République du Mali

Un Peuple-Un But-Une Foi

Direction Nationale de L'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale d'Ingénieurs Abderhamane Baba Touré (ENI-ABT)

Departement de Génie Informatiques et Télécommunications(GIT)

Rapport de TP2

Encadré par :

Docteur Goita, MC

Réalisé par:

Aliou Sow

Fatoumata Binta Keita

Hamady Gackou

Sommaire :

Introduction 3
Description de SQLite : 4
Présentation des scripts de création des tables : 5
L'interface principale de l'application : 12
Fenêtre pour gérer les unités fonctionnelles : 12
La connexion à la base de données : 13
Quelques exemples d'interrogation : 14
Conclusion 15

Introduction

Dans cette seconde partie du projet, nous allons reprendre le schéma relationnel de la gestion de l'hôpital, et de l'implémenter dans le système de gestion de base de données SQLite Studio. Nous réaliserons une interface JAVA. En premier lieu, nous utiliserons l'interpréteur SQL pour exécuter les commandes de création de la base ainsi que les tables composant la base de données. Après avoir élaboré l'interface et se connecter à la base de données, nous insérerons des données dans les tables et nous procèderons par la suite, d'exécuter les commandes d'interrogation de la base à partir de l'interface JAVA. Nous terminerons par une conclusion récapitulant le déroulement des travaux et les réalisations atteintes.

Description de SQLite :

SQLite est un système de base de données ou une bibliothèque proposant un moteur de bases de données relationnelles. Il repose sur une écriture en C, un langage de programmation impératif, et sur une accessibilité via le langage SQL (Structured Query Language). **SQLite** présente la particularité d'être directement intégré aux programmes et dans l'application utilisant sa bibliothèque logicielle alors que ses concurrents comme MySQL reproduisent de leur côté le schéma classique client-serveur. Avec **SQLite**, la base de données est intégralement stockée dans un fichier indépendant du logiciel. Créé au début des années 2000 par D. Richard Hipp, **SQLite** propose un accès plus rapide aux données, mais aussi plus structuré et avec davantage de sécurité. À noter que, contrairement à une majorité de systèmes de gestion de base de données (SGBD), SQLite est basé sur un typage dynamique plutôt que sur un typage statique pour le contenu des cellules. On a utilisé le SQLite pour ses avantages, qui sont les suivantes : elle utilise le langage SQL qui, est un standard ; elle est portable ; elle est facile d'usage ; elle permet aux programmes de communiquer directement avec la base de données sans reproduire le schéma classique client-serveur.

Présentation des scripts de création des tables :

