

**DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE UNIVERSITÉ DES SCIENCES ET DE LA
TECHNOLOGIE HOUARI BOUMEDIEN**



FACULTÉ D'INFORMATIQUE

Réalisé par:

- HALLI Wiam Maria
- Mankour Youssra
- Achouri Hadjer
- Bentahar Sofian

TABLE OF CONTENTS

1.1 Introduction:	3
1.2 Buts	3
1.3 Objective	3
1.4 Technologies utilisées.....	3
1.5 Les classes de java utilisées.....	5
1.6 Conception de L'application.....	12
1.7 Conclusion.....	12



Gestion clinique medical

Le Rapport

1.1 Introduction:

Découvrez le système de gestion hospitalière, une solution complète pour la gestion efficace des hôpitaux. Enregistrant les détails des patients, assurant une facturation informatisée et facilitant la recherche de disponibilité des médecins et des informations sur les patients, ce système offre une approche intégrée et sécurisée. Accessible via des identifiants uniques, il garantit une gestion conviviale et efficace, améliorant ainsi la qualité des soins et la prise de décisions dans les hôpitaux multiservices.

1.2 Buts:

1. Interface conviviale pour l'utilisateur
2. Simplicité et rapidité
3. Coût réduit et efficacité
4. Gestion efficace des informations des patients
5. Processus de diagnostic optimisé

1.3 Objectifs:

- 1) Définir le rôle de l'hôpital.
- 2) Enregistrer les informations des patients admis.
- 3) Enregistrer les diagnostics des patients.
- 4) Gérer les informations sur les maladies et les médicaments associés.

1.4 Technologies utilisées:

1-Java

Le projet a été codé en Java, en exploitant les fonctionnalités étendues du langage pour garantir la flexibilité et la robustesse de l'application. Nous avons utilisé les bibliothèques Java appropriées pour des fonctionnalités spécifiques. En outre, nous avons adopté des pratiques de programmation orientée objet pour organiser efficacement le code et faciliter sa maintenance.

2-Java Database Connectivity









Qu'est-ce que JDBC ?

JDBC est une API Java pour exécuter des instructions SQL. Bien que le nom JDBC soit une marque déposée et non un acronyme, il est souvent associé à Java Database Connectivity. Composé d'un ensemble de classes et d'interfaces écrites en langage Java, JDBC offre une API standard pour les développeurs d'outils et de bases de données, permettant de créer des applications de bases de données en utilisant une API purement Java. On l'utilise ici pour connecter le projet avec la base de donnée MySQL.

3- DataBase:

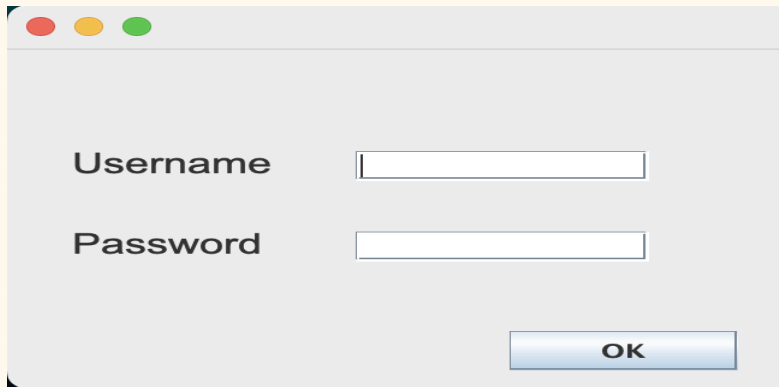
Pour la base de données, nous avons choisi d'utiliser le système de gestion de bases de données relationnelles MySQL. MySQL a été sélectionné en raison de sa robustesse, de sa fiabilité et de sa large adoption dans l'industrie. En tant que solution open-source, MySQL offre une excellente performance et une grande souplesse, ce qui en fait un choix idéal pour notre projet. De plus, sa compatibilité avec les applications Java via le pilote JDBC MySQL a simplifié l'intégration de la base de données dans notre système, facilitant ainsi le développement et la maintenance de l'application. En utilisant MySQL, nous avons pu garantir une gestion efficace des données hospitalières tout en assurant la sécurité et l'intégrité des informations stockées.

