et procédures sont encore manuelles ; ce qui rend le travail des acteurs du système fastidieux et complexe. Les acteurs ne l'assument pas entièrement. Les tâches journalières non achevées, les erreurs, les enregistrements non exhaustifs et les oublis ont de grandes répercussions sur la qualité du travail et par conséquent, un effet négatif sur les résultats attendus au niveau de l'hôpital.

De ce qui précède, et pour que l'hôpital militaire de Goma améliore la qualité du travail, il serait utile et nécessaire d'avoir un logiciel (le concevoir en rapport avec ses attentes) pour la gestion des patients qui le fréquentent.

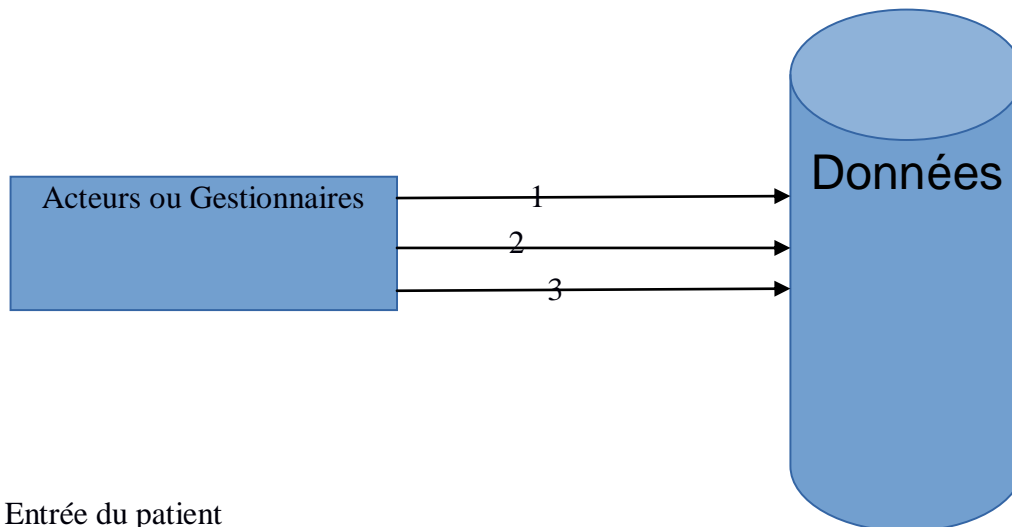
#### **5.5 MCC du système d'information actuel**

Sous ce point, nous réalisons le diagramme du contexte qui montre comment les informations devraient circuler entre différents acteurs internes et externes au système. L'identification des acteurs communiquant avec le système ainsi que les messages qui transitent de part et d'autre s'avère cependant indispensable.

##### **5.5.1 Identification des acteurs**

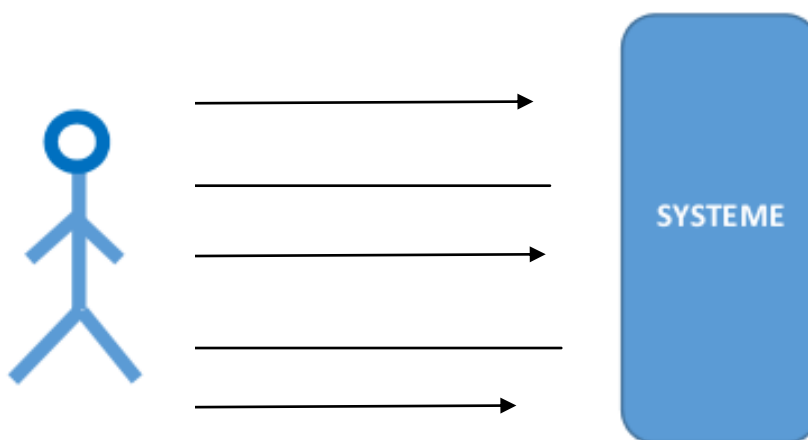
Ce sous point identifie les différents acteurs qui interagiront avec le système. Il s'agit effectivement du réceptionniste, des infirmiers et des médecins. Ils pourront effectuer les commandes reprises dans les fonctionnalités du logiciel.

### 5.5.2 Schéma du Modèle Conceptuel de Communication (MCC) du système d'information



1. Entrée du patient
2. Signalisation d'ajout du patient
3. Modification des informations du patient

### 5.5.3 Schéma du MCC du système Informatique



1. le gestionnaire ajoute un patient dans le système. Il est sensé remplir un formulaire d'ajout de patient prévu à cet effet.
2. le système prend en compte l'ajout et actualise la base des données puis affiche au gestionnaire la liste déjà actualisée et l'historique qu'il a gardé.
3. le gestionnaire modifie ou ajoute certaines informations pour le patient suivant le service.
4. le système prend en compte les modifications et actualise sa base des données.
5. le gestionnaire signale au système la sortie du patient.

Pour détailler plus ces processus, voici la liste de flux du système :

#### 5.5.4 Liste de flux

1. Le malade se présente à la réception pour se faire enregistrer et paie la fiche de consultation ;
2. le réceptionniste oriente le malade vers le bureau de l'infirmier ou du médecin qui va le consulter ;
3. le malade parle de ses problèmes au consultant et répond aux questions qui lui sont posées ;
4. le consultant envoie le malade au laboratoire accompagné du bon d'examens de laboratoire ou la fiche pour de prélèvements en vue des examens ;
5. le malade se présente au laboratoire avec son bon d'examens ou sa fiche ;
6. le laborantin, après examens, remet la fiche ou le bon de laboratoire au malade avec les résultats des différents examens demandés ;
7. le malade revient chez le consultant avec la fiche de laboratoire ou bon d'examens de laboratoire avec les résultats qu'il remet au consultant ;
8. le consultant diagnostique la maladie et prescrit un traitement au malade (malade en ambulatoire) ; le malade qui nécessite l'hospitalisation est envoyé dans le service d'hospitalisation pour sa prise en charge et son suivi ;
9. le malade qui ne nécessite pas l'hospitalisation est envoyé à la pharmacie pour le retrait des médicaments lui prescrits (achat) ;
10. le malade est envoyé dans la salle d'observation ;
11. les infirmiers traitent le malade ;
12. le pharmacien ou le préposé à la pharmacie donne les médicaments au malade ;
13. le malade est facturé par la comptabilité ;
14. le malade paie sa facture à la caisse.

#### 5.5.5 Tableau des acteurs.

Numéro	Acteur	Type	Signification
1	Malade	Interne	Malade venant se faire soigner
2	Réceptionniste	interne	Accueil de toute personne se présentant à l'hôpital
3	Consultant	Interne	Consultation quotidienne du malade se présentant et administration ou prescription des traitements.
4	Laborantin	Interne	effectue les examens de laboratoire prescrits par le consultant.
5	infirmiers de salles d'hospitalisation	Interne	Traitement et suivi des malades internés
6	Chargé de la pharmacie	Interne	servir le malade sur base de la prescription du consultant.
7	Comptable	Interne	gère la facturation et la caisse.

### 5.5.6 Modèle Organisationnel de Traitement.

Les règles d'organisation sont celles réglant l'orientation générale et amenant à effectuer les actions à une prestation.

RO1 : les réceptions des cas de consultation commencent à 08h00 et se terminent à 12h00.

RO2 : les traitements de malades ne prennent pas fin, ils s'effectuent 24h/24h.

RO3 : Le personnel médical tel que les infirmiers et médecins.

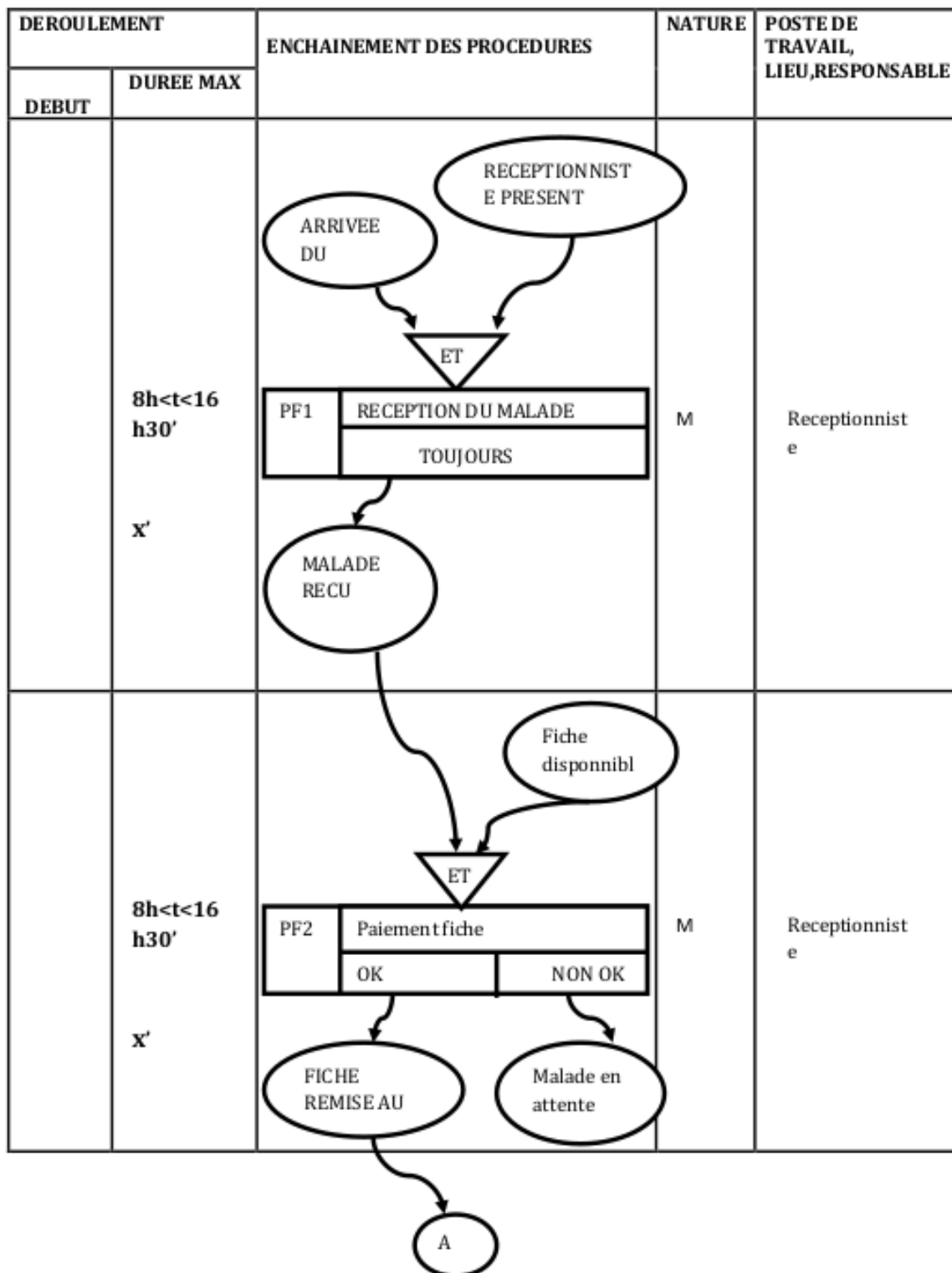
RO4 : Si le malade qui se présente suivait déjà un traitement, il a l'obligation de se présenter avec son carnet de soins.

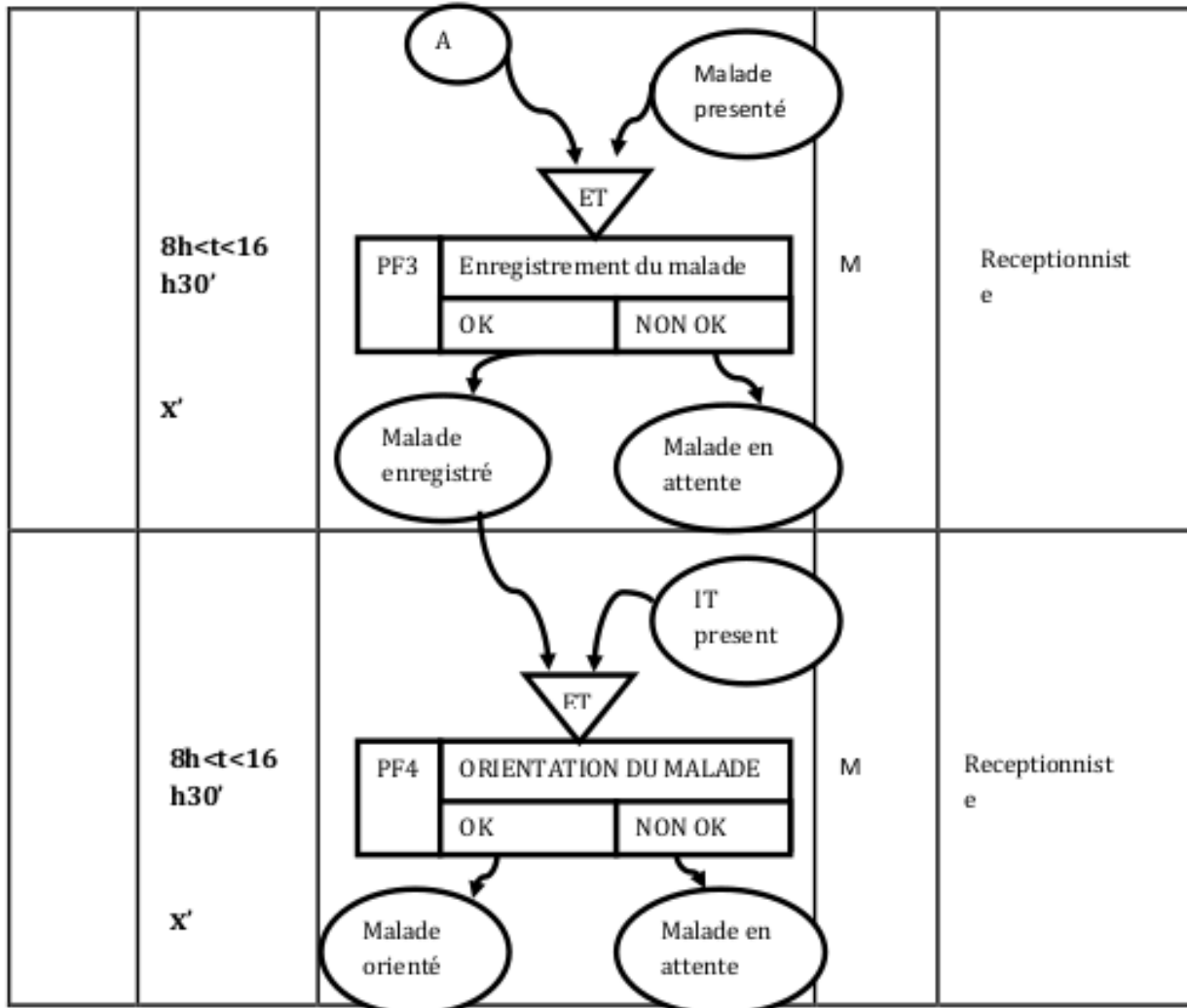
### 5.5.7 Tableaux des procédures fonctionnelles.