Dans cette partie, nous allons présenter le contenu du fichier SQL de création des tables composant la base de données :

```
--

-- Fichier généré par SQLiteStudio v3.3.3 sur sam. août 27
12:46:04 2022

--

-- Encodage texte utilisé : UTF-8


-- Table : boite_tiroir

CREATE TABLE boite_tiroir (_NUMBOITE int (11) NOT NULL PRIMARY
KEY, NUMMALADE int (11) DEFAULT NULL REFERENCES malade (id),
NUMCHARIO int (11) NOT NULL REFERENCES chario (NUMCHARIO));


-- Table : cahier_correspondence

CREATE TABLE cahier_correspondence (_NUMCAHIERSUIVIE int (11)
NOT NULL PRIMARY KEY, NUMSECRETAIRE int (11) NOT NULL,
_DATEEENVOIE date NOT NULL, _LIEU longtext NOT NULL, _DATERETOUR
date NOT NULL);


-- Table : cahier_de_suivi

CREATE TABLE cahier_de_suivi (NUMSUIVI int (11) NOT NULL PRIMARY
KEY, _NUM_SURVEILLANTE int (11) NOT NULL REFERENCES Surveillante
(N°surveillante), NUMCHAMBRE varchar (255) NOT NULL REFERENCES
chambre (id), DATE datetime NOT NULL, NOMPRATICIEN varchar (255)
NOT NULL);


-- Table : chambre

CREATE TABLE chambre (id int NOT NULL PRIMARY KEY, nom varchar
(255) NOT NULL);




-- Table : chario

CREATE TABLE chario (NUMCHARIO int (11) NOT NULL PRIMARY KEY);
```

```
-- Table : dieteticienne

CREATE TABLE dieteticienne (id int NOT NULL PRIMARY KEY UNIQUE,
nom varchar (255) NOT NULL, idunite int NOT NULL REFERENCES
unite_fonctionnelle (id));


-- Table : entree_stock

CREATE TABLE entree_stock (_ID_ENTREE_STOCK int (11) NOT NULL
PRIMARY KEY, _ID_FICHE_J int (11) NOT NULL REFERENCES
fiche_journaliere (_ID_FICHE_J));


-- Table : etat_commande

CREATE TABLE etat_commande (_ID_ETAT int (11) NOT NULL PRIMARY
KEY, _ID_FOURNISSEURS int (11) NOT NULL REFERENCES fournisseur
(id), _NUM_PHARMACIEN int (11) NOT NULL REFERENCES pharmacien
(id), _DATE_JOUR date DEFAULT NULL, _REFERENCE longtext DEFAULT
NULL, _DATE_DERNIERE_COMMANDE date DEFAULT NULL, _STOCK_ACTUEL
int (11) DEFAULT NULL, _CON_JOUR int (11) DEFAULT NULL);


-- Table : examen

CREATE TABLE examen (id INT PRIMARY KEY UNIQUE NOT NULL, nom
VARCHAR (255) NOT NULL, idmedecin INT REFERENCES medecin (id));


-- Table : exemplaire

CREATE TABLE exemplaire (NUMEXEMPLAIRE int (11) NOT NULL PRIMARY
KEY, _NUMFICHE int (11) NOT NULL REFERENCES fiche_manuscrit
(_NUMFICHE));


-- Table : fiche_journaliere

CREATE TABLE fiche_journaliere (_ID_FICHE_J int (11) NOT NULL
PRIMARY KEY, _NUM_PHARMACIEN int (11) NOT NULL REFERENCES
pharmacien (id), _REFERENCE_MED longtext DEFAULT NULL);

-- Table : fiche_manuscrit

CREATE TABLE fiche_manuscrit (_NUMFICHE int (11) NOT NULL
PRIMARY KEY, NUMSERICE int (11) NOT NULL REFERENCES service (id),
_NOMMALADE int (11) NOT NULL, _SEXE varchar (255) NOT NULL,
_ADRESSE varchar (1024) NOT NULL, _NUMUNITE int (11) NOT NULL
```

```
REFERENCES unite_fonctionnelle (id), _NUMCHAMBRE int (11) NOT NULL REFERENCES chambre (id), _INDICATION longtext NOT NULL);


-- Table : fiche_repas

CREATE TABLE fiche_repas (_ID_REPAS int (11) NOT NULL PRIMARY KEY, NUMMALADE int (11) NOT NULL REFERENCES malade (id), _ID_DIETETICIENNE int (11) NOT NULL REFERENCES dieteticienne (id), NUM_CHAMBRE int (11) NOT NULL REFERENCES chambre (id), NUMUNITE int (11) NOT NULL REFERENCES unite_fonctionnelle (id), _ID_INTENDANCE int (11) NOT NULL, _COMPOSITION_REPAS longtext DEFAULT NULL);


-- Table : fournisseur

CREATE TABLE fournisseur (id INT UNIQUE NOT NULL PRIMARY KEY, nom VARCHAR (255) NOT NULL);


-- Table : infirmiere

CREATE TABLE infirmiere (id int NOT NULL PRIMARY KEY, nom varchar (255) NOT NULL, idunite int NOT NULL REFERENCES unite_fonctionnelle (id));


-- Table : laboratoire

CREATE TABLE laboratoire

(

   id int not null primary key,

   nom varchar(255) not null

);




-- Table : lit

CREATE TABLE lit (N_lil Int NOT NULL, N_chambre Int NOT NULL REFERENCES chambre (id), CONSTRAINT lit_PK PRIMARY KEY (N_lil), CONSTRAINT lit_Chambre_FK FOREIGN KEY (N_chambre) REFERENCES Chambre (id));
```

```
-- Table : livre_de_destockage

CREATE TABLE livre_de_destockage (_IDLIVRE int (11) NOT NULL
PRIMARY KEY, REFERENCEMENT longtext DEFAULT NULL, _DATE date
DEFAULT NULL);


-- Table : malade

CREATE TABLE malade (id int NOT NULL PRIMARY KEY, nom varchar
(255) NOT NULL, idservice int NOT NULL REFERENCES service (id),
idunite int NOT NULL REFERENCES unite_fonctionnelle (id),
idchambre int NOT NULL REFERENCES chambre (idchambre));


-- Table : medecin

CREATE TABLE medecin (id INT, nom VARCHAR (255), idservice INT);


-- Table : Medicament

CREATE TABLE Medicament(

Reference_medica Varchar (50)  ,

doses_prescrit varchar (50),

date_prescription date,

N_post    Int    ,CONSTRAINT    Medicament_PK    PRIMARY    KEY
(Reference_medica)


);


-- Table : ordonnance

CREATE TABLE ordonnance (_NUM_ORDONNACE int (11) NOT NULL
PRIMARY KEY);


-- Table : pharmacie

CREATE TABLE pharmacie (id int NOT NULL PRIMARY KEY UNIQUE, nom
varchar (255) NOT NULL);
```