-Les tables de la database:

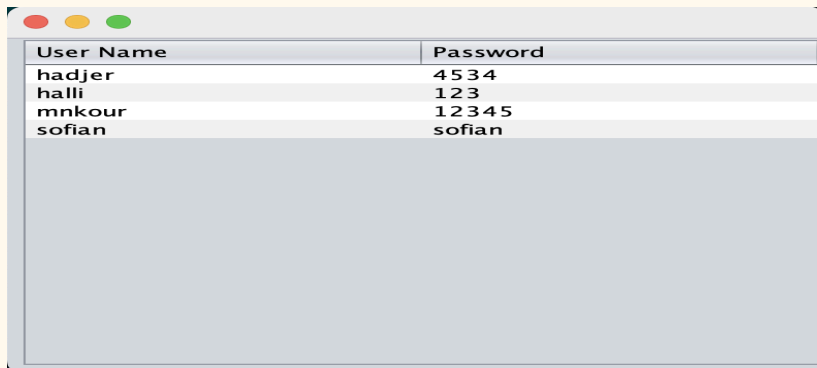
admitpatient_room PatientID  int Disease varchar(50) AdmitDate date RoomNo int NN DoctorID int NN AP_Remarks varchar(50)	bill_room DischargeID int BillingDate date RoomCharges int ServiceCharges int PaymentMode varchar(20) PaymentModeDetails varchar(100) TotalCharges int NoOfDays int TotalRoomCharges int BillNo  varchar(15)	dischargepatient_room AdmitID int DischargeDate date DP_Remarks varchar(50)	doctor DoctorID  int DoctorName varchar(20) FatherName varchar(20) Email varchar(50) ContactNo varchar(11) Qualifications varchar(50) Gender char(1) BloodGroup varchar(5) DateOfJoining date Address varchar(100)
patientregistration PatientID  int PatientName varchar(20) FatherName varchar(20) Email varchar(30) ContactNo varchar(11) Age int Remarks varchar(100) Gen char(1) BG varchar(3) Address varchar(100)	registration name varchar(20) user_name  varchar(20) password varchar(20) email_id varchar(30) contact_no int	room RoomNo  int RoomType varchar(10) RoomCharges int RoomStatus varchar(10)	services ServiceName varchar(20) ServiceDate date PatientID int ServiceCharges int ServiceID  int
users user_name  varchar(20) password varchar(20)			

1.5 Les classes de java utilisé:

- **Login.java:** Cette classe comme son nom le montre est pour se connecter à l'application quand on l'exécute on aura cet écran qui nous demande d'entrer un username est un password:



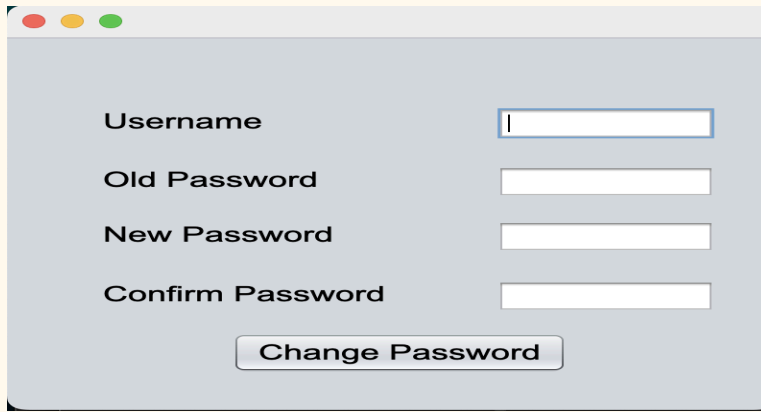
- **LoginDetails.java:** Cette class nous fournis avec les nom des utilisateurs est leur mot de pass quand on l'exécute on aura cet écran:



User Name	Password
hadjer	4534
halli	123
mnkour	12345
sofian	sofian

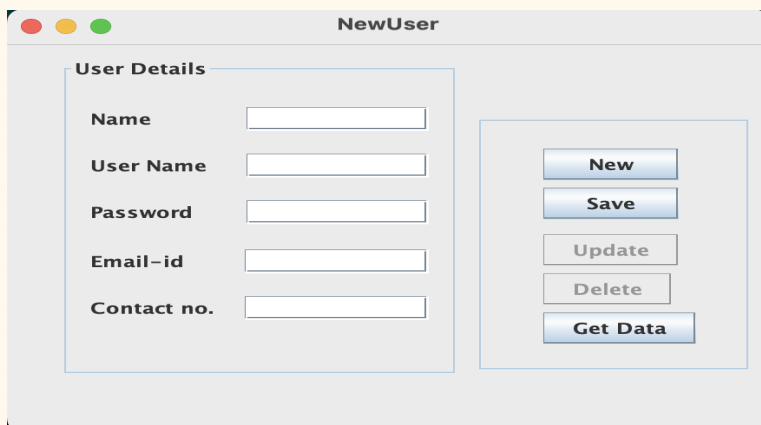
- **Connect.java:** Cette classe permet de connecter java avec le JDBC pour que la base de données MySQL fonctionne.

- **ChangePassword.java:** Cette classe permet les utilisateurs de changer leur mot de passe elle affiche l'écran ci dessous



A screenshot of a Java Swing window titled "Change Password". The window has a light gray background and standard macOS window controls (red, yellow, green buttons) in the top-left corner. It contains four text input fields arranged vertically, each preceded by a label: "Username", "Old Password", "New Password", and "Confirm Password". Below the input fields is a single button labeled "Change Password".

- **NewUser.java:** Cette classe nous permet de cree un nouveau utilisateur elle affiche cet écran:



A screenshot of a Java Swing window titled "NewUser". The window has a light gray background and standard macOS window controls in the top-left corner. It features a "User Details" section on the left, which contains five text input fields labeled "Name", "User Name", "Password", "Email-id", and "Contact no.". To the right of this section is a vertical stack of five buttons: "New", "Save", "Update", "Delete", and "Get Data".

- **Entry.java:** Cette class nous permet d'introduire un nouveau docteur et écrire ses informations elle affiche cet écran:

- **DocRec.java:** Cette class affiche les docteurs déjà introduits et enregistrés elle nous donne cet écran :

Doctor ID	Doctor Name	Father Name	Address	Contact No	Email ID	Qualifications	Gender	Blood Group	Joining Date
1	halli kamel	mohamed	bab ezzoua...	0772392343	kamelhalli@...	doctor	M	O-	2010-03-20

- **UserRecord.java:** Cette classe donne les informations des utilisateurs elle affiche cet écran:

Name	User Name	password	Contact No	Email ID
hadjer	hadjer	4534	213	hadjer@gmail....
halli	halli	123	641162171	xyz
mnkour	mnkour	12345	2147483647	mnkour@gmail...
sofian	sofian	sofian	727475457	sofian@gmail.c...

- **Registration.java:**Cette classe permet de introduire les patients et leur informations elle nous donne cet écran:

Patient Detail's

Patient ID:

Name:

Father's Name:

Address:

Contact No.:

Email-id:

Age:

Gender: ▼

Blood Group: ▼

Information:

Buttons: Save, New, Update, Delete, GetData

- **PatientRec.java:**Cette classe nous donne les patients deja sauvgardés elle affiche cet écran:

Patient ID	Patient Name	Father Name	Address	Contact No	Email ID	Age	Gender	Blood Group	Remarks
11111	halli maria	mohamed	Alger	07727277...	hallimaria@...	19	F	C++	none

- **Room.java:**Cette classe nous permet d'introduire une nouvelle chambre et donner son type et le prix de la chambre et aussi les chambres disponibles et les chambres réservées elle affiche ceci:

Room Info

Room No.:

Room Type: ▼

Room Charges (Per day):

Buttons: New, Save, Update, Delete, Get Data

Room No.	Room Type	Room Charges	Room Status
1	General	5000	Vacant
2	General	2500	Vacant
3	Deluxe	10000	Booked

- **Services.java:** Cette class nous permet d'introduire des nouveau services leur date de service et leur prix et les associer avec les patients disponible elle affiche ceci:

Services Info

Service Name

Service Date (YYYY-MM-DD)

Patient ID

Patient Name

Service Charges

New Save Delete Update Get Data

Patient ID	Patient Name
11111	halli maria

- **ServicesRec.java:** Cette classe affiche les services qui ont ete deja introduit comme on le voit ci dessous:

Service ID	Service Name	Service Date	Patient ID	Patient Name	Service Charges
16	General Consultation	2021-03-01	11111	halli maria	5000

- **Bill_Room.java:** Cette class est pour les details de paiement pour le service et la chambre elle affiche cet fenetre:

Patient Info

Patient ID >

Patient Name

Gender

Blood Group

Disease

Admit Date (YYYY-MM-DD)

Room No.