#### 5.5.7.1 Processus de réception

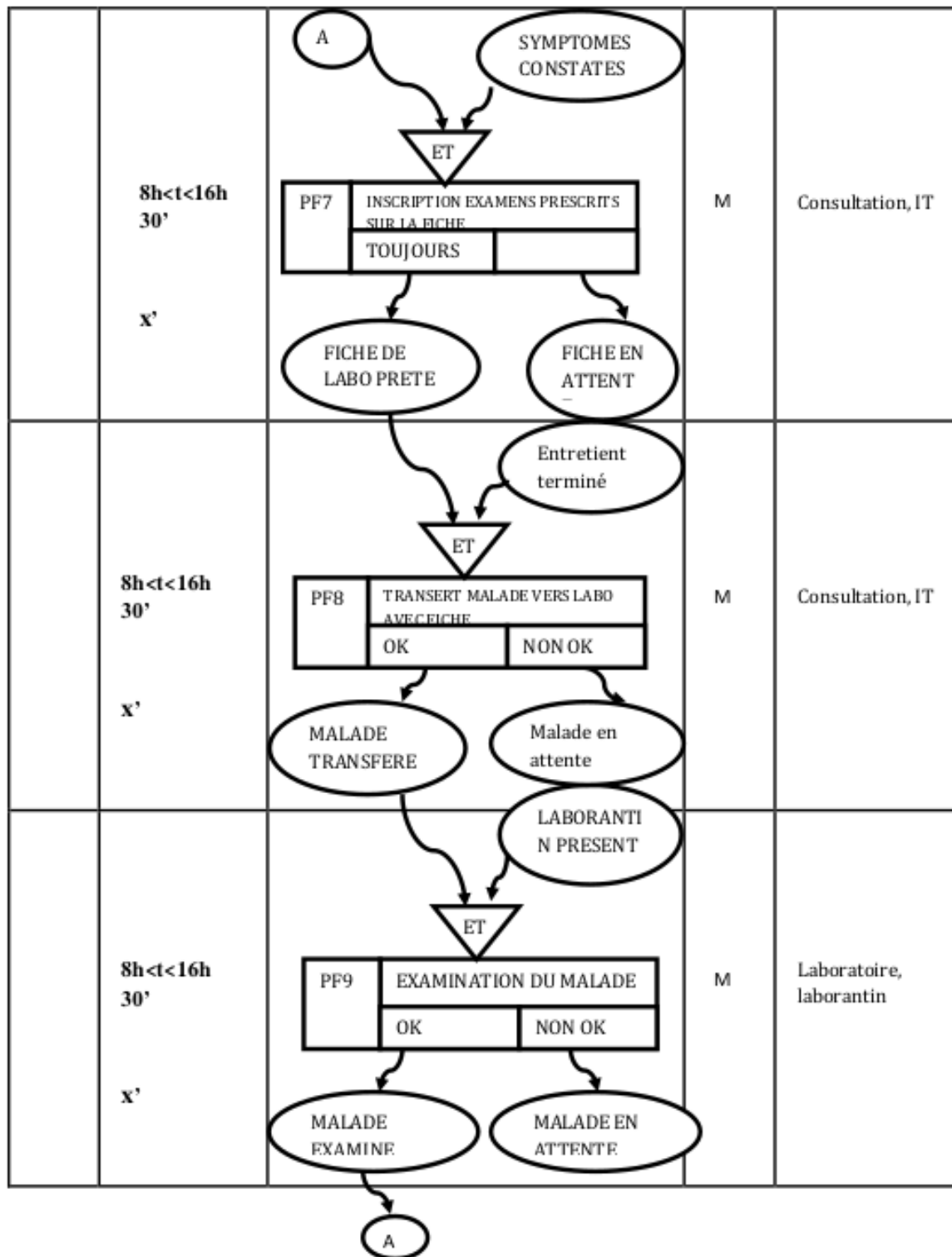
PF	DEROULEMENT		ACTIONS	NATURE	POSTE DE TRAVAIL		
	Début	Durée max			Lieu	Responsable	Ressource
PF1	8h<t<16h30'	x'	RECEPTION DU MALADE	M	RECEPTION	RECEPTIONNISTE	
PF2	8h<t<16h30'	x'	Paiement fiche de consultation	M	RECEPTION	réceptionniste	
PF3	8h<t<16h30'	x'	Enregistrement arrivée du malade	M	RECEPTION	réceptionniste	
PF4	8h<t<16h30'	x'	Orientation du malade	M	RECEPTION	réceptionniste	

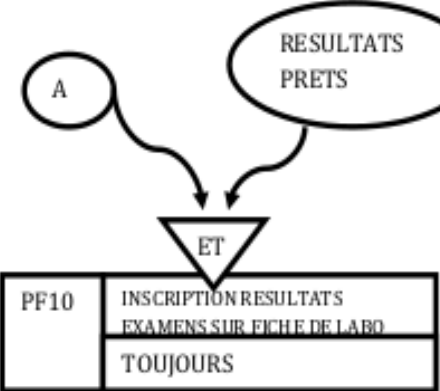
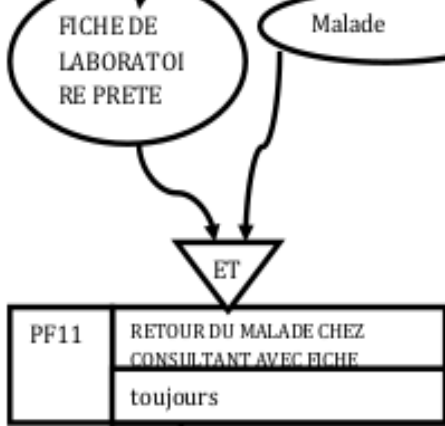

## 5.5.7.2 Tableaux de déroulement des procédures fonctionnelles

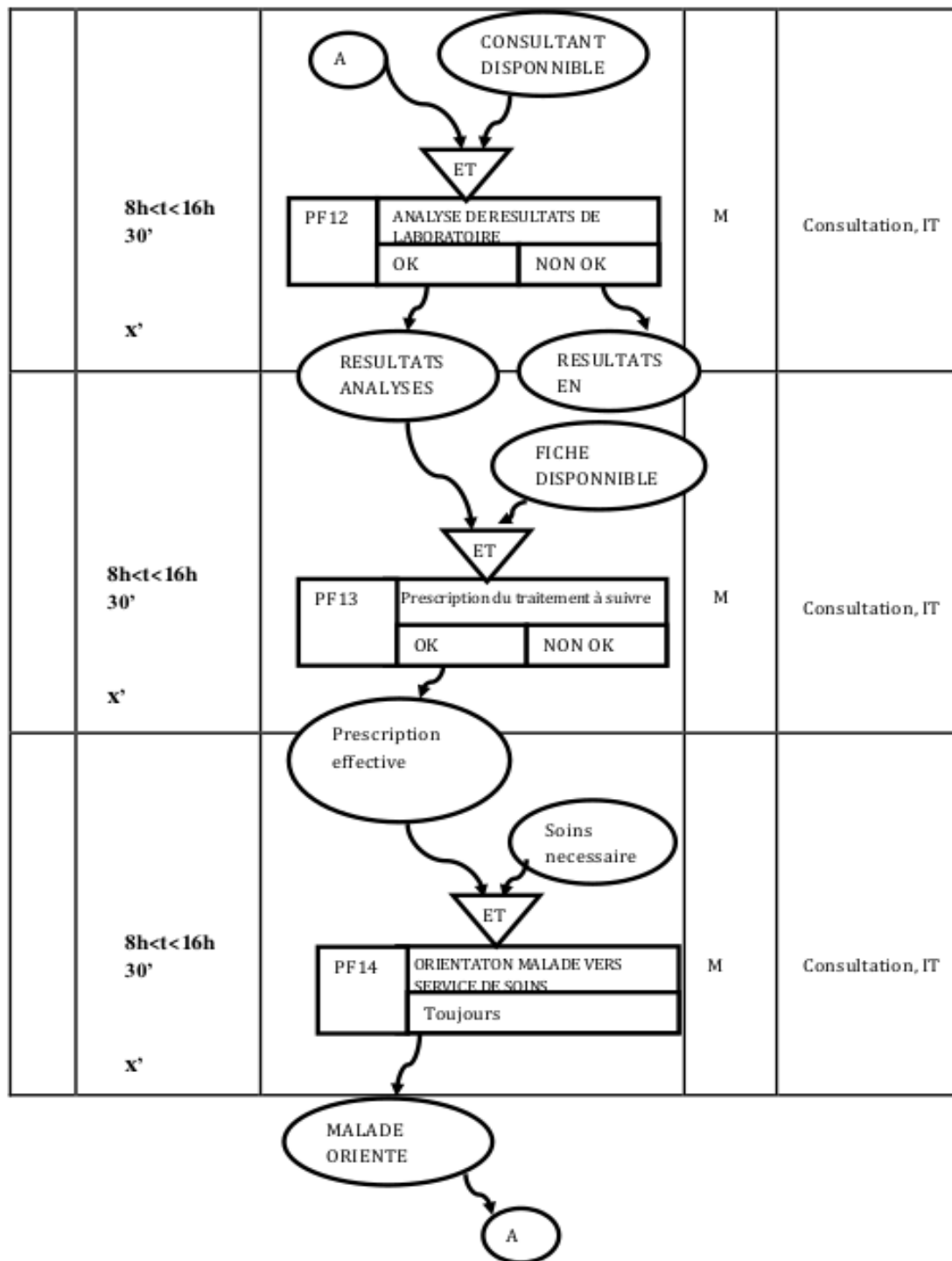


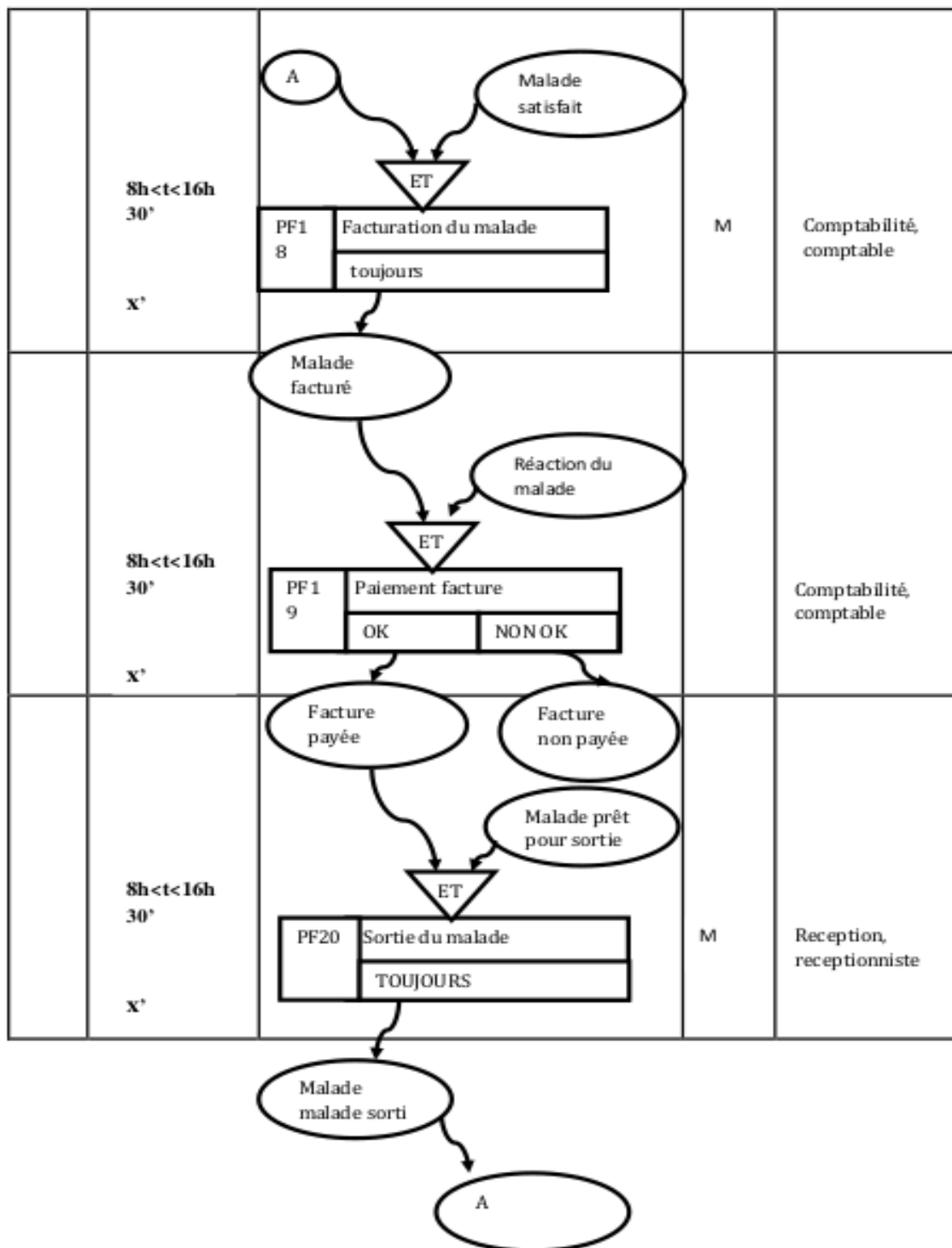


DEROULEMENT		ENCHAÎNEMENT DES PROCEDURES	NATURE	POSTE DE TRAVAIL, LIEU, RESPONSABLE
DEBUT	DUREE MAX			
	<p>8h &lt; t &lt; 16h 30'</p> <p>x'</p>	<pre> graph TD     A1([Arrivée du malade]) --&gt; ET1{ET}     A2([IT disponible]) --&gt; ET1     ET1 --&gt; B1[PF5   Accueil du malade   TOUJOURS]     B1 --&gt; C1([Malade installé])           </pre>	M	Consultation, IT
	<p>8h &lt; t &lt; 16h 30'</p> <p>x'</p>	<pre> graph TD     A2([MALADE PRET]) --&gt; ET2{ET}     C1([Malade installé]) --&gt; ET2     ET2 --&gt; B2[PF6   DIALOGUE AVEC LE MALADE   TOUJOURS]     B2 --&gt; C2([ENTRETIEN TERMINE])     C2 --&gt; A([A])           </pre>	M	Consultation, IT



	<p>8h &lt; t &lt; 16h 30'</p> <p>x'</p>		M	Laboratoire, laborantin
	<p>8h &lt; t &lt; 16h 30'</p> <p>x'</p>		M	Laboratoire, laborantin
				





A l'issue de cette phase, nous pouvons maintenant, non seulement avoir une vue détaillée de l'état actuel de l'organisation de ces services, mais aussi avoir un aperçu sur les différentes activités et traitements qui se font dans la gestion de patients. Le diagramme de contexte réalisé au niveau de cette étape a donné un premier aperçu sur l'application à concevoir, ouvrant ainsi la porte à la deuxième étape de MERISE intitulée « Conception et modélisation ».

## 5.6 Conception et modélisation du système informatique

L'objet de la présente partie est d'essayer d'étendre la représentation des modèles effectués au niveau conceptuel en y intégrant les aspects techniques plus proches des préoccupations physiques, laquelle représentation sera utile pour pouvoir comprendre et mieux décrire les concepts à implémenter au cours de la réalisation du projet.

Après la phase d'analyse, il s'en suit directement la phase de conception qui requiert les données issues de la phase d'analyse pour ensuite passer à l'implémentation dans un environnement de développement.

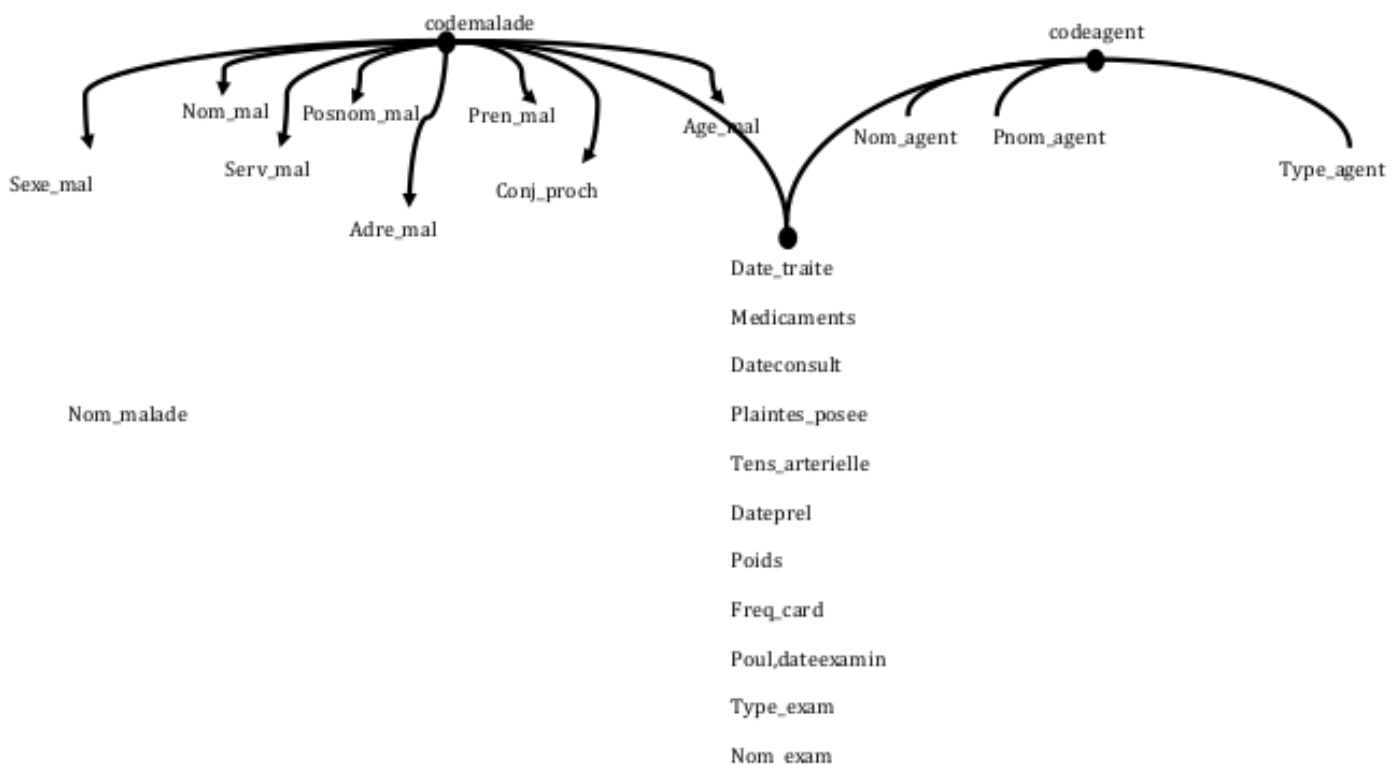
### 5.6.1. Modèle Conceptuel des Données.

