Hospital Management System

MADHUKAR TEMBA
19BAI1161
CSE2004 PROJECT
REVIEW 3

Aim

The aim is to design a hospital management system for a hospital which will keep track of the patient details and their diagnosis, details can be searched by the doctor and updated or added.

The system uses Java for the front end UI and MySQL is used for the database.

Why DBMS for this work

As there are a lot of patients in a hospital keeping track of each patients details and their diagnosis can be a hectic task if normal data structures are used.

By storing the patient’s data in tables we can make this task a lot easier.

DBMS will allow us to add, remove, update, search any patients details with ease so that is the reason why DBMS is being used.

Limitations of current system

The current system has shifted from computers back to papers due to recent malware attacks in many hospitals. We can clearly see that this is a major problem as computers can do the paper work much faster.

Database to be used

MySQL database is being used to store the patient data currently.

It is being accessed by the Java program using SQL commands.

Hardware Requirements

- This code can be run on any 32bit or 64bit device running Java and MySQL.

Software Requirements

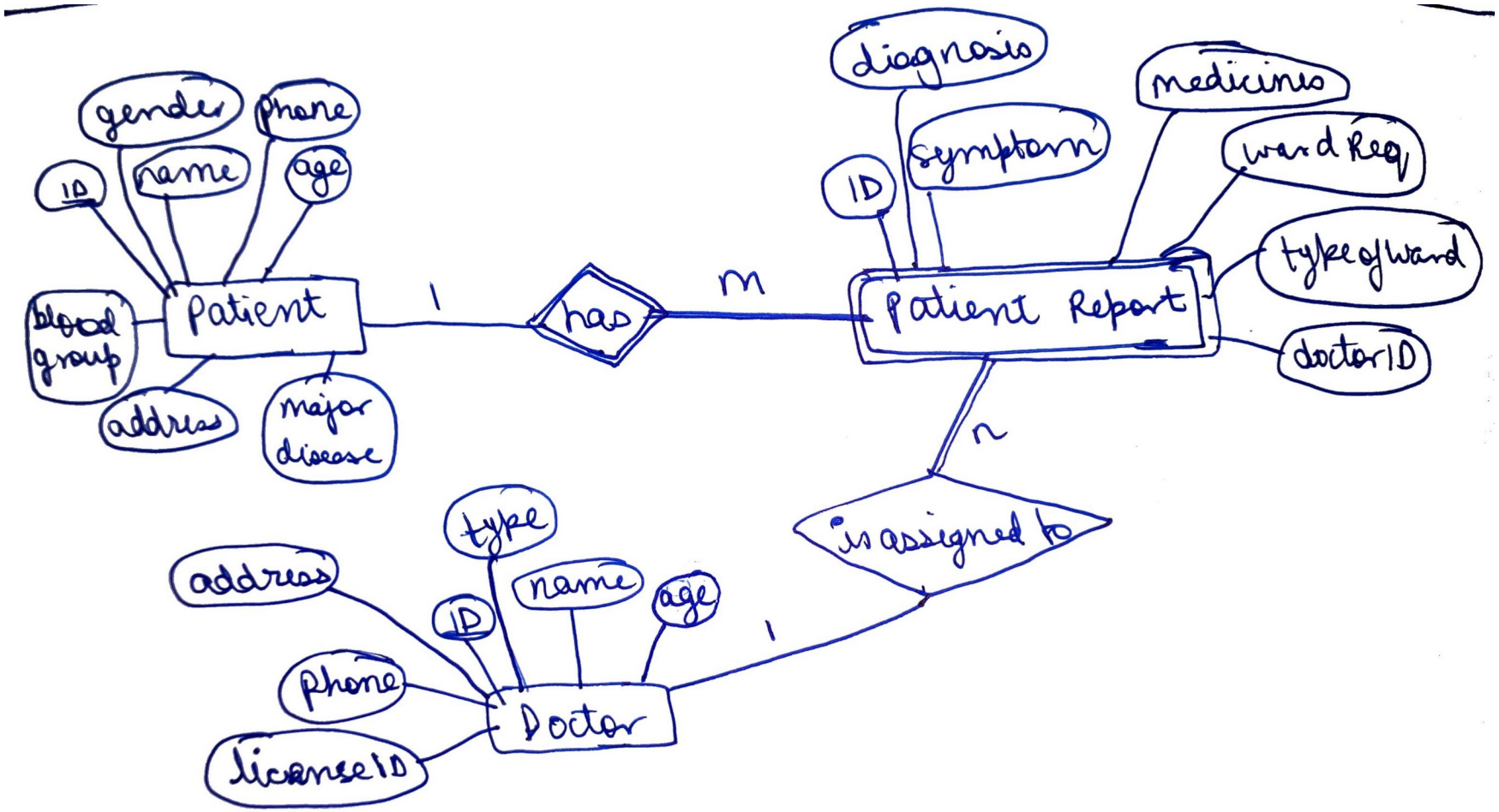
- This project’s minimum requirements are:

1.NetBeans 8.2 or above.