Doctor ID

Doctor Name

Discharge Date (YYYY-MM-DD)

New Save Delete Update GetData

Patient ID	Patient ...	Service...
11111	halli m...	5000

Payment Details

Room Charges No. Of Days Total Room Charges

Service Charges Billing Date (YYYY-MM-DD)

Payment Mode

Payment Mode Details

Total Charges Bill No:

- **Bill_RoomRec.java:** Cette classe affiche les paiement déjà submit.
- **Patient_Admit_Room.java:** Cette class demande d'entrer les information du patient apres etre admis par exemple qui est son docteur et quelle maladie il a elle affiche ceci:

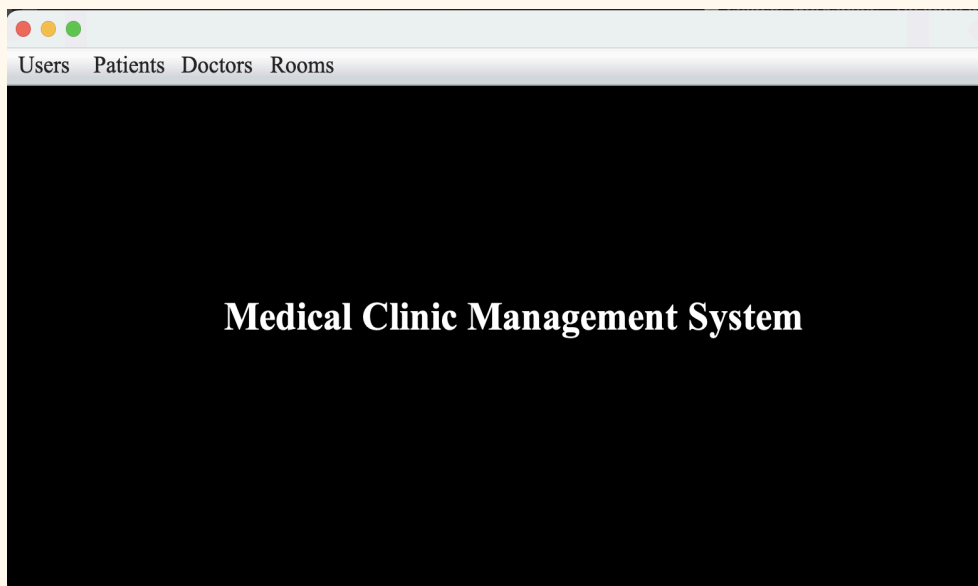
- **Patient_Admit_RoomRec.java:** Cette classe affiches les donnes des patients deja admits elle nous donne ceci:

Patient ID	Patient Name	Gender	Blood Group	Disease	Admit Date	Room No	Doctor ID	Doctor Name	Remarks
11111	halli maria	F	C++	Typhoid fever	2021-03-02	1	1	halli kamel	High temp ...