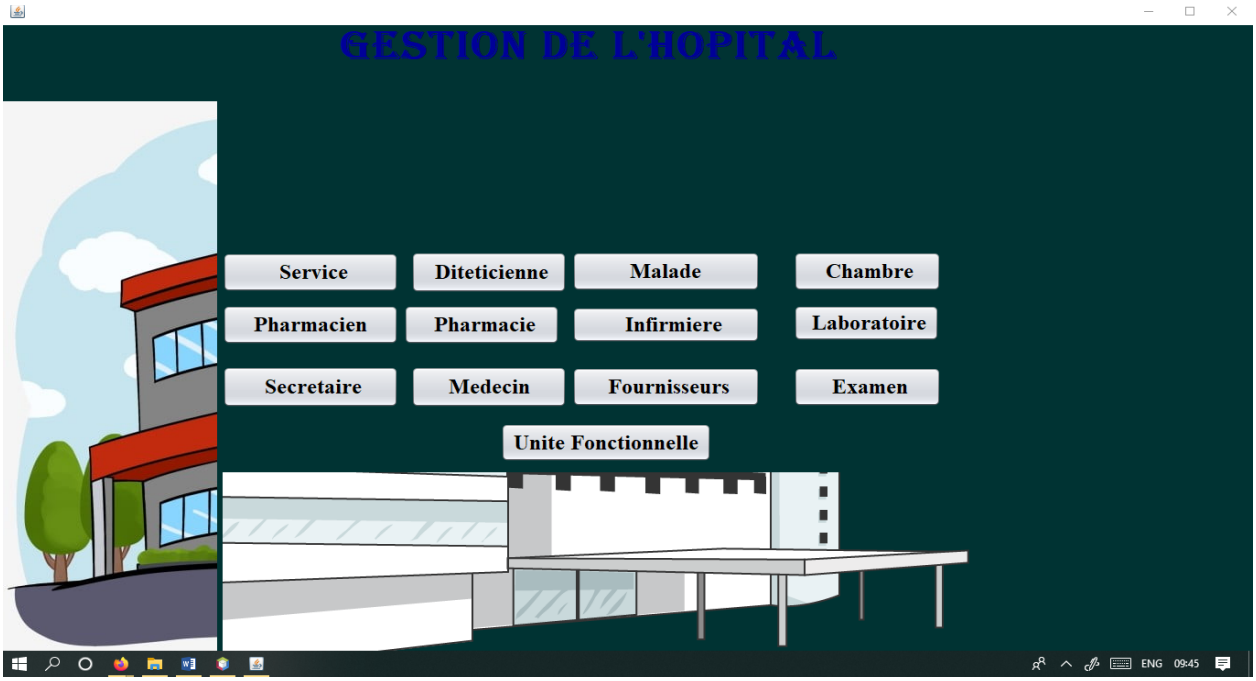
```
-- Table : pharmacien

CREATE TABLE pharmacien (id int NOT NULL PRIMARY KEY, nom varchar
(255) NOT NULL, idpharmacie int NOT NULL REFERENCES pharmacie
(id));


-- Table : prelevermennt

CREATE TABLE prelevermennt (_NUM_PRELEVEMENT int (11) NOT NULL
PRIMARY KEY, NUMTRANSMISSION int (11) NOT NULL REFERENCES
transmission_prelevement (NUMTRANSMISSION), _NUM_SURVEILLANTE
int (11) NOT NULL REFERENCES Surveillante (N°surveillante),
_NUM_ORDONNACE int (11) NOT NULL REFERENCES ordonnance
(_NUM_ORDONNACE), _NUM_INFIRMIERE int (11) NOT NULL REFERENCES
infirmiere (id));


-- Table : prescription

CREATE TABLE prescription (_NUMPRESCRIPTION int (11) NOT NULL
PRIMARY KEY, _NUM_TABLEAU int (11) NOT NULL REFERENCES
Tableau_de_bord (N°tableau), NUMMEDECIN int (11) NOT NULL
REFERENCES medecin (id), NUMCHARIO int (11) NOT NULL REFERENCES
chario (NUMCHARIO), _NUM_TRANSPRES int (11) NOT NULL REFERENCES
transmission_prescrition (_NUM_TRANSPRES), _TYPE longtext NOT
NULL);


-- Table : responsable

CREATE TABLE responsable (NUMTTRAITEMENT int (11) NOT NULL
PRIMARY KEY, NUMSERICE int (11) NOT NULL REFERENCES service
(id));


-- Table : secretaire

CREATE TABLE secretaire (id INT UNIQUE PRIMARY KEY, nom VARCHAR
(255) NOT NULL, prenom VARCHAR (255) NOT NULL);




-- Table : service

CREATE TABLE service

(
```

```
   id int not null primary key,
   nom varchar(255) not null ,
   specialite varchar(255) not null
);

-- Table : Service_intendance
CREATE TABLE Service_intendance(
        Code_dieteticienne   Varchar (10) NOT NULL ,
        N_unite_fonctionnelle Int NOT NULL,
        CONSTRAINT    Service_intendance_PK    PRIMARY    KEY
(Code_dieteticienne),
        CONSTRAINT    Service_intendance_Unite_Fonctionnelle_FK