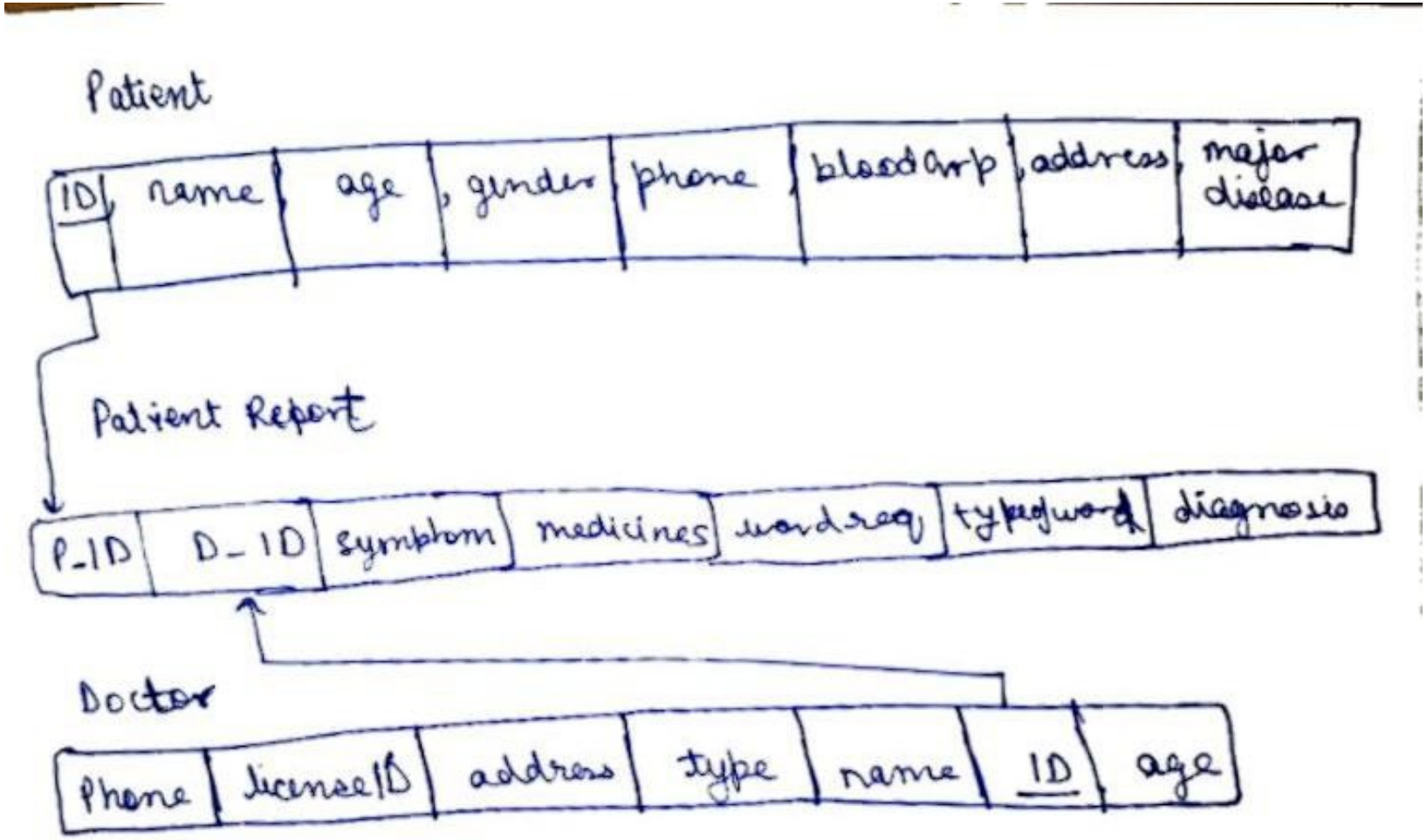
2.MySQL 5.5 or similar SQL database.

3.Java JDBC.