Après collection des données, l'on aboutit au dictionnaire des données suivant :

NUM	IDENTIFIATION	SIGNIFICATION	TYPE	DOMAINE	CONTROLE
1	Num_id	Numero d'identité	AN-15		unique
2	Nom_mal	Nom du malade	AN-15		
3	Pnom_mal	Postnom du malade	AN-15		
4	Sex_mal	Sexe du malade	AN-15		
5	Serv_mal	Service du malade	AN-15		
6	Age_mal	Age du malade	AN-15		
7	Adr_mal	Adresse du malade	AN-15		
8	Tel_mal	Telephone du malade	AN-15		
9	Typ_mal	Type du malade	AN-15		
10	Conj_pro	Conjoint ou proche du malade	AN-15		
11	Date_entr_obs	Date d'entrée en observation	AN-15		
12	Date_sort_obs	Date de sortie en observation	AN-15		
13	Etat_mal	Etat du malade	AN-15		
14	Diag_entr_mal	Diagnostic entrée du malade	AN-15		
15	Diag_sort_mal	Diagnostic sortie du malade	AN-15		
16	Med	Medicaments	AN-15		
17	Temp_mal	Temperature du malade	AN-15		
18	Poul_mal	Poul du malade	AN-15		
19	Res_clin	Résumé clinique	AN-15		
20	Ex_recq	Examens requis	AN-15		
21	Conclu	Conclusion	AN-15		
22	Date_deb_prisemed	Date debut prise medicaments	AN-15		
23	Date_fin_prisemed	Date fin prise de medicaments	AN-15		
24	Nom_inf_tr	Nom infirmier traitant	AN-15		
25	TA	Tension artérielle	AN-15		
26	BP	Besoins perturbés	AN-15		
27	FC	Fréquence cardiaque	AN-15		
28	CAT	Conduite à tenir	AN-15		
29	Obs	Observation	AN-15		
30	N_med	Nom medecin	AN-15		
31	Date	Date	AN-15		

32	Num_fich_surv	Numero fiche surveillance	AN-15		
33	Plaintes posées		AN-15		
34	Ant_malade	Antecedents du malade	AN-15		
35	Pnom_inf_trait	Postnom infirmier traitant	AN-15		
36	Poid_mal	Poids du malade	AN-15		
			AN-15		
			AN-15		
			AN-15		
			AN-15		

### 5.6.2 Graphe des dépendances fonctionnelles

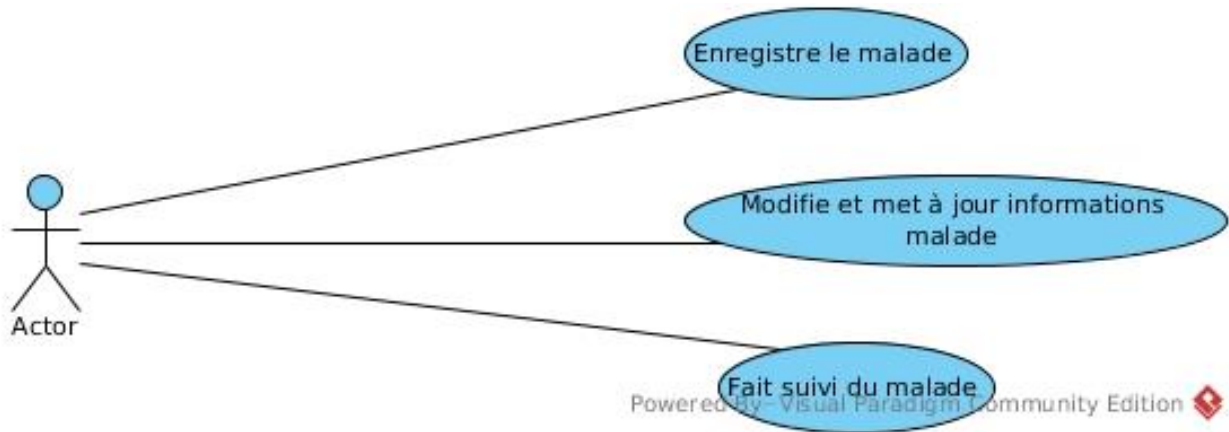


## 5.7 Réalisation

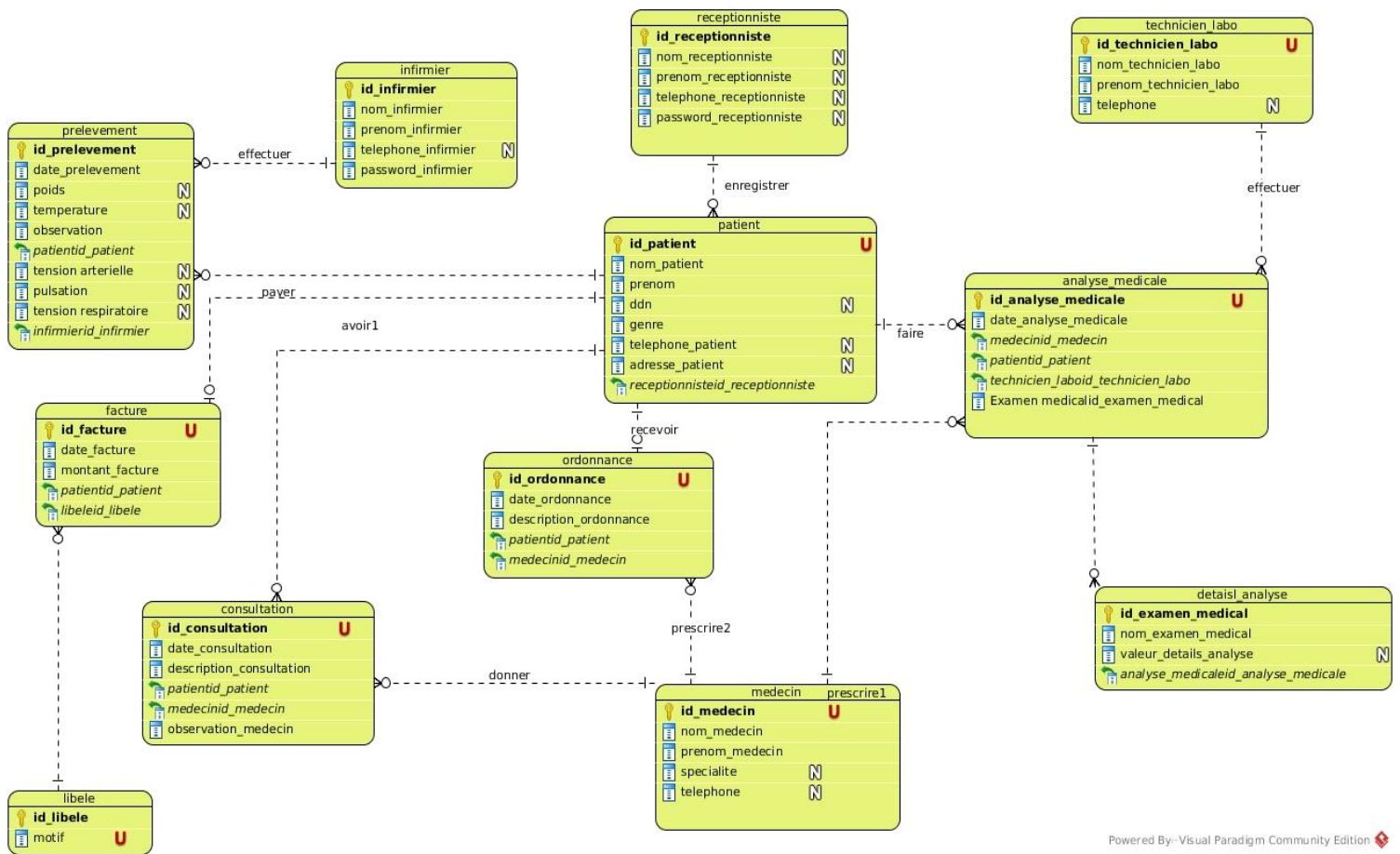
La présente partie explique les grandes lignes ayant guidé l'implémentation du logiciel pour les différentes actions liées à la gestion des malades ainsi que les différentes fonctionnalités de ce logiciel. Ce dernier permet d'enregistrer les malades et de les suivre pendant leur hospitalisation et tout au long de leur existence, chaque fois qu'ils se référeront à la structure de soins concernée. Le diagramme des "use case" est présenté dans la première partie de l'application ainsi que le diagramme Entité-relation, une présentation de quelques scénarios d'utilisation du simulateur est

faite dans la dernière partie.

### 5.7.1 Diagramme de cas d'utilisation



## 5.7.2 Diagramme Entité-relation



Powered By: Visual Paradigm Community Edition

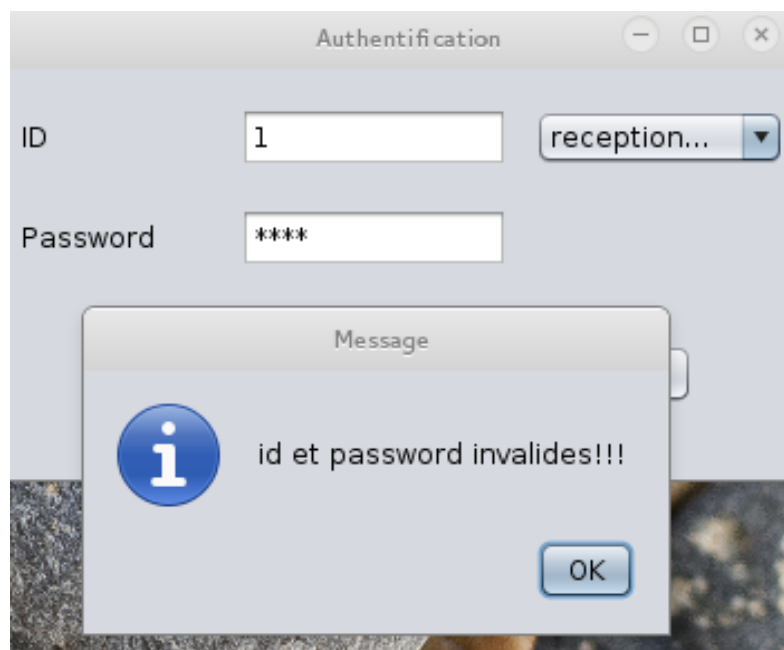
## 5.7.3 Schéma fonctionnel des interfaces

Sous ce sous point, nous présentons certaines interfaces et quelques explications sur leurs fonctionnalités.



Après l'installation et le déploiement de l'application, la première fenêtre s'ouvrant est celle de l'authentification ; l'utilisateur devra insérer son id et son mot de passe sans oublier de choisir sa fonction dans une combo-box.

Si l'id ou le mot de passe ne correspond pas à un compte valide, un message s'affiche pour signaler qu'il y a erreur. L'utilisateur devra alors soit bien écrire ses informations de login, soit faire attention lors de la sélection de son service.



Autrement, il devra s'enregistrer en sélectionnant le service concerné sur les différents services proposés se trouvant sur des formulaires.

The image shows a screenshot of a registration form window titled "Enregistrement medecin". The form consists of several input fields: "Nom", "Prenom", "Telephone", "Specialite", and "Password". The "Specialite" field is a large text area with a scrollbar. At the bottom of the form is a button labeled "Enregi...".

Selon la fonction, chaque service a sa fenêtre regroupant les fonctionnalités lui relatives. Le réceptionniste pourra rechercher un malade et l'afficher dans la "jtable" pour consulter ses informations. Il peut aussi enregistrer un patient avec toutes les informations le concernant, et les modifier en cas d'erreur de frappe.

Id	Nom	Prenom	Date de naiss...	Telephone	Genre	Adresse
1	kakasu	christian	1993-10-02	+243994907...	M	Goma-Himbi1-...
2	kandanda	jean paul	1962-09-06	+243998088...	M	
3	c	c	2015-10-23	24355	H	r
4	dada	d	2014-02-05	24355	F	r
5	a	a	2015-10-23	1234	F	fff
6	d	d	2015-10-23	356	H	vfwv
7						

L'infirmier de la salle de triage pourra rechercher le malade, et ainsi ajouter des informations sur sa fiche et des observations éventuelles à envoyer au médecin pour consultation.

Id	Nom	Prenom	Date de nai...	Telephone	Genre	Adresse
1	kakasu	christian	1993-10-02	+24399490...	M	Goma-Himbi...
2	kandanda	jean paul	1962-09-06	+24399808...	M	
3	c	c	2015-10-23	24355	H	r
4	dada	d	2014-02-05	24355	F	r
5	a	a	2015-10-23	1234	F	fff
6	d	d	2015-10-23	356	H	vfwv

Poids  tension\_arterielle  T\_respiratoire

Temperature  pulsation  Infirmier\_id

**Observation**

Enregistrer

Le médecin pourra rechercher le patient et l'afficher dans la "jtable". Il pourra ensuite compléter son

observation, prescrire une analyse médicale et plus tard une ordonnance.



The screenshot shows a window titled "Medecin" with a search bar labeled "Recherche" and a search icon. Below the search bar is a table with the following data:

Id	Nom	Prenom	Date Nais...	Telephone	Genre
1	kakasu	christian	1993-10-02	+243994...	M
2	kandanda	jean paul	1962-09-06	+243998...	M
3	c	c	2015-10-23	24355	H
4	dada	d	2014-02-05	24355	F
5	a	a	2015-10-23	1234	F

Below the table is a section labeled "Observation" with a large empty text area. At the bottom of the window are two buttons: "Enregistrer" and "Ordonnance".

Pour des raisons de clarté et de simplicité, nous n'avons présenté que certaines fenêtres ; le projet ainsi que son rapport vont accompagner ce travail et seront remis.

## CHAPITRE 6 : CONCLUSION

Vu l'évolution du monde actuel, des technologies naissent et s'améliorent dans un seul but : rendre agréable le quotidien de l'humain. Dans le domaine médical, l'informatique se veut importante, utile et influente dans le sens où les établissements de soins, et particulièrement les hôpitaux font partie intégrante des structures qu'elle devra beaucoup aider et appuyer sur les plans de l'organisation, de prestations et de l'information. La gestion manuelle de l'information étant encore dominante dans la plupart des structures de soins, devra laisser la place à la gestion informatisée des données ; celle-ci a l'avantage de faciliter la collecte exhaustive des données en évitant le plus possible d'erreurs, permet le calcul et le traitement rapides et aisés des données. Elle rend l'analyse facile des données à partir des tableaux de sortie programmés. Ce travail a passé en revue les différents concepts d'ordre général utilisés dans le cadre de la conception d'un logiciel dans le développement de deux premiers chapitres, il a abordé la méthodologie de recherche au chapitre trois avant d'aborder les chapitres quatre et cinq consacrés à la collecte des données, leur analyse et discussion, ainsi que l'étude cas.

Au chapitre quatre, nous avons eu à déterminer le processus de la prise en charge du patient à partir de l'interview réalisée auprès des prestataires. Le chapitre cinq nous a permis de relever certaines insuffisances rencontrées dans le système trouvé en place. Il offre à la fin une analyse informatique succincte de l'ancien système, en relevant certaines insuffisances et proposant de solutions palliatives.

Le présent travail a mis en exergue quelques difficultés auxquelles est buté le personnel infirmier et médical en matière de la gestion des données sur la prise en charge des malades au niveau des structures hospitalières, cela par différents processus et circulation des informations entre les acteurs.

Le logiciel présenté est en version 1.0 ; il reste néanmoins certaines améliorations à faire pour une bonne interaction, aussi sa production pour les différents Os.