FOREIGN      KEY      (N_unite_fonctionnelle)      REFERENCES
unite_fonctionnelle(id)
);

-- Table : Surveillante
CREATE TABLE Surveillante(
        N°surveillante  Int NOT NULL ,
          Nom           varchar (50) not null,
          Prenom             varchar (50) not null,
          adresse       varchar (50),
          age           int (5),
          sexe               varchar (5),
          N_unite_fonctionnelle int (10)
     ,CONSTRAINT surveillante__PK PRIMARY KEY (N°surveillante)

     ,CONSTRAINT  surveillante_unité_fonctionnelle_FK  FOREIGN
KEY (N_unite_fonctionnelle) REFERENCES Unite_Fonctionnelle(id)
);
-- Table : Tableau_de_bord
CREATE TABLE Tableau_de_bord(
```

```
        N°tableau              Int NOT NULL ,
          N°medecin                int not null
     ,CONSTRAINT Tableau_de_bord__PK PRIMARY KEY (N°tableau)


     ,CONSTRAINT medecin_FK FOREIGN KEY (N°medecin) REFERENCES
Medecin(id)
);


-- Table : transmettre
CREATE TABLE transmettre (NUMPHARMA int (11) NOT NULL PRIMARY
KEY, NUMSECRETAIRE int (11) NOT NULL REFERENCES secretaire
(id));


-- Table : transmission_prelevement
CREATE TABLE transmission_prelevement (NUMTRANSMISSION int (11)
NOT NULL PRIMARY KEY, NUMSECRETAIRE int (11) NOT NULL REFERENCES
secretaire (id), NUMLABORATOIRE int (11) NOT NULL REFERENCES
laboratoire (id));


-- Table : transmission_prescrition
CREATE TABLE transmission_prescrition (_NUM_TRANSPRES int (11)
NOT NULL PRIMARY KEY, NUMSECRETAIRE int (11) NOT NULL REFERENCES
secretaire (id), NUMPHARMA int (11) NOT NULL REFERENCES
pharmacie (id));


-- Table : unite_fonctionnelle
CREATE TABLE unite_fonctionnelle (id int NOT NULL, nom varchar
(255) NOT NULL, idservice int NOT NULL, "" REFERENCES service
(id), PRIMARY KEY (id));
```