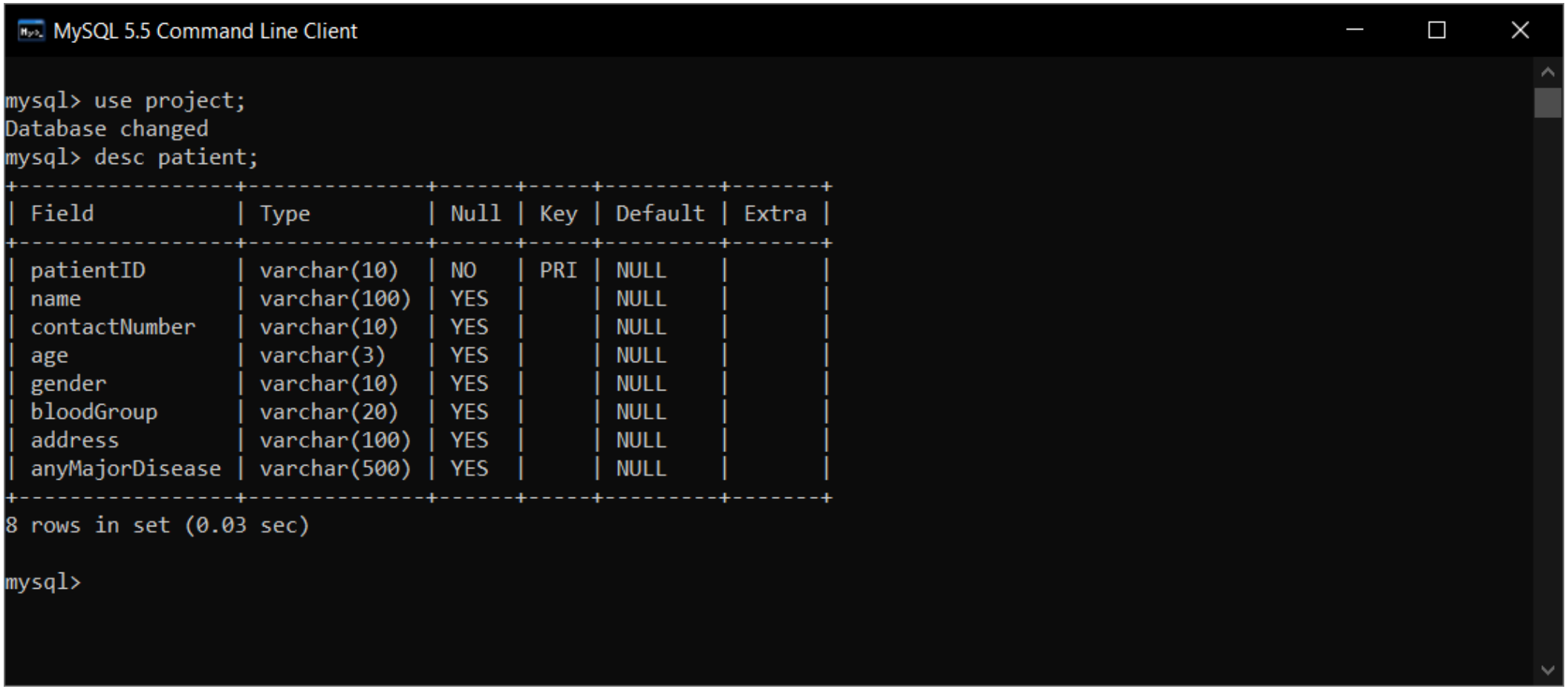
ER Diagram:


gender
phone
name
age
ID
blood group
address
major disease
Patient
1
has
M
diagnosis
medicines
symptoms
ID
ward Req
type of ward
Patient Report
doctor ID
N
is assigned to
1
type
address
ID
name
age
phone
licenseID
Doctor