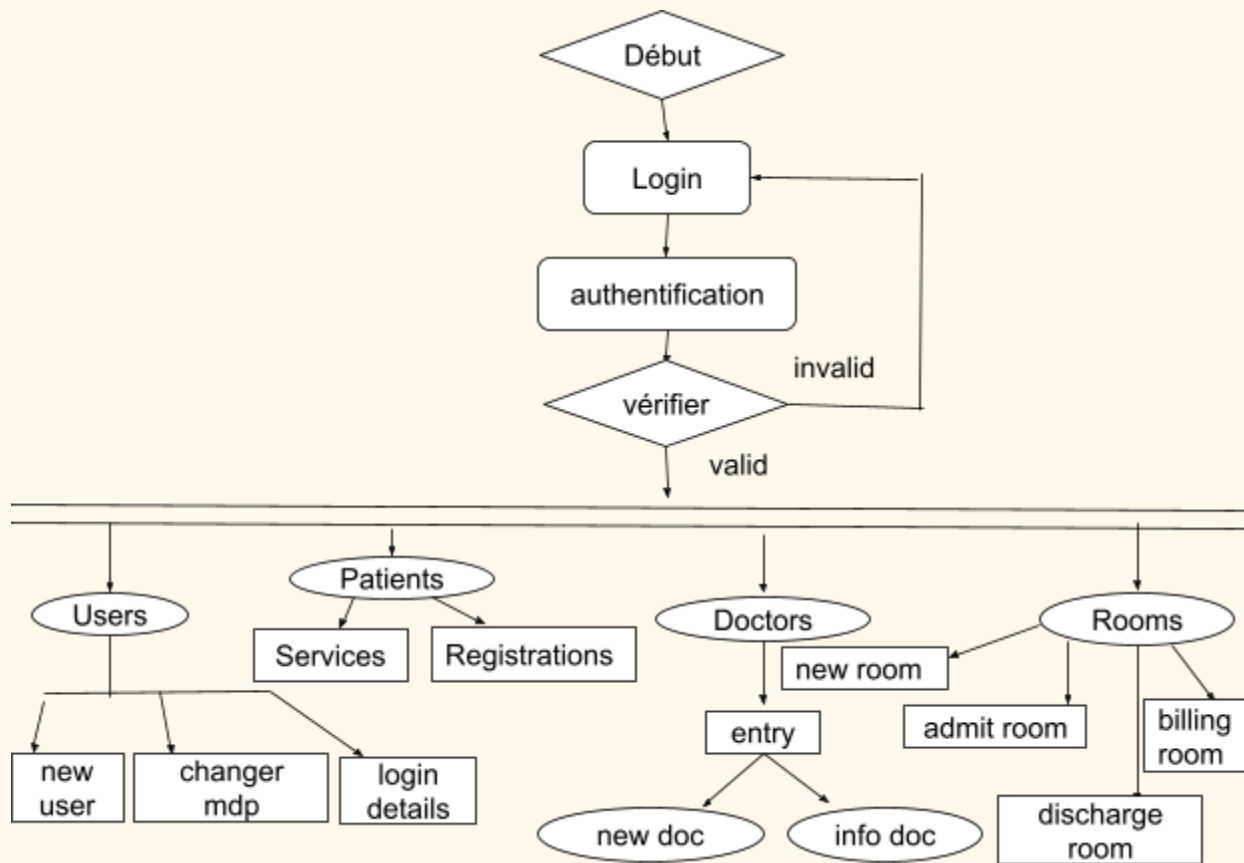
- **Patient_Discharge_Room.java:** Cette classe est pour quand le patient sort de l'hôpital on doit mentionner la date de sa sortie elle affiche cet écran:

The screenshot shows a Java Swing window titled "Patient Discharge Info". It contains a form with the following fields: Patient ID (with a small ">" button), Patient Name, Gender, Blood Group, Disease, Admit Date (with a "(YYYY-MM-DD)" label), Room No., Doctor ID, Doctor Name, Discharge Date (with a "(YYYY-MM-DD)" label), and Remarks (a large text area). To the right of the form is a vertical panel with five buttons: "New", "Save", "Delete", "Update", and "Get Data".

- **Patient_Discharge_RoomRec.java:** Cette classe affiche les détails de la sortie des patients.
- **Main.java:** La classe main affiche le menu principale ou on peut accéder à tous les services L'application offre elle affiche ce menu:



1.6 Conception de L'application:



1.7 Conclusion:

Le Système de Gestion d'une clinique médicale est une solution puissante qui peut aider les hôpitaux à améliorer leurs opérations, à renforcer la satisfaction des patients et à augmenter leurs revenus. Son interface conviviale et ses flux de travail intuitifs facilitent l'accès et la gestion des informations des patients pour les professionnels de la santé, permettant de suivre les ressources et d'optimiser leur utilisation. Grâce à ses capacités de reporting et d'analyse robustes, les hôpitaux peuvent obtenir des insights sur leur performance et prendre des décisions basées sur les données pour stimuler l'amélioration continue.