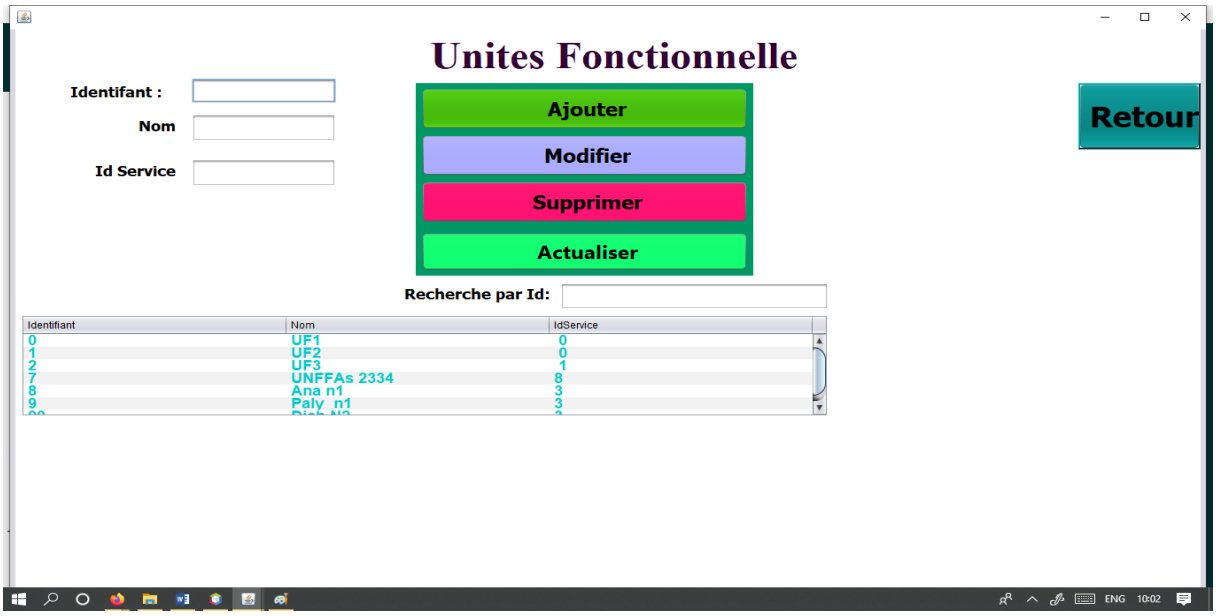
L'interface principale de l'application :