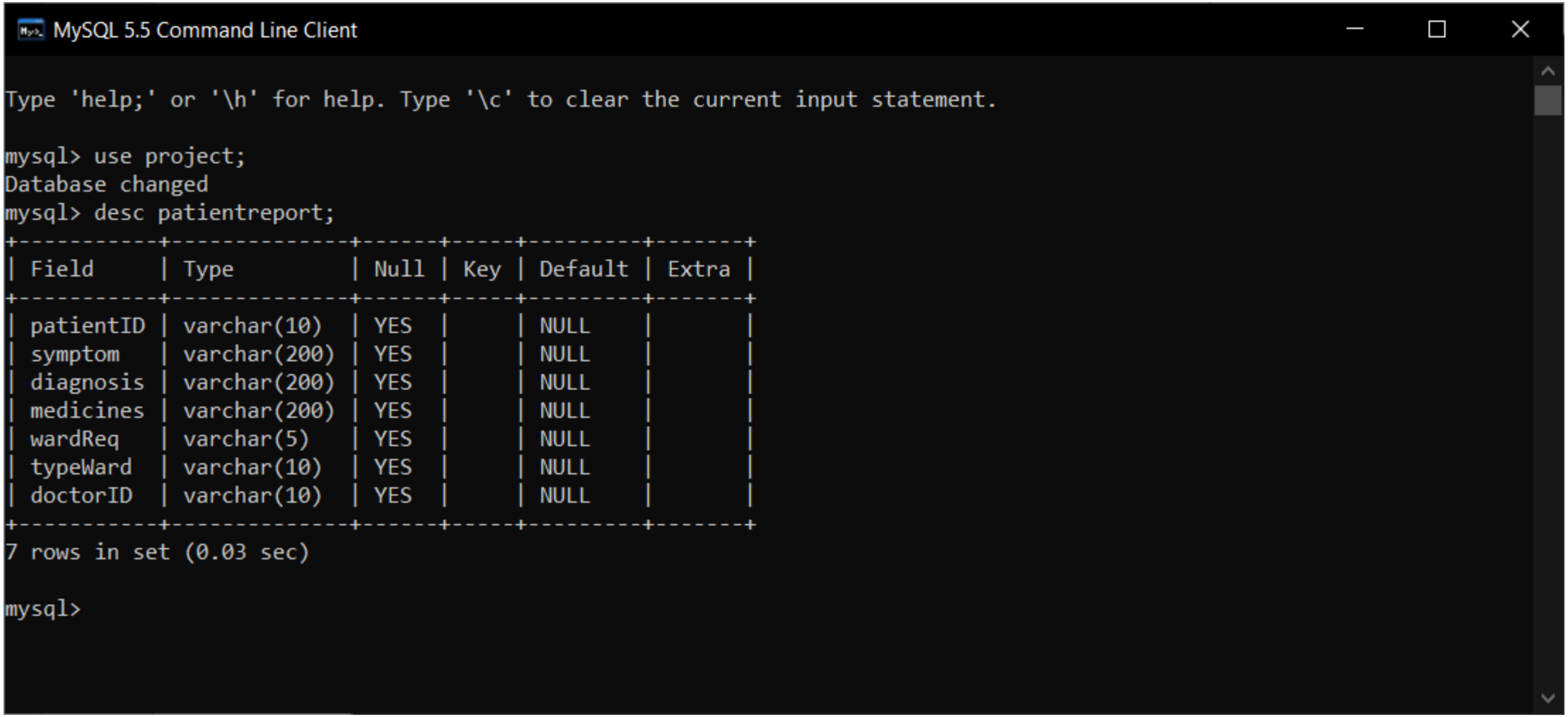
Relational Schema:



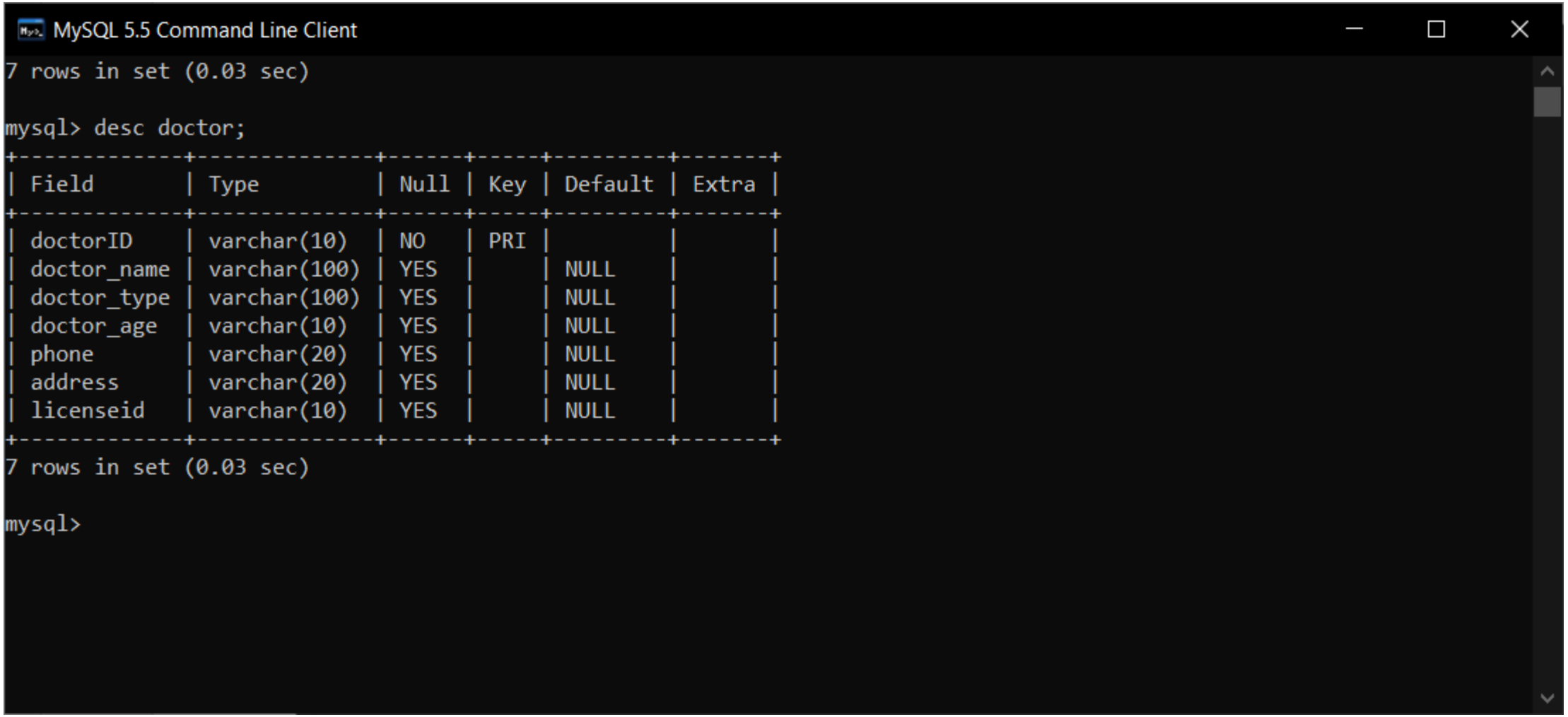
Patient table:



Patient Report table:



Doctor table:



Queries:

- For creating the tables:

create table <table-name> (c1 varchar(100), c2 varchar(100) not null, ... cn varchar(100), CONSTRAINT PRIMARY KEY (c1));

Add a new patient to the patient database:

- "insert into patient values ('"+patientID+"','"+name+"','"+contactNumber+"','"+age+"','"+gender+"','"+bloodGroup+"','"+address+"','"+anyMajorDisease+"')"

Add the patient symptoms and other details in the database:

- First, we check if the patient and doctor mentioned are present in the database:

- "select * from patient where patientID = '"+patientID+"'"

- "select * from Doctor where doctorID = '"+doctorID+"';"

- If both are present then we add the details:

- "insert into patientreport values ('"+patientID+"','"+symptoms+"','"+diagnosis+"','"+medicines+"','"+ward_req+"','"+ward_type+"','"+doctorID+"')"

To display all the details of the patients and the doctors we use:

- "select * from Doctor"
- And for patients:
- "select patient.*, patientreport.*, Doctor.doctor_name, Doctor.phone from patient, patientreport, Doctor where patient.patientID = patientreport.patientID and Doctor.doctorID = patientreport.doctorID"

To update a patient record we use:

- "select * from patient where patientID = '"+pid+"'");
- Then if the patient is present in the database, then we update the details:
- "update patient set name = '"+name+"', contactNumber = '"+contactno+"', age = '"+age+"', gender = '"+gender+"', bloodGroup = '"+bloodgroup+"', address = '"+address+"', anyMajorDisease = '"+anymajordisease+"' where patientID = '"+patientID+"'"

To add a new doctor to the database:

- "insert into Doctor values ("+id+"',"+name+"',"+type+"',"+age+"',"+ph+"',"+addr+"',"+lno+"')"

User Interface

- We begin at the login screen.