---

## Bibliographie

1. CLAUDE TAKENGA, Informatique de gestion, ULPGL/FSTA, 2015
2. K KYAMAKYA, Cours de bases de données et systèmes d'information, Génie informatique/Université de Kinshasa
3. Guézélou, P. Modélisation des données : Approche pour la conception des bases des données. (<http://philippe.guezelou.free.fr/mcd/mcd.htm>),2006.
4. Kevin E. Kline - Daniel Kline et Brand Hunt, SQL in a Nutshell, O'Reilly Media, Inc. - 2004 (ISBN 978-0-596-00481-1).
5. Laurent Audibert, Bases de données - de la modélisation au SQL, Ellipses, 2009

## Annexe : Code Java implémentant les classes du logiciel

```

/*wqnt
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author christian
 */
public class Centrale {

    Connection con = SingletonConnection.getConnection();
    public Statement st = con.createStatement();
    public ResultSet rs;

    public Centrale() throws SQLException {
        //Connection con = SingletonConnection.getConnection();
        //Statement st = con.createStatement();
    }

    public boolean update(Object ob, String table, int cond) throws SQLException {
        try {
            if (ob instanceof patient && table.equalsIgnoreCase("patient")) {
                patient p = (patient) ob;
                return st.executeUpdate("update " + table + " set nom_patient=" + p.getNom_patient() +
                    ",prenom_patient=" + p.getPrenom_patient() + ",date_naissance=" + p.getDate_naissance() +
                    ",genre=" + p.getGenre() + ",telephone_patient=" + p.getTelephone_patient() +
                    ",adresse_patient=" + p.getAdresse_patient() + " where id_patient= " + cond + " ") != 0;
            } else if (ob instanceof analyse_medicale && table.equalsIgnoreCase("analyse_medicale"))
            {
                analyse_medicale ana = (analyse_medicale) ob;

```

```

        return st.executeUpdate("update " + table + " set date_analyse_medicale=" +
ana.getDate_analyse_medicale() + ",medecinid_medecin=" + ana.getMedecinid_medecin() + ",
patientid_patient=" + ana.getPatientid_patient() + ",technicien_laboid_technicien_labo=" +
ana.getTechnicien_laboid_technicien_labo() + "where id_analyse_medicale= " + cond + ") != 0;

    } else if (ob instanceof consultation && table.equalsIgnoreCase("consultation")) {
        consultation q = (consultation) ob;
        return st.executeUpdate("update " + table + " set date_consultation=" +
q.getDate_consultation() + ",description_consultation=" + q.getDescription_consultation() +
",patientid_patient=" + q.getPatientid_patient() + ",medecinid_medecin=" +
q.getMedecinid_medecin() + "where id_consultation= " + cond + ") != 0;

    } else if (ob instanceof facture && table.equalsIgnoreCase("facture")) {
        facture r = (facture) ob;
        return st.executeUpdate("update " + table + " set date_facture=" + r.getDate_facture() +
",montant_facture=" + r.getMontant_facture() + ",patientid_patient=" + r.getPatientid_patient() +
" where id_facture= " + cond + ") != 0;

    } else if (ob instanceof infirmier && table.equalsIgnoreCase("infirmier")) {
        infirmier s = (infirmier) ob;
        return st.executeUpdate("update " + table + " set nom_infirmier=" +
s.getNom_infirmier() + ",prenom_infirmier=" + s.getPrenom_infirmier() + ",
telephone_infirmier=" + s.getTelephone_infirmier() + ",password_infirmier;\n"
+ "=" + s.getPassword_infirmier() + "where id_infirmier= " + cond + ") != 0;

    } else if (ob instanceof medecin && table.equalsIgnoreCase("medecin")) {
        medecin t = (medecin) ob;
        return st.executeUpdate("update " + table + " set nom_medecin=" + t.getNom_medecin()
+ ", prenom_medecin=" + t.getPrenom_medecin() + ", specialite=" + t.getSpecialite() + ",
telephone_medecin=" + t.getTelephone_medecin() + ", password_medecin=" +
t.getPassword_medecin() + "where id_medecin= " + cond + ") != 0;

    } else if (ob instanceof ordonnance && table.equalsIgnoreCase("ordonnance")) {
        ordonnance u = (ordonnance) ob;
        return st.executeUpdate("update " + table + " set date_ordonnance=" +
u.getDate_ordonnance() + " description_ordonnance=" + u.getDescription_ordonnance() +
",patientid_patient=" + u.getPatientid_patient() + ",medecinid_medecin=" +
u.getMedecinid_medecin() + "where id_ordonnance= " + cond + ") != 0;

    } else if (ob instanceof prelevement && table.equalsIgnoreCase("prelevement")) {
        prelevement v = (prelevement) ob;
        return st.executeUpdate("update " + table + " set date_prelevement=" +
v.getDate_prelevement() + ",poids=" + v.getPoids() + ", temperature=" + v.getTemperature() + ",
observation=" + v.getObservation() + ",tension_arterielle=" + v.getTension_arterielle() +
",pulsation=" + v.getPulsation() + ",tension_respiratoire=" + v.getTension_respiratoire() + ",
infirmierid_infirmier=" + v.getInfirmierid_infirmier() + ",patientid_patient=" +
v.getPatientid_patient() + "where id_prelevement= " + cond + ") != 0;

    } else if (ob instanceof receptionniste && table.equalsIgnoreCase("receptionniste")) {
        receptionniste w = (receptionniste) ob;

```

```

        return st.executeUpdate("update " + table + " set nom_receptionniste=" +
w.getNom_receptionniste() + ",prenom_receptionniste=" + w.getPrenom_receptionniste() +
",telephone_receptionniste=" + w.getTelephone_receptionniste() + ", password_receptionniste="
+ w.getPassword_receptionniste() + "where id_receptionniste= " + cond + "'") != 0;

    } else if (ob instanceof technicien_lab0 && table.equalsIgnoreCase("technicien_lab0")) {
        technicien_lab0 p = (technicien_lab0) ob;
        return st.executeUpdate("update " + table + " set nom_technicien_lab0=" +
p.getNom_technicien_lab0() + ", prenom_technicien_lab0=" + p.getPrenom_technicien_lab0() +
", telephone_technicien_lab0=" + p.getTelephone_technicien_lab0() +
",password_technicien_lab0=" + p.getPassword_technicien_lab0() + "where id_technicien_lab0= "
+ cond + "'") != 0;

    } else if (ob instanceof examen && table.equalsIgnoreCase("examen")) {
        examen x = (examen) ob;
        return st.executeUpdate("update " + table + " set nom_examen=" + x.getNom_examen()
+ "where id_prelevement= " + cond + "'") != 0;

    } else if (ob instanceof details_analyse && table.equalsIgnoreCase("details_analyse")) {
        details_analyse y = (details_analyse) ob;
        return st.executeUpdate("update " + table + " set valeur_details_analyse=" +
y.getValeur_details_analyse() + ", analyse_medicaleid_analyse_medicale=" +
y.getAnalyse_medicaleid_analyse_medicale() + ",examenid_examen=" +
y.getExamenid_examen() + "where id_examen= " + cond + "'") != 0;

    }

} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e.getMessage());
}
st.close();
return false;
}

public boolean inserer(Object ob, String table) throws SQLException {
    try {
        if (ob instanceof analyse_medicale && table.equalsIgnoreCase("analyse_medicale")) {
            analyse_medicale z = (analyse_medicale) ob;
            return st.executeUpdate("insert into " + table + "
(date_analyse_medicale,medecinid_medecin,patientid_patient,technicien_laboid_technicien_lab0)
values (" + z.getDate_analyse_medicale() + "," + z.getMedecinid_medecin() + "," +
z.getPatientid_patient() + "," + z.getTechnicien_laboid_technicien_lab0() + ")") != 0;

        } else if (ob instanceof consultation && table.equalsIgnoreCase("consultation")) {
            consultation a = (consultation) ob;
            return st.executeUpdate("insert into " + table +
"(date_consultation,description_consultation,patientid_patient,medecinid_medecin) values (" +
a.getDate_consultation() + "," + a.getDescription_consultation() + "," + a.getPatientid_patient() +
"," + a.getMedecinid_medecin() + ")") != 0;

        } else if (ob instanceof facture && table.equalsIgnoreCase("facture")) {

```

```

        facture b = (facture) ob;
        return st.executeUpdate("insert into " + table + "
(date_facture,montant_facture,patientid_patient) values (" + b.getDate_facture() + "," +
b.getMontant_facture() + "," + b.getPatientid_patient() + ")") != 0;
    } else if (ob instanceof infirmier && table.equalsIgnoreCase("infirmier")) {
        infirmier c = (infirmier) ob;
        return st.executeUpdate("insert into " + table + "
(nom_infirmier,prenom_infirmier,telephone_infirmier,password_infirmier) values (" +
c.getNom_infirmier() + "," + c.getPrenom_infirmier() + "," + c.getTelephone_infirmier() + "," +
c.getPassword_infirmier() + ")") != 0;
    } else if (ob instanceof medecin && table.equalsIgnoreCase("medecin")) {
        medecin d = (medecin) ob;
        return st.executeUpdate("insert into " + table + "
(nom_medecin,prenom_medecin,specialite,telephone_medecin,password_medecin) values (" +
d.getNom_medecin() + "," + d.getPrenom_medecin() + "," + d.getSpecialite() + "," +
d.getTelephone_medecin() + "," + d.getPassword_medecin() + ")") != 0;
    } else if (ob instanceof ordonnance && table.equalsIgnoreCase("ordonnance")) {
        ordonnance e = (ordonnance) ob;
        return st.executeUpdate("insert into " + table + "
(date_ordonnance,description_ordonnance,patientid_patient,medecinid_medecin) values (" +
e.getDate_ordonnance() + "," + e.getDescription_ordonnance() + "," + e.getPatientid_patient() +
"," + e.getMedecinid_medecin() + ")") != 0;
    } else if (ob instanceof patient && table.equalsIgnoreCase("patient")) {
        patient f = (patient) ob;
        return st.executeUpdate("insert into " + table + "
(nom_patient,prenom_patient,date_naissance,genre,telephone_patient,adresse_patient) values (" +
f.getNom_patient() + "," + f.getPrenom_patient() + "," + f.getDate_naissance() + "," +
f.getGenre() + "," + f.getTelephone_patient() + "," + f.getAdresse_patient() + ")") != 0;
    } else if (ob instanceof prelevement && table.equalsIgnoreCase("prelevement")) {
        prelevement g = (prelevement) ob;
        return st.executeUpdate("insert into " + table + "
(date_prelevement,poids,temperature,observation,tension_arterielle,pulsation,tension_respiratoire,in
firmierid_infirmier,patientid_patient) values (" + g.getDate_prelevement() + "," + g.getPoids() +
"," + g.getTemperature() + "," + g.getObservation() + "," + g.getTension_arterielle() + "," +
g.getPulsation() + "," + g.getTension_respiratoire() + "," + g.getInfirmierid_infirmier() + "," +
g.getPatientid_patient() + ")") != 0;
    } else if (ob instanceof receptionniste && table.equalsIgnoreCase("receptionniste")) {
        receptionniste h = (receptionniste) ob;
        return st.executeUpdate("insert into " + table + "
(nom_receptionniste,prenom_receptionniste,telephone_receptionniste,password_receptionniste)
values (" + h.getNom_receptionniste() + "," + h.getPrenom_receptionniste() + "," +
h.getTelephone_receptionniste() + "," + h.getPassword_receptionniste() + ")") != 0;
    } else if (ob instanceof technicien_lab0 && table.equalsIgnoreCase("technicien_lab0")) {
        technicien_lab0 i = (technicien_lab0) ob;
        return st.executeUpdate("insert into " + table + "
(nom_technicien_lab0,prenom_technicien_lab0,telephone_technicien_lab0,password_technicien_la
bo) values (" + i.getNom_technicien_lab0() + "," + i.getPrenom_technicien_lab0() + "," +
i.getTelephone_technicien_lab0() + "," + i.getPassword_technicien_lab0() + ")") != 0;
    } else if (ob instanceof examen && table.equalsIgnoreCase("examen")) {
        examen j = (examen) ob;

```

```

        return st.executeUpdate("insert into " + table + " (nom_examen) values (" +
j.getNom_examen() + ")") != 0;
    } else if (ob instanceof details_analyse && table.equalsIgnoreCase("details_analyse")) {
        details_analyse k = (details_analyse) ob;
        return st.executeUpdate("insert into " +
(valeur_details_analyse,analyse_medicaleid_analyse_medicale,examenid_examen) values (" +
k.getValeur_details_analyse() + "," + k.getAnalyse_medicaleid_analyse_medicale() + "," +
k.getExamenid_examen() + ")") != 0;
    }

    } catch (Exception e) {
    }
    st.close();
    return false;
}

public boolean effacer(String table, String condition) throws SQLException {
    if (table.equalsIgnoreCase("analyse_medicale")) {
        return st.executeUpdate("delete from " + table + " where id_analyse_medicale=" +
condition + """) != 0;
    } else if (table.equalsIgnoreCase("consultation")) {
        return st.executeUpdate("delete from " + table + " where id_consultation=" + condition +
""") != 0;
    } else if (table.equalsIgnoreCase("details_analyse")) {
        return st.executeUpdate("delete from " + table + " where id_details_analyse=" + condition +
""") != 0;
    } else if (table.equalsIgnoreCase("examen")) {
        return st.executeUpdate("delete from " + table + " where id_examen=" + condition + """) !=
0;
    } else if (table.equalsIgnoreCase("facture")) {
        return st.executeUpdate("delete from " + table + " where id_facture=" + condition + """) !=
0;
    } else if (table.equalsIgnoreCase("infirmier")) {
        return st.executeUpdate("delete from " + table + " where id_infirmier=" + condition + """)
!= 0;
    } else if (table.equalsIgnoreCase("libele")) {
        return st.executeUpdate("delete from " + table + " where id_libele=" + condition + """) != 0;
    } else if (table.equalsIgnoreCase("libele")) {
        return st.executeUpdate("delete from " + table + " where id_libele=" + condition + """) != 0;
    } else if (table.equalsIgnoreCase("medecin")) {
        return st.executeUpdate("delete from " + table + " where id_medecin=" + condition + """) !=
0;
    } else if (table.equalsIgnoreCase("ordonnance")) {
        return st.executeUpdate("delete from " + table + " where id_ordonnance=" + condition +
""") != 0;
    } else if (table.equalsIgnoreCase("patient")) {
        return st.executeUpdate("delete from " + table + " where id_patient=" + condition + """) !=
0;
    } else if (table.equalsIgnoreCase("prelevement")) {
        return st.executeUpdate("delete from " + table + " where id_prelevement=" + condition +

```

```

""") != 0;
    } else if (table.equalsIgnoreCase("receptionniste")) {
        return st.executeUpdate("delete from " + table + " where id_receptionniste=" + condition +
""") != 0;
    } else if (table.equalsIgnoreCase("technicien_labo")) {
        return st.executeUpdate("delete from " + table + " where id_technicien_labo=" + condition
+ """) != 0;
    }

    return false;
}

public List<patient> recherchePatient(String mc) throws SQLException {
    List<patient> pat = new ArrayList<>();
    PreparedStatement ps = con.prepareStatement("select * from patient where nom_patient like ?
");
    ps.setString(1, "%" + mc + "%");

    ResultSet rs = ps.executeQuery();
    while (rs.next()) {
        patient p = new patient();
        p.setId_patient(rs.getString("id_patient"));
        p.setNom_patient(rs.getString("nom_patient"));
        p.setPrenom_patient(rs.getString("prenom_patient"));
        p.setDate_naissance(rs.getString("date_naissance"));
        p.setGenre(rs.getString("genre"));
        p.setTelephone_patient(rs.getString("telephone_patient"));
        p.setAdresse_patient(rs.getString("adresse_patient"));
        pat.add(p);
    }

    ps.close();
    return pat;
}

public List<Examen> rechercheex(String m) throws SQLException {
    List<Examen> ex = new ArrayList<>();
    PreparedStatement ps = con.prepareStatement("select * from examen where nom_examen like
? ");
    ps.setString(1, "%" + m + "%");
    ResultSet rs = ps.executeQuery();
    while (rs.next()) {
        Examen examen = new Examen();
        examen.setNom_examen(rs.getString("nom_examen"));
        ex.add(examen);
    }
    ps.close();
    return ex;
}

```

```

public void initData(JTable jt) {
    try {
        SingletonConnection co = new SingletonConnection();
        Connection connection = co.getConnection();
        Statement st = connection.createStatement();
        String request = "SELECT * FROM patient";
        ResultSet rs = st.executeQuery(request);
        while (rs.next()) {
            DefaultTableModel model = (DefaultTableModel) jt.getModel();
            model.addRow(new Object[]{rs.getString("id_patient"), rs.getString("nom_patient"),
rs.getString("prenom_patient"), rs.getString("date_naissance"), rs.getString("telephone_patient"),
rs.getString("genre"), rs.getString("adresse_patient")});
        }

    } catch (Exception e) {

    }
}

public void initDataex(JTable jt) {
    try {
        SingletonConnection co = new SingletonConnection();
        Connection connection = co.getConnection();
        Statement st = connection.createStatement();
        String request = "SELECT * FROM examen";
        ResultSet rs = st.executeQuery(request);
        while (rs.next()) {
            DefaultTableModel model = (DefaultTableModel) jt.getModel();
            model.addRow(new Object[]{rs.getString("id_examen"), rs.getString("nom_examen")});
        }

    } catch (Exception e) {

    }
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

/**
 *
 * @author christian
 */