Cette interface représente l'accueil de notre application de gestion de la base de données hôpital. A partir de cette interface, on peut cliquer sur le bouton de création de gestionnaires d'une table en vue de la gérer. Elle est représentée ci-dessous.



Fenêtre pour gérer les unités fonctionnelles :

A partir de cette fenêtre, nous pouvons ajouter, modifier, supprimer, ou rechercher une unité fonctionnelle de notre base de données. Il existe une fenêtre pareille pour toutes les autres tables manipulées par l'application. Elle est représentée ci-dessous.



La connexion à la base de données :

Pour se connecter à notre base de données, nous avons utilisé le jar sqlite-jdbc-3.7.2 et par la suite, on a importé les packages nécessaires à la connexion. Dans chaque fichier séparé communiquant avec la base de données, on a procédé par une connexion à la base de données lors d'une requête. Pour cela, nous allons présenter le script de base pour la connexion à notre base de données. Ce script est représenté ci-dessous :

```
//La connexion est faite après avoir importé les packages nécessaires à la connexion à la
//base de données au début du fichier de la classe correspondante. Ces packages sont les
//suivantes :

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.DriverManager;

import java.sql.SQLException;

// Connexion à la base dans un bloc try , catch :
try {
            Class.forName("org.sqlite.JDBC");
            con = DriverManager.getConnection("jdbc:sqlite:hopital.db");
            java.sql.Statement st = con.createStatement();
/*
             //  Cette partie est dédié aux requêtes



  */
          con.close();
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace() ;
        }

    }
```

Quelques exemples d'interrogation :

-La liste des services : La liste des services est affichée dans le jTable ci-dessous

Identifiant	Nom	Specialite
0	S1	Doctomologie
2	S2	Dicta1
1	S3	diabetie
4	S7	diarrhee
5	S5	palydisme
6	S6	Diagnostic
7	S7	Diagnostic
11	Somadic	Diagnostic Rpp

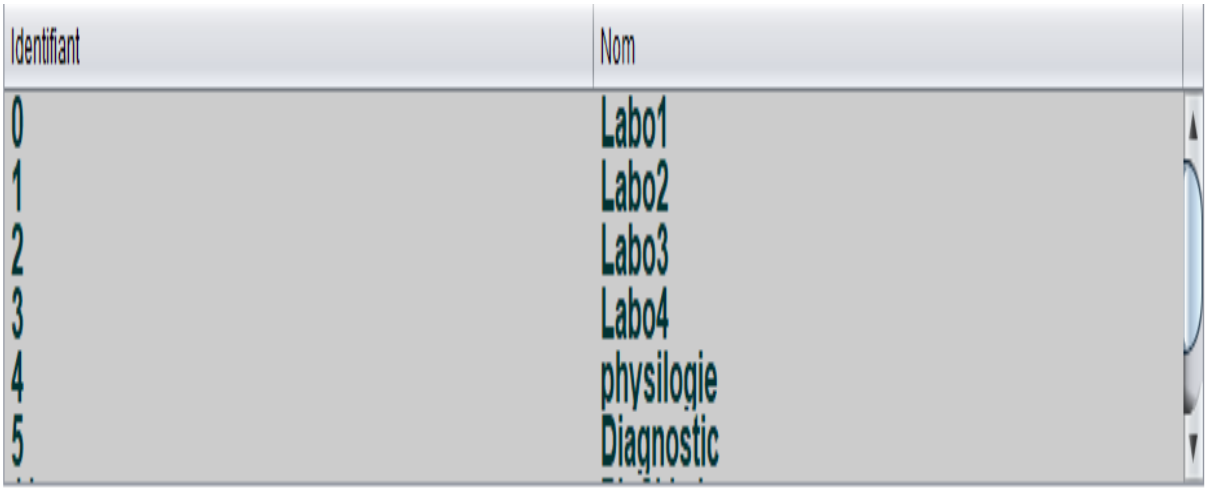
-La liste des unités fonctionnelles : Les unités fonctionnelles de l'hôpital sont affichées dans le jTable ci-dessous