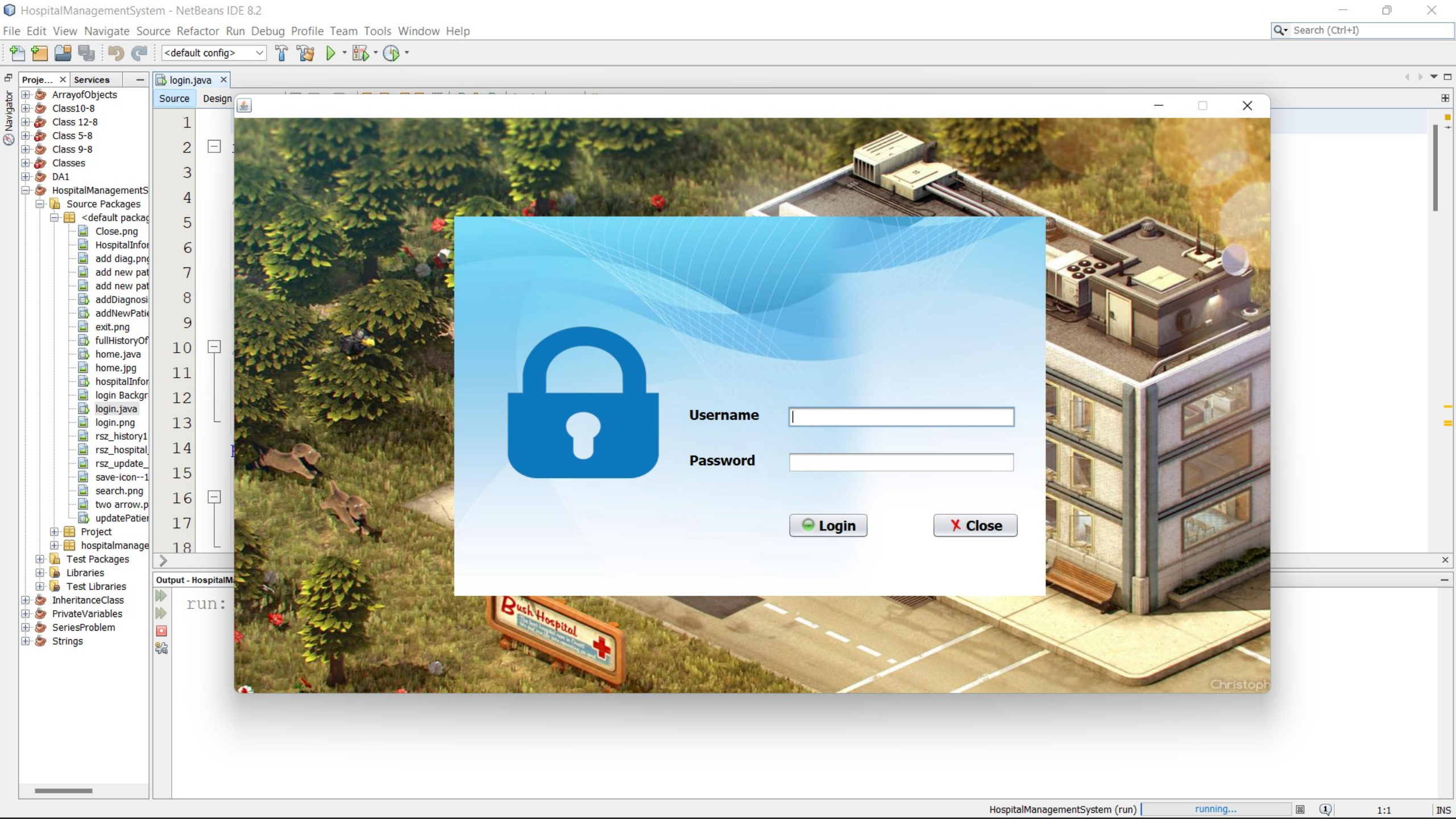
- If the user is a doctor, then he/she has to enter the doctor’s credentials to go to the patient’s page. It the user is an administrator then he/she has to enter the admin username and password to go to admin page.