```

```
public class Examen {

    private String nom_examen;
    private int id_examen, categorieid_categorie;

    public Examen() {
    }

    public Examen(String nom_examen, int id_examen, int categorieid_categorie) {
        this.nom_examen = nom_examen;
        this.id_examen = id_examen;
        this.categorieid_categorie = categorieid_categorie;
    }

    public String getNom_examen() {
        return nom_examen;
    }

    public int getId_examen() {
        return id_examen;
    }

    public int getCategorieid_categorie() {
        return categorieid_categorie;
    }

    public void setNom_examen(String nom_examen) {
        this.nom_examen = nom_examen;
    }

    public void setId_examen(int id_examen) {
        this.id_examen = id_examen;
    }

    public void setCategorieid_categorie(int categorieid_categorie) {
        this.categorieid_categorie = categorieid_categorie;
    }

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
```

```
import java.util.logging.Level;
import java.util.logging.Logger;
/**
 *
 * @author christian
 */
public class SingletonConnection {
    private static Connection connection;

    static {
        try {
            // Class.forName("com.jdbc.mysql.Driver");

            try {
                connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/Hopital",
"root", "dell");
                System.out.println("connection etablie A la BD");

            } catch (SQLException e) {
                System.out.println("Aucune connection A la BD"+e);
            }
        } catch (Exception ex) {
            Logger.getLogger(SingletonConnection.class.getName()).log(Level.SEVERE, null, ex);
        }
    }

    public static Connection getConnection() {
        return connection;
    }
}

/**
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;
import java.sql.Date;

/**
 *
 * @author christian
 */
public class analyse_medicale implements Serializable {
```

```
private int id_analyse_medicale;
private String date_analyse_medicale;

private int medecinid_medecin;
private int patientid_patient;
private int technicien_laboid_technicien_lab;

public analyse_medicale() {
}

public analyse_medicale(String date_analyse_medicale, int medecinid_medecin, int
patientid_patient) {
    this.date_analyse_medicale = date_analyse_medicale;
    this.medecinid_medecin = medecinid_medecin;
    this.patientid_patient = patientid_patient;
}

public analyse_medicale(String date_analyse_medicale, int medecinid_medecin, int
patientid_patient, int technicien_laboid_technicien_lab) {
    this.date_analyse_medicale = date_analyse_medicale;

    this.medecinid_medecin = medecinid_medecin;
    this.patientid_patient = patientid_patient;
    this.technicien_laboid_technicien_lab = technicien_laboid_technicien_lab;
}

public int getId_analyse_medicale() {
    return id_analyse_medicale;
}

public String getDate_analyse_medicale() {
    return date_analyse_medicale;
}

public int getMedecinid_medecin() {
    return medecinid_medecin;
}

public int getPatientid_patient() {
    return patientid_patient;
}

public int getTechnicien_laboid_technicien_lab() {
    return technicien_laboid_technicien_lab;
}

public void setId_analyse_medicale(int id_analyse_medicale) {
    this.id_analyse_medicale = id_analyse_medicale;
}
```

```
public void setDate_analyse_medicale(String date_analyse_medicale) {
    this.date_analyse_medicale = date_analyse_medicale;
}

public void setMedecinid_medecin(int medecinid_medecin) {
    this.medecinid_medecin = medecinid_medecin;
}

public void setPatientid_patient(int patientid_patient) {
    this.patientid_patient = patientid_patient;
}

public void setTechnicien_laboid_technicien_laboid(int technicien_laboid_technicien_laboid) {
    this.technicien_laboid_technicien_laboid = technicien_laboid_technicien_laboid;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;
import java.sql.Date;

/**
 *
 * @author christian
 */
public class consultation implements Serializable {

    private int id_consultation;
    private String date_consultation;
    private String description_consultation;
    private int patientid_patient;
    private int medecinid_medecin;

    public consultation() {
    }

    public consultation(String date_consultation, String description_consultation, int
patientid_patient, int medecinid_medecin) {
        this.date_consultation = date_consultation;
        this.description_consultation = description_consultation;
        this.patientid_patient = patientid_patient;
        this.medecinid_medecin = medecinid_medecin;
    }
}
```

```
}

public int getId_consultation() {
    return id_consultation;
}

public String getDate_consultation() {
    return date_consultation;
}

public String getDescription_consultation() {
    return description_consultation;
}

public int getPatientid_patient() {
    return patientid_patient;
}

public int getMedecinid_medecin() {
    return medecinid_medecin;
}

public void setId_consultation(int id_consultation) {
    this.id_consultation = id_consultation;
}

public void setDate_consultation(String date_consultation) {
    this.date_consultation = date_consultation;
}

public void setDescription_consultation(String description_consultation) {
    this.description_consultation = description_consultation;
}

public void setPatientid_patient(int patientid_patient) {
    this.patientid_patient = patientid_patient;
}

public void setMedecinid_medecin(int medecinid_medecin) {
    this.medecinid_medecin = medecinid_medecin;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;
```

```
import java.io.Serializable;

/**
 *
 * @author christian
 */
public class details_analyse implements Serializable {

    private int id_details_analyse;
    private String valeur_details_analyse;
    private int analyse_medicaleid_analyse_medicale;
    private int examenid_examen;

    public details_analyse() {
    }

    public details_analyse(String valeur_details_analyse, int analyse_medicaleid_analyse_medicale,
int examenid_examen) {

        this.valeur_details_analyse = valeur_details_analyse;
        this.analyse_medicaleid_analyse_medicale = analyse_medicaleid_analyse_medicale;
        this.examenid_examen = examenid_examen;
    }

    public int getId_details_analyse() {
        return id_details_analyse;
    }

    public String getValeur_details_analyse() {
        return valeur_details_analyse;
    }

    public int getAnalyse_medicaleid_analyse_medicale() {
        return analyse_medicaleid_analyse_medicale;
    }

    public int getExamenid_examen() {
        return examenid_examen;
    }

    public void setId_details_analyse(int id_details_analyse) {
        this.id_details_analyse = id_details_analyse;
    }

    public void setValeur_details_analyse(String valeur_details_analyse) {
        this.valeur_details_analyse = valeur_details_analyse;
    }

    public void setAnalyse_medicaleid_analyse_medicale(int analyse_medicaleid_analyse_medicale)
```

```
{
    this.analyse_medicaleid_analyse_medicale = analyse_medicaleid_analyse_medicale;
}

public void setExamenid_examen(int examenid_examen) {
    this.examenid_examen = examenid_examen;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;

/**
 *
 * @author christian
 */
public class examen implements Serializable {
    private int id_examen;
    private String nom_examen;

    public examen() {
    }

    public examen(String nom_examen) {
        this.nom_examen = nom_examen;
    }

    public int getId_examen() {
        return id_examen;
    }

    public String getNom_examen() {
        return nom_examen;
    }

    public void setId_examen(int id_examen) {
        this.id_examen = id_examen;
    }

    public void setNom_examen(String nom_examen) {
        this.nom_examen = nom_examen;
    }
}
```

```
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;
import java.sql.Date;

/**
 *
 * @author christian
 */
public class facture implements Serializable {

    private int id_facture;
    private Date date_facture;
    private int montant_facture;
    private int patientid_patient;

    public facture() {
    }

    public facture(Date date_facture, int montant_facture, int patientid_patient) {
        this.date_facture = date_facture;
        this.montant_facture = montant_facture;
        this.patientid_patient = patientid_patient;
    }

    public int getId_facture() {
        return id_facture;
    }

    public Date getDate_facture() {
        return date_facture;
    }

    public int getMontant_facture() {
        return montant_facture;
    }

    public int getPatientid_patient() {
        return patientid_patient;
    }
}
```

```
public void setId_facture(int id_facture) {
    this.id_facture = id_facture;
}

public void setDate_facture(Date date_facture) {
    this.date_facture = date_facture;
}

public void setMontant_facture(int montant_facture) {
    this.montant_facture = montant_facture;
}

public void setPatientid_patient(int patientid_patient) {
    this.patientid_patient = patientid_patient;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;

/**
 *
 * @author christian
 */
public class infirmier implements Serializable {

    private int id_infirmier;
    private String nom_infirmier;
    private String prenom_infirmier;
    private String telephone_infirmier;
    private String password_infirmier;

    public infirmier() {
    }

    public infirmier(String nom_infirmier, String prenom_infirmier, String telephone_infirmier,
String password_infirmier) {
        this.nom_infirmier = nom_infirmier;
        this.prenom_infirmier = prenom_infirmier;
        this.telephone_infirmier = telephone_infirmier;
        this.password_infirmier = password_infirmier;
    }
}
```

```
public int getId_infirmier() {
    return id_infirmier;
}

public String getNom_infirmier() {
    return nom_infirmier;
}

public String getPrenom_infirmier() {
    return prenom_infirmier;
}

public String getTelephone_infirmier() {
    return telephone_infirmier;
}

public String getPassword_infirmier() {
    return password_infirmier;
}

public void setId_infirmier(int id_infirmier) {
    this.id_infirmier = id_infirmier;
}

public void setNom_infirmier(String nom_infirmier) {
    this.nom_infirmier = nom_infirmier;
}

public void setPrenom_infirmier(String prenom_infirmier) {
    this.prenom_infirmier = prenom_infirmier;
}

public void setTelephone_infirmier(String telephone_infirmier) {
    this.telephone_infirmier = telephone_infirmier;
}

public void setPassword_infirmier(String password_infirmier) {
    this.password_infirmier = password_infirmier;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;
```

```
import java.io.Serializable;

/**
 *
 * @author christian
 */
public class medecin implements Serializable{

    private int id_medecin;
    private String nom_medecin;
    private String prenom_medecin;
    private String specialite;
    private String telephone_medecin;
    private String password_medecin;

    public medecin() {
    }

    public medecin(String nom_medecin, String prenom_medecin, String specialite, String
telephone_medecin, String password_medecin) {

        this.nom_medecin = nom_medecin;
        this.prenom_medecin = prenom_medecin;
        this.specialite = specialite;
        this.telephone_medecin = telephone_medecin;
        this.password_medecin = password_medecin;
    }

    public int getId_medecin() {
        return id_medecin;
    }

    public String getNom_medecin() {
        return nom_medecin;
    }

    public String getPrenom_medecin() {
        return prenom_medecin;
    }

    public String getSpecialite() {
        return specialite;
    }

    public String getTelephone_medecin() {
        return telephone_medecin;
    }

    public String getPassword_medecin() {
        return password_medecin;
    }
}
```

```
}

public void setId_medecin(int id_medecin) {
    this.id_medecin = id_medecin;
}

public void setNom_medecin(String nom_medecin) {
    this.nom_medecin = nom_medecin;
}

public void setPrenom_medecin(String prenom_medecin) {
    this.prenom_medecin = prenom_medecin;
}

public void setSpecialite(String specialite) {
    this.specialite = specialite;
}

public void setTelephone_medecin(String telephone_medecin) {
    this.telephone_medecin = telephone_medecin;
}

public void setPassword_medecin(String password_medecin) {
    this.password_medecin = password_medecin;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;
import java.sql.Date;

/**
 *
 * @author christian
 */
public class ordonnance implements Serializable {

    private int id_ordonnance;
    private Date date_ordonnance;
    private String description_ordonnance;
    private int patientid_patient;
    private int medecinid_medecin;
```

```
public ordonnance() {
}

public ordonnance(Date date_ordonnance, String description_ordonnance, int patientid_patient,
int medecinid_medecin) {
    this.date_ordonnance = date_ordonnance;
    this.description_ordonnance = description_ordonnance;
    this.patientid_patient = patientid_patient;
    this.medecinid_medecin = medecinid_medecin;
}

public int getId_ordonnance() {
    return id_ordonnance;
}

public Date getDate_ordonnance() {
    return date_ordonnance;
}

public String getDescription_ordonnance() {
    return description_ordonnance;
}

public int getPatientid_patient() {
    return patientid_patient;
}

public int getMedecinid_medecin() {
    return medecinid_medecin;
}

public void setId_ordonnance(int id_ordonnance) {
    this.id_ordonnance = id_ordonnance;
}

public void setDate_ordonnance(Date date_ordonnance) {
    this.date_ordonnance = date_ordonnance;
}

public void setDescription_ordonnance(String description_ordonnance) {
    this.description_ordonnance = description_ordonnance;
}

public void setPatientid_patient(int patientid_patient) {
    this.patientid_patient = patientid_patient;
}

public void setMedecinid_medecin(int medecinid_medecin) {
    this.medecinid_medecin = medecinid_medecin;
}
```

```
    }  
}  
  
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package metier;  
  
import java.io.*;  
import java.util.Date;  
  
//import java.sql.Date;  
/**  
 *  
 * @author christian  
 */  
public class patient implements Serializable {  
  
    private String id_patient;  
    private String nom_patient;  
    private String prenom_patient;  
    private String date_naissance;  
    private String genre;  
    private String telephone_patient;  
    private String adresse_patient;  
  
    public patient() {  
    }  
  
    public patient(String nom_patient, String prenom_patient, String date_naissance, String genre,  
String telephone_patient, String adresse_patient) {  
  
        this.nom_patient = nom_patient;  
        this.prenom_patient = prenom_patient;  
        this.date_naissance = date_naissance;  
        this.genre = genre;  
        this.telephone_patient = telephone_patient;  
        this.adresse_patient = adresse_patient;  
    }  
  
    public String getAdresse_patient() {  
        return adresse_patient;  
    }  
  
    public String getId_patient() {  
        return id_patient;  
    }  
}
```

```
}

public void setAdresse_patient(String adresse_patient) {
    this.adresse_patient = adresse_patient;
}

public String getNom_patient() {
    return nom_patient;
}

public String getPrenom_patient() {
    return prenom_patient;
}

public String getDate_naissance() {
    return date_naissance;
}

public String getGenre() {
    return genre;
}

public String getTelephone_patient() {
    return telephone_patient;
}

public void setId_patient(String id_patient) {
    this.id_patient = id_patient;
}

public void setNom_patient(String nom_patient) {
    this.nom_patient = nom_patient;
}

public void setPrenom_patient(String prenom_patient) {
    this.prenom_patient = prenom_patient;
}

public void setDate_naissance( String date_naissance) {
    this.date_naissance = date_naissance;
}

public void setGenre(String genre) {
    this.genre = genre;
}

public void setTelephone_patient(String telephone_patient) {
    this.telephone_patient = telephone_patient;
}
```

```
@Override
public String toString() {
    return "patient{" + "id_patient=" + id_patient + ", nom_patient=" + nom_patient + ",
prenom_patient=" + prenom_patient + ", date_naissance=" + date_naissance + ", genre=" + genre +
", telephone_patient=" + telephone_patient + ", adresse_patient=" + adresse_patient + '}';
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;
import java.sql.Date;

/**
 *
 * @author christian
 */
public class prelevement implements Serializable {

    private int id_prelevement;
    private String date_prelevement;
    private String poids;
    private String temperature;
    private String observation;
    private String tension_arterielle;
    private String pulsation;
    private String tension_respiratoire;
    private String infirmierid_infirmier;
    private int patientid_patient;

    public prelevement() {
    }

    public prelevement(String date_prelevement, String poids, String temperature, String
observation, String tension_arterielle, String pulsation, String tension_respiratoire, String
infirmierid_infirmier, int patientid_patient) {
        this.date_prelevement = date_prelevement;
        this.poids = poids;
        this.temperature = temperature;
        this.observation = observation;
        this.tension_arterielle = tension_arterielle;
        this.pulsation = pulsation;
    }
}
```

```
        this.tension_respiratoire = tension_respiratoire;
        this.infirmierid_infirmier = infirmierid_infirmier;
        this.patientid_patient = patientid_patient;
    }

    public int getId_prelevement() {
        return id_prelevement;
    }

    public String getDate_prelevement() {
        return date_prelevement;
    }

    public String getPoids() {
        return poids;
    }

    public String getTemperature() {
        return temperature;
    }

    public String getObservation() {
        return observation;
    }

    public String getTension_arterielle() {
        return tension_arterielle;
    }

    public String getPulsation() {
        return pulsation;
    }

    public String getTension_respiratoire() {
        return tension_respiratoire;
    }

    public String getInfirmierid_infirmier() {
        return infirmierid_infirmier;
    }

    public int getPatientid_patient() {
        return patientid_patient;
    }

    public void setId_prelevement(int id_prelevement) {
        this.id_prelevement = id_prelevement;
    }

    public void setDate_prelevement(String date_prelevement) {
```

```
        this.date_prelevement = date_prelevement;
    }

    public void setPoids(String poids) {
        this.poids = poids;
    }

    public void setTemperature(String temperature) {
        this.temperature = temperature;
    }

    public void setObservation(String observation) {
        this.observation = observation;
    }

    public void setTension_arterielle(String tension_arterielle) {
        this.tension_arterielle = tension_arterielle;
    }

    public void setPulsation(String pulsation) {
        this.pulsation = pulsation;
    }

    public void setTension_respiratoire(String tension_respiratoire) {
        this.tension_respiratoire = tension_respiratoire;
    }

    public void setInfirmierid_infirmier(String infirmierid_infirmier) {
        this.infirmierid_infirmier = infirmierid_infirmier;
    }

    public void setPatientid_patient(int patientid_patient) {
        this.patientid_patient = patientid_patient;
    }
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;

/**
 *
 * @author christian
 */
```

```
*/  
public class receptionniste implements Serializable {  
  
    private int id_receptionniste;  
    private String nom_receptionniste;  
    private String prenom_receptionniste;  
    private String telephone_receptionniste;  
    private String password_receptionniste;  
  
    public receptionniste() {  
    }  
  
    public receptionniste(String nom_receptionniste, String prenom_receptionniste, String  
telephone_receptionniste, String password_receptionniste) {  
        this.nom_receptionniste = nom_receptionniste;  
        this.prenom_receptionniste = prenom_receptionniste;  
        this.telephone_receptionniste = telephone_receptionniste;  
        this.password_receptionniste = password_receptionniste;  
    }  
  
    public int getId_receptionniste() {  
        return id_receptionniste;  
    }  
  
    public String getNom_receptionniste() {  
        return nom_receptionniste;  
    }  
  
    public String getPrenom_receptionniste() {  
        return prenom_receptionniste;  
    }  
  
    public String getTelephone_receptionniste() {  
        return telephone_receptionniste;  
    }  
  
    public String getPassword_receptionniste() {  
        return password_receptionniste;  
    }  
  
    public void setId_receptionniste(int id_receptionniste) {  
        this.id_receptionniste = id_receptionniste;  
    }  
  
    public void setNom_receptionniste(String nom_receptionniste) {  
        this.nom_receptionniste = nom_receptionniste;  
    }  
  
    public void setPrenom_receptionniste(String prenom_receptionniste) {  
        this.prenom_receptionniste = prenom_receptionniste;  
    }  
}
```

```
}

public void setTelephone_receptionniste(String telephone_receptionniste) {
    this.telephone_receptionniste = telephone_receptionniste;
}

public void setPassword_receptionniste(String password_receptionniste) {
    this.password_receptionniste = password_receptionniste;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package metier;

import java.io.Serializable;

/**
 *
 * @author christian
 */
public class technicien_labor implements Serializable{
    private int id_technicien_labor;
    private String nom_technicien_labor;
    private String prenom_technicien_labor;
    private String telephone_technicien_labor;
    private String password_technicien_labor;

    public technicien_labor() {
    }

    public technicien_labor(String nom_technicien_labor, String prenom_technicien_labor, String
telephone_technicien_labor, String password_technicien_labor) {
        this.nom_technicien_labor = nom_technicien_labor;
        this.prenom_technicien_labor = prenom_technicien_labor;
        this.telephone_technicien_labor = telephone_technicien_labor;
        this.password_technicien_labor = password_technicien_labor;
    }

    public int getId_technicien_labor() {
        return id_technicien_labor;
    }

    public String getNom_technicien_labor() {
        return nom_technicien_labor;
    }
}
```

```
}

public String getPrenom_technicien_lab() {
    return prenom_technicien_lab;
}

public String getTelephone_technicien_lab() {
    return telephone_technicien_lab;
}

public String getPassword_technicien_lab() {
    return password_technicien_lab;
}

public void setId_technicien_lab(int id_technicien_lab) {
    this.id_technicien_lab = id_technicien_lab;
}

public void setNom_technicien_lab(String nom_technicien_lab) {
    this.nom_technicien_lab = nom_technicien_lab;
}

public void setPrenom_technicien_lab(String prenom_technicien_lab) {
    this.prenom_technicien_lab = prenom_technicien_lab;
}

public void setTelephone_technicien_lab(String telephone_technicien_lab) {
    this.telephone_technicien_lab = telephone_technicien_lab;
}

public void setPassword_technicien_lab(String password_technicien_lab) {
    this.password_technicien_lab = password_technicien_lab;
}

}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
```

```
import metier.Centrale;
import metier.SingletonConnection;
import metier.infirmier;

/**
 *
 * @author christian
 */
public class Ainfirmier extends javax.swing.JFrame {

    Connection con = SingletonConnection.getConnection();

    /**
     * Creates new form Ainfirmier
     */
    public Ainfirmier() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jtfnom_infirmier = new javax.swing.JTextField();
        jtfprenom_infirmier = new javax.swing.JTextField();
        jtftelephone_infirmier = new javax.swing.JTextField();
        jpfpassword_infirmier = new javax.swing.JPasswordField();
        jbenregistrer_infirmier = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Enregistrement infirmier");

        jLabel1.setText("Nom");

        jLabel2.setText("Prenom");

        jLabel3.setText("Password");

        jLabel4.setText("Telephone");

        jbenregistrer_infirmier.setText("Enregistrer");
```



```

        .addComponent(jtftelephone_infirmier,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jpfpasword_infirmier,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3))))
        .addGap(18, 18, 18)
        .addComponent(jbenregistrer_infirmier)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

pack();
} // </editor-fold>

private void jbenregistrer_infirmierActionPerformed(java.awt.event.ActionEvent evt) {
// enregistrer

try {
// Enregistrer

Statement str = con.createStatement();

setVisible(false);
infirmier r = new infirmier(jtfnom_infirmier.getText(), jtfprenom_infirmier.getText(),
jtftelephone_infirmier.getText(), jpfpasword_infirmier.getText());
Centrale c = new Centrale();
c.inserer(r, "infirmier");
str.close();
Authentication aut = new Authentication();
aut.show();

} catch (SQLException ex) {
Logger.getLogger(Areceptionniste.class.getName()).log(Level.SEVERE, null, ex);
}

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
/* Set the Nimbus look and feel */
//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
 * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