Identifiant	Nom	IdService
0	UF1	0
1	UF2	0
2	UF3	1
7	UNFFAs 2334	8
8	Ana n1	3
9	Paly n1	3

-Les pharmaciens : Les pharmaciens sont affichés dans le jTable ci-dessous

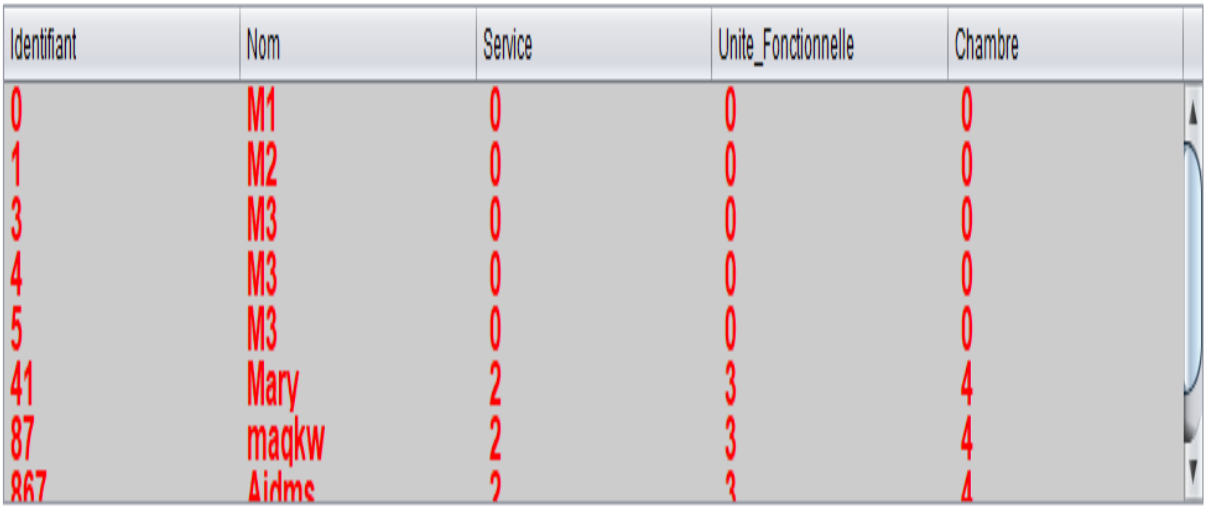
Identifiant	Nom	Id Pharmacie
1	Hamady Gackou	1
2	Fatoumata Binta Keita	1
3	Aliou Sow	1
41	Bigette	BBas
44	Bigettett	q BBas
22	Amadou	2
21	Mamma	2

-Les laboratoires : Les laboratoires de l'hôpital sont affichés dans le jTable ci-dessous



Identifiant	Nom
0	Labo1
1	Labo2
2	Labo3
3	Labo4
4	physilogie
5	Diagnostic

-Les Malades : Les Malades de l'hôpital sont affichés dans le jTable suivant



Identifiant	Nom	Service	Unite_Fonctionnelle	Chambre
0	M1	0	0	0
1	M2	0	0	0
3	M3	0	0	0
4	M3	0	0	0
5	M3	0	0	0
41	Mary	2	3	4
87	maqkw	2	3	4
867	Aidms	2	3	4

Conclusion

Cette deuxième partie du projet nous a permis de comprendre beaucoup de choses. En gros, on a appris : comment concevoir une application, de bas niveau, pour gérer un système d'informations ; comment exploiter les documentations dans un projet informatique ; comment se repérer dans un travail d'équipe. …