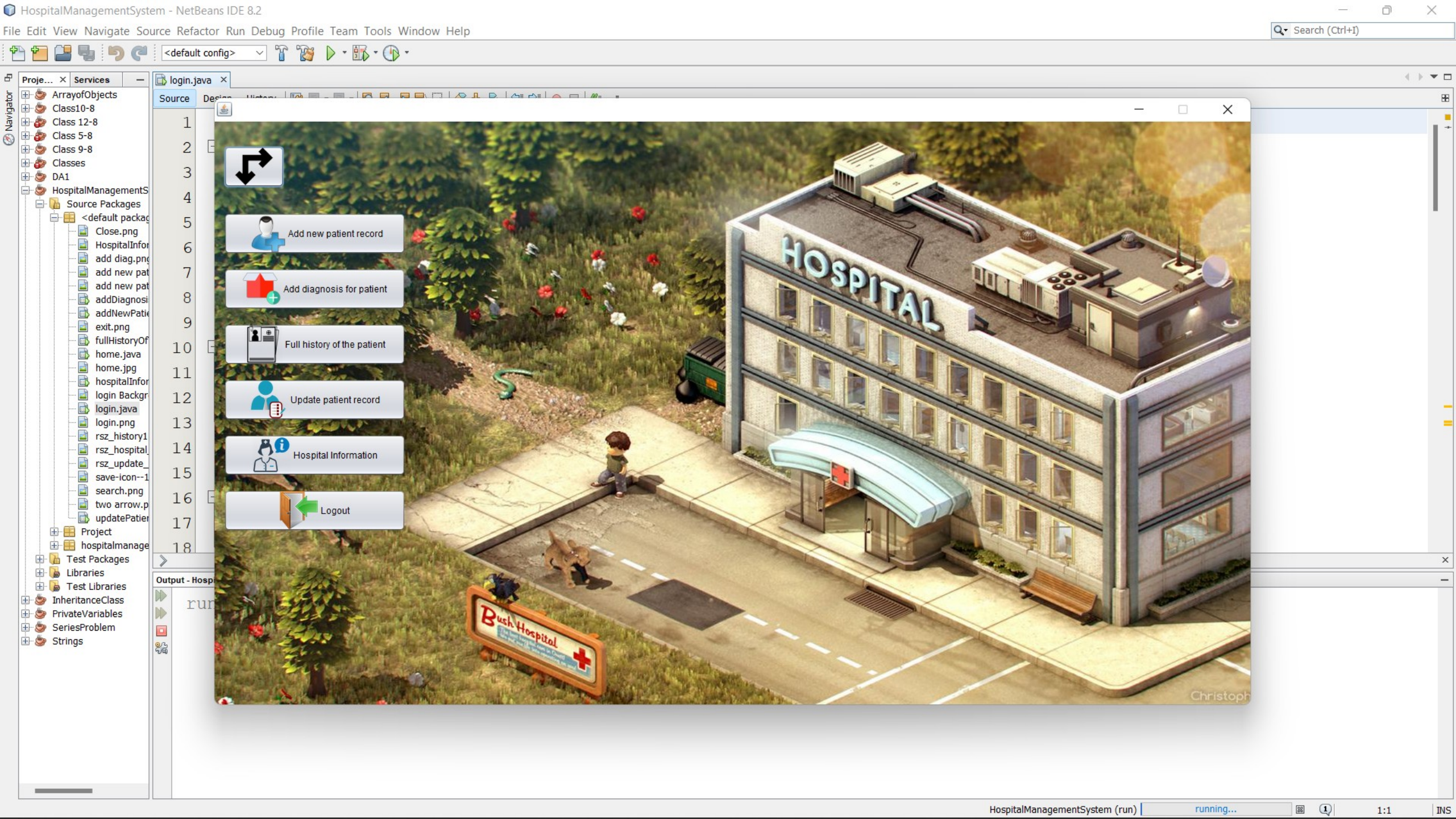
In the doctor interface we have:

1. Add new patient record.
2. Add diagnosis for the patient.
3. Check history of the patient.
4. Update patient record.
5. Hospital Information.
6. Exit to login screen.

- In the admin interface we have:

1. Add a doctor.
2. See all doctors.
3. Exit to login screen.


HospitalManagementSystem - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Proje... Services
login.java
Source Design
ArrayofObjects
Class10-8
Class 12-8
Class 5-8
Class 9-8
Classes
DA1
HospitalManagementS
Source Packages
<default packag
Close.png
HospitalInfor
add diag.png
add new pat
add new pat
addDiagnosi
addNewPatie
exit.png
fullHistoryOf
home.java
home.jpg
hospitalInfor
login Backgr
login.java
login.png
rsz_history1
rsz_hospital_
rsz_update_
save-icon--1
search.png
two arrow.p
updatePatier
Project
hospitalmanage
Test Packages
Libraries
Test Libraries
InheritanceClass
PrivateVariables
SeriesProblem
Strings
Username
Password
Login
Close
Bush Hospital
Output - HospitalM
run:
HospitalManagementSystem (run)
running...
1:1
INS