```

```

    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Ainfirmier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Ainfirmier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Ainfirmier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Ainfirmier.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Ainfirmier().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JButton jbenregistrer_infirmier;
private javax.swing.JPasswordField jpfpassword_infirmier;
private javax.swing.JTextField jtfnom_infirmier;
private javax.swing.JTextField jtfprenom_infirmier;
private javax.swing.JTextField jtftelephone_infirmier;
// End of variables declaration
}

```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import metier.Centrale;
import metier.SingletonConnection;
import metier.medecin;

/**
 *
 * @author christian
 */
public class Amedecin extends javax.swing.JFrame {

    Connection con = SingletonConnection.getConnection();

    /**
     * Creates new form Amedecin
     */
    public Amedecin() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jtfnom_medecin = new javax.swing.JTextField();
        jtfprenom_medecin = new javax.swing.JTextField();
        jtftelephone_medecin = new javax.swing.JTextField();
    }
}
```

```

jScrollPane1 = new javax.swing.JScrollPane();
jtspecialite_medecin = new javax.swing.JTextArea();
jpfpassword_medecin = new javax.swing.JPasswordField();
jbenregistrer_medecin = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Enregistrement medecin");

jLabel1.setText("Nom");

jLabel2.setText("Prenom");

jLabel3.setText("Telephone");

jLabel4.setText("Specialite");

jLabel5.setText("Password");

jtspecialite_medecin.setColumns(20);
jtspecialite_medecin.setRows(5);
jScrollPane1.setViewportView(jtspecialite_medecin);

jbenregistrer_medecin.setText("Enregistrer");
jbenregistrer_medecin.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jbenregistrer_medecinActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel1)
            .addComponent(jLabel2)
            .addComponent(jLabel3)
            .addComponent(jLabel4)
            .addComponent(jLabel5))
        .addGap(26, 26, 26)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
            .addComponent(jpfpassword_medecin)
            .addComponent(jtfprenom_medecin)
            .addComponent(jtftelephone_medecin)
            .addComponent(jtfnom_medecin)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 220,
Short.MAX_VALUE))

```

```

        .addContainerGap(20, Short.MAX_VALUE))
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jbenregistrer_medecin, javax.swing.GroupLayout.PREFERRED_SIZE,
82, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(102, 102, 102))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel1)
                .addComponent(jtfnom_medecin, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jtfprenom_medecin, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(jtftelephone_medecin, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel4)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jpfpassword_medecin,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel5))
            .addGap(18, 18, 18)
            .addComponent(jbenregistrer_medecin)
            .addContainerGap(20, Short.MAX_VALUE))
        );

    pack();
} // </editor-fold>

private void jbenregistrer_medecinActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        // Enregistrer

```

```

        Statement str = con.createStatement();

        setVisible(false);
        medecin r = new medecin(jtfnom_medecin.getText(), jtfprenom_medecin.getText(),
jtaspécialite_medecin.getText(),jtftelephone_medecin.getText(), jpfpassword_medecin.getText());
        Centrale c = new Centrale();
        c.inserer(r, "medecin");
        str.close();
        Authentification aut = new Authentification();
        aut.show();

    } catch (SQLException ex) {
        Logger.getLogger(Areceptionniste.class.getName()).log(Level.SEVERE, null, ex);
    }

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Amedecin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Amedecin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Amedecin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Amedecin.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

```

```

E, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Amedecin().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JButton jbenregistrer_medecin;
private javax.swing.JPasswordField jpfpassword_medecin;
private javax.swing.JTextArea jtaspecialite_medecin;
private javax.swing.JTextField jtfnom_medecin;
private javax.swing.JTextField jtfprenom_medecin;
private javax.swing.JTextField jtfttelephone_medecin;
// End of variables declaration
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import metier.Centrale;
import metier.SingletonConnection;
import metier.receptionniste;

/**
 *
 * @author christian
 */

```

```

public class Areceptionniste extends javax.swing.JFrame {

    // SingletonConnection co=new SingletonConnection();
    Connection con = SingletonConnection.getConnection();

    /**
     * Creates new form Areceptionniste
     */
    public Areceptionniste() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jtfnom_receptionniste = new javax.swing.JTextField();
        jtfprenom_receptionniste = new javax.swing.JTextField();
        jfttelephone_receptionniste = new javax.swing.JTextField();
        jpfpassword_receptionniste = new javax.swing.JPasswordField();
        jbenregistrer_receptionniste = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Enregistrement receptionniste");

        jLabel1.setText("Nom");

        jLabel2.setText("Prenom");

        jLabel3.setText("Telephone");

        jLabel4.setText("Password");

        jbenregistrer_receptionniste.setText("Enregistrer");
        jbenregistrer_receptionniste.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jbenregistrer_receptionnisteActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

```

```

getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup())
                    .addContainerGap()

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel1)
                    .addComponent(jLabel2)
                    .addComponent(jLabel3)
                    .addComponent(jLabel4))
                .addGap(32, 32, 32)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jtfnom_receptionniste)
                .addComponent(jtfprenom_receptionniste,
javax.swing.GroupLayout.DEFAULT_SIZE, 150, Short.MAX_VALUE)
                .addComponent(jtftelephone_receptionniste)
                .addComponent(jpfpasword_receptionniste)))
            .addGroup(layout.createSequentialGroup())
                .addGap(115, 115, 115)
                .addComponent(jbenregistrer_receptionniste,
javax.swing.GroupLayout.PREFERRED_SIZE, 82,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(44, Short.MAX_VALUE))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
            .addGap(13, 13, 13)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                .addComponent(jLabel1)
                .addComponent(jtfnom_receptionniste,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jtfprenom_receptionniste,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup())
                    .addComponent(jLabel3)
                    .addGap(18, 18, 18)
                    .addComponent(jLabel4))
                .addGroup(layout.createSequentialGroup())

```

```

        .addComponent(jtftelephone_receptionniste,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jpfpassword_receptionniste,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)
        .addComponent(jbenregistrer_receptionniste)
        .addContainerGap(20, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void jbenregistrer_receptionnisteActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // Enregistrer

        Statement str = con.createStatement();

        setVisible(false);
        receptionniste r = new receptionniste(jtfnom_receptionniste.getText(),
jtfprenom_receptionniste.getText(), jtftelephone_receptionniste.getText(),
jpfpassword_receptionniste.getText());
        Centrale c = new Centrale();
        c.inserer(r, "receptionniste");

        Authentication aut = new Authentication();
        aut.show();

    } catch (SQLException ex) {
        Logger.getLogger(Areceptionniste.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
            }
        }
    }
}

```

```

        break;
    }
}
} catch (ClassNotFoundException ex) {

```

```

java.util.logging.Logger.getLogger(Areceptionniste.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(Areceptionniste.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

```

```

java.util.logging.Logger.getLogger(Areceptionniste.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```

java.util.logging.Logger.getLogger(Areceptionniste.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Areceptionniste().setVisible(true);
    }
});
}

```

```

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JButton jbenregistrar_receptionniste;
private javax.swing.JPasswordField jpfpassword_receptionniste;
private javax.swing.JTextField jtfnom_receptionniste;
private javax.swing.JTextField jtfprenom_receptionniste;
private javax.swing.JTextField jfttelephone_receptionniste;
// End of variables declaration
}

```

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```

package frames;

```

```
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import metier.Centrale;
import metier.SingletonConnection;
import metier.technicien_lab;

/**
 *
 * @author christian
 */
public class Atechnicien_lab extends javax.swing.JFrame {

    Connection con = SingletonConnection.getConnection();

    /**
     * Creates new form Atechnicien_lab
     */
    public Atechnicien_lab() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jtfnom_technicien_lab = new javax.swing.JTextField();
        jtfprenom_technicien_lab = new javax.swing.JTextField();
        jtfttelephone_technicien_lab = new javax.swing.JTextField();
        jpfpassword_technicien_lab = new javax.swing.JPasswordField();
        jbenregistrer_technicien_lab = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Enregistrement technicien de lab");

        jLabel1.setText("Nom");
```

```

jLabel2.setText("Prenom");

jLabel3.setText("Telephone");

jLabel4.setText("Password");

jpfpassword_technicien_lab0.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jpfpassword_technicien_lab0ActionPerformed(evt);
    }
});

jbenregistrar_technicien_lab0.setText("Enregistrar");
jbenregistrar_technicien_lab0.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jbenregistrar_technicien_lab0ActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(32, 32, 32)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel1)
            .addComponent(jLabel2)
            .addComponent(jLabel3)
            .addComponent(jLabel4))
        .addGap(20, 20, 20)
        .addComponent(jbenregistrar_technicien_lab0,
javax.swing.GroupLayout.PREFERRED_SIZE, 89,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jtfnom_technicien_lab0, javax.swing.GroupLayout.DEFAULT_SIZE,
150, Short.MAX_VALUE)
        .addComponent(jtfprenom_technicien_lab0)
        .addComponent(jtftelephone_technicien_lab0)
        .addComponent(jpfpassword_technicien_lab0)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jbenregistrar_technicien_lab0,
javax.swing.GroupLayout.PREFERRED_SIZE, 89,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(22, 22, 22))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(16, 16, 16)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(jLabel1)
        .addComponent(jtfnom_technicien_lab,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(jtfprenom_technicien_lab,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(jtftelephone_technicien_lab,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(jLabel4)
        .addComponent(jpfpassword_technicien_lab,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jbenregistrer_technicien_lab)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );

    pack();
} // </editor-fold>

private void jpfpassword_technicien_labActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jbenregistrer_technicien_labActionPerformed(java.awt.event.ActionEvent evt) {
    // Enregistrer
    try {
        // Enregistrer

        Statement str = con.createStatement();

        setVisible(false);
        technicien_lab r = new technicien_lab(jtfnom_technicien_lab.getText(),
jtfprenom_technicien_lab.getText(), jtftelephone_technicien_lab.getText(),
jpfpassword_technicien_lab.getText());
        Centrale c = new Centrale();
        c.inserer(r, "technicien_lab");
        str.close();
        Authentication aut = new Authentication();
        aut.show();
    }
}

```

```

    } catch (SQLException ex) {
        Logger.getLogger(Areceptionniste.class.getName()).log(Level.SEVERE, null, ex);
    }

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Atechnicien_lab0.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Atechnicien_lab0.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Atechnicien_lab0.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Atechnicien_lab0.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    }
    </editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Atechnicien_lab0().setVisible(true);
        }
    });
}

```

```
}

// Variables declaration - do not modify
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JButton jbenregistrer_technicien_lab;
private javax.swing.JPasswordField jpfpassword_technicien_lab;
private javax.swing.JTextField jtfnom_technicien_lab;
private javax.swing.JTextField jtfprenom_technicien_lab;
private javax.swing.JTextField jtftelephone_technicien_lab;
// End of variables declaration
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import metier.SingletonConnection;

/**
 *
 * @author christian
 */
public class Authentification extends javax.swing.JFrame {

    Connection conn = SingletonConnection.getConnection();

    /**
     * Creates new form accueil
     */
    public Authentification() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.