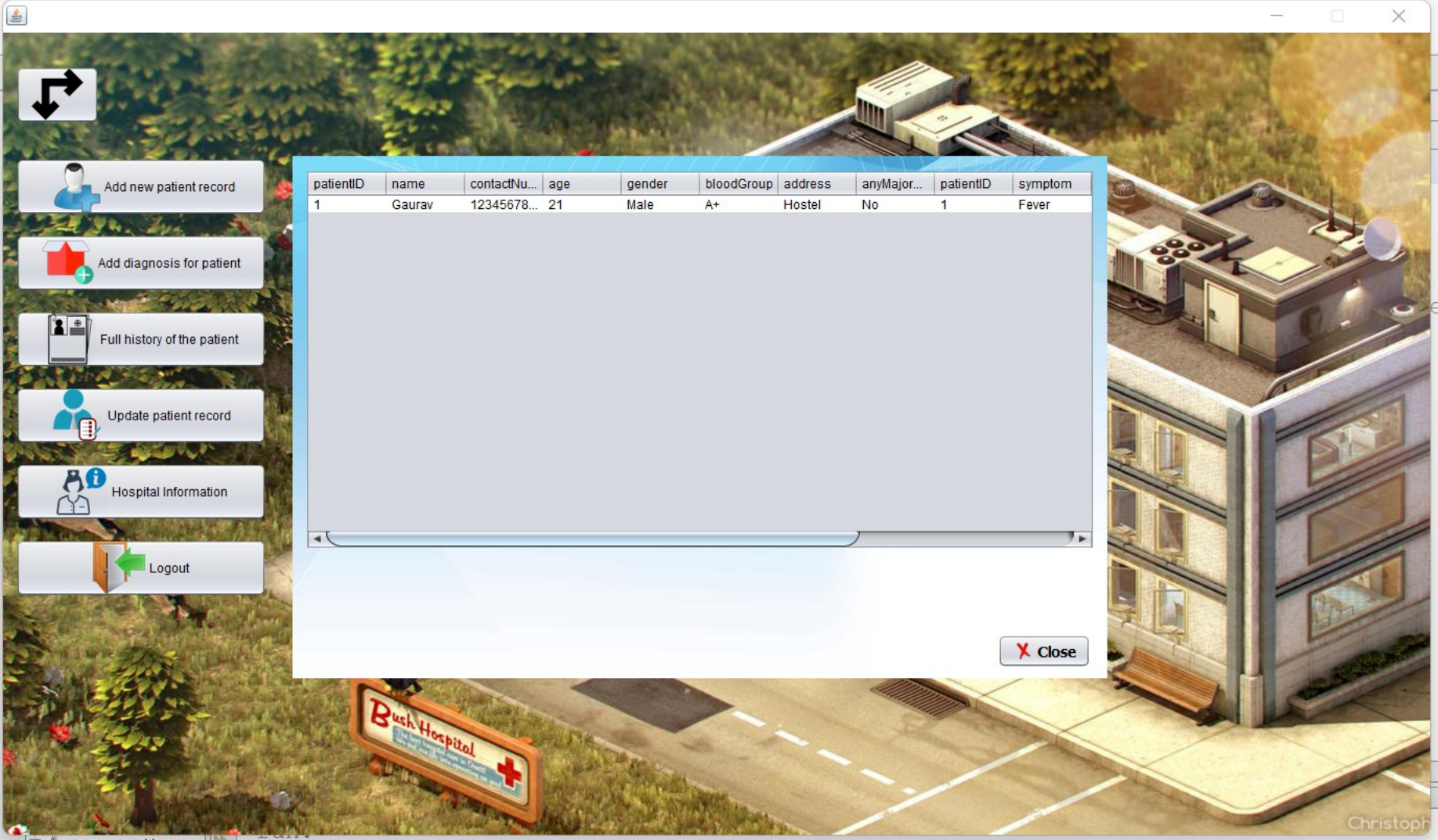

HospitalManagementSystem - NetBeans IDE 8.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Proje... Services
ArrayofObjects
Class10-8
Class 12-8
Class 5-8
Class 9-8
Classes
DA1
HospitalManagementS
Source Packages
<default packag
Close.png
HospitalInfor
add diag.png
add new pat
add new pat
addDiagnosi
addNewPatie
exit.png
fullHistoryOf
home.java
home.jpg
hospitalInfor
login Backgr
login.java
login.png
rsz_history1
rsz_hospital_
rsz_update_
save-icon--1
search.png
two arrow.p
updatePatier
Project
hospitalmanage
Test Packages
Libraries
Test Libraries
InheritanceClass
PrivateVariables
SeriesProblem
Strings
login.java
Source
Add new patient record
Add diagnosis for patient
Full history of the patient
Update patient record
Hospital Information
Logout
HOSPITAL
Bush Hospital
Output - Hospi
HospitalManagementSystem (run)
running...
1:1
INS

PrivateVariables
SeriesProblem
Strings

HospitalManagementSystem (run) running... 1:1 INS

PrivateVariables
SeriesProblem
Strings

HospitalManagementSystem (run) running... 1:1 INS



Conclusion

To conclude, we have made an hospital management system using Java JDBC and MySQL.

Future Work:

1. Can add more tables making it a bit bigger.

2. Convert all the tables into a higher normal form.

3. Try NoSQL instead of SQL.

4. Run the program on a website with a server.

Sources:

1. https://www.youtube.com/watch?v=LaOpvTj-Kcc
2. https://www.youtube.com/watch?v=HySWyiIfU1M
3. https://www.youtube.com/watch?v=WuBcTJnIuzo