```

```
*/
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jPasswordField1 = new javax.swing.JPasswordField();
    jComboBox1 = new javax.swing.JComboBox();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Authentication");
    setAlwaysOnTop(true);
    setBackground(java.awt.Color.cyan);
    setForeground(new java.awt.Color(102, 102, 255));

    jLabel1.setText("ID");

    jLabel2.setText("Password");

    jPasswordField1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jPasswordField1ActionPerformed(evt);
        }
    });

    jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"receptionniste", "medecin", "infirmier", "technicien de labo" }));
    jComboBox1.setToolTipText("selectionner ici...");
    jComboBox1.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            jComboBox1MouseClicked(evt);
        }
    });

    jButton1.setText("Connecter");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Enregistrer");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton2ActionPerformed(evt);
        }
    })
}
```

```
});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel2))
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jtfid, javax.swing.GroupLayout.PREFERRED_SIZE, 120,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jComboBox1, 0, 112, Short.MAX_VALUE))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jButton1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jButton2)))
            .addGap(10, 10, 10)
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(jtfid, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jComboBox1))
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addGap(10, 10, 10)
        );
pack();
} // </editor-fold>
```

```
private void jPasswordField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // connecter
    try {

        PreparedStatement psr = conn.prepareStatement("select * from receptionniste where
id_receptionniste=? and password_receptionniste=?");
        PreparedStatement psm = conn.prepareStatement("select * from medecin where id_medecin
= ? and password_medecin=?");
        PreparedStatement psi = conn.prepareStatement("select * from infirmier where id_infirmier
= ? and password_infirmier=?");
        PreparedStatement pst = conn.prepareStatement("select * from technicien_labor where
id_technicien_labor = ? and password_technicien_labor=?");

        String id = jtfid.getText();
        String pass = jPasswordField1.getText();
        Object selectedItem = jComboBox1.getSelectedItem();
        if (selectedItem == "receptionniste") {

            psr.setString(1, id);
            psr.setString(2, pass);
            ResultSet rs = psr.executeQuery();
            if (rs.next()) {
                Reception rc = new Reception();
                rc.show();
                this.setVisible(false);

            } else {
                JOptionPane.showMessageDialog(null, "id et password invalides!!!");
                // System.out.println("id et password invalides!!!");
            }

            psr.close();

        }
        if (selectedItem == "medecin") {
            psm.setString(1, id);
            psm.setString(2, pass);
            ResultSet rs = psm.executeQuery();
            if (rs.next()) {
                Medecin md = new Medecin();
                md.getJtfidmed_medecin().setText(jtfid.getText());
                md.show();
                this.setVisible(false);

            } else {
```

```
JOptionPane.showMessageDialog(null, "id et password invalides!!!");
// System.out.println("id et password invalides!!!");
}

psm.close();

}

if (selectedItem == "infirmier") {

    psi.setString(1, id);
    psi.setString(2, pass);
    ResultSet rs = psi.executeQuery();
    if (rs.next()) {

        Triage tr = new Triage();
        tr.getJtfid_triage().setText(jtfid.getText());
        tr.show();
        this.setVisible(false);
    } else {
        JOptionPane.showMessageDialog(null, "id et password invalides!!!");
        // System.out.println("id et password invalides!!!");
    }

    psi.close();

}

if (selectedItem == "technicien de labo") {

    pst.setString(1, id);
    pst.setString(2, pass);
    ResultSet rs = pst.executeQuery();
    if (rs.next()) {
        Labo lb = new Labo();
        lb.show();
        this.setVisible(false);
    } else {
        JOptionPane.showMessageDialog(null, "id et password invalides!!!");
        // System.out.println("id et password invalides!!!");
    }

    pst.close();

} else if (selectedItem == null) {
    JOptionPane.showMessageDialog(null, "veuillez selectionner votre fonction");
}

//Logger.getLogger(PatientMetierImpl.class.getName()).log(Level.SEVERE, null, ex);
```

```
} catch (Exception ex) {
    Logger.getLogger(SingletonConnection.class.getName()).log(Level.SEVERE, null, ex);
}

}

private void jComboBox1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // Enregistrer

    try {

        // String id = jTextField1.getText();
        //String pass = jPasswordField1.getText();
        Object selectedItem = jComboBox1.getSelectedItem();

        //faut creer des vues pour l'enregistrement du personnel
        if (selectedItem == "receptionniste") {

            Areceptionniste ar = new Areceptionniste();
            setVisible(false);
            ar.show();

        }
        if (selectedItem == "medecin") {
            Amedecin am = new Amedecin();
            setVisible(false);
            am.show();

        }

        if (selectedItem == "infirmier") {

            Ainfirmier ai = new Ainfirmier();
            setVisible(false);
            ai.show();

        }

        if (selectedItem == "technicien de labo") {

            Atechnicien_lab0 at = new Atechnicien_lab0();
            setVisible(false);
            at.show();

        }

    }
}
```

```

        //else if(selectedItem == null) {
        // JOptionPane.showMessageDialog(null, "veuillez selectionner votre fonction");
        // }
    } catch (Exception e) {
    }

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    <!-- editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) "-->
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Authentication.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Authentication.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Authentication.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Authentication.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
    }
    </editor-fold>
    </editor-fold>
    </editor-fold>
    </editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {

```

```
        public void run() {
            new Authentication().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JTextField jTextField1;
// End of variables declaration
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

import java.sql.SQLException;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.table.DefaultTableModel;
import metier.Centrale;
import metier.Examen;
import metier.patient;

/**
 *
 * @author christian
 */
public class Labo extends javax.swing.JFrame {

    /**
     * Creates new form Examens
     */
    public Labo() {
        initComponents();
    }
}
```

```

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jScrollPane1 = new javax.swing.JScrollPane();
    jTable1 = new javax.swing.JTable();
    jTextField1 = new javax.swing.JTextField();
    jLabel1 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    addWindowListener(new java.awt.event.WindowAdapter() {
        public void windowOpened(java.awt.event.WindowEvent evt) {
            formWindowOpened(evt);
        }
    });

    jTable1.setModel(new javax.swing.table.DefaultTableModel(
        new Object [][] {
            {null, null, null},
            {null, null, null},
            {null, null, null},
            {null, null, null}
        },
        new String [] {
            "Id", "Examen", "Resultat"
        }
    ));
    jTable1.setColumnSelectionAllowed(true);
    jTable1.setDragEnabled(true);
    jScrollPane1.setViewportView(jTable1);

    jTable1.getColumnModel().getSelectionModel().setSelectionMode(javax.swing.ListSelectionModel
.MULTIPLE_INTERVAL_SELECTION);
    if (jTable1.getColumnModel().getColumnCount() > 0) {
        jTable1.getColumnModel().getColumn(0).setHeaderValue("Id");
        jTable1.getColumnModel().getColumn(1).setHeaderValue("Examen");
        jTable1.getColumnModel().getColumn(2).setHeaderValue("Resultat");
    }

    jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            jTextField1KeyReleased(evt);
        }
    }

```

```

});

jLabel1.setText("Recherche");

jButton1.setText("Enregistrer");

jButton2.setText("Ordonnance");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 150,
                    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 681,
                    Short.MAX_VALUE))
            .addContainerGap())
        .addGroup(layout.createSequentialGroup()
            .addGap(231, 231, 231)
            .addComponent(jButton1)
            .addGap(10, 10, 10)
            .addComponent(jButton2)
            .addContainerGap())
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 242,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(10, 10, 10)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton1)
                .addComponent(jButton2))
            .addContainerGap())
);

```

```

    pack();
} // </editor-fold>

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    try {
        // TODO add your handling code here:
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        model.setRowCount(0);
        new Centrale().initDataex(jTable1);
    } catch (SQLException ex) {
        Logger.getLogger(Medecin.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jTextField1KeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    Centrale c = null;
    try {
        c = new Centrale();
    } catch (SQLException ex) {
        Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
    }

    if (!jTextField1.getText().trim().equalsIgnoreCase("")) {
        List<Examen> exa = null;
        try {

            exa = c.rechercheex(jTextField1.getText());
        } catch (SQLException ex) {
            Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
        }
        exa.stream().forEach((Examen p) -> {

            DefaultTableModel tableModel = (DefaultTableModel) jTable1.getModel();
            tableModel.setRowCount(0);
            tableModel.addRow(new Object[] {p.getId_examen(), p.getNom_examen()});

        });
    } else {
        try {
            new Centrale().initDataex(jTable1);
        } catch (SQLException ex) {
            Logger.getLogger(Triage.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

/**
 * @param args the command line arguments

```

```

*/
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Labo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Labo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Labo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Labo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Labo().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
private javax.swing.JTextField jTextField1;

```

```
// End of variables declaration
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
import metier.Centrale;
import metier.analyse_medicale;
import metier.consultation;
import metier.patient;

/**
 *
 * @author christian
 */
public class Medecin extends javax.swing.JFrame {

    /**
     * Creates new form Medecin
     */
    public Medecin() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jtfRecherche = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
    }
}
```

```
jLabel2 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
jTextArea1 = new javax.swing.JTextArea();
jButton2 = new javax.swing.JButton();
jtfidmed_medecin = new javax.swing.JTextField();
jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Medecin");
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowOpened(java.awt.event.WindowEvent evt) {
        formWindowOpened(evt);
    }
});

jtfRecherche.setToolTipText("Tapez queleque chose pour la recherche");
jtfRecherche.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jtfRechercheActionPerformed(evt);
    }
});
jtfRecherche.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        jtfRechercheKeyReleased(evt);
    }
});

jLabel1.setText("Recherche");

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {"", null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null},
        {null, null, null, null, null, null}
    },
    new String [] {
        "Id", "Nom", "Prenom", "Date Naissance", "Telephone", "Genre"
    }
));
jTable1.setToolTipText("selectionner une ligne pour enregistrer.....");
jScrollPane1.setViewportView(jTable1);

jLabel2.setText("Observation");

jTextArea1.setColumns(20);
jTextArea1.setRows(5);
jScrollPane2.setViewportView(jTextArea1);

jButton2.setText("Enregistrer");
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jtfidmed_medecin.setEditable(false);

jButton1.setText("Ordonnance");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
                Short.MAX_VALUE)
            .addComponent(jScrollPane2)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                false)
                .addComponent(jtfRecherche, javax.swing.GroupLayout.PREFERRED_SIZE,
                    150, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jLabel1)
                    .addGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jLabel2)
                    .addComponent(jButton2)
                    .addGap(18, 18, 18)
                    .addComponent(jButton1)))
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(235, 235, 235)
                .addComponent(jLabel2))
            .addGroup(layout.createSequentialGroup()
                .addGap(157, 157, 157)
                .addComponent(jButton2)
                .addGap(18, 18, 18)
                .addComponent(jButton1)))
            .addGap(0, 0, Short.MAX_VALUE))
        .addGap()
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```

        .addGroup(layout.createSequentialGroup())
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jtfRecherche, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel1)
                .addComponent(jtfidmed_medecin, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 110,
Short.MAX_VALUE)
            .addGap(24, 24, 24)
            .addComponent(jLabel2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(33, 33, 33)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jButton2)
                .addComponent(jButton1))
            .addGap(18, 18, 18)
    );

    pack();
} // </editor-fold>

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    try {
        // TODO add your handling code here:
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        model.setRowCount(0);
        new Centrale().initData(jTable1);
    } catch (SQLException ex) {
        Logger.getLogger(Medecin.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jtfRechercheKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    Centrale c = null;
    try {
        c = new Centrale();
    } catch (SQLException ex) {
        Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
    }

    if (!jtfRecherche.getText().trim().equalsIgnoreCase("")) {
        List<patient> pat = null;
        try {

```

```

        pat = c.recherchePatient(jtfRecherche.getText());
    } catch (SQLException ex) {
        Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
    }
    pat.stream().forEach((patient p) -> {

        DefaultTableModel tableModel = (DefaultTableModel) jTable1.getModel();
        tableModel.setRowCount(0);
        tableModel.addRow(new Object[]{p.getId_patient(), p.getNom_patient(),
p.getPrenom_patient(), p.getDate_naissance(), p.getTelephone_patient(), p.getGenre()});

    });
} else {
    try {
        new Centrale().initData(jTable1);
    } catch (SQLException ex) {
        Logger.getLogger(Triage.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // enregistrer
    Date date = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("YYYY-MM-dd hh:mm:ss");
    String format = sdf.format(date);
    if (jTable1.getSelectedRow() == -1) {
        JOptionPane.showMessageDialog(rootPane, "veuillez selectionner une ligne");
    } else {
        if (!jTextArea1.getText().isEmpty()) {
            try {
                String valueOf = String.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 0));
                Integer idpat = Integer.valueOf((String) jTable1.getValueAt(jTable1.getSelectedRow(),
0));

                Integer idmed = Integer.valueOf(jtfidmed_medecin.getText());
                consultation cons = new consultation(format, jTextArea1.getText(), idpat, idmed);
                boolean ins = new Centrale().inserer(cons, "consultation");
                jTextArea1.setText("");

                //

                if (ins == true) {
                    //insertion de l'analyse medicale pour id medecin et patient.
                    analyse_medicale anamed = new analyse_medicale(format,idmed,idpat,0);
                    new Centrale().inserer(anamed, "analyse_medicale");
                    JOptionPane.showMessageDialog(this, "enregistrement reussi");
                    //new Labo().setVisible(true);

                } else {
                    JOptionPane.showMessageDialog(this, "enregistrement echouE");
                }
            }
        }
    }
}

```

```

    }
    } catch (SQLException ex) {
        Logger.getLogger(Medecin.class.getName()).log(Level.SEVERE, null, ex);
        System.out.println(ex);
    }
} else {
    JOptionPane.showMessageDialog(this, "le champ de texte observation ne peut etre
vide");
}

}

}

private void jtfRechercheActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Medecin.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Medecin.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Medecin.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

```

```
java.util.logging.Logger.getLogger(Medecin.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Medecin().setVisible(true);
    }
});
}

public JTextField getJtfidmed_medecin() {
    return jtfidmed_medecin;
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextField jtfRecherche;
private javax.swing.JTextField jtfidmed_medecin;
// End of variables declaration
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

//import java.sql.Date;
import java.sql.Connection;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
```

```
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import metier.Centrale;
import metier.SingletonConnection;
import metier.patient;

/**
 *
 * @author christian
 */
public class Reception extends javax.swing.JFrame {

    // private Centrale centrale;
    /**
     * Creates new form Reception
     */
    public Reception() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jtfrecherchepat = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jtfnompat = new javax.swing.JTextField();
        jtfprenompat = new javax.swing.JTextField();
        jftfelpat = new javax.swing.JTextField();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox();
        jLabel7 = new javax.swing.JLabel();
        jtfadresse = new javax.swing.JTextField();
        jXDatePicker3 = new org.jdesktop.swing.JXDatePicker();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTable2 = new javax.swing.JTable();
    }
}
```

```
jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportView(jTable1);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Reception");
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowOpened(java.awt.event.WindowEvent evt) {
        formWindowOpened(evt);
    }
});

jtfrecherchepat.addInputMethodListener(new java.awt.event.InputMethodListener() {
    public void inputMethodTextChanged(java.awt.event.InputMethodEvent evt) {
        jtfrecherchepatInputMethodTextChanged(evt);
    }
    public void caretPositionChanged(java.awt.event.InputMethodEvent evt) {
    }
});
jtfrecherchepat.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jtfrecherchepatActionPerformed(evt);
    }
});
jtfrecherchepat.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        jtfrecherchepatKeyReleased(evt);
    }
});

jLabel1.setText("Recherche");

jLabel2.setText("Nom");

jLabel3.setText("Prenom");

jLabel4.setText("Telephone");

jButton2.setText("Enregistrer");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```

        jButton2ActionPerformed(evt);
    }
});

jButton3.setText("Modifier");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

jLabel5.setText("Sexe");

jLabel6.setText("Date naiss");

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Aucun",
"Homme", "Femme" }));

jLabel7.setText("Adresse");

jTable2.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Id", "Nom", "Prenom", "Date de naissance", "Telephone", "Genre", "Adresse"
    }
));
jTable2.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable2MouseClicked(evt);
    }
});
jScrollPane2.setViewportView(jTable2);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 640, true)
        .addContainerGap()
    )
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton3)
        .addComponent(jButton2)
    )
    .addContainerGap()
);

```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(layout.createSequentialGroup())

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
    .addGroup(layout.createSequentialGroup())
    .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jtfnompat,
javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(56, 56, 56))
    .addGroup(layout.createSequentialGroup())
    .addComponent(jLabel4)
    .addGap(29, 29, 29)
    .addComponent(jtftelpat,
javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE)))
    .addGap(189, 189, 189))
    .addGroup(layout.createSequentialGroup())
    .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 47,
Short.MAX_VALUE)
    .addComponent(jtfprenompat,
javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(245, 245, 245)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jLabel6)
    .addComponent(jLabel7)
    .addComponent(jLabel5)))
    .addGroup(layout.createSequentialGroup())
    .addGap(0, 0, Short.MAX_VALUE)
    .addComponent(jButton2)
    .addGap(34, 34, 34)
    .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(37, 37, 37)))
    .addContainerGap(237, Short.MAX_VALUE))))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jLabel1)
    .addGap(30, 30, 30)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jtfadresse, javax.swing.GroupLayout.PREFERRED_SIZE, 159,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jtfrecherchepat, javax.swing.GroupLayout.PREFERRED_SIZE, 201,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jXDatePicker3, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE))
    .addContainerGap()
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jtfrecherchepat, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel1))
        .addGap(15, 15, 15)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jtfnompat, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jtfadresse, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel7))
    .addComponent(jLabel2))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jtfprenompat, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jXDatePicker3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel6)))
    .addComponent(jLabel3))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jtftelpat, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jLabel4, javax.swing.GroupLayout.Alignment.TRAILING))
        .addGap(18, 18, 18)

```

```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jButton2)
            .addComponent(jButton3))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT_SIZE, 130,
Short.MAX_VALUE)
        .addContainerGap()
    );

    pack();
} // </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // Enregistrer
    int a;
    jtfnompat.getText();
    jtfprenompat.getText();
    if (jtfnompat.getText().trim().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(jtfnompat, "Le champ 'nom' ne doit pas etre vide");
    } else if (jtfprenompat.getText().trim().equalsIgnoreCase("")) {
        JOptionPane.showMessageDialog(jtfnompat, "Le champ 'prenom' ne doit pas etre vide");
    } else {
        try {
            Centrale c = new Centrale();
            Object selectedItem = jComboBox1.getSelectedItem();
            SimpleDateFormat simpleDateFormat = new SimpleDateFormat("dd-MM-yyyy");
            String format = simpleDateFormat.format(jXDatePicker3.getDate());
            if (selectedItem == "Homme") {
                patient p;
                p = new patient(jtfnompat.getText(), jtfprenompat.getText(), format, "H",
jftfelpat.getText(), jtfadresse.getText());
                boolean inserer = c.inserer(p, "patient");
                DefaultTableModel tableModel = (DefaultTableModel) jTable2.getModel();
                // tableModel.setRowCount(0);
                tableModel.addRow(new Object[]{p.getId_patient(), p.getNom_patient(),
p.getPrenom_patient(), p.getTelephone_patient(), p.getDate_naissance(), p.getGenre(),
p.getAdresse_patient()});
                if (inserer = true) {
                    JOptionPane.showMessageDialog(jtfnompat, "Enregistrement reussi");
                } else {
                    JOptionPane.showMessageDialog(jtfnompat, "Enregistrement echoue");
                }
            } else if (selectedItem == "Femme") {
                patient pe;
                pe = new patient(jtfnompat.getText(), jtfprenompat.getText(), format, "F",
jftfelpat.getText(), jtfadresse.getText());
                boolean inserer = c.inserer(pe, "patient");
                DefaultTableModel tableModel = (DefaultTableModel) jTable2.getModel();
                // tableModel.setRowCount(0);
                tableModel.addRow(new Object[]{pe.getId_patient(), pe.getNom_patient(),

```

```

pe.getPrenom_patient(), pe.getTelephone_patient(), pe.getDate_naissance(), pe.getGenre(),
pe.getAdresse_patient());
    if (inserer = true) {
        JOptionPane.showMessageDialog(jtfnompat, "Enregistrement reussi");
    } else {
        JOptionPane.showMessageDialog(jtfnompat, "Enregistrement echouE");
    }
} else {
    JOptionPane.showMessageDialog(jtfnompat, "veuillez selectionner votre genre");
}

} catch (SQLException ex) {
    Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
}

}
}

private void jXDatePicker2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    //modifier

    DefaultTableModel tableModel = (DefaultTableModel) jTable2.getModel();
    if (jTable2.getSelectedRow() == -1) {
        if (tableModel.getRowCount() == 0) {
            JOptionPane.showMessageDialog(jtfnompat, "la table est vide...");

        } else {

            JOptionPane.showMessageDialog(jtfnompat, "veuillez selectionner une ligne...");
        }

    } else {

        Centrale c = null;
        try {
            c = new Centrale();
        } catch (SQLException ex) {
            Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
        }
        Object selectedItem = jComboBox1.getSelectedItem();
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("YY-MM-dd");
        String form = simpleDateFormat.format(jXDatePicker3.getDate());
        if (selectedItem == "Femme") {

            try {

```

```

        String id = String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow(), 0));
        int id_int = Integer.parseInt(id);
        patient p = new patient(jtfnompat.getText(), jtfprenompat.getText(), form, "F",
jftelpat.getText(), jtfadresse.getText());
        System.out.println(p);
        boolean update = new Centrale().update(p, "patient", id_int);
        // System.out.println(update);
        if (update == false) {
            JOptionPane.showMessageDialog(this, "Modification echouee");
        } else {
            JOptionPane.showMessageDialog(this, "Modification reussie");
            jtfnompat.setText("");
            jtfprenompat.setText("");
            jftelpat.setText("");
            jtfadresse.setText("");
        }

    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }

} else if (selectedItem == "Aucun") {
    JOptionPane.showMessageDialog(jtfnompat, "veuillez selectionner votre genre");

} else {
    try {
        String id = String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow(), 0));
        int id_int = Integer.parseInt(id);
        patient p = new patient(jtfnompat.getText(), jtfprenompat.getText(), form, "M",
jftelpat.getText(), jtfadresse.getText());
        System.out.println(p);
        boolean update = new Centrale().update(p, "patient", id_int);
        if (update == false) {
            JOptionPane.showMessageDialog(this, "Modification echouee");
        } else {
            JOptionPane.showMessageDialog(this, "Modification reussie");
            jtfnompat.setText("");
            jtfprenompat.setText("");
            jftelpat.setText("");
            jtfadresse.setText("");
        }
    } catch (SQLException ex) {
        System.out.println(ex.getMessage());
    }
}
}

private void jTable2MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

```

```

jtfnompat.setText(String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow(), 1)));
jtfprenompat.setText(String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow(), 2)));
jftfelpat.setText(String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow(), 4)));
jtfadresse.setText(String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow(), 6)));
jComboBox1.setSelectedItem(String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow(),
5)));
jXDatePicker3.setFormats("dd-MM-yyyy");

jXDatePicker3.setDate(Date.valueOf(String.valueOf(jTable2.getValueAt(jTable2.getSelectedRow()
, 3))));
}

private void jtfrecherchepatActionPerformed(java.awt.event.ActionEvent evt) {

}

private void jtfrecherchepatInputMethodTextChanged(java.awt.event.InputMethodEvent evt) {
// TODO add your handling code here:
}

private void jtfrecherchepatKeyReleased(java.awt.event.KeyEvent evt) {
// TODO add your handling code here:
Centrale c = null;
try {
c = new Centrale();
} catch (SQLException ex) {
Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
}

if (!jtfrecherchepat.getText().trim().equalsIgnoreCase("")) {
List<patient> pat = null;
try {

pat = c.recherchePatient(jtfrecherchepat.getText());
} catch (SQLException ex) {
Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
}
pat.stream().forEach((patient p) -> {

DefaultTableModel tableModel = (DefaultTableModel) jTable2.getModel();
tableModel.setRowCount(0);
tableModel.addRow(new Object[]{p.getId_patient(), p.getNom_patient(),
p.getPrenom_patient(), p.getDate_naissance(), p.getTelephone_patient(), p.getGenre(),
p.getAdresse_patient()}));

});
} else {
try {
new Centrale().initData(jTable2);
} catch (SQLException ex) {

```

```

        Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    try {
        new Centrale().initData(jTable2);
    } catch (SQLException ex) {
        Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Setn the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
java.util.logging.Logger.getLogger(Reception.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Reception.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Reception.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Reception.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

    /* Create and display the form */

```

```
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Reception().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JTable jTable2;
private org.jdesktop.swingx.JXDatePicker jXDatePicker3;
private javax.swing.JTextField jtfadresse;
private javax.swing.JTextField jtfnompat;
private javax.swing.JTextField jtfprenompat;
private javax.swing.JTextField jtfrecherchepat;
private javax.swing.JTextField jftelpat;
// End of variables declaration
}

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package frames;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.JTextField;
```

```
import javax.swing.table.DefaultTableModel;
import metier.Centrale;
import metier.SingletonConnection;
import metier.patient;
import metier.prelevement;

/**
 *
 * @author christian
 */
public class Triage extends javax.swing.JFrame {

    /**
     * Creates new form Triage
     */
    public Triage() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jtfRecherche = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTable1 = new javax.swing.JTable();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jtfPoids = new javax.swing.JTextField();
        jtfTemperature = new javax.swing.JTextField();
        jLabel4 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jButton2 = new javax.swing.JButton();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jtfTensionArterielle = new javax.swing.JTextField();
        jtfPulsation = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        jtfTensionRespiratoire = new javax.swing.JTextField();
        jLabel8 = new javax.swing.JLabel();
        jtfid_triage = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    }
}
```

```
setTitle("Triage");
addWindowListener(new java.awt.event.WindowAdapter() {
    public void windowOpened(java.awt.event.WindowEvent evt) {
        formWindowOpened(evt);
    }
});

jtfRecherche.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jtfRechercheActionPerformed(evt);
    }
});
jtfRecherche.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        jtfRechercheKeyReleased(evt);
    }
});

jLabel1.setText("Recherche");

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null},
        {null, null, null, null, null, null, null}
    },
    new String [] {
        "Id", "Nom", "Prenom", "Date de naissance", "Telephone", "Genre", "Adresse"
    }
));
jTable1.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jTable1MouseClicked(evt);
    }
});
jScrollPane1.setViewportViewView(jTable1);

jLabel2.setText("Poids");

jLabel3.setText("Temperature");

jLabel4.setFont(new java.awt.Font("DejaVu Sans", 1, 12)); // NOI18N
jLabel4.setText("      Observation");

jTextArea1.setColumns(20);
jTextArea1.setRows(5);
jScrollPane2.setViewportViewView(jTextArea1);

jButton2.setText("Enregistrer");
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jLabel5.setText("tension_arterielle");

jLabel6.setText("pulsation");

jLabel7.setText("T_respiratoire");

jLabel8.setText("Infirmier_id");

jtfid_triage.setEditable(false);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addGap(27, 27, 27)
            .addComponent(jtfRecherche, javax.swing.GroupLayout.PREFERRED_SIZE, 193,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 721,
                    Short.MAX_VALUE)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                    .addComponent(jLabel3)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(8, 8, 8)
                        .addComponent(jLabel2)))
                .addGap(28, 28, 28)
                .addComponent(jtfPoids, javax.swing.GroupLayout.DEFAULT_SIZE, 100,
                    Short.MAX_VALUE)
                .addComponent(jtfTemperature)
                .addGap(18, 18, 18)
            )
        )
);
```

```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel5)
            .addComponent(jLabel6))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jtfTensionArterielle,
javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jtfPulsation, javax.swing.GroupLayout.PREFERRED_SIZE,
100, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel7)
            .addComponent(jLabel8))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(jtfTensionRespiratoire,
javax.swing.GroupLayout.DEFAULT_SIZE, 129, Short.MAX_VALUE)
            .addComponent(jtfid_triage)
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(jScrollPane2))))
        .addContainerGap()
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 223,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(239, 239, 239))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(300, 300, 300))))
        );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jtfRecherche, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel1))

```

```

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jtfPoids, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel15)
        .addComponent(jLabel2)
        .addComponent(jtfTensionArterielle, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel17)
        .addComponent(jtfTensionRespiratoire,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jtfTemperature, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel3)
        .addComponent(jLabel6))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jtfPulsation, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jLabel8)
        .addComponent(jtfid_triage, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(12, 12, 12)
        .addComponent(jLabel4, javax.swing.GroupLayout.DEFAULT_SIZE, 25,
Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton2)
        .addContainerGap())
    );

    pack();
} // </editor-fold>

private void jtfRechercheKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    Centrale c = null;
    try {
        c = new Centrale();
    } catch (SQLException ex) {

```

```

        Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
    }

    if (!jtfRecherche.getText().trim().equalsIgnoreCase("")) {
        List<patient> pat = null;
        try {

            pat = c.recherchePatient(jtfRecherche.getText());
        } catch (SQLException ex) {
            Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
        }
        pat.stream().forEach((patient p) -> {

            DefaultTableModel tableModel = (DefaultTableModel) jTable1.getModel();
            tableModel.setRowCount(0);
            tableModel.addRow(new Object[]{p.getId_patient(), p.getNom_patient(),
p.getPrenom_patient(), p.getDate_naissance(), p.getTelephone_patient(), p.getGenre(),
p.getAdresse_patient()});

        });
    } else {
        try {
            new Centrale().initData(jTable1);
        } catch (SQLException ex) {
            Logger.getLogger(Triage.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}

private void formWindowOpened(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    try {
        DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
        model.setRowCount(0);
        new Centrale().initData(jTable1);
    } catch (SQLException ex) {
        Logger.getLogger(Reception.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

    if (jTable1.getSelectedRow() == -1) {
        if (model.getRowCount() == 0) {
            JOptionPane.showMessageDialog(this, "la table est vide...");

        } else {

```

```

        JOptionPane.showMessageDialog(this, "veuillez selectionner une ligne...");
    }

} else {
    String valueOf = String.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 0));
    Integer idpat = Integer.valueOf((String) jTable1.getValueAt(jTable1.getSelectedRow(), 0));
    // System.out.println(idpat);

}
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    //Enregistrer

    try {
        if (jTable1.getSelectedRow() == -1) {
            JOptionPane.showMessageDialog(rootPane, "veuillez selectionner une ligne");

        } else {
            if (!jtfPoids.getText().isEmpty() && !jtfTemperature.getText().isEmpty() &&
!jtfTensionArterielle.getText().isEmpty() && !jtfTensionRespiratoire.getText().isEmpty() &&
!jtfid_triage.getText().isEmpty() && !jTextArea1.getText().isEmpty()) {
                String valueOf = String.valueOf(jTable1.getValueAt(jTable1.getSelectedRow(), 0));
                Integer idpat = Integer.valueOf((String) jTable1.getValueAt(jTable1.getSelectedRow(),
0));

                Date date = new Date();
                SimpleDateFormat sdf = new SimpleDateFormat("YYYY-MM-dd hh:mm:ss");
                String format = sdf.format(date);
                System.out.println(format);

                prelevement prelv = new prelevement(format, jtfPoids.getText(),
jtfTemperature.getText(), jTextArea1.getText(), jtfTensionArterielle.getText(),
jtfPulsation.getText(), jtfTensionRespiratoire.getText(), jtfid_triage.getText(), idpat);
                new Centrale().inserer(prelv, "prelevement");
            } else {
                JOptionPane.showMessageDialog(rootPane, "Aucun champ ne doit etre vide");

            }
        }

    } catch (SQLException ex) {
        Logger.getLogger(Triage.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jtfRechercheActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**

```

```
* @param args the command line arguments
*/
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Triage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Triage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Triage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Triage.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Triage().setVisible(true);

    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
```

```
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTable jTable1;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextField jtfPoids;
private javax.swing.JTextField jtfPulsation;
private javax.swing.JTextField jtfRecherche;
private javax.swing.JTextField jtfTemperature;
private javax.swing.JTextField jtfTensionArterielle;
private javax.swing.JTextField jtfTensionRespiratoire;
private javax.swing.JTextField jtfid_triage;
// End of variables declaration

public JTextField getJtfid_triage() {
    return jtfid_triage;
}
